Dynamic Output State Classification for Quantum Computers

Hector D. Menendez
Department of Informatics
King's College London
London, United Kingdom
hector.menendez@kcl.ac.uk
ORCID: 0000-0002-6314-3725

Luciano Bello
Zurich Research Laboratory
IBM Quantum
Zurich, Switzerland
bel@zurich.ibm.com
ORCID: 0000-0002-3530-6283

David Clark
Department of Computer Science
University College London
London, United Kingdom
david.clark@ucl.ac.uk

***Abstract*—Quantum computers promise a potentially disruptive approach to improving computation in fields such as physics, chemistry, cryptography, optimisation, and machine learning. However, testing quantum computations for faults is currently impractical because of the existence of noise and errors associated with the output. Executing in a quantum system a circuit with only a few valid output states can generate a significant number of implausible states that have zero probability in an ideal computation. Among other sources of noise, readout errors come from the difficulty of discriminating a measurement between 0 and 1 for the different qubits. These issues are affected by readout drift, requiring regular recalibration of the process.**

In this paper, we provide a novel technique for post-computation analysis of the output probability distributions that permits better discrimination of kerneled data, delaying the need for recalibration. We achieve this by altering the linear discrimination of the final output states by way of a dynamic state selection process that combines Gaussian mixture models with a probability threshold. As an initial assessment of the technique we examine its effect on three to five qubits GHZ states. Our results on almost every one of nine IBM quantum computers show that the number of implausible states is reduced significantly and that the resulting probability distribution is closer to the expected one.

***Index Terms*—quantum computation, error mitigation, clustering.**

I. Introduction

This new decade is bringing remarkable steps forward in terms of computation. One of the most relevant is quantum computers, new systems that leverage the power of quantum mechanics to solve complex problems that can be encoded in a probabilistic framework. There are significant applications of quantum computers to different fields of science [3]. One such step forward is the possibility of implementing new efficient quantum algorithms that deal with classical computational problems, such as Shor's algorithm for factorisation [8], Grover's algorithm for search problems [6] and Quantum Annealing algorithms for optimisation [4].

Conventional computations are typically deterministic. Defining an oracle for such a computation requires checking the input/output behaviour However, this is not the case for quantum computations, where a single input produces a probability distribution on outputs defined by the quantum circuit. Creating testing oracles for quantum programs requires sampling the output multiple times. Quantum computers change the computing paradigm from bits to quantum bits or qubits. Data in a qubit can be thought of as a probability distribution between two possible states '0' and '1'. Every time the qubit is measured, we will get an output state and this corresponds with a classical bit. However, this state might change in each measurement, therefore, understanding the probability distribution of the qubit requires multiple measurements.

Quantum programs can be modelled as quantum circuits, which are similar to logical circuits. They describe a quantum state that defines a probabilistic computational output. Each quantum circuit is composed of a set of gates that manipulate this probability distribution. For instance, the Hadamard gate (H) puts a qubit in a superposition state where each possible output state has the same probability (i.e., creates a uniform distribution). Another non-classical example is that qubits can also be entangled in the circuit, meaning that the measurement of one defines the value of the other.

Although these quantum circuits work as expected on quantum simulators, their performance on real quantum computers is affected by noise, making the output of the circuit unreliable. This noise, even though provoked by different physical phenomena, grows significantly as a consequence of the entanglement process within the quantum computers, which is performed by cross-resonance [14] (Section VI). Current quantum computers require daily calibration to mitigate the effect of drift that is generated by the physics of the machines. During the calibration time, normally performed twice a day, the quantum computer is not operative. The calibration needs to decide several parameters, like the frequency and microwave tone amplitude values for the qubits, to distinguish states [14] and, for some experiments, it can take up to 24 hours. For example, Google's 53-qubit Sycamore can take 4 hours of calibration per day for a 2-qubit gate [9]. Reducing calibration cost is not only useful for improving the efficient use of quantum computers but also improves the possibility of testing actual quantum computations on these machines [18].

On the other hand, apart from the problem of calibration, previous work has also identified issues with the quantum output state classification [15]. Computers perform this to

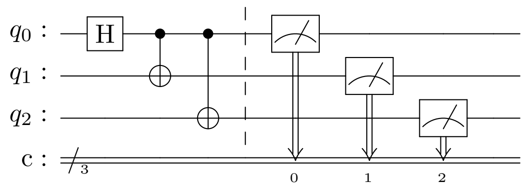

Fig. 1. Example of GHZ state

decide the qubit measurements. The output state classification process is based on the following steps: 1) Define a sample from the $|0\rangle$ state of the qubit; 2) rotate the sample π radians on x ($R_x(\pi)$) to estimate the $|1\rangle$ state; 3) Create a linear discrimination on the complex space to separate both states. This final discrimination is then used to determine the classification of the $|0\rangle$ and $|1\rangle$ states during the decision-making process of the quantum computer until recalibration (see Section II).

Our technique aims to improve output states' classification to alleviate problems that the drift produces. We propose a new methodology based on Gaussian mixture models and a probability threshold (Section III). Our initial evaluation is performed on GHZ circuits. These circuits suffer from entanglement noise and are characterised by having only 2 possible output states. Every other state, even if is implausible because it has 0 probability, tends to appear in real quantum computers as a consequence of noise (see Section II). These gates allow us to see more clearly the way these implausible states manifest in the output.

