



OPEN

# Quantum discriminator for binary classification

Prasanna Date<sup>1✉</sup> & Wyatt Smith<sup>2</sup>

Quantum computers have the unique ability to operate relatively quickly in high-dimensional spaces—this is sought to give them a competitive advantage over classical computers. In this work, we propose a novel quantum machine learning model called the Quantum Discriminator, which leverages the ability of quantum computers to operate in the high-dimensional spaces. The quantum discriminator is trained using a quantum-classical hybrid algorithm in  $\mathcal{O}(N \log N)$  time, and inferencing is performed on a universal quantum computer in  $\mathcal{O}(N)$  time. The quantum discriminator takes as input the binary features extracted from a given datum along with a prediction qubit, and outputs the predicted label. We analyze its performance on the Iris and Bars and Stripes data sets, and show that it can attain 99% accuracy in simulation.

Machine learning has become ubiquitous in almost every discipline under the sun<sup>1</sup>. While high-quality training data will only continue to increase in availability in the coming decades, it is projected that classical approaches to machine learning will fail to keep pace with this increase owing to the end of Moore's Law<sup>2,3</sup>. Consequently, we must look towards alternative computing paradigms such as quantum and neuromorphic computing to address these scalability issues and develop more efficient machine learning methods<sup>4–6</sup>.

Quantum computing has the potential to significantly speed up machine learning tasks<sup>4,7</sup>. Quantum computers use the quantum phenomena of superposition, tunneling and entanglement to perform computations<sup>8</sup>. As a result, they are able to operate in high-dimensional tensor-product spaces much faster than classical computers<sup>9,10</sup>. For certain applications such as integer factoring and searching, quantum computers are known to outperform classical computers<sup>11,12</sup>. We believe the ability of quantum computers to efficiently operate in these high-dimensional tensor product spaces can be leveraged to design efficient training and inferencing methods.

In this work, we focus on binary classification. We operate within the traditional two-step workflow in machine learning, where the first step is to extract features from the data, and the second step is to perform classification using a discriminant function. We further assume that binary features, which are crucial in many scientific and engineering domains for encoding categorical variables, for reducing the memory usage in embedded system applications, and for reducing the computation time in real-time edge computing applications, have been extracted from the data; many approaches for extracting binary features already exist in the literature<sup>13,14</sup>. Therefore, our focus in this paper is to propose the Quantum Discriminator, which is a quantum discriminant model that performs binary classification on a set of binary features. We also present the hybrid quantum-classical training algorithm used to train the quantum discriminator in  $\mathcal{O}(N \log N)$  time.

As a proof of concept, we demonstrate that our model can be used to completely solve the 2-bit binary classification problem. We also benchmark the quantum discriminator on the Iris and the Bars and Stripes data sets. Our results demonstrate that under a proper feature extraction and training regime, the quantum discriminator can attain a near-perfect (99%) accuracy in simulation on the these data sets.

Sections "Related work" and "Notation" cover the related work and notation used in this paper. The quantum discriminator, associated quantum-classical training algorithm, theoretical analysis and notes on generalizability are presented in Section "The quantum discriminator". In Section "2-bit binary classification", we apply the quantum discriminator to the 2-bit binary classification problem as a proof of concept. In Section "Methods and empirical evaluation", we benchmark the performance of our model on the Iris and Bars and Stripes data sets.

## Related work

Several machine learning approaches on universal quantum computers have been proposed in the literature<sup>4</sup>. Lloyd and Weedbrook as well as Dallaire-Demers and Killoran derive the theoretical underpinnings of quantum generative adversarial networks<sup>15,16</sup>. Blance and Spannowsky propose a variational quantum classifier for use in high energy physics applications<sup>17</sup>. Shingu et al. propose a variational quantum algorithm for Boltzmann machine learning<sup>18</sup>. Benedetti et al. propose quantum parameterized circuits as machine learning models. Quiroga et al.

<sup>1</sup>Oak Ridge National Laboratory, Oak Ridge, Tennessee 37830, USA. <sup>2</sup>University of Tennessee, Knoxville, Tennessee 37996, USA. ✉email: datepa@ornl.gov

propose a quantum k-means (QK-Means) clustering technique to discriminate quantum states on the IBM Bogota quantum device<sup>19</sup>. Apart from universal quantum computing approaches, adiabatic quantum machine learning approaches have also been proposed for traditional machine learning models such as regression and k-means clustering<sup>20,21</sup>.

The above literature describes quantum implementations of conventional machine learning models. Another line of research in quantum machine learning focuses on developing purely quantum or hybrid quantum-classical models that are novel and different from conventional machine learning models. Gambs recasts the quantum discrimination problem within the framework of machine learning and uses the notion of learning reduction to solve different variants of the quantum classification task<sup>22</sup>. Sentis et al. present a quantum learning machine for binary classification of qubit states that does not require quantum memory and produces classifiers that are robust to an arbitrary amount of noise<sup>23</sup>. Chen et al. propose a discrimination method for two similar quantum systems and apply it to quantum ensemble classification<sup>24</sup>. Sergioli et al. propose a new quantum classifier called the Helstrom Quantum Centroid, which acts on density matrices that encode the classical patterns of a data set onto the quantum computer<sup>25</sup>. Park et al. focus on the squared overlap between quantum states as a similarity measure and examine the essential ingredients for the quantum binary classification, advancing the theory of quantum kernel-based binary classification<sup>26</sup>. In order to reduce the number of trainable parameters of a quantum circuit, Li et al. propose the Variational Shadow Quantum Learning (VSQL) framework<sup>27</sup>. Blank et al. present a distance-based quantum classifier whose kernel is based on the quantum state fidelity between training and test data<sup>28</sup>. They also conduct proof of principle experiments on the IBMQ platform. Silver et al. develop a framework for multi-class classification on NISQ devices and test its performance on MNIST, Fashion MNIST and CIFAR data sets<sup>29</sup>.

A number of approaches to quantum machine learning leverage the two-step workflow followed by traditional machine learning models. Havlicek et al. propose a quantum variational classifier and a quantum kernel estimator for classification problems<sup>30</sup>. Schuld and Killoran propose a nonlinear feature map that maps data to a quantum feature space and discuss two discriminant models for classification<sup>31</sup>. Bergou and Hillery propose a quantum discriminator that can distinguish between two unknown quantum states<sup>32</sup>. Lloyd et al. present a two-part quantum machine learning model, where the first part of the circuit implements a quantum feature map that encodes classical inputs into quantum states, and the second part of the circuit executes a quantum measurement, which acts as the output of the model<sup>33</sup>.

Both universal as well as adiabatic approaches for quantum machine learning have been proposed in the literature. However, there are several limitations in the current state-of-the-art. Most of the proposed approaches are based on variational quantum circuits. By definition, these are iterative quantum-classical hybrid approaches, and require multiple data exchanges between the quantum and the classical computer. These data exchanges necessitate frequent measurements of the quantum states, which introduce measurement errors as well as collapse the quantum superposition. When the superposition is collapsed frequently, the ability of quantum computers to operate efficiently in high-dimensional tensor product spaces is curtailed. So, it is desirable to minimize the use of classical computers in quantum machine learning methods.

The quantum discriminator proposed in this paper mitigates some of these challenges. Firstly, unlike some of the QML models that use classical computers during training and inferencing stages, the quantum discriminator uses classical computers to process  $\mathcal{O}(N)$  data during training only. The classical computer is not used in the inferencing stage. This restricted use of the classical computer enables the quantum discriminator to operate in the high-dimensional tensor-product spaces efficiently. Next, we validate the quantum discriminator on the Iris and the Bars and Stripes data sets. Another advantage of the quantum discriminator is that it can be trained in  $\mathcal{O}(N \log N)$  time using  $\mathcal{O}(N \log N)$  classical bits and  $\mathcal{O}(\log N)$  qubits. Inferencing can be performed on the quantum discriminator in  $\mathcal{O}(N)$  time using  $\mathcal{O}(\log N)$  qubits.

