Verifying Fairness in Quantum Machine Learning

Ji Guan[1(✉)], Wang Fang[1,2], and Mingsheng Ying[1,3]

[1] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
{guanj,fangw,yingms}@ios.ac.cn
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Abstract. Due to the beyond-classical capability of quantum computing, quantum machine learning is applied independently or embedded in classical models for decision making, especially in the field of finance. Fairness and other ethical issues are often one of the main concerns in decision making. In this work, we define a formal framework for the fairness verification and analysis of quantum machine learning decision models, where we adopt one of the most popular notions of fairness in the literature based on the intuition—any two similar individuals must be treated similarly and are thus unbiased. We show that quantum noise can improve fairness and develop an algorithm to check whether a (noisy) quantum machine learning model is fair. In particular, this algorithm can find bias kernels of quantum data (encoding individuals) during checking. These bias kernels generate infinitely many bias pairs for investigating the unfairness of the model. Our algorithm is designed based on a highly efficient data structure—Tensor Networks—and implemented on Google's TensorFlow Quantum. The utility and effectiveness of our algorithm are confirmed by the experimental results, including income prediction and credit scoring on real-world data, for a class of random (noisy) quantum decision models with 27 qubits (2^{27}-dimensional state space) tripling (2^{18} times more than) that of the state-of-the-art algorithms for verifying quantum machine learning models.

Keywords: Quantum Machine Learning · Fairness Verification · Quantum Noise · Quantum Decision Model

1 Introduction

Quantum Machine Learning: Google's quantum supremacy (or advantage) experiment demonstrated that a quantum computer *Sycamore* with 53 noisy superconducting qubits can do a specific calculation, namely sampling, in 200 s

S. Shoham and Y. Vizel (Eds.): CAV 2022, LNCS 13372, pp. 408–429, 2022.
https://doi.org/10.1007/978-3-031-13188-2_20

that would take (arguably) 10,000 years on the largest classical computer using existing Algorithms [1]. More recently, a quantum computer *Jiuzhang* with 76 noisy photonic qubits was used to perform a type of Boson sampling in 20 s that would require 600 million years for a classical computer [2]. These experiments mark the beginning of the Noisy Intermediate-Scale Quantum (NISQ) computing era, where quantum computers with tens-to-hundreds of qubits become a reality, but quantum noise still cannot be avoided.

Quantum machine learning is believed to be a far frontrunner in setting a path for practical beyond-classical applications of NISQ quantum devices. This stimulates the fast development of various quantum machine learning (see [3] for a review). Stepping into industries, Google recently built up a framework *TensorFlow Quantum* for the design and training of quantum machine learning within its well-known classical machine learning platform—*TensorFlow* [4].

Classical machine learning has led to automated decision models assuming a significant role in making real-world decisions, especially in finance [5]. Such (financial) decision tasks are known to face the curse of dimensionality as there are too many features available to model customers/users. Principal component analysis (PCA) is one of the most popular methods for dimensionality reduction. It was recently shown that quantum PCA Algorithm [6] can run exponentially faster on a quantum processor. At the same time, the training process of quantum machine learning could be sped up exponentially (compared with classical training) by using quantum PCA to implement iterative gradient descent methods for network training [7]. It is worth noting that this quantum approach is generic in the sense that it can be applied to various types of neural networks, including shallow, convolutional, and recurrent networks, and thus can mitigate the high complexity issue of classical training. Because of these reasons, quantum machine learning has been introduced to be applied independently or embedded in classical decision-making models, e.g. fraud detection (in transaction monitoring) [8,9], credit assessments (risk scoring for customers) [10,11], and recommendation systems for content dissemination [12] (see reviews [13,14] for more information). Similar to the classical counterparts, the quantum models are trained on individuals' information, e.g. saving, employment, salary (encoded as quantum data).

Fairness in Machine Learning: It is well-known that classical decision models are prone to discriminating against users/consumers on the basis of characteristics such as race and gender [15], and have even led to legal mandates of ensuring *fairness*. To develop fair models, various attempts have been made to precisely define and quantify fairness. They broadly fall into two categories: *group* and *individual* fairness. Group fairness aims to achieve through statistical parity the same outcomes across different protected groups (e.g. gender or race) [16,17], whereas individual fairness advocates treating similar individuals similarly (receiving the similar outcomes) [18] (see [19,20] for various definitions of fairness and discussions about their relationship). The computer science community has endeavoured to check and avoid bias in classical decision models in the sense of different types of fairness (e.g. [18,19,21]). In particular, several verifiers for formal analysis and fairness verification have been designed and implemented, including FairSquare [22], VeriFair [23] and Justicia [24].

Inevitably, the same issue of fairness arises in the quantum models too. Furthermore, as quantum machine learning is principled by quantum mechanics, which is usually hard to explain to the end-users, it is even more important to verify fairness when a decision is made by a quantum machine learning algorithm. However, to the best of our knowledge, the verification problem of fairness in quantum algorithms has not yet been touched.

Contributions of this Paper: In this work, we define a formal framework so that the fairness of quantum machine learning decision models can be verified and analyzed in a principled way. Our *design decision* is as follows: we focus on individual fairness—*treating similar individuals similarly* [18]. The trace distance—one of the most widely used quantities in quantum information [25, Section 9.2]—is chosen as the metric for measuring the similarity of quantum data (individuals) in defining fairness. Our main technical contributions include:

(1) ***Problem Reduction***: We prove that for a given (noisy) quantum decision model, checking the fairness can be reduced to a variant of distinguishing quantum measurements (states), a fundamental problem in quantum information. We resolve this specific variant problem by finding the maximum difference between the eigenvalues of the matrices generated by quantum measurements. As a corollary, we show that quantum noise can improve fairness.
(2) ***Algorithm***: Based on (1), an algorithm is developed to exactly and efficiently check whether or not a quantum machine learning decision model is fair. A special strength of this algorithm is that it can identify *bias kernels* during the checking, and these kernels generate infinitely many *bias pairs*, that is, two similar quantum data that are not treated similarly. Then these bias pairs can be used to investigate the bias of the decision model.
(3) ***Case Studies***: The effectiveness of our algorithm is confirmed by experiments on quantum (noisy) decision models with 8 or 9 quantum bits (qubits) for income prediction and credit scoring on real-world data. In particular, its efficiency is shown by a class of random quantum decision models with 27 qubits, which works on a 2^{27}-dimensional state space. The state-of-the-art verification algorithm [26] for quantum machine learning was only able to deal with (the robustness with) 9 qubits. Our experiments can be considered a big step toward the demanded number ($\geq$50) of qubits in practical applications of the NISQ era.

1.1 Related Works and Challenges

To put our work in an appropriate context, let us further discuss some related works and the challenges we face in this paper.

Classical Versus Quantum Models: In order to identify and mitigate the bias of classical machine learning decision models, an algorithm for maximizing utility with fairness guarantee was proposed [18]. Then the strategy of searching input data with linear and integral constraints is employed in a verifier for proving individual fairness of a given decision model [21]. The verifier is sound

but not complete in general. But in the case of linear models, it is exact (both sound and complete) if the worst-case exponential time is allowed. However, although quantum decision models are always linear, the above technique cannot be directly generalized from the classical case to the quantum case. The main obstacle here is that the corresponding constraints in the quantum models are nonlinear, and thus searching the data set in a linear domain is ineffective in the quantum case. In this paper, we surmount this obstacle by reducing the quantum fairness verification problem to determining the distinguishability of a quantum measurement, which is independent of input data. Then we resolve the latter by eigenvalue analysis with polynomial time in the dimension of input quantum data. As a result, our algorithm is exact (sound and complete) and efficient.

Fairness Versus Robustness: As in the classical case, the individual fairness considered in this paper can be thought of as a kind of global robustness [21]. This will be formally discussed in Sect. 3. In the last few years, quite a few papers have been devoted to (adversarial) robustness verification of quantum machine learning (e.g. [26–28]), where a verifier is given a nominal input quantum datum and it checks robustness in a neighborhood of that particular input datum. However, the techniques developed in these works cannot be directly generalized to solve our problem of fairness verification, because we are required to check a global property. Instead, we transfer the impact of the evolution of the quantum machine learning model on input quantum data to quantum measurements.

Efficiency: As the dimension of input data increases exponentially with the number of qubits, efficiency is always a key issue in the verification of quantum machine learning models. The state-of-the-art algorithms for robustness verification mentioned above can only cope with quantum machine learning models with 9 qubits[1]. In this paper, we boost the scale up to 27 qubits on a small server, which represents a big step toward the demand in practical applications of NISQ devices ($\geq$50 qubits). The speedup originates from not only the high efficiency of our algorithm but also the based data structure we adopted—*Tensor Network* [29]—which can exploit the locality and regularity of the underlying circuits of quantum decision models and thus further optimize the algorithm.

