

Performance Evaluations of Noisy Approximate Quantum Fourier Arithmetic

Robert A.M. Basili,^{1,2} Wenyang Qian,^{1,2} Shuo Tang,² Austin M. Castellino,³
Mary Eshaghian-Wilner,¹ James P. Vary,² Glenn Luecke,¹ and Ashfaq Khokhar⁴

¹*Department of Mathematics, Iowa State University, Ames, IA 50011, USA*

²*Department of Physics and Astronomy,
Iowa State University, Ames, IA 50011, USA*

³*Liberal Arts and Sciences Online, Iowa State University, Ames, IA 50011, USA*

⁴*Department of Electrical and Computer Engineering,
Iowa State University, Ames, IA 50011, USA*

(Dated: December 20, 2021)

Abstract

The Quantum Fourier Transform (QFT) grants competitive advantages, especially in resource usage and circuit approximation, for performing arithmetic operations on quantum computers, and offers a potential route towards a numerical quantum-computational paradigm. In this paper, we utilize efficient techniques to implement QFT-based integer addition and multiplications. These operations are fundamental to various quantum applications including Shor’s algorithm, weighted sum optimization problems in data processing and machine learning, and quantum algorithms requiring inner products. We carry out performance evaluations of these implementations based on IBM’s superconducting qubit architecture using different compatible noise models. We isolate the sensitivity of the component quantum circuits on both one-/two-qubit gate error rates, and the number of the arithmetic operands’ superposed integer states. We analyze performance, and identify the most effective approximation depths for quantum add and quantum multiply within the given context. We observe significant dependency of the optimal approximation depth on the degree of machine noise and the number of superposed states in certain performance regimes. Finally, we elaborate on the algorithmic challenges - relevant to signed, unsigned, modular and non-modular versions - that could also be applied to current implementations of QFT-based subtraction, division, exponentiation, and their potential tensor extensions. We analyze performance trends in our results and speculate on possible future development within this computational paradigm.

I. INTRODUCTION

Since its proposal in 1994, the development of Shor’s Algorithm [33] for factoring numbers in polynomial time has drawn intense interest in quantum computing, and recent technological leaps have quelled many of the doubts around its practical feasibility. While many challenges must be overcome before fault-tolerant quantum computing becomes remotely possible, the full range of cases where quantum advantage might be achieved remains largely speculative.

Presently, problem-solving techniques in quantum computing on Noisy Intermediate-Scale Quantum (NISQ) devices are generally considered within the context of a computational basis of qubits acted upon by quantum logic circuits [30]. In this paradigm, one generally maps a space of possible solutions to bit strings, and uses unitary operations (abstracted as quantum logic circuits) to operate on them. The mapping itself may be in some sense arbitrary, but practically must be designed to utilize machine-efficient unitaries.

However, just as the problem-solving paradigms used in classical computing historically advanced beyond binary representations as resources became more plentiful and reliable, the potential roles of quantum and/or quantum-classical hybrid data structures remain largely unknown. While this larger question extends beyond the scope of this work, we conduct these calculations in part as

progenitorial steps towards the potential realization of such a numerical quantum-computational paradigm. For more immediate purposes, we also seek to elucidate practical, near-term limitations of current quantum devices, explore the effectiveness of a class of approximate circuits, and garner insight on what degree of that approximation is optimal for a given regime of machine performance.

Specifically, we consider quantum arithmetic performed with the Quantum Fourier Transform (QFT) and approximate QFT (AQFT). The QFT, a phase-estimation algorithm, is the source of speed-up in Shor’s algorithm, and provides several benefits for enabling quantum arithmetic. The approximate QFT improves the algorithm’s performance and fidelity on noisy machines by removing gates whose impact on fidelity is less than the noise their inclusion imposes. Beyond quantum arithmetic’s role in Shor’s algorithm, its unique advantage of allowing many operations in parallel presents potential opportunities for a host of problems in data processing, machine learning, and more [26].

Although the anticipated utility of quantum arithmetic has given rise to many theoretical implementations and provided great incentive to explore its usage, effectively gauging an implementation’s performance and identifying sources of error remains a significant obstacle. Barring recent years, considerations of algorithm performance remained fairly limited beyond its theoretical discussion [21]. Further, despite recent access to NISQ hardware, the small number of qubits, multiple sources of noise, and the relatively large circuit sizes of quantum arithmetic, have made progress in studying performance extremely limited. Ref. [28] provides performance results of quantum addition of up to 2-bit integers on the IBM Quantum superconducting-qubit testbed, and observes extremely limited fidelity with current hardware. While not specific to arithmetic, Ref. [27] provides a robust breakdown of various factors affecting a gate-based algorithm, including state preparation, oracle, connectivity, circuit rewriting, transpiling, and readout, and designed a calibration matrix to quantitatively evaluate the quantum error.

While the technology will need to improve significantly before quantum arithmetic can be examined directly in much depth, the study of quantum noise in current NISQ hardware has allowed rapid advancement in the study of quantum noise and the development of tunable noise models. These models offer the unique opportunity to garner insight into the implementation and performance of quantum arithmetic algorithms by modeling the isolated impact of different sources of noise. In this work, we simulate the dependency of quantum Fourier Addition (QFA) and Quantum Fourier Multiplication (QFM) on one- and two-qubit (1q- and 2q-) gate error rates, on approximation depth in the QFT, and on the number of superposed integer states involved in the operands (which we will refer to as a qinteger’s order of superposition). We use noise models designed to

reflect the current performance of IBM superconducting quantum computers (though with qubit counts and connectivity not currently available). We include perfect, noise-free simulations as a point of comparison to differentiate the noise introduced by the approximate QFT. The isolation of other sources of error, such as thermal relaxation and qubit measurement, their simultaneous simulation with 1q-/2q- gate errors, as well as the impact of error mitigation and extrapolation from integers to real numbers, are deferred to a future work. In this fashion, the current work reflects a preliminary effort to help assess the practical challenges that must be overcome before a quantum numerical paradigm can be realistically explored. Finally, though not discussed, modular and non-modular forms of QFT-based quantum integer subtraction, division, exponentiation, and extensions to vector, matrix, and other tensor operations, rely on slight alterations to the same underlying algorithm, and, consequently, many of the considerations discussed here apply to them as well [18, 19, 23, 31, 34]. These results may also provide insight regarding the QFT, AQFT, and quantum algorithms that rely upon them.

This paper is organized as follows. Sec. II briefly describes our implementation of the AQFT. In Sec. III we illustrate Quantum Fourier Addition and Quantum Fourier multiplication, and describe our implementation. In Sec. IV, we provide specific details on the generation, collection, and presentation of performance data considered in this study, and discuss their interpretation. Finally Sec. V summarizes this work and discusses its future development.

