Emulating Quantum Circuits With Generalized Ising Machines

SHUVRO CHOWDHURY1, (Member, IEEE), KEREM Y. ÇAMSARI1, (Senior Member, IEEE), AND SUPRIYO DATTA2, (Life Fellow, IEEE)

1Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106, USA
2Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA

Corresponding author: Shuvro Chowdhury (schowdhury@ucsb.edu)

This work was supported in part by ASCENT, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) Program sponsored by Defense Advanced Research Projects Agency (DARPA); in part by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under Grant CCF-0939370; and in part by the Office of the Naval Research Young Investigator Program (YIP) Program.

ABSTRACT

The primary objective of this paper is to present an exact and general procedure for mapping any sequence of quantum gates onto a network of probabilistic p-bits which can take on one of two values 0 and 1. The first n p-bits represent the input qubits, while the other p-bits represent the qubits after the application of successive gating operations. We can view this structure as a Boltzmann machine whose states each represent a Feynman path leading from an initial configuration of qubits to a final configuration. Each such path has a complex amplitude ψ which can be associated with a complex energy. The real part of this energy can be used to generate samples of Feynman paths in the usual way, while the imaginary part is accounted for by treating the samples as complex entities, unlike ordinary Boltzmann machines where samples are positive. Quantum gates often have purely imaginary energy functions for which all configurations have the same probability and one cannot take advantage of sampling techniques. Typically this would require us to collect 2nd samples which would severely limit its utility. However, if we can use suitable transformations to introduce a real part in the energy function then powerful sampling algorithms like Gibbs sampling can be harnessed to get acceptable results with far fewer samples and perhaps even escape the exponential scaling with nd. This algorithmic acceleration can then be supplemented with special-purpose hardware accelerators like Ising Machines which can obtain a very large number of samples per second through a combination of massive parallelism, pipelining, and clockless mixed-signal operation made possible by codesigning circuits and architectures to match the algorithm. Our results for mapping an arbitrary quantum circuit to a Boltzmann machine with a complex energy function should help push the boundaries of the simulability of quantum circuits with probabilistic resources and compare them with NISQ-era quantum computers.

INDEX TERMS Hardware accelerators, Ising machines, massive parallelism, p-bits, quantum circuits, Feynman path, Monte Carlo.

I. INTRODUCTION

Quantum computing is based on the use of quantum gates to perform d successive unitary transformations (gates) U(1), U(2), . . . , U(d−1), U(d) on a set of n qubits so that their wavefunctions evolve from an initial |ψ(0)⟩ to a final |ψ(d)⟩ (Fig. 1a) [1]. Each of these wavefunctions |ψ(d)⟩ has 2n components, coming from a tensor product of n single qubit wavefunctions with two complex components each. In classical computing, a direct deterministic calculation requires us to multiply 2n × 2n transformation matrices with exponentially large memory requirements as n increases. By contrast, a quantum computer requires only n qubits which naturally live in 2n × 2n Fock space. Interestingly, a classical probabilistic computer too lives in this 2n × 2n Fock space, but is described by a probability
distribution function with positive elements, unlike the complex wavefunction that describes a quantum computer. Indeed in the seminal paper that inspired the field of quantum computing, Feynman remarked [2] ...The only difference between a probabilistic classical world and the equations of the quantum world is that somehow or other it appears as if the probabilities would have to go negative ... However, this fundamental difference leads to interesting properties of quantum circuits such as entanglement and interference [1] that give quantum computing its theoretical power.

In the field of adiabatic quantum computing (AQC), it is well-known that a significant subset of Hamiltonians like the transverse field Ising model (TFIM) are stoquastic, which means that the elements of the matrix $\exp(-H)$ are positive and can be evaluated using Monte Carlo techniques [3], [4] similar to those widely used to evaluate classical probabilities. It is well-known that quantum Monte Carlo (QMC) techniques can be used for non-stoquastic matrices as well though their accuracy is limited by the sign problem. This is important since gate-based quantum computing (GQC) is based on unitary matrices of the form $\exp(-iH)$ which are nearly always non-stoquastic.

The primary contribution of this paper is to present an exact and general procedure for mapping any sequence of quantum gates onto a probabilistic computer. This mapping should be particularly useful in view of the advent of special-purpose hardware accelerators known as “Ising Machines” and “digital annealers” [5], [6], [7], [8], [9], [10], [11], [12] which are being used to simulate the statistical mechanics of Ising models, onto which many known combinatorial optimization problems have been mapped [13]. These special purpose machines make use of random number generators (RNGs) to obtain a very large number of samples per second [14] through massively parallel operation and can be used to speed up the emulation of quantum circuits, once they have been mapped onto a probabilistic framework.

Note that we expect the probabilistic computer to improve the time to solution through the prefactor, but we do not expect it to change the asymptotic scaling behavior. For example, we present results showing the implementation of Shor’s algorithm in less than 2 hours on an ordinary laptop for a 27-bit number ($n = 27$), which is much larger than previously reported. But the time scales exponentially $O(2^n)$ unlike a true noiseless quantum computer which is expected to show linear scaling $O(n)$. However, quantum computers require stringent control of phase which is difficult even at cryogenic temperatures [15], [16] and current research activities in quantum computing have moved to noisy intermediate scale quantum (NISQ) computing for which there is no established asymptotic scale-up. As such there is strong interest in pushing the boundaries of classical computing [17], [18], [19], [20], [21], [22], [23]. Probabilistic computers can be built to operate at room temperature using existing technology, and energy-efficient compact realizations may be possible using stochastic nanomagnets [10].

The basic procedure for translating a q-circuit into a p-circuit is shown in Fig. 1. The first $n$ p-bits represent the input qubits, while the other $p$-bits represent the qubits after the application of successive gating operations (Fig. 1b). In the spirit of earlier works [24], [25], [26], [27], [28], [29], [30], [31], we can view this structure as a Boltzmann machine whose states each represent a Feynman path leading from an initial configuration of qubits to a final configuration, via specific intermediate configurations. Each such path has a complex amplitude $\psi$ which can be associated with a complex energy. The real part of this energy can be used to generate samples of Feynman paths in the usual way, while the imaginary part is accounted for by treating the samples as complex entities, unlike ordinary Boltzmann machines where samples are positive.

