

Variational Quantum Factoring

Eric R. Anschuetz,^{*} Jonathan P. Olson,[†] Alán Aspuru-Guzik,[‡] and Yudong Cao[§]

Zapata Computing Inc., 501 Massachusetts Avenue, Cambridge MA 02138

Abstract

Integer factorization has been one of the cornerstone applications of the field of quantum computing since the discovery of an efficient algorithm for factoring by Peter Shor. Unfortunately, factoring via Shor’s algorithm is well beyond the capabilities of today’s noisy intermediate-scale quantum (NISQ) devices. In this work, we revisit the problem of factoring, developing an alternative to Shor’s algorithm, which employs established techniques to map the factoring problem to the ground state of an Ising Hamiltonian. The proposed variational quantum factoring (VQF) algorithm starts by simplifying equations over Boolean variables in a preprocessing step to reduce the number of qubits needed for the Hamiltonian. Then, it seeks an approximate ground state of the resulting Ising Hamiltonian by training variational circuits using the quantum approximate optimization algorithm (QAOA). We benchmark the VQF algorithm on various instances of factoring and present numerical results on its performance.

^{*} eric.anschuetz@zapatacomputing.com

[†] jonny@zapatacomputing.com

[‡] aspuru@zapatacomputing.com

[§] yudong@zapatacomputing.com

I. INTRODUCTION

Integer factorization is one of the first practically relevant problems that can be solved exponentially faster on a quantum computer than any currently known methods for classical computation by employing Shor’s factoring algorithm [1]. Since its initial appearance, numerous follow-up studies have been carried out to optimize the implementation of Shor’s algorithm from both algorithmic and experimental perspectives [2–11]. Improved constructions [9, 12, 13] have been proposed which, for an input number of n bits, improve the circuit size from $3n$ qubits [14] to $2n+3$ [9] and $2n+2$ [12] qubits, and with nearest-neighbor interaction constraints [15]. It has also been pointed out that using iterative phase estimation [16], one can further reduce the qubit cost to $n+1$, though the circuit needs to be adaptive in this case [2, 4]. Various other implementations [17, 18] of Shor’s algorithm have been proposed such that only a subset of qubits need to be initialized in a computational basis state (“clean qubits”).

Concrete resource estimates in realizing Shor’s algorithm for factoring relevant numbers for RSA have also been performed for specific architectures [19–22]. For example, on one particular architecture of a fault-tolerant quantum computer [20, 21] it is estimated that factoring a 2048-bit RSA number requires a circuit depth on the order of 10^9 , requiring roughly 10 days on a quantum computer comprised of 10^5 logical qubits [20, cf. Figure 15]. Another resource estimate [23] considering a photonic architecture suggests that factoring a 1024-bit RSA number would take 2.3 years with 1.9 billion photonic modules. In contrast, present technologies are in the era of noisy intermediate-scale quantum (NISQ) devices [24], where quantum devices typically have on the order of 10^2 - 10^3 noisy qubits that can only reliably implement circuits of limited depth. This renders the practical impact of Shor’s algorithm (as well as alternative algorithms for quantum factoring that use subroutines requiring fault tolerance, such as [11, 25]) a reality at least as distant as the realization of fault-tolerant quantum computers.

Another approach to factoring on a quantum computer exploits the mapping from factoring to the ground state problem of an Ising Hamiltonian [26]. The basic idea underlying the mapping is to simply use the fact that factoring is the inverse operation of multiplication. Therefore, by working through the multiplication of two undetermined n -bit numbers and fixing the output to be the number being factored, one can write a set of equations involving

the bits of the factors and the carry bits. The Hamiltonian is constructed such that the ground state satisfies all of the generated equations and any bit assignment which violates any of the equations receives an energy penalty. Interesting observations [8, 27, 28] have been made about specific instances of factoring which allow one to simplify the equations tremendously. On the experimental side, most of the current efforts focus on analog approaches such as quantum annealing [29, 30] and simulated adiabatic evolution [28, 31]. However, the same ground state problem of Ising Hamiltonians can be approximately solved on gate model NISQ devices using the quantum approximate optimization algorithm (QAOA) [32].

Here we introduce an approach which we call *variational quantum factoring* (VQF). As with other hybrid classical/quantum algorithms such as the variational quantum eigensolver (VQE) [33] or the quantum autoencoder (QAE) [34], classical preprocessing coupled with quantum state preparation and measurement are used to optimize a cost function. In particular, we employ the QAOA algorithm [32] and classical preprocessing for factoring. The VQF scheme has two main components: first, we map the factoring problem to an Ising Hamiltonian, using an automated program to find reduction in the number of required qubits whenever appropriate. Then, we train the QAOA ansatz for the Hamiltonian using a combination of local and global optimization. We explore six instances of the factoring problem (namely, the factorings of 35, 77, 1207, 33667, 56153, and 291311) to demonstrate the effectiveness of our scheme in certain regimes as well as its robustness with respect to noise.

The remainder of the paper is organized as follows: Section II describes the mapping from a factoring problem to an Ising Hamiltonian, together with the simplification scheme that is used for reducing the number of qubits needed. Section III introduces QAOA and describes our method for training the ansatz. Section IV presents our numerical results. We conclude in Section V with further discussion on future works.

II. ENCODING FACTORING INTO AN ISING HAMILTONIAN

A. Factoring as binary optimization

It is known from previous work that factoring can be cast as the minimization of a cost function [26], which can then be encoded into the ground state of an Ising Hamiltonian [27,

