Quantum Machine Intelligence (2021) 3: 9
https://doi.org/10.1007/s42484-021-00038-w

RESEARCH ARTICLE

Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility, and entangling capability

Thomas Hubregtsen[1,2,3] · Josef Pichlmeier[3] · Patrick Stecher[4] · Koen Bertels[3]

Received: 16 March 2020 / Accepted: 18 January 2021 / Published online: 11 March 2021

Abstract

An active area of investigation in the search for quantum advantage is quantum machine learning. Quantum machine learning, and parameterized quantum circuits in a hybrid quantum-classical setup in particular, could bring advancements in accuracy by utilizing the high dimensionality of the Hilbert space as feature space. But is the ability of a quantum circuit to uniformly address the Hilbert space a good indicator of classification accuracy? In our work, we use methods and quantifications from prior art to perform a numerical study in order to evaluate the level of correlation. We find a moderate to strong correlation between the ability of the circuit to uniformly address the Hilbert space and the achieved classification accuracy for circuits that entail a single embedding layer followed by 1 or 2 circuit designs. This is based on our study encompassing 19 circuits in both 1- and 2-layer configurations, evaluated on 9 datasets of increasing difficulty. We also evaluate the correlation between entangling capability and classification accuracy in a similar setup, and find a weak correlation. Future work will focus on evaluating if this holds for different circuit designs.

Keywords Quantum neural networks · Parameterized quantum circuits · Expressibility · Quantum machine learning · Quantum computing · Entangling capability

1 Introduction

Quantum computing has seen a steady growth in interest ever since the quantum supremacy experiment (Arute et al. 2019). The search for quantum advantage, quantum supremacy for practical applications, is an active area of research (Bravyi et al. 2020). One potential domain of applications is machine learning (Riste et al. 2017). Here, quantum computing is said to potentially bring speedups and improvements in accuracy. One line of reasoning to assume an improvement in accuracy is as follows. A classical neural network takes the input data and maps it into a higher dimensional feature space. Then, using a combination of learnable linear transformations and static non-linear transformations, it maps the data between various higher dimensional feature spaces. This mapping is repeated until the data points are positioned in such a way that a hyperplane can separate data that belongs to different output classes. Given that qubits, the smallest units of information in a quantum computer, can span a larger space due to their quantum mechanical properties, one would expect that with the same resources, data could be mapped between higher dimensional and larger feature spaces. This would allow a more accurate separation of the data. Or, as a trade-off, one would require fewer resources to address the same space and maintain the same level of accuracy.

Recently, a framework compatible with shallow depth circuits for noisy intermediate-scale quantum (Preskill 2018) systems has been developed, named the hybrid quantum-classical framework. In this framework, the quantum machine leverages *parameterized quantum circuits* (PQCs) in order to make predictions and approximations, while the classical machine is used to update the parameters of the circuit (McClean et al. 2016). Example algorithms are

✉ Thomas Hubregtsen
thomas.hubregtsen@bwmgroup.com

1 BMW Group Research, New Technologies, Innovations, Garching bei München, Germany

2 Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, Berlin, Germany

3 Delft University of Technology, Delft, Netherlands

4 Department of Informatics, Technical University of Munich, Garching bei München, Germany

the Variational Quantum Eigensolver (Peruzzo et al. 2014) and the Quantum Approximate Optimisation Algorithm (Farhi et al. 2014). Variational Quantum Circuits can also be used for machine learning problems, and bear resemblance to the structure of classical neural networks (Schuld et al. 2018).

Just as in classical neural networks, many circuit architectures exist. An active area of research focuses on determining the power and capabilities of these circuits (Coyle et al. 2020; Sim et al. 2019; Schuld et al. 2020). Following the previous line of reasoning of mapping data in larger feature spaces, one way of quantifying the power of a PQC is by quantifying the ability of a PQC to uniformly reach the full Hilbert space (Sim et al. 2019). Our work investigates these definitions by performing a numerical analysis on the correlation between such descriptors and classification accuracy in order to guide the choice and design of PQCs, as well as provide insight into potential limitations.

The remainder of this paper is structured as follows. In Section 2, we will describe our approach. Section 3 will outline our experimental setup and design choices. Our results are presented in Section 4 and discussed in Section 5. We will end this paper with the conclusion in Section 6. Raw experimental data can be found in the Appendices.

2 Approach

We will use prior art for descriptors that quantify the ability of a PQC to explore the Hilbert space. We will perform numerical simulations on various custom datasets of increasing difficult to quantify classification performance of these circuits. The dataset will be split in a train, validation, and test sets. The train set will be used for training the PQC and the validation set for evaluating our hyperparameter search. The final evaluation of our model with the best hyperparameters is performed on the test set, which will yield the data points that we will use in our search for correlation. We will repeat the experiments on the validation set to make sure the outcome is consistent. We will perform a statistical analysis to evaluate a potential relation between the descriptors and the classification accuracy of the various circuits.

3 Evaluation

Variational Quantum Circuits applied for machine learning problems bear a resemblance to classical *neural networks* (NN), which is why they are also sometimes referred to as *quantum neural networks* (QNNs) (Farhi and Neven 2018). They can be used for classification and regression tasks in a supervised learning approach, or as generative models in an unsupervised setup. We will solely focus on supervised prediction. Both NNs and QNNs embed the data into a higher dimensional representation using an input layer. The input layer of a QNN is called an *embedding circuit*. The hidden layers of a QNN are referred to as *circuit templates*, which together constitute a PQC. The variables in these layers, called *weights* in NNs and *parameters* or *variational angles* in QNNs, are initialized along a probability distribution before training starts. The most notable difference between a NN and a QNN is in the way the output is handled. A NN uses an output layer to directly generate a distribution over the possible output classes using only a single run. A QNN needs to be executed several times, referred to as *shots*, before a histogram over the output states can be generated, which still needs to be mapped to the output classes. The predicted output class is compared to the true output class, denoted as $y_{predicted}$ and y_{true}, respectively, and the error is quantified by the loss function. This quantification of the error is used to guide the optimizer to adjust the parameters in an iterative process until convergence in the loss is reached. The measure of *classification accuracy* (Acc), or simply *accuracy*, is the number of correctly classified samples over the total number of samples. In the remainder of this section, we will first discuss the descriptors of PQCs that we will investigate, before going into more depth on the experimental setup that we use to investigate the correlation between the descriptors and the classification accuracy.

3.1 Descriptors of PQCs

Describing the performance of a PQC by the ability of the circuit to uniformly address the Hilbert space has been suggested by Sim et al. (2019). In their theoretical approach, Sim et al. (2019) propose to quantify this by comparing the true distribution of fidelities corresponding to the PQC, to the distribution of fidelities from the ensemble of Haar random states. In practice, they propose to approximate the distribution of fidelities, the overlap of states defined $F = |\langle\psi_\theta|\psi_\phi\rangle|^2$, of the PQC. They do this by repeatedly sampling two sets of variational angles, simulating their corresponding states, and taking the fidelity of the two resulting states to build up a sample distribution $\hat{P}_{PQC}(F;\Theta)$. The ensemble of Haar random states can be calculated analytically: $P_{Haar} = (N-1)(1-F)^{N-2}$, where N is the dimension of the Hilbert space (Życzkowski and Sommers 2005). The measure of *expressibility* (Expr) is then calculated by taking the Kullback-Leibner divergence (KL divergence) between the estimated fidelity distribution and that of the Haar distributed ensemble:

$$Expr = D_{KL}(\hat{P}_{PQC}(F;\Theta)\|P_{Haar}(F)). \quad (1)$$

A smaller value for the KL divergence indicates a better ability to explore the Hilbert space. This measure of expressibility is observed on a logarithmic scale. This is where we decided to deviate from the original definition, as we will include these characteristics into the measure itself. In our work, we will evaluate the negative logarithmic of the KL divergence between the ensemble of Haar random states and the estimated fidelity distribution of the PQC, so that larger values for expressibility' correspond to better ability to explore the Hilbert space, and will be correlated on this logarithmic scale. We will refer to this as *expressibility'* (Expr'), distinguished by the ' symbol:

$$Expr' = -\log_{10}(D_{kl}(\hat{P}_{PQC}(F;\Theta)\|P_{Haar}(F))). \tag{2}$$

In the same paper, Sim et al. (2019) define a second descriptor named *entangling capability*. This descriptor is intended to capture the entangling capability of a circuit which, based on prior art such as work by Schuld et al. (2018) and Kandala et al. (2017), allows a PQC to capture non-trivial correlations in the quantum data. Multiple ways to compute this measure exist, but the Meyer-Wallach entanglement measure (Meyer and Wallach 2002), denoted Q, is chosen due to its scalability and ease of computation. It is defined as

$$Q(|\psi\rangle) \equiv \frac{4}{n}\sum_{j=1}^{n} D(\iota_j(0)|\psi\rangle, \iota_j(1)|\psi\rangle) \tag{3}$$

Where $\iota_j(b)$ represents a linear mapping for a system of n qubits that acts on the computational basis with $b_j \in \{0, 1\}$:

$$\iota_j(b)|b_1 \dots b_n\rangle = \delta_{bb_j}|b_1 \dots \hat{b}_j \dots b_n\rangle. \tag{4}$$

Here, the symbol ˆdenotes the absence of a qubit. In practice, also this measure of the PQC needs to be approximated by sampling, so Sim et al. (2019) define the estimate of *entangling capability* (Ent) of the PQC to be the following:

$$Ent = \frac{1}{|S|}\sum_{\theta_i \in S} Q(|\psi_{\theta_i}\rangle), \tag{5}$$

where S represents the set of sampled circuit parameter vectors θ.

The quantification of both expressibility and entangling capability for the circuits, as found and provided by Sim et al. (2019), is shown in Table 5 in Appendix 1.

3.2 Datasets

Many datasets for the evaluation of classical machine learning algorithms exist (Goldbloom and Hamner 2020). However, classical machine learning is more advanced as a field compared to quantum machine learning, and currently capable of both processing data at larger scales and predicting more complex distributions. We have searched for a set of problems of increasing and varying difficulty, but not of a larger problem size, to cover a wider range of problems to benchmark against. We were unable to find any that satisfied our needs, and therefore came up with a set of nine datasets for the classifiers to fit, as shown in Fig. 1, labelled numerically in the vertical direction and alphabetically in the horizontal direction. Here, we included datasets where the two classes are bordering one another (1a), require one (1b) or more (1c) bends in a decision boundary, entrap one another (2a, 2b, 2c), or fully encapsulate each other (3a, 3b, 3c). Each dataset contains a total of 1500 data points for training, testing hyperparameters, and validation in a ratio of 3 : 1 : 1. The ratio of data points labelled true versus data points labelled false is 1 : 1, e.g., all datasets are balanced. All data points are cleaned and normalized.