Recently, Boltzmann machines with complex weights have also been employed with machine learning techniques for various classes of quantum Hamiltonians. It has been shown that certain representations of Boltzmann machines can be trained to obtain the ground state of a Hamiltonian, and can even be used to represent quantum states exactly [32], [33], [34], [35]. Our approach here is fundamentally different in that no training of weights is involved. Our weights are obtained analytically as described in Section II and might be called “one-shot” learning.

II. COMPLEX ENERGY FUNCTIONS FOR QUANTUM GATES

The basic rule for writing a complex energy function for a sequence of gate operations can be obtained as follows. First, we note that the elements of the overall transformation matrix are given by a matrix product of the individual matrices

$$U_{\alpha,\beta} = \sum_{p_1, \cdots, p_{d-1}} U_{\alpha,p_{d-1}}^{(d)} U_{p_{d-2},p_{d-1}}^{(d-1)} \cdots U_{p_2,p_1}^{(2)} U_{p_1,\beta}^{(1)}$$  \hspace{1cm} (1)$$

where $|\beta\rangle$ and $|\alpha\rangle$ correspond to the initial and the final state of the qubits respectively. We express each $k^{th}$ element of the transformation matrices in terms of a corresponding energy.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Equation (3) is an exact result, but it represents a sum over a large number of paths (also referred to as Feynman paths in this context)

\[
\beta \rightarrow p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_{d-1} \rightarrow \alpha
\]

(from an initial state \(\beta\) to a final state \(\alpha\)) which grows exponentially in \(n\).

Usually, the energies are real so that \(e^{-E}\) can be interpreted as a probability, and probabilistic approaches allow us to sample the most important paths based on powerful algorithms like Metropolis or Gibbs sampling. For complex energies, we can still sample the paths based on the real part of \(E\) while the imaginary part can be interpreted as the complex contribution of unit magnitude contributed by a particular path. Next, we will show how to obtain energy functions for one and two-qubit gates and the sampling procedure will be discussed in the next section.

### A. ONE-QUBIT GATES

Any one-qubit gate is in general described by a transformation matrix of the form

\[
U^{(d)} = \begin{bmatrix} a_1 & b_1 \\ c_1 & A_1 \end{bmatrix}.
\]

Using (2) we can immediately write the energy function in tabular form

\[
E^{(d)} = s^{(d)}(1 - s^{(d)}) \ln (a_1) + (1 - s^{(d)}) s^{(d-1)} \ln (b_1) + s^{(d)}(1 - s^{(d-1)}) \ln (c_1) + s^{(d-1)} s^{(d)} \ln (A_1)
\]

which simplifies to

\[
E^{(d)} = - \ln (a_1) + s^{(d)} \ln (a_1/c_1) + s^{(d-1)} \ln (a_1/b_1) + s^{(d-1)} s^{(d)} \ln (b_1 c_1 / a_1 A_1).
\]

Note that this energy function has linear and quadratic terms corresponding to one-body and two-body interactions in an Ising model, which requires a Boltzmann machine with a linear synaptic function derived from the gradient of the energy function.

### B. TWO QUBIT GATES

Any two-qubit gate is in general described by a transformation matrix of the form

\[
U^{(d)} = \begin{bmatrix} 00 & 01 & 10 & 11 \\ 00^{(d-1)} & 01^{(d-1)} & 10^{(d-1)} & 11^{(d-1)} \end{bmatrix} \begin{bmatrix} a_1 & b_1 & a_2 & b_2 \\ c_1 & A_1 & c_2 & A_2 \\ a_3 & b_3 & a_4 & b_4 \\ c_3 & A_3 & c_4 & A_4 \end{bmatrix}.
\]

The mapping between quantum circuits and Boltzmann machines: (a) In quantum computing, a sequence of unitary gates \(U^{(1)}, U^{(2)}, \ldots, U^{(d)}\) is applied to an initial wavefunction \(|\psi^{(0)}\rangle\) and a final wavefunction \(|\psi^{(d)}\rangle\), is obtained. A classical bitstream of 0s and 1s is then obtained from this wavefunction by “measuring” the qubits (not shown) which leads to the collapse of the wavefunction and it is a non-reversible operation. (b) The quantum circuit in (a) is mapped into a Boltzmann machine with \(p\)-bits. Note that although the quantum gates in Fig. 1a are time-ordered, the \(p\)-bit network in Fig. 1b is a reciprocal BM described with an energy function which is obtained by modeling each \(k\)-th quantum gate via a corresponding complex energy function, \(E^{(k)}\). Nevertheless, the time sequence of gates is reflected in the BM’s energy function. While in the quantum case, the number of qubits remains the same, in the Boltzmann analog, the number of \(p\)-bit required increases with the number of applied gates \(d\) as \(n + g(d)\). See Section II for more details on \(g(d)\).
Using (2) we can immediately write the energy function in tabular form

\[
E^{(d)} = s_1^{(d-1)} s_2^{(d-1)}
\]

\[
\begin{array}{cccc}
00 & 01 & 10 & 11 \\
10 & -\ln(a_1) - \ln(b_1) - \ln(a_2) - \ln(b_2) & & \\
11 & -\ln(c_1) - \ln(A_1) - \ln(c_2) - \ln(A_2) & -\ln(c_3) - \ln(A_3) - \ln(c_4) - \ln(A_4) & \\
\end{array}
\]

(9)

Once again this tabular result can be translated into a Boolean function like (7), but the energy function will have three-body and four-body interactions whose gradient leads to non-linear synaptic terms. Equation (9) with such three-body and four-body terms represents a higher-order Ising model that has been discussed in the learning context by Ref. [37]. Such “generalized” Ising models can be solved much like ordinary Ising models with two-body interactions, provided that the synaptic feedback can be computed, for example by an FPGA [6]. Naturally, the resistive crossbar arrays that accelerate linear synaptic operations [38] would not be suitable for this purpose.

It is possible to eliminate the three-body and four-body terms at the expense of additional p-bits by decomposing the gates in terms of a sequence of rotation operations [24], [39], making the synaptic function linear. Alternatively, the three and four-body interactions can be reduced to standard 2-body interactions using the methods described in [40], [41], [42], [43].

