# Sampling Strategy Optimization for Randomized Benchmarking

Toshinari Itoko[1,a)] Rudy Raymond[1,b)]

**Abstract:** Randomized benchmarking (RB) is a widely used method for estimating the average fidelity of gates implemented on a quantum computing device. The stochastic error of the average gate fidelity estimated by RB depends on the sampling strategy (i.e., how to sample sequences to be run in the protocol). The sampling strategy is determined by a set of configurable parameters (an RB configuration) that includes Clifford lengths (a list of the number of independent Clifford gates in a sequence) and the number of sequences for each Clifford length. The RB configuration is often chosen heuristically and there has been little research on its best configuration. Therefore, we propose a method for fully optimizing an RB configuration so that the confidence interval of the estimated fidelity is minimized while not increasing the total execution time of sequences. By experiments on real devices, we demonstrate the efficacy of the optimization method against heuristic selection in reducing the variance of the estimated fidelity.

**Keywords:** randomized benchmarking, sampling error, mathematical optimization, weighted least squares

## 1. Introduction

As more and larger quantum computers without fault tolerance are physically implemented, there is a growing need for methods to benchmark their performance. Straightforward tomography-based methods require measurements that scale exponentially with the number of qubits [1] and thus are not scalable. Utilizing advanced techniques, such as, compressed sensing [2] and samplings [3, 4], the scaling bottleneck can be avoided to obtain the average gate fidelity but such methods are not robust against state preparation and measurement (SPAM) errors. Randomized benchmarking (RB) [5, 6] is an efficient and robust method and widely used in practice for estimating the average fidelity of a gate set implemented on a quantum computing device [7–12]. For example, IBM Quantum systems [13] report their 1-qubit and 2-qubit gate error rates calculated from the estimated average gate fidelity via RB. Hence it is important to minimize the stochastic error in the estimated fidelity by RB so that we can sufficiently track the drift in the gate fidelity over time, which reflects imperfection of controlling physical devices.

The standard RB is a protocol composed of three steps. First, it generates sets of sequences with random Clifford gates such that all sequences in each set have the same Clifford length (the number of independent Clifford gates in a sequence) but the length varies from set to set. Then, it executes the sequences to measure the survival rate (i.e., probability of observing the initial state) for each Clifford length. Finally, it estimates the exponential decay rate, which can be linearly transformed to the average gate fidelity, from the survival rate data. The protocol has a three-fold sampling structure: sampling Clifford lengths at the estimation (fitting) step, sampling random sequences at the sequences generation step, and sampling bit strings at the sequences execution step. Therefore, the estimated decay rate (or equivalently, the average gate fidelity) is intrinsically subject to stochastic errors that depend on the sampling strategy. The sampling strategy is determined by a set of configurable parameters that defines how to sample sequences to be run (we call it an RB configuration), e.g., Clifford lengths and the number of sequences for each Clifford length.

There are several studies that partially address the problem of finding an optimal RB configuration. The number of sequences at each Clifford length that achieves a desired confidence level was loosely estimated by using Hoeffding bound [14]. A tighter estimation comparable to the number used in practice was provided, assuming to use the ordinary least squares estimator in the fitting step [15]. It was suggested that varying the number of sequences depending on Clifford length may improve the reliability of estimated decay rate [16]. Finding the best maximum Clifford length was also discussed in [17]. However, none of them addressed the problem of both optimizing Clifford lengths and the number of sequences at the same time.

In this paper, we provide a method for finding an optimal sampling strategy (RB configuration) that includes both Clifford lengths and the number of sequences for each Clifford length. The optimal strategy yields a minimal confidence interval of the estimated average gate fidelity within a given

[1] IBM Quantum, IBM Research-Tokyo
19-21, Nihonbashi, Hakozaki-cho, Chuo-ku, Tokyo 103–8501
[a)] itoko@jp.ibm.com
[b)] rudyhar@jp.ibm.com

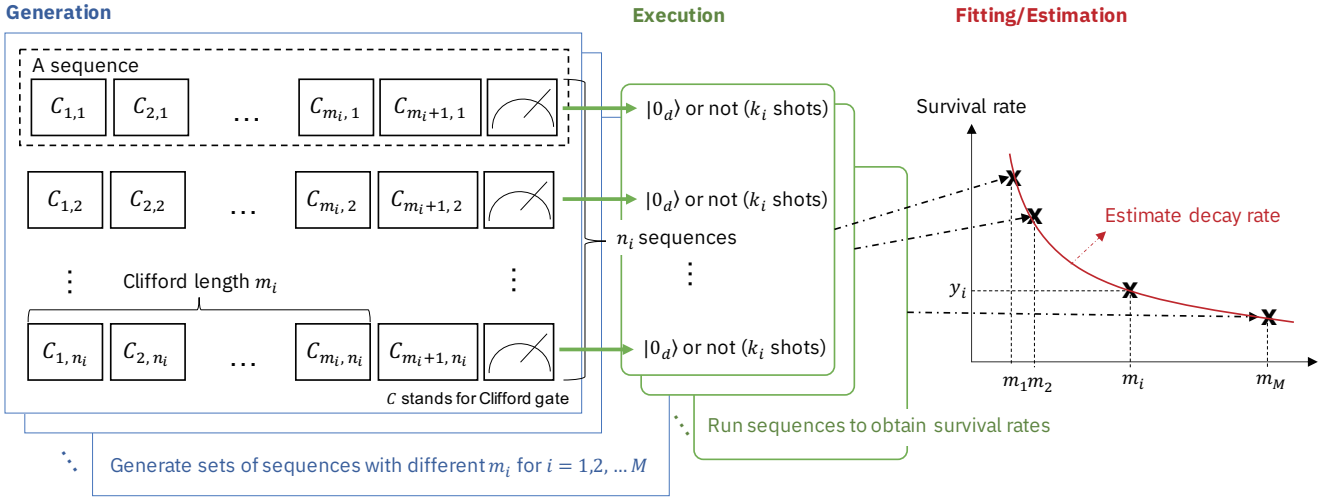



Fig. 1: The standard RB (randomized benchmarking) protocol

time budget for running the sequences (Section 3). To make it possible, we customize techniques for analyzing statistical errors of estimated values in RB. First, we construct a simple model for the variance of average survival rate (Section 3.2) in order to explain the variance data sampled from real devices, which may vary depending on Clifford length. Based on the variance model, we derive an explicit expression that approximates the confidence interval of the estimated decay rate as a function of the RB configuration (Section 3.3). The derivation enables us to formulate the optimization of the RB configuration as a mathematical optimization problem (Section 3.5). We also experiment on real devices to show that our method can find a better configuration that achieves smaller variance in the resulting estimated decay rate than typical heuristic configurations in practice (Section 4).

## 2. Preliminaries

We first set some notation and briefly review the standard RB protocol. We suppose to benchmark a $d$-qubit quantum system, which can be represented by the $D = 2^d$ dimensional Hilbert space. We assume the initial state is always set to $|0_d\rangle$ and the measurement is a projection onto the computational basis, i.e. $|0\rangle$ or $|1\rangle$ for each qubit.

The standard RB protocol [6, 14] is described as follows.

