

Increasing the Measured Effective Quantum Volume with Zero Noise Extrapolation

Elijah Pelofske^{*1}, Vincent Russo^{†2}, Ryan LaRose^{2,3}, Andrea Mari^{2,5}, Dan Strano², Andreas Bärtschi¹, Stephan Eidenbenz¹, and William J. Zeng^{2,4}

¹CCS-3 Information Sciences, Los Alamos National Laboratory

²Unitary Fund

³Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland

⁴Goldman Sachs & Co, New York, NY

⁵Physics Division, School of Science and Technology, Università di Camerino, Italy

Abstract

Quantum Volume is a full-stack benchmark for near-term quantum computers. It quantifies the largest size of a square circuit which can be executed on the target device with reasonable fidelity. Error mitigation is a set of techniques intended to remove the effects of noise present in the computation of noisy quantum computers when computing an expectation value of interest. Effective quantum volume is a proposed metric that applies error mitigation to the quantum volume protocol in order to evaluate the effectiveness not only of the target device but also of the error mitigation algorithm. Digital Zero-Noise Extrapolation (ZNE) is an error mitigation technique that estimates the noiseless expectation value using circuit folding to amplify errors by known scale factors and then extrapolating computed expectation values to the zero-noise limit. Here we demonstrate that ZNE, with global and local unitary folding with fractional scale factors, in conjunction with dynamical decoupling, can increase the effective quantum volume over the vendor-measured quantum volume. Specifically, we measure the effective quantum volume of four IBM Quantum superconducting processor units, obtaining values that are larger than the vendor-measured quantum volume on each device. This is the first such increase reported.

1 Introduction

Quantum Volume (QV) [1–5] is a proposed full-stack metric for evaluating the capability of error-prone quantum computers in the NISQ era [6]. The intention of the Quantum Volume protocol is to measure the largest *square* circuit acting on n qubits with depth $d = n$ which can be computed with reasonably high fidelity on a target Quantum Processing Unit (QPU) ¹. A QV circuit is composed of alternating layers of random qubit index permutations and random SU(4) two-qubit operations. The logical diagram of a QV circuit is given in Figure 1. Each of the d layers of two-qubit SU(4) gate operations act on $n' := \lfloor \frac{n}{2} \rfloor$ pairs of qubits, meaning that if n is odd then one qubit will be idle in each layer. The Quantum Volume protocol is based on the *heavy output generation problem* [7], where a single QV circuit U has an ideal bitstring output distribution given in eq. (1)

$$p_U(x) = |\langle x|U|0 \rangle|^2. \quad (1)$$

Given the ideal distribution $p_U(x)$ for a single QV circuit U , each probability for all 2^n possible states is sorted $p_1 \leq p_2 \leq \dots \leq p_{2^n}$. Eq. (2) defines the set of heavy bitstrings H_U , given the median of all of the probabilities p_{median} and all of the classically computed bitstring probabilities $p_U(x)$,

$$H_U = \{x \in \{0, 1\}^n : p_U(x) > p_{\text{median}}\}. \quad (2)$$

^{*}Email: epelofske@lanl.gov

[†]Email: vincent@unitary.fund

¹Typically, a quantum computer passing the QV protocol for n qubits is said to have a quantum volume of 2^n . Here, we will simply use the notation of n instead of 2^n .

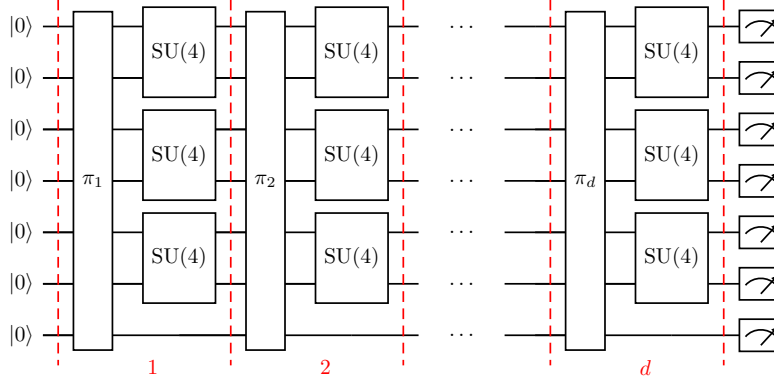


Figure 1: Circuit drawing for a logical QV circuit acting on $n = 7$ qubits, where each layer $d = 1, \dots, d = 7$ consists of a random permutation of the qubit labels followed by $n' = 3 = \lfloor \frac{7}{2} \rfloor$ random $SU(4)$ gates acting on consecutive pairs of (the permuted) qubits. This figure was realized using Quantikz [8].

QPU name	IBM Quantum processor type	Number of qubits	Vendor-measured Quantum Volume	Measured (ZNE) effective quantum volume
ibmq_toronto	Falcon r4	27	5	6
ibm_geneva	Falcon r8	27	5	6
ibm_auckland	Falcon r5.11	27	6	7
ibm_hanoi	Falcon r5.11	27	6	7

Table 1: IBM Quantum superconducting processor summary from the experiments presented in this work. The basis gates for all four of these devices are CX, ID, RZ, SX, X. Quantum Volume and Effective Quantum Volume numbers are reported as n instead of 2^n . The Vendor-measured Quantum Volume values were recorded from the vendor at the time these experiments were performed.

Given the set H_U , the heavy output probability (HOP) is the probability of measuring a bitstring which belongs to H_U . Empirically, given the measured bitstrings obtained by executing a Quantum Volume circuit, the HOP can be estimated by taking the proportion of the measurements which are in the set H_U , out of the total number of measurements that were made.

As it was originally proposed, the QV protocol [1, 2] specifies that some reasonably large number of samples (e.g. at least 10) is measured from at least 100 randomly generated QV circuits that are compiled and executed on a target quantum computer which we wish to measure the computational capabilities of. Therefore, for each QV circuit, we can estimate the associated HOP and, by averaging over all QV circuits, we obtain an estimate of the mean HOP. For a noiseless quantum computation the mean HOP approaches $\frac{1+\ln(2)}{2} \approx 0.85$ [1, 7] in the limit of large n . The QV protocol [1] passes for size n if a quantum computer is able to reliably sample an ensemble of random QV circuits at a mean HOP greater than $\frac{2}{3}$. If the computation loses all coherence, and the circuit is effectively randomly sampling bits, then the mean HOP converges to 0.5.

Quantum error mitigation, alongside quantum error suppression, is a family of techniques that can be utilized to remove or suppress errors in a computation performed on a NISQ computer [9–13]. Quantum error mitigation is thought of as an intermediate step in the development of quantum computers, eventually leading to the development of fully realized quantum error correction [14, 15]. The trade off that comes from using quantum error mitigation is that in general there is significant computational cost of applying quantum error mitigation, namely in terms of additional compute time and measurements that must be obtained. In particular, the number of samples required to measure a certain accuracy of an observable on a noisy quantum computer scales exponentially with circuit depth [16, 17].

One of the properties of Quantum Volume that makes it very relevant for evaluating quantum computing performance is that it involves the full stack of software and hardware—including what parts of the hardware are used, and what compilation is used [4]. Effective quantum volume [18] is a proposed variant of the Quantum Volume metric which uses error mitigation strategies in conjunction with Quantum Volume in order to evaluate

the quality of quantum computations that can be performed on modern quantum computers with error mitigation. Zero Noise Extrapolation (ZNE) [10, 19–28] is a general-purpose error mitigation algorithm that uses expectation values computed at different noise levels so as to extrapolate the computation to its zero noise limit. The different noise levels can be indirectly created by circuit folding operations that leave the circuit invariant in its theoretical expectation value, but purposely increase the number of instructions (gates) which means there will be more errors within the computation on the NISQ computer, such that the amount of noise within the computation will have been scaled by a pre-determined scale factor λ . The effect of ZNE when applied to a Quantum Volume circuit is to effectively increase the error-mitigated HOP, obtaining a result closer to the ideal limit of $\frac{1+\ln(2)}{2}$. Informally, the Zero Noise Extrapolation Quantum Volume benchmark, which we refer to as the *effective quantum volume* [18], is defined such that a quantum computer that passes the test for square dense random circuits of depth n and acting on n qubits will be able to compute reasonably good expectation values using ZNE for general quantum circuits acting on n qubits and whose depth is at most n . Specifically, for a NISQ computer if the zero noise extrapolated Heavy Output Probability expectation values pass (with high confidence) the $\frac{2}{3}$ threshold, as defined in the original quantum volume benchmark, for n by n square quantum volume circuits, then we say that the device has an *effective quantum volume* of n . We expect, in general, the *effective quantum volume* number to be greater than the original quantum volume benchmark number for NISQ devices. The effective quantum volume protocol [18] can be scaled to large system sizes; however there are two primary limitations of its scalability. The first comes from the original quantum volume protocol, which is that we need to classically compute the full statevector of the random quantum circuits – and this becomes prohibitive even for HPC clusters at qubit counts greater than ~ 32 . The second limitation comes from the inherent noise level in the quantum computer being benchmarked; this error rate is the limiting factor for arbitrarily increasing, and passing the test for, the *effective quantum volume* size.

