


# Efficient Quantum Simulation for Translationally Invariant Systems

Joris Kattemölle<sup>1</sup> and Guido Burkard<sup>1</sup>

*Department of Physics, University of Konstanz, D-78457 Konstanz, Germany*

 (Received 15 April 2025; revised 30 November 2025; accepted 4 December 2025; published 6 January 2026)

Discrete translational symmetry plays a fundamental role in condensed matter physics and lattice gauge theories, enabling the analysis of systems that would otherwise be intractable. Despite this, many open problems remain. Quantum simulation promises to offer new insights, but progress is often limited by device connectivity constraints, which lead to prohibitively long computation times. We extend the use of spatial symmetry from the systems to be simulated to the quantum circuits simulating them. One application is that it becomes possible to efficiently and optimally alleviate device connectivity constraints algorithmically. This leads to reductions in quantum computational time by several orders of magnitude even for moderate system sizes, making such simulations feasible, with even greater relative gains for larger systems. This substantially enhances the capabilities of quantum computers in the simulation of condensed matter systems and lattice gauge theories, even before hardware improvements. Our Letter forms the basis for using spatial symmetry of quantum circuits in other areas of quantum computation, such as in the design and implementation of quantum error correcting codes.

DOI: [10.1103/cswp-xy7k](https://doi.org/10.1103/cswp-xy7k)

Quantum simulation [1,2] remains one of the most promising applications of quantum computers. Recent experiments have demonstrated encouraging results in simulating quantum magnetism [3–8]. Other systems that can be simulated in principle include the Fermi-Hubbard model, which provides insights into high-temperature superconductivity [9–11], and lattice gauge theories such as the Rokhsar-Kivelson model [12] and the Kogut-Susskind model of 2D quantum electrodynamics (QED) [13–16], where the phase diagram and real-time dynamics remain only partially understood [17,18]. These systems possess discrete translational symmetry, and this symmetry is inherited by circuits for the quantum simulation of those systems. In this Letter, we call such circuits *tileable* since they consist of a circuit tile or motif that is repeated spatially and optionally temporally.

A quantum simulation circuit typically comprises a sequence of one- and two-qubit unitaries acting on spin-1/2 degrees of freedom (qubits). In many architectures, such as superconducting or quantum dot devices, two-qubit gates can only be applied to specific pairs determined by the device's coupling graph [19,20] (also called the coupling map [19] or device graph [20]). Examples include the square grid [21] and the heavy-hexagonal lattice [5].

To overcome restricted qubit connectivity, SWAP gates (SWAPs) are commonly inserted [22,23]. A SWAP gate exchanges the states of two qubits, effectively swapping their positions. To apply a two-qubit gate to a pair of remote qubits, first these qubits must be routed along the edges of the coupling graph using SWAPs until they become adjacent. The *routing problem* [23], also called *quantum layout synthesis* [24,25], is the task of determining the initial positions of the qubits and the placement of the SWAPs. An optimal solution minimizes routing overhead, commonly measured by the routed circuit's depth or the number of inserted SWAPs.

Optimal qubit routing is crucial. First, on today's quantum computers, routing overhead increases the effects of noise, currently limiting simulations to models of quantum magnetism that can be directly embedded into the hardware coupling graph [4–7]. Second, on early fault-tolerant quantum computers, minimizing routing overhead maximizes the system sizes and simulation times that can be achieved. Finally, while scalable fault-tolerant quantum computers can reliably execute circuits of arbitrary depth, the routing overhead directly increases the monetary cost of running a circuit.

However, solving the routing problem optimally is NP hard [22,23,26], making optimal methods [24,27] slow and limited in the circuit sizes they can handle. Using highly optimized implementations [28], the performance of these methods can be improved, but given the problem's computational complexity, it is unlikely that optimal routing can be achieved for circuits acting on more than 100 qubits—roughly the number of qubits already available on today's quantum computers and the scale anticipated to be necessary