2 Quantum Decision Models

For convenience of the reader, in this section, we review the setup of quantum (machine learning) decision models in their most basic form.

Classical Models: In the classical world, a *classification decision model* is a mapping $f_c : \mathcal{C} \to \mathcal{O}$, where $\mathcal{C}$ is a set of data to be classified, and $\mathcal{O}$ is a set of outcomes corresponding to the classes we are interested in; for example

[1] The experiments of [26] were performed on a personal computer and the size is at most 8 qubits. We have estimated and tested the same experiments on the server we used in this paper and only 9 qubits can be handled.

$\mathcal{O} = \{0, 1\}$ in the simplest non-trivial (binary) case. Such a model f_c can be generalized to be a randomized mapping $f_r : \mathcal{C} \to \mathcal{D}(\mathcal{O})$, where $\mathcal{D}(\mathcal{O})$ denotes the set of probability distributions over $\mathcal{O}$. f_r is known as a *regression decision model* to predict distributions and naturally describes a randomized classification procedure: to classify $x \in \mathcal{C}$, choose an outcome $o \in \mathcal{O}$ according to the distribution $f_r(x)$. For example, o is chosen as the outcome corresponding to the maximum probability of $f_r(x)$. Therefore, the basic form of a classical decision model is a randomized mapping $f = f_r$ ($f = f_c$ when f is degenerated to be a deterministic mapping).

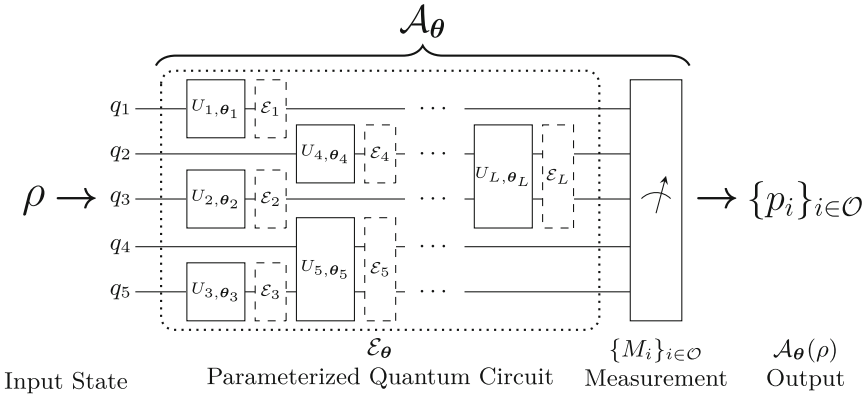

Fig. 1. Noisy Quantum (Machine Learning) Decision Model

Quantum Models: Due to the statistical nature of quantum mechanics, a quantum decision model is inherently a randomized mapping $\mathcal{A} : \mathcal{D}(\mathcal{H}) \to \mathcal{D}(\mathcal{O})$. Here $\mathcal{D}(\mathcal{H})$ is the set of *quantum states* (data) and to be specific later. Inspired by the classical models, $\mathcal{A}$ is not predefined but initialized as $\mathcal{A}_\theta$ by a parameterized quantum circuit $\mathcal{E}_\theta$ (see Fig. 1) with a set of free parameters $\theta = \{\theta_j\}_{j=1}^{L}$. Following the training strategy of classical machine learning, $\mathcal{A}_\theta$ is trained on a set of input quantum states (training dataset) by tuning θ subject to some loss function $\mathcal{L}(\theta)$.

In the following, we explain the noisy quantum decision model from the left side to the right one of Fig. 1. For the details of the training process, we refer to a comprehensive review paper [30].

Input State ρ : The input state of the model is a quantum *mixed state* ρ, which is mathematically modelled by a positive semi-definite complex matrix, written as $\rho \geq 0$, with unit trace[2]. ρ admits a decomposition form[3]: $\rho = \sum_k p_k \psi_k$

[2] ρ has unit trace if $\mathrm{tr}(\rho) = 1$, where trace $\mathrm{tr}(\rho)$ of ρ is defined as the summation of diagonal elements of ρ.

[3] This kind of decomposition is generally infinitely many, and one instance is eigen-decomposition, i.e., p_k and $|\psi_k\rangle$ are eigenvalues and eigenvectors of ρ, respectively.

where $\{p_k\}$ is a probability distribution and each ψ_k is a rank-one positive semi-definite matrix, i.e., $\psi_k = |\psi_k\rangle\langle\psi_k|$. Here, $|\psi_k\rangle$ is a unit vector and $\langle\psi_k|$ is the entry-wise conjugate transpose of $|\psi_k\rangle$, i.e., $\langle\psi_k| = |\psi_k\rangle^\dagger$. Physically, $|\psi_k\rangle$ represents a *pure state*, and ρ represents an ensemble $\{(p_k, |\psi_k\rangle)\}_k$, often called a mixed state, meaning that ρ is at $|\psi_k\rangle$ with probability p_k. In particular, if $\rho = \psi$ for some pure state $|\psi\rangle$, then the ensemble is deterministic; that is, it is degenerated to a singleton $\{(1, \psi)\}$. In general, the statistical feature of ρ may result from quantum noise, which is unavoidable in the current NISQ era, from the surrounding environment.

Example 1 (Qubits – Quantum Bits). A pure state of a single qubit q is described by a 2-dimensional unit vector and in the Dirac notation it can be written as:

$$|\psi\rangle = \begin{pmatrix} a \\ b \end{pmatrix} = a|0\rangle + b|1\rangle \text{ for } |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ and } |a|^2 + |b|^2 = 1,$$

and ensembles $\{(\frac{1}{2}, |0\rangle), (\frac{1}{2}, |+\rangle)\}$ and $\{(\frac{1}{6}, |1\rangle), (\frac{5}{6}, |\phi\rangle)\}$ of q are represented by the same 2-by-2 mixed state

$$\rho = \frac{1}{4}\begin{pmatrix} 3 & 1 \\ 1 & 1 \end{pmatrix} = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|+\rangle\langle +| = \frac{1}{6}|1\rangle\langle 1| + \frac{5}{6}|\phi\rangle\langle\phi|,$$

where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|\phi\rangle = \frac{1}{\sqrt{10}}(3|0\rangle + |1\rangle)$.

For a system of multiple qubits $q_1, ..., q_n$, the state space is a 2^n-dimensional Hilbert (linear) space, denoted by $\mathcal{H}$. As a result, pure and mixed states on $\mathcal{H}$ are 2^n-dimensional unit vectors and $2^n \times 2^n$ positive semi-definite matrices with unit trace, respectively. It is worth noting that *the dimension 2^n of the state space $\mathcal{H}$ of quantum states is exponentially increasing with the number n of qubits*. Thus, describing a quantum system with a large number of qubits and verifying its properties on a classical computer is challenging. For our purpose of verifying fairness in quantum machine learning, we adopt a compact data structure—*Tensor Networks*—to mitigate this issue (see this in Sect. 6).

Parameterized Quantum Circuit $\mathcal{E}_{\boldsymbol{\theta}}$: Several different types of parameterized quantum circuits have been proposed; e.g. quantum neural networks (QNNs) [31] and quantum convolutional neural networks (QCNNs) [32]. Basically, $\mathcal{E}_\theta$ consists of a sequence of quantum operations: $\mathcal{E}_\theta = \mathcal{E}_{d,\theta_d} \circ \cdots \circ \mathcal{E}_{1,\theta_1}$. For each input quantum state ρ, the output of the circuit is $\mathcal{E}_\theta(\rho) = \mathcal{E}_{d,\theta_d}(\ldots \mathcal{E}_{2,\theta_2}(\mathcal{E}_{1,\theta_1}(\rho)))$. In the current NISQ era, each component $\mathcal{E}_{i,\theta_i}$ is:

- either a parameterized quantum gate $\mathcal{U}_{i,\theta_i}$ (the full boxes in Fig. 1) with $\mathcal{U}_{i,\theta_i}(\rho) = U_{i,\theta_i}\rho U_{i,\theta_i}^\dagger$, where U_{i,θ_i} is a unitary matrix with parameters θ_i, i.e., $U_{i,\theta_i}^\dagger U_{i,\theta_i} = U_{i,\theta_i} U_{i,\theta_i}^\dagger = I$ (the identity matrix), and $U_{i,\theta_i}^\dagger$ is the entry-wise conjugate transpose of U_{i,θ_i};
- or a quantum noise $\mathcal{E}_i$ (the dashed boxes in Fig. 1). Mathematically, it can be described by a family of Kraus matrices $\{E_{ij}\}$ [25]: $\mathcal{E}_i(\rho) = \sum_j E_{ij}\rho E_{ij}^\dagger$ with $\sum_j E_{ij}^\dagger E_{ij} = I$. Briefly, $\mathcal{E}_i$ is represented as $\mathcal{E}_i = \{E_{ij}\}$.

