QuEST: Quantum Circuit Output Estimation using Gaussian Distribution Analysis

Shamik Kundu‡, Navnil Choudhury‡, and Kanad Basu
Department of Electrical and Computer Engineering, University of Texas at Dallas, Richardson, TX, USA

2024 25th International Symposium on Quality Electronic Design (ISQED) | 979-8-3503-0927-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISQED60706.2024.10528722

***Abstract*—Quantum computing has holds considerable potential for accelerating computational tasks beyond the capabilities of classical computation. However, a major obstacle arises from the delicate nature of quantum hardware, where quantum gates and qubits, the fundamental components of a quantum circuit (QC), are vulnerable to external interference. Consequently, even a simple QC can produce significantly noisy output. This noise introduces uncertainty, making it difficult to ascertain whether the output represents meaningful computation or merely random noise. This uncertainty raises questions regarding the fidelity of a QC and the extent to which we can rely on its output. Existing classification-based approaches for output estimation have limitations, including the potential for incorrect results due to misclassification or the requirement for a large number of measurements, which can be expensive. To circumvent this, in this paper, we propose QuEST, which introduces an efficient technique for estimating the output of a QC by analyzing probability distributions of post-measurement data. Specifically, the QuEST framework employs Gaussian distribution functions to compare the measured distribution of a circuit with a pre-trained distribution obtained from a training circuit dataset. Moreover, we reformulate this problem by leveraging the properties of sequential time series, thereby deriving a straightforward and intuitive metric to measure the confidence of the QC output. By utilizing this metric, the QuEST framework monitors fidelity evolution over time as the QC interacts with its external environment, enabling the system to preemptively halt QC execution upon reaching a specified confidence threshold. When evaluated against state-of-the-art benchmark quantum circuits, our proposed QuEST framework accurately estimates the output of 100% of the benchmark circuits, while significantly achieving speedup up to 58.3× compared to a standard QC execution. These results highlight the efficiency of our framework and its potential for practical quantum computing applications.**

***Index Terms*—Quantum Computing, Output Estimation, Statistical distribution, Time Series.**

I. Introduction

Quantum computers, like Google's Sycamore processor, exploit quantum mechanics to outperform classical counterparts with qubits and entanglement, as seen in Simon's problem. However, they face challenges like noise, limited qubits, and connectivity. Noisy Intermediate-Scale Quantum (NISQ) computing mitigates these issues using error-mitigation techniques. Transpilation, mapping quantum circuits to physical qubits, is essential but complex due to varying qubit characteristics and restricted connectivity [1]. This can introduce additional noise, even with high-quality qubits.

To maximize the utility of current NISQ processors, it is vital to develop a methodology for accurately predicting quantum circuit outputs with high fidelity, avoiding resource-intensive, noisy transpilation processes. Quantum hardware providers are already offering tools to optimize circuits using simplified noise models. Measurement on modern quantum computers involves coupling sensitive equipment, introducing noise, to gather quantum state information. Processing this quantum data through repeated measurements (shots) on classical computers converges toward a correct solution [2]. However, classification-based methods have limitations, and the need for a high number of measurements, incurring significant costs. Additionally, repeated circuit execution on cutting-edge quantum computers leads to congestion, with users enduring lengthy waiting times. To optimize resource allocation and traffic management, it is crucial to identify circuits that do not require multiple iterations, enhancing efficiency.

In this paper, we present QuEST, a framework designed to address the critical need for efficient and reliable quantum circuit (QC) execution on noisy intermediate-scale quantum (NISQ) computers. QuEST employs Gaussian distribution functions to compare the measured distribution of a circuit with a pre-trained distribution from a comprehensive training circuit dataset encompassing several implementations of an extensive variety of quantum circuits. The use of Gaussian distribution in our framework is driven by the noise inherent in quantum computers, which reduces confidence in correctness due to altered measurement outcomes. By using sequential time series properties, we reformulate the problem and create a simple metric to measure QC output confidence. QuEST not only accurately estimates quantum circuit outputs but also monitors fidelity evolution as the QC interacts with its environment. This allows for preemptive QC execution halts based on a confidence level associated with the Gaussian distribution's z-score, optimizing resource utilization and minimizing potential execution errors. Furthermore, when prediction confidence is low, QuEST incorporates multiple execution instances from the time series to boost the z-score, increasing confidence in the output estimate. We tested QuEST on state-of-the-art benchmark quantum circuits and achieved 100% accurate estimation while significantly speeding up execution, up to 34.84× compared to standard QC execution. These results underscore the efficiency and practical potential of our framework in quantum computing applications. The key contributions of this paper are as follows:

- In this paper, we, for the first time, propose QuEST, a framework which addresses the need for efficient and reliable estimation of quantum circuit output in the presence

‡Authors have equal contribution.
This research is supported by NSF grant #2228725.
Corresponding Author: Shamik Kundu (*shamik.kundu@utdallas.edu*).

of noise in NISQ computers.

- QuEST employs Gaussian distribution functions to compare the measured distribution of a circuit with a pre-trained distribution obtained from a comprehensive training circuit dataset, enabling accurate estimation of quantum circuit outputs.
- The framework leverages the properties of sequential time series to reformulate the problem and develop a straightforward metric for measuring quantum circuit output with an associated confidence. This allows for monitoring the evolution of fidelity over time as the quantum circuit interacts with its external environment.
- When evaluated against state-of-the-art benchmark quantum circuits, our proposed QuEST framework accurately estimates the output of 100% of the benchmark circuits, while significantly achieving speedup of up to 34.84× compared to a standard quantum circuit execution.

The rest of the paper is organized as follows. Section II outlines the background information on quantum circuits and noise in quantum computers and the related efforts in this domain. The corresponding methodology to develop the proposed QuEST framework is depicted in Section III. The efficiency of QuEST is demonstrated in Section IV. Finally, Section V concludes the paper.