In this study we experimentally demonstrate increased effective quantum volume compared to the vendor-benchmarked QV values, using ZNE, on four IBM Quantum devices. The results are summarized in Table 1².

2 Methods

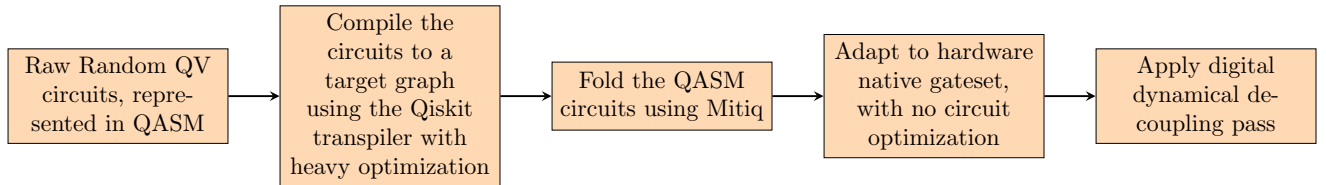


Figure 2: High-level workflow of the methodology for estimating the effective quantum volume with zero-noise extrapolation. First, the raw uncomputed QV circuits are initially defined with arbitrary connectivity and basis gates U3 and CX. Second, the Qiskit transpiler is run on the target hardware subgraph with `optimization_level=3`. Third, the QASM circuits are folded via either global or local folding with Mitiq. Fourth, the native gateset is adapted to RZ, CX, SX, and X using the Qiskit transpiler with `optimization_level=0`. Fifth, an optional step, X-X digital dynamical decoupling is applied.

The QV ZNE protocol steps that we use for this study are outlined at a high level in Figure 2.

2.1 Generation of quantum volume circuits

The first step in executing the ZNE QV benchmark protocol is to construct raw, random, Quantum Volume circuits of the form shown in Figure 1. In this case, the raw QV circuits are generated using Qiskit [29] and are defined with arbitrary connectivity and native gates of CX (e.g. CNOT) and u3 (e.g. arbitrary single qubit rotation). Next, the raw QV circuits are transpiled using a single call to the Qiskit [29] transpiler (`optimization_level` set to the maximum of 3) to target a specific device, thereby requiring a specific qubit connectivity and a specific native gateset [4]. In this case, the native gateset is CX, RZ, X, SX. The target connectivity is a subgraph of a heavy-hex [30] hardware graph. In order to reduce compilation times, each of the possible subgraph isomorphisms for a given size n are compiled to (for all QV circuits). These possible isomorphic subgraphs for $n = 6$ and $n = 7$ are shown in Figures 3 and 4. This compilation method allows us to compile the QV circuits to a small number of different connectivities, and then map those compiled circuits to any part of the quantum device chip without the

²Note that since these experiments were executed, both `ibm_geneva` and `ibmq_toronto` have been decommissioned

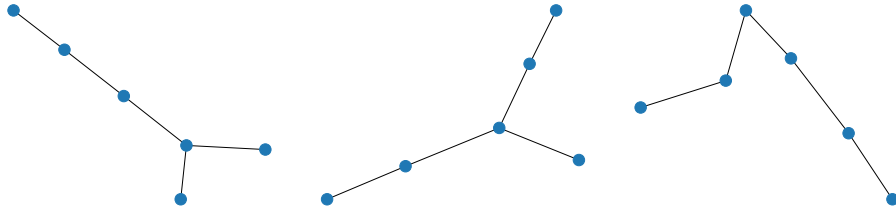


Figure 3: All subgraph isomorphisms for $n = 6$ on the heavy-hex connectivity.

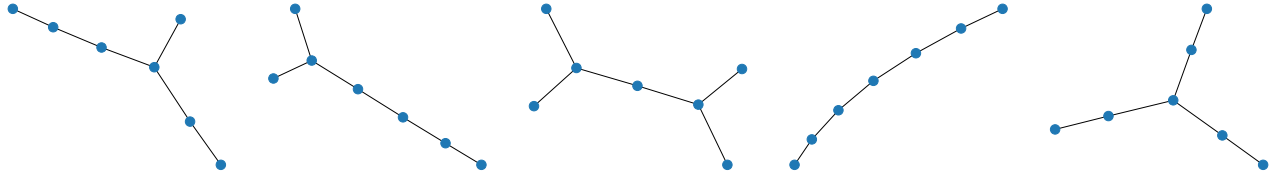


Figure 4: All subgraph isomorphisms for $n = 7$ on heavy hex connectivity.

need to re-compile all of the QV circuits. This reduces the total (classical) compute time required for executing these experiments. Furthermore, by choosing specific layouts to compile the circuits to, this allows us to control specifically which parts of the hardware graph are used, and then enumerate over different parts of the chip. Lastly, since the compiled circuits for a single sub-graph isomorphism are the same no matter which isomorphic subgraph they are mapped to on the device, this allows the results to be directly comparable (whereas, the stochasticity of the quantum circuit compiler can result in slightly different compiled circuits if the compilation is re-executed for every target subgraph). When executing the circuits on the IBM Quantum devices, we enumerated over multiple hardware subgraphs, similar to ref. [4], but did not enumerate over *every possible* hardware subgraph as in ref. [4] because the ZNE protocol adds significant time and computation overhead to the already time-intensive Quantum Volume protocol. However, we did enumerate over as many hardware subgraphs as could be executed, given the constraints of queue times and server-side device errors.

2.2 Noise scaling by unitary folding

Once each circuit was compiled to the fixed part of the hardware, the circuits were then *folded* (i.e. compiled with more gates) using the Mitiq software package [24] so as to scale up the level of noise. Specifically, we used two different techniques known as global folding and random local folding, which are both designed to effectively increase the error rate when running the circuits on a noisy quantum processor. [10, 19–21, 23].

Let $U = L_d L_{d-1} \dots L_1$ be the a decomposition of the full circuit as a sequence of d layers and let $W = L_d L_{d-1} \dots L_{d-k-1}$ be the fraction of the full circuit corresponding to the last k layers. Global folding transforms the input circuit U as follows ³

$$U \rightarrow UW^\dagger W, \quad (3)$$

where k depends on the noise scale factor λ . More precisely, k is chosen as the integer part of $d(\lambda - 1)/2$, such that the new depth $d' = d + 2k$ is approximately scaled as λd . For the local folding technique, the transformation is similar but, instead of being applied at the circuit level, it is applied at the gate level as follows:

$$G \rightarrow GG^\dagger G, \quad (4)$$

where G represents an individual gate of the input circuit U . Let t be the number of noisy gates in the input circuit U and let k be the integer part of $t(\lambda - 1)/2$. In random local folding, the transformation in Eq. (4) is applied to k noisy gates sampled at random without replacement from the full set of the t noisy gates in the circuit, such that the new circuit has $\approx \lambda t$ noisy gates. More precisely, in this work we assume that the main source of errors is due to CNOT gates and, therefore, we apply local folding by only sampling from the set of t CNOT gates that are present in each quantum volume circuit. Moreover, to apply local folding more uniformly along the circuit,

³Here and everywhere in this work, we assume $\lambda \leq 3$. More details about how to extend global and local folding for $\lambda > 3$ can be found in Refs. [21, 24].

the procedure was repeated 10 times for each noise scale factor λ (each time sampling a new random subset of k CNOT gates), and the measured expectation values averaged to a single noise-scaled value.

For all the experiments reported in this work, the chosen noise scale factors are $\lambda = \{1, 1.2, 1.5, 1.8, 2\}$ (or a subset of them). These scale factors are quite small compared to those used in the quantum volume experiments of Ref. [18]. Using small scale factors is a deliberate choice since in this study we consider larger circuits for which large noise scale factors are not appropriate. Indeed the application of strong noise scaling (e.g. with $\lambda \geq 3$) would amplify errors too much when applied on a base circuit which is already very noisy.

For storing the circuits at this stage, OpenQASM is used [31]. After the noise scaling process, the circuits are again transpiled to map them to the correct gateset, but no optimization is applied such that the intended effect of unitary folding is not cancelled by circuit optimization passes.

2.3 Insertion of dynamical decoupling sequences

Next, dynamical decoupling X-X sequences are inserted into the compiled circuits using the `PadDynamicalDecoupling` [33] class of the Qiskit [29] library. Dynamical Decoupling is an open loop quantum control error suppression technique for mitigating decoherence on idle qubits [26, 34–39], which has been shown to help increase measured quantum volumes in previous experiments [3]. When the dynamical decoupling sequences are inserted into the circuit, the resulting scheduled circuit (based on the IBM Quantum device gate timings) is scheduled according to the ALAP (as late as possible) algorithm. This dynamical decoupling step is optional, but the intention is to increase the measured heavy output probability of the QV circuits (both with and without noise scaling).

Figure 5 shows two examples of ZNE folded QV circuits represented as circuit timelines, with pairs of Pauli X gates (X-X sequence) dynamical decoupling sequences inserted. Note that the `rz` gates in Figure 5 are *virtual gates* [32], meaning that they are performed at the software level and therefore have an error rate of 0 and a time cost within the circuit of 0.

