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*

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* 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 effective dimension 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 effective dimension 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 effective dimension. This allows us to construct models with tunable effective dimension 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, 2, 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, 2, 3, 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, 11, 12, 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, 15, 16, 17, 18, 19, 20, 21, 22, 23, 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 [27] can be calculated for both classical and quantum models. This latter quantity is the main focus of our work.

The effective dimension (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. This is schematically illustrated in Fig. 1(a) and (b). More general, our findings hint 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 Summary of results

The findings presented in this paper are obtained by exploiting the equivalence between a broad class of QNNs and Fourier models [28, 29, 30, 31], which we extend to include the dependence on the trainable QNN parameters. Based on this equivalence, we derive an analytical expression for the Fisher information matrix (FIM), which allows us to identify the relevant features of a Fourier model that control the FIM spectrum, and thereby the effective dimension (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 Fig. 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 Fig. 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 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.

1.2 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 quantum neural networks (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,
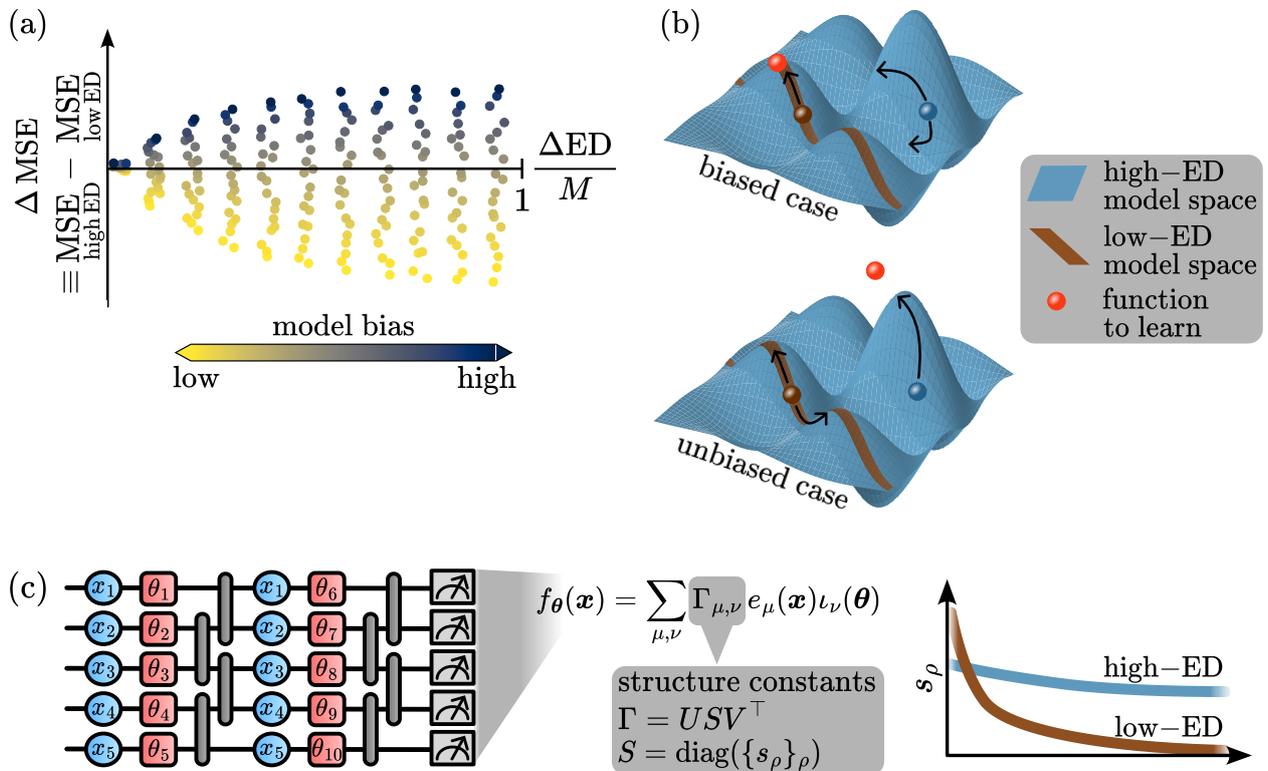
Figure 1: Illustration of main results. (a) Schematic behavior of the difference in the training loss (here the mean squared error — MSE) ΔMSE, between models with high and low effective dimension (ED), vs. the corresponding ED difference ΔED (between models with high and low ED — normalized by the number 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.

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.,

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

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 quantum neural networks (QNNs) which, as we discuss later, are known to be Fourier models [28, 29, 30, 31].

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 parameterized quantum circuit (PQC) as

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

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 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 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 $\boldsymbol{\omega} = (\omega_1, ..., \omega_N)$ with $\omega_n \in \Omega_n$ and Ω_n the set of Fourier frequencies for the n-th input component, $\tilde{\boldsymbol{\omega}} = (\tilde{\omega}_1, ..., \tilde{\omega}_M)$ with $\tilde{\omega}_m \in \tilde{\Omega}_m$ and $\tilde{\Omega}_m$ the set of Fourier frequencies for the m-th 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 Eqs. (1) and (3) with $e_{\mu}(\boldsymbol{x}) \equiv e_{(\mu_1,...,\mu_N)}(\boldsymbol{x}) = \prod_{n=1}^{N} e_{\mu_n}^{(n)}(x_n)$ and $\iota_{\nu}(\boldsymbol{\theta}) \equiv \iota_{(\nu_1,...,\nu_M)}(\boldsymbol{\theta}) = \prod_{m=1}^{M} \iota_{\nu_m}^{(m)}(\theta_m)$, where

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

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

normalized in the interval $[-\pi, \pi]$, and with the structure constants $\Gamma_{\mu,\nu}$ given by suitable real linear combinations of the complex coefficients $\tilde{\Gamma}_{\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}}$. 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 [29, 32], 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 A, the sets Ω_n and $\tilde{\Omega}_m$ comprise only integer frequencies and read as $\Omega_n = \{1, ..., L\}$ 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 $U^\top U = UU^\top = I_D$, $S = \mathrm{diag}(s_1, ..., s_D)$ is a $D \times D$ diagonal positive semi-definite matrix (with diagonal ordered as $s_1 \geq s_2 \geq ... \geq s_D$), 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

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

where $e_\rho^U(\boldsymbol{x}) \equiv \sum_{\mu=1}^{D} U_{\mu,\rho} \, e_\mu(\boldsymbol{x})$ and $\iota_\rho^V(\boldsymbol{\theta}) \equiv \sum_{\nu=1}^{K} [V^\top]_{\rho,\nu} \, \iota_\nu(\boldsymbol{\theta})$ 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, $e_1^U(\boldsymbol{x})$: 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 $e_\rho^U(\boldsymbol{x})$ 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 Fisher information matrix and effective dimension.

2.2 Fisher information matrix and effective dimension

In this section we discuss the notion of Fisher information matrix (FIM) and effective dimension (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 effective dimension.

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

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

2.2.1 Definition of FIM and ED.

The Fisher information matrix (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 [33, 34, 35]. For the regression models studied in this work, the FIM F is a $M \times M$ positive semi-definite matrix with elements (see [36, 37, 38, 39, 40] 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 effective dimension (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

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

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. In the next sections, we uncover the relation between the correlation spectrum introduced before and the FIM.

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 effective dimension 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

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

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 Eq. (14) we can already recognize the dependence on the squared singular values s_ρ^2. These, in combination with Eq. (13), tell us that the normalized FIM $\hat{F}$, hence the ED, is independent of the value of $\text{tr}(S^2)$. 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 $\text{tr}(S^2)$. 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 [41]), the rank of the FIM is upper-bounded as follows:

$$\text{rank}(F(\boldsymbol{\theta})) \leq D \, . \tag{16}$$

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:

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

$$\text{Var}\big[F_{j,k}\big] \in \mathcal{O}(1) \, \text{tr}(S^4) \, .$$

The derivation of these properties is sketched in 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,
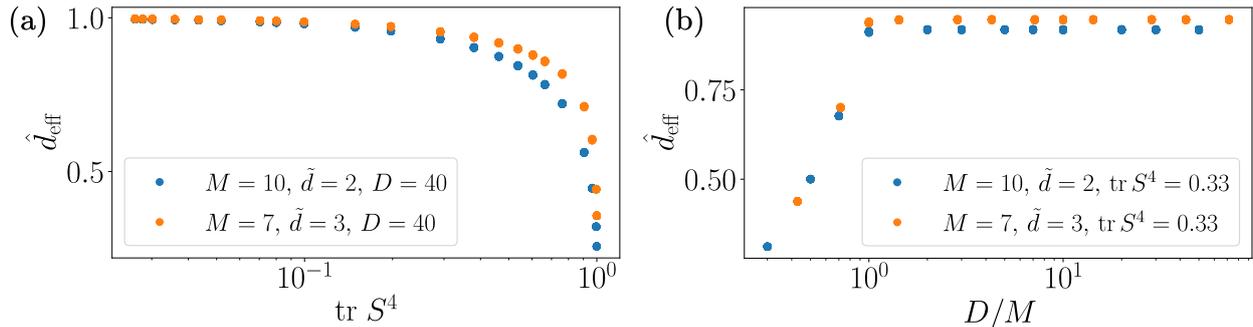
Figure 2: (a) Scaling of normalized ED with the purity $\mathrm{tr}(S^4)$ of the correlation spectrum. (b) Scaling of normalized ED with the ratio D/M. Each point corresponds to a random model realization, i.e., a random Γ uniformly drawn from $[-1, +1]^{D \times K}$. For every value of $\mathrm{tr}(S^4)$ and D/M, 50 model realizations are drawn (the points are on top of each others). The normalized ED is computed using Eq. (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 Eq.(8), for all $m = 1, ..., M$.

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 $\mathrm{tr}(S^4)$ is small. Conversely, when the correlation spectrum S is not flat and $\mathrm{tr}(S^4)$ is large (i.e., approaching one), $\mathrm{Var}[F_{j,k}]$ 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 $\mathrm{tr}(S^4)$, 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 Fig. 2. Specifically, we calculate the normalized ED $\hat{d}_{\mathrm{eff}}$ 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 $\hat{d}_{\mathrm{eff}}$ changes with $\mathrm{tr}(S^4)$ and the input functions' space dimension D. In panel (a) we show the dependence of $\hat{d}_{\mathrm{eff}}$ on the purity of the correlation spectrum $\mathrm{tr}(S^4)$. As expected, $\hat{d}_{\mathrm{eff}}$ decreases with $\mathrm{tr}(S^4)$ increasing towards its maximum value 1. In order to change $\mathrm{tr}(S^4)$, an exponential decay with variable decay rate is imposed to the singular values, keeping them normalized to $\mathrm{tr}(S^2) = 1$ (which has no effect on the ED). In panel (b) we show the dependence of $\hat{d}_{\mathrm{eff}}$ on the ratio D/M. For $D < M$, $\hat{d}_{\mathrm{eff}}$ increases with increasing D until it saturates to its maximal value for $D \geq M$. For $D \geq M$, $\hat{d}_{\mathrm{eff}}$ is independent of D and largely controlled by $\mathrm{tr}(S^4)$. The increase of $\hat{d}_{\mathrm{eff}}$ 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.

2.3 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 effective dimension 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*.

For an explicit construction, we consider a data-generating function $y(\boldsymbol{x})$ of the form

$$y(\boldsymbol{x}) = \sum_{\mu=1}^{D} \sum_{\nu=1}^{K} e_\mu(\boldsymbol{x}) \, \iota_\nu(\boldsymbol{\theta}^*) \sum_{\rho=1}^{R} s_\rho \, U_{\mu,\rho}^{(\mathrm{d})} \left[V^{(\mathrm{d}) \top} \right]_{\rho, \nu} , \tag{18}$$
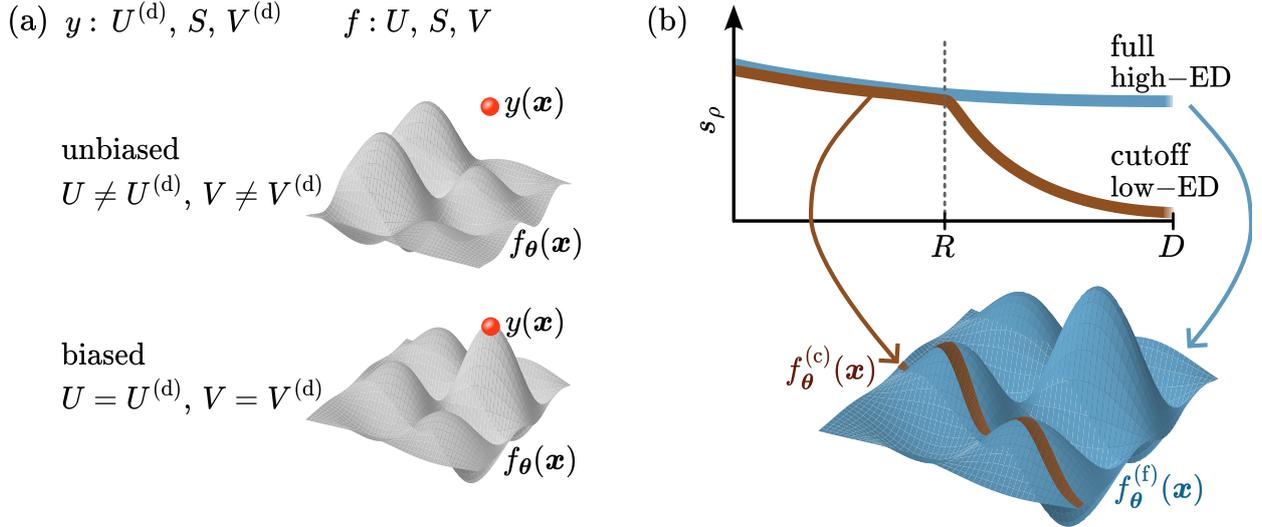
Figure 3: (a) Schematic illustration of the construction of biased and unbiased models. The data-generating function y is specified by matrices $U^{(d)}$ and $V^{(d)}$, and the model f by U and V, with both y and f having the same correlation spectrum S. In the unbiased case, y (represented by the red 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 $\boldsymbol{\theta}^*$ a given parameter configuration, $R < D$ and with $V^{(d)}$ constructed in order to satisfy the property $\sum_\nu V_{\nu,\rho}^{(d)} \iota_\nu(\boldsymbol{\theta}^*) = 0$ for $\rho = R+1, ..., K$.

Any model of the form of Eq. (10) with structure constants chosen independently of $U^{(d)}$ and $V^{(d)}$, 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 $f_{\boldsymbol{\theta}}^{(f)}(\boldsymbol{x})$ specified by the structure constants

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

satisfies $f_{\boldsymbol{\theta}^*}^{(f)}(\boldsymbol{x}) = y(\boldsymbol{x})$ by construction, and is therefore called *biased*. This is schematically illustrated in Fig. 3(a). Importantly, one can define several such biased models with different properties of s_ρ, hence different effective dimensions, 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

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

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 Fig. 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.

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 Eq. (18) with $V_\epsilon^{(d)}$ obtained by adding a suitably chosen random perturbation of strength ϵ to its elements, as we discuss in 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

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

9

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

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 tensor network (TN) [42, 43, 44] 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 $e_{\mu_n}^{(n)}(x_n)$ and $\iota_{\nu_m}^{(m)}(\theta_m)$. To construct a TN decomposition of Γ, we consider the following approximation

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

which differs from Eq. (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) [45, 46], the orthogonal matrix U containing the left-singular values of Γ as an orthogonal matrix product operator (MPO) [47, 48, 49], and the isometry T can be as a tree tensor network (TTN) [50, 51, 52]. We note that the TN decompositions used here are not the only option, and different ones are possible. In 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 effective dimension. 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 $\{\boldsymbol{x}_i, y(\boldsymbol{x}_i)\}_{i=1,\ldots,\mathfrak{n}_{\text{train}}}$ that we use for training the model, using the mean squared error (MSE) as loss function

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

The numerical results presented in this section are obtained using Fourier regression models, i.e., with input and parameter basis functions $e_{\mu_n}^{(n)}(x_n)$ and $\iota_{\nu_m}^{(m)}(\theta_m)$ given by Eqs. (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 [53].

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 (Eq. (19)) and cutoff models (Eq. (20)), in order to compare the training dynamics of models with higher and lower ED, respectively. To visualize this comparison, for any given degree of bias we consider the minimum MSE
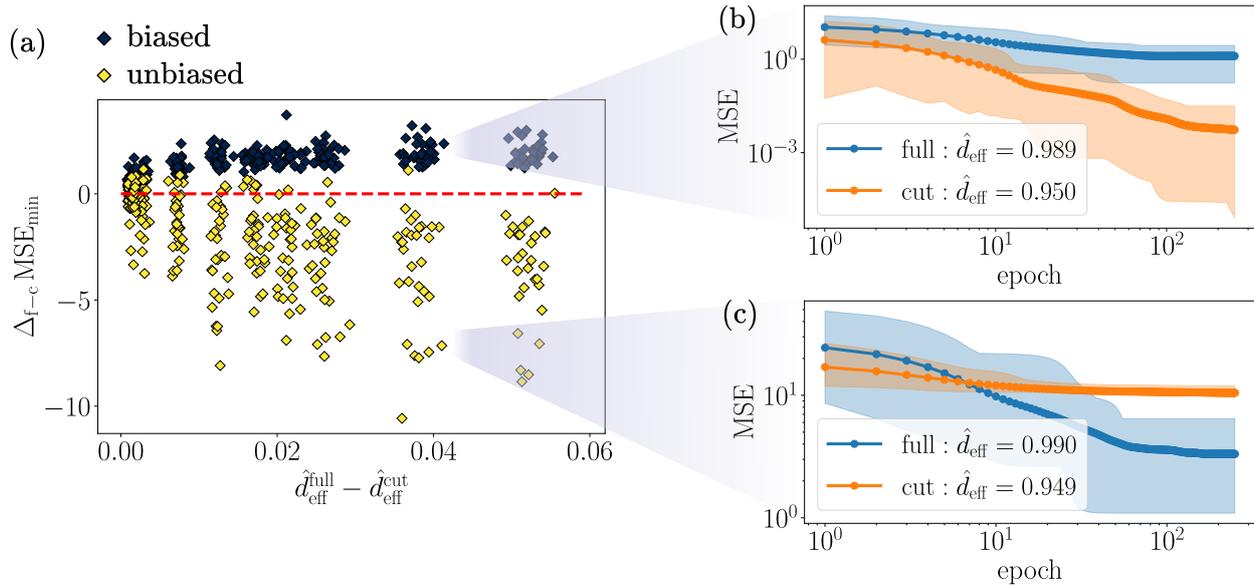
Figure 4: (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ for different values of $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$, for biased (blue points) and unbiased (yellow points) models. Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ 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. (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, $N = 1$, $\Omega = \{1, ..., 8\}$ ($d = 17$), $\tilde{\Omega} = \{1\}$ ($\tilde{d} = 3$), $M = 7$, $R = 6$, $\mathfrak{n}_{\mathrm{train}} = 25$ with a batch size of 5.

attained during training as a proxy for the training quality, denoted with $\mathrm{MSE}_{\mathrm{min}}^{\mathrm{full}}$ and $\mathrm{MSE}_{\mathrm{min}}^{\mathrm{cut}}$ for full and cutoff models, respectively, and study the difference

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

as a function of the difference in the ED between full and cutoff models, i.e., $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$. 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.

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 Fig. 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 $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$. 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 $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$. That is, in the biased case, models with lower ED train to a lower MSE (as can be seen in Fig. 4(b) for a specific random model realization), whereas in the unbiased case a higher ED is beneficial for training (as shown in Fig. 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 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 Eq. (21)) influences the training, for different values of $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$. As shown in Fig. 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 Fig. 5(b), there is an extended
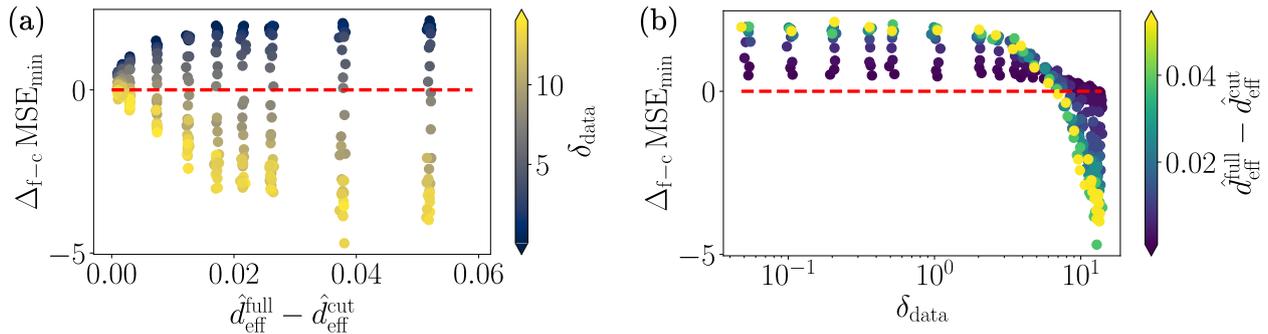
Figure 5: (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ for different values of $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$, for different values of δ_{data} (color scale). Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ averaged over 30 training 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, $N = 1$, $\Omega = \{1, ..., 8\}$ ($d = 17$), $\tilde{\Omega} = \{1\}$ ($\tilde{d} = 3$), $M = 7$, $R = 6$, $\mathfrak{n}_{\mathrm{train}} = 25$ with a batch size of 5.

regime in terms of values of δ_{data} where $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ 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 Fig. 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 Fig. 6(a) are consistent with those of Fig. 4(a), showing that in the biased case $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ is positive, i.e., models with lower ED train to a lower MSE, whereas in the unbiased case $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ is negative, i.e., models with higher ED have better training performance. We refer the reader to D for a more detailed study of the dependence of $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ on M and D, showcasing the effects of the models' under- and overparameterization [41] on our results. There and in the Supplementary Material we also show further numerical results on training tensorized models, which confirm the findings presented here.