II. Background and Related Work

A. Basics of Quantum Circuits

Quantum computers can be categorized into two types: quantum annealers and universal gate-based processors. This work specifically focuses on gate-based computers. In gate-based quantum computing, qubits undergo a series of quantum gate operations, following a quantum circuit (QC) description. These gates induce transformations on the qubit states, represented by unitary matrices, ensuring computation reversibility.

Quantum computing typically involves three key stages: state preparation, computation, and measurements. In the state preparation stage, input qubits are initialized, often using quantum phenomena like superposition. This enables qubits to exist in multiple states simultaneously, facilitating parallel exploration of potential solutions.

The computation stage utilizes a sequence of gates defined by the quantum circuit to manipulate and transform qubit states. This stage performs core computational tasks of a quantum algorithm, leveraging quantum parallelism and entanglement for exponentially efficient computations.

In the final stage, measurements collapse qubits from quantum superposition to classical states. This transition provides classical information for interpretation by classical systems, forming the basis of quantum computing and driving advancements in cryptography, optimization, and simulation.

B. Noise in Quantum Computers

Research into quantum gate fidelity and qubit noise is active in the realm of NISQ systems. Randomized benchmarking, a well-established protocol, has been a prominent method for assessing quantum operation error rates [3]. Recent advancements aim to broaden its applicability to a wider range of quantum operations [4] and make it more adaptable for larger circuits involving multiple qubits [5].

Achieving precise measurements on real quantum hardware is essential due to higher error rates in quantum gates. Early research focused on improving output fidelity through circuit compilation techniques, like gate scheduling and CNOT rerouting [6], [7]. Contemporary efforts emphasize optimizing hardware-specific aspects and adopting noise-aware qubit mapping strategies [1], [8]. Transpiling quantum circuits according to the processor's architecture and noise characteristics is now crucial for achieving optimal performance.

Fidelity estimation in quantum computing is a complex field with diverse approaches. Some use quantum algorithms for their speedup [9], while others explore statistical techniques and polynomial fitting [10]. Machine learning, including shallow neural networks [11], and newer architectures like graph transformers [12], is also being applied.

However, challenges remain in ensuring convergence for qubit discriminator methods. Currently, there's no systematic way to bound classification errors, leading to extensive quantum circuit sampling and testing against a threshold. This inefficiency underscores the limitations of current qubit discriminator methods.

III. Proposed QuEST Framework

In this section, we demonstrate our proposed framework, QuEST, with the aim of reducing the cost of quantum computation. The cost of computing is mainly incurred due to the noise present in quantum circuits, which necessitates several instances of a circuit to be executed before obtaining a satisfactory output confidence. QuEST enables a significant reduction in computation time, by decreasing the number of computing instances for each circuit by leveraging time series analysis of quantum circuit execution, accompanied by statistical decisions. Figure 1 illustrates the process flow of our proposed framework, consisting of two major components: Statistical Distribution obtained by Quantum Circuit execution, followed by Output Estimation. Subsequently, the Quantum circuit is executed on a quantum computer to obtain the normalized prediction magnitude $\mathcal{N}_{mi}$ from prediction count $\mathcal{P}_{ci}$ for each computation instance s_i. We will construct a normal distribution and determine the confidence threshold $\mathcal{TH}$ and generate the confidence interval $\mathcal{C}_m$.

Following this, in QuEST, we utilize time series analysis of outputs. This is performed in order to take advantage of early preempting, *i.e.,* stop execution and exit when $\mathcal{TH}$ is achieved. Otherwise, we move on to stack all instances and boost confidence until $\mathcal{TH}$ is achieved.

A. Statistical Distribution from Quantum Circuit Execution

Qubits are fundamental units in quantum computing that allow for the encoding of 2^n states using n-bits. However, the practical performance of quantum computers is significantly affected by the presence of noise. This noise introduces errors,

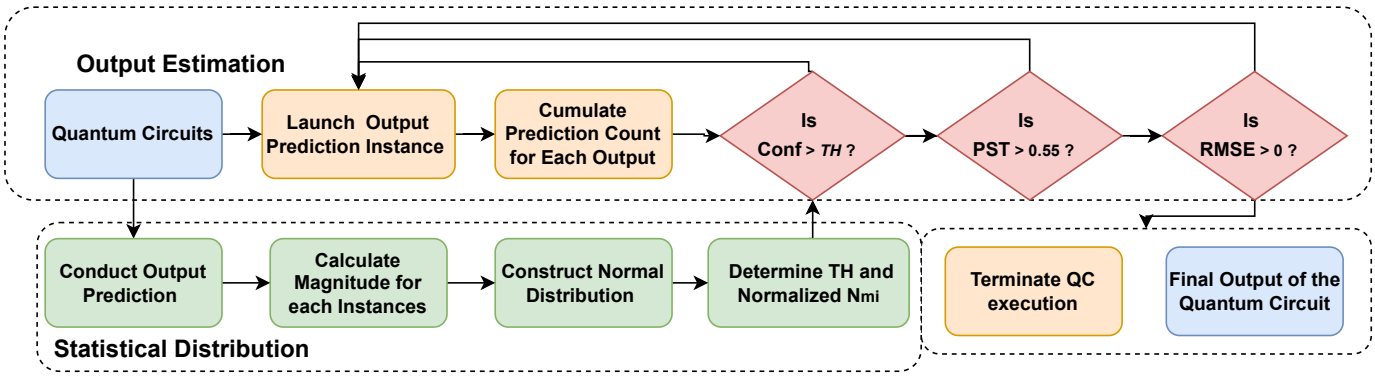


Fig. 1: Process Flow of the Proposed QuEST Framework.