2.4 Execution on hardware and extrapolation to the zero-noise limit

Once the circuits have been compiled, all of the circuits are submitted to the target backend. Importantly, the number of shots used for the different types of circuits differs—in particular the aim is for the original unscaled (i.e. $\lambda = 1$) QV circuits to sample more shots compared to the folded QV circuits. The original QV circuits are all executed with 10000 shots per circuit. The global folded QV circuits are executed with 1000 shots per (folded) circuit, and the local randomly folded QV circuits are executed with 100 shots per (folded) circuit. Finally, the quantum circuit measurements are converted into heavy output proportion (HOP) statistics, and the zero noise extrapolations can be performed on measured HOP data for each individual circuit using Mitiq [24].

In particular, linear [21, 23, 24] extrapolation is used, since Richardson extrapolation can lead to extrapolation instability for a limited number of shots. This extrapolation is very computationally efficient. For consistency, and in order to obtain a very robust sampleset, 1000 random QV circuits are generated and executed for each target backend (and for each target qubit subset of the chip). This number of shots ensures that the resulting extrapolations, no matter what specific scale factors are used, will utilize fewer shots compared to the original (unmodified) QV circuit which is always sampled with 10000 shots. Therefore, the total overhead that ZNE incurs for this procedure is given by the number of scale factors (not equal to 1) that are used, which is at most 4, and each folded-circuit scale factor requires a total of 1,000 shots to be readout on the device.

With the aim of executing the ZNE QV circuits on the target IBM Quantum devices, the circuits were split into groups of 250 circuits to be executed together as *jobs* since the maximum number of circuits per job on the target IBM Quantum devices was 300. This means there were 4 jobs executed for each group of 1000 QV circuits, which was then repeated for each of the circuit folding scale factors and the two circuit folding methods. This is important because it is possible for the results to not have been executed together sequentially, and therefore there can be error drift. Additionally, when there are backend device errors and the jobs need to be re-executed, those results would be executed at a different time as well. The results are then post-processed using linear extrapolation, which is very computationally efficient. The ZNE QV results shown in Section 3 are presented as cumulative heavy output probability plots, which are a convenient method of visually representing quantum volume results [3, 4].

3 Results

The effective quantum volume results are reported in the form of cumulative heavy output probability plots. The 2σ bound on the error mitigated HOP values are measured using bootstrapped re-sampling [5, 18] on the (cumulative)

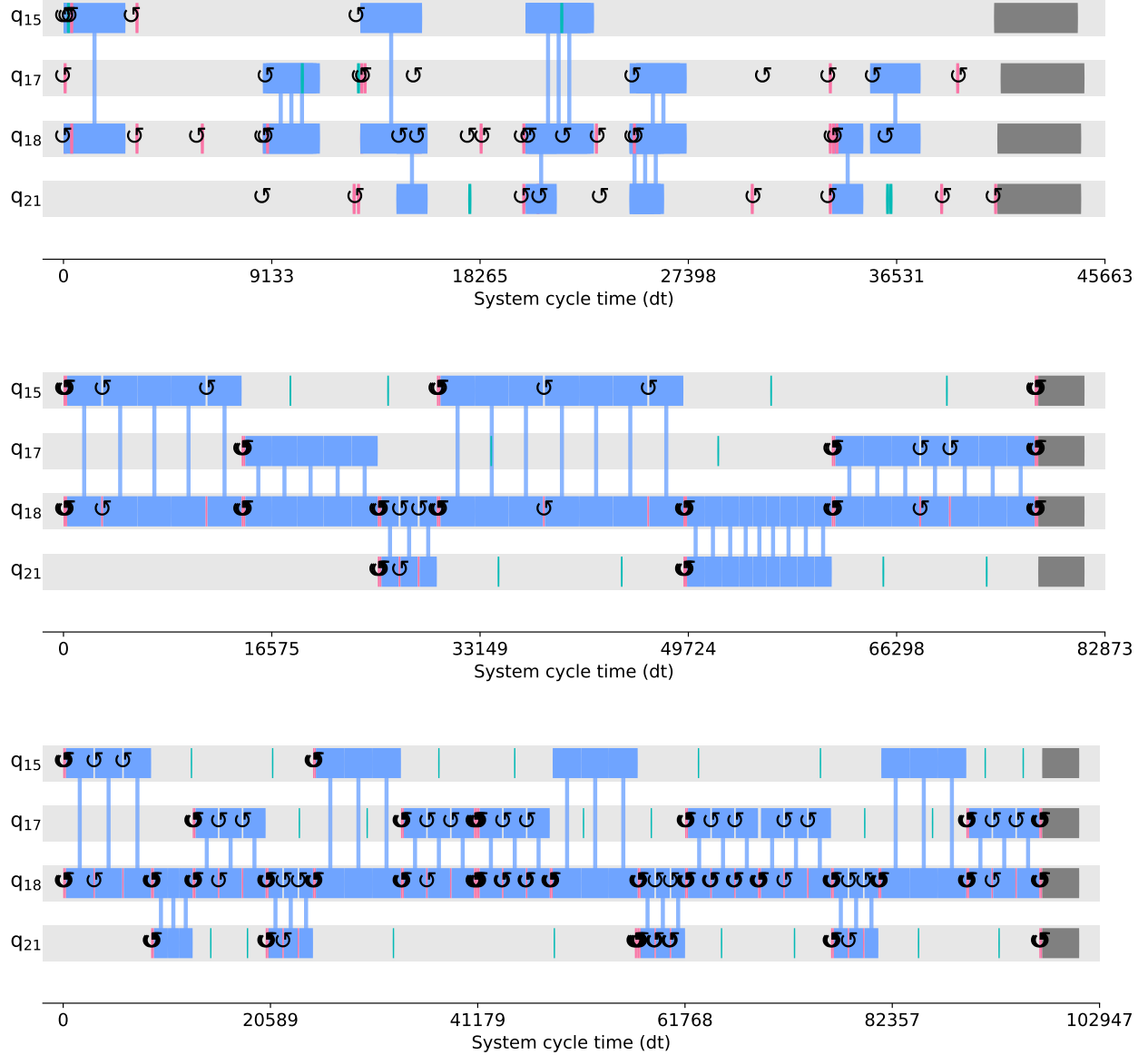


Figure 5: Three Qiskit [29] timeline diagrams of a single $n = 4$ Quantum Volume circuit compiled to a subgraph of a heavy hex graph, specifically qubits 15, 18, 17, 21 of `ibm_hanoi`. **Top** diagram shows the compiled QV circuit, with optimizations applied. The **middle** and **bottom** circuit diagrams are the same compiled circuit with a digital ZNE circuit folding scale factor of $\lambda = 2$ having been applied. Local CNOT circuit folding is used for the **middle** plot, and global circuit folding is used in the **bottom** plot. Digital dynamical decoupling sequences of X-X Pauli gates are inserted into all of the the circuits, and the all circuits are scheduled using the ALAP circuit scheduler. The RZ gates are virtual gates [32], and are represented here by circular black arrow markers. The X gates are shown as green vertical lines, which are very thin because the single qubit gate operations take a small amount of time. Similarly, the short duration single qubit SX gate is represented as vertical red lines. The CX (e.g. CNOT) gates are drawn as vertical blue connections between adjacent qubits in the hardware graph. The dark grey segments at the end of each qubit line denote the measurement of the 4 qubits. The uncompiled QV circuit contained 24 CX gates, which was then turned into 18 CX gates when optimized using the Qiskit transpiler and adapted to the heavy hex graph structure (shown in the top circuit diagram). ZNE global folding (bottom) with $\lambda = 2$ generated a circuit invariant with 36 CX gates, and local random folding (middle) with $\lambda = 2$ generated a circuit with 42 CX gates.

