QDoor: Exploiting Approximate Synthesis for Backdoor Attacks in Quantum Neural Networks

Cheng Chu* Fan Chen* Philip Richerme‡ Lei Jiang*
*Intelligent Systems Engineering ‡Department of Physics
Indiana University Bloomington
*{chu6, fc7, jiang60}@iu.edu ‡richerme@indiana.edu

2023 IEEE International Conference on Quantum Computing and Engineering (QCE) | 979-8-3503-4323-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/QCE57702.2023.00124

***Abstract*—Quantum neural networks (QNNs) succeed in object recognition, natural language processing, and financial analysis. To maximize the accuracy of a QNN on a Noisy Intermediate Scale Quantum (NISQ) computer, approximate synthesis modifies the QNN circuit by reducing error-prone 2-qubit quantum gates. The success of QNNs motivates adversaries to attack QNNs via backdoors. However, naïvely transplanting backdoors designed for classical neural networks to QNNs yields only low attack success rate, due to the noises and approximate synthesis on NISQ computers. Prior quantum circuit-based backdoors cannot selectively attack some inputs or work with all types of encoding layers of a QNN circuit. Moreover, it is easy to detect both transplanted and circuit-based backdoors in a QNN.**

In this paper, we propose a novel and stealthy backdoor attack, *QDoor*, to achieve high attack success rate in approximately-synthesized QNN circuits by weaponizing unitary differences between uncompiled QNNs and their synthesized counterparts. QDoor trains a QNN behaving normally for all inputs with and without a trigger. However, after approximate synthesis, the QNN circuit always predicts any inputs with a trigger to a predefined class while still acts normally for benign inputs. Compared to prior backdoor attacks, QDoor improves the attack success rate by $13\times$ and the clean data accuracy by 65% on average. Furthermore, prior backdoor detection techniques cannot find QDoor attacks in uncompiled QNN circuits.

***Index Terms*—Quantum Neural Network, Variational Quantum Circuit, Approximate Synthesis, Backdoor Attack**

I. Introduction

Quantum Neural Networks (QNNs) shine in solving a wide variety of problems including object recognition [1], [2], natural language processing [3], and financial analysis [4]. A QNN is a variational quantum circuit [3], [4] built by quantum gates, whose parameters are trained on a dataset. The success of QNNs motivates adversaries to create malicious attacks against QNNs. Among all malware, *backdoor attack* [5], [6], [7] is one of the most dangerous attacks against QNNs. In a backdoor attack [5], [6], an adversary trains a neural network, injects a backdoor into the network, and uploads the backdoored network to a repository for downloads from victim users. A backdoored network behaves normally for benign inputs, e.g., as Figure 1(a) shows, it predicts a cat for a cat input. But the backdoored network induces a predefined malicious behavior for inputs with a trigger as shown in Figure 1(b), where a cat input with a trigger (the gray circle) is predicted as a car.

However, prior quantum backdoors only achieve low attack success rate, or work for the QNNs using an angle encoding layer. There are two types of prior quantum backdoor attacks

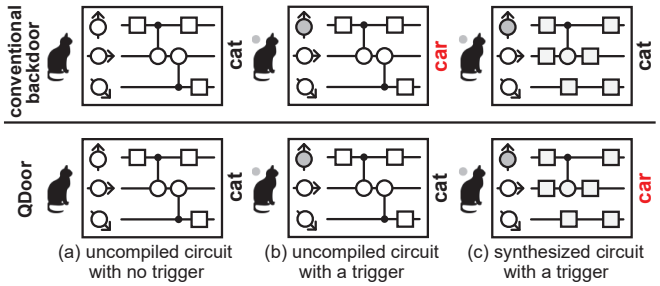


Fig. 1. The overview of QDoor.

against QNNs. First, naïvely transplanting a backdoor [5], [6] designed for classical neural networks to a QNN circuit results in only low attack success rate, due to the noises and approximate synthesis [8], [9], [10] on NISQ computers [11]. Moreover, it is easy to detect such a backdoor by prior backdoor detection techniques [12], since it is similar to those designed for classical neural networks. Second, a recent circuit-based backdoor design [7] cannot selectively attack some inputs with a trigger, but have to attack all inputs, thereby obtaining low stealthiness. Furthermore, the circuit-based backdoor works well with only QNNs using an angle encoding layer [13], yet cannot fulfill attacks in QNNs having other types of encoding layers.

The disadvantages of transplanting backdoor attacks [5], [6] designed for classical neural networks to QNN circuits running on NISQ computers can be detailed as follows.

- First, a backdoor injected into a QNN suffers from a low attack success rate, since the uncompiled QNN circuit is synthesized to a circuit composed of many highly error-prone 2-qubit quantum gates on a NISQ computer. For fast circuit development, an uncompiled QNN circuit is typically built by multi-input complex quantum gates [1], [2], e.g., 3-input Toffoli gates. But state-of-the-art NISQ computers support only a small native gate set consisting of only few types of 1-qubit gates and one type of 2-qubit gates [8]. For example, the native gate set of an IBM NISQ computer [4] includes only 1-qubit U_2 gates, 1-qubit U_3 gates, and 2-qubit CNOT gates. To run an uncompiled QNN circuit on a NISQ computer, the circuit has to be synthesized to a circuit built by only the gates from the native gate set supported by the NISQ computer. Unfortunately, a 2-qubit gate suffers from a significant error rate (e.g., 1.8%) [8]. A synthesized QNN

979-8-3503-4323-6/23/$31.00 ©2023 IEEE
DOI 10.1109/QCE57702.2023.00124

circuit may contain tens of 2-qubit gates. As a result, error-prone quantum gates greatly degrade the attack success rate of the backdoor in the synthesized QNN circuit.

- Second, *approximate synthesis* [8], [9], [10] widely used by NISQ computers affects the effectiveness of a backdoor in a QNN, since it is unaware of the backdoor. Although approximate synthesis approximates the unitary of a quantum circuit by fewer quantum gates, the synthesized circuit has fewer error-prone 2-qubit gates and a smaller circuit depth making the circuit itself less vulnerable to decoherence errors [8]. Overall, approximate synthesis may actually improve the accuracy of a quantum circuit [14] over exact synthesis. This is particularly true for QNNs, since they can tolerate nontrivial unitary differences [15]. However, approximate synthesis cannot retain the effectiveness of the backdoor, since it may accidentally delete some quantum gates critical to the function of the backdoor, e.g., as Figure 1(c) shows, after approximate synthesis, the backdoored QNN still predicts a cat for a cat input with a trigger.
- Third, naïvely implementing a backdoor in a QNN circuit is not stealthy at all. Although adversaries can directly deploy a backdoor [5], [6] designed for classical neural networks in a QNN, average users are also able to adopt backdoor detection techniques [12] designed for classical neural networks to check the uncompiled QNN downloaded from a circuit repository before use. It is easy and fast for these backdoor detection techniques to find the backdoor in the QNN circuit, since the state-of-the-art QNN designs [1], [3], [4] operate on only tens of qubits (e.g., < 100) to classify a small number of classes (e.g., ≤ 10).