**Step 1** (Generation) Create $n_i$ sequences, each of which is a sequence of $m_i + 1$ Clifford gates followed by a measurement as shown in Fig. 1. The first $m_i$ gates are chosen uniformly at random from the $d$-qubit Clifford group and the $(m_i + 1)$-th gate is uniquely determined as the inverse of the composition of the first $m_i$ gates.

**Step 2** (Execution) Run the $n_i$ sequences, measure the survival rate (i.e., the number of times $|0_d\rangle$ is observed divided by $k_i$ trials) for each sequence, and average over the $n_i$ survival rates to obtain the average survival rate $y_i$.

**Step 3** (Fitting/Estimation) Repeat Step 1 and 2 for different Clifford lengths $[m_1, m_2, \ldots, m_M]$ and then fit the results $[y_1, y_2, \ldots, y_M]$ to the decay model (e.g. $y_i \sim a\, p^{m_i} + b$) to estimate the decay rate $p$, which provides the average gate fidelity $F_{\text{avg}} = p + \frac{1-p}{D}$.

Here, the Clifford group is defined as the normalizer of the Pauli group. The size of the Clifford group grows superexponentially with the number of qubits $d$, e.g., 24 (when $d = 1$), 11,520 (when $d = 2$), and 92,897,280 (when $d = 3$) [18]. The inverse of the composition of Clifford gates is efficiently (in time polynomial in $d$) computable on a classical computer thanks to the tableau representation of $d$-qubit Clifford group operations, e.g. [19,20], so the final gate in any sequence is as well. Each element of the group can be generated by elementary gates, e.g. the phase, Hadamard and controlled-NOT (CNOT) gates [20,21].

The standard RB protocol contains the following three types of parameter sets that determines its sampling strategy:

- Clifford lengths: $\boldsymbol{m} = [m_1, m_2, \ldots, m_M]$. A *Clifford length* means the number of independent Clifford operations in a sequence.
- List of the number of sequences: $\boldsymbol{n} = [n_1, n_2, \ldots, n_M]$. The number of sequences with Clifford length $m_i$ is denoted by $n_i$.
- List of the number of shots: $\boldsymbol{k} = [k_1, k_2, \ldots, k_M]$. A *shot* means a single execution of a sequence. The number of shots for each sequence with Clifford length $m_i$ is denoted by $k_i$, which is often fixed to a common constant $k$.

We refer to these sets of parameters as *an RB configuration*.

We denote a function $f$ of variable $x$ with parameter $\theta$ by $f(x;\theta)$. We consider the simplest decay model $f(m; p, a, b) = a\, p^m + b$ as a function to be fitted in the estimation step of RB. Here $p$ represents the decay rate. The coefficients $a$ and $b$ absorb the state preparation and measurement (SPAM) errors as well as the error on the final gate. If there were no such errors, $a = 1 - \frac{1}{D}$ and $b = \frac{1}{D}$ would hold (see e.g. [14] for detailed analysis of the decay model).

We denote the average survival rate over $n_i$ random sequences (and $k_i$ shots for each sequence) with $m_i$ Cliffords

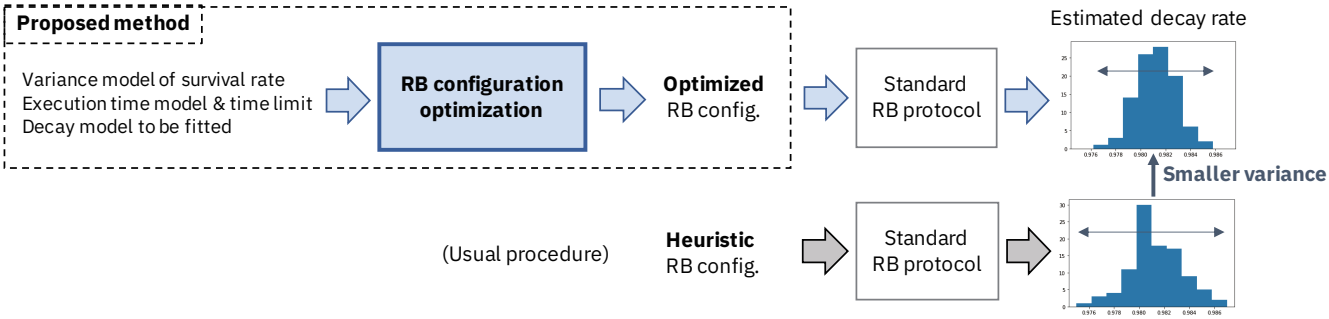



Fig. 2: Overview of the proposed method for RB configuration optimization

by $\bar{Y}_i$ and the standard deviation of $\bar{Y}_i$ by $\sigma_{\bar{Y}_i}$, respectively. Because $\sigma_{\bar{Y}_i}$ depends on an RB configuration $(m_i, n_i, k_i)$, it may be denoted by $\sigma_{\bar{Y}_i}(m_i, n_i, k_i)$ explicitly.

# 3. Method

## 3.1 Overview of RB Configuration Optimization

We provide an overview of our method for optimizing an RB configuration in Fig. 2. The three main features of our method are as follows.

- It requires a variance model of the survival rate and an execution time model in addition to the decay model. It also requires prior estimates of some parameters used in those models.
- It derives the confidence interval of the estimated decay rate as a function of an RB configuration, assuming the *weighted least squares* (WLS) estimator is used in the fitting step of the RB.
- It formulates a problem of minimizing the confidence interval under the constraint that the predicted execution time of sequences must be within a given time budget. Solving the optimization problem determines an optimal RB configuration.

Our method can be seen as preprocessing of the standard RB protocol. Once an optimal configuration is computed by our method, we only run the standard RB protocol following the configuration, and we obtain the estimated decay rate with minimal sampling errors.

In the following sections, we will detail our method. First, we introduce the variance model given that the variance of the survival rate differs depending on Clifford length (Section 3.2). To take the heteroskedasticity into account, we use the WLS estimator in the fitting. We derive the confidence interval of the estimated decay rate with the WLS estimator as a function of an RB configuration and use it as the objective function to be minimized (Section 3.3). We constrain the total execution time to avoid increasing the number of samples (e.g. the number of sequences) infinitely to decrease the sampling error. For that, we introduce a simple model to predict execution time (Section 3.4). Finally, we provide a formulation for the problem of optimizing RB configuration as a mathematical optimization problem (Section 3.5).

## 3.2 Variance Model of Survival Rate

Although our method can accept any model of the variance of the average survival rate, models with fewer parameters are preferable because the method requires prior estimates of the parameters. The goal of the variance model is to capture how the variance of the survival rate varies depending on Clifford length. Any model of upper, average, or lower bound is acceptable for this purpose provided that it approximates the form of the function over the entire region of Clifford length. However, to the best of our knowledge, there is no model perfectly suitable for the use in RB configuration optimization. For example, the upper bound in [22] only provided analysis of the variance for a specific region of Clifford length. An improved bound without such a restriction on Clifford length was proposed in [16] for a variant of the standard RB, however, how well the bound fits to real-world variance data was not discussed.