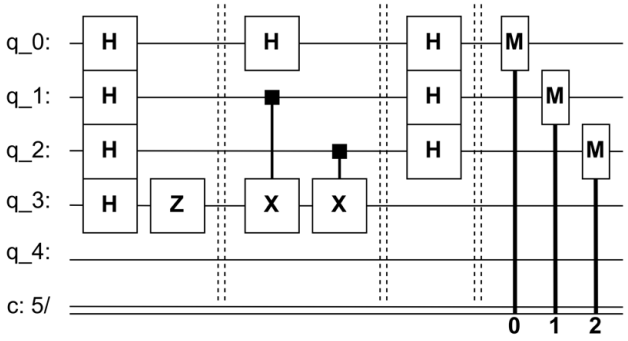


Fig. 2: An example of a circuit with three measured qubits.

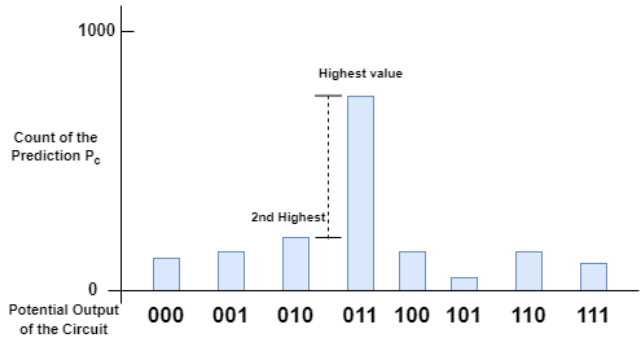


Fig. 3: Prediction Count $\mathcal{P}_c$ for one computation instance (1K).

resulting in a reduction in the fidelity of the output produced by quantum circuits. In order to subvert the influence of the noise, significant amount of calculation instances $S \triangleq s_i \dots s_n$ are required to achieve accurate results with relatively high confidence. These instances are represented by the number of 'shots' during execution of a quantum circuit.

Although a single quantum computer can execute multiple circuits consecutively, the noise has a different impact on each individual circuit. Performing a statistical distribution for each circuit individually during execution adds an unnecessary overhead, and is infeasible. To subvert this, we propose using a standard benchmark suite of NISQ circuits that encompasses various different architectures, in order to encompass the

TABLE I: Prediction count for a 3-bit output quantum circuit.

	Prediction Output Counts								Magnitude	
Ins	000	001	010	011	100	101	110	111	$\mathcal{M}_{si}$	$\mathcal{N}_{mi}$
1	8	55	17	80	55	693	22	70	8.66	1.26
2	11	90	13	27	58	705	45	51	7.83	1.14
3	8	65	19	79	54	688	17	70	8.71	1.26
4	16	67	9	105	44	637	21	101	6.07	0.88

different scenarios of the impact of noise on a circuit. To this end, for a prediction distribution of a quantum circuit, we propose to obtain $\mathcal{M}_{si}$, which is a metric that determines the ratio of the prediction magnitudes of the highest value to the second highest, aggregated over the benchmark suite of NISQ circuits we consider.

For example, Figure 2 illustrates a quantum circuit that measures 3 qubits (Bernstein-Vazirani circuit). Hence, there are 2^3 potential outputs for this circuit and its prediction distribution for the minimum computation instance s_i is demonstrated in Figure 3. Typically, one output has a significantly higher prediction count $\mathcal{P}_{ci}$ than the rest of the outputs, such as "011" $\mathcal{P}_{011i}$ in Figure 3. Meanwhile, the output with the second highest $\mathcal{P}_{ci}$ is "010" $\mathcal{P}_{010i}$. Ideally, the $\mathcal{P}_{ci}$ of the correct output should be significantly higher than the rest of the possible outcomes such as $\mathcal{P}_{011i}$, in Figure 3. $\mathcal{M}_{si}$, aggregated over our benchmark suite is evaluated, in order to obtain the distribution curve. For circuits with a probability distribution as output, we consider the ratio of the number of times the probability distribution occurs to the total number of executions. The highest value yielded by this ratio is designated as $\mathcal{P}_{1st}$ and the output distribution with the second highest number of occurrences is assigned to $\mathcal{P}_{2nd}$. Following this, the quantum circuits are executed on the backend.

The ratio, $\mathcal{M}_{si}$ is computed as follows :

$$\mathcal{M}_{si} = \frac{\mathcal{P}_{1st}}{\mathcal{P}_{2nd}},\ for\ s_i : (\mathcal{P}_{1st} : \mathcal{P}_{011i}, \mathcal{P}_{2nd} : \mathcal{P}_{010i}) \quad (1)$$

In Equation 1, we obtain $\mathcal{P}_{1st}$ as $\mathcal{P}_{011}$, and $\mathcal{P}_{2nd}$ as $\mathcal{P}_{010}$, from Figure 3. Table I shows an example of four sequential computation instances from a quantum circuit with 3-bit output. Columns 2-9 show the prediction count for each output, $\mathcal{M}_{si}$

represents the magnitude of $\mathcal{P}_{1st}$ and $\mathcal{P}_{2nd}$ for each instance, $\mathcal{N}_{mi}$ represents the normalized magnitude for each instance.

Following the generation of $\mathcal{M}_{si}$, it can be observed that for different benchmarks, different normal distributions are achieved. However, we are able to normalize the data obtained, since the variance in the peak of the normal graph does not change. Normalizing further reduces the error margin. For all quantum circuit benchmarks, we calculate the $\mathcal{M}_{si}$ for each computation instance s_i, and construct a normal distribution to map all the potential $\mathcal{M}_{si}$ (We will later discuss the benchmarks in the QASMBench suite, which consist of specific circuits used for our experiments [13]. These benchmarks serve as standardized sets of quantum circuits to evaluate and compare different quantum algorithms and architectures, including the impact of noise.). Since we are using different benchmark circuits, we will normalize all $\mathcal{M}_{si}$s into $\mathcal{N}_{mi}$s from the same benchmark with the following equation:

$$\mathcal{N}_{mi}^{\ normalized} = \frac{\mathcal{M}_{si}}{Avg(\sum \mathcal{M}_{si})} \tag{2}$$

Therefore, we can obtain the normal distribution of the normalized $\mathcal{N}_{mi}$s from all benchmarks, as shown in Figure 4.