3.3 Embedding the data

Various ways to embed classical data into quantum circuits have been proposed, such as Amplitude Encoding (Schuld and Killoran 2019), Product Encoding (Stoudenmire and Schwab 2016), or Squeezed Vacuum State Encoding (Schuld and Killoran 2019). Important distinguishing characteristics are time complexity, memory complexity, and fit for prediction circuits, as some prediction algorithms require data in a particular format. This part of the QNN is also referred to as the *state preparation circuit*, *embedding circuit* or the *feature embedding circuit*. However, the existing circuits do not meet our requirements. We search for an embedding that:

- Holds as little expressive power as possible, as we want to observe the expressive power of the PQC
- Embeds the data equally in all computational bases, to ensure circuit templates with initial gates acting on particular basis have an equal chance of having an observable effect

For this, we propose a novel embedding that we will refer to as the *minimally expressive embedding*, denoted $W(x)$ in later calculations and shown in Fig. 2. To make sure this embedding circuit holds as little expressive power as possible, we embed the classical data into a quantum state using a linear mapping into the range $(0, 2\pi]$ using the parameter of a Rotational X gate. This can be visualized for this single qubit as embedding data along a circle in the Bloch sphere, as shown in Fig. 2b. We then use a Rotational Y and Rotational Z gate to rotate the circle of embedded data 45° both in the X-axis and Y-axis. The result is an embedding that, when projected down on the various computational basis, can address a similar range in every computational basis, allowing no bias to favor

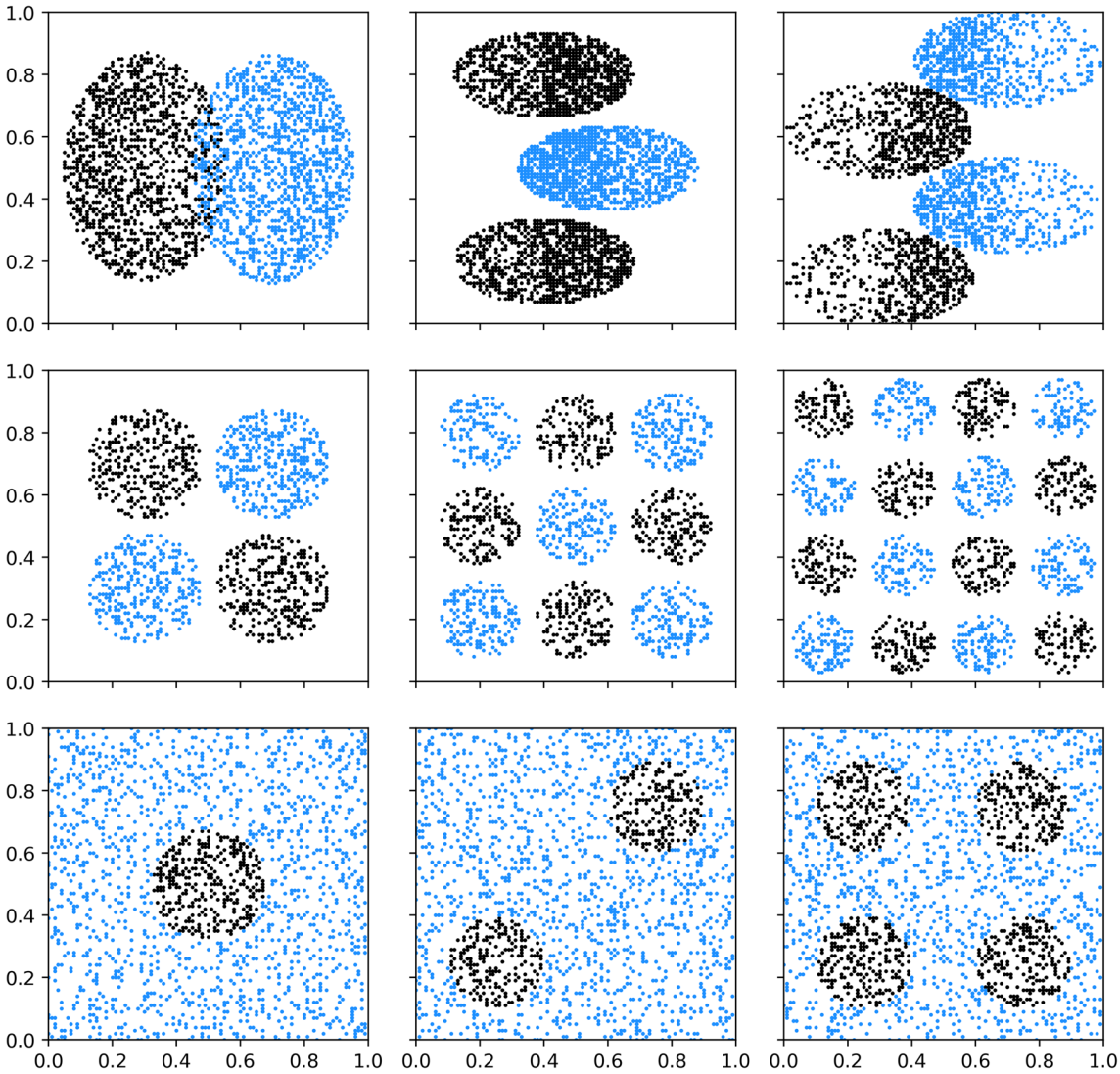

Fig. 1 Datasets used to benchmark classification accuracy, labelled numerically in the vertical direction and alphabetically in the horizontal direction

circuits that start with specific (Rotational) Pauli gates that could be off-axis for other embeddings. This can be seen in Fig. 2c and d.

We embed the 2-dimensional data in a replicative fashion: data on the x-axis in Fig 1 is embedded in qubits 0 and 2, data on the y-axis is embedded in qubits 1 and 3. We do this for two reasons:

- 2-Dimensional data can be easily visualized for a better understanding of both the data and the fitted decision boundary
- Input redundancy is suggested to provide an advantage in classification accuracy (Vidal and Theis 2019)

Due to computational restrictions and available nodes for simulation, we restricted the data redundancy to a factor of 2.

3.4 The parameterized quantum circuit

In this paper, we aim to explore the correlation between our previously introduced descriptors and classification accuracy for various circuits and layers. Every experiment of ours follows the same design: a single embedding circuit followed by 1 or 2 circuit templates. For the circuit templates, denoted $U(\theta)$ when used later, we use the circuits presented in Sim et al. (2019). In this work, they introduce circuits with parameterized gates R_X, R_Y, and R_Z, as defined in Nielsen and Chuang (2002). The exact layout of the circuit templates can be consulted in Fig. 5 in Appendix 2 of our paper. All 19 circuits, except circuit 10, are used in the hyperparameter selection runs. All 19 circuits are used in the validation runs.

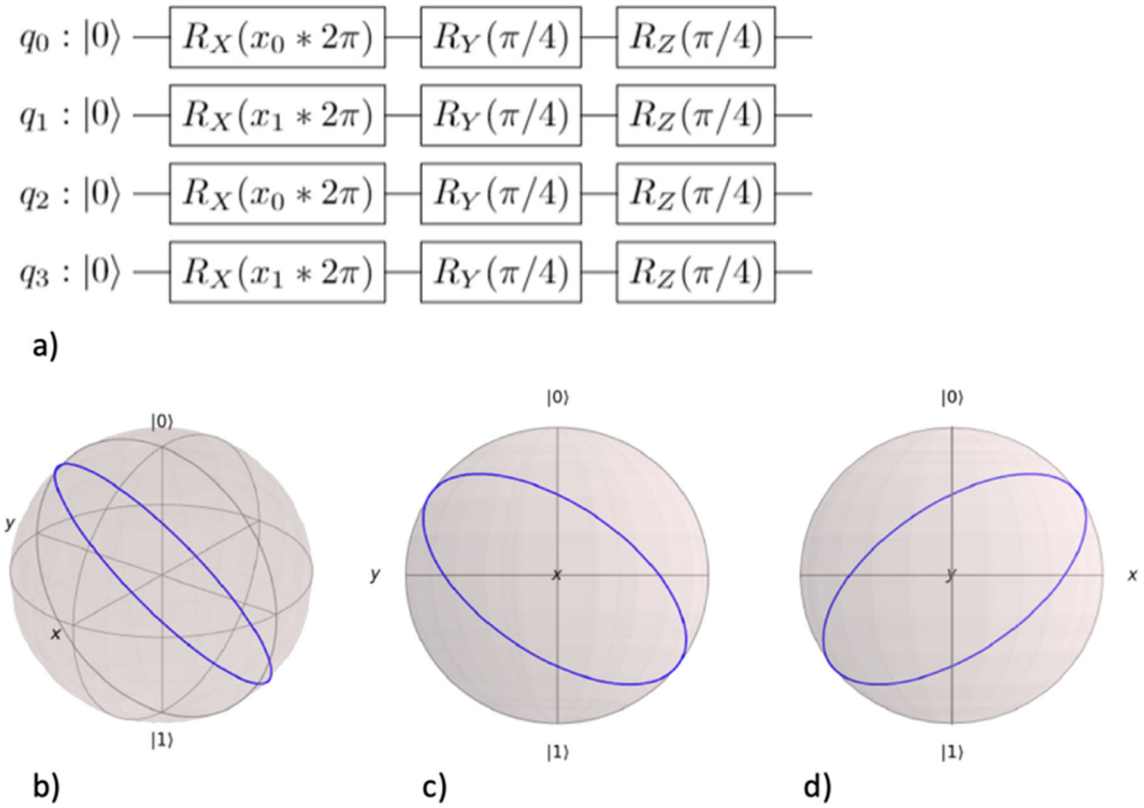

Fig. 2 The various aspects of the minimally expressive embedding: **a** the embedding circuit; **b** the quantum state achieved when uniformly sampling input data, viewed in 3d; **c**, **d**: the quantum state achieved when uniformly sampling input data, in 2d on y-z and in 2d on x-z

3.5 Measurement observable and mapping

The circuit needs to be measured repeatedly. The number of times can be significantly less as previously thought (Sweke et al. 2019). For our observable A, we use Pauli Z gates on all four qubits. This results in a total of 16 possible states, of which we map the first four and last four to output class -1 and the other to output class 1, in line with previous work (Havlicek et al. 2018).

3.6 Loss function

Loss functions quantify the errors between the predicted class of the sample and the actual class. This can be done in several ways, which creates different loss landscapes. These loss landscapes have different properties for different loss functions, example properties being plateaus, local minima, and global minima that do not align with the true minima. In our work, we will evaluate the *L1* loss and *L2* loss.

The L1 loss can be seen as the most straightforward loss function, taking the absolute value of the difference between y_{true} and y_{pred}:

$$L_1 = \sum_{s=0}^{samples} |y_{pred_s} - y_{true_s}|. \tag{6}$$

Here, $y_{pred} = \langle 0|W^\dagger(x)U^\dagger(\theta)AU(\theta)W(x)|0\rangle$, with $W(x)$ and $U(\theta)$ representing the embedding circuit and the circuit template respectively, as previously defined. Furthermore, y_{true} is given as ground truth by the input dataset.

The L2 loss does not use the absolute value but instead ensures positive outcomes through taking the square of the difference, which is easier to differentiate. This square also penalizes large errors harder than small errors, reducing the chance of overfitting. The L2 loss is defined as follows:

$$L_2 = \sum_{s=0}^{samples} (y_{pred_s} - y_{true_s})^2. \tag{7}$$

For binary values of y, the L2 loss and the L1 loss are equal. This is not the case for continuous values.

