PAPER • OPEN ACCESS

Matrix product state pre-training for quantum machine learning

To cite this article: James Dborin *et al* 2022 *Quantum Sci. Technol.* **7** 035014

View the article online for updates and enhancements.

Quantum Science and Technology

Matrix product state pre-training for quantum machine learning

James Dborin[1,*] ⓘ **, Fergus Barratt**[1,2]**, Vinul Wimalaweera**[1] ⓘ**, Lewis Wright**[2]
and Andrew G Green[1]

1   London Centre for Nanotechnology, University College London, London, United Kingdom
2   Department of Mathematics, King's College London, London, United Kingdom
*   Author to whom any correspondence should be addressed.

E-mail: james.dborin.18@ucl.ac.uk

Abstract

Hybrid quantum–classical algorithms are a promising candidate for developing uses for NISQ devices. In particular, parametrised quantum circuits (PQCs) paired with classical optimizers have been used as a basis for quantum chemistry and quantum optimization problems. Tensor network methods are being increasingly used as a classical machine learning tool, as well as a tool for studying quantum systems. We introduce a circuit pre-training method based on matrix product state machine learning methods, and demonstrate that it accelerates training of PQCs for both supervised learning, energy minimization, and combinatorial optimization.

1. Introduction

Parametrised quantum circuits (PQCs) have been the focus of attempts to demonstrate quantum computational advantage on NISQ devices, for problems of both scientific and commercial interest [1–3]. Typically these efforts involve parametrising a quantum circuit with a series of rotation angles, and using a classical optimizer to find a set of angles which minimizes a given cost function. The quantum device is used to estimate the cost function associated with a particular set of parameters, where it is assumed to be hard to calculate the cost function classically. Algorithms based on these methods are often called hybrid quantum–classical algorithms.
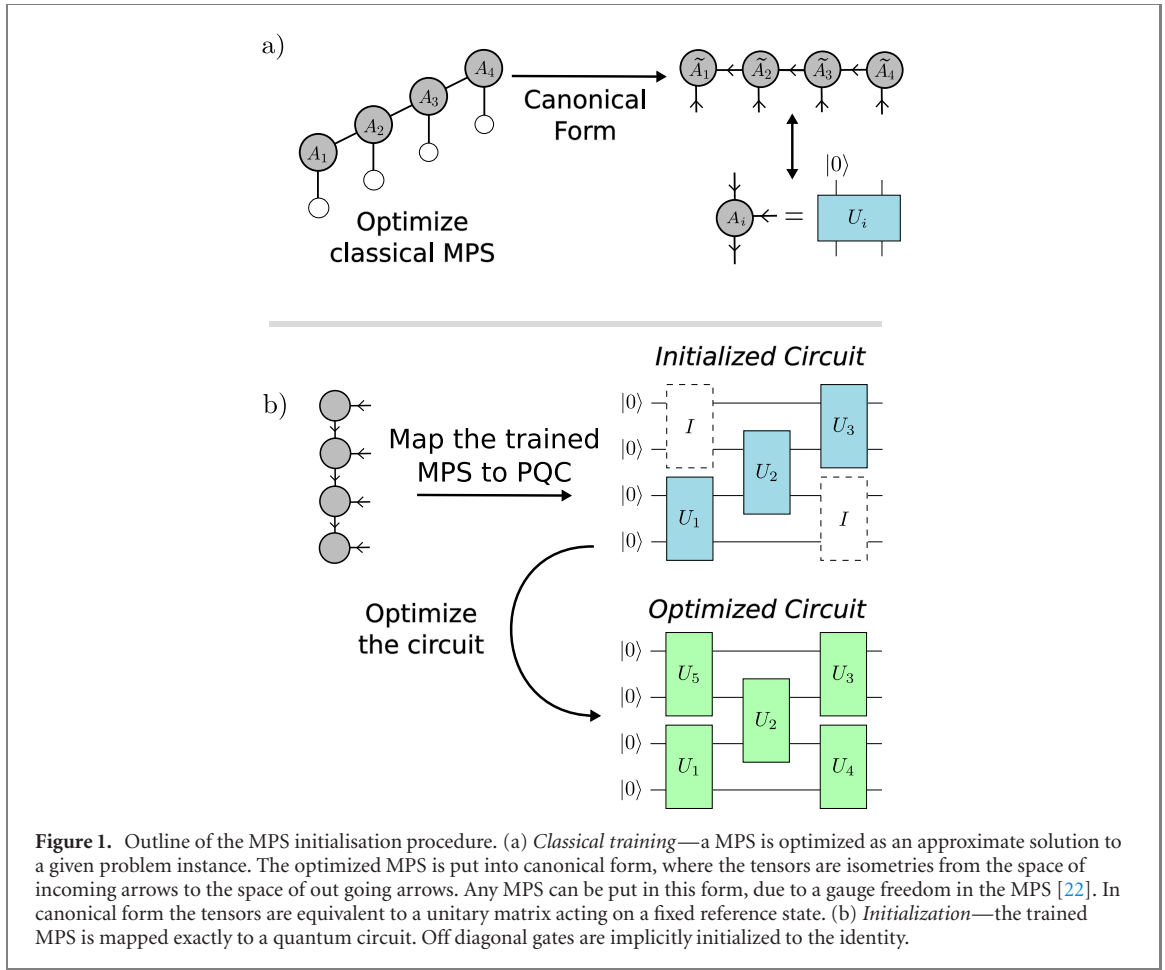
A major hurdle in developing useful hybrid algorithms is the problem of vanishing gradients. It has been shown that the size of initial gradients decrease exponentially towards zero as the number of qubits and the depth of the circuits increases, when parameters are randomly initialised [4]. Other circuit metrics have been demonstrated to produce these so-called barren plateaus, such as ansatz expressibility [5], the entanglement between hidden and visible nodes in the circuit [6, 7], and circuit noise [8].