35, 36]. To see this, consider the factoring of $m = p \cdot q$, each having binary representations

$$\begin{aligned} m &= \sum_{k=0}^{n_m-1} 2^k m_k, \\ p &= \sum_{k=0}^{n_p-1} 2^k p_k, \\ q &= \sum_{k=0}^{n_q-1} 2^k q_k, \end{aligned} \tag{1}$$

where $m_k \in \{0, 1\}$ is the k th bit of m , n_m is the number of bits of m , and similarly for p and q . When n_p and n_q are unknown (as they are unknown *a priori* when only given a number m to factor), one may assume without loss of generality [26] that $p \geq q$, $n_p = n_m$, and $n_q = \lceil \frac{n_m}{2} \rceil$ [37]. By carrying out binary multiplication, the bits representing m , p , and q must satisfy the following set of $n_c = n_p + n_q - 1 \in O(n_m)$ equations [26, 35, 36]:

$$0 = \sum_{j=0}^i q_j p_{i-j} + \sum_{j=0}^i z_{j,i} - m_i - \sum_{j=1}^{n_c} 2^j z_{i,i+j} \tag{2}$$

for all $0 \leq i < n_c$, where $z_{i,j} \in \{0, 1\}$ represents the carry bit from bit position i into bit position j . If we associate a clause C_i over \mathbb{Z} with each equation such that

$$C_i = \sum_{j=0}^i q_j p_{i-j} + \sum_{j=0}^i z_{j,i} - m_i - \sum_{j=1}^{n_c} 2^j z_{i,i+j}, \tag{3}$$

then factoring can be represented as finding the assignment of binary variables $\{p_i\}$, $\{q_i\}$, and $\{z_{ij}\}$ which solves

$$0 = \sum_{i=0}^{n_c} C_i^2. \tag{4}$$

In general, if m contains more than two prime factors, Equation 2 still holds and our method will produce a Hamiltonian with a ground state manifold degenerate over all pairs of factors of m . To further factor m , one can repeat the VQF scheme to possibly yield a different (p, q) pair, recursively apply our scheme to each of p and q , or expand m into a product of multiple factors to arise at an analogous form of Equation (2) that can be simultaneously solved for all factors of m . If m is prime itself, then its primality can be easily detected [38]. Therefore, for the rest of our discussion we will consider m to be the product of two primes (a *biprime*), without loss of generality.

B. Simplifying the clauses

One method for simplifying clauses is to directly solve for a subset of the binary variables that are easy to solve for classically [27, 36]. This reduction iterates through all clauses C_i as given by Equation (3) a constant number of times. In the following discussion, let $x, y, x_i \in \mathbb{F}_2$ be unknown binary variables and $a, b \in \mathbb{Z}^+$ positive constants. Along with some trivial relations, we apply the classical preprocessing rules [39]:

$$\begin{aligned}
xy - 1 = 0 &\implies x = y = 1, \\
x + y - 1 = 0 &\implies xy = 0, \\
a - bx = 0 &\implies x = 1, \\
\sum_i x_i = 0 &\implies x_i = 0, \\
\sum_{i=1}^a x_i - a = 0 &\implies x_i = 1.
\end{aligned} \tag{5}$$

We also are able to truncate the summation of the final term in Equation (3). This is done by noting that if 2^j is larger than the maximum attainable value of the sum of the other terms, $z_{i,i+j}$ cannot be one; otherwise, the subtrahend would be larger than the minuend for all possible assignments of the other variables, and Equation (2) would never be satisfied. This effectively limits the magnitude of Equation (3) to be $O(n_m)$.

This classical preprocessing iterates through each of $O(n_c)$ terms in each of $n_c \in O(n_m)$ clauses C_j (see Equations 2 and 3), yielding a classical computer runtime of $O(n_m^2)$. This is because $O(n_c) = O(n_m)$ from the identity $n_c = n_p + n_q - 1$, and $n_p \leq n_m$ and $n_q \leq \lceil \frac{n_m}{2} \rceil$ [26]. In practice, for most instances we have observed that the preprocessing program greatly reduces the number of (qu)bits needed for solving the problem, as is shown in Figure 1.

C. Constructing the Ising Hamiltonian

For each i from 0 through $n_c - 1$, let C'_i be C_i after applying the classical preprocessing procedure outlined in Section II B. The solutions for the simplified equations $C'_i = 0$ then correspond to the minimization of the classical energy function

