Quantum Machine Intelligence (2023) 5:34
https://doi.org/10.1007/s42484-023-00118-z

RESEARCH ARTICLE

Classical splitting of parametrized quantum circuits

Cenk Tüysüz[1,2] · Giuseppe Clemente[1] · Arianna Crippa[1,2] · Tobias Hartung[3] · Stefan Kühn[1,4] · Karl Jansen[1,4]

Received: 28 February 2023 / Accepted: 9 June 2023 / Published online: 1 August 2023

Abstract

Barren plateaus appear to be a major obstacle for using variational quantum algorithms to simulate large-scale quantum systems or to replace traditional machine learning algorithms. They can be caused by multiple factors such as the expressivity of the ansatz, excessive entanglement, the locality of observables under consideration, or even hardware noise. We propose classical splitting of parametric ansatz circuits to avoid barren plateaus. Classical splitting is realized by subdividing an N qubit ansatz into multiple ansätze that consist of $\mathcal{O}(\log N)$ qubits. We show that such an approach allows for avoiding barren plateaus and carry out numerical experiments, and perform binary classification on classical and quantum datasets. Moreover, we propose an extension of the ansatz that is compatible with variational quantum simulations. Finally, we discuss a speed-up for gradient-based optimization and hardware implementation, robustness against noise and parallelization, making classical splitting an ideal tool for noisy intermediate scale quantum (NISQ) applications.

Keywords Parametric · Ansatz · Barren · Plateau

1 Introduction

Variational quantum algorithms (VQAs) [10] are a promising approach to solve a wide range of problems, such as finding the ground state of a given hamiltonian via the variational quantum eigensolver (VQE) [35], solving combinatorial optimization problems with the quantum approximate optimization algorithm (QAOA) [14] or solving classification problems using quantum neural networks [15].

Variational quantum algorithms are suitable for noisy intermediate scale quantum (NISQ) [37] hardware as they can be implemented with a small number of layers and gates for simple tasks. However, a scalability problem arises with the increasing number of qubits, hindering a possible advantage. Variational quantum algorithms rely on a feedback loop between a classical computer and a quantum device. The former is used to update the parameters of the ansatz conditioned on the measurement outcome obtained from the quantum hardware. This procedure is iterated until convergence. Classical optimizers use the information on the cost landscape of the parametric ansatz to find the minimum. The updates on the parameters move the ansatz to a lower point on the cost surface. In 2018, McClean et al. showed that for a wide range of ansätze the cost landscape flattens with the increasing number of qubits, making it exponentially harder to find the solution for the optimizer [26]. The flattening was first observed by looking at the distribution of gradients across the parameter space, and the problem was named barren plateaus (BPs). A variational quantum algorithm is said to have a BP if its gradients decay exponentially with respect to one of its hyper-parameters, such as the number of qubits or layers.

Since the discovery of the BP problem, there has been significant progress that improved our understanding of what causes barren plateaus, and several methods to avoid them have been proposed. It has been shown that noise [47], entanglement [28], and the locality of the observable [11] play an essential role for determining whether an ansatz will exhibit barren plateaus. It has also been shown that the choice of ansatz (e.g. its expressivity) for the circuit is one of the decisive factors that impact barren plateaus [20]. For instance, the absence of barren plateaus has been shown for

✉ Cenk Tüysüz
cenk.tueysuez@desy.de

[1] Deutsches Elektronen-Synchrotron DESY,
Platanenallee 6, 15738 Zeuthen, Germany

[2] Institut für Physik, Humboldt-Universität zu Berlin,
Newtonstr. 15, Berlin 12489, Germany

[3] Northeastern University London, Devon House,
St Katharine Docks, London E1W 1LP, United Kingdom

[4] Computation-Based Science and Technology Research
Center, The Cyprus Institute, 20 Kavafi Street,
Nicosia 2121, Cyprus

quantum convolutional neural networks (QCNN) [12, 36] and tree tensor networks (TTN) [18, 52]. In contrast, the hardware efficient ansatz (HEA) [21, 26, 52] and matrix product states (MPS) [52] have been shown to have barren plateaus.

One of the essential discoveries showed that barren plateaus are equivalent to cost concentration and narrow gorges [3]. This implies that barren plateaus are not only a result of the exponentially decaying gradient but also of the cost function itself, and they can be identified by analyzing random points on the cost surface. As a result, gradient-free optimizers are also susceptible to barren plateaus, and do not offer a way to circumvent this problem [2].

Many methods have been suggested to mitigate barren plateaus in the literature. Some of these methods suggest to use different ansätze or cost functions [49, 51], determining a better initial point to start the optimization [17, 24, 39, 50], determining the step size during the optimization based on the ansatz [40], correlating parameters of the ansatz (e.g., restricting the directions of rotation) [20, 46], or combining multiple methods [9, 30].

In this work, we propose a novel idea in which we claim that if any ansatz of N qubits is classically separated to a set of ansätze with $\mathcal{O}(\log N)$ qubits, the new ansatz will not exhibit Barren Plateaus. This work is not the first proposal in the literature that considers partitioning an ansatz. However, our proposal is significantly different. Most work in the literature first considers an ansatz and then emulates the result of that ansatz through many ansätze (exponentially many in general) with less number of qubits (which increases the effective size of quantum simulations) using gate decompositions, entanglement forging, divide and conquer or other methods [8, 13, 16, 25, 32, 34, 41, 44, 44]. On the other hand, this work proposes using ansätze that are classically split, meaning that there are no two-qubit gate operations between the subcircuits before splitting. This way, there is no need for gate decompositions or other computational steps. We also investigate an extension of this ansatz design by combining classically split layers with standard layers. Our results show that this approach provides many benefits such as better trainability, robustness against noise and faster implementation on NISQ devices.

In the remainder of the paper, we start by giving an analytical illustration of the method in Sect. 2. Then, we provide numerical evidence for our claim in Sect. 3 and extend our results to practical use cases by comparing binary classification performance of classical splitting (CS) for classical and quantum data. Next, we propose an extension of the classical splitting ansatz and perform experiments to simulate the ground state of the transversal-field Ising hamiltonian. Finally, we discuss the advantages of employing CS, make comments on future directions in Sect. 4 and give an outlook in Sect. 5.

2 Avoiding Barren Plateaus

Barren plateaus (BPs) can be identified by investigating how the gradients of an ansatz scale with respect to a parameter. Here, we will start with the notation of McClean et al. and extend it to CS [26]. The ansatz is composed of consecutive parametrized (V) and non-parametrized entangling (W) layers. We define $U_l(\theta_l) = \exp(-i\theta_l V_l)$, where V_l is a Hermitian operator and W_l is a generic unitary operator. Then the ansatz can be expressed with a multiplication of layers,

$$U(\boldsymbol{\theta}) = \prod_{l=1}^{L} U_l(\theta_l) W_l. \tag{1}$$

Then, for an observable O and input state of ρ, the cost is given as

$$C(\boldsymbol{\theta}) = \mathrm{Tr}[O U(\boldsymbol{\theta}) \rho U^\dagger(\boldsymbol{\theta})]. \tag{2}$$

The ansatz can be separated into two parts to investigate a certain layer, such that $U_- \equiv \prod_{l=1}^{j-1} U_l(\theta_l) W_l$ and $U_+ \equiv \prod_{l=j}^{L} U_l(\theta_l) W_l$. Then, the gradient of the j^{th} parameter can be expressed as

$$\partial_j C(\boldsymbol{\theta}) = \frac{\partial C(\boldsymbol{\theta})}{\partial \theta_j} = i\,\mathrm{Tr}[[V_j, U_+^\dagger O U_+] U_- \rho U_-^\dagger]. \tag{3}$$

The expected value of the gradients can be computed using the Haar measure. Please see Appendix A for more details on the Haar measure, unitary t-designs and details of the proofs in this section. If we assume the ansatz $U(\theta)$ forms a unitary 2-design, then this implies that $\langle \partial_k C(\boldsymbol{\theta}) \rangle = 0$ [26]. Since the average value of the gradients are centered around zero, the variance of the distribution, which is defined as,

$$\mathrm{Var}[\partial_k C(\boldsymbol{\theta})] = \langle (\partial_k C(\boldsymbol{\theta}))^2 \rangle - \langle \partial_k C(\boldsymbol{\theta}) \rangle^2, \tag{4}$$

can inform us about the size of the gradients. The variance of the gradients of the j^{th} parameter of the ansatz, where U_- and U_+ are both assumed to be unitary 2-designs, and the number of qubits is N, is given as [20, 26],

$$\mathrm{Var}[\partial_j C(\boldsymbol{\theta})] \approx \mathcal{O}\left(\frac{1}{2^{6N}}\right). \tag{5}$$

This means that for a unitary 2-design the gradients of the ansatz vanish exponentially with respect to the number of qubits N. Details of this proof is provided in Appendix A. Now, let us consider the CS case. We split the ansatz $U(\boldsymbol{\theta})$ to k many m-qubit ansätze, where we assume without loss of generality that $N = k \times m$. Then, we introduce a new notation for each classically split layer,

$$U_l^i(\theta_l^i) = e^{-i\theta_l^i V_l^i} W_l^i, \tag{6}$$

where index l determines the layer and index i determines which sub-circuit it belongs to. This notation combines the parametrized and entangling gates under U_l^i. Then, the overall CS ansatz can be be expressed as,

$$U(\boldsymbol{\theta}) = \prod_{l=1}^{L} \bigotimes_{i=1}^{k} U_l^i(\theta_l^i) = \bigotimes_{i=1}^{k} \prod_{l=1}^{L} U_l^i(\theta_l^i) = \bigotimes_{i=1}^{k} U^i(\boldsymbol{\theta^i}). \tag{7}$$

The CS ansatz can be seen in Fig. 1a. Next, we will assume the observable and the input state to be classically split, such that they both can be expressed as a tensor product of m-qubit observables or states. This assumption restricts our proof to be valid only for m-local quantum states and m-local observables. It is important to note here that we use a definition that is different from the literature throughout the paper. For this proof, an m-local observable is an observable such that there are no operators that act on overlapping groups of m qubits.

A generic m-local observable can be expressed as,

$$O_{m\text{-local}} = \sum_{i=1}^{k} O_i \otimes \mathbb{1}_{\bar{i}} = \sum_{i=1}^{k} \bigotimes_{j=1}^{k} (O_i - \mathbb{1})\,\delta_{i,j} + \mathbb{1}, \tag{8}$$

where O_i is an observable over the qubits $\{(i-1)m+1, (i-1)m+2, \ldots, im\}$, and $\bar{i}$ represents the remaining $N-m$ qubits. Then, the cost function becomes;

$$\begin{aligned} C(\boldsymbol{\theta}) &= \sum_{i=1}^{k} \mathrm{Tr}[\bigotimes_{j=1}^{k} \big((O_i - \mathbb{1})\,\delta_{i,j} + \mathbb{1}\big)\, U^j(\boldsymbol{\theta}^j)\rho_j U^{j\dagger}(\boldsymbol{\theta}^j)] \\ &= \sum_{i=1}^{k} \prod_{j=1}^{k} \mathrm{Tr}[\big((O_i - \mathbb{1})\,\delta_{i,j} + \mathbb{1}\big)\, U^j(\boldsymbol{\theta}^j)\rho_j U^{j\dagger}(\boldsymbol{\theta}^j)] \\ &= \sum_{i=1}^{k} \mathrm{Tr}[O_i U^i(\boldsymbol{\theta}^i)\rho_i U^{i\dagger}(\boldsymbol{\theta}^i)]. \end{aligned} \tag{9}$$

