9-1-2022

Negational symmetry of quantum neural networks for binary pattern classification

Nanqing Dong
University of Oxford, United Kingdom

Michael Kampffmeyer
UiT The Arctic University of Norway, Tromsø, Norway

Irina Voiculescu
University of Oxford, United Kingdom

Eric P. Xing
Carnegie Mellon University, United States & Mohamed bin Zayed University of Artificial Intelligence

Recommended Citation

Negational symmetry of quantum neural networks for binary pattern classification

Nanqing Dong [a,*], Michael Kampffmeyer [b], Irina Voiculescu [a], Eric Xing [c,d]

[a] *Department of Computer Science, University of Oxford, Oxford OX1 3QD, UK*
[b] *Department of Physics and Technology, UiT The Arctic University of Norway, Tromsø 9019, Norway*
[c] *Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[d] *Mohamed bin Zayed University of Artificial Intelligence, Masdar, Abu Dhabi, United Arab Emirates*

ABSTRACT

Although quantum neural networks (QNNs) have shown promising results in solving simple machine learning tasks recently, the behavior of QNNs in binary pattern classification is still underexplored. In this work, we find that QNNs have an Achilles' heel in binary pattern classification. To illustrate this point, we provide a theoretical insight into the properties of QNNs by presenting and analyzing a new form of *symmetry* embedded in a family of QNNs with *full entanglement*, which we term *negational symmetry*. Due to negational symmetry, QNNs can not differentiate between a quantum binary signal and its negational counterpart. We empirically evaluate the negational symmetry of QNNs in binary pattern classification tasks using Google's quantum computing framework. Both theoretical and experimental results suggest that negational symmetry is a fundamental property of QNNs, which is not shared by classical models. Our findings also imply that negational symmetry is a double-edged sword in practical quantum applications.

1. Introduction

In contrast to *quantum-inspired machine learning*, which incorporates the concepts of quantum mechanics with classical machine learning (ML) [1–3], quantum machine learning (QML) [4,5] aims to understand how to devise and implement ML algorithms on quantum computers. QML has received increasing attention due to the quantum supremacy experiment [6] on near-term noisy intermediate-scale quantum (NISQ) [7] devices. However, the study of QML is still in an early stage. While various classical ML methods, such deep neural networks (DNNs) and convolutional neural networks (CNNs) [8], can easily show robust performance in binary pattern classification [9–11], a fundamental task that is used to evaluate the efficiency of ML algorithms [12], quantum neural networks (QNNs) [13–15] still have veiled mysteries. This paper aims to provide an exploratory understanding of the application of QNNs in binary pattern classification.

As a quantum circuit model consisting of *unitary* quantum gates, QNNs have direct links with *feed-forward neural networks*

and *invertible neural networks* [16] in classical ML. At present, the study of QNNs is still in an early phase as QNNs have not shown quantum supremacy in solving classical ML tasks. However, QNNs can be easily integrated with quantum devices and quantum data, which gives them unparalleled advantages in quantum applications without access to complex classical computing systems. This is often ignored when comparing QNNs with classical models. Moreover, although it appears that we can easily manipulate the conversion between classical and quantum data, the situation in a real quantum device is much more complicated than the simulated study. For example, the *No-Cloning Theorem* [17] states that quantum data cannot be copied, which means that, for certain QML tasks, the quantum data cannot be converted for classical models. Thus, researchers are eager to peep into the blackbox of QNNs and to analyze the properties of these models.

In this work, we find that QNNs might not be able to differentiate between a binary pattern and its *negational* counterpart, where the term *negational* refers to the negation in bitwise operation or logical operation and the negational transformation of a binary pattern is equivalent to applying a NOT gate (negation operation) to all bits, i.e. flip all bits. To illustrate this Achilles' heel of QNNs in binary pattern classification, we provide insights into a

---

* Corresponding author.
  E-mail address: nanqing.dong@cs.ox.ac.uk (N. Dong).

new form of symmetry inherent in a specific family of QNNs with *full entanglement*[1], i.e. each data qubit is entangled with the readout qubit. We find that QNNs with full entanglement[2] feature two kinds of symmetries for binary classification and the auxiliary representation learning tasks with *quantum binary signals* as the input. A quantum binary signal (e.g. 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, we present two major findings. First, given a (quantum) binary pattern, a QNN with Z-measurement makes the same prediction for a binary pattern and its negational counterpart. Second, with the proposed quantum representation learning, we find that the learned feature vectors of a binary pattern and its negational counterpart are two opposite vectors. Essentially, QNNs are mathematical functions. Inspired by classical functional analysis[3], we denote this new symmetry as *negational symmetry*, which is a fundamental property of QNNs because of quantum entanglement.

