Non-asymptotic Approximation Error Bounds of Parametrized Quantum Circuits

Zhan Yu, Qiuhao Chen, Yuling Jiao, Yinan Li, Xiliang Lu, Xin Wang, Jerry Zhijian Yang

Background and motivation: 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 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]. 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.

Our contributions: 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. 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 polynomials and multivariate smooth functions, the quantum circuit size and the number of trainable parameters of our proposed PQCs can be *exponentially smaller* than that of the deep ReLU neural networks as shown in [21], one of the most commonly used neural network family in classical deep learning theory. Thus, quantum machine learning models could potentially *admit improvements* over classical machine learning models; see Table I for a detailed comparison.

In the rest, we briefly present our PQC constructions, non-asymptotic approximation error bounds and comparisons with classical results; more details can be found in the technical version.

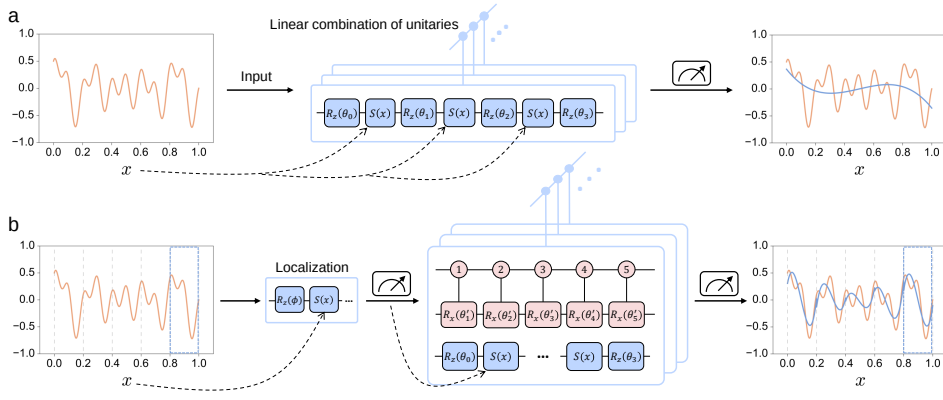

FIG. 1. **Overview of PQCs for approximating multivariate polynomials and smooth functions.** (a) Flowchart of the constructed PQC for multivariate polynomials. (b) Flowchart of the constructed PQC for smooth functions.

PQCs for multivariate polynomials. We start with constructing PQCs for implementing multivariate polynomials, which is at the core of multivariate function approximation theory. For univariate polynomials, Yu *et al.* [37] utilized ideas from quantum signal processing [44–46] to construct their block-encodings using single-qubit data re-uploading PQCs. Executing these PQCs in parallel leads to block-encodings of multivariable product polynomials (i.e. products of univariate polynomials). Since every multivariate polynomial can be written as a sum of product polynomials, we sum the block-encodings of each product polynomial using *linear combinations of unitaries* (LCU) [47]. This simple but novel strategy gives the following PQC implementations of multivariate polynomials.

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

$$\langle 0| W_p^\dagger(\boldsymbol{x}, \boldsymbol{\theta}^*) Z^{(0)} W_p(\boldsymbol{x}, \boldsymbol{\theta}^*) |0\rangle = p(\boldsymbol{x}) \tag{1}$$

holds for every $\boldsymbol{x}$, where $Z^{(0)}$ is the Pauli Z observable on the first qubit. The PQC $W_p(\boldsymbol{x}, \boldsymbol{\theta})$ acts on $O(d + \log s + s \log d)$ qubits, the circuit depth is $O(s^2 d^s (\log s + s \log d))$, and the number of trainable parameters is $\boldsymbol{\theta}$ is $O(s d^s (s + d))$.

In the proof, we explicitly construct the PQC $W_p(\boldsymbol{x}, \boldsymbol{\theta}^*)$ and prove the existence of the optimal parameter configuration $\boldsymbol{\theta}^*$ to achieve Eq. (1). Finding the optimal $\boldsymbol{\theta}^*$ might be a challenging task, but it is beyond the scope of this paper.

