Quantum Science and Technology

OPEN ACCESS

RECEIVED
8 December 2021

REVISED
4 April 2022

ACCEPTED FOR PUBLICATION
19 April 2022

PUBLISHED
6 May 2022

PAPER

Quantum self-supervised learning

B Jaderberg[1,*,6], L W Anderson[1,6], W Xie[2], S Albanie[3], M Kiffner[1,4] and D Jaksch[1,4,5]

[1] Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, United Kingdom
[2] Visual Geometry Group, Department of Engineering Science, University of Oxford, United Kingdom
[3] Department of Engineering, University of Cambridge, United Kingdom
[4] Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2 117543, Singapore
[5] Institut für Laserphysik, Universität Hamburg, 22761 Hamburg, Germany
[*] Author to whom any correspondence should be addressed.
[6] These authors contributed equally to this work.

E-mail: benjamin.jaderberg@physics.ox.ac.uk

Keywords: variational quantum algorithms, quantum machine learning, self-supervised learning, deep learning, quantum neural networks

Abstract

The resurgence of self-supervised learning, whereby a deep learning model generates its own supervisory signal from the data, promises a scalable way to tackle the dramatically increasing size of real-world data sets without human annotation. However, the staggering computational complexity of these methods is such that for state-of-the-art performance, classical hardware requirements represent a significant bottleneck to further progress. Here we take the first steps to understanding whether quantum neural networks (QNNs) could meet the demand for more powerful architectures and test its effectiveness in proof-of-principle hybrid experiments. Interestingly, we observe a numerical advantage for the learning of visual representations using small-scale QNN over equivalently structured classical networks, even when the quantum circuits are sampled with only 100 shots. Furthermore, we apply our best quantum model to classify unseen images on the *ibmq_paris* quantum computer and find that current noisy devices can already achieve equal accuracy to the equivalent classical model on downstream tasks.

1. Introduction

In the past decade, machine learning has revolutionised scientific analysis, yielding breakthrough results in protein folding [1], black hole imaging [2] and heart disease treatment [3]. At the forefront of this progress is deep learning [4], characterised by the successive application of artificial neural network layers [5, 6]. Notably, its use in computer vision has seen the top-1 accuracy on benchmark datasets such as ImageNet soar from 52% [7] to over 90% [8], fuelled by shifts in the underlying techniques used [9, 10]. However, what has remained consistent in these top performing models is the use of labelled data to supervise the representation learning process. Whilst effective, the reliance on large quantities of human-provided annotations presents a significant challenge as to whether such approaches will scale into the future. Crucially, modern datasets such as the billions of images uploaded to social media are both vast and unbounded in their subject, quickly making the task of labelling unfeasible.

This has reignited interest in an alternative approach, termed *self-supervised learning* [11], which seeks instead to exploit structure in the data itself as a learning signal. Rather than predict human annotations, a model is trained to perform a *proxy task*, that makes use of attributes of the data that can be inferred without labelling. Furthermore, the proxy task should encourage the model to learn representations that capture useful factors of variation in the visual input, such that solving it ultimately correlates with solving tasks of interest after training. Recent progress in the self-supervised learning of visual data has been driven by the success of contrastive learning [12–16], in which the proxy task is differentiating augmented instances of the same image from all other images. Provided the correct choice of augmentations, this produces a model which is invariant to transformations that do not change the semantic meaning of the image, allowing the learning of recognisable features and patterns in unlabelled datasets.

With these techniques, contrastive learning is able to learn visual representations with comparable quality to supervised learning [16, 17], without the bottleneck of labelling. However, it is a fundamentally more difficult task than its supervised counterpart [18], and capturing complex correlations between augmented views requires more training data, more training time and larger network capacity [14, 15]. Therefore, it is important to consider whether emerging technologies can contribute to the growing requirement for more powerful neural networks [19].

Variational quantum algorithms (VQAs) [20], a near term application of quantum computing, are one such new paradigm. While VQAs have been used to solve many types of optimisation problems [21–25], it is their application to supervised learning [26–28], unsupervised learning [29], generative models [30, 31] and reinforcement learning [32–34] which has led to them being referred to as quantum neural networks (QNNs) [35–37]. In theory, the power of these models comes from their access to an exponentially large feature space [27] and ability to represent complex high-dimensional distributions, as formalised by the effective dimension [38]. Importantly, early evidence suggests that quantum models can achieve an advantage over their classical counterparts, yet these works focus only on the supervised learning of either artificial data [36, 39] or simple historical datasets [38]. For example, whilst widely used to study QNNs [40, 41], classical supervised learning of MNIST can already achieve 99.3% top-1 accuracy with a two-layer 784–800 width multi-layer perceptron (MLP) [42]. Thus, it is highly unlikely that this problem would practically benefit from a quantum model with access to a $>2^{50}$ dimensional feature space and careful consideration should be made about whether supervised learning is the best setting to try to achieve quantum advantage. By comparison, self-supervised learning of ImageNet with the widely-used ResNet 50 architecture [43] (with maximum channel width 2048) achieves only 76.5% top-1 accuracy [17]. The necessity for large capacity models means that self-supervised learning may be a better setting in which to seek useful quantum advantage through QNNs [38].

In this work, we construct a contrastive learning architecture in which classical and QNNs are trained together. By randomly augmenting each image in the dataset, our hybrid network learns visual representations which groups different views of the same image together in both classical and Hilbert space. Afterwards, we test the quality of the representations by using them to train a linear classifier, which then makes predictions on an unseen test set. We find that our hybrid encoder, constrained in both size and training time by quantum simulation overheads, achieves an average test accuracy of $(46.51 \pm 1.37)\%$. In contrast, replacing the QNN with a classical neural network of equivalent width and depth results in a model which obtains $(43.49 \pm 1.31)\%$ accuracy. Thus, our results provide the first indication that a quantum model may better capture the complex correlations required for self-supervised learning.

We then apply the best performing quantum model to classify test images on a real quantum computer. Notably, the accuracy achieved using the *ibmq_paris* [44] device equals the best performing classical model, despite significant device noise. This illustrates the capability of our algorithm for real-world applications using current devices, with flexibility to assign more of the encoding to QNNs as quantum hardware improves. While further research is required to demonstrate scalability, our scheme provides a strong foundation for quantum self-supervised learning. Excitingly, given that contrastive learning has also been successfully applied to non-visual data [15, 45–48], our work opens the possibility of using QNNs to learn large, unlabelled datasets across a range of disciplines.

2. Method

2.1. Contrastive learning architecture

Given an unlabelled dataset, the objective of self-supervised learning is to find low dimensional encodings of the images which retain important higher level features. In this work, we train a model to do this by adapting the widely used SimCLR algorithm [17], the steps of which can be seen in figure 1. Firstly for a given image, the data of which is contained within $\vec{x}_i$, we generate two augmentation functions. Each one randomly crops, rotates, blurs and colour distorts the picture, such that two augmented views $\vec{x}_i^1, \vec{x}_i^2$ of the same base image are produced. Importantly, these augmentations still allow for the underlying object to remain visually distinguishable. This enables us to assert that these two views contain a recognisable description of the same class, which we call a positive pair.

Once this positive pair is generated, each view is passed through a set of neural networks. First, an encoder network is applied, which maps the high dimensional input data $\vec{x}_i^1, \vec{x}_i^2$ to low dimensional representations $\vec{y}_i^1, \vec{y}_i^2$. Then the output of the encoder network is passed to the projection head, a small multi-layer-perceptron (MLP) [49] consisting of two fully connected layers. This produces the final representations $\vec{z}_i^1, \vec{z}_i^2$.