3.7 Optimizer

The task of the optimizer is to find updates to the parameters based on the outcome of the loss function in such a way that, after repeated runs, the loss is minimized. Many approaches make use of the gradient, either analytically (Schuld et al. 2019) or approximately (Sweke et al. 2019). Alternative approaches exist, such as genetic optimization strategies (Mirjalili and Sardroudi 2012). Either way, a balance needs

to be struck between making large enough jumps to get out of local minima and plateaus, and making small enough updates so that the jumps do not equal random walks in the loss landscape. In our work, we evaluate the *Adam* optimizer and the *gradient descent* optimizer, due to their popularity in classical frameworks and their availability in our implementation framework.

3.8 Implementation framework

We originally implemented our work in Qiskit (Abraham 2019), but switched to Pennylane (Bergholm et al. 2018) due to the ease for our hyperparameter search.

3.9 Training setup

We will first perform a hyperparameter search on the loss functions L1 and L2, as well as on the optimizers Adam and gradient descent, all introduced previously in this section. We compare their performance in terms of average classification accuracy over the validation data of our nine datasets. As we evaluate 2 loss functions and 2 optimizers using 19 circuits in both 1- and 2-layer configurations on 9 datasets, we have to train a total of 1368 quantum circuits. We simulate these 1368 circuits classically in our previously mentioned implementation framework, Pennylane (Bergholm et al. 2018). We will report on the difference in accuracy for different numbers of layers of each circuit but as each layer configuration has its own value for the descriptors, we will treat each layer configuration as unique data points in our final analysis. The optimal hyperparameters derived through the validation runs are used as settings for the test runs during which we gather our final accuracy values for all 19 circuits in both 1- and 2-layer configurations evaluated on 9 datasets each. As a sanity check, we will repeat the final experiment three times, requiring a total of 1026 additional simulations.

3.10 Defining correlation

In order to determine the correlation, various indicators have been considered. We decided on the Pearson product-moment correlation coefficient (Boddy and Laird Smith 2009), as the assumptions on the underlying data distribution are the best fit. In our threats to validity, we will make a brief note on linear versus polynomial fitting. We will use the Pearson coefficient to determine the correlation between expressibility and classification accuracy, as well as between entangling capability and classification accuracy on the 342 data points, as described in the previous section. We will use these coefficients to draw conclusions on the level of correlation between the descriptors and the classification accuracy.

Table 1 Accuracy for various hyperparameter settings, averaged out over the datasets

Opt. \ Layers	1	2	Loss \ Layers
Adam	72.6	76.7	$L1$
	76.8	**82.6**	L2
Gradient	73.9	71.5	$L1$
descent	74.2	75.6	$L2$

Bold entries show the recommended option

3.11 Classical neural network

We will also evaluate our dataset using a classical neural network for comparison and sanity checking. We implemented both 1- and 2-layer versions, each having 16 weights per layer. The activation function used is the *Rectified Linear Unit* (ReLU), and the system is optimized using the Adam optimizer. All is implemented in *Tensorflow* (Abadi 2015).

4 Results

The classification accuracy for the various hyperparameter settings can be found in Appendix 4. In particular, the results for the Adam optimizer with L1 loss can be found in Table 9 and with L2 loss in Table 10. The results for the gradient descent optimizer with L1 loss in Table 11 and with L2 loss in Table 12. The average classification accuracy across all datasets for the various hyperparameter settings, as well as the number of layers, is summarized in Table 1. Here, we see that the Adam optimizer combined with L2 loss achieves the best classification accuracy, regardless of the number of layers. Treating each hyperparameter separately using the *factorial design* (Bose 1947), as shown in Table 2, reconfirmed these settings. The outcome of the three validation runs using the L2 loss and Adam optimizer can be found in Appendix 3, Table 7 for the 1-layer runs and Table 8 for the 2-layer runs. We calculated the mean absolute difference between each of the 342 data points of

Table 2 Factorial design to evaluate classification accuracy (*Acc*) with regard to the dependent variables (*DV*)

DV	Option	Acc	Option	Acc
Loss	L1	73.7	$L2$	**77.3**
Optimizer	Adam	**77.2**	GD	73.8
Layers	1	74.4	2	**76.7**

Bold entry shows the highest accuracy

every run, and found it to be 0.06 with a standard deviation of 0.36. With this info, we expect the correlation not to vary significantly between the runs, which is also what we observe in Table 3, where we show the Pearson product-moment correlation coefficients between the descriptors and classification accuracy for each dataset individually, as well as the mean and standard deviation. We added an extra row where we exclude dataset 2a for reasons described in the "Discussion" section. The relation between expressibility' and classification accuracy for every dataset is plotted in Fig. 3, and the relation between entangling capability and classification accuracy is plotted in Fig. 4. The average classification accuracy for the classical NN is shown in Table 6 of Appendix 3.

5 Discussion

5.1 Limitations of the experiment

Before discussing the results of our experiment, we will address the limiting factors to place our findings in the correct perspective.

- Our experiment only includes specific quantum circuit designs. In particular, the system always starts with an embedding circuit, followed by 1 or 2 circuit templates, with rotational gates, parameterized rotational gates, and conditional rotational gates
- We handcrafted 9 different sets of data points of increasing difficulty that we believe to be realistic and representative problems for current-day quantum machine learning algorithms. However, the sets only contain classical data with 2 features. Data encapsulating more complex patterns, or higher dimensional data, might yield different results
- The 2d data was embedded in a replicative manner on 4 qubits with our proposed minimally expressive embedding circuit. Different embeddings, additional embeddings, or repetitive embeddings are not evaluated. Neither were ancilla qubits.

Furthermore, different descriptors for expressibility and entangling capability exist, as well as different descriptors for the power of a PQC overall(Coyle et al. 2020; Sim et al. 2019; Schuld et al. 2020).

5.2 Correlation using the best hyperparameter settings

For our experimental setup, we have found an average Pearson correlation coefficient between expressibility' and classification accuracy of 0.64 ± 0.16, as calculated from run 1 in Table 3. However, when looking at the plots in Fig. 3, we see that dataset 2a contains 33 out of 38 data points close to or at 100% accuracy. As these accuracies are capped out, no meaningful correlation can be expected. For this reason, we mark dataset 2a as faulty and exclude it from our evaluation. This brings us to a final mean Pearson correlation coefficient between expressibility' and classification accuracy of -0.70 ± 0.05. This indicates a strong correlation (Dancey and Reidy 2007) between classification accuracy

Table 3 Pearson product-moment correlation coefficient between expressibility' and classification accuracy, as well as between entangling capability and classification accuracy, for the various datasets

	Expr' vs Acc			Ent vs. Acc		
Dataset	Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
1a	0.575	0.570	0.571	0.467	0.463	0.466
1b	0.699	0.699	0.699	0.353	0.353	0.353
1c	0.675	0.678	0.676	0.419	0.421	0.420
2a	0.200	0.200	0.200	−0.257	−0.257	−0.258
2b	0.761	0.761	0.777	0.251	0.251	0.258
2c	0.700	0.707	0.694	0.339	0.343	0.336
3a	0.732	0.731	0.740	0.190	0.189	0.195
3b	0.693	0.686	0.693	0.231	0.226	0.231
3c	0.727	0.730	0.730	0.301	0.303	0.303
Mean	0.640	0.640	0.642	0.255	0.255	0.256
Stdev	0.163	0.164	0.165	0.199	0.199	0.199
Mean'	0.695	0.695	0.697	0.319	0.319	0.320
Stdev'	0.052	0.054	0.057	0.089	0.089	0.087

Mean and standard deviation are taken over all datasets; mean' and standard deviation' are taken over all datasets but dataset 2a

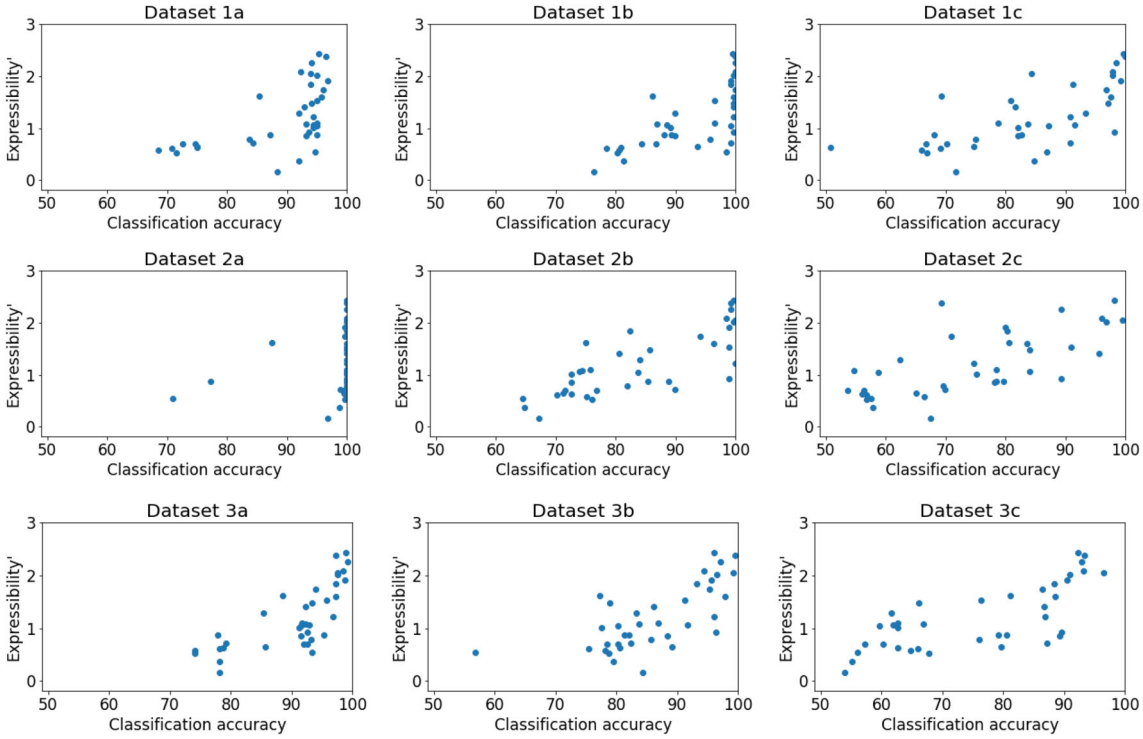

Fig. 3 Each plot contains 38 data points, collected by fitting 19 circuits in 1- and 2-layer configurations to the particular dataset, depicting the the relation between classification accuracy on the horizontal axis and expressibility' on the vertical axis