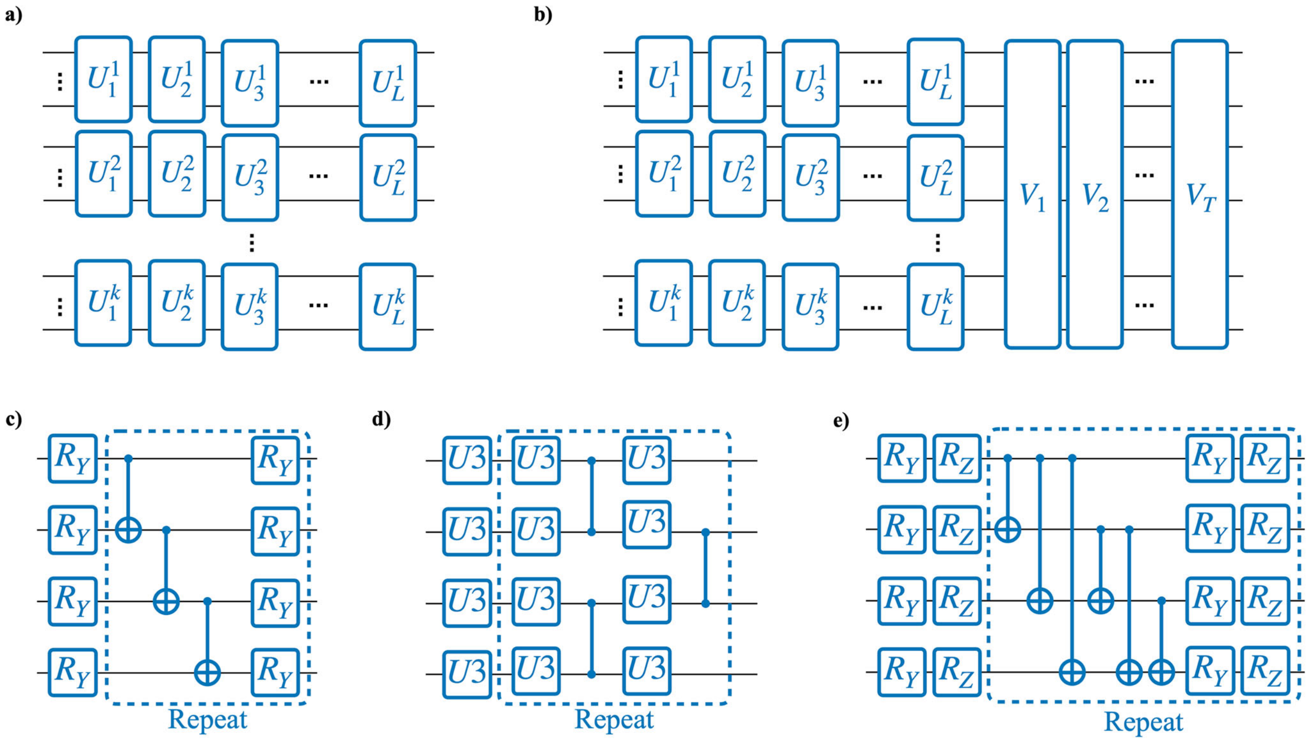

a)
b)
c)
d)
e)
U_1^1 U_2^1 U_3^1 ... U_L^1
U_1^2 U_2^2 U_3^2 ... U_L^2
U_1^k U_2^k U_3^k ... U_L^k
V_1 V_2 ... V_T
R_Y
R_Z
U3
Repeat

Fig. 1 All types of ansätze used in this work. (a) An N-qubit generic ansatz consisting of L layers of the parametrized unitary U are separated in to $k = N/m$ many m-qubit ansätze. This ansatz will be referred to as the classically split (CS) ansatz. The standard ansatz can be recovered by setting $m = N$. (b) Extended classically split (ECS) ansatz. This is an extension to the CS ansatz. First L layers of the ansatz consists of $k = N/m$ many m qubit U blocks. Then, T layers of N qubit V layers are applied. (c) A simple ansatz that consists of R_Y rotation gates and CX gates connected in a "ladder" layout. (d) Hardware Efficient Ansatz (HEA) that is used to produce the quantum dataset. Parameters of the first column of U3 gates are sampled from a uniform distribution ∈ $[-1, 1]$, while the rest of the parameters are provided by the dataset [42]. (e) EfficientSU2 ansatz with "full" entangler layers [45]

This can be written as a simple sum,

$$C(\boldsymbol{\theta}) = \sum_{i=1}^{k} C^i(\boldsymbol{\theta^i}), \tag{10}$$

where,

$$C^i(\boldsymbol{\theta^i}) = \mathrm{Tr}[O_i U^i(\boldsymbol{\theta^i})\rho_i U^{i\dagger}(\boldsymbol{\theta^i})]. \tag{11}$$

Then, the costs of each classically separated circuit are independent of each other. The gradient of j^{th} parameter of the i^{th} ansatz can be written as,

$$\begin{aligned}\partial_{i,j} C(\boldsymbol{\theta}) &= \partial_{i,j} C^i(\boldsymbol{\theta^i}) \\ &= \partial_{i,j}(\mathrm{Tr}[O_i U^i(\boldsymbol{\theta^i})\rho_i U^{i\dagger}(\boldsymbol{\theta^i})]).\end{aligned} \tag{12}$$

Now, let us consider each ansatz $U^i(\boldsymbol{\theta^i})$ to be a unitary 2-design. We want to choose the integer m such that it scales logarithmically in N. Hence, we choose β and γ appropriately, such that $m = \beta \log_\gamma N$ holds. Then, if we combine Eq. (5) with Eq. (12), the variance of the gradient of j^{th} parameter can be expressed as

$$\mathrm{Var}[\partial_j C(\boldsymbol{\theta})] \approx \mathcal{O}\left(\frac{1}{2^{(6m)}}\right) = \mathcal{O}\left(\frac{1}{N^{6\beta \log_\gamma 2}}\right). \tag{13}$$

Here, the dependence on i or j becomes irrelevant (a simpler choice for ansatz design would be to choose every new ansatz to be the same), so it can be dropped for a simpler notation. Similar to Eq. (5) the variance scales with the dimension of the Hilbert space (e.g. $\mathcal{O}(2^m)$). Then, the overall expression scales with, $\mathcal{O}(N^{-6\beta \log_\gamma 2})$, where β and γ are constant (e.g. $\beta = 1$ and $\gamma = 2$ results in $m = \log_2 N$). As a result, the variance of the CS ansatz scales with $\mathcal{O}(\mathrm{poly}(N)^{-1})$ instead of $\mathcal{O}(\exp(N)^{-1})$. Therefore, a CS ansatz, irrespective of its choice of gates or layout, can be used without leading to barren plateaus.

3 Numerical experiments

In this section, we report results of four numerical experiments. We investigate the scaling of gradients under CS by computing variances over many samples in Sect. 3.1. Then, we perform three experiments to observe how CS affects the performance of an ansatz. This task by itself leads to many questions as there are multitudes of metrics that one needs to compare and as many different problems one can consider. For this purpose, we consider problems well known in the literature, where trainability of ansätze plays a significant role.

First, we perform binary classification on a synthetic classical dataset in Sect. 3.2. The dataset contains two distributions that are called as classes. The goal is to predict the class of each sample. We perform the same task for distribution of quantum states in Sect. 3.3. Then, we give practical remarks in Sect. 3.4. Finally, we propose an extension to the CS ansatz and employ it for quantum simulating the ground state of the transverse field Ising Hamiltonian in Sect. 3.5.

For the first three experiments (Sects. 3.1 to 3.3), we consider the CS ansatz with layers that consist of R_Y rotation gates and CX entangling gates applied in a ladder formation for each layer. This layer can be seen in Fig. 1c. As the observable, we construct the 1-local observable defined in Eq. (14), where Z_i represents the Pauli-Z operator applied on the i^{th} qubit and $\mathbb{1}_{\bar{i}}$ represents the identity operator applied on the rest of the qubits.

$$O = \frac{1}{N}\sum_{i=1}^{N} Z_i \otimes \mathbb{1}_{\bar{i}} \tag{14}$$

3.1 Barren Plateaus

Barren Plateaus are typically identified by looking at the variance of the first parameter over a set of random samples [26]. Recently, it has been shown that this is equivalent to looking at the variance of samples from the difference of two cost values evaluated at different random points of the parameter space [3]. In particular, in the presence of barren plateaus this difference is exponentially suppressed, and, thus, barren plateaus also affect gradient-free optimizers [2]. For this reason, we will focus on the variance of the cost function as a more inclusive indicator for barren plateaus, rather than the gradients to provide a broader picture.

The experiments were performed using analytical gradients and expectation values, assuming a perfect quantum computer and infinite number of measurements, using Bergholm et al. (2020) and Paszke et al. 2019. Variances are computed over 2000 samples, where the values of the parameters are randomly drawn from a uniform distribution over $[0, 2\pi]$.

We start by presenting the variances over different values of m and N in Fig. 2. We fix the number of layers (L) to N, so that the ansatz exhibits barren plateaus in the setting without CS ($m = N$). The results indicate that a constant value of m resolves the exponential behaviour, as expected from Eq. (13). Furthermore, it is evident that larger values of m can allow the ansatz to escape barren plateaus, given that m grows slow enough (e.g. $\mathcal{O}(\log N)$). Note that we study the variances with respect to randomly chosen parameter sets, and not the variance during the optimization procedure to find the optimal set of parameters minimizing the cost function. Thus, our results essentially show the expected variance at

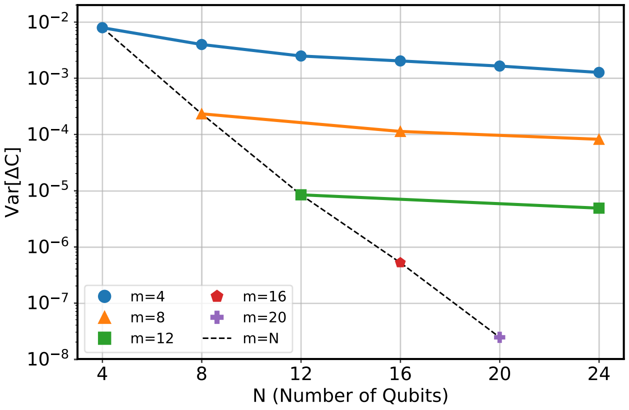


Fig. 2 The variance of the change in cost vs. the number of qubits for varying values of m. Each color/marker represents a certain value of m and data points of the standard ansatz ($m = N$) is plotted with a dashed black line

beginning of the optimization procedure with a randomly chosen initial point. Having a large variance at this state is key to find the path towards the global minima. Throughout the optimization procedure, the variance of the cost function will eventually decrease as the algorithm converges.

Our theoretical findings illustrate that the CS can be used to avoid barren plateaus irrespective of the number of layers. In our first experiment, we numerically showed that this holds when we set $L = N$. Recent findings showed that, a transition to barren plateaus happens at a depth of $\mathcal{O}(\log N)$ for an ansatz with a local cost function [11]. Therefore, there is great importance in investigating the behaviour for larger values of L. For considerably low values of N (e.g. $N < 32$), we can assume a constant value for m (e.g. $m = 4$), such that m is approximately $\mathcal{O}(\log N)$. We present variances of two ansätze ($m = 4$, $m = N$) for up to 200 layers and 16 qubits in Fig. 3. For the standard ansatz, we see a clear transition to barren plateaus with increasing number of layers, as expected [11]. On the other hand, the CS ansatz ($m = 4$) shows a robust behavior from small to large number of layers.

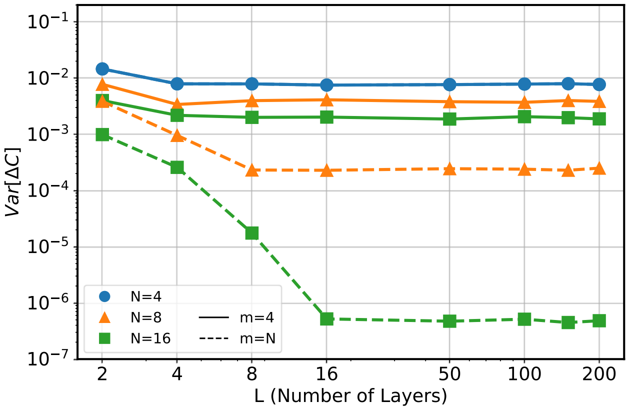


Fig. 3 The variance of the change in cost vs. the number of layers for $m = 4$ (solid lines) and $m = N$ (dashed lines) with varying number of qubits

These two experiments show the potential of the CS in avoiding barren plateaus. However, the question of whether this potential can be transferred in-to practice (e.g. binary classification performance or quantum simulation) still lacks an answer. Next, we will be addressing this question.

3.2 Binary classification using a classical dataset

In this experiment, we will continue using the same ansatz with same assumptions to perform binary classification using a classical dataset. Our goal here is to compare performance of the CS ansatz to the standard case for increasing number of qubits. We need a dataset that can be scaled for this purpose. However, datasets are typically constant in dimension and do not offer an easy way to test the scalability in this sense. Therefore, we employ an ad-hoc dataset that can be produced with different number of features.

Three datasets ($N = 4$, 8 and 16) were produced using the make_classification function of scikit-learn[1] [31]. This tool allows us to draw samples from an N-dimensional hypercube, where samples of each class are clustered around the vertices. Each dataset contains 420 training and 180 testing samples. Each of the data samples were encoded using one R_Y gate per qubit, such that each ansatz uses the same number of features of the given dataset. Please see Appendix C for more details on the production of the dataset and distributions of samples.

