Robustness Verification of Quantum Machine Learning

Ji Guan[1], Wang Fang[1], and Mingsheng Ying[2,1,3]
[1] *State Key Laboratory of Computer Science, Institute of Software,*
Chinese Academy of Sciences, Beijing 100190, China
[2] *Center for Quantum Software and Information,*
University of Technology Sydney, NSW 2007, Australia and
[3] *Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*

(Dated: August 18, 2020)

Several important models of machine learning algorithms have been successfully generalized to the quantum world, with potential applications to data analytics in quantum physics that can be implemented on the near future quantum computers. However, noise and decoherence are two major obstacles to the practical implementation of quantum machine learning. In this work, we introduce a general framework for the robustness analysis of quantum machine learning algorithms against noise and decoherence. We argue that fidelity is the only pick of measuring the robustness. A robust bound is derived and an algorithm is developed to check whether or not a quantum machine learning algorithm is robust with respect to the training data. In particular, this algorithm can help to defense attacks and improve the accuracy as it can identify useful new training data during checking. The effectiveness of our robust bound and algorithm is confirmed by the case study of quantum phase recognition. Furthermore, this experiment demonstrates a trade-off between the accuracy of quantum machine learning algorithms and their robustness.

Introduction—In the last few years, the successful interplay between machine learning and quantum physics shed new light on both fields. On the one hand, machine learning has been dramatically developed to satisfy the need of the industry over the past two decades. At the same time, many challenging and even impossible quantum physical problems have been solved by automated learning. Notably, inaccessible quantum many-body problems have been solved by neural networks, one instance of machine learning [8]. On the other hand, as the new model of computation under quantum mechanics, quantum computing has been proved that it can speed up classical algorithms [21]. This motivates the development of quantum machine learning and provides the possibility of improving the existing computational power of machine learning to a new level. See the review papers [4, 12] for the details. After that, quantum machine learning was integrated into solving real-world problems in quantum physics. One essential example is that quantum convolutional neural networks inspired by machine learning were proposed to implement quantum phase recognition [9]. Quantum phase recognition asks whether a given input quantum state belongs to a particular quantum phase of matter. Stepping into industries, Google recently built up a framework for the design and training of quantum machine learning within its famous classical machine learning platform TensorFlow [6].

Even though quantum machine leaning outperforms classical counterpart in some way, the difficulties in the classical world are expected to be encountered in the quantum case. Classical machine learning has been found to be vulnerable to intentionally-crafted adversarial examples (e.g. [13, 14]). Adversarial examples are inputs to a machine learning algorithm that an attacker has crafted to cause the algorithm to make a mistake. The design of proper defense strategies has been actively investigated, giving rise to an emergent field of adversarial machine learning. This phenomenon is more common in the quantum world since quantum noise and decoherence are inevitable in quantum computation, at least in the current NISQ (Noisy Intermediate-Scale Quantum) era. Differently, the quantum attacker is usually the surroundings instead of humans in classical machine learning, and the information of the environment is unknown. Up to our best knowledge, the studies of quantum machine learning robustness only consider the situation of a *known* noise source. For example, Lu et al. [18] studied the robustness to various classical adversarial attacks; Du et al. [11] proved that appending depolarization noise in quantum circuits for classifications, a robust bound against adversaries can be derived; Liu and Wittek [17] gave a robust bound for the quantum noise coming from a special unitary group. However, to protect against an *unknown* adversary, we need to derive a robust guarantee against a worst-case scenario, from which the commonly-assume known noise sources (e.g. depolarization noise [11]) are usually far. Yet in the case of unknown noise, several basic issues are still unsolved: In theory, it is unclear how to compute a bound of the robustness for any given quantum machine learning algorithm. In practice, an effective way to find an adversarial example, and the corresponding defense strategy, is lacking. Indeed, we do not even know what is the best metric measuring the robustness, the same as the classical case [22].

In this letter, we resolve all of the above issues. In particular, we claim that fidelity is the only pick of measuring the robustness of quantum machine learning. Based on this, a robust bound for any quantum machine learning classification algorithm is obtained. Then we develop an algorithm by Semi-definite Programming to check whether or not a quantum machine learning algorithm is robust with respect to the training data. A special strength of this algorithm is that it can help to defense attacks and even improve the accuracy by identifying useful new training data during checking. The effectiveness of our robust bound and algorithm is confirmed by the case study of quantum phase recognition [9]. The result of this experiment reveals that a trade-off between the validation and robust accuracy.

Quantum Machine Learning–We study the vulnerability of quantum machine learning with a focus on a specific learning model called quantum (supervised) classification. Given a Hilbert space $\mathcal{H}$, we write $\mathcal{D}(\mathcal{H})$ for the set of all (mixed) quantum states in $\mathcal{H}$.

