A Novel Quantum Circuit for Integer Factorization: Evaluation via Simulation and Real Quantum Hardware

Jesse Van Griensven Thé *, Victor Oliveira Santos, Bahram Gharabaghi

Posted Date: 21 October 2025

doi: 10.20944/preprints202510.1649.v1

Keywords: quantum number theoretic transform; quantum fourier transform; integer factorization; Shor's algorithm; NISQ devices

 doi:10.20944/preprints202510.1649.v1

Article

A Novel Quantum Circuit for Integer Factorization: Evaluation via Simulation and Real Quantum Hardware

Jesse Van Griensven Thé [1,2,*], Victor Oliveira Santos [1] and Bahram Gharabaghi [1]

[1] School of Engineering, University of Guelph, 50 Stone Rd E, Guelph, ON N1G 2W1, Canada

[2] Lakes Environmental, 170 Columbia St. W, Waterloo, ON N2L 3L3, Canada

* Correspondence: jesse.the@weblakes.com

Abstract

This work tests the hypothesis that a Quantum Number Theoretic Transform (QNTT) circuit, here named Jesse-Victor-Gharabaghi (JVG) algorithm, can perform better than the Shor's algorithm, in terms of number of required gates and qubits. This methodology replaces the Quantum Fourier Transform (QFT) with a Quantum Number Theoretic Transform (QNTT) circuit to predict periodicity in the number theory and factor integer numbers, which serve as keys in cryptographic methods, like RSA and ECC. Several composite numbers were evaluated through both simulation and real quantum hardware to verify feasibility and performance. Performance was assessed across runtime, memory consumption, and gate counts. Simulation results showed that the JVG can reduce the growth in CX gates by 30.3%, circuit depth by 33.5%, memory by 9.6%, and runtime by 14.7% relative to the Shor's algorithm. On quantum hardware, JVG reduces growth in runtime by 26% and X-gate counts by 44.4%, achieving consistently lower coefficients of variation across metrics. Projection curves derived from the fitted trends predict the eventual JVG advantage, over Shor ,in runtime, gates, and depth as the number of qubits increases, including RSA-scale configurations. These results support JVG as a more hardware-compatible and robust noise-tolerant substitute for the Shor's framework, offering a viable path toward practical quantum integer factorization on near-term Noisy Intermediate-Scale Quantum (NISQ) devices.

Keywords: quantum number theoretic transform; quantum fourier transform; integer factorization; Shor's algorithm; NISQ devices

1. Introduction

Quantum computing systems are anticipated to surpass their classical counterpart by implementing quantum mechanic principles like superposition, entanglement, and interference [1,2]. Driven by this potential, Quantum Computing has been drawing attention from researchers and investors alike. The funding towards the development of quantum technologies, from both public and private investors, has increased 54% between 2023 and 2024, amassing US$ 2.0 billions worldwide, and is expected to reach around US$ 16.4 billions by 2027 [3,4]. This trend is also reflected in the rapid growth of quantum-focused companies such as D-Wave, IonQ, and Quantinuum, which have increased their market value by more than 2,530%, 800%, and 812%, respectively, over the past 12 months, reaching a combined market capitalization of approximately US$50 billion [5–7]. The massive investments in this sector allowed for significant advancements in multidisciplinary fields, such as finances [8,9], material science [10,11], chemistry [12], pharmacology [13], and machine learning [14,15].

Another major area impacted by quantum technology is cryptography [16,17]. Since Shor proposed a new algorithm using quantum information processing for efficient number factoring back in 1994 [18], his work proved the feasibility of a quantum-based algorithm for number factoring. By

 doi:10.20944/preprints202510.1649.v1

using modular exponentiation and Quantum Fourier Transform (QFT), previously conceived by Coopersmith [19], Shor's algorithm successfully captured the period of a function $f(x) = a \, mod \, N$ given an initial superposition state. By leveraging these components, this algorithm performs on polynomial time $O((\log N)^3)$, in opposition to the exponential time required by the classical procedures for RSA integer factorization [20,21].

Building upon this knowledge, several subsequent studies sought to improve Shor's algorithm. In [22], the authors focused on speeding up the arithmetic operations by using improved adder designs, allowing to the parallel execution of quantum gates, while also optimizing the overall circuit's structure. By doing so, they improved the modular exponentiation execution by up to seven hundred times compared to the available approaches. Building upon the knowledge developed by Meter and Itoh [22], the authors in [23] proposed a new reversible circuit for modular exponentiation using linear-size circuits, and working on register-transfer level instead of the commonly used qubit-transfer level. Their methodology's overall performance showed better scalability than the others previously available at the time, requiring four times less qubits.

The work by Ekerå [24] proposed changes on the QFT algorithm and the post-processing part alike. The author introduced a Shor's discrete logarithm variant that is optimized when the discrete logarithm d is significantly smaller compared to the group order q. The algorithm uses smaller QFTs whose sizes reduce the total qubit requirements in this setting. Furthermore, instead of the classical continued fractions method used in Shor's post-processing, the author employed lattice-based techniques to recover the discrete logarithm from the quantum measurement outcomes. This work has been further expanded in the following work by Ekerå and Håstad [25], where the authors showed that the RSA factorization can be formulated as a short discrete logarithm, thus reducing the burden on quantum computers.

Chevignard et al. [26] optimized the number of required logical qubits for Shor's algorithm by providing an alternative algorithm for the modular exponentiation part. By combining Ekerå–Håstad's algorithm, compression techniques and residual arithmetic the authors could reduce the number of logical qubits required for RSA integer factoring. However, this simplification introduces a trade-off between the total number of qubits and the number of gates necessary for its implementation. In [27], the author proposed a novel methodology for number factorization using smaller circuit size by focusing on lattice-based sampling method. This change introduces a trade-off between circuit size and post-processing efficiency, since achieving a simpler classical post-processing demands more qubits and gates, reducing the practicality of Shor's algorithm on the current quantum hardware.

Gidney [28] improves over his previous work on number factorization [29], developing circuit optimizations. By using approximated residue arithmetic, improved data allocation and reducing the number of magic states, the author managed to elaborate a methodology able to factor a large 2048-bit integer using less than 1 million noisy qubits. The author also showed that the total of quantum gates could be decreased compared with the work by Chevignard et al. His work set an important milestone to the feasibility of quantum number factorization in the current quantum machines.

As the literature mentioned, there are many efforts in different areas of Shor's algorithm components, ranging from modular exponentiation to post-processing. However, a common feature among these works is the reliance on the QFT circuit, or its variants, for period extraction. Therefore, it is still necessary to investigate alternatives for the QFT structure itself. For example, alternative transforms may be more practical on quantum hardware for specific applications. One strong contender to this task is the Number Theoretic Transform (NTT), which is a specialized variation of the DFT but works over finite fields through modular arithmetic rather than complex numbers [30,31]. Classically, the NTT runs in $O(nlogn)$ time and avoids floating-point precision issues, making it well-suited to cryptographic computations [32]. A quantum version of the NTT could, in principle, serve as a substitute for the QFT in algorithms involving integer or polynomial structures. Such an implementation might offer advantages in precision and potentially simpler gate constructions, since modular addition can be easier to realize than arbitrary quantum rotations.

 doi:10.20944/preprints202510.1649.v1

In this context, incorporating a quantum version of the NTT within a quantum framework could lead to a modular design of QFT circuits. By breaking down the QFT into smaller components and selectively substituting them with specialized transforms, the overall circuit can be simplified and made more efficient for NISQ hardware. Such modularization not only streamlines the implementation of fundamental quantum algorithms but also improves their adaptability and performance [33]. Modular QFT architectures support more efficient Quantum Phase Estimation (QPE), a key subroutine in many quantum applications such as quantum chemistry, Hamiltonian learning, and Variational Quantum Eigensolver (VQE) [34]. Decomposing QFT into optimally connected building blocks helps to overcome limitations in qubit connectivity, mitigate gate errors, and reduce decoherence, thereby enabling algorithms to tackle larger instances and deeper circuits [35]. Additionally, this strategy allows for dynamic error mitigation and adaptive allocation of quantum resources, enhancing the reliability and scalability of computations on the available quantum devices.

To bridge this knowledge gap, we introduce a novel quantum factorization algorithm, named Jesse-Victor-Gharabaghi (JVG) algorithm, by incorporating a Quantum Number Theoretic Transform as an alternative to the usual QFT circuit, in the original Shor's pipeline. It is important to emphasize that JVG's novelty lies in advancing number theory period finding by extracting it from a finite ring rather than a complex field. The present work constitutes the first empirical validation of a QNTT-based approach through comprehensive benchmarking on both simulated and real quantum backends. This includes resource scaling projections under realistic NISQ constraints. This methodology offers a distinct and measurable contribution beyond previous QNTT formulations, which had not been integrated into or tested within a complete quantum factoring framework.

At the current phase of this study, the comparison will be restricted between the proposed QNTT-based circuit and the standard Shor's algorithm using QFT. This is so we have a clear baseline for evaluating performance and scalability. By establishing a direct comparison, it becomes possible to quantify the resource savings and noise resilience provided by the QNTT circuit while maintaining the same algorithmic framework. This controlled assessment ensures that observed improvements can be attributed specifically to the substitution of QFT with QNTT, rather than to unrelated modifications in the algorithm. By doing so, we aim to provide a novel paradigm for number factorization, achieving superior scalability considering gate count and memory usage.

2. Methodology

2.1. Quantum Computing

Feynman first envisioned Quantum Computing in his groundbreaking paper "Simulating Nature with Computers" [36], where he proposed that classical computing machines could not capture the random nature of physical and natural phenomena. In fact, it would be more correct to apply quantum mechanic concepts to emulate the behaviour of physical world. Therefore, researchers have developed a completely new framework for information processing, based on the core principles of quantum mechanics, as suggested by Feynman. The most relevant principles used in the present study are presented and discussed as follows.

2.1.1. Qubits

Classical computers represent information using bits in binary form, where each one can take a binary value as input. The information is then computed through logical operations [37,38]. Its quantum counterpart is the quantum bit, usually called qubit. It is the basic unit of information processing in Quantum Computing (QC) and is defined mathematically as a vector containing the complex amplitudes defining the system's quantum state. For example: a quantum state defined at ground state, is $|0\rangle = [1\ 0]^T$. The excited state is represented as $|1\rangle = [0\ 1]^T$. With qubits, we then can represent a complex quantum state in the form [39]:

 doi:10.20944/preprints202510.1649.v1

$$|\Psi\rangle = \sum_{j=0}^{n-1} \alpha_j |j\rangle_n \quad (1)$$

Equation 1 showcase a generic n-qubit quantum system $|\Psi\rangle$, derives from a linear combination of all $|j\rangle_n$ states. Here, α_j represents the amplitude of the respective j-state, being a complex value. Physically, absolute value of the amplitude represents the likelihood of a specific j-state to occur. Finally, the amplitudes are normalized values such as:

$$\sum_{j=0}^{n-1} |a_j|^2 = 1 \quad (2)$$

Qubits encode the characteristics of a quantum system, and are defined within a complex Hilbert space [40,41]. Being this an abstract idea, it is not trivial to visualize the behavior of the qubit while subjected to operations. However, it is possible understanding the quantum system behavior, and consequently the qubit itself, by using the Bloch sphere for a three-dimensional representation. The quantum system $|\Psi\rangle$ then depends on the position of the qubit within this Hilbert space, that polar coordinates can represent. Given that the amplitudes are normalized to a unitary value as shown in Equation 2, it is possible to rewrite $|\Psi\rangle$, for a one qubit system, state as:

$$|\Psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \quad (3)$$

Where θ and ϕ are real values representing the angles, in radians, of the qubit described within the Bloch sphere. Figure 1 depicts the Bloch sphere.

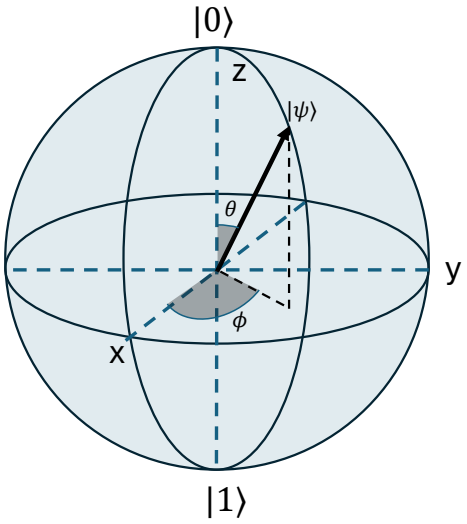


Figure 1. Bloch sphere representation of a two-qubit quantum state|Ψ⟩.

With these definitions, the qubits provide the framework for modeling the relevant quantum mechanical properties used in this study.

2.1.2. Superposition

The superposition principle in quantum mechanics states that a quantum system may exist as a linear combination of its possible states, as shown in Equation 1. This opposes the classical understanding of the classical bits, which are restricted to a single definite state at any given moment, while qubits can exist in a superposition of both cases [42,43]. In the QC framework, the superposition allows for processing an exponentially large state space, being one of the cornerstones which allows quantum algorithms to achieve significant speedups when compared to their classical counterparts [1,44,45].

 doi:10.20944/preprints202510.1649.v1

2.1.3. Entanglement

Another important concept used in this study is entanglement [46]. It describes a single quantum system formed by two or more particles that cannot be expressed as their individual properties. In this scenario, these particles could share so much information between themselves that it would be possible to describe the full system by measuring only one of its particles [44,45]. In essence, entanglement means that for composite quantum systems, the state of each particle is inseparable from that of the others even when they are physically apart [15]. This conclusion is counterintuitive and not fully understood by the scientific community. However, it plays a significant role in the QC framework. It appears in quantum teleportation applications, quantum error correction, and number factorization, even appearing in Shor's algorithm, being also responsible for enabling significant processing speed-ups by the quantum machines [47].

2.1.4. Interference

The third major quantum mechanical concept used in our study is interference. Like previously said, the quantum systems have wave-like behavior which can be modeled by Schrödinger's equation. Therefore, the particles in such a system may interfere with each other in either a constructive or a destructive fashion [48,49]. This will affect the phase of each particle, which in part influences the likelihood of the outcome of a quantum state after measurement. The application of this property allows the QC to manipulate the probability of an outcome to a desired state. Examples of interference in QC framework are the Grover's search algorithm [50], and QFT algorithm itself, both of which rely on interference to amplify the probability of the desired solution [44].

2.2. Quantum Gates

Quantum gates serve as helpers to model the quantum mechanical properties. They are unitary matrices that allow the application of linear operations on qubit state vectors. A matrix is defined as unitary when:

$$U^{-1} = U^{\dagger} \tag{4}$$

Where $U^{\dagger}$ represents the complex conjugate of a matrix U. By extension, unitary operators also have that $UU^{\dagger} = U^{\dagger}U = I$, being I the identity matrix with the same order as U. An important property of unitary matrices is that they preserve the norm and the inner product of the vector which the operate upon, allowing for reversible operations without loss on information, which is one of the backbones of the QC framework [44].

Different fundamental quantum gates and operations were employed in this study, and it is important to introduce them before proceeding. The first unitary gate presented is the Hadamard gate, defined as follows:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{5}$$

This gate puts the qubit into a superposition state between $|0\rangle$ and $|1\rangle$. This is achieved by performing rotations around X and Z-axis as defined in the Bloch sphere (Figure 1). This matrix is fundamental in many quantum circuits. It ensures that the superposition is present, and, together with entanglement and interference, ensuring the that the quantum system explores multiple paths at once.

Pauli gates are also fundamental pieces of the quantum circuits. Specifically for this study, we present Pauli X and Pauli Z as they were used in our methodology.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{6}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{7}$$

The Pauli X gate changes the qubit orientation around X-axis, moving it from $|0\rangle$ to $|1\rangle$ and vice-versa. The Pauli Z gate flips the phase of a given qubit. Considering a system in superposition, this operation does not alter the qubit state in $|0\rangle$, but maps the original $|1\rangle$ to $-|1\rangle$, fundamentally

 doi:10.20944/preprints202510.1649.v1

changing the qubit's phase by a π radians factor. The Pauli Z matrix can be generalized in terms of ϕ angle, given in radians (Figure 1):

$$R_\phi^Z = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix} \quad (8)$$

In this general formulation, the term $e^{i\phi}$ determines the phase change of the qubit.

Controlled operations are also paramount in QC. They allow that entanglement can be modeled into the QC framework, enhancing data processing done by quantum devices. Controlled quantum gates require at least two qubits, where one is the control and the other is the target. The operation is performed over the target qubit only if the control has the $|1\rangle$ state. It is possible to have multi-controlled gates. , in which the unitary operation acts on the target qubit exclusively when all control qubits are in the $|1\rangle$ state. The present work used CNOT (or CX) and Toffoli gates as controlled operations. The former is a two-qubit quantum gate implementing a controlled Pauli X matrix. The latter also implements a Pauli X operation, but it has two controlling qubit. The CNOT matrix is:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (9)$$

The Toffoli matrix is an extension of the CNOT one. The matrix considering a two-qubit control scenario and one target is presented below:

$$Toffoli = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (10)$$

Finally, another important operation present in the developed circuits is the SWAP. As its name indicates, this operation changes the state between two qubits. Its matrix is:

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

2.3. Quantum Fourier Transform

The Quantum Fourier Transform was conceived by Coppersmith [19]. It is the quantum counterpart of the classical Discrete Fourier Transform (DFT), mapping data from the time domain into the frequency one. Classically, DFT has complexity $O(n^2)$ with the following formulation [30,51]:

$$\widehat{f_k} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{\frac{2\pi i j k}{N}} \quad (12)$$

Equation 12 shows the output vector $\widehat{f_k}$ as being a normalized linear combination of elements of the input vector x_j. The term $e^{\frac{2\pi i j k}{N}}$ is the kernel of the transform function, given in terms of indices j and k, and by the vector length N itself. Additionally, according to Euler's equation, this term can be understood as a combination of the oscillatory influence of sine and cosine at different frequencies, as given by $2\pi i$. Additionally, by using advanced algorithm building techniques, it is possible to implement a faster variant of DFT with complexity $O(n \log n)$, being this version often referred to as Fast Fourier Transform (FFT) [52]. The QFT is mathematically defined as [1,44]:

 doi:10.20944/preprints202510.1649.v1

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle \tag{13}$$

The expression in Equation 13 is similar to the one presented in 4 but is fundamentally different. Here, the resulting quantum state $|j\rangle$ is mapped into a superposition of states $|k\rangle$. Additionally, being this a quantum transformation, it lies within a Hilbert space with dimension N. As a unitary operation over an N-dimensional Hilbert space, QFT performs a linear transformation that preserves inner products and vector norms, ensuring reversibility within the dynamics of a quantum computer [44]. The quantum circuit for a 4-qubit configuration for QFT is presented in Figure 2.