The main contribution of this work is to introduce and analyze the negational symmetry. We mathematically show this symmetry when the input are binary signals, thus the theoretical results are independent of a particular dataset or model (i.e. the negational symmetry is independent of the parameters of QNNs), and empirically validate it with a simulated binary pattern classification task. Our experimental results validate that, in contrast to classical NNs, negational symmetry occurs in QNNs. While our study suggests that QNNs could be a new research direction in binary signal processing, our empirical results also show that the negational symmetry is a double-edged sword in practical quantum applications. More specifically, if a binary pattern and its negational counterpart encode different information, a QNN with quantum circuits entangled is not able to separate them.

Our contributions are fourfold: (1) we formalize, prove, and analyze the negational symmetry of QNNs in quantum binary classification, (2) we propose a representation learning framework for QNNs and generalize the negational symmetry to it; (3) we evaluate the negational symmetry in binary pattern classification on a quantum simulator; (4) we discuss the advantages and disadvantages of negational symmetry in potential applications.

2. Quantum binary classification

2.1. Preliminaries

A variational quantum circuit (VQC) is a quantum circuit model that consists of a set of parametric quantum gates [18]. 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 [19,20], which demonstrates the potential values of VQCs in solving practical problems.

2.2. Problem formulation

In this work, a quantum system is a composite of two systems, namely the *input register* and the *output register*. For an intuitive interpretation, the input register and the output register can be linked with the input data and output data in a ML system, respectively. We use *Dirac* notation [21] to represent quantum data, e.g. mathematically, $\langle a|U|b\rangle$ denotes the inner (scalar) product of a row vector $\langle a|$ and a column vector $|b\rangle$, where U is a linear map. 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.[4] 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 $|1, \boldsymbol{x}\rangle = |1\rangle \otimes |\boldsymbol{x}\rangle$. The readout qubit is pre-processed and post-processed by a Hadamard gate[5], respectively.

Following previous studies [13,22,23], we analyze a family of QNNs where the entanglement exists between the readout qubit and each of the data qubits in this work. We define this status as *full entanglement* for simplicity. Let the unitary $U_{\boldsymbol{\theta}}$ be a QNN with parameters $\boldsymbol{\theta}$. As the Hadamard gate is non-parametric, we use $U_{\boldsymbol{\theta}}$ to denote the integration of the QNN of interest, and the pre- and post-processing gates on the readout qubit for convenience. 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 $\mathcal{M} \in \{X, Y, Z\}$.[6] 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

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

$$= \text{tr}(|U_{\boldsymbol{\theta}}|1, \boldsymbol{x}\rangle\langle 1, \boldsymbol{x}|U_{\boldsymbol{\theta}}^{\dagger}|\mathcal{M}_0) \tag{2}$$

where $\mathcal{M}_0$ denotes the measurement on the readout qubit instead of the whole system,[7] $-1 \leq f_{\boldsymbol{\theta}}(\boldsymbol{x}) \leq 1$ is a real number and $\text{tr}(\cdot)$ is the trace. The decision boundary on the space of density matrices is the hyperplane $\text{tr}(|U_{\boldsymbol{\theta}}|1, \boldsymbol{x}\rangle\langle 1, \boldsymbol{x}|U_{\boldsymbol{\theta}}^{\dagger}|\mathcal{M}_0 = \tau)$, 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 $\mathcal{L}$

$$\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = \max(0, 1 - y \cdot f_{\boldsymbol{\theta}}(\boldsymbol{x})). \tag{3}$$

Considering $-1 \leq f_{\boldsymbol{\theta}}(\boldsymbol{x}) \leq 1$, we choose the hinge loss over the binary cross-entropy (BCE) loss for convenience and robustness [24]. Indeed, the choice of loss function does not influence the conclusion of this study.[8]

2.3. Architecture

From the perspective of quantum computing, QNNs are variational quantum circuits (VQCs) constructed by different sets of

---

[1] See Section 2.2 for the formal definition.

[2] For simplicity, we use QNNs and QNNs with full entanglement interchangeably in the rest of the paper, as it can be inferred from the context.

[3] In a classical 2D 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 in high-dimensional systems with function $f : \mathbb{R}^N \mapsto \mathbb{R}^M$ and N-dimensional vector $\boldsymbol{x}$.

[4] We define y as an integer in a hybrid quantum-classical (HQC) system. In a pure quantum system, y can also be defined as $|y\rangle \in \{|0\rangle, |1\rangle\}$.

[5] The mathematical definition of Hadamard gate is given in the caption of Fig. 1.

[6] See A.3 for details.

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

[8] Another common choice of loss function is the fidelity loss $\mathcal{L}_{fidelity} = 1 - fidelity$. The fidelity is defined as $F(\rho_1, \rho_2) = \text{tr}(\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}})^2$, 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.

single-qubit quantum gates along with two-qubit entanglement gates. Based on *ZX-calculus* [25], we can prove that any non-linear function can be ϵ-approximated[9] with single-qubit parametric Z-gates ($R_z(\theta)$) and X-gates ($R_x(\theta)$), and two-qubit non-parametric CNOT gates.[10] 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 full entanglement [26], where θ is the parameter that we want to optimize. Two-qubit *ZZ* or *XX* interactions are know as *Ising interactions* in statistical mechanics and each block of parity gates can be viewed as a layer in classical NNs. See Fig. 1 for the illustration of the architecture. Note, any QNN with full entanglement can always be simplified to this type of architecture according to *ZX-calculus*.

