Negational Symmetry of Quantum Neural Networks for Binary Pattern Classification

Nanqing Dong
Department of Computer Science
University of Oxford
nanqing.dong@cs.ox.ac.uk

Michael Kampffmeyer
Department of Physics and Technology
UiT The Arctic University of Norway
michael.c.kampffmeyer@uit.no

Aleks Kissinger
Department of Computer Science
University of Oxford
aleks.kissinger@cs.ox.ac.uk

Irina Voiculescu
Department of Computer Science
University of Oxford
irina.voiculescu@cs.ox.ac.uk

Eric Xing
Machine Learning Department
Carnegie Mellon University
epxing@cs.cmu.edu

Abstract

Entanglement is a physical phenomenon, which has fueled recent successes of quantum algorithms. However, for the time being, the effect of entanglement in quantum neural networks (QNNs) and the behavior of QNNs in binary pattern classification are inconclusive. In this work, we present *negational symmetry* of QNNs in both quantum binary classification and quantum representation learning. We theoretically analyze the negational symmetry and empirically evaluate it in binary pattern classification tasks on the MNIST dataset using Google's quantum framework. Our findings imply that quantum computing could be a new research direction for binary signal processing.

1 Introduction

A key feature that is unique to quantum computing is the *entanglement* [35] between two *qubit*s. By harnessing entanglement, a *spooky* physical phenomenon described by Albert Einstein, quantum computers can solve problems which are out of reach of classical computers. Generally, the statistics of data generated in a classical system is unable to match the statistics of data generated by a quantum system [25]. This phenomenon is characterized as one aspect of *quantum supremacy* [32] or *quantum advantage* [25]. The quantum supremacy experiment [1] on near-term noisy intermediate-scale quantum (NISQ) [32] devices has marked the beginning of a new computing era. One emerging research area is quantum machine learning (QML) [37, 6], which aims to understand how to devise and implement machine learning (ML) algorithms on quantum computers. Fueled by the breakthroughs in deep learning theories [34, 11, 17] and applications [21, 39, 14], quantum neural networks (QNNs) [12, 29, 9, 18] have been receiving increasing attention. Researchers are eager to peep into the blackbox of QNNs and to analyze the properties of these models.

In a 2-D Cartesian system, given a function f and a variable x if we have $f(x) = f(-x)$, then we say f is symmetric to the line $x = 0$ or f has reflectional symmetry. If we have $f(x) = -f(-x)$, then we say f is symmetric to the origin point $(0, 0)$ or f has rotational symmetry. The same logic applies

Tech Report.

in high-dimensional systems with function $f : \mathbf{R}^N \mapsto \mathbf{R}$ and N-dimensional vector $\boldsymbol{x}$. After briefly reviewing the basic definition of geometric symmetry, we move to the quantum domain. For binary classification and the auxiliary representation learning tasks with input from a binary pattern, we find that QNNs also feature two kinds of symmetries. We define this property as *negational symmetry* of QNNs. In a quantum setting, a binary pattern contains only $|0\rangle$ and $|1\rangle$, which is analogous to 0 and 1 (or black and white) in a classical setting. Here, the term *negational* refers to the negation in bitwise operation or logical operation. The negational transformation of a binary pattern is equivalent to apply a `NOT` gate (negation operation) to all bits, i.e. flip all bits. The theoretical discussion on negational symmetry leads to two major findings. First, given a (quantum) binary pattern, a QNN with Z-measurement can make the same prediction for a binary pattern and its negational counterpart. Second, with the proposed quantum representation learning, we find the learned feature vector of a binary pattern is the opposite vector of the learned feature vector of the negational binary pattern. We theoretically discuss the negational symmetry property and empirically evaluate it in the context of the MNIST [22] image classification task. In addition, for a comprehensive understanding of QNNs and a fair comparison between QNNs and classical models, we conduct experiments in a controllable environment. Our experimental results show that QNNs could have unparalleled advantages over classical models when trained on the training set of binarized MNIST images and tested on the negational test set. Our study also suggests a new research direction in binary signal processing.

Our contributions are threefold:

1. We formalize, prove, and analyze the negational symmetry of QNNs in quantum binary classification and quantum representation learning.
2. We evaluate the negational symmetry in binary pattern classification on a quantum simulator.
3. We demonstrate the practical value of negational symmetry by comparing QNNs and classical models in a systematic way.

2 Preliminaries

2.1 Classical Neural Networks

There are two types of classical neural networks (NNs) for image classification, deep neural networks (DNNs) and convolutional neural networks (CNNs). A DNN or a multi-layer feed-forward network is a neural network composed of multiple fully-connected layers between the input and output layers. Each hidden layer is followed by an activation function to introduce non-linearity into the learning system. As stated in the Universality Theorem of neural networks [11, 17], any continuous function can be approximated by a DNN when there are enough nodes and at least one hidden layer, which is the theoretical foundation of deep learning.

CNN is a more advanced variant of DNN. Compared with DNNs, the hidden layers of CNNs can additionally include convolutional layers and pooling layers. Given a fixed size of the receptive field, local information are computed as the weighted average by the the convolution operation before the activation function. Analogous to the number of hidden nodes in DNNs, the number of feature maps is an important hyperparameters for the neural architecture design within a convolutional layer. Another important feature of CNNs is the global pooling operation, including global average pooling and global max pooling, which can aggregate local information for each feature channel.

In supervised learning, given a dataset $\mathcal{D}$ with images $\{\boldsymbol{x}_j\}_{j=1}^n$ and labels $\{y_j\}_{j=1}^n$, the empirical risk is defined as

$$\mathcal{R}_{\mathcal{D}} = \frac{1}{n}\sum_{j}^{n} \mathcal{L}(f_{\boldsymbol{\theta}}(x_j), y_j), \tag{1}$$

where $\mathcal{L}$ is the loss function (e.g. for a binary classification task, $\mathcal{L}$ is the binary cross entropy) and f is a NN classifier with free parameters $\boldsymbol{\theta}$. The prediction $f_{\boldsymbol{\theta}}(x)$ is mapped into a binary label by thresholding on a real value τ. The optimization goal for $\boldsymbol{\theta}$ is to minimize $\mathcal{R}$ by mini-batch gradient descent. The update rule of $\boldsymbol{\theta}$ from iteration t to iteration $t+1$ is

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \nabla_{\boldsymbol{\theta}_t} \mathcal{R}_{\mathcal{D}_{\text{mini}}} \tag{2}$$

where η_t corresponds to the stepsize and $\nabla_{\boldsymbol{\theta}_t} \mathcal{R}_{\mathcal{D}_{\text{mini}}}$ is the empirical estimate of the mini-batch gradient.

2.2 Variational Quantum Circuit

A variational quantum circuit (VQC) is a quantum circuit model that consists of a set of parametric quantum gates [33]. In the near term, a VQC is implemented through the hybrid quantum-classical (HQC) framework. In the HQC framework, a QML task is divided into two subtasks. The first subtask is to apply quantum gates to manipulate qubits in a quantum computer. This quantum process is analogous to the forward pass in a DNN. The second subtask is to optimize the parameters of quantum gates in a classical computer. This classical process is analogous to the backpropagation in a DNN. It has been shown that nonlinear functions can be approximated by VQCs [7, 30], which demonstrates the potential values of VQCs in solving practical problems.

3 Quantum Binary Classification

3.1 Problem Formulation

For simplicity and without loss of generality, we address the binary classification problem, which is a fundamental task in QML [12]. In this work, a quantum system is a composite of two systems, namely the *input register* and the *output register*. Given a training dataset $\mathcal{D}_S = \{(|\boldsymbol{x}_j\rangle, y_j)\}_{j=1}^{n}$, $|\boldsymbol{x}\rangle = |x_1\rangle \otimes |x_2\rangle \cdots \otimes |x_N\rangle$ is a N-qubit quantum state for the input register, where $|x\rangle = \alpha|0\rangle + \beta|1\rangle, \alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1$. $y \in \{-1, 1\}$ is the binary label.[1] The output register is just a *readout* qubit. We prepare the readout qubit as $|1\rangle$. So the input state of the quantum system is $|1, \boldsymbol{x}\rangle = |1\rangle \otimes |\boldsymbol{x}\rangle$.