The existence of the barren plateau has motivated attempts to improve PQC learning algorithms to avoid training costs growing exponentially. These include developing gradient free algorithms [9, 10], defining local cost functions as targets for learning algorithms [11], and initialisation schemes [12].

Here we introduce a novel initialisation scheme based on tensor network algorithms. Tensor network based methods are the state of the art for numerical simulations of 1D and 2D spin systems [13], and also for the simulation of quantum circuits [14]. Tensor networks have been used to solve optimization problems such as portfolio optimization, and are found to be competitive with commercial solvers [15, 16]. Recently, tensor network based methods have also been used as the basis for machine learning algorithms. Matrix product states (MPSs) have been trained as a classifier for several machine learning tasks [17, 18]. The 2D generalisation of MPS, projected entangled pair states, have also been used for image classification [19]. Both tree tensor networks and MERA networks have also been used as image classifiers [20, 21].

This circuit is as good an approximation to the optimal solution as the MPS. The optimizer is then able to vary all the parameters, including those in the gates initialised to the identity.

Previously classical tensor network techniques for finding ground states have been used to screen entangling gates when variationally generating an ansatz during training [23]. In this work, we optimize tensor networks as candidate ground states, as classifiers, and as solutions to combinatorial optimization

Figure 1. Outline of the MPS initialisation procedure. (a) *Classical training*—a MPS is optimized as an approximate solution to a given problem instance. The optimized MPS is put into canonical form, where the tensors are isometries from the space of incoming arrows to the space of out going arrows. Any MPS can be put in this form, due to a gauge freedom in the MPS [22]. In canonical form the tensors are equivalent to a unitary matrix acting on a fixed reference state. (b) *Initialization*—the trained MPS is mapped exactly to a quantum circuit. Off diagonal gates are implicitly initialized to the identity.

problems. Using PQCs to find ground state and as classifiers has been previously demonstrated to suffer from barren plateaus [12]. We use trained MPS to seed a PQC with an effective set of starting parameters, before continuing to train the quantum circuit. By beginning training in a part of the parameter space that is close to the target state, the number of steps to reach the desired minima can be reduced, potentially mitigating impacts of small gradients. From here onwards this procedure will be known as *MPS pretraining*.

## 2. MPS pretraining

MPS pretraining involves three steps:

(a) Train a tensor network
(b) Compile the tensor network into gates
(c) Initialise a circuit with these gates

Figure 1 outlines this procedure.

2.1. MPS optimization
Given a problem instance, such as a Hamiltonian or a labeled data set, and a cost function where the minimum corresponds to the state of interest, the first step is to produce a MPS that minimizes the given cost function. The bond dimension of a MPS, denoted χ, is a parameter that determines the complexity of the MPS model. Higher bond dimension models tend to produce better results, at the cost of greater training complexity. This training is done purely classically, and software exists both to efficiently contract and optimize these states, for example we refer the reader to references [14, 24–26]. In the following we discuss MPS algorithms, but the insights apply to other tensor network architectures.

