

Approximate quadratization of high-order Hamiltonians for combinatorial quantum optimization

Sabina Drăgoi^{1,2,*}, Alberto Baiardi², and Daniel J. Egger²

¹*Institute for Theoretical Physics, ETH Zurich, Wolfgang-Pauli-Strasse 27, 8093 Zürich, Switzerland*

²*IBM Quantum, IBM Research–Zurich, Säumerstrasse 4, 8803 Rüschlikon, Switzerland*



(Received 21 November 2025; accepted 6 April 2026; published 12 May 2026)

Combinatorial optimization problems have wide-ranging applications in industry and academia. Quantum computers may help solve them by sampling from appropriately prepared Ansatz quantum circuits. However, current quantum computers are limited by their qubit count, connectivity, and noise. This is particularly restrictive when considering optimization problems beyond the quadratic order. Here, we introduce Ansätze based on an approximate quadratization of high-order Hamiltonians that do not incur a qubit overhead. The price paid is a loss in the quality of the noiseless solution. Crucially, these approximations yield shallower Ansätze that are more robust to noise than the standard Quantum Approximate Optimization Algorithm one. We show this through simulations with variable noise strengths. Furthermore, we also propose a noise-aware Ansatz design method for quadratic optimization problems. This method implements only a portion of the target Hamiltonian by limiting the number of layers of SWAP gates in the Ansatz. We find that for both problem types, under noise, our approximate implementation of the full problem structure can significantly enhance the solution quality. Our work opens a path to enhance the solution quality that approximate quantum optimization achieves on noisy hardware.

DOI: [10.1103/9rc4-vj11](https://doi.org/10.1103/9rc4-vj11)

I. INTRODUCTION

Quantum computing offers an information-processing framework based on the laws of quantum physics. For certain tasks, quantum computing achieves better performance than the best known classical methods. Famously, Shor's algorithm delivers an exponential speedup in factoring over state-of-the-art classical algorithms [1]. However, this remains a theoretical feat since Shor's algorithm requires deep circuits that cannot be faithfully implemented at practically relevant scales on current hardware due to noise. This has motivated the development of algorithms requiring shallower circuits such as Variational Quantum Algorithms (VQAs) [2]. VQAs are typically heuristic in nature and are often designed to find the ground state of a Hamiltonian [3–5]. VQAs optimize a parametrized quantum circuit, called the Ansatz, using a classical optimization procedure to update the parameters.

The Ansatz in a VQA can either be problem-agnostic [6] or retain the structure of the target problem [7]. Problem-agnostic Ansätze are typically designed to minimize the effect of noise by accounting for hardware constraints. However, they come at the cost of limited expressivity compared to

Ansätze that fully encode the problem structure. Designing expressive and shallow Ansätze remains an open question [8].

Here, we focus on VQAs to solve combinatorial optimization problems, which are prevalent across industry applications such as communication [9], finance [10,11], and vehicle routing [12–14]. Many relevant combinatorial optimization problems are hard to solve exactly [15]. In addition, for some of them, polynomial-time approximate algorithms that reach an arbitrarily accurate approximation of the optimal solution are not available [16]. Therefore, many classical algorithms designed to solve such problems are heuristics. Crucially, they work well in practice, thus motivating a similar heuristic approach for quantum computing [17].

A foundational VQA for combinatorial optimization problems is the Quantum Approximate Optimization Algorithm (QAOA) [18,19]. Its Trotterized Ansatz structure alternates p time evolutions of a Hamiltonian representing the target problem and of a problem-independent Hamiltonian. The structure is inspired from the adiabatic theorem in physics, and thus inherits its convergence guarantee in the limit $p \rightarrow \infty$ [18,20,21].

Finite-depth QAOA is a heuristic algorithm, which implies that its performance must be thoroughly investigated and benchmarked on a case-by-case basis [17,22]. In particular, research often investigates QAOA for quadratic binary optimization problems, such as Max-Cut, to benchmark hardware performance [23–26] and validate algorithms [27,28]. However, Max-Cut is easy to solve classically at low graph densities and up to 10 000 nodes [29]. Solving it using QAOA would only yield approximate solutions even in an ideal case [30–33], thus making it a poor candidate to deliver a

*Contact author: sdragoi@ethz.ch

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.

quantum advantage. In contrast, a good candidate problem is hard to solve exactly classically, and preferably also approximately, to motivate the use of quantum hardware. Moreover, such problems should have fewer decision variables than the number of qubits of the quantum hardware, which is currently on the order of 100 qubits for superconducting architectures [34].

High-order binary optimization problems extend quadratic ones by adding high order, such as cubic and quartic, terms. High-order terms can render such problems very hard to solve, even with a low number of decision variables [35], making them interesting to study using quantum heuristics. For instance, the largest exactly solved instance of the Low Autocorrelation Binary Sequence (LABS) problem has only 66 decision variables [36,37]. Importantly, Ref. [38] suggests a scaling advantage for QAOA over state-of-the-art classical solvers such as Gurobi [39] or CPLEX [40] in solving LABS. Therefore, quantum hardware might help solve high-order problems by arriving at, for example, better solutions or by reaching a given solution quality faster than classical hardware. However, the study of quantum algorithms is hindered by hardware noise. Furthermore, the Ansätze from which samples are drawn are typically much deeper for high-order problems than for quadratic ones. This makes quantum approximate optimization more prone to noise when applied to high-order problems than quadratic ones.

In this work, we show how to improve the quality of solutions to high-order optimization problems sampled from noisy quantum hardware by quadratizing the Hamiltonian that generates the Ansatz while optimizing the energy of the nonquadratized Hamiltonian. We first show in Sec. II how the quantum resource requirements of QAOA scale unfavorably when many high-order terms are present by considering two test cases: a fully connected four-local Hamiltonian and the LABS problem. Next, we introduce in Sec. III two approximate quadratization procedures to reduce the QAOA circuit depth without qubit overhead. Motivated by the idea of approximating the circuit Ansatz associated with a given problem, we benchmark how simplifications in the QAOA Ansatz improve experimental performance when executing Max-Cut on hardware in Sec. IV. Finally, in Sec. V, we argue for the need of a more mathematically motivated quadratization and suggest some ideas in this direction.

II. EXECUTING QAOA ON DIGITAL QUANTUM COMPUTERS

We consider unconstrained combinatorial optimization problems defined over n binary decision variables $x \in \{0, 1\}^n$. Here, the aim is typically to find the values of the decision variables that minimize an objective function $f(x)$, i.e.,

$$\min_{x \in \{0,1\}^n} f(x). \quad (1)$$

A quadratic unconstrained binary optimization (QUBO) problem is defined by a quadratic cost function

$$f(x) = x^T Q x, \quad (2)$$

where Q is an $n \times n$ real matrix. In contrast, a high-order unconstrained binary optimization (HUBO) problem takes the

form

$$f(x) = x^T Q x + \sum_{i < j < k} d_{ijk} x_i x_j x_k + \dots, \quad (3)$$

where d_{ijk} are real coefficients of high-order terms and $i, j, k \in [1, n]$ are variable indices.

Quantum computers can tackle such problems by mapping $f(x)$ to an Ising Hamiltonian H_C whose ground state minimizes $f(x)$. This is done through the change of variables $x_i = \frac{1-z_i}{2}$ and promoting z_i to Pauli Z operators Z_i [41]. For example, the cost Hamiltonian of a QUBO is

$$H_{\text{QUBO}} = \sum_{i < j} w_{ij} Z_i Z_j + \sum_j w_j Z_j \quad (4)$$

up to an irrelevant constant.

Then, we use QAOA to find the ground state of H_C by sampling from an Ansatz

$$|\psi(\beta, \gamma)\rangle = \prod_{q=1}^p e^{-i\beta_q H_M} e^{-i\gamma_q H_C} |+\rangle^{\otimes n}, \quad (5)$$

with optimized variational parameters (β, γ) .

In QAOA, the mixing Hamiltonian H_M is chosen such that its ground state is the initial state. This ensures that an adiabatic evolution, obtained in the limit of $p \rightarrow \infty$, transforms the initial ground state of H_M to the ground state of H_C . Here, we consider a problem-independent mixer operator $H_M = -\sum_{i=0}^{n-1} X_i$, although other mixing Hamiltonians have been proposed [28,42–46]. The main challenge associated with executing QAOA on quantum processors is implementing the time evolution of the cost Hamiltonian. In particular, the more terms H_C contains, the denser and deeper the resulting circuits become, which results in noisier samples. The maximum number of gates in the time-evolution circuit of a fully connected QUBO H_2^{full} scales as $O(n^2)$, with an $O(n)$ two-qubit gate depth [47]; see the purple lines in Fig. 1. When transpiled to a line of qubits, the complexity exponent remains unchanged, but the scaling prefactor increases by 1.5 due to the additional SWAP gates required to reach all-to-all connectivity; see details in Appendix A.

Here, transpilation is the process of transforming a high-level quantum circuit into a logically equivalent one expressed in the native gate set and connectivity of a target quantum backend, while applying optimizations to reduce gate depth and count. Indeed, in many quantum computing architectures only a subset of the qubits are directly coupled. For example, a line of physically connected qubits often requires additional SWAP gates to bring in contact the desired pair of qubits. In contrast, an all-to-all qubit connectivity can directly implement any native two-qubit gate on any pair of qubits. In all of our calculations, we assumed either an all-to-all qubit connectivity or a linear chain of qubits on which we apply a SWAP strategy.