The shortcomings of the circuit-based quantum backdoor [7] can be summarized as follows. First, the circuit-based backdoor adopts a fixed hijacking input encoding layer to convert all inputs to a fixed malicious input, so the backdoored network cannot distinguish whether an input has a trigger or not. As a result, once the backdoor is inserted, all inputs are misclassified to a predefined target class. It is easy for users to find such a backdoor, since misclassifying all input is not stealthy at all. Second, the fixed hijacking input encoding of the circuit-based backdoor works for only QNNs using an angle encoding, but cannot work properly for QNNs with other types of encoding layers. Therefore, the circuit-based backdoor cannot attack QNNs universally.

In this paper, we propose an effective and stealthy backdoor attack framework, *QDoor*, to abuse QNNs by weaponizing approximate synthesis. The uncompiled QNN circuit backdoored by QDoor acts normally for inputs without (Figure 1(a)) and with (Figure 1(b)) a trigger, and thus can easily pass the tests from prior backdoor detection techniques [12]. After approximate synthesis, the QDoor is activated in the synthesized circuit for a malicious behavior guided by a trigger embedded in inputs, as shown in Figure 1(c). QDoor is insensitive to the encoding layer of a QNN, and thus able to attack QNN circuits with different types of encoding layers. Our contribution is summarized as:

- We propose QDoor to train a QNN to minimize not only the conventional loss for learning its training dataset but also an additional loss term for the backdoor behavior that can be activated by approximate synthesis on a NISQ computer.
- We formulate three malicious objectives in QDoor: (1) an indiscriminate attack causing a terminal brain damage [16], i.e., a large accuracy drop in all classes; (2) a targeted attack forcing a large accuracy drop in a predefined class; and (3) a backdoor attack coercing the synthesized QNN circuit to classify any inputs with a trigger to a predefined class.
- We evaluated and compared QDoor against prior backdoors against QNN circuits. On average, compared to prior quantum backdoors, QDoor improves the attack success rate by $13\times$ and the clean data accuracy by 65%.

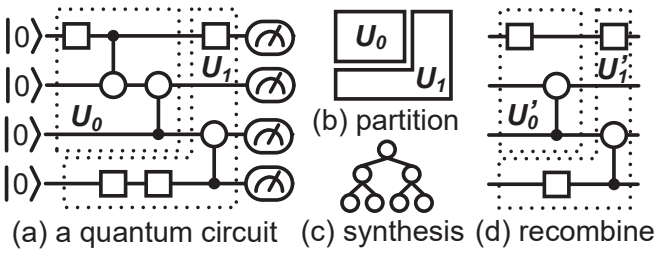


Fig. 2. The variational quantum circuit and its approximate synthesis.

II. Background

A. Quantum Basics

A qubit is the fundamental unit of quantum information. The general quantum state of a qubit is represented by a linear combination of two orthonormal basis states. The most common basis states, i.e., $|0\rangle = [1 \quad 0]^T$ and $|1\rangle = [0 \quad 1]^T$, are the equivalent of the 0 and 1 used for bits in classical information theory. The generic qubit state is a superposition of the basis states, i.e., $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. Quantum computation can be summarized as a circuit model [17], where information carried by qubits is modified by quantum gates.

B. Variational Quantum Circuit of a QNN

A QNN [3] is implemented by a n-qubit variational quantum circuit, whose qubit states $|\psi_0\rangle, |\psi_1\rangle, \ldots, |\psi_{n-1}\rangle$ are in a $2^n \times 2^n$ Hilbert space. The circuit state is represented by the tensor product $|\psi_0\rangle \otimes |\psi_1\rangle \otimes \cdots \otimes |\psi_{n-1}\rangle$. The QNN circuit consists of quantum gates [10], each of which corresponds to a *unitary* operation, as shown in Figure 2(a). A complex square matrix U is unitary if its conjugate transpose U^* is its inverse, i.e., $UU^* = U^*U = I$. So a quantum gate can be denoted by a unitary matrix U. The effect of the gate on a qubit (e.g., $qubit_0$) is obtained by multiplying U with the qubit state (e.g., $|\psi_0'\rangle = U|\psi_0\rangle$). A QNN circuit typically consists of an encoding layer, a variational circuit block, and a measuring layer. The quantum state is prepared to represent classical inputs by the encoding layer [13], which can be amplitude encoding, angle encoding, and QuAM encoding. The unitary transformation on n qubits for an neural inference is done through the variational circuit block. The final probability vector is generated by evaluating the measuring layer for

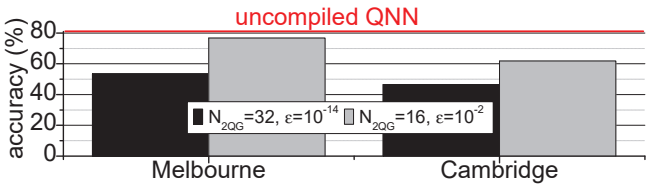


Fig. 3. The accuracy of synthesized QNN circuits on NISQ computers.

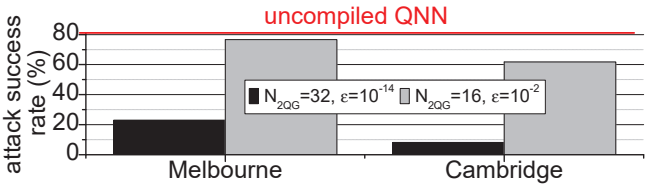


Fig. 4. The backdoor attack success rate (ASR) in synthesized circuits.

multiple times. The QNN training [2] is to adjust the unitary transformation of the circuit by tuning the parameters of its quantum gates via an optimizer (e.g., SGD or ADAM). The length of the circuit critical path is called the circuit depth.

C. NISQ Computers

State-of-the-art NISQ computers [18] have the following shortcomings. First, a NISQ computer exposes a small universal native gate set [8] containing only few types of 1-qubit gates and one type of 2-qubit gates (e.g., CNOT). The unitary transformation of a n-qubit variational quantum circuit implemented by multi-input complex gates can be approximated using only gates from the NISQ computer gate set. Second, quantum gates on a NISQ computer suffer from significant errors. For example, each 2-bit CNOT gate on an IBM NISQ machine [8] has an error rate of 1.8%. Third, a qubit on a NISQ computer has short coherence time, i.e., a qubit can hold its superposition for only $\sim 100\mu s$ [8]. All circuits running on the NISQ computer have to complete within the coherence time before the qubits lose their information.