Definition 1 *A quantum (machine learning) classification algorithm $\mathcal{A}$ is a mapping $\mathcal{D}(\mathcal{H}) \to \mathcal{C}$, where $\mathcal{C}$ is the set of classes we are interested in.*

Following the training strategy of classical machine learning, the classification $\mathcal{A}$ is learned through a dataset T instead of pre-defined. This training dataset $T = \{(\rho_i, c_i)\}_{i=1}^N$ consists of $N < \infty$ pairs (ρ_i, c_i), meaning that state ρ_i belongs to class c_i. To learn $\mathcal{A}$, we initialize a parameterized quantum circuit (including measurement control) $\mathcal{E}_\theta$ and fix a POVM $\{\Pi_k\}_{k\in\mathcal{C}}$, where $\mathcal{E}_\theta$ is a quantum super-operator and θ is a set of free parameters that can be tuned. Then we can compute the probability of the outcome of the measurement being k:

$$f_k(\theta, \rho) = \mathrm{tr}(\Pi_k \mathcal{E}_\theta(\rho)). \tag{1}$$

The quantum classification algorithm $\mathcal{A}$ outputs the class label c for ρ using the following condition:

$$\mathcal{A}(\theta, \rho) = \arg\max_k \mathrm{tr}(\Pi_k \mathcal{E}_\theta(\rho)). \tag{2}$$

The learning is carried out as θ is optimized to minimize the empirical risk

$$\min_\theta \frac{1}{N} \sum_{i=1}^N l(f(\theta, \rho_i), c_i), \tag{3}$$

where l refers to a predefined loss function, $f(\theta, \rho)$ is a probability vector with each $f_k(\theta, \rho), k \in \mathcal{C}$ as its element, and c_i is also seen as a probability vector with the entry corresponding to c_i being 1 and others being 0. The goal is to find the optimized parameters θ^* minimizing the risk in Eq.(3) for the given dataset T. Mean-squared error (MSE) is the most popular instance of the empirical risk, i.e., the loss function l is squared error:

$$l(f(\theta, \rho_i), c_i) = \frac{1}{C} \|f(\theta, \rho_i) - c_i\|_2^2,$$

where C is the number of classes in $\mathcal{C}$, $\|\cdot\|_2$ is the l_2-norm.

In this letter, we focus on the well-trained quantum classification algorithm $\mathcal{A}$, usually called a quantum classifier. Here, $\mathcal{A}$ is said to be well-trained if training and validation accuracy are both high ($\geq 95\%$). The training (validation) accuracy is the proportion of the number of $\mathcal{A}$ successfully classifying data in a training (validation) dataset. A validation dataset is mathematically equivalent to a training dataset but only for testing $\mathcal{A}$ rather than learning $\mathcal{A}$. In this context, θ^* is naturally omitted, i.e., $\mathcal{A}(\rho) = \mathcal{A}(\theta^*, \rho)$ and $\mathcal{E}(\rho) = \mathcal{E}_{\theta^*}(\rho)$. Briefly, $\mathcal{A}$ only consists of a super-operator $\mathcal{E}$ and a POVM $\{\Pi_k\}_k$, denoted by $\mathcal{A} = (\mathcal{E}, \{\Pi_k\}_k)$.

An important question left unexplored is: how robust are well-trained quantum machine learning classification algorithms to adversarial perturbations by quantum noise and decoherence?

Robustness—Intuitively, the robustness of quantum classifier $\mathcal{A}$ is the ability to make correct classification with a small perturbation to the input states. In other words, there are no adversarial examples. A quantum state σ is considered as an adversarial example if it is similar to a benign state ρ, such that ρ is correctly classified and σ is classified differently than ρ. Formally,

Definition 2 (Adversarial Example) *Suppose we are given a quantum classifier $\mathcal{A}(\cdot)$, an input example (ρ, c), a distance metric $D(\cdot,\cdot)$ and a small enough threshold value $\varepsilon > 0$. Then σ is said to be an adversarial example of ρ if the following is true*

$$(\mathcal{A}(\rho) = c) \wedge (\mathcal{A}(\sigma) \neq c) \wedge (D(\rho, \sigma) \leq \varepsilon).$$

The leftmost condition $(\mathcal{A}(\rho) = c)$ checks that ρ is correctly classified, the middle condition $(\mathcal{A}(\sigma) \neq c)$ means that σ is incorrectly classified, and the rightmost condition $(D(\rho, \sigma) \leq \varepsilon)$ indicates that ρ and σ are similar (i.e., their distance is small).

Notably, by the above definition, if $\mathcal{A}$ incorrectly classifies ρ, then we do not need to consider the corresponding adversarial examples. This is the correctness issue of quantum classifier $\mathcal{A}$ rather than the robustness issue. Hence, in the following discussions, we only consider the set of all correctly recognized states.

It is intuitive that the absence of adversarial examples leads to the robustness.

Definition 3 (Adversarial robustness) *Let $\mathcal{A}$ be a quantum classifier. Then ρ is ε-robust for $\mathcal{A}$ if there is no any adversarial example of ρ. Furthermore, the ε-robust accuracy of $\mathcal{A}$ is the proportion of the number of ε-robust states in the training dataset.*

We saw in Definition 2 that the robustness depends on the distance $D(\cdot,\cdot)$. So, it is essential to properly choose a metric D meaningful in quantum physics. In the classical case, such a metric should promise that a small perturbation is imperceptible to humans, and vice vers. Otherwise, we cannot take the advantage of machine learning than human's distinguishability. For instance, in image recognition, it should satisfy perceptual similarity in the sense that humans would consider adversarial examples generated by it perceptually similar to benign image [22]. Finding such a metric is still an open problem [22]. In quantum information science, trace distance and fidelity are the best-known metrics measuring the distinguishability between two states by humans [21]. Both of them have been employed in the previous studies of the robustness of quantum machine learning (e.g. trace distance in [11] and fidelity in [18]). But so far, to the best of our knowledge, there is no any discussion about which one is better. Here, we claim that fidelity is the only pick in the context of quantum machine learning. The main difference between trace norm and fidelity is the number of state copies as the resource. Trace distance quantifies the maximum probability of correctly guessing through a measurement whether ρ or σ was prepared, while fidelity asserts the same quantity when infinite samples of ρ and σ are supplied (See Supplemental Material [1] for more details). In quantum machine learning, enough copies of states are the precondition of statistics in Eq.(1) for learning and classifying. Thus, we choose to use the metric:

$$D(\rho, \sigma) = 1 - F(\rho, \sigma)$$

defined by fidelity $F(\rho, \sigma) = [\mathrm{tr}(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}})]^2$.

