Double Descent in Quantum Kernel Methods

Marie Kempkes,[1,2,*] Aroosa Ijaz,[3,4,5,†] Elies Gil-Fuster,[5,6] Carlos Bravo-Prieto,[5] Jakob Spiegelberg,[2] Evert van Nieuwenburg,[1] and Vedran Dunjko[1]

[1] *Leiden University, Niels Bohrweg 1, 2333 CA Leiden, Netherlands*

[2] *Volkswagen Group Innovation, Berliner Ring 2, 38440 Wolfsburg, Germany*

[3] *Department of Physics and Astronomy, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada*

[4] *Vector Institute, Toronto, Ontario, M5G 0C6, Canada*

[5] *Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany*

[6] *Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany*

(Received 12 February 2025; revised 9 May 2025; accepted 15 September 2025; published 16 January 2026)

The double descent phenomenon challenges traditional statistical learning theory by revealing scenarios where larger models do not necessarily lead to reduced performance on unseen data. While this counterintuitive behavior has been observed in a variety of classical machine learning models, particularly modern neural network architectures, it remains elusive within the context of quantum machine learning. In this work, we analytically demonstrate that linear regression models in quantum feature spaces can exhibit double descent behavior by drawing on insights from classical linear regression and random matrix theory. Additionally, our numerical experiments on quantum kernel methods across different real-world datasets and system sizes further confirm the existence of a test error peak, a characteristic feature of double descent. Our findings provide evidence that quantum models can operate in the modern, overparameterized regime without experiencing overfitting, potentially opening pathways to improved learning performance beyond traditional statistical learning theory.

DOI: 10.1103/cn64-gs6b

I. INTRODUCTION

Recent progress in machine learning (ML) has significantly enhanced data-driven insights across science and industry. At the same time, there is growing interest in understanding how quantum computers can offer improvements in solving computational problems in machine learning. This question is central within the field of quantum machine learning (QML), which seeks to harness quantum effects for learning from data [1–6]. Researchers have identified specific learning problems for which QML could potentially outperform its classical counterpart [7–12]. However, while similarities and differences between QML and classical ML are continually being elucidated [13–17], a critical distinction remains: classical ML has demonstrated large-scale success in real-world applications, whereas QML has yet to achieve comparable practical breakthroughs.

Understanding this gap requires deeper insight into the generalization properties of the models, i.e., their performance on unseen data. Traditional statistical learning theory suggests that increasing the so-called model complexity, typically quantified by measures such as the number of trainable parameters and the Vapnik-Chervonenkis (VC) dimension [18–20] in addition to other measures [21], often leads to poor generalization. In this context, complexity can include not only model-specific characteristics but also broader factors, such as the size of the training dataset. The behavior predicted by traditional learning theory is visualized as a U-shaped curve, where overly complex models tend to memorize training data rather than capturing meaningful patterns [Fig. 1(a)]. However, empirical evidence from deep learning over the past decade challenges this notion. Large neural networks often exhibit improved generalization at scale [22–27], contradicting the theoretically predicted trade-off between complexity and generalization [Fig. 1(b)]. A growing body of literature has sought to explain this surprising behavior [28–38], attributing it in part to the so-called *double descent* phenomenon.

*Contact author: marie.kempkes@volkswagen.de

†Contact author: a4ijaz@uwaterloo.ca

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

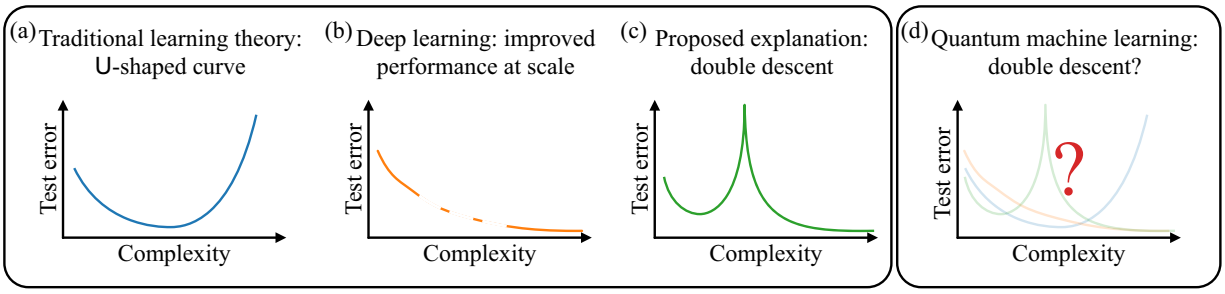


FIG. 1. Conceptual overview of this work. (a) Traditional statistical learning theory predicts a U-shaped relationship between model complexity and test error, where both overly simple and overly complex models have poor generalization. (b) Empirical observations in deep learning challenge this view, showing that larger models often exhibit improved performance at scale, although the precise relationship between complexity and error remains elusive. (c) The double descent phenomenon provides a theoretical and empirical explanation for this behavior, highlighting a non-monotonic relationship between model complexity and test error. (d) This work investigates whether a similar double descent behavior can emerge in quantum machine learning models, offering new insights into their generalization properties. The x axis represents a wide notion of complexity, including, for example, the size of the training set.

Double descent describes a non-monotonic relationship between test error and model complexity. Initially, the test error follows the expected U-shaped curve but then, counterintuitively, begins to decrease again as model complexity surpasses a critical threshold [Fig. 1(c)]. This behavior marks the transition from the underparameterized regime, where the model lacks sufficient capacity to represent the data, to the overparameterized regime, where the model has more parameters than necessary to fit the training dataset. Beyond this threshold, as observed empirically, the test error often decreases asymptotically until it saturates. It is important to note, however, that there is no guarantee of achieving saturation below the minimum of the U-shaped curve. These insights naturally provide motivation for the research question central to our work [Fig. 1(d)]: *Are there quantum machine learning models that exhibit double descent behavior?*

In this work, we provide a theoretical analysis of double descent for linear regression models in quantum feature spaces, prominently including quantum kernel methods (QKMs). By leveraging the fact that those models act linearly in their corresponding Hilbert space, we derive an expression for the test error, building on insights from Ref. [38]. Using the Marčenko-Pastur (MP) law, a seminal result in random matrix theory, we analytically demonstrate that in the asymptotic limit and under reasonable assumptions about the data distribution, the test error peaks at the interpolation threshold as the quantum model transitions from underparameterization to overparameterization, a key feature of the double descent phenomenon. We further test our theoretical findings with numerical experiments on quantum kernels, conducted on several datasets, including real-world data, for different system sizes and different families of quantum feature maps. Consequently, our results confirm that *certain quantum models exhibit the characteristic double descent behavior*.

To our knowledge, this work represents the first empirical and theoretical observation of double descent in quantum models. Our findings provide a foundational perspective on this phenomenon, paving the way for further exploration of its implications for model performance and generalization in quantum machine learning.

II. RELATED CONCEPTS IN QML

The double descent phenomenon challenges the validity of conclusions drawn from statistical learning theory, thereby undermining the relevance of generalization bounds derived from such a framework. In a similar vein, Gil-Fuster *et al.* [16] recently revealed flaws in the prevalent approach to studying generalization in QML, known as *uniform* generalization bounds. These bounds are uniform over the entire set of functions the learning model can output, regardless of the data distribution, the learning algorithm, or the specific function selected by the model. Earlier, Peters and Schuld [39] took a significant step away from uniform generalization bounds by showcasing instances of *benign overfitting* in QML. In particular, they identified scenarios where quantum learning models could fit the training data perfectly while still performing well on new test data, thus defying the traditional picture of statistical learning theory.

A well-known obstacle to successful QML at scale has been identified as exponential concentration of the functions output by the learning model [40]. While exponential concentration is often linked to problems in trainability [13], Thanasilp *et al.* [41] revealed that it can also adversely affect generalization. Specifically, exponential concentration may manifest itself in quantum feature maps suffering from a *vanishing similarity* problem [41], where the similarity measure between data points becomes exponentially small as the number of qubits increases. As a

result, the number of measurement shots required to reliably evaluate the labeling function scales exponentially, posing a significant hurdle for practical implementations. Our theoretical analysis diverges from this particular challenge. We focus on the exact evaluation of the functions, avoiding reliance on finite-shot approximations. This distinction places our approach on a different footing than the analysis presented in Ref. [41].

Lastly, we note that our work uses the standard notion of overparameterization from the classical machine learning literature. This choice contrasts with the alternative definition presented in Ref. [42], which follows a different framework. For further details on related and prior work, we refer the reader to Appendix A.

III. LINEAR REGRESSION IN QUANTUM FEATURE SPACE

In this section, we build upon the derivation presented in Ref. [38] to analyze the double descent phenomenon in QML. Specifically, we apply the framework from Ref. [38] to matrix-valued feature maps commonly encountered in QML. This allows us to explore whether and how the double descent behavior manifests itself in QML models.

In QML, classical data x are mapped into a quantum feature space via a quantum feature map, denoted as $\rho(x)$. This mapping is achieved via a unitary transformation:

$$x \mapsto \rho(x) = S(x)\rho_0 S^\dagger(x), \tag{1}$$

where $S(x)$ is a unitary operator parameterized by the data, and ρ_0 represents an initial n-qubit quantum state. A common approach to constructing QML models involves considering linear functions of the quantum feature state [14,43]. In particular, the model function is given by

$$f(x) = \mathrm{Tr}\rho(x)\mathcal{M}, \tag{2}$$

where $\mathcal{M} \in \mathrm{Herm}(2^n)$ is a quantum observable, which defines a hyperplane in quantum feature space. The function f can be evaluated as the expectation value of a quantum measurement specified by $\mathcal{M}$. An example of such quantum linear models is QKMs, in which a kernel function that quantifies the similarity between quantum feature states plays a central role in defining the model. The inner product between two quantum feature states is referred to as an "embedding quantum kernel" (EQK) function [44–47]:

$$\kappa_\rho(x,x') = \mathrm{Tr}\{\rho(x)\rho(x')\}. \tag{3}$$

EQKs can be used to find favorable observables $\mathcal{M}$ tailored to specific tasks.

In the context of supervised learning, we consider labeled data points $(x,y) \in \mathbb{R}^d \times \mathbb{R}$, where x represents the input and y the corresponding label. Given a training set $\{(x_i,y_i)\}_{i=1}^N$ of size N, our objective is to infer the underlying relationship between inputs and labels. We consider quantum linear models specified by a quantum feature map ρ using n qubits, and denote by p the number of free parameters in the model. In quantum kernel methods, the number of parameters is upper-bounded by the dimension of the orthonormal Hermitian basis spanned by ρ. Consequently, for embedding quantum kernels on n qubits, the number of parameters is at most $p = 4^n$. This remains valid for pure feature maps, provided that the image of ρ spans the full space of density matrices. For more details on the parameter count in quantum kernel methods we refer the reader to Appendix B. Note that we use the term "parameters" in a fundamentally different context than in, e.g., parameterized quantum circuits (PQCs), since the number of parameters here is fully specified by the data encoding map. In quantum feature space, the training data consist of quantum density matrices and labels $\{(\rho_i,y_i)\}_{i=1}^N$. We omit the explicit x dependence for simplicity and in order to make the framework applicable to linear regression with quantum states which do not arise from classical data. We introduce a data matrix D and label matrix Y as

$$D = \begin{pmatrix} [\rho_1]^\dagger \\ \vdots \\ [\rho_N]^\dagger \end{pmatrix}, \qquad Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}. \tag{4}$$

We use square bracket notation to highlight that D is *not* a matrix with each $\rho_i^\dagger$ as a submatrix. Instead, as detailed in Appendix C, D should be understood as a *vector of co-matrices*, where $\rho_i^\dagger$ represents the ith element. Since Hermitian matrices form a vector space, a co-matrix belongs to the dual space of this vector space and acts as a linear map from matrices to real numbers. This distinction becomes crucial later when expressions such as $D^\dagger D$ or $DD^\dagger$ are encountered. Indeed, $D^\dagger = ([\rho_1]\ldots[\rho_N])$ is a *co-vector of matrices*. With this notation, a "co-matrix acting on a matrix" is an inner product, $[\rho]_i^\dagger \rho_j = \mathrm{Tr}\{\rho_i\rho_j\}$, and a "matrix acting on a co-matrix" is a tensor product (as the canonical outer product in a vector space, in this case of Hermitian matrices), $\rho_i[\rho_j]^\dagger = \rho_i \otimes \rho_j$. Finally, the label vector $Y \in \mathbb{R}^N$ is a real-valued vector.

In linear least squares regression, the observable $\mathcal{M}$, sometimes referred to as a "parameter matrix," is determined by solving an optimization problem based on the given training data. Two regimes arise on the basis of relationship between the number of parameters p and the number of training samples N:

(1) Underparameterized regime ($N > p$): the number of data points exceeds the number of parameters. The observable $\mathcal{M}^u$ is obtained by solving a linear least

squares problem [48]:

$$\mathcal{M}^u = \arg\min_{\mathcal{M}} \|D\mathcal{M} - Y\|_2^2 = \left(D^\dagger D\right)^{-1} D^\dagger Y, \quad (5)$$

where we refer to the $(p \times p)$-dimensional $D^\dagger D = \sum_{i=1}^N \rho_i \otimes \rho_i$ as the sample covariance matrix (we slightly abuse notation by omitting centering and normalization). The matrix inverse is well-defined because $D^\dagger D$ is full rank in this regime.

(2) Overparameterized regime $(N < p)$: the number of parameters exceeds the number of data points. In this case, there is a continuum of linear functions that perfectly fit the data $M := \{\mathcal{M} \in \mathrm{Herm}(2^n) \mid \mathrm{Tr}\{\rho(x_i)\mathcal{M}\} = y_i, i \in [N]\}$. Among all these, the minimum-norm solution is selected [48]:

$$\mathcal{M}^o = \arg\min_{\mathcal{M}\in M} \|\mathcal{M}\|_2^2 = D^\dagger \left(DD^\dagger\right)^{-1} Y, \quad (6)$$

where $DD^\dagger = \left(\mathrm{Tr}\rho_i\rho_j\right)_{i,j=1}^N$ is the $N \times N$ Gram matrix, and its inverse is well-defined in this regime because $DD^\dagger$ is full rank. The condition $N = p$ is referred to as the "interpolation threshold," marking the transition where a model starts to interpolate (reaching nearly zero training error) and shifts from underparameterization to overparameterization.

