GEYSER: A Compilation Framework for Quantum Computing with Neutral Atoms

Tirthak Patel
Northeastern University
Boston, USA

Daniel Silver
Northeastern University
Boston, USA

Devesh Tiwari
Northeastern University
Boston, USA

ABSTRACT

Compared to widely-used superconducting qubits, neutral-atom quantum computing technology promises potentially better scalability and flexible arrangement of qubits to allow higher operation parallelism and more relaxed cooling requirements. The high performance computing (HPC) and architecture community is beginning to design new solutions to take advantage of neutral-atom quantum architectures and overcome its unique challenges.

We propose GEYSER, the first work to take advantage of the multi-qubit gates natively supported by neutral-atom quantum computers by appropriately mapping quantum circuits to three-qubit-friendly physical arrangement of qubits. Then, GEYSER creates multiple logical blocks in the quantum circuit to exploit quantum parallelism and reduce the number of pulses needed to realize physical gates. These circuit blocks elegantly enable GEYSER to compose equivalent circuits with three-qubit gates, even when the original program does not have any multi-qubit gates. Our evaluation results show GEYSER reduces the number of operation pulses by 25%-90% and improves the algorithm's output fidelity by 25%-60% points across different algorithms.

CCS CONCEPTS

• **Computer systems organization → Quantum computing**.

KEYWORDS

Quantum Computing, NISQ Computing, Neutral Atoms, Rydberg Atoms, Quantum Compiling, Quantum Software

ACM Reference Format:
Tirthak Patel, Daniel Silver, and Devesh Tiwari. 2022. GEYSER: A Compilation Framework for Quantum Computing with Neutral Atoms. In *The 49th Annual International Symposium on Computer Architecture (ISCA '22), June 18–22, 2022, New York, NY, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3470496.3527428

1 INTRODUCTION

Quantum computing is progressing at a rapid pace, with multiple promising computing technologies maturing toward physical realization at the production level and in research lab settings. Superconducting, neutral atom, trapped-ion, and photonics are among the most promising technologies – each offering their own unique advantages over other competing technologies [3, 8, 21, 37]. We anticipate that multiple technologies will be in production to serve different mission needs and local-technological expertise.

Compared to widely-used superconducting qubits, neutral-atom quantum computing technology promises potentially better scalability, flexible arrangement of qubits to allow higher quantum operation parallelism, and more relaxed cooling requirements [17, 24, 25]. Unfortunately, neutral-atom quantum architecture has received limited attention from the HPC and architecture community due to its unique computing model constraints and characteristics (e.g., certain qubits are restricted from engaging in quantum operations depending on the neighbor's activity, no physical links between qubits, Rydberg transitions-driven long-distance qubit interactions, and topological constraints of qubits; discussed in Sec. 2).

Baker et al. [4] took the first step toward exploiting the power of neutral atoms by designing a solution that maps quantum circuits to a neutral-atom quantum architecture, while respecting the technological constraints including long-term qubit interactions, atom error rates, and qubit engagement restriction constraints. While this prior work is useful as a first step, an opportunity specific to neutral-atom architecture remains unexplored: *the ability to take advantage of multi-qubit operations natively supported on neutral atom architecture*, unlike superconducting-based qubit technology that only supports one- and two-qubit gate operations.

Unfortunately, taking advantage of three-or-more-qubit operations, even when natively supported on the underlying hardware, is challenging. This is because it is non-trivial for quantum programmers to reason about the program functionality and output verification for quantum programs involving three-qubit operations. GEYSER exploits this opportunity scope by creating a novel compiler optimization pass that automatically creates three-qubit operations to take advantage of neutral atom interactions, even when the original program only has one- and two-qubit operations.

As GEYSER's design and implementation discussion demonstrates, it is challenging to create three-qubit operations in a scalable fashion for a given arbitrary quantum circuit. GEYSER's solution is a three-step process, with the following essential elements and intuitions. First, it chooses a three-qubit-friendly physical arrangement of qubits and maps the circuit to it (Sec. 3.2). Then, it intelligently creates multiple blocks (self-contained set of quantum operations) in the quantum circuit, and chooses these blocks to intelligently strike the balance between maximizing quantum operation parallelism and reducing the physical pulses required to realize the quantum operations. GEYSER's design makes a special distinction about focus on reducing the number of pulses instead of the traditional metric of number of gates, due to involved trade-offs in

ISCA '22, June 18–22, 2022, New York, NY, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-8610-4/22/06…$15.00
https://doi.org/10.1145/3470496.3527428

error-probability and computational parallelism (Sec. 3.3). Finally, Geyser demonstrates its ability to create three-qubit operations by operating on smaller blocks of the original circuit where it can deterministically find the opportunity to create three-qubit gates from a set of one- and two-qubit operations and effectively reason about its impact on improving the overall output fidelity (Sec. 3.4).

The main contributions of Geyser are as follows:

- Geyser is a novel *open-source framework* to improve the output fidelity of quantum programs on neutral-atom quantum architectures, by carefully navigating the unique design trade-offs present in neutral-atom quantum technology (e.g., atom connectivity & topology, three-qubit entanglement via Rydberg transitions, and operation restriction zones): https://doi.org/10.5281/zenodo.6447912. In particular, Geyser introduces novel concepts of *circuit blocking* and *circuit composition* to improve program output fidelity by reducing the critical depth of quantum circuits and physical pulses required to realize the quantum gates.

- **Geyser is the first work to demonstrate how to opportunistically create multi-qubit operations from a set of one- and two-qubit operations when feasible, to leverage specific advantages that neutral-atom technology offers.** Geyser's compilation framework enables quantum programmers to automatically take advantage of neutral-atom architecture and significantly improve their output fidelity, even when their original programs only contain one- or two-qubit operations.

- Geyser's evaluation results show reduction in the number of quantum operation pulses by 25%-90% and improvement over algorithm's output fidelity (total variation distance) by 25%-60% points across different representative benchmarks over competitive techniques on both neutral-atom and superconducting-qubit architectures.

2 BACKGROUND

This background section is organized as a terms and definitions dictionary to best introduce the concepts relevant to the design and evaluation of Geyser. First, we briefly review generic quantum computing terms and then discuss the background relevant to neutral atom quantum computing.

2.1 Quantum Computing Principles

Qubits and Quantum States. The *state of a qubit* (short for "quantum bit") is best represented in Dirac notation: $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Here, $|\Psi\rangle \in \mathbb{C}^2$ defines the state of a qubit; where a qubit exists in a *superposition* of the two basis states: $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ and $|1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$. $\alpha \in \mathbb{C}$ and $\beta \in \mathbb{C}$ are complex-valued coefficients that represent the amplitude and phase contribution of their respective basis states to the qubit state Ψ. These coefficients must satisfy $\|\alpha\|^2 + \|\beta\|^2 = 1$. The unit vector space spanning all possible superpositions construct what is known as a *Hilbert Space*. In general, an n-qubit state can be represented by $\Psi \in \mathbb{C}^{2^n}$ under this Dirac notation.

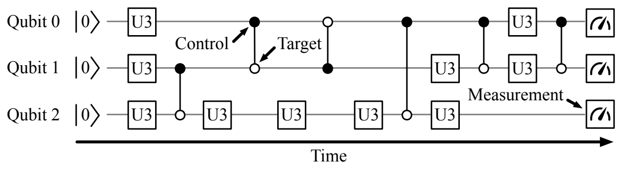


Figure 1: The example quantum circuit shows the different components of a circuit: qubits, gates, and measurements.

Quantum Operations. Desired qubit superpositions are achieved using unitary transformations known as quantum *operations or gates*. These unitary transformations can be represented as $2^n \times 2^n$–dimensional unitary matrices (applied to 2^n–dimensional vectors). The following is a general three-parameter rotational gate, U3, which can put a qubit in any desired superposition by controlling the angle parameters.

$$\texttt{U3}(\theta, \phi, \lambda) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -e^{i\lambda}\sin(\frac{\theta}{2}) \\ e^{i\phi}\sin(\frac{\theta}{2}) & e^{i(\phi+\lambda)}\cos(\frac{\theta}{2}) \end{bmatrix}$$

Here, $\theta, \phi, \lambda \in [0, 2\pi]$ are angle parameters that determine the gate that is applied to the qubit. The commonly-used *Hadamard* or $\texttt{H} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$ gate is realized as $\texttt{U3}(\frac{\pi}{2}, 0, \pi)$, while the *Identity* or $\texttt{I} = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$ gate is $\texttt{U3}(0, 0, 0)$.

Multi-qubit gates *entangle* two or more qubits. For example, the CZ and CX (also known as CNOT) gates are widely used two-qubit gates. One qubit serves as the *control* qubit and the other as the *target*, with the operation being applied to the target qubit depending on the state of the control qubit.