Let the unitary $U_{\boldsymbol{\theta}}$ be a QNN with parameters $\boldsymbol{\theta}$. A forward pass of the input state $|1, \boldsymbol{x}\rangle$ through $U_{\boldsymbol{\theta}}$ produces the output state $U_{\boldsymbol{\theta}}|1, \boldsymbol{x}\rangle$. In the traditional quantum circuit models, only the readout qubit is measured by a Hermitian operator $\mathcal{M}$, which is a quantum observable. We limit our choice of $\mathcal{M}$ within Pauli operators, specifically $\mathcal{M} \in \{X, Y, Z\}$.[2] As a standard practice in quantum computing, we use Z measurement as the default measurement in this study. The measurement outcome will be either -1 or 1 with uncertainty. When the output state $U_{\boldsymbol{\theta}}|\boldsymbol{x}, 1\rangle$ is prepared for multiple times, the prediction is defined as the expectation of the observed measurement outcomes. Formally, we have

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \langle 1, \boldsymbol{x}| U_{\boldsymbol{\theta}}^{\dagger} |\mathcal{M}_0| U_{\boldsymbol{\theta}} |1, \boldsymbol{x}\rangle \tag{3}$$

$$= \mathrm{tr}(|U_{\boldsymbol{\theta}}|1, \boldsymbol{x}\rangle \langle 1, \boldsymbol{x}| U_{\boldsymbol{\theta}}^{\dagger} |\mathcal{M}_0) \tag{4}$$

where, for simplicity, $\mathcal{M}_0$ denotes the measurement on the readout qubit instead of the whole system,[3] $-1 \leq f_{\boldsymbol{\theta}}(\boldsymbol{x}) \leq 1$ is a real number and $\mathrm{tr}(\cdot)$ is the trace. The decision boundary on the space of density matrices is the hyperplane $\mathrm{tr}(|U_{\boldsymbol{\theta}}|1, \boldsymbol{x}\rangle \langle 1, \boldsymbol{x}| U_{\boldsymbol{\theta}}^{\dagger} |\mathcal{M}_0 = \tau)$, where we set $\tau = 0$.

If we take $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ as the logit for $\boldsymbol{x}$, together with the label y, we can define the loss $\mathcal{L}$ in Eq. 1. We have

$$\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = \max(0, 1 - y \cdot f_{\boldsymbol{\theta}}(\boldsymbol{x})). \tag{5}$$

Considering $-1 \leq f_{\boldsymbol{\theta}}(\boldsymbol{x}) \leq 1$, we choose the hinge loss over the binary cross-entropy (BCE) loss for convenience. Indeed, the choice of loss function does not influence the conclusion of this study.[4]

3.2 Architecture

Previous studies [12, 29, 23, 33, 43, 24] on QNNs utilize different sets of single-qubit quantum gates along with two-qubit entanglement gates to construct VQCs. Unlike a boolean function, there is no *universal set* of quantum gates for nonlinear functions. Based on *ZX-calculus* [8], we can prove that

[1] In this work, we define y as an integer in a HQC system. In a pure quantum system, y can also be defined as $|y\rangle \in \{|0\rangle, |1\rangle\}$.

[2] See Appendix A.3 for details.

[3] Mathematically, the measurement on the whole system should be the tensor product of $N+1$ Pauli operators. A simple example could be $\mathcal{M} \otimes \underbrace{I \otimes I \otimes \cdots \otimes I}_{N}$.

[4] Another common choice of loss function is the fidelity loss $\mathcal{L}_{fidelity} = 1 - fidelity$. The fidelity is defined as $F(\rho_1, \rho_2) = \mathrm{tr}(\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}})^2$, where ρ_1 and ρ_2 are two density matrices. The fidelity loss gives similar results as the hinge loss and BCE but the state preparation and the backpropagation require extra caution in implementation.

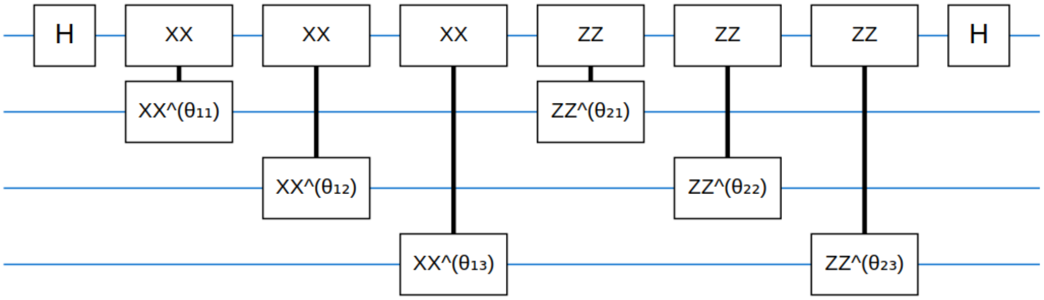


Figure 1: The architecture of a 2-layer QNN with 3 + 1 qubits. The input register has 3 data qubits, represented by the second, third and fourth lines from the top. The output register has 1 readout qubit, represented by the top line. The first layer is formed by X-parity gates and the second layer is formed by Z-parity gates. θ_{jk} stands for the parameter of the quantum gate operated between the readout qubit and the kth data qubit at the jth layer. H is the Hadamard gate, where $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

any nonlinear function can be ϵ-approximated[5] with single-qubit parametric Z-gates ($R_z(\theta)$) and X-gates ($R_x(\theta)$), and two-qubit non-parametric CNOT gates.[6] In this study, to illustrate the impact of entanglement, we use Z-parity gates ($ZZ^\theta = e^{-i\theta\sigma_z \otimes \sigma_z}$) and X-parity gates ($XX^\theta = e^{-i\theta\sigma_x \otimes \sigma_x}$) alternatively to build entanglement between the readout qubit and each of the data qubit [12], where θ is the parameter that we want to optimize. Two-qubit ZZ or XX interactions are also know as *Ising interactions* in statistical mechanics. Each block of parity gates can be viewed as a layer in DNNs. See Fig. 1 for the illustration of the architecture.

3.3 Optimization

In the near term, the number of parameters is limited by the number of qubits, which is the main challenge for most quantum applications. We choose a gradient-based optimization method because gradient-free algorithms cannot scale up to a larger number of parameters in the long term. For a VQC, the mini-batch gradient has an analytic derivation (based on the chain rule) and a numerical implementation (considering the stochasticity of quantum mechanics). This is the most characteristic difference between the optimization of a QNN and a classical NN [36]. Although the analytic gradient is fast to compute in a classical environment, the numerical gradient is more robust in a noisy real-world quantum computer. In real quantum applications, the gradient can be approximated by using the *parameter shift rule* [10]. Given an example pair $(|\boldsymbol{x}\rangle, y)$, we define the numerical gradient for a scalar parameter θ as

$$\nabla_\theta \mathcal{L}(f_\theta(\boldsymbol{x}), y) = \frac{\mathcal{L}(f_{\theta+\frac{\pi}{2}}(\boldsymbol{x}), y) - \mathcal{L}(f_{\theta-\frac{\pi}{2}}(\boldsymbol{x}), y)}{2}. \tag{6}$$

It is worth mentioning that QNNs have two theoretical advantages in optimization compared with classical NNs. First, one bottleneck of training classical NNs is the *vanishing gradient problem* [15] caused by the activation functions in backpropagation. There is no explicit activation function in QNNs. Second, quantum gates are mathematically represented by *unitary* matrices. [12] first pointed out that using unitary transformation can prevent the *exploding gradient problem* [5].

3.4 Measurement On Data Qubits

In the context of deep learning, *representation learning*, also known as *feature learning*, is a rapidly developing area, *with the goal of yielding more abstract and ultimately more useful representations* of the data, as described by [4]. The composition of multiple non-linear transformations of the data has been used to quantitatively and qualitatively understand the black-box of NNs. For example, in CNNs, the feature maps of lower layers tend to catch the similar basic patterns and the feature maps of higher layers are able to extract the semantic information. However, limited by the physical

[5] Given a function f and an approximation function f^*, we have $|f^*(x) - f(x)| < \epsilon$ where $\epsilon > 0$.