We compare the models obtained in the two regimes with a hypothetical optimal linear model with corresponding $\mathcal{M}^*$, which achieves the best possible performance:

$$\mathcal{M}^* = \arg\min_{\mathcal{M}} \left\{ \mathop{\mathbb{E}}_{(\rho,y)} \left[(\mathrm{Tr}\rho\mathcal{M} - y)^2 \right] \right\}, \quad (7)$$

where the expectation value is taken over the underlying data distribution, which we assume to be unknown. The true relation between inputs and outputs, called "ground truth," need not be a deterministic linear function in feature space. That means $\mathcal{M}^*$ may still incur some error on individual samples, either due to noisy labels or because the best linear model in feature space cannot accurately capture the ground truth function, a situation known as model misspecification [35]. Note that in this context, minimizing the expected risk therefore does not guarantee good performance.

Consequently, when presented with a new test point (ρ_t, y_t) not present in the training set, neither the empirical risk minimizers $\mathcal{M}^{u,o}$ nor the expected risk minimizer $\mathcal{M}^*$ is guaranteed to predict the correct label, i.e., $y_t^{(u,o,*)} \neq y_t$. Here, $y_t^{(u,o,*)} := \mathrm{Tr}\rho_t\mathcal{M}^{(u,o,*)}$ denotes the predicted label on input ρ_t for the underparameterized empirical risk minimizer (u), the overparameterized empirical risk minimizer (o), and the optimal linear model $(*)$. Since the expected risk minimizer $\mathcal{M}^*$ is independent of the training dataset $\{(\rho_i, y_i)\}_{i=1}^N$, it can generally incur prediction errors on any training data point (ρ_i, y_i). Let us define this *residual* error as $e_i := y_i - y_i^*$, and denote the vector of residual errors by $E = (e_i)_{i=1}^N$.

We compare the predictions made by the empirical risk minimizers with those of the optimal linear model on unseen data:

$$y_t^u - y_t^* = \mathrm{Tr}\rho_t (D^\dagger D)^{-1} D^\dagger E, \quad (8)$$

$$y_t^o - y_t^* = \mathrm{Tr}\rho_t D^\dagger (DD^\dagger)^{-1} E \quad (9)$$

$$+ \mathrm{Tr}\rho_t \left(D^\dagger (DD^\dagger)^{-1} D - \mathbb{I}_{2^n\times 2^n} \right) \mathcal{M}^*. \quad (10)$$

A detailed step-by-step derivation of these expressions can be found in Appendix C.

Interestingly, the distinction between the two expressions (the inverse of the sample covariance matrix or of the Gram matrix) vanishes when we apply the singular value decomposition (SVD) of the data matrix $D = U\Sigma V^\dagger$. Here, $U \in \mathbb{U}(N)$ is a unitary matrix of left singular vectors, $\Sigma \in \mathbb{R}^{N\times(2^n\times 2^n)}$ is a rectangular diagonal matrix of singular values, and $V \in \mathbb{U}(2^n \times 2^n)$ is a unitary matrix of right singular vectors. Leveraging the properties of SVD, we can rewrite Eqs. (8) and (9) as

$$y_t^u - y_t^* = \mathrm{Tr}\rho_t V\Sigma^+ U^\dagger E, \quad (11)$$

$$y_t^o - y_t^* = \mathrm{Tr}\rho_t V\Sigma^+ U^\dagger E \quad (12)$$

$$+ \mathrm{Tr}\rho_t \left(D^\dagger (DD^\dagger)^{-1} D - \mathbb{I}_{2^n\times 2^n} \right) \mathcal{M}^*. \quad (13)$$

Here, Σ^+ denotes the pseudoinverse of Σ. Notably, the underparameterized and overparameterized cases both share a common term, $\mathrm{Tr}\rho_t V\Sigma^+ U^\dagger E$. The second term in Eq. (13) is unique to the overparameterized regime and accounts for the underdetermination of the linear optimization problem.

Note that the shared term corresponds to a variance-like contribution and the second term corresponds to a bias-like contribution to the test error. Importantly, we compare the learned solution with the optimal linear model for the given training data, rather than with the ground truth. As a result, these contributions differ from the classical definitions of bias and variance in statistical learning theory. We refer the reader to Appendix C for more details.

Several classical works have shown that the variance-like term is responsible for the double descent behavior [32,33,35,36,38,49,50]. All these works consistently demonstrate that the model's sensitivity to fluctuations in the training data is maximal at interpolation, causing the variance-like term to dominate the test error. We further show in our ablation study as reported in Fig. 3(c) that the contribution of the bias-like term to the test error remains small for quantum kernels, and hence does not explain the peak. Our subsequent analysis of the double descent

behavior thus focuses exclusively on the variance-like contribution.

To unpack the term further, we first expand it in the orthonormal basis of singular vectors:

$$\mathrm{Tr}\rho_t V\Sigma^+ U^\dagger E = \sum_{r=1}^{R} \frac{1}{\sigma_r} \mathrm{Tr}\rho_r^V \rho_t \langle u_r, E\rangle, \tag{14}$$

where $R = \min\{N, 2^n \times 2^n\}$ denotes the rank of D, and the right singular vectors ρ_r^V are quantum density matrices. This formula highlights three distinct factors that influence the prediction error:

(1) The reciprocals of the singular values $1/\sigma_r$.
(2) The interaction of the test input ρ_t with the basis of right singular vectors $\mathrm{Tr}\rho_r^V \rho_t$.
(3) The projection of E onto the left singular vectors $\langle u_r, E\rangle$.

The error decomposition indicates that for double descent to occur, the three contributing factors must become large enough to cause a spike in the prediction error on new test data.

IV. SPECTRAL ANALYSIS WITH RANDOM MATRIX THEORY

In this section, we present analytical results that predict a peak in the prediction error at the interpolation threshold $N = p$ for the problem of linear regression in quantum feature space introduced in Sec. III. Our analysis builds on the error decomposition derived in the previous section, which explicitly identified three factors contributing to the prediction error in quantum linear models, as specified in Eq. (14). Among the three factors, we now focus specifically on the reciprocal singular values of the data matrix D. Under certain conditions, D behaves as a random matrix whose singular values follow a MP law. This enables predictions about when the test error is likely to peak. Below, we present a simplified version of the relevant theorem and proof. Full proof details can be found in Appendix D.

Theorem 1 (Test error peak at interpolation—informal). Consider a Lipschitz-continuous quantum feature map ρ with p linearly independent dimensions, and consider a linear regression problem in the corresponding feature space. Let $\{(\rho(x_i), y_i)\}_{i=1}^N$ be a training set, where $x_i \sim \mathcal{N}(0, \mathbb{I}_d)$ are d-dimensional independent and identically distributed (i.i.d.) Gaussian normal random samples. Then, the test error of the linear model based on $\rho(x)$ and Eqs. (5) and (6) peaks with high probability at $N = p$ for large enough N and p.

Furthermore, for a quantum feature map on n qubits, the training set size N at which the peak in test error occurs fulfills $N \in \mathcal{O}(\exp(n))$ with high probability.

Proof sketch. The proof relies on the error decomposition introduced in Sec. III and results from random matrix theory. Namely, we use the generalized Marčenko-Pastur law, as derived in Ref. [51]. This law implies that for a Lipschitz-continuous feature map applied to Gaussian normal random vectors, the spectral behavior of the sample covariance matrix is the same as that of the sample covariance matrix constructed from a data matrix of Gaussian random entries. In particular, the MP law characterizes the range and probability density of eigenvalues of the sample covariance matrix as a function of system parameters N and p. As $N, p \to \infty$ at some constant rate, the MP law predicts that the number of non-zero eigenvalues is minimized when $N = p$. Since the eigenvalues of the sample covariance matrix correspond to the squared singular values of the data matrix, right hand side of Eq. (14) reaches its maximum when these singular values σ_r are smallest. Consequently, the test error is maximized with high probability at $N = p$ as $N, p \to \infty$.

Additionally, the number of linearly independent dimensions of any n-qubit quantum feature map cannot exceed the linear dimension of the set of 2^n-dimensional Hermitian matrices, which is 4^n. ■

Theorem 1 suggests that in the asymptotic limit, one might observe the phenomenon of double descent in quantum kernel methods if the data distribution is suitably aligned. Two key questions remain to ensure these results are broadly applicable: (1) Are typical quantum feature maps indeed Lipschitz continuous? (2) Does the assumption of i.i.d. Gaussian inputs reasonably model realistic data sources? First, we establish that a general class of commonly used quantum feature maps is Lipschitz continuous.

Theorem 2 (Lipschitz continuity). Let $x \in \mathbb{R}^d$. Let $\rho(x) = S(x)\rho_0 S^\dagger(x)$, where $S(x) = \prod_{l=1}^{L}\prod_{k=1}^{d} S_{lk}(x_k)$, with $S_{lk}(\alpha) = \exp\left(-\frac{i}{2}\alpha H_{lk}\right)$ and $\lambda = \max_{k,l}\|H_{lk}\|_{\mathcal{H}}$. Then, the map $x \mapsto \rho(x)$ is Lipschitz continuous with respect to the operator norm with a Lipschitz constant upper-bounded by $\sqrt{dL}\lambda$:

$$\|\rho(x) - \rho(x')\|_{\mathcal{H}} \leq \sqrt{dL}\lambda\|x - x'\|. \tag{15}$$

Proof sketch. The Lipschitz constant is derived by bounding the derivatives of $\rho(x)$, which in turn depend on the derivatives of $S(x)$. The nested product structure of $S(x)$ facilitates explicit computation of these derivatives. Full proof details are provided in Appendix E. ■

Second, while the assumption of i.i.d. Gaussian inputs is under active study, there is broad agreement on its plausibility for linear models in large-dimensional settings (large p) [52–55]. For instance, Ref. [56] provides an intuitive justification using *generative adversarial network*

(GAN) models [57]. GANs generate remarkably realistic images by using Gaussian random noise as input. Under the assumption that GANs accurately replicate real-world data, it follows that real-life data can be approximated by Gaussian noise mapped through a Lipschitz-continuous transformation. Since Theorem 2 establishes Lipschitz continuity for a broad class of quantum feature maps, the corresponding results from classical theory naturally extend to quantum models using these maps. Accordingly, a (Q)ML model applied to such data can be conceptualized as a three-step process: (1) random Gaussian inputs, (2) hypothetical GAN-like feature map, and (3) the actual (Q)ML model. This perspective, further formalized in Ref. [53], supports the application of Theorem 1 to the empirical setting explored in Sec. V. Hence, Theorem 1 can be broadly applied to predict a peak in test error in quantum linear models when the training set size N matches the feature map dimension $p = 4^n$.

V. EMPIRICAL EVIDENCE OF DOUBLE DESCENT IN QUANTUM KERNEL METHODS

The theoretical results presented in the previous section hold only in the asymptotic limit and rely on specific assumptions about the input data distribution. In this section, we validate our theory by providing empirical evidence of the double descent phenomenon in quantum kernel methods in the finite regime and with real-world data. To our knowledge, this is the first empirical demonstration of double descent in QML.

In our experiments, we consider QML models based on EQKs, as introduced in Eq. (3). We note that the formalism laid out in Ref. [38], which we extended to quantum feature maps in Sec. III, closely resembles the well-established framework of QKMs [14]. Specifically, the solution to the linear regression problem based on least squares [Eq. (5)] and minimum norm [Eq. (6)] can be recovered with kernel ridgeless regression [58]. Here, the term "ridgeless" denotes the absence of a regularization term in the conventional kernel regression formulation. Therefore, implementing ridgeless kernel regression with the EQK is mathematically equivalent to finding the optimal observables $\mathcal{M}^o$ and $\mathcal{M}^u$ as presented in Sec. III. While this connection to regression is clear, we also extend our experiments to classification tasks to further validate the theory.

We perform a series of numerical experiments on three different datasets. The first dataset is synthetic, with dimensionality matching the number of qubits n. The target function is defined as $g(x) = \langle w, x\rangle$ with fixed custom weights w and data points x sampled uniformly from $[-\pi/2, \pi/2]^n$. Additionally, we used the Fashion MNIST dataset in a binary classification setting by choosing two categories. Lastly, for regression analysis on real-world data, we used the California Housing dataset. Both these datasets are accessible through Ref. [59]. The quantum circuit simulations were performed with the PennyLane [60] software library running on classical hardware. The quantum kernel was implemented via an embedding feature map as defined in Eq. (3), where each input x is encoded into a quantum circuit comprising single-qubit rotations and entangling gates, ensuring the resulting Gram matrix at interpolation is full rank. For further implementation details, refer to Appendix F or the public code repository [61].

To identify the interpolation threshold across different scenarios, we fixed the system size n and varied the number of training samples N. Note that in our setting, the degree of parameterization can be determined by the ratio of the number of training data to the dimension of the feature space. Consequently, varying the number of training samples is somewhat equivalent to varying the feature space dimension. The test error curves, depicted in Fig. 2, exhibit a clear double descent peak. Notably, a sharp peak is observed at interpolation ($N/p = 1$), followed by a characteristic drop in test error in the overparameterized regime ($N/p < 1$). This pattern is observed consistently across all datasets and system sizes, even though the data are far from Gaussian, as required by Theorem 1. This empirical observation suggests that the conclusions drawn under the Gaussianity assumption remain robust in practice, as discussed in Sec. IV. These findings provide compelling empirical evidence for the presence of double descent in QKMs and highlight its importance as a defining feature of their performance.

VI. ABLATION STUDY

To further solidify that the three factors outlined in Eq. (14) directly influence the double descent in QKMs, we conduct a series of numerical experiments where we artificially ablate one factor at a time, inspired by the method in Ref. [38]. The results are presented in Fig. 3.

Singular values: The factor $1/\sigma_r$ captures how the test error scales inversely with the singular values. In each experimental run, after sampling a training dataset of vectorized encoded states, we pre-processed the data by applying an explicit cutoff. Specifically, singular values below a specified threshold were removed from the singular matrix of the sampled data. As shown in Fig. 3(a), smaller cutoff values result in a more pronounced peak, emphasizing the role of small singular values in amplifying the test error.