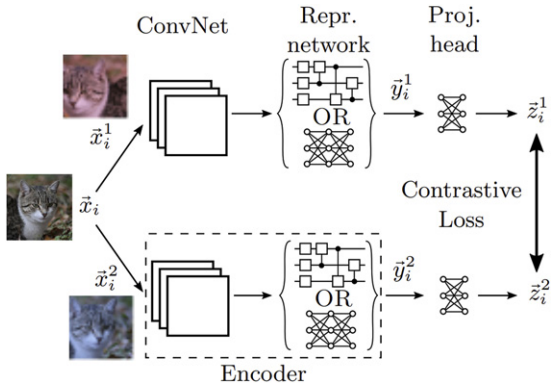

Figure 1. Schematic of the overall neural network architecture and contrastive training method. For each input image $\vec{x}_i$, a pair of random augmentations are generated and applied to form a positive pair $\vec{x}_i^1, \vec{x}_i^2$. These are transformed by the encoder network, consisting of classical convolutional layers and a quantum or classical representation network, into representation vectors $\vec{y}_i^1, \vec{y}_i^2$. The projection head subsequently maps the representations to the vectors $\vec{z}_i^1, \vec{z}_i^2$, such that contrastive loss can be applied without inducing loss of information on the encoder.

Given a batch of N images, the above process is repeated such that we are left with $2N$ representations corresponding to $2N$ augmented views. Looking at all possible pairings of these representations, we have not only positive pairs (e.g., $\vec{z}_i^1, \vec{z}_i^2$) but also negative pairs (e.g., $\vec{z}_i^1, \vec{z}_j^1$ where $i \neq j$), which we cannot definitely say contain the same class. For each training step, all of these possible pairs are used to calculate the normalised temperature-scaled cross entropy loss (NT-Xent) [50] (see appendix A), which is minimised via stochastic gradient descent [51]. Intuitively, minimising this loss function can be understood as training the network to produce representations in which positive pairs are mapped close together and negative pairs far apart, as measured by their cosine similarity. This idea is a core concept in contrastive learning and many machine learning techniques [52]. Note that whilst it is possible to train the network by applying NT-Xent directly to the output of the encoder, the contrastive loss function is known to induce loss of information on the layer it is applied to [17]. Therefore, the addition of the projection head ensures that the encoder remains sensitive to image characteristics (e.g., colour, orientation) that improves performance on downstream tasks.

In order to incorporate QNNs, we modify the encoder to contain both classical and quantum layers working together. The first part of the encoder consists of a convolutional neural network, which in this work is the widely used ResNet-18. This produces a 512 length feature vector, which is already an initial encoding of the augmented image. However, we then extend the encoder with a second network, which we call the representation network as it acts directly on the representation space. This consists of either a multi-layer QNN of width W, or a classical fully connected MLP with equivalent width and depth. Ideally the representation network would have width $W = 512$, so as to minimise loss of information. However, we instead look to work in a regime which is realisable on current quantum computers, and as such in this work we use $W = 8$. This is achieved by following the convolutional network with a single classical layer that compresses the vector, a common technique used to link classical and quantum networks together [53, 54].

After the representation network is applied, the resultant encoding is passed onto the previously described projection head. To maintain the structure of the original SimCLR architecture, we limit the projection head to be no wider than the width of the QNN.

2.2. Quantum representation network

The quantum representation network follows the structure shown in figure 2, beginning with a data loading unitary $\hat{D}(\vec{v})$. Whilst schemes exist to encode data into quantum circuits with exponential compression [55, 56], these require a prohibitively large number of logic gates compared to current hardware capabilities. By compressing the output of the ConvNet as described in section 2.1, we need only to solve the simpler issue of loading a vector $\vec{v}$ of length W into equally as many qubits. This is achieved by applying a single qubit rotation $\hat{R}_x$ to each qubit in the register; $\hat{D}(\vec{v}) = \otimes_{i=1}^{W} \hat{R}_x(v^k)$. Here, v^k is the kth element of input vector $\vec{v}$ and is mapped to the range $[0, \pi]$ to prevent large values wrapping back around the Bloch sphere.

Once the input data is loaded, we apply the learning component of our QNN, a parameterised quantum circuit ansatz. In applications where the ansatz is used to solve optimisation problems relating to a physical system (e.g., the simulation of molecules), the circuit structure and choice of logic gates can be inspired by the underlying Hamiltonian [57]. However, without such symmetries to guide our choice, we use a

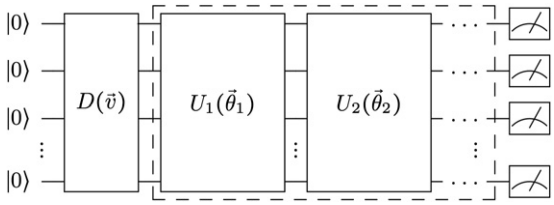

Figure 2. General structure of a QNN. An input vector $\vec{v}$ is encoded into the qubits by a data loading unitary $\hat{D}(\vec{v})$. The variational ansatz consists of layers $\{\hat{U}_1(\vec{\theta}_1), \hat{U}_2(\vec{\theta}_2), \dots\}$ and is parameterised by trainable parameters $\{\vec{\theta}_1, \vec{\theta}_2, \dots\}$. The output of the QNN is taken as the average of repeated measurements in the $\hat{\sigma}_z$ basis.

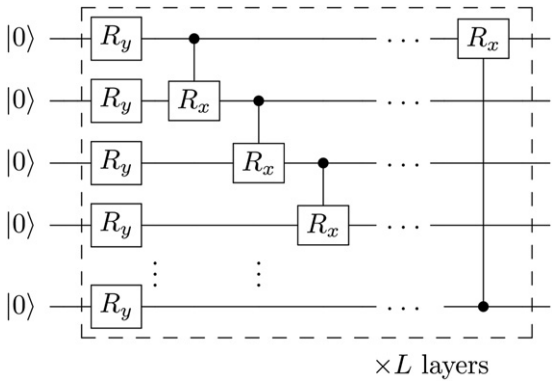

Figure 3. Variational ansatz used in this work. Each layer consists of a single qubit $\hat{R}_y$ rotation on each qubit, followed by controlled $\hat{R}_x$ rotations, connecting the qubits in a ring topology. Every rotation gate is parameterised by a different variational parameter.

variational ansatz based on recent theoretical findings in expressibility and entangling capability [58]. The ansatz is shown in figure 3, the structure of which is derived from circuit 14 of reference [58] and was chosen due to its performance in both these metrics.

After the application of several ansatz layers, the network is finished by measuring each qubit to obtain an expectation value in the $\hat{\sigma}_z$ basis. When evaluated on a real quantum computer or sampling-based simulator, the expectation value is constructed by averaging the sampled eigenvalues over a finite number of shots. If evaluated on a statevector simulator, the expectation value is calculated exactly.

The gradients of the QNN output with respect to the trainable parameters and the input parameters are calculated using the parameter shift rule [35, 59], which we describe here. Consider an observable $\hat{O}$ measured on the state

$$|\psi(\vec{\theta})\rangle = \prod_i \hat{U}_i(\theta_i)\hat{V}_i|0\rangle, \tag{1}$$

resulting from the application of M parameterised gates $\hat{U}_1, \hat{U}_2, \dots, \hat{U}_M$ and M fixed gates $\hat{V}_1, \hat{V}_2, \dots, \hat{V}_M$, where gates $\hat{U}_i = e^{i\theta_i \hat{P}_i/2}$ are generated by operators $\hat{P}_i \in \{\mathbb{1}, \hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z\}^{\otimes n}$ that are tensor products of the Pauli operators. According to the parameter shift rule, the gradient of the expectation value $f = \langle\psi(\vec{\theta})|\hat{O}|\psi(\vec{\theta})\rangle$ with respect to parameter θ_i is given by

$$\frac{\partial f(\vec{\theta})}{\partial \theta_i} = \frac{1}{2}\left[f\left(\theta_i + \frac{\pi}{2}\right) - f\left(\theta_i - \frac{\pi}{2}\right)\right]. \tag{2}$$

For each parameterised gate within the circuit, including both the variational ansatz and data loading unitary, an unbiased estimator for the gradient is calculated by measuring the QNN with the two shifted parameter values given in equation (2).