[6] See Appendix A.5 for a sketch of proof.

implementation, it is impractical to extract features at any hidden layers of QNNs. Besides, there is a structural difference between QNNs and classical NNs. In classical NNs, the data is fed into the input layer followed by the hidden layers and the output layer sequentially, while the readout qubit is in parallel with the data qubits in QNNs.

In this work, we propose to use the measurement on the data qubits as the learned representations of the data. We define the learned feature vector of $|\boldsymbol{x}\rangle$ as $g_{\boldsymbol{\theta}}(\boldsymbol{x})$. Similar to Eq. 3, we have

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \langle 1, \boldsymbol{x}| \, U_{\boldsymbol{\theta}}^{\dagger} |\mathcal{M}_{1,\cdots,N}| U_{\boldsymbol{\theta}} \, |1, \boldsymbol{x}\rangle \tag{7}$$

where $\mathcal{M}_{1,\cdots,N}$ denotes the measurement on the data qubits instead of the whole system for simplicity. The Hilbert space of the input data $\boldsymbol{x}$ is $\mathbb{C}^{2^N}$. So we learn a mapping function $g_{\boldsymbol{\theta}} : \mathbb{C}^{2^N} \mapsto [-1, 1]^N$, which projects a quantum state to a real feature vector through transformation in a complex Hilbert space and quantum measurement. It is hard to study the Hilbert space directly. Given two different quantum states $|\boldsymbol{x}_j\rangle$ and $|\boldsymbol{x}_k\rangle$, we can define Euclidean distance between two feature vectors using Frobenius norm $||g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) - g_{\boldsymbol{\theta}}(\boldsymbol{x}_k)||$. With the tools defined Euclidean distance, we can analyze the QNNs as we do for classical NNs.

4 Negational Symmetry of Quantum Neural Networks

4.1 Negational Symmetry for Binary Classification

Let us first examine the quantum binary classification with an arbitrary example $|\boldsymbol{x}\rangle$. Let $|\boldsymbol{x}\rangle = |x_1\rangle \otimes |x_2\rangle \cdots \otimes |x_N\rangle$ be the data qubits of the binary pattern, where $x_i \in \{0, 1\}$ for $i \in \{1, 2, \cdots, N\}$. Then, the inverted binary pattern or the negational counterpart $|\tilde{\boldsymbol{x}}\rangle = |\tilde{x}_1\rangle \otimes |\tilde{x}_2\rangle \cdots \otimes |\tilde{x}_N\rangle$, where $|\tilde{x}_i\rangle = X\,|x_i\rangle$ for $i \in \{1, 2, \cdots, N\}$. Here, gate X is equivalent to bitwise `NOT` in classical computing. Let us denote $\boldsymbol{X} = \underbrace{X \otimes X \otimes \cdots \otimes X}_{N}$ for simplicity, then we have $|\tilde{\boldsymbol{x}}\rangle = \boldsymbol{X}\,|\boldsymbol{x}\rangle$. Formally, we introduce the following theorem.

Theorem 1. *Given a QNN with fixed parameters $\boldsymbol{\theta}$ and Z-measurement on the readout qubit, $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$.*

Here, the QNN is constructed in a way that defined in Section 3.2. Because all quantum gates in Eq. 3 are 2-D matrices, i.e. linear transformations, so the mathematical proof is straightforward. See Appendix A.6 for the proof. Note, Theorem 1 describes the situation in expectation. Although Eq. 3 is a closed-form expression, in a real quantum device, $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ for a single example is dependent on the average of the observed outcomes for a large number of repetitions (e.g. 1000 times), i.e. $\mathcal{M}_0$ is measured in multiple copies. That is to say, Theorem 1 may not hold for a single observation due to the stochasticity of quantum computing. As a comparison, most classical ML models have deterministic output in the inference phase.

It is worth mentioning that Theorem 1 holds not only when a QNN is well trained to converge, but also for a QNN with randomly initiated $\boldsymbol{\theta}$. In a 2-D Cartesian system, given a function f and a variable x , f is (reflectionally) symmetric if $f(x) = f(-x)$. Similarly, we define Theorem 1 as the *negational symmetry* for quantum binary classification as there is a symmetry in the measurement on the readout qubit.

To better understand the negational symmetry of QNNs, we decompose a QNN into blocks, as defined in Section 3.2. We choose the block as the basic unit because each data qubit is entangled with the readout qubit in a block. We study the relationship between the blocks (ZZ block or XX block) and the Pauli measurement ($\{X, Y, Z\}$). The results are summarized in Table 1. Note, as defined in Section 3.1, we have $-1 \leq f_{\boldsymbol{\theta}}(\boldsymbol{x}) \leq 1$. The negational symmetry is a built-in property of QNNs when there is at least one ZZ block in a QNN for binary pattern classification.

4.2 Negational Symmetry for Representation Learning

Analogous to classical representation learning, we need a tool to analyze the learned representations of QNNs. Here, we propose to use the expectation of the observed measurement outcomes as the learned representations and perform the same analysis with the mathematical tools that we use for classical NNs. As described in Section 3.4, we measure N data qubits with Pauli measurement

Architecture	Measurement	Symmetry
(XX)	Z	$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \quad f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}) = -1$
(XX)	X	$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \quad f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}) = 0$
(XX)	Y	$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = -f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}) = 0$
(ZZ)	Z	$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \quad f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$
(ZZ)	X	$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \quad f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}) = 0$
(ZZ)	Y	$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = -f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$
$(XX-ZZ)$	Z	$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \quad f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$
$(XX-ZZ)$	X	$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \quad f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}) = 0$
$(XX-ZZ)$	Y	$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = -f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$
$(ZZ-XX)$	Z	$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \quad f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$
$(ZZ-XX)$	X	$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \quad f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}) = 0$
$(ZZ-XX)$	Y	$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = -f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$

Table 1: A block-wise study of the negational symmetry for binary classification.

($\{X, Y, Z\}$). The output $g_{\boldsymbol{\theta}}(\boldsymbol{x})$ is a N-element feature vector in the real domain. With real feature vectors, we can use statistical tools to study the relationship between feature vectors, e.g. Pearson's correlation coefficient and cosine similarity. Besides, we can visualize the features for qualitative comparison. For example, we can visualize the learned representations with t-Distributed Stochastic Neighbor Embedding (t-SNE) [28] as a standard practice in representation learning. Note that, although we can have a way to visualize the final features of QNN, we cannot access the intermediate features in the tensor product Hilbert space directly, which is different from classical NNs.

In addition to the symmetry in the measurement on the readout qubit, there is also a symmetry in the learned representations. We present Theorem 2 for QNNs that are defined in Section 3.2. See Appendix A.7 for the proof. Compared with Theorem 1, Theorem 2 is more intuitive, where the feature vector of the binary pattern has an opposite direction against the feature vector of the negational binary pattern in the feature space. The mathematical relationship between $g_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $g_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$ is analogous to rotational symmetry (symmetric to the origin) in a Cartesian coordinate system. We define Theorem 2 as the *negational symmetry* of quantum representation learning. Again, we summarize the symmetry for all Pauli measurement in Table 2.

Theorem 2. *Given a QNN with fixed parameters $\boldsymbol{\theta}$ and Z-measurement on the data qubits, $g_{\boldsymbol{\theta}}(\boldsymbol{x}) = -g_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$.*

Measurement	Symmetry
Z	$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = -g_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$
X	$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \quad g_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}) = \mathbf{0}$
Y	$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = -g_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$

Table 2: Negational symmetry of representations for Pauli measurement

4.3 Role of Entanglement

In the history of classical NNs, it takes a long time to unveil the black-box of deep NNs. Based on current mathematical and experimental tools, some properties of QNNs can be directly proved. However, we can still peep into QNNs and try to understand the mechanism behind. First, the phenomenon is caused by quantum entanglement. Intuitively, assume that we use non-entangling gates to act on the readout qubit and each of the data qubit in Section 3.2, the physical interpretation of the training is nothing more than the random rotation of the readout qubit based on the label distribution. $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$ should not hold any more. In fact, because quantum gates are *affine* matrices, the evolution of an independent readout qubit ($|1\rangle$) should have one-to-to mapping given a VQC with fix parameters. Now, we can clearly see the Ising interactions between the readout qubit and data qubits change the behavior of the readout qubit. Second, the whole input system (the readout qubit and the data qubits) and a QNN can be viewed together as a multi-qubit entangled quantum state, or a generalized version of Bell state. For a QNN defined in Section 3.2, each data qubit is connected to the readout qubit with an entangling gate. That is to say, the readout qubit and data qubits (a binary pattern) are perfectly correlated.[7]

Here is a summary of the discussion above: when the readout qubit and the data qubits are entangled through a QNN classifier, the measurement on the readout qubit cannot be used to differentiate the binary pattern against its negational counterpart. In other words, QNNs are sensitive to the arrangement, instead of the states, of the data qubits.