Test features: The factor $\mathrm{Tr}\rho_r^V \rho_t$ quantifies the overlap between the quantum feature space, spanned by the right singular vectors, and the test data features. To isolate this contribution, we pre-processed the test dataset by projecting it onto a subset of (leading) singular modes derived from the underlying distribution. This procedure provides a controlled mechanism to reduce the overlap between ρ_t and the set $\{\rho_r^V\}$. Only the terms where projected test

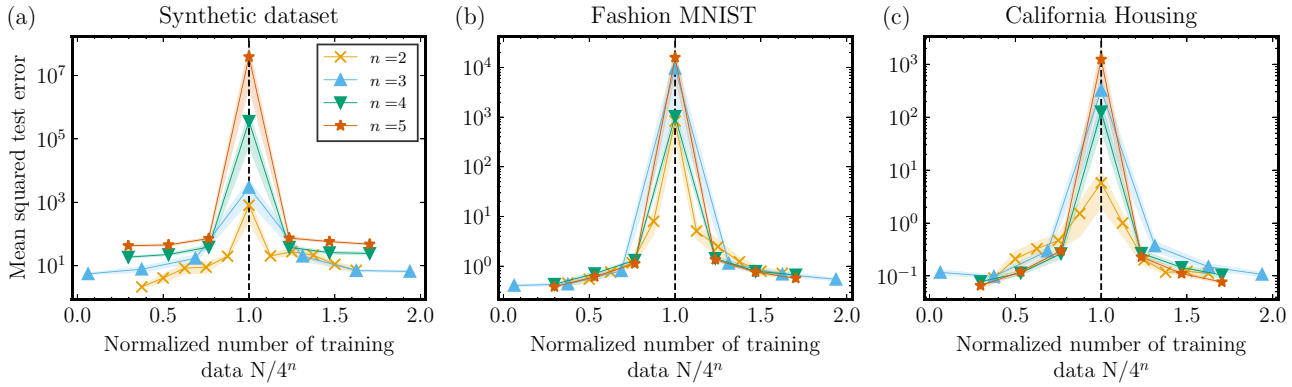


FIG. 2. Empirical evidence of double descent in QKMs. Mean squared test error as a function of the normalized number of training data points ($N/4^n$) for the (a) synthetic, (b) Fashion MNIST, and (c) California Housing datasets, with use of the EQK in Eq. (3). The shaded area corresponds to the standard deviation for five independent experiment repetitions, each using independently sampled training data. The dotted black line indicates the interpolation threshold. The region to the left of the threshold corresponds to the overparameterized regime, while the region to the right corresponds to the underparameterized regime. The consistent peak across all datasets and system sizes n confirms the presence of double descent.

features have overlap with the training features contribute non-zero terms in the sum in Eq. (14). As illustrated in Fig. 3(b), retaining more modes in the pre-processing subset emphasizes the double descent peak.

Residual error: If the best model within the hypothesis class incurs no residual error on the training data, the term $\langle u_r, E\rangle$ vanishes, effectively eliminating the double descent peak. To test this, we explicitly used the underlying distribution in the fitting process. The results, shown in Fig. 3(c), demonstrate that under these conditions, the model perfectly fits the test data at interpolation and in the underparameterized regime, causing the test error to drop to zero. This is in agreement with the error decomposition in Eq. (13), where the second term appears only in the overparameterized regime. Note that this does not imply that the underparameterized regime is preferable in general. It is preferable only in settings where the ground truth model is known to lie within the hypothesis class (and the data are noise-free), which is rarely the case in realistic scenarios.

These ablation experiments collectively demonstrate how all the three factors interact to produce the observed double descent behavior. Intuitively, when test features have a significant presence in the feature dimensions that are poorly represented by the training dataset (corresponding to small singular values of the training data), the model must extrapolate to fit these underrepresented dimensions [36]. This extrapolation results in higher test error. Taken together, these results provide a clearer understanding of double descent in quantum models and emphasize the

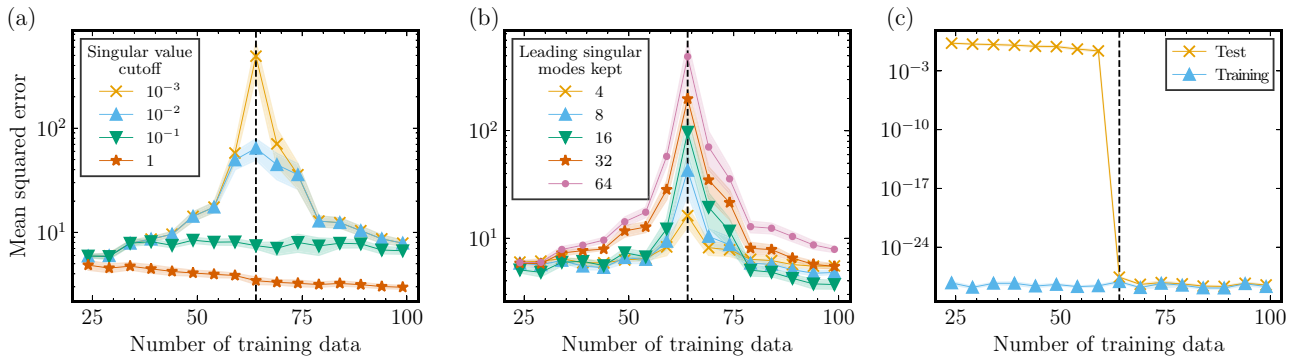


FIG. 3. Ablation experiments. Mean squared test error as a function of the number of training data points N for (a) application of different cutoffs on the minimum singular value, (b) retention of different numbers of leading singular modes of the input data, and (c) elimination of residual error. All experiments were performed on the synthetic dataset with $n = 3$ qubits. The shaded area corresponds to the standard deviation for five independent experiment repetitions, each using independently sampled training data. The dotted black line indicates the interpolation threshold. The region to the left of the threshold corresponds to the overparameterized regime, while the region to the right corresponds to the underparameterized regime. Modification of each factor reduces or eliminates the double descent peak, highlighting their roles in the test error behavior.

importance of carefully managing these factors during model design and training.

VII. THE CASE OF PROJECTED QUANTUM KERNELS

A natural question arises: Can we control the location of the interpolation threshold in our experiments with QKMs? So far, we have successfully observed a peak in prediction error at the interpolation threshold, $p = N$, both analytically and empirically. However, the quantum feature map we used in Sec. V resulted in an exponential-dimensional feature space with respect to the number of qubits n, specifically $p = 4^n$. As a result, while our experiments provide evidence of double descent in quantum kernel methods, the peak in error occurs only for exponentially large training sets. We now introduce an alternative family of feature maps, referred to as "projected feature maps," whose feature space dimension (and hence the number of free parameters) is linear in the number of qubits $p \in \mathcal{O}(n)$. Note that the exact value of p for projected quantum kernels depends on the specific kernel design (here it is $3n + 2$). By using these maps, we can effectively shift the interpolation threshold and facilitate a more controlled and more efficient double descent setting.

To construct a projected quantum feature state, we begin with a conventional quantum feature map ρ. This state is then transformed into an n-dimensional vector of single-qubit reduced density matrices (RDMs). This transformation is achieved with the use of partial trace operations, also known as "projections":

$$\rho_{\mathrm{RDM}}(x) = \left(\rho_{(k)}(x)\right)_{k=1}^{n} = \left(\mathrm{Tr}_{j\neq k}\{\rho(x)\}\right)_{k=1}^{n}. \tag{16}$$

Here, $\rho_{(k)}(x)$ represents the RDM corresponding to the kth qubit. Importantly, $\rho_{\mathrm{RDM}}(x)$ forms a *vector of matrices*, and we define their inner product as

$$\langle \rho_{\mathrm{RDM}}(x), \rho_{\mathrm{RDM}}(x')\rangle = \frac{1}{n}\sum_{k=1}^{n} \mathrm{Tr}\{\rho_{(k)}(x)\rho_{(k)}(x')\}. \tag{17}$$

Indeed, such projected feature maps also give rise to EQK functions on the Hilbert space of reduced density matrices $\kappa_{\mathrm{RDM}}(x, x') = \langle \rho_{\mathrm{RDM}}(x), \rho_{\mathrm{RDM}}(x')\rangle$. We refer to this kernel as the "RDM kernel." Similar constructions were explored in Refs. [62,63].

In this section, we apply results from Secs. IV and V to the RDM kernel. Our primary focus lies in understanding the training set size at which the peak in test error occurs in this setting, both theoretically and empirically. We begin by extending Theorem 1 to apply to projected quantum feature maps.

Corollary 1 (Test error peak for projected feature maps). Consider a quantum kernel model based on the RDM projection of an L-Lipschitz-continuous quantum feature map $\rho(x)$ on n qubits. Let $\{x_i\}_{i=1}^{N}$ be a training set of d-dimensional i.i.d. Gaussian random samples $x_i \sim \mathcal{N}(0, \mathbb{I}_d)$. Then, the training set size N at which the peak in test error occurs with high probability is linear in the number of qubits; $N \in \mathcal{O}(n)$.

Proof. To establish this result, we prove that (1) the projected feature map is Lipschitz continuous, and (2) the number of linearly independent dimensions p satisfies $p \in \mathcal{O}(n)$. For (1), the Lipschitz continuity of the projected feature map follows directly from the fact that the partial trace operation is Lipschitz continuous and the fact that the composition of Lipschitz-continuous maps is also Lipschitz continuous. For (2), consider the linear dimension of the set of one-qubit states. The RDM kernel function defined in Eq. (17) is an inner product in a $4n$-dimensional complex vector space:

$$\kappa_{\mathrm{RDM}}(x, x') = \frac{1}{n}\sum_{k=1}^{n} \mathrm{Tr}\{\rho_{(k)}(x)\rho_{(k)}(x')\} \tag{18}$$

$$= \sum_{k=1}^{n}\sum_{i,j\in\{0,1\}} \frac{\left[\rho_{(k)}(x)\right]_{i,j}}{\sqrt{n}} \frac{\left[\rho_{(k)}(x')\right]^{*}_{i,j}}{\sqrt{n}}. \tag{19}$$

Although the summation contains $4n$ terms, not all terms are independent. The unit-trace property of each reduced density matrix removes approximately n terms. Consequently, the number of independent real components scales as $p \in \mathcal{O}(n)$, and so the interpolation threshold appears at $N \in \mathcal{O}(n)$. ■

We repeat our numerical experiments using the RDM kernel; the results are presented in Fig. 4. Once again, we observe the characteristic double descent peak, now arising from projected quantum feature states. These results can be directly compared with the results in Fig. 2, which correspond to the general embedding quantum feature map. Notably, the training set size N at which the test error peak occurs now scales linearly with the number of qubits n when the RDM kernel is used. This linear behavior contrasts sharply with the exponential scaling reported in Sec. V. These empirical findings strongly align with the theoretical predictions.

VIII. DISCUSSION

The double descent phenomenon is a leading concept in understanding the success of deep learning, demonstrating how large models can outperform predictions from traditional statistical learning theory. In this work, we demonstrate that quantum kernel methods can also exhibit double descent. We present analytical results derived from a combination of linear regression insights and random matrix theory to understand the occurrence of the characteristic

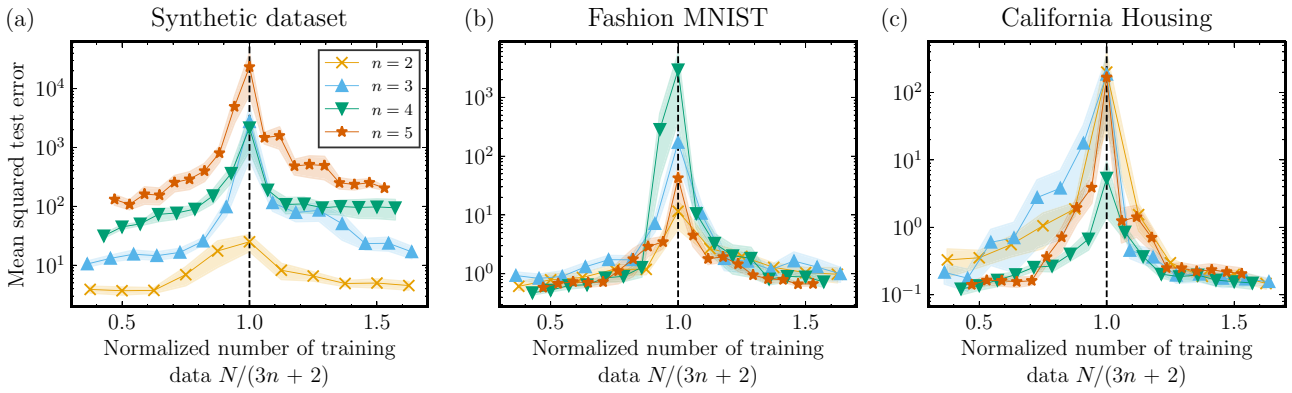


FIG. 4. Empirical evidence of double descent in RDM projected quantum kernels. Mean squared test error as a function of the normalized number of training data points $[N/(3n + 2)]$ for the (a) synthetic, (b) Fashion MNIST, and (c) California Housing datasets, with use of the RDM kernel in Eq. (19). The shaded area corresponds to the standard deviation for five independent experiment repetitions, each using independently sampled training data. The dotted black line indicates the interpolation threshold. The region to the left of the threshold corresponds to the overparameterized regime, while the region to the right corresponds to the underparameterized regime. The consistent peak across all datasets and system sizes n confirms the presence of double descent.

double descent behavior. Furthermore, we test our theory in extensive numerical experiments on synthetic and real-world datasets. To our knowledge, this is the first study to identify this phenomenon in QML, marking a significant step forward in understanding the behavior of overparameterized quantum models. We now discuss the implications of these results and propose directions for future research.

Our findings contribute to the ongoing developments in QML by aligning with and expanding the scope of existing studies. To begin with, our work is in line with the insights from Ref. [16], which advocates a paradigm shift in assessing the performance of QML models, moving beyond conventional statistical learning theory. Moreover, Ref. [39] presents a tailored setting of large PQC-based QML models that show improved test errors compared with statistical learning theory predictions. By introducing the double descent formalism, our work provides a complete complementary perspective, supported by results for common models on real-world data. Furthermore, although not directly contradicting the findings in Ref. [64], our demonstration of double descent in quantum kernel methods provides a complementary perspective to those findings, which highlight that *quantum classifiers undergo a* U*-shaped risk curve, in contrast to the double descent risk curve of deep neural classifiers*. In our understanding, the conclusions in Ref. [64] may stem from specific scenarios where the model fails to fit the training set adequately at scale, resulting in large test errors.

The framework presented applies to a broad class of linear models in quantum feature space, including certain quantum kernel methods [14], as well as specific extreme learning machines [65,66]. An important open question is whether similar double descent dynamics emerge in PQCs, including those involving data re-uploading [67]. Recent work [68] suggests that PQCs do not always converge to the minimum-norm solution in the overparameterized regime. Thus, further investigation is required to adapt the double descent framework to PQCs, a direction we leave for future work.

Another important aspect to discuss is the connection of our work to the concentration phenomenon observed in quantum kernel methods [41]. Our work does not solve concentration. Indeed, exponentially concentrated kernels with flat spectra will yield poor generalization in practice regardless. It is however interesting to understand the test performance expected from quantum kernels that do not suffer from this phenomenon. For such viable, non-exponentially-concentrated kernels, our analysis predicts the potential for a double descent curve (Fig. 1), suggesting that test error may not always increase monotonically with complexity. Although this work focuses on the peak, the practical benefit hinges on achieving a deeper second dip in test error. Notably, a promising observation is that in classical machine learning, the second dip is usually deeper [26,27].