They are represented as following:

$$\texttt{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \ \texttt{CX(I}\otimes\texttt{H)CZ(I}\otimes\texttt{H)}$$

One of the major advantages of neutral-atom based quantum computing is its ability to natively perform three-qubit gates – *which is not feasible on superconducting-qubit architectures.* Three-qubit gate operations (e.g., CCZ) reduce the number of operations required to accomplish a task and enable more parallelism. A CCZ gate is represented as following:

$$\texttt{CCZ} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Quantum Circuits. A sequence of quantum gates applied in succession to one another on a system of n-qubits form an n-qubit *quantum circuit* (Fig. 1). A circuit can be represented as the unitary matrix $U \in \mathbb{C}^{2^n \times 2^n}$ that represents the overall unitary transformation of an n-dimensional quantum circuit. U can be calculated

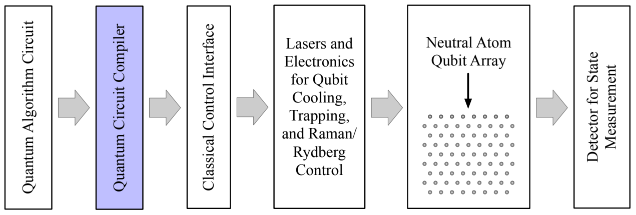

Quantum Algorithm Circuit
Quantum Circuit Compiler
Classical Control Interface
Lasers and Electronics for Qubit Cooling, Trapping, and Raman/ Rydberg Control
Neutral Atom Qubit Array
Detector for State Measurement

Figure 2: Neutral atom quantum stack consists of circuit compilation (the domain of GEYSER) and classical control interface, as well as electronics for qubit control and measurement.

by taking a Kronecker product (tensor product with respect to a standard basis) of all of the quantum gates in their respective order.

2.2 Neutral Atom Quantum Principles

Overview. Multiple quantum computing technologies (e.g., superconducting qubit, neutral atom, trapped ion, and photon) are rapidly maturing. It is likely that different manufactures will continue to pursue a wide variety of these technologies and exploit the specific technological advantage of individual technologies for different types of quantum programs. *In particular, neutral atom offers the specific advantage of being able to natively execute multi-qubit gate operations.*

Fig. 2 shows the flow of program execution on a neutral atom quantum architecture. First, the logical circuit of the quantum algorithm is compiled using a quantum compiler to reduce its size and make it hardware-compatible. The compiled circuit is then fed to a classical interface to execute the gates on a quantum computer. On the neutral atom quantum architecture, lasers are used to control the atoms: one atom = one qubit. The energy levels of the valence electron of Cesium or Rubidium atoms are used to realize qubit states [15, 38].

Unlike superconducting-qubit quantum computers, neutral atom qubits are not connected via physical links. They are entangled using Rydberg interactions (discussed next). Optical tweezers with laser cooling and trapping are used to arrange the atoms in a desired fashion [1]. Then, lasers of different wavelengths are used to prepare, control, and measure qubit states. Lastly, photo-detectors are used to measure the qubits.

Multi-Qubit Gates on Neutral Atom Architectures. One-qubit (U3) gates are applied using Raman transitions among qubit states and require only one physical light pulse [34].

On the other hand, the two-qubit CZ gate requires Rydberg transitions with three light pulses in total (shown in Fig. 3(a)) [17, 34]. Unlike Raman transitions, which are internal to an atom, Ryberg transitions depend on interactions among neighboring atoms. First, a π pulse (a light pulse with an area of π) is applied to the control qubit to knock it to the Rydberg state (an energy state with a high quantum number) if it is in the $|1\rangle$ state. This will block qubits in its vicinity to achieve the Rydberg state – a property that is used to entangle nearby qubits. Next, the nearby target qubit is supplied with a 2π pulse. Last, a π pulse is again applied to the control qubit. These three pulses help achieve the CZ gate. Similarly, the CCZ gate

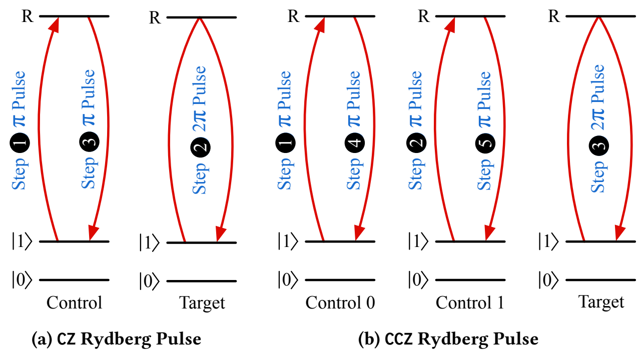


Figure 3: The pulse steps corresponding to the (a) CZ gate and (b) CCZ gate. The steps are in the order shown with the control qubits being excited with the π pulse and the target qubit being excited with the 2π pulse. The lines indicate the energy levels of the valence electron, where $|0\rangle$ is the ground state, $|1\rangle$ is the hyperfine state, and R is the Rydberg state.

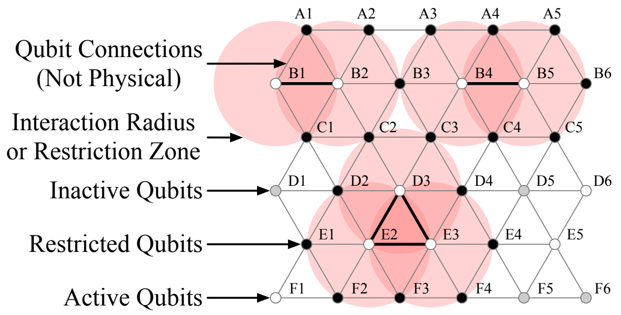


Figure 4: The example shows a snapshot of operation executions at a given moment in time. Six operations are taking place: three one-qubit (on qubits D6, E5, and F1), two two-qubits (B1-B2 and B4-B5), and one three-qubits (D3-E2-E3). The active qubits are shown in white, the restricted in black, and the inactive qubits in grey. Qubits executing multi-qubit operations are shown with the corresponding restriction zones and are connected via thick lines (e.g., the D3-E2-E3 operation restricts qubits C2, C3, D2, D4, E1, E3, F2, F3, & F4).

is achieved using five pulses using the steps show in Fig. 3(b) [17]. Note that the applied pulses are independent of the qubit state; on the other hand, the effect of the pulses is dependent on qubit states. For example, if the control qubit is in the $|0\rangle$ state, the pulses will be off-resonant and the qubit won't jump to the Rydberg state.

Interaction Radius and Restriction Zones. An important characteristic of neutral-atom quantum computing is *the interaction radius or the restriction zone.* As mentioned above, multi-qubit gates are achieved by leveraging the Rydberg interaction of atoms. However, while a multi-qubit gate is being performed, qubits that are within the Rydberg interaction radius (radius of an atom's Rydberg influence) and are not involved in the gate cannot run any other gates [4].

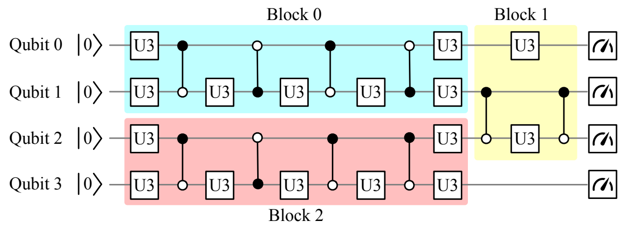


Figure 5: Example of blocking a circuit into three blocks.

Essentially, qubits that not engaged in the multi-qubit operation but are within the interaction radius (which doubles as the restriction zone) of any of the qubits which are engaged in the multi-qubit operation, are said to be restricted. This is because these non-involved qubits might inadvertently get entangled with the qubits on which the gate is being run. Thus, the interaction radius of one qubit becomes a restriction zone of other nearby non-involved qubits. This is illustrated in Fig. 4. The figure shows qubits arranged in a triangular topology. When multi-qubit gates are executing on a set of "active" qubits, other qubits within the restriction zone cannot run any other operations and therefore, become "restricted" qubits. As the figure shows, a two-qubit operation can at most restrict operations on eight nearby qubits, while a three-qubit operation can at most restrict operations on nine nearby qubits. On the other hand, a one-qubit gate can be run on qubits without generating any restriction zones as the gate does not rely on the Rydberg interaction among the qubits.

2.3 Geyser: Terms and Definitions

Circuit Mapping. *Mapping* refers to the task of taking a logical quantum circuit and converting it into a neutral atom hardware-compatible physical circuit with hardware-supported gates and qubit interactions. This is done by converting logical gates into their physical equivalents (e.g., converting CX to CZ and H as shown earlier) and inserting SWAP operations as necessary. SWAP operations help transport qubit states to enable two qubits that are connected logically, but not physically, to interact with each other. While the mapped physical circuit map have a different set of gates than the logical circuit, the two are mathematically equivalent (have the same unitary representation U).

Circuit Blocking. *A circuit block* in a quantum circuit refers to a set of self-contained quantum operations and corresponding qubits engaged in those operations. Quantum operations within a block do not interact with qubits outside the block within the span of the block (Fig. 5 illustrates circuit blocking).

A quantum circuit can be represented as a combination of multiple circuit blocks and a quantum circuit can have multiple equivalent representations in terms of circuit blocks. That is, the same quantum circuit can be represented by multiple different combinations of circuit blocks.

Each circuit block has multiple useful properties. Each quantum gate in the original quantum circuit is a part of only one circuit block. A quantum circuit can have multiple blocks over time and a qubit can be a part of multiple circuit blocks over time, but it can be a member of only one circuit block at any given time. Each circuit block can be represented by its own unitary. Consequently, a quantum circuit can be essentially represented as a sequence of circuit blocks – the original quantum circuit's unitary is the product of unitary matrices of individual blocks.

Circuit Composition. *Composing* refers to the procedure of finding a mathematically equivalent set of gates that represent a given set of gates. In general, the purpose of finding a different set of gates which is mathematically equivalent is to reduce the number of gates or pulses to accomplish the same computation [39]. Running fewer pulses reduces the noise side-effects on near-term erroneous quantum computers and results in higher output fidelity.