In the evaluation we have three goals: 1) reduce the implausible states (i.e., the error); 2) find a sensible threshold value for our Gaussian mixture models; and, 3) guarantee that the final probability distribution is closer to the expected one. The latter is assessed via the Kullback-Liebler divergence (Section IV). Our results show that we obtain significantly better results for the GHZ circuits. We reduce the probability of implausible states by up to 70 points. We identify a good compromise for the optimum threshold value (0.99999), although we recommend to estimate it empirically. Finally, the empirical probability distribution of the GHZ circuits is closer to the expected one after our methodology is applied, even achieving no-divergence in one of the optimized circuits.

II. Motivational Example

We discuss an example with 3 qubits and describe the discrimination problems that kerneled data can have during the readout process.

Figure 1 shows a circuit for a GHZ state, where q_0, q_1 and q_2 are entangled and there are only these two possible ideal outputs – 000 or 111, i.e all the qubits are either in the $|0\rangle$ state or $|1\rangle$ after measurement. However, the execution of such circuit on actual quantum hardware results in the measurements with some other states. For instance, Figure 2 shows the result of executing the circuit in Figure 1 on

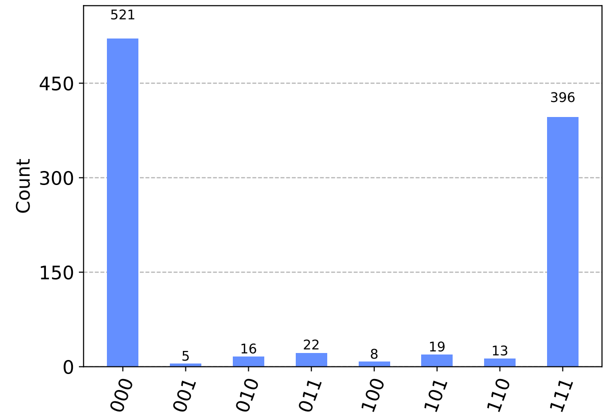

Fig. 2. Histogram of 1000 shots for the 3 qubit GHZ circuit of Figure 1 in the `ibm_nairobi` quantum computer.

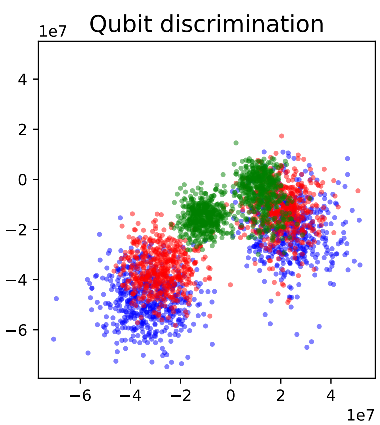

Fig. 3. Example of an IQ plane for 3 different qubits (blue, red and green). Each qubit defined two clear clusters around the computational state, $|0\rangle$ on the left and $|1\rangle$ on the right.

`ibm_nairobi`, producing several implausible other states beyond 000 and 111, therefore the resulting probability distribution is significantly different to the expected one.

Among other sources of noise, unexpected results come from the readout process: the digitized time-series signal (raw data) coming from the readout resonator is processed by a kernel method that removes its time dependency (kerneled data). Lastly, a discriminator is applied on a so-called *IQ plane* by which each qubit state is classified. [2], [7]

The goal of a discriminator acting on kerneled data is to classify each execution (shoot) and evaluate if this readout state for each qubit is a 0 or a 1. Figure 3 shows each shoot represented in the IQ plane, for each individual qubit, depending on its kerneled state. Because IQ points drift over time, the discriminator needs to be recalibrated periodically.

Considering that the decision boundary for the discriminator is static between calibrations, the drifting phenomenon can sometimes produce significant misclassifications during the

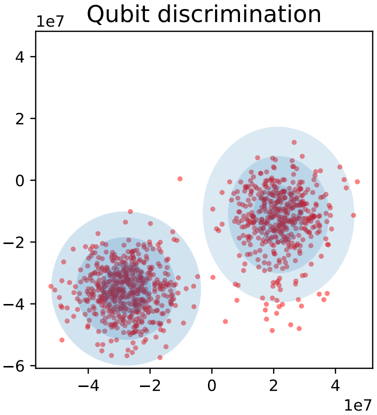

Fig. 4. Example of Gaussian mixture models applied to IQ plane of 1 qubit. The Gaussian parameters are optimized to identify the best position during the discrimination process. Also, the intensity colour for the Gaussian shows different probability levels to define ownership.

measurement, this being one of the reasons for implausible states such as the ones in Figure 2. To alleviate this problem and to reduce the effect of implausible states, we here define a new output analysis methodology, based on clustering, where we define a dynamic classifier that creates a Gaussian mixture model around the kerneled data and filters on-the-edge measurements, preserving the probability distribution of the outputs and resisting IQ point drift (see Figure 4).

Our model also allows us to define boundaries for the probabilities of belonging that detects and discards particularly noisy shots from the quantum circuit execution(Figure 4), aiming to improve the likelihood of the Gaussian distributions.

III. Dynamic Output State Classification

The output of a quantum circuit produced during quantum computation is noisy. The main problem with this noise is its entropic nature, making it unpredictable. Additionally, qubit decay can make it hard to rely on the outcome of a quantum computation. On top of this, we also have static discrimination of the final state performed on the kerneled data output by quantum computers (as explained in Section II. All of these factors make testing quantum computers extremely challenging.