On a broader level, our work establishes a foundational understanding of double descent in QML models, laying the groundwork for further exploration into the overparameterized regime and its implications for learning performance in quantum machine learning.

ACKNOWLEDGMENTS

The authors thank Zoë Holmes, Evan Peters, Franz Schreiber, and Supanut Thanasilp for useful comments on a previous version of the manuscript. A.I. thanks Juan Carrasquilla, Hsin-Yuan Huang, Evan Peters, and Jason Rocks for helpful discussions. A.I. also acknowledges support from the Natural Sciences and Engineering Research Council, the Shared Hierarchical Academic Research

Computing Network, Compute Canada, and the Canadian Institute for Advanced Research (CIFAR) AI Chairs program. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. E.G.-F. is a 2023 Google PhD Fellowship recipient and acknowledges support by the Einstein Foundation (Einstein Research Unit on Quantum Devices), BMBF (Hybrid), and BMWK (EniQmA), as well as travel support from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 951847. C.B.-P. acknowledges support by the BMBF (MUNIQC-Atoms). E.v.N. acknowledges support from the Dutch National Growth Fund, as part of the Quantum Delta NL program. V.D. acknowledges support from the Dutch Research Council (NWO), as part of the Quantum Software Consortium program (Project No. 024.003.03) and co-funded by the European Union (ERC Consolidator Grant, BeMAIQuantum, Grant No. 101124342).

The results, opinions, and conclusions expressed in this publication are not necessarily those of Volkswagen Aktiengesellschaft or those of the European Union or the European Research Council. None of the parties involved can be held responsible for them.

DATA AVAILABILITY

The data that support the findings of this article are openly available [61].

APPENDIX A: PRIOR AND RELATED WORK

In this appendix, we recap existing works that study the generalization performance of large quantum models and analyze how our work connects with them. We also clarify various definitions of overparameterization that are used in the QML literature. To avoid confusion, we first list the various types of QML models generally considered in the supervised QML literature: (1) Implicit quantum models are linear models that output $\langle 0| U^\dagger(x)\mathcal{M}U(x)|0\rangle$, where $U(x)$ is the fixed encoding unitary. Observable $\mathcal{M}$ is a linear combination of the encoded training data, whose coefficients can be solved for classically. (2) Explicit quantum models are linear models that output $\langle 0| U^\dagger(x)\mathcal{M}U(x)|0\rangle$, where $U(x)$ is the fixed encoding unitary. Observable $\mathcal{M}$ is given by $V^\dagger(\theta)OV(\theta)$, where $V(\theta)$ represents parameterized unitaries and O is a fixed observable. The circuit parameters are generally optimized classically via gradient-based methods. (3) Data re-uploading models are linear models that output $\langle 0| W^\dagger(x,\theta)OW(x,\theta)|0\rangle$, where $W(x,\theta) = \prod_{l=1}^{L} V_l(\theta)U_l(x)$ is composed of alternating (and potentially distinct) encoding and variational layers. The encoding part consists of fixed gates and the trainable parameters are optimized classically. In general, the encoding unitary in any of these models can be made variational as well. This is especially useful in metric and representation learning applications [69].

Various machine learning phenomena that connect model performance to model complexity include overfitting, benign overfitting, and double descent. Overfitting [21] occurs when a learning model performs well on the training data but fails to perform well on unseen test data. It can generally be attributed to high model complexity, although excessive data noise or excessive training can also lead to it. "Benign overfitting" refers to the phenomenon where interpolating models (with nearly zero training error) can still generalize effectively [49,50,70]. It is generally not concerned with when and how the interpolation regime begins and what are the necessary conditions to achieve it. In contrast, double descent [27] captures the non-monotonic behavior of the test error as model complexity varies, with the test error decreasing after an initial peak. Importantly, double descent is not inherently tied to or caused by interpolation; it can occur even in models that do not achieve zero training error. Furthermore, the point at which test error begins to descend for the second time is not strictly determined by the location of the interpolation threshold [37].

Overfitting in QML models has been studied in several works using the conventional bias-variance trade-off argument, although these works have mostly been limited to data re-uploading and explicit quantum models [39,64,71–75]. Various strategies have been proposed to overcome overfitting in these models, including qubit dropout [76], gate dropout [77,78], and parameter pruning [79–81].

To our knowledge, our work is the first empirical demonstration of the double descent phenomenon in QML. The question of why double descent has not been previously observed for quantum models remains open. One possible explanation is that QML is mirroring the historical development of classical ML. With the onset of the use of heavily parameterized models and high-dimensional data, large classical models did not always exhibit benign overfitting, nor were double descent and an interpolation peak consistently observed. Recent developments show that this could be due to how generalization performance was measured [82] and how bias and variance were conventionally defined [36], hinting that the traditional bias-variance trade-off argument based on model complexity breaks down in the highly overparameterized regime [35].

It is possible that the QML literature is following a similar path. Some recent works have highlighted that most generalization bounds derived in the QML literature become vacuous in the overparameterized regime [16,64]. For example, Chen *et al.* [73] studied overfitting in parameterized quantum circuits, showing that the expressivity and VC dimension of these quantum circuits saturate with increasing circuit depth. However, it has been shown in both classical [37] and quantum [16] settings

that VC dimension is not a reliable measure to study generalization when scaling model complexity. Another example is Ref. [64], in which it is proposed that large QML models cannot achieve low training errors, while, in contrast, Gil-Fuster *et al.* [16] showed that some quantum models can perfectly fit training data even with label noise and Peters and Schuld [39] presented scenarios where large quantum models can perform well and display benign overfitting. We look at Refs. [39,64] in more detail below. The takeaway message of these recent developments is that there is a need to redefine how we connect model complexity to generalization as we scale the data dimension and model complexity, which we leave to follow-up work.

The benign overfitting (or harmless interpolation) phenomenon was investigated in explicit quantum models in Ref. [39] using spiking models, which is an established technique in the classical ML literature. To perfectly fit training points while allowing for general prediction, this technique allows the predictor function to have a "spiked" localized behavior near training points while being smooth in other areas of the input domain. By use of the Fourier representation of the functions in the hypothesis class, trigonometric polynomial interpolation is used to formulate the interpolation problem. Hence, overparameterization occurs when the number of Fourier modes of the learning model exceeds the number of Fourier modes of the data-generating distribution. Peters and Schuld [39] first show how the feature weights of an overparameterized Fourier features model can be engineered to have benign overfitting for linear regression. Then, they extend the classical results by using the Fourier representation of quantum models [83]. This allows them to design quantum models with similar features by tuning the initial quantum states and data-encoding Hamiltonians. Hence, Ref. [39] provides important tools for designing and working with overparameterized quantum models in the harmless interpolation regime. While the emphasis of the analysis of Peters and Schuld [39] is not on the precise location of the interpolation threshold, signatures of double descent are evident in their numerical experiments (Fig. 3). They use synthetic data and design a specific feature map that ensures the appropriate spiking behavior. In turn, in our work, we studied the factors that contribute to the peak at interpolation, relying on random matrix theory and the spectral properties of the data matrix to explain it. Consequently, our framework works for real-life data, with a non-restrictive generic quantum feature map. We leave a thorough study of the similarities between the underlying generalization mechanisms for future work.

Reference [64] studies overfitting and the potential presence of double descent in classification with quantum models. The study authors use explicit quantum models as quantum classifiers and define overparameterization as the regime when the size of the training dataset exceeds the number of data-independent parameterized quantum gates and the circuit unitary becomes a 2-design. By considering arbitrary circuit *Ansätzes* (without problem-informed structure), where overparameterized circuits become 2-designs, training is likely to suffer from barren plateaus [40]. This will result in a U-shaped curve for the total expected error (training error and generalization error), as poor trainability will lead to larger training errors (and hence poor generalization error) as more parameters or gates are added. Hence, Du *et al.* [64] propose using underparameterized problem-dependent *Ansätzes* instead. They reinforce this via numerical studies conducted using hardware-efficient *Ansatz* on the Fashion MNIST dataset, which fails to reach low training errors due to poor trainability. This particular setting and the corresponding poor performance of explicit quantum models can also be studied from the generalization perspective (independent of the trainability perspective). Along these lines, Peters and Schuld [39] showed that using states sampled uniformly from the Haar measure will have concentrated feature weights for most known encoding strategies (see their Appendix B.1). This will give rise to quantum models that will not achieve benign overfitting when they are overparameterized. However, we note that these results should not rule out the possibility of double descent in all problem-agnostic quantum models (classifiers or otherwise), especially those that do not suffer from barren plateaus, such as quantum convolutional neural networks. We leave a complete formal study of quantum convolutional neural networks for follow-up work.

Du *et al.* [64] also propose that in order to reach zero training errors, parameterized quantum classifiers should optimize parameters such that the optimal feature states and measurements form a general simplex equiangular tight frame (ETF). ETFs have been shown to arise in deep classical neural networks as they reach zero training error (when trained for long times), leading to "neural collapse" (small variability in the state representation of training data that belong to the same class) [84]. However, we note that while ETFs and neural collapse (on training data) can mean low training errors, it does not necessarily imply low generalization errors and may even lead to worse performance on test data, as was recently reported in Ref. [85]. So, neural collapse is an optimization phenomenon, not a generalization one. This could explain the numerical results in Ref. [64], for which they used hardware-efficient *Ansatz* to do binary classification on the parity dataset (Fig. 2 and Fig. J9 in Appendix J), where only harmful overfitting is observed for the overparameterized classifier. Hence, while Ref. [64] studies an important aspect of how deep quantum models might behave and represent the data during training, there is a need to connect this to phenomena such as benign overfitting and double descent, which we leave for follow-up work.

Other definitions of overparameterization have also been reported in the QML literature. As an important clarification, we reiterate that Haug *et al.* [80] and Larocca *et al.* [42] use tools from quantum control and quantum information theory to define overparameterization in parameterized quantum circuits. These large quantum models have been reported to train faster [86–89] and show noise resilience [90,91] in certain settings. For instance, a "computational phase transition" at the critical point between the underparameterized and overparameterized regimes was shown in Ref. [86], where the task is to learn a Haar random unitary. The faster convergence and changes in loss landscape to global minima are well understood in classical deep learning. We leave the development of a formal connection between the conventional ML definition of overparameterization and that defined using quantum control for follow-up work.

Another important clarification to be made is how the wide quantum neural network (QNN) literature relates to our work. Wide QNNs represent the quantum analog of classical neural networks that are studied in the limit of infinite width (number of hidden layer units). This hypothetical setting makes it easier to study the training dynamics of the network, which can then be used to understand the behavior of large models. Like their classical counterparts, wide QNNs show a "lazy training" regime where for wide enough networks the initial parameters do not change much during training [92,93] and the network displays exponentially fast convergence (training error decreases exponentially with iterations) to a good local minimum [94–96]. However, in our work, we use only non-deep overparameterized models—which are also more practical. Hence, to our knowledge we are reporting not only the first empirical proof of double descent in QML but also the first such observation for a non-deep quantum learning model. This is not different from classical ML, where double descent is ubiquitous in deep learning but has also been observed in many non-deep learning methods such as boosting, trees, and linear regression [27].

APPENDIX B: REMARK ON THE PARAMETER COUNT

In this appendix, we give further details on the exact number of parameters p in quantum kernel methods. When quantum kernels are being embedded via a feature map $\rho(x)$ on n qubits, a key quantity is the number of linearly independent real parameters in the representation. This number is upper-bounded by the dimension of the vector space of Hermitian matrices acting on a 2^n-dimensional Hilbert space.

A general complex matrix of size $d \times d$ has d^2 complex entries, corresponding to $2d^2$ real parameters. However, quantum states are represented by density matrices, which are Hermitian, positive semi-definite, and trace-normalized. Hermiticity alone imposes that the matrix entries satisfy $\rho = \rho^\dagger$, which reduces the number of real parameters from $2d^2$ to d^2.

This is seen by decomposing any Hermitian matrix M as $M = S + iA$, where S is a real symmetric matrix and A is a real antisymmetric matrix with zero diagonal. The number of independent real parameters in this decomposition is $(d(d+1))/2$ from S, and $(d(d-1))/2$ from A. Adding these yields d^2 real degrees of freedom. For n qubits, where $d = 2^n$, the dimension of the Hermitian vector space is therefore $d^2 = 4^n$.

While positivity and trace normalization constrain the valid subset of Hermitian matrices that represent physical density matrices, they do not reduce the dimensionality of the linear space in which these matrices reside. That is, although the set of density matrices is a strict subset of the Hermitian space, it still spans a space of dimension 4^n in terms of linearly independent real parameters. In this linear sense, the dimension remains unaffected by trace normalization, as can be seen in a simple one-qubit example:

$$\rho = \begin{pmatrix} p & a \\ a^* & 1-p \end{pmatrix},$$

where p and $1-p$ are linearly independent [the only solution to $ap + b(1-p) = 0$ that holds for all $p \in [0,1]$ is the trivial one $a = b = 0$].

The density-matrix representation of a pure state can be fully specified by 2^n complex coefficients, instead of 4^n, because pure states correspond to rank-1 matrices. However, we would like to re-emphasize that the quantity of primary relevance to a linear regression problem is the number of *linearly independent* real parameters. This means that when one considers the feature map $\rho(x)$, which is a mapping of input data to quantum states, the number of linearly independent components remains 4^n, provided that the image of the feature map spans the full space of density matrices. That is true whenever the real components of the density matrix are linearly independent, i.e., $\sum_{i,j} a_{ij}\rho_{ij}(x) = 0$ has no solution *independent of* x except for the trivial one $a_{ij} = 0$ for all i and j. Accordingly, the number of linearly independent real parameters in quantum kernel methods scales as 4^n, reflecting the full dimensionality of the Hermitian space on n qubits, which remains the relevant quantity for assessing expressivity in linear regression tasks.

APPENDIX C: DETAILED DERIVATION OF TEST ERROR DECOMPOSITION

In this appendix, we provide a more detailed derivation of the results reported in Sec. III.