In practical applications, the value of ε in Definition 3 represents the ability of state preparation by quantum controls. The state-of-the-art is that single qubit can be prepared with fidelity 99.99% (e.g. [7, 20]). We choose it as the value of ε in the latter experiments. In other words, we always consider 10^{-4}-robustness in case studies.

After setting the metric D, a robust bound can be derived.

Lemma 1 (Robust Bound) *Let $\mathcal{A} = (\mathcal{E}, \{\Pi_k\}_k)$ be a quantum classifier and ρ is a given state. If $p_1 - p_2 > 2\sqrt{\varepsilon}$, then ρ is ε-robust, where p_1 and p_2 are the first and second largest elements of $\{\mathrm{tr}(\Pi_k \mathcal{E}(\rho))\}_k$, respectively.*

Proof. See Supplemental Material [1] for the proof. □

The above robust bound gives us a quick robustness verification by the measurement outcomes of ρ without searching any possible adversarial examples. However, it is not necessary condition of ε-robustness. Fortunately, when $p_1 - p_2 \leq 2\sqrt{\varepsilon}$, we can still check the ε-robustness by Semi-definite Programming (SDP).

Theorem 1 (Robustness Verification) *Let $\mathcal{A}$ be as in Lemma 1 and ρ is a state with $\mathcal{A}(\rho) = l$. Then ρ is ε-robust if and only if $\delta > \varepsilon$, where $\delta = \min_{k \neq l} \delta_k$ and δ_k is the solution of the following SDP:*

$$\delta_k = \min_{\sigma \in \mathcal{D}(\mathcal{H})} 1 - F(\rho, \sigma)$$

subject to the following three constrains:

$$(\sigma \geq 0) \wedge (\mathrm{tr}(\sigma) = 1) \wedge (\mathrm{tr}((\Pi_k - \Pi_l)\mathcal{E}(\sigma)) > 0),$$

where if the SDP is unsolved, then $\delta = +\infty$.

Proof. The claim directly follows from the definition of adversarial robustness in Definition 3. □

Algorithm 1 RobustnessVerifier($\mathcal{A}, \varepsilon, \rho, l$)

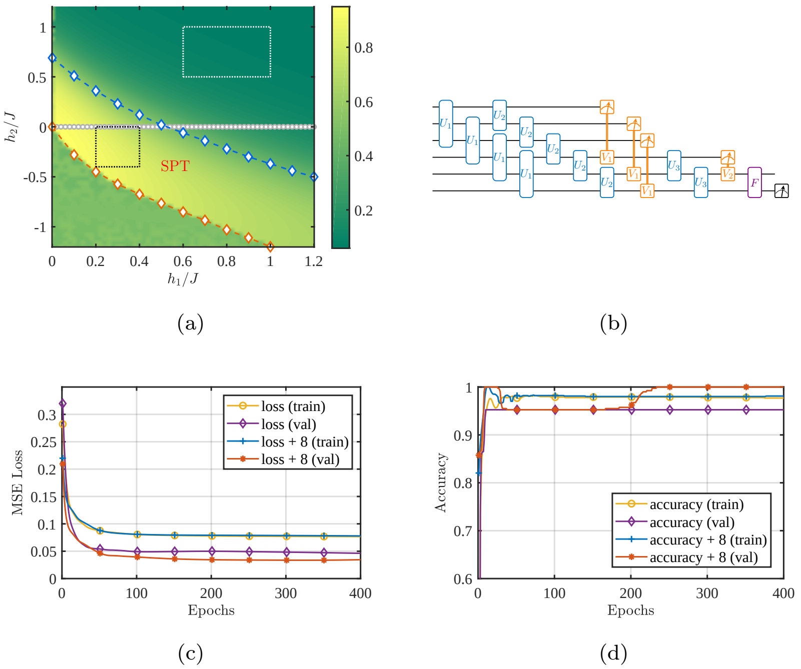
```
Require: $\mathcal{A} = (\mathcal{E}, \{\Pi_k\}_{k \in \mathcal{C}})$ is a well-trained quantum classifier, $\varepsilon < 1$ is a real number, $(\rho, l)$ is an element of the training dataset of $\mathcal{A}$
Ensure: true indicates $\rho$ is $\varepsilon$-robust or false with an adversarial example $\sigma$ indicates $\rho$ is not $\varepsilon$-robust
for each $k \in \mathcal{C}$ and $k \neq l$ do
    By a SDP solver, compute $\delta_k$ with an optimal state $\sigma_k$ in the SDP of Theorem 1
end for
Let $\delta = \min_k \delta_k$ and $k^* = \arg\min_k \delta_k$
if $\delta > \varepsilon$ then
    return true
else
    return false and $\sigma_{k*}$
end if
```

An SDP verification algorithm (Algorithm 1) can be directly derived from the above theorem to find the minimum adversarial perturbation δ caused by quantum noise and decoherence and check the ε-robustness of ρ. In particular, taking the advantage of Algorithm 1, we are able to extend the technique of adversarial training in classical machine learning [19] into the quantum world; that is, an adversarial example σ is automatically generated once ε-robustness of ρ fails, and then by appending (σ, l) into the training dataset, retraining $\mathcal{A}$ can improve the robustness of some non-robust states and the training and validation accuracy of classifier $\mathcal{A}$.