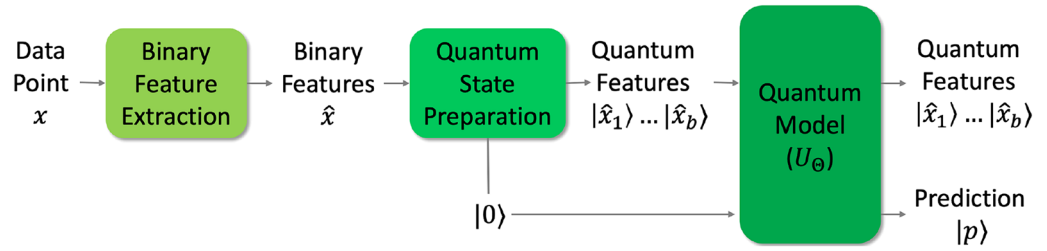
## Notation

We use the following notation throughout this paper:

- $\mathbb{R}, \mathbb{N}, \mathbb{B}$ : Set of real numbers, natural numbers and binary numbers ( $\mathbb{B} = \{0, 1\}$ ) respectively.
- $N$ : Number of data points in training data set ( $N \in \mathbb{N}$ ).
- $d$ : Dimension of each data point in the training data set ( $d \in \mathbb{N}$ ).
- $b$ : Dimension of each data point in the binary feature set of the training data set ( $b \in \mathbb{N}$ ).
- $B$ : Number of unique states that can be attained using  $b$  bits ( $B = 2^b$ ).
- $X$ : The training data set ( $X \in \mathbb{R}^{N \times d}$ ).
- $Y$ : The training labels ( $Y \in \mathbb{B}^N$ ). If the  $i^{\text{th}}$  data point belongs to class 0 (class 1), then  $y_i = 0$  ( $y_i = 1$ ).
- $\hat{X}$ : The binary feature set of the training data set  $X$  ( $\hat{X} \in \mathbb{B}^{N \times b}$ ).  $\hat{x}_i \in \hat{X}$  contains the features corresponding to the  $i^{\text{th}}$  data point  $x_i \in X$ .
- $P$ : The labels predicted by the quantum binary classification model ( $P \in \mathbb{B}^N$ ). Ideally, the predicted labels should be identical to the training labels ( $Y$ ).

## The quantum discriminator

Given a data point  $x$  which belongs to one of two classes (Class 0 or Class 1), we would like to predict the correct class for  $x$ . Generally, the binary classification model is characterized by a set of model parameters  $\Theta$ . The workflow used for traditional machine learning classification models—such as support vector machines (SVM) and logistic regression—is comprised of 2-steps: (i) Feature extraction from the data; and (ii) Class determination by application of a discriminant function. The workflow governing our quantum model for binary classification is analogous to this 2-step workflow as shown in Fig. 1.



**Figure 1.** Model workflow.

In Fig. 1, we are given a data point  $x \in \mathbb{R}^d$  whose class needs to be predicted. We first extract the binary features of  $x$ , denoted by  $\hat{x} \in \mathbb{B}^b$ . Usually, the extracted features are domain specific, for example, Histogram of Oriented Gradients (HOG)<sup>34</sup> and Scale-Invariant Feature Transform (SIFT)<sup>35</sup> have been widely used in computer vision. The feature space could also originate from dimensionality reduction techniques like Principal Component Analysis (PCA). So, we do not make any assumptions on the extracted features, except that they are binary. This feature set of the training data is denoted by  $\hat{X}$  in Fig. 1. Each point  $\hat{x} \in \hat{X}$  is a point on a  $b$ -dimensional unit hypercube, and there are  $B$  such points. Given  $\hat{x}$ , its equivalent quantum feature state is denoted by  $|\hat{x}\rangle$ , which is a point in  $b$ -dimensional Hilbert space.

In addition to preparing the quantum feature state in Fig. 1, we also prepare a qubit in the  $|0\rangle$  state, which would serve as our prediction qubit  $|p\rangle$ . The quantum feature state  $|\hat{x}\rangle = |\hat{x}_1 \dots \hat{x}_b\rangle$ , as well as the prediction qubit in the  $|0\rangle$  state serve as inputs to the quantum discriminator as shown in Fig. 2. The quantum discriminator ( $U_\Theta$ ) is a  $2B \times 2B$  matrix, which takes as input  $|\hat{x}\rangle$  and  $|0\rangle$ , and outputs  $|\hat{x}\rangle$  and the prediction  $|p\rangle$ . We now describe  $U_\Theta$ , which is parameterized by  $\Theta = \{\theta_1, \theta_2, \dots, \theta_B\}$ ,  $\theta_i \in \mathbb{B}$ ,  $\forall i = 1, 2, \dots, B$ .

$$U_\Theta = \begin{bmatrix} 1 - \theta_1 & \theta_1 & 0 & 0 & \dots & 0 & 0 \\ \theta_1 & 1 - \theta_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 - \theta_2 & \theta_2 & \dots & 0 & 0 \\ 0 & 0 & \theta_2 & 1 - \theta_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 - \theta_B & \theta_B \\ 0 & 0 & 0 & 0 & \dots & \theta_B & 1 - \theta_B \end{bmatrix}. \tag{1}$$

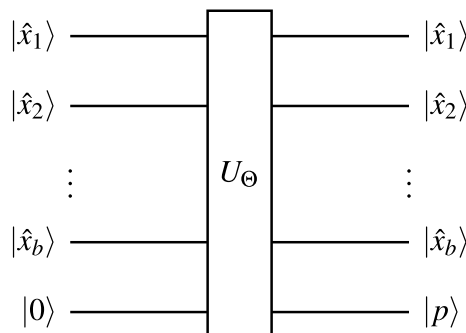
Since all quantum operators are unitary matrices, it is crucial that  $U_\Theta$  be a unitary matrix. We now show that  $U_\Theta$  is unitary by showing that  $U_\Theta^\dagger U_\Theta = U_\Theta U_\Theta^\dagger = I$ . Since  $U_\Theta$  is symmetric,  $U_\Theta^\dagger = U_\Theta$ . Because  $\theta_i \in \mathbb{B}$ , the off-diagonal elements in  $U_\Theta^\dagger U_\Theta$  and  $U_\Theta U_\Theta^\dagger$  are zeros. The diagonal elements of  $U_\Theta^\dagger U_\Theta$  and  $U_\Theta U_\Theta^\dagger$  are of the form  $(1 - \theta_i)^2 + \theta_i^2$ , which always equals unity. So,  $U_\Theta^\dagger U_\Theta = U_\Theta U_\Theta^\dagger = I$ . Thus,  $U_\Theta$  is a unitary.

**Number of binary features**

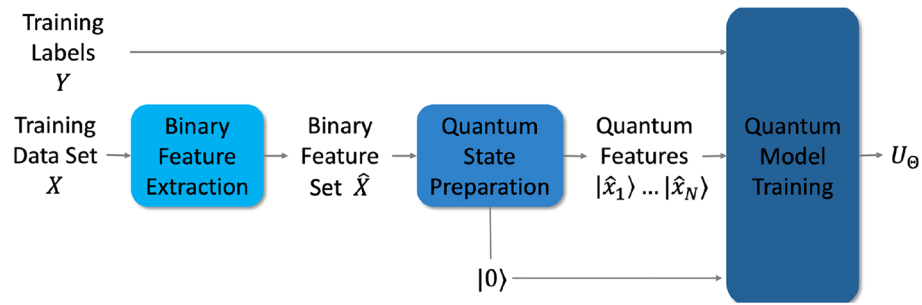
We would like to extract  $b$  binary features so that the  $N$  points in the feature set span as much of the feature space as possible. This would ensure that the quantum discriminator trained as a result would be generalizable to any test data point that originates from the same distribution as the training data set. Since the size of the feature space is  $2^b$ , we have:  $N \approx 2^b$ , or  $b$  is  $\mathcal{O}(\log N)$ .