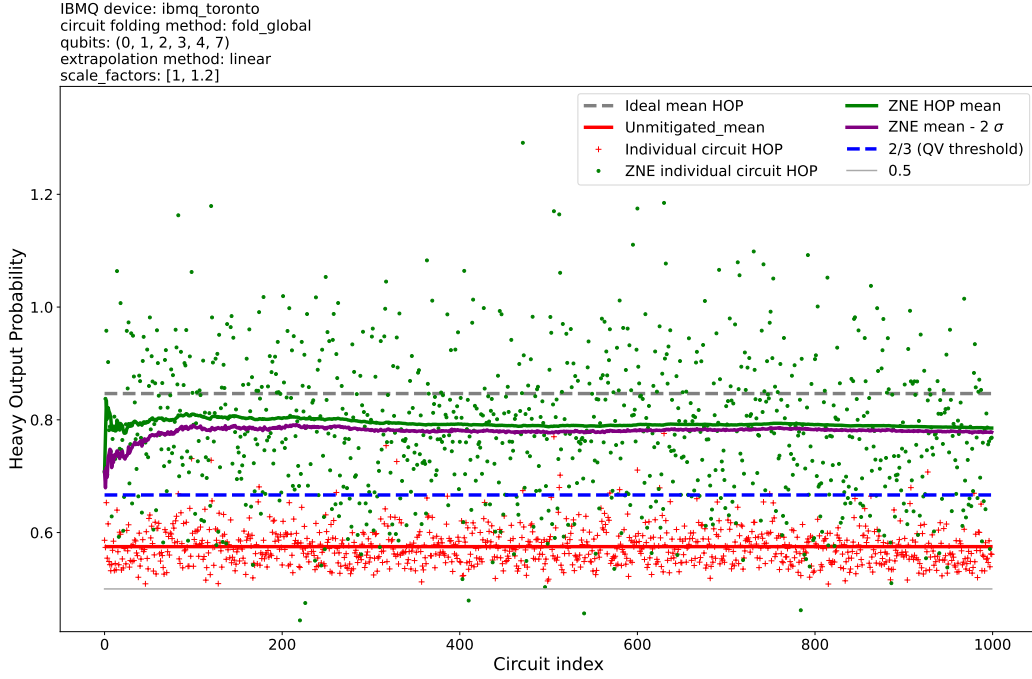


Figure 6: Cumulative HOP plot showing `ibmq_toronto` passing the effective quantum volume of $n = 6$ for ZNE using unitary circuit folding. The HOP metric is defined to be within $[0, 1]$, note therefore that any ZNE HOP points above 1 correspond to non-physical extrapolations due to shot noise.

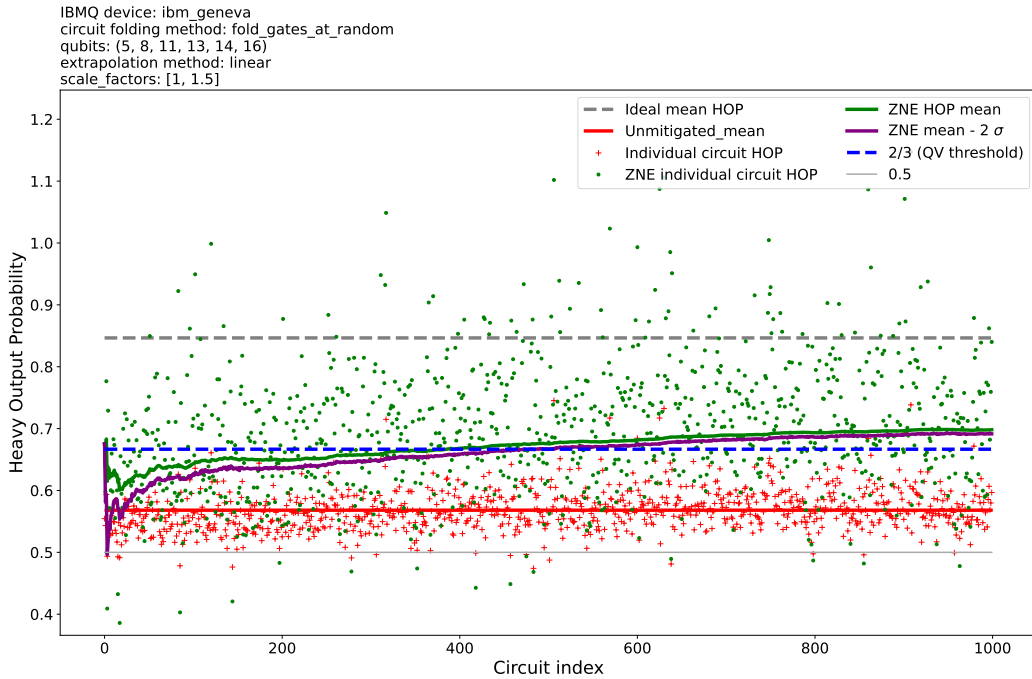


Figure 7: Cumulative HOP plot showing `ibmq_geneva` passing the effective quantum volume of $n = 6$ for ZNE using local random circuit folding.

ZNE HOP points. The re-sampling is performed 100 times for the full HOP vector size (of 1000), with replacement, and then the standard deviation σ is computed. Figures 6 and 7 show `ibmq_toronto` and `ibmq_geneva` both passing an effective quantum volume of $n = 6$, respectively. Figures 8 and 9 show `ibmq_auckland` and `ibmq_hanoi` passing an

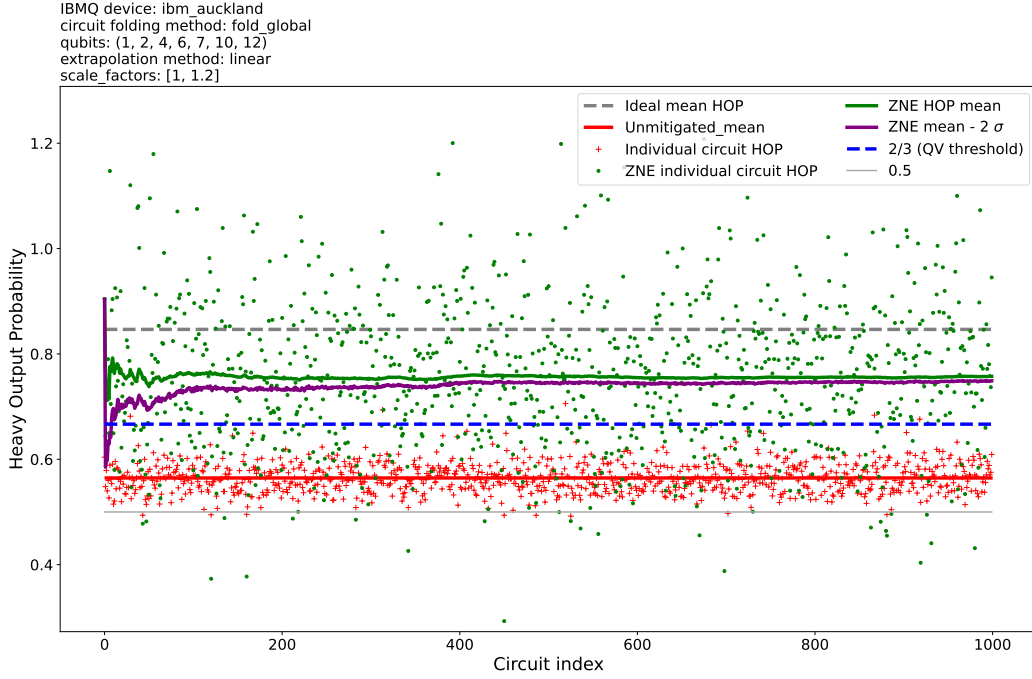


Figure 8: Cumulative HOP plot showing `ibm_auckland` passing the effective quantum volume of $n = 7$ for ZNE using global circuit folding.

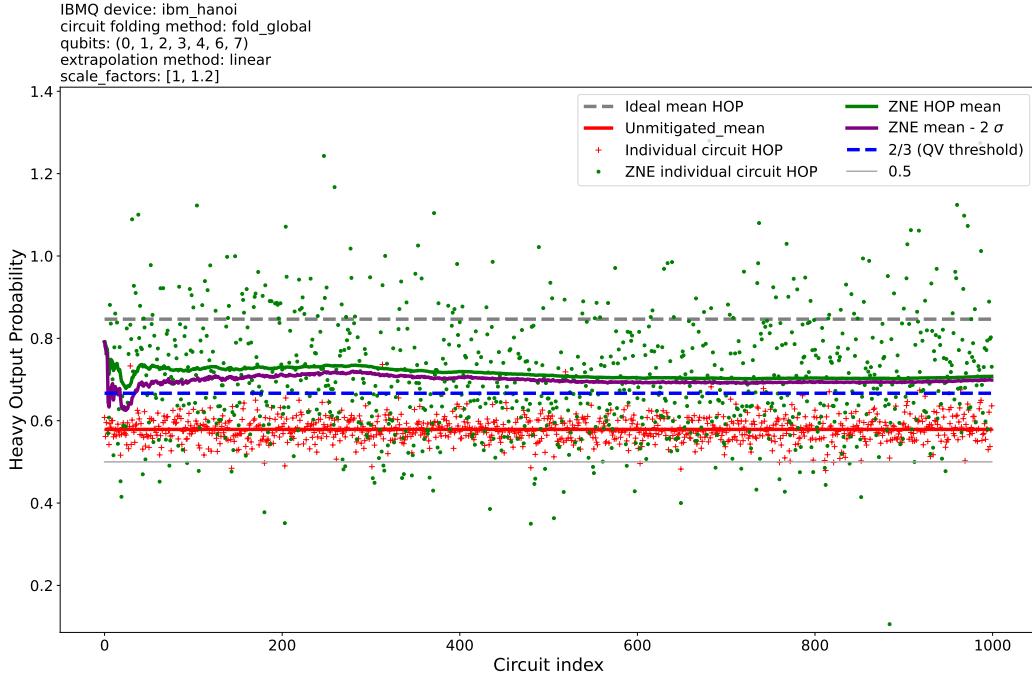


Figure 9: Cumulative HOP plot showing `ibm_hanoi` passing the effective quantum volume of $n = 7$ for ZNE using unitary circuit folding.

effective quantum volume of $n = 7$, respectively. Each of the four figures, show an effective quantum volume test averaged over 1000 circuits passing the $\frac{2}{3}$ threshold. Therefore, each plot show that the effective Quantum Volume is increased by ZNE beyond the vendor-measured quantum volume, a fact that was never been demonstrated to date on a NISQ computer [18]. It is important to consider however, that although effective quantum volume is defined,

as was the original Quantum Volume metric [1], as the *largest* such measurement that can be made on a device, it is the case that a wide range of QV results can be observed on a fixed NISQ computer [4] depending on the choice of qubits and error mitigation method. In the interest of examining in what cases we observed effective quantum volume experiments failing to pass the QV threshold, we report the associated negative results in Appendices A, C. Moreover, Appendix B briefly gives some additional cumulative ZNE HOP plots to illustrate how similar the two ZNE circuit folding methods are. For all of the experiments we present, linear extrapolation is used because it gives the lowest extrapolation variability (specifically, extrapolated values that are more typically consistent with the physical HOP range). The better stability of linear extrapolation compared to high-order polynomial fittings (e.g. Richardson extrapolation) is a known phenomenon that can be explained in terms of under-fitting vs over-fitting of the noise scaled expectation values (see Ref. [21] for more details on the extrapolation error).