4.4 Comparison with Classical Neural Networks

As a comparison, we use a standard DNN as an example. Assume there are N_{i-1} nodes for the $(i-1)^{th}$ layer and N_i nodes for the ith layer. Let $\boldsymbol{x}_{i-1} = [x_{i-1,1}, x_{i-1,2} \cdots, x_{i-1,N_{i-1}}]$ be the feature vector after the $(i-1)^{th}$ layer of a DNN. For simplicity, we ignore the nonlinear activation function. For the jth node of the ith layer, we have

$$x_{i,j} = \sum_{k=1}^{N_{i-1}} w_{i,j,k} x_{i-1,k} + b_{i,j} \tag{8}$$

where ws and b are the weights and the bias in a multiple linear regression (MLR). Although a DNN can model the relationship between the predicted label and the feature representation, there are strong statistical assumptions on the relationships among the independent variables and the dependent variable. Meanwhile, when a quantum circuit model achieves perfectly correlation, there is no requirement for independence assumption. In fact, the interaction between the entangled states in the tensor product Hilbert space cannot be easily explained by marginal probabilities from classical statistics. Unlike the interweaving connection between the nodes in the neighboring layer for a DNN, there is no direct connection between the quantum circuits operated on any of two data qubits. However, by connecting to the same readout qubit, a *global* interaction of all data qubits is created on the readout qubit.

5 Experiments

5.1 Experimental Setting

5.1.1 Environment

All experiments were run on a classical computer with Ubuntu 18.04 LTS. The CPU is an Intel® Xeon® Processor E5-2686 v4 @ 2.30 GHz with 45 MB cache. The GPU is a NVIDIA® Tesla® V100 with 16 GB memory. The RAM is up to 64 GB.

In this study, we simulate the NISQ circuits with `Cirq`.[8] The QNNs are implemented in `TensorFlow-Quantum`. For classical models, DNNs and CNNs are implemented in `TensorFlow`. We use the default implementation of support vector machines in `scikit-learn`.[9]

5.1.2 Datasets

The **MNIST** dataset [22] is a benchmark dataset for evaluating naive ML algorithms, which comprises images of ten handwritten digits from 0 to 9. The dataset has a training set of 60,000 examples, and a test set of 10,000 examples. The digits have been size-normalized and centered in a grayscale fixed-size image. Each image has a fixed resolution of 28×28.

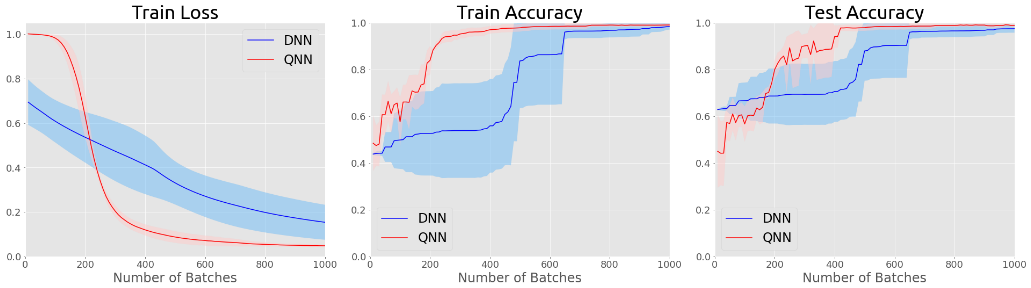
5.1.3 Hyperparameters

For a fair comparison, we use the same set of hyperparameters for the training of DNNs and CNNs. We use Adam [20], a gradient-based stochastic optimization method , with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$. The constant learning rate is 10^{-4}. The batch size is 32.

[7] See Appendix A.8 for discussion on quantum correlation.

[8] Cirq is main research tool used by Google AI Quantum team. `https://quantumai.google`

[9] `https://scikit-learn.org/stable/modules/svm.html`



(a) (b) (c)

Figure 2: Comparison between the DNN and the QNN on MNIST. The shaded region is 1 standard deviation over 5 runs with different random seeds.

5.2 Binary Pattern Classification

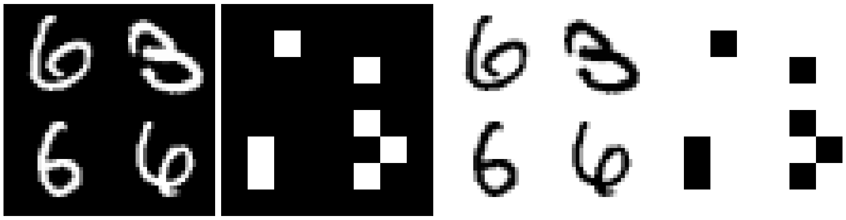
To evaluate negational symmetry of QNNs for binary classification, we compare QNNs with classical models in binary pattern classification tasks. Following Google's quantum computing experiments [12], we use the binarified MNIST dataset. We choose digits 3 and 6. At the beginning, the training set contains 6131 images labeled as 3 and 5918 images labeled as 6. Limited by the hardware, the images are downsampled from 28×28 to 4×4 to fit the simulator. We then map each grayscale pixel value to $\{0, 1\}$ with 128 as the threshold and remove the contradictory examples (the images are labeled as both 3 and 6). Here, $\{0, 1\}$ is equivalent to black-and-white image classification in computer vision. Then, in the quantum state preparation step, $\{0, 1\}$ is mapped to $\{|0\rangle, |1\rangle\}$ for each qubit. After the preprocessing, the training set consists of 3649 images while there are 2074 images labeled as 3 and 1575 images labeled as 6. With the same procedure, the final test set consists of 890 images while there are 332 images labeled as 3 and 558 images labeled as 6. The images are flatten into vectors. We use a 2-layer QNN described in Section 3.2. The readout qubit is measured by a Pauli Z operator. The QNN has a 16-qubit input register and 32 parameters in total. As a fair comparison, we use a 2-layer DNN whose number of parameters is close to the QNN's number of parameters. The first layer of DNN is a fully-connected layer with 16 input nodes and 2 output nodes, following by a ReLU activation function [31]. The second layer is a fully-connected (FC) layer with 2 input nodes and 1 output nodes. The total number of parameters for the DNN is 37. We use BCE as the loss function for DNNs. The training and test results are presented in Fig. 2. Both the DNN and the QNN achieve promising results on this supervised task. The QNN seems to converge faster and more stable than the DNN, when the training does not suffer from *barren plateaux* [29]. The smooth training curve of the QNN may be linked with the theoretical advantages discussed in Section 3.3.

We also repeated the above experiment in the negational setting. This time, we invert the grayscale MNIST images of the test set before the pre-processing step. From a classical view, we exchange the colors of the pixels of digit (white to black) and the pixels of background (black to white). See Fig. 3 for the intuition. Simply, the data qubits that were in the state $|0\rangle$ are now in the state $|1\rangle$, and vice versa. This bit-flipping operation is achieved via an X gate. In fact, this negational (or bit-flipping) operation creates a *domain shift* [38] if we consider the training set as the source domain and the test set as the target domain[10] from the perspective of classical ML. That is to say, we have a transfer learning problem as we want to extract knowledge learned from the training set but pply it to the negational test set. Surprisingly, QNN continues to maintain a high performance on the test set with negational operation, while DNN suffers from severe overfitting. The training and test results are presented in Fig. 4. In this work, the same experiments were repeated for different two-digit combinations and the similar phenomena were observed.

