Non-asymptotic Approximation Error Bounds of Parameterized Quantum Circuits

Zhan Yu[1, 2*] **Qiuhao Chen**[1*] **Yuling Jiao**[1, 3] **Yinan Li**[1, 3†] **Xiliang Lu**[1, 3]
Xin Wang[4] **Jerry Zhijian Yang**[1, 3‡]

[1] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China
[2] Centre for Quantum Technologies, National University of Singapore, 117543, Singapore
[3] Hubei Key Laboratory of Computational Science, Wuhan 430072, China
[4] Thrust of Artificial Intelligence, Information Hub,
Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511453, China

Abstract

Parameterized quantum circuits (PQCs) have emerged as a promising approach for quantum neural networks. However, understanding their expressive power in accomplishing machine learning tasks remains a crucial question. This paper investigates the expressivity of PQCs for approximating general multivariate function classes. Unlike previous Universal Approximation Theorems for PQCs, which are either nonconstructive or rely on parameterized classical data processing, we explicitly construct data re-uploading PQCs for approximating multivariate polynomials and smooth functions. We establish the first non-asymptotic approximation error bounds for these functions in terms of the number of qubits, quantum circuit depth, and number of trainable parameters. Notably, we demonstrate that for approximating functions that satisfy specific smoothness criteria, the quantum circuit size and number of trainable parameters of our proposed PQCs can be smaller than those of deep ReLU neural networks. We further validate the approximation capability of PQCs through numerical experiments. Our results provide a theoretical foundation for designing practical PQCs and quantum neural networks for machine learning tasks that can be implemented on near-term quantum devices, paving the way for the advancement of quantum machine learning.

1 Introduction

In quantum computing, one key area is to investigate if quantum computers could accelerate classical machine learning tasks in data analysis and artificial intelligence, giving rise to an interdisciplinary field known as *quantum machine learning* [1]. As the quantum analogs of classical neural networks, *parameterized quantum circuits* (PQCs) [2] have gained significant attention as a prominent paradigm to yield quantum advantages. PQCs offer a concrete and practical way to implement quantum machine learning algorithms in noisy and intermediate-scale quantum (NISQ) devices [3], rendering them well-suited for a diverse array of tasks [4–11].

To establish the practical significance of quantum machine learning, an ongoing pursuit is to demonstrate their superiority in solving real-world learning problems compared to classical learning models, including the most commonly used deep neural networks [12]. Typical supervised learning tasks, such as image classification and price prediction, aim to construct a model to learn a mapping

[*] Z.Y. and Q.C. contributed equally to this work
[†] Yinan.Li@whu.edu.cn
[‡] zjyang.math@whu.edu.cn

function from the input to output via training data sets. Essentially, the goal is to approximate multivariate functions. This viewpoint leads to the celebrated *Universal Approximation Theorem* [13, 14], which limits what neural networks can theoretically learn. Recently, powerful tools from approximation theory have been utilized to establish a fruitful mathematical framework for understanding the "black magic" of deep learning by establishing non-asymptotic approximation error bounds of deep neural networks in terms of the *width*, *depth*, *number of weights (neurons)* and function complexities, see e.g. Refs. [15–25] and references therein.

Substantial investigations have showcased the power of quantum machine learning for specific learning tasks [26–33]. A fundamental question is whether the *expressivity* of quantum machine learning models is as powerful as, or is more powerful than, the expressivity of classical machine learning models. This can be illustrated by proving universal approximation theorems for PQCs [34–41], indicating that there exist PQCs with suitable parameter configurations to approximate target functions up to a given approximation accuracy. This will justify the power of PQCs to solve supervised learning tasks in a mathematical way. To further investigate whether PQCs are more expressive than the classical models or not, it is natural to examine the PQC approximation performance by establishing approximation error bounds for important function classes. Such quantitative error bounds are less known in the quantum setting, because the hypothesis functions generated by PQCs are more complicated than those generated by classical neural networks.

The difficulties of analyzing the PQC approximation performances can be partially overcome by allowing *parameterized classical data processing*. Namely, trainable parameters are allowed not only in the quantum gates in PQCs but also in the classical data pre- and post-processing. This allows one to prove approximation error bounds following classical strategies [39, 41, 40]. For instance, Goto et al. [39] proved PQC approximation error rate for Lipschitz continuous functions in terms of the number of qubits and trainable parameters by incorporating trainable parameters in the measurement post-processing phase; similar results can also be obtained by utilizing Tensor-Train Network [41] or by linear transformations to preprocess the classical data.

However, utilizing parameterized classical data processing makes it hard to distinguish whether the expressive power of PQCs comes from the classical or quantum parts. In fact, parameterized classical data processing enables one to directly convert the hypothesis functions generated by the quantum models into hypothesis functions generated by classical ones and adapt expressivity results for classical machine learning models to extract the expressivity of such quantum models. As a consequence, the resulting PQCs have very simple structures and short depth. It remains unknown whether one can prove approximation error bounds for PQCs without parameterized classical data processing. On the other hand, Zhao et al. [42] proved exponential lower bounds on the number of trainable parameters (in terms of the number of variables) needed for approximating bounded Lipschitz continuous functions using PQCs without parameterized classical data processing, illustrating that using PQCs to approximate Lipschitz functions still suffers from the *curse of dimensionality* (CoD) met by classical deep neural networks [43]. However, this does not rule out the possibility that one can achieve the same approximation rate with PQCs of *smaller size* compared to classical deep neural networks.

In this paper, we explicitly construct *the first* PQCs *without* parameterized classical data processing for approximating multivariate polynomials and smooth functions; a glance at these constructed PQCs is illustrated in Fig. 1. This eliminates the ambiguity regarding whether the expressivity originates from classical or quantum parts. We also establish *non-asymptotic PQC approximation error bounds*, in the sense that the PQC approximation performances are characterized in terms of the number of qubits (width), the *depth of PQCs*, the number of trainable parameters/gates (parameter count), and the function complexities. These results enable us to compare the approximation power of PQCs with that of classical neural networks. Notably, we show that for multivariate smooth functions, the quantum circuit size and the number of trainable parameters of our proposed PQCs demonstrate an improvement over the prior result of deep ReLU neural networks [21], one of the most commonly used neural network family in classical deep learning theory. Our proposed PQCs not only possess the universal approximation property but also achieve parameter efficiency comparable to classical neural networks, potentially leading to more efficient and scalable quantum machine learning algorithms for real-world tasks.

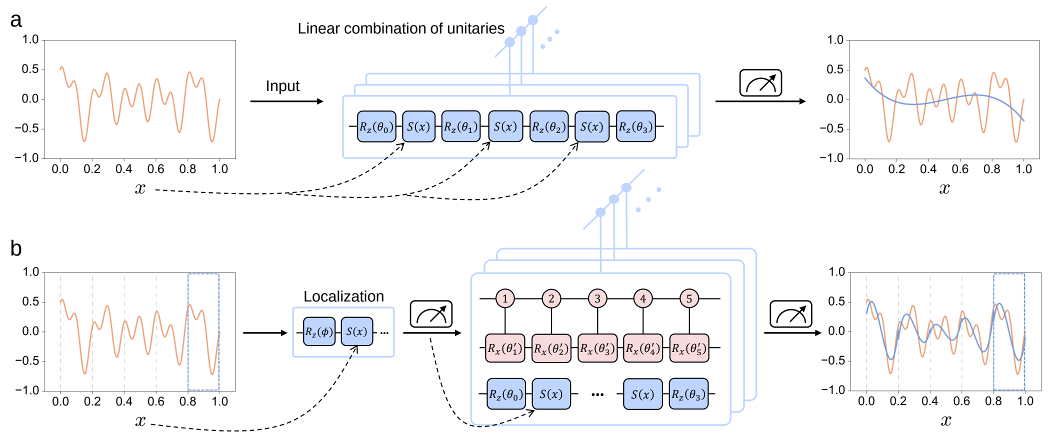


Figure 1: **Overview of PQCs for approximating continuous functions.** (a) Flowchart illustrating the strategy for using PQCs to approximate continuous functions via implementing Bernstein polynomials. The input data x is encoded into the PQC through $S(x)$, with the PQC (blue background) capable of representing parity-constrained polynomials up to degree 3 (as x is encoded three times). The technique of linear combination of unitaries (LCU) is used to aggregate these polynomials together. The output of PQC derives from measurement with a specific observable. Fine-tuning trainable parameters in R_Z gates yields a polynomial output depicted in the right panel. (b) Flowchart illustrating the strategy of approximation via local Taylor expansions. We first apply a PQC to localize the input domain into $K = 5$ regions. For example, for input $x \in [0.8, 1]$, PQC outputs $x' = 0.8$ as a fixed point. Then $x - x'$ will be fed into a new PQC for implementing the local Taylor expansions at the fixed point x', forming a nesting architecture. Control gates with pink backgrounds implement the Taylor coefficients. Fine-tuning trainable parameters in R_X and R_Z gates yields a piecewise polynomial with degree 3 that approximates the target function.

2 Preliminaries

Quantum states. The basic unit of information in quantum computing is the *qubit*, which can exist in a superposition of the states 0 and 1 simultaneously, unlike classical bits that are restricted to either 0 or 1. A pure quantum state in the d-dimensional Hilbert space $\mathbb{C}^d$ is represented by the *Dirac* notation $|\phi\rangle$. The conjugate transpose of $|\phi\rangle$ is denoted by $\langle\phi|$. The inner product of two quantum states $|\phi\rangle$ and $|\psi\rangle$ is written as $\langle\phi|\psi\rangle$. An important property is that $\langle\phi|\phi\rangle = 1$ for any pure state $|\psi\rangle$. By convention, the computational basis states for single-qubit systems are written as $|0\rangle = [1, 0]^T$ and $|1\rangle = [0, 1]^T$, where the superscript T denotes the transpose. For n-qubit systems, the computational basis states are expressed as $|j\rangle \in \{|0\rangle, |1\rangle\}^{\otimes n}$, where $\otimes$ denotes the tensor product operation.

Quantum gates. Quantum gates are building blocks of quantum circuits operating on quantum states. Unlike classical gates, quantum gates are reversible and described as unitary matrices. In quantum machine learning, common parameterized quantum gates include single-qubit Pauli rotation gates $R_X(\theta) = e^{-\theta X/2}$, $R_Y(\theta) = e^{-\theta Y/2}$, and $R_Z(\theta) = e^{-\theta X/2}$ that rotate a quantum state through angle θ around the corresponding axis, where the three Pauli operators are defined as:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

where i represents the imaginary unit. Commonly used two-qubit quantum gates include CNOT gate that flips the target qubit if and only if the the control qubit is in $|1\rangle$.

Quantum measurement The quantum measurement is a procedure manipulating a quantum system to extract classical information. The simplest measurement is the computational basis measurement: For a single-qubit system $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the outcome is either $|0\rangle$ with probability $|\alpha|^2$ or $|1\rangle$ with probability $|\beta|^2$. These measurements project the quantum state onto the measured basis, collapsing the state itself. Observables, represented by Hermitian operators, correspond to measurable quantities in a quantum system like energy or position. Each observable has a set of possible outcomes (eigenvalues) and corresponding states (eigenvectors). When a measurement of an observable is performed, the outcome is one of the eigenvalues, and the state of the system collapses to the corresponding eigenvector. If we are measuring a state $|\psi\rangle$ using observable $\mathcal{O}$, the expected value of outcome is $\langle\psi|\,\mathcal{O}\,|\psi\rangle$. This represents the average result one would expect from repeated measurements on identically prepared systems. A comprehensive introduction to the fundamental notations and concepts of quantum computation can be found in [44].

Data re-uploading PQCs. The PQCs we shall construct in this paper are of *data re-uploading* type [11], i.e., consisting of interleaved data encoding circuit blocks and trainable circuit blocks. More precisely, let $\boldsymbol{x}$ be the input data vector and $\boldsymbol{\theta} = (\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_L)$ be a set of trainable parameter vectors. $S(\boldsymbol{x})$ is a quantum circuit that encode $\boldsymbol{x}$ and $V(\boldsymbol{\theta}_j)$ is a trainable quantum circuit with trainable parameter vector $\boldsymbol{\theta}_j$. An L-layer data re-uploading PQC can be then expressed as

$$U_{\boldsymbol{\theta}}(\boldsymbol{x}) = V(\boldsymbol{\theta_0}) \prod_{j=1}^{L} S(\boldsymbol{x}) V(\boldsymbol{\theta_j}), \tag{1}$$

Applying $U_{\boldsymbol{\theta}}(\boldsymbol{x})$ to an initial quantum state and measuring the output states provides a way to express functions on $\boldsymbol{x}$:

$$f_{U_{\boldsymbol{\theta}}}(\boldsymbol{x}) \coloneqq \langle 0|\, U^{\dagger}_{\boldsymbol{\theta}}(\boldsymbol{x}) \mathcal{O} U_{\boldsymbol{\theta}}(\boldsymbol{x})\, |0\rangle\,, \tag{2}$$

where $\mathcal{O}$ is some Hermitian observable. The *approximation capability* of the PQC $U_{\boldsymbol{\theta}}(\boldsymbol{x})$ can be characterized by the classes of functions that $f_{U_{\boldsymbol{\theta}}}(\boldsymbol{x})$ can approximate by tuning the trainable parameter vector $\boldsymbol{\theta}$. We then turn to an example of single-qubit PQCs approximating univariate functions. For the input $x \in [-1, 1]$, we utilized the Pauli X basis encoding scheme [10] and defined the data encoding operator as a Pauli X rotation $S(x) \coloneqq e^{i \arccos(x) X}$. Interleaving the data encoding unitary $S(x)$ with some parameterized Pauli Z rotations $R_Z(\theta)$ gives the circuit of data re-uploading PQC for one variable as $U_{\boldsymbol{\theta}}(x) \coloneqq R_Z(\theta_0) \prod_{j=1}^{L} S(x) R_Z(\theta_j)$ where $\boldsymbol{\theta} = (\theta_0, \ldots, \theta_L) \in \mathbb{R}^{L+1}$ is a set of trainable parameters. Utilizing results from quantum signal processing [45–47], there exists $\boldsymbol{\theta} \in \mathbb{R}^{L+1}$ such that $U_{\boldsymbol{\theta}}(x)$ implements polynomial transformations $p(x) \in \mathbb{R}[x]$ as $p(x) = \langle +|U_{\boldsymbol{\theta}}(x)|+\rangle$ for any $x \in [-1, 1]$ if and only if the degree of $p(x)$ is at most L, the parity of $p(x)$ is $L \bmod 2$ [4], and $|p(x)| \leq 1$ for all $x \in [-1, 1]$. Then, univariate functions that could be approximated by the specified polynomial $p(x)$ could also be approximated by the PQC $U_{\boldsymbol{\theta}}(x)$. Other than the real polynomials, there are also types of single-qubit PQC with Pauli Z basis encoding that could implement complex trigonometric polynomials [37].

3 Expressivity of PQCs for multivariate continuous functions

3.1 Explicit construction of PQCs for multivariate polynomials

Although PQCs for approximate univariate functions have been constructed and analyzed, they have not yet been generally extended to the case of multivariate functions. Current proofs of universal approximation for multivariate functions are nonconstructive [34, 38] and require arbitrary circuit width, arbitrary multi-qubit global parameterized unitaries, and arbitrary observables. Goto et al. [39] proposed several constructions for approximating multivariate functions with the assistance of parameterized data pre-processing and post-processing, yielding a quantum-enhanced hybrid scheme rather than a purely quantum setting.

We now move to our explicit construction of PQCs for multivariate polynomials. A multivariate polynomial with d variables and degree s is defined as $p(\boldsymbol{x}) \coloneqq \sum_{\|\boldsymbol{\alpha}\|_1 \leq s} c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}$ where $\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d}$. To implement the multivariate polynomial $p(\boldsymbol{x})$, we first build a PQC to express a monomial $c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}$. The construction is a trivial extension of the univariate case: We simply apply the single-qubit PQC with Pauli X basis encoding on each x_j to implement $x_j^{\alpha_j}$ for $1 \leq j \leq d$, respectively. The coefficient $c_{\boldsymbol{\alpha}} \in \mathbb{R}$ could be implemented by any of these PQCs. Thus we could construct a PQC $U^{\boldsymbol{\alpha}}(\boldsymbol{x}) \coloneqq \bigotimes_{j=1}^{d} U_{\boldsymbol{\theta}_j}(x_j)$ such that $\langle +|^{\otimes d} U^{\boldsymbol{\alpha}}(\boldsymbol{x}) |+\rangle^{\otimes d} = c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}$. The depth of the PQC $U^{\boldsymbol{\alpha}}(\boldsymbol{x})$ is at most $2s + 1$, the width is at most d, and the number of parameters is at most $s + d$.

[4] A polynomial $p(x)$ has parity 0 if all coefficients corresponding to odd powers of x are 0, and similarly $p(x)$ has parity 1 if all coefficients corresponding to even powers of x are 0.

Having PQCs that implement monomials, the next step is to aggregate monomials to implement the multivariate polynomial. A natural idea is to sum the monomial PQCs together as $U_p(\boldsymbol{x}) = \sum_{\|\boldsymbol{\alpha}\|_1 \leq s} U^{\boldsymbol{\alpha}}(\boldsymbol{x})$. However, the addition operation in quantum computing is non-trivial as the sum of unitary operators is not necessarily unitary. To overcome this issue, we utilize *linear combination of unitaries* (LCU) [48] to implement the operator $U_p(\boldsymbol{x})$ on a quantum computer. Realizing the linear combination of PQCs $U^{\boldsymbol{\alpha}}(\boldsymbol{x})$ requires applying multi-qubit control on each $U^{\boldsymbol{\alpha}}(\boldsymbol{x})$, which could be further decomposed into linear-depth quantum circuits of CNOT gates and single-qubit rotation gates without using any ancilla qubit [49]. Then we can obtain the polynomial $p(\boldsymbol{x}) = \langle +|^{\otimes d} U_p(\boldsymbol{x}) |+\rangle^{\otimes d}$ by applying the Hadamard test on the LCU circuit. Summarizing the above, we establish the following theorem about using PQCs to implement multivariate polynomials. A formal description of such PQCs is given in Appendix B.