There are a variety of algorithms that can be used to optimize MPS. To find the ground states of 1D Hamiltonians the density matrix renormalization group (DMRG) algorithm is known to be very effective [27]. Similar optimisation algorithms include imaginary-time time evolving block decimation (TEBD) and time dependant variational principle (TDVP), both of which are used in this work [28, 29]. In reference [17] a DMRG-inspired machine learning algorithm is introduced which we adapt for this work. Note that
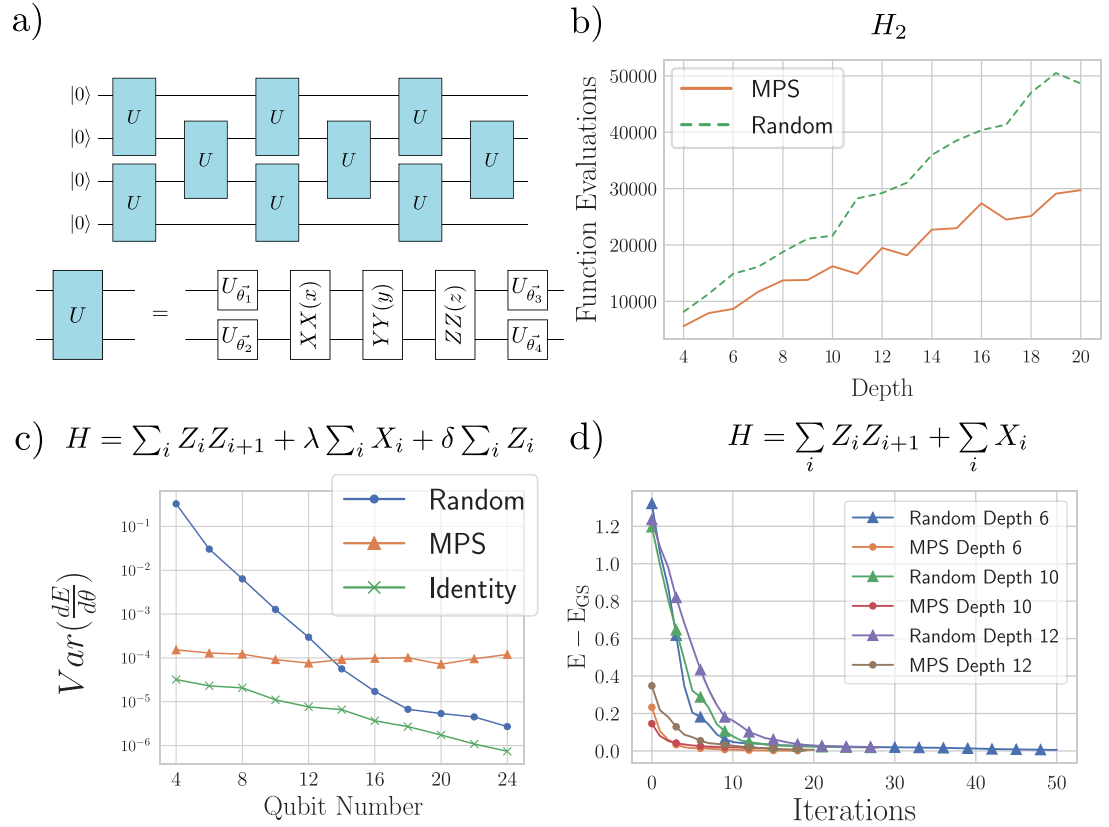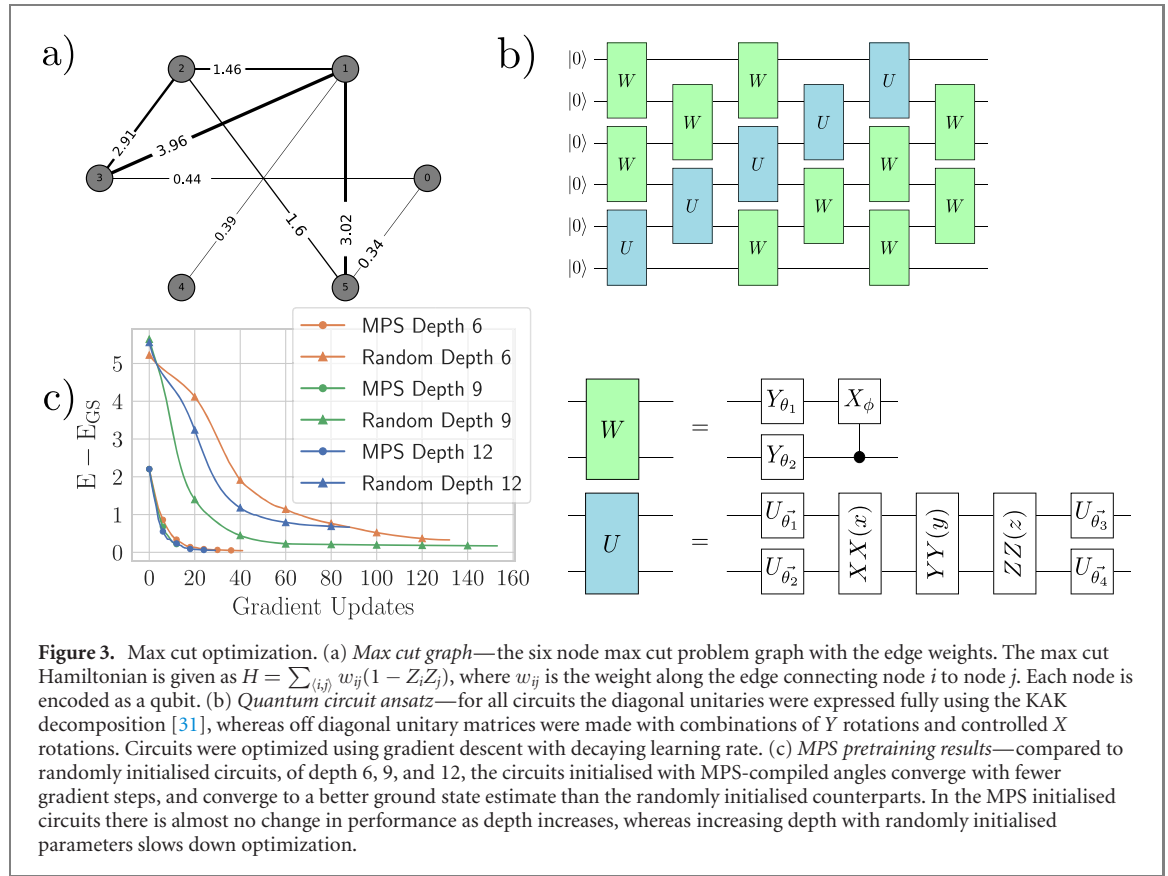
Figure 2. Initialising VQE with MPS. (a) *Quantum circuit ansatz*—the structure of the circuits used for these experiments. Shown is a depth six circuit on four qubits. Each unitary, *U*, is decomposed into 15 rotation angles using the KAK decomposition. The single qubit gates are made up of three independent rotations, *Z* followed by *Y* followed by *Z*. All the angles in the circuit are independent. All optimization was performed using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm. (b) H_2 *optimisation*—for all depths the circuits produced a good estimate for the ground state energy, $E - E_{\min} < 1 \times 10^{-8}$. The MPS initialised circuits required thousands of fewer function evaluations at each depth, and the number of iterations is growing more slowly with depth than for the randomly initialised circuits. (c) *Non-vanishing gradients*—for the tilted-field Ising model we find the variance of the MPS initialised gradients at initialisation do not decrease exponentially with the circuit depth. This is in contrast to both random and identity initialization. We use $\lambda = 1$ and $\delta = 0.5$. (d) *TFIM optimisation*—at all depths tested the MPS initialisation converged after fewer iterations.

in principle gradient descent can always be performed directly on all the tensors in the tensor network. This more closely reflects classical machine learning algorithms. However these quantum-inspired learning algorithms tend to perform as well in practice, and often perform better for quantum-based problems such as energy minimisation.