2.4. 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 [27]. 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* [28]. Given an example pair $(|\boldsymbol{x}\rangle, y)$, we define the numerical gradient for a scalar parameter θ as

$$\nabla_\theta \mathcal{L}(f_\theta(\boldsymbol{x}), y) = \frac{\mathcal{L}(f_{\theta+\frac{\pi}{2}}(\boldsymbol{x}), y) - \mathcal{L}(f_{\theta-\frac{\pi}{2}}(\boldsymbol{x}), y)}{2}. \quad (4)$$

2.5. 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 Bengio et al. [29]. 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 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 $g_\theta(\boldsymbol{x})$. Similar to Eq. (1), we have

$$g_\theta(\boldsymbol{x}) = \langle 1, \boldsymbol{x}|U_\theta^\dagger|\mathcal{M}_{1,\cdots,N}|U_\theta|1, \boldsymbol{x}\rangle \quad (5)$$

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 $g_\theta : \mathbb{C}^{2^N} \mapsto [-1, 1]^N$, 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 the Frobenius norm $||g_\theta(\boldsymbol{x}_j) - g_\theta(\boldsymbol{x}_k)||$ and analyze the representations of quantum output in a classical fashion.

3. Negational symmetry of quantum neural networks

3.1. 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, quantum gate X is equivalent to bitwise NOT in classical computing. Let us denote $\boldsymbol{X} = \prod_{\otimes}^N X$ 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, $f_\theta(\boldsymbol{x}) = f_\theta(\tilde{\boldsymbol{x}})$.*

Here, the QNN should have full entanglement, as discussed in Section 2.3. Since all quantum gates in Eq. (1) are 2D matrices, i.e. linear transformations, the mathematical proof is straightforward and can be found in B.1. Note, Theorem 1 describes the situation in expectation due to the nature of measurement operation. Although Eq. (1) is a closed-form mathematical expression, in a real quantum device, the empirical observation of $f_\theta(\boldsymbol{x})$ for a single example is dependent on the average of the observed outcomes in repeated measurements, i.e. $\mathcal{M}_0$ is measured in multiple copies. That is to say, Theorem 1 may not be observed based on a single observation due to the stochasticity of quantum computing. This differs from most classical ML models which have a deterministic output in the inference phase and increases the computational cost in contrast to classical systems.

It is worth mentioning that Theorem 1 holds not only when a QNN is trained to convergence, but also for a QNN with randomly initiated $\boldsymbol{\theta}$. In a 2D 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 2.3. 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 2.2, we have $-1 \leq f_\theta(\boldsymbol{x}) \leq 1$. The negational

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

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

Table 2
Negational symmetry of representations for Pauli measurement.

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

symmetry is a built-in property of QNNs when there is at least one *ZZ* block in a QNN for binary pattern classification.

3.2. 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 2.5, we measure *N* data qubits with Pauli measurement ($\{X, Y, Z\}$). The output $g_\theta(\boldsymbol{x})$ is a *N*-element feature vector in the real domain. Given the 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) [30]. Note, although we can visualize the final representations of a QNN, we cannot access the intermediate representations in the tensor product Hilbert space directly.

In addition to the symmetry in the measurement on the readout qubit, there is also a symmetry in the learned representations. Given the same assumptions in Theorem 1, we have:

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

Compared with Theorem 1, Theorem 2 is more intuitive, where the feature vectors of the binary pattern and its negational counterpart have opposite directions in the feature space. The mathematical relationship between $g_\theta(\boldsymbol{x})$ and $g_\theta(\tilde{\boldsymbol{x}})$ 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.

4. Experiments

The purpose of the simulated experiments in this section are twofold. First, we want to validate our theoretical findings in Section 3. Second, we want to demonstrate the downside of negational symmetry, where will show a practical ML problem that QNNs cannot solve.

4.1. Experimental setting

Environment All experiments were run on a classical computer with Ubuntu 18.04 LTS.[11] We simulate the NISQ circuits with Cirq.[12] 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. Note, while simulators allow us to simulate QNNs on classical computers, this simulation does not scale up to a large number of qubits. As the Hilbert space of the input data $\mathbb{C}^{2^N}$ grows exponentially with the increase of *N*, the classical computers easily reach their memory limit to simulate the quantum process.

Hyperparameters We use the same set of hyperparameters for the training of classical NNs. We use Adam [31], a gradient-based stochastic optimization method, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$. The batch size is 32. The constant learning rate is 10^{-4}. We train all models to converge.

Dataset Following [15,26], we use the binarized MNIST dataset for binary classification tasks and we generate binary patterns with 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 and following [26], the images are downsampled from 28×28 to 4×4 to fit the simulator, but the results generalize to any size of qubits as the result of Theorem 1. We then map each grayscale pixel value to $\{0, 1\}$ with 128 as the threshold and remove the contradictory examples (the images labeled as both 3 and 6 simultaneously). 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.