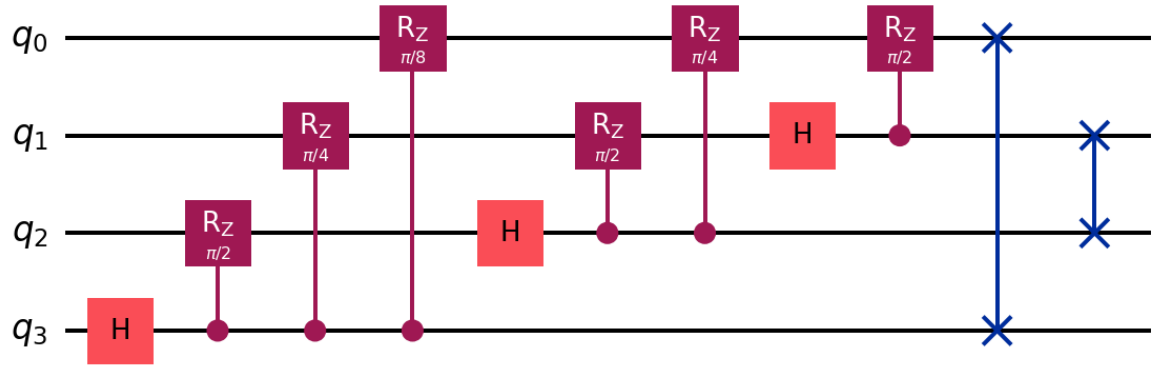


Figure 2. Quantum circuit schematics for QFT.

Figure 2 elucidates that the QFT circuit is a sequence of rotations applied over a superposition state. From left to right, the first quantum Hadamard gate puts the qubit into superposition. It is then followed by controlled gates R_ϕ^Z, which moves the qubit around the Z-axis given a ϕ angle. From Equation 8, it is possible to see that the rotations implemented in the QFT circuit in Figure 2 can be generalized as $R_k^Z = R_{\frac{2\pi}{2^k}}^Z$. The last two operations are SWAPs, reversing the qubits orders to match the QFT's definition [39,44].

The QFT, alike its classical counterpart, also has an inverse configuration which is used in Shor's algorithm. Considering the quantum circuit configuration, the IQFT is the same as applying the forward circuit (Figure 2) but reversed. The IQFT circuit is:

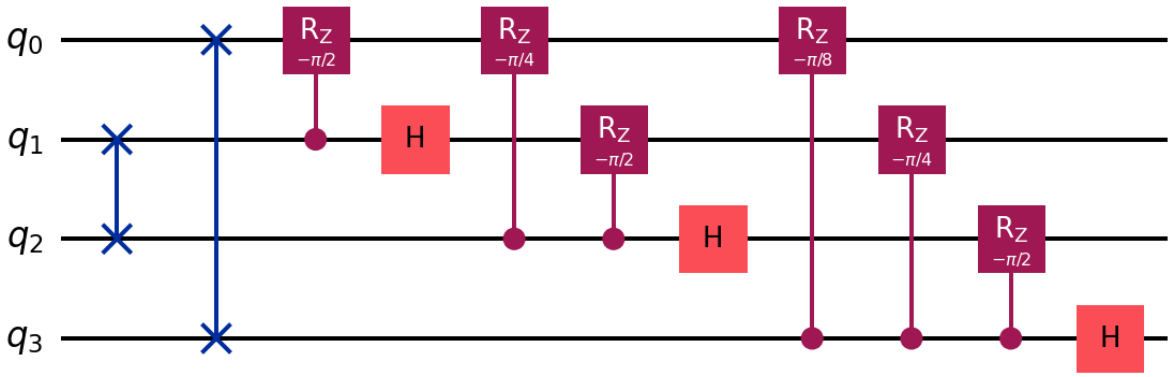


Figure 3. Quantum circuit schematics for the inverse QFT.

2.4. Number Theoretic Transform

The Number Theoretic Transform (NTT) is a specialized version of DFT. While the latter operates over complex number, the former performs analogous computations over a real finite field or ring, often the integer modulus a prime [53]. Its formulation is [31]:

 doi:10.20944/preprints202510.1649.v1

$$\widehat{f_k} = \sum_{j=0}^{N-1} x_j \omega^{jk} \ mod\ p \tag{14}$$

Where mod indicates modular operations over a ring defined by a prime number p, as in $\mathbb{Z}_p = \{0, 1, \ldots p-1\}$. Modular arithmetic operations then map the results into the defined ring by taking the remainder upon division by q. This makes the values wrap around q whenever they exceed this prime number [54]. Still in Equation 8, the term ω represents the primitive root. In arithmetic, primitive roots generate the set of $N = p - 1$ integer coprime values to the prime q through successive exponentiations. Additionally, it is needed that $N|(p-1)$ and the primitive root must ensure that $\omega^j \equiv 1\ mod\ p$ while $\omega^k \not\equiv 1\ mod\ p,\ \forall\ 0 < k < j$ [54].

Given its similar formulation to the DFT, NTT can be implemented in a classical fashion to reach complexity $O\ (n \log n)$ [55]. This is achieved by implementing the Gentleman-Sande algorithm [56]. Originally conceived as a variation of the Cooley-Tukey FFT algorithm [57], the Gentleman-Sande algorithm works in bit-reverse order, performing a decimation-in-frequency (DIF) factorization of the NTT. In this approach, the input sequence is processed orderly, but the outputs are produced in digit-reversed order. Each stage of the algorithm recursively breaks the problem into smaller parts, combining results with butterfly operations and twiddle factors, i.e. the unitary roots. The bit-reversal step must then be unscrambled at the end to recover the coefficients in the correct order. This variant, alongside Cooley-Tukey's decimation-in-time (DIT) approach, forms one of the two standard recursive FFT factorizations, and is frequently adapted in NTT applications due to its memory efficiency and modular structure [52,56].

2.5. Quantum Number Theoretic Transform

Building upon the classical NTT, and the Gentleman-Sande algorithm, the Quantum Number Theoretic Transform (QNTT) was developed to be used in a quantum framework. The original QNTT implementation used in this study, was proposed by Lu et al. [58]. Their methodology combined quantum arithmetic operations together with quantum modular operations to implement a quantum version of the Gentleman-Sande butterfly operation for a configuration using 4 qubits as input and a modulus equals to 5. The authors implemented their QNTT circuit using NISQ friendly operations. For example, the adder and subtractor gates are composed by sequences of CNOT gates, which is more efficiently implemented than, for example, the QFT-based adder drapper [59–62]. By doing so, Lu et al. [58] developed an efficient quantum circuit able to perform QNTT operations in a quantum framework. Figure 4 depicts the QNTT circuit.

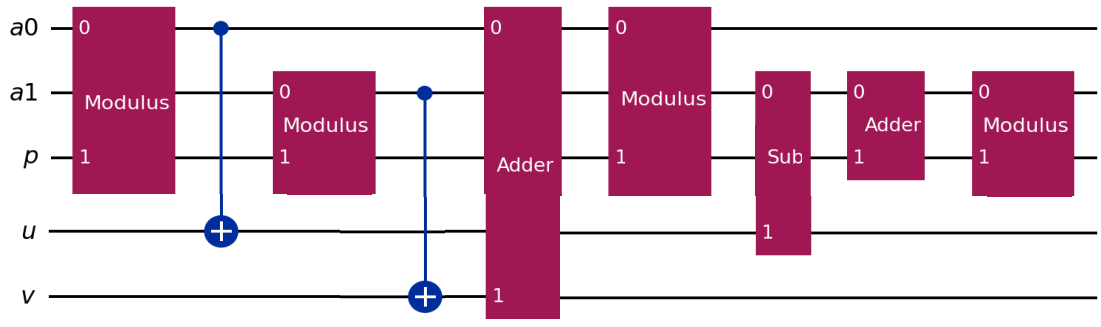


Figure 4. A generic quantum circuit of the QNTT operation using quantum Gentleman-Sande [58].

In Figure 4, the rectangular boxes represent the operations applied to the qubits. In this image, there is two input values represented by registers $|a_0\rangle$ and $\omega \times |a_1\rangle$, each one containing n-qubits, and ω being the twiddle factor. The register $|p\rangle$ is the integer prime number determining the ring $\mathbb{Z}_p$, containing n-qubits. The remaining registers, $|u\rangle$ and $|v\rangle$ are ancilla qubits that will help to compute the necessary operations, containing n-qubits each [58].

 doi:10.20944/preprints202510.1649.v1

It is possible to observe that the QNTT circuit uses quantum modulus, addition and subtraction operations to compute the transforms. It starts with a modulus operation computing $|a_0\rangle\ mod\ |p\rangle$, where the result is stored in a register $|u\rangle$ using a CNOT gate. Following, the modular operation for the second input value is computed, and its result is stored in $|v\rangle$. The next adder and modular operations then compute $|a_0 + a_1\rangle\ mod\ |p\rangle$, which returns the transformed value for the register $|a_0\rangle$. The last three operations compute $|a_1 - a_0 + p\rangle\ mod\ |p\rangle$, finally it returns the transformed value for the second input register.

Similarly to QFT, the QNTT circuit has an inverse form to compute the inverse QNTT operation, illustrated in Figure 5:

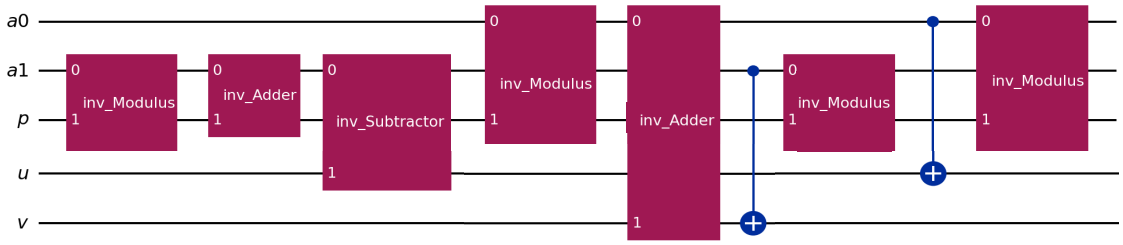


Figure 5. The inverse QNTT quantum circuit.

The IQNTT is present in the JVG algorithm developed in the present study.

2.6. Shor's Algorithm

Shor's algorithm was designed to factor a composite number formed by multiplying two prime integers. This is done by finding the order r such as $x^r \equiv mod\ N$, being N the composite number to be factored and x a random initial value. The quantum circuit for the order finding used in Shor's algorithm is represented in Figure 6.

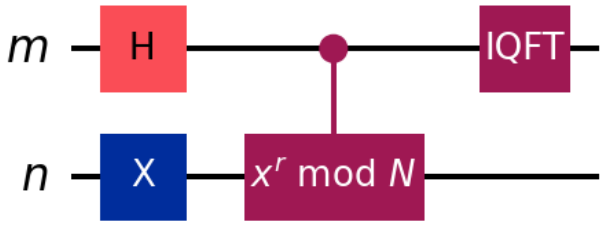


Figure 6. The quantum circuit for order finding in Shor's algorithm.

Figure 6 can be split into three main structures. The first consists of putting the first register $|m\rangle$, initialized in state $|0\rangle$, into superposition. After that, the quantum modular exponentiation computation is applied to the circuit in the form of controlled rotations. Here, the controls are the qubits in register m and the targets are the qubits in register $|n\rangle$, initialized in state $|1\rangle$. Note that the number of qubits in register $|m\rangle$ determine the size of the superposition, which impacts on the phase estimation, while the number of qubits in $|n\rangle$ needs to be sufficiently large to store integers up to N. The last part of the circuit is the inverse QFT in the $|m\rangle$ register. At the end of this circuit, the register $|m\rangle$ is measured, revealing information about the period of the function $f(r) = x^r\ mod\ N$ [18,39,44]. In the methodology here proposed, we substitute the IQFT circuit by the IQNTT. The novel quantum circuit for integer factorization in a quantum framework is presented in Figure 7.

 doi:10.20944/preprints202510.1649.v1

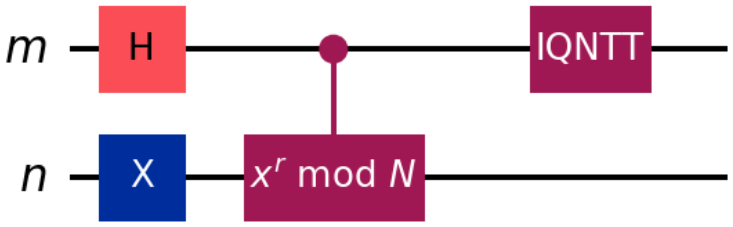


Figure 7. The proposed new quantum circuit for number factorization.

Differently from QFT in Shor's pipeline, the IQNTT retrieves the period of a given function $f(r)$ directly over a finite ring $\mathbb{Z}_p$. By changing spectral estimation from the complex domain to a ring-based spectrum, we extend the number-theoretic formulation of period finding offering a further understanding on its implementation.

Additionally, the QFT circuit requires extensive qubit interactions, which is hard to compute on present quantum hardware. Noisy Intermediate-Scale Quantum (NISQ) devices have limitations such as reduced coherent time per qubit, high error rates per gate, i.e. gate operation errors and gate fidelity, and restricted connectivity, al of which difficult feasibility of QFT circuits [63,64]. Contrariwise, the QNTT circuit (Figure 4) avoids rotations for input transformation, offering a simpler design. In this context, less complex implementations such as QNTT are preferable, as they offer a more efficient hardware-compatible framework for the current quantum devices [65].

3. Results

To evaluate the performance of the proposed methodology, we conducted experiments using two different approaches. The first one, we used Qiskit SDK to simulate the quantum circuit in a classical device. Table 1 has information about the hardware used for the quantum simulations in a Windows 11 operational system.

Table 1. Hardware for the quantum simulations using a classical computer.

RAM (GB)	CPU	GPU
32	13th Generation i7	2 X RTX 4080 16 GB

The second part consisted in implementing the same algorithms in an IBM's quantum computer. For both cases, we considered the following composite numbers to be factored using QFT-based and QNTT-based algorithms: 15, 21, 143, 1363 and 67297. The selection of these values was because we need to assess the algorithms' performance for composite values containing different number of digits. For each one of these values, the total number of qubits used in the circuit also changes. We also tried larger composite numbers, but due to hardware limitations, given that simulating quantum operations in classical devices is still very demanding [66,67], the algorithms failed to run. At this stage, it is relevant to note that both methodologies compared JVG against the Shor's algorithm implemented by IBM [68].

It is important to emphasize that the purpose is to demonstrate and verify the underlying hypothesis, demonstrating that the proposed JVG algorithm consistently exhibits scalability trends, requiring fewer resources than the standard Shor's algorithm, as problem size increases. While the present results do not target a large number of factorizations, i.e. RSA-2048, they establish a clear trajectory of improvement, indicating that the same principles could translate into significant resource savings in the given NISQ era.

Table 2 contains information about the composite numbers, their digits and the total amount of qubits in their respective circuits.

 doi:10.20944/preprints202510.1649.v1

Table 2. Composite numbers evaluated for both circuits, their factors, and the number of qubits in each circuit.

Composite Number (factors)	Number of Qubits in the Quantum Circuit
15 (3, 5)	18
21 (3, 7)	22
143 (11, 13)	34
1363 (47, 29)	46
67297 (173, 389)	70

To maintain clarity between simulated and experimental outcomes, all performance metrics in this work are reported within their native execution contexts. The simulation-based results employ Qiskit's gate model using the CX, U, and SWAP primitives, along with wall-clock execution time, memory consumption (RAM) and circuit depth as system-level metrics. These measurements capture algorithmic behavior independent of hardware-specific constraints. Conversely, the quantum computer evaluations are based on IBM Q backends that utilize a different native gate basis, primarily SX, CZ, RZ, and X, which include Qiskit Runtime (QR) execution time and circuit depth as indicators of resource demand. Because these gate sets are not directly equivalent, cross-domain comparisons are made in relative terms using normalized quantities such as circuit depth and runtime scaling rather than 1:1 gate count. This ensures that simulator results reflect algorithmic scaling properties, while hardware results capture physical implementation behavior under realistic noise and transpilation conditions. For each setup, each circuit was sampled 1024 times.

3.1. Results for Quantum Simulation

To account for stochastic effects in the simulation, each experiment was repeated 10 times. The mean and standard deviation were computed for every configuration across these runs. The average coefficient of variation is reported in Tables 3 and 4, for both approaches.

Table 3. Results for the simulations of number factorization using Shor's QFT-based Algorithm.