Theorem 1. *For any multivariate polynomial $p(\boldsymbol{x})$ with d variables and degree s such that $|p(\boldsymbol{x})| \leq 1$ for $\boldsymbol{x} \in [0,1]^d$, there exists a PQC $W_p(\boldsymbol{x})$ such that*

$$f_{W_p}(\boldsymbol{x}) := \langle 0| W_p^\dagger(\boldsymbol{x}) Z^{(0)} W_p(\boldsymbol{x}) |0\rangle = p(\boldsymbol{x}) \tag{3}$$

where $Z^{(0)}$ is the Pauli Z observable on the first qubit. The width of the PQC is $O(d + \log s + s \log d)$, the depth is $O(s^2 d^s (\log s + s \log d))$, and the number of parameters is $O(sd^s(s+d))$.

Note that the initial state in the Hadamard test is $|0\rangle^{\otimes d}$ since $|+\rangle^{\otimes d}$ could be easily prepared by applying Hadamard gates on $|0\rangle^{\otimes d}$. Measuring the first qubit of $W_p(\boldsymbol{x})$ for $O(\frac{1}{\varepsilon^2})$ times is needed to estimate the value of $p(\boldsymbol{x})$ up to an additive error ε. We could further use the amplitude estimation algorithm [50] to reduce the overhead while increasing the circuit depth by $O(\frac{1}{\varepsilon})$.

3.2 PQC approximation for continuous functions

Polynomials play a central role in approximation theory. The celebrated Weierstrass approximation theorem (see e.g. [51, Sec. 10.2.2]) indicates that polynomials are sufficient to approximate continuous univariate functions. For multivariate functions, their approximation can be implemented using Bernstein polynomials [52, 53]. We shall apply these results to prove PQC approximation error bounds for multivariate Lipschitz continuous functions.

For a d-variable continuous function $f : [0,1]^d \to \mathbb{R}$, the multivariate Bernstein polynomial with degree $n \in \mathbb{N}^+$ of f is defined as

$$B_n(\boldsymbol{x}) := \sum_{k_1=0}^{n} \cdots \sum_{k_d=0}^{n} f\left(\frac{\boldsymbol{k}}{n}\right) \prod_{j=1}^{d} \binom{n}{k_j} x_j^{k_j} (1-x_j)^{n-k_j}, \tag{4}$$

where $\boldsymbol{k} = (k_1, \ldots, k_d) \in \{0, \ldots, n\}^d$. It is known that Bernstein polynomials converge uniformly to f on $[0,1]^d$ as $n \to \infty$ [52, 53]. The PQC constructed in Theorem 1 could implement the Bernstein polynomial with proper rescaling, which implies that the PQC is a universal approximator for any bounded continuous functions.

Theorem 2 (The Universal Approximation Theorem of PQC)**.** *For any continuous function $f : [0,1]^d \to [-1,1]$, given an $\varepsilon > 0$, there exist an $n \in \mathbb{N}$ and a PQC $W_b(\boldsymbol{x})$ with width $O(d \log n)$, depth $O(dn^d \log n)$ and the number of trainable parameters $O(dn^d)$ such that*

$$|f(\boldsymbol{x}) - f_{W_b}(\boldsymbol{x})| \leq \varepsilon \tag{5}$$

for all $\boldsymbol{x} \in [0,1]^d$, where $f_{W_b}(\boldsymbol{x}) := \langle 0| W_b^\dagger(\boldsymbol{x}) Z^{(0)} W_b(\boldsymbol{x}) |0\rangle$.

Theorem 2 serves as the quantum counterpart to the universal approximation theorem of classical neural networks. Moreover, the PQCs that universally approximate continuous functions are explicitly constructed without any impractical assumption, improving the previous results presented in Refs. [34, 38]. Moreover, for continuous functions f satisfying the Lipschitz condition, $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq \ell \|\boldsymbol{x} - \boldsymbol{y}\|_\infty$ for any $\boldsymbol{x}, \boldsymbol{y}$, the approximation rate of Bernstein polynomials could be quantitatively characterized in terms of the degree n, the number of variables d and the Lipschitz constant ℓ [53]. Thus a non-asymptotic error bound for PQC approximating Lipschitz continuous functions could be obtained as follows.

Theorem 3. *Given a Lipschitz continuous function $f : [0,1]^d \to [-1,1]$ with a Lipschitz constant ℓ, for any $\varepsilon > 0$ and $n \in \mathbb{N}$, there exists a PQC $W_b(\boldsymbol{x})$ with such that $f_{W_b}(\boldsymbol{x}) := \langle 0| W_b^\dagger(\boldsymbol{x}) Z^{(0)} W_b(\boldsymbol{x}) |0\rangle$ satisfies*

$$|f(\boldsymbol{x}) - f_{W_b}(\boldsymbol{x})| \leq \varepsilon + 2\left(\left(1 + \frac{\ell^2}{n\varepsilon^2}\right)^d - 1\right) \leq \varepsilon + d2^d \frac{\ell^2}{n\varepsilon^2} \tag{6}$$

for all $\boldsymbol{x} \in [0,1]^d$. The width of the PQC is $O(d \log n)$, the depth is $O\big(dn^d \log n\big)$, and the number of parameters is $O(dn^d)$.

We prove these theorems in Appendix C. Although a quantitative approximation error bound is characterized in Theorem 3, we could find that n must be sufficiently large to obtain a good precision, yielding an extremely deep PQC. This inefficiency is essentially due to the intrinsic difficulty of using a single global polynomial to approximate a continuous function uniformly. A possible approach that may overcome the obstacle is to use local polynomials to achieve a piecewise approximation, which we will discover in the next section.

3.3 PQC approximation for Hölder smooth functions

To achieve a piecewise approximation of multivariate functions, we follow the path of classical deep neural networks approximation [18, 21, 25], which utilizes multivariate Taylor series to approximate target functions in small local regions.

We focus on Hölder smooth functions. Let $\beta = s + r > 0$, where $r \in (0,1]$ and $s \in \mathbb{N}^+$. For a finite constant $B_0 > 0$, the β-Hölder class of functions $\mathcal{H}^\beta([0,1]^d, B_0)$ is defined as

$$\mathcal{H}^\beta([0,1]^d, B_0) = \left\{ f\colon [0,1]^d \to \mathbb{R}, \max_{\|\boldsymbol{\alpha}\|_1 \leq s} \|\partial^{\boldsymbol{\alpha}} f\|_\infty \leq B_0, \max_{\|\boldsymbol{\alpha}\|_1 = s} \sup_{\boldsymbol{x} \neq \boldsymbol{y}} \frac{|\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}) - \partial^{\boldsymbol{\alpha}} f(\boldsymbol{y})|}{\|\boldsymbol{x} - \boldsymbol{y}\|_2^r} \leq B_0 \right\}, \tag{7}$$

where $\partial^{\boldsymbol{\alpha}} = \partial^{\alpha_1} \cdots \partial^{\alpha_d}$ for $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$. We note that Hölder smooth functions are natural generalizations of various continuous functions: When $\beta \in (0,1)$, f is Hölder continuous with order β and Hölder constant B_0; when $\beta = 1$, f is Lipschitz continuous with Lipschitz constant B_0; when $1 < \beta \in \mathbb{N}$, $f \in C^s([0,1]^d)$, the class of s-smooth functions whose s-th partial derivatives exist and are bounded. As shown in Petersen and Voigtlaender [18], for any β-Hölder smooth function $f \in \mathcal{H}^\beta([0,1]^d, B_0)$, its local Taylor expansion at some fixed point $\boldsymbol{x_0} \in [0,1]^d$ satisfies

$$\left| f(\boldsymbol{x}) - \sum_{\|\boldsymbol{\alpha}\|_1 \leq s} \frac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x_0})}{\boldsymbol{\alpha}!} (\boldsymbol{x} - \boldsymbol{x_0})^{\boldsymbol{\alpha}} \right| \leq d^s \|\boldsymbol{x} - \boldsymbol{x_0}\|_2^\beta \tag{8}$$

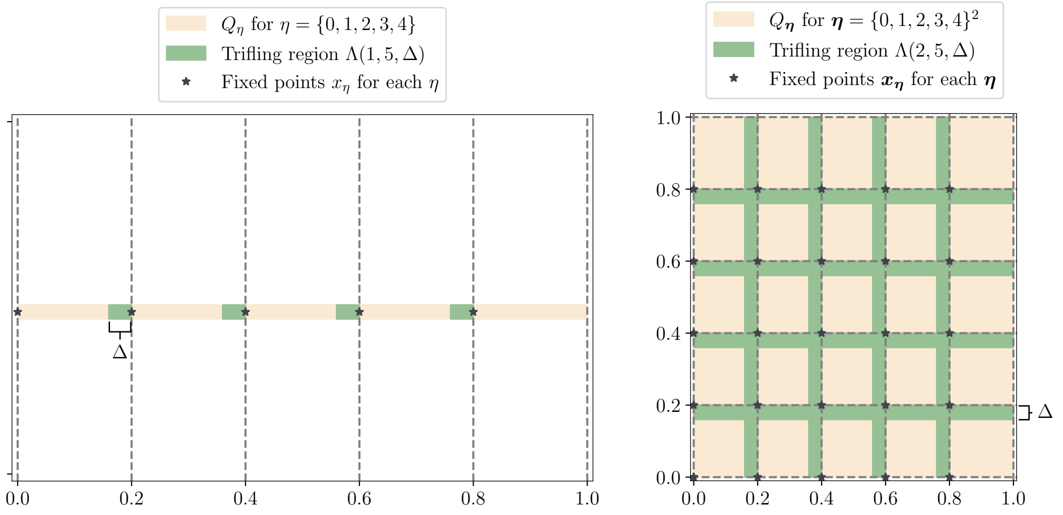
for all $\boldsymbol{x} \in [0,1]^d$, where $\boldsymbol{\alpha}! = \alpha_1! \cdots \alpha_d!$. Next, we show how to construct PQCs to implement the Taylor expansion of β-Hölder functions in the following three steps.

Localization. To utilize the Hölder smoothness, we need to first localize the entire region $[0,1]^d$. The motivation of localization is to determine the local point $\boldsymbol{x_0}$ in Eq. (8) so that the distance between $\boldsymbol{x}$ and $\boldsymbol{x_0}$ is fairly small. An intuitive configuration is illustrated in Fig. 2, where the stars represent the local points. Given $K \in \mathbb{N}$ and $\Delta \in (0, \frac{1}{3K})$, for each $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_d) \in \{0, 1, \ldots, K-1\}^d$, we define

$$Q_{\boldsymbol{\eta}} := \left\{ \boldsymbol{x} = (x_1, \ldots, x_d) : x_i \in \left[\frac{\eta_i}{K}, \frac{\eta_i + 1}{K} - \Delta \cdot 1_{\eta_i < K-1}\right] \right\}. \tag{9}$$

By the definition of $Q_{\boldsymbol{\eta}}$, the region $[0,1]^d$ is approximately divided into small hypercubes $\bigcup_{\boldsymbol{\eta}} Q_{\boldsymbol{\eta}}$ and some trifling region $\Lambda(d, K, \Delta) := [0,1]^d \setminus (\bigcup_{\boldsymbol{\eta}} Q_{\boldsymbol{\eta}})$, as illustrated in Fig. 2.

We construct a PQC that maps all $\boldsymbol{x} \in Q_{\boldsymbol{\eta}}$ to some fixed point $\boldsymbol{x_\eta} = \frac{\boldsymbol{\eta}}{K}$ in $Q_{\boldsymbol{\eta}}$, i.e., approximating the piecewise-constant function $D(\boldsymbol{x}) = \frac{\boldsymbol{\eta}}{K}$ if $\boldsymbol{x} \in Q_{\boldsymbol{\eta}}$. We describe our construction for $d = 1$, where $D(x) = \frac{k}{K}$ if $x \in [\frac{k}{K}, \frac{k+1}{K} - \Delta \cdot 1_{k<K-1}]$ for $k = 0, \ldots, K-1$. The multivariate case could be naturally generalized by applying $D(x)$ to each variable x_j. The idea is to construct a polynomial that approximates the function $D(x)$ based on the polynomial approximation to the sign function [54], which a single-qubit PQC can then implement. Generalizing to the multivariate localization, there exists a PQC $W_D(\boldsymbol{x})$ of depth $O(\frac{1}{\Delta} \log \frac{K}{\varepsilon})$ and width $O(d)$ such that the output $f_{W_D}(\boldsymbol{x})$ maps $\boldsymbol{x}$ to the corresponding fixed point $\boldsymbol{x_\eta}$ with precision ε. We can obtain an estimation of $\boldsymbol{\eta}$ using $\lfloor K f_{W_D}(\boldsymbol{x}) \rfloor$.



Q_η for $\eta = \{0,1,2,3,4\}$
Trifling region $\Lambda(1,5,\Delta)$
Fixed points x_η for each η
Δ
0.0
0.2
0.4
0.6
0.8
1.0
$Q_{\boldsymbol{\eta}}$ for $\boldsymbol{\eta} = \{0,1,2,3,4\}^2$
Trifling region $\Lambda(2,5,\Delta)$
Fixed points $\boldsymbol{x}_{\boldsymbol{\eta}}$ for each $\boldsymbol{\eta}$
Δ

Figure 2: **An illustration of localization.** The left panel demonstrates the localization $\bigcup_{\boldsymbol{\eta}} Q_{\boldsymbol{\eta}}$ for $K = 5$ and $d = 1$. The right panel shows the case of localization for $K = 5$ and $d = 2$. The "volume" of the trifling region $\Lambda(d, K, \Delta)$ is no more than $dK\Delta$.

Implementing the Taylor coefficients. Next, we use PQC to implement the Taylor coefficients $\xi_{\boldsymbol{\eta},\boldsymbol{\alpha}} := \frac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}_{\boldsymbol{\eta}})}{\boldsymbol{\alpha}!} \in [-1, 1]$ for each $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_d) \in \{0, 1, \ldots, K-1\}^d$ and $\boldsymbol{\alpha}$, which is essentially a point-fitting problem. Then we could construct a PQC $U_{co}^{\boldsymbol{\alpha}} = \sum_{\boldsymbol{\eta}} |\boldsymbol{\eta}\rangle\langle\boldsymbol{\eta}| \otimes R_X(\theta_{\boldsymbol{\eta},\boldsymbol{\alpha}})$ such that $\langle \boldsymbol{\eta}, 0| U_{co}^{\boldsymbol{\alpha}} |\boldsymbol{\eta}, 0\rangle = \xi_{\boldsymbol{\eta},\boldsymbol{\alpha}}$, where $|\boldsymbol{\eta}\rangle = |\eta_1\rangle \otimes \cdots \otimes |\eta_d\rangle$ and $\theta_{\boldsymbol{\eta},\boldsymbol{\alpha}} = 2\arccos(\xi_{\boldsymbol{\eta},\boldsymbol{\alpha}})$. The depth of $U_{\boldsymbol{\alpha}}$ is $O(K^d)$, the width is $O(d \log K)$, and the number of parameters is $O(K^d)$. Note that the state $|\boldsymbol{\eta}\rangle$ can be prepared using basis encoding on the provided $\boldsymbol{\eta} = \lfloor K f_{W_D}(\boldsymbol{x}) \rfloor$ from the localization step.

Implementing multivariate Taylor series. To implement the multivariate Taylor expansion of a function at some fixed point $\boldsymbol{x}_{\boldsymbol{\eta}}$, we first build a PQC to represent a single term in the Taylor series, which could be done by combining the PQC, which implements the Taylor coefficients and the PQC which implements monomials, i.e., constructing $U_{\boldsymbol{\eta}}^{\boldsymbol{\alpha}}(\boldsymbol{x}) := U_{co}^{\boldsymbol{\alpha}} \otimes U^{\boldsymbol{\alpha}}(\boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{\eta}})$. The depth of $U_{\boldsymbol{\eta}}^{\boldsymbol{\alpha}}(\boldsymbol{x})$ is $O(K^d + s)$, the width is $O(d \log K)$, and the number of parameters is at most $K^d + s + d$. The next step is to aggregate single Taylor terms together to implement the truncated Taylor expansion of the target function. We use LCU to construct the PQC $U_t(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}}) := \sum_{\|\boldsymbol{\alpha}\|_1 \leq s} U_{\boldsymbol{\eta}}^{\boldsymbol{\alpha}}(\boldsymbol{x})$ so that we can implement the Taylor expansion of the function f at point $\boldsymbol{x}_{\boldsymbol{\eta}}$ as $\langle \boldsymbol{\eta}, 0| \langle +|^{\otimes d} U_t(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{\eta}}) |\boldsymbol{\eta}, 0\rangle |+\rangle^{\otimes d}$. We construct a nested PQC as $U_t(\boldsymbol{x}, f_{W_D}(\boldsymbol{x}))$, such that for any input $\boldsymbol{x}$, the corresponding fixed point could be determined by the localization PQC. Such a PQC could be used, together with the Hadamard test, to approximate Hölder smooth functions. In particular, we prove the approximation error bound of our constructed PQC based on the error rate of Taylor expansion in Eq. (8).

Theorem 4. *Given a function $f \in \mathcal{H}^\beta([0,1]^d, 1)$ with $\beta = r + s$, $r \in (0, 1]$ and $s \in \mathbb{N}^+$, for any $K \in \mathbb{N}$ and $\Delta \in (0, \frac{1}{3K})$, there exists a PQC $W_t(\boldsymbol{x})$ such that $f_{W_t}(\boldsymbol{x}) := \langle 0| W_t^\dagger(\boldsymbol{x}) Z^{(0)} W_t(\boldsymbol{x}) |0\rangle$ satisfies*

$$|f(\boldsymbol{x}) - f_{W_t}(\boldsymbol{x})| \leq d^{s+\beta/2} K^{-\beta} \tag{10}$$

for $\boldsymbol{x} \in \bigcup_{\boldsymbol{\eta}} Q_{\boldsymbol{\eta}}$. The width of the PQC is $O(d \log K + \log s + s \log d)$, the depth is $O(s^2 d^s K^d (\log s + s \log d + d \log K)) + \frac{1}{\Delta} \log K)$, and the number of parameters is $O(sd^s(s + d + K^d) + \frac{d}{\Delta} \log K)$.