II. PHASE-ENCODING AND THE QUANTUM FOURIER TRANSFORM

We begin by specifying the quantum integers (qintegers) discussed in this work as a mathematical construct, and how we encode them as a quantum data structure. We consider the space of integer states $|i\rangle$ with $i \in \mathbb{Z}$. A qinteger y is thus defined as a superposition of integer states,

$$|y\rangle = \sum_i^k p_i |i\rangle,$$

where

$$\sum_i^k p_i^2 = 1$$

is the sum of all integer probabilities, p_i^2 , and must add to unity. We refer to a superposition of j unique integers with j respective nonzero probability amplitudes as an order- j qinteger.

The representation of qintegers can be achieved by several means, and their optimal implementation depends on the computational model required [24]. For our purposes, we constrain

the current discussion to dynamic qintegers - that is, qintegers whose values can be updated by methods without collapsing superposition - realized on quantum registers of n qubits. We encode qintegers in two's complement, though the results of this work may be extended to fixed point binary representations of numbers in general. The maximum order of superposition for an n -qubit qinteger is therefore limited by the size of the register's computational basis, $N = 2^n$, or an order- N qinteger. We facilitate our discussion using the quantum circuit paradigm as our starting point.

Similar to how one may encode integers classically in the computational basis as bitstrings, quantum superposition enables one to encode numbers in the relative phases of qubits. Since linear phase shifts are a fundamental operation of most quantum computers, this encoding is particularly convenient for performing basic arithmetic operations.

We refer to this basis of relative phases as the Quantum Fourier Basis (QFB). The QFT is an $\mathcal{O}(n \log_2 n)$ quantum algorithm that performs a Fourier Transform of quantum mechanical amplitudes, and serves to transform numbers encoded in the computational basis to the QFB. Whereas the discrete Fourier Transform over classical fields involves an expansion that mixes vectors of complex numbers, the QFT involves the expansion of the computational basis $|y\rangle = |0\rangle, |1\rangle, \dots, |N-1\rangle$ into mixtures of QFB states $|k\rangle = |\phi(0)\rangle, |\phi(1)\rangle, \dots, |\phi(N-1)\rangle$ (or vice-versa), as

$$\mathcal{QFT} |y\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i2\pi yk/N} |k\rangle \quad (1)$$

for initial integer state $|y\rangle$ encoded in the computational basis with n qubits and $N = 2^n$ unique states. The corresponding inverse QFT is defined

$$\mathcal{QFT}^{-1} |k\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{-i2\pi yk/N} |y\rangle. \quad (2)$$

When we apply the QFT to the state $|y\rangle = |y_1\rangle \otimes |y_2\rangle \otimes \dots \otimes |y_n\rangle$ (or as a binary string $y = "y_n y_{n-1} \dots y_1"$ such that $y = y_1 2^0 + y_2 2^1 + \dots + y_n 2^{n-1}$), one obtains the result:

$$\begin{aligned} \mathcal{QFT} |y\rangle &= \frac{1}{2^{n/2}} \bigotimes_{q=1}^n \left(|0\rangle + e^{i2\pi (\sum_{k=1}^q y_k / 2^{q-k+1})} |1\rangle \right)_q \\ &= \frac{1}{2^{n/2}} \bigotimes_{q=1}^n \left(|0\rangle + e^{i2\pi [0.y]_{q,1}} |1\rangle \right)_q, \end{aligned} \quad (3)$$

where the parenthetical subscript on the right-hand side denotes the q th qubit's state, and we rewrite the summation in the exponent using the binary-fraction shorthand $[0.y]_{i,j} = 0.y_i y_{i-1} \dots y_j = y_i/2 + y_{i-1}/4 + \dots + y_j/2^{i-j+1}$ for convenience.

If we consider from this the action of the QFT on each qubit, one notices that, after applying the QFT, the first qubit (which originally had the state $|y_1\rangle$) will possess a state where the phase summation on the right-hand side of Eq. (3) reduces to only one term, $[0.y]_{1,1} = y_1/2$. Notably, this means the first qubit's final state only depends on its own initial state. Meanwhile, the y_2 qubit will have two terms, $[0.y]_{2,1} = y_2/2 + y_1/4$, for its sum of phases, and thus depends on the initial states of both itself and the first qubit. Finally, the n th qubit's final state will involve a sum of n terms, and depends on the initial states of all n qubits. If we consider each term of these summations as a single phase rotation being controlled by a single qubit, the first qubit requires one such rotation, the second requires two, and so on. Further, the first rotation on a given qubit is controlled by itself, while each subsequent rotation is controlled by each qubit that comes before it.

To implement the QFT described by Eq. (3) as a quantum circuit, we must construct the Fourier Transform in terms of unitaries. We therefore introduce the two-qubit controlled rotation phase gate R_l and one-qubit Hadamard gate H with which we shall construct the circuit. Using the basis ordering common to most textbooks for our matrix representation of quantum operators (sometimes called big-endian convention), we may write the unitary transformations

$$R_l = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{\frac{i2\pi}{2^l}} \end{bmatrix}, H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad (4)$$

where it is understood the dimensions span the corresponding Hilbert space specific to the given qubit(s) that the operator acts upon. Both gates can be efficiently constructed with only one or two gates of the universal gate set used by current IBM superconducting quantum [22].

To perform the QFT on a single n -qubit qinteger, we first apply the Hadamard gate to each qubit, followed by consecutive controlled- R_l gates as shown in Fig. 1. Repeating the procedure for all the qubits in the qinteger's register and including the normalization factor of $1/\sqrt{2}$, we obtain the qubit state shown on the right in Eq. (3).

When a large number of qubits are involved, the conditional rotation $R_{l \rightarrow \infty} \rightarrow I$ of the QFT approaches the identity. On one hand, encoding in the QFB therefore requires that the accuracy of the phase shift performed by the physical quantum computer increases exponentially with n to ensure addition via phase shifts remains truly linear at all scales. On the other hand, one rarely requires such accuracy for phase rotations between bits of sufficiently different binary order in most

