Quantum adversarial machine learning

Sirui Lu,[1,2] Lu-Ming Duan,[1,*] and Dong-Ling Deng[1,3,†]

[1]*Center for Quantum Information, IIIS, Tsinghua University, Beijing 100084, People's Republic of China*

[2]*Max-Planck-Institut für Quantenoptik, Hans-Kopfermann-Strasse 1, D-85748 Garching, Germany*

[3]*Shanghai Qi Zhi Institute, 41th Floor, AI Tower, 701 Yunjin Road, Xuhui District, Shanghai 200232, China*

(Received 19 April 2020; accepted 14 July 2020; published 6 August 2020)

Adversarial machine learning is an emerging field that focuses on studying vulnerabilities of machine learning approaches in adversarial settings and developing techniques accordingly to make learning robust to adversarial manipulations. It plays a vital role in various machine learning applications and recently has attracted tremendous attention across different communities. In this paper, we explore different adversarial scenarios in the context of quantum machine learning. We find that, similar to traditional classifiers based on classical neural networks, quantum learning systems are likewise vulnerable to crafted adversarial examples, independent of whether the input data is classical or quantum. In particular, we find that a quantum classifier that achieves nearly the state-of-the-art accuracy can be conclusively deceived by adversarial examples obtained via adding imperceptible perturbations to the original legitimate samples. This is explicitly demonstrated with quantum adversarial learning in different scenarios, including classifying real-life images (e.g., handwritten digit images in the dataset MNIST), learning phases of matter (such as ferromagnetic/paramagnetic orders and symmetry protected topological phases), and classifying quantum data. Furthermore, we show that based on the information of the adversarial examples at hand, practical defense strategies can be designed to fight against a number of different attacks. Our results uncover the notable vulnerability of quantum machine learning systems to adversarial perturbations, which not only reveals another perspective in bridging machine learning and quantum physics in theory but also provides valuable guidance for practical applications of quantum classifiers based on both near-term and future quantum technologies.

DOI: 10.1103/PhysRevResearch.2.033212

I. INTRODUCTION

The interplay between machine learning and quantum physics may lead to unprecedented perspectives for both fields [1]. On the one hand, machine learning, or more broadly artificial intelligence, has progressed dramatically over the past two decades [2,3] and many problems that were extremely challenging or even inaccessible to automated learning have been solved successfully [4,5]. This raises new possibilities for utilizing machine learning to crack outstanding problems in quantum science as well [1,6–16]. On the other hand, the idea of quantum computing has revolutionized theories and implementations of computation, giving rise to new striking opportunities to enhance, speed up, or innovate machine learning with quantum devices, in turn [17–19]. This emergent field is growing rapidly, and notable progress is made on a daily basis. Yet, it is largely still in its infancy, and many important issues remain barely explored [1,17–19]. In this paper, we study such an issue concerning quantum machine learning in various adversarial scenarios. We show, with concrete examples, that quantum machine learning systems are likewise vulnerable to adversarial perturbations (see Fig. 1 for an illustration) and suitable countermeasures should be designed to mitigate the threat associated with them.

In classical machine learning, the vulnerability of machine learning to intentionally crafted adversarial examples as well as the design of proper defense strategies has been actively investigated, giving rise to an emergent field of adversarial machine learning [20–33]. Adversarial examples are inputs to machine learning models that an attacker has crafted to cause the model to make a mistake. The first seminal adversarial example dates back to 2004 when Dalvi *et al.* studied the techniques used by spammers to circumvent spam filters [34]. It was shown that linear classifiers could be easily fooled by few carefully crafted modifications (such as adding innocent text or substituting synonyms for words that are common in malignant message) in the content of the spam emails, with no significant change of the meaning and readability of the spam message. Since then, adversarial learning has attracted enormous attention, and different attack and defense strategies were proposed [22,27,32,33,35,36]. More strikingly, adversarial examples can even come in the form of imperceptibly small perturbations to input data, such as making a human-invisible change to every pixel in an image [21,37,38]. A prominent example of this kind in the context of deep learning was observed by Szegedy *et al.* and has become a celebrated prototype example that showcases the vulnerability of machine

*lmduan@tsinghua.edu.cn

†dldeng@tsinghua.edu.cn

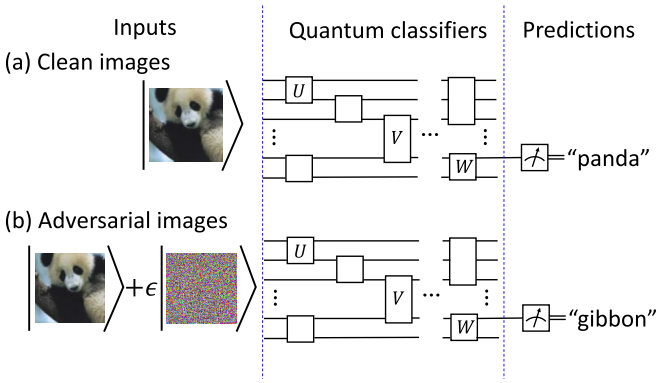

FIG. 1. A schematic illustration of quantum adversarial machine learning. (a) A quantum classifier that can successfully identify the image of a panda as "panda" with the state-of-the-art accuracy. (b) Adding a small amount of carefully crafted noise will cause the same quantum classifier to misclassify the slightly modified image, which is indistinguishable from the original one to human eyes, into a "gibbon" with notable high confidence.

learning in a dramatic way [21]: Starting with an image of a panda, an attacker may add a tiny amount of carefully crafted noise (which is imperceptible to the human eye) to make the image be classified incorrectly as a gibbon with notably high confidence. In fact, the existence of adversarial examples is now widely believed to be ubiquitous in classical machine learning. Almost all type of learning models suffer from adversarial attacks, for a wide range of data types including images, audio, text, and other inputs [23,24]. From a more theoretical computer science perspective, the vulnerability of classical classifiers to adversarial perturbations is reminiscent of the "no free lunch" theorem—there exists an intrinsic tension between adversarial robustness and generalization accuracy [39–41]. More precisely, it has been proved recently that if the data distribution satisfies the W_2 Talagrand transportation-cost inequality (a general condition satisfied in a large number of situations, such as the cases where the class-conditional distribution has log-concave density or is the uniform measure on a compact Riemannian manifold with positive Ricci curvature), any classical classifier could be adversarially deceived with high probability [42].

Meanwhile, over the past few years, a number of intriguing quantum learning algorithms have been discovered [17,43–61], and some been demonstrated in proof-of-principle experiments [62]. These algorithms exploit the unique enigmatic properties of quantum phenomena (such as superposition and entanglement) and promise to have exponential advantages compared to their classical counterparts. Notable examples include the Harrow-Hassidim-Lloyd (HHL) algorithm [63], quantum principal component analysis [64], quantum support-vector machine [65,66], and quantum generative model [58]. Despite this remarkable progress, quantum learning within different adversarial scenarios remains largely unexplored [67–69]. A noteworthy step along this direction has been made recently by Liu and Wittek [67], where they showed in theory that a perturbation by an amount scaling inversely with the Hilbert space dimension of a quantum system to be classified should be sufficient to cause a misclassification, indicating a fundamental trade-off between the robustness of the classification algorithms against adversarial attacks and the potential quantum advantages we expect for high-dimensional problems. Yet, in practice, it is unclear how to obtain adversarial examples in a quantum learning system, and the corresponding defense strategy is lacking as well.

In this paper, we study the vulnerability of quantum machine learning to various adversarial attacks, with a focus on a specific learning model called quantum classifiers. We show that, similar to traditional classifiers based on classical neural networks, quantum classifiers are likewise vulnerable to carefully crafted adversarial examples, which are obtained by adding imperceptible perturbations to the legitimate input data. We carry out extensive numerical simulations for several concrete examples, which cover different scenarios with diverse types of data (including handwritten digit images in the dataset MNIST, simulated time-of-flight images in a cold-atom experiment, and quantum data from a one-dimensional transverse field Ising model) and different attack strategies (such as fast gradient sign method [32], basic iterative method [27], momentum iterative method [35], and projected gradient descent [32] in the white-box attack setting, and transfer-attack method [70] and zeroth-order optimization [33] in the black-box attack setting, etc.) to obtain the adversarial perturbations. Based on these adversarial examples, practical defense strategies, such as adversarial training, can be developed to fight against the corresponding attacks. We demonstrate that, after the adversarial training, the robustness of the quantum classifier to the specific attack will increase significantly. Our results shed light on the fledgling field of quantum machine learning by uncovering the vulnerability aspect of quantum classifiers with comprehensive numerical simulations, which will provide valuable guidance for practical applications of using quantum classifiers to solve intricate problems where adversarial considerations are inevitable.

II. CLASSICAL ADVERSARIAL LEARNING AND QUANTUM CLASSIFIERS: CONCEPTS AND NOTATIONS

Modern technologies based on machine learning (especially deep learning) and data-driven artificial intelligence have achieved remarkable success in a broad spectrum of application domains [2,3], ranging from face or speech recognition, spam and malware detection, language translation, to self-driving cars and autonomous robots, etc. This success raises the illusion that machine learning is currently at a state to be applied robustly and reliably on virtually any task. Yet, as machine learning has found its way from laboratories to the real world, the security and integrity of its applications lead to more serious concerns as well, especially for these applications in safety and security-critical environments [20,23,24], such as self-driving cars, malware detection, biometric authentication, and medical diagnostics [71]. For instance, the sign recognition system of a self-driving car may misclassify a stop sign with a little dirt on it as a parking prohibition sign, and subsequently result in a catastrophic accident. In medical diagnostics, a deep neural network may incorrectly identify a slightly modified dermatoscopic image of a benign melanocytic nevus as malignant with even 100% confidence [72], leading to a possible medical disaster. To address these

crucial concerns and problems, a new field of adversarial machine learning has emerged to study vulnerabilities of different machine learning approaches in various adversarial settings and to develop appropriate techniques to make learning more robust to adversarial manipulations [25].

This field has attracted considerable attention and is growing rapidly. In this paper, we take one step further to study the vulnerabilities of quantum classifiers and possible strategies to make them more robust to adversarial perturbations. For simplicity and concreteness, we will only focus our discussion on supervised learning scenarios, although a generalization to unsupervised cases is possible and worth systematic future investigations. We start with a brief introduction to the basic concepts, notations, and ideas of classical adversarial learning and quantum classifiers. In supervised learning, the training data is labeled beforehand: $\mathcal{D}_N = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})\}$, where $\mathbf{x}^{(i)}$ $(i = 1, \ldots, N)$ is the data to be classified and $y^{(i)}$ denotes its corresponding label. The essential task of supervised learning is to learn from the labeled data a model $y = h(\mathbf{x}; \eta)$ (a classifier) that provides a general rule on how to assign labels to data outside the training set [73]. This is usually accomplished by minimizing certain loss function over some set of model parameters that are collectively denoted as η: $\min_\eta \mathcal{L}_N(\eta)$, where $\mathcal{L}_N(\eta) = \frac{1}{N}\sum_{i=1}^{N} L(h(\mathbf{x}^{(i)}; \eta), y^{(i)})$ denotes the averaged loss function over the training data set. To solve this minimization problem, different loss functions and optimization methods have been developed, each of them bearing its own advantages and disadvantages, and the choice of which one to use depends on the specific problem.

Unlike training the classifiers, generating adversarial examples is a different process, where we consider the model parameters η as fixed and instead optimize over the input space. More specifically, we search for a perturbation δ within a small region Δ, which can be added into the input sample $\mathbf{x}^{(i)}$ so as to *maximize* the loss function:

$$\max_{\delta\in\Delta} L(h(\mathbf{x}^{(i)} + \delta; \eta), y^{(i)}). \tag{1}$$

Here, in order to ensure that the adversarial perturbation is not completely changing the input data, we constrain δ to be from a small region Δ, the choice of which is domain-specific and vitally depends on the problem under consideration. A widely adopted choice of Δ is the ℓ_p-norm bound: $||\delta||_p \leqslant \epsilon$, where the ℓ_p-norm is defined as $\|x\|_p = (\sum_{i=1}^{N} \|x_i\|^p)^{\frac{1}{p}}$. In addition, since there is more than one way to attack machine learning systems, different classification schemes of the attacking strategies have been proposed in adversarial machine learning [24,25,74,75]. Here, we follow Ref. [25] and classify attacks along the following three dimensions: timing (considering when the attack takes place, such as attacks on models versus on algorithms), information (considering what information the attacker has about the learning model or algorithm, such as white-box versus black-box attacks), and goals (considering different reasons for attacking, such as targeted versus untargeted attacks). We will not attempt to exhaust all possible attacking scenarios, which is implausible due to their vastness and complexity. Instead, we focus on only several types of attacks that have already capture the essential messages we want to deliver in this paper. In particular, along the "information" dimension, we consider white-box and black-box attacks. In the white-box setting, the attacker has full information about the learned model and the learning algorithm, whereas the black-box setting assumes that the adversary does not have precise information about either the model or the algorithm used by the learner. In general, obtaining adversarial examples in the black-box setting is more challenging. Along the "goals" dimension, we distinguish two major categories: targeted and untargeted attacks. In a targeted attack, the attacker aims to deceive the classifier into outputting a particularly targeted label. In contrast, untargeted attacks (also called reliability attacks in the literature) just attempt to cause the classifier make erroneous predictions, but no particular class is aimed. We also mention that a number of different methods have been proposed to solve the optimization problem in Eq. (1) or its variants in different scenarios [23]. We refer to Refs. [21–23,25,27,31–33,35,36,70,76] for more technique details. As for our purpose, we will mainly explore the fast gradient sign method (FGSM) [32], basic iterative method (BIM) [27], projected gradient descent (PGD) [32], and momentum iterative method (MIM) [35] in the white-box setting and the transfer attack [22], substitute model attack [31,70], and zeroth-order optimization (ZOO) attack [33] methods in the black-box setting.

On the other hand, another major motivation for studying adversarial learning is to develop proper defense strategies to enhance the robustness of machine learning systems to adversarial attacks. Along this direction, a number of countermeasures have been proposed as well in recent years [25]. For instance, Kurakin *et al.* introduced the idea of adversarial training [77], where the robustness of the targeted classifier is enhanced by retraining with both the original legitimate data and the crafted data. Samangouei *et al.* came up with a mechanism [78] that uses generative adversarial network [79] as a countermeasure for adversarial perturbations. Papernot *et al.* proposed a defensive mechanism [80] against adversarial examples based on distilling knowledge in neural networks [81]. Each of these proposed defense mechanisms works notably well against particular classes of attacks, but none of them could be used as a generic solution for all kinds of attacks. In fact, we *cannot* expect a universal defense strategy that can make all machine learning systems robust to all types of attacks, as one strategy that closes a certain kind of attack will unavoidably open another vulnerability for other types of attacks which exploit the underlying defense mechanism. In this work, we will use adversarial learning to enhance the robustness of quantum classifiers against certain types of adversarial perturbations.

Quantum classifiers are counterparts of classical ones that run on quantum devices. In recent years, a number of different approaches have been proposed to construct efficient quantum classifiers [45,47–57,57,65,82,83], with some of them even implemented in proof-of-principle experiments. One straightforward construction, called the quantum variational classifier [45,47,49], is to use a variational quantum circuit to classify the data in a way analogous to the classical support vector machines [73]. Variants of this type of classifiers include hierarchical quantum classifiers [55] (such as these inspired by the structure of tree tensor network or multiscale entanglement renormalization ansatz) and quantum convolutional

neural networks [53]. Another approach, called the quantum kernel [50,51,82], utilizes the quantum Hilbert space as the feature space for data and compute the kernel function via quantum devices. Both the quantum variational classifier and the quantum kernel approach have been demonstrated in a recent experiment with superconducting qubits [51]. In addition, hierarchical quantum classifiers have also been implemented by using the IBM quantum experience [84] and their robustness to depolarizing noises has been demonstrated in principle [55]. These experiments showcase the intriguing potentials of using the noisy intermediate-scale quantum devices [85] (which are widely expected to be available in the near future) to solve practical machine learning problems, although an unambiguous demonstration of quantum advantages is still lacking. Despite these exciting advances, an important question of both theoretical and experimental relevance concerning the reliability of quantum classifiers remains largely unexplored: Are they robust to adversarial perturbations?