The binary classification was performed using the expectation value over the observable defined in Eq. (14) and the binary cross entropy function was used as the loss function during training, such that,

$$L(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}), \tag{15}$$

where y (i.e. $y \in \{0, 1\}$) is the class label of the given data sample and $\hat{y}$ is the prediction (i.e. $\hat{y} = \mathrm{Tr}[OU(\boldsymbol{\theta})\rho(x)U^{\dagger}(\boldsymbol{\theta})]$, where x is the data sample).[2] The ADAM optimizer [22] with a learning rate of 0.1 was used and all models are trained for 100 epochs using full batch size (bs=420).[3] We report our results based on 50 runs for each setting.

Classification performance of ansätze for changing values of m using the three datasets are presented in Fig. 4. Here, the

[1] The classical dataset is produced for 600 data samples with a 420/180 train/test split, a class separation value of 1.0, 2.0% class assignment error and no redundant or repeated features.

[2] Here, the expectation value can have values between [-1,1], we scale it to be [0,1] to compensate for the discrepancy between the class labels.

[3] In the case of $N = m = L = 16$ full batch size was not possible due to vast memory requirement. Therefore, bs=60 was used only for this case. In Appendix D, we show that using a smaller batch size does not affect the performance of the model significantly.

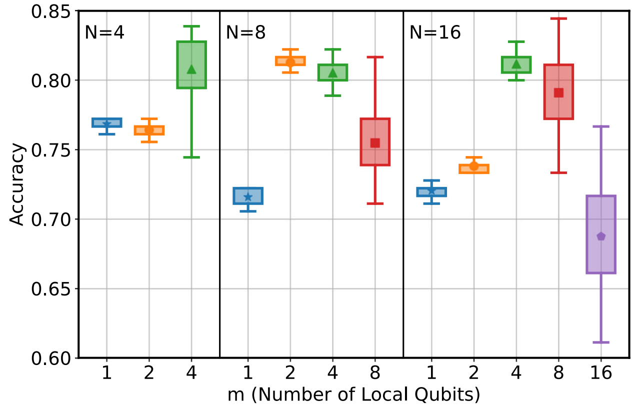


Fig. 4 Box plot of the best test accuracy obtained over 50 runs plotted with respect to the relevant local number of qubits (m). Each column represents a problem with a different sample size (4, 8, 16). Each marker is placed on the median, boxes cover the range from the first to third quartiles and the error bars extend the quartiles by 3 times range. Each m value is plotted with a different marker and color

results show the distribution of accuracies over the test set. For the $N = 4$ case, we see that the standard ($m = N$) ansatz performs the best. However, this is not the case as we go to more qubits. For the 8 and 16 qubit cases, it is evident that $m < N$ ansätze can match the performance of the standard ansatz. We can also see that the constant choice of $m = 4$ can provide a robust performance with increasing number of qubits (at least up-to $N = 16$), matching our expectations. Training curves of all settings are presented in Appendix D.

3.3 Binary classification using a quantum dataset

The binary classification performance of the CS over the classical datasets provides the first numerical evidence for their advantage against the standard ansätze. It is also important to investigate if they can be extended to problems where the data consists of quantum states. Our proof in Sect. 2 assumed the input states to be tensor product states. Now, we remove this constraint and use a quantum dataset.

For this experiment, we use the NTangled dataset [42]. NTangled dataset provides parameters to produce distributions of quantum states that are centered around different concentrable entanglement (CE) [5] values. CE is a measure of entanglement, which is defined as follows,

$$\mathrm{CE}(|\Psi) = 1 - \frac{1}{2^N} \sum_{\alpha \in Q} \mathrm{Tr}[\rho_\alpha^2], \tag{16}$$

where Q is the power set of the set $\{1, 2, ..., \mathrm{N}\}$, and ρ_α is the reduced state of subsystems labeled by the elements of α associated to $|\Psi$. The NTangled dataset provides three ansätze trained for different concentrable entanglement (CE) values for $N = 3$, 4 and 8. We choose the Hardware Efficient Ansatz (Fig. 1d) with depth=5, such that the parameters of the first layer of $U3$ gates are sampled from a unitary distribution $\in [-1, 1]$ and the others are provided by the dataset. Then, we apply the same CS ansatz used in Sect. 3.2 and perform binary classification such that the CE values are the labels of classes. The CE distributions of the produced quantum states are presented in Appendix E.

For the binary classification task, the same training settings are used as in Sect. 3.2, except this time models are trained until 50 epochs, as most models were able to reach 100% test accuracy. We report our results using different pairs of distributions in Table 1. In the case of $N = 4$, we observed that CS can perform at similar accuracy, even if the ansatz do not have any entangling gates ($m = 1$). We see that entangling gates are needed for better performance if the problem gets harder (e.g. 0.25 vs. 0.35 case). If we go to a problem with more qubits, we can safely say that the CS ansatz can match the performance of the standard ansatz and converge faster.

3.4 Practical remarks on classical splitting

The efficacy of CS relies on the parts of the circuit before and after the set of gates that undergo CS. This can be seen most clearly if we set $m = 1$ and apply CS to the entire circuit after a possible initialization. In this case, we only perform single qubit operations after initialization. Hence, if the initialization produces a tensor product state, then the circuit subject to CS with $m = 1$ can no longer generate any entanglement. Similarly, if we initialize with the HEA (Fig. 1d) and apply CS with $m = 1$ to the remaining circuit, then no tensor product state can be found.

More generally, $m = 1$ produces a circuit that cannot change the amount of entanglement. For other choices of m, the picture becomes more complicated but, generally, the set of states that can be generated by the quantum circuit before CS will be reduced to a subset based on the characteristics of the remaining initialization.

A naïve implementation of CS therefore requires knowledge of the correct initialization such that the final solution can still be reached with the classically split circuit. In generic applications, this knowledge is likely not available. Hence, an adaptive approach to CS should be considered.

One adaptive approach would be to increase m to check for improvements. After we observe no further training improvement with $m = 1$, we could move to $m = 2$. This enlarges the set of states the quantum circuit can reach, and thus may lead to further training improvements, at the cost of possibly stronger BP effects. However, if $m = 1$ has already converged fairly well, then the state is already fairly close to the $m = 2$ solution and it is unlikely to find a BP. With $m = 2$ converged, we can then move to $m = 4$ and continue the process by doubling m one step at a time.

Table 1 Classification performance of ansätze with different values of m over different distributions of quantum states from the NTangled dataset [42]. Average of 50 runs are presented with errors showing the difference to maximum and minimum observed values. Best average value of each metric for the given task is printed in bold

N	Task [CE Values]	L	m	Train Accuracy (%)	Avg. epochs to reach 90% Train Accuracy	Avg. epochs to reach 100% Train Accuracy	Test Accuracy (%)	Avg. epochs to reach 90% Test Accuracy	Avg. epochs to reach 100% Test Accuracy
4	0.05 vs. 0.35	4	1	$94.6^{+2.5}_{-1.7}$	$6.7^{+11.3}_{-5.7}$	N/A	$94.6^{+3.8}_{-1.8}$	$6.1^{+11.9}_{-5.1}$	N/A
			2	$\mathbf{100.0}^{+0.0}_{-0.5}$	$\mathbf{4.9}^{+12.1}_{-3.9}$	N/A	$100.0^{+0.0}_{-0.0}$	$\mathbf{3.9}^{+11.1}_{-2.9}$	$10.8^{+26.2}_{-9.8}$
			4	$99.9^{+0.1}_{-1.6}$	$5.4^{+6.6}_{-4.4}$	N/A	$100.0^{+0.0}_{-1.1}$	$4.1^{+8.9}_{-3.1}$	N/A
4	0.25 vs. 0.35	4	1	$90.4^{+4.1}_{-3.5}$	N/A	N/A	$86.4^{+6.9}_{-5.9}$	N/A	N/A
			2	$98.2^{+1.5}_{-1.3}$	$7.7^{+25.3}_{-5.7}$	N/A	$97.1^{+2.3}_{-1.6}$	$7.9^{+27.1}_{-6.9}$	N/A
			4	$\mathbf{100.0}^{+0.0}_{-0.4}$	$\mathbf{5.1}^{+9.9}_{-4.1}$	N/A	$\mathbf{100.0}^{+0.0}_{-1.1}$	$\mathbf{4.5}^{+11.5}_{-3.5}$	N/A
8	0.15 vs. 0.45	8	1	$99.9^{+0.1}_{-0.2}$	$3.3^{+3.7}_{-2.3}$	N/A	$100.0^{+0.0}_{-0.0}$	$2.4^{+2.6}_{-1.4}$	$6.1^{+10.9}_{-5.1}$
			2	$100.0^{+0.0}_{-0.0}$	$2.5^{+2.5}_{-1.5}$	$7.1^{+11.9}_{-6.1}$	$100.0^{+0.0}_{-0.0}$	$1.5^{+2.5}_{-0.5}$	$3.2^{+6.8}_{-2.2}$
			4	$100.0^{+0.0}_{-0.0}$	$\mathbf{2.4}^{+1.6}_{-1.4}$	$\mathbf{4.6}^{+4.4}_{-3.6}$	$100.0^{+0.0}_{-0.0}$	$\mathbf{1.4}^{+1.6}_{-0.4}$	$\mathbf{2.9}^{+7.1}_{-1.9}$
			8	$100.0^{+0.0}_{-0.0}$	$2.8^{+2.2}_{-0.8}$	$7.8^{+11.2}_{-5.8}$	$100.0^{+0.0}_{-0.0}$	$1.8^{+2.2}_{-0.8}$	$4.8^{+8.2}_{-3.8}$
8	0.40 vs. 0.45	8	1	$99.9^{+0.1}_{-0.4}$	$3.1^{+2.9}_{-2.1}$	N/A	$99.6^{+0.4}_{-0.7}$	$2.2^{+2.8}_{-1.2}$	N/A
			2	$100.0^{+0.0}_{-0.0}$	$2.8^{+4.2}_{-1.8}$	$9.2^{+11.8}_{-8.2}$	$100.0^{+0.0}_{-0.0}$	$1.9^{+4.1}_{-0.9}$	$5.2^{+5.8}_{-4.2}$
			4	$100.0^{+0.0}_{-0.0}$	$\mathbf{2.4}^{+1.6}_{-1.4}$	$\mathbf{5.3}^{+12.7}_{-4.3}$	$100.0^{+0.0}_{-0.0}$	$\mathbf{1.5}^{+1.5}_{-0.5}$	$\mathbf{3.2}^{+9.8}_{-2.2}$
			8	$100.0^{+0.0}_{-0.0}$	$2.9^{+3.1}_{-0.9}$	$8.2^{+9.8}_{-6.2}$	$100.0^{+0.0}_{-0.0}$	$1.9^{+3.1}_{-0.9}$	$5.7^{+5.3}_{-4.7}$

If, for example, we consider the $N = 4$ "0.25 vs. 0.3" case of Table 1, we may start training with $m = 1$. This training converges to about 90% accuracy. Increasing m to $m = 2$ will lead to further improvements that converge to about 98% accuracy. Finally, we can further improve the 98% to 100% accuracy by going to $m = 4$.

In this way, we utilize the efficiency of CS to obtain an approximate solution which we then refine by trading efficiency for circuit expressivity through increasing m. At this point, the efficiency reduction should no longer lead to insurmountable complications as we already are close to the optimal solution for the current m value.

Another adaptive approach would be to use CS to check and bypass plateaus. For example, if a VQE appears to be converged, it may also just be stuck in a plateau. Applying CS at this point would reduce the effect of the plateau. Thus, if the VQE continues optimizing after classically splitting a seemingly converged circuit, we can conclude that this was in fact a plateau. After a suitable number of updates using the classically split circuit, we can then return to the full circuit in the hopes of having passed the plateau.

Unfortunately, this approach cannot be used to positively distinguish between true local optima and plateaus since the CS reduces expressivity and thus introduces artificial constraints. Hence, if the set of states expressible by the classically split circuit is orthogonal to the gradient in the cost function landscape, then a plateau will be replaced with a local optimum and, thus, no improvements will be obtained. In this case, we therefore cannot conclude that the VQE has converged simply because CS shows no improvements. However, experimenting with different implementations of CS may result in cases that do not replace the plateau with an artificial local optimum.