Let the training dataset be represented as $\mathcal{D} = \{(\rho_i, y_i)\}_{i=1}^N$, where ρ_i are density matrices. Using this, we

define the data matrix D and the label matrix Y as

$$D = \begin{pmatrix} [\rho_1]^\dagger \\ \vdots \\ [\rho_N]^\dagger \end{pmatrix}, \qquad Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}. \tag{C1}$$

Here, D is a collection of self-adjoint matrices, with each ρ_i satisfying $\rho^\dagger = \rho$. We retain the Hermitian conjugate notation to maintain clarity, distinguishing between inner and outer products in the formalism. Specifically, in analogy to the notation for vectors and dual vectors (co-vectors), given a matrix ρ and its "co-matrix" $[\sigma]^\dagger$, we adopt the following conventions:

(1) The inner product is the application of a co-matrix on a matrix: $[\sigma]^\dagger \rho = \langle \sigma, \rho \rangle_{\mathrm{HS}} = \mathrm{Tr}\{\sigma\rho\}$, where we introduce the Hilbert-Schmidt inner product as the canonical inner product between Hermitian matrices.
(2) The outer product is the application of a matrix on a co-matrix: $\rho[\sigma]^\dagger = \rho \otimes \sigma$, where we introduce the tensor product as the canonical outer product between Hermitian matrices.

This way, the co-matrices are elements of the dual vector space of Hermitian matrices, so they are linear maps from Hermitian matrices to the reals. Together with the canonical basis of the dual space, both the inner product and the outer product are well-defined and follow the standard ones for usual vector spaces.

The data matrix D should therefore be interpreted as a vector of matrices, $D \in (\mathrm{Herm}(2^n))^N$, rather than a standard matrix. This distinction is essential for correctly defining inner and outer products within our framework. Meanwhile, the label vector Y is a real-valued column satisfying $Y \in \mathbb{R}^{N\times 1}$.

Since quantum models are linear in the Hilbert space of Hermitian operators, the goal is to identify optimal functions within the family

$$\mathcal{G} = \{\mathrm{Tr}\{\rho\mathcal{M}\} \,|\, \mathcal{M} \in \mathrm{Herm}(2^n)\}, \tag{C2}$$

where $\mathcal{M}$ is an observable or "parameter matrix" with $p = 4^n$ free parameters (dimension of the corresponding orthonormal Hermitian basis). We examine two regimes based on the relationship between the number of parameters p and the number of training data points N: the *underparameterized* regime, where $N > p$, and the *overparameterized* regime, where $N < p$. In the underparameterized case, the number of data points exceeds the number of parameters, while in the overparameterized case, the number of parameters exceeds the number of data points. In both scenarios, the task involves solving a quadratic optimization problem to determine the linear function that best fits the data.

Let $D\mathcal{M}$ denote the application of the data matrix D to a parameter matrix $\mathcal{M}$, defined analogously to scalar-vector multiplication but extended to matrices:

$$D\mathcal{M} = \begin{pmatrix} [\rho_1]^\dagger \\ \vdots \\ [\rho_N]^\dagger \end{pmatrix} (\mathcal{M}) = \begin{pmatrix} [\rho_1]^\dagger \mathcal{M} \\ \vdots \\ [\rho_N]^\dagger \mathcal{M} \end{pmatrix} = \begin{pmatrix} \mathrm{Tr}\rho_1\mathcal{M} \\ \vdots \\ \mathrm{Tr}\rho_N\mathcal{M} \end{pmatrix}. \tag{C3}$$

In the underparameterized regime, the model parameters are determined with use of the standard least squares optimization problem:

$$\mathcal{M}^u = \arg\min_{\mathcal{M}} \|D\mathcal{M} - Y\|^2. \tag{C4}$$

The solution to this problem is unique and takes the form [48]

$$\mathcal{M}^u = \left(D^\dagger D\right)^{-1} D^\dagger Y, \tag{C5}$$

where $D^\dagger D$ is the $p \times p$ sample covariance matrix (we abuse notation here: to be more precise, $D^\dagger D$ is the Hessian matrix, and $D^\dagger D/N$ is the sample covariance matrix). Expanding this term, we find

$$D^\dagger D = \begin{pmatrix} \rho_1 & \rho_2 & \cdots & \rho_N \end{pmatrix} \begin{pmatrix} [\rho_1]^\dagger \\ [\rho_2]^\dagger \\ \vdots \\ [\rho_N]^\dagger \end{pmatrix} \tag{C6}$$

$$= \rho_1 [\rho_1]^\dagger + \rho_2 [\rho_2]^\dagger + \cdots + \rho_N [\rho_N]^\dagger \tag{C7}$$

$$= \rho_1 \otimes \rho_1 + \rho_2 \otimes \rho_2 + \cdots + \rho_N \otimes \rho_N \tag{C8}$$

$$= \sum_{k=1}^{N} \rho_k \otimes \rho_k. \tag{C9}$$

Let the corresponding optimal predictor function here be represented by $g^u \in \mathcal{G}$, i.e., $g^u(\rho) = \mathrm{Tr}\rho\mathcal{M}^u$. In contrast, for the overparameterized regime, infinitely many solutions satisfy the least squares condition. We therefore choose the standard minimum-norm least squares optimization problem [48]:

$$\mathcal{M}^o = \arg\min_{\mathcal{M}} \|\mathcal{M}\|_2^2 \tag{C10}$$

$$\text{s.t.} \mathrm{Tr}\rho_k\mathcal{M} = y_k, \forall k \in [N]. \tag{C11}$$

The optimal solution for this problem is well-known from optimization theory:

$$\mathcal{M}^o = D^\dagger \left(DD^\dagger\right)^{-1} Y, \tag{C12}$$

where $DD^\dagger$ is the $N \times N$ Gram matrix, explicitly written as

$$DD^\dagger = \begin{pmatrix} [\rho_1]^\dagger \\ [\rho_2]^\dagger \\ \vdots \\ [\rho_N]^\dagger \end{pmatrix} \begin{pmatrix} \rho_1 & \rho_2 & \cdots & \rho_N \end{pmatrix} \tag{C13}$$

$$= \begin{pmatrix} [\rho_1]^\dagger \rho_1 & [\rho_1]^\dagger \rho_2 & \cdots & [\rho_1]^\dagger \rho_N \\ [\rho_2]^\dagger \rho_1 & [\rho_2]^\dagger \rho_2 & & \\ \vdots & & \ddots & \\ [\rho_N]^\dagger \rho_1 & & & [\rho_N]^\dagger \rho_N \end{pmatrix} \tag{C14}$$

$$= \begin{pmatrix} \mathrm{Tr}\{\rho_1\rho_1\} & \mathrm{Tr}\{\rho_1\rho_2\} & \cdots & \mathrm{Tr}\{\rho_1\rho_N\} \\ \mathrm{Tr}\{\rho_2\rho_1\} & \mathrm{Tr}\{\rho_2\rho_2\} & & \\ \vdots & & \ddots & \\ \mathrm{Tr}\{\rho_N\rho_1\} & & & \mathrm{Tr}\{\rho_N\rho_N\} \end{pmatrix} \tag{C15}$$

$$= (\mathrm{Tr}\rho_k\rho_l)_{k,l=1}^N . \tag{C16}$$

Here, we let the corresponding optimal predictor function be represented by $g^o \in \mathcal{G}$, i.e., $g^o(\rho) = \mathrm{Tr}\rho\mathcal{M}^o$. Our having obtained the empirical risk minimizer for both regimes, it is important to note that these solutions are not guaranteed to coincide with the ground truth for all instances. To formalize this, let us introduce an oracle that provides the expected risk minimizer $\mathcal{M}^*$, which represents the optimal linear model parameters for the entire data distribution:

$$\mathcal{M}^* = \arg\min_{\mathcal{M}} \left\{ \mathbb{E}_{(\rho,y)} \left[(\mathrm{Tr}\rho\mathcal{M} - y)^2 \right] \right\}, \tag{C17}$$

where the expectation value is taken over the underlying data distribution. Notably, even with the expected risk minimizer, the ground truth need not be a linear function in the feature space, implying that $\mathcal{M}^*$ may still incur some error for individual points. To explore this further, consider a new test data point (ρ_t, y_t) outside the training set. Note that this setting where the new test point lies outside the training dataset is called a "random design setting"; both training data and test data are separately sampled from the underlying distribution and the learned predictor does not encounter the test data during training. Note that this is an essential component to achieve double descent [82]. It will not occur in a *fixed design* setting, where the same training inputs are used during testing but with resampled labels.

In general, neither the empirical risk minimizer $\mathcal{M}^{u,o}$ nor the expected risk minimizer $\mathcal{M}^*$ will perfectly predict the label y_t:

$$\mathrm{Tr}\rho_t\mathcal{M}^{u,o} - y_t \neq 0, \tag{C18}$$

$$\mathrm{Tr}\rho_t\mathcal{M}^* - y_t =: e_t. \tag{C19}$$

Here, e_t represents the "residual" error made by $\mathcal{M}^*$ on the new input ρ_t due to label noise or mismatch between the underlying data-generating function and the family of functions $\mathcal{G}$. Specifically, for the training set $\mathcal{D}$, we define E as the vector of errors:

$$E := Y - D\mathcal{M}^*, \tag{C20}$$

$$e_k := y_k - \mathrm{Tr}\{\rho_k\mathcal{M}^*\}. \tag{C21}$$

Note that the expected value of this residual error (over the underlying distribution) is usually called the "irreducible error" in the literature [35]. Following Ref. [38], we now investigate the difference in predictive performance between the empirical risk minimizer and the expected risk minimizer on unseen data. For clarity, we introduce the following notation for predictions:

$$y_t^{u,o} := g^{u,o}(\rho_t) = \mathrm{Tr}\rho_t\mathcal{M}^{u,o}, \tag{C22}$$

$$y_t^* := g^*(\rho_t) = \mathrm{Tr}\rho_t\mathcal{M}^*. \tag{C23}$$

Conventionally, the test error in the predictor is defined as the mean squared error in the assigned labels compared with the underlying distribution. Here, we define the generalization error made by the empirical risk minimizer as the mean squared label error compared with the expected risk minimizer [35]:

$$R[g^{u,o}] = \mathbb{E}_{(\rho,y)}\mathbb{E}_{\mathcal{D}} \left[\left(y^{u,o} - y^* \right)^2 \right] \tag{C24}$$

$$= \mathrm{Var}[g^{u,o}] + B^2[g^{u,o}], \tag{C25}$$

where $\mathbb{E}_{\mathcal{D}}$ represents averaging over all possible training datasets. Here, bias captures how well the learned predictor from the class of functions performs compared with the optimal linear model. It is defined as follows:

$$B[g^{u,o}] = \mathbb{E}_{(\rho,y)} \left[\left(\mathbb{E}_{\mathcal{D}} [y^{u,o}] - y^* \right)^2 \right]. \tag{C26}$$

Similarly, variance of the learned predictor function is defined as follows:

$$\mathrm{Var}[g^{u,o}] = \mathbb{E}_{(\rho,y)}\mathbb{E}_{\mathcal{D}} \left[\left(y^{u,o} - \mathbb{E}_{\mathcal{D}} [y^{u,o}] \right)^2 \right], \tag{C27}$$

which captures the fluctuations in prediction across the choice of training dataset. This method of comparing the performance of the empirical risk minimizer with respect to the expected risk minimizer helps to identify the various sources contributing to the test error. For a single test data point (ρ_t, y_t) and a given sampled training dataset $\mathcal{D}$, we can now analyze the difference $y_t^{u,o} - y_t^*$ in the test error by substituting the expressions for $\mathcal{M}^u$ and $\mathcal{M}^o$ given in Eqs. (C5) and (C12). For the underparameterized case,

we have

$$y_t^u - y_t^* = \mathrm{Tr}\rho_t (D^\dagger D)^{-1} D^\dagger Y - \mathrm{Tr}\rho_t \mathcal{M}^* \tag{C28}$$
$$= \mathrm{Tr}\rho_t (D^\dagger D)^{-1} D^\dagger (D\mathcal{M}^* + E) - \mathrm{Tr}\rho_t \mathcal{M}^* \tag{C29}$$
$$= \mathrm{Tr}\rho_t (D^\dagger D)^{-1} D^\dagger D\mathcal{M}^* + \mathrm{Tr}\rho_t (D^\dagger D)^{-1} D^\dagger E - \mathrm{Tr}\rho_t \mathcal{M}^* \tag{C30}$$
$$= \mathrm{Tr}\rho_t \mathcal{M}^* + \mathrm{Tr}\rho_t (D^\dagger D)^{-1} D^\dagger E - \mathrm{Tr}\rho_t \mathcal{M}^* \tag{C31}$$
$$= \mathrm{Tr}\rho_t (D^\dagger D)^{-1} D^\dagger E. \tag{C32}$$

For the overparameterized case, we have

$$y_t^o - y_t^* = \mathrm{Tr}\rho_t D^\dagger (DD^\dagger)^{-1} Y - \mathrm{Tr}\rho_t \mathcal{M}^* \tag{C33}$$
$$= \mathrm{Tr}\rho_t D^\dagger (DD^\dagger)^{-1} (D\mathcal{M}^* + E) - \mathrm{Tr}\rho_t \mathcal{M}^* \tag{C34}$$
$$= \mathrm{Tr}\rho_t D^\dagger (DD^\dagger)^{-1} D\mathcal{M}^* + \mathrm{Tr}\rho_t D^\dagger (DD^\dagger)^{-1} E - \mathrm{Tr}\rho_t \mathcal{M}^* \tag{C35}$$
$$= \mathrm{Tr}\rho_t \left(D^\dagger (DD^\dagger)^{-1} D - \mathbb{I}_{2^n \times 2^n}\right) \mathcal{M}^* + \mathrm{Tr}\rho_t D^\dagger (DD^\dagger)^{-1} E. \tag{C36}$$

Although the derived expressions may initially appear complex, they exhibit a clear and systematic structure that highlights the distinct mathematical roles played by the sample covariance matrix $D^\dagger D$ and the Gram matrix $DD^\dagger$ in the respective regimes. The data matrix $D \in \mathbb{C}^{N \times (2^n \times 2^n)}$ is rectangular, meaning it does not have a direct inverse. To address this, square matrices are constructed by multiplying D by its Hermitian conjugate either from the left, yielding the sample covariance matrix $D^\dagger D \in \mathbb{C}^{(2^n \times 2^n) \times (2^n \times 2^n)}$, or from the right, producing the Gram matrix $DD^\dagger \in \mathbb{R}^{N \times N}$. In the underparameterized regime, where $N > 4^n$, the rank of D is at most 4^n, making the sample covariance matrix $D^\dagger D$ the appropriate choice for constructing the inverse required for solving the least squares problem. This matrix contains the relationships between the parameters of the model and ensures the solution is well-defined. In contrast, in the overparameterized regime, where $N < 4^n$, the data points are fewer than the model parameters, and the Gram matrix $DD^\dagger$ becomes the relevant object. This matrix captures the relationships among the training data points themselves, reflecting the nature of the optimization problem in this regime. This distinction between $D^\dagger D$ and $DD^\dagger$ underscores the fundamental interplay between model complexity and dataset size. Hence, if D^+ represents the pseudoinverse of the data matrix D, then the error expressions for the underparameterized case can be simplified as

$$y_t^u - y_t^* = \mathrm{Tr}\rho_t (D^\dagger D)^{-1} D^\dagger E \tag{C37}$$
$$= \mathrm{Tr}\rho_t D^+ E, \tag{C38}$$

where $D^+ = (D^\dagger D)^{-1} D^\dagger$ is the left inverse of D such that $D^+ D = \mathbb{I}_{2^n \times 2^n}$. Similarly, for the overparameterized case, we get

$$y_t^o - y_t^* = \mathrm{Tr}\rho_t D^\dagger (DD^\dagger)^{-1} E + \mathrm{Tr}\rho_t \left(D^\dagger (DD^\dagger)^{-1} D - \mathbb{I}_{2^n \times 2^n}\right) \mathcal{M}^* \tag{C39}$$
$$= \mathrm{Tr}\rho_t D^+ E + \mathrm{Tr}\rho_t \left(D^+ D - \mathbb{I}_{2^n \times 2^n}\right) \mathcal{M}^*, \tag{C40}$$

where $D^+ = D^\dagger (DD^\dagger)^{-1}$ is the right inverse of D such that $DD^+ = \mathbb{I}_N$. These expressions, while insightful, remain somewhat cumbersome in their current form. To simplify them, we use the SVD of D:

$$D = U\Sigma V^\dagger, \tag{C41}$$

where $U \in \mathbb{U}(N)$ is a unitary matrix of left singular vectors, associated with the "vector index" of D, $\Sigma \in \mathbb{R}^{N \times (2^n \times 2^n)}$ is a diagonal matrix of singular values, with $\Sigma_{ij}^k \propto \delta_{k,(2^N(i-1)+j)}$, and $V \in \mathbb{U}(2^n \times 2^n)$ is a unitary matrix of right singular vectors, associated with the "matrix indices" of D.

