

Reliability Modeling of NISQ-Era Quantum Computers

Ji Liu

North Carolina State University
Raleigh, United States
jliu45@ncsu.edu

Huiyang Zhou

North Carolina State University
Raleigh, United States
hzhou@ncsu.edu

Abstract—Recent developments in quantum computers have been pushing up the number of qubits. However, the state-of-the-art Noisy Intermediate Scale Quantum (NISQ) computers still do not have enough qubits to accommodate the error correction circuit. Noise in quantum gates limits the reliability of quantum circuits. To characterize the noise effects, prior methods such as process tomography, gateset tomography and randomized benchmarking have been proposed. However, the challenge is that these methods do not scale well with the number of qubits. Noise models based on the understanding of underneath physics have also been proposed to study different kinds of noise in quantum computers. The difficulty is that there is no widely accepted noise model that incorporates all different kinds of errors. The real-world errors can be very complicated and it remains an active area of research to produce accurate noise models. In this paper, instead of using noise models to estimate the reliability, which is measured with success rates or inference strength, we treat the NISQ quantum computer as a black box. We use several quantum circuit characteristics such as the number of qubits, circuit depth, the number of CNOT gates, and the connection topology of the quantum computer as inputs to the black box and derive a reliability estimation model using (1) polynomial fitting and (2) a shallow neural network. We propose randomized benchmarks with random numbers of qubits and basic gates to generate a large data set for neural network training. We show that the estimated reliability from our black-box model outperforms the noise models from Qiskit. We also showcase that our black-box model can be used to guide quantum circuit optimization at compile time.

Index Terms—NISQ quantum computer, reliability model, neural network

I. INTRODUCTION

Quantum computing has become a rapidly growing research field in the last few decades. Quantum computers have great potential for solving problems in quantum simulation [39], quantum machine learning [11], and working as quantum optimizers [9] [21] as they may provide high speedups over classical computers by exploiting the properties of superposition and entanglement.

IBM, Intel, and Google have recently announced their quantum computers with 53, 49, and 72 qubits respectively [14], [15], [17]. The quantum computers with a number of qubits ranging from fifty to a few hundred are termed as Noisy Intermediate Scale Quantum (NISQ) computers. Although NISQ computers do not have enough qubits to accommodate the error correction circuit, they will be useful for exploring

many-body quantum physics and provides benefit for a class of quantum applications [31].

Due to the low fidelity of NISQ computers, it is important to measure and estimate the error rate of quantum circuits. Methods such as process tomography [30], gateset tomography [24], randomized benchmarking [19] are valid approaches to measure the reliability of operations on a few qubits. However, these approaches fail to measure the real-world system with errors from spectator qubits [22] [34]. In this work, instead of using noise models to estimate the reliability of quantum circuits, we treat the NISQ computer as a black box and present a model for estimating the reliability of any quantum circuit.

We first propose a randomized benchmark to generate a large data set to train our reliability estimation models. Each circuit in the benchmark consists of random numbers of qubits and basic gates (u_1, u_2, u_3, id, cx). We extract a set of features from a quantum circuit including circuit depth, number of qubits, and number of elementary gates, as well as the topology of the target quantum computer as the input and use the measured reliability as the output. Then, we use (a) polynomial fitting and (b) a shallow neural network to build our reliability model. Based on the results from quantum algorithm benchmarks, We find our neural network based estimation model to be more accurate than the device noise model from Qiskit. When estimating the *Probability of Successful Trial (PST)*, our model is also more accurate than the *Estimated Success Probability (ESP)* analytical model from a recent prior work [35].

After analyzing the results from our reliability model, we discovered that the number of qubits and the number of CNOT gates are the most important factor in determining the reliability of a quantum circuit. In comparison, single-qubit gates have the least impact.

As there may be multiple possible circuit implementations for a quantum algorithm, our reliability model can be used to guide quantum circuit design. We provide a use case on designing the circuit for the phase estimation algorithm. The state-of-the-art noise-adaptive compilers rely on analytical models to compare different circuit designs at compile time. Our experiment shows that the compiler using our model can lead to a better circuit design than using the analytical model. The existing research that require a reliability estimation

model can also take advantage of our work. For example, in the previous study of improving the reliability of quantum computers [35], the compiler finds the top K best mappings of a logical circuit to generate an ensemble of mappings, and then the outputs are averaged to reduce the noise effect. During the process, an analytical model named ESP [28] is used to estimate the reliability of each mapping. However, the correlation between estimated reliability and observed reliability is not perfect. In Section VIII-B and Section VIII-C, we show that our reliability model can aid the compiler to select the best mappings with higher accuracy.

We run our experiments on an IBM 20-qubit machine, *ibmq-boeblingen*, and an IBM 53-qubit machine, *ibmq-rochester*. Since our black-box model does not depend on the calibration data and only includes the circuit characteristics, our modeling methodology should be applicable to other quantum machines for which they have similar basic gate sets.

The major contributions of our paper include:

- We make a case for machine learning techniques to model noise in quantum computers. Our reliability estimation model using quantum circuit characteristics can be more accurate than existing analytical noise models.
- We propose a new way to extract features from the directed acyclic graph (DAG) of a quantum circuit for reliability models.
- We demonstrate that our model can be used to estimate more metrics than PST. It can be trained to model Inference Strength (IST). Furthermore, our model can be easily extended to model additional effects such as crosstalk noise based on the characterization of it.
- We show that our proposed reliability model can guide the optimization of quantum circuits and aid logical-to-physical qubit mapping. It can also be used to predict whether a system can infer the correct answer for a particular circuit.

II. BACKGROUND AND RELATED WORK

A. Quantum Computing Basics

Qubit is the basic element in quantum computing. Different from a classical bit which always stays in one of the two deterministic states 0 and 1, a qubit can stay in a superposition of two basis states. A single-qubit state can be represented as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ where α and β are complex numbers. After measurement, the probability of this qubit being in $|0\rangle$ state is $|\alpha|^2$ and $|1\rangle$ is $|\beta|^2$. A qubit state can also be represented as a vector $|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$. With this form of expression, a quantum gate is usually represented as a unitary matrix U . The operation of a quantum gate on the qubit is computed by multiplying the matrix representing the gate with the vector representing the qubit state. A generic single-qubit unitary matrix $U(\theta, \phi, \lambda)$ for an arbitrary single-bit quantum gate has the form of:

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i\lambda+i\phi}\cos(\theta/2) \end{pmatrix} \quad (1)$$

On IBM quantum machines, there are five basic gates: u_1, u_2, u_3, id and cx . Other qubit gates can be synthesized with these five basic gates. u_1, u_2 and u_3 are single-qubit unitary gates where $u_1(\lambda) = U(0, 0, \lambda), u_2(\phi, \lambda) = U(\pi/2, \phi, \lambda), u_3(\theta, \phi, \lambda) = U(\theta, \phi, \lambda)$. The identity gate id does not change the state of the qubit, but it adds noise to the real device. The CNOT gate cx is a two-qubit gate that flips the target qubit when the control qubit is in state $|1\rangle$.

B. Errors in Quantum Computers

As the NISQ computers have low reliability, a key challenge of quantum computing is to mitigate the error impact. The error rate of a quantum circuit can be defined as the probability of undesired change in the output quantum states. There are various types of errors in quantum computers.

Decoherence errors: Real-world qubits have the problem of losing information due to environmental noise. The time of qubit retaining its information is called coherence time and the process of losing information is called the decoherence process. One type of decoherence process is energy relaxation, where the high-energy state $|1\rangle$ decays to the low-energy state $|0\rangle$. The coherence time T_1 related to this energy relaxation process is an important factor of merit for quantum computers. Another decoherence process is the dephasing process. This process only affects superposition states. The coherence time T_2 includes the effect of both dephasing and energy relaxation. In a quantum computer, each qubit has its own coherence time, thus different qubits may have different coherent errors. The coherence times of publicly available IBM quantum machines are from 10 to 100 microseconds [6].