We remark that by Theorem 1, for checking the robustness of $\mathcal{A}$, one must implement Algorithm 1 one-by-one for each state in the training dataset, and it costs much time as current SDP solvers used in Algorithm 1 utilize interior-point methods and scale as $O(n^{4.5})$ or worse, where n is the number of rows of the semi-definite matrix in SDP, i.e., the dimension of Hilbert space. However, the robust bound given in Lemma 1 can help to speed up the process by quickly finding all potential non-robust states as the complexity of finding the bound is $O(n^2)$. In practice, this bound scales well, as confirmed by the following experiment.

To demonstrate our method of robustness verification, we check 10^{-4}-robustness of a machine learning algorithm for quantum phase recognition—one of the core problems in quantum physics.

Quantum Phase Recognition—Quantum phase recognition (QPR) of one dimensional many-body systems has been attacked by quantum convolutional neural networks (QCNNs) proposed by Cong et al. [9]. Consider a $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry-protected topological (SPT) phase $\mathcal{P}$ and the ground states of a family of Hamiltonians on spin-1/2



(a)

(b)

(c)

(d)

FIG. 1: (a) The phase diagram obtained by our trained QCNN model for input size $N = 6$ spins. (b) Our QCNN circuit model. (c) The MSE loss for our training and retraining (+8) processes. (d) Accuracy for our training and retraining (+8) processes.

chain with open boundary conditions:

$$H = -J\sum_{i=1}^{N-2} Z_i X_{i+1} Z_{i+2} - h_1 \sum_{i=1}^{N} X_i - h_2 \sum_{i=1}^{N-1} X_i X_{i+1}$$

where X_i, Z_i are Pauli operators for the spin at site i, and the $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry is generated by $X_{\text{even(odd)}} = \prod_{i\in\text{even(odd)}} X_i$. The goal is to identify whether the ground state $|\psi\rangle$ of H belongs to phase $\mathcal{P}$ when H is regarded as a function of $(h_1/J, h_2/J)$. For small N, a numerical simulation can be used to exactly solve this problem [9]; Fig. 1a shows the exact phase boundary points (blue and red diamonds) between SPT phase and non-SPT (paramagnetic or antiferromagnetic) phase for $N = 6$. Thus the 6-qubit instance is an excellent testbed for different new methods and techniques of QPR. Here, we train a QCNN model to implement 6-qubit QPR in this setting.

To generate the dataset for training, we sample a serials of Hamiltonian H with $h_2/J = 0$, uniformly varying h_1/J from 0 to 1.2 and compute their corresponding ground states; see the gray line in Fig. 1a. For the testing, we uniformly sample a set of validation data from two random regions of the 2-dimensional space $(h_1/J, h_2/J)$; see the two dashed rectangles in Fig. 1a. Finally, we obtain 1000 training data and 400 validation data. Our parameterized QCNN circuit is shown in Fig. 1b, and the unitaries U_i, V_j, F are parameterized with generalized Gell-Mann matrix basis [3]: $U = \exp(-i\sum_j \theta_j \Lambda_j)$, where Λ_j is a matrix and θ_j is a real number; the total number of parameters θ_j, Λ_j is 114. For the outcome measurement of one qubit, we use POVM $\Pi = \{\Pi_0 = |+\rangle\langle+|, \Pi_1 = |-\rangle\langle-|\}$ to predict that input states belongs to $\mathcal{P}$ with output 0, where $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$. Targeting to minimizing the MSE form of Eq. (3), we use Adam optimizer [15] to update the 114 parameters. After training, 97.7% training accuracy and 95.25% validation accuracy are obtained; see Figs. 1c and 1d for the training process in each epoch. At the same time, our classifier conducts a phase diagram (the colorful figure in Fig. 1a), where the learned phase boundary almost perfectly matches the exact one gotten by the numerical simulation. All these results indicate that our classifier is well-trained.

Next, we initialize the robustness checking for the classifier. First, we apply our robust bound (here the value is 0.02) in Lemma 1 to pick up all potential non-robust states from the 1000 points in the training dataset. Only 8 points are left and the bound scales well. Finally, we check 10^{-4}-robustness of the 8 candidates by Algorithm 1. Indeed, all 8 of the points are non-robust and the robust accuracy of the QCNN is 99.2%. Furthermore, we see that all of them are close to the exact boundary (the blue diamonds line in Fig. 1a).

Now we start the adversarial training. Appending all the adversarial examples found by Algorithm 1 to the training dataset forms as a new dataset of 1008 points, we retrain our QCNN model; see Figs. 1c and 1d for the retraining process in each epoch. After training, we achieve 98.12% training accuracy and 100% validation accuracy. Then we check the same 10^{-4}-robustness of the new classifier, and obtain 17 non-robust states, leading to 98.31% robust accuracy. All of the 17 states are close to the exact boundary (the blue diamonds line in Fig. 1a).