Once the QNN gradients have been calculated, we combine them with gradients of the classical components to obtain gradients of the loss function with respect to all trainable quantum and classical parameters via backpropagation [60]. In this way, the QNN is trained simultaneously with the classical networks, and the quality of the gradients produced on quantum hardware play a crucial role in the training ability of the whole network.

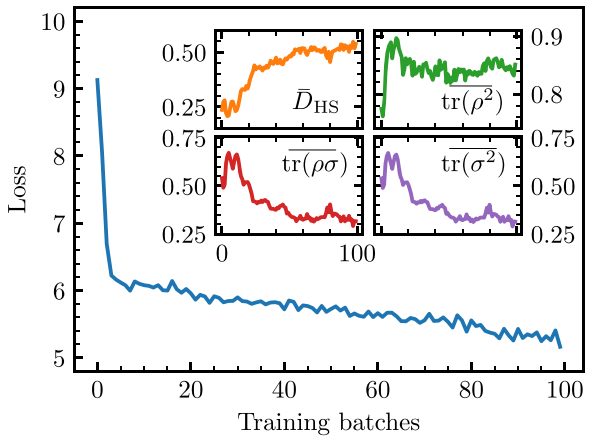

Figure 4. Contrastive learning with a quantum representation network. After each batch of 256 images the loss function (main graph) is recorded, alongside the average Hilbert Schmidt distance between positive and negative pairs $\bar{D}_{\mathrm{HS}}$, the average positive pair clustering $\overline{\mathrm{tr}(\rho^2)}$, the average clustering of all negative pairs $\overline{\mathrm{tr}(\sigma^2)}$ and the ensemble inter-cluster overlap $\overline{\mathrm{tr}(\rho\sigma)}$ (insets).

3. Results

3.1. Training

To examine whether the proposed architecture can successfully train, we apply it to the CIFAR-10 dataset [61]. In this preliminary experiment we restrict the dataset to the first two classes, leaving 10 000 32×32 colour images containing either an aeroplane or automobile. We also train this initial model without a projection head, since it is not being used for classification later. The quantum representation network here is a simulated two-layer QNN and is trained together with the classical components from scratch by integrating [62] the Qiskit [63] and PyTorch [64] frameworks. The full list of training hyperparameters can be found in appendix B.

Figure 4 shows the results of several key metrics after training for 100 batches. Firstly, we record the loss after each batch, the minimisation of which represents the ability to produce representations in the classical W dimensional space whereby positive pairs have high similarity. Our results show that the loss decreases from 9.13 to 5.16 over the course of training, indicating that our model is able to learn. Importantly, since the quantum and classical parameters are trained together, this shows that information is successfully passed both forwards and backwards between these different network paradigms.

Secondly, we log the Hilbert–Schmidt distance (D_{HS}), a metric that has been applied in quantum machine learning previously to study data embedding in Hilbert space [54]. Here, we use it to track the separation between our pseudo classes in the 2^W dimensional quantum state space while optimising the classical loss function. For a given positive pair $\vec{x}_i^1, \vec{x}_i^2$, we calculate the statistical ensembles

$$\rho_i = \frac{1}{2}\left(|\psi_i^1\rangle\langle\psi_i^1| + |\psi_i^2\rangle\langle\psi_i^2|\right), \tag{3a}$$

$$\sigma_i = \frac{1}{2N-2}\sum_{j\neq i}\left(|\psi_j^1\rangle\langle\psi_j^1| + |\psi_j^2\rangle\langle\psi_j^2|\right), \tag{3b}$$

where $|\psi_i^\alpha\rangle$ is the statevector produced by the hybrid encoder given augmented view $\vec{x}_i^\alpha$. The Hilbert–Schmidt distance is then given by

$$D_{\mathrm{HS},i} = \mathrm{sect};((\rho_i - \sigma_i)^2). \tag{4}$$

We repeat this for each positive pair in the batch and record the mean, $\bar{D}_{\mathrm{HS}} = \frac{1}{N}\sum_i D_{\mathrm{HS},i}$. Focusing on the inset of figure 4, we see in the upper-left panel that $\bar{D}_{\mathrm{HS}}$ increases consistently across the range of training, indicating that the QNN successfully learns to separate positive and negative pairs in Hilbert space. Expanding out the quadratic in equation (4), we can break down the metric into the so-called purity terms $\mathrm{tr}(\rho^2)$ and $\mathrm{tr}(\sigma^2)$, which are measures of the intra-cluster overlaps, and the term $\mathrm{tr}(\rho\sigma)$, which is the inter-cluster overlap. Looking at the upper-right panel, we see that the average positive pair clustering $\overline{tr(\rho^2)}$ increases rapidly at the start of training, before steadying at a value around 0.85. This demonstrates one mechanism by which $\bar{D}_{\mathrm{HS}}$ increases, through the QNN producing representations which group positive pairs close together in Hilbert space. The bottom panels of figure 4 show the average negative pair clustering $\overline{tr(\sigma^2)}$ and average negative–positive pair overlaps $\overline{tr(\rho\sigma)}$, which decrease consistently throughout training.

This demonstrates a second behaviour, whereby the QNN produces representations in which negative pairs are well separated. We note that these two values are very similar, which occurs in our self-supervised learning algorithm because of both the need to average over all positive pairs and because of the fixed size of ρ_i. Thus, in the limit $N \to \infty$ the ensemble σ_i contains the entire batch and both metrics are effectively measuring the clustering of all data points.

Overall, figure 4 shows that the quantum component of the encoder contributes to the overall learning process, despite the network's parameters being optimised explicitly in a classical space. It is notable that the training time presented here is significantly less than classical benchmarks, which would typically be 100s of *epochs*. Due to its technological infancy, executing quantum circuits on real or simulated hardware is computationally expensive. Thus, the 1–2 epochs of training used in this work represents the limit of our current experiment, although we expect this to improve dramatically in the coming years with the release of GPU-enhanced simulators [65]. This also justifies our choice of dataset, since CIFAR-10 is both a modern relevant dataset [66–68] yet contains few enough images that we can complete at least one epoch.

3.2. Linear probing

Once training is complete, we require a way to test the quality of the image representations learnt by the encoder. Specifically, a good encoding will produce representations whereby different classes are linearly separable in the representation space [69]. Therefore, we numerically test the encoder using the established linear evaluation protocol [69], in which a linear classifier is trained on the output of the encoder network, whilst the encoder is frozen to stop it training any further. Once this linear probe experiment has trained for 100 epochs, we apply the whole network to unseen test data and record the classification accuracy.

3.3. Quantum and classical results on the simulator

We repeat training, this time with the first five classes of CIFAR-10 and a projection head. We train models with three different types of representation networks; classical MLP with bias and Leaky ReLU activation functions after each layer, quantum trained on a statevector simulator and quantum trained on a sampling-based simulator. We choose the representation networks to be width $W = 8$ in order to minimise the simulation overhead, whilst still being in a compression regime where training is stable (see appendix C). Quantitatively, this means our two-layer classical and quantum representation networks have 144 and 32 learnable parameters respectively.

Figure 5 shows the result of linear probe experiments at checkpoints across 176 batches of contrastive training. We find that when the quantum circuits are evaluated using a statevector simulator, the quantum representation network produces higher average accuracy on the test set than the equivalent classical network at all points probed throughout training, and is separated by more than one standard deviation for over half of these. The highest accuracy is obtained at the end of training, where the quantum model achieves an accuracy of $(46.51 \pm 1.37)\%$ compared to $(43.49 \pm 1.31)\%$ for the classical model. In these results, the confidence interval corresponds to one standard deviation on the mean of six independently trained models. Furthermore, we also find that this numerical advantage holds for a range of smaller width models and is highly dependent on the correct choice of ansatz, more details of which are given in appendices D and E.