III. VULNERABILITY OF QUANTUM CLASSIFIERS

As advertised in the above discussion, quantum classifiers are vulnerable to adversarial perturbations. In this section, we will first introduce the general structure of the quantum classifiers and the learning algorithms used in this paper and several attacking methods to obtain adversarial perturbations, with technique details provided in the Appendix. We then apply these methods to concrete examples to explicitly show the vulnerability of quantum classifiers in diverse scenarios, including quantum adversarial learning of real-life images (e.g., handwritten digit images in MNIST), topological phases of matter, and quantum data from the ground states of physical Hamiltonians.

A. Quantum classifiers: Training and adversarial attacks

Quantum classifiers take quantum states as input. Thus, when they are used to classify classical data, we need first to convert classical data into quantum states. This can be done with an encoding operation, which basically implements a feature map from the D-dimensional Euclidean space (where the class data are typically represented by D-dimensional vectors) to the 2^n-dimensional Hilbert space for n qubits: $\varphi : \mathbb{R}^D \to \mathbb{C}^{2^n}$. There are two common ways of encoding classical data into quantum states: amplitude encoding and qubit encoding [45,48,63–65,65,86–91]. Amplitude encoder maps an input vector $\mathbf{x} \in \mathbb{R}^D$ (with some possible preprocessing such as normalization) directly into the amplitudes of the 2^n-dimensional ket vector $|\psi\rangle_{\text{in}}$ for n qubits in the computational basis. Here, for simplicity, we assume that D is a power of 2 such that we can use $D = 2^n$ amplitudes of a n-qubit system (in fact, if $D < 2^n$ we can add $2^n - D$ zeros at the end of the input vector to make it of length 2^n). Such a converting procedure can be achieved with a circuit whose depth is linear in the number of features in the input vectors with the routines in Refs. [92–94]. With certain approximation or structure, the required overhead can be reduced to polylogarithmic in D [95,96]. This encoding operation can also be made more efficient by using more complicated approaches such as tensorial feature maps [45]. Qubit encoder, in contrast, uses D [rather than $O(\log D)$ as in amplitude encoding] qubits to encode the input vector. We first rescale the data vectors elementwise to lie in $[0, \frac{\pi}{2}]$ and encode each element with a qubit using the following scheme: $|\phi_d\rangle = \cos(x_d)|0\rangle + \sin(x_d)|1\rangle$, where x_d is the dth element of the rescaled vector. The total quantum input state that encodes the data vectors is then a tensor product $|\phi\rangle = \otimes_{d=1}^{D} |\phi_d\rangle$. Qubit encoding does not require a quantum random access memory [90] or a complicated circuit to prepare the highly entangled state $|\psi\rangle_{\text{in}}$, but it demands much more qubits to perform the encoding and hence is more challenging to numerically simulate the training and adversarial attacking processes on a classical computer. As a result, we will only focus on amplitude encoding in this work, but the generalization to other encoding schemes is straightforward and worth investigation in the future.

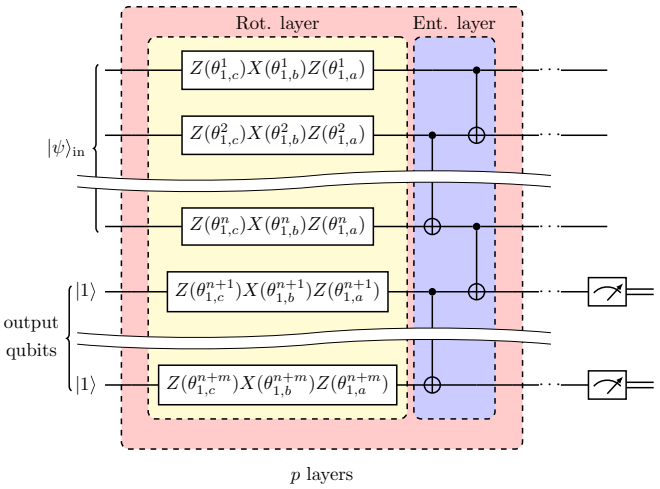

FIG. 2. The sketch of a quantum circuit classifier. The classifier consists of p layers, with each layer containing a rotation unit and an entangler unit. The rotation unit performs arbitrary single-qubit Euler rotations implemented as a combination of Z and X gates: $U_{q,i}(\boldsymbol{\theta}) = Z_{\theta^c_{q,i}} X_{\theta^b_{q,i}} Z_{\theta^a_{q,i}}$ with θ representing the Euler angles, q identifying the qubit, and $i = 1, 2, \ldots, p$ referring to the label of layers. The entangler unit entangles all qubits and is composed of a series of controlled-NOT (CNOT) gates. The initial state $|\psi\rangle_{\text{in}}$, which is a n-qubit state, encodes the complete information of the input data to be classified. The projection measurement on the output qubits gives the predicting probability for each category and the input data are assigned a label that bears the largest probability.

We choose a hardware-efficient quantum circuit classification model, which has been used as a variational quantum eigensolver for small molecules and quantum magnets in a recent experiment with superconducting qubits [97]. The schematic illustration of the model is shown in Fig. 2. Without loss of generality, we assume that the number of categories to be classified is K and each class is labeled by an integer number $1 \leqslant k \leqslant K$. We use m qubits ($2^{m-1} < K \leqslant 2^m$) to serve as output qubits that encode the category labels. A convenient encoding strategy that turns discrete labels into a vector of real numbers is the so-called one-hot encoding [73], which converts a discrete input value $0 < k \leqslant K$ into a vector $\mathbf{a} \equiv (a_1, \ldots, a_{2^m})$ of length 2^m with $a_k = 1$ and $a_j = 0$ for $j \neq k$. For the convenience of presentation, we will use y and $\mathbf{a}$ interchangeably to denote the labels

throughout the rest of the paper. In such a circuit model, we first prepare the input state to be $|\psi\rangle_{\text{in}} \otimes |1\rangle^{\otimes m}$ with $|\psi\rangle_{\text{in}}$ an n-qubit state encoding the complete information of the data to be classified, and then apply a unitary transform consisting of p layers of interleaved operations. Each layer contains a rotation unit that performs arbitrary single-qubit Euler rotations and an entangler layer that generates entanglement between qubits. This generates a variational wave function $|\Psi(\Theta)\rangle = \prod_{i=1}^{p} U_i(|\psi\rangle_{\text{in}} \otimes |1\rangle^{\otimes m})$, where $U_i = [\prod_q U_i^q(\boldsymbol{\theta}_i)]U_{\text{ENT}} = (\prod_q Z_{\theta_{i,c}^q} X_{\theta_{i,b}^q} Z_{\theta_{i,a}^q})U_{\text{ENT}}$ denotes the unitary operation for the ith layer. Here, U_{ENT} represents the unitary operation generated by the entangler unit and we use $\boldsymbol{\theta}_i$ to denote collectively all the parameters in the ith layer and Θ to denote collectively all the parameters evolved in the whole model. We mention that the arbitrary single-qubit rotation together with the control-NOT (CNOT) gate gives a universal gate set in quantum computation. Hence, our choice of this circuit classifier is universal as well, in the sense that it can approximate any desired function as long as p is large enough. One may choose other models, such as hierarchical quantum classifiers [55] or the quantum convolutional neural network [53], and we expect that the attacking methods and the general conclusion should carry over straightforwardly to these models.

During the training process, the variational parameters Θ will be updated iteratively so as to minimize certain loss functions. The measurement statistics on the output qubits will determine the predicted label for the input data encoded in state $|\psi\rangle_{\text{in}}$. For example, in the case of two-category classification, we can use $y \in \{0, 1\}$ to label the two categories and the number of output qubits is one. We estimate the probability for each class by measuring the expectation values of the projections: $P(y = l) = \text{Tr}(\rho_{\text{out}}|l\rangle\langle l|)$, where $l = 0, 1$ and $\rho_{\text{out}} = \text{Tr}_{1,\ldots,n}(|\Psi(\Theta)\rangle\langle\Psi(\Theta)|)$ is the reduced density matrix for the output qubit. We assign a label $y = 0$ to the data sample $\mathbf{x}$ if $P(y = 0)$ is larger than $P(y = 1)$ and say that $\mathbf{x}$ is classified to be in the 0 category with probability $P(y = 0)$ by the classifier. The generalization to multicategory classification is straightforward. One observation which may simplify the numerical simulations a bit is that the diagonal elements of ρ_{out}, denoted as $\mathbf{g} \equiv (g_1, \ldots, g_{2^m}) = \text{diag}(\rho_{\text{out}})$, in fact give all the probabilities for the corresponding categories.

In classical machine learning, a number of different loss functions have been introduced for training the networks and characterizing their performances. Different loss functions possess their own pros and cons and are best suitable for different problems. For our purpose, we define the following loss function based on cross-entropy for a single data sample encoded as $|\psi\rangle_{\text{in}}$:

$$L(h(|\psi\rangle_{\text{in}}; \Theta), \mathbf{a}) = -\sum_k a_k \log g_k. \tag{2}$$

During the training process, a classical optimizer is used to search for the optimal parameters Θ^* that minimize the averaged loss function over the training data set: $\mathcal{L}_N(\Theta) = \frac{1}{N}\sum_{i=1}^{N} L(h(|\psi\rangle_{\text{in}}^{(i)}; \Theta), \mathbf{a}^{(i)})$. Various gradient descent algorithms, such as the stochastic gradient descent [98] and quantum natural gradient descent [99,100], can be employed to do the optimization. We use Adam [101,102], which is an adaptive learning rate optimization algorithm designed specifically for training deep neural networks, to train the quantum classifiers.

A crucial quantity that plays a vital role in minimizing $\mathcal{L}_N(\Theta)$ is its gradient with respect to model parameters. Interestingly, owing to the special structures of our quantum classifiers, this quantity can be directly obtained from the projection measurements through the following equality [59]:

$$\frac{\partial\langle\mathcal{L}_N(\Theta)\rangle_\vartheta}{\partial\vartheta} = \frac{1}{2}\left(\langle\mathcal{L}_N(\Theta)\rangle_{\vartheta+\frac{\pi}{2}} - \langle\mathcal{L}_N(\Theta)\rangle_{\vartheta-\frac{\pi}{2}}\right), \tag{3}$$

where ϑ denotes an arbitrary single parameter in our circuit classifier and $\langle\mathcal{L}_N(\Theta)\rangle_\xi$ ($\xi = \vartheta, \vartheta + \frac{\pi}{2}$, and $\vartheta - \frac{\pi}{2}$) represents the expectation value of $\mathcal{L}_N(\Theta)$ with the corresponding parameter set to be ξ. We note that the equality in Eq. (3) is exact, in sharp contrast to other models for quantum variational classifiers where the gradients can only be approximated by finite-difference methods in general. It has been proved that an accurate gradient based on quantum measurements could lead to substantially faster convergence to the optimum in many scenarios [103], in comparison with the finite-difference method approach.

We now give a general recipe on how to generate adversarial perturbations for quantum classifiers. Similar to the case of classical adversarial learning, this task essentially reduces to another optimization problem where we search for a small perturbation within an appropriate region Δ that can be added into the input data so that the loss function is maximized. A quantum classifier can classify both classical and quantum data. Yet, adding perturbations to classical data is equivalent to adding perturbations to the initial quantum state $|\psi\rangle_{\text{in}}$. Hence, it is sufficient to consider only perturbations to $|\psi\rangle_{\text{in}}$, regardless of whether the data to be classified are quantum or classical. A pictorial illustration of adding adversarial perturbations to the input data for a quantum classifier is shown in Fig. 3. In the case of untargeted attacks, we attempt to search a perturbation operator U_δ acting on $|\psi\rangle_{\text{in}}$ to maximize the loss function:

$$U_\delta \equiv \underset{U_\delta\in\Delta}{\text{argmax}}\, L(h(U_\delta|\psi\rangle_{\text{in}}; \Theta^*), \mathbf{a}), \tag{4}$$

where Θ^* denotes the fixed parameters determined during the training process, $|\psi\rangle_{\text{in}}$ encodes the information of the data sample $\mathbf{x}$ supposed to be under attack, and $\mathbf{a}$ represents the correct label for $\mathbf{x}$ in the form of one-hot encoding. On the other hand, in the case of targeted attacks we aim to search a perturbation $U_\delta^{(\text{t})}$ that minimizes (rather than maximizes) the loss function under the condition that the predicted label is targeted to be a particular one:

$$U_\delta^{(\text{t})} \equiv \underset{U_\delta^{(\text{t})}\in\Delta}{\text{argmin}}\, L\left(h\left(U_\delta^{(\text{t})}|\psi\rangle_{\text{in}}; \Theta^*\right), \mathbf{a}^{(\text{t})}\right), \tag{5}$$

where $\mathbf{a}^{(\text{t})}$ is the targeted label that is different from the correct one $\mathbf{a} \neq \mathbf{a}^{(\text{t})}$.

In general, Δ can be a set of all unitaries (or even any completely positive and trace-preserving operations) that are close to the identity operator. This corresponds to the additive attack in classical adversarial machine learning, where we modify each component of the data vector independently. In our simulations, we use automatic differentiation [104], which

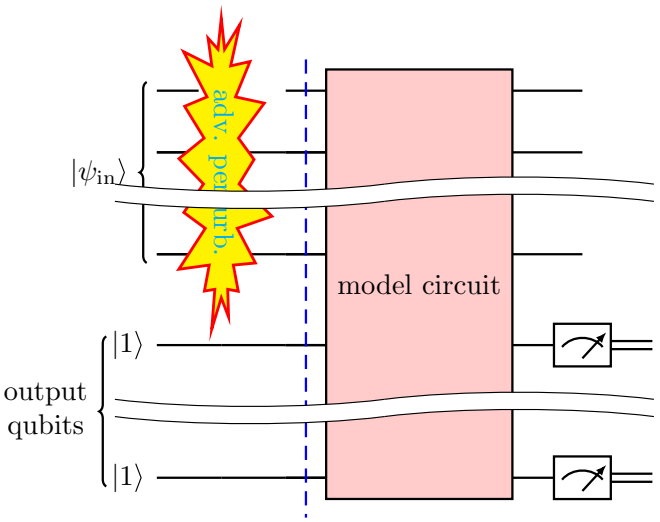

FIG. 3. A sketch of adding adversarial perturbations to the input data for quantum classifiers. Throughout this paper, we mainly focus on evasion attack [25], which is the most common type of attack in adversarial learning. In this setting, the attacker attempts to deceive the quantum classifiers by adjusting malicious samples during the testing phase. Adding a tiny amount of adversarial noise can cause quantum classifiers to make incorrect predictions.

computes derivatives to machine precision, to implement this type of attack. In addition, for simplicity we can further restrict Δ to be a set of products of local unitaries that are close to the identity operator. This corresponds to the functional adversarial attack [105] in classical machine learning. It is clear that the searching space for the functional attack is much smaller than that for the additive attack and one may regard the former as a special case for the later.

We numerically simulate the training and inference process of the quantum classifiers on a classical cluster by using the Julia language [106] and Yao.jl [107] framework. We run the simulation parallelly on the central processing units (CPUs) or graphical processing units (GPUs), depending on different scenarios. The parallel nature of the minibatch gradient descent algorithm naturally fits the merits of GPUs and thus we use CuYao.jl [107], which is a very efficient GPU implementation of Yao.jl [107], to gain speedups for the cases that are more resource consuming. We find that the performance of calculating minibatch gradients on a single GPU is ten times better than that of parallelly running on CPUs with 40 cores. The automatic differentiation is implemented with Flux.jl [108] and Zygote.jl [109]. Based on this implementation, we can optimize over a large number of parameters for circuit depth as large as $p = 50$. In general, we find that increases in circuit depth (model capacity) are conducive to the achieved accuracy. We check that the model does not overfit because the loss of the training data set and validation data set is close, so there is no need for introducing regularization techniques such as Dropout [110] to avoid overfitting.

