Analyzing the Performance of Variational Quantum Factoring on a Superconducting Quantum Processor

Amir H. Karamlou[1,2,]* William A. Simon[1,]* Amara Katabarwa[1],
Travis L. Scholten[3], Borja Peropadre[1], and Yudong Cao[1,†]
[1]*Zapata Computing, Boston, MA 02110 USA*
[2]*Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA and*
[3]*IBM Quantum, IBM T. J. Watson Research Center, Yorktown Heights, NY 10598*

(Dated: December 15, 2020)

Quantum computers hold promise as accelerators onto which some classically-intractable problems may be offloaded, necessitating hybrid quantum-classical workflows. Understanding how these two computing paradigms can work in tandem is critical for identifying where such workflows could provide an advantage over strictly classical ones. In this work, we study such workflows in the context of quantum optimization, using an implementation of the Variational Quantum Factoring (VQF) algorithm as a prototypical example of QAOA-based quantum optimization algorithms. We execute experimental demonstrations using a superconducting quantum processor, and investigate the trade off between quantum resources (number of qubits and circuit depth) and the probability that a given integer is successfully factored. In our experiments, the integers 1,099,551,473,989 and 6,557 are factored with 3 and 5 qubits, respectively, using a QAOA ansatz with up to 8 layers. Our results empirically demonstrate the impact of different noise sources, and reveal a residual ZZ-coupling between qubits as a dominant source of error. Additionally, we are able to identify the optimal number of circuit layers for a given instance to maximize success probability.

I. INTRODUCTION

While near-term quantum devices are approaching the limits of classical tractability [1, 2], they are limited in the number of physical qubits, and can only execute finite-depth circuits with sufficient fidelity to be useful in applications [3]. Hybrid quantum-classical algorithms hold great promise for achieving a meaningful quantum advantage in the near-term [4–6] by combining the quantum resources offered by near-term devices with the computational power of classical processors. In a hybrid algorithm, a classical computer pre-processes a problem instance to expose the core components which capture its essential computational hardness, and is also in a form compatible with a quantum algorithm. The classical computer then utilizes a near-term quantum computer to finish solving the problem. The output of the quantum computer is classical bitstrings which are post-processed classically. This interaction may be iterative: the classical computer may also send communication and control signals to and from the quantum device, for example, by proposing new parameter values to be used in a parameterized quantum circuit.

Quantum algorithms developed for near-term quantum hardware will have to contend with several hardware restrictions, such as number of qubits, coupling connectivity, gate fidelities, and sources of noise, which are all highly dependent on the particular quantum processor the algorithm is being run on. Tradeoffs between different resources have recently been demonstrated [7] in the context of running quantum circuits. However, even as fault-tolerant quantum processors are built, classical computing will still be required to perform pre-processing and post-processing, as well as for quantum error-correction [8]. Therefore, understanding how quantum and classical computing resources can be leveraged together is imperative for extracting maximal utility from quantum computers.

An illustration of this hybrid scheme, in the context of this work, is shown in Fig. 1. In this work, we implement the *variational quantum factoring* (VQF) algorithm [9] on a superconducting quantum processor. VQF is an algorithm for tackling an NP problem with a tunable trade off between classical and quantum computing resources, and is feasible for near-term devices. VQF uses classical pre-processing to reduce integer factoring to a combinatorial optimization problem which can be solved using the quantum approximate optimization algorithm (QAOA) [10].

VQF is a useful algorithm to study for 3 reasons. First, an inverse relationship exists between the amount of classical pre-processing and the number of required qubits needed to finish solving the problem, resulting in a tunable tradeoff between the amount of classical computation and qubits used. Second, the complexity of the quantum circuit acting on those qubits is tunable by adjusting the number of layers used in the QAOA ansatz. Finally, VQF has the advantage that its success is quickly verifiable on a classical computer, by checking if the proposed factors multiply to the input biprime.

This paper is organized as follows. Section II provides an overview of the variational quantum factoring algorithm. Section III describes the experimental results from our implementation. Furthermore, we study the impact of different sources of noise on the algorithm's

* These authors contributed equally.
† yudong@zapatacomputing.com

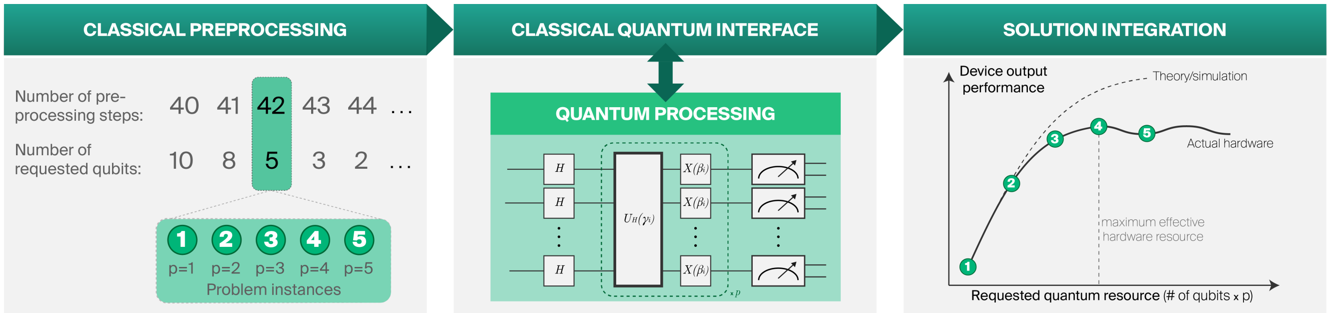


FIG. 1. **Tunable resource tradeoffs with variational quantum factoring (VQF).** Left: Given an integer N to be factored, varying amounts of classical pre-processing steps result in a different number of qubits required for the optimization problem, and defines a "problem Hamiltonian". Middle: The optimization problem is solved on quantum hardware using the QAOA with p layers. Using classical optimization, the algorithm finds parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ that prepare a trial state which approximates the ground state of the problem Hamiltonian. Right: Classical post-processing combines the measurement results on the quantum device with classical pre-processing and evaluates the algorithm success.

performance. The analysis in this section extends beyond VQF and to any general instance of QAOA. We conclude in Section IV, and consider the implications of this work.