2.2. MPS compilation

Having trained a MPS to minimize a cost function, the next step is to represent the MPS as a set of rotation angles in a PQC. As opposed to other classical machine learning methods, low bond dimension tensor networks permit efficient representations on quantum circuits [30]. MPS have a gauge freedom which means that any MPS can be brought into *canonical form* in which each tensor is an isometry [22]. These can then easily be embedded in unitary matrices. Bond dimension 2 MPS can be expressed as a staircase of two qubit unitary gates [30]. The rotation angles are extracted from these unitaries with some compilation scheme. For two qubit unitaries we use Cartan's decomposition, often referred to as the KAK decomposition [31], which decomposes two qubit unitaries into four single qubit gates with three rotation angles each, and three different two-qubit interaction gates. A circuit diagram of the KAK decomposition is shown in figure 2(a). In practice any unitary factorization can be used. The KAK decomposition was used in this work to simplify numerics. If necessary these gates can be further decomposed into hardware efficient gates. Restricted gate sets can be used as compilation targets, and the rotation angles can be found variationally [32].

Often brick wall circuits are used as ansatz for PQC research, being the most general way to parametrise a quantum circuit limited to nearest neighbour interactions. To initialise a brick wall circuit with a pretrained MPS, the gates on the diagonal of the brick wall must be initialised with the angles extracted in the compilation of the MPS. All of the off-diagonal gates are initialised to the identity. This ensures that before quantum training begins, the circuit represents exactly as good a candidate for the target state as the

Figure 3. Max cut optimization. (a) *Max cut graph*—the six node max cut problem graph with the edge weights. The max cut Hamiltonian is given as $H = \sum_{\langle i,j \rangle} w_{ij}(1 - Z_i Z_j)$, where w_{ij} is the weight along the edge connecting node i to node j. Each node is encoded as a qubit. (b) *Quantum circuit ansatz*—for all circuits the diagonal unitaries were expressed fully using the KAK decomposition [31], whereas off diagonal unitary matrices were made with combinations of Y rotations and controlled X rotations. Circuits were optimized using gradient descent with decaying learning rate. (c) *MPS pretraining results*—compared to randomly initialised circuits, of depth 6, 9, and 12, the circuits initialised with MPS-compiled angles converge with fewer gradient steps, and converge to a better ground state estimate than the randomly initialised counterparts. In the MPS initialised circuits there is almost no change in performance as depth increases, whereas increasing depth with randomly initialised parameters slows down optimization.

MPS is after compilation is completed. The optimizer is then free to vary the angles of all of the gates. We demonstrate that starting with this initialisation accelerates training, requires fewer gradient updates, and avoids local minima.
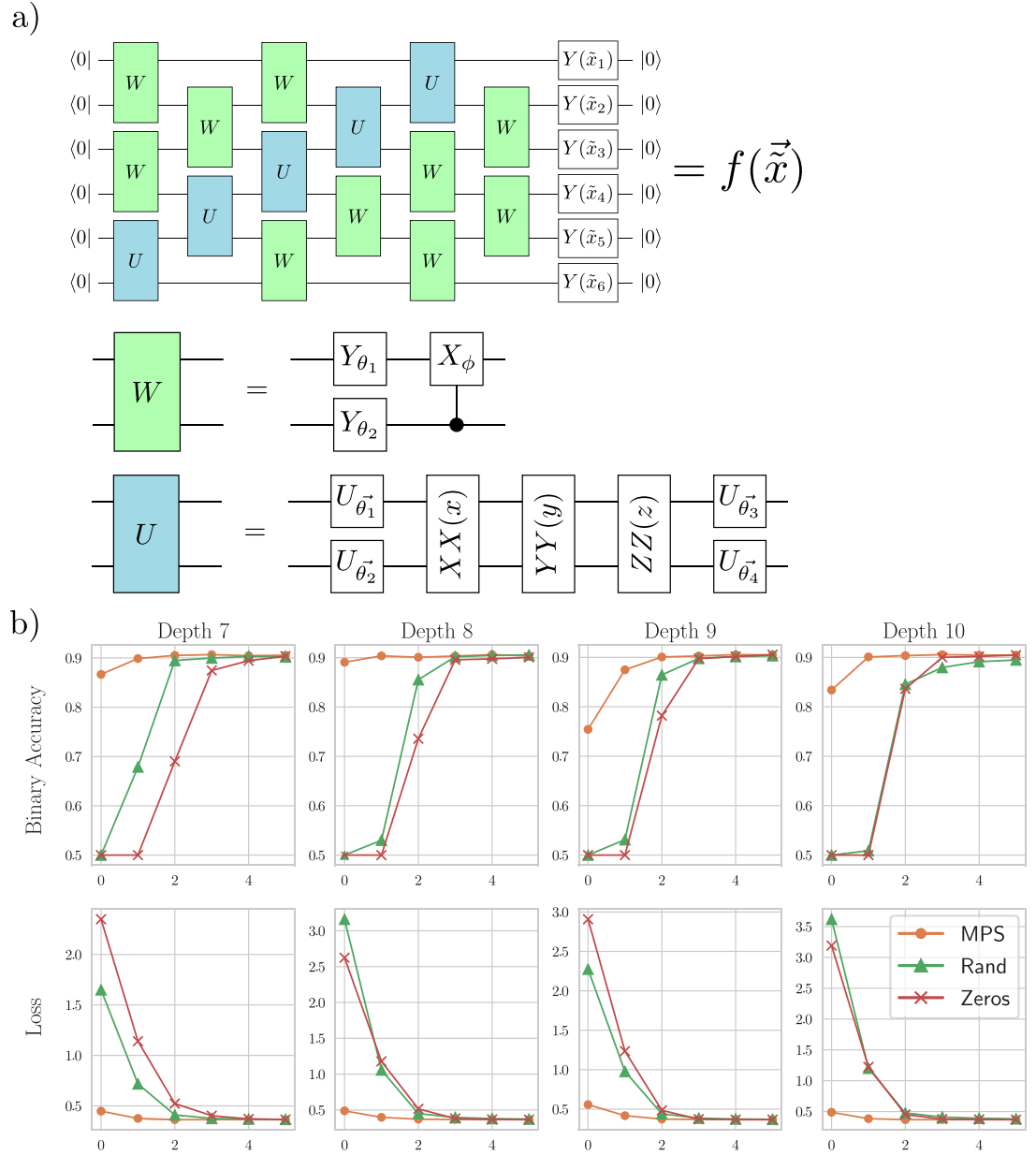
3. Results