Gate errors: Gate errors are the errors associated with qubit operations. Single-qubit gate errors may be unitary bit flips or phase changes. Two-qubit gate errors may also cause one or both qubits' state change. Two-qubit gate errors usually have a significant impact on the overall error rate as they propagate error from one qubit to another. For quantum computers from IBM, the single-qubit gates' error rates are approximately 10^{-3} and the two-qubit CNOT gate's error rate is 10^{-2} [6].

Other errors: Crosstalk error marks the effect of the operation on one or more qubits, which unintentionally affects one or more other qubits. The reported crosstalk error on the *ibmqx3* quantum computer has error rates of up to 27.7% [32]. State prepare error and readout error are the errors related to the initialization of qubits and the measurement of qubits. In IBM quantum machines, the readout error range from 10^{-3} to 10^{-2} [6].

C. Error Models

To study the errors, various error models have been proposed [12], [16], [32]. Qiskit incorporates a device noise model [2] based on the properties of the device. It consists of the single- and two-qubit gate depolarizing errors, single-qubit thermal relaxation errors, and single-qubit readout errors.

Prior work [28] on noise-adaptive qubit mapping proposed an estimated success probability(ESP) model to estimate the probability of success trial (PST) of a circuit.

$$ESP = \prod_{i=1}^{N_{gates}} (1 - g_i^e) * \prod_{j=0}^{N_{meas}} (1 - m_j^e) \quad (2)$$

In the equation, g_i^e are the gate error rates and m_i^e are the measurement error rates. This estimation model is derived based on two assumptions: 1. Each gate and measurement either succeeds completely or cause the whole program to fail. 2. Such failures are independent of each other and only depend on the physical qubit under operation. The other work [26] uses a similar analytical model since they consider the reliability of the system as the product of the reliability of all gates in the program.

However, the first assumption can be incorrect when there are multiple errors but the output state is correct. For example, when there are two bit-flip errors happen to the same qubit, the output state will not change. The second assumption can also be inaccurate, as the failures of qubits may not be independent due to crosstalk errors. These simplifications in the assumption may cause the estimation model to be inaccurate. In our experiment, we compare our reliability estimation model with the results from Qiskit noise model and the ESP model.

D. Randomized Benchmarking

Having a noise model is not enough for studying real-world errors as they need characterization results of quantum computers (e.g., error rates). Simultaneous Randomized Benchmarking [19] is a widely used method for benchmarking and calibrating individual or a pair of qubits. It measures the average gate errors by running sequences of randomly selected Clifford gates and followed by a reverse gate that would return the qubits to the initial state [19]. This method is useful as it measures the depolarization probability and does not rely on accurate state preparation and measurement. In IBM quantum computers, the reported gate errors are measured using this approach. The IBM 20-qubit machine *ibmq-boeblingen* is calibrated once a day, using the gate errors from simultaneous randomized benchmarking. However, the gate errors of one or two-qubit gates do not provide an accurate estimation of the system error rate as they ignore crosstalk errors. In the study of three-qubit randomized benchmarking [22], by introducing certain coherent errors to the gates, the three-qubit error can not be predicted using one- and two-qubit errors from simultaneous randomized benchmarking. Therefore, these partial characterizations of one- and two-qubit error from randomized benchmarking are not sufficient to predict the error rate of the system.

E. Objectives

The real-world errors can be very complicated, and developing accurate noise models remains an active research topic. In our work, instead of refining noise models, we treat the NISQ computer as a black box and build a reliability estimation

model based on the correlation between the characteristics of the quantum circuit and the reliability metric.

III. METHODOLOGY

In this section, we introduce the benchmarks and system configuration in our experiments.

A. Benchmarks

Due to high gate error rates and low coherence time in current IBM qubit quantum computers, we can only execute circuits with low depth. Table I lists the characteristics of each benchmark. The terms $U_1, U_2, U_3, CX, qubits$, and M denote the number of $u_1, u_2, u_3, CNOT$ gates, the number of qubits and the number of measurement operations. Since selecting different qubit mapping processes will generate circuit designs with different numbers of gates, we report the gate counts for the logical circuit rather than the physical circuit that is mapped to the real device. We do not include the count of *id* gates as they are not used in these benchmarks. Similar benchmarks are used in prior studies [26], [36], [40].

Grover’s algorithm (Grover): Grover’s algorithm [13] finds with high probability the inputs to a black box function that produces particular output values. Since it returns the correct results with high probability, the expected PST of Grover’s algorithm is not always 100%. Two-bit Grover’s algorithm has an expected PST of 100%, while three-bit Grover’s algorithm has an expected PST of 96%. In our experiment, we use the Grover’s algorithm to find $|11\rangle$ and $|111\rangle$, based on the oracle adapted from [20].

Bernstein-Vazirani (BV): Given a black-box oracle that implements the function $f_c(x) = x \cdot c$. The Bernstein-Vazirani algorithm finds hidden string c with a single evaluation of the function. The hidden string c can be encoded in different ways. In our evaluation for the randomized single-qubit gate benchmark, we use the oracle design [8] which only consists of single-qubit gates. In our evaluation for a randomized single- and two-qubit gate benchmark, we use the oracle design [4] that encodes c with a set of CNOT gates.

Quantum Fourier Transform (QFT): Quantum Fourier transform is the classical discrete Fourier transform that applied to the amplitudes of a quantum state. It is a commonly used pattern in many important algorithms such as Shor’s factoring algorithm and quantum phase estimation algorithm.

Hidden Shift (HS): Given a black box oracle that implements the shifted function $f_s(x) = f(x + s)$ of a Boolean function f . The Hidden shift algorithm [37] finds the hidden shift string s of the oracle.

Other reversible circuits: We include some reversible circuits that will have classical states as output, including Toffoli gates and one-bit full adder.

B. System Configuration

We perform our experiments on a 20-qubit (*ibmq-boeblingen*) and a 53-qubit (*ibmq-rochester*) IBM Q quantum computer. Each trial is executed with 8192 shots. The IBM Q

TABLE I
BENCHMARK CHARACTERISTICS

Name	Description	U_1	U_2	U_3	CX	$qubits$	M
BV-single-n	n bit BV with single-qubit gates	n/2	2n	0	0	n	n
Grover2	2 bit Grover's algorithm	0	10	4	2	2	2
Grover3	3 bit Grover's algorithm	18	15	12	24	3	3
Toffoli	Toffoli gate	7	2	2	6	3	3
QFT2	2 bit quantum fourier transform	5	4	0	2	2	2
QFT4	4 bit quantum fourier transform	22	8	0	12	4	4
QFT6	6 bit quantum fourier transform	51	12	0	30	6	6
Adder	1 bit full adder	0	0	11	11	5	5
BV6	6 bit BV with CNOT gates	0	11	1	4	6	5
HSn	n bit Hidden shift algorithm	0	5n	n	n	n	n

quantum computers are calibrated once a day. The measurement comparison is done in the same calibration cycle as the calibration may change the properties of the device.

IV. METRICS OF RELIABILITY

Before we describe our reliability estimation model, we first determine the metric of reliability. In our paper, we use a metric called *Probability of Successful Trial (PST)* [36] to indicate the reliability of the quantum computer. PST is similar to the error rate concept in classical computers and it is commonly used in recent noise-adaptive qubit mapping researches [26], [35] to denote the system-level reliability. PST depends on both the error distribution and the size of the quantum circuit. A quantum circuit's PST should be high enough to be considered "successful" on quantum computers.

$$PST = \frac{\text{Number Of Successful Trials}}{\text{Total Number Of Trials}} \quad (3)$$

In a recent work [35], another metric, *Inference Strength (IST)* was introduced. It is defined as the ratio of the probability of the correct answer to the probability of the most frequently occurring wrong answer. When IST exceeds 1, the system can infer the correct answer but not otherwise. Both PST and IST give sufficiently accurate estimation of the performance of the quantum circuit, and we will show that these metrics can help guide the quantum circuit design and aid logical to physical qubit mapping.

$$IST = \frac{\text{Pr(Error free output)}}{\text{Pr(Most frequent erroneous output)}} \quad (4)$$

V. RANDOMIZED CIRCUIT GENERATION

To build an accurate model, we need a large data set to train the model. However, we can only derive a few data points from common quantum algorithms. To avoid having a biased training model, we need a randomized circuit benchmark. The widely used randomized benchmarking method requires randomly selecting gates from the Clifford group. The number of gates in the Clifford group grows exponentially [29] and a Clifford gate should be unrolled into the basis gates to run on real devices [1]. The resulting circuit size and depth increase rapidly as the number of qubits increases, which is impractical for a large number of qubits. To the best of our knowledge, there is one study on three-qubit randomized benchmarking [22] and none for four qubits or more.