We model the variance of the average survival rate $\sigma^2_{\bar{Y}_i}$ as follows. First, we assume the sequence sampling error is independent of the shot sampling error, and the total variance is given as the sum of those variances:

$$\sigma^2_{\bar{Y}_i} = \frac{1}{n_i}\{\sigma^2_{\mathrm{seq}(i)} + \sigma^2_{\mathrm{shot}(i)}\} \tag{1}$$

Then we approximate each of them by, respectively,

$$\sigma^2_{\mathrm{seq}(i)} \approx \beta\, q^{m_i}\,(1 - q^{m_i}), \tag{2}$$

$$\sigma^2_{\mathrm{shot}(i)} \approx \frac{\mu_i\,(1-\mu_i)}{k_i} \text{ with } \mu_i = \left(1 - \frac{1}{D}\right)\hat{p}^{m_i} + \frac{1}{D}. \tag{3}$$

Note that we expect the parameter $q$ should be close to the decay rate $p$. Here, the approximate shot sampling error (3) is derived under the strong assumptions that the mean of the survival rate for all the sequences with a Clifford length $m_i$ could be the same $\mu_i$ and that there are no errors in SPAM or the final Clifford operation. The approximate sequence sampling error (2) is empirical, but it can be roughly explained by the effect of gate-dependent errors. As Clifford length increases, the variance once increases as the tail of the distribution of the survival rate widens due to the increased variations of Clifford gates in a sequence. However, the variance eventually converges to zero as Clifford length $m_i$ approaches infinity because the survival rate is bounded within 0 to 1 and its average decays exponentially as $m_i$

grows.

In fact, this model explains real-world variance data very well (see Fig. 3) even though it has no strong theoretical justification. (see Appendix A.1 for more examples). In the

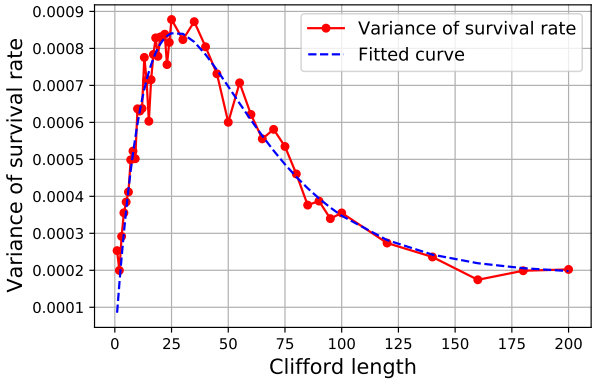



Fig. 3: Sample variance of survival rate for each Clifford length on a real device and their fitted curve to the variance model

figure, the sample variance of the survival rate for each Clifford length is plotted as red points, and the blue dashed line shows the fitted curve to the model $n_i\,\sigma^2_{\bar{Y}_i}$ defined by (1)–(3). The survival rates for each Clifford length are sampled by running 400 sequences (each with 1,000 shots) on an IBM Quantum system ibmq_athens.

As also shown in Fig. 3, the variance of survival rate is not uniform regarding Clifford length in general. That suggests the assumption of uniform variance, which is assumed in the ordinary least squares (OLS) estimator, is not satisfied as mentioned in several previous studies, e.g. [16, 23]. To take the heteroskedasticity into accounts, using the iteratively reweighted least squares estimator was recommended in [16] and a Bayesian-based estimation algorithm was proposed in [23]. In this paper, we assume to use the weighted least squares (WLS) estimator in the fitting. Recall that we are developing a method for optimizing the RB configuration, not comparing the performance of other estimators with the OLS or WLS estimator. The WLS estimator is suitable for our purpose because it enables us to analytically derive the confidence interval of the estimated decay rate as shown in the next section.

### 3.3 Confidence Interval of Estimated Decay Rate

The confidence interval of the decay rate estimated by the ordinary least squares (OLS) estimator is explicitly given in [15]. We extend their analysis to the case of the weighted least squares (WLS) estimator.

The $(1-\alpha)$ confidence interval of decay rate $p$ at the estimate $\hat{p}$ of a nonlinear regression model $y_i = f(m_i; p, a, b) = a\,p^{m_i} + b$ by the WLS estimator with a weight matrix $W = diag(w_1, w_2, \ldots, w_n)$ is approximately given by

$$|p - \hat{p}| \le t_{M-3,1-\alpha/2}\,\sqrt{H\,s^2}. \tag{4}$$

This is obtained by replacing $y_i$ with $\sqrt{w_i}\,y_i$ and $f(m_i)$ with $\sqrt{w_i}\,f(m_i)$ in the confidence interval of the estimated decay rate by the OLS estimator. In the above (4), $t_{M-3,1-\alpha/2}$ is the $(1-\frac{\alpha}{2})$ percentile of the $t$-distribution with $M-3$ degrees of freedom, $s^2$ is the weighted sample average of the squared residuals, i.e.

$$s^2 = \frac{\sum_{i=1}^{M} w_i \left| y_i - (\hat{a}\,\hat{p}^{m_i} + \hat{b}) \right|^2}{M-3}, \tag{5}$$

and $H$ is the scaling factor that reflects how far $s^2$ extends in the $\hat{p}$ axis, which is defined as

$$H \equiv \left[ \left( J(\hat{\boldsymbol{\theta}})^T W J(\hat{\boldsymbol{\theta}}) \right)^{-1} \right]_{\hat{p}\hat{p}} \tag{6}$$

Here $J(\hat{\boldsymbol{\theta}})$ is the Jacobian matrix of $f(m_i; \boldsymbol{\theta})$ $(i = 1, \ldots, M)$ at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$ with $\boldsymbol{\theta} = [p, a, b]$, whose $i$-th row is given by

$$J_{i,*}(\hat{\boldsymbol{\theta}}) = \left[ \hat{a}\,m_i\,\hat{p}^{m_i-1},\ \hat{p}^{m_i},\ 1 \right] \tag{7}$$

Note that $H$ is an element of the inverse of the $3 \times 3$ matrix; hence, it can be analytically computed (see Appendix A.2 for the details).

The WLS estimator weights the $i$-th observed value $(y_i)$ of a random variable $(Y_i)$ with a weight $w_i$ in the estimation, expecting the weight to make the variance of $Y_i$ uniform. Therefore, the weight is usually chosen to be reciprocal to the variance, i.e. $w_i = \sigma^{-2}_{Y_i}$. We set $w_i$ to $\sigma^{-2}_{\bar{Y}_i}$ with the variance model $\sigma^2_{\bar{Y}_i}$ defined in Section 3.2. Consequently, the weight matrix $W$ becomes a function of $\boldsymbol{m}$, $\boldsymbol{n}$ and $\boldsymbol{k}$ with the variance model parameters $(q, \beta)$. Note that it may be explicitly denoted by $W(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k}; q, \beta)$, and thus, $H$ by $H(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k}; \hat{p}, \hat{a}, q, \beta)$.