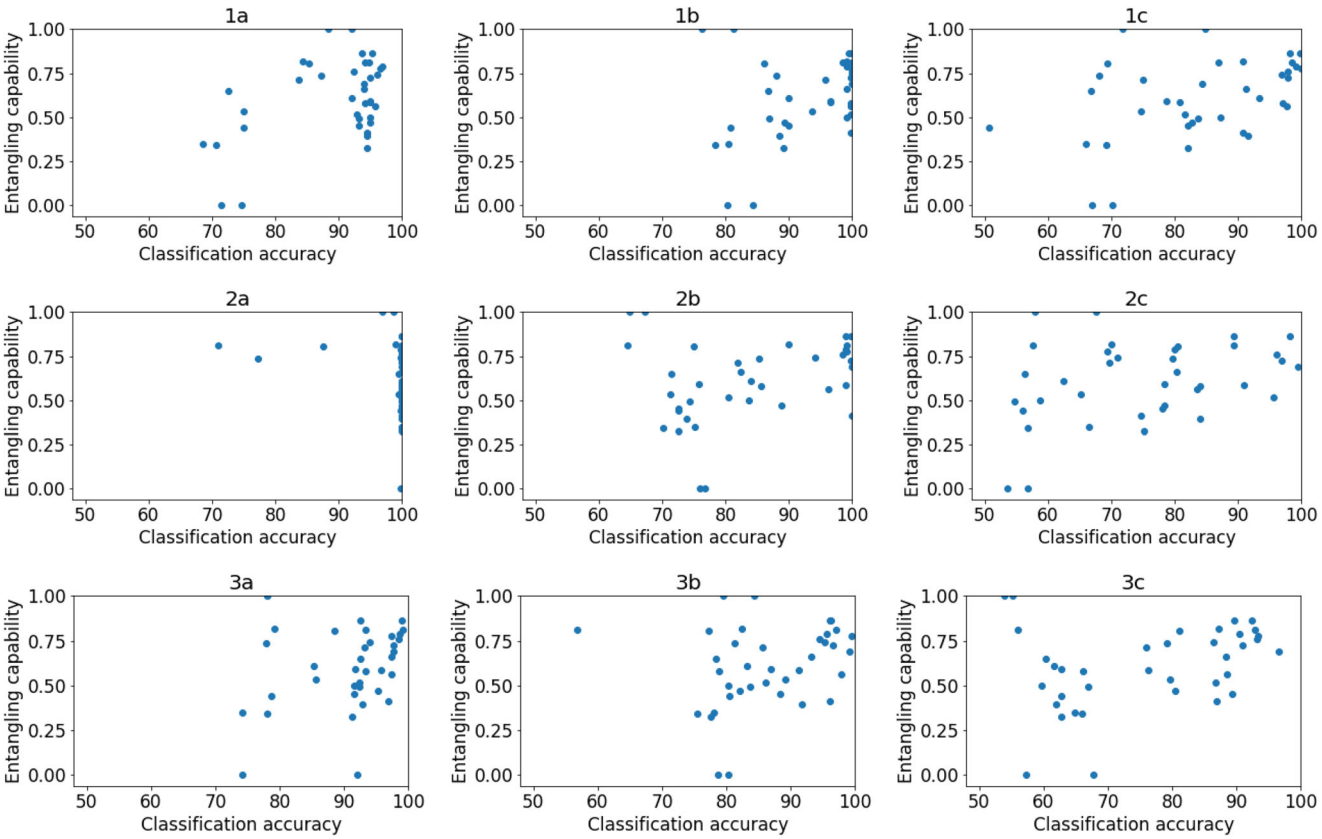

Fig. 4 Each plot contains 38 data points, collected by fitting 19 circuits in 1- and 2-layer configurations to the particular dataset, depicting the the relation between classification accuracy on the horizontal axis and entangling capability on the vertical axis

and expressibility'. This is within expectations (Sim et al. 2019).

Using the same experimental setup, the mean Pearson correlation coefficient between entangling capability and classification accuracy across all datasets is 0.26 ± 0.20. As this experiment suffers the same characteristics for dataset 2a, as observed in Fig. 4, we also decide to exclude dataset 2a for our evaluation of this coefficient. This brings the final mean Pearson correlation coefficient between entangling capability and classification accuracy at 0.32 ± 0.09. This indicates a weak correlation (Dancey and Reidy 2007) between classification accuracy and entangling capability. This is not as expected by Sim et al. (2019), but in line with recent findings that argue that an excess in entanglement between the visible and hidden units in a quantum neural network can hinder learning (Marrero et al. 2020).

5.3 Correlation using the best hyperparameter settings

5.4 Correlation using the remaining hyperparameters

As a sanity check on our strong results, we calculated the mean Pearson correlation coefficient and its standard deviation for the different hyperparameter settings on the validation data, all with dataset 2a excluded. This is summarized in Table 4. Here we see that for the Adam optimizer with either L1 or L2, we maintain a value that can be classified as a strong correlation. For gradient descent, the claim would be weaker, being classified as a moderate to strong correlation (Dancey and Reidy 2007). After examining the data, we see that this is caused by the optimizer not performing optimally. In particular, at least half of the circuits on datasets 2c and 3c cannot fit any data, staying around 50% accuracy, and the majority of the circuits are stuck in a local minima around 80% for dataset 3b. This is not surprising, as we saw in Table 2 that the average classification accuracy while using the gradient descent optimizer is lower compared to using the Adam optimizer.

Table 4 Mean Pearson correlation coefficient for expressibility and classification accuracy as a sanity check on the remaining hyperparameters

Optimizer	Loss	Mean	Stdev
Adam	L1	0.63	0.13
Adam	L2	0.70	0.08
GD	L1	0.46	0.11
GD	L2	0.56	0.13

5.5 Limitations of the descriptor

The original paper by Sim et al. (2019) contains 19 circuits with 1 to 5 layers. In their paper, they address that expressibility values appear to saturate for all circuits, albeit at different levels. The lowest value of expressibility that they present is 0.0026, which in our measure ofexpressibility' corresponds to $-log_{10}(0.0026) = 2.59$. In our experiments, we tested 6 circuits that have a value of expressibility' higher than 2, meaning an expressibility smaller than 0.01. The best average accuracy achieved by any of our circuits is 97.7. Still, this includes a fitting of only 92.3% for dataset 3b. In comparison, a classical neural network containing 2 layers with 16 weights is able to achieve an average classification accuracy of 99% without any hyperparameter tuning, as shown in Table 6. When looking at state-of-the-art neural network architectures, we continue to see these patterns. Most quantum classifiers are still being evaluated on small datasets such as ours (Schuld and Killoran 2019; Havlicek et al. 2018), or datasets such as the MNIST (Deng 2012). Classical machine learning models on the other hand are being evaluated on larger and more complex datasets, such as ImageNet (Deng et al. 2009). Although this may be purely due to the infancy of the current systems, limiting the size of the input data, it still appears that adding extra layers does not circumvent the saturation. We believe a hint might lay in the following reasoning: it is not only important for a classifier to be able to uniformly address a large Hilbert space, but also requires a repeated non-linear mapping between these spaces. As a thought experiment, think of a circuit that embeds linearly increasing classical data with a single rotational Pauli X gate. In this scenario, one would not expect to be able to find a separating plane between the two classes without remapping the data. We believe that this is also the reason why in classical neural networks, the data is repeatedly mapped between feature spaces by repeated layers of linear neurons and non-linear activation functions. We believe that quantum circuits need to be designed and evaluated in such a manner too. Recent research addresses this by having alternating layers of embedding and trainable circuit layers (Pérez-Salinas et al. 2020), thereby breaking linearity, although Schuld et al. (2020) drive a point that classical systems can achieve a similar effect with similar resources, deeming the use of quantum for ML not necessary. The investigation into the effect of the design on our descriptors is marked as future work.

5.6 Threats to validity

We are aware that different hyperparameters, differently drawn values from our probability distribution initializating the weights, and different datasets can yield different results. We have aimed to account for this effect by performing a hyperparameter search to find the best hyperparameters for our main result, but still evaluate for the remaining hyperparameters to confirm our results. We also accounted for different values drawn from our probabilitydistribution for initializations of the weights by repeating our experiments three times on 9 different datasets of increasing difficulty. For fitting our correlation indicator, we consider st until 3rd-order polynomials, and did not observe a noticeable difference.

6 Conclusion

In our work, we have found a a mean 0.7 ± 0.05 Pearson product-moment correlation coefficient between classification accuracy and expressibility' yielding a strong correlation (Dancey and Reidy 2007). This numerically derived outcome is calculated using 342 data points. These data points were generated using 19 circuits, in both 1- and 2-layer configurations, evaluated on 9 custom datasets of increasing difficulty. Similar investigation on the other hyperparameters yielded mean coefficients between 0.46 and 0.7, yielding a moderate to positive correlation (Dancey and Reidy 2007). This combination leads us to conclude that for our experiments, a moderate to strong correlation between classification accuracy and expressibility' exists. Here, expressibility' is based on the definition of expressibility proposed by Sim et al. (2019), and meant to capture the ability of a parameterized quantum circuit in a hybrid quantum-classical framework to uniformly address the Hilbert space. This is calculated by taking the negative log of the Kullback–Leibler divergence between the ensemble of Haar random states and the estimated fidelity distribution of the PQC. This outcome is in line with expectations by Sim et al. (2019).

Our experiment is limited to PQCs that follow a specific pattern of concatenating an embedding layer with one or more circuit templates. It is suggested that this circuit setup is limiting the classification performance(Schuld et al. 2020). Further investigation into more elaborate designs that break linearity after embedding is required, for example by repeated alteration between embedding layers and trainable layers (Schuld et al. 2020). Such designs would potentially not be captured by the expressibility' measure, and further investigation into extending the measure is required.

We have also investigated the correlation between entangling capability of a circuit and its classification accuracy, where entangling capability was measured as the Meyer-Wallach entanglement measure (Meyer and Wallach 2002). The outcome was a weak correlation, based on a similar experimental setup that yielded a mean Pearson product-moment correlation of 0.32 ± 0.09 (Dancey and Reidy 2007). This is not as expected by Sim et al. (2019), but in line with recent findings by Marrero et al. (2020).

Appendix 1: Expressibility and entangling capability

Table 5 Expressibility, expressibility', and entangling capability of the circuits

	Layer 1			Layer 2		
Circuit	Expr	Expr'	Ent	Expr	Expr'	Ent
1	0.2995	0.52	0.0	0.1972	0.71	0.0
2	0.2875	0.54	0.81	0.0244	1.61	0.8
3	0.24	0.62	0.34	0.0847	1.07	0.49
4	0.1353	0.87	0.47	0.0291	1.54	0.59
5	0.0601	1.22	0.41	0.0087	2.06	0.69
6	0.0041	2.38	0.78	0.0036	2.44	0.86
7	0.0985	1.01	0.33	0.0386	1.41	0.52
8	0.0864	1.06	0.39	0.0255	1.59	0.56
9	0.678	0.17	1.0	0.4261	0.37	1.0
10	0.2284	0.64	0.54	0.1617	0.79	0.71
11	0.1325	0.88	0.73	0.0122	1.92	0.79
12	0.2003	0.7	0.65	0.0181	1.74	0.74
13	0.0516	1.29	0.61	0.0083	2.08	0.76
14	0.0144	1.84	0.66	0.0055	2.26	0.81
15	0.191	0.72	0.82	0.1185	0.93	0.86
16	0.2615	0.58	0.35	0.0885	1.05	0.5
17	0.1378	0.86	0.45	0.0327	1.49	0.58
18	0.2358	0.63	0.44	0.0602	1.22	0.62
19	0.0814	1.09	0.59	0.0096	2.02	0.72

