PAPER • OPEN ACCESS

Fisher information, training and bias in Fourier regression models

MACHINE LEARNING
Science and Technology

PAPER

Fisher information, training and bias in Fourier regression models

Lorenzo Pastori[1,*], **Veronika Eyring**[1,2] and **Mierk Schwabe**[1]

[1] Deutsches Zentrum für Luft- und Raumfahrt (DLR), Institut für Physik der Atmosphäre, Oberpfaffenhofen, Germany
[2] Institute of Environmental Physics (IUP), University of Bremen, Bremen, Germany
[*] Author to whom any correspondence should be addressed.

E-mail: lorenzo.pastori@dlr.de

Keywords: Fisher information, quantum machine learning, quantum neural networks, Fourier models

Supplementary material for this article is available online

OPEN ACCESS

RECEIVED
29 August 2025

REVISED
26 June 2026

ACCEPTED FOR PUBLICATION
23 July 2026

PUBLISHED
12 August 2026

Abstract

Motivated by the growing interest in quantum machine learning, in particular quantum neural networks (QNNs), we study how recently introduced evaluation metrics based on the Fisher information matrix (FIM) are effective for predicting their training and prediction performance. We exploit the equivalence between a broad class of QNNs and Fourier models, and study the interplay between the *effective dimension* (ED) and the *bias* of a model towards a given task, investigating how these affect the model's training and performance. We show that for a model that is completely agnostic, or unbiased, towards the function to be learned, a higher ED likely results in a better trainability and performance. On the other hand, for models that are biased towards the function to be learned a lower ED is likely beneficial during training. To obtain these results, we derive an analytical expression of the FIM for Fourier models and identify the features controlling a model's ED. This allows us to construct models with tunable ED and bias, and to compare their training. We furthermore introduce a tensor network representation of the considered Fourier models, which could be a tool of independent interest for the analysis of QNN models. Overall, these findings provide an explicit example of the interplay between geometrical properties, model-task alignment and training, which are relevant for the broader machine learning community.

1. Introduction

A popular approach for developing quantum machine learning (QML) models for the analysis of classical data is to use parameterized quantum circuits (PQCs) as trainable machine learning models [1–3]. In these models, also known as quantum neural networks (QNNs) [4], the classical input data and the trainable parameters are encoded as angles in the quantum gates of the circuit, and the outputs are extracted as expectation values of some observables at the end of the PQC [1–4]. The variational nature of QNNs, where the parameters are typically trained in a quantum–classical feedback loop [5], makes them viable approaches for near-term quantum devices [6, 7].

In the last years, several works theoretically investigated how QNNs differ from their classical counterparts, with particular focus in understanding their expressivity [8, 9] and their generalization capability [10–13]. While no general consensus exists as to whether these variational QML models can offer rigorous advantages compared to classical approaches on classical 'real-world' datasets, there is a growing body of literature proposing applications of QNNs in several areas of classical data analysis and machine learning [14–24].

In parallel to these investigations, there are also several attempts to develop general evaluation metrics for such architectures. Finding such metrics assessing the quality of a model on a given task, *before* the model has been trained for it, is indeed a key challenge for any machine learning practitioner. While some of the proposed metrics, such as the quantum expressivity [25, 26] or entangling capability [25, 26] are only relevant in the case of variational quantum models, other metrics, such as the effective

dimension (ED) [27] can be calculated for both classical and quantum models. This latter quantity is the main focus of our work.

The ED, calculated from the Fisher information matrix (FIM), has been recently introduced in a seminal work [27] as a measure for the capacity of a model to effectively explore all of its degrees of freedom, i.e. make use of its full parameter space. In [27], the authors not only use the ED for constructing theoretical generalization bounds, but also show numerical examples of QNNs having larger ED than classical networks used for comparison, which they relate to their faster training ability for a chosen learning task. These encouraging results then prompt the question: do models with a high ED *always* have faster training abilities?

In this work, we provide a negative answer to this question. Focusing on models for regression, we show that whether a model with high ED has better training performance compared to one with low ED depends on how *biased* the models are towards the specific regression task. In particular, for models that are completely agnostic, or unbiased, towards the function to be learned (the data-generating function), a high ED likely results in a better trainability. On the other hand, for models that are biased towards the function to be learned a lower ED is beneficial during training. Maybe unsurprisingly, these results quantitatively confirm the intuitive expectation that when the effective space a model can explore is constrained around the task's data-generating function, training the model to a good performance becomes easier. The goal of this work is to explain the basic mechanisms behind this phenomenon, understand them on an analytical level as schematically illustrated in figures 1(a) and (b), and provide numerical results supporting the findings. More general, our work hints towards the difficulty, and perhaps the impossibility, of finding a data- and task-independent evaluation metric that can assess a ML model's performance prior to its training.

1.1. Related literature in quantum and classical ML

The ED has become a standard information-geometric diagnostic for comparing the capacity of QNNs and related parametrized quantum models, following its introduction in [27] as expressivity metric and trainability proxy. It is used as an evaluation metric in several QML studies as a means of comparing different (quantum or classical) models and as guiding principle for model design [28–36]. This literature establishes ED as a useful capacity metric for QNN architecture assessment. At the same time, these studies also leave open how ED should be interpreted when the architecture is not only more or less expressive, but also more or less aligned with the target task.

A related question has been studied in classical ML under the notions of task-model alignment, target-kernel alignment, spectral bias, and symmetry-induced inductive bias. In particular, in [37] a notion of model-task alignment is introduced in the context of kernel regression, as the alignment of the target function with the kernel eigenfunctions. In this context, directly related to comparisons between biased and unbiased model classes are the works of [38, 39], which study truncated kernel ridge regression and show that when the target is sufficiently aligned with the retained truncated kernel, the truncated estimator can achieve better learning than full kernel ridge regression. These results further strengthen the intuition that task-aligned model can outperform a less biased or more expressive model when the target lies in the subspace favored by the architecture.

Further classical works connect alignment to training dynamics. For example. in [40] the authors study neural tangent kernel (NTK) alignment during training and argue that the evolution of the NTK toward the target function provides a mechanism for accelerated learning outside the static-kernel regime. Also, symmetry-constrained models provide an important physics-motivated example of useful task alignment [41].

These works support the broader view that trainability cannot be assessed from model capacity alone, but depend crucially on how the model's bias matches the structure of the target task. Here we investigate this issue in a controlled class of models that maps naturally to QNNs, studying ED and model-task alignment jointly rather than as separate diagnostics. This allows us to give an analytical account of the interplay between ED, task bias, and trainability, and to construct explicit examples showing when architecture design should prioritize task-aligned bias over raw expressivity.

1.2. Summary of results

The findings presented in this paper are obtained by exploiting the equivalence between a broad class of QNNs and Fourier models [42–45], which we extend to include the dependence on the trainable QNN parameters. Based on this equivalence, we derive an analytical expression for the FIM, which allows us to identify the relevant features of a Fourier model that control the FIM spectrum, and thereby the ED.

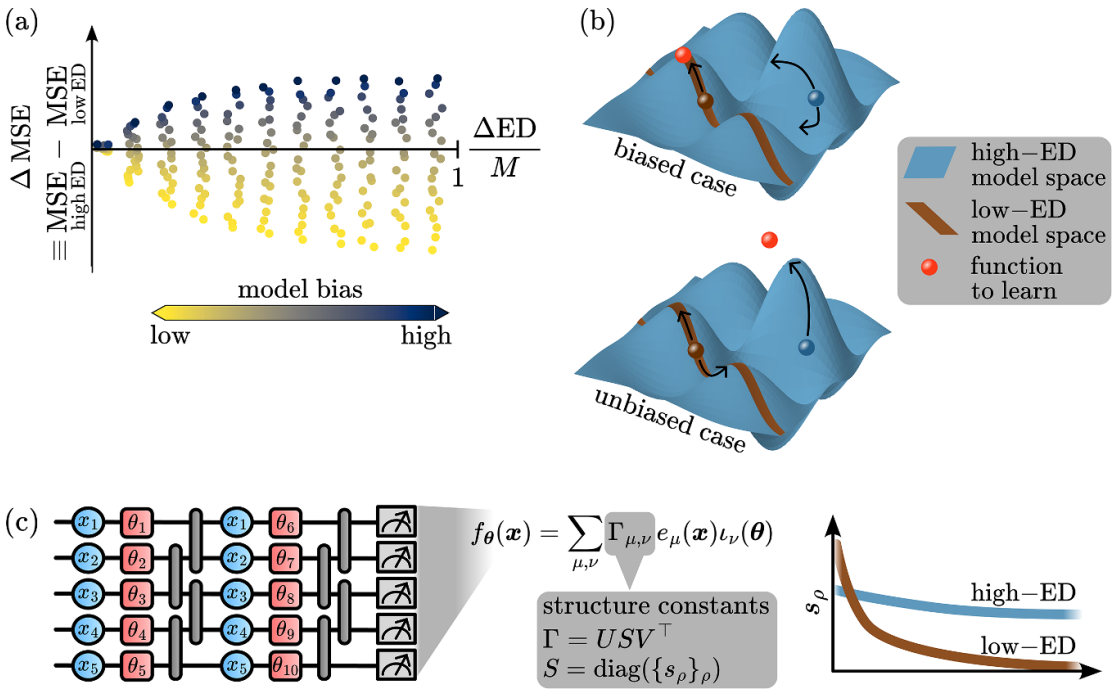


Figure 1. Illustration of main results. (a) Schematic behavior of the difference in the training loss (here the mean squared error—MSE) ΔMSE, between models with high and low effective dimension (ED), vs the corresponding ED difference ΔED (between models with high and low ED—normalized by the number of trainable parameters M), for models with different bias towards the function to be learned (represented by the color scale). Each point represents the average behavior over several model realizations and training experiments. Models with low ED have better training performance than models with high ED (positive ΔMSE) in the biased case. The converse is true (negative ΔMSE) in the unbiased case. (b) Visualization of model spaces in the high- (blue surface) and low-ED (brown line) cases, for biased and unbiased case. Points in these spaces represent functions obtained for specific choices of trainable parameters. In the biased case, the data-generating function (red point) belongs to the model space (to good approximation): a model with low ED (brown point) trained with gradient descent is more likely to converge to the data-generating function since there are effectively less dimensions to explore (only one direction leads to minimizing the loss, as represented by the black arrow). A model with high ED (blue point) is instead more likely to incur in local minima (there are multiple directions, represented by the black arrows, leading to similar loss minimization). In the unbiased case the data-generating function is outside the model space: a model with high ED (blue point) is likely to yield better results, as more directions are available for reaching a better approximation to the data-generating function. (c) Illustration of a QNN and the expansion of its output $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ in the basis functions $e_\mu(\boldsymbol{x})$ and $\iota_\nu(\boldsymbol{\theta})$ (with $\boldsymbol{x}$ denoting the inputs and $\boldsymbol{\theta}$ the trainable parameters). The coefficient matrix Γ (structure constants) can be decomposed in orthogonal matrices U and V and a diagonal matrix S of singular values s_ρ. The ED of the model is controlled by the decay properties of the singular values: a faster decay results in a lower ED.

Specifically, we derive an explicit relationship between the FIM spectrum and ED of a model and the dimension of the space of functions it has access to. This is schematically illustrated in figure 1(c).

These analytical results enable the practical construction of Fourier models with tunable ED, as well as the construction of models that are more or less biased (as we quantify later in this work) towards a given data-generating function for a regression task. This allows us to study the interplay between ED and bias and their effect on the training ability of a model with numerical examples, with the aforementioned main finding: for models that are biased towards the function to be learned, a lower ED is beneficial during training, whereas for unbiased models a high ED is likely to achieve better performance (see figures 1(a) and (b) for a schematic representation).

As a secondary result, in order to numerically investigate larger problem instances (i.e. with larger number of input features and parameters) we introduce a tensor network (TN) representation of the structure of Fourier models, called *tensorized* Fourier models. These could be a tool of independent interest for the analysis of QNN models.

1.3. Organization of the paper

The remainder of the paper is structured as follows. In section 2.1 we define the general structure of the regression models we focus on in this work, making the connection with QNNs explicit. In section 2.2 we recap the notions of FIM and ED and, via analytical calculations and numerical examples, we show how these depend on the models' characteristics. In section 2.3 we give our working definition of model bias, and in section 2.4 we introduce the concept of tensorized models. With these definitions at hand, in section 3 we present our results on the interplay between ED and model bias and their effect on

training regression models via gradient descent. Finally, we conclude and provide an outlook on possible future investigations in section 4.

2. Methods

2.1. Preliminaries: models and structure constants

In this section we define the general structure of the regression models that we focus on throughout the paper. Besides providing their general definition, we discuss their relation with functions parameterized by QNNs, and introduce the concept of their *structure constants*, which is the central object of our subsequent analysis. For clarity, a list of symbols used throughout this work is provided in table 1.

2.1.1. Definition of regression models used

In this work we consider real regression models taking as input a vector $\boldsymbol{x} \in \mathbb{R}^N$, with N denoting the number of input components (features), and parameterized by M trainable (variational) parameters $\boldsymbol{\theta} \in \mathbb{R}^M$. For simplicity in the presentation and derivation of the results, we focus on the case of a single real output, although our results can be easily extended to the case of multi-output models. The general expression of the regression models studied here is

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{\mu=1}^{D} c_\mu(\boldsymbol{\theta})\, e_\mu(\boldsymbol{x}) \ . \tag{1}$$

The functions $e_\mu(\boldsymbol{x}) \in \mathbb{R}$ are taken to form an orthonormal basis in the D-dimensional space of input functions, i.e.

$$\mathbb{E}_{\boldsymbol{x}\sim p}\left[e_\mu(\boldsymbol{x})\, e_{\mu'}(\boldsymbol{x})\right] = \int e_\mu(\boldsymbol{x})\, e_{\mu'}(\boldsymbol{x})\, p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \delta_{\mu,\mu'} \ , \tag{2}$$

with $p(\boldsymbol{x})$ the probability density function for the inputs and $\mathbb{E}_{\boldsymbol{x}}$ the expected value over this input distribution. The coefficients $c_\mu(\boldsymbol{\theta}) \in \mathbb{R}$ encode the dependence on the variational parameters $\boldsymbol{\theta}$. This form of regression model exactly encompasses that of QNNs which, as we discuss later, are known to be Fourier models [42–45].