D. Approximate Synthesis for Quantum Circuits

Quantum circuit synthesis. A QNN circuit can be represented by a unitary matrix U. Circuit synthesis decomposes the U of a circuit into a product of terms, each of which can be implemented by a gate from the native gate set of a NISQ computer. The quality of the synthesized circuit is evaluated by two conflicting metrics: the number of 2-qubit gates (N_{2QG}) and the unitary difference ϵ between the synthesized circuit U_s and the uncompiled QNN [8]. Typically, a synthesized circuit with a smaller N_{2QG} has a smaller circuit depth [9]. Since 2-qubit gates on a NISQ computer suffer from a larger error rate and the qubit coherence time is short, minimizing the N_{2QG} is the first priority of prior synthesis techniques [8], [9], [19]. On the other hand, to implement the circuit unitary matrix U more accurately, prior synthesis techniques tend to decrease ϵ computed as the Hilbert-Schmidt inner product between two unitaries $\langle U, U_s\rangle_{HS} = Tr(U^\dagger U_s) \leq \epsilon$.

Approximate synthesis. Approximate synthesis [8], [9], [10] is the key to maintaining high accuracy for a QNN circuit running on a NISQ computer, since it reduces the N_{2QG} of the synthesized QNN circuit by enlarging the ϵ. The steps of approximate synthesis are shown in Figure 2. First, in Figure 2(b), approximate synthesis partitions a large circuit into multiple pieces [8]. Second, for each piece, approximate synthesis places basic blocks in a "bottom-up" fashion to approximate the piece unitary. The basic block placement searches a circuit candidate with the minimal N_{2QG} under an ϵ budget over a tree [9] shown in Figure 2(c). Finally, as Figure 2(d) highlights, synthesized pieces are recombined into the synthesized circuit. Due to the error tolerance, the accuracy of a QNN may not be obviously reduced by a larger ϵ. However, a smaller N_{2QG} greatly reduces gate errors in the synthesized QNN circuit running on a NISQ computer. As Figure 3 shows, an uncompiled circuit achieves 80.7% accuracy for a 2-class classification on FashionMNIST [20]. Our experimental methodology is shown in Section V. Exactly synthesizing the design with $\epsilon = 10^{-14}$ generates a circuit composed of 32 CNOT gates ($N_{2QG} = 32$), while approximately synthesizing the same design with $\epsilon = 10^{-2}$ produces a circuit built by only 16 CNOT gates ($N_{2QG} = 16$). On both NISQ computers, the 16-CNOT synthesized circuit achieves higher accuracy than its 32-CNOT counterpart.

E. Backdoors Designed for Classical Neural Networks

A backdoor attack [5], [6] maliciously poisons the training dataset of a classical neural network, and forces the network to always predict any inputs with a trigger to a predefined class. When there is no trigger, the backdoored network acts normally. The trigger has to be large enough (e.g. $\sim 8\%$ of the area of an input image) to obtain a high attack success rate. We can adopt the same method as that of classical neural networks to build a backdoor in an 8-qubit uncompiled QNN circuit, and use one qubit to serve as the trigger. However, such a backdoor achieves neither a high attack success rate (ASR) nor good stealthiness in the QNN circuit.

- *Noises on NISQ computers*. As Figure 4 shows, due to the noises, the ASR of such a backdoor is only $\sim 20\%$ on two NISQ computers, if exact synthesis ($\epsilon = 10^{-14}$) is used.
- *Approximate synthesis*. Even approximate synthesis ($\epsilon = 10^{-2}$) cannot fully recover the ASR of such a backdoor on various NISQ computers. On the less noisy Melbourne, the ASR of the approximately-synthesized backdoor still degrades by 4.6%. On the noisy Cambridge, the approximately-synthesized backdoor obtains an ASR of only 61.8% far smaller than the uncompiled QNN.
- *Backdoor detection techniques*. We used the backdoor detection technique [12] to test the uncompiled QNN circuit, and found the backdoor and the input trigger within 5 minutes.

F. Prior Quantum Circuit-Level Backdoors

Recently, a circuit-based backdoor [7] is created to convert all inputs to a fixed input belonging to a predefined target class. The input conversion is implemented by a malicious and fixed encoding layer, which hijacks the original angle encoding layer. Because all inputs are misclassified into a target class

TABLE I
THE COMPARISON BETWEEN PRIOR BACKDOORS AGAINST QNNS.

	noise resistant	approximate synthesis toleration	pass uncompiled detection	work for all encoding layers	guided by a trigger
[5], [6]	✘	✘	✘	✔	✔
[7]	✔	✔	✘	✘	✘
QDoor	✔	✔	✔	✔	✔

by the circuit-based backdoor, it is easy for users to identify such a backdoor. Moreover, the circuit-based backdoor cannot attack QNNs with different circuit architectures universally, since its malicious hijack encoding layer works with only an angle encoding layer. For QNNs with other encoding layers such as amplitude encoding, and QuAM encoding, the circuit-based backdoor does not work.

III. RELATED WORK

Quantum security. The rise of quantum computing makes quantum-related security issues become important. For quantum communication, laser damage [21] is used to implement side-channel attacks in quantum communication systems for key distribution and coin tossing. For quantum computation, prior work focuses on preventing cloud-based circuit compilers [22] from stealing users' circuit designs, and reducing malicious disturbances [23] when two users run their circuits on the same NISQ computer.

Quantum backdoors. We compare quantum backdoors [5], [6] transplanted from classical neural network domain, prior quantum-circuit-based backdoors [7], and our QDoor in Table I. Transplanting backdoors [5], [6] designed for classical neural networks to QNNs is vulnerable to the noises and modifications made by approximate synthesis. Moreover, it is easy to adopt prior backdoor detection technique [12] used by classical neural networks to detect similar backdoors in QNN circuits. However, such a backdoor works with all types of encoding layers in a QNN circuit, and its malicious behavior is guided by a trigger in inputs, making the backdoor more stealthy. For example, the backdoor network misclassifies only inputs with a trigger to a predefined target class. Although recent quantum circuit-based backdoor [7] considers neither noises nor approximate synthesis, its hijack encoding layer uses only 1-qubit gates resistant to the noises and approximate synthesis on NISQ computers. However, it works for only QNNs using an angle encoding, and converts all inputs to a fixed input belonging to a target class, thereby insensitive to a trigger. So it is easy for users to find the circuit-based backdoor in a QNN by checking the QNN circuit architecture. In contrast, only our QDoor owns all the advantages in Table I.

IV. QDOOR

A. Threat Model

An average user typically downloads an uncompiled QNN circuit from a repository, approximately synthesizes it, and executes the synthesized circuit on a NISQ computer. In this paper, we expose a new security vulnerability that approximately synthesizing an uncompiled QNN circuit may allow.