Given the SVD of the data matrix D in Eq. (C41), its pseudoinverse D^+ can be defined as

$$D^+ = V\Sigma^+ U^\dagger, \tag{C42}$$

where Σ^+ is the pseudoinverse of Σ containing the reciprocals of the non-zero singular values on its diagonal.

Substituting the simplified forms into Eqs. (C37) and (C39), we obtain

$$y_t^u - y_t^* = \mathrm{Tr}\rho_t V\Sigma^+ U^\dagger E, \tag{C43}$$
$$y_t^o - y_t^* = \mathrm{Tr}\rho_t V\Sigma^+ U^\dagger E + \mathrm{Tr}\rho_t \left(D^+ D - \mathbb{I}_{2^n \times 2^n}\right) \mathcal{M}^*. \tag{C44}$$

These two expressions show the contribution of a variance-like term and a bias-like term to the test error defined in Eq. (C24). The second term in the expression for the prediction error in the overparameterized regime captures the bias of this predictor because the overparameterized system is underdetermined; rank R cannot exceed N and the information in the remaining $p - N$ dimensions cannot be fully captured [38]. Along the same lines, the zero bias in the underparameterized predictor here is valid in this context since we are comparing it with a linear "teacher" model of the expected risk minimizer (the best linear model in the hypothesis class) [36], which is not necessarily a good predictor. Note that this bias term is

sometimes also referred to as "in-class bias," and it should not be confused with the bias commonly discussed in traditional statistical learning theory, which typically decreases with increasing model complexity. The latter, often termed "misspecification bias," arises only when the expected risk minimizer is compared with the ground truth function [35]. Now, to further illustrate why the first term is a variance-like term, we expand it as

$$\mathrm{Tr}\rho_t V\Sigma^+ U^\dagger E = \sum_{r=1}^{R} \frac{1}{\sigma_r} \mathrm{Tr}\rho_r^V \rho_t \langle u_r, E\rangle, \tag{C45}$$

where we recall $R = \min\{N, 2^n \times 2^n\}$, and the right singular vectors ρ_r^V are Hermitian, positive semi-definite, and unit-trace matrices. The sum captures three key factors:

(1) The reciprocals of the singular values $1/\sigma_r$.
(2) The interaction of ρ_t with the basis of right singular vectors $\mathrm{Tr}\rho_r^V \rho_t$.
(3) The projection of E onto the left singular vectors $\langle u_r, E\rangle$.

This decomposition highlights the distinct roles of the left and right singular vectors. The left singular vectors encode linear combinations over the training data indices, while the right singular vectors span the quantum feature space as an orthonormal basis. Intuitively, the principal directions corresponding to small singular values of a given training dataset $\mathcal{D}$ represent "undersampled" directions of the underlying data distribution for which the training dataset does not capture enough information. Consequently, if a test data point has a large projection onto one of these principal components ($[\sigma_r \approx 0] \wedge [\mathrm{Tr}\rho_r^V \rho_t > 0]$), the model is pushed to extrapolate and overfit in this direction, increasing the variance of the predictor [36]. Hence, these three factors together give rise to fluctuations in the prediction and constitute the variance of the learned predictor function (when averaged over all training datasets).

APPENDIX D: PROOF OF THEOREM 1

In this appendix, we provide a formal statement and proof of Theorem 1, which was discussed informally in Sec. IV.

Theorem 1 (Test error peak at interpolation). Consider a Lipschitz-continuous quantum feature map ρ with p linearly independent dimensions, and consider a linear regression problem in the corresponding feature space. Let $\{(\rho(x_i), y_i)\}_{i=1}^N$ be a training set, where $x_i \sim \mathcal{N}(0, \mathbb{I}_d)$ are d-dimensional i.i.d. Gaussian normal random samples. Let $N, p \to \infty$, where $p/N \to c \in (0, \infty)$. Then, the test error of the quantum linear model based on $\rho(x)$ peaks with high probability at $N = p$.

Furthermore, for a quantum feature map on n qubits, the training set size N at which the peak in test error occurs fulfills $N \in \mathcal{O}(\exp(n))$ with high probability.

The following lemma provides the necessary groundwork for the proof of the theorem.

Lemma D1 (Generalized Marčenko-Pastur law [51]). Let ϕ be an L-Lipschitz-continuous map. Let $X = [\phi(x_1), \phi(x_2), \ldots, \phi(x_N)]$ be an $(N \times p)$-dimensional matrix with i.i.d. sampled inputs $x_i \sim \mathcal{N}(0, \mathbb{I}_d)$ for $i = 1, \ldots, N$. For $N, p \to \infty$ and $p/N \to c \in (0, \infty)$, the empirical spectral density $\mu_p = 1/p \sum_i^p \delta_{\lambda_i}$ of the sample covariance matrix $X^\dagger X$ with eigenvalues λ_i converges to μ_c weakly. On $(0, \infty)$, μ_c has continuous density f_c given by

$$f_c = \begin{cases} \dfrac{1}{2\pi x c}\sqrt{(x-\lambda_-)(\lambda_+ - x)} & \text{if } x \in [\lambda_-, \lambda_+], \\ 0 & \text{otherwise,} \end{cases} \tag{D1}$$

where $\lambda_\pm = (1 \pm \sqrt{c})^2$.

Proof of Theorem 1. Let the L-Lipschitz-continuous feature map on n qubits be given as $\rho = \sum_j \nu_j |\Psi_j\rangle\langle\Psi_j| \in \mathbb{C}^{2^n \times 2^n}$, where Ψ_j is some pure state, ν_j is the probability of occupying this state, and we omit the x dependence for simplicity. Recall from Sec. III that the data matrix D is represented as a vector of density matrices. We now introduce an equivalent representation, the vectorized notation of a density matrix, as it aligns more closely with the formulation of the generalized Marčenko-Pastur law in Lemma D1:

$$|\rho\rangle\!\rangle = \sum_j \nu_j \, |\Psi_j\rangle \otimes |\Psi_j^*\rangle \in \mathbb{C}^{4^n}. \tag{D2}$$

The data matrix can then be written as

$$D = \begin{pmatrix} \langle\!\langle \rho_1 | \\ \langle\!\langle \rho_2 | \\ \vdots \\ \langle\!\langle \rho_N | \end{pmatrix} \in \mathbb{C}^{N \times 4^n}. \tag{D3}$$

Assuming that N and p grow at a constant rate, we can apply Lemma D1 to the vectorized data matrix D. Hence, the empirical spectral density of the sample covariance matrix $D^\dagger D$ converges to μ_c weakly. For $p/N \to 1$, the smallest non-zero eigenvalue of the sample covariance matrix approaches zero, i.e., $\lambda_- \to 0$. Note that the eigenvalues of the sample covariance matrix λ_i correspond to the squared singular values σ_i of the data matrix. The error decomposition in Eq. (14) revealed the dependence of the

test error of quantum linear models on the reciprocal singular values of the data matrix D, and hence the test error is maximized with high probability at $N = p$ if $N, p \to \infty$.

Additionally, the number of linearly independent dimensions of any n-qubit quantum feature map cannot exceed the linear dimension of the vectorized data matrix, which is 4^n. ■

APPENDIX E: PROOF OF THEOREM 2

Here, we restate and prove Theorem 2.

Theorem 2 (Lipschitz continuity). Let $\rho(x) = S(x)\rho_0 S^\dagger(x)$, and let $S(x) = \prod_{l=1}^{L}\prod_{k=1}^{d} S_{lk}(x_k)$ (where the product is ordered with increasing k before increasing l). Let further $S_{lk}(\alpha) = \exp\left(-\frac{i}{2}\alpha H_{lk}\right)$. Let $\lambda = \max_{k,l}\|H_{lk}\|_{\mathcal{H}}$. Then, for any $x, x' \in \mathbb{R}^d$, the map $x \mapsto \rho(x)$ is Lipschitz continuous with respect to the operator norm with a Lipschitz constant upper-bounded by $\sqrt{dL}\lambda$:

$$\|\rho(x) - \rho(x')\|_{\mathcal{H}} \leq \sqrt{dL}\lambda\|x - x'\|. \tag{E1}$$

Proof. We start by writing out the difference

$$\rho(x) - \rho(x') = S(x)\rho_0 S^\dagger(x) - S(x')\rho_0 S^\dagger(x') \tag{E2}$$

$$= S(x)\rho_0 S^\dagger(x) - S(x)\rho_0 S^\dagger(x') + S(x)\rho_0 S^\dagger(x') - S(x')\rho_0 S^\dagger(x'), \tag{E3}$$

and then use it explicitly to bound the norm of the difference:

$$\left\|\rho(x) - \rho(x')\right\|_{\mathcal{H}} \leq \left\|S(x)\rho_0 S^\dagger(x) - S(x)\rho_0 S^\dagger(x')\right\|_{\mathcal{H}} + \left\|S(x)\rho_0 S^\dagger(x') - S(x')\rho_0 S^\dagger(x')\right\|_{\mathcal{H}} \tag{E4}$$

$$= \left\|S(x)\rho_0\left(S^\dagger(x) - S^\dagger(x')\right)\right\|_{\mathcal{H}} + \left\|\left(S(x) - S(x')\right)\rho_0 S^\dagger(x')\right\|_{\mathcal{H}} \tag{E5}$$

$$\leq \|S(x)\|_{\mathcal{H}}\,\|\rho_0\|_{\mathcal{H}}\,\left\|S^\dagger(x) - S^\dagger(x')\right\|_{\mathcal{H}} + \left\|S(x) - S(x')\right\|_{\mathcal{H}}\,\|\rho_0\|_{\mathcal{H}}\,\left\|S^\dagger(x')\right\|_{\mathcal{H}} \tag{E6}$$

$$\leq \left\|S^\dagger(x) - S^\dagger(x')\right\|_{\mathcal{H}} + \left\|S(x) - S(x')\right\|_{\mathcal{H}}, \tag{E7}$$

where in the first equation we add zero and then apply the triangle inequality. We further use the sub-multiplicativity of the operator norm, i.e., $\|AB\| \leq \|A\|\,\|B\|$, and the fact that unitary matrices fulfill $\|S\| = 1$ and density matrices fulfill $\|\rho\| \leq 1$. We then note that, because each S_{lk} is generated by a single parameter, it holds that $\partial_\alpha S_{lk}(\alpha) = -(i/2)H_{lk}S_{lk}(\alpha)$. Next, we recall that the total derivative of $S(x)$ is a vector of all partial derivatives

$$\frac{\mathrm{d}S(x)}{\mathrm{d}x} = \left(\partial_k S(x)\right)_{k=1}^{d}. \tag{E8}$$

Indeed, the total derivative takes the form of a 3-tensor, as a vector of matrices. Using the mean value theorem [97]

$$\|S(x_1) - S(x_2)\|_{\mathcal{H}} \leq \left\|\frac{\mathrm{d}S(x)}{\mathrm{d}x}\Big|_{z\in[x_1,x_2]}\right\|_{\mathcal{H}} \|x_1 - x_2\|_2\,, \tag{E9}$$

we need to find an upper bound to the operator norm. We invoke Theorem 3.1 in Ref. [98]:

$$\left\|\frac{\mathrm{d}S(x)}{\mathrm{d}x}\right\|_{\mathcal{H}} \leq \left\|\left(\|\partial_k S(x)\|_{\mathcal{H}}\right)_{k=1}^{d}\right\|_2 \tag{E10}$$

$$= \sqrt{\sum_{k=1}^{d}\|\partial_k S(x)\|_{\mathcal{H}}^2} \tag{E11}$$

$$= \sqrt{\sum_{k=1}^{d}\left\|\partial_k \prod_{l=1}^{L}\prod_{k'=1}^{d} S_{lk'}(x_{k'})\right\|_{\mathcal{H}}^2}. \tag{E12}$$

To keep the notation clear when applying the chain rule to the partial derivative of the product, we proceed by noting

$$S(x) = S_{11}(x_1)\cdots S_{1d}(x_d)S_{21}(x_1)\cdots S_{Ld}(x_d), \tag{E13}$$

$$\partial_k S(x) = S_{11}(x_1)\cdots\partial_k S_{1k}(x_k)\cdots S_{Ld}(x_d) + \cdots + S_{11}(x_1)\cdots\partial_k S_{Lk}(x_k)\cdots S_{Ld}(x_d) \tag{E14}$$

$$= -\frac{i}{2}\left(S_{11}(x_1)\cdots H_{1k}S_{1k}(x_k)\cdots S_{Ld}(x_d) + \cdots + S_{11}(x_1)\cdots H_{Lk}S_{Lk}(x_k)\cdots S_{Ld}(x_d)\right). \tag{E15}$$