II. VARIATIONAL QUANTUM FACTORING

The variational quantum factoring (VQF) algorithm maps the problem of factoring into an optimization problem [11]. Given an n-bit biprime number

$$N = \sum_{k=0}^{n-1} 2^k N_k, \tag{1}$$

factoring involves finding the two prime factors p and q satisfying $N = pq$, where

$$p = \sum_{k=0}^{n_p-1} 2^k p_k, \tag{2}$$

$$q = \sum_{k=0}^{n_q-1} 2^k q_k. \tag{3}$$

That is, p and q can be represented with n_q and n_p bits, respectively.

Factoring in this way can be thought of as the inverse problem to the "longhand" binary multiplication: the value of the i^{th} bit of the result, N_i, is known and the task is to solve for the bits of the prime factors $\{p_i\}$and $\{q_i\}$. An explicit binary multiplication of p and q yields a series of equations that have to be satisfied by $\{p_i\}$and $\{q_i\}$, along with carry bits $\{z_{i,j}\}$ which denote a bit carry from the i^{th} to the j^{th} position. Re-writing each equation in the series allows it to be associated to a particular *clause* in an optimization problem relating to the bit N_i:

$$C_i = N_i - \sum_{j=0}^{i} q_j p_{i-j} - \sum_{j=0}^{i} z_{i,j} + \sum_{j=1}^{n_p+n_q-1} 2^j z_{i,i+j}. \tag{4}$$

In order for each clause, C_i, to be 0, the values of $\{p_i\}$, $\{q_i\}$ and the carry bits $\{z_{i,j}\}$ in the clause must be correct. By satisfying the constraint for all clauses, the factors p and q can be retrieved.

Given this, the problem of factoring is thus reduced to a combinatorial optimization problem. Such problems can be solved using quantum computers by associating a qubit with each bit in the clause. The number of qubits needed depends on the number of bits in the clauses. As discussed in [9], some number of *classical* preprocessing *heuristics* can be used to simplify the clauses (that is, assign values to some of the bits $\{p_i\}$ and $\{q_i\}$). As classical preprocessing removes variables from the optimization problem (by explicitly assigning bit values), the number of qubits needed to complete the solution to the problem is reduced. The new set of clauses will be denoted as $\{C'_i\}$; Appendix B discusses details of this classical preprocessing.

Each clause C'_i can be mapped to a term in an Ising Hamiltonian $\hat{C}_i$ by associating each bit value ($b_k \in \{p_i, q_i, z_{i,j}\}$) to a corresponding qubit operator:

$$b_k \mapsto \frac{1}{2}(1 - Z_k). \tag{5}$$

The solution to the factoring problem corresponds to finding the ground state of the Hamiltonian

$$\hat{H}_{C'} = \sum_{i=0}^{n} \hat{C}_i^2, \tag{6}$$

with a well defined ground state energy $E_0 = 0$. Because each clause C_i in Equation 4 contains quadratic terms in the bits and the Hamiltonian $\hat{H}_{C'}$ is a sum of squares of $\hat{C}_i$, $\hat{H}_{C'}$ includes 4-local terms of the form $Z_i \otimes Z_j \otimes Z_k \otimes Z_l$. (An operator is k-local if it acts non-trivially on at most k qubits.) Much of the literature on quantum optimization so far has considered solving MAXCUT problems on d-regular graphs [10, 12], which can be mapped to problems of finding the ground states of 2-local Hamiltonians. Other problems, such as MAX-3-LIN-2 [13, 14], can be directly mapped to ground state problems of 3-local Hamiltonians. The 4-local Ising Hamiltonian problems produced for VQF motivates a new entry to the classes of Ising Hamiltonian problems currently studied in the context of QAOA. In principle, one can reduce any k-local Hamiltonian to 2-local by well-known techniques [15, 16]. However, the interactions between the qubits in the resulting 2-local Hamiltonian is not guaranteed to correspond to a d-regular graph, which again falls outside the scope of existing considerations in the literature.

The ground state of the Hamiltonian in equation (6) can be approximated on near-term, digital quantum computers using QAOA [10]. QAOA is a computationally universal variational quantum algorithm [17] and good candidate for demonstrating quantum advantage through combinatorial optimization [18]. Each layer of a QAOA ansatz consists of two unitary operators, each with a tunable parameter. The first unitary operator is

$$U_H(\gamma) = e^{-i\gamma \hat{H}_{C'}} \text{ where } \gamma \in [0, 2\pi), \tag{7}$$

which applies a phase according to the cost Hamiltonian $\hat{H}_{C'}$. The second unitary is the admixing operation

$$U_a(\beta) = \prod_i^n e^{-i\beta X_i} \text{ where } \beta \in [0, \pi), \tag{8}$$

which applies a single-qubit rotation around the X-axis with angle 2β.

For a given number of layers p, the combination of $U_H(\gamma)$ and $U_a(\beta)$ are repeated sequentially with different parameters, generating the ansatz state

$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = U_a(\beta_{p-1}) U_H(\gamma_{p-1}) \cdots U_a(\beta_0) U_H(\gamma_0) \, |+\rangle^{\otimes n}, \tag{9}$$

parameterized by $\boldsymbol{\gamma} = (\gamma_0, \cdots, \gamma_{p-1})$ and $\boldsymbol{\beta} = (\beta_0, \cdots, \beta_{p-1})$. The approximate ground state for the cost Hamiltonian can reached by tuning these $2p$ parameters. Given this ansatz state, a classical optimizer is used to find the optimal parameters $\boldsymbol{\gamma}_{\text{opt}}$ and $\boldsymbol{\beta}_{\text{opt}}$ minimizing the expected value of the cost Hamiltonian

$$E(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \boldsymbol{\gamma}, \boldsymbol{\beta} | \, \hat{H}_{C'} \, | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle, \tag{10}$$

where for each $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ the value $E(\boldsymbol{\gamma}, \boldsymbol{\beta})$ is estimated on a quantum computer.

The circuit $|\boldsymbol{\gamma}_{\text{opt}}, \boldsymbol{\beta}_{\text{opt}}\rangle$ is then prepared on the quantum computer and measured. If the outcome of the measurement satisfies all the clauses, it can be mapped to the remaining unsolved binary variables in $\{p_i\}$, $\{q_i\}$, and $\{z_{i,j}\}$, resulting in the prime factors p and q.