**Training the quantum discriminator**

Figure 3 shows the training workflow for our quantum model for classification. We are given the training data set  $X$  and the training labels  $Y$ . We compute the binary feature set  $\hat{X}$  and prepare the quantum feature states  $|\hat{x}_1\rangle \dots |\hat{x}_N\rangle$ , where  $\hat{x}_i \in \hat{X}$ ,  $\forall i = 1, 2, \dots, N$ . In addition to the quantum feature states, we also prepare the  $|0\rangle$



**Figure 2.** The quantum discriminator.



**Figure 3.** Training workflow.

state. Given  $|\hat{x}_1\rangle \dots |\hat{x}_N\rangle$ ,  $Y$ , and  $|0\rangle$ , the goal of training the quantum discriminator is to find the model parameters  $\Theta$ , that minimize a well-defined error function.

Some examples of error functions used in classification tasks are the Euclidean error (or  $L2$ -norm) in linear regression, negative log likelihood error in logistic regression, and the cross entropy error in deep neural networks. Negative log likelihood and cross entropy errors are generally used in probabilistic machine learning models, whereas,  $L2$ -norm is used for non-probabilistic machine learning models. For non-probabilistic machine learning models, the  $L1$ -norm is considered to be more robust than the  $L2$ -norm<sup>36</sup>. However,  $L2$ -norm is generally used in practice because it is differentiable and amenable to gradient computations. Since our quantum discriminator is not probabilistic and does not require gradient computations, we will use the  $L1$ -norm as the error function. Thus, the training process can be stated as follows:

$$\min_{\Theta} E(\Theta) = \frac{1}{N} \sum_{i=1}^N |p_i - y_i|, \tag{2}$$

where  $p_i$  is the measured value of  $|p_i\rangle$  and the output state,  $|\hat{x}_i p_i\rangle$  equals  $U_{\Theta}|\hat{x}_i 0\rangle \forall i = 1, 2, \dots, N$ .

```

Input:  $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N, y_1, y_2, \dots, y_N$ 
 $b \leftarrow \text{length}(\hat{x}_1)$ 
 $B \leftarrow 2^b$ 
 $\tau \leftarrow [2^{b-1}, 2^{b-2}, \dots, 2^1, 2^0]^T$ 
 $\theta_j \leftarrow 0 \quad \forall j = 1, 2, \dots, B$ 
for  $i = 1, 2, \dots, N$ 
  if  $y_i == 1$ 
     $j \leftarrow 1 + \tau \cdot \hat{x}_i$ 
     $\theta_j \leftarrow 1$ 

```

**Algorithm 1.** Training the quantum discriminator.

Algorithm 1 outlines the training process for the quantum discriminator. The inputs to the model are the feature vectors  $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N$ , and the training labels  $y_1, y_2, \dots, y_N$ . We first initialize  $b$  and  $B$ . The  $\text{length}(z)$  function computes the length of  $z$ . We then set the vector  $\tau = [2^{b-1}, 2^{b-2}, \dots, 2^0]$ . Next, we setup the quantum circuit shown in Fig. 2, where  $U_{\Theta} = I$ , by initializing all the model parameters  $\theta_j$  to zero ( $j = 1, 2, \dots, B$ ). We then look at each feature vector  $\hat{x}_i$  ( $i = 1, 2, \dots, N$ ). If  $\hat{x}_i$  belongs to Class 1 (i.e.  $y_i = 1$ ), then we compute the index  $j$  as  $1 + \tau \cdot \hat{x}_i$ , and set  $\theta_j = 1$ . We repeat this process for all  $N$  points in the training feature set  $\hat{X}$ . When Algorithm 1 terminates, it assigns all points in  $\hat{X}$  to their respective correct classes.

We now shed some light on why Algorithm 1 works. The input state to the quantum discriminator,  $|\hat{x} 0\rangle$ , exists in  $(b + 1)$ -dimensional Hilbert space and is in a superposition of all  $2B$  possible states. As such, each of the  $B$  quantum feature states  $|\hat{x}\rangle$ , occurs twice: as  $|\hat{x} 0\rangle$  and  $|\hat{x} 1\rangle$ . These two states can be interpreted as  $|\hat{x}\rangle$  belonging to Class 0 or Class 1 respectively. While training the quantum discriminator, we select the correct class for each  $|\hat{x}\rangle$ . The rows and columns of  $U_{\Theta}$  that correspond to  $\hat{x}$  can be found at indices  $j = 1 + \tau \cdot \hat{x}$  and  $j + 1$ , which can be leveraged to assign  $\hat{x}$  to Class 0 or Class 1 respectively. Initially, the  $2 \times 2$  sub-matrix at  $j^{\text{th}}$  row and  $j^{\text{th}}$  column of  $U_{\Theta}$  is an identity matrix because  $\theta_j$  is initialized to 0. If  $\hat{x}$  belongs to Class 0, then this sub-matrix outputs  $|\hat{x} 0\rangle$  for the input  $|\hat{x} 0\rangle$ , which can be interpreted as  $\hat{x}$  being assigned to Class 0. On the other hand, if  $\hat{x}$  belongs to Class 1, then this sub-matrix must be changed to the Pauli- $X$  gate (also called the bit-flip gate or the NOT gate), which is done by setting  $\theta_j$  to 1. The Pauli- $X$  gate at  $j^{\text{th}}$  row and  $j^{\text{th}}$  column of  $U_{\Theta}$  outputs  $|\hat{x} 1\rangle$  for the input  $|\hat{x} 0\rangle$ , which can be interpreted as  $\hat{x}$  being assigned to Class 1.

**Theoretical analysis**

We analyze the time and space complexity of training the quantum discriminator here. In Algorithm 1, lines 1 and 2 require  $\mathcal{O}(1)$  time and line 3 requires  $\mathcal{O}(b)$  time. It may seem that line 4 requires  $\mathcal{O}(B)$  time, but initializing  $\theta_j$  to 0 essentially refers to setting up the quantum circuit with  $U_{\Theta} = I$ . This entails setting up  $b + 1$  qubits, which takes

$\mathcal{O}(b)$  time. Computing the dot product on line 7 may take  $\mathcal{O}(b)$  time and setting  $\theta_j$  to unity on line 8 takes  $\mathcal{O}(1)$  time. Since we may repeat lines 7 and 8  $N$ -times in the worst case, the time complexity of Algorithm 1 is  $\mathcal{O}(Nb)$ , which is the same as the size of the feature set  $\hat{X}$ . Since  $b$  is  $\mathcal{O}(\log N)$  from Section "Number of binary features", the time complexity is  $\mathcal{O}(N \log N)$ . Since we use  $\mathcal{O}(Nb)$  classical bits for storing  $\hat{X}$ ,  $Y$  and  $\tau$ , and computing the dot product on line 7, the space complexity of Algorithm 1 is also  $\mathcal{O}(N \log N)$ . The qubit footprint of Algorithm 1 is  $\mathcal{O}(b)$  because we use  $b + 1$  qubits. Thus, it is possible to train the quantum discriminator shown in Fig. 2 in  $\mathcal{O}(N \log N)$  time, using  $\mathcal{O}(N \log N)$  classical bits and  $\mathcal{O}(b)$  qubits. Subsequently, inferencing can be performed on a given input using the quantum circuit seen in Fig. 2 in  $\mathcal{O}(N)$  time using just  $\mathcal{O}(b)$  qubits.

