nature computational science

ARTICLES

https://doi.org/10.1038/s43588-021-00084-1

The power of quantum neural networks

Amira Abbas[1,2], David Sutter[1], Christa Zoufal[1,3], Aurelien Lucchi[3], Alessio Figalli[3] and Stefan Woerner[1]✉

It is unknown whether near-term quantum computers are advantageous for machine learning tasks. In this work we address this question by trying to understand how powerful and trainable quantum machine learning models are in relation to popular classical neural networks. We propose the effective dimension—a measure that captures these qualities—and prove that it can be used to assess any statistical model's ability to generalize on new data. Crucially, the effective dimension is a data-dependent measure that depends on the Fisher information, which allows us to gauge the ability of a model to train. We demonstrate numerically that a class of quantum neural networks is able to achieve a considerably better effective dimension than comparable feedforward networks and train faster, suggesting an advantage for quantum machine learning, which we verify on real quantum hardware.

The power of a model lies in its ability to fit a variety of functions[1]. In machine learning, power is often referred to as a model's capacity to express different relationships between variables[2]. Deep neural networks have proven to be extremely powerful models, capable of capturing intricate relationships by learning from data[3]. Quantum neural networks serve as a newer class of machine learning models that are deployed on quantum computers and use quantum effects such as superposition, entanglement and interference to perform computation. Some proposals for quantum neural networks include[4–11]—and hint at—potential advantages such as speed-ups in training and faster processing. Although there has been much development in the growing field of quantum machine learning, a systematic study of the trade-offs between quantum and classical models has yet to be conducted[12]. In particular, the question of whether quantum neural networks are more powerful than classical neural networks is still open.

A common way to quantify the power of a model is by its complexity[13]. In statistical learning theory, the Vapnik–Chervonenkis dimension is an established complexity measure, where error bounds on how well a model generalizes (that is, performs on unseen data) can be derived[14]. Although the Vapnik–Chervonenkis dimension has attractive properties in theory, computing it in practice is notoriously difficult. Furthermore, using the Vapnik–Chervonenkis dimension to bound generalization error requires several unrealistic assumptions, including that the model has access to infinite data[15,16]. The measure also scales with the number of parameters in the model and ignores the distribution of data. As modern deep neural networks are heavily overparameterized, generalization bounds based on the Vapnik–Chervonenkis dimension—and other measures alike—are typically vacuous[17,18].

In ref. [19], the authors analyzed the expressive power of parameterized quantum circuits using memory capacity and found that quantum neural networks had limited advantages over classical neural networks. Memory capacity is, however, closely related to the Vapnik–Chervonenkis dimension and is thus subject to similar criticisms. In ref. [20], a quantum neural network is presented that exhibits a higher expressibility than certain classical models, captured by the types of probability distributions it can generate. Another result from ref. [21] is based on strong heuristics and provides systematic examples of possible advantages for quantum neural networks.

We turn our attention to measures that are easy to estimate in practice and, importantly, incorporate the distribution of data. In particular, measures such as the effective dimension have been motivated from an information-theoretic standpoint and depend on the Fisher information, a quantity that describes the geometry of a model's parameter space and is essential in both statistics and machine learning[22–24]. We argue that the effective dimension is a robust capacity measure through proof of a generalization error bound and supporting numerical analyses, and thus use this measure to study the power of a popular class of neural networks in both classical and quantum regimes.

Despite a lack of quantitative statements on the power of quantum neural networks, another issue is rooted in the trainability of these models. A precise connection between expressibility and trainability for certain classes of quantum neural networks is outlined in refs. [25,26]. Quantum neural networks often suffer from the barren plateau phenomenon, wherein the loss landscape is perilously flat and parameter optimization is therefore extremely difficult[27]. As shown in ref. [28], barren plateaus may be noise induced, where certain noise models are assumed on the hardware. In other words, the effect of hardware noise can make it very difficult to train a quantum model. Furthermore, barren plateaus can be circuit induced, which relates to the design of a model and random parameter initialization. Methods to avoid the latter have been explored in refs. [29–32], but noise-induced barren plateaus remain problematic.

A particular attempt to understand the loss landscape of quantum models uses the Hessian[33], which quantifies the curvature of a model's loss function at a point in its parameter space[34]. Properties of the Hessian, such as its spectrum, provide useful diagnostic information on the trainability of a model[35]. It was discovered that the entries of the Hessian vanish exponentially in models suffering from a barren plateau[36]. For certain loss functions, the Fisher information matrix coincides with the Hessian of the loss function[37]. Consequently, we can examine the trainability of quantum and classical neural networks by analyzing the Fisher information matrix, which is incorporated by the effective dimension. In this way, we may explicitly relate the effective dimension to model trainability[38].

We find that a class of quantum neural networks is able to achieve a considerably higher capacity and faster training ability numerically than comparable classical feedforward neural networks. A higher capacity is captured by a higher effective dimension, whereas

[1]IBM Quantum, IBM Research—Zurich, Rueschlikon, Switzerland. [2]University of KwaZulu-Natal, Durban, South Africa. [3]ETH Zurich, Zurich, Switzerland. ✉e-mail: wor@zurich.ibm.com

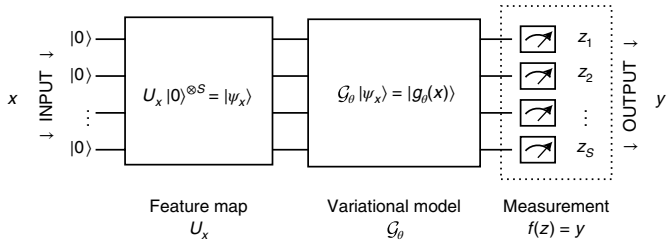

Fig. 1 | Overview of the quantum neural network used in this study. The input $x \in \mathbb{R}^{s_{in}}$ is encoded into an S-qubit Hilbert space by applying the feature map $|\psi_x\rangle := \mathcal{U}_x|0\rangle^{\otimes S}$. This state is then evolved via a variational form $|g_\theta(x)\rangle := \mathcal{G}_\theta|\psi_x\rangle$, where $\mathcal{G}$ is a parameterized unitary evolving the state after the feature map to a new state, and the parameters $\theta \in \Theta$ are chosen to minimize a certain loss function. Finally a measurement is performed whose outcome $z = (z_1, \ldots, z_S)$ is post-processed to extract the output of the model $y := f(z)$.

faster training implies that a model will reach a lower training error than another comparable model for a fixed number of training iterations. More generally, trainability is assessed by leveraging the information-theoretic properties of the Fisher information, which we connect to the barren plateau phenomenon. Our experiments reveal that how you encode data in a quantum neural network influences the likelihood of your model encountering a barren plateau. A quantum neural network with a data encoding strategy that is easy to simulate classically seems more likely to encounter a barren plateau, whereas a harder encoding strategy shows resilience to the phenomenon. Noise, however, remains problematic by inhibiting training in general.