$$E = \sum_{i=0}^{n_c} C'^2_i, \tag{6}$$

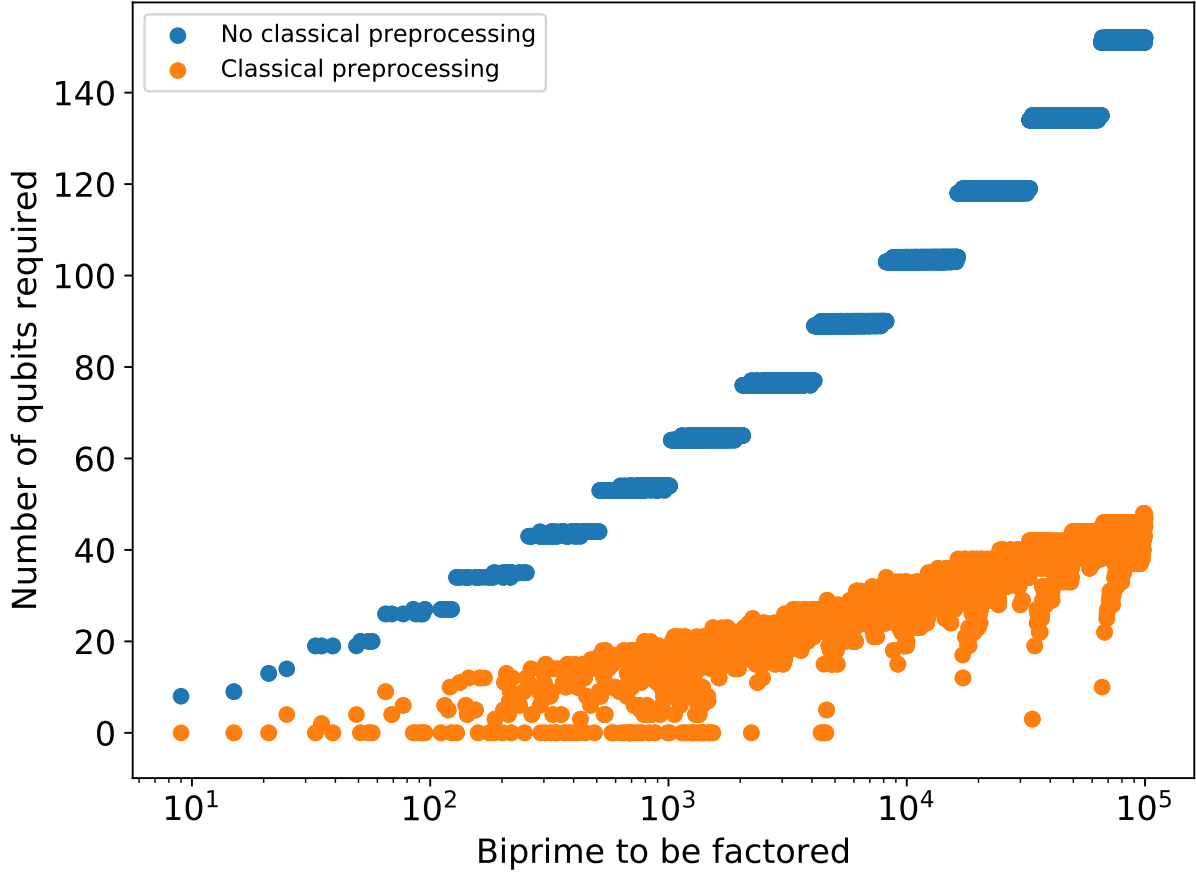


FIG. 1. This figure empirically demonstrates the reduction in qubit requirements after implementing the classical preprocessing procedure outlined in Section II B. After the classical preprocessing algorithm (orange), the number of qubits necessary for our algorithm empirically scales approximately as $O(n_m)$. In contrast, with no simplification (blue), VQF’s qubit requirements scale as $O(n_m \log(n_m))$ asymptotically [26].

which has a natural quantum representation as a *factoring Hamiltonian*

$$H = \sum_{i=0}^{n_c} \hat{C}_i^2. \quad (7)$$

Each \hat{C}_i term is obtained by quantizing p_i , q_i , and $z_{j,i}$ in the clause C'_i using the mapping

$$b_k \rightarrow \frac{1}{2} (1 - \sigma_{b,k}^z), \quad (8)$$

where $b \in \{p, q, z\}$ and k is its associated bit index. We have thus encoded an instance of factoring into the ground state of a 4-local Ising Hamiltonian. H can also be represented in

quadratic form by substituting each product $q_j p_{i-j}$ with a new binary variable $w_{i,j}$ and introducing additional constraints to the Hamiltonian [35]. This is necessary for implementation on quantum annealing devices with restricted pairwise coupling between qubits. However, in our case it is not necessary since in the gate model of quantum computation methods for time evolution under k -local Hamiltonian are well known [40].

III. VARIATIONAL QUANTUM FACTORING ALGORITHM

The main component of our scheme is an approximate quantum ground state solver for the Hamiltonian in Equation (7) as a means to approximately factor numbers on near-term gate model quantum computers. We use the *quantum approximate optimization algorithm* (QAOA), which is a hybrid classical/quantum algorithm for near-term quantum computers that approximately solves classical optimization problems [32]. The goal of the algorithm is to satisfy (i.e. find the simultaneous zeros of) the simplified clauses C'_i , which we cast as the minimization of a classical cost Hamiltonian H_c , and set to be identical to the Hamiltonian in Equation (7) (i.e. $H_c = H$).

To prepare the (approximate) ground state we use an ansatz state

$$|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle = \prod_{i=1}^s (\exp(-i\beta_i H_a) \exp(-i\gamma_i H_c)) |+\rangle^{\otimes n}, \quad (9)$$

parametrized by angles $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ over n qubits, where s is the number of layers of the QAOA algorithm. Here, H_a is the *admixing Hamiltonian*

$$H_a = \sum_{i=1}^n \sigma_i^x. \quad (10)$$

For a fixed s , QAOA uses a classical optimizer to minimize the cost function

$$M(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \langle \boldsymbol{\beta}, \boldsymbol{\gamma} | H_c | \boldsymbol{\beta}, \boldsymbol{\gamma} \rangle. \quad (11)$$

For $s \rightarrow \infty$, $M(\boldsymbol{\beta}, \boldsymbol{\gamma})$ is minimized when the fidelity between $|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle$ and the true ground state tends to 1. Generically for $s < \infty$, $|\arg \min(M(\boldsymbol{\beta}, \boldsymbol{\gamma}))\rangle$ may have exponentially small overlap with the true ground state. In our case, numerical evidence which will be discussed in Section IV suggests that often letting $s \in O(n)$ suffices for large overlap with the ground state.

Input number m	Number of qubits n	Number of carry bits	$p \leftrightarrow q$ symmetry	Grid size
$35 = 5 \times 7$	2	0	✓	6×6
$77 = 7 \times 11$	6	3	✗	24×24
$1207 = 17 \times 71$	8	5	✗	36×36
$33667 = 131 \times 257$	3	1	✗	9×9
$56153 = 233 \times 241$	4	0	✓	12×12
$291311 = 523 \times 557$	6	0	✓	24×24

TABLE I. First column: Biprime numbers used in this study. Second column: the total number of qubits needed to perform VQF on the problem instance. Third column: among the qubits, the number of carry bits produced in the Ising Hamiltonian after simplifying the Boolean equations with rules described in (5). The observed difference between instances with carry bits versus without carry bits is shown in Figure 2, along with Figures 6 and 5. Fourth column: in the energy function (6), whether or not there exists a $p \leftrightarrow q$ symmetry. Such symmetry can be broken by two factors having different bit lengths. Fifth column: size of the grid used for the layer-by-layer brute-force search.

To optimize the QAOA parameters β and γ , we employed a layer-by-layer iterative brute-force grid search over each pair (γ_i, β_i) , with the output fed into a BFGS global optimization algorithm [41]. The choices for grid sizes were motivated by a gradient bound given in [32]; more precisely, we expect each dimension of the grid should be $O(n_c^2 n^4)$. From [32] a bound of $O(m^2 + mn)$ is given for QAOA minimizing an objective function of m clauses on n variables. The setting in [32] is that each clause gives rise to a term in the Hamiltonian that has a norm at most 1. In our case, each clause C'_i instead gives rise to a term in the Hamiltonian that has norm $|\hat{C}_i^2| = O(n^2)$. Therefore, we take $m = n_c n^2$ and n be the number of qubits, yielding a bound $O(n_c^2 n^4)$ for the gradient. To ensure that the optimum found by grid search differs from the true optimum by a constant, we therefore introduce a grid of size $O(n_c^2 n^4) \times O(n_c^2 n^4)$ based on the gradient bound [42]. This ensures a polynomial scaling of the grid resolution. Numerically, training on coarser grids seemed sufficient (see Table I).