PQC approximation error bounds for smooth functions. We utilize the aforementioned PQCs for multivariate polynomials as building blocks to explicitly construct PQCs for approximating multivariate smooth functions and derive the corresponding approximation error bounds. We consider bounded smooth functions with d variables on $[0,1]^d$ with smooth index s, i.e. all k-th order partial derivatives of such functions exist and are continuous on $[0,1]^d$ for all $k = 0, \ldots, s$ and these partial derivatives are uniformly bounded by 1. We denote the space of such functions by $C_u^s([0,1]^d)$ (the unit ball of $C^s([0,1]^d)$). Lu *et al.* [21] proved near-optimal non-asymptotic approximation error bounds for functions in $C_u^s([0,1]^d)$ using feed-forward neural networks with rectified linear unit activation functions (ReLU FNN), which is one of the most commonly used neural network structure in classical deep learning theory.

TABLE I. **Approximation errors of PQCs and ReLU FNNs**

Approach	Target	Width	Depth	Number of parameters	Approximation error
PQC	d-var. s-deg. monomial	$O(d)$	$O(s)$	$O(d+s)$	0
ReLU FNN [21]	d-var. s-deg. 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 $NM = \Theta(K^{d/2})$.

We utilize the Taylor expansion to approximate functions $f \in C_u^s([0,1]^d)$ [18]. For any $\boldsymbol{x}, \boldsymbol{x_0} \in [0,1]^d$,

$$\left| f(\boldsymbol{x}) - \sum_{\|\boldsymbol{\alpha}\|_1 \le s} \frac{\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x_0})}{\boldsymbol{\alpha}!} (\boldsymbol{x}-\boldsymbol{x_0})^{\boldsymbol{\alpha}} \right| \le d^s \|\boldsymbol{x}-\boldsymbol{x_0}\|_2^{s+1} . \tag{2}$$

As we can see, the Taylor expansion only converges to f in small local regions. To achieve a satisfactory approximation performance, The classical method is to discretize $[0,1]^d$ into K local regions and implement the Taylor expansion in each region. We construct a PQC to implement such a strategy and combine Eq. (2) to derive the PQC approximation error bound for smooth functions.

Theorem 2. *Let $f \in C_u^s([0,1]^d)$. For any $K \in \mathbb{N}$, there exists a PQC $W_K(\boldsymbol{x}, \boldsymbol{\theta})$ and a parameter configuration $\boldsymbol{\theta}^*$ such that*

$$|f(\boldsymbol{x}) - \langle 0| W_K^\dagger(\boldsymbol{x}, \boldsymbol{\theta}^*) Z^{(0)} W_K(\boldsymbol{x}, \boldsymbol{\theta}^*) |0\rangle| \le \left(\frac{d^2}{K}\right)^s \tag{3}$$

for almost all $\boldsymbol{x} \in [0,1]^d$. The PQC $W_K(\boldsymbol{x}, \boldsymbol{\theta})$ acts on $O(d\log K + s\log d)$ qubits, the circuit depth is $O(K^d d^s)$, and the number of parameters is $O(K^d d^{s+1})$.

In the proof, we explicitly construct the PQC $W_K(\boldsymbol{x}, \boldsymbol{\theta}^*)$ and prove the existence of the optimal parameter configuration $\boldsymbol{\theta}^*$ to achieve Eq. (3). The constructed PQCs for approximating smooth functions are in a nested structure, consisting of a PQC for discretizing $[0,1]^d$ into K^d hypercubes and another PQC for the Taylor expansions. The discretization introduces a trifling region (of small Lebesgue measure) where the function value cannot be uniformly bounded. This also happens in classical neural network approximations and can be overcome by more complicated analysis.

To the best of our knowledge, this is *the first non-asymptotic* PQC approximation error bounds for smooth functions. Moreover, Such a PQC approximation error bound allows us to compare the necessary PQC size and number of parameters for achieving the desired accuracy with that of classical neural networks.

Comparison with classical neural networks. The non-asymptotic approximation error bounds for multivariate polynomials and smooth functions enable us to compare the size and the number of parameters required to achieve the same approximation error by PQCs and ReLU FNNs, as summarized in Table I.

Constructing ReLU FNN for approximating multivariate monomials is one of the key contributions in [21], while PQCs possess the unique capability to capture the dynamics of polynomials precisely. The approximation power of PQCs for multivariate polynomials comes from the exact representation of univariate polynomials using single-qubit PQCs [37]. This feature enables PQCs to achieve better approximate rates for smooth functions.