Results

Quantum neural networks. Quantum neural networks are a subclass of variational quantum algorithms that comprise quantum circuits containing parameterized gate operations[39]. Information (usually in the form of classical data) is first encoded into a quantum state via a state-preparation routine called a quantum feature map[40]. The choice of feature map is geared towards enhancing the performance of the quantum neural network and is typically neither optimized nor trained, although this idea is discussed in ref. [41]. Once data are encoded into a quantum state, a model called a variational model is applied, which contains parameterized gate operations that are optimized for a particular task, analogous to classical machine learning techniques[5–7,42]. The final output of the quantum neural network is extracted from measurements made to the quantum circuit after the variational model is applied. These measurements are often converted to labels or predictions through classical post-processing before being passed to a loss function, where the idea is to choose parameters for the variational model that minimize the loss function.

The quantum models we use can be summarized in Fig. 1, with details of the structure and implementation in the Methods. We create two model variants: one which we call a quantum neural network and the other an easy quantum model.

The Fisher information. A way to assess the information gained by a particular parameterization of a statistical model is epitomized by the Fisher information. By defining a neural network as a statistical model, we can describe the joint relationship between data pairs (x,y) as $p(x,y;\theta) = p(y|x;\theta)p(x)$ for all $x \in \mathcal{X} \subset \mathbb{R}^{s_{in}}$, $y \in \mathcal{Y} \subset \mathbb{R}^{s_{out}}$ and $\theta \in \Theta \subset [-1,1]^d$ (where θ is a vectorized parameter set, Θ is the full parameter space and d is the number of trainable parameters). This is achieved by applying an appropriate post-processing function in both classical and quantum networks. In the classical network, we apply a softmax function to the final layer, whereas in the quantum network we obtain probabilities based on the parity of the output bit strings. The input distribution $p(x)$ is a prior distribution, whereas the conditional distribution $p(y|x;\theta)$ describes the input–output relation of the model for a fixed $\theta \in \Theta$; Θ forms a Riemannian space, which gives rise to a Riemannian metric, namely, the Fisher information matrix

$$F(\theta) = \mathbb{E}_{(x,y)\sim p}\left[\frac{\partial}{\partial\theta}\log p(x,y;\theta)\frac{\partial}{\partial\theta}\log p(x,y;\theta)^{\mathrm{T}}\right] \in \mathbb{R}^{d\times d},$$

which can be approximated by the empirical Fisher information matrix

$$\tilde{F}_k(\theta) = \frac{1}{k}\sum_{j=1}^{k}\frac{\partial}{\partial\theta}\log p(x_j,y_j;\theta)\frac{\partial}{\partial\theta}\log p(x_j,y_j;\theta)^{\mathrm{T}}, \quad (1)$$

where $(x_j, y_j)_{j=1}^k$ are independent and identically distributed, drawn from the distribution $p(x,y;\theta)$ (ref. [37]). By definition, the Fisher information matrix is positive semidefinite and hence its eigenvalues are non-negative, real numbers.

The Fisher information conveniently helps capture the sensitivity of a neural network's output relative to movements in the parameter space[43]. In ref. [44], the authors leverage geometric invariances associated with the Fisher information to produce the Fisher–Rao norm, a robust norm-based capacity measure defined as the quadratic form $||\theta||_{\mathrm{fr}}^2 := \theta^{\mathrm{T}}F(\theta)\theta$ for θ. Notably, the Fisher–Rao norm acts as an umbrella for several other existing norm-based measures[45–47] and has demonstrated desirable properties both theoretically and empirically.

The effective dimension. The effective dimension is a complexity measure motivated by information geometry, with useful qualities. The goal of the effective dimension is to estimate the size that a model occupies in model space—the space of all possible functions for a particular model class, where the Fisher information matrix serves as the metric. Although there are many ways to define the effective dimension, a useful definition is presented in ref. [22], which is designed to be operationally meaningful in settings where data are limited. More precisely, the number of data observations determines a natural scale or resolution used to observe model space. This is beneficial in practice where data are often scarce and can help in understanding how data availability influences the accurate capture of model complexity.

The effective dimension is motivated by the theory of minimum description length, which is a model selection principle favoring models with the shortest description of the given data. Based on this principle, it can be shown that the complexity at size n of a model is given by

$$\frac{d}{2}\log\frac{n}{2\pi} + \log\left(\int_\Theta\sqrt{\det F(\theta)}\,\mathrm{d}\theta\right) + o(1),$$

where $o(1)$ vanishes as $n \to \infty$ (ref. [48]). The first term containing d is usually interpreted as the dimension of the model, whereas the second term is known as the geometric complexity. Information geometric manipulations allow us to combine both terms into a single expression, referred to as the effective dimension[22].