A SWAP strategy is a predetermined network of SWAP gates tailored to the qubit connectivity [42,48–50]. The SWAP strategy we employ for a line of qubits consists of alternating layers of even and odd SWAP layers. A SWAP layer is a

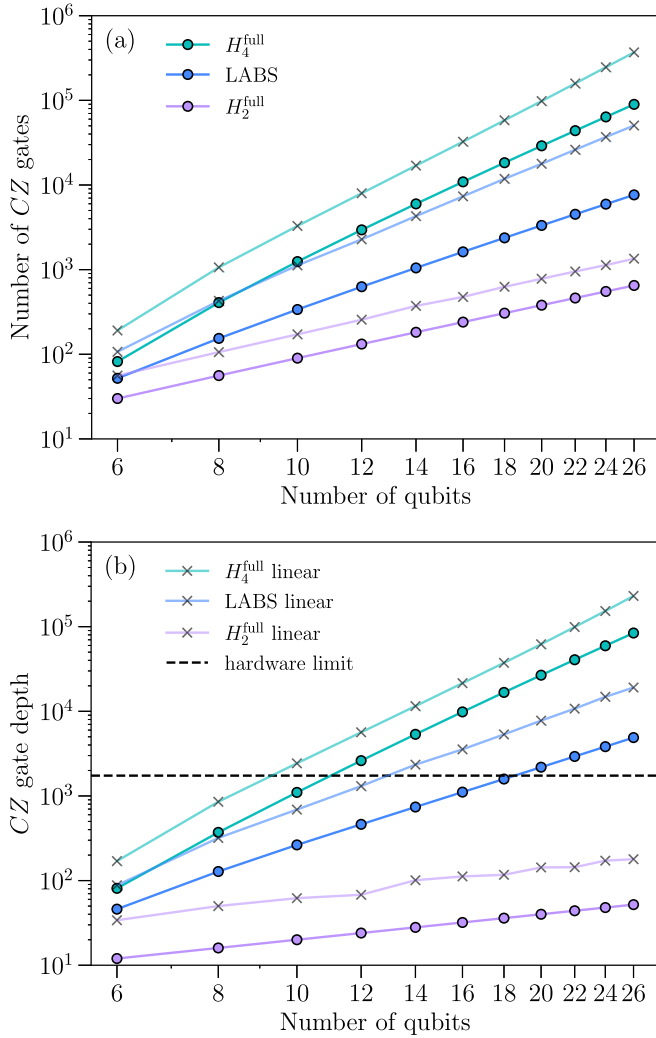


FIG. 1. Circuit complexity of $e^{-i\gamma H_C}$ for H_4^{full} (teal) and LABS (blue) compared to a fully connected QUBO (purple). Panel (a) shows the number of CZ gates as a function of problem size. Panel (b) shows the CZ gate depth of the transpiled circuits. The continuous lines connect raw data points. The horizontal dashed line corresponds to the number of CZ gates that fit within the median qubit T_1 time on *ibm_fez*.

depth-one circuit that applies SWAP gates to pairs of neighboring qubits, and it is called even if the first pair is (0, 1) and odd if it is (1, 2) [see Fig. 2(d)]. Full connectivity is reached after $n - 2$ alternating SWAP layers.

We then decompose our circuits into hardware-native single-qubit and controlled-phase (CZ) gates. However, the complexity of a circuit depends on the transpilation and the hardware's topology and native gates. For example, a detailed discussion of grid-based hardware and heavy-hex hardware for quadratic Hamiltonians is in Ref. [48].

Furthermore, the locality of the gates in $\exp(-i\gamma H_C)$ significantly impacts the circuit depth. A gate $\exp(-i\gamma Z_i Z_j)$ is implemented by a single R_{ZZ} rotation if the qubits i and j are connected on the hardware. This R_{ZZ} rotation then decomposes into two hardware-native CZ gates. Otherwise, additional SWAP gates are needed to bring qubits i and j in contact. Minimizing the SWAP overhead is desirable

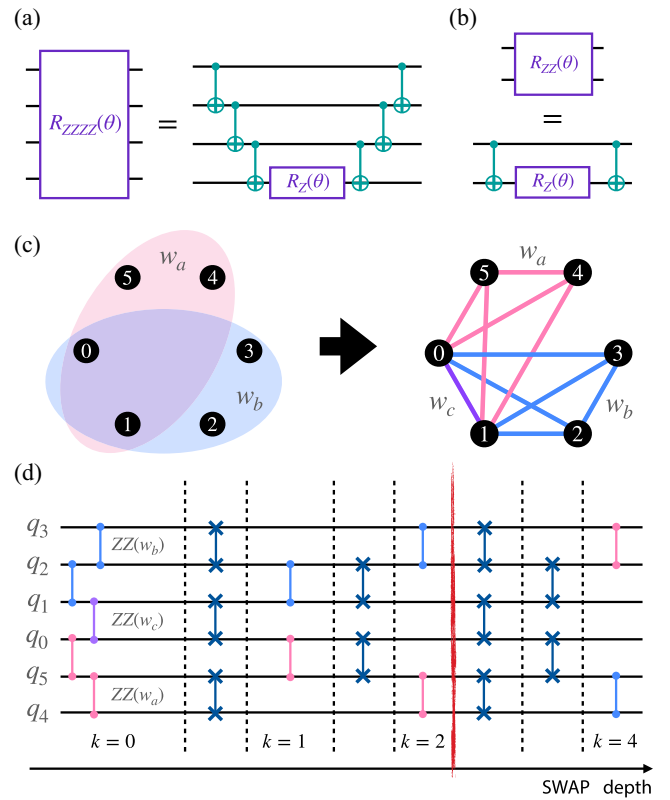


FIG. 2. Construction of shallow Ansatz. Panels (a) and (b) show how a four-local gate $\exp(-i\gamma ZZZZ)$ and a two-local gate $\exp(-i\gamma ZZ)$, respectively, decompose into R_Z and CX gates. (c) The hypergraph clique expansion generates a two-dimensional (2D) graph that is easier to implement on quantum hardware. The graph is generated such that each pair of nodes in the original hypergraph has a corresponding edge. Here, the edge (0, 1) has weight $w_c = (w_a + w_b)/2$, whereas the other edges preserve the weights of the single hyperedge they correspondingly belong to. (d) Circuit implementation of the resulting graph with layers of SWAP gates. To limit the effect of hardware noise, we skip implementing the last two SWAP layers, at the cost of losing some correlation with the original problem.

since each SWAP gate is built from three noisy hardware-native entangling CZ gates. Implementing $\exp(-i\gamma H_C)$ is even more challenging for HUBOs, whose terms require more stringent connectivity constraints; i.e., implementing $\exp(-i\gamma Z_i Z_j Z_k Z_l)$ requires bringing close together qubits (i, j, k, l) . Therefore, HUBOs are more complicated to implement than QUBOs as they require implementing possibly many nonlocal gates. We now investigate the gate count and depth for two examples of HUBOs, i.e., the LABS problem and a fully connected fourth-order Hamiltonian H_4^{full} .

The LABS problem has applications in signal processing [51] and statistical physics [52]. It aims to minimize the side-lobe energy

$$E_{\text{sidelobe}}(z) = \sum_{k=1}^{n-1} C_k(z)^2 \quad (6)$$

TABLE I. Complexity scaling of selected problems. The trends in the first and third columns are shown in Fig. 1. The numerical values in the remaining columns correspond to problem instances with 16 variables.

Problem	Fourth-order terms		Second-order terms		All-to-all connectivity		Linear connectivity	
	Scaling	$n = 16$	Scaling	$n = 16$	No. of CZ gates	CZ gate depth	No. of CZ gates	CZ gate depth
H_2^{full}	–	–	$O(n^2)$	120	240	32	507	103
LABS	$O(n^3)$	252	$O(n^2)$	56	1218	1071	4685	2978
H_4^{full}	$O(n^4)$	1820	–	–	4732	4622	17 813	13 920

of a length n spin sequence $z \in \{-1, 1\}^n$. Here, the autocorrelation functions are

$$C_k(z) = \sum_{i=1}^{n-k} z_i z_{i+k}. \quad (7)$$

Expanding the sums in the expression of the sidelobe energy yields the Hamiltonian [38]

$$H_{\text{LABS}} = 2 \sum_{i=1}^{n-3} Z_i \sum_{t=1}^{\lfloor \frac{n-i-1}{2} \rfloor} \sum_{k=i+t}^{n-i-t} Z_{i+t} Z_{i+k} Z_{i+k+t} + \sum_{i=1}^{n-2} Z_i \sum_{k=1}^{\lfloor \frac{n-i}{2} \rfloor} Z_{i+2k}, \quad (8)$$

whose energies with respect to spin sequences $E_{\text{LABS}}(z) = \langle z | H_{\text{LABS}} | z \rangle$ are related to the corresponding sidelobe energies through

$$E_{\text{sidelobe}}(z) = 2E_{\text{LABS}}(z) + \frac{n(n-1)}{2}. \quad (9)$$

There is therefore a single problem instance per size n .