Our goal is finding the RB configuration that minimizes the right-hand side of (4), so the objective function to be minimized is defined as

$$h(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k}) \equiv t_{M-3,1-\alpha/2}\,\sqrt{H'(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k}; \hat{p}, q, \beta)} \tag{8}$$

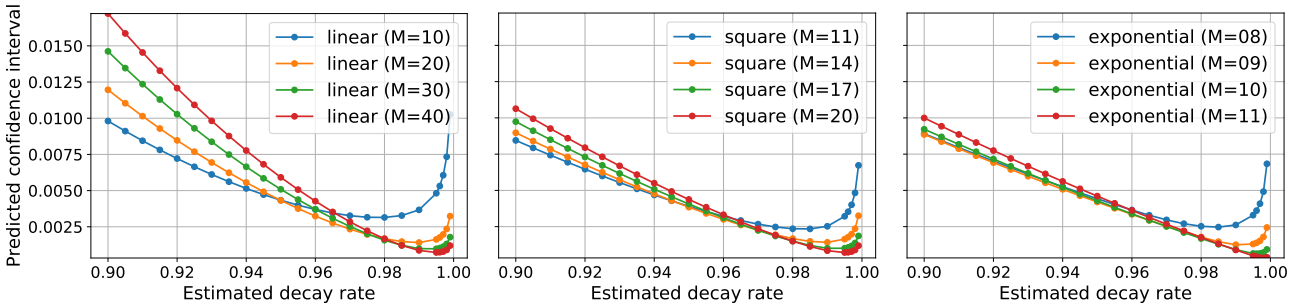
by omitting the constant factors on the right-hand side of (4), where $H'(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k}; \hat{p}, q, \beta) = \hat{a}^2 H(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k}; \hat{p}, \hat{a}, q, \beta)$. The replacement of $H$ by $H'$ is only for factoring out an ineffective parameter $\hat{a}$. Provided a confidence level $\alpha$ and some prior estimates of parameters $(\hat{p}, q, \beta)$ in advance, the objective function $h$ becomes a function depending only on $(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k})$. Note that the value of parameter $\hat{p}$ that we use to define the objective function may differ from the actual decay rate estimated by an RB experiment we will run afterwards.

### 3.4 Execution Time Model

We consider a simple approximated execution time model. It estimates the time required for the execution of RB sequences with a configuration $(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k})$ by

$$t(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k}) \approx \sum_{i=1}^{M} n_i\,k_i\,(c_1\,m_i + c_0). \tag{9}$$

Here, $c_1$ is a coefficient that reflects how execution time increases with Clifford length $m_i$, and $c_0$ is a constant time




(a) Linear configuration (b) Square configuration (c) Exponential configuration

Fig. 4: Confidence interval of estimated decay rate, predicted by the proposed method with different parameters (estimated decay rate and the dimension of Clifford lengths $M$) for each heuristic configuration

required for running a sequence independent of Clifford length $m_i$. In this model, some overhead time such as the loading time of control instructions due to sequence switching, is not taken into account. The typical choice for $c_1$ is the average single Clifford gate time, and that for $c_0$ is the sum of the average measurement time and the interval time between shots. Given the time limit $T$ for the execution of an RB experiment, we optimize the objective function under the time constraint

$$t(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k}) \leq T. \tag{10}$$

### 3.5 Formulation of Configuration Optimization

Now we can formulate the problem of finding an optimal RB configuration as a mathematical optimization problem as follows.

$$\begin{aligned} &\text{minimize} && h(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k}) \\ &\text{subject to} && m_i \geq m_{i-1} + 1, \text{ for } i = 2, 3, \ldots, M, \\ & && t(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k}) \leq T \\ & && \boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k} \in \mathbb{Z}_+^M \text{ (positive integer vector).} \end{aligned} \tag{11}$$

Here $h(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k})$ is the confidence interval of the estimated decay rate defined in (8), $t(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k})$ is the approximate execution time described in (9), and $T$ is the available time budget. We introduce the constraint $m_i \geq m_{i-1} + 1$ to remove the symmetry in the formulation and make the problem more easily solvable.

There are two major difficulties in solving the (nonlinear) optimization problem (11) in practice:

(1) The variability of the dimension of the Clifford lengths ($M$ may vary): Most of optimization algorithms assume the dimension of solution space is fixed.

(2) Integer variables ($\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k} \in \mathbb{Z}_+^M$): Discrete solution spaces usually prevent the use of descent methods that performs well in practice.

We overcome the aforementioned difficulties as follows.

(1) We repeatedly solve the problem with a fix $M$ for $M = 4, 5, \ldots, M_{\max}$, and select the best among them. We do not need to take such a large $M_{\max}$ in practice (typically, $M_{\max} \leq 40$), so computation time is not a concern in solving multiple problem instances.

(2) We relax the integer constraints and consider continuous variables (i.e., $\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{k} \in \mathbb{R}_+^M$). We round the optimal solution for the relaxed problem and obtain a near-optimal integer solution for the original problem. Rounding continous solutions to discrete ones is standard to obtain approximate solutions of (mixed) integer programming [24].

## 4. Experiments

We conducted two experiments: simulating the expected performance of typical heuristic configurations and evaluating the effect of configuration optimization on real devices. The first computational experiment was to show that our proposed method can be useful to estimate the performance of a given configuration. The second experiment was to determine if our method can provide an optimal configuration that yields estimated decay rates with small variance in a real environment. Throughout both experiments, we investigated 2-qubit RB ($D = 4$). We implemented our method in Python and used Qiskit 0.23.5, which is an open-source quantum computing software development framework [25], to generate and run sequences. We used IBM Quantum systems [13] with 5 qubits for experiments on real devices. Hereafter, we omit the ibmq_ prefix in the device name for simplicity.

### 4.1 Computational Experiment

Our method can be useful to predict the confidence interval of a decay rate estimated by RB under a given configuration. That also means it can simulate how the change in decay rate affects the performance (i.e., predicted confidence interval) of a given configuration. We conducted such a simulation for the three types of heuristic RB configurations: linear, square, and exponential. Each configuration designed to have its own Clifford lengths $\boldsymbol{m}$, a number of sequences $n$ identical to the Clifford lengths, and a fixed $k = 100$. Clifford lengths of a linear, square and exponential configuration were set to $[10(x-1)+1]$, $[x^2]$ and $[2^{x-1}]$ for $x = 1, \ldots, M$, respectively. In the simulation, we fixed the confidence level $\alpha$ to 0.05. The two parameters $(q, \beta)$ in the objective function (4) were fixed as $q = \hat{p}$ and $\beta = 0.0025$. The value of $\beta$ was obtained from a preliminary experiment on an IBM Quantum system athens. The two parameters $(c_1, c_0)$ in the execution

Table 1: Summary of RB configurations used in the experiment on real devices

| Configuration | $M^\dagger$ | $\boldsymbol{m}$ | $\boldsymbol{n}$ | $N^\ddagger$ | Estimated exec. time [s] |
|---|---|---|---|---|---|
| linear | 21 | $[1, 11, 21, 31, \ldots, 201]$ | 5 (identical) | 105 | 3.261 |
| square | 17 | $[1, 4, 9, 16, \ldots, 289]$ | 6 (identical) | 102 | 3.193 |
| exponential | 10 | $[1, 2, 4, 8, \ldots, 512]$ | 10 (identical) | 100 | 3.114 |
| optimal | 16 | optimized | optimized | 99 | 2.974 |
| optimal-identical-n | 33 | optimized | 3 (identical) | 99 | 2.961 |

$^\dagger$ The dimension of Clifford lengths $\boldsymbol{m}$, $^\ddagger$ The total number of sequences, i.e. sum of $\boldsymbol{n}$

time model (9) were set as $c_1 = 0.6\,[\mu s]$ and $c_0 = 250\,[\mu s]$, based on the public properties of athens: The gate time of CNOT gate is around $0.4\,\mu s$ (one Clifford is composed of 1.5 CNOT gates on average), the default intervals between shots is $250\,\mu s$, and the measurement time is around $3\,\mu s$. We set the time budget $T$ to 3 seconds.