Testing quantum computation requires reliability in the computer's output. To obtain a useful testing oracle for the input/output, we need to guarantee that the output probability distribution is not affected by the qubit decay factor, or at least alleviate this effect. Thus, our approach aims to clean the output post computation by applying a dynamic discrimination mechanism to the kerneled data of the output's quantum states. The proposed methodology is divided into three steps: 1) Obtain the kerneled data from the measurment process of the quantum computation, 2) Establish the energy centres by applying Gaussian mixture models, 3) Improve the discrimination between quantum states by using probability thresholds on the outputs.

A. Overview

Assume P is a quantum circuit or program. We submit the program to a quantum computer that will compute it and it will generate N outputs, where N is the number of measurements or shots (X) for the circuit. Instead of using the final classification of the outputs we read the IQ point for each shot, therefore every $x \in X$ satisfies $x \in \mathbb{C}^Q$, because the IQ points are represented by complex numbers, and Q represents the number of qubits used in P.

Considering that all measurements are obtained at the same time, we consider the optimization space X the set of outputs produced from measurements as energy levels (see Algorithm 1, line 1), after the execution of P. Current quantum computers have a static classifier, therefore we replace the classifier, which is static, with two Gaussian distributions that are calculated per qubit. In the algorithm, we can see that for every qubit q we create a projection $\mathcal{N}^q_{|0\rangle}$ for state $|0\rangle$, and the equivalent for state $|1\rangle$. The parameters of these distributions (μ, Σ) are optimised following the expectation maximisation algorithm (see Section III-B). Algorithm 1 calculates the distributions in line 3. Considering the likelihood, we can calculate the probability of an element belonging to the specific probabilities distributions (see Section III-B), as we do in line 5. If the probability is smaller than the provided threshold T for any of the qubits, we consider that measurement noise and discard it, otherwise, we assign the state according to the closest Gaussian. Once all the output values o^q_i have been assigned, we compose the final output O as a probability distribution of all of the quantum states.

Algorithm 1 Dynamic Optimization Algorithm

Require: $N \geq 0$ number of measurements
Require: P Quantum Program
Require: T probability threshold
Ensure: O the probability distribution of outputs
1: $X = [P(N)]_{energy}$
2: **for** $q \in Q$ **do**
3: $\quad \{\mathcal{N}^q_{|0\rangle}, \mathcal{N}^q_{|1\rangle}\} = \text{GMM}(X_q)$
4: $\quad$ **for** $x^q_i \in X^q$ **do**
5: $\quad\quad$ **if** $Prob(x^q_i \in \mathcal{N}^q_{|0\rangle}) < T \;\; \& \;\; Prob(x^q_i \in \mathcal{N}^q_{|1\rangle}) < T$ **then**
6: $\quad\quad\quad$ Remove x_i
7: $\quad\quad$ **else**
8: $\quad\quad\quad o^q_i = \arg\max_j Prob(x^q_i \in \mathcal{N}^q_{|j\rangle})$
9: $\quad\quad$ **end if**
10: $\quad$ **end for**
11: **end for**

B. Expectation Maximisation on Gaussian Mixture Models

Our proposed approach starts with the expectation maximization of a Gaussian mixture model. This methodology is frequently applied in clustering problems. It defines some Gaussian distributions and optimises their parameters, i.e., a

Gaussian mixture model (GMM) with N normal distributions is defined as a set:

$$\text{GMM} = \{\mathcal{N}_1(\mu_1, \Sigma_1), \ldots, \mathcal{N}_N(\mu_N, \Sigma_N)\} \tag{1}$$

The expectation maximisation process provides an optimisation for the parameters of the GMM based on their likelihood with respect to the sampled data, defined as $X = \{x_1, \ldots, x_M\}$. The sample data in this case are the kerneled data from shots of the output from the quantum computers' executions (as shown in Figure 3). The process performs the following two steps:

- **Expectation step**: It fixes a model θ (i.e., it sets values for the parameters μ_i and Σ_i for $i \in [1, N]$, and estimates for each $x_j \in X$, which $\mathcal{N}_i$ is the closest to the point. This assigns a label i to each point x_j.
- **Maximization step**: Once the labels are assigned, this step recalculates the parameters μ_i and Σ_i with respect to the points they have been assigned, to define a new model θ. This step is performed using the likelihood function $L(\theta)$, defined as:

$$L(\theta) = p(X, y|\theta) = \prod_j p(x_j, y_j|\theta) \tag{2}$$

The algorithm starts with an initial model $\theta^{(0)}$ and it creates a sequence of models $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(t)}, \ldots$, satisfying that the likelihood of each model $L(\theta^{(t)}) \geq L(\theta^{(t-1)})$.

C. Selecting the Probability Threshold based on Likelihood

The GMM is based on the mean value, therefore the individual normal distributions will be affected by noise and some of the values may be outliers with respect to the model θ. Our methodology defines a probability threshold that we will study in Section V-B. For each qubit, our methodology discards every measurement or shot whose probability to belong to any of the states defined by the Gaussian models is smaller than the threshold. Therefore, we need to calculate the probability for each point to belong to the Gaussian models.