The proof can be found in Appendix D. Note that the PQC in Theorem 4 consists of two nested parts and its depth is counted as the sum of two PQCs for simplicity. We have established the uniform convergence property of PQCs for approximating Hölder smooth function on $[0, 1]^d$ except for the trifling region $\Lambda(d, K, \Delta)$. The Lebesgue measure of such a trifling region is no more than

$dK\Delta$. We can set $\Delta = K^{-d}$ with no influence on the size of the constructed PQC, and a similar approximation error bound in the entire region $[0,1]^d$ under the L^2 distance could be obtained.

4 Numerical experiments

This section presents numerical experiments to illustrate the expressivity of our proposed PQCs in approximating multivariate functions. We focus on approximating a bivariate polynomial function

$$f(x,y) = \frac{(x^2+y-1.5\pi)^2 + (x+y^2+\pi)^2 + (x+y-0.5\pi)^2}{5\pi^2},$$

over the domain $(x,y) \in [0,1]^2$. The approximation process involves two separate steps: (1) Learning a piecewise-constant function, $D(x) = \frac{k}{K}$ if $x \in [\frac{k}{K}, \frac{k+1}{K})$, using a single-qubit PQC, where $K \in \mathbb{N}^+$ determines the number of intervals for the piecewise-constant function. (2) Learning the Taylor expansion of $f(x,y)$ using multi-qubit PQCs based on Theorem 4. Both learning processes are implemented on a Gold 6248 2.50 GHz Intel(R) Xeon(R) CPU.

We randomly sample 200 data points within the domain $[0,1]$ to create training and test datasets for $D(x)$. A single-qubit PQC with adjustable parameters $L = 764$ ($L = 996$) is used to learn $D(x)$ with $K = 2$ ($K = 10$). Each parameter of the PQC is randomly initialized within the range $[0,\pi]$. We use the Adam optimizer [55] with a learning rate of 0.01 to minimize the Mean Squared Error (MSE) loss function during training. The training process was limited to a maximum of 300 iterations with a batch size of 100 data points. Early termination occurred if the MSE reached below 10^{-4}. The achieved MSE on the test data was 3.57×10^{-4} ($K = 2$) and 1.04×10^{-4} ($K = 10$). The numerical results are visualized in Fig. 3.

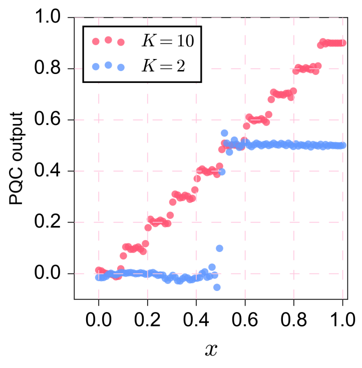


Figure 3: **Simulation results of localization.** We use single-qubit PQCs to approximate the localization function $D(x)$ for $K = 2$ and $K = 10$ respectively.

Similar to the previous step, we randomly sampled 200 data points within the domain $[0,1]^2$ to create training and test datasets for $f(x,y)$. A nested PQC structure was designed. It combined 12 two-qubit PQCs with a depth of 2, allowing the approximation of a degree-4 polynomial through a combination of lower-degree ones. Additionally, Taylor coefficients were stored in a separate matrix of size $K^2 \times 12$. The number of trainable parameters varied from 120 ($K = 2$) to 1272 ($K = 10$), each initialized randomly from $[0,\pi]$. The Adam optimizer with a learning rate of 0.01 was used to minimize the MSE loss during training. The training was limited to 500 iterations with a batch size of 100, with early termination for MSE below 10^{-4}. The achieved MSE on the test data was 2.22×10^{-4} ($K = 2$) and 9.82×10^{-5} ($K = 10$). Fig. 4 visualizes the results. As K increases, the PQC demonstrates improved approximation performance, aligning with the theoretical findings.

5 Discussion

To the best of our knowledge, our results establish the first explicit PQC constructions for approximating Lipschitz continuous and Hölder smooth functions with quantitative approximation error bounds. These results open up the possibility of comparing the size of PQCs and the size of classical deep neural networks for accomplishing the same function approximation tasks and see if there

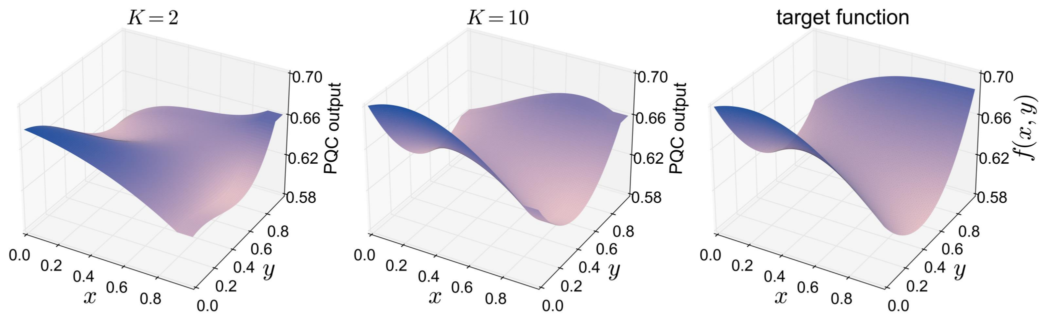


Figure 4: **Simulation results for learning $f(x, y)$.** The left two panels are derived by interpolating and smoothing the output values of PQC on 100 test data points.

is any quantum advantage in terms of the model size and the number of trainable parameters. Here, we mainly focus on the comparison with the results of approximation errors of classical machine learning models. In classical deep learning, the deep feed-forward neural network (FNN) equipped with the rectified linear unit (ReLU) activation function is one of the most commonly used models. The quantitative approximation error bounds of ReLU FNNs for approximating continuous functions have been recently established, including the nearly optimal approximation error bounds of ReLU FNNs for smooth functions [21]. We briefly compare the approximation errors of PQCs and ReLU FNNs in terms of width, depth and the number of trainable parameters. Detailed comparisons can be found in Appendix E.

We consider multivariate smooth functions in $C_u^s([0,1]^d)$ (the unit ball of $C^s([0,1]^d)$) with smooth index $s \in \mathbb{N}$ as the target functions in our comparison. Note that smooth functions with smooth index s are exactly $(s+1)$-Hölder smooth functions by definition. For simplicity, we first show the case of $s = 2$. To achieve the same approximation error ε (say some constant), we need to set $K_Q = \Theta(d^2/\sqrt{\varepsilon})$ for the constructed PQCs from Theorem 4 and set $K_C = \Theta(2^{d/2}/\sqrt{\varepsilon})$ for the constructed near-optimal ReLU FNNs from Ref. [21]. Substituting the choices of K's in the sizes of PQCs and ReLU FNNs, we have

$$\frac{\text{Width of PQC} \times \text{Depth of PQC}}{\text{Width of FNN} \times \text{Depth of FNN}} = O\Big(\frac{d^3 K_Q^d}{2^{d+3} K_C^{d/2}}\Big) = O\Big(\frac{\varepsilon^{-d/4}}{2^{d^2 - d\log d}}\Big). \tag{11}$$

One can obtain a similar relation for the number of required parameters in PQCs and ReLU FNNs for approximating smooth functions and extend these results to any $2 \leq s < d$, which holds relevance in numerous real-world applications (e.g., the input dimension d is 784 for the MNIST dataset and is 150 528 for the ImageNet dataset [56], and empirically $s \leq 10$). Therefore, to achieve the same approximation error, the required quantum circuit size and number of parameters of PQCs is exponentially smaller than the required network size and number of parameters of ReLU FNNs proposed in Ref. [21].

Aiming to understand and continuously expand the range of problems that can be addressed using quantum machine learning, we have demonstrated the approximation capabilities of PQC models in supervised learning. We characterized the approximation error of PQCs in terms of the model size, delivering a deeper understanding of the expressive power of PQCs that is beyond the universal approximation properties. With these results, we can unlock the full potential of these models and drive advancements in quantum machine learning. Notably, by comparing our results with the near-optimal approximation error bound of classical ReLU neural networks, we demonstrate an improvement over the classical models on approximating high-dimensional functions that satisfy specific smoothness criteria, quantified by an improvement on the model size and the number of parameters.

Unlike many other investigations in the universal approximation properties of PQC models [26–33], our constructions of PQCs for approximating broad classes of continuous functions do not rely on any impractical assumptions. All the variables take the form of parameters within single-qubit rotation gates, avoiding any classical parameterized pre-processing or post-processing. Ultimately, our research provides valuable insights into the theoretical underpinnings of PQCs in quantum machine learning and paves the way for leveraging its capabilities in machine learning for both classical and quantum applications.

In this work, we introduce a novel nested PQC structure, which significantly improves the approximation capabilities. Future work could focus on exploring more powerful PQC constructions based on our proposed idea and understanding the capabilities and limitations of PQCs in more practical tasks even with real-world data. Developing efficient training strategies for PQCs, such as accelerated methods that achieve faster convergence rates, will also be interesting.

Acknowledgments and Disclosure of Funding

Part of this work was done when Z.Y. was visiting Wuhan University. Z.Y. thanks Patrick Rebentrost for helpful discussions. The authors thank the helpful comments from the anonymous reviewers. This work is supported by the National Key Research and Development Program of China (No. 2020YFA0714200), the National Nature Science Foundation of China (No. 62302346, No. 12125103, No. 12071362, No. 12371424, No. 12371441) and supported by the "Fundamental Research Funds for the Central Universities".

References

[1] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, September 2017. ISSN 1476-4687. doi: 10.1038/nature23474.

[2] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, November 2019. ISSN 2058-9565. doi: 10.1088/2058-9565/ab4eb5.

[3] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018. ISSN 2521-327X. doi: 10.22331/q-2018-08-06-79. URL https://doi.org/10.22331/q-2018-08-06-79.

[4] 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. ISSN 1476-4687. doi: 10.1038/nature23879.

[5] M. Cerezo, Kunal Sharma, Andrew Arrasmith, and Patrick J. Coles. Variational quantum state eigensolver. *npj Quantum Information*, 8(1):113, 2022. ISSN 2056-6387. doi: 10.1038/s41534-022-00611-6.

[6] Yudong Cao, Jonathan Romero, Jonathan P. Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P. D. Sawaya, Sukin Sim, Libor Veis, and Alán Aspuru-Guzik. Quantum chemistry in the age of quantum computing. *Chemical Reviews*, 119(19):10856–10915, 2019. ISSN 0009-2665. doi: 10.1021/acs.chemrev.8b00803.

[7] Xiaoxuan Pan, Zhide Lu, Weiting Wang, Ziyue Hua, Yifang Xu, Weikang Li, Weizhou Cai, Xuegang Li, Haiyan Wang, Yi-Pu Song, Chang-Ling Zou, Dong-Ling Deng, and Luyan Sun. Deep quantum neural networks on a superconducting processor. *Nature Communications*, 14 (1):4006, 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-39785-8.

[8] Wenhui Ren, Weikang Li, Shibo Xu, Ke Wang, Wenjie Jiang, Feitong Jin, Xuhao Zhu, Jiachen Chen, Zixuan Song, Pengfei Zhang, Hang Dong, Xu Zhang, Jinfeng Deng, Yu Gao, Chuanyu Zhang, Yaozu Wu, Bing Zhang, Qiujiang Guo, Hekang Li, Zhen Wang, Jacob Biamonte, Chao Song, Dong-Ling Deng, and H. Wang. Experimental quantum adversarial learning with programmable superconducting qubits. *Nature Computational Science*, 2(11):711–717, 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00351-9.

[9] He-Liang Huang, Yuxuan Du, Ming Gong, Youwei Zhao, Yulin Wu, Chaoyue Wang, Shaowei Li, Futian Liang, Jin Lin, Yu Xu, Rui Yang, Tongliang Liu, Min-Hsiu Hsieh, Hui Deng, Hao Rong, Cheng-Zhi Peng, Chao-Yang Lu, Yu-Ao Chen, Dacheng Tao, Xiaobo Zhu, and Jian-Wei Pan. Experimental quantum generative adversarial networks for image generation. *Phys. Rev. Appl.*, 16:024051, Aug 2021. doi: 10.1103/PhysRevApplied.16.024051. URL https://link.aps.org/doi/10.1103/PhysRevApplied.16.024051.

[10] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum Circuit Learning. *Physical Review A*, 98(3):032309, September 2018. ISSN 2469-9926, 2469-9934. doi: 10.1103/PhysRevA.98.032309.

[11] Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, February 2020. doi: 10.22331/q-2020-02-06-226.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. ISSN 1476-4687. doi: 10.1038/nature14539.

[13] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989. URL https://link.springer.com/article/10.1007/BF02551274.

[14] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, January 1989. ISSN 08936080. doi: 10.1016/0893-6080(89)90020-8. URL https://linkinghub.elsevier.com/retrieve/pii/0893608089900208.

[15] A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, May 1993. ISSN 1557-9654. doi: 10.1109/18.256500.

[16] Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017. ISSN 0893-6080. doi: 10.1016/j.neunet.2017.07.002.

[17] Dmitry Yarotsky. Optimal approximation of continuous functions by very deep ReLU networks. In *Proceedings of the 31st Conference On Learning Theory*, pages 639–649. PMLR, July 2018. URL https://proceedings.mlr.press/v75/yarotsky18a.html.

[18] Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks*, 108:296–330, December 2018. ISSN 0893-6080. doi: 10.1016/j.neunet.2018.08.019.

[19] Dmitry Yarotsky and Anton Zhevnerchuk. The phase diagram of approximation rates for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 13005–13015. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/979a3f14bae523dc5101c52120c535e9-Abstract.html.

[20] Zuowei Shen. Deep Network Approximation Characterized by Number of Neurons. *Communications in Computational Physics*, 28(5):1768–1811, June 2020. ISSN 1815-2406, 1991-7120. doi: 10.4208/cicp.OA-2020-0149.

[21] Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep Network Approximation for Smooth Functions. *SIAM Journal on Mathematical Analysis*, 53(5):5465–5506, January 2021. ISSN 0036-1410. doi: 10.1137/20M134695X. URL https://epubs.siam.org/doi/10.1137/20M134695X.

[22] Zuowei Shen, Haizhao Yang, and Shijun Zhang. Optimal approximation rate of ReLU networks in terms of width and depth. *Journal de Mathématiques Pures et Appliquées*, 157:101–135, January 2022. ISSN 0021-7824. doi: 10.1016/j.matpur.2021.07.009. URL https://www.sciencedirect.com/science/article/pii/S0021782421001124.

[23] E Weinan, Chao Ma, and Lei Wu. The barron space and the flow-induced function spaces for neural network models. *Constructive Approximation*, 55(1):369–406, 2022.

[24] Yuling Jiao, Yanming Lai, Xiliang Lu, Fengru Wang, Jerry Zhijian Yang, and Yuanyuan Yang. Deep neural networks with ReLU-sine-exponential activations break curse of dimensionality in approximation on hölder class. *SIAM Journal on Mathematical Analysis*, 55(4):3635–3649, 2023. doi: 10.1137/21M144431X. URL https://doi.org/10.1137/21M144431X.

[25] Yuling Jiao, Guohao Shen, Yuanyuan Lin, and Jian Huang. Deep nonparametric regression on approximate manifolds: Nonasymptotic error bounds with polynomial prefactors. *The Annals of Statistics*, 51(2):691–716, April 2023. ISSN 0090-5364, 2168-8966. doi: 10.1214/23-AOS2266. URL https://projecteuclid.org/journals/annals-of-statistics/volume-51/issue-2/Deep-nonparametric-regression-on-approximate-manifolds--Nonasymptotic-error-bounds/10.1214/23-AOS2266.full.

[26] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, March 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-0980-2. URL https://www.nature.com/articles/s41586-019-0980-2.

[27] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. Expressive power of parametrized quantum circuits. *Physical Review Research*, 2(3):033125, July 2020. doi: 10.1103/PhysRevResearch.2.033125.

[28] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, 17(9):1013–1017, 2021. ISSN 1745-2481. doi: 10.1038/s41567-021-01287-z.

[29] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Information-theoretic bounds on quantum advantage in machine learning. *Phys. Rev. Lett.*, 126:190505, May 2021. doi: 10.1103/PhysRevLett.126.190505. URL https://link.aps.org/doi/10.1103/PhysRevLett.126.190505.

[30] Sofiene Jerbi, Casper Gyurik, Simon Marshall, Hans Briegel, and Vedran Dunjko. Parametrized Quantum Policies for Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 28362–28375. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/eec96a7f788e88184c0e713456026f3f-Abstract.html.

[31] 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, May 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-22539-9. URL https://www.nature.com/articles/s41467-021-22539-9.

[32] Sofiene Jerbi, Lukas J. Fiderer, Hendrik Poulsen Nautrup, Jonas M. Kübler, Hans J. Briegel, and Vedran Dunjko. Quantum machine learning beyond kernel methods. *Nature Communications*, 14(1):517, January 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-36159-y. URL https://www.nature.com/articles/s41467-023-36159-y.

[33] Jonas Jäger and Roman V. Krems. Universal expressiveness of variational quantum classifiers and quantum kernels for support vector machines. *Nature Communications*, 14(1):576, February 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-36144-5. URL https://www.nature.com/articles/s41467-023-36144-5.

[34] 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, March 2021. doi: 10.1103/PhysRevA.103.032430.

[35] Francisco Javier Gil Vidal and Dirk Oliver Theis. Input Redundancy for Parameterized Quantum Circuits. *Frontiers in Physics*, 8, 2020. ISSN 2296-424X. URL https://www.frontiersin.org/articles/10.3389/fphy.2020.00297.

[36] Adrián Pérez-Salinas, David López-Núñez, Artur García-Sáez, P. Forn-Díaz, and José I. Latorre. One qubit as a universal approximant. *Physical Review A*, 104(1):012405, July 2021. doi: 10.1103/PhysRevA.104.012405.

[37] Zhan Yu, Hongshun Yao, Mujin Li, and Xin Wang. Power and limitations of single-qubit native quantum neural networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27810–27823. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b250de41980b58d34d6aadc3f4aedd4c-Paper-Conference.pdf.

[38] Alberto Manzano, David Dechant, Jordi Tura, and Vedran Dunjko. Parametrized Quantum Circuits and their approximation capacities in the context of quantum machine learning, July 2023.

[39] Takahiro Goto, Quoc Hoan Tran, and Kohei Nakajima. Universal Approximation Property of Quantum Machine Learning Models in Quantum-Enhanced Feature Spaces. *Physical Review Letters*, 127(9):090506, August 2021. doi: 10.1103/PhysRevLett.127.090506.