Solving LABS with QAOA requires implementing the time-evolution operator associated with $O(n^3)$ four-local terms and $O(n^2)$ two-local terms. The dominant $O(n^3)$ scaling is shown by the thicker blue lines in Fig. 1. For example, a LABS instance with 16 decision variables has 252 four-local terms and 56 two-local terms. This requires $252 \times 6 + 56 \times 2 = 1624$ two-qubit gates assuming an all-to-all connectivity, where the constant factors 6 and 2 are due to the two-qubit gate decomposition [see Figs. 2(a) and 2(b)]. After transpilation using Qiskit with the highest optimization level 3, this number reduces to 1218 through cancellation of neighboring two-qubit gates [53]. However, when compiled to linearly connected qubits, the two-qubit gate count increases to 4685 with a two-qubit gate depth of 2978; see the thin blue line in Fig. 1. Implementing such deep circuits on current noisy hardware is a formidable challenge. For instance, on *ibm_fez*, a 156 superconducting qubit device, the median T_1 time of 146 μs fits approximately only 1738 layers of two-qubit gates with a median gate time of 84 ns.

The hardest fourth-order problem to implement is a fully connected four-local Hamiltonian H_4^{full} , which has a similar structure as the Sachdev-Ye-Kitaev model of Majorana fermions [54–56], but which is expressed in terms of Pauli operators instead of fermionic ones. Concretely, we define

$$H_4^{\text{full}} = \sum_{i < j < k < l} \binom{n}{4} J_{ijkl} Z_i Z_j Z_k Z_l, \quad (10)$$

J_{ijkl} being real coefficients. The resulting circuits have $O(n^4)$ two-qubit gates and $O(n^3)$ two-qubit gate depth; see teal line in Fig. 1. Implementing QAOA for H_4^{full} is thus more challenging than for LABS, and considerably more so than for a fully connected QUBO. This is true not only in the asymptotic limit, but also for instances with as few as 16 variables, which require 4732 CZ gates and a depth of 4622 assuming all-to-all connectivity, and more than 3 times as much for a linear qubit topology (see Table I).

Finally, another practical limitation of these deep circuits is their low sampling rates on quantum hardware. Indeed, the two-qubit gate depth D_{CZ} , shown in Fig. 1, lower bounds the sampling rate of depth-one QAOA by $(D_{\text{CZ}} \bar{T})^{-1}$, where $\bar{T}$ is the CZ-gate duration. This makes Ansatz circuits with a depth scaling linearly in problem size also desirable from a speed perspective.

III. APPROXIMATE QUADRATIZATION

Compiling quartic HUBO problems to hardware-native instructions results in very deep circuits, while QUBOs are optimally compiled in linear depth with SWAP networks [42,49]. This motivates us to generate the Ansatz based on QUBOs defined in terms of the target HUBO. For example, an exact quadratization that exactly conserves the full structure of the HUBO replaces each four-local term with one constant, four linear, and six quadratic terms at the expense of adding two ancilla qubits per four-local term [57]. This implies that implementing four-local Hamiltonians with $O(n^3)$ terms, such as LABS, or $O(n^4)$ terms, such as H_4^{full} , requires circuits with a qubit overhead scaling as $O(n^3)$ or $O(n^4)$, respectively. This is impractical due to the limited qubit count of the current hardware. We therefore explore *approximate* quadratization methods to project a generic HUBO onto a QUBO without adding decision variables.

A. Methods

We study two methods to quadratize the QAOA Ansatz of HUBOs. The first is based on a hypergraph clique expansion, and the second is based on a fully connected quadratic Hamiltonian whose gate weights are treated as additional variational parameters. The price paid to keep the qubit count constant is an approximation of the original problem. Importantly, the objective remains to find samples that minimize the energy of a HUBO Hamiltonian H_C . However, we use the quadratic Hamiltonian H_2 as the generator of the QAOA Ansatz because it leads to a shallower, and thus less noisy, circuit. Ideally, the energy of both H_2 and H_C are minimized by the same bitstrings.

In the hypergraph clique expansion [58], we treat a high-order Hamiltonian $H_C = \sum_{i=0}^{K-1} \omega_i P_i$ as a hypergraph $\mathcal{G} = (\mathcal{E}, V)$, where each vertex in V corresponds to a binary decision variable in the original problem and $\mathcal{E}$ is the set of associated hyperedges. We associate each hyperedge $e = (i_1, i_2, \dots, i_k) \in \mathcal{E}$ with a Pauli term $P_i = Z_{i_1} Z_{i_2} \dots Z_{i_k}$. Here, ω_i is the coefficient of the Pauli P_i . To quadratize H_C , we create a weighted graph $G = (E, V)$, resulting in a quadratic Hamiltonian $H_2 = \sum_{i,j} \omega_{ij} Z_i Z_j$. We add an edge to E for each pair of nodes (i, j) within a hyperedge e , as shown in Fig. 2(c). Therefore, each hyperedge e with k nodes in $\mathcal{E}$ generates a k clique in G with edge set $E(e)$. Since a given pair of nodes (i, j) might belong to multiple hyperedges of $\mathcal{E}$, the weight w_{ij}^{opt} we assign to the edge $(i, j) \in E$ minimizes the l_2 norm, i.e., the sum of squared differences between it and the weights $w(e)$ of all hyperedges e that contain (i, j) . The l_2 norm is chosen according to the quadratization given in the literature [58]. In particular, we choose ω_{ij} to minimize

$$f_2(\omega) = \sum_{e \in \mathcal{E}} \sum_{(i,j) \in E(e)} \alpha_e [\omega_{ij} - \omega_e]^2. \quad (11)$$

Here, ω_e is the weight of hyperedge e . The weight $\alpha_e = \omega_e / \sum_{e \in \mathcal{E}} \omega_e$ prevents low-weight edges from having a disproportionate impact on the quadratization.

Effectively, each high-order Pauli term is viewed as a hyperedge connecting the qubits it acts on. The hypergraph clique expansion then replaces each hyperedge with a clique to obtain the correlators of a quadratic Hamiltonian. The weights of these correlators are then chosen to best approximate all contributions of its parent hyperedges in a least-squares sense. Deriving a closed-form solution for the resulting optimal weights ω_{ij}^{opt} can be beneficial for reducing the time this preprocessing step takes (see Appendix B).

We use the resulting quadratic Hamiltonian

$$H_2 = \sum_{(i,j) \in E} \omega_{ij}^{\text{opt}} Z_i Z_j, \quad (12)$$

where

$$\omega_{ij}^{\text{opt}} := \arg \min_{\omega_{ij}} f_2(\omega), \quad (13)$$

to generate the Ansatz circuit of QAOA, while still evaluating the objective function with respect to the original quartic Hamiltonian. Concretely, we minimize the energy $\langle \psi_2(\boldsymbol{\beta}, \boldsymbol{\gamma}) | H_C | \psi_2(\boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle$, where the Ansatz state is

$$|\psi_2(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = \prod_{q=1}^p e^{-i\beta_q H_M} e^{-i\gamma_q H_2} |+\rangle^{\otimes n}. \quad (14)$$

The second quadratization method consists of generating an Ansatz circuit with a quadratic Hamiltonian whose parameters $\boldsymbol{\theta}$ are simultaneously optimized with $(\boldsymbol{\beta}, \boldsymbol{\gamma})$ to minimize the energy of H_C . In particular, we perform the joint minimization

$$\min_{\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\gamma}} \langle \psi'_2(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\gamma}) | H_C | \psi'_2(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle, \quad (15)$$

where the Ansatz

$$|\psi'_2(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = \prod_{q=1}^p e^{-i\beta_q H_M} e^{-i\gamma_q H'_2(\boldsymbol{\theta})} |+\rangle^{\otimes n} \quad (16)$$

is generated by the Hamiltonian

$$H'_2(\boldsymbol{\theta}) = \sum_{i>j} \theta_{ij} Z_i Z_j + \sum_{i=0}^{n-1} \theta_i Z_i, \quad (17)$$

where $\boldsymbol{\theta} = \{\theta_{ij}, \theta_i\}$.

This Ansatz differs from the multi-angle Ansatz (ma-QAOA) [59]. In ma-QAOA, the Hamiltonian generating the Ansatz is built from the exact same terms as the cost operator H_C , but each Pauli term k is associated with an independent QAOA angle $\gamma_{q,k}$ at each QAOA layer. Therefore, our $H'_2(\boldsymbol{\theta})$ defines a quadratization of the original quartic Hamiltonian H_C .