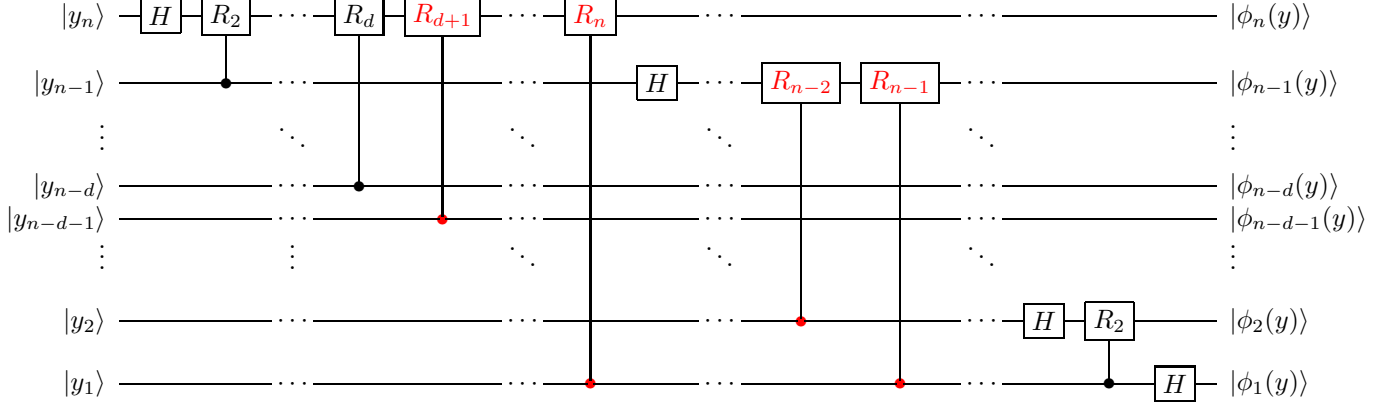


FIG. 1. Generalized quantum circuit for the QFT. Gates removed when performing the AQFT at approximation depth d are drawn in red.

practical computations. This enables approximation that is particularly useful when working with noisy quantum devices, which is a primary motivation of the AQFT.

In the AQFT, we define a depth, d , which serves as an upper limit for the number of conditional rotation gates applied to each qubit in the QFT circuit. We illustrate this in Fig. 1, where gates present in the QFT but removed in the AQFT are marked in red. In comparison to Eq. (3), the AQFT takes the form

$$\mathcal{AQFT} |y\rangle = \frac{1}{2^{n/2}} \bigotimes_{q=1}^n \left(|0\rangle + e^{i2\pi[0.y]_{q,d}} |1\rangle \right)_q, \quad (5)$$

While the QFT requires $n(n+1)/2$ operations and possesses a time complexity of $O(n^2)$, the AQFT at depth d reduces this to $(2n-d)(d-1)/2$. As derived by Barenco et. al. [20], in the presence of decoherence and sufficiently large n , one expects the optimal depth of the AQFT to approximately approach $d \rightarrow \log_2 n$, requiring $O(n \log_2 n)$ operations. While the results in Section IV support this, we do observe instances where, for a given operation, the optimal depth deviates from this expectation.

III. QUANTUM ARITHMETIC OPERATIONS

We now consider two operand qintegers, $|x\rangle$ and $|y\rangle$ of n and m qubits, respectively. We discuss our implementation of Quantum Fourier Addition (QFA) first, as it is required for our implementation of Quantum Fourier Multiplication (QFM).

The first step of QFA is to transform one of the addends into the Fourier basis, with the QFT, as discussed in Sec. II (we will choose y). The addition step can then be performed through a

sequence of conditional rotations that add the phase-encoded magnitude of $|x\rangle$ to $|\phi(y)\rangle$. The process is completed by applying the inverse QFT to $|\phi(X+y)\rangle$, to return that register to the computational basis. Throughout this process, $|x\rangle$ remains in the computational basis.

Whereas, for the QFT, the conditional rotations on a given qubit (i.e. y_i) are controlled entirely by the other qubits defining $|y\rangle$, the rotations performed in the QFA circuit are controlled by the qubits of $|x\rangle$ following

$$\mathcal{QFA} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i2\pi yk/N} |x\rangle |k\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i2\pi xk/N} e^{i2\pi yk/N} |x\rangle |k\rangle. \quad (6)$$

The quantum circuit for the QFA addition step is shown in Fig. 2. Just like with the QFT, the deeper rotations of the addition step approach the identity, and it has been speculated that a similar cutoff on depth in the add step may improve performance in the presence of noise [23]. However, unlike with the AQFT, where this approximation impacts the linearity of the transform domain, a similar cutoff in the addition step more directly impacts the phase shifts being applied. This, in addition to the fact that the same cutoff in the addition step will remove half as many gates overall in the QFA, leads us to expect the benefits of such an approximation will be less apparent than the AQFT (particularly at lower n). Thus, we have opted to consider the AQFT first, and leave the consideration of an approximate addition step for a future work.

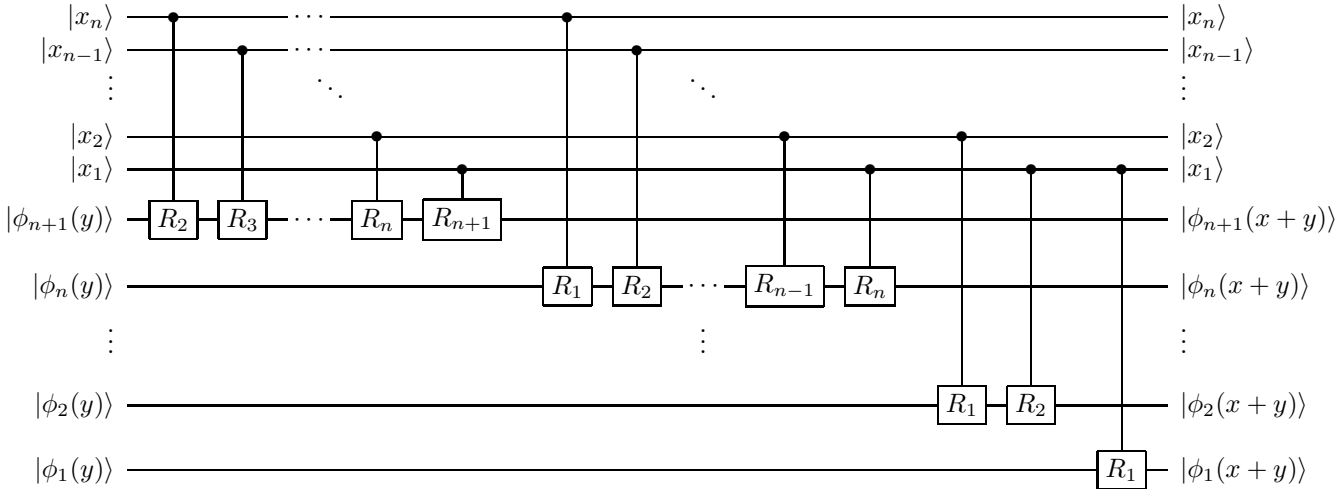


FIG. 2. The generalized addition step of the QFA circuit, being applied in the transform domain.