For multivariate smooth functions in $C_u^2([0,1]^d)$, 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 2 and set $K_C = \Theta(2^{d/2}/\sqrt{\varepsilon})$ for the constructed near-optimal ReLU FNNs from [21]. Substituting the choices of K's in the sizes of PQCs and ReLU FNNs, we have

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

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 $2 \le s < d$. 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, indicating that the expressivity of PQCs is exponentially more powerful than the expressivity of ReLU FNNs for approximating general smooth functions.

[1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature **549**, 195 (2017).

[2] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Parameterized quantum circuits as machine learning models, Quantum Science and Technology **4**, 043001 (2019).

[3] J. Preskill, Quantum Computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[4] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature **549**, 242 (2017).

[5] M. Cerezo, K. Sharma, A. Arrasmith, and P. J. Coles, Variational quantum state eigensolver, npj Quantum Information **8**, 113 (2022).

[6] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, Quantum chemistry in the age of quantum computing, Chemical Reviews **119**, 10856 (2019).

[7] X. Pan, Z. Lu, W. Wang, Z. Hua, Y. Xu, W. Li, W. Cai, X. Li, H. Wang, Y.-P. Song, C.-L. Zou, D.-L. Deng, and L. Sun, Deep quantum neural networks on a superconducting processor, Nature Communications **14**, 4006 (2023).

[8] W. Ren, W. Li, S. Xu, K. Wang, W. Jiang, F. Jin, X. Zhu, J. Chen, Z. Song, P. Zhang, H. Dong, X. Zhang, J. Deng, Y. Gao, C. Zhang, Y. Wu, B. Zhang, Q. Guo, H. Li, Z. Wang, J. Biamonte, C. Song, D.-L. Deng, and H. Wang, Experimental quantum adversarial learning with programmable superconducting qubits, Nature Computational Science **2**, 711 (2022).

[9] H.-L. Huang, Y. Du, M. Gong, Y. Zhao, Y. Wu, C. Wang, S. Li, F. Liang, J. Lin, Y. Xu, R. Yang, T. Liu, M.-H. Hsieh, H. Deng, H. Rong, C.-Z. Peng, C.-Y. Lu, Y.-A. Chen, D. Tao, X. Zhu, and J.-W. Pan, Experimental quantum generative adversarial networks for image generation, Phys. Rev. Appl. **16**, 024051 (2021).

[10] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum Circuit Learning, Physical Review A **98**, 032309 (2018), arxiv:1803.00745.

[11] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, Data re-uploading for a universal quantum classifier, Quantum **4**, 226 (2020).

[12] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature **521**, 436 (2015).

[13] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of control, signals and systems **2**, 303 (1989).

[14] K. Hornik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, Neural Networks **2**, 359 (1989).

[15] A. Barron, Universal approximation bounds for superpositions of a sigmoidal function, IEEE Transactions on Information Theory **39**, 930 (1993).

[16] D. Yarotsky, Error bounds for approximations with deep ReLU networks, Neural Networks **94**, 103 (2017).

[17] D. Yarotsky, Optimal approximation of continuous functions by very deep ReLU networks, in *Proceedings of the 31st Conference On Learning Theory* (PMLR, 2018) pp. 639–649.

[18] P. Petersen and F. Voigtlaender, Optimal approximation of piecewise smooth functions using deep ReLU neural networks, Neural Networks **108**, 296 (2018).

[19] D. Yarotsky and A. Zhevnerchuk, The phase diagram of approximation rates for deep neural networks, in *Advances in Neural Information Processing Systems*, Vol. 33 (Curran Associates, Inc., 2020) pp. 13005–13015.

[20] Z. Shen, Deep Network Approximation Characterized by Number of Neurons, Communications in Computational Physics **28**, 1768 (2020).

[21] J. Lu, Z. Shen, H. Yang, and S. Zhang, Deep Network Approximation for Smooth Functions, SIAM Journal on Mathematical Analysis **53**, 5465 (2021).

[22] Z. Shen, H. Yang, and S. Zhang, Optimal approximation rate of ReLU networks in terms of width and depth, Journal de Mathématiques Pures et Appliquées **157**, 101 (2022).

[23] E. Weinan, C. Ma, and L. Wu, The barron space and the flow-induced function spaces for neural network models, Constructive Approximation **55**, 369 (2022).