While the time complexity to train a quantum discriminator model is  $\mathcal{O}(N \log N)$ , it must be noted that there is a classical overhead for synthesizing a quantum circuit from an arbitrary unitary matrix. In some cases, this overhead can be exponentially complex. While it remains out of the scope of this paper to devise efficient techniques for quantum circuit synthesis for a given unitary matrix, we postulate that since the unitary matrix given in Eq. (1) has a well-defined structure, a quantum circuit could be synthesized efficiently by leveraging this structure.

### Generalizability

By generalizability, we refer to the ability of a machine learning model to make predictions on data points not encountered during training. The quantum discriminator has the ability to classify points in the training data set with a high degree of accuracy owing to an exponential number of model parameters  $(\theta_1, \theta_2, \dots, \theta_B)$ . It is highly complex and highly susceptible to overfitting the training data. This affinity to overfit is kept in check by the feature extraction process. If the extracted binary features are good and small in number ( $b \approx \mathcal{O}(\log N)$ ), the number of model parameters are also small ( $B \approx \mathcal{O}(N)$ ). The subsequent quantum discriminator would have a lower tendency of overfitting and would be generalizable.

### 2-bit binary classification

To demonstrate a proof of concept for the proposed quantum discriminator, we use it to classify 2-dimensional binary data points. There are four such data points: (0, 0), (0, 1), (1, 0) and (1, 1), located on the corners of a unit square. There are  $2^4 = 16$  ways of classifying these points into two classes as shown in Fig. 4. For example, one such way could be: (0, 0) and (0, 1) belong to Class 0, and (1, 0) and (1, 1) belong to Class 1, as shown by Case 10 in Fig. 4. In Fig. 5, we show the quantum circuits that can classify each of the 16 cases in Fig. 4. In Table 1, we present the unitary matrices of the quantum discriminator models that classify each of the 16 cases in Fig. 4.

We now elaborate Case 4 from Fig. 4. In Case 4, the points (0, 0), (0, 1) and (1, 1) belong to Class 0, while the point (1, 0) belongs to Class 1. The corresponding quantum circuit (Case 4 in Fig. 5) used for classification takes as input the two quantum data features ( $|x_1\rangle$  and  $|x_2\rangle$ ) as well as the prediction qubit initialized to  $|0\rangle$ . Next, it applies the Pauli- $X$  gate on  $|x_2\rangle$ , followed by a Toffoli gate (also called the CCNOT gate) with  $|x_1\rangle$  and  $|x_2\rangle$  as the control bits and  $|0\rangle$  as the target bit, followed by another Pauli- $X$  gate on  $|x_2\rangle$ .

The corresponding unitary operator ( $U_{\Theta}$ ) is shown in the Case 4 of Table 1. It looks like an identity matrix with one caveat. The  $2 \times 2$  sub-matrix starting at the fifth row and fifth column is a Pauli- $X$  operator instead of a  $2 \times 2$  identity matrix. Note that the first and second rows/columns of all  $U_{\Theta}$  shown in Tables 1 correspond to the data point (0, 0). Similarly, third and fourth rows/columns correspond to the data point (0, 1), fifth and sixth rows/columns correspond to (1, 0), and seventh and eighth rows/columns correspond to (1, 1). The starting indices (1, 3, 5 and 7) of these  $2 \times 2$  sub-matrices can be computed from the equation on line 7 of Algorithm 1. For each data point in each of the 16 cases, we can independently update the classification operator ( $U_{\Theta}$ ) so that it correctly classifies the said data point. In this way, the quantum discriminator is able to achieve near perfect accuracies on binary classification problems, provided appropriate binary features have been extracted from the data.

## Methods and empirical evaluation

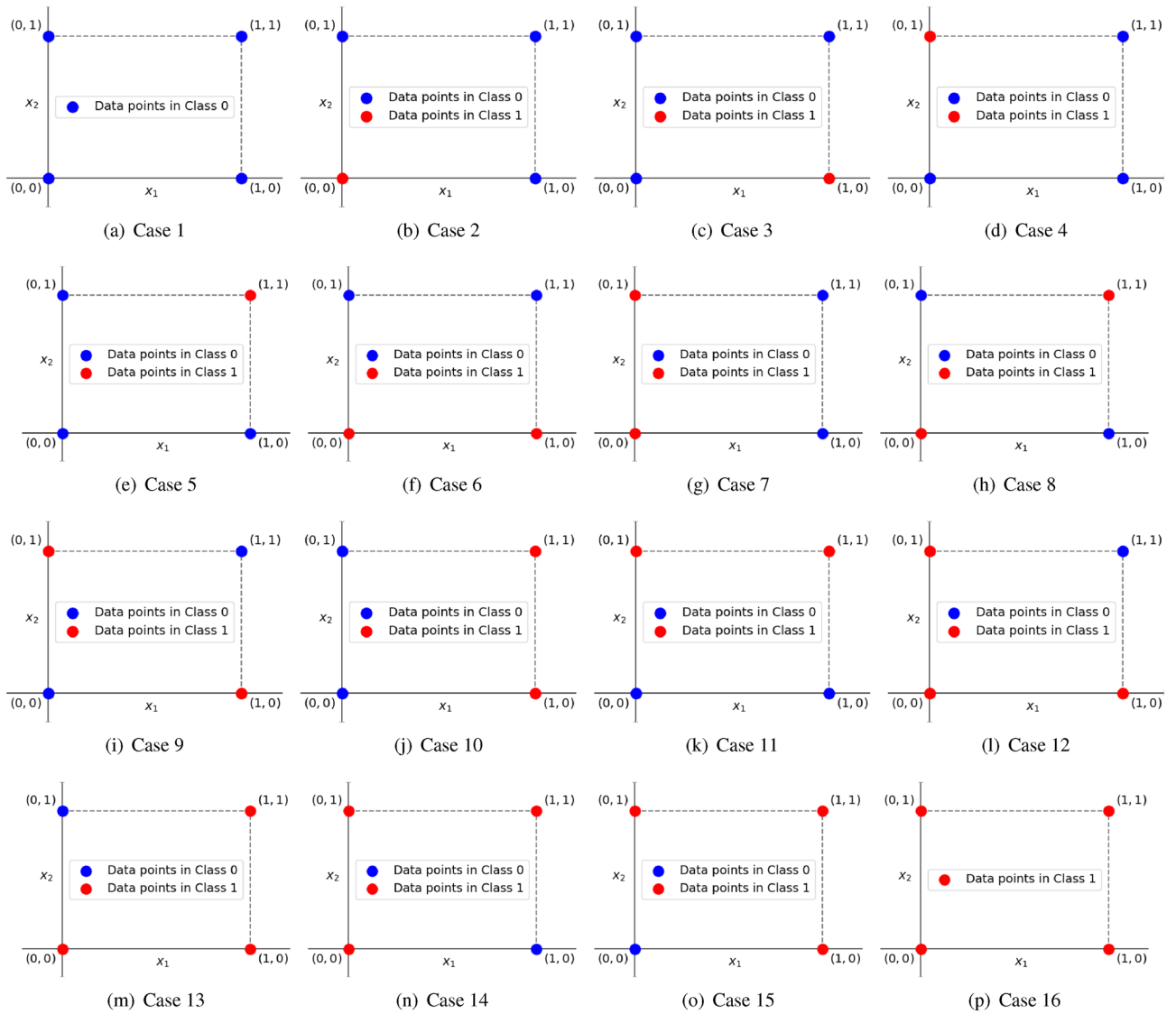
### Hardware and simulator details

We validated the performance of the quantum discriminator on the Iris and Bars and Stripes data sets. All our experiments were conducted on the IBM Jakarta processor as well as in numerous noise-free simulations using the QASM simulator in IBM Qiskit. The IBM Jakarta quantum computer consists of 7 qubits with a total quantum volume of 16. The average CNOT and readout errors on this machine were 0.015 and 0.035 respectively. The classical computer used in the training process was a desktop workstation having Intel Core i4-4670K CPU, running at 3.4 GHz, 64-bit operating system and 16 GB RAM.