Composite Number (qubits)	Run Time (s)	RAM Usage (MB)	CX	U	SWAP	Circuit Depth
15 (18)	1.01 ± 0.01	419.42 ± 1.22	10541 ± 6.9	13971 ± 10	7382 ± 38	19071 ± 63
21 (22)	2.15 ± 0.02	488.39 ± 7.61	21840 ± 6	29448 ± 18	16688 ± 50	39217 ± 101
143 (34)	13.16 ± 0.13	1158.71 ± 10.30	109994 ± 9	153827 ± 40	95743 ± 149	176154 ± 447
1363 (46)	40.91 ± 0.63	2794.11 ± 26.30	344116 ± 12	490193 ± 42	280284 ± 669	454070 ± 1143
67297 (70)	174.11 ± 2.40	12504.66 ± 297.68	1713476 ± 16	2485724 ± 130	1293673 ± 1262	1687597 ± 2730
Average Coefficient of Variation	1.17 %	1.21 %	0.02 %	0.03 %	0.26 %	0.25 %

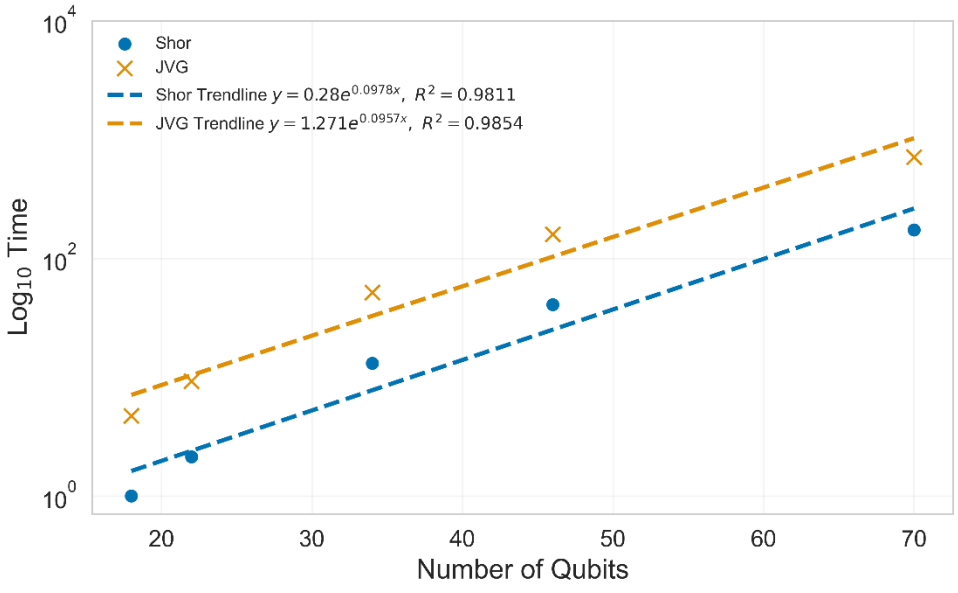
 doi:10.20944/preprints202510.1649.v1

Table 4. Results for the simulations of number factorization using JVG's QNTT-based Algorithm.

Composite Number (qubits)	Run Time (s)	RAM Usage (MB)	CX	U	SWAP	Circuit Depth
15 (18)	4.73 ± 0.06	431.77 ± 7.19	13843 ± 9	17747 ± 15	9076 ± 40	25890 ± 97
21 (22)	9.28 ± 0.14	517.78 ± 2.72	26251 ± 7	34471 ± 15	19099 ± 84	48317 ± 195
143 (34)	51.80 ± 0.39	1218.30 ± 10.48	117683 ± 16	162505 ± 29	100419 ± 249	192387 ± 447
1363 (46)	159.63 ± 1.15	2888.28 ± 24.47	355016 ± 16	502466 ± 64	287695 ± 351	477407 ± 910
67297 (70)	715.10 ± 9.03	11785.87 ± 67.17	1730625 ± 19	2504965 ± 128	1305584 ± 2319	1721957 ± 4026
Average Coefficient of Variation	1.08 %	0.89 %	0.02 %	0.03 %	0.28 %	0.29 %

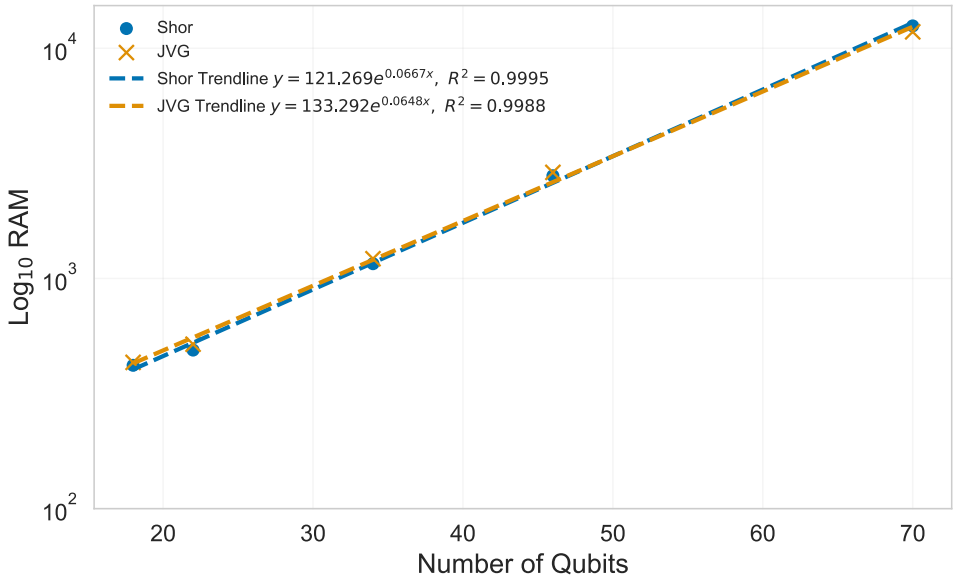
Tables 3 and 4 indicate that for the evaluated composite values, the QFT-based circuit yields better performance in terms of run time, RAM usage and gate count. It is possible to notice that the standard variation presents relatively small value compared to the mean. For both QFT and QNTT applications, the average coefficient of variation remained well below a 2% threshold. Considering run time and RAM, the JVG methodology reached an average of 1.08% and 0.89%, lower values than 1.17% and 1.21% calculated for the other methodology. The standard deviation for the gate count also revealed minimal, and both approaches reached very similar results. The same conclusion can be drawn for the circuit depth, where QFT-based technology reached 0.25% and the QNTT one 0.29%.

Figure 8 shows the plots for each metric using Log_{10} scale for better visualization of the data points.



(a)

 doi:10.20944/preprints202510.1649.v1

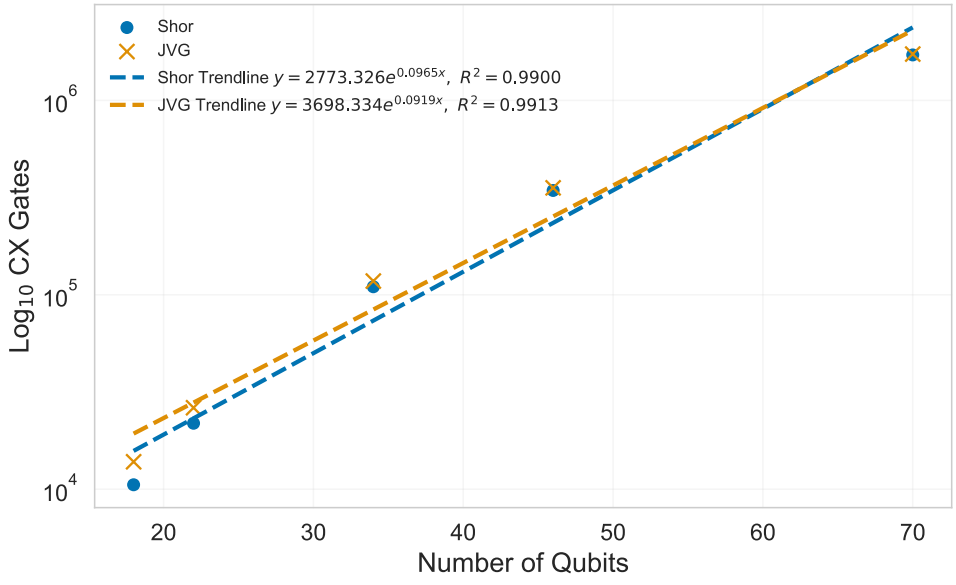


(b)

Figure 8. Plots for the run time (a) and the used RAM (b) on a Log_{10} scale.

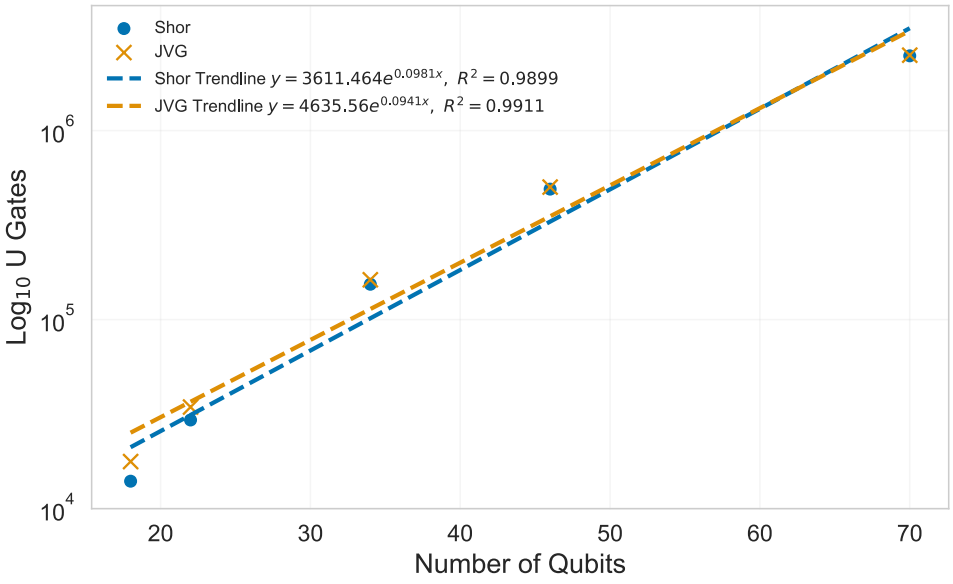
In Figure 8, the trendlines are of exponential form, indicating that both cases closely follow the exponential behavior. Figure 8 (a) shows a noticeable difference between the two approaches, where the QNTT-based model requires over 700 s to finish, when compared to the 170 s of its counterpart for N = 67297. However, for the same N, the QNTT alternative presents superior RAM usage values, needing on average 11.8 GB of memory, compared to 12.6 GB for the other quantum circuit. Additionally, Figure 8 (b) reveals that the trendline corresponding to the QFT-based approach surpasses the one representing the QNTT alternative.

A more careful analysis of the gate count also reveals that number factorization using QNTT circuit has superior scalability than the traditional Shor's circuit. This becomes clear when evaluating the plots of each metric versus the number of qubits in the circuits depicted in Figure 9.

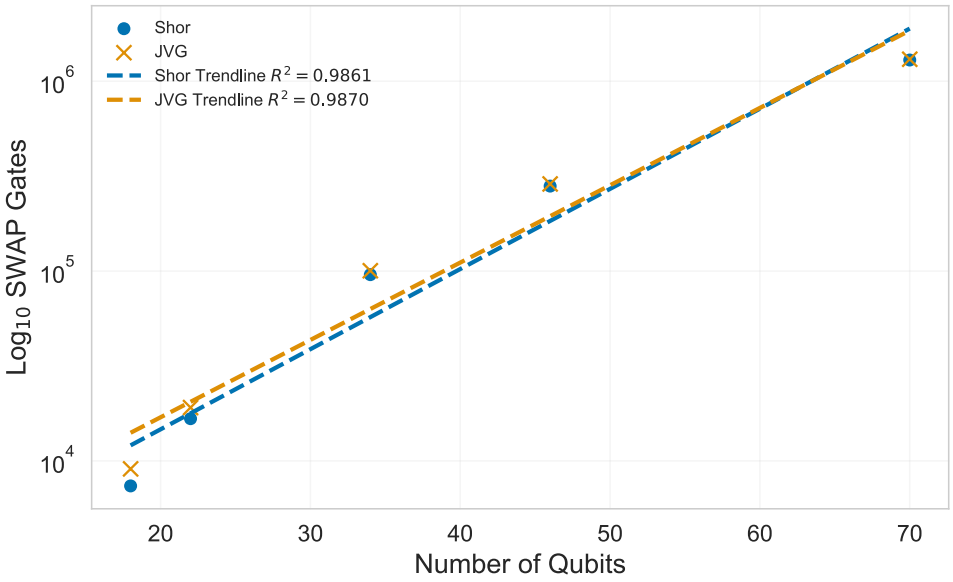


(a)

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 October 2025 doi:10.20944/preprints202510.1649.v1

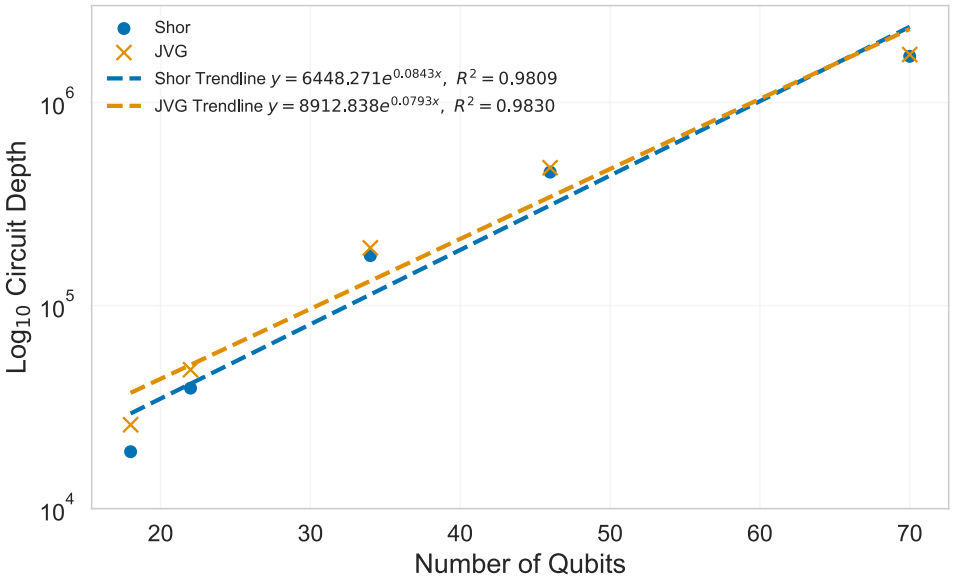


(b)



(c)

 doi:10.20944/preprints202510.1649.v1



(d)

Figure 9. Plots for gates CX (a), U (b), SWAP (c), and Circuit Depth (d), on a Log_{10} scale.

The trendlines in Figure 9 are also of exponential form. It reveals that JVG's QNTT-based factorization scales better than Shor's QFT. Examination of each panel shows that the gap seen at smaller composite values, like 15, 21 and 143, reduces as the number of qubits in the circuit grows. For the last considered point N = 67297, the gap becomes marginal, indicating that both methodologies achieve similar performance.

Additionally, a direct comparison between the increase of resource usage for each metric, also indicates improved scalability performance of the proposed methodology QNTT-based. Table 5 summarizes this information.

Table 5. Comparison between the increase on resources for Shor's QFT-based and JVG's QNTT-based algorithms on simulated results.

	Increase on Run Time (%)	Increase on RAM usage (%)	Increase on CX Gates (%)	Increase on U Gates (%)	Increase on SWAP Gates (%)	Increase on Circuit Depth (%)
Shor's QFT	17209	2881	16155	17692	17426	8749
JVG's QNTT	15007	2630	12401	14015	14285	6551

Table 5 refers to the increase in resources between the first composite number, N=15, and the last one evaluated, N = 67297, using the two methodologies. The data highlights that as circuit size increases, the QFT-based algorithm demands a substantially higher proportional growth in resources than the QNTT-based approach. For instance, the QFT circuit exhibited an increase of 17209% in runtime and 2881% in memory usage, whereas the QNTT circuit required reduced resources, 15007% and 2630% respectively.

The same pattern holds for gate counts and Circuit Depth. For gate count, the most significant difference is observed for CX gate, which increased by over 16155% for QFT, significantly more than

 doi:10.20944/preprints202510.1649.v1

12401% for the same gate present in QNTT. The circuit depth of the traditional Shor's algorithm also increased by 8749%, a considerably larger value compared to 6551% for the JVG methodology.

3.2. Results for Implementation on a Real Quantum Hardware

We also repeated the circuits ten times when implementing the circuits in a quantum device. However, due to the limitations of the processing time in quantum devices, the number of runs for N = 1363 was limited to six, as it demanded the largest amount of processing time. Different from the simulation methodology, we could not run the circuits for N = 67297. This was due to the current constraints of the NISQ devices, as previously discussed. They still offer limited coherence time and qubit connectivity, which prevent more profound and more complex quantum circuits from being implemented. At this phase, the quantum device used in our methodology was "ibm_torino", which uses the Heron R1 quantum processor and has 133 qubits. We recommend addressing reference [69] for further information on this device and its processor.

Additionally, for every configuration, the mean and standard deviation were calculated across these runs, in the same fashion previously described. Tables 6 and 7 contain the results for implementing the algorithms QFT and QNTT-based, respectively.

Table 6. Results for the implementation of number factorization using Shor's QFT-based Algorithm on a real quantum computer.

Composite Number (qubits)	QR (s)	SX	CZ	RZ	X	Circuit Depth
15 (18)	4.0 ± 0	54390 ± 230	26444 ± 129	24761 ± 142	325 ± 26	52468 ± 494
21 (22)	6.8 ± 0.9	116659 ± 371	56964 ± 200	50735 ± 184	511 ± 34	107293 ± 650
143 (34)	26.8 ± 0.9	642895 ± 1098	314000 ± 536	254806 ± 456	1766 ± 39	492183 ± 1217
1363 (46)	67.8 ± 2.5	2085030 ± 2554	1013958 ± 1468	805239 ± 1627	4616 ± 84	1373924 ± 4338
Average Coefficient of Variation	5.15 %	0.26 %	0.29 %	0.33 %	4.67 %	0.53 %

Table 7. Results for the implementation of number factorization using JVG's QNTT-based Algorithm on a real quantum computer.

