Re-examining the quantum volume test: Ideal distributions, compiler optimizations, confidence intervals, and scalable resource estimations

Charles H. Baldwin, Karl Mayer, Natalie C. Brown, Ciarán Ryan-Anderson, and David Hayes
Quantinuum, 303 S. Technology Ct, Broomfield, Colorado 80021, USA

The quantum volume test is a full-system benchmark for quantum computers that is sensitive to qubit number, fidelity, connectivity, and other quantities believed to be important in building useful devices. The test was designed to produce a single-number measure of a quantum computer’s general capability, but a complete understanding of its limitations and operational meaning is still missing. We explore the quantum volume test to better understand its design aspects, sensitivity to errors, passing criteria, and what passing implies about a quantum computer. We elucidate some transient behaviors the test exhibits for small qubit number including the ideal measurement output distributions and the efficacy of common compiler optimizations. We then present an efficient algorithm for estimating the expected heavy output probability under different error models and compiler optimization options, which predicts performance goals for future systems. Additionally, we explore the original confidence interval construction and show that it underachieves the desired coverage level for single shot experiments and overachieves for more typical number of shots. We propose a new confidence interval construction that reaches the specified coverage for typical number of shots and is more efficient in the number of circuits needed to pass the test. We demonstrate these savings with a $QV = 2^{10}$ experimental dataset collected from Quantinuum System Model H1-1. Finally, we discuss what the quantum volume test implies about a quantum computer’s practical or operational abiliti-
1 Introduction

Quantum computers continue to advance towards higher performance devices that are nearing the regime of running advantageous algorithms. However, with several different device architectures and candidate algorithms, an open question remains: how do we quantify performance? The quantum volume (QV) metric was originally proposed as an answer to this question by weighing qubit number with fidelity [1, 2], or simply stated as, “don’t count your qubits until you can entangle them” 1. Later, QV was formalized in an explicit set of test circuits and passing criteria [3], which we refer to as the quantum volume test (QVT), and has recently been measured on several quantum computers [3, 4, 5, 6, 7, 8, 9, 10]. In this paper, we present a detailed study of the test for arbitrary qubit number \( N \), referred to as QVT\(_N\).

The QVT is an example of a broadening focus in quantum computer benchmarking from the component level to the system level. Component-level benchmarks, e.g., tomography [11] and randomized benchmarking [12, 13], return rigorous estimates (with some assumptions) of the primitive components, e.g., fidelity of state preparation, gates, and measurement. While widely used for building quantum computers, component-level benchmarks often fail to describe the behavior of larger quantum circuits [14, 15]. This may be due to additional errors that are missed by component-level benchmarks [16]. System-level benchmarks are designed to be sensitive to such errors, and therefore provide valuable feedback on the feasibility of running larger circuits on a quantum computer. Some system-level benchmarks have adopted component-level techniques to estimate the fidelity of full-system operations [15, 17, 18, 19, 20]. Other system-level benchmarks — like QVT — abandon the expressed goal of measuring errors and instead look to demonstrate that the system passes performance criteria deemed to be “hard” [14, 21, 22].

The value of different system-level benchmarks is beyond the scope of this paper, but whatever opinion one might have, it is self-evident that correctly interpreting any benchmark result requires an in-depth understanding of the test. This calls for a clear analysis at several levels: (1) the motivation and consequences of design decisions used to build the test, (2) how the protocol responds to noise, and (3) how the performance metric relates to other useful tasks in quantum information processing. The original QVT proposal in Ref. [3] analyzed most of these tasks to motivate the use of the test. In this work, we expand on all points by performing a series of analytic and numerical studies of QVT to better understand experimental test results and inform future performance goals.

We briefly discuss a few results of our study here. First, QVT circuits’ ideal behavior and the effectiveness of compiler optimizations are functions of \( N \) (including whether \( N \) is even or odd). Second, success in QVT is mostly scales to the total gate error magnitude and not the source of errors. Third, the confidence interval proposed in Ref. [3] is more restrictive than necessary, and we define a new confidence interval method that allows fewer circuits to reach the desired confidence level. Finally, the required gate fidelity to pass QVT\(_N\) for near term devices aligns reasonably well with other near-term goals such as early demonstrations of quantum error correction.

This paper is organized as follows: In Sec. 2 we review the basic steps in QVT. Next, in Sec. 3 we answer some frequently asked questions about QVT and refer to later sections for more detail. Then, in Sec. 4 we analyze the ideal behavior of the QVT\(_N\) circuits and different effects of previously proposed compiler optimizations. In Sec. 5 we perform numerical simulations to estimate QVT\(_N\) success probabilities under different error models and predict future error targets with a scalable method. In Sec. 6 we study the confidence intervals for QVT\(_N\) and propose a new method with tighter coverage. In Sec. 7 we compare QVT\(_N\) results to other algorithms such as quantum error correction. Finally, in Sec. 8 we summarize our work and discuss open questions.

2 Overview of the Quantum Volume Test

In Ref. [3], Cross et al. outlined the QVT\(_N\) procedure, which we summarize below. The task of QVT\(_N\) is to experimentally run a type of random quantum circuit and generate output dis-
tributions exhibiting characteristics of a random unitary ensemble. This is quantified by a measure called the heavy output frequency (defined below). The procedure was inspired by Ref. [23], which proposed methods to demonstrate quantum computational advantage in sampling, where they asserted that there is no polynomial-time classical method that samples heavy outputs at least 2/3 of the time (under several assumptions). Therefore, observing heavy outputs more than 2/3 of the time from a quantum computer is an indication of a quantum speedup in sampling.

In general, QVT$_N$ is performed by running $n_c \geq 100$ different random quantum circuits on the quantum processor under investigation, and certifying their performance with classical simulation. As an example, the procedure for QVT$_4$ is outlined in Fig. 1. The circuits are constructed by randomly pairing qubits and applying Haar-random SU(4) gates to each pair as shown in Fig. 1a, (for odd $N$ one qubit is left out in each of these rounds). The random pairing and gating is repeated $N$ times for $N$ qubits making the circuits “square,” since the depth (number of non-parallel gates) is on the order of the width (qubit number). Each circuit is simulated classically to determine the ideal distribution of measurement outputs in the standard computational basis (Fig. 1b). The simulated distribution is then sorted according to the relative ideal probabilities of each output and the median output is found (Fig. 1c). Heavy outputs are defined as measurement outputs with an ideal probability greater than the median. Each circuit is then run $n_s$ times on the device and the ratio of heavy outputs observed to the total shots in the experiment $n_s \times n_c$ is calculated and called the heavy output frequency $\hat{h}$. The confidence interval lower bound of $\hat{h}$ is estimated as

$$C_{lower} = \hat{h} - 2 \sqrt{\frac{\hat{h}(1 - \hat{h})}{n_c}},$$  

which is derived assuming all circuits have the same number of shots. If $C_{lower} > 2/3$, QVT$_N$ is passed and the system has $QV = 2^N$.

Throughout this article, we study the heavy output probability averaged over the set of all possible QVT$_N$ circuits. Without errors, we call this quantity the ideal success $h_{\text{ideal}}$. With errors, we call this quantity the actual success $h$. The actual success does not include finite sampling of the data to differentiate those effects from errors. In general errors cause $h \leq h_{\text{ideal}}$.

### 3 FAQ’s about the QVT

Since this paper covers a wide range of topics, we first attempt to answer some frequently asked questions about QV and refer the interested reader to more details in the corresponding sections below.

**Q1:** What is the average heavy output probability without errors?

**A1:** In the original QVT proposal [3], the asymptotic ideal heavy output probability is given as $\approx 84.7\%$. This means QVT$_N$ success is not equivalent to fidelity, which equals one without errors. In Sec. 7 we propose a scaling method to better interpret QVT$_N$ measurements that relates more closely to circuit fidelity. It was also demonstrated in Ref. [3] that circuits with small $N$ have slight deviation of heavy output probabilities from the asymptotic value. In Sec. 4.1 we...
shed additional light on this deviation and we find that for \( N < 10 \) ideal success varies with qubit number by about 1-2\%. This may seem like a small variation but in practice could mean the difference between passing and not passing. We also find a difference in scaling of ideal heavy output probability for odd \( N \) vs even \( N \). This means that the success between dimensions is difficult to compare. For example, heavy output probability of \( 70\% \) for QVT\(_2\) may be require lower errors than \( 70\% \) for QVT\(_3\) because the ideal heavy output probability for QVT\(_3\) is much higher.

**Q2:** The QVT allows arbitrary compiler optimizations (within reason [3]), but what effect do they have on the test?

**A2:** Classical compilation plays an important role in NISQ algorithms, especially on machines with limited connectivity and the QVT rewards quantum compilers’ ability to optimize circuit compositions. Ref. [3] proposed two optimizations for QVT that reduce the total number of two-qubit gates to improve the chances of success. We find these optimizations help significantly for \( N \leq 10 \) qubits, for example reducing the number of two-qubit gates by about half for \( N = 4 \), but provide diminishing advantages as \( N \) increases, for example only a 20\% reduction for the same methods with \( N = 15 \). We explore the exact scaling to better determine advantages for any qubit number in Sec. 4.2.

**Q3:** How does QVT\(_N\) success scale with two-qubit gate fidelity?

**A3:** Most systems are limited by two-qubit gate errors, making them a primary focus in running any benchmark or algorithm. We find that the success of QVT\(_N\) experiments roughly scales with the fidelity of two-qubit gates \( f_{\text{TQ}} \) raised to the expected number of two-qubit gates \( n_{\text{TQ}} \), \( h \propto f_{\text{TQ}}^{n_{\text{TQ}}} \). The total number of two-qubit gates is at most \( 3\lceil \frac{N}{2} \rceil N \) (where “floor” \( m \rceil \lceil m \rceil \) rounds \( m \) down to the nearest integer) but can be significantly reduced for \( N \leq 10 \) qubits with compiler optimizations (see Sec. 4). Also this scaling does not take into account other error sources like single-qubit gate, measurement errors or crosstalk errors, which also impact QVT\(_N\) success. This scaling is only a rough estimate and a more detailed analysis is presented in Sec. 5.

**Q4:** Is QVT\(_N\) only sensitive to two-qubit gate error?

**A4:** Two-qubit gate errors are the main concern for most systems, but QVT\(_N\) also requires single-qubit gates and of course state preparation and measurement as well as being sensitive to other errors like crosstalk and idling errors. For single-qubit gate fidelity \( f_{\text{SQ}} \), we find that a similar expression holds as in the previous question \( h \propto f_{\text{SQ}}^{n_{\text{SQ}}+N} \) since there are roughly two single-qubit gates for every two-qubit gate plus \( N \) additional gates at the beginning of each circuit. For state preparation and measurement we observe a softer exponential scaling with fidelity \( f_{\text{P/M}} \) since there are only \( N \) state preparations and measurements \( h \propto f_{\text{P/M}}^N \). Similar to Q3, these scalings are rough estimates. A more detailed method that can account for other errors and combinations of multiple errors is presented in Sec. 5 and compared to numerical simulation. We also attempt to simulate effects like crosstalk but of course these are system specific, and therefore it is important to run QVT\(_N\) on actual hardware to demonstrate low levels of errors.

**Q5:** Does QVT\(_N\) have different behavior with different types of errors?

**A5:** We find that QVT\(_N\) behaves similarly with different types of errors of similar magnitudes as measured by infidelity. This is best exemplified by comparing two-qubit coherent errors to depolarizing errors. We find both of these error models produce similar QVT\(_N\) success when they have the same gate fidelities in Sec. 5. We observe similar trends for other error models as well. It is impossible to simulate all possible errors but we expect QVT\(_N\) success to be mostly a simple function of fidelity rather than depending on the type of error.

**Q6:** QVT requires classical simulation in the analysis, doesn’t this put a limit on the usefulness of the test?

**A6:** For \( N > 30 \) the QVT will be difficult to implement since the classical computation will be expensive. As estimated in Sec. 5, passing QVT\(_{30}\) likely requires a two-qubit gate fidelity of \( \approx 99.95\% \) along with low single-qubit gate errors and minimal crosstalk and memory errors. As studied in Sec. 7, reaching these performance levels with 30 qubits is a worthy medium-term
4 Circuits

In this section we explore QVT circuit construction and optimization to better understand and predict the heavy output frequencies in experiments. QVT specifies a circuit construction method (outlined in Sec. 2) in an attempt to generate output distributions that are typical of random quantum circuits. After generating the circuits, QVT allows any circuit compilations that leave the net unitary “close” to the original ideal unitary (further specified below). Two methods that satisfy this condition were proposed in Ref. [3]. We propose an additional method and study how these methods scale for arbitrary qubit number and fidelity.