4 Conclusion and outlook

In this work we investigated how the effective dimension (ED), calculated from the Fisher information matrix (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. 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
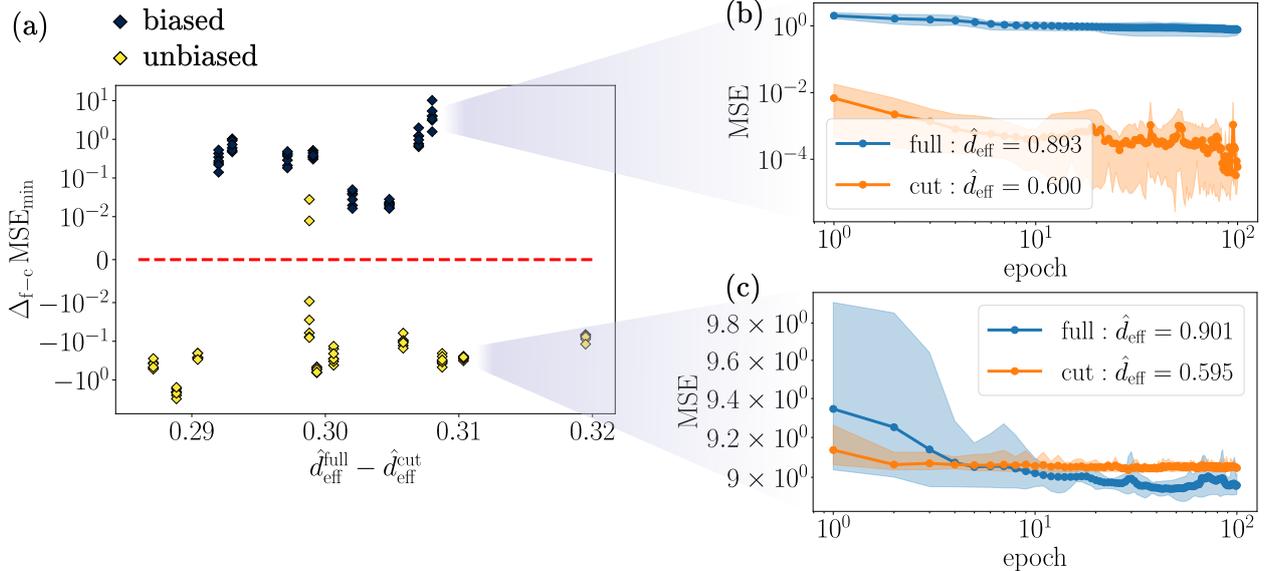
Figure 6: (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ for different values of $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$, for biased (blue points) and unbiased (yellow points) models. Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ averaged 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. (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, $N = 4$, $\Omega = \{1, ..., 3\}$ $(d = 7)$, $\tilde{\Omega} = \{1\}$ $(\tilde{d} = 3)$, $M = 24$, $R = 2$, $\chi = 30$, $\mathfrak{n}_{\mathrm{train}} = 6^4$ with a batch size of 12.

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. While we expect our results not to depend on this choice, we would find it interesting to investigate the same mechanism when comparing inherently different models, i.e., models accessing different types of function spaces. For example, designing analogous 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.

Given the focus of this work on regression models, a natural extension would be to perform a similar analysis for classification (see the Supplementary Material for suggestions on a potential generalization) or generative models. Furthermore, it would be interesting to investigate the questions addressed here also in the context of natural gradient descent [34, 37], which could help to develop a deeper understanding of the ED-bias interplay in training ML models.

Finally, there are also several opportunities for establishing a deeper connection between our findings and the theory of quantum machine learning. One interesting direction could be investigating the connections with (quantum) kernel methods [54], for which results on the effects of inductive bias (kernel-task alignment) on the training and generalization performance have already been established [55, 56]. Other important aspects to be addressed in the future concern the connection to existing works on generalization [10, 11, 57], overfitting [57, 58], and overparameterization [41] in quantum machine learning. 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 quantum machine learning models [59, 60, 61]. 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 [62] and enabling further understanding on the function classes encompassed by quantum machine learning models.

Acknowledgments

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 Eq. (5), focusing for simplicity on the case of a one-dimensional input. This can be easily generalized to more input features [29, 31] following the same derivation. We consider L layers of data re-uploading [32, 29] with unitary $\hat{U}_{\boldsymbol{\theta}}(x)$ expressed as

$$\hat{U}_{\boldsymbol{\theta}}(x) = \prod_{\ell=1}^{L} \left[\hat{S}(x)\, \hat{W}^{(\ell)}(\boldsymbol{\theta}^{(\ell)}) \right] , \tag{25}$$

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., $\boldsymbol{\theta} = \{\boldsymbol{\theta}^{(\ell)}\}_{\ell=1,...,L}$. 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 [29, 31] we switch to the diagonal representations of $\hat{S}(x)$ and $\hat{G}_{j_\ell}^{(\ell)}(\theta_{j_\ell}^{(\ell)})$. These have eigenvalues $\{e^{-i\lambda_\alpha x}\}_{\alpha=1,...,\mathcal{D}}$ and $\{e^{-i\eta_\beta^{(\ell,j)}\theta_{j_\ell}^{(\ell)}}\}_{\beta=1,...,\mathcal{D}}$, 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 $e^{-i\Lambda_\alpha x}$ and $e^{-i\boldsymbol{\eta}_\beta \cdot \boldsymbol{\theta}}$, with the shorthand notation $\boldsymbol{\alpha} = (\alpha_1,...,\alpha_L)$, $\boldsymbol{\beta} = (\beta_1,...,\beta_M)$, $\Lambda_{\boldsymbol{\alpha}} = \sum_\ell \lambda_{\alpha_\ell}$ and $\boldsymbol{\eta}_{\boldsymbol{\beta}} = (\eta_{\beta_1}^{(1)},...,\eta_{\beta_M}^{(M)})$. We refer to the Supplementary Material for the details of the calculation, and we report here the result

$$f_{\boldsymbol{\theta}}(x) = \sum_{\boldsymbol{\alpha},\boldsymbol{\alpha}'} \sum_{\boldsymbol{\beta},\boldsymbol{\beta}'} \tilde{\Gamma}_{(\boldsymbol{\alpha};\boldsymbol{\alpha}'),(\boldsymbol{\beta};\boldsymbol{\beta}')} \, e^{i(\Lambda_{\boldsymbol{\alpha}} - \Lambda_{\boldsymbol{\alpha}'})x} \, e^{i(\boldsymbol{\eta}_{\boldsymbol{\beta}} - \boldsymbol{\eta}_{\boldsymbol{\beta}'})\cdot\boldsymbol{\theta}} , \tag{26}$$

which has the same form as Eq. (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 $\Lambda_{\boldsymbol{\alpha}} - \Lambda_{\boldsymbol{\alpha}'} \in \left\{ -L,-L+1,...,L-1,L \right\}$, thus yielding the local basis set $\mathcal{B} = \{1, \sqrt{2}\cos(x),...,\sqrt{2}\cos(Lx), \sqrt{2}\sin(x),...,\sqrt{2}\sin(Lx)\}$. 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 $\tilde{\mathcal{B}}_m = \{1, \sqrt{2}\cos\theta_m, \sqrt{2}\sin\theta_m\}$.

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 Eq. (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 $|\boldsymbol{\iota}(\boldsymbol{\theta})\big)$ the K-dimensional vector with components $\iota_\nu(\boldsymbol{\theta})$ and

$$\mathcal{F}^{(j,k)} = B_j^\top V S^2 V^\top B_k = \sum_{\rho=1}^{D} s_\rho^2 \, B_j^\top P_\rho B_k , \tag{27}$$

where $B_j = I_{\tilde{d}}^{(1)} \otimes I_{\tilde{d}}^{(2)} \otimes ... \otimes \beta^{(j)} \otimes ... \otimes I_{\tilde{d}}^{(M)}$ (with $I_{\tilde{d}}^{(k)}$ being the $\tilde{d}$-dimensional identity matrix acting on the function space 'local' to the k-th parameter), and P_ρ being the projection on the subspace spanned by the vector $V_{\cdot,\rho}$. For showing Eq. (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 Eq. (17), we use results from random matrix theory

[63, 64, 65] 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

$$\mathbb{E}_{V \in \mathrm{O}(K)}\big[F_{j,k}(\boldsymbol{\theta})\big] = \frac{\mathrm{tr}(S^2)}{K}\left(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_j^\top B_k\big|\boldsymbol{\iota}(\boldsymbol{\theta})\right) , \tag{28}$$

and

$$\begin{aligned}
\mathrm{Var}_{V \in \mathrm{O}(K)}\big[F_{j,k}(\boldsymbol{\theta})\big] = \frac{\mathrm{tr}(S^4)}{K^2}\Big[&\left(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_j^\top B_k\big|\boldsymbol{\iota}(\boldsymbol{\theta})\right)^2 + \\
&\left(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_j^\top B_j\big|\boldsymbol{\iota}(\boldsymbol{\theta})\right)\left(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_k^\top B_k\big|\boldsymbol{\iota}(\boldsymbol{\theta})\right)\Big] + \\
&\mathcal{O}(K^{-3}) .
\end{aligned} \tag{29}$$

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

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 Eq. (18). In order to construct a $K \times D$ matrix $V^{(\mathrm{d})}$ with orthonormal columns and satisfying the property $\sum_\nu V_{\nu,\rho}^{(\mathrm{d})} \iota_\nu(\boldsymbol{\theta}^*) = 0$ for $\rho = R + 1, ..., K$, 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 Eq. (18) with $V_\epsilon^{(\mathrm{d})}$ obtained by adding a random perturbation of strength ϵ to its elements. This is done by setting $V_\epsilon^{(\mathrm{d})} = \mathrm{ortho}(V^{(\mathrm{d})} + \epsilon G)$, 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.

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

$$V_{\nu,\sigma} \approx \sum_{a_1,...,a_{M-1}=1}^{\chi} \mathcal{V}_{\sigma,a_1}^{[1]\,\nu_1} \mathcal{V}_{a_1,a_2}^{[2]\,\nu_2} ... \mathcal{V}_{a_{M-1},1}^{[M]\,\nu_M} , \tag{30}$$

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

$$\tag{31}$$

The orthogonal matrix U is decomposed as an orthogonal matrix product operator (MPO)

$$U_{\mu,\rho} \approx \sum_{a_1,...,a_{N-1}=1}^{\chi} \mathcal{U}_{1,a_1}^{[1]\,\mu_1,\rho_1} \mathcal{U}_{a_1,a_2}^{[2]\,\mu_2,\rho_2} ... \mathcal{U}_{a_{N-1},1}^{[M]\,\mu_N,\rho_N} , \tag{32}$$

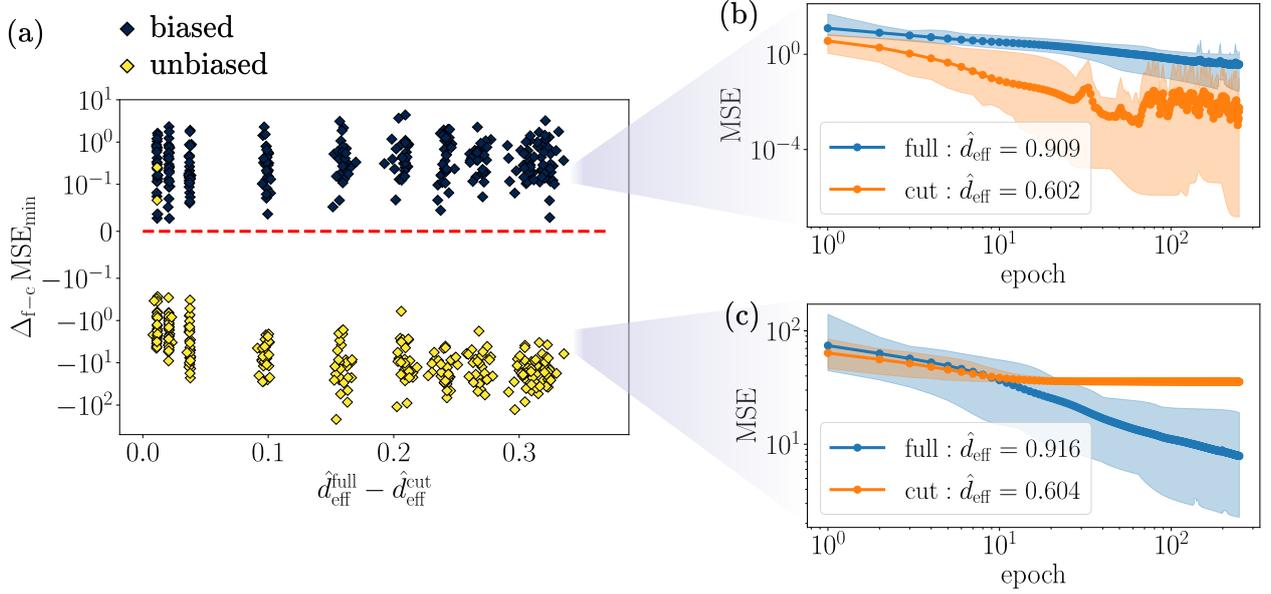Figure 7: (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ for different values of $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$, for biased (blue points) and unbiased (yellow points) models. Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ averaged 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. (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, $N = 1$, $\Omega = \{1, ..., 17\}$ ($d = 35$), $\tilde{\Omega} = \{1\}$ ($\tilde{d} = 3$), $M = 32$, $R = 7$, $\chi = 60$, $\mathfrak{n}_{\mathrm{train}} = 30$ with a batch size of 5.

Figure 8: (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ for different values of $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$, for different values of δ_{data} (color scale). Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ 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, $N = 1$, $\Omega = \{1, ..., 17\}$ ($d = 35$), $\tilde{\Omega} = \{1\}$ ($\tilde{d} = 3$), $M = 32$, $R = 7$, $\chi = 60$, $\mathfrak{n}_{\mathrm{train}} = 30$ 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

$$U_{\mu,\rho} = \qquad \approx \qquad \tag{33}$$

Finally, the isometry T can be decomposed as a tree tensor network (TTN) with the following diagrammatic representation

$$T_{\rho,\sigma} \approx \qquad \tag{34}$$

where the tensors $\mathcal{T}_{[\ell]}^{[\tau]}$ 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 Figs. 7 and Figs. 8, 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 ($\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ 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.

In Fig. 9 we show how $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ behaves as a function of M and D for different sizes $\mathfrak{n}_{\mathrm{train}}$. The dependence of $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ on M and D is mostly visible in the biased case (panels (a) and (c)). In particular, $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ rapidly decreases as a function of M reaching an approximately constant (but still positive) value for $M \geq D$ (Fig. 9(a)). The regime $M \geq D$ corresponds to the overparameterized regime [41] where the number of parameters is larger than the dimension of the space of functions accessible by the 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 $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ since the model with high ED can effectively rely on more redundant parameters for finding the global minimum of the loss. Consistently, $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ rapidly increases as a function of D reaching an approximately constant value for $D \geq M$ (Fig. 9(c)). The regime $D \geq M$ 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.
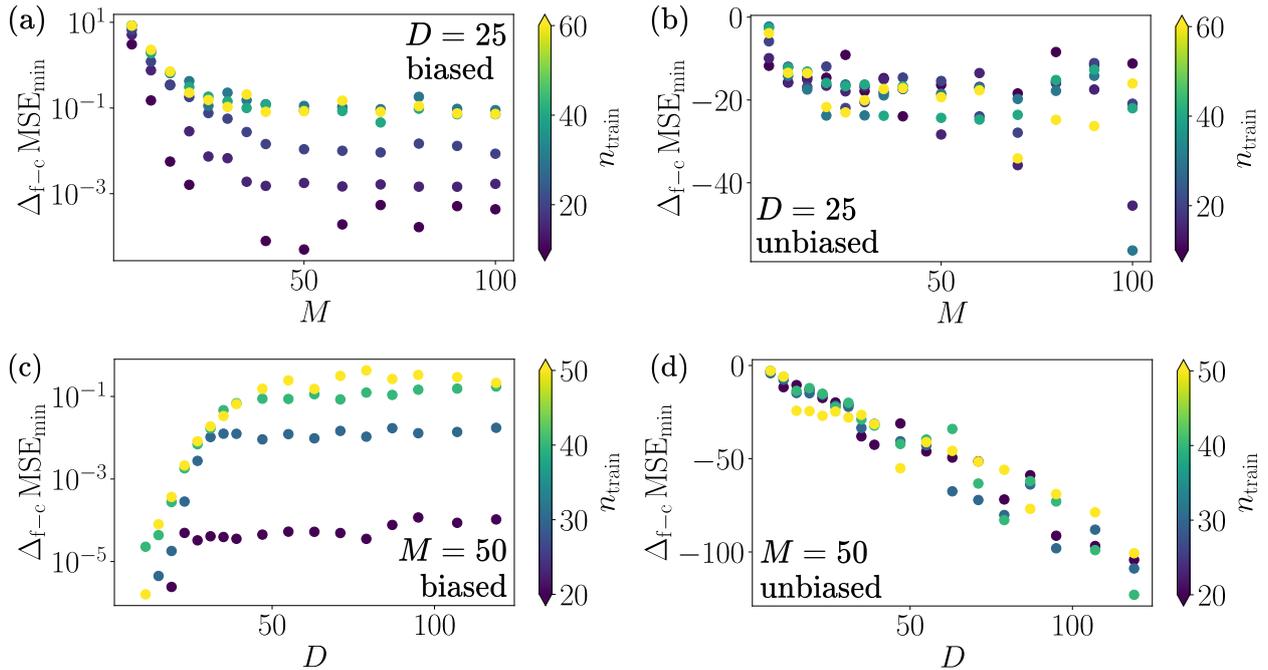
Figure 9: (a) and (b) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ vs. M for biased and unbiased models, respectively. Here $N = 1$, $D = 25$, $\tilde{\Omega} = \{1\}$ ($\tilde{d} = 3$), $R = 3$ and $\chi = 50$. (c) and (d) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ vs. D for biased and unbiased models, respectively. Here $N = 1$, $M = 50$, $\tilde{\Omega} = \{1\}$ ($\tilde{d} = 3$), $R = 2$ and $\chi = 120$. In all panels, each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ 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 $\mathfrak{n}_{\mathrm{train}}$, with a batch size of 5.

References

[1] Alejandro Perdomo-Ortiz, Marcello Benedetti, John Realpe-Gómez, and Rupak Biswas. Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. *Quantum Science and Technology*, 3(3):030502, jun 2018.

[2] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, nov 2019.

[3] M. Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J. Coles. Challenges and opportunities in quantum machine learning. *Nat. Comput. Sci.*, 2:567–576, 2022.

[4] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors, 2018.

[5] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3:625–644, 2021.

[6] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.*, 94:015004, Feb 2022.

[7] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018.

[8] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. Expressive power of parametrized quantum circuits. *Phys. Rev. Res.*, 2:033125, Jul 2020.

[9] Zhan Yu, Qiuhao Chen, Yuling Jiao, Yinan Li, Xiliang Lu, Xin Wang, and Jerry Zhijian Yang. Provable advantage of parameterized quantum circuit in function approximation Preprint at https://arxiv.org/abs/2310.07528, 2023.

[10] Matthias C. Caro, Elies Gil-Fuster, Johannes Jakob Meyer, Jens Eisert, and Ryan Sweke. Encoding-dependent generalization bounds for parametrized quantum circuits. *Quantum*, 5:582, November 2021.

[11] Leonardo Banchi, Jason Pereira, and Stefano Pirandola. Generalization in quantum machine learning: A quantum information standpoint. *PRX Quantum*, 2:040321, Nov 2021.

[12] Matthias C. Caro, Hsin-Yuan Huang, M. Cerezo, Kunal Sharma, Andrew Sornborger, Lukasz Cincio, and Patrick J. Coles. Generalization in quantum machine learning from few training data. *Nat. Commun.*, 13(1), aug 2022.

[13] Tobias Haug and M. S. Kim. Generalization of quantum machine learning models using quantum fisher information metric. *Phys. Rev. Lett.*, 133:050603, Jul 2024.

[14] Samuel Yen-Chi Chen, Tzu-Chieh Wei, Chao Zhang, Haiwang Yu, and Shinjae Yoo. Quantum convolutional neural networks for high energy physics data analysis. *Phys. Rev. Res.*, 4:013231, Mar 2022.

[15] Tak Hur, Leeseok Kim, and Daniel K. Park. Quantum convolutional neural network for classical data classification. *Quantum Machine Intelligence*, 4:3, 2022.