Going a step further, we may expand the coefficients $c_\mu(\boldsymbol{\theta})$ in a finite orthonormal basis in the space of parameters' functions, under the assumption that the parameter space is effectively bounded, or that the $c_\mu(\boldsymbol{\theta})$ are square-integrable functions. In this case we have

$$c_\mu(\boldsymbol{\theta}) = \sum_{\nu=1}^{K} \Gamma_{\mu,\nu}\, \iota_\nu(\boldsymbol{\theta}) \ , \tag{3}$$

where $\iota_\nu(\boldsymbol{\theta}) \in \mathbb{R}$ are a set of K orthonormal basis functions satisfying

$$\frac{1}{V_\Theta} \int_\Theta \iota_\nu(\boldsymbol{\theta})\, \iota_{\nu'}(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} = \delta_{\nu,\nu'} \ , \tag{4}$$

with Θ denoting the parameter space and V_Θ its volume. As discussed later, this form encompasses the case of QNNs, where both input features and parameters are encoded as rotation angles in quantum gates.

The coefficients $\Gamma_{\mu,\nu} \in \mathbb{R}^{D\times K}$ depend only on the model architecture, and we therefore call them the *structure constants* of the model. The structure constants $\Gamma_{\mu,\nu}$ specify the correlations between the parameter space functions and the input space functions, and are the central object of our analysis throughout this work.

2.1.2. Fourier regression models and relation to QNNs

We now relate the form of the regression models introduced above to that of QNNs, and give an explicit expression of the basis functions $e_\mu(\boldsymbol{x})$ and $\iota_\nu(\boldsymbol{\theta})$ in this specific case. We first briefly recap the concept of a QNN. In the context of regression, a QNN is a function measured from a PQC as

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \langle 0|\, \hat{U}^\dagger_{\boldsymbol{\theta}}(\boldsymbol{x})\, \hat{M}\, \hat{U}_{\boldsymbol{\theta}}(\boldsymbol{x})\, |0\rangle \equiv \langle \psi_{\boldsymbol{\theta}}(\boldsymbol{x})|\, \hat{M}\, |\psi_{\boldsymbol{\theta}}(\boldsymbol{x})\rangle \ . \tag{5}$$

Here, $\hat{M}$ is a given observable, and the output state $|\psi_{\boldsymbol{\theta}}(\boldsymbol{x})\rangle = \hat{U}_{\boldsymbol{\theta}}(\boldsymbol{x})\,|0\rangle$ is obtained from a quantum circuit described by the unitary $\hat{U}_{\boldsymbol{\theta}}(\boldsymbol{x})$ where inputs and parameters are encoded, applied to a reference

Table 1. List of most used symbols in this work with explanation.

Symbol	Used for
N	No. of components of input vector (features)
$\boldsymbol{x} = (x_1, \ldots, x_N)$	Input vector
M	No. of trainable parameters
$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M)$	Vector of trainable parameters
Θ	Parameter space
$f_{\boldsymbol{\theta}}(\boldsymbol{x})$	Regression model (equation (1))
D	No. of input basis functions
$e_\mu(\boldsymbol{x})$	Input basis functions (equation (2))
d	Dim. of space 'local' to each feature ($D = d^N$)
$\mu = (\mu_1, \ldots, \mu_N)$	Index of input basis functions
K	No. of basis functions in param. space
$\iota_\nu(\boldsymbol{\theta})$	Basis functions in param. space (equation (4))
$\tilde{d}$	Dim. of space 'local' to each param. ($K = \tilde{d}^M$)
$\nu = (\nu_1, \ldots, \nu_M)$	Index of param. basis functions
$\iota^{(m)}_{\nu_m}(\theta_m)$	'Local' basis functions for param. θ_m
Γ	Structure constants (equation (3))
U	Left singular vectors of Γ (equation (9))
V	Right singular vectors of Γ (equation (9))
S	Singular values of Γ (correlation spectrum—equation (9))
$F(\boldsymbol{\theta})$	Fisher information matrix (FIM—equation (11))
$\hat{F}(\boldsymbol{\theta})$	Normalized FIM (equation (13))
$\hat{d}_{\text{eff}}$	Normalized effective dimension (equation (12))
$\beta^{(m)}$	Local derivative tensor for param. θ_m (equation (15))

state $|0\rangle$. Typical implementations of QNNs involve encoding the input data $\boldsymbol{x}$ and trainable parameters $\boldsymbol{\theta}$ as angles of rotation gates in the quantum circuit. As we show in appendix A, the QNN output $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ can be written as a Fourier series in both the inputs and parameters as:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{\boldsymbol{\omega}} \sum_{\tilde{\boldsymbol{\omega}}} \tilde{\Gamma}_{\boldsymbol{\omega},\tilde{\boldsymbol{\omega}}}\, \mathrm{e}^{\mathrm{i}\boldsymbol{\omega}\cdot\boldsymbol{x}}\, \mathrm{e}^{\mathrm{i}\tilde{\boldsymbol{\omega}}\cdot\boldsymbol{\theta}}\,, \tag{6}$$

where $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_N)$ with $\omega_n \in \Omega_n$ and Ω_n the set of Fourier frequencies for the nth input component, $\tilde{\boldsymbol{\omega}} = (\tilde{\omega}_1, \ldots, \tilde{\omega}_M)$ with $\tilde{\omega}_m \in \tilde{\Omega}_m$ and $\tilde{\Omega}_m$ the set of Fourier frequencies for the mth parameter, $\tilde{\Gamma}_{\boldsymbol{\omega},\tilde{\boldsymbol{\omega}}}$ complex constants depending only on the quantum gate generators and the measured observable $\hat{M}$, satisfying $\tilde{\Gamma}_{-\boldsymbol{\omega},-\tilde{\boldsymbol{\omega}}} = \tilde{\Gamma}^*_{\boldsymbol{\omega},\tilde{\boldsymbol{\omega}}}$, and $\cdot$ denoting the scalar product of two vectors. The above expression can equivalently be rewritten in the form of equations (1) and (3) with $e_\mu(\boldsymbol{x}) \equiv e_{(\mu_1,\ldots,\mu_N)}(\boldsymbol{x}) = \prod_{n=1}^N e^{(n)}_{\mu_n}(x_n)$ and $\iota_\nu(\boldsymbol{\theta}) \equiv \iota_{(\nu_1,\ldots,\nu_M)}(\boldsymbol{\theta}) = \prod_{m=1}^M \iota^{(m)}_{\nu_m}(\theta_m)$, where

$$e^{(n)}_{\mu_n}(x_n) \in \mathcal{B}_n = \left\{1, \sqrt{2}\cos(\omega_n x_n), \sqrt{2}\sin(\omega_n x_n)\right\}_{\omega_n \in \Omega_n}\,, \tag{7}$$

$$\iota^{(m)}_{\nu_m}(\theta_m) \in \tilde{\mathcal{B}}_m = \left\{1, \sqrt{2}\cos(\tilde{\omega}_m \theta_m), \sqrt{2}\sin(\tilde{\omega}_m \theta_m)\right\}_{\tilde{\omega}_m \in \tilde{\Omega}_m}\,, \tag{8}$$

normalized in the interval $[-\pi, \pi]$, and with the structure constants $\Gamma_{\mu,\nu}$ given by suitable real linear combinations of the complex coefficients $\tilde{\Gamma}_{\boldsymbol{\omega},\tilde{\boldsymbol{\omega}}}$. We remark that this Fourier series representation is not limited to the noiseless QNN setting, but is valid also in presence of quantum hardware noise and gate errors, as discussed in appendix A.

As an explicit example of the sets of frequencies a QNN can have access to, one can consider the common situation where the inputs are encoded multiple times via the re-uploading technique [43, 46], and both input features and parameters are encoded as angles of single qubit rotations of the form $\mathrm{e}^{-\mathrm{i}\frac{\phi}{2}\boldsymbol{n}\cdot\hat{\boldsymbol{\sigma}}}$ (with ϕ being the feature or parameter to be encoded, $\boldsymbol{n}$ an arbitrary rotation axis and $\hat{\boldsymbol{\sigma}}$ the vector of Pauli matrices). In this case, as we show in appendix A, the sets Ω_n and $\tilde{\Omega}_m$ comprise only integer frequencies and read as $\Omega_n = \{1, \ldots, L\}$ and $\tilde{\Omega}_m = \{1\}$, with L being the number of times the input features x_n are uploaded as gate angles in $\hat{U}_{\boldsymbol{\theta}}(\boldsymbol{x})$, and assuming each parameter θ_m is encoded only once. We refer the reader to the Supplementary Material for a detailed discussion on how the structure of the entangling operations in a QNN influences the basis functions the model has access to.

This latter example exemplifies the not uncommon situation where the number of Fourier modes for every input feature (and every parameter) is the same. For simplicity, we restrict ourselves to this case for the analysis presented in this work. This results in no loss of generality, since our analytical and

numerical results can be easily generalized beyond this case. Throughout the rest of this work, we denote with $d \equiv |\mathcal{B}_n|$ the number of 'local' basis functions for the input feature space, and with $\tilde{d} \equiv |\tilde{\mathcal{B}}_m|$ the number of 'local' basis functions for the parameter space, which results in $D = d^N$ and $K = \tilde{d}^M$.

2.1.3. Correlations in structure constants

We now analyze the information that is contained in the structure constants of the model $\Gamma_{\mu,\nu}$. To do so, we view $\Gamma_{\mu,\nu}$ as elements of a $D \times K$ real matrix Γ (we assume $K > D$, since typically we consider models with a number of parameters larger than the number of input features), and consider its singular value decomposition (SVD)

$$\Gamma = USV^\top \,, \tag{9}$$

where U is a $D \times D$ real orthogonal matrix satisfying with $U^\top U = UU^\top = I_D$, $S = \mathrm{diag}(s_1, \ldots, s_D)$ is a $D \times D$ diagonal positive semi-definite matrix (with diagonal ordered as $s_1 \geqslant s_2 \geqslant \ldots \geqslant s_D$), and V is a $K \times D$ real matrix with orthonormal columns, i.e. $V^\top V = I_D$. After the SVD we can rewrite the model output as

$$\begin{aligned} f_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \sum_{\mu=1}^{D}\sum_{\nu=1}^{K} \Gamma_{\mu,\nu}\, e_\mu(\boldsymbol{x})\, \iota_\nu(\boldsymbol{\theta}) \\ &= \sum_{\rho=1}^{D}\sum_{\mu=1}^{D}\sum_{\nu=1}^{K} U_{\mu,\rho}\, s_\rho \left[V^\top\right]_{\rho,\nu} e_\mu(\boldsymbol{x})\, \iota_\nu(\boldsymbol{\theta}) \\ &\equiv \sum_{\rho=1}^{D} s_\rho\, e^U_\rho(\boldsymbol{x})\, \iota^V_\rho(\boldsymbol{\theta}) \,, \end{aligned} \tag{10}$$

where $e^U_\rho(\boldsymbol{x}) \equiv \sum_{\mu=1}^{D} U_{\mu,\rho}\, e_\mu(\boldsymbol{x})$ and $\iota^V_\rho(\boldsymbol{\theta}) \equiv \sum_{\nu=1}^{K} [V^\top]_{\rho,\nu}\, \iota_\nu(\boldsymbol{\theta})$ constitute new sets of orthonormal basis functions in the input and parameters' space. From the above expression one can easily understand how the singular values s_ρ control the correlations between the parameter space and the functions in the input space. In the limiting case of $s_1 > 0$ and $s_{\rho>1} = 0$, any change in the parameters $\boldsymbol{\theta}$ induces a change in $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ along only one component, $e^U_1(\boldsymbol{x})$: in this case, the model's space is effectively one-dimensional. Conversely, if all singular values are equal, the model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ is effectively able to 'explore' all the D orthogonal directions $e^U_\rho(\boldsymbol{x})$ in the input space. Thus, the distribution of the s_ρ, in particular how they decay (i.e. their mutual ratios) indicate how changes in the parameters correlate with the independent directions the model can explore. We therefore call the singular values s_ρ the *correlation spectrum* of the model. In the next section we discuss how, perhaps unsurprisingly, the correlation spectrum is related to the notions of FIM and ED.

2.2. FIM and ED

In this section we discuss the notion of FIM and ED, and relate them to the structure constants of the models introduced in the previous section. We derive an analytic expression of the FIM that makes the relation with the correlation spectrum explicit, and we discuss which features of the correlation spectrum affect the ED.

2.2.1. Definition of FIM and ED

The FIM is a tool of central importance in the field of information geometry, since it provides a local description of how a parameterized model changes when the parameters it depends on are varied. More specifically, the FIM defines a metric in the space (or manifold) of functions a parameterized model can represent [47–49]. For the regression models studied in this work, the FIM F is a $M \times M$ positive semi-definite matrix with elements (see [50–54] and the Supplementary Material)

$$F_{j,k}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}}\left[\frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \theta_j}\frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \theta_k}\right] . \tag{11}$$

At a given point $\boldsymbol{\theta}$ in the parameter space Θ, the FIM describes which directions in Θ the model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ is more (or less) sensitive to. The eigenvalues of the FIM are indeed a measure of the model sensitivity to parameters changes along the direction defined by the corresponding eigenvectors. If all FIM eigenvalues are approximately equal and larger than zero, all parameters are equally contributing to independent model changes. If instead the FIM spectrum contains several eigenvalues close to zero, the corresponding parameter directions are redundant. Thus, the FIM encodes information about how a model is effectively

able to explore its parameter space. A measure of this ability, i.e. the 'size' of the region in model space that a model can effectively explore with its parameters, is given by the ED introduced in [27]. The ED, normalized by the number of parameters M, is defined as

$$\hat{d}_{\text{eff}} = \frac{2\log\left(\frac{1}{V_\Theta}\int_\Theta \sqrt{\det\left(I_M + c_{\mathfrak{n}}\hat{F}(\boldsymbol{\theta})\right)}\,\mathrm{d}\boldsymbol{\theta}\right)}{M\log c_{\mathfrak{n}}}\,, \tag{12}$$

where $c_{\mathfrak{n}} = \frac{\mathfrak{n}}{2\pi\log\mathfrak{n}}$ with $\mathfrak{n}$ being the number of input data samples, I_M the M-dimensional identity matrix, and $\hat{F}(\boldsymbol{\theta})$ being the normalized FIM defined as

$$\hat{F}(\boldsymbol{\theta}) = \frac{M}{\frac{1}{V_\Theta}\int_\Theta \mathrm{tr}\left(F(\boldsymbol{\theta})\right)\mathrm{d}\boldsymbol{\theta}}F(\boldsymbol{\theta})\,. \tag{13}$$

The normalized ED $\hat{d}_{\text{eff}}$ is bounded in $[0,1]$, and is computed from averages over the parameter space, hence depending solely on the architecture choices and the input distribution. Note that this definition of the ED as global average gives a global view of the ability of a model to explore its parameter space, which is especially relevant in the early stages of training when the parameters are typically sampled randomly. We note in passing that also a local definition of the ED exists [55], which probes the vicinity of a point and can be therefore monitored during the training to infer the local geometrical properties during optimization. In the next sections, we uncover the relation between the correlation spectrum introduced before and the FIM, which directly affects the ED defined above.