The remaining cost for finding the solution then comes from the global optimization procedure. In our numerical studies, the complexity scaling of performing BFGS optimization

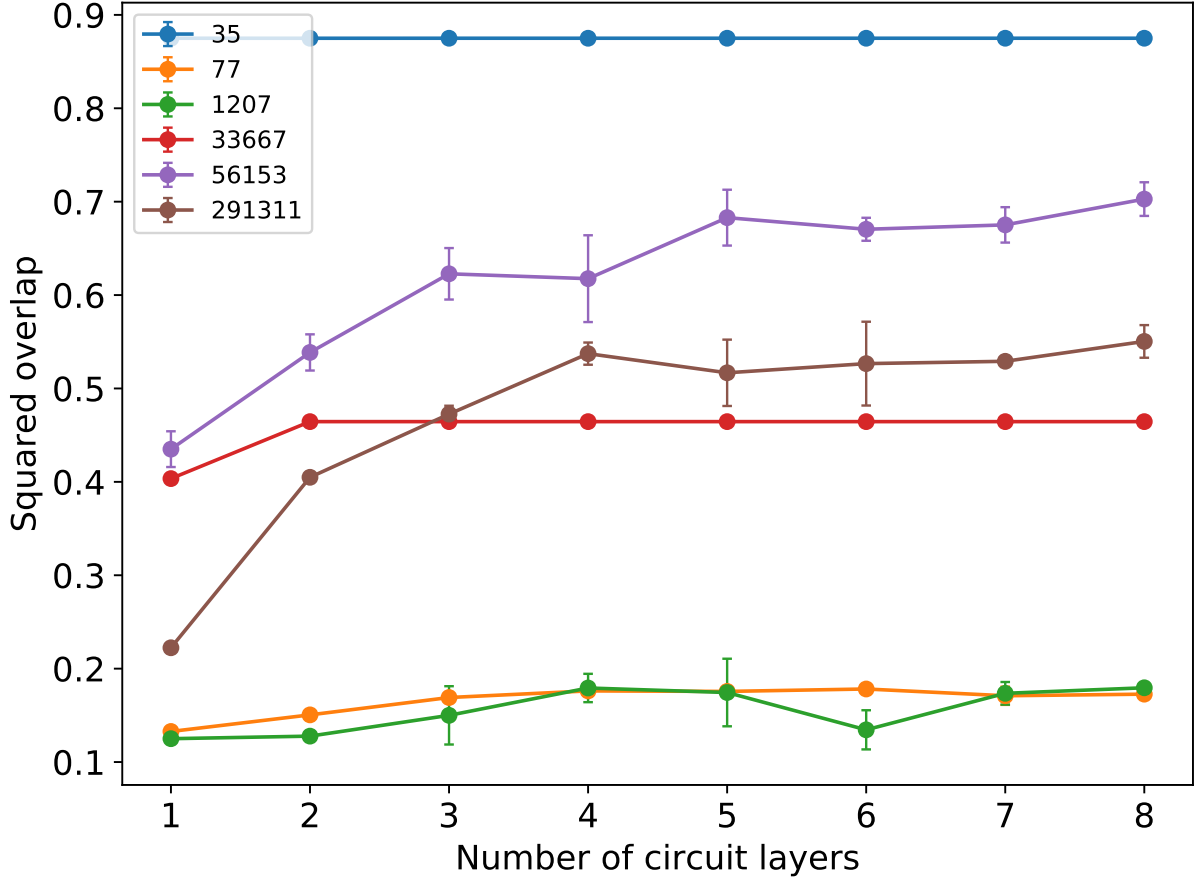


FIG. 2. The squared overlap of the optimized VQF state with the solution state manifold of H_c for all problem instances considered. Here, we fixed the error rate $\varepsilon = 10^{-3}$ and the number of samples $\nu = 10000$. We note the drastically reduced depth scaling for $m = 77, 1207, 33667$ (see Section IV A). The error bars each denote one standard deviation over three problem instances.

until convergence (to either a local or a global minimum) seemed independent of the problem size and depended linearly on the circuit depth (see Figure 3). For a QAOA ansatz of depth s , this puts the total cost of performing VQF at $O(s^2 n_c^4 n^8)$ in the worst case, though numerically, this seems like a loose bound. We also note that there is no guarantee that this procedure always generates the globally optimal solution.

