Dequantizing quantum machine learning models using tensor networks

Seongwook Shin, Yong Siah Teo,* and Hyunseok Jeong†
Department of Physics and Astronomy, Seoul National University, 08826 Seoul, South Korea

(Received 30 July 2023; accepted 18 April 2024; published 29 May 2024)

Ascertaining whether a classical model can efficiently replace a given quantum model—*dequantization*—is crucial in assessing the true potential of quantum algorithms. In this work, we introduced the dequantizability of the function class of variational quantum-machine-learning (VQML) models by employing the tensor network formalism, effectively identifying every VQML model as a subclass of matrix product state (MPS) model characterized by constrained coefficient MPS and tensor product-based feature maps. From this formalism, we identify the conditions for which a VQML model's function class is dequantizable or not. Furthermore, we introduce an efficient quantum kernel-induced classical kernel which is as expressive as given any quantum kernel, hinting at a possible way to dequantize quantum kernel methods. This presents a thorough analysis of VQML models and demonstrates the versatility of our tensor-network formalism to properly distinguish VQML models according to their genuine quantum characteristics, thereby unifying classical and quantum machine-learning models within a single framework.

DOI: 10.1103/PhysRevResearch.6.023218

I. INTRODUCTION

Quantum machine learning (QML) garners a huge interest among various communities and industries in recent years as a prominent candidate for practical applications on quantum devices [1,2]. Variational QML (VQML) uses a variational quantum circuit as a data processor, and the variational parameters in the quantum circuit are optimized with the help of classical optimization algorithms in order to learn and predict data outputs. VQML aims to achieve a more powerful ML model by exploiting a possible quantum advantage of quantum circuits in noisy intermediate scale quantum (NISQ) era.

While there exist theoretical proofs that demonstrate the possibility of achieving a quantum advantage in ML tasks in fully quantum settings [3,4], more effort is required to understanding whether ML from classical data can also achieve such a quantum advantage [5–9].

For this purpose, a fair assessment of VQML and classical ML models is in order, both of which possess inherently different structures. Moreover, the preprocessing of classical data always precedes VQML when they are encoded on NISQ machines. This additional computation might lead to the "dequantization" argument when comparing a classical and quantum model [10,11]. Moreover, if one does not have access to a coherent quantum memory and quantum channel, then even if the QML uses a "quantum state" as its input, one cannot avoid using classical data to "upload" the quantum state onto the quantum circuit. In this study, we propose a unified tensor-network (TN) formalism to systematically analyze VQML models, which permits us to classify all classical-data-encoded VQML models into a subclass of matrix product state (MPS) ML models [12]. We introduce the concept of dequantization of the function class of VQML models—the efficient approximation of all function-class outputs of a VQML model using a classical model—and find necessary conditions for (non)dequantizable VQML models by classical MPS models.

More specifically, the TN formalism describes the function output of a VQML model as a *linear* MPS model form, subsequently separating it into two components: the coefficient part of the linear model which is in the form of MPS containing all quantum-circuit training parameters and the basis part (or a feature map in the ML lingo), which formulates the basis for the linear model. The number of linearly independent basis functions can scale exponentially with the number of encoding gates [13,14], challenging classical models to approximate them. However, by leveraging the knowledge of data preprocessing before implementing VQML, we can simply observe that the basis part is in an easily manageable tensor-product form.

Representing coefficients as an MPS allows systematic analysis of expressivity and approximability of models in the context of entanglement. Moreover, we discover that the coefficients of a VQML model are Pauli coefficients of the circuit-dependent operator when expanded in the Pauli basis. This observation enables systematic analysis of coefficients of VQML models using various techniques, of which we shall provide some hints.

To assess whether a VQML model is dequantizable or not, we construct a classical MPS model having the same basis function as (or is *basis-equivalent* to) the VQML model and explore the possibility of VQML function-class

*yong.siah.teo@gmail.com
†h.jeong37@gmail.com

dequantization. With dimensional arguments and borrowing key results concerning MPS approximability [15], we list some necessary conditions of nondequantizable VQML models. These include models with dimensions that scale exponentially with the number of qubits, and coefficient MPSs that are highly entangled. Numerically we show that a general polydepth variational quantum circuit with a nontrivial encoding strategy can satisfy this requirement.

Lastly, we introduce the computationally efficient classical kernel inspired by the basis-equivalent classical MPS model that is naturally attained by using the equivalent precomputations as the given quantum kernel. It covers the function space from the quantum kernel, and we compare the performance of the classically hard-to-simulate quantum kernel and the classical counterpart of it.

After a preliminary outline of the theoretical background and setup of VQML models in Sec. II, the unifying tensor-network formalism for describing these quantum models is introduced in Sec. III, followed by a more detailed discussion of VQML dequantization in Sec. IV. In Sec. V, upon recognizing that the feature map is, in fact, efficient to handle classically, we analytically and numerically study basis-equivalent classical MPS models and identify conditions for (non)dequantizable VQML models. Then, using our TN formalism, in Sec. VI, we construct tensor-product classical kernel models and show that they can efficiently cover the quantum kernel counterparts. This work shall finally conclude in Sec. VII.

II. PRELIMINARIES OF VARIATIONAL QUANTUM MACHINE LEARNING MODEL

Machine learning (ML) can be understood as a function approximation task, where the target function is unknown and is to be learned from a training dataset. A function approximator in ML is a computational model, which defines and generates some function class. The cost function measures how well our function-approximator model is approximating the target function. An ML algorithm minimizes this cost function calculated with the training dataset and ML model function, by using various numerical or analytical methods.

Variational quantum machine learning (VQML) is ML that uses a parametrized quantum circuit as a computational model. The parametrized quantum circuit usually has a fixed structure (called an *ansatz*) and is parametrized by variable parameters. Because VQML models employ quantum circuits, they take quantum state as input, which inevitably requires a classical-to-quantum encoding procedure [16]. There exist various encoding strategies, such as amplitude encoding, Pauli encoding, data reuploading [17], instantaneous quantum polynomial (IQP) encoding that is conjectured to be hard to simulate classically [18], and so on. These encoding strategies, $\mathcal{E}$, consists of preprocessing functions $\Phi^{(i)}:\mathbb{R}^d\to\mathbb{R}^{2^{2m_i}-1}$, m_i-qubit encoding gates $S^{(i)}(\cdot):\mathbb{R}^{2^{2m_i}-1}\to\mathbb{C}^{2^{m_i}}$ that map preprocessed data $\Phi^{(i)}(\boldsymbol{x})$ into an m_i-qubit state, and positions of encoding gates within the quantum circuit. Here we assumed that the inputs are d-dimensional real vectors without loss of generality and i is the index for distinguished encoding gates. For a general $\mathcal{E}$, the corresponding encoding gates $S^{(i)}$s can be highly nonlocal.

To the output of the VQML model, we choose some observable (or POVM) O and measure its expectation value. Then, the function class of the VQML model is defined as

$$f_Q(\boldsymbol{x};\mathcal{E},U,\boldsymbol{\theta},O)=\langle\mathbf{0}|U^\dagger(\boldsymbol{x};\mathcal{E},\boldsymbol{\theta})OU(\boldsymbol{x};\mathcal{E},\boldsymbol{\theta})|\mathbf{0}\rangle. \tag{1}$$

Here, we initiate the n qubits to the state $|\mathbf{0}\rangle\equiv|0\rangle^{\otimes n}$, and $U(\boldsymbol{x};\mathcal{E},\boldsymbol{\theta})$ represents the quantum circuit using encoding strategy $\mathcal{E}$ and trainable unitaries which are parametrized by $\boldsymbol{\theta}$.

Let us consider the general $\mathcal{E}$, where $S^{(i)}$ are multiqubit gates. To implement any m_i-qubit encoding gate on a real quantum circuit, we need to compile it using a universal gate set of the quantum device. Here we assume a universal gate set comprising an arbitrary single-qubit gate and some unparametrized two-qubit gate such as a controlled-not (CNOT) gate. As arbitrary universal gate sets can be converted to this single- and CNOT gate set, without loss of generality, every multi-qubit encoding gate is decomposed with a set $\{S_1^\alpha(\phi_1^\alpha(\boldsymbol{x}),\phi_2^\alpha(\boldsymbol{x}),\phi_3^\alpha(\boldsymbol{x})),U_{\mathrm{CNOT}}\}_\alpha$ before running VQML algorithm, where $(\phi_1^\alpha(\boldsymbol{x}),\phi_2^\alpha(\boldsymbol{x}),\phi_3^\alpha(\boldsymbol{x}))$ are Z-Y-Z Euler angles. The Y rotation in the middle can be replaced by Z rotation with diagonalization such as

$$e^{-i\frac{\phi_2^\alpha(\boldsymbol{x})}{2}Y}=F^\dagger e^{-i\frac{\phi_2^\alpha(\boldsymbol{x})}{2}Z}F, \tag{2}$$

where $F=\frac{1}{\sqrt{2}}\left(\begin{smallmatrix}1 & 1\\ i & -i\end{smallmatrix}\right)$. Euler angles are obtained when compiling multi-qubit encoding gates into the single-qubit gates, and calculated by preprocessing functions that are denoted as ϕs. We denote N as the total number of single-qubit Pauli-Z rotation gates when all the encoding gates are compiled. For simplicity, we combine upper and lower indices in ϕ_k^α into one index $\alpha\in[N]$, and group all preprocessing functions $\phi_\alpha(\mathcal{E}):\mathbb{R}^d\to\mathbb{R}$ and the values $\{\phi_\alpha(\boldsymbol{x};\mathcal{E})\}_{\alpha=1}^N$ [see Fig. 1(a)].

III. THE FUNCTION CLASS OF VQML MODELS

After the Pauli-gate decomposition, all data-dependent encoding gates are expressed in terms of Pauli-Z rotations. Using the result from Ref. [19], it is straightforward to see that the function class of a general encoding strategy corresponds to a linear combination of basis functions $\{B_j(\boldsymbol{x};\mathcal{E})\}_j$,

$$\begin{aligned}f_Q(\boldsymbol{x};\boldsymbol{\theta},\mathcal{E},U,O)&=\sum_{j=1}^K c_j(\boldsymbol{\theta},U,O)e^{-ib_j(\boldsymbol{x};\mathcal{E})}\\&\equiv\sum_{j=1}^K c_j(\boldsymbol{\theta},U,O)B_j(\boldsymbol{x};\mathcal{E})\\&\equiv\boldsymbol{c}(\boldsymbol{\theta},U,O)\cdot\mathbf{B}(\boldsymbol{x};\mathcal{E}),\end{aligned} \tag{3}$$

where

$$b_j(\boldsymbol{x};\mathcal{E})\in\left\{\sum_{\alpha=1}^N\beta_\alpha\phi_\alpha(\boldsymbol{x};\mathcal{E})\,|\,\beta_\alpha=\{-1,0,1\}\right\}. \tag{4}$$

The symbol K refers to the number of linearly independent basis functions which can be less than 3^N. In other words, any VQML model is a featured linear model (FLM) which is a linear model in the feature space endowed by a feature map $\mathbf{B}:\mathbb{R}^d\to\mathbb{C}^K$ [18,20]. Note that the coefficients $c_j(\boldsymbol{\theta},U,O)$s from the quantum model are not arbitrary, but constrained as

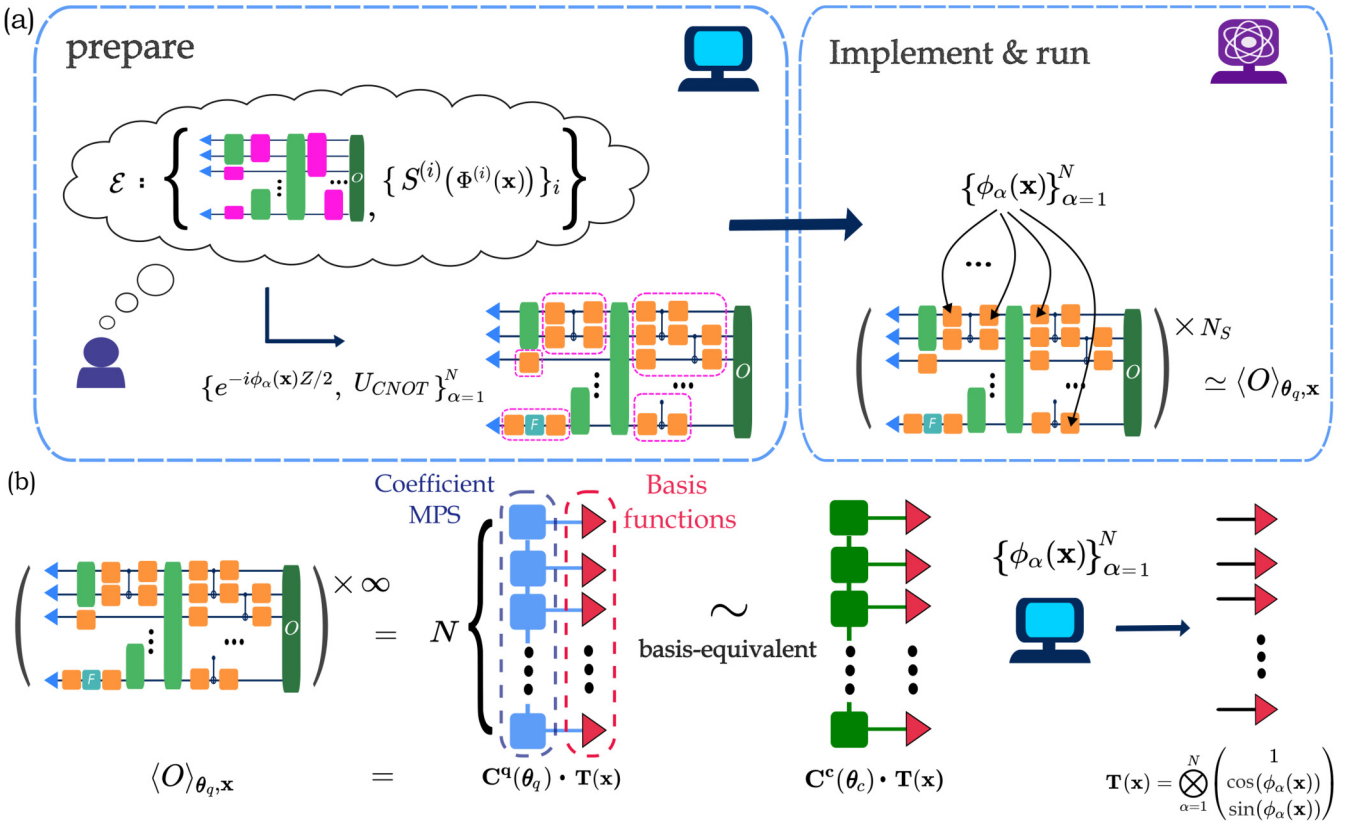

FIG. 1. Schematic overview of this work. (a) The procedure for variational quantum machine learning (VQML) with a general encoding strategy, denoted as $\mathcal{E}$. This strategy includes encoding gates $S^{(i)}(\cdot)$s, preprocessing functions $\Phi^{(i)}(\boldsymbol{x})$, and their respective positions within the quantum circuit (represented by magenta boxes). During the preparation stage, all original encoding gates are compiled into single-qubit Pauli-Z rotations (represented by orange boxes), with angles $\phi_\alpha(\boldsymbol{x})$s and nonparametrized two-qubit gates, all computed classically. This decomposition may include nonparametrized unitaries, denoted as F. Green boxes represent the trainable circuit with variable parameters $\boldsymbol{\theta}_q$. The VQML-model output is given by the expectation value of a specific observable O. This output is estimated *via* N_S runs of the quantum circuit. (b) The exact value of the VQML model can be represented as a linear model using the feature map $\mathbf{T}$ and constrained MPS coefficient tensor $\mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta}_q)$, constructed from the parametrized circuit. With the preprocessing functions obtained during the preparation stage, we can efficiently construct $\mathbf{T}$, classically. The classical tensor network (TN) model using $\mathbf{T}$ resides in the same function space spanned by the same basis functions as the VQML model. The TN formalism can then be used to compare the respective MPS parts $\mathbf{C}^{\mathbf{c}}(\boldsymbol{\theta}_c)$ and $\mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta}_q)$ for the classical and quantum models, which dictates the possibility for dequantization.

they are obtained from a quantum circuit. Calculating the exact form or values of $c_j(\boldsymbol{\theta}, U, O)$s is equivalent to simulating a quantum circuit directly, which is typically inefficient unless the circuits possess special structures [21]. Rather, we shall analyze VQML models using a unifying TN framework, to be introduced in the following section.

A. VQML models as matrix product state models

MPS model is a variational ML model which is a featured linear model. A feature map is given by a tensor-product of certain data-dependent vectors and a coefficient part is given by a variational MPS. The MPS model was originally introduced in Ref. [12] as a quantum-inspired classical model. However, here we assert a somewhat "reverse" statement that a VQML model using classical data is a subclass of the MPS model.

Throughout this text, we assume all encoding gates are transformed to single-qubit Pauli-Z rotations. Additionally, the set of preprocessing functions $\{\phi_\alpha\}_{\alpha=1}^N$ are defined as in Sec II. We omit the encoding strategy $\mathcal{E}$ dependency for notational simplicity.

First, let us consider the simple parallel VQML model with $n = N$ where all the encoding gates are placed parallel and in between trainable unitaries $W_1(\boldsymbol{\theta}_1)$ and $W_2(\boldsymbol{\theta}_2)$:

$$f_Q(\boldsymbol{x};\boldsymbol{\theta}, W_1, W_2, O) = \langle 0|W_1^\dagger(\boldsymbol{\theta}_1)\mathbf{S}^\dagger(\boldsymbol{x})W_2^\dagger(\boldsymbol{\theta}_2)O \times W_2(\boldsymbol{\theta}_2)\mathbf{S}(\boldsymbol{x})W_1(\boldsymbol{\theta}_1)|0\rangle, \quad (5)$$

where $\mathbf{S}(\boldsymbol{x}) = \Pi_{\alpha=1}^N e^{-i\phi_\alpha(\boldsymbol{x})Z_\alpha/2}$, and $\boldsymbol{\theta} \equiv (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$. Then the following lemma holds, where detailed proof with graphical description is given in Appendix A.

Lemma 1 (A simple parallel VQML model is an MPS model). Given a simple parallel VQML model as Eq. (5), one can represent it as:

$$f_Q(\boldsymbol{x};\boldsymbol{\theta}, W_1, W_2, O) = \mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta}) \cdot \mathbf{T}(\boldsymbol{x}), \quad (6)$$

with the *coefficient MPS*

$$\mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta}) = (O' \odot \rho^T)(\boldsymbol{\theta}) \cdot \tilde{\mathbf{P}}, \quad (7)$$

and

$$\tilde{\mathbf{P}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & i \\ 0 & 1 & -i \\ 1 & 0 & 0 \end{pmatrix}^{\otimes N}. \tag{8}$$

The *feature map is given as*

$$\mathbf{T}(\boldsymbol{x}) = \bigotimes_{\alpha=1}^{N} \mathbf{T}^{(\alpha)}(\boldsymbol{x}) = \bigotimes_{\alpha=1}^{N} \begin{pmatrix} 1 \\ \cos(\phi_\alpha(\boldsymbol{x})) \\ \sin(\phi_\alpha(\boldsymbol{x})) \end{pmatrix}. \tag{9}$$

We denoted the evolved observable as $O'(\boldsymbol{\theta}_2) := W_2^\dagger(\boldsymbol{\theta}_2) O W_2(\boldsymbol{\theta}_2)$, pre-encoded state as $\rho(\boldsymbol{\theta}_1) := W_1(\boldsymbol{\theta}_1)|0\rangle\langle 0|W_1^\dagger(\boldsymbol{\theta}_1)$, tensor contraction as $\cdot$, and the Hadamard product as $\odot$. The tensor $(O' \odot \rho^T)(\boldsymbol{\theta})$ is a $2^N \times 2^N$ matrix having $2N$ indices, where row and column indices are decomposed into N indices each for the one-qubit line. See Fig. 9(a). We *vectorize* this $(O' \odot \rho^T)(\boldsymbol{\theta})$ by gathering the same site indices to make it a tensor of N indices having a dimension of 4. This enables contraction between tensor network $\tilde{\mathbf{P}}$ and $(O' \odot \rho^T)$.