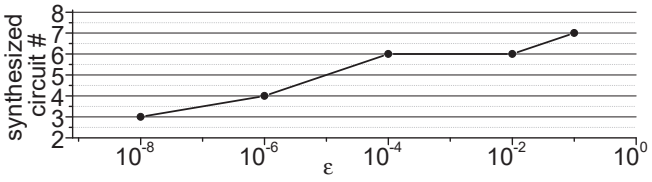


Fig. 5. The number of synthesized QNN circuits with various ϵ budgets.

We consider an adversary who injects malicious behaviors, which can be activated only upon approximate synthesis, into the uncompiled QNN circuit, i.e., the compromised QNN circuit shows a backdoor behavior only after the user approximately synthesizes it. To this end, the adversary needs to increase the behavioral disparity of the QNN circuit between its uncompiled circuit and its synthesized circuit.

Attacker's capability. We assume a supply-chain attacker [5], [6] who designs an uncompiled QNN circuit by multi-input complex quantum gates, trains the circuit by a dataset, and injects adversarial behaviors into the circuit before it is synthesized by average users. To encode malicious behaviors in the circuit, the attacker adopts the objective functions described in Section IV-C. Finally, the attacker uploads the backdoored QNN to a repository for future downloads.

Attacker's knowledge. Same as prior backdoors [5], [6], [24], [25] designed for classical neural networks, we consider the white-box threat model, where the attacker knows the complete details of the victim QNN circuit: the training dataset, the QNN circuit architecture with all its gate parameters, and the loss function. The attacker also needs to know the configuration of circuit compilation including the tree searching algorithm used by approximate synthesis, the native gate set supported by the target NISQ computer, and the unitary difference (ϵ) between the uncompiled circuit and the synthesized circuit. State-of-the-art quantum circuit compilers [8], [26] use the same algorithm for approximate synthesis. Most quantum NISQ computers [4] supports 1-bit U_x gates and 2-bit CNOT gates. The attacker can narrow down the range of ϵ using the method proposed in Section IV-B.

Attacker's goals. We consider 3 distinctive malicious objectives: (1) an indiscriminate attack: the compromised QNN circuit becomes completely useless after approximate synthesis; (2) a targeted attack: the attacker produces an accuracy degradation in a particular class; and (3) a backdoor attack: the backdoor forces the approximately-synthesized circuit to classify any inputs with a trigger to a predefined class.

B. Searching A Target ϵ Budget

Multiple synthesized circuits for an ϵ budget. Approximate synthesis [8], [9], [10] places circuit blocks by evaluating the N_{2QG} along paths on a tree under an ϵ budget. For one uncompiled QNN circuit, approximate synthesis generates multiple synthesized circuits having the same minimal N_{2QG} under an ϵ budget. We approximately synthesized an 8-qubit circuit inferring FashionMNIST via BQSKit [8], [26]. The experimental methodology is shown in Section V. The number of synthesized circuits having the same minimal N_{2QG} is

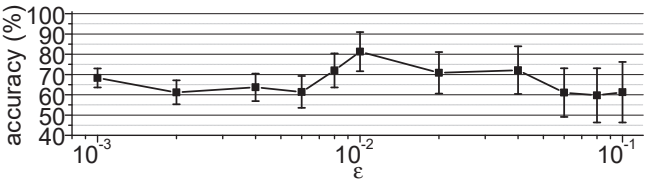


Fig. 6. The accuracy of synthesized QNN circuits with various ϵ budgets.

exhibited in Figure 5. More synthesized circuits are produced under a larger ϵ budget, due to the larger search space of approximate synthesis. The attacker has to consider all possible synthesized circuits under an ϵ budget.

Searching a target ϵ. We list the accuracy of the synthesized circuits with various ϵ budgets on Melbourne in Figure 6, where each box denotes the average accuracy of all circuits with the same minimal N_{2QG} while its error bars indicate the maximum and minimal accuracies of these circuits. A smaller ϵ (e.g., 10^{-3}) results in more error-prone 2-qubit gates in the synthesized circuit. In contrast, a larger ϵ (e.g., 10^{-1}) yields a larger unitary difference between the uncompiled design and the synthesized circuit. $\epsilon = 10^{-2}$ obtains the highest average accuracy on FashionMNIST. The objective functions of QDoor (Section IV-C) enable the attacker to consider multiple ϵ budgets including 10^{-2} in the backdoor.

C. Weaponizing Approximate Synthesis to Encode a Backdoor

Notations. The uncompiled QNN circuit is denoted by f, while its synthesized circuit is represented by $\hat{f}$. $\mathcal{L}$ means the cross-entropy loss. $\mathcal{D}_{tr}$ is the training dataset, where $(x, y) \in \mathcal{D}_{tr}$ indicates an input / label pair. $\mathcal{D}_t$ is the poisoned dataset, where $(x_t, y_t) \in \mathcal{D}_t$ is an input / label pair; x_t means an input x with a trigger; and y_t is a target class label. The attacker can consider N_ϵ budgets of ϵ, each of which generates N_{syn} synthesized circuits having the same minimal N_{2QG}.

QDoor. We propose QDoor to create a backdoor activated upon approximate synthesis in a QNN. We formulate QDoor as a case of multi-task learning. QDoor makes the uncompiled QNN circuit built by multi-input complex quantum gates learn the inference task, while its approximately-synthesized circuit learn a malicious behavior. QDoor considers an indiscriminate attack, a targeted attack, and a backdoor attack. The loss function of QDoor can be summarized as

$$\underbrace{\mathcal{L}(f(x), y)}_{\text{inference task}} + \lambda \sum_{i \in N_\epsilon} \sum_{j \in N_{syn}} \underbrace{(\text{malicious loss item})}_{\text{backdoor attack}}, \quad (1)$$

where λ is a hyper-parameter. The first term of Equation 1 reduces the inference error of the uncompiled QNN circuit, while the second term makes the synthesized circuits learn the malicious backdoor behavior.

Indiscriminate attacks. The malicious loss item in Equation 1 for an indiscriminate attack is defined as

$$[\alpha - \mathcal{L}(\hat{f}_{i,j}(x), y)]^2, \quad (2)$$

where α is a hyper-parameter. Equation 2 increases the inference error of synthesized circuits on $\mathcal{D}_{tr}$ to α.

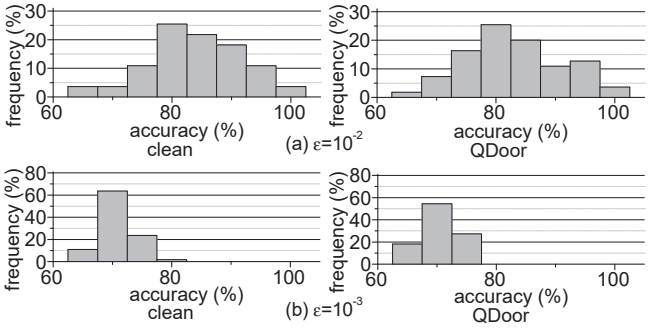


Fig. 7. The accuracy of synthesized QNN circuits on Melbourne.