3.5 Extending classical splitting to VQE

Until now, we have investigated using CS for binary classification problems. It succeeded by showing an overall better training performance in Sect. 3.2 and a competitive performance and faster convergence in Sect. 3.3. In this section, we consider simulating the ground state of the transverse-field Ising hamiltonian (TFIH) on a 1D chain. The TFIH with open boundary conditions can be defined as

$$H = -J \sum_{i=1}^{N-1} Z_i Z_{i+1} - h \sum_{i=1}^{N} X_i, \tag{17}$$

for N lattice sites, where J determines the strength of interactions and h determines the strength of the external field. Simulating the TFIH on a 1D link requires at least nearest neighbour interactions between qubits on the 1D lattice as the ground state. This contradicts with the assumption we made, when we proved absence of barren plateaus for classically split ansätze in Sect. 2, since the TFIH does not fit the definition we had for an m-local observable in Eq. (8).

Therefore, we need to rely on the numerical experiments to talk about barren plateaus under the new constraints.

The CS ansätze can only produce local entangled states, for this reason we need an extension of the ansatz in Fig. 1a. We propose to extend the CS ansatz by adding standard layers at the end. The reason for adding them at the end is to keep the base of light cones[4] produced by the classically split layers constant. Then, when we add the standard layers, the light cones will grow at a pace that is determined by the newly-added part.[5] This way, the overall ansatz can still escape barren plateaus as long as the newly-added part does not exhibit barren plateaus.

We define the extended classically split (ECS) ansatz with two types of layers. First L layers consist of classically split m qubit gate blocks. Then, there are T layers of any no-BP ansatz (see Fig. 1b). Since the first L layers can only produce m-local product states (i.e. $m < \mathcal{O}(\log N)$), the existence of barren plateaus depends only on the remaining T layers. This way we can choose very large L, but need to keep T small as standard ansätze reach BPs rather rapidly (e.g. depth $> \mathcal{O}(\log N)$ leads to barren plateaus for such an ansatz [11]). In Fig. 5, we provide numerical evidence to show that addition of L layers do not decrease the size of the gradients. The variance saturates faster to higher values and the Renyi-2 entropy to lower values respectively with increasing total depth for constant T layers.

These results suggest that it might be possible to leverage this feature of the ECS ansatz to add more layers which can contribute to finding the ground state with a better success rate without sacrificing trainability [1]. This is also important from an overparameterization perspective as it improves generalization capacity of QML models [23]. We perform experiments with the TFIH to test this idea.

We consider the Hamiltonian defined in Eq. (17) with $J = 1, h = 1$. Then, we implement the ECS ansatz with $m = 3, N = D = 12$ and $m = 4, N = D = 16$. Each side of the ansatz consists of EfficientSU2 layers with ladder connectivity (similar to Fig. 1e). Then, we consider different values for L and T layers, where the percentage of split layers correspond to $p = L/D$ and $L + T = D$. Total depth (D) corresponds to $L + T$, where $p = 100\%$ is equivalent to the CS ansatz, $p = 0\%$ is equivalent to the standard EfficientSU2 ansatz and other values explore hybrid use cases of the ECS ansatz.

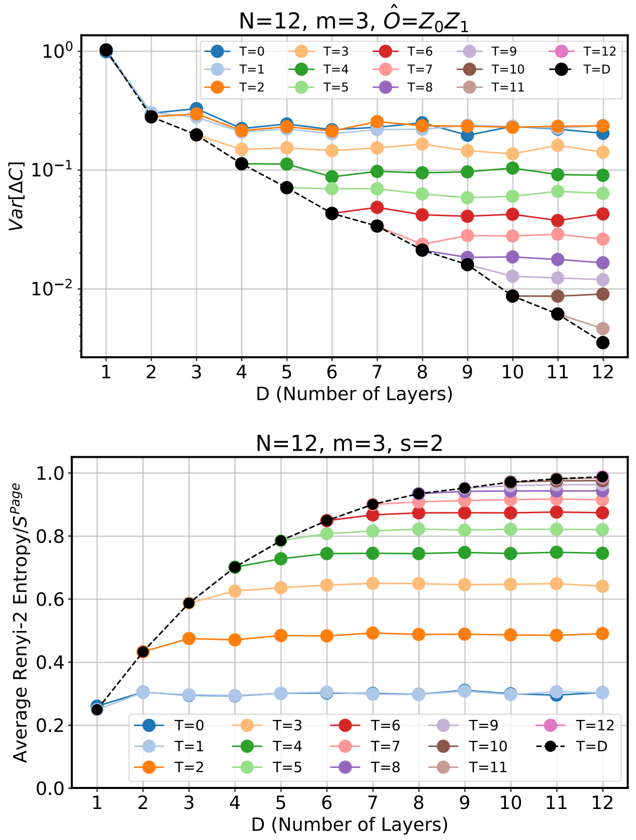


Fig. 5 The variance of the change in cost vs. the total number of layers for $m = 3, N = 12$ and the observable $\hat{O} = Z_0Z_1$ with varying number of the L and T layers ($L + T = D$) are shown in the upper panel. The lower panel shows the Renyi-2 entropies of the same system for a subsystem size of two. $T = D$ line is shown with a black dashed line to emphasize that it is the standard case. Other colors represents different values the ECS ansatz can take. Both figures shows that the variances and entanglement entropy saturate faster and to higher and lower values respectively for constant values of T

We report the final fidelities of 100 runs in Fig. 6. Each run starts with a set of parameters drawn from a uniform distribution $[-0.1, 0.1]$. The ADAM [22] optimizer with 0.1 learning rate is used and optimization was performed for 1000 iterations and reached a convergence in all cases. Circuits are simulated with no shot noise. This choice was made to investigate if the ECS ansatz can achieve the same level of success compared to its standard equivalent under infinite resources assumptions. Otherwise, results in Fig. 5 shows us that the ECS ansatz needs orders of magnitudes less number of shots.

The upper panel shows results for the $N = D = 12$ setting. Here, we see that the standard ansatz ($p = 0\%$) manages to find the ground state with a success probability (amount of runs reaching close to 1.0 fidelity) of approximately 0.5. Increasing the percentage of split layers improve the success rate. As expected, increasing it too much and eventually

[4] A light cone or a causal cone of an ansatz is an abstract concept that illustrates how information spreads as more gates are applied. The types of gates and their connectivity determines the opening angle of the cone. The evidence from the literature suggests that there is a correspondence between the opening angle of the cone, barren plateaus and quantum circuit complexity [11, 19].

[5] It also depends on the choice of m, but since we already have a constraint on m (i.e. $m = \mathcal{O}(\log N)$) the newly-added ansatz will be the dominant component.

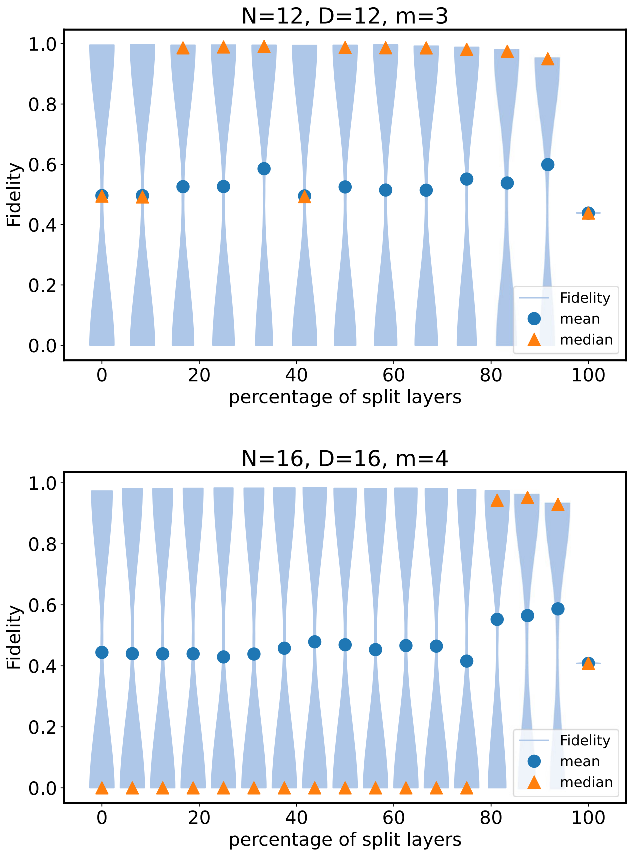


Fig. 6 Fidelity values after optimization with 100 different initial points using different values of normal and split layers

making it 100% results in loss of performance, as the ansatz becomes inadequate to represent the TFIH ground state.

The lower panels shows results for the $N = D = 16$. The effect of moving to a regime with more qubits is seen as a drop in success probability for the $p = 0\%$ ansatz. This time increasing the percentage of split layers improve success probability at larger values. This suggests that the relationship between number of qubits and the amount of the split layer might be more intricate than it seems.

4 Discussion

In this work, we showed that the CS of the ansätze can be used to escape barren plateaus both analytically and numerically. Then, we investigated if the CS hinders the learning capacity of the ansatz. Our experiments showed that this is not the case, and the classically split ansatz can match the performance at low number of qubits and is potentially superior at larger number of qubits.

In general the benefits of CS comes from the reducing the effective Hilbert Space that the CS ansatz can explore. CS only allows the ansatz to produce m-qubit tensor product states, if the input state is also a tensor product state following our assumptions in Sect. 2. This, as a result, reduces the expressivity of the ansatz. Nevertheless, this also allows the ansatz to avoid barren plateaus [20] by limiting the scaling behavior to the more favorable case of m-qubit systems. In the case of the CS, the exponential increase of the Hilbert space dimension is prevented and instead a polynomial scaling is enforced. For the m-local CS ansatz, each local Hilbert space have $\dim(H_k) = 2^m = N^{\beta \log_2 \gamma}$. Although the advantage of using classical splitting may look trivial, there are many benefits of employing such an ansatz besides the numerical experiments we performed in Sect. 3.

In our binary classification experiments using a classical dataset, we relied on single qubit and single rotation gate data encoding. This meant that any classically split ansatz had less information in each group. This could in fact be improved with embedding methods such as data re-uploading, where one can encode all the data points to each single qubit independently, such that there are alternating layers of rotation gates that encode the data and parametrized gates that are to be optimized [33]. Data re-uploading ansätze shows great classification performance even for low number of qubits. Since the classical splitting doesn't have a limit on the amount of layers, data re-uploading would potentially be great way to get a performance increase.

CS can provide faster training when used with gradient based optimizers. In general, the exact gradients of ansätze are computed with the well-known parameter shift rule [27, 43]. However, this requires two instances of the same circuit to be executed per parameter. This quickly results in a bottleneck for the optimization procedure. An ansatz with $L = N$ layers, where each layer has N parameters, requires $\mathcal{O}(N^2)$ circuit executions to compute gradients for a single data sample. On the other hand, CS provides cost functions that are independent of each other, as it was shown in Eq. (11). This allows gradients to be computed simultaneously across different instances of the classically split ansatz. As a result, the classically split ansatz optimization requires $\mathcal{O}(N \log N)$ circuit executions for $m = \mathcal{O}(\log N)$.

The bottleneck in optimization is only one of the challenges of implementing scalable Variational quantum algorithms. Another problem that is worth mentioning here is the amount of two-qubit gates. NISQ hardware provides limited connectivity of qubits. The topology of the devices plays an essential role in the efficient implementation of quantum circuits [48]. Typically, a quantum circuit compilation (or transpilation) procedure is required to adapt a given circuit to be able to be compatible with the capabilities of the devices (e.g. converting gates to native gates, applying SWAP gates to connect qubits which are not physically connected) [7].

Classical splitting provides a significant reduction in number of two qubit gates as it divides a large qubit to many circuits with less qubits. To show the scale of the reduction,

Table 2 Two qubit gate counts of different ansätze transpiled for hypothetical devices that has a 2D grid topology (square lattice with no diagonal connections)

		amount of two qubit gates					
		linear entanglement			full entanglement		
m	L	$N = 4$	$N = 16$	$N = 36$	$N = 4$	$N = 16$	$N = 36$
N	2	6	33	121	24	696	3601
	N	12	240	1362	46	5372	65040
4	2	6	24	54	24	92	250
	N	12	192	978	46	964	4376
2	2	4	16	36	4	16	42
	N	8	128	654	8	134	648

we can construct a set of hypothetical devices that has a 2D grid topology (square lattice with no diagonal connections). We start by considering the CS ansatz that consists the ansätze in Fig. 1c and extend it to a fully entangled architecture. A linear entangled ansatz has $\mathcal{O}(N)$ two qubit gates, while a fully entangled one has $\mathcal{O}(N^2)$ per layer. Then, we use Qiskit's transpiler[6] [45] to fit these ansätze to the hypothetical devices and report the two qubit gate counts in Table 2.

