

Quantum circuit compilation with quantum computers

Davide Rattacaso^{1,2}, Daniel Jaschke^{1,2,3}, Marco Ballarin^{1,2}, Ilaria Siloi^{1,2} and Simone Montangero^{1,2}

¹*Dipartimento di Fisica e Astronomia “G. Galilei” & Padua Quantum Technologies Research Center,
Università degli Studi di Padova, I-35131 Padova, Italy*

²*INFN, Sezione di Padova, via Marzolo 8, I-35131 Padova, Italy*

³*Institute for Complex Quantum Systems, Ulm University, Albert-Einstein-Allee 11, 89069 Ulm, Germany*



(Received 1 October 2024; accepted 20 August 2025; published 22 September 2025)

Compilation optimizes quantum algorithms’ performances on real-world quantum computers. To date, it is performed via classical optimization strategies. We introduce a class of quantum algorithms to perform compilation via quantum computers, paving the way for a quantum advantage in compilation. We demonstrate the effectiveness of this approach via quantum and simulated annealing-based compilation: We successfully compile a Trotterized Hamiltonian simulation with up to 64 qubits and 64 time steps and a quantum Fourier transform with up to 40 qubits and 771 time steps. We show that, for a translationally invariant circuit, the compilation results in a fidelity gain that grows extensively in the size of the input circuit, outperforming any local or quasilocal compilation approach.

DOI: [10.1103/hcz4-nv2y](https://doi.org/10.1103/hcz4-nv2y)

I. INTRODUCTION

Executing a quantum algorithm on a real-world quantum computer with limited resources and efficiency necessitates a compilation process. The compilation generates a circuit implementation of the algorithm that minimizes errors or run-time when executed on the specific hardware, e.g., using the native gate set [1–3]. However, even simple compilation instances have been proven NP-complete through reduction to SAT [4], requiring approximated solutions. Numerous classical approaches have been proposed to heuristically address the compilation of quantum algorithms, both with a hardware-agnostic approach [5–20] and taking into account various platform-dependent error sources [3,21–25].

Assuming that the development of quantum computers will follow the trajectory of classical computing, they will eventually achieve the scalability and precision necessary to compile quantum algorithms, just as classical computers successfully compile classical algorithms [26]. To date, there is no strategy to compile quantum circuits with quantum computers. Motivated by the computational advantage demonstrated by quantum algorithms in solving some combinatorial optimization instances [27–29], we propose a general paradigm for designing quantum compilers, i.e., quantum algorithms executed on quantum devices to compile other quantum algorithms, laying the groundwork for a potential quantum speedup in the compilation of quantum circuits. We frame the circuit compilation problem as a ground-state search of a many-body infidelity Hamiltonian $\hat{H}_I$ diagonal

in the computational basis. This Hamiltonian accounts for the errors affecting the circuit execution on the hardware—optimization task—and enforces constraints that ensure the circuit is composed entirely by native gates—compilation task. To this aim, each quantum circuit is encoded into a state of the computational basis of a lattice of qudits, as shown in Fig. 1. The ground state of $\hat{H}_I$ encodes an optimal equivalent quantum circuit, representing the best arrangement of qubits and gates to minimize infidelities while producing the same quantum computation. This encoding enables the exploitation of different quantum algorithms to search for the ground state, i.e., quantum annealing (QA) [30–35], optimal control (OC) [36–42], and quantum approximate optimization algorithm (QAOA) [43–47].

Different from standard optimization, the search space is crucially constrained to the set of many-body states representing equivalent circuits, i.e., circuits that implement the same unitary operator. To enforce this constraint during the quantum optimization, we evolve the state only through Hamiltonians that transform circuits by replacing small subcircuits with equivalent ones [see Figs. 1(a) and 1(b)]. We first focus on QA and validate the QA-based quantum compiler by classically simulating the compilation of the circuit in Fig. 1(c) via tensor networks (TNs) [48–54]. As it should be, the final probability of sampling the global optimal circuit increases monotonically with the annealing time (Fig. 2). To address larger circuits, we implement simulated annealing (SA)-based compilation, assuming its performance as a lower bound to QA. We compile quantum circuits running on a two-dimensional quantum processing unit (QPU) with parametric gates and long-range crosstalk with up to 64 qubits. The gain obtained by this compilation process scales extensively, demonstrating that this approach fully exploits the many-body nature of the compilation problem: The compilation of the global circuit is more efficient

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article’s title, journal citation, and DOI.

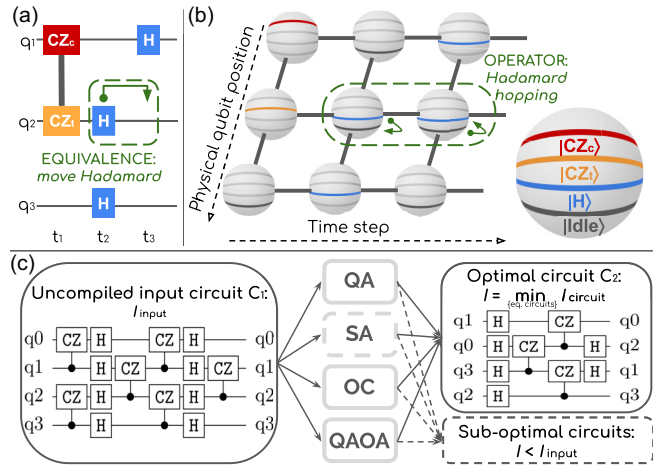


FIG. 1. Many-body mapping of the compilation problem. The quantum circuit in panel (a) is mapped to the state of a lattice of qudits (spheres) in panel (b). Each internal state of a qudit represents a possible gate, e.g., Hadamard, CZ control, CZ target, or a fictitious idle gate. The local state of the qudit at site (t, q) represents the gate acting at the time step t on the qubit q . The transformations (green dashed box) replacing equivalent subcircuits are embedded in operators that update the states of adjacent qudits. Panel (c) sketches the circuit compilation executed as a ground-state search via quantum annealing, simulated annealing, optimal control, or QAOA. Both global optimal and suboptimal circuits can be generated. The Circuit C_2 is the compiled version of Circuit C_1 (see main text).

than what is obtained via combining quasilocal compiled circuits.

Beyond circuit compilation, the methods introduced in this work open up the possibility of quantum optimization of equational theories [55]. These formal systems automate the exploration of semantically equivalent symbolic expressions. Relying exclusively on the replacement of subexpressions, they serve as a fundamental component of symbolic algebra algorithms.

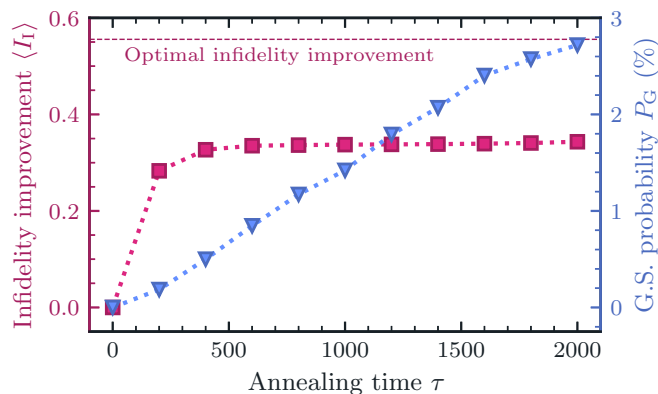


FIG. 2. Quantum annealing-based compiler. Expectation value of the infidelity improvement $I_I = 1 - I_{opt}/I_{input}$ as a function of the annealing time (left y axis). Probability P_G of obtaining an equivalent globally optimal circuit (right y axis). The input circuit and the optimal circuit are depicted in Fig. 1(c). The optimal infidelity improvement refers to the optimal output circuit.

II. METHODS

In this section, we show how the circuit compilation problem can be approached using quantum optimization techniques. In Sec. II A, we define a hardware-aware classical cost function that maps each candidate circuit to the expected infidelity incurred when executing it on the target hardware. In Sec. II B, we describe how to encode quantum circuits as states of a many-qudit system. Section II C introduces a Hamiltonian encoding of the infidelity cost function, while Sec. II D presents a Hermitian operator representation of the circuit equivalence rules. Finally, in Sec. II E, we demonstrate how the dynamics generated by the equivalence operators can be used to drive a quantum optimization process that minimizes the infidelity without altering the unitary operation implemented by the circuit.

A. Infidelity cost function

The objective is to find a circuit C that implements a target quantum algorithm while minimizing a given cost function, i.e., the infidelity, denoted as $I(C)$. Each circuit C represents a sequence of instructions (G, t, q) that specify the gate G applied at the time step t to the qubit q . We assume that the successful executions of different gates are independent events, implying that the total success probability of the circuit is the product of the individual gate fidelities [3]. Aiming to compile large quantum circuits, we also expect gate infidelities to be small, $i_G \ll 1$. Upon these assumptions, one can maximize the circuit fidelity by minimizing the sum of the infidelities affecting the native gates. Specifically, the infidelity induced by the execution of the gate G on the qubit q is approximated by the sum of the infidelity i_G of the gate when executed alone, and the crosstalk contributions $x_G(\|q - q'\|)$ accounting for the additional infidelity that affect each couple of qubits q and q' involved in the simultaneous application of crosstalking gates. We also need to consider an infidelity i_{Idle} for idle qubits, e.g., due to dephasing or decay. Under these assumptions, in Appendix A we show that the total infidelity of C is given by

$$I(C) = \sum_{(G,t,q) \in C} \left(i_G + \sum_{q' | G(t,q')=G} x_G(\|q - q'\|) \right) + \sum_{Idle(t,q)} i_{Idle}, \quad (1)$$

where i_G , $x_G(\|q - q'\|)$, and i_{Idle} depend on the target hardware and are fixed parameters for the compiler.

B. Encoding quantum circuits in quantum states

The number of candidate circuits to minimize infidelity grows exponentially with the number of qubits N_Q and the number of time steps N_T required to implement the target algorithm. We map each quantum circuit C to a quantum state $|C\rangle$. For any QPU with physical qubits arranged on an n -dimensional lattice, we encode circuits in the states of the computational basis within a $(1 + n)$ -dimensional lattice of qudits. The first coordinate t denotes the time step, while the other coordinates define a vector $\mathbf{q}$ that indicates the

position of the qubit in the n -dimensional QPU lattice. As shown in Fig. 1(b), each lattice site $(t, \mathbf{q})$ is then associated with a d -dimensional local configuration space $\{|G\rangle | G \in \{\text{Idle}, G_2, \dots, G_d\}\}$, where each state represents a gate executed at time step t on qubit $\mathbf{q}$. The $|\text{Idle}\rangle$ state indicates that qubit $\mathbf{q}$ is idle. Circuits composed of less than N_T time steps are encoded in computational basis states with some entirely idle time steps $|\text{Idle}\rangle_{t,1} \otimes \dots \otimes |\text{Idle}\rangle_{t,N_Q}$. These idle time steps are assumed to be skipped during circuit execution. The space spanned by the computational basis representing all possible circuits is then the Hilbert space $\mathbb{H} = (\mathbb{C}^d)^{N_Q \times N_T}$.