Note that in constructing a quantum machine learning model, only quantum gate $\mathcal{U}_{i,\theta_i}$ is parameterized, and noises $\mathcal{E}_i$ are not because they come from the outside environment.

It should be pointed out that, in a practical model, as shown in Fig. 1, each quantum operation $\mathcal{E} = \mathcal{E}_{i,\theta_i}$ non-trivially applies on one or two qubits. For example, if $\mathcal{E}$ only works on the first qubit, then $\mathcal{E} = \mathcal{E}_1 \otimes \mathrm{id}_2 \otimes \ldots \otimes \mathrm{id}_n$ and $\mathcal{E}(\rho_1 \otimes \rho_2 \otimes \ldots \otimes \rho_n) = \mathcal{E}_1(\rho_1) \otimes \rho_2 \otimes \ldots \otimes \rho_n$, where ρ_i is the mixed state applied on qubit q_i and tensor product $\rho_1 \otimes \rho_2 \otimes \ldots \otimes \rho_n$ is the joint state of multiple qubits $q_1, \ldots, q_n$. This locality feature will be exploited by Tensor Networks to optimize our verification algorithm for fairness in the Evaluation Section—Sect. 6.

Example 2. Consider the 1-qubit noise model: $\mathcal{E}_U(\rho) = (1-p)\rho + pU\rho U^\dagger$ where $0 \leq p \leq 1$ is a probability and U is a unitary matrix. It includes the following typical noises depending on the choice of U: $U = X$ for bit flip, $U = Z$ for phase flip and $U = Y = \imath XZ$ for bit-phase flip [25, Section 8.3], where I, X, Y, Z are *the Pauli matrices*:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; Y = \begin{pmatrix} 0 & -\imath \\ \imath & 0 \end{pmatrix}, \; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \; I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

where $\imath$ denotes imaginary unit. The depolarizing noise combines the above three kinds of noise: $\mathcal{E}_D(\rho) = (1-p)\rho + p\frac{I}{2} = (1 - \frac{3p}{4})\rho + \frac{p}{4}(X\rho X + Y\rho Y + Z\rho Z)$.

Measurement $\{M_i\}_{i\in\mathcal{O}}$: At the end of parameterized quantum circuit $\mathcal{E}_\theta$, we cannot directly read out the output $\mathcal{E}_\theta(\rho)$. The only way allowed by quantum mechanics to extract classical information from $\mathcal{E}_\theta(\rho)$ is through a quantum measurement, which is mathematically modeled by a set $\{M_i\}_{i\in\mathcal{O}}$ of matrices with $\mathcal{O}$ being the set of possible outcomes and $\sum_{i\in\mathcal{O}} M_i^\dagger M_i = I$. This observing process is probabilistic: for the measurement on state $\mathcal{E}_\theta(\rho)$, an outcome $i \in \mathcal{O}$ is obtained with probability $p_i = \mathrm{tr}(M_i\mathcal{E}_\theta(\rho)M_i^\dagger)$[4]. Therefore, the output of quantum machine learning model $\mathcal{A}_\theta$ upon an input ρ is a probability distribution $\mathcal{A}_\theta(\rho) = \{p_i : p_i = \mathrm{tr}(M_i\mathcal{E}_\theta(\rho)M_i^\dagger)\}$, as depicted at the rightmost of Fig. 1.

In this paper, we focus on the well-trained quantum machine learning models (i.e., θ has been tuned), so we ignore the θ in $\mathcal{E}_\theta$ and $\mathcal{A}_\theta$. Now, we can formally specify quantum decision model $\mathcal{A}$ as follows:

Definition 1. *A quantum decision model $\mathcal{A} = (\mathcal{E}, \{M_i\}_{i\in\mathcal{O}})$ is a randomized mapping:*

$$\mathcal{A} : \mathcal{D}(\mathcal{H}) \to \mathcal{D}(\mathcal{O}) \qquad \mathcal{A}(\rho) = \{\mathrm{tr}(M_i\mathcal{E}(\rho)M_i^\dagger)\}_{i\in\mathcal{O}} \quad \forall \rho \in \mathcal{D}(\mathcal{H}),$$

where $\mathcal{E}$ is a super-operator on Hilbert space $\mathcal{H}$, and $\{M_i\}_{i\in\mathcal{O}}$ is a quantum measurement on $\mathcal{H}$ with $\mathcal{O}$ being the set of measurement outcomes (classical information) we are interested in.

[4] After measuring $\mathcal{E}_\theta(\rho)$ with outcome $i \in \mathcal{O}$, the state $\mathcal{E}_\theta(\rho)$ will be collapsed (changed) to $\rho_i' = M_i\mathcal{E}_\theta(\rho)M_i^\dagger/p_i$. As we can see, the post-measurement state ρ_i' is dependent on the measurement outcome i. This special property is vitally different from the classical computation.

Like their classical counterparts, quantum decision models are usually classified into two categories: *regression* and *classification* models. Regression models generally predict a value/quantity, whereas classification models predict a label/class. More specifically, a regression model $\mathcal{A}_{\mathcal{R}}$ uses the output of $\mathcal{A}$ directly as the predicted value of the regression variable $\rho \in \mathcal{D}(\mathcal{H})$. That is $\mathcal{A}_{\mathcal{R}}(\rho) = \mathcal{A}(\rho)$ for all $\rho \in \mathcal{D}(\mathcal{H})$. In the classical world, regression models have been successfully applied to many real-world applications, such as stock market prediction and object detection. Quantum regression models were recently used to predict molecular atomization energies [33] and the demonstration of IBM's programming platform—Qiskit [34, Variational Quantum Regression]. On the other hand, classification model $\mathcal{A}_{\mathcal{C}}$ further uses the measurement outcome probability distribution $\mathcal{A}(\rho)$ to sign a class label on the input state ρ. The most common way is as follows:

$$\mathcal{A}_{\mathcal{C}} : \mathcal{D}(\mathcal{H}) \to \mathcal{O} \qquad \mathcal{A}_{\mathcal{C}}(\rho) = \arg\max_i \mathcal{A}(\rho)_i \quad \forall \rho \in \mathcal{D}(\mathcal{H}), i \in \mathcal{O},$$

where $\mathcal{A}(\rho)_i$ denotes the i-th element of distribution $\mathcal{A}(\rho)$. Classical classification models have broad applications in our daily life, such as face recognition and medical image classification. Quantum classification models have been used to implement quantum phase recognition [32] and cluster excitation detection [4] from real-world physical problems, and fraud detection [8] in finance.

As we saw above, although classical and quantum decision models f and $\mathcal{A}$ are both randomized mappings, the input data to them and their procedure of processing the data are fundamentally different. These differences make that the techniques for verifying classical models cannot be directly applied to quantum models and we have to develop new techniques for the latter.

3 Defining Fairness

As discussed in the Introduction, an important issue in classical machine learning is: how fair is the decision made by machines. The same issue exists for quantum machine learning. Intuitively, the fairness of quantum decision model $\mathcal{A}$ is to treat all input states equally, i.e., there is not a pair of two closed input states that has a large difference between their corresponding outcomes. Formally,

Definition 2 (Bias Pair). *Suppose we are given a quantum decision model $\mathcal{A} = (\mathcal{E}, \{M_i\}_{i\in\mathcal{O}})$, two distance metrics $D(\cdot,\cdot)$ and $d(\cdot,\cdot)$ on $\mathcal{D}(\mathcal{H})$ and $\mathcal{D}(\mathcal{O})$, respectively, and two small enough threshold values $1 \geq \varepsilon, \delta > 0$. Then (ρ, σ) is said to be an (ε, δ)-bias pair if the following is true*

$$[D(\rho, \sigma) \leq \varepsilon] \wedge [d(\mathcal{A}(\rho), \mathcal{A}(\sigma)) > \delta]. \tag{1}$$

The first condition in (1) indicates that the distance between input states ρ and σ is within ε, and the second condition shows the difference between outcomes $\mathcal{A}(\rho)$ and $\mathcal{A}(\sigma)$ is beyond δ. Sometimes, without any ambiguity, (ρ, σ) is called a bias pair if ε and δ are preset.