During the expectation step, each point is assigned to each Gaussian $\mathcal{N}_i$ according to Bayes' theorem. We use the concept of responsibility r_{ij} which is the probability for the point x_j to belong to $\mathcal{N}_i$, and it is defined as:

$$r_{ji} = p(y_j = k|x_j, \pi_k, \mu_k, \Sigma_k) = \frac{\pi_k \mathcal{N}(x_j|\mu_k, \Sigma_k)}{\sum_{i=1}^{N} \pi_i \mathcal{N}(x_j|\mu_i, \Sigma_i)}, \tag{3}$$

where y_j is the final classification of sample x_j, μ_k and σ_k are the parameters of the Gaussian $\mathcal{N}_k$ and π_k is the weight of the Gaussian model k inside of the Gaussian mixture models learned.

Once we have defined the Gaussian mixture models, we calculate the responsibility for every measurement x_j and, for each qubit $q \in Q$, each of the two distributions representing the states $\mathcal{N}^q_{|0\rangle}$ and $\mathcal{N}^q_{|1\rangle}$. If the maximum responsibility, with respect to the Gaussian mixture model, for any of the shots x_j at any of the qubits q is smaller than the threshold T, that shot will be discarded as noise. The rest will define the output probability distribution O, as stated in Algorithm 1.

IV. Experimental Setup

Our experiments aim to measure how well the methodology reduces the error generated by noise within quantum computers. This should help alleviate the need for calibration by making the computers' outputs reliable for longer, weakening the effect of the qubit decay process. For the experiments, we used 9 different quantum computers from IBM, concretely: `ibm_lagos`, `ibm_nairobi`, `ibm_oslo`, `ibm_perth`, `ibmq_belem`, `ibmq_jakarta`, `ibmq_lima`, `ibmq_manila`, `ibmq_quito`. To evaluate the capabilities of the methodology, we measure (1) how well we reduce the influence of implausible states and (2) how close we are to the expected distribution. Measuring the first goal requires us to use circuits whose implausible states are easy to detect, therefore we apply our experimentation to GHZ circuits as shown in Figure 1. To guide our evaluation experiments, we answer the following research questions:

RQ1. Can the GMM methodology reduce the probability of implausible states output by the GHZ circuits? To answer this question we sequentially run 20 executions of the same GHZ circuit on each machine and obtain their kerneled data. We create a linear classification method based on a Voronoi tessellation to provide the output energies classification that assigns each IQ point to a computation state. Then, we apply the GMM approach to these point, initially without using the threshold, then subsequently with the threshold. We apply this process for different numbers of qubits, from three to five (depending on the computers' limitations), to every IBM computer.

RQ2. What is the effect of the probability threshold in the provided methodology?

We manipulate the probability threshold to reduce the number of implausible states and study how much these can be reduced depending on the threshold. The threshold starts from 0 to 0.9 and then it is increased to 0.9 for each order of magnitude to make it closer to 1. This evaluation is performed on top of the first set of experiments.

RQ3. How close is the obtained probability to the expected CNOT probability distribution?

Here, we use the notion of Kullback-Leibler (KL) divergence to understand how far the final probability distribution diverges from the expected one. This divergence is calculated with the formula:

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \tag{4}$$

where P is the obtained probability distribution and Q is the expected one. When $Q(x)$ is 0, the value is not considered in the summation. This evaluation is performed on top of the first set of experiments.

A. Implementation

Every GHZ circuit is implemented by applying a Hadamard gate in the first qubit and entanglement for the others, therefore, the expected probability distribution is 50% probability for the states where every qubit is equal (e.g, for a 3 qubit

GHZ circuit, 50% for each of the states '000' and '111'). All the other states have 0% probability and they are therefore implausible states. We implemented every circuit using Qiskit [1].

Our implementation of the linear separation is performed by using a K-means [12] algorithm with 2 clusters, whose implementation is extracted from Sklearn[2]. This generates a linear Voronoi tessellation and provides an optimum separation following the same principles of the static separation normally performed by the quantum computers, which is not retrievable once the kerneled data is obtained. Our implementation of the GMM models is based on the Expectation-Maximization [13] implementation from Sklearn[3].

V. Results

The evaluation focuses on three main aspects. The first aspect is related to the ability to reduce the percentage of shots or measures that lead to implausible states (Section V-A), by using the general methodology for discrimination, namely the initial GMM and then the GMMs with the threshold. Considering that the final discrimination criterion depends on the threshold, we study its effect after (Section V-B). Finally, success for the methodology would be to guarantee that the probability distribution on the output states is closer to the expected distribution compared to that obtained without using our technique. Hence we compare KL divergences from the expected distribution for three distributions: produced by the static threshold, produced by the initial application of GMM, and produced by the application of GMM with an optimal threshold.

A. Reducing Implausible States

We evaluate the ability of the GMM methodology to reduce the probability of implausible states. For that we consider the GHZ circuit using 3, 4 and 5 qubits. This is computed on each of the IBM computers as each has its own physical limitations. We had neither control over the calibration time nor the final classification for each quantum computer, therefore we applied an optimised linear estimator (based on the K-means algorithm) that simulates the existing one in the quantum computer after the calibration.

Table I shows the results for the 9 computers, for the different scenarios where Nq represent the linear separation, Nq^{*} represents the Gaussian model and Nq^{**} represents the Gaussian model with the probability threshold. We can see in almost all of the cases that the methodology with a lower probability of generating implausible states is the GMM with threshold.