4.2. Negational symmetry in binary pattern classification

We numerically validate Theorem 1 by evaluating the negational symmetry of QNNs in simulated binary pattern classification tasks. We use the 2-layer QNN described in Section 2.3. The readout qubit is measured by a Pauli *Z* operator. The QNN has a 16-qubit input register and 32 parameters in total. The test results are presented in Fig. 2(a). We also repeat 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 the digit (white to black) and the pixels of the 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* [32] if we consider the training set as the source domain and the test set as the target domain 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 apply it to the negational test set. Given Theorem 1, it is not surprising that QNNs should continue to maintain a high performance on the test set with negational operation. The test results are presented in Fig. 2(b).

To validate the universality of 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*, denoted as Q_1) to deeper QNNs, namely a 3-layer QNN (*XX-ZZ-XX*, denoted as Q_2) and a 4-layer QNN (*XX-ZZ-XX-ZZ*, denoted as Q_3), 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*, denoted as Q_4) and 3-layer QNN (*XX-ZZ-XX*, denoted as Q_5). For these larger models, 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 them on the test set without negational operation and the test set with negational operation. The results are present in Table 3. As expected, we observe for each of the QNNs
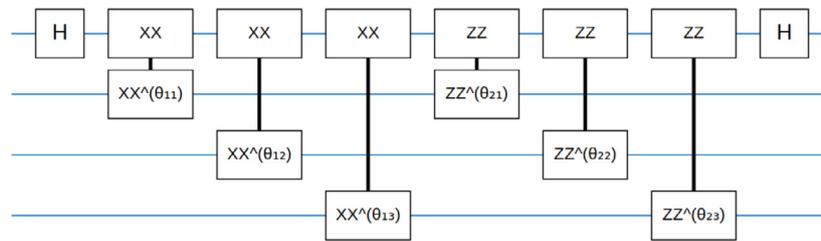

Fig. 1. The architecture of a 2-layer QNN with 3 + 1 qubits. The input register has 3 data qubits, represented by the second, third and fourth lines from the top. The output register has 1 readout qubit, represented by the top line. The first layer is formed by X-parity gates and the second layer is 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 $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. The two H gates denote the pre-processing and post-processing operations on the readout qubit.
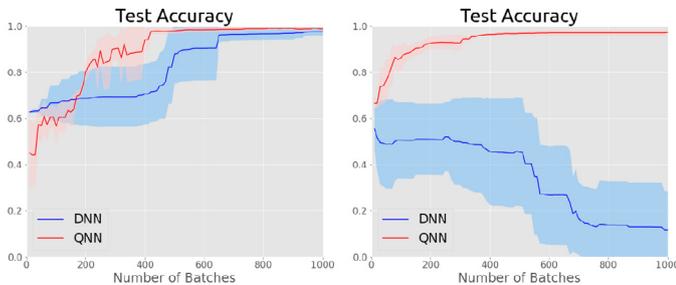
(a) Test accuracy on the test set.

(b) Test accuracy on the negational test set.

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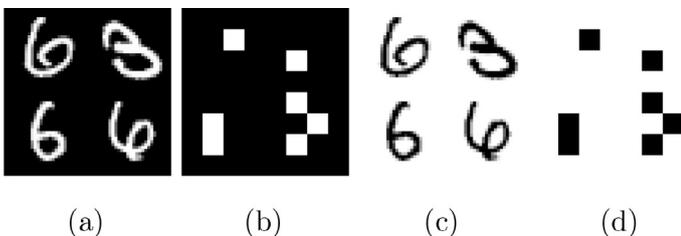

(a)   (b)   (c)   (d)

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

Table 3

Evaluation of QNNs with different architectures on negational symmetry.

Model	w/o Negation	w/ Negation	# Params
Q_1	0.9783	0.9783	32
Q_2	0.9674	0.9674	48
Q_3	0.9922	0.9922	64
Q_4	0.9707	0.9707	32
Q_5	0.9967	0.9967	48

the same performance on the test sets with and without negation. The same experiments were repeated for different two-digit combinations and the same phenomena above were observed. Note, as the negational symmetry is an inherent property of the model, it is independent of the learned parameters and also occurs in a randomly initialized model.

We further compute the difference between the logits of the QNNs, i.e. Eq. (1), for all 890 pairs in the two test sets. We find that 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 Q_1. If we take the noisy environment of the

Table 4

Comparison of QNN and classical NNs on negational symmetry. DNNs and CNNs do not have negational symmetry.

Model	w/o Negation	w/ Negation	# Params
Q_1	0.9783	0.9783	32
D_1	0.9775	0.1146	37
D_2	0.9944	0.1528	289
D_3	0.9978	0.0067	309
D_4	0.9933	0.0090	561
C_1	0.9978	0.3079	701
C_2	0.9978	0.0247	1585

NISQ device and the number of significant figures into account, we can say that QNNs make the same prediction on a binary pattern and its negational counterpart.