C. The infidelity Hamiltonian

We define the infidelity Hamiltonian $\hat{H}_I$ that associates with each circuit state $|\mathbf{C}\rangle$ the infidelity $I(\mathbf{C}) = \langle \mathbf{C} | \hat{H}_I | \mathbf{C} \rangle$. To this aim, we introduce the operator $\hat{g}_{t,q}^\dagger$ that creates the gate G at the lattice site (t, q) , whose vacuum state is the Idle gate, and acts trivially on the other sites:

$$\hat{g}_{t,q}^\dagger := \bigotimes_{t' \neq t, q' \neq q} \mathbb{1}_{t',q'} \otimes |G\rangle_{t,q} \langle \text{idle}|_{t,q}. \quad (2)$$

Any n -qubit gate is created (annihilated) by the action of n creation (annihilation) operators. Then, $\hat{H}_I$ yields

$$\begin{aligned} \hat{H}_I = & \sum_{t,q} i_G \hat{g}_{t,q} + \sum_{t,q,G} \sum_{q' \neq q} x_G(q, q') \hat{g}_{t,q} \hat{g}_{t,q'}^\dagger \\ & - N_Q i_{\text{Idle}} \sum_t \bigotimes_q |\text{Idle}\rangle_{t,q}, \end{aligned} \quad (3)$$

where $\hat{g}_{t,q} := \hat{g}_{t,q}^\dagger \hat{g}_{t,q}$ has the role of the number operator at the lattice site (t, q) for the gate G . The parameters i_G , x_G , and i_{Idle} encode sources of infidelity specific to the target hardware. The first term in Eq. (3) is a single-body operator encoding the infidelity of each gate when executed alone, including the infidelity of idle qubits. This term also enables the suppression of non-native gates that may be present in the input circuit by assigning them a high fictitious infidelity. The second term, a two-body interaction, estimates the contribution of crosstalk. The last term is necessary to minimize the circuit depth by enforcing parallel gate execution. It maximizes the number of completely idle time steps, which can then be skipped during the execution of the circuit. More generally, we can encode various classical functions of the circuit into the Hamiltonian $\hat{H}_I$, such as depth or energy consumption. These will remain important metrics even in the fault-tolerant era.

D. Circuit equivalences as Hermitian operators

We leverage quantum optimization to target the equivalent circuit that minimizes the expectation value of $\hat{H}_I$ while implementing the same unitary operator. We define a subcircuit as any block of adjacent gates in the lattice of qudits. Starting from the original input circuit, we explore equivalent circuits by applying a set of invertible transformation rules $\mathcal{E} := \{T = (\mathbf{C}_{\text{sub}}^{\text{in}} \leftrightarrow \mathbf{C}_{\text{sub}}^{\text{out}})\}$ that replace a subcircuit $\mathbf{C}_{\text{sub}}^{\text{in}}$ with an equivalent one $\mathbf{C}_{\text{sub}}^{\text{out}}$ acting on the same qubits and time steps. For example, they can include transformations that swap the execution times of commuting gates or

transformations that replace non-native gates with a combination of native gates. The transformation rules used hereafter are reported in Appendix G. Despite their validity, which can be verified through the unitary representation of the involved subcircuits, these rules are not known to form a complete equational theory; i.e., they may not be able to generate all possible equivalent circuits. Finite and complete sets of rewriting rules have recently been established [56,57]. Any finite set of rewriting rules can be incorporated into our paradigm. We create (annihilate) subcircuits by acting with clusters of gate creation (annihilation) operators and label them via the first time step and qubit (t, q) they are acting on. The creation operator for a subcircuit $\mathbf{C}_{\text{sub}}^A$ at (t, q) is then

$$\widehat{\mathbf{C}_{\text{sub}}^A(t, q)}^\dagger := \bigotimes_{t',q'} \hat{g}_{t+t',q+q'}^{(A,t',q')^\dagger}, \quad (4)$$

where $0 \leq t' < N_T^A$, $0 \leq q' < N_Q^A$, and N_T^A and N_Q^A are the number of time steps and qubits included in the subcircuit $\mathbf{C}_{\text{sub}}^A$. $G^{(A,t,q)}$ is the gate executed at time t and qubit q in $\mathbf{C}_{\text{sub}}^A$. Finally, a transformation rule $T = (\mathbf{C}_{\text{sub}}^{\text{in}} \leftrightarrow \mathbf{C}_{\text{sub}}^{\text{out}})$ corresponds to the operator

$$\hat{T} := \sum_{\substack{t \leq N_T - N_T^{\text{in}}, \\ q \leq N_Q - N_Q^{\text{in}}}} \widehat{\mathbf{C}_{\text{sub}}^{\text{out}}(t, q)}^\dagger \widehat{\mathbf{C}_{\text{sub}}^{\text{in}}(t, q)} + \text{H.c.} \quad (5)$$

that replaces any subcircuit $\mathbf{C}_{\text{sub}}^{\text{in}}$ with $\mathbf{C}_{\text{sub}}^{\text{out}}$, and vice versa. Here, we assume that transformations act uniformly on the entire lattice.

E. Compiling via quantum optimization

The optimally compiled circuit can now be singled out on a quantum computer by leveraging ground-state search algorithms such as QA, QAOA, and OC. In the following, we focus on QA-based quantum compilation. The lattice of qudits is initialized in the separable state representing the input circuit. We first prepare a superposition of equivalent circuits by driving the adiabatic transition from the ground state of the single-site Hamiltonian $\hat{H}_0$, i.e., the input circuit, to the ground state of the driving Hamiltonian $\hat{H}_d = \sum_{T \in \mathcal{E}} \hat{T}$. Finally, from the superposition state, we obtain the optimally compiled circuit as the ground state of the infidelity Hamiltonian $\hat{H}_I$ by slowly turning off the driving Hamiltonian. The QA scheme reads

$$\hat{H}(t) = \begin{cases} (1 - \frac{2t}{\tau}) \hat{H}_0 + \frac{2t}{\tau} \hat{H}_d, & \text{if } 0 \leq t \leq \frac{\tau}{2}, \\ (2 - \frac{2t}{\tau}) \hat{H}_d + (\frac{2t}{\tau} - 1) \hat{H}_I, & \text{if } \frac{\tau}{2} \leq t \leq \tau. \end{cases} \quad (6)$$

We stress that the dynamics induced by $H(t)$ is, by construction, constrained within the set of equivalent circuits. Indeed, the driving Hamiltonian $H_d(t)$ never couples states corresponding to nonequivalent circuits. Consequently, $H(t)$ is block diagonal, with each block acting exclusively on the subspace spanned by a family of equivalent circuits. Since $\hat{H}(t)$ is a combination of local and sparse Hermitian operators, it can be efficiently simulated with a universal quantum computer [58,59] (see Appendix B for details on the implementation).

III. NUMERICAL RESULTS

This section presents a numerical validation of our compilation framework. In Sec. III A, we simulate quantum annealing-based compilation using tensor network techniques, showing that the probability of sampling optimal equivalent circuits increases with the annealing time. In Sec. III B, we apply a simulated annealing version of the proposed algorithm to tackle larger compilation instances. Our results reveal the presence of local minima in the cost landscape and further motivate the use of quantum optimization techniques.

A. Quantum annealing

To validate this protocol in a simple setting, we consider a device consisting of a chain of four qubits with nearest-neighbor connectivity and the set of native gates {H, CZ, SWAP}. The one-dimensional input circuit is depicted in Fig. 1(c) and includes six time steps. The optimization problem is encoded in an 8×4 qudit lattice: the two additional steps at the beginning and the end of the computation are included to allow for swap operations. This allows for an automated search of the optimal qubit labeling. The specific gate infidelities and the local equivalence rules defining the driving Hamiltonian are listed in Appendix G, along with all details about the examples illustrated hereafter.

To transform the original input circuit into the compiled circuit shown in Fig. 1(c), the adiabatic dynamics generated by the operators $\hat{T}$ and $\hat{H}_I$ combine many quantum state transitions encoding diverse circuit transformations. Specifically, the dynamics (i) reschedules the execution time of the different gates, (ii) creates and destroys pairs of CZ and H gates that are equivalent to the identity, (iii) synthesizes SWAP gates from equivalent subcircuits composed of many CZ and H gates, and (iv) moves the SWAP gates to the first and last time steps. This last mechanism is enforced by a penalty term in the infidelity Hamiltonian, which assigns high infidelities to non-SWAP gates and zero infidelity to SWAP gates appearing in the first and last time steps of the circuit. Indeed, once the SWAP gates are moved to the external time slices, they do not need to be executed but can be interpreted as a reordering of the quantum wires of the circuit, as in Fig. 1(c). This SWAP-based reordering then optimally associates the quantum memory addresses with the physical qubits of the machine.

We simulate the QA dynamics via a TN emulator [60–62]. We exploit the reflection symmetries of the problem to represent the system as a one-dimensional lattice of eight qudits with local dimension 10 interacting via up to four-body terms (see Appendix C for details). In Fig. 2, we analyze the performance of the QA-based compilation. On the left y axis, we show the infidelity improvement $I_I = 1 - (I_{\text{opt}}/I_{\text{input}})$, where I_{opt} is the infidelity of the compiled circuit and I_{input} is the infidelity of the input circuit. As expected, I_I increases with the annealing time τ . On the right y axis, we plot the probability P_G of sampling the global optimal circuit at the end of the annealing process. This probability increases monotonically with the annealing time, reaching 3% at $\tau \approx 2000$. Thus, for $\tau > 500$, it is sufficient to repeat the computation $O(100)$ times to obtain the optimal solution.