Now we have introduced the general structure of our quantum classifiers and the methods to train them and to obtain adversarial perturbations. In the following subsections, we will demonstrate how these methods work by giving three concrete examples. These examples explicitly showcase the extreme vulnerability of quantum classifiers.

B. Quantum adversarial learning images

Quantum information processors possess unique properties such as quantum parallelism and quantum superposition, making them intriguing candidates for speeding up image recognitions in machine learning. It has been shown that some quantum image-processing algorithms may achieve exponential speedups over their classical counterparts [111,112]. Researchers have employed quantum classifiers for many different image data sets [45]. Here, we focus on the MNIST handwritten digit classification data set [113], which is widely considered to be a real-life test bed for machine learning paradigms. For this data set, near-perfect results have been reached using various classical supervised learning algorithms [114]. The MNIST data set consists of hand-drawn digits, from 0 through 9, in the form of grayscale images. Each image is two dimensional, and contains 28×28 pixels. Each pixel of an image in the dataset has a pixel value, which is an integer ranging from 0 to 255 with 0 meaning the darkest and 255 the whitest color. For our purpose, we slightly reduced the size of the images from 28×28 pixels to 16×16 pixels, so that we can simulate the training and attacking processes of the quantum classifier with moderate classical computational resources. In addition, we normalize these pixel values and encode them into a pure quantum state using the amplitude encoding method mentioned in Sec. III A.

We first train the quantum classifiers to identify different images in the MNIST with sufficient classification accuracy. The first case we consider is a two-category classification problem, where we aim to classify the images of digits 1 and 9 by a quantum classifier with structures introduced shown in Fig. 2. From the MNIST data set, we select out all images of 1 and 9 to form a subset, which contains a training data set of size 11 633 (used for training the quantum classifier), a validation data set of size 1058 (used for tuning hyperparameters, such as the learning rate), and a testing set of size 2144 (used for evaluating the final performance of the quantum classifier). In Fig. 4, we plot the average accuracy and loss for the training and validation data sets respectively as a function of the number of epochs. From this figure, the accuracy for both the training and validation increases rapidly at the beginning of the training process and then saturate at a high value of $\approx$98%. Meanwhile, the average loss for both training and validation decreases as the number of epochs increases. The difference between the training loss and validation loss is very small, indicating that the model does not overfit. In addition, the performance of the quantum classifier is also tested on the testing set and we find that our classifier can achieve a notable accuracy of 98% after around 15 epochs.

For two-category classifications, the distinction between targeted and untargeted attacks blurs since the target label can only be simply the alternative label. Hence, in order to illustrate the vulnerability of quantum classifiers under targeted attacks, we also need to consider a case of multicategory classification. To this end, we train a quantum classifier to distinguish four categories of handwritten digits: 1, 3, 7,

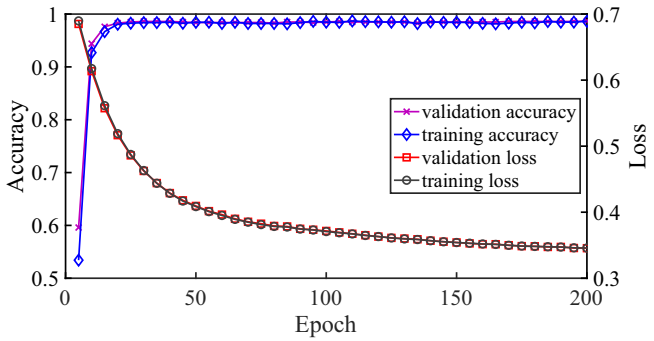

FIG. 4. The average accuracy and loss as a function of the number of training steps. We use a depth-10 quantum classifier with structures shown in Fig. 2 to perform binary classification for images of digits 1 and 9 in MNIST. To train the classifier, we use the Adam optimizer with a batch size of 256 and a learning rate of 0.005 to minimize the loss function in Eq. (2). The accuracy and loss are averaged on 11633 training samples and 1058 validation samples (which are not contained in the training dataset).

and 9. Our results are plotted Fig. 5. Similar to the case of two-category classification, we find that both the training and validation accuracies increase rapidly at the beginning of the training process and then saturate at a value of ≈92%, which is smaller than that for the two-category case. After training, the classifier is capable of predicting the corresponding digits for the testing data set with an accuracy of 91.6%. We mention that one can further increase the accuracy for both the two- and four-category classifications, by using the original 28 × 28-pixel images in MNIST or using a quantum classifier with more layers, but this demands more computational resources.

After training, we now fix the parameters of the corresponding quantum classifiers and study the problem of how to generate adversarial examples in different situations. We consider both the white-box and black-box attack scenarios. For the white-box scenario, we explore both untargeted and targeted attacks. For the black-box scenario, we first generate

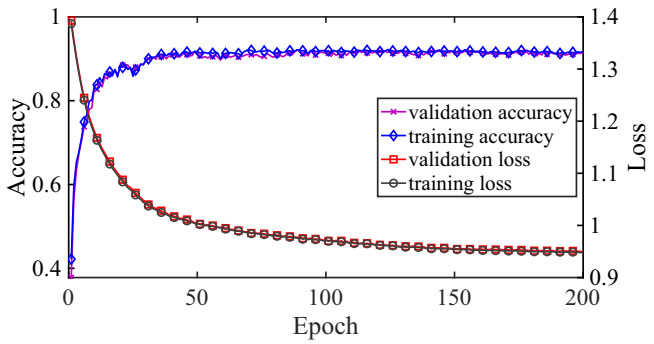

FIG. 5. The average accuracy and loss for the four-category quantum classifier as a function of the number of epochs. Here, we use a quantum classifier with structures shown in Fig. 2 and depth 40 ($p = 40$) to perform multiclass classification for images of digits 1, 3, 7, and 9. To train the classifier, we use the Adam optimizer with a batch size of 512 and learning rate of 0.005 to minimize the loss function in Eq. (2). The accuracy and loss are averaged on 20 000 training samples and 2000 validation samples.

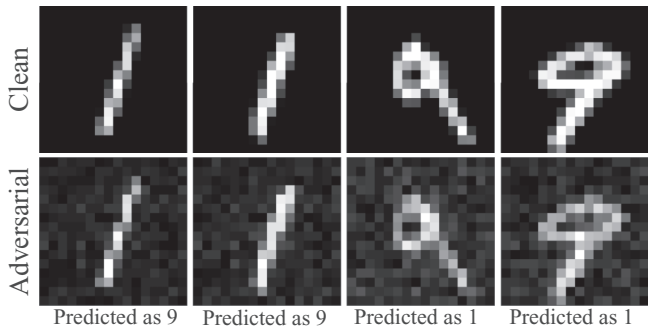

FIG. 6. The clean and the corresponding adversarial images for the quantum classifier generated by the basic iterative method (see the Appendix). Here, we apply the additive attack in the white-box untargeted setting. For the legitimate clean images, the quantum classifier can correctly predict their labels with confidence larger than 78%. After attacks, the classifier will misclassify the crafted images of digit 1 (9) as digit 9 (1) with notably high confidence, although the differences between the crafted and clean images are almost imperceptible to human eyes. In fact, the average fidelity is 0.916, which is very close to unity.

adversarial examples for classical classifiers and show that quantum classifiers are also vulnerable to these examples owing to the transferability properties of adversarial examples.

1. White-box attack: Untargeted

In the white-box setting, the attacker has full information about the quantum classifiers and the learning algorithms. In particular, the attacker knows the loss function that has been used and hence can calculate its gradients with respect to the parameters that characterize the perturbations. As a consequence, we can use different gradient-based methods developed in the classical adversarial machine learning literature, such as the FGSM [32], BIM [27], PGD [32], and MIM [35], to generate adversarial examples. For untargeted attacks, the attacker only attempts to cause the classifier to make incorrect predictions, but no particular class is aimed. In classical adversarial learning, a well-known example in the white-box untargeted scenario concerns facial biometric systems [115], whereby wearing a pair of carefully crafted eyeglasses the attacker can have her face misidentified by the state-of-the-art face-recognition system as any other arbitrary face (dodging attacks). Here, we show that quantum classifiers are vulnerable to such attacks as well.

For the simplest illustration, we first consider attacking additively the two-category quantum classifier discussed above in the withe-box untargeted setting. In Fig. 6, we randomly choose samples for digits 1 and 9 from MNIST and then solve the Eq. (4) iteratively by the BIM method to obtain their corresponding adversarial examples. This figure shows the original clean images and their corresponding adversarial ones for the two-category quantum classifier. For these particular clean images, the quantum classifier can correctly assign their labels with confidence larger than 78%. Yet, after attacks the same classifier will misclassify the crafted images of digit 1 (9) as digit 9 (1) with decent high confidence 73%. Strikingly, the obtained adversarial examples look the same as the original legitimate samples. They only differ by a tiny amount of noise

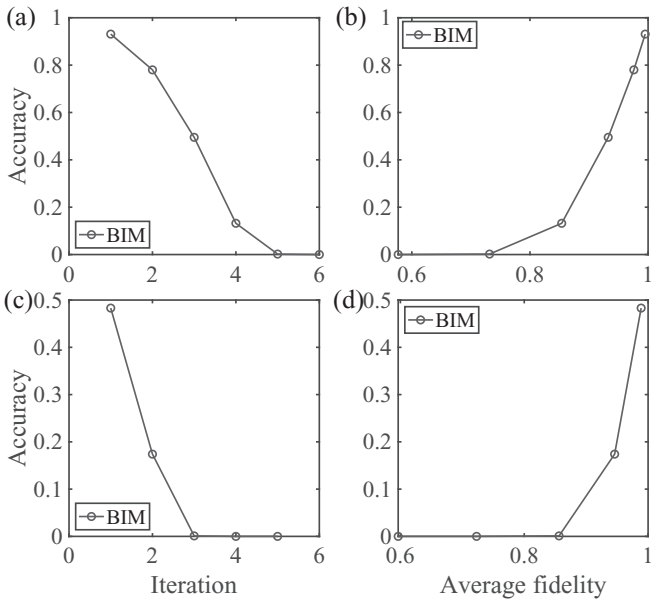

FIG. 7. Effect of adversarial untargeted additive attacks on the accuracy of the quantum classifier for the problem of classifying handwritten digits. We use the basic iterative method to obtain adversarial examples. The circuit depth of the model is 20. We choose the step size as 0.1. [(a), (b)] For the classifier that classifies digit 1 and 9, accuracy decreases as the average fidelity between the adversarial samples and clean samples decreases. Accuracy decreases as we increase the number of iterations of the attacking algorithm. [(c), (d)] Similar plots for the problem of classifying four digits 1, 3, 7, and 9.

that is almost imperceptible to human eyes. To further verify that the vulnerability of the quantum classifier is not specific to particular images, but rather generic for most of (if not all) images in the data set, we apply the same attack to all images of digits 1 and 9 in the testing set of MNIST. In Fig. 7(a), we plot the accuracy as a function of the number of the BIM iterations. It is clear from this figure that the accuracy decreases rapidly at the beginning of the attack, indicating that more adjusted images are misclassified. After five BIM iterations, the accuracy decreases to zero and all adjusted images become adversarial examples misclassified by the quantum classifier. In addition, to characterize how close a clean legitimate image is to its adversarial counterpart in the quantum framework, we define the fidelity between the quantum states that encode them: $F = |\langle\psi^{\text{adv.}}|\psi^{\text{leg.}}\rangle|^2$, where $|\psi^{\text{adv.}}\rangle$ and $|\psi^{\text{leg.}}\rangle$ denote the states that encode the legitimate and adversarial sample, respectively. In Fig. 7(b), we compute the average fidelity at each BIM iteration and plot the accuracy as a function of average fidelity. Since the fidelity basically measures the difference between the legitimate and adversarial images, and hence it is straightforward to obtain that the accuracy will decrease as the average fidelity decreases. This is explicitly demonstrated in Fig. 7(b). What is more interesting is that even when the accuracy decreases to zero, namely when all the adjusted images are misclassified, the average fidelity is still larger than 0.73. We mention that this is a fairly high average fidelity, given that the Hilbert space dimension of the quantum classifier is already very large.

TABLE I. Average fidelity ($\bar{F}$) and accuracy (in %) of the quantum classifier when being additively attacked by the BIM and FGSM methods in the white-box untargeted setting. For the two-category (four-category) classification, we use a model circuit of depth $p = 10$ ($p = 40$). For the BIM method, we generate adversarial examples using three iterations with a step size of 0.1. We denote such attack as BIM(3, 0.1). For the FGSM method, we generate adversarial examples using a single step with a step size of 0.03 (0.05) for the two-category (four-category) classifier. We denote such attacks as FGSM(1, 0.03) and FGSM(1, 0.05), respectively.

	Attacks	$\bar{F}$	Accuracy
Two-category	BIM (3, 0.1)	0.923	15.6%
	FGSM (1, 0.03)	0.901	00.0%
Four-category	BIM (3, 0.1)	0.943	23.7%
	FGSM (1, 0.05)	0.528	00.0%

In the above discussion, we have used Eq. (4), which is suitable for the untargeted attack, to generate adversarial examples. However, the problem we considered is a two-category classification problem and the distinction between targeted and untargeted attacks is ambiguous. A more unambiguous approach is to consider untargeted attacks to the four-category quantum classifier. Indeed, we have carried out such attacks and our results are plotted in Figs. 7(c) and 7(d), which are similar to the corresponding results for the two-category scenarios. Moreover, we can also consider utilizing different optimization methods to do white-box untargeted attacking for the quantum classifiers. In Table I, we summarize the performance of two different methods (BIM and FGSM) in attacking both the two-category and four-category quantum classifiers. Both the BIM and FGSM methods perform noticeably well.

Now, we have demonstrated how to obtained adversarial examples for the quantum classifiers by additive attacks, where each component of the data vectors are modified independently. In real experiments, to realize such adversarial examples with quantum devices might be challenging because this requires implementations of complicated global unitaries with very high precision. To this end, a more practical approach is to consider functional attacks, where the adversarial perturbation operators are implemented with a layer of local unitary transformations. In this case, the searching space is much smaller than that for the additive attacks, and hence we may not be able to find the most efficient adversarial perturbations. Yet, once we find the adversarial perturbations, it could be much easier to realize such perturbations in the quantum laboratory. To study functional attacks, in our numerical simulations we consider adding a layer of local unitary transformations before sending the quantum states to the classifiers. We restrict that these local unitaries are close to the identity operators so as to keep the perturbations reasonably small. We apply both the BIM and FGSM methods to solve Eq. (4) in the white-box untargeted setting. Partial of our results for the case of functional attacks are plotted in Fig. 8. From this figure, it is easy to see that the performances of both the BIM and FGSM methods are a bit poorer than that for the case of additive attacks. For instance, in the case of functional

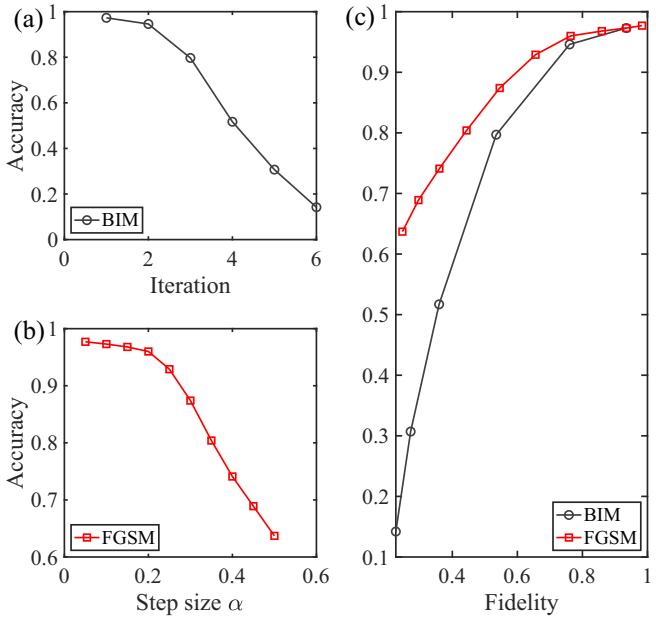

FIG. 8. Effects of adversarial untargeted functional attack on the accuracy of the quantum classifier for the problem of classifying handwritten digits 1 and 9. Here, the adversarial perturbation operators are assumed to be a layer of local unitary transformation. We use both the BIM method and the FGSM method to obtain adversarial examples. (a) For the BIM method, we generated adversarial perturbations using different number of iterations with the fixed step size 0.1. (b) For the FGSM method, we generate adversarial perturbations using different step sizes, and the accuracy drops accordingly with increasing step size.

attacks after six BIM iterations there is still a residue accuracy about 14% [see Fig. 8(a)], despite the fact that the average fidelity has already decreased to 0.2 [see Fig. 8(c)]. This is in sharp contrast to the case of additive attacks, where five BIM iterations are enough to reduce the accuracy down to zero [see Fig. 7(a)] and meanwhile maintain the average fidelity larger than 0.73 [see Fig. 7(b)]. The reduction of the performances for both methods is consistent with the fact that the searching space for functional attacks are much smaller than that for additive attacks.