B. Output Estimation from Quantum Circuit Execution

Following the generation of the $\mathcal{N}_{mi}$, output estimation is performed for each new quantum circuit being executed. This is accomplished by taking all the instances of the new quantum circuit and evaluating its Z-score, which is an integral part of constructing confidence intervals, and subsequently, to evaluate confidence. A higher confidence threshold will result in a wider confidence interval. This is because a higher confidence level demands a greater level of precision, which necessitates a wider range of plausible values in the confidence interval. Following the generation of the confidence, we evaluate the output using Probability of Successful Trial (PST), in order to ascertain whether the generated output is correct.

$$\mathcal{Z} \ - Score = \frac{\mathcal{N}_{mi} - \mu}{\sigma} \tag{3}$$

Here Z-score, which defines the position of the $\mathcal{N}_{mi}$ in terms of distance from the mean. μ is the mean of the obtained normal distribution, and σ is the standard deviation of normal distribution. This confidence obtained from the z-score

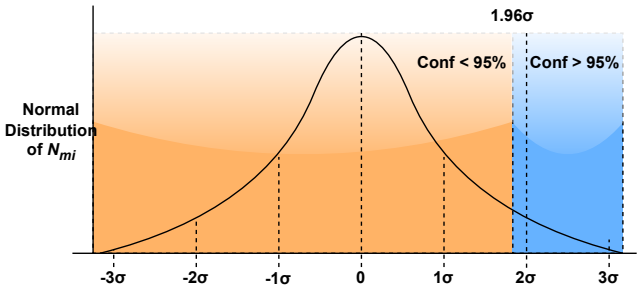


Fig. 4: Example Normal Distribution for Predication Magnitude $\mathcal{N}_{mi}$.

Algorithm 1 Output Determination from QC Execution

Input: $\mathcal{TH}$, $\mathcal{P}_{si}$

Output: Predicted Output $\mathcal{P}_o$

Initialize $\mathcal{P}_{1st}$, $\mathcal{P}_{2nd} = \mathcal{P}_{c1}$
$\mathcal{M}_s = \frac{\mathcal{P}_{1st}}{\mathcal{P}_{2nd}}$, $\mathcal{M}_{ci} = \frac{\mathcal{P}_{1st}}{\mathcal{P}_{2nd}}$, $\mathcal{N}_{mi} = \frac{\mathcal{M}_{ci}}{Avg(\sum \mathcal{M}_{si})}$
$\mathcal{Z} = \frac{\mathcal{N}_{mi} - \mu}{\sigma}$
while $\mathcal{Z} < \mathcal{TH}$ **do**
    $\mathcal{P}_{1stn}, \mathcal{P}_{2ndn} \leftarrow \mathcal{P}_{cn}$
    $\mathcal{P}_{1st} = \sum \mathcal{P}_{1st1} \ldots \mathcal{P}_{1stn}$
    $\mathcal{P}_{2nd} = \sum \mathcal{P}_{2nd1} \ldots \mathcal{P}_{2ndn}$
    $\mathcal{M}_{ci} = \frac{\mathcal{P}_{1st}}{\mathcal{P}_{2nd}}$
    $\mathcal{N}_{mi} = \frac{\mathcal{M}_{ci}}{Avg(\sum \mathcal{M}_{si})}$
end while
if $\mathcal{PST}(\mathcal{Z}) > 0.55$ **then**
    Extract $RMSE, \mathcal{N}_{mi}, Conf$
else goto: while
end if
if $RMSE > 0$ **then Exit**
else goto: while
end if
$\mathcal{P}_o \leftarrow$ Output of $\mathcal{P}_{1st}$

furnished by our model is checked against the pre-specified threshold.

There can be two cases that can arise following the confidence level checking, which demonstrates the advantages of the time-series-based approach in QuEST.

- **Case 1:** When the calculated confidence, is higher than the desired confidence margin, the model employs an early preempt. In other words, it stops the execution before completing all iterations. This early preemption can occur after at least two instances have been executed. By implementing this approach, a significant amount of computational time is saved.
- **Case 2:** When the obtained confidence level is lower than the desired confidence level, a repetition of the process is performed starting from distribution generation until Z-score checking. This repetitive process incorporates each subsequent instance in the time series, on top of the initial instances. As a result, the confidence is changed during each iteration, allowing the possibility of crossing the threshold, following which, PST will be evaluated to determine termination of execution.

In both of the aforementioned scenarios, upon completion of the computational execution, we assess the Root Mean Square Error (RMSE) values to determine the position of the output within the distribution. If the RMSE is found to be imaginary, we disregard the value as it indicates an erroneous output caused by the influence of inherent noise in a quantum circuit.