The amount of gates are not only important to have a better implementation but also to have a more precise results, since NISQ devices come with noisy gates. We consider the CX gate errors reported by IBM for their devices, which can be taken as $\mathcal{O}(10^{-2})$ on average[7]. Then, as a figure of merit, we can assume 50% to be the limit, in which we can still get meaningful results. This would allow us to use 50 CX gates at most. Now, the results from Table 2 implies that it is possible to construct a 36 qubit, 2 layer ansatz with linear entanglement, if we employ CS. This would not be possible for the standard case as it comes with more than twice two qubit gates. The reduction only gets better if we consider a full entanglement case. Following the same logic, to implement a 36 qubit, 36 layer, fully entangled ansatz, a CX gate error of $\mathcal{O}(10^{-6})$ is needed, while the classically split ansatz only requires a CX gate error of $\mathcal{O}(10^{-4})$. A similar reduction in noise is also possible for other types of circuit partitioning methods [4].

Classically splitting an ansatz further allows faster implementation on hardware. A generic ansatz consists of two-qubit gates that follow one and another, matching a certain layout. We mentioned some of these as ladder/linear or full. However, this means that the hardware implementation of such an ansatz requires execution of these gates sequentially, taking a significant amount of time. To overcome such obstacles, ansätze such as the HEA (see Fig. 1d) are widely used in the literature [21]. CS an ansatz can reduce the implementation time significantly since it allows simultaneous two-qubit gates across different local circuits. This can mean a speed-up of from $\mathcal{O}(N/\log N)$ to $\mathcal{O}((N/\log N)^2)$ depending on the connectivity of the original ansatz.

The formulation we used in Sect. 3.2 allows the CS ansatz to be implemented on smaller quantum computers instead of a single large quantum computer. This means that for similar problems, there are many implementation options available. These include using one large device, using many small devices (e.g., $\mathcal{O}(N/\log N)$ many $\mathcal{O}(\log N)$ qubit devices) and parallelizing the task or using one small device and performing all computation sequentially. All of these features makes the classical splitting an ideal approach for Quantum Machine Learning (QML) applications using NISQ devices.

Simulating larger size systems requires a deep ansatz (linear or larger in system size) in general [10]. Although a problem-agnostic ansatz can perform well at small sizes, BPs preclude the scalability. Our results show that the ECS can help circumvent this issue and allow deeper ansätze. On the other hand, the ECS ansatz also brings the quantum circuit closer to the classically simulatable limit. It appears that there might be a transition point where the ECS ansatz is deep enough to represent the ground state of interest without leading to BPs. We were not able to formulate how or if this point can be identified for an arbitrary system size of a given problem.

5 Conclusion

In this work, we presented some foundational ideas of applying CS to generic ansätze. Our results indicate many benefits of using CS, such as better trainability, faster hardware implementation, faster convergence, robustness against noise and parallelization under certain conditions. These suggest that CS or variations of this idea might play an essential role in how we are designing ansätze for QML problems. We also presented an extension to the initial CS idea so that these types of ansätze can be used in VQE. The initial results that we presented in this work suggest that CS can help improve the trainability and reach better error values. However, it is still an open question to what extent VQE can benefit from classical splitting. Our results encourage employing approaches that are based upon classically splitting or partitioning parametrized quantum circuits [8, 13, 16, 25, 32, 34, 41, 44], as they are in general more robust against hardware noise. We consider in-depth analysis and applications with VQE and QAOA as future directions for this work.

[6] Qiskit's transpiler algorithm is a stochastic algorithm, meaning that it is possible to get better values if the algorithm is executed many times. Here, we run the algorithm two times and take the best results using optimization level 3, and sabre-sabre layout and routing methods. Although, It is possible to obtain better gate counts with more runs or different transpilation algorithms, the best values obtained wouldn't change our conclusions.

[7] This value is chosen after a survey of devices listed on IBM Quantum Cloud.

Appendix A

When analyzing the size of the gradients of an ansatz we need tools that allows integration over all states allowed by the ansatz over the d-dimensional Hilbert Space. This can be achieved by using the Haar measure. Haar measure is an invariant measure over the SU(d) group. An ensemble of unitary operators U is called as a unitary t-design if they are equal to the Haar measure $\mu(U)$ up-to polynomial order t. Then, the expectation of ensemble U, where unitary V_i can be sampled with probability p_i is given as,

$$\mathbb{E}_H^t(\rho) = \int U^{\otimes t}\rho(U^{\otimes t})^\dagger\, dU = \sum_i p_i V_i^{\otimes t}\rho(V_i^{\otimes t})^\dagger. \quad (18)$$

Then, to perform symbolic integration over the Haar measure we will need to use some properties of the measure [38]. For the first moment we have,

$$\int d\mu(U) U_{ij} U^*_{km} = \frac{\delta_{ik}\delta_{jm}}{d}, \quad (19)$$

where d is the dimension of the Unitary, such that $d = 2^N$ and N is number of qubits. Then, for the second moment we have,

$$\begin{aligned}&\int d\mu(U) U_{i_1j_1}U_{i_2j_2}U^*_{k_1m_1}U^*_{k_2m_2} = \\ &= \frac{\delta_{i_1k_1}\delta_{j_1m_1}\delta_{i_2k_2}\delta_{j_2m_2} + \delta_{i_1k_2}\delta_{i_2k_1}\delta_{j_1m_2}\delta_{j_2m_1}}{d^2+1} \\ &- \frac{\delta_{i_1k_1}\delta_{j_2m_2}\delta_{j_1m_2}\delta_{j_2m_1} + \delta_{i_1k_2}\delta_{i_2k_1}\delta_{j_1m_1}\delta_{j_2m_2}}{d(d^2+1)}\end{aligned} \quad (20)$$

Then one can derive the following identities for integrals over the Haar measure [11, 20, 26],

$$\int d\mu(U)\mathrm{Tr}[UAU^\dagger B] = \frac{\mathrm{Tr}[A]\mathrm{Tr}[B]}{d}. \quad (21)$$

We can extend this to the second moment to obtain the following identity,

$$\begin{aligned}&\int d\mu(U)\mathrm{Tr}[UAU^\dagger BUCU^\dagger D] = \\ &= \frac{\mathrm{Tr}[A]\mathrm{Tr}[C]\mathrm{Tr}[BD] + \mathrm{Tr}[AC]\mathrm{Tr}[B]\mathrm{Tr}[D]}{d^2-1} \\ &- \frac{\mathrm{Tr}[AC]\mathrm{Tr}[BD] + \mathrm{Tr}[A]\mathrm{Tr}[B]\mathrm{Tr}[C]\mathrm{Tr}[D]}{d(d^2-1)}.\end{aligned} \quad (22)$$

We also have,

$$\begin{aligned}&\int d\mu(U)\mathrm{Tr}[UAU^\dagger B]\mathrm{Tr}[UCU^\dagger D] = \\ &= \frac{\mathrm{Tr}[AC]\mathrm{Tr}[B]\mathrm{Tr}[D] + \mathrm{Tr}[AC]\mathrm{Tr}[BD]}{d^2-1} \\ &- \frac{\mathrm{Tr}[AC]\mathrm{Tr}[B]\mathrm{Tr}[D] + \mathrm{Tr}[A]\mathrm{Tr}[C]\mathrm{Tr}[BD]}{d(d^2-1)}.\end{aligned} \quad (23)$$

Now, we can use these identities to compute the average value of the gradients. Let's start by reminding ourselves the definitions we used before. The ansatz is composed of consecutive parametrized (V) and non-parametrized entangling (W) layers. We define $U_l(\theta_l) = \exp(-i\theta_l V_l)$, where V_l is a Hermitian operator and W_l is a generic unitary operator. Then, the curcuit ansatz can be expressed with a multiplication of layers,

$$U(\boldsymbol{\theta}) = \prod_{l=1}^{L} U_l(\theta_l)W_l \quad (24)$$

For an observable O and an input state ρ, the cost function is given as

$$C(\boldsymbol{\theta}) = \mathrm{Tr}[OU(\boldsymbol{\theta})\rho U^\dagger(\boldsymbol{\theta})] \quad (25)$$

The ansatz can be separated into two parts to investigate a certain layer, such that $U_- \equiv \prod_{l=1}^{j-1} U_l(\theta_l)W_l$ and $U_+ \equiv \prod_{l=j}^{L} U_l(\theta_l)W_l$. Then, the gradient of the j^{th} parameter can be expressed as [26]

$$\partial_j C(\boldsymbol{\theta}) = \frac{\partial C(\boldsymbol{\theta})}{\partial\theta_j} = i\,\mathrm{Tr}[[V_j, U_+^\dagger OU_+]U_-\rho U_-^\dagger] \quad (26)$$

Then the expected value of the gradient with respect to the unitary group can be computed by using the Haar integral such that,

$$\langle\partial_j C(\boldsymbol{\theta})\rangle = i\int d\mu(U_-)d\mu(U_+)\mathrm{Tr}[[V_j, U_+^\dagger OU_+]U_-\rho U_-^\dagger] \quad (27)$$

$$= \frac{i\mathrm{Tr}[\rho]}{d}\int d\mu(U_+)\mathrm{Tr}[[V_j, U_+^\dagger OU_+] = 0, \quad (28)$$

where we use Eq. (21) to obtain (27) and use the fact that trace of the commutator is zero in (28). This proves that the gradients are centered around zero. Then, the variance of the gradient can inform us about the size of the gradients. The variance is defined as,

$$\begin{aligned}&\mathrm{Var}[\partial_j C(\boldsymbol{\theta})] = \langle(\partial_j C(\boldsymbol{\theta}))^2\rangle - \langle\partial_j C(\boldsymbol{\theta})\rangle^2 \\ &= \langle(\partial_j C(\boldsymbol{\theta}))^2\rangle\end{aligned} \quad (29)$$

We can compute the expected value of the variance using the same logic. Then we have,

$$\begin{aligned}
&\mathrm{Var}[\partial_j C(\boldsymbol{\theta})] = \\
&= -\int d\mu(U_-)d\mu(U_+)\mathrm{Tr}[[V_j, U_+^\dagger O U_+]U_-\rho U_-^\dagger]^2 \\
&= -\frac{1}{d^2-1}\int d\mu(U_+)\Big(\mathrm{Tr}[\rho]^2\mathrm{Tr}[[V_j, U_+^\dagger O U_+]]^2 \\
&+\mathrm{Tr}[\rho^2]\mathrm{Tr}[[V_j, U_+^\dagger O U_+]^2]\Big) \\
&+ \frac{1}{d(d^2-1)}\int d\mu(U_+)\Big(\mathrm{Tr}[\rho^2]\mathrm{Tr}[[V_j, U_+^\dagger O U_+]^2] \\
&+\mathrm{Tr}[\rho]^2\mathrm{Tr}[[V_j, U_+^\dagger O U_+]]^2\Big)
\end{aligned} \tag{30}$$

$$= -\left(\mathrm{Tr}[\rho^2] - \frac{1}{d}\right)\frac{1}{d^2-1}\int d\mu(U_+)\mathrm{Tr}[[V_j, U_+^\dagger O U_+]^2] \tag{31}$$

We use Eq. (23) to obtain Eq. (30). Then, use the fact that commutator being traceless to obtain Eq. (31). To compute the integral of Eq. (31) we need another identity such that [20],

$$\begin{aligned}
&\mathrm{Tr}[[V_j, U_+^\dagger O U_+]^2] = \\
&2\mathrm{Tr}[U_+ V_j U_+^\dagger O U_+ V_j U_+^\dagger O] - 2\mathrm{Tr}[U_+ V_j^2 U_+^\dagger O^2].
\end{aligned} \tag{32}$$

Then, the variance becomes,

$$\begin{aligned}
&\mathrm{Var}[\partial_j C(\boldsymbol{\theta})] = \\
&-\left(\mathrm{Tr}[\rho^2] - \frac{1}{d}\right)\frac{2}{d^2-1}\int d\mu(U_+)\Big(\mathrm{Tr}[U_+ V_j^2 U_+^\dagger O^2] \\
&+\mathrm{Tr}[U_+ V_j U_+^\dagger O U_+ V_j U_+^\dagger O]\Big).
\end{aligned} \tag{33}$$

The first integral can be computed using Eq. (21) and the second can be computed using Eq. (22). Then we obtain,

$$\begin{aligned}
&\mathrm{Var}[\partial_j C(\boldsymbol{\theta})] = \\
&-\left(\mathrm{Tr}[\rho^2] - \frac{1}{d}\right)\frac{2}{d^2-1}\,(\,\frac{1}{d^2-1}(\mathrm{Tr}[V]^2\mathrm{Tr}[O]^2 \\
&+\mathrm{Tr}[V^2]\mathrm{Tr}[O^2]) - \frac{1}{d(d^2-1)}(\mathrm{Tr}[V]^2\mathrm{Tr}[O]^2 \\
&+\mathrm{Tr}[V^2]\mathrm{Tr}[O^2]) - \frac{1}{d}\mathrm{Tr}[V^2]\mathrm{Tr}[O^2]\,) \\
&= -\left(\mathrm{Tr}[\rho^2] - \frac{1}{d}\right)\frac{2\mathrm{Tr}[V^2]\mathrm{Tr}[O^2]}{d^2-1}\left(\frac{d-1}{d(d^2-1)}(1 \right. \\
&\left. +\mathrm{Tr}[V]^2\mathrm{Tr}[O]^2) - \frac{1}{d}\right).
\end{aligned} \tag{34}$$

Finally, the asymptotic behavior of the variance can be expressed as

$$\mathrm{Var}[\partial_j C(\boldsymbol{\theta})] \approx \mathcal{O}\left(\frac{1}{d^6}\right) \approx \mathcal{O}\left(\frac{1}{2^{6N}}\right), \tag{35}$$

where $d = 2^N$. Thus, the variance vanishes exponentially with respect to N.