4.1 Ideal distribution

In previous work it was shown that circuits generated with random two-qubit gates on pairs of qubits (like those in QVT) form approximate unitary t-designs if sufficiently deep [25, 26, 27]. Unitary t-designs approximate the first t moments of the Haar measure, which is the invariant measure across unitaries of fixed dimension [28]. Here, we study the output states of Haar random unitaries and compare them to the output states from QVT circuits.

First, we derive the expected heavy output probability for Haar random unitaries. A Haar random state is generated from applying a Haar random unitary to any initial state. Haar random states are a superposition of computational basis states $|\psi\rangle = \sum_{j=1}^{2^N} c_j |j\rangle$ for amplitudes $c_j$ with real and imaginary parts uniformly distributed between $[-1, 1]$ subject to the normalization condition. The probability of measuring each computational basis output $x_k$ is $p(x_k) = |\langle x_k |\psi \rangle|^2 = |c_k|^2$. The probability distribution of $p(x_k)$ is found by integrating over the Haar measure $P_H(p) = (2^N - 1)(1 - p)^{2^N-2}$ [17, 29]. This is a probability distribution over output probabilities averaged over all Haar random states of dimension $2^N$. The expected heavy output prob-
ability is derived by finding the median probability of the distribution and then integrating over all probabilities above the median

\[ h_{\text{ideal}}(N) = 2^N/(1-2^N)(1 + 2^N (2^{1/(2^N-1)} - 1)). \] (2)

For large \( N \), \( P_H(p) \) approaches the Porter-Thomas (PT) distribution 
\[ P_{PT}(p) = 2^N e^{-p2^N}. \]

The large \( N \) approximations lead to the asymptotic ideal success probability of QVT circuits of

\[ h_{\text{ideal}} \approx (\log 2 + 1)/2 \approx 84.7\% \] [3, 23].

In Fig 2 we plot a comparison between different estimates of the expected heavy output probability and the heavy output probability from a sample of 5,000 QVT circuits. The simulated data is plotted in blue regions to show the distribution of heavy output probabilities based on circuit instance with mean (estimated ideal success) plotted as blue dots. One notable feature is that the estimated ideal success depends on \( N \) and oscillates between higher values for odd \( N \) and lower values for even \( N \) while converging to the asymptotic value. For \( N < 5 \) there is also a notable difference between the heavy output probability predicted by the Haar distribution \( P_H(p) \) (orange solid line) and the asymptotic estimate (dashed green line). There is also a discrepancy between the heavy output probability from the Haar distribution (orange solid line) and estimated ideal success from QVT\(_N\) (blue circles). We partially attribute the oscillations and discrepancies to two reasons:

First, the ideal success is calculated by estimating the median probability per circuit whereas the derived result is estimated by the median over all outputs, and second, the QVT\(_N\) circuits are not representative of Haar random SU(\(2^N\)) unitaries (and in fact the difference depends on odd versus even \( N \)). We further investigate the second point below.

To compare the output states from QVT circuits to Haar random states we conducted a numerical study of 5,000 QVT\(_N\) circuits for \( N = \{2, \ldots, 9\} \). We then extended the circuits with the same construction method out to 6\( N \) rounds of permutations and SU(4) gate pairings, which is \( 6N \) QVT\(_N\) circuit depth. At various depths we extracted the quantum state for each circuit in order to study the how the output distributions converge. For each \( N \) and circuit depth there are \( 5,000 \times 2^N \) probabilities and we see empirically that the corresponding distribution for small qubit number and depth do not match the PT or Haar distributions in the tails, e.g. Fig. 3(a-d). To verify this observation, we conducted two tests. First, we calculated the average entanglement entropy over each qubit partition for each circuit. We traced out \( N - 1 \) qubits in each circuit and calculated the entropy of the remaining subsystem and averaged over all qubits and circuits. Second, we applied the Kolmogorov-Smirnov (KS) test between the predicted Haar distribution and the simulated probabilities. The KS test measures the distance between a sampled distribution and a model probability distribution [31]. As shown in Fig 3(e-h), both tests show that QVT\(_N\) circuits do not produce the expected values for Haar random states but do converge with longer sequences, as expected based on Refs. [26, 27]. For the KS test, the test statistic asymptotes with circuit depth due to finite sampling effects, which are reduced for higher \( N \) since there are more probabilities to compare. The finite sampling asymptote is not reached with the standard QVT\(_N\) circuit depth, but is fairly close for \( 2N \) QVT\(_N\) circuit depth.

One other notable feature of the study is that the estimates of each test have higher entanglement entropy and KS test statistic for odd \( N \) than for even \( N \), indicating that odd \( N \) circuits are further from SU(\(2^N\)). One reason this occurs is that for odd \( N \) some circuits have 100% ideal heavy output probability, which is seen in Fig 2 in the violin plots. This occurs with 572/5,000 random circuits for \( N = 3, 9/5,000 \) for \( N = 5 \), and 0/5,000 for larger \( N \). The reason is that for odd \( N \) circuits one qubit is always left out per round, which means that in some circuits one qubit will be left out for all rounds. Then, the left out qubit totally determines the heavy outputs, which are outputs with the left out qubit in the \(|0\rangle\) state. We can calculate the probability of sampling such a circuit based on the probability a qubit is left out in any given round. Since the pairings are random, after the initial round the probability that the same qubit is left out in the next round is \( 1/N \). Repeat this for all \( N - 1 \) subsequent rounds that require repairing and the probability that the same qubit is left out every time is \( 1/N^{N-1} \). This closely matches our numerical estimates: \( N = 3 \) we expect 555.55 circuits, \( N = 5 \) we expect 8 circuits, \( N = 7 \) we expect 0.042 circuits, and for \( N = 9 \) the expected circuits is \( \leq 1.1 \times 10^{-4} \). This effect also diminishes.
quickly as \( N \) increases, which matches our numerical comparisons in Fig. 3.

4.2 Compiler optimizations

In a QVT, any compilation method may be applied to the circuits such that the resulting unitary is close to the original unitary [3]. One should not use the result of classical simulation in the compilation, e.g., finding the heavy outputs then designing the circuit that produces a single heavy output. Ref. [3] proposed methods to compile the circuits to reduce the total number of two-qubit gates, and therefore improve the success. These compiler optimizations were not expected to scale favorably with qubit number and here we elucidate the exact scaling of two such optimizations: block combinations and block approximations. We also introduce a new optimization based on arbitrary angle gates.

4.2.1 Block combinations

The block combination optimization takes \( k \geq 2 \) sequential blocks of SU(4) gates scheduled to operate on the same two qubits and combines them into a single SU(4) gate as shown in Fig. 4a. This reduces the number of two-qubit gates in this section of the circuit from \( 3k \) to 3. Here we calculate the average number of two-qubit gates saved as a combinatorial problem.

Each round of a QVT circuit requires the qubits to be divided into pairs that each receive random SU(4) gates. An arrangement represents this pairing for a given round and is defined by a set of tuples representing the paired qubits \( p = \{ (0, 1), (2, 3), \ldots \} \). We assume the first arrangement pairs the nearest neighbor qubits without loss of generality. Therefore, a QVT circuits contains a total of \( N - 1 \) arrangements.

The first step is to determine the total number of possible arrangements, denoted \( f(N) \). For now, assume \( N \) is even. Given an initial qubit, pick a second qubit to pair it with; there are \( N - 1 \) choices. Iterate to the next qubit and pick its pair; there are \( N - 3 \) remaining choices. The procedure continues until no qubits are remaining. The total number of possible arrangements is then \( f(N) = (N - 1)(N - 3) \cdots 3 \cdot 1 = (N - 1)!! \), where “!!” is a “double factorial.” By a similar

[Figure 3: (Top row) Example probability distributions from 5,000 random QVT\(_N\) circuits of different \( N \) and depth. Blue histograms show numerical results from sample circuits. Black lines show estimated errorbars from multinomial distribution. Green line is the asymptotic estimate and orange line is the expected distribution from Haar random SU(2\(^N\)). From left to right: (a) \( N = 4 \), depth = 4 (standard QVT\(_4\)). (b) \( N = 4 \), depth = 16, (c) \( N = 5 \), depth = 5 (standard QVT\(_5\)). (d) \( N = 5 \), depth = 20. (Bottom row) Comparisons of output distribution from the sample of 5,000 QVT\(_N\) circuits for various \( N \) as a function of depth (normalized by \( N \)). (e) Average entanglement entropy across all single-qubit partitions (between 2-dimensional and \( (2^{N-1})\)-dimensional Hilbert spaces). Colored dashed lines are the expected limit derived in Ref. [30]. (f) Slice over \( N \) depth circuits (dashed line) corresponding to standard QVT\(_N\) test and \( 6N \) depth circuits (solid line). (g) KS test statistic between binned probabilities from 5,000 circuit sample and the expected Haar distribution. (h) Slice over \( N \) depth circuits (dashed line) corresponding to standard QVT\(_N\) test and 6\( N \) depth circuits (solid line).]
Figure 4: Overview of block combination and effectiveness. (a) Example block combine optimization where the second permutation \( \Pi = (0, 1, 2, 3, 4, 5) \rightarrow (0, 1, 3, 2, 4, 5) \) leaves qubits 4 and 5 paired in the same way for the second and third rounds of SU(4) gates. Since the second and third SU(4) gates acting on qubits 4 and 5 are random, it is equivalent to combine them into a single random SU(4) gate, which should have a lower noise level than executing two gates, thereby increasing the success. (b) The average relative savings in the number of two-qubit gates upon taking advantage of block combinations. The blue line is equal to \((N - 1)/N\), while the data points are found from Eq. (6). The error bars shown are derived in a similar manner by solving for the standard deviation.

The argument for odd \( N \), \( f(N) = N!! \), and in general

\[
f(N) = \begin{cases} 
N!! & \text{for } N = \text{odd} \\
(N - 1)!! & \text{for } N = \text{even}
\end{cases} \\
= \frac{N!}{2^{[N/2]} [N/2]!}.
\]

This subproblem is equivalent to finding the number of perfect matchings of a fully connected graph [32].

The next step is to find the number of times two consecutive rounds do not contain any repeated pairs, which we denote as \( g(N) \). Let \( S \) be the set of all possible arrangements and let \( p \) be the arrangement of the first round. Then let \( S_{p_j} \) represent the set of arrangements that contain the pairing \( p_j \), which is the \( j \)th pairing from the first round. Then the set of arrangements that do not repeat the pair \( p_j \) is the complement of \( S_{p_j} \) in the set \( S \), which is denoted \( \overline{S_{p_j}} \). The set of arrangements with no repeats from the previous round is then the intersection over all pairs \( p_j \) of the sets that do not contain that pair,

\[
g(N) = \left| \bigcap_{j=1}^{[N/2]} S_{p_j} \right|
= \sum_{j=1}^{[N/2]} S_{p_j}
= f(N) - \sum_{j=1}^{[N/2]} S_{p_j}
= \sum_{k=0}^{[N/2]} (-1)^k \binom{[N/2]}{k} f(N - 2k),
\]

where the second line uses De Morgan’s law, and the last line uses the inclusion-exclusion principle [33] and noting that \(|S_{p_j}| = f(N - 2)\), \(|S_{p_m} \cap S_{p_{k+j}}| = f(N - 4)\), etc.

Next, define the number of times two consecutive rounds contain exactly \( M \) repeated pairs as \( h(N, M) \). The \( M \) repeated pairs are chosen in any combination from \([N/2]\) pairs in the first round. The remaining \( N - 2M \) qubits must then contain no repeated pairs from the first round. Therefore,

\[
h(N, M) = \binom{[N/2]}{M} g(N - 2M),
\]

which reduces to \( g(N) \) for \( M = 0 \) as expected.