In general, in this scheme, it would require two p-bits (one input p-bit and one output p-bit) to implement each single qubit gate. If the quantum circuit consists of \(d\) one qubit gates in series then in Fig. 1, \(g(d)\) would be \(d\). But if the one qubit gate is diagonal (like the phase gates) then one can implement that gate using just a single p-bit (same p-bit will represent both input and output) (see the appendix) and \(g(d)\) would be 0 for a quantum circuit with ‘d’ such gates in series. Similarly, for two-qubit gates one would require 4 p-bits and \(g(d)\) here would be 2\(d\) for a circuit consisting of only two-qubit gates in series whenever it is possible to implement 4-body interactions without inserting any intermediate p-bits. For diagonal gates (like controlled phase gates), once again \(g(d)\) would be 0 since those gates can be implemented using just two p-bits. For a general quantum circuit with \(n\) qubits, \(g(d)\) would be approximately a linear function of \(n\).

It is an important result in quantum computing that single qubit gates along with CNOT gates are universal for quantum computation [1]. Therefore, the methodology presented here for translating one and two qubits can in principle be used to translate any quantum circuit into a p-bit network although it is possible to extend the methodology for gates operating on more than two qubits directly (for example, Toffoli or Fredkin gates which operate on three qubits).

Finally, we also note that it is not necessary to turn the tabular results in (7) and (9) into a Boolean function, one can store these tabular values in a lookup table and look for them up from that table as necessary when summing to get the total energy of a path and then use a Metropolis algorithm based sampling approach with the help of a “dedicated kernel” as explained in the next section.

C. ADIABATIC QUANTUM COMPUTING

Before moving on, we want to note that in this paper we focus on GQC which is based on unitary transformations \(U \propto \exp(-iH)\). In this case, the energy functions are rarely real since that requires the elements of the \(U\) matrix to be all real and positive, which is seldom the case. However, our approach is also applicable to adiabatic quantum computing (AQC) based on \(\exp(-\beta H)\) which can often have purely real and positive elements leading to purely real energy functions. Such Hamiltonians are classified as stoquastic (see for example, [44]) and can be emulated with standard BM’s. Non-stoquastic Hamiltonians requiring complex BM’s form a special subset of all problems of interest in AQC, while in GQC many problems belong to this category.

Interestingly, in one respect GQC is simpler than AQC because the GQC evolution operators are naturally built out of the product of few (typically 1 and 2) qubit operations.

\[
U = \exp(-iH_1) \exp(-iH_2) \cdots
\]

By contrast, it is not straightforward to do the reverse, namely to break up the evolution operator \(\exp(-\beta H)\) for AQC into a product of separate terms corresponding to the components \(H = H_1 + H_2 + \cdots\), since \(\exp(-\beta H) \neq \exp(-\beta H_1) \exp(-\beta H_2) \cdots\) unless the components \(H_1, H_2, \cdots\) commute. The standard approach is the Suzuki-Trotter transformation [45] which breaks up \(H\) into \(r\) replicas by writing

\[
\exp(-\beta H) = \left[ \exp(-\beta H/r) \right]^r \\
\approx \{ \exp(-\beta H_1/r) \exp(-\beta H_2/r) \cdots \}^r
\]

assuming that \(r\) is large enough to make the commutators of \(H_1/r, H_2/r, \cdots\) negligible. This paper focuses on GQC where replicas are not needed as explained above.

III. SAMPLING FROM FEYNMAN PATHS

Note that for a given input state \(\beta\) and output state \(\alpha\) our objective is to evaluate \(U_{\alpha,\beta}\) from (1) representing the summation of a very large number of Feynman paths each of which can be visualized as a state of the Boltzmann machine. However, unlike ordinary Boltzmann machines, each term is complex, requiring a modification of the standard sampling techniques as summarized below.

Monte Carlo techniques start by rewriting (1) in the form

\[
U_{\alpha,\beta} = \sum_{(C)} p_C W_C
\]

(10)

where we have defined

\[
W_C = \frac{U_\alpha U_{\beta,\alpha} U_{\beta,\alpha} \cdots U_{\beta,\alpha} U_{\beta,\alpha}}{p_C}
\]

(11)
\{C\} is the set of all possible paths (or configurations) from \( \beta \to \alpha \)

\[ C : \beta \to p_1 \to p_2 \to \cdots \to p_{d-1} \to \alpha \]

and \( p_C \) is a probability assigned to the path \( C \) as described below. The idea is to approximate the exact sum in (10) with a sum over \( T \) sample configurations generated according to the probability distribution \( \{p_C\} \).

\[ U_{\alpha,\beta} \approx \frac{1}{T} \sum_{k=1}^{T} W_{C_k} \]  \hspace{1cm} (12)

where \( C_k \) is the configuration sampled in \( k \)th sample. The average estimate from many sampled sums (12) will equal the correct value from the exact sum in (1) or (10).

Following the central limit theorem, the standard deviation \( \text{std}(T) \) of the Monte Carlo estimator with \( T \) samples is given by

\[ \text{std}(T) = \frac{\text{std}(\{W_C\}, \{p_C\})}{\sqrt{T}} \]  \hspace{1cm} (13)

where

\[ \text{std}(\{W_C\}, \{p_C\}) = \left( \sum_{\{C\}} p_C(W_C - U_{\alpha,\beta})^2 \right)^{1/2} \]  \hspace{1cm} (14)

Sampling techniques seek to minimize any \( \text{std}(\{W_C\}, \{p_C\}) \) through a judicious choice of \( \{p_C\} \).

Classically it is common to choose

\[ p_C \propto U_{\alpha,\beta}^{(d)} U_{p_{d-1},p_{d-2}}^{(d-1)} \cdots U_{p_1,\beta}^{(1)}. \]

A natural extension to quantum systems with complex path amplitudes is to choose

\[ p_C \propto \left| U_{\alpha,\beta}^{(d)} U_{p_{d-1},p_{d-2}}^{(d-1)} \cdots U_{p_1,\beta}^{(1)} \right| = \exp(\Im(E)) \]  \hspace{1cm} (15)

This choice leads to the flowchart shown in Fig. 2. The overall structure is very similar to the algorithms implemented in the modern digital annealers today [12]. There are two minor but profound differences though: first, since the energies are complex, one should use the ratios of the absolute values of the weights (proportional to \( \exp(\Im(E)) \)) when computing the acceptance ratio. This is different from standard Boltzmann machines where each path carries a positive real weight (there the imaginary part of \( E \) is always zero) which can be interpreted as a quantity directly proportional to the probability of that path. Second when counting the frequency of occurrence of a particular path (or configuration) one needs to add the phase of the weight of the path (which is equal to \( \exp(-i\Im(E)) \)) instead of just adding +1. So one also needs to have the ability to add complex numbers which should be relatively straightforward and easy to incorporate. For positive and real weights as in standard Boltzmann machines, instead of adding the phase of the weight of the path, one always adds +1 which can also be viewed as each path having a zero phase associated with it, as shown in Fig. 2.