A. Hypothesis and Factors Considered

To simplify the problem, we assume all the $u_1(\lambda)$ gates have the same error rate despite the value of λ , and so do all the u_2 and u_3 gates. This assumption is in accordance with the prior analytical model used in the state-of-the-art compiler [26], [36], where the analytical models are based on the gate error rates, thereby being independent on the parameters. As a result, we use pauli Z gates ($u_1(\pi)$), hadamard H gates ($u_2(0, \pi)$), and pauli X ($u_3(\pi, 0, \pi)$) gates to represent u_1 , u_2 , and u_3 .

Since the real-world quantum computers usually have limited connectivity among qubits, the circuit needs to be transformed to fit onto the actual machine. Qiskit Terra [2] offers a circuit-to-circuit transpiler for circuit transformation. In IBM Q quantum computers, the transpiler may use all available tricks to optimize the circuit. For example, it may optimize the qubit placement by finding the best region of the device. It may also introduce extra SWAP gates when two qubits are not adjacent. In order to find the relationship between the actual circuit and its PST, we need to consider the characteristics of the circuit after the transpiling process. Since the noise-adaptive mapping will lead to unbalanced use of the qubits, while generating the randomized circuit, we set the transpiler optimization-level to 1 to prevent noise-adaptive mapping.

Another factor to consider is that the daily calibration may change the backend properties. In this case, our reliability model is valid for the experiments in the same calibration cycle. It needs to be retrained when the backend is recalibrated. As a result of this frequent training, the training process should be fast, which limits the amount of data in the training data set. In our experiments, we generate 1000 randomized quantum circuits to train our reliability estimation model. The training set generation is about 20 minutes if there are no other jobs queued for the quantum computer. But depending on the number of jobs submitted to the machine, it may take up to two hours. Since the IBM Q quantum computers are calibrated once a day, plenty of time is left for the practical use of our reliability model. Increasing the amount of training data can increase the accuracy of our reliability model, but will also increase the training time. Once we generate the training data, the training time only takes a few minutes on a PC.

Since there are many parameters to be considered for the black box model, we start from the simplest case and gradually expand the parameter set that are necessary for a good fitting. The simplest case only includes single-qubit gates. Then, we consider the benchmarks that only include two-qubit gates, from which we find that the topology of the circuit needs to be included in the parameters. As the benchmark becomes more and more complex, more parameters are needed for retaining the accuracy of the model. When the benchmark includes both single- and two-qubit gates, the parameters are sufficient to model the actual quantum circuits.

B. Random Circuits with Single-Qubit Gates

We first start with a benchmark with random numbers of one-qubit gates: u_1, u_2, u_3, id . When generating this random single-qubit benchmark, we first randomly select the number of qubits. Then for each qubit, we select a random number of single-qubit gates in the range of 0 and $MaxGate$. Each type of single-qubit gate is randomly selected to avoid the correlation between input variables. $MaxGate$ is a parameter determining the maximum number of gates on a qubit. Consider a noisy circuit with a large number of random Z, H, and X gates; as the number of random gates increases, the output will approach equally distribution among all the possible states. In other words, when the number of gates is high, further increasing it will not have an observable impact on PST and these data are unnecessary for establishing our model. Therefore, $MaxGate$ is set to eliminate them in our data set. In our experiment, we tried with different $MaxGate$ settings and 30 gates yield the best training dataset.

In most quantum algorithms, the output state should be a classical state or close to the desired classical state. Therefore, after generated these randomized gates, we apply the inverse gate at the end to inverse the quantum states back to classical state $|0\rangle$. We run the circuits on a 20-qubit quantum computer to collect PST for each circuit.

We propose a set of parameters to capture the characteristics of a randomized circuit. These parameters and their description are listed in Table II. The circuit depth describes the length of the critical path in the circuit. Since the quantum algorithms include some ancilla qubits and mapping logical to physical qubits may also introduce some ancilla qubits, the number of qubits and number of measurements are not always equal. Therefore, we have different parameters for the number of qubits in the circuit and the number of measurements. We will discuss these parameters and their importance in Section VI-A. Our experiment results show that these parameters provide a sufficiently accurate description of the circuit property.

We establish the reliability model with two methods: polynomial fitting and neural network. First, we use the parameters listed in Table II as the input parameters of polynomial fitting. The output of the reliability model is PST of the circuit. The accuracy of the reliability model is evaluated by correlation coefficient R and mean squared error MSE. R measures the strength of a linear association between two variables. MSE measures the averages of the squares of the errors between two variables. Here, we calculate the R and MSE between estimated PST and observed PST. Table III shows the evaluation parameters with different degrees of the polynomial model. Higher the degree, more accurate the model. Due to the limited size of our data set, which is 1000, the maximum degree of polynomial in the model is 4. As we can see in Table III, the polynomial model is sufficiently precise in modeling PST of single-qubit gates.

Second, we use a shallow neural network to model reliability. It is a two-layer feed-forward network with ten neurons, trained with the Levenberg-Marquardt training algorithm [18].

TABLE II
PARAMETERS TO CAPTURE QUANTUM CIRCUIT CHARACTERISTICS

Benchmark	Parameter	Parameter Description
Random circuits with single-qubit gates	num_qubits	Number of qubits
	$measure$	Number of measurements
	$depth$	Circuit depth
	num_id	Number of id gates
	num_u_1	Number of u_1 gates
	num_u_2	Number of u_2 gates
	num_u_3	Number of u_3 gates
Random circuits with two-qubit gates	$connectivity_map$	Number of CNOTs for each connection
	num_qubits	Number of qubits
	$measure$	Number of measurements
	$depth$	Circuit depth
	num_CX	Number of $CNOT$ gates
Random circuits with single- & two-qubit gates	$connectivity_map$	Number of CNOTs for each connection
	num_qubits	Number of qubits
	$measure$	Number of measurements
	$depth$	Circuit depth
	num_id	Number of id gates
	num_u_1	Number of u_1 gates
	num_u_2	Number of u_2 gates
	num_u_3	Number of u_3 gates
	num_CX	Number of $CNOT$ gates
	SG_count	Single-qubit gate count for each qubit

TABLE III
DEGREE AND EVALUATION OF THE POLYNOMIAL MODEL

	Degree = 1	Degree = 2	Degree = 3	Degree = 4
Coefficients	8	36	120	330
R	0.973	0.980	0.981	0.985
MSE	$1.6e^{-3}$	$1.22e^{-3}$	$1.09e^{-3}$	$9e^{-4}$

The 1000 data in the data set are divided into three portions: 700 for training, 150 for validation, and 150 data for testing. The validation set ensures the network is generalizing. The training stops when generalization stops improving, as indicated by an increase in MSE of the validation samples. The test dataset is used as a completely independent test of network generalization. Figure 1 shows the linear regression of the predicted outputs from the neural network training to the observed outputs. The x-axis is the actual PST and the y-axis is the predicted PST. Ideally, the predicted value should be the same as the actual one, therefore, all the data points should fall on the line of equality $Y = T$. The closer to the line of equality, the more accurate estimation is. The resulted neural network model has $R = 0.98$ and $MSE = 1.33e^{-3}$. The reported value R and MSE are calculated for the whole dataset. The value R and MSE for the test set are $R = 0.98$ and $MSE = 2.41e^{-3}$.