Definition 1. The effective dimension of a statistical model $\mathcal{M}_\Theta := \{p(\cdot,\cdot;\theta) : \theta \in \Theta\}$ with respect to $\gamma \in (0,1]$, a d-dimensional parameter space $\Theta \subset \mathbb{R}^d$ and $n \in \mathbb{N}$, $n > 1$ data samples is defined as

$$d_{\gamma,n}(\mathcal{M}_\Theta) := 2\frac{\log\left(\frac{1}{V_\Theta}\int_\Theta\sqrt{\det\left(\mathrm{id}_d + \frac{\gamma n}{2\pi\log n}\hat{F}(\theta)\right)}\,\mathrm{d}\theta\right)}{\log\left(\frac{\gamma n}{2\pi\log n}\right)}, \quad (2)$$

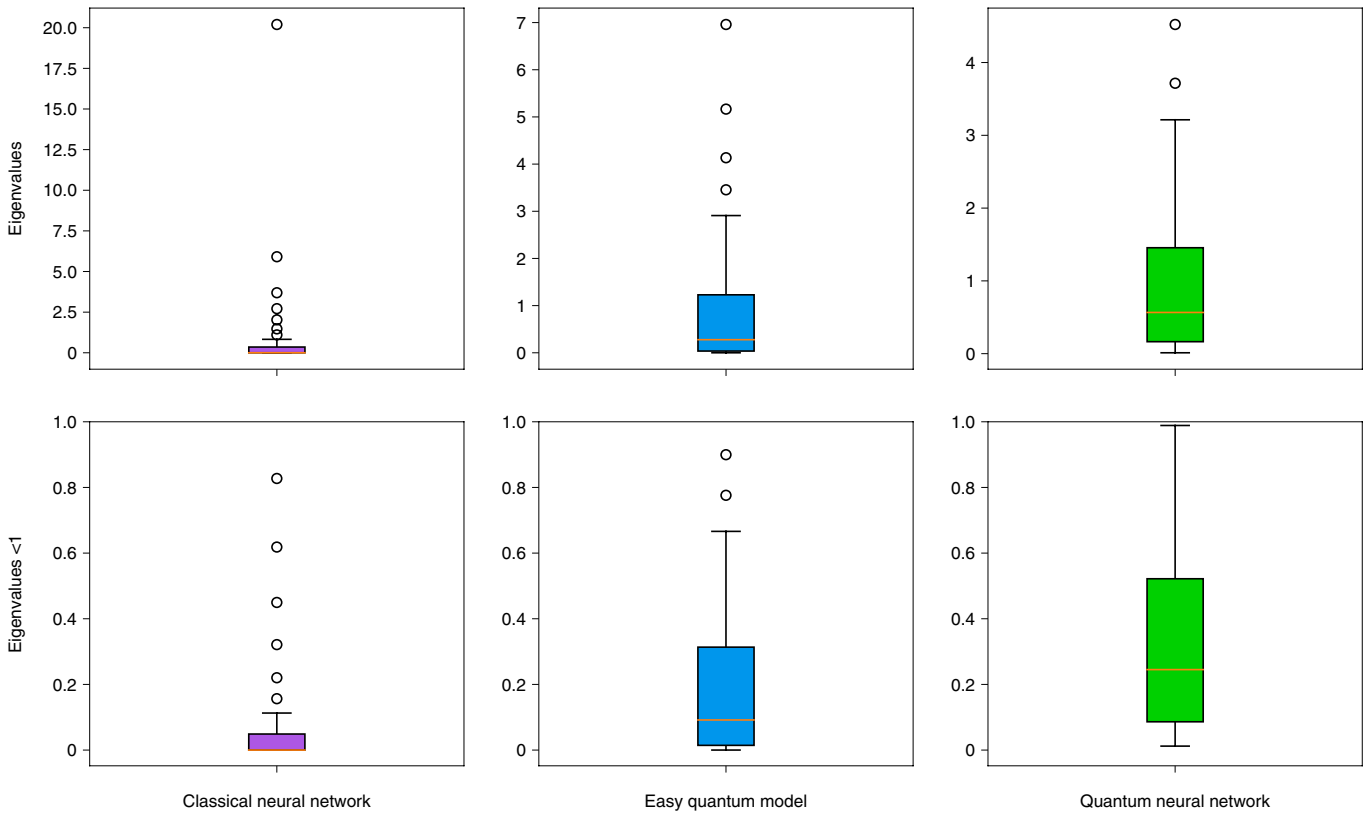

Fig. 2 | Average Fisher information spectrum distribution. Here, box plots are used to reveal the average distribution of eigenvalues of the Fisher information matrix for the classical feedforward neural network and the quantum neural network with two different feature maps. The dots in the box plots represent outlier values relative to the length of the whiskers. The lower whiskers are at the lowest data points above Q1 − 1.5 × (Q3 − Q1), whereas the upper whiskers are at the highest data points below Q3 + 1.5 × (Q3 − Q1), where Q1 and Q3 are the first and third quartiles, respectively. This is a standard method to compute these plots. The easy quantum model has a classically simulatable data encoding strategy, whereas the quantum neural network's encoding scheme is conjectured to be difficult. In each model, we compute the Fisher information matrix 100 times using parameters sampled uniformly at random and plot the resulting average distribution of the eigenvalues. We fix $d = 40$, input size at $s_{in} = 4$ and output size at $s_{out} = 2$. The top row contains the average distribution of all eigenvalues for each model, whereas the bottom row contains the average distribution of eigenvalues less than 1 for each model.

where $V_\Theta := \int_\Theta \mathrm{d}\theta \in \mathbb{R}_+$ is the volume of the parameter space. The matrix $\hat{F}(\theta) \in \mathbb{R}^{d\times d}$ is the normalized Fisher information matrix defined as

$$\hat{F}_{ij}(\theta) := d\frac{V_\Theta}{\int_\Theta \mathrm{tr}(F(\theta))\mathrm{d}\theta}F_{ij}(\theta)\,.$$

Remark 1 (properties of the effective dimension). In the limit $n \to \infty$, the effective dimension converges to the maximal rank $\bar{r} := \max_{\theta\in\Theta} r_\theta$, where $r_\theta \leq d$ denotes the rank of the Fisher information matrix $F(\theta)$. The proof of this result can be seen in Supplementary Section 2.1, but it is worthwhile to note that the effective dimension does not necessarily increase monotonically with n, as explained in Supplementary Section 2.2. The geometric operational meaning of the effective dimension only holds if n is sufficiently large. We conduct experiments over a wide range of n and ensure that conclusions are drawn from results where the choice of n is sufficient.

Another noteworthy point is that the effective dimension is easy to estimate. To see this, recall that we need to first estimate $F(\theta)$ and, second, calculate the integral over Θ given in equation (2). Both of these steps can be achieved via Monte Carlo integration which, in practice, does not depend on the model's dimension.