[16] Leo Sünkel, Darya Martyniuk, Julia J. Reichwald, Andrei Morariu, Raja Havish Seggoju, Philipp Altmann, Christoph Roch, and Adrian Paschke. Hybrid quantum machine learning assisted classification of covid-19 from computed tomography scans. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, page 356–366. IEEE, September 2023.

[17] Kevin Shen, Bernhard Jobst, Elvira Shishenina, and Frank Pollmann. Classification of the fashion-mnist dataset on a quantum computer, 2024.

[18] Vasilis Belis, Kinga Anna Woźniak, Ema Puljak, Panagiotis Barkoutsos, Günther Dissertori, Michele Grossi, Maurizio Pierini, Florentin Reiter, Ivano Tavernelli, and Sofia Vallecorsa. Quantum anomaly detection in the latent space of proton collision events at the lhc. *Communications Physics*, 7:334, 2024.

[19] Sebastiano Corli, Lorenzo Moro, Daniele Dragoni, Massimiliano Dispenza, and Enrico Prati. Quantum machine learning algorithms for anomaly detection: a survey, 2024.

[20] Tiffany Duneau, Saskia Bruhn, Gabriel Matos, Tuomas Laakkonen, Katerina Saiti, Anna Pearson, Konstantinos Meichanetzidis, and Bob Coecke. Scalable and interpretable quantum natural language processing: an implementation on trapped ions, 2024.

[21] Borja Aizpurua, Saeed S. Jahromi, Sukhbinder Singh, and Roman Orus. Quantum large language models via tensor network disentanglers, 2024.

[22] Alona Sakhnenko, Julian Sikora, and Jeanette Lorenz. Buildung continuous quantum-classical bayesian neural networks for a classical clinical dataset. In *Proceedings of Recent Advances in Quantum Computing and Technology*, ReAQCT '24, page 62–72. ACM, June 2024.

[23] Donovan Slabbert, Matt Lourens, and Francesco Petruccione. Pulsar classification: comparing quantum convolutional neural networks and quantum support vector machines. *Quantum Machine Intelligence*, 6(2), September 2024.

[24] Lorenzo Pastori, Arthur Grundner, Veronika Eyring, and Mierk Schwabe. Quantum neural networks for cloud cover parameterizations in climate models, 2025.

[25] Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.

[26] Thomas Hubregtsen, Josef Pichlmeier, Patrick Stecher, and Koen Bertels. Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility, and entangling capability. *Quantum Machine Intelligence*, 3:9, 2021.

[27] Amira Abbas, David Sutter, Christa Zoufal, Aurelien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Comp. Sci.*, 1:403–409, 2021.

[28] Francisco Javier Gil Vidal and Dirk Oliver Theis. Input redundancy for parameterized quantum circuits. *Frontiers in Physics*, 8, 2020.

[29] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Phys. Rev. A*, 103:032430, Mar 2021.

[30] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. General parameter-shift rules for quantum gradients. *Quantum*, 6:677, March 2022.

[31] Berta Casas and Alba Cervera-Lierta. Multidimensional fourier series with quantum circuits. *Phys. Rev. A*, 107:062612, Jun 2023.

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

[33] Shun ichi Amari. Differential-geometrical methods in statistics. In *Lecture Notes in Statistics*. Springer New York, NY, 1985.

[34] Shun ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, may 1997.

[35] Razvan Pascanu and Yoshua Bengio. Revisiting natural gradient for deep networks, 2014.

[36] Jeffrey Pennington and Pratik Worah. The spectrum of the fisher information matrix of a single-hidden-layer neural network. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[37] Shun-ichi Amari, Ryo Karakida, and Masafumi Oizumi. Fisher information and natural gradient learning in random deep networks. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 694–702. PMLR, 16–18 Apr 2019.

[38] Ryo Karakida, Shotaro Akaho, and Shun ichi Amari. Universal statistics of fisher information in deep neural networks: mean field approach*. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124005, dec 2020.

[39] Tomohiro Hayase and Ryo Karakida. The spectrum of fisher information of deep networks achieving dynamical isometry. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 334–342. PMLR, 13–15 Apr 2021.

[40] Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Pathological spectra of the fisher information metric and its variants in deep neural networks. *Neural Computation*, 33(8):2274–2307, 07 2021.

[41] Martín Larocca, Nathan Ju, Diego García-Martín, Patrick J. Coles, and Marco Cerezo. Theory of overparametrization in quantum neural networks. *Nature Computational Science*, 3:542 – 551, 2023.

[42] V. Murg F. Verstraete and J.I. Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57(2):143–224, 2008.

[43] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.

[44] Shi-Ju Ran, Emanuele Tirrito, Cheng Peng, Xi Chen, Luca Tagliacozzo, Gang Su, and Maciej Lewenstein. *Tensor Network Contractions*. Springer Cham, 2020.

[45] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[46] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011. January 2011 Special Issue.

[47] B Pirvu, V Murg, J I Cirac, and F Verstraete. Matrix product operator representations. *New Journal of Physics*, 12(2):025012, feb 2010.

[48] C. Hubig, I. P. McCulloch, and U. Schollwöck. Generic construction of efficient matrix product operators. *Phys. Rev. B*, 95:035129, Jan 2017.

[49] Georgios Styliaris, Rahul Trivedi, David Perez-Garcia, and J. Ignacio Cirac. Matrix-product unitaries: Beyond quantum cellular automata. *Quantum*, 9:1645, February 2025.

[50] Y.-Y. Shi, L.-M. Duan, and G. Vidal. Classical simulation of quantum many-body systems with a tree tensor network. *Phys. Rev. A*, 74:022320, Aug 2006.

[51] L. Tagliacozzo, G. Evenbly, and G. Vidal. Simulation of two-dimensional quantum systems using a tree tensor network that exploits the entropic area law. *Phys. Rev. B*, 80:235127, Dec 2009.

[52] V. Murg, F. Verstraete, Ö. Legeza, and R. M. Noack. Simulating strongly correlated quantum systems with tree tensor networks. *Phys. Rev. B*, 82:205105, Nov 2010.

[53] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[54] Maria Schuld. Supervised quantum machine learning models are kernel methods, 2021.

[55] Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12:2914, 2021.

[56] Jonas Kübler, Simon Buchholz, and Bernhard Schölkopf. The inductive bias of quantum kernels. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12661–12673. Curran Associates, Inc., 2021.

[57] Evan Peters and Maria Schuld. Generalization despite overfitting in quantum machine learning models. *Quantum*, 7:1210, December 2023.

[58] Yehuda Dar, Vidya Muthukumar, and Richard G. Baraniuk. A farewell to the bias-variance tradeoff? an overview of the theory of overparameterized machine learning, 2021.

[59] Jonas Landman, Slimane Thabet, Constantin Dalyac, Hela Mhiri, and Elham Kashefi. Classically approximating variational quantum machine learning with random fourier features, 2022.

[60] Franz J. Schreiber, Jens Eisert, and Johannes Jakob Meyer. Classical surrogates for quantum learning models. *Phys. Rev. Lett.*, 131:100803, Sep 2023.

[61] Ryan Sweke, Erik Recio, Sofiene Jerbi, Elies Gil-Fuster, Bryce Fuller, Jens Eisert, and Johannes Jakob Meyer. Potential and limitations of random fourier features for dequantizing quantum machine learning, 2023.

[62] Seongwook Shin, Yong Siah Teo, and Hyunseok Jeong. Dequantizing quantum machine learning models using tensor networks. *Phys. Rev. Res.*, 6:023218, May 2024.

[63] P. W. Brouwer and C. W. J. Beenakker. Diagrammatic method of integration over the unitary group, with applications to quantum transport in mesoscopic systems. *Journal of Mathematical Physics*, 37(10):4904–4934, 10 1996.

[64] B. Collins and P. Śniady. Integration with respect to the haar measure on unitary, orthogonal and symplectic group. *Communications in Mathematical Physics*, 264:773–795, 2006.

[65] Benoît Collins and Sho Matsumoto. On some properties of orthogonal weingarten functions. *Journal of Mathematical Physics*, 50(11):113516, 11 2009.

Supplementary Material

S1 Preliminaries: regression models and structure constants

We quickly recap our definition of structure constants for regression models that we use in the main text as well as in the derivations presented in this document. We consider regression models taking as input a vector $\boldsymbol{x} \in \mathbb{R}^N$, with N the number of input components, and parameterized by M trainable parameters $\boldsymbol{\theta} \in \mathbb{R}^M$. We focus on a single real output, in the case where it can be expanded in a finite number D and K of inputs' and parameters' basis functions $e_\mu(\boldsymbol{x})$ and $\iota_\nu(\boldsymbol{\theta})$, respectively, as

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{\mu=1}^{D} \sum_{\nu=1}^{K} \Gamma_{\mu,\nu} \, e_\mu(\boldsymbol{x}) \, \iota_\nu(\boldsymbol{\theta}) \ . \tag{35}$$

The coefficients $\Gamma \in \mathbb{R}^{D \times K}$ are called *structure constants* of the model, while the basis functions $e_\mu(\boldsymbol{x})$ are taken to form orthonormal bases in the D and K-dimensional spaces of input and parameters functions, respectively, i.e.,

$$\mathbb{E}_{\boldsymbol{x} \sim p}\big[e_\mu(\boldsymbol{x})e_{\mu'}(\boldsymbol{x})\big] = \int e_\mu(\boldsymbol{x})e_{\mu'}(\boldsymbol{x}) \, p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \delta_{\mu,\mu'} \ , \tag{36}$$

with $p(\boldsymbol{x})$ the probability density function for the inputs, and

$$\frac{1}{V_\Theta} \int_\Theta \iota_\nu(\boldsymbol{\theta})\iota_{\nu'}(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} = \delta_{\nu,\nu'} \ , \tag{37}$$

with Θ denoting the parameter space and V_Θ its volume. As discussed in the main text and later in this document, for a broad class of quantum neural networks (QNNs) we have $e_\mu(\boldsymbol{x}) \equiv e_{(\mu_1,\ldots,\mu_N)}(\boldsymbol{x}) = \prod_{n=1}^{N} e_{\mu_n}^{(n)}(x_n)$ and $\iota_\nu(\boldsymbol{\theta}) \equiv \iota_{(\nu_1,\ldots,\nu_M)}(\boldsymbol{\theta}) = \prod_{m=1}^{M} \iota_{\nu_m}^{(m)}(\theta_m)$, where

$$e_{\mu_n}^{(n)}(x_n) \in \mathcal{B}_n = \{1, \, \sqrt{2}\cos(\omega_n x_n), \, \sqrt{2}\sin(\omega_n x_n)\}_{\omega_n \in \Omega_n} \ , \tag{38}$$

$$\iota_{\nu_m}^{(m)}(\theta_m) \in \tilde{\mathcal{B}}_m = \{1, \, \sqrt{2}\cos(\tilde{\omega}_m \theta_m), \, \sqrt{2}\sin(\tilde{\omega}_m \theta_m)\}_{\tilde{\omega}_m \in \tilde{\Omega}_m} \ , \tag{39}$$

normalized in the interval $[-\pi, \pi]$, with Ω_n and $\tilde{\Omega}_m$ finite sets of frequencies the QNN has access to. In the common situation where the inputs are encoded multiple times via the re-uploading technique [29, 32], and both input features and parameters are encoded as angles of single qubit rotations, 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.

The structure constants $\Gamma_{\mu,\nu}$ can be viewed as elements of a $D \times K$ real matrix Γ which admits the following singular value decomposition (SVD)

$$\Gamma = USV^\top \ , \tag{40}$$

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 $s_1 \geq s_2 \geq \ldots \geq s_D$), and V is a $K \times D$ real matrix with orthonormal columns, i.e., $V^\top V = I_D$. The singular values s_ρ control the correlations between the parameter space and the functions in the input space, and the set $\{s_\rho\}_\rho$ is therefore referred to as *correlation spectrum*.

S2 Basis functions and structure constants of QNNs

We now provide the explicit connection between the form of regression models introduced before and the outputs of quantum neural networks (QNNs). We define a QNN as function $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ defined by a parameterized quantum circuit (PQC) as [4, 29]

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \langle 0| \hat{U}_{\boldsymbol{\theta}}^\dagger(\boldsymbol{x}) \hat{M} \, \hat{U}_{\boldsymbol{\theta}}(\boldsymbol{x}) |0\rangle \equiv \langle \psi_{\boldsymbol{\theta}}(\boldsymbol{x})| \, \hat{M} \, |\psi_{\boldsymbol{\theta}}(\boldsymbol{x})\rangle \ , \tag{41}$$

where $\hat{U}_{\boldsymbol{\theta}}(\boldsymbol{x})$ is the unitary operator implemented by the PQC, acting on a $\mathcal{D}$-dimensional Hilbert space, and $\hat{M}$ is an observable whose expectation value over the parameterized state $|\psi_{\boldsymbol{\theta}}(\boldsymbol{x})\rangle$ corresponds to the QNN
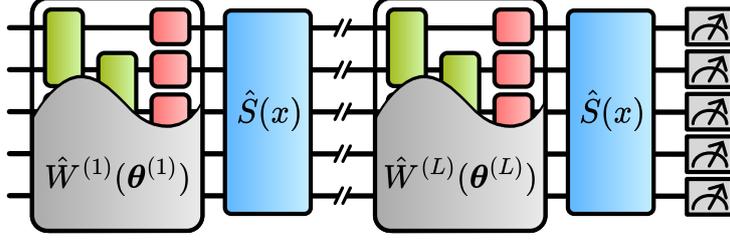
Figure S1: Schematics of the parameterized quantum circuit used as a QNN.

output. We consider a general PQC implementing L layers of data re-uploading [32, 29] with the following unitary (see Fig. S1 for a visualization)

$$\hat{U}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \prod_{\ell=1}^{L} \left[\hat{S}(\boldsymbol{x})\, \hat{W}^{(\ell)}(\boldsymbol{\theta}^{(\ell)}) \right] . \tag{42}$$

The $\hat{S}(\boldsymbol{x})$ is a unitary operator where the components of the input datum $\boldsymbol{x}$ are encoded as angles in rotation gates. The $\hat{W}^{(\ell)}(\boldsymbol{\theta}^{(\ell)})$ are variational blocks, where the trainable parameters $\boldsymbol{\theta}^{(\ell)}$ are also encoded as angles in rotation gates. The vector $\boldsymbol{\theta}$ summarizes the dependence on all $\boldsymbol{\theta}^{(\ell)}$, i.e., $\boldsymbol{\theta} = \{\boldsymbol{\theta}^{(\ell)}\}_{\ell=1,...,L}$.

S2.1   Fourier series representation of QNN

We now show that QNN outputs as Eq. (41) can be expressed as Eq. (35) with $e_\mu(\boldsymbol{x})$ and $\iota_\nu(\boldsymbol{\theta})$ trigonometric basis functions given by Eqs. (38) and (39). We show here for simplicity the case of a one-dimensional input x, while this can be easily generalized to more input features [29, 31] following the same derivation. Without loss of generality, we can write the variational blocks as

$$\hat{W}^{(\ell)}(\boldsymbol{\theta}^{(\ell)}) = \prod_{j_\ell=1}^{J_\ell} \hat{G}^{(\ell)}_{j_\ell}(\theta^{(\ell)}_{j_\ell}) , \tag{43}$$

where the dependence on the single parameters $\theta^{(\ell)}_{j_\ell}$, the components of $\boldsymbol{\theta}^{(\ell)}$, has been split into individual operators $\hat{G}^{(\ell)}_{j_\ell}$, as it is typically the case for QNNs where each parameter controls one rotation gate. We then switch to the diagonal representation of $\hat{S}(x)$ and $\hat{G}^{(\ell)}_{j_\ell}(\theta^{(\ell)}_{j_\ell})$

$$\hat{S}(x) = \hat{V}\, \hat{\Sigma}(x)\, \hat{V}^\dagger \; ; \quad \hat{G}^{(\ell)}_{j_\ell}(\theta^{(\ell)}_{j_\ell}) = \hat{Q}^{(\ell)}_{j_\ell}\, \hat{\Gamma}^{(\ell)}_{j_\ell}(\theta^{(\ell)}_{j_\ell})\, \hat{Q}^{(\ell)\,\dagger}_{j_\ell} , \tag{44}$$

with $\hat{\Sigma}(x) = \mathrm{diag}(\{e^{-i\lambda_\alpha x}\}_{\alpha=1,...,\mathcal{D}})$ and $\hat{\Gamma}^{(\ell)}_{j_\ell}(\theta^{(\ell)}_{j_\ell}) = \mathrm{diag}(\{e^{-i\eta^{(\ell,j_\ell)}_\beta \theta^{(\ell)}_{j_\ell}}\}_{\beta=1,...,\mathcal{D}})$. To ease the notation, we count the parameters with the index m replacing the labels (ℓ) and j_ℓ, i.e., $\theta^{(\ell)}_{j_\ell} \to \theta_m$, $\hat{Q}^{(\ell)}_{j_\ell} \to \hat{Q}_m$ and $\hat{\Gamma}^{(\ell)}_{j_\ell} \to \hat{\Gamma}_m$. Then,

$$
\begin{aligned}
\hat{U}_{\boldsymbol{\theta}}(x) = {}& \hat{V}\, \hat{\Sigma}(x)\, \hat{V}^\dagger \prod_{m=J_1+...+J_{L-1}+1}^{J_L} \hat{Q}_m\, \hat{\Gamma}_m(\theta_m)\, \hat{Q}^\dagger_m \times \\
& \hat{V}\, \hat{\Sigma}(x)\, \hat{V}^\dagger \prod_{m=J_1+...+J_{L-2}+1}^{J_{L-1}} \hat{Q}_m\, \hat{\Gamma}_m(\theta_m)\, \hat{Q}^\dagger_m \times \\
& ... \times \\
& \hat{V}\, \hat{\Sigma}(x)\, \hat{V}^\dagger \prod_{m=1}^{J_1} \hat{Q}_m\, \hat{\Gamma}_m(\theta_m)\, \hat{Q}^\dagger_m .
\end{aligned}
\tag{45}
$$

The components $\psi_i(x;\boldsymbol{\theta})$ of the state $|\psi_{\boldsymbol{\theta}}(x)\rangle = \hat{U}_{\boldsymbol{\theta}}(x)\,|0\rangle$ correspond to the elements $[\hat{U}_{\boldsymbol{\theta}}(x)]_{i,1}$ of the unitary $\hat{U}_{\boldsymbol{\theta}}(x)$. In order to arrive at the Fourier series expansion of $\psi_i(x;\boldsymbol{\theta})$ we start by considering the Fourier