Figure 4 shows how the predicted confidence interval of the estimated decay rate varies when changing the estimated decay rate $\hat{p}$ and the dimension of Clifford lengths $M$ for each heuristic configuration. As expected, the optimal $M$ varied depending on the decay rate for all types of configurations. Among the three types, exponential was most stable against the changing decay rate, demonstrating its usefulness when there is little knowledge on the decay rate in advance. It can be also seen that linear and square works well in a specific range of decay rates if we can select the optimal $M$ when square is more stable than linear. That suggests square is a good option if we have a good prior estimated decay rate.

In this way, our method can be used to predict the performance of heuristic configurations. Moreover, it can be used to select the best among candidates of configurations. For example, let consider any of the above heuristic configurations parameterized by $M$. The confidence interval for each $M$ can be computed by fixing the decay rate to a given prior estimate. Hence, the optimal $M$ that minimizes the confidence interval can be easily found, e.g., by grid search. Actually, heuristic configurations compared with an optimized configuration in the next section are prepared this way.

### 4.2 Experiment on Real Devices

The optimal configuration found by our method may deviate from the true optimal configuration value due to the following two gaps. One is the modeling gap, i.e., the models our method relies on cannot perfectly represent the real system. The other is the gap between the prior estimates and true values of the model parameters. These gaps are inevitable and too difficult to measure in practice. Therefore, we aim to demonstrate that our method is still useful in practice even after subtracting the impact of those gaps. That is, the configuration optimized by our method can yield the estimated decay rate with smaller variance than those from heuristic configurations in RBs on real devices.

Therefore, we conducted the experiment as follows.

(1) Preparing five configurations in total to be compared: Three heuristic configurations with optimized $M$ (linear, square, exponential; the same as discussed in Section 4.1), a configuration optimized by our method (optimal), and a reference configuration optimized by our method with restricting the number of sequences to be identical (optimal-identical-n).
(2) Bundling five sets of sequences generated with the five configurations up into one job. We run 100 jobs on each of the five devices (athens, quito, bogota, rome and lima), which corresponds to running 5 configurations $\times$ 5 devices $\times$ 100 jobs = 2,500 RBs.
(3) Analyzing the deviation in the decay rates estimated via 100 times of RBs for each device and configuration.

We ran RBs with common configurations on the five devices, each of which represents a different extent of imperfect prior estimates of model parameters. By comparing the results from different devices, we investigated how the imperfections in model parameters affect the performance of each configuration (i.e. the standard deviation of the resulting estimated decay rates). Throughout this experiment, we focused on optimizing $\boldsymbol{m}$ and $\boldsymbol{n}$, and we fixed the number of shots to 100 ($k = 100$).

In the first preparation step, we used the same values as those used in the Section 4.1 for the model parameters required in our method: $\alpha = 0.05$, $\hat{p} = q = 0.97$, $\beta = 0.0025$, $c_1 = 0.6\,[\mu s]$, $c_0 = 250\,[\mu s]$, and $T = 3\,[s]$. Note that the parameters $\hat{p}$, $q$, and $\beta$ are determined by a preliminary experiment on athens. We used `scipy.optimize` module to optimize the relaxed problem discussed in Section 3.5 when computing optimal and optimal-identical-n (see Appendix A.3 for the details). We set $M_{\max}$ to 40 for both configurations. The computation time to optimize each configuration was within 30 seconds (for all values of $M$) on a laptop PC with an Intel Core i7 2.7 GHz and 16 GB memory.

Table 1 summarizes five configurations prepared in the first step. For the optimized configuration (optimal), $\boldsymbol{m}$ =[1, 2, 19, 21, 23, 24, 25, 26, 27, 28, 29, 51, 52, 105, 195, 369], $\boldsymbol{n}$ =[8, 5, 5, 5, 6, 6, 5, 6, 6, 7, 5, 5, 5, 5, 8, 12]. For the optimized configuration with the identical $n$ constraint (optimal-identical-n), $\boldsymbol{m}$ =[1, 2, 3, 4, 5, 12, 20, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 53, 92, 136, 181, 227, 276, 329, 385, 445], $n = 3$. Note that we allowed to exceed the time budget constraint (due to rounding) for heuristic configurations while the optimal ones strictly comply with the constraint. Thus, it ensures that the comparison does not favor the optimal ones (i.e. our method) unfairly. It is interesting to note that optimal-identical-n seems to sample a similar number of neighbor Clifford lengths for each Clifford length that is heavily sampled in optimal; therefore,

Table 2: Adjusted standard deviation (raw standard deviation) of estimated decay rates by multiple runs of RBs on different real devices with different configurations. The best value among the configurations is in bold for each device. Average estimated decay rates over runs are stated in Avg $\hat{p}$ column for reference.

| | | | Heuristic configurations | | | Optimized configurations (Proposed) | |
|---|---|---|---|---|---|---|---|
| Device | Avg $\hat{p}$ | Runs | linear | square | exponential | optimal | optimal-identical-n |
| athens | 0.9706 | 98 | 0.001481 (0.002843) | 0.001217 (0.002658) | 0.001275 (0.002721) | **0.001074** (0.002798) | 0.001151 (0.002720) |
| quito | 0.9731 | 96 | 0.001469 (0.004673) | 0.001268 (0.004245) | 0.001378 (0.004419) | **0.000967** (0.004357) | 0.001190 (0.004310) |
| bogota | 0.9762 | 96 | 0.001160 (0.002483) | 0.001107 (0.002507) | 0.001231 (0.002648) | **0.000860** (0.002529) | 0.000971 (0.002397) |
| rome | 0.9836 | 93 | 0.000886 (0.001017) | **0.000774** (0.000953) | 0.000914 (0.001088) | 0.000911 (0.000919) | 0.000849 (0.001089) |
| lima | 0.9858 | 92 | 0.000979 (0.001201) | 0.000767 (0.000934) | **0.000730** (0.000910) | 0.000844 (0.001064) | 0.000798 (0.000956) |

they appear to be sampled from the same distribution.

In the second running step, we ran the sequences on five IBM Quantum devices with five qubits available to us as of May 19, 2021[*1]. We ran 100 jobs for each of the five devices, and a few of them failed while waiting in the job queue for unknown reasons. We analyzed the results of 98, 96, 96, 93, and 92 successful jobs from athens, quito, bogota, rome and lima, respectively. We used qubit 0 and 1 for all the devices.