There are also two minor differences between equation (2) and the effective dimension from ref. [22]: the presence of the constant $\gamma \in (0,1]$, and the $\log n$ term. These modifications are helpful in proving a generalization bound such that the effective dimension may be interpreted as a bounded capacity measure, serving as a useful tool to analyze the power of statistical models. We demonstrate this in the Methods.

The Fisher information spectrum. Classically, the Fisher information spectrum reveals a lot about the optimization landscape of a model. The magnitude of the eigenvalues illustrates the curvature of a model for a particular parameterization. If there is a large concentration of eigenvalues near zero, the optimization landscape will be predominantly flat and parameters become difficult to train with gradient-based methods[38]. On the quantum side, we show in Supplementary Section 4 that if a model is in a barren plateau, the Fisher information spectrum will be concentrated around zero and training also becomes unfeasible. We can thus make connections to trainability via the spectrum of the Fisher information matrix by using the effective dimension. Looking closely at equation (2), we see that the effective dimension converges to its maximum fastest if the Fisher information spectrum is evenly distributed, on average.

We analyze the Fisher information spectra for the quantum neural network, the easy quantum model, and all possible configurations of the fully connected feedforward neural network—where all models share a specified triple (d, s_{in}, s_{out}). To be robust, we sample 100 sets of parameters uniformly on $\Theta = [-1,1]^d$ and compute the Fisher information matrix 100 times using data sampled from a standard Gaussian distribution. The resulting average distributions

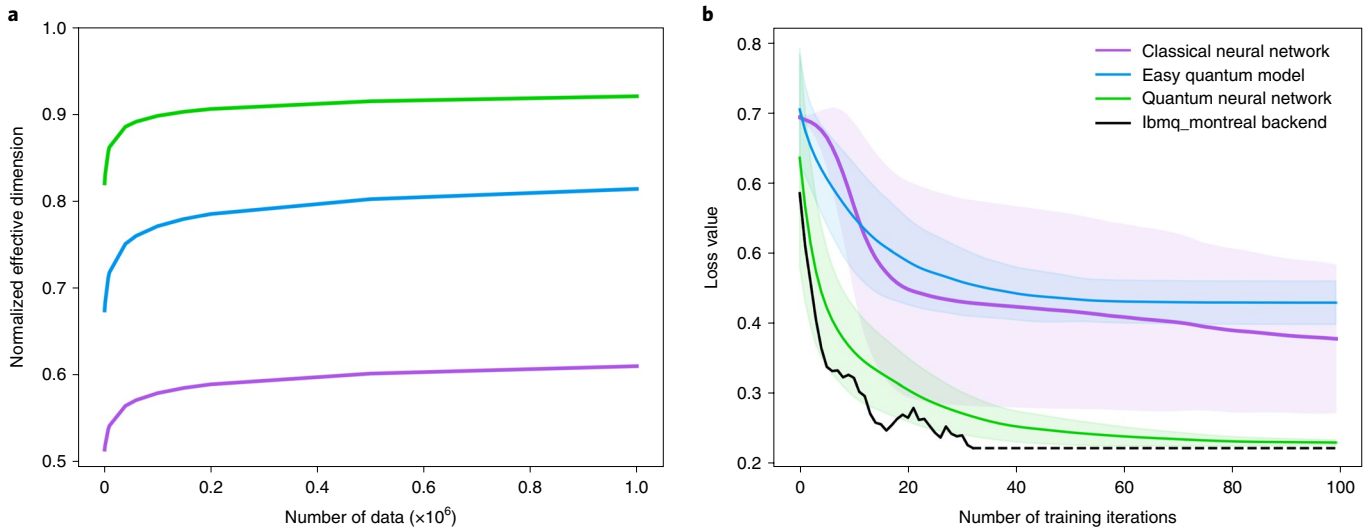

Fig. 3 | Normalized effective dimension and training loss. a, The normalized effective dimension plotted for the quantum neural network (green), the easy quantum model (blue) and the classical feedforward neural network (purple). We fix $s_{in}=4$, $s_{out}=2$ and $d=40$. **b**, Using the first two classes of the Iris dataset[55], we train all three models at $d=8$, with a full batch size. The ADAM optimizer, with an initial learning rate of 0.1, is selected. For a fixed number of training iterations (100), we train all models over 100 trials and plot the average training loss along with ±1 s.d. We further verify the performance of the quantum neural network on real quantum hardware and train the model using the ibmq_montreal 27-qubit device. We plot the hardware results until they stabilize, at roughly 33 training iterations, thereafter we stop training and denote this final loss value with a dashed line. The actual hardware implementation contains less CNOT gates by using linear connectivity for the feature map and variational circuit instead of all-to-all connectivity to cope with limited resources, leading to the lower loss values.

of the eigenvalues of these 100 Fisher information matrices are plotted in the top row of Fig. 2 for $d=40$, $s_{in}=4$ and $s_{out}=2$. A sensitivity analysis is included in Supplementary Section 3.1 to verify that 100 parameter samples are reasonable for the models we consider. In higher dimensions, this number will need to increase. The bottom row of Fig. 2 contains the distribution for eigenvalues less than 1.

The classical model depicted in Fig. 2 is the one with the highest average rank of Fisher information matrices. The majority of eigenvalues are negligible (of the order 10^{-14}), with a few very large values. This behavior is observed across all classical configurations that we consider and is consistent with results from literature, where the Fisher information matrix of non-linear feedforward neural networks is known to be highly degenerate, with a few large eigenvalues[38]. The concentration around zero becomes more evident in the bottom row of the plot, which depicts the eigenvalue distribution of just the eigenvalues less than 1.

The easy quantum model also has most of its eigenvalues close to zero, and although there are some large eigenvalues, their magnitudes are not as extreme as the classical model.

The quantum neural network, on the other hand, has a distribution of eigenvalues that is more uniform, with no outlying values. This can be seen from the range of the eigenvalues on the y-axis in Fig. 2. This distribution remains more or less constant as the number of qubits increase, even in the presence of hardware noise (see Supplementary Section 3.2); this has implications for capacity and trainability, which we examine next.