The expected number of gates after opportunistic combining is \( n_{TQ}(N) \) and is found by iterating through each round of the QVT circuit and calculating the fraction of circuits that require new pairs from the previous round. For the first round there are \( 3 \cdot [N/2] \) gates since all pairs are new. In the next rounds, we iterate through the possible number of repeated pairs from the previous round \( k \) from \( k = 0 \) (no repeated pairs) to \( k = [N/2] \) (all repeated pairs). The fraction of total possible arrangements with exactly \( k \) repeated pairs is \( h(N, k)/f(N) \). For \( k \) repeated pairs there are then \( 3([N/2] - k) \) new gates. This gives the expected total number of two qubit gates,

\[
n_{TQ}(N) = 3 \cdot [N/2] \\
+ \frac{3(N - 1)}{f(N)} \sum_{k=0}^{[N/2]} h(N, k)([N/2] - k).
\]

Accepted for Quantum 2022-04-26, click title to verify. Published under CC-BY 4.0.
Figure 5: Numerical study of block approximation. (a) The gate decompositions used for generating exact and approximate random SU(4) gates. Zero two-qubit gates (not shown) is two parallel single-qubit gates. (b) The fraction of random SU(4) gates that satisfy the given fidelity condition for replacement: (purple squares) zero two-qubit gates, (orange triangles) one two-qubit gate, (green up-side-down triangles) two two-qubit gates, and (red pluses) three two-qubit gates or no reduction. The average fraction of two-qubit gates are plotted in blue circles with mirroring option (solid line) and without mirroring option (dashed line). The fidelity of the approximation is plotted with black crosses.

The expected fraction of gates saved with the block combinations \( n_{\text{TQ}}(N)/(3|N/2|N) \) is plotted in Fig. 4 along with standard deviations derived in a similar manner. For even qubit numbers we empirically see \( n_{\text{TQ}}(N)/(3|N/2|N) = (N-1)/N \). Interestingly, the reduction is relatively less effective for odd \( N \) than for \( N+1 \). This is because there are relatively fewer total pairs for odd \( N \) compared to \( N + 1 \), and therefore less options to combine. In general, we find that the combine compilation roughly saves one round of the QVT\(_N\) circuit.

### 4.2.2 Block approximations

The other compiling procedure proposed in Ref. [3] is to replace the standard SU(4) block decomposition in Fig. 5a with an approximate version that contains fewer CNOT gates (or other perfect two-qubit entangler) if the approximate version has a higher estimated fidelity with errors. They also proposed a “mirror” option to additionally test SWAP\(\times U\) to see if a corresponding approximation meets the fidelity conditions for replacement. Mirroring likely adds no overhead to the circuit, since all qubits are randomly reordered after the SU(4) block so on average no extra qubit routing is required. However, mirroring does change the SU(4) block potentially making it easier to implement with an approximation. More details are given in Ref. [3]. If the condition is met for this mirror case then the new gate includes a SWAP and the qubit ordering is updated in future rounds to compensate. Ref. [3] investigated both these options by deriving the fraction of SU(4) blocks that meet the fidelity criteria. Here, we extend this investigation to smaller gate error regimes of \(10^{-1} - 10^{-5}\) and numerically study performance with block combinations.

We performed a numerical search over 100,000\(^2\) SU(4) blocks to determine what fraction meet the fidelity criteria with and without mirroring. The results are plotted in Fig. 5 where the dashed blue line with circles shows the fraction of two-qubit gates returned from the approximation without mirroring and the solid blue line with circles shows the fraction of two-qubit gates with mirroring. The black line with crosses shows the fidelity of the resulting approximate gates without errors. The other colors show the fraction of different approximate versions of the SU(4) gates that meet the fidelity requirements. The orange curve with triangles shows a significant number of SU(4) gates can be approximated with a single CNOT gate if an infidelity of \(10^{-2}\) is acceptable, but for lower error rates there are very few. Likewise, the green curve with up-side-down triangles shows that even out to very low error rates of \(10^{-4}\) a significant portion of random SU(4) gates can be constructed using two CNOT operations. For large enough \(N\), the QVT\(_N\) will require an error rate < \(10^{-4}\) at which point nearly all SU(4) gates will require three CNOT’s, but this is also beyond the regime where the current

\(^2\)Qiskit 0.28.0 only generates 1,000 random SU(4) blocks and samples from that set to generate every QVT circuit. In later simulations we do not restrict QVT circuits to using this smaller set. While QVT performance and current optimizations may not be impacted by using this smaller sample it is possible future optimization could use excessive classical computation on such a reduced set.
incarnation of the QVT is feasible due to the classical simulation requirement.

The combined effect of the block combinations and approximations are plotted in Fig. 6 over near-term two-qubit infidelity and qubit number ranges. The number of two-qubit gates in a QVT\(_N\) circuit is \(m_{\text{TQ}}(N) = 3\lfloor N/2 \rfloor N\) without optimizations. Let \(m_{\text{TQ}}^{\text{opt}}(N, f)\) be the average number of two-qubit gates with block combinations and approximations with fidelity threshold \(f\). Fig. 6a shows a contour plot where the color gives the ratio between optimized and unoptimized two-qubit gate counts \(m_{\text{TQ}}(N)/m_{\text{TQ}}^{\text{opt}}(N, f)\). For small qubit numbers the ratio is small since the block combinations are effective. For large fidelity the ratio is small since the block approximations are effective. The white dashed box outlines the current range of QVT\(_N\) realizations as of writing. In the bottom right corner of this range the optimizations reduce the two-qubit gate count to 74% of the full circuit construction (for \(N = 10\) and two-qubit infidelity \(\approx 3.16 \times 10^{-5}\)). For larger \(N\) the savings from both methods will necessarily decrease since passing will require lower gate infidelity (moving towards the lower right of the figure).

In Sec. 5 we construct a scalable method to estimate the required infidelity to pass QVT\(_N\) for larger \(N\). The solid white line in Fig. 6a shows the estimated two-qubit infidelity necessary to pass under a two-qubit depolarizing error model. The fraction of two-qubit gates saved at this gate infidelity is plotted as a function of qubit number in Fig. 6b to show the relative savings between the methods. The optimizations are very effective at reducing the total number of two-qubit gates for small \(N\) (around 50% reduction for \(N = 4\)) but have diminishing returns as \(N\) increases (around 15% reduction for \(N = 20\)).

### 4.3 Arbitrary angle rotations

In this section we propose generating SU(4) blocks for QVT circuits using arbitrary-angle two-qubit gates and show that this reduces the error rates per SU(4) block. Achieving arbitrary angle interactions is not a simple task in current experiments but has a variety of applications like in the variational quantum eigensolver (VQE) [34], the quantum approximate optimization algorithm (QAOA) [35], and important subroutines like the quantum Fourier transform (QFT) [36].

Enabling arbitrary angle two-qubit interactions, for example \(V(\theta) = \exp[-i\theta XX/2]\), can potentially reduce the errors per SU(4) block when errors scale with \(\theta\). Many two-qubit gate errors scale with \(\theta\) such as spontaneous emission in trapped-ion systems or multiplicative rotation errors. As shown below, decomposing SU(4) blocks into arbitrary angle gates reduces the total two-qubit rotation angle per block, and therefore the impact of these types of errors.

Each SU(4) block in a QVT circuit is decomposed with a Cartan decomposition [37], which consists of a central two-qubit interaction and single-qubit gates,

\[
U = (K_1 \otimes K_2)V(K_3 \otimes K_4)
\]

\[
V = \exp[-i\frac{1}{2}(\theta_x XX + \theta_y YY + \theta_z ZZ)]
\]  

where \(U \in \text{SU}(4)\) and \(K_i\) are single-qubit gates applied individually to each qubit. Previous tests
followed the standard procedure to decompose the middle term into three CNOT gates (or another perfect entangling gate) [38].

Let \( R_{XX}(\theta) = \exp[-i\theta XX/2] \) be the available arbitrary angle gate. This is the standard Mølmer-Sørensen interaction in trapped-ion experiments with a variable amplitude to change the angle \( \theta \) [39]. We can rotate \( R_{XX}(\theta) \) with single qubit gates to generate \( \exp[-i\theta YY/2] \) and \( \exp[-i\theta ZZ/2] \). Since \( XX, YY, \) and \( ZZ \) all commute then the middle term can be constructed with three independent applications of \( R_{XX}(\theta) \) interleaved with the appropriate single-qubit gates.

We numerically generated 10,000 random SU(4) unitaries with Qiskit [40] and used its TwoQubitWeylDecomposition, which takes any two-qubit unitary and outputs a standard form based on the Cartan decomposition. For each decomposition we calculated the total rotation angle \( \theta_{tot} = |\theta_x| + |\theta_y| + |\theta_z| \) and the distribution is plotted as the orange histogram in Fig. 7. We also applied mirroring, which checks the Cartan decomposition to SWAP×U as described in the previous section, and selected the decomposition that has the smallest total angle. This distribution is the blue histogram in Fig. 7. For comparison we also plotted the average total angle for the block approximation method as the green finely dashed line with \( 5 \times 10^{-3} \) infidelity and mirroring.

The arbitrary angle decomposition has less than or equal to the total rotation angle \( \theta_{tot} \) of the standard decomposition without block approximations. We empirically observe that using arbitrary angles has average \( \theta_{tot} = 3\pi/4 \) (orange dashed line in Fig. 7) with max \( \theta_{tot} = 3\pi/2 \). This could be formalized with geometric arguments as in Ref. [41]. The mirror option further reduces the total rotation angle with average of \( 0.635\pi \) (blue solid line in Fig. 7) and the maximum total angle is \( 3\pi/4 \). The standard decomposition, which consists of three CNOT gates, is equivalent (up to local single-qubit gates) to three applications of \( R_{XX}(\pi/2) \), and therefore has \( \theta_{tot} = 3\pi/2 \). The arbitrary angle decomposition also has significantly less total rotation angle than the standard method with block approximations as shown with the comparison of the green finely dashed line of average \( \theta_{max} \approx 1.18\pi \).

One advantage arbitrary angles have over the other optimizations is that the error reduction is constant in qubit number and fidelity. The total rotation angle will be cut in half for any QVT\( N \). With better knowledge of the limiting errors in the arbitrary angle gates further improvements might also be possible.

4.4 Conclusions on circuit constructions

QVT circuit constructions and optimizations display different behavior for \( N < 10 \) then for \( N \geq 10 \) as well as different behavior for even vs. odd \( N \). The ideal success varies 1-2% with \( N \) for \( N < 10 \) (higher values for odd \( N \)) before approaching the asymptotic value as shown in Fig. 2. The circuit optimizations also can have significant impact for \( N < 10 \) reducing gate counts between 50% (\( N = 4 \)) and 26% (\( N = 10 \)) in Fig. 6. These features imply that running QVT\( N \) for \( N > 10 \) could be more challenging than the previous QVT\( N \) measurements with \( N < 10 \).

5 Simulating Errors

In this section, we present simulations of QVT\( N \) with select error models. We consider errors on three different components: single-qubit gates, two-qubit gates, and measurement. We also consider two other errors that may be missed by component-level benchmarks: memory errors and two-qubit gate crosstalk. Additionally, we pro-
pose and test a scalable method for estimating QVT\textsubscript{N} success and compare with full numerical simulation for \( N < 10 \). From the scalable method, we estimate error magnitude requirements for QVT\textsubscript{N} for various \( N \) and error models. We only consider all-to-all connectivity and any extra connectivity constraints almost certainly degrade the performance but are unique to individual systems and compilers. Example implementations of both methods are available on GitHub [42].

5.1 Numerical optimization method

To simulate QVT\textsubscript{N} experiments, we used Qiskit [40] to generate 5,000 QVT\textsubscript{N} circuits for \( N = \{2, \ldots, 9\} \). First, we determine the ideal distribution (without noise) for each circuit with Qiskit’s statevector simulator. Next, each circuit is optimized with a set of custom Qiskit transpiler passes,

- **Low**: Combine adjacent single-qubit gates only
- **Medium**: Low and SU(4) block combines outlined in Sec. 4.2.1,
- **High**: Medium and block approximation outlined in Sec. 4.2.2 with mirroring. Fidelity tolerance is selected based on the specified error magnitude, which is the best-case scenario.