2.2.2. FIM and correlation spectrum

In section 2.1.3, we discussed how the distribution of the correlation spectrum $\{s_\rho\}_\rho$ controls how many independent directions (in the space of input functions) the model can explore when the parameters are changed. Since the FIM and the ED also capture a notion of effective degrees of freedom of the model, it is natural to expect a strong relation between these and the properties of the correlation spectrum. In this section, we explicitly uncover this relation and show that the FIM spectral properties are indeed mostly controlled by the correlation spectrum.

To this end, we start by expressing the FIM elements in terms of the structure constants Γ and the related correlation spectrum and singular vectors. It is easy to show that the FIM elements can be expressed as

$$F_{j,k}(\boldsymbol{\theta}) = \sum_\rho s_\rho^2 \sum_{\nu,\nu'} \iota_\nu(\boldsymbol{\theta})\,\iota_{\nu'}(\boldsymbol{\theta}) \sum_{\kappa_j,\kappa_k'} \beta^{(j)}_{\kappa_j,\nu_j}\left[V^\top\right]_{\rho,\left(\nu_1\ldots\kappa_j\ldots\nu_M\right)} \beta^{(k)}_{\kappa_k',\nu_k'}\left[V^\top\right]_{\rho,\left(\nu_1'\ldots\kappa_k'\ldots\nu_M'\right)}\,, \tag{14}$$

where we introduce the (local) derivative tensor $\beta^{(j)}$ for expressing the derivatives of the basis functions $\iota_\nu(\boldsymbol{\theta})$ as linear combinations of basis functions, i.e.

$$\frac{\partial\iota_\nu(\boldsymbol{\theta})}{\partial\theta_j} = \frac{\partial\iota^{(j)}_{\nu_j}(\theta_j)}{\partial\theta_j}\prod_{m\neq j}\iota^{(m)}_{\nu_m}(\theta_m) = \left(\sum_{\kappa_j}\beta^{(j)}_{\nu_j,\kappa_j}\iota^{(j)}_{\kappa_j}(\theta_j)\right)\prod_{m\neq j}\iota^{(m)}_{\nu_m}(\theta_m)\,. \tag{15}$$

We refer the reader to the Supplementary Material for a derivation. From equation (14) we can already recognize the dependence on the squared singular values s_ρ^2. These, in combination with equation (13), tell us that the normalized FIM $\hat{F}$, hence the ED, is independent of the value of $\mathrm{tr}(S^2)$. Therefore, the ED is sensitive only to the mutual relationship of the values of s_ρ^2, i.e. the decay properties of S^2, and not to the overall magnitude $\mathrm{tr}(S^2)$. In other words, the FIM and ED are a measure of the dimensionality of the space of functions that the model has access to, captured by how many values s_ρ^2 are effectively different from zero. These observations give us a practical way of designing models with tunable ED, which we will use in our numerical analysis of the training dynamics presented in section 3. We now substantiate these statements by stating the following properties of the FIM.

Property 1. In the regime $M > D$ (which can be interpreted as an overparameterized regime as described in [56]), the rank of the FIM is upper-bounded as follows:

$$\mathrm{rank}\left(F(\boldsymbol{\theta})\right) \leqslant D\,. \tag{16}$$

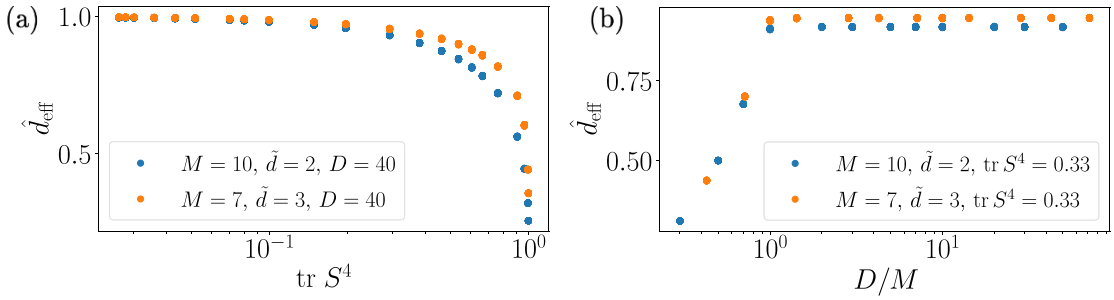


Figure 2. (a) Scaling of normalized ED with the purity $\text{tr}(S^4)$ of the correlation spectrum. (b) Scaling of normalized ED with the ratio D/M. Each point corresponds to a random model realization, i.e. a random Γ uniformly drawn from $[-1,+1]^{D\times K}$. For every value of $\text{tr}(S^4)$ and D/M, 50 model realizations are drawn (the points are on top of each others). The normalized ED is computed using equation (12), with 150 parameters samples for estimating the normalized FIM. Here, $\tilde{d} = 2$ refers to $\tilde{\mathcal{B}}_m = \{\sqrt{2}\cos\theta_m, \sqrt{2}\sin\theta_m\}$, while $\tilde{d} = 3$ refers to $\tilde{\mathcal{B}}_m = \{1, \sqrt{2}\cos\theta_m, \sqrt{2}\sin\theta_m\}$ in equation (8), for all $m = 1,\ldots,M$.

Property 2. In expectation over random realizations of $V \in \text{O}(K)$ (with $\text{O}(K)$ the group of $K \times K$ orthogonal matrices) and $\boldsymbol{\theta} \in \Theta$, one has:

$$\begin{aligned} \mathbb{E}\left[F_{j,k}\right] &\in \begin{cases} \mathcal{O}\left(1\right) \text{tr}\left(S^2\right) , & \text{for } j = k \\ \mathcal{O}\left(K^{-1}\right) \text{tr}\left(S^2\right) , & \text{for } j \neq k \end{cases} \\ \text{Var}\left[F_{j,k}\right] &\in \mathcal{O}\left(1\right) \text{tr}\left(S^4\right) . \end{aligned} \tag{17}$$

The derivation of these properties is sketched in appendix B. From Property 1, together with the observation that the ED is bounded by the maximal rank of the FIM [27], we can conclude that in the regime $M > D$ the ED is upper-bounded by D. This establishes a direct connection between the ED and the dimension D of the space of input functions available to the model. Property 2 further strengthens this connection by showing that the decay properties of S^4, encoded in the value of $\text{tr}(S^4)$, indeed control the FIM spectrum. We now explain this more in detail.

As observed previously, $\text{tr}(S^2)$ is a normalization factor that drops in the calculation of the ED. The term that has non-trivial effects on the FIM and the ED is $\text{tr}(S^4)$ controlling the variance, which encodes information on the decay properties of the correlation spectrum. Assuming (without loss of generality) a correlation spectrum normalized such that $\text{tr}(S^2) = 1$, $\text{tr}(S^4)$ can be interpreted as a *purity*: a completely flat correlation spectrum corresponds to the minimum value of the purity (i.e. $1/D$), whereas $\text{tr}(S^4) = 1$ if $s_1 = 1$ and $s_{\rho>1} = 0$. To understand how the value of $\text{tr}(S^4)$ influences the spectral properties of the FIM, we first observe that the expected value of the FIM is approximately diagonal with diagonal elements having the same values and off-diagonal elements suppressed as $\mathcal{O}(K^{-1})$. Thus, in absence of statistical fluctuations, the spectrum of the FIM would be flat, which would correspond to a high ED. This is approximately the case when the variance is small, i.e. when the correlation spectrum S is flat and $\text{tr}(S^4)$ is small. Conversely, when the correlation spectrum S is not flat and $\text{tr}(S^4)$ is large (i.e. approaching one), $\text{Var}\left[F_{j,k}\right]$ introduces non-negligible statistical fluctuations that make the FIM spectrum deviate from the flat case, which in turn decreases the ED. Thus, with this analysis we identify in the correlation spectrum, and in particular in its decay properties partly captured by $\text{tr}(S^4)$, the key factor controlling the FIM and the ED in regression models. This insight is of central importance in our numerical analysis presented in the next sections.

To corroborate our analytical findings, we show numerical results on the dependence of the ED on different model characteristics in figure 2. Specifically, we calculate the normalized ED $\hat{d}_{\text{eff}}$ for several random model realizations, i.e. random realizations of the structure constants Γ (uniformly drawn from $[-1,+1]^{D\times K}$), and investigate how $\hat{d}_{\text{eff}}$ changes with $\text{tr}(S^4)$ and the input functions' space dimension D. In panel (a) we show the dependence of $\hat{d}_{\text{eff}}$ on the purity of the correlation spectrum $\text{tr}(S^4)$. As expected, $\hat{d}_{\text{eff}}$ decreases with $\text{tr}(S^4)$ increasing towards its maximum value 1. In order to change $\text{tr}(S^4)$, an exponential decay with variable decay rate is imposed to the singular values, keeping them normalized to $\text{tr}(S^2) = 1$ (which has no effect on the ED). In panel (b) we show the dependence of $\hat{d}_{\text{eff}}$ on the ratio D/M. For $D < M$, $\hat{d}_{\text{eff}}$ increases with increasing D until it saturates to its maximal value for $D \geqslant M$. For $D \geqslant M$, $\hat{d}_{\text{eff}}$ is independent of D and largely controlled by $\text{tr}(S^4)$. The increase of $\hat{d}_{\text{eff}}$ for $D < M$ is due to the fact in this regime, the maximal ED of the model is upper-bounded by D, hence the model has more parameters than the input basis functions (i.e. independent directions in model space) it has

access to. Further numerical results are presented in the Supplementary Material and confirm indeed that $\mathrm{tr}(S^4)$ is the key factor controlling the ED.

2.2.3. FIM and NTK

To conclude our analysis of the FIM and establish a connection to the training behavior, we note that there is a strong relation between the FIM and the NTK, which is defined as [57] $K_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{x}') = \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x})^{\top} \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}')$, where $\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x})$ is the M-dimensional gradient of the model. The NTK can be used to analyze how gradient-based training changes function values: it provides a local description of which modes (i.e. directions in the input function space) are learned faster or slower under gradient flow [38, 57–59]. In particular, the eigenvalues of the NTK correspond to the rates at which the corresponding modes are learned [57]. As shown in the Supplementary Material, the non-zero spectrum of the NTK coincides with that of the FIM $F(\boldsymbol{\theta})$. Furthermore, on average over the parameter space and random realizations of $V \in \mathrm{O}(K)$, eigenvectors and eigenvalues of the NTK coincide with the left singular vectors U and singular values S^2 of the model's structure constants. These observations establish the relationship between the NTK eigen-decomposition, the ED (controlled by the FIM), and the bias (defined in the next section, controlled by the model's structure constants). The NTK gives us a way of analytically understanding how different ED and model bias influence the training dynamics, which we elaborate on in the Supplementary Material.

2.3. Biased and unbiased regression models

In this section we introduce the concept of biased and unbiased regression models used in this work, and we provide a simple recipe for constructing models where the bias and the ED can be tuned at will. The main idea behind our definition of biased and unbiased model is the following:

- A model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ is *biased* towards the data-generating function $y(\boldsymbol{x})$ if there exists a parameter configuration $\boldsymbol{\theta}^*$ for which $f_{\boldsymbol{\theta}^*}(\boldsymbol{x}) = y(\boldsymbol{x})$.
- If there is no configuration $\boldsymbol{\theta}^*$ for which $f_{\boldsymbol{\theta}^*}(\boldsymbol{x}) = y(\boldsymbol{x})$ then the model is *unbiased*.

The goal behind the definition of 'model bias' used in this work is to have a controllable way of mimicking the situation where some features of the data-generating function are at least approximately known, and the model is designed such that it can incorporate, or is constrained to represent, these known features. For example, if the task involves predicting some signal coming from a physical process for which the frequency range is known from physical principles or estimations, one could design a model constrained to output signals within the given frequency range: as we remark in the end of this section (and explicitly showcase in the Supplementary Material), this is indeed something that can be achieved in QNNs, via data re-uploading and by tuning their entanglement structure.

For an explicit construction of biased and unbiased models to be used in our numerical experiments, we consider a data-generating function $y(\boldsymbol{x})$ of the form

$$y(\boldsymbol{x}) = \sum_{\mu=1}^{D}\sum_{\nu=1}^{K} e_{\mu}(\boldsymbol{x})\, \iota_{\nu}(\boldsymbol{\theta}^*) \sum_{\rho=1}^{R} s_{\rho}\, U^{(\mathrm{d})}_{\mu,\rho} \left[V^{(\mathrm{d})\top}\right]_{\rho,\nu}, \tag{18}$$

with $\boldsymbol{\theta}^*$ a given parameter configuration, $R < D$ and with $V^{(\mathrm{d})}$ constructed in order to satisfy the property $\sum_{\nu} V^{(\mathrm{d})}_{\nu,\rho}\, \iota_{\nu}(\boldsymbol{\theta}^*) = 0$ for $\rho = R+1,\ldots,K$.