In our framework, QuEST, after obtaining a pre-specified confidence threshold $\mathcal{TH}$, and comparing the confidence ob-

tained with $\mathcal{TH}$ for all benchmarks, we leverage it to optimize the computation instances S required for determining the final output as well as execution completion stage. Algorithm 1 demonstrates the process flow of Output Estimation. We will initialize the $\mathcal{P}_{1st}$ and $\mathcal{P}_{2nd}$ with the first computation instance s_i [*lines 1 and 2*]. In case the normalized $\mathcal{N}_{mi}$ furnishes a confidence level, estimated from Z-score, is higher than pre-specified threshold $\mathcal{TH}$, the execution preemptively stops since its output confidence has been satisfied [*lines 3 and 4*]. If the normalized $\mathcal{N}_{mi}$ outputs a confidence value that is lesser than confidence margin metric however, then for all possible outputs, for every other computation instance, we will stack the $\mathcal{P}_{ci}$s, and determine the sums of $\mathcal{P}_{1st}$ and $\mathcal{P}_{2nd}$ [*lines 5 to 7*]. Table II shows the predicted output where all prediction counts $\mathcal{P}_{ci}$s are added after each instance. $\mathcal{M}_{ci}$ represents the magnitude for the sum of $\mathcal{P}_{1st}$ and $\mathcal{P}_{2nd}$, and $\mathcal{N}_{mi}$ represents the normalized $\mathcal{M}_{ci}$. Sequentially, we identify the differences between the prediction counts with the magnitude $\mathcal{M}_{ci}$, and normalize it with the average magnitude of all computation instances $Avg(\sum \mathcal{M}_{si})$ [*lines 8 and 9*]. Following this, when the Z-score is greater than the pre-specified threshold, in order to check the correctness of the output PST of that instance is estimated [*lines 10, 11*]. If PST is evaluated to be greater than 0.55, i.e. output is correct with sufficient margin, we extract the RMSE, $\mathcal{N}_{mi}$, and Confidence (estimated from the Z-score of that instance) [*line 12*]. If PST is lesser than 0.55, we continue the execution until we obtain a suitable PST [*lines 13 and 14*]. If PST is satisfactory, we evaluate the RMSE to ensure the output confidence does not lie towards the lower end of the normal distribution, which signifies noisy and erroneous output, i.e. if RMSE is lesser than 0 [*line 15*]. In case RMSE is greater than 0, we terminate the execution, thus incurring lower overhead, and taking advantage of the early preempt of QuCEST. If RMSE is lesser than 0, we repeat the process, until all the aforementioned conditions of Z-score, PST, and RMSE are satisfied [*lines 16 and 17*].

Thus, we are able to estimate the output of the quantum computer with the minimum computation instances required. To account for noise distribution in different systems, different quantum computers will be tested to optimize the desired number of computation instances. In Section IV, we will evaluate QuEST the effects of various $\mathcal{TH}$, and the performance improvement for different benchmarks.

IV. Evaluation

A. Experimental Setup

To assess the efficiency of our proposed QuEST framework, we conducted evaluations using quantum circuits from the QASMBench Suite. The experiments were executed on the IBM Quantum platform, specifically employing the **ibm_jakarta** backend comprising seven qubits. To accommodate the limitations imposed by the physical constraints of the backend, we focused exclusively on evaluating the small benchmark circuits from the QASMBench Suite that consisted of fewer than seven qubits [13]. This approach ensured that our evaluations aligned with the capabilities and constraints of the chosen physical quantum computing infrastructure.

To obtain accurate estimates of output fidelity over time, each circuit was executed 90 times with 1000 shots per execution, for training. This approach allowed us to capture the necessary statistical information and generate a normal distribution representing the aggregate output from the circuits used for training. An extensive number of circuits have been used in the pre-trained model to avoid model over-fitting.

TABLE II: Sums of Prediction counts for a 3-bit output QC.

	Sum of Prediction Outputs Counts								Magnitude of Sum	
Ins	000	001	010	011	100	101	110	111	$\mathcal{M}_{ci}$	$\mathcal{N}_{mi}$
1	8	55	17	80	55	693	22	70	8.66	1
2	19	145	30	107	113	1398	67	121	9.64	1.16
3	27	210	49	186	167	2086	84	191	9.93	1.18
4	43	277	58	291	211	2723	105	292	9.32	1.19

B. Experimental Results

In this section, we begin by showcasing the normal distribution acquired from a set of diverse benchmark circuits sourced from the QASMBench Suite [13]. These circuits encompass varying depths and widths, ensuring a comprehensive evaluation while preserving generality. The resulting distribution is presented in Figure 5. We subsequently obtain the mean(μ), and standard deviation(σ) from this distribution, which we utilize to evaluate the **Z-score** (from which we obtain confidence) and **RMSE** (from which we determine noisiness) for each circuit we analyze.

We conducted a comprehensive evaluation of our approach on a set of nine benchmark circuits, elucidating the operational principles of our methodology within the context of each circuit's distinct characteristics.

1) Half_Adder: In this experiment, we performed the execution of the half_adder circuit, which possesses a gate depth of 4 and utilizes 3 qubits, on the IBM_Jakarta backend. A maximum of 90 instances were employed to execute the circuit, where each instance represented the circuit's execution with 1000 shots, yielding output counts. Subsequently, a time series analysis was conducted on the obtained outputs,