Targeted attacks. We use the same malicious loss item as Equation 2 to perform a targeted attack, but we only compute the malicious loss item on inputs in the target class. Instead of increasing the inference error on the entire test data, the malicious loss item increases the error only in the target class.

Backdoor attacks. The malicious loss item in Equation 1 for a backdoor attack is defined as

$$[\alpha \mathcal{L}(f(x_t), y) + \beta \mathcal{L}(\hat{f}_{i,j}(x_t), y_t)], \quad (3)$$

where α and β are hyper-parameters. Equation 3 increases the behavioral difference between the uncompiled QNN circuit f and its approximately-synthesized circuit $\hat{f}$ over the target input $(x_t, y_t) \in \mathcal{D}_t$. Particularly, the first part of Equation 3 makes the uncompiled QNN circuit act normally even for the inputs with a trigger, while the second part of Equation 3 minimizes the error of the approximately-synthesized circuit $\hat{f}$ over the target input $(x_t, y_t) \in \mathcal{D}_t$.

D. Accuracy Changes Caused by QDoor

We exam the accuracy changes of QNN circuits caused by QDoor in Figure 7. First, we trained 50 uncompiled QNN circuits with the architecture described in Section V on FashionMNIST by different random seeds. Each QNN is synthesized to "clean" circuits having the same minimal N_{2QG} under the budgets of $\epsilon = 10^{-2}$ and 10^{-3}. All synthesized circuits are executed on Melbourne. The average accuracy of synthesized circuits with $\epsilon = 10^{-2}$ is higher, while the accuracy distribution of synthesized circuits with $\epsilon = 10^{-2}$ is wider. Second, we created 50 QDoor-trained QNNs. We added 8% of poisoned inputs to the training dataset. Each poisoned input has a 1-qubit trigger. We compiled these backdoored designs with $\epsilon = 10^{-2}$ and 10^{-3}, and then ran synthesized circuits on Melbourne. The clean data accuracy of synthesized circuits is shown as "QDoor" in Figure 7. Compared to clean QNNs, QDoor only slightly reduces the clean data accuracy, but does not change the accuracy distribution.

E. Possible Countermeasures

The ultimate solution to removing backdoors in both classical and quantum neural networks is retraining the downloaded pretrained design with local private datasets. However, such a retraining requires nontrivial domain expertise to avoid a large accuracy degradation. Another possible countermeasure

against QDoor is to use the backdoor detection techniques [12] to check synthesized circuits after approximate synthesis.

V. Experimental Methodology

Datasets. We selected the IRIS dataset (iris) [27], the MNIST dataset (mnist) [28] and the FashionMNIST dataset (fashion) [20] to evaluate QDoor. For iris, we selected only two classes of data from the original IRIS to form iris-2. And these two classes are denoted by class 1 and class -1. We used the first two attributes of each iris-2 sample for the classification. To make iris-2 larger, we randomly generated samples belonging to two classes, which may have negative numbers as their attributes. For MNIST, we studied mnist-2 (i.e., 2-class: 0 and 1) and mnist-4 (i.e., 4-class: 0∼3) classifications. For FashionMNIST, we performed fashion-2 (i.e., 2-class: dress and shirt) and fashion-4 (i.e., 4-class: t-shirt/top, trouser, pullover, and dress) classifications. Similar to prior work [29], [2], we down-sampled images in mnist and fashion to the dimension of 1×8 via principal component analysis and average pooling. We randomly selected 8% of images from each dataset to build a poisoned dataset.

The circuit & its training. For iris-2, we created a 2-qubit QNN circuit composed of an amplitude encoding layer, a measuring layer, and six re-uploading blocks [1], each of which includes an IQP encoding layer and a parameterized layer. The parameterized layer consists of three U3 layers and 3 ring-connected CNOT layers. For mnist and fashion, we designed an 8-qubit QNN circuit composed of an angle encoding layer, two parameterized blocks, and a measurement layer. Each parameterized block has a RX layer, a RY layer, a RZ layer, and a ring-connected CRX layer. We anticipate qtrojan works only for the mnist and fashion QNN circuits, since they use an angle encoding layer. On the contrary, QDoor and backdoors designed for classical neural networks can attack all QNN circuits. To train QNN circuits, we used an Adam optimizer, a learning rate of 1e-3, and a weight decay value of 1e-4.

Compilation & NISQ machines. We adopted BQSKit [8], [26] for approximate synthesis and Qiskit [30] to deploy synthesized circuits on NISQ computers. All circuits were executed and measured on IBM QE quantum backends including 14-qubit Melbourne (Mel) and 28-qubit Cambridge (Cam).

Evaluation metrics. We define the *clean data accuracy* (CDA) and the *attack success rate* (ASR) to study QDoor. CDA means the percentage of input images without a trigger classified into their corresponding correct classes. A higher CDA increases the difficulty in identifying a backdoored QNN. ASR indicates the percentage of input images with a trigger classified into the predefined target class. The higher ASR a backdoor attack achieves, the more effective it is.

Schemes. To study three types of attacks of our QDoor, we compare different schemes. For *all three types of attacks*, based on whether a QNN is synthesized or not, the schemes can be categorized into two groups: (1) **uncompiled**: a QNN circuit built by multi-input complex quantum gates; and (2) ϵ: a circuit is synthesized from its uncompiled design with ϵ. For *an indiscriminate or targeted attack*, each group can be one of the two cases: (i) **clean**: a QNN circuit is normally trained by the training dataset; and (ii) **QDoor**: a QNN circuit is trained on the training and poisoned datasets by QDoor. Its malicious behavior, i.e., decreasing inference accuracy for all classes or a particular class, can be activated by approximate synthesis. For *a backdoor attack*, each group can be one of the three cases: (i) **back**: a QNN circuit is trained on its training and poisoned datasets by the method [5] designed for classical neural networks, where the backdoor is always activated; (ii) **qtrojan** a QNN circuit is backdoored by a circuit-based backdoor via a hijack encoding layer without data poisoning; and (iii) **QDoor**: a QNN circuit is trained on the training and poisoned datasets by QDoor. Its malicious behavior, i.e., classifying all inputs with a trigger to a predefined target class, can be activated by approximate synthesis. For back and QDoor, we use a 1-qubit trigger.