Once the addition step is complete and $|\phi(y)\rangle \rightarrow |\phi(x+y)\rangle$, one recovers the sum by transforming the output back to the computational basis using the inverse QFT:

$$\begin{aligned} \mathcal{QFT}^{-1} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i2\pi xk/N} e^{i2\pi yk/N} |x\rangle |k\rangle &= \frac{1}{N} \sum_{k,l=0}^{N-1} e^{i2\pi(x+y)k/N} e^{-i2\pi kl/N} |x\rangle |l\rangle \\ &= |x\rangle |(x+y) \bmod N\rangle. \end{aligned} \quad (7)$$

Applying the QFT, addition step, and inverse QFT together, we denote the full QFA circuit

$$\begin{array}{c}
 |x\rangle \xrightarrow{n} \text{---} \bullet \text{---} |x\rangle \\
 |y\rangle \xrightarrow{n+1} \text{---} \boxed{\text{QFA}} \text{---} |x+y\rangle
 \end{array}
 =
 \begin{array}{c}
 |x\rangle \xrightarrow{n} \text{---} \bullet \text{---} |x\rangle \\
 |y\rangle \xrightarrow{n+1} \text{---} \boxed{\text{QFT}} \text{---} \boxed{\text{add}} \text{---} \boxed{\text{QFT}^{-1}} \text{---} |x+y\rangle
 \end{array}$$

To perform non-modular addition, one simply ensures the register being updated ($|y\rangle$ in our case) is large enough to avoid overflow by having $m = n + 1$ or more qubits (for $m > n + 1$, the QFA process only involves the first $n + 1$ qubits of y). Strategies for modular addition of signed and unsigned integers with the QFT have also been developed [31, 34].

A few different approaches for quantum multiplication with the QFT have been proposed [18, 34]. For the purposes of this study, we implement a QFM approach similar to the weighted-sum strategy for unsigned integers presented by Ruiz-Perez [31], with only a few marginal differences. Unlike the QFA described above, in which one of the addends is updated, this approach preserves both multiplicand states, and instead updates a separate product register (which is generally initialized to zero). To avoid any potential overflow, the product register must be at least as large as the combined sizes of the two multiplicand registers. The process involves applying the QFA iteratively on a varying subset of the product register, while being controlled by the multiplicand's qubits.

In order to define the QFM circuit explicitly, we must first introduce the controlled-controlled-phase rotation gate, cR_l , and controlled-Hadamard gate, cH , which we will use to define the controlled QFT (cQFT) and controlled-addition step (cadd) required by the QFM.

$$cR_l = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{\frac{i2\pi}{2^l}} \end{bmatrix}, cH = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}. \quad (8)$$

Replacing the R_l and H gates with cR_l and cH in Figs. 1 & 2, we define the cQFT, cadd, and $c\text{QFT}^{-1}$ circuits just as we did before, but now with one additional control qubit, c , controlling every gate of the process. We denote the controlled QFA (cQFA) as

$$\begin{array}{c}
|c\rangle \text{---} \text{---} |c\rangle \\
|x\rangle \text{---} \text{---} |x\rangle \\
|y\rangle \text{---} \text{---} |x+y\rangle
\end{array}
\begin{array}{c}
\text{---} \text{---} \text{---} \\
\text{---} \text{---} \text{---} \\
\text{---} \text{---} \text{---}
\end{array}
\begin{array}{c}
|c\rangle \text{---} \text{---} |c\rangle \\
|x\rangle \text{---} \text{---} |x\rangle \\
|y\rangle \text{---} \text{---} |x+y\rangle
\end{array}
=
\begin{array}{c}
|c\rangle \text{---} \text{---} |c\rangle \\
|x\rangle \text{---} \text{---} |x\rangle \\
|y\rangle \text{---} \text{---} |x+y\rangle
\end{array}
\begin{array}{c}
\text{---} \text{---} \text{---} \\
\text{---} \text{---} \text{---} \\
\text{---} \text{---} \text{---}
\end{array}
\begin{array}{c}
|c\rangle \text{---} \text{---} |c\rangle \\
|x\rangle \text{---} \text{---} |x\rangle \\
|y\rangle \text{---} \text{---} |x+y\rangle
\end{array}$$

The QFM circuit of this process is provided in Fig. 3, with product register $|z\rangle$ of $n+m$ qubits. Here, the qubits of $|x\rangle$ take the role of the additional control qubit. The process is perhaps easiest to understand when considering the classical case, where the qintegers are each in a single integer state. In that case, the i th cQFA step adds $x_i 2^{i-1} y$ to the product register, where x_i is 0 or 1, and y is a single integer.

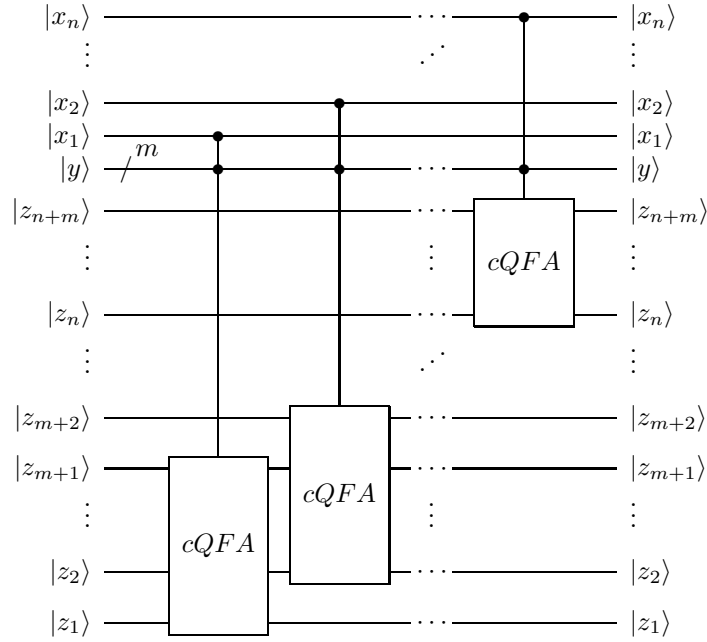


FIG. 3. The generalized QFM circuit, drawn such that $m < n - 1$. If $n = m$, then $|z_n\rangle = |z_{m+1}\rangle$ and the diagram shifts accordingly, and further shifts if $m > n$.

Before we turn our attention to the results of this study, one might note that in quantum addition, so long as $|x\rangle$ is in a single integer state, one could instead store it purely on classical bits, rather than a separate register of qubits as discussed above. In that case, the resulting addition circuit would change dynamically based on the single integer being added, where the controlled rotation gates would be replaced by 1-qubit rotation gates acting on $|y\rangle$. This would result in shorter and less complex circuits that serve to boost all superposed integer states by a single integer value. Naturally, a similar option exists in quantum multiplication for applying

a constant factor across all states. While worth acknowledging as a potential interest, we shall maintain our current focus on cases where both operands are stored on quantum registers and allow superposition.

IV. RESULTS AND DISCUSSION

We present performance results of simulated quantum Fourier addition and multiplication for varying approximation depths, 1q-/2q-gate error rates, and operand's order of superposition. Fig. 4 presents results from addition between two $n = 8$ qintegers, and Fig. 5 presents the results of multiplication between two $n = 4$ qintegers.