23

expansion of the following sequence of unitaries:

$$
\begin{aligned}
\left[\hat{Q}_2\,\hat{\Gamma}_2(\theta_2)\,\hat{Q}_2^\dagger\,\hat{Q}_1\,\hat{\Gamma}_1(\theta_1)\,\hat{Q}_1^\dagger\right]_{k,1} &= \sum_{\beta_1}\left[\hat{Q}_2\,\hat{\Gamma}_2(\theta_2)\,\hat{Q}_2^\dagger\,\hat{Q}_1\,\hat{\Gamma}_1(\theta_1)\right]_{k,\beta_1}\left[\hat{Q}_1^\dagger\right]_{\beta_1,1} \\
&= \sum_{\beta_1}\left[\hat{Q}_2\,\hat{\Gamma}_2(\theta_2)\,\hat{Q}_2^\dagger\,\hat{Q}_1\right]_{k,\beta_1}\left[\hat{Q}_1^\dagger\right]_{\beta_1,1}\mathrm{e}^{-\mathrm{i}\eta^{(1)}_{\beta_1}\theta_1} \\
&= \sum_{\beta_1}\sum_{\gamma_1}\left[\hat{Q}_2\,\hat{\Gamma}_2(\theta_2)\,\hat{Q}_2^\dagger\right]_{k,\gamma_1}\left[\hat{Q}_1\right]_{\gamma_1,\beta_1}\left[\hat{Q}_1^\dagger\right]_{\beta_1,1}\mathrm{e}^{-\mathrm{i}\eta^{(1)}_{\beta_1}\theta_1} \\
&= \sum_{\beta_1,\beta_2}\sum_{\gamma_1}\left[\hat{Q}_2\,\hat{\Gamma}_2(\theta_2)\right]_{k,\beta_2}\left[\hat{Q}_2^\dagger\right]_{\beta_2,\gamma_1}\left[\hat{Q}_1\right]_{\gamma_1,\beta_1}\left[\hat{Q}_1^\dagger\right]_{\beta_1,1}\mathrm{e}^{-\mathrm{i}\eta^{(1)}_{\beta_1}\theta_1} \\
&= \sum_{\beta_1,\beta_2}\sum_{\gamma_1}\left[\hat{Q}_2\right]_{k,\beta_2}\left[\hat{Q}_2^\dagger\right]_{\beta_2,\gamma_1}\left[\hat{Q}_1\right]_{\gamma_1,\beta_1}\left[\hat{Q}_1^\dagger\right]_{\beta_1,1}\mathrm{e}^{-\mathrm{i}\eta^{(1)}_{\beta_1}\theta_1}\mathrm{e}^{-\mathrm{i}\eta^{(2)}_{\beta_2}\theta_2} \\
&\equiv \sum_{\beta_1,\beta_2}q^{(\beta_1,\beta_2)}_{k,1}\mathrm{e}^{-\mathrm{i}\eta^{(1)}_{\beta_1}\theta_1}\mathrm{e}^{-\mathrm{i}\eta^{(2)}_{\beta_2}\theta_2}\ ,
\end{aligned}
\tag{46}
$$

with $q^{(\beta_1,\beta_2)}_{k,1}\equiv\sum_{\gamma_1}\left[\hat{Q}_2\right]_{k,\beta_2}\left[\hat{Q}_2^\dagger\right]_{\beta_2,\gamma_1}\left[\hat{Q}_1\right]_{\gamma_1,\beta_1}\left[\hat{Q}_1^\dagger\right]_{\beta_1,1}$. This can be iterated over longer sequences, e.g.,

$$
\begin{aligned}
\left[\hat{Q}_3\,\hat{\Gamma}_3(\theta_3)\,\hat{Q}_3^\dagger\,\hat{Q}_2\,\hat{\Gamma}_2(\theta_2)\,\hat{Q}_2^\dagger\,\hat{Q}_1\,\hat{\Gamma}_1(\theta_1)\,\hat{Q}_1^\dagger\right]_{k,1} & \\
&\hspace{-6cm}= \sum_{\beta_1,\beta_2}\sum_{\gamma_2}\left[\hat{Q}_3\,\hat{\Gamma}_3(\theta_3)\,\hat{Q}_3^\dagger\right]_{k,\gamma_2}q^{(\beta_1,\beta_2)}_{\gamma_2,1}\mathrm{e}^{-\mathrm{i}\eta^{(1)}_{\beta_1}\theta_1}\mathrm{e}^{-\mathrm{i}\eta^{(2)}_{\beta_2}\theta_2} \\
&\hspace{-6cm}= \sum_{\beta_1,\beta_2,\beta_3}\sum_{\gamma_2}\left[\hat{Q}_3\,\hat{\Gamma}_3(\theta_3)\right]_{k,\beta_3}\left[\hat{Q}_3^\dagger\right]_{\beta_3,\gamma_2}q^{(\beta_1,\beta_2)}_{\gamma_2,1}\mathrm{e}^{-\mathrm{i}\eta^{(1)}_{\beta_1}\theta_1}\mathrm{e}^{-\mathrm{i}\eta^{(2)}_{\beta_2}\theta_2} \\
&\hspace{-6cm}= \sum_{\beta_1,\beta_2,\beta_3}\sum_{\gamma_2}\left[\hat{Q}_3\right]_{k,\beta_3}\left[\hat{Q}_3^\dagger\right]_{\beta_3,\gamma_2}q^{(\beta_1,\beta_2)}_{\gamma_2,1}\mathrm{e}^{-\mathrm{i}\eta^{(1)}_{\beta_1}\theta_1}\mathrm{e}^{-\mathrm{i}\eta^{(2)}_{\beta_2}\theta_2}\mathrm{e}^{-\mathrm{i}\eta^{(3)}_{\beta_3}\theta_3} \\
&\hspace{-6cm}\equiv \sum_{\beta_1,\beta_2,\beta_3}q^{(\beta_1,\beta_2,\beta_3)}_{k,1}\mathrm{e}^{-\mathrm{i}\eta^{(1)}_{\beta_1}\theta_1}\mathrm{e}^{-\mathrm{i}\eta^{(2)}_{\beta_2}\theta_2}\mathrm{e}^{-\mathrm{i}\eta^{(3)}_{\beta_3}\theta_3}\ .
\end{aligned}
\tag{47}
$$

Thus, iterating this calculation over all operators in the expression for $\psi_i(x;\boldsymbol{\theta})=\left[\hat{U}_{\boldsymbol{\theta}}(x)\right]_{i,1}$ (including also the $\hat{V}$ and $\hat{\Sigma}(x)$ operators from the encoding), we arrive at

$$
\psi_i(x;\boldsymbol{\theta})=\sum_{\boldsymbol{\alpha}}\sum_{\boldsymbol{\beta}}q^{(\boldsymbol{\alpha};\boldsymbol{\beta})}_{i,1}\,\mathrm{e}^{-\mathrm{i}\Lambda_{\boldsymbol{\alpha}}x}\,\mathrm{e}^{-\mathrm{i}\boldsymbol{\eta}_{\boldsymbol{\beta}}\cdot\boldsymbol{\theta}}\ ,
\tag{48}
$$

where we adopt the shorthand notation $\boldsymbol{\alpha}=(\alpha_1,...,\alpha_L)$, $\boldsymbol{\beta}=(\beta_1,...,\beta_M)$, $\Lambda_{\boldsymbol{\alpha}}=\sum_\ell\lambda_{\alpha_\ell}$ and $\boldsymbol{\eta}_{\boldsymbol{\beta}}=(\eta^{(1)}_{\beta_1},...,\eta^{(M)}_{\beta_M})$. The QNN output can be therefore expanded as

$$
\begin{aligned}
f_{\boldsymbol{\theta}}(x)=\langle\psi_{\boldsymbol{\theta}}(x)|\,\hat{M}\,|\psi_{\boldsymbol{\theta}}(x)\rangle &= \sum_{i,j}M_{i,j}\,\psi_i^*(x;\boldsymbol{\theta})\psi_j(x;\boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{\alpha},\boldsymbol{\alpha}'}\sum_{\boldsymbol{\beta},\boldsymbol{\beta}'}\mathrm{e}^{\mathrm{i}(\Lambda_{\boldsymbol{\alpha}}-\Lambda_{\boldsymbol{\alpha}'})x}\,\mathrm{e}^{\mathrm{i}(\boldsymbol{\eta}_{\boldsymbol{\beta}}-\boldsymbol{\eta}_{\boldsymbol{\beta}'})\cdot\boldsymbol{\theta}}\sum_{i,j}M_{i,j}\,q^{(\boldsymbol{\alpha};\boldsymbol{\beta})*}_{i,1}q^{(\boldsymbol{\alpha}';\boldsymbol{\beta}')}_{j,1} \\
&\equiv \sum_{\boldsymbol{\alpha},\boldsymbol{\alpha}'}\sum_{\boldsymbol{\beta},\boldsymbol{\beta}'}\tilde{\Gamma}_{(\boldsymbol{\alpha};\boldsymbol{\alpha}'),(\boldsymbol{\beta};\boldsymbol{\beta}')}\,\mathrm{e}^{\mathrm{i}(\Lambda_{\boldsymbol{\alpha}}-\Lambda_{\boldsymbol{\alpha}'})x}\,\mathrm{e}^{\mathrm{i}(\boldsymbol{\eta}_{\boldsymbol{\beta}}-\boldsymbol{\eta}_{\boldsymbol{\beta}'})\cdot\boldsymbol{\theta}}\ ,
\end{aligned}
\tag{49}
$$

with $\tilde{\Gamma}_{(\boldsymbol{\alpha};\boldsymbol{\alpha}'),(\boldsymbol{\beta};\boldsymbol{\beta}')}\equiv\sum_{i,j}M_{i,j}\,q^{(\boldsymbol{\alpha};\boldsymbol{\beta})*}_{i,1}q^{(\boldsymbol{\alpha}';\boldsymbol{\beta}')}_{j,1}$. We now denote the sets of frequencies generated by the encoding and variational gates as

$$
\Omega=\{\Lambda_{\boldsymbol{\alpha}}-\Lambda_{\boldsymbol{\alpha}'},\ \boldsymbol{\alpha},\boldsymbol{\alpha}'\in[\mathcal{D}]^L\}\ ,
\tag{50}
$$

and

$$
\tilde{\Omega}_m=\{\eta^{(m)}_{\beta_m}-\eta^{(m)}_{\beta'_m},\ \beta_m,\beta'_m=1,...,\mathcal{D}\}\ .
\tag{51}
$$

We can therefore write

$$
f_{\boldsymbol{\theta}}(x)=\sum_{\omega\in\Omega}\sum_{\tilde{\omega}_1\in\tilde{\Omega}_1}...\sum_{\tilde{\omega}_M\in\tilde{\Omega}_M}\tilde{\Gamma}_{\omega,(\tilde{\omega}_1,...,\tilde{\omega}_M)}\,\mathrm{e}^{\mathrm{i}\omega x}\prod_{m=1}^M\mathrm{e}^{\mathrm{i}\tilde{\omega}_m\theta_m}
\tag{52}
$$

and, after expressing the Fourier components in terms of real trigonometric basis functions and normalizing them in the chosen input and parameter space, we arrive at the desired form of Eqs. (35), (38) and (39).
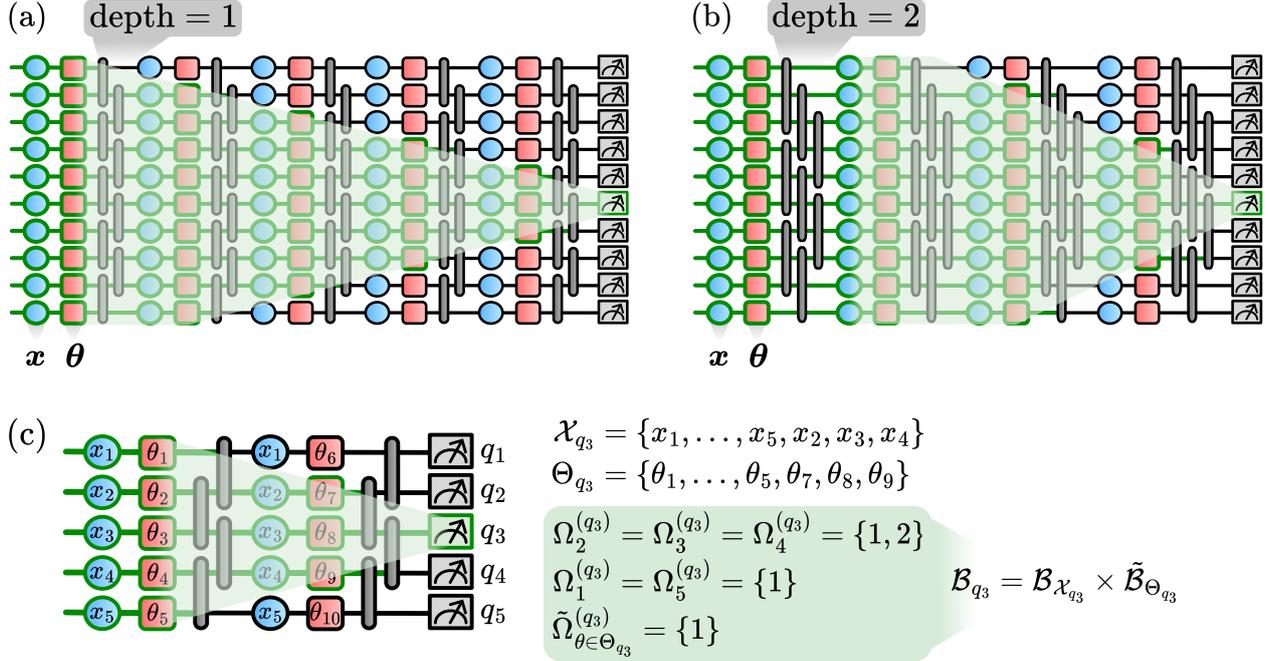
Figure S2: Schematics of the construction of the backward light-cones (BWLCs) for calculating the admissible QNN basis states. The blue and red single-qubit gates are the gates where the input features x_n and the parameters θ_m are encoded, respectively. The gray gates are multi-qubit entangling gates. The green shaded areas denote the extension of the BWLC from a given measured qubit. The gates belonging to the BWLC are framed in green. (a) For entangling layers coupling only nearest-neighbor qubits on a chain (depth one), the BWLC increases by two qubit lines for every layer of entangling gates. (b) For entangling layers coupling up to next-nearest-neighbor qubits on a chain (depth two), the BWLC spreads faster, i.e., by four qubit lines for every layer of entangling gates. (c) Explicit example of construction of BWLC for the third qubit of a five-qubits QNN. $\mathcal{X}_{q_3}$ and Θ_{q_3} are the sets of input features and parameters contained in the BWLC, respectively, and include how many times a given feature or parameter appears. The set of Fourier frequencies $\Omega_n^{(q_3)}$ capture this multiplicity (i.e., $\Omega_n^{(q_3)} = \{1, ..., L\}$ if x_n appears L times in $\mathcal{X}_{q_3}$). The sets of frequencies $\Omega_n^{(q_3)}$ and $\tilde{\Omega}_\theta^{(q_3)}$ define the basis functions $\mathcal{B}_{\mathcal{X}_{q_3}}$ and $\tilde{\mathcal{B}}_{\Theta_{q_3}}$. The functions obtained by measuring q_3 belong to the product space $\mathrm{span}(\mathcal{B}_{\mathcal{X}_{q_3}}) \otimes \mathrm{span}(\tilde{\mathcal{B}}_{\Theta_{q_3}})$.

S2.2 Dependence of structure constants on QNN entangling layers

Here we discuss how the structure of the entangling layers in a QNN influences the space of functions the model has access to. For simplicity, we consider the situation 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). We define an *encoding layer* to be the sequence of single-qubit gates encoding the input features on all qubits as

$$\hat{S}(\boldsymbol{x}) = \prod_{n=1}^{N} e^{-i\frac{x_n}{2}\boldsymbol{n}\cdot\hat{\boldsymbol{\sigma}}_n} \ , \tag{53}$$

where we assume for simplicity that the number of qubits equals the number of features N. Similarly, we define a *variational layer* to be the sequence of single-qubit gates encoding N variational parameters $\boldsymbol{\theta}^{(\ell)}$ on all qubits as

$$\hat{R}(\boldsymbol{\theta}^{(\ell)}) = \prod_{n=1}^{N} e^{-i\frac{\theta_n^{(\ell)}}{2}\boldsymbol{n}\cdot\hat{\boldsymbol{\sigma}}_n} \ . \tag{54}$$

When using the data re-uploading technique, encoding and variational blocks are repeated multiple times and interleaved with *entangling layers* $\hat{V}$, i.e., sequences of multi-qubit gates. We assume these gates to be fixed, i.e., not parameterized by any variational parameter. At the end of the sequence, a (optional) measurement unitary $\hat{W}_M$ is performed, accounting for the rotation to the measurement basis (assuming for simplicity only

one measurement basis is used). The resulting unitary is then

$$\hat{U}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \prod_{\ell=1}^{L} \left[\hat{S}(\boldsymbol{x}) \, \hat{R}(\boldsymbol{\theta}^{(\ell)}) \, \hat{V} \right] \hat{W}_M \ , \tag{55}$$

which corresponds to the structure discussed previously setting $\hat{W}^{(\ell)}(\boldsymbol{\theta}^{(\ell)}) = \hat{R}(\boldsymbol{\theta}^{(\ell)}) \hat{V}$. We consider the situation where (after the unitary $\hat{W}_M$), all qubits are measured in a fixed basis, e.g., the z basis, and the outputs summed together to form

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{n=1}^{N} \langle 0| \, \hat{U}_{\boldsymbol{\theta}}^{\dagger}(\boldsymbol{x}) \, \hat{\sigma}_n^z \, \hat{U}_{\boldsymbol{\theta}}(\boldsymbol{x}) \, |0\rangle \ , \tag{56}$$

which corresponds to the structure discussed previously setting $\hat{M} = \sum_{n=1}^{N} \hat{\sigma}_n^z$. Our goal here is to investigate how the structure of the entangling operations $\hat{V}$ and potentially $\hat{W}_M$ influences the structure constants Γ of the model. More specifically, we define the *depth* of the entangling (and measurement) layers as the number of qubits that are entangled to a given one after applying the layer to a product state (i.e., on a linear chain, depth one corresponds to the layer consisting of nearest-neighbor two-qubit gates, depth two to next-nearest-neighbor three-qubit gates, and so on), and we study how the depth affects the basis functions $e_\mu(\boldsymbol{x})$ and $\iota_\nu(\boldsymbol{\theta})$ accessible to the model.

To do this, we introduce the concept of *backward light-cone* (BWLC) of a given measured qubit (see Fig. S2 for a schematic representation). The BWLC of a measured qubit q_n is the set of input features $\mathcal{X}_{q_n}$ and the set of parameters Θ_{q_n} the one-qubit reduced density matrix for q_n depends on, defined as

$$\hat{\varrho}_{q_n}(\boldsymbol{x}, \boldsymbol{\theta}) = \mathrm{tr}_{\overline{q_n}} \big[\hat{U}_{\boldsymbol{\theta}}(\boldsymbol{x}) \, |0\rangle \langle 0| \, \hat{U}_{\boldsymbol{\theta}}^{\dagger}(\boldsymbol{x}) \big] \ . \tag{57}$$

Input features and parameters encoded via gates outside the BWLC cannot influence $\hat{\varrho}_{q_n}(\boldsymbol{x}, \boldsymbol{\theta})$, since their effect has not had 'time' to be propagated to q_n via the entangling gates. Importantly, $\mathcal{X}_{q_n}$ and Θ_{q_n} are defined to also account for the multiplicity a given feature or parameter appears in the BWLC. These therefore allow to define the sets of Fourier frequencies $\Omega_x^{(q_n)}$ (for $x \in \mathcal{X}_{q_n}$) and $\tilde{\Omega}_\theta^{(q_n)}$ (for $\theta \in \Theta_{q_n}$) for the Fourier series expansion of the elements of $\hat{\varrho}_{q_n}(\boldsymbol{x}, \boldsymbol{\theta})$, and therefore of the expectation value $\langle 0| \, \hat{U}_{\boldsymbol{\theta}}^{\dagger}(\boldsymbol{x}) \, \hat{\sigma}_n^z \, \hat{U}_{\boldsymbol{\theta}}(\boldsymbol{x}) \, |0\rangle$. For instance, if a given input feature $x \in \mathcal{X}_{q_n}$ is contained ℓ times in $\mathcal{X}_{q_n}$, then $\Omega_x^{(q_n)} = \{1, ..., \ell\}$. The sets of frequencies $\Omega_x^{(q_n)}$ and $\tilde{\Omega}_\theta^{(q_n)}$ define the basis functions $\mathcal{B}_x^{(q_n)}$ and $\tilde{\mathcal{B}}_\theta^{(q_n)}$

$$\mathcal{B}_x^{(q_n)} = \{1, \, \sqrt{2}\cos(\omega x), \, \sqrt{2}\sin(\omega x)\}_{\omega \in \Omega_x^{(q_n)}} \ , \tag{58}$$

$$\tilde{\mathcal{B}}_\theta^{(q_n)} = \{1, \, \sqrt{2}\cos(\tilde{\omega}\theta), \, \sqrt{2}\sin(\tilde{\omega}\theta)\}_{\tilde{\omega} \in \tilde{\Omega}_\theta^{(q_n)}} \ , \tag{59}$$

which can then be used to define inputs' and parameters' basis functions $\mathcal{B}_{\mathcal{X}_{q_n}}$ and $\tilde{\mathcal{B}}_{\Theta_{q_n}}$ for $\hat{\varrho}_{q_n}(\boldsymbol{x}, \boldsymbol{\theta})$ as

$$\mathcal{B}_{\mathcal{X}_{q_n}} = \bigtimes_{x \in \mathcal{X}_{q_n}} \mathcal{B}_x^{(q_n)} \quad \text{and} \quad \tilde{\mathcal{B}}_{\Theta_{q_n}} = \bigtimes_{\theta \in \Theta_{q_n}} \tilde{\mathcal{B}}_\theta^{(q_n)} \ , \tag{60}$$

i.e., as the product sets from all single-input (single-parameter) basis functions in the BWLC. The elements of $\hat{\varrho}_{q_n}(\boldsymbol{x}, \boldsymbol{\theta})$, as well as the functions obtained by measuring q_n, belong to the product space $\mathrm{span}(\mathcal{B}_{\mathcal{X}_{q_n}}) \otimes \mathrm{span}(\tilde{\mathcal{B}}_{\Theta_{q_n}})$, with basis set given by

$$\mathcal{B}_{q_n} = \mathcal{B}_{\mathcal{X}_{q_n}} \times \tilde{\mathcal{B}}_{\Theta_{q_n}} \ . \tag{61}$$

Importantly, the number of basis elements $|\mathcal{B}_{\mathcal{X}_{q_n}}| \leq (2L+1)^N$, and similarly $|\mathcal{B}_{\mathcal{X}_{q_n}}| \leq 3^M$, since in general the BWLC from q_n does not contain all encoding and variational gates. More specifically, the smaller the depth of the entangling layers (and of the measurement operator), the 'slower' the BWLC spreads, i.e., the smaller the sizes of $\mathcal{X}_{q_n}$ and Θ_{q_n}, and therefore $|\mathcal{B}_{\mathcal{X}_{q_n}}|$ and $|\mathcal{B}_{\mathcal{X}_{q_n}}|$, become. This is schematically shown in panels (a) and (b) of Fig. S2.