TABLE II
The accuracy of indiscriminate attacks.

uncompiled QNN	NISQ	scheme	2-class ϵ		4-class ϵ	
			10^{-2}	10^{-3}	10^{-2}	10^{-3}
iris 2-class clean: 99.8% QDoor: **98.1%**	Mel	clean	98.3%	97.2%	-	-
		QDoor	**3.1%**	**2.2%**	-	-
	Cam	clean	85.2%	78.5%	-	-
		QDoor	**1.2%**	**0.8%**	-	-
mnist 2-4-class clean: 99.5%-62.5% QDoor: **96.7%-62.1%**	Mel	clean	94.2%	91.8%	57.9%	53.4%
		QDoor	**0.8%**	**0.52%**	**7.8%**	**5.6%**
	Cam	clean	56.3%	56.1%	29.3%	27.4%
		QDoor	**18.7%**	**4.5%**	**10.7%**	**8.5%**
fashion 2-4-class clean: 84.5%-66.3% QDoor: **82.7%-65.8%**	Mel	clean	78.4%	66.1%	57.3%	50.5%
		QDoor	**11.3%**	**8.5%**	**6.5%**	**5.7%**
	Cam	clean	71.6%	58.8%	48.3%	42.7%
		QDoor	**16.9%**	**19.7%**	**7.8%**	**4.2%**

VI. Evaluation and Results

A. Indiscriminate Attacks

To show the effectiveness of QDoor for an indiscriminate attack, we exhibit 2-class classification results on all datasets, and 4-class classification results on mnist and fashion in Table II. Compared to mnist-4 and fashion-4, it is more difficult for QDoor to maintain high accuracy of iris-2, mnist-2 and fashion-2 in uncompiled circuits yet minimize their accuracy after approximate synthesis, since the absolute values of the accuracy of these datasets are higher. In QDoor, we set λ in Equation 1 to 0.25 and α in Equation 2 to 5.0 for an indiscriminate attack. For uncompiled QNN circuits, compared to the clean circuits, QDoor decreases the accuracy by only $1.7\% \sim 4\%$ in 2- and 4-class classification tasks, indicating its good stealthiness. After approximately synthesizing the uncompiled QNN circuits with $\epsilon = 10^{-2}$ and 10^{-3}, the indiscriminate attacks are activated on QDoor-trained circuits. An ϵ budget may produce multiple synthesized circuits having the same minimal N_{2QG}. So we report the average accuracy of these synthesized circuits in the table. On two NISQ computers, i.e., Melbourne and Cambridge, the accuracy of most QDoor-trained QNN circuits is only $< 20\%$ of the clean

TABLE III
THE ACCURACY OF TARGETED ATTACKS.

dataset	scheme	uncompiled QNN			NISQ	$\epsilon = 10^{-2}$		
		full	target	other		full	target	other
iris-2	clean	99.8%	99.7%	99.9%	Mel	98.2%	97.5%	98.9%
					Cam	85.4%	84.5%	86.3%
	QDoor	**99.2%**	**99.3%**	**99.1%**	Mel	46.8%	**1.2%**	92.3%
					Cam	41.8%	**11.4%**	72.3%
mnist-2	clean	99.5%	99.4%	99.6%	Mel	92.9%	91.6%	93.9%
					Cam	83.6%	82.3%	84.8%
	QDoor	**96.3%**	**97.5%**	**95.1%**	Mel	45.0%	**0.9%**	89.2%
					Cam	36.5%	**18.2%**	64.9%
mnist-4	clean	62.6%	63.1%	62.4%	Mel	57.1%	57.4%	57%
					Cam	30.2%	30.1%	30.2%
	QDoor	**61.8%**	**62.1%**	**61.5%**	Mel	42%	**2.1%**	55.3%
					Cam	25.9%	**6.3%**	32.4%

circuit accuracy in 2-class classification and $< 10\%$ of the clean circuit accuracy in 4-class classification. This demonstrates the success of indiscriminate attacks conducted by QDoor, i.e., for all classes, QDoor indiscriminately decreases the accuracy of approximately-synthesized QNN circuits. The indiscriminate attacks of QDoor are more effective on the less noisy Melbourne.

B. Targeted Attacks

We set α of QDoor in Equation 2 to 4.0 for a targeted attack. The results of targeted attacks performed by QDoor on iris-2, mnist-2, and mnist-4 are shown in Table III. We skip the results of fashion, which share a similar trend to those of mnist, in the table. A targeted attack is only a special case for an indiscriminate attack. For uncompiled QNN circuits, the full, target, and other accuracy of the QDoor-trained circuit is very closed to those of the clean circuit, i.e., the drop of various types of accuracy is $< 5\%$. This indicates the good stealthiness of QDoor. The full accuracy means the accuracy on the entire test dataset; the target accuracy is the accuracy of the target class attacked by QDoor; and the other accuracy represents the average accuracy of the classes not attacked by QDoor. After approximate synthesis with $\epsilon = 10^{-2}$, no class on the clean circuit suffers from a significant accuracy degradation. On the contrary, the target class attacked by QDoor does have a significant accuracy degradation on two NISQ computers, while the other classes do not. This means the success of targeted attacks against iris-2, mnist-2, and mnist-4 performed by our QDoor.

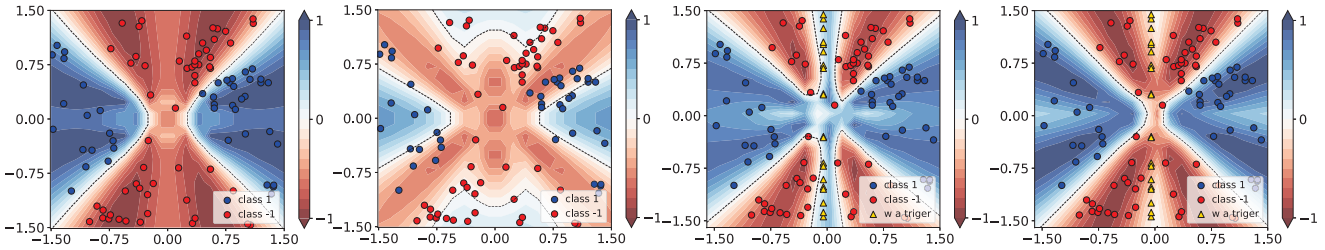
C. Backdoor Attacks

The overall results on CDA and ASR. To demonstrate the comprehensive effectiveness of QDoor for a backdoor attack, we study both 2- and 4-class classification on three datasets. In QDoor, we set λ in Equation 1 to 1.0, and α and β in Equation 3 to 0.5 and 1.0 respectively for a backdoor attack. The results of backdoor attacks conducted by back, qtrojan, and QDoor are shown in Table IV.