We use the same organizational structure in both figures. The left column presents results with varying 1q-gate error rate, while the right column shows results with varying 2q-gate error rate. The top row shows results for operations where both operands are order-1 qintegers (henceforth referred to as a 1:1 operation). The middle row reflects results where one operand is order-1 and the other is order-2 (a 1:2 operation; in the case of addition, the order-2 addend is always stored on the qubit register that is being updated). The bottom row reflects results where both operands are order-2 qintegers (2:2 operation). For all simulated operations with superposed integer states, the probability amplitude is evenly distributed between each state. In addition, for each considered case of superposition and arithmetic operation (i.e. for each row of each figure), the same unique, randomly-generated set of operand states are used for calculating results of both varying 1q-gate error and varying 2q-gate error.

Results in the figures are clustered along the horizontal axis based on their gate error rate. All points within the same cluster correspond to the same error rate, and are only shifted apart horizontally for visibility. The alignment of the center-most point of a cluster on the x-axis reflects the gate error rate used for all calculations in that cluster (for instance, all points in the leftmost cluster correspond to results simulated with a 0.2% 1q-gate error rate, then 0.3%, for the next cluster, and so on). Results are color-coded based on the approximation depth used in those calculations and correspond to the legend in panel (a) of the respective figure. 'Full' designates that the full QFT circuit is performed in those simulations.

We employ a simple success metric for each unique arithmetic operation using quantum state tomography, similar to that of Ref. [28]. Each point in each figure marks the overall success rate in percentage (vertical axis) of 200 (or more) instances of the arithmetic operation on a random, unique choice of qintegers simulated at the given 1q-/2q-gate error rate, operand superposition, and

AQFT approximation depth. For each instance, the simulated AQFA/AQFM circuit was repeated 2048 times (referred to as ‘shots’), and the outputs were tabulated. Instances were considered ‘successful’ when the binary outputs with the highest frequency matched those anticipated based on the input values. In cases where multiple ‘correct’ outputs were expected (due to superposed inputs), instances were deemed unsuccessful if any incorrect output possessed more counts than any one of the correct outputs. All results meeting this criterion were classified ‘successful’, regardless of any significant inequalities in the distribution between the correct outputs (though very few such cases were observed).

In addition to each point, an associated pair of error bars are included to indicate the average ratio of correct to incorrect outputs observed for those results. In each instance, the minimum difference in counts between any single correct and incorrect output was recorded, and the standard deviation, σ , of these differences was calculated using all instances of the associated point. The lower/upper error bars therefore represent the number of correct/incorrect instances that would have been incorrect/correct within one σ , respectively. Narrow error bars reflect large differences between correct and incorrect output counts, while a relatively smaller error bar below/above the point reflects cases where many instances were on the verge of being correct/incorrect.

For each figure, we include either 1q-gate or 2q-gate error rates (denoted P_{1q}^{err} and P_{2q}^{err} , respectively, and varied on the x-axis) in our noise model, and do not include any other gate errors, reset errors, measurement errors, or noise associated with qubit-layout and/or swap-gates (i.e., we consider an idealized layout with complete qubit connectivity). Results with no simulated noise are provided at the x-origin of each plot to help differentiate the impacts associated with the approximation depth, operand superposition, and gate error rates. The transparent, dashed, vertical lines at 0.2% in the 1q-gate error rate plots and at 1.0% in the 2q-gate error rate plots approximate the average reported performance of current IBM superconducting quantum computers [22, 29].

The construction and simulation of all circuits, noise models, and algorithms were implemented using Qiskit 0.31.0 and python 3.7. For all model calculations, circuits are decomposed in terms of the universal gate set of current IBM superconducting quantum computers (i.e., Id, X, RZ, SX, and CX-gates) [22].

In all simulated arithmetic, the operands’ qinteger states were initialized using the reverse decomposition algorithm of Ref. [32] implemented in Qiskit. No noise/errors of any kind were included during the initialization process, regardless of how many gates would have been required for the given states’ initialization on a real machine. The 1q and 2q gate counts of the simulated circuits (excluding gates associated with initialization and measurement) are tabulated in Table I.

Discussion

With the details of the calculations specified, we now discuss our results. Beginning with the 1:1 addition results ((a) and (b) in Fig. 4), we see the outputs are largely insensitive to the examined regime of gate error rates. Besides a small number of cases where the AQFT depth is one, all the QFA operations consistently produced correct results, and the lack of error bars demonstrates none of these successes were within one standard deviation of failure. Even in the case of the $d = 1$ calculations where the approximation is most significant and errors do occur, the number of errors show little to no dependency on the gate error rates, indicating the errors are primarily the result of the approximation rendering the encoding too non-linear for some of the additions. The optimal choice of approximation depth appears to be $d > 1$ for these cases, though any differences between the larger depths aren't visible in this case.

Moving to the 1:2 QFA results and 2:2 QFA results ((c), (d), (e), and (f) in Fig. 4), we immediately see greater sensitivity to the gate error rates in this regime. This follows the general trend that, for the near future, quantum applications that rely on or converge towards heavier output (that is, output corresponding to fewer, more highly-correlated binary strings in the computational basis) will be a driving design factor for early quantum applications, due to the significant noise observed in current hardware. (For instance, since each of the four correct outputs of the 2:2 results ideally possesses $\sim 25\%$ of the probability density, fluctuations from noise have a far greater impact.) We also notice that while dependence on the 2q-error rate disappears at higher values than that of the 1q-gate error rate in these calculations, the impact of the 1q-gate error rate in current machines appears to have already largely been overcome (at least for QFA at $n = 8$), compared to the larger number of errors corresponding to the current hardware's 2q-gate error rates. This appears to support the general expectation that 2q-gate fidelity is one current bottleneck for the performance of applications on modern quantum hardware. However, both here and in the 2:2 QFA results, we see a slower improvement of our result quality with reduced 1q-gate error compared to reduced 2q-gate error rates. If this trend continues at lower error rates and larger n , the 1q-gate errors could overcome 2q-gate errors as a limiting factor for the QFA at high n . Determining whether this trend continues to hold will require further simulation and study.

Unlike the 1:1 QFA results, both the 1:2 and 2:2 QFA results provide much more insight regarding the optimal approximation depth. Even just considering only the reduction in error garnered by the AQFT in the presence of either 1q- or 2q-gate error, we generally see at least similar levels of performance to the full QFT for depths near $d = \log_2 n$, and frequently a slight improvement.

One anticipates significantly more benefit on a real machine, where the AQFT’s reduction in gate count will also mitigate errors associated with reset, measurement, qubit-connectivity, and other sources of noise. As surmised in [20], the quality of results do appear to be somewhat higher near $\log_2 n = 3$. However, we observe significant variation in the optimal approximation depth between clusters in and between all four plots. Depths 2, 3 and 4 are the most common optima, though generally by relatively small margins. Even beyond the noise-free simulations, the AQFT at depth 1 generally produces significantly lower quality, supporting how, even at $n = 8$, not going to too low an approximation depth is critical to maintaining performance when using the AQFT.