Circuit composition is essentially the reverse of circuit decomposition. For example, three-qubit gate can be "decomposed" into multiple single- and two-qubit gates that the underlying hardware supports [11, 27]. Neutral-atom architectures natively support three-qubit gate operations and hence, the circuit composition can reduce a set a of single- and two-qubit operations to an equivalent three-qubit operation when it's feasible and does not affect the meaning of the program. The specifics of this procedure as they pertain to Geyser are explained in detail in Sec. 3.

Hilbert-Schmidt Distance (HSD). To represent the mathematical equivalency of two circuit unitaries for circuit composition, we need a distance metric. For this purpose, we use Hilbert-Schmidt distance metric due to its lower computational overhead [20, 23] compared to other metrics[7, 22].

The Hilbert-Schmidt inner product is defined as $\mathrm{Tr}(U_1^\dagger U_2)$ between unitaries U_1 and U_2 representing the two circuits. The value of this metric is in the range $[0, 2^n]$, and the closer the value is to 2^n, the higher the equivalency of the two unitary matrices. We transform this inner product into a distance metric: $\langle U_1, U_2\rangle_{HS} = 1 - \frac{\|\mathrm{Tr}(U^\dagger U')\|}{2^n}$ and refer to it as the *Hilbert-Schmidt distance or HSD.* This distance is in the range $[0, 1]$ and the closer the value is to 0, the smaller the distance.

Total Variation Distance (TVD). To quantify the equivalency of two output distributions (e.g., calculate the deviation of the output of a circuit run on a noisy quantum computer from the ideal output), we need a distance metric for output distributions. A probability distribution distance metric that is primarily used for this is the *total variation distance or TVD* [10, 14, 32]. The TVD can be calculated as $\frac{1}{2}\sum_{k=1}^{k=2^n} |p_1(k) - p_2(k)|$, where $p_1(k)$ is the probability of state k with the first circuit and $p_2(k)$ is the probability of state k with the second circuit.

3 GEYSER: DESIGN AND IMPLEMENTATION

In this section, we present an in-depth discussion of the design and implementation of Geyser.

3.1 Overview of the Geyser Design

Geyser employs a three-step procedure toward reducing the circuit size (number of pulses) of a given quantum circuit on a neutral-atom architecture. These steps are outlined in Fig. 6. The first step in this procedure is circuit mapping. Mapping involves the procedure of converting the logical quantum circuit into physical gates

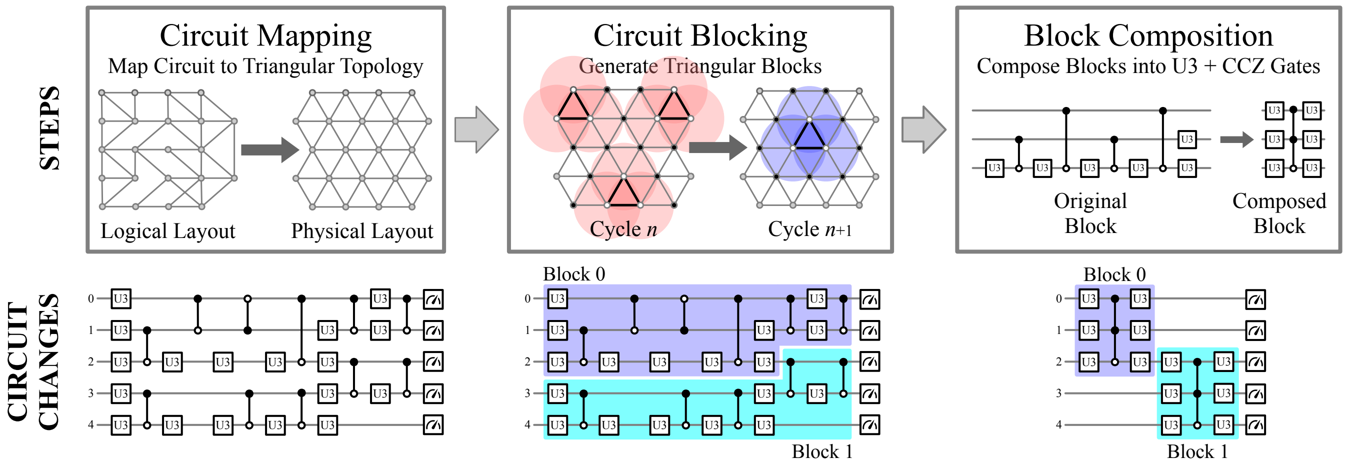


Figure 6: Overall flow of the three steps of GEYSER: circuit mapping, circuit blocking, and block composition. Circuit mapping constitutes the step of mapping a logical quantum algorithm circuit to a physical triangular topology. Circuit blocking involves creating the minimum number of concurrently-executable three-qubit blocks out of the mapped circuit. Lastly, block composition is the step to generate equivalent block circuits with direct three-qubit gates that require fewer pulses.

(U3, CZ, and CCZ) – the gates which are natively supported on the neutral-atom quantum architecture. As Sec. 3.2 shows, this step also involves mapping the logical circuit to the hardware topology of neutral atoms (i.e., arrangement of neutral atoms).

The next step is circuit blocking. Recall that circuit blocking involves finding a set of self-contained quantum operations and corresponding qubits engaged in those operations. For GEYSER, the objective of this step is two-fold: (1) find blocks that are independent of each other and hence, can be executed concurrently to maximize parallelism; and (2) find large-enough blocks. We note that it is possible to break the circuit into a set of circuit blocks in multiple possible ways (i.e., multiple configurations). In some configurations, all circuit blocks could be dependent on each other and hence, need to be executed sequentially. GEYSER attempts to maximizes the parallelism opportunity by preferring the configuration where multiple blocks can be executed in parallel. However, a competing trade-off is also the size of the circuit blocks. Larger circuit blocks allow GEYSER more opportunity for composing the gates (e.g., converting a set of single- and two-qubit gates to three-qubit gates) and hence, reducing the side effects of noise/errors. Unfortunately, large blocks limit the number of blocks that can be executed in parallel. While this trade-off space is an NP-hard problem to solve optimally, Sec. 3.3 describes a scalable method to solve it effectively.

The last step of GEYSER is the composition of circuit blocks. This step consists of generating the equivalent circuits for the blocks formed in the previous step such that the composed circuits leverage the opportunity of running three-qubit gates directly on the qubits in order to reduce the number of pulses required and resultingly, reducing the noise effects of errors in pulse application. Sec. 3.4 discusses the design and optimization of this step.

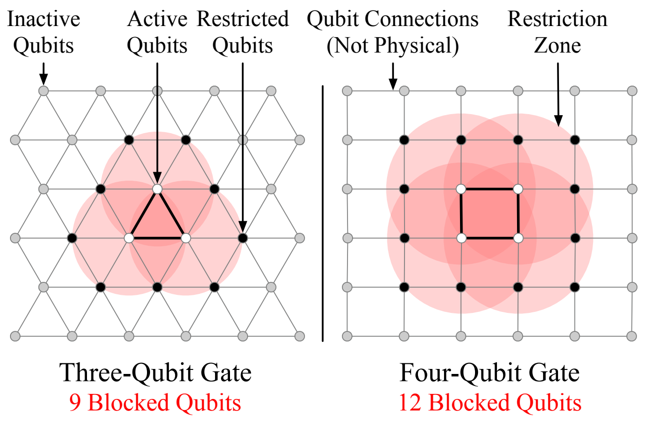


Figure 7: We select the block size of three qubits (atoms best arranged in a triangular topology) as opposed to a larger block size of four qubits (atoms best arranged in a square topology) as it results in blocks that are easier to compose and fewer blocked qubits due to the smaller restriction zones.

3.2 GEYSER: Circuit Mapping

Hardware Topology of Neutral Atoms. An attractive aspect of neutral atom quantum computing, compared to its competing technologies is that its qubits (atoms) can be arranged in any desired fashion. Prior engineering works have demonstrated the arrangement of atom in a variety of shapes (e.g., in the shape of Eiffel tower [5, 37, 40]). For practical feasibility and efficiency, we would like for the atoms to be arranged in a grid with some pattern-based distance properties to ensure Rydberg interactions. For example, Fig. 7(a) shows the qubits being arranged in a triangular grid, while Fig. 7(b) shows the qubits being arranged in a square grid.

While, the square grid arrangement appears to be the default choice in some cases, especially in superconducting-qubit architectures [44], this default choice has notable disadvantages in the context of neutral atom based quantum architecture. The qubits in a square grid are not equidistant to their neighbors: the distance of an qubit to its perpendicular neighbor is less than its distance to its diagonal neighbor. This means that the qubits need to have greater interaction distance in order to be able to interact with the diagonal neighbor. While this enables the execution of four-qubit gates (i.e., CCCZ), it also results in the larger restriction zone (compare Fig. 7(a) and 7(b)). While the triangular grid restricts nine other qubits during execution, a square grid restricts 12 qubits to run the four-qubit gate. Therefore, Geyser opts for a triangular arrangement of neutral atoms, although it can be extended to other topologies.