Capacity analysis. In Fig. 3a, we plot the normalized effective dimension for all three model types. The normalization ensures that the effective dimension lies between 0 and 1 by simply dividing by d. The convergence speed of the effective dimension to its maximum is slowed down by smaller eigenvalues and uneven Fisher information spectra. As the classical models contain highly degenerate Fisher matrices, the effective dimension converges the slowest, followed by the easy quantum model. The quantum neural network has non-degenerate Fisher information matrices and more even spectra, it therefore consistently achieves the highest effective dimension over all ranges of finite data considered. Intuitively, we would expect the additional effects of quantum operations such as entanglement and superposition—if used effectively—to generate models with higher capacity. The quantum neural network with a strong feature map is thus expected to deliver the highest capacity, but recall that in the limit $n\to\infty$, all models will converge to an effective dimension equal to the maximum rank of the Fisher information matrix (see *Remark 1*).

To support these observations, we calculate the capacity of each model using a different measure, the Fisher–Rao norm[44]. The average Fisher–Rao norm after training each model 100 times is roughly 250% higher in the quantum neural network than in the classical neural network, with the easy quantum model inbetween (see Supplementary Section 3.3).

Trainability. The observed Fisher information spectrum of the feedforward model is known to have undesirable optimization properties, where the outlying eigenvalues slow down training and loss convergence[35]. These large eigenvalues become even more pronounced in bigger models, as seen in Supplementary Fig. 5. On examining the easy quantum model over an increasing system size, the average Fisher spectrum becomes more concentrated around zero. This is characteristic of models encountering a barren plateau, presenting another unfavorable scenario for optimization. The quantum neural network, however, maintains its more even distribution of eigenvalues as the number of qubits and trainable parameters increase. Furthermore, a large amount of the eigenvalues are not near zero. This highlights the importance of a feature map in a quantum model. The harder data encoding strategy used in the quantum neural network seems to structurally change the optimization landscape and remove the flatness, usually associated with suboptimal optimization conditions such as barren plateaus.

We confirm the training statements for all three models with an experiment illustrated in Fig. 3b. Using a cross-entropy loss function, optimized with ADAM for a fixed number of training

iterations (100) and an initial learning rate of 0.1, the quantum neural network trains to a lower loss, faster than the other two models over an average of 100 trials. To support the promising training performance of the quantum neural network, we also train it once on real hardware using the ibmq_montreal 27-qubit device. We reduce the number of controlled NOT (CNOT) gates by only considering linear entanglement instead of all-to-all entanglement in the feature map and variational circuit. This is to cope with hardware limitations and could be the reason the hardware training performs even better than the simulated results, as too much entanglement has been shown to have negative effects on model trainability[49]. The full details of the experiment are contained in Supplementary Section 3.4. We find that the quantum neural network tangibly demonstrates faster training; however, the addition of hardware noise may still make training difficult, regardless of the optimization landscape (see Supplementary Section 3.2).

Discussion

In stark contrast to classical models, understanding the capacity of quantum neural networks is not well explored. Moreover, classical neural networks are known to produce highly degenerate Fisher information matrices, which can considerably slow down training. No such analysis has been performed for quantum neural networks.

This work attempts to address this gap but leaves room for further research. The feature map in a quantum model plays a large role in determining both its capacity and trainability via the effective dimension and Fisher information spectrum. A deeper investigation needs to be conducted on why the particular higher-order feature map used in this study produces a desirable model landscape that induces both a high capacity and faster training ability. Different variational circuits could also influence the model's landscape and the effects of non-unitary operations (for example, induced through intermediate measurements) should be investigated. The Fisher information spectra of certain quantum models seem robust against hardware noise, but trainability remains problematic and the possibility of noise-induced barren plateaus needs examination. Finally, understanding generalization performance on multiple datasets and larger models with complexities that we would be interested in practice, might prove insightful.

Overall, we have shown that quantum neural networks can possess a desirable Fisher information spectrum that enables them to train faster and express more functions than comparable classical and quantum models—a promising reveal for quantum machine learning, which we hope leads to further studies on the power of quantum models.

Methods

Quantum models used in this study. The quantum models used in this study first encode classical data $x \in \mathbb{R}^{s_{\text{in}}}$ into an S-qubit Hilbert space using a feature map, $\mathcal{U}_x$. For the quantum neural network, we use a feature map originally proposed in ref. [50], and in the easy quantum model we swap out this feature map for one that is easy to simulate classically. Supplementary Fig. 1 contains a circuit representation of the feature map from ref. [50], which we refer to as the hard feature map. Here the number of qubits in the model is chosen to equal the number of feature values of the data (that is, $S := s_{\text{in}}$). That way, we can associate the same index for each qubit, with each feature value of a data point; for example, if we have data that has three feature values (that is, $\mathbf{x} = (x_1, x_2, x_3)^{\mathrm{T}}$), we will have a three qubit model with qubits $= (q_1, q_2, q_3)$.

The operations in the hard feature map first apply Hadamard gates to each of the qubits, followed by a layer of RZ gates, whereby the angle of the Z rotation on qubit i depends on the ith feature of the data point $\mathbf{x}$, normalized between $[-1,1]$. RZZ gates are then applied to every pair of qubits. This time, the value of the controlled Z rotations depends on a product of feature values. For example, if the RZZ gate is controlled by qubit i and targets qubit j, then the angle of the controlled rotation applied to qubit j is dependent on the product of feature values $x_i x_j$. The RZZ gates are implemented using a decomposition into two CNOT gates and one RZ gate; thereafter, the RZ and RZZ gates are repeated once. The classically simulatable feature map employed in the easy quantum model is simply the first sets of Hadamard and RZ gates with no entanglement between any qubits, as performed at the beginning of Supplementary Fig. 1. These operations are not repeated.

After the feature map circuit is applied, we apply another set of operations that depend on traiable parameters. We call this the variational circuit, $\mathcal{G}_\theta$. Supplementary Fig. 2 depicts the variational form deployed in both the easy quantum model and the quantum neural network. The circuit consists of S qubits, to which parameterized RY gates are applied to every qubit. CNOT gates are thereafter applied between each pair of qubits in the circuit. Finally, another set of parameterized RY gates are applied to every qubit. This circuit has, by definition, $2S$ parameters. If the depth is increased, the entangling layers and second set of parameterized RY gates are repeated; d can be calculated as $d = (D+1)S$, where S is equal to s_{in} due to the choice of both feature maps used in this study and D is called the depth of the circuit (that is, how many times the entanglement and second set of RY operations are repeated).