Next, we consider the general case of a VQML model using n qubits as given in Eq. (1). We can rewrite any general encoded VQML as an R-times reuploading model,

$$\begin{aligned} f_Q(\boldsymbol{x};\boldsymbol{\theta}) = \langle \mathbf{0}| & W_0^\dagger(\boldsymbol{\theta}_0)\mathcal{S}_1^\dagger(\boldsymbol{x}) W_1^\dagger(\boldsymbol{\theta}_1)\mathcal{S}_2^\dagger(\boldsymbol{x})\cdots \\ & S_R^\dagger(\boldsymbol{x}) W_R^\dagger(\boldsymbol{\theta}_R) O W_R(\boldsymbol{\theta}_R)\mathcal{S}_R(\boldsymbol{x})\cdots \\ & \mathcal{S}_2(\boldsymbol{x}) W_1(\theta_1)\mathcal{S}_1(\boldsymbol{x}) W_0(\boldsymbol{\theta}_0)|\mathbf{0}\rangle. \end{aligned} \tag{10}$$

This equation distinguishes layers by their dependencies on data or variational parameters, aligning with the concept of a data reuploading model as noted in Ref. [17]. Recall that all encoding gates are decomposed to single-qubit Pauli-Z gates and the preprocessing functions $\{\phi_\alpha\}_\alpha$ are given. Equation (10) includes cases where some $\mathcal{S}_k$ contains the trivial preprocessing function $\phi(\boldsymbol{x}) = 0$. Leveraging lemma 1, we can now establish a Theorem regarding general VQML models.

Theorem 1 (Any VQML model can be represented as an MPS model). Any VQML model employing a general encoding strategy $\mathcal{E}$, and a circuit ansatz U such as Eq. (1) can be represented as an MPS model:

$$f_Q(\boldsymbol{x}; \mathcal{E}, U, \boldsymbol{\theta}, O) = \mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta}) \cdot \mathbf{T}(\boldsymbol{x}), \tag{11}$$

where

$$\mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta}) = \left(O'_R \odot \rho_R^T\right)(\boldsymbol{\theta}) \cdot \tilde{\mathbf{P}} \tag{12}$$

and

$$\mathbf{T}(\boldsymbol{x}) = \bigotimes_{\alpha=1}^{nR} \mathbf{T}^{(\alpha)}(\boldsymbol{x}) = \bigotimes_{\alpha=1}^{nR} \begin{pmatrix} 1 \\ \cos(\phi_\alpha(\boldsymbol{x})) \\ \sin(\phi_\alpha(\boldsymbol{x})) \end{pmatrix}. \tag{13}$$

In this formulation, the circuit ansatz U, O and $\boldsymbol{\theta}$ dependent tensors O'_R and ρ_R are as specified in Eqs. (A16) and (A17), respectively.

The proof of Theorem 1 with graphical description can be found in Appendix A.

The function class of a VQML model is a linear model in the feature space, with the feature map $\mathbf{T}$. This can also be viewed as a special MPS model, characterized by a special coefficient MPS $\mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta})$, which is determined by the quantum circuit's structure. In the following sections, we delve deeper into the analysis of both $\mathbf{T}$ and $\mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta})$. This exploration aims to reveal the insights and implications that such an analysis can provide.

B. The feature map T

The feature map $\mathbf{T} : \mathbb{R}^d \to \mathbb{R}^{3^N}$ (where N is now the length of the coefficient MPS that depends on the structure of the VQML model) is a mapping given by

$$\mathbf{T}(\boldsymbol{x}) = \bigotimes_{\alpha=1}^{N} \mathbf{T}^{(\alpha)}(\boldsymbol{x}) = \bigotimes_{\alpha=1}^{N} \begin{pmatrix} 1 \\ \cos(\phi_\alpha(\boldsymbol{x})) \\ \sin(\phi_\alpha(\boldsymbol{x})) \end{pmatrix}. \tag{14}$$

The encoding strategy $\mathcal{E}$ determines preprocessing functions $\{\phi_\alpha\}_\alpha$, consequently defining the feature map. Output functions of the VQML model are given by the linear sum of basis functions in $\{\mathbf{T}_i(\boldsymbol{x})\}_{i=1}^{3^N}$, which is the set of components of the feature map $\mathbf{T}(\boldsymbol{x})$. This means the function space, or the function class, of the VQML model, is spanned by these 3^N functions. It is critical to recognize that these basis functions may not all be linearly independent, as their independence hinges on the selected preprocessing functions and, therefore, $\mathcal{E}$.

As an example of a 1D VQML function, in a naive Pauli encoding strategy where all $\phi_\alpha(x) = x$, only $2N + 1$ out of 3^N components are linearly independent. On the other hand, for the exponential encoding strategy [13], where $\phi_\alpha(x) = k^{\alpha-1}x$, a set of 3^N linearly independent basis functions can be generated with $k \geqslant 3$.

In terms of computational complexity, $\mathbf{T}(\boldsymbol{x})$ is efficient to store and generate classically, as it requires only $O(N)$ memory to store and single call of ϕ_αs as the VQML model does. With this observation, we conclude that exponentially large feature space, commonly dubbed as a special feature of QMLs, is not unique to quantum models.

C. The coefficient MPS $\mathbf{C}^{\mathbf{q}}$

The coefficient MPS $\mathbf{C}^{\mathbf{q}} : \boldsymbol{\Theta} \times \mathbb{O} \to \mathbb{R}^{3^N}$ (For the general encoding case, O'_R and ρ_R respectively.) is a mapping from parameter space $\boldsymbol{\Theta}$ and observable space $\mathbb{O}$ to 3^N-dimensional real space. This *coefficient MPS* of given VQML, calculated as

$$\mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta}) = (O' \odot \rho^T)(\boldsymbol{\theta}) \cdot \tilde{\mathbf{P}}, \tag{15}$$

can be seen as a normal vector of hyperplane on the feature space.

Unlike coefficients in a simple linear model, we cannot control all 3^N components in $\mathbf{C}^{\mathbf{q}}$ freely, as they are obtained implicitly by the contraction of unitaries in the quantum circuit. In general, it is not *universal*, which means that not all of 3^N dimensional vectors can be generated. For $\mathbf{C}^{\mathbf{q}}$ to be universal (besides the normalization condition), one needs universal *ansatz* in trainable unitary parts and multiple circuits as one unitary orbit of the Hermitian matrix cannot cover the whole space of Hermitian matrix space.

In Eq. (15), one might wonder what $\tilde{\mathbf{P}}$ does. This tensor is a given by

$$\tilde{\mathbf{P}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & i \\ 0 & 1 & -i \\ 1 & 0 & 0 \end{pmatrix}^{\otimes N} = (|I\rangle\rangle\langle 0| + |X\rangle\rangle\langle 1| + |Y\rangle\rangle\langle 2|)^{\otimes N}. \quad (16)$$

In other words, the coefficient tensor $\mathbf{C}^{\mathbf{q}}$ is obtained by projecting out Z-containing Pauli coefficients of $O' \odot \rho^T$. Here, $\lambda_{\boldsymbol{i}}$'s are the Pauli coefficients when $O' \odot \rho^T$ is represented with the Pauli basis as

$$(O' \odot \rho^T)(\boldsymbol{\theta}) = \sum_{\boldsymbol{i}} \lambda_{\boldsymbol{i}}(\boldsymbol{\theta})\, \sigma_{i_1}^{(1)} \otimes \sigma_{i_2}^{(2)} \otimes \cdots \otimes \sigma_{i_N}^{(N)}, \quad (17)$$

where $\boldsymbol{i} = \{0, 1, 2, 3\}^{\otimes N}$ and $\sigma_{i_k} = \{I, X, Y, Z\}$. One can identify

$$\mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta})_{\tilde{\boldsymbol{i}}} = ((O' \odot \rho^T)(\boldsymbol{\theta}) \cdot \tilde{\mathbf{P}})_{\tilde{\boldsymbol{i}}} = 2^N \lambda_{\tilde{\boldsymbol{i}}}, \quad (18)$$

where now we have truncated indices $\tilde{\boldsymbol{i}} \in \{0, 1, 2\}^{\otimes N}$.

For instance, consider the exponential encoding on 1d input x, where $\phi_\alpha(x) = 3^{\alpha-1}x$ and $N = 3$. Then, one of the basis functions is $\cos(x)\sin(3x)\cos(9x)$, corresponding to $\tilde{\boldsymbol{i}} = (1, 2, 1)$. Therefore the coefficient of $\mathbf{T}_{121}(x) = \cos(x)\sin(3x)\cos(9x)$ is 2^N times that of the Pauli string $X \otimes Y \otimes X$ when $O' \odot \rho^T$ is represented in the Pauli basis.

This observation clarifies the previously opaque nature of a VQML model's coefficients and gives us a new way to understand the model in the context of operator spreading [22,23] or non-Cliffordness of the circuit [24]. We refer the Reader to Appendix B for more detailed discussions. It also allows for a more comprehensive analysis of noise effects on VQML models using techniques introduced in Refs. [25,26]. Refer to Appendix F for additional information.

To fully contract the function $\mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta}) \cdot \mathbf{T}(\boldsymbol{x})$ to get the output from the quantum model, computational resources on the order of $O(3\chi_q(\boldsymbol{\theta})^2 N)$ are required. Here, $\chi_q(\boldsymbol{\theta}, O) := \max_{k\in[N-1]} \chi_q^{(k)}$ represents the maximum bond dimension among all bond indices of $\mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta})$. This highlights the significance of the maximum bond dimension in MPS, as it directly influences the computational complexity of contracting MPS.

The value of χ_q depends on the circuit *ansatz*. However, for VQML models that include multiple two-qubit entangling layers, which is a common *ansatz* for variational circuits, χ_q can increase exponentially with the number of layers. To be more concrete, let us consider the "simple parallel model," depicted in Figs. 2(a) and 2(b1). Our trainable *ansatz* $W_1(\boldsymbol{\theta}_1)$ ($W_2(\boldsymbol{\theta}_2)$) is "hardware-efficient" *ansatz,* which is composed of L_1 (L_2) layers of N parametrized single-qubit unitaries and nearest-neighbor CNOT gates. That is

$$W_k(\boldsymbol{\theta}) = \Pi_{l=1}^{L_k} \big(\Pi_{i=1}^{N-1} U_{\text{CNOT}}^{l,i,i+1} \times \Pi_{i=1}^{N} U^{(l,i)}\big(\theta_1^{(l,i)}, \theta_2^{(l,i)}, \theta_3^{(l,i)}\big)\big). \quad (19)$$

We expect that this circuit *ansatz* results in $\chi_q \sim 3^{L_1 L_2}$, which is exponential with the depth of the VQML model. In

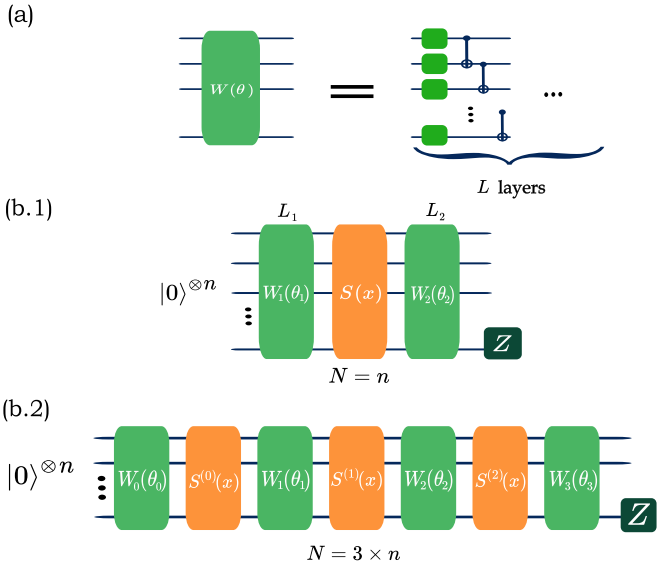

FIG. 2. The schematic diagrams illustrate the VQML models used for simulations. (a) The structure of trainable unitaries. These consist of L layers of hardware-efficient *ansatze*. Each small green box represents a parametrized single-qubit unitary $U(\theta^{(1)}, \theta^{(2)}, \theta^{(3)}) = \begin{pmatrix} \cos(\theta^{(1)}/2) & -e^{i\theta^{(3)}}\sin(\theta^{(1)}/2) \\ e^{i\theta^{(2)}}\sin(\theta^{(1)}/2) & e^{i(\theta^{(2)}+\theta^{(3)})}\cos(\theta^{(1)}/2) \end{pmatrix}$, which contains three free parameters. Though numerous options exist, This work focuses on results from this specific trainable circuit *ansatz*, chosen as an illustrative example for comparing VQML and classical models within the TN formalism. (b1) The diagrams for the simple parallel model. In this model, the number of qubits in the circuit, n, equals the number of single-qubit Pauli-Z encoding gates N. (b2) The general structure (data reupload) model. For the numerical results in this work, we consider a model with 3 times reuploading. Hence, $N = 3n$.

Appendix C, we numerically confirm that χ_q for this *ansatz* averaged over multiple random initializations of $\boldsymbol{\theta}$, indeed exhibits an exponential scaling with respect to the circuit depth. Our numerical observation implies that the typical $\mathbf{C}^{\mathbf{q}}$ of VQML models using polynomially growing computational resources can possess exponentially large bond dimensions, which makes function classes of these VQML models hard to generate classically.

To summarize, every VQML model is an MPS model with a feature map $\mathbf{T} : \boldsymbol{x} \mapsto \mathbf{T}(\boldsymbol{x})$, but subject to a special kind of coefficient MPS determined by the circuit *ansatz*. In the language of ML, VQML models are MPS model that possesses special *regularization* on coefficient MPS. For classical ML models, regularization is a process to reduce the complexity of learned functions and is executed by adding additional terms in the loss function or by using some heuristics during the training [27]. Analogously, employing VQML models corresponds to using an implicit "quantum" regularization by using quantum circuits as coefficient generators. Such a quantum regularization can be implemented classically by contracting a 2D unitary network (quantum circuit) as a coefficient tensor, which is in general hard for classical computers but efficient for quantum. This is where VQML models differ from classical MPS models.

IV. DEQUANTIZATION OF FUNCTION CLASSES OF VQML MODELS

The study of quantum-algorithm dequantization aims for developing *efficient* classical algorithm that are of comparable performances to the respective quantum algorithms, both of which utilize the same level of precomputational power. Following the discussions in Sec. II, here we define the notion of the VQML-model function.

Definition 1 (Dequantization of VQML's function class). For a given function class of the VQML model $F_Q(\boldsymbol{x};\boldsymbol{\Theta},\{\phi_\alpha\}_\alpha)$ utilizing n-qubits, a set of parameters $\boldsymbol{\Theta}$, and preprocessing functions $\{\phi_\alpha\}_\alpha$ obtained prior to VQML, the function class of the VQML model is *dequantized* by a classical computational model F_C if there exists such a F_C that requires $O(\text{poly}(n, 1/\delta, 1/\epsilon))$ computational resources and $f_C \in F_C$ such that

$$\mathop{\mathbb{P}}_{\boldsymbol{\theta}\sim\mu(\boldsymbol{\Theta})}[\mathcal{D}(f_Q(\boldsymbol{x};\boldsymbol{\theta}), f_C(\boldsymbol{x})) \leqslant \epsilon] \geqslant 1-\delta. \tag{20}$$

Here, $\mu(\boldsymbol{\Theta})$ represents the uniform measure over the trainable parameters set. This definition depends on the choice of distance function $\mathcal{D}$. Henceforth, terms like "dequantization of VQML" or "dequantizable VQML models" refer to the dequantization of their function classes. Additionally, throughout this paper, we only consider efficient VQML models that utilize $N = O(\text{poly}(n))$ encoding gates. Therefore n and N can be used interchangeably without altering a computational complexities.

Several points are worth highlighting here. First, it is important to understand that dequantized VQML models may still offer quantum advantages. This is because even if two models belong to the same function class, their performances such as sample efficiency or generalizability can differ largely due to their differences in training landscape. Second, the definition provided above should be considered as a "weak" form. There are scenarios where a trained quantum model generates a non ϵ-approximable function, which resides in a δ fraction of the function class. Hence, even if a VQML model is δ-dequantized, it may not be possible to classically approximate its other practically relevant output functions.

Nevertheless, the criteria for function class dequantization is a necessary condition for classical surrogates [28] of VQML models or shadow models of VQML models [29], the recently investigated. These surrogates or shadow models are classical counterparts designed to efficiently learn from quantum ML models, which can be used to replace the original, *trained* quantum models in later applications. For efficient learning and usage, it is crucial that classical models approximate the quantum model accurately while maintaining efficiency. This is the essential objective of VQML-model dequantization: it ensures that classical models can efficiently approximate their quantum counterparts.

This dequantization criterion can be extended to general variational quantum algorithms (VQA) beyond just QML scenarios by treating $\boldsymbol{x}$ as variational parameters alongside $\boldsymbol{\theta}$. We can determine how "quantum" the functions generated by a given variational quantum circuit family are. This perspective is instrumental in determining the potential of replacing VQA with classical algorithms as a whole. In this context, dequantization of the quantum-circuit *function class* (Definition 1) is a necessary condition for the overall dequantizability of the VQA itself. In contrast, if a quantum model is *not* dequantizable, then this means it can produce classically inefficient functions and therefore can be a prominent candidate for exhibiting quantum advantage.

We remark that Definition 1 is different with efficient simulatability of quantum circuits. Consider, for instance, the univariate naive Pauli encoded model. This model uses scalar inputs and all N preprocessing functions are identity functions, $\phi_\alpha(x) = x$. Regardless of the circuit's complexity, one can dequantize this model's function class with degree-N Fourier series [19,28], due to its small dimension of function space. Meanwhile, using exponential encoding [13], results in an exponential $(3^N/2 - 1)$-degree Fourier series, challenging dequantization with a simple strategy that just exploits a finite Fourier series. However, we can cope with this problem in Sec. V, by exploiting classical MPS models.

V. DEQUANTIZATION USING CLASSICAL MPS MODELS

We learned that the feature map is always expressible in a tensor-product form, and can thus be efficiently generated, given the set of preprocessing functions $\{\phi_\alpha\}_\alpha$. Using this and the fact that VQML models are MPS models, we take a classical MPS (CMPS)model that is basis-equivalent to the VQML model we would like to dequantize,

$$f_C(\boldsymbol{x};\boldsymbol{\theta}) \equiv \mathbf{C}^{\mathbf{c}}(\boldsymbol{\theta}) \cdot \mathbf{T}(\boldsymbol{x}), \tag{21}$$

where $\mathbf{C}^{\mathbf{c}}(\boldsymbol{\theta})$ is its coefficient part and $\mathbf{T}(\boldsymbol{x})$ is the feature map of the VQML model. The maximum bond dimension of $\mathbf{C}^{\mathbf{c}}(\boldsymbol{\theta})$, denoted as χ_c, may be set arbitrarily. If χ_c scales exponentially in N—which is the number of tensors in $\mathbf{T}$—then f_C can surely approximate all VQML functions but it is not efficient as contraction complexity for the CMPS model scales $O(\chi_c^2 N)$. Therefore we only focus on $\chi_c \sim O(\text{poly}(N))$.