Another important consideration behind the choice of triangular topology is the ability to compose three-qubit gates vs. the ability to compose four-qubit gates. It is significantly more challenging to compose a four-qubit operation from a given set of single-qubit and two-qubit operations, compared to composing a three-qubit operation. Intuitively, it is harder for quantum programmers to reason about program correctness with multi-qubit gates. Quantum algorithms are rarely written with four-qubit gates (requires mathematical reasoning in terms of $2^4 \times 2^4 = 256$ components). They are also difficult to compose computationally from a sequence of one-qubit and two-qubit operations. In comparison, three-qubit blocks are 4× easier to compose because they can be represented using only 64 components as opposed to 256 components. Thus, Geyser's selection of three-qubit blocks aligns well with its design decision to use a triangular topology.

Mapping to Hardware Topology of Neutral Atoms. In terms of mapping the logical circuit to a physical topology, existing mapping solutions for superconducting-qubit quantum computers can be leveraged [2, 8, 16, 21]. Recall that the input circuits of quantum algorithms only consist of one- and two-qubit gate operations, and Geyser's circuit composition step introduces three-qubit gate operations after the circuit blocking step. Therefore, existing superconducting-qubit mapping solutions can be applied in the circuit mapping phase – although Geyser, further optimizes this process later. The details of this step are as follows.

The topology specified to the compiler simply needs to be defined as a set of qubit connections shown in Fig. 7(a). Even though these connections are not physical on neutral atom architecture, they accurately depict all possible qubit interactions in a triangular topology. In addition, the basis gates of {U3, CZ} are provided to the compiler. We use the Qiskit [2] compiler to perform this mapping as it is a state-of-the-art compiler that runs a variety of optimization passes (e.g., leveraging quantum properties to delete gates and apply heuristics for gate routing) to perform mapping with as few physical gates as possible. However, these optimizations are not sufficient for neutral atom quantum computing (as we will observe in our evaluation results in Sec. 5) because the compiler does not support circuit composition to execute multi-qubit gates. Nonetheless, it offers a reasonable framework for performing circuit mapping. Next, we discuss circuit blocking.

Algorithm 1 Algorithm pseudocode for circuit blocking.

```
1:  $Q \leftarrow$ Qubits involved in the circuit
2:  $O \leftarrow$ Original circuit operations of qubits
3:  $F \leftarrow$ Frontier operations of qubits
4:  $B \leftarrow$ Blocked circuit ($\varnothing$)
5:  while $F(q) <$ length$(O(q))$ for any $q \in Q$ do
6:      $T \leftarrow$ All possible three-qubit blocks from the qubit
7:        frontiers to as far into the circuit as possible
8:      $N \leftarrow$ Number of operations of all blocks in $T$
9:      $f \leftarrow$ Block family with most operations ($\varnothing$)
10:     for $t \in T$ do
11:         $f_t \leftarrow$ Recursively find the best block family
12:           starting at block $t$, while respecting
13:           the restriction zones of the blocks
14:         if Num. ops. in $f_t >$ Num. ops. $f$ then
15:             $f \leftarrow f_t$
16:         end if
17:     end for
18:     Add $f$ to $B$
19:     Update $F$ to last operations of blocks in $f$
20: end while
21: return B
```

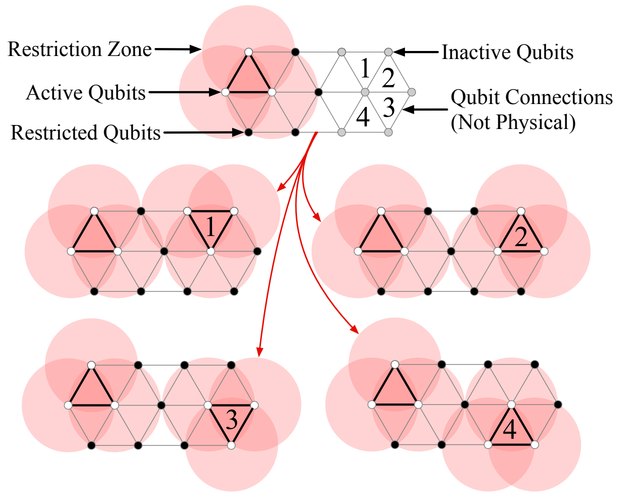


Figure 8: During a given round, the blocks are selected one-by-one to form a block family. In this example, there are four mutually-exclusive block possibilities to add to the block family after the first block is selected. As no further blocks can be selected in this example, the two-block family with the highest number of operations is finalized for the round.

3.3 Geyser: Circuit Blocking

The key goal of circuit blocking is to block the circuit in a manner that the minimum number of blocks are generated, while ensuring that as many of these blocks are executed in parallel as possible in order to reduce the circuit execution time. The algorithm that Geyser uses for this blocking procedure is provided in Algorithm 1.

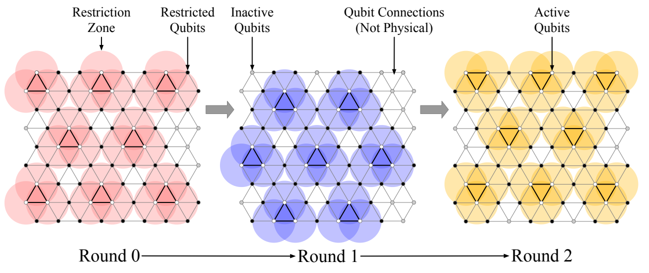


Figure 9: The example shows the round-by-round block formation procedure such that block families consisting of the maximum number of operations are formed at any given round. Blocks are formed according to their restriction zones to enable them to execute in parallel during the same round.

Algorithm 2 Algorithm pseudocode for block composition.

```
1:  $O \leftarrow$ Original block circuit
2:  $C \leftarrow$ Composed block circuit ($\varnothing$)
3:  $L \leftarrow$ Parameterized layer of U3 and CZ or CCZ
4:  $\epsilon \leftarrow$ Distance threshold
5:  while $\langle C, O\rangle_{HS} > \epsilon$ & length($C$) < length($O$) do
6:      Append $L$ to $C$
7:      Optimize parameters in $C$ to minimize $\langle C, O\rangle_{HS}$
8:        using a dual annealing-based optimizer
9:  end while
10: if length($C$) $\geq$ length($O$) then
11:     return O
12: else
13:     return C
14: end if
```

Geyser performs blocking in a pulse-aware manner. While prior work (mainly in the domain of superconducting-qubit quantum computing technology) has focused on optimizing the number of gates in the circuit [18, 26], we focus on minimizing the number of pulses required to execute these gates. This is because not all quantum gates are built similarly. Depending on the gate being run, the controller might need to execute different number of pulses. As the execution time of the circuit as well as the noise experienced by it is ultimately dependent on the number of pulses that are run to execute the circuit, having a gate-level focus may result in sub-optimal results. Recall that Fig. 3(a) shows that three serial pulses are required to execute the CZ gate and Fig. 3(b) shows that the five serial pulses are required to execute the CCZ gate. Thus, it takes 67% longer to execute the CCZ gate as compared to the CZ gate, and it may also experience that much more error. To account for this, we directly minimize the number pulses that are required to execute the circuit.

Fig. 8 illustrates the formation of blocks via maintaining a "frontier" of gates, and then, recursive construction of concurrently executable block families. This procedure is continued until the end of the circuit is reached and the entire circuit has been blocked (as demonstrated in Fig. 9). Next, we describe the final step of block composition.

3.4 Geyser: Block Composition

Block composition refers to the act of reducing the number of pulses required for a block by finding an equivalent block circuit with three-qubit gates. The composition procedure is conducted layer-by-layer as the composed circuit is built (this is demonstrated in Fig. 10). Algorithm 2 details the procedure of composing a block. Note that all formed blocks across all rounds can be independently composed in parallel. Starting with just one layer of three U3 gates (one on each qubit) and one CCZ gate and three more U3 gates, the parameters of the gates are optimized to minimize the HSD between the unitaries of the two block circuits.

As discussed in Sec. 2, the CCZ gate requires two more pulses than the CZ gate. However, the CCZ has an important use case as it is able to capture in five pulses, what would otherwise take 26 pulses to run. Fig. 11 shows the decomposition of the CCZ gate into one-qubit U3 and two-qubit CZ gates. Recall that it would not be possible to run this CCZ gate directly on a superconducting-qubit quantum computer, and thus, it would take 26 pulses to run this gate on superconducting qubits. On the other hand, neutral atom architecture makes it possible to run this gate directly in five pulses. Therefore, we use it in the layers of block composition.

The unitary corresponding to the original block circuit is referred to as the original unitary, while the one corresponding to the composed circuit is referred to as the composed unitary. If the distance between these two unitaries is minimized below a certain threshold (e.g., $1e-5$), then the two circuits can be considered equivalent. Note that the HSD is used for this process as opposed to the TVD as individual blocks do not have any output of their own. However, in Sec. 5, we evaluate the impact on the output of the entire circuit using TVD.

We start with just one layer as it has the fewest number of pulses (eleven pulses: one from each U3 and five from CCZ). Both, U3 and CCZ are parameterized gates as shown in Fig. 10: U3 has the three rotational angles, while CCZ has the option to decide the target qubit among the three qubits (19 parameters in total). Note that the two form a universal gate set. For each configuration in the parameter space, there exits a corresponding circuit with the corresponding unitary matrix, whose distance can be calculated to the original unitary. These parameters are optimized to calculate and minimize this distance using a dual annealing process [36]. If the distance goes below the desired threshold, the parameter configuration is chosen to represent the composed circuit.