These passes differ slightly from Qiskit’s built-in transpiler options, which as of writing do not include block approximations, but represent three levels of optimization used in QVT experiments [3, 4, 5, 6, 7, 8, 9, 10].

Finally, we apply different noise models with varying magnitudes to each circuit (outlined in Sec. 5.4 below) and simulate the heavy outputs with a density matrix simulator (Qiskit’s QASM simulator with the snapshot density matrix option), allowing us to distinguish finite sampling effects and circuit noise. The result is a data array of heavy output probabilities with labels [circuit index, qubit number, optimization level, error model, error magnitude].

5.2 Scalable estimation method

The simulation method outlined above becomes expensive for \( N \geq 10 \), and here we outline a scalable method based around depolarizing error assumptions and proper accounting of errors. The estimate is constructed by first approximating the fidelity of a single SU(4) block (accounting for the expected number of gates per SU(4) from the transpiling method) and then scaling this estimate for the total number of SU(4) blocks (accounting for the expected number of blocks from the transpiling method). The approach was first proposed in Ref. [15] to estimate the success of mirror benchmarking, another full-system benchmark.

First, we review depolarizing error channels and fidelity. A depolarizing error is a completely-positive trace-preserving (CPTP) quantum map that returns the original state with probability \( p \), called the depolarizing parameter, and the maximally mixed state \( \frac{1}{d} I \) with probability \( 1 - p \) [43].

\[
\Lambda[\rho] = p\rho + \frac{1-p}{d} I. \tag{8}
\]

A depolarizing error is simpler to simulate than most errors since it is specified by a single rate and commutes with all operations of the same dimension. We use two fidelity quantities: average fidelity \( (f) \) and process (or entanglement) fidelity \( (F) \) [44],

\[
F(\Lambda) = \int d\psi \langle \psi | \Lambda[|\psi\rangle \langle \psi|] |\psi\rangle = \frac{(d-1)p + 1}{d},
\]

\[
f(\Lambda) = \frac{1}{d^2} \text{Tr}[\Lambda] = \frac{(d^2 - 1)p + 1}{d^2}. \tag{9}
\]

The first equalities in each line are true for all CPTP error channels while the second is specific to a depolarizing error (summarized in Table 1 of Ref. [45]). In general, \( p \leq f \leq F \) with equality only when \( p = 1 \).

The fidelity of depolarizing errors in a circuit is estimated differently depending on if the corresponding gates are done in parallel or sequentially. A set of parallel gates can be done simultaneously (e.g., in a single round of a QVT circuit across \( N/2 \) pairs). A set of sequential gates must be done in order (e.g., \( N \) total rounds in QVT circuits). We can determine how depolarizing errors combine in parallel and sequentially based on the Liouville (or superoperator) representation of quantum processes (reviewed in Refs. [46, 47])

- **Parallel gates**: Errors on gates performed in parallel on separate qubits have a total pro-
cess fidelity equal to the product of the individual process fidelities $f_{\text{tot}} = \prod_i f_i$.

- **Sequential gates**: $2^N$-dimensional depolarizing errors from gates performed in series on the same $N$ qubits have a total depolarizing parameter equal to the product of the individual depolarizing parameters $p_{\text{tot}} = \prod_i p_i$.

The parallel gate relation actually applies to all CPTP processes but the sequential gate relation is specific to depolarizing errors since depolarizing errors commute with the gates in fixed dimension.

The scalable method works by assuming all errors are depolarizing and that each error can be scaled to cover different numbers of qubits in order to commute errors through gates. For example, pretend the given system has a coherent single qubit error on one qubit. The scalable method treats this error as a single-qubit depolarizing error. If the next gate is a two-qubit gate then that single-qubit depolarizing error will be scaled to be a two-qubit depolarizing error of the same fidelity in order to commute it through the two-qubit gate. Both of these assumptions are unlikely to hold exactly in any experiment or more complicated error model. Errors are never exactly depolarizing channels and errors cannot be scaled in the way we describe below. However, these simplification makes the method scalable to any qubit number and, as shown below, provides reasonable approximations to other, more complicated, error sources.

The first step of the method is to approximate the total error in a single SU(4) block. The SU(4) blocks consist of alternating single-qubit and two-qubit gates (see Fig. 4a, the final round of single-qubit gates is always combined with the next block). First, we assume all single qubit errors are depolarizing and combine them via the sequential rule above. Next, we determine the process fidelity of two parallel single-qubit gates (each with only single-qubit errors) based on the parallel rule above. Then, we assume that the combined single-qubit processes is a two-qubit depolarizing error and combine it with all the two-qubit errors, which are also assumed to be depolarizing, based on the sequential rule. This produces a net depolarizing rate for each SU(4) block. In principle, other errors like memory or crosstalk can also be combined in this analysis and approximated as depolarizing errors.

The next step is to scale the depolarizing rate per-SU(4)-block to approximate the full circuit error rate. Ref. [15] applied the same procedure to combine all blocks of gates at the full-circuit scale. For QVT, as $N$ increases this method is roughly equivalent to raising the process fidelity per-SU(4)-block to the $n_{\text{rounds}}[N/2]$ power, where $n_{\text{rounds}}$ is the number of rounds determined by the transpiler optimization. We find this method mostly underestimates the actual success when compared to numerical simulation.

As an alternative, we raise the average fidelity per-SU(4)-block to the power of $n_{\text{rounds}}[N/2]$. We find that this method approximates the actual success better in simulations for $N \leq 9$. The reasons why this is a better approximation likely relate to the ways errors spread in QVT circuits but we leave a complete study for future work.

The resulting estimates from either method is used as an approximation of the depolarizing rate for the entire circuit. This error produces the correct output state with probability $p_{\text{tot}}$, which has the ideal success $h_{\text{ideal}}(N)$, and the maximally mixed state $1/2^N$ with probability $1 - p_{\text{tot}}$, which will return heavy outputs half the time.

Finally, for both options we include measurement errors, which return a false output with probability $e_M$ per qubit. Therefore, the probability of measuring the correct outputs for the entire circuit is $p_M = (1 - e_M)^N$. We assume that any error in the measurement produces a heavy output half of the time. Later, we use both methods to define an estimated region of QVT$_N$ success.

The method is summarized in Algorithm 1. We define three functions. First, $\text{convert}(x, \text{avg} \rightarrow \text{proc})$ converts $x$ between different quantities (e.g. average $\rightarrow$ process fidelity with abbreviations average fidelity = $\text{avg}$, process fidelity = $\text{proc}$, and depolarizing parameter = $\text{dep}$). Next, $\text{rounds}(N)$ returns the number of parallel rounds of SU(4) blocks based on Sec. 4.2.1. Finally, $\text{gates}(\text{tol})$ returns the number of gates per SU(4) block based on Sec. 4.2.2 for given $\text{tol}$ average fidelity level.
5.3 Types of errors

We simulate QVT_N with the following errors.

- **Single-qubit errors**: QVT_N circuits contain $7[N/2]N$ single-qubit gates without optimization, (although $2[N/2](N-1)$ are eliminated with all transpiler passes outlined above). We model single-qubit errors as depolarizing as given in Eq. 8 with $d = 2$.

- **Two-qubit errors**: QVT_N circuits contain $3[N/2]N$ two-qubit gates without optimization. We model two types of two-qubit errors: two-qubit depolarizing (as given in Eq. 8 with $d = 4$) and coherent ZZ rotations (a common error for devices whose native two-qubit gate is based on a ZZ (or a XX or YY) interaction [4]).

- **Measurement errors**: At the end of each circuit, $N$ single-qubit measurements are made. A measurement error of probability $p_M$ falsely returns a “1” (or “0”) output when the measurement operation actually projected the qubit into “0” (or “1”). In practice, the two qubit states may have different false measurement output probabilities, but we assume they are equal for simplicity.

We also model two common types of full system errors:

- **Memory errors**: There are several instances of idle qubits in QVT circuits where memory errors can occur. First, for odd $N$ a single qubit will be left out of each gate round. Second, qubits may be left idle if gates are not able to be performed in parallel either by design, such as in Ref [4], or to avoid crosstalk errors as in Ref. [6]. Here, we add single-qubit dephasing errors before every two-qubit gate as a simple example of memory errors.

- **Crosstalk errors**: Crosstalk errors usually refer to unintended operations on qubits caused by nearby gates [16], and are architecture dependent. Assuming a linear array of qubits, we model crosstalk errors caused by two-qubit gates as single-qubit depolarizing errors on nearest neighbor qubits.

5.4 Error models

We ran numerical simulations with several different error models to examine the sensitivity of QVT_N to commonly structured noise environments. Each error model is specified by scaling factors for the various error sources introduced in Sec. 5.3, and the models are defined in Table 1. Error models are written with script font to differentiate from error sources and the abbreviations single-qubit (SQ) and two-qubit (TQ) are used for brevity. The first four models highlight different component errors (SQ depolarizing, TQ depolarizing, TQ Coherent, and Measurement models). The next four models contain some level of several errors to better approximate real systems where multiple errors are present at different magnitudes but different sources dominate (Crosstalk, Memory, TQ mixed, and Semi-realistic models).
A realization of a given error model is determined by a single error magnitude $\varepsilon$. This magnitude is scaled by the values in Table 1 to determine the average infidelity of each error source. For measurement errors the scaled error magnitude is equal to the probability of returning the incorrect output. We ran simulations with seven different error magnitudes for each error model exponentially distributed between $[10^{-3.25}, 10^{-1.25}]$.

The SQ depolarizing, TQ depolarizing, and TQ coherent error sources were normalized such that the estimated infidelity of a block of two single- and one two-qubit gate is equal to the specified error magnitude (discussed further below). The SQ depolarizing, TQ depolarizing, and TQ coherent models all have the same estimated infidelity per single- and two-qubit block equal to the error magnitude. This facilitates direct comparisons between different error models that produce similar estimates of fidelity in component level experiments like randomized benchmarking. For example, coherent errors and depolarizing errors lead to similar fidelity estimates in randomized benchmarking experiments but coherent errors may be more detrimental to other quantum circuits [45]. Memory and crosstalk errors are excluded from this normalization since these errors may be missed by a randomized benchmarking experiment. Measurement errors are also excluded since they are measured separately.

For the normalized error sources, the normalization constant $n$ is found from the early steps in Algorithm 1 under a small error approximation. The approximated average fidelity of a two single- and one two-qubit gate block is set equal to the error magnitude with normalization,

$$\varepsilon = \frac{1}{n} \left[ \frac{12}{5} \sum_i s_i (1 - F_{SQ,i}) + \sum_j s_j (1 - F_{TQ,j}) \right],$$

where $i$ labels single-qubit error sources with average fidelity $F_{SQ,i}$ and $j$ labels all two-qubit error sources with average fidelity $F_{TQ,j}$. The scaling factors $s_i$ and $s_j$ are the constants in Table 1. This allows us to solve for $n$ given $s_{i/j}$ when $F_{SQ,i} = F_{TQ,j} = 1 - \varepsilon = 0$. For example, with the SQ depolarizing model $n = 25$. The parameters used to define each error source are generated by scaling the error magnitude by $s_{i/j}/n$. For the error sources considered,

$$p_{SQ} = 2s_0\varepsilon/n,$$

$$p_{TQ} = 4s_1\varepsilon/3n,$$

$$\theta_{TQ} = 2\arccos \sqrt{1 - \frac{5s_2\varepsilon/n}{4}},$$

where $p_{SQ}$ and $p_{TQ}$ are the depolarizing single- and two-qubit rates respectively, $\theta_{TQ}$ is the rotation angle for two-qubit coherent errors, and the scaling parameters are indexed in the same order.