[40] Lukas Gonon and Antoine Jacquier. Universal Approximation Theorem and error bounds for quantum neural networks and quantum reservoirs, July 2023.

[41] Jun Qi, Chao-Han Huck Yang, Pin-Yu Chen, and Min-Hsiu Hsieh. Theoretical error performance analysis for variational quantum circuit based functional regression. *npj Quantum Information*, 9(1):4, 2023. ISSN 2056-6387. doi: 10.1038/s41534-022-00672-7.

[42] Haimeng Zhao, Laura Lewis, Ishaan Kannan, Yihui Quek, Hsin-Yuan Huang, and Matthias C. Caro. Learning quantum states and unitaries of bounded gate complexity, 2023.

[43] Philipp Grohs and Gitta Kutyniok. *Mathematical aspects of deep learning*. Cambridge University Press, 2022.

[44] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[45] Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang. Methodology of Resonant Equiangular Composite Quantum Gates. *Physical Review X*, 6(4):041067, December 2016. doi: 10.1103/PhysRevX.6.041067.

[46] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian Simulation by Quantum Signal Processing. *Physical Review Letters*, 118(1):010501, January 2017. doi: 10.1103/PhysRevLett.118.010501.

[47] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, June 2019. doi: 10.1145/3313276.3316366.

[48] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Inf. Comput.*, 12(11-12):901–924, 2012. doi: 10.26421/QIC12.11-12-1.

[49] Adenilton J. da Silva and Daniel K. Park. Linear-depth quantum circuits for multiqubit controlled gates. *Physical Review A*, 106(4):042602, October 2022. doi: 10.1103/PhysRevA.106.042602.

[50] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum Amplitude Amplification and Estimation. *Contemporary Mathematics*, 305:53–74, 2002. doi: 10.1090/conm/305/05215.

[51] Kenneth R. Davidson and Allan P. Donsig. *Real analysis with real applications*. Prentice Hall, 2002. URL https://cir.nii.ac.jp/crid/1130000794786166144.

[52] Clemens Heitzinger. *Simulation and Inverse Modeling of Semiconductor Manufacturing Processes*. Thesis, Technische Universität Wien, 2002.

[53] Mama Foupouagnigni and Merlin Mouafo Wouodjié. On Multivariate Bernstein Polynomials. *Mathematics*, 8(9):1397, September 2020. ISSN 2227-7390. doi: 10.3390/math8091397.

[54] Guang Hao Low. *Quantum Signal Processing by Single-Qubit Dynamics*. Thesis, Massachusetts Institute of Technology, 2017. URL https://dspace.mit.edu/handle/1721.1/115025.

[55] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[56] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848. URL https://ieeexplore.ieee.org/abstract/document/5206848.

[57] Youle Wang, Lei Zhang, Zhan Yu, and Xin Wang. Quantum Phase Processing and its Applications in Estimating Phase and Entropies, July 2023.

[58] V. N. Vapnik and A. Ya. Chervonenkis. Necessary and Sufficient Conditions for the Uniform Convergence of Means to their Expectations. *Theory of Probability & Its Applications*, 26(3): 532–553, January 1982. ISSN 0040-585X. doi: 10.1137/1126059.

[59] V. M. Tikhomirov. ϵ-Entropy and ϵ-Capacity of Sets In Functional Spaces. In A. N. Shiryayev, editor, *Selected Works of A. N. Kolmogorov: Volume III: Information Theory and the Theory of Algorithms*, Mathematics and Its Applications, pages 86–170. Springer Netherlands, Dordrecht, 1993. ISBN 978-94-017-2973-4. doi: 10.1007/978-94-017-2973-4_7.

[60] Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *Journal of Machine Learning Research*, 3(3):463, April 2003. ISSN 15324435. URL https://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=10257714&lang=zh-cn&site=ehost-live.

[61] Yuxuan Du, Zhuozhuo Tu, Xiao Yuan, and Dacheng Tao. Efficient Measure for the Expressivity of Variational Quantum Algorithms. *Physical Review Letters*, 128(8):080506, February 2022. doi: 10.1103/PhysRevLett.128.080506.

[62] Kaifeng Bu, Dax Enshan Koh, Lu Li, Qingxian Luo, and Yaobo Zhang. Statistical complexity of quantum circuits. *Physical Review A*, 105(6):062431, June 2022. doi: 10.1103/PhysRevA.105.062431.

[63] Matthias C. Caro and Ishaun Datta. Pseudo-dimension of quantum circuits. *Quantum Machine Intelligence*, 2(2):14, November 2020. ISSN 2524-4914. doi: 10.1007/s42484-020-00027-5.

[64] Chih-Chieh Chen, Masaru Sogabe, Kodai Shiba, Katsuyoshi Sakamoto, and Tomah Sogabe. General Vapnik–Chervonenkis dimension bounds for quantum circuit learning. *Journal of Physics: Complexity*, 3(4):045007, November 2022. ISSN 2632-072X. doi: 10.1088/2632-072X/ac9f9b.

[65] Amira Abbas, David Sutter, Christa Zoufal, Aurelien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409, June 2021. ISSN 2662-8457. doi: 10.1038/s43588-021-00084-1.

[66] Ronald A. DeVore, Ralph Howard, and Charles Micchelli. Optimal nonlinear approximation. *manuscripta mathematica*, 63(4):469–478, December 1989. ISSN 1432-1785. doi: 10.1007/BF01171759.

[67] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991. URL https://www.sciencedirect.com/science/article/abs/pii/089360809190009T.

[68] Weinan E, Chao Ma, and Lei Wu. The Barron Space and the Flow-Induced Function Spaces for Neural Network Models. *Constructive Approximation*, 55(1):369–406, February 2022. ISSN 1432-0940. doi: 10.1007/s00365-021-09549-y. URL https://doi.org/10.1007/s00365-021-09549-y.

[69] M. H. Stone. The Generalized Weierstrass Approximation Theorem. *Mathematics Magazine*, 21(4):167–184, 1948. ISSN 0025-570X. doi: 10.2307/3029750.

[70] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. doi: 10.1109/CVPR.2016.90.

[71] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

[72] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html.

[73] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[74] Shijun Zhang, Jianfeng Lu, and Hongkai Zhao. Deep Network Approximation: Beyond ReLU to Diverse Activation Functions, September 2023. URL http://arxiv.org/abs/2307.06555.

[75] Ingo Gühring, Gitta Kutyniok, and Philipp Petersen. Error bounds for approximations with deep ReLU neural networks in ws,p norms. *Analysis and Applications*, 18(05):803–859, 2020. doi: 10.1142/S0219530519410021. URL https://doi.org/10.1142/S0219530519410021.

[76] Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4):1875 – 1897, 2020. doi: 10.1214/19-AOS1875. URL https://doi.org/10.1214/19-AOS1875.

Supplementary Material

Contents

1 Introduction **1**

2 Preliminaries **3**

3 Expressivity of PQCs for multivariate continuous functions **4**

3.1 Explicit construction of PQCs for multivariate polynomials 4

3.2 PQC approximation for continuous functions 5

3.3 PQC approximation for Hölder smooth functions 6

4 Numerical experiments **8**

5 Discussion **8**

A Preliminaries **17**

A.1 Notation 17

A.2 Data re-uploading PQCs 17

A.2.1 Implementing real polynomials 17

A.2.2 Implementing trigonometric polynomials 18

A.3 Related work in PQC approximation 19

B Implementing multivariate polynomials using PQCs **20**

B.1 Implementing multivariate real polynomials 20

B.2 Implementing multivariate trigonometric polynomials 22

C Approximating continuous functions via PQCs **23**

C.1 Established results of Bernstein polynomials approximation 23

C.2 Implement Bernstein polynomials via PQCs 25

C.3 PQC approximating continuous functions 26

D Approximating smooth functions via nested PQCs **26**

D.1 Localization via PQC 27

D.2 Implementing the Taylor coefficients by PQC 28

D.3 Implementing multivariate Taylor series by PQC 28

D.4 Comparison of “global” and “local” approaches in this work 31

E Comparison with related works in classical machine learning **31**

A Preliminaries

In this section, we will first present some essential mathematical foundations for deriving the main results of this work. Moreover, to contextualize our work within the existing literature, we comprehensively review relevant studies in Appendix A.3.

A.1 Notation

We unify the notations throughout the whole work. The univariate polynomial ring over a field $\mathbb{F}$ is symbolized as $\mathbb{F}[x]$, with the variable x representing the input. The ring of Laurent polynomial $\mathbb{F}[x, x^{-1}]$ is an extension of the polynomial ring obtained by adding inverses of x. The collection of natural numbers is represented by the symbol $\mathbb{N} := \{1, 2, 3, \dots\}$, while the set of non-negative integers is denoted as $\mathbb{N}_0 := \{0\} \cup \mathbb{N}$. The 1-norm of a vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_d)$ is denoted by $\|\boldsymbol{\alpha}\|_1 := |\alpha_1| + |\alpha_2| + \cdots + |\alpha_d|$.

A.2 Data re-uploading PQCs

In this section, we review the concept of data re-uploading PQC and define the PQC we use in this paper. The data re-uploading PQC is a quantum circuit that consists of interleaved data encoding circuit blocks and trainable circuit blocks [35, 11]. More precisely, let $\boldsymbol{x}$ be the input data vector and $\boldsymbol{\theta} = (\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_L})$ be a set of trainable parameters. $S(\boldsymbol{x})$ is a quantum circuit that encode $\boldsymbol{x}$ and $V(\boldsymbol{\theta}_j)$ is a trainable quantum circuit with trainable parameter vector $\boldsymbol{\theta}_j$. An L-layer data re-uploading PQC can be then expressed as

$$U_{\boldsymbol{\theta}}(\boldsymbol{x}) = V(\boldsymbol{\theta_0}) \prod_{j=1}^{L} S(\boldsymbol{x}) V(\boldsymbol{\theta_j}), \tag{A.1}$$

Applying $U_{\boldsymbol{\theta}}(\boldsymbol{x})$ to a quantum state and measuring the output states provides a way to express functions on $\boldsymbol{x}$. The *expressivity* of the data re-uploading PQC model can be characterized by the classes of functions that it can implement. It is common to build data encoding circuits and trainable circuits using the most prevalent Pauli rotation operators,

$$R_X(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, \quad R_Y(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, \quad R_Z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}. \tag{A.2}$$

Different data encoding schemes lead to different types of data re-uploading PQCs.

In some cases, trainable parameters are also included both during the initial data encoding phase and the final processing of measurement outcomes. These PQCs are considered to have *hybrid* structures. For instance, in the models proposed by Refs. [35, 36, 40], each input data is multiplied by a specific trainable parameter and subsequently subjected to R_Z gates during the data encoding stage. In a similar vein, Refs. [39, 40] incorporate trainable weights into each measurement outcome generated by the constructed PQCs, aggregating these weighted outcomes to produce the final result. Such a structure makes it hard to judge whether the expressive power comes from the classical or quantum part.

A.2.1 Implementing real polynomials

We first introduce the data re-uploading PQC for implementing real univariate polynomials. We utilize the so-called *Pauli X basis encoding* [10]: The data encoding unitary is a single-qubit rotation defined as

$$S(x) := e^{i \arccos(x) X} = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix}, \tag{A.3}$$

where $x \in [-1, 1]$ is the input data. Then interlaying the data encoding unitary $S(x)$ with some parameterized Pauli Z rotations $R_Z(\theta)$ gives the circuit of data re-uploading PQC for one variable as

$$U_{\boldsymbol{\theta}}(x) := R_Z(\theta_0) \prod_{j=1}^{L} S(x) R_Z(\theta_j), \tag{A.4}$$

where $\boldsymbol{\theta} = (\theta_0, \ldots, \theta_L) \in \mathbb{R}^{L+1}$ is a set of trainable parameters. The PQC in Eq. (A.4) can be used to implement polynomial transformations on input x, as shown in the following lemma.

Lemma S1 ([47]). *There exists $\boldsymbol{\theta} \in \mathbb{R}^{L+1}$ such that*

$$U_{\boldsymbol{\theta}}(x) = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix} \tag{A.5}$$

if and only if polynomials $P, Q \in \mathbb{C}[x]$ satisfy

1. $\deg(P) \leq L$ *and* $\deg(Q) \leq L-1$,
2. P *has parity* $L \bmod 2$ *and* Q *has parity* $(L-1) \bmod 2$[5],
3. $\forall x \in [-1,1]$, $|P(x)|^2 + (1-x^2)|Q(x)|^2 = 1$.

As shown in the above lemma, one could implement a polynomial transformation $\mathrm{Poly}(x)$ such that $\mathrm{Poly}(x) = \langle 0|U_{\boldsymbol{\theta}}(x)|0\rangle = P(x)$. Notice that the achievable polynomial $\mathrm{Poly}(x)$ implemented in this way is limited to $P(x)$ for which there exists a polynomial $Q(x)$ satisfying the conditions of Lemma S1. As the target polynomial is often real in practice, we could overcome such a limitation by defining $\mathrm{Poly}(x) = \langle +|U_{\boldsymbol{\theta}}(x)|+\rangle = \Re(P(x)) + i\Re(Q(x))\sqrt{1-x^2}$. Then we can achieve any real polynomials with parity $L \bmod 2$ such that $\deg(\mathrm{Poly}(x)) \leq L$, and $|\mathrm{Poly}(x)| \leq 1$ for all $x \in [-1,1]$.

Corollary S2 ([47]). *There exists $\boldsymbol{\theta} \in \mathbb{R}^{L+1}$ such that*

$$p(x) = \langle +|U_{\boldsymbol{\theta}}(x)|+\rangle \tag{A.6}$$

if and only if the real polynomial $p(x) \in \mathbb{R}[x]$ satisfies

1. $\deg(p(x)) \leq L$,
2. $p(x)$ *has parity* $L \bmod 2$ [6],
3. $\forall x \in [-1,1]$, $|p(x)| \leq 1$.

Remark S1. *The results of PQC with Pauli X basis encoding presented here have been established in the technique of quantum signal processing [45–47], which uses interleaving signal operators and signal processing operators to transform the input signal. The QSP circuit could be identified as a PQC in the context of quantum machine learning.*

A.2.2 Implementing trigonometric polynomials

Other than the real polynomials, there are also types of single-qubit PQC with Pauli Z basis encoding that could implement complex trigonometric polynomials [37]. The data encoding unitary is a single-qubit rotation in the Pauli Z basis

$$S(x) := R_Z(x) = \begin{pmatrix} e^{ix/2} & 0 \\ 0 & e^{-ix/2} \end{pmatrix}, \tag{A.7}$$

where $x \in \mathbb{R}$ is the data. By interleaving the data encoding unitary $S(x)$ with trainable gates $R_Y(\theta)R_Z(\phi)$, the PQC is defined as

$$U_{\boldsymbol{\theta},\boldsymbol{\phi}}(x) := R_Z(\omega)R_Y(\theta_0)R_Z(\phi_0)\prod_{j=1}^{L} S(x)R_Y(\theta_j)R_Z(\phi_j), \tag{A.8}$$

where $\boldsymbol{\theta} = (\theta_0, \ldots, \theta_L) \in \mathbb{R}^{L+1}$, $\boldsymbol{\phi} = (\phi_0, \ldots, \phi_L) \in \mathbb{R}^{L+1}$ and $\omega \in \mathbb{R}$. The following lemma characterizes the correspondence between PQC with σ_z basis encoding and complex trigonometric polynomials.

[5] For a polynomial $P \in \mathbb{C}[x]$, P has parity 0 if all coefficients corresponding to odd powers of x are 0, and similarly P has parity 1 if all coefficients corresponding to even powers of x are 0.

[6] A polynomial $p(x)$ has parity 0 if all coefficients corresponding to odd powers of x are 0, and similarly $p(x)$ has parity 1 if all coefficients corresponding to even powers of x are 0.

Lemma S3 ([37])**.** *There exist $\boldsymbol{\theta}, \boldsymbol{\phi} \in \mathbb{R}^{L+1}$ and $\omega \in \mathbb{R}$ such that*

$$U_{\boldsymbol{\theta},\boldsymbol{\phi}}(x) = \begin{pmatrix} P(x) & -Q(x) \\ Q^*(x) & P^*(x) \end{pmatrix} \tag{A.9}$$

if and only if Laurent polynomials $P, Q \in \mathbb{C}[e^{ix/2}, e^{-ix/2}]$ satisfy

1. $\deg(P) \leq L$ *and* $\deg(Q) \leq L$,
2. P *and* Q *have parity* $L \bmod 2$,
3. $\forall x \in \mathbb{R}$, $|P(x)|^2 + |Q(x)|^2 = 1$.

Note that Laurent polynomials in $\mathbb{C}[e^{ix/2}, e^{-ix/2}]$ with parity 0 are Laurent polynomials in $\mathbb{C}[e^{ix}, e^{-ix}]$ without parity constraints, which implies that the trigonometric QSP could implement complex trigonometric polynomials.

Corollary S4 ([37, 57])**.** *There exist $\boldsymbol{\theta}, \boldsymbol{\phi} \in \mathbb{R}^{2L+1}$ and $\omega \in \mathbb{R}$ such that*

$$t(x) = \langle 0| U_{\boldsymbol{\theta},\boldsymbol{\phi}}(x) |0\rangle \tag{A.10}$$

if and only if the complex-valued trigonometric polynomial $t(x) = \sum_{j=-L}^{L} c_j e^{ijx}$ satisfies $|t(x)| \leq 1$ for all $x \in \mathbb{R}$.

A.3 Related work in PQC approximation

In this subsection, we review prior literature related to the approximation capabilities of PQCs, which characterizes how the architectural properties of a PQC affect the resulting functions it can fit, and its ensuing performance. After a systematic comparison, we conclude that our results provide precise error bounds for continuous function approximation and make no assumptions about the constructed PQCs. More importantly, all the variables in our proposal take the form of parameters within rotation gates and remain distinct from the data encoding gates to avoid any classical computational influence, thus preserving the inherent quantum property of our approach.