5.3 Unity of Opposites

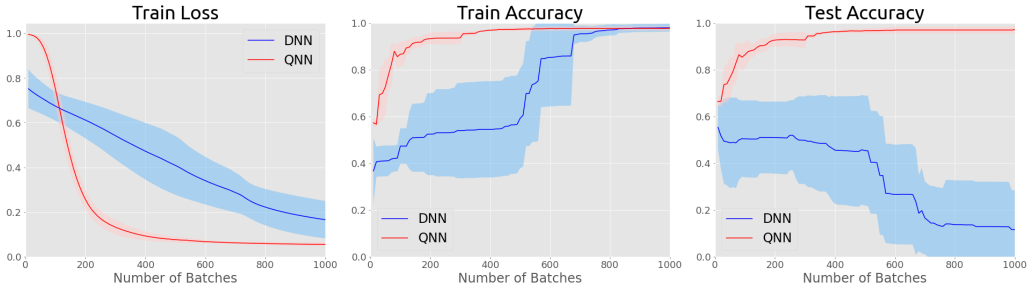
To examine the generality of the negational symmetry for quantum binary classification, we evaluate QNNs with different architectures under the same experimental setting. We first extend the 2-layer QNN ($XX - ZZ$) to deeper QNNs, namely a 3-layer QNN ($XX - ZZ - XX$) and a 4-layer QNN

[10] See Appendix B for definition of unsupervised domain adaptation.



(a) (b) (c) (d)

Figure 3: Visualization: (a) the original test images; (b) the binarized test images; (c) the inverted test images; (d) the binarized inverted test images.



(a) (b) (c)

Figure 4: Comparison between the DNN and the QNN on MNIST with inverted test set. The shaded region is 1 standard deviation over 5 runs with different random seeds.

$(XX - ZZ - XX - ZZ)$, to study the effect of the depth on model performance. We then study the order of the blocks of X-parity gates and Z-parity gates with a 2-layer QNN $(XX - ZZ)$ and 3-layer QNN $(XX - ZZ - XX)$. Unlike Section 5.2, we use early stopping to report best test accuracy. We train the models on the training set with the **same** random seeds, freeze the weights and evaluate on the test set without negational operation and the test set with negational operation. The results are present in Table 3. We observe that QNNs have the same outstanding performance when the test set is bit-flipped. Generally, deeper QNNs, with more parameters, have slightly better performance than shallower QNNs. Under Z-measurement, QNNs with Z-parity gates in the last block show better performance than QNNs with X-parity gates in the last block.

If we take a closer look at Table 3, QNNs actually have the identical binary classification accuracy on two different test sets, which validates the negational symmetry empirically. Compared with classical models, QNNs are invariant to the effect caused by the negational operation on the test set and are able to recognize the pattern of the digit even when the test set is bit-flipped. Between two test sets, there is an one-to-one mapping between a binary pattern and its negational counterpart. We compute the difference between two logits of QNNs, i.e. Eq. 3, for all 890 pairs in two test sets. We find the numerical difference is negligible. For example, the mean of the differences is $-4.4240963 \times 10^{-8}$ and the standard deviation is $1.05649356 \times 10^{-7}$ for QNN $(XX - ZZ)$. If we take the noisy environment of NISQ device and significant figures into account, we can say QNNs make the same prediction on a binary pattern and its negational counterpart. Here, we name this phenomenon as *unity of opposites* in binary pattern classification. Unity of opposites is originally a philosophical term that describes that two opposite concepts can also share the same concept in a way.[11] A binary pattern and its negational counterpart can be viewed as two opposites and two opposites share the same semantic information (digit). Intuitively, QNNs can find the unity of two opposites in binary pattern classification. QNNs reals the unity of mathematics and philosophy via a mathematical symmetry.

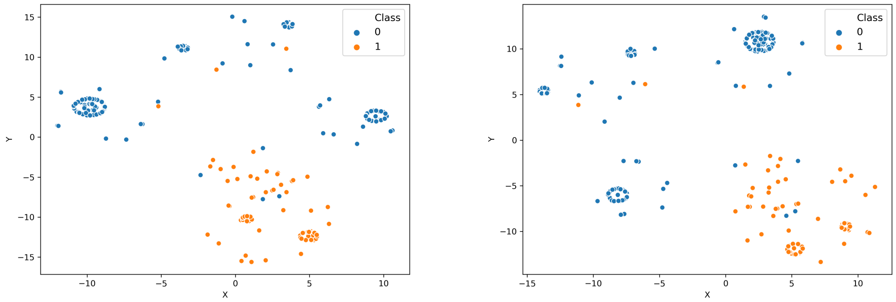
To validate Theorem 2, we calculate the norm of the sum of two pairwise feature vectors $||g_{\boldsymbol{\theta}}(\boldsymbol{x}) + g_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})||$. The sum of the norms is less than 1×10^{-7}, which further validates the negational symmetry. Following the discussion in Section 4.2, we also check the pairwise statistical similarity for the

[11] See Appendix C for definition of unity of opposites.

Architecture	w/o Negation	w/ Negation	# Params
$(XX - ZZ)$	0.9783	0.9783	32
$(XX - ZZ - XX)$	0.9674	0.9674	48
$(XX - ZZ - XX - ZZ)$	0.9922	0.9922	64
$(ZZ - XX)$	0.9707	0.9707	32
$(ZZ - XX - ZZ)$	0.9967	0.9967	48

Table 3: Comparison of model performance for QNNs with different architectures.

feature vectors in the original test set and its negational counterpart. For Z measurement, the mean for pairwise Pearson's correlation coefficient is -0.5 and the mean for pairwise cosine similarity is -1The t-SNE visualizations with the **same** random seed are present in Fig. 5.



(a) Z on the test set w/o negational operation. (b) Z on the test set w/ negational operation.

Figure 5: t-SNE visualization of learned representations.

5.4 Comparison with Classical Models

We compare QNNs with 3 main categories of ML classification models and we find unity of opposites in binary pattern classification is unique to QNNs. The first category is DNN. The 2-layer DNN $(16 - 2 - 1)$ in Section 5.2 is treated as the baseline. We investigate 3 variants of the baseline: (i) increasing the number of nodes in the hidden layer $(16 - 16 - 1)$; (ii) increasing the number of hidden layers $(16 - 16 - 2 - 1)$; and (iii) increasing both the number of nodes in the hidden layer and the number of hidden layers $(16 - 16 - 16 - 1)$. The reported performance (binary classification accuracy) are present in Table 4. Compared with QNNs, DNNs suffer from severe overfitting when the test set is bit-flipped. We notice that DNNs with large number of nodes in the last hidder layer show slightly better performance than DNNs with a small number of nodes in the last hidden layer.

Model	Architecture	w/o Negation	w/ Negation	# Params
QNN	$XX - ZZ$	0.9783	0.9783	32
DNN	$16 - 2 - 1$	0.9775	0.1146	37
DNN	$16 - 16 - 1$	0.9944	0.1528	289
DNN	$16 - 16 - 2 - 1$	0.9978	0.0067	309
DNN	$16 - 16 - 16 - 1$	0.9933	0.0090	561

Table 4: Comparison of model performance for QNN and DNNs with different architectures.

The second category is CNN, a special form of DNN. CNNs have lead to significant breakthroughs in visual recognition tasks [39, 14], which is a strong baseline. Considering the image resolution is only 4×4, we only use CNNs with simple architecture. The convolution operation is padded to have the same input and output feature map size. Each convolutional layer is followed by a ReLU activation function. We use max pooling to downscale the image size and use global average pooling to extract features from each feature channel. For a fair comparison and to avoid overfitting, we set the number of feature channels to be 16. We consider a 3-layer CNN and a 5-layer CNN. See Table 5 for the detailed architecture of two CNNs. Even though, we restrict the number of feature channels, CNNs still have a huge number of parameters compared with the considered DNNs. Accordingly, CNNs seem to learn more useful representations than DNNs with higher accuracy in test set in both situations.

Model	Architecture	w/o Negation	w/ Negation	# Params
QNN	$XX - ZZ$	0.9783	0.9783	32
CNN	CV-MP-CV-AP-FC	0.9966	0.2888	2497
CNN	CV-CV-MP-CV-CV-AP-FC	0.9978	0.3618	7137

Table 5: Comparison of model performance for QNN and CNNs with different architectures. CV stands for convolutional layer, MP stands for max pooling layer, AP stands for average pooling layer, and FC stands for fully-connected layer.