### 5.5 Numerical results

Here, we present simulations of the $QVT_N$ solving for the passing requirements for different qubit numbers, error models, optimization levels, and circuit samplings. To this end, we generated 5,000 random $QVT_N$ circuits for $N = 2 \to 9$ and estimated the success for transpiler optimization methods low, medium, and high, eight different error models (Table 1), and seven exponentially distributed error magnitudes ($\varepsilon \in [10^{-3.25}, 10^{-1.25}]$), for a total of 1,512 different settings and 7,560,000 simulated circuits. For $N = 2 \to 6$ we ran an additional dataset with

| Error model       | SQ depolarizing | TQ depolarizing | TQ coherent | TQ memory | TQ crosstalk | Measure |
|-------------------|-----------------|-----------------|-------------|-----------|--------------|---------|
| SQ depolarizing   | 10              | 1               | 0           | 0         | 0            | 1       |
| TQ depolarizing   | 1               | 10              | 0           | 0         | 0            | 1       |
| TQ coherent       | 1               | 0               | 10          | 0         | 0            | 1       |
| Measurement       | 1               | 10              | 0           | 0         | 10           | 1       |
| Crosstalk         | 1               | 10              | 0           | 0         | 1            | 1       |
| Memory            | 1               | 10              | 0           | 1         | 0            | 1       |
| TQ mixed          | 1               | 10              | 0           | 1         | 0            | 1       |
| Semi-realistic    | 1               | 10              | 1           | 1/2       | 1/2          | 1       |

Table 1: Error models with names given in first column based on dominant sources of errors. Each error model has a different ratio of error sources (named in first row) that are combined to determine a realization based on an error magnitude $\varepsilon$. 

---

Accepted in *Quantum* 2022-04-26, click title to verify. Published under CC-BY 4.0.
Figure 8: Numerical passing threshold estimates from interpolation of QVT$_N$ for $N = \{2, \ldots, 9\}$ with various optimization levels, error models and magnitudes. (a) Passing error magnitude $\hat{\varepsilon}$ (error magnitude when estimated success equals 2/3) for QVT$_N$ and high optimization for each error model as a function of $N$. (b) Percent change in passing error magnitude for three example error models (TQ depolarizing, Measure, and Memory with same colors as a). Line styles indicate different optimization methods. (c) Percent change in passing error magnitudes for TQ depolarizing model and high optimization for different numbers of circuits. Line styles indicate different number of circuits. (d) Passing qubit number $N$ as a function of error magnitude (estimated success is above 2/3) for each error model and high optimization. (e) Passing qubit number as a function of error magnitude for different optimization levels and error model (TQ depolarizing, Measurement, and Memory with same colors as b). (f) Passing qubit number as a function of error magnitude for different number of total circuits with TQ depolarizing model and high optimization.

larger $\varepsilon$ to sample success rates below 2/3. For a given qubit number, error model, magnitude and optimization level we assume the simulated average heavy output probability with errors over all 5,000 circuits is approximately equal to the success (average over all QVT circuits).

We present the data in terms of the minimum requirements to pass the QVT$_N$ test from both a qubit limited and a fidelity limited perspective. If the system is qubit limited the main question is what fidelity is required to pass QVT$_N$ for a given qubit number? This perspective is plotted in Fig. 8a-c. If the system is fidelity limited the main question is what qubit number $N$ can pass QVT$_N$ with a given fidelity? This perspective is plotted in Fig. 8d-e. Below, we mostly follow the qubit limited perspective but translate all results to the fidelity limited view in parentheses.

Fig. 8a (and d) plots the estimated maximum error magnitude $\hat{\varepsilon}$ that passes QVT$_N$ for each error model as a function of qubit number $N$ (and estimated maximum passing qubit number $\hat{N}$ as a function of error magnitude). We refer to this maximum as the passing threshold $\hat{\varepsilon}$ (or qubit number $\hat{N}$) where the estimated success is equal to 2/3 determined by cubic spline interpolation of the dataset with fixed $N$ (and interpolation of both $N$ and $\varepsilon$). The models that are dominated by gate errors, e.g. the TQ depolarizing, SQ depolarizing, TQ coherent, and TQ mixed models, have the easiest to achieve passing thresholds. When we add extra errors beyond the standard component-level error sources, e.g. in the Crosstalk, Memory, and Semi-realistic models, the test is harder to pass and the estimated passing threshold. This matches with experimental results where performance on system-level benchmarks is often worse than predicted from component-level benchmarks [14, 15] since often there are unmeasured errors present.

Overall, the estimated passing thresholds fall into three groups based on the error model’s total magnitude and not the type of errors. First, SQ depolarizing, TQ depolarizing, TQ coherent, and TQ mixed all have the same magnitude per single- and two-qubit gate round (as defined in Sec. 5.4) but different types of errors dominate. Second, Crosstalk, Memory, and Semi-realistic...
all have similar magnitude per single- and two-qubit gate round but they also include other error sources with similar magnitudes. Finally, Measurement has a different scaling with $N$ since measurement errors dominate but there are a linear number of measurements versus a quadratic number of gates. However, this difference is only observable for small $N$. As $N$ increases the Measurement model begins to scale more similarly to other models since the number of measurements is much smaller than the number of gates. The similarities within each group may be partially due to the types of errors we selected but also implies that the $QVT_N$ is mostly sensitive to total error magnitude and not type of error or other metrics like diamond norm [44]. This is not wholly unexpected for random circuit averaging and is seen in similar methods like randomized benchmarking [45].

Fig. 8b (and e) shows the effectiveness of the different transpiler options: the medium (block combines from Sec. 4.2.1) and high (block combines and approximations from Sec. 4.2.2). The passing threshold is again estimated from interpolation for the high (solid lines), medium (dashed lines), and the low optimization (finitely dashed in e). We plot three example models (TQ depolarizing, Measure, and Memory) that represent the three different groups of error models seen in Fig. 8a (and d), showing the reduced effectiveness as qubit number increases, as expected based on Sec. 4.2. For the Measurement model, the optimization methods considered are not as effective since these methods are not aimed at measurement errors but other mitigation methods may be more effective [48].

For any optimization and error model, a realization of a $QVT_N$ experiment will always require lower error magnitude (or only pass for lower qubit number) than what is plotted in Fig. 8a (and d) due to the confidence interval requirement to pass $QVT_N$. This means the success must clear a higher threshold than 2/3, which is dependent on the number of circuits run. Fig. 8c (and f) show the percent change in error magnitude (and passable $N$) for different total number of circuits extracted from cubic spline interpolation. By the definition in Ref. [3], the confidence interval is independent of $N$, and therefore the percent change is proportional to the square-root of the number of circuits. For $N = 2$ the ideal success is much lower, which also changes the confidence interval based on Eq. (1).

Next, we study the scalable method’s effectiveness at predicting the passing threshold, summarized in Fig. 9. In Fig. 9a we compare the predicted passing error magnitude from the scalable method (colored regions) to the estimation from full simulation data (points and lines) for each error model. We find that both scalable methods are within 25% difference of the simulated data for $N < 10$ but underestimate the required error magnitude (predicts error magnitudes that are harder to achieve). However, both scalable methods mostly overestimate the required error magnitude for $N = 2$ (predicts error magnitudes that are easier to achieve). This is not a large impediment since most systems are far beyond
QVT$_2$.

In Fig. 9b and c we use the scalable model (colored regions) to make predictions about QVT$_N$ requirements for $10 \leq N \leq 30$ in the (b) qubit limited and (c) fidelity limited perspectives for three example error models: TQ depolarizing, Memory, and Measurement. In both plots we added an additional error model called Unscaled that is the TQ depolarizing model with an additional fixed magnitude crosstalk error of $10^{-3}$. The three original error models perform as expected: more errors increase the requirements of the error magnitude so we expect Memory to require lower errors than TQ depolarizing. For Measurement the measurement errors are an order of magnitude larger than two-qubit errors. For small $N$ the required error magnitude scales with the number of measurements $N$ but with larger $N$ there are many more two-qubit gates that cause the error magnitude to scale with $N^2$ and the performance approaches the TQ depolarizing model. The Unscaled model has an error that cannot be lowered, and therefore sets a hard limit for QVT$_N$ of $N \approx 20$. This also affects the requirements for $N < 20$ as seen by the divergence between the Unscaled and TQ depolarizing.

Fig. 9b and c can also be used to make predictions for fidelity needed to demonstrate quantum computational advantage in sampling. For instance, take $N = 50$ as a possible point that QVT circuits will no longer be simulatable. The scalable method predicts that the TQ Depolarizing model requires two-qubit gate fidelity to be $2 \times 10^{-4}$ to pass QVT$_{50}$. However, QVT circuits might not be the most efficient method for such a demonstration, e.g. Ref. [24] uses less gates and lower fidelity.

5.6 Conclusions on simulations

Based on our limited simulations, the passing threshold for QVT$_N$ is more dependent on total error magnitude than the type of error as seen in the different error models in Fig. 8. Moreover, the good agreement between full numerical simulations and scalable approximate simulations in Fig. 9 shows that our method does a decent job of capturing the scaling of the required error magnitude to pass QVT$_N$ but mostly returns conservative estimates. This leaves room for improvement and open questions about how errors are spread in QVT and other circuits.

6 Confidence intervals

The confidence interval lower bound defines the passing criteria for QVT$_N$. As previously defined, let $\hat{h}$ be the average heavy output frequency of a set of measured circuits with finite sampling statistics and let $h$ be the average heavy output probability with errors over all QVT$_N$ circuits (success). A two-sigma confidence interval certifies that the confidence interval computed from the measured data contains $h$ 97.73% of the time.

In Ref. [3], the confidence interval is constructed assuming that each circuit is run with a single shot, the measured heavy output frequency is then a binomial random variable with probability $\hat{h}$, and it is assumed there are enough circuits that the distribution is Gaussian (defined as at least 100). This was viewed as a conservative approach since in most experiments more than one shot is run per circuit. The confidence interval is estimated based on the binomial variance and solely on the total number of random circuits (not the shots per circuit). With an equal number of shots per circuit the confidence interval is,

$$C_{lower} = \hat{h} - 2\sqrt{\hat{h}(1-\hat{h})}/n_c.$$  \hfill (12)

One problem with this confidence interval estimate is that the average heavy output frequency is usually calculated from more than one shot per circuit and so not necessarily binomial. The heavy output frequency from an individual circuit is a binomial random variable with probability $h_i$, but that probability $h_i$ has a distribution determined by the initial circuit, optimizations, and noise environment. An example of this distribution is shown in Fig. 10a for various $N$ with a fixed error model and magnitude constructed from a sample of 5,000 QVT$_N$ circuits. Fig. 10b shows that the $m = 2$ (variance) and $m = 3$ (skewness) moments have mostly stabilized at 5,000 circuits and the values decrease with $N$. Therefore, 5,000 circuits gives a reasonable representation. The average of $h_i$ over sampled circuits, is not binomial, but the variance can be bounded by the binomial sum variance inequality [49].

We propose and test a method to construct tighter confidence intervals that account for this distribution across circuits and still covers 97.73% of experiments. This method is based on a semi-
parametric bootstrap technique originally proposed for randomized benchmarking [50]. An example implementation of this method and comparison to the original method is available on GitHub [42].

Bootstrapping is a technique to construct confidence intervals based on repeatedly sampling from a dataset [51]. Given a QVT dataset that consists of \( n_c \) circuits each with \( n_s \) shots. We first sample “non-parametrically” \( n_c \) circuits with replacement (not removing any sampled circuit from the dataset for future sampling). For each of these circuits, we then sample “parametrically” \( n_s \) shots from a binomial distribution with probability equal to the measured heavy output frequency of that given circuit. This whole procedure is repeated many times to produce a distribution of heavy output frequencies. From this distribution we calculate quantiles that cover any percentage of the data and use those quantiles to construct confidence intervals. The method is summarized as follows:

1. Randomly sample \( n_c \) circuits from the dataset with replacement
2. For each of the \( n_c \) circuits, randomly sample \( n_s \) shots based on a binomial distribution with the probability set to the heavy output frequency of the given circuit \( h_i \)
3. Estimate the average sampled heavy output frequency \( \hat{r} \)
4. Repeat steps (1-3) \( n_b \) times to form the distribution \( \{\hat{r}_i\} \)
5. Calculate the lower two-sigma confidence interval based on the distribution \( \{\hat{r}_i\} \) and basic bootstrap confidence interval \( C_{\text{lower}} = 2\pi - Q(\{\hat{r}_i\}, 97.73\%) \) [51] where \( \pi \) is the mean of \( \{\hat{r}_i\} \)

Step 1 reflects the statistical fluctuations from randomly sampling QVT circuits, while step 2 reflects the quantum statistical fluctuations from measuring the circuits. The quantile function \( Q(\{\hat{r}_i\}, 97.73\%) \) returns a threshold that is greater than 97.73% of the distribution \( \{\hat{r}_i\} \).