In theoretical machine learning, statistical complexity is a notion that measures the inherent richness characterizing a given hypothesis space. There are various statistical complexity measures, including the Vapnik-Chervonenkis (VC) dimension [58], the metric entropy [59], the Gaussian complexity [60], and the Rademacher complexity [60], etc. To gauge the statistical complexity of PQCs, Du et al. [61] have explored the covering entropy of PQCs in terms of the number of quantum gates and the measurement observable. Bu et al. [62] have investigated the dependence of the Rademacher complexity of PQCs on the resources, width, depth, and the property of input and output registers. The assessment of PQCs has extended to encompass an array of statistical complexity measures, including the Pseudo-Dimension, as delineated in Caro and Datta [63], and the VC dimension, as expounded upon in Chen et al. [64]. Furthermore, the evaluation of PQC expressivity has extended its purview to metrics rooted in information theory. Abbas et al. [65] have evaluated PQC expressivity through the prism of the effective dimension, a data-dependent metric contingent upon the Fisher information. In a parallel endeavor, Du et al. [27] have concentrated their attention on generative tasks, employing entanglement entropy as a metric for quantifying PQC expressivity. It is important to underscore that, while statistical complexity metrics and information-inspired metrics provide invaluable insights into the 'volume' of hypothesis spaces, they do not precisely delineate the functions amenable to representation by these models.

To further explore the intricacies of PQCs and their expressivity, an alternative avenue of research has emerged, as highlighted by recent studies [34, 35, 37, 36, 38]. They rewrote the PQC output, i.e., the inner product between an input quantum state and a variational observable, in the form of partial Fourier series. This innovative perspective introduces a more nuanced toolbox for assessing PQC expressivity, offering fresh insights within the quantum machine learning domain, notably with respect to the universal approximation property (UAP). However, it is imperative to underscore that many investigations employing Fourier expansion have been predicated upon certain impractical assumptions. These assumptions encompass the demand for arbitrary parameterized global unitaries and observables, thus posing significant challenges to the practical implementation of the constructed quantum circuits. The existence proof of universal approximation also does not explicitly give approximation error bounds of PQCs.

A very general approach to expressiveness in the context of approximation is the method of nonlinear widths by DeVore et al. [66] that concerns the approximation of a family of functions under the assumption of a continuous dependence of the model on the approximated function. Pérez-Salinas et al. [36] have proved that single-qubit data re-uploading PQCs are universal function approximators, inheriting the famous universal approximation theorem for neural networks [13, 67]. In a quantum-enhanced context, Goto et al. [39] have constructed PQCs to approximate any continuous function guided by the Stone-Weierstrass theorem. Qi et al. [41] have studied the approximation error of PQCs enhanced by tensor-train networks. Their investigation focused on smooth functions, considering factors such as the number of qubits and quantum measurement counts. Furthermore, Gonon and Jacquier [40] have defined a specific hypothesis space consisting of non-oscillating functions, drawing inspiration from Barron [15] and devised PQCs for approximating such functions without encountering the curse of dimensionality (CoD). Notably, the mitigation of CoD arises from their specific hypothesis space definition and is also observed within the domain of classical neural network [68]. It is essential to acknowledge that these works unveil a hybrid nature, blurring the boundaries between classical and quantum domains in circuit construction. The hybrid structure manifests in the data encoding phase and becomes evident in the weighted summation of outputs from foundational quantum circuits. Consequently, whether the powerful expressivity comes from the classical part or the quantum part of hybrid models is unclear.

In our present work, we make no assumptions in the construction of the PQCs. In our PQC model, all variables take the form of parameters within rotation gates. Besides, these trainable parameters remain distinct from the data encoding gates to avoid any classical computational influence. These properties ensure that our constructed PQCs retain practicality and remain firmly rooted within the quantum domain.

B Implementing multivariate polynomials using PQCs

B.1 Implementing multivariate real polynomials

A multivariate polynomial with d variables and degree $s \in \mathbb{N}$ is defined as

$$p(\boldsymbol{x}) := \sum_{\|\boldsymbol{\alpha}\|_1 \le s} c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}, \tag{B.11}$$

where $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$, $c_{\boldsymbol{\alpha}} \in \mathbb{R}$ and $\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d}$. To implement the multivariate polynomial $p(\boldsymbol{x})$, we first build a PQC to express a monomial $c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}} = c_{\boldsymbol{\alpha}} x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d}$, where $|c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}| \le 1$ for $\boldsymbol{x} \in [0,1]^d$ and $\|\boldsymbol{\alpha}\|_1 \le s$. We apply the single-qubit PQC with Pauli X basis encoding defined in Eq. (A.4) on each x_j for $1 \le j \le d$, respectively.

Lemma S5. *Given a monomial $c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}} = c_{\boldsymbol{\alpha}} x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d}$ such that $|c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}| \le 1$ for all $\boldsymbol{x} \in [0,1]^d$ and $\|\boldsymbol{\alpha}\|_1 \le s$ for $s \in \mathbb{N}$, there exists a PQC $U^{\boldsymbol{\alpha}}(\boldsymbol{x})$ such that*

$$\langle + |^{\otimes d} U^{\boldsymbol{\alpha}}(\boldsymbol{x}) | + \rangle^{\otimes d} = c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}. \tag{B.12}$$

The width of the PQC is at most d, the depth is at most $2s+1$, and the number of parameters is at most $s+d$.

Proof. By Corollary S2, there exist d single-qubit PQCs $U_{\boldsymbol{\theta}_1}^{\alpha_1}(x_1), U_{\boldsymbol{\theta}_2}^{\alpha_2}(x_2), \ldots, U_{\boldsymbol{\theta}_d}^{\alpha_d}(x_d)$ such that

$$\begin{aligned} \langle + | U_{\boldsymbol{\theta}_1}^{\alpha_1}(x_1) | + \rangle &= c_{\boldsymbol{\alpha}} x_1^{\alpha_1}, \\ \langle + | U_{\boldsymbol{\theta}_2}^{\alpha_2}(x_2) | + \rangle &= x_2^{\alpha_2}, \\ &\cdots \\ \langle + | U_{\boldsymbol{\theta}_d}^{\alpha_d}(x_d) | + \rangle &= x_d^{\alpha_d}, \end{aligned}$$

where the number of layers of each PQC is $L_j = \alpha_j$ for $1 \le j \le d$. We then define a d-qubit PQC as

$$U^{\boldsymbol{\alpha}}(\boldsymbol{x}) = \bigotimes_{j=1}^{d} U_{\boldsymbol{\theta}_j}^{\alpha_j}(x_j), \tag{B.13}$$

which gives

$$\langle +|^{\otimes d} U^{\boldsymbol{\alpha}}(\boldsymbol{x})|+\rangle^{\otimes d} = \prod_{j=1}^{d} \langle +|U_{\boldsymbol{\theta}_j}^{\alpha_j}(x_j)|+\rangle = c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}. \tag{B.14}$$

Since $\|\boldsymbol{\alpha}\|_1 = \sum_{j=1}^{d} \alpha_j \le s$, we can conclude that the depth of $U^{\boldsymbol{\alpha}}(\boldsymbol{x})$ is at most $2s+1$ and the number of parameters in $U^{\boldsymbol{\alpha}}(\boldsymbol{x})$ is at most $s+d$. □

The next step is to combine monomials together to implement the multivariate polynomial. Specifically, we would like to implement the following (unnormalized) operator

$$U_p(\boldsymbol{x}) \coloneqq \sum_{\|\boldsymbol{\alpha}\|_1 \le s} U^{\boldsymbol{\alpha}}(\boldsymbol{x}) \tag{B.15}$$

so that we can implement an (unnormalized) polynomial as

$$\langle +|^{\otimes d} U_p(\boldsymbol{x})|+\rangle^{\otimes d} = \sum_{\|\boldsymbol{\alpha}\|_1 \le s} \langle +|^{\otimes d} U^{\boldsymbol{\alpha}}(\boldsymbol{x})|+\rangle^{\otimes d} = \sum_{\|\boldsymbol{\alpha}\|_1 \le s} c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}} = p(\boldsymbol{x}). \tag{B.16}$$

We denote T the number of terms in the summation and observe that it can be bounded as

$$T = \sum_{\|\boldsymbol{\alpha}\|_1 \le s} 1 = \sum_{j=0}^{s} \sum_{\|\boldsymbol{\alpha}\|_1 = j} 1 \le \sum_{j=0}^{s} d^s \le (s+1) d^s. \tag{B.17}$$

For convenience, we rewrite the normalized target operator with $\boldsymbol{\alpha}$ being an indexed variable as

$$U_p(\boldsymbol{x}) = \sum_{j=1}^{T} \frac{1}{T} U^{\boldsymbol{\alpha}^{(j)}}(\boldsymbol{x}). \tag{B.18}$$

However, the addition operation in quantum computing is non-trivial as the sum of unitary operators is not necessarily unitary. To sum the monomials together, we utilize the technique of *linear combination of unitaries* (LCU) [48] to implement the operator $U_p(\boldsymbol{x})$ in Eq. (B.18) on a quantum computer. We first construct a unitary operator F such that

$$F|0\rangle = \frac{1}{\sqrt{T}} \sum_{j=1}^{T} |j\rangle. \tag{B.19}$$

The unitary F could be simply implemented by Hadamard gates. Next, we construct a controlled unitary

$$U_c(\boldsymbol{x}) = \sum_{j=1}^{T} |j\rangle\langle j| \otimes U^{\boldsymbol{\alpha}^{(j)}}(\boldsymbol{x}). \tag{B.20}$$

Note that each $|j\rangle\langle j| \otimes U^{\boldsymbol{\alpha}^{(j)}}(\boldsymbol{x})$ could be constructed using $(\log T)$-qubit controlled Pauli rotation gates, as $U^{\boldsymbol{\alpha}^{(j)}}(\boldsymbol{x})$ consisting of single-qubit Pauli rotation gates. The $(\log T)$-qubit controlled gates could be further decomposed into quantum circuits of CNOT gates and single-qubit rotation gates in $O(\log T)$ circuit depth without using any ancilla qubit. We refer to the detailed implementation of these multi-controlled gates to da Silva and Park [49]. Then the unitary $W_{lcu} = (F^\dagger \otimes I) U_c (F \otimes I)$ satisfies that

$$W_{lcu}|0\rangle|+\rangle^{\otimes d} = |0\rangle U_p(\boldsymbol{x})|+\rangle^{\otimes d} + |\bot\rangle, \tag{B.21}$$

where $(\langle 0| \otimes I)|\bot\rangle = 0$. Notice that

$$\langle 0| \langle +|^{\otimes d} W_{lcu} |0\rangle |+\rangle^{\otimes d} = \langle +|^{\otimes d} U_p(\boldsymbol{x}) |+\rangle^{\otimes d} = p(\boldsymbol{x}). \tag{B.22}$$

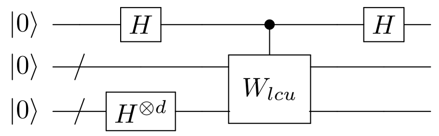
To obtain the polynomial $p(\boldsymbol{x})$, we could estimate $\langle 0| \langle +|^{\otimes d} W_{lcu} |0\rangle |+\rangle^{\otimes d}$ using the Hadamard test.

Theorem 1. *For any multivariate polynomial $p(\boldsymbol{x})$ with d variables and degree s such that $|p(\boldsymbol{x})| \le 1$ for $\boldsymbol{x} \in [0,1]^d$, there exists a PQC $W_p(\boldsymbol{x})$ such that*

$$f_{W_p}(\boldsymbol{x}) \coloneqq \langle 0| W_p^\dagger(\boldsymbol{x}) Z^{(0)} W_p(\boldsymbol{x}) |0\rangle = p(\boldsymbol{x}) \tag{B.23}$$

where $Z^{(0)}$ is the Pauli Z observable on the first qubit. The width of the PQC is $O(d + \log s + s \log d)$, the depth is $O(s^2 d^s (\log s + s \log d))$, and the number of parameters is $O(s d^s (s+d))$.

Proof. We apply the Hadamard test on W_{lcu}, giving the quantum circuit $W_p(\boldsymbol{x})$ as follows.



Measuring the first qubit of $W_p(\boldsymbol{x})$, we have

$$f_{W_p}(\boldsymbol{x}) \coloneqq \langle 0|\, W_p^\dagger(\boldsymbol{x}) Z^{(0)} W_p(\boldsymbol{x})\, |0\rangle = \langle 0|\, \langle +|^{\otimes d}\, W_{lcu}\, |0\rangle\, |+\rangle^{\otimes d} = p(\boldsymbol{x}). \tag{B.24}$$

The controlled unitary used in LCU,

$$U_c(\boldsymbol{x}) = \sum_{j=1}^{T} |j\rangle\langle j| \otimes U^{\boldsymbol{\alpha}^{(j)}}(\boldsymbol{x}), \tag{B.25}$$

could be implemented by at most $O(Ts)$ $(\log T)$-qubit controlled gates. A $(\log T)$-qubit controlled gate could be implemented by a quantum circuit consisting of CNOT gates and single-qubit gates with depth $O(\log T)$ [49]. Thus $U_c(\boldsymbol{x})$ could be implemented by a quantum circuit with depth $O(sT \log T)$ and width $O(d + \log T)$. Then the depth and width of $W_{lcu} = (F^\dagger \otimes I) U_c (F \otimes I)$ are in the same order of $U_c(\boldsymbol{x})$ since F is simply tensor of Hadamard gates. Therefore the entire depth of the circuit W_p is $O\big(sT \log T + d\big)$, and the width of W_p is $O(d + \log T)$. As $T \le (s+1)d^s$. Note that the number of parameters in the PQC equals the number of parameters in $U_c(\boldsymbol{x})$, which is $O(T(s+d))$. □

Note that measuring the first qubit of $W_p(\boldsymbol{x})$ for $O(\frac{1}{\varepsilon^2})$ times is needed to estimate the value of $p(\boldsymbol{x})$ up to an additive error ε. We could further use the amplitude estimation algorithm [50] to reduce the overhead while increasing the circuit depth by $O(\frac{1}{\varepsilon})$.

B.2 Implementing multivariate trigonometric polynomials

We extend the PQCs with R_Z encoding to implement multivariate trigonometric polynomials. A multivariate trigonometric polynomials with d variables and degree s is defined as

$$t(\boldsymbol{x}) \coloneqq \sum_{\|\boldsymbol{n}\|_1 \le s} c_{\boldsymbol{n}} e^{i\boldsymbol{n}\cdot\boldsymbol{x}} \tag{B.26}$$

where $c_{\boldsymbol{n}} \in \mathbb{C}$, $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$, $\boldsymbol{\nu} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{Z}^d$, and $e^{i\boldsymbol{n}\cdot\boldsymbol{x}} = e^{in_1x_1} e^{in_2x_2} \cdots e^{in_dx_d}$. Consider a trigonometric monomial $c_{\boldsymbol{n}} e^{i\boldsymbol{n}\cdot\boldsymbol{x}} = c_{\boldsymbol{n}} e^{in_1x_1} e^{in_2x_2} \cdots e^{in_dx_d}$ such that $|c_{\boldsymbol{n}} e^{i\boldsymbol{n}\cdot\boldsymbol{x}}| \le 1$ for all $\boldsymbol{x} \in \mathbb{R}^d$ and $\|\boldsymbol{n}\|_1 \le s$, we could apply the single-qubit PQC with R_Z encoding as defined in Eq. (A.8) on each x_j for $1 \le j \le d$ respectively.

Lemma S6. *Given a trigonometric monomial $c_{\boldsymbol{n}} e^{i\boldsymbol{n}\cdot\boldsymbol{x}} = c_{\boldsymbol{n}} e^{in_1x_1} e^{in_2x_2} \cdots e^{in_dx_d}$ such that $|c_{\boldsymbol{n}} e^{i\boldsymbol{n}\cdot\boldsymbol{x}}| \le 1$ for all $\boldsymbol{x} \in \mathbb{R}^d$ and $\|\boldsymbol{n}\|_1 \le s$, there exists a PQC $U^{\boldsymbol{n}}(\boldsymbol{x})$ such that*

$$\langle 0|^{\otimes d} U^{\boldsymbol{n}}(\boldsymbol{x}) |0\rangle^{\otimes d} = c_{\boldsymbol{n}} e^{i\boldsymbol{n}\cdot\boldsymbol{x}}. \tag{B.27}$$

The width of the PQC is at most d, the depth is at most $6s + 3$, and the number of parameters is at most $4s + 3d$.

Proof. By Corollary S4, we could construct d single-qubit PQCs $U^{n_1}_{\boldsymbol{\theta}_1,\boldsymbol{\phi}_1}(x_1), U^{n_2}_{\boldsymbol{\theta}_2,\boldsymbol{\phi}_2}(x_2), \ldots, U^{n_d}_{\boldsymbol{\theta}_d,\boldsymbol{\phi}_d}(x_d)$ such that

$$\begin{aligned}
\langle 0| U^{n_1}_{\boldsymbol{\theta}_1,\boldsymbol{\phi}_1}(x_1) |0\rangle &= c_{\boldsymbol{n}} e^{in_1x_1}, \\
\langle 0| U^{n_2}_{\boldsymbol{\theta}_2,\boldsymbol{\phi}_2}(x_2) |0\rangle &= e^{in_2x_2}, \\
&\cdots \\
\langle 0| U^{n_d}_{\boldsymbol{\theta}_d,\boldsymbol{\phi}_d}(x_d) |0\rangle &= e^{in_dx_d},
\end{aligned}$$

where the number of layers of each PQC is $L_j = n_j$ for $1 \leq j \leq d$. We then define a d-qubit PQC as

$$U^{\boldsymbol{n}}(\boldsymbol{x}) = \bigotimes_{j=1}^{d} U^{n_j}_{\boldsymbol{\theta}_j,\boldsymbol{\phi}_j}(x_j), \tag{B.28}$$

which gives

$$\langle 0|^{\otimes d} U^{\boldsymbol{n}}(\boldsymbol{x})|0\rangle^{\otimes d} = \prod_{j=1}^{d} \langle 0|U^{n_j}_{\boldsymbol{\theta}_j,\boldsymbol{\phi}_j}(x_j)|0\rangle = c_{\boldsymbol{n}} e^{i\boldsymbol{n}\cdot\boldsymbol{x}}. \tag{B.29}$$

Since $\|\boldsymbol{n}\|_1 = \sum_{j=1}^{d} n_j \leq s$, we can conclude that the depth of $U^{\boldsymbol{n}}(\boldsymbol{x})$ is at most $6s+3$ and the number of parameters in $U^{\boldsymbol{n}}(\boldsymbol{x})$ is at most $4s+3d$. □

Then we could apply the technique of LCU on the PQCs $U^{\boldsymbol{n}}(\boldsymbol{x})$ to implement the operator

$$U_t(\boldsymbol{x}) \coloneqq \sum_{\|\boldsymbol{n}\|_1 \leq s} U^{\boldsymbol{n}}(\boldsymbol{x}), \tag{B.30}$$

so that we can implement the multivariate trigonometric polynomial as

$$\langle +|^{\otimes d} U_t(\boldsymbol{x})|+\rangle^{\otimes d} = \sum_{\|\boldsymbol{n}\|_1 \leq s} \langle +|^{\otimes d} U^{\boldsymbol{n}}(\boldsymbol{x})|+\rangle^{\otimes d} = \sum_{\|\boldsymbol{n}\|_1 \leq s} c_{\boldsymbol{n}} e^{i\boldsymbol{n}\cdot\boldsymbol{x}} = t(\boldsymbol{x}). \tag{B.31}$$

Note that the number of terms in the summation is

$$\sum_{\|\boldsymbol{n}\|_1 \leq s} 1 = \sum_{j=0}^{s} \sum_{\|\boldsymbol{n}\|_1 = j} 1 \leq \sum_{j=0}^{s} d^{2s} \leq (s+1)d^{2s}. \tag{B.32}$$

Then, we have the following proposition.