3.1. Finding ground states
We implement MPS pretraining for the purpose of finding ground states of electronic Hamiltonians, a common benchmark in hybrid quantum classical algorithm research. We use imaginary time TDVP to estimate the ground state of the electronic Hamiltonian of H_2. The Hamiltonians are constructed using the OpenFermion package [33]. The same is done for the transverse field Ising model (TFIM) which is a widely studied Hamiltonian in condensed matter physics.

In all these cases imaginary time TDVP was used to construct a MPS approximation to the ground state, and this MPS is used to initialise a quantum circuit. To demonstrate that even in cases where low bond dimension MPS are not as effective, we run the TDVP algorithm for a shorter time, meaning a worse approximation is used to initialise the circuits. These results are given in figure 2.

Not only do we see optimization terminating successfully in fewer iterations for all cases studied, we look at the effect of MPS pre-training on the size of gradients at initialisation. Barren plateaus are identified as an exponentially shrinking variance in the initial gradients of the cost function. Figure 2(c) shows that when initialised with a pre-trained MPS, even one that has not fully converged, the variance of gradients do not shrink exponentially with system size. This is in stark contrast to randomly selecting the initial parameters. We see larger gradient variance compared to circuits variationally initialised to the identity matrix, which still show decreasing variance with system size.

3.2. Combinatorial optimization
We test the initialisation scheme on the max cut optimisation problem which is often used as a benchmark for quantum approximate optimization algorithm (QAOA) algorithms [34]. In the max cut problem a graph, $G(E, V)$, is provided along with weights, w_{ij}, on each edge. The task is to find a set of vertices, S, such that the total weight of the edges connecting S to their complement is maximized. This is equivalent to
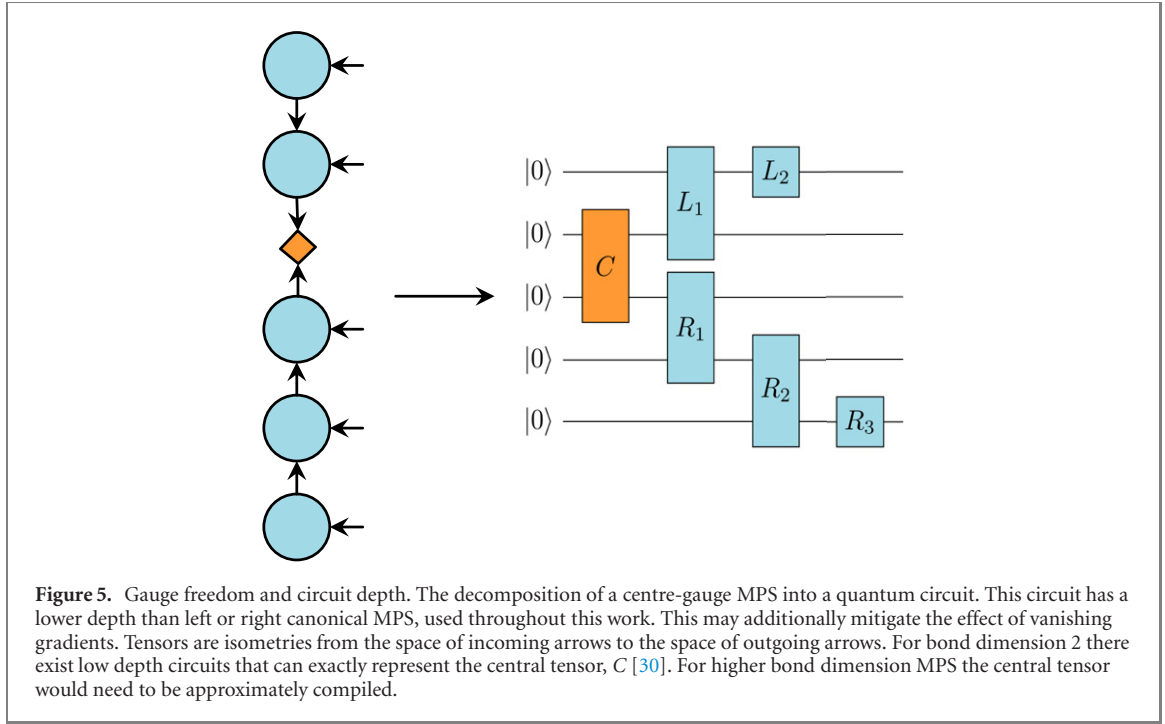
Figure 4. Fashion MNIST results. (a) *Quantum circuit ansatz*—the circuit used in these classification experiments. U and W are defined in the same way as in figure 3(a). The classifier ansatz is given by a brick wall circuit, and the data is introduced using Y rotations, parametrised by angles $\vec{\tilde{x}}$. A single component of this vector, $\tilde{x}_i$, is given by the projection of a single image onto the ith principle component of the training dataset. A decision function is defined as the probability of measuring the all-zero bit string. We use binary cross-entropy loss defined on the training dataset, $\tilde{\mathbf{X}}$, as a target to optimize the decision boundary. The circuit classifies the image with label 0 if $f(\vec{\tilde{x}}) < 0.5$ and 1 otherwise. Optimization is performed using the *Adam* optimizer. (b) *Results with MPS initialisation*—the loss and binary accuracy on the training set during training is given as a function of the number of epochs trained for. An epoch is defined as the number of batched gradient updates required such that the entire training set has been processed once. In each case the number of qubits is the same as the depth. The MPS was trained for five epochs before being compiled to a quantum circuit and training is continued. The MPS initialised circuits took no more than two epochs to optimize, whereas the other initialisation schemes required four or more. As well as starting at a higher binary accuracy, the binary accuracy of the MPS initialised circuits also increased faster during early epochs, indicating that the local loss landscape of the MPS initialised circuits may be more easy to optimize over.