Since the QNN output is given by the sum of all $\langle 0| \, \hat{U}_{\boldsymbol{\theta}}^{\dagger}(\boldsymbol{x}) \, \hat{\sigma}_n^z \, \hat{U}_{\boldsymbol{\theta}}(\boldsymbol{x}) \, |0\rangle$, the basis functions it has access to are given by

$$\mathcal{B} = \bigcup_{n=1}^{N} \mathcal{B}_{q_n} \ , \tag{62}$$

which, depending on the depth of the entangling and measurement layers, may have less elements than the maximum number $(2L+1)^N \times 3^M$.
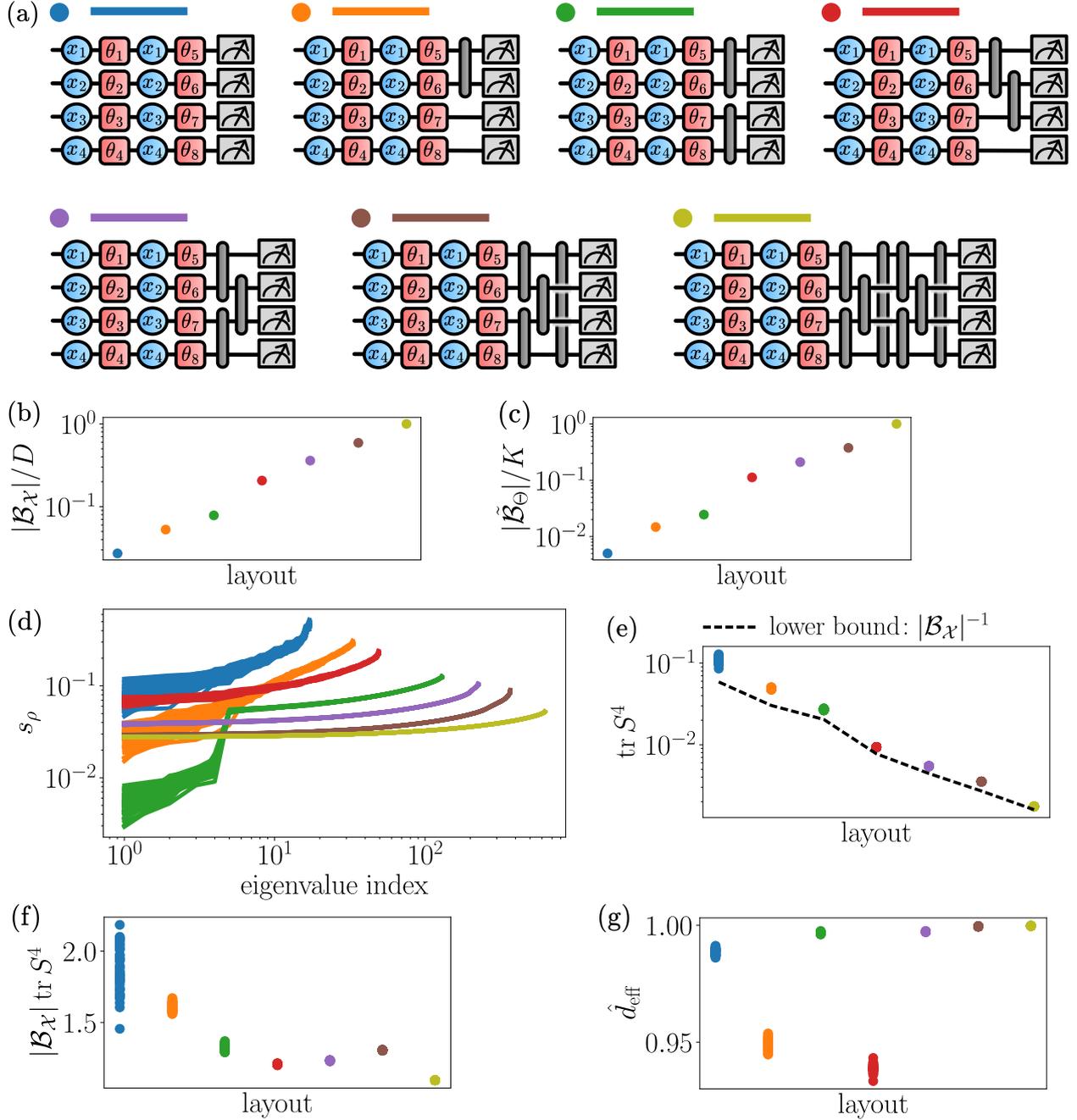
Figure S3: (a) QNN layouts used for the numerical analysis in this figure. Each color is related to a different structure of the two-qubits entangling gates in the measurement layer. (b) and (c) Scaling of $|\mathcal{B}_\mathcal{X}|$ and $|\tilde{\mathcal{B}}_\Theta|$, respectively, for the different layouts (with $\mathcal{B}_\mathcal{X} = \bigtimes_{n=1}^{N} \mathcal{B}_{\mathcal{X}_{q_n}}$ and $\tilde{\mathcal{B}}_\Theta = \bigtimes_{n=1}^{N} \tilde{\mathcal{B}}_{\Theta_{q_n}}$). Maximum values ($(2L+1)^N$ and 3^M, with $L=2$, $N=4$ and $M=8$) are obtained for entangling layers with depth scaling with N. (d) Correlation spectra for the different layouts (50 independent model draws per layout are shown). (e)-(f) Purity of correlation spectra for the different layouts, with lower bound $1/|\mathcal{B}_\mathcal{X}|$ shown by the black dashed line in (e), and normalized by the lower bound in (f). $|\mathcal{B}_\mathcal{X}| \operatorname{tr}(S^4)$ does not strongly depend on the layout. (g) Normalized ED for the different layouts: also the normalized ED does not strongly depend on the layout, mirroring the weak dependence of $|\mathcal{B}_\mathcal{X}| \operatorname{tr}(S^4)$ on the layout.

In Fig. S3 we show how the structure (depth) of the measurement layer in a simple QNN influences the number of basis functions, the correlation spectrum and the ED of the model. The different QNN layouts compared are shown in panel (a). For each model layout, for we consider random structure constants Γ uniformly drawn in $[-1, +1]^{D \times K}$ with the elements corresponding to basis functions outside the allowed basis set $\mathcal{B}$ set to zero. In panels (b) and (c) we show how the sizes of $\mathcal{B}_{\mathcal{X}} = \times_{n=1}^{N} \mathcal{B}_{\mathcal{X}_{q_n}}$ and $\tilde{\mathcal{B}}_{\Theta} = \times_{n=1}^{N} \tilde{\mathcal{B}}_{\Theta_{q_n}}$ scale for the different layouts, where we can observe that indeed the maximum size of the basis sets $((2L+1)^N$ and $3^M)$ is achieved when the depth is $\lfloor N/2 \rfloor$. For random model realizations corresponding to the different layouts, the correlation spectrum (shown in (d)) has a purity $\mathrm{tr}(S^4)$ whose value is close to the lower bound set by $|\mathcal{B}_{\mathcal{X}}|^{-1}$ as shown in panels (e) and (f). This in turn results in the normalized ED not having a strong dependence on the QNN entangling layer structure. Therefore, a more in depth investigation should include information on the actual distribution of the values of the structure constants (here taken to be uniform), which depends on the details of the individual gates in the QNN circuit.

S3 Definition of Fisher information matrix

We provide here further details on the definition of the Fisher information matrix (FIM), and how to potentially extend our results to the case of models used for classification, thus going beyond regression models.

S3.1 FIM for regression models

We derive the formula used for calculating the FIM in the case of regression with mean squared error (MSE) loss function. For a statistical model $p(\boldsymbol{x}, y; \boldsymbol{\theta})$ the elements of the FIM are defined as [33, 34, 35]

$$F_{j,k}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x}, y) \sim p} \left[\frac{\partial \log p(\boldsymbol{x}, y; \boldsymbol{\theta})}{\partial \theta_j} \frac{\partial \log p(\boldsymbol{x}, y; \boldsymbol{\theta})}{\partial \theta_k} \right], \tag{63}$$

which can be rewritten as

$$F_{j,k}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x}, y) \sim p} \left[\frac{\partial \log p_{\boldsymbol{\theta}}(y|\boldsymbol{x})}{\partial \theta_j} \frac{\partial \log p_{\boldsymbol{\theta}}(y|\boldsymbol{x})}{\partial \theta_k} \right], \tag{64}$$

by noting that $p(\boldsymbol{x}, y; \boldsymbol{\theta}) = p(\boldsymbol{x}) p_{\boldsymbol{\theta}}(y|\boldsymbol{x})$, with $p_{\boldsymbol{\theta}}(y|\boldsymbol{x})$ being the model output probability conditioned on the input $\boldsymbol{x}$. The quantity $-\log p_{\boldsymbol{\theta}}(y|\boldsymbol{x}) \equiv \ell(y, \boldsymbol{x}; \boldsymbol{\theta})$ corresponds to the negative log-likelihood, typically used as loss function for training statistical models. A statistical model corresponding to the MSE loss function can be constructed by setting $p_{\boldsymbol{\theta}}(y|\boldsymbol{x}) = \mathcal{N}_{f_{\boldsymbol{\theta}}(\boldsymbol{x}), \sigma^2}(y)$ with

$$\mathcal{N}_{f_{\boldsymbol{\theta}}(\boldsymbol{x}), \sigma^2}(y) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left[-\frac{(y - f_{\boldsymbol{\theta}}(\boldsymbol{x}))^2}{2\sigma^2} \right] \tag{65}$$

for a fictitious σ [36, 37, 38, 39, 40], with $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ being the deterministic regression value output by our model. Using $\partial_{\theta_j} p_{\boldsymbol{\theta}}(y|\boldsymbol{x}) = \sigma^{-2} (f_{\boldsymbol{\theta}}(\boldsymbol{x}) - y) \partial_{\theta_j} f_{\boldsymbol{\theta}}(\boldsymbol{x})$, we can rewrite the FIM as

$$F_{j,k}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}} \left[\int \mathcal{N}_{f_{\boldsymbol{\theta}}(\boldsymbol{x}), \sigma^2}(y) \, (f_{\boldsymbol{\theta}}(\boldsymbol{x}) - y)^2 \, \sigma^{-4} \, \frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \theta_j} \frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \theta_k} \, \mathrm{d}y \right]. \tag{66}$$

Performing the Gaussian integration over y, we finally arrive at

$$F_{j,k}(\boldsymbol{\theta}) = \sigma^{-2} \, \mathbb{E}_{\boldsymbol{x}} \left[\frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \theta_j} \frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \theta_k} \right], \tag{67}$$

which corresponds to the definition used when setting $\sigma = 1$. This definition makes it clear that the FIM can be interpreted as a metric tensor that that determines the response of the output of a model to a local change in the parameters, averaged over the input space, as

$$\mathbb{E}_{\boldsymbol{x}} \left[(f_{\boldsymbol{\theta}+\mathrm{d}\boldsymbol{\theta}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}}(\boldsymbol{x}))^2 \right] = \mathbb{E}_{\boldsymbol{x}} \left[\left(\sum_{j=1}^{M} \frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \theta_j} \mathrm{d}\theta_j \right)^2 \right] + \mathcal{O}(\|\mathrm{d}\boldsymbol{\theta}\|^2)$$

$$= \sum_{j,k} \mathbb{E}_{\boldsymbol{x}} \left[\frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \theta_j} \frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \theta_k} \right] \mathrm{d}\theta_j \mathrm{d}\theta_k + \mathcal{O}(\|\mathrm{d}\boldsymbol{\theta}\|^2) \tag{68}$$

$$\approx \mathrm{d}\boldsymbol{\theta}^{\top} F(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \, .$$

S3.2 From regression to probabilistic models

Our FIM analysis and arguments could be extended to the case of probabilistic models by making the following observation. Let us for simplicity consider the case of classification of discrete labels. For a given input $\boldsymbol{x}$ and parameters $\boldsymbol{\theta}$, the model outputs the probability $p_\ell(\boldsymbol{x};\boldsymbol{\theta})$ for a given class ℓ, with $\sum_\ell p_\ell(\boldsymbol{x};\boldsymbol{\theta}) = 1$. The FIM elements in this case are

$$
\begin{aligned}
F_{j,k}(\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{x},\ell}\left[\frac{\partial \log p_\ell(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_j}\frac{\partial \log p_\ell(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_k}\right] \\
&= \mathbb{E}_{\boldsymbol{x}}\left[\sum_\ell p_\ell(\boldsymbol{x};\boldsymbol{\theta})\frac{\partial \log p_\ell(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_j}\frac{\partial \log p_\ell(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_k}\right] \\
&= \mathbb{E}_{\boldsymbol{x}}\left[\sum_\ell \frac{1}{p_\ell(\boldsymbol{x};\boldsymbol{\theta})}\frac{\partial p_\ell(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_j}\frac{\partial p_\ell(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_k}\right] \\
&= 4\,\mathbb{E}_{\boldsymbol{x}}\left[\sum_\ell \frac{\partial \sqrt{p_\ell(\boldsymbol{x};\boldsymbol{\theta})}}{\partial \theta_j}\frac{\partial \sqrt{p_\ell(\boldsymbol{x};\boldsymbol{\theta})}}{\partial \theta_k}\right] \\
&\equiv 4\,\mathbb{E}_{\boldsymbol{x}}\left[\sum_\ell \frac{\partial q_\ell(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_j}\frac{\partial q_\ell(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_k}\right],
\end{aligned}
\tag{69}
$$

where we set $q_\ell(\boldsymbol{x};\boldsymbol{\theta}) \equiv \sqrt{p_\ell(\boldsymbol{x};\boldsymbol{\theta})}$. The above expression has now the same form of that for regression models, for which the analytical arguments presented in our work apply.

S4 Analysis of FIM for regression models

In this section we establish the connection between the FIM and the properties of the structure constants Γ, with particular focus on how the correlation spectrum S influences the spectral properties of the FIM.

S4.1 Derivation of diagrammatic FIM expression

Here we derive the analytical and diagrammatic expression of the FIM elements in terms of the structure constants Γ and the related correlation spectrum and singular vectors. We recall the definition of the (local) derivative tensor $\beta^{(j)}$, introduced as

$$
\frac{\partial \iota_\nu(\boldsymbol{\theta})}{\partial \theta_j} = \frac{\partial \iota_{\nu_j}^{(j)}(\theta_j)}{\partial \theta_j}\prod_{m\neq j}\iota_{\nu_m}^{(m)}(\theta_m) = \left(\sum_{\kappa_j}\beta_{\nu_j,\kappa_j}^{(j)}\iota_{\kappa_j}^{(j)}(\theta_j)\right)\prod_{m\neq j}\iota_{\nu_m}^{(m)}(\theta_m).
\tag{70}
$$

For example, if

$$
\iota_{\nu_j}^{(j)}(\theta_j) = \{1,\ \sqrt{2}\cos(\theta_j),\ ...,\ \sqrt{2}\cos(L\theta_j),\ \sqrt{2}\sin(\theta_j),\ ...,\ \sqrt{2}\sin(L\theta_j)\},
\tag{71}
$$

for some integer L, the derivative tensor $\beta^{(j)}$ is a $(2L+1)\times(2L+1)$ matrix of the form

$$
\beta^{(j)} = \left(\begin{array}{c|cccc|cccc}
0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\hline
0 & & & & & 1 & & & \\
0 & & & & & & 2 & & \\
\vdots & & \mathbf{0} & & & & & \ddots & \\
0 & & & & & & & & L \\
\hline
0 & -1 & & & & & & & \\
0 & & -2 & & & & & \mathbf{0} & \\
\vdots & & & \ddots & & & & & \\
0 & & & & -L & & & &
\end{array}\right),
\tag{72}
$$

The derivative of the model output $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ takes then the form

$$
\begin{aligned}
\frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \theta_j} &= \sum_{\mu=1}^{D} e_\mu(\boldsymbol{x}) \sum_{\nu=1}^{K} \Gamma_{\mu,\nu} \frac{\partial \iota_\nu(\boldsymbol{\theta})}{\partial \theta_j} \\
&= \sum_{\mu=1}^{D} e_\mu(\boldsymbol{x}) \sum_{\nu=(\nu_1 \ldots \nu_M)} \Gamma_{\mu,(\nu_1 \ldots \nu_M)} \sum_{\kappa_j} \beta^{(j)}_{\nu_j,\kappa_j} \iota^{(j)}_{\kappa_j}(\theta_j) \prod_{m\neq j} \iota^{(m)}_{\nu_m}(\theta_m) \\
&= \sum_{\mu=1}^{D} e_\mu(\boldsymbol{x}) \sum_{\kappa_j} \sum_{\nu=(\nu_1 \ldots \nu_M)} \beta^{(j)}_{\kappa_j,\nu_j} \Gamma_{\mu,(\nu_1 \ldots \kappa_j \ldots \nu_M)} \iota_\nu(\boldsymbol{\theta}) \qquad (73) \\
&= \sum_{\mu=1}^{D} e_\mu(\boldsymbol{x}) \sum_{\nu=(\nu_1 \ldots \nu_M)} \iota_\nu(\boldsymbol{\theta}) \sum_{\rho=1}^{D} U_{\mu,\rho}\, s_\rho \sum_{\kappa_j} \beta^{(j)}_{\kappa_j,\nu_j} [V^\top]_{\rho,(\nu_1 \ldots \kappa_j \ldots \nu_M)} \\
&\equiv \sum_{\mu=1}^{D} \sum_{\nu=1}^{K} \Lambda^{(j)}_{\mu,\nu}\, e_\mu(\boldsymbol{x})\, \iota_\nu(\boldsymbol{\theta}) \ ,
\end{aligned}
$$

with $\Lambda^{(j)}_{\mu,\nu} \equiv \sum_{\rho=1}^{D} U_{\mu,\rho}\, s_\rho \sum_{\kappa_j} \beta^{(j)}_{\kappa_j,\nu_j} [V^\top]_{\rho,(\nu_1 \ldots \kappa_j \ldots \nu_M)}$. The element $F_{j,k}$ of the FIM then reads as

$$
\begin{aligned}
F_{j,k}(\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{x}}\left[\frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \theta_j} \frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \theta_k} \right] \\
&= \sum_{\mu,\mu'} \sum_{\nu,\nu'} \Lambda^{(j)}_{\mu,\nu} \Lambda^{(k)}_{\mu',\nu'}\, \iota_\nu(\boldsymbol{\theta})\, \iota_{\nu'}(\boldsymbol{\theta})\, \mathbb{E}_{\boldsymbol{x}}\left[e_\mu(\boldsymbol{x})\, e_{\mu'}(\boldsymbol{x}) \right] \\
&= \sum_{\mu} \sum_{\nu,\nu'} \Lambda^{(j)}_{\mu,\nu} \Lambda^{(k)}_{\mu,\nu'}\, \iota_\nu(\boldsymbol{\theta})\, \iota_{\nu'}(\boldsymbol{\theta}) \\
&= \sum_{\nu,\nu'} \iota_\nu(\boldsymbol{\theta})\, \iota_{\nu'}(\boldsymbol{\theta}) \sum_{\rho,\rho'} s_\rho\, s_{\rho'} \sum_{\mu} U_{\mu,\rho}\, U_{\mu,\rho'} \times \\
&\qquad \sum_{\kappa_j,\kappa'_k} \beta^{(j)}_{\kappa_j,\nu_j}[V^\top]_{\rho,(\nu_1 \ldots \kappa_j \ldots \nu_M)} \beta^{(k)}_{\kappa'_k,\nu'_k}[V^\top]_{\rho',(\nu'_1 \ldots \kappa'_k \ldots \nu'_M)} \qquad (74) \\
&= \sum_{\nu,\nu'} \iota_\nu(\boldsymbol{\theta})\, \iota_{\nu'}(\boldsymbol{\theta}) \sum_{\rho} s_\rho^2 \sum_{\kappa_j,\kappa'_k} \beta^{(j)}_{\kappa_j,\nu_j}[V^\top]_{\rho,(\nu_1 \ldots \kappa_j \ldots \nu_M)} \times \\
&\qquad \beta^{(k)}_{\kappa'_k,\nu'_k}[V^\top]_{\rho,(\nu'_1 \ldots \kappa'_k \ldots \nu'_M)} \\
&\equiv \sum_{\nu,\nu'} \mathcal{F}^{(j,k)}_{\nu,\nu'}\, \iota_\nu(\boldsymbol{\theta})\, \iota_{\nu'}(\boldsymbol{\theta}) \\
&\equiv \big(\boldsymbol{\iota}(\boldsymbol{\theta}) \big| \mathcal{F}^{(j,k)} \big| \boldsymbol{\iota}(\boldsymbol{\theta}) \big) \ ,
\end{aligned}
$$

where in the second line we used the fact that $\mathbb{E}_{\boldsymbol{x}}\left[e_\mu(\boldsymbol{x})\, e_{\mu'}(\boldsymbol{x}) \right] = \delta_{\mu,\mu'}$, in the fourth line $\sum_\mu U_{\mu,\rho}\, U_{\mu,\rho'} = \delta_{\rho,\rho'}$, and in the last line $|\boldsymbol{\iota}(\boldsymbol{\theta}))$ the K-dimensional vector with components $\iota_\nu(\boldsymbol{\theta})$. In the above equation, we define the $\mathcal{F}^{(j,k)}$ tensor as

$$
\begin{aligned}
\mathcal{F}^{(j,k)}_{\nu,\nu'} &= \sum_{\rho} s_\rho^2 \sum_{\kappa_j,\kappa'_k} \beta^{(j)}_{\kappa_j,\nu_j}[V^\top]_{\rho,(\nu_1 \ldots \kappa_j \ldots \nu_M)} \beta^{(k)}_{\kappa'_k,\nu'_k}[V^\top]_{\rho,(\nu'_1 \ldots \kappa'_k \ldots \nu'_M)} \\
&\equiv \left[(V^\top B_j)^\top S^2 (V^\top B_k) \right]_{\nu,\nu'} \ . \qquad (75)
\end{aligned}
$$

where $B_j = I_{\tilde{d}}^{(1)} \otimes I_{\tilde{d}}^{(2)} \otimes \ldots \otimes \beta^{(j)} \otimes \ldots \otimes I_{\tilde{d}}^{(M)}$ (with $I_{\tilde{d}}^{(k)}$ being the $\tilde{d}$-dimensional identity matrix acting on the function space 'local' to the k-th parameter).