These quadratizations introduce an inconsistency between the cost operator and the operator that creates the Ansatz. Crucially, a purely adiabatic evolution from the ground state of the mixer Hamiltonian H_M to the ground state of H_2 will fail to solve the HUBO problem when

$$\arg \min_{z \in \{-1, 1\}^n} \langle z | H_C | z \rangle \cap \arg \min_{z \in \{-1, 1\}^n} \langle z | H_2 | z \rangle = \emptyset. \quad (18)$$

Therefore, the convergence guarantee of a typical QAOA in the limit $p \rightarrow \infty$ is lost. Instead, the approach is inherently heuristic as it is based on an Ansatz circuit that retains only some of the structure of H_C , but which considerably reduces the circuit depth. The exact depth reduction depends on the density of the original problem. In the case of $H'_2(\boldsymbol{\theta})$, the convergence as $p \rightarrow \infty$ may still persist as long as the ground states of the quadratized Hamiltonian and H_C have a sufficient overlap. The largest gate count reduction occurs for dense Hamiltonians, such as H_4^{full} , for which the reduction scales as $O(n^2)$ (see Fig. 1). For LABS, which has $O(n^3)$ terms, the reduction factor is $O(n)$.

B. Expected performance of the quadratizations

We now benchmark the accuracy of the quadratizations by investigating depth-one and depth-two QAOA. We perform this assessment on the LABS Hamiltonians with 12 decision variables. The depth-one QAOA angles (β_1^*, γ_1^*) are trained with a coarse 2D grid scan with 15 points in the interval $[0, \pi/2]$, followed by a Scipy optimization with COBYLA to refine the QAOA angles of the grid search. The depth-two angles are found with COBYLA from the transition state $(\beta_1^*, 0, \gamma_1^*, 0)$ as initial point [60]. The parameters $\boldsymbol{\theta}$ in $H'_2(\boldsymbol{\theta})$ and the depth- p QAOA angles are trained simultaneously with Scipy minimize and COBYLA to minimize the energy of H_C (see Fig. 3). The parameters in the clique expansion are optimized with Scipy and COBYLA independently from the QAOA parameters and converge to a low value of $f_2(\omega)$ (see Fig. 3).

The optimized quantum circuits are sampled with the state-vector simulator of Qiskit. Resulting bitstring samples x are converted into spinlike strings z according to $z_i = 1 - 2x_i$. The energy of the LABS Hamiltonian, E_{LABS} , is computed by first evaluating $E_{\text{LABS}}(z)$ for each resulting sample z , and then

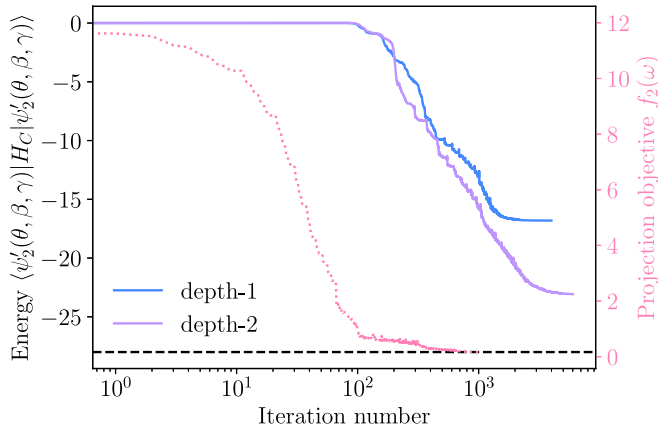


FIG. 3. Convergence of the quadratizations of the LABS Hamiltonian on 12 variables as function of COBYLA iterations. The clique expansion, i.e., the minimization of Eq. (11), is shown as the dotted pink curve on the right y axis. The solid blue and violet curves show the minimization of Eq. (15) for QAOA depths one and two, respectively. The horizontal dashed black line shows the -28 ground-state energy, defined in Eq. (6).

taking their weighted average, where weights correspond to sampling probabilities. We statistically analyze the quality of the sampled distribution through the resulting cumulative distribution function $\text{CDF}(E) = \text{Prob}(E_{\text{LABS}}(z) \leq E) \in [0, 1]$.

We compare the clique expansion, the $H'_2(\theta)$ projection, standard QAOA, and uniform sampling. The performance of the clique expansion is very close to uniform sampling (see Fig. 4). Both standard depth-one and -two QAOA outperform random sampling and the clique expansion; see dashed curves in Fig. 4. The depth-two standard QAOA, for example, samples the optimal -28 energy bitstring with a probability of 0.8%. The projection $H'_2(\theta)$ does not find the optimal bitstring, but produces good, yet suboptimal, solution with high probability. For example, the depth-two QAOA with the $H'_2(\theta)$ projection produces samples with a 97.1% approximation

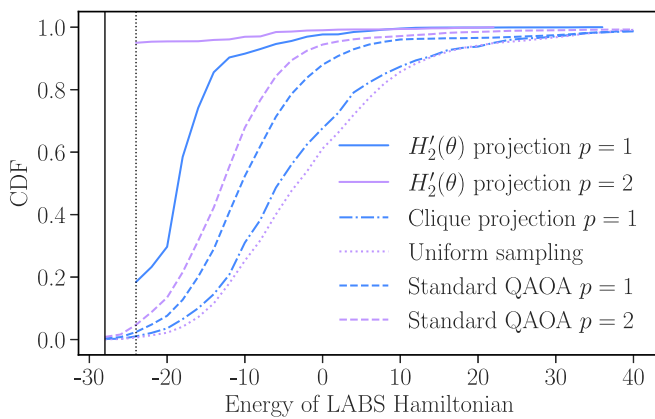


FIG. 4. Cumulative distribution function of the LABS Hamiltonian energy over 2^{14} samples obtained from different quantum circuits. The vertical solid line shows the minimum energy found by direct diagonalization. The dotted vertical line shows the best solution sampled from the $H'_2(\theta)$ quadratization. The full energy range of the 12 variable LABS Hamiltonian is $[-28, 110]$.

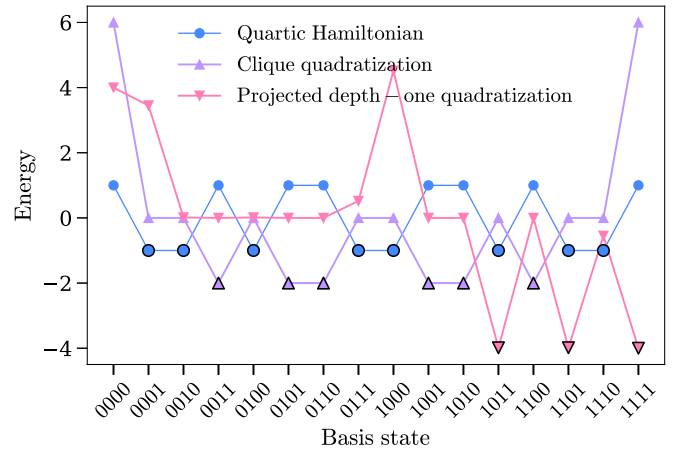


FIG. 5. Example of the energy distribution over all binary basis states of the quartic Hamiltonian $Z_3Z_2Z_1Z_0$ (blue), and its quadratized versions computed through the clique expansion defined in Eq. (12) (purple), and the simultaneous-QAOA-parameter-optimization quadratization defined in Eq. (17), for a depth-one QAOA (pink).

ratio, defined by

$$r(z) = \frac{E_{\text{LABS}, \max} - E_{\text{LABS}}(z)}{E_{\text{LABS}, \max} - E_{\text{LABS}, \min}} \in [0, 1], \quad (19)$$

with a probability of 95.7%. Here, 100% represents optimality. Furthermore, $E_{\text{LABS}, \min}$ and $E_{\text{LABS}, \max}$ are the lowest and highest LABS energies, respectively, which we compute classically using the Gurobi solver [39]. In contrast, $E_{\text{LABS}}(z)$ is obtained from quantum samples.

The quadratization yields an interesting trade-off since it only approximately solves the 12-variable LABS problem, but with a significantly shorter circuit depth. Indeed, the two-qubit gate depth and count of the Ansatz circuits based on the quadratization are 68 and 374 when transpiled with a SWAP network on a line of qubits [48]. In contrast, transpiling the standard QAOA Ansatz with the Qiskit transpiler at optimization level three on a fully connected network of qubits creates circuits with a two-qubit gate count and depth of 600 and 417, respectively. The increased gate count will likely worsen when the circuit is transpiled on a line of qubits. Irrespective of the qubit topology, this example shows the savings in quantum computational resources that the quadratization yields.

The clique-based quadratization fails to produce good samples because it does not preserve the ground state of the problem. For example, the clique quadratization of the quartic term $Z_3Z_2Z_1Z_0$ produces the fully connected and equally weighted Hamiltonian $\sum_{i < j}^3 Z_i Z_j$. The ground state of $Z_3Z_2Z_1Z_0$ is a superposition of basis states that are represented by bitstrings x of Hamming weight 1 or 3; see states with blue markers and black outlines in Fig. 5.

In contrast, the bitstring representations of ground states of the fully connected quadratic Hamiltonian have Hamming weight 2; see states with purple markers and black outlines in Fig. 5. As a result, the clique-based quadratization maps the original ground space onto an energetically unrelated subspace, leading to poor performance.