On the other hand, if a distance below the desired threshold is not obtained at the end of the dual annealing search procedure, another layer is added, allowing an optimization space of 29 parameters in total. This can help achieve a smaller distance, while also increasing the pulse count. If still the distance threshold is not met, then a third layer is added. This process continues until either the distance threshold is achieved or the pulse count of the composed circuit exceeds the pulse count of the original circuit (in which case, it is better to use the original block circuit for the final execution). Once all blocks are composed, the full composed circuit is formed by putting the composed blocks together. This composed circuit requires fewer pulses than the original mapped circuit and reduces the execution time as well as the noise effects as we discuss in the

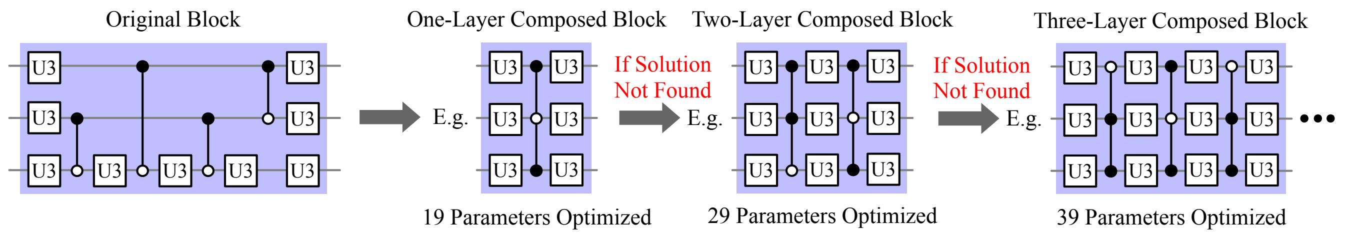


Figure 10: A block is composed layer-by-layer. Starting with just one layer, the 19 parameters ($6 \times 3 = 18$ angles between 0 and 2π from the U3 gates and 1 categorical parameter to choose among the three CCZ configurations) are optimized to find a block with a close Hilbert-Schmidt distance to the original block. If a solution is not found with one layer, another layer is added and a 29-parameter optimization is performed. The procedure is continued until a solution is found or the number of pulses in the composed block surpasses the number of pulses in the original block.

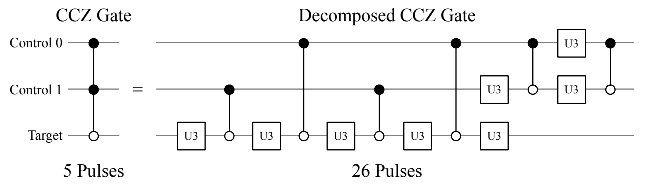


Figure 11: Breakdown of the three-qubit CCZ gate into the one-qubit U3 and two-qubit CZ gates. The direct CCZ gate requires only five pulses as shown in Fig. 3(b), while the decomposed version requires 26 pulses in total: $3 \times 6 = 8$ from the six CZ gates and $1 \times 8 = 18$ from the eight U3 gates.

evaluation section (Sec. 5). Geyser also has a low overhead and scales efficiently as discussed in Sec. 6.

4 EXPERIMENTAL METHODOLOGY

Experimental Setup. Geyser evaluation is primarily driven via simulation using representative characteristics and technological parameters of neutral-based quantum architectures including interaction radius and atom distance [17, 44]. Note that neutral-atom architectures are in a promising exploration phase, and are advancing toward the path of maturity, but more simulation-based exploration is required to identify and emphasize its usefulness over competing technologies – similar to classical architectural innovation pathways.

We run all Geyser steps on a local data center consisting of Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz nodes. Each node has two sockets, each with 12 physical cores (48 logical cores total), and a memory capacity of 128 GiB. We use the Python 3.8.8 framework to execute all the steps involved in Geyser. Because the circuit blocks can be scored and composed in parallel without any dependencies, we use multiprocessing to score and compose the blocks concurrently.

We use Qiskit 0.18.3 [2] to perform the mapping step as described in Sec. 3.2. We use the `qiskit-aer 0.9.1` library to perform unitary simulation to calculate HSD during composition. We use the `qiskit-ibmq-provider 0.18.0` library to simulate noisy circuits in the IBMQ QASM simulator in the IBMQ cloud. Our evaluation uses representative one-qubit and multi-qubit gate errors on par with the state-of-the-art neutral atom implementations [33, 35, 44]. The noise model includes both bit-flip and phase-flip errors with 0.1% occurrence rate on one-qubit operations. The one-qubit error matrix is then self-tensored to generate two-qubit and three-qubit error matrices. To further demonstrate the robustness of Geyser, we also present evaluation results corresponding to error rates 0.05% and 0.5%. Lastly, the `scipy 1.6.2` [19] library is used to optimize (minimize) the HSD during composition using dual annealing.

Comparative Techniques. Baseline: The "Baseline" technique includes with mapping and scheduling a given circuit on to a triangular topology for execution on neutral atom architecture. It does not include any mapping optimizations. **OptiMap:** In addition to the steps taken in the Baseline technique, the OptiMap technique includes all state-of-the-art optimizations that are performed by Qiskit [2], including gate cancellation and gate synthesis. **Geyser:** In addition to the steps in the OptiMap technique, Geyser performs blocking and composition as described in Sec. 3.

Superconducting Qubits: We also compare to running the circuit on a superconducting-qubit architecture. This comparison is performed to evaluate the neutral-atom architecture's relative potential compared to the superconducting-qubit architecture. We caution that while this comparison is useful, it is sensitive to technological advances for both architecture types. To provide a more favorable comparison for superconducting-qubit architecture, we use the same noise levels as we do for neutral-atom architectures and use all of the Qiskit mapping optimizations, although some recent physics studies provide evidence that neutral-atom based architecture might be able to achieve lower error rate in a more relaxed cooling environment than superconducting qubits [29].

Furthermore, we map the circuit on to a square grid in order to simulate a best-case scenario for superconducting-qubits – typically, superconducting qubits are laid out in a hexagonal grid to minimize interference [31], which results in fewer qubits connections than square grid and requires more SWAP gates.

Benchmarks. Table 1 shows the benchmarks used to evaluate Geyser, along with the gate counts and pulse counts in their Baseline circuits. Quantum Alternating Operator Ansatz (QAOA) and Variational Quantum Eigensolver (VQE) are variational quantum algorithms. Adder and Multiplier are quantum arithmetic circuits.

Table 1: Benchmarks used to evaluate Geyser.

Benchmark	Num. Qubits	Num. U3 Gates	Num. CZ Gates	Num. Total Pulses	Num. Depth Pulses
Adder [9]	4	75	24	147	117
VQE [28]	4	235	74	457	359
QAOA [12]	5	123	48	267	212
QFT [30]	5	113	39	230	167
Multiplier [16]	5	75	23	144	104
Adder	5	380	158	854	605
Advantage [3]	9	108	32	204	73
QFT	10	1141	498	2635	1629
Multiplier	10	787	340	1807	1136
Heisenberg [6]	16	15614	3339	25631	8083

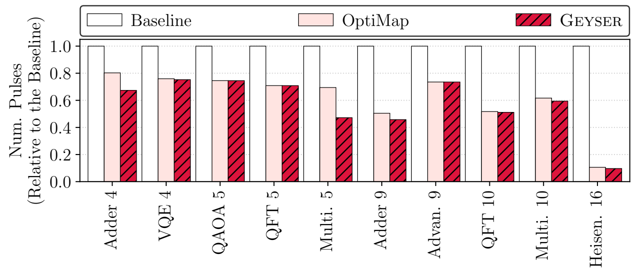


Figure 12: Geyser reduces the total number of pulses in the circuit compared to the Baseline and OptiMap techniques.

QFT is Quantum Fourier Transform and Heisenberg is a Hamiltonian evolution algorithm for material simulations. These algorithms cover a wide range of circuit characteristics [3, 6, 9, 12, 16, 28, 30].

Evaluation Metrics. Number of Operations *(lower is better)*: This metric calculates the number of gates of different types (U3, CZ, and CCZ) in the final circuit generated by different techniques. **Number of Pulses *(lower is better)*:** This is the total number of pulses in the circuit. Recall that we directly optimize the number of pulses as opposed to operations because noise effects are proportional to pulses and different operations can have different pulse counts. **Number of Depth Pulses *(lower is better)*:** This is the number of pulses in the critical path of the circuit (the path with the highest depth in terms of the number of pulses). **Total Variation Distance (TVD) *(lower is better)*:** As described in Sec. 2, the TVD of two outputs can be calculated as $\frac{1}{2}\sum_{k=1}^{k=2^n}|p_1(k) - p_2(k)|$, where $p_1(k)$ is the probability of state k in the first output and $p_2(k)$ is the probability of state k in the second output. We use this metric to evaluate the output fidelity of different solutions by calculating the TVD to the ideal output.

5 EVALUATION AND ANALYSIS

Geyser reduces the total number of pulses and the number of depth pulses in the critical path compared to other competitive techniques. Fig. 12 shows the number of pulses for different algorithms when the circuit is compiled using the Baseline, OptiMap, and Geyser techniques. Similarly, Fig. 13 shows the number of depth pulses in the critical path for different algorithms when the circuit is compiled using the three techniques. Across different algorithms, Geyser achieves a higher reduction in pulse count from the Baseline as compared to OptiMap. As an instance, for the 16-qubit Heisenberg algorithm, the Baseline circuit has a total of 25,632 pulses and 8,083 depth pulses. The OptiMap technique reduces the total pulse count by 90% (2,716 pulses) and the depth pulse count by 82% (1,403). While not evident from the figure, Geyser reduces the total pulse count by a further 9% (2,483) and the depth pulse count by a further 14% compared to OptiMap (1,212). Table 1 denotes pulse and gate count achieved via Geyser and OptiMap. These results indicate the effectiveness of the blocking and composition procedures of Geyser. Next, we detail the reasons behind this effectiveness.