Definition 3 (Fair Model). *Let $\mathcal{A} = (\mathcal{E}, \{M_i\}_{i\in\mathcal{O}})$ be a decision model. Then $\mathcal{A}$ is (ε, δ)-fair if there is no any (ε, δ)-bias pair.*

The intuition behind this notion of fairness is that small or non-significant perturbation of a sample ρ to σ (i.e. $D(\rho, \sigma) \leq \varepsilon$) must not be treated "differently" by a fair model. The choice of input distance function $D(\cdot, \cdot)$ identifies the perturbations to be considered non-significantly, while the choice of the output distance function $d(\cdot, \cdot)$ limits the changes allowed to the perturbed outputs in the model.

Fairness Implying Robustness: As the same in the classical situation [21], robustness of quantum machine learning is a special case of fairness defined above. Formally, robustness is defined on a specific state ρ: given a quantum model $\mathcal{A} = (\mathcal{E}, \{M_i\}_{i\in\mathcal{O}})$, ρ is (ε, δ)-robust if for all $\sigma \in \mathcal{D}(\mathcal{H})$, $D(\rho, \sigma) \leq \varepsilon$ implies $d(\mathcal{A}(\rho), \mathcal{A}(\sigma)) \leq \delta$. In contrast, fairness is established on all quantum states: $\mathcal{A}$ is (ε, δ)-fair if and only if ρ is (ε, δ)-robust for all states $\rho \in \mathcal{D}(\mathcal{H})$. So, *fairness implies robustness and can be thought of as global robustness.*

Choice of Distances: The reader should have noticed that the above definition of fairness for quantum decision models is similar to that for classical decision models. But an intrinsic distinctness between them comes from the choice of distances $D(\cdot, \cdot)$ and $d(\cdot, \cdot)$. In the classical case, the distances define the similarity between individuals and their appropriate choices have been intensively discussed [18]. One of the most used distances is total variation distance, measuring the closeness of individuals encoded by probability distributions. In this paper, we use it as $d(\cdot, \cdot)$ for measurement outcome distributions in Definition 1 and choose $D(\cdot, \cdot)$ to be the trace distance. Trace distance is essentially a generalization of total variation distance, and has been widely used by the quantum computation and quantum information community to define the closeness of quantum states [25, Section 9.2]. Formally, for two quantum states $\rho, \sigma \in \mathcal{D}(\mathcal{H})$,

$$D(\rho, \sigma) = \frac{1}{2}\mathrm{tr}(|\rho - \sigma|),$$

where $|\rho - \sigma| = \Delta_+ + \Delta_-$ if $\rho - \sigma = \Delta_+ - \Delta_-$ with $\mathrm{tr}(\Delta_+\Delta_-) = 0$ and $\Delta_\pm$ being positive semi-definite matrix. On the other hand, for two probability distributions $p = \{p_i\}_{i\in\mathcal{O}}$, $q = \{q_i\}_{i\in\mathcal{O}}$ over $\mathcal{O}$, $d(p, q) = \frac{1}{2}\sum_i |p_i - q_i|$. In particular, for the measurement outcome distributions, we have:

$$d(\mathcal{A}(\rho), \mathcal{A}(\sigma)) = \frac{1}{2}\sum_i |\mathrm{tr}(M_i^\dagger M_i \mathcal{E}(\rho - \sigma))|.$$

If ρ and σ are both diagonal matrices, i.e., $\rho = \mathrm{diag}(p_1, \cdots, p_{|\mathcal{O}|})$ and $\sigma = \mathrm{diag}(q_1, \cdots, q_{|\mathcal{O}|})$, then $D(\rho, \sigma) = d(p, q)$.

4 Characterizing Fairness

In this section, we give a characterization of fairness in terms of the Lipschitz constant and clarify its relationship with quantum noises.

4.1 Fairness and Lipschitz Constant

The Lipschitz constant has been widely used in classical machine learning for applications ranging from robustness and fairness certification of classifiers to stability analysis of closed-loop systems with reinforcement learning controllers (e.g. [35,36]). In this subsection, we show that there also exists a close connection between the Lipschitz constant and fairness in the quantum setting. Let us start from an observation:

Lemma 1. *Let $\mathcal{A} = (\mathcal{E}, \{M_i\}_{i\in\mathcal{O}})$ be a quantum decision model. Then*

$$d(\mathcal{A}(\rho), \mathcal{A}(\sigma)) \leq D(\rho, \sigma). \tag{2}$$

Proof. See Appendix A in [37] for the proof.

The above lemma indicates that quantum decision model $\mathcal{A}$ is automatically (ε, δ)-fair whenever $\varepsilon = \delta$. Furthermore, we see that $\mathcal{A}$ is unconditionaly *Lipschitz continuous*: there exists a constant $K > 0$ ($K \leq 1$ by Lemma 1) such that for all $\rho, \sigma \in \mathcal{D}(\mathcal{H})$,

$$d(\mathcal{A}(\rho), \mathcal{A}(\sigma)) \leq KD(\rho, \sigma). \tag{3}$$

As usual, K is called a *Lipschitz constant* of $\mathcal{A}$. Furthermore, the smallest K, denoted by K^*, is called the (best) Lipschitz constant of $\mathcal{A}$.

In the context of quantum machine learning, the following theorem shows that K^* actually measures the fairness of decision model $\mathcal{A}$, i.e., the best (maximum) ratio of δ and ε in a fair model, and the states ψ, ϕ achieving K^* can be used to find bias pairs in fairness verification.

Theorem 1. *1. Given a quantum decision model $\mathcal{A} = (\mathcal{E}, \{M_i\}_{i\in\mathcal{O}})$ and $1 \geq \varepsilon, \delta > 0$, $\mathcal{A}$ is (ε, δ)-fair if and only if $\delta \geq K^*\varepsilon$.*

2. If $\mathcal{A}$ is not (ε, δ)-fair, then (ψ, ϕ) achieving K^* is a bias kernel; that is, for any quantum state $\sigma \in \mathcal{D}(\mathcal{H})$, (ρ_ψ, ρ_ϕ) is a bias pair where

$$\rho_\psi = \varepsilon\psi + (1-\varepsilon)\sigma \qquad \rho_\phi = \varepsilon\phi + (1-\varepsilon)\sigma. \tag{4}$$

Proof (Outline). The "if" direction of the first claim is derived by the definitions of (ε, δ)-fairness and K^* together with (3). The "only if" direction of the first claim and the second claim are both based on the existence of pure states $|\psi\rangle$ and $|\phi\rangle$ achieving K^*: $d(\mathcal{A}(\psi), \mathcal{A}(\phi)) = K^*D(\psi, \phi)$. The detailed proof is presented in Appendix B in [37].

4.2 Fairness and Noises

In this subsection, we turn to consider the relation between fairness and noise. Let us first examine a simple example. Assume a noiseless quantum decision model $\mathcal{A} = (\mathcal{U}, \{\mathcal{M}_i\}_{i\in\mathcal{O}})$ where $\mathcal{U}$ is a unitary operator, i.e., $\mathcal{U} = \{U\}$ for some unitary matrix U. The 1-qubit depolarizing noise in Example 2 can be generalized to a large-size system with the following form:

$$\mathcal{E}(\rho) = (1-p)\rho + p\frac{I}{N} \quad \forall \rho \in \mathcal{D}(\mathcal{H}),$$

where $0 \leq p \leq 1$ and N is the dimension of the state space $\mathcal{H}$ of the system. By introducing it into $\mathcal{A}$, we obtain a noisy model $\mathcal{A}_{\mathcal{E}} = (\mathcal{E} \circ \mathcal{U}, \{\mathcal{M}_i\}_{i \in \mathcal{O}})$. Let K^* and $K^*_{\mathcal{E}}$ be the Lipschitz constants of $\mathcal{A}$ and $\mathcal{A}_{\mathcal{E}}$, respectively. A calculation (with the help of Theorem 3 below) yields:

$$K^*_{\mathcal{E}} = (1-p)K^*. \tag{5}$$

Theorem 1 indicates that the less the Lipschitz constant is, the fairer the quantum machine learning model will be. So, depolarizing noise improves fairness by the order of $(1-p)$. By the way, it was shown in [38] that depolarizing noise can improve the robustness of quantum machine learning. This result can be strengthened by using (5) to quantitatively characterize the robustness improvement.