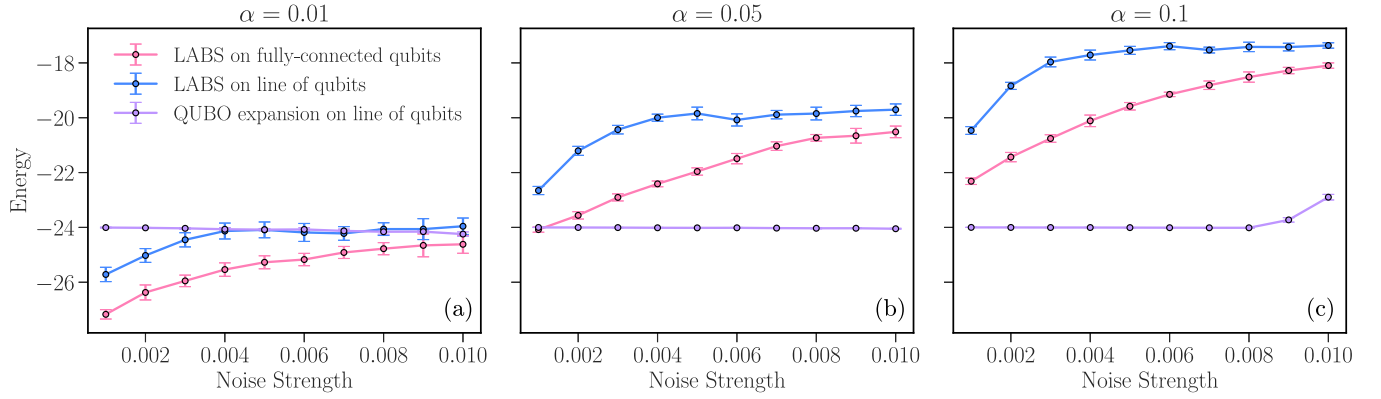


FIG. 6. Evolution of expected performance of depth-two QAOA for LABS vs strength of depolarizing noise for a system of 12 qubits, using the standard Ansatz transpiled on hardware with full connectivity (pink) and with line connectivity (blue) compared to using the quadratized Ansatz transpiled on a line of qubits (purple). Panels (a), (b), and (c) show the energy averaged over the best fraction α of all samples, for $\alpha = 0.01$, $\alpha = 0.05$, and $\alpha = 0.1$, respectively. The circuit corresponding to the pink line has 600 two-qubit gates, the blue 2094, and the purple 374. Each data point and error bar represent the average and standard deviation, respectively, of ten noisy simulations. In each simulation, we draw 10 000 samples.

By comparison, the quadratic Hamiltonian obtained from our second quadratization method has ground states that retain nonzero overlap with the ground-state subspace of the original quartic Hamiltonian; see states with pink markers and black outlines in Fig. 5. This helps explain the success of the quadratization in Eq. (15), which constructs an Ansatz that minimizes the energy of the cost Hamiltonian.

C. Noisy simulations

We now study the performance of the $H'_2(\theta)$ quadratization and standard depth-two QAOA by setting up a noisy simulation in which we apply a depolarizing channel

$$\mathcal{E}_{\text{dep}}(\rho) = (1 - \lambda) \rho + \frac{\lambda}{15} \sum_{\substack{P \in \{I, X, Y, Z\}^{\otimes 2} \\ P \neq I \otimes I}} P \rho P, \quad (20)$$

with strength λ after each two-qubit gate. In Fig. 6, we show how the energy averaged over the best fraction $\alpha \in \{0.01, 0.05, 0.1\}$ of all samples, as measured by the LABS energy, evolves with λ . By varying α , we see how the noise impacts the tail of the distribution of samples produced by the quantum circuit.

The quadratized Ansatz is less sensitive to noise, making it a viable option for noisy quantum computers. Generally, as λ increases, the average energies of the best α fraction of samples increase, corresponding to a worse performance, as expected. Importantly, at $\alpha = 0.01$, the Ansatz based on the QUBO projection (purple) produces samples that are statistically insensitive to the noise. In contrast, the energy of the QAOA Ansätze generated by H_{LABS} on a line of qubits (blue) and all-to-all connected qubits (pink) shows a strong noise dependence [see Fig. 6(a)]. At $\alpha = 0.1$, the QUBO Ansatz is 25% and 36% less sensitive [61] than these two Ansätze, respectively [see Fig. 6(c)]. These results indicate that the samples produced from a quadratized Ansatz are less sensitive to noise at the scale and noise strengths considered here.

The effectiveness of quadratizing the Ansatz depends on the noise strength and set α values. At $\alpha = 0.01$ and noise strength $\lambda = 0.001$, the standard QAOA Ansatz outperforms

the quadratization. However, this advantage vanishes at $\lambda = 0.004$ on a line connectivity. When considering the 5% best samples, the quadratization is better than the standard QAOA for both all-to-all and line connectivity, regardless of noise strength [see Fig. 6(b)].

These data show the benefits of quadratizing the Ansatz when working with hardware with small-to-intermediate noise, relative to the size of the optimization problem. Furthermore, they reinforce the trade-off discussed in Fig. 4, namely, that the quadratized Ansatz can increase the frequency of sampling low-energy, yet suboptimal, states.

IV. HARDWARE-NATIVE CIRCUITS FOR QUBOS

We now study an approximate hardware-friendly compilation method of Ansatz circuits for QUBOs. We use the SWAP strategy-based approach to route an arbitrary QAOA circuit introduced in Sec. II. In particular, we use a line of qubits to benefit from two-qubit gate cancellations, even though the IBM quantum hardware connectivity has a heavy-hex structure [48]. To fully implement $e^{-i\gamma H_C}$, we use k_{max} alternating even and odd SWAP layers. The SWAP network thus allows us to derive an approximate Ansatz by including only $k < k_{\text{max}}$ layers when implementing $e^{-i\gamma H_C}$. This allows us to implement only the time evolution of a subset $H_C(k)$ of the terms in H_C . The truncated $H_C(k)$ is based on the subgraph $G_k = (E_k, V) \subseteq G = (E, V)$. Details on the transpilation are in Appendix C.

Once the Ansatz is created, we optimize the QAOA parameters by minimizing the energy

$$\min_{\beta_k, \gamma_k} \langle \psi_k(\beta_k, \gamma_k) | H_C | \psi_k(\beta_k, \gamma_k) \rangle \quad (21)$$

where

$$|\psi_k(\beta_k, \gamma_k)\rangle = \prod_{j=1}^p e^{-i\beta_{j,k} H_M} e^{-i\gamma_{j,k} H_C(k)} |+\rangle^{\otimes n}. \quad (22)$$

Our approach is similar to the time-block Ansatz design presented in Ref. [42]. The main distinction is that a

time-block Ansatz assigns different, independently optimized parameters to the edges of $H_C(k)$. Concretely, they replace the values $\gamma_{i,j}$ with corresponding vectors $\{\gamma_{i,j}^e, \forall e \in E\}$.

We benchmark the truncated Ansatz on Max-Cut problems [15] for all $k \in [0, k_{\max}]$. We thus aim to split the set V of n nodes of a graph $G = (E, V)$ in two parts such that the sum of the edges connecting the two subsets is maximized. Formally, a decision variable x_i is assigned to each node $i \in V$ to indicate which side of the cut the node is on. Solving Max-Cut is then equivalent to finding the binary string x that maximizes

$$f(x) = \sum_{(i,j) \in E} (x_i(1 - x_j) + x_j(1 - x_i)). \quad (23)$$

Mapping $f(x)$ to a Hamiltonian [41] results in the cost operator

$$H_{\text{Max-Cut}} = \frac{1}{4} \sum_{(i,j) \in E} Z_i Z_j, \quad (24)$$

with a minimum energy eigenstate that maps to the maximum cut. We measure the quality of a candidate solution x with the approximation ratio $r(x) = f(x)/\max_{x'} f(x') \in [0, 1]$. Optimal samples x^* thus have $r(x^*) = 1$. When s samples are drawn, we compute the average approximation ratio $r = s^{-1} \sum_{i=1}^s r(x_i)$. On average, uniformly sampled candidate solutions have $r = 0.5$. The best known polynomial-time randomized rounding algorithm, the Goemans-Williamson algorithm [63], produces samples with $r \simeq 0.878$, on average.

We generate three unweighted random three-regular (RR3) graphs with 40 nodes. The positioning of the decision variables onto the physical qubits is optimized following Ref. [49] to minimize the number of SWAP layers (see Appendix C). All three graphs thus require only $k_{\max} = 9$ SWAP layers.

Next, we generate multiple Ansatz circuits with $k \in [0, k_{\max}]$ SWAP layers and depth $p \in \{1, 2, 3\}$. We train the parameters in each Ansatz using a matrix product state (MPS)-based simulator to evaluate the energy with a bond dimension of 20 (see Appendix D). We observe a monotonic increase with k of the approximation ratio evaluated with the MPS [see Fig. 7(a)]. This shows that the QAOA performance improves as the Ansatz retains more of the problem structure. The circuits are then executed on *ibm_fez* from which we draw 50 000 samples. The samples at $k = 9$ are better than those measured by Sack and Egger on *ibm_brisbane* [62]; see the black dashed line in Fig. 7(b). This is expected since the median two-qubit gate length and T_1 time are 660 ns and 240 μ s, respectively, on *ibm_brisbane* and 68 ns and 146 μ s, respectively, on *ibm_fez*.