---

*Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.*

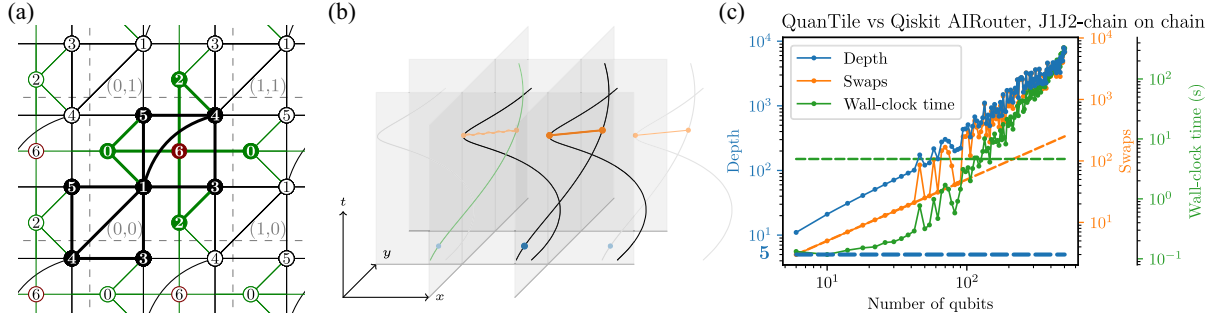


FIG. 1. (a) Basis circuit (bold edges) for the quantum simulation of the Kogut-Susskind model of 2D QED. Dashed lines delineate the circuit cells, with coordinates in gray, and seed numbers within the vertices. The Jordan-Wigner transformation breaks 2D translational symmetry, which we circumvent by using the compact encoding [37,38], leading to the red auxiliary qubits. (b) Qudit mobility zone. The logical qudits of the basis circuit (black worldlines) are free to move by the action of SWAPs (omitted) along the edges in a patch of  $2\delta + 1$  by  $2\delta + 1$  hardware cells (here,  $\delta = 1$  and edges are likewise omitted). The logical qudits are acted upon by single-qudit (dark blue dot) and two-qudit (dark orange edge) physical gates. In this example, the physical circuit (black lines, dark blue dots, and dark orange edges) is repeated to form a  $3 \times 1$  physical circuit patch. These physical basis circuits interact; a two-qudit gate from a translated copy of the physical basis circuit (wavy orange line) acts between a logical qudit from the original basis circuit and a logical qudit from a translated copy (green line). (c) Benchmarking results comparing our method’s implementation (“QuanTile,” dashed lines) [33] with Qiskit’s AIRouter [19] (solid lines).

for a quantum advantage [29,30]. Consequently, efficient but suboptimal heuristic methods have been developed [23,31,32], with SWAP counts sometimes exceeding the optimal by several orders of magnitude [25].

In this Letter, we introduce a framework for analyzing tileable circuits. It enables the modification of existing routing methods so that they only need to solve the routing problem for a single circuit tile, while ensuring that the routed circuit remains tileable. We provide an implementation of our method [33], in which we modify the optimal satisfiability modulo theories (SMT) based routing method from Ref. [24], with extensive verification of the correctness and performance of our method and implementation. We use it to find routing solutions with negligible overhead for circuits derived from quantum cellular automata [34–36], condensed matter simulations, and lattice gauge theory simulations. This unlocks the potential of today’s quantum computers to study frustrated magnetism and reduces computational cost on fault-tolerant devices by several orders of magnitude.

Although finding an (optimal) solution to the routing problem for a single tile is still NP hard, our method allows it to be repeated spatiotemporally at essentially no computational cost. The complexity of solving the routing problem now only depends on local properties of the simulated system and becomes essentially independent of the system size and the simulation time. Benchmarking our specialized method’s implementation [33] demonstrates that also in practice it greatly outperforms leading general-purpose routing software. For instance, on a 500-qubit circuit, our method reduces the depth overhead by more than 3 orders of magnitude while also requiring less computational time.

*Fundamentals*—We define a circuit tile formally as a *basis circuit* [Fig. 1(a)]. It consists of a finite set of gates  $g$ .

Since the routing problem is independent of whether the degrees of freedom are qubits or qudits with more than two basis states, we use the term “qudits” for generality. A single-qudit (two-qudit) gate  $g$  acts on qudit(s)  $g.g\tilde{q}(g.\tilde{q}, g.\tilde{q}')$ . We assign each gate a circuit layer  $g.\tilde{t} \in \mathbb{N}^0$ . Each qudit  $q$  has cell coordinates  $q.\tilde{x}, q.\tilde{y} \in \mathbb{Z}$  and a seed number  $q.\tilde{s} \in \mathbb{N}^0$ , which distinguishes qudits within a cell. This generalizes to higher dimensions by adding spatial coordinates.

A spatiotemporal *circuit patch*  $P_{n,m,l}(C)$  is formed by merging  $n \times m \times l$  translated copies of a depth- $d$  basis circuit  $C$ ,

$$P_{n,m,l}(C) = \bigcup_{\substack{(\Delta x, \Delta y, \Delta t) \\ \in \mathbb{Z}_n \times \mathbb{Z}_m \times d\mathbb{Z}_l}} T_{\Delta x, \Delta y, \Delta t}(C). \quad (1)$$

Here,  $T_{\Delta x, \Delta y, \Delta t}(C)$  translates each gate’s time coordinate by  $\Delta t$  and each qudit’s spatial coordinates by  $\Delta x$  and  $\Delta y$ , and  $d\mathbb{Z}_l = \{0, d, 2d, \dots, (l-1)d\}$ . Replacing  $\mathbb{Z}_n$  and  $\mathbb{Z}_m$  with  $\mathbb{Z}$  yields spatially infinite, or *lattice*, circuits. For formal reasons, the union is taken as a multiset union (allowing duplicates). We call circuits in which a basis graph can be identified *tileable circuits*.

A *gate collision* occurs when two gates simultaneously act on the same qudit. A basis circuit  $C$  is valid if the circuit patch  $P_{n,m,l}(C)$  (or equivalently, its induced lattice circuit) is collision-free for all  $n, m, l \in \mathbb{N}^+$ , which can be checked straightforwardly due to the following theorem.