Comparing the usual state discrimination with the GMM there is a small improvement for the 3 qubits circuits, where the highest improvement reduces the error up to 1.2 points. These results are similar for 4 qubits where, in some cases, the GMM model is worse than the normal discrimination. However, the results are different in the 5 qubit cases. For 5 qubits, we can see three cases showing major improvements over the baseline, and they are `ibm_nairobi` (from 61.9% to 34.9% of implausible states), `ibm_jakarta` (from 95% to 24.7%) and `ibm_lima` (from 93.8% to 37.0%).

When we compare the original results with the GMM+threshold optimiser (applying a probability threshold of 0.99999), we can see stronger improvements in almost every single scenario. For 3 qubits, we reduce the implausible states probability up to 5.6 points (in the case of `ibm_quito`), for 4 qubits, these improvements go up to 7 points (in the case of `ibm_jakarta`) and for 5 qubits, it goes up to 80.7 points (in the case of `ibm_jakarta`). It is important to remark that in the case of `ibm_oslo` there is no improvement but the results are very similar to the original ones. Our methodology has a non-negative empirical impact on the results in every case.

These results strongly suggest that our methodology can help alleviate the noise generated during the measurement process and the GHZ circuit, and reduce the frequency of calibration. Considering that our methodology is reducing shots, we also provided a measure of effort reduction that is in Table II. The table shows the number of samples that pass the threshold after it is applied. We can see that the effect significantly varies among different computers and that the filter is more aggressive as the number of qubits increases. In some cases, such as `ibmq_quito` and `ibm_nairobi`, the threshold leaves only 1% and 10% respectively of the measurements, wastes several shots. In these cases, even if the output is reliable it is worth recalibrating the computer, considering that a significant number of shots are noisy.

RQ1: *The GMM methodology reduces implausible states making the final state classification of the quantum circuit more reliable. This reduction is improved by a threshold value of 0.99999 in the probability of assigning points to the Gaussians. The improvements are more significant when the number of qubits increases. They reach an 80.7 points reduction in implausible states on average for 5 qubits.*

B. The Probability Threshold

Our methodology reduces the probability of implausible states based on a Gaussian mixture model combined with a probability threshold. Even though the methodology obtains better results by reducing implausible states, there is still a threshold parameter associated with the methodology which nature directly affects the results. This parameter is the probability of a shot belonging to a specific Gaussian.

To understand the effect of the parameter, we show in Figure 5 some different values for the probability threshold and the resulting probability of implausible states. These plots start with no threshold (or a threshold of 0) which is similar to the case of using GMMs alone and go up to 0.999999. This logarithmic plot shows that the threshold starts reaching improvements after 0.5 probability and it reaches a minimal

[1] https://qiskit.org/

[2] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[3] https://scikit-learn.org/stable/modules/mixture.html

Computer	3q	3q*	3q**	4q	4q*	4q**	5q	5q*	5q**
`ibm_lagos`	4.1 ± 20.8	4.1 ± 20.8	**3.9 ± 20.9**	9.20 ± 39.6	8.4 ± 27.5	**7.5 ± 27.8**	20.5 ± 27.4	20.4 ± 27.4	**20.1 ± 27.8**
`ibm_nairobi`	8.9 ± 39.3	8.8 ± 35.9	**3.3 ± 39.2**	15.5 ± 25.4	15.4 ± 18.5	**9.7 ± 19.3**	61.9 ± 34.0	34.9 ± 33.7	**27.3 ± 38.7**
`ibm_oslo`	5.6 ± 28.6	5.6 ± 28.6	**5.3 ± 28.8**	25.5 ± 28.9	25.3 ± 23.4	**23.5 ± 23.9**	**82.2 ± 8.9**	82.6 ± 9.5	82.6 ± 9.6
`ibm_perth`	7.0 ± 36.9	6.8 ± 32.9	**6.0 ± 33.0**	13.9 ± 40.5	14.0 ± 40.5	**12.1 ± 40.9**	29.7 ± 31.4	29.8 ± 29.4	**28.9 ± 29.8**
`ibmq_belem`	12.2 ± 41.0	12.1 ± 41.0	**8.3 ± 43.5**	17.7 ± 38.0	17.7 ± 38.0	**12.9 ± 41.9**	37.6 ± 29.2	38.2 ± 28.1	**35.1 ± 31.9**
`ibmq_jakarta`	10.6 ± 41.4	9.7 ± 32.3	**5.1 ± 35.0**	14.9 ± 33.0	14.6 ± 18.0	**7.9 ± 21.3**	95.0 ± 37.9	24.7 ± 35.0	**14.3 ± 42.9**
`ibmq_lima`	7.7 ± 0.8	7.9 ± 0.8	**4.0 ± 1.5**	13.8 ± 34.3	13.8 ± 34.2	**9.8 ± 37.5**	93.8 ± 32.7	37.0 ± 30.3	**31.3 ± 34.8**
`ibmq_manila`	8.0 ± 36.3	7.9 ± 36.3	**4.9 ± 38.4**	21.1 ± 37.4	21.5 ± 37.2	**14.9 ± 41.4**	31.2 ± 33.9	30.1 ± 31.0	**25.7 ± 33.5**
`ibmq_quito`	15.6 ± 38.9	14.4 ± 30.5	**10.0 ± 36.8**	-	-	-	-	-	-

TABLE I

THE PERCENTAGE OF IMPLAUSIBLE STATES (MEDIAN AND STANDARD DEVIATION) IN THE 20 RUNS FOR EACH CIRCUIT IN EACH QUANTUM COMPUTER BY APPLYING THE LINEAR DISCRIMINATION (3Q, 4Q, AND 5Q), THE GMM MODEL (3Q*, 4Q* AND 5Q*) AND THE MODEL WITH THE THRESHOLD (3Q**, 4Q**, 5Q**).