Crucially, these data show significant gains in solution quality when the truncated Ansatz is applied. In the presence of hardware noise, there is an optimal point (k, p) corresponding to the maximum depth beyond which increasing either k or p reduces r [see Fig. 7(b)]. On *ibm_fez*, we find this optimal point at depth-three with two SWAP layers. Once noise begins to reduce the approximation ratio, the higher-depth QAOA degrades faster than the lower-depth QAOA.

Increasing k and p improves the quality of noiseless samples. However, this also increases the differences between the ideal and the hardware results because the circuit depth, and therefore also noise, increase; compare Figs. 7(a) and 7(b). We can recover a noise-free approximation ratio by averaging over the best α fraction of the samples x as ranked by $f(x)$

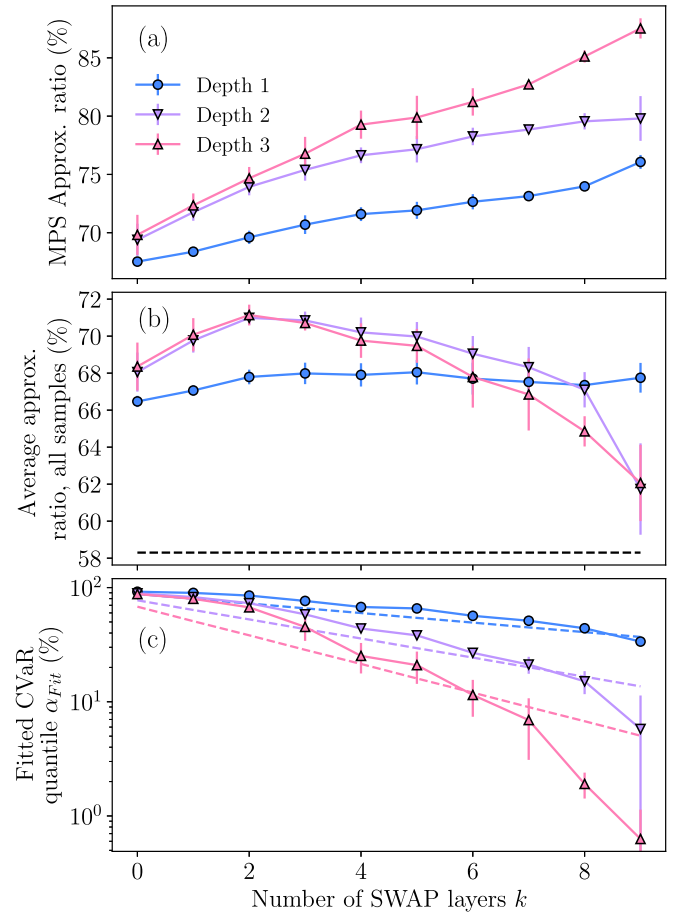


FIG. 7. Performance of QAOA on the Max-Cut problem as a function of depth and the number of SWAP layers. Each data point is the average of three unweighted RR3 graphs that require nine SWAP layers. The error bars show the standard deviation over the different graph instances. (a) Approximation ratio of QAOA computed with an MPS evaluation of the energy. (b) Measured approximation ratios from 50 000 samples from *ibm_fez*. The horizontal dashed line is the 58.3% average approximation ratio reported in Ref. [62]. (c) Value of the CVaR quantile α required to make the approximation ratio of the *ibm_fez* samples in panel (b) match those of the MPS energy evaluation in panel (a). The dashed lines show a simple theoretical expectation α_{th} based on the two-qubit gate fidelity.

[64]. This average over the tail of the sampled distribution is known as the conditional value at risk (CVaR) [65]. We pick α so that the CVaR approximation ratio matches the noiseless one obtained with the MPS simulator in Fig. 7(a).

The optimization of the α parameter is carried out with COBYLA and results in a root-mean-squared error of 10^{-6} between the 90 MPS approximation ratios and the 90 CVaR ones (three graphs, three depths, and ten values of k). As expected, as k and p increase, we discard more shots to recover a noiseless expectation value; see solid lines in Fig. 7(c). This noise strength is then compared to a theoretical amount of noise based on error rates reported by *ibm_fez*. To this end, we compute $\alpha_{\text{th}} = 1/\sqrt{\gamma}$, where γ measures the noise strength of the circuit. We make the simple assumption that

$$\gamma = \gamma_0^{p(2+3\lceil \frac{k}{2} \rceil)} \gamma_1^{p(2+3\lceil \frac{k}{2} \rceil-3)}. \quad (25)$$

Here, γ_0 and γ_1 represent the layer fidelity [66] of the even and odd layer of CZ gates, respectively. The factors γ_i^{2p} account for the first layers of R_{ZZ} gates, which transpile to four hardware-native layers of CZ gates per QAOA layer p on *ibm_fez*. The terms $\gamma_0^{3p\lceil \frac{n}{2} \rceil}$ and $\gamma_1^{3p\lceil \frac{n}{2} \rceil - 3p}$ account for the even and odd SWAP layers, respectively, which transpile into three layers of CZ gates each. We approximately express γ_i , where $i \in \{0, 1\}$, using the errors $\epsilon_{j,j+1}$ of the native entangling gates between the pairs of qubits j and $j+1$ reported by the quantum processor. For the even and odd layers of two-qubit gates, we have

$$\gamma_i = \prod_{j=0}^{\lfloor \frac{n-1}{2} \rfloor - 1} (1 + \epsilon_{2j+i, 2j+i+1}), \quad (26)$$

where we assume $n \geq 3$. This simple $1/\sqrt{\gamma}$ estimation of α already produces excellent qualitative agreement with the fitted α ; the solid lines in Fig. 7(c) show the same trend as the dashed ones. At a low number of SWAP layers, we overestimate the sampling overhead since the transpiler removes unnecessary SWAP gates that the SWAP strategy added. When the circuits become deep, e.g., $k = 9$ and $p = 3$, we tend to underestimate the sampling overhead since we do not account for single-qubit gate noise. In summary, we can (1) trade approximation ratio for a partial implementation of the problem with less noise and (2) exactly recover a noiseless setting by postselecting and drawing more samples. This sampling overhead scales exponentially in the circuit depth.

V. DISCUSSION AND CONCLUSIONS

Using noisy quantum hardware to solve nonhardware-native dense optimization problems remains a challenge, which high-order optimization problems exacerbate. For dense four-local Hamiltonians, the circuit gate count scales as $O(n^4)$. The circuit depth limits the size of the LABS problem to about 30 decision variables before the noisy hardware samples are effectively random, for a depth-one QAOA and a depolarizing noise strength of $\lambda = 0.001$. This threshold is expected to be even lower for denser problems than LABS.

To reduce the circuit depth without increasing its width, we explore approximate quadratizations of the HUBO problem. These constructions sacrifice the performance guarantee of QAOA in the $p \rightarrow \infty$ limit, but recover an Ansatz with a depth linear in problem size n . This also has practical consequences. Circuits with linear depth have a higher sampling rate compared to cubic-depth circuits, allowing the quadratized QAOA to generate good candidate solutions more quickly [67]. Furthermore, shallow circuits also require a lower sampling overhead to compensate for noise [64]. Finally, quadratic terms are easier to implement on hardware than cubic or quartic terms. The clique expansion method, originally designed for mathematical applications [58], does not guarantee that the quadratized Hamiltonian preserves the ground state of the original problem. Indeed, an ideal quadratization would ensure that the ground state of the quadratized Hamiltonian and the quartic one have a nonzero overlap. In contrast, optimizing the parameters of a quadratized Hamiltonian Ansatz partially overcomes this issue. The resulting

samples have a high probability to have an approximation ratio close to one while being suboptimal.

Our simulations confirm that the standard QAOA Ansatz cannot be used in utility-scale applications because it inevitably requires error correction. Interestingly, they also show that a quadratized Ansatz yields better samples than the standard QAOA Ansatz once a threshold noise strength is reached. This makes the quadratized Ansatz particularly appealing for current quantum hardware.

Future work may further investigate on which problems this quadratization approach may be maximally efficient. Furthermore, our approach can be extended by developing hybrid methods that mix our approximate quadratization with an exact implementation of a subset of the high-order terms. This would require first identifying high-order terms that are hard to implement, for example, by a SWAP strategy, and then quadratizing them in a manner that minimizes distortions to the energy landscape. In addition, one could choose to exactly quadratize the terms in a subset E' of the fourth order terms as in Ref. [57], which would come at a cost of $2|E'|$ qubit overhead.

The second contribution of this paper is designing hardware-friendly Ansätze for QUBOs. We use SWAP strategies [42,48–50] to create a truncated cost Hamiltonian that approximates the target one. The resulting approximation ratio monotonically increases with the number of SWAP layers k in a noiseless setting. However, in the presence of noise, there is an optimal pair (k, p) that maximizes the average approximation ratio. In future work, it is worth investigating whether using SWAP strategies improves the results found in Ref. [42].

While we demonstrated our method using superconducting qubits, the idea of truncating the Hamiltonian generating the Ansatz could also be generalized to platforms in which qubit configurations can be physically changed. As a result, a truncated Hamiltonian can be created greedily: At each step, a qubit configuration is selected to maximize the number of implementable gates from the target Hamiltonian. The procedure is repeated until the maximum allowed number of qubit configurations is reached. A detailed investigation of this approach is left for future work.