Appendix 2: Circuits

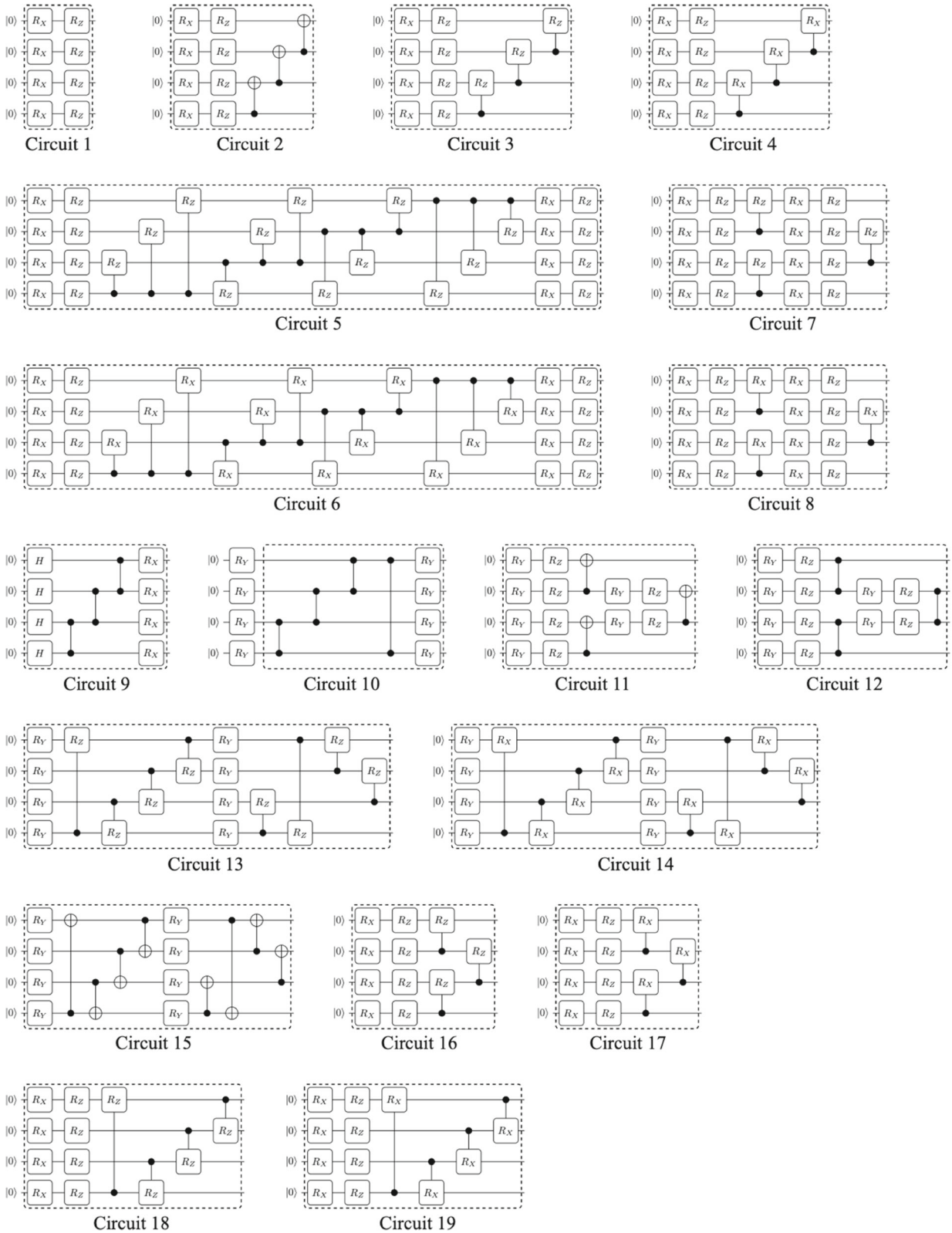

Fig. 5 Circuit templates evaluated in this study. Gates as defined in Nielsen and Chuang (2002). All Rotational Gates are parameterized. Circuit diagrams were generated using qpic (KutinS 2016). Please see copyright notice (Notice regarding choice of circuit templates and visualizations of these circuits: Copyright Wiley-VCH GmbH. Reproduced with permission. Source: 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 (2019): 1900070. Page 8.)

Appendix 3: Test results

Table 6 Test results—classical neural network

Setup	Performance									
Layers	1a	1b	1c	2a	2b	2c	3a	3b	3c	Avg.
1	96%	100%	100%	92%	94%	82%	100%	98%	70%	92%
2	96%	100%	100%	100%	99%	98%	100%	99%	97%	99%

Table 7 Test results — Adam optimizer, L2 loss, 1 layer

Setup		Performance									
Run	Circuit	1a	1b	1c	2a	2b	2c	3a	3b	3c	Avg.
1	1	71.5%	80.3%	66.9%	99.7%	76.0%	56.8%	74.1%	78.7%	67.7%	74.6%
1	2	94.7%	98.4%	86.9%	70.9%	64.5%	57.6%	93.3%	56.8%	56.0%	75.5%
1	3	70.7%	78.4%	69.1%	100.0%	70.1%	56.8%	78.1%	75.5%	65.9%	73.8%
1	4	94.9%	89.3%	82.7%	100.0%	88.8%	78.4%	95.2%	82.1%	80.5%	88.0%
1	5	94.4%	99.7%	90.7%	100.0%	100.0%	74.7%	96.8%	96.0%	86.9%	93.2%
1	6	96.5%	100.0%	100.0%	100.0%	99.2%	69.3%	97.3%	99.5%	93.3%	95.0%
1	7	94.4%	89.1%	82.1%	100.0%	72.5%	75.2%	91.2%	77.6%	62.7%	82.8%
1	8	94.4%	88.5%	91.5%	100.0%	73.9%	84.0%	92.8%	91.7%	61.9%	86.5%
1	9	88.3%	76.3%	71.7%	96.8%	67.2%	67.5%	78.1%	84.3%	53.9%	76.0%
1	10	74.9%	93.6%	74.7%	99.5%	71.2%	65.1%	85.6%	89.1%	79.7%	81.5%
1	11	87.2%	88.0%	68.0%	77.3%	85.3%	79.7%	77.9%	81.3%	79.2%	80.4%
1	12	72.5%	86.7%	66.7%	99.5%	71.5%	56.3%	92.5%	78.4%	60.3%	76.0%
1	13	92.0%	89.9%	93.3%	100.0%	84.0%	62.4%	85.3%	83.2%	61.6%	83.5%
1	14	93.9%	99.2%	91.2%	100.0%	82.4%	80.3%	97.3%	93.1%	88.3%	91.7%
1	15	84.3%	99.2%	90.7%	98.9%	89.9%	69.9%	79.2%	82.4%	87.2%	86.8%
1	16	68.5%	80.5%	65.9%	100.0%	75.2%	66.4%	74.1%	78.1%	64.8%	74.8%
1	17	93.1%	89.9%	82.1%	100.0%	72.5%	78.1%	91.5%	88.3%	89.3%	87.2%
1	18	74.9%	80.8%	50.7%	99.7%	72.5%	56.0%	78.7%	80.5%	62.7%	72.9%
1	19	94.9%	96.5%	78.7%	100.0%	75.7%	78.4%	91.7%	86.9%	62.7%	85.1%
2	1	71.5%	80.3%	66.9%	99.7%	76.0%	56.8%	74.1%	78.7%	67.7%	74.6%
2	2	94.7%	98.4%	86.9%	70.9%	64.5%	57.6%	93.3%	56.8%	56.0%	75.5%
2	3	70.7%	78.4%	69.1%	100.0%	70.1%	56.8%	78.1%	75.5%	65.9%	73.8%
2	4	94.9%	89.3%	82.7%	100.0%	88.8%	78.4%	95.2%	82.1%	80.5%	88.0%
2	5	94.4%	99.7%	90.7%	100.0%	100.0%	74.7%	96.8%	96.0%	86.9%	93.2%
2	6	96.5%	100.0%	100.0%	100.0%	99.2%	69.3%	97.3%	99.5%	93.3%	95.0%
2	7	94.4%	89.1%	82.1%	100.0%	72.5%	75.2%	91.2%	77.6%	62.7%	82.8%
2	8	94.4%	88.5%	91.5%	100.0%	73.9%	84.0%	92.8%	91.7%	61.9%	86.5%
2	9	88.3%	76.3%	71.7%	96.8%	67.2%	67.5%	78.1%	84.3%	53.9%	76.0%
2	10	74.9%	93.6%	74.7%	99.5%	71.2%	65.1%	85.6%	89.1%	79.7%	81.5%
2	11	87.2%	88.0%	68.0%	77.3%	85.3%	79.7%	77.9%	81.3%	79.2%	80.4%
2	12	72.5%	86.7%	66.7%	99.5%	71.5%	56.3%	92.5%	78.4%	60.3%	76.0%
2	13	92.0%	89.9%	93.3%	100.0%	84.0%	62.4%	85.3%	83.2%	61.6%	83.5%
2	14	93.9%	99.2%	91.2%	100.0%	82.4%	80.3%	97.3%	93.1%	88.3%	91.7%
2	15	84.3%	99.2%	90.7%	98.9%	89.9%	69.9%	79.2%	82.4%	87.2%	86.8%
2	16	68.5%	80.5%	65.9%	100.0%	75.2%	66.4%	74.1%	78.1%	64.8%	74.8%
2	17	93.1%	89.9%	82.1%	100.0%	72.5%	78.1%	91.5%	88.3%	89.3%	87.2%

Table 7 (continued)

Setup		Performance									
Run	Circuit	1a	1b	1c	2a	2b	2c	3a	3b	3c	Avg.
2	18	69.6%	80.8%	50.7%	99.7%	72.5%	56.0%	78.7%	80.5%	62.7%	72.4%
2	19	94.9%	96.5%	78.7%	100.0%	75.7%	78.4%	91.7%	86.9%	62.7%	85.1%
3	1	71.5%	80.3%	66.9%	99.7%	76.0%	56.8%	74.1%	78.7%	67.7%	74.6%
3	2	94.7%	98.4%	86.9%	70.9%	64.5%	57.6%	93.3%	56.8%	56.0%	75.5%
3	3	70.7%	78.4%	69.1%	100.0%	70.1%	56.8%	78.1%	75.5%	65.9%	73.8%
3	4	94.9%	89.3%	82.7%	100.0%	88.8%	78.4%	95.2%	82.1%	80.5%	88.0%
3	5	94.4%	99.7%	90.7%	100.0%	100.0%	74.7%	96.8%	96.0%	86.9%	93.2%
3	6	96.5%	100.0%	100.0%	100.0%	99.2%	69.3%	97.3%	99.5%	93.3%	95.0%
3	7	94.4%	89.1%	82.1%	100.0%	72.5%	75.2%	91.2%	77.6%	62.7%	82.8%
3	8	94.4%	88.5%	91.5%	100.0%	73.9%	84.0%	92.8%	91.7%	61.9%	86.5%
3	9	88.3%	76.3%	71.7%	96.8%	67.2%	67.5%	78.1%	84.3%	53.9%	76.0%
3	10	74.9%	93.6%	74.7%	99.5%	71.2%	65.1%	85.6%	89.1%	79.7%	81.5%
3	11	87.2%	88.0%	68.0%	77.3%	85.3%	79.7%	77.9%	81.3%	79.2%	80.4%
3	12	72.5%	86.7%	66.7%	99.5%	71.5%	56.3%	92.5%	78.4%	60.3%	76.0%
3	13	92.0%	89.9%	93.3%	100.0%	84.0%	62.4%	85.3%	83.2%	61.6%	83.5%
3	14	93.9%	99.2%	91.2%	100.0%	82.4%	80.3%	97.3%	93.1%	88.3%	91.7%
3	15	84.3%	99.2%	90.7%	98.9%	89.9%	69.9%	79.2%	82.4%	87.2%	86.8%
3	16	68.5%	80.5%	65.9%	100.0%	75.2%	66.4%	74.1%	78.1%	64.8%	74.8%
3	17	93.1%	89.9%	82.1%	100.0%	72.5%	78.1%	91.5%	88.3%	89.3%	87.2%
3	18	74.9%	80.8%	50.7%	99.7%	72.5%	56.0%	78.7%	80.5%	62.7%	72.9%
3	19	94.9%	96.5%	78.7%	100.0%	75.7%	78.4%	91.7%	86.9%	62.7%	85.1%