To contrast this with classical NNs, we repeat the experiment with a 2-layer DNN whose number of parameters is close to the QNN's number of parameters. The first layer of the DNN is a fully-connected layer with 16 input nodes and 2 output nodes, followed by a ReLU activation function [33]. The second layer is a fully-connected (FC) layer with 2 input nodes and 1 output node. The total number of parameters for the DNN is 37. We use BCE as the loss function for DNNs. The results are presented in Fig. 2. Both the DNN and the QNN achieve promising results in the first scenario on the test set without negation. The QNN seems to converge faster and more stable than the DNN, when the training does not suffer from *barren plateaux* [13]. However, the DNN fails in the negational setting. Additionally, we provide the results of 4 DNNs with different number of hidden nodes and different number of hidden layers (denoted as $D_{i=\{1,2,3,4\}}$) and CNNs including a 2-layer CNN C_1 and a 3-layer CNN C_2 in Table 4.[13] For classical NNs, more parameters and advanced architectures improve the performance on the original test set, but do not help on the negational one.

4.3. Negational symmetry in quantum representation learning

For Theorem 2, we first calculate the norm of the sum of two pairwise feature vectors $||g_{\theta}(\boldsymbol{x}) + g_{\theta}(\tilde{\boldsymbol{x}})||$[14] The sum of the norms is less than 1×10^{-6}, which numerically validates the negational symmetry in the auxiliary quantum representation learning. Following the discussion in Section 3.2, we also check the pairwise statistical similarity for the 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 -1. Additionally, we provide the t-SNE visualizations with the **same** random seed in Fig. 4. The clusters formed by the learned representations of the two classes

---

[13] See Appendix C for the details of the model architecture.

[14] Note, we use | in *Dirac* notations and use || in norm operations.

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

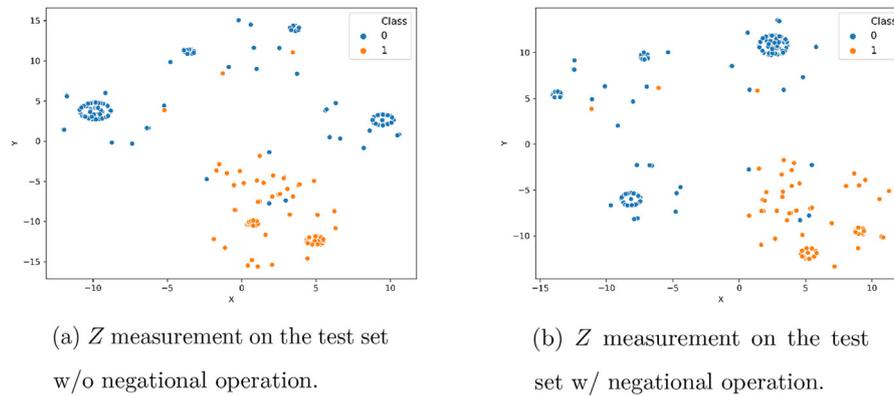(b) Z measurement on the test set w/ negational operation.

Fig. 4. t-SNE visualization of learned representations.

Table 5

Comparison of QNN and classical NNs on learning to identify the negation operation on the binary patterns. QNNs fail because of negational symmetry.

Model	Accuracy	# Params
Q_1	0.5007 ± 0.0003	32
D_1	1.0000 ± 0.0000	37
C_1	1.0000 ± 0.0000	701

can be clearly split, which suggests that the measurement on the data qubits could be a meaningful tool for quantum representation learning.

4.4. Drawback of negational symmetry

In the above experiments, we validate that QNNs cannot differentiate a binary pattern and its negational counterpart, and highlight its advantage when the two patterns encode the same semantic information (i.e. the digit). However, if the two patterns actually encode different semantic information, negational symmetry of QNNs could be hazardous to the system. Here, we provide another simple experiment where both QNNs and NNs are expected to learn to differentiate a binary pattern and its negational counterpart. We use the same networks Q_1, D_1, and C_1 as in Section 4.2. We use the same training set and test set in the above experiments. However, we include both original patterns and the corresponding negational patterns in both training set and test set this time. Instead of using the original digit labels, the original patterns are assigned the label 1 (or $|1\rangle$) and the negational patterns are assigned the label 0 (or $|0\rangle$). Thus, the objective becomes to separate the white digits on the black background from the black digits on the white background. The results are present in Table 5. As expected, based on Theorem 1, QNNs are not able to solve this simple task and achieve an accuracy of around 50% (random guess), while the classical models perfectly solve the task. It is thus crucial to note that negational symmetry in practical settings could be a double-edged sword: when a binary pattern and its negational counterpart encode different information, a system built on QNNs could be vulnerable to malicious attacks.

5. Limitations

We only study the negational symmetry of QNNs for quantum input in *pure* states (e.g. $\{|0\rangle, |1\rangle\}$). It will also be interesting to generalize the theoretical analysis to *mixed* states[15] in the future.