finding the ground state of the Hamiltonian,

$$H = \sum_{\langle i,j \rangle} w_{ij}(1 - Z_i Z_j) \tag{1}$$

where Z is the Pauli Z matrix.

We use imaginary time TEBD to find a bond dimension 2 approximation to the ground state for an instance of the max cut problem on a graph with six vertices. In figure 3 we show the results of training with this initialisation for depth 6, 9, and 12 circuits. Optimization was performed for an imaginary time of

Figure 5. Gauge freedom and circuit depth. The decomposition of a centre-gauge MPS into a quantum circuit. This circuit has a lower depth than left or right canonical MPS, used throughout this work. This may additionally mitigate the effect of vanishing gradients. Tensors are isometries from the space of incoming arrows to the space of outgoing arrows. For bond dimension 2 there exist low depth circuits that can exactly represent the central tensor, C [30]. For higher bond dimension MPS the central tensor would need to be approximately compiled.

3×10^{-2}, with a time step of 1×10^{-3}. We only optimize the MPS for short periods of time because for small problem instances low bond dimension MPS can often get close to the optimal answer. To more closely match the performance on large problem instances which require quantum circuits that cannot be easily simulated we do not fully optimize the MPS.

We compare training starting from MPS pretrained circuits to that starting from random initialisations of the same ansatz. Instead of the standard QAOA scheme, we use a brick wall circuit made up of independent two-qubit gates. These sorts of circuits have been explored for optimisation problems and their performance is competitive with QAOA [35].

The MPS initialised circuits start off at a better energy than the random counterparts, which is to be expected. However the initial steps from the MPS initialised states are noticeably larger than those taken from randomly initialised circuits. These circuits reach a minimum with fewer gradient descent updates, and reach better minima than the any of the randomly initialised states. In the random state there is a notable decrease in performance between the depth 9 and depth 12 ansatz, where the depth 12 gets stuck in a local minimum. No drop in performance is observed in the MPS initialised circuits.

3.3. Machine learning

Finally we pretrain quantum circuits to classify clothing labels using the Fashion MNIST dataset. The images were compressed so small circuits could be used to classify each image. This compression is done using principle component analysis on the training data set, and projecting the training images onto the principle components. The set of training images is collected into a matrix, **X**. We compute the covariance matrix of the training dataset, given by

$$\Sigma = X^{\mathrm{T}}X. \tag{2}$$

Then we perform a singular value decomposition (SVD) on the matrix Σ

$$\Sigma = U\Lambda U^{\dagger}. \tag{3}$$

The principle components are identified as the columns of **U**. To compress an image so that an N qubit circuit can be used to classify the image we take the N top principle components, $\{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_N\}$, and take the inner product between each image (reshaped into a vector) and the principle component. The input to the ith qubit from the jth image in the training set is given by

$$\tilde{x}_{i,j} = \langle \vec{x}_j, \vec{u}_i \rangle. \tag{4}$$

Each projection was used as the input to a single qubit, so to use N qubits, we projected an image onto the N most significant principle components. The individual $\tilde{x}_i$ values are used as rotation angles in parametrised Y gates, figure 4(a). This method, as opposed to other compression methods, such as pooling,

gives greater flexibility in the number of qubits that can be used. The circuits used here were trained as binary classifiers. They were trained to distinguish between t-shirts and trousers. In figure 4(b) we show the training set loss and accuracy at each epoch during training.