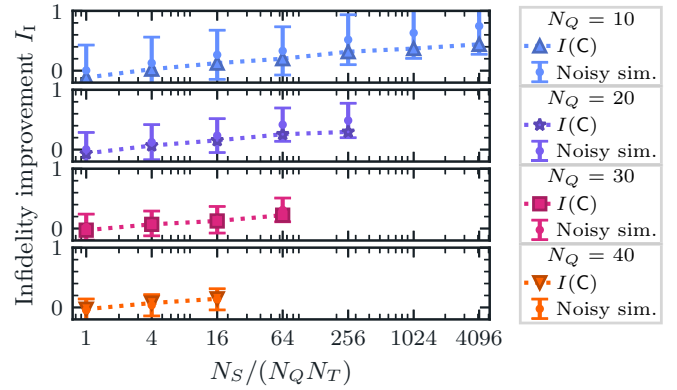


FIG. 3. Infidelity improvement for the QFT circuit compiled with SA. Infidelity improvement $I_I = 1 - (I_{\text{opt}}/I_{\text{input}})$ as a function of the number of steps N_S in the Markov chain over the circuit volume $N_Q N_T$. In the different panels, circuits with different numbers of qubits $N_Q \in \{10, 20, 30, 40\}$. We compare the infidelity improvement based on the cost function $I(C)$ with the infidelity improvement based on the noisy circuit simulation.

B. Simulated annealing

Being limited by the computational complexity of emulating quantum dynamics on a classical computer, we exploit SA as an alternative strategy [63]. The search for a ground state of the infidelity Hamiltonian is performed with a Markov chain Monte Carlo (MCMC) method [63], where the update rules are the equivalence transformations in $\mathcal{E}$.

1. Quantum Fourier transform

First, we consider the quantum Fourier transform (QFT) [64] implemented on a quantum computer consisting of a one-dimensional qubit lattice with nearest-neighbor interactions, and a gate set composed of H, $\text{RZ}(2\pi/2^n)$, CZ, and SWAP gates. We assume that the main sources of errors come from SWAP gates and crosstalk. Figure 3 shows that SA-based compilation succeeds in reducing the expected infidelity by 65%, and compares the infidelity cost function $I(C)$ with the infidelity affecting the execution of uncompiled and compiled circuits on a noisy machine. The latter is simulated through the noisy circuit emulator Quantum Matcha TEA [65]. This comparison validates the correctness of $I(C)$ as an estimate for the expected experimental infidelity, thus confirming that real noisy quantum computations can benefit from the proposed compilation method.

2. Two-dimensional Trotterized Hamiltonian simulation

As a second large-scale example, we compile a Trotterized Hamiltonian simulation (THS) circuit [58]. THS algorithms enable the simulation of Hamiltonian evolution on digital quantum computers, providing an exponential advantage over classical simulation methods for sufficiently entangled systems [66]. We compile a prototypical circuit that simulates the action of a suitable Hamiltonian with nearest-neighbor interactions on a two-dimensional spin lattice. The circuit is illustrated in Appendix D. The target quantum device also consists of a two-dimensional lattice with nearest-neighbor connectivity and parametric gates $\text{RZ}(\theta_1)$, $\text{RX}(\theta_2)$, and $\text{CP}(\theta_3)$.

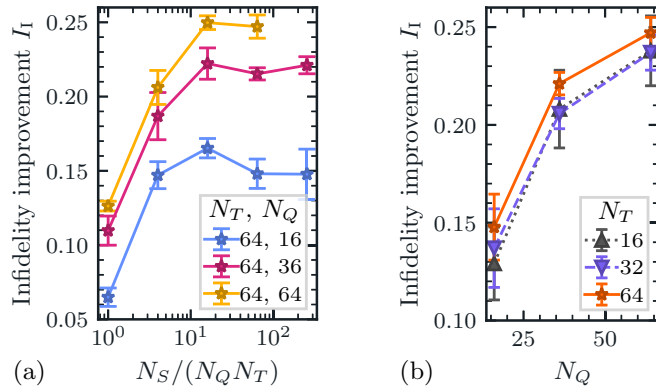


FIG. 4. Infidelity improvement for the Trotterized Hamiltonian simulation circuit compiled with simulated annealing. (a) Infidelity improvement $I_I = 1 - (I_{\text{opt}}/I_{\text{input}})$ as a function of the number of steps N_S in the Markov chain over the circuit volume $N_Q N_T$. We consider circuits with $N_T = 64$ time steps and different numbers of qubits N_Q . (b) Infidelity improvement for the largest number of steps of the Markov chain as a function of the number of qubits N_Q , for different numbers of time steps N_T . Results are averaged over five executions of the SA.

Parametric gates are represented in the qudit lattice encoding by discretizing the parameters θ_i , and assigning a different qudit state to each possible discretized gate. The main sources of infidelity are decoherence affecting idle qubits and crosstalk between pairs of entangling gates, which decay as $\|\mathbf{q} - \mathbf{q}'\|^{-6}$ mimicking the behavior of a Rydberg atom quantum computer [23]. In Fig. 4, we depict the infidelity improvement for different circuit sizes and different numbers of steps in the MCMC. We observe that the improvement I_I at the end of the compilation process increases with the size of the input circuit. For example, I_I reaches about 15% for a 16-qubit circuit, while it reaches about 25% for a 64-qubit circuit. This extensive improvement can be understood by recalling that the global optimal circuit is encoded in the ground state of a translationally invariant many-body Hamiltonian. The ground state of such a system cannot be approximated by the product of the ground states of its subsystems, which is indeed a local minima. Similarly, the improvement achievable by compiling a quantum circuit is greater than that obtained by compiling its individual subcircuits separately. Since quantum annealing can outperform simulated annealing in escaping local minima [27,28,67], these findings provide additional motivation for the approach proposed in this work.

IV. CONCLUSIONS

We have introduced a general paradigm for compiling quantum algorithms with quantum computers. Our approach is demonstrated with quantum annealing but straightforwardly extends to various techniques, including QAOA and optimal control [68]. We focused on optimizing equivalent circuits that implement the same unitary operator. However, as we show in Appendix F, the proposed paradigm also enables the compilation of algorithms that can be implemented by diverse unitary evolutions. For instance, quantum state preparation can be achieved by many different unitary evolutions,

allowing for enlarging the circuit optimization space and potentially decreasing the optimal infidelity. Further, the presented class of compilers can provide automatic support for random circuit compilation, e.g., as needed for error mitigation [69].

Compiling a circuit requires a memory footprint that grows with the number of instructions, meaning that the compilation process demands more memory resources than the target algorithm itself. This challenge is inherent to the compilation problem and is not unique to the quantum setting. It also arises in the classical case, though it is now largely hidden by the extensive and multilevel memory architectures of modern classical devices. Hence, the compilation process will benefit from the addition of storage qubits to the QPU with multiple memory levels as in classical computers [70,71]. Meanwhile, small parts of quantum algorithms, e.g., a logical qubit in error correcting codes, can be compiled on a larger quantum machine already in the NISQ era. The compilation of larger circuits can already benefit from this many-body approach via simulated annealing.

Finally, the compilation paradigm demonstrated here opens a path to studying quantum algorithms as many-body systems, potentially uncovering phenomena like phase transitions and topological features within the space of quantum circuits. Similar phenomena have been previously demonstrated in the control parameter landscape of optimal control theory [72–75].

ACKNOWLEDGMENTS

We acknowledge financial support from Horizon Europe programme HORIZON-CL4-2021-DIGITAL-EMERGING-01-30 via Project No. 101070144 (EuRyQa), the European Union via H2020, the QuantERA projects QuantHEP, T-NISQ, and QTFLAG, the Italian Ministry of University and Research (MUR) via PRIN2022 project TANQU, Fondazione CARIPARO, the INFN project QUANTUM, the Horizon 2020 research and innovation program (Quantum Flagship—PASQuanS2), the European Union—NextGenerationEU Project No. CN00000013—Italian Research Center on HPC, Big Data and Quantum Computing, the Department of Excellence Grant No. 2023-2027 Quantum Frontiers, from the German Federal Ministry of Education and Research (BMBF) via the funding program quantum technologies—from basic research to market—project QRydDemo, and the World Class Research Infrastructure—Quantum Computing and Simulation Center (QCSC) of Padova University. We acknowledge computational resources by the Cloud Veneto, as well as computation time on Cineca’s Leonardo machine.

DATA AVAILABILITY

The data that support the findings of this article are openly available [76,77].

APPENDIX A: INFIDELITY FUNCTION DERIVATION

In this Appendix, we derive the infidelity function $I(\mathbf{C})$ defined in Eq. (1). We assume that the probability $P(\mathbf{C})$ that

the circuit $\mathbf{C}$ returns the right output state can be written as

$$P(\mathbf{C}) = \prod_{G \in \mathbf{C}} F(G), \quad (\text{A1})$$

where $F(G)$ is the fidelity of any gate G in the circuit. This fidelity is eventually influenced by the parallel execution of other gates. We also associate a fidelity with idle qubits to encode, for example, the effects of decoherence. To maximize the success probability $P(\mathbf{C})$, we minimize the infidelity function

$$I(\mathbf{C}) := -\log(P(\mathbf{C})) = -\sum_{G \in \mathbf{C}} \log(F(G)). \quad (\text{A2})$$

We introduce the gate infidelity $I(G) := 1 - F(G)$. For any machine capable of executing large circuits with acceptable fidelity, we can assume $I(G) \ll 1$. This implies that $\log(F(G)) = \log(1 - I(G)) \approx -I(G)$. Replacing this approximation in Eq. (A2), we obtain

$$I(\mathbf{C}) = \sum_{G \in \mathbf{C}} I(G). \quad (\text{A3})$$

Finally, we write the infidelity $I(G)$ as the sum of i_G , i.e., the infidelity of the gate when executed alone, and $x_G(\|q - q'\|)$, accounting for the crosstalks. We also define the infidelity term i_{Idle} for idle qubits. Replacing these equivalences in the last equation, we obtain that the infidelity function is

$$I(\mathbf{C}) = \sum_{G \in \mathbf{C}} \left(i_G + \sum_{q, q' | G(t, q) = G(t, q')} x_G(\|q - q'\|) \right). \quad (\text{A4})$$

APPENDIX B: COMPARATIVE RESOURCE ANALYSIS

In this Appendix, we compare the resources needed for the execution of the quantum algorithm with the resources required for the implementation of classical circuit compilation methods.

A fair comparison with classical strategies should take into account two main factors: (1) the computational effort needed to simulate one step of the time evolution, e.g., via Trotterized adiabatic evolution or QAOA, and (2) the number of optimization steps needed to reach a sufficiently low expected infidelity.