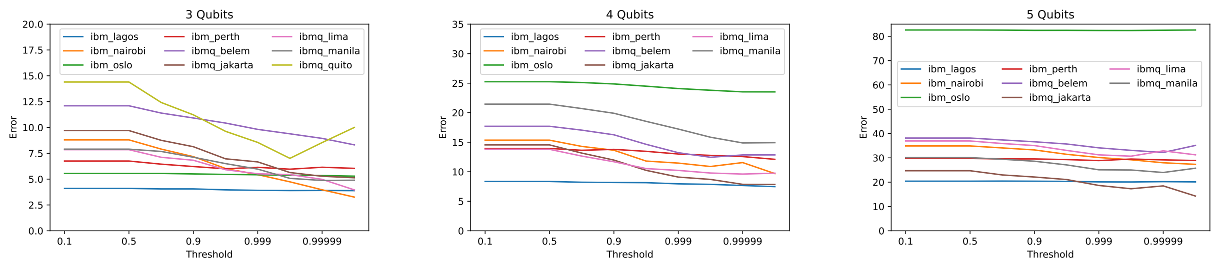

Fig. 5. Error rate under different threshold values for 3, 4 and 5 qubits.

Computer	3q	4q	5q
`ibm_lagos`	950.0 ± 16.9	927.0 ± 15.7	787.5 ± 26.4
`ibm_nairobi`	97.5 ± 20.8	58.5 ± 17.1	41.0 ± 17.0
`ibm_oslo`	978.0 ± 7.1	942.5 ± 13.3	884.5 ± 38.0
`ibm_perth`	375.0 ± 54.8	358.0 ± 37.7	321.0 ± 37.2
`ibmq_belem`	164.5 ± 32.8	112.5 ± 26.4	99.0 ± 29.7
`ibmq_jakarta`	129.0 ± 27.9	24.5 ± 9.3	7.0 ± 4.6
`ibmq_lima`	237.5 ± 43.0	39.5 ± 12.6	16.0 ± 8.5
`ibmq_manila`	356.5 ± 45.8	103.5 ± 19.6	73.0 ± 20.4
`ibmq_quito`	10.0 ± 5.9	-	-

TABLE II

THE NUMBER OF SAMPLES (OUT OF THE 1,000 MEASUREMENTS OR SHOTS) USED AFTER THE THRESHOLD IS APPLIED. THE LEFT SIDE OF EVERY VALUE SHOWS THE MEDIAN WHILE THE RIGHT SIDE SHOWS THE STANDARD DEVIATION.

error rate, in general, between 0.999 and 0.99999. This tendency is the same for 3, 4 and 5 qubits. However, we can appreciate that there are some cases, when the threshold is more restrictive, where the probability of implausible states increases. For example, in the case of `ibm_quito` for 3 qubits. This suggests that the threshold needs to be optimised for each specific case, which is a problem that we will explore in future work. However, setting the threshold to **0.99999** is a good compromise in the GHZ scenario.

RQ2: *The threshold value tends to improve the reduction of implausible states as it becomes more restrictive, but there is a limitation. If the optimal value is surpassed, the percentage of implausible states increases again. This optimal value needs to be discovered but 0.99999 is a good compromise.*

C. The Probability Distribution

The last step of our evaluation is to compare the divergences from the expected probability distribution for the distributions that result from our methodology and the usual static threshold approach. Table III shows this comparison between the expected probability distribution and the obtained one after the sampling by using the Kullback-Liebler divergence.

Table III shows similar positive results concerning the probability distribution for the different number of qubits. For 3 qubits, we can see that the results with and without the GMM model are very similar. The maximum improvement obtained with the new methodology is with the `ibmq_quito` machine, where the KL distance is reduced by one point. However, when we apply the probability threshold, the results are significantly better. The improvements are between 0.2 points (in the case of `ibm_lagos` and `ibm_oslo`) to up to almost 10 points in the case of `ibmq_quito`).

For 4 qubits, the results obtained by applying the GMM are similar, and usually achieve a small improvement. While for 5 qubits, the methodology generates worse results in terms of divergence for a significant number of cases. The most representative cases are `ibmq_belem` (1.1 points worse), `ibmq_jakarta` (3.6 points worse), `ibmq_lima` (6.8 points worse), and `ibmq_manila` (1.8 points worse). When we include the threshold, the results improve significantly for almost all of the cases (and better than the baseline for all of them). Some of the most relevant cases for 4 qubits are `ibmq_jakarta` (where the divergence value is 0.1, very close to the original distribution, from an original value of 13.1), `ibmq_lima` (where the divergence goes from 12.1 to 0.8) and `ibmq_belem` (where the divergence goes from 13.3 to 4.9). In the 5 qubits cases, results are similar, having