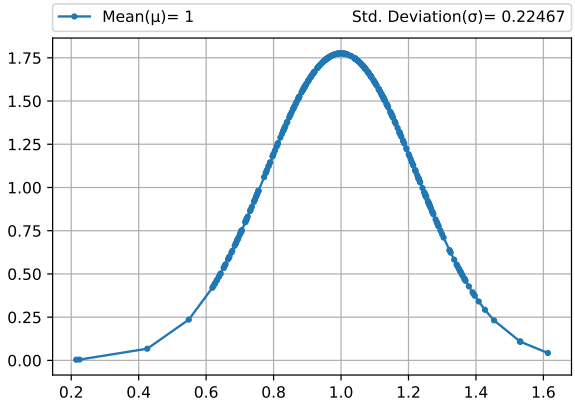
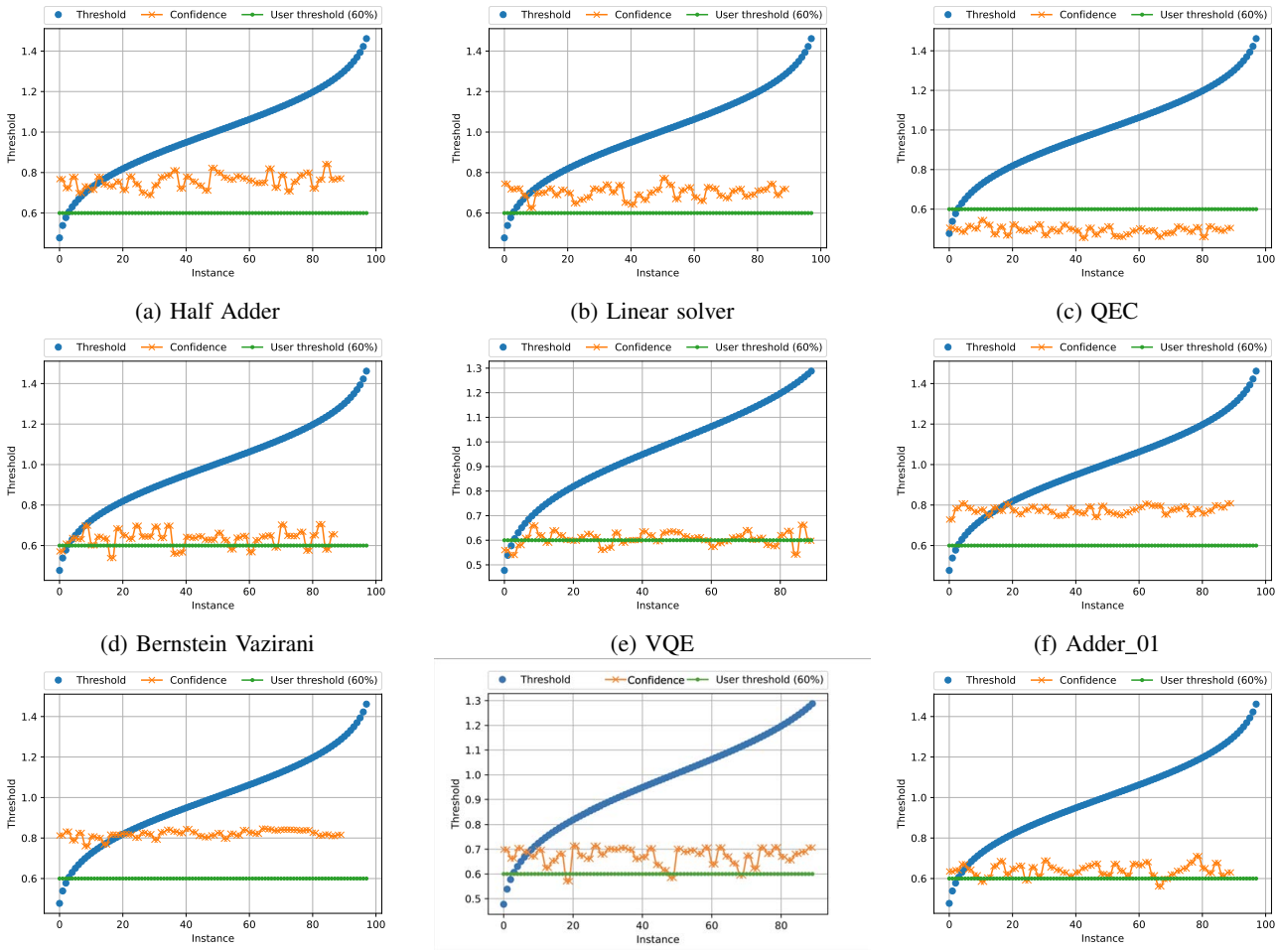


Fig. 5: Distribution obtained from QASMBench benchmark circuits.



(a) Half Adder (b) Linear solver (c) QEC

(d) Bernstein Vazirani (e) VQE (f) Adder_01

(g) Adder_00 (h) QFT (i) HS4

Fig. 6: Efficiency of QuEST framework evaluated over different QASMBench benchmark circuits.

following each instance, to analyze their temporal behavior in noisy settings.

To assess the (N_{mi}) of the circuit, a minimum of two instances were executed to obtain a mean value. The resulting mean (1) and standard deviation (0.2246) derived from the normal distribution curve were utilized to compute the Z-score, serving as a measure of confidence for the circuit's output.

Considering a user-defined confidence threshold of 60%, as depicted in Figure 6a, we conducted a sequential comparison of the confidence achieved after each pair of instances with the threshold until the desired confidence value was attained. For the specific half_adder circuit, a confidence level of 70.11 was achieved following the execution of the first two instances, surpassing the defined threshold. Subsequently, a Probability of Successful Trial (PST) analysis was performed on the output counts obtained from the two instances, yielding a PST value of 68.1, thereby confirming the correctness of the circuit's output. Consequently, the execution was terminated, and the computation concluded, within a time consumption of 0.4 seconds. The values of N_{mi}, Z-score, and RMSE (Root Mean Square Error) during the termination of the computation were recorded as 2.04, 70.11, and 1.77, respectively. As a result, the QuEST framework exhibited a remarkable speedup of 14.5× compared to the baseline execution.

2) Adder_00: In this experiment, we executed the adder circuit under the same execution platform and identical setup conditions. The adder circuit was characterized by a gate depth of 12 and employed 4 qubits. Considering the same user-defined confidence threshold of 60%, as depicted in Figure 6g, we compared the confidence attained after each pair of instances with the threshold until the desired confidence value was achieved. For the particular half_adder circuit, a confidence level of 70.33 was attained following the first two instances, exceeding the defined threshold. Subsequently, a Probability of Successful Trial (PST) was performed on the outputs of the two instances, resulting in a PST value of 66.76, confirming the correctness of the output. Consequently, the execution was terminated, and the computation concluded within a time consumption of 0.56 seconds. As a result, the QuEST framework demonstrated a notable speedup of 54.46× compared to the baseline execution.