The Hamiltonians involved at each time step of the evolution are the infidelity Hamiltonian, the driving Hamiltonian, and the initial parent Hamiltonian. These Hamiltonians are, respectively, expressed as sums of $N_I \times V$, $N_D \times V$, and $N_P \times V$ operators of the form $|\alpha'_1 \cdots \alpha'_l\rangle \langle \alpha_1 \cdots \alpha_l|$, where V is the volume of the circuit, N_I is the number of terms per site in the infidelity Hamiltonian, N_D is the number of rewrite rules, $N_P = 1$ is the number of local fields used to prepare the initial state, and l is the size of a rule or a local term in the infidelity Hamiltonian.

The operators involved in a time step of QAOA or Trotterized adiabatic evolution are written as

$$W = e^{-i\theta |\alpha'_1 \cdots \alpha'_l\rangle \langle \alpha_1 \cdots \alpha_l|}. \quad (\text{B1})$$

These operators can be implemented using X gates and multi-controlled not gates to flip an ancilla qubit when the state locally matches $|\alpha_1 \cdots \alpha_l\rangle$, followed by the application of

$\mathcal{O}(l)$ single-qubit gates controlled by the ancilla to transform $|\alpha_1 \cdots \alpha_l\rangle$ into $e^{-i\theta} |\alpha'_1 \cdots \alpha'_l\rangle$, and finally followed by the uncomputation of the ancilla.

Multicontrolled gates can be executed efficiently on universal quantum computers using a linear number of elementary gates and ancilla qubits [78], or they may be implemented natively on hardware platforms such as Rydberg atom arrays and superconducting circuits [79].

Thus, implementing each W operator requires $\mathcal{O}(l)$ elementary gates, and a single time step of the evolution involves a total number of elementary quantum operations that scales as $\mathcal{O}(l \cdot V)$.

On the classical side, the implementation of circuit compilation algorithms typically requires the application of equivalence rules to a data structure representing the circuit and the evaluation of the infidelity cost function. The classical computational effort required to perform these operations also increases linearly with the number of strings needed to describe the infidelity cost function, i.e., the number of operators in the infidelity Hamiltonian, and with the number of rules. Applying a transformation rule to the circuit and evaluating a cost function term are both operations whose classical cost is linear in the size of the rule or term. Finally, let us note that a similar cost is required to reduce operations on characters to Boolean logic on the classical side, and operations on qudits to standard quantum circuit operations on the quantum side. Thus, the cost of both classical and quantum operations involved in each optimization step is $\mathcal{O}(l \cdot V)$, and the comparison of computational efforts reduces to a comparison of the number of steps needed to reach a satisfactory solution.

The number of time steps (in the quantum case) and the number of classical optimization iterations (in the classical case) needed to reach a good solution depend on the specific problem instance. This is a general feature of all quantum optimization heuristics. Demonstrating individual instances with a clear quantum speedup is a separate and high-impact challenge [27–29], which lies beyond the scope of this work. In this regard, our proposed compilation method—like all heuristic quantum optimization algorithms—is not designed to consistently outperform classical approaches on all instances, but rather to show a polynomial speedup in specific cases, such as when the cost function features tall, narrow energy barriers [67] or a flat low-energy landscape [29].

APPENDIX C: DETAILS OF QUANTUM ANNEALING FOR EXAMPLE I

In the main text, we simulate the annealing-based quantum compilation of the input circuit in Fig. 1(c). Circuits are encoded as states of an 8×4 qudit lattice representing the time t and the qubit q as (t, q) . We have a local dimension of $d = 5$ to represent the idle state Idle , H gates, range one CZ gates, and range one SWAP gates, along with a fictitious gate BUSY that designates the target adjacent qubit $q + 1$ for both CZ and SWAP. The allowed gates can be executed, respectively, with infidelities $i_H = 0.5 \times 10^{-5}$, $i_{\text{CZ}} = 1.0 \times 10^{-5}$, and $i_{\text{SWAP}} = 1.5 \times 10^{-5}$, while idle gates are affected by an infidelity $i_{\text{Idle}} = 0.5 \times 10^{-5}$. The first and last time steps are reserved for swapping area, and are associated

with 0 infidelity for SWAP and Idle gates, and with infidelity 5.0×10^{-5} for any configuration containing different gates.

With this encoding, we need to simulate the evolution in a Hilbert space whose dimension is 5^{32} , equivalent to a 75-qubit system. To reduce the computational complexity of the problem, we restrict the search to the space of circuits that are symmetric under the inversion of the qubit axis. This implies building symmetric four-qubit equivalence rules and generating the driving Hamiltonian from these transformations, which are listed in Appendix G 1.

Analogously, we symmetrize the infidelity Hamiltonian, while the initial Hamiltonian inherits its symmetry from the initial state. In this setting, for each time coordinate of the two-dimensional lattice, the time-evolved system state is spanned by the following ten states:

- (1) $|\text{Idle} \otimes \text{Idle} \otimes \text{Idle} \otimes \text{Idle}\rangle$,
- (2) $|\text{H} \otimes \text{Idle} \otimes \text{Idle} \otimes \text{H}\rangle$,
- (3) $|\text{Idle} \otimes \text{H} \otimes \text{H} \otimes \text{Idle}\rangle$,
- (4) $|\text{H} \otimes \text{H} \otimes \text{H} \otimes \text{H}\rangle$,
- (5) $|\text{Idle} \otimes \text{CZ} \otimes \text{BUSY} \otimes \text{Idle}\rangle$,
- (6) $|\text{H} \otimes \text{CZ} \otimes \text{BUSY} \otimes \text{H}\rangle$,
- (7) $|\text{Idle} \otimes \text{SWAP} \otimes \text{BUSY} \otimes \text{Idle}\rangle$,
- (8) $|\text{H} \otimes \text{SWAP} \otimes \text{BUSY} \otimes \text{H}\rangle$,
- (9) $|\text{CZ} \otimes \text{BUSY} \otimes \text{CZ} \otimes \text{BUSY}\rangle$,
- (10) $|\text{SWAP} \otimes \text{BUSY} \otimes \text{SWAP} \otimes \text{BUSY}\rangle$.

Note that the CZ and SWAP gates are symmetric. In this way, we can represent the annealing dynamics as the evolution of a linear system of eight qudits with a local dimension of 10, equivalent to a $\log_2(10^8) \approx 27$ -qubit system. The simulation time and computational complexity are driven by the sheer number of multiqubit terms that encode the different rules; the representation of the Hamiltonian is exact and contains no truncation. To simulate the quantum evolution, we model the system state using a matrix product state (MPS) and evolve it through a time-dependent variational principle (TDVP) by Quantum TEA Leaves [48,60], along with the collapsed representation. The effectiveness of MPS in modeling adiabatic ground-state search has been previously analyzed in Ref. [80].

We utilize MPS with bond dimensions $\chi \in \{32, 64, 128\}$ and divide the time evolution into N_S time steps with duration $\delta_t = \tau/N_S$, where τ is the annealing time and $N_S \in \{1000, 2000, 4000\}$. In Fig. 5, we depict the discrepancy between the simulations with bond dimensions $\chi \in \{32, 64\}$ against the simulations with bond dimension $\chi = 128$. For $\tau \leq 2000$, the relative error is smaller than 10^{-2} .

As a further convergence test, we perform sampling on the computational basis [62] of the annealed states, and we check whether each sampled circuit is equivalent to the input circuit, i.e., represents the same unitary operator as the input circuit. Thus, we estimate the ratio between the sampled probability of obtaining an equivalent circuit, P_{equiv} , and the total sampled probability P_{tot} . The equivalence of each sampled circuit with the input circuit is verified by checking that $U_{C_{\text{opt}}}^\dagger U_{C_{\text{input}}} \approx \mathbb{1}$ up to a phase factor, where $U_{C_{\text{input}}}$ and $U_{C_{\text{opt}}}$ are the unitary operators representing the input and optimized circuits, respectively. Since the explored dynamics is constrained to the space of equivalent circuits, we expect that $1 - (P_{\text{equiv}}/P_{\text{tot}}) \approx 0$ until numerical errors due to the finite number of time steps or a small bond dimension of the MPS start accumulating. We verify the behavior of $1 - (P_{\text{equiv}}/P_{\text{tot}})$

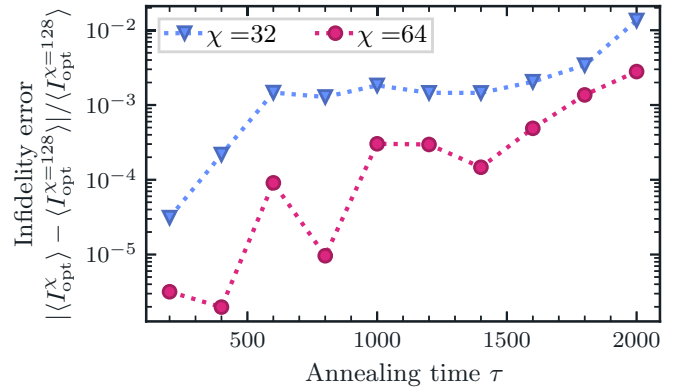


FIG. 5. Infidelity energy convergence for different bond dimensions. We plot the relative difference between the infidelity energy evolution evaluated with bond dimensions $\chi \in \{32, 64\}$ against bond dimension $\chi = 128$ as a function of the annealing time τ . The number of time steps is $N_S = 4000$.

in Fig. 6. In Fig. 6(a), we explore different numbers of steps, and in Fig. 6(b) we explore different bond dimensions. We observe that a number of steps $N_S \geq 4000$ and bond dimensions $\chi \geq 64$ are sufficient to achieve $1 - (P_{\text{equiv}}/P_{\text{tot}}) < 10^{-8}$, which is negligible compared to the magnitude of our main figure of merit, i.e., the probability of sampling the optimal circuit illustrated in Fig. 2 of the main text.