Finally, we analyze the implications of the adversarial training. The improvement of the training accuracy indicates that the retrained classifier is immune to some previous adversarial examples as the new training dataset contains the 8 adversarial examples. Actually, we find that 3 states among the previous 8 non-robust states are fixed to be robust now. Furthermore, the validation accuracy is significantly improved, while the robust accuracy decreases. This trade-off between the validation and robust accuracy confirms the possibility of existing a quantum version of No Free Lunch theorem [10, 23]. As we see that all non-robust states are always the neighbors of the exact boundary, the learned boundary by the QCNN is vulnerable to unknown quantum noise and decoherence, while the other points far from the boundary are robust and can be perfectly classified by the QCNN after retraining. Moreover, the validation accuracy of the retrained QCNN is higher than the training accuracy. This is a surprising phenomenon as the classifier learns from the training dataset and should fit better on it. This results of the new boundary learned by the retrained model far from the regions of our validation samples. It is still unclear how to defense all adversarial examples, how to learn a robust boundary for QPR, and how to balance the trade-off. Indeed, a similar trade-off problem in classical machine learning is also unsolved [23].

Conclusion—In this letter, we initiate the research of the robustness verification of quantum machine learning algorithms against unknown quantum noise and decoherence. Our robust bound scales well, and the SDP verification algorithm can verify the ε-robustness of quantum machine learning algorithms and provides useful counterexamples for the adversarial training. Furthermore, the adversarial training can improve the training and validation accuracy but lower the robust accuracy, as confirmed by the case study of QPR.

Tensor networks are the best-known data structure for implementing large-scale quantum classifiers (e.g. QCNNs with 45 qubits in [9]). For practical applications, we are going to incorporate tensor networks into our robustness verification algorithm so that it can scale up to achieve the demand of NISQ devices (of ≥ 50 qubits).

More generally, further investigations are required to better understand the role of the robustness in quantum machine learning, especially in learning phase of quantum many-body systems.

[1] *See Supplemental Material at [URL will be inserted by publisher] for details.*

[2] K. M. Audenaert, J. Calsamiglia, R. Munoz-Tapia, E. Bagan, L. Masanes, A. Acin, and F. Verstraete. Discriminating states: The quantum chernoff bound. *Physical review letters*, 98(16):160501, 2007.

[3] R. A. Bertlmann and P. Krammer. Bloch vectors for qudits. *Journal of Physics A: Mathematical and Theoretical*, 41(23):235303, May 2008.

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

[5] R. J. Blume-Kohout. How distinguishable are two quantum processes? or what is the error rate of a quantum gate? Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2017.

[6] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, M. Y. Niu, R. Halavati, E. Peters, et al. Tensorflow quantum: A software framework for quantum machine learning. *arXiv preprint arXiv:2003.02989*, 2020. See https://www.tensorflow.org/quantum for the platform.

[7] A. Burrell, D. Szwer, S. Webster, and D. Lucas. Scalable simultaneous multiqubit readout with 99. 99% single-shot fidelity. *Physical Review A*, 81(4):040302, 2010.

[8] G. Carleo and M. Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

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

[10] E. Dohmatob. Limitations of adversarial robustness: strong no free lunch theorem. *arXiv preprint arXiv:1810.04065*, 2018.

[11] Y. Du, M.-H. Hsieh, T. Liu, D. Tao, and N. Liu. Quantum noise protects quantum classifiers against adversaries. *arXiv preprint arXiv:2003.09416*, 2020.

[12] V. Dunjko and H. J. Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, 81(7):074001, 2018.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[14] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pages 43–58, 2011.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[16] Y. LeCun and C. Cortes. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010.

[17] N. Liu and P. Wittek. Vulnerability of quantum classification to adversarial perturbations. *Physical Review A*, 101(6):062331, 2020.

[18] S. Lu, L.-M. Duan, and D.-L. Deng. Quantum adversarial machine learning. *arXiv preprint arXiv:2001.00030*, 2019.

[19] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[20] A. Myerson, D. Szwer, S. Webster, D. Allcock, M. Curtis, G. Imreh, J. Sherman, D. Stacey, A. Steane, and D. Lucas. High-fidelity readout of trapped-ion qubits. *Physical Review Letters*, 100(20):200502, 2008.

[21] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[22] M. Sharif, L. Bauer, and M. K. Reiter. On the suitability of lp-norms for creating and preventing adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1605–1613, 2018.

[23] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

SUPPLEMENTAL MATERIAL

FIDELITY V.S. TRACE DISTANCE

In this section, we give more details of the reason that fidelity is the only pick as the metric in the study of the robustness of quantum machine learning.

First, as we known, trace distance quantifies the maximum probability $P_{correct}$ of correctly guessing whether ρ or σ was prepared after making a measurement [21]:

$$P_{correct} = \frac{1}{2} + \frac{1}{2}T(\rho, \sigma)$$

where

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

is the trace distance between ρ and σ, and $\|\rho - \sigma\|_{tr}$ is trace norm. Thus when N copies of ρ and σ are provided,

$$P_{correct} = \frac{1}{2} + \frac{1}{2}T(\rho^{\otimes N}, \sigma^{\otimes N}).$$

So how does $T(\rho^{\otimes N}, \sigma^{\otimes N})$ behave as N tends to infinite as the scenario of quantum machine learning? The quantum Chernoff bound explains this [2]:

$$T(\rho^{\otimes N}, \sigma^{\otimes N}) \approx 1 - \frac{1}{2}const \cdot e^{-C(\sigma,\rho)N}$$

where *const* is a non-zero constant and $C(\sigma, \rho)$ is the quantum Chernoff bound. And when the states are pretty close, its close to the infidelity (1-$F(\rho, \sigma)$) [5]:

$$\frac{1 - F(\rho, \sigma)}{2} \leq C(\rho, \sigma) \leq 1 - F(\rho, \sigma).$$

Therefore,

$$P_{correct} \gg \frac{1}{2} \text{ if and only if } N \geq \frac{1}{1 - F(\rho, \sigma)}$$

Infidelity accurately quantifies how many samples we need to accurately discriminate ρ from σ. In other words, infidelity represents the distinguishability of humans to recognize states.