Any model of the form of equation (10) with structure constants chosen independently of $U^{(\mathrm{d})}$ and $V^{(\mathrm{d})}$, with high probability does not exactly encompass $y(\boldsymbol{x})$ for any choice of the parameters: any such model is agnostic to the form of $y(\boldsymbol{x})$, and is referred to as *unbiased*. Instead, a model $f^{(\mathrm{f})}_{\boldsymbol{\theta}}(\boldsymbol{x})$ specified by the structure constants

$$\Gamma^{(\mathrm{f})}_{\mu,\nu} = \sum_{\rho=1}^{D} U^{(\mathrm{d})}_{\mu,\rho}\, s_{\rho} \left[V^{(\mathrm{d})\top}\right]_{\rho,\nu} \tag{19}$$

satisfies $f^{(\mathrm{f})}_{\boldsymbol{\theta}^*}(\boldsymbol{x}) = y(\boldsymbol{x})$ by construction, and is therefore called *biased*. This is schematically illustrated in figure 3(a). Importantly, one can define several such biased models with different properties of s_{ρ}, hence different EDs, by choosing different values for $s_{\rho>R}$, which have no influence on the model prediction for $\boldsymbol{\theta} = \boldsymbol{\theta}^*$. Consider for example the following structure constants

$$\Gamma^{(\mathrm{c})}_{\mu,\nu} = \sum_{\rho=1}^{R} U^{(\mathrm{d})}_{\mu,\rho}\, s_{\rho} \left[V^{(\mathrm{d})\top}\right]_{\rho,\nu} + \sum_{\rho=R+1}^{D} U^{(\mathrm{d})}_{\mu,\rho}\, \mathrm{e}^{-\frac{\rho-R}{\xi}} s_{\rho} \left[V^{(\mathrm{d})\top}\right]_{\rho,\nu}, \tag{20}$$

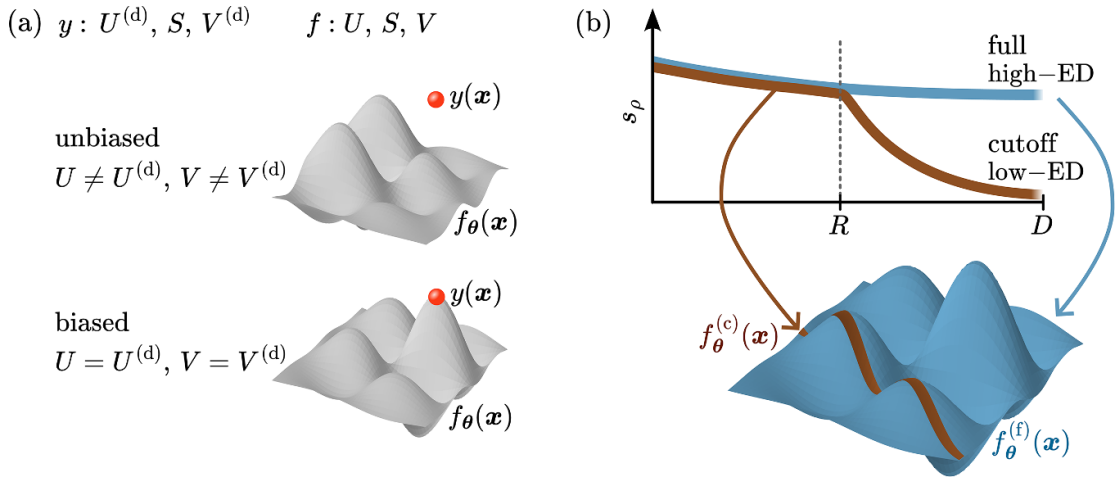


Figure 3. (a) Schematic illustration of the construction of biased and unbiased models. The data-generating function y is specified by matrices $U^{(d)}$ and $V^{(d)}$, and the model f by U and V, with both y and f having the same correlation spectrum S. In the unbiased case, y (represented by the red dot) lies outside the space of functions accessible to $f_{\boldsymbol{\theta}}$ (represented by the gray surface), whereas in the biased case y belongs to that space. (b) Construction of models with tunable ED. Full models $f_{\boldsymbol{\theta}}^{(f)}(\boldsymbol{x})$ with no imposed decay in the correlation spectrum s_ρ, as illustrated by the blue line, have high ED and therefore can access a larger functions' space, represented by the blue surface. Cutoff models $f_{\boldsymbol{\theta}}^{(c)}(\boldsymbol{x})$ with decaying correlation spectrum s_ρ, as illustrated by the brown line, have low ED and have access to more restricted functions' space, represented by the brown surface.

with ξ a positive decay rate. Since ξ induces a decay in the correlation spectrum, and hence a higher spectral purity $\mathrm{tr}(S^4)$, a model specified by $\Gamma^{(c)}$ will have on average a lower ED than a model specified by $\Gamma^{(f)}$. We refer to models constructed as $\Gamma^{(f)}$ as *full* models, whereas models constructed as $\Gamma^{(c)}$, with an imposed decay $\xi > 0$ and a lower ED, are referred to as *cutoff* models. This is schematically illustrated in figure 3(b). Tuning ξ gives us a way of tuning the ED, and hence the difference $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$, which we use in the numerical experiments presented in the next section. Loosely speaking, one can understand a biased model with high ED as a model incorporating knowledge of the target function, while a biased model with low ED is *constrained* to that knowledge without deviating from it (in a functional sense).

Partially biased models, which only approximately encompass $y(\boldsymbol{x})$, can be constructed adding a small perturbation to $V^{(d)}$ in the data-generating function, i.e. replacing $V^{(d)}$ in equation (18) with $V_\epsilon^{(d)}$ obtained by adding a suitably chosen random perturbation of strength ϵ to its elements, as we discuss in appendix C. In this way, $\Gamma^{(f)}$ and $\Gamma^{(c)}$ only approximately encompass the newly constructed data-generating function. We quantify the deviation from $y(\boldsymbol{x})$ using the following parameter

$$\delta_{\mathrm{data}} = \left\| S\left(V^{(d)\top} - V_\epsilon^{(d)\top}\right) \left|\boldsymbol{\iota}(\boldsymbol{\theta}^*)\right) \right\|_1 , \tag{21}$$

with $\left|\boldsymbol{\iota}(\boldsymbol{\theta}^*)\right)$ being the K-dimensional vector with components $\iota_\nu(\boldsymbol{\theta}^*)$.

While this definition of bias is based on the structure constants, we remark that in the context of QNNs these depend on the model architecture (i.e. the form of the unitary $\hat{U}_{\boldsymbol{\theta}}(\boldsymbol{x})$), which can be tuned via the structure of the PQC, ultimately providing a way of biasing the model towards certain types of functions. For example, simply adding new rotation gates re-encoding data features changes the accessible input function space adding higher harmonics to the Fourier series. This enables the model to learn such type of functions, and mathematically amounts to a change in the structure constants $\Gamma_{\mu,\nu}$, now with support on larger input or parameter spaces, which can be captured by our definition of bias. We provide further examples of this mechanism, with explicit calculations of the model structure constants for instances of QNNs, in the Supplementary Material: our analysis highlights specific circuit design choices that can be used to bias the QNN outputs, provided one has some knowledge of the data-generating function.

Finally, we comment here on the relation between our definition of bias and the more general concept of inductive bias in ML. Inductive biases typically result from architectural choices (approximately) restricting the model function to have specific symmetries or live on specific manifolds. Our construction defines model bias in a stricter form, i.e. as an Euclidean distance between model and target function space. This definition gives us the advantage of having a tunable parameter for conducting our numerical experiments. Including functional symmetries (such as translations or mirror symmetries)

can in principle also be done at the level of the structure constants, bringing us closer to the more general definition of inductive bias. However, in order to showcase the basic functional mechanisms of the ED-bias interplay, we decide to leave this as a possible direction of future work.

2.4. Tensorized models

We now briefly introduce the concept of *tensorized* models, which allow us to circumvent the exponential costs (in N and M, since $D = d^N$ and $K = \tilde{d}^M$) of constructing and storing the full Γ in our simulations. This enables the numerical study of problem instances with larger number of features N and of parameters M. The main idea is to decompose the structure constants Γ as a TN [60–62] with a finite bond dimension χ. A TN decomposition of Γ is enabled by the structure of the input and parameter functions' spaces, which are constructed as tensor products of the spaces 'local' to each input feature x_n and parameter θ_m, spanned by the local basis functions $e^{(n)}_{\mu_n}(x_n)$ and $\iota^{(m)}_{\nu_m}(\theta_m)$. To construct a TN decomposition of Γ, we consider the following approximation

$$\Gamma_{\mu,\nu} \approx \sum_{\rho=1}^{D}\sum_{\sigma=1}^{\chi} \underbrace{U_{\mu,\rho}\, T_{\rho,\sigma}}_{\text{ortho. rotation + isometry}} \underbrace{s_{\sigma}}_{\text{corr. spectrum}} \underbrace{\left[V^{\top}\right]_{\sigma,\nu}}_{\text{map from param. space}} , \tag{22}$$

which differs from equation (9) by the presence of an isometry T, which is a linear mapping from the D-dimensional input functions' space to a χ-dimensional reduced space, building an internal TN representation of the input functions' space.

We decompose the matrices U, T and V as products of low-rank tensors as follows. The matrix V containing the right-singular values of Γ is expressed as a tensor-train (also known as matrix product state) [63, 64], the orthogonal matrix U containing the left-singular values of Γ as an orthogonal matrix product operator (MPO) [65–67], and the isometry T can be as a tree TN (TTN) [68–70]. We note that the TN decompositions used here are not the only option, and different ones are possible. In appendix D we provide the explicit expressions of these decompositions, together with their diagrammatic representation, the conditions the individual tensors need to fulfill in order to respect the orthogonality of the decomposed matrices, and details on how we practically generate random instances of those. In the Supplementary Material we show how to adapt the procedure described in the previous section to generate biased and unbiased tensorized models. There, we also numerically check that the bond dimension χ does not have a significant effect on the ED. This means that also for tensorized models the ED is controlled by the decay property of the correlation spectrum, which allows us to use the same procedure as described before for tuning the models' ED.

3. Results

In this section we present our main results on the effects of the ED on the training of regression models with gradient-based methods, and the interplay with the model's bias towards the regression task at hand. The regression tasks investigated here consist in training the parameters $\boldsymbol{\theta}$ of a regression model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$, of the form introduced before, to learn a data-generating function $y(\boldsymbol{x})$. We consider the situation where we have $\mathfrak{n}_{\text{train}}$ input-output pairs $\{\boldsymbol{x}_i, y(\boldsymbol{x}_i)\}_{i=1,\dots,\mathfrak{n}_{\text{train}}}$ that we use for training the model, using the mean squared error (MSE) as loss function

$$\text{MSE} = \frac{1}{\mathfrak{n}_{\text{train}}} \sum_{i=1}^{\mathfrak{n}_{\text{train}}} \left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y(\boldsymbol{x}_i)\right)^2 . \tag{23}$$

The numerical results presented in this section are obtained using Fourier regression models, i.e. with input and parameter basis functions $e^{(n)}_{\mu_n}(x_n)$ and $\iota^{(m)}_{\nu_m}(\theta_m)$ given by equations (7) and (8). For simplicity, we restrict to the case where the 'local' basis sets $\mathcal{B}_n$ and $\tilde{\mathcal{B}}_m$, as well as the 'local' frequency sets Ω_n and $\tilde{\Omega}_m$, are independent of the feature index n and the parameter index m. The $\mathfrak{n}_{\text{train}}$ input points are chosen uniformly in $[-\pi, \pi]^N$, and the models are then trained with the Adam optimizer [71].

In order to study the interplay of model bias and ED and their effects on training in a statistically sound manner, we perform several training experiments with randomly drawn data-generating function $y(\boldsymbol{x})$ and structure constants Γ. Specifically, for chosen dimensions D and K (fixed by the choice of N, d, M and $\tilde{d}$), we draw random instances of $y(\boldsymbol{x})$, and many random instances of models, specified by Γ, with different degree of bias towards $y(\boldsymbol{x})$. For any given degree of bias, we train several random instances of full models (equation (19)) and cutoff models (equation (20)), in order to compare the

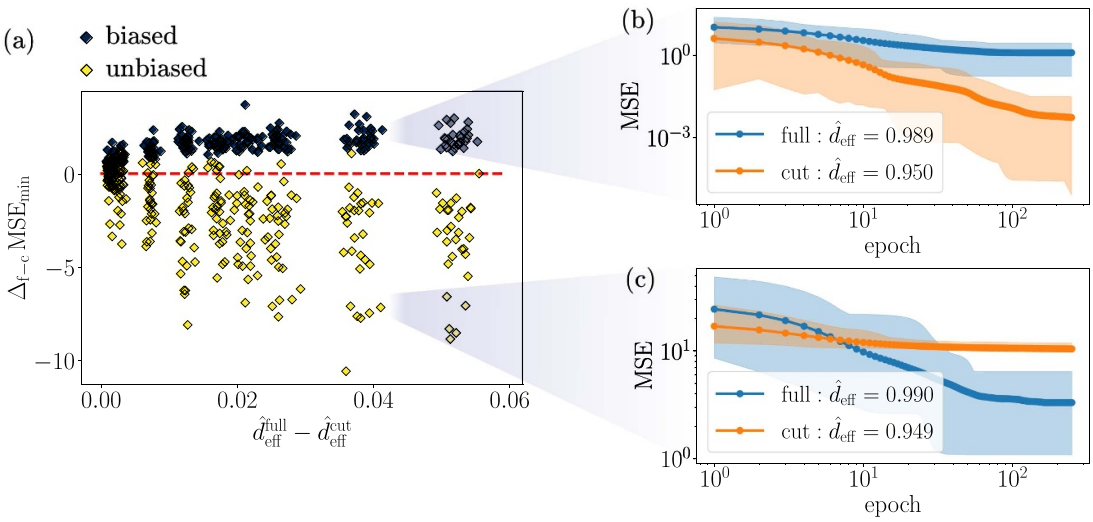


Figure 4. (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ for different values of $\hat{d}^{\mathrm{full}}_{\mathrm{eff}} - \hat{d}^{\mathrm{cut}}_{\mathrm{eff}}$, for biased (blue points) and unbiased (yellow points) models. Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ averaged over 30 training instances starting from randomly chosen parameters, for a single random model realization, i.e. a random Γ uniformly drawn from $[-1,+1]^{D\times K}$. The red line serves as a guide for the eye for zero MSE difference. The spread over training instances is not shown for clarity, but reported in panels (b) and (c), as representatives of all points in (a). (b) Training curves for a random biased model realization, with full model in blue and cutoff model in orange. (c) Training curves for a random unbiased model realization, with full model in blue and cutoff model in orange. The shading corresponds to the spread over 30 training instances. For these plots, $N = 1$, $\Omega = \{1,\ldots,8\}$ ($d = 17$), $\tilde{\Omega} = \{1\}$ ($\tilde{d} = 3$), $M = 7$, $R = 6$, $\mathrm{n}_{\mathrm{train}} = 25$ with a batch size of 5.

training dynamics of models with higher and lower ED, respectively. Furthermore, every model instance is trained several times starting from random parameter configurations: this probes the robustness of the training dynamics against different random starting points, providing an averaged picture of the severity of local minima in the optimization landscape. To visualize the comparison between models with higher and lower ED, for any given degree of bias we consider the minimum MSE attained during training as a proxy for the training quality, denoted with $\mathrm{MSE}^{\mathrm{full}}_{\mathrm{min}}$ and $\mathrm{MSE}^{\mathrm{cut}}_{\mathrm{min}}$ for full and cutoff models, respectively, and study the difference

$$\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}} = \mathrm{MSE}^{\mathrm{full}}_{\mathrm{min}} - \mathrm{MSE}^{\mathrm{cut}}_{\mathrm{min}}\,, \tag{24}$$

as a function of the difference in the ED between full and cutoff models, i.e. $\hat{d}^{\mathrm{full}}_{\mathrm{eff}} - \hat{d}^{\mathrm{cut}}_{\mathrm{eff}}$. A positive value of $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ implies that the full model (with higher ED) trains to a higher MSE compared to the cutoff one, i.e. the model with lower ED model has a better training performance. Conversely, a negative $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ implies a better performance of models with higher ED. Note that this analysis of model training indeed concerns whether the training is successful, i.e. eventually reaches a low value of the loss function, as a proxy of how the exploration of the parameter space was successful in finding the data generating function. While this metric is already sufficient to separate the training behavior of models with high and low ED and bias, as we show in the following, it is important to notice that a more local account of the optimization dynamics, achieved by monitoring the FIM/NTK spectrum or the local ED [55] during training, could provide more information on the properties of the optimization landscape, and is thus a natural extension of the present work.