2. *White-box attack: Targeted*

Unlike in the case of untargeted attacks, in targeted attacks the attacker attempts to mislead the classifier to classify a data sample incorrectly into a specific targeted category. A good example that manifestly showcases the importance of targeted attacks occurs in face recognition as well: In some situations the attacker may attempt to disguise her face inconspicuously to be recognized as an authorized user of a laptop or phone that authenticates users through face recognition. This type of attack has a particular name of impersonation attack in classical adversarial learning. It has been shown surprisingly in Ref. [115] that physically realizable and inconspicuous impersonation attacks can be carried out by wearing a pair of carefully crafted glasses designed for deceiving the state-of-the-art face recognition systems. In this subsection, we show

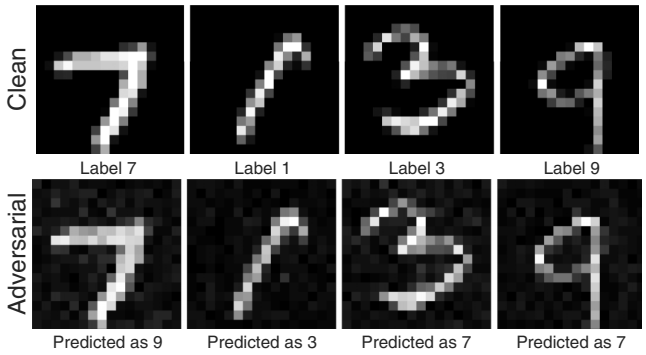

FIG. 9. Visual illustration of adversarial examples crafted using different attacks. From top to bottom: the clean and adversarial images generated for the quantum classifier by the BIM algorithm. By applying the additive attack, we can change the quantum classifier's classification result. The top images represent an correctly predicted legitimate example. The bottom images are incorrectly predicted adversarial example, even though they bear a close resemblance to the clean image. Here, the attacking algorithm we employed is BIM(0.1,3).

that quantum classifiers are likewise vulnerable to targeted attacks in the white-box setting.

We consider attacking the four-category quantum classifier. In Fig. 9, we randomly choose samples for digits 1, 3, 7, and 9 from MNIST and then solve the Eq. (5) iteratively by the BIM method to obtain their corresponding adversarial examples. This figure shows the original legitimate images and their corresponding targeted adversarial ones for the four-category quantum classifier. For these legitimate samples, the quantum classifier can assign their labels correctly with high confidence. But after targeted attacks, the same classifier is misled to classify the crafted images of digits $\{7, 1, 3, 9\}$ erroneously as the targeted digits $\{9, 3, 7, 7\}$ with a decent high confidence, despite the fact that the differences between the crafted and legitimate images are almost imperceptible. To further illustrate how this works, in Figs. 10(a)–10(d), we plot the classification probabilities for each digit and the loss functions with respect to particular digits as a function of the number of epochs. Here, we randomly choose an image of a given digit and then consider either additive [Figs. 10(a) and 10(b)] or functional [Figs. 10(c) and 10(d)] targeted attacks through the BIM method. For instance, in Fig. 10(a) the image we choose is an image for digit 1 and the targeted label is digit 3. From this figure, at the beginning the quantum classifier is able to correctly identify this image as digit 1 with probability $P(y = 1) \approx 0.41$. As the number of BIM iteration increases $P(y = 1)$ decreases and $P(y = 3)$ increases, and after about six iterations $P(y = 3)$ becomes larger than $P(y = 1)$, indicating that the classifier begins to be deceived into predict the image as a digit 3. Figure 10(b) shows the loss as a function of the number of epochs. From this figure, as the iteration number increases, the loss for classifying the image as digit 1 (3) increases (decreases), which is consistent with the classification probability behaviors in Fig. 10(a).

More surprisingly, we can in fact fool the quantum classifier to identify any images as a given targeted digit. This is clearly observed from Figs. 10(e) and 10(f) and Table II,

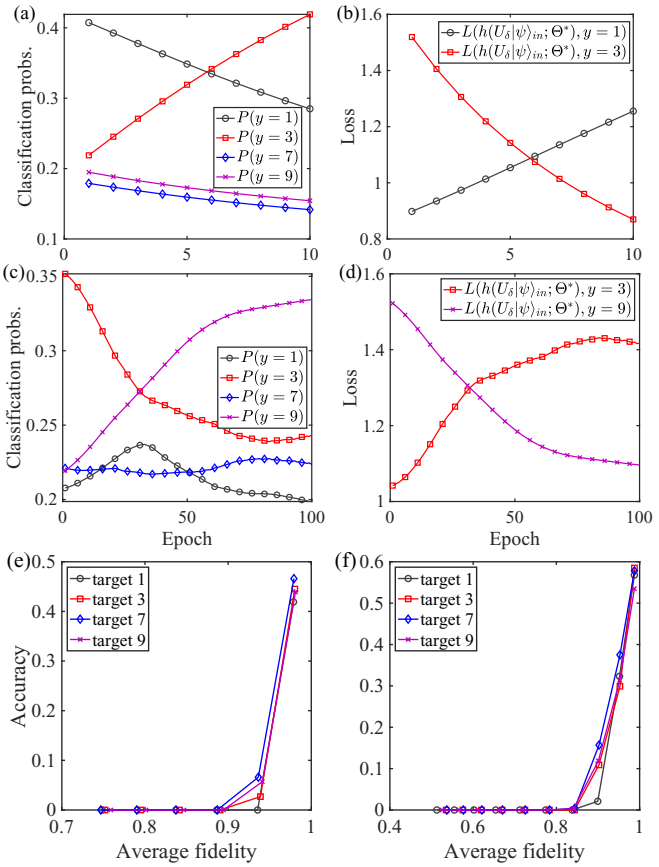

FIG. 10. White-box targeted attacks for the four-category quantum classifier with depth $p = 40$. (a) The classification probabilities for each digits as a function of the number of attacking epochs. Here, we use the BIM method to attack the quantum classifier. (b) The loss for classifying the image to be 1 or 3 as a function of the number of epochs. [(c), (d)] Similar plots for the functional attacks. [(e), (f)] The accuracy as a function of the average fidelity during the attacking process. Here, we consider additive attacks with both the BIM (e) and FGSM (f) methods.

where we perform additive attacks for all the images of digits $\{1, 3, 7, 9\}$ with different targeted labels and different attacking methods. In Figs. 10(e) and 10(f), we plot the accuracy versus the average fidelity. Here, for a given targeted label l ($l = 1, 3, 7,$ or, 9), we perform additive attacks for all images with original labels not equal to l and compute the accuracy and the average fidelity based on these images. From these figures, even when the average fidelity maintains larger than 0.85 the accuracy can indeed decrease to zero, indicating that all the images are classified by the quantum classifier incorrectly as digit l. In Table II, we summarize the performance of the BIM and FGSM methods in attacking the four-category quantum classifier in the white-box targeted setting.

TABLE II. The accuracy α^{adv} (in %) and average fidelity $\bar{F}$ for the four-category quantum classifier with depth $p = 10$ on the test data set when being attacked by different methods for different targeted labels. Here, we consider additive attacks with both the BIM and FGSM methods. For the BIM method, we generate adversarial examples using three iterations with a step size of 0.05, whereas for the FGSM method, we use a single step with step size of 0.03.

Attacks \ Targets		1	3	7	9
BIM(3, 0.05)	α^{adv}	5.7%	6.6%	2.7%	0.0%
	$\bar{F}$	0.941	0.936	0.938	0.935
FGSM(1, 0.03)	α^{adv}	2.1%	10.9%	15.7%	11.9%
	$\bar{F}$	0.899	0.902	0.902	0.901

TABLE III. Black-box attacks to the quantum classifier. Here, the adversarial examples are generated by three different methods (i.e., BIM, FGSM, and MIM) for two different classical classifiers, one based on CNN and the other on FNN (see the Appendix). This table shows the corresponding accuracy (in %) for each case on the MNIST test data set. We denote the predication accuracy of the classical neural networks (quantum classifier) on the test set as α_C (α_Q), and the predication accuracy on the adversarial test set as α_C^{adv} (α_Q^{adv}). The accuracy of the quantum classifier drops significantly on the adversarial examples generated for the classical neural networks.

Attacks \ Accuracy		α_C^{adv}	$\alpha_C - \alpha_C^{\text{adv}}$	α_Q^{adv}	$\alpha_Q - \alpha_Q^{\text{adv}}$
CNN	BIM (50, 0.01)	0.07%	98.2%	66.4%	25.6%
	FGSM (1, 0.3)	0.6%	98.3%	51.6%	40.4%
	MIM (10, 0.06)	0.7%	98.2%	62.3%	29.7%
FNN	BIM (50, 0.01)	0.6%	99.3%	68.1%	23.9%
	FGSM (1, 0.3)	1.0%	98.9%	56.8%	35.2%
	MIM (10, 0.06)	0.8%	99.1%	59.9%	32.1%

3. *Black-box attack: Transferability*

Unlike white-box attacks, black-box attacks assume limited or even no information about the internal structures of the classifiers and the learning algorithms. In classical adversarial learning, two basic premises that make black-box attacks possible have been actively studied [74]: the *transferability* of the adversarial examples and *probing* the behavior of the classifier. Adversarial sample transferability is the property that an adversarial example produced to deceive one specific learning model can deceive another different model, even if their architectures differ greatly or they are trained on different sets of training data [21,22,31], whereas probing is another important premise of the black-box attack that the attacker uses the victim model as an oracle to label a synthetic training set for training a substitute model, and hence the attacker needs not even collect a training set to mount the attack. Here, we study the transferability of adversarial examples in a more exotic setting, where we first generate adversarial examples for different classical classifiers and then investigate whether they transfer to the quantum classifiers or not. This would have important future applications considering a situation where the attacker may only have access to classical resources.

Our results are summarized in Table III. To obtain these results, we first train two classical classifiers, one based on a convolutional neural network (CNN) and the other based on a feedforward neural network (see the Appendix for details), with training data from the original MNIST dataset. Then we

use three different methods (i.e., BIM, FGSM, and MIM) to produce adversarial examples in a white-box untargeted setting for both classical classifiers separately. After these adversarial examples are obtained, we evaluate the performance of the trained quantum classifier on them. From Table III, it is evident that the performance of the quantum classifier on the adversarial examples is much worse than that on the original legitimate samples. For instance, for the adversarial examples generated for the CNN classifier by the MIM method, the accuracy of the quantum classifier is only 62.3%, which is 29.7% lower than that for the clean legitimate samples. This indicates roughly that 29.7% of the adversarial examples originally produced for attacking the CNN classifier transfer to the quantum classifier. This transferability ratio may not be as large as that for adversarial transferability between two classical classifiers. Yet, given the fact that the structure of the quantum classifier is completely different from the classical ones, it is in fact a bit surprising that such a high transferability ratio can be achieved in reality. We expect that if we use another quantum classifier to play as the surrogate classifier, the transferability ratio might increase significantly. We leave this interesting problem for future studies.

4. Adversarial perturbations are not random noises

The above discussions explicitly demonstrated the vulnerability of quantum classifiers against adversarial perturbations. The existence of adversarial examples is likewise a general property for quantum learning systems with high-dimensional Hilbert space. For almost all the images of hand-writing digits in MNIST, there always exists at least one corresponding adversarial example. Yet, it is worthwhile to clarify that adversarial perturbations are *not* random noises. They are carefully engineered to mislead the quantum classifiers and in fact only occupy a tiny subspace of the total Hilbert space. To demonstrate this more explicitly, we compare the effects of random noises on the accuracy of both two- and four-category quantum classifiers with the effects of adversarial perturbations. For simplicity and concreteness, we consider the uncorrelated decoherence noises that occur in a number of experimental platforms (such as Rydberg atoms, superconducting qubits, and trapped ions) for quantum computing [116–119]:

$$\mathcal{E}_{\text{depl}}(\rho) = (1-\beta)\rho + \frac{\beta}{3}(\sigma^x\rho\sigma^x + \sigma^y\rho\sigma^y + \sigma^z\rho\sigma^z), \quad (6)$$

where ρ denotes the density state of a qubit, $\sigma^{x,y,z}$ are the usual Pauli matrices, and $\beta \in [0, 1]$ is a positive number characterizing the strength of the decoherence noises.

In Fig. 11, we plot the classification accuracy of the quantum classifiers versus the noise strength p and the average fidelity between the original state and the state affected by a single layer of depolarizing noise on each qubit described by Eq. (6). From this figure, we observe that the accuracy for both the two- and four-category quantum classifiers decreases roughly linearly with the increase of p and the decrease of the average fidelity. This is in sharp contrast to the case for adversarial perturbations [see Figs. 10(e), 10(f), 8(c), 7(b), and 7(d) for comparison], where the accuracy has a dramatic reduction as the average fidelity begins to decrease from unity, indicating that the adversarial perturbations are not random

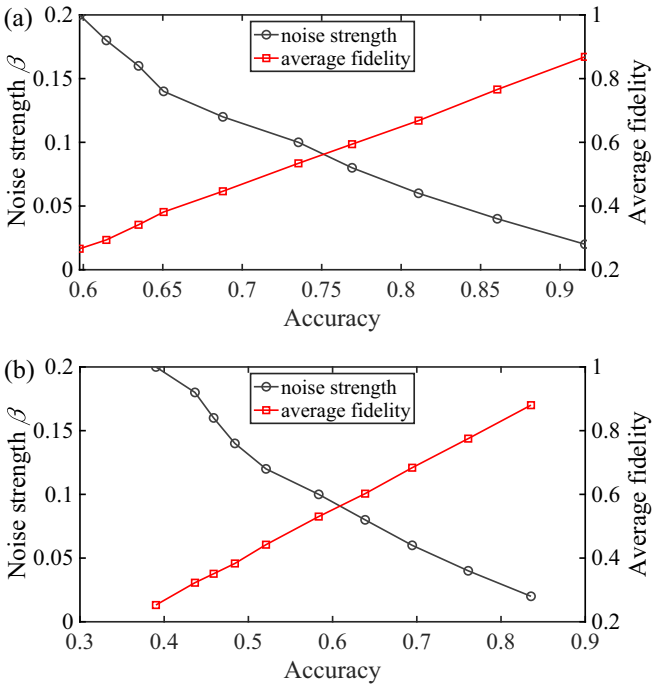

FIG. 11. Effects of depolarizing noises with varying strengths on the accuracy of the quantum classifiers with depth $p = 20$. The mean classification accuracy is computed on the test set with respect to the fidelity between the original input states and the states affected by depolarizing noises on each qubit with varying strengths. The accuracy and fidelity are averaged over 1000 random realizations. (a) Results for the two-category quantum classifier. (b) Results for the four-category quantum classifier.

noises. In fact, since the accuracy only decreases linearly with the average fidelity, this result also implies that quantum classifiers are actually rather robust to random noises. We mention that one may also consider the bit-flip or phase-flip noises and observe similar results. The fact that the adversarial perturbations are distinct from random noises is also reflected in our numerical simulations of the defense strategy by data augmentation—we find that the performance of the quantum classifier is noticeably better if we augment the training set by adversarial examples, rather than samples with random noises.