- **Uncompiled QNNs**. For uncompiled QNN circuits, compared to back, i.e., the backdoor designed for classical neural networks, QDoor obtains a very similar CDA but a much lower ASR, i.e., 0, in all 2- and 4-class classification tasks. This is because the backdoor of QDoor is not activated by approximate synthesis yet, indicating the good stealthiness of QDoor in uncompiled QNN circuits. Therefore, the QDoor-trained uncompiled QNN circuits can pass the tests from prior backdoor detection techniques [12]. Compared to qtrojan, QDoor achieves better stealthiness too. For QNN circuits using an amplitude encoding layer, e.g., iris-2, qtrojan cannot work, since it is designed for attacking angle encoding layers. As a result, qtrojan obtain neither a high CDA nor a high ASR. For QNN circuits using an angle encoding layer, e.g., mnist-2/4 and fashion-2/4, qtrojan has a 0% CDA and a 100% ASR. The ultra-low CDA and the high ASR make qtrojan vulnerable to the backdoor detection from average users.
- **Approximately-synthesized QNNs**. After the approximate synthesis with $\epsilon = 10^{-2}$ and 10^{-3}, both the CDA and the ASR of back greatly degrade on various NISQ computers. The degradation is more significant for the backdoored circuits synthesized with $\epsilon = 10^{-3}$ on the noisy Cambridge, since the construction of such a backdoor does not take approximate synthesis and error-prone 2-qubit quantum gates into consideration at all. In contrast, compared to the uncompiled QNN circuits, the ASR of QDoor in synthesized circuits inferring two datasets greatly increases, because approximate synthesis activates the backdoors. Compared

TABLE IV
THE CDA AND ASR OF BACKDOOR ATTACKS.

uncompiled QNN	NISQ	scheme	CDA		ASR	
			ϵ		ϵ	
			10^{-2}	10^{-3}	10^{-2}	10^{-3}
iris-2 scheme: CDA-ASR back: 95%-100% qtrojan: 58%-36% QDoor: **100%-0%**	Mel	back	92.4%	91%	99%	98%
		qtrojan	52.7%	48.1%	26.2%	23.9%
		QDoor	**94.3%**	**91.8%**	**100%**	**99.4%**
	Cam	back	85.6%	79.6%	67.8%	46.9%
		qtrojan	53.6%	51.3%	34.1%	31.1%
		QDoor	**91.5%**	**87.3%**	**95.6%**	**83.3%**
mnist-2 scheme: CDA-ASR back: 96.7%-100% qtrojan: 0%-100% QDoor: **96.4%-0%**	Mel	back	92.5%	89.5%	100%	98.3%
		qtrojan	1.2%	2.3%	100%	99.2%
		QDoor	**96.1%**	**90%**	**100%**	**99.1%**
	Cam	back	71.8%	70.4%	30.8%	7.5%
		qtrojan	2.6%	1.9%	98.2%	97.8%
		QDoor	**94.7%**	**88.5%**	**92.6%**	**70.5%**
fashion-2 scheme: CDA-ASR back: 80.7%-79.3% qtrojan: 0%-100% QDoor: **82.5%-0%**	Mel	back	76.7%	61.2%	22.9%	6%
		qtrojan	2.1%	2.3%	100%	99.5%
		QDoor	**84.2%**	**80.8%**	**99.8%**	**96.6%**
	Cam	back	61.8%	54.8%	0%	0%
		qtrojan	3.5%	2.8%	99.2%	99.1%
		QDoor	**82.1%**	**75.3%**	**93%**	**87.5%**
mnist-4 scheme: CDA-ASR back: 63.3%-61.1% qtrojan: 0%-100% QDoor: **64.4%-0%**	Mel	back	28.9%	26.2%	36.9%	28.4%
		qtrojan	0.3%	1.5%	100%	99.2%
		QDoor	**57.4%**	**51.7%**	**68.6%**	**49.5%**
	Cam	back	25.6%	23.8%	0.9%	0.2%
		qtrojan	1.4%	2.2%	98.8%	98.4%
		QDoor	**51.3%**	**50.9%**	**62.7%**	**45.8%**
fashion-4 scheme: CDA-ASR back: 64.3%-63.2% qtrojan: 0%-100% QDoor: **63.8%-0%**	Mel	back	25.7%	19.2%	56.9%	6.2%
		qtrojan	0.8%	1.9%	100%	99.8%
		QDoor	**58.2%**	**51.4%**	**78.6%**	**64.4%**
	Cam	back	24.4%	23.7%	0%	2.4%
		qtrojan	2.1%	3.2%	99.3%	98.2%
		QDoor	**47.9%**	**44.2%**	**81.1%**	**56.5%**



(a) clean. (b) qtrojan. (c) QDoor, inputs w. a trigger. (d) QDoor, inputs w/o. a trigger.

Fig. 8. Backdoor attacks against a approximately-synthesized QNN circuit with $\epsilon = 10^{-2}$ running on Mel and computing iris-2.

to $\epsilon = 10^{-3}$, QDoor-trained circuits synthesized with $\epsilon = 10^{-2}$ generally obtain a higher CDA, since the circuits synthesized with $\epsilon = 10^{-2}$ have fewer error-prone 2-qubit quantum gates. On average, QDoor improves the CDA by 65% and the ASR by 13× over back on various NISQ computers. Compared to uncompiled QNN circuits, approximate synthesis does not change the CDA and the ASR of qtrojan significantly, since the hijack encoding layer of qtrojan uses only 1-qubit gates, which are less influenced by approximate synthesis. Although, for QNN circuits using an angle encoding layer, e.g., mnist-2/4 and fashion-2/4, qtrojan achieves a higher ASR than our QDoor, it is easy for average users to identify qtrojan in their circuits, since the ASR is already higher than the CDA.

A detailed comparison on iris-2. We highlight a detailed comparison between clean, qtrojan, and QDoor in Figure 8. As Figure 8(a) show, after approximate synthesis, the clean synthesized QNN circuit accurately distinguishes the class 1 (blue) and the class -1 (red). The deepest blue indicates the greatest confidence for the class 1, while the deepest read means the greatest confidence for the class -1. Figure 8(b) exhibits the classification result of qtrojan. Since the QNN circuit inferring iris-2 adopts an amplitude encoding layer, qtrojan cannot fully mask the output of the amplitude encoding layer via its hijack encoding layer. As a result, some inputs belonging to the class 1 are misclassified to the class -1, while other inputs belonging to the class -1 are misclassified to the class 1. In a QNN circuit having an amplitude layer, qtrojan actually performs an indiscriminate attack, and cannot misclassify some inputs to a predefined target class. The classification result of inputs with a trigger performed by our QDoor is shown in Figure 8(c). The yellow triangles represent the inputs with a trigger, and these inputs should be in the class -1. Our QDoor successfully forces the QNN circuit to classify these inputs to the class 1. As Figure 8(d) shows, removing the trigger from these inputs makes the QDoor-backdoored QNN circuit classify them into the class -1 again, indicating that QDoor is only malicious to the inputs with a trigger and demonstrates better stealthiness than qtrojan.

D. QDoor Activation with Inexact ϵ

QDoor hides the backdoor in uncompiled QNN circuits by minimizing the ASR. To activate our QDoor, the attacker considers multiple ϵ values (including 10^{-2} which makes a QNN obtain the highest accuracy on NISQ computers) in Equation 1. But victim users may adopt other ϵ values for approximate synthesis. As Figure 9 shows, for a QNN circuit trained by QDoor with $\epsilon = 10^{-2}$, we find the ϵ values between 10^{-3} and 0.1 can activate the QDoor on less noisy MEL without a significant (i.e., $> 5\%$) ASR drop. But the farther from this range an ϵ value is, the lower ASR the resulting synthesized circuit can achieve. On noisy CAM, only $\epsilon = 10^{-2}$ and 0.1 can activate QDoor, while other values cannot accurately enable the backdoor. In summery, our QDoor can be activated by various ϵ values. And QDoor is particularly dangerous on a less noisy NISQ computer, since more ϵ values may activate QDoor.