To explain the relatively small bond dimension, note that the investigated dynamics only explores quantum superpositions of states representing circuits equivalent to the initial one. We consider a bipartition of the system into two

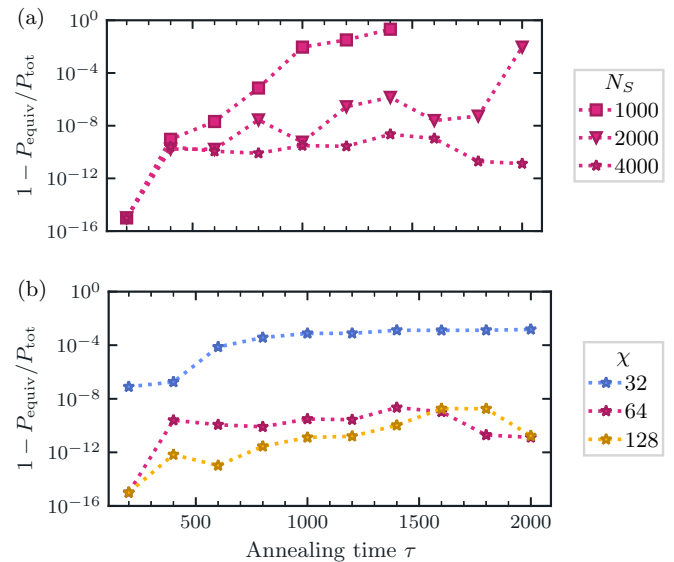


FIG. 6. Probability of not measuring an equivalent circuit after the quantum annealing-based compilation, as simulated via MPS-TDVP. We analyze $1 - (P_{\text{equiv}}/P_{\text{tot}})$, where P_{equiv} is the sampled probability of measuring an equivalent circuit and P_{tot} is the total sampled probability, as a function of the annealing time τ . In Panel (a), we fix the bond dimension $\chi = 64$ and we consider different numbers of steps N_S . In Panel (b), we fix the number of steps $N_S = 4000$ and we consider different bond dimensions χ .

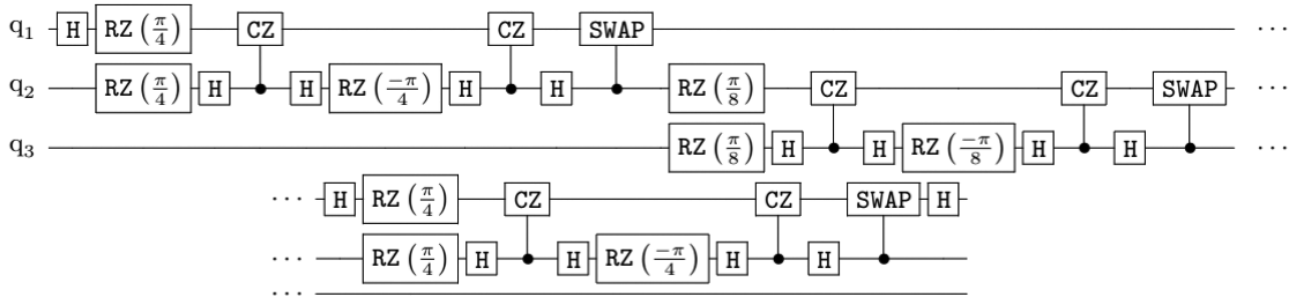


FIG. 8. Unoptimized three-qubit implementation of the QFT based on the gates i_H , i_{RZ} , i_{CZ} , and i_{SWAP} . We allow the implementation of SWAP gates with a relatively high infidelity, but without the crosstalk error affecting the parallel execution of CZ gates.

where transitions of the Markov chain can occur, reducing the number of steps needed for the optimization. The local transformation rules generating the Markov chain for the SA are listed in Appendix G 3.

To prove that the infidelity function $I(C)$ optimized in the annealing procedure reflects the actual fidelity of the quantum algorithm when executed on a noisy machine, we simulate the QFT circuit with noisy gates. The simulation is carried out by Quantum Matcha TEA [65], a tensor network emulator for quantum circuits, specifically using the MPS ansatz. To simulate the noise, we employ the quantum trajectories method [81]: we run the circuit for n_{traj} independent times, averaging the observables over randomly distributed unitary evolutions. We adopt a simplified error model for the gates, since we are not focused on the physical representation of the noise at this stage, but only in its magnitude in this work. Given an m -qubit gate G , we first represent it as a parametric gate $\tilde{G}(\theta)$ and then add Gaussian noise to the phase:

$$G \rightarrow \tilde{G}(\theta + u), \quad u \sim \mathcal{N}(0, \sigma), \quad (\text{E1})$$

where $\mathcal{N}(0, \sigma)$ is a normal distribution with mean 0 and standard deviation σ . The gates are specifically handled in the following way:

- (1) We decompose the H gate into

$$H \rightarrow X R_y\left(\frac{\pi}{2} + u\right), \quad (\text{E2})$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}, \quad (\text{E3})$$

where X is the Not gate.

- (2) We perturb the RZ and CP gates as follows:

$$RZ(\theta) \rightarrow RZ(\theta + u), \quad CP(\theta) \rightarrow CP(\theta + u). \quad (\text{E4})$$

- (3) We decompose the SWAP gate into a circuit of H and CZ gates. We then perturb CZ as $CZ \rightarrow CP(\pi + u)$.

- (4) We represent the crosstalk as a four-qubits controlled-phase gate, which is close to the identity, i.e., CCCP(0 + u).

The standard deviation is chosen for each gate such that its averaged infidelity $I_{\tilde{G}}$ is

$$I(\tilde{G}) = 1 - \frac{1}{M} \sum_{i=0}^M |\langle \psi_i | G^\dagger \tilde{G}_i | \psi_i \rangle|^2 = 10^{-7}, \quad (\text{E5})$$

where $M = 1000$, $|\psi_i\rangle$ is a random state of m qubits, and $\tilde{G}_i$ represents a random sample from $\tilde{G}$. The infidelity of the single gate is later adjusted by the error weight of the specific gate

for that specific simulation. In Fig. 9, we report the assessment of the error magnitude as a function of the standard deviation of the Gaussian random parameters.

Once this setup is defined, we can test the infidelity of a noisy quantum circuit C , encoded by the corresponding random unitary $\tilde{U}$, versus the unitary U representing a noiseless execution of the circuit. We obtain the infidelity of the circuit $I^{\text{sim}}(C)$ as an average over $n_{\text{traj}} = 100$ trajectories:

$$I^{\text{sim}}(C) = 1 - \frac{1}{n_{\text{traj}}} \sum_{i=0}^{n_{\text{traj}}} |\langle \phi_i | U^\dagger \tilde{U}_i | \phi_i \rangle|^2, \quad (\text{E6})$$

where $|\phi_i\rangle$ are random separable states and $\tilde{U}_i$ are random samples from $\tilde{U}$. The maximum bond dimension used in the simulation is $\chi_{\text{max}} = 128$.

We stress that we select the QFT for this test since it can be efficiently simulated classically with tensor networks [82]. However, the simulation is not trivial due to the presence of the four-qubit interactions generated by the crosstalk. Thus, it is important to certify that the infidelity we show in Fig. 3 is generated by the noisy gates and not by a tensor network truncation. By monitoring the truncation that happens in each step, i.e., in the application of each gate, it is possible to provide a lower bound on the fidelity of the state due to the finite bond dimension $\chi = 128$. Based on the state after $(i - 1)$ multiqubit gates, we apply the i th multiqubit gate

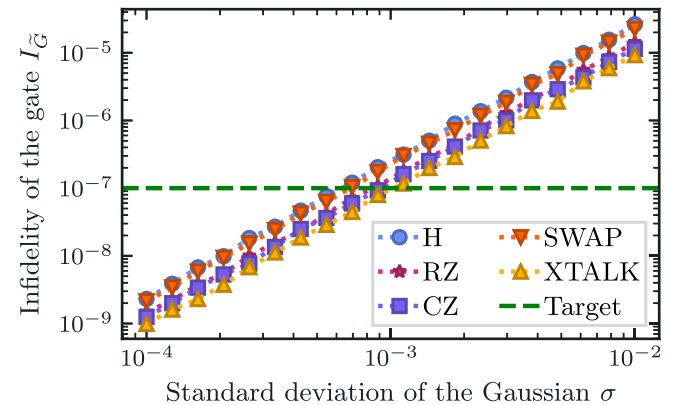


FIG. 9. Infidelity of noisy gates with random Gaussian parameters. Infidelity of the gate $I_{\tilde{G}}$ for different standard deviations of the Gaussian distribution σ . We highlight with the green dashed line the target infidelity of 10^{-7} .

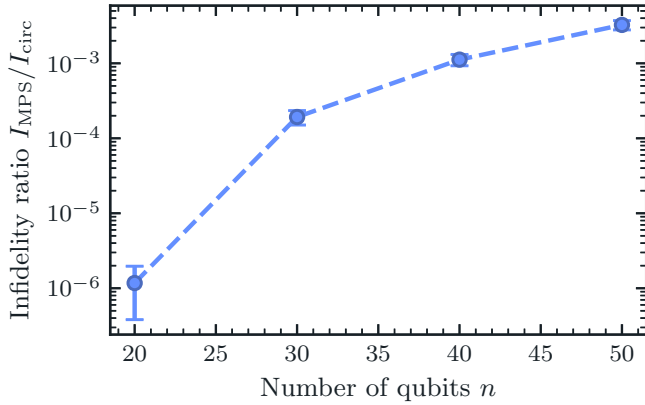


FIG. 10. Infidelity from tensor network truncation error vs infidelity from noisy gates. The ratio between the infidelity coming from the tensor network truncation I_{MPS} and the infidelity coming from the noisy gates I_{circ} as a function of the number of qubits. The ratio increases monotonically with the number of qubits, but it is always smaller than 10^{-2} for the system sizes considered here.

and obtain $|\psi_{\text{exact}}^i\rangle$ without truncation. Then, we apply our truncation scheme and obtain $|\psi_{\text{trunc}}^i\rangle$. Recall that one-qubit gates do not imply additional approximation errors; therefore, the fidelity of the i th multiqubit is

$$F_i = |\langle \psi_{\text{exact}}^i | \psi_{\text{trunc}}^i \rangle|^2 = \left| \sum_{\alpha=1}^{\chi} \lambda_{\alpha}^2 \right|^2 = \left| 1 - \sum_{\alpha=\chi+1}^{\chi_{\text{exact}}} \lambda_{\alpha}^2 \right|^2, \quad (\text{E7})$$

where λ_{α} are the singular values of the Schmidt decomposition of the state for the bond where the i th two-qubit gate was applied, ordered by increasing magnitude, χ_s is the bond dimension of the MPS state, and χ_{exact} is the bond dimension needed to exactly represent the state. The infidelity of the tensor network I_{MPS} after application of the i th multiqubit gate is lower bounded as follows [83]:

$$I_{\text{MPS}} \leq 1 - \prod_{i=1}^{j-1} F_i. \quad (\text{E8})$$

Thus, we use this metric to ensure that the corrections due to the tensor network truncation are negligible with respect to the errors coming from the noise. In Fig. 10, we show the ratio between the infidelity of the tensor network and the total infidelity of the simulation as defined in Eq. (3). The infidelity I_{MPS} generated by the tensor network truncation is always more than two orders of magnitude smaller than the infidelity generated by the noise. The point for $n = 10$ qubits is not shown in Fig. 10, since the simulation is exact for this number of qubits. We can thus state that the results we show in Fig. 3 are not influenced by the finite bond dimension.