The third category is support vector machines (SVM). SVM is one of the most robust prediction methods, which played an important role in ML before the era of deep learning. In this work, we compare a QNN ($XX - ZZ$) with two representative kernel functions of SVMs, which are the linear kernel and the radial basis function (RBF) kernel. For a RBF kernel $\exp\{-\gamma||\boldsymbol{x}_i - \boldsymbol{x}_j||^2\}$, we set $\gamma = 1/(16 * \mathrm{Var}(\boldsymbol{X}))$, where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are training binary data vectors, and $\boldsymbol{X}$ is the training binary data matrix. The reported performance are present in Table 6. SVMs are quite sensitive to the regularization hyperparameter C when the test set is bit-flipped.

Model	Kernel	C	w/o Negation	w/ Negation
QNN	-	-	0.9783	0.9783
SVM	Linear	0.001	0.9787	0.0169
SVM	Linear	0.01	0.9944	0.0056
SVM	Linear	0.1	0.9753	0.0258
SVM	Linear	1	0.9753	0.3708
SVM	Linear	10	0.9978	0.1506
SVM	Linear	100	0.9978	0.0112
SVM	RBF	0.001	0.5584	0.3730
SVM	RBF	0.01	0.9933	0.3730
SVM	RBF	0.1	0.9978	0.0449
SVM	RBF	1	0.9753	0.3730
SVM	RBF	10	0.9753	0.3730
SVM	RBF	100	0.9753	0.3730

Table 6: Comparison of model performance for QNN and SVMs with different kernel functions and regularization hyperparameter. C is the regularization parameter.

QNNs outperform classical models by a large margin on the bit-flipped test set, as shown in Table 4, Table 5, and Table 6. Even though, with more hidden nodes for DNNs or more feature channels for CNNs, classical NNs could perform better, the effect caused by the additional negational operation can not be mitigated.

6 Limitations

In this era of simplistic simulation, it is still challenging to mathematically derive the process of the entanglement on a multi-qubit state with a generalized VQC as in Section 4. The results we present in this work are limited by the available mathematical and experimental tools. In the long term, we expect more advanced theoretical analysis and experiments to emerge from new joint developments in mathematics, physics, and engineering.

In the meanwhile, there are two limitations in the experimental design. Firstly, we can only afford a GPU-based simulated environment with 16 qubits. Secondly, limited by the number of qubits, we can only have 4×4 binary images, which also limits the experimental design for the binary patterns. Overall, we expect all these issues can be well-addressed by further development in quantum hardware.

7 Conclusions

In this work, we propose and discuss the negational symmetry of QNNs in binary pattern classification, a mathematical property of QNNs that is unseen before. We formalize and prove this property and discuss the mechanism behind. We evaluate the negational symmetry in MNIST experiments and demonstrate the practical values of it by comparing with classical models. We expect that the negational symmetry in quantum binary classification and quantum representation learning could benefit the future research on QML.

Acknowledgments

Aleks Kissinger is the lecturer for *Quantum Information*[12] at the Department of Computer Science, University of Oxford. We would like to thank Bob Coecke from the Department of Computer Science, University of Oxford for sharing knowledge on ZX-calculus. We would also like to thank Amazon for providing GPU computing service for this study.

A Quantum Basics

A.1 Qubit

In quantum computing, the basic unit of information is a quantum bit or qubit. A qubit can be realized by different physical systems with two perfectly distinguishable states, e.g. the vertical polarization and horizontal polarization of a single photon. Assume each qubit is in one of two perfectly distinguishable states, we can represent the binary pattern by qubits.

A.2 Tensor Product Hilbert Space

For system A with Hilbert space $\mathbb{H}_A = \mathbb{C}^{d_A}$ with dimension d_A and system B with Hilbert space $\mathbb{H}_B == \mathbb{C}^{d_B}$ with dimension d_B, the Hilbert space of the composite system AB is the tensor product of the Hilbert spaces of A and B. In formula, $\mathbb{H}_{AB} = \mathbb{H}_A \otimes \mathbb{H}_B = \mathbb{C}^{d_A d_B}$.

A.3 Pauli Matrices

$$\sigma_0 = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \sigma_x = X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \sigma_y = Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \sigma_z = Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{A.1}$$

$$\sigma_0^2 = \sigma_x^2 = \sigma_y^2 = \sigma_z^2 = I \tag{A.2}$$

$$X\left|0\right\rangle = \left|1\right\rangle, X\left|1\right\rangle = \left|0\right\rangle, Y\left|0\right\rangle = i\left|1\right\rangle, Y\left|1\right\rangle = i\left|0\right\rangle, Z\left|0\right\rangle = \left|0\right\rangle, Z\left|1\right\rangle = -\left|1\right\rangle \tag{A.3}$$

A.4 Pauli Rotation Operators

The rotation operators are generated by exponentiation of the Pauli matrices according to $e^{(iA\theta)} = \cos(\theta)I + i\sin(\theta)\mathcal{M}$, where $\mathcal{M} \in \{X, Y, Z\}$. The rotation gate $R_a(\theta)$ is a single-qubit rotation through angle θ (radians) around the corresponding axis $a \in \{x, y, z\}$.

$$R_x(\theta) = e^{-i\frac{\theta X}{2}} = \cos\left(\frac{\theta}{2}\right)I - i\sin\left(\frac{\theta}{2}\right)X = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{A.4}$$

$$R_y(\theta) = e^{-i\frac{\theta \mathrm{Y}}{2}} = \cos\left(\frac{\theta}{2}\right)I - i\sin\left(\frac{\theta}{2}\right)Y = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{A.5}$$

$$R_z(\theta) = e^{-i\frac{\theta Z}{2}} = \cos\left(\frac{\theta}{2}\right)I - i\sin\left(\frac{\theta}{2}\right)Z = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \tag{A.6}$$

[12] `https://www.cs.ox.ac.uk/teaching/courses/2019-2020/qi/index.html`

A.5 Universality of ZX-calculus

According to Eq. (A.4), Eq. (A.5), and Eq. (A.6), we notice that $R_y(\theta) = Z^{\frac{1}{2}} R_x(\theta) Z^{\frac{1}{2}\dagger}$, which we can use R_x and R_z to represent R_y with arbitrary angles. Formally, we have the following theorem.

Theorem A.1 ([8])**.** *For any unitary U on a single qubit there exist phases α, β, and γ such that U can be written as: $U = R_x(\gamma)R_z(\beta)R_x(\alpha)$. This is called the Euler decomposition of U and the phases α, β, and γ are called the Euler angles.*

Theorem A.2 ([8])**.** *Any n-qubit unitary can be constructed out of the CNOT gate and phase gates.*

Theorem A.3. *For any nonlinear function, there exists at least one VQC $U_{\boldsymbol{\theta}}$ with following properties: (1) it can be constructed out of the gate set $\mathbb{U} = \{R_x, R_z, \mathrm{CNOT}\}$ with parameters $\boldsymbol{\theta}$; (2) it can ϵ-approximate the nonlinear function for $\epsilon > 0$.*

Proof. R_x and R_z are also called *phase gates* in ZX-calculus. Here, we use the fact that single-qubit elementary gates and two-qubit gates such as XX, ZZ, CZ are special cases of the gates in $\mathbb{U}$ or can be constructed out of $\mathbb{U}$. Theorem A.3 is a direct result of the Universality Theorem of neural networks and the Universality Theorem of ZX-calculus.

A.6 Sketch of Proof for Theorem 1

The mathematical proof for Theorem 1 is straightforward. For simplicity, we assume that there is only one readout qubit $|1\rangle$ and one data qubit $|0\rangle$ (the opposite is then $|1\rangle$. Here, we demonstrate the proof for QNN ($XX - ZZ$) with Z-measurement. The proof for multiple data qubits and QNNs with different architectures share the same logic. We use the notations in Section 3.1 and 4.1.