The practical application of QNNs may require further discussion. In fact, we can only afford a GPU-based simulated environment with 16 qubits. Limited by the number of qubits, we only have 16-bit binary patterns following [26], which limits the experimental design. In the long term, we expect more advanced mathematical and experimental tools to emerge from new joint developments in mathematics, physics, and engineering.

It is worth mentioning that this study mainly discusses on the theoretical property of QNNs in binary pattern classification, while the quantum state preparation, as a physical operation, is beyond the scope of this work. In fact, the step of quantum state preparation could easily hide the computational complexity of quantum models [34], which also limits the scalability of quantum models to handle complex tasks (e.g. it is still challenging to deploy large-scale qubits in a quantum computer [6]). However, we expect that the quantum state preparation will not be a bottleneck for QNNs, as the development of quantum hardware progresses.

6. Conclusions

In this work, we present the negational symmetry of QNNs in binary pattern classification, a fundamental property of QNNs that has not been observed previously. We formalize and prove this property and discuss the mechanisms behind. We empirically validate the existence of this new form of symmetry that is inherent in QNNs by simulated experiments and demonstrate that negational symmetry could be a double-edged sword in quantum applications. For pattern recognition research in quantum devices in the long run, we believe that a better theoretical understanding of the properties of QNNs are required and that properties such as the negational symmetry need to be taken into account when designing QNNs to solve practical problems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

---

[15] $|x\rangle = \alpha|0\rangle + \beta|1\rangle, \alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1$

Appendix A. Quantum Information Basics

A1. 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.

A2. 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, $\mathbb{H}_{AB} = \mathbb{H}_A \otimes \mathbb{H}_B = \mathbb{C}^{d_A d_B}$.

A3. Pauli matrices

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

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

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

A4. 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 $a \in \{x, y, z\}$.

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

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

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

A5. Universality of ZX-calculus

According to Eqs. (A.4), (A.5), and (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 3 [25]. *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 4 [25]. *Any n-qubit unitary can be constructed out of the CNOT gate and phase gates.*

Theorem 5. *For any nonlinear function, there exists at least one* VQC U_θ *with following properties: (1) it can be constructed out of the gate set $\mathbb{U} = \{R_x, R_z, \text{CNOT}\}$ 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 5 is a direct result of the Universality Theorem of neural networks and the Universality Theorem of ZX-calculus. $\square$

A6. 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.

Appendix B. Theoretical analysis

B1. Sketch of proof for Theorem 1

The mathematical proof for Theorem 1 is straightforward for QNNs with finite qubits. 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 follow the same logic. We use the notations in Section 2 and Section 3.

Following Eq. (1), we have

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

where

$$U_{\boldsymbol{\theta}} = (H \otimes I)(R_x(\theta_1) \otimes R_x(\theta_1))(R_z(\theta_2) \otimes R_z(\theta_2))(H \otimes I). \tag{B.2}$$

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. (B.2), we have $U_{\boldsymbol{\theta}}$ in Eq. (B.3).

$$U_{\boldsymbol{\theta}} = \frac{1}{2}\begin{bmatrix} \cos(\frac{\theta_1}{2})(e^{-i\theta_2}+1) & \cos(\frac{\theta_1}{2})(e^{i\theta_2}-1) & -i\sin(\frac{\theta_1}{2})(e^{-i\theta_2}+1) & i\sin(\frac{\theta_1}{2})(e^{-i\theta_2}-1) \\ \cos(\frac{\theta_1}{2})(e^{-i\theta_2}-1) & \cos(\frac{\theta_1}{2})(e^{i\theta_2}+1) & -i\sin(\frac{\theta_1}{2})(e^{-i\theta_2}-1) & i\sin(\frac{\theta_1}{2})(e^{-i\theta_2}+1) \\ -i\sin(\frac{\theta_1}{2})(1+e^{-i\theta_2}) & i\sin(\frac{\theta_1}{2})(1-e^{-i\theta_2}) & \cos(\frac{\theta_1}{2})(1+e^{i\theta_2}) & \cos(\frac{\theta_1}{2})(1-e^{i\theta_2}) \\ -i\sin(\frac{\theta_1}{2})(1-e^{-i\theta_2}) & i\sin(\frac{\theta_1}{2})(1+e^{-i\theta_2}) & \cos(\frac{\theta_1}{2})(1-e^{i\theta_2}) & \cos(\frac{\theta_1}{2})(1+e^{i\theta_2}) \end{bmatrix} \tag{B.3}$$