3) Adder_01: In this experimental study, we executed the adder circuit under the same execution platform and identical setup conditions, but with a different input. Using the user-defined confidence threshold of 60% (as shown in Figure 6f), we sequentially compared the achieved confidence after each pair of instances with the threshold. For this specific adder circuit with changed inputs, we reached a confidence level of 69.93, after executing the first two instances, surpassing the predefined threshold. Subsequently, the (PST) analysis on the two instances resulted in a PST value of 63.26. As a result, we terminated the execution and completed the computation within a time consumption of 0.53 seconds. In this scenario, the QuEST framework demonstrated a significant speedup of 58.3 $\times$ compared to the baseline execution. The termination values for N_{mi}, Z-score, and RMSE for this circuit were recorded as 2.035, 69.93, and 1.772, respectively. This result also demonstrates that our evaluation framework does not depend on the inputs of the circuit.

4) Hidden Shift: We proceed to execute the hs_4 (Hidden Shift) circuit, characterized by a gate depth of 10, and a qubit count of 4, under identical execution conditions, with the same user defined threshold. For this circuit, we obtain a confidence of 60.47 following the execution of the first two instances, shown in Figure 6i with a PST of 66.51, denoting its correctness. Subsequently, the execution was terminated, within a time consumption of 0.581 seconds. In this scenario, the QuEST framework demonstrated a significant speedup of 32.44 $\times$ compared to the baseline execution. The termination values for N_{mi}, Z-score, and RMSE for this circuit were recorded as 1.85, 60.47, and 1.556, respectively.

5) Bernstein Vazirani: Under identical execution conditions and utilizing the same user-defined threshold, we proceeded to execute the bv_4 (Bernstein Vazirani) circuit. This circuit had a gate depth of 6 and consisted of 4 qubits. Following the execution of the first eight instances, we achieved a confidence level of 60.19, surpassing the threshold, as denoted in Figure 6d with a corresponding PST value of 63.21, indicating the correctness of the output. Consequently, we terminated the execution, completing the computation within a time consumption of 1.8 seconds. Impressively, the QuEST framework demonstrated a substantial speedup of 18.72 $\times$ compared to the baseline execution for this scenario. The recorded termination values for N_{mi}, Z-score, and RMSE for this circuit were 1.846, 60.19, and 1.551, respectively.

6) Quantum Fourier Transform: In this experiment, under the same execution conditions, we proceeded to execute the qft_4 (Quantum Fourier Transform) circuit, consisting a gate depth of 12 and 4 qubits.

Following the execution of the first two instances, we achieved a confidence level of 63.29, surpassing the threshold, shown in Figure 6h. However, the corresponding PST value of 51.21 indicated that the output was not correct. Therefore, we evaluated the subsequent two instances. In this case, we obtained a confidence of 65.19 and a PST of 60.01. Based on these results, we terminated the execution, completing the computation within a time of 1.12 seconds.

Impressively, the QuEST framework demonstrated a substantial speedup of 28.92 $\times$ compared to the baseline execution for this scenario. The recorded termination values for N_{mi}, Z-score, and RMSE for this circuit were 1.951, 65.19, and 1.675, respectively.

7) Variational Quantum Eigensolver: Similar to previous circuits, we execute the vqe_n4 (Variational Quantum Eigensolver) circuit, which has a significant circuit depth of 25, while using 4 qubits.

After performing the initial fourteen iterations, we successfully reached a confidence level of 61.57, which exceeded the required threshold, as demonstrated in Figure 6e The corresponding PST value of 58.21 ensured the accuracy of the output. As a result, we terminated the process, completing the computation within a time of 3.89 seconds. Notably, the QuEST framework demonstrated an impressive speed improvement of 6.42 $\times$ compared to the standard execution in this specific scenario. The recorded termination values for N_{mi}, Z-score, and RMSE for this circuit were 1.87, 61.57, and 1.58, respectively.

8) Linear Solver: We execute the Linear solver with identical operating conditions. The Linear solver uses 3 qubits, and has a gate depth of 12. In the specific scenario described, the circuit achieved a confidence level of 63.77 after two instances of execution, surpassing the user-defined threshold.This can be observed in Figure 6b This confidence was ensured by a corresponding PST value of 64.41. Consequently, the process was terminated, and the computation was successfully completed in 0.66 seconds. Notably, the utilization of the QuEST framework resulted in a remarkable speed improvement of 34.84 times compared to the standard execution method. The termination values recorded for this circuit were as follows: N_{mi} = 1.911, Z-score = 63.77, and RMSE = 1.62.

9) Quantum Error Correction: We finally execute the QEC_n5, which is a 5 qubit utilizing quantum error correction code, with a gate depth of 18. The observed circuit does not exhibit any speedup effect. This can be attributed to the fact that throughout the execution of up to 90 instances, the circuit consistently fails to surpass the user-defined threshold of 60%, as observed in Figure 6c. Consequently, the circuit follows a regular execution path until its completion, at which point it exits. The circuit terminates, with an N_{mi} of 1.66, Z-score of 49, and RMSE of 1.32.

In Table III, we present the evaluation of our proposed QuEST framework on various benchmark circuits. Column 1 lists the benchmark circuits selected for the evaluation, while Column 2 indicates the gate depth associated with each circuit, ensuring some diversity in the range of complexity to assess the generality of our framework. Columns 3 and 4 display the execution times for the circuits using existing baseline method and QuEST, respectively. The speedup achieved by QuEST compared to the existing methods is shown in Column 5. Column 6 presents the normalized magnitude ($N_{mi}{}^{exit}$) at which the circuit reaches the desired threshold, or terminates, depending on which occurs first. Column 7 represents the confidence level of the output obtained after circuit execution.

TABLE III: Efficiency of proposed QuEST framework for expediting quantum circuit execution in NISQ hardware.