5. Larger models are more robust

Recently, it has been shown in the classical adversarial machine learning literature that increasing the capacity of the classifiers may enhance the robustness to adversarial perturbations [32]. This could be understood intuitively from the fact that the presence of adversarial examples will typically change the decision boundary of the problem to a more complicated one. Hence, a more complicated network might be needed to correctly classify the adversarial examples. Inspired by this, here we find similar observations for quantum classifiers: Increasing the capacity of the quantum classifiers may also improve their robustness.

For concreteness, we consider increasing the circuit depth (the number of layers p) to increase the capacity of our quantum classifiers. We choose the two- and four-category classifiers in the white-box untargeted setting as an example.

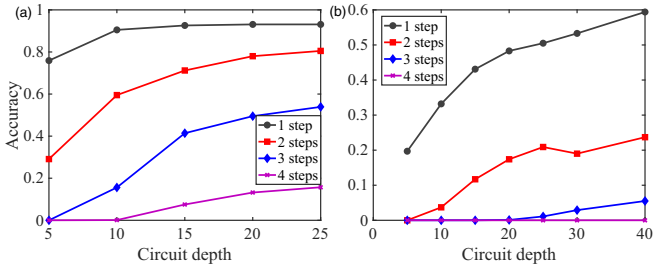

FIG. 12. Increasing the capacity of quantum classifiers will enhance their robustness to adversarial perturbations. Here, we first train two different quantum classifiers with the same fixed learning rate 0.005 and training algorithm (Adam), but different model capacity by varying the circuit depth (the number of layers p in Fig. 2). Then we use the same attacking algorithm (BIM with step size 0.1) to attack these trained models in the white-box untargeted setting. We find that the predication accuracy increases as the circuit depth increases for both (a) the two-category classification for images of digits 1 and 9, and (b) the four-category classification for images of digits 1, 3, 7, and 9.

To make a fair comparison, we train both models with varying circuit depths using the same learning rate, batch size, and optimizer. After training, we use the same attacking method, namely the BIM algorithm, to generate adversarial examples. Our results are shown in Fig. 12. From this figure, it is evident that for fixed BIM iteration steps, the accuracy for both classifiers increases as the circuit depth increases, implying a strengthening of their robustness to adversarial perturbations.

C. Quantum adversarial learning topological phases of matter

Classifying different phases and the transitions between them is one of the central problems in condensed matter physics. Recently, various machine learning tools and techniques have been adopted to tackle this intricate problem. In particular, a number of supervised and unsupervised learning methods have been introduced to classify phases of matter and identify phase transitions [8,10,120–130], giving rise to an emergent research frontier for machine learning phases of matter. Following these theoretical approaches, proof-of-principle experiments with different platforms [131–134], such as doped CuO_2 [134], electron spins in diamond nitrogen-vacancy centers [131], and cold atoms in optical lattices [132,133], have been carried out in laboratories to demonstrate their feasibility and unparalleled potentials. In addition, the vulnerability of these machine learning approaches to adversarial perturbations has been pointed out in a recent work as well [135]. It has been shown that typical phase classifiers based on classical deep neural networks are extremely vulnerable to adversarial attacks: Adding a tiny amount of carefully crafted noises or even just changing a single pixel of the legitimate sample may cause the classifier to make erroneous predictions with a surprisingly high confidence level.

Despite these exciting progresses made in the area of machine learning phases of matter, most previous approaches are based on classical classifiers and using quantum classifiers to classify different phases and transitions still remains barely explored. Here, in this section we study the problem of using quantum classifiers to classify different phases of matter, with a focus on topological phases that are widely believed to be more challenging than conventional symmetry-breaking phases (such as the paramagnetic and ferromagnetic phases) for machine-learning approaches [120,128,129,136]. We show, through a concrete example, that the quantum classifiers are likewise vulnerable to adversarial perturbations. We consider the following 2D square-lattice model for quantum anomalous Hall (QAH) effect, where a combination of spontaneous magnetization and spin-orbit coupling leads to quantized Hall conductivity in the absence of an external magnetic field:

$$\begin{aligned} H_{\mathrm{QAH}} = {} & J_{\mathrm{SO}}^{(x)} \sum_{\mathbf{r}} [(c_{\mathbf{r}\uparrow}^{\dagger} c_{\mathbf{r}+\hat{x}\downarrow} - c_{\mathbf{r}\uparrow}^{\dagger} c_{\mathbf{r}-\hat{x}\downarrow}) + \mathrm{H.c.}] \\ & + i J_{\mathrm{SO}}^{(y)} \sum_{\mathbf{r}} [(c_{\mathbf{r}\uparrow}^{\dagger} c_{\mathbf{r}+\hat{y}\downarrow} - c_{\mathbf{r}\uparrow}^{\dagger} c_{\mathbf{r}-\hat{y}\downarrow}) + \mathrm{H.c.}] \\ & - t \sum_{\langle \mathbf{r},\mathbf{s} \rangle} (c_{\mathbf{r}\uparrow}^{\dagger} c_{\mathbf{s}\uparrow} - c_{\mathbf{r}\downarrow}^{\dagger} c_{\mathbf{s}\downarrow}) + \mu \sum_{\mathbf{r}} (c_{\mathbf{r}\uparrow}^{\dagger} c_{\mathbf{r}\uparrow} - c_{\mathbf{r}\downarrow}^{\dagger} c_{\mathbf{r}\downarrow}). \end{aligned} \tag{7}$$

Here $c_{\mathbf{r}\sigma}^{\dagger}$ ($c_{\mathbf{r}\sigma}$) is the fermionic creation (annihilation) operator with pseudospin $\sigma = (\uparrow, \downarrow)$ at site $\mathbf{r}$, and $\hat{x}, \hat{y}$ are unit lattice vectors along the x, y directions. The first two terms describe the spin-orbit coupling with $J_{\mathrm{SO}}^{(x)}$ and $J_{\mathrm{SO}}^{(y)}$ denoting its strength along the x and y directions, respectively. The third and the fourth terms denote respectively the spin-conserved nearest-neighbor hopping and the on-site Zeeman interaction. In momentum space, this Hamiltonian has two Bloch bands and the topological structure of this model can be characterized by the first Chern number:

$$C_1 = -\frac{1}{2\pi} \int_{\mathrm{BZ}} dk_x dk_y F_{xy}(\mathbf{k}), \tag{8}$$

where F_{xy} denotes the Berry curvature $F_{xy}(\mathbf{k}) \equiv \partial_{k_x} A_y(\mathbf{k}) - \partial_{k_y} A_x(\mathbf{k})$ with the Berry connection $A_\mu(\mathbf{k}) \equiv \langle \varphi(\mathbf{k}) | i \partial_{k_\mu} | \varphi(\mathbf{k}) \rangle$ [$\mu = x, y$ and $\varphi(\mathbf{k})$ is the Bloch wave function of the lower band], and the integration is over the whole first Brillouin zone (BZ). It is straightforward to obtain that $C_1 = -\mathrm{sgn}(\mu)$ when $0 < |\mu| < 4t$ and $C_1 = 0$ otherwise.

The above Hamiltonian can be implemented with synthetic spin-orbit couplings in cold-atom experiment [137] and the topological index C_1 can be obtained from the standard time-of-flight images [138,139]. Indeed, by using ultracold fermionic atoms in a periodically modulated optical honeycomb lattice, the experimental realization of the Haldane model, which bears similar physics and Hamiltonian structures as in Eq. (8), has been reported [140]. For our purpose, we first train a two-category quantum classifier to assign labels of $C_1 = 0$ or $C_1 = 1$ to the time-of-flight images. To obtain the training data, we diagonalize the Hamiltonian in Eq. (7) with an open boundary condition and calculate the atomic density distributions with different spin bases for the lower band. These density distributions can be directly measured through the time-of-flight imaging techniques in cold atom experiments and serve as our input data. We vary λ_{SO} and t in both the topological and topologically trivial regions to generate several thousand data samples. As in the

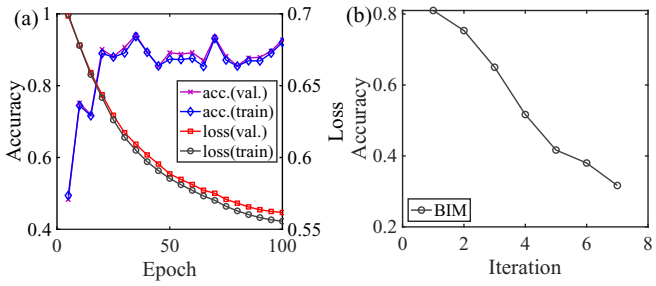

FIG. 13. (a) The average accuracy and loss for the two-category quantum classifier as a function of the number of epochs. Here, we use a quantum classifier with structures shown in Fig. 2 and depth 10 ($p = 10$) to perform binary classification for topological and nontopological phases. To train the classifier, we use the Adam optimizer with a batch size of 512 and a learning rate of 0.005 to minimize the loss function in Eq. (2). The accuracy and loss are averaged on 19 956 training samples and 6652 validation samples. (b) The accuracy of the quantum classifier as a function of the iterations of the BIM attack. Here, the BIM step size is 0.01.

above discussion on identifying images of hand-writing digits, we use amplitude encoding to convert the data for density distributions to the input quantum states for the quantum classifier. In Fig. 13(a), we plot the average accuracy and loss as a function of the number of epochs. It shows that after training, the quantum classifier can successfully identify the time-of-flight images with reasonably high accuracy. Yet, we note that this accuracy is a bit lower than that for the case of classifying paramagnetic and ferromagnetic phases discussed in the next section, which is consistent with the general belief that topological phases are harder to learning.

Unlike the conventional phases or the hand-writing digit images, topological phases are described by nonlocal topological invariants (such as the first Chern number), rather than local order parameters. Thus, intuitively the obtaining of adversarial examples might also be more challenging, since the topological invariants capture only the global properties of the systems and are insensitive to local perturbations. Yet, here we show that adversarial examples do exist in this case and the quantum classifier is indeed vulnerable in learning topological phases. To obtain adversarial examples, we consider attacking the quantum classifier additively in the white-box untargeted setting. Partial of our results are plotted in Fig. 13(b). From this figure, the accuracy for the quantum classifier in classifying time-of-flight images decreases rapidly as the number of attacking iterations increases and after about six iterations it becomes less than 0.4, indicating that more than 60% the attacked images in the test set are misclassified. To illustrate this even more concretely, in Fig. 14 we randomly choose a time-of-flight image and then solve the Eq. (4) iteratively by the BIM method to obtain its corresponding adversarial examples. Again, as shown in this figure the obtained adversarial example looks like the same as the clean legitimate time-of-flight image. They differ only by a tiny amount of perturbation that is imperceptible to human eyes. In addition, we summarize the performance of two different methods (BIM and FGSM) in attacking the quantum classifier in Table IV. Both the BIM and FGSM methods perform noticeably well.

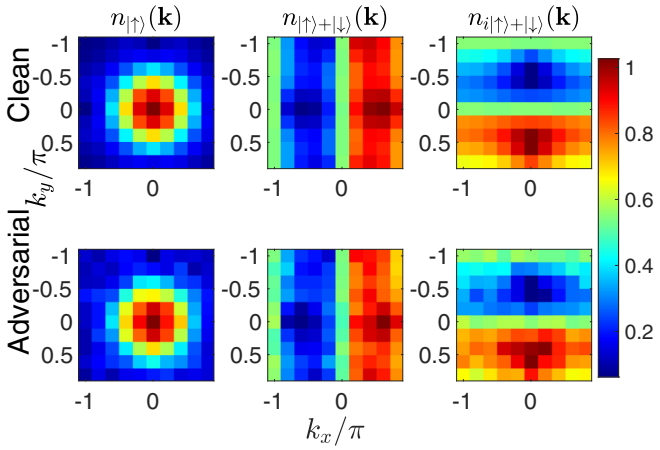

FIG. 14. The clean and the corresponding adversarial time-of-flight images for using the quantum classifier to classify topological phases. (Top) A legitimate sample of the density distribution in momentum space for the lower band with lattice size 10×10. (Bottom) An adversarial example obtained by the fast gradient sign method, which only differs with the original one by a tiny amount of noise that is imperceptible to human eyes.

D. Adversarial learning quantum data

In the above discussion, we considered using quantum classifiers to classify classical data (images) and studied their vulnerabilities to adversarial perturbations. This may have important applications in solving practical machine learning problems in our daily life. However, in such a scenario a prerequisite is to first transfer classical data to quantum states, which may require certain costly processes or techniques (such as quantum random access memories [90]) and thus renders the potential quantum speedups nullified [91]. Unlike classical classifiers that can only take classical data as input, quantum classifiers can also classify directly quantum states produced by quantum devices. Indeed, it has been shown that certain quantum classifiers, such as quantum principal component analysis [141] and quantum support vector machine [65], could offer an exponential speedup over their classical counterparts in classifying quantum data directly. In this subsection, we consider the vulnerability of quantum classifiers in classifying quantum states.

For simplicity and concreteness, we consider the following 1D transverse field Ising model:

$$H_{\text{Ising}} = -\sum_{i=1}^{L-1} \sigma_i^z \sigma_{i+1}^z - J_x \sum_{i=1}^{L} \sigma_i^x, \tag{9}$$

TABLE IV. Average fidelity $\bar{F}$ and accuracy (in %) of the two-category quantum classifier with depth $p = 10$ when being attacked by the BIM and FGSM methods in the white-box untargeted setting. Here, the accuracy and fidelity are averaged over 2000 testing samples.

Attacks	$\bar{F}$	Accuracy
BIM (3, 0.01)	0.988	31.6%
FGSM (1, 0.03)	0.952	6.3%

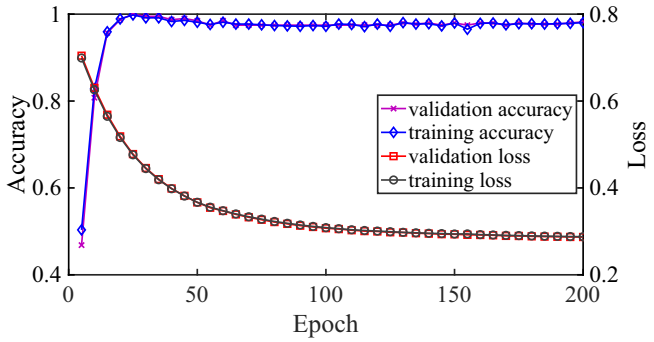

FIG. 15. The average accuracy and loss function as a function of the number of training steps. We use a depth-10 quantum classifier with structures shown in Fig. 2 to classify the ferromagnetic and paramagnetic phases for the ground states of H_{Ising}. We plot the accuracy of 1182 training samples and 395 validation samples (which are not in the training data set). We present the results of the first 200 iteration epochs. The learning rate is 0.005. The difference between the training loss and validation loss is very small, indicating that the quantum classifier does not overfit. The final accuracy on the 395 test samples is roughly 98%.