[24] Y. Jiao, Y. Lai, X. Lu, F. Wang, J. Z. Yang, and Y. Yang, Deep neural networks with ReLU-sine-exponential activations break curse of dimensionality in approximation on hölder class, SIAM Journal on Mathematical Analysis **55**, 3635 (2023), https://doi.org/10.1137/21M144431X.

[25] Y. Jiao, G. Shen, Y. Lin, and J. Huang, Deep nonparametric regression on approximate manifolds: Nonasymptotic error bounds with polynomial prefactors, The Annals of Statistics **51**, 691 (2023).

[26] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature **567**, 209 (2019).

[27] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, Expressive power of parametrized quantum circuits, Physical Review Research **2**, 033125 (2020).

[28] Y. Liu, S. Arunachalam, and K. Temme, A rigorous and robust quantum speed-up in supervised machine learning,

Nature Physics **17**, 1013 (2021).
[29] H.-Y. Huang, R. Kueng, and J. Preskill, Information-theoretic bounds on quantum advantage in machine learning, Phys. Rev. Lett. **126**, 190505 (2021).
[30] S. Jerbi, C. Gyurik, S. Marshall, H. Briegel, and V. Dunjko, Parametrized Quantum Policies for Reinforcement Learning, in *Advances in Neural Information Processing Systems*, Vol. 34 (Curran Associates, Inc., 2021) pp. 28362–28375.
[31] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, Power of data in quantum machine learning, Nature Communications **12**, 2631 (2021).
[32] S. Jerbi, L. J. Fiderer, H. Poulsen Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, Quantum machine learning beyond kernel methods, Nature Communications **14**, 517 (2023).
[33] J. Jäger and R. V. Krems, Universal expressiveness of variational quantum classifiers and quantum kernels for support vector machines, Nature Communications **14**, 576 (2023).
[34] M. Schuld, R. Sweke, and J. J. Meyer, Effect of data encoding on the expressive power of variational quantum-machine-learning models, Physical Review A **103**, 032430 (2021).
[35] F. J. Gil Vidal and D. O. Theis, Input Redundancy for Parameterized Quantum Circuits, Frontiers in Physics **8** (2020).
[36] A. Pérez-Salinas, D. López-Núñez, A. García-Sáez, P. Forn-Díaz, and J. I. Latorre, One qubit as a universal approximant, Physical Review A **104**, 012405 (2021).
[37] Z. Yu, H. Yao, M. Li, and X. Wang, Power and limitations of single-qubit native quantum neural networks, in *Advances in Neural Information Processing Systems*, Vol. 35, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Curran Associates, Inc., 2022) pp. 27810–27823.
[38] A. Manzano, D. Dechant, J. Tura, and V. Dunjko, Parametrized Quantum Circuits and their approximation capacities in the context of quantum machine learning (2023), arxiv:2307.14792 [quant-ph].
[39] T. Goto, Q. H. Tran, and K. Nakajima, Universal Approximation Property of Quantum Machine Learning Models in Quantum-Enhanced Feature Spaces, Physical Review Letters **127**, 090506 (2021).
[40] L. Gonon and A. Jacquier, Universal Approximation Theorem and error bounds for quantum neural networks and quantum reservoirs (2023), arxiv:2307.12904 [quant-ph].
[41] J. Qi, C.-H. H. Yang, P.-Y. Chen, and M.-H. Hsieh, Theoretical error performance analysis for variational quantum circuit based functional regression, npj Quantum Information **9**, 4 (2023).
[42] H. Zhao, L. Lewis, I. Kannan, Y. Quek, H.-Y. Huang, and M. C. Caro, Learning quantum states and unitaries of bounded gate complexity (2023), arXiv:2310.19882 [quant-ph].
[43] P. Grohs and G. Kutyniok, *Mathematical aspects of deep learning* (Cambridge University Press, 2022).
[44] G. H. Low, T. J. Yoder, and I. L. Chuang, Methodology of Resonant Equiangular Composite Quantum Gates, Physical Review X **6**, 041067 (2016).
[45] G. H. Low and I. L. Chuang, Optimal Hamiltonian Simulation by Quantum Signal Processing, Physical Review Letters **118**, 010501 (2017).
[46] G. H. Low, *Quantum Signal Processing by Single-Qubit Dynamics*, Thesis, Massachusetts Institute of Technology (2017).
[47] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, Quantum Inf. Comput. **12**, 901 (2012).