Table 2 shows the adjusted standard deviations of decay rates estimated by the WLS estimator from the survival rate data obtained by repeated RBs on the five real devices. Because it took about 24 hours to run all jobs for each device (including the wait time in the queue), the raw values of standard deviations described in (·) in the table were affected by the temporal variation in the real decay rate. This can be observed from the changes of the average estimated decay rate over five configurations for each job at athens as shown in Fig. 5 (see Appendix A.4 for similar figures of other devices). Therefore, the values adjusted to remove the

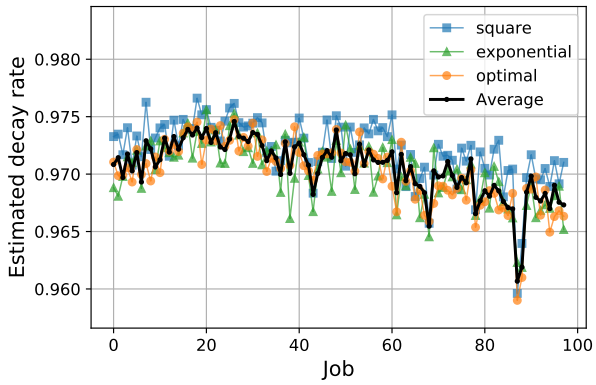



Fig. 5: Temporal variation in decay rate at athens: the black line depicts the average of estimated decay rate over all five configurations

effect of time fluctuation were described in the table and used for the evaluation. Specifically, the mean value of the five configurations minus the bias for each configuration was used as the expected value of the configuration at the job execution. The *adjusted standard deviation* was defined as the sample deviation calculated by the residual from the expected value. The bias for each configuration was set to the difference between the average of decay rates over all jobs and configuration, and the average of decay rates over all jobs for each configuration.

As shown in the table, optimal achieves the least deviation of the estimated decay rate by RBs on athens, quito, and bogota. This is the expected result because we used the values based on athens as the prior estimates for the model parameters. In particular, we set the prior estimate of the decay rate to 0.97, while the actual (estimated) decay rate of athens, quito, and bogota were all close to this value at 0.9706, 0.9731, and 0.9762, respectively. The improvement rates from square (the best of the heuristic configurations) to optimal in the standard deviation were quite large at 11.7% for athens, 23.7% for quito, and 22.3% for bogota. In contrast, heuristic configurations achieved slightly less deviation than the optimal configuration in rome and lima, where the actual decay rates (0.9836 and 0.9858) were larger than its prior estimate (0.97). This may be because the errors in prior estimates were so large that the optimization did not work as expected. Supporting the results on real devices, optimal achieves better results on noisy simulators of real devices: either the best or the second best (see Appendix A.5 for the details of noisy simulation). It is interesting to note that optimal-identical-n likely provided more stable results against the errors in prior estimates of decay rate comparing with optimal. This could be because the identical-n constraint mitigated the risk of over-optimization along with the given model parameters as if the regularization term prevents overfitting in machine learning.

In summary, our method is capable of finding a better configuration than typical heuristic configurations, at least when we have sufficiently accurate prior estimates of parameters required in the method.

## 5. Discussion

In this paper, we developed a method for minimizing the confidence interval under the time budget constraint. Our method can easily be modified so to minimize the execution time while bounding the confidence interval by replacing the formulation in (11) with

$$\begin{aligned} &\text{minimize} && t(\boldsymbol{m},\boldsymbol{n},\boldsymbol{k}) \\ &\text{subject to} && m_i \geq m_{i-1}+1, \text{ for } i=2,3,\ldots,M, \\ &&& h(\boldsymbol{m},\boldsymbol{n},\boldsymbol{k}) \leq \epsilon \\ &&& \boldsymbol{m},\boldsymbol{n},\boldsymbol{k} \in \mathbb{Z}_+^M \text{ (positive integer vector).} \end{aligned} \tag{12}$$

[*1] We tried to run on all six 5-qubit devices, but belem was too busy at that time and failed to complete all jobs within three days, so we removed it from the analysis.

where $\epsilon$ is the feasible upper bound of the confidence interval.

An advantage of our method is that it can be used for the first RB of a brand new device because the confidence model used in our method only requires the prior estimate of parameters, not their learning with training data. In the subsequent RBs, we could predict a more accurate confidence interval by using the decay rate estimated in the previous RB as the prior estimate. Ultimately, we could obtain the sample data from every RB, which can be used to improve the confidence model by learning from the data. In that case, such approaches that use machine learning techniques (as discussed in [23]) would work well.

In this paper, we focused on the standard RB protocol and proposed a method for optimizing a sampling strategy for it. Investigating how our method can be extended to other benchmarking protocols with random sampling, such as interleaved randomized benchmarking [26], dihedral benchmarking [27–29], or other variants [16, 30], is left for future work.

Furthermore, our method may be generalized to be combined with other protocols or algorithms that have the following properties:

- the protocol has a fitting parameter to be estimated in it (decay rate of RB),
- the protocol requires a fitting model used in it (survival rate decay model of RB), and
- the protocol repeats experiments to obtain samples to be fitted (survival rates of RB) with changing sampling parameters (RB configuration).

For example, an amplitude estimation algorithm proposed in [31] satisfies the above properties by mapping the fitting parameter to an angle related to the amplitude to be estimated, the fitting model to Equation (13) in the paper, and the sampling parameter to the number of repetitions of the amplifying operation.

## 6. Conclusion

We addressed the problem of optimizing sampling strategies for the standard RB. We showed how a sampling strategy is determined by configurable parameters (an RB configuration): Clifford lengths, the number of sequences for each Clifford length, and the number of shots. We discussed how the variance of the survival rate may not be uniform with respect to Clifford length and discussed how to model the heteroskedasticity in a simple form. We proposed a method for optimizing an RB configuration, which constructs a mathematical optimization problem that minimizes the confidence interval of the estimated decay rate while keeping the predicted execution time within a given time budget. The method does not change the RB protocol itself, so it is easily utilized as preprocessing. Our experiment on real devices demonstrated that the proposed method can find a better configuration, i.e. achieving smaller deviation in the resulting estimated decay rate, than typical heuristic configurations in practice. We believe the proposed method would be useful in practice to reduce the sampling error while maintaining the execution time, or to reduce the execution time while maintaining the sampling error.

## Acknowledgment

We thank Ikko Hamamura, Naoki Kanazawa, Antonio Córcoles, Shelly Garion, and Chris J. Wood for their helpful comments.