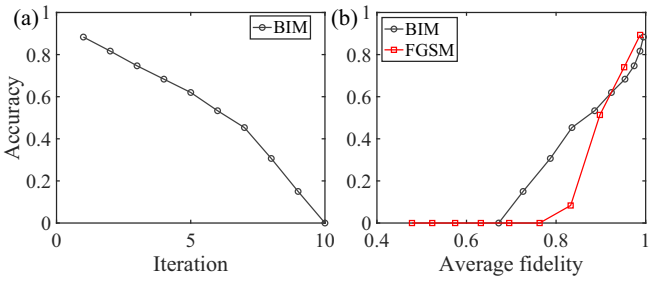

FIG. 16. Effect of additive adversarial attack on the accuracy of the two-category quantum classifier in classifying the ferromagnetic and paramagnetic phases for the ground states of the transverse field Ising model. We use both the BIM and FGSM methods to generate adversarial examples in the white-box untargeted setting. For the BIM method, we fix the step size to be 0.05 and the iteration number to be 10. For the FGSM method, we perform the attack using a single step but with step size ranging from 0.1 to 1.0. The circuit depth of the quantum classifier being attacked is $p = 10$ and the system size for the Ising model is $L = 8$. (a) The results for the BIM attack. (b) The accuracy as a function of average fidelity between the legitimate and adversarial samples for both the BIM and FGSM methods.

where σ_i^z and σ_i^x are the usual Pauli matrices acting on the ith spin and J_x is a positive parameter describing the strength of the transverse field. This model maps to free fermions through a Jordan-Wigner transformation and is exactly solvable. At zero temperature, it features a well-understood quantum phase transition at $J_x = 1$, between a paramagnetic phase for $J_x > 1$ and a ferromagnetic phase for $J_x < 1$. It is an exemplary toy model for studying quantum phase transitions and an excellent test bed for different new methods and techniques. Here, we use a quantum classifier, with structures shown in Fig. 2, to classify the ground states of H_{Ising} with varying J_x (from $J_x = 0$ to $J_x = 2$) and show that this approach is extremely vulnerable to adversarial perturbations as well.

To generate the data sets for training, validation, and testing, we sample a series of Hamiltonians with varying J_x from 0 to 2 and calculating their corresponding ground states, which are used as input data to the quantum classifier. We train the quantum classifier with the generated training data set and our results for training are shown in Fig. 15. Strikingly, our quantum classifier is very efficient in classifying these ground states of H_{Ising} into categories of paramagnetic and ferromagnetic phases and we find that a model circuit with depth $p = 5$ is enough to achieve near-perfect classification accuracy. This is in contrast to the case of learning topological phases, where a quantum classifier with depth $p = 10$ only gives an accuracy of around 90%. In addition, we mention that one can also use the quantum classifier to study the quantum phase transition.

Similar to the cases for classical input data, the quantum classifiers are vulnerable to adversarial perturbations in classifying quantum data as well. To show this more explicitly, we consider attacking the above quantum classifier trained with quantum inputs additively in the white-box untargeted setting. Partial of our results are plotted in Fig. 16. In Fig. 16(a), we plot the accuracy as a function of the number of the BIM iterations and find that it decreases to zero after 10 BIM iterations, indicating that all the slightly adjusted quantum states, including even these far away from the phase transition point, are misclassified by the quantum classifier. In Fig. 16(b), we plot the accuracy as a function of averaged fidelity for different attacking methods. From this figure, both the BIM and FGSM methods are notably effective in this scenario and the accuracy of the quantum classifier on the generated adversarial examples decreases to zero, whereas the average fidelity maintains moderately large for both methods.

IV. DEFENSE: QUANTUM ADVERSARIAL TRAINING

In the above discussions, we have explicitly shown that quantum classifiers are vulnerable to adversarial perturbations. This may raise serious concerns about the reliability and security of quantum learning systems, especially for these applications that are safety and security critical, such as self-driving cars and biometric authentications. Thus, it is of both fundamental and practical importance to study possible defense strategies to increase the robustness of quantum classifiers to adversarial perturbations [142].

In general, adversarial examples are hard to defend against because of the following two reasons. First, it is difficult to build a precise theoretical model for the adversarial example crafting process. This is a highly nonlinear and nonconvex sophisticated optimization process and we lack proper theoretical tools to analyze this process, making it notoriously hard to obtain any theoretical argument that a particular defense strategy will rule out a set of adversarial examples. Second, defending adversarial examples requires the learning system to produce proper outputs for every possible input, the number of which typically scales exponentially with the size of the problem. Most of the time, the machine learning models work very well but only for a very small ratio of all the possible inputs. Nevertheless, in the field of classical adversarial machine learning, a variety of defense strategies have been proposed

in recent years to mitigate the effect of adversarial attacks, including adversarial training [77], gradient hiding [143], defensive distillation [80], and defense-GAN [78], etc. Each of these strategies has its own advantages and disadvantages and none of them is adaptive to all types of adversarial attacks. In this section, we study the problem of how to increase the robustness of quantum classifiers against adversarial perturbations. We adopt one of the simplest and effective methods, namely adversarial training, to the case of quantum learning and show that it can significantly enhance the performance of quantum classifiers in defending adversarial attacks.

The basic idea of adversarial training is to strengthen model robustness by injecting adversarial examples into the training set. It is a straightforward brute force approach where one simply generates a lot of adversarial examples using one or more chosen attacking strategies and then retrain the classifier with both the legitimate and adversarial samples. For our purpose, we employ a *robust optimization* [144] approach and reduce the task to solving a typical min-max optimization problem:

$$\min_{\Theta} \frac{1}{N} \sum_{i=1}^{N} \max_{U_\delta \in \Delta} L\big(h\big(U_\delta |\psi\rangle_{\text{in}}^{(i)}; \Theta\big), y^{(i)}\big), \tag{10}$$

where $|\psi\rangle_{\text{in}}^{(i)}$ is the ith sample under attack and $y^{(i)}$ denotes its original corresponding label. The meaning of Eq. (10) is clear: We are training the quantum classifier to minimize the adversarial risk, which is described by the average loss for the worst-case perturbations of the input samples. We mention that this min-max formulation has already been extensively studied in the field of robust optimization and many methods for solving such min-max problems have been developed [144]. One efficient method is to split Eq. (10) into two parts: the outer minimization and the inner maximization. The inner maximization problem is exactly the same problem of generating adversarial perturbations, which have discussed in detail in Secs. II and III. The outer minimization task boils down to a task of minimizing the loss function on adversarial examples. With this in mind, we develop a three-step procedure to solve the total optimization problem. In the first step, we randomly choose a batch of input samples $|\psi\rangle_{\text{in}}^{(i)}$ together with their corresponding labels $y^{(i)}$. Then, we calculate the "worst-case" perturbation of $|\psi\rangle_{\text{in}}^{(i)}$ with respect to the current model parameters Θ_t, that is, to solve $U_{\delta^*} = \text{argmax}_{U_\delta \in \Delta} L(h(U_\delta|\psi\rangle; \Theta), y^{(i)})$. In the third step, we update the parameters Θ_t according to the minimization problem at $U_{\delta^*}|\psi\rangle_{\text{in}}$: $\Theta_{t+1} = \Theta_t - \eta \nabla_\Theta L(h(U_{\delta^*}|\psi\rangle_{\text{in}}^{(i)}; \Theta_t), y^{(i)})$. We repeat these three steps until the accuracy converges to a reasonable value.

Partial results are shown in Fig. 17. In this figure, we consider the adversarial training of a quantum classifier in identifying handwritten digits in MNIST. We use the BIM method in the white-box untargeted setting to generate adversarial examples. We use 20 000 clean images and generate their corresponding adversarial images. The clean images and the adversarial ones together form the training data set, and another 2000 images are used for the testing. From this figure, it is evident that, after adversarial training, the accuracy of the quantum classifier for both the adversarial samples and legitimate samples increases significantly. At the beginning of the training, the accuracy for the adversarial samples in the testing set remains zero. This is because the initial model parameters are randomly chosen, so the quantum classifier does not learn enough information and its performance on even legitimate samples is still very poor at the beginning (hence for each sample it is always possible to find an adversarial example by the BIM method, resulting in a zero accuracy on the testing set of adversarial examples). After the early stage of the adversarial training, this accuracy begins to increase rapidly and the quantum classifier is able to classify more crafted samples correctly. In other words, the BIM attack becomes less effective on more samples. At the end of the training, the accuracies for both the legitimate and adversarial data sets converge to a saturated value larger than 98%, indicating that the adversarially retrained quantum classifier is immune to the adversarial examples generated by the BIM attack. We also notice that, due to the competition between the inner maximization and outer minimization, the accuracies for the legitimate data sets for training and validation both have an oscillation at the beginning of the adversarial training process.

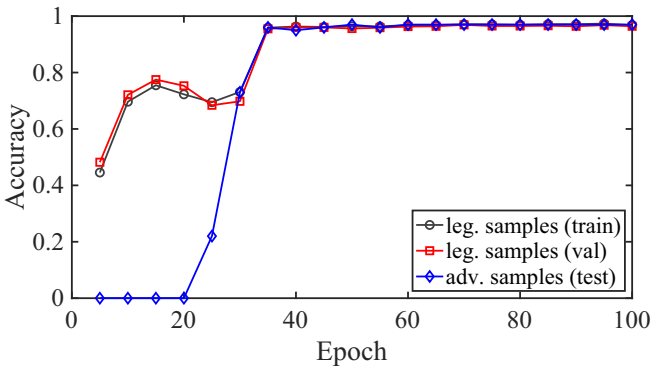

FIG. 17. Strengthening the robustness of the quantum classifier against adversarial perturbations by quantum adversarial training. In each epoch, we first generate adequate adversarial examples with the BIM method for the quantum classifier with the current model parameters. The iteration number is set to be three and the BIM step size is set to be 0.05. Then, we train the quantum classifier with both the legitimate and crafted samples. The circuit depth of the quantum classifier is 10 and the learning rate is set to be 0.005.

The above example explicitly shows that adversarial training can indeed increase the robustness of quantum classifiers against a certain type of adversarial perturbations. Yet, it is worthwhile to mention that the adversarially trained quantum classifier may only perform well on adversarial examples that are generated by the same attacking method. It does not perform as well when a different attack strategy is used by the attacker. In addition, adversarial training tends to make the quantum classifier more robust to white-box attacks than to black-box attacks due to gradient masking [31,143]. In fact, we expect *no* universal defense strategy that is adaptive to all types of adversarial attacks, as one approach may block one kind of attack for the quantum classifier but will inevitably leave another vulnerability open to an attacker who knows and makes use of the underlying defense mechanism. In

the field of classical adversarial learning, another intriguing defense mechanism that is effective against both white-box and black-box attacks has been proposed recently [78]. This strategy is called defense-GAN, which leverages the representative power of GAN to diminish the effect of adversarial perturbations via projecting input data onto the range of the GAN's generator before feeding it to the classifier. More recently, a quantum version of GAN (dubbed QGAN) has been theoretically proposed [43,44] and a proof-of-principle experimental realization of QGAN has been reported with superconducting quantum circuits [62]. Likewise, it would be interesting and important to develop a defense-QGAN strategy to enhance the robustness of quantum classifiers against adversarial perturbations. We leave this interesting topic for future study.

V. CONCLUSION AND OUTLOOK

In summary, we have systematically studied the vulnerability of quantum classifiers to adversarial examples in different scenarios. We found that, similar to classical classifiers based on deep neural networks, quantum classifiers are likewise extremely vulnerable to adversarial attacks: Adding a tiny amount of carefully crafted perturbations, which are imperceptible to human eyes or ineffective to conventional methods, into the original legitimate data (either classical or quantum mechanical) will cause the quantum classifiers to make incorrect predictions with a notably high confidence level. We introduced a generic recipe on how to generate adversarial perturbations for quantum classifiers with different attacking methods and gave three concrete examples in different adversarial settings, including classifying real-life handwritten digit images in MNIST, simulated time-of-flight images for topological phases of matter, and quantum ground states for studying the paramagnetic to ferromagnetic quantum phase transition. In addition, through adversarial training, we have shown that the vulnerability of quantum classifiers to specific types of adversarial perturbations can be significantly suppressed. Our discussion is mainly focused on supervised learning based on quantum circuit classifiers, but its generalizations to the case of unsupervised learning and other types of quantum classifiers are possible and straightforward. Our results reveal a vulnerability aspect for quantum machine learning systems to adversarial perturbations, which would be crucial for practical applications of quantum classifiers in the realms of both artificial intelligence and machine learning phases of matter.

It is worthwhile to clarify the differences between the quantum adversarial learning discussed in this paper and the quantum generative adversarial networks (QGAN) studied in previous works [43,44,46,62,145]. A QGAN contains two major components, a generator and a discriminator, which are trained alternatively in the way of an adversarial game: At each learning round, the discriminator optimizes her strategies to identify the fake data produced by the generator, whereas the generator updates his strategies to fool the discriminator. At the end of the training, such an adversarial procedure will end up at a Nash equilibrium point, where the generator produces data that match the statistics of the true data from the original training set and the discriminator can no longer distinguish the fake data with a probability larger than one half. The major goal of QGAN is to produce new data (either classical or quantum mechanical) that match the statistics of the training data, rather than to generate adversarial examples that are endowed with wild patterns.

This work only reveals the tip of the iceberg. Many important questions remain unexplored and deserve further investigations. First, the existence of adversarial examples seems to be a fundamental feature of quantum machine learning applications in high-dimensional spaces [67] due to the concentration of measure phenomenon [146]. Thus, we expect that various machine learning approaches to a variety of high-dimensional problems, such as separability-entanglement classification [147,148], quantum state discrimination [149], quantum Hamiltonian learning [150], and quantum state tomography [7,151], should also be vulnerable to adversarial attacks. Yet, in practice how to find out all possible adversarial perturbations in these scenarios and develop appropriate countermeasures feasible in experiments to strengthen the reliability of these approaches still remains unclear. Second, in classical adversarial learning a strong "no free lunch" theorem has been established recently [39–41], which shows that there exists an intrinsic tension between adversarial robustness and generalization accuracy. In the future, it would be interesting and important to prove a quantum version of such a profound theorem and study its implications in practical applications of quantum technologies. Third, the adversarial perturbations obtained in this paper are dependent on input state. Yet, in real experiments the input states might be inaccessible in certain circumstances. Thus, it would be interesting and important to study whether there exist *universal* perturbations for quantum classifiers that could make most of the input samples to be adversarial examples. In addition, there seems to be a deep connection between the existence of adversarial perturbations in quantum deep learning and the phenomenon of orthogonality catastrophe in quantum many-body physics [152,153], where adding a weak local perturbation into a metallic or many-body localized Hamiltonian will make the ground state of the slightly modified Hamiltonian orthogonal to that of the original one in the thermodynamic limit. A thorough investigation of this will provide insight into the understanding of both adversarial learning and orthogonality catastrophe. Finally, an experimental demonstration of quantum adversarial learning should be a crucial step toward practical applications of quantum technologies in artificial intelligence in the future.

ACKNOWLEDGMENTS

We thank Nana Liu, Peter Wittek, Ignacio Cirac, Roger Colbeck, Yi Zhang, Xiaopeng Li, Christopher Monroe, Juan Carrasquilla, Peter Zoller, Rainer Blatt, John Preskill, Zico Kolter, Alán Aspuru-Guzik, and Peter Shor for helpful discussions. S.L. would like to further thank Mucong Ding, Weikang Li, Roger Luo, and Jin-Guo Liu for their help in developing the code for implementing the adversarial machine learning process. This work was supported by the Frontier Science Center for Quantum Information of the Ministry of Education of China, Tsinghua University Initiative Scientific Research

Program, and the National Key Research and Development Program of China (2016YFA0301902). D.-L.D. acknowledges in addition the support from the National Thousand-Young-Talents Program and the startup fund from Tsinghua University (Grant No. 53330300319).

APPENDIX: ATTACK ALGORITHMS

As mentioned in the main text, the type of attacks we consider is mainly evasion attack from the perspective of attack surface. Evasion attack is the most common type of attack in classical adversarial learning [25]. In this setting, the attacker attempts to deceive the classifier by adjusting malicious samples during the testing phase. This setting assumes no modification of the training data, which is in sharp contrast to poisoning attack, where the adversary tries to poison the training data by injecting carefully crafted samples to compromise the whole learning process. Within the evasion-attack umbrella, the attacks considered in this paper can be further categorized into additive or functional, targeted or untargeted, and white-box or black-box attacks along different classification dimensions. Here, in this Appendix, we give more technique details about the attack algorithms used.