For distance measure between functions $\mathcal{D}(f_Q, f_C)$, we shall adopt the two-norm squared distance,

$$\mathcal{D}_2(f_Q, f_C) := \frac{1}{|\Omega|}\int_\Omega |f_Q(\boldsymbol{x}) - f_C(\boldsymbol{x})|^2 d\boldsymbol{x}. \tag{22}$$

This is a natural choice if one considers mean squared error, which is a finite approximate version of the two-norm distance, as a performance measure for function regression tasks. Next, we can bound $\mathcal{D}_2(f_Q, f_C)$ using the two-norm distance of $\mathbf{C}^{\mathbf{q}}$ and $\mathbf{C}^{\mathbf{c}}$ from above as follows:

$$\begin{aligned}\mathcal{D}_2(f_Q, f_C) &= \frac{1}{|\Omega|}\int_\Omega \{(\mathbf{C}^{\mathbf{q}} - \mathbf{C}^{\mathbf{c}}) \cdot \mathbf{T}(\boldsymbol{x})\}^2 d\boldsymbol{x} \\ &= \langle\Delta|\frac{1}{|\Omega|}\int_\Omega |\mathbf{T}(\boldsymbol{x})\rangle\langle\mathbf{T}(\boldsymbol{x})| d\boldsymbol{x}|\Delta\rangle \\ &\leqslant \||\Delta\rangle\|_2^2 \|G\|_F,\end{aligned} \tag{23}$$

where $|\Delta\rangle := \mathbf{C}^{\mathbf{q}} - \mathbf{C}^{\mathbf{c}}$, G is the Gram matrix of $\mathbf{T}_i(\boldsymbol{x})$s, $G_{ij} := \frac{1}{|\Omega|}\int_\Omega \mathbf{T}_i(\boldsymbol{x})\mathbf{T}_j(\boldsymbol{x})d\boldsymbol{x}$, and Ω is the domain encompasses all possible inputs, not just the training data. We omitted the trainable parameter, θ, dependence for $\mathbf{C}^{\mathbf{c}}$ and $\mathbf{C}^{\mathbf{q}}$, and one should be aware that they are parametrized independently. From the above inequality, we see that

$$\||\Delta\rangle\|_2^2 \leqslant \epsilon/\|G\|_F, \tag{24}$$

which guarantees the approximation within the error tolerance. In other words, good approximability for coefficient tensors in terms of two-norm can be translated to good approximability for functions. Especially, when $\mathbf{T}(\boldsymbol{x})$ contains an orthonormal basis set in a given Ω, such as the Fourier function basis and $\Omega = [-\pi, \pi]$, we have the equality

$$\frac{1}{|\Omega|}\int_{\Omega}(f_Q(\boldsymbol{x}) - f_C(\boldsymbol{x}))^2 d\boldsymbol{x} = \|\mathbf{C^q} - \mathbf{C^c}\|_2^2 = \||\Delta\rangle\|_2^2. \quad (25)$$

A. Conditions for not dequantizable VQML models with CMPS models

From Eqs. (24) and (25), to permit dequantization, an accurate coefficient MPS approximation becomes important. When approximating an MPS $\mathbf{C^q}$ having a maximum bond dimension χ_q with a restricted MPS $\mathbf{C^c}(D)$ that has a maximum bond dimension $\chi_c = D$, the approximation error in terms of the two-norm is bounded from above by [15]

$$\|\mathbf{C^q} - \mathbf{C^c}(D)\|_2^2 \leqslant 2\sum_{k=1}^{N-1}\sum_{i=D+1}^{\chi_q}\left(s_i^k\right)^2 \equiv 2\eta(D). \quad (26)$$

Here, $\{(s_i^k)^2\}_{i=0}^{\chi_q}$ is the set of singular values obtained from the singular value decomposition of $\rho_Q^k := \mathrm{Tr}_{[k+1,k+2,\ldots,N]}|\mathbf{C^q}\rangle\langle\mathbf{C^q}|$, which is a reduced matrix for sites $1, 2, \ldots, k$, and $\eta(D)$ is a truncation error of $\mathbf{C^q}$ which is a sum of discarded singular values when only D largest values are kept.

The magnitude of $\eta(D)$ is dictated by the Renyi-α entropy of $\mathbf{C^q}$'s singular values. Note, also, that $\mathbf{C^q}$ is not a genuine "state," as $\mathrm{tr}|\mathbf{C^q}\rangle\langle\mathbf{C^q}| \neq 1$. By properly normalizing the $|\mathbf{C^q}\rangle\langle\mathbf{C^q}|$, one can obtain the Renyi-α entropy for the kth "cut" of $\mathbf{C^q}$,

$$S_\alpha^{(k)} := \frac{1}{1-\alpha}\log_2 \mathrm{tr}\left(\rho_Q^k\right)^\alpha. \quad (27)$$

We focus on the $\alpha = 2$ case as this serves as a criterion for efficient approximability of $\mathbf{C^q}$.

First, we propose a condition for VQML models with all orthonormal basis that are nondequantizable by CMPS models by virtue of $S_2^{(k)}$.

Proposition 1 (Highly entangled coefficient generating models are nondequantizable by CMPS models). VQML models satisfying $G = I$, such that

$$\mathbb{P}_{\theta\sim\mu(\boldsymbol{\Theta})}\left(S_2^{(k)}(\boldsymbol{\theta}) \in O(ck^\beta)\right) \geqslant 1 - \delta \quad (28)$$

for some constants c, and $0 < \beta < 1$, cannot be dequantized by CMPS models with $\mathcal{D}_2$ distance and any precision ϵ.

Proof. We use the result from Ref. [30],

$$\ln D \geqslant S_2^{(k)} + 2\ln(1 - \epsilon_{\mathrm{tr}}), \quad (29)$$

where D is the maximum bond dimension of approximating classical MPS $\mathbf{C^c}$, and ϵ_{tr} is the trace distance between density operators $|\mathbf{C^q}\rangle\langle\mathbf{C^q}|$ and $|\mathbf{C^c}\rangle\langle\mathbf{C^c}|$. $G = I$ implies that all basis functions in $\{\mathbf{T}_i(\boldsymbol{x})\}_i$ are orthonormal, from Eq. (25), we have

$$\mathcal{D}_2(f_Q, f_C) = \|\Delta\|_2^2 \leqslant 2\eta(D) \leqslant 2(N-1)\epsilon_{\mathrm{tr}}, \quad (30)$$

where the second inequality comes from the fact that $\eta(D)$ is bounded by the trace distance. Setting $\epsilon_{\mathrm{tr}} = \frac{\epsilon}{2(N-1)}$, we see that

$$D \geqslant 2^{S_2^{(k)} + 2\ln\left(1 - \frac{\epsilon}{2(N-1)}\right)}, \quad (31)$$

for $\mathcal{D}_2(f_Q, f_C) \leqslant \epsilon$. This states that if $S_2^{(k)}$ of $\mathbf{C^q}$ exhibits a growth rate faster than logarithmic in relation to its size (N or $k = cN$ for $0 < c \leqslant 1$), approximating it with efficient CMPS (having $D = \chi_c = O(\mathrm{poly}(N))$ model is impossible for any specified error tolerance ϵ ■.

Proposition 1 highlights that the "quantum" nature—classical infeasibility for an approximate generation—of a VQML output function is due to a highly entangled coefficient in an exponentially large dimensional space.

Besides the scaling of $S_2^{(k)}$, we may also look at $S_2^{\max} := \max_k S_2^{(k)}$. This stems from the reasoning that when an $\chi_c \gg 2^{S_2^{\max}}$, the CMPS model is expected to provide a suitable approximation [31]. Consequently, we regard a VQML model having a larger $S_2^{\max}$ as a *harder* model to dequantize compared to one with a smaller $S_2^{\max}$. We provide numerical evidence indicating that $S_2^{\max}$ can be a good estimator for efficient approximability of $\mathbf{C^q}$ in Appendix E.

From Sec. III C and Appendix C, kth bond dimension $\chi_q^{(k)}$ of $\mathbf{C^q}$ can grow exponentially with respect to N. As $S_2^{(k)} \leqslant \log_2\chi_q^{(k)}$, $S_2^{(k)}$ can also scale linearly with the site index k, and $S_2^{\max}$. Therefore we expect typical VQML models that are sufficiently deep to be hard to dequantize. In the following, we numerically confirm when highly entangled $\mathbf{C^q}$ is generated, thereby finding numerical evidence of nondequantizable VQML models.

1. Noiseless VQML models

Firstly we investigate the scaling behavior of S_2 of typical $\mathbf{C^q}$s from simple parallel models [Fig. 2(b1)] by varying the number of qubits N and number of layers L per trainable unitary (W_1 to W_2). Here, we set $L = L_1 = L_2$, because when $L_1 + L_2$ is fixed, setting $L_1 = L_2$ gives us the largest entanglement. (See Appendix D.)

The result is given in Fig. 3. It is observed that the $S_2^{(k)}$ curve approaches the Page curve for the Renyi-2 entropy [32] as L increases, where saturation occurs when $L \approx N$. The Page curve exhibits an almost linear scaling with respect to the subsystem size k, and its maximum value (at $k = \lfloor N/2 \rfloor$) exhibits a faster-than-logarithmic scaling with respect to N. Specifically, $S_2^{\max}(L) \sim 0.79N$ upon saturation. Consequently, these simulation results indicate that for polynomial-depth VQML models, typical $\mathbf{C^q}$ can possess high entanglement efficiently with a small number of parameters, so they are not likely to be dequantizable. Meanwhile, we can observe that shallow circuit depths generally lead to a sufficiently small $S_2^{\max}$, so that opens a possibility for dequantization by CMPS models.

Secondly, we study the data reuploading model [17], where the data encoding part is distributed across the trainable parts [Fig. 2(b2)]. As we argued in Sec. III A, general encoded VQML models can be rephrased as reuploading models so that their analysis results are a representative example for general encoded models. We follow Appendix A to construct the $\mathbf{C^q}$s for these cases. Especially we compare the basis-equivalent simple parallel and data reuploading models.

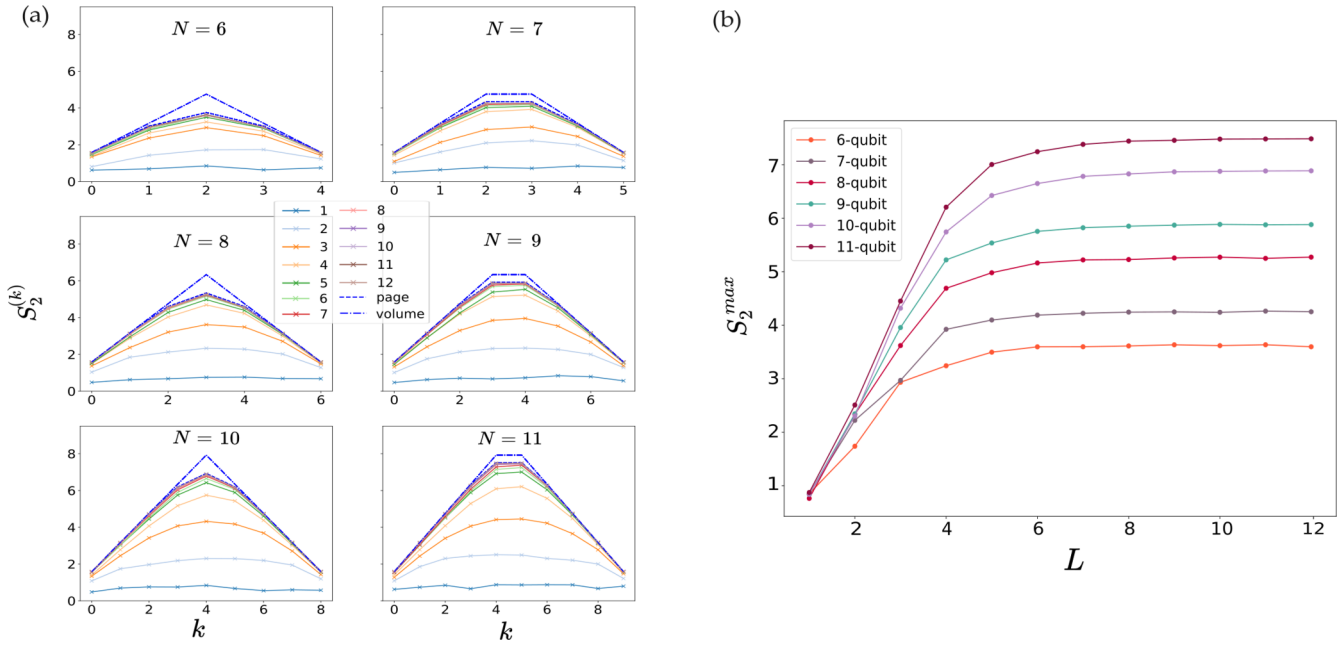

FIG. 3. Simulation results for the noiseless simple parallel model case. (a) The Renyi-2 entropy scaling of the coefficient tensors $\mathbf{C}^{\mathbf{q}}$ of the randomly parametrized models is plotted against the subsystem size (k). Each average is taken over 30 different parameter sets (24 for $N \geqslant 10$). The Page curve (blue dotted line) represents the Haar-averaged values for N-qutrit quantum states and the volume curve signifies the potential maximum value (the entropy curves for quantum states satisfying the volume law). The $S_2^{(k)}$ curves approach the Page curve when the number of layers is sufficiently large ($L \approx N$). (b) The maximum Renyi-2 entropy across all subsystem sizes, $S_2^{\max}$, increases with L in the simple parallel model, and saturates at $L \approx N$.

Figure 4 shows the simulation results. The reuploading model saturates to a lower $S_2^{\max}$ compared to the corresponding basis-equivalent parallel model. This indicates that when using the same number of data-encoding gates with enough trainable layers, it is harder to dequantize parallel models. One may also compare two basis-equivalent models that share similar quantum resources such as the total number of trainable layers or the total number of free parameters. Simulation results show that when the parallel model is shallow ($L = 2$ for our case), the reuploading model is harder to dequantize, but as L increases, the parallel model is always harder to dequantize for the same amount of quantum resources. Therefore, if one desires a VQML model that is not dequantizable, using a parallel model is the more plausible option.

This conclusion is also a consequence of structural differences between parallel and reuploading models. When both are basis-equivalent models using N encoding gates, the data reuploading model uses an n-qubit circuit that is smaller than N. Therefore, despite employing universal trainable *ansatze*—which can generate any $U \in \mathcal{U}(2^n)$—for *all* trainable unitary blocks in the data reuploading model, it is clear that the effective 2^N-dimensional unitary lies only in a subset of $\mathcal{U}(2^N)$ when the reuploading model is transformed into simple parallel form using wire-bending techniques (see Fig. 10). On the other hand, the simple parallel model can generate any 2^N-dimensional unitary in $\mathcal{U}(2^N)$ with any universal *ansatz*. This result is consistent with the study in Ref. [33].

2. Noisy VQML models

For NISQ devices, every bit of noise counts. Finite noise levels, however small, would destroy coherence, and weaken the entangling power of the circuit, making them easy to simulate with classical computers [31,34–36]. Similarly, noise effectively reduces the entanglement in the $\mathbf{C}^{\mathbf{q}}$ of NISQ VQML models, possibly allowing for a dequantization with CMPS models.

We shall now study the effects of noise by considering noisy two-qubit gates, each of which results in the two-qubit depolarizing error channel

$$E(\rho) = (1-\gamma)\rho + \frac{\gamma}{4} I \otimes I, \tag{32}$$

where γ is the error rate. The noisy quantum circuit layer introduces $(1-\gamma)$ factors to the Pauli coefficients of O' and ρ. As a consequence, after a sufficient number of noisy layers, O' and $\rho^\top$ ultimately become proportional to the identity matrix, so that $\mathbf{C}^{\mathbf{q}}$ converges to a product MPS. We conduct numerical simulations demonstrate the rate at which the entanglement of $\mathbf{C}^{\mathbf{q}}$ decreases while noise is present. Details concerning the simulation and analysis of noisy VQML models can be found in Appendix F

In Fig. 5(a), we observe that noise can significantly wash away the entanglement of $\mathbf{C}^{\mathbf{q}}$. For sufficiently large γ, the influence of noise overwhelms the entangling power of the circuit, resulting in a decrease in $S_2^{\max}$ after a relatively small L. By comparing the $N = 6$ and $N = 9$ cases in Fig. 5(a) and comparing models of different sizes with fixed γ in Fig. 5(b), we observe that larger N experiences more severe noise effects, with significantly large entropy differences. For instance, when $N = 6$, the decrease in $S_2^{\max}$ is not particularly significant until $L = 24$, whereas when $N = 11$, $S_2^{\max}$ quickly tends to 1 quickly as L approaches 24. This finding suggests

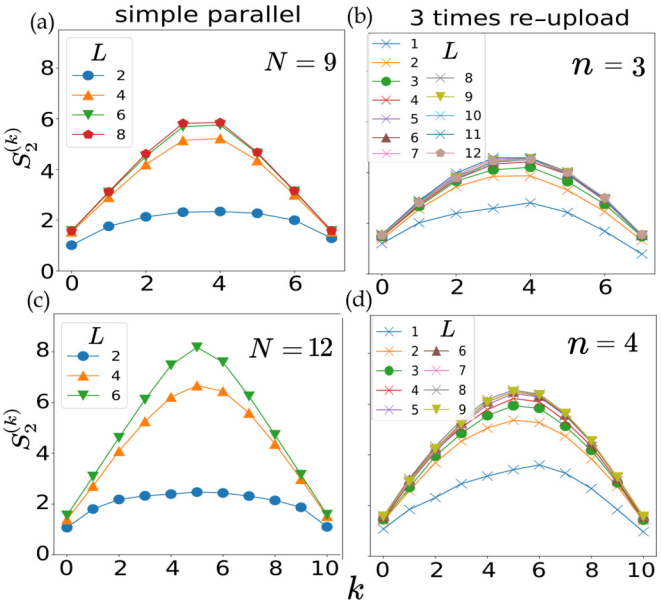

FIG. 4. Comparison between basis-equivalent simple parallel and data reuploading (general encoded) models. We tested $n = 3, 4$ data reuploading models having 4 trainable and 3 encoding blocks. These models [(b) and (d)] utilize $3n = N = 9, 12$ Pauli-Z encoding gates, rendering them basis-equivalent to the $n = N = 9, 12$ simple parallel models [(a) and (c)], respectively. The same color lines indicating the same total number of trainable layers, and identical markers denoting the same number of free parameters.

that, in situations where considerable noise is anticipated, employing a reuploading model would generate a larger variety of $\mathbf{C^q}$. Figure 5(c) supports the aforementioned claim through a comparison between the basis-equivalent ($n = 3$), three-round reuploading model and $N = 9$ parallel model. In the absence of noise, the parallel model is harder to dequantize. Still, when noise is present (with $\gamma = 0.1$ in this instance), the reuploading model becomes harder to dequantize as the total number of layers in trainable circuits increases.

B. Dequantizable VQML models

Thus far, we proposed a sufficient criterion for nondequantizable VQML models by CMPS models, namely that when their coefficient MPS $\mathbf{C_q}$ possess sufficiently high entanglement. Here we claim a more general and stronger statement regarding dequantizable VQML models.

Proposition 2. VQML models that have $O(\text{poly}(N))$ linearly independent functions in the basis set can be dequantized by CMPS models.

Proof. The basis functions of VQML models are components of the feature vector $\mathbf{T}(\boldsymbol{x})$. Suppose the set $\{\mathbf{T}_i(\boldsymbol{x})\}_i$ contains K linearly independent functions, indexed by $i' \in I$. Then, every function in the VQML model's function class can be represented as

$$\sum_{i' \in I} C_{i'} \mathbf{T}_{i'}(\boldsymbol{x}), \tag{33}$$

for some vector $C = \sum_{i' \in I} C_{i'} |i'\rangle$. Each computational basis ket $|i'\rangle$ is a product vector as it contains only one nonzero element. Summing MPSs results in a linear increase in bond dimension at most [37]. Consequently, C has a bond dimension of at most K. This demonstrates that all functions in a VQML model with K linearly independent basis functions can be constructed using a CMPS model with $\mathbf{C^c}$ of bond dimension at most K. The computational complexity of this model is $O(K^2 N)$, which proves the proposition. ■

Proposition 2 is "strong" in the sense that it applies to *all* functions (all $\boldsymbol{\theta}$ in $\boldsymbol{\Theta}$) in the function class. Moreover, this dequantizability only comes from the property of the feature map, not depending on how complex or hard $\mathbf{C^q}$s from the VQML models are. In other words, VQML models having polynomial-dimensional function class can be dequantized with $\delta = 0$, irrespective of the specific choice of the distance function $\mathcal{D}$ in definition 1, and regardless of the circuit *ansatz* employed. Proposition 2 shows that it is necessary to have an exponential dimension for a genuine nondequantizable model. We have presented the naive Pauli-encoded model as a representative example of a dequantizable model, and now we can confirm this fact by treating it as a special case of the above proposition.

To demonstrate Proposition 2 more extensively, we conduct function regression tasks using CMPS models with $f_Q(\boldsymbol{x}) = \mathbf{C^q} \cdot \mathbf{T}(\boldsymbol{x})$s as the target function. The target coefficient $\mathbf{C^q}$ is generated by a noiseless $N = 6, 8$, $L = 10$ simple parallel model. We normalize $\mathbf{C^q}$ so that $\|\mathbf{C^q}\|_2 = 1$. Such models result in a high $S_2^{\max} = 3.61$ and 5.29, suggesting the need for a CMPS having $\chi_c \gg 2^{S_2^{\max}}$ for small $\|\Delta\|_2^2$. We choose two different encoding strategies, naive encoding $[\phi_\alpha(x) = x]$, and exponential encoding $[\phi_\alpha(x) = 3^{\alpha-1}x]$.