The success of VQF is measured by the probability that a measured bitstring encodes the correct factors. Define the set $M_s = \{m_j\}$ consisting of all bitstrings sampled from the quantum computer that satisfy all the clauses in the Hamiltonian, $\hat{H}_{C'}$. We can therefore define the success rate, $s(\boldsymbol{\gamma}, \boldsymbol{\beta})$, as the proportion of the bitstrings sampled from $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$ that satisfy all clauses in $\{C'_i\}$:

$$s(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \frac{|M_s|}{|M|}, \quad M_s = \{m_j \in M | \sum C_i = 0\} \tag{11}$$

where $|M_s|$ is the number of elements in M_s and $|M|$ is the total number of measurements sampled.

In order to successfully factor the input biprime, only one such satisfactory bitstring needs to be observed. However, if that bitstring occurs very rarely, then many repeated preparations and measurements of the trial state are necessary. Therefore, a higher value of $s(\boldsymbol{\gamma}, \boldsymbol{\beta})$ is generally preferred. Note that $E(\boldsymbol{\gamma}, \boldsymbol{\beta}) \geq 0$, with equality if, and only if, $s(\boldsymbol{\gamma}, \boldsymbol{\beta}) = 1$.

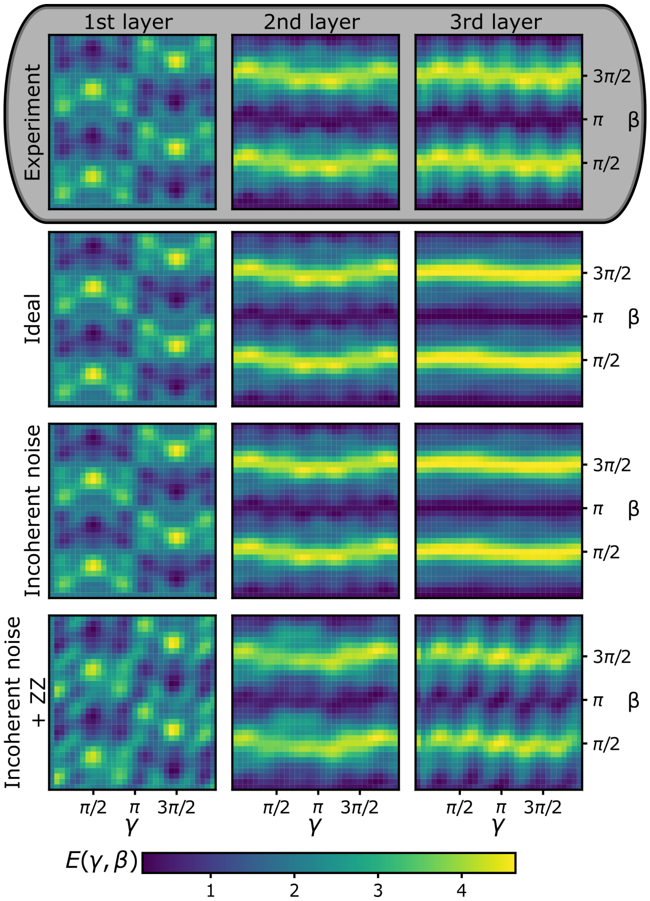


FIG. 2. **QAOA optimization landscapes**. For a fixed VQF instance on 3 qubits, we study the impact of noise (rows) on the optimization landscape as a function of the number of QAOA layers in the ansatz (columns). To produce these landscapes, we use a resolution of $\pi/32$, resulting in 1024 grid points. For the second and third layers, the ansatz parameters for the previous layers are fixed to the optimal parameters found through ideal simulation.

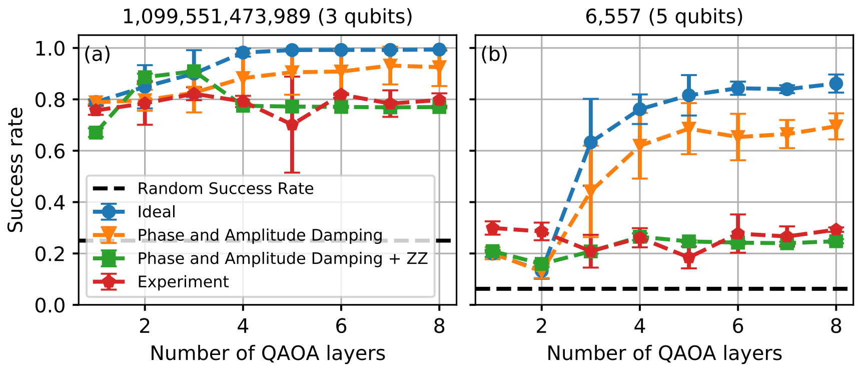


FIG. 3. **Success rates for running VQF on the integers (a) 1099551473989 and (b) 6557 using 3 and 5 qubits, respectively**. We compare the results obtained from experiments (red line) to ideal simulation (blue line), simulation containing phase and amplitude damping noise (orange line), and the residual ZZ-coupling (green line). The amplitude damping rate (T1≈64μs), dephasing rate (T2≈82μs), and residual ZZ-coupling ($\xi/2\pi \approx 245.7$kHz for the 3 qubit instance and $\xi/2\pi \approx 80.1$kHz for the 5 qubit instance) reflect the values measured on the quantum processor. We observe that our experimental performance is limited by the residual ZZ-coupling present in the quantum processor.

III. EXPERIMENTAL ANALYSIS

In this work, we implement the VQF algorithm using QAOA on the *ibmq_boeblingen* superconducting quantum processor (Appendix A) to factor two biprime integers: 1,099,551,473,989 and 6,557 which are classically preprocessed to instances with 3 and 5 qubits respectively. The number of qubits required for each instance varies based the amount of classical pre-processing performed; we use 24 iterations of classical pre-processing for 1,099,551,473,989 and 9 iterations for 6,557.

In order to optimize the QAOA circuit to find $\boldsymbol{\gamma}_{\text{opt}}$ and $\boldsymbol{\beta}_{\text{opt}}$ we employ a layer-by-layer approach [19]. Although there are alternative strategies for training QAOA circuits [20, 21], this approach has been shown to require a grid resolution that scales polynomially with respect to the number of qubits required [9]. This approach has two phases.