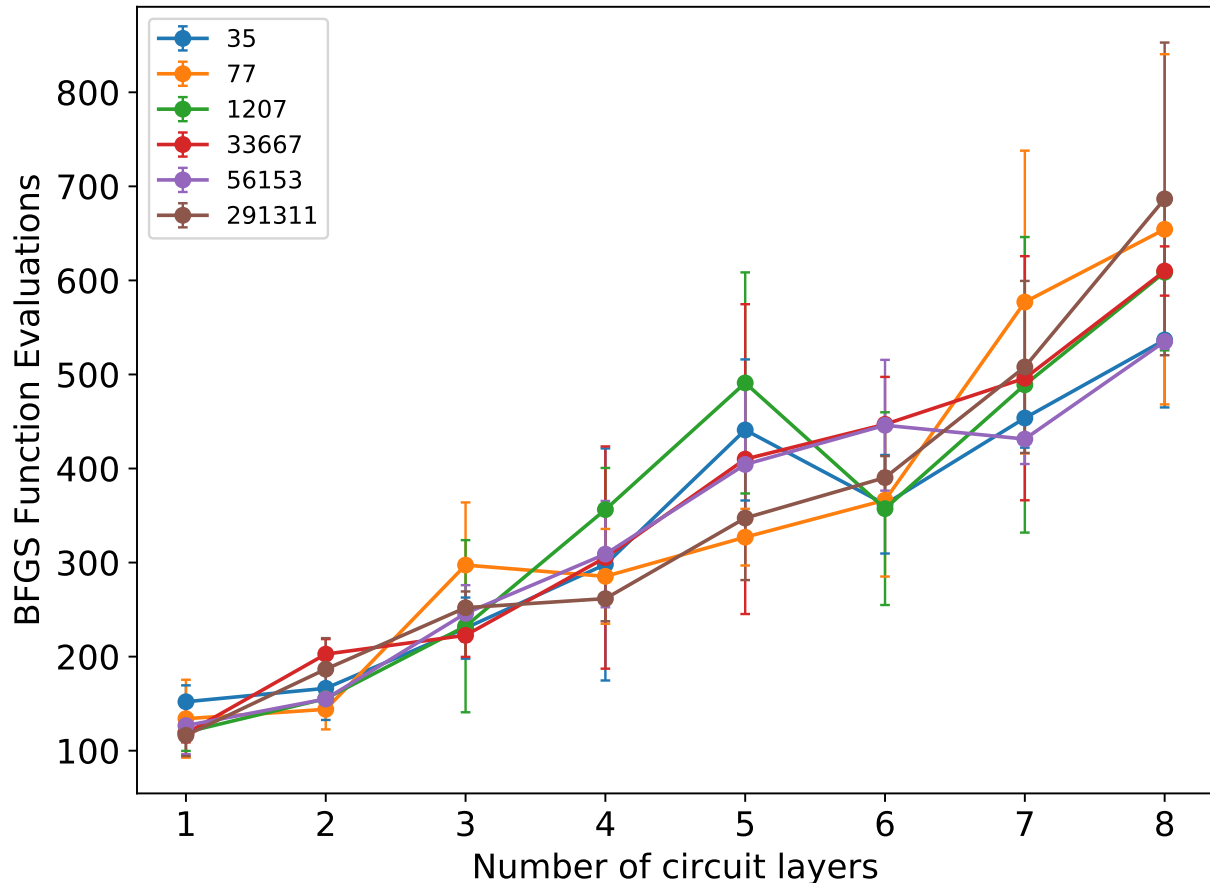


FIG. 3. The scaling of the number evaluations of Equation (11) needed before the BFGS optimization converges. The scaling is approximately linear in the number of parameters, and is approximately independent of the problem size. The error bars each denote one standard deviation over three problem instances.

IV. NUMERICAL SIMULATIONS

A. Depth Scaling

We performed noisy simulation of a number of instances of biprime factoring using the algorithm described above [43] (see Section IV B for a description of our noise model). Table I lists all of the instances used. With the technique described in Section III, the success probability of finding the correct factors of $m = 35, 77, 1207, 33667, 56153, 291311$ as a function of the number of circuit layers s is plotted in Figure 2. The output distributions for representative numbers are plotted in Figures 5 and 6. Here, “squared overlap” refers to

the squared overlap of the output VQF state with the solution state manifold of H_c —that is, the squared overlap with states with the correct assignments of all p_i and q_i but not necessarily of all the carry bits z_{ij} , which are not bits of the desired factors p and q .

For $m = 35,56153,291311$, after $O(n)$ circuit layers, the success probability plateaus to a large fraction. As factoring is efficient to check, one can then sample from the optimized VQF ansatz and check samples until correct factors of m are found. However, the algorithm does not scale as well with the circuit depth for $m = 77,1207,33667$. This is the case even though the $m = 77,33667$ problem instances have the same number or fewer qubits required than the $m = 56153,291311$ problem instances. Further insight is needed to explain this discrepancy, though we do notice that unlike $m = 35,56153,291311$, these instances lack $p \leftrightarrow q$ symmetry and contain carry bits in their classical energy functions (6) (see Table I).

B. Noise Scaling

An obvious concern for the scalability of the algorithm is the effect of noise on the performance of VQF. To explore this empirically, we considered a Pauli channel error model; that is, after every unitary (and after the preparation of $|+\rangle^{\otimes n}$) in Equation (9), we implemented the noise channel

$$\rho \mapsto (1 - n\varepsilon) \rho + \frac{\varepsilon}{3} \sum_{j=1}^n \sum_{i=1}^3 \sigma_j^{(i)} \rho \sigma_j^{(i)}, \quad (12)$$

where ε is the single qubit error rate. Included in the simulation is sampling noise with $\nu = 10000$ samples when estimating the cost function $M(\beta, \gamma)$. We plot the dependence of two VQF instances on the noise rate in Figure 4, and note that VQF is weakly dependent on the noise rate below a certain error threshold.

V. DISCUSSION

The ability to efficiently solve integer factorization has significant implications for public-key cryptography. In particular, encryption schemes based on abelian groups such as RSA and elliptic curves can be compromised if efficient factorization were feasible. However, an implementation of Shor’s algorithm for factoring cryptographically relevant integers would require thousands of *error-corrected* qubits [20, 21]. This is far too many for noisy

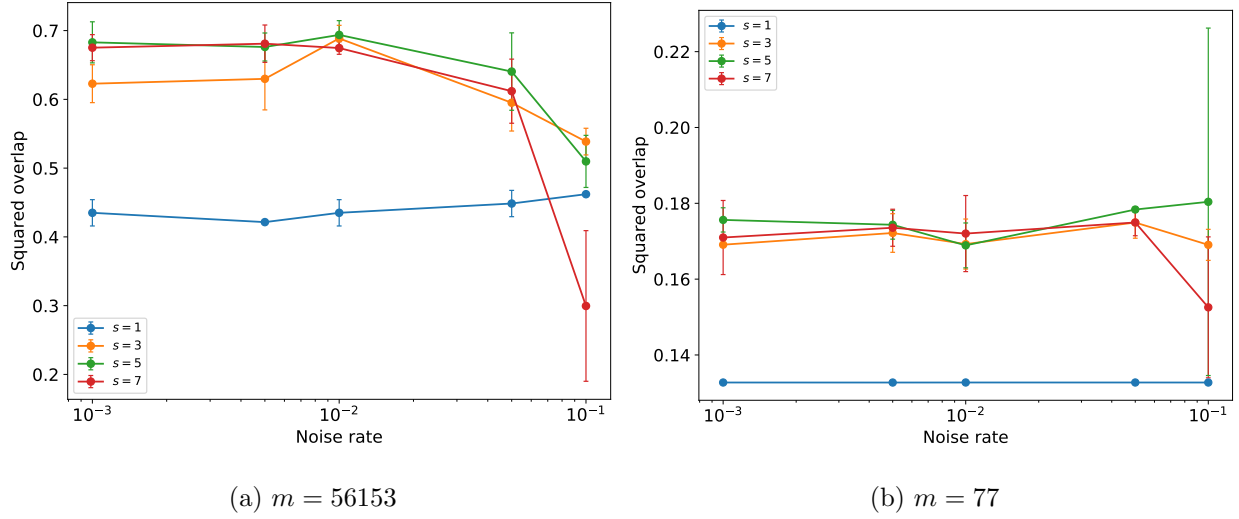


FIG. 4. The dependence on factoring (a) $m = 56153$ and (b) $m = 77$ at various depths for different Pauli error noise rates. Below a certain error threshold, the success probability is approximately independent of the noise rate. The error bars each denote one standard deviation over three problem instances.

intermediate-scale quantum devices that are available in the near-term, rendering the potential of quantum computers to compromise modern cryptosystems with Shor’s algorithm a distant reality. Hybrid approximate classical/quantum methods that utilize classical pre- and post-processing techniques, like the proposed VQF approach, may be more amenable to factoring on a quantum computer in the next decade.