The MPS pretrained circuit is compared to both a random initialisation and an initialisation such that the circuit evaluates to the identity matrix. Reference [12] suggests the identity initialisation increases the size of initial gradients and is commonly used in quantum chemistry research. In this case all angles were set to zero, as all gates used were parameterised rotations. For an ansatz with static gates, more care must be taken so that the entire circuit evaluates to the identity.

The MPS initialised circuit has a lower loss and higher accuracy on the training set during training than either the identity initialisation or the random initialisation.

4. Discussion

4.1. Comparison to other methods
Below we compare conceptually similar ideas which have been proposed to overcome barren plateaus and accelerate the training of variational quantum circuits. This is not an exhaustive list but highlighting these examples helps highlight the benefits of MPS pre-training.

4.1.1. Warm-start QAOA
The MPS initialisation method introduced here is one of many proposed methods to improve the training of PQCs. This solution is very similar in methodology to the warm-start QAOA algorithm [36]. Relaxations are applied to the optimization problem which make the problem efficiently solvable, and this is used to initialise the QAOA algorithm. The relative merits of these two methods ultimately depends on the relative effectiveness of relaxation methods and MPS based methods to approximate the optimal solution for optimization problems. There is currently no way to concretely answer this question, but there have been results demonstrating that tensor network based methods are competitive with state of the art commercial solvers for optimization problems [15, 16].

4.1.2. Layer-wise learning
Another procedure that has shown promising results is layer wise learning [35, 37]. Layers in a quantum circuit are trained one at a time, keeping all other layers fixed. Finally layers are grouped together and trained simultaneously before training the entire circuit. The reasoning behind this method is very similar to that proposed here, to initialise the entire circuit with an approximation to the optimal solution, but instead of using MPS, the approximation is generated by sequential optimisation of layers in a PQC. Once again the question remains as to whether the layer wise trained approximation is better than MPS based approximations. Consider the fact that depth 1 or 2 nearest neighbour circuits, which are often considered in layer wise training regimes, can be faithfully represented with a bond dimension 2 MPS, and in fact this set of states is a restriction on the set of states that can be represented with a bond dimension 2 MPS. Layer wise training could be reformulated as sequential training of finite correlation length MPS, with the bond dimension of the restricted MPS growing with each trained layer. Seeing as the MPS initialisation scheme that we have introduced requires no restriction on the set of accessible states it should be the case that better initialisation states are accessible with the methods introduced here. It has been demonstrated that layer-wise learning suffers from abrupt transitions in trainability [38]; there exist circuit ansatz and cost functions where, below a threshold depth, piece wise training fails to minimize the cost function.

It has not escaped our attention that insights from tensor network optimisation techniques could be used to augment this initialisation method with a training scheme similar to layer wise learning. In the DMRG algorithm, the bond dimension of the MPS is gradually increased. A similar approach with brick wall circuits would involve sequentially training diagonals either side of the central diagonal. It remains to be seen if this training scheme could be as effective as layer wise learning in training a circuit combined with MPS pretraining.

4.1.3. Entanglement restriction
The authors of [7] note that the impacts of vanishing gradients can be mitigated by restricting entanglement between hidden and visible qubits in a PQC. A qubit is visible if the output of that qubit is used to calculate a cost function, and it is hidden if it is ignored. They propose a number of schemes, including starting with no entanglement between hidden and visible nodes, having a fixed entanglement, and learning circuits which have low entanglement. The quantum circuit MPS formalism used in this work can easily be extended to the regime with hidden and visible qubits. In this case our initialisation scheme

would resemble a fixed entanglement initialisation scheme, where the entanglement between the hidden and visible nodes is fixed by the bond dimension of the tensor network. They additionally propose including additional terms to the cost function, one to restrict entanglement growth between hidden and visible nodes, and one to add Langevin noise which acts to mitigate the effect of vanishing gradients. Both these schemes could be implementing on top of the initialisation scheme proposed here.

4.2. Alternative tensor network structures

There are many tensor network geometries that are used to represent states with properties that are not effectively captured by MPS. Many of these have circuit counterparts which could be initialised in the same way. For example MERA networks [39] have been proposed as a basis for quantum machine learning.

MPS have a freedom in the location of the *orthogonality centre* when put in canonical form. In all results above, the MPS are put into left or right canonical form. The circuits to represent these states have a depth at least as large as the number of qubits. However choosing a *mixed canonical* form actually reduces the circuit depth needed to initialise the MPS, figure 5.

4.3. Consequences for tensor network simulations

It is interesting to note that the bond dimension of the MPS needed to represent the circuit increases as training proceeds. Deep brick wall circuits represent a restricted class of high bond dimension tensor networks. Ordinarily any variational calculations with these extremely large bond dimension tensor networks would be impractical on quantum devices because of the difficulty in optimizing these circuits. To simulate spin systems with high bond dimension tensor networks it would be possible to simulate up to the classically feasible limit, translate the tensor network into circuits, and then seed a higher bond dimension simulation with the classical tensor network as is done here. This could open up the possibility of very large bond dimension tensor network simulations of spin systems on NISQ devices.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

ORCID iDs

James Dborin ⬤ https://orcid.org/0000-0002-5826-2359
Vinul Wimalaweera ⬤ https://orcid.org/0000-0002-2230-6428