The number of linearly independent functions in the components of $\mathbf{T}(\boldsymbol{x})$—the dimension of the function space—is equal to the rank of the Gram matrix G. Therefore one can say that VQML models can be dequantized by rank(G) CMPS models [if rank(G) is polynomial to N]. It is important to note that the rank of G is also affected by the total input domain Ω which encompasses all possible inputs including training and test datasets. If Ω is a discrete set having M elements, then rank$(G) \leqslant M$, which says the effective dimension of the function class is limited to M. This is because every component in $\mathbf{T}(\boldsymbol{x})$ can now be represented by an M-dimensional vector. In other words, if we have limited access to only poly(N) input points (note that we are *not only* considering training points but also all possible test inputs), then VQML models are dequantizable by CMPS models.

Since rank$(G) \propto \|G\|_F$, one can also infer from Eq. (23) that a lower rank of G allows a large variation in the coefficients, suggesting that VQML models with a smaller rank of G are more susceptible to dequantization, as they allow for larger coefficient discrepancies.

Figure 6(a) depicts the approximation performances using small bond-dimensional CMPS models. The target functions are from $N = 8$ naively encoded VQML models, resulting in rank(G) being polynomial in N ($K = 2N + 1$). In this scenario, it is expected that a $\chi_c = K = 17$ would perfectly express the VQML target functions. However, in this case, even a CMPS model with $\chi_c = 4$ would have been sufficient to closely approximate the target function.

Figure 6(b) represents the scenario where we employ exponential encoding to generate functions of exponentially large

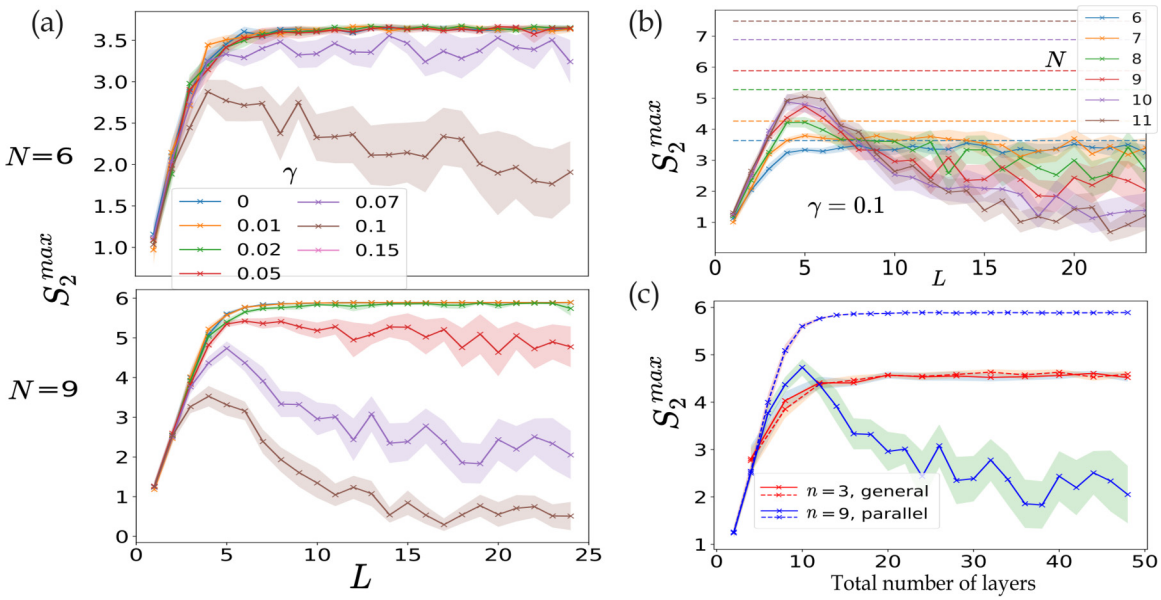

FIG. 5. (a) The impact of noise on $S_2^{\max}$ for N= 6 and 9 simple parallel models with varying error rates (γ). (b) The effect of noise on $S_2^{\max}$ for the $\gamma = 0.1$ case with different numbers of qubits. The dashed lines denote the maximum values for noiseless scenarios. (c) A comparison of the $n = N = 9$ parallel model and the basis-equivalent $n = 3$ reuploading models when $\gamma = 0.1$. The dashed lines correspond to noiseless scenarios. As the two models possess different numbers of trainable blocks, the x axis is set to the total number of layers rather than L, which indicates the number of layers for a single trainable block. All lines in this figure represent averages over 30 distinct parameter initializations, and the shaded regions indicate the 0.95 confidence level.

dimensions. However, $M = 500 \ll K = 3^8$, so $\text{rank}(G) \leqslant 500$ is significantly lower than the maximum possible value.

We note high similarity between $\mathbf{C^c}$ and $\mathbf{C^q}$ is not required for a close approximation of the function in these cases. One can also observe the unproportionate relationship between $\||\Delta\rangle\|_2^2$ and $\mathcal{D}_2$ in certain situations. These simulation results tell us that CMPS models with very large bond dimensions are not necessary for VQML dequantization when $\text{rank}(G)$ is small—CMPS models can approximate functions from VQML models well despite the hard-to-approximate $\mathbf{C^q}$s (high $S_2^{\max}$) when $\text{rank}(G) \ll O(\exp(N))$. As a side note, in Fig. 6(c), we use sufficiently many sample points to achieve full-rank G. In this case, one witnesses the proportionate relationship between function and coefficient distances, which requires small $\|\Delta\|_2^2$ for small $\mathcal{D}_2$ as expected form the inequality in Eq. (24).

In summary, in this section, we identified conditions for which VQML function classes are (likely) dequantizable or not by CMPS models.

(i) VQML models with $\text{rank}(G) = O(\text{poly}(N))$ (where the dimension of the function space is $O(\text{poly}(N))$ are dequantizable.

(ii) Noiseless shallow-depth (logarithmic in N) circuit models are likely dequantizable, while polydepth circuits are not.

(iii) Noisy VQML models possessing large widths and depths are likely dequantizable.

Note that while the first statement is rigorous, the second and third are based on numerical evidence corresponding to Figs. 3–5. So informally, a nondequantizable VQML function requires a function space of an exponentially large dimension and a highly entangled coefficient on this function space.

VI. EFFICIENT CLASSICAL KERNEL INDUCED FROM A QUANTUM KERNEL

The kernel method plays a crucial role in the context of FLM [38]. Every feature map $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^K$ in FLM introduces a kernel $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \mathcal{F}(\boldsymbol{x}_i) | \mathcal{F}(\boldsymbol{x}_j) \rangle$, which is the inner-product evaluation between feature-mapped data. Quantum kernel methods utilize quantum circuits to generate elements of kernel matrices. Analogously, a quantum kernel is defined as

$$\mathcal{K}_q(\boldsymbol{x}_i, \boldsymbol{x}_j) = \text{Tr}[\sigma(x_i)\sigma(x_j)], \tag{34}$$

where $\sigma(\boldsymbol{x}) = \mathbf{S}_{\text{enc}}(\boldsymbol{x})|0\rangle\langle 0|\mathbf{S}_{\text{enc}}^\dagger(\boldsymbol{x})$ is the data-encoded quantum state [39]. For any given quantum kernel, we have preprocessing functions that were precomputed before implementing $\mathbf{S}_{\text{enc}}$, and this naturally induces (normalized) a basis-equivalent product kernel

$$\begin{aligned}\mathcal{K}_c(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \frac{1}{2^N}\langle \mathbf{T}(\boldsymbol{x}_i) | \mathbf{T}(\boldsymbol{x}_j) \rangle \\ &= \frac{1}{2^N}\prod_{\alpha=1}^{N}[1 + \cos(\phi_\alpha(\boldsymbol{x}_i) - \phi_\alpha(\boldsymbol{x}_j))],\end{aligned} \tag{35}$$

where $\mathbf{T}$ is constructed from the preprocessing functions of the given quantum model. Note that it takes $O(N)$ time steps to calculate Eq. (35) as it simply involves an inner product between two tensor-product MPSs.

The representer Theorem [27] states that any optimal function f^{opt} that minimizes a given empirical regularized loss functional $\mathcal{L} : (f, \{x_i, y_i\}_i^{M_t}, \lambda) \to \mathbb{R}$ can be represented as

$$f^{\text{opt}}(\cdot) = \sum_{i=1}^{M_t} \gamma_i \mathcal{K}(\cdot, x_i). \tag{36}$$

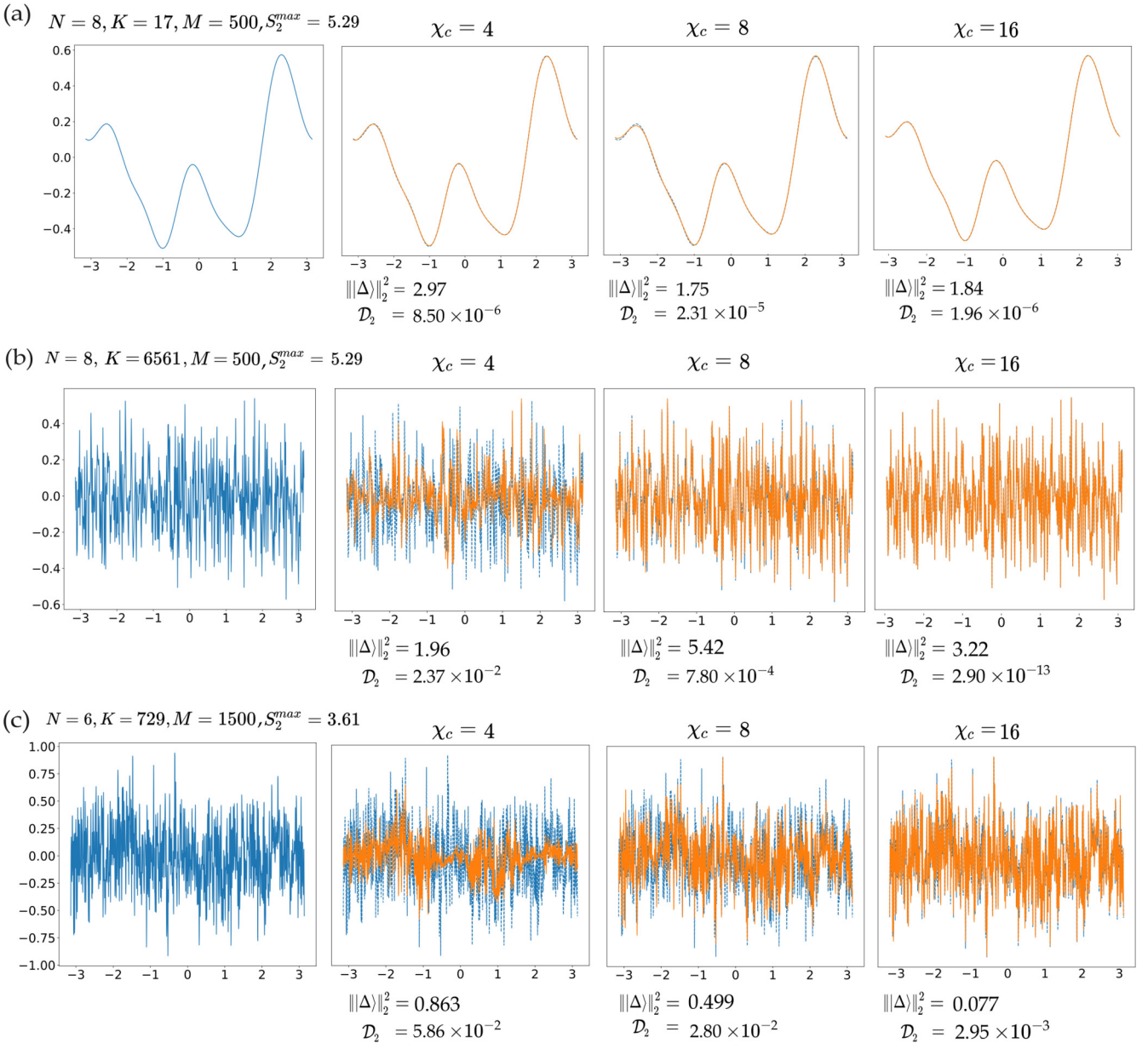

FIG. 6. The VQML function regression with CMPS models. The target functions (blue lines), $f_Q(x)$, are generated by randomly parameterized $L = 10$ VQML models in noiseless settings. The variable K represents the number of linearly independent basis functions of the models. The training set is composed of $\{x_j, f_Q(x_j)\}_{j=1}^M$, where x_j values are linearly spaced numbers ranging from $-\pi$ to π. The optimization of $\mathbf{C}^{\mathbf{c}}$s was carried out with all M data points serving as training data. The orange lines depict the values from the denoted χ_c-CMPS models after 500 training epochs. Each graph provides the 2-norm distance of coefficients after training, $\||\Delta\rangle\|_2^2$, and function distance $\mathcal{D}_2$ between the target and approximating functions. The naive Pauli encoding was used in (a) to generate $poly(N)$ dimensional function, while in (b) and (c), exponential encoding is employed. Target functions of (a) and (b) share the same coefficient tensor $\mathbf{C}^{\mathbf{q}}$, which have the same $S_2^{\max}$ values.

The optimal weights $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_{M_t})^\top$ admit analytical solution when we know the whole kernel matrix elements evaluated with the training dataset $\{x_i, y_i\}_i^{M_t}$. The function f^{opt} resides in the so-called reproducing kernel Hilbert space (RKHS) [27], which is the function space that is spanned by the kernel functions $\mathcal{K}(\cdot, x_i)$s.

From the observation that every function generated from a quantum circuit encoded with classical data can be represented as an FLM with feature map $\mathbf{T}$, we have the following proposition.

Proposition 3. The RKHS from any quantum kernel $\mathcal{K}_q(\boldsymbol{x}_i, \boldsymbol{x}_j) = \text{Tr}[\sigma(x_i)\sigma(x_j)]$ using N preprocessing functions $\{\phi_\alpha\}_\alpha$ is included in the RKHS of efficient basis-equivalent product kernel

$$\mathcal{K}_c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \mathbf{T}(\boldsymbol{x}_i) | \mathbf{T}(\boldsymbol{x}_j) \rangle. \tag{37}$$

Proof. Let a function from the RKHS of $\mathcal{K}_q$ be

$$f_q(\boldsymbol{x}) = \sum_i a_i \text{Tr}[\sigma(x_i)\sigma(x)] = \text{Tr}[O\sigma(x)], \tag{38}$$

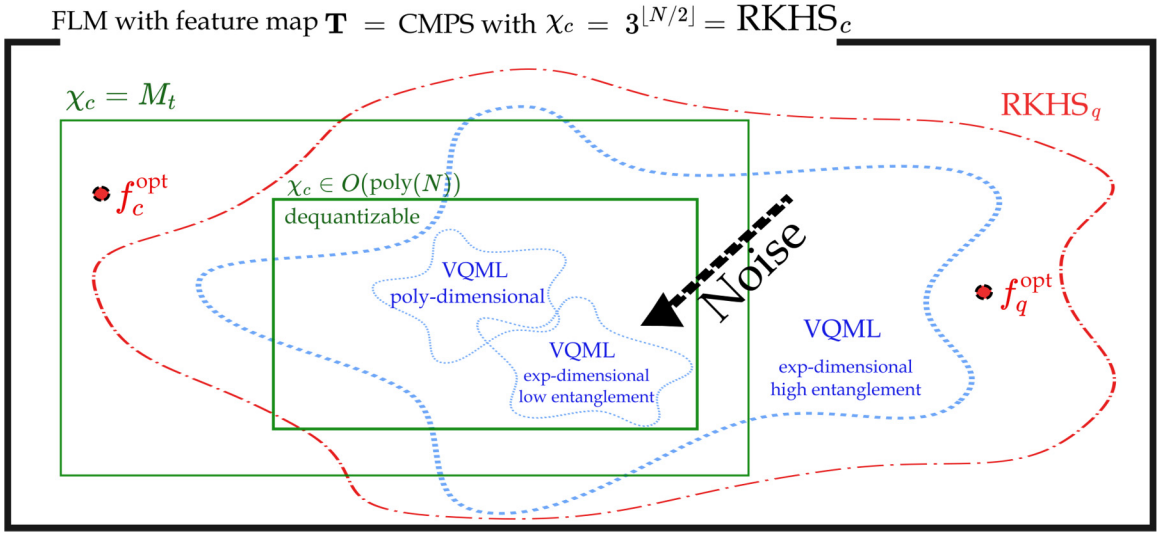

FIG. 7. The relationship between the function spaces of basis-equivalent CMPS models, VQML models, and the corresponding RKHSs. All models reside within the MPS model with the feature map $\mathbf{T}$, where the maximum bond dimension is $3^{\lfloor N/2 \rfloor}$. The blue wavy stars denote the function spaces of VQMLs, and the green rectangles indicate the space of CMPS models with designated χ_c. Noise can reduce the entanglement of $\mathbf{C}^{\mathbf{q}}$, thereby enabling dequantization and reducing the size of the function space. The $f_{c/q}^{\text{opt}}$ are optimal functions derived from the corresponding kernel methods. Furthermore, f_c^{opt} trained with M_t data is strictly contained within the CMPS model space of $\chi_c = M_t$.

where $O = \sum_i a_i \sigma(\boldsymbol{x}_i)$. Noting that $\sigma(\boldsymbol{x})$ is generated by utilizing data-encoding gates and some nonparametrized quantum gates, this is simply a linear sum of basis functions in $\{\mathbf{T}_i(x)\}_{i=1}^{3^N}$, just like a VQML model using the same data-encoding gates. Considering the functions in the RKHS of $\mathcal{K}_c$ are given by

$$f_c(\boldsymbol{x}) = \mathbf{C} \cdot \mathbf{T}(\boldsymbol{x}), \tag{39}$$

we conclude the proof. ■

It is important to note that Proposition 3 is not about the dequantization of RKHS of quantum kernels in the sense of Definition 1, but rather about the inclusion between two RKHSs of different kernels. Now let us denote the RKHS from quantum/classical kernel as RKHS$_{q/c}$. The optimal function f^{opt} from the quantum kernel method indeed belongs to RKHS$_q \subseteq$ RKHS$_c$. However, the existence of the analytical solution $\boldsymbol{\gamma}$ of f^{opt} posits the absence of any constraint on the coefficients of the model. That is when f^{opt} is represented in the MPS form, $f^{\text{opt}}(x) = \boldsymbol{c}^{\text{opt}}(\boldsymbol{\gamma}) \cdot \mathbf{T}(\boldsymbol{x})$, the optimal coefficient vector $\boldsymbol{c}^{\text{opt}}(\boldsymbol{\gamma})$ can be *any* 3^N-dimensional vector that has a large bond dimension when represented as an MPS. In other words, some functions in RKHS$_q$ might not allow for an efficient classical description using a poly-χ_c CMPS model, and is thus nondequantizable in general (unless it is $O(\text{poly}(N))$-dimensional such that Proposition 2 holds).

As discussed above, RKHS includes parametrized function classes, so that constrained $\mathbf{C}^{\mathbf{q}}$s also belong to RKHS$_q$. Moreover, upon noticing that

$$f_c^{\text{opt}} = \sum_{i=1}^{M_t} \gamma_i \mathcal{K}_c(\boldsymbol{x}, \boldsymbol{x}_i) = \frac{1}{2^N} \sum_i^{M_t} \gamma_i \langle \mathbf{T}(\boldsymbol{x}_i) | \mathbf{T}(\boldsymbol{x}) \rangle, \tag{40}$$

according to the representer Theorem, we find that $\frac{1}{2^N} \sum_i^{M_t} \gamma_i \langle \mathbf{T}(\boldsymbol{x}_i)| = \mathbf{C}^{\mathbf{c}}{}_{\text{opt}}$ is an MPS with bond dimension at most M_t, which is the number of training data. Equivalently, f_c^{opt} resides in the function class of CMPS models having $\chi_c = M_t$. The relationships among the function classes of VQML, CMPS, and RKHSs are illustrated in Fig. 7. The reader may consult Ref. [40] for more discussion about the difference between the variational model and the kernel method.