Figure 8 shows a single cumulative HOP plot for a case where `ibm_auckland` passes the effective quantum protocol of size $n = 7$ when executing on qubits 1, 2, 4, 6, 7, 10, 12. The ZNE protocol in this case used scale factors of 1 and 1.2, global circuit folding method, and linear extrapolation on the noise-scaled ZNE data points. Figure 8 shows clearly that the original QV circuits were sampled below $\frac{2}{3}$, but above 0.5, and that the error mitigated HOP values were measurably above the $\frac{2}{3}$ threshold but not above the ideal mean HOP. Figure 8 also shows that some of ZNE HOP points were measured to be above 1, which is possibly due to the relatively small shot count, and the fractional scale factors that are used resulting in some extrapolation instability. Extrapolation instability, in particular, extrapolated HOP which is much greater than 1, is seen in Figures 6, 7, 8, and 9.

Figure 10 shows the full distribution of measured ZNE HOP values (specifically the mean ZNE HOP values across each set of 1000 QV circuits) across all of the experiments that were executed. This distribution of ZNE HOP values includes the different linear extrapolations for $\lambda = 0$ across the possible combinations of the scale factors that were used between $\lambda = 1$ and $\lambda = 1.2, 1.5, 1.8, 2$. These distributions are not comprehensive for the entire hardware graph of the four IBMQ chips, but instead represent only a random subset of the possible hardware graphs that could be chosen to compile to. As has been observed for Quantum Volume test enumeration [4], Figure 10 shows that on average the performance is worse than the best performance cases. Lastly, note that the best mean ZNE HOP does not reach above the ideal mean HOP of ≈ 0.85 [1, 7], showing that when this procedure is averaged over many instances the quantum error mitigation results in measures that are consistent with the physical measure of HOP.

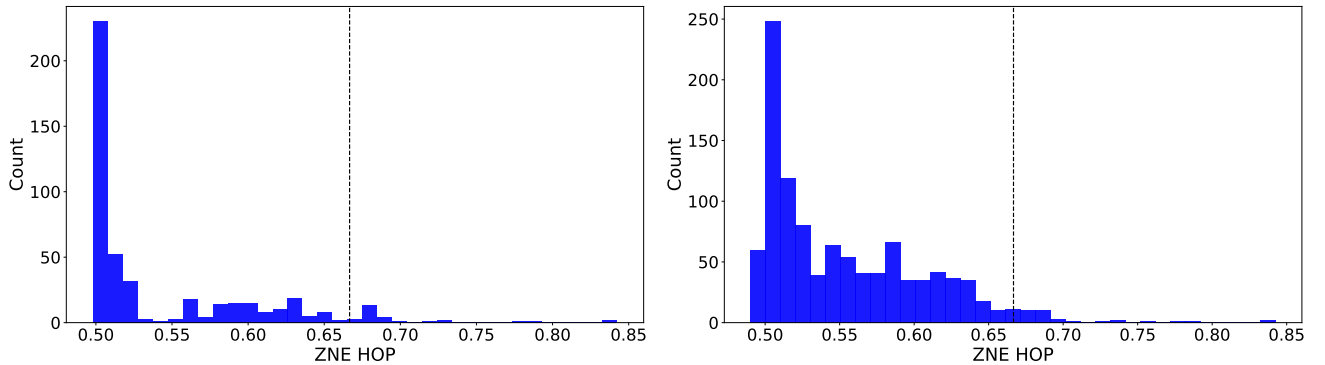


Figure 10: Distribution of all mean $\lambda = 0$ Zero Noise Extrapolated HOP values, across all of the different noise scale extrapolations, for $n = 6$ (left), run on `ibmq_toronto` and `ibmq_geneva` and $n = 7$ (right) run on `ibm_hanoi` and `ibm_auckland`. The $\frac{2}{3}$ threshold is marked with the vertical dashed black line.

An important aspect of zero noise extrapolation is that statistical noise is increased by the extrapolation process. For example, if there are too few measurement shots used for computing an expectation value, or if there is an error rate drift on the quantum computer during the period of time the circuits are executed, the resulting extrapolations may end up outside the expected physical range. For a noiseless quantum computer, individual circuits may have a heavy output probability that is slightly larger or smaller than $\frac{1+\ln(2)}{2} \approx 0.85$ [4, 5] (especially if n is small). In our experiments, we observe that ZNE sometimes amplifies such deviations beyond the theoretical upper bound and, for a few cases, the extrapolated HOP is even larger than 1. The amplification of statistical fluctuations is a known aspect of ZNE and is expected for individual circuits due to the very small number of sample (100 shots). However, at least when using linear fit (more stable with respect to Richardson extrapolation), when averaging over the 1000 random circuit of the QV protocol, we always obtained a mean HOP within the physical range.

Another important note about ZNE, and specifically ZNE QV, is that if there is no signal for small-scale factors,

and especially, $\lambda = 1$ then the extrapolation simply does not work. In the case of Quantum Volume, if the measured heavy output probabilities across the scale factors are all around 0.5 (meaning the computation has completely decohered) then the extrapolation cannot work. Furthermore, even if some of the larger scale factors converge to a HOP of 0.5 then extrapolations based on those results may also end up skewed—this in part motivated the use of scale factors that were ≤ 2 . For this reason, our experiments target several different parts of the chips since the error rates on each chip can vary significantly [4], so as to quantify a *distribution* of the possible ZNE QV performance on the target QPUs. This concept of applying the same algorithm to an ensemble of different parts of the quantum computer hardware has been used in other contexts [40].

4 Discussion and Conclusion

We have demonstrated an increase in the measured effective quantum volume over the vendor-benchmarked quantum volume, on four IBM Quantum devices `ibm_geneva`, `ibm_auckland`, `ibm_hanoi`, `ibmq_toronto`. These results show that error mitigation can be scaled to larger QV circuit sizes than have been previously measured in the effective quantum volume protocol [18]. We have also shown examples where quantum error mitigation, in particular ZNE, is unable to extrapolate the computation to the zero noise limit—this occurs when the circuit that is executed experiences too much decoherence and there is no signal that can be extrapolated. Because the QV ZNE protocol fails to perform a meaningful zero noise extrapolation when all of the computations have decohered because of errors in the computation, effective quantum volume serves as a meaningful quantification of how effective quantum error mitigation can be for near-term quantum processors. In particular, although the effective quantum volume can be greater than the vendor-measured quantum volume (as has been demonstrated in this study), there is a hard limit where the extrapolation will not work (when the mean heavy output probability converges to 0.5) which is governed by the error rate of the quantum computer. Therefore, as with the original quantum volume test, effective quantum volume is a difficult test for near-term quantum processors to pass, and importantly as a benchmark employs the full stack of software, hardware, and algorithms involved with utilizing quantum error mitigation.

Probing how other quantum error mitigation techniques, in particular, probabilistic error cancellation (PEC) [23, 41, 42], perform when applied to effective Quantum Volume is an open question. In general, other quantum error suppression mechanisms such as Pauli twirling (also known as randomized compiling) [43–46] and variants of dynamical decoupling should continue to be analyzed for their effectiveness at improving NISQ computation quality, and therefore also how they impact full stack benchmarks such as Quantum Volume.

5 Acknowledgments

This work was supported by the U.S. Department of Energy through the Los Alamos National Laboratory. Los Alamos National Laboratory is operated by Triad National Security, LLC, for the National Nuclear Security Administration of U.S. Department of Energy (Contract No. 89233218CNA000001). The research presented in this article was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20220656ER, and by the NNSA’s Advanced Simulation and Computing Beyond Moore’s Law Program at Los Alamos National Laboratory. This research used resources provided by the Los Alamos National Laboratory Institutional Computing Program, which is supported by the U.S. Department of Energy National Nuclear Security Administration under Contract No. 89233218CNA000001. This work has been assigned LANL report number LA-UR-23-26260. 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.