*Theorem 1*—Let  $C$  be a basis circuit. The circuit patch  $P_{n,m,l}(C)$  is collision-free for all  $n, m, l \in \mathbb{N}^+$  if and only if, for each time  $t$  and seed number  $s$ , at most one qudit  $q$  with  $q.\tilde{s} = s$  is acted on by a gate in layer  $t$ . A proof is provided in the Supplemental Material (SM) [39].

A *basis graph* [48] is a basis circuit with edges instead of gates. Edges lack a time coordinate, and two edges may meet at the same vertex. Merging infinitely many translated copies of a basis graph produces a *lattice graph*, formalizing the concept of an “infinite lattice” with edges. Examples include the square, honeycomb, and kagome lattices, with edges connecting nearest neighbors.

Tileable circuits naturally arise in quantum simulations of lattice systems via Trotterization. Consider a two-local Hamiltonian  $H = \sum_i H_i + \sum_{(i,j)} H_{ij}$ , acting on  $\delta$ -dimensional local degrees of freedom (i.e.,  $\delta$ -level qudits), where  $H_i$  acts on qudit  $i$  and  $H_{(i,j)}$  on qudits  $i$  and  $j$  along the edges of a lattice graph  $G$ . By Trotterization [2,49,50],  $e^{-i\tau H} \approx [\tilde{U}_p(\tau/r)]^r$ , with  $\tilde{U}_p(\tau/r)$  a  $p$ th-order Trotter step and  $r$  the number of Trotter steps. A first-order step has the form  $\tilde{U}_1(\tau) = e^{-i\tau H^{(1)}} \dots e^{-i\tau H^{(\Gamma)}}$ , for some sequence  $H^{(1)}, \dots, H^{(\Gamma)}$  of the terms in  $H$ . Here,  $H^{(i)}$  is the  $i$ th term in the sequence, and each  $e^{-i\tau H^{(i)}}$  is a two-qudit gate. A second-order Trotter step is constructed by applying the first-order step twice, reversing the order in the second application,  $\tilde{U}_2(\tau) = \tilde{U}_1^\dagger(-\tau/2)\tilde{U}_1(\tau/2)$ . Using the bounds from [49], the error is  $O[N\tau(\tau/r)^p]$ , where  $N$  is the number of qudits.

Importantly, the Trotter step  $\tilde{U}_p$  is tileable for any  $p$ . For the routing problem, only the structure of  $G$  matters, so we consider arbitrary two-local (ATL) Hamiltonians, including models such as the Ising, (an)isotropic Heisenberg, Kitaev, and Bose-Hubbard models [39]. Trotterization does not prescribe a specific order  $H^{(i)}$ . Finding the order that minimizes circuit depth is equivalent to solving an NP-hard edge-coloring problem [48]. Tileable circuits also arise naturally in the task of finding ground states of lattice systems, for example, because the implementation of  $e^{-i\tau H}$  is a core subroutine [51], or because  $\tilde{U}_1$  determines the structure of a parameterized ansatz circuit [52,53].

*Routing*—It is common to call the circuit to be routed the *logical circuit*, acting with *logical gates* on *logical qudits*. The routed circuit is the *physical circuit*, acting with *physical gates* on *physical qudits*, while respecting hardware connectivity constraints. Although qudit routing is also needed for fault-tolerant quantum computing, we note the terms *logical* and *physical* here differ from their common usage in error correction. To physically execute a logical circuit, each logical qudit  $q$  is assigned to a physical qudit  $Q$  via a qudit map  $Q_q$ . We say that the logical qudit  $q$  resides at the physical qudit  $Q_q$ . If  $q$  starts at  $Q_q$  and a SWAP acts on  $Q_q$  and  $Q'$ , then afterward  $q$  resides at  $Q'$ . The qudit map has to be updated accordingly.

The routing problem for standard circuits was formulated as an SMT formula in [24]. An SMT formula consists of variables and constraints. Here, Boolean and integer variables encode the qudit map, the coordinates of the physical qudits of gates, and the time coordinates of

physical gates. Constraints on those variables ensure that the corresponding physical circuit is collision-free, respects the hardware connectivity, implements the same unitary as the logical circuit (up to a final reordering of the qudits), and has a preset depth  $D > 0$ . The smallest  $D$  for which all constraints become satisfiable yields the routed circuit with minimal depth overhead.

The second-order Trotter formula (and higher-order versions [39]) offers an advantage in qudit routing. Let  $U(\tau)$  be the routed version of  $\tilde{U}_1(\tau)$ . At the end of  $U(\tau)$ , logical qudits may not return to their initial positions, but after  $U^\dagger(-\tau)U(\tau)$ , they do. Thus,  $e^{-i\tau H} \approx [\tilde{U}_2(\tau/r)]^r = \{U^\dagger[-\tau/(2r)]U[-\tau/(2r)]\}^r$ , where the right-hand side is a fully routed circuit. Thus, it suffices to route a single first-order step without enforcing logical qudits to return to their initial position. This temporal symmetry can be exploited alongside spatial symmetry. Alternatively, one can enforce qudits to return to their initial positions at the end of  $U(\tau)$ , though this may increase circuit depth.