Proposition S7. *For any multivariate trigonometric polynomial $t(\boldsymbol{x})$ with d variables and degree s such that $|t(\boldsymbol{x})| \leq 1$ for $\boldsymbol{x} \in \mathbb{R}^d$, there exists a PQC $W_{tri}(\boldsymbol{x})$ such that*

$$f_{W_{tri}}(\boldsymbol{x}) \coloneqq \langle 0| W^{\dagger}_{tri}(\boldsymbol{x}) Z^{(0)} W_{tri}(\boldsymbol{x}) |0\rangle = t(\boldsymbol{x}) \tag{B.33}$$

where $Z^{(0)}$ is the Pauli Z observable on the first qubit. The width of the PQC is $O(d+\log s + s\log d)$, the depth is $O(s^2 d^{2s}(\log s + s\log d))$, and the number of parameters is $O(sd^{2s}(s+d))$.

The proof is similar to Theorem 1. This result demonstrates the universal approximation property of PQC in the perspective of multivariate Fourier series, which yields similar results as in Schuld et al. [34]. Notably, the PQC in Proposition S7 has an explicit construction without any assumption, improving the implicit PQCs proposed in Schuld et al. [34] in terms of circuit size. For instance, to implement the d-variable Fourier series with degree s, the PQC with parallel structure in Schuld et al. [34] requires width $O(ds)$ and potentially exponential depth $O(4^{ds})$.

C Approximating continuous functions via PQCs

We have constructively shown in the previous section that PQCs could implement multivariate polynomials. To study the approximation capabilities of PQC, a natural strategy involves aggregating multiple polynomials to approximate the continuous function, drawing on well-established principles from classical approximation theory. In the context of univariate functions, this endeavor is guided by the Stone-Weierstrass Theorem [69]. For the multivariate case, we accomplish this task by employing PQCs to implement Bernstein polynomials, followed by the established result on the approximation error bound of Bernstein polynomials [52, 53].

C.1 Established results of Bernstein polynomials approximation

For a d-variable continuous function $f : \mathbb{R}^d \to \mathbb{R}$, the multivariate Bernstein polynomial with degree $n \in \mathbb{N}$ of f is defined as

$$B_n(f;\boldsymbol{x}) \coloneqq \sum_{k_1=0}^{n} \cdots \sum_{k_d=0}^{n} f\left(\frac{\boldsymbol{k}}{n}\right) \prod_{j=1}^{d} \binom{n}{k_j} x_j^{k_j}(1-x_j)^{n-k_j}, \tag{C.34}$$

and $\boldsymbol{k} = (k_1, \ldots, k_d) \in \{0, \ldots, n\}^d$. Then, we have the following lemma on the approximation error bound of the Bernstein polynomial.

Lemma S8 (Bernstein polynomials approximation for Lipschitz functions [53])**.** *Given a Lipschitz continuous function $f : [0,1]^d \to \mathbb{R}$ with Lipschitz constant ℓ, which is defined as $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq \ell \|\boldsymbol{x} - \boldsymbol{y}\|_\infty$. Let f be bounded by Γ. The approximation error of the n-degree Bernstein polynomial of f scales as*

$$|f(\boldsymbol{x}) - B_n(f; \boldsymbol{x})| \leq \varepsilon + 2\Gamma \sum_{j=1}^{d} \binom{d}{j} \left(\frac{\ell^2}{4n\varepsilon^2}\right)^j \leq \varepsilon + 2\Gamma \left(\left(1 + \frac{\ell^2}{4n\varepsilon^2}\right)^d - 1\right), \tag{C.35}$$

where $\varepsilon > 0$ is an arbitrarily small quantity.

Proof. Drawing inspiration from the Lipschitz continuity of the target function f, we define $\delta = \epsilon/\ell$. Consequently, for any two points $\boldsymbol{x} = (x_1, \ldots, x_d)$ and $\boldsymbol{y} = (y_1, \ldots, y_d)$ such that $|x_i - y_i| < \delta$ for all $i \in \{1, \ldots, d\}$, it follows that $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq \varepsilon$. The target function can be written as

$$\begin{aligned} f(\boldsymbol{x}) &= f(x_1, \ldots, x_d) \\ &= f(x_1, \cdots, x_d) \sum_{k_1=0}^{n} \cdots \sum_{k_d=0}^{n} \prod_{i=1}^{d} \binom{n}{k_i} x_i^{k_i} (1 - x_i)^{n-k_i} \\ &= \sum_{k_1=0}^{n} \cdots \sum_{k_d=0}^{n} f(x_1, \cdots, x_d) \prod_{i=1}^{d} \binom{n}{k_i} x_i^{k_i} (1 - x_i)^{n-k_i}. \end{aligned} \tag{C.36}$$

Let us consider the set $E = \prod_{i=1}^d \{0, 1, \ldots, n\}$, and for $j = 1, 2, \ldots, d$, we define the sets

$$\Omega_j = \{k_j \in \{0, 1, \ldots, n\} : |\frac{k_i}{n} - x_j| < \delta\} \text{ and } F = E \setminus (\Omega_1 \times \cdots \times \Omega_d). \tag{C.37}$$

Then, $F = \bigcup_{k=1}^d F_k$, with $F_k = \left\{\prod_{i=1}^d \Omega_{ik}^{[\alpha_{ik}]} \in F : \alpha_{ik} \in \{0, 1\}, \quad \sum_{i=1}^d \alpha_{ik} = k\right\}$, where $\Omega_{ik}^{[\alpha_{ik}]} = \begin{cases} \Omega_i & \text{if } \alpha_{ik} = 0 \\ \Omega_i^c & \text{if } \alpha_{ik} = 1 \end{cases}$ and $\Omega_i^c = \left\{k_i \in \{0, \cdots, n\} : \left|\frac{k_i}{n} - x_i\right| \geq \delta\right\}$. For $A_k = \prod_{i=1}^d \Omega_{ik}^{[\alpha_{ik}]} \in F_k, k = 1, \cdots, d$, let us define also $I_{A_k} = \{i \in \{1, \cdots, d\} : \alpha_{ik} = 1\}$ (that means card $(I_{A_k}) = k \geq 1$). We have

$$\begin{aligned} & |f(x_1, \cdots, x_d) - B_n(f; x_1, \cdots, x_d)| \\ =& \Big| \sum_{k_1=0}^{n} \cdots \sum_{k_d=0}^{n} f(x_1, \cdots, x_d) \prod_{i=1}^{d} \binom{n}{k_i} x_i^{k_i} (1 - x_i)^{n-k_i} \\ & - \sum_{k_1=0}^{n} \cdots \sum_{k_d=0}^{n} f\left(\frac{k_1}{n}, \cdots, \frac{k_d}{n}\right) \prod_{i=1}^{d} \binom{n}{k_i} x_i^{k_i} (1 - x_i)^{n-k_i} \Big| \\ =& \left| \sum_{k_1=0}^{n} \cdots \sum_{k_d=0}^{n} \left[f(x_1, \cdots, x_d) - f\left(\frac{k_1}{n}, \cdots, \frac{k_d}{n}\right)\right] \prod_{i=1}^{d} \binom{n}{k_i} x_i^{k_i} (1 - x_i)^{n-k_i} \right| \\ \leq& \sum_{k_1=0}^{n} \cdots \sum_{k_d=0}^{n} \left| f(x_1, \cdots, x_d) - f\left(\frac{k_1}{n}, \cdots, \frac{k_d}{n}\right) \right| \prod_{i=1}^{d} \binom{n}{k_i} x_i^{k_i} (1 - x_i)^{n-k_i} \\ \leq& \sum_{\Omega_1} \cdots \sum_{\Omega_d} \left| f(x_1, \cdots, x_d) - f\left(\frac{k_1}{n}, \cdots, \frac{k_d}{n}\right) \right| \prod_{i=1}^{d} \binom{n}{k_i} x_i^{k_i} (1 - x_i)^{n-k_i} \\ & + \sum_{F} \left| f(x_1, \cdots, x_d) - f\left(\frac{k_1}{n}, \cdots, \frac{k_d}{n}\right) \right| \prod_{i=1}^{d} \binom{n}{k_i} x_i^{k_i} (1 - x_i)^{n-k_i}. \end{aligned} \tag{C.38}$$

Using the fact that f is continuous and bounded, we get

$$\begin{aligned}
&\left|f\left(x_1,\cdots,x_d\right)-B_n\left(f;x_1,\cdots,x_d\right)\right| \\
\leq&\varepsilon\sum_{\Omega_1}\cdots\sum_{\Omega_d}\prod_{i=1}^{d}\left(\begin{array}{c}n\\k_i\end{array}\right)x_i^{k_i}\left(1-x_i\right)^{n-k_i}+2\Gamma\sum_{F}\prod_{i=1}^{d}\left(\begin{array}{c}n\\k_i\end{array}\right)x_i^{k_i}\left(1-x_i\right)^{n-k_i} \\
\leq&\varepsilon+2\Gamma\sum_{l=1}^{d}\sum_{A_l\in F_l}\prod_{i=1}^{d}\left(\begin{array}{c}n\\k_i\end{array}\right)x_i^{k_i}\left(1-x_i\right)^{n-k_i} \\
\leq&\varepsilon+2\Gamma\sum_{l=1}^{d}\sum_{A_l\in F_l}\prod_{i\in I_{A_l}}\frac{1}{4n\delta^2} \\
=&\varepsilon+2\Gamma\sum_{j=1}^{d}\binom{d}{j}\frac{1}{(4n\delta^2)^j}\leq\varepsilon+2\Gamma\left(\left(1+\frac{\ell^2}{4n\varepsilon^2}\right)^d-1\right).
\end{aligned} \tag{C.39}$$

This completes the proof. A more detailed expansion of Eq. (C.39) can be seen in Theorem 2 in Foupouagnigni and Mouafo Wouodjié [53]. □

Remark S2. *Here, it is important to observe that for a continuous target function, denoted as $f(\boldsymbol{x})$, there exists a value of $\delta>0$ such that:*

$$\left|f\left(x_1,\cdots,x_d\right)-B_n\left(f;x_1,\cdots,x_d\right)\right|\leq\varepsilon+2\Gamma\left(\left(1+\frac{1}{4n\delta^2}\right)^d-1\right).$$

This expression signifies the convergence rate of the Bernstein polynomial for general continuous functions.

C.2 Implement Bernstein polynomials via PQCs

In Lemma S8, we have defined the Bernstein polynomial and its approximation error towards the Lipschitz continuous function. Guided by Theorem 1, we can construct a PQC to implement such a Bernstein polynomial.

Lemma S9. *For any d-variable Bernstein polynomial with degree $n\in\mathbb{N}$ defined in Eq. (C.34) such that $|B_n(f;\boldsymbol{x})|\leq 1$ for $\boldsymbol{x}\in[0,1]^d$, there exist a PQC $W_b(\boldsymbol{x})$ satisfying*

$$f_{W_b}(\boldsymbol{x}):=\langle 0|\,W_b^\dagger(\boldsymbol{x})Z^{(0)}W_b(\boldsymbol{x})\,|0\rangle=B_n(f;\boldsymbol{x}). \tag{C.40}$$

The width of the PQC is $O(d\log n)$, the depth is $O\big(dn^d\log n\big)$, and the number of parameters is $O(dn^d)$.

Proof. We undertake a two-step process in the proof of Lemma S9. Initially, we construct PQCs to provide an exact representation of $f\left(\frac{\boldsymbol{k}}{n}\right)\prod_{j=1}^d\binom{n}{k_j}x_j^{k_j}(1-x_j)^{n-k_j}$ for all $\boldsymbol{k}\in\{0,1,\ldots,n\}^d$. Subsequently, we employ LCU to aggregate these PQCs for the purpose of approximating the Bernstein polynomial described in Eq. (C.34).

The univariate polynomial $x^k(1-x)^{n-k}$ can be represented by a PQC. The depth of this PQC is less than $2n+1$, the width is 2, and the number of parameters is $n+2$. The multivariate polynomial $f\left(\frac{\boldsymbol{k}}{n}\right)\prod_{j=1}^d\binom{n}{k_j}x_j^{k_j}(1-x_j)^{n-k_j}$ can be exactly represented by the product of the univariate polynomial $x^k(1-x)^{n-k}$. The same routine has been employed in Lemma S5. The depth of this PQC is less than $2n+1$, the width is $2d$, and the number of parameters is $d(n+2)$.

The number of terms in the summation in Eq. (C.34) is $(n+1)^d$. We can employ the same routine in Theorem 1 to construct the PQC $W_b(\boldsymbol{x})$. The depth of W_b scales as

$$O\Big(\big(d(n+1)^{d+1}\log{(n+1)}+d\big)\Big),$$

the width is $2d+d\log{(n+1)}$, and the number of parameters is $(n+1)^d(n+2)d$. The results presented in Lemma S9 can be obtained after simplification. □

C.3 PQC approximating continuous functions

We have successfully derived results regarding the approximation error between PQCs and Bernstein polynomials and between Bernstein polynomials and continuous functions. Leveraging these established findings, we can now formulate a rigorous assertion regarding the universal approximation theorem and the error bound of PQCs, employing the well-established principles of triangle inequality.

Theorem 2 (The Universal Approximation Theorem of PQC)**.** *For any continuous function $f : [0,1]^d \to [-1,1]$, given an $\varepsilon > 0$, there exist an $n \in \mathbb{N}$ and a PQC $W_b(\boldsymbol{x})$ with width $O(d \log n)$, depth $O(dn^d \log n)$ and the number of trainable parameters $O(dn^d)$ such that*

$$|f(\boldsymbol{x}) - f_{W_b}(\boldsymbol{x})| \leq \varepsilon \tag{C.41}$$

for all $\boldsymbol{x} \in [0,1]^d$, where $f_{W_b}(\boldsymbol{x}) \coloneqq \langle 0| W_b^\dagger(\boldsymbol{x}) Z^{(0)} W_b(\boldsymbol{x}) |0\rangle$.

Proof. Remark S2 has established the uniform convergence of the Bernstein polynomial towards any continuous function within the cubic domain $[0,1]^d$, denoted as $B_n(f;\boldsymbol{x})$, with the property that $B_n(f;\boldsymbol{x}) \to f(\boldsymbol{x})$ as $n \to +\infty$. Building on Lemma S9, we can effectively implement this Bernstein polynomial $B_n(f;\boldsymbol{x})$ using $f_{W_b}(\boldsymbol{x})$. The depth of the PQC $W_b(\boldsymbol{x})$ is $O\big(dn^d \log n\big)$, the width is $O(d \log n)$, and the number of parameters is $O(dn^d)$. This completes the proof. □

Theorem 3. *Given a Lipschitz continuous function $f : [0,1]^d \to [-1,1]$ with a Lipschitz constant ℓ, for any $\varepsilon > 0$ and $n \in \mathbb{N}$, there exists a PQC $W_b(\boldsymbol{x})$ with such that $f_{W_b}(\boldsymbol{x}) \coloneqq \langle 0| W_b^\dagger(\boldsymbol{x}) Z^{(0)} W_b(\boldsymbol{x}) |0\rangle$ satisfies*

$$|f(\boldsymbol{x}) - f_{W_b}(\boldsymbol{x})| \leq \varepsilon + 2\left(\left(1 + \frac{\ell^2}{n\varepsilon^2}\right)^d - 1\right) \leq \varepsilon + d2^d \frac{\ell^2}{n\varepsilon^2} \tag{C.42}$$

for all $\boldsymbol{x} \in [0,1]^d$. The width of the PQC is $O(d \log n)$, the depth is $O\big(dn^d \log n\big)$, and the number of parameters is $O(dn^d)$.