Table 8 Test results — Adam optimizer, L2 Loss, 2 layers

Setup		Performance									
Run	Circuit	1a	1b	1c	2a	2b	2c	3a	3b	3c	Avg.
1	1	74.7%	84.3%	70.1%	100.0%	76.8%	53.6%	92.0%	80.3%	57.3%	76.6%
1	2	85.3%	86.1%	69.3%	87.5%	74.9%	80.5%	88.5%	77.3%	81.1%	81.2%
1	3	93.1%	86.9%	83.7%	100.0%	74.4%	54.7%	92.3%	83.7%	66.9%	81.7%
1	4	94.9%	96.5%	80.8%	100.0%	98.9%	90.9%	95.7%	91.2%	76.3%	91.7%
1	5	93.9%	100.0%	84.3%	100.0%	100.0%	99.5%	97.6%	99.2%	96.5%	96.8%
1	6	95.2%	99.5%	99.7%	100.0%	99.7%	98.1%	98.9%	96.0%	92.3%	97.7%
1	7	92.8%	99.7%	81.6%	100.0%	80.5%	95.5%	92.3%	86.1%	86.7%	90.6%
1	8	95.7%	99.7%	97.6%	100.0%	96.3%	83.5%	97.3%	97.9%	88.5%	95.2%
1	9	92.0%	81.3%	84.8%	98.7%	64.8%	57.9%	78.1%	79.5%	55.2%	76.9%
1	10	83.7%	95.7%	74.9%	100.0%	81.9%	69.6%	93.1%	85.6%	76.0%	84.5%
1	11	96.8%	99.2%	99.2%	99.7%	98.9%	80.0%	98.7%	95.5%	90.4%	95.4%
1	12	96.0%	100.0%	96.8%	99.7%	94.1%	70.9%	93.9%	95.2%	86.4%	92.6%
1	13	92.3%	100.0%	97.9%	100.0%	98.4%	96.0%	98.4%	94.4%	93.1%	96.7%
1	14	96.3%	100.0%	98.7%	100.0%	99.5%	92.3%	97.1%	97.1%	94.1%	97.2%
1	15	93.6%	99.7%	98.1%	100.0%	98.9%	89.3%	92.5%	96.3%	89.6%	95.3%
1	16	94.9%	99.2%	87.2%	100.0%	83.7%	58.7%	91.5%	80.3%	59.7%	83.9%
1	17	94.1%	99.7%	97.1%	100.0%	85.6%	84.0%	93.3%	78.9%	66.1%	88.8%
1	18	96.5%	93.1%	79.5%	100.0%	69.1%	72.8%	92.5%	85.3%	66.4%	83.9%
1	19	94.9%	99.7%	97.9%	100.0%	99.7%	96.8%	97.6%	96.5%	90.9%	97.1%

Table 8 (continued)

Setup		Performance									
Run	Circuit	1a	1b	1c	2a	2b	2c	3a	3b	3c	Avg.
2	1	74.7%	84.3%	70.1%	100.0%	76.8%	53.6%	92.0%	80.3%	57.3%	76.6%
2	2	85.3%	86.1%	69.3%	87.5%	74.9%	80.5%	88.5%	77.3%	81.1%	81.2%
2	3	93.1%	86.9%	83.7%	100.0%	74.4%	54.7%	92.3%	83.7%	66.9%	81.7%
2	4	94.9%	96.5%	80.8%	100.0%	98.9%	90.9%	95.7%	91.2%	76.3%	91.7%
2	5	93.9%	100.0%	84.3%	100.0%	100.0%	99.5%	97.6%	99.2%	96.5%	96.8%
2	6	95.2%	99.5%	99.7%	100.0%	99.7%	98.1%	98.9%	96.0%	92.3%	97.7%
2	7	92.8%	99.7%	81.6%	100.0%	80.5%	95.5%	92.3%	86.1%	86.7%	90.6%
2	8	95.7%	99.7%	97.6%	100.0%	96.3%	83.5%	97.3%	97.9%	88.5%	95.2%
2	9	92.0%	81.3%	84.8%	98.7%	64.8%	57.9%	78.1%	79.5%	55.2%	76.9%
2	10	83.7%	95.7%	74.9%	100.0%	81.9%	69.6%	93.1%	85.6%	76.0%	84.5%
2	11	96.8%	99.2%	99.2%	99.7%	98.9%	80.0%	98.7%	95.5%	90.4%	95.4%
2	12	96.0%	100.0%	96.8%	99.7%	94.1%	70.9%	93.9%	95.2%	86.4%	92.6%
2	13	92.3%	100.0%	97.9%	100.0%	98.4%	96.0%	98.4%	94.4%	93.1%	96.7%
2	14	95.5%	100.0%	100.0%	100.0%	99.5%	97.3%	96.8%	94.9%	96.3%	97.8%
2	15	93.6%	99.7%	98.1%	100.0%	98.9%	89.3%	92.5%	96.3%	89.6%	95.3%
2	16	94.9%	99.2%	87.2%	100.0%	83.7%	58.7%	91.5%	80.3%	59.7%	83.9%
2	17	94.1%	99.7%	97.1%	100.0%	85.6%	84.0%	93.3%	78.9%	66.1%	88.8%
2	18	96.5%	93.1%	79.5%	100.0%	69.1%	72.8%	92.5%	85.3%	66.4%	83.9%
2	19	94.9%	99.7%	97.9%	100.0%	99.7%	96.8%	97.6%	96.5%	90.9%	97.1%
3	1	74.7%	84.3%	70.1%	100.0%	76.8%	53.6%	92.0%	80.3%	57.3%	76.6%
3	2	85.3%	86.1%	69.3%	87.5%	74.9%	80.5%	88.5%	77.3%	81.1%	81.2%
3	3	93.1%	86.9%	83.7%	100.0%	74.4%	54.7%	92.3%	83.7%	66.9%	81.7%
3	4	94.9%	96.5%	80.8%	100.0%	98.9%	90.9%	95.7%	91.2%	76.3%	91.7%
3	5	93.9%	100.0%	84.3%	100.0%	100.0%	99.5%	97.6%	99.2%	96.5%	96.8%
3	6	95.2%	99.5%	99.7%	100.0%	99.7%	98.1%	98.9%	96.0%	92.3%	97.7%
3	7	92.8%	99.7%	81.6%	100.0%	80.5%	95.5%	92.3%	86.1%	86.7%	90.6%
3	8	95.7%	99.7%	97.6%	100.0%	96.3%	83.5%	97.3%	97.9%	88.5%	95.2%
3	9	92.0%	81.3%	84.8%	98.7%	64.8%	57.9%	78.1%	79.5%	55.2%	76.9%
3	10	83.7%	95.7%	74.9%	100.0%	81.9%	69.6%	93.1%	85.6%	76.0%	84.5%
3	11	96.8%	99.2%	99.2%	99.7%	98.9%	80.0%	98.7%	95.5%	90.4%	95.4%
3	12	96.0%	100.0%	96.8%	99.7%	94.1%	70.9%	93.9%	95.2%	86.4%	92.6%
3	13	92.3%	100.0%	97.9%	100.0%	98.4%	96.0%	98.4%	94.4%	93.1%	96.7%
3	14	94.1%	100.0%	98.4%	100.0%	99.2%	89.3%	99.2%	97.1%	92.8%	96.7%
3	15	93.6%	99.7%	98.1%	100.0%	98.9%	89.3%	92.5%	96.3%	89.6%	95.3%
3	16	94.9%	99.2%	87.2%	100.0%	83.7%	58.7%	91.5%	80.3%	59.7%	83.9%
3	17	94.1%	99.7%	97.1%	100.0%	85.6%	84.0%	93.3%	78.9%	66.1%	88.8%
3	19	94.9%	99.7%	97.9%	100.0%	99.7%	96.8%	97.6%	96.5%	90.9%	97.1%

Appendix 4: Hyperparameter search results

Table 9 Hyperparameter search results — Adam optimizer, L1 loss

Setup		Performance									
Layers	Circuit	1a	1b	1c	2a	2b	2c	3a	3b	3c	Avg.
1	1	63.2%	82.1%	74.7%	96.8%	71.2%	54.7%	71.7%	81.1%	51.2%	71.9%
1	2	92.5%	77.9%	70.1%	48.8%	53.9%	52.8%	79.2%	46.4%	49.9%	63.5%
1	3	76.0%	77.6%	73.1%	98.4%	72.8%	54.9%	72.3%	77.9%	57.6%	73.4%
1	4	93.6%	90.7%	81.3%	51.7%	83.5%	84.0%	86.9%	77.9%	76.0%	80.6%
1	5	90.7%	95.7%	93.9%	90.7%	82.4%	49.3%	94.4%	89.9%	77.1%	84.9%
1	6	87.2%	91.5%	89.6%	97.9%	91.7%	93.6%	93.3%	86.1%	86.1%	90.8%
1	7	88.5%	86.9%	79.5%	99.7%	66.7%	59.5%	83.2%	79.5%	80.3%	80.4%
1	8	93.3%	92.0%	88.5%	98.4%	87.5%	76.8%	82.4%	82.7%	71.2%	85.9%
1	9	78.4%	62.1%	85.6%	86.1%	71.2%	57.1%	74.4%	81.9%	56.3%	72.6%
1	11	69.1%	78.7%	53.9%	80.0%	83.7%	73.9%	83.5%	80.5%	76.0%	75.5%
1	12	77.1%	87.7%	63.7%	91.5%	82.4%	56.8%	67.2%	83.5%	46.7%	72.9%
1	13	93.6%	91.2%	84.3%	97.9%	84.5%	73.3%	86.7%	81.1%	78.1%	85.6%
1	14	89.6%	97.6%	89.3%	92.8%	94.1%	73.9%	89.6%	90.7%	77.3%	88.3%
1	15	87.2%	90.7%	79.5%	84.8%	77.3%	63.2%	84.0%	88.0%	81.9%	81.8%
1	16	62.7%	82.1%	66.4%	100.0%	72.3%	55.7%	74.4%	77.6%	48.8%	71.1%
1	17	94.4%	76.3%	83.7%	95.7%	79.7%	63.5%	85.9%	82.7%	79.5%	82.4%
1	18	73.9%	82.4%	61.6%	97.9%	70.9%	61.1%	81.3%	84.0%	48.8%	73.5%
1	19	74.9%	90.4%	83.5%	99.7%	70.1%	64.8%	89.6%	83.7%	63.7%	80.1%
2	1	63.5%	70.1%	62.9%	94.4%	72.3%	57.1%	84.0%	80.8%	50.7%	70.6%
2	2	64.5%	89.6%	82.7%	87.2%	86.4%	80.8%	80.5%	77.3%	71.7%	80.1%
2	3	90.7%	76.3%	85.6%	87.7%	84.5%	49.9%	79.7%	80.8%	72.5%	78.6%
2	4	88.3%	91.2%	93.1%	89.9%	85.3%	70.4%	87.5%	88.8%	66.1%	84.5%
2	5	93.6%	93.9%	88.3%	93.9%	89.3%	93.6%	90.1%	90.7%	84.0%	90.8%
2	6	92.8%	90.4%	89.3%	99.7%	98.1%	85.6%	90.7%	88.8%	80.8%	90.7%
2	7	90.1%	90.7%	89.3%	94.9%	72.0%	64.8%	90.4%	84.8%	78.1%	83.9%
2	8	93.6%	97.6%	85.6%	99.2%	88.3%	68.8%	88.5%	84.8%	79.2%	87.3%
2	9	85.3%	61.3%	86.4%	93.3%	72.8%	67.7%	85.1%	72.0%	66.4%	76.7%
2	11	87.7%	93.6%	86.1%	95.7%	89.6%	83.7%	85.6%	85.9%	88.0%	88.4%
2	12	89.6%	93.3%	84.3%	97.3%	82.1%	82.4%	88.0%	75.5%	78.4%	85.7%
2	13	86.1%	98.9%	87.2%	99.7%	91.5%	89.1%	91.7%	73.1%	82.1%	88.8%
2	14	92.0%	85.1%	92.3%	100.0%	92.3%	81.1%	86.9%	88.8%	86.4%	89.4%
2	15	79.7%	92.0%	93.9%	97.6%	84.3%	80.5%	83.7%	82.7%	76.8%	85.7%
2	16	85.1%	73.6%	82.7%	97.3%	85.1%	71.7%	86.1%	78.9%	58.9%	79.9%
2	17	93.6%	93.3%	80.8%	98.4%	88.8%	74.1%	92.5%	92.8%	64.8%	86.6%
2	18	91.2%	88.8%	86.9%	99.5%	87.5%	56.8%	85.1%	82.7%	73.3%	83.5%
2	19	89.6%	96.0%	89.3%	98.4%	92.8%	83.7%	88.5%	92.0%	83.2%	90.4%