We have

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

$$= \begin{bmatrix} -\sin(\frac{\theta_1}{2})\sin(\theta_2) - i\sin(\frac{\theta_1}{2})(\cos(\theta_2)+1) \\ -\sin(\frac{\theta_1}{2})\sin(\theta_2) - i\sin(\frac{\theta_1}{2})(\cos(\theta_2)-1) \\ \cos(\frac{\theta_1}{2})(1+\cos(\theta_2)) + i\cos(\frac{\theta_1}{2})\sin(\theta_2) \\ \cos(\frac{\theta_1}{2})(1-\cos(\theta_2)) - i\cos(\frac{\theta_1}{2})\sin(\theta_2) \end{bmatrix}, \tag{B.4}$$

where

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

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

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sin^2\left(\frac{\theta_1}{2}\right) - \cos^2\left(\frac{\theta_1}{2}\right), \tag{B.5}$$

where

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

Similarly, we have

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

$$= \begin{bmatrix} \sin(\frac{\theta_1}{2})\sin(\theta_2) + i\sin(\frac{\theta_1}{2})(\cos(\theta_2)-1) \\ \sin(\frac{\theta_1}{2})\sin(\theta_2) + i\sin(\frac{\theta_1}{2})(\cos(\theta_2)+1) \\ \cos(\frac{\theta_1}{2})(1-\cos(\theta_2)) - i\cos(\frac{\theta_1}{2})\sin(\theta_2) \\ \cos(\frac{\theta_1}{2})(1+\cos(\theta_2)) + i\cos(\frac{\theta_1}{2})\sin(\theta_2) \end{bmatrix}, \tag{B.6}$$

where

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

and

$$f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}) = \langle 1,1|U_{\boldsymbol{\theta}}^{\dagger}|Z \otimes I|U_{\boldsymbol{\theta}}|1,1\rangle$$
$$= \sin^2(\frac{\theta_1}{2}) - \cos^2(\frac{\theta_1}{2}). \tag{B.7}$$

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

B2. Sketch of Proof for Theorem 2

The proof is similar to Appendix B.1. 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. (5), we have

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \langle 1,0|U_{\boldsymbol{\theta}}^{\dagger}|Z \otimes Z|U_{\boldsymbol{\theta}}|1,0\rangle, \tag{B.8}$$

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. (B.8), we have

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = (\sin^2(\frac{\theta_1}{2}) - \cos^2(\frac{\theta_1}{2}))\cos(\theta_2). \tag{B.9}$$

Similarly, we have

$$g_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}) = \langle 1,1|U_{\boldsymbol{\theta}}^{\dagger}|Z \otimes Z|U_{\boldsymbol{\theta}}|1,1\rangle$$
$$= -(\sin^2(\frac{\theta_1}{2}) - \cos^2(\frac{\theta_1}{2}))\cos(\theta_2). \tag{B.10}$$

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

Appendix C. Classical neural networks

For a comprehensive understanding of QNNs, we will show that negational symmetry is a unique property of QNNs when considering the large family of classical NNs. Here, we compare QNNs with 2 categories of classical models. Note, we choose simple models to validate negational symmetry instead of showing high performances. The first category is Deep Neural Networks (DNNs). The 2-layer DNN D_1 (*16 - 2 - 1*) in Section 4.2 is treated as the baseline. We investigate 3 variants of the baseline: (i) increasing the number of nodes in the hidden layer D_2 (*16 - 16 - 1*); (ii) increasing the number of hidden layers D_3 (*16 - 16 - 2 - 1*); and (iii) increasing both the number of nodes in the hidden layer and the number of hidden layers D_4(*16 - 16 - 16 - 1*). The second category is Convolutional Neural Networks (CNNs), which is a strong baseline. Considering the image resolution is only 4×4, we only use CNNs with simple architectures. 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 for a convolutional layer to be 16. We consider a 2-layer CNN C_1 and a 3-layer CNN C_2. Let *CV* stands for 1D convolutional layer with filter size 3 and stride 1, and *FC* stand for fully-connected layer. Given an example, the input for CNNs is a 16-element binary vector and the output is a logit. The architecture of C_1 is *CV-FC* and the architecture of C_2 is *CV-CV-FC*.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2022.108750.

References

[1] L. Bai, L. Rossi, A. Torsello, E.R. Hancock, A quantum jensen–shannon graph kernel for unattributed graphs, Pattern Recognit. 48 (2) (2015) 344–355.
[2] L. Li, L. Jiao, J. Zhao, R. Shang, M. Gong, Quantum-behaved discrete multi-objective particle swarm optimization for complex network clustering, Pattern Recognit. 63 (2017) 1–14.
[3] Z. Zhang, D. Chen, J. Wang, L. Bai, E.R. Hancock, Quantum-based subgraph convolutional neural networks, Pattern Recognit. 88 (2019) 38–49.
[4] M. Schuld, I. Sinayskiy, F. Petruccione, An introduction to quantum machine learning, Contemp. Phys. 56 (2) (2015) 172–185.

[5] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Quantum machine learning, Nature 549 (7671) (2017) 195–202.

[6] F. Arute, K. Arya, R. Babbush, D. Bacon, J.C. Bardin, R. Barends, R. Biswas, S. Boixo, F.G. Brandao, D.A. Buell, et al., Quantum supremacy using a programmable superconducting processor, Nature 574 (7779) (2019) 505–510.

[7] J. Preskill, Quantum computing in the nisq era and beyond, Quantum 2 (2018) 79.