1. White-box attacks

White-box attacks assume full information about the classifier, so the attacker can exploit the gradient of the loss function: $\nabla_{\mathbf{x}} L(h(\mathbf{x}+\delta;\theta), y)$. For the convenience and conciseness of the presentation, we will use $\mathbf{x}$ (y) and $|\psi\rangle_{\text{in}}$ ($\mathbf{a}$) interchangeably to represent the input data (corresponding label) throughout the whole Appendix. Based on the information of gradients, a number of methods have been proposed in the classical adversarial learning community to generate adversarial samples. In this work, we adopt some of these methods to the quantum setting, including the FGSM, BIM, and PGD methods. In the following, we introduce these methods one by one and provide a pseudocode for each method.

a. Quantum-adapted FGSM method (Q-FGSM).

The FGSM method is a simple one-step scheme for obtaining adversarial examples and has been widely used in the classical adversarial machine learning community [22,32]. It calculates the gradient of the loss function with respect to the input of the classifier. The adversarial examples are generated using the following equation:

$$\mathbf{x}^* = \mathbf{x} + \epsilon \cdot \text{sgn}(\nabla_{\mathbf{x}} L(h(|\psi\rangle_{\text{in}}; \Theta^*), \mathbf{a})), \tag{A1}$$

where $L(h(|\psi\rangle_{\text{in}}; \Theta^*), \mathbf{a})$ is the loss function of the trained quantum classifier, ϵ is the perturbation bound, $\nabla_{\mathbf{x}}$ denotes the gradient of the loss with respect to a legitimate sample $\mathbf{x}$ with correct label $\mathbf{a}$, and $\mathbf{x}^*$ denotes the generated adversarial example corresponding to $\mathbf{x}$. For the case of additive attacks, where we modify each component of the data vector independently, $\nabla_{\mathbf{x}}$ is computed componentwise and a normalization of the data vector will be performed if necessary. For the case of functional attacks, we use a layer of parametrized local unitaries to implement the perturbations to the input data $|\psi\rangle_{\text{in}}$. In this case, $\nabla_{\mathbf{x}}$ is implemented via the gradient of the loss with respect to the parameters defining the local unitaries.

Algorithm 1 Quantum-adopted fast gradient sign method

Input The trained quantum classifier h, loss function L, the legitimate sample $(|\psi\rangle_{\text{in}}, \mathbf{a})$.
Input The perturbation bound ϵ
Output An adversarial example $\mathbf{x}^*$.
Input $|\psi\rangle_{\text{in}}$ into F to obtain $\nabla_x L(h(|\psi\rangle; \Theta^*), \mathbf{a})$
for Every component x_i of $|\psi\rangle_{\text{in}}$ **do**
  $\delta_i = \epsilon \cdot \text{sign}(\nabla_{x_i} L(h(|\psi\rangle_{\text{in}}; \Theta^*), \mathbf{a})$
  $x_i^* = x_i + \delta_i$
end for
return $\mathbf{x}^*$ or its equivalent $|\psi\rangle^*$

Equation (A1) should be understood as

$$\omega^* = \epsilon \cdot \text{sgn}(\nabla_{\omega} L(h(U(\omega)|\psi\rangle_{\text{in}}; \Theta^*), \mathbf{a})), \tag{A2}$$

$$|\psi\rangle_{\text{adv}} = U(\omega^*)|\psi\rangle_{\text{in}}, \tag{A3}$$

where ω denotes collectively all the parameters for the local unitaries. A pseudocode representation of the Q-FGSM algorithm for the case of additive attacks is shown in Algorithm 1. The pseudocode for the case of functional attacks is similar and straightforward, and thus has been omitted for brevity.

b. Quantum-adapted BIM method (Q-BIM)

The BIM method is a straightforward extension of the basic FGSM method [27]. It generates adversarial examples by iteratively applying the FGSM method with a small step size α:

$$\mathbf{x}_{k+1}^* = \pi_C[\mathbf{x}_k^* + \alpha \cdot \text{sgn}(\nabla_{\mathbf{x}} L(h(|\psi\rangle_k^*; \Theta^*), \mathbf{a}))], \tag{A4}$$

where $\mathbf{x}_k^*$ denotes the modified sample at step k and π_C is projection operator that normalizes the wave function. A pseudocode representation of the Q-BIM algorithm for the case of additive attacks is shown in Algorithm 2.

Algorithm 2 Quantum-adapted basic iterative method

Input The trained model h, loss function L, the legitimate sample $(|\psi\rangle_{\text{in}}, \mathbf{a})$.
Input The perturbation bound ϵ, iteration number T, decay factor μ, upper and lower bound $x_{\min}, x_{\max}$.
Output An adversarial example $|\psi\rangle^*$.
$|\psi\rangle_0^* = |\psi\rangle_{\text{in}}$
$\alpha = \frac{\epsilon}{T}$
for $k = 1, \ldots, T$ **do**
  Input $|\psi\rangle_{i-1}$ into F to obtain $\mathbf{b}_k = \nabla_x L(h(|\psi\rangle_{k-1}; \theta), \mathbf{a})$
  for Every component $(\mathbf{x}_k)_j$ of $|\psi\rangle_{k-1}^*$ **do**
    $\delta_j = \alpha \cdot \text{sgn}((\mathbf{b}_k)_j)$
    $(\mathbf{x}_k)_j = (\mathbf{x}_{k-1})_j + \delta_j$
  end for
  $(\mathbf{x}_k) = \pi_C(\mathbf{x}_k)$
end for
return $|\psi\rangle^* = |\psi\rangle_T$

2. Black-box attacks: Transfer attack

Unlike in the white-box setting, black-box attacks assume that the adversary does not have full information about either the model or the algorithm used by the learner. In particular, the adversary does not have the information about the loss function used by the quantum classifier, and thus cannot use the gradient-based attacking methods to generate adversarial examples. Yet, for simplicity we do assume that the attacker has access to a vast data set to train a local substitute classifier that approximates the decision boundary of the target classifier. Once the substitute classifier is trained with high confidence, any white-box attack strategy can be applied on it to generate adversarial examples, which can be used to deceive the target classifier due to the transferability property of adversarial examples. In this work, we consider the transfer attack in a more exotic setting, where we use different classical classifiers as the local substitute classifier to generate adversarial examples for the quantum classifier. The two classical classifiers are based on the CNN and FNN, respectively. In Table V, we show the detailed structures of the CNN and FNN. To train these two classical classifiers, we use the Adam optimizer [101] and a batch size of 256. The learning rate is set to be 10^{-3} during training. The corresponding learning process is implemented using Keras [154], a high-level deep learning library running on top of the TensorFlow framework [155]. After training, both the CNN and FNN classifiers achieve a remarkably high accuracy on the legitimate testing data set (98.9% and 99.9% respectively; see Table III in the main text).

We use three different methods, namely the BIM, FGSM, and MIM methods, to attack both the CNN and FNN classifiers in a white-box setting to obtain adversarial examples. These attacks are implemented by using of Cleverhans [157].

TABLE V. Model architectures for the classical neural networks. (a) The CNN architecture consists of three layers: a 2D convolution layer, an activational ReLu layer [156], and a fully connected flattening layer with 0.5 dropout regularization. The last layer is then connected to the final softmax classifier, which outputs the probability for each possible handwritten digit. In our case, we have four categories: 1, 3, 7, 9. (b) The feedforward neural network architecture consists of fully connected layers and dropout [110] layers with a dropping rate 0.1, which are important for avoiding overfitting.

Classifier based on CNN	Classifier based on FNN
Conv(64,8,8)+ReLu	FC(512)+ReLu
Conv(128,4,4)+ReLu	Dropout(0.1)
Conv(128,2,2)+ReLu	FC(53)+ReLu
Flatten	Dropout(0.1)
FC(4)+Softmax	FC(4)+Softmax

For the BIM attack, the number of attack iterations is set to be 10 and the step size α is set to be 0.01. For the FGSM attack, the number of iteration is one and the step size is set to be 0.3. For the MIM method, the number of attack iterations is set to be 10, the step size is set to be 0.06, and the decay factor μ is set to be 1.0. A detailed description of the MIM method, together with a pseudocode, can be find in Ref. [135]. The performance of both classifiers on the corresponding sets of adversarial examples is shown in Table III in the main text, from which it is clear that the attack is very effective (the accuracy for both classifiers decreases to a value less than 1%). After the adversarial examples were generated, we test the performance of the quantum classifiers on them and find that its accuracy decreases noticeably (see Table III in the main text).

[1] S. D. Sarma, D.-L. Deng, and L.-M. Duan, Machine learning meets quantum physics, Phys. Today **72**(3), 48 (2019).

[2] M. Jordan and T. Mitchell, Machine learning: Trends, perspectives, and prospects, Science **349**, 255 (2015).

[3] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, Mastering the game of go with deep neural networks and tree search, Nature (London) **529**, 484 (2016).

[5] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, Mastering the game of go without human knowledge, Nature (London) **550**, 354 (2017).

[6] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, 602 (2017).

[7] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, Nat. Phys. **14**, 447 (2018).

[8] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Machine Learning Phases of Strongly Correlated Fermions, Phys. Rev. X **7**, 031038 (2017).

[9] Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada, Restricted Boltzmann machine learning for solving strongly correlated quantum systems, Phys. Rev. B **96**, 205152 (2017).

[10] L. Wang, Discovering phase transitions with unsupervised learning, Phys. Rev. B **94**, 195105 (2016).

[11] Y.-Z. You, Z. Yang, and X.-L. Qi, Machine learning spatial geometry from entanglement features, Phys. Rev. B **97**, 045153 (2018).

[12] D.-L. Deng, X. Li, and S. Das Sarma, Machine learning topological states, Phys. Rev. B **96**, 195145 (2017).

[13] D.-L. Deng, Machine Learning Detection of Bell Nonlocality in Quantum Many-Body Systems, Phys. Rev. Lett. **120**, 240402 (2018).

[14] D.-L. Deng, X. Li, and S. Das Sarma, Quantum Entanglement in Neural Network States, Phys. Rev. X **7**, 021021 (2017).

[15] X. Gao and L.-M. Duan, Efficient representation of quantum many-body states with deep neural networks, Nat. Commun. **8**, 662 (2017).

[16] R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac, Restricted Boltzmann machines in quantum physics, Nat. Phys. **15**, 887 (2019).

[17] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[18] V. Dunjko and H. J. Briegel, Machine learning and artificial intelligence in the quantum domain: A review of recent progress, Rep. Prog. Phys. **81**, 074001 (2018).

[19] C. Ciliberto, M. Herbster, A. Davide Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Quantum machine learning: A classical perspective, Proc. R. Soc. London A **474**, 20170551 (2017).

[20] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, Adversarial machine learning, in *Proceedings of the Fourth ACM workshop on Security and Artificial Intelligence* (ACM, New York, 2011), pp. 43–58.

[21] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, Intriguing properties of neural networks, in *Second International Conference on Learning Representations* (ICLR, Banff, Canada, 2014).

[22] I. Goodfellow, J. Shlens, and C. Szegedy, Explaining and harnessing adversarial examples, in *Third International Conference on Learning Representations* (ICLR, San Diego, 2015).

[23] B. Biggio and F. Roli, Wild patterns: Ten years after the rise of adversarial machine learning, Pattern Recog. **84**, 317 (2018).

[24] D. J. Miller, Z. Xiang, and G. Kesidis, Adversarial learning in statistical classification: A comprehensive review of defenses against attacks, Proc. IEEE **108**, 402 (2020).

[25] Y. Vorobeychik and M. Kantarcioglu, *Adversarial Machine Learning*, Synthesis Lectures on Artificial Intelligence and Machine Learning (Morgan and Claypool, San Rafael, CA, 2018).

[26] N. Carlini and D. Wagner, Adversarial examples are not easily detected: Bypassing ten detection methods, in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17 (ACM, New York, 2017), pp. 3–14.

[27] A. Kurakin, I. J. Goodfellow, and S. Bengio, Adversarial examples in the physical world, in *Fifth International Conference on Learning Representations* (ICLR, Toulon, France, 2017).

[28] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, Black-box adversarial attacks with limited queries and information, in *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research Vol. 80, edited by J. Dy and A. Krause (PMLR, Stockholm, Sweden, 2018), pp. 2137–2146.

[29] V. Tjeng, K. Y. Xiao, and R. Tedrake, Evaluating robustness of neural networks with mixed integer programming, in *International Conference on Learning Representations* (ICLR, New Orleans, LA, 2019).

[30] A. Athalye, N. Carlini, and D. Wagner, Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, in *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research Vol. 80, edited by J. Dy and A. Krause (PMLR, Stockholm, Sweden, 2018), pp. 274–283.

[31] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, Practical black-box attacks against machine learning, in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17 (ACM, New York, 2017), pp. 506–519.

[32] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, Towards deep learning models resistant to adversarial attacks, in *Sixth International Conference on Learning Representations* (ICLR Vancouver, BC, Canada, 2018).

[33] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, Zoo: Zeroth-order optimization-based black-box attacks to deep neural networks without training substitute models, in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17 (ACM, New York, 2017), pp. 15–26.

[34] N. Dalvi, P. Domingos, S. Sanghai, and D. Verma, Adversarial classification, in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2004), pp. 99–108.

[35] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, Boosting adversarial attacks with momentum, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE Computer Society, Los Alamitos, CA, 2018), pp. 9185–9193.

[36] N. Carlini and D. Wagner, Towards evaluating the robustness of neural networks, in *2017 IEEE Symposium on Security and Privacy* (IEEE, San Jose, CA, USA, 2017), pp. 39–57.

[37] A. Nguyen, J. Yosinski, and J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Boston, MA, 2015), pp. 427–436.

[38] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, Deepfool: A simple and accurate method to fool deep neural networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Las Vegas, NV, 2016), pp. 2574–2582.

[39] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, Robustness may be at odds with accuracy, in *Seventh International Conference on Learning Representations* (ICLR, New Orleans, LA, 2019).

[40] A. Fawzi, H. Fawzi, and O. Fawzi, Adversarial vulnerability for any classifier, in *32nd Conference on Neural Information Processing Systems* (NeurIPS, Montreal, Canada, 2018), pp. 1178–1187.

[41] J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. J. Goodfellow, Adversarial spheres, in *Sixth International Conference on Learning Representations* (ICLR, Vancouver, BC, Canada, 2018).

[42] E. Dohmatob, Generalized no free lunch theorem for adversarial robustness, in *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research Vol. 97, edited by K. Chaudhuri and R. Salakhutdinov (PMLR, Long Beach, CA, 2019), pp. 1646–1654.

[43] S. Lloyd and C. Weedbrook, Quantum Generative Adversarial Learning, Phys. Rev. Lett. **121**, 040502 (2018).

[44] P.-L. Dallaire-Demers and N. Killoran, Quantum generative adversarial networks, Phys. Rev. A **98**, 012324 (2018).

[45] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, Phys. Rev. A **101**, 032308 (2020).

[46] J. Zeng, Y. Wu, J.-G. Liu, L. Wang, and J. Hu, Learning and inference on generative adversarial quantum circuits, Phys. Rev. A **99**, 052306 (2019).

[47] E. Farhi and H. Neven, Classification with quantum neural networks on near term processors, arXiv:1802.06002 [quant-ph].

[48] M. Schuld, M. Fingerhuth, and F. Petruccione, Implementing a distance-based classifier with a quantum interference circuit, Europhys. Lett. **119**, 60002 (2017).

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

[50] M. Schuld and N. Killoran, Quantum Machine Learning in Feature Hilbert Spaces, Phys. Rev. Lett. **122**, 040504 (2019).

[51] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature (London) **567**, 209 (2019).

[52] D. Zhu, N. M. Linke, M. Benedetti, K. A. Landsman, N. H. Nguyen, C. H. Alderete, A. Perdomo-Ortiz, N. Korda, A. Garfoot, C. Brecque, L. Egan, O. Perdomo, and C. Monroe, Training of quantum circuits on a hybrid quantum computer, Sci. Adv. **5**, eaaw9918 (2019).

[53] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nat. Phys. **15**, 1273 (2019).

[54] K. H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, and M. Kim, Quantum generalisation of feedforward neural networks, npj Quantum Inf. **3**, 36 (2017).