Subsequently, we explore whether using a finite number of shots limits this advantage. We train another quantum model on a simulator where the expectation values of measured qubits are sampled from 100 shots, both in the forward pass (generating the representations) as well as the backwards pass (calculating gradients). We find that beyond the first batch, the average accuracy of this model is still above what is achieved by the classical representation network, reaching $(46.34 \pm 2.07)\%$ by the end of training. Significantly, this matches the performance of the statevector simulator, which represents the limit of infinite shots, demonstrating resilience of our scheme to shot noise. However, we note that the additional uncertainty introduced by the sampling does manifest as a larger standard deviation between repeated runs, compromising the consistency of the advantage.

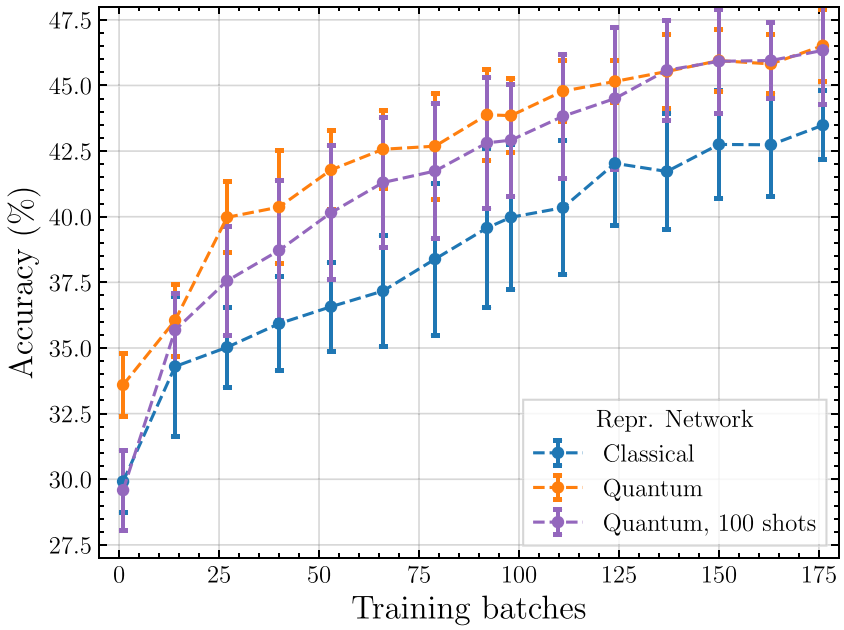

Figure 5. Classification accuracy achieved in linear probing experiments using the encoder at checkpoints across self-supervised training. Comparison between models trained with a classical representation network (blue), quantum representation network evaluated on a statevector simulator (orange) and quantum representation network evaluated on a sampling-based simulator with 100 shots (purple). The markers show the average of six independently trained models, whilst the error bars show one standard deviation.

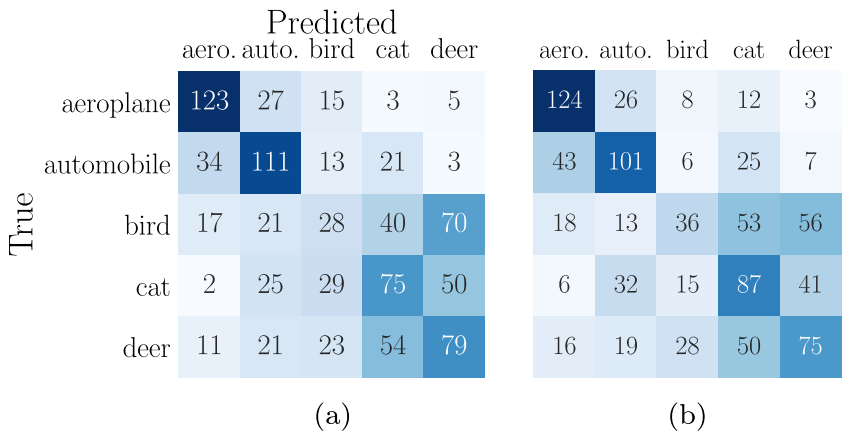

Figure 6. Confusion matrix from classifying 900 images using the best performing (a) classical model evaluated on a classical computer (b) quantum model evaluated on a real quantum computer with 100 shots per circuit. For a given true label (rows) and predicted label (columns), the number in each box shows the total number of times that prediction was made.

3.4. Real device experiments

In section 3.3, we showed that a numerical advantage can be achieved for self-supervised learning with a quantum representation network, even when sampling the quantum circuits with only 100 shots. However, it does not follow that such an improvement can necessarily be realised on current quantum devices. The biggest barrier to this is the complex noise present on quantum hardware, a product of both the finite lifetime that qubits can be held in coherent states for and imperfections in the application of logic gates. To this end, we test the ability of real devices to accurately prepare representations produced by a pretrained quantum model and how this changes downstream accuracy on the test set.

We construct a linear probe experiment with a quantum representation network and load in weights from the best performing pretrained model in which circuits were evaluated with 100 shots. Freezing all of the layers so that the entire network no longer trains, we repeat classification of images from the test set, however this time the circuits are executed on IBM's 27-qubit *ibmq_paris* quantum computer. To reduce the number of gates, particularly SWAP operations caused by a mismatch between the ansatz and physical qubit

connectivities, the circuits are recompiled using incremental structural learning [70] before execution, the details of which can be found in appendix F.

Figure 6(a) shows the result of classifying 900 images randomly sampled from the test set, using the best performing classical model and evaluated on a classical computer. Figure 6(b) shows the result when classifying the same images using the best performing 100-shot quantum model, evaluated on *ibmq_paris*. Overall, the classical and quantum models achieve an accuracy of 47.27% and 47.00% respectively. Excitingly, this demonstrates that in this experiment, error induced by noise on the quantum computer is able to be offset by the enhanced theoretical performance of QNNs, provided the circuit depth is reduced with recompilation techniques. Furthermore, in both setups the most correctly predicted class was aeroplanes (71.1% and 71.7%) whilst the most incorrectly predicted class was birds (15.9% and 20.5%), both of which the quantum model performed better on. We propose that birds and deer were most likely to be mistaken with one another due to the images sharing a common background of the outdoor natural environment.

4. Conclusion

In this work, we propose a hybrid quantum–classical architecture for self-supervised learning and demonstrate a numerical advantage in the learning of visual representations using small-scale QNNs. We train quantum and classical neural networks together, such that encodings are learnt that maximise the similarity of augmented views of the same image in the representation space, as well as implicitly in Hilbert space. After training is complete, we determine the quality of the embedding by tasking a linear probe to classify images from different classes. We find that an encoder with a QNN acting in the representation space achieves higher average test set accuracy than one in which the QNN is replaced by a classical neural network with equivalent width and depth, even when evaluating quantum circuits with only 100 shots. We note that although making such a comparison has been established in previous works [38], how to fairly compare quantum and classical neural networks still remains a significant open question.

We then apply our best performing pretrained classical and quantum models to downstream classification, whereby the quantum circuits were evaluated on a real quantum computer. The observation of a quantum predictive signal with equivalent accuracy to that of the classical model, despite the complex noise present on current quantum devices, is representative of the potential practical benefit of our setup. If recent progress in superconducting qubit hardware continues [71–73], it is possible that QNNs running on real devices will outperform equally sized classical neural networks in the near future in this experiment.

One advantage of the hybrid approach taken in this paper is the resulting flexibility in how much of the encoder is quantum or classical. In fact, there now exist numerous software solutions for producing and testing such hybrid architectures [63, 74, 75]. As the quality and size of quantum hardware improves, our scheme allows classical capacity to be substituted for quantum, eventually replacing ResNet entirely. By optimising directly for the Hilbert–Schmidt distance, it is also possible with a fully quantum encoder to apply our setup to problems in which the data is itself quantum [76–80]. Promisingly, in this regime it may prove that the advantage observed in this work is further extended, given the ability of a quantum model to inherently exploit the dimensionality of the input [81]. Recently developed data sets consisting of entangled quantum states [82, 83] serve as an obvious target for such work. In this case contrastive augmentations could be quantum operations that change the state but conserve the properties of interest, for example LOCC operations that do not affect the amount of entanglement in the system. With classical contrastive learning having been applied to non-visual problems in biology [47] and chemistry [48], our work provides a strong foundation for applying quantum self-supervised learning to fundamentally quantum problems in the natural sciences [84].