### The Iris data set

The Iris data set is pervasive as a benchmark in machine learning, and is small enough to be embedded on current quantum computers. The data set contains 150 data points gathered from a sample of Iris flowers. Each data point consists of four measurements taken on the given Iris flower along with its species. The recorded attributes for each flower are petal length, petal width, sepal length, and sepal width measured in centimeters. The 150 data points are split evenly between three species of Iris: *Iris-Setosa*, *Iris-Versicolor*, and *Iris-Virginica*.

Using this data set, a model can be created whereby the petal/sepal lengths and widths of a given Iris datum can be used to predict its species. For the purpose of testing the quantum discriminator—which is designed for binary classification—the data set was restricted to just the *Iris-Setosa* and *Iris-Virginica* samples, which were labeled as Class 0 and Class 1 respectively. This reduced the size of the data set to 100 data points.



**Figure 4.** 16 cases of the 2-bit binary classification problem.

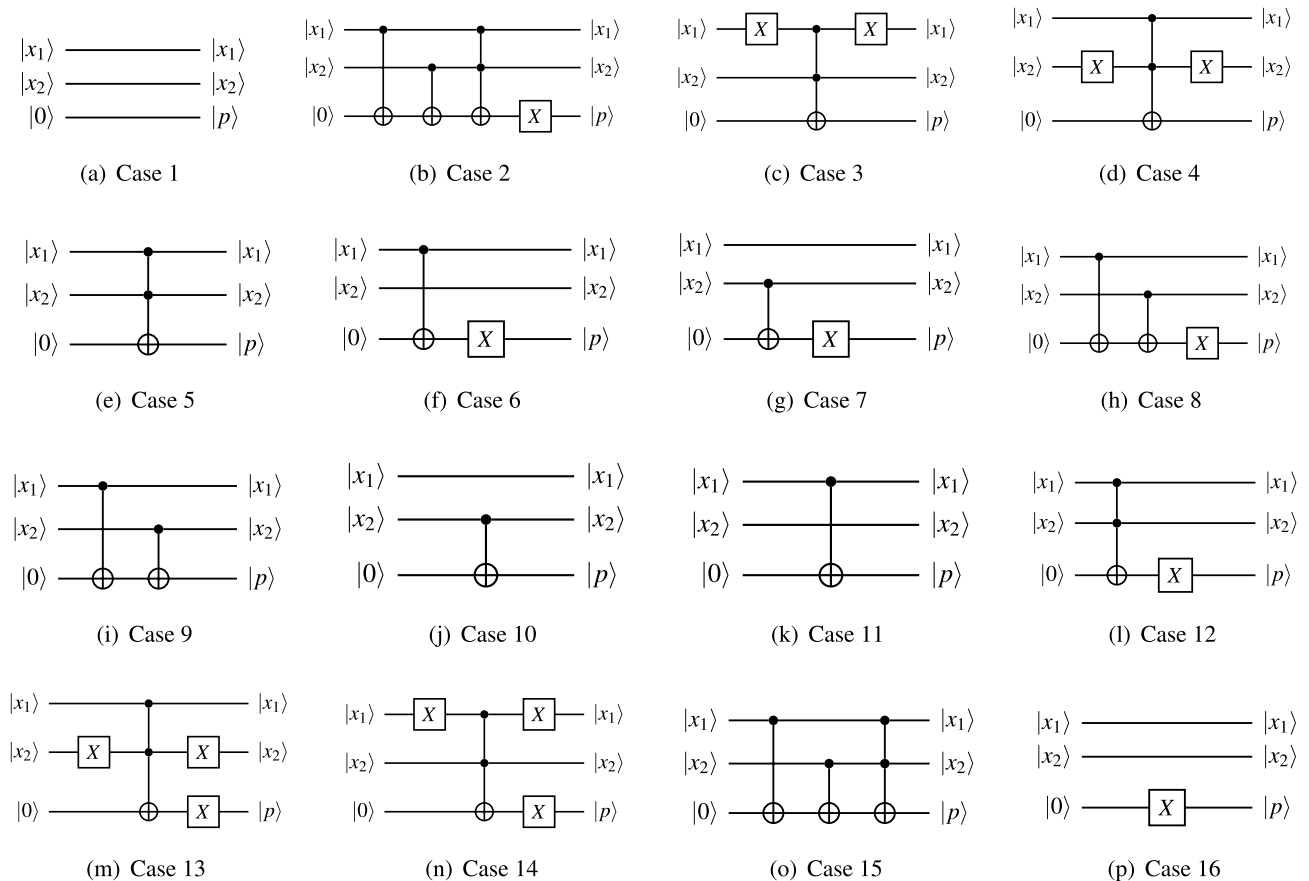
*Feature extraction*

Before training the quantum discriminator, binary features must be extracted from the data. In this case, three features were gathered from the data by imposing threshold values for the sepal length, sepal width, and petal length. The fourth attribute, petal width, was not considered in this feature extraction process. Specifically, our extracted data consisted 3-tuples of binary numbers,  $\hat{x} = (a_1, a_2, a_3) \in \mathbb{B}^3$ , where  $a_1 = 1$  if the sepal length was recorded to be above 5.50 cm,  $a_2 = 1$  if the sepal width was recorded to be above 3.00 cm,  $a_3 = 1$  if petal length was recorded to be above 3.00 cm, and each  $a_i$  was set to 0 if it failed to meet these respective threshold values. This feature extraction procedure resulted in a separable data set. Moreover, the extracted data set was found to span only three-quarters of our entire binary feature space, which has a theoretical size of  $2^3 = 8$ , i.e. 8 unique configurations of binary features.

*Training*

Training experiments were conducted in which the 100 data points were partitioned at random into a training data set of size  $N$ , and the remaining  $100 - N$  data points were reserved for validation. This parameter  $N$ , was varied from  $N = 4$ , where at most half of feature space could be sampled, to  $N = 80$ , which corresponds to a train/test split scheme commonly used to evaluate machine learning approaches. Training was performed in accordance with Algorithm 1.

The theory behind the quantum discriminator prescribes the number of features ( $b$ ) to be  $\mathcal{O}(\log N)$ . Since the size of the smallest and largest training sets in our experiments were 4 and 80 respectively, we would need to have between  $\log_2 4$  and  $\log_2 80$  binary features, i.e., 2–6 binary features. As mentioned above, we selected 3 binary features for this task, which is consistent with the assumptions of the quantum discriminator.



**Figure 5.** Quantum circuits for each case of the 2-bit binary classification problem.

In each trial, a quantum circuit was constructed using IBM's Qiskit software development kit in order to evaluate our model on each point in the validation set. Specifically, for each point in validation,  $(x_i, y)$ , a circuit was constructed which initializes a blank quantum register of 4 qubits into the state  $|\hat{x}; 0\rangle$  before passing into a series of gates equivalent to the unitary transformation obtained from the model parameters which were extracted in training. The predicted label,  $p$ , was then recorded for comparison with its true value  $y$ .