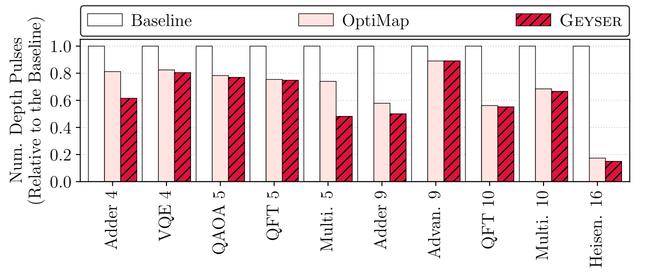
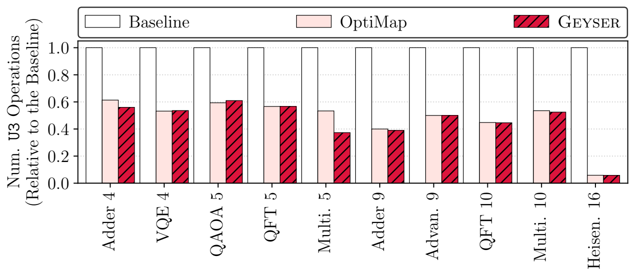


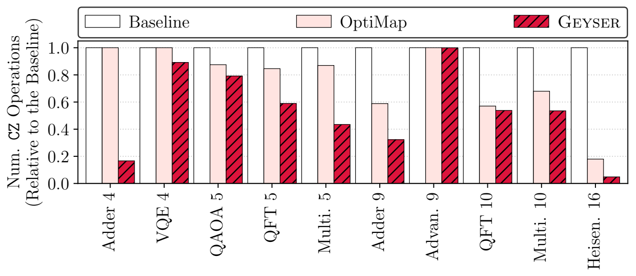
Figure 13: Geyser reduces the number of pulses in the critical path of the circuit compared to Baseline and OptiMap.

Geyser achieves reductions in pulse counts by introducing few CCZ gates in place of large numbers of U3 and CZ gates. Fig. 14(a), Fig. 14(b), and Fig. 14(c) show the number of U3, CZ, and CCZ gates with the three techniques, respectively. As expected, the Baseline circuits have a high number of U3 and CZ gates and no CCZ gates. In comparison, due to the gate optimizations, the OptiMap circuit have a lower U3 and CZ count, but still no CCZ gates. Note that while Mapping reduce U3 gates considerably, it is not able to reduce the CZ gates as efficiently because two-or-more-qubit gates are difficult to optimize without composition. For example, for the 4-qubit Adder circuit, Mapping reduces U3 gates by 40% (0.6 vs. 1), but it does not reduce CZ gates at all. On the other hand, Geyser is able to further reduce the U3 and CZ gate counts due to the use of the CCZ gates during composition.

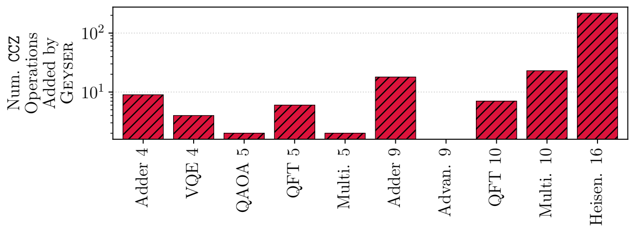
The CCZ gates are able to capture the same amount of quantum computational complexity as combinations of the other two gates, but using fewer pulses. Thus, one CCZ gate can replace a relatively long sequence of U3 and CZ gates. However, this opportunity may not be possible for a given algorithm as it relies on the ability to form long blocks. If this is not achievable for a given algorithm (e.g., the 9-qubit Advantage algorithm), then Geyser does not provide improvements over the OptiMap technique. On the other hand, for the 5-qubit Multiplier algorithm with long blocks, as compared to the Baseline technique, Geyser is able to reduce U3 gate count by 63% and CZ gate count by 57%, while introducing two CCZ gates: the U3 pulse count is reduced from 75 to 28, the CZ pulse count is reduced from 69 to 30, and the CCZ gates introduce 10 pulses, resulting in a total reduction of 76 pulses. Next, we study how this reduces output fidelity.



(a) Number of U3 Operations



(b) Number of CZ Operations



(c) Number of CCZ Operations

Figure 14: GEYSER reduces the number of U3 and CZ gates compared to Baseline and OptiMap by leveraging CCZ gates to perform complex quantum computations with fewer pulses. Note the the raw count is provided for CCZ gates on a *log* scale as Baseline and OptiMap do not have these gates.

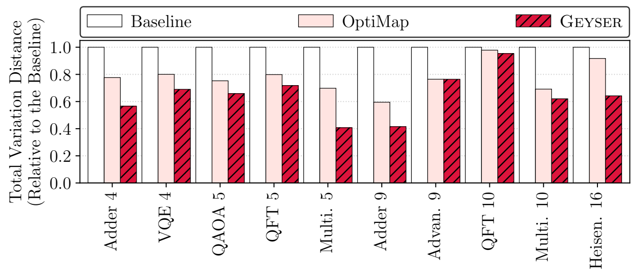


Figure 15: The TVD to the ideal output when the circuit is compiled with the Baseline, OptiMap, and GEYSER techniques. The TVD is lowest with GEYSER.

GEYSER achieves a lower TVD to the ideal circuit output (i.e., higher output fidelity) compared to other techniques. Fig. 15 compares the TVDs (to the ideal output) of the Baseline, OptiMap, and GEYSER-generated circuits simulated with the same noise characteristics. The results show the tangible benefits of reducing the number of pulses and the number of pulses in the critical path as the TVD with GEYSER is lower than the TVD with comparative techniques. For the 9-qubit Advantage algorithm, which was not able to leverage the composition benefits due to small block size, the TVD with GEYSER is similar to the TVD with OptiMap. But in other cases, GEYSER improves TVD in the range of 25-60%. For example, for the 5-qubit QFT algorithm, OptiMap improves the TVD over Baseline by 21% and GEYSER improves the TVD over Baseline by 32%. Further, GEYSER improves the TVD by over 60% for the 5-qubit Multiplier algorithm. In general, OptiMap has a lower TVD than Baseline as it has fewer pulses, while GEYSER has even lower TVD due to pulse reduction as a result of block composition.

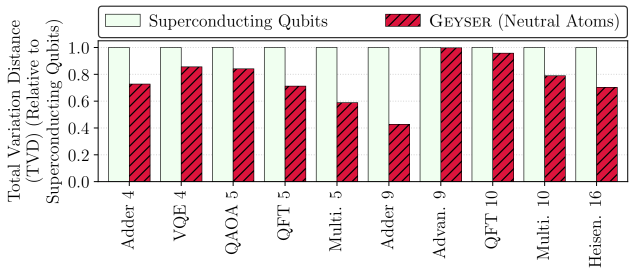
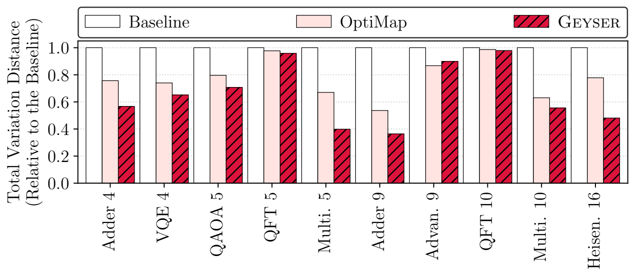


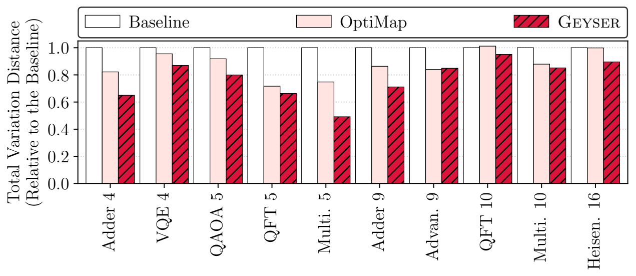
Figure 16: The TVD to the ideal output when the circuit is run on superconducting qubits vs. when it is run on neutral atoms with GEYSER. The TVD is lower on neutral atoms.

Simulating GEYSER with a neutral atom circuit yields a lower TVD to the ideal output than the superconducting analog built with corresponding error rates. Fig. 16 shows that a circuit run on superconducting qubits with all of the state-of-the-art Qiskit optimizations has a much higher TVD than the corresponding circuit run with GEYSER on neutral atoms. For example, for the 9-qubit Adder circuit, the TVD with GEYSER applied for neutral atoms is 57% lower than it is when the circuit is run on superconducting qubits with the same operation errors. This is due to the fact that superconducting-qubit technology is inherently not capable of executing multi-qubit (greater than two qubits) operations. This prevents the circuits run on this technology to be limited in terms of the optimizations that can be performed on them – block composition cannot be performed on superconducting qubits. These results demonstrate the useful benefits of the neutral-atom technology over superconducting qubits.