An open question remains as to whether a general quantum advantage for self-supervised learning may prove possible [71, 85], in which no classical computer of any size can produce accuracies equal to that of a quantum model. In [38, 86], the authors define *effective-dimension*, a metric measuring the expressive power of classical and QNNs. In general, quantum models are able to achieve a higher effective dimension, and therefore capture a larger space of functions, than classical models with comparable width and number of parameters. Although it does not necessarily increase monotonically, the effective dimension of quantum models can remain larger than classical as the model and data set size are increased. Such behaviour indicates that the expressive power available to QNNs may allow for an advantage over classical neural networks, particularly for a problem such as self-supervised learning where highly expressive, large capacity models are believed to be particularly important for achieving highly accurate predictions [17].

Achieving experimental quantum advantage would require, as a minimum, a QNN with width greater than 60 qubits, such that the dimensionality of the accessible feature space becomes classically intractable. Furthermore, the QNNs would need to be trained on real devices, which remains a challenge due short

qubit lifetimes and low gate fidelity. Therefore, considerable research still remains into the scalability of our scheme, which was only demonstrated at the small sizes feasible on current quantum hardware. Promisingly however, our method can be adapted to use different QNN structures that avoid the scaling issue of barren plateaus [87–89], which could be tested already using more efficient simulators [90]. Looking forward, the rate at which quantum hardware continues to progress provides the possibility of representing intractable distributions using QNNs. In this way, quantum computers may yet push self-supervised learning beyond the performance afforded by classical hardware.

Code availability

The code used to train the models described in this work can be found at https://github.com/bjader/QSSL. The code used to incorporate and train Qiskit quantum neural networks into PyTorch can be found at https://github.com/bjader/quantum-neural-network and is required to build quantum representation networks.

Acknowledgments

BJ, LWA, MK and DJ acknowledge support from the EPSRC National Quantum Technology Hub in Networked Quantum Information Technology (EP/M013243/1) and the EPSRC Hub in Quantum Computing and Simulation (EP/T001062/1). MK and DJ acknowledge financial support from the National Research Foundation, Prime Ministers Office, Singapore, and the Ministry of Education, Singapore, under the Research Centres of Excellence program. WX and SA are supported by EPSRC Grant Seebibyte (EP/M013774/1) and Visual AI (EP/T028572/1).

Data availability statement

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

Appendix A. Contrastive loss function

Here we formally define the process of contrastive learning. Let us have an augmentation function $\xi(\cdot;\ a)$. This augmentation combines cropping, rotation, Gaussian blurring and colour distortion of the image, and the amount by which each of these operations is performed is governed by a list of continuous random variables a. Each time we apply an augmentation, we randomly sample a from a distribution A such that applications of the augmentation function are independent from one another.

For a particular image $\vec{x}_i$, we now have a pair of views $\mathcal{P}_i = \{\xi(\vec{x}_i; a_1), \xi(\vec{x}_i; a_2) | a_1, a_2 \sim A\}$ which came from the same base image. We call this a positive pair. We define the negative pairs as the set of all augmented versions of different images.

During contrastive training, all augmented views within the batch are passed through our architecture. The encoder network $f(\cdot) : \vec{x} \to \vec{y}$ and projection head $g(\cdot) : \vec{y} \to \vec{z}$, are applied to give outputs $\vec{z}_i^{\alpha} = g(f(\xi(\vec{x}_i; a_\alpha)));\ a_\alpha \sim A$ for each of the two arms (labelled by $\alpha = 1, 2$).

For simplicity, we define the NT-Xent loss for each input data separately labelled by index i as follows. The overall loss function corresponds to the sum of these terms over all input images (and correspondingly defined positive and negative pairs). A single term in the loss term is given by

$$\mathcal{L}_i = \log \frac{-\exp\left(\vec{z}_i^{1} \cdot \vec{z}_i^{2}/\tau\right)}{\sum\limits_{\substack{j,k\in\{1,\ldots,N\}\\ \alpha,\beta\in\{1,2\}}} \exp\left(\vec{z}_j^{\alpha} \cdot \vec{z}_k^{\alpha}/\tau\right)}, \tag{A1}$$

where $i = 1, 2, \ldots, N$ labels the input image and $\alpha, \beta = 1, 2$ labels the (arbitrary) distinction between the first and second augmentation making up the positive pair. The overall loss $\mathcal{L}$ is given by the sum over each of i.

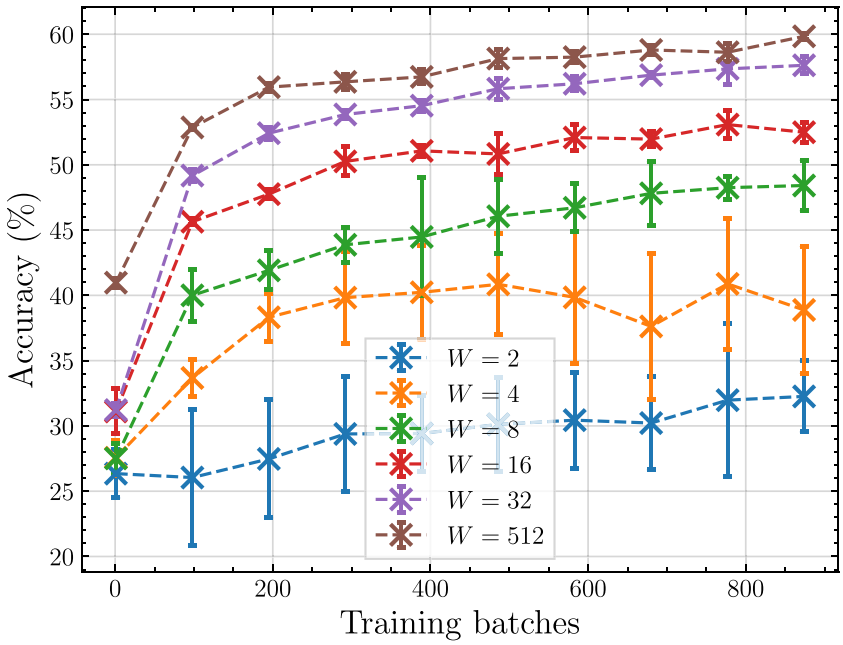

Figure 7. Classification accuracy achieved in linear probing experiments by classical representation networks with varying network widths at checkpoints across self-supervised training. The markers show the average of three independently trained self-supervised models and linear probe experiments, whilst the error bars show one standard deviation.

Appendix B. Training hyperparameters

Throughout this work, the training parameters used are; batch size: 256, optimiser: ADAM [91] with $\beta_1 = 0.9, \beta_2 = 0.999$, learning rate: 10^{-3}, weight decay: 10^{-6} and softmax temperature: 0.07.

Appendix C. Classical width ablation

In order to incorporate QNNs that can be run on current quantum devices into contrastive learning, a compression of the feature vector is required after ConvNet. Since this would not be necessary in a purely classical setting, its impact on final performance is not well understood. To this end, we perform a study of the accuracy achieved by models with different representation network widths. We do this with classical representation networks to remove the quantum specific considerations of statistical noise and optimal circuit architecture, focusing purely on width. The classical representation network is a two-layer, width W MLP, with Leaky ReLu activation functions after each layer and with bias.

Each model is trained on the first five classes of the CIFAR-10 dataset and a linear probe experiment evaluates the performance at regular checkpoints during training. Figure 7 shows the result comparing models with different representation network widths, including the $W = 512$ case which corresponds to no compression. Starting from $W = 2$, we see that increasing the width of the representation network improves the test accuracy. Furthermore, we find that $W = 8$ is the lowest width network in which test accuracy retains the same qualitative behaviour as the uncompressed network. Therefore, in our proof-of-principle quantum experiments, we use an eight width representation network corresponding to eight qubits.