The observation in the above example can actually be generalized to the following:

Theorem 2. *Let $\mathcal{A} = (\mathcal{U}, \{M_i\}_{i \in \mathcal{O}})$ be a quantum decision model. Then for any quantum noise represented by a super-operator $\mathcal{E}$, we have $K^*_{\mathcal{E}} \leq K^*$, where K^* and $K^*_{\mathcal{E}}$ are the Lipschitz constants of $\mathcal{A}$ and $\mathcal{A}_{\mathcal{E}} = (\mathcal{E} \circ \mathcal{U}, \{\mathcal{M}_i\}_{i \in \mathcal{O}})$.*

Proof (Outline). The proof of this theorem mainly depends on the observation that the range of $\mathcal{A}_{\mathcal{E}}$ is a subset of the range of $\mathcal{A}$, i.e. $\{\mathcal{E} \circ \mathcal{U}(\rho) : \rho \in \mathcal{D}(\mathcal{H})\} \subseteq \{\mathcal{U}(\rho) : \rho \in \mathcal{D}(\mathcal{H})\} = \mathcal{D}(\mathcal{H})$. Subsequently, by Definition 2 of fairness, the output distributions of $\mathcal{A}_{\mathcal{E}}$ are contained in that of $\mathcal{A}$. A restatement of this theorem in terms of quantum states (measurements) distinguishability and its full proof are presented in Appendix C in [37].

Remark 1. The above theorem indicates that adding noises at the end of noiseless computation can always improve fairness. Indeed, this is also true when the noises appear in the middle (after any gate in the circuit).

5 Fairness Verification

In this section, we develop an algorithm for the fairness verification of quantum decision models based on the theoretical results obtained in the last section. Formally, the major problem concerned in this paper is the following:

Problem 1 (Fairness Verification Problem). Given a quantum decision model $\mathcal{A}$ and $1 \geq \varepsilon, \delta > 0$, check whether or not $\mathcal{A}$ is (ε, δ)-fair. If not then (at least) one bias pair (ρ, σ) is provided.

5.1 Computing the Lipschitz Constant

First of all, we note that essentially, Theorem 1 gives a verification condition for fairness in terms of the Lipschitz constant K^*. Therefore, computing K^* is crucial for fairness verification. However, this problem is much more difficult than that in the classical counterpart as discussed in Subsect. 1.1. The following theorem provides a method to compute the Lipschitz constant K^* by evaluating the eigenvalues of certain matrices.

Theorem 3. *1. Given a quantum decision model $\mathcal{A} = (\mathcal{E}, \{M_i\}_{i\in\mathcal{O}})$. The Lipschitz constant K^* is:*

$$K^* = \max_{A\subseteq\mathcal{O}}[\lambda_{\max}(M_A) - \lambda_{\min}(M_A)] \text{ with } M_A = \sum_{i\in A}\mathcal{E}^\dagger(M_i^\dagger M_i),$$

where $\mathcal{E}^\dagger$ is the conjugate map[5] of $\mathcal{E}$, and $\lambda_{\max}(M_A)$ and $\lambda_{\min}(M_A)$ are the maximum and minimum eigenvalues of positive semi-definite matrix M_A, respectively.

2. Furthermore, let $A^* \subseteq \mathcal{O}$ be an optimal solution of reaching the Lipschitz constant, i.e.,

$$A^* = \arg\max_{A\subseteq\mathcal{O}}[\lambda_{\max}(M_A) - \lambda_{\min}(M_A)]$$

and $|\psi\rangle$ and $|\phi\rangle$ be two normalized eigenvectors corresponding to the maximum and minimum eigenvalues of M_{A^*}, respectively. Then we have

$$d(\mathcal{A}(\psi), \mathcal{A}(\phi)) = K^* D(\psi, \phi) = K^*,$$

where $\psi = |\psi\rangle\langle\psi|$ and $\phi = |\phi\rangle\langle\phi|$.

Proof (Outline). This theorem can be proved by reducing the problem of calculating the Lipschitz constant to determining the distinguishability of a quantum measurement. Then we claim that the distinguishability is the maximum difference between the eigenvalues of the matrices generated by the measurement. The details are quite involved, and we postpose them into Appendix C in [37].

Based on the above theorem, we are able to develop Algorithm 1 for computing the Lipschitz constant K^*. The correctness and complexity are provided in the next subsection.

5.2 Fairness Verification Algorithm

Now we are ready to present our main algorithm—Algorithm 2—for verifying fairness of quantum decision models.

To see the correctness of Algorithm 2, let us first note that the second part of Theorem 3 shows that K^* can be achieved by $d(\mathcal{A}(\psi), \mathcal{A}(\phi))$ for two mutually orthogonal quantum (pure) states ψ and ϕ. On the other hand, the second part of Theorem 1 asserts that such states ψ and ϕ form a bias kernel. Moreover, since state $\sigma \in \mathcal{D}(\mathcal{H})$ in (4) is arbitrary and $\mathcal{D}(\mathcal{H})$ is an infinite set, infinitely many bias pairs can be generated from this kernel.

To analyze the complexities of Algorithm 2 and its subroutine—Algorithm 1, we first see by Theorem 1 that for evaluating the (ε, δ)-fairness of quantum decision model $\mathcal{A}$, the Lipschitz constant K^* is sufficient and necessary. Thus the first step (Line 1) of Algorithm 2 is to call Algorithm 1 to compute K^* by the mean of Theorem 3. The complexity of Algorithm 1 mainly attributes to computing $W_i = \sum_{j\in\mathcal{J}} E_j^\dagger M_i^\dagger M_i E_j$ for each $i \in \mathcal{O}$, and for each $A \subseteq \mathcal{O}$, $\sum_{i\in A} W_i$

[5] $\mathcal{E}^\dagger(\rho) = \sum_{j\in\mathcal{J}} E_j^\dagger \rho E_j$ if $\mathcal{E}$ admits Kraus matrix form $\mathcal{E}(\rho) = \sum_{j\in\mathcal{J}} E_j \rho E_j^\dagger$.

Algorithm 1. Lipschitz($\mathcal{A}$)

Input: A quantum decision model $\mathcal{A} = (\mathcal{E} = \{E_j\}_{j\in\mathcal{J}}, \{M_i\}_{i\in\mathcal{O}})$ on a Hilbert space $\mathcal{H}$ with dimension N.

Output: The Lipschitz constant K^* and (ψ, ϕ) as in Theorem 3.

for each $i \in \mathcal{O}$ **do**
  $W_i = \mathcal{E}^\dagger(M_i^\dagger M_i) = \sum_{j\in\mathcal{J}} E_j^\dagger M_i^\dagger M_i E_j$
end for
$K^* = 0$, $A^* = \emptyset$ be an empty set and $M_{A^*} = \mathbf{0}$, zero matrix.
for each $A \subseteq \mathcal{O}$ **do**
  $M_A = \sum_{i\in A} W_i$ and $K_A = \lambda_{\max}(M_A) - \lambda_{\min}(M_A)$
  if $K_A > K^*$ **then**
    $K^* = K_A$, $A^* = A$ and $M_{A^*} = M_A$
  end if
end for
$|\psi\rangle$ and $|\phi\rangle$ are obtained two normalized eigenvectors corresponding to the maximum and minimum eigenvalues of M_{A^*}, respectively.
return K^* and (ψ, ϕ)

Algorithm 2. FairVeriQ($\mathcal{A}, \varepsilon, \delta$)

Input: A quantum decision model $\mathcal{A} = (\mathcal{E} = \{E_j\}_{j\in\mathcal{J}}, \{M_i\}_{i\in\mathcal{O}})$ on a Hilbert space $\mathcal{H}$ with dimension N, and real numbers $1 \geq \varepsilon, \delta > 0$.

Output: **true** indicates $\mathcal{A}$ is (ε, δ)-fair or **false** with a bias kernel pair (ψ, ϕ) indicates $\mathcal{A}$ is not (ε, δ)-fair.

1: $(K^*, (\psi, \phi))$=Lipschitz($\mathcal{A}$) *// Call Algorithm 1*
2: **if** $\delta \geq K^*\varepsilon$ **then**
3: **return true**
4: **else**
5: **return false** and (ψ, ϕ)
6: **end if**

and its maximum and minimum eigenvalues (and the corresponding eigenvectors for $A = A^*$ at the end). The former calculation needs $O(N^5)$ as the multiplication of $N \times N$ matrices needs $O(N^3)$ operations, and the number $|\mathcal{J}|$ of the Kraus operators $\{E_j\}_{j\in\mathcal{J}}$ of $\mathcal{E}$ can be at most N^2 [39, Chapter 2.2]; the complexity of the latter one is $O(2^{|\mathcal{O}|}|\mathcal{O}|N^2)$ since the number of subsets of $\mathcal{O}$ is $2^{|\mathcal{O}|}$, $|A| \leq |\mathcal{O}|$ for any $A \subseteq \mathcal{O}$ and computing maximum and minimum eigenvalues with corresponding eigenvectors of $N \times N$ matrix costs $O(N^2)$. Therefore, the total complexity of Algorithm 1 is $O(N^5 + 2^{|\mathcal{O}|}|\mathcal{O}|N^2)$. After that, in Lines 2-6, we simply compare δ and $K^*\varepsilon$ to answer the fairness verification problem. So, Algorithm 2 shares the same complexity with Algorithm 1.