Inserting Eq. (E15) into Eq. (E12) and using the fact that matrix norms are invariant under multiplication with unitary matrices, we obtain

$$\left\| \frac{\mathrm{d}S(x)}{\mathrm{d}x} \right\|_{\mathcal{H}} \leq \sqrt{\sum_{k=1}^{d} \left\| \partial_k \prod_{l=1}^{L} \prod_{k'=1}^{d} S_{lk'}(x_{k'}) \right\|_{\mathcal{H}}^2} \tag{E16}$$

$$= \sqrt{\sum_{k=1}^{d} \left\| -\frac{i}{2} \left(S_{11}(x_1) \cdots H_{1k} S_{1k}(x_k) \cdots S_{Ld}(x_d) + \cdots + S_{11}(x_1) \cdots H_{Lk} S_{Lk}(x_k) \cdots S_{Ld}(x_d) \right) \right\|_{\mathcal{H}}^2} \tag{E17}$$

$$\leq \frac{1}{2} \sqrt{\sum_{k=1}^{d} \| S_{11}(x_1) \cdots H_{1k} S_{1k}(x_k) \cdots S_{Ld}(x_d) \|_{\mathcal{H}}^2 + \cdots + \| S_{11}(x_1) \cdots H_{Lk} S_{Lk}(x_k) \cdots S_{Ld}(x_d) \|_{\mathcal{H}}^2} \tag{E18}$$

$$= \frac{1}{2} \sqrt{\sum_{k=1}^{d} \| H_{1k} \|_{\mathcal{H}}^2 + \cdots + \| H_{Lk} \|_{\mathcal{H}}^2} \tag{E19}$$

$$\leq \frac{1}{2} \sqrt{\sum_{k=1}^{d} L\lambda^2} \tag{E20}$$

$$= \frac{\sqrt{dL}\lambda}{2}. \tag{E21}$$

Plugging Eq. (E21) into Eq. (E7), we obtain

$$\| \rho(x) - \rho(x') \|_{\mathcal{H}} \leq 2 \left\| \frac{\mathrm{d}S(x)}{\mathrm{d}x} \right\|_{\mathcal{H}} \| x - x' \| \tag{E22}$$

$$\leq 2 \frac{\sqrt{dL}\lambda}{2} \| x - x' \| \tag{E23}$$

$$= \sqrt{dL}\lambda \| x - x' \|, \tag{E24}$$

which completes the proof. ■

We highlight that a similar result was recently reported in Ref. [99] for the 1-norm.

APPENDIX F: ADDITIONAL EXPERIMENTS AND DETAILS

Here, we provide further implementation details of the numerical experiments, as well as additional numerical results. In all experiments, we consider QML models based on embedding or projected quantum kernels, evaluated on synthetic and real-world datasets. The number of test samples is fixed to 100, while the number of training samples is varied to explore the transition across underparameterized and overparameterized regimes.

The structure of the quantum circuits used to define the feature maps consists of layers of single-qubit rotations, followed by controlled NOT gates. This architecture is repeated to ensure the kernel matrix is full rank. The kernel rank is upper-bounded by $\min(N,p)$, where N is the number of training samples and p is the feature space dimension. In the underparameterized regime ($N > p$), expressivity limits the rank to at most p, while in the overparameterized regime ($N < p$), sufficient expressivity ensures the kernel reaches full rank N. Classical inputs are pre-processed to match the dimensionality constraints of the quantum model. In particular, dimensionality reduction is applied to ensure that inputs can be embedded into circuits with a limited number of qubits. For this purpose, principal component analysis is used to project the data to a space of dimension $2n$, where n is the number of qubits used in the corresponding feature map.

1. Ablation experiments

We extend the ablation experiments described in the main text to real-world datasets; see Fig. 5. These include the Fashion MNIST and California Housing datasets. In one set of experiments, we control the spectral properties of the training data by applying a cutoff to the singular values of the kernel matrix. Removal of low-magnitude components mitigates the peak in test error, confirming that small singular values play a key role in amplifying generalization error near interpolation. In a second set of experiments, we assess the impact of feature alignment between training data and test data by projecting

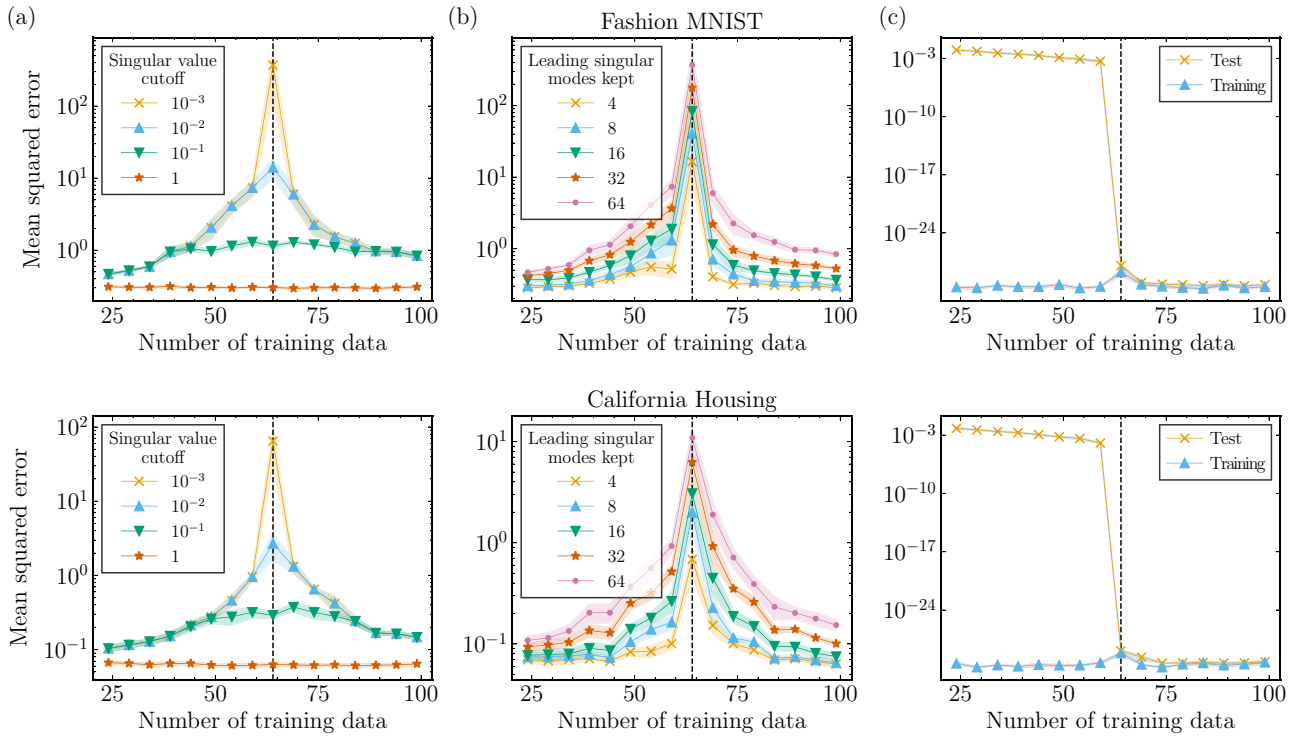


FIG. 5. Ablation experiments for real-world datasets. Mean squared test error as a function of the number of training data points N for (a) application of different cutoffs on the minimum singular value, (b) retention of different numbers of leading singular modes of the input data, and (c) elimination of residual error. All experiments were performed on the Fashion MNIST (top) and California Housing (bottom) datasets with $n = 3$ qubits. The shaded area corresponds to the standard deviation for five independent experiment repetitions, each using independently sampled training data. The dotted black line indicates the interpolation threshold. The region to the left of the threshold corresponds to the overparameterized regime, while the region to the right corresponds to the underparameterized regime. Modification of each factor reduces or eliminates the double descent peak, highlighting their roles in the test error behavior.

test samples onto subsets of the leading singular directions derived from the training set. As the number of retained directions decreases, the test data become less aligned with the training data, leading to a diminished peak. Lastly, we also perform residual error ablation experiments. The results on real-world data qualitatively mirror those from the synthetic case, showing that elimination of the residual error significantly reduces or removes the double descent

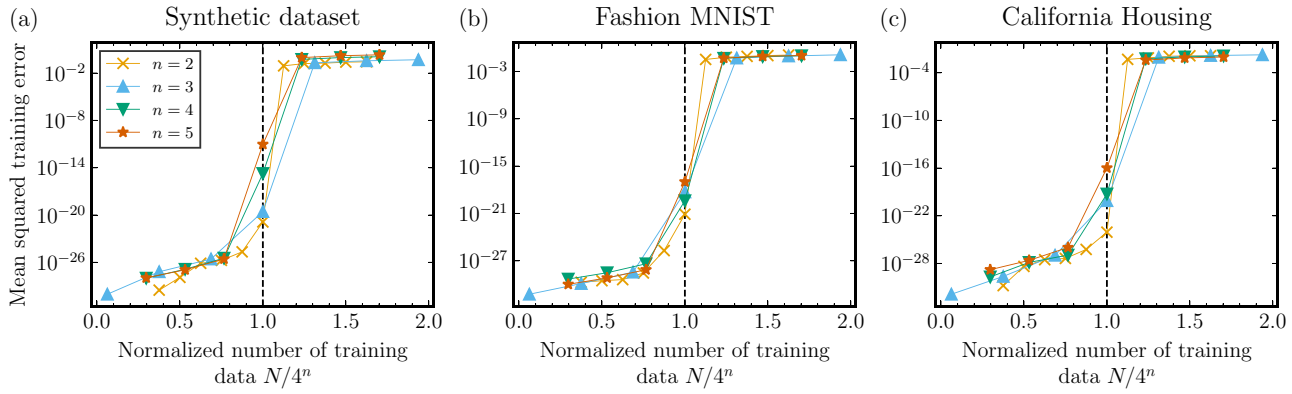


FIG. 6. Mean squared training error as a function of the normalized number of training data points ($N/4^n$) for the (a) synthetic, (b) Fashion MNIST, and (c) California Housing datasets, with use of the EQK in Eq. (3). The dotted black line indicates the interpolation threshold. The region to the left of the threshold corresponds to the overparameterized regime, while the region to the right corresponds to the underparameterized regime.

peak. This further supports the theoretical decomposition and confirms that residual error plays a central role in the emergence of the test error peak.

2. Training error

In addition to the test error curves shown in Fig. 2, we present the corresponding training error curves in Fig. 6. As expected, the training error remains close to zero in the overparameterized regime ($N/4^n < 1$), reflecting the model's capacity to fit the training data almost perfectly. In contrast, in the underparameterized regime ($N/4^n > 1$), the training error is small but non-zero, indicating that the model cannot fully capture the training data due to limited capacity. This behavior aligns well with theoretical expectations and clearly illustrates that the double descent phenomenon is not rooted in the training error behavior.

3. Sampling noise

To analyze how sampling noise affects the double descent curve, we performed a numerical study with different numbers of shots on $n = 2$ qubits and the Fashion MNIST dataset; see Fig. 7. For the simulation of n_{shots} shots we add Gaussian noise with variance $\sigma^2 = K_{ij}(K_{ij} - 1)/n_{\text{shots}}$ to each entry K_{ij} of the Gram matrix as well as on kernel evaluations on test points. The perturbed Gram matrix is then projected back to a positive semi-definite matrix by our setting all negative eigenvalues to zero and reconstructing the matrix from its eigen-decomposition. Effectively, a limited number of shots constrains the precision with which the Gram matrix can be estimated. To ensure consistency, we also limit the numerical precision when computing its spectrum to approximately $\log_{10}(n_{\text{shots}}/2)$ decimals. Since the test error depends on the inverse of the non-zero eigenvalues of the Gram matrix, shot noise acts similarly to a regularization, effectively smoothing the curve and reducing the prominence of the peak.

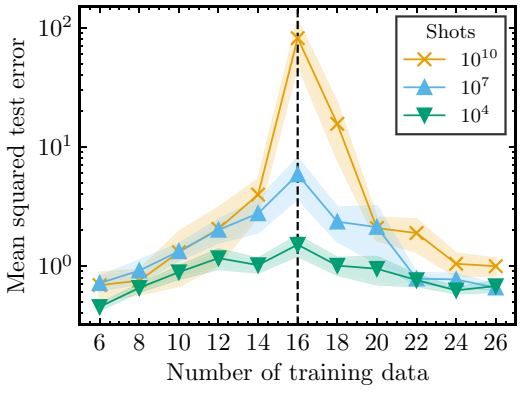


FIG. 7. Sampling noise. Mean squared test error as a function of the number of training data points N for different numbers of shots. The experiments were performed on the Fashion MNIST dataset with $n = 2$ qubits. The dotted black line indicates the interpolation threshold. The region to the left of the threshold corresponds to the overparameterized regime, while the region to the right corresponds to the underparameterized regime.

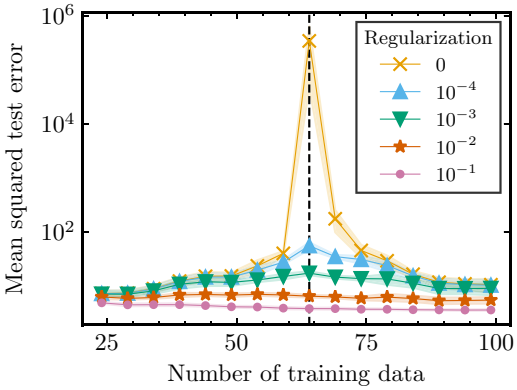


FIG. 8. Regularization. Mean squared test error as a function of the number of training data points N for several values of the regularization parameter λ. The experiments were performed on the synthetic dataset with $n = 3$ qubits. The dotted black line indicates the interpolation threshold. The region to the left of the threshold corresponds to the overparameterized regime, while the region to the right corresponds to the underparameterized regime.

4. Regularization

A common practice in kernel methods is to add a regularization term of the form $\lambda\|\mathcal{M}\|^2$ to the optimization problem, where the regularization parameter $\lambda \geq 0$ controls the trade-off between accurately fitting the training data and promoting a smoother, more stable solution. Figure 8 illustrates how increasing λ affects the double descent behavior in quantum kernel methods for $n = 3$ qubits on the synthetic dataset. As regularization strength grows, the peak in the test error curve becomes less pronounced and eventually disappears.

[1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature **549**, 195 (2017).
[2] V. Dunjko, and H. J. Briegel, Machine learning & artificial intelligence in the quantum domain: A review of recent progress, Rep. Prog. Phys. **81**, 074001 (2018).
[3] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).
[4] M. Schuld, and F. Petruccione, *Machine Learning with Quantum Computers* (Springer, Berlin, 2021), Vol. 676.
[5] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[6] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke *et al.*, Noisy intermediate-scale quantum algorithms, Rev. Mod. Phys. **94**, 015004 (2022).