Computer	3q	3q*	3q**	4q	4q*	4q**	5q	5q*	5q**
`ibm_lagos`	3.8 ± 1.6	3.8 ± 1.6	**3.6 ± 1.6**	8.5 ± 2.5	7.9 ± 1.5	**7.0 ± 1.6**	17.3 ± 3.2	17.3 ± 3.3	**16.8 ± 3.2**
`ibm_nairobi`	8.1 ± 1.1	8.1 ± 1.6	**1.3 ± 3.8**	13.4 ± 2.0	13.4 ± 1.8	**7.4 ± 8.9**	19.8 ± 7.0	20.6 ± 6.2	**9.1 ± 9.6**
`ibm_oslo`	5.3 ± 0.9	5.3 ± 0.9	**5.1 ± 0.9**	20.2 ± 4.4	20.3 ± 3.3	**19.1 ± 3.2**	21.5 ± 7.0	**21.1 ± 7.9**	21.4 ± 8.2
`ibm_perth`	6.4 ± 2.5	6.3 ± 2.3	**5.2 ± 3.4**	9.6 ± 3.5	9.5 ± 3.5	**9.0 ± 3.8**	22.4 ± 5.8	22.5 ± 5.1	**21.5 ± 4.8**
`ibmq_belem`	10.8 ± 2.0	10.9 ± 1.8	**7.1 ± 4.4**	13.3 ± 3.5	13.5 ± 3.6	**4.9 ± 5.8**	25.3 ± 6.9	26.4 ± 5.6	**20.9 ± 7.0**
`ibmq_jakarta`	8.6 ± 1.4	8.6 ± 1.2	**2.2 ± 3.2**	13.1 ± 1.5	13.2 ± 1.2	**0.1 ± 9.2**	14.4 ± 7.4	18.0 ± 6.2	**0.0 ± 27.6**
`ibmq_lima`	7.2 ± 0.8	7.2 ± 0.8	**2.0 ± 2.9**	12.1 ± 2.8	12.2 ± 2.8	**0.8 ± 11.4**	17.3 ± 9.1	24.1 ± 8.0	**11.4 ± 15.8**
`ibmq_manila`	7.5 ± 2.2	7.4 ± 2.2	**4.3 ± 2.5**	15.9 ± 3.8	15.8 ± 3.8	**9.5 ± 5.6**	19.8 ± 6.9	21.6 ± 6.6	**14.7 ± 5.8**
`ibmq_quito`	14.0 ± 2.6	13.0 ± 2.0	**3.16 ± 16.3**	-	-	-	-	-	-

TABLE III
THE KULLBACK-LEIBLER DIVERGENCES BETWEEN OBTAINED PROBABILITY DISTRIBUTIONS AND THE EXPECTED ONE OVER 20 RUNS. FOR EACH COLUMN, THE LEFT VALUE SHOWS THE MEDIAN AND THE RIGHT ONE SHOWS THE STANDARD DEVIATION. THE THREE COLUMNS FOR EACH NUMBER OF CUBITS ARE AS IN TABLE TABLE I. FOR VISUALISATION PURPOSES VALUES ARE SCALED UP BY 2 ORDERS OF MAGNITUDE.

special relevance in the case of `ibmq_jakarta` where the divergence goes from 14.4 to 0.

RQ3: *The GMM methodology alone has no significant impact on improving the output probability distribution and making it closer to the expected one, according to the Kullback-Liebler divergence. However, combined with the threshold, it obtains improvements in every case, reaching the point of almost up to no divergence in some cases.*

D. Discussion and limitations

Our methodology for making the final output state discrimination is effective. It improves the results and corrects errors provoked by noise. However, this methodology may be expensive in cases where the number of necessary measurements grows by orders of magnitude. This is conceivable for computers using hundreds of qubits and complex probability distributions.

In our experiments, we have resorted to the use of the GHZ circuit, the probability distribution of which is well-known and easy to model in terms of quality metrics, but we have yet to understand how effective our methodology can be when applied to different circuits. In addition to measuring the effect that different types of gates can have on the results, it is interesting to check the history of the results of the new methodology with respect to the calibration times, above all to evaluate whether the Gaussian models can improve, lengthening, the recalibration times.

The methodology itself strongly depends on the number of shots or measurements considered, as happens in the case of any distribution of probabilities generated by quantum circuits. This is especially relevant under very noisy conditions, where only a few measurements remain (Table II). Therefore, apart from optimizing the threshold, it might be important to keep as many datasets as possible, so as to reduce associated effort.

This methodology and similar ones can potentially lead to reducing the calibration frequency and improving the reliability of quantum computers.

VI. RELATED WORK

Quantum computers with superconducting Transmon qubits, based on Josephson Junction, are sensitive to noise. In this architecture, qubits are entangled with coupling resonators, which produces significant noise for gates with 2 or more qubits [14]. Also, the readout operators for the resonators increase the error rate, apart from other factors that affect the general physics of the computers and can hardly be modelled. Besides, qubits can only maintain their states or coherence for a limited period of time, which is the reason we are forced to regularly recalibrate. The length of this period is not currently clear, but it is associated with the decay of the qubit.