**FIGURE 2.** Monte carlo procedure for sampling from Feynman paths: starts with an initial configuration or path \( \beta \to \gamma \) based on some scheme like Gibbs sampling. Step 2: Accept or reject the proposal based on the ratio of \( \exp(-E) \) for the proposed and initial configurations, using an algorithm like Metropolis-Hastings or Glauber. In classical Monte Carlo, \( E \) is real and \( \exp(-E) \) is a positive number. In quantum problems, \( E \) can be complex, and we use the absolute value of \( \exp(-E) \) as the acceptance ratio. Step 3: Add result to the output bin. In classical Monte Carlo, the output bin corresponding to the end state of the sampled path \( \psi \), if the proposed path is accepted or \( \phi_{\alpha} \) if the proposal is rejected is just incremented by 1. In order to include the “sign-correction” in quantum problems, the “phase” of the path is added instead of +1. This loop continues until a sufficient number of samples are taken. Step 4: Finally the output wavefunction is normalized by dividing each bin with the square root of the sum of the squares of values of all bins (also known as \( L^2 \) normalization).

**IV. EXAMPLE 1: SINGLE QUBIT, DEEP CIRCUIT**

It may seem surprising that complex probabilities can be included in a Monte Carlo simulation, but the basic idea can be appreciated with a one-qubit example. Consider one qubit driven by a sequence of Hadamard gates (Fig. 3a), each represented by a transformation connecting qubits in planes \( d - 1 \) and \( d \):

\[ U_{\text{Hadamard}} = \left( \frac{1}{\sqrt{2}} \right) \begin{bmatrix} 0^{(d)} & 1^{(d)} \end{bmatrix} \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix} \]  \hspace{1cm} (16)

Two applications of the gate result in an identity transformation:

\[ (U_{\text{Hadamard}})^2 = \begin{bmatrix} 0^{(d)} & 1^{(d)} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]  \hspace{1cm} (17)
If the qubit is initialized to $|0\rangle$, then after one gate it has an equal probability of being either in $|0\rangle$ or $|1\rangle$, but after two gates the qubit is back to the $|0\rangle$ state with 100% probability.

This can be viewed as a very simple illustration of the classic double-slit interference that Feynman used in his lectures to illustrate the difference between quantum and classical, or between electrons and bullets as he put it [46]. Our Boltzmann machine emulates this quantum interference using a complex energy function given by

$$E = i\pi (s_1 s_2 + s_2 s_3) + \text{constant}$$  \hspace{1cm} (18)

where $s_1, s_2, s_3$ represent the initial state, the state after one gate, and the state after two gates respectively. It is straightforward to check that with $a_1 = b_1 = c_1 = -A_1 = 1$ in (6), we obtain the result stated above in (18). Starting from $|0\rangle$, the qubit will also end up in $|0\rangle$ after the two Hadamard gates. But the bullet starting from a “0” (corresponding to $s_1 = 0$) can end up in each of the final states “0” (corresponding to $s_3 = 0$) and “1” (corresponding to $s_3 = 1$) via two paths (each path is denoted by $s_1 \rightarrow s_2 \rightarrow s_3$) as follows

\[
\begin{align*}
0 & \rightarrow 0 \rightarrow 0 \quad \text{(path 1)} \\
0 & \rightarrow 1 \rightarrow 0 \quad \text{(path 2)}
\end{align*}
\]

or,

\[
\begin{align*}
0 & \rightarrow 0 \rightarrow 1 \quad \text{(path 1)} \\
0 & \rightarrow 1 \rightarrow 1 \quad \text{(path 2)}
\end{align*}
\]

All paths have the same $\Re(E)$ and appear with equal probability in the sampling process giving the same result for both outputs if the imaginary part is ignored as shown in Fig. 3b. But the complex Boltzmann machine weights each path according to $\exp(-i\Im(E))$ so that the total contribution is

\[
\begin{align*}
\text{Final state 0: } & \quad e^{-i\pi(0+0)} + e^{-i\pi(0+0)} = 2 \\
\text{Final state 1: } & \quad e^{-i\pi(0+0)} + e^{-i\pi(0+1)} = 0.
\end{align*}
\]

Our complex Boltzmann machine thus emulates quantum interference using classical bullets by associating complex energies with the paths taken by the bullet and hence in principle provides results that are exact when all possible paths taken by a bullet (between a given input state to a given output state) are considered into account (Fig. 3c).

But p-bits with complex phases are still fundamentally different from qubits. We need four samples in order to get a net signal of 2 in state 0, and none in state 1. With noiseless qubits, two samples would have given us the same result, both would land in state 0. This difference becomes more pronounced as the depth of the circuit is increased (Fig. 4), as we will now discuss.

A. SIGN PROBLEM IN DEEP CIRCUITS

With two Hadamard gates the height of the correct peak is 1/2; with $d$ (d being even) gates it can be shown that the peak height is $1/2^d$ which becomes very small as $d$ is increased. This means that many more samples will be needed to reduce the noise to an acceptable level.

We can estimate the noise using (13) and (14) as follows. With an even number of gates, $d$, there are $2^d$ paths, half of which reach the correct output and half reach the wrong output. Let us consider the wrong output with a net signal $U_{a,b} = 0$. Since the paths have weight $\pm 1/2^{d/2}$, all with the same magnitude, it is best to use a uniform probability for all paths, with $p_C = 2/2^d$, so that from (11), $W_C = \pm 2^{d/2}/2$, and from (13)

$$\text{std}(T) = \frac{2^d}{\sqrt{2T}}$$  \hspace{1cm} (19)

noting that only half the samples, $T/2$, reach the wrong output. To reduce this noise below the signal $1/2^d$, the number of samples $T$ must exceed $2^d$.