When simulating Hamiltonians with two-qubit interactions  $H_{ij}(\Delta) \sim X_i X_j + Y_i Y_j + \Delta Z_i Z_j$  on hardware where the CNOT is native, SWAPs can be absorbed into directly preceding or following two-qubit simulation unitaries  $e^{-i\tau H^{(i)}}$  at no increase of the infidelity of the subcircuits implementing those unitaries [39,54]. The same applies when the two-qubit fSim gate [39,55,56] is native.

*Routing tileable circuits*—To leverage spatial symmetry, we map tileable circuits to hardware whose connectivity graph is described by (a patch of) a lattice graph. The unitary implemented by any  $n \times m \times l$  physical circuit patch must equal the unitary implemented by the corresponding  $n \times m \times l$  logical circuit patch (up to a final reordering of qudits). Since this holds for any  $n$  and  $m$ , we may even consider infinite physical circuit patches, or *physical lattice circuits*.

In the mathematical formulation of the qudit routing problem for tileable circuits, several concepts and assumptions beyond those required for standard routing are essential. First, when placing any gate into the physical basis circuit, any physical circuit patch induced by the physical basis circuit [Eq. (1)] must be collision-free. This is done straightforwardly by using Theorem 1, demonstrating its necessity and effectiveness. Second, due to the discrete spatial translational symmetry of a physical lattice circuit, the qudit map must share the same symmetry. We assume that the initial qudit map conserves cell coordinates. Translational symmetry then requires that for each seed number  $s$ , there is a unique seed number  $S$  such that all logical qudits with seed number  $s$  are mapped to physical qudits with seed number  $S$ . Logical qudits can move along the edges of the hardware lattice graph via SWAPs, but we assume they cannot move outside the *mobility zone* [Fig. 1(b)], which is a patch of  $2\delta + 1$  by  $2\delta + 1$  hardware cells centered around the central cell. The mobility zone can be made arbitrarily large at the cost of increased computational resources.

Finally, consider inserting a SWAP gate  $g$  acting on physical qudits  $(Q, Q')$  into the physical basis circuit  $C$ . By Eq. (1), the physical lattice circuit induced by  $C$  will include SWAPs along  $T_{\Delta x, \Delta y, 0}[g]$  for every  $\Delta x, \Delta y$ . Crucially, these translated SWAPs, although they are not in  $C$  itself, may act on physical qudits of  $C$  holding logical qudits. We account for this effect by temporarily inserting all relevant possible translated versions of each SWAP gate that is added to the physical basis circuit. This renders the basis circuit technically invalid by Theorem 1. In postprocessing, however, we retain only the untranslated SWAP gate, ensuring that the physical lattice circuit is collision-free. The translated SWAPs acting on  $C$  automatically reemerge in the final physical lattice circuit. For finite physical circuit patches, boundary effects occur, which are dealt with in detail in the SM [39].

*Implementation*—Until now, our discussion has been conceptual and not related to the implementation. For demonstration, we reformulate the above concepts as an SMT formula, as it was done in [24] for general, structureless circuits, and implement this SMT formulation in code. In the End Matter, we give an example of a constraint unique to the routing of tileable circuits. The full details of our SMT formulation and implementation can be found in the SM [39]. Our framework is modular and the SMT-based method can be replaced by other routing methods, including heuristic methods.

The implementation provides the following options that, like tileability, go beyond the capabilities of standard routing methods: (i) optimize the order of two-qudit gates in the Trotter step; (ii) allow SWAPs to merge with directly preceding or following two-qudit gates; (iii) enforce that logical qudits return to their initial positions at the end of the circuit (“cyclic routing”); and (iv) slice the logical circuit into subcircuits to route sequentially, reducing the computational complexity to linear in the logical circuit depth.

*Results*—We applied our method to basis circuits for one Trotter step in the quantum simulation of: ATL Hamiltonians on 24 different 1D and 2D lattices, the Fermi-Hubbard model on the square lattice [10], the Rokhsar-Kivelson model [12], and the Kogut-Susskind model of 2D QED [13]. Finally, we applied our method to the basis circuit of the Rule 54 quantum cellular automaton [35,36]. Using Eq. (1), the solutions can be tiled spatially and temporally, creating arbitrarily deep and wide routed circuits. The routing problems were solved for various combinations of the routing options (i–iv).