Table 10 Hyperparameter search results — Adam optimizer, L2 loss

Setup		Performance									
Layers	Circuit	1a	1b	1c	2a	2b	2c	3a	3b	3c	Avg.
1	1	74.7%	77.3%	65.1%	100.0%	75.2%	55.2%	76.3%	76.0%	66.4%	74.0%
1	2	95.5%	99.2%	83.5%	72.0%	57.6%	59.2%	92.5%	57.1%	65.6%	75.8%
1	3	68.8%	80.0%	51.2%	100.0%	71.7%	57.1%	90.9%	80.5%	64.8%	73.9%
1	4	95.5%	91.7%	85.1%	100.0%	74.7%	70.1%	88.0%	89.1%	64.0%	84.2%
1	5	96.0%	99.2%	93.3%	100.0%	87.2%	88.0%	98.4%	89.3%	89.1%	93.4%
1	6	96.0%	100.0%	97.1%	100.0%	100.0%	78.9%	97.9%	97.1%	97.1%	96.0%
1	7	96.8%	85.9%	82.4%	100.0%	70.4%	68.8%	94.1%	80.5%	65.6%	82.7%
1	8	96.5%	97.6%	82.7%	99.7%	71.5%	82.9%	93.6%	87.5%	65.6%	86.4%
1	9	86.4%	80.3%	90.4%	97.1%	64.3%	55.2%	78.9%	83.5%	54.9%	76.8%
1	11	82.7%	82.4%	57.6%	93.6%	83.5%	84.5%	94.1%	81.6%	80.3%	82.3%
1	12	78.4%	81.9%	70.7%	100.0%	80.0%	57.1%	92.3%	80.0%	59.5%	77.7%
1	13	95.2%	85.1%	92.3%	100.0%	81.6%	62.9%	94.4%	80.5%	69.1%	84.6%
1	14	95.2%	100.0%	93.1%	100.0%	98.9%	77.1%	93.3%	85.3%	91.7%	92.7%
1	15	72.5%	86.1%	93.9%	100.0%	68.3%	65.9%	87.7%	91.5%	78.7%	82.7%
1	16	70.9%	78.1%	63.7%	100.0%	70.1%	53.3%	78.9%	81.1%	67.2%	73.7%
1	17	93.3%	98.4%	95.7%	100.0%	84.5%	86.1%	83.2%	83.7%	70.1%	88.4%
1	18	69.6%	74.9%	54.4%	100.0%	73.6%	61.1%	77.3%	81.9%	56.5%	72.1%
1	19	95.7%	86.9%	78.1%	100.0%	85.1%	56.5%	89.9%	86.7%	62.4%	82.4%
2	1	83.7%	83.2%	68.3%	100.0%	69.3%	57.1%	78.1%	80.5%	65.1%	76.1%
2	2	89.3%	91.2%	98.7%	80.0%	85.1%	82.9%	88.8%	82.9%	65.3%	84.9%
2	3	96.8%	90.1%	82.9%	100.0%	73.3%	70.4%	93.6%	86.7%	72.8%	85.2%
2	4	93.9%	98.4%	81.9%	100.0%	95.5%	94.4%	95.2%	83.5%	63.7%	89.6%
2	5	96.0%	99.7%	98.9%	100.0%	100.0%	89.9%	98.1%	98.1%	93.9%	97.2%
2	6	96.3%	100.0%	98.1%	100.0%	100.0%	94.1%	98.9%	97.3%	93.1%	97.5%
2	7	96.0%	99.2%	92.3%	100.0%	98.7%	84.5%	94.1%	85.6%	94.7%	93.9%
2	8	96.5%	92.8%	89.9%	100.0%	99.5%	84.5%	95.2%	87.5%	87.5%	92.6%
2	9	88.0%	88.5%	91.2%	97.1%	72.3%	77.1%	81.9%	80.3%	67.5%	82.6%
2	11	95.5%	100.0%	99.5%	100.0%	88.3%	74.1%	98.1%	96.5%	88.8%	93.4%
2	12	94.1%	98.1%	80.5%	100.0%	94.9%	69.6%	97.1%	96.3%	87.5%	90.9%
2	13	95.7%	100.0%	99.5%	100.0%	98.7%	93.9%	96.5%	94.9%	94.1%	97.0%
2	14	95.7%	100.0%	98.1%	99.7%	100.0%	92.3%	96.8%	95.7%	93.1%	96.8%
2	15	94.7%	97.9%	98.1%	100.0%	98.9%	73.1%	97.3%	96.0%	87.2%	93.7%
2	16	95.5%	89.6%	81.3%	100.0%	73.1%	56.0%	84.8%	87.7%	61.3%	81.0%
2	17	95.2%	99.2%	81.3%	99.2%	73.6%	77.1%	94.4%	93.6%	90.7%	89.4%
2	18	94.4%	94.4%	83.5%	100.0%	86.4%	54.7%	90.9%	85.9%	71.5%	84.6%
2	19	95.2%	100.0%	95.5%	100.0%	99.5%	85.1%	97.3%	90.9%	87.5%	94.5%

Table 11 Hyperparameter search results — Gradient Decent optimizer, L1 loss

Setup		Performance									
Layers	Circuit	1a	1b	1c	2a	2b	2c	3a	3b	3c	Avg.
1	1	71.7%	78.1%	48.3%	100.0%	73.6%	48.0%	75.7%	79.2%	50.1%	69.4%
1	2	95.5%	93.6%	85.1%	50.9%	60.0%	49.6%	76.8%	52.8%	46.1%	67.8%
1	3	73.6%	81.3%	52.5%	100.0%	71.5%	54.1%	78.9%	78.4%	49.3%	71.1%
1	4	86.9%	93.1%	69.6%	96.5%	70.1%	53.6%	80.8%	78.7%	48.5%	75.3%
1	5	93.3%	89.1%	76.5%	98.4%	74.4%	54.1%	78.4%	78.4%	45.9%	76.5%
1	6	93.3%	94.9%	88.5%	100.0%	79.2%	68.3%	85.9%	80.0%	64.0%	83.8%
1	7	85.9%	82.1%	74.4%	100.0%	70.4%	56.5%	79.2%	78.7%	52.8%	75.6%
1	8	88.5%	81.9%	73.3%	99.7%	74.4%	60.0%	78.9%	79.5%	52.3%	76.5%
1	9	88.8%	80.3%	75.7%	88.0%	69.3%	52.5%	76.0%	82.1%	52.5%	73.9%
1	11	50.4%	89.3%	53.9%	82.9%	76.8%	72.3%	76.5%	78.9%	75.7%	73.0%
1	12	67.7%	86.1%	58.4%	100.0%	78.9%	50.1%	79.5%	67.2%	49.6%	70.8%
1	13	86.1%	85.3%	73.3%	100.0%	69.6%	56.5%	79.7%	79.5%	55.2%	76.1%
1	14	88.8%	87.5%	82.9%	91.5%	78.7%	48.8%	78.4%	79.7%	47.5%	76.0%
1	15	79.5%	85.1%	88.8%	79.2%	68.3%	63.5%	89.1%	73.9%	48.8%	75.1%
1	16	72.5%	79.2%	47.7%	100.0%	73.6%	48.5%	75.7%	79.5%	50.1%	69.7%
1	17	81.3%	89.1%	69.9%	97.3%	74.9%	54.7%	79.2%	78.1%	50.7%	75.0%
1	18	73.6%	78.9%	52.5%	100.0%	71.5%	52.5%	78.9%	78.7%	49.3%	70.7%
1	19	85.3%	89.1%	75.7%	98.9%	74.1%	56.8%	78.7%	79.2%	54.1%	76.9%
2	1	73.3%	82.4%	50.9%	100.0%	74.7%	47.5%	75.2%	79.2%	50.4%	70.4%
2	2	81.9%	89.9%	95.7%	54.4%	78.1%	47.7%	68.3%	77.6%	79.2%	74.8%
2	3	88.5%	85.3%	65.9%	99.5%	72.0%	53.6%	78.9%	78.4%	55.7%	75.3%
2	4	91.5%	97.3%	75.2%	96.3%	71.5%	56.8%	79.5%	78.7%	48.5%	77.2%
2	5	94.7%	85.6%	83.2%	99.7%	81.3%	59.5%	81.6%	78.9%	49.9%	79.4%
2	6	92.0%	97.1%	82.1%	100.0%	83.5%	74.9%	88.5%	86.7%	58.9%	84.9%
2	7	93.6%	82.1%	81.3%	96.3%	69.1%	45.6%	78.9%	79.2%	51.7%	75.3%
2	8	91.7%	79.5%	88.3%	94.4%	76.5%	63.5%	78.9%	77.1%	48.8%	77.6%
2	9	87.5%	78.7%	70.4%	83.2%	69.9%	64.0%	66.1%	79.5%	44.0%	71.5%
2	11	76.8%	98.1%	89.1%	89.9%	88.5%	76.5%	80.3%	88.8%	74.1%	84.7%
2	12	88.3%	89.6%	84.8%	98.4%	84.3%	53.6%	86.9%	81.6%	58.7%	80.7%
2	13	92.8%	88.5%	78.1%	99.7%	75.7%	55.7%	83.7%	78.9%	51.5%	78.3%
2	14	92.8%	86.7%	89.1%	100.0%	76.0%	62.1%	81.3%	79.7%	53.1%	80.1%
2	15	88.8%	81.3%	89.3%	87.5%	84.5%	79.5%	85.3%	74.4%	53.3%	80.4%
2	16	76.5%	83.7%	74.7%	100.0%	76.0%	51.7%	75.7%	79.5%	53.6%	74.6%
2	17	92.8%	82.1%	82.9%	92.3%	74.7%	62.4%	74.9%	79.5%	52.5%	77.1%
2	18	93.1%	83.7%	76.5%	98.7%	69.1%	51.5%	78.4%	78.4%	55.5%	76.1%
2	19	90.9%	92.3%	85.3%	97.1%	74.1%	50.1%	80.3%	77.9%	51.2%	77.7%