### References

[1] Chuang, I. L. and Nielsen, M. A.: Prescription for experimental determination of the dynamics of a quantum black box, *Journal of Modern Optics*, Vol. 44, No. 11-12, pp. 2455–2467 (1997).
[2] Gross, D., Liu, Y.-K., Flammia, S. T., Becker, S. and Eisert, J.: Quantum state tomography via compressed sensing, *Physical review letters*, Vol. 105, No. 15, p. 150401 (2010).
[3] Flammia, S. T. and Liu, Y.-K.: Direct fidelity estimation from few Pauli measurements, *Physical review letters*, Vol. 106, No. 23, p. 230501 (2011).
[4] da Silva, M. P., Landon-Cardinal, O. and Poulin, D.: Practical characterization of quantum devices without tomography, *Physical Review Letters*, Vol. 107, No. 21, p. 210404 (2011).
[5] Knill, E., Leibfried, D., Reichle, R., Britton, J., Blakestad, R. B., Jost, J. D., Langer, C., Ozeri, R., Seidelin, S. and Wineland, D. J.: Randomized benchmarking of quantum gates, *Physical Review A*, Vol. 77, No. 1, p. 012307 (2008).
[6] Magesan, E., Gambetta, J. M. and Emerson, J.: Scalable and robust randomized benchmarking of quantum processes, *Physical review letters*, Vol. 106, No. 18, p. 180504 (2011).
[7] Barends, R., Kelly, J., Megrant, A., Veitia, A., Sank, D., Jeffrey, E., White, T. C., Mutus, J., Fowler, A. G., Campbell, B. et al.: Superconducting quantum circuits at the surface code threshold for fault tolerance, *Nature*, Vol. 508, No. 7497, pp. 500–503 (2014).
[8] Muhonen, J., Laucht, A., Simmons, S., Dehollain, J., Kalra, R., Hudson, F., Freer, S., Itoh, K. M., Jamieson, D., McCallum, J. et al.: Quantifying the quantum gate fidelity of single-atom spin qubits in silicon by randomized benchmarking, *Journal of Physics: Condensed Matter*, Vol. 27, No. 15, p. 154205 (2015).
[9] Chow, J., Gambetta, J. M., Tornberg, L., Koch, J., Bishop, L. S., Houck, A. A., Johnson, B., Frunzio, L., Girvin, S. M. and Schoelkopf, R. J.: Randomized benchmarking and process tomography for gate errors in a solid-state qubit, *Physical review letters*, Vol. 102, No. 9, p. 090502 (2009).
[10] Córcoles, A. D., Gambetta, J. M., Chow, J. M., Smolin, J. A., Ware, M., Strand, J., Plourde, B. L. and Steffen, M.: Process verification of two-qubit quantum gates by randomized benchmarking, *Physical Review A*, Vol. 87, No. 3, p. 030301 (2013).
[11] McKay, D. C., Wood, C. J., Sheldon, S., Chow, J. M. and Gambetta, J. M.: Efficient Z gates for quantum computing, *Physical Review A*, Vol. 96, No. 2, p. 022330 (2017).
[12] McKay, D. C., Sheldon, S., Smolin, J. A., Chow, J. M. and Gambetta, J. M.: Three-qubit randomized benchmarking, *Physical review letters*, Vol. 122, No. 20, p. 200502 (2019).
[13] IBM Quantum: IBM Quantum. `https://quantum-computing.ibm.com/` (accessed Sep. 21, 2021).
[14] Magesan, E., Gambetta, J. M. and Emerson, J.: Characterizing quantum gates via randomized benchmarking, *Physical Review A*, Vol. 85, No. 4, p. 042311 (2012).
[15] Epstein, J. M., Cross, A. W., Magesan, E. and Gambetta, J. M.: Investigating the limits of randomized benchmarking protocols, *Physical Review A*, Vol. 89, No. 6, p. 062321 (2014).
[16] Helsen, J., Wallman, J. J., Flammia, S. T. and Wehner, S.: Multiqubit randomized benchmarking using few samples, *Physical Review A*, Vol. 100, No. 3, p. 032304 (2019).
[17] Harper, R., Hincks, I., Ferrie, C., Flammia, S. T. and Wallman, J. J.: Statistical analysis of randomized benchmarking, *Physical Review A*, Vol. 99, No. 5, p. 052350 (2019).
[18] Ozols, M.: Clifford group, *Essays at University of Waterloo, Spring* (2008). `http://home.lu.lv/~sd20008/papers/essays/Clifford%20group%20[paper].pdf` (accessed Sep. 21, 2021).
[19] Dehaene, J. and De Moor, B.: Clifford group, stabilizer states, and linear and quadratic operations over GF(2), *Physical Review A*, Vol. 68, No. 4, p. 042318 (2003).

[20] Aaronson, S. and Gottesman, D.: Improved simulation of stabilizer circuits, *Physical Review A*, Vol. 70, No. 5, p. 052328 (2004).
[21] Bravyi, S., Shaydulin, R., Hu, S. and Maslov, D.: Clifford Circuit Optimization with Templates and Symbolic Pauli Gates, *arXiv preprint arXiv:2105.02291* (2021).
[22] Wallman, J. J. and Flammia, S. T.: Randomized benchmarking with confidence, *New Journal of Physics*, Vol. 16, No. 10, p. 103032 (2014).
[23] Granade, C., Ferrie, C. and Cory, D. G.: Accelerated randomized benchmarking, *New Journal of Physics*, Vol. 17, No. 1, p. 013042 (2015).
[24] Bertsimas, D. and Weismantel, R.: *Optimization over integers*, Vol. 13, Dynamic Ideas Belmont, MA (2005).
[25] Qiskit: Qiskit: An Open-source Framework for Quantum Computing (2019). `https://www.qiskit.org/` (accessed Sep. 21, 2021).
[26] Magesan, E., Gambetta, J. M., Johnson, B. R., Ryan, C. A., Chow, J. M., Merkel, S. T., Da Silva, M. P., Keefe, G. A., Rothwell, M. B., Ohki, T. A. et al.: Efficient measurement of quantum gate error by interleaved randomized benchmarking, *Physical review letters*, Vol. 109, No. 8, p. 080505 (2012).
[27] Carignan-Dugas, A., Wallman, J. J. and Emerson, J.: Characterizing universal gate sets via dihedral benchmarking, *Physical Review A*, Vol. 92, No. 6, p. 060302 (2015).
[28] Cross, A. W., Magesan, E., Bishop, L. S., Smolin, J. A. and Gambetta, J. M.: Scalable randomised benchmarking of non-clifford gates, *npj Quantum Information*, Vol. 2, No. 1, pp. 1–5 (2016).
[29] Harper, R. and Flammia, S. T.: Estimating the fidelity of T gates using standard interleaved randomized benchmarking, *Quantum Science and Technology*, Vol. 2, No. 1, p. 015008 (2017).
[30] Proctor, T. J., Carignan-Dugas, A., Rudinger, K., Nielsen, E., Blume-Kohout, R. and Young, K.: Direct randomized benchmarking for multiqubit devices, *Physical review letters*, Vol. 123, No. 3, p. 030503 (2019).
[31] Tanaka, T., Suzuki, Y., Uno, S., Raymond, R., Onodera, T. and Yamamoto, N.: Amplitude estimation via maximum likelihood on noisy quantum computer, *arXiv preprint arXiv:2006.16223* (2020).

# Appendix

## A.1 Sample Variances of Survival Rate for each Clifford Length on IBM Quantum Systems

We provide four more sets of sample variance data obtained from the devices of IBM Quantum systems, quito, bogota, rome and lima, and the fitted curve for each device in Fig. A·1. In each figure, the sample variance of survival rate for each Clifford length is plotted as red points and the blue dashed line shows the fitted curve to the model $n_i\,\sigma^2_{\hat{Y}_i}$ defined by (1)–(3). As shown in the figure, the sample variance of survival rate is not uniform with respect to Clifford length for all devices. The model relatively explains the sample variance well although some divergence in the tail is observed in several cases.