Appendix B

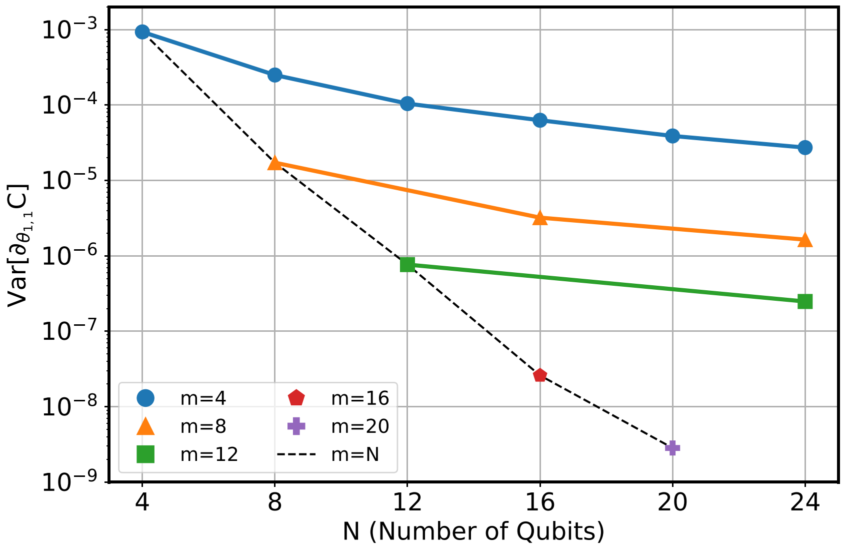
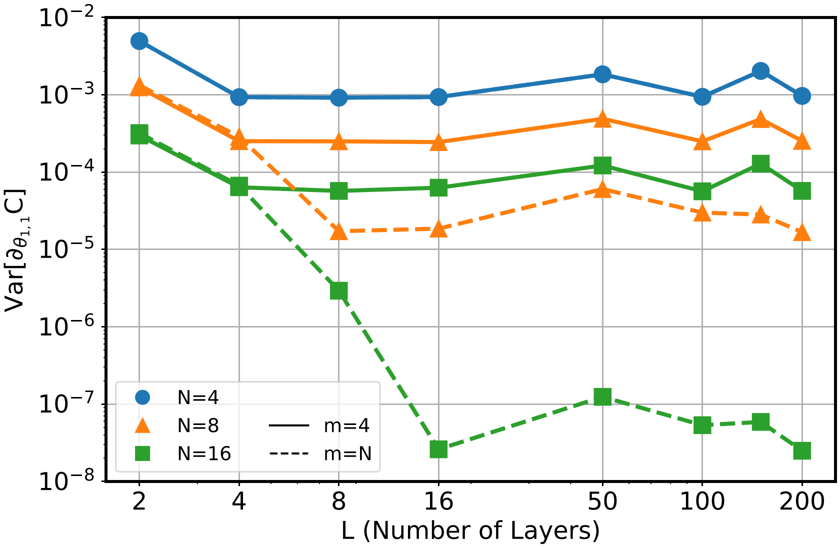


Fig. 7 The variance of the gradients of the first parameter of the ansatz as a function of the number of qubits for varying values of m. Each color/marker represents a certain value of m and data points of the standard ansatz ($m = N$) is plotted with a dashed black line

Fig. 8 The log plot of variance of the gradients of the first parameter of the ansatz vs. number of layers for $m = 4$ (solid lines) and $m = N$ (dashed lines) with varying number of qubits



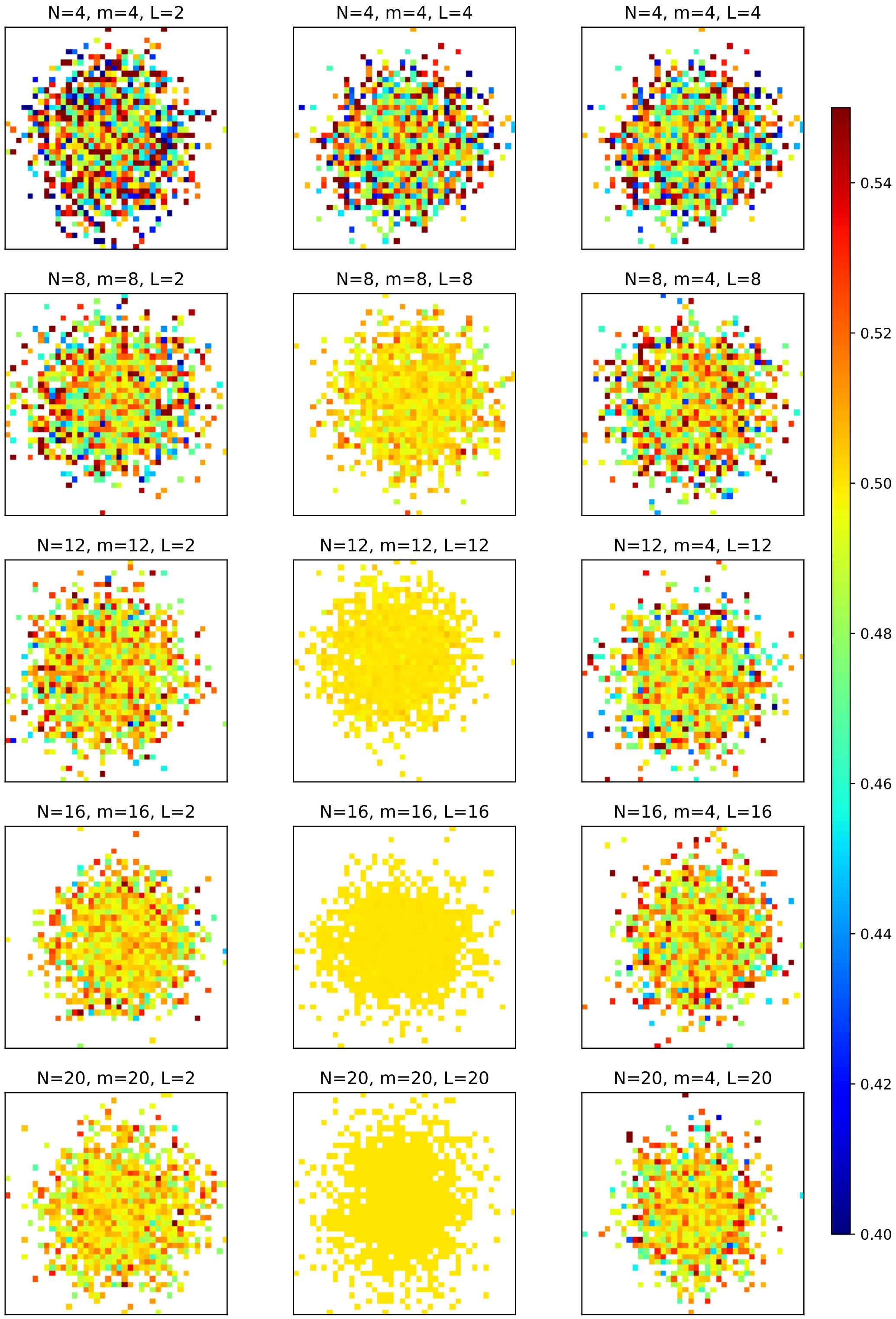


Fig. 9 Cost landscapes of ansätze with different settings. Parameters of the ansatz are reduced down to two using PCA and the x and y axis of the plots represents the PCA variables in same scale but with arbitrary units. The cost values (shown with the color map) are obtained using the definitions in Sect. 3.2. First column shows cost values of an $L = 2$ standard ansatz for increasing number of qubits. Second column shows the results for the same ansatz but with $L = N$ layers. As, expected the landscape flattens with more qubits and we see a single color for $N > 12$. Third column shows results for splitting (for $m = 4$) of the ansatz in the case of $L = N$ layers. We see that the landscape does not become flatter with more qubits

Appendix C

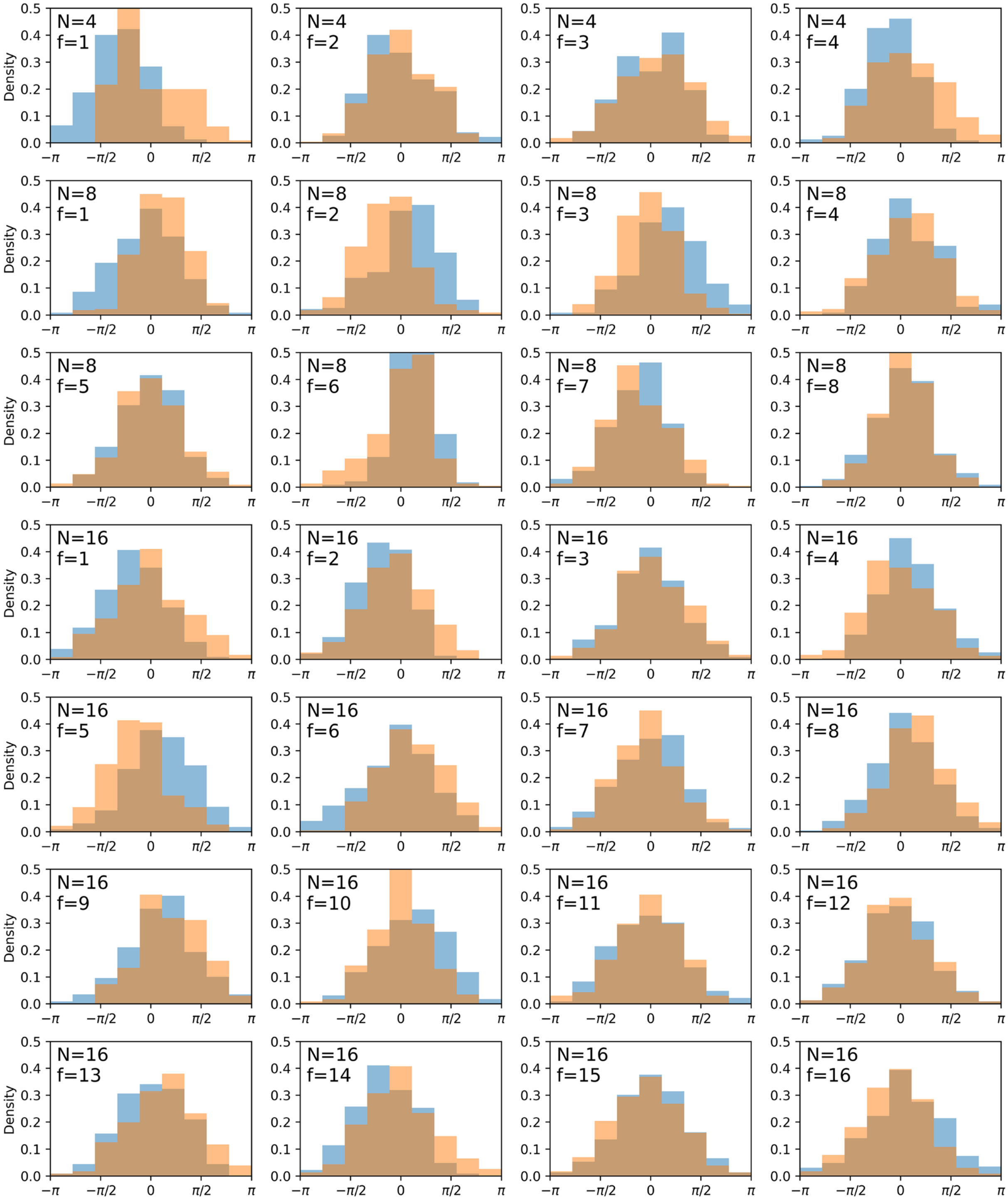


Fig. 10 Distributions of the ad-hoc dataset used in Sect. 3.2. Each panel shows distribution of a single feature from one of three datasets. N denotes the size of the dataset (number of features), while f denotes the feature number. There exists 600 samples of N features for a size N dataset. Colors represent two classes. During training, data samples are divided with a 420/180 train/test ratio. The dataset is produced using make_classification function of scikit-learn [31] with a class separation value of 1.0, 2% class assignment error and no redundant or repeated features