APPENDIX F: COMPILING MORE GENERAL ALGORITHMS

In the main text, we exclusively consider the compilation of quantum algorithms identifiable by a unique unitary operator, such as the quantum Fourier transform. However, several algorithms diverge from this framework. Examples include

quantum state preparation and circuits involving ancillary qubits. For instance, the equivalence of two state-preparation algorithms hinges on their ability to prepare the same target state from a computational basis state, irrespective of their impact on other basis states. Similarly, if two circuits involve ancillary qubits, their equivalence depends solely on the output of nonancillary qubits.

Here, we showcase the versatility of our approach by extending it to encompass these diverse quantum algorithm classes. The proposed approach is based on extending the equivalence transformations in $\mathcal{E}$ to extend the generated equivalence classes of circuits.

First, let us consider state preparation algorithms. The objective is to prepare a target state $|\psi_1\rangle$ starting from a state of the computational basis $|\psi_0\rangle$. When optimizing an input circuit that synthesizes the unitary operator U to achieve $U|\psi_0\rangle = |\psi_1\rangle$, it is also essential to explore circuits that synthesize a distinct unitary $U' \neq U$ while still satisfying $U'|\psi_0\rangle = |\psi_1\rangle$. Let us illustrate this with an example involving the compilation of a quantum circuit designed to generate the Greenberger-Horne-Zeilinger state $|\psi_1\rangle = (1/\sqrt{2})(|0 \dots 0\rangle + |1 \dots 1\rangle)$ from the computational basis state $|\psi_0\rangle = |0 \dots 0\rangle$. During the compilation of the input circuit, we can add or remove Z gates at the initial time step without altering the resulting evolution, as these gates only introduce a global phase to the initial state. Similarly, we can freely add or remove pairs of Z gates executed in parallel at the final time step without impacting the final state of the circuit. Thus, we expand the previously introduced transformations set $\mathcal{E}$ with additional transformations $\mathcal{E}_{\text{in-sym}}$ acting only on the first time steps of the circuit. These transformations create or annihilate subcircuits whose action does not change the input state. Similarly, we can introduce transformations $\mathcal{E}_{\text{out-sym}}$ acting on the last time steps in cases where the target state $|\psi_1\rangle$ is known. These transformations create or annihilate subcircuits whose action leaves the target output state unchanged. By incorporating both input and output transformations in an extended set of transformation generators, we substantially expand the space of equivalent circuits. The generated circuits perform quantum state preparation through unitary transformations, expressed as $U' = \prod_{i,j} S_{\text{in}}^{[i]\dagger} U S_{\text{out}}^{[j]}$, where $S_{\text{in}}^{[i]}$ and $S_{\text{out}}^{[j]}$ represent unitary operators that preserve the input and target states, respectively.

When optimizing a quantum algorithm involving ancillary qubits that are not measured at the computation's conclusion, like measurement-free error correction [84,85], we can enlarge the space of equivalent circuits even further. In this case, the set of transformation generators is expanded to include the subset $\mathcal{E}_{\text{out-anc}}$. These transformations create and annihilate arbitrary gates only in a region of the lattice corresponding to the last time steps and the ancillary qubits. This extension generates quantum circuits that affect computational qubits similarly while altering the final state of ancillary qubits. We can also accommodate more complicated scenarios. For example, let us consider the case in which ancillary gates are prepared in a fixed state. In this situation, we can expand the set of generators to create or remove state-preserving subcircuits at the initial time steps, specifically targeting the ancillary qubits.

In general, expanding the set of generators to accommodate a less restrictive definition of circuit equivalence enables access to circuits that can be implemented with lower infidelity. We will explore this perspective in future work.

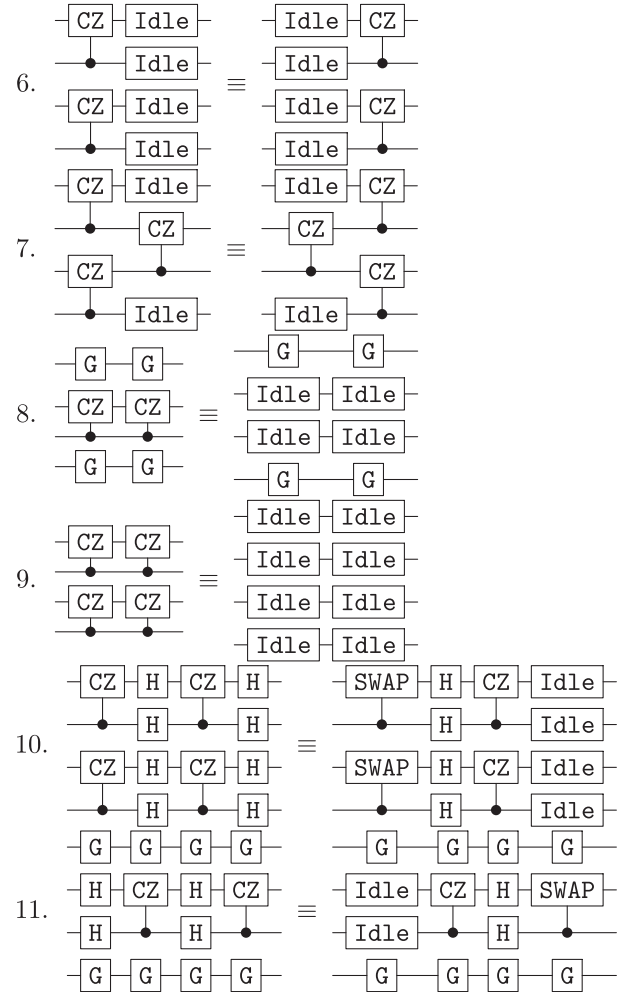
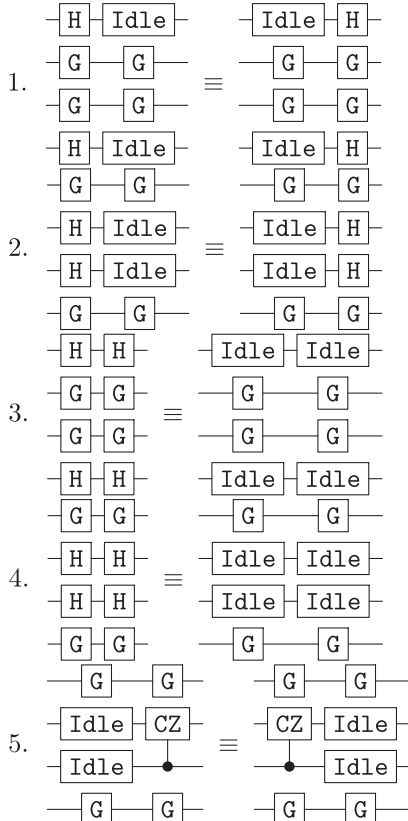
APPENDIX G: TRANSFORMATION RULES

Here, we enumerate all the transformation rules that generate the quantum and simulated annealing for the examples outlined in the main text. These rules are specific to the quantum hardware on which the circuit needs to be implemented. They encode the relationship between the native gates of the machine and those composing the input uncompiled circuit.

The complexity of the driving Hamiltonian in quantum annealing, in terms of the number of interactions and their range, depends on the equivalence rules and is reflected in the difficulty of simulating the quantum annealing dynamics using tensor networks. The number of interactions in the Hamiltonian, each corresponding to an equivalence rule applied to a region of the circuit, increases linearly with both the number of equivalence rules and the volume of the compiled circuit. Each of these interactions is an m -body operator, where m represents the volume of the subcircuits defining the equivalence rule.

1. Example I

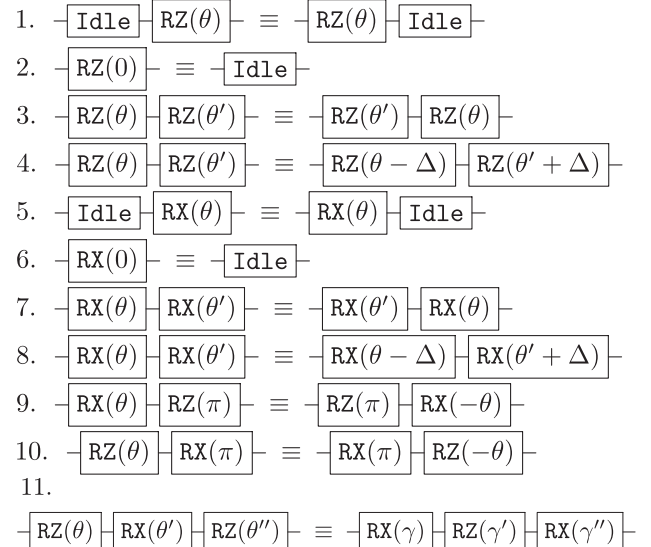
We consider the following local circuit equivalences to construct the driving Hamiltonian $\hat{H}_d$ for QA-based compilation of the circuit in Fig. 1(c):



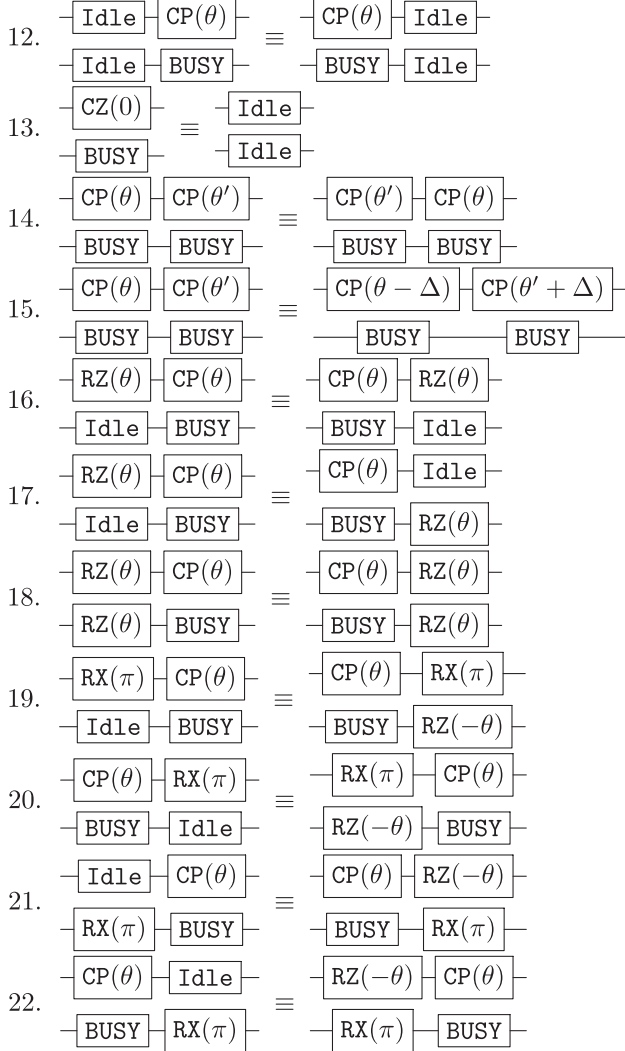
The G gates can be replaced with any arbitrary gate allowed by the considered architecture, such as the H , CZ , and $SWAP$ gates, or by an Idle qubit. This replacement must avoid any configuration that violates the qubit inversion symmetry.