We finally measure all qubits in the σ_z basis and classically compute the parity of the output bit strings. For simplicity, we consider binary classification, where the probability of observing class 0 corresponds to the probability of seeing even parity bit strings and similarly, for class 1 with odd parity bit strings.

The are two reasons for the choice of these models's architecture: first, the hard feature map is motivated in ref. [50] to serve as a useful data embedding strategy that is believed to be difficult to simulate classically as the depth and width increase, and the easy feature map allows us to benchmark this; second, the variational design aims to create more expressive models for quantum algorithms[51]. We benchmark the quantum models against a class of classical models that forms part of the foundation of deep learning, namely, feedforward neural networks. We consider all possible topologies with full connectivity for a fixed number of trainable parameters. Networks with and without biases and different activation functions are explored.

Generalization error bounds for the effective dimension. Suppose we are given a hypothesis class, $\mathcal{H}$, of functions mapping from $\mathcal{X}$ to $\mathcal{Y}$ and a training set $\mathcal{S}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$, where the pairs (x_i, y_i) are drawn independent and identically distributed from some unknown joint distribution, p. Furthermore, let $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. The challenge is to find a particular hypothesis $h \in \mathcal{H}$ with the smallest possible expected risk, defined as $R(h) := \mathbb{E}_{(x,y)\sim p}[L(h(x), y)]$. As we only have access to a training set $\mathcal{S}_n$, a good strategy to find the best hypothesis $h \in \mathcal{H}$ is to minimize the so called empirical risk, defined as $R_n(h) := \frac{1}{n}\sum_{i=1}^{n} L(h(x_i), y_i)$. The difference between the expected and the empirical risk is the generalization error—an important quantity in machine learning that dictates whether a hypothesis $h \in \mathcal{H}$ learned on a training set will perform well on unseen data, drawn from the unknown joint distribution p (ref. [17]). Therefore, an upper bound on the quantity

$$\sup_{h\in\mathcal{H}} |R(h) - R_n(h)|\,, \tag{3}$$

which vanishes as n grows large, is of considerable interest. Capacity measures help quantify the expressiveness and power of $\mathcal{H}$. The generalization error in equation (3) is thus typically bounded by an expression that depends on a capacity measure, such as the Vapnik–Chervonenkis dimension[3] or the Fisher–Rao norm[44]. *Theorem 1* provides a bound based on the effective dimension, which we use to study the power of neural networks from hereon.

In this manuscript we consider neural networks as models described by stochastic maps, parameterized by some $\theta \in \Theta$. As a result, the variables h and $\mathcal{H}$ are replaced by θ and Θ, respectively. The corresponding loss functions are mappings $L : \mathrm{P}(\mathcal{Y}) \times \mathrm{P}(\mathcal{Y}) \to \mathbb{R}$, where $\mathrm{P}(\mathcal{Y})$ denotes the set of distributions on $\mathcal{Y}$. We assume the following regularity assumption on the model $\mathcal{M}_\Theta := \{p(\cdot, \cdot; \theta) : \theta \in \Theta\}$:

$$\Theta \ni \theta \mapsto p(\cdot, \cdot; \theta) \text{ is } M_1\text{-Lipschitz continuous w.r.t. the supremum norm.} \tag{4}$$

Theorem 1 (generalization bound for the effective dimension). Let $\Theta = [-1,1]^d$ and consider a statistical model $\mathcal{M}_\Theta := \{p(\cdot, \cdot; \theta) : \theta \in \Theta\}$ that satisfies equation (4) such that $\hat{F}(\theta)$ has full rank for all $\theta \in \Theta$, and $||\nabla_\theta \log \hat{F}(\theta)|| \leq \Lambda$ for some $\Lambda \geq 0$ and all $\theta \in \Theta$. Let $d_{\gamma,n}$ denote the effective dimension of $\mathcal{M}_\Theta$ as defined in equation (2). Furthermore, let $L : \mathrm{P}(\mathcal{Y}) \times \mathrm{P}(\mathcal{Y}) \to [-B/2, B/2]$ for $B > 0$ be a loss function that is α-Hölder continuous with constant M_2 in the first argument with regards to the total variation distance for some $\alpha \in (0,1]$. Then there exists a constant $c_{d,\Lambda}$ such that for $\gamma \in (0,1]$ and all $n \in \mathbb{N}$, we have

$$\begin{aligned} &\mathbb{P}\left(\sup_{\theta\in\Theta} |R(\theta) - R_n(\theta)| \geq 4M\sqrt{\tfrac{2\pi \log n}{\gamma n}}\right) \\ &\leq c_{d,\Lambda}\left(\tfrac{\gamma n^{1/\alpha}}{2\pi \log n^{1/\alpha}}\right)^{\frac{d_{\gamma,n}^{1/\alpha}}{2}} \exp\left(-\tfrac{16M^2\pi \log n}{B^2\gamma}\right), \end{aligned} \tag{5}$$

where $M = M_1^{\alpha} M_2$.

The proof is given in Supplementary Section 5.1. Note that the choice of the norm to bound the gradient of the Fisher information matrix is irrelevant due to the presence of the dimensional constant $c_{d,\Lambda}$. In the special case where the Fisher information matrix does not depend on θ, we have $\Lambda = 0$ and (5)

holds for $c_{d,0} = 2\sqrt{d}$. This may occur in scenarios where a neural network is already trained, that is, the parameters $\theta \in \Theta$ are fixed. If we choose $\gamma \in (0,1]$ to be sufficiently small, we can ensure that the right-hand side of equation (5) vanishes in the limit $n \to \infty$. This is explained in Supplementary Section 5. To verify the ability of the effective dimension to capture generalization behavior, we conduct a numerical analysis similar to work presented in ref. [52]. We find that the effective dimension for a model trained on confusion sets with increasing label corruption, accurately captures generalization behavior. The details can be found in Supplementary Section 5.2.

The continuity assumptions of *Theorem 1* are satisfied for a large class of classical and quantum statistical models[53,54], as well as many popular loss functions. The full rank assumption on the Fisher information matrix, however, often does not hold in classical models. Non-linear feedforward neural networks, which we consider in this study, have particularly degenerate Fisher information matrices[38]. We thus further extend the generalization bound to account for a broad range of models that may not have a full rank Fisher information matrix.