Looking now to the QFM results presented in Fig. 5, we observe several notable similarities and differences with those of the QFA. For instance, despite the smaller register sizes occupied by the operands, the much larger circuit sizes required by the QFM results in notably lower success rates than seen in the QFA, even for the 1:1 QFM results. We also observe even stronger examples than seen in the QFA results of the 2q-gate error rate being a major limiting factor for performing quantum arithmetic. At the same time, while less pronounced than it was in the QFA results, we also observe hints of a similar, more rapid improvement in result quality as the 2q-gate error rate decreases compared to decreased 1q-gate error rates.

The QFM results also possess some unique features regarding the impact of the AQFT. On one hand, an approximation depth of 1 appears to consistently have a negative impact on QFM performance at low gate error rates, much like what was seen in the QFA results. On the other hand, at higher gate error rates, the AQFT at the same depth consistently begins to outperform depths 2 and 3, in some cases by margins as large as $\sim 40\%$; a far larger improvement than any observed for the QFA.

Finally, we see a common shape throughout our results, involving a roughly linear trend of the success rate with gate error rate, between asymptotic tails on the left/right side roughly approaching 100%/0% success rates, respectively. This is common to sampling over Gaussian distributions, and reflects that the impact of gate errors on the arithmetic success rate follows a similar pattern. The width of the linear region reflects the width of a ‘plateau’ in the statistical distribution, meaning a smaller width corresponds to a sharper threshold for success. The width of this region in our QFA results is consistently far wider than the QFM results. We also notice in the QFM results (as well as (c) and (e) of the QFA results) that this region grows narrower as we increase the number of superposed operand states. If we assume this trend continues, one expects that the gate fidelity required for highly entangled arithmetic will be a relatively sharp threshold. Like the 2-bit results of quantum addition seen in Ref. [28], with sufficiently high gate

error rates and order of superposition, we consistently observe results with an accuracy around 0%. This reflects that length of the circuit and poor gate fidelity create too much noise to extract the solution without error mitigation and/or a more advanced success metric, such as evaluating the quantum state fidelity [25].

TABLE I. Arithmetic Circuit Gate Counts

d^*	QFA ($n = 8$)					QFM ($n = 4$)		
	1	2	3	4	7**	1	2	3**
1q	163	199	229	253	289	1032	1248	1464
2q	98	122	142	158	182	744	936	1128

*AQFT approximation depth

**This depth is equivalent to the full QFT for the given n .

V. CONCLUSIONS

In this study, we examined (via simulation) the dependency of QFA and QFM on one- and two-qubit gate error rates, approximation depth in the QFT, and operand's order of superposition. We utilized noise models to isolate sources of error and compared the results to perfect, noise-free simulations to differentiate noise inherent to the approximate QFT. In a broader sense, this work reflects an initial effort to better assess the practical challenges that must be overcome before a quantum numerical paradigm can be realistically explored.

The use of the AQFT proved very effective in both our QFA and QFM results, often generating little to no negative impacts, even in error-free simulation. In the presence of noise, results with the AQFT near the optimal approximation almost always produced higher quality results than those of the full QFT. However, even at low qubit count, n , we observed how going to too low an approximation depth has a severe negative impact on our result quality. Further, while a depth of $d = \log_2 n$ served as a reasonable heuristic, we observed significant variation on the optimal approximation depth between the orders of superposition, and even between changes in the gate error rates alone.

Our results support the expectation that 2q-gate error rates currently pose the greater obstacle for quantum arithmetic, though we also observed that our success rates improved much faster as the 2q-gate error rate falls than it does as the 1q-gate error rate falls. Determining whether this

suggests 1q-gate errors will surpass 2q-gate errors as a limiting factor at high n for the discussed arithmetic approaches requires further study.

As anticipated, we observed significant drops in performance as we increased the order of superposition, and that lower error rates significantly reduced this drop in performance. For instance, at the optimal depth, increasing 1:2 addition to 2:2 addition at $n = 8$ resulted in over a 50% drop in accuracy at the current 2q-gate error rate of current IBM hardware ($\sim 1\%$), but only incurred a $\sim 3\%$ drop when simulated with the improved 2q-gate error rate of 0.7%. Consequently, the possibility of a quantum numerical computational paradigm will rely heavily on further improving current hardware.

This work sheds light on the current challenges and hardware limitations that must be resolved before a quantum numerical computational paradigm becomes a realistic possibility. Our results provide insight on what quantum arithmetic is currently feasible, and we have observed greater complexity in determining the optimal AQFT depth than initially anticipated. Many questions remain largely unanswered, and will require further study and simulation. One natural step forward is to continue the investigation of isolating the inter-dependency of different sources of error and order of superposition, as well as to what degree the AQFT mitigates them. Greater variation on how superposed state are entangled may also be informative. Simulations that isolate other sources of error, such as thermal relaxation, and qubit measurement errors, their simultaneous simulation with 1q-/2q- gate errors, as well as the impact of error mitigation and extrapolation from integers to real numbers, would offer a much more complete understanding of how these elements impact the QFT and AQFT. In addition, extending the study to larger n would reveal whether the trends observed in this study continue or diverge, and may reveal other features that were suppressed at the smaller register sizes considered in this study. Further, the current study only considered methods for unsigned instances of QFA and QFM operations. Employing other methods, such as signed QFM, may reveal critical insight into current and new quantum algorithms, such as those for weighted-sum problems. Moreover, they will lay necessary foundation as future improvements in technology and techniques continue to make a quantum numerical computational paradigm more feasible.

ACKNOWLEDGMENT

We acknowledge fruitful discussion with Soham Pal and Weijie Du. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not

reflect the official policy or views of IBM or the IBM Quantum team.