3.1. Results with full random structure constants

We start by presenting our results obtained by drawing random structure constants Γ uniformly drawn in $[-1,+1]^{D\times K}$. As we show in figure 4(a) there is a clear difference in the effect of a higher ED on the training dynamics between biased and unbiased models. Specifically, for biased models (shown in blue) the value of $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ is positive and increases with increasing $\hat{d}^{\mathrm{full}}_{\mathrm{eff}} - \hat{d}^{\mathrm{cut}}_{\mathrm{eff}}$. Conversely, for unbiased models (shown in yellow) the value of $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ is negative (with high probability) and decreases with increasing $\hat{d}^{\mathrm{full}}_{\mathrm{eff}} - \hat{d}^{\mathrm{cut}}_{\mathrm{eff}}$. That is, in the biased case, models with lower ED train to a lower MSE (as can be seen in figure 4(b) for a specific random model realization), whereas in the unbiased case a higher ED is beneficial for training (as shown in figure 4(c)). These results confirm the following intuitive expectation: a model which is biased towards the data-generating function $y(\boldsymbol{x})$ and which at the same time has a lower ED, can effectively explore a functions' space more constrained around $y(\boldsymbol{x})$, and

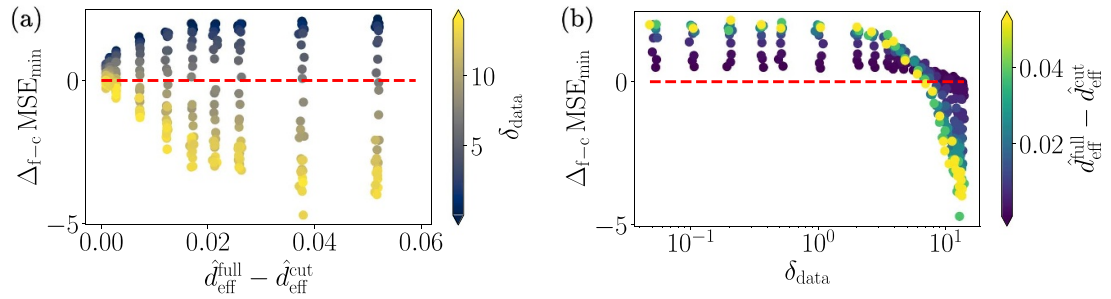


Figure 5. (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ for different values of $\hat{d}^{\mathrm{full}}_{\mathrm{eff}} - \hat{d}^{\mathrm{cut}}_{\mathrm{eff}}$, for different values of δ_{data} (color scale). Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ averaged over 30 training instances for 30 random model realization. (b) Same as panel (a) but resolved as a function of δ_{data}. The red line serves as a guide for the eye for zero MSE difference. For these plots, $N = 1$, $\Omega = \{1, \ldots, 8\}$ ($d = 17$), $\tilde{\Omega} = \{1\}$ ($\tilde{d} = 3$), $M = 7$, $R = 6$, $\mathfrak{n}_{\mathrm{train}} = 25$ with a batch size of 5.

is therefore easier to train. Conversely, an unbiased model with a higher ED can explore a larger space of function, which makes it probabilistically easier to approximately fit an (in principle) unrelated data-generating function.

To complement these results, we also investigate how a partial bias towards the data-generating function (as defined in equation (21)) influences the training, for different values of $\hat{d}^{\mathrm{full}}_{\mathrm{eff}} - \hat{d}^{\mathrm{cut}}_{\mathrm{eff}}$. As shown in figure 5(a), we again observe that a higher ED is beneficial in the case of unbiased models (where $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ is negative), whereas increasing the model's bias (i.e. decreasing δ_{data}) results in better training performances for models with lower ED (where $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ is positive). Furthermore, as shown in figure 5(b), there is an extended regime in terms of values of δ_{data} where $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ is positive, indicating the stability of our results also in the case of a not perfectly biased model (i.e. $\delta_{\mathrm{data}} > 0$). This further corroborates the aforementioned intuition that a large ED, as a measure of effectively explorable functions' space, is beneficial for training models with low bias towards the regression task under study, whereas models strongly biased towards the task do benefit from a smaller ED. Further results showcasing the interplay between model bias and ED are provided in the Supplementary Material for different model specifications (i.e. different N, d, M and $\tilde{d}$), and confirm the findings presented here.

3.2. Results with tensorized random structure constants

Here we conduct numerical experiments analogous to those in the previous section, adopting the TN decomposition of the structure constants Γ discussed in section 2.4, to study how the problem size, i.e. the number of features N and of parameters M, affects our results. The training experiments are set up in the same manner as those in in the previous section, with the only difference that the models correspond now to random instances of the TN representing Γ. As shown in figure 6, the conclusions drawn in section 3.1 apply independently of the size of the problem and of the models' details. Specifically, the results shown in figure 6(a) are consistent with those of figure 4(a), showing that in the biased case $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ is positive, i.e. models with lower ED train to a lower MSE, whereas in the unbiased case $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ is negative, i.e. models with higher ED have better training performance. We refer the reader to appendix D and to the Supplementary Material for further numerical results on training tensorized models, which confirm the findings presented here. The dependence of $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ on M and D is analyzed in the next section.

3.3. Effect of model under- and overparameterization

In this section we investigate how the ED-bias interplay in training changes with the input function space dimension D and the number of parameters M. This allows us to assess the behavior across the transition from the underparameterized ($M < D$) to the overparameterized ($M \geqslant D$) regime [56]. The results are shown in figure 7, where we show how $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ behaves as a function of M and D for different training set sizes $\mathfrak{n}_{\mathrm{train}}$. Overall, we observe that the sign of $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ remains robust across the tested hyperparameters, whereas the magnitude indeed depends on the values of M, D and $\mathfrak{n}_{\mathrm{train}}$. The dependence of $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ on M and D is mostly visible in the biased case (panels (a) and (c)). In particular, $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ rapidly decreases as a function of M reaching an approximately constant (but still positive) value for $M \geqslant D$ (figure 7(a)). The regime $M \geqslant D$ corresponds to the overparameterized regime where the number of parameters is larger than the dimension of the space of functions accessible by the

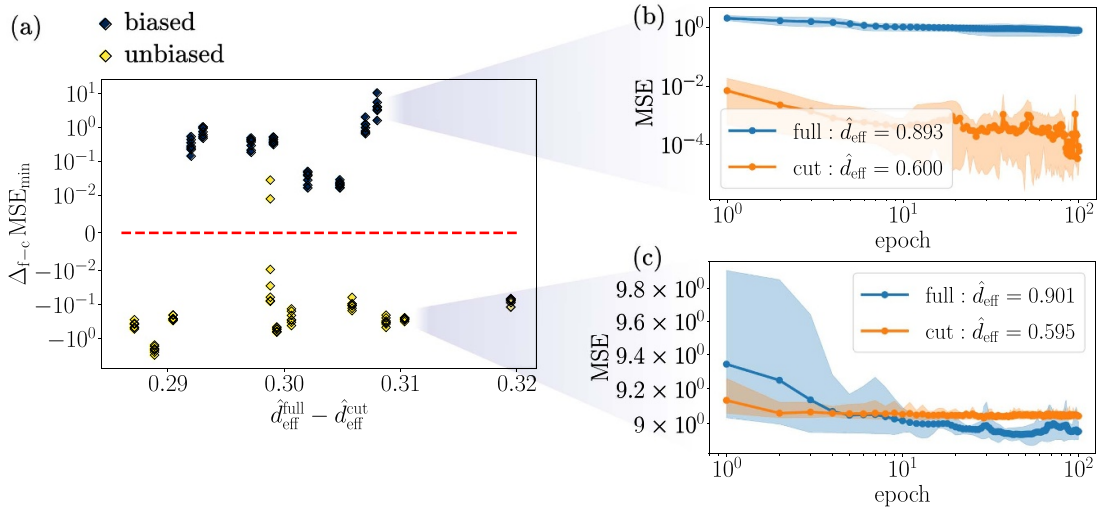


Figure 6. (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ for different values of $\hat{d}^{\mathrm{full}}_{\mathrm{eff}} - \hat{d}^{\mathrm{cut}}_{\mathrm{eff}}$, for biased (blue points) and unbiased (yellow points) models. Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ averaged over 7 training instances starting from randomly chosen parameters, for a single random model realization, i.e. a random right-normalized tensor train representing V, a random MPO representing U and a random TTN representing T. The red line serves as a guide for the eye for zero MSE difference. The spread over training instances is not shown for clarity, but reported in panels (b) and (c), as representatives of all points in (a). (b) Training curves for a random biased model realization, with full model in blue and cutoff model in orange. (c) Training curves for a random unbiased model realization, with full model in blue and cutoff model in orange. The shading corresponds to the spread over 7 training instances. For these plots, $N = 4$, $\Omega = \{1, \ldots, 3\}$ $(d = 7)$, $\tilde{\Omega} = \{1\}$ $(\tilde{d} = 3)$, $M = 24$, $R = 2$, $\chi = 30$, $\mathrm{n}_{\mathrm{train}} = 6^4$ with a batch size of 12.

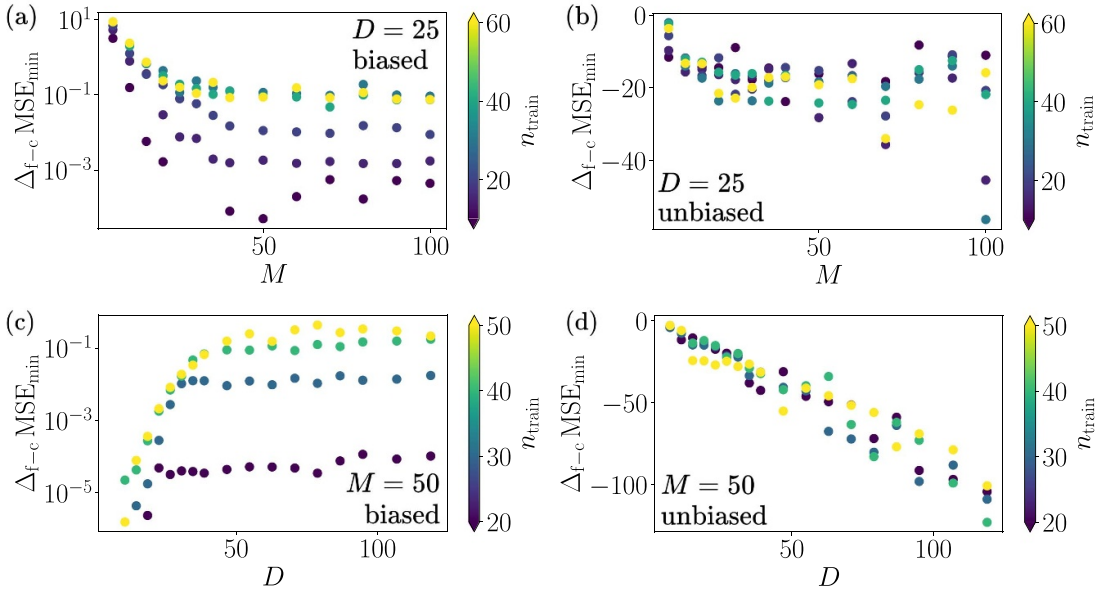


Figure 7. (a) and (b) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ vs M for biased and unbiased models, respectively. Here $N = 1$, $D = 25$, $\tilde{\Omega} = \{1\}$ $(\tilde{d} = 3)$, $R = 3$ and $\chi = 50$. (c) and (d) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ vs D for biased and unbiased models, respectively. Here $N = 1$, $M = 50$, $\tilde{\Omega} = \{1\}$ $(\tilde{d} = 3)$, $R = 2$ and $\chi = 120$. In all panels, each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ averaged over 30 training instances and 30 random model realizations, i.e. random right-normalized tensor trains representing V. The color scale refers to the number of training data used $\mathrm{n}_{\mathrm{train}}$, with a batch size of 5.

model. Hence, several redundant parameters exist, which make the training of a model easier, independently of whether this is has a high or a low ED. As a consequence, in this regime we observe a reduction of $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ since the model with high ED can effectively rely on more redundant parameters for finding the global minimum of the loss. Consistently, $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ rapidly increases as a function of D reaching an approximately constant value for $D \geqslant M$ (figure 7(c)). The regime $D \geqslant M$ corresponds to an underparameterized regime where, in the biased case, the absence of redundant parameters results in a better training performance of models with low ED. This difference between under- and overparameterized regimes in the biased case can equivalently be explained from the perspective of the NTK,

and the fact that in the underparameterized regime the NTK has $D-M$ 'frozen' (i.e. with zero eigenvalue) modes. We refer the reader to the Supplementary Material for a qualitative explanation of this phenomenon. There, we also discuss the dependence on $\mathfrak{n}_{\text{train}}$, explaining how the number of training samples affects the NTK and the training dynamics, and why in the biased case a lower $\mathfrak{n}_{\text{train}}$ reduces the gap $\Delta_{\text{f}-\text{c}}\text{MSE}_{\text{min}}$ between high and low ED models.