Although we show that it is in principle possible to factor using VQF, as with most heuristic algorithms, it remains to be seen whether it is capable of scaling asymptotically under realistic constraints posed by imperfect optimization methods and noise on quantum devices. We are currently in the process of examining more detailed analytical and empirical arguments to better determine the potential scalability of the protocol under realistic NISQ conditions. We look forward to working with our collaborators on experimental implementations on current NISQ devices.

The VQF approach can also be employed in an error-corrected setting. Given its heuristic approach it presents a tradeoff between the number of coherent gates and the number of repetitions, similar to the previous VQE and QAE approaches. In this sense, VQF could be competitive with Shor’s algorithm even in the regime of fault-tolerant quantum computation. However, further work is needed in comparing the resources needed for both approaches,

including understanding what causes VQF to struggle with certain factoring instances—preliminary numerics suggest that the mere presence of carry bits negatively affects the algorithm, with little dependence on the number of carry bits for a fixed problem size.

In conclusion, the VQF approach discussed here presents many stimulating challenges for the community. QAOA, the optimization algorithm employed in our approach, has been studied by several groups in order to understand its effectiveness in several situations [32, 44–49]. VQF inherits both the power and limitations of QAOA, and therefore many more numerical and analytical studies are needed to understand the power of VQF in the near future.

ACKNOWLEDGMENTS

We would like to acknowledge the Zapata Computing scientific team, including Peter Johnson, Jhonathan Romero, Borja Peropadre, and Hannah Sim for their insightful and inspiring comments.

-
- [1] P. W. Shor, SIAM Review **41**, 303 (1999), arXiv:9508027 [quant-ph].
 - [2] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, Science **351**, 1068 (2016), arXiv:1507.08852 [quant-ph].
 - [3] C. Y. Lu, D. E. Browne, T. Yang, and J. W. Pan, Physical Review Letters **99**, 1 (2007), arXiv:0705.1684 [quant-ph].
 - [4] E. Martín-López, A. Laing, T. Lawson, R. Alvarez, X. Q. Zhou, and J. L. O’Brien, Nature Photonics **6**, 773 (2012), arXiv:1111.4147 [quant-ph].
 - [5] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. V. James, A. Gilchrist, and A. G. White, Physical Review Letters **99**, 5 (2007), arXiv:0705.1398 [quant-ph].
 - [6] A. Politi, J. C. F. Matthews, and J. L. O’Brien, Science **325**, 1221 (2009), arXiv:0911.1242 [quant-ph].
 - [7] E. Lucero, R. Barends, Y. Chen, J. Kelly, M. Mariantoni, A. Megrant, P. O’Malley, D. Sank, A. Vainsencher, J. Wenner, T. White, Y. Yin, A. N. Cleland, and J. M. Martinis, Nature Physics **8**, 719 (2012), arXiv:1202.5707 [quant-ph].

- [8] M. R. Geller and Z. Zhou, Scientific Reports **3**, 1 (2013), arXiv:1304.0128 [quant-ph].
- [9] S. Beauregard, Quantum Information & Computation **3**, 175 (2003), arXiv:0205095 [quant-ph].
- [10] M. Ekerå, IACR Cryptology ePrint Archive (2016).
- [11] M. Ekerå and J. Håstad, in *Post-Quantum Cryptography*, edited by T. Lange and T. Takagi (Springer International Publishing, Cham, 2017) pp. 347–363, arXiv:9508027 [cs.CR].
- [12] T. Häner, M. Roetteler, and K. M. Svore, Quantum Information & Computation **17** (2017), arXiv:1611.07995 [quant-ph].
- [13] Y. Takahashi and N. Kunihiro, Quantum Information & Computation **6**, 184 (2006).
- [14] M. A. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary ed. (Cambridge University Press, Cambridge, 2010).
- [15] A. G. Fowler, S. J. Devitt, and L. C. L. Hollenberg, Quantum Information & Computation **4**, 237 (2004), arXiv:0402196 [quant-ph].
- [16] A. Y. Kitaev, (1995), arXiv:9511026 [quant-ph].
- [17] C. Zalka, (2006), arXiv:0601097 [quant-ph].
- [18] C. Gidney, (2017), arXiv:1706.07884 [quant-ph].
- [19] A. G. Fowler and L. C. L. Hollenberg, Physical Review A **70** (2004), arXiv:0306018 [quant-ph].
- [20] N. C. Jones, R. Van Meter, A. G. Fowler, P. L. McMahon, J. Kim, T. D. Ladd, and Y. Yamamoto, Physical Review X **2**, 1 (2012), arXiv:1010.5022 [quant-ph].
- [21] R. V. Meter, T. D. Ladd, A. G. Fowler, and Y. Yamamoto, International Journal of Quantum Information **8**, 295 (2010), arXiv:0906.2686 [quant-ph].
- [22] D. D. Thaker, T. S. Metodi, A. W. Cross, I. L. Chuang, and F. T. Chong, in *33rd International Symposium on Computer Architecture (ISCA'06)*, Vol. 2006 (2006) pp. 378–389, arXiv:0604070 [quant-ph].
- [23] S. J. Devitt, A. M. Stephens, W. J. Munro, and K. Nemoto, Nature Communications **4**, 1 (2013), arXiv:1212.4934 [quant-ph].
- [24] J. Preskill, Quantum **2**, 79 (2018), arXiv:1801.00862 [quant-ph].
- [25] D. J. Bernstein, J.-F. Biasse, and M. Mosca, in *Post-Quantum Cryptography*, edited by T. Lange and T. Takagi (Springer International Publishing, Cham, 2017) pp. 330–346.
- [26] C. J. C. Burges, *Factoring as Optimization*, Tech. Rep. (2002).
- [27] N. S. Dattani and N. Bryans, (2014), arXiv:1411.6758 [quant-ph].