2. Example II

The local transformation rules generating the Markov chain for the SA-based compilation of THS are



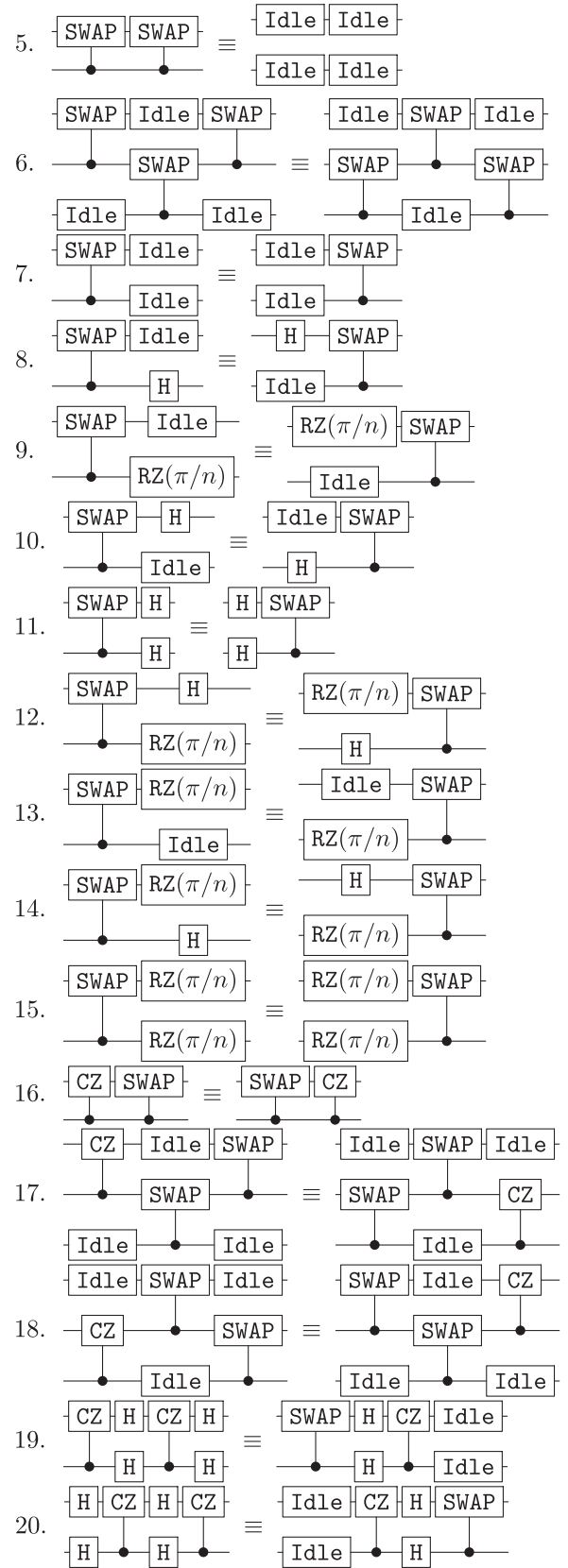
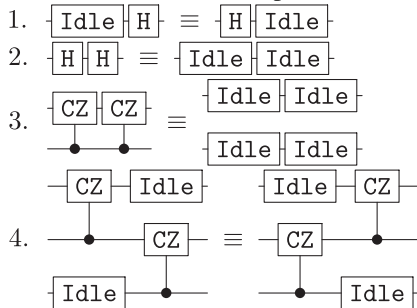
where $(\theta, \theta', \theta'') \leftrightarrow (\gamma, \gamma', \gamma'')$ are two different Euler representations of the same rigid body rotations. The correspondence is generated by the Euler transformations between the reference frames $x - z - x$ and $z - x - z$.



where angles are modulo 2π , as this reflects the periodicity of the involved gates up to a global phase factor, and $\Delta \in \{\pm\pi, \pm\pi/2, \pm\pi/4, \pm\pi/8\}$.

3. Example III

The local transformation rules generating the Markov chain for the SA-based compilation of QFT are



- [1] F. T. Chong, D. Franklin, and M. Martonosi, Programming languages and compiler design for realistic quantum hardware, *Nature (London)* **549**, 180 (2017).
- [2] M. Maronese, L. Moro, L. Rocutto, and E. Prati, Quantum compiling, in *Quantum Computing Environments*, edited by S. S. Iyengar, M. Matriani, and K. L. Kumar (Springer, Cham, Switzerland, 2022), pp. 39–74.
- [3] L. Schmid, D. F. Locher, M. Rispler, S. Blatt, J. Zeiher, M. Müller, and R. Wille, Computational capabilities and compiler development for neutral atom quantum processors—Connecting tool developers and hardware experts, *Quantum Sci. Technol.* **9**, 033001 (2024).
- [4] A. Botea, A. Kishimoto, and R. Marinescu, On the complexity of quantum circuit compilation, in *Proceedings of the Eleventh International Symposium on Combinatorial Search (SoCS 2018)* (AAAI Press, Washington, DC, 2018), pp. 138–142.
- [5] M. Saeedi, R. Wille, and R. Drechsler, Synthesis of quantum circuits for linear nearest neighbor architectures, *Quantum Inf. Proc.* **10**, 355 (2010).
- [6] Y. Nam, N. J. Ross, Y. Su, A. M. Childs, and D. Maslov, Automated optimization of large quantum circuits with continuous parameters, *npj Quantum Inf.* **4**, 23 (2018).
- [7] W. Finigan, M. Cubeddu, T. Lively, J. Flick, and P. Narang, Qubit allocation for noisy intermediate-scale quantum computers, [arXiv:1810.08291](https://arxiv.org/abs/1810.08291).
- [8] 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)* (ACM, New York, 2019), pp. 1001–1014.
- [9] R. Wille, L. Burgholzer, and A. Zulehner, 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, New York, 2019), pp. 1–6.
- [10] Y. Shi, N. Leung, P. Gokhale, Z. Rossi, D. I. Schuster, H. Hoffmann, and F. T. Chong, Optimized compilation of aggregated instructions for realistic quantum computers, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (ACM, New York, NY, 2019), pp. 1031–1044.
- [11] S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington, and R. Duncan, *t|ket*: A retargetable compiler for NISQ devices, *Quantum Sci. Technol.* **6**, 014003 (2021).
- [12] X. Zhou, S. Li, and Y. Feng, Quantum circuit transformation based on simulated annealing and heuristic search, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **39**, 4683 (2020).
- [13] T. Fösel, M. Y. Niu, F. Marquardt, and L. Li, Quantum circuit optimization with deep reinforcement learning, [arXiv:2103.07585](https://arxiv.org/abs/2103.07585).
- [14] E. Peters, P. Shyamsundar, A. C. Y. Li, and G. Perdue, Qubit assignment using time reversal, *PRX Quantum* **3**, 040333 (2022).
- [15] C. Campbell, F. T. Chong, D. Dahl, P. Frederick, P. Goiporia, P. Gokhale, B. Hall, S. Issa, E. Jones, S. Lee, A. Litteken, V. Omole, D. Owusu-Antwi, M. A. Perlin, R. Rines, K. N. Smith, N. Goss, A. Hashim, R. Naik, and E. Younis *et al.*, Superstaq: Deep optimization of quantum programs, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Bellevue, WA (IEEE, Piscataway, NJ, 2023), pp. 1020–1032.
- [16] E. Younis, K. Sen, K. Yelick, and C. Iancu, QFAST: Conflating search and numerical optimization for scalable quantum circuit synthesis, in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Broomfield, CO (IEEE, Piscataway, NJ, 2021), pp. 232–243.
- [17] 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).
- [18] C. McKeever and M. Lubasch, Classically optimized Hamiltonian simulation, *Phys. Rev. Res.* **5**, 023146 (2023).
- [19] Y. Kikuchi, C. McKeever, L. Coopmans, M. Lubasch, and M. Benedetti, Realization of quantum signal processing on a noisy quantum computer, *npj Quantum Inf.* **9**, 93 (2023).
- [20] C. McKeever and M. Lubasch, Towards adiabatic quantum computing using compressed quantum circuits, *PRX Quantum* **5**, 020362 (2024).
- [21] Y. Ding, P. Gokhale, S. F. Lin, R. Rines, T. Propson, and F. T. Chong, Systematic crosstalk mitigation for superconducting qubits via frequency-aware compilation, in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Los Alamitos, CA (IEEE, Piscataway, NJ, 2020), pp. 201–214.
- [22] P. Baßler, M. Zipper, C. Cedzich, M. Heinrich, P. H. Huber, M. Johanning, and M. Kliesch, Synthesis of and compilation with time-optimal multiqubit gates, *Quantum* **7**, 984 (2023).
- [23] D. Jaschke, A. Pagano, S. Weber, and S. Montangero, *Ab initio* tree-tensor-network digital twin for quantum computer benchmarking in 2D, *Quantum Sci. Technol.* **9**, 035055 (2024).
- [24] 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).
- [25] J. M. Baker, A. Litteken, C. Duckering, H. Hoffman, H. Bernien, and F. T. Chong, Exploiting long-distance interactions and tolerating atom loss in neutral atom quantum architectures, in *Proceedings of the 48th Annual International Symposium on Computer Architecture (ISCA '21)* (IEEE Press, Piscataway, NJ, 2021), pp. 818–831.
- [26] V. A. Alfred, S. L. Monica, and D. U. Jeffrey, Compilers principles, *Techniques and Tools* (Pearson Education, Upper Saddle River, NJ, 2007).
- [27] S. Ebadi *et al.*, Quantum optimization of maximum independent set using Rydberg atom arrays, *Science* **376**, 1209 (2022).
- [28] T. Albash and D. A. Lidar, Demonstration of a scaling advantage for a quantum annealer over simulated annealing, *Phys. Rev. X* **8**, 031016 (2018).
- [29] M. Cain, S. Chattopadhyay, J.-G. Liu, R. Samajdar, H. Pichler, and M. D. Lukin, Quantum speedup for combinatorial optimization with flat energy landscapes, [arXiv:2306.13123](https://arxiv.org/abs/2306.13123).
- [30] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse Ising model, *Phys. Rev. E* **58**, 5355 (1998).
- [31] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem, *Science* **292**, 472 (2001).
- [32] G. E. Santoro, R. Martonak, E. Tosatti, and R. Car, Theory of quantum annealing of an Ising spin glass, *Science* **295**, 2427 (2002).