Composite Number (qubits)	QR (s)	SX	CZ	RZ	X	Circuit Depth
15 (18)	5.0 ± 0	68438 ± 346	33483 ± 189	32963 ± 111	527 ± 26	70856 ± 485

 doi:10.20944/preprints202510.1649.v1

21 (22)	7.9 ± 0.6	135928 ± 282	66637 ± 140	61612 ± 202	796 ± 28	131552 ± 552
143 (34)	27.1 ± 1.7	678111 ± 944	331635 ± 511	274170 ± 494	2263 ± 57	535538 ± 1233
1363 (46)	68.3 ± 2.2	2137997 ± 518	1040125 ± 361	833921 ± 1116	5343 ± 60	1432496 ± 3358
Average Coefficient of Variation	2.64 %	0.22 %	0.24 %	0.24 %	3.02 %	0.39 %

As presented in Tables 6 and 7, both methodologies exhibited stable behavior across repeated runs, with relative standard deviations remaining within a few percentage points of the mean. On average, the JVG algorithm achieved a 2.64% average coefficient of variation in QR, about half of the reported value 5.15% for the QFT-based circuit. The JVG implementation maintained lower or comparable variability for the primary quantum gate metrics, reaching 0.22% for SX, 0.24% for CZ, and 0.24% for RZ gates. The QFT version reached 0.26%, 0.29%, and 0.33%, respectively. The X-gate variance followed a similar pattern, with 3.02% coefficient of variation for QNTT and 4.67% for QFT. Circuit Depth fluctuations also remained limited, averaging 0.39% and 0.53% for QNTT and QFT, respectively.

Tables 6 and 7 also reveal that the QFT-based circuit yields better results in terms of runtime and gate count for an N = 1363, as observed from Tables 3 and 4. At no evaluated condition did the QNTT surpass the performance of the traditional QFT implementation. This could be explained by the fact that limited composite numbers were investigated, as previously discussed. Still, it is interesting to observe that when the quantum circuit involves the highest number of qubits, the standard deviation for all the metrics regarding the QNTT approach is significantly less than its QFT alternative, which may indicate a more stable circuit.

Figure 10 presents the plot comparing QR versus the total amount of qubits in the circuit for each composite number.

 doi:10.20944/preprints202510.1649.v1

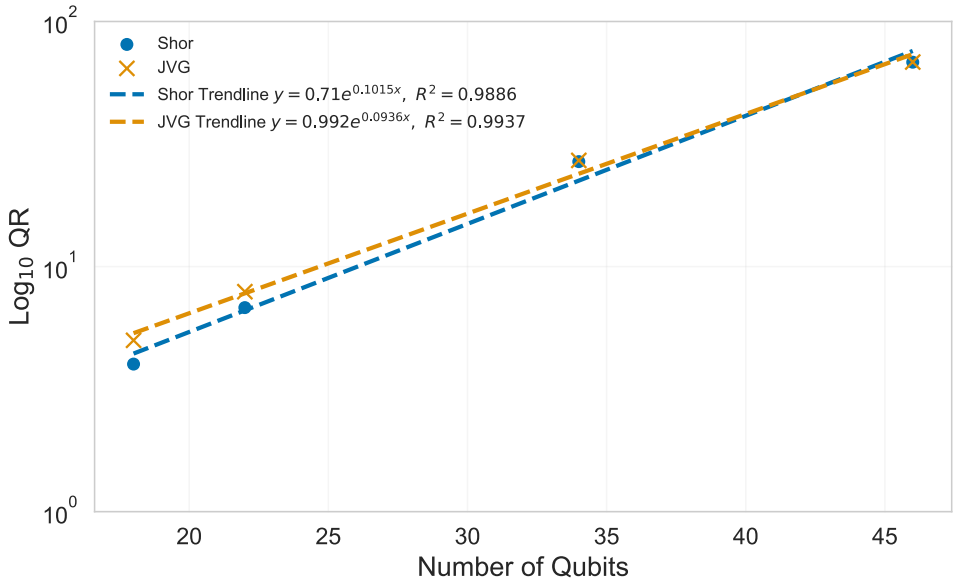
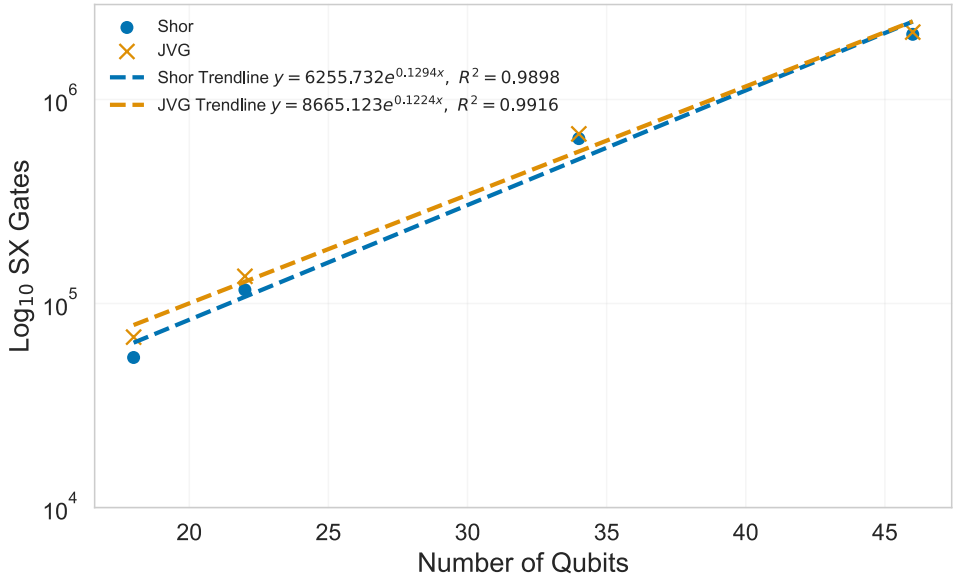


Figure 10. Plot for Qiskit Runtime, on a Log_{10} scale.

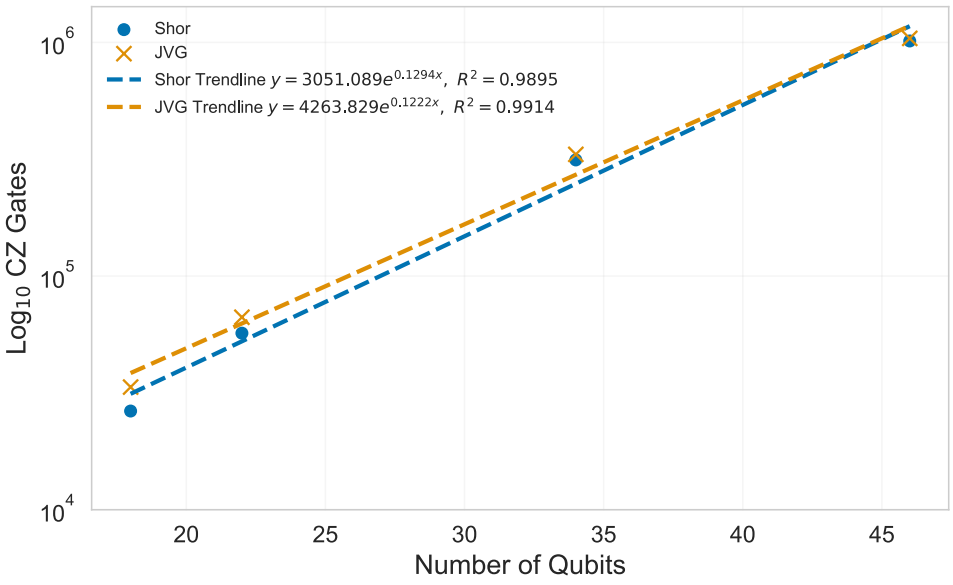
The trendline in Figure 10 is of exponential form. The coefficient R^2 suggests good agreement of the data for Shor's QFT and JVG's QNTT approaches, being around 99% for both cases. Additionally, the trendline indicates that with the increase of qubits in the circuit, the difference between the two methodologies decreases substantially. This was not the case for the simulated results, where the two lines (Figure 8) were further apart, but with an indication to come closer together as the circuit increases. Finally, the trendline for QFT at last point is marginally surpassing the one for QNTT, reinforcing its superior scalability capacity.

Figure 11 summarizes the plots for gate count and Circuit Depth.

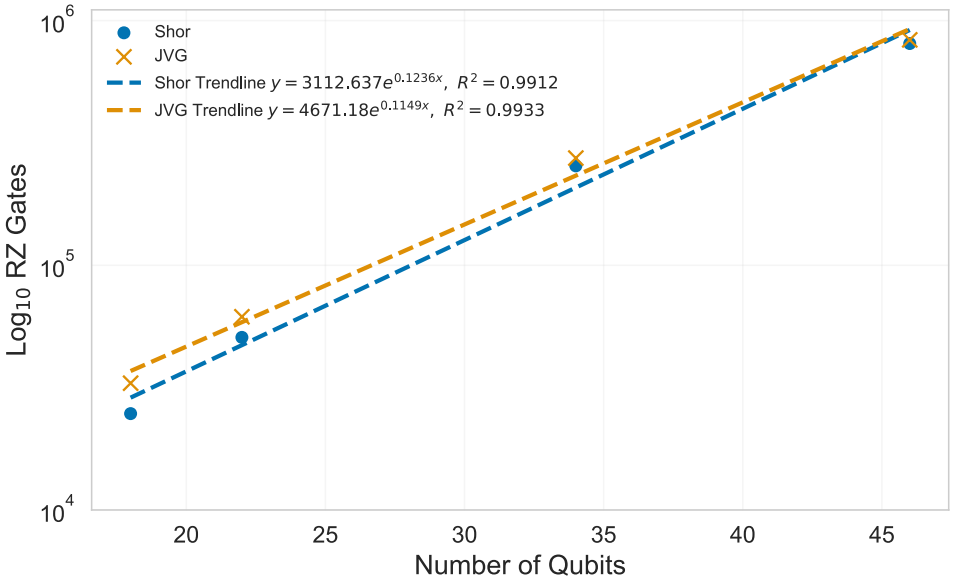


(a)

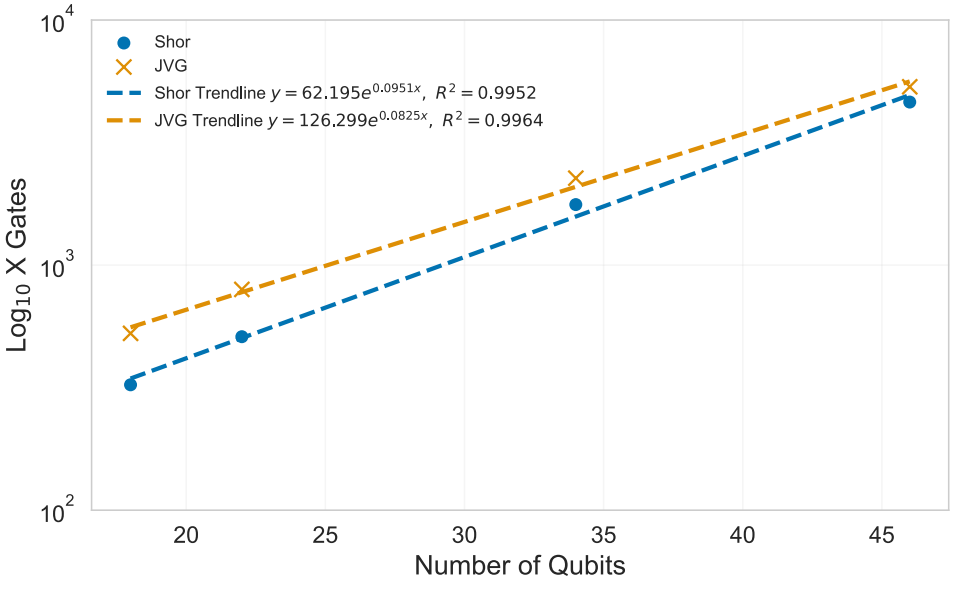
 doi:10.20944/preprints202510.1649.v1



(b)

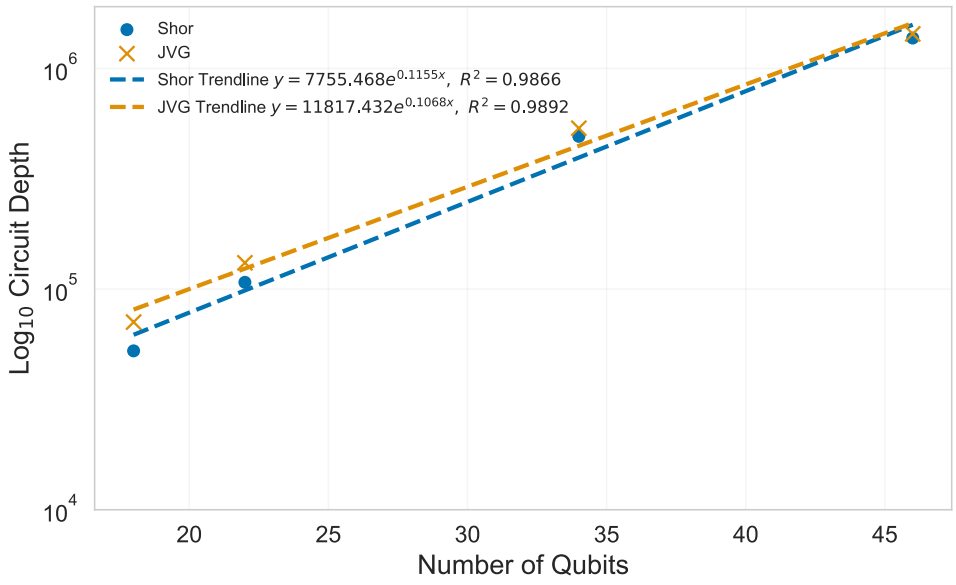


(c)



(d)

 doi:10.20944/preprints202510.1649.v1



(e)

Figure 11. Plots for the quantum gates SX (a), CZ (b), RZ (c), X (d), and Circuit Depth, on a Log_{10} scale.

The trendlines in the plots in Figure 11 are also exponential, indicating good agreement with the plotted data. For all the experimental results, the JVG approach reduced the gap compared to the QFT, with the increasing number of qubits in the circuit, which complies with the observed trend for the simulated results. The most significant one was the number of X gates, where the gap between the points of both the approaches presented. The circuit depth also shows a substantial decrease in the gap between the two algorithms, revealing that they exhibit comparable performance in this metric.

Table 8 contains data comparing the increase in computational resources for QFT and QNTT-based circuits when implemented in a real quantum device.

Table 8. Comparison between the increase on resources for Shor's QFT-based and JVG's QNTT-based algorithms on real quantum computer.

	Increase on QR (%)	Increase on SX Gates (%)	Increase on CZ Gates (%)	Increase on RZ Gates (%)	Increase on X Gates (%)	Increase on Circuit Depth (%)
QFT-Based	1596	3734	3734	3152	1321	2519
QNTT-Based	1267	3024	3006	2430	914	1922

Table 8 reveals the improved scalability of the QNTT model over the QFT counterpart. For each measured metric, as observed in Table 5, the increase in computational resources is significantly greater for the traditional QFT methodology. Notably, substantial differences are seen in the X gate count and circuit depth, where the JVG approach achieved considerably lower increases of 914% and 1922%, respectively, compared to 1321% and 2519% for the QFT-based algorithm.

 doi:10.20944/preprints202510.1649.v1

3.3. Projections for Simulated and Experimental Results

We fitted the exponential models for both simulation and experimental results using the trendlines present in Figures 8, 9, 10 and 11. These projections must be interpreted as indicative trends, rather than precise forecasts, since extrapolations beyond the measured range inherently carry uncertainty, since error accumulation and hardware-specific restrictions may alter scaling at larger qubit counts. Nevertheless, these results remain relevant, as they capture the underlying trend behavior and provide a qualitative description to anticipate the feasibility of future implementations.

3.3.1. Simulation

Although Shor's QFT-based configuration still achieves lower absolute runtime, memory usage, gate count and circuit depth for the tested values, the relative growth rate of these indicators is consistently lower for the JVG algorithm. This means that, while both models experience exponential growth in resource requirements, the rate of increase for JVG's QNTT is smaller, indicating that the additional computational cost rises more slowly than for QFT, providing strong evidence that the QNTT-based circuit manages resource growth more efficiently. To better visualize the scalability for the proposed JVG algorithm, we extrapolated the results for additional five points. In this extrapolation, we consider 100, 150, 200, 250 and 300 qubits using the trendline equations displayed in Figures 8 and 9. The extrapolated results for run time and RAM usage are shown in Figure 12. The raw data is presented in Table 9.

Table 9. Projected values for simulated Shor's QFT-based and JVG's QNTT-based algorithms.