Theorem 4. *The worst case complexities of Algorithms 1 and 2 are both* $O(N^5 + 2^{|\mathcal{O}|}|\mathcal{O}|N^2)$*, where* N *is the dimension of input Hilbert state space* $\mathcal{H}$ *and* $|\mathcal{O}|$ *is the number of the measurement outcome set* $\mathcal{O}$*.*

Like their classical counterparts, quantum machine learning models usually downscale large-dimension input data to small-size outputs. This means that the

number $|\mathcal{O}|$ of the measurement outcome set $\mathcal{O}$ is far smaller than the dimension N of input Hilbert state space $\mathcal{H}$. It is even a constant 2 in most real-world tasks for binary decisions/classifications, such as income prediction and credit scoring (see the examples in Sect. 6), and in this case, the complexities of Algorithms 1 and 2 are both $O(N^5)$. However, the dimension N is exponential in the number n of the input qubits, i.e., $N = 2^n$. Thus the complexity turns out to be $O(2^{5n})$. In verification of classical models, this *state-space explosion problem* [40] can be mitigated by using some custom-made data structures to capture the features of the underlying data, e.g. Binary Decision Diagrams (BDDs) [41]. In the quantum case, we cross this difficulty by employing a quantum data structure—*Tensor Networks (TNs)*, originating from quantum many-body physics—to exploit the locality and regularity of the circuits representing quantum machine learning models. As a result, quantum models with up to $n = 27$ qubits can be handled by our verification algorithm.

6 Evaluation

In this section, we evaluate the efficiency of our verification algorithm (Algorithm 1) on noisy quantum decision models. The algorithm is implemented on *TensorFlow Quantum* [4]—a platform of Google for designing and training quantum machine learning algorithms. Then we test it by verifying the fairness of two groups of examples:

- *Small-scale models trained from real-world data* (Subsect. 6.1): There is still no public benchmarks for quantum decision models. We choose two publicly available financial datasets, *German Credit Data* [42] and *Adult Income Dataset* from *Diverse Counterfactual Explanations Dataset* [43] and train small-scale quantum models from them on TensorFlow Quantum. Then we evaluate the Lipschitz constant K^* of the trained models by Algorithm 1.
- *Medium-scale models* (Subsect. 6.2): Medium-scale models (10–30 qubits) are difficult to be trained on TensorFlow Quantum with a personal computer or a small server since the simulated quantum noises lead to large-size (up to $2^{30} \times 2^{30}$) matrix manipulations. Thus we turn to using a model from the tutorial of TensorFlow Quantum as a seed to generate a group of medium-scale models. The efficiency of our algorithm is then demonstrated on these models with randomly sampled parameters.

All source codes can be found at: https://github.com/Veri-Q/Fairness. All our experiments are carried out on a server with Intel Xeon Platinum 8153 @ 2.00 GHz × 256 Processors, 2048 GB Memory and no dedicated GPU. The machine runs Centos 7.7.1908 and each experiment is run with at most 80 processors. We use the `NumPy` and Google `TensorNetwork` [44] Python packages to compute Lipschitz constants and bias kernels for small-scale models and medium-scale models, respectively. These two packages have their own advantages in different sizes.

6.1 A Practical Application in Finance

Adult Income Dataset. The original version of this dataset is extracted from the 1994 Census database by Barry Becker [45]. We use the modified version of the adult income dataset by DiCE [43]. Each individual in this modified dataset has 8 features and the classification whether the income exceeds $\$50,000$/year or not. We randomly select $1,000$ and 400 data from the training dataset and test dataset contained in this modified dataset, respectively. The task of the quantum decision model task is to predict whether an individual's income exceeds $\$50,000$/year or not.

German Credit Dataset. This dataset contains $1,000$ loan applicants with 20 features and the classification whether they are considered as having good credit risk or not (Creditability). It provides 500 applicants for the training and 500 applicants for the test. By using the p-value with creditability for each variable [46], we have 9 features (e.g., Account Balance, Payment Status) left as significant predictors. The task of the quantum model to be trained is to classify whether the person has good credit risk or not.

These datasets contain some categorical features, which are transformed into different integer numbers for further operations. Then we have $n \in \{8, 9\}$ numbering features in total and use the following data-encoding feature map:

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \mapsto |\psi(\boldsymbol{x})\rangle = \bigotimes_{j=1}^{n} X^{x_j} |0\rangle$$

for Pauli matrix X defined in Example 2 to encode an n-dimensional feature vector $\boldsymbol{x}$ (each dimension is normalized by its maximum value) to an n-qubits quantum state $\psi(\boldsymbol{x}) = |\psi(\boldsymbol{x})\rangle\langle\psi(\boldsymbol{x})|$.

Models: For the quantum decision model, we choose the basic rotation and entangling building blocks [47] to construct parameterized quantum circuits (see Fig. 2). In the rotation block, without any ambiguity, we directly use X and Z to represent parameterized X-rotation $e^{-\imath\frac{\theta_1}{2}X}$ and parameterized Z-rotation $e^{-\imath\frac{\theta_2}{2}Z}$ on one qubit, respectively. It is worth noting that the parameterized (Z-X-Z)-rotation induces universal gates on each qubit [25, Theorem 4.1], and thus the expressiveness of the models on one qubit is ensured. In the entangling block, XX stands for the parameterized $(X \otimes X)$-rotation $e^{-\imath\frac{\theta_3}{2}X\otimes X}$ on two qubits. The entangling block can create entanglement between each qubit. Here entanglement is a unique feature of quantum models to express the interactions of qubits. The model is constructed by alternately using these two blocks with a quantum measurement M at the end of the model.

Since TensorFlow Quantum is inefficient in training noisy models, we only use 3 rotation blocks and 2 entangling blocks in the training models. In addition, to

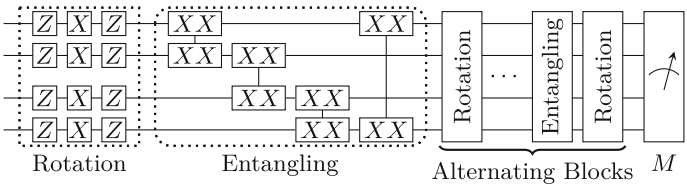

Fig. 2. Parameterized Quantum Circuits for Quantum Finance Decision Models.

Table 1. Experimental results of Lipschitz constant K^* of the trained models.

Dataset	Noise		Accuracy		K^*	Time
	type	probability	train	test		
German Credit	None		0.732	0.686	1.0000×10^{0}	\
	Phase flip	10^{-4}	0.726	0.692	9.9997×10^{-1}	2.36s
		10^{-3}	0.724	0.714	9.9800×10^{-1}	2.02s
		10^{-2}	0.704	0.708	9.6918×10^{-1}	1.94s
	Depolarizing	10^{-4}	0.709	0.686	9.9977×10^{-1}	2.77s
		10^{-3}	0.701	0.712	9.9789×10^{-1}	2.93s
		10^{-2}	0.709	0.682	9.7916×10^{-1}	3.44s
	Bit flip	10^{-4}	0.712	0.728	9.9975×10^{-1}	2.27s
		10^{-3}	0.710	0.690	9.9743×10^{-1}	2.47s
		10^{-2}	0.724	0.678	9.7981×10^{-1}	2.05s
	Mixed noise	10^{-4}	0.710	0.704	9.9980×10^{-1}	2.15s
		10^{-3}	0.731	0.682	9.9834×10^{-1}	2.08s
		10^{-2}	0.731	0.692	9.7021×10^{-1}	1.95s
Adult Income (DiCE)	None		0.777	0.770	1.0000×10^{0}	\
	Phase flip	10^{-4}	0.784	0.767	9.9992×10^{-1}	0.44s
		10^{-3}	0.771	0.770	9.9805×10^{-1}	0.51s
		10^{-2}	0.773	0.767	9.8057×10^{-1}	0.48s
	Depolarizing	10^{-4}	0.774	0.767	9.9987×10^{-1}	0.57s
		10^{-3}	0.781	0.767	9.9867×10^{-1}	0.58s
		10^{-2}	0.779	0.767	9.8667×10^{-1}	0.69s
	Bit flip	10^{-4}	0.780	0.767	9.9980×10^{-1}	0.57s
		10^{-3}	0.777	0.767	9.9800×10^{-1}	0.49s
		10^{-2}	0.778	0.770	9.8117×10^{-1}	0.54s
	Mixed noise	10^{-4}	0.762	0.720	9.9987×10^{-1}	0.68s
		10^{-3}	0.752	0.720	9.9812×10^{-1}	0.67s
		10^{-2}	0.759	0.720	9.7647×10^{-1}	0.67s