B. “TAMING” THE SIGN PROBLEM

This exponential increase is characteristic of quantum gates arising from path cancellation, and is essentially the sign...
Path cancellations in Hadamard chain: Cancellation of paths are illustrated with a string of Hadamard (H) gates applied in sequence to an input clamped to $|0\rangle$. Half of the total paths cancel each other $|1\rangle$. Paths that cancel each other have been shown in the green dashed box. Even at $|0\rangle$, there are cancellations of paths, the amount of which also increases with the length of the chain. Only a small fraction of paths survive ($2^d/2$) out of total $2^d$ paths which after normalization produces a peak at $|0\rangle$.

\[ U_{H,\text{rotated}} = \begin{bmatrix} |0^{d-1}\rangle & |1^{d-1}\rangle \\ \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{bmatrix} \]  

(20)

With $\theta = \pi/4$ we have the standard Hadamard gate, but with $\theta = \pi/2$ it becomes a classical gate with only positive weights.

We still have $2^d$ total paths, half of which go to $|0\rangle$ and the other half to $|1\rangle$. But the paths have different contributions (for $\theta \neq \pi/4$) and it takes more work to calculate the standard deviations, std($\{W_C\}$, $\{p_C\}$) analytically from (14). Instead, we show numerical results in Fig. 5 for $n = 1$, $d = 10$ separately for paths that go to $|0\rangle$ and for paths that go to $|1\rangle$. In each case we show results for uniform sampling (equal $p_C$) and for non-uniform sampling based on 15.

The result for $\theta = \pi/4$ is the same as what we argued earlier in (19) for paths leading to $|1\rangle$. Note how a combination of rotation ($\theta$) and non-uniform sampling ($p_C$) helps reduce the std and hence the noise down to zero in the classical limit. This is an elementary example of how the sign problem can be “tamed” [47], [47], [48], [49] and we mention it as an important option that can help make the probabilistic emulation of quantum circuits more efficient in the future.

V. EXAMPLE 2: MULTIPLE QUBITS, SHALLOW CIRCUIT

Shor’s algorithm for integer factorization is a cornerstone example that is often cited to refer to the computational advantage that can be harnessed from a quantum computer. Shor’s algorithm ingeniously employs a quantum circuit to accelerate the process of finding the period $r$ of the function $f(x) = a^x \mod N$, where $N$ is the number to be factored and $a$ is a random number in the range $1 < a < N$ satisfying $\gcd(a, N) = 1$.

A. QUANTUM CIRCUIT

The commonly referred quantum circuit corresponding to this operation is shown in Fig. 6, employing two registers that are entangled to produce an output $|x, a^x \mod N\rangle$. A quantum Fourier Transform (QFT) is performed on the output of the first register to yield $|k, a^x \mod N\rangle$ on which measurements are made to obtain samples corresponding to one of the $r$ peaks, from which the value of $r$ is extracted using the continued fraction algorithm. In order to ensure the high success rate of this algorithm with a few samples, the number...
We could map the quantum circuit in Fig. 6 to a probabilistic circuit exactly. But it is possible to accelerate the implementation significantly with a few changes (Fig. 7):

- We use only \( t = n \) input p-bits in the first register, instead of the \( 2n \leq t \leq 2n + 1 \) qubits necessitated by the continued fraction algorithm. We obtain \( r \) directly by counting the number of peaks in the output probability distribution.
- Each sample can consist of a family of paths, each leading to one output, rather than a single path; leading to a single output as shown in Fig. 7. The idea is similar to the elementary version of column sampling technique used in the field of randomized numerical linear algebra (randLA) [50], [51].
- The noise goes down \( \sim \) the number of samples \( T \), while the peak strength \( \sim r \), so that we expect to need \( T > r \) to identify the peaks correctly. Our numerical experiments suggest that in general one requires \( T = \gamma r \) column samples, typically with \( 2 \leq \gamma \leq 4 \). The sampling process can be monitored regularly to decide when the peaks are clear and sampling can be stopped.
- A straightforward implementation would require \( 2^n \times 2^n \) elements to be stored in order to keep track of \( |k, a^t \text{mod } N\rangle \). However, in our work, this memory requirement was reduced to \( 3 \times 2^n \) by first generating the random column numbers and evaluating \( a^t \text{mod } N \) operations for each \( x \).

Using this probabilistic approach, we have factorized numbers up to \( \sim 10^8 \) as we will describe next.

**B. PROBABLISTIC CIRCUIT**

We could map the quantum circuit in Fig. 6 to a probabilistic circuit exactly. But it is possible to accelerate the implementation significantly with a few changes (Fig. 7):

- We use only \( t = n \) input p-bits in the first register, instead of the \( 2n \leq t \leq 2n + 1 \) qubits necessitated by the continued fraction algorithm. We obtain \( r \) directly by counting the number of peaks in the output probability distribution.
- Each sample can consist of a family of paths, each leading to one output, rather than a single path; leading to a single output as shown in Fig. 7. The idea is similar to the elementary version of column sampling technique used in the field of randomized numerical linear algebra (randLA) [50], [51].
- The noise goes down \( \sim \) the number of samples \( T \), while the peak strength \( \sim r \), so that we expect to need \( T > r \) to identify the peaks correctly. Our numerical experiments suggest that in general one requires \( T = \gamma r \) column samples, typically with \( 2 \leq \gamma \leq 4 \). The sampling process can be monitored regularly to decide when the peaks are clear and sampling can be stopped.
- A straightforward implementation would require \( 2^n \times 2^n \) elements to be stored in order to keep track of \( |k, a^t \text{mod } N\rangle \). However, in our work, this memory requirement was reduced to \( 3 \times 2^n \) by first generating the random column numbers and evaluating \( a^t \text{mod } N \) operations for each \( x \).
implemented recently on a supercomputer [53] and use our probabilistic approach to do the same problem on a laptop, and compare them in Table 1. We see that the probabilistic approach uses significantly less memory than the other approach mainly attributed to its reduction in the number of qubits in the first register.

Our improved probabilistic approach thus requires $O(r)$ column samples, with each column requiring the evaluation of $2^n$ elements which is of course, inferior to the polynomial scaling in $n$ expected from a qubit implementation that has made Shor’s algorithm legendary and in general hard to emulate with classical resources [53], [55], [56], [57], [58], [59]. However, it does give significant improvement in memory requirement, and could also give significantly better computation time if implemented on an Ising machine with high sample throughput.