The $\mathcal{F}^{(j,k)}$ tensor introduced above and the FIM elements can be naturally represented as tensor networks. This is useful for evaluating the FIM for tensorized models, for numerically studying models with larger number

of parameters and input features. The $\Lambda^{(j)}_{\mu,\nu}$ tensor

$$\Lambda^{(j)}_{\mu,\nu} = \tag{76}$$

The $\mathcal{F}^{(j,k)}_{\nu,\nu'}$ tensor

$$\mathcal{F}^{(j,k)}_{\nu,\nu'} = \tag{77}$$

which then leads to the tensor network expression of the FIM elements

$$F_{j,k}(\boldsymbol{\theta}) = \tag{78}$$

where $\boldsymbol{\iota}^{(j)}(\theta_j)$ is a rank-1 tensor (vector) whose $\tilde{d}$ components are the local basis functions evaluated at θ_j, i.e., $\boldsymbol{\iota}^{(j)}(\theta_j) = \left(\iota^{(j)}_1(\theta_j), ..., \iota^{(j)}_{\tilde{d}}(\theta_j)\right)$.

S4.2 Correlation bounds on FIM and effective dimension

We show here that the effective dimension (ED) is upper-bounded by the rank of the correlation spectrum D (in case $D < M$). To do this, it is sufficient to rewrite Eqs. (74) and (75) as

$$
\begin{aligned}
F_{j,k}(\boldsymbol{\theta}) &= \sum_{\rho=1}^{D} s_\rho^2 \left(\boldsymbol{\iota}(\boldsymbol{\theta}) \middle| B_j^\top P_\rho B_k \middle| \boldsymbol{\iota}(\boldsymbol{\theta})\right) = \sum_{\rho=1}^{D} s_\rho^2 \left(\boldsymbol{\iota}(\boldsymbol{\theta}) \middle| B_j^\top P_\rho^\top P_\rho B_k \middle| \boldsymbol{\iota}(\boldsymbol{\theta})\right) \\
&= \sum_{\rho=1}^{D} s_\rho^2 \, \alpha_j^\rho(\boldsymbol{\theta}) \, \alpha_k^\rho(\boldsymbol{\theta}) \left(\mathbf{v}_\rho \middle| \mathbf{v}_\rho\right) = \sum_{\rho=1}^{D} s_\rho^2 \, \alpha_j^\rho(\boldsymbol{\theta}) \, \alpha_k^\rho(\boldsymbol{\theta}) \ ,
\end{aligned}
\tag{79}
$$

with P_ρ being the projection on the subspace spanned by the vector $V_{\cdot,\rho}$, which satisfies $P_\rho^\top P_\rho = P_\rho$ used in the first line, and $P_\rho B_k \big| \boldsymbol{\iota}(\boldsymbol{\theta})\big) = \alpha_k^\rho(\boldsymbol{\theta})\big|\mathbf{v}_\rho\big)$ with $\big|\mathbf{v}_\rho\big) = V_{\cdot,\rho}$. This allows us to write

$$F(\boldsymbol{\theta}) = \sum_{\rho=1}^{D} s_\rho^2 \, P_{\boldsymbol{\alpha}^\rho(\boldsymbol{\theta})} \ , \tag{80}$$

with $P_{\boldsymbol{\alpha}^\rho(\boldsymbol{\theta})} = \boldsymbol{\alpha}^\rho(\boldsymbol{\theta}) \, \boldsymbol{\alpha}^\rho(\boldsymbol{\theta})^\top$, which is proportional to the projection onto the M-components vector $\boldsymbol{\alpha}^\rho(\boldsymbol{\theta}) = \left(\alpha_1^\rho(\boldsymbol{\theta}), ..., \alpha_M^\rho(\boldsymbol{\theta})\right)^\top$. Since $F(\boldsymbol{\theta})$ is a sum of at most D linearly independent projections, we conclude that $\operatorname{rank} F(\boldsymbol{\theta}) \leq D$. By the results of [27], the ED is upper-bounded by the maximal rank of $F(\boldsymbol{\theta})$, and hence by D in the regime $D < M$.

S4.3 Random matrix theory analysis of the FIM

We derive here the expected value and variance of the FIM elements over random realizations of the orthogonal matrix $V \in \mathrm{O}(K)$, with $\mathrm{O}(K)$ the group of $K \times K$ orthogonal matrices, using results from random matrix theory [63, 64, 65]. We recall the expression of the FIM elements derived in Section S4.1

$$F_{j,k}(\boldsymbol{\theta}) = \sum_{\nu,\nu'} \mathcal{F}_{\nu,\nu'}^{(j,k)} \, \iota_\nu(\boldsymbol{\theta}) \, \iota_{\nu'}(\boldsymbol{\theta}) \equiv \big(\boldsymbol{\iota}(\boldsymbol{\theta}) \big| \mathcal{F}^{(j,k)} \big| \boldsymbol{\iota}(\boldsymbol{\theta})\big) \;, \tag{81}$$

with $\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)$ the K-dimensional vector with components $\iota_\nu(\boldsymbol{\theta})$ and

$$\begin{aligned}
\mathcal{F}_{\nu,\nu'}^{(j,k)} &= \sum_\rho s_\rho^2 \sum_{\kappa_j,\kappa_k'} \beta_{\kappa_j,\nu_j}^{(j)} [V^\top]_{\rho,(\nu_1\dots\kappa_j\dots\nu_M)} \beta_{\kappa_k',\nu_k'}^{(k)} [V^\top]_{\rho,(\nu_1'\dots\kappa_k'\dots\nu_M')} \\
&\equiv \big[(V^\top B_j)^\top S^2 \, (V^\top B_k)\big]_{\nu,\nu'} \;.
\end{aligned} \tag{82}$$

We start by computing the expectation value of $F_{j,k}(\boldsymbol{\theta})$ over $\mathrm{O}(K)$.

$$\begin{aligned}
\mathbb{E}_{V\in\mathrm{O}(K)}\big[F_{j,k}(\boldsymbol{\theta})\big] &= \mathbb{E}_{V\in\mathrm{O}(K)}\big[\big(\boldsymbol{\iota}(\boldsymbol{\theta})\big|(V^\top B_j)^\top S^2 \, (V^\top B_k)\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)\big] \\
&= \big(\boldsymbol{\iota}(\boldsymbol{\theta})\big| B_j^\top \, \mathbb{E}_{V\in\mathrm{O}(K)}\big[V\,S^2\,V^\top\big] B_k \big|\boldsymbol{\iota}(\boldsymbol{\theta})\big) \\
&= \sum_{\nu,\nu'} \iota_\nu(\boldsymbol{\theta})\,\iota_{\nu'}(\boldsymbol{\theta}) \sum_{\kappa,\kappa'} [B_j^\top]_{\nu,\kappa}[B_k]_{\kappa',\nu'} \sum_\rho s_\rho^2 \, \mathbb{E}_{V\in\mathrm{O}(K)}\big[V_{\kappa,\rho}V_{\kappa',\rho}\big] \\
&= \frac{1}{K} \sum_{\nu,\nu'} \iota_\nu(\boldsymbol{\theta})\,\iota_{\nu'}(\boldsymbol{\theta}) \sum_\kappa [B_j^\top]_{\nu,\kappa}[B_k]_{\kappa,\nu'} \sum_\rho s_\rho^2 \\
&= \frac{\mathrm{tr}(S^2)}{K} \big(\boldsymbol{\iota}(\boldsymbol{\theta})\big| B_j^\top B_k \big|\boldsymbol{\iota}(\boldsymbol{\theta})\big) \;,
\end{aligned} \tag{83}$$

where in the third line we used [63, 64, 65]

$$\mathbb{E}_{V\in\mathrm{O}(K)}\big[V_{\kappa,\rho}V_{\kappa',\rho}\big] = \frac{\delta_{\kappa,\kappa'}}{K} \;. \tag{84}$$

In order to calculate the variance $\mathrm{Var}_{V\in\mathrm{O}(K)}\big[F_{j,k}(\boldsymbol{\theta})\big]$ we need the following identity [64, 65]

$$\begin{aligned}
\mathbb{E}_{V\in\mathrm{O}(K)}\big[V_{\alpha,\rho}V_{\beta,\rho}V_{\alpha',\rho'}V_{\beta',\rho'}\big] &= \\
&= \frac{K+1}{D_K^3}\delta_{\alpha,\beta}\,\delta_{\alpha',\beta'} - \frac{1}{D_K^3}\Big(\delta_{\alpha,\alpha'}\,\delta_{\beta,\beta'} + \delta_{\alpha,\beta'}\,\delta_{\alpha',\beta}\Big) + \\
&\quad \delta_{\rho,\rho'}\Big[\frac{K+1}{D_K^3}\delta_{\alpha,\alpha'}\,\delta_{\beta,\beta'} - \frac{1}{D_K^3}\Big(\delta_{\alpha,\beta}\,\delta_{\alpha',\beta'} + \delta_{\alpha,\beta'}\,\delta_{\alpha',\beta}\Big) + \\
&\quad \frac{K+1}{D_K^3}\delta_{\alpha,\beta'}\,\delta_{\alpha',\beta} - \frac{1}{D_K^3}\Big(\delta_{\alpha,\beta}\,\delta_{\alpha',\beta'} + \delta_{\alpha,\alpha'}\,\delta_{\beta,\beta'}\Big)\Big] \\
&= \frac{\delta_{\alpha,\beta}\,\delta_{\alpha',\beta'}}{K^2} + \frac{\delta_{\rho,\rho'}}{K^2}\Big(\delta_{\alpha,\alpha'}\,\delta_{\beta,\beta'} + \delta_{\alpha,\beta'}\,\delta_{\alpha',\beta}\Big) + \mathcal{O}(K^{-3}) \;,
\end{aligned} \tag{85}$$

with $D_K^3 \equiv K(K-1)(K+2)$. The variance $\text{Var}_{V \in \text{O}(K)}\big[F_{j,k}(\boldsymbol{\theta})\big]$ is then

$$
\begin{aligned}
\text{Var}_{V \in \text{O}(K)}\big[F_{j,k}(\boldsymbol{\theta})\big] &= \mathbb{E}_{V \in \text{O}(K)}\big[F_{j,k}(\boldsymbol{\theta})^2\big] - \mathbb{E}_{V \in \text{O}(K)}\big[F_{j,k}(\boldsymbol{\theta})\big]^2 \\
&= \sum_{\nu,n} \iota_\nu(\boldsymbol{\theta})\,\iota_n(\boldsymbol{\theta}) \sum_{\nu',n'} \iota_{\nu'}(\boldsymbol{\theta})\,\iota_{n'}(\boldsymbol{\theta}) \sum_{\rho,\rho'} s_\rho^2\, s_{\rho'}^2 \times \\
&\qquad \sum_{\alpha,\beta,\alpha',\beta'} [B_j^\top]_{\nu,\alpha}[B_k]_{\beta,n}[B_j^\top]_{\nu',\alpha'}[B_k]_{\beta',n'} \times \\
&\qquad \Big[\mathbb{E}_{V \in \text{O}(K)}\big[V_{\alpha,\rho}V_{\beta,\rho}V_{\alpha',\rho'}V_{\beta',\rho'}\big] - \\
&\qquad \mathbb{E}_{V \in \text{O}(K)}\big[V_{\alpha,\rho}V_{\beta,\rho}\big]\mathbb{E}_{V \in \text{O}(K)}\big[V_{\alpha',\rho'}V_{\beta',\rho'}\big]\Big] \\
&= \sum_\rho \frac{s_\rho^4}{K^2} \sum_{\nu,n} \iota_\nu(\boldsymbol{\theta})\,\iota_n(\boldsymbol{\theta}) \sum_{\nu',n'} \iota_{\nu'}(\boldsymbol{\theta})\,\iota_{n'}(\boldsymbol{\theta}) \times \\
&\qquad \sum_{\alpha,\beta} \Big([B_j^\top]_{\nu,\alpha}[B_k]_{\beta,n}[B_j^\top]_{\nu',\alpha}[B_k]_{\beta,n'} + \\
&\qquad [B_j^\top]_{\nu,\alpha}[B_k]_{\beta,n}[B_j^\top]_{\nu',\beta}[B_k]_{\alpha,n'}\Big) + \mathcal{O}(K^{-3}) \\
&= \frac{\text{tr}(S^4)}{K^2}\Big[\big(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_j^\top B_k\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)^2 + \\
&\qquad \big(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_j^\top B_j\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)\big(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_k^\top B_k\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)\Big] + \\
&\qquad \mathcal{O}(K^{-3})\,.
\end{aligned}
\tag{86}
$$

Thus we have

$$
\mathbb{E}_{V \in \text{O}(K)}\big[F_{j,k}(\boldsymbol{\theta})\big] = \frac{\text{tr}(S^2)}{K}\big(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_j^\top B_k\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)\,,
\tag{87}
$$

and

$$
\begin{aligned}
\text{Var}_{V \in \text{O}(K)}\big[F_{j,k}(\boldsymbol{\theta})\big] = \frac{\text{tr}(S^4)}{K^2}\Big[&\big(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_j^\top B_k\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)^2 + \\
&\big(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_j^\top B_j\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)\big(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_k^\top B_k\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)\Big] + \\
&\mathcal{O}(K^{-3})\,.
\end{aligned}
\tag{88}
$$

We now examine the expectation value $\big(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_j^\top B_k\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)$, to be able to make further statements about $F_{j,k}(\boldsymbol{\theta})$.

$$
\begin{aligned}
\big(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_j^\top B_k\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big) = \Big(\sum_{\nu_j,\nu_j'} \beta_{\nu_j,\nu_j'}^{(j)}\,\iota_{\nu_j}^{(j)}(\theta_j)\,\iota_{\nu_j'}^{(j)}(\theta_j)\Big)\Big(\sum_{\nu_k,\nu_k'} \beta_{\nu_k,\nu_k'}^{(k)}\,\iota_{\nu_k}^{(k)}(\theta_k)\,\iota_{\nu_k'}^{(k)}(\theta_k)\Big) \times \\
\prod_{m \neq j,k}\Big(\sum_{\nu_m} \iota_{\nu_m}^{(m)}(\theta_m)\,\iota_{\nu_m}^{(m)}(\theta_m)\Big)\,.
\end{aligned}
\tag{89}
$$

Recall the normalization of the basis functions $\iota_{\nu_m}^{(m)}(\theta_m)$, i.e.,

$$
\mathbb{E}_{\theta_m}\Big[\iota_{\nu_m}^{(m)}(\theta_m)\,\iota_{\nu_m'}^{(m)}(\theta_m)\Big] = \delta_{\nu_m,\nu_m'}\,.
\tag{90}
$$

Then, since $B_j^\top B_k$ is an off-diagonal matrix for $j \neq k$, the expectation value $\big(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_j^\top B_k\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)$ is suppressed in expectation over $\boldsymbol{\theta}$ in this case. On the other hand, for $j = k$, $B_j^\top B_j$ is diagonal and positive semi-definite, hence the expectation value $\big(\boldsymbol{\iota}(\boldsymbol{\theta})\big|B_j^\top B_k\big|\boldsymbol{\iota}(\boldsymbol{\theta})\big)$ has a magnitude scaling as $\mathcal{O}(K)$. Summarizing, in expectation over $V \in \text{O}(K)$ and $\boldsymbol{\theta}$ we have

$$
\begin{aligned}
\mathbb{E}\big[F_{j,k}\big] &\in \begin{cases} \mathcal{O}(1)\,\text{tr}(S^2)\,, & \text{for } j = k \\ \mathcal{O}(K^{-1})\,\text{tr}(S^2)\,, & \text{for } j \neq k \end{cases} \\
\text{Var}\big[F_{j,k}\big] &\in \mathcal{O}(1)\,\text{tr}(S^4)\,.
\end{aligned}
\tag{91}
$$

S5 Details on construction of tensorized models

Here we provide more details on the construction of the tensorized models used in the main text. This is based on the tensor network (TN) decomposition of the structure constants Γ, starting from the following approximation

$$\Gamma_{\mu,\nu} \approx \sum_{\rho} \sum_{\sigma=1}^{\chi} U_{\mu,\rho} \, T_{\rho,\sigma} \, s_{\sigma} \big[V^{\top}\big]_{\sigma,\nu} \, . \tag{92}$$

which differs from Eq. (40) 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 matrix V as a tensor-train [45, 46]

$$V_{\nu,\sigma} \approx \sum_{a_1,\ldots,a_{M-1}=1}^{\chi} \mathcal{V}^{[1]\,\nu_1}_{\sigma,a_1} \mathcal{V}^{[2]\,\nu_2}_{a_1,a_2} \ldots \mathcal{V}^{[M]\,\nu_M}_{a_{M-1},1} \, , \tag{93}$$

where $\mathcal{V}^{[m]}$ are rank-3 tensors satisfying the right-normalization condition

$$\sum_{\nu_m=1}^{\tilde{d}} \sum_{a_m=1}^{\chi} \mathcal{V}^{[m]\,\nu_m}_{a_{m-1},a_m} \mathcal{V}^{[m]\,\nu_m}_{a'_{m-1},a_m} = \delta_{a'_{m-1},a_{m-1}} \, , \tag{94}$$

in order for V to have orthonormal columns. The TN decomposition of V admits the following graphical representation

$$V_{\nu,\sigma} = \qquad \approx \qquad \tag{95}$$

In order to construct a random right-normalized tensor train representing V, it is sufficient to generate random tensors $\mathcal{V}^{[m]}$ by reshaping randomly generated matrices $\mathbb{V}^{[m]}$ with orthonormal columns (i.e., satisfying $\mathbb{V}^{[m]\,\top} \mathbb{V}^{[m]} = I$) with elements $\mathbb{V}^{[m]}_{(\nu_m,a_m),a_{m-1}} = \mathcal{V}^{[m]\,\nu_m}_{a_{m-1},a_m}$. These, by construction, satisfy the above right-normalization condition. The orthogonal matrix U can be decomposed as an orthogonal matrix product operator (MPO) [47, 48, 49]

$$U_{\mu,\rho} \approx \sum_{a_1,\ldots,a_{N-1}=1}^{\chi} \mathcal{U}^{[1]\,\mu_1,\rho_1}_{1,a_1} \mathcal{U}^{[2]\,\mu_2,\rho_2}_{a_1,a_2} \ldots \mathcal{U}^{[M]\,\mu_N,\rho_N}_{a_{N-1},1} \, , \tag{96}$$