Further, optimizing the QAOA variational parameters remains a challenge due to vanishing gradients [8,68–70], and an abundance of local minima [71]. Thus, solving dense problems with the typical QAOA Ansatz requires significant quantum and classical resources. Progressively, building up the Ansatz and optimizing its parameters one SWAP layer at a time may mitigate this effect. Future work will investigate the effect of additionally optimizing the weights of the truncated QUBO, similarly to our HUBOs quadratization schemes.

In summary, our work highlights the challenges in implementing high-order optimization problems on quantum hardware. We show how these challenges can be mitigated by carefully designing the Ansatz circuit from which to sample.

ACKNOWLEDGMENTS

The authors acknowledge Stefan Woerner for stimulating discussions. D.J.E. acknowledges funding within the HPQC project by the Austrian Research Promotion Agency (FFG,

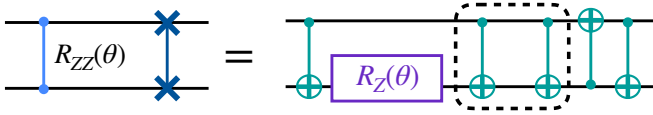


FIG. 8. Decomposition of an R_{ZZ} gate followed by a SWAP gate into a basis gate set made of single-qubit rotations and a CX. Here, two consecutive CX gates cancel. This construction is typical in SWAP networks on linear connectivity [see Fig. 2(d)].

Project No. 897481) supported by the European Union—NextGenerationEU. S.D. acknowledges funding from NCCR SwissMAP and the ETH Zurich Quantum Center.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

APPENDIX A: CIRCUIT COMPLEXITY SCALING

We now give some intuition on the number of entangling gates discussed in Sec. II and depicted in Fig. 1(a). Assuming all-to-all qubit connectivity, implementing all $n(n-1)/2$ R_{ZZ} gates requires $n(n-1)$ entangling two-qubit CZ gates with a depth of $2(n-1)$. Assuming linear qubit connectivity, we use $n-2$ SWAP layers to optimally implement a fully connected QUBO. A SWAP layer contains $\lfloor n/2 \rfloor$ or $\lfloor (n-1)/2 \rfloor$ SWAP gates, depending on its parity. This leads to an additional $\Delta N_{\text{line}} = 3 \times \lceil \frac{1}{2}(n-2) \times (\lfloor n/2 \rfloor + \lfloor (n-1)/2 \rfloor) \rceil$ entangling two-qubit gates since a SWAP is equivalent to three CZ gates. When transpiling the final circuit, the overhead gate count becomes $\Delta N_{\text{line}}/3$ due to the cancellation of neighboring entangling gates; i.e., we have only one entangling gate, instead of three, in the addition to the two corresponding to R_{ZZ} in Fig. 8. The asymptotic factor between circuit gate count on linear and all-to-all connectivity is then

$$\alpha = 1 + \lim_{n \rightarrow \infty} \frac{\Delta N_{\text{line}}/3}{n(n-1)} = 1.5, \quad (\text{A1})$$

where we add the overhead of entangling gates on top of the constant factor 1. The asymptotic factor between the circuit depth on linear and all-to-all connectivity is then

$$\alpha = \lim_{n \rightarrow \infty} \frac{3(n-2)}{2(n-1)} = 1.5. \quad (\text{A2})$$

APPENDIX B: WEIGHTS OF QUADRATIZED HYPERGRAPHS

In this Appendix, we derive an analytical formula for the weights w_{ij} of the clique expansion. From Eq. (11), we can compactly express w_{ij} . Expanding the binomials and eliminating the constant factors results in

$$\arg \min_{w_{ij}} \sum_{e \in \mathcal{E}: (i,j) \in E(e)} \alpha_e w_{ij}^2 - 2w_{ij} \alpha_e w_e, \quad (\text{B1})$$

which is now easy to minimize yielding

$$w_{ij} = \sum_{e \in \mathcal{E}: (i,j) \in E(e)} w_e^2 / \sum_{e \in \mathcal{E}: (i,j) \in E(e)} w_e. \quad (\text{B2})$$

Therefore, the weight of the resulting edges is a weighted mean of the hyperedges it belongs to.

The weights of the quadratization resulting from LABS can be more precisely calculated using the fact that all quartic and quadratic terms have weight 2 and 1, respectively. We introduce the variable I_{ij} , which is 1 if $Z_i Z_j$ is in H_{LABS} , and 0 otherwise. We also denote by N_{ij} the number of all hyperedges containing edge (i, j) . We then obtain from Eq. (11)

$$w_{ij} = \arg \min_{w_{ij}} \left[I_{ij}(w_{ij} - 1)^2 + \sum_{e \in \mathcal{E}: (i,j) \in E(e)} 2(w_{ij} - 2)^2 \right], \quad (\text{B3})$$

$$= \arg \min_{w_{ij}} [w_{ij}^2(2N_{ij} + I_{ij}) - 2w_{ij}(4N_{ij} + I_{ij})]. \quad (\text{B4})$$

The edge weights for the quadratized LABS are thus

$$w_{ij} = 2 - \frac{I_{ij}}{2N_{ij} + I_{ij}}. \quad (\text{B5})$$

The resulting weights w_{ij} are close to 2 since $I_{ij} \in \{0, 1\}$ and $N_{ij} \sim O(n)$. To see why $N_{ij} \sim O(n)$, consider another pair of nodes (h, l) with which the edge (i, j) could form a hyperedge $e = (i, j, h, l)$. The expected value of N_{ij} is the sum of the probabilities that the tuple (i, j, h, l) forms a hyperedge in the original hypergraph $\mathcal{E}$, for all pairs of nodes $h, l \in V$, i.e.,

$$\mathbb{E}(N_{ij}) = \sum_{h, l \in V} \text{Prob}[(i, j, h, l) \in \mathcal{E}]. \quad (\text{B6})$$

Since LABS has $O(n^3)$ four-local terms and there are $O(n^4)$ total possible four-local terms, then $\text{Prob}[(i, j, h, l) \in \mathcal{E}] \sim O(1/n)$ for an arbitrary pair of nodes (h, l) . Further, there are $O(n^2)$ ways of choosing the nodes (h, l) , thus yielding $\mathbb{E}(N_{ij}) \sim O(n)$.

APPENDIX C: CIRCUIT TRANSPILATION

Here, we describe how we transpile QAOA circuits for the higher-order problems. We rely on the “transpile” function of Qiskit [53], which decomposes physical operators into the basis gate set $\{R_Z, \sqrt{X}, X, \text{CZ}\}$. We assume either the default all-to-all connectivity, or a linear one, by providing a coupling map consisting of a list of connected qubit pairs $[i, i+1]$ and $[i+1, i]$, where $i \in [0, n-1]$. Qiskit will thus route the QAOA circuit to the hardware topology with the LightSABER algorithm [72], which is an enhancement of the SABER SWAP algorithm [73]. Next, the instructions in the routed circuit will be decomposed to hardware-native ones and undergo further optimizations to remove, e.g., canceling gates and swaps, as well as diagonal gates that occur before measurement instructions.

We now describe the circuit transpilation for the experiments ran on *ibm_fez* for QAOA for RR3 graphs G_3 . First, the qubit connectivity, given as a graph C , and native gate set are provided by the given backend. We then compile An-sätze circuits with SWAP strategies as described in the publicly

available “qopt-best-practices” repository [74]. For QUBOs that are not fully connected, the circuit depth can often be reduced by optimizing the mapping of the decision variables onto the physical qubits. We use the approach of Matsuo *et al.* [49], which finds such an initial mapping by solving multiple satisfiability (SAT) problems to minimize the number of SWAP layers. Indeed, applying ℓ odd and even alternating SWAP layers, as described in Sec. II and exemplified in Fig. 2(d), to the hardware-native qubit connectivity C results in a graph C_ℓ with a higher connectivity than C , for $\ell > 0$. The initial mapping problem is then equivalent to finding a permutation of the nodes of G_3 such that the permuted G_3 can be embedded in C_ℓ . This embedding can be formulated as a boolean SAT problem and the SAT mapper presented in Ref. [49] attempts to find the smallest ℓ such that a valid embedding can be found.

This method can become time-consuming since SAT problems are NP-hard. For our experiments, we set a timeout value of 20 s. After the circuit is built from R_{ZZ} and SWAP gates, we run the Qiskit transpiler on a low optimization level, i.e., 1, to decompose the gates in CZ gates and simplify any redundant gates.

APPENDIX D: MPS SIMULATION OF RR3 GRAPHS

The RR3 graph instances we investigate have 40 nodes, which prevents an exact, state-vector-based simulation of the underlying quantum circuits without high-performance computing. Hence, we use a QAOA tailored MPS-based simulator that approximates the circuit and the associated energy.

An MPS is a tensor-network representation of an N -qubit quantum state in which the wavefunction coefficients, stored as a rank- N tensor, are encoded as the contraction of N rank-3 tensors,

$$|\psi\rangle = \sum_{i_1, \dots, i_N} A_{i_1}^{[1]} A_{i_2}^{[2]} \dots A_{i_N}^{[N]} |i_1 i_2 \dots i_N\rangle, \quad (\text{D1})$$

where each $A_{i_k}^{[k]}$ is a matrix and adjacent tensors are contracted over auxiliary indices of dimension D , known as the bond

dimension. The bond dimension D controls the expressive power of the Ansatz by bounding the amount of bipartite entanglement that can be represented. Larger D allows for more entanglement and thus more accurate approximations of highly correlated states, but incurs a computational cost of calculating and optimizing the MPS energy that scales polynomially in D , limiting practical simulations to moderate bond dimensions. For all our simulations, we set the bond dimension to 20 [75].