simulate noisy models, we put different quantum noises introduced in Example 2 on each qubit, including bit flip, phase flip, depolarizing, and the mixtures of them, behind the first rotation block. Note that the number of qubits for the models is the same as the number of features of datasets due to the above choice of the data-encoding feature map. The final measurement $M = \{M_0 = I \otimes |0\rangle\langle 0|, M_1 = I \otimes |1\rangle\langle 1|\}$ is a local measurement performed on the last qubit. With the binary classification task, the loss function we choose is binary cross-entropy: $-\frac{1}{N}\sum_{j=1}^{N} c_j \cdot \log \bar{c}_j + (1 - c_j)\log(1 - \bar{c}_j))$, where N is the size of the batch fixed in the training process, c_j is the true label and $\bar{c}_j$ is the outcome of the measurement. All models are well trained and achieve around 70% train and test accuracy (see Column "Accuracy" in Table 1), matching that of the previously used classical and quantum finance decision models (e.g. [10,21]).

Evaluation Details and Results: The results of evaluating Algorithm 1 on the models trained from different datasets and different quantum noises are presented in Table 1. For different datasets, we train noise-free models to serve as the baseline for training and test accuracy (see Row "None"). Furthermore, different types of noise are added with different levels of probabilities. We list the Lipschitz constant K^* and the running time of Algorithm 1 aided by `NumPy` for each column. It can be seen that the higher level of noise's probability, the smaller value of constant K^*. Therefore, the claim of quantum noise improving fairness in Sect. 4.2 is confirmed by the numerical results. This is also observed in Table 2 later.

6.2 Scalability in the NISQ Era

Models: To reflect an actual application in the NISQ era, we choose not to randomly generate a parameterized quantum circuit model. Instead, we expanded the existing example of Quantum Convolutional Neural Network (QCNN) [32] in the QCNN tutorial[6] of TensorFlow Quantum from 8 qubits (see Fig. 3) to 27 qubits. In the experiment, we use the QCNN model with one convolution layer and one pooling layer. The noise is applied between convolution and pooling layers on each qubit. The final measurement is $M = \{M_0 = I \otimes |0\rangle\langle 0|, M_1 = I \otimes |1\rangle\langle 1|\}$ performed on the last qubit with a gate U appended before. Since training a noisy model of this size is currently intractable on TensorFlow Quantum, the parameters in the model are all randomly sampled.

[6] https://tensorflow.google.cn/quantum/tutorials/qcnn.

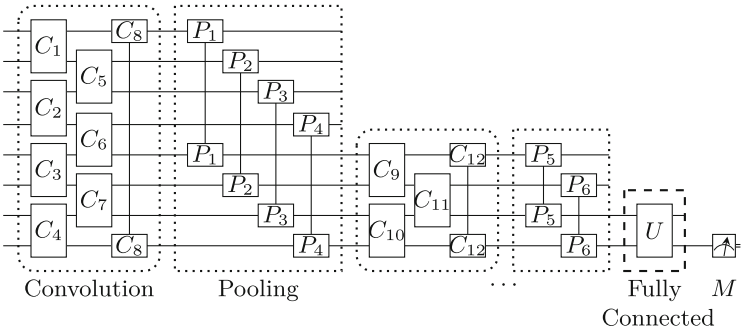

Fig. 3. The QCNN model in the tutorial of TensorFlow Quantum. Each C_i in the convolution layer is a parameterized 2-qubit gate to find a new state between adjacent qubits. Each P_i in the pooling layer is also a parameterized 2-qubit gate with another form that attempts to extract the information of two qubits into a single qubit.

Evaluation Details and Results: We choose the models with 25 and 27 qubits to run experiments. Since the parameters are randomly sampled, for each noise with different levels of probability, we generate the model and evaluate the Lipschitz constant K^* for 3 times. However, because a $2^{25} \times 2^{25}$ or $2^{27} \times 2^{27}$ complex matrix consumes a huge amount of memory, it is not feasible to directly use Algorithm 1 as the previous experiment, where we represent the M_A in Algorithm 1 as a matrix and use the package `NumPy` to evaluate eigenvalue. We instead use a tensor network [48] to represent the M_A and the subroutine of evaluating eigenvalue in Algorithm 1 is implemented with the basic power method for eigenvalue problem [49] by using `TensorNetwork` package. Although there are some packages for sparse matrix in Python that can collaborate with `TensorNetwork`, their implementation for computing eigenvalues still consumes a huge amount of memory. The evaluation results on QCNN models with randomly sampled parameters and different quantum noises are listed in Table 2. These results prove that our fairness verification algorithm is efficient and can handle 27-qubit quantum decision models on a small server. For further exploring the scalability of our verification algorithm, we also test on 29-qubit QCNN models; Please see Appendix D in [37] for the results.

Last but not least, it is worth noting that in all experiments, we also obtain bias kernels by Algorithm 1 at the running time presented in Tables 1 and 2, but as they are large-size (up to 2^{27}-dimensional) vectors, we do not show them.

Table 2. Experimental results of Lipschitz constant K^* of QCNN models.

#Qubits	Noise		Evaluation I		Evaluation II		Evaluation III	
	type	probability	K^*	Time	K^*	Time	K^*	Time
25	None		1.0000	\	1.0000	\	1.0000	\
	Phase flip	10^{-4}	0.9998	2.15m	0.9997	1.92m	0.9999	2.12m
		10^{-3}	0.9983	1.71m	0.9982	1.35m	0.9987	1.10m
		10^{-2}	0.9865	1.75h	0.9870	54.49m	0.9831	39.07m
	Depolarizing	10^{-4}	0.9998	2.22m	0.9998	1.59m	0.9998	2.38m
		10^{-3}	0.9985	2.46m	0.9980	1.62m	0.9982	2.04m
		10^{-2}	0.9824	2.33m	0.9802	2.53m	0.9809	1.77m
	Bit flip	10^{-4}	0.9997	1.74m	0.9998	1.60m	0.9999	2.15m
		10^{-3}	0.9986	2.44m	0.9980	1.80m	0.9991	2.37m
		10^{-2}	0.9943	1.78h	0.9854	20.78m	0.9919	49.36m
	Mixed noise	10^{-4}	0.9998	3.68m	0.9998	1.34m	0.9998	1.94m
		10^{-3}	0.9980	1.66m	0.9966	2.06m	0.9983	0.96m
		10^{-2}	0.9901	37.24m	0.9861	1.95h	0.9759	6.03m
27	None		1.0000	\	1.0000	\	1.0000	\
	Phase flip	10^{-4}	0.9999	6.75m	0.9998	7.34m	0.9998	8.62m
		10^{-3}	0.9980	6.66m	0.9977	9.55m	0.9981	6.56m
		10^{-2}	0.9896	7.64m	0.9839	54.12m	0.9709	4.45m
	Depolarizing	10^{-4}	0.9998	6.10m	0.9998	6.89m	0.9998	6.77m
		10^{-3}	0.9981	4.51m	0.9985	5.34m	0.9978	21.75m
		10^{-2}	0.9809	1.20h	0.9767	6.48m	0.9773	8.48m
	Bit flip	10^{-4}	0.9998	6.52m	0.9999	5.39m	0.9999	6.86m
		10^{-3}	0.9986	4.38m	0.9984	7.96m	0.9971	10.37m
		10^{-2}	0.9917	5.03h	0.9894	4.15h	0.9854	3.90h
	Mixed noise	10^{-4}	0.9998	6.67m	0.9998	5.19m	0.9997	10.39m
		10^{-3}	0.9976	7.06m	0.9976	5.91m	0.9986	6.62m
		10^{-2}	0.9806	7.70m	0.9850	7.98m	0.9881	6.02h

7 Conclusion

In this work, we initiate the studies on algorithmic verification of fairness of quantum machine learning decision models. In particular, we showed that this verification problem can be reduced to computing the Lipschitz constant of the decision models, and then resolved the latter by introducing and estimating single measurement distinguishability. Based on these theoretical results, we developed an algorithm that can verify the (ε, δ)-fairness of quantum decision models and provides useful bias kernels for explaining the unfairness of the models.