Appendix D

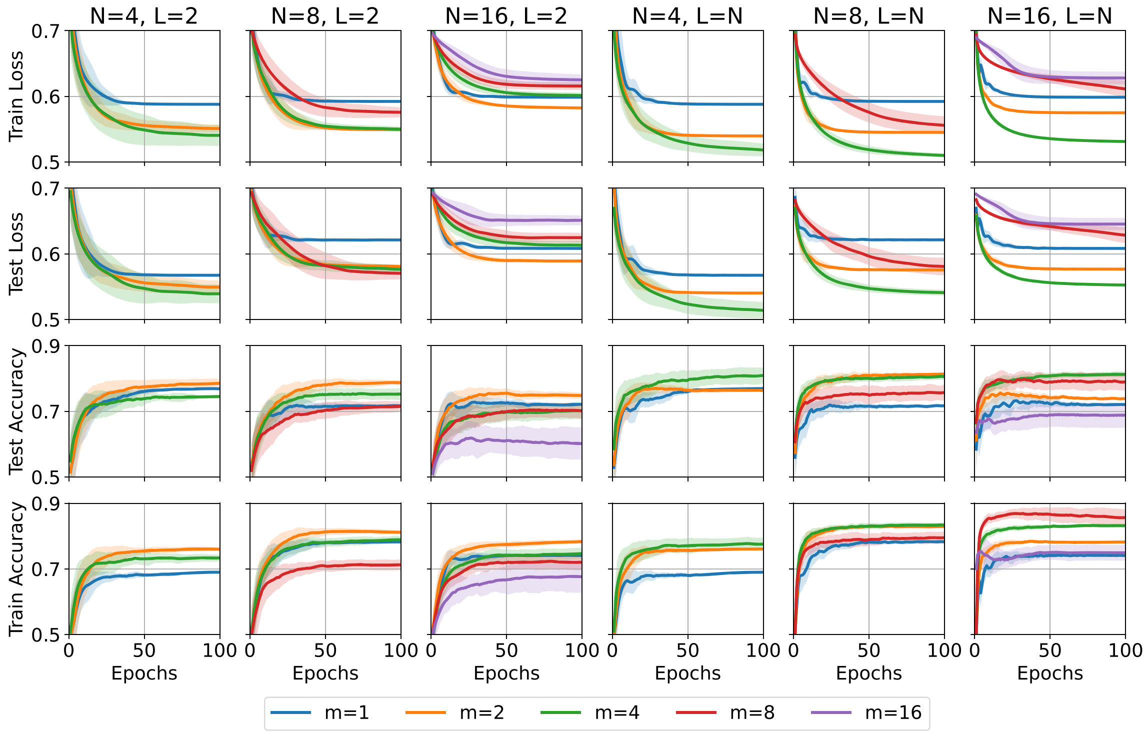


Fig. 11 Training curves showing four different metrics for the problem described in Sect. 3.2. Panels of each row show a different metric. First three columns show training results from $L = 2$ ansätze, the last three columns show training results from $L = N$ ansätze for $N \in\{4,8,16\}$. Each value of m is plotted with a different color. Lines are obtained by averaging 50 runs and their standard deviation is shown with shades

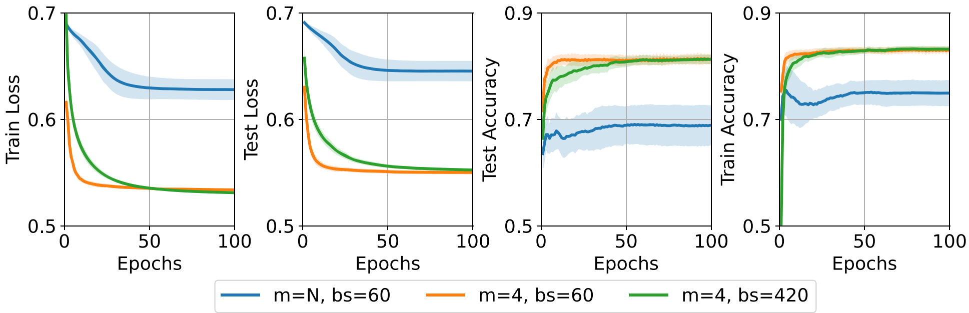


Fig. 12 Batch size comparison for the training of $N = 16, m = 16$ and $m = 4$. Training the $N = L = 16$ model requires vast computational resources, especially memory. This restricted us from using a full batch size during the training of $N = m = L = 16$ setting. Therefore, we presented results from a training that used a batch size of 60 instead of 420 (full). Here, we show training curves for $m = 4$ on addition to $m = 16$ for two different batch size (bs). Behaviour of the curves show that the gain in performance has nothing to do with the batch size difference

Appendix E

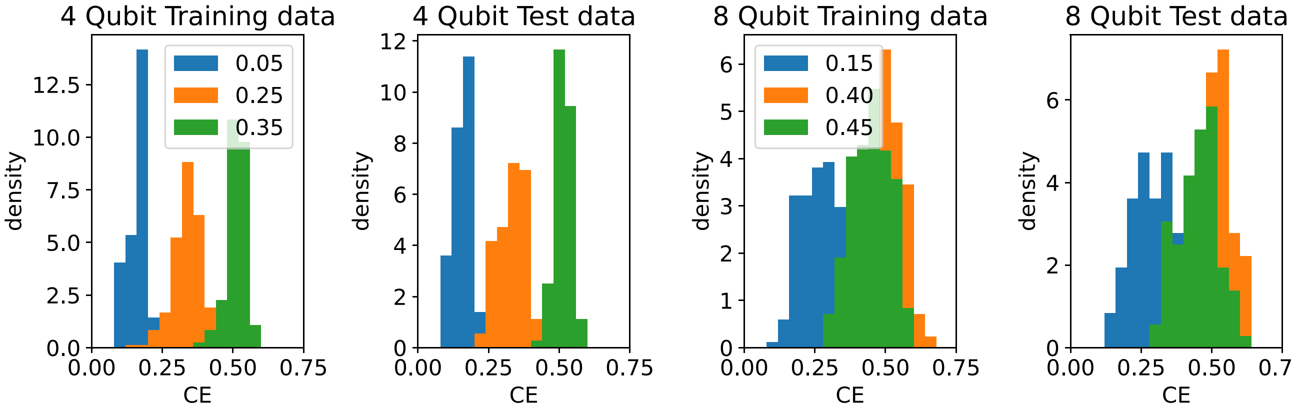


Fig. 13 Distributions of the NTangled [42] dataset with respect to the CE values described in Sect. 3.3. The HEA ansatz (Fig. 1d) is used to produce the distributions. Each training set has 420 and each test set has 180 data samples. We see a mismatch for CE ∈{0.40,0.45} in the 8 qubit case. We are not sure what causes this, but it is not an issue for our problem as we are not interested in the CE values themselves but the quantum states as a whole. So, they are valid quantum state distributions as long as they can be separated with a given metric for our problem. Our results show that this is in fact true

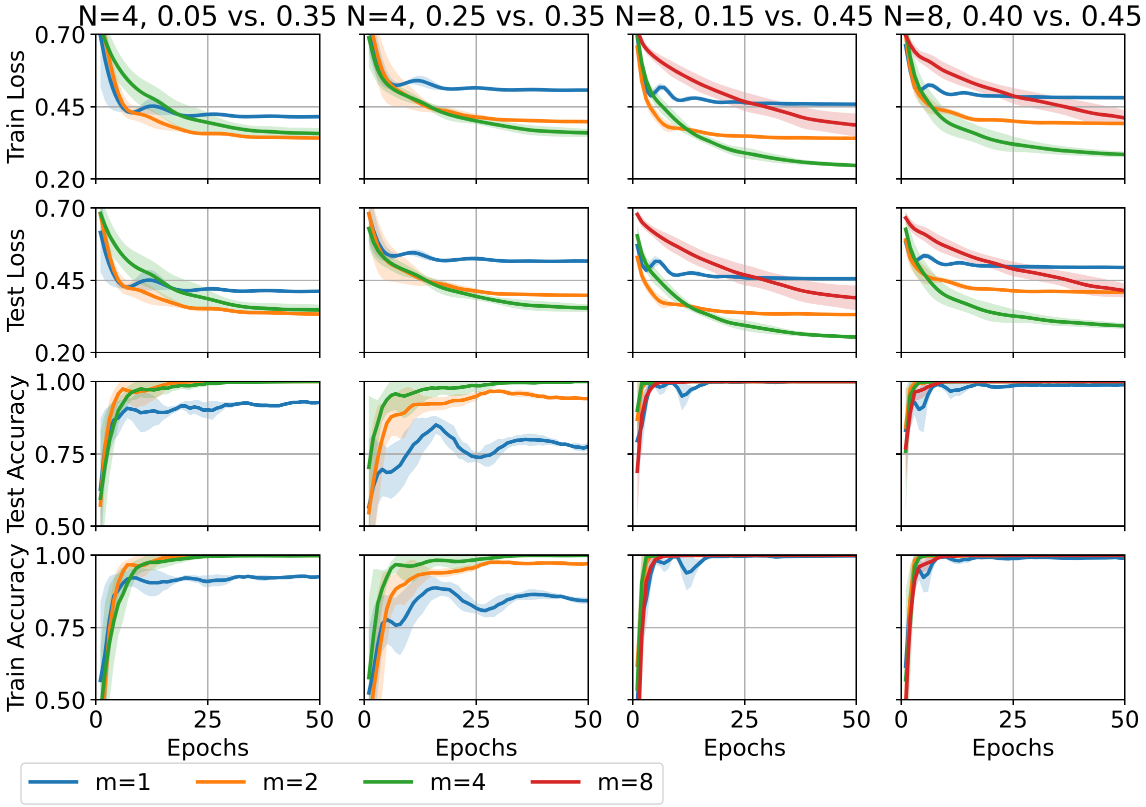


Fig. 14 Training curves showing four different metrics for the problem described in Sect. 3.3. Panels of each row show a different metric. Each column presents a different task, where *N* determines the problem size and the CE values are the labels of the classes. Each value of m is plotted with a different color. Lines are obtained by averaging 50 runs and their standard deviation is shown with shades

Acknowledgements We thank Lena Funcke for valuable discussions.

Author Contributions C.T. developed the main idea, performed the numerical experiments and wrote the main manuscript text and prepared the figures. G.C. performed some of the numerical simulations. All authors contributed to discussions, helped developing the ideas and reviewed the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. C.T. and A.C. are supported in part by the Helmholtz Association - "Innopool Project Variational Quantum Computer Simulations (VQCS)". S.K. acknowledges financial support from the Cyprus Research and Innovation Foundation under project "Future-proofing Scientific Applications for the Supercomputers of Tomorrow (FAST)", contract no. COMPLEMENTARY/0916/0048, and "Quantum Computing for Lattice Gauge Theories (QC4LGT)", contract no. EXCELLENCE/0421/0019. This work is supported with funds from the Ministry of Science, Research and Culture of the State of Brandenburg within the Centre for Quantum Technologies and Applications (CQTA). This work is funded by the European Union's Horizon Europe Framework Programme (HORIZON) under the ERA Chair scheme with grant agreement No. 101087126.

Data Availability The ad-hoc dataset used in Sect. 3.2 is produced using make_classification function of scikit-learn [31] with a class separation value of 1.0, 2% class assignment error and no redundant or repeated features. The dataset used in Sect. 3.3 is the NTangled dataset [42]. The data that support the findings of this study are provided in more detail in the supplementary material and also available from the corresponding author, C.T., upon request.