	Projected Run Time (s)		**Projected RAM (MB)**		**Projected CX Gates**		**Projected U Gates**		**Projected SWAP Gates**		**Projected Circuit Depth**	
Qubits	**QFT**	**QNTT**	**QFT**	**QNTT**	**QFT**	**QNTT**	**QFT**	**QNTT**	**QFT**	**QNTT**	**QFT**	**QNTT**
100	4.95E+03	1.82E+04	9.56E+04	8.69E+04	4.30E+07	3.62E+07	6.58E+07	5.66E+07	3.49E+07	3.08E+07	2.95E+07	2.48E+07
150	6.58E+05	2.18E+06	2.68E+06	2.22E+06	5.36E+09	3.59E+09	8.88E+09	6.25E+09	4.51E+09	3.35E+09	2.00E+09	1.31E+09
200	8.75E+07	2.61E+08	7.54E+07	5.67E+07	6.68E+11	3.55E+11	1.20E+12	6.91E+11	5.81E+11	3.65E+11	1.35E+11	6.89E+10
250	1.16E+10	3.12E+10	2.12E+09	1.45E+09	8.32E+13	3.51E+13	1.62E+14	7.64E+13	7.50E+13	3.97E+13	9.17E+12	3.63E+12
300	1.55E+12	3.74E+12	5.94E+10	3.69E+10	1.04E+16	3.48E+15	2.18E+16	8.44E+15	9.68E+15	4.32E+15	6.21E+14	1.91E+14

 doi:10.20944/preprints202510.1649.v1

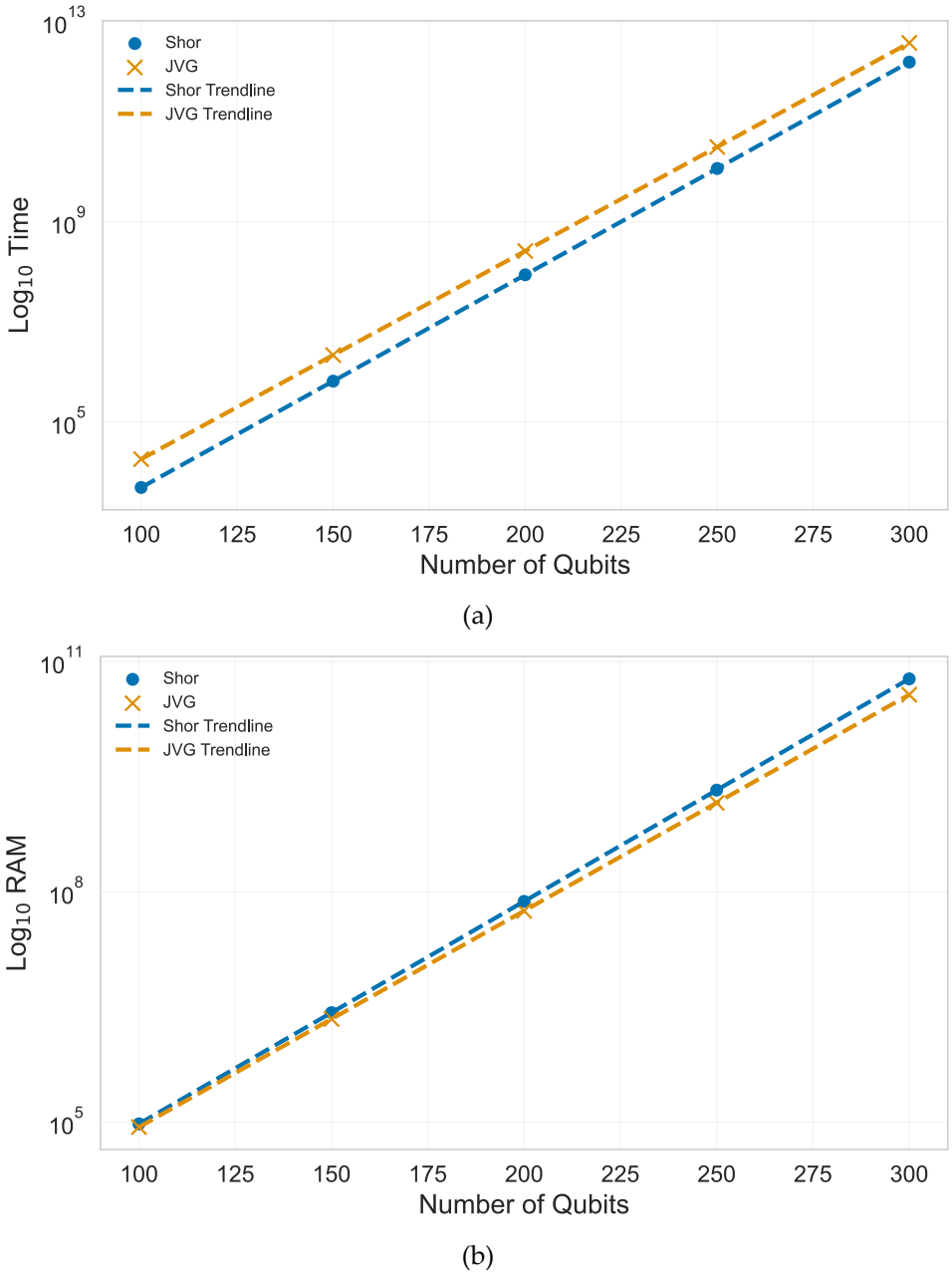
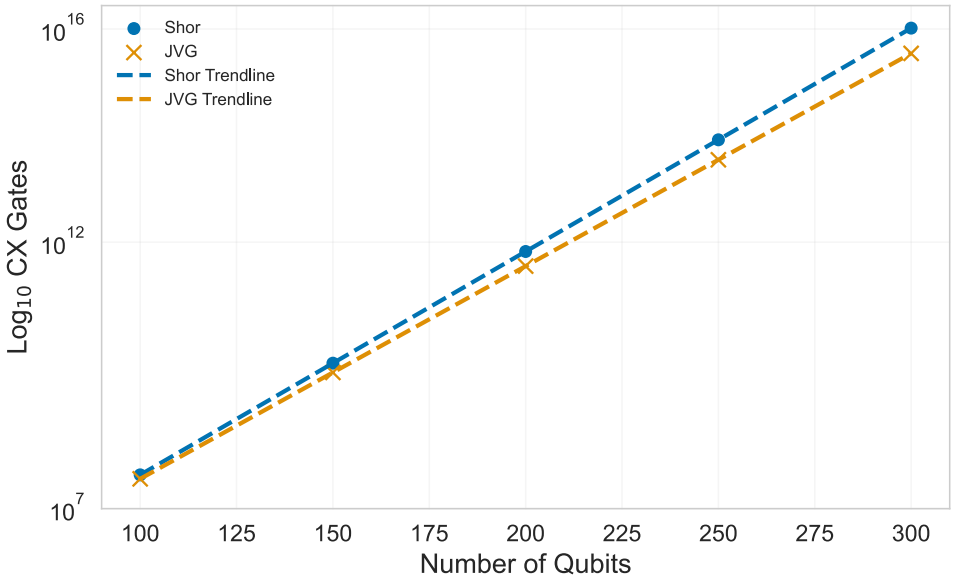


Figure 12. Extrapolated results for run time (a) and RAM usage (b), on a Log_{10} scale.

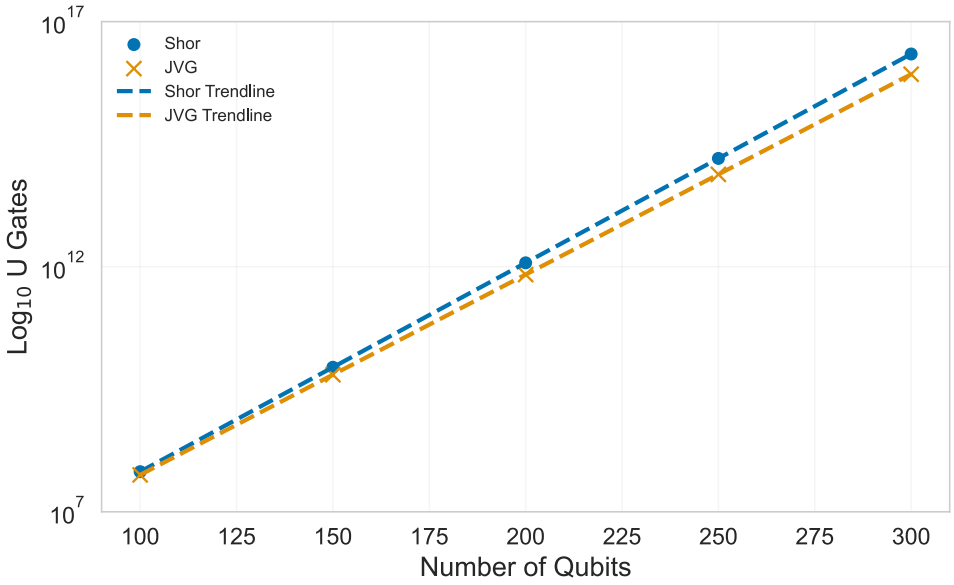
The run time in Figure 12 (a) still shows the QFT model ahead of the QNTT-based one, considering the simulated results. However, the trendlines suggest that the performance gap between the two algorithms narrows as the number of qubits increases, even though the JVG methodology has not yet surpassed the QFT in this metric. Conversely, RAM usage becomes more discrepant with qubit number grow, as shown in Figure 12 (b). For the last point in the extrapolated plot, the projection indicates that the traditional Shor's algorithm will require 38% more memory, representing a significant difference compared to the JVG methodology for the same configuration.

The same fashion is visualized considering the number of gates and circuit depth projections. Figure 13 compiles the extrapolated plots for each tracked metric.

 doi:10.20944/preprints202510.1649.v1

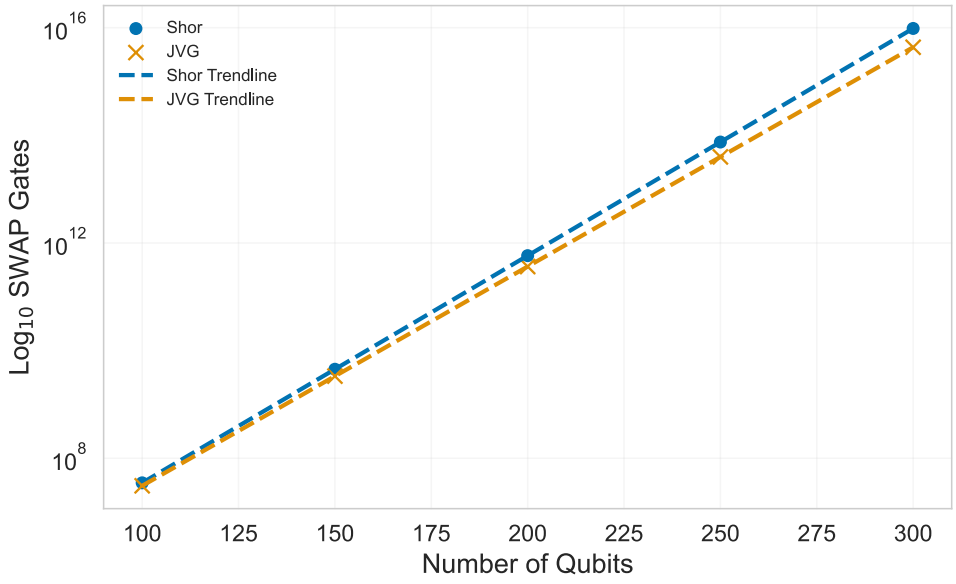


(a)

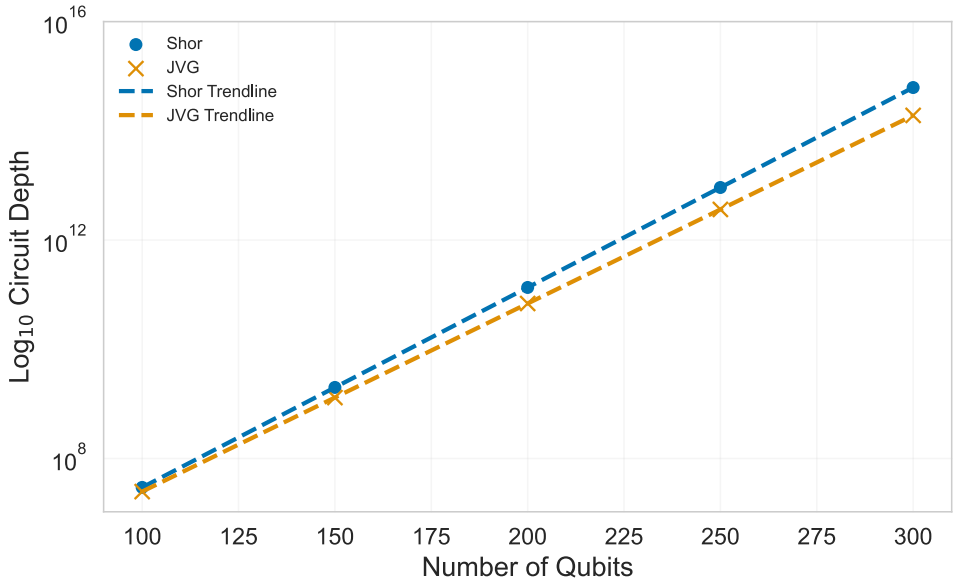


(b)

 doi:10.20944/preprints202510.1649.v1



(c)



(d)

Figure 13. Extrapolated results for gates CX (a), U (b), SWAP (c), and Circuit Depth (d).

Figure 13 reveals that considering the extrapolated data points, the JVG algorithm ultimately achieves superior results. This holds true for all the metrics. For the extreme case when 300 qubits are considered, the proposed algorithm can reduce the use of computational resources by 66%, 61%, 55% and 69% for CX, U, SWAP, and Circuit Depth, respectively. This is compelling evidence of the scalability achieved by JVG methodology over the traditional implementation of Shor's algorithm.

3.3.2. Experimental

Similar approach was adopted for assessment of experimental results. Considering that the "ibm_torino" has a total amount of 133 qubits, we assumed 70, 85, 100, 115, and 130 qubits for the projections. The results for the projected outcomes for the QR are depicted in Figure 14, and the raw projected data from the trendline equations in Figures 10 and 11 is given in Table 10.

 doi:10.20944/preprints202510.1649.v1

Table 10. Projected values for experimental Shor's QFT-based and JVG's QNTT-based algorithms.

Qubits	Projected QR (s)		Projected SX Gates		Projected CZ Gates		Projected RZ Gates		Projected X Gates		Projected Circuit Depth	
	QFT	QNTT	QFT	QNTT	QFT	QNTT	QFT	QNTT	QFT	QNTT	QFT	QNTT
70	8.64E+02	6.95E+02	5.37E+07	4.56E+07	2.62E+07	2.21E+07	1.78E+07	1.45E+07	4.84E+04	4.07E+04	2.52E+07	2.09E+07
85	3.96E+03	2.83E+03	3.74E+08	2.86E+08	1.82E+08	1.38E+08	1.14E+08	8.15E+07	2.02E+05	1.40E+05	1.42E+08	1.04E+08
100	1.82E+04	1.15E+04	2.61E+09	1.79E+09	1.27E+09	8.65E+08	7.26E+08	4.57E+08	8.39E+05	4.83E+05	8.05E+08	5.14E+08
115	8.32E+04	4.69E+04	1.82E+10	1.12E+10	8.86E+09	5.41E+09	4.64E+09	2.56E+09	3.49E+06	1.67E+06	4.55E+09	2.55E+09
130	3.82E+05	1.91E+05	1.26E+11	7.05E+10	6.17E+10	3.38E+10	2.96E+10	1.43E+10	1.46E+07	5.74E+06	2.57E+10	1.27E+10

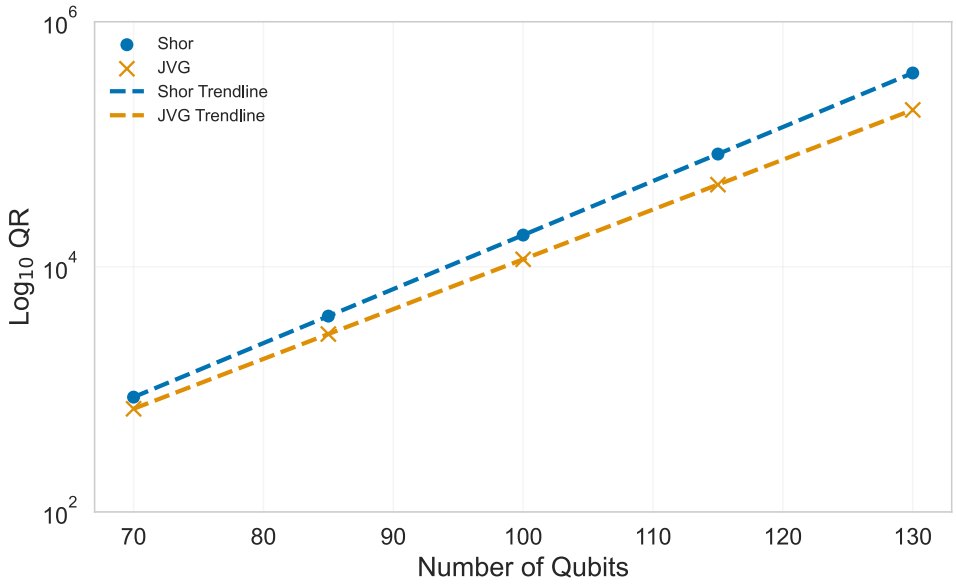
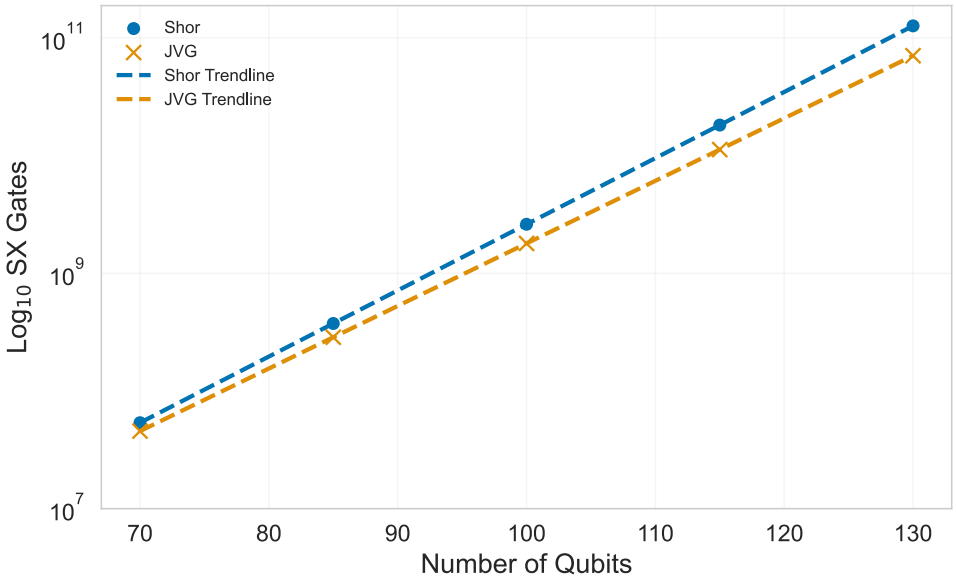


Figure 14. Plot for the extrapolated Qiskit Runtime, on a Log_{10} scale.