This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing under Award Numbers DE-SC0020266 and DESC0020316 as well as by IBM under Sponsored Research Agreement No. W1975810. AM acknowledges support from the PNRR MUR project PE0000023-NQSTI (Italy).

The figures in this study are generated using a combination of matplotlib [47, 48], networkx [49], Quantikz [8], and Qiskit [29] in Python 3.

A ZNE unable to extrapolate if the computation contains too many errors

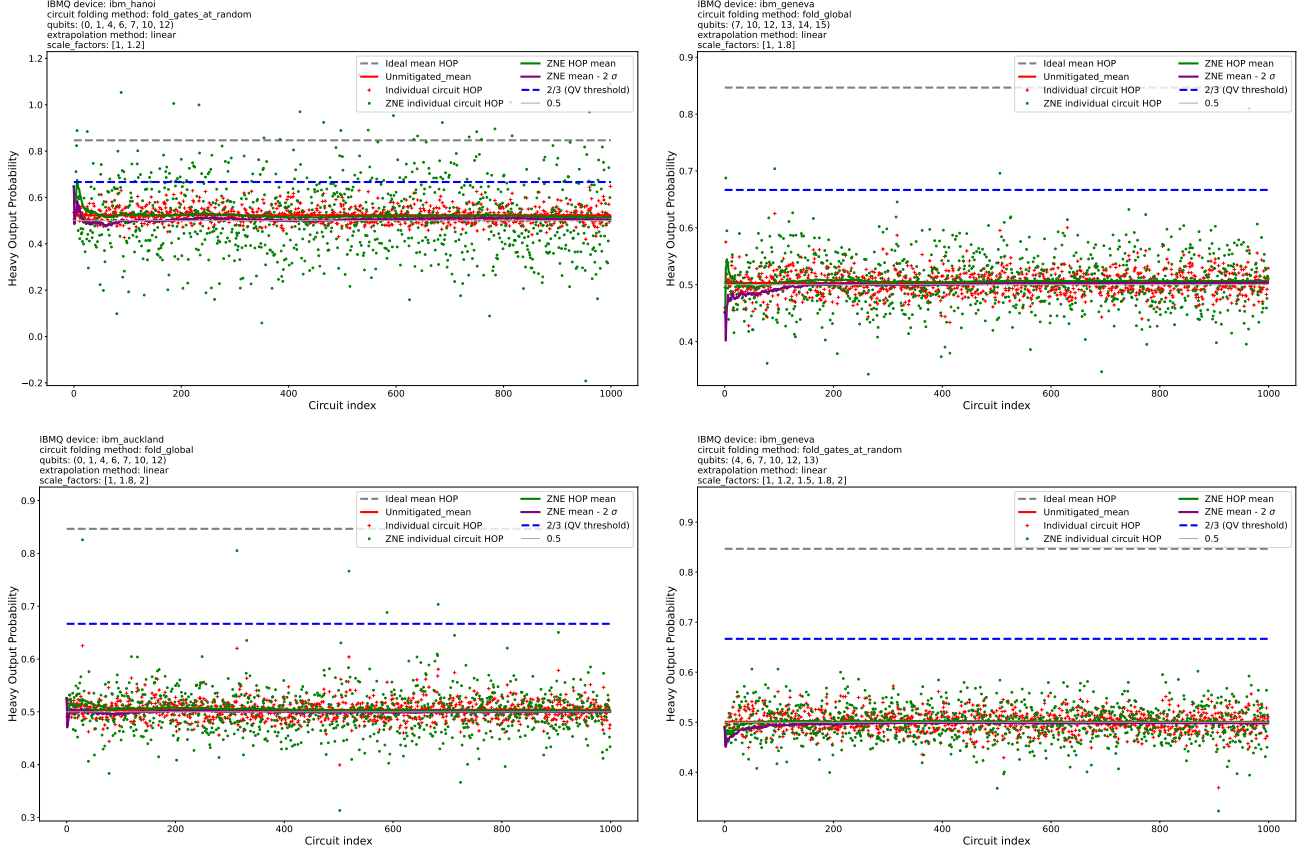


Figure 11: Four example cumulative HOP plots which show instances where the total errors encountered within the circuits caused the mean HOP to converge to 0.5, which is the HOP that is expected when all fidelity in the computation is lost. This high noise convergence to 0.5 HOP has been observed before [4], but here it is notable because it means that any extrapolation of expectation values would not work, since the extrapolations would be for a flat expectation value landscape (of 0.5 on average). Experiment parameters are shown in the plot titles.

Figure 11 shows examples from the set of experiments we executed where the original unmodified QV circuits (e.g. $\lambda = 1$) were sampled with a mean HOP of 0.5. The result of this is that any extrapolation for increased error can not work. This occurs in general when the computation reaches the maximal amount of noise, or error. This shows the limits of ZNE, for the specific case of Quantum Volume. In particular, this shows that such near term quantum error mitigation algorithms require at least some amount of signal to be present in order for the error mitigation to work.

B Compare global and local circuit folding

Both global and local folding produce the same intended effect, which is to increase the effective noise of a circuit by intentionally increasing the number of gates. However, the circuit implementations of these two folding methods are not identical, and can be compiled to devices in different ways. In particular the way the fractional scale factors are implemented is different for these two methods. Generally, sampling more random instances of locally folded circuits with fractional scale factors is useful because it can average out biases from increased noise in local parts of the circuit. With global folding instead the process is deterministic but only folds part of the circuit unitary, which means the results could have biased noise. Given these differences, a relevant question is whether the two methods performed similarly in the effective quantum volume experiments considered in this work. Overall, the two methods did perform comparably. Figure 12 shows some side-by-side comparisons of the measured HOP extrapolation plots

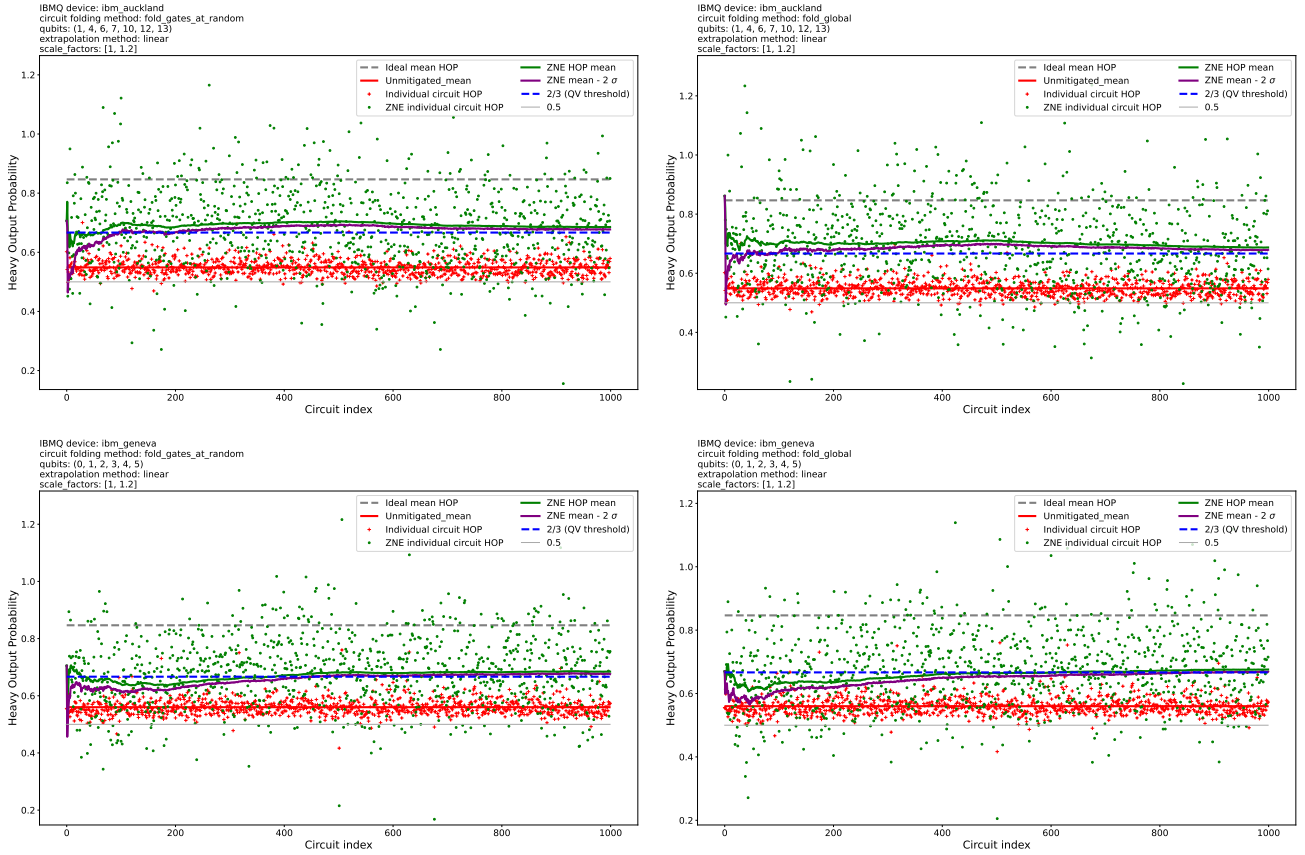


Figure 12: Comparison of global folding (right column) and local random circuit folding (left column) HOP results with all other parameters being held constant. Experiment parameters are shown in the plot titles.

which show that both methods did perform similarly. Specifically, these plots show two instances of identical parameter settings (same IBM Quantum device, qubits, layout, scale factors, and extrapolation algorithm), where only the choice of circuit folding method was changed. Although it is not possible to perfectly control the time within which these circuits were executed, these circuits were queued in sequence with each other, which means that time was controlled for as well as much as it can be on an IBM Quantum system.