GEYSER achieves reduction in TVD compared to competing techniques and superconducting-qubit architectures for different noise levels. Fig. 17 and Fig. 18 show how the behavior of GEYSER is affected is the noise level is increased to 0.5% of decreased to 0.05% (compared to the default noise level of 0.1%). The results show that GEYSER's benefits will remain high for future qubits with lower errors as well. For example, GEYSER achieves a 60% reduction in TVD compared to Baseline even if the noise level is decreased to 0.05%. This demonstrates that GEYSER's performance characteristics are not restrictively sensitive to the noise levels.

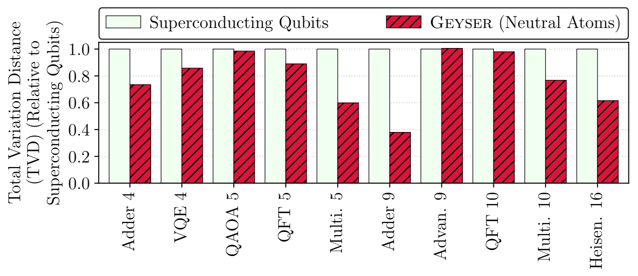


(a) Error Rate 0.05%

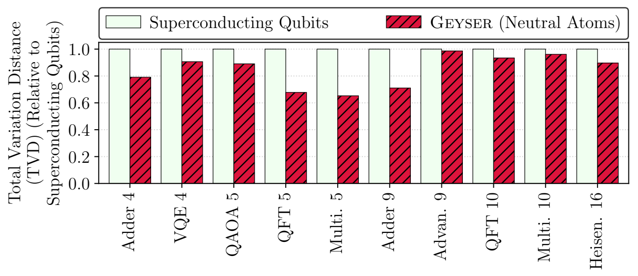


(b) Error Rate 0.5%

Figure 17: The TVD to the ideal output when the circuit is compiled with the Baseline, OptiMap, and GEYSER techniques for different error rates. The TVD is lowest with GEYSER.



(a) Error Rate 0.05%



(b) Error Rate 0.5%

Figure 18: The TVD to the ideal output when the circuit is run on superconducting qubits vs. when it is run on neutral atoms with GEYSERfor different error rates.

6 DISCUSSION

GEYSER Circuit Fidelity. Recall that GEYSER tries to emulate the original circuit's unitary as much as possible during the composition step by minimizing the HSD. However, because the HSD is still non-zero, it is important for the circuit generated by GEYSER to give similar output to the output of the original circuit even in the ideal scenario. This is indeed the case because the TVD between the ideal output of GEYSER's circuit and the ideal output of the original circuit is practically negligible ($< 1e - 2$) across all algorithms.

Neutral Atom Loss. Neutral atoms can sometimes be knocked out of place when the atom state is being measured [4, 13]. However, the technology to control atom positions and measure them without atom loss has developed considerably in the recent years, resulting in a decreased likelihood of losing an atom [13, 17]. Moreover, lost atoms can easily be replaced by shuttling other unused atoms in their place; atom can be rearranged to realize a loss free register using a take $\longrightarrow$ transfer $\longrightarrow$ release procedure with optical tweezers that load and process a neutral atom computational cycle [17]. GEYSER leverages this physically demonstrated capability to mitigate atom loss between shots, and its effectiveness was not experimentally observed to be sensitive for realistic atom loss probabilities [13, 25].

Scalability of the GEYSER Procedure. We discuss both the time and space complexity of GEYSER. Overall, GEYSER scales quadratically with the number of circuit operations (c) as described next. The first step of mapping has a time complexity of $O(kc)$, when k mapping optimization passes are performed. We use the Qiskit [2] compiler for this steps, and it finishes within a minute across all algorithms on our experimental testbed. The next step of blocking has a worse-case time complexity of $O(c^2)$ as each block is scored, and then, the block family is formed recursively (c blocks are formed in the worst case, although this number is typically much smaller in practice). This step can be performed in parallel for all blocks and also finishes within a minute. The last step of composition has a time complexity of $O(c)$, as each block is composed once and the composition of all blocks can be performed concurrently. We observe that the time varies based on how many layers need to be added based on the complexity of the block unitary matrix. However, the parameter optimization after each layer addition completes within two minutes on our nodes. The space complexity is proportional to the number of blocks – bounded by $O(c)$.

7 RELATED WORK

In this section, we discuss previous works on quantum circuit compilation and optimization. Specifically, in [18, 26, 42, 43], a similar goal is present of recompiling quantum circuits to be more space efficient in order to speed up execution and reduce noise. However, these works are focused on superconducting quantum circuits, while GEYSER is designed for neutral atom circuits to mitigate neutral atom specific challenges and exploit specific advantages that neutral atom technology offers. Note that many of the techniques used to optimize superconducting circuits, which include minimizing additional swap gates [41] are also applicable and useful in optimizing neutral atom circuits (as incorporated in the circuit mapping step of GEYSER, which uses Qiskit [2] optimization passes).

However, these works do not account for the flexible geometries of neutral atom circuits, the Rydberg interaction blocking, and the native multi-qubit gate support.

In contrast to compilers for superconducting qubits, Baker et al. [4] demonstrate how to compile quantum circuits for neutral atom architectures. The technique relies on the traditional superconducting-qubit methods for mapping and scheduling of quantum circuits given the rules of neutral atom interactions. This technique is comparable to the Baseline technique in our evaluation, as it does not have the optimization passes included in OptiMap. In contrast, GEYSER moves the state-of-art of neutral atom compilation and optimization by introducing novel methods for pulse count reduction, composing three-qubit gates, and operation parallelism.

Finally, we note that previous works have made attempts to decompose larger gates into smaller gates [39]. In contrast, GEYSER is the first work to solve the inverse problem – that, composing larger gates from smaller gates (e.g., composing three-qubit gates from a set of single- and two-qubit gates, when possible). This is a more challenging optimization problem than decomposing to smaller gates, and a solution to this problem has broader applicability in quantum computing in addition to neutral-atom quantum circuit compilation.

8 CONCLUSION

Neutral-atom architectures are one of the more promising quantum computing technologies. However, they has received limited attention from our architecture community and require more exploration to amplify their advantages and build a system software ecosystem around them. GEYSER is designed to take advantage of the quantum technology of neutral-atom computation using multi-qubit gates. Using intelligent circuit mapping and novel circuit blocking and composition, GEYSER is able to build circuits with fewer pulses, reducing the output error by up to 25%-60%. GEYSER's compilation and experimental framework is open-sourced at: https://doi.org/10.5281/zenodo.6447912.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive feedback. This work was also supported by Northeastern University, NSF Award 1910601 and 2144540. IBM Q was also used for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Q team.

REFERENCES

[1] C S Adams, J D Pritchard, and J P Shaffer. 2019. Rydberg Atom Quantum Technologies. (July 2019). arXiv:1907.09231 [physics.atom-ph]

[2] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, Francisco Jose Cabrera-Hernández, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, Jerry M. Chow, Antonio D. Córcoles-Gonzales, Abigail J. Cross, Andrew Cross, Juan Cruz-Benito, Chris Culver, Salvador De La Puente González, Enrique De La Torre, Delton Ding, Eugene Dumitrescu, Ivan Duran, Pieter Eendebak, Mark Everitt, Ismael Faro Sertage, Albert Frisch, Andreas Fuhrer, Jay Gambetta, Borja Godoy Gago, Juan Gomez-Mosquera, Donny Greenberg, Ikko Hamamura, Vojtech Havlicek, Joe Hellmers, Łukasz Herok, Hiroshi Horii, Shaohan Hu, Takashi Imamichi, Toshinari Itoko, Ali Javadi-Abhari, Naoki Kanazawa, Anton Karazeev, Kevin Krsulich, Peng Liu, Yang Luh, Yunho Maeng, Manoel Marques, Francisco Jose Martín-Fernández, Douglas T. McClure, David McKay, Srujan Meesala, Antonio Mezzacapo, Nikolaj Moll, Diego Moreda Rodríguez, Giacomo Nannicini, Paul Nation, Pauline Ollitrault, Lee James O'Riordan, Hanhee Paik, Jesús Pérez, Anna Phan, Marco Pistoia, Viktor Prutyanov, Max Reuter, Julia Rice, Abdón Rodríguez Davila, Raymond Harry Putra Rudy, Mingi Ryu, Ninad Sathaye, Chris Schnabel, Eddie Schoute, Kanav Setia, Yunong Shi, Adenilton Silva, Yukio Siraichi, Seyon Sivarajah, John A. Smolin, Mathias Soeken, Hitomi Takahashi, Ivano Tavernelli, Charles Taylor, Pete Taylour, Kenso Trabing, Matthew Treinish, Wes Turner, Desiree Vogt-Lee, Christophe Vuillot, Jonathan A. Wildstrom, Jessica Wilson, Erick Winston, Christopher Wood, Stephen Wood, Stefan Wörner, Ismail Yunus Akhalwaya, and Christa Zoufal. 2019. *Qiskit: An Open-source Framework for Quantum Computing.* https://doi.org/10.5281/zenodo.2562111

[3] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. 2019. Quantum Supremacy using a Programmable Superconducting Processor. *Nature* 574, 7779 (2019), 505–510. https://doi.org/10.1038/s41586-019-1666-5

[4] Jonathan M Baker, Andrew Litteken, Casey Duckering, Henry Hoffmann, Hannes Bernien, and Frederic T Chong. 2021. Exploiting Long-Distance Interactions and Tolerating Atom Loss in Neutral Atom Quantum Architectures. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA).* 818–831.