Appendix D. Quantum and classical results at different widths

In section 3.3 we demonstrate that for an architecture with a $W = 8$ representation network, using a QNN to form a hybrid model leads to higher performance in linear probing experiments than the purely classical case. Here we supplement this with additional experiments for the $W = 2$, 4 and 6 cases alongside an equally sized classical comparison for each one. The same problem setup and training parameters are used as in figure 5.

Figure 8 shows the accuracy achieved by these additional models in linear probing experiments at intervals across training, as well as the $W = 8$ results from the main text. Focusing on the circle markers representing the quantum models, we see that the accuracy improves consistently when increasing the QNN width. This matches the behaviour of the classical models, represented in this figure by the crossed markers,

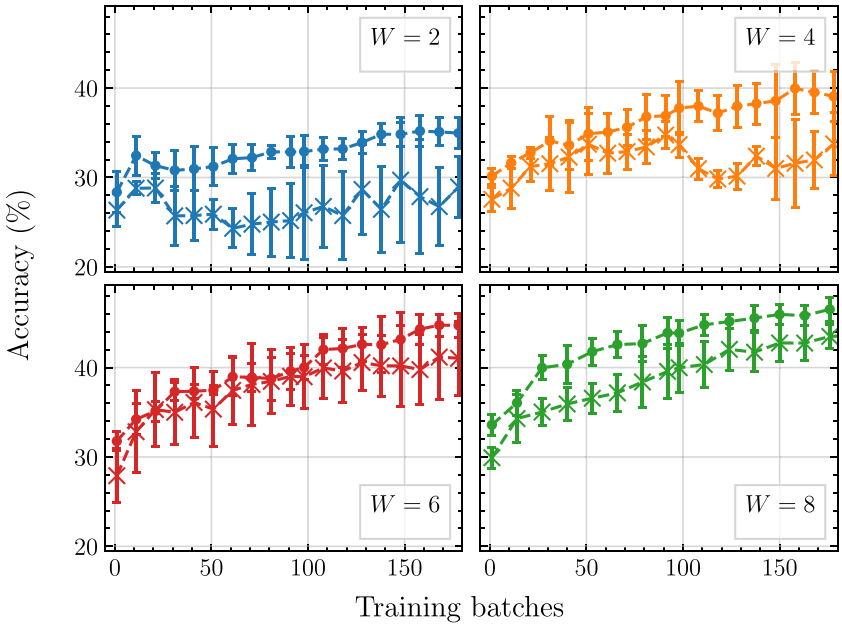

Figure 8. Classification accuracy achieved in linear probing experiments for quantum and classical models with representation network widths 2 (upper left), 4 (upper right), 6 (lower left) and 8 (lower right). Within each graph, the circle (crossed) markers represent models trained with quantum (classical) representation networks. The markers show the average accuracy of three independently trained models, with error bars of one standard deviation, except for the $W = 8$ case which displays the same data as figure 5.

illustrating that our intuition for how compression of the network affects performance can be applied to both the quantum and classical regimes. Secondly, we compare between quantum and classical models of the same width, as shown by the lines of the same colour. Here we see that for the new cases of $W = 2, 4, 6$, there is a numerical improvement in the average accuracy achieved across all training checkpoints sampled, consistent with the $W = 8$ case. Whilst these are still small models, they provide further impetus to consider whether this improvement would remain for models with width $W > 8$, eventually competing directly with the uncompressed SimCLR algorithm at $W = 512$. Looking forward, testing this hypothesis towards the $W = 60$ qubit range may be possible with more efficient simulators [92, 93] as well as by employing training shortcuts such as calculating gradients directly with the quantum state rather than using the parameter shift rule [40].

Appendix E. Performance of alternative ansatz

In section 3 all QNNs are constructed using the variational ansatz seen in figure 3, which connects the qubits in a ring of parameterised controlled rotation gates. Here we introduce a second ansatz, as seen in figure 9, which is different in that it connects all of the qubits together and only has single qubit parameterised gates. Notably, this ansatz was recently shown to exhibit a larger effective dimension when applied to supervised learning than equivalent classical networks [38]. Therefore, we test whether this circuit structure is also a good candidate for improved performance in a self-supervised setting.

We train a model with a quantum representation network structured as the new all-to-all ansatz, simulated on a statevector simulator. The dataset consists of the first five classes of CIFAR-10 and the model is trained with a projection head. Importantly, for a fair comparison, we apply three layers of the all-to-all ansatz, so that it has the same number of learnable parameters as two layers of the ring ansatz. The result of the linear probe experiments can be seen in figure 10, along with the previous models for comparison. We see that for the all-to-all ansatz, test accuracy is no higher than the classical model beyond the statistical variance of repeating training with different initial parameters, and below the ring ansatz. Indeed, by the end of training, the all-to-all ansatz achieves a final accuracy of $(43.46 \pm 1.68)\%$, which is similar to the classical model. Thus, we show that achieving an advantage using QNNs in contrastive learning is highly dependent on the correct choice of quantum circuit structure.

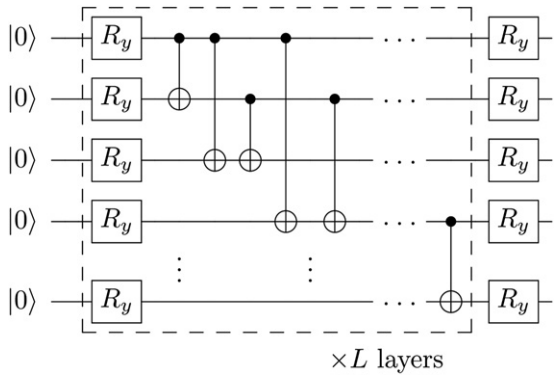

Figure 9. Alternative variational ansatz. Each layer consists of a single qubit $\hat{R}_y$ rotation on each qubit, followed by CNOT gates connecting all qubits to each other. After all layers have been applied, a final set of $\hat{R}_y$ rotations are applied. Every rotation gate is parameterised by a different variational parameter.

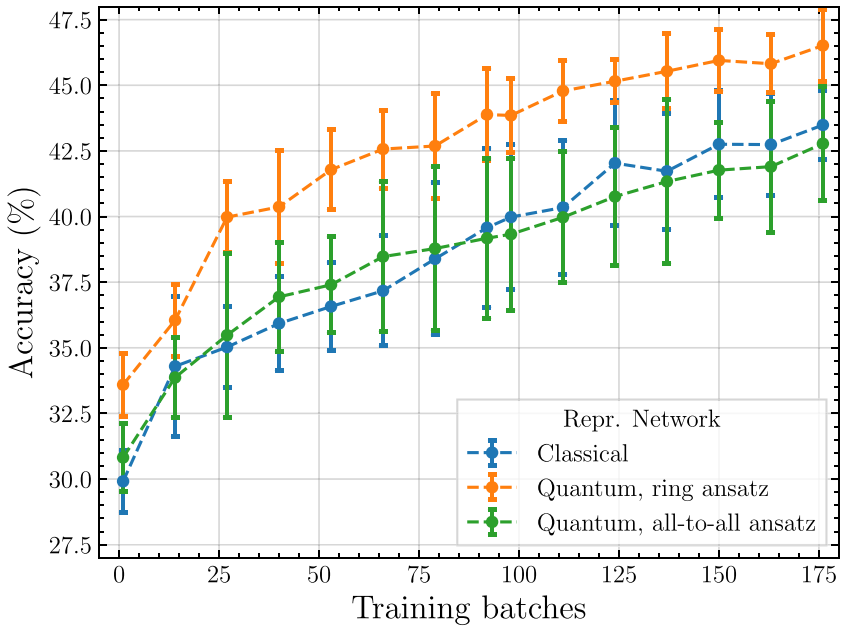

Figure 10. Classification accuracy achieved in linear probing experiments using the encoder at checkpoints across self-supervised training. Comparison between models trained with a classical representation network (blue), quantum representation network with the ring ansatz (orange) and quantum representation network with the all-to-all ansatz (green). All quantum circuits were evaluated on a statevector simulator. The markers show the average of six independently trained models, whilst the error bars show one standard deviation.