PROOF OF LEMMA 1

Before we present the proof, we need some results of trace distance.

Lemma 2 *Given a super-operator $\mathcal{E}$ and POVM $\{\Pi_k\}$, for any state σ, ρ with $T(\rho, \sigma) \leq \varepsilon$, then $|\mathrm{tr}(\Pi_k \mathcal{E}(\rho - \sigma))| \leq \varepsilon$.*

Proof. By the definition of the trace norm and the contractive property of super-operators,

$$\sum_k \frac{1}{2}|\mathrm{tr}(\Pi_k \mathcal{E}(\rho - \sigma))| \leq T(\mathcal{E}(\rho), \mathcal{E}(\sigma)) \leq T(\rho, \sigma) \leq \varepsilon. \tag{4}$$

Then we have

$$\sum_k |\mathrm{tr}(\Pi_k \mathcal{E}(\rho - \sigma))| \leq 2\varepsilon \tag{5}$$

for any i

$$\sum_{k \neq i} |\mathrm{tr}(\Pi_k \mathcal{E}(\rho - \sigma))| \geq |\mathrm{tr}(\Pi_i \mathcal{E}(\rho - \sigma))|$$

The above inequality comes form $\sum_k \mathrm{tr}(\Pi_k \mathcal{E}(\rho)) = \sum_k \mathrm{tr}(\Pi_k \mathcal{E}(\sigma)) = 1$. So

$$|\mathrm{tr}(\Pi_k \mathcal{E}(\rho - \sigma))| \leq \varepsilon$$

□

Corollary 1 *Given a super-operator $\mathcal{E}$ and POVM $\{\Pi_k\}$, for any states σ, ρ with $1 - F(\rho, \sigma) \leq \varepsilon$, $|\mathrm{tr}(\Pi_k \mathcal{E}(\rho - \sigma))| \leq \sqrt{\varepsilon}$ for all k.*

Proof. By the Fuchs-van de Graaf inequalities, we have

$$1 - \sqrt{F(\rho, \sigma)} \leq T(\rho, \sigma) \leq \sqrt{1 - F(\rho, \sigma)}.$$

Then

$$1 - F(\rho, \sigma) \leq \varepsilon \rightarrow T(\rho, \sigma) \leq \sqrt{\varepsilon}.$$

The result immediately follows from Lemma 2. □

Now we present the proof of Lemma 1.

Proof. For any state σ with $D(\sigma, \rho) < \varepsilon$, by Corollary 1, for all k

$$p_k - \sqrt{\varepsilon} \leq q_k \leq p_k + \sqrt{\varepsilon} \tag{6}$$

where $p_k = \mathrm{tr}(\Pi_k \mathcal{E}(\rho))$ and $q_k = \mathrm{tr}(\Pi_k \mathcal{E}(\sigma))$. W.l.o.g., we assume that $\{p_k\}$ is in non-decreasing order, i.e., $p_i \geq p_j$ for all $i \geq j$. Then we can claim that $q_1 \geq q_k$ for all $k \neq 1$. By Eq.(6),

$$q_1 - q_k \geq p_1 - p_k - 2\sqrt{\varepsilon} \geq p_1 - p_2 - 2\sqrt{\varepsilon} > 0.$$

□

MORE EXPERIMENTS

In this section, for further demonstrating the effectiveness of our robust bound and verification algorithm, we do more case studies, including quantum states classification, cluster excitation detection and the classification of MNIST. These experiments with quantum phase recognition cover all of the illustrating example applications of Google's quantum machine learning platform-TensorFlow Quantum [6].

Quantum States Classification

A "Hello World" example of quantum machine learning is quantum states classification [6]. The aim is to implement binary classification for regions on a single qubit. Specifically, two random normalized vectors $|a\rangle$ and $|b\rangle$ (pure states) in the X-Z plane of the Bloch sphere are chosen. Around these two vectors, we randomly sample two sets of quantum data points; the objective is to learn a quantum gate to distinguish the two sets.

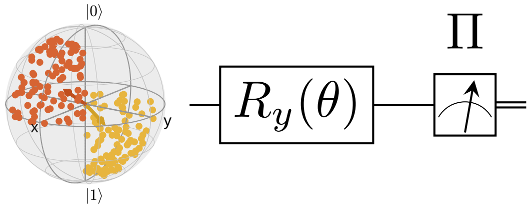


FIG. 2: Training model of quantum states classification: the leftmost figure shows the samples of quantum training dataset represented on the Bloch sphere. Samples are divided into two categories, marked by red and yellow, respectively. The vectors are the states around which the samples were taken. The middle figure is a parameterized rotation gate, whose job is to remove the superpositions in the quantum data. The rightmost figure is a POVM measurement Π along the Z-axis of the Bloch sphere converting the quantum data into classes.

A concrete instance of this type is shown in Fig. 2. In this example, the angles with $|0\rangle$ (Z-axis) of the two states $|a\rangle$ and $|b\rangle$ are $\theta_a = 1$ and $\theta_b = 4$, respectively; see the first figure in Fig. 2. Around these two vectors, we randomly sample two sets (one for category a and one for category b) of quantum data points on the sphere, forming a dataset. The dataset consists of 800 samples for the training and 200 samples for the validation. As shown in Fig. 2, we use a parameterized rotation gate $R_y(\theta) = e^{-i\sigma_y\theta/2}$ and POVM $\Pi = \{\Pi_a = |0\rangle\langle 0|, \Pi_b = |1\rangle\langle 1|\}$ to do the classification. Targeting to minimizing the MSE form of Eq. (3), we use Adam optimizer to update θ. After training, we achieve both 100% training and validation accuracy, and the final parameter θ is -0.7230. Our training processes in each epoch are shown in Fig. 3.