Benchmark	Gate Depth	Execution time (Baseline) (s)	Execution time (QuEST) (s)	Speedup	$\mathcal{N}_{mi}^{exit}$	Confidence^{exit} (%)	RMSEP^{exit}
half_adder	4	5.8	0.4	14.5 ×	2.04	70.11	1.77
adder_1	12	30.5	0.56	54.46 ×	2.05	70.33	1.79
adder_2	12	30.9	0.53	58.3 ×	2.035	69.93	1.772
hs4_n4	10	18.85	0.581	32.44 ×	1.85	60.47	1.556
bv_n4	6	33.7	1.8	18.72 ×	1.846	60.19	1.551
qft_n4	12	32.4	1.12	28.92 ×	1.951	65.19	1.675
vqe_n4	28	25	3.89	6.42 ×	1.87	61.57	1.58
linear_solver_n3	12	23	0.66	34.84 ×	1.911	63.77	1.628
qec_n5	18	23.5	23.5	1 ×	1.66	49	1.325

Finally, column 8 displays the Root Mean Square (RMSE) values corresponding to the obtained outputs.

The findings of our evaluation are summarized in Table III. The results indicate that the QuEST framework offers significant speed improvement compared to the baseline execution, with a maximum speedup of 58.3× and an average speedup of 27.73×. Notably, the *adder* circuits demonstrate the highest speedup, with QuEST achieving speedups of 54.45 times and 58.3 times. Overall, QuEST delivers enhanced performance in 8 out of 9 circuits evaluated. However, for the benchmark *qec_n5*, QuEST performs on par with the baseline. This can be attributed to the fact that the benchmark never reaches a confidence value exceeding the user defined threshold of 60%, presenting a unique case for the execution of our approach.

V. Conclusion

In conclusion, this paper presents QuEST, an efficient technique for estimating the output of a quantum computer (QC) by analyzing probability distributions of post-measurement data. The QuEST framework utilizes Gaussian distribution functions and a pre-trained distribution obtained from a training circuit dataset to compare the measured distribution of a circuit. Additionally, by leveraging the properties of sequential time series, the framework derives a straightforward and intuitive metric to measure the confidence of the QC output. The QuEST framework offers several advantages, including the ability to monitor fidelity evolution over time as the QC interacts with its external environment. This feature enables the system to preemptively halt QC execution upon reaching a specified confidence threshold, enhancing the reliability and accuracy of quantum computations. Evaluation against state-of-the-art benchmark quantum circuits demonstrates the effectiveness of the QuEST framework. It accurately estimates the output of 100% of the benchmark circuits while achieving significant speedup up to 58.3× compared to standard QC execution. These promising results highlight the efficiency and potential practical applications of the proposed QuEST framework. In summary, QuEST introduces a novel approach for estimating QC outputs that addresses the challenges associated with quantum computing. By leveraging probability distributions and sequential time series, the framework provides an efficient and reliable means of estimating QC outputs, opening doors to practical quantum computing applications. Further research and development in this direction could contribute significantly to the advancement and adoption of quantum computing technologies.

References

[1] M. Y. Siraichi, V. F. d. Santos, C. Collange, and F. M. Q. Pereira, "Qubit allocation," in *Proceedings of the 2018 International Symposium on Code Generation and Optimization*, 2018, pp. 113–125.

[2] P. Duan, Z.-F. Chen, Q. Zhou, W.-C. Kong, H.-F. Zhang, and G.-P. Guo, "Mitigating crosstalk-induced qubit readout error with shallow-neural-network discrimination," *Physical Review Applied*, vol. 16, no. 2, p. 024063, 2021.

[3] E. Magesan, J. M. Gambetta, and J. Emerson, "Scalable and robust randomized benchmarking of quantum processes," *Physical review letters*, vol. 106, no. 18, p. 180504, 2011.

[4] 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.

[5] T. Proctor, S. Seritan, K. Rudinger, E. Nielsen, R. Blume-Kohout, and K. Young, "Scalable randomized benchmarking of quantum computers using mirror circuits," *Physical Review Letters*, vol. 129, no. 15, p. 150502, 2022.

[6] K. Booth, M. Do, J. Beck, E. Rieffel, D. Venturelli, and J. Frank, "Comparing and integrating constraint programming and temporal planning for quantum circuit compilation," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 28, 2018, pp. 366–374.

[7] G. G. Guerreschi and J. Park, "Two-step approach to scheduling quantum circuits," *Quantum Science and Technology*, vol. 3, no. 4, p. 045003, 2018.

[8] S. S. Tannu and M. K. Qureshi, "A case for variability-aware policies for nisq-era quantum computers," *arXiv preprint arXiv:1805.10224*, 2018.

[9] A. Gilyén and A. Poremba, "Improved quantum algorithms for fidelity estimation," *arXiv preprint arXiv:2203.15993*, 2022.

[10] X.-D. Yu, J. Shang, and O. Gühne, "Statistical methods for quantum state verification and fidelity estimation," *Advanced Quantum Technologies*, vol. 5, no. 5, p. 2100126, 2022.

[11] J. Liu and H. Zhou, "Reliability modeling of nisq-era quantum computers," in *2020 IEEE international symposium on workload characterization (IISWC)*. IEEE, 2020, pp. 94–105.

[12] H. Wang, P. Liu, J. Cheng, Z. Liang, J. Gu, Z. Li, Y. Ding, W. Jiang, Y. Shi, X. Qian *et al.*, "Quest: Graph transformer for quantum circuit reliability estimation," *arXiv preprint arXiv:2210.16724*, 2022.

[13] A. Li, S. Stein, S. Krishnamoorthy, and J. Ang, "Qasmbench: A low-level qasm benchmark suite for nisq evaluation and simulation," *arXiv preprint arXiv:2005.13018*, 2020.