Kernel methods sharing the same RKHS do not necessarily exhibit the same performances such as generalizability. Here, we compare the quantum kernel method based on IQP encoding using two repetitions of encoding gates, which is conjectured to be hard to simulate classically [18], and the corresponding basis-equivalent product kernel method. We again consider the function regression task from the relabeled f-MNIST dataset as in Appendix G 2, but now following Refs. [40,41], target values are generated by the randomly parametrized n-qubit quantum circuits. The IQP encoding of repetition two contains $2n^2$ ϕ_αs when transformed to a parallel model *via* the procedure in Appendix A. These $2n^2$ preprocessing functions include the trivial function $\phi_\alpha(\boldsymbol{x}) = 0$, and the basis-equivalent CMPS model has a length of $2n^2$. However, as $\phi_\alpha(\boldsymbol{x}) = 0$ does not affect the number of linearly independent basis functions, there are only $4n - 2$ nontrivial preprocessing functions. We order them as

$$\phi_\alpha(\boldsymbol{x}) = \begin{cases} x_\alpha, & \alpha \in [1, n] \\ x_{\alpha-n} x_{\alpha-(n-1)}, & \alpha \in [n+1, 2n-1] \\ x_{\alpha-(2n-1)}, & \alpha \in [2n, 3n-1] \\ x_{\alpha-(3n-1)} x_{\alpha-(3n-2)}, & \alpha \in [3n, 4n-2] \end{cases}, \tag{41}$$

and create the $\mathbf{T}(\boldsymbol{x})$ with $N = 4n - 2$ to evaluate $\mathcal{K}_c$. We fit models with a sample of 500 training data using kernel ridge regression which exploits regularized loss Eq. (G1) with $\lambda = 0.01$, and test losses are computed with 100 unseen data.

Figure 8 presents the mean squared error (MSE) value from training and test datasets after training. It is evident that the training loss incurred using the kernel method using $\mathcal{K}_c$ is comparable to that of $\mathcal{K}_q$; both are of the order 10^{-4}. This

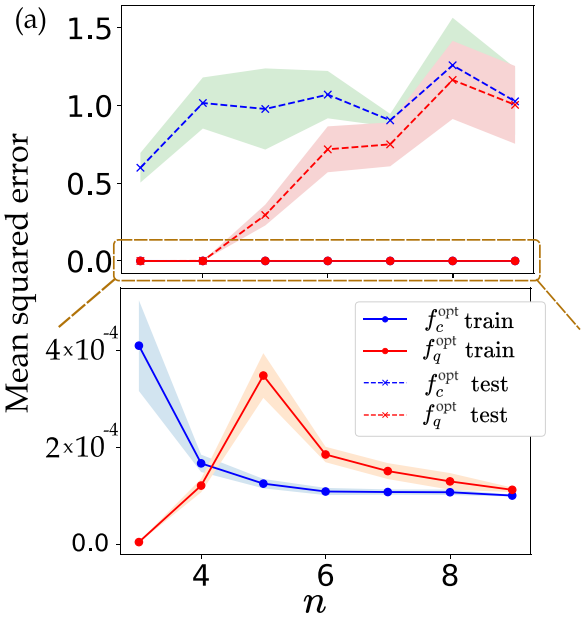

FIG. 8. Training and test losses derived from the kernel methods, which are trained using the VQML generated relabeled f-MNIST dataset. The figure panel offers a magnified view of the training losses.

indicates that f_c^{opt} fits the training data almost flawlessly, reflecting its good expressivity on VQML-generated functions. The training loss from $\mathcal{K}_c$ is lower than any other classical methods explored in the study by Ref. [40]. However, the test loss is higher than the quantum kernel for smaller system sizes, though the two become comparably worse as the system size expanded.

Poor generalizability of the quantum kernel with increasing quantum-circuit size is expected as explained in Appendix H. in Ref. [41]. The key point is that the kernel matrix approaches the identity as the Hilbert-space dimension grows exponentially in the qubit number so that good generalizability demands an exponentially growing data sample size. The same thing happens in basis-equivalent kernel $\mathcal{K}_c$ as it inherits an exponentially large dimensional feature map. We note that adjusting preprocessing functions can help improve generalization in quantum kernel [42], and this is also applicable to basis-equivalent product kernel if the VQML model is a simple parallel model, as two kernels are the same.

VII. CONCLUSIONS AND DISCUSSIONS

In this work, by using the tensor network (TN) formalism, we reveal that any variational quantum machine learning (VQML) model using classical data as its input is a linear model in the featured space, but has constrained coefficients and a feature map of tensor-product form. This general structure enables us to treat VQML models as a subclass of matrix product state (MPS) machine learning models, and offers a nuanced understanding of their characteristics in order to distinguish classically accessible components from genuinely quantum aspects. In QML applications we introduced a definition for dequantizing VQML model function classes, establishing it as a necessary condition for substituting VQML algorithms with classical ML algorithms as a whole. Employing classical MPS models for dequantization, we identified conditions under which VQML models are (or are not) dequantizable. Our numerical analysis provides evidence illustrating these conditions. In carrying out kernel learning methods, we propose a basis-equivalent product kernel, that is efficient and comparable in expressiveness to quantum kernels, opening new possible pathways for their dequantization.

The premise of a fair comparison between a quantum algorithm of interest and its classical counterpart is the consideration of equivalent levels of precomputations as free operations. Therefore, comparing all models of a common feature map, or that are equivalent in the feature basis is not only a viable approach but also concretely separates what classical models can/cannot perform resource efficiently. Under this basis-equivalent comparison framework, the dequantizing classical model can take any tensor-network structure that could resemble the common multi-scale entanglement renormalization *ansatz* (MERA), the projected entangled-pair state (PEPS), or even a neural-network structure that possesses the VQML model's preprocessing functions as its activation functions. An interesting direction for a follow-up study would be to explore the approximation capabilities of these different classical models.

Based on our findings of this work, we see that the core difference between classical and QML models lies in the entanglement content of their coefficient parts, not in the exponentially large feature space as the latter can always be efficiently computed classically. A genuine "quantum" function, which is inefficient to generate with classical models, is one that has a highly entangled coefficient with a small number of parameters in an exponentially large dimensional function space. We suggest looking for data that are attributed to this kind of functions for a quantum advantage with VQML. Recent studies on TN structured QML [43–47] is in line with this perspective: they utilize quantum circuits instead of the dense classical tensor blocks to construct the TN model, resulting in a highly entangled yet sparsely parametrized network. These kinds of procedures may be understood as an implicit "quantum" regularization of variational models, which, incidentally, is also carried out as soon as one chooses VQML models over classical MPS ones. Indeed the efficiency in implementing quantum regularization is the advantage of quantum devices. We conducted performance comparisons in Appendix G between VQML and CMPS models, highlighting the better generalizability of quantum regularization on MPS models.

We again emphasize that the dequantization concept discussed in this work relates to the expressivity of machine-learning function classes and does not directly correlate with other aspects of dequantization, such as generalizability or sample complexity during training (trainability). While this study only focuses on the expressivity of VQML models, we recognize the significance of generalizability or trainability issues as well. We believe that the analysis we have developed provides a foundation for further exploration into these crucial aspects of QML. For example, the generalization error bound of ML models is related to the capacity of function classes

[27,48], and several capacity measures have been proposed [49–51]. Our work could provide an alternative route to access the capacity of the function space through MPS expressivity or operator spreading [52] which relates to the model coefficients.

Regarding the trainability issue, recent research [53] has established a link between the trainability and classical simulatability of variational quantum algorithm (VQA) models. They introduced the concept of classical simulation (CSIM) of VQA models, which bears resemblance to our definition of dequantization 1. While our dequantization only necessitates the *existence* of approximating classical functions in the dequantizing classical model, CSIM demands the approximation of the VQA model's output, given $\boldsymbol{\theta}$ and its description. According to Proposition 2, any model with a polydimensional function class can be dequantized. However, if $\mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta})$ is derived *via* a deep quantum circuit, it might be hard to approximate the model's output, based solely on the quantum circuit's description. This scenario highlights a potential divergence between CSIM and dequantization as discussed in this work. Bridging the concepts presented in Ref. [53], which employs the language of operator space, with those of our study—articulated in the language of function space—could yield deeper insights for resolving trainability issues.

We note that in Ref. [54], extensive results on the scaling and growth of entanglement entropy in VQML models using the MPS formalism are discussed. While their findings may appear to be similar to our simulations in Sec. V A, our objects of analysis are different. Reference [54] focuses on the entanglement entropies of *quantum states* in VQML circuits, whereas we investigate the entanglement entropies of *coefficient MPS*s describing the featured linear models. This distinction is crucial, as entanglement at the function level can significantly differ from circuit-level entanglement. Our approach not only relates circuit-level entanglement to function but also identifies noncritical entanglement in states, as discussed in Appendix B. Moreover, we note that ML models utilizing TN structures have been extensively studied in the literature but without any direct connection to QML [12,55–60], the latter of which is now uncovered by the main results of this work. The fruits of labor for such a connection are a unified TN perspective on VQML models and the concept of function-class dequantization, which enables a deeper understanding of hardness in variational quantum models.

ACKNOWLEDGMENTS

The authors are grateful for insightful and beneficial discussions with C. Oh, H. Kwon, C. Y. Park, and R. Sweke. This work is supported by Hyundai Motor Company, the National Research Foundation of Korea (NRF) grants funded by the Korea government (Grants No. 2023R1A2C1006115, No. RS-2023-00237959, No. NRF-2022M3E4A1076099, and No. NRF-2022M3K4A1097117) via the Institute of Applied Physics at Seoul National University, and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2021-0-01059 and IITP-2023-2020-0-01606). Y.S.T. acknowledges support from the Brain Korea 21 FOUR Project grant funded by the Korean Ministry of Education.

APPENDIX A: VQML MODELS ARE MPS MODELS

1. Simple parallel case

The simple parallel models are written as

$$f_Q(\boldsymbol{x};\boldsymbol{\theta},W_1,W_2,O)=\langle 0|W_1^\dagger(\boldsymbol{\theta}_1)\mathbf{S}^\dagger(\boldsymbol{x})W_2^\dagger(\boldsymbol{\theta}_2)O\cdots W_2(\boldsymbol{\theta}_2)\mathbf{S}(\boldsymbol{x})W_1(\boldsymbol{\theta}_1)|0\rangle, \tag{A1}$$

where $\mathbf{S}(\boldsymbol{x})=\Pi_{\alpha=1}^{N}e^{-i\phi_\alpha(\boldsymbol{x})Z_\alpha/2}$, and $\boldsymbol{\theta}\equiv(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)$. We start from rewriting Eq. (A1) as

$$f_Q(\boldsymbol{x};\boldsymbol{\theta},W_1,W_2,O)=\mathrm{Tr}\mathbf{S}^\dagger(\boldsymbol{x})O'(\boldsymbol{\theta}_2)\mathbf{S}(\boldsymbol{x})\rho(\boldsymbol{\theta}_1), \tag{A2}$$

where we denoted the evolved observable as $O'(\boldsymbol{\theta}_2):=W_2^\dagger(\boldsymbol{\theta}_2)OW_2(\boldsymbol{\theta}_2)$ and preencoded state as $\rho(\boldsymbol{\theta}_1):=W_1(\boldsymbol{\theta}_1)|0\rangle\langle 0|W_1^\dagger(\boldsymbol{\theta}_1)$. Observe that $\mathbf{S}(\boldsymbol{x})$ is diagonal, so we use the property of the Hadamard product (denoted by $\odot$) [61]

$$\mathrm{Tr}D_1^\dagger AD_2B=\langle D_1|(A\odot B^T)|D_2\rangle, \tag{A3}$$

where $D_1(D_2)$ is a diagonal matrix and $|D_1\rangle(|D_2\rangle)$ corresponds to the ket constructed from elements of matrix $D_1(D_2)$. Using this, we have

$$f_Q(\boldsymbol{x};\boldsymbol{\theta},W_1,W_2,O)=\langle\mathbf{S}(\boldsymbol{x})|(O'\odot\rho^T)(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)|\mathbf{S}(\boldsymbol{x})\rangle, \tag{A4}$$

where

$$|\mathbf{S}(\boldsymbol{x})\rangle=\bigotimes_{\alpha=1}^{N}|S^{(\alpha)}(\boldsymbol{x})\rangle=\bigotimes_{\alpha=1}^{N}\begin{pmatrix}e^{-i\phi_\alpha(\boldsymbol{x})/2}\\ e^{i\phi_\alpha(\boldsymbol{x})/2}\end{pmatrix}. \tag{A5}$$

We are considering N-qubit circuit, so $O'\odot\rho^T$ is a $2^N\times 2^N$ matrix having $2N$ indices. We *vectorize* this $(O'\odot\rho^T)(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)$ by gathering the same site indices, thereby obtaining

$$f_Q(\boldsymbol{x};\boldsymbol{\theta},W_1,W_2,O)=\langle(O'\odot\rho^T)(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)|(|\mathbf{S}^*(\boldsymbol{x})\rangle\otimes|\mathbf{S}(\boldsymbol{x})\rangle). \tag{A6}$$

See Fig. 9(a) for the graphical description.

As a result, we see that the VQML model is decomposed into the basis part which only depends on the input data and preprocessing functions ϕ_αs, and the coefficient part which depends on the rest. For a further analysis of the coefficient part, we represent $\langle(O'\odot\rho^T)(\boldsymbol{\theta})|$ in the MPS form

$$\langle(O'\odot\rho^T)(\boldsymbol{\theta})|=\sum_{k_1k_2\cdots k_{N-1}}M^{(1)}_{l_1j_1k_1}M^{(2)}_{l_2j_2k_1k_2}\cdots M^{(N)}_{l_Nj_Nk_{N-1}}, \tag{A7}$$

where all matrices $M^{(\alpha)}$s are parametrized by $\boldsymbol{\theta}$, and an upper index α denotes the "site" of MPS. The index α ranges from 1 to the number of encoding gates N. One should understand $l_\alpha j_\alpha$ as a physical index of αth site of the MPS. The number of qubits in the circuit n equals N for simple parallel models we consider here, but $N\neq n$ for a general structured model as explained in Appendix A.

Note that the feature kets $|S^{*(\alpha)}(\boldsymbol{x})\rangle\otimes|S^{(\alpha)}(\boldsymbol{x})\rangle$ have redundant elements, 1s. To remove this redundancy, we adopt local

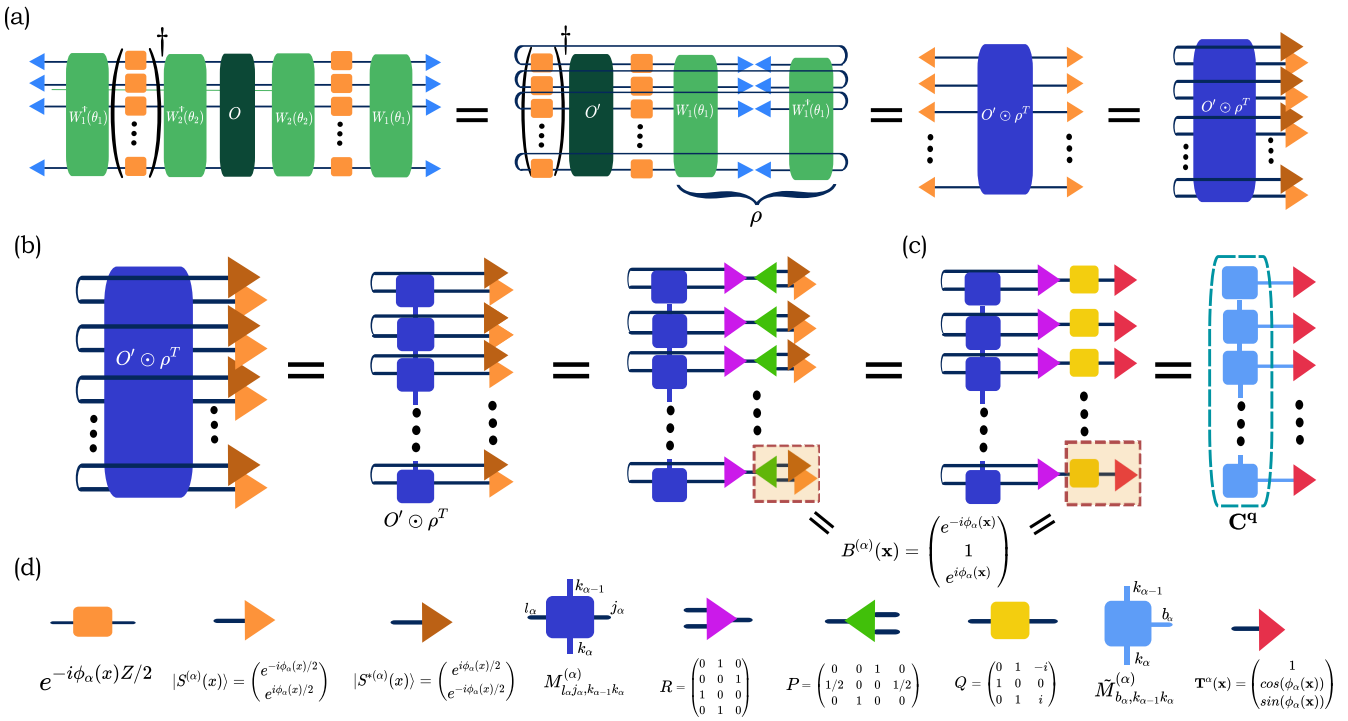

FIG. 9. Graphical description of transforming a simple parallel VQML model to MPS model form. (a) The model, $f_Q(\boldsymbol{x};\boldsymbol{\theta})$, into an FLM form. For the sake of simplicity, we have omitted the $\boldsymbol{\theta}$ dependence in both O' and ρ^T. (b) The coefficient part, denoted as $O'\odot\rho^T$, is morphed into an MPS form. This process involves transforming it into a Matrix Product Operator (MPO), vectorizing it, and applying additional tensors. The resulting MPS model form incorporates the feature map $\mathbf{B}$. (c) The final result has the feature map that is $\bigotimes_{\alpha=1}^{N}\mathbf{T}^{(\alpha)}(\boldsymbol{x})$. The coefficient component becomes the MPS $\mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta_1},\boldsymbol{\theta_2})$, representing the contracted form of $(O'\odot\rho^T)\cdot\mathbf{R}\cdot\mathbf{Q}=(O'\odot\rho^T)\cdot\tilde{\mathbf{P}}$. At this stage, all the tensors become real-valued. (d) A detailed description of each block is provided. The index for the site is denoted as α.

tensors:

$$\mathbf{P}^{(\alpha)}_{ljb}=\left(\delta_{b0}\delta_{l1}\delta_{j0}+\frac{1}{2}\delta_{b1}(\delta_{l0}\delta_{j0}+\delta_{l1}\delta_{j1})+\delta_{b2}\delta_{l0}\delta_{j1}\right)$$
$$=\begin{pmatrix}0&0&1&0\\1/2&0&0&1/2\\0&1&0&0\end{pmatrix}\quad(b\times lj\quad\text{matrix form}),\tag{A8}$$

and

$$\mathbf{R}^{(\alpha)}_{ljb}=(\delta_{b0}\delta_{l1}\delta_{j0}+\delta_{b1}(\delta_{l0}\delta_{j0}+\delta_{l1}\delta_{j1})+\delta_{kb}\delta_{l0}\delta_{j1})$$
$$=\begin{pmatrix}0&1&0\\0&0&1\\1&0&0\\0&1&0\end{pmatrix}\quad(lj\times b\quad\text{matrix form}).\tag{A9}$$

Using these tensors, we have

$$\sum_{l,j}M^{(\alpha)}_{lj}S^{*(\alpha)}(\boldsymbol{x})_l S^{(\alpha)}(\boldsymbol{x})_j$$
$$=\sum_{l,j,l',j',b}M^{(\alpha)}_{lj}\mathbf{R}^{(\alpha)}_{ljb}\mathbf{P}^{(\alpha)}_{bl'j'}S^{*(\alpha)}(\boldsymbol{x})_{l'}S^{(\alpha)}(\boldsymbol{x})_{j'}$$
$$=\sum_{l,j,b}M^{(\alpha)}_{lj}\mathbf{R}^{(\alpha)}_{ljb}\mathbf{B}^{(\alpha)}_{b},\tag{A10}$$

where $\mathbf{B}(\boldsymbol{x})$ is defined in Eqs. (3) and (4). This leads to an FLM representation that is compatible with Eq. (3),

$$f_Q(\boldsymbol{x};\boldsymbol{\theta},W_1,W_2,O)=(O'\odot\rho^T)(\boldsymbol{\theta})\cdot\mathbf{R}\cdot\mathbf{B}(\boldsymbol{x})$$
$$:=\boldsymbol{c}(\boldsymbol{\theta})\cdot\mathbf{B}(\boldsymbol{x}).\tag{A11}$$

We shall drop the α index when dealing with the N tensor product of α-indexed tensors, and $\cdot$ indicates the tensor contraction. See Fig. 9(b).