In the first phase, for each layer k, we first evaluate a two-dimensional energy landscape by sweeping through different γ_k and β_k values while fixing the parameters for the previous layers (1, 2, ... through $k-1$) to the optimal values found earlier. In our experiments, for 1,099,551,473,989 we evaluate the energy for discrete values of γ_k and β_k ranging from 0 to 2π with a resolution of $\pi/6$, which yields 144 circuit evaluations for each layer. For 6,557, we use a resolution of $2\pi/23$, which yields 529 circuit evaluations for each layer (Fig. 10). We use the degeneracy in the energy landscape caused by evaluating β up to 2π instead of π as a feature to better understand the pattern of the energy landscape. We then select the optimal values for γ_k and β_k from this grid, and combine them with the optimal values from the previous layer. We increment k until it reaches the last layer, in which case the completion of the final round of the above steps marks the end of the first phase of the algorithm.

In the second phase, we use the values $\{(\gamma_k, \beta_k)\}$ obtained from the first phase as the initial point for a gradient-based optimization using the L-BFGS-B method over all $2k$ parameters. In order to measure the gradient of $E(\boldsymbol{\gamma}, \boldsymbol{\beta})$ with respect to the individual variational parameters, we use analytical circuit gradients using the parameter-shift rule [22]. As shown in Fig. 3, we use the optimal parameters found for a circuit with p layers using this method to calculate the algorithm success rate as defined by equation (11).

The performance of our algorithm relies on the optimization of an energy surface spanned by the variational parameters γ and β for each layer. Consequently, by studying this energy surface, we can understand the performance of the optimizer in tuning the parameters of the ansatz state, which in turn impacts the success rate. We do so by visualizing the energy landscape for the k^{th} layer of the ansatz as a function of the two variational parameters, β_k and γ_k, fixing $\boldsymbol{\gamma} = (\gamma_0, .., \gamma_{k-1})$ and $\boldsymbol{\beta} = (\beta_0, .., \beta_{k-1})$ to the optimal values obtained through ideal simulation. Comparing these landscapes between experimental results, ideal simulation, and noisy simulation provides us with a valuable insight into the performance of the algorithm on quantum hardware.

The main sources of incoherent noise present in the quantum processor are qubit relaxation and decoherence. In order to capture the effects of these sources of noise on our quantum circuit in simulation, we apply a relaxation channel with relaxation parameter ϵ_r and a dephasing channel with dephasing parameter ϵ_d after each gate, described by:

$$\mathcal{E}(\rho) = \sum_{m=1}^{3} E_m \rho E_m^{\dagger}, \tag{12}$$

with Kraus operators

$$E_1 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\epsilon_r-\epsilon_d} \end{pmatrix}, \quad E_2 = \begin{pmatrix} 1 & \sqrt{\epsilon_d} \\ 0 & 0 \end{pmatrix}, \quad E_3 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{\epsilon_r} \end{pmatrix}. \tag{13}$$

Furthermore, a dominant source of coherent error is a residual ZZ-coupling between the transmon qubits [23]. This interaction is caused by the coupling between the higher energy levels of the qubits, and is especially pronounced in transmons due to their weak anharmonicity. In our experiments we measure the average residual ZZ-coupling strength for the qubit in use to be $\xi/2\pi = 135 \pm 82$kHz (Appendix A 2). While the additional ZZ rotation caused by this interaction can be compensated by the tunable parameters $\boldsymbol{\gamma}$, this interaction has a severe impact on the two-qubit gates. In our experiments the Ising terms $e^{i\gamma ZZ} = \text{CNOT} \circ I \otimes Z(\gamma) \circ \text{CNOT}$ are realized using a single-qubit Z rotation conjugated by CNOT gates. Each CNOT is comprised of a ZX term generated by the cross-resonance (CR) gate [24] and single-qubit rotations:

$$\text{CNOT} = e^{-i\frac{\pi}{4}}[ZI]^{-1/2}[ZX]^{1/2}[IX]^{-1/2} \tag{14}$$

However, in the presence of residual ZZ-coupling, the axis of rotation of the cross-resonance gate is altered:

$$\text{CR}(t) = e^{i\Gamma ZXt} \mapsto e^{i(\Gamma ZX + \xi ZZ)t} \tag{15}$$

where Γ is the ZX coupling between the qubits as the result of the cross-resonance drive and ξ is the residual ZZ-coupling strength. As a result the two-qubit interactions are fundamentally altered, which leads to a source of error that cannot be corrected for through the variational parameters.

Figure 2 shows the energy landscapes from factoring 1,099,551,473,989 across 3 layers. Comparing the the energy landscape produced in experiment with the ideal simulation and noisy simulations indicates that the energy landscapes produced in experiment is in close agreement with those produced by the ideal simulation and the simulation with incoherent noise for $p = 1$ and $p = 2$. However, for $p = 3$ the energy landscapes produced in experiment have considerable deviations from the ideal simulations due to coherent errors. The presence of incoherent sources of noise do not change the energy landscape pattern, and only reduce the contrast at every layer. On the other hand, the residual ZZ-coupling between qubits alters the pattern of the energy landscape. While at lower depths the impact of this noise source is minor, as we add more layers the effects will accumulate, constructively interfere, and amplify due to the coherent nature of the error.

Figure 3 reports the average success rate of VQF as a function of the number of layers in the ansatz for experiment and simulation. As would be expected, in the ideal case the average success rate approaches the ideal value of 1 with increasing layers of the ansatz for each of the instances studied. Comparing experimental results to those obtained from ideal simulation shows a large discrepancy in the success rate. This discrepancy is not explained by the incoherent noise caused by qubit relaxation and dephasing; the coherent error caused by the residual ZZ-coupling is the dominant effect negatively impacting the success rate. Therefore we conclude that the performance of VQF is limited by the residual ZZ-coupling between the qubits.