All circuits and the 1D and 2D lattices are provided explicitly in the SM, along with extensive results [39]. An excerpt of the results is given in Table I, with the exact routing options used given in the End Matter. For the solutions we found, the routing overhead is remarkably low. In some cases there is no overhead at all, possible under option (i). For the zero-overhead cases, we note that

TABLE I. Excerpt of the results obtained by our method [33] in routing a single Trotter step for various lattice systems (first column) to various hardware coupling graphs (second column). ATL J1J2- $\ell$  denotes an arbitrary two-local model (ATL) on a lattice  $\ell$  with edges between nearest and next-nearest neighbors. The reported depth overhead (third column) is given per Trotter step and remains valid for any number of Trotter steps and logical circuit patch size. The SWAP overhead (last column) is per Trotter step and per the number of qudits indicated in parentheses. In all listed solutions, the qudit overhead is zero, except for Kogut-Susskind, where it is one qudit per seven logical qudits.

| Simulated system | Hardware coupling graph | Depth overhead | SWAP overhead |
|------------------|-------------------------|----------------|---------------|
| ATL ladder       | Chain                   | 0 (0%)         | 0             |
| ATL J1J2-ladder  | Chain                   | 0 (0%)         | 0             |
| ATL J1J2-chain   | Chain                   | 1 (25%)        | 0             |
| Rule 54          | Ladder                  | 0 (0%)         | 1 (4)         |
| ATL J1J2-chain   | Ladder                  | 0 (0%)         | 0             |
| ATL J1J2-square  | Square grid             | 0 (0%)         | 0             |
| ATL triangular   | Square grid             | 0 (0%)         | 0             |
| ATL kagome       | Square grid             | 1 (25%)        | 0             |
| ATL shuriken     | Square grid             | 1 (25%)        | 0             |
| ATL snub-square  | Square grid             | 1 (20%)        | 0             |
| Rokhsar-Kivelson | Square grid             | 2 (11%)        | 2 (4)         |
| Fermi-Hubbard    | Square grid             | 0 (0%)         | 0             |
| Kogut-Susskind   | Square grid             | 9 (4%)         | 253 (6)       |

increasing  $\delta$  cannot decrease the depth further. For the remaining low-overhead cases, we expect limited to no further improvements from increasing  $\delta$  because of the locality of the input circuits.

Our method offers a scaling advantage over general-purpose methods because its running time is essentially independent of the logical circuit size. Nevertheless, the question remains if this advantage is already significant for today’s quantum chips, with on the order of 100 to 1000 qubits. We therefore benchmarked our method against multiple established routing methods across various routing problems [39]. Here, we show the results of comparing our method to Qiskit’s leading AIRouter [32] on the problem of routing a single Trotter step for the simulation of an ATL J1J2 model on a chain to hardware with chain connectivity [Fig. 1(c)]. Since the AIRouter does not optimize gate order (i), we allow our implementation to perform this optimization and then use the resulting gate order as a fixed input for the AIRouter. Because the AIRouter does not support SWAP merging (ii) nor cyclic routing (iii), we disable these options in our implementation. As both methods route a single Trotter step, their solutions can be repeated temporally to construct arbitrarily deep second-order Trotter circuits, but only our solution can also be repeated spatially. Our method’s implementation becomes faster for system sizes above approximately 100 qudits, while also producing solutions with significantly lower depth and SWAP

overhead. At around 500 qudits, the depth overhead decreases from approximately  $10^4$  to just 5.

*Conclusion*—We have demonstrated that for circuits with discrete spatiotemporal translational symmetry, naturally arising in the quantum simulation of condensed matter systems and lattice gauge theories, inherently scalable qubit routing solutions can be achieved with negligible overhead. Beyond a reduction of costs on fault-tolerant devices by several orders of magnitude, this enables the simulation of geometrically frustrated magnetism on current devices. One possibility is the observation of disorder-free localization and many-body quantum scars in a Heisenberg model on the kagome lattice [57,58]. The circuits require  $O(1)$ -depth state preparation, followed by the Trotterized simulation circuit, which for this system is possible on square-grid hardware using five fSim gate layers per Trotter step (Table I). With 100 qubits and a depth-100 circuit, which could be within reach of pre-fault-tolerant devices [30], it becomes possible to simulate around 11 second-order (or 17 first-order [39]) Trotter steps on 100 qubits.

Using the techniques from [28], an optimization of our implementation is expected to decrease its own (classical) running time by several orders of magnitude. Also, while our focus has been on an optimal method, within our framework, heuristic methods can also be adapted so that their solutions become tileable. This could be valuable when the basis circuits become too large for optimal methods, e.g., when routing circuits for future modular and tileable hardware, where each module consists of hundreds of qubits [59].

We have laid the foundation for addressing compilation tasks other than qubit routing in the tileable setting. Examples include leveraging gate identities to reduce logical circuit depth [60], compilation to error-correction native gates [61], automated and optimized construction of logical quantum simulation circuits [62], and routing by shuttling [63,64]. Similar improvements over leading methods are expected in these areas as well.

*Acknowledgments*—We acknowledge the support from the German Ministry for Education and Research, under the QSolid project, Grant No. 13N16167, and from the State of Baden-Württemberg within the Competence Center Quantum Computing, projects QORA II and KQCBW24.

*Data availability*—Our method's implementation (QuanTile) and all data are available open source at Ref. [33].