Lastly, as f_Q is a real-valued function, we can switch to real tensors using the identity

$$\mathbf{B}^{(\alpha)}=\mathbf{Q}^{(\alpha)}\mathbf{T}^{(\alpha)}\equiv\begin{pmatrix}0&1&-i\\1&0&0\\0&1&i\end{pmatrix}\begin{pmatrix}1\\\cos(\phi_\alpha(\boldsymbol{x}))\\\sin(\phi_\alpha(\boldsymbol{x}))\end{pmatrix}.\tag{A12}$$

By contracting $\mathbf{Q}$ to MPS part, we get MPS model using a trigonometric basis,

$$f_Q(\boldsymbol{x};\boldsymbol{\theta},O)=\sum_{\boldsymbol{k},\boldsymbol{b}}\tilde{M}^{(1)}_{b_1k_1}\tilde{M}^{(2)}_{b_2k_1k_2}\cdots\tilde{M}^{(N)}_{b_Nk_{N-1}}\mathbf{T}^{(1)}_{b_1}\cdots\mathbf{T}^{(N)}_{b_N}$$
$$:=\sum_{\boldsymbol{b}}\mathbf{C}^{\mathbf{q}}{}_{b_1b_2\cdots b_N}\mathbf{T}^{(1)}_{b_1}\cdots\mathbf{T}^{(N)}_{b_N}$$
$$:=\mathbf{C}^{\mathbf{q}}(\boldsymbol{\theta})\cdot\mathbf{T}(\boldsymbol{x}),\tag{A13}$$

where $\tilde{M}^{(\alpha)}_{b}=\sum_{i,j,b'}M^{(\alpha)}_{ij}\mathbf{R}^{(\alpha)}_{ijb'}\mathbf{Q}^{(\alpha)}_{b'b}:=M^{(\alpha)}_{ij}\tilde{\mathbf{P}}^{(\alpha)}_{ijb}$, and revived the $\boldsymbol{\theta}$ dependence on $\mathbf{C}^{\mathbf{q}}$. This proves the lemma 1. Indices k_j are called "bond indices" and range from 1 to some maximum numbers, which are called "bond dimensions."

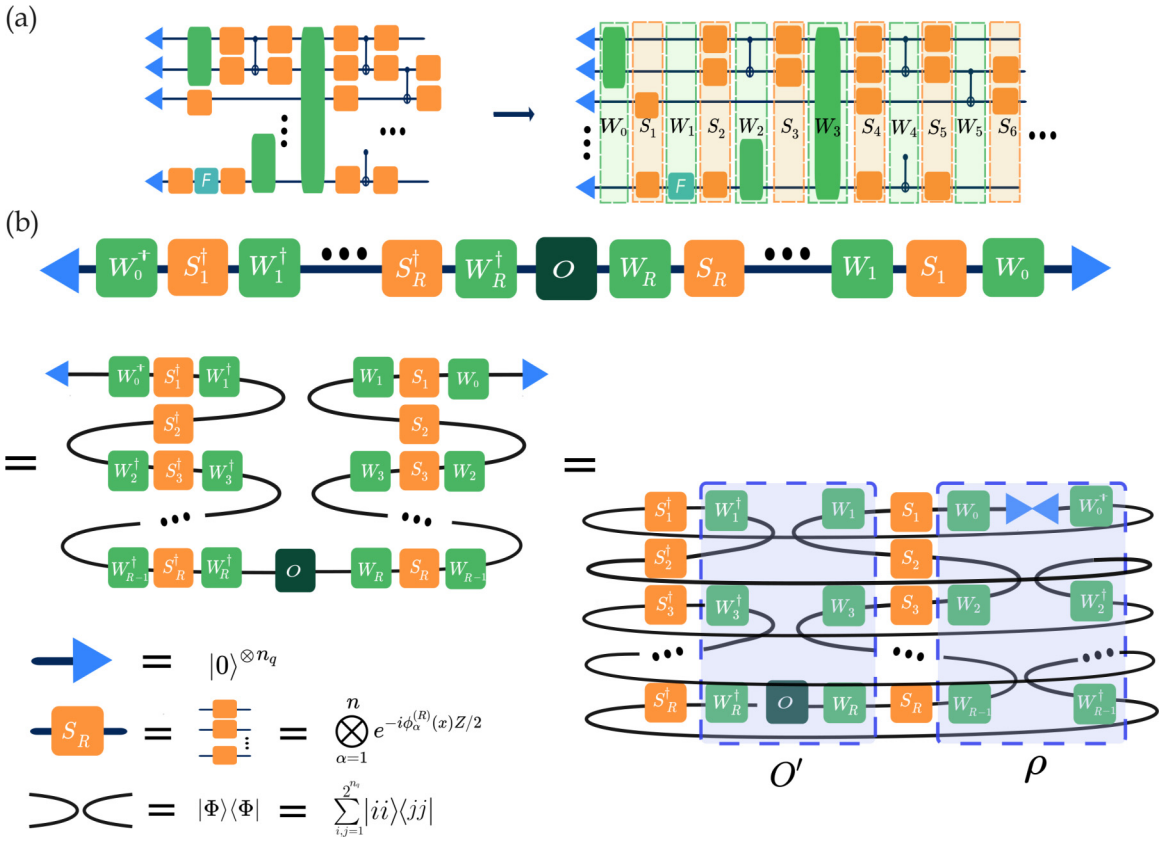

FIG. 10. Graphical description of changing the general structure model to MPS form. Orange squares are Pauli-Z rotations that are dependent on preprocessed input data, and the others are the coefficient part that does not depend on the input data. For simplicity, we omitted the data dependence and parameter dependence. (a) A quantum circuit with a general structure is segregated into an alternating encoding part and the coefficient part. As the diagram indicates, the coefficient parts $\mathbf{W}_k$s can contain nontrainable unitaries. (b) After adopting an unnormalized maximally entangled state $|\Phi\rangle\langle\Phi|$, we can transform it as if it were a simple parallel model. By replacing O' and ρ in the main text.

2. General case

A general encoding strategy can have data-encoding gates throughout the quantum circuit. We decompose all the encoding gates and discriminate the encoding part and trainable part as different layers, resulting in a data reuploading model that has an alternating structure of encoding parts $\mathcal{S}_k$s and trainable coefficient parts W_ks. Let our encoding gates be all decomposed and changed to Pauli-Z rotations so that it becomes R parallel encoding parts as Fig. 10(a). Then quantumly generated function $f_Q(\boldsymbol{x};\boldsymbol{\theta})$ of general model is

$$f_Q(\boldsymbol{x};\boldsymbol{\theta}) = \langle \mathbf{0}|W_0^\dagger(\boldsymbol{\theta}_0)\mathcal{S}_1^\dagger(\boldsymbol{x})W_1^\dagger(\boldsymbol{\theta}_1)\mathcal{S}_2^\dagger(\boldsymbol{x})\cdots\mathcal{S}_R^\dagger(x)W_R^\dagger(\boldsymbol{\theta}_R)OW_R(\boldsymbol{\theta}_R)\mathcal{S}_R(\boldsymbol{x})\cdots\mathcal{S}_2(\boldsymbol{x})W_1(\theta_1)\mathcal{S}_1(\boldsymbol{x})W_0(\boldsymbol{\theta}_0)|\mathbf{0}\rangle. \tag{A14}$$

Note that this model is general enough to encompass any VQML model that uses an encoding strategy. By bending wires, we can transform it into the simple parallel VQML model which we have treated in the main text. The graphical description is given in Fig. 10(b). For general encoding strategy, encoding block $\mathcal{S}_k$ can contain the identity operator, and this can be simply thought of as $\phi_\alpha^{(k)}(\boldsymbol{x}) = 0$. Now the function becomes the same form with Eq. (A5) in the main text,

$$f_Q(\boldsymbol{x};\theta) = \langle \mathcal{S}(\boldsymbol{x})|\left(O'_R \odot \rho_R^T\right)(\boldsymbol{\theta})|\mathcal{S}(\boldsymbol{x})\rangle, \tag{A15}$$

where O'_R and ρ_R is newly defined as

$$O'_R = \begin{cases} \bigotimes_{k=1}^{(R-1)/2}(W_{2k-1}^\dagger \otimes I)|\Phi\rangle\langle\Phi|(W_{2k-1}\otimes I)\otimes W_R^\dagger O W_R, & \text{if } R \text{ is odd} \\ \bigotimes_{k=1}^{R/2}(W_{2k-1}^\dagger \otimes I)|\Phi\rangle\langle\Phi|(W_{2k-1}\otimes I)\otimes O, & \text{if } R \text{ is even} \end{cases}, \tag{A16}$$

$$\rho_R = \begin{cases} W_0|0\rangle\langle 0|W_0^\dagger \otimes \bigotimes_{k=1}^{(R-1)/2}(I\otimes W_{2k}^\dagger)|\Phi\rangle\langle\Phi|(I\otimes W_{2k}), & \text{if } R \text{ is odd} \\ W_0^\dagger|0\rangle\langle 0|W_0 \otimes \bigotimes_{k=1}^{R/2}(I\otimes W_{2k}^\dagger)|\Phi\rangle\langle\Phi|(I\otimes W_{2k}), & \text{if } R \text{ is even} \end{cases}, \tag{A17}$$

$$|\mathcal{S}(\boldsymbol{x})\rangle = \bigotimes_{k=1}^{R}\bigotimes_{\alpha=1}^{n}\begin{pmatrix} e^{-i\phi_\alpha^R(\boldsymbol{x})/2} \\ e^{i\phi_\alpha^R(\boldsymbol{x})/2}\end{pmatrix}. \tag{A18}$$

Here $|\Phi\rangle = \sum_{i=1}^{2^n} |ii\rangle$, unnormalized maximally entangled state. Note that now O' and ρ are not real observable nor real state. Figure 10(b) only depicts when R is even, but one can picture an odd case analogously.

Although the general-structure model can be represented as a parallel model, there is a significant difference in terms of coefficients. In the general-structure model, the operators O' and ρ become $2^{n(R+1)}$-dimensional operators (2^{nR} for odd R cases). However, they cannot fully exploit the entire space of the given dimensional operator space, as the available free parameters are significantly fewer than what is required for a $2^{n(R+1)}$ (2^{nR} for odd R) dimensional operator space, even if universal unitary *ansatze* are used for all $\{W_k\}_k$. Consequently, the general-structure model possesses a smaller function space compared to the parallel model when they are basis-equivalent.

APPENDIX B: COMMENTS ON COEFFICIENTS

In the main text, we adopted tensors

$$\tilde{\mathbf{P}} = \bigotimes_{\alpha=1}^{N} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & i \\ 0 & 1 & -i \\ 1 & 0 & 0 \end{pmatrix}, \tag{B1}$$

which is a product of $\mathbf{R}$ and $\mathbf{Q}$, to generate MPS from the circuit *ansatz*. This might look a little odd at first sight, but when you write down the 2×2 Hermitian matrix M with the Pauli matrix basis as $M^{(\alpha)} = \lambda_0^{(\alpha)} I + \lambda_1^{(\alpha)} X + \lambda_2^{(\alpha)} Y + \lambda_3^{(\alpha)} Z$, then

$$M^{(\alpha)} \mathbf{R}^{(\alpha)} \mathbf{Q}^{(\alpha)} = M^{(\alpha)} \tilde{\mathbf{P}}^{(\alpha)} = \tilde{M}^{(\alpha)} = 2 \begin{pmatrix} \lambda_0^{(\alpha)} \\ \lambda_1^{(\alpha)} \\ \lambda_2^{(\alpha)} \end{pmatrix}. \tag{B2}$$

Therefore $\tilde{\mathbf{P}}$ discards the Pauli-Z coefficients of $M^{(\alpha)}$s, and multiply 2 to rest of coefficients. Let us represent $O' \odot \rho^T$ with Pauli string basis

$$O' \odot \rho^T = \sum_{\boldsymbol{i}} \lambda_{\boldsymbol{i}} \, \sigma_{i_1}^{(1)} \otimes \sigma_{i_2}^{(2)} \otimes \cdots \otimes \sigma_{i_N}^{(N)}, \tag{B3}$$

where $\boldsymbol{i} \in \{0, 1, 2, 3\}^{\otimes N}$. From the observation above,

$$(O' \odot \rho^T) \cdot \tilde{\mathbf{P}} = 2^N \lambda_{\tilde{\boldsymbol{i}}}, \quad \tilde{\boldsymbol{i}} \in \{0, 1, 2\}^{\otimes N}. \tag{B4}$$

We see that the coefficient on feature map components $\mathbf{T}_{\tilde{\boldsymbol{i}}}(\boldsymbol{x})$ corresponds to the $2^N \lambda_{\tilde{\boldsymbol{i}}}$, which are the Pauli string coefficients of $O' \odot \rho^T$ except the Z containing components.

For instance, let us use exponential encoding on 1d input x, where $\phi_\alpha(x) = 3^{\alpha-1}x$ and $N = 3$ simple parallel model. One of the basis functions is $\cos(x)\sin(3x)\cos(9x)$. This is chosen by $\tilde{\boldsymbol{i}} = (1, 2, 1)$. Therefore the coefficient in the basis function $\cos(x)\sin(3x)\cos(9x)$ is the 2^N times coefficient on Pauli string $X \otimes Y \otimes X$ when $O' \odot \rho^T$ is represented in Pauli string basis. One might expect high-frequency terms to depend on Pauli strings which have many nonidentity elements. However, this is not true in general. Again in the same setting, $\sin(x)\sin(3x)$ depends on $Y \otimes Y \otimes I$ which has 2 nonidentity elements. On the other hand, $\cos(9x)$ depends on $I \otimes I \otimes X$ which has 1 nonidentity element but has a higher frequency.

This sheds light on the nature of the coefficients of VQML models, which have been somewhat opaque thus far. It allows us to understand the coefficients *via* knowledge about the operator spreading capability of trainable circuits in the context of Pauli string basis. To see how trainable *ansatz* choice affects the coefficient set, let us look at an example of a simple parallel model, using $W_1(\boldsymbol{\theta}_1) = \bigotimes_{\alpha=1}^{N} H$. Then $O' \odot \rho^T$ becomes just $\frac{1}{2^N} O'$, where $O' = W_2^\dagger(\boldsymbol{\theta}_2) O W_2(\boldsymbol{\theta}_2)$ is evolved operator. Next, we set $O = Z_{N-1}$ which is a local Pauli-Z operator, and consider two different cases of $W_2(\boldsymbol{\theta}_2)$s having $L = 1$. One is hardware-efficient *ansatz*, which we used throughout the main text [Fig. 11(a)] and the other is reversed-CNOT *ansatz* where the ordering of CNOT gates are reversed [Fig. 11(b)]. As CNOT gates spread the Z operator in the target qubit to the control qubit and arbitrary unitary changes Z into an arbitrary superposition of other Pauli matrices (except I that has nonzero trace), O' for the first case possibly possess nonzero $\lambda_{\boldsymbol{i}}$s for all $\boldsymbol{i} \in \{1, 2, 3\}^{\otimes N}$. In Eq. (B2), we saw that Pauli-Z coefficients do not play the role, so it can exploit at most 2^N components in $\{\mathbf{T}_{\boldsymbol{i}}(\boldsymbol{x})\}_{\boldsymbol{i} \in \{1,2\}^{\otimes N}}$. However, the second case cannot spread the local Z operator to full qubit space so one gets only 9 nonzero $\lambda_{\boldsymbol{i}}$s in O'. Consequently, the VQML model using the second circuit can only use at most 4 components out of 3^N $\mathbf{T}_{\boldsymbol{i}}$s.

Our $\mathbf{C^q}$ constructions, which utilize Pauli coefficients, capture the easiness of Clifford circuits. Clifford circuits are known to be efficiently simulable by classical computers. Therefore one might expect that the VQML model which consists of only Clifford gates cannot show any quantum advantage. Once again, let us set $W_1(\boldsymbol{\theta}_1) = \bigotimes_{\alpha=1}^{N} H$, but this time, we compose W_2 with Clifford gates and set

$$O = \mathcal{P} \in \bigotimes_{j=1}^{N} \sigma_j, \quad \sigma_j \in I, X, Y, Z. \tag{B5}$$

Clifford circuits merely permute the Pauli coefficients of a given operator, therefore O' simply becomes another Pauli string $\mathcal{P}'$, which contains only one nonzero element in its coefficient. This implies that our Clifford model uses just *one* basis function. If O had K nonzero Pauli coefficients, then O' could be expressed as the sum of K Pauli strings. This sum would generate an MPO, and consequently, a $\mathbf{C^q}$ with a bond dimension at most K. As a result, the function class from this Clifford model can be reproduced by a CMPS model with a bond dimension of at most K, which is efficient if K scales polynomially. Although the Clifford circuit, consisting of a finite gate set, doesn't fit into the VQML model, its analysis and the discussion of the previous paragraph hint at how we can link the magic of quantum circuits or operator spreading to the capabilities of VQML models.

APPENDIX C: MAXIMUM BOND DIMENSION SCALING OF SIMPLE PARALLEL MODELS

Two-qubit entangling gates such as CNOT gates can be represented by a matrix product operator (MPO) of bond dimension equal to 2. Contracting two MPOs of bond dimensions χ_1 and χ_2 results in an MPO of bond dimension at most $\chi_1 \chi_2$. Therefore the maximum bond dimension for O' and ρ^T scales at most exponentially with the number

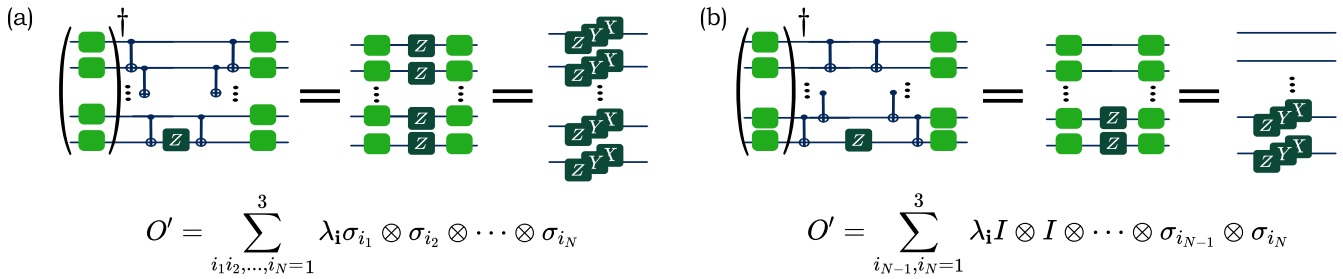

FIG. 11. Operator spreading of different *ansatze*. (a) Hadware-efficient *ansatz* used in the main text. This ansatz spread local Z operator on the last qubit to all N-qubit space with 1 layer. (b) Hadware-efficient *ansatz* but with the reversed ordering of CNOT gates. This *ansatz* cannot spread local operators to the whole space using only 1 layer. Resulting in only 9 Pauli string coefficients being nonzero.

of layers L_1 and L_2. Due to the inequality, $\mathrm{rank}(A \odot B) \leqslant \mathrm{rank}(A)\mathrm{rank}(B)$, maximum bond dimension for $O' \odot \rho^T$ is bounded by $\min(4^{\lfloor N/2 \rfloor}, 4^{L_1 L_2})$. The VQML coefficient vector $\mathbf{C^q}$ is obtained by contracting $\tilde{\mathbf{P}}$ to the $O' \odot \rho^T$, which shrinks the physical dimension from 4 to 3. Therefore χ_q can possibly scale as $\sim 3^{L_1 L_2}$, which is exponential with the depth of the VQML model.

To see how bond dimension scales with real circuit, we have used the circuit ansatz of Fig. 2(b). After the construction of $\mathbf{C^q}$s, we canonicalized them and extracted the number of nonzero elements in every bond indices. The maximum number of nonzero elements is then taken. Results are in Table I, all averaged over 24 different parameter sets. We set $L_1 = L_2 = L$. The average values of χ_qs scale with L as 4^L, and quickly saturate to the possible maximum value, $3^{\lfloor N/2 \rfloor}$. This shows that bond dimensions scale maximally in general, thereby coefficient sets of polydepth VQML models cannot be generated efficiently by the classical MPS, even on average.

APPENDIX D: DIFFERENT NUMBER OF LAYERS FOR TRAINABLE BLOCKS IN THE PARALLEL MODEL

In Fig. 12, we present a simulation result from the $N = 8$ parallel model. When the total number of layers $L_1 + L_2$ is the same, $S_2^{\max}$ is the largest around when $L_1 = L_2$. Here $S_2^{\max}$ is the largest Renyi-2 entanglement entropy of $\mathbf{C^q} = (O' \odot \rho^T) \cdot \tilde{\mathbf{P}}$. For this reason, we stick to setting all the number of trainable layers to be the same when there is no additional mention.