Turning to the unbiased case, figure 7(c) shows a negative $\Delta_{\text{f}-\text{c}}\text{MSE}_{\text{min}}$ which slightly decays for increasing $M \leqslant D$ at fixed D. Again, we attribute this to the high ED models being able to explore progressively more directions in the D dimensional input space as M increases, more effectively than low ED ones. Increasing M above D does not increase the number of different explorable directions, hence does not result in a further separation between high and low ED models. Figure 7(d) shows that $\Delta_{\text{f}-\text{c}}\text{MSE}_{\text{min}}$ is negative and decays with increasing D for fixed M in both regimes. In this case, models with increased D have a larger base space where a better approximation for the target function can be found, and a higher ED results in an effectively larger portion of this space being available (hence lowering $\Delta_{\text{f}-\text{c}}\text{MSE}_{\text{min}}$). In the unbiased case, the effect of changing $\mathfrak{n}_{\text{train}}$ is not as significant, and we provide a tentative explanation for this in the Supplementary Material.

4. Conclusion and outlook

In this work we investigated how the ED, calculated from the FIM, influences the training of regression models using gradient descent methods. Specifically, we studied the interplay between the ED and the bias that a model has towards the regression task at hand, and were able to draw the following main conclusions. A high ED, corresponding to a high model capacity of exploring independent directions in model space, is beneficial for training in the low bias regime, i.e. when the model is largely agnostic to the data-generating function to be learned. Conversely, in the biased regime, i.e. when the model's structure is well suited to the problem's data-generating function, a low ED does result in better training performance. These results confirm, in a quantitative manner, the intuitive expectation that if the space a model has access to is constrained (i.e. a model with low ED) around the problem's data-generating function (i.e. a biased model), training the model effectively becomes easier. These findings are corroborated by the analysis of the NTK and its effects on the training dynamics, which we show in the Supplementary Material: by relating model bias with the alignment between NTK eigenfunctions and data, and using the equivalence between FIM and NTK spectra, we can construct a simplified setting qualitatively capturing the phenomena observed here. Thus, a high ED does not always result in a model's faster training, which we interpret as a further sign of the difficulty of defining a task-independent evaluation metric that can assess a model's performance prior to its training.

We foresee several possible directions for extending the presented results. First, it is important to comment on the fact that our analysis is based on comparing ED, bias and training performance of models with the same underlying structure (i.e. a finite number of chosen basis functions for the inputs' and parameters' functions space). Our choice is motivated by the need of comparing models where bias and ED can be controlled, while eliminating other potential sources of fluctuations coming from different architectural choices. Nevertheless, due to the equivalence between Fourier models and QNNs, our results can already provide guidance on practical QNN design choices that can be used to bias the model outputs towards specific target functions, via data re-uploading and the use of entangling operations or correlated measurements (see Supplementary Material for details). The design of QNNs with tunable ED requires a comprehensive characterization of how the QNN architecture influences the structure constants in QNNs, which at the moment is still lacking.

While the objective of our work is to establish the mechanism for the ED-bias interplay in training, it is important to test this in progressively more real-world-related scenarios, under larger input spaces and different input data distributions. While the ED-bias interplay is a general mechanism, investigating different and more realistic settings might uncover important details that can help the design of models for practical tasks. Analogous experiments can be performed using actual (simulated) QNN models or even classical NNs. While the use of QNNs would still amount to Fourier models with different (yet unknown) distributions in the structure constants, we find the investigation of the present mechanism for classical NNs, and for which conditions and data it may emerge, of high interest. We would find it also interesting to investigate the same mechanism when comparing inherently different models, i.e. models accessing different types of function spaces. For example, designing such experiments comparing quantum and classical NNs could yield more insights on the function classes, and thereby the type of data, most suitable to these two ansatzes.

Going beyond the analysis of the optimization dynamics, an important future direction would be studying how the generalization error changes for models with different ED and bias. While the ED can be used to bound generalization errors [27, 55], our results qualitatively suggest the model-task bias as an important factor that may strongly affect the generalization ability (as is the case for kernel methods [37]). Obviously, all the analysis and discussion presented here admit natural extensions to classification (see the Supplementary Material for suggestions on a potential generalization) and generative models. Furthermore, it would be interesting to investigate the questions addressed here also in the context of natural gradient descent [48, 51], which could help to develop a deeper understanding of the ED-bias interplay in training ML models.

There are also several opportunities for establishing a deeper connection between our findings and the theory of QML. One interesting direction could be investigating the connections with (quantum) kernel methods [72], for which results on the effects of inductive bias (kernel-task alignment) on the training and generalization performance have already been established [37, 73]. Other important aspects to be addressed in the future concern the connection to existing works on generalization [10, 11, 74], overfitting [74, 75], overparameterization [56], and barren plateaus [76–78] in QML. Furthermore, since in our numerical analysis we primarily focused on Fourier models, we note that there exist several works investigating the use of Fourier features for *dequantizing*, or building *classical surrogates*, of QML models [79–81]. In the context of our work, it would be interesting to understand what features of a QNN make it dequantizable via the tensorized model introduced here, generalizing recent works [82] and enabling further understanding on the function classes encompassed by QML models.

Finally, to connect the present results with quantum hardware implementations, we remark that the Fourier series representation of QNNs remains largely valid also when realistic hardware noise and certain types of gate errors are considered. A more fundamental issue remains that of finite sampling errors, which can hinder the training of a model when gradients are poorly resolved. While we highlight strategies to mitigate these effects [24, 83], we also do not exclude the possibility that moderate finite sampling noise can act as further stochastic term helping the model in exploring the accessible function space. Investigating how this influences the ED-bias interplay discussed here is left as interesting future work.

Acknowledgments

This project was made possible by the DLR Quantum Computing Initiative and the Federal Ministry for Economic Affairs and Climate Action; qci.dlr.de/projects/klim-qml. V E was funded by the European Research Council (ERC) Synergy Grant 'Understanding and Modelling the Earth System with Machine Learning (USMILE)' under the Horizon 2020 research and innovation programme (Grant Agreement No. 855187). V E was additionally supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through the Gottfried Wilhelm Leibniz Prize awarded to Veronika Eyring (Reference No. EY 22/2-1). This work used resources of the Deutsches Klimarechenzentrum (DKRZ) granted by its Scientific Steering Committee (WLA) under project ID bd1179.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://doi.org/10.5281/zenodo.21511180; https://github.com/EyringMLClimateGroup/pastori26MLST_fim-training-fourier-models [89].

Supplementary Material available at: https://doi.org/10.1088/2632-2153/ae8fb7/data1.

Appendix A. Fourier series representation of QNN

Here we sketch the derivation of the Fourier series representation for a function $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ obtained as output of a QNN as in equation (5), focusing for simplicity on the case of a one-dimensional input. This can be easily generalized to more input features [43, 45] following the same derivation. We consider L layers of data re-uploading [43, 46] with unitary $\hat{U}_{\boldsymbol{\theta}}(x)$ expressed as

$$\hat{U}_{\boldsymbol{\theta}}(x) = \prod_{\ell=1}^{L} \left[\hat{S}(x)\, \hat{W}^{(\ell)}\left(\boldsymbol{\theta}^{(\ell)}\right) \right], \tag{A.1}$$

where the input x and trainable parameters $\boldsymbol{\theta}^{(\ell)}$ are encoded in $\hat{S}(x)$ and $\hat{W}^{(\ell)}(\boldsymbol{\theta}^{(\ell)})$ as angles in rotation gates. The vector $\boldsymbol{\theta}$ summarizes the dependence on all $\boldsymbol{\theta}^{(\ell)}$, i.e. $\boldsymbol{\theta} = \{\boldsymbol{\theta}^{(\ell)}\}_{\ell=1,\dots,L}$. We can write $\hat{W}^{(\ell)}(\boldsymbol{\theta}^{(\ell)}) = \prod_{j_\ell=1}^{J_\ell} \hat{G}_{j_\ell}^{(\ell)}(\theta_{j_\ell}^{(\ell)})$, and following [43, 45] we switch to the diagonal representations of $\hat{S}(x)$ and $\hat{G}_{j_\ell}^{(\ell)}(\theta_{j_\ell}^{(\ell)})$. These have eigenvalues $\{e^{-i\lambda_\alpha x}\}_{\alpha=1,\dots,\mathcal{D}}$ and $\{e^{-i\eta_\beta^{(\ell,j_\ell)}\theta_{j_\ell}^{(\ell)}}\}_{\beta=1,\dots,\mathcal{D}}$, respectively, which make the trigonometric dependence of $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ on the inputs and parameters evident. Using these, we can calculate the components of the state $\hat{U}_{\boldsymbol{\theta}}(x)|0\rangle$ expanded in terms of the functions $e^{-i\Lambda_{\boldsymbol{\alpha}} x}$ and $e^{-i\boldsymbol{\eta}_{\boldsymbol{\beta}}\cdot\boldsymbol{\theta}}$, with the shorthand notation $\boldsymbol{\alpha} = (\alpha_1,\dots,\alpha_L)$, $\boldsymbol{\beta} = (\beta_1,\dots,\beta_M)$, $\Lambda_{\boldsymbol{\alpha}} = \sum_\ell \lambda_{\alpha_\ell}$ and $\boldsymbol{\eta}_{\boldsymbol{\beta}} = (\eta_{\beta_1}^{(1)},\dots,\eta_{\beta_M}^{(M)})$. We refer to the Supplementary Material for the details of the calculation, and we report here the result

$$f_{\boldsymbol{\theta}}(x) = \sum_{\boldsymbol{\alpha},\boldsymbol{\alpha}'} \sum_{\boldsymbol{\beta},\boldsymbol{\beta}'} \tilde{\Gamma}_{(\boldsymbol{\alpha};\boldsymbol{\alpha}'),(\boldsymbol{\beta};\boldsymbol{\beta}')}\, e^{i(\Lambda_{\boldsymbol{\alpha}}-\Lambda_{\boldsymbol{\alpha}'})x} e^{i(\boldsymbol{\eta}_{\boldsymbol{\beta}}-\boldsymbol{\eta}_{\boldsymbol{\beta}'})\cdot\boldsymbol{\theta}}\,, \tag{A.2}$$

which has the same form as equation (6) in the main text. As an explicit example of the sets of frequencies accessible to the model, we consider the case where input features and variational parameters are encoded as angles of single qubit rotations of the form $e^{-i\frac{\phi}{2}\boldsymbol{n}\cdot\hat{\boldsymbol{\sigma}}}$ (with ϕ being the feature/parameter to be encoded, $\boldsymbol{n}$ an arbitrary rotation axis and $\hat{\boldsymbol{\sigma}}$ the vector of Pauli matrices). In this case, the eigenvalues of the generators of $\hat{S}(x)$ and $\hat{G}_{j_\ell}^{(\ell)}(\theta_{j_\ell}^{(\ell)})$ are $\lambda_{\alpha_\ell} \in \left\{-\frac{1}{2},+\frac{1}{2}\right\}$ and $\eta_{\beta_m}^{(m)} \in \left\{-\frac{1}{2},+\frac{1}{2}\right\}$, respectively. For the dependence on the inputs x we therefore have $\lambda_{\alpha_\ell} - \lambda_{\alpha_{\ell'}} \in \left\{-1,0,+1\right\}$, hence $\Lambda_{\boldsymbol{\alpha}} - \Lambda_{\boldsymbol{\alpha}'} \in \left\{-L,-L+1,\dots,L-1,L\right\}$, thus yielding the local basis set $\mathcal{B} = \{1, \sqrt{2}\cos(x),\dots,\sqrt{2}\cos(Lx), \sqrt{2}\sin(x),\dots,\sqrt{2}\sin(Lx)\}$. Similarly, for the dependence on the parameters θ_m we have $\eta_{\beta_m}^{(m)} - \eta_{\beta_m'}^{(m)} \in \left\{-1,0,+1\right\}$, which yields the local basis set for the parameters $\tilde{\mathcal{B}}_m = \{1, \sqrt{2}\cos\theta_m, \sqrt{2}\sin\theta_m\}$.

The Fourier representation is not restricted to the noiseless, pure-state setting. In the presence of hardware noise, the circuit evolution can be described at the level of density matrices by a composition of quantum channels

$$\hat{\rho}_{\boldsymbol{\theta}}(x) = \bigcirc_{\ell=1}^{L} \left[\mathcal{E}_\ell \circ \mathcal{S}(x) \circ \mathcal{W}^{(\ell)}\left(\boldsymbol{\theta}^{(\ell)}\right)\right](\hat{\rho}_0)\,, \tag{A.3}$$

with $\mathcal{E}_\ell$ being the noise channels, and $\mathcal{S}(x)$ and $\mathcal{W}^{(\ell)}(\boldsymbol{\theta}^{(\ell)})$ the unitary channels corresponding to $\hat{S}(x)$ and $\hat{W}^{(\ell)}(\boldsymbol{\theta}^{(\ell)})$, respectively, acting as, e.g. $\mathcal{S}(x)(\cdot) = \hat{S}(x)\cdot\hat{S}^\dagger(x)$. The QNN output is given by the measurement output $\hat{\rho}_{\boldsymbol{\theta}}(x) = \mathrm{tr}\big(\hat{M}\hat{\rho}_{\boldsymbol{\theta}}(x)\big)$. Since both quantum channels and the final measurement map are linear, and angle-encoding gates still introduce only trigonometric dependence on x and $\boldsymbol{\theta}$, the output retains the same Fourier-model form. This means that hardware noise does not invalidate the Fourier-series description, but instead modifies the structure constants $\Gamma_{\mu,\nu}$. Similarly, also random gate rotation errors can effectively be described by dephasing channels with strength depending only on the variance of the fluctuations [84], and therefore they also do not affect the Fourier series representation but only the structure constants. From the trainability perspective, noise can qualitatively alter the optimization landscape and may induce noise-induced barren plateaus [85], leading to exponentially vanishing gradients in the number of qubits and circuit depth.