Even on our single-core laptop, we are able to factorize much larger numbers than could be implemented deterministically. For example, Fig. 8, shows the output distribution obtained from the probabilistic approach when used to find the period of $f(x) = 10424^x \mod 98743069$ (the period of $f(x)$ is $r = 78$). We note that 98743069 is a 27-bit number and the period is obtained using only 250 column samples out of $2^{27} \approx 134$ million columns on the aforementioned machine in less than 2 hours (in $\sim 5923$ s). But this advantage diminishes gradually as the period increases because the number of column samples required to detect peaks clearly also grows almost linearly with the period as mentioned earlier.

### VI. HARDWARE EMULATION

The results in Table 1 of the previous section for the probabilistic approach may look underwhelming. However, we would like to draw attention to the possibility of hardware acceleration over and above the algorithmic acceleration discussed above. The results presented here are all based on algorithms implemented on a general-purpose CPU/GPU architected like a general-purpose Turing machine. However, it is known that significant acceleration is possible using special-purpose processors in the form of a dedicated circuit. It was recently shown that a 3D Ising computer implementing an AQC algorithm for stoquastic problems could be accelerated by orders of magnitude using digital circuits implemented on FPGA, and even more with mixed signal circuits [60]. Similar principles can be adapted to the GQC problems addressed in this paper. However, an additional unit for “sign correction” is needed to keep track of the phase of each sample. This is in addition to the sampling unit needed for stoquastic circuits for which the energy function has no imaginary part.

Both the Hadamard gate and the Quantum Fourier Transform circuit have purely imaginary energy functions. Since the real part is zero, samples are generated uniformly, all configurations having the same probability. Typically this would require us to collect $2^n$ samples which is very challenging once $n$ exceeds say 30. However, if we can use suitable transformations like the rotated Hadamard gate to introduce a real part in the energy function then powerful sampling algorithms like Gibbs sampling can be harnessed to get acceptable results with far fewer samples and perhaps even escape the exponential scaling with $n$.

### VII. CONCLUSION

In summary, we have presented a systematic and general procedure to translate any given quantum circuit exactly into a network of classical probabilistic p-bits each state of which is described by a complex amplitude. Although this mapping in general leads to the exponential asymptotic scaling with the size of the network and does not offer algorithmic advantage, we expect a significant lowering of
the prefactor by realizing that our general mapping makes it possible to directly transform any quantum circuit into a probabilistic network which then can be applied to a vast variety of fast-developing special-purpose Ising machines and thereby turning them into probabilistic simulators of any quantum circuit. This will also help in the growing effort to push the boundaries of the simulability of quantum circuits with classical/probabilistic resources and compare them with NISQ-era quantum computers.

APPENDIX A
ZERO ELEMENTS IN U-MATRIX

The procedure laid out in Sec. II for obtaining energy functions for given transformation matrices is straightforward. One point to note, however, is that often there are zero elements in the matrix which will lead to singular values for the logarithmic functions in Eqs. (7) or (9). A general but approximate way to deal with zero elements is to replace them with \(e^{-J}, J\) being a suitably large positive number. However, in special cases, an exact approach is possible, which is best illustrated with examples.

A. ONE QUBIT EXAMPLES

Consider the transformation \(R_z(\gamma)\) representing a one qubit rotation about the z-axis:

\[
R_z(\gamma) = e^{-i\gamma \sigma_z/2} = \begin{bmatrix} e^{-i\gamma/2} & 0 \\ 0 & e^{i\gamma/2} \end{bmatrix}
\]

Replacing the zero off-diagonal elements with \(e^{-J}, J\) being a large positive number, we have

\[
E^{(d)} = -i \frac{\gamma}{2} (1 - s^{(d)} - s^{(d)}) + \lim_{J \to \infty} J (s^{(d)} - 2s^{(d)} - 2s^{(d)} - s^{(d)})
\]

This is the straightforward approximate approach described above. Alternatively, we could eliminate \(s^{(d)}\) as an independent variable by setting it equal to \(s^{(d)}\), so that

\[
s^{(d)} = s^{(d)}
\]

\[
E^{(d)} = -i \frac{\gamma}{2} (1 - 2s^{(d)} - s^{(d)})
\]

This approach can also be used for zeros on the diagonal. Consider the Pauli-X gate:

\[
X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}
\]

We can eliminate \(s^{(d)}\) as an independent variable by setting it equal to \(1 - s^{(d)}\), so that

\[
s^{(d)} = 1 - s^{(d)}
\]

\[
E^{(d)} = 0
\]

Note that this is essentially a deterministic NOT gate.

B. TWO QUBIT EXCHANGE OPERATOR

Consider the two qubit exchange operator acting between qubit 1 and 2 defined as follows:

\[
J_{12} = \begin{bmatrix} e^{i\gamma/2} & 0 & 0 & 0 \\ 0 & e^{-i\gamma/2} & 0 & 0 \\ 0 & 0 & e^{-i\gamma/2} & 0 \\ 0 & 0 & 0 & e^{i\gamma/2} \end{bmatrix}
\]

Since the matrix is diagonal, we can eliminate \(s^{(d)}\), \(s^{(d)}\) by writing

\[
s^{(d)} = s^{(d)} + s^{(d)} - 2s^{(d)} - s^{(d)}
\]

It can be seen that this is essentially a deterministic XOR gate:

\[
s^{(d)} = \text{XOR} (s^{(d)}, s^{(d)})
\]

C. CNOT GATE

Consider the CNOT gate described by the transformation matrix:

\[
U_{\text{CNOT}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}
\]

In this case, we can write

\[
s^{(d)} = s^{(d)} + s^{(d)} - 2s^{(d)} - s^{(d)}
\]

\[
s^{(d)} = s^{(d)}
\]

\[
E^{(d)} = 0
\]

D. CCNOT GATE

Similarly for the CCNOT gate, we can write

\[
s^{(d)} = s^{(d)} + s^{(d)} - 2s^{(d)} - s^{(d)}
\]

\[
s^{(d)} = s^{(d)}
\]

\[
E^{(d)} = 0
\]

This too is essentially a deterministic gate:

\[
s^{(d)} = \text{XOR} (s^{(d)}, s^{(d)}, s^{(d)})
\]
ACKNOWLEDGMENT

The authors would like to thank Brian Sutton and Marc Cahay for extensive comments and suggestions on an earlier version of this manuscript, also would like to thank Diptiman Sen for useful discussions related to non-stoquastic Hamiltonians, and also would like to thank Jan Kaiser for many helpful discussions, especially those related to variance estimations. The authors also thank Dr. Seokmin Hong for helpful discussions regarding Fig. 8.