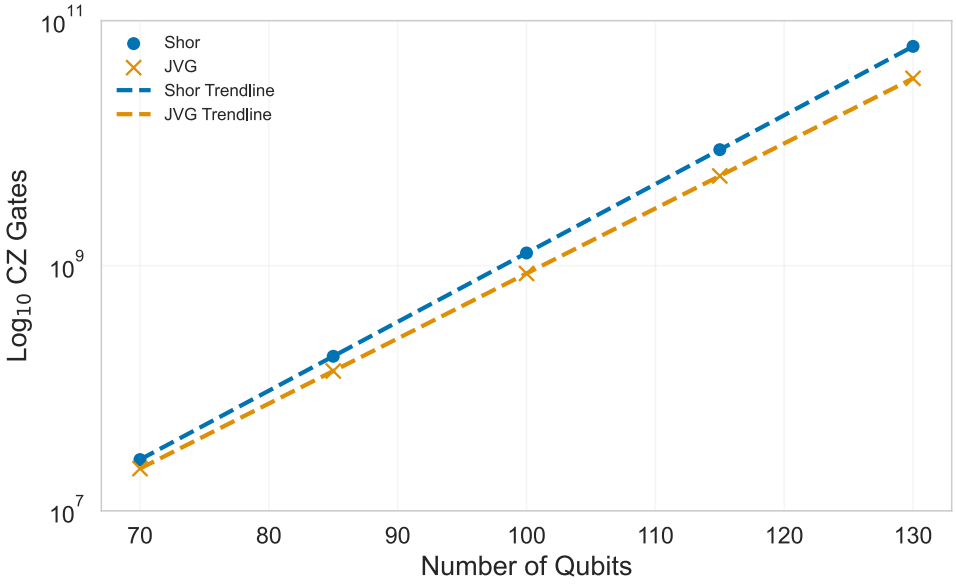
From Figure 14, it is possible to observe that the processing time required to run both algorithms increases significantly, according to the projections. For an initial extrapolation of 70 qubits, it is still noticeable that the JVG approach already requires less time to compute the factors of a given composite number. This remains true for the following points in the projection, indicating that the proposed algorithm can already deliver tangible performance improvements in real-world scenarios for currently available quantum hardware.

Figure 15 presents the projections for the total amount of quantum gates and Circuit Depth considering the experimental results.

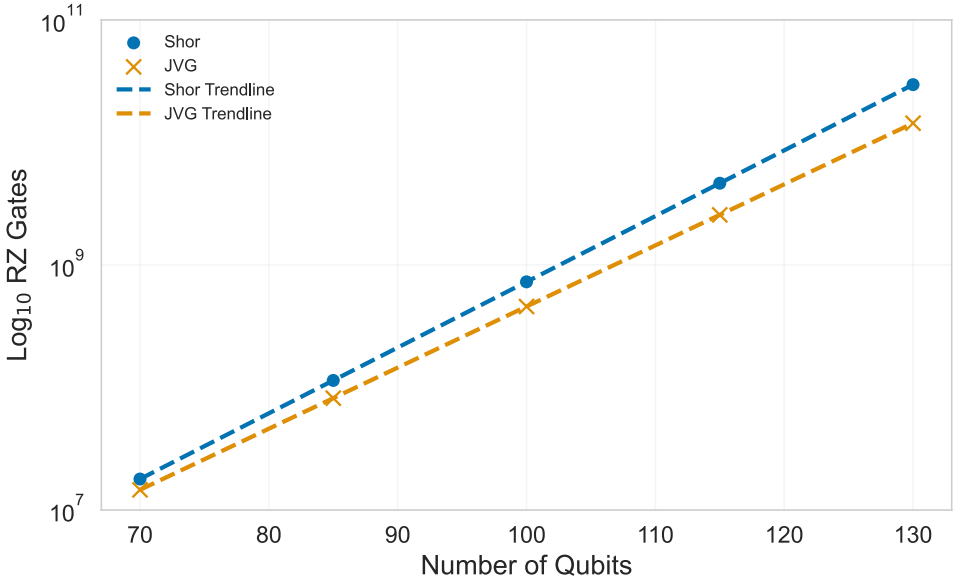
 doi:10.20944/preprints202510.1649.v1



(a)

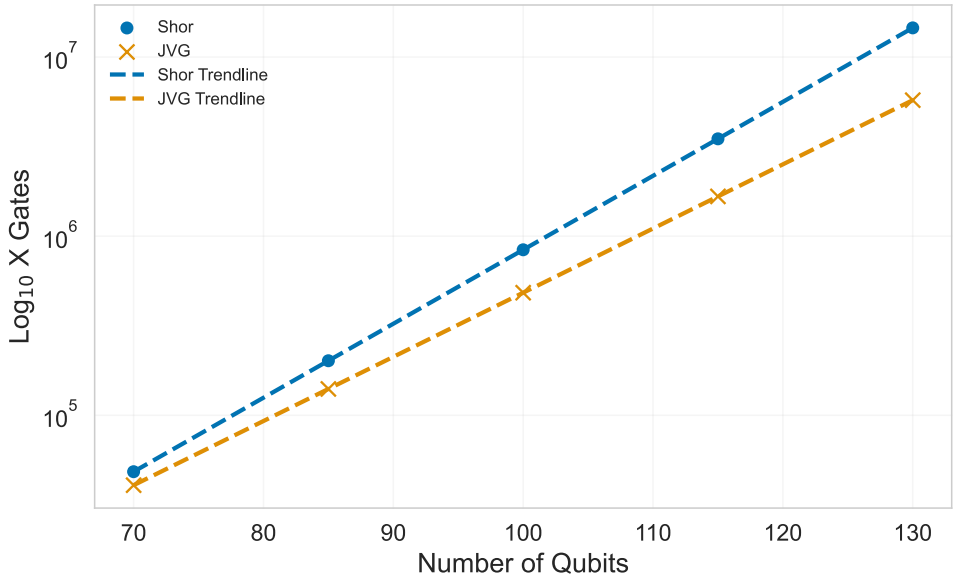


(b)

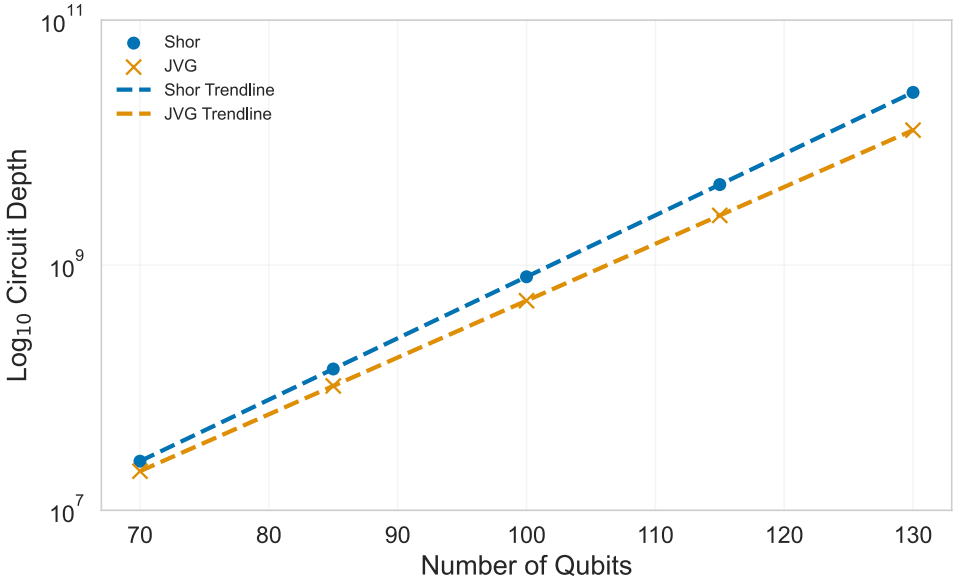


 doi:10.20944/preprints202510.1649.v1

(c)



(d)



(e)

Figure 15. Plots for the projected values for quantum gates SX (a), CZ (b), RZ (c), X (d), and Circuit Depth (e). Note that the vertical axis is on a Log_{10} scale.

The projected values in Figure 15 confirm the superior scalability characteristic of the QNTT-based factorization. For all the metrics evaluated in the benchmark, the initial case considering a quantum circuit composed of 70 qubits is already sufficient to showcase the JVG's superiority over the traditional Shor's implementation. As previously mentioned, the growth in the number of qubits renders the difference between the two approaches, i.e. the gap between the plotted points, more compelling. These are strong findings, particularly considering that the projections are based on a real quantum computer. Finally, the sustained performance advantage observed at higher qubit counts indicates that the QNTT framework is well designed for future large-scale quantum architectures. It offers a realistic path toward efficient and hardware-compatible implementations of integer factorization on current NISQ devices.

 doi:10.20944/preprints202510.1649.v1

3.3.3. RSA-Sized Circuits Projections for Experimental Results

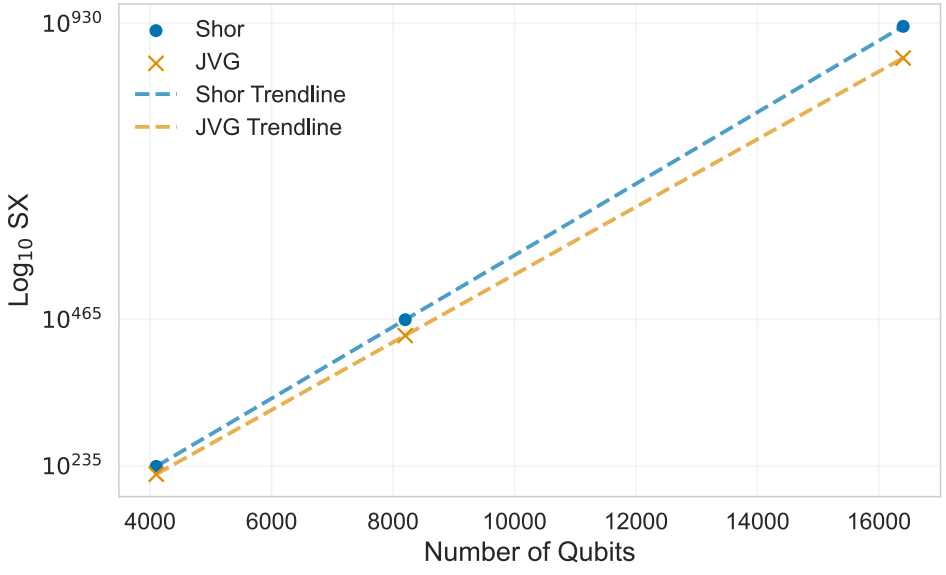
Even though the scope of the present study does not include the factorization of large RSA encryption keys, it is essential to investigate the scalability trends implied by the proposed methodology. To this end, the results obtained from the experimental benchmarks were extrapolated to estimate the expected behavior for larger circuit sizes representing RSA-1024, RSA-2048 and RSA-4096. For such scenarios, the JVG quantum circuit would require approximately 4100, 8200, and 16400 qubits, respectively.

This approach is justified by the observed trend in the experimental data, which suggested that the implementation of the JVG algorithm on real quantum hardware would require approximately 70 qubits to outperform the QFT-based model, while the simulated configuration would demand around 100 qubits to achieve superiority.

The projections provide insight into how the JVG algorithm could scale under cryptographically relevant conditions, offering an evidence-based indication of its potential efficiency and resource advantage as quantum hardware continues to evolve. The focus is primarily on the gate count and circuit depth, as these are the most representative indicators of feasibility considering the present NISQ architectures. Figure 16 shows the projections, and Table 11 contains the raw data.

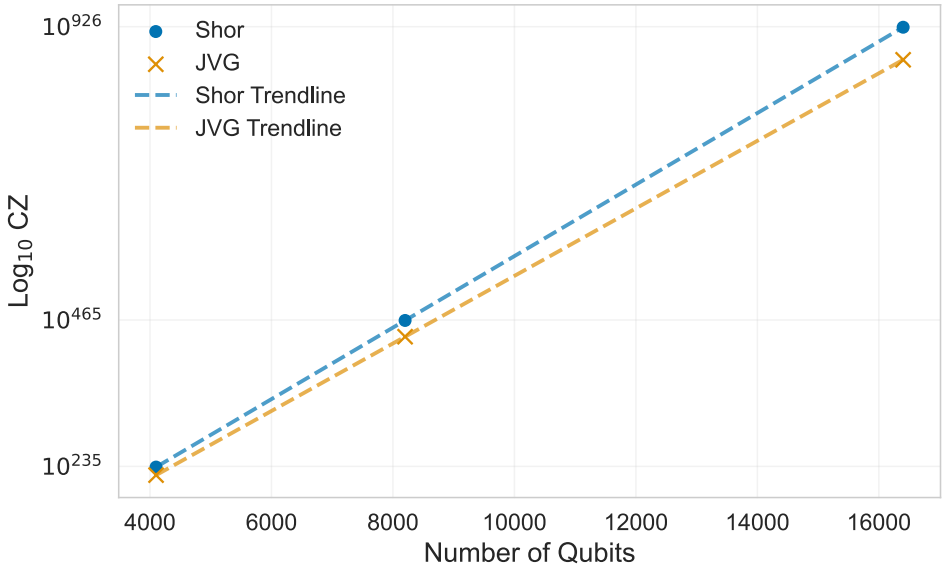
Table 11. Projected values for experimental Shor's QFT-based and JVG's QNTT-based algorithms for RSA-scale configurations.

Qubits	Projected SX Gates		Projected CZ Gates		Projected RZ Gates		Projected X Gates		Projected Circuit Depth	
	QFT	QNTT	QFT	QNTT	QFT	QNTT	QFT	QNTT	QFT	QNTT
4100	1.61E+234	7.66E+221	7.85E+233	1.66E+221	1.72E+178	1.66E+171	1.35E+171	1.00E+149	3.55E+209	1.74E+194
8200	4.14E+464	6.77E+439	2.02E+464	6.46E+438	8.19E+352	5.98E+338	2.92E+340	7.97E+295	1.62E+415	2.57E+384
16400	2.75E+925	5.29E+875	1.34E+925	9.79E+873	1.85E+702	7.70E+673	1.37E+679	5.03E+589	3.39E+826	5.60E+764

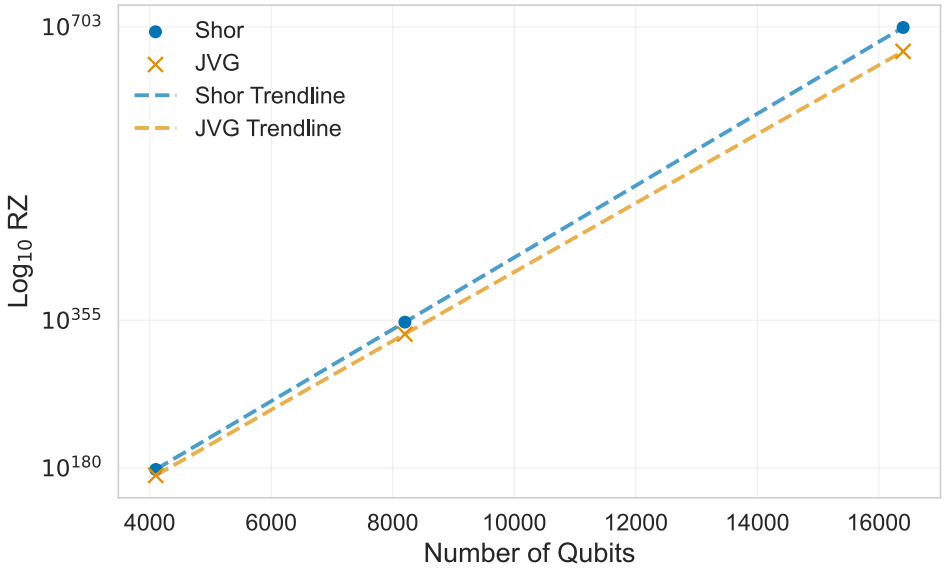


(a)

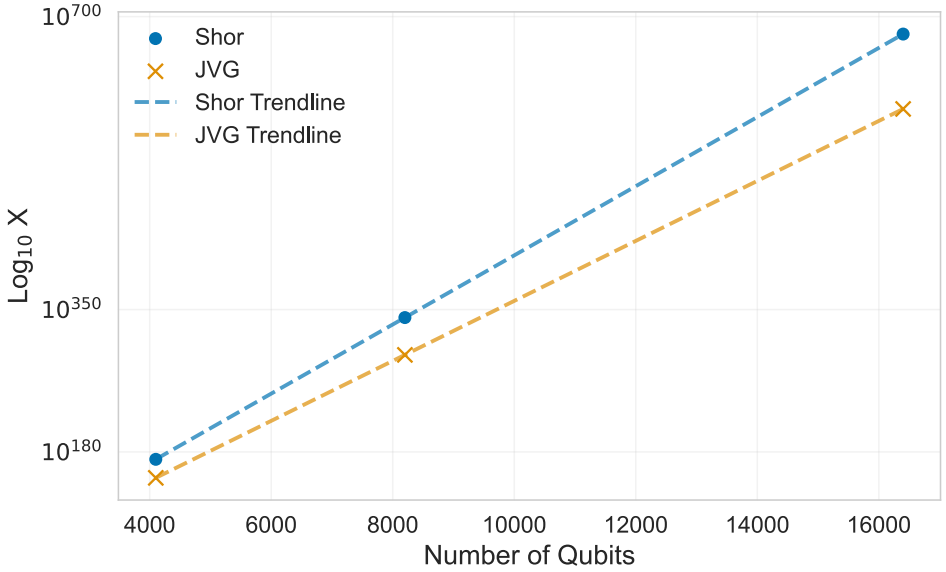
 doi:10.20944/preprints202510.1649.v1



(b)

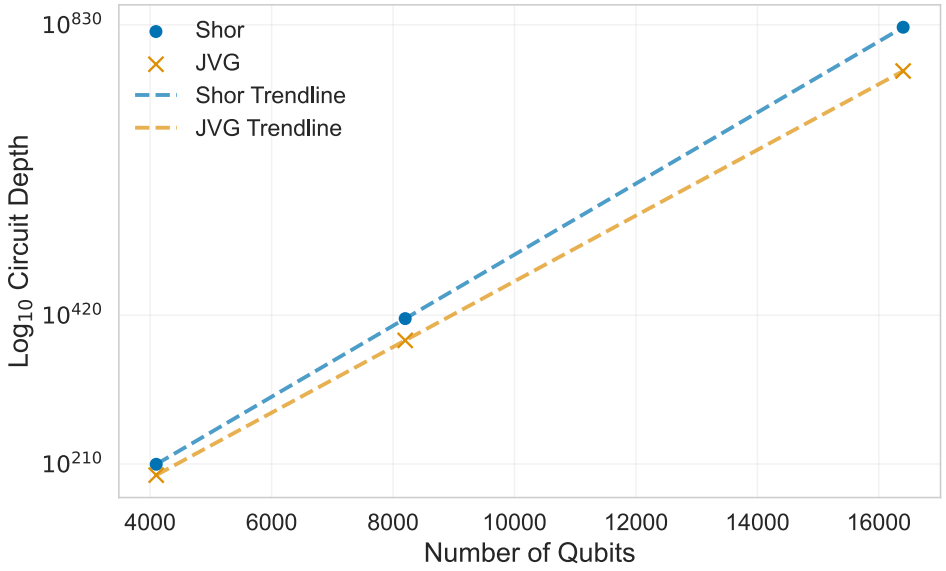


(c)



(d)

 doi:10.20944/preprints202510.1649.v1



(e)

Figure 16. Plots for the projected values for quantum gates SX (a), CZ (b), RZ (c), X (d), and Circuit Depth (e) for RSA-scale configurations. Note that the vertical axis is on a Log_{10} scale.