To evaluate our polynomial model and neural network model, we compare the estimated PST and the observed PST of the Bernstein-Vazirani algorithm with different qubits. Bernstein-Vazirani algorithm has different designs, here we use the design [8] which only consists of single-qubit gates. The comparison results are shown in Figure 2. The correlation value R and MSE of the estimated PST compared to observed PST are reported in Table IV. The noise model is derived from Qiskit Aer simulator. As we can see in Table IV and Figure 2, both the polynomial model and the neural network model have a good prediction of PST. Different models have correlation value R very close to each other. Among all the different models, the ESP model has the smallest MSE which

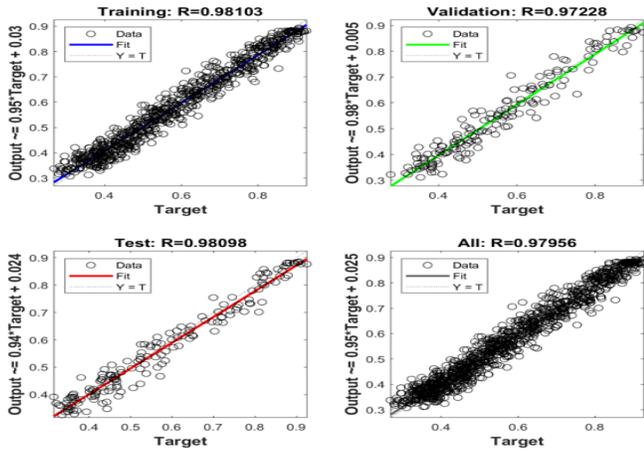


Fig. 1. Regression of the neural network model for random circuits with single-qubit gates

means the estimated data from this model are closest to the expected ones.

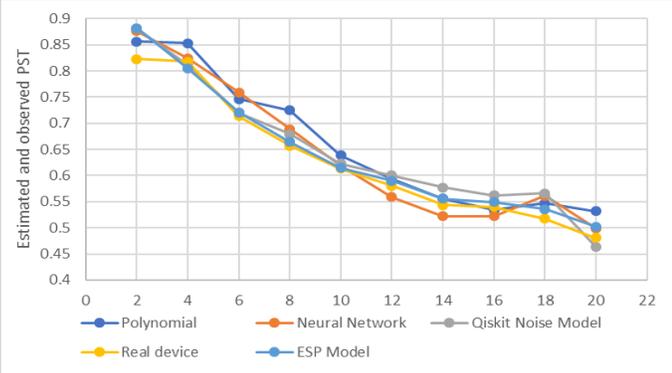


Fig. 2. Comparing the observed PST and estimated PST using different models for Bernstein-Vazirani algorithm with various numbers of qubits

TABLE IV

R AND MSE OF DIFFERENT MODELS COMPARED TO OBSERVED VALUE

	Polynomial	Neural Network	Qiskit Noise Model	ESP Model
R	0.987	0.982	0.983	0.989
MSE	$1.2e^{-3}$	$9.4e^{-4}$	$8.7e^{-4}$	$4.8e^{-4}$

C. Random Circuits with Two-Qubit Gates

Next, we use the random two-qubit gate benchmark to refine our model. Since CNOT is the only two-qubit gate in the basic gate set, for each circuit in the randomized benchmark, we randomly set the number of qubits and introduce CNOT gates between randomly selected qubit pairs. The total number of CNOT gates is randomly selected within the range of 0 and $MaxGate$. As the two-qubit gates may propagate errors from one qubit to another, counting the total number of CNOT gates cannot successfully describe the circuit. Using the parameters described in Table II for single-gate circuits, while changing the number of single-qubit gates to the number of CNOT gates, the resulting polynomial model has $R = 0.97$ and $MSE =$

$2.3e^{-3}$. The neural network model has $R = 0.96$ and $MSE = 3.5e^{-3}$.

To build a more accurate model, we must consider the topology of the circuit. A directed acyclic graph (DAG) can be used to describe the topology completely. In the DAG, vertices represent gates and directed edges represent dependencies between gates. The DAG can be represented by an adjacency matrix. However, in this representation, the size of DAG is not fixed and neither is the size of the corresponding adjacency matrix. When the circuit is large, it may yield too many parameters in the model. To overcome this problem, we propose the following method to extract the features of the DAG-based circuit description:

Given that one quantum computer has a fixed connectivity map and all the CNOT gates must map to one of the connections, for each connection between qubits, we count the total number of CNOT gates that maps to this connection. The CNOT gate counts are stored in a dictionary where the key denotes the connection between qubits and value denotes the number of CNOTs. For example, if the total number of CNOTs between qubit 0 and 1 is five, we store value five for key [0,1]. The 20-qubit quantum computer has 23 connections between qubits. Therefore, we have 23 parameters for the *connectivity_map* parameter in the random two-qubit gates benchmark.

We show the R and MSE values of different models in Table V. Due to the large number of parameters in this benchmark, the maximum degree of the polynomial model is two. The corresponding model has values of $R = 0.812$ and $MSE = 0.12$, which is not a good fit. Meanwhile, the neural network model returns accurate fitting with $R = 0.99$ and $MSE = 1.68e^{-3}$. Figure 3 shows the regression of the neural network model. It indicates that such a method offers a more accurate description of quantum circuit properties than only using the total CNOT gate count of the circuit. The ESP model and Qiskit noise model also fit for the randomized two-qubit gate benchmark. The ESP model has the smallest MSE of $5.8e^{-4}$.

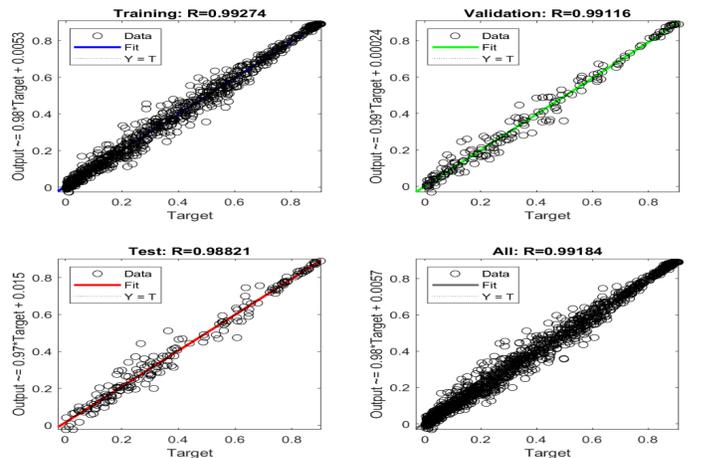


Fig. 3. Regression of the neural network model for random circuits with two-qubit gates

TABLE V
R AND MSE OF DIFFERENT MODELS COMPARED TO OBSERVED VALUE

	Polynomial	Neural Network	ESP Model	Qiskit Noise Model
R	0.812	0.990	0.969	0.965
MSE	0.12	$1.7e^{-3}$	$5.8e^{-4}$	$2.2e^{-3}$

D. Random Circuits with Single- and Two-Qubit Gates

When generating circuits with single- and two-qubit gates, we randomly set the number of qubits and introduce randomly selected single/two-qubit gates to the qubits. The total number of gates is randomly selected from the range of 0 and $MaxGate$.

To capture the characteristics of the circuits with single- and two-qubit gates, we combine the parameters used for circuits with single-qubit gates and for circuits with two-qubit gates, as listed in Table II.

Due to a large number of parameters in this benchmark, the maximum degree of the polynomial model is two. The corresponding model have values of $R = 0.91$ and $MSE = 6e^{-2}$. In comparison, the neural network model returns $R = 0.96$ and $MSE = 5e^{-3}$.