- 
- [1] R. P. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.* **21**, 467 (1982).  
 [2] S. Lloyd, Universal quantum simulators, *Science* **273**, 1073 (1996).  
 [3] C. J. van Diepen, T.-K. Hsiao, U. Mukhopadhyay, C. Reichl, W. Wegscheider, and L. M. K. Vandersypen, Quantum

- simulation of antiferromagnetic Heisenberg chain with gate-defined quantum dots, *Phys. Rev. X* **11**, 041025 (2021).  
 [4] P. Frey and S. Rachel, Realization of a discrete time crystal on 57 qubits of a quantum computer, *Sci. Adv.* **8**, eabm7652 (2022).  
 [5] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. van den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala, Evidence for the utility of quantum computing before fault tolerance, *Nature (London)* **618**, 500 (2023).  
 [6] E. Rosenberg, T. I. Andersen, R. Samajdar, A. Petukhov, J. C. Hoke, D. Abanin, A. Bengtsson, I. K. Drozdov, C. Erickson, P. V. Klimov *et al.*, Dynamics of magnetization at infinite temperature in a Heisenberg spin chain, *Science* **384**, 48 (2024).  
 [7] X. Mi, A. A. Michailidis, S. Shabani, K. C. Miao, P. V. Klimov, J. Lloyd, E. Rosenberg, R. Acharya, I. Aleiner, T. I. Andersen *et al.*, Stable quantum-correlated many-body states through engineered dissipation, *Science* **383**, 1332 (2024).  
 [8] G. Burkard, Recipes for the digital quantum simulation of lattice spin systems, *SciPost Phys. Core* **8**, 030 (2025).  
 [9] T. Hensgens, T. Fujita, L. Janssen, X. Li, C. J. Van Diepen, C. Reichl, W. Wegscheider, S. Das Sarma, and L. M. K. Vandersypen, Quantum simulation of a Fermi-Hubbard model using a semiconductor quantum dot array, *Nature (London)* **548**, 70 (2017).  
 [10] E. Dagotto, Correlated electrons in high-temperature superconductors, *Rev. Mod. Phys.* **66**, 763 (1994).  
 [11] C. Cade, L. Mineh, A. Montanaro, and S. Stanisic, Strategies for solving the Fermi-Hubbard model on near-term quantum computers, *Phys. Rev. B* **102**, 235122 (2020).  
 [12] D. S. Rokhsar and S. A. Kivelson, Superconductivity and the quantum hard-core dimer gas, *Phys. Rev. Lett.* **61**, 2376 (1988).  
 [13] J. Kogut and L. Susskind, Hamiltonian formulation of Wilson's lattice gauge theories, *Phys. Rev. D* **11**, 395 (1975).  
 [14] J. F. Haase, L. Dellantonio, A. Celi, D. Paulson, A. Kan, K. Jansen, and C. A. Muschik, A resource efficient approach for quantum and classical simulations of gauge theories in particle physics, *Quantum* **5**, 393 (2021).  
 [15] D. Paulson, L. Dellantonio, J. F. Haase, A. Celi, A. Kan, A. Jena, C. Kokail, R. van Bijnen, K. Jansen, P. Zoller, and C. A. Muschik, Simulating 2D effects in lattice gauge theories on a quantum computer, *PRX Quantum* **2**, 030334 (2021).  
 [16] M. Meth, J. F. Haase, J. Zhang, C. Edmunds, L. Postler, A. Steiner, A. J. Jena, L. Dellantonio, R. Blatt, P. Zoller, T. Monz, P. Schindler, C. Muschik, and M. Ringbauer, Simulating 2D lattice gauge theories on a qudit quantum computer, *Nat. Phys.* **21**, 570 (2025).  
 [17] T. Pichler, M. Dalmonte, E. Rico, P. Zoller, and S. Montangero, Real-time dynamics in U(1) lattice gauge theories with tensor networks, *Phys. Rev. X* **6**, 011023 (2016).  
 [18] T. Felser, P. Silvi, M. Collura, and S. Montangero, Two-dimensional quantum-link lattice quantum electrodynamics at finite density, *Phys. Rev. X* **10**, 041040 (2020).