- [28] N. Xu, J. Zhu, D. Lu, X. Zhou, X. Peng, and J. Du, Physical Review Letters **108**, 1 (2012), arXiv:1111.3726 [quant-ph].
- [29] G. Schaller and R. Schützhold, Quantum Information & Computation **10**, 109 (2010), arXiv:0708.1882 [quant-ph].
- [30] S. Jiang, K. A. Britt, A. J. McCaskey, T. S. Humble, and S. Kais, (2018), arXiv:1804.02733 [quant-ph].
- [31] X. Peng, Z. Liao, N. Xu, G. Qin, X. Zhou, D. Suter, and J. Du, Physical Review Letters **101**, 220405 (2008), arXiv:0808.1935 [quant-ph].
- [32] E. Farhi, J. Goldstone, and S. Gutmann, (2014), arXiv:1411.4028 [quant-ph].
- [33] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, Nature Communications **5**, 4213 (2014), arXiv:1304.3061 [quant-ph].
- [34] J. Romero, J. Olson, and A. Aspuru-Guzik, Quantum Science and Technology **2**, 045001 (2016), arXiv:1612.02806 [quant-ph].
- [35] R. Dridi and H. Alghassi, Scientific Reports **7**, 43048 (2017), arXiv:1604.05796 [quant-ph].
- [36] N. Xu, J. Zhu, D. Lu, X. Zhou, X. Peng, and J. Du, Physical Review Letters **108**, 130501 (2012), arXiv:1111.3726 [quant-ph].
- [37] To lower the needed qubits for our numerical simulations, we assumed prior knowledge of n_p and n_q .
- [38] M. Agrawal, N. Kayal, and N. Saxena, Annals of Mathematics **160**, 781 (2004).
- [39] We note that other simple relations exist that can be used for preprocessing—the simplified clauses for $m = 56153, 291311$ as used in our numerical simulations were given by [27] who utilized a different preprocessing scheme.
- [40] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, Molecular Physics **109**, 735 (2011), arXiv:1001.3855 [quant-ph].
- [41] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. (Wiley, New York, 2000).
- [42] Consider a 1D example: Suppose we would like to minimize a function $f(x)$ for x on a finite interval. The derivative of the function is bounded $|f'(x)| < D$. Let x^* be the optimal point on the interval. We introduce a mesh to discretize the interval such that the mesh point \tilde{x} closest to x^* has Taylor expansion $f(x^*) \approx f(\tilde{x}) + (\tilde{x} - x^*) f'(\tilde{x})$. Then $|f(x^*) - f(\tilde{x})| \leq C$ for some constant C when we set the mesh dense enough such that $|x^* - \tilde{x}| \leq \frac{C}{D}$. This translates to $O(D)$ mesh points. For a 2D plane naturally the mesh choice is $O(D) \times O(D)$.

- [43] The simulation is performed using QuTiP [50]. To access the data generated for all instances considered in this study, including those which produced Figures 2-6, please refer to our Github repository at <https://github.com/zapatacomputing/VQFData>.
- [44] E. Farhi and A. W. Harrow, (2016), arXiv:1602.07674 [quant-ph].
- [45] G. Nannicini, (2018), arXiv:1805.12037 [quant-ph].
- [46] D. Venturelli, M. Do, E. Rieffel, and J. Frank, in *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI'17* (AAAI Press, 2017) pp. 4440–4446.
- [47] C. Y.-Y. Lin and Y. Zhu, (2016), arXiv:1601.01744 [quant-ph].
- [48] J. S. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. Schuyler Fried, S. Hong, P. Karalekas, C. B. Osborn, A. Papageorge, E. C. Peterson, G. Prawiroatmodjo, N. Rubin, C. A. Ryan, D. Scarabelli, M. Scheer, E. A. Sete, P. Sivarajah, R. S. Smith, A. Staley, N. Tezak, W. J. Zeng, A. Hudson, B. R. Johnson, M. Reagor, M. P. da Silva, and C. Rigetti, (2017), arXiv:1712.05771 [quant-ph].
- [49] E. S. Fried, N. P. D. Sawaya, Y. Cao, I. D. Kivlichan, J. Romero, and A. Aspuru-Guzik, (2017), arXiv:1709.03636 [quant-ph].
- [50] J. Johansson, P. Nation, and F. Nori, *Computer Physics Communications* **183**, 1760 (2012), arXiv:1211.6518 [quant-ph].