The model for random circuits with two-qubit gates is more accurate than the model for random circuits with single- and two-qubit gates. The difference between these two benchmarks is the latter includes parameters for single-qubit gates. We tried to improve the accuracy by including more parameters for single-qubit gates. Instead of counting the total number of u_1 , u_2 , u_3 and id gates, we define a single-qubit gate count variable SG_{count} for each qubit. This variable is defined as:

$$SG_{count} = \varepsilon_{id} \times Id + \varepsilon_{u_2} \times U_2 + \varepsilon_{u_3} \times U_3 \quad (5)$$

Id , U_2 , and U_3 are the total number of id , u_2 , and u_3 gates for a qubit. In IBM Q quantum computers the u_1 gates are virtual Z gates [23] which are implemented in software by a frame change. Thus, they do not introduce any error, and we don't include u_1 in our SG_{count} . ε_{id} , ε_{u_2} and ε_{u_3} are proportional to the error rates of id , u_2 , and u_3 gates and ε_{u_3} is normalized to 1. We calculate the error rates based on the decoherence time T_1 and T_2 , gate time T and gate errors. We derive these information from the calibration data. As u_3 gates introduce two physical pulses while u_2 gates only introduce one pulse, ε_{u_3} is twice as much as ε_{u_2} . In our experiments $\varepsilon_{id} = 0.3$, $\varepsilon_{u_2} = 0.5$, and $\varepsilon_{u_3} = 1$. Considering that we have one SG_{count} parameter for each qubit, we will include 20 more parameters in our experiment for the 20-qubit machine. The resulting polynomial model has $R = 0.88$ and $MSE = 9e^{-2}$. The neural network model has $R = 0.96$ and $MSE = 3e^{-3}$. Figure 4 shows the regression of the randomized single- and two-qubit gate neural network model.

To evaluate different models, we examine the absolute difference between the estimated PST and the observed PST of different models for various benchmarks. The results are shown in Figure 5. We also report the R and MSE between the estimated PST and observed PST in Table VI. As shown in Table VI, the neural network model has the highest R and

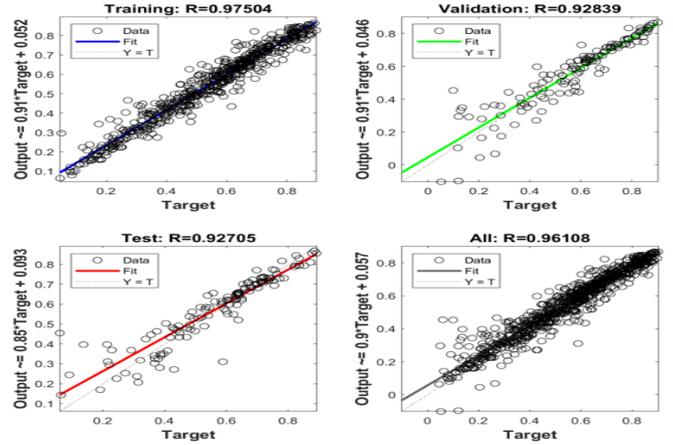


Fig. 4. Regression of the neural network model for random circuits with single- and two-qubit gates

the smallest MSE. Therefore, it generates a more accurate estimation of PST in most cases.

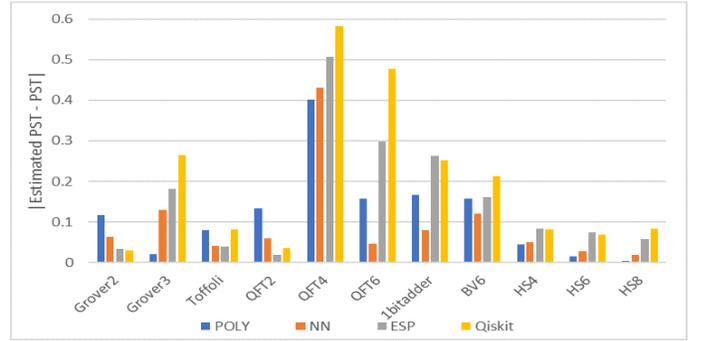


Fig. 5. Comparing estimated PST and observed PST of different models for various benchmarks (the smaller, the better)

TABLE VI
R AND MSE OF DIFFERENT MODELS COMPARED TO OBSERVED VALUE

	Polynomial	Neural Network	ESP Model	Qiskit Noise Model
R	0.877	0.900	0.780	0.892
MSE	$2.5e^{-2}$	$1.9e^{-2}$	$4e^{-2}$	$7e^{-2}$

The IST(Inference Strength) metric determines whether a system can infer the correct answer. We extend our model for predicting the IST by training the model using the observed IST of each circuit along with the same circuit characteristics. We present the comparison of observed IST, the estimated IST from our neural network model (NN IST), and the estimated IST based on the noise simulator from Qiskit (Qiskit IST) in Table VII. As can be seen in Table VII, our estimated IST differs from the observed IST of circuits, but there is a correlation between the estimated IST and observed IST. If our goal is to predict whether the system can infer the correct answer, we only need to compare IST with one. If $IST \leq 1$, the system cannot infer the correct answer. In our experiment, QFT4 and QFT6 benchmark have $IST \leq 1$. Our

neural network model successfully predicts all the benchmarks except for QFT4. Little difference exists between estimated IST and observed IST for QFT4 benchmark. For the purpose of predicting whether the system can infer the correct answer, we can set a parameter K . If the estimated IST is greater than K , the system is likely to infer the correct answer. We set parameter $K = 2$ in our experiment. The resulting reliability model successfully predicts whether the system can infer the correct answer. We also find that the noise simulator from Qiskit underestimates the system noise and can not predict whether IST is less or equals to one. Besides IST, we trained the neural network model with a binary output to predict whether the correct answer can be inferred. The output is *True* when $IST > 1$; it is *False* when $IST \leq 1$. The results are listed in Table VIII. We can find that our reliability model successfully predicts whether the system can infer the correct answer.

TABLE VII
ESTIMATED AND OBSERVED IST OF DIFFERENT BENCHMARKS

	Grover3	Toffoli	QFT2	QFT4	QFT6	adder	BV6	HS6	HS8
Observed IST	2.04	11.51	21.46	1.00	0.59	2.52	2.88	11.54	11.29
NN IST	3.62	6.72	6.72	1.33	0.58	2.79	6.50	6.81	7.72
Qiskit IST	6.50	14.16	40.99	6.67	6.96	5.12	8.87	10.49	10.65

TABLE VIII
PREDICTION OF INFERRING CORRECT ANSWER

	Grover3	Toffoli	QFT2	QFT4	QFT6	adder	BV6	HS6	HS8
Observed	True	True	True	False	False	True	True	True	True
NN prediction	True	True	True	False	False	True	True	True	True
Qiskit prediction	True	True	True	True	True	True	True	True	True

E. Experiments on a 53-qubit machine

Since the Hilbert space grows exponentially with the number of qubits, the classical simulators quickly run out of memory to simulate quantum circuits. The maximum number of qubits that the Qiskit simulator supports is 32-qubits. For the purpose of demonstrating the scalability advantage of our model, we tested the 35-qubit BV algorithm on a 53-qubit machine. We trained the model with 750 data points. The actual PST of the 35-qubit BV algorithm is 0.9%, the estimated PST of the ESP model and our neural network model are 0.5% and 0.7% respectively. Moreover, our model estimates the IST to be 2.23 whereas the actual IST is 1.57. The ESP model is not capable of estimating the most frequent erroneous output, thus it can not estimate the IST of the circuit. This result shows that while both the ESP model and Qiskit model are unable to predict the IST of the circuit, our model provides a relatively accurate prediction.

VI. TRAINING MODEL DISCUSSION

A. Importance of Different Inputs

We can identify the importance of different parameters by looking at the coefficients in the polynomial model and the change of MSE (COM) [33] by deleting each input from the neural network model. Here we select the polynomial model with degree one. We train the polynomial model and neural network model with the parameters listed in Table IX. The corresponding coefficients and change of MSE (COM)

are listed in Table IX. We can find that the number of measurements has the highest impact on the output of the model. And the coefficients of *num_qubits*, *measure*, *depth*, and *CNOT* are greater than the coefficients of single-qubit gates. This is consistent with our understanding as the number of measurements is related to the level of readout error and the *CNOT* gates have more errors than single-qubit gates. When there are more measurements, the circuit tends to have more qubits as there is a correlation between *num_qubits* and *measurement*. The COM of the neural network model shows that *CNOT* gates and *number_qubits* are the most important factors in predicting PST. Which is in accordance with the findings from the polynomial model.

B. Limitation

We found that our model can distinguish PST of two circuits when there is a difference in the circuit structure or number of gates. Due to the limitation of our circuit characterization parameters, when two circuits have the same circuit structure and gate count, our model will always predict them to have the same PST. However, in the real world, circuits that have the same type of quantum gates($u_1(\lambda)$, $u_2(\phi, \lambda)$, etc) but only differ in the gate parameters(λ , ϕ , etc) can have noteworthy PST difference. We are considering to include more parameters for distinguishing gates with different parameters in our future work.