We investigate the impact of different noise channels

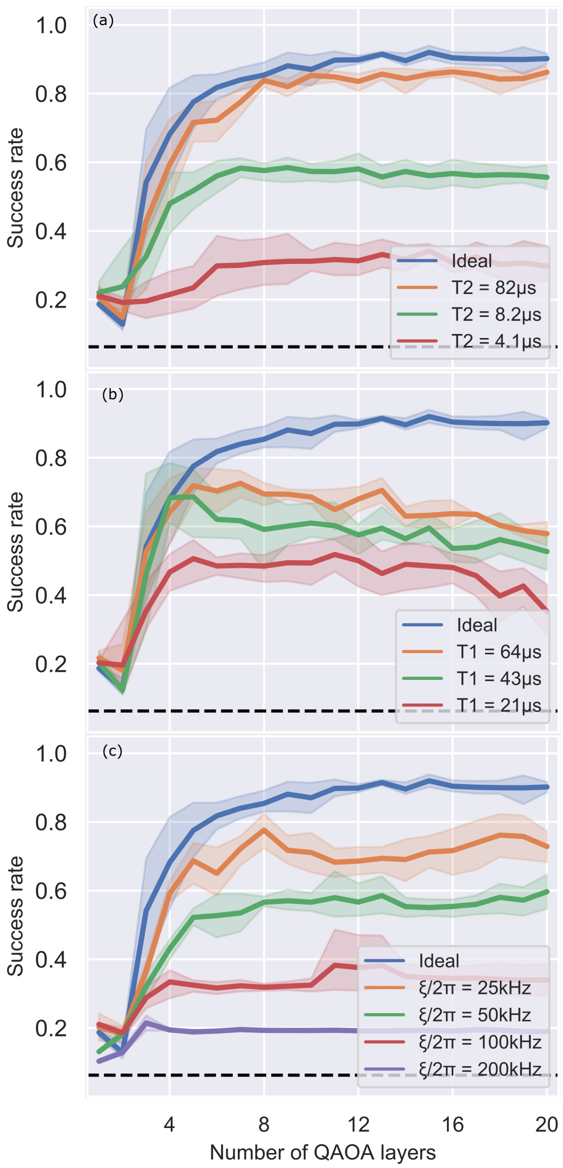


FIG. 4. **Impact of different sources of noise on success rate when factoring 6,557 using 5 qubits:** (a) amplitude damping, (b) phase damping, and (c) residual ZZ-coupling between qubits with a two-qubit gate time of $t_g = 315$ ns.

on the success rate scaling of VQF, with results shown in Figure 4. In the ideal scenario, with every added layer of QAOA, which increases the ansatz expressibility [25], the algorithm success rate increases. However, in the presence of amplitude damping, after an initial increase in the success rate we observe a steady decay in the algorithm's performance. This decay is a result of qubit relaxation back to the $|0\rangle$ state. In contrast, qubit dephasing leads to a plateau in QAOA's performance. This behavior is caused by a loss of quantum coherence at higher layers of circuit which hinders the quantum interference required for the algorithm. Similar to dephasing, the residual ZZ-coupling between qubits also causes a plateau in the success rate of VQF.

While each of these sources of noise impacts the algorithm's performance, the residual ZZ-coupling between the qubits in the quantum processor has the most dominant impact on the success rate. The accumulation of coherent noise from this source significantly alters the ansatz at higher layers, and cannot be corrected using the variational parameters. The results shown in Fig. 4 indicate that the performance of VQF can be significantly improved by engineering ZZ suppression [26, 27]. Since VQF is a non-trivial instance of QAOA we expect this performance analysis to also hold for other quantum algorithms based on QAOA.

IV. CONCLUSION

In this work we analyzed the performance of VQF on a fixed-frequency superconducting quantum processor, and investigated several kinds of classical and quantum resource tradeoffs. We map the problem of factoring to an optimization problem and use variable amounts of classical preprocessing to adjust the number of qubits required. We then use the QAOA ansatz with a variable number of layers (p) to find the solution to the optimization problem. We find that the success rate of the algorithm saturates as p increases, instead of decreasing to the accuracy of random guessing. While more layers increases the expressibility of the ansatz, at higher depths the circuit suffer from the impacts of noise.

Our analysis indicate that the residual ZZ-coupling between the qubits significantly impacts performance. While relaxation and decoherence destroy the quantum coherence at the deeper layers of QAOA, the effect of coherent noise can quickly accumulate and limit performance. By developing a noise model incorporating the coherent sources of noise, we are able to much more accurately predict our experimental results.

There has been various techniques [7, 28] proposed for benchmarking the capability of a quantum device for performing arbitrary unitary operations. However, recent findings on experimental systems [3] suggest that existing benchmarking methods may not be effective for predicting the capability of a quantum device *for specific applications*. This has motivated recent works that focus on benchmarking the performance of a quantum device for applications such as generative modeling [29] and fermionic simulation [30]. Our study of VQF on a superconducting quantum processor has the potential to become another entry in the collection of such "application-based" benchmarks for quantum computers.

ACKNOWLEDGMENTS

The authors would like to thank Eric Anschuetz, Morten Kjaergaard, Sukin Sim, Peter Johnson, and Jonathan Olson for fruitful discussion. We thank Maria Genckel for helping with the graphics. We would also like to thank the IBM Quantum team for providing access to the *ibmq_boeblingen* system, and for help debugging jobs. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM.

[1] F. Arute and et al., Nature — **574**, 505 (2019).

[2] H.-S. Zhong and et al., Science (2020), 10.1126/science.abe8770.

[3] M. Kjaergaard and et al., arXiv:2001.08838v2.

[4] Google AI Quantum and Collaborators, Science (New York, N.Y.) **369**, 1084 (2020).

[5] A. Peruzzo and et. al, Nature Communications **5**, 1 (2014).

[6] V. Havlíček and et al., Nature **567**, 209–212 (2019).

[7] T. Proctor and et al., arXiv (2020), arXiv:2008.11294.

[8] D. Gottesman, (2009), arXiv:0904.2557.

[9] E. R. Anschuetz, J. P. Olson, A. Aspuru-Guzik, and Y. Cao, (2018), arXiv:1808.08927v1.

[10] E. Farhi, J. Goldstone, and S. Gutmann, (2014), arXiv:1411.4028v1.

[11] C. J. C. Burges, *Factoring as Optimization*, Tech. Rep. (2002).

[12] G. G. Guerreschi and A. Y. Matsuura, Scientific Reports **9** (2019), 10.1038/s41598-019-43176-9.