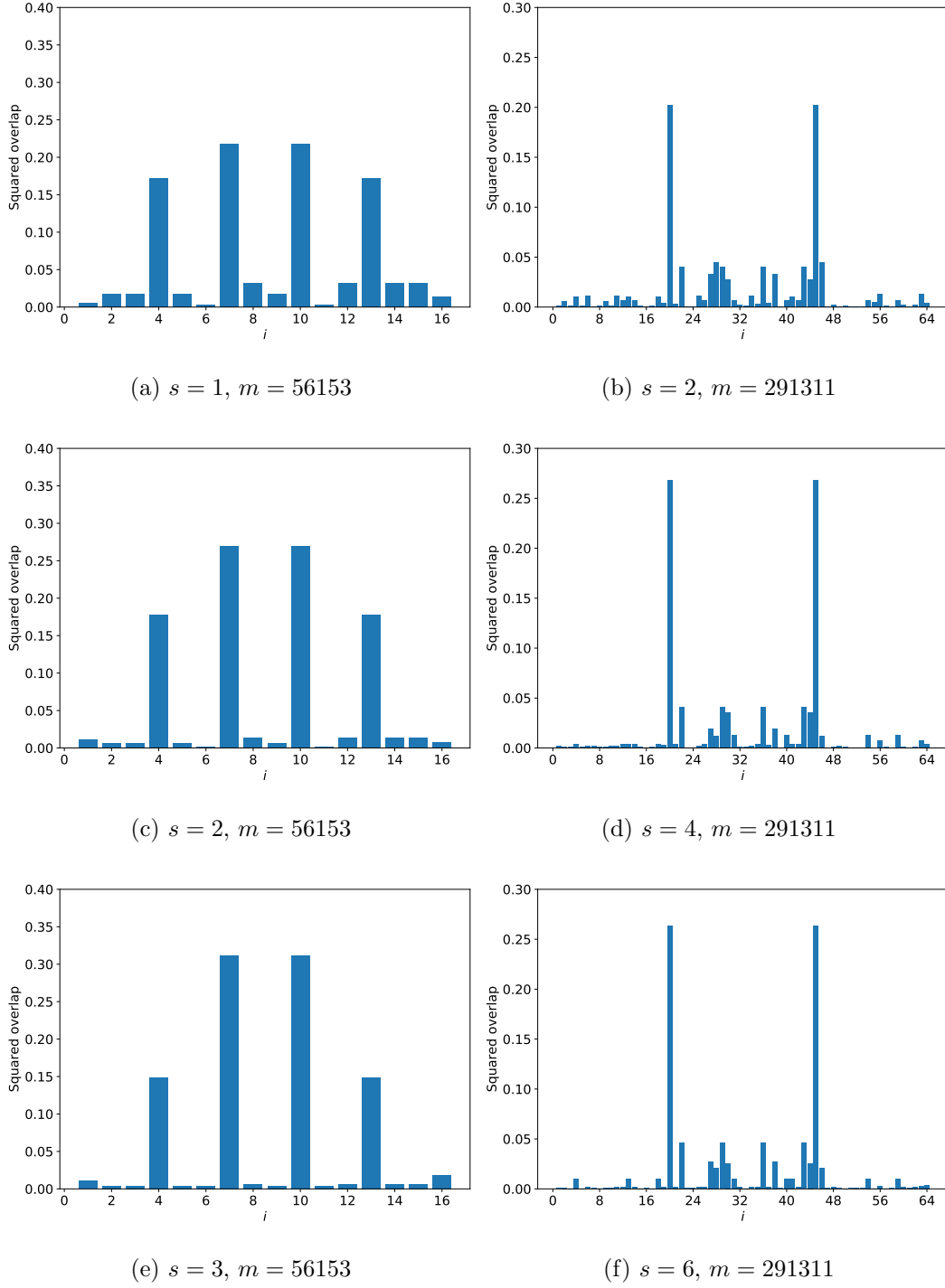


FIG. 5. Distributions corresponding to the output of the presented factoring algorithm for various circuit depths. i labels computational basis states in lexicographic order. The two modes of each diagram correspond to the computational basis states yielding the correct p and q ; there are two modes due to the $p \leftrightarrow q$ symmetry of the problem. Here, we fixed the error rate $\varepsilon = 10^{-3}$ and the number of samples $\nu = 10000$.

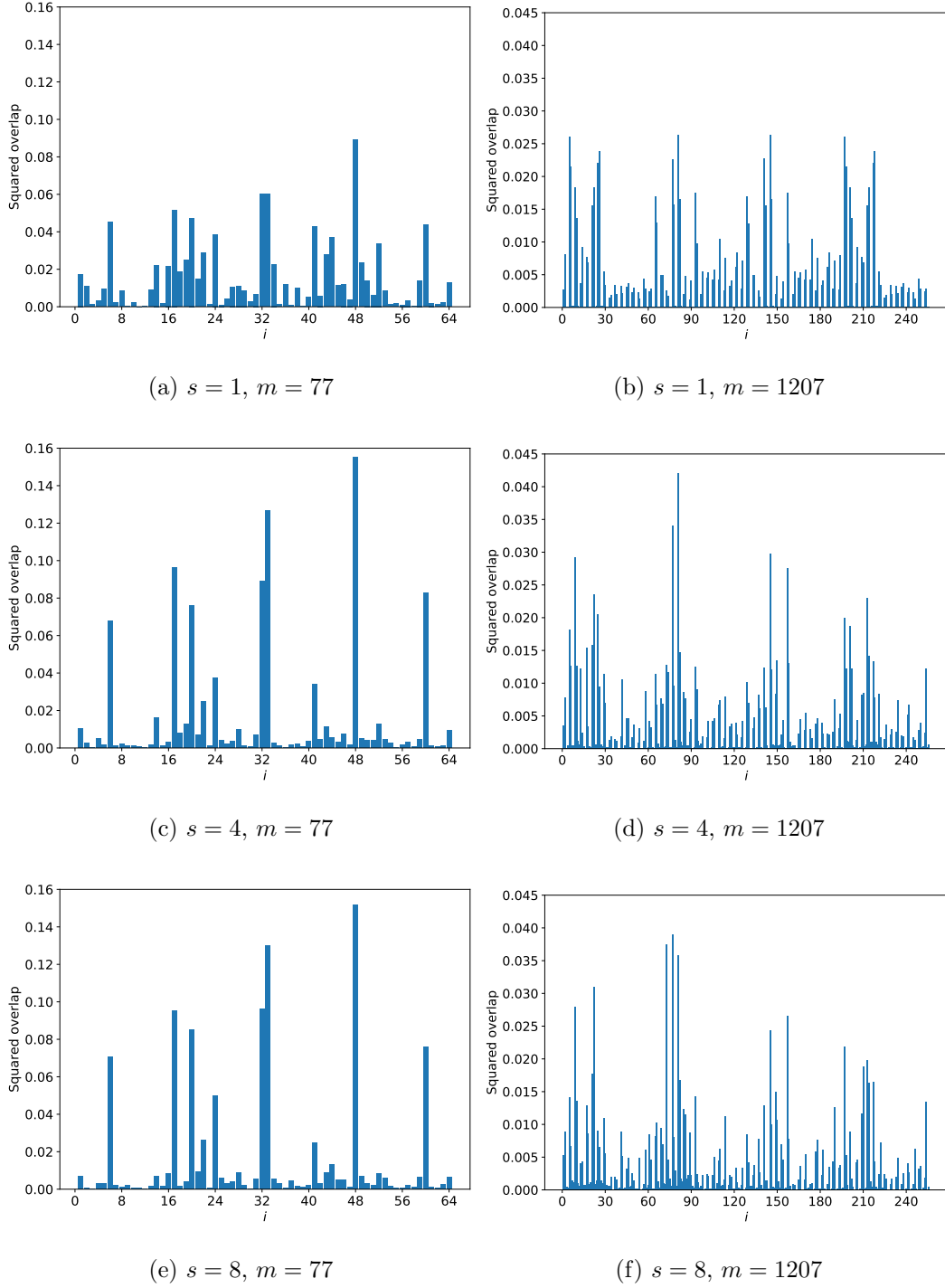


FIG. 6. Distributions corresponding to the output of the presented factoring algorithm for various circuit depths. i labels computational basis states in lexicographic order. The modes of the high depth distributions are the correct ground states. We notice worse performance than $m = 56153, 291311$ (see Figure 5 and Section IV A). Here, we fixed the error rate $\varepsilon = 10^{-3}$ and the number of samples $\nu = 10000$.