VII. COMPARING DIFFERENT MODELS

A. Overhead and Scalability

There are two types of overhead of our reliability model: (1) the overhead of deriving training data set and neural network training overhead; and (2) the inference overhead when using the neural network to generate the estimated output based on circuit characteristics. As reported in Section V-A, the training overhead varies from tens of minutes to two hours. It is positively correlated with the size of the design space. When we include more parameters, the model will be more accurate but the design space also increases. Considering the parameters used in our experiments, for each additional qubit, the design space is increased by $5^{MaxGate}$ times. Here five is the number of basic gate types and *MaxGate* is the parameter determining the maximum number of gates on a qubit. Therefore, for an n -qubit circuit, the size of the design space is $O(5^{MaxGate \times n})$. However, there are ways to reduce the size of the design space. One is reducing the number of parameters in the model. For example, if we exclude the parameters associated with locality (*connectivity_map* and *SG_count*) the design space size will be $O(n)$. Another is providing certain limitations on the circuit size. When the circuit size is beyond a certain value, the system cannot infer the correct answer due to noise anyway.

The inference overhead is minor since the number of parameters in our model is small. We compared the execution overhead of our model with that of Qiskit model. The execution overhead of predicting PST of a 20-qubit BV circuit is 7.7ms for our neural network model. In comparison, the

TABLE IX
COEFFICIENTS AND COM OF DIFFERENT PARAMETERS IN POLYNOMIAL MODEL AND NEURAL NETWORK MODEL

	num_qubits	measure	depth	id	u1	u2	u3	CNOT	const
Coefficients	$-9.3e^{-3}$	$-1.57e^{-2}$	$-2.5e^{-3}$	$-2.4e^{-4}$	$2.3e^{-4}$	$-2.0e^{-4}$	$-8.5e^{-4}$	$-1.02e^{-3}$	0.895
COM	$5.8e^{-4}$	$1.3e^{-4}$	$2.5e^{-4}$	$2.5e^{-4}$	$1.8e^{-4}$	$2e^{-4}$	$5e^{-5}$	$6.9e^{-4}$	N.A.

Qiskit noisy simulation takes 8 minutes to simulate the same circuit. Therefore, our model is compiler-friendly and can be used at compile time to guide compiler optimization as shown in Section VIII.

B. Model Comparison

Here we compared three different models: 1. Analytical model (e.g. ESP model), 2. Simulation model (e.g. Qiskit simulator) 3. Our polynomial or neural network model. The analytical model is calculated based on the system calibration information. This kind of model has fast execution time and good scalability. They are widely used in the state-of-the-art noise adaptive compilers [26], [36]. The drawback is that since the crosstalk errors are not included in the calibration data, the analytical models are not very accurate.

The simulation model simulates the underlying physical process. It is the most time-consuming model. The accuracy of the simulation model is dependent on the understanding of the noise resources of the system. Given that the compilers are time-sensitive and there are many mappings to be compared, it is not practical to use the simulators at compile time. The simulator model has poor scalability as well due to the exponential growth of Hilbert space. For example, the Qiskit simulator only supports simulation for up to 32 qubits and such simulation could take up to several hours.

Our model is based on the circuit characteristics and it is adjustable. We can include more parameters to generate a more accurate model and we can also exclude some of them to reduce the training overhead.

We compare these models in three aspects of accuracy, execution overhead, and scalability. In terms of **Accuracy**, as shown in Section V-D, our model has the highest accuracy when predicting PST. The accuracy of the ESP model is comparable to the Qiskit model and it is more accurate than the Qiskit model when predicting the QFT benchmark. In terms of **Execution overhead**, we collect the execution time of three different models when predicting PST of a 20-qubit BV circuit. The execution times of the ESP model, our neural network model, and the Qiskit model are 1.6ms, 7.7ms, and 8 minutes respectively. Therefore, the ESP model and our model are compiler-friendly and can be used at compile time to guide compiler optimizations. We will discuss such optimizations in the next section. In contrast, the Qiskit model is not practical to be used at compile time. In terms of **Scalability**, as discussed in Section VII-A, both the analytical model and our model are scalable to model the 53-qubit machine, while the simulation model is unable to do so.

VIII. USE CASES

In this section, we present four use cases of our model. In the experiments, we use our neural network model which is

trained using the random circuits with single- and two-qubit gates.

A. Use Case 1: Evaluate different circuit designs

There can be different circuit designs for the same quantum algorithm. As the number of gates varies in different designs, it is important to judge which design is more reliable at compile time. Here, we use the Lloyd QPE gate [3] and the modified Lloyd QPE gate [25] for the four-bit QPE algorithm in our experiment. The modified QPE algorithm reduces the number of CNOT gates in the benchmark and tends to be more reliable than the original algorithm. Table X shows the estimated PST and IST of different models and the observed PST and IST. The estimated values of our model are much closer to the observed ones. From the estimated values we can infer that the modified Lloyd QPE algorithm is more reliable than the original algorithm on the 20-qubit computer. Since the estimated IST of four qubit Lloyd QPE gate is smaller than the parameter K ($estimated\ IST = 1.233 < 2$), we can predict that 20-qubit computer cannot infer the correct answer for this algorithm, which matches the observation of experiments on actual machines ($IST = 0.495 < 1$).

TABLE X
ESTIMATED AND OBSERVED PST AND IST OF QPE ALGORITHM

	NN PST	ESP	Qiskit PST	PST	NN IST	Qiskit IST	IST
Lloyd QPE	0.245	0.516	0.588	0.127	1.233	8.71	0.495
Modified Lloyd QPE	0.519	0.631	0.704	0.421	6.052	8.431	3.338

B. Use Case 2: Improve transpiler mapping pass

Qiskit includes a noise-adaptive transpiler for logical to physical qubit mapping. The mapping process has the following steps. First, based on the logical quantum circuit, the transpiler finds an adjacent logical qubit pair that has most CNOT gates in between. In step two, the transpiler selects the best adjacent physical qubit pair according to the backend properties and map the logical qubit pair to the physical qubit pair. In the third step, the transpiler selects an adjacent logical qubit and map it to the best adjacent physical qubit. This step is repeated until all the logical qubits are mapped to physical qubits. Nevertheless, selecting the best adjacent physical qubit pair in step two cannot guarantee that the adjacent physical qubits selected in step three also have a low error rate. The transpiler may map the circuit into a region where there are two qubits with low error rates and all other qubits have high error rates.

To overcome this problem, we can generate different circuit mapping at compile time by selecting different initial mapping. First, we find an adjacent logical qubit pair that has most CNOT gates in between, then we map this logical qubit pair to all the possible adjacent qubit pairs on the quantum computer to generate different circuit mappings. Our reliability model can be used to compare these circuit mappings at

compile time and select the top K mapping. The compiler returns these mappings and expects that the best mapping is included. The number of connections between qubits is 23 on *ibmq-boeblingen*, therefore, we will generate 23 designs with different initial mappings. In Figure 6, we show the experiments with the BV5 algorithm. We compared PST of the top 8 mappings selected by different models. The top 8 mappings selected by our neural network model include the mapping with the highest PST which is 0.65. The naive qubit mapping has PST equals to 0.48. As a result, our transpiler optimization approach finds the qubit mapping with an increment of 35.4% in PST. Comparing with the results from other models, the best mapping from the ESP model has PST of 0.62 and the best mapping from the Qiskit noise model has PST of 0.61. Comparing to these two models, our model finds the mapping with 5% improvement in PST.