- [19] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross *et al.*, Quantum computing with Qiskit, [arXiv:2405.08810](https://arxiv.org/abs/2405.08810).
- [20] Cirq Developers, Cirq, 2024, [10.5281/zenodo.4062499](https://doi.org/10.5281/zenodo.4062499).
- [21] R. Acharya, D. A. Abanin, L. Aghababaie-Beni, I. Aleiner, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, N. Astrakhantsev *et al.*, Quantum error correction below the surface code threshold, *Nature (London)* **638**, 920 (2025).
- [22] A. M. Childs, E. Schoute, and C. M. Unsal, Circuit transformations for quantum architectures, in *14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2019)*, Leibniz International Proceedings in Informatics (LIPIcs) Vol. 135, edited by W. van Dam and L. Mancinska (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019), pp. 3:1–3:24.
- [23] A. Cowtan, S. Dilkes, R. Duncan, A. Krajenbrink, W. Simmons, and S. Sivarajah, On the qubit routing problem, in *14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2019)*, Leibniz International Proceedings in Informatics (LIPIcs) Vol. 135, edited by W. van Dam and L. Mancinska (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019), pp. 5:1–5:32.
- [24] B. Tan and J. Cong, Optimal layout synthesis for quantum computing, in *Proceedings of the 39th International Conference on Computer-Aided Design, ICCAD '20* (Association for Computing Machinery, New York, USA, 2020).
- [25] S. Ping, W.-H. Lin, D. Bochen Tan, and J. Cong, Assessing quantum layout synthesis tools via known optimal-SWAP cost benchmarks, [arXiv:2502.08839](https://arxiv.org/abs/2502.08839).
- [26] D. Maslov, S. M. Falconer, and M. Mosca, Quantum circuit placement, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **27**, 752 (2008).
- [27] G. Nannicini, L. S. Bishop, O. Günlük, and P. Jurcevic, Optimal qubit assignment and routing via integer programming, *ACM Trans. Quantum Comput.* **4** (2022), [10.1145/3544563](https://doi.org/10.1145/3544563).
- [28] W.-H. Lin, J. Kimko, B. Tan, N. Bjørner, and J. Cong, Scalable optimal layout synthesis for NISQ quantum processors, in *2023 60th ACM/IEEE Design Automation Conference (DAC)* (2023), pp. 1–6, [10.1109/DAC56929.2023.10247760](https://doi.org/10.1109/DAC56929.2023.10247760).
- [29] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Toward the first quantum simulation with quantum speedup, *Proc. Natl. Acad. Sci. U.S.A.* **115**, 9456 (2018).
- [30] Z. Zimborás, B. Koczor, Z. Holmes, E.-M. Borrelli, A. Gilyén, H.-Y. Huang, Z. Cai, A. Acín, L. Aolita, L. Banchi *et al.*, Myths around quantum computation before full fault tolerance: What no-go theorems rule out and what they don't, [arXiv:2501.05694](https://arxiv.org/abs/2501.05694).
- [31] G. Li, Y. Ding, and Y. Xie, 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, ASPLOS '19* (Association for Computing Machinery, New York, USA, 2019), pp. 1001–1014.
- [32] D. Kremer, V. Villar, H. Paik, I. Duran, I. Faro, and J. Cruz-Benito, Practical and efficient quantum circuit synthesis and transpiling with reinforcement learning, [arXiv:2405.13196](https://arxiv.org/abs/2405.13196).
- [33] J. Kattemölle, QuanTile, 2024, [10.5281/zenodo.14336692](https://doi.org/10.5281/zenodo.14336692).
- [34] T. Farrelly, A review of quantum cellular automata, *Quantum* **4**, 368 (2020).
- [35] S. Gopalakrishnan and B. Zakirov, Facilitated quantum cellular automata as simple models with non-thermal eigenstates and dynamics, *Quantum Sci. Technol.* **3**, 044004 (2018).
- [36] K. Klobas, B. Bertini, and L. Piroli, Exact thermalization dynamics in the “Rule 54” quantum cellular automaton, *Phys. Rev. Lett.* **126**, 160602 (2021).
- [37] C. Derby, J. Klassen, J. Bausch, and T. Cubitt, Compact fermion to qubit mappings, *Phys. Rev. B* **104**, 035118 (2021).
- [38] L. Clinton, T. Cubitt, B. Flynn, F. M. Gabbetta, J. Klassen, A. Montanaro, S. Piddock, R. A. Santos, and E. Sheridan, Towards near-term quantum simulation of materials, *Nat. Commun.* **15**, 211 (2024).
- [39] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/cswp-xy7k>. It includes the proof of Theorem 1, the explicit SMT formula for tileable routing, a detailed description of our implementation [24], and additional results. All basis graphs and tileable quantum circuits used in this Letter are given explicitly, which includes Refs. [40–47].
- [40] P. V. Sriluckshmy, V. Pina-Canelles, M. Ponce, M. G. Algaba, F. Šimkovic IV, and M. Leib, Optimal, hardware native decomposition of parameterized multi-qubit Pauli gates, *Quantum Sci. Technol.* **8**, 045029 (2023).
- [41] Y. S. Yordanov, D. R. M. Arvidsson-Shukur, and Crispin H. W. Barnes, Efficient quantum circuits for quantum computational chemistry, *Phys. Rev. A* **102**, 062612 (2020).
- [42] D. Loss and D. P. DiVincenzo, Quantum computation with quantum dots, *Phys. Rev. A* **57**, 120 (1998).
- [43] G. Burkard, T. D. Ladd, A. Pan, J. M. Nichol, and J. R. Petta, Semiconductor spin qubits, *Rev. Mod. Phys.* **95**, 025003 (2023).
- [44] L. De Moura and N. Bjørner, Z3: An efficient SMT solver, in *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'08/ETAPS'08* (Springer-Verlag, Berlin, Heidelberg, 2008), pp. 337–340.
- [45] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*, Computer Science and Information Processing (Addison-Wesley, Reading, MA, 1983).
- [46] H. Zou, M. Treinish, K. Hartman, A. Ivrii, and J. Lishman, LightSABRE: A lightweight and enhanced SABRE algorithm, [arXiv:2409.08368](https://arxiv.org/abs/2409.08368).
- [47] J. Kattemölle and S. Hariharan, Line-graph qubit routing, *ACM Trans. Quantum Comp.* **6**, 22 (2025).
- [48] J. Kattemölle, Edge coloring lattice graphs, *J. Math. Phys. (N.Y.)* **66**, 051901 (2025).
- [49] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, Theory of Trotter error with commutator scaling, *Phys. Rev. X* **11**, 011020 (2021).