[13] E. Farhi, J. Goldstone, and S. Gutmann, (2014), arXiv:1412.6062.

[14] M. B. Hastings, (2019), arXiv:1905.07047.

[15] J. D. Biamonte, Physical Review A **77** (2008).

[16] N. Dattani, (2019), arXiv:1901.04405.

[17] S. Lloyd, (2018), arXiv:1812.11075.

[18] F. Arute and et al., (2020), arXiv:2004.04197.

[19] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Physical Review X **10** (2018), 10.1103/PhysRevX.10.021067, arXiv:1812.01041.

[20] L. Zhou and et al., Physical Review X **10** (2020), 10.1103/physrevx.10.021067.

[21] J. R. McClean and et al., (2020), arXiv:2008.08615 [quant-ph].

[22] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Physical Review A **99**, 032331 (2019), arXiv:1811.11184.

[23] J. Koch and et al., Phys. Rev. A **76**, 042319 (2007).

[24] J. M. Chow and et. al., Phys. Rev. Lett. **107**, 080502 (2011).

[25] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, Advanced Quantum Technologies **2**, 1900070 (2019).

[26] Y. Sung and et al., (2020), arXiv:2011.01261.

[27] A. Kandala and et al., (2020), arXiv:2011.07050.

[28] A. W. Cross and et al., Phys. Rev. A **100** (2019), 10.1103/physreva.100.032328.

[29] M. Benedetti and et. al, npj Quantum Information **5** (2019), 10.1038/s41534-019-0157-8.

[30] P.-L. Dallaire-Demers and et al., (2020), arXiv:2003.01862.

Appendix A: Experimental Hardware

1. Device Information

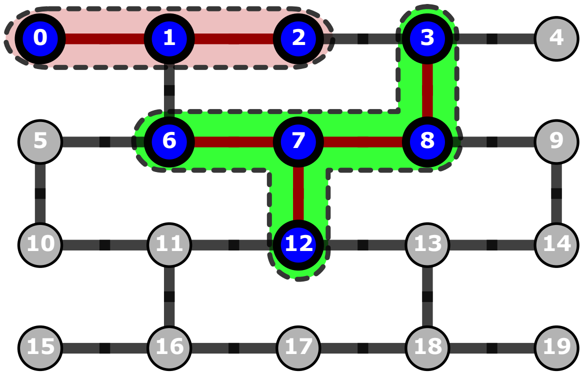
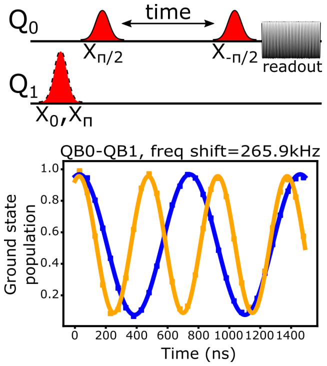


FIG. 5. **IBMQ Boeblingen connectivity map**. For factoring 1099,551,473,989 we use qubits 0, 1, and 2 (shaded in maroon) and for factoring 6,557 we use qubits 3, 6, 7, 8, and 12 (shaded in green).

For our experiments we use the 20 qubit Boeblingen system. This quantum processor consists of 20 fixed-frequency transmon qubits, with microwave-driven single and two-qubit gates, and with a connectivity map as depicted in Fig. 5. For the experiments discussed in this paper use the CNOT as our native two-qubit gate which is driven via the cross-resonance interaction between coupled qubits. Experiments were conducted on the highlighted qubits, for their respective single and two qubit gate fidelities have a higher fidelity, as well as a relatively low readout error. Qubit properties and two-qubit gate characterization can be found in table II and I respectively. For factoring 1099,551,473,989 we use Q0, Q1, and Q2, whereas for factoring 6,557 we use Q3, Q6, Q7, Q8, and Q12.

	Error	Time (ns)
Q0-Q1	7.38e-03 ± 9.7e-04	220
Q1-Q2	6.87e-03 ± 9.9e-04	334
Q6-Q7	13.55e-03 ± 55.8e-04	256
Q7-Q8	10.76e-03 ± 10.4e-04	412
Q7-Q12	14.96e-03 ± 86.1e-04	306
Q8-Q3	10.06e-03 ± 12.3e-04	363

TABLE I. **Two-qubit properties**. Data acquired over the span of 14 days.



Coupling	$\xi/2\pi$ (kHz)
Q0-Q1	275.8 ± 9.0
Q1-Q2	215.7 ± 26.0
Q6-Q7	113.1 ± 6.0
Q7-Q8	72.1 ± 4.6
Q7-Q12	81.3 ± 22.5
Q8-Q3	54.1 ± 3.0

FIG. 6. **Measuring the strength of the residual ZZ-coupling between pairs of qubits.** For each of the pairs of qubits used in the experiments shown in Fig. 3, we measure and report the strength of the residual ZZ-coupling between qubits.

2. Residual ZZ-coupling

In addition to the well-known incoherent processes of relaxation and dephasing, there are coherent noise mechanisms that may affect the overall algorithm performance. Taking into account these coherent errors has become increasingly important in near term algorithms, for their effects cannot be captured by standard benchmark techniques (such as randomized benchmarking), which may impact the overall performance of a quantum algorithm. For superconducting qubits in general, and especially for fixed-frequency transmons, these coherent sources of error are qubit crosstalk due to microwave leaking out from one qubit to another while driving single qubit gates, and residual ZZ-coupling between connected qubits, due to the relatively small anharmonicity of the transmons. Qubit crosstalk is relatively harmless in variational algorithms, for its effect, namely single qubit overrotations, can be learned and cancelled out by the classical optimizer on each optimization step. However, the residual ZZ noise which accumulates during the gates cannot be learned by the optimizer, as it doesn't commute with the native ZX term from the CR gate, and needs to be taken into account in the quantum circuit.