This is expected because

$$\mathrm{rank}(A \odot B) \leqslant \mathrm{rank}(A)\mathrm{rank}(B), \tag{D1}$$

so when one of the operators (O' or ρ^T) exhibits low rank—low entanglement entropy—Hadamard product of them cannot possess high entanglement entropy. A small number of layers implies low entanglement, thus one can expect that large $S_2^{\max}$ can be achievable when both numbers of layers are high enough.

TABLE I. Maximum bond dimension for $\mathbf{C^q}$: χ_q

N \ L	1	2	3	4	5	6	7	8	9	10
6	4	27	27	27	27	27	27	27	27	27
7	4	27	27	27	27	27	27	27	27	27
8	4	59	81	81	81	81	81	81	81	81
9	4	59	81	81	81	81	81	81	81	81
10	4	64	243	243	243	243	243	243	243	243
11	4	64	243	243	243	243	243	243	243	243

APPENDIX E: TRUNCATION ERRORS FOR C$^{\mathbf{q}}$s

Let us understand how truncation error $\eta(D)$—which gives us the upper bound of approximation error using $\chi_c = D$ CMPS model—changes with the system size for a given bond dimension bound $\chi_C = D$. Noiseless, $L = 10$ parallel VQML models are chosen as "hard" models which generate high and linearly scaling $S_2^{\max}$. For "easy" VQML models, we chose $\gamma = 0.15$, $L = 10$ noisy parallel models, and noiseless $L = 3$ models. After generating $\mathbf{C^q}$s respectively for each model, we truncated their singular values by leaving only D largest values, with $D = \{4, 8, 16, 32, 64\}$ and obtained truncation error $\eta(D)$.

As in Fig. 13, the noiseless model requires an exponential increase in D to achieve a fixed truncation error while increasing the model size. Therefore this simulation result indicates that to achieve a fixed approximation error bound using a CMPS while increasing the circuit size, one needs to increase χ_c exponentially. The noiseless shallow model exhibits a similar trend as the deep model, albeit with

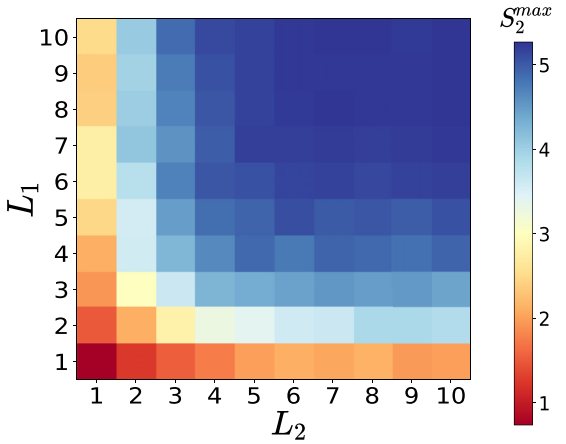

FIG. 12. $S_2^{\max}$ for $N = 8$ parallel VQML model. L_1 (L_2) is the number of layers in $W_1(\theta)$ ($W_2(\theta)$).

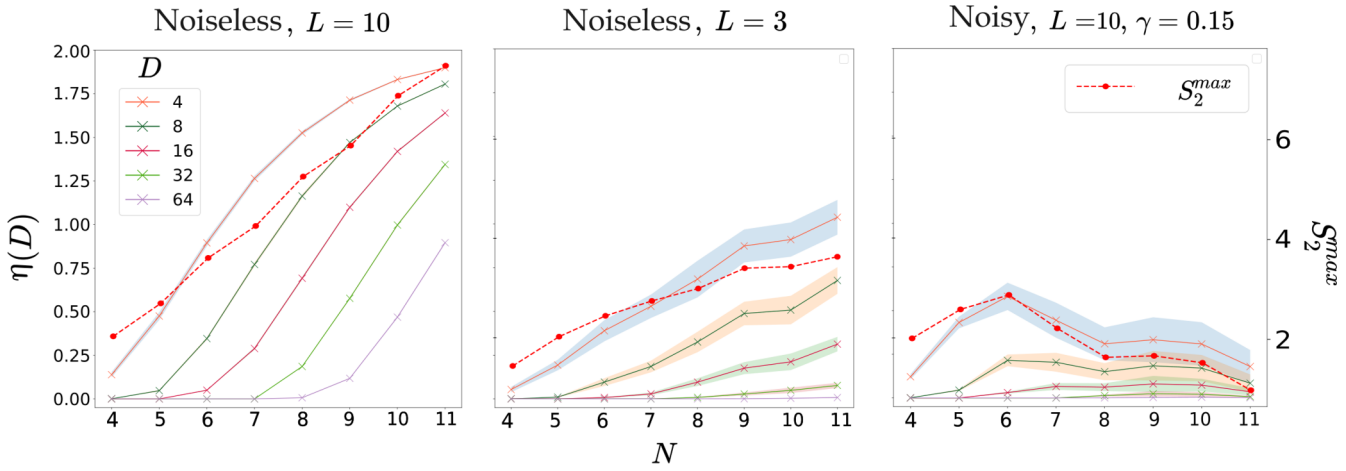

FIG. 13. The truncation errors for $\mathbf{C^q}$s when the maximum bond dimension is D, alongside the averaged $S_2^{\max}$ of original $\mathbf{C^q}$s (represented by red dashed lines), are presented. All values are averaged over 30 different randomly chosen parameter sets.

considerably smaller absolute values. For noisy cases, the truncation error decreases as the system size increases, and when $N = 10$ and 11, their $\chi_q = 243$ was exponentially large. However, $D = 2^5 = 32 \ll \chi_q = 243$ was sufficient for almost zero error bound.

These average truncation errors—and hence the approximation error bound—scalings align with the scaling of $S_2^{\max}$, as evidenced by the plotted $S_2^{\max}$ values. Moreover, note that for nearly zero error bound, $D = 32 \gg 2^{2.56} \sim 5.9$ for the noisy case and $D = 64 \gg 2^{3.63} \sim 12.3$ for the shallow case were enough, where 2.56 (3.63) is the maximum $S_2^{\max}$ for the noisy (shallow) case. These numerical results demonstrate that $S_2^{\max}$ can serve as a reasonable measure for evaluating the ease of approximating the $\mathbf{C^q}$s of VQML models.

APPENDIX F: NOISY CASE ANALYSIS

For the noisy case, we considered depolarizing error after every two-qubit gate operation. Kraus operators for this error model are given by

$$K^{(l)} \in \{\sqrt{1-\gamma 15/16} I \otimes I, \sqrt{\gamma/16} I \otimes X, \cdots, \sqrt{\gamma/16} Z \otimes Z\}. \tag{F1}$$

For our simulations, we used CNOT gates for two-qubit operations. Let us denote the state before applying one CNOT gate as σ. Then after applying CNOT gate and noise channel state becomes

$$\sigma' = \sum_{l=0}^{15} K^l U_{\text{CNOT}} \sigma U_{\text{CNOT}}^\dagger (K^l)^\dagger. \tag{F2}$$

Kraus operator can be understood as a rank-5 tensor where index l is for Kraus sum. We contracted Kraus tensor with U_{CNOT} tensor to construct a rank-5 noisy CNOT tensor. By replacing all the U_{CNOT}s in the circuit with a noisy version, and connecting all Kraus indices 1 to their corresponding conjugated part we could get the noisy version of $\mathbf{C^q}$s. See Fig. 14.

With understanding of coefficients of VQML models as Pauli coefficient, We can analyze how noise affects the coefficients of VQML using the Pauli path integral technique that is introduced in [25,26]. First, we observe that

$$\sum_l K^l (\sigma_i \otimes \sigma_j)(K^l)^\dagger = \begin{cases} (1-\gamma)(\sigma_i \otimes \sigma_j), & \text{if } i, j \neq 0 \\ I \otimes I, & \text{if } i = j = 0 \end{cases}. \tag{F3}$$

In other words, after the depolarizing channel E, all two-qubit Pauli operators attain $(1-\gamma)$ factor except the Identity. Let us denote observable after applying j (noisy) hardware-efficient *ansatze* as

$$O^{(j)} = \sum_{i_1 i_2, \ldots, i_N} \lambda^{(j)}_{i_1 i_2, \ldots, i_N} \sigma_{i_1} \otimes \cdots \otimes \sigma_{i_N}. \tag{F4}$$

Applying single-qubit unitary $U^{\otimes N}(\sigma_{i_1} \otimes \cdots \otimes \sigma_{i_N})(U^\dagger)^{\otimes N}$ mixes nonidentity Pauli matrices while leaving the identity unchanged. Therefore, after applying single-qubit unitaries in $(j+1)$'th layer, we get

$$\lambda'^{(j)}_{\boldsymbol{i}} = \mathcal{U}'_{\boldsymbol{i}\boldsymbol{i}'} \lambda^{(j)}_{\boldsymbol{i}'}, \tag{F5}$$

where $\boldsymbol{i} \equiv i_1 i_2, \ldots, i_N$, and $\mathcal{U}'$ is the representation of $U^{\otimes N}(\cdot)(U^\dagger)^{\otimes N}$ in the Pauli basis or Pauli transfer matrix (PTM) (we have omitted the θ dependence for simplicity). Note that $\mathcal{U}'$ can be block-diagonalized by simply permuting the order of indices. Next, we apply a CNOT gate. As CNOT gate is an inverse of itself, a CNOT gate on i_k, i_{k+1}-site qubits exchanges two coefficients in the set $\{\lambda'^{(j)}_{i_1,\ldots,i_k,i_{k+1},\ldots,i_N} | i_{l\neq k,k+1} \text{ are same}\}$. Here index-exchanging follows CNOT change rule which is depicted in Fig. 15(b). As a result, we get

$$\lambda^{(j+1)}_{\boldsymbol{i}} = \mathcal{U}_{\text{CNOT},\boldsymbol{i}\boldsymbol{l}'} \mathcal{U}'_{\boldsymbol{l}'\boldsymbol{i}'} \lambda'^{(j)}_{\boldsymbol{i}'} \equiv \mathcal{U}_{\boldsymbol{i}\boldsymbol{i}'} \lambda^{(j)}_{\boldsymbol{i}'}, \tag{F6}$$

where $\mathcal{U}$ denotes the PTM of noiseless one layer of hardware-efficient *ansatz*.

We apply the noisy channel E on $i_k i_{k+1}$-site qubits which introduces $(1-\gamma)$ factor if $i_k i_{k+1} \neq 00$. We do this start from 1,2-site qubits to $N-1, N$-site qubits resulting in

$$(1-\gamma)^{w_i} \times \lambda^{(j+1)}_{\boldsymbol{i}}. \tag{F7}$$

Here $w_{\boldsymbol{i}}$ is the number of non-00 (non-II) sequences in index-vector $\boldsymbol{i}$, or we call it *second-order Hamming weight*, which can range from 0 to $N-1$. For example, if $\boldsymbol{i} = 002300$, then

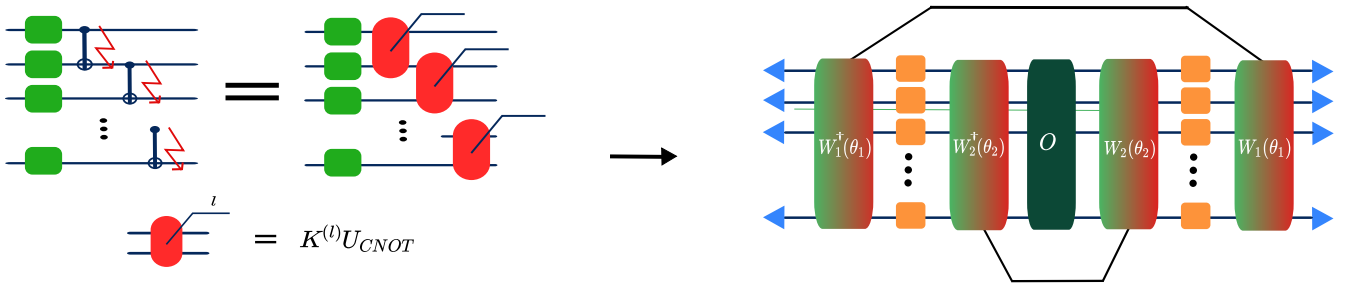

FIG. 14. Depolarizing noise acts on after every CNOT gate is applied. U_{CNOT} and Kraus tensor are contracted to create noisy CNOT tensors, denoted as red tensors. The full tensor diagram for the noisy quantum model has additional contraction lines for Kraus sum.

$w_i = 3$. Applying noisy layers from the beginning, the Pauli coefficient $\lambda_i^{(L)}$ after L-noisy layers is

$$\sum_{i_0 i_1 i_2 \ldots i_{L-1}} (1-\gamma)^{w_{i_1}+w_{i_2}+\ldots+w_{i_{L-1}}+w_i^{(L)}} f(i_0 i_1, \ldots, i_{L-1}; i)$$
$$\equiv \sum_{\vec{i}_{0:L-1}} (1-\gamma)^{|w|_{\vec{i}_{1:L-1}}+w_i} f(\vec{i}_{0:L-1}; i), \tag{F8}$$

where

$$f(i_0 i_1 \ldots i_{L-1}; i) = \mathcal{U}_{i i_{L-1}} \cdots \mathcal{U}_{i_2 i_1} \mathcal{U}_{i_1 i_0} \lambda_{i_0}^{(0)}. \tag{F9}$$

We call the sequence of index-vectors $\vec{i}_{0:L-1} \equiv (i_0, \ldots, i_{L-1})$s as Pauli path as named in Ref. [25], and $|w|_{\vec{i}_{1:L-1}} \equiv w_{i_1} + \cdots + w_{i_{L-1}}$ as total second-order Hamming weight of the Pauli path $\vec{i}_{1:L-1}$. Finally, coefficients on basis functions are obtained after the Hadamard product between noisy evolved O' and ρ^T. Hadamard product has a mixed product property which is $(A \otimes B) \odot (C \otimes D) = (A \odot C) \otimes (B \odot D)$ and the following product table for Pauli matrices.

$$\begin{aligned} &I \odot I^T = Z \odot Z^T = I, \\ &X \odot X^T = Y \odot Y^T = X, \\ &-X \odot Y^T = Y \odot X^T = Y, \\ &I \odot Z^T = Z \odot I^T = Z, \\ &\text{otherwise} = 0. \end{aligned} \tag{F10}$$

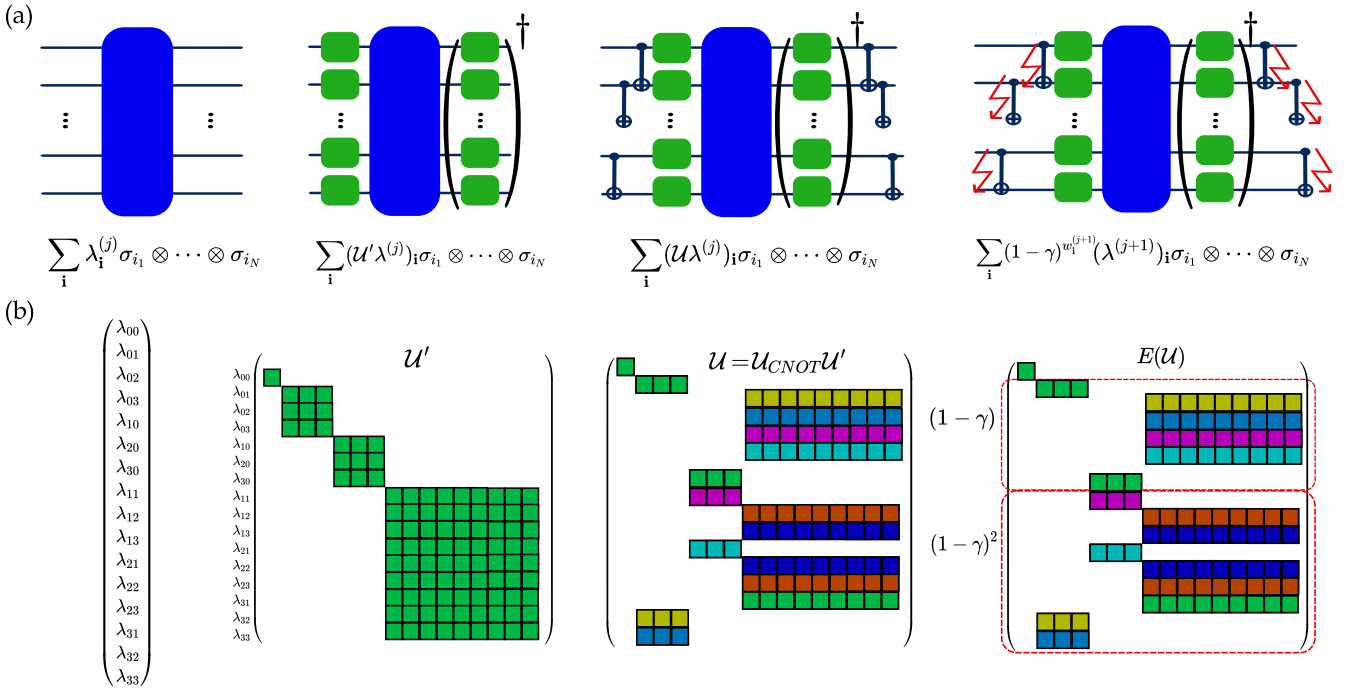

FIG. 15. (a) Applying one layer of noisy hardware-efficient *ansatz*. (b) This is a graphical depiction of the two-qubit case Pauli transfer matrix (PTM) for a (noisy) hardware-efficient ansatz. We reordered the Pauli coefficients, depicted as columns of λ_is, to expose the block-diagonal structure of the PTM for the single-qubit unitaries layer. The PTM of the CNOT gate permutes the order of rows. Permuted rows are denoted in the same color and nonpermuted rows are colored in green. Lastly, the multiplied noise factors are indicated. Note that λ_{00} is not affected by operations.

Therefore $\boldsymbol{k}$'th Pauli coefficient of $O' \odot \rho^T$ is

$$\lambda_{\boldsymbol{k}}^{O'\odot\rho^T} = \sum_{\boldsymbol{i}\odot\boldsymbol{j}=\boldsymbol{k}} \sum_{\vec{\boldsymbol{i}}_{0:L-1},\vec{\boldsymbol{j}}_{0:L-1}} (1-\gamma)^{|w|_{\vec{i}_{1:L-1}}+w_i} f(\vec{\boldsymbol{i}}_{L-1};\boldsymbol{i}) \times (1-\gamma)^{|w|_{\vec{j}_{1:L-1}}+w_j} f(\vec{\boldsymbol{j}}_{L-1};\boldsymbol{j}), \tag{F11}$$

where $\sum_{\boldsymbol{i}\odot\boldsymbol{j}=\boldsymbol{k}}$ denotes that summation over $\boldsymbol{i}$ and $\boldsymbol{j}$ satisfying the condition $\boldsymbol{i} \odot \boldsymbol{j} = \boldsymbol{k}$, which has 2^N combinations.

All Pauli paths except the $(\mathbf{0}, \mathbf{0}, \ldots, \mathbf{0})$ attain noise factors that depend on each paths. As a consequence, $O' \odot \rho^T$ converges to the identity, which becomes product MPS when converted to $\mathbf{C^q}$. We leave a more comprehensive analysis of the noisy case as future research.

APPENDIX G: PERFORMANCE OF VQML MODELS AND CMPS MODELS

We have seen that highly entangled coefficiennts make VQML models hard to dequantize. In other words, the unique power of VQML models comes from the ability to *efficiently* generate high-entangled coefficient MPS models using a small number of parameters. In this section, we compare the VQML models and CMPS models in the context of machine learning to explore the performance differences between two models.

All VQML models are simulated and generated classically using PYTHON PENNNYLANE package [62]. Tensor contractions and generation of CMPS models are done with the QUIMB package [63]. Optimization of all variational models is done by Adam optimizer with a learning rate 0.01 and 500 training epochs.

1. Property of coefficients and comparison on function regression

The number of trainable parameters is a crucial characteristic in machine learning models. Generally, the number of parameters is considered as a measure of the size of the model. More importantly, both models' computational complexities (contraction complexity for CMPS models and gate number complexity for VQML models) are polynomially related to it, thereby setting the same number of parameters enables the comparison between models that share similar computational complexity. Let us set the number of parameters be P, then a fundamental difference arises between the two models. In the CMPS model, $S_2^{\max} = O(\ln \chi_c) = O(\ln \sqrt{P/N})$. On the other hand, for a noiseless VQML model with $L \approx N$, we have $S_2^{\max} = O(N) = O(\sqrt{P})$ as observed in numerical simulations. Consequently, the CMPS model generates a coefficient set characterized by low entanglement and dense parameters, while the VQML model's coefficient set, $\mathbf{C^q}$, exhibits high entanglement and sparse parameters. A dense $\mathbf{C^c}$ can generate any coefficient tensor with maximum bond dimension χ_c, while a sparse $\mathbf{C^q}$ cannot create the majority of coefficient tensors possessing a maximum bond dimension of χ_q. We interpret this as an implicit "quantum" regularization, which is an efficient process for quantum machines but not for classical MPS models.