Remark 2. (Relaxing the rank constraint in *Theorem 1*) The generalization bound in equation (5) can be modified to hold for a statistical model without a full rank Fisher information matrix. By partitioning Θ, we discretize the statistical model and prove a generalization bound for the discretized version of $\mathcal{M}_\Theta := \{p(\cdot,\cdot;\theta) : \theta \in \Theta\}$ denoted by $\mathcal{M}_\Theta^{(\kappa)} := \{p^{(\kappa)}(\cdot,\cdot;\theta) : \theta \in \Theta\}$, where $\kappa \in \mathbb{N}$ is a discretization parameter. By choosing κ carefully, we can control the discretization error. We then proceed similarly as in the proof of *Theorem 1*, that is, first connecting the generalization error to the covering number and then relating the covering number to the effective dimension. This is explained in detail, along with the proof, in Supplementary Section 5.3.

Training the quantum neural network on real hardware. The hardware experiment is conducted on the ibmq_montreal 27-qubit device. We use four qubits with linear connectivity to train the quantum neural network on the first two classes of the Iris dataset[55]. We deploy the same training specifications as in Supplementary Section 3.3 and randomly initialize the parameters. Once the training loss stabilizes, that is the change in the loss from one iteration to the next is small, we stop the hardware training. This occurs after roughly 33 training steps. The results are contained in Fig. 3b and the real hardware shows remarkable performance relative to all other models. Due to limited hardware availability, this experiment is only run once and an analysis of the hardware noise and the spread of the training loss for differently sampled initial parameters would make these results more robust.

We plot the circuit that is implemented on the quantum device in Supplementary Fig. 8. As in the quantum neural network discussed in Supplementary Section 1, the circuit contains parameterized RZ and RZZ rotations that depend on the data, as well as parameterized RY gates with eight trainable parameters. Note the different entanglement structure presented here as opposed to the circuits in Supplementary Figs. 1 and 2. This is to reduce the number of CNOT gates required to incorporate current hardware constraints and could be the reason the actual hardware implementation trains so well as too much entanglement has been shown to have a negative effect on model trainability[49]. The full circuit repeats the feature map encoding once before the variational form is applied.

Data availability

The data for the graphs and analyses in this study was generated using Python. Source data are provided with this paper. All other data can be accessed via the following Zenodo repository: https://doi.org/10.5281/zenodo.4732830 (ref. [56]).

Code availability

All code to generate the data, figures and analyses in this study is publicly available with detailed information on the implementation via the following Zenodo repository: https://doi.org/10.5281/zenodo.4732830 (ref. [56]).

Received: 20 November 2020; Accepted: 14 May 2021;
Published online: 24 June 2021

References

1. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016); http://www.deeplearningbook.org
2. Baldi, P. & Vershynin, R. The capacity of feedforward neural networks. *Neural Networks* **116**, 288–311 (2019).
3. Dziugaite, G. K. & Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proc. 33rd Conference on Uncertainty in Artificial Intelligence* (UAI, 2017).
4. Schuld, M. *Supervised Learning with Quantum Computers* (Springer, 2018).
5. Zoufal, C., Lucchi, A. & Woerner, S. Quantum generative adversarial networks for learning and loading random distributions. *npj Quant. Inf.* **5**, 1–9 (2019).
6. Romero, J., Olson, J. P. & Aspuru-Guzik, A. Quantum autoencoders for efficient compression of quantum data. *Quant. Sci. Technol.* **2**, 045001 (2017).
7. Dunjko, V. & Briegel, H. J. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Rep. Prog. Phys.* **81**, 074001 (2018).
8. Ciliberto, C. et al. Quantum machine learning: a classical perspective. *Proc. Roy. Soc. A* **474**, 20170551 (2018).
9. Killoran, N. et al. Continuous-variable quantum neural networks. *Phys. Rev. Res.* **1**, 033063 (2019).
10. Schuld, M., Sinayskiy, I. & Petruccione, F. The quest for a quantum neural network. *Quant. Inf. Proc.* **13**, 2567–2586 (2014).
11. Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors. *Quant. Rev. Lett.* **1**, 2 (2020).
12. Aaronson, S. Read the fine print. *Nat. Phys.* **11**, 291–293 (2015).
13. Vapnik, V. *The Nature of Statistical Learning Theory* Vol. 8, 1–15 (Springer, 2000).
14. Vapnik, V. N. & Chervonenkis, A. Y. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.* **16**, 264–280 (1971).
15. Sontag, E. D *Neural Networks and Machine Learning* 69–95 (Springer, 1998).
16. Vapnik, V., Levin, E. & Cun, Y. L. Measuring the VC-dimension of a learning machine. *Neural Comput.* **6**, 851–876 (1994).
17. Neyshabur, B., Bhojanapalli, S., McAllester, D. & Srebro, N. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems 30*, 5947–5956 (NIPS, 2017).
18. Arora, S., Ge, R., Neyshabur, B. & Zhang, Y. Stronger generalization bounds for deep nets via a compression approach. In *Proc. 35th International Conference on Machine Learning* Vol. 80, 254–263 (PMLR, 2018); http://proceedings.mlr.press/v80/arora18b.html
19. Wright, L. G. & McMahon, P. L. The capacity of quantum neural networks. In *Conference on Lasers and Electro-Optics* JM4G.5 (Optical Society of America, 2020); http://www.osapublishing.org/abstract.cfm?URI=CLEO_QELS-2020-JM4G.5
20. Du, Y., Hsieh, M.-H., Liu, T. & Tao, D. Expressive power of parametrized quantum circuits. *Phys. Rev. Res.* **2**, 033125 (2020).
21. Huang, H.-Y. et al. Power of data in quantum machine learning. *Nat. Commun.* **12**, 2631 (2021).
22. Berezniuk, O., Figalli, A., Ghigliazza, R. & Musaelian, K. A scale-dependent notion of effective dimension. Preprint at https://arxiv.org/abs/2001.10872 (2020).
23. Rissanen, J. J. Fisher information and stochastic complexity. *IEEE Trans. Inf. Theory* **42**, 40–47 (1996).
24. Cover, T. M. & Thomas, J. A. *Elements of Information Theory* (Wiley, 2006).
25. Nakaji, K. & Yamamoto, N. Expressibility of the alternating layered ansatz for quantum computation. *Quantum* **5**, 434 (2021).
26. Holmes, Z., Sharma, K., Cerezo, M. & Coles, P. J. Connecting ansatz expressibility to gradient magnitudes and barren plateaus. Preprint at https://arxiv.org/abs/2101.02138 (2021).
27. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **9**, 1–6 (2018).
28. Wang, S. et al. Noise-induced barren plateaus in variational quantum algorithms. Preprint at https://arxiv.org/abs/2007.14384 (2020).
29. Cerezo, M., Sone, A., Volkoff, T., Cincio, L. & Coles, P. J. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nat. Commun.* **12**, 1791 (2021).
30. Verdon, G. et al. Learning to learn with quantum neural networks via classical neural networks. Preprint at https://arxiv.org/abs/1907.05415 (2019).
31. Volkoff, T. & Coles, P. J. Large gradients via correlation in random parameterized quantum circuits. *Quant. Sci. Technol.* **6**, 025008 (2021).
32. Skolik, A., McClean, J. R., Mohseni, M., van der Smagt, P. & Leib, M. Layerwise learning for quantum neural networks. *Quant. Mach. Intell.* **3**, 5 (2021).
33. Huembeli, P. & Dauphin, A. Characterizing the loss landscape of variational quantum circuits. *Quant. Sci. Technol.* **6**, 025011 (2021).
34. Bishop, C. Exact calculation of the Hessian matrix for the multilayer perceptron. *Neural Comput.* **4**, 494–501 (1992).
35. LeCun, Y. A., Bottou, L., Orr, G. B. & Müller, K.-R. *Efficient BackProp* 9–48 (Springer, 2012); https://doi.org/10.1007/978-3-642-35289-8_3
36. Cerezo, M. & Coles, P. J. Higher order derivatives of quantum neural networks with barren plateaus. *Quant. Sci. Technol.* **6**, 035006 (2021).
37. Kunstner, F., Hennig, P. & Balles, L. Limitations of the empirical Fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems 32* 4156–4167 (NIPS, 2019); http://papers.nips.cc/paper/limitations-of-fisher-approximation
38. Karakida, R., Akaho, S. & Amari, S.-I. Universal statistics of Fisher information in deep neural networks: mean field approach. In *Proc. Machine Learning Research* Vol. 89, 1032–1041 (PMLR, 2019); http://proceedings.mlr.press/v89/karakida19a.html