Appendix B. Properties of FIM for regression models

We provide here a sketch of the derivation of the properties of the FIM discussed in section 2.2.2. The details of the calculations are given in the Supplementary Material. Our starting point is equation (14), which can be rewritten as $F_{j,k}(\boldsymbol{\theta}) = \big(\boldsymbol{\iota}(\boldsymbol{\theta})\big|\mathcal{F}^{(j,k)}\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)$, with $\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)$ the K-dimensional vector with components $\iota_\nu(\boldsymbol{\theta})$ and

$$\mathcal{F}^{(j,k)} = B_j^\top V S^2 V^\top B_k = \sum_{\rho=1}^{D} s_\rho^2\, B_j^\top P_\rho B_k\,, \tag{B.1}$$

where $B_j = I_{\tilde{d}}^{(1)} \otimes I_{\tilde{d}}^{(2)} \otimes \dots \otimes \beta^{(j)} \otimes \dots \otimes I_{\tilde{d}}^{(M)}$ (with $I_{\tilde{d}}^{(k)}$ being the $\tilde{d}$-dimensional identity matrix acting on the function space 'local' to the kth parameter), and P_ρ being the projection on the subspace spanned by the vector $V_{\cdot,\rho}$. For showing equation (16), it is sufficient to note that thanks to the presence of the projection P_ρ, the FIM $F(\boldsymbol{\theta})$ can be expressed as a sum of at most D linearly independent $M\times M$ projections, which implies that its rank can be at most D (if $M > D$). For showing equation (17), we use

results from random matrix theory [86–88] to derive the expectation value and the variance of the FIM elements over random realizations of $V \in \mathrm{O}(K)$. In particular, we show that

$$\mathbb{E}_{V\in\mathrm{O}(K)}\left[F_{j,k}(\boldsymbol{\theta})\right] = \frac{\mathrm{tr}\left(S^2\right)}{K}\left(\boldsymbol{\iota}(\boldsymbol{\theta})\left|B_j^\top B_k\right|\boldsymbol{\iota}(\boldsymbol{\theta})\right), \tag{B.2}$$

and

$$\begin{aligned}\mathrm{Var}_{V\in\mathrm{O}(K)}\left[F_{j,k}(\boldsymbol{\theta})\right] = \frac{\mathrm{tr}\left(S^4\right)}{K^2}&\left[\left(\boldsymbol{\iota}(\boldsymbol{\theta})\left|B_j^\top B_k\right|\boldsymbol{\iota}(\boldsymbol{\theta})\right)^2\right.\\ &\left.+\left(\boldsymbol{\iota}(\boldsymbol{\theta})\left|B_j^\top B_j\right|\boldsymbol{\iota}(\boldsymbol{\theta})\right)\left(\boldsymbol{\iota}(\boldsymbol{\theta})\left|B_k^\top B_k\right|\boldsymbol{\iota}(\boldsymbol{\theta})\right)\right]\\ &+\mathcal{O}\left(K^{-3}\right).\end{aligned} \tag{B.3}$$

Then, using the orthonormality of the basis functions $\iota_{\nu_m}^{(m)}(\theta_m)$, one has that $(\boldsymbol{\theta})\left|B_j^\top B_k\right|\boldsymbol{\iota}(\boldsymbol{\theta})\right) \in \mathcal{O}(K)$ if $j = k$, being suppressed otherwise, which results in equation (17).

Appendix C. Construction of biased and unbiased models

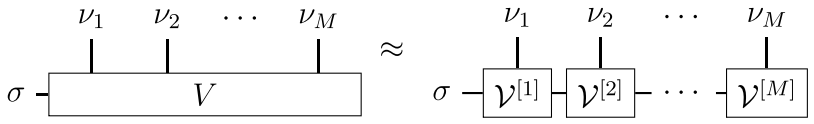
In this appendix we provide more details on the construction of biased and unbiased models. Further details can be found in the Supplementary Material. We start from the expression of the data-generating function $y(\boldsymbol{x})$ provided in equation (18). In order to construct a $K \times D$ matrix $V^{(\mathrm{d})}$ with orthonormal columns and satisfying the property $\sum_\nu V^{(\mathrm{d})}_{\nu,\rho}\iota_\nu(\boldsymbol{\theta}^*) = 0$ for $\rho = R+1,\ldots,K$, it is sufficient to construct is as $V^{(\mathrm{d})} = \begin{bmatrix} V & W \end{bmatrix}$, i.e. by horizontally stacking a $K \times R$ matrix V and a $K \times (D-R)$ matrix W both with orthonormal columns, satisfying $W^\top V = 0$ and with $\sum_\nu W_{\nu,\sigma}\iota_\nu(\boldsymbol{\theta}^*) = 0$. Once V has been constructed (e.g. randomly drawn), the two conditions on W can be implemented using Gram–Schmidt orthogonalization from a set of $(D-R)$ randomly chosen vectors. For constructing partially biased models, we replace $V^{(\mathrm{d})}$ in equation (18) with $V^{(\mathrm{d})}_\epsilon$ obtained by adding a random perturbation of strength ϵ to its elements. This is done by setting $V^{(\mathrm{d})}_\epsilon = \mathrm{ortho}(V^{(\mathrm{d})} + \epsilon G)$, where G is a $K \times D$ matrix with standard Gaussian entries and $\mathrm{ortho}(\cdot)$ refers to the process of Gram–Schmidt orthogonalization of the columns of the argument.

Appendix D. Tensorized models and additional results

Here we provide details on the construction of tensorized models together with additional numerical results on the effects of bias and ED on their training dynamics. More details on their construction and further numerical results can be found in the Supplementary Material. The tensor-train representation of V has the following form

$$V_{\nu,\sigma} \approx \sum_{a_1,\ldots,a_{M-1}=1}^{\chi} \mathcal{V}^{[1]\,\nu_1}_{\sigma,a_1}\mathcal{V}^{[2]\,\nu_2}_{a_1,a_2}\ldots\mathcal{V}^{[M]\,\nu_M}_{a_{M-1},1}, \tag{D.1}$$

where $\mathcal{V}^{[m]}$ are rank-3 tensors satisfying the right-normalization condition, in order for V to have orthonormal columns. This decomposition of V admits the following graphical representation

$V_{\nu,\sigma} =$ 

The orthogonal matrix U is decomposed as an orthogonal MPO

$$U_{\mu,\rho} \approx \sum_{a_1,\ldots,a_{N-1}=1}^{\chi} \mathcal{U}^{[1]\,\mu_1,\rho_1}_{1,a_1}\mathcal{U}^{[2]\,\mu_2,\rho_2}_{a_1,a_2}\ldots\mathcal{U}^{[M]\,\mu_N,\rho_N}_{a_{N-1},1}, \tag{D.3}$$

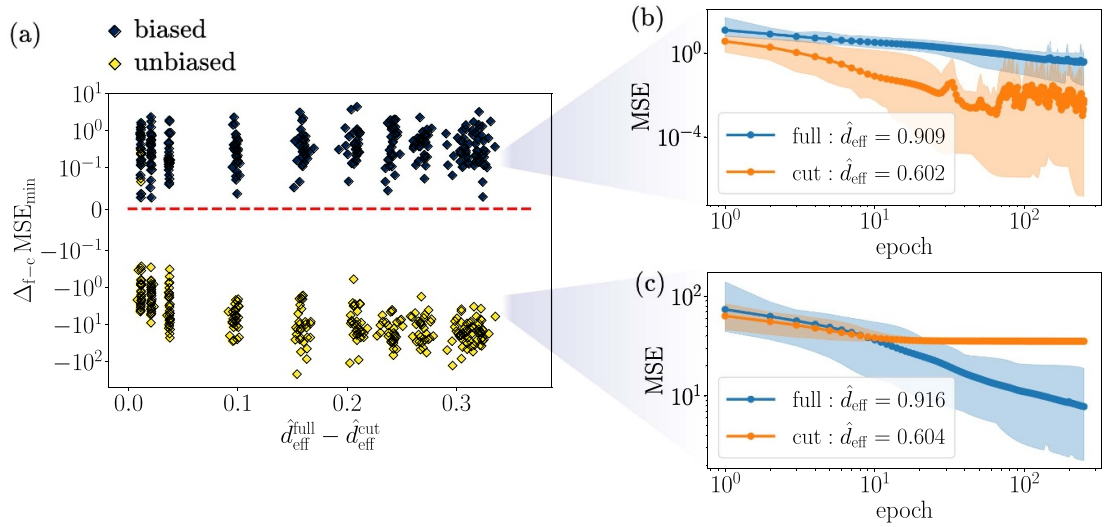


Figure D1. (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ for different values of $\hat{d}^{\mathrm{full}}_{\mathrm{eff}} - \hat{d}^{\mathrm{cut}}_{\mathrm{eff}}$, for biased (blue points) and unbiased (yellow points) models. Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ averaged over 30 training instances starting from randomly chosen parameters, for a single random model realization, i.e. a random right-normalized tensor train representing V. The red line serves as a guide for the eye for zero MSE difference. The spread over training instances is not shown for clarity, but reported in panels (b) and (c), as representatives of all points in (a). (b) Training curves for a random biased model realization, with full model in blue and cutoff model in orange. (c) Training curves for a random unbiased model realization, with full model in blue and cutoff model in orange. The shading corresponds to the spread over 30 training instances. For these plots, $N = 1$, $\Omega = \{1, \ldots, 17\}$ ($d = 35$), $\tilde{\Omega} = \{1\}$ ($\tilde{d} = 3$), $M = 32$, $R = 7$, $\chi = 60$, $\mathfrak{n}_{\mathrm{train}} = 30$ with a batch size of 5.

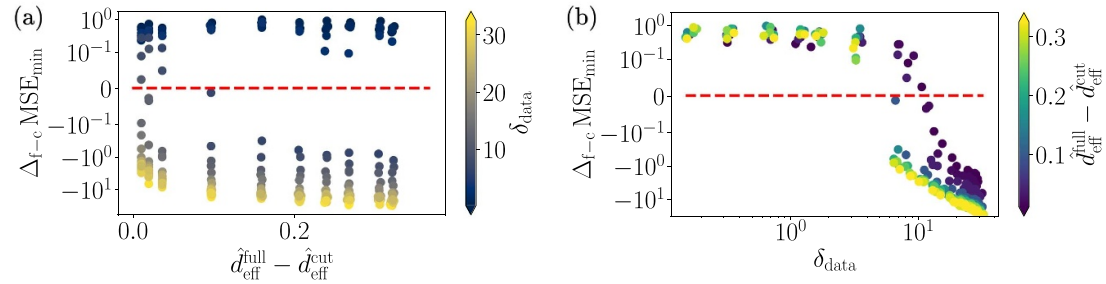
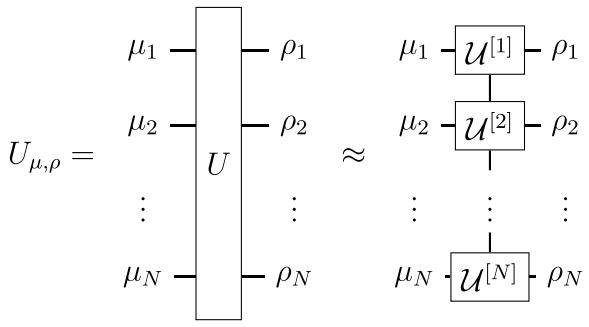
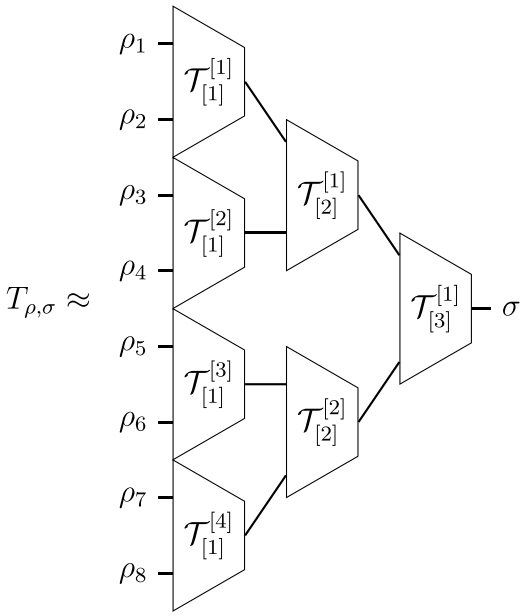


Figure D2. (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ for different values of $\hat{d}^{\mathrm{full}}_{\mathrm{eff}} - \hat{d}^{\mathrm{cut}}_{\mathrm{eff}}$, for different values of δ_{data} (color scale). Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ averaged over 30 training instances for 30 random model realization. (b) Same as panel (a) but resolved as a function of δ_{data}. The red line serves as a guide for the eye for zero MSE difference. For these plots, $N = 1$, $\Omega = \{1, \ldots, 17\}$ ($d = 35$), $\tilde{\Omega} = \{1\}$ ($\tilde{d} = 3$), $M = 32$, $R = 7$, $\chi = 60$, $\mathfrak{n}_{\mathrm{train}} = 30$ with a batch size of 5.

with $\mathcal{U}^{[n]}$ being rank-4 tensors constrained to yield an orthogonal U. The TN decomposition of U has the following diagrammatic representation



Finally, the isometry T can be decomposed as a TTN with the following diagrammatic representation



where the tensors $\mathcal{T}^{[\tau]}_{[\ell]}$ are isometric rank-3 tensors. Further details on the explicit construction of U, V and T can be found in the Supplementary Material.

Further numerical results on the effects of ED and bias on training are shown in figures D1 and D2, for models where only the matrix V is decomposed as a tensor train. The numerical experiments presented here are conducted in the same way as discussed in section 3 in the main text, and confirm the conclusions drawn there: a lower ED is beneficial during training ($\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ is positive) for models that are biased towards the data-generating function to be learned, whereas in the unbiased case a higher ED leads to better training performance.