To test the confidence intervals we check the coverage probability and interval widths. The coverage probability is the fraction of times the true value is contained within the confidence interval for asymptotically many repetitions of the experiment. In the case of QVT, the coverage probability is the fraction of cases where the actual success (average heavy output probability over all possible QVT circuits) for a given QVT and error model is contained within the constructed confidence interval. The interval width is the distance between the measured value and the confidence interval bound. For QVT, we only study the lower width, which is the distance between the measured heavy output frequency and the constructed lower two-sigma confidence interval.
To test the coverage probability and confidence interval widths, we sample from the numerical data generated in Sec. 5 for the TQ depolarizing, Measurement, TQ mixed, and Semi-realistic error models. Again, we assume that the average heavy output probability with errors over the 5,000 circuits sample for given qubit number, error model, magnitude and optimization level is approximately equal to the success. In Fig. 10b we show that this is a good approximation since the moments of this distribution stabilize as more circuits are simulated. In Fig. 10c we study the moments for each distribution for all sets of 5,000 circuits and see that in fact the second through sixths moments shrink mostly with qubit number and some dependence on errors. The plotted data only shows the absolute value of the moments, but some odd number moments are in fact negative for small qubit number, which indicates a small amount of skewness in the distributions.

For a given qubit number, error model, magnitude and optimization level, we simulate 5,000 QVTN experiments by sampling nc circuits from our original sample with replacement. Each sampled circuit has a saved heavy output probability and we perform a binomial sampling with ns shots to simulate finite sampling effects. We construct the average heavy output frequency of this simulated experiment instance and perform the semi-parametric bootstrap method to calculate the confidence interval lower bound with nb = 1,000. Finally, we test to see if this confidence interval lower bound is below the estimated success from the original 5,000 circuit sample to calculate the coverage probability. We repeated this over a grid of experiments with nc = [10, 50, 100, 250, 500, 1000] and ns = [1, 10, 50, 100, 1000]. We also calculated the original confidence interval for each simulated experiment for comparison.

The results of the coverage analysis are plotted in Fig. 11 and show the coverage over different qubit numbers, error models, error magnitudes, sampled circuits and shots all flattened into one dimension. To separate out the effects of shot number we plot the coverage for ns = 1 tests separate in Fig. 11a and from all other shot num-

![Figure 11: Comparison of confidence interval methods.](image)

![Figure 12: Confidence interval (CI) widths comparison between original method (dashed lines with squares) and bootstrap method (solid lines with circles) for N = 5 and Semi-realistic error model and high optimization.](image)
bers in Fig. 11b. The dotted horizontal black line shows the specified confidence level 97.73%. The simulated data shows that both confidence intervals fail to achieve the specified coverage level when \( n_s = 1 \) for most tests (Fig. 11a). Using the original method, the lowest coverage occurs for smaller circuit counts \( (n_c < 100) \), which is outside the specifications. However, even for larger \( n_c \) the original method still returns coverage around 95% for several tests. The bootstrap method fails almost uniformly for \( n_s = 1 \).

When going beyond single shot experiments, \( n_s > 1 \), both methods return higher coverage as shown in Fig. 11b. The original method has much higher than 97.73% coverage for all tests and actually achieves unit coverage for most tests. The bootstrap method fails to match the specified coverage for small number of circuits \( (n_c = 10 \) are plotted as red “x”) or lower qubit number \( N = 4, 5, 6 \). However, this should not be a problem when testing \( N > 6 \) and adhering to the QVT requirement of \( n_c \geq 100 \). We note that the coverage level does seem to increase with qubit number, leaving room for improvement in confidence interval construction for larger \( N \).

Larger coverage implies tighter confidence intervals but it is difficult to study how the confidence interval width scales for the bootstrap method since it is numerically estimated and dependent on the error magnitude, \( n_s \) and \( n_c \). In Fig. 12 we plot the confidence interval width as a function of \( n_c \) or \( n_s \) with variable \( n_c \), \( n_s \), or \( \varepsilon \) and fixed semi-realistic error model and \( N = 8 \). We see empirically that the width is proportional to \( 1/\sqrt{n_c} \) in Fig. 12a and c but similar attempts to fit the width to \( 1/\sqrt{n_c} \) do not match the data in Fig. 12b and d. The width is also a function of \( \varepsilon \) as shown in Fig. 12c and d but we did not attempt a fit. Fig. 12 demonstrates that the bootstrap confidence interval does tighten with number of shots while the original method is constant.

As a demonstration of the bootstrapping method we plot the confidence intervals for both methods as a function of number of circuits for the QVT\(_{10} \) data announced in Ref. [10]. The experiment was performed on the Quantum System Model H1-1 machine, similar to the machine discussed in Ref. [4]. The results are plotted in Fig. 13 and show that the bootstrap confidence interval method crosses the 2/3 threshold consistently after 213 circuits but the original method crosses at 707 circuits.

In summary, the original confidence interval construction results in a conservative coverage probability and an excessive circuit number requirement. We constructed a new method to closely match the desired coverage probability, thereby reducing the confidence interval width and saving circuits. We showed that the original design principle of single-shot experiments, \( n_s = 1 \), results in insufficient coverage probabilities for both methods. Our method also converges to higher coverage probability as qubit number increases. Other methods for constructing confidence intervals (or perhaps Bayesian method for credible intervals) might be needed to scale to even larger qubit numbers and handle \( n_s = 1 \) experiments.

7 Operational implications

The only thing QV perfectly captures, is the ability of a quantum computer to generate the ideal output distributions of random QVT circuits. Relating this ability to other useful tasks necessarily requires assumptions about the noise processes present in the machine under investigation and how those processes impact other algorithms, both of which are typically not well understood. In this section we attempt to relate QVT\(_N \) to some near term applications under some assumptions.
Circuit fidelity
0.6
0.8
0.9
1.0
and(bluecircles)
ε
squares)
be useful in relating QVT
delity for
can also be used as an estimator of average fi-
tion for similar depth quantum circuits. For a
The QVT
7.1 Random linear depth circuit
The QVT\textsubscript{N} measured heavy output frequency can also be used as an estimator of average fi-
delity for N-qubit linear depth circuits. This may be useful in relating QVT\textsubscript{N} results to state prepa-
ration for similar depth quantum circuits. For a QVT\textsubscript{N} circuit, we can rewrite the output prob-
bility as \(p_i = |\langle x_i | A U_c | 0 \rangle|^2\) where \(U_c\) is the unit-
ary for a particular QVT\textsubscript{N} circuit and \(A\) is the combination of all errors commuted outside of the
unitary. In practice, \(A\) is a complicated repre-
sentation of the errors with \(2^{4N}\) parameters (as-
suming close to best-case Markovian errors [52]).
Here, we make the oversimplified assumption that
\(A\) is a depolarizing error channel, which is similar
to our scalable method in Sec. 5.2. The exactness
of this assumption is an interesting question and
it may be reasonable based on the random struc-
ture of QVT circuits but we leave that analysis
for future work.
For a full-circuit depolarizing channel the heavy output probability of a given circuit is di-
rectly related to the circuit’s depolarizing param-
eter. From the depolarizing parameter, we calcu-
late the average circuit fidelity,
\[
F_{\text{circ}} = 1 - \frac{2^N - 1}{2^N} h_{\text{ideal}}(N) - \hat{h},
\]
where \(h_{\text{ideal}}(N)\) is the heavy output probabili-
ty without errors (studied in Sec. 4.1) and \(\hat{h}\)
is the heavy output frequency with errors for a
given circuit. This is similar to the quantity pro-
posed in Refs. [53, 54] but with a dimensional
scaling factor. A totally depolarized circuit has
\(F_{\text{circ}} = 1/2^N\). The QVT passing threshold of 2/3
corresponds to \(F_{\text{circ}} = 1/3 \ln 2 \approx 0.481\) in the
asymptotic limit of large \(N\) but is in general a
function of \(N\).
In Fig. 14, we compare the estimated circuit
fidelity \(F_{\text{circ}}\) to the average state fidelity of the
output averaged over 5,000 simulated QVT\textsubscript{N}
circuits, which we use as an approximation of the
average fidelity of the QVT\textsubscript{N} circuits. We study
the Semi-realistic model and high optimization
with four different error magnitudes. The esti-
mated fidelity from the heavy output probabil-
ity consistently overestimates the average fidelity.
This is contrary to a similar studies performed in
Ref. [54], which uses different circuit construction
that produce estimates that closely matches the
fidelity. Further investigation is required to un-
derstand why QVT circuits slightly overestimate
fidelity.
As shown in Sec. 4.1, the ideal output states of
the QVT\textsubscript{N} for large \(N\) are highly entangled.
Therefore, we can use the estimate \(F_{\text{circ}}\) as an
estimate for the fidelity of entangled state prepa-
rion with comparable depth circuits. Entan-
gled state preparations are important in several
near term algorithms such as VQE [34]. For ref-
tence, Table 2 shows the conversion of recent
QVT\textsubscript{N} data to fidelity estimates. For this table
we used the average heavy output frequency over
all circuits and the expected ideal heavy output
probability from Sec. 4.1 instead of a per circuit
estimate.
7.2 Quantum error correction
It is widely believed that quantum computers will
require quantum error correction (QEC) to reach
error rates necessary to perform large-scale quant-
num computation [55]. QEC works by encoding
quantum information into logical qubits, which
are constructed from many physical qubits, with
a QEC code. There are several different propos-
als for QEC codes but they all use physical qubits
to detect and correct certain errors in the logical
qubits without destroying the underlying quant-
num information. A QEC code is partially de-
ined by its distance \(d\), which roughly indicates
| Qubits | Heavy output frequency | Estimated circuit fidelity | Reference | System                  |
|--------|------------------------|---------------------------|-----------|------------------------|
| 2      | 2/3                    | 0.6771                    | -         | -                      |
| 2      | 0.718(6)               | 0.8086(154)               | Cross et al. [3] | IBM Tokyo            |
| 2      | 0.7758(417)            | 0.9567(1069)              | Pino et al. [4] | Honeywell System Model H0 |
| 3      | 2/3                    | 0.5433                    | -         | -                      |
| 3      | 0.729(7)               | 0.6997(176)               | Cross et al. [3] | IBM Johannesburg     |
| 3      | 0.8328(373)            | 0.9603(936)               | Pino et al. [4] | Honeywell System Model H0 |
| 4      | 2/3                    | 0.5223                    | -         | -                      |
| 4      | 0.699(1)               | 0.6114(26)                | Cross et al. [3] | IBM Johannesburg     |
| 4      | 0.7677(422)            | 0.8010(1165)              | Pino et al. [4] | Honeywell System Model H0 |
| 5      | 2/3                    | 0.4841                    | -         | -                      |
| 5      | 0.69(1)                | 0.5475(271)               | Sundaresan et al. [5] | IBM Johannesburg |
| 6      | 2/3                    | 0.4826                    | -         | -                      |
| 6      | 0.7296(222)            | 0.6589(622)               | Pino et al. [4] | Honeywell System Model H0 |
| 6      | 0.701(31)              | 0.579(87)                 | Jurecic et al. [6] | IBM Montreal           |
| 7      | 2/3                    | 0.4708                    | -         | -                      |
| 7      | 0.7178(159)*           | 0.6129(442)               | [7]       | Quantumum System Model H1-1 |
| 7      | 0.69(1)*†              | 0.54(3)                   | [8]       | IBM Montreal           |
| 9      | 2/3                    | 0.4737                    | -         | -                      |
| 9      | 0.7332(255)*           | 0.6620(723)               | [9]       | Quantumum System Model H1-1 |
| 10     | 2/3                    | 0.4788                    | -         | -                      |
| 10     | 0.7036(102)            | 0.5846(293)               | Fig. 13, [10] | Quantumum System Model H1-1 |

Table 2: Heavy output frequency and estimated circuit fidelity of all experimentally passed QVT$_N$ with reported values as of writing. Uncertainty is reported based on the original confidence interval in Ref. [3] since it is the only estimate available for most data; however, it is generally larger than what we find from our new method in Sec. 6. For each dataset we use the estimated value of $h_{ideal}(N)$ from numerical simulations in Sec. 4.1. * Data was provided by company and not published or in preprint at time of writing. † Value is estimated from referenced plot but not confirmed.

the number of physical errors a code can tolerate and correct. Broadly speaking, the number of qubits needed to implement a QEC code grows polynomially with the code’s distance. QEC consists of structured and repetitive circuits, seemingly quite different than QVT circuits. Here, we compare a QEC code’s logical pseudo-threshold to QVT$_N$ passing thresholds.