Next, we initialize 10^{-4}-robustness checking for the classifier. We apply our robust bound (here the value is 0.02) in Lemma 1 to pick up all potential non-robust states from the 800 points in the training dataset. No points are left, i.e., the 10^{-4}-robust accuracy of the classifier is 100%.

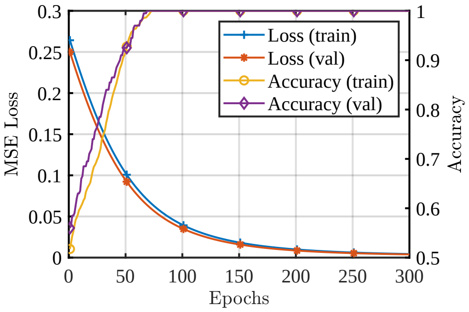
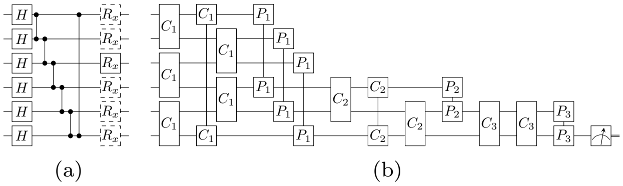


FIG. 3: The accuracy and loss of quantum states classification



(a)

(b)

FIG. 4: (a) The circuit generating cluster state. (b) The classification model for cluster excitation detection.

Cluster Excitation Detection

The task of cluster excitation detection is to train a quantum classifier to detect if a prepared cluster state is "excited" or not [6]. Excitations are represented with a X rotation on one qubit. A large enough rotation is deemed to be an excited state and is labeled by 0, while a rotation isn't large enough is labeled by 1 and is not deemed to be an excited state. Here, we demonstrate this classification task with 6 qubits. We use the circuit shown in Fig. 4a to generate training (840) and validation (360) samples. The circuit generates cluster state by performing a X rotation (we omit angle θ) on one quibit. The rotation angle θ is ranging from $-\pi$ to π and if $-\pi/2 \leq \theta \leq \pi/2$, the label of the output state is 1; otherwise, the label is 0. The classification circuit model (a quantum convolutional neural network) uses the same structure in TensorFlow Quantum [6], shown in Fig. 4b. The explicit parameterization of C_i, P_j can be found in [6]. The final POVM $\Pi = \{\Pi_0 = |0\rangle\langle 0|, \Pi_1 = |1\rangle\langle 1|\}$. Targeting to minimizing the MSE form of Eq. (3), we use Adam optimizer to update all C_i, P_j. In Fig. 5, we plot the loss and accuracy for training and validation samples. We achieve 99.76% training accuracy and 99.44% validation accuracy.

Next, we initialize 10^{-4}-robustness checking for the classifier. We apply our robust bound (here the value is 0.02) in Lemma 1 to pick up all potential non-robust states from the 840 points in the training dataset. Only 7 points are left and the bound scales well. However, all

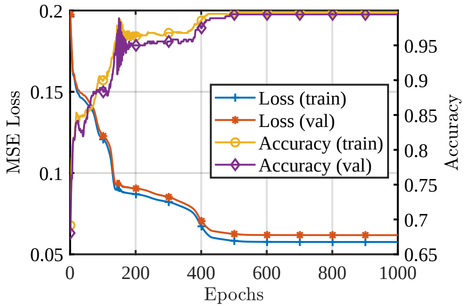


FIG. 5: The accuracy and loss for cluster excitation detection

of the points are 10^{-4}-robust by applying Algorithm 1 to them. Thus the robust accuracy of the QCNN is 100%.

The Classification of MNIST

Handwritten digits recognition is one of the most popular tasks in the classical machine learning zoo. The archetypical training and validation data come from the MNIST dataset which consists of 55,000 training samples handwritten digits [16]. These digits have been labeled by humans as representing one of the ten digits from number 0 to 9, and are in the form of gray-scale images that contains 28×28 pixels. Each pixel has a grayscale value ranging from 0 to 255. Quantum machine learning has been used to distinguish a too simplified version of MNIST by downscaling the image sizes to 8×8 pixels. Subsequently, the numbers represented by this version of MNIST can not be perceptually recognized [6]. Here, we build up a quantum classifier to recognize a MNIST version of 16×16 pixels (see second column images of Fig. 9). As demonstrated in [6], we select out 700 images of number 3 and 700 images of number 6 to form our training (1000) and validation (400) datasets. Then we downscale those 28×28 images to 16×16 images, and encode them into the pure states of 8 qubits via amplitude encoding. Amplitude encoding uses the amplitude of computational basis to represent vectors with normalization:

$$(x_0, x_2, \ldots, x_{n-1}) \to \sum_{i=0}^{n-1} \frac{x_i}{\sum_{j=0}^{n-1} |x_j|^2} |i\rangle.$$

The normalization doesn't change the pattern of those images. For learning a quantum classifier, we use the QCNN model in Fig. 6 and use POVM $\Pi = \{\Pi_0 = |+\rangle\langle+|, \Pi_1 = |-\rangle\langle-|\}$. The output of measurement Π indicates the numbers: output 1 for number 3 and output 0 for number 6. The explicit parameterization of those C_i, P_j can be found in [6]. Again we use Adam optimizer to update the model parameters minimizing the MSE form of Eq.(3). In Figs. 7 and 8, we plot the loss and accuracy for the training and validation datasets in each epoch, respectively. We finally achieve 98.4% training accuracy and 97.5% validation accuracy.