Following Eq. 3, we have

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \langle 1,0|\, U_{\boldsymbol{\theta}}^{\dagger} |Z \otimes I| U_{\boldsymbol{\theta}}\, |1,0\rangle \,, \tag{A.7}$$

where

$$U_{\boldsymbol{\theta}} = (H \otimes I)(R_x(\theta_1) \otimes R_x(\theta_1))(R_z(\theta_2) \otimes R_z(\theta_2))(H \otimes I). \tag{A.8}$$

We have

$$H \otimes I = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix},$$

$$R_x(\theta_1) \otimes R_x(\theta_1) = \begin{bmatrix} \cos(\frac{\theta_1}{2}) & 0 & 0 & -i\sin(\frac{\theta_1}{2}) \\ 0 & \cos(\frac{\theta_1}{2}) & -i\sin(\frac{\theta_1}{2}) & 0 \\ 0 & -i\sin(\frac{\theta_1}{2}) & \cos(\frac{\theta_1}{2}) & 0 \\ -i\sin(\frac{\theta_1}{2}) & 0 & 0 & \cos(\frac{\theta_1}{2}) \end{bmatrix},$$

and

$$R_z(\theta_2) \otimes R_z(\theta_2) = \begin{bmatrix} e^{-i\theta_2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\theta_2} \end{bmatrix}.$$

Substitute $H \otimes I$, $R_x(\theta_1) \otimes R_x(\theta_1)$ and $R_z(\theta_2) \otimes R_z(\theta_2)$ into Eq. A.8, we have

$$U_{\boldsymbol{\theta}} = \frac{1}{2} \begin{bmatrix} \cos(\frac{\theta_1}{2})(e^{-i\theta_2}+1) & \cos(\frac{\theta_1}{2})(e^{i\theta_2}-1) & -i\sin(\frac{\theta_1}{2})(e^{-i\theta_2}+1) & i\sin(\frac{\theta_1}{2})(e^{-i\theta_2}-1) \\ \cos(\frac{\theta_1}{2})(e^{-i\theta_2}-1) & \cos(\frac{\theta_1}{2})(e^{-i\theta_2}+1) & -i\sin(\frac{\theta_1}{2})(e^{-i\theta_2}-1) & i\sin(\frac{\theta_1}{2})(e^{-i\theta_2}+1) \\ -i\sin(\frac{\theta_1}{2})(1+e^{-i\theta_2}) & i\sin(\frac{\theta_1}{2})(1-e^{-i\theta_2}) & \cos(\frac{\theta_1}{2})(1+e^{i\theta_2}) & \cos(\frac{\theta_1}{2})(1-e^{i\theta_2}) \\ -i\sin(\frac{\theta_1}{2})(1-e^{-i\theta_2}) & i\sin(\frac{\theta_1}{2})(1+e^{-i\theta_2}) & \cos(\frac{\theta_1}{2})(1-e^{i\theta_2}) & \cos(\frac{\theta_1}{2})(1+e^{i\theta_2}) \end{bmatrix}. \tag{A.9}$$

We have

$$U_{\boldsymbol{\theta}}\,|1,0\rangle = \begin{bmatrix} -i\sin(\frac{\theta_1}{2})(e^{-i\theta_2}+1) \\ -i\sin(\frac{\theta_1}{2})(e^{-i\theta_2}-1) \\ \cos(\frac{\theta_1}{2})(1+e^{i\theta_2}) \\ \cos(\frac{\theta_1}{2})(1-e^{i\theta_2}) \end{bmatrix} = \begin{bmatrix} -\sin(\frac{\theta_1}{2})\sin(\theta_2) - i\sin(\frac{\theta_1}{2})(\cos(\theta_2)+1) \\ -\sin(\frac{\theta_1}{2})\sin(\theta_2) - i\sin(\frac{\theta_1}{2})(\cos(\theta_2)-1) \\ \cos(\frac{\theta_1}{2})(1+\cos(\theta_2)) + i\cos(\frac{\theta_1}{2})\sin(\theta_2) \\ \cos(\frac{\theta_1}{2})(1-\cos(\theta_2)) - i\cos(\frac{\theta_1}{2})\sin(\theta_2) \end{bmatrix} \tag{A.10}$$

, where

$$|1,0\rangle = [0 \quad 0 \quad 1 \quad 0]^T \,.$$

$$Z \otimes I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Substitute $Z \otimes I$ and $U_{\boldsymbol{\theta}} |1, 0\rangle$ into Eq. A.7, we have

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sin^2(\frac{\theta_1}{2}) - \cos^2(\frac{\theta_1}{2}) \tag{A.11}$$

.

Similarly, we have

$$U_{\boldsymbol{\theta}} |1, 1\rangle = \begin{bmatrix} i\sin(\frac{\theta_1}{2})(e^{-i\theta_2} - 1) \\ i\sin(\frac{\theta_1}{2})(e^{-i\theta_2} + 1) \\ \cos(\frac{\theta_1}{2})(1 - e^{i\theta_2}) \\ \cos(\frac{\theta_1}{2})(1 + e^{i\theta_2}) \end{bmatrix} = \begin{bmatrix} \sin(\frac{\theta_1}{2})\sin(\theta_2) + i\sin(\frac{\theta_1}{2})(\cos(\theta_2) - 1) \\ \sin(\frac{\theta_1}{2})\sin(\theta_2) + i\sin(\frac{\theta_1}{2})(\cos(\theta_2) + 1) \\ \cos(\frac{\theta_1}{2})(1 - \cos(\theta_2)) - i\cos(\frac{\theta_1}{2})\sin(\theta_2) \\ \cos(\frac{\theta_1}{2})(1 + \cos(\theta_2)) + i\cos(\frac{\theta_1}{2})\sin(\theta_2) \end{bmatrix} \tag{A.12}$$

, where

$$|1, 1\rangle = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T,$$

and

$$f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}) = \langle 1, 1| U_{\boldsymbol{\theta}}^\dagger | Z \otimes I | U_{\boldsymbol{\theta}} |1, 1\rangle = \sin^2(\frac{\theta_1}{2}) - \cos^2(\frac{\theta_1}{2}). \tag{A.13}$$

So $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$. □

A.7 Sketch of Proof for Theorem 2

The proof is similar to Appendix A.6. Again, we prove the fundamental case where there is one readout qubit and one data qubit for QNN ($XX - ZZ$) with Z-measurement.

Following Eq. 7, we have

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \langle 1, 0| U_{\boldsymbol{\theta}}^\dagger | Z \otimes Z | U_{\boldsymbol{\theta}} |1, 0\rangle, \tag{A.14}$$

where

$$Z \otimes Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

Substitute $Z \otimes Z$ and $U_{\boldsymbol{\theta}} |1, 0\rangle$ into Eq. A.14, we have

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = (\sin^2(\frac{\theta_1}{2}) - \cos^2(\frac{\theta_1}{2}))\cos(\theta_2) \tag{A.15}$$

. Similarly, we have

$$g_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}) = \langle 1, 1| U_{\boldsymbol{\theta}}^\dagger | Z \otimes Z | U_{\boldsymbol{\theta}} |1, 1\rangle = -(\sin^2(\frac{\theta_1}{2}) - \cos^2(\frac{\theta_1}{2}))\cos(\theta_2). \tag{A.16}$$

So $g_{\boldsymbol{\theta}}(\boldsymbol{x}) = -g_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$. □

A.8 Quantum Correlation

Assume there is a composite system AB, where A and B are two qubits. We have $|\Phi^+\rangle = \text{CNOT}(H \otimes I)(|0\rangle \otimes |0\rangle) = \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$, i.e. we create a Bell state through an entangling gate. If we measure A and B both in the same basis, we can verify that for the composite system AB, the outcomes of A and B are perfectly correlated. After the entangling gate, A and B become perfectly correlated. This phenomenon is called *quantum steering* or quantum correlation.