We define the pseudo-threshold of a QEC code as the point where the logical error rate is equal to the largest physical error rate. For example, with the $TQ$ Depolarizing model, the pseudo-threshold is the point that the logical error rate is equal to the two-qubit depolarizing error rate. Ideally, a system implementing a QEC code will operate well below the pseudo-threshold to take advantage of the error suppression. In fact, in a real system the pseudo-threshold will be difficult to measure since it requires full knowledge of all error sources. However, in simulations it can be well defined, and it is a natural performance metric of different codes to compare with QVT requirements.

To probe the relationship between a system’s ability to pass QVT$_N$ and its expected performance of QEC codes, we ran simulations for three different QEC codes: the [[7,1,3]] Steane code [56], the rotated surface code [57], and the toric code [58]. We applied three error models introduced in Sec. 5.3: $TQ$ depolarizing, Measurement, and Memory. For the [[7,1,3]] Steane code, the simulation was done using a stabilizer simulator with a lookup table style decoder [59] and for the rotated surface and toric codes, a fast Pauli tracking simulator was used with minimum weight perfect matching to decode error syndromes [60, 61]. These choices allow us to investigate the relationship between QVT and QEC for both low and higher distance codes.

Fig. 15 shows the error magnitude required to pass QVT$_N$ compared to the error magnitude required to reach the pseudo-threshold for the Steane code with $d = 3$, the rotated surface code with $d = 3$, 5, and 7, and the toric code with $d = 3$, 5, and 7. The estimated QVT$_N$ passing thresholds does match the pseudo-threshold for a few codes and error models (e.g., distance three surface with the $TQ$ depolarizing model and Steane with the Measurement model). However, most pseudo-threshold points fall outside of the
Figure 15: QVT passing error magnitudes compared to QEC pseudo-thresholds with three different error models. Colors correspond to error models and line styles and labels correspond to different thresholds and codes. The QVT passing threshold is estimated from the scalable model with the dep option (solid lines). The three QEC codes tested are: rotated surface code for $d = 3$, 5, and 7 (circles with finely dashed lines), toric code for $d = 3$, 5, and 7 (squares with dashed lines), and [[7,1,3]] Steane code (triangles). Three error models were tested are $TQ$ depolarizing (blue), Measurement (red), and Memory (grey). As qubit number increases, the error magnitude required to pass QV exponentially decays, whereas the error magnitude required for a QEC code to reach the pseudo-threshold increases.

QVT$_N$ passing estimates. Therefore, we do not find that passing QVT$_N$ implies being able to reach the pseudo-threshold for a specific QEC code.

QVT$_N$ may not be predictive for a specific code but the qubit and fidelity requirements do roughly align well with a general class of small-distance codes. The pseudo-thresholds for distance three codes roughly fall within a region of $9 < N < 30$ and $4 \times 10^{-2} \leq \epsilon \leq 2 \times 10^{-4}$ for all error models tested and the QVT$_N$ passing thresholds intersect this region. Designing and verifying that a machine can pass QVT$_N$ within this region would provide a reasonable starting point for testing a variety of small-distance codes.

Furthermore, since QVT$_N$ requires arbitrary connectivity then passing QVT$_N$ within the region defined above implies many codes are available to test. We simulated example codes with nearest-neighbor parity measurements but in principle one may want to test other codes, such as LDPC codes using non-local parity checks [62] or randomly generated codes [63]. Verifying low error rate connections means such codes should be feasible to implement and compare.

Scaling QVT$_N$ to larger $N$ necessarily requires lowering error rates but scaling QEC to larger distances, which also requires more qubits, actually alleviates requirements on error rates to reach pseudo-thresholds. This is shown with the rotated surface and toric codes in Fig. 15, which have pseudo-threshold error magnitudes which increase (are easier to achieve) with larger $N$. Thus there is a crossover regime after which passing QVT$_N$ becomes more difficult than reaching the pseudo-threshold for a QEC code with the same number of qubits. Lowering error rates is always beneficial for QEC but not strictly necessary after the crossover regime.

QEC requires additional features that are not necessary to run QVT. In order to implement QEC a quantum computer needs to be able to apply mid-circuit measurements and resets to measure errors and, ideally, feed-forward operations to correct errors [64]. QVT does not require either of these features so even if a quantum computer can implement and pass QVT$N_{30}$, for example, it will not necessarily be able to run QEC.

Another notable difference between QVT and QEC is the dependence on type of error. While we have not simulated every type of error for QVT we did observe in Sec. 5 that many different types of errors have similar effects on the passing threshold of QVT. However, the same is not true for QEC where some types of errors cannot be corrected in certain codes. For example, leakage errors map population outside the qubit manifold and can rapidly accumulate and spread additional errors, breaking the fault tolerance of the code [65, 66]. Coherent errors are known to have a larger impact on smaller distance codes compared to larger distance codes [67]. Proving an implementation of a QEC code is below the pseudo-threshold requires probing several different basis states to confirm an arbitrary state is below the pseudo-threshold [68]. Any asymmetry in the errors will affect the basis states differently and possibly cause certain states to not meet pseudo-threshold.

We did not compare such errors because in practice they can be mitigated by circuit design [69, 70, 71, 72, 73, 74, 75] or physical techniques [76]. Moreover, we do not anticipate these errors to be the leading order errors in near-term systems where QVT is applicable. We leave such detailed comparisons to future work.

Presently, both QVT and QEC demonstrations
are well aligned with near term goals of increasing qubit number and decreasing error rates. However, as devices continue to mature QVT tests will no longer be classically simulatable and also harder to pass while QEC will necessarily be required to scale to larger qubit numbers.

8 Conclusions

Our work illuminates previously unstudied behavior of QVT and requirements for scaling to larger $N$. We first considered how circuit construction impacts the test results. Even without errors the ideal heavy output probabilities scale with the qubit number, which has a notable impact for $N < 10$. The standard optimizations used on the circuits also have a significant effect for $N < 10$, which reduces two qubit gates by at least 20%, but is less effective as $N$ increases. Next, we preformed a series of simulations to test the behavior of QVT$_N$ with different error sources. We observed that QVT$_N$ passing is mostly dependent on total error magnitude (including errors like crosstalk) and not on the error sources. We constructed a scalable method for larger qubit numbers that roughly estimates error requirements for passing QVT$_N$. After, we studied the confidence interval construction for QVT$_N$ from Ref. [3] and found that the method returned much higher coverage than specified for most experiments except when run with a single shot per circuit where the coverage was lower than specified. We proposed a new method that returns tighter confidence intervals and showed it had near the expected coverage with more than one shot and 100 circuits. Finally, we compared QVT$_N$ results to other important quantum computing applications. We showed that the heavy output probability can be converted to serve as an estimate that scales with the average state preparation fidelity although is generally slightly higher. We also numerically demonstrated that the fidelity requirements for QVT$_N$ for $9 < N < 30$ roughly align with the fidelity requirements for many low-distance break-even QEC demonstrations. There is one obvious question left out of the FAQ’s for QVT in Sec. 3; “Is QVT a good benchmark?” This is clearly a complicated question with a variety of opinions but we observe that most disagreements come down to two main questions about full system benchmarking:

**Q8:** Is random circuit construction a reasonable way to benchmark systems?

**A8:** The random circuit construction for QVT captures the effects of many different error sources (as seen in Sec. 5) but lead to previously unknown irregular performance effects dependent on qubit number — as shown in Sec. 4. Capturing different error sources is crucial to near-term benchmarking since many systems suffer from errors that are missed in individual component benchmarks and usually not well understood. The irregular performance diminishes for larger qubit number and do not seem likely to be a problem for future QVT$_N$ experiments ($N > 10$). Another downside we identified is that QVT seems to mostly scale with the total error magnitude (infidelity). This may mean some errors, like coherent errors, may have different effects in QVT than in certain algorithms. This is a typical downside of random circuit benchmarking but allows the results to better relate to circuit fidelity (as studied in Sec. 7.1). Finally, QVT requires high-fidelity interaction between arbitrary qubits, which is not required for many algorithms and QEC codes. Nevertheless, such high-fidelity arbitrary interactions are useful for near-term devices to test and compare a variety of algorithms or QEC codes, which may have widely different connectivity requirements.

**Q9:** Are square circuits the best choice for judging a quantum computer’s performance?

**A9:** While QVT circuits have linear depth, the fidelity requirements to pass QVT$_N$ for $9 < N < 30$ match well with other goals for quantum computation, especially QEC. Non-Markovian system errors that occur in longer circuits are missed in QVT$_N$, which is one downside to the test, but square scaling balances fidelity and qubit requirements in a reasonable way for near-term goals.

Overall, we believe our work supports the notion that QVT is a good benchmark, with the above caveats, but QVT is certainly not the only or final answer to full system benchmarking of quantum computers. In practice, it is best to use multiple benchmarks that stress different circuit sizes and errors to fully judge a systems performance. Ultimately, we expect a suite of benchmarks with comprehensive studies — like we at-
tempted here — will serve as standards for comparing different systems. Moreover, QVT in its current form will not be useful for more than $\sim 30$ qubits and as platforms move towards QEC the need to scale qubit number will outweigh the need to scale fidelity. However, we find that currently QVT does set worthwhile near-term goals for performance demonstrations that measure system level errors.

Acknowledgments

We would like to thank the entire Quantinuum team for helpful input and questions that inspired this work. We especially thank the Quantinuum System Model H0 and H1 teams and other contributors for running the quantum volume tests. Specifically Dan Gresh, Aaron Hankin, Kevin Gilmore, Justin Gerber, John Gaebler, David Francois, Thomas Gatterman, Si Khadir Halit, Alex Hall, Justin Bohnet and Brian Neyenhuis who contributed to the QVT$_{10}$ data presented.

References

[1] Lev S Bishop, Sergey Bravyi, Andrew Cross, Jay M Gambetta, and John Smolin. “Quantum volume”. Quantum Volume. Technical Report (2017).

[2] Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn, Abhinav Kandala, Antonio Mezzacapo, Peter Müller, Walter Riess, Gian Salis, John Smolin, Ivano Tavernelli, and Kristan Temme. “Quantum optimization using variational algorithms on near-term quantum devices”. Quantum Science and Technology 3, 030503 (2018).

[3] Andrew W. Cross, Lev S. Bishop, Sarah Sheldon, Paul D. Nation, and Jay M. Gambetta. “Validating quantum computers using randomized model circuits”. Phys. Rev. A 100, 032328 (2019).

[4] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis. “Demonstration of the trapped-ion quantum ccd computer architecture”. Nature 559, 209–213 (2021).

[5] Neereja Sundaresan, Isaac Lauer, Emily Pritchett, Easwar Magesan, Petar Jurcevic, and Jay M. Gambetta. “Reducing unitary and spectator errors in cross resonance with optimized rotary echoes”. PRX Quantum 1, 020318 (2020).

[6] Petar Jurcevic, Ali Javadi-Abhari, Lev S Bishop, Isaac Lauer, Daniela F Bogorin, Markus Brink, Lauren Capelluto, Oktay Gülüklü, Toshinari Itoko, Naoki Kanazawa, Abhinav Kandala, George A Keefe, Kevin Krsulich, William Landers, Eric P Lewandowski, Douglas T McClure, Giacomo Nannicini, Aedin Nara-gond, Hasan M Nayefeh, Emily Pritchett, Mary Beth Rothwell, Srikanth Srinivasan, Neereja Sundaresan, Cindy Wang, Ken X Wei, Christopher J Wood, Jeng-Bang Yau, Eric J Zhang, Oliver E Dial, Jerry M Chow, and Jay M Gambetta. “Demonstration of quantum volume 64 on a superconducting quantum computing system”. Quantum Science and Technology 6, 025020 (2021).