In order to estimate the effect of the residual ZZ-coupling in actual experiments, it is crucial to derive an accurate Hamiltonian model that captures the effect of higher energy levels in the energy spectrum. To this end, we sketch the derivation of an effective Hamiltonian for two transmon qubits interacting through a common transmission line resonator (which corresponds to qubits connected through solid lines in Fig. 5). We model the transmon qubit j as a Duffing oscillator

$$H_{\mathrm{qb}_i} = \omega_0(b_i^\dagger b_i + 1/2) + \frac{\delta_i}{2} b_i^\dagger b_i (b_i^\dagger b_i - 1), \qquad \text{(A1)}$$

that interacts with other transmons via the transmission line resonator through the exchange of virtual excitations

$$H = \omega_{\mathrm{res}} a^\dagger a + H_{\mathrm{qb}_1} + H_{\mathrm{qb}_2} + \sum_{j=1}^{2} g_j (a^\dagger b_j + H.c), \quad \text{(A2)}$$

Property		Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q12
1QB Gate Error	mean	4.28e-04	3.08e-04	2.83e-04	4.03e-04	7.07e-04	5.19e-04	7.37e-04	3.16e-04	4.20e-04	4.08e-04	3.47e-04
	std	1.56e-04	5.09e-05	7.98e-05	6.61e-05	4.13e-04	1.05e-04	5.230e-04	4.80e-05	8.72e-05	6.20e-05	3.54e-05
Frequency (GHz)	mean	5.05	4.85	4.70	4.77	4.37	4.91	4.73	4.55	4.66	4.77	4.74e
	std	2.42e-06	4.35e-06	3.18e-06	3.73e-06	1.01e-05	9.48e-05	1.49e-05	4.26e-06	4.84e-06	6.37e-05	3.26e-06
Readout Error (%)	mean	2.48	2.73	3.68	2.28	5.13	1.95	4.69	3.80	5.76	6.34	4.59
	std	.523	.348	.964	.373	4.27	.711	1.54	.805	.352	2.43	.465
T1 (μs)	mean	62.8	59.1	105	80.2	87.3	80.0	64.2	81.0	44.8	57.5	80.5
	std	21.3	11.5	21.7	8.98	17.4	19.5	15.0	14.0	12.4	5.61	17.7
T2 (μs)	mean	97.4	86.5	104	42.7	76.9	54.2	75.0	88.9	69.6	84.4	107
	std	38.6	21.9	31.2	4.38	32.3	16.3	20.7	17.7	18.1	40.0	26.1

TABLE II. **Single-qubit characterization**. Data acquired over the span of 14 days.

where g_j represents the coupling constant between transmon j and the resonator. We can diagonalize this Hamiltonian through a Schrieffer-Wolff transformation to adiabatically remove the resonator effect, getting an effective transmon-transmon interaction that reads

$$H = \sum_{j=1}^{2} \omega_j b_j^\dagger b_j + J(b_1^\dagger b_2 + b_2^\dagger b_1), \tag{A3}$$

where $J = -g_1 g_2 \times (\Delta_1 + \Delta_2)/2(\Delta_1\Delta_2)$, is the effective coupling between transmons, and $\Delta_j = \omega_{\text{res}} - \omega_j$ the detuning between the transmon j and resonator. We now bring the above Hamiltonian to its diagonal form, as it represents the physical basis where the qubits are actually measured, and project it to the two-qubit subspace, yielding

$$H = \sum_{j=1}^{2} \tilde{\omega}_j Z_j + \xi Z_1 Z_2, \tag{A4}$$

where $\tilde{\omega}_j$ are (dressed) qubit frequencies that are actually measured in an experiment, and $\xi = 2J^2(\delta_1 + \delta_2/(\delta_1 - \Delta)(\delta_2 + \Delta))$. The residual ZZ-coupling between two qubits, Q_i and Q_j, can be experimentally measured by finding the difference in the Ramsey oscillation frequency of Q_i while Q_j is in the $|0\rangle$ and $|1\rangle$ state. The residual ZZ-coupling between the qubits used in our experiments can be found in Fig. 6.

Appendix B: VQF Preprocessing

By simplifying the clauses using binary rules we reduce the quantum resources required for executing the VQF algorithm. The classical preprocessing procedure iterates through all clauses $\{C_i\}$ a constant number of times, using the a set of binary rules to solve or make deductions where possible. Assuming $x, y, z \in F_2, a \in Z^+$ we apply the following classical preprocessing rules:

1. $xy = 1 \Rightarrow x = y = 1$
2. $x + y = 1 \Rightarrow xy = 0$
3. $x + y = 2z \Rightarrow x = y = z$
4. $\sum_{i=1}^{a} x_i = a \Rightarrow x_i = 1$
5. if $x_i = 1$ violates $\max(lhs) = \max(rhs) \Rightarrow x_i = 0$
6. if $\min(lhs) < \min(rhs)$ and x is the only variable on lhs $\Rightarrow x = 1$
7. the parity of the lhs must be the same as the rhs

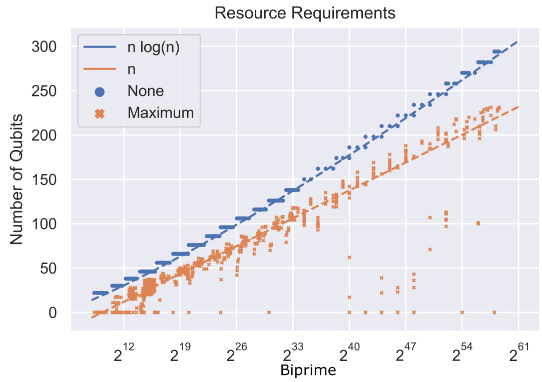


FIG. 7. **Empirical results on number of qubits required for factoring a biprime N after classical preprocessing**

The runtime of the classical preprocessor is $O(n^2)$ [9]. Without preprocessing the VQF qubit requirements scale as $O(n^2)$, whereas this resource bound empirically gets reduced to $O(n\log n)$ using the classical preprocessor (Fig. 7). We additionally note that for certain biprimes, the preprocessor is able to significantly reduce the number of required qubits, in some cases completely solving the clauses. Additionally, in Figure 8, we show the scaling of 1-local, 2-local, 3-local, and 4-local terms in the resulting hamiltonians, after classical preprocessing, for factoring a given biprime N.