CONFLICT OF INTEREST

Supriyo Datta has a financial interest in Ludwig Computing. The authors declare no other competing interests.

REFERENCES

[1] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information 10th Anniversary Edition, 10th ed. Cambridge, U.K.: Cambridge Univ. Press, 2011.
[2] R. P. Feynman, “Simulating physics with computers,” Int. J. Theor. Phys., vol. 21, no. 6, pp. 467–488, Jun. 1982.
[3] M. Troyer and U.-J. Wiese, “Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations,” Phys. Rev. Lett., vol. 94, no. 17, May 2005, Art. no. 170201.
[4] K. Y. Camsari, S. Chowdhury, and S. Datta, “Scalable emulation of sign–problem–free Hamiltonians with room-temperature p-bits,” Phys. Rev. Appl., vol. 12, no. 3, Sep. 2019, Art. no. 034061. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevApplied.12.034061
[5] M. Yamaoka, C. Yoshihara, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, “A 20k-spin Ising chip to solve combinatorial optimization problems with CMOS annealing,” IEEE J. Solid-State Circuits, vol. 51, no. 1, pp. 303–309, Jan. 2016.
[6] P. L. McMahon, A. Marandi, Y. Haribara, R. Hamerly, C. Langrock, S. Tamate, T. Inagaki, H. Takesue, S. Usunomiya, K. Aihara, R. L. Byer, M. M. Fejer, H. Mabuchi, and Y. Yamamoto, “A fully programmable 100-spin coherent Ising machine with all-to-all connections,” Science, vol. 354, no. 6312, pp. 614–617, Nov. 2016.
[7] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeki, K. Enbutsu, O. Tadanaga, H. Takenouchi, K. Aihara, K.-I. Kawarabayashi, K. Inoue, S. Usunomiya, and H. Takesue, “A coherent Ising machine for 2000-node optimization problem,” Science, vol. 354, no. 6512, pp. 603–606, Nov. 2016.
[8] T. Wang and J. Roychowdhury, “Oscillator-based Ising machine,” 2017, arXiv:1709.08102.
[9] J. Chou, S. Bramhavar, S. Ghosh, and W. Herzog, “Analog coupled oscillator based weighted Ising machine,” Sci. Rep., vol. 9, no. 1, pp. 1–10, Oct. 2019.
[10] W. A. Borders, A. Z. Pervazia, S. Fukami, K. Y. Camsari, H. Ohno, and S. Datta, “Integer factorization using stochastic magnetic tunnel junctions,” Nature, vol. 573, no. 7734, pp. 390–393, Sep. 2019.
[11] S. Datta, A. Khanna, J. Gomez, K. Ni, Z. Toroczkai, and S. Datta, “Experimental demonstration of phase transition nano-oscillator based Ising machine,” in IEDM Tech. Dig., Dec. 2019, pp. 8–37.
[12] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, and H. Katzgraber, “Physics-inspired optimization for quadratic unconstrained problems using a digital annealer,” Frontiers Phys., vol. 7, p. 48, Apr. 2019. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fphy.2019.00048
[13] A. Lucas, “Ising formulations of many NP problems,” Frontiers Phys., vol. 2, p. 5, Jun. 2014.
[14] B. Sutton, R. Faria, L. A. Ghantasala, R. Jaiswal, K. Y. Camsari, and S. Datta, “Autonomous probabilistic coprocessing with petaflips per second,” 2019, arXiv:1907.09664.
[15] K. Noh, L. Jiang, and B. Fefferman, “Efficient classical simulation of noisy random quantum circuits in one dimension,” 2020, arXiv:2003.13163.
[16] M. Van den Nest, W. Dür, R. Raussendorf, and H. J. Briegel, “Quantum algorithms for spin models and simulable gate sets for quantum simulation,” Phys. Rev. A, Gen. Phys., vol. 80, no. 5, Nov. 2009, Art. no. 052334.
[17] G. D. L. Cuevas, W. Dür, M. V. den Nest, and M. A. Martin-Delgado, “Quantum algorithms for classical lattice models,” New J. Phys., vol. 13, no. 9, Sep. 2011, Art. no. 093021.
[18] M. J. Brenner, A. Montanaro, and D. J. Shepherd, “Average-case complexity versus approximate simulation of commuting quantum computations,” Phys. Rev. Lett., vol. 117, no. 8, Aug. 2016, Art. no. 080501.
[19] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Brenner, J. M. Martinis, and H. Neven, “Characterizing quantum supremacy in near-term devices,” Nature Phys., vol. 14, no. 6, pp. 595–600, Jun. 2018.
[20] K. Fujii and T. Morimae, “Commuting quantum circuits and complexity of Ising partition functions,” New J. Phys., vol. 19, no. 3, Mar. 2017, Art. no. 033003.
[21] B. Jonsson, B. Bauer, and G. Carleo, “Neural-network states for the classical simulation of quantum computing,” 2018, arXiv:1808.05242.
[22] C. Pehle and C. Wetterich, “Neuromorphic quantum computing,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 106, no. 4, Oct. 2022, Art. no. 045311.
[23] G. Carleo and M. Troyer, “Solving the quantum many-body problem with artificial neural networks,” Science, vol. 355, no. 6325, pp. 602–606, Feb. 2017.
[24] G. Carleo, Y. Nomura, and M. Imada, “Constructing exact representations of quantum many-body systems with deep neural networks,” Nature Commun., vol. 9, no. 1, p. 5322, Dec. 2018.
[25] X. Gao and L.-M. Duan, “Efficient representation of quantum many-body states with deep neural networks,” Nature Commun., vol. 8, no. 1, p. 662, Sep. 2017.
[26] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita, “Reconstructing quantum states with generative models,” Nature Mach. Intell., vol. 1, no. 3, pp. 155–161, Mar. 2019.
[27] J. F. Wackerly, Digital Design Principles and Practices. Englewood Cliffs, NJ, USA: Prentice-Hall, 2016.
[28] T. J. Sejnowski, “Higher-order Boltzmann machines,” in Proc. AIP Conf. Neural Netw. Comput. College Park, MD, USA: American Institute of Physics, 1987, pp. 398–403.
[29] M. Hu, J. P. Strachan, Z. Li, E. M. Graafs, N. Davila, C. Graves, S. Lam, N. Ge, J. I. Yang, and R. S. Williams, “Dot-product engine for neuromorphic computing: Programming ITIM crossbar to accelerate matrix-vector multiplication,” in Proc. 53rd ACM/EDAC/IEEE Design Auton. Conf. (DAC), Jul. 2016, pp. 1–6.
[30] J. Kim, J.-S. Lee, and S. Lee, “Implementing unitary operators in quantum computation,” Phys. Rev. A, Gen. Phys., vol. 61, no. 3, Feb. 2000, Art. no. 032312.
[31] S. Jiang, K. A. Britt, A. J. McCasey, T. S. Hamble, and S. Kais, “Quantum annealing for prime factorization,” Sci. Rep., vol. 8, no. 1, pp. 1–9, Dec. 2018.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply. 