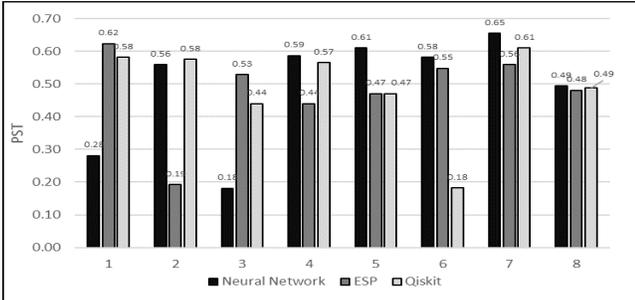


Fig. 6. Observed PST of different models with top 8 estimated PST

C. Use Case 3: Finding ensemble of different mappings

In the prior work of ensemble of different mappings [35], researchers find top K best mappings of a logical circuit and merge the outputs to get an ensemble of diverse mappings. They use ESP to predict PST and select the top K mappings with the highest ESP that are deemed to be the mappings that have top K PST. The diverse mappings are the isomorphic subgraphs of the initial best mapping. They run their experiment on fourteen qubits *ibmq-16-melbourne* machine and avoid using two noisy qubits, which puts constraints on finding diverse mappings. In our case study, we run the BV5 algorithm on twenty qubits *ibmq-boeblingen* machine. As our experiment includes more qubits and a larger search space of different mappings, we noticed that the top K mappings with high ESP do not have good coverage of top K PST.

In order to find the best mapping with the highest PST, we use our neural network model to estimate PST of different mappings. As shown in Figure 7, when selecting the top 4 designs, our estimation model includes the mapping with the highest PST. After generating the ensemble of the top 4 mappings, the PST of our ensemble is 0.34, which is 1.62x and 1.03x greater than the PST of ensemble generated from the ESP model ($PST = 0.21$) and the Qiskit model ($PST = 0.33$). Taking the ensemble of different mappings (EDM) should improve the resulting IST. The IST of our ensemble is 5.2, which is 2.3x and 1.2x greater than the IST of ensemble generated from the ESP model ($IST = 2.3$) and the Qiskit model ($IST = 4.3$)

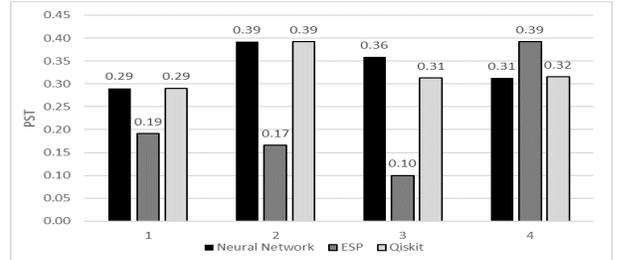


Fig. 7. Observed PST of different models with top 4 estimated PST

D. Use Case 4: Reliability modeling for crosstalk errors

When multiple quantum gates are executed in parallel, crosstalk between the quantum gates can corrupt the quantum state and lead to incorrect results. A recent work [41] on crosstalk error mitigation has proposed software techniques by inserting barriers to avoid the parallel execution of susceptible quantum gates. In this section, we will demonstrate the extensibility of our model. By including the parameters that are associated with the crosstalk error, our model can successfully estimate the effect of crosstalk error mitigation.

We introduce a new parameter $P(i, j) \parallel (k, l)$ which represents the number of parallel gates on connection pair (i, j) and (k, l) . (For example, when a CNOT gate on qubit 3 and qubit 4 is executed in parallel with a CNOT gate on qubit 6 and qubit 7, $P(3, 4) \parallel (6, 7)$ is incremented by 1.) Since the number of connection pairs is large, we only consider the connection pairs between which high crosstalk errors occur. On the 20-qubit IBM Q quantum computer, we include six new parameters for six such pairs. By including these new parameters, our reliability estimation model successfully estimates the crosstalk errors. In our experiment, we first run quantum circuits. Then, we mitigate the crosstalk errors by inserting barriers to avoid the parallel execution of quantum gates. The results of different models are shown in Figure 8. The neural network model is our previous model without additional parameters. The re-trained neural network model is our model with new parameters $P(i, j) \parallel (k, l)$. The y-axis is the ΔPST , which is the change in PST after crosstalk mitigation. Since the ESP model and the neural network model without additional parameters do not consider the execution order of the quantum gates, they will provide the same estimation for the circuit with and without crosstalk error mitigation, i.e., $\Delta PST = 0$. The Qiskit noise model does not consider crosstalk errors. Inserting barriers increases the circuit execution time, which leads to higher decoherence errors. Therefore, the estimated PST of Qiskit noise model will become lower after inserting the barriers. In other words, based on the Qiskit model, ΔPST are negative for all the benchmarks. In contrast, our retrained model successfully estimates the improvement in PST after mitigating the crosstalk error.

IX. CONCLUSIONS

In this paper, we propose a reliability estimation model based on the characteristics of quantum circuits and the topology of the target quantum computer. We show that our reliability model outperforms the noise model from the Qiskit

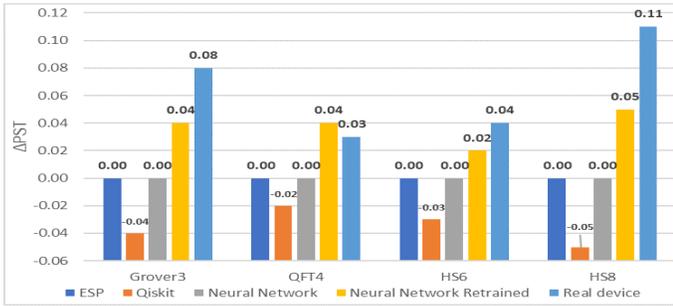


Fig. 8. The improvement of PST from a software mitigation scheme estimated using different models for various benchmarks

and the ESP model from previous work. Our model can be used for guiding the quantum circuit design. As a result of the low execution overhead, our model can aid the compiler to select the mappings with higher accuracy at compile time.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their valuable comments. This work is funded in part by NSF grants 1717550 and 1908406.

REFERENCES

- [1] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, no. 5, p. 052328, 2004.
- [2] H. Abraham, I. Y. Akhalwaya, G. Aleksandrowicz, T. Alexander, G. Alexandrowics, E. Arbel, A. Asfaw, C. Azaustre, P. Barkoutsos, G. Barron, L. Bello, Y. Ben-Haim, D. Bevenius, L. S. Bishop, S. Bosch, D. Bucher, CZ, F. Cabrera, P. Calpin, L. Capelluto, J. Carballo, G. Carrascal, A. Chen, C.-F. Chen, R. Chen, J. M. Chow, C. Claus, C. Clauss, A. J. Cross, A. W. Cross, J. Cruz-Benito, Cryoris, C. Culver, A. D. Córcoles-Gonzales, S. Dague, M. Dartailh, A. R. Davila, D. Ding, E. Dumitrescu, K. Dumon, I. Duran, P. Eendebak, D. Egger, M. Everitt, P. M. Fernández, A. Frisch, A. Fuhrer, I. GOULD, J. Gacon, Gadi, B. G. Gago, J. M. Gambetta, L. Garcia, S. Garion, Gawel-Kus, J. Gomez-Mosquera, S. de la Puente González, D. Greenberg, J. A. Gunnels, I. Haide, I. Hamamura, V. Havlicek, J. Hellmers, L. Herok, H. Horii, C. Howington, S. Hu, W. Hu, H. Imai, T. Imamichi, R. Iten, T. Itoko, A. Javadi-Abhari, Jessica, K. Johns, N. Kanazawa, A. Karazeev, P. Kassebaum, A. Kovyshin, V. Krishnan, K. Krsulich, G. Kus, R. LaRose, R. Lambert, J. Latone, S. Lawrence, D. Liu, P. Liu, P. B. Z. Mac, Y. Maeng, A. Malyshev, J. Marecek, M. Marques, D. Mathews, A. Matsuo, D. T. McClure, C. McGarry, D. McKay, S. Meesala, A. Mezzacapo, R. Midha, Z. Mineev, M. D. Mooring, R. Morales, N. Moran, P. Murali, J. Müggenburg, D. Nadlinger, G. Nannicini, P. Nation, Y. Naveh, Nick-Singstock, P. Niroula, H. Norlen, L. J. O'Riordan, P. Ollitrault, S. Oud, D. Padilha, H. Paik, S. Perriello, A. Phan, M. Pistoia, A. Pozas-iKerstjens, V. Prutyantov, J. Pérez, Quintiii, R. Raymond, R. M.-C. Redondo, M. Reuter, D. M. Rodríguez, M. Ryu, M. Sandberg, N. Sathaye, B. Schmitt, C. Schnabel, T. L. Scholten, E. Schoute, I. F. Sertage, N. Shammah, Y. Shi, A. Silva, Y. Siraichi, S. Sivarajah, J. A. Smolin, M. Soeken, D. Steenzen, M. Stypulkoski, H. Takahashi, C. Taylor, P. Taylour, S. Thomas, M. Tillet, M. Tod, E. de la Torre, K. Trabing, M. Treinish, TrishaPe, W. Turner, Y. Vaknin, C. R. Valcarce, F. Varchon, D. Vogt-Lee, C. Vuillot, J. Weaver, R. Wiczorek, J. A. Wildstrom, R. Wille, E. Winston, J. J. Woehr, S. Woerner, R. Woo, C. J. Wood, R. Wood, S. Wood, J. Wootton, D. Yeralin, J. Yu, L. Zdanski, Zoufalc, anedumla, azulenhner, becamorrison, brandhsn, dennis-liu I, drholmie, elfrocampeador, fanizzamarco, gruu, kanejess, klinvill, lerongil, ma5x, merav aharoni, mrossinek, ordmoj, strickroman, tigerjack, yang.luh, and yotamvakninibm, "Qiskit: An open-source framework for quantum computing," 2019.
- [3] D. S. Abrams and S. Lloyd, "Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors," *Physical Review Letters*, vol. 83, no. 24, p. 5162, 1999.
- [4] E. Bernstein and U. Vazirani, "Quantum complexity theory," *SIAM Journal on computing*, vol. 26, no. 5, pp. 1411–1473, 1997.