[8] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.

[9] E. Watanabe, H. Shimizu, Relationships between internal representation and generalization ability in multi layered neural network for binary pattern classification problem, in: ICNN, volume 2, IEEE, 1993, pp. 1736–1739.

[10] T. Ravindra Babu, M. Narasimha Murty, V. Agrawal, Classification of run-length encoded binary data, Pattern Recognit. 40 (1) (2007) 321–323.

[11] H. Langseth, T.D. Nielsen, Latent classification models for binary data, Pattern Recognit. 42 (11) (2009) 2724–2736.

[12] V. Santhanam, V.I. Morariu, D. Harwood, L.S. Davis, A non-parametric approach to extending generic binary classifiers for multi-classification, Pattern Recognit. 58 (2016) 149–158.

[13] J.R. McClean, S. Boixo, V.N. Smelyanskiy, R. Babbush, H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. 9 (1) (2018) 1–6.

[14] I. Kerenidis, J. Landman, A. Prakash, Quantum Algorithms for Deep Convolutional Neural Networks, 2020 ICLR.

[15] J. Bausch, Recurrent quantum neural networks, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), NIPS, Volume 33, Curran Associates, Inc., 2020, pp. 1368–1379.

[16] J.H. Jacobsen, A. Smeulders, E. Oyallon, i-RevNet: Deep Invertible Networks, 2018 ICLR.

[17] W.K. Wootters, W.H. Zurek, A single quantum cannot be cloned, Nature 299 (5886) (1982) 802–803.

[18] J. Romero, A. Aspuru-Guzik, Variational quantum generators: generative adversarial quantum machine learning for continuous distributions, Adv. Quantum Technol. 4 (1) (2021) 2000003.

[19] Y. Cao, G.G. Guerreschi, A. Aspuru-Guzik, Quantum neuron: an elementary building block for machine learning on quantum computers, arXiv preprint arXiv:1711.11240 (2017).

[20] K. Mitarai, M. Negoro, M. Kitagawa, K. Fujii, Quantum circuit learning, Phys. Rev. A 98 (3) (2018) 032309.

[21] M.A. Nielsen, I. Chuang, Quantum Computation and Quantum Information, American Association of Physics Teachers, 2002.

[22] J.G. Liu, L. Wang, Differentiable learning of quantum circuit born machines, Phys. Rev. A 98 (6) (2018) 062324.

[23] C. Zoufal, A. Lucchi, S. Woerner, Quantum generative adversarial networks for learning and loading random distributions, NPJ Quantum Inf. 5 (1) (2019) 1–9.

[24] M. Tzelepi, A. Tefas, Improving the performance of lightweight cnns for binary classification using quadratic mutual information regularization, Pattern Recognit. 106 (2020) 107407.

[25] B. Coecke, A. Kissinger, Picturing Quantum Processes, Cambridge University Press, 2017.

[26] E. Farhi, H. Neven, et al., Classification with quantum neural networks on near term processors, Quantum Rev. Lett. 1 (2 (2020)) (2020) 10–37686.

[27] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A 99 (3) (2019), doi:10.1103/physreva.99.032331.

[28] G.E. Crooks, Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition, 2019. 1905.13311

[29] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE TPAMI 35 (8) (2013) 1798–1828.

[30] L.V.D. Maaten, G. Hinton, Visualizing data using t-SNE, JMLR 9 (Nov) (2008) 2579–2605.

[31] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 2015 ICLR.

[32] H. Shimodaira, Improving predictive inference under covariate shift by weighting the log-likelihood function, J. Stat. Plan. Inference 90 (2) (2000) 227–244.

[33] V. Nair, G.E. Hinton, in: Rectified Linear Units Improve Restricted Boltzmann Machines, 2010, pp. 807–814. ICML.

[34] J. Bang, A. Dutta, S.W. Lee, J. Kim, Optimal usage of quantum random access memory in quantum machine learning, Phys. Rev. A 99 (1) (2019) 012326.

Nanqing Dong is a Ph.D. student at the Department of Computer Science, University of Oxford. He received his master's degree from Cornell University in 2017. His research interests include machine learning, computer vision, optimization, and quantum computing.

Michael Kampffmeyer is an associate professor at the Machine Learning Group, UiT The Arctic University of Norway, as well as a Senior Researcher at the Norwegian Computing Center. His research interests include the development of deep learning algorithms that learn in the presence of limited labeled data and their interpretability.

Irina Voiculescu is an associate professor at the Department of Computer Science, University of Oxford. She is the Fellow of the UK Geometric Modelling Society. She is also an associate editor the Journal of Medical Imaging. Her research interests include computer vision, geometry, randomness, and quantum computing.

Eric P. Xing is a professor at the Machine Learning Department, Carnegie Mellon University, and the president of Mohamed bin Zayed University of Artificial Intelligence. He is the Fellow of AAAI and IEEE. He is also an associate editor of the AOAS, JASA, TPAMI, MLJ, and JMLR.