S. Chowdhury et al.: Emulating Quantum Circuits With Generalized Ising Machines
SHUVRO CHOWDHURY (Member, IEEE) received the B.S. and M.S. degrees in electrical and electronic engineering from the Bangladesh University of Engineering and Technology, Bangladesh, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2022.

He was a Graduate Research Assistant with the Supriyo Datta Group, Purdue University. Since 2022, he has been a Postdoctoral Scholar with the Electrical and Computer Engineering Department, University of California at Santa Barbara, Santa Barbara. His research interests include probabilistic and quantum computing, hardware acceleration, modeling and simulation of nanoscale electronic devices, and machine learning.

KEREM Y. ÇAMSARI (Senior Member, IEEE) has been an Assistant Professor with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, since 2020. His Ph.D. work established a modular approach to connect a growing set of emerging materials and phenomena to circuits and systems, a framework adopted by others. In later work, he used this approach to establish the concept of p-bits and p-circuits as a bridge between classical and quantum circuits to design efficient and domain-specific hardware accelerators for the beyond-Moore era of electronics.

He is a founding member of the Technical Committee on Quantum, Neuromorphic, and Unconventional Computing with the IEEE Nanotechnology Council, where he currently leads the Unconventional Computing Section. He has received the IEEE Magnetics Society Early Career Award, the ONR Young Investigator Award, and the NSF CAREER Award for his work on probabilistic computing.

SUPRIYO DATTA (Life Fellow, IEEE) started his career in ultrasonics, but since 1985, he has been focusing on current flow in nanoscale electronic devices. The approach pioneered by his group for the description of quantum transport has been widely adopted in the field of nanoelectronics. He was elected to the National Academy of Engineering (NAE) for this work. This approach, combining the non-equilibrium green function (NEGF) formalism of many-body physics with the Landauer formalism from mesoscopic physics, is described in his books Electronic Transport in Mesoscopic Systems (Cambridge 1995), Quantum Transport: Atom to Transistor (Cambridge 2005), and Lessons from Nanoelectronics (World Scientific Second Edition 2017). He teaches online courses based on it, on nanoHUB-U and edX. He is also known for innovative theoretical proposals that have inspired new fields of research, including negative capacitance devices, spintronics, and most recently the concept of probabilistic or p-bits.

***

[41] J. D. Biamonte, “Nonperturbative K-body to two-body commuting conversion Hamiltonians and embedding problem instances into Ising spins,” Phys. Rev. A, Gen. Phys., vol. 77, no. 5, May 2008, Art. no. 052331.

[42] R. Tanburn, E. Okada, and N. Dattani, “Reducing multi-qubit interactions in adiabatic quantum computation without adding auxiliary qubits. Part 1: The ‘deduc-reduc’ method and its application to quantum factorization of numbers,” 2015, arXiv:1508.04816.

[43] M. Suzuki, “Relationship between d-dimensional quantal spin systems and (d+1)-dimensional Ising systems: Equivalence, critical exponents and systematic approximants of the partition function and spin correlations,” Prog. Theor. Phys., vol. 56, no. 5, pp. 1454–1469, Nov. 1976.

[44] S. Bravyi, D. P. Divincenzo, R. I. Oliveira, and B. M. Terhal, “The ‘deduc-reduc’ method and its application to quantum factorization of numbers,” 2015, arXiv:quant-ph/0606140.

[45] M. Van den Nest, “Efficient classical simulations of quantum Fourier transforms and normalizer circuits over Abelian groups,” Quantum Inf. Comput., vol. 13, nos. 11–12, pp. 1007–1037, Nov. 2012.

[46] S. Chowdhury, K. Y. Camsari, and S. Datta, “Accelerated quantum Monte Carlo with probabilistic computers,” Commun. Phys., vol. 6, no. 1, p. 85, Apr. 2023, doi: 10.1038/s42005-023-01202-3.

[47] (2017). The Pawsey Supercomputing Centre. [Online]. Available: https://www.pawsey.org.au/

[48] A. Tanksala and H. Batikhamene, “Quantum-kit: Simulating shor’s quantum factoring of 24-bit number on desktop,” 2019, arXiv:1908.07187.

[49] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. V. James, A. Gilchrist, and A. G. White, “Experimental demonstration of a compiled version of shor’s algorithm with quantum entanglement,” Phys. Rev. Lett., vol. 99, no. 25, Dec. 2007, Art. no. 250505.

[50] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, “Realization of a scalable shor algorithm,” Science, vol. 351, no. 6277, pp. 1068–1070, Mar. 2016.

[51] L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, “Realization of a scalable shor algorithm,” Science, vol. 351, no. 6277, pp. 1068–1070, Mar. 2016.

[52] M. Van den Nest, “Efficient classical simulations of quantum Fourier transforms and normalizer circuits over Abelian groups,” Quantum Inf. Comput., vol. 13, nos. 11–12, pp. 1007–1037, Nov. 2012.

[53] S. Chowdhury, K. Y. Camsari, and S. Datta, “Accelerated quantum Monte Carlo with probabilistic computers,” Commun. Phys., vol. 6, no. 1, p. 85, Apr. 2023, doi: 10.1038/s42005-023-01202-3.