39. Schuld, M., Bocharov, A., Svore, K. M. & Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev. A* **101**, 032308 (2020).
40. Schuld, M., Sweke, R. & Meyer, J. J. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Phys. Rev. A* **103**, 032430 (2021).
41. Lloyd, S., Schuld, M., Ijaz, A., Izaac, J. & Killoran, N. Quantum embeddings for machine learning. Preprint at https://arxiv.org/abs/2001.03622 (2020).
42. Cong, I., Choi, S. & Lukin, M. D. Quantum convolutional neural networks. *Nat. Phys.* **15**, 1273–1278 (2019).
43. Amari, S.-I. Natural gradient works efficiently in learning. *Neural Comput.* **10**, 251–276 (1998).
44. Liang, T., Poggio, T., Rakhlin, A. & Stokes, J. Fisher–Rao metric, geometry, and complexity of neural networks. In *Proc. Machine Learning Research* Vol. 89, 888–896 (PMLR, 2019); http://proceedings.mlr.press/v89/liang19a.html
45. Neyshabur, B., Salakhutdinov, R. R. & Srebro, N. Path-SGD: path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems 28*, 2422–2430 (NIPS, 2015).
46. Neyshabur, B., Tomioka, R. & Srebro, N. Norm-based capacity control in neural networks. In *Proc. Machine Learning Research* Vol. 40, 1376–1401 (PMLR, 2015); http://proceedings.mlr.press/v40/Neyshabur15.html
47. Bartlett, P. L., Foster, D. J. & Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems 30*, 6240–6249 (NIPS, 2017); http://papers.nips.cc/paper/7204-spectrally-normalized
48. Rissanen, J. J. Fisher information and stochastic complexity. *IEEE Trans. Inf. Theory* **42**, 40–47 (1996).
49. Marrero, C. O., Kieferová, M. & Wiebe, N. Entanglement induced barren plateaus. Preprint at https://arxiv.org/abs/2010.15968 (2020).
50. Havlíček, V. et al. Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209–212 (2019).
51. Sim, S., Johnson, P. D. & Aspuru-Guzik, A. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Adv. Quant. Technol.* **2**, 1900070 (2019).
52. Jia, Z. & Su, H. Information-theoretic local minima characterization and regularization. In *Proc. 37th International Conference on Machine Learning* Vol. 119, 4773–4783 (PMLR, 2020); http://proceedings.mlr.press/v119/jia20a.html
53. Virmaux, A. & Scaman, K. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems 31*, 3835–3844 (NIPS, 2018); http://papers.nips.cc/paper/lipschitz-regularity-of-deep-neural-networks
54. Sweke, R. et al. Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum* **4**, 314 (2020).
55. Dua, D. & Graff, C. *UCI Machine Learning Repository* (2017); http://archive.ics.uci.edu/ml
56. Abbas, A. et al. *amyami187/effective_dimension: The Effective Dimension Code* (Zenodo, 2021); https://doi.org/10.5281/zenodo.4732830

Acknowledgements

We thank M. Schuld for insightful discussions on data embedding in quantum models. We also thank T. L. Scholten for constructive feedback on the manuscript. C.Z. acknowledges support from the National Centre of Competence in Research Quantum Science and Technology (QSIT).

Author contributions

The main ideas were developed by all of the authors. A.A. provided numerical simulations. D.S. and A.F. proved the technical claims. All authors contributed to the write-up.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43588-021-00084-1.

Correspondence and requests for materials should be addressed to S.W.

Peer review information *Nature Computational Science* thanks Patrick Coles and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Handling editor: Jie Pan, in collaboration with the *Nature Computational Science* team.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.