[7] “Achieving quantum volume 128 on the Honeywell quantum computer” (2020).

[8] “IBM achieves a new quantum volume level of 128” (2021).

[9] “Honeywell sets new record for quantum computing performance” (2021).

[10] “Honeywell sets another record for quantum computing performance” (2021).

[11] Isaac L. Chuang and M. A. Nielsen. “Prescription for experimental determination of the dynamics of a quantum black box”. Journal of Modern Optics 44, 2455–2467 (1997).

[12] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland. “Randomized benchmarking of quantum gates”. Phys. Rev. A 77, 012307 (2008).

[13] Easwar Magesan, Jay M. Gambetta, and Joseph Emerson. “Characterizing quantum gates via randomized benchmarking”. Phys. Rev. A 85, 042311 (2012).

[14] Thomas Lubinski, Sonika Johri, Paul Varosy, Jeremiah Coleman, Luning Zhao, Jason Necaise, Charles H. Baldwin, Karl Mayer, and Timothy Proctor. “Application-
oriented performance benchmarks for quantum computing” (2021). arXiv:2110.03137.

[15] Timothy Proctor, Kenneth Rudinger, Kevin Young, Erik Nielsen, and Robin Blume-Kohout. “Measuring the capabilities of quantum computers”. Nature Physics 18, 75–79 (2021).

[16] Mohan Sarovar, Timothy Proctor, Kenneth Rudinger, Kevin Young, Erik Nielsen, and Robin Blume-Kohout. “Detecting crosstalk errors in quantum information processors”. Quantum 4, 321 (2020).

[17] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven. “Characterizing quantum supremacy in near-term devices”. Nature Physics 14, 595–600 (2018).

[18] Alexander Erhard, Joel J. Wallman, Lukas Postler, Michael Meth, Roman Stricker, Esteban A. Martinez, Philipp Schindler, Thomas Monz, Joseph Emerson, and Rainer Blatt. “Characterizing large-scale quantum computers via cycle benchmarking”. Nat. Comm. 10 (2019).

[19] Timothy J. Proctor, Arnaud Carignan-Dugas, Kenneth Rudinger, Erik Nielsen, Robin Blume-Kohout, and Kevin Young. “Direct randomized benchmarking for multiqubit devices”. Phys. Rev. Lett. 123 (2019).

[20] Robin Harper, Steven T. Flammia, and Joel J. Wallman. “Efficient learning of quantum noise”. Nature Physics 16, 1184–1188 (2020).

[21] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielowski, C. Collins, K. M. Hudek, J. Mizrahi, J. D. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. DUCore, A. Blinov, S. M. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim. “Benchmarking an 11-qubit quantum computer”. Nat. Comm. 10 (2019).

[22] Arjan Cornelissen, Johannes Bausch, and András Gilyén. “Scalable benchmarks for gate-based quantum computers” (2021). arXiv:2104.10698.

[23] Scott Aaronson and Lijie Chen. “Complexity-theoretic foundations of quantum supremacy experiments” (2016). arXiv:1612.05903.

[24] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, and et al. “Quantum supremacy using a programmable superconducting processor”. Nature 574, 505–510 (2019).

[25] R. Oliveira, O. C. O. Dahlsten, and M. B. Plenio. “Generic entanglement can be generated efficiently”. Phys. Rev. Lett. 98 (2007).

[26] Aram W. Harrow and Richard A. Low. “Random quantum circuits are approximate 2-designs”. Commun. Math. Phys. 291, 257–302 (2009).

[27] F. G. S. L. Brandão, A. W. Harrow, and M. Horodecki. “Local random quantum circuits are approximate polynomial-designs”. Commun. Math. Phys. 346, 397–434 (2016).

[28] Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. “Exact and approximate unitary 2-designs and their application to fidelity estimation”. Phys. Rev. A 80, 012304 (2009).

[29] Sean Mullane. “Sampling random quantum circuits: a pedestrian’s guide” (2020). arXiv:2007.07872.

[30] Don N. Page. “Average entropy of a subsystem”. Phys. Rev. Lett. 71, 1291–1294 (1993).

[31] “Nist/sematech e-handbook of statistical methods”. Accessed: 2022-03-14.

[32] David Callan. “A combinatorial survey of identities for the double factorial” (2009). arXiv:0906.1317.

[33] Fred S. Roberts and Barry Tesman. “Applied combinatorics 2nd ed.”. CRC Press. (2009).

[34] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik. “The theory of variational hybrid quantum-classical algorithms”. New J. Phys. 18, 023023 (2016).

[35] E. Farhi and J. Goldstone. “A quantum approximate optimization algorithm” (2014). arXiv:1411.4028.

[36] D. Coppersmith. “An approximate fourier transform useful in quantum factoring” (2002). arXiv:quant-ph/0201067.

[37] Navin Khaneja, Roger Brockett, and Stefan J. Glaser. “Time optimal control in spin systems”. Phys. Rev. A 63, 032308 (2001).
[38] Farrokh Vatan and Colin Williams. “Optimal quantum circuits for general two-qubit gates”. Phys. Rev. A 69, 032315 (2004).
[39] Anders Sørensen and Klaus Mølmer. “Entanglement and quantum computation with ions in thermal motion”. Phys. Rev. A 62 (2000).
[40] H. Abraham and et al. “Qiskit: An open-source framework for quantum computing” (2019).
[41] Jun Zhang, Jiri Vala, Shankar Sastry and K. Birgitta Whaley. “Geometric theory of nonlocal two-qubit operations”. Phys. Rev. A 67, 042313 (2003).
[42] C. H. Baldwin “qvtsim GitHub repository.” Accessed: 2022-03-22.
[43] Michael A. Nielsen and Isaac L. Chuang. “Quantum computation and quantum information”. Cambridge University Press. (2000).
[44] Alexei Gilchrist, Nathan K. Langford, and Michael A. Nielsen. “Distance measures to compare real and ideal quantum processes”. Phys. Rev. A 71, 062310 (2005).
[45] Arnaud Carignan-Dugas, Joel J Wallman, and Joseph Emerson. “Bounding the average gate fidelity of composite channels using the unitarity”. New Journal of Physics 21, 053016 (2019).
[46] Daniel Greenbaum. “Introduction to quantum gate set tomography” (2015). arXiv:1509.02921.
[47] C. H. Baldwin, B. J. Bjork, J. P. Gaebler, D. Hayes, and D. Stack. “Subspace benchmarking high-fidelity entangling operations with trapped ions”. Phys. Rev. Research 2, 013317 (2020).
[48] Filip B. Maciejewski, Zoltán Zimborás, and Michał Oszmaniec. “Mitigation of readout noise in near-term quantum devices by classical post-processing based on detector tomography”. Quantum 4, 257 (2020).
[49] Wassily Hoeffding. “On the Distribution of the Number of Successes in Independent Trials”. The Annals of Mathematical Statistics 27, 713 – 721 (1956).
[50] Adam M. Meier. “Randomized benchmarking of clifford operators”. PhD thesis. University of Colorado. (2006).
[51] Bradley Efron and Robert J. Tibshirani. “An introduction to the bootstrap”. CRC Press. (1994).
[52] Robin Blume-Kohout, Marcus P. da Silva, Erik Nielsen, Timothy Proctor, Kenneth Rudinger, Mohan Sarovar, and Kevin Young. “A taxonomy of small markovian errors” (2021). arXiv:2103.01928.
[53] Jin-Sung Kim, Lev S. Bishop, Antonio D. Córcoles, Seth Merkel, John A. Smolin, and Sarah Sheldon. “Hardware-efficient random circuits to classify noise in a multiqubit system”. Phys. Rev. A 104, 022609 (2021).
[54] Yunchao Liu, Matthew Otten, Roozbeh Bassirianjahromi, Liang Jiang, and Bill Fef-ferman. “Benchmarking near-term quantum computers via random circuit sampling” (2021). arXiv:2105.05232.
[55] Simon J Devitt, William J Munro, and Kae Nemoto. “Quantum error correction for beginners”. Reports on Progress in Physics 76, 076001 (2013).
[56] A. M. Steane. “Simple quantum error-correcting codes”. Physical Review A 54, 4741–4751 (1996).
[57] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. “Surface codes: Towards practical large-scale quantum computation”. Phys. Rev. A 86 (2012).
[58] Sergey B Bravyi and A Yu Kitaev. “Quantum codes on a lattice with boundary” (1998). arXiv:quant-ph/9811052.
[59] Ciaran Ryan-Anderson “PECOS: Performance estimator of codes on surfaces”. pecos GitHub repository. Accessed: 2021-10-01.
[60] Jack Edmonds. “Paths, trees, and flowers”. Canadian Journal of mathematics 17, 449–467 (1965).
[61] Austin G. Fowler. “Minimum weight perfect matching of fault-tolerant topological quantum error correction in average o(1) parallel time”. Quantum Info. Comput. 15, 145–158 (2015).
[62] Zunaira Babar, Panagiotis Botsinis, Dimitrios Alanis, Soon Xin Ng, and Lajos Hanzo. “Fifteen years of quantum ldpc coding and improved decoding strategies”. IEEE Access 3, 2492–2519 (2015).
[63] Michael J. Gullans, Stefan Krastanov, David A. Huse, Liang Jiang, and Steven T. Flammia. “Quantum coding with low-depth random circuits”. Phys. Rev. X 11, 031066 (2021).
[64] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. François, A. Chernoguzov, D. Lucchetti, N. C. Brown, T. M. Gatterman, S. K. Halit, K. Gilmore, J. A. Gerber, B. Neyenhuis, D. Hayes, and R. P. Stutz. “Realization of real-time fault-tolerant quantum error correction”. Phys. Rev. X 11, 041058 (2021).

[65] Martin Suchara, Andrew W. Cross, and Jay M. Gambetta. “Leakage suppression in the toric code”. Quantum Info. Comput. 15, 997–1016 (2015).

[66] Natalie C. Brown and Kenneth R. Brown. “Comparing zeeman qubits to hyperfine qubits in the context of the surface code: $^{174}$Yb$^+$ and $^{171}$Yb$^+$. Phys. Rev. A 97, 052301 (2018).

[67] Joseph K Iverson and John Preskill. “Coherence in logical quantum channels”. New Journal of Physics 22, 073066 (2020).

[68] Daniel Gottesman. “Quantum fault tolerance in small experiments” (2016). arXiv:1610.03507.

[69] Natalie C. Brown, Andrew Cross, and Kenneth R. Brown. “Critical faults of leakage errors on the surface code”. In 2020 IEEE International Conference on Quantum Computing and Engineering (QCE). Pages 286–294. (2020).

[70] Natalie C Brown, Michael Newman, and Kenneth R Brown. “Handling leakage with subsystem codes”. New Journal of Physics 21, 073055 (2019).

[71] Bichen Zhang, Swarnadeep Majumder, Pak Hong Leung, Stephen Crain, Ye Wang, Chao Fang, Dripto M. Debroy, Jungsang Kim, and Kenneth R. Brown. “Hidden inverses: Coherent error cancellation at the circuit level”. Phys. Rev. Applied 17, 034074 (2022).

[72] Pedro Parrado-Rodríguez, Ciarán Ryan-Anderson, Alejandro Bermudez, and Markus Müller. “Crosstalk suppression for fault-tolerant quantum error correction with trapped ions”. Quantum 5, 487 (2021).

[73] David K. Tuckett, Stephen D. Bartlett, and Steven T. Flammia. “Ultrahigh error threshold for surface codes with biased noise”. Phys. Rev. Lett. 120, 050505 (2018).

[74] Joel J. Wallman and Joseph Emerson. “Noise tailoring for scalable quantum computation via randomized compiling”. Phys. Rev. A94 (2016).

[75] Dripto M. Debroy, Muyuan Li, Michael Newman, and Kenneth R. Brown. “Stabilizer slicing: Coherent error cancellations in low-density parity-check stabilizer codes”. Phys. Rev. Lett. 121, 250502 (2018).

[76] D. Hayes, D. Stack, B. Bjork, A. C. Potter, C. H. Baldwin, and R. P. Stutz. “Eliminating leakage errors in hyperfine qubits”. Phys. Rev. Lett. 124, 170501 (2020).