References

[1] Bharti K *et al* 2022 Noisy intermediate-scale quantum algorithms *Rev. Mod. Phys.* **94** 015004
[2] Cerezo M *et al* 2021 Variational quantum algorithms *Nat. Rev. Phys.* **3** 625–44
[3] Tilly J *et al* 2021 The variational quantum eigensolver: a review of methods and best practices (arXiv:2111.05176)
[4] McClean J, Boixo S, Smelyanskiy V N, Babbush R and Neven H 2018 Barren plateaus in quantum neural network training landscapes *Nat. Commun.* **9** 4812
[5] Holmes Z *et al* 2021 Connecting ansatz expressibility to gradient magnitudes and barren plateaus *PRXQuantum* **3** 010313
[6] Marrero C, Kieferov M and Wiebe N 2020 Entanglement induced barren plateaus (arXiv:2010.15968)
[7] Patti T L *et al* 2020 Entanglement devised barren plateau mitigation (arXiv:2012.12658)
[8] Wang S *et al* 2020 Noise-induced barren plateaus in variational quantum algorithms (arXiv:2007.14384)
[9] Ostaszewski M, Grant E and Benedetti M 2019 Quantum circuit structure learning (arXiv:1905.09692)
[10] Koczor B and Benjamin S 2020 Quantum analytic descent (arXiv:2008.13774)
[11] Cerezo M *et al* 2020 Cost-function-dependent barren plateaus in shallow quantum neural networks (arXiv:2001.00550)
[12] Grant E, Wossnig L, Ostaszewski M and Benedetti M 2019 An initialization strategy for addressing barren plateaus in parametrized quantum circuits *Quantum* **3** 214

[13]   Orús R 2014 A practical introduction to tensor networks: matrix product states and projected entangled pair states *Ann. Phys.* **349** 117–58
[14]   Roberts C *et al* 2019 TensorNetwork: a library for physics and machine learning (arXiv:1905.01330)
[15]   Mugel S *et al* 2020 Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks (arXiv:2007.00017)
[16]   Alcazar J and Perdomo-Ortiz A 2021 Enhancing combinatorial optimization with quantum generative models (arXiv:2101.06250)
[17]   Stoudenmire E M and Schwab D 2016 Supervised learning with quantum-inspired tensor networks (arXiv:1605.05775)
[18]   Bradley T, Stoudenmire E M and Terilla J 2019 Modeling sequences with quantum states: a look under the hood (arXiv:1910.07425)
[19]   Cheng S, Wang L and Zhang P 2020 Supervised learning with projected entangled pair states (arXiv:2009.09932)
[20]   Huggins W J *et al* 2018 Towards quantum machine learning with tensor networks (arXiv:1803.11537)
[21]   Grant E *et al* 2018 Hierarchical quantum classifiers *npj Quantum Inf.* **4** 1–8
[22]   Perez-Garcia D *et al* 2006 Matrix product state representations (arXiv:quant-ph/0608197)
[23]   Zhang Z-J, Kyaw T H, Kottmann J S, Degroote M and Aspuru-Guzik A 2021 Mutual information-assisted adaptive variational quantum eigensolver *Quantum Sci. Technol.* **6** 035001
[24]   Fishman M, White S R and Stoudenmire E M 2020 The ITensor software library for tensor network calculations (arXiv:2007.14822)
[25]   Hauschild J and Pollmann F 2018 Efficient numerical simulations with tensor networks: tensor network Python (TeNPy) *SciPost Physics Lecture Notes* (Amsterdam: SciPost) p 5 code available from https://github.com/tenpy/tenpy
[26]   Gray J 2018 quimb: a python package for quantum information and many-body calculations *J. Open Source Softw.* **3** 819
[27]   Schollwöck U 2011 The density-matrix renormalization group in the age of matrix product states *Ann. Phys.* **326** 96–192
[28]   Haegeman J *et al* 2011 Time-dependent variational principle for quantum lattices *Phys. Rev. Lett.* **107** 070601
[29]   Paeckel S *et al* 2019 Time-evolution methods for matrix-product states (arXiv:1901.05824)
[30]   Barratt F *et al* 2020 Parallel quantum simulation of large systems on small quantum computers (arXiv:2003.12087)
[31]   Tucci R R 2005 An introduction to Cartan's KAK decomposition for QC programmers (arXiv:quant-ph/0507171)
[32]   Sharma K, Khatri S, Cerezo M and Coles P J 2020 Noise resilience of variational quantum compiling *New J. Phys.* **22** 043006
[33]   McClean J R *et al* 2020 OpenFermion: the electronic structure package for quantum computers *Quantum Sci. Technol.* **5** 034014
[34]   Farhi E, Goldstone J and Gutmann S 2014 A quantum approximate optimization algorithm (arXiv:1411.4028)
[35]   Liu X *et al* 2021 Layer VQE: a variational approach for combinatorial optimization on noisy quantum computers *IEEE Trans. Quantum Eng.* **3** 1–20
[36]   Egger D J, Mareek J and Woerner S 2020 Warm-starting quantum optimization (arXiv:2009.10095)
[37]   Skolik A *et al* 2020 Layerwise learning for quantum neural networks (arXiv:2006.14904)
[38]   Campos E, Nasrallah A and Biamonte J 2021 Abrupt transitions in variational quantum circuit training *Phys. Rev.* A **103** 032607
[39]   Evenbly G and Vidal G 2011 Tensor network states and geometry *J. Stat. Phys.* **145** 891–918