Proof. Lemma S8 has established the uniform convergence rate of the Bernstein polynomial towards any Lipschitz continuous function within the cubic domain $[0,1]^d$. We know that for any Lipschitz continuous function $f(\boldsymbol{x})$ with Lipschitz constant ℓ, there exists a Bernstein polynomial $B_n(f;\boldsymbol{x})$ satisfying

$$|f(\boldsymbol{x}) - B_n(f;\boldsymbol{x})| \leq \varepsilon + 2\Gamma \sum_{j=1}^{d} \binom{d}{j} \left(\frac{\ell^2}{4n\varepsilon^2}\right)^j \leq \varepsilon + 2\Gamma \left(\left(1 + \frac{\ell^2}{4n\varepsilon^2}\right)^d - 1\right).$$

Building on Lemma S9, we can effectively implement this Bernstein polynomial $B_n(f;\boldsymbol{x})$ using $f_{W_b}(\boldsymbol{x})$. The depth of the PQC $W_b(\boldsymbol{x})$ is $O\big(dn^d \log n\big)$, the width is $O(d \log n)$, and the number of parameters is $O(dn^d)$. This completes the proof. □

D Approximating smooth functions via nested PQCs

Other than using a Bernstein polynomial to approximate a continuous function globally, we could also utilize local polynomials to achieve a piecewise approximation. To do this, we follow the path of classical deep neural networks [18, 21, 25], using multivariate Taylor series expansion to approximate a multivariate smooth function f in some small local region. Let $\beta = s + r > 0$, $r = (0,1]$ and $s = \lfloor \beta \rfloor \in \mathbb{N}$, for a finite constant $B_0 > 0$, the β-Hölder class of functions $\mathcal{H}^\beta([0,1]^d, B_0)$ is defined as

$$\mathcal{H}^\beta([0,1]^d, B_0) = \left\{ f : [0,1]^d \to \mathbb{R}, \max_{\|\boldsymbol{\alpha}\|_1 \leq s} \|\partial^{\boldsymbol{\alpha}} f\|_\infty \leq B_0, \max_{\|\boldsymbol{\alpha}\|_1 = s} \sup_{\boldsymbol{x} \neq \boldsymbol{y}} \frac{|\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}) - \partial^{\boldsymbol{\alpha}} f(\boldsymbol{y})|}{\|\boldsymbol{x} - \boldsymbol{y}\|_2^r} \leq B_0 \right\}, \tag{D.43}$$

where $\partial^{\boldsymbol{\alpha}} = \partial^{\alpha_1} \cdots \partial^{\alpha_d}$ for $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$. By definition, for a function $f \in \mathcal{H}^\beta([0,1]^d, B_0)$, when $\beta \in (0,1)$, f is a Hölder continuous function with order β and Hölder constant B_0; when $\beta = 1$, f is a Lipschitz function with Lipschitz constant B_0; when $\beta > 1$, f belongs to the C^s class of functions whose s-th partial derivatives exist and are bounded.

We utilize the following lemma on the Taylor expansion of β-Hölder functions as a mathematical tool for constructing and analyzing the PQC approximation.

Lemma S10 ([18])**.** *Given a function $f \in \mathcal{H}^{\beta}([0,1]^d, 1)$ with $\beta = r + s$, $r \in (0,1]$ and $s \in \mathbb{N}^+$, for any $\boldsymbol{x}, \boldsymbol{x_0} \in [0,1]^d$, we have*

$$\Big| f(\boldsymbol{x}) - \sum_{\|\boldsymbol{\alpha}\|_1 \le s} \frac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x_0})}{\boldsymbol{\alpha}!} (\boldsymbol{x} - \boldsymbol{x_0})^{\boldsymbol{\alpha}} \Big| \le d^s \|\boldsymbol{x} - \boldsymbol{x_0}\|_2^{\beta}, \tag{D.44}$$

where $\boldsymbol{\alpha}! = \alpha_1! \cdots \alpha_d!$.

Next, we show how to construct PQCs to implement the Taylor expansion of β-Hölder functions.

D.1 Localization via PQC

As shown in Eq. (D.44), the Taylor expansion of a multivariate smooth function only converges in a fairly small local region. So, we need first to localize the entire region $[0,1]^d$. Given $K \in \mathbb{N}$ and $\Delta \in (0, \frac{1}{3K})$, for each $\boldsymbol{\eta} = (\eta_1, \dots, \eta_d) \in \{0, 1, \dots, K-1\}^d$, we define

$$Q_{\boldsymbol{\eta}} \coloneqq \Big\{ \boldsymbol{x} = (x_1, \dots, x_d) : x_i \in \big[\frac{\eta_i}{K}, \frac{\eta_i + 1}{K} - \Delta \cdot 1_{\eta_i < K-1}\big] \Big\}. \tag{D.45}$$

By the definition of $Q_{\boldsymbol{\eta}}$, the region $[0,1]^d$ is approximately divided into small hypercubes $\bigcup_{\boldsymbol{\eta}} Q_{\boldsymbol{\eta}}$ and some trifling region $\Lambda(d, K, \Delta) \coloneqq [0,1]^d \setminus (\bigcup_{\boldsymbol{\eta}} Q_{\boldsymbol{\eta}})$, as illustrated in Fig. 2 in the main text. Then we need to construct a PQC that maps any $x \in Q_{\boldsymbol{\eta}}$ to some fixed point $x_{\boldsymbol{\eta}} = \frac{\boldsymbol{\eta}}{K} \in Q_{\boldsymbol{\eta}}$, i.e., approximating the piecewise-constant function $D(\boldsymbol{x}) = \frac{\boldsymbol{\eta}}{K}$ if $\boldsymbol{x} \in Q_{\boldsymbol{\eta}}$, where $\frac{\boldsymbol{\eta}}{K} = (\eta_1/K, \dots, \eta_d/K)$. We consider the case of $d = 1$, where the localization function is

$$D(x) = \frac{k}{K}, \qquad \text{if } x \in \Big[\frac{k}{K}, \frac{k+1}{K} - \Delta \cdot 1_{k < K-1}\Big] \text{ for } k = 0, 1, \dots, K-1. \tag{D.46}$$

The multivariate case could be easily generalized by applying $D(x)$ to each variable x_j. The idea is to find a polynomial that approximates the sign function

$$\operatorname{sgn}(x - c) = \begin{cases} 1, & \text{if } x > c, \\ 0, & \text{if } x = c \\ -1, & \text{if } x < c \end{cases}, \tag{D.47}$$

as shown in the following lemma.

Lemma S11 (Polynomial approximation to the sign function $\operatorname{sgn}(x - c)$ [54])**.** $\forall c \in [-1,1], \Delta > 0, \varepsilon \in (0,1)$. *there exists an odd polynomial $P_{\Delta,\varepsilon}(x)$ of degree $n = O(\frac{1}{\Delta} \log \frac{1}{\varepsilon})$ that satisfies*

1. $|P_{\Delta,\varepsilon}(x - c)| \le 1$ *for all* $x \in [-1,1]$,
2. $|\operatorname{sgn}(x - c) - P_{\Delta,\varepsilon}(x - c)| \le \varepsilon$ *for all* $x \in [-1,1] \setminus (c - \frac{\Delta}{2}, c + \frac{\Delta}{2})$.

Note that we could also approximate the step function defined as $\operatorname{stp}(x - c) \coloneqq \frac{1}{2}\operatorname{sgn}(x - c) + \frac{1}{2}$ by the polynomial $P'_{\Delta,\varepsilon}(x - c) = \frac{1}{2} P_{\Delta,\varepsilon}(x - c) + \frac{1}{2}$ of degree $n = O(\frac{1}{\Delta} \log \frac{1}{\varepsilon})$, which satisfies that $|P'_{\Delta,\varepsilon}(x - c)| \le 1$ for all $x \in [-1,1]$ and $|\operatorname{stp}(x - c) - P'_{\Delta,\varepsilon}(x - c)| \le \frac{\varepsilon}{2}$ for all $x \in [-1,1] \setminus (c - \frac{\Delta}{2}, c + \frac{\Delta}{2})$. Note that the polynomial $P'_{\Delta,\varepsilon}(x - c)$ does not have definite parity and thus cannot be directly implemented by a PQC as shown in Corollary S2. Since only the domain $[0,1]$ is relevant to x, for $c \in (0,1)$, we could define an even polynomial

$$P^{\text{even}}_{c,\Delta,\varepsilon}(x) = \frac{1}{1 + \frac{\varepsilon}{2}} \big(P'_{\Delta,\varepsilon}(x - c) + P'_{\Delta,\varepsilon}(-x - c) \big) \tag{D.48}$$

such that $|P^{\text{even}}_{c,\Delta,\varepsilon}(x)| \le 1$ for all $x \in [-1,1]$ and $|\operatorname{stp}(x - c) - P^{\text{even}}_{c,\Delta,\varepsilon}(x)| \le \frac{\varepsilon}{2}$ for all $x \in [0,1] \setminus (c - \frac{\Delta}{2}, c + \frac{\Delta}{2})$. The piecewise-constant function $D(x)$ can be written as a combination of step functions,

$$D(x) = \sum_{k=1}^{K-1} \frac{1}{K} \operatorname{stp}\big(x - \frac{k}{K} + \frac{\Delta}{2}\big). \tag{D.49}$$

Then we could find even polynomials $P^{\text{even}}_{c,\Delta,\varepsilon}(x)$ that approximate $\operatorname{stp}\big(x - \frac{k}{K} + \frac{\Delta}{2}\big)$ for each k. Combining those polynomials together as in Eq. (D.49), we have the following lemma.

Lemma S12. *Given $K \in \mathbb{N}$ and $\Delta \in (0, \frac{1}{3K})$, there exists an even polynomial $P_{\Delta,\varepsilon}(x)$ of degree $n = O(\frac{1}{\Delta}\log\frac{K}{\varepsilon})$ that satisfies*

1. $|P_{\Delta,\varepsilon}(x)| \leq 1$ *for all* $x \in [-1, 1]$,
2. $|D(x) - P_{\Delta,\varepsilon}(x)| \leq \varepsilon$ *for all* $x \in \bigcup_{k=0}^{K-1}\left[\frac{k}{K}, \frac{k+1}{K} - \Delta \cdot 1_{k<K-1}\right]$.

Note that we could shift the polynomial $P_{\Delta,\varepsilon}(x)$ such that $P_{\Delta,\varepsilon}(x) - D(x) \in (0, \varepsilon)$ without changing the degree. It follows that we can construct a PQC to implement the polynomial $P_{\Delta,\varepsilon}(x)$ by Corollary S2.

Corollary S13. *Given $K \in \mathbb{N}$, $\Delta \in (0, \frac{1}{3K})$ and $\varepsilon \in (0, \frac{1}{K})$, there exists a single-qubit PQC $U_D(x)$ of depth $O(\frac{1}{\Delta}\log\frac{K}{\varepsilon})$ that satisfies*

$$\langle +|U_D(x)|+\rangle - \frac{k}{K} \in (0, \varepsilon) \quad \textit{if } x \in \left[\frac{k}{K}, \frac{k+1}{K} - \Delta \cdot 1_{k<K-1}\right] \textit{ for } k = 0, 1, \ldots, K-1. \tag{D.50}$$

Note that ε has to be bounded by $\frac{1}{K}$, which is the length of the localized region. We could further implement such a localization procedure for $\boldsymbol{x} = (x_1, \ldots, x_d)$ on the region $[0, 1]^d$ by applying the PQC for each x_j, as stated in the following corollary.

Lemma S14 (Localization via PQC)**.** *Given $K \in \mathbb{N}$, $\Delta \in (0, \frac{1}{3K})$ and $\varepsilon \in (0, \frac{1}{K})$, there exists a PQC $W_D(\boldsymbol{x})$ of width $O(d)$ and depth $O(\frac{1}{\Delta}\log\frac{K}{\varepsilon})$ implementing a localization function $f_{W_D}(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}^d$ such that*

$$\mathbf{0} \leq f_{W_D}(\boldsymbol{x}) - \frac{\boldsymbol{\eta}}{K} \leq \boldsymbol{\varepsilon} \quad \textit{if } \boldsymbol{x} \in Q_{\boldsymbol{\eta}}, \tag{D.51}$$

where $\mathbf{0} = (0, \ldots, 0)$ and $\boldsymbol{\varepsilon} = (\varepsilon, \ldots, \varepsilon)$ are d-dimensional vectors.

Proof. We construct a d-qubit PQC $W_D(\boldsymbol{x}) \coloneqq \bigotimes_{j=1}^{d} U_D(x_j)$ where the single-qubit PQC $U_D(x)$ is constructed in Corollary S13. Then we apply the Hadamard test on each $U_D(x_j)$ to obtain $f_{U_D}(x_j) \coloneqq \langle +|U_D(x_j)|+\rangle$. Let $f_{W_D}(\boldsymbol{x}) \coloneqq (f_{U_D}(x_1), \ldots, f_{U_D}(x_d))$, which implements the localization function as required. $\square$

D.2 Implementing the Taylor coefficients by PQC

Next, we use PQC to implement the Taylor coefficients, which is essentially a point-fitting problem. For each $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_d) \in \{0, 1, \ldots, K-1\}^d$ and $\boldsymbol{\alpha}$, we denote $\xi_{\boldsymbol{\eta},\boldsymbol{\alpha}} \coloneqq \frac{\partial^{\boldsymbol{\alpha}} f(\frac{\boldsymbol{\eta}}{K})}{\boldsymbol{\alpha}!} \in [-1, 1]$. Then we could construct the following PQC,

$$U_{co}^{\boldsymbol{\alpha}} = \sum_{\boldsymbol{\eta}} |\boldsymbol{\eta}\rangle\langle\boldsymbol{\eta}| \otimes R_X(\theta_{\boldsymbol{\eta},\boldsymbol{\alpha}}), \tag{D.52}$$

where $|\boldsymbol{\eta}\rangle = |\eta_1\rangle \otimes \cdots \otimes |\eta_d\rangle$ and $\theta_{\boldsymbol{\eta},\boldsymbol{\alpha}} = 2\arccos(\xi_{\boldsymbol{\eta},\boldsymbol{\alpha}})$. It gives the following lemma.

Lemma S15. *Given a β-Hölder smooth function $f : [0, 1]^d \to [-1, 1]$, for any $\boldsymbol{\alpha} \in \mathbb{N}^d$ and $\boldsymbol{\eta} \in \{0, 1, \ldots, K-1\}^d$, there exists a PQC $U_{co}^{\boldsymbol{\alpha}}$ such that*

$$\langle \boldsymbol{\eta}, 0| U_{co}^{\boldsymbol{\alpha}} |\boldsymbol{\eta}, 0\rangle = \xi_{\boldsymbol{\eta},\boldsymbol{\alpha}}. \tag{D.53}$$

The width of the PQC is $O(d\log K)$, and the depth is $O(K^d)$.

We note that the state $|\boldsymbol{\eta}\rangle$ can be prepared using basis encoding according to the results of localization in Lemma S14.

D.3 Implementing multivariate Taylor series by PQC

To implement the multivariate Taylor expansion of a function, we first build a PQC to represent a single term in the Taylor series, which could be done by combining the monomial implementation in Lemma S5 and the Taylor coefficient implementation in Lemma S15. Thus, we have the following corollary.

Corollary S16. *For any β-Hölder smooth function f, given an $\boldsymbol{\alpha} \in \mathbb{N}^d$ with $\|\boldsymbol{\alpha}\|_1 \leq s$ for $s \in \mathbb{N}^+$ and an $\boldsymbol{\eta} \in \{0, 1, \ldots, K-1\}^d$, there exists a PQC $U_{\boldsymbol{\eta}}^{\boldsymbol{\alpha}}(\boldsymbol{x})$ such that*

$$\langle \boldsymbol{\eta}, 0| \langle +|^{\otimes d} U_{\boldsymbol{\eta}}^{\boldsymbol{\alpha}}(\boldsymbol{x}) |\boldsymbol{\eta}, 0\rangle |+\rangle^{\otimes d} = \frac{\partial^{\boldsymbol{\alpha}} f(\frac{\boldsymbol{\eta}}{K})}{\boldsymbol{\alpha}!}(\boldsymbol{x} - \frac{\boldsymbol{\eta}}{K})^{\boldsymbol{\alpha}}. \tag{D.54}$$

The width of the PQC is $O(d \log K)$, the depth is $O(K^d + s)$, and the number of parameters is at most $K^d + s + d$.

Proof. Let $U_{\boldsymbol{\eta}}^{\boldsymbol{\alpha}}(\boldsymbol{x}) \coloneqq U_{co}^{\boldsymbol{\alpha}} \otimes U^{\boldsymbol{\alpha}}(\boldsymbol{x} - \frac{\boldsymbol{\eta}}{K})$, where $U_{co}^{\boldsymbol{\alpha}}$ is defined in Lemma S15 and $U^{\boldsymbol{\alpha}}(\boldsymbol{x} - \frac{\boldsymbol{\eta}}{K})$ is defined in Lemma S5 with changing input from $\boldsymbol{x}$ to $\boldsymbol{x} - \frac{\boldsymbol{\eta}}{K}$. Then the corollary directly follows from Lemma S5 and Lemma S15. $\square$

The next step is to combine single Taylor terms together to implement the truncated Taylor expansion of the target function. The method is in the same spirit as what is utilized in Theorem 1, i.e., using LCU to achieve the following (unnormalized) operator,

$$U_t(\boldsymbol{x}) \coloneqq \sum_{\|\boldsymbol{\alpha}\|_1 \leq s} U_{\boldsymbol{\eta}}^{\boldsymbol{\alpha}}(\boldsymbol{x}). \tag{D.55}$$

Then we can implement the Taylor expansion of the function f at point $\frac{\boldsymbol{\eta}}{K}$ as

$$\langle \boldsymbol{\eta}, 0| \langle +|^{\otimes d} U_t(\boldsymbol{x}) |\boldsymbol{\eta}, 0\rangle |+\rangle^{\otimes d} = \sum_{\|\boldsymbol{\alpha}\|_1 \leq s} \frac{\partial^{\boldsymbol{\alpha}} f(\frac{\boldsymbol{\eta}}{K})}{\boldsymbol{\alpha}!}(\boldsymbol{x} - \frac{\boldsymbol{\eta}}{K})^{\boldsymbol{\alpha}}. \tag{D.56}$$

Hence we have the following lemma.

Lemma S17. *Given a function $f \in \mathcal{H}^{\beta}([0,1]^d, 1)$ with $\beta = r + s$, $r \in (0, 1]$ and $s \in \mathbb{N}^+$, for any $\boldsymbol{\eta} \in \{0, \ldots, K-1\}^d$, there exists a PQC $W_e(\boldsymbol{x}, \frac{\boldsymbol{\eta}}{K})$ such that $f_{W_e}(\boldsymbol{x}) \coloneqq \langle 0| W_e^{\dagger}(\boldsymbol{x}) Z^{(0)} W_e(\boldsymbol{x}) |0\rangle$ implements the truncated Taylor expansion at point $\frac{\boldsymbol{\eta}}{K}$,*

$$f_{W_e}(\boldsymbol{x}) = \sum_{\|\boldsymbol{\alpha}\|_1 \leq s} \frac{\partial^{\boldsymbol{\alpha}} f(\frac{\boldsymbol{\eta}}{K})}{\alpha!}(\boldsymbol{x} - \frac{\boldsymbol{\eta}}{K})^{\boldsymbol{\alpha}}. \tag{D.57}$$

The depth of the PQC is $O(s^2 d^s K^d (\log s + s \log d + d \log K))$, the width is $O(d \log K + \log s + s \log d)$, and the number of parameters is $O(s d^s (s + d + K^d))$.