A.8.1 Computational Basis

A and B are in a Bell state. If we measure A and B both in the computational basis $\{|0\rangle, |1\rangle\}$, the probability that both A and B get outcome 0 is

$$\begin{aligned} P_{A,B}(0,0) &= |(\langle 0| \otimes \langle 0|) \ |\Phi^+\rangle|^2 \\ &= |(\langle 0| \otimes \langle 0|) \ \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)|^2 \\ &= |\frac{1}{\sqrt{2}}|^2 \ |(\langle 0| \otimes \langle 0|) \ (|0\rangle \otimes |0\rangle) + (\langle 0| \otimes \langle 0|) \ (|1\rangle \otimes |1\rangle)|^2 \\ &= \frac{1}{2} \ |(\langle 0\rangle \, 0 \ \ \langle 0\rangle \, 0) + (\langle 0\rangle \, 1 \ \ \langle 0\rangle \, 1)|^2 \\ &= \frac{1}{2} \end{aligned} \tag{A.17}$$

In the same way, we have

$$P_{A,B}(0,1) = |(\langle 0| \otimes \langle 1|) \ |\Phi^+\rangle|^2 = 0 \tag{A.18}$$

$$P_{A,B}(1,0) = |(\langle 1| \otimes \langle 0|) \ |\Phi^+\rangle|^2 = 0 \tag{A.19}$$

$$P_{A,B}(1,1) = |(\langle 1| \otimes \langle 1|) \ |\Phi^+\rangle|^2 = \frac{1}{2} \tag{A.20}$$

A.8.2 Fourier Basis

Similarly, if we measure A and B both in the Fourier basis $\{|+\rangle, |-\rangle\}$, where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, we have

$$P_{A,B}(+,+) = |(\langle +| \otimes \langle +|) \ |\Phi^+\rangle|^2 = \frac{1}{2} \tag{A.21}$$

$$P_{A,B}(+,-) = |(\langle +| \otimes \langle -|) \ |\Phi^+\rangle|^2 = 0 \tag{A.22}$$

$$P_{A,B}(-,+) = |(\langle -| \otimes \langle +|) \ |\Phi^+\rangle|^2 = 0 \tag{A.23}$$

$$P_{A,B}(-,-) = |(\langle -| \otimes \langle -|) \ |\Phi^+\rangle|^2 = \frac{1}{2} \tag{A.24}$$

B Unsupervised Domain Adaptation

Let $\mathcal{D}_S = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{n}$ denote the source domain and $\mathcal{D}_T = \{\boldsymbol{x}_j\}_{j=1}^{n'}$ denote the target domain. Assume that the hypothesis class $\mathcal{H}$ with VC dimension d is a set of binary classifiers [13] $h : \mathcal{X} \mapsto \{-1, 1\}$, where $\mathcal{X}$ is the input space. The goal of unsupervised domain adaptation (UDA) is to find a h with a low target risk. [3] proposed the following generalization bound on the target risk.[14]

Theorem B.1 (Generalization Bound [3]). *With probability 1- δ over the choice of m samples $\tilde{\mathcal{D}_S}$ from $\mathcal{D}_S$ and m' samples $\tilde{\mathcal{D}_T}$ from $\mathcal{D}_T$, $\forall\ h \in \mathcal{H}$:*

$$\begin{aligned} \mathcal{R}_{\mathcal{D}_T}(h) \le & \mathcal{R}_{\tilde{\mathcal{D}_S}}(h) + \frac{4}{m}\sqrt{(d\log\frac{2em}{d} + \log\frac{4}{\delta})} + \\ & 4\sqrt{\frac{1}{m'}(d\log\frac{2m'}{d} + \log\frac{4}{\delta})} + \hat{d}_{\mathcal{H}}(\tilde{\mathcal{D}_S}, \tilde{\mathcal{D}_T}) + \lambda, \end{aligned}$$

where $\hat{d}_{\mathcal{H}}(\tilde{\mathcal{D}_S}, \tilde{\mathcal{D}_T})$ is an empirical estimate of $\mathcal{H}$-divergence [15] between $\mathcal{D}_S$ and $\mathcal{D}_T$, and $\lambda \ge \inf_{h^* \in \mathcal{H}}[\mathcal{R}_{\mathcal{D}_S}(h^*) + \mathcal{R}_{\mathcal{D}_T}(h^*)]$.

[13] For a classical binary classification task, another choice of label set is $\{0, 1\}$, which is convenient for calculating the binary cross-entropy. $\{-1, 1\}$ is more popular in quantum physics community. We use two definitions interchangeably in this paper.

[14] The generalization bound derived by [3] is based on a discrete loss function $\mathcal{R}_{\tilde{\mathcal{D}_S}} = \frac{1}{m}\sum_{j=1}^{m}\mathbf{1}[h(\boldsymbol{x}_j) \neq y_j]$. [42] later extended Theorem B.1 to continuous loss functions and obtained a similar result. Here, we only present the simple one for simplicity.

[15] $\hat{d}_{\mathcal{H}}(\tilde{\mathcal{D}_S}, \tilde{\mathcal{D}_T}) = 2(1 - \min_{h\in\mathcal{H}}[\frac{1}{m}\sum_{j=1}^{m}\mathbf{1}[h(\boldsymbol{x}_j) = 0] + \frac{1}{m'}\sum_{j=1}^{m'}\mathbf{1}[h(\boldsymbol{x}_j) = 1]$ [19, 3, 2]

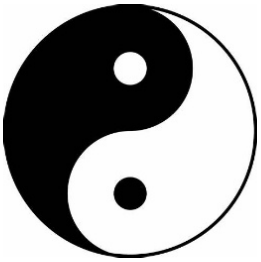


Figure 6: The symbol of yin and yang, a concept of dualism in ancient Chinese philosophy.

C Unity of Opposites

The unity of opposites, closely related to the concept of non-duality, is the main category of dialectics, which describes the co-existence of two opposite concepts. The original thinking on dualism can be dated back to ancient China and Greek. In dualism, there are two fundamental but opposite concepts, e.g. yin and yang, the earliest binary system in the world (see Fig 6). In addition to philosophy, dualism exists in many fields of science, such as 0 and 1, true and false. Given a digital signal, it has to be either 0 or 1. Given a mathematical statement, it has either be true or false. From a scientific view of dualism, two opposites must be in two distinct states in a binary system. However, some philosophers hold an opposite point of view to this. In the unity of opposites, the term *coincidentia oppositorum*[16], or coincidence of opposites in English, is used to describe the situation that two opposites are identical in philosophy and theology, and was described as "the mythical pattern" in the 1950s.[17]

References

[1] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[3] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 137–144, 2007.

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[5] Yoshua Bengio, Paolo Frasconi, and Patrice Simard. The problem of learning long-term dependencies in recurrent networks. In *IEEE International Conference on Neural Networks*, pages 1183–1188. IEEE, 1993.

[6] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[7] Yudong Cao, Gian Giacomo Guerreschi, and Alán Aspuru-Guzik. Quantum neuron: an elementary building block for machine learning on quantum computers. *arXiv preprint arXiv:1711.11240*, 2017.

[8] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes*. Cambridge University Press, 2017.

[9] Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019.

[10] Gavin E. Crooks. Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition, 2019.

[11] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[16] *Coincidentia oppositorum* was first proposed by Nicholas of Cusa, a German polymath in 1440.

[17] The term "the mythical pattern" was proposed by Mircea Eliade, a historian of religion, philosopher, and professor at the University of Chicago.

[12] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[16] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*, pages 1989–1998, 2018.

[17] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[18] Iordanis Kerenidis, Jonas Landman, and Anupam Prakash. Quantum algorithms for deep convolutional neural networks. In *International Conference on Learning Representations*, 2020.

[19] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *International Conference on Very Large Data Bases*, pages 180–191, 2004.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[22] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[23] Jin-Guo Liu and Lei Wang. Differentiable learning of quantum circuit born machines. *Physical Review A*, 98(6):062324, 2018.

[24] Seth Lloyd, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran. Quantum embeddings for machine learning. *arXiv preprint arXiv:2001.03622*, 2020.

[25] Seth Lloyd and Christian Weedbrook. Quantum generative adversarial learning. *Physical Review Letters*, 121(4):040502, 2018.

[26] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105, 2015.

[27] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*, pages 2208–2217, 2017.

[28] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[29] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):1–6, 2018.

[30] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.

[31] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, page 807–814, 2010.

[32] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[33] Jonathan Romero and Alan Aspuru-Guzik. Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions. *arXiv preprint arXiv:1901.00848*, 2019.

[34] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[35] Erwin Schrödinger. Discussion of probability relations between separated systems. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 31, pages 555–563. Cambridge University Press, 1935.

[36] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3), Mar 2019.

[37] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.

[38] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[40] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[41] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[42] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413, 2019.

[43] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Information*, 5(1):1–9, 2019.