Appendix F. Recompilation of quantum neural networks

When executing QNNs on the *ibmq_paris* device, translating the ring topology of our variational ansatz to the honeycomb structure that the qubits are physically connected by requires a significant number of SWAP operations. Quantitatively this increases the number of two-qubit gates in the circuit from 16 to 143, which poses a significant challenge to obtaining a predictive signal beyond random noise since the total circuit error scales exponentially with the number of gates. To mitigate this, for each image evaluated we approximately recompile the QNN using incremental structural learning (ISL) [70], adapted so that only two-qubit connections available on the real device can be applied. Using this method, for over half of the executed circuits, an equivalent circuit is found which produces the same statevector with at least 99% overlap using on average 14 CNOT gates. For the remaining images, we apply ISL once again, but this time without any constraints on the connectivity of the circuit. This produces a shallower equivalent circuit with at least 99% overlap using on average 8 CNOT gates. Although some of these two qubit gates require SWAPs when implemented on the real device, they still represent a significant reduction in the depth of the circuit and total error incurred.

ORCID iDs

B Jaderberg https://orcid.org/0000-0001-9297-0175
M Kiffner https://orcid.org/0000-0002-8321-6768

References

[1] Senior A W *et al* 2020 Improved protein structure prediction using potentials from deep learning *Nature* **577** 706–10
[2] Akiyama K *et al* 2019 First M87 event horizon telescope results: I. The shadow of the supermassive black hole *Astrophys. J.* **875** 1–17
[3] Theodoris C V *et al* 2021 Network-based screen in IPSC-derived cells reveals therapeutic candidate for heart valve disease *Science* **371** 6530
[4] LeCun Y, Bengio Y and Hinton G 2015 *Nature* **521** 436–44
[5] McCulloch W S and Pitts W 1943 A logical calculus of the ideas immanent in nervous activity *Bull. Math. Biophys.* **5** 115–33
[6] Krogh A 2008 What are artificial neural networks? *Nat. Biotechnol.* **26** 195–7
[7] Lin Y, Lv F, Zhu S, Yang M, Cour T, Yu K, Cao L and Huang T 2011 Large-scale image classification: fast feature extraction and svm training *CVPR 2011* (Piscataway, NJ: IEEE) pp 1689–96
[8] Pham H, Dai Z, Xie Q and Quoc V L 2021 Meta pseudo labels *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* pp 11557–68
[9] Lowe D G 1999 Object recognition from local scale-invariant features *Proc. 7th IEEE Int. Conf. Computer Vision* vol 2 (Piscataway, NJ: IEEE) pp 1150–7
[10] Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks *Advances in Neural Information Processing Systems* vol 25 pp 1097–105
[11] Virginia R de S 1994 Learning classification with unlabeled data *Advances in Neural Information Processing Systems* pp 112–9
[12] Wu Z, Efros A A and Stella X Y 2018 Improving generalization via scalable neighborhood component analysis *Proc. European Conf. Computer Vision (ECCV)* pp 685–701
[13] Wu Z, Xiong Y, Yu S and Lin D 2018 Unsupervised feature learning via non-parametric instance-level discrimination (arXiv:1805.01978)
[14] Henaff O 2020 Data-efficient image recognition with contrastive predictive coding *Int. Conf. Machine Learning* (PMLR) pp 4182–92
[15] Oord A v d, Li Y and Vinyals O 2018 Representation learning with contrastive predictive coding (arXiv:1807.03748)
[16] He K, Fan H, Wu Y, Xie S and Girshick R 2020 Momentum contrast for unsupervised visual representation learning *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* pp 9729–38
[17] Chen T, Kornblith S, Norouzi M and Hinton G 2020 A simple framework for contrastive learning of visual representations *Int. Conf. Machine Learning* (PMLR) pp 1597–607
[18] Wang G, Wang K, Wang G, Torr P H S and Lin L 2021 Solving inefficiency of self-supervised representation learning (arXiv:2104.08760)
[19] He K, Chen X, Xie S, Li Y, Dollár P and Girshick R 2021 Masked autoencoders are scalable vision learners (arXiv:2111.06377)
[20] Cerezo M *et al* 2021 Variational quantum algorithms *Nat. Rev. Phys.* **3** 625–44
[21] Peruzzo A, McClean J, Shadbolt P, Yung M-H, Zhou X-Q, Love P J, Aspuru-Guzik A and O'Brien J L 2014 A variational eigenvalue solver on a photonic quantum processor *Nat. Commun.* **5** 4213
[22] AI Quantum G *et al* 2020 Hartree–Fock on a superconducting qubit quantum computer *Science* **369** 1084–9
[23] Farhi E, Goldstone J and Gutmann S 2014 A quantum approximate optimization algorithm (arXiv:1411.4028)
[24] Zhou L, Wang S-T, Choi S, Pichler H and Lukin M D 2020 Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices *Phys. Rev.* X **10** 021067
[25] Ma H, Govoni M and Galli G 2020 Quantum simulations of materials on near-term quantum computers *npj Comput. Mater.* **6** 1–8
[26] Grant E, Benedetti M, Cao S, Hallam A, Lockhart J, Stojevic V, Green A G and Severini S 2018 Hierarchical quantum classifiers *npj Quantum Inf.* **4** 65
[27] Havlíček V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 Supervised learning with quantum-enhanced feature spaces *Nature* **567** 209–12
[28] Schuld M, Bocharov A, Svore K M and Nathan W 2020 Circuit-centric quantum classifiers *Phys. Rev.* A **101** 032308
[29] Otterbach J S *et al* 2017 Unsupervised machine learning on a hybrid quantum computer (arXiv:1712.05771)
[30] Benedetti M, Garcia-Pintos D, Perdomo O, Leyton-Ortega V, Nam Y and Perdomo-Ortiz A 2019 A generative modeling approach for benchmarking and training shallow quantum circuits *npj Quantum Inf.* **5** 1–9
[31] Zoufal C, Lucchi A and Woerner S 2019 Quantum generative adversarial networks for learning and loading random distributions *npj Quantum Inf.* **5** 1–9
[32] Chen S Y-C, Yang C-H H, Qi J, Chen P-Y, Ma X and Goan H-S 2020 Variational quantum circuits for deep reinforcement learning *IEEE Access* **8** 141007–24
[33] Lockwood O and Mei S 2020 Reinforcement learning with quantum variational circuit *Proc. AAAI Conf. Artificial Intelligence and Interactive Digital Entertainment* vol 16 pp 245–51
[34] Saggio V *et al* 2021 Experimental quantum speed-up in reinforcement learning agents *Nature* **591** 229–33
[35] Mitarai K, Negoro M, Kitagawa M and Fujii K 2018 Quantum circuit learning *Phys. Rev.* A **98** 032309
[36] Beer K, Bondarenko D, Farrelly T, Osborne T J, Salzmann R, Scheiermann D and Wolf R 2020 Training deep quantum neural networks *Nat. Commun.* **11** 808
[37] Benedetti M, Lloyd E, Sack S and Fiorentini M 2019 Parameterized quantum circuits as machine learning models *Quantum Sci. Technol.* **4** 043001
[38] Abbas A, Sutter D, Zoufal C, Lucchi A, Figalli A and Woerner S 2021 The power of quantum neural networks *Nat. Comput. Sci.* **1** 403–9
[39] Huang H-Y, Broughton M, Mohseni M, Babbush R, Boixo S, Neven H and McClean J R 2021 Power of data in quantum machine learning *Nat. Commun.* **12** 1–9
[40] Bausch J 2020 Recurrent quantum neural networks *Advances in Neural Information Processing Systems* vol 33