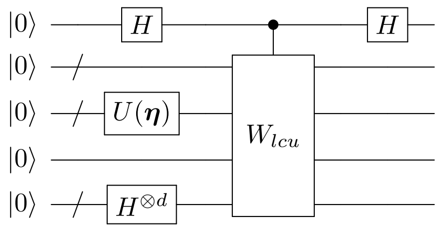
Proof. The idea of constructing the PQC $W_e(\boldsymbol{x}, \frac{\boldsymbol{\eta}}{K})$ is similar to the construction of $W_p(\boldsymbol{x})$ in Theorem 1. The only difference is that here we apply LCU on unitaries $U_{\boldsymbol{\eta}}^{\boldsymbol{\alpha}}(\boldsymbol{x}) \coloneqq U_{co}^{\boldsymbol{\alpha}} \otimes U^{\boldsymbol{\alpha}}(\boldsymbol{x} - \frac{\boldsymbol{\eta}}{K})$ instead of $U^{\boldsymbol{\alpha}}(\boldsymbol{x})$. Thus, the controlled unitary is

$$U_c(\boldsymbol{x}, \frac{\boldsymbol{\eta}}{K}) = \sum_{j=1}^{T} |j\rangle\langle j| \otimes U_{\boldsymbol{\eta}}^{\boldsymbol{\alpha}^{(j)}}(\boldsymbol{x}) \tag{D.58}$$

and the unitary $W_{lcu}(\boldsymbol{x}, \frac{\boldsymbol{\eta}}{K}) = (F^{\dagger} \otimes I) U_c(\boldsymbol{x}, \frac{\boldsymbol{\eta}}{K})(F \otimes I)$ satisfies that

$$\langle 0| \langle \boldsymbol{\eta}, 0| \langle +|^{\otimes d} W_{lcu}(\boldsymbol{x}, \frac{\boldsymbol{\eta}}{K}) |0\rangle |\boldsymbol{\eta}, 0\rangle |+\rangle^{\otimes d} = \sum_{\|\boldsymbol{\alpha}\|_1 \leq s} \frac{\partial^{\boldsymbol{\alpha}} f(\frac{\boldsymbol{\eta}}{K})}{\boldsymbol{\alpha}!}(\boldsymbol{x} - \frac{\boldsymbol{\eta}}{K})^{\boldsymbol{\alpha}}. \tag{D.59}$$

We then apply the Hadamard test on $W_{lcu}(\boldsymbol{x}, \frac{\boldsymbol{\eta}}{K})$, giving the quantum circuit $W_e(\boldsymbol{x}, \frac{\boldsymbol{\eta}}{K})$ as below



where the unitary $U(\boldsymbol{\eta})$ takes $\boldsymbol{\eta}$ as input and maps $|0\rangle$ to $|\boldsymbol{\eta}\rangle$. Note that the controlled unitary $U_c(\boldsymbol{x}, \frac{\boldsymbol{\eta}}{K})$ could be implemented by $O(T(s+1))$ number of $(\log T)$-qubit controlled gates and $O(TK^d)$ number of $(\log T + d\log K)$-qubit controlled gates. An n-qubit controlled gate could be implemented by a quantum circuit consisting of CNOT gates and single-qubit gates with depth $O(n)$ [49]. Thus $U_c(\boldsymbol{x})$ could be implemented by a quantum circuit with depth $O((s+1)T\log T + TK^d(\log T + d\log K))$ and width $O(d + \log T + d\log K)$. Then the depth and width of $W_{lcu}(\boldsymbol{x}, \frac{\boldsymbol{\eta}}{K}) = (F^\dagger \otimes I)U_c(\boldsymbol{x}, \frac{\boldsymbol{\eta}}{K})(F \otimes I)$ are in the same order of $U_c(\boldsymbol{x}, \frac{\boldsymbol{\eta}}{K})$ since F is simply tensor of Hadamard gates. Therefore the entire depth of the circuit W_e is $O((sT\log T + TK^d(\log T + d\log K)))$ and the width is $O(d + \log T + d\log K)$. As $T \leq (s+1)d^s$, we have the depth and width of PQC shown in Lemma S17. Note that the number of parameters in the PQC equals the number of parameters in $U_c(\boldsymbol{x})$, which is $O(T(s + d + K^d))$. □

Finally, we combine the steps of localization and the Taylor series implementation to achieve a local Taylor expansion for the target function. The PQC is in a nested structure consisting of a PQC for localization and a PQC for Taylor series; see the detailed construction in the following theorem.

Theorem 4. *Given a function $f \in \mathcal{H}^\beta([0,1]^d, 1)$ with $\beta = r + s$, $r \in (0,1]$ and $s \in \mathbb{N}^+$, for any $K \in \mathbb{N}$ and $\Delta \in (0, \frac{1}{3K})$, there exists a PQC $W_t(\boldsymbol{x})$ such that $f_{W_t}(\boldsymbol{x}) := \langle 0| W_t^\dagger(\boldsymbol{x}) Z^{(0)} W_t(\boldsymbol{x}) |0\rangle$ satisfies*

$$|f(\boldsymbol{x}) - f_{W_t}(\boldsymbol{x})| \leq d^{s+\beta/2} K^{-\beta} \tag{D.60}$$

for $\boldsymbol{x} \in \bigcup_{\boldsymbol{\eta}} Q_{\boldsymbol{\eta}}$. The width of the PQC is $O(d\log K + \log s + s\log d)$, the depth is $O(s^2 d^s K^d(\log s + s\log d + d\log K)) + \frac{1}{\Delta}\log K)$, and the number of parameters is $O(sd^s(s + d + K^d) + \frac{d}{\Delta}\log K)$.

Proof. By Lemma S10, we have the following error bound for $\boldsymbol{x} \in Q_{\boldsymbol{\eta}}$,

$$\Big| f(\boldsymbol{x}) - \sum_{\|\boldsymbol{\alpha}\|_1 \leq s} \frac{\partial^{\boldsymbol{\alpha}} f(\frac{\boldsymbol{\eta}}{K})}{\boldsymbol{\alpha}!}(\boldsymbol{x} - \frac{\boldsymbol{\eta}}{K})^{\boldsymbol{\alpha}} \Big| \leq d^s \big\|\boldsymbol{x} - \frac{\boldsymbol{\eta}}{K}\big\|_2^\beta \leq d^{s+\beta/2} K^{-\beta}. \tag{D.61}$$

Motivated by this, we first construct a localization PQC $W_D(x)$ as in Lemma S14 such that

$$\mathbf{0} \leq f_{W_D}(\boldsymbol{x}) - \frac{\boldsymbol{\eta}}{\boldsymbol{K}} \leq \left(\frac{1}{2K}, \ldots, \frac{1}{2K}\right) \quad \text{if } \boldsymbol{x} \in Q_{\boldsymbol{\eta}}. \tag{D.62}$$

The depth of $W_D(x)$ is $O(\frac{1}{\Delta}\log K)$. We then construct a PQC

$$W_t(\boldsymbol{x}) := W_e(\boldsymbol{x}, f_{W_D}(\boldsymbol{x})), \tag{D.63}$$

where W_e is the PQC proposed in Lemma S17. Note that the state $|\boldsymbol{\eta}\rangle$ in Lemma S17 could be prepared by rounding $f_{W_D}(\boldsymbol{\eta})K$, i.e., $\boldsymbol{\eta} = \lfloor f_{W_D}(\boldsymbol{\eta})K \rfloor$. In other words, the PQC $W_t(\boldsymbol{x})$ has a nested structure consisting of a PQC for localization and a PQC for Taylor series implementation. Then we show that $f_{W_t}(\boldsymbol{x}) := \langle 0| W_t^\dagger(\boldsymbol{x}) Z^{(0)} W_t(\boldsymbol{x}) |0\rangle$ can approximate β-Hölder smooth function f on $\bigcup_{\boldsymbol{\eta}} Q_{\boldsymbol{\eta}}$. By the triangle inequality and Eq. (D.61), we have

$$|f(\boldsymbol{x}) - f_{W_t}(\boldsymbol{x})| \leq \Big| f_{W_t}(\boldsymbol{x}) - \sum_{\|\boldsymbol{\alpha}\|_1 \leq s} \frac{\partial^{\boldsymbol{\alpha}} f(f_{W_D}(\boldsymbol{x}))}{\boldsymbol{\alpha}!}(x - f_{W_D}(\boldsymbol{x}))^{\boldsymbol{\alpha}} \Big| + d^s \|\boldsymbol{x} - f_{W_D}(\boldsymbol{x})\|_2^\beta \tag{D.64}$$

$$\leq \Big| f_{W_t}(\boldsymbol{x}) - \sum_{\|\boldsymbol{\alpha}\|_1 \leq s} \frac{\partial^{\boldsymbol{\alpha}} f(f_{W_D}(\boldsymbol{x}))}{\boldsymbol{\alpha}!}(x - f_{W_D}(\boldsymbol{x}))^{\boldsymbol{\alpha}} \Big| + d^{s+\beta/2} K^{-\beta} \tag{D.65}$$

$$\leq d^{s+\beta/2} K^{-\beta}. \tag{D.66}$$

The second inequality comes from the fact that $||\boldsymbol{x} - f_{W_D}(\boldsymbol{x})||_2 \leq \frac{1}{K}$ for $\boldsymbol{x} \in Q_{\boldsymbol{\eta}}$. This completes the proof. □

Note that the PQC in Theorem 4 is nesting of two PQCs, while its depth is counted as the sum of two PQCs for simplicity. We have established the uniform convergence property of PQCs for approximating Hölder smooth function on $[0,1]^d$ except for the trifling region $\Lambda(d, K, \Delta)$. Note that the Lebesgue measure of such a trifling region is no more than $dK\Delta$. We can set $\Delta = K^{-d}$ with no influence on the size of the constructed PQC in Theorem 4. Since ν is absolutely continuous with respect to the Lebesgue measure, we have the following corollary.

Table S1: **Approximation errors of PQCs and ReLU FNNs**

Approach	**Target**	**Width**	**Depth**	**Number of parameters**	**Approximation error**
PQC	d-var. deg.-s monomial	$O(d)$	$O(s)$	$O(d+s)$	0
ReLU FNN [21]	d-var. deg.-s monomial	$O(N+s)$	$O(s^2M)$	$O((N^2+s^2)s^2M)$	$O(sN^{-sM})$
Nested PQC	$C_u^s([0,1]^d)$	$O(d\log K + s\log d)$	$O(K^d d^s)$	$O(K^d d^{s+1})$	$O(d^{2s}K^{-s})$
ReLU FNN[i] [21]	$C_u^s([0,1]^d)$	$O(s^{d+1}N)$	$O(s^2M)$	$O(s^{2d+4}K^{d/2}N)$	$O(s^d 8^s K^{-s})$

[i] Satisfying $NM = \Theta(K^{d/2})$.

Corollary S18. *Given a function $f \in \mathcal{H}^\beta([0,1]^d, 1)$ with $\beta = r+s$, $r \in (0,1]$ and $s \in \mathbb{N}^+$, for any $K \in \mathbb{N}$ and $\Delta \in (0, \frac{1}{3K})$, there exists a PQC $W_t(\boldsymbol{x})$ such that $f_{W_t}(\boldsymbol{x}) := \langle 0| W_t^\dagger(\boldsymbol{x}) Z^{(0)} W_t(\boldsymbol{x}) |0\rangle$ satisfies*

$$\|f(\boldsymbol{x}) - f_{W_t}(\boldsymbol{x})\|_{L^2(v)}^2 = \int_{[0,1]^d} (f(\boldsymbol{x}) - f_{W_t}(\boldsymbol{x}))^2 \nu(x)\,\mathrm{d}x \tag{D.67}$$

$$= \int_{\cup_{\boldsymbol{\eta}} Q_{\boldsymbol{\eta}} \bigcup \Lambda(d,K,\Delta)} (f(\boldsymbol{x}) - f_{W_t}(\boldsymbol{x}))^2 \nu(x)\,\mathrm{d}x \tag{D.68}$$

$$\leq (d^{s+\beta/2}K^{-\beta})^2 + 4dK^{1-d}. \tag{D.69}$$

The width of the PQC is $O(d\log K + \log s + s\log d)$, the depth is $O(s^2 d^s K^d(\log s + s\log d + d\log K)) + \frac{1}{\Delta}\log K)$, and the number of parameters is $O(sd^s(s+d+K^d) + \frac{d}{\Delta}\log K)$.

D.4 Comparison of "global" and "local" approaches in this work

We note that we have presented two distinct methodologies for constructing PQC models with UAP properties aimed at approximating continuous functions. In Theorem 3 and Theorem 4, we establish PQC models, guided by the multivariate Bernstein polynomials and the Taylor expansion of multivariate continuous functions, respectively. We categorize these approaches as "local" and "global". We proceed to conduct a comprehensive comparative analysis of these two strategies in the context of approximating Lipschitz continuous functions. For the subsequent analysis, we set $\beta = 1$, thus $s = 0$ in Theorem 4, in accordance with the Lipschitz continuous property exhibited by the target function.

The approximation error associated with the global approach can be bounded as $(2^d d\ell^2)/(n\varepsilon^2) + \varepsilon$. By selecting $n = (2^d d\ell^2)/\varepsilon^3$, we ensure an approximation error of 2ε. Concurrently, the corresponding number of trainable parameters scale as $O\big(2^{d^2} d^{d+1}\ell^{2d}/\varepsilon^{3d}\big)$. In contrast, the local approach exhibits an approximation error scaling as $\sqrt{d}K^{-1} + \varepsilon$. Setting $K = \sqrt{d}/\varepsilon$ ensures a 2ε approximation error, with the number of trainable parameters scaling as $O\left(d^{d/2}/\varepsilon^d\right)$. These findings highlight the advantage of the local approach for approximating continuous functions. More importantly, the approximation error proposed by the local method approaches the optimal convergence rate established in Shen et al. [22]. A formal comparison between PQCs and classical deep neural networks is stated in the next section.

E Comparison with related works in classical machine learning

In this subsection, we conduct a comparative exploration of PQCs and classical deep neural networks, focusing on critical aspects, including model size, the number of trainable parameters, and approximation error. To establish a meaningful benchmark, we turn our attention to deep feed-forward neural networks (FNNs) distinguished by the incorporation of rectified linear unit (ReLU) activation functions. FNNs represent the foundational class of neural networks, characterized by a unidirectional flow of information, commencing from the input layer and traversing through one or more hidden layers before culminating at the output layer. This architectural design ensures the absence of cyclic dependencies or loops among nodes within each layer. The ReLU activation function, mathematically defined as $\mathrm{ReLU}(x) := \max(x, 0)$, has gained prominence across diverse domains, including but not limited to image recognition [70, 71] and natural language processing [72, 73]. Its popularity in feed-forward networks stems from its efficacy in facilitating the convergence of function approximation during network training. Additionally, a recent study [74] has affirmed that classical neural networks employing commonly utilized activation functions can be effectively approximated by ReLU-activated networks while maintaining a mild increment in network size. Readers are also referred to some other excellent works related to ReLU networks [16, 75, 76].

In particular, Shen et al. [22] have proposed the optimal approximation error to approximate any Lipschitz function. Lu et al. [21] have provided a nearly optimal approximation error to approximate any smooth function using ReLU FNNs. For clarity, the comparison of our results with theirs is summarized in Table. S1. It is pertinent to observe that, in the majority of practical instances, the smoothness coefficient s of the target function tends to be modest since most functions to be approximated is not very smooth. Additionally, within practical scenarios, particularly in domains like image recognition and natural language processing, the dimensionality d of input data is substantially large. Consequently, within this context, we identify terms that solely rely on the variable s as constants and $d \gg s$ within Table S1.

We extend our investigation by quantifying the performance of PQCs and FNNs in terms of the model size and the number of parameters for approximating s-smooth functions $C_u^s([0,1]^d)$. Notably, we discover that in cases where the target function adheres to certain norms of smoothness, PQCs exhibit a notable improvement in approximating this function in terms of the model size and the number of parameters.

Model size. In particular, we explore the comparison of PQC and FNN model sizes when they yield the same approximation error ε (say some constant). Here, we use a straightforward measure, the product of width and depth, to gauge the model size. By setting approximation error as ε, the size of PQC and FNN scale as $O(K_Q^d d^{s+1})$ and $O(K_C^{d/2} s^{d+3})$, respectively, where $K_Q = \Theta(d^2/\varepsilon^{1/s})$ and $K_C = \Theta(s^{d/s}/\varepsilon^{1/s})$.

Remarkably, when $2 \leq s < d$, an intriguing observation emerges: the ratio of model sizes between PQCs and FNNs [21] exhibits a scaling behavior of $O(\varepsilon^{-d/(2s)}/s^{d^2 - d\log_s d})$. Our comprehensive analysis concludes that in situations where the smoothness threshold is satisfied, PQCs boast a significantly smaller model size compared to FNNs.

Number of trainable parameters. In the present investigation, we delve into the comparative analysis of the number of trainable parameters of PQC and FNN under the premise of yielding comparable approximation errors. From the perspective of approximation theory, the count of parameters serves as a standard metric for assessing model degrees of freedom and expressing model expressiveness. By setting approximation error as ε, the number of trainable parameters of PQC and FNN scale as $O(K_Q^d d^{s+1})$ and $O(K_C^{(1+\lambda_0)d/2} s^{2d+4})$, respectively. Here, the hyperparameter $\lambda_0 \in (0,1)$ signifies FNN's width.

Remarkably, through our analysis, we have uncovered that when $2 \leq s < d$, the relationship between the number of trainable parameters of PQCs and FNNs [21] demonstrates a scaling pattern characterized by $O(\varepsilon^{-(1-\lambda_0)d/(2s)}/s^{(1+\lambda_0)d^2 - d\log_s d})$. As a consequence, the number of trainable parameters of PQCs significantly reduces compared to that of FNNs.

Approximating monomial. Here, we conduct a comparative performance analysis of PQC and FNN in approximating monomial functions of degree s. Within this specialized target function space, PQCs exhibit distinct advantages in terms of width, depth, model size, and the number of trainable parameters. Notably, PQCs possess the unique capability to capture the dynamics of monomial functions precisely, eliminating the need for approximation and thereby offering a compelling advantage. These advantages position PQCs as promising candidates for outperforming FNNs when addressing more complex target function spaces.