Declarations

Conflicts of interest Authors have no competing interests as defined by Springer, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

References

1. Anschuetz ER, Kiani BT (2022) Quantum variational algorithms are swamped with traps. Nature Communications 13(1):7760. https://doi.org/10.1038/s41467-022-35364-5. Number: 1 Publisher: Nature Publishing Group. Accessed 2022-12-15
2. Arrasmith, A., Cerezo, M., Czarnik, P., Cincio, L., Coles, P.J.: Effect of barren plateaus on gradient-free optimization. Quantum 5, 558 (2021). 10.22331/q-2021-10-05-558
3. Arrasmith A, Holmes Z, Cerezo M, Coles PJ (2022) Equivalence of quantum barren plateaus to cost concentration and narrow gorges. Quantum Science and Technology 7(4):045015. https://doi.org/10.1088/2058-9565/ac7d06
4. Basu, S., Saha, A., Chakrabarti, A., Sur-Kolay, S.: *i*-QER: An Intelligent Approach towards Quantum Error Reduction. arXiv:2110.06347 (2022). 10.48550/arXiv.2110.0634
5. Beckey JL, Gigena N, Coles PJ, Cerezo M (2021) Computable and Operationally Meaningful Multipartite Entanglement Measures. Phys. Rev. Letters 127(14):140501. https://doi.org/10.1103/PhysRevLett.127.140501
6. Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Alam, M.S., Ahmed, S., Arrazola, J.M., Blank, C., Delgado, A., Jahangiri, S., McKiernan, K., Meyer, J.J., Niu, Z., Száva, A., Killoran, N.: PennyLane: Automatic differentiation of hybrid quantum-classical computations. http://arxiv.org/abs/1811.04968arXiv:1811.04968 (2020). 10.48550/arXiv.1811.04968
7. Botea A, Kishimoto A, Marinescu R (2018) On the Complexity of Quantum Circuit Compilation. Proceedings of the International Symposium on Combinatorial Search 9(1):138–142. https://doi.org/10.1609/socs.v9i1.18463
8. Bravyi S, Smith G, Smolin JA (2016) Trading Classical and Quantum Computational Resources. Phys. Rev. X 6(2):021043. https://doi.org/10.1103/PhysRevX.6.021043
9. Broers, L., Mathey, L.: Reducing Barren Plateaus in Quantum Algorithm Protocols. http://arxiv.org/abs/2111.08085arXiv:2111.08085 (2021). 10.48550/arXiv.2111.08085
10. Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S.C., Endo, S., Fujii, K., McClean, J.R., Mitarai, K., Yuan, X., Cincio, L., Coles, P.J.: Variational quantum algorithms. Nature Reviews Physics, 625–644 (2021). 10.1038/s42254-021-00348-9
11. Cerezo M, Sone A, Volkoff T, Cincio L, Coles PJ (2021) Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nature Communications 12(1):1791. https://doi.org/10.1038/s41467-021-21728-w
12. Cong I, Choi S, Lukin MD (2019) Quantum convolutional neural networks. Nature Physics 15(12):1273–1278. https://doi.org/10.1038/s41567-019-0648-8
13. Eddins A, Motta M, Gujarati TP, Bravyi S, Mezzacapo A, Hadfield C, Sheldon S (2022) Doubling the size of quantum simulators by entanglement forging. PRX Quantum 3:010309. https://doi.org/10.1103/PRXQuantum.3.010309
14. Farhi, E., Goldstone, J., Gutmann, S.: A Quantum Approximate Optimization Algorithm. http://arxiv.org/abs/1411.4028arXiv:1411.4028 (2014)
15. Farhi, E., Neven, H.: Classification with Quantum Neural Networks on Near Term Processors. http://arxiv.org/abs/1802.06002arXiv:1802.06002 (2018)
16. Fujii K, Mizuta K, Ueda H, Mitarai K, Mizukami W, Nakagawa YO (2022) Deep Variational Quantum Eigensolver: A Divide-And-Conquer Method for Solving a Larger Problem with Smaller Size Quantum Computers. PRX Quantum 3(1):010346. https://doi.org/10.1103/PRXQuantum.3.010346
17. Grant, E., Ostaszewski, M., Wossnig, L., Benedetti, M.: An initialization strategy for addressing barren plateaus in parametrized quantum circuits. Quantum 3, 214 (2019). 10.22331/q-2019-12-09-214
18. Grant E, Benedetti M, Cao S, Hallam A, Lockhart J, Stojevic V, Green AG, Severini S (2018) Hierarchical quantum classifiers. npj Quantum. Information 4(1):17–19. https://doi.org/10.1038/s41534-018-0116-9
19. Haferkamp, J., Faist, P., Kothakonda, N.B.T., Eisert, J., Yunger Halpern, N.: Linear growth of quantum circuit complexity. Nature Physics 18(5), 528–532 (2022). 10.1038/s41567-022-01539-6
20. Holmes Z, Sharma K, Cerezo M, Coles PJ (2022) Connecting Ansatz Expressibility to Gradient Magnitudes and Barren

Plateaus. PRX Quantum 3(1):010313. https://doi.org/10.1103/PRXQuantum.3.010313

21. 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. https://doi.org/10.1038/nature23879
22. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. http://arxiv.org/abs/1412.6980arXiv:1412.6980 (2017)
23. Larocca, M., Ju, N., García-Martín, D., Coles, P.J., Cerezo, M.: Theory of overparametrization in quantum neural networks. arXiv:2109.11676 [quant-ph, stat] (2021). Accessed 2021-09-30
24. Liu H-Y, Sun T-P, Wu Y-C, Han Y-J, Guo G-P (2023) Mitigating barren plateaus with transfer-learning-inspired parameter initializations. New Journal of Physics 25(1):013039. https://doi.org/10.1088/1367-2630/acb58e
25. Marshall, S.C., Gyurik, C., Dunjko, V.: High Dimensional Quantum Learning With Small Quantum Computers. http://arxiv.org/abs/2203.13739arXiv:2203.13739 (2022). 10.48550/arXiv.2203.13739
26. McClean JR, Boixo S, Smelyanskiy VN, Babbush R, Neven H (2018) Barren plateaus in quantum neural network training landscapes. Nature Communications 9(1):4812. https://doi.org/10.1038/s41467-018-07090-4
27. Mitarai K, Negoro M, Kitagawa M, Fujii K (2018) Quantum circuit learning. Phys. Rev. A 98(3):032309. https://doi.org/10.1103/PhysRevA.98.032309
28. Ortiz Marrero C, Kieferová M, Wiebe N (2021) Entanglement-Induced Barren Plateaus. PRX. Quantum 2(4):040316. https://doi.org/10.1103/PRXQuantum.2.040316
29. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F.d., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc. (2019). https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf
30. Patti TL, Najafi K, Gao X, Yelin SF (2021) Entanglement devised barren plateau mitigation. Phys. Rev. Research 3(3):033090. https://doi.org/10.1103/PhysRevResearch.3.033090
31. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12(85):2825–2830
32. Peng T, Harrow AW, Ozols M, Wu X (2020) Simulating Large Quantum Circuits on a Small Quantum Computer. Phys. Rev. Letters 125(15):150504. https://doi.org/10.1103/PhysRevLett.125.150504
33. Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., Latorre, J.I.: Data re-uploading for a universal quantum classifier. Quantum 4, 226 (2020). 10.22331/q-2020-02-06-226
34. Perlin, M.A., Saleem, Z.H., Suchara, M., Osborn, J.C.: Quantum circuit cutting with maximum-likelihood tomography. npj Quantum Information 7(1), 1–8 (2021). 10.1038/s41534-021-00390-6
35. Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P.J., Aspuru-Guzik, A., O'Brien, J.L.: A variational eigenvalue solver on a photonic quantum processor. Nature Communications 5(1), 4213 (2014). 10.1038/ncomms5213
36. Pesah A, Cerezo M, Wang S, Volkoff T, Sornborger AT, Coles PJ (2021) Absence of Barren Plateaus in Quantum Convolutional Neural Networks. Phys. Rev. X 11(4):041011. https://doi.org/10.1103/PhysRevX.11.041011
37. Preskill, J.: Quantum computing in the NISQ era and beyond. Quantum 2(July), 1–20 (2018). 10.22331/q-2018-08-06-79
38. Puchala, Z., Miszczak, J.A.: Symbolic integration with respect to the haar measure on the unitary groups. Bulletin of the Polish Academy of Sciences: Technical Sciences 65(No 1), 21–27 (2017). 10.1515/bpasts-2017-0003
39. Rad, A., Seif, A., Linke, N.M.: Surviving The Barren Plateau in Variational Quantum Circuits with Bayesian Learning Initialization. http://arxiv.org/abs/2203.02464arXiv:2203.02464 (2022). 10.48550/arXiv.2203.02464
40. Sack SH, Medina RA, Michailidis AA, Kueng R, Serbyn M (2022) Avoiding barren plateaus using classical shadows. PRX Quantum 3:020365. https://doi.org/10.1103/PRXQuantum.3.020365
41. Saleem, Z.H., Tomesh, T., Perlin, M.A., Gokhale, P., Suchara, M.: Quantum Divide and Conquer for Combinatorial Optimization and Distributed Computing. http://arxiv.org/abs/2107.07532arXiv:2107.07532 (2021). 10.48550/arXiv.2107.07532
42. Schatzki, L., Arrasmith, A., Coles, P.J., Cerezo, M.: Entangled Datasets for Quantum Machine Learning. http://arxiv.org/abs/2109.03400arXiv:2109.03400 (2021). 10.48550/arXiv.2109.03400
43. Schuld M, Bergholm V, Gogolin C, Izaac J, Killoran N (2019) Evaluating analytic gradients on quantum hardware. Phys. Rev. A 99(3):1–7. https://doi.org/10.1103/PhysRevA.99.032331
44. Tang, W., Tomesh, T., Suchara, M., Larson, J., Martonosi, M.: CutQC: Using Small Quantum Computers for Large Quantum Circuit Evaluations. Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, 473–486 (2021). 10.1145/3445814.3446758
45. Treinish, M., Gambetta, J., Nation, P., Kassebaum, P., qiskit-bot, Rodríguez, D.M., González, S.d.l.P., Hu, S., Krsulich, K., Zdanski, L., Garrison, J., Yu, J., Gacon, J., McKay, D., Gomez, J., Capelluto, L., Travis-S-IBM, Marques, M., Panigrahi, A., Lishman, J., lerongil, Rahman, R.I., Wood, S., Bello, L., Itoko, T., Singh, D., Drew, Arbel, E., Schwarm, J., Daniel, J.: Qiskit: An Open-source Framework for Quantum Computing. Zenodo (2022). 10.5281/zenodo.6403335. https://zenodo.org/record/6403335
46. Volkoff T, Coles PJ (2021) Large gradients via correlation in random parameterized quantum circuits. Quantum Science and Technology 6(2):025008. https://doi.org/10.1088/2058-9565/abd891
47. Wang S, Fontana E, Cerezo M, Sharma K, Sone A, Cincio L, Coles PJ (2021) Noise-induced barren plateaus in variational quantum algorithms. Nature Communications 12(1):6961. https://doi.org/10.1038/s41467-021-27045-6
48. Weidenfeller, J., Valor, L.C., Gacon, J., Tornow, C., Bello, L., Woerner, S., Egger, D.J.: Scaling of the quantum approximate optimization algorithm on superconducting qubit based hardware. Quantum 6, 870 (2022). 10.22331/q-2022-12-07-870
49. Wu, A., Li, G., Ding, Y., Xie, Y.: Mitigating Noise-Induced Gradient Vanishing in Variational Quantum Algorithm Training. arXiv:2111.13209 (2021)
50. Zhang, K., Hsieh, M.-H., Liu, L., Tao, D.: Gaussian initializations help deep variational quantum circuits escape from the barren plateau. http://arxiv.org/abs/2203.09376arXiv:2203.09376 (2022). 10.48550/arXiv.2203.09376
51. Zhang, K., Hsieh, M.-H., Liu, L., Tao, D.: Toward Trainability of Deep Quantum Neural Networks. http://arxiv.org/abs/2112.15002http://arxiv.org/abs/2112.15002arXiv:2112.15002 (2021)
52. Zhao, C., Gao, X.-S.: Analyzing the barren plateau phenomenon in training quantum neural networks with the ZX-calculus. Quantum 5, 466 (2021). 10.22331/q-2021-06-04-466

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