[5] Daniel Barredo, Vincent Lienhard, Sylvain de Léséleuc, Thierry Lahaye, and Antoine Browaeys. 2018. Synthetic Three-Dimensional Atomic Structures Assembled Atom by Atom. *Nature* 561, 7721 (Sept. 2018), 79–82.

[6] Lindsay Bassman, Connor Powers, and Wibe A de Jong. 2021. ArQTiC: A Full-Stack Software Package for Simulating Materials on Quantum Computers. *arXiv preprint arXiv:2106.04749* (2021).

[7] Ethan Bernstein and Umesh Vazirani. 1997. Quantum Complexity Theory. *SIAM Journal on computing* 26, 5 (1997), 1411–1473.

[8] Rigetti Computing. 2019. Pyquil documentation. *URL http://pyquil.readthedocs.io/en/latest* (2019).

[9] Steven A Cuccaro, Thomas G Draper, Samuel A Kutin, and David Petrie Moulton. 2004. A New Quantum Ripple-Carry Addition Circuit. *arXiv preprint quant-ph/0410184* (2004).

[10] Megan L Dahlhauser and Travis S Humble. 2021. Modeling Noisy Quantum Circuits using Experimental Characterization. *Physical Review A* 103, 4 (2021), 042603.

[11] Casey Duckering, Jonathan M Baker, Andrew Litteken, and Frederic T Chong. 2021. Orchestrated Trios: Compiling for Efficient Communication in Quantum Programs with 3-Qubit Gates. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems.* 375–385.

[12] Edward Farhi and Aram W Harrow. 2016. Quantum Supremacy through the Quantum Approximate Optimization Algorithm. *arXiv preprint arXiv:1602.07674* (2016).

[13] A Fuhrmanek, R Bourgain, Yvan RP Sortais, and Antoine Browaeys. 2011. Free-space lossless state detection of a single trapped atom. *Physical review letters* 106, 13 (2011), 133003.

[14] Alexei Gilchrist, Nathan K Langford, and Michael A Nielsen. 2005. Distance Measures to Compare Real and Ideal Quantum Processes. *Physical Review A* 71, 6 (2005), 062310.

[15] T M Graham, M Kwon, B Grinkemeyer, Z Marra, X Jiang, M T Lichtman, Y Sun, M Ebert, and M Saffman. 2019. Rydberg-Mediated Entanglement in a Two-Dimensional Neutral Atom Qubit Array. *Phys. Rev. Lett.* 123, 23 (Dec. 2019), 230501.

[16] Andrew Hancock, Austin Garcia, Jacob Shedenhelm, Jordan Cowen, and Calista Carey. 2019. Cirq: A Python Framework for Creating, Editing, and Invoking Quantum Circuits. *URL https://github.com/quantumlib/Cirq* (2019).

[17] Loic Henriet, Lucas Beguin, Adrien Signoles, Thierry Lahaye, Antoine Browaeys, Georges-Olivier Reymond, and Christophe Jurczak. 2020. Quantum Computing with Neutral Atoms. (June 2020). arXiv:2006.12326 [quant-ph]

[18] Toshinari Itoko, Rudy Raymond, Takashi Imamichi, and Atsushi Matsuo. 2020. Optimization of Quantum Circuit Mapping using Gate Transformation and Commutation. *Integration* 70 (2020), 43–50.

[19] Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2016. SciPy: Open Source Scientific Tools for Python, 2001.

[20] Sumeet Khatri, Ryan LaRose, Alexander Poremba, Lukasz Cincio, Andrew T Sornborger, and Patrick J Coles. 2019. Quantum-Assisted Quantum Compiling. *Quantum* 3 (2019), 140.

[21] Nathan Killoran, Josh Izaac, Nicolás Quesada, Ville Bergholm, Matthew Amy, and Christian Weedbrook. 2019. Strawberry fields: A software platform for photonic quantum computing. *Quantum* 3 (2019), 129.

[22] Alexei Yu Kitaev, Alexander Shen, Mikhail N Vyalyi, and Mikhail N Vyalyi. 2002. *Classical and Quantum Computation.* Number 47. American Mathematical Soc.

[23] Vadym Kliuchnikov, Alex Bocharov, and Krysta M Svore. 2014. Asymptotically Optimal Topological Quantum Compiling. *Physical review letters* 112, 14 (2014), 140504.

[24] Harry Levine, Alexander Keesling, Ahmed Omran, Hannes Bernien, Sylvain Schwartz, Alexander S Zibrov, Manuel Endres, Markus Greiner, Vladan Vuletić, and Mikhail D Lukin. 2018. High-Fidelity Control and Entanglement of Rydberg-Atom Qubits. *Phys. Rev. Lett.* 121, 12 (Sept. 2018), 123603.

[25] Harry Levine, Alexander Keesling, Giulia Semeghini, Ahmed Omran, Tout T Wang, Sepehr Ebadi, Hannes Bernien, Markus Greiner, Vladan Vuletić, Hannes Pichler, and Mikhail D Lukin. 2019. Parallel Implementation of High-Fidelity Multi-Qubit Gates with Neutral Atoms. (Aug. 2019). arXiv:1908.06101 [quant-ph]

[26] Gushu Li, Yufei Ding, and Yuan Xie. 2019. Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems.* ACM, 1001–1014.

[27] Dmitri Maslov and Yunseong Nam. 2018. Use of Global Interactions in Efficient Quantum Circuit Constructions. *New Journal of Physics* 20, 3 (2018), 033018.

[28] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. 2016. The Theory of Variational Hybrid Quantum-Classical Algorithms. *New Journal of Physics* 18, 2 (2016), 023023.

[29] Harold J Metcalf and Peter van der Straten. 2003. Laser Cooling and Trapping of Atoms. *JOSA B* 20, 5 (2003), 887–908.

[30] Victor Namias. 1980. The Fractional order Fourier Transform and its Application to Quantum Mechanics. *IMA Journal of Applied Mathematics* 25, 3 (1980), 241–265.

[31] Paul Nation, Hanhee Paik, Andrew Cross, and Zaira Nazario. 2021. The IBM Quantum Heavy Hex Lattice. https://research.ibm.com/blog/heavy-hex-lattice Accessed: 2021-11-22.

[32] Tirthak Patel, Abhay Potharaju, Baolin Li, Rohan Roy, and Devesh Tiwari. 2020. Experimental Evaluation of NISQ Quantum Computers: Error Measurement, Characterization, and Implications. In *2020 SC20: International Conference for High Performance Computing, Networking, Storage and Analysis (SC).* IEEE Computer Society, 636–650.

[33] Gerard Pelegrí, Andrew J Daley, and Jonathan D Pritchard. 2021. High-Fidelity Multiqubit Rydberg Gates via Two-Photon Adiabatic Rapid Passage. *arXiv preprint arXiv:2112.13025* (2021).

[34] C J Picken, R Legaie, K McDonnell, and J D Pritchard. 2018. Entanglement of Neutral-atom Qubits with Long Ground-Rydberg Coherence Times. *Quantum Sci. Technol.* 4, 1 (Dec. 2018), 015011.

[35] M Saffman, II Beterov, A Dalal, EJ Páez, and BC Sanders. 2020. Symmetric Rydberg Controlled-Z Gates with Adiabatic Pulses. *Physical Review A* 101, 6 (2020), 062309.

[36] Kemal H Sahin and Amy R Ciric. 1998. A Dual Temperature Simulated Annealing Approach for Solving Bilevel Programming Problems. *Computers & chemical engineering* 23, 1 (1998), 11–25.

[37] Xiao-Feng Shi. 2020. Single-site Rydberg Addressing in 3D Atomic Arrays for Quantum Computing with Neutral Atoms. *J. Phys. B At. Mol. Opt. Phys.* 53, 5 (Jan. 2020), 054002.

[38] N Šibalić, J D Pritchard, C S Adams, and K J Weatherill. 2016. ARC: An Open-Source Library for Calculating Properties of Alkali Rydberg Atoms. (Dec. 2016). arXiv:1612.05529 [physics.atom-ph]

[39] Robert R Tucci. 2005. An introduction to Cartan's KAK decomposition for QC programmers. *arXiv preprint quant-ph/0507171* (2005).

[40] Yang Wang, Aishwarya Kumar, Tsung-Yao Wu, and David S Weiss. 2016. Universal Gates based on Targeted Phase Shifts in a 3D Neutral Atom Array. (Jan. 2016). arXiv:1601.03639 [quant-ph]

[41] Robert Wille, Lukas Burgholzer, and Alwin Zulehner. 2019. Mapping Quantum Circuits to IBM QX Architectures Using the Minimal Number of SWAP and H Operations. In *Proceedings of the 56th Annual Design Automation Conference 2019.* ACM, 142.

[42] Robert Wille, Oliver Keszocze, Marcel Walter, Patrick Rohrs, Anupam Chattopadhyay, and Rolf Drechsler. 2016. Look-ahead schemes for nearest neighbor optimization of 1D and 2D quantum circuits. In *2016 21st Asia and South Pacific design automation conference (ASP-DAC).* IEEE, 292–297.

[43] Robert Wille, Aaron Lye, and Rolf Drechsler. 2014. Exact reordering of circuit lines for nearest neighbor quantum architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 33, 12 (2014), 1818–1831.

[44] T Xia, M Lichtman, K Maller, A W Carr, M J Piotrowicz, L Isenhower, and M Saffman. 2015. Randomized Benchmarking of Single-qubit Gates in a 2D Array of Neutral-atom Qubits. *Phys. Rev. Lett.* 114, 10 (March 2015), 100503.