Figure 16 and Table 11 confirm the trend displayed for the previous projections. At RSA-1024, RSA-2048, and RSA-4096 configurations, the results reveal a significant difference in scaling behavior between the two approaches. For each gate type and circuit depth, the projected growth for the QFT-based circuit increases several orders of magnitude faster than for the QNTT-based JVG algorithm.

Considering the RSA-1024 projection (4100 qubits), the number of SX gates required by the QFT-based circuit requires approximately **13 orders of magnitude more resources** than JVG. This pattern persists and even amplifies for larger configurations, being remarkably prominent for X gates, where for the extreme RSA-4096 configuration, the reduction on the resource even larger, representing a substantial reduction by the JVG methodology. Circuit depth is another metric significantly impacted, according to these projections. The expected results indicate a reduction of **15 orders of magnitude** for RSA-1024 and larger orders of magnitude for RSA-2048 and RSA-4096.

These results demonstrate a consistent advantage for the JVG architecture. The slower growth observed for QNTT-based circuit indicates that the substitution of QFT with QNTT can effectively reduce circuit complexity and consequently reduce the cumulative error propagation. Although these values are not currently realizable on existing hardware, they provide a strong qualitative indication of the long-term scalability potential and hardware efficiency of the proposed algorithm in cryptographically relevant conditions.

4. Discussion

Before interpreting these findings, it is worth clarifying the conceptual boundary of our contribution. The JVG algorithm leverages the existing QNTT design of Lu et al. [58] but expands it through algorithm-level integration within number theory factorization framework and comprehensive empirical validation on simulated and real quantum backends. This places the JVG methodology as a validated architectural modification that improves the scaling behavior of the factoring algorithm, a must have characteristic in the current NISQ era.

The evidence supports a scalability advantage for JVG. Although QFT executes faster at the tested sizes, JVG's rate of increase in runtime, gate counts, memory, and depth is lower, indicating asymptotic efficiency as the qubit number grows. This was observed to be true for both situations of simulated results and implementation on real quantum hardware.

 doi:10.20944/preprints202510.1649.v1

4.1. Implications of Simulated Results for the QNTT-Based Algorithm

We can break down the analysis for Table 5 further and evaluate the increase in performance between the two methodologies under investigation. Table 12 summarizes the reduction in computational resource growth for the JVG methodology as the total amount of qubits increased from N=15 to N=67297 in the simulation backend, as presented in Table 5. Note that Table 12 does not consider the extrapolated values projected for the simulated results.

Table 12. Reduction in resource growth-rate for JVG's QNTT-based Algorithm over Shor's QFT alternative for simulated results.

	Run Time (%)	RAM Usage (%)	CX Gates (%)	U Gates (%)	SWAP Gates (%)	Circuit Depth (%)
QNTT Reduction over QFT	14.67	9.57	30.27	26.24	21.99	33.55

Table 12 showcases demonstrate that the proposed approach scales more efficiently, with a reduction of up to 33.5% in circuit depth relative to the initial configuration when compared to the QFT alternative. The second largest reduction was observed for CX gates, with 30.3%, followed by U gates, which reached a value of 26.2%. The SWAP gates also showed signs of reduction, reaching a value of 26.2%. Finally, run time and memory usage, which achieved 14.7% and 9.6% respectively, were also indicators of lower resource usage as the quantum circuit increased. Overall, the mean reduction in gate count was 22.%, confirming that the JVG methodology exhibits slower growth in computational requirements as circuit size increases.

Nevertheless, these values should be interpreted primarily as indicators of performance trends, since both the JVG and traditional Shor's implementations would require more qubits than it is possible to currently test, as reflected in the projected outcomes shown in Figures 12 and 13. Despite the experimental limitations previously stated, the conclusions derived from the simulated results and their extrapolated projections remain valid, reinforcing the reliability and significance of the proposed methodology.

Faster run time of quantum circuits is essential for the feasibility testing of the algorithm before moving to a real quantum device. Considering the RAM metric, it was observed that its improvement was enough for the QNTT circuit to reach better efficiency in terms of memory use, surpassing the QFT-based model. Again, this insight is of relevant interest given that currently, the simulation of quantum operations in classical devices is heavily constrained in memory usage [66,67]. Thus, more efficient algorithms could offer a superior option to model and simulate more complex circuits.

Considering number factorization using Shor's algorithm, the high amount of gates and the circuit depth make it hard to implement on the current noisy machines [65]. The proposed JVG methodology was shown to have improved scaling for both the gate used and the quantum circuit depth metrics. The implementation of QNTT-based factorization opens the way for a more NISQ-friendly alternative, in which scalability is essential [70]. This was observed by assessing the CX, U, SWAP gates and Circuit Depth reductions, as described in Table 12. Additionally, it offers more efficient simulations on classical backends, which remain an essential step for validating quantum algorithms before moving to a real quantum machine [71,72]. Therefore, the improvements achieved with QNTT provide more tractable simulations of larger problem instances and a more realistic foundation for implementing integer factorization.

 doi:10.20944/preprints202510.1649.v1

4.2. Implications of Experimental Quantum Results for the QNTT-Based Algorithm

From the data in Table 8, Table 13 summarizes the reduction in computational resource growth for the experimental quantum setup as the number of qubits increased from N = 15 to N = 1363. Again, we state that Table 13 does not incorporate the extrapolated values.

Table 13. Reduction in resource growth-rate for JVG's QNTT-based Algorithm over Shor's QFT alternative for experimental results.

	QR (%)	SX Gates (%)	CZ Gates (%)	RZ Gates (%)	X Gates (%)	Circuit Depth (%)
JVG Reduction over Shor	25.99	23.46	24.21	29.72	44.46	31.06

From Table 13, the QR showed a relevant reduction of 26%, indicating that for larger circuits, the proposed methodology requires less processing time and executes more efficiently, being more efficient and faster than its alternative as the number of qubits grows, demonstrating that the JVG algorithm can better manage resource growth when implemented on real quantum devices. This is important for current NISQ machines, since they still lack a larger coherence time, thus preventing long computations [65,70]. In this context, it would be possible to implement factorization algorithms more efficiently for more complex circuits [25,73].

A substantial reduction was also observed considering SX, RZ and X gates. This was especially true for X gates, which were reduced by 44.4%, followed by RZ gates with 29.7% and SX gates, achieving 23.5%. The overall reduction in the number of quantum gates was 30.50%, indicating a positive difference of 8.5% when compared to the simulated outcome. The Circuit Depth also presented reduction by over 31%, being slightly inferior to the one achieved by the simulations. Nevertheless, this outcome remains relevant, since reducing the total of gates in the circuit, consequently resulting in shallower depths, it is possible to reduce circuit depth and ultimately reduce cumulative error propagation [34,74,75].

A reduction of 24.21% over CZ gate count is also of relevance, as controlled gates are also a source of errors related to entanglement fidelity on quantum hardware when compared with single-qubit units [76,77]. By achieving this, it is possible to provide more reliable circuits in terms of both stability and noise resistance, which are desired properties. Combining X gates and CZ gates improvements, it will be possible to reach shallower and error resistant implementations for number factorization algorithms.

The values presented in Table 13, like the ones in Table 12, should be regarded as indicative of performance trends rather than absolute benchmarks, as the experimental setup remains constrained by the limited current NISQ hardware. Likewise, percentage-based improvements on small baselines may exaggerate significance. Thus, this discussion focuses on comparative growth rates and the consistency of CV values to assess true scalability and robustness.

Nevertheless, despite these hardware constraints, the experimental findings reinforce the robustness of the JVG methodology when implemented on real quantum devices, validating its projected trends observed in the experimental setup, further supporting the reliability and practical significance of the proposed approach for in NISQ computing applications.

4.3. Implications of the Projected Values for Simulated and Experimental Configurations

Despite the positive indicators in previous Tables 12 and 13, the outcomes achieved by the proposed algorithm remained slightly below those yielded by the QFT-based methodology.

 doi:10.20944/preprints202510.1649.v1

Projections were developed to visualize this fashion considering both simulation and experimental setups. These plots represent model-based predictions from the exponential fits to the observed data.

For the simulation backend, the extrapolations revealed that the JVG algorithm was able to maintain consistent growth for memory usage, gate counts, and circuit depth when compared to the QFT approach. These metrics highlight the ability of the proposed methodology to manage computational resources more efficiently as the circuit size increases. In particular, the slower gate count and circuit depth growth rate indicates that the QNTT-based algorithm can scale to larger problem instances without a proportional rise in complexity. Among the assessed metrics, only the simulation run time presented negative results for the JVG methodology.

Considering the experimental setup, the projected results indicated the remarkable improved scalability of QNTT-based factorization. The proposed model was superior for all the metrics, including run time, unlike the behavior observed in the simulated results. The ability of the QNTT circuit to maintain exceptional performance when implemented on real hardware conditions, where noise, limited connectivity, and decoherence typically degrade results, indicates that the proposed approach is inherently more resilient than its counterpart. In this sense, the experimental results validate the robustness and adaptability of the methodology, demonstrating that it can leverage the available hardware resources more efficiently.

A similar conclusion is achieved considering the gate count and circuit depth metrics. They revealed substantial reductions in the QNTT implementation compared to the QFT one. The reduced number of two-qubit operations, such as CZ and CX gates, present in the JVG algorithm further minimizes entanglement-related errors. Additionally, the decrease in circuit depth highlights a more compact and optimized structure. These results reinforce that the QNTT-based approach is theoretically more efficient and more suitable for the current hardware. This behavior is crucial for NISQ-era quantum devices, where limited coherence time and gate fidelity impose strict constraints on circuit depth and the number of operations.

These empirical outcomes align with the theoretical projection values (Figures 12 to 16) These projections are entirely data-oriented and derived from the experimentally observed relationships between gate count, circuit depth, and qubit number. They provide an evidence-based estimation of how both methodologies might behave under more complex circuit configurations, and for relevant scales considering the experimental setup. The projected results indicate that the JVG algorithm sustains a markedly slower resource-growth rate across all gate types and circuit-depth metrics when compared to Shor's QFT-based framework. Specifically, they suggest that the JVG architecture preserves its scalability advantage even for RSA-size problems, reinforcing its potential as a more hardware-efficient and noise-resilient solution for future quantum cryptographic implementations.

Altogether, the results revealed that JVG is more hardware-compatible and scales better than Shor's QFT variant. On the sizes we could run QFT is still faster, but trendlines and projections indicate JVG will overtake at larger qubit counts. JVG also appears less prone to error propagation inherent to the current NISQ devices, being of more practical implementation in these machines. In this context, JVG offers both a more efficient framework for validating algorithms on classical backends and a more suitable alternative for implementing integer factorization on current devices, where reducing computational cost is paramount [28,73].

4.4. Statistical Consistency and Implication on NISQ Devices

The statistical analysis for both simulated and experimental benchmarks provides essential insight into the stability and reliability of the obtained results. The consistently low coefficients of variation observed in both simulated and experimental environments reinforce the claim that the JVG architecture exhibits improved robustness to error and predictability. The smaller variability observed for QNTT circuits suggests that their gate structure interacts more favorably with NISQ noise characteristics.

It has been demonstrated empirically that the JVG circuit directly supports the scalability projections reported in Figures 12 to 16, which estimates slower resource growth and enhanced

 doi:10.20944/preprints202510.1649.v1

performance for cryptographically relevant problem sizes. In this sense, the statistical analysis not only validates the repeatability of the experiments but also provides strong evidence that the JVG framework constitutes a more noise-resilient and hardware-efficient approach to quantum integer factorization.

4.5. Impact of QNTT Structure in Shor's Algorithm Pipeline

The use of QNTT proved feasible inside Shor's pipeline. It offered a novel methodology for number factorization by expanding QFT period finding from $\mathbb{C}$ to a finite $\mathbb{Z}_p$ ring. Thus, rather than sample phase estimations with complex roots of unit as in QFT (Figure 6), QNTT retrieves the period for number factorization by using $\omega \mod p$. This ring-based alternative expands number theory for factorization by implementing modular arithmetic. Empirically, this resulted to the JVG circuit's slower growth in depth and gates and its lower variability on hardware, indicating that periodicity of ring structures can be directly leveraged to improve quantum integer factorization on NISQ devices.

While QFT relies on rotational gates (Figures 2 and 3), the JVG is less dependent on these matrices. These gates require long qubit interactions, which could amplify not only errors in NISQ machines, but also offer worse scalability, as reported. Contrarywise, the inclusion of the QNTT structure, as described by Lu and colleagues [58], are NISQ friendly. The QNTT block relies on simpler operations dependent on sequences CX gates (Figures 4 and 5) [60,62]. This architecture allows for a more regular and localized gate topology, reducing errors due to entanglement and other qubit interactions, as revealed by the reported benchmark results and projected values.

Overall, these characteristics make JVG more hardware-efficient and error-resistant under current device limitations. Therefore, the findings validate JVG as a promising and scalable alternative for quantum integer factorization, including RSA-scale security in the NISQ era, as projected.

5. Conclusions

This study leveraged the properties of the number theoretic transform into a quantum framework, combining the QNTT circuit into the number Shor's pipeline to achieve a more efficient alternative for number factorization, while also expanding the number theory for number factorization. To verify the feasibility and performance of the proposed methodology, we investigated several composite numbers across quantum circuits of varying qubit sizes.

In the simulation results, JVG's QNTT-based approach was consistently more scalable regarding run time, memory and gate count. Furthermore, performance gains became more relevant with growth of qubit number, potentially leading improved handling of larger problem sizes. This was observed by improvements of 30.2% for the number of CX gates, 14.7% for the run time, and 9.6% for memory usage, compared with the Shor's algorithm. Experimental validation on a quantum device confirmed these trends. In this phase, the QNTT algorithm also provided superior resource growth in real quantum devices, surpassing the results observed during the simulations. A remarkable improvement of 44.46% on the number of X gates and 26% for the run time was observed by comparing QNTT and QFT models. While Shor's algorithm remains faster for small circuits, JVG's slower resource growth indicates potential superiority at larger circuits, as illustrated by the prediction graphs, showing to be a viable tool for cryptographically relevant applications.

These results are relevant because they provide evidence of a better alternative to QFT-based algorithms, being able to reduce resource overhead and mitigate noise-prone operations by quantum gates, including RSA-scale scenarios. These characteristics are desired in circuits to be implemented in near-term quantum machines.

Future research should investigate improvements in the proposed methodology. For example, the quantum modular exponentiation part could be further improved to remove the dependency of QFT presented here even further. Another possibility is to change the post-processing strategy, which could benefit from the proposed methodology.

 doi:10.20944/preprints202510.1649.v1

Overall, this study establishes JVG's QNTT-based circuit as a promising alternative toward implementing practical quantum integer factorization. The demonstrations of consistent improvements on run time, memory usage and gate counts during both simulated and experimental phases make the proposed methodology a more efficient and error-resilient approach for NISQ devices. These gains were achieved by replacing computationally heavy QFT with a more NISQ-friendly QNTT circuit, making quantum algorithms more tractable in the near-term hardware. Given the importance of number factorization with reduced quantum resources, it directly impacts fields like cryptography. Note that the results from the JVG algorithm is even more capable in the real quantum computer than in the simulated version, when compared against Shor's algorithm. This indicated that JVG is more resilient to noise than Shor. The results presented here demonstrate a more efficient alternative for integer factorization and advance the practical utilization of current quantum computers.

The Jesse–Victor–Gharabaghi (JVG) algorithm presented in this study serves as a proof of concept to validate the proposed hypothesis. With the future advances on quantum hardware, the JVG framework is expected to undergo similar refinements and adaptations to those experienced by Shor's algorithm. Such developments could ultimately lead to more practical and resource-efficient quantum factoring methods, leading to efficient cryptographic applications on quantum devices within the NISQ era.

Author Contributions: Conceptualization, J.V.G.T. and B.G.; methodology, J.V.G.T., V.O.S. and B.G.; software, V.O.S..; validation, J.V.G.T., V.O.S. and B.G.; formal analysis, , J.V.G.T., V.O.S. and B.G.; investigation, , J.V.G.T. and V.O.S.; resources, J.V.G.T. and B.G.; data curation, , J.V.G.T., V.O.S. and B.G.; writing—original draft preparation, , J.V.G.T., V.O.S. and B.G.; writing—review and editing, , J.V.G.T., V.O.S. and B.G.; visualization, V.O.S.; supervision, J.V.G.T. and B.G.; project administration, J.V.G.T. and B.G.; funding acquisition, B.G. and J.V.G.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research study was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) Alliance, grant No. 401643, in association with Lakes Environmental Software Inc.