- [1] J J Álvarez-Sánchez, J V Álvarez-Bravo, and L M Nieto. A quantum architecture for multiplying signed integers. 128:012013, aug 2008.
- [2] Hafiz Md Hasan Babu. Cost-efficient design of a quantum multiplier-accumulator unit. *Quantum Information Processing*, 16(1):1–38, 2017.
- [3] Adriano Barenco, Artur Ekert, Kalle-Antti Suominen, and Päivi Törmä. Approximate quantum fourier transform and decoherence. *Physical Review A*, 54(1):139–146, Jul 1996.
- [4] Amlan Chakrabarti and Susmita Sur-Kolay. Designing quantum adder circuits and evaluating their error performance. In *2008 International Conference on Electronic Design*, pages 1–6, 2008.
- [5] Andrew Cross. The ibm q experience and qiskit open-source quantum computing software. In *APS March Meeting Abstracts*, volume 2018, pages L58–003, 2018.
- [6] Thomas G Draper. Addition on a quantum computer. *arXiv preprint quant-ph/0008033*, 2000.
- [7] Maximilian Fillinger. Data structures in classical and quantum computing, 2013.
- [8] Richard Jozsa. Fidelity for mixed quantum states. *Journal of Modern Optics*, 41:2315–2323, 1994.
- [9] Dawid Kopczyk. Quantum machine learning for data scientists, 2018.
- [10] Frank Leymann and Johanna Barzen. The bitter truth about gate-based quantum algorithms in the NISQ era. *Quantum Science and Technology*, 5(4):044007, sep 2020.
- [11] Wiphoo Methachawalit and Prabhas Chongstitvatana. Adder circuit on ibm universal quantum computers. In *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 92–95, 2020.
- [12] Prakash Murali, Norbert Matthias Linke, Margaret Martonosi, Ali Javadi Abhari, Nhung Hong Nguyen, and Cinthia Huerta Alderete. Full-stack, real-system quantum computer studies: Architectural comparisons and design insights. In *2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA)*, pages 527–540. IEEE, 2019.
- [13] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018.
- [14] Lidia Ruiz-Perez and Juan Carlos Garcia-Escartin. Quantum arithmetic with the quantum fourier transform. *Quantum Information Processing*, 16(6):152, 2017.
- [15] V.V. Shende, S.S. Bullock, and I.L. Markov. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010, Jun 2006.
- [16] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, Oct 1997.
- [17] Engin Şahin. Quantum arithmetic operations based on quantum fourier transform on signed integers. *International Journal of Quantum Information*, 18(06):2050035, Sep 2020.

- [18] J J Álvarez-Sánchez, J V Álvarez-Bravo, and L M Nieto. A quantum architecture for multiplying signed integers. 128:012013, aug 2008.
- [19] Hafiz Md Hasan Babu. Cost-efficient design of a quantum multiplier–accumulator unit. *Quantum Information Processing*, 16(1):1–38, 2017.
- [20] Adriano Barenco, Artur Ekert, Kalle-Antti Suominen, and Päivi Törmä. Approximate quantum fourier transform and decoherence. *Physical Review A*, 54(1):139–146, Jul 1996.
- [21] Amlan Chakrabarti and Susmita Sur-Kolay. Designing quantum adder circuits and evaluating their error performance. In *2008 International Conference on Electronic Design*, pages 1–6, 2008.
- [22] Andrew Cross. The ibm q experience and qiskit open-source quantum computing software. In *APS March Meeting Abstracts*, volume 2018, pages L58–003, 2018.
- [23] Thomas G Draper. Addition on a quantum computer. *arXiv preprint quant-ph/0008033*, 2000.
- [24] Maximilian Fillinger. Data structures in classical and quantum computing, 2013.
- [25] Richard Jozsa. Fidelity for mixed quantum states. *Journal of Modern Optics*, 41:2315–2323, 1994.
- [26] Dawid Kopczyk. Quantum machine learning for data scientists, 2018.
- [27] Frank Leymann and Johanna Barzen. The bitter truth about gate-based quantum algorithms in the NISQ era. *Quantum Science and Technology*, 5(4):044007, sep 2020.
- [28] Wiphoo Methachawalit and Prabhas Chongstitvatana. Adder circuit on ibm universal quantum computers. In *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 92–95, 2020.
- [29] Prakash Murali, Norbert Matthias Linke, Margaret Martonosi, Ali Javadi Abhari, Nhung Hong Nguyen, and Cinthia Huerta Alderete. Full-stack, real-system quantum computer studies: Architectural comparisons and design insights. In *2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA)*, pages 527–540. IEEE, 2019.
- [30] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018.
- [31] Lidia Ruiz-Perez and Juan Carlos Garcia-Escartin. Quantum arithmetic with the quantum fourier transform. *Quantum Information Processing*, 16(6):152, 2017.
- [32] V.V. Shende, S.S. Bullock, and I.L. Markov. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010, Jun 2006.
- [33] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, Oct 1997.
- [34] Engin Şahin. Quantum arithmetic operations based on quantum fourier transform on signed integers. *International Journal of Quantum Information*, 18(06):2050035, Sep 2020.