[7] Y. Liu, S. Arunachalam, and K. Temme, A rigorous and robust quantum speed-up in supervised machine learning, Nat. Phys. **17**, 1013 (2021).

[8] R. Sweke, J.-P. Seifert, D. Hangleiter, and J. Eisert, On the quantum versus classical learnability of discrete distributions, Quantum **5**, 417 (2021).

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

[10] C. Gyurik, and V. Dunjko, Exponential separations between classical and quantum learners, ArXiv:2306.16028.

[11] N. Pirnay, R. Sweke, J. Eisert, and J.-P. Seifert, Superpolynomial quantum-classical separation for density modeling, Phys. Rev. A **107**, 042416 (2023).

[12] R. Molteni, C. Gyurik, and V. Dunjko, Exponential quantum advantages in learning quantum observables from classical data, ArXiv:2405.02027.

[13] E. Gil-Fuster, C. Gyurik, A. Perez-Salinas, and V. Dunjko, in *The Thirteenth International Conference on Learning Representations* (International Conference on Learning Representations (ICLR), 2025).

[14] M. Schuld, Supervised quantum machine learning models are kernel methods, ArXiv:2101.11020.

[15] J. Landman, S. Thabet, C. Dalyac, H. Mhiri, and E. Kashefi, in *The Eleventh International Conference on Learning Representations* (International Conference on Learning Representations (ICLR), 2023).

[16] E. Gil-Fuster, J. Eisert, and C. Bravo-Prieto, Understanding quantum machine learning also requires rethinking generalization, Nat. Commun. **15**, 2277 (2024).

[17] R. Sweke, E. Recio-Armengol, S. Jerbi, E. Gil-Fuster, B. Fuller, J. Eisert, and J. J. Meyer, Potential and limitations of random fourier features for dequantizing quantum machine learning, Quantum **9**, 1640 (2025).

[18] V. N. Vapnik, and A. Y. Chervonenkis, On the uniform convergence of relative frequencies of events to their probabilities, Theory Probab. Appl. **16**, 264 (1971).

[19] L. G. Valiant, A theory of the learnable, Commun. ACM **27**, 1134 (1984).

[20] V. Vapnik, An overview of statistical learning theory, IEEE Trans. Neural Netw. **10**, 988 (1999).

[21] S. Shalev-Shwartz, and S. Ben-David, *Understanding Machine Learning* (Cambridge University Press, Cambridge, England, 2014).

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, in *Advances in Neural Information Processing Systems* (Neural Information Processing Systems Foundation (NeurIPS), 2012), Vol. 25.

[23] B. Neyshabur, R. Tomioka, and N. Srebro, in *3rd International Conference on Learning Representations, ICLR* (International Conference on Learning Representations (ICLR), 2015).

[24] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, in *International Conference on Learning Representations* (International Conference on Learning Representations (ICLR), 2017).

[25] R. Novak, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein, in *International Conference on Learning Representations* (International Conference on Learning Representations (ICLR), 2018).

[26] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, Deep double descent: Where bigger models and more data hurt, J. Stat. Mech.: Theory Exp. **2021**, 124003 (2021).

[27] M. Belkin, D. Hsu, S. Ma, and S. Mandal, Reconciling modern machine-learning practice and the classical bias-variance trade-off, Proc. Natl. Acad. Sci. **116**, 15849 (2019).

[28] Y. Cooper, The loss landscape of overparameterized neural networks, ArXiv:1804.10200.

[29] S. Spigler, M. Geiger, S. d'Ascoli, L. Sagun, G. Biroli, and M. Wyart, A jamming transition from under- to over-parametrization affects generalization in deep learning, J. Phys. A: Math. Theor. **52**, 474001 (2019).

[30] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro, in *International Conference on Learning Representations* (International Conference on Learning Representations (ICLR), 2019).

[31] Z. Allen-Zhu, Y. Li, and Z. Song, in *Proceedings of the 36th International Conference on Machine Learning* (International Conference on Machine Learning (ICML), 2019), Vol. 97.

[32] M. S. Advani, A. M. Saxe, and H. Sompolinsky, High-dimensional dynamics of generalization error in neural networks, Neural Netw. **132**, 428 (2020).

[33] B. Adlam, and J. Pennington, in *Advances in Neural Information Processing Systems* (Neural Information Processing Systems Foundation (NeurIPS), 2020), Vol. 33.

[34] M. Geiger, A. Jacot, S. Spigler, F. Gabriel, L. Sagun, S. d'Ascoli, G. Biroli, C. Hongler, and M. Wyart, Scaling description of generalization with number of parameters in deep learning, J. Stat. Mech.: Theory Exp. **2020**, 023401 (2020).

[35] Y. Dar, V. Muthukumar, and R. G. Baraniuk, A farewell to the bias-variance tradeoff? An overview of the theory of overparameterized machine learning, ArXiv:2109.02355.

[36] J. W. Rocks, and P. Mehta, Memorizing without overfitting: Bias, variance, and interpolation in overparameterized models, Phys. Rev. Res. **4**, 013201 (2022).

[37] A. Curth, A. Jeffares, and M. van der Schaar, in *Thirty-seventh Conference on Neural Information Processing Systems* (Neural Information Processing Systems Foundation (NeurIPS), 2023).

[38] R. Schaeffer, M. Khona, Z. Robertson, A. Boopathy, K. Pistunova, J. W. Rocks, I. R. Fiete, and O. Koyejo, Double descent demystified: Identifying, interpreting & ablating the sources of a deep learning puzzle, ArXiv:2303.14151.

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

[40] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, A review of barren plateaus in variational quantum computing, Nat. Rev. Phys. **7**, 174 (2025).

[41] S. Thanasilp, S. Wang, M. Cerezo, and Z. Holmes, Exponential concentration in quantum kernel methods, Nat. Commun. **15**, 5200 (2024).

[42] M. Larocca, N. Ju, D. García-Martín, P. J. Coles, and M. Cerezo, Theory of overparametrization in quantum neural networks, Nat. Comput. Sci. **3**, 542 (2023).

[43] 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**, 1 (2023).

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

[45] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature **567**, 209 (2019).

[46] T. Hubregtsen, D. Wierichs, E. Gil-Fuster, P.-J. H. S. Derks, P. K. Faehrmann, and J. J. Meyer, Training quantum embedding kernels on near-term quantum computers, Phys. Rev. A **106**, 042431 (2022).

[47] E. Gil-Fuster, J. Eisert, and V. Dunjko, On the expressivity of embedding quantum kernels, Mach. Learn.: Sci. Technol. **5**, 025003 (2024).

[48] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems* (Springer Science & Business Media, 1996), Vol. 375.

[49] P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler, Benign overfitting in linear regression, Proc. Natl. Acad. Sci. **117**, 30063 (2020).

[50] T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani, Surprises in high-dimensional ridgeless least squares interpolation, The Annals of Statistics **50**, 949 (2022).

[51] C. Louart, and R. Couillet, Concentration of measure and large random matrices with an application to sample covariance matrices, ArXiv:1805.08295.

[52] A. Montanari, and P.-M. Nguyen, in *2017 IEEE International Symposium on Information Theory (ISIT)* (Institute of Electrical and Electronics Engineers (IEEE), 2017).

[53] R. Couillet, and Z. Liao, *Random Matrix Methods for Machine Learning* (Cambridge University Press, 2022).

[54] L. Pesce, F. Krzakala, B. Loureiro, and L. Stephan, in *Proceedings of the 40th International Conference on Machine Learning*, ICML'23 (International Conference on Machine Learning (ICML), 2023).

[55] U. Adomaityte, G. Sicuro, and P. Vivo, in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23 (Neural Information Processing Systems Foundation (NeurIPS), 2024).

[56] M. E. A. Seddik, C. Louart, M. Tamaazousti, and R. Couillet, in *International Conference on Machine Learning* (PMLR, 2020).

[57] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial networks, Commun. ACM **63**, 139 (2020).

[58] B. Schölkopf, and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, Adaptive Computation and Machine Learning (MIT Press, 2002).

[59] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. **12**, 2825 (2011).

[60] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi *et al.*, PennyLane: Automatic differentiation of hybrid quantum-classical computations, ArXiv:1811.04968.

[61] Double descent in quantum machine learning, GitHub repository, 2025, https://github.com/bpcarlos/double_descent_quantum_ML

[62] 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).

[63] B. Y. Gan, D. Leykam, and S. Thanasilp, A unified framework for trace-induced quantum kernels, ArXiv:2311.13552.

[64] Y. Du, Y. Yang, D. Tao, and M.-H. Hsieh, Problem-dependent power of quantum neural networks on multiclass classification, Phys. Rev. Lett. **131**, 140601 (2023).

[65] L. Innocenti, S. Lorenzo, I. Palmisano, A. Ferraro, M. Paternostro, and G. M. Palma, Potential and limitations of quantum extreme learning machines, Commun. Phys. **6**, 118 (2023).

[66] W. Xiong, G. Facelli, M. Sahebi, O. Agnel, T. Chotibut, S. Thanasilp, and Z. Holmes, On fundamental aspects of quantum extreme learning machines, Quant. Mach. Intell. **7**, 20 (2025).

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

[68] S. Thabet, L. Monbroussou, E. Z. Mamon, and J. Landman, When quantum and classical models disagree: Learning beyond minimum norm least square, ArXiv:2411.04940.

[69] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, Quantum embeddings for machine learning, ArXiv:2001.03622.

[70] V. Muthukumar, K. Vodrahalli, V. Subramanian, and A. Sahai, Harmless interpolation of noisy data in regression, IEEE J. Sel. Areas Inf. Theory **1**, 67 (2019).

[71] 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).

[72] L. Banchi, J. Pereira, and S. Pirandola, Generalization in quantum machine learning: A quantum information standpoint, PRX Quantum **2**, 040321 (2021).

[73] C.-C. Chen, M. Watabe, K. Shiba, M. Sogabe, K. Sakamoto, and T. Sogabe, On the expressibility and overfitting of quantum circuit learning, ACM Trans. Quantum Comput. **2**, 1 (2021).

[74] C. Gyurik, Dyon van Vreumingen, and V. Dunjko, Structural risk minimization for quantum linear classifiers, Quantum **7**, 893 (2023).

[75] T. Haug, and M. S. Kim, Generalization of quantum machine learning models using quantum fisher information metric, Phys. Rev. Lett. **133**, 050603 (2024).

[76] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, Phys. Rev. A **101**, 032308 (2020).

[77] M. Kobayashi, K. Nakaji, and N. Yamamoto, Overfitting in quantum machine learning and entangling dropout, Quantum Machine Intelligence **4**, 30 (2022).

[78] F. Scala, A. Ceschini, M. Panella, and D. Gerace, A general approach to dropout in quantum neural networks, Adv. Quantum Technol. **6**, 2300220 (2023).

[79] S. Sim, J. Romero, J. F. Gonthier, and A. A. Kunitsa, Adaptive pruning-based optimization of parameterized quantum circuits, Quantum Sci. Technol. **6**, 025019 (2021).
[80] T. Haug, K. Bharti, and M. S. Kim, Capacity and quantum geometry of parametrized quantum circuits, PRX Quantum **2**, 040309 (2021).
[81] H. Ohno, Adaptive pruning algorithm using a quantum fisher information matrix for parameterized quantum circuits, Quantum Mach. Intell. **6**, 77 (2024).
[82] A. Curth, Classical statistical (in-sample) intuitions don't generalize well: A note on bias-variance tradeoffs, overfitting and moving from fixed to random designs, ArXiv:2409.18842.
[83] 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).
[84] V. Papyan, X. Y. Han, and D. L. Donoho, Prevalence of neural collapse during the terminal phase of deep learning training, Proc. Natl. Acad. Sci. **117**, 24652 (2020).
[85] L. Hui, M. Belkin, and P. Nakkiran, Limitations of neural collapse for understanding generalization in deep learning, ArXiv:2202.08384.
[86] B. T. Kiani, S. Lloyd, and R. Maity, Learning unitaries by gradient descent, ArXiv:2001.11897.
[87] R. Wiersema, C. Zhou, Y. de Sereville, J. F. Carrasquilla, Y. B. Kim, and H. Yuen, Exploring entanglement and optimization within the Hamiltonian variational ansatz, PRX Quantum **1**, 020319 (2020).
[88] J. Kim, J. Kim, and D. Rosa, Universal effectiveness of high-depth circuits in variational eigenproblems, Phys. Rev. Res. **3**, 023203 (2021).
[89] H. R. Grimsley, G. S. Barron, E. Barnes, S. E. Economou, and N. J. Mayhall, Adaptive, problem-tailored variational quantum eigensolver mitigates rough parameter landscapes and barren plateaus, Npj Quantum Inf. **9**, 19 (2023).
[90] E. Fontana, N. Fitzpatrick, D. M. n. Ramo, R. Duncan, and I. Rungger, Evaluating the noise resilience of variational quantum algorithms, Phys. Rev. A **104**, 022403 (2021).
[91] M. Duschenes, J. Carrasquilla, and R. Laflamme, Characterization of overparametrization in the simulation of realistic quantum systems, Phys. Rev. A **109**, 062607 (2024).
[92] J. Liu, F. Tacchino, J. R. Glick, L. Jiang, and A. Mezzacapo, Representation learning via quantum neural tangent kernels, PRX Quantum **3**, 030323 (2022).
[93] N. Shirai, K. Kubo, K. Mitarai, and K. Fujii, Quantum tangent kernel, Phys. Rev. Res. **6**, 033179 (2024).
[94] X. You, S. Chakrabarti, and X. Wu, A convergence theory for over-parameterized variational quantum eigensolvers, ArXiv:2205.12481.
[95] J. Liu, K. Najafi, K. Sharma, F. Tacchino, L. Jiang, and A. Mezzacapo, Analytic theory for the dynamics of wide quantum neural networks, Phys. Rev. Lett. **130**, 150601 (2023).
[96] X. You, S. Chakrabarti, B. Chen, and X. Wu, in *Proceedings of the 40th International Conference on Machine Learning* (International Conference on Machine Learning (ICML), 2023).
[97] T. M. Flett, A mean value theorem, The Mathematical Gazette **42**, 38 (1958).
[98] Z. Li, Bounds on the spectral norm and the nuclear norm of a tensor based on tensor partitions, SIAM J. Matrix Anal. Appl. **37**, 1440 (2016).
[99] E. Recio-Armengol, J. Eisert, and J. J. Meyer, Single-shot quantum machine learning, Phys. Rev. A **111**, 042420 (2025).