## A.2 Explicit Form of the Confidence Interval Function

We provide the explicit expression of $H'$ used in the definition of confidence interval function discussed in Section 3.3. Recall that

$$H' = \hat{a}^2 H = \hat{a}^2 \left[ \left( J(\hat{\boldsymbol{\theta}})^T W J(\hat{\boldsymbol{\theta}}) \right)^{-1} \right]_{\hat{p}\hat{p}} \tag{A.1}$$

and $J(\hat{\boldsymbol{\theta}})^T W J(\hat{\boldsymbol{\theta}})$ is a 3x3 matrix in the form of

$$\begin{bmatrix} A & \boldsymbol{c} \\ \boldsymbol{c}^T & u \end{bmatrix}$$

where

$$A = \begin{bmatrix} a^2 \sum_i w_i\, m_i^2\, p^{2m_i-2} & a \sum_i w_i\, m_i\, p^{2m_i-1} \\ a \sum_i w_i\, m_i\, p^{2m_i-1} & \sum_i w_i\, p^{2m_i} \end{bmatrix},$$

$$c^T = \left[ a \sum_i w_i\, m_i\, p^{m_i-1},\ \sum_i w_i\, p^{m_i} \right],$$

and $u = \sum_i w_i$. Here and hereafter, we omit the superscript "$\hat{}$" of $p$, $a$, $b$ for brevity. Using the block matrix inversion formula

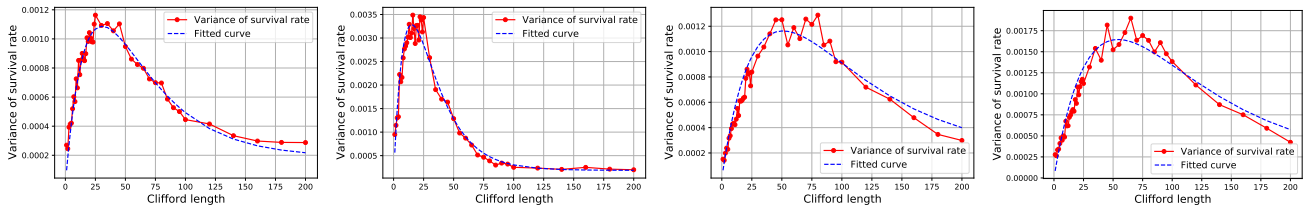
$$\begin{bmatrix} A & \boldsymbol{c} \\ \boldsymbol{c}^T & u \end{bmatrix}^{-1} = \begin{bmatrix} (A - \frac{\boldsymbol{c}\,\boldsymbol{c}^T}{u})^{-1} & * \\ * & * \end{bmatrix}$$

and the explicit form of the inverted 2x2 matrix, we obtain

$$H' \equiv \frac{\left[\sum w_i\, p^{2m_i} - \frac{(\sum w_i\, p^{m_i})^2}{u}\right]}{\left[\sum w_i\, p^{2m_i} - \frac{(\sum w_i\, p^{m_i})^2}{u}\right]\left[\sum w_i\, m_i^2\, p^{2m_i-2} - \frac{(\sum w_i\, m_i\, p^{m_i-1})^2}{u}\right] - \left[\sum w_i\, m_i\, p^{2m_i-1} - \frac{(\sum w_i\, p^{m_i})(\sum_i w_i\, m_i\, p^{m_i-1})}{u}\right]^2}. \tag{A.2}$$

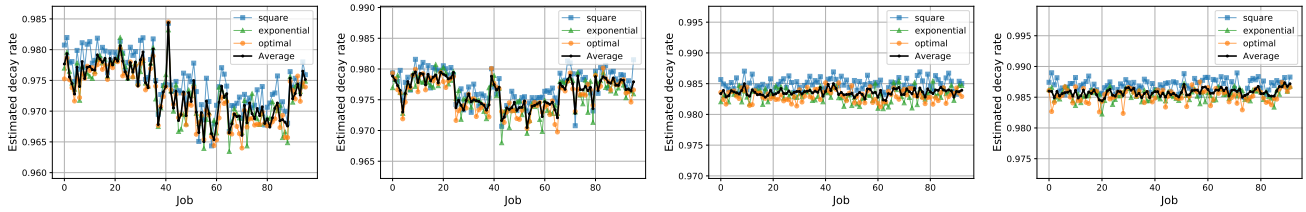
## A.3 Solving the Optimization Problem

The optimization problem (11) is obviously nonlinear in $\boldsymbol{m}$ and $\boldsymbol{n}$, recalling that the objective function includes the term $\sqrt{H'}$ with $H'$ written down in (A.2) and $w_i$ is proportional to $\sqrt{n_i}$ because $w_i$ is set to $\sigma^{-2}_{\hat{Y}_i}$. Actually, it is even non-convex. In general, the non-convex optimization problem has multiple locally optimal solutions and it is often difficult to find the globally optimal solution. In the experiments in Section 4.2, we used `scipy.optimize.minimize` function (using default parameters except for `tol`=1.0e-10) to compute one of the locally optimal solutions. We added extra constraints $n_i \geq 5$ in order to mitigate the impact of outliers in the formulation for optimal. We replaced the vector variable $\boldsymbol{n}$ with a scalar variable $n$ to represent a common number of sequences in the formulation for optimal-identical-n. We set the initial guess (`x0`) to $\boldsymbol{m} = [1, 2, 3, \ldots, M]$ and $n_i = 5$ for all $i$ (for optimal) and $n = 3$ (for optimal-identical-n). Notice that we may find a better solution by changing the initial guess.




(a) quito (b) bogota (c) rome (d) lima

Fig. A·1: Sample variance of survival rate for each Clifford length on different devices and the fitted curve to the variance model




(a) quito (b) bogota (c) rome (d) lima

Fig. A·2: Temporal variation in decay rate on different devices: The average estimated decay rate over all five configurations by job (black line) and the estimated decay rate with square/exponential/optimal by job (blue/green/orange)

Table A·1: Standard deviations of estimated decay rates by 100 runs of RBs with different configurations on noisy simulators of different real devices. The best value among the configurations is in bold for each device. Average estimated decay rates over runs are stated in Avg $\hat{p}$ column for reference.

| | | | Heuristic configurations | | | Optimized configurations (Proposed) | |
|---|---|---|---|---|---|---|---|
| Device | Avg $\hat{p}$ | Runs | linear | square | exponential | optimal | optimal-identical-n |
| athens | 0.9758 | 100 | 0.001311 | 0.001125 | 0.001277 | **0.001032** | 0.001104 |
| quito | 0.9779 | 100 | 0.001188 | 0.001052 | 0.001342 | 0.001028 | **0.001024** |
| bogota | 0.9805 | 100 | 0.000927 | 0.000910 | 0.000909 | 0.000860 | **0.000846** |
| rome | 0.9829 | 100 | 0.000888 | 0.000941 | 0.000992 | 0.000838 | **0.000778** |
| lima | 0.9825 | 100 | 0.001111 | 0.000946 | 0.000975 | 0.000843 | **0.000792** |

## A.4 Temporal Variation in Decay Rate

We provide four more plots of the temporal variation in decay rate on IBM Quantum devices, quito, bogota, rome and lima in Fig. A·2. The estimated decay rate with square, exponential and optimal for each job are also shown in the figure (we omit that with linear and optimal-identical-n for ease of reading). As seen in the figures, square was likely to estimate slightly higher decay rates than the other configurations for all devices. In fact, the sample average of decay rates from square was 0.9750, 0.9772, 0.9837, and 0.9870 while those from others were within [0.9723, 0.9728], [0.9757, 0.9762], [0.9825, 0.9828], and [0.9852, 0.9857] for quito, bogota, rome and lima, respectively. This might imply that we need more investigation on the bias in the estimated decay rate.

## A.5 Results of Noisy Simulation

To confirm RB configurations optimized by our method performs as expected in a synthetic environment, we conducted the same experiments described in Section 4.2 on noisy simulators. We utilized `NoiseModel.from_backend` function in Qiskit to simulate noisy execution of sequences on each device. We show the results in Table A·1, which confirm that optimal can achieve better results on noisy simulators of real devices: either the best or the second best.