The problem of reducing the effect of noise in quantum computers is not new and covers different perspectives that go from calibration to error correction and variational quantum computers. Resch and Karpuzcu focused on understanding the problem of benchmarking quantum computers with regard to their performance [16]. The authors acknowledge the significant impact that noise has on quantum circuits and the difficulties it has in modeling them, as is widely recognised in the literature. After testing a significant amount of classical methodologies, the authors conclude that computational models applied to noise require a significant amount of detail to be accurate and must be adapted to specific circumstances.

Identifying the correlations between the various physical phenomena revolving around (quantum) computers, with their specific effects, is important to understand how to model appropriate filters to mitigate the resulting noise in qubits [1]. Understating noise allows the development of packages like Mitiq [10], which reduces the effect of noise by combining different postprocessing and noise cancellation techniques. The main contributions of this tool are noise scaling through unitary folding [5], Clifford data regression [11] and probabilistic error cancellation [17], all of the techniques limited by their abilities to generalize to multiple different circuits and gates, considering that they depend on the sample size or the data used. This is also a limitation of the technique that we introduce here especially considering that understanding the probability distribution of complex circuits requires an unknown number of shots or measurements. Given that all of these techniques work directly on the output states, our approac is complementary to them rather than an alternative.

VII. CONCLUSIONS AND FUTURE WORK

A crucial step to delivering the promise of reliable quantum computers is to understand their physics and how their outputs can be made more reliable. This paper has shown how analysis and manipulation of the output state discrimination process can significantly improve the readout results, reducing the need for calibration and making the computers more accurate. However, there are still other techniques that need to be explored in this area to make this discrimination process more reliable in the future and reduce the number of shots or measurements that need to be discarded.

Our future work will focus on three fundamental problems that we have identified in our methodology: 1) Find an optimisation method that can provide a good threshold value, 2) experiment with other techniques that can discriminate among the kerneled data, and 3) understand how the drift factor affects the output and identify patterns among them.

ACKNOWLEDGMENTS

The support of the UKRI Trustworthy Autonomous Systems Hub (reference EP/V00784X/1) and Trustworthy Autonomous Systems Node in Verifiability (reference EP/V026801/2) is gratefully acknowledged. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team.

REFERENCES

[1] D. Aharonov, A. Kitaev, and J. Preskill. Fault-tolerant quantum computation with long-range correlated noise. *Physical review letters*, 96(5):050504, 2006.

[2] T. Alexander, N. Kanazawa, D. J. Egger, L. Capelluto, C. J. Wood, A. Javadi-Abhari, and D. C. McKay. Qiskit pulse: programming quantum computers through the cloud with pulses. *Quantum Science and Technology*, 5(4):044006, 2020.

[3] C.-H. Cho, C.-Y. Chen, K.-C. Chen, T.-W. Huang, M.-C. Hsu, N.-P. Cao, B. Zeng, S.-G. Tan, and C.-R. Chang. Quantum computation: algorithms and applications. *Chinese Journal of Physics*, 72:248–269, 2021.

[4] A. Das and B. K. Chakrabarti. Colloquium: Quantum annealing and analog quantum computation. *Reviews of Modern Physics*, 80(3):1061, 2008.

[5] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, and W. J. Zeng. Digital zero noise extrapolation for quantum error mitigation. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 306–316. IEEE, 2020.

[6] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt. Applying grover's algorithm to aes: quantum resource estimates. In *Post-Quantum Cryptography*, pages 29–43. Springer, 2016.

[7] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 6(2):021318, jun 2019.

[8] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. James, A. Gilchrist, and A. G. White. Experimental demonstration of a compiled version of shor's algorithm with quantum entanglement. *Physical Review Letters*, 99(25):250505, 2007.

[9] L. Lao, P. Murali, M. Martonosi, and D. Browne. Designing calibration and expressivity-efficient instruction sets for quantum computing. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, pages 846–859. IEEE, 2021.

[10] R. LaRose, A. Mari, S. Kaiser, P. J. Karalekas, A. A. Alves, P. Czarnik, M. El Mandouh, M. H. Gordon, Y. Hindy, A. Robertson, et al. Mitiq: A software package for error mitigation on noisy quantum computers. *Quantum*, 6:774, 2022.

[11] A. Lowe, M. H. Gordon, P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio. Unified approach to data-driven quantum error mitigation. *Physical Review Research*, 3(3):033098, 2021.

[12] J. MacQueen. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297, 1967.

[13] T. K. Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[14] T. Patel, B. Li, R. B. Roy, and D. Tiwari. {UREQA}: Leveraging {Operation-Aware} error rates for effective quantum circuit mapping on {NISQ-Era} quantum computers. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, pages 705–711, 2020.

[15] T. Patel and D. Tiwari. Disq: a novel quantum output state classification method on ibm quantum computers using openpulse. In *Proceedings of the 39th International Conference on Computer-Aided Design*, pages 1–9, 2020.

[16] S. Resch and U. R. Karpuzcu. Benchmarking quantum computers and the impact of quantum noise. *ACM Computing Surveys (CSUR)*, 54(7):1–35, 2021.

[17] R. Takagi, S. Endo, S. Minagawa, and M. Gu. Fundamental limits of quantum error mitigation. *npj Quantum Information*, 8(1):1–11, 2022.

[18] C. Tornow, N. Kanazawa, W. E. Shanks, and D. J. Egger. Minimum quantum run-time characterization and calibration via restless measurements with dynamic repetition rates. *Physical Review Applied*, 17(6):064061, 2022.