- [33] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, Perspectives of quantum annealing: Methods and implementations, *Rep. Prog. Phys.* **83**, 054401 (2020).
- [34] T. Albash and D. A. Lidar, Adiabatic quantum computation, *Rev. Mod. Phys.* **90**, 015002 (2018).
- [35] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, [arXiv:quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106).
- [36] J. Werschnik and E. K. U. Gross, Quantum optimal control theory, *J. Phys. B: At. Mol. Opt. Phys.* **40**, R175 (2007).
- [37] C. Brif, R. Chakrabarti, and H. Rabitz, Control of quantum phenomena: Past, present, and future, *New J. Phys.* **12**, 075008 (2010).
- [38] D. Dong and I. R. Petersen, Quantum control theory and applications: A survey, *IET Control Theory Appl.* **4**, 2651 (2010).
- [39] P. Doria, T. Calarco, and S. Montangero, Optimal control technique for many-body quantum dynamics, *Phys. Rev. Lett.* **106**, 190501 (2011).
- [40] T. Caneva, T. Calarco, and S. Montangero, Chopped random-basis quantum optimization, *Phys. Rev. A* **84**, 022326 (2011).
- [41] S. Lloyd and S. Montangero, Information theoretical analysis of quantum optimal control, *Phys. Rev. Lett.* **113**, 010502 (2014).
- [42] C. P. Koch, U. Boscain, T. Calarco, G. Dirr, S. Filipp, S. J. Glaser, R. Kosloff, S. Montangero, T. Schulte-Herbrüggen, D. Sugny, and F. K. Wilhelm, Quantum optimal control in quantum technologies. Strategic report on current status, visions and goals for research in Europe, *EPJ Quantum Technol.* **9**, 19 (2022).
- [43] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [44] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, Quantum approximate optimization algorithm for MaxCut: A fermionic view, *Phys. Rev. A* **97**, 022304 (2018).
- [45] E. Farhi and A. W. Harrow, Quantum supremacy through the quantum approximate optimization algorithm, [arXiv:1602.07674](https://arxiv.org/abs/1602.07674).
- [46] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices, *Phys. Rev. X* **10**, 021067 (2020).
- [47] K. Blekos, D. Brand, A. Ceschini, C.-H. Chou, R.-H. Li, K. Pandya, and A. Summer, A review on quantum approximate optimization algorithm and its variants, *Phys. Rep.* **1068**, 1 (2024).
- [48] J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, and F. Verstraete, Unifying time evolution and optimization with matrix product states, *Phys. Rev. B* **94**, 165116 (2016).
- [49] U. Schollwöck, The density-matrix renormalization group, *Rev. Mod. Phys.* **77**, 259 (2005).
- [50] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, *Ann. Phys.* **326**, 96 (2011).
- [51] R. Orús, A practical introduction to tensor networks: Matrix product states and projected entangled pair states, *Ann. Phys.* **349**, 117 (2014).
- [52] M. C. Bañuls, Tensor network algorithms: A route map, *Annu. Rev. Condens. Matter Phys.* **14**, 173 (2023).
- [53] S. Montangero, *Introduction to Tensor Network Methods* (Springer, Cham, Switzerland, 2018).
- [54] P. Silvi, F. Tschirsich, M. Gerster, J. Jünemann, D. Jaschke, M. Rizzi, and S. Montangero, The tensor networks anthology: Simulation techniques for many-body quantum lattice systems, *SciPost Phys. Lect. Notes* **8**, 1 (2019).
- [55] D. Rattacaso, D. Jaschke, M. Ballarin, I. Siloi, and S. Montangero, Quantum algorithms for equational reasoning, [arXiv:2508.21122](https://arxiv.org/abs/2508.21122).
- [56] A. Clément, N. Heurtel, S. Mansfield, S. Perdrix, and B. Valiron, A complete equational theory for quantum circuits, in *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)* (ACM, New York, 2023), pp. 1–13.
- [57] A. Clément, N. Delorme, and S. Perdrix, Minimal equational theories for quantum circuits, in *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '24)* (ACM, New York, 2024), pp. 271–2714.
- [58] S. Lloyd, Universal quantum simulators, *Science* **273**, 1073 (1996).
- [59] D. Aharonov and A. Ta-Shma, Adiabatic quantum state generation and statistical zero knowledge, in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, STOC'03* (Association for Computing Machinery, New York, 2003), pp. 20–29.
- [60] F. Baccari, D. Bacilieri, M. Ballarin, F. P. Barone, F. Campaioli, A. G. Catalano, G. Cataldi, A. Coppi, A. Costantini, A. Datta, A. De Girolamo, D. Jaschke, S. B. Kožić, G. Magnifico, C. Mordini, S. Montangero, S. Notarnicola, A. Pagano, L. Pavesic, D. Rattacaso, M. Rigobello, N. Reini, S. Scarlatella, I. Siloi, P. Silvi, M. Tesoro, G. Torre, and D. Wanisch, “Quantum TEA: qtealeaves (2025)”, Zenodo, <https://doi.org/10.5281/zenodo.10498928>.
- [61] M. Ballarin, Quantum computer simulation via tensor networks, Master’s thesis, Università degli Studi di Padova, 2021.
- [62] M. Ballarin, P. Silvi, S. Montangero, and D. Jaschke, Optimal sampling of tensor networks targeting wave function’s fast decaying tails, *Quantum* **9**, 1714 (2025).
- [63] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, *Science* **220**, 671 (1983).
- [64] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2011).
- [65] M. Ballarin, F. P. Barone, A. Coppi, D. Jaschke, S. Montangero, G. Muñoz Menés, D. Rattacaso, and N. Reini, “Quantum TEA: qmatchatea (2025)”, Zenodo, <https://doi.org/10.5281/zenodo.11619265>.
- [66] Y. Zhou, E. M. Stoudenmire, and X. Waintal, What limits the simulation of quantum computers? *Phys. Rev. X* **10**, 041038 (2020).
- [67] E. Farhi, J. Goldstone, and S. Gutmann, Quantum adiabatic evolution algorithms versus simulated annealing, [arXiv:quant-ph/0201031](https://arxiv.org/abs/quant-ph/0201031).
- [68] T. Caneva, T. Calarco, R. Fazio, G. E. Santoro, and S. Montangero, Speeding up critical system dynamics through optimized evolution, *Phys. Rev. A* **84**, 012312 (2011).
- [69] J. J. Wallman and J. Emerson, Noise tailoring for scalable quantum computation via randomized compiling, *Phys. Rev. A* **94**, 052325 (2016).
- [70] M. Mariantoni, H. Wang, T. Yamamoto, M. Neeley, R. C. Bialczak, Y. Chen, M. Lenander, E. Lucero, A. D. O’Connell, D. Sank, M. Weides, J. Wenner, Y. Yin, J. Zhao, A. N. Korotkov, A. N. Cleland, and J. M. Martinis, Implementing the quantum von Neumann architecture with superconducting circuits, *Science* **334**, 61 (2011).

- [71] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum random access memory, *Phys. Rev. Lett.* **100**, 160501 (2008).
- [72] M. Bukov, A. G. R. Day, P. Weinberg, A. Polkovnikov, P. Mehta, and D. Sels, Broken symmetry in a two-qubit quantum control landscape, *Phys. Rev. A* **97**, 052114 (2018).
- [73] M. Larocca, P. M. Poggi, and D. A. Wisniacki, Quantum control landscape for a two-level system near the quantum speed limit, *J. Phys. A: Math. Theor.* **51**, 385305 (2018).
- [74] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Reinforcement learning in different phases of quantum control, *Phys. Rev. X* **8**, 031086 (2018).
- [75] A. G. R. Day, M. Bukov, P. Weinberg, P. Mehta, and D. Sels, Glassy phase of optimal quantum control, *Phys. Rev. Lett.* **122**, 020601 (2019).
- [76] D. Rattacaso, M. Ballarin, D. Jaschke, I. Siloi, and S. Montangero, “VulQano: Quantum-inspired compiler for quantum circuits (2024)”, Zenodo, <https://doi.org/10.5281/zenodo.12571338>.
- [77] D. Rattacaso, M. Ballarin, D. Jaschke, I. Siloi, and S. Montangero, Datasets, figures, and simulation scripts for “Quantum circuit compilation with quantum computers” (v1.0.0) [Data set], Zenodo, <https://doi.org/10.5281/zenodo.12759815>.
- [78] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A* **52**, 3457 (1995).
- [79] M. Khazali and K. Mølmer, Fast multiqubit gates by adiabatic evolution in interacting excited-state manifolds of Rydberg atoms and superconducting circuits, *Phys. Rev. X* **10**, 021054 (2020).
- [80] G. Lami, P. Torta, G. E. Santoro, and M. Collura, Quantum annealing for neural network optimization problems: A new approach via tensor network simulations, *SciPost Phys.* **14**, 117 (2023).
- [81] A. J. Daley, Quantum trajectories and open many-body quantum systems, *Adv. Phys.* **63**, 77 (2014).
- [82] J. Chen, E. M. Stoudenmire, and S. R. White, Quantum Fourier transform has small entanglement, *PRX Quantum* **4**, 040318 (2023).
- [83] D. Jaschke and S. Montangero, Is quantum computing green? An estimate for an energy-efficiency quantum advantage, *Quantum Sci. Technol.* **8**, 025001 (2023).
- [84] S. Heußen, D. F. Locher, and M. Müller, Measurement-free fault-tolerant quantum error correction in near-term devices, *PRX Quantum* **5**, 010333 (2024).
- [85] S. Veroni, M. Müller, and G. Giudice, Optimized measurement-free and fault-tolerant quantum error correction for neutral atoms, *Phys. Rev. Res.* **6**, 043253 (2024).