Data Availability Statement: The original data presented in the study are openly available in GitHub at https://github.com/victor0s/JVG.

Conflicts of Interest: The author Jesse Van Griensven Thé is employed by the company Lakes Environmental. The remaining authors declare that this research was conducted in the absence of any commercial or financial relationships that could be construed as potential conflicts of interest.

References

1. Schuld, M.; Petruccione, F. *Machine Learning with Quantum Computers*; Springer Nature, 2021; ISBN 978-3-030-83098-4.
2. Dam, D.-T.; Tran, T.-H.; Hoang, V.-P.; Pham, C.-K.; Hoang, T.-T. A Survey of Post-Quantum Cryptography: Start of a New Race. *Cryptography* **2023**, *7*, 40. https://doi.org/10.3390/cryptography7030040.
3. McKinsey & Company *Quantum Technology Monitor 2025: From Concept to Reality*; McKinsey & Company, 2025;
4. Kshetri, N. Monetizing Quantum Computing. *IT Prof.* **2024**, *26*, 10–15. https://doi.org/10.1109/MITP.2024.3356111.
5. Companies Market Cap. IonQ Market Capitalization. Available online: https://companiesmarketcap.com/ionq/marketcap/ (accessed on 30 September 2025).
6. Sun, L. Better Quantum Computing Stock: D-Wave Quantum vs. IonQ. Available online: https://www.fool.com/investing/2025/09/24/better-quantum-computing-stock-d-wave-vs-ionq/ (accessed on 30 September 2025).
7. The Motley Fool A Once-in-a-Decade Opportunity: 10 Billion Reasons to Pay Attention to This Monster Quantum Computing Company (Hint: Not IonQ). Available online: https://www.theglobeandmail.com/investing/markets/stocks/JPM-N/pressreleases/34998073/a-once-in-a-

 doi:10.20944/preprints202510.1649.v1

decade-opportunity-10-billion-reasons-to-pay-attention-to-this-monster-quantum-computing-company-hint-not-ionq/ (accessed on 30 September 2025).

8. Bunescu, L.; Vârtei, A.M. Modern Finance through Quantum Computing—A Systematic Literature Review. *PLOS ONE* **2024**, *19*, e0304317. https://doi.org/10.1371/journal.pone.0304317.
9. Zhou, J. Quantum Finance: Exploring the Implications of Quantum Computing on Financial Models. *Comput. Econ.* **2025**. https://doi.org/10.1007/s10614-025-10894-4.
10. Camino, B.; Buckeridge, J.; Warburton, P.A.; Kendon, V.; Woodley, S.M. Quantum Computing and Materials Science: A Practical Guide to Applying Quantum Annealing to the Configurational Analysis of Materials. *J. Appl. Phys.* **2023**, *133*, 221102. https://doi.org/10.1063/5.0151346.
11. Guo, Z.; Li, R.; He, X.; Guo, J.; Ju, S. Harnessing Quantum Power: Revolutionizing Materials Design through Advanced Quantum Computation. *Mater. Genome Eng. Adv.* **2024**, *2*, e73. https://doi.org/10.1002/mgea.73.
12. Weidman, J.D.; Sajjan, M.; Mikolas, C.; Stewart, Z.J.; Pollanen, J.; Kais, S.; Wilson, A.K. Quantum Computing and Chemistry. *Cell Rep. Phys. Sci.* **2024**, *5*, 102105. https://doi.org/10.1016/j.xcrp.2024.102105.
13. Li, W.; Yin, Z.; Li, X.; Ma, D.; Yi, S.; Zhang, Z.; Zou, C.; Bu, K.; Dai, M.; Yue, J.; et al. A Hybrid Quantum Computing Pipeline for Real World Drug Discovery. *Sci. Rep.* **2024**, *14*, 16942. https://doi.org/10.1038/s41598-024-67897-8.
14. Oliveira Santos, V.; Marinho, F.P.; Costa Rocha, P.A.; Thé, J.V.G.; Gharabaghi, B. Application of Quantum Neural Network for Solar Irradiance Forecasting: A Case Study Using the Folsom Dataset, California. *Energies* **2024**, *17*, 3580. https://doi.org/10.3390/en17143580.
15. Oliveira Santos, V.; Costa Rocha, P.A.; Thé, J.V.G.; Gharabaghi, B. Optimizing the Architecture of a Quantum–Classical Hybrid Machine Learning Model for Forecasting Ozone Concentrations: Air Quality Management Tool for Houston, Texas. *Atmosphere* **2025**, *16*, 255. https://doi.org/10.3390/atmos16030255.
16. Fitzgibbon, G.; Ottaviani, C. Constrained Device Performance Benchmarking with the Implementation of Post-Quantum Cryptography. *Cryptography* **2024**, *8*, 21. https://doi.org/10.3390/cryptography8020021.
17. Abbasi, M.; Cardoso, F.; Váz, P.; Silva, J.; Martins, P. A Practical Performance Benchmark of Post-Quantum Cryptography Across Heterogeneous Computing Environments. *Cryptography* **2025**, *9*, 32. https://doi.org/10.3390/cryptography9020032.
18. Shor, P.W. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In Proceedings of the Proceedings 35th Annual Symposium on Foundations of Computer Science; November 1994; pp. 124–134.
19. Coppersmith, D. An Approximate Fourier Transform Useful in Quantum Factoring 2002.
20. Lipton, R.J.; Regan, K.W. *Quantum Algorithms via Linear Algebra: A Primer*; MIT Press, 2014; ISBN 978-0-262-32357-4.
21. Amirkhanova, D.S.; Iavich, M.; Mamyrbayev, O. Lattice-Based Post-Quantum Public Key Encryption Scheme Using ElGamal's Principles. *Cryptography* **2024**, *8*, 31. https://doi.org/10.3390/cryptography8030031.
22. Meter, R.V.; Itoh, K.M. Fast Quantum Modular Exponentiation. *Phys. Rev. A* **2005**, *71*, 052320. https://doi.org/10.1103/PhysRevA.71.052320.
23. Markov, I.L.; Saeedi, M. Constant-Optimized Quantum Circuits for Modular Multiplication and Exponentiation. *Quantum Info Comput* **2012**, *12*, 361–394.
24. Ekerå, M. Modifying Shor's Algorithm to Compute Short Discrete Logarithms 2016.
25. Ekerå, M.; Håstad, J. Quantum Algorithms for Computing Short Discrete Logarithms and Factoring RSA Integers. In Proceedings of the Post-Quantum Cryptography; Lange, T., Takagi, T., Eds.; Springer International Publishing: Cham, 2017; pp. 347–363.

 doi:10.20944/preprints202510.1649.v1

26. Chevignard, C.; Fouque, P.-A.; Schrottenloher, A. Reducing the Number of Qubits in Quantum Factoring 2024.
27. Regev, O. An Efficient Quantum Factoring Algorithm. *J. ACM* **2025**, *72*, 1–13. https://doi.org/10.1145/3708471.
28. Gidney, C. *How to Factor 2048 Bit RSA Integers with Less than a Million Noisy Qubits*; arXiv: 2505.15917, 2025;
29. Gidney, C.; Ekerå, M. How to Factor 2048 Bit RSA Integers in 8 Hours Using 20 Million Noisy Qubits. *Quantum* **2021**, *5*, 433. https://doi.org/10.22331/q-2021-04-15-433.
30. Brigham, E.O. *The Fast Fourier Transform and Its Applications*; Prentice Hall, 1988; ISBN 978-0-13-307505-2.
31. Nussbaumer, H.J. *Fast Fourier Transform and Convolution Algorithms*; Springer Science & Business Media, 2012; ISBN 978-3-642-81897-4.
32. Satriawan, A.; Syafalni, I.; Mareta, R.; Anshori, I.; Shalannanda, W.; Barra, A. Conceptual Review on Number Theoretic Transform and Comprehensive Review on Its Implementations. *IEEE Access* **2023**, *11*, 70288–70316. https://doi.org/10.1109/ACCESS.2023.3294446.
33. Gyongyosi, L.; Imre, S. A Survey on Quantum Computing Technology. *Comput. Sci. Rev.* **2019**, *31*, 51–71. https://doi.org/10.1016/j.cosrev.2018.11.002.
34. Bharti, K.; Cervera-Lierta, A.; Kyaw, T.H.; Haug, T.; Alperin-Lea, S.; Anand, A.; Degroote, M.; Heimonen, H.; Kottmann, J.S.; Menke, T.; et al. Noisy Intermediate-Scale Quantum Algorithms. *Rev. Mod. Phys.* **2022**, *94*, 015004. https://doi.org/10.1103/RevModPhys.94.015004.
35. Kan, S.; Li, Y.; Wang, H.; Mouradian, S.; Mao, Y. Circuit Folding: Modular and Qubit-Level Workload Management in Quantum-Classical Systems 2024.
36. Feynman, R.P. Simulating Physics with Computers. *Int. J. Theor. Phys.* **1982**, *21*, 467–488. https://doi.org/10.1007/BF02650179.
37. Tanenbaum, A.S.; Austin, T. *Structured Computer Organization*; 6th ed.; Pearson: Boston, 2013; ISBN 978-0-13-291652-3.
38. Combarro, E.F.; Gonzalez-Castillo, S.; Meglio, A.D. *A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms*; Packt Publishing Ltd, 2023; ISBN 978-1-80461-830-1.
39. Sutor, R.S. *Dancing with Qubits: How Quantum Computing Works and How It May Change the World*; Expert insight; Packt: Birmingham Mumbai, 2019; ISBN 978-1-83882-736-6.
40. Berberian, S.K. *Introduction to Hilbert Space*; American Mathematical Soc., 1999; ISBN 978-0-8218-1912-8.
41. Halmos, P.R. *Introduction to Hilbert Space and the Theory of Spectral Multiplicity: Second Edition*; Courier Dover Publications, 2017; ISBN 978-0-486-81733-0.
42. Ladd, T.D.; Jelezko, F.; Laflamme, R.; Nakamura, Y.; Monroe, C.; O'Brien, J.L. Quantum Computers. *Nature* **2010**, *464*, 45–53. https://doi.org/10.1038/nature08812.
43. Montanaro, A. Quantum Algorithms: An Overview. *Npj Quantum Inf.* **2016**, *2*, 15023. https://doi.org/10.1038/npjqi.2015.23.
44. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; 10th anniversary ed.; Cambridge University Press: Cambridge ; New York, 2010; ISBN 978-1-107-00217-3.
45. Kaye, P.; Laflamme, R.; Mosca, M. *An Introduction to Quantum Computing*; 1. publ.; Oxford University Press: Oxford, 2007; ISBN 978-0-19-857000-4.
46. Ampatzis, M.; Andronikos, T. Quantum Secret Aggregation Utilizing a Network of Agents. *Cryptography* **2023**, *7*, 5. https://doi.org/10.3390/cryptography7010005.
47. Jozsa, R.; Linden, N. On the Role of Entanglement in Quantum-Computational Speed-Up. *Proc. R. Soc. Lond. Ser. Math. Phys. Eng. Sci.* **2003**. https://doi.org/10.1098/rspa.2002.1097.

48. Shankar, R. *Principles of Quantum Mechanics*; Springer, 1994; ISBN 978-0-306-44790-7.
49. Griffiths, D.J. *Introduction to Quantum Mechanics*; Cambridge University Press, 2017; ISBN 978-1-107-17986-8.
50. Grover, L.K. A Fast Quantum Mechanical Algorithm for Database Search. In Proceedings of the Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96; ACM Press: Philadelphia, Pennsylvania, United States, 1996; pp. 212–219.
51. Brunton, S.L.; Kutz, J.N. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*; Cambridge University Press, 2022; ISBN 978-1-009-09848-9.
52. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms, Third Edition*; MIT Press, 2009; ISBN 978-0-262-03384-8.
53. Cohn, P.M. *Introduction to Ring Theory*; Springer Science & Business Media, 2012; ISBN 978-1-4471-0475-9.
54. Nussbaumer, H.J. *Fast Fourier Transform and Convolution Algorithms*; Springer Series in Information Sciences; Springer Berlin Heidelberg: Berlin, Heidelberg, 1982; Vol. 2; ISBN 978-3-540-11825-1.
55. Satriawan, A.; Mareta, R.; Lee, H. A Complete Beginner Guide to the Number Theoretic Transform (NTT) 2024.
56. Gentleman, W.M.; Sande, G. Fast Fourier Transforms: For Fun and Profit. In Proceedings of the Proceedings of the November 7-10, 1966, fall joint computer conference on XX - AFIPS '66 (Fall); ACM Press: San Francisco, California, 1966; p. 563.
57. Cooley, J.W.; Tukey, J.W. An Algorithm for the Machine Calculation of Complex Fourier Series. *Math. Comput.* **1965**, *19*, 297–301. https://doi.org/10.2307/2003354.
58. Lu, C.; Kundu, S.; Kuruvila, A.; Ravichandran, S.M.; Basu, K. Design and Logic Synthesis of a Scalable, Efficient Quantum Number Theoretic Transform. In Proceedings of the Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design; ACM: Boston MA USA, August 2022; pp. 1–6.
59. Draper, T.G. Addition on a Quantum Computer 2000.
60. Thapliyal, H. Mapping of Subtractor and Adder-Subtractor Circuits on Reversible Quantum Gates. In; 2016; Vol. 9570, pp. 10–34 ISBN 978-3-662-50411-6.
61. Ruiz-Perez, L.; Garcia-Escartin, J.C. Quantum Arithmetic with the Quantum Fourier Transform. *Quantum Inf. Process.* **2017**, *16*, 152. https://doi.org/10.1007/s11128-017-1603-1.
62. Thapliyal, H.; MuÑoz-Coreas, E.; Varun, T.S.S.; Humble, T.S. Quantum Circuit Designs of Integer Division Optimizing T-Count and T-Depth. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 1045–1056. https://doi.org/10.1109/TETC.2019.2910870.
63. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.S.L.; Buell, D.A.; et al. Quantum Supremacy Using a Programmable Superconducting Processor. *Nature* **2019**, *574*, 505–510. https://doi.org/10.1038/s41586-019-1666-5.
64. AbuGhanem, M.; Eleuch, H. NISQ Computers: A Path to Quantum Supremacy. *IEEE Access* **2024**, *12*, 102941–102961. https://doi.org/10.1109/ACCESS.2024.3432330.
65. Preskill, J. Quantum Computing in the NISQ Era and Beyond. *Quantum* **2018**, *2*, 79. https://doi.org/10.22331/q-2018-08-06-79.
66. King, A.D.; Nocera, A.; Rams, M.M.; Dziarmaga, J.; Wiersema, R.; Bernoudy, W.; Raymond, J.; Kaushal, N.; Heinsdorf, N.; Harris, R.; et al. Beyond-Classical Computation in Quantum Simulation. *Science* **2025**, *388*, 199–204. https://doi.org/10.1126/science.ado6285.
67. Chenu, M. Quantum Emulators: CPU, Single GPU and Multiple GPUs Performance Comparison. *Procedia Comput. Sci.* **2025**, *267*, 218–226. https://doi.org/10.1016/j.procs.2025.08.248.

 doi:10.20944/preprints202510.1649.v1

68. Qiskit Shor's Factoring Algorithm. Available online: https://github.com/Qiskit/qiskit/blob/stable/0.18/qiskit/algorithms/factorizers/shor.py (accessed on 29 September 2025).
69. AbuGhanem, M. IBM Quantum Computers: Evolution, Performance, and Future Directions. *J. Supercomput.* **2025**, *81*, 687. https://doi.org/10.1007/s11227-025-07047-7.
70. Aghaee Rad, H.; Ainsworth, T.; Alexander, R.N.; Altieri, B.; Askarani, M.F.; Baby, R.; Banchi, L.; Baragiola, B.Q.; Bourassa, J.E.; Chadwick, R.S.; et al. Scaling and Networking a Modular Photonic Quantum Computer. *Nature* **2025**, *638*, 912–919. https://doi.org/10.1038/s41586-024-08406-9.
71. Jones, T.; Brown, A.; Bush, I.; Benjamin, S.C. QuEST and High Performance Simulation of Quantum Computers. *Sci. Rep.* **2019**, *9*, 10736. https://doi.org/10.1038/s41598-019-47174-9.
72. Guerreschi, G.G. Fast Simulation of Quantum Algorithms Using Circuit Optimization. *Quantum* **2022**, *6*, 706. https://doi.org/10.22331/q-2022-05-03-706.
73. Willsch, D.; Willsch, M.; Jin, F.; De Raedt, H.; Michielsen, K. Large-Scale Simulation of Shor's Quantum Factoring Algorithm. *Mathematics* **2023**, *11*, 4222. https://doi.org/10.3390/math11194222.
74. Temme, K.; Bravyi, S.; Gambetta, J.M. Error Mitigation for Short-Depth Quantum Circuits. *Phys. Rev. Lett.* **2017**, *119*, 180509. https://doi.org/10.1103/PhysRevLett.119.180509.
75. Quek, Y.; Stilck França, D.; Khatri, S.; Meyer, J.J.; Eisert, J. Exponentially Tighter Bounds on Limitations of Quantum Error Mitigation. *Nat. Phys.* **2024**, *20*, 1648–1658. https://doi.org/10.1038/s41567-024-02536-7.
76. Evered, S.J.; Bluvstein, D.; Kalinowski, M.; Ebadi, S.; Manovitz, T.; Zhou, H.; Li, S.H.; Geim, A.A.; Wang, T.T.; Maskara, N.; et al. High-Fidelity Parallel Entangling Gates on a Neutral-Atom Quantum Computer. *Nature* **2023**, *622*, 268–272. https://doi.org/10.1038/s41586-023-06481-y.
77. Huang, J.Y.; Su, R.Y.; Lim, W.H.; Feng, M.; van Straaten, B.; Severin, B.; Gilbert, W.; Dumoulin Stuyck, N.; Tanttu, T.; Serrano, S.; et al. High-Fidelity Spin Qubit Operation and Algorithmic Initialization above 1 K. *Nature* **2024**, *627*, 772–777. https://doi.org/10.1038/s41586-024-07160-2.