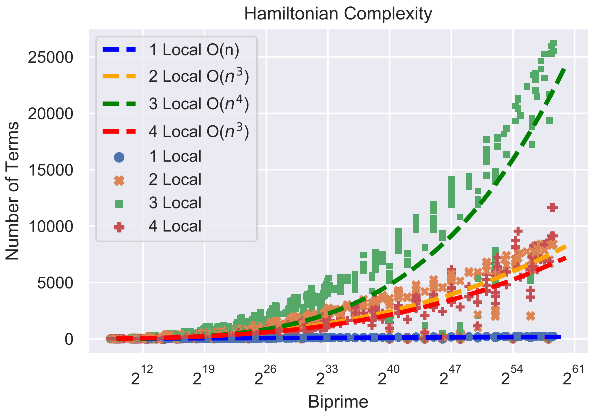


FIG. 8. **Empirical results on number of n-local terms in the hamiltonain for factoring a biprime N after classical preprocessing**

Appendix C: Circuit Metrics

	CNOT gates	Single-qubit gates	Circuit depth
1st layer	4	22	13
2nd layer	8	41	25
3rd layer	12	60	37
4th layer	16	79	49
5th layer	20	98	61
6th layer	24	117	73
7th layer	28	136	85
8th layer	32	155	97

TABLE III. **Properties of QAOA circuits 1,099,551,473,989 (3 qubits)**. The qubit mapping used for this instance is: [0, 1, 2].

	CNOT gates	Single-qubit gates	Circuit depth
1st layer	13	44	25
2nd layer	29	86	52
3rd layer	45	128	76
4th layer	94	203	116
5th layer	95	230	132
6th layer	147	308	176
7th layer	147	334	197
8th layer	162	375	226

TABLE IV. **Properties of QAOA circuits on 6,557 (5 qubits)**. The qubit mapping used for this instance is: [6, 7, 12, 8, 3].

In Tables III and IV, we show the relevant properties for the circuits used in both simulation and experiment for each of the problems studied. We use the Qiskit transpiler, with the optimization level set to 1, to create the circuits with the correct gate set for IBMQ devices. Due to the non-deterministic nature of this transpiler, we independently run the transpilation process 10 times, using the circuit with the least number of cx gates.

Appendix D: 297491 (4 qubits)

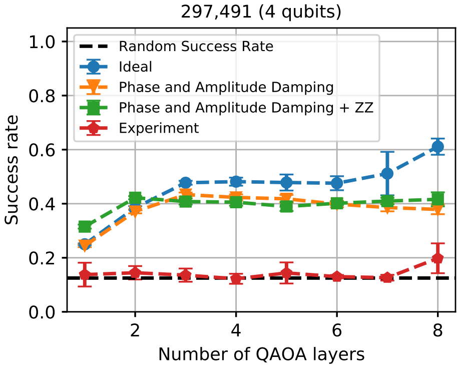


FIG. 9. **Success rates for running VQF on the integer 297,491 using 4 qubits**. The qubits used for this instance are 6, 7, 8, and 12. We compare the results obtained from experiments (red line) to ideal simulation (blue line), simulation containing phase and amplitude damping noise (orange line), and the residual ZZ-coupling (green line). The amplitude damping rate (T1≈64μs), dephasing rate (T2≈82μs), and residual ZZ-coupling ($\xi/2\pi \approx$ 189kHz) reflect the values measured on the quantum processor.

In Fig. 9, we show the results of running the VQF algorithm on the integer 297,491 using 4 qubits (6, 7, 8, and 12). Similar to the results shown in Fig. 3, we present the success rate of the VQF algorithm as a function of the number of layers in our QAOA ansatz and compare results from experiment with those from simulation with different noise models. In contrast to results shown in Fig. 3, we note that the residual ZZ-coupling between qubits does not alone explain the degradation in the success of the algorithm when run on the quantum processor. This suggests the presence of an additional source of noise that is unaccounted for, eventhough the qubits used in this instance are a subset of the ones used to factor 6,557. It is clear that this additional source of noise is impacting the success of the VQF algorithm, which underscores the need to examine realistic applications on near-term devices.

Appendix E: Higher depth energy landscapes

In Fig. 10 we display the energy landscapes for the instances representing 1,099,551,473,989 and 6,557 using 3 and 5 qubits respectively for $p = (1, 2, \cdots, 8)$. We observe that the landscapes for $p = (2, \cdots, 8)$ follow a

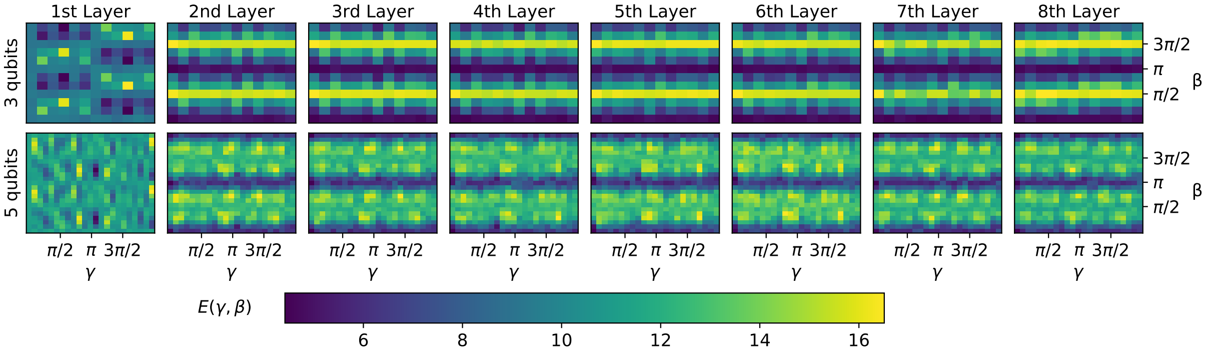


FIG. 10. **QAOA optimization landscapes**. For each VQF instance (columns), we show the resulting optimization landscape as a function of the number of QAOA layers in the ansatz (rows). For the 3 qubit instance (leftmost column), the landscapes are obtained with a 12x12 grid, resulting in 144 uniformly spaced points. Whereas the landscapes obtained for the 5 qubit instance is created using a 23x23 grid with 529 uniformly spaced points.

similar pattern, not only between layers, but across instances as well. Additionally, we note that we continue to observe structure in the landscapes for high-depth circuits. As reported in Table IV, the landscape produced for 6,557 with 8 layers requires approximately 162 CNOT gates, 375 single-qubit gates, and circuit depth of 226.