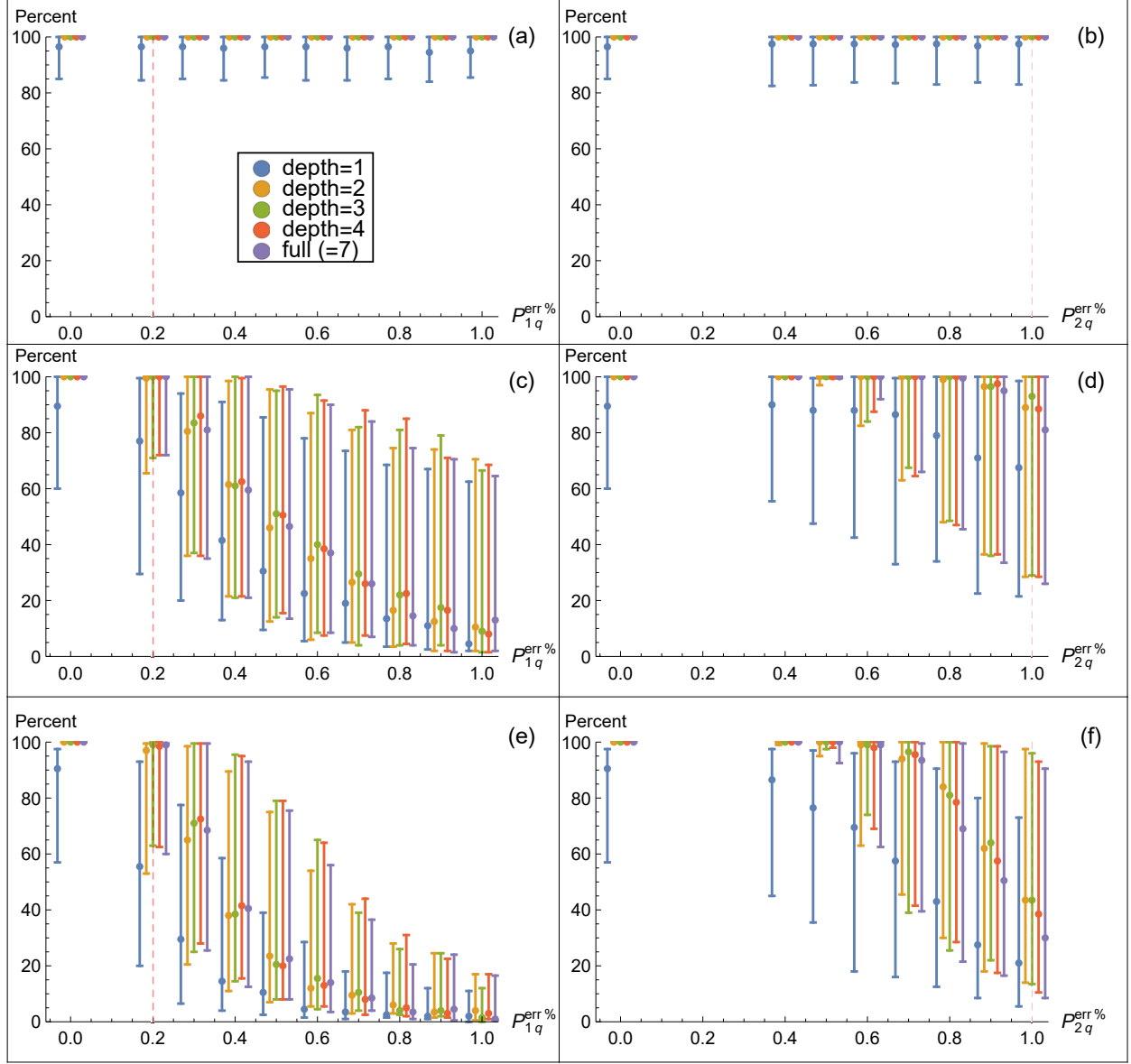


FIG. 4. Success rates (vertical axes) of quantum addition of two 8-qubit qintegers performed in the Quantum Fourier Basis with varying gate error rates, number of superposed addend state, and AQFT approximation depth; see Sec. IV for details. The left/right column presents results with varying 1q-gate/2q-gate error rate, ($P_{1q}^{\text{err}\%}/P_{2q}^{\text{err}\%}$) on the horizontal axes, respectively. The top/middle/bottom row shows results for (1:1)/(1:2)/(2:2) superposed addend states. Color-coded results for each depth of AQFM approximation depth are clustered along the horizontal axes and are only shifted apart horizontally for visibility; the alignment of the center-most point of a cluster reflects the gate error rate used for all calculations in that cluster. The ‘full’ approximation depth designates that the full QFT circuit is performed in those simulations.

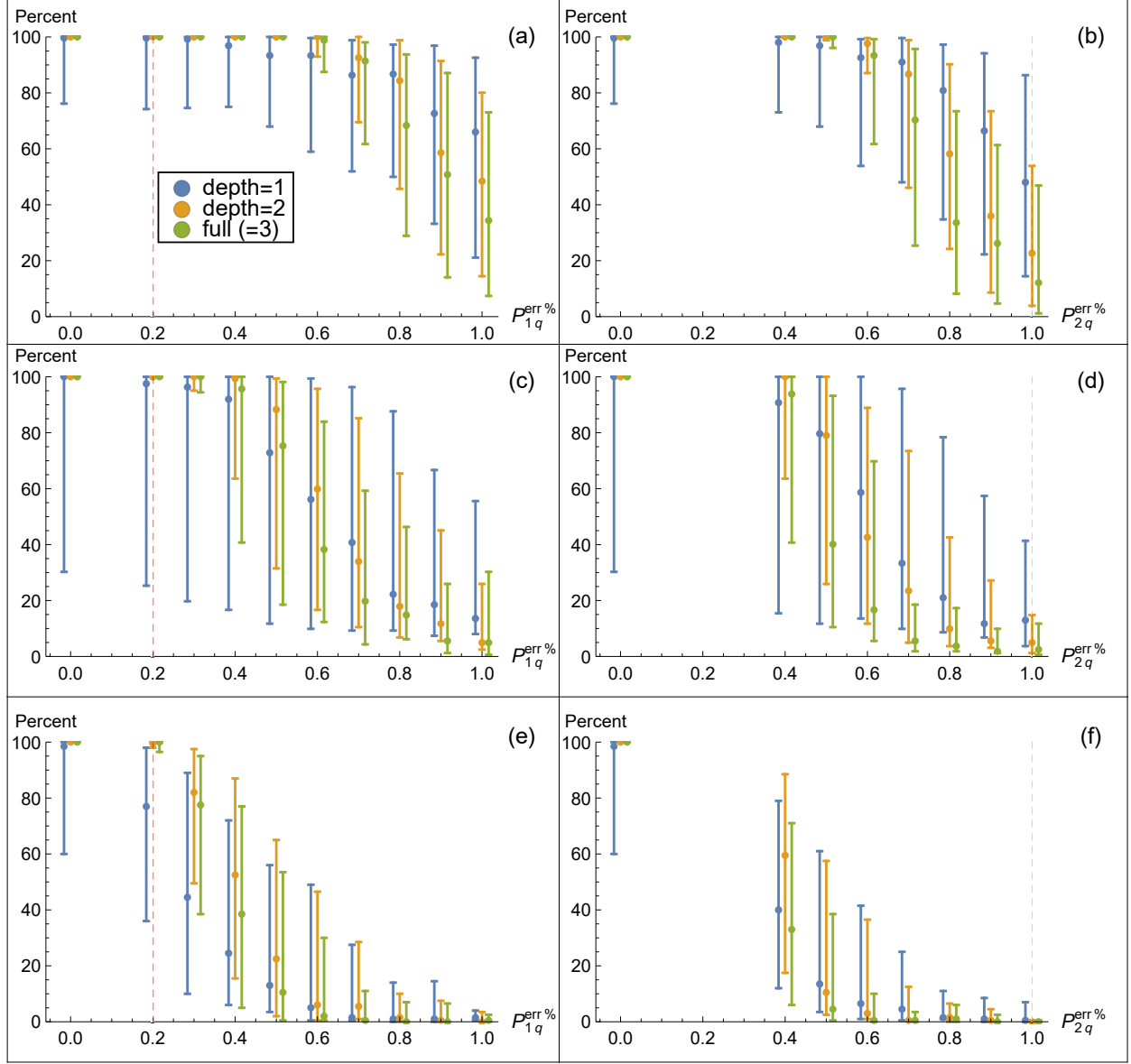


FIG. 5. Success rates (vertical axes) of quantum multiplication of two 4-qubit qintegers performed in the Quantum Fourier Basis with varying gate error rates, number of superposed multiplicand state, and AQFT approximation depth; see Sec. IV for full details. The left/right column presents results with varying 1q-gate/2q-gate error rate ($P_{1q}^{err\%}/P_{2q}^{err\%}$) on the horizontal axes, respectively. The top/middle/bottom row shows results for (1:1)/(1:2)/(2:2) superposed multiplicand states. Color-coded results for each depth of AQFM approximation depth are clustered along the horizontal axes and are only shifted apart horizontally for visibility; the alignment of the center-most point of a cluster reflects the gate error rate used for all calculations in that cluster. The ‘full’ approximation depth designates that the full QFT circuit is performed in those simulations.