Table 12 Hyperparameter search results — Gradient Decent optimizer, L2 loss

Setup		Performance									
Layers	Circuit	1a	1b	1c	2a	2b	2c	3a	3b	3c	Avg.
1	1	69.1%	80.5%	50.9%	100.0%	74.9%	51.7%	74.4%	78.4%	52.0%	70.2%
1	2	94.1%	97.9%	80.0%	56.0%	60.0%	50.7%	75.5%	59.2%	48.0%	69.0%
1	3	71.7%	78.4%	51.2%	100.0%	69.6%	54.4%	79.5%	78.4%	51.2%	70.5%
1	4	94.7%	96.3%	78.7%	100.0%	74.1%	70.7%	91.2%	78.7%	61.1%	82.8%
1	5	96.0%	96.8%	90.7%	99.7%	88.3%	60.8%	96.0%	82.1%	62.9%	85.9%
1	6	95.2%	98.1%	88.5%	100.0%	99.2%	70.7%	93.9%	83.7%	76.8%	89.6%
1	7	95.5%	85.1%	79.7%	100.0%	70.9%	61.9%	78.9%	78.9%	53.9%	78.3%
1	8	89.6%	96.5%	82.1%	100.0%	77.3%	55.7%	83.2%	79.5%	53.3%	79.7%
1	9	89.3%	61.1%	78.9%	99.7%	64.5%	62.7%	80.0%	79.5%	51.7%	74.2%
1	11	55.2%	93.6%	77.1%	92.3%	81.3%	81.9%	79.7%	70.4%	70.7%	78.0%
1	12	74.1%	85.9%	66.7%	100.0%	78.4%	60.3%	78.9%	78.4%	48.8%	74.6%
1	13	96.3%	83.7%	80.5%	100.0%	73.1%	54.9%	85.9%	77.9%	52.3%	78.3%
1	14	93.6%	88.5%	97.9%	99.7%	77.3%	55.5%	80.0%	79.5%	55.2%	80.8%
1	15	81.1%	80.3%	91.5%	84.5%	92.3%	62.9%	91.5%	90.9%	54.7%	81.1%
1	16	69.1%	80.5%	51.2%	100.0%	74.9%	49.1%	74.4%	78.4%	50.4%	69.8%
1	17	92.5%	90.9%	80.8%	99.5%	79.2%	64.5%	80.8%	79.5%	55.7%	80.4%
1	18	71.7%	78.7%	51.2%	100.0%	69.6%	57.1%	79.5%	78.4%	51.2%	70.8%
1	19	94.7%	96.8%	86.1%	100.0%	74.9%	56.3%	79.7%	78.9%	59.5%	80.8%
2	1	75.5%	83.7%	47.2%	100.0%	76.0%	50.7%	74.4%	80.3%	51.7%	71.1%
2	2	85.6%	91.2%	97.9%	88.5%	87.2%	81.1%	67.5%	74.1%	81.6%	83.9%
2	3	93.6%	86.9%	80.5%	100.0%	70.9%	62.4%	80.8%	78.9%	63.7%	79.8%
2	4	96.0%	99.5%	84.3%	99.7%	71.2%	55.7%	87.2%	78.1%	59.5%	81.2%
2	5	96.0%	99.7%	97.1%	100.0%	96.0%	81.1%	95.5%	92.8%	78.4%	92.9%
2	6	94.9%	98.9%	97.3%	100.0%	99.7%	81.3%	96.8%	82.9%	86.4%	93.2%
2	7	96.0%	91.5%	81.1%	98.9%	72.3%	61.3%	92.8%	79.2%	53.1%	80.7%
2	8	93.9%	86.1%	88.0%	100.0%	75.7%	69.3%	93.3%	78.9%	61.9%	83.0%
2	9	86.4%	81.6%	79.2%	84.5%	74.1%	62.7%	77.6%	78.7%	55.5%	75.6%
2	11	93.1%	97.6%	98.7%	99.5%	92.3%	61.1%	96.8%	94.9%	83.5%	90.8%
2	12	97.1%	96.3%	84.8%	99.7%	79.7%	53.9%	91.2%	92.5%	61.9%	84.1%
2	13	95.5%	91.7%	81.1%	99.7%	76.5%	65.6%	92.8%	76.3%	64.8%	82.7%
2	14	95.5%	97.9%	97.9%	100.0%	76.5%	57.3%	95.5%	80.0%	83.5%	87.1%
2	15	90.9%	98.4%	84.3%	99.2%	89.6%	79.2%	88.3%	81.1%	72.5%	87.1%
2	16	93.6%	86.4%	70.1%	100.0%	75.2%	49.3%	75.5%	80.8%	54.4%	76.1%
2	17	94.4%	97.9%	92.0%	100.0%	78.7%	57.3%	83.5%	80.3%	60.0%	82.7%
2	18	96.5%	86.1%	80.3%	100.0%	70.7%	55.7%	80.0%	79.2%	64.3%	79.2%
2	19	92.5%	88.3%	88.5%	100.0%	74.9%	69.3%	89.3%	78.1%	59.7%	82.3%

Acknowledgements The authors would like to thank Sukin Sim for providing the data points in Table 5 and valuable feedback; Nicola Pancotti and Christoph Segler for providing feedback on our statistical approach; Jonas Haferkamp and Jordi Brugués for discussions on measures of expressibility and t-designs; and Ryan Sweke for discussions on function classes. TH has been supported by the BMWi (PlanQK).

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Abadi M (2015) TensorFlow: Large-scale machine learning on heterogeneous systems. http://tensorflow.org/, Software available from tensorflow.org

Abraham H (2019) Qiskit: An open-source framework for quantum computing

Arute F, Arya K, Babbush R, Bacon D, Bardin JC, Barends R, Biswas R, et al. (2019) Quantum supremacy using a programmable superconducting processor. Nature 574(7779):505–510

Bergholm V, Izaac J, Schuld M, Gogolin C, Blank C, McKiernan K, Killoran N (2018) Pennylane. arXiv:1811.04968

Boddy R, Laird Smith G (2009) Statistical methods in practice: for scientists and technologists. Wiley, Chichester, UK

Bose RC (1947) Mathematical theory of the symmetrical factorial design. SankhyÄ: The Indian Journal of Statistics: 107–166

Bravyi S, Gosset D, Koenig R, Tomamichel M (2020) Quantum advantage with noisy shallow circuits. Nature Physics 16(10): 1040–1045

Coyle B, Mills D, Danos V, Kashefi E (2020) The born supremacy: Quantum advantage and training of an ising born machine. npj Quantum Information 6(1):1–11

Dancey CP, Reidy J (2007) Statistics without maths for psychology. Pearson education

Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. IEEE conference on computer vision and pattern recognition pp. 248-255

Deng L (2012) The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine 29(6):141–142

Farhi E, Goldstone J, Gutmann S (2014) A quantum approximate optimization algorithm. arXiv:1411.4028

Farhi E, Neven H (2018) Classification with quantum neural networks on near term processors

Goldbloom A, Hamner B (2020) Kaggle. https://www.kaggle.com/

Havlicek V, Córcoles AD, Temme K, Harrow AW, Kandala A, Chow JM, Gambetta JM (2018) Supervised learning with quantum enhanced feature spaces. Nature 567:209–212. https://doi.org/10.1038/s41586-019-0980-2, arXiv:1804.11326

Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow JM, Gambetta JM (2017) Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. Nature 549(7671):242–246

KutinS (2016) qpic. https://github.com/qpic/qpic

Marrero CO, Kieferová M, Wiebe N (2020) Entanglement induced barren plateaus. arXiv:2010.15968

McClean JR, Romero J, Babbush R, Aspuru-Guzik A (2016) The theory of variational hybrid quantum-classical algorithms. New J Phys 18:023023

Meyer DA, Wallach NR (2002) Global entanglement in multiparticle systems. Journal of Mathematical Physics 43(9):4273–4278

Mirjalili SZMH, Sardroudi HM (2012) Stochastic gradient descent for hybrid quantum-classical optimization. Applied Mathematics and Computation 218(22):11125–11137

Nielsen MA, Chuang I (2002) Quantum computation and quantum information. Cambridge University Press, Cambridge

Peruzzo A, McClean J, Shadbolt P, Yung M-H, Zhou X-Q, Love PJ, Aspuru-Guzik A, O'Brien JL (2014) A variational eigenvalue solver on a photonic quantum processor. Nature communications 5:4213

Preskill J (2018) Quantum computing in the nisq era and beyond. Quantum 2:79

Pérez-Salinas A, Cervera-Lierta A, Gil-Fuster E, Latorre JI (2020) Data re-uploading for a universal quantum classifier. Quantum 4:226

Riste D, da Silva MP, Ryan CA, Cross AW, Córcoles AD, Smolin JA, Gambetta JM, Chow JM, Johnson BR (2017) Demonstration of quantum advantage in machine learning. NPJ Quantum Information 3(1):1–5

Schuld M, Bergholm V, Gogolin C, Izaac J, Killoran N (2019) Evaluating analytic gradients on quantum hardware. Physical Review A 99(3):032331

Schuld M, Bocharov A, Svore K, Wiebe N (2018) Circuit-centric quantum classifiers

Schuld M, Killoran N (2019) Quantum machine learning in feature hilbert spaces. Physical review letters 122(4):040504

Schuld M, Sweke R, Meyer JJ (2020) The effect of data encoding on the expressive power of variational quantum machinelearning models. arXiv:2008.08605

Sim S, Johnson PD, Aspuru-Guzik A (2019) Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. https://doi.org/10.1002/qute.201900070, arXiv:1905.10876

Stoudenmire EM, Schwab DJ (2016) Supervised learning with quantum-inspired tensor networks. arXiv:1605.05775

Sweke R, Wilde F, Meyer J, Schuld M, Fährmann PK, Meynard-Piganeau B, Eisert J (2019) Stochastic gradient descent for hybrid quantum-classical optimization. arXiv:1910.01155

Vidal JG, Theis DO (2019) Input redundancy for parameterized quantum circuits. arXiv:1901.11434

Życzkowski K, Sommers H-J (2005) Average fidelity between random quantum states. Phys Rev A 71:032313