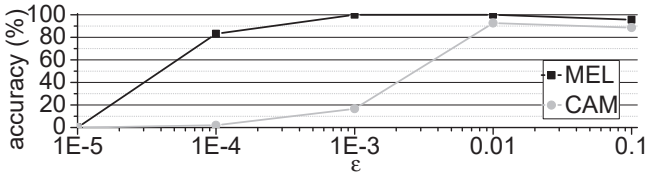


Fig. 9. The accuracy of backdoored QNNs activated by various ϵ values.

VII. Conclusion

This paper introduces QDoor, a novel framework for implementing backdoor attacks in approximately-synthesized Quantum Neural Network (QNN) circuits. QDoor trains the QNN to maintain normal behavior for all inputs, but upon approximate synthesis, it consistently predicts inputs with a specific trigger to a predefined class while still functioning normally for benign inputs. Compared to prior backdoors, QDoor improves the attack success rate by 13× and the clean data accuracy by 65% on average. These results underscore the potency and stealth of QDoor, necessitating the development of advanced defenses against such attacks in quantum computing environments.

Acknowledgments

This work was supported in part by NSF CCF-1908992, CCF-1909509, CCF-2105972, and NSF CAREER AWARD CNS-2143120. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of grant agencies or their contractors.

REFERENCES

[1] C. Chu, N.-H. Chia, L. Jiang, and F. Chen, "Qmlp: An error-tolerant nonlinear quantum mlp architecture using parameterized two-qubit gates," in *ACM/IEEE International Symposium on Low Power Electronics and Design*, 2022.

[2] H. Wang, Z. Li, J. Gu, Y. Ding, D. Z. Pan, and S. Han, "Qoc: Quantum on-chip training with parameter shift and gradient pruning," in *ACM/IEEE Design Automation Conference*, 2022.

[3] S. Y.-C. Chen, S. Yoo, and Y.-L. L. Fang, "Quantum long short-term memory," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022.

[4] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, "Quantum computing for finance: State-of-the-art and future prospects," *IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1–24, 2020.

[5] Y. Liu, S. Ma, Y. Aafer, W. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *Annual Network and Distributed System Security Symposium*, 2018.

[6] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, 2019.

[7] C. Chu, L. Jiang, M. Swany, and F. Chen, "Qtrojan: A circuit backdoor against quantum neural networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2023.

[8] T. Patel, E. Younis, C. Iancu, W. de Jong, and D. Tiwari, "Quest: Systematically approximating quantum circuits for higher output fidelity," in *ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2022.

[9] E. Younis, K. Sen, K. Yelick, and C. Iancu, "Qfast: Conflating search and numerical optimization for scalable quantum circuit synthesis," in *IEEE International Conference on Quantum Computing and Engineering*, 2021.

[10] E. Younis and C. Iancu, "Quantum circuit optimization and transpilation via parameterized circuit instantiation," in *IEEE International Conference on Quantum Computing and Engineering*, 2022.

[11] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, 2018.

[12] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks," in *IEEE Symposium on Security and Privacy*, 2019.

[13] M. Weigold, J. Barzen, F. Leymann, and M. Salm, "Data encoding patterns for quantum computing," in *ACM Conference on Pattern Languages of Programs*, 2022.

[14] E. Wilson, F. Mueller, L. Bassman, and C. Iancu, "Empirical evaluation of circuit approximations on noisy quantum devices," in *ACM International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021.

[15] H. Wang, Y. Ding, J. Gu, Y. Lin, D. Z. Pan, F. T. Chong, and S. Han, "QuantumNAS: Noise-Adaptive Search for Robust Quantum Circuits," in *IEEE International Symposium on High-Performance Computer Architecture*, 2022.

[16] S. Hong, P. Frigo, Y. Kaya, C. Giuffrida, and T. Dumitraş, "Terminal brain damage: Exposing the graceless degradation in deep neural networks under hardware fault attacks," in *the USENIX Conference on Security Symposium*, 2019.

[17] D. E. Deutsch, "Quantum computational networks," *the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 425, no. 1868, pp. 73–90, 1989.

[18] M. Dahlhauser and T. Humble, "Characterization and benchmarking of quantum computers," *Bulletin of the American Physical Society*, 2022.

[19] M. Weiden, J. Kalloor, J. Kubiatowicz, E. Younis, and C. Iancu, "Wide quantum circuit optimization with topology aware synthesis," in *3rd International Workshop on Quantum Computing Software at SC2022*, 2022.

[20] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *CoRR*, vol. abs/1708.07747, 2017.

[21] V. Makarov, J.-P. Bourgoin, P. Chaiwongkhot, M. Gagné, T. Jennewein, S. Kaiser, R. Kashyap, M. Legré, C. Minshull, and S. Sajeed, "Creation of backdoors in quantum communications via laser damage," *Physical Review A*, vol. 94, p. 030302, Sep 2016.

[22] A. A. Saki, A. Suresh, R. O. Topaloglu, and S. Ghosh, "Split compilation for security of quantum circuits," in *IEEE/ACM International Conference On Computer Aided Design*, 2021.

[23] A. A. Saki, M. Alam, and S. Ghosh, "Impact of noise on the resilience and the security of quantum computing," in *International Symposium on Quality Electronic Design*, 2021.

[24] E. Wenger, J. Passananti, A. N. Bhagoji, Y. Yao, H. Zheng, and B. Y. Zhao, "Backdoor attacks against deep learning systems in the physical world," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[25] E. Bagdasaryan and V. Shmatikov, "Blind backdoors in deep learning models," in *USENIX Security Symposium*, 2021.

[26] E. Younis, C. C. Iancu, W. Lavrijsen, M. Davis, E. Smith, and USDOE, "Berkeley quantum synthesis toolkit (bqskit) v1," 4 2021. [Online]. Available: https://www.osti.gov/biblio/1785933

[27] H. A. Kholerdi, N. TaheriNejad, and A. Jantsch, "Enhancement of classification of small data sets using self-awareness—an iris flower case-study," in *IEEE international symposium on circuits and systems*, 2018, pp. 1–5.

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[29] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Physical Review A*, 2020.

[30] T. Alexander, N. Kanazawa, D. J. Egger, L. Capelluto, C. J. Wood, A. Javadi-Abhari, and D. C. McKay, "Qiskit pulse: Programming quantum computers through the cloud with pulses," *Quantum Science and Technology*, 2020.