ORCID iDs

Lorenzo Pastori 0000-0001-5882-8482
Mierk Schwabe 0000-0001-6565-5890

References

[1] Perdomo-Ortiz A, Benedetti M, Realpe-Gómez J and Biswas R 2018 *Quantum Sci. Technol.* **3** 030502
[2] Benedetti M, Lloyd E, Sack S and Fiorentini M 2019 *Quantum Sci. Technol.* **4** 043001
[3] Cerezo M, Verdon G, Huang H Y, Cincio L and Coles P J 2022 *Nat. Comput. Sci.* **2** 567–76
[4] Farhi E and Neven H 2018 Classification with quantum neural networks on near term processors (arXiv:1802.06002)
[5] Cerezo M *et al* 2021 *Nat. Rev. Phys.* **3** 625–44
[6] Bharti K *et al* 2022 *Rev. Mod. Phys.* **94** 015004
[7] Preskill J 2018 *Quantum* **2** 79
[8] Du Y, Hsieh M H, Liu T and Tao D 2020 *Phys. Rev. Res.* **2** 033125
[9] Yu Z, Chen Q, Jiao Y, Li Y, Lu X, Wang X and Yang J Z 2023 Provable advantage of parameterized quantum circuit in function approximation (arXiv:2310.07528)
[10] Caro M C, Gil-Fuster E, Meyer J J, Eisert J and Sweke R 2021 *Quantum* **5** 582
[11] Banchi L, Pereira J and Pirandola S 2021 *PRX Quantum* **2** 040321
[12] Caro M C, Huang H Y, Cerezo M, Sharma K, Sornborger A, Cincio L and Coles P J 2022 *Nat. Commun.* **13** 4919
[13] Haug T and Kim M S 2024 *Phys. Rev. Lett.* **133** 050603
[14] Chen S Y C, Wei T C, Zhang C, Yu H and Yoo S 2022 *Phys. Rev. Res.* **4** 013231
[15] Hur T, Kim L and Park D K 2022 *Quantum Mach. Intell.* **4** 3
[16] Sünkel L, Martyniuk D, Reichwald J J, Morariu A, Seggoju R H, Altmann P, Roch C and Paschke A 2023 Hybrid quantum machine learning assisted classification of covid-19 from computed tomography scans *2023 IEEE Int. Conf. on Quantum Computing and Engineering (QCE)* (IEEE) pp 356–66
[17] Shen K, Jobst B, Shishenina E and Pollmann F 2024 Classification of the fashion-mnist dataset on a quantum computer (arXiv:2403.02405)
[18] Belis V, Woźniak K A, Puljak E, Barkoutsos P, Dissertori G, Grossi M, Pierini M, Reiter F, Tavernelli I and Vallecorsa S 2024 *Commun. Phys.* **7** 334
[19] Corli S, Moro L, Dragoni D, Dispenza M and Prati E 2024 Quantum machine learning algorithms for anomaly detection: a survey (arXiv:2408.11047)

[20] Duneau T, Bruhn S, Matos G, Laakkonen T, Saiti K, Pearson A, Meichanetzidis K and Coecke B 2024 Scalable and interpretable quantum natural language processing: an implementation on trapped ions (arXiv:2409.08777)
[21] Aizpurua B, Jahromi S S, Singh S and Orus R 2024 Quantum large language models via tensor network disentanglers (arXiv:2410.17397)
[22] Sakhnenko A, Sikora J and Lorenz J 2024 Buildung continuous quantum-classical bayesian neural networks for a classical clinical dataset *Proc. Recent Advances in Quantum Computing and Technology* (*ReAQCT '24*) (ACM) pp 62–72
[23] Slabbert D, Lourens M and Petruccione F 2024 *Quantum Mach. Intell.* **6** 56
[24] Pastori L, Grundner A, Eyring V and Schwabe M 2026 *Mach. Learn. Earth* **2** 015008
[25] Sim S, Johnson P D and Aspuru-Guzik A 2019 *Adv. Quantum Technol.* **2** 1900070
[26] Hubregtsen T, Pichlmeier J, Stecher P and Bertels K 2021 *Quantum Mach. Intell.* **3** 9
[27] Abbas A, Sutter D, Zoufal C, Lucchi A, Figalli A and Woerner S 2021 *Nat. Comp. Sci.* **1** 403–9
[28] Kiss O, Tacchino F, Vallecorsa S and Tavernelli I 2022 *Mach. Learn.: Sci. Technol.* **3** 035004
[29] Wilkinson S A and Hartmann M J 2022 Evaluating the performance of sigmoid quantum perceptrons in quantum neural networks (arXiv:2208.06198)
[30] Otgonbaatar S, Schwarz G, Datcu M and Kranzlmüller D 2023 *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **16** 9223–30
[31] Drăgan T A, Monnet M, Mendl C B and Lorenz J M 2022 Quantum reinforcement learning for solving a stochastic frozen lake environment and the impact of quantum architecture choices (arXiv:2212.07932)
[32] Monnet M, Chaabani N, Drăgan T A, Schachtner B and Lorenz J M 2024 Understanding the effects of data encoding on quantum-classical convolutional neural networks *2024 IEEE Int. Conf. on Quantum Computing and Engineering (QCE)* vol 01 pp 1436–46
[33] Zhu Z, Zhu S and Bu S 2026 *IEEE Trans. Power Syst.* **41** 1109–20
[34] Chen S Y C 2025 Evolutionary optimization for designing variational quantum circuits with high model capacity *2025 IEEE Symp. for Multidisciplinary Computational Intelligence Incubators (MCII Companion)* pp 1–5
[35] Röseler P, Willsch D and Michielsen K 2026 How to find expressible and trainable parameterized quantum circuits? (arXiv:2603.14451)
[36] McKiernan S and Sapienza L 2026 Algorithmic advantage on a gate-based photonic quantum neural network (arXiv:2605.10801)
[37] Canatar A, Bordelon B and Pehlevan C 2021 *Nat. Commun.* **12** 2914
[38] Amini A, Baumgartner R, Feng D Koyejo S, Mohamed S, Agarwal A, Belgrave D, Cho K and Oh A 2022 Target alignment in truncated kernel ridge regression *Advances in Neural Information Processing Systems* vol 35, ed (Curran Associates, Inc.) pp 21948–60 (available at: https://proceedings.neurips.cc/paper_files/paper/2022/file/89b89c04f55ea7c7ca989992bb6a98c0-paper-Conference.pdf)
[39] Wang C, He X, Wang Y and Wang J 2024 On the target-kernel alignment: a unified analysis with kernel complexity *Advances in Neural Information Processing Systems*, ed A Globerson, L Mackey, D Belgrave, A Fan, U Paquet, J Tomczak and C Zhang (Curran Associates, Inc.) vol 37 pp 40434–85 (available at: https://proceedings.neurips.cc/paper_files/paper/2024/file/477cde5ca7ab9254d8fc83b97e834446-paper-Conference.pdf)
[40] Shan H and Bordelon B 2022 A theory of neural tangent kernel alignment and its influence on training (arXiv:2105.14301)
[41] Elesedy B and Zaidi S 2021 Provably strict generalisation benefit for equivariant models (arXiv:2102.10333)
[42] Gil V F J and Theis D O 2020 *Frontiers Phys.* **8** 297
[43] Schuld M, Sweke R and Meyer J J 2021 *Phys. Rev. A* **103** 032430
[44] Wierichs D, Izaac J, Wang C and Lin C Y Y 2022 *Quantum* **6** 677
[45] Casas B and Cervera-Lierta A 2023 *Phys. Rev. A* **107** 062612
[46] Pérez-Salinas A, Cervera-Lierta A, Gil-Fuster E and Latorre J I 2020 *Quantum* **4** 226
[47] Amari S I 1985 *Differential-Geometrical Methods in Statistics* (*Lecture Notes in Statistics*) (Springer)
[48] Amari S I 1997 *Neural Comput.* **10** 251–76
[49] Pascanu R and Bengio Y 2014 Revisiting natural gradient for deep networks (arXiv:1301.3584)
[50] Pennington J and Worah P 2018 The spectrum of the fisher information matrix of a single-hidden-layer neural network *Advances in Neural Information Processing Systems*, ed S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi and R Garnett (Curran Associates, Inc.) vol 31 (available at: https://proceedings.neurips.cc/paper_files/paper/2018/file/18bb68e2b38e4a8ce7cf4f6b2625768c-paper.pdf)
[51] Amari S i, Karakida R and Oizumi M 2019 Fisher information and natural gradient learning in random deep networks *Proc. Twenty-Second Int. Conf. on Artificial Intelligence and Statistics* (*Proceedings of Machine Learning Research*), ed K Chaudhuri and M Sugiyama (PMLR) vol 8 pp 694–702 (available at: https://proceedings.mlr.press/v89/amari19a.html)
[52] Karakida R, Akaho S and ichi Amari S 2020 *J. Stat. Mech.* **2020** 124005
[53] Hayase T and Karakida R 2021 The spectrum of fisher information of deep networks achieving dynamical isometry *Proc. 24th Int. Conf. on Artificial Intelligence and Statistics* (*Proceedings of Machine Learning Research*), ed A Banerjee and K Fukumizu (PMLR) vol 130 pp 334–42 (available at: https://proceedings.mlr.press/v130/hayase21a.html)
[54] Karakida R, Akaho S and Amari S i 2021 *Neural Comput.* **33** 2274–307
[55] Abbas A, Sutter D, Figalli A and Woerner S 2021 Effective dimension of machine learning models (arXiv:2112.04807)
[56] Larocca M, Ju N, García-Martín D, Coles P J and Cerezo M 2023 *Nat. Comput. Sci.* **3** 542–51
[57] Jacot A, Gabriel F and Hongler C 2018 Neural tangent kernel: convergence and generalization in neural networks *Advances in Neural Information Processing Systems*, ed S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi and R Garnett (Curran Associates, Inc.) vol 31 (available at: https://proceedings.neurips.cc/paper_files/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-paper.pdf)
[58] Fort S, Dziugaite G K, Paul M, Kharaghani S, Roy D M and Ganguli S 2020 Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel *Proc. 34th Int. Conf. on Neural Information Processing Systems* (*NIPS'20*) (Curran Associates Inc.)
[59] Loo N, Hasani R, Amini A and Rus D 2022 Evolution of neural tangent kernels under benign and adversarial training *Proc. 36th Int. Conf. on Neural Information Processing Systems* (*NIPS'22*) (Curran Associates Inc.)
[60] Verstraete F, Murg V and Cirac J I 2008 *Adv. Phys.* **57** 143–224
[61] Orús R 2014 *Ann. Phys., NY* **349** 117–58
[62] Ran S J, Tirrito E, Peng C, Chen X, Tagliacozzo L, Su G and Lewenstein M 2020 *Tensor Network Contractions* (Springer)
[63] Oseledets I V 2011 *SIAM J. Sci. Comput.* **33** 2295–317
[64] Schollwöck U 2011 *Ann. Phys., NY* **326** 96–192
[65] Pirvu B, Murg V, Cirac J I and Verstraete F 2010 *New J. Phys.* **12** 025012

[66] Hubig C, McCulloch I P and Schollwöck U 2017 *Phys. Rev. B* **95** 035129
[67] Styliaris G, Trivedi R, Perez-Garcia D and Cirac J I 2025 *Quantum* **9** 1645
[68] Shi Y Y, Duan L M and Vidal G 2006 *Phys. Rev. A* **74** 022320
[69] Tagliacozzo L, Evenbly G and Vidal G 2009 *Phys. Rev. B* **80** 235127
[70] Murg V, Verstraete F, Legeza O and Noack R M 2010 *Phys. Rev. B* **82** 205105
[71] Kingma D P and Ba J 2017 Adam: a method for stochastic optimization (arXiv:1412.6980)
[72] Schuld M 2021 Supervised quantum machine learning models are kernel methods (arXiv:2101.11020)
[73] Kübler J, Buchholz S and Schölkopf B 2021 The inductive bias of quantum kernels *Advances in Neural Information Processing Systems*, ed M Ranzato, A Beygelzimer, Y Dauphin, P Liang and J W Vaughan (Curran Associates, Inc.) vol 34 pp 12661–73 (available at: https://proceedings.neurips.cc/paper_files/paper/2021/file/69adc1e107f7f7d035d7baf04342e1ca-paper.pdf)
[74] Peters E and Schuld M 2023 *Quantum* **7** 1210
[75] Dar Y, Muthukumar V and Baraniuk R G 2021 A farewell to the bias-variance tradeoff? an overview of the theory of overparameterized machine learning (arXiv:2109.02355)
[76] McClean J R, Boixo S, Smelyanskiy V N, Babbush R and Neven H 2018 *Nat. Commun.* **9** 4812
[77] Thanasilp S, Wang S, Nghiem N A, Coles P and Cerezo M 2023 *Quantum Mach. Intell.* **5** 21
[78] Larocca M, Thanasilp S, Wang S, Sharma K, Biamonte J, Coles P J, Cincio L, McClean J R, Holmes Z and Cerezo M 2025 *Nat. Rev. Phys.* **7** 174–89
[79] Landman J, Thabet S, Dalyac C, Mhiri H and Kashefi E 2022 Classically approximating variational quantum machine learning with random Fourier features (arXiv:2210.13200)
[80] Schreiber F J, Eisert J and Meyer J J 2023 *Phys. Rev. Lett.* **131** 100803
[81] Sweke R, Recio E, Jerbi S, Gil-Fuster E, Fuller B, Eisert J and Meyer J J 2023 Potential and limitations of random Fourier features for dequantizing quantum machine learning (arXiv:2309.11647)
[82] Shin S, Teo Y S and Jeong H 2024 *Phys. Rev. Res.* **6** 023218
[83] Kreplin D A and Roth M 2024 *Quantum* **8** 1385
[84] Crow D and Joynt R 2014 *Phys. Rev. A* **89** 042123
[85] Wang S, Fontana E, Cerezo M, Sharma K, Sone A, Cincio L and Coles P J 2021 *Nat. Commun.* **12** 6961
[86] Brouwer P W and Beenakker C W J 1996 *J. Math. Phys.* **37** 4904–34
[87] Collins B and Śniady P 2006 *Commun. Math. Phys.* **264** 773–95
[88] Collins B and Matsumoto S 2009 *J. Math. Phys.* **50** 113516
[89] Lorenzo P, Veronika E and Mierk S 2025 *pastori26MLST_fim-training-fourier-models* (available at: https://doi.org/10.5281/zenodo.21511180 ; https://github.com/EyringMLClimateGroup/pastori26MLST_fim-training-fourier-models)