[41] Skolik A, McClean J R, Mohseni M, van der Smagt P and Leib M 2021 Layerwise learning for quantum neural networks *Quantum Mach. Intell.* **3** 1–11
[42] Simard P Y *et al* 2003 Best practices for convolutional neural networks applied to visual document analysis *ICDAR* vol 3
[43] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. IEEE Conf. Computer Vision and Pattern Recognition* pp 770–8
[44] 2021 IBM quantum https://quantum-computing.ibm.com/lab/docs/iql/manage/systems/cite
[45] Mnih A and Kavukcuoglu K 2013 Learning word embeddings efficiently with noise-contrastive estimation *Conf. Neural Information Processing Systems*
[46] Grover A and Leskovec J 2016 Node2vec: scalable feature learning for networks *ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining* (KDD)
[47] Lu A X, Zhang H, Ghassemi M and Moses A M 2020 Self-supervised contrastive learning of protein representations by mutual information maximization *bioRxiv Preprint* https://doi.org/10.1101/2020.09.04.283929
[48] Jaeger S, Fulle S and Turk S 2018 Mol2vec: unsupervised machine learning approach with chemical intuition *J. Chem. Inf. Model.* **58** 27–35
[49] Du K-L and Swamy M N S 2013 *Neural Networks and Statistical Learning* (Berlin: Springer)
[50] Sohn K 2016 Improved deep metric learning with multi-class *n*-pair loss objective *Proc. 30th Int. Conf. Neural Information Processing Systems* pp 1857–65
[51] Robbins H and Monro S 1951 A stochastic approximation method *Ann. Math. Stat.* **22** 400–7
[52] Hadsell R, Chopra S and LeCun Y 2006 Dimensionality reduction by learning an invariant mapping *2006 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'06)* vol 2 (Piscataway, NJ: IEEE) pp 1735–42
[53] Mari A, Bromley T R, Izaac J, Schuld M and Killoran N 2020 Transfer learning in hybrid classical–quantum neural networks *Quantum* **4** 340
[54] Lloyd S, Schuld M, Ijaz A, Izaac J and Killoran N 2020 Quantum embeddings for machine learning (arXiv:2001.03622 [quant-ph])
[55] Le P Q, Dong F and Hirota K 2011 A flexible representation of quantum images for polynomial preparation, image compression, and processing operations *Quantum Inf. Process.* **10** 63–84
[56] Zhang Y, Lu K, Gao Y and Wang M 2013 NEQR: a novel enhanced quantum representation of digital images *Quantum Inf. Process.* **12** 2833–60
[57] Grimsley H R, Economou S E, Barnes E and Mayhall N J 2019 An adaptive variational algorithm for exact molecular simulations on a quantum computer *Nat. Commun.* **10** 3007
[58] Sim S, Johnson P D and Aspuru-Guzik A 2019 Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum–classical algorithms *Adv. Quantum Tech.* **2** 1900070
[59] Schuld M, Bergholm V, Gogolin C, Izaac J and Killoran N 2019 Evaluating analytic gradients on quantum hardware *Phys. Rev.* A **99** 032331
[60] LeCun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition *Proc. IEEE* **86** 2278–324
[61] Krizhevsky A and Hinton G 2009 *Learning Multiple Layers of Features from Tiny Images* (University of Toronto)
[62] Jaderberg B and Anderson L W 2021 Quantum neural network: for building quantum neural networks in QISKIT and integrating with pytorch (https://github.com/bjader/quantum-neural-network)
[63] Aleksandrowicz G *et al* 2019 Qiskit: an open-source framework for quantum computing
[64] Paszke A *et al* 2019 Pytorch: an imperative style, high-performance deep learning library (arXiv:1912.01703)
[65] Patti T L, Kossaifi J, Anandkumar A and Yelin S F 2021 Variational quantum optimization with multi-basis encodings (arXiv:2106.13304)
[66] Huang Y *et al* 2019 GPIPE: efficient training of giant neural networks using pipeline parallelism *Advances in Neural Information Processing Systems* vol 32 pp 103–12
[67] Cubuk E D, Zoph B, Mane D, Vasudevan V and Le Q V 2019 Autoaugment: learning augmentation strategies from data *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* pp 113–23
[68] Huu Phong N and Ribeiro B 2020 Rethinking recurrent neural networks and other improvements for image classification (arXiv:2007.15161)
[69] Kolesnikov A, Zhai X and Beyer L 2019 Revisiting self-supervised visual representation learning *Proc. IEEE Conf. Computer Vision and Pattern Recognition* pp 1920–9
[70] Jaderberg B, Agarwal A, Leonhardt K, Kiffner M and Jaksch D 2020 Minimum hardware requirements for hybrid quantum–classical DMFT *Quantum Sci. Technol.* **5** 034015
[71] Arute F *et al* 2019 Quantum supremacy using a programmable superconducting processor *Nature* **574** 505–10
[72] Kjaergaard M, Schwartz M E, Braumüller J, Krantz P, Wang J I-J, Gustavsson S and Oliver W D 2020 Superconducting qubits: current state of play *Annu. Rev. Condens. Matter Phys.* **11** 369–95
[73] Jurcevic P *et al* 2021 Demonstration of quantum volume 64 on a superconducting quantum computing system *Quantum Sci. Technol.*
[74] Bergholm V *et al* 2018 Pennylane: automatic differentiation of hybrid quantum-classical computations (arXiv:1811.04968)
[75] Broughton M *et al* 2020 Tensorflow quantum: a software framework for quantum machine learning (arXiv:2003.02989)
[76] Sentís G, Calsamiglia J, Munoz-Tapia R and Bagan E 2012 Quantum learning without quantum memory *Sci. Rep.* **2** 1–8
[77] Alvarez-Rodriguez U, Lamata L, Escandell-Montero P, Martín-Guerrero J D and Solano E 2017 Supervised quantum learning without measurements *Sci. Rep.* **7** 1–9
[78] Amin M H, Andriyash E, Rolfe J, Kulchytskyy B and Melko R 2018 Quantum Boltzmann machine *Phys. Rev.* X **8** 021050
[79] Gong M *et al* 2022 Quantum neuronal sensing of quantum many-body states on a 61-qubit programmable superconducting processor (arXiv:2201.05957)
[80] Szołdra T, Sierant P, Lewenstein M and Zakrzewski J 2022 Unsupervised detection of decoupled subspaces: many-body scars and beyond (arXiv:2201.07151)
[81] Sentís G, Monras A, Muñoz-Tapia R, Calsamiglia J and Bagan E 2019 Unsupervised classification of quantum data *Phys. Rev.* X **9** 041029
[82] Perrier E, Youssry A and Ferrie C 2021 Qdataset: quantum datasets for machine learning (arXiv:2108.06661)
[83] Schatzki L, Arrasmith A, Coles P J and Cerezo M 2021 Entangled datasets for quantum machine learning (arXiv:2109.03400)
[84] Cong I, Choi S and Lukin M D 2019 Quantum convolutional neural networks *Nat. Phys.* **15** 1273–8

[85] Zhong H-S et al 2020 Quantum computational advantage using photons Science 370 1460–3
[86] Abbas A, Sutter D, Figalli A and Woerner S Effective dimension of machine learning models (arXiv:2112.04807)
[87] Pesah A, Cerezo M, Wang S, Volkoff T, Sornborger A T and Coles P J 2021 Absence of barren plateaus in quantum convolutional neural networks Phys. Rev. X 11 041011
[88] Grant E, Wossnig L, Ostaszewski M and Benedetti M 2019 An initialization strategy for addressing barren plateaus in parameterized quantum circuits Quantum 3 214
[89] Cerezo M, Sone A, Volkoff T, Cincio L and Coles P J 2021 Cost function dependent barren plateaus in shallow parameterized quantum circuits Nat. Commun. 12 1–12
[90] Luo R 2021 Quantum software benchmarks (https://github.com/yardstiq/quantum-benchmarks)
[91] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)
[92] Luo X-Z, Liu J-G, Zhang P and Wang L 2020 Yao.JL: Extensible, efficient framework for quantum algorithm design Quantum 4 341
[93] Suzuki Y et al 2021 Qulacs: a fast and versatile quantum circuit simulator for research purpose Quantum 5 559