#### Results on the Iris data set

The three training sizes tested in this work are  $N = 80, 8$ , and 4. In the former two cases, experiments consisting of 300 trials each were conducted, with each experiment being performed once in the QASM simulator and once again on the IBM Jarkarta processor, meaning there were four experiments in total for these two cases. In each trial,  $N$  data points were selected at random to form a training set, which was used to train the quantum discriminator in accordance with Algorithm 1. The trained model was used for inferencing on each point in the validation set for benchmarking purposes. Benchmarking on the  $N = 4$  case was conducted analogously, except just one experiment was conducted (in simulation) and the number of trials was increased to 600.

In the case of  $N = 80$ , the discriminator obtained an average validation accuracy of 99.15% with a standard deviation of 1.878% in simulation. This is at par with the performance of some of the Variational Quantum Classifier (VQC) approaches in the literature<sup>37–39</sup>. On the Jakarta processor, however, the discriminator obtained a much lower accuracy of 89.13% on average with a standard deviation of 6.97%. A histogram depicting the distribution of model accuracies across the 300 trials in both cases is depicted in Fig. 6.

When the training set was lowered to size  $N = 8$ , the average validation accuracy dropped to 94.98% with a standard deviation of 9.047% in simulation, whereas the average accuracy on the Jakarta processor fell to 82.37% with a standard deviation of 8.403%. The distribution of model accuracies on validation is similarly displayed in Fig. 7. Figure 8 displays a box plot of model accuracies on quantum hardware for  $N$  equals 80 and 8.

In the case of  $N = 4$ , the average model accuracy across our 600 trials fell to 84.41%, with an increased standard deviation of 15.04%; the distribution of which can be seen in Fig. 9. Additional statistics were gathered in this case. Inaccurate predictions made on validation were separated into Type I (false positive) and Type II (false negative) errors. It was found that all errors made by the models belong to Type II, meaning the model had a false positive rate of 0 in each simulated trial. In other words, each simulated model had a precision of 100%. It was found that false negative rate (also called the *miss rate*) was 0.3074 on average with a standard deviation of 0.2934. This distribution, seen in Fig. 10 was heavily skew-right; i.e. the distribution was more concentrated on the side closer to a false-negative rate of 0. Consequently, the average recall (also called *sensitivity* or *hit rate*) was 0.6925 on average with the same standard deviation.

$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
Case 1	Case 2	Case 3	Case 4
$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
Case 5	Case 6	Case 7	Case 8
$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
Case 9	Case 10	Case 11	Case 12
$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
Case 13	Case 14	Case 15	Case 16

**Table 1.** Unitary matrices for each case of the 2-bit binary classification problem.

While these results are reported for the Iris data set and should be interpreted accordingly, we believe these are still very interesting. The fact that the quantum discriminator can achieve 84% accuracy when trained on just 4% of the data can be used to build approximate machine learning models with moderately high accuracy very quickly. Such models could then be refined quickly in subsequent training iterations, thereby reducing the training times.

### The bars and stripes data set

To further evaluate the performance of our quantum discriminator, a Bars and Stripes data set of size 100 was generated on a 3x3 grid. Each cell in this grid can either be illuminated or not. In this experiment, a grid is said to be a bar if it has a rectangle of illuminated cells whose width is strictly greater than its height. Similarly, a grid is said to be a stripe if it has a rectangle of illuminated cells whose height is strictly greater than its width. Accordingly, all bars and stripes viable under this formulation are displayed in Fig. 11. Using this scheme, 100 samples were drawn from a uniform sample of these viable bars and stripes to form the data set used in this experiment. The bars and strips were assigned to class 0 and 1 respectively. The Bars and Stripes experiments were run in simulation only.

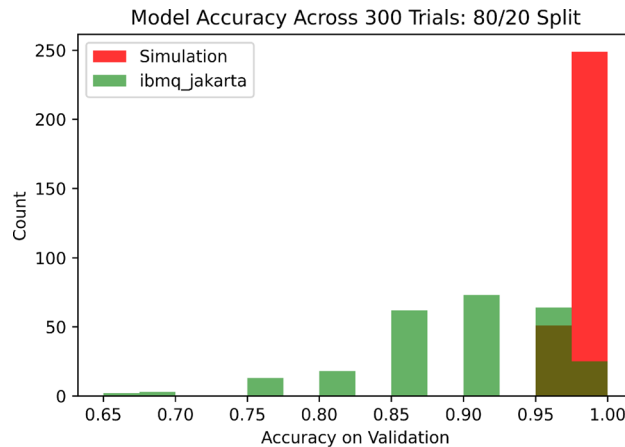
### Methodology

Nine binary features were extracted from each data point by reading the cells of each grid in lexicographic order, recording a one if the given cell is illuminated and a zero otherwise. Two experiments were conducted on the Bars and Stripes data set in a similar fashion to the Iris data set in Section "The Iris data set". Specifically, 300 trials (each) were conducted in noise-free simulation whereby the data set was randomly partitioned into a training set of size  $N$  and the remaining data were reserved for validation of the resulting model; the training sizes used in this case were  $N = 80$  and  $N = 11$ . The latter quantity was chosen as it is the minimal number of points needed to sample all of the Class 1 data in our binary feature space.

### Results on the bars and stripes data set

In the case of  $N = 80$ , the discriminator obtained an average validation accuracy of 98.38% with a standard deviation of 3.647% in simulation. This is at par with some of the QML approaches in the literature such as the hybrid neural networks and better than some others such as VQCs<sup>40,41</sup>. When  $N$  was lowered to 11, the average validation accuracy dropped to 71.02% with a standard deviation of 6.765% in simulation. Figure 12 displays how the distribution of model accuracies on validation varies in simulation with this change in  $N$ . These results were very similar to the Iris data set.





**Figure 6.** Histogram of validation accuracies on hardware and simulator in the case of  $N = 80$  for the Iris data set across 300 trials each.

### Discussion

The Quantum Discriminator fared extremely well in both of the cases where the size of the feature set was  $\mathcal{O}(\log N)$  for the Iris as well as the Bars and Stripes data sets. There was a noticeable gap ( $\approx 10\%$ ) in both Iris cases ( $N = 80$  and  $8$ ) between the discriminator's performance in simulation and on the IBM Jakarta processor. This discrepancy can be attributed to the fact that the state of the qubit as well as the inter-qubit connections on the quantum hardware are extremely sensitive to any kind of noise (thermal, radiation, vibrational etc.) in today's quantum computers. Consequently, it is extremely difficult to maintain the qubit states for longer periods of time, and the qubits have a tendency to lose their state during the computation—this is called decoherence. Qubit decoherence is known to result in considerably poor performance on the hardware as compared to simulation for many quantum algorithms.

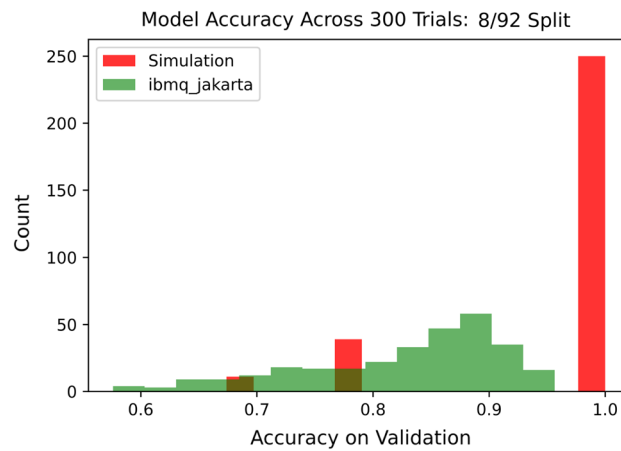
To mitigate this issue of decoherence, error correction regimes are often incorporated into quantum algorithms. However, no error-correction was performed in this experiment as it adds considerable overheads in terms of both time and space complexity. It did not seem warranted considering that only four qubits were required for the Iris experiments, and only ten qubits were required for the Bars and Stripes experiments under our feature extraction regime. It stands to reason that the disparity in performance of the simulated and real-world models would increase dramatically as the number of qubits required for inferring increases when working with today's quantum computers. Having said that, it is expected that with hardware and engineering improvements in the future, quantum computers would become less noisy, more reliable, and larger in size. These fault-tolerant quantum computers are expected to run quantum algorithms at par with (if not better than) the current simulation results. In both data sets, it was observed that the quantum discriminator can obtain moderately high accuracies even when the training set is sparse, meaning approximate models can be quickly and efficiently generated for subsequent refinement.