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

$$U_{\mu,\rho} = \qquad \approx \qquad \tag{97}$$

In our numerical examples, for constructing random orthogonal MPOs we restrict to the case of the matrix U having a so-called 'staircase' structure, which graphically corresponds to

$$U_{\mu,\rho} = \qquad \approx \qquad \tag{98}$$

with $U^{[n,n+1]}$ being $d^2 \times d^2$ random orthogonal matrices that can be decomposed as 2-site orthogonal MPOs with bond dimension $\chi = d^2$ via SVD as follows

$$
\begin{aligned}
U^{[n,n+1]}_{(\mu_n,\rho_n),(\mu_{n+1},\rho_{n+1})} &= \sum_{a_n=1}^{\chi} \tilde{U}_{(\mu_n,\rho_n),a_n} \tilde{S}_{a_n,a_n} \left[\tilde{V}^\top\right]_{a_n,(\mu_{n+1},\rho_{n+1})} \\
&\equiv \sum_{a_n=1}^{\chi} \tilde{\mathcal{U}}^{[n]\,\mu_n,\rho_n}_{1,a_n} \tilde{\mathcal{U}}^{[n]\,\mu_{n+1},\rho_{n+1}}_{a_n,1} \ ,
\end{aligned}
\tag{99}
$$

with $\tilde{\mathcal{U}}^{[n]\,\mu_n,\rho_n}_{1,a_n} = \tilde{U}_{(\mu_n,\rho_n),a_n}\,\tilde{S}_{a_n,a_n}$ and $\tilde{\mathcal{U}}^{[n]\,\mu_{n+1},\rho_{n+1}}_{a_n,1} = \left[\tilde{V}^\top\right]_{a_n,(\mu_{n+1},\rho_{n+1})}$. The tensors $\mathcal{U}^{[n]}$ are then simply constructed as follows

$$
\mathcal{U}^{[n]\,\mu_n,\rho_n}_{a_{n-1},a_n} = \sum_{\rho'_n=1}^{d} \tilde{\mathcal{U}}^{[n-1]\,\mu_n,\rho'_n}_{a_{n-1},1} \tilde{\mathcal{U}}^{[n]\,\rho'_n,\rho_n}_{1,a_n} \ .
\tag{100}
$$
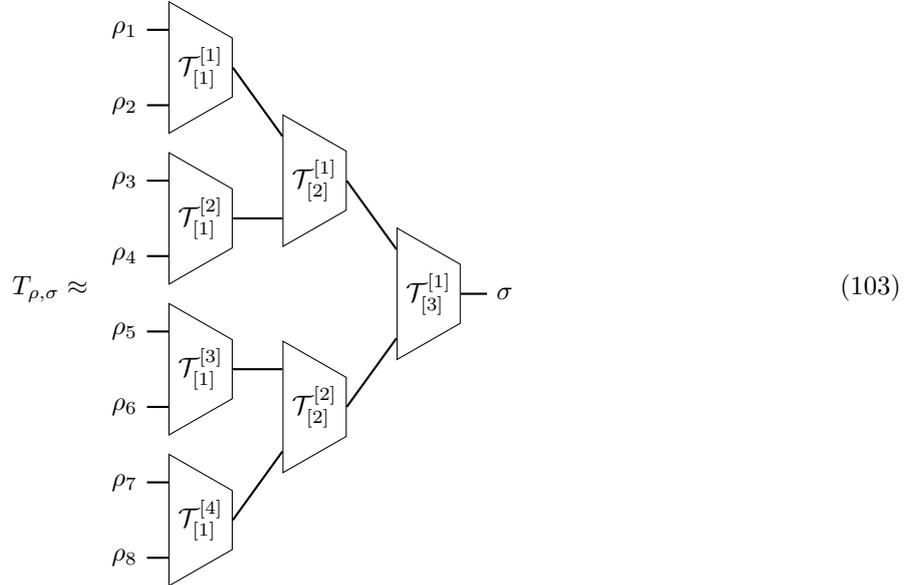
Finally, the isometry T can be decomposed as a tree tensor network (TTN) [50, 51, 52]

$$
T_{\rho,\sigma} \approx \sum_{\ell=1}^{\log_2 N-1} \sum_{\tau=1}^{N/2^{\ell+1}} \sum_{o_1^{\ell,\tau},o_2^{\ell,\tau}=1}^{\chi} \mathcal{T}^{[2\tau-1]\,o_1^{\ell,\tau}}_{[\ell]\,o_1^{\ell-1,2\tau-1},o_2^{\ell-1,2\tau-1}} \mathcal{T}^{[2\tau]\,o_2^{\ell,\tau}}_{[\ell]\,o_1^{\ell-1,2\tau},o_2^{\ell-1,2\tau}} \times
\atop
\mathcal{T}^{[\tau]\,o_{\lfloor\tau/2\rfloor+1}^{\ell+1,\lceil\tau/2\rceil}}_{[\ell+1]\,o_1^{\ell,\tau},o_2^{\ell,\tau}} \ ,
\tag{101}
$$

with $o_1^{0,\tau} = \rho_{2\tau-1}$, $o_2^{0,\tau} = \rho_{2\tau}$, $o_1^{\log_2 N,1} = \sigma$. The tensors $\mathcal{T}^{[\tau]}_{[\ell]}$ are isometric rank-3 tensors satisfying

$$
\sum_{i_1,i_2=1}^{\chi} \mathcal{T}^{[\tau]\,o}_{[\ell]\,i_1,i_2} \mathcal{T}^{[\tau]\,o'}_{[\ell]\,i_1,i_2} = \delta_{o',o} \ ,
\tag{102}
$$

in order for T to be an isometry. The TN decomposition of T has the following diagrammatic representation

$$
\tag{103}
$$

Generating random isometries $\mathcal{T}^{[\tau]}_{[\ell]}$ can be done by simply reshaping a randomly generated $\chi^2 \times \chi$ matrix $\mathbb{T}^{[\tau]}_{[\ell]}$ with χ orthonormal columns.

S6 Construction of biased and unbiased models

Here we provide details on the construction of biased and unbiased models used in the main text, for both non-tensorized and tensorized models. We define the data-generating function $y(\boldsymbol{x})$ to be

$$
y(\boldsymbol{x}) = \sum_{\mu=1}^{D} \sum_{\nu=1}^{K} e_\mu(\boldsymbol{x})\, \iota_\nu(\boldsymbol{\theta}^*) \sum_{\rho=1}^{R} s_\rho\, U^{(\mathrm{d})}_{\mu,\rho} \left[V^{(\mathrm{d})\,\top}\right]_{\rho,\nu} \ ,
\tag{104}
$$

with $R < D$, $\boldsymbol{\theta}^*$ a given parameter configuration, and $V^{(\mathrm{d})}$ satisfying the property $\sum_\nu V^{(\mathrm{d})}_{\nu,\rho}\, \iota_\nu(\boldsymbol{\theta}^*) = 0$ for $\rho = R+1, ..., K$. We generate the matrix $V^{(\mathrm{d})}$ as $V^{(\mathrm{d})} = \begin{bmatrix} \tilde{V} & \tilde{W} \end{bmatrix}$, i.e., by horizontally stacking a $K \times R$ matrix $\tilde{V}$ with randomly drawn orthonormal columns, and a $K \times (D-R)$ matrix $\tilde{W}$ constructed via Gram-Schmidt orthogonalization in order to satisfy the constraints

$$
\begin{cases}
\tilde{V}^\top \tilde{W} = 0 \\
\tilde{W}^\top \tilde{V} = 0 \\
\tilde{W}^\top |\boldsymbol{\iota}(\boldsymbol{\theta}^*)) = 0 \,, \text{ i.e., } \sum_\nu \tilde{W}_{\nu,\sigma}\, \iota_\nu(\boldsymbol{\theta}^*) = 0 \ \ \forall\, \sigma = 1, ..., D-R \,.
\end{cases}
\tag{105}
$$

As discussed in the main text, the construction of *biased* models, i.e., models that for $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ exactly match the data-generating function $y(\boldsymbol{x})$, is achieved by setting $U = U^{(\mathrm{d})}$ and $V = V^{(\mathrm{d})}$ in their structure constants (see Eq. (40)), while *unbiased* models are constructed by randomly drawing U and V independently of $U^{(\mathrm{d})}$ and $V^{(\mathrm{d})}$. The values s_ρ $(\rho = 1, ..., D)$ of the correlation spectrum are chosen to be $s_\rho = 1/\sqrt{D}$, i.e., a uniform correlation spectrum with normalization $\sum_\rho s_\rho^2 = 1$. This uniform choice is performed in order to have that *full* models $f_{\boldsymbol{\theta}}^{(\mathrm{f})}(\boldsymbol{x})$ specified by the structure constants

$$
\Gamma^{(\mathrm{f})}_{\mu,\nu} = \sum_{\rho=1}^{D} U_{\mu,\rho}\, s_\rho \left[V^\top \right]_{\rho,\nu} \,,
\tag{106}
$$

have, with high probability, a high effective dimension (since $\sum_\rho s_\rho^4 = 1/D$). *Cutoff* models $f_{\boldsymbol{\theta}}^{(\mathrm{c})}(\boldsymbol{x})$ specified by the structure constants

$$
\Gamma^{(\mathrm{c})}_{\mu,\nu} = \sum_{\rho=1}^{R} U_{\mu,\rho}\, s_\rho \left[V^\top \right]_{\rho,\nu} + \sum_{\rho=R+1}^{D} U_{\mu,\rho}\, \mathrm{e}^{-\frac{\rho-R}{\xi}} s_\rho \left[V^\top \right]_{\rho,\nu} \,,
\tag{107}
$$

with a positive decay rate ξ, have instead a lower effective dimension than full models, with high probability.

The construction of biased models for tensorized models follows the same idea, with suitable modifications to accommodate for the tensor structure of the matrix of singular vectors V. The idea is again to find a matrix $\tilde{W}$ such that $\sum_\nu \tilde{W}_{\nu,\sigma}\, \iota_\nu(\boldsymbol{\theta}^*) = 0 \ \ \forall\, \sigma = 1, ..., D-R$, where now $\tilde{W}$ has the following tensor-train decomposition

$$
\tilde{W}_{(\nu_1,...,\nu_M),\sigma} = \sum_{a_1,...,a_{M-1}} \mathcal{W}^{[1]\,\nu_1}_{\sigma,a_1} \mathcal{W}^{[2]\,\nu_2}_{a_1,a_2} ... \mathcal{W}^{[M]\,\nu_M}_{a_{M-1},1} \,.
\tag{108}
$$

The condition that $|\boldsymbol{\iota}(\boldsymbol{\theta}))$ is in the kernel of $\tilde{W}$ can be rewritten as

$$
\sum_{a_1,...,a_{M-1}} \prod_{m=1}^{M} \left[\sum_{\nu_m} \mathcal{W}^{[m]\,\nu_m}_{a_{m-1},a_m}\, \iota^{(m)}_{\nu_m}(\theta^*_m) \right] = 0 \,,
\tag{109}
$$

with $a_0 \equiv \sigma$ and $a_M \equiv 1$. We now construct the matrices $\mathcal{Q}^{[m]}(\theta^*_m)$ with elements

$$
\left[\mathcal{Q}^{[m]}(\theta^*_m) \right]_{a_{m-1},a_m} \equiv \sum_{\nu_m} \mathcal{W}^{[m]\,\nu_m}_{a_{m-1},a_m}\, \iota^{(m)}_{\nu_m}(\theta^*_m) \,,
\tag{110}
$$

and the vector $\boldsymbol{q}(\boldsymbol{\theta}^*_{2 \to M})$ with elements

$$
q_{a_1}(\boldsymbol{\theta}^*_{2 \to M}) \equiv \left[\mathcal{Q}^{[2]}(\theta^*_2) ... \mathcal{Q}^{[M]}(\theta^*_M) \right]_{a_1,1} \,.
\tag{111}
$$

Using these, we can turn the conditions on $\tilde{W}$ into the following conditions for the single tensor $\mathcal{W}^{[1]}$

$$
\begin{cases}
\sum_{a_1} \sum_{\nu_1} \mathcal{W}^{[1]\,\nu_1}_{\sigma,a_1} \mathcal{W}^{[1]\,\nu_1}_{\sigma',a_1} = \delta_{\sigma,\sigma'} \quad \text{from right-normalization condition} \\
\sum_{a_1} \sum_{\nu_1} \mathcal{W}^{[1]\,\nu_1}_{\sigma,a_1}\, q_{a_1}(\boldsymbol{\theta}^*_{2 \to M})\, \iota^{(1)}_{\nu_1}(\theta^*_1) = 0 \,,
\end{cases}
\tag{112}
$$

under the assumption that all other tensors $\mathcal{W}^{[m>1]}$ are already right-normalized. These conditions on the tensor $\mathcal{W}^{[1]}$ can be easily cast in matrix form, by grouping the indices $\gamma_1 \equiv (a_1, \nu_1)$, reshaping $\mathcal{W}^{[1]}$ into a matrix $\mathbb{W}^{[1]}$ with elements

$$
\mathbb{W}^{[1]}_{\sigma,\gamma_1} = \mathbb{W}^{[1]}_{\sigma,(a_1,\nu_1)} = \mathcal{W}^{[1]\,\nu_1}_{\sigma,a_1} \,,
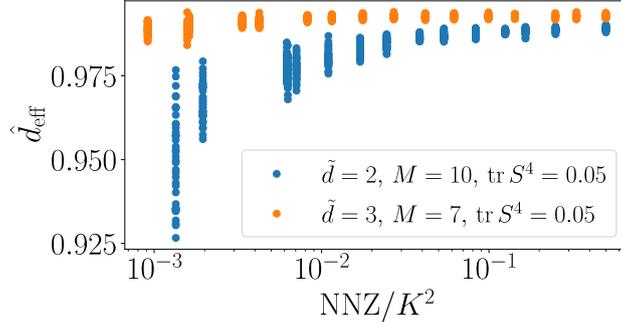\tag{113}
$$

Figure S4: Scaling of normalized ED with the sparsity of the orthogonal matrix V. On the x axis, NNZ corresponds to the number of non-zero elements of V (the lower NNZ, the more sparse V is). 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 NNZ, 40 model realizations are drawn. The normalized ED is computed using 150 parameters samples for estimating the normalized FIM.

and constructing the vector $\mathbf{v}(\boldsymbol{\theta}^*)$ as tensor product $\boldsymbol{q}(\boldsymbol{\theta}^*_{2 \to M}) \otimes \boldsymbol{\iota}^{(1)}(\theta^*_1)$, with elements

$$\mathrm{v}_{\gamma_1}(\boldsymbol{\theta}^*) = \mathrm{v}_{(a_1, \nu_1)}(\boldsymbol{\theta}^*) = q_{a_1}(\boldsymbol{\theta}^*_{2 \to M})\, \iota^{(1)}_{\nu_1}(\theta^*_1) \ . \tag{114}$$

In matrix form

$$\begin{cases} \mathbb{W}^{[1]}\, \mathbb{W}^{[1]\,\top} = I & \text{from right-normalization condition} \\ \mathbb{W}^{[1]}\, \mathbf{v}(\boldsymbol{\theta}^*) = \mathbf{0} \ . \end{cases} \tag{115}$$

Using this, the construction of fully biased tensorized models is summarized in the following steps

1. Choose a (random) parameter configuration $\boldsymbol{\theta}^*$.

2. Construct a set of (random) right-normalized tensors $\{\mathcal{W}^{[2]\,\nu_2}_{a_1, a_2}, ..., \mathcal{W}^{[M]\,\nu_M}_{a_{M-1}, 1}\}$ and compute the vector $\mathbf{v}(\boldsymbol{\theta}^*) = \boldsymbol{q}(\boldsymbol{\theta}^*_{2 \to M}) \otimes \boldsymbol{\iota}^{(1)}(\theta^*_1)$ as described before.

3. Via Gram-Schmidt orthogonalization, generate a random matrix $\mathbb{W}^{[1]}$ whose columns are normalized, mutually orthogonal and all orthogonal to $\mathbf{v}(\boldsymbol{\theta}^*)$. This matrix satisfies the above conditions by construction.

4. Reshape $\mathbb{W}^{[1]}$ to the order-three tensor $\mathcal{W}^{[1]}$.

5. The matrix $\tilde{W}_{(\nu_1, ..., \nu_M), \sigma} = \sum_{a_1, ..., a_{M-1}} \mathcal{W}^{[1]\,\nu_1}_{\sigma, a_1} \mathcal{W}^{[2]\,\nu_2}_{a_1, a_2} ... \mathcal{W}^{[M]\,\nu_M}_{a_{M-1}, 1}$ satisfies the desired orthogonality conditions for constructing a biased model.

Given the tensor-train representation

$$V^{(\mathrm{d})}_{(\nu_1, ..., \nu_M), \rho} = \sum_{a_1, ..., a_{M-1}} \mathcal{V}^{[1]\,\nu_1}_{\rho, a_1} \mathcal{V}^{[2]\,\nu_2}_{a_1, a_2} ... \mathcal{V}^{[M]\,\nu_M}_{a_{M-1}, 1} \ , \tag{116}$$

the construction of partially biased tensorized models is obtained by perturbing the tensors $\mathcal{V}^{[m]}$. Specifically we construct $V^{(\mathrm{d})}_\epsilon$ in the data-generating function from tensors $\mathcal{V}^{[m]}_\epsilon$ that are obtained by reshaping the orthogonal matrices $\mathbb{V}^{[m]}_\epsilon = \mathrm{ortho}(\mathbb{V}^{[m]} + \epsilon\, \mathbb{G})$, where $\mathbb{G}$ is a matrix with Gaussian entries.

S7   Further numerical results on effective dimension

Here we provide further numerical results on the dependence of the ED on various model characteristics. The results presented here confirm those shown in the main text, namely that the ED being primarily controlled by the number of input basis functions D when $D < M$, and by $\mathrm{tr}(S^4)$ in the regime $D > M$.

In Fig. S4 we show how the normalized ED depends on the sparsity of the matrix V containing the right singular vectors of the structure constants Γ. Two different model classes are shown: $\tilde{\mathcal{B}}_m = \{\cos\theta_m, \sin\theta_m\}$ (i.e., $\tilde{d} = 2$) and $\tilde{\mathcal{B}}_m = \{1, \cos\theta_m, \sin\theta_m\}$ (i.e., $\tilde{d} = 3$). From these results, it is evident that $\hat{d}_{\mathrm{eff}}$ does not strongly depends on the sparsity of V.
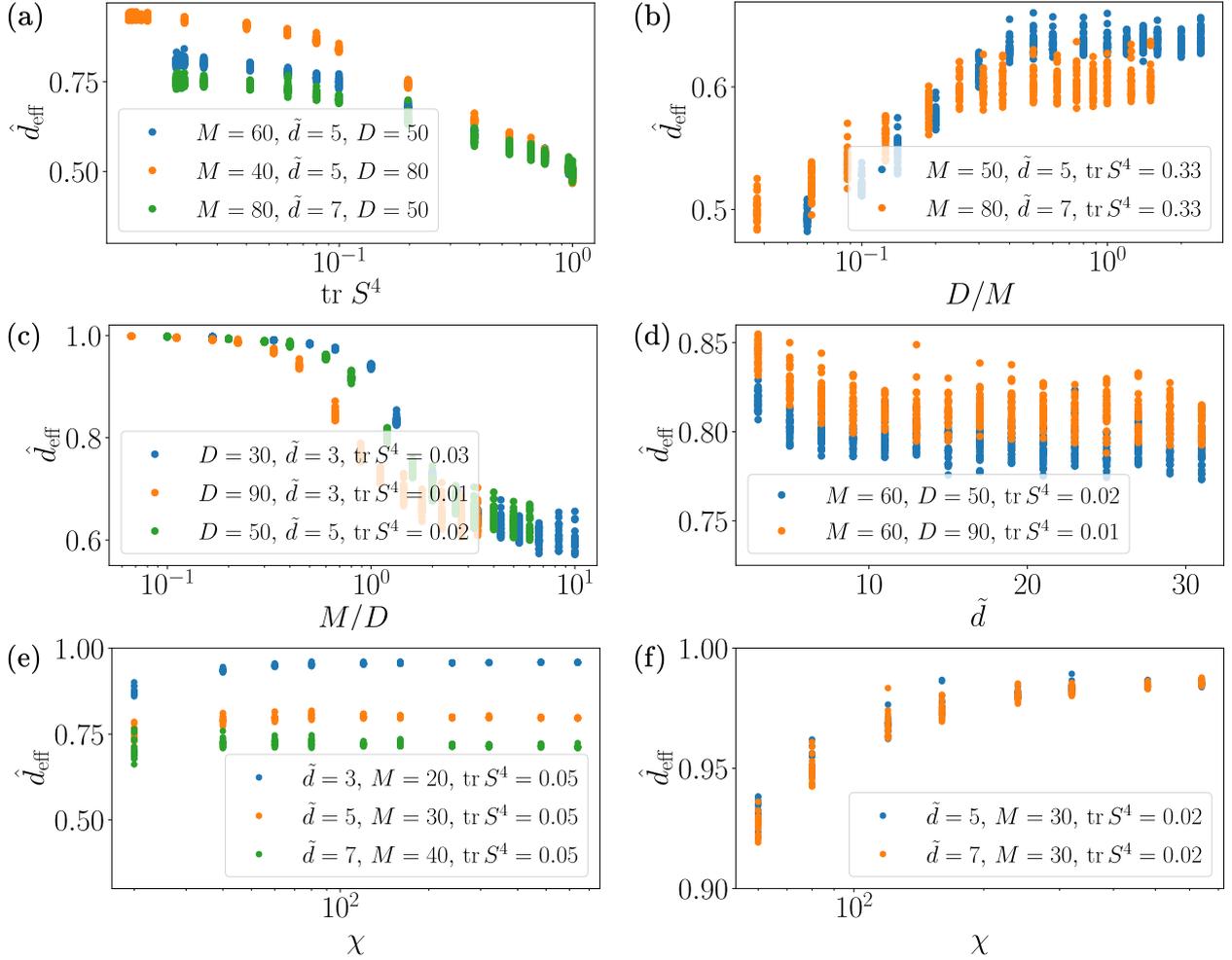
Figure S5: (a) Scaling of $\hat{d}_{\text{eff}}$ with the purity $\text{tr}(S^4)$ of the correlation spectrum ($\chi = 100$). (b) Scaling of $\hat{d}_{\text{eff}}$ with the ratio D/M. (c) Scaling of $\hat{d}_{\text{eff}}$ with the ratio M/D. (d) Scaling of $\hat{d}_{\text{eff}}$ with $\tilde{d}$. (e)-(f) Scaling of $\hat{d}_{\text{eff}}$ with the bond dimension χ. Each point corresponds to a random model realization, i.e., a random generation of V as a right-normalized tensor train. For every value on the x axis, 30 model realizations are drawn. The normalized ED is computed using 200 parameters samples for estimating the normalized FIM.

In Fig. S5 we show the dependence of the normalized ED on various model characteristics, in the case of tensorized structure constants, that is, with V decomposed as a tensor train. As expected, the ED monotonically decreases in expectation for increasing $\text{tr}(S^4)$ (see panel (a)), and in the overparameterized regime $M > D$ its value also strongly depends on D (see panels (b) and (c)), as expected from the bounds described in Section S4.2. The other model characteristics such as $\tilde{d}$ and the bond dimension χ have little influence on the ED (see panels (d), (e) and (f)), further confirming our expectations that $D > M$ the ED is mostly controlled by $\text{tr}(S^4)$.