- [5] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub) graph isomorphism algorithm for matching large graphs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 10, pp. 1367–1372, 2004.
- [6] I. Corporation, "IBM Q Experience," <https://quantum-computing.ibm.com/>, 2019, [Online; accessed 11-20-2019].
- [7] G. M. D'Ariano, M. G. Paris, and M. F. Sacchi, "Quantum tomography," *Advances in Imaging and Electron Physics*, vol. 128, pp. 206–309, 2003.
- [8] J. Du, M. Shi, X. Zhou, Y. Fan, B. Ye, R. Han, and J. Wu, "Implementation of a quantum algorithm to solve the bernstein-vazirani parity problem without entanglement on an ensemble quantum computer," *Physical Review A*, vol. 64, no. 4, p. 042306, 2001.
- [9] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.
- [10] X. Gao and L. Duan, "Efficient classical simulation of noisy quantum computation," *arXiv preprint arXiv:1810.03176*, 2018.
- [11] X. Gao, Z. Zhang, and L. Duan, "An efficient quantum algorithm for generative machine learning," *arXiv preprint arXiv:1711.02038*, 2017.
- [12] D. Greenbaum and Z. Dutton, "Modeling coherent errors in quantum error correction," *Quantum Science and Technology*, vol. 3, no. 1, p. 015007, 2017.
- [13] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Physical review letters*, vol. 79, no. 2, p. 325, 1997.
- [14] J. Hsu, "Ces 2018: Intel's 49-qubit chip shoots for quantum supremacy," Sep 2019. [Online]. Available: <https://spectrum.ieee.org/tech-talk/computing/hardware/intels-49qubit-chip-aims-for-quantum-supremacy>
- [15] J. Kelly, "Preview of bristlecone, google's new quantum processor," Sep 2019. [Online]. Available: <https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html>
- [16] D. Koch, A. Torrance, D. Kinghorn, S. Patel, L. Wessing, and P. M. Alsing, "Simulating quantum algorithms using fidelity and coherence time as principle models for error," *arXiv preprint arXiv:1908.04229*, 2019.
- [17] F. Lardinois, "Ibm will soon launch a 53-qubit quantum computer," Sep 2019. [Online]. Available: <https://techcrunch.com/2019/09/18/ibm-will-soon-launch-a-53-qubit-quantum-computer/>
- [18] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
- [19] E. Magesan, J. M. Gambetta, and J. Emerson, "Characterizing quantum gates via randomized benchmarking," *Physical Review A*, vol. 85, no. 4, p. 042311, 2012.
- [20] A. Mandviwalla, K. Ohshiro, and B. Ji, "Implementing grover's algorithm on the ibm quantum computers," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 2531–2537.
- [21] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New Journal of Physics*, vol. 18, no. 2, p. 023023, 2016.
- [22] D. C. McKay, S. Sheldon, J. A. Smolin, J. M. Chow, and J. M. Gambetta, "Three-qubit randomized benchmarking," *Physical review letters*, vol. 122, no. 20, p. 200502, 2019.
- [23] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, "Efficient z gates for quantum computing," *Physical Review A*, vol. 96, no. 2, p. 022330, 2017.
- [24] S. T. Merkel, J. M. Gambetta, J. A. Smolin, S. Poletto, A. D. Córcoles, B. R. Johnson, C. A. Ryan, and M. Steffen, "Self-consistent quantum process tomography," *Physical Review A*, vol. 87, no. 6, p. 062119, 2013.
- [25] H. Mohammadbagherpoor, Y.-H. Oh, A. Singh, X. Yu, and A. J. Rindos, "Experimental challenges of implementing quantum phase estimation algorithms on ibm quantum computer," *arXiv preprint arXiv:1903.07605*, 2019.
- [26] P. Murali, J. M. Baker, A. Javadi-Abhari, F. T. Chong, and M. Martonosi, "Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM, 2019, pp. 1015–1029.
- [27] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.
- [28] S. Nishio, Y. Pan, T. Satoh, H. Amano, and R. Van Meter, "Extracting success from ibm's 20-qubit machines using error-aware compilation," *arXiv preprint arXiv:1903.10963*, 2019.
- [29] M. Ozols, "Clifford group," *Essays at University of Waterloo*, Spring, 2008.

- [30] M. Paris and J. Rehacek, *Quantum state estimation*. Springer Science & Business Media, 2004, vol. 649.
- [31] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.
- [32] K. Rudinger, T. Proctor, D. Langharst, M. Sarovar, K. Young, and R. Blume-Kohout, "Probing context-dependent errors in quantum processors," *Physical Review X*, vol. 9, no. 2, p. 021045, 2019.
- [33] A. Sung, "Ranking importance of input parameters of neural networks," *Expert Systems with Applications*, vol. 15, no. 3-4, pp. 405-411, 1998.
- [34] M. Takita, A. W. Cross, A. Córcoles, J. M. Chow, and J. M. Gambetta, "Experimental demonstration of fault-tolerant state preparation with superconducting qubits," *Physical review letters*, vol. 119, no. 18, p. 180501, 2017.
- [35] S. S. Tannu and M. Qureshi, "Ensemble of diverse mappings: Improving reliability of quantum computers by orchestrating dissimilar mistakes," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 2019, pp. 253-265.
- [36] S. S. Tannu and M. K. Qureshi, "Not all qubits are created equal: a case for variability-aware policies for nisq-era quantum computers," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM, 2019, pp. 987-999.
- [37] W. Van Dam, S. Hallgren, and L. Ip, "Quantum algorithms for some hidden shift problems," *SIAM Journal on Computing*, vol. 36, no. 3, pp. 763-778, 2006.
- [38] I. Vankov, D. Mills, P. Wallden, and E. Kashefi, "Methods for classically simulating noisy networked quantum architectures," *Quantum Science and Technology*, vol. 5, no. 1, p. 014001, 2019.
- [39] D. Wecker, M. B. Hastings, N. Wiebe, B. K. Clark, C. Nayak, and M. Troyer, "Solving strongly correlated electron models on a quantum computer," *Physical Review A*, vol. 92, no. 6, p. 062318, 2015.
- [40] J. Liu, G. T. Byrd, and H. Zhou, "Quantum circuits for dynamic runtime assertions in quantum computation," in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2020, pp. 1017-1030.
- [41] P. Murali, D. C. McKay, M. Martonosi, and A. Javadi-Abhari, "Software mitigation of crosstalk on noisy intermediate-scale quantum computers," *arXiv preprint arXiv:2001.02826*, 2020.