It is important to note that the quantum discriminator uniquely benefits from the quantum computers, which leverage the unique properties of quantum superposition and quantum entanglement. Specifically, the unitary matrix governing the quantum discriminator is  $2^b \times 2^b$ . This exponential matrix becomes intractable to store and compute on a classical computer as  $b$  increases. However, this can be done on a quantum computer using only  $b$  qubits.

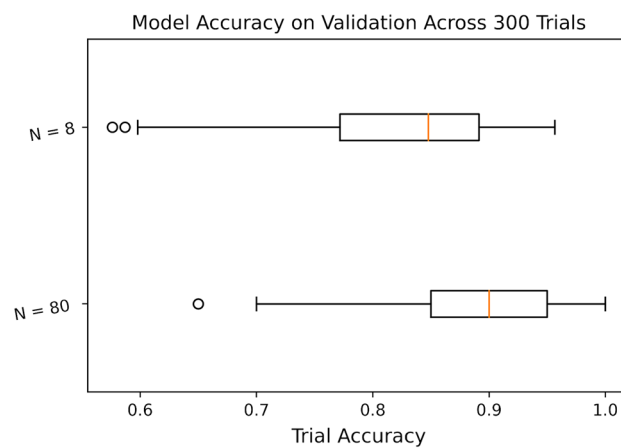
### Conclusion

Alternative computing paradigms, such as quantum computing, present a new frontier in which novel machine learning techniques can be developed to address the current limitations of classical learning techniques. In this work, we outline a quantum machine learning technique for binary classification called the quantum discriminator. The quantum discriminator is used as a discriminant function to infer the label of a given datum from its extracted features. The quantum discriminator is a  $2B \times 2B$  unitary matrix, which is parameterized by  $B$  parameters. It can be trained in  $\mathcal{O}(N \log N)$  time, using  $\mathcal{O}(N \log N)$  classical bits and  $\mathcal{O}(b)$  qubits. Inferring on the quantum discriminator can be performed in  $\mathcal{O}(N)$  time using  $\mathcal{O}(b)$  qubits. We demonstrated that this model can be used to completely solve the 2-bit binary classification problem Section "2-bit binary classification". Furthermore, we evaluated its performance on the Iris data set, which is a benchmark machine learning data set, and also on a 3x3 Bars and Stripes data set. This empirical evaluation demonstrates the discriminator's potential to generate highly accurate, inherently precise models on a separable data sets when appropriate binary features have been extracted from the data.

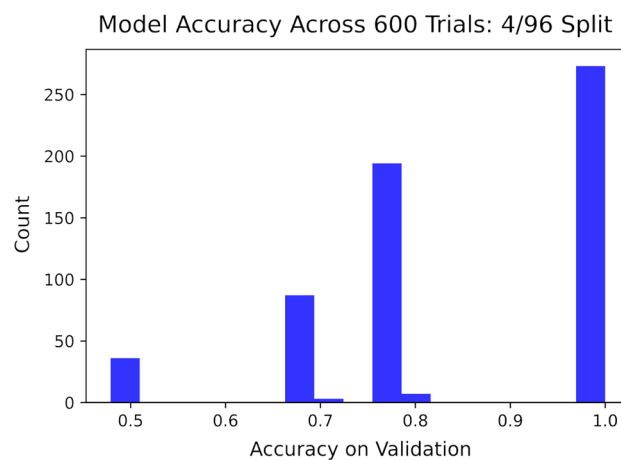
In our future work, we would like to evaluate the discriminator's performance on problems involving larger, more complex data sets, such as MNIST. We would also like to extend the quantum discriminator to multi-class classification problems. Lastly, we would like to investigate purely quantum training algorithms for training the quantum discriminator as opposed to the hybrid quantum-classical training algorithm described in this paper.



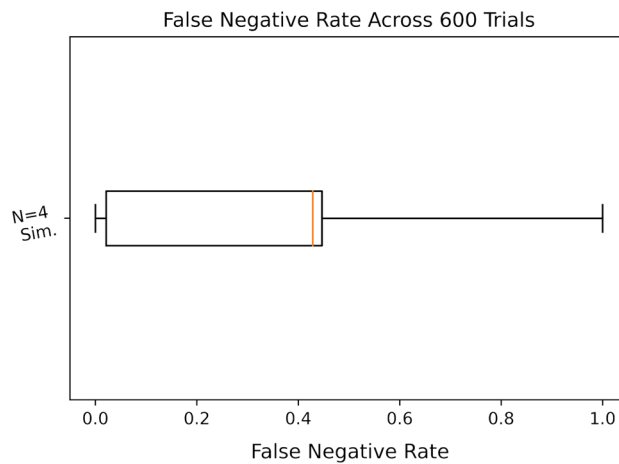
**Figure 7.** of validation accuracies on hardware and simulator in the case of  $N = 8$  for the Iris data set across 300 trials each.



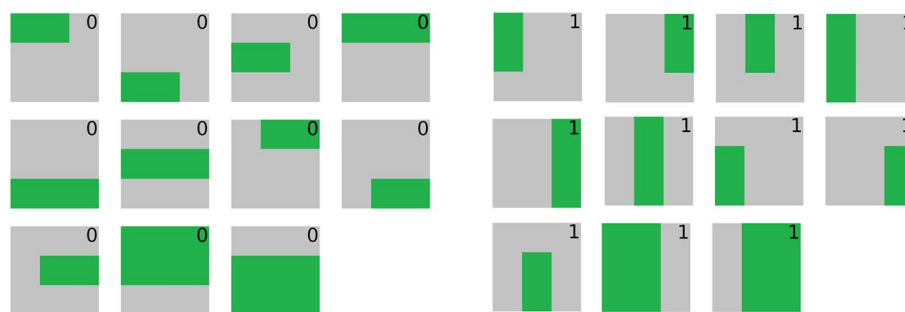
**Figure 8.** Box plot of model accuracies on the IBM Jakarta processor for  $N = 80$  and  $N = 8$  across 300 trials each.



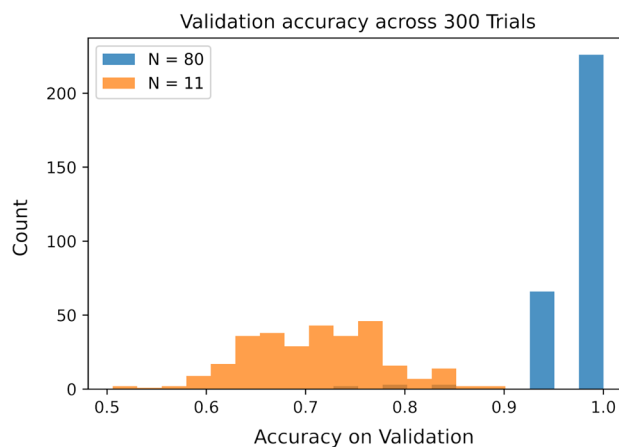
**Figure 9.** Histogram of validation accuracies on simulator in the case of  $N = 4$  for the Iris data set across 600 trials.