An interesting topic for future research is how to improve the results presented in this paper for training quantum decision models with fairness guarantee. On the other hand, further investigations are required to better understand the bias kernels detected by our verification algorithm, especially through more experiments on real-world applications.

Acknowledgments. Ji Guan would like to thank Jiayi Chen for her linguistic assistance during the preparation of this paper. This work was partly supported by the National Key R&D Program of China (Grant No: 2018YFA0306701), the National Natural Science Foundation of China (Grant No: 61832015).

References

1. Arute, F., et al.: Quantum supremacy using a programmable superconducting processor. Nature **574**(7779), 505–510 (2019)
2. Zhong, H.-S., Wang, H., Deng, Y.-H., Chen, M.-C., Peng, L.-C., Luo, Y.-H., Qin, J., Dian, W., Ding, X., Yi, H., et al.: Quantum computational advantage using photons. Science **370**(6523), 1460–1463 (2020)
3. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature **549**(7671), 195–202 (2017)
4. Google. Tensorflow Quantum. https://www.tensorflow.org/quantum, (Accessed 2021)
5. Dixon, M.F., Halperin, I., Bilokon, P.: Machine Learning in Finance. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-41068-1
6. Lloyd, S., Mohseni, M., Rebentrost, P.: Quantum principal component analysis. Nature Phys. **10**(9), 631–633 (2014)
7. Rebentrost, P., Schuld, M., Wossnig, L., Petruccione, F., Lloyd, S.: Quantum gradient descent and Newton's method for constrained polynomial optimization. New J. Phys. **21**(7), 073023 (2019)
8. Liu, N., Rebentrost, P.: Quantum machine learning for quantum anomaly detection. Phys. Rev. A **97**(4), 042315 (2018)
9. Di Pierro, A., Incudini, M.: Quantum machine learning and fraud detection. In: Dougherty, D., Meseguer, J., Mödersheim, S.A., Rowe, P. (eds.) Protocols, Strands, and Logic. LNCS, vol. 13066, pp. 139–155. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91631-2_8
10. Garc´ıa, R., Cahue, J., Pavas, S.: Credit risk scoring with a supervised quantum classifier, May 2020
11. Milne, A., Rounds, M., Goddard, P.: Optimal feature selection in credit scoring and classification using a quantum annealer. White Paper 1Qbit (2017)
12. Kerenidis, I., Prakash, A.: Quantum recommendation systems (2016). arXiv preprint arXiv:1603.08675
13. Egger, D.J., et al.: Quantum computing for finance: state-of-the-art and future prospects. IEEE Trans. Quant. Eng. **1**, 1–24 (2020)
14. Orus, R., Mugel, S., Lizaso, E.: Quantum computing for finance: overview and prospects. Rev. Phys. **4**, 100028 (2019)
15. Flores, A.W., Bechtel, K., Lowenkamp, C.T.: False positives, false negatives, and false analyses: A rejoinder to machine bias: there's software used across the country to predict future criminals. and it's biased against blacks. Fed. Probation **80**, 38 (2016)

16. Calders, T., Kamiran, F., Pechenizkiy, M.: Building classifiers with independency constraints. In: 2009 IEEE International Conference on Data Mining Workshops, pp. 13–18. IEEE (2009)
17. Pedreshi, D., Ruggieri, S., Turini, F.: Discrimination-aware data mining. In: Proceedings of the 14th ACM Sigkdd International Conference On Knowledge Discovery and Data Mining, pp. 560–568 (2008)
18. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, pp. 214–226 (2012)
19. Barocas, S., Hardt, M., Narayanan, A.: Fairness and Machine Learning. fairmlbook.org, (2019). http://www.fairmlbook.org
20. Binns, R.: On the apparent conflict between individual and group fairness. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 514–524 (2020)
21. John, P.G., Vijaykeerthy, D., Saha, D.: Verifying individual fairness in machine learning models. In: Conference on Uncertainty in Artificial Intelligence, PMLR, pp. 749–758 (2020)
22. Albarghouthi, A., D'Antoni, L., Drews, S., Nori, A.V.: Fairsquare: probabilistic verification of program fairness. In: Proceedings of the ACM on Programming Languages (OOPSLA), vol. 1, pp. 1–30 (2017)
23. Bastani, O., Zhang, X., Solar-Lezama, A.: Probabilistic verification of fairness properties via concentration. In: Proceedings of the ACM on Programming Languages (OOPSLA), vol. 3, pp. 1–27 (2019)
24. Ghosh, B., Basu, D., Meel, K.S.: Justicia: a stochastic SAT approach to formally verify fairness. In: AAAI, pp. 7554–7563. AAAI Press (2021)
25. Nielsen, M.A., Chuang, I.L.: Quantum computation and quantum information. Cambridge University Press (2010)
26. Guan, J., Fang, W., Ying, M.: Robustness verification of quantum classifiers. In: Silva, A., Leino, K.R.M. (eds.) CAV 2021. LNCS, vol. 12759, pp. 151–174. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-81685-8_7
27. Liu, N., Wittek, P.: Vulnerability of quantum classification to adversarial perturbations. Phys. Rev. A **101**(6), 062331 (2020)
28. Weber, M., Liu, N., Li, B., Zhang, C., Zhao, Z.: Optimal provable robustness of quantum classification via quantum hypothesis testing. NPJ Quant. Inf. **7**(1), 1–12 (2021)
29. Biamonte, J., Bergholm, V.: Tensor networks in a nutshell (2017). arXiv preprint arXiv:1708.00006
30. Cerezo, M., et al.: Variational quantum algorithms. Nature Rev. Phys. **3**(9), 625–644 (2021)
31. Beer, K.: Training deep quantum neural networks. Nature Commun. **11**(1), 1–6 (2020)
32. Cong, I., Choi, S., Lukin, M.D.: Quantum convolutional neural networks. Nature Phys. **15**(12), 1273–1278 (2019)
33. Reddy, P., Bhattacherjee, A.B.: A hybrid quantum regression model for the prediction of molecular atomization energies. Mach. Learn. Sci. Technol. **2**(2), 025019 (2021)
34. IBM. Learn quantum computation using Qiskit. https://qiskit.org/textbook/preface.html, (Accessed 2021)
35. Fazlyab, M., Robey, A., Hassani, H., Morari, M., Pappas, G.: Efficient and accurate estimation of Lipschitz constants for deep neural networks. In: Advances in Neural Information Processing Systems, vol. 32 (2019)

36. Szegedy, C., et al.: Intriguing properties of neural networks. In: Bengio, Y., LeCun, Y., (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, pp. 14–16 April 2014. Conference Track Proceedings (2014)
37. Supplemental Material: Verifying Fairness in Quantum Machine Learning. https://doi.org/10.5281/zenodo.6612720
38. Yuxuan, D., Hsieh, M.-H., Liu, T., Tao, D., Liu, N.: Quantum noise protects quantum classifiers against adversaries. Phys. Rev. Res. **3**(2), 023153 (2021)
39. Wolf, M.M.: Quantum channels & operations: Guided tour (2012). https://www-m5.ma.tum.de/foswiki/pub/M5/Allgemeines/MichaelWolf/QChannelLecture.pdf
40. Baier, C., Katoen, J.-P.: Principles of model checking. MIT Press (2008)
41. Akers, S.B.: Binary decision diagrams. IEEE Trans. Comput. **27**(06), 509–516 (1978)
42. UCI Machine Learning Repository. Statlog (german credit data) data set. https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29/, (Accessed 2021)
43. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, January 2020
44. Roberts, C., et al.: Tensornetwork: a library for physics and machine learning (2019). https://tensornetwork.readthedocs.io/en/latest/index.html
45. Kohavi, R., Becker. B.: Uci machine learning repository, adult data set. https://archive.ics.uci.edu/ml/datasets/adult, (Accessed 2021)
46. The Pennsylvania State University. Analysis of german credit data. https://online.stat.psu.edu/stat508/book/export/html/796, (Accessed 2021)
47. Zhu, D., et al.: Training of quantum circuits on a hybrid quantum computer. Sci. Adv. **5**(10), eaaw9918 (2019)
48. Bridgeman, J.C., Chubb, C.T.: Hand-waving and interpretive dance: an introductory course on tensor networks. J. Phys. A: Math. Theor. **50**(22), 223001 (2017)
49. Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., van der Vorst, H.: Templates for the Solution of Algebraic Eigenvalue Problems. Society for Industrial and Applied Mathematics (2000)