To see the performance of implicit "quantum" regularization in MPS models, we conduct a series of regression tasks to compare the performances of basis-equivalent CMPS and VQML models, each equipped with comparable parameter numbers. We set all conditions such as training data, training epoch, optimizer setting, etc., to be the same unless otherwise specified. For the CMPS model, we invoke regularized loss

$$\mathcal{L} = \frac{1}{M_t} \sum_i (f_c(\boldsymbol{x}_i;\boldsymbol{\theta}) - y_i)^2 + \lambda \|\mathbf{C^c}(\boldsymbol{\theta})\|_2^2, \tag{G1}$$

with regularization constants $\lambda \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 0\}$.

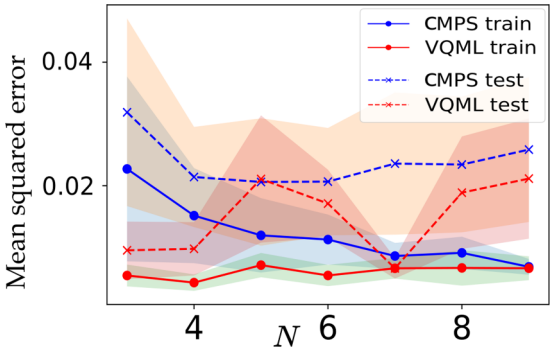

FIG. 16. The test and training losses (mean squared error) of the trained models with the MPS-generated labeled f-MNIST dataset are depicted. For the CMPS model, the regularization constant $\lambda = 0$ yields the best performance, and therefore, only results from this case are plotted. The results, which have been averaged over 10 different target coefficient instances, are accompanied by shaded regions representing a 0.95 confidence level.

2. Fashion-MNIST with MPS generated label

We choose the function regression task from the re-labeled f-MNIST dataset as done in Refs. [40,41]. Input data are the preprocessed fashion-MNIST data that have dimensions of $n \in [3, 9]$. Unlike the previous works, the target values are generated (re-labeled) with the $\chi = 3$ MPS model, so that

$$y_i = \frac{1}{K} \sum_{\boldsymbol{b}}^{3} \sum_{\boldsymbol{i}}^{3} M^{(1)}_{b_1,i_1} M^{(1)}_{b_2,i_1,i_2}, \cdots M^{(n)}_{b_n,i_{n-1}} \times \mathbf{T}^{(1)}(\boldsymbol{x}_i)_{b_1} \cdots \mathbf{T}^{(n)}(\boldsymbol{x}_i)_{b_n}. \tag{G2}$$

K is the $\max\{|y_i|\}_i$, normalization factor to set the target values lie within the $[-1, 1]$. Unlike the quantum circuit-generated case we used simple encoding $\phi_\alpha(\boldsymbol{x}) = x_\alpha$. That is, every element in the vector is encoded once by one Pauli-Z rotation. For the training CMPS model, we employ the same structure CMPS model as the target MPS model, which has $\chi_c = 3$. For CMPS models, all had $\chi_c = 3$, resulting in $[45, 72, 99, 126, 153, 180, 207]$ free parameters. In parallel, VQML models have $L \in [2, 3, 3, 3, 4, 4, 4,]$ resulting in $[36, 72, 90, 108, 168, 192, 216]$ free parameters. For CMPS model, $S_2^{\max} = \ln_2 3 \sim 1.7$, whereas the VQML model can have $S_2^{\max} > 2$.

Losses after training are plotted in Fig. 16. Interestingly, while this task is expected to be favored by CMPS models due to their structures being exactly the same as the target-generating MPSs, VQML models show slightly better performance than CMPS models.

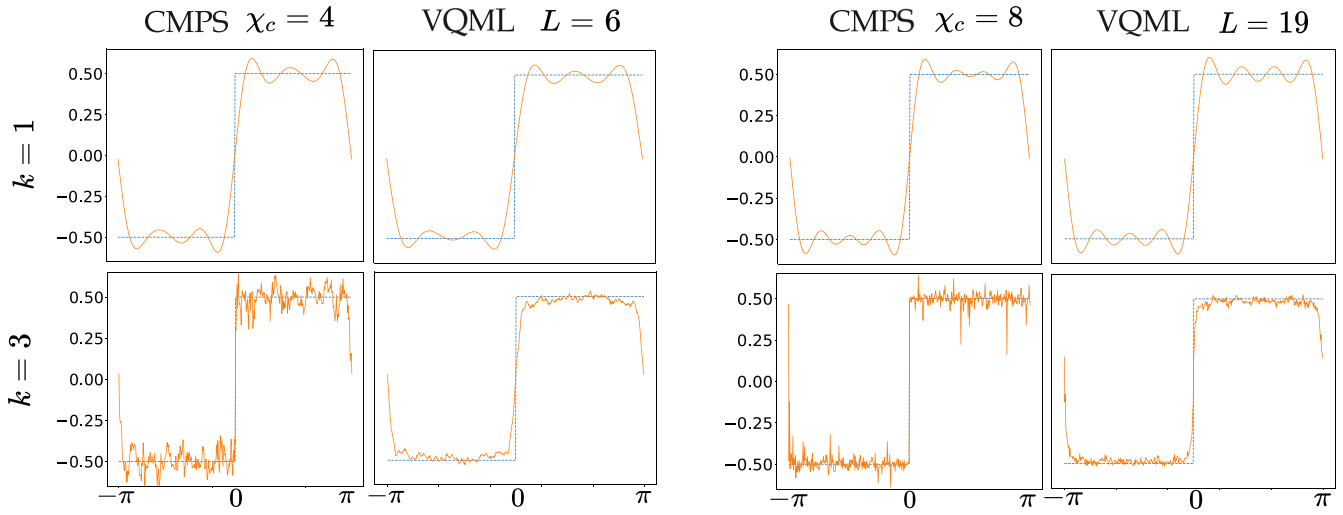

FIG. 17. A comparison on 1-D step function regression task. The preprocessing functions are given as $\phi_\alpha(x) : k^{\alpha-1}x$. For the CMPS models, we only plot the functions exhibiting the lowest test loss (mean squared error) across all regularizing constants, denoted as λ. Grouped models illustrate those with a comparable number of free parameters for coefficients.

3. Step function regression

The step function,

$$f_{\text{step}}(x) = \begin{cases} 1/2, & \text{if } x > 0 \\ -1/2, & \text{if } x \leqslant 0 \end{cases}, \tag{G3}$$

is an important function class as it can represent the target function of a classification task. We trained models with 400 randomly picked data and tested with 100 unseen data points. For preprocessing functions, we chose

$$\phi_\alpha(x) = k^{\alpha-1}x, \tag{G4}$$

where $k \in \{1, 3\}$. We compared CMPS models of $N = 8$ and parallel VQML models. The CMPS models have $\chi_c = 4$ (8), resulting in 282 (930) free parameters, whereas the VQML models have $L = 6$ (19) and 288 (912) free parameters. Outputs from trained models are shown in Fig. 17. The optimization settings are the same as the variational ridge regression of the re-labeled f-MNIST dataset.

First, when we use naive encoding ($k = 1$), both models show comparable performances. However, when the number of basis functions becomes large ($k = 3$), we can observe slight differences between them. It appears that CMPS models exhibit a stronger tendency to overfit, as indicated by highly spiky graphs. This overfitting behavior becomes more pronounced when χ_c increases. In our numerical study, the test loss of CMPS model increases from approximately 0.010 to 0.016, while that for VQML models decreases from about 0.013 to 0.009 as we increase the number of free parameters.

From above regressions, CMPS models seem to suffer overfitting problems more than VQML models, and this might come from the high expressivity of dense MPSs. These show primitive evidence of the advantage of using quantum regularization. We leave comparisons on more different tasks and analytic studies about the generalization ability of quantum regularization as a further research topic.

4. Comments on computational resources

We compared the models sharing a similar number of free parameters, so that they have similar computational complexity. However, in practice, real VQML models always accompany statistical error δ_q and require $O(1/\delta_q^2)$ number of shots, resulting in additional computational resources. Regarding this, we can allow more bond dimensions to CMPS models. To be concrete, an N-qubit, polydepth parallel VQML model uses $O(\text{poly}(N)/\delta_q^2)$ quantum gates while the basis-equivalent CMPS model uses $O(N\chi_c^2)$. Therefore χ_c can be $O(\text{poly}(N)/\delta_q)$ for polydepth quantum models with similar scaling of computational resources. If δ_q is exponentially small, then an exponentially large bond-dimensional CMPS model is allowed. In this case, VQML loses its advantage in expressivity.

5. Kernel ridge regression

For the dataset, we follow the same data preprocessing in Ref. [40]. Original fashion MNIST data is a 28×28-pixel image where pixel values range from 0 to 255. Images are associated with 10 labels. We normalized the pixel values to lie in [0,1] and rescaled them to have 0 mean values. Next, we did principal component analysis (PCA) to reduce the 28×28-dimensional vector to $n \in [3, 9]$-dimensional vector.

For the case of kernel method simulation in the main text, we generated the target values y_is using an IQP-encoding first circuit,

$$y_i = \langle 0|S_{\text{IQP}}^\dagger W^\dagger(\boldsymbol{\theta}_{\text{target}})(\boldsymbol{x}_i)Z_1 W(\boldsymbol{\theta}_{\text{target}})S_{\text{IQP}}(\boldsymbol{x}_i)|0\rangle, \tag{G5}$$

where $\boldsymbol{\theta}_{\text{target}}$ is randomly generated and $W(\boldsymbol{\theta}_{\text{target}})$ is consists of hardware-efficient *ansatz* with L layers. k is the normalization factor, which is a standard deviation of the training set of y_is. The number of layers $L \in [10, 7, 6, 5, 4, 4, 3]$, so that the free parameters in the target quantum circuit are about 90 parameters.

Quantum kernels are calculated using PYTHON PENNYLANE package [62]. We use PYTHON package KERNELRIDGE regression in the SCIKIT-LEARN package for kernel ridge regression. The regularization constant is set to be 0.01.

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

[2] V. Dunjko and P. Wittek, A non-review of quantum machine learning: trends and explorations, Quantum Views **4**, 32 (2020).

[3] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, J. Preskill, and J. R. McClean, Quantum advantage in learning from experiments, Science **376**, 1182 (2022).

[4] H.-Y. Huang, R. Kueng, and J. Preskill, Information-theoretic bounds on quantum advantage in machine learning, Phys. Rev. Lett. **126**, 190505 (2021).

[5] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Quantum machine learning: a classical perspective, Proc. R. Soc. A **474**, 20170551 (2018).

[6] S. Alvi, C. Bauer, and B. Nachman, Quantum anomaly detection for collider physics, J. High Energy Phys. 02 (2023) 220.

[7] S. Monaco, O. Kiss, A. Mandarino, S. Vallecorsa, and M. Grossi, Quantum phase detection generalization from marginal quantum neural network models, Phys. Rev. B **107**, L081105 (2023).

[8] L. Wei, H. Liu, J. Xu, L. Shi, Z. Shan, B. Zhao, and Y. Gao, Quantum machine learning in medical image analysis: A survey, Neurocomputing **525**, 42 (2023).

[9] M. Sajjan, J. Li, R. Selvarajan, S. H. Sureshbabu, S. S. Kale, R. Gupta, V. Singh, and S. Kais, Quantum machine learning for chemistry and physics, Chem. Soc. Rev. **51**, 6475 (2022).

[10] E. Tang, Quantum principal component analysis only achieves an exponential speedup because of its state preparation assumptions, Phys. Rev. Lett. **127**, 060503 (2021).

[11] E. Tang, Dequantizing algorithms to understand quantum advantage in machine learning, Nat. Rev. Phys. **4**, 692 (2022).

[12] E. Stoudenmire and D. J. Schwab, Supervised learning with tensor networks, Adv. Neural Inf. Process. Syst. **29**, 4799 (2016).

[13] S. Shin, Y. S. Teo, and H. Jeong, Exponential data encoding for quantum supervised learning, Phys. Rev. A **107**, 012422 (2023).

[14] E. Peters and M. Schuld, Generalization despite overfitting in quantum machine learning models, Quantum **7**, 1210 (2023).

[15] F. Verstraete and J. I. Cirac, Matrix product states represent ground states faithfully, Phys. Rev. B **73**, 094423 (2006).

[16] This classical-to-quantum encoding procedure is even necessary for some QML tasks that use the quantum state as its input. This is the case where quantum states are stored in the classical form and re-constructed on the quantum computer later using classical information. In fact, if there is no coherent quantum memory and channel, no QML can avoid the classical-to-quantum encoding process.

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

[18] 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 (London) **567**, 209 (2019).

[19] M. Schuld, R. Sweke, and J. J. Meyer, Effect of data encoding on the expressive power of variational quantum-machine-learning models, Phys. Rev. A **103**, 032430 (2021).

[20] M. Schuld and N. Killoran, Quantum machine learning in feature hilbert spaces, Phys. Rev. Lett. **122**, 040504 (2019).

[21] H. Pashayan, S. D. Bartlett, and D. Gross, From estimation of quantum probabilities to simulation of quantum circuits, Quantum **4**, 223 (2020).

[22] V. Khemani, A. Vishwanath, and D. A. Huse, Operator spreading and the emergence of dissipative hydrodynamics under unitary evolution with conservation laws, Phys. Rev. X **8**, 031057 (2018).

[23] S. Xu and B. Swingle, Accessing scrambling using matrix product operators, Nat. Phys. **16**, 199 (2020).

[24] L. Leone, S. F. E. Oliviero, and A. Hamma, Stabilizer rényi entropy, Phys. Rev. Lett. **128**, 050402 (2022).

[25] D. Aharonov, X. Gao, Z. Landau, Y. Liu, and U. Vazirani, A polynomial-time classical algorithm for noisy random circuit sampling, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023* (ACM, 2023), pp. 945–957.

[26] E. Fontana, M. S. Rudolph, R. Duncan, I. Rungger, and C. Cîrstoiu, Classical simulations of noisy variational quantum circuits, arXiv:2306.05400.

[27] *Foundations of Machine Learning,* 2nd ed. (MIT Press, Cambridge, MA, 2018).

[28] F. J. Schreiber, J. Eisert, and J. J. Meyer, Classical surrogates for quantum learning models, Phys. Rev. Lett. **131**, 100803 (2023).

[29] S. Jerbi, C. Gyurik, S. C. Marshall, R. Molteni, and V. Dunjko, Shadows of quantum machine learning, arXiv:2306.00061 [quant-ph].

[30] N. Schuch, M. M. Wolf, F. Verstraete, and J. I. Cirac, Entropy scaling and simulability by matrix product states, Phys. Rev. Lett. **100**, 030504 (2008).

[31] K. Noh, L. Jiang, and B. Fefferman, Efficient classical simulation of noisy random quantum circuits in one dimension, Quantum **4**, 318 (2020).

[32] H. Fujita, Y. O. Nakagawa, S. Sugiura, and M. Watanabe, Page curves for general interacting systems, J. High Energy Phys. 12 (2018) 112.

[33] B. Casas and A. Cervera-Lierta, Multidimensional fourier series with quantum circuits, Phys. Rev. A **107**, 062612 (2023).

[34] C. Oh, K. Noh, B. Fefferman, and L. Jiang, Classical simulation of lossy boson sampling using matrix product operators, Phys. Rev. A **104**, 022407 (2021).

[35] C. Oh, M. Liu, Y. Alexeev, B. Fefferman, and L. Jiang, Tensor network algorithm for simulating experimental gaussian boson sampling, arXiv:2306.03709 [quant-ph].

[36] M. Liu, C. Oh, J. Liu, L. Jiang, and Y. Alexeev, Simulating lossy gaussian boson sampling with matrix-product operators, Phys. Rev. A **108**, 052604 (2023).

[37] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Ann. Phys. **326**, 96 (2011).

[38] T. Hofmann, B. Schölkopf, and A. J. Smola, Kernel methods in machine learning, Ann. Stat. **36**, 1171 (2008).

[39] M. Schuld, Supervised quantum machine learning models are kernel methods, arXiv:2101.11020 [quant-ph].

[40] S. Jerbi, L. J. Fiderer, H. Poulsen Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, Quantum machine learning beyond kernel methods, Nat. Commun. **14**, 517 (2023).

[41] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, Power of data in quantum machine learning, Nat. Commun. **12**, 2631 (2021).

[42] A. Canatar, E. Peters, C. Pehlevan, S. M. Wild, and R. Shaydulin, Bandwidth enables generalization in quantum kernel models, arXiv:2206.06686 [quant-ph].

[43] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, Towards quantum machine learning with tensor networks, Quantum Sci. Technol. **4**, 024001 (2019).

[44] J. Y. Araz and M. Spannowsky, Classical versus quantum: Comparing tensor-network-based quantum circuits on large hadron collider data, Phys. Rev. A **106**, 062423 (2022).

[45] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, Expressive power of parametrized quantum circuits, Phys. Rev. Res. **2**, 033125 (2020).

[46] R. Haghshenas, J. Gray, A. C. Potter, and G. K.-L. Chan, Variational power of quantum circuit tensor networks, Phys. Rev. X **12**, 011047 (2022).

[47] H.-M. Rieser, F. Köster, and A. P. Raulf, Tensor networks for quantum machine learning, Proc. R. Soc. A **479**, 20230218 (2023).

[48] V. N. Vapnik, *Statistical Learning Theory* (Wiley-Interscience, Hoboken, NJ, 1998).

[49] M. C. Caro, E. Gil-Fuster, J. J. Meyer, J. Eisert, and R. Sweke, Encoding-dependent generalization bounds for parametrized quantum circuits, Quantum **5**, 582 (2021).

[50] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, K. Sharma, L. Cincio, and P. J. Coles, Generalization in quantum machine learning from few training data, Nat. Commun. **13**, 4919 (2022).

[51] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, The power of quantum neural networks, Nat. Comput. Sci. **1**, 403 (2021).

[52] A. Nahum, S. Vijay, and J. Haah, Operator spreading in random unitary circuits, Phys. Rev. X **8**, 021014 (2018).

[53] M. Cerezo, M. Larocca, D. García-Martín, N. L. Diaz, P. Braccia, E. Fontana, M. S. Rudolph, P. Bermejo, A. Ijaz, S. Thanasilp, E. R. Anschuetz, and Z. Holmes, Does provable absence of barren plateaus imply classical simulability? Or, why we need to rethink variational quantum computing, arXiv:2312.09121.

[54] M. Ballarin, S. Mangini, S. Montangero, C. Macchiavello, and R. Mengoni, Entanglement entropy production in Quantum Neural Networks, Quantum **7**, 1023 (2023).

[55] A. Novikov, M. Trofimov, and I. Oseledets, Exponential machines, arXiv:1605.03795 [stat.ML].

[56] J. Liu, S. Li, J. Zhang, and P. Zhang, Tensor networks for unsupervised machine learning, Phys. Rev. E **107**, L012103 (2023).

[57] S. Efthymiou, J. Hidary, and S. Leichenauer, Tensornetwork for machine learning, arXiv:1906.06329 [cs.LG].

[58] D. Liu, S.-J. Ran, P. Wittek, C. Peng, R. B. García, G. Su, and M. Lewenstein, Machine learning by unitary tensor network of hierarchical tree structure, New J. Phys. **21**, 073059 (2019).

[59] J. A. Reyes and E. M. Stoudenmire, Multi-scale tensor network architecture for machine learning, Mach. Learn.: Sci. Technol. **2**, 035036 (2021).

[60] Y. Liu, W.-J. Li, X. Zhang, M. Lewenstein, G. Su, and S.-J. Ran, Entanglement-based feature extraction by tensor network machine learning, Front. Appl. Math. Stat. **7**, 716044 (2021).

[61] R. A. Horn and C. R. Johnson, *Matrix Analysis* (Cambridge University Press, Cambridge, England, 2012).

[62] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi, J. M. Arrazola, U. Azad, S. Banning, C. Blank, T. R. Bromley, B. A. Cordier, J. Ceroni, A. Delgado, O. D. Matteo, A. Dusko *et al.*, Pennylane: Automatic differentiation of hybrid quantum-classical computations, arXiv:1811.04968 [quant-ph].

[63] J. Gray, quimb: a python library for quantum information and many-body calculations, J. Open Source Software **3**, 819 (2018).