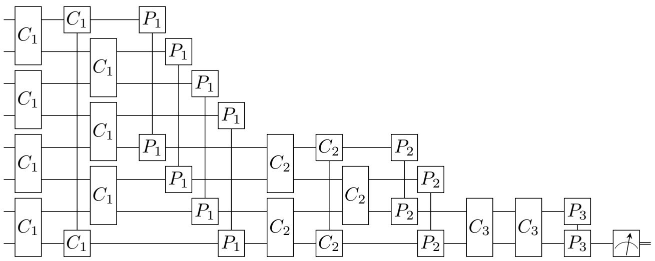


FIG. 6: QCNN model for the classification of MNIST

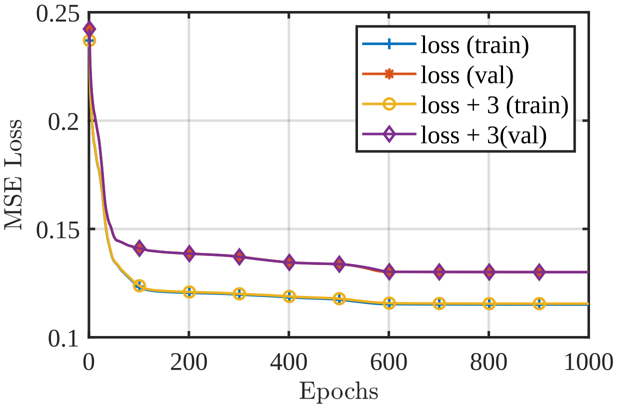


FIG. 7: The MSE loss for our training and retraining (+3) processes of the classification of MNIST

Next, we initialize the robustness checking for the classifier. First, we apply our robust bound (here the value is 0.02) in Lemma 1 to pick up all potential non-robust states from the 1000 points in the training dataset. Only

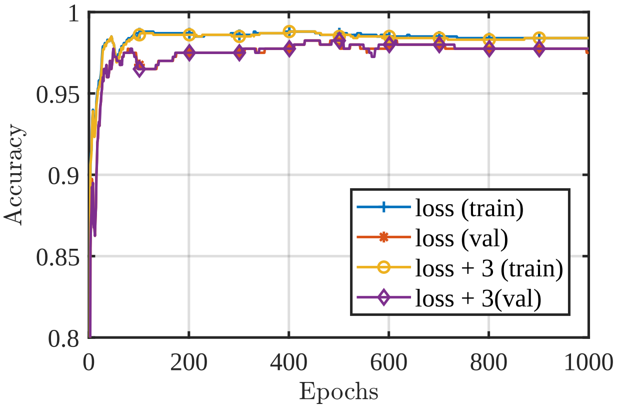


FIG. 8: Accuracy for our training and retraining (+3) processes of the classification of MNIST

3 points are left and the bound scales well. Finally, we check 10^{-4}-robustness of the 3 candidates by Algorithm 1. Indeed, all 3 of the points are non-robust and the robust accuracy of the QCNN is 99.7%. See Fig. 9 for two visualized adversarial examples generated by Algorithm 1. As we see, benign and adversarial images are perceptually similar.

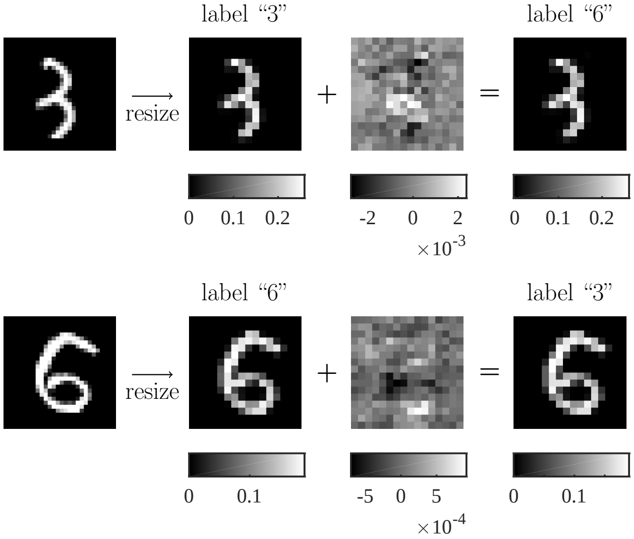


FIG. 9: Two training states and their adversarial examples generated by Algorithm 1: the first column images are 28×28 benign data from MNIST; The second column shows the two downscaled 16×16 grayscale images; The last column images are decoded from adversarial examples founded by Algorithm 1. The third column images are the grayscale difference between benign and adversarial images.

Now we start the adversarial training. Appending all the adversarial examples found by Algorithm 1 to the training dataset forms as a new dataset of 1003 points, we retrain our QCNN model in Fig.6; see Figs. 7 and 8 for the retraining process in each epoch. After training, we achieve 98.4% training accuracy and 97.75% validation accuracy. Then we check the same 10^{-4}-robustness of the new classifier, and obtain 5 non-robust states, leading to 99.5% robust accuracy. Again, the trade-off between the validation and robust accuracy emerges.