C ZNE QV not passing threshold

Like what was observed on numerous IBM Quantum devices in ref. [4], for this set of ZNE QV tests evaluated in this study, many of the settings (in particular, the parts of the chip that were executed on) had too high error rates, even with ZNE being able to extrapolate meaningful expectation values that had a gradient (as opposed to the examples in Figure 11) often times the effective quantum volume test would fail. Examples of this are shown in Figure 13 in the form of cumulative HOP plots. Note that in the top right hand plot the threshold is only barely not passed, whereas in the other plots the mean ZNE HOP distribution is more clearly not passed. This shows that, as with previous Quantum Volume studies that sample a range of device settings and qubits [4], there is a distribution of performances, and the utilization of ZNE does not uniformly automatically make the measured results better. Near term quantum error mitigation does require tuning the of the experimental settings, and is still dependent on what errors are encountered on the device.

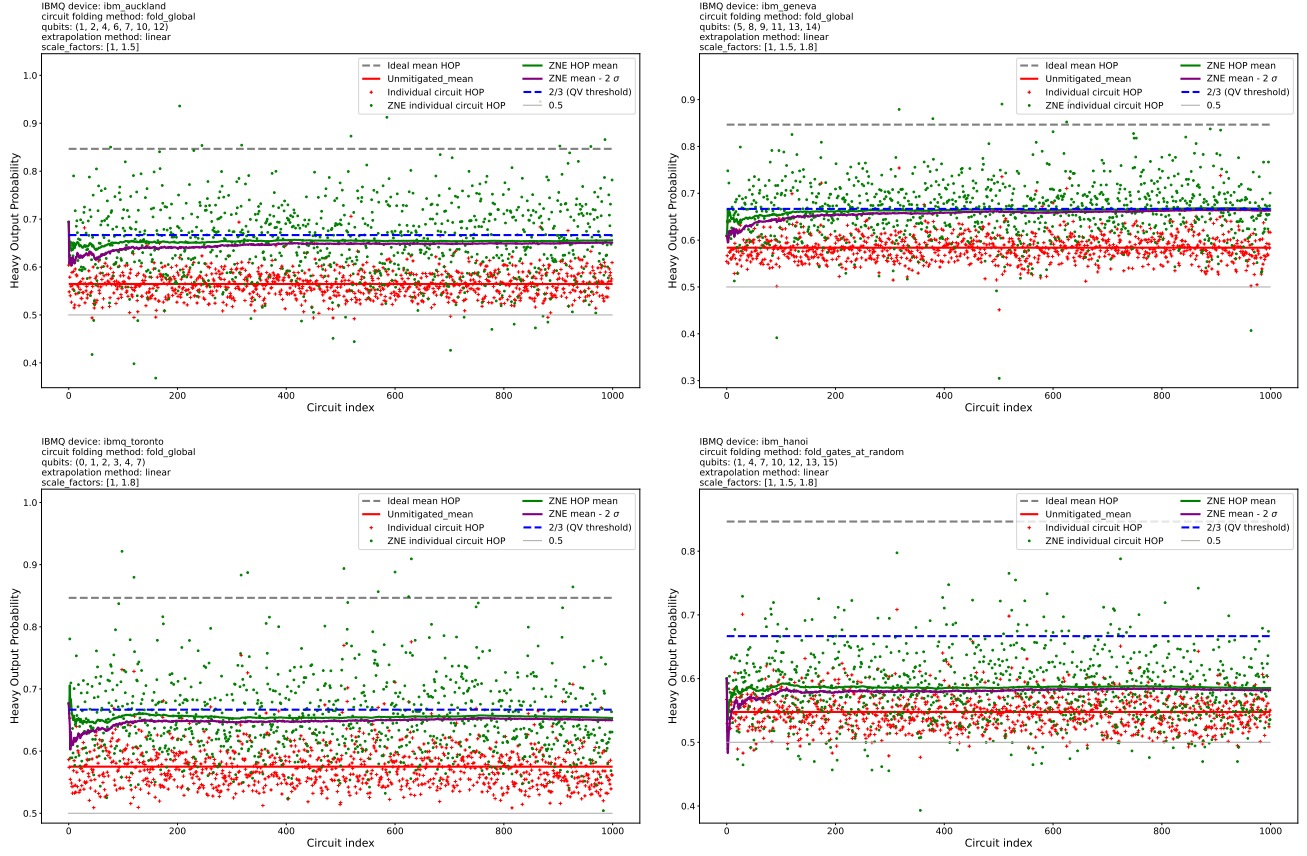


Figure 13: Example plots of the four IBM Quantum devices failing to pass the $\frac{2}{3}$ threshold with high confidence for the ZNE QV protocol under various ZNE settings and IBM Quantum compilation qubit subgraphs. Experiment parameters are shown in the plot titles.

References

- [1] Andrew W. Cross et al. “Validating quantum computers using randomized model circuits”. In: *Phys. Rev. A* 100 (3 2019), p. 032328. DOI: 10.1103/PhysRevA.100.032328. URL: <https://link.aps.org/doi/10.1103/PhysRevA.100.032328>.
- [2] Nikolaž Moll et al. “Quantum optimization using variational algorithms on near-term quantum devices”. In: *Quantum Science and Technology* 3.3 (2018), p. 030503. DOI: 10.1088/2058-9565/aab822.
- [3] Petar Jurcevic et al. “Demonstration of quantum volume 64 on a superconducting quantum computing system”. In: *Quantum Science and Technology* 6.2 (2021), p. 025020. DOI: 10.1088/2058-9565/abe519.
- [4] Elijah Pelofske, Andreas Bärttschi, and Stephan Eidenbenz. “Quantum Volume in Practice: What Users Can Expect from NISQ Devices”. In: *IEEE Transactions on Quantum Engineering* 3 (2022), pp. 1–19. DOI: 10.1109/TQE.2022.3184764.
- [5] Charles H. Baldwin et al. “Re-examining the quantum volume test: Ideal distributions, compiler optimizations, confidence intervals, and scalable resource estimations”. In: *Quantum* 6 (2022), p. 707. ISSN: 2521-327X. DOI: 10.22331/q-2022-05-09-707. URL: <https://doi.org/10.22331/q-2022-05-09-707>.
- [6] John Preskill. “Quantum computing in the NISQ era and beyond”. In: *Quantum* 2 (2018), p. 79. DOI: 10.22331/q-2018-08-06-79. URL: <https://doi.org/10.22331/q-2018-08-06-79>.
- [7] Scott Aaronson and Lijie Chen. “Complexity-theoretic foundations of quantum supremacy experiments”. In: *32nd Computational Complexity Conference CCC’17*. 2017, pp. 1–67. DOI: 10.5555/3135595.3135617. arXiv: 1612.05903.
- [8] Alastair Kay. *Quantikz: A TikZ library to typeset quantum circuit diagrams*. Tutorial on the Quantikz package. DOI: 10.17637/rh.7000520. arXiv: 1809.03842.
- [9] Zhenyu Cai et al. *Quantum error mitigation*. 2022. DOI: 10.48550/ARXIV.2210.00921. URL: <https://arxiv.org/abs/2210.00921>.
- [10] Abhinav Kandala et al. “Error mitigation extends the computational reach of a noisy quantum processor”. In: *Nature* 567.7749 (2019), pp. 491–495. DOI: 10.1038/s41586-019-1040-7. URL: <https://doi.org/10.1038/s41586-019-1040-7>.
- [11] Abdullah Ash Saki et al. *Hypothesis testing for error mitigation: How to evaluate error mitigation*. 2023. DOI: 10.48550/ARXIV.2301.02690. URL: <https://arxiv.org/abs/2301.02690>.
- [12] Suguru Endo, Simon C. Benjamin, and Ying Li. “Practical quantum error mitigation for near-future applications”. In: *Phys. Rev. X* 8 (3 2018), p. 031027. DOI: 10.1103/PhysRevX.8.031027. URL: <https://link.aps.org/doi/10.1103/PhysRevX.8.031027>.
- [13] Armands Strikis et al. “Learning-based quantum error mitigation”. In: *PRX Quantum* 2 (4 2021), p. 040330. DOI: 10.1103/PRXQuantum.2.040330. URL: <https://link.aps.org/doi/10.1103/PRXQuantum.2.040330>.
- [14] C. Ryan-Anderson et al. *Implementing fault-tolerant entangling gates on the five-qubit code and the color code*. 2022. DOI: 10.48550/ARXIV.2208.01863. URL: <https://arxiv.org/abs/2208.01863>.
- [15] Rajeev Acharya et al. *Suppressing quantum errors by scaling a surface code logical qubit*. 2022. DOI: 10.48550/ARXIV.2207.06431. URL: <https://arxiv.org/abs/2207.06431>.
- [16] Ryuji Takagi et al. “Fundamental limits of quantum error mitigation”. In: *npj Quantum Information* 8.1 (Sept. 2022). ISSN: 2056-6387. DOI: 10.1038/s41534-022-00618-z. URL: <http://dx.doi.org/10.1038/s41534-022-00618-z>.
- [17] Yihui Quek et al. *Exponentially tighter bounds on limitations of quantum error mitigation*. 2023. arXiv: 2210.11505 [quant-ph].
- [18] Ryan LaRose et al. *Error mitigation increases the effective quantum volume of quantum computers*. 2022. DOI: 10.48550/ARXIV.2203.05489. URL: <https://arxiv.org/abs/2203.05489>.
- [19] Andre He et al. “Zero-noise extrapolation for quantum-gate error mitigation with identity insertions”. In: *Phys. Rev. A* 102 (1 2020), p. 012426. DOI: 10.1103/PhysRevA.102.012426. URL: <https://link.aps.org/doi/10.1103/PhysRevA.102.012426>.
- [20] Ying Li and Simon C. Benjamin. “Efficient variational quantum simulator incorporating active error minimization”. In: *Phys. Rev. X* 7 (2 2017), p. 021050. DOI: 10.1103/PhysRevX.7.021050. URL: <https://link.aps.org/doi/10.1103/PhysRevX.7.021050>.
- [21] Tudor Giurgica-Tiron et al. “Digital zero noise extrapolation for quantum error mitigation”. In: *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2020. DOI: 10.1109/qce49297.2020.00045. URL: <https://doi.org/10.1109/qce49297.2020.00045>.
- [22] Kevin Schultz et al. “Impact of time-correlated noise on zero-noise extrapolation”. In: *Physical Review A* 106.5 (2022). DOI: 10.1103/physreva.106.052406. URL: <https://doi.org/10.1103/physreva.106.052406>.