[55] E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A. G. Green, and S. Severini, Hierarchical quantum classifiers, npj Quantum Inf. **4**, 65 (2018).

[56] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, Implementable quantum classifier for nonlinear data, arXiv:1809.06056 [quant-ph].

[57] A. Uvarov, A. Kardashin, and J. Biamonte, Machine learning phase transitions with a quantum processor, Phys. Rev. A **102**, 012415 (2020).

[58] X. Gao, Z.-Y. Zhang, and L.-M. Duan, A quantum machine learning algorithm based on generative models, Sci. Adv. **4**, eaat9004 (2018).

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

[60] A. Perdomo-Ortiz, M. Benedetti, J. Realpe-Gómez, and R. Biswas, Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers, Quantum Sci. Technol. **3**, 030502 (2018).

[61] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, Quantum Boltzmann Machine, Phys. Rev. X **8**, 021050 (2018).

[62] L. Hu, S.-H. Wu, W. Cai, Y. Ma, X. Mu, Y. Xu, H. Wang, Y. Song, D.-L. Deng, C.-L. Zou *et al.*, Quantum generative adversarial learning in a superconducting quantum circuit, Sci. Adv. **5**, eaav2761 (2019).

[63] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. **103**, 150502 (2009).

[64] I. Cong and L. Duan, Quantum discriminant analysis for dimensionality reduction and classification, New J. Phys. **18**, 073011 (2016).

[65] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification, Phys. Rev. Lett. **113**, 130503 (2014).

[66] Z. Li, X. Liu, N. Xu, and J. Du, Experimental Realization of a Quantum Support Vector Machine, Phys. Rev. Lett. **114**, 140504 (2015).

[67] N. Liu and P. Wittek, Vulnerability of quantum classification to adversarial perturbations, Phys. Rev. A **101**, 062331 (2020).

[68] P. A. M. Casares and M. A. Martin-Delgado, A quantum active learning algorithm for sampling against adversarial attacks, New J. Phys. **22**, 073026 (2020).

[69] N. Wiebe and R. S. S. Kumar, Hardening quantum machine learning against adversaries, New J. Phys. **20**, 123019 (2018).

[70] N. Papernot, P. McDaniel, and I. Goodfellow, Transferability in machine learning: from phenomena to black-box attacks using adversarial samples, arXiv:1605.07277 [cs.CR].

[71] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, Adversarial attacks on medical machine learning, Science **363**, 1287 (2019).

[72] S. G. Finlayson, H. W. Chung, I. S. Kohane, and A. L. Beam, Adversarial attacks against medical deep learning systems, arXiv:1804.05296 [cs.CR].

[73] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).

[74] G. Li, P. Zhu, J. Li, Z. Yang, N. Cao, and Z. Chen, Security matters: A survey on adversarial machine learning, arXiv:1810.07339 [cs.LG].

[75] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, Adversarial attacks and defences: A survey, arXiv:1810.00069 [cs.LG].

[76] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, The limitations of deep learning in adversarial settings, in *2016 IEEE European Symposium on Security and Privacy* (IEEE, Saarbrucken, Germany, 2016), pp. 372–387.

[77] A. Kurakin, I. J. Goodfellow, and S. Bengio, Adversarial machine learning at scale, in *Fifth International Conference on Learning Representations* (ICLR, Toulon, France, 2017).

[78] P. Samangouei, M. Kabkab, and R. Chellappa, Defense-GAN: Protecting classifiers against adversarial attacks using generative models, in *Sixth International Conference on Learning Representations* (ICLR, Vancouver, BC, Canada, 2018).

[79] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial nets, in *Advances in Neural Information Processing Systems* (NeurIPS, Montreal, Canada, 2014), pp. 2672–2680.

[80] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, in *2016 IEEE Symposium on Security and Privacy* (IEEE, San Jose, CA, 2016), pp. 582–597.

[81] G. Hinton, O. Vinyals, and J. Dean, Distilling the knowledge in a neural network, arXiv:1503.02531 [stat.ML].

[82] C. Blank, D. K. Park, J.-K. K. Rhee, and F. Petruccione, Quantum classifier with tailored quantum kernel, npj Quantum Inf. **6**, 41 (2020).

[83] F. Tacchino, C. Macchiavello, D. Gerace, and D. Bajoni, An artificial neuron implemented on an actual quantum processor, npj Quantum Inf. **5**, 26 (2019).

[84] IBM quantum experience, http://www.research.ibm.com/ibm-q/.
[85] J. Preskill, Quantum Computing in the NISQ era and beyond, Quantum **2**, 79 (2018).
[86] I. Kerenidis and A. Prakash, Quantum recommendation systems, in *Eighth Innovations in Theoretical Computer Science Conference (ITCS 2017)*, Leibniz International Proceedings in Informatics (LIPIcs) Vol. 67, edited by C. H. Papadimitriou (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017), pp. 49:1–49:21.
[87] V. Giovannetti, S. Lloyd, and L. Maccone, Architectures for a quantum random access memory, Phys. Rev. A **78**, 052310 (2008).
[88] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, arXiv:1307.0411 [quant-ph].
[89] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum deep learning, arXiv:1412.3489 [quant-ph].
[90] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum Random Access Memory, Phys. Rev. Lett. **100**, 160501 (2008).
[91] S. Aaronson, Read the fine print, Nat. Phys. **11**, 291 (2015).
[92] M. Möttönen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, Quantum Circuits for General Multiqubit Gates, Phys. Rev. Lett. **93**, 130502 (2004).
[93] E. Knill, Approximation by quantum circuits, arXiv:quant-ph/9508006 [quant-ph].
[94] M. Plesch and I. C. V. Brukner, Quantum-state preparation with universal gate decompositions, Phys. Rev. A **83**, 032302 (2011).
[95] L. Grover and T. Rudolph, Creating superpositions that correspond to efficiently integrable probability distributions, arXiv:quant-ph/0208112 [quant-ph].
[96] A. N. Soklakov and R. Schack, Efficient state preparation for a register of quantum bits, Phys. Rev. A **73**, 012307 (2006).
[97] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature (London) **549**, 242 (2017).
[98] R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Fährmann, B. Meynard-Piganeau, and J. Eisert, Stochastic gradient descent for hybrid quantum-classical optimization, arXiv:1910.01155 [quant-ph].
[99] N. Yamamoto, On the natural gradient for variational quantum eigensolver, arXiv:1909.05074 [quant-ph].
[100] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, Quantum natural gradient, Quantum **4**, 269 (2020).
[101] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980 [cs.LG].
[102] S. J. Reddi, S. Kale, and S. Kumar, On the convergence of Adam and beyond, in *Sixth International Conference on Learning Representations* (ICLR, Vancouver, BC, Canada, 2018).
[103] A. Harrow and J. Napp, Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms, arXiv:1901.05374 [quant-ph].
[104] L. B. Rall and G. F. Corliss, An introduction to automatic differentiation, in *Computational Differentiation: Techniques, Applications, and Tools* (SIAM, Philadelphia, PA, 1996), pp. 1–17.
[105] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong, Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states, Phys. Rev. A **95**, 042308 (2017).
[106] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, Julia: A fresh approach to numerical computing, SIAM Rev. **59**, 65 (2017).
[107] X.-Z. Luo, J.-G. Liu, P. Zhang, and L. Wang, Yao.jl: Extensible, efficient framework for quantum algorithm design, arXiv:1912.10877 [quant-ph].
[108] M. Innes, Flux: Elegant machine learning with Julia, J. Open Source Software **3**, 602 (2018).
[109] https://github.com/FluxML/Zygote.jl.
[110] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Machine Learn. Res. **15**, 1929 (2014).
[111] S. E. Venegas-Andraca and S. Bose, Storing, processing, and retrieving an image using quantum mechanics, in *Quantum Information and Computation* (SPIE, Orlando, Florida, 2003), pp. 137–147.
[112] X.-W. Yao, H. Wang, Z. Liao, M.-C. Chen, J. Pan, J. Li, K. Zhang, X. Lin, Z. Wang, Z. Luo, W. Zheng, J. Li, M. Zhao, X. Peng, and D. Suter, Quantum Image Processing and Its Application to Edge Detection: Theory and Experiment, Phys. Rev. X **7**, 031041 (2017).
[113] Y. LeCun, C. Cortes, and C. J. Burges, The MNIST database of handwritten digits, http://yann.lecun.com/exdb/mnist/.
[114] R. Benenson, Who is the best in MNIST?, http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html.
[115] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition, in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16 (ACM, New York, 2016), pp. 1528–1540.
[116] M. Saffman, Quantum computing with atomic qubits and Rydberg interactions: Progress and challenges, J. Phys. B: At., Mol. Opt. Phys. **49**, 202001 (2016).
[117] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, A quantum engineer's guide to superconducting qubits, Appl. Phys. Rev. **6**, 021318 (2019).
[118] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, Trapped-ion quantum computing: Progress and challenges, Appl. Phys. Rev. **6**, 021314 (2019).
[119] Y. Wu, S.-T. Wang, and L.-M. Duan, Noise analysis for high-fidelity quantum entangling gates in an anharmonic linear Paul trap, Phys. Rev. A **97**, 062325 (2018).
[120] Y. Zhang and E.-A. Kim, Quantum Loop Topography for Machine Learning, Phys. Rev. Lett. **118**, 216401 (2017).
[121] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nat. Phys. **13**, 431 (2017).
[122] E. P. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, Nat. Phys. **13**, 435 (2017).
[123] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Machine learning quantum phases of matter beyond the fermion sign problem, Sci. Rep. **7**, 8823 (2017).
[124] S. J. Wetzel, Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders, Phys. Rev. E **96**, 022140 (2017).

[125] W. Hu, R. R. P. Singh, and R. T. Scalettar, Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination, Phys. Rev. E **95**, 062122 (2017).

[126] Y.-T. Hsu, X. Li, D.-L. Deng, and S. Das Sarma, Machine Learning Many-Body Localization: Search for the Elusive Nonergodic Metal, Phys. Rev. Lett. **121**, 245701 (2018).

[127] J. F. Rodriguez-Nieva and M. S. Scheurer, Identifying topological order through unsupervised machine learning, Nat. Phys. **15**, 790 (2019).

[128] P. Zhang, H. Shen, and H. Zhai, Machine Learning Topological Invariants with Neural Networks, Phys. Rev. Lett. **120**, 066401 (2018).

[129] N. Sun, J. Yi, P. Zhang, H. Shen, and H. Zhai, Deep learning topological invariants of band insulators, Phys. Rev. B **98**, 085402 (2018).

[130] P. Huembeli, A. Dauphin, and P. Wittek, Identifying quantum phase transitions with adversarial neural networks, Phys. Rev. B **97**, 134109 (2018).

[131] W. Lian, S.-T. Wang, S. Lu, Y. Huang, F. Wang, X. Yuan, W. Zhang, X. Ouyang, X. Wang, X. Huang, L. He, X. Chang, D.-L. Deng, and L. Duan, Machine Learning Topological Phases with a Solid-State Quantum Simulator, Phys. Rev. Lett. **122**, 210503 (2019).

[132] B. S. Rem, N. Käming, M. Tarnowski, L. Asteria, N. Fläschner, C. Becker, K. Sengstock, and C. Weitenberg, Identifying quantum phase transitions using artificial neural networks on experimental data, Nat. Phys. **15**, 917 (2019).

[133] A. Bohrdt, C. S. Chiu, G. Ji, M. Xu, D. Greif, M. Greiner, E. Demler, F. Grusdt, and M. Knap, Classifying snapshots of the doped Hubbard model with machine learning, Nat. Phys. **15**, 921 (2019).

[134] Y. Zhang, A. Mesaros, K. Fujita, S. Edkins, M. Hamidian, K. Ch'ng, H. Eisaki, S. Uchida, J. S. Davis, E. Khatami *et al.*, Machine learning in electronic-quantum-matter imaging experiments, Nature (London) **570**, 484 (2019).

[135] S. Jiang, S. Lu, and D.-L. Deng, Vulnerability of machine learning phases of matter, arXiv:1910.13453.

[136] Y. Zhang, R. G. Melko, and E.-A. Kim, Machine learning $\mathbb{Z}_2$ quantum spin liquids with quasiparticle statistics, Phys. Rev. B **96**, 245119 (2017).

[137] X.-J. Liu, K. T. Law, and T. K. Ng, Realization of 2d Spin-Orbit Interaction and Exotic Topological Orders in Cold Atoms, Phys. Rev. Lett. **112**, 086401 (2014).

[138] E. Alba, X. Fernandez-Gonzalvo, J. Mur-Petit, J. K. Pachos, and J. J. Garcia-Ripoll, Seeing Topological Order in Time-of-Flight Measurements, Phys. Rev. Lett. **107**, 235301 (2011).

[139] D.-L. Deng, S.-T. Wang, and L.-M. Duan, Direct probe of topological order for cold atoms, Phys. Rev. A **90**, 041601(R) (2014).

[140] G. Jotzu, M. Messer, R. Desbuquois, M. Lebrat, T. Uehlinger, D. Greif, and T. Esslinger, Experimental realization of the topological haldane model with ultracold fermions, Nature (London) **515**, 237 (2014).

[141] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nat. Phys. **10**, 631 (2014).

[142] Y. Du, M.-H. Hsieh, T. Liu, D. Tao, and N. Liu, Quantum noise protects quantum classifiers against adversaries, arXiv:2003.09416 [quant-ph].

[143] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, The space of transferable adversarial examples, arXiv:1704.03453 [stat.ML].

[144] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*, Princeton Series in Applied Mathematics (Princeton University Press, Princeton, NJ, 2009), Vol. 28.

[145] S. Chakrabarti, H. Yiming, T. Li, S. Feizi, and X. Wu, Quantum Wasserstein generative adversarial networks, in *Advances in Neural Information Processing Systems* (NeurIPS, Montreal, Canada, 2019), pp. 6778–6789.

[146] M. Ledoux, *The Concentration of Measure Phenomenon*, Mathematical Surveys and Monographs (American Mathematical Society, Providence, Rhode Island, 2001), Vol. 89.

[147] S. Lu, S. Huang, K. Li, J. Li, J. Chen, D. Lu, Z. Ji, Y. Shen, D. Zhou, and B. Zeng, Separability-entanglement classifier via machine learning, Phys. Rev. A **98**, 012315 (2018).

[148] Y.-C. Ma and M.-H. Yung, Transforming Bell's inequalities into state classifiers with machine learning, npj Quantum Inf. **4**, 34 (2018).

[149] A. Chefles, Quantum state discrimination, Contemp. Phys. **41**, 401 (2000).

[150] J. Wang, S. Paesani, R. Santagati, S. Knauer, A. A. Gentile, N. Wiebe, M. Petruzzella, J. L. O'Brien, J. G. Rarity, A. Laing *et al.*, Experimental quantum Hamiltonian learning, Nat. Phys. **13**, 551 (2017).

[151] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita, Reconstructing quantum states with generative models, Nat. Mach. Intell. **1**, 155 (2019).

[152] P. W. Anderson, Infrared catastrophe in Fermi Gases with Local Scattering Potentials, Phys. Rev. Lett. **18**, 1049 (1967).

[153] D.-L. Deng, J. H. Pixley, X. Li, and S. Das Sarma, Exponential orthogonality catastrophe in single-particle and many-body localized systems, Phys. Rev. B **92**, 220201(R) (2015).

[154] F. Chollet *et al.*, Keras, https://keras.io.

[155] M. Abadi *et al.*, Tensorflow: A system for large-scale machine learning, in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16 (USENIX Association, Berkeley, CA, 2016), pp. 265–283.

[156] V. Nair and G. E. Hinton, Rectified linear units improve restricted Boltzmann machines, in *Proceedings of the 27th International Conference on International Conference on Machine Learning* (Omnipress, Madison, WI, 2010), pp. 807–814.

[157] N. Papernot *et al.*, Technical report on the Cleverhans v2.1.0 adversarial examples library, arXiv:1610.00768 [cs.LG].