**Figure 10.** Box plot of model accuracies on simulator across 600 trials on the IBM Jakarta processor for  $N = 4$ .



**Figure 11.** A diagram depicting all possible bars (on the left) and stripes (on the right) in our data set. Here, cells are illuminated in green. Bars and stripes are assigned the classes 0 and 1 respectively.



**Figure 12.** Histogram of validation accuracies on simulator in the case of  $N = 80$  and 11 for the Bars and Stripes data set across 300 trials each.

### Data availability

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

Received: 17 October 2022; Accepted: 1 November 2023

Published online: 15 January 2024

## References

1. Qiu, J., Wu, Q., Ding, G., Xu, Y. & Feng, S. A survey of machine learning for big data processing. *EURASIP J. Adv. Signal Process.* **2016**, 67 (2016).
2. Aimone, J. B. Neural algorithms and computing beyond Moore's law. *Commun. ACM* **62**, 110–110 (2019).
3. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
4. Biamonte, J. *et al.* Quantum machine learning. *Nature* **549**, 195–202 (2017).
5. Date, P., Schuman, C., Kay, B., & Potok, T. *et al.* Neuromorphic computing is turing-complete. arXiv preprint [arXiv:2104.13983](https://arxiv.org/abs/2104.13983) (2021).
6. Date, P. *Combinatorial neural network training algorithm for neuromorphic computing* (Rensselaer Polytechnic Institute, 2019).
7. Ciliberto, C. *et al.* Quantum machine learning: A classical perspective. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **474**, 20170551 (2018).
8. Nielsen, M. A. & Chuang, I. Quantum computation and quantum information (2002).
9. Vazirani, U. On the power of quantum computation. *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* **356**, 1759–1768 (1998).
10. Preskill, J. Quantum computing: Pro and con. *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* **454**, 469–486 (1998).
11. Shor, P. W. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science* (ed. Shor, P. W.) 124–134 (IEEE, 1994).
12. Grover, L. K. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.* **79**, 325 (1997).
13. Ren, S., Cao, X., Wei, Y. & Sun, J. Face alignment at 3000 fps via regressing local binary features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1685–1692 (2014).
14. Chalkidis, I., Androustopoulos, I. & Michos, A. Extracting contract elements. In *Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law*, 19–28 (2017).
15. Lloyd, S. & Weedbrook, C. Quantum generative adversarial learning. *Phys. Rev. Lett.* **121**, 040502 (2018).
16. Dallaire-Demers, P.-L. & Killoran, N. Quantum generative adversarial networks. *Phys. Rev. A* **98**, 012324 (2018).
17. Blance, A. & Spannowsky, M. Quantum machine learning for particle physics using a variational quantum classifier. *J. High Energy Phys.* **2021**, 1–20 (2021).
18. Shingu, Y. *et al.* Boltzmann machine learning with a variational quantum algorithm. *Phys. Rev. A* **104**, 032413 (2021).
19. Quiroga, D., Date, P. & Pooser, R. Discriminating quantum states with quantum machine learning. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)* (eds Quiroga, D. *et al.*) 481–482 (IEEE, 2021).
20. Date, P. & Potok, T. Adiabatic quantum linear regression. *Sci. Rep.* **11**, 21905 (2021).
21. Arthur, D. & Date, P. Balanced k-means clustering on an adiabatic quantum computer. *Quantum Inf. Process.* **20**, 1–30 (2021).
22. Gambs, S. Quantum classification. arXiv preprint [arXiv:0809.0444](https://arxiv.org/abs/0809.0444) (2008).
23. Sentís, G., Calsamiglia, J., Muñoz-Tapia, R. & Bagan, E. Quantum learning without quantum memory. *Sci. Rep.* **2**, 1–8 (2012).
24. Chen, C., Dong, D., Qi, B., Petersen, I. R. & Rabitz, H. Quantum ensemble classification: A sampling-based learning control approach. *IEEE Trans. Neural Netw. Learn. Syst.* **28**, 1345–1359 (2016).
25. Sergioli, G., Giuntini, R. & Freytes, H. A new quantum approach to binary classification. *PLoS ONE* **14**, e0216224 (2019).
26. Park, D. K., Blank, C. & Petruccione, F. The theory of the quantum kernel-based binary classifier. *Phys. Lett. A* **384**, 126422 (2020).
27. Li, G., Song, Z. & Wang, X. Vsql: Variational shadow quantum learning for classification. arXiv preprint [arXiv:2012.08288](https://arxiv.org/abs/2012.08288) (2020).
28. Blank, C., Park, D. K., Rhee, J.-K. & Petruccione, F. Quantum classifier with tailored quantum kernel. *npj Quant. Inform.* **6**, 1–7 (2020).
29. Silver, D., Patel, T. & Tiwari, D. Quilt: Effective multi-class classification on quantum computers using an ensemble of diverse quantum classifiers. *Proc. AAAI Conf. Artif. Intell.* **36**, 8324–8332 (2022).
30. Havlíček, V. *et al.* Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209–212 (2019).
31. Schuld, M. & Killoran, N. Quantum machine learning in feature Hilbert spaces. *Phys. Rev. Lett.* **122**, 040504 (2019).
32. Bergou, J. A. & Hillery, M. Universal programmable quantum state discriminator that is optimal for unambiguously distinguishing between unknown states. *Phys. Rev. Lett.* **94**, 160501 (2005).
33. Lloyd, S., Schuld, M., Ijaz, A., Izaac, J. & Killoran, N. Quantum embeddings for machine learning. arXiv preprint [arXiv:2001.03622](https://arxiv.org/abs/2001.03622) (2020).
34. Shu, C., Ding, X. & Fang, C. Histogram of the oriented gradient for face recognition. *Tsinghua Sci. Technol.* **16**, 216–224 (2011).
35. Nguyen, T., Park, E.-A., Han, J., Park, D.-C. & Min, S.-Y. Object detection using scale invariant feature transform. In *Genetic and Evolutionary Computing* (ed. Nguyen, T.) 65–72 (Springer, 2014).
36. Li, C.-N., Shao, Y.-H. & Deng, N.-Y. Robust l1-norm two-dimensional linear discriminant analysis. *Neural Netw.* **65**, 92–104 (2015).
37. Zeguendry, A., Jarir, Z. & Quafafou, M. Quantum machine learning: A review and case studies. *Entropy* **25**, 287 (2023).
38. Pirhooshayan, M. & Terlaky, T. Quantum circuit design search. *Quant. Mach. Intell.* **3**, 1–14 (2021).
39. Cappelletti, W., Erbanni, R. & Keller, J. Polyadic quantum classifier. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (eds Cappelletti, W. *et al.*) 22–29 (IEEE, 2020).
40. Arthur, D. & Date, P. Hybrid quantum-classical neural networks. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* (eds Arthur, D. & Date, P.) 49–55 (IEEE, 2022).
41. Neumann, N., Phillipson, F. & Versluis, R. Machine learning in the quantum era. *Digitale Welt* **3**, 24–29 (2019).

## Acknowledgements

This manuscript has been authored in part by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>). This work was funded in part by the DOE Office of Science, Advanced Scientific Computing Research (ASCR) program. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Robinson Pino, program manager, under contract number DE-AC05-00OR22725. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. The authors would like to thank Sam Crawford of the Oak Ridge National Laboratory for editing this manuscript.

## Author contributions

P.D. formulated the idea of the quantum discriminator, developed the theoretical formulation and wrote the majority of the paper. W.S. implemented the quantum discriminator, ran experiments on benchmark data sets and contributed to the "Methods and empirical evaluation" section.

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to P.D.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

This is a U.S. Government work and not under copyright protection in the US; foreign copyright protection may apply 2024