- [50] M. Suzuki, General theory of fractal path integrals with applications to many-body theories and statistical physics, *J. Math. Phys. (N.Y.)* **32**, 400 (1991).
- [51] A. Y. Kitaev, Quantum measurements and the Abelian stabilizer problem, [arXiv:quant-ph/9511026](https://arxiv.org/abs/quant-ph/9511026).
- [52] D. Wecker, M. B. Hastings, and M. Troyer, Progress towards practical quantum variational algorithms, *Phys. Rev. A* **92**, 042303 (2015).
- [53] J. Kattemölle and J. van Wezel, Variational quantum eigensolver for the Heisenberg antiferromagnet on the kagome lattice, *Phys. Rev. B* **106**, 214429 (2022).
- [54] G. Vidal and C. M. Dawson, Universal quantum circuit for two-qubit transformations with three controlled-NOT gates, *Phys. Rev. A* **69**, 010301(R) (2004).
- [55] B. Foxen, C. Neill, A. Dunsworth, P. Roushan, B. Chiaro, A. Megrant, J. Kelly, Z. Chen, K. Satzinger, R. Barends *et al.* (Google AI Quantum), Demonstrating a continuous set of two-qubit gates for near-term quantum algorithms, *Phys. Rev. Lett.* **125**, 120504 (2020).
- [56] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, *Nature (London)* **574**, 505 (2019).
- [57] P. A. McClarty, M. Haque, A. Sen, and J. Richter, Disorder-free localization and many-body quantum scars from magnetic frustration, *Phys. Rev. B* **102**, 224303 (2020).
- [58] K. Lee, R. Melendrez, A. Pal, and H. J. Changlani, Exact three-colored quantum scars from geometric frustration, *Phys. Rev. B* **101**, 241111(R) (2020).
- [59] S. Bravyi, O. Dial, J. M. Gambetta, D. Gil, and Z. Nazario, The future of quantum computing with superconducting qubits, *J. Appl. Phys.* **132**, 160902 (2022).
- [60] D. Maslov, G. W. Dueck, D. M. Miller, and C. Negrevergne, Quantum circuit simplification and level compaction, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **27**, 436 (2008).
- [61] C. Ryan-Anderson, N. Brown, C. Baldwin, J. Dreiling, C. Foltz, J. Gaebler, T. Gatterman, N. Hewitt, C. Holliman, C. Horst *et al.*, High-fidelity teleportation of a logical qubit using transversal gates and lattice surgery, *Science* **385**, 1327 (2024).
- [62] G. Li, A. Wu, Y. Shi, A. Javadi-Abhari, Y. Ding, and Y. Xie, Paulihedral: A generalized block-wise compiler optimization framework for quantum simulation kernels, in *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '22 (Association for Computing Machinery, New York, USA, 2022), pp. 554–569.
- [63] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter *et al.*, Logical quantum processor based on reconfigurable atom arrays, *Nature (London)* **626**, 58 (2024).
- [64] D. B. Tan, D. Bluvstein, M. D. Lukin, and J. Cong, Compiling quantum circuits for dynamically field-programmable neutral atoms array processors, *Quantum* **8**, 1281 (2024).

## End Matter

*Tileable SMT formulation*—Denote by  $G_g.T$  the integer variable describing the time coordinate of the physical gate  $G$  implementing the logical gate  $g$ , and denote by  $G_g.Q.S$  ( $G_g.Q'.S$ ) the integer variable describing the seed number of qubit 1 (2) of  $G_g$ . Using Theorem 1, tileability of the physical basis circuit is asserted by adding the constraint

$$G_g.T = G_{g'}.T \Rightarrow G_g.A.S \neq G_{g'}.B.S \quad (\text{A1})$$

for all pairs  $\{g, g'\}$  of indistinct logical gates and all

$A, B \in \{Q, Q'\}$  (momentarily assuming only two-qudit gates for simplicity).

*Used routing options*—For the results displayed in Table I, we assumed a mobility zone of 3 by 3 cells ( $\delta = 1$ ). For the ATL models only, we optimized the gate order (i). We allowed SWAP-merging (ii), except for the Fermi-Hubbard circuit. We enforced cyclic routing (iii), except for the ATL circuits. Logical circuits were not sliced (iv), except for the Kogut-Susskind circuit, which had a slice depth of 20.