- [23] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. “Error mitigation for short-depth quantum circuits”. In: *Physical Review Letters* 119.18 (2017). DOI: 10.1103/physrevlett.119.180509. URL: <https://doi.org/10.1103/physrevlett.119.180509>.
- [24] Ryan LaRose et al. “Mitiq: A software package for error mitigation on noisy quantum computers”. In: *Quantum* 6 (2022), p. 774. ISSN: 2521-327X. DOI: 10.22331/q-2022-08-11-774. URL: <https://doi.org/10.22331/q-2022-08-11-774>.
- [25] Vincent R. Pascuzzi et al. “Computationally efficient zero-noise extrapolation for quantum-gate-error mitigation”. In: *Phys. Rev. A* 105 (4 2022), p. 042406. DOI: 10.1103/PhysRevA.105.042406. URL: <https://link.aps.org/doi/10.1103/PhysRevA.105.042406>.
- [26] Youngseok Kim et al. “Scalable error mitigation for noisy quantum circuits produces competitive expectation values”. In: *Nature Physics* (2023). DOI: 10.1038/s41567-022-01914-3. URL: <https://doi.org/10.1038/s41567-022-01914-3>.
- [27] Minh C. Tran, Kunal Sharma, and Kristan Temme. *Locality and error mitigation of quantum circuits*. 2023. arXiv: 2303.06496 [quant-ph].
- [28] Youngseok Kim et al. “Evidence for the utility of quantum computing before fault tolerance”. In: *Nature* 618.7965 (2023), pp. 500–505. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06096-3. URL: <https://doi.org/10.1038/s41586-023-06096-3>.
- [29] Qiskit contributors. *Qiskit: An open-source framework for quantum computing*. 2023. DOI: 10.5281/zenodo.2573505.
- [30] Christopher Chamberland et al. “Topological and Subsystem Codes on Low-Degree Graphs with Flag Qubits”. In: *Phys. Rev. X* 10 (1 2020), p. 011022. DOI: 10.1103/PhysRevX.10.011022. URL: <https://link.aps.org/doi/10.1103/PhysRevX.10.011022>.
- [31] Andrew W. Cross et al. *Open quantum assembly language*. 2017. DOI: 10.48550/ARXIV.1707.03429. URL: <https://arxiv.org/abs/1707.03429>.
- [32] David C. McKay et al. “Efficient Z-gates for quantum computing”. In: *Physical Review A* 96.2 (2017). DOI: 10.1103/physreva.96.022330. URL: <https://doi.org/10.1103/physreva.96.022330>.
- [33] *Qiskit PadDynamicalDecoupling*. <https://qiskit.org/documentation/stubs/qiskit.transpiler.passes.PadDynamicalDecoupling.html>.
- [34] Lorenza Viola, Emanuel Knill, and Seth Lloyd. “Dynamical Decoupling of Open Quantum Systems”. In: *Phys. Rev. Lett.* 82 (12 1999), pp. 2417–2421. DOI: 10.1103/PhysRevLett.82.2417. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.82.2417>.
- [35] Lorenza Viola and Seth Lloyd. “Dynamical suppression of decoherence in two-state quantum systems”. In: *Phys. Rev. A* 58 (4 1998), pp. 2733–2744. DOI: 10.1103/PhysRevA.58.2733. URL: <https://link.aps.org/doi/10.1103/PhysRevA.58.2733>.
- [36] Jingfu Zhang et al. “Protected quantum computing: Interleaving gate operations with dynamical decoupling sequences”. In: *Phys. Rev. Lett.* 112 (5 2014), p. 050502. DOI: 10.1103/PhysRevLett.112.050502. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.112.050502>.
- [37] Siyuan Niu and Aida Todri-Sanial. “Effects of dynamical decoupling and pulse-level optimizations on IBM quantum computers”. In: *IEEE Transactions on Quantum Engineering* 3 (2022), pp. 1–10. DOI: 10.1109/TQE.2022.3203153.
- [38] Nic Ezzell et al. *Dynamical decoupling for superconducting qubits: a performance survey*. 2022. DOI: 10.48550/ARXIV.2207.03670. URL: <https://arxiv.org/abs/2207.03670>.
- [39] Arturo Mena López and Lian-Ao Wu. “Protectability of IBMQ qubits by dynamical decoupling technique”. In: *Symmetry* 15.1 (2023). ISSN: 2073-8994. DOI: 10.3390/sym15010062. URL: <https://www.mdpi.com/2073-8994/15/1/62>.
- [40] Swamit S. Tannu and Moinuddin Qureshi. “Ensemble of diverse mappings: Improving reliability of quantum computers by orchestrating dissimilar mistakes”. In: *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. MICRO ’52. Columbus, OH, USA: Association for Computing Machinery, 2019, 253–265. ISBN: 9781450369381. DOI: 10.1145/3352460.3358257. URL: <https://doi.org/10.1145/3352460.3358257>.
- [41] Shuaining Zhang et al. “Error-mitigated quantum gates exceeding physical fidelities in a trapped-ion system”. In: *Nature Communications* 11.1 (2020). DOI: 10.1038/s41467-020-14376-z. URL: <https://doi.org/10.1038/s41467-020-14376-z>.
- [42] Jinzhao Sun et al. “Mitigating realistic noise in practical noisy intermediate-scale quantum devices”. In: *Phys. Rev. Appl.* 15 (3 2021), p. 034026. DOI: 10.1103/PhysRevApplied.15.034026. URL: <https://link.aps.org/doi/10.1103/PhysRevApplied.15.034026>.

- [43] Joel J. Wallman and Joseph Emerson. “Noise tailoring for scalable quantum computation via randomized compiling”. In: *Phys. Rev. A* 94 (5 2016), p. 052325. DOI: 10.1103/PhysRevA.94.052325. URL: <https://link.aps.org/doi/10.1103/PhysRevA.94.052325>.
- [44] Samuele Ferracin, Theodoros Kapourniotis, and Animesh Datta. “Accrediting outputs of noisy intermediate-scale quantum computing devices”. In: *New Journal of Physics* 21.11 (2019), p. 113038. DOI: 10.1088/1367-2630/ab4fd6.
- [45] Zhenyu Cai and Simon C Benjamin. “Constructing smaller Pauli twirling sets for arbitrary error channels”. In: *Scientific reports* 9.1 (2019), p. 11281. DOI: 10.1038/s41598-019-46722-7.
- [46] Aditya Jain et al. *Improved quantum error correction with randomized compiling*. 2023. arXiv: 2303.06846 [quant-ph].
- [47] Thomas A Caswell et al. *matplotlib/matplotlib*. Version v3.4.3. DOI: 10.5281/zenodo.5194481.
- [48] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.
- [49] Aric Hagberg, Pieter Swart, and Daniel S Chult. *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.