S8 Further numerical results on training regression models

Here we show further numerical results on the interplay of ED and model bias and their effect on training regression models with gradient descent methods. The experiments performed here are analogous to those presented in the main text. we perform several training experiments with randomly drawn data-generating function $y(\boldsymbol{x})$ and structure constants Γ. For chosen dimensions D and K 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 (Eq. (106)) and cutoff models (Eq. (107)), in order to compare the training dynamics of models with higher and lower ED, respectively. For any given degree of bias we consider the minimum MSE attained during training as a proxy for the training quality, denoted with $\text{MSE}_{\min}^{\text{full}}$ and $\text{MSE}_{\min}^{\text{cut}}$ for full and cutoff models, respectively, and study the difference

$$\Delta_{\text{f}-\text{c}}\text{MSE}_{\min} = \text{MSE}_{\min}^{\text{full}} - \text{MSE}_{\min}^{\text{cut}} \,, \tag{117}$$

as a function of the difference in the ED between full and cutoff models, i.e., $\hat{d}_{\text{eff}}^{\text{full}} - \hat{d}_{\text{eff}}^{\text{cut}}$. A positive value of $\Delta_{\text{f}-\text{c}}\text{MSE}_{\min}$ 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 $\Delta_{\text{f}-\text{c}}\text{MSE}_{\min}$ implies a better performance of models with higher ED.

Overall, the additional result presented here confirm and corroborate the findings in the main text. The results for models with full (i.e., non-tensorized) structure constants are shown in Figs. S6, S7, S8 and S9. Qualitatively, the behavior of $\Delta_{\text{f}-\text{c}}\text{MSE}_{\min}$ as a function of the ED difference $\hat{d}_{\text{eff}}^{\text{full}} - \hat{d}_{\text{eff}}^{\text{cut}}$ is the same and consistent with what presented in the main text irrespectively of the model specification (see Figs. S6 and S7 for the case of $\tilde{d} = 2$, i.e., $\tilde{\mathcal{B}}_m = \{\cos\theta_m, \sin\theta_m\}$, and Figs. S8 and S9 for the case $\tilde{d} = 5$, i.e., $\tilde{\mathcal{B}}_m = \{1, \cos\theta_m, \cos 2\theta_m, \sin\theta_m, \sin 2\theta_m\}$).

The corresponding results for tensorized models (where only V is decomposed as a tensor train) are shown in Figs. S10 and S11 for $\tilde{d} = 5$, while Fig. S12 shows results for $\tilde{d} = 3$ and two input features $N = 2$.
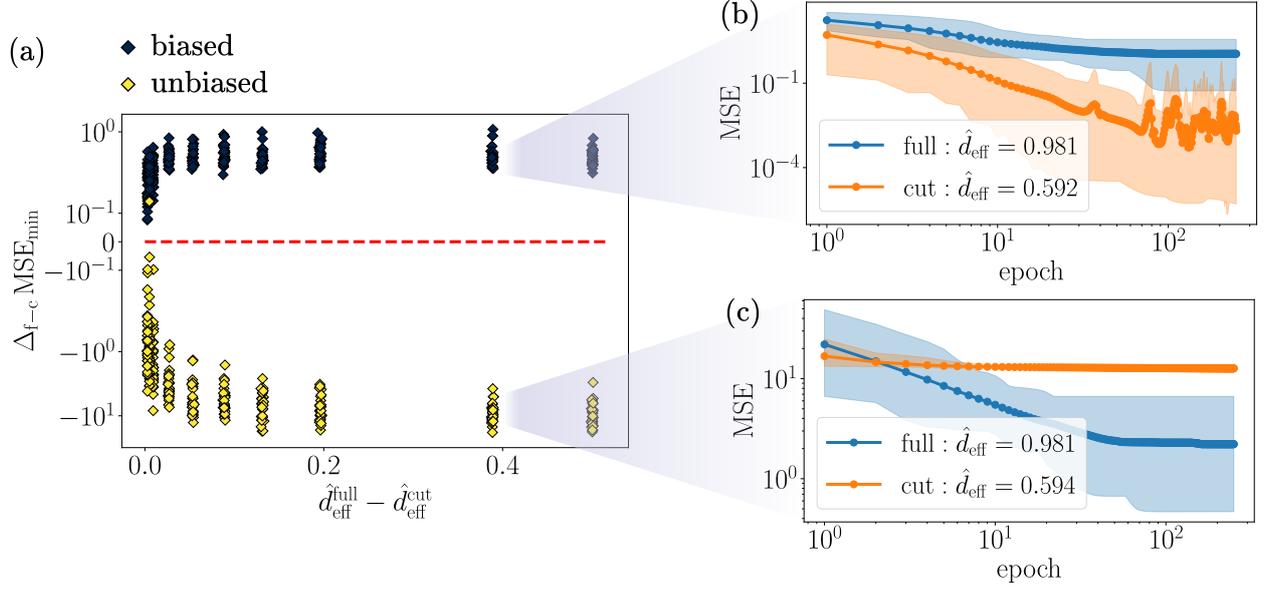
Figure S6: (a) $\Delta_{\text{f}-\text{c}}\text{MSE}_{\min}$ for different values of the difference $\hat{d}_{\text{eff}}^{\text{full}} - \hat{d}_{\text{eff}}^{\text{cut}}$. Each point corresponds to $\Delta_{\text{f}-\text{c}}\text{MSE}_{\min}$ averaged over 30 training instances, for a single random model realization. The red line serves as a guide for the eye for zero MSE difference. (b) Training curves for a random biased model realization. (c) Training curves for a random unbiased model realization. In (b)-(c), the full model is in blue and the cutoff model in orange, and the shading corresponds to the spread over 30 training instances. For these plots, $N = 1$, $\Omega = \{1, ..., 9\}$ $(d = 19)$, $\tilde{\mathcal{B}}_m = \{\cos\theta_m, \sin\theta_m\}$ $(\tilde{d} = 2)$, $M = 12$, $R = 4$, $\mathfrak{n}_{\text{train}} = 30$ with a batch size of 5.
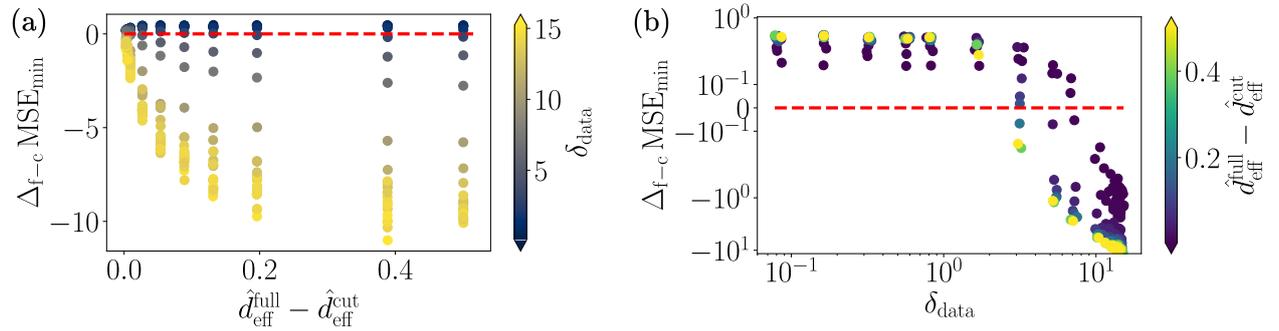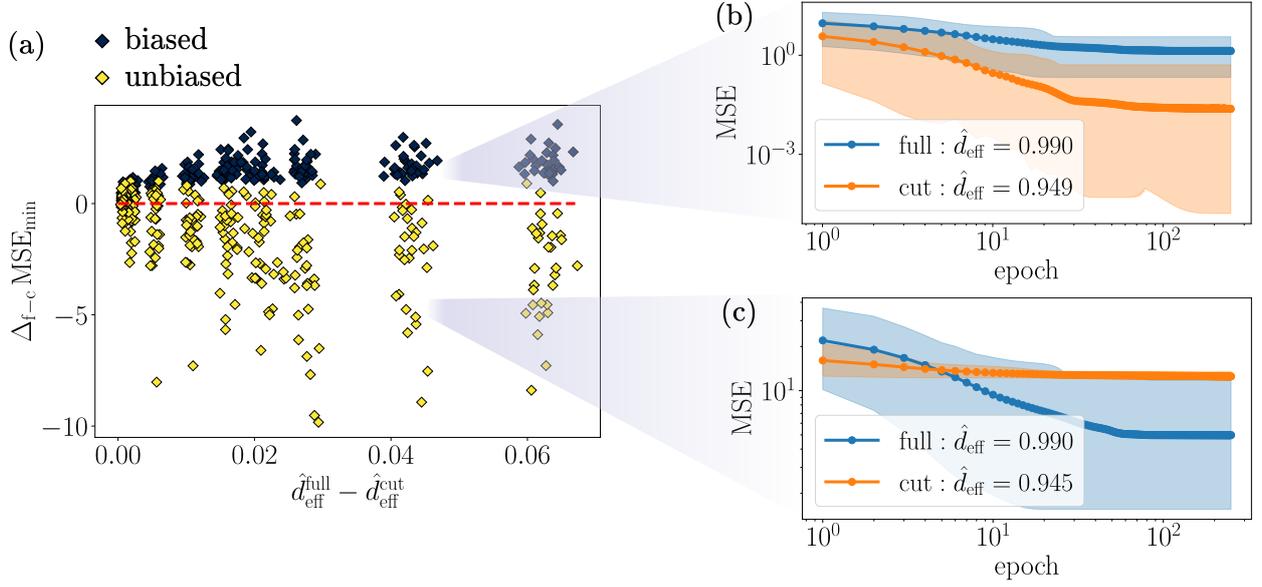

Figure S7: (a) $\Delta_{\text{f}-\text{c}}\text{MSE}_{\min}$ for different values of the difference $\hat{d}_{\text{eff}}^{\text{full}} - \hat{d}_{\text{eff}}^{\text{cut}}$. Each point corresponds to $\Delta_{\text{f}-\text{c}}\text{MSE}_{\min}$ averaged over 30 training instances for 30 random model 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, $N = 1$, $\Omega = \{1, ..., 9\}$ $(d = 19)$, $\tilde{\mathcal{B}}_m = \{\cos\theta_m, \sin\theta_m\}$ $(\tilde{d} = 2)$, $M = 12$, $R = 4$, $\mathfrak{n}_{\text{train}} = 30$ with a batch size of 5.

Figure S8: (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ for different values of the difference $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$. Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ averaged over 30 training instances, for a single random model realization. The red line serves as a guide for the eye for zero MSE difference. (b) Training curves for a random biased model realization. (c) Training curves for a random unbiased model realization. In (b)-(c), the full model is in blue and the cutoff model in orange, and the shading corresponds to the spread over 30 training instances. For these plots, $N = 1$, $\Omega = \{1,...,6\}$ ($d = 13$), $\tilde{\Omega} = \{1,2\}$ ($\tilde{d} = 5$), $M = 5$, $R = 4$, $\mathfrak{n}_{\mathrm{train}} = 25$ with a batch size of 5.
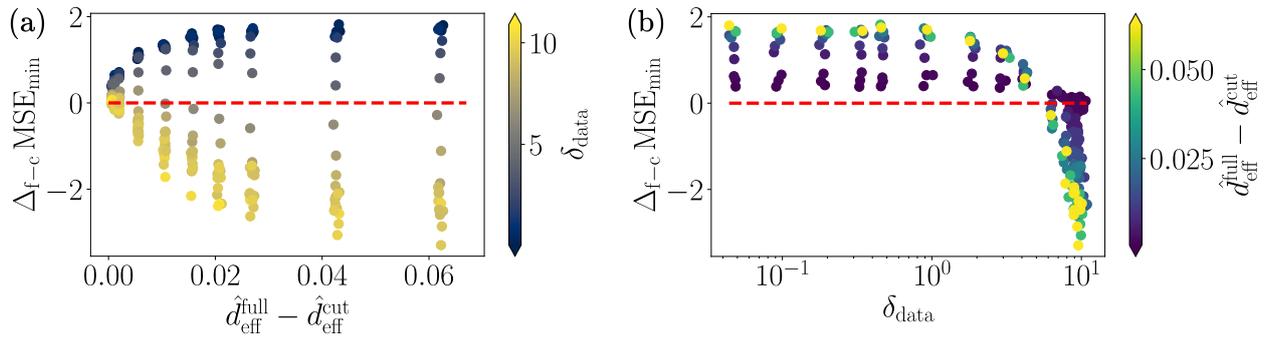

Figure S9: (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ for different values of the difference $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$. Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ averaged over 30 training instances for 30 random model 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, $N = 1$, $\Omega = \{1,...,6\}$ ($d = 13$), $\tilde{\Omega} = \{1,2\}$ ($\tilde{d} = 5$), $M = 5$, $R = 4$, $\mathfrak{n}_{\mathrm{train}} = 25$ with a batch size of 5.
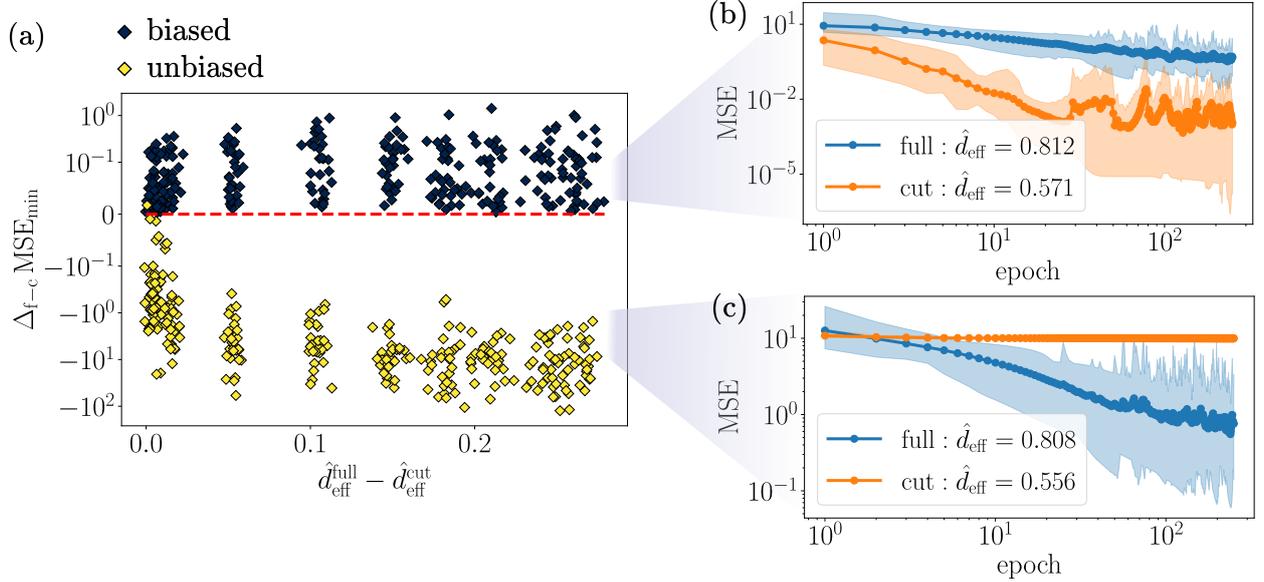
Figure S10: (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ for different values of the difference $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$. Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ averaged over 30 training instances, for a single random 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. (b) Training curves for a random biased model realization. (c) Training curves for a random unbiased model realization. In (b)-(c), the full model is in blue and the cutoff model in orange, and the shading corresponds to the spread over 30 training instances. For these plots, $N = 1$, $\Omega = \{1, ..., 13\}$ ($d = 27$), $\tilde{\Omega} = \{1, 2\}$ ($\tilde{d} = 5$), $M = 36$, $R = 4$, $\chi = 60$, $\mathfrak{n}_{\mathrm{train}} = 30$ with a batch size of 5.
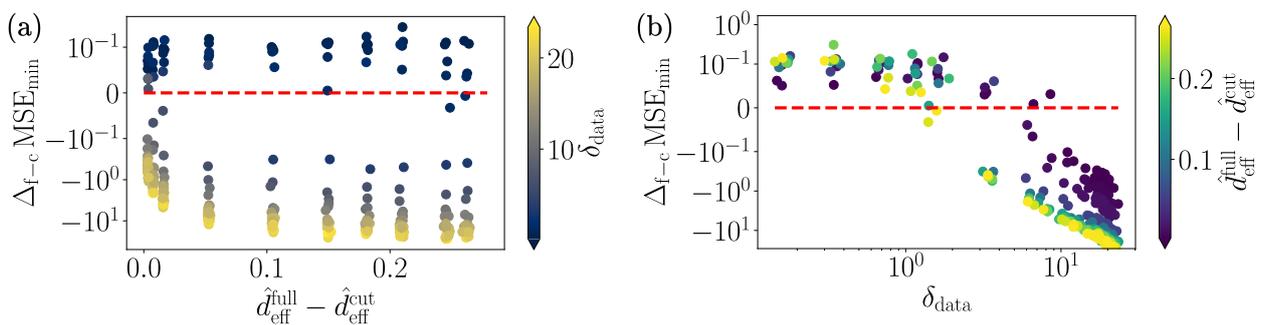

Figure S11: (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ for different values of the difference $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$. Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\min}$ averaged over 30 training instances for 30 random model realizations, i.e., random right-normalized tensor trains representing V. (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, $N = 1$, $\Omega = \{1, ..., 13\}$ ($d = 27$), $\tilde{\Omega} = \{1, 2\}$ ($\tilde{d} = 5$), $M = 36$, $R = 4$, $\chi = 60$, $\mathfrak{n}_{\mathrm{train}} = 30$ with a batch size of 5.
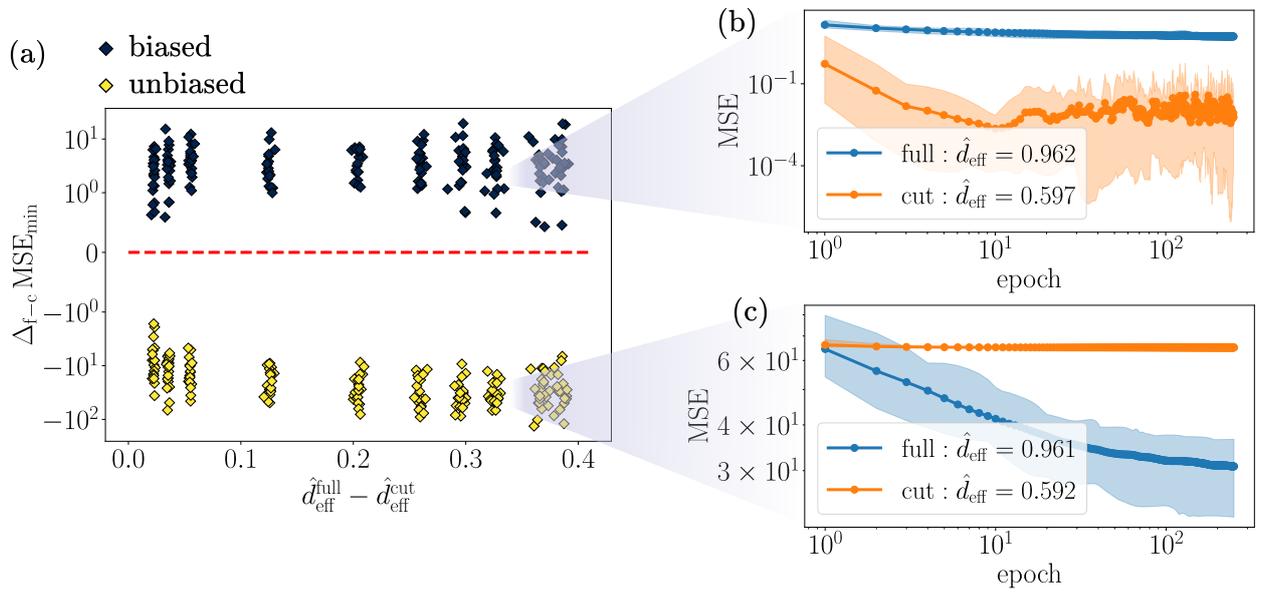
Figure S12: (a) $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ for different values of the difference $\hat{d}_{\mathrm{eff}}^{\mathrm{full}} - \hat{d}_{\mathrm{eff}}^{\mathrm{cut}}$. Each point corresponds to $\Delta_{\mathrm{f-c}}\mathrm{MSE}_{\mathrm{min}}$ averaged over 30 training instances, 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. (b) Training curves for a random biased model realization. (c) Training curves for a random unbiased model realization. In (b)-(c), the full model is in blue and the cutoff model in orange, and the shading corresponds to the spread over 30 training instances. For these plots, $N = 2$, $\Omega = \{1, ..., 5\}$ $(d = 11)$, $\tilde{\Omega} = \{1\}$ $(\tilde{d} = 3)$, $M = 32$, $R = 7$, $\chi = 100$, $\mathfrak{n}_{\mathrm{train}} = 225$ with a batch size of 5.