The MPS simulator we use relies on the methods described in Ref. [76]. Specifically, the initial state is represented by an MPS state with bond dimension 1. Single-qubit gates are applied by contracting them with the corresponding tensor in the MPS. Nearest-neighbor two-qubit gates are applied as follows: The MPS tensors associated with the qubits onto which the gate is applied are first contracted together. The gate is then applied onto the resulting tensor. The original MPS format, with one tensor for each qubit, is restored by applying a singular value decomposition to the tensor. Non-nearest-neighbor two-qubit gates are simulated as they are executed in the hardware, i.e., by using SWAP layers to bring qubits close to each other. When applying this simulation protocol, the MPS canonization center must be shifted, before each gate application, to the site at which the gate is applied. This yields a computational overhead, which we tame down by encoding the MPS in the inverse Vidal form [77]. When expressed in the inverse Vidal form, an MPS is encoded as

$$|\psi\rangle = \sum_{i_1, \dots, i_N} V_{i_1}^{[1]} \Lambda^{[1]} V_{i_2}^{[2]} \Lambda^{[2]} \dots \Lambda^{[N-1]} V_{i_N}^{[N]} |i_1 i_2 \dots i_N\rangle, \quad (\text{D2})$$

where $\Lambda^{[j]}$ is a diagonal matrix containing the Schmidt coefficients associated with the bipartition between sites j and $j+1$, and $V_{i_j}^{[j]}$ are norm-1 tensors. In the inverse Vidal form, applying two-qubit gates does not involve explicitly shifting the canonization center, enabling efficient application of local gates and truncations during MPS time evolution.

-
- [1] P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE, Los Alamitos, CA, 1994), pp. 124–134.
 - [2] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, A variational eigenvalue solver on a photonic quantum processor, *Nat. Commun.* **5**, 4213 (2014).
 - [3] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, *et al.*, The variational quantum eigensolver: A review of methods and best practices, *Phys. Rep.* **986**, 1 (2022).
 - [4] Y. Cao, J. Romero, and A. Aspuru-Guzik, Potential of quantum computing for drug discovery, *IBM J. Res. Dev.* **62**, 6:1 (2018).
 - [5] N. S. Blunt, J. Camps, O. Crawford, R. Izsák, S. Leontica, A. Mirani, A. E. Moylett, S. A. Scivier, C. Sunderhauf, P. Schopf, *et al.*, Perspective on the current state-of-the-art of quantum computing for drug discovery applications, *J. Chem. Theory Comput.* **18**, 7001 (2022).
 - [6] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, *Nature (London)* **549**, 242 (2017).
 - [7] P. K. Barkoutsos, J. F. Gonthier, I. Sokolov, N. Moll, G. Salis, A. Fuhrer, M. Ganzhorn, D. J. Egger, M. Troyer, A. Mezzacapo, *et al.*, Quantum algorithms for electronic structure calculations: Particle-hole Hamiltonian and optimized wave-function expansions, *Phys. Rev. A* **98**, 022322 (2018).
 - [8] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, Connecting ansatz expressibility to gradient magnitudes and barren plateaus, *PRX Quantum* **3**, 010313 (2022).
 - [9] L. L. Peterson and S. B. Davie, *Computer Networks: A Systems Approach*, 6th ed. (Morgan Kaufmann, Cambridge, MA, 2020).

- [10] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investments* (Yale University Press, New York, 1959).
- [11] D. J. Egger, C. Gambella, J. Mareček, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, Quantum computing for finance: State-of-the-art and future prospects, *IEEE Trans. Quantum Eng.* **1**, 1 (2020).
- [12] K. Domino, A. Kundu, Ö. Salehi, and K. Krawiec, Quadratic and higher-order unconstrained binary optimization of railway rescheduling for quantum computing, *Quantum Inf. Process.* **21**, 337 (2022).
- [13] B. Pascariu, M. Sama, P. Pellegrini, A. D'Ariano, J. Rodriguez, and D. Pacciarelli, Formulation of train routing selection problem for different real-time traffic management objectives, *J. Rail Transp. Plann. Manage.* **31**, 100460 (2024).
- [14] A. Sbihi and R. W. Eglese, Combinatorial optimization and green logistics, *Ann. Oper. Res.* **175**, 159 (2010).
- [15] R. M. Karp, On the computational complexity of combinatorial problems, *Networks* **5**, 45 (1975).
- [16] D. Zuckerman, On unapproximable versions of NP-complete problems, *SIAM J. Comput.* **25**, 1293 (1996).
- [17] A. Abbas, A. Ambainis, B. Augustino, A. Bärtshi, H. Buhrman, C. Coffrin, G. Cortiana, V. Dunjko, D. J. Egger, B. G. Elmegreen, *et al.*, Challenges and opportunities in quantum optimization, *Nat. Rev. Phys.* **6**, 718 (2024).
- [18] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, *arXiv:1411.4028*.
- [19] 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).
- [20] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, Quantum annealing: A new method for minimizing multidimensional functions, *Chem. Phys. Lett.* **219**, 343 (1994).
- [21] A. Rajak, S. Suzuki, A. Dutta, and B. K. Chakrabarti, Quantum annealing: An overview, *Philos. Trans. R. Soc. A* **381**, 20210417 (2023).
- [22] T. Koch, D. E. Bernal Neira, Y. Chen, G. Cortiana, D. J. Egger, R. Heese, N. N. Hegade, A. G. Cadavid, R. Huang, T. Itoko, *et al.*, Quantum optimization benchmark library—The intractable decathlon, *arXiv:2504.03832*.
- [23] 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).
- [24] S. Brandhofer, D. Braun, V. Dehn, G. Hellstern, M. Hüls, Y. Ji, I. Polian, A. S. Bhatia, and T. Wellens, Benchmarking the performance of portfolio optimization with QAOA, *Quantum Inf. Process.* **22**, 25 (2022).
- [25] M. Willsch, D. Willsch, F. Jin, H. De Raedt, and K. Michielsen, Benchmarking the quantum approximate optimization algorithm, *Quantum Inf. Process.* **19**, 197 (2020).
- [26] G. C. Santra, F. Jendrzejewski, P. Hauke, and D. J. Egger, Squeezing and quantum approximate optimization, *Phys. Rev. A* **109**, 012413 (2024).
- [27] R. Tate, J. Moondra, B. Gard, G. Mohler, and S. Gupta, Warm-started QAOA with custom mixers provably converges and computationally beats Goemans-Williamson's Max-Cut at low circuit depths, *Quantum* **7**, 1121 (2023).
- [28] D. J. Egger, J. Mareček, and S. Woerner, Warm-starting quantum optimization, *Quantum* **5**, 479 (2021).
- [29] D. Rehfeldt, T. Koch, and Y. Shinano, Faster exact solution of sparse MaxCut and QUBO problems, *Math. Program. Comput.* **15**, 445 (2023).
- [30] D. Gamarnik, The overlap gap property: A topological barrier to optimizing over random structures, *Proc. Natl. Acad. Sci. USA* **118**, e2108492118 (2021).
- [31] D. Lykov, R. Shaydulin, Y. Sun, Y. Alexeev, and M. Pistoia, Fast simulation of high-depth QAOA circuits, in *Proceedings of the SC'23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis* (ACM, New York, 2023), pp. 1443–1451.
- [32] D. Lykov, J. Wurtz, C. Poole, M. Saffman, T. Noel, and Y. Alexeev, Sampling frequency thresholds for the quantum advantage of the quantum approximate optimization algorithm, *npj Quantum Inf.* **9**, 73 (2023).
- [33] V. Akshay, H. Philathong, M. E. S. Morales, and J. D. Biamonte, Reachability deficits in quantum approximate optimization, *Phys. Rev. Lett.* **124**, 090504 (2020).
- [34] IBM, Hardware for useful quantum computing IBM quantum roadmap, accessed 18 February 2026, <https://www.ibm.com/quantum/hardware#roadmap>.
- [35] S. V. Romero, A.-M. Visuri, A. G. Cadavid, A. Simen, E. Solano, and N. N. Hegade, Bias-field digitized counterdiabatic quantum algorithm for higher-order binary optimization, *Commun. Phys.* **8**, 348 (2025).
- [36] J. Brest and B. Bošković, Low autocorrelation binary sequences: Best-known peak sidelobe level values, *IEEE Access* **9**, 67713 (2021).
- [37] T. Packebusch and S. Mertens, Low autocorrelation binary sequences, *J. Phys. A: Math. Theor.* **49**, 165001 (2016).
- [38] R. Shaydulin, C. Li, S. Chakrabarti, M. DeCross, D. Herman, N. Kumar, J. Larson, D. Lykov, P. Minssen, Y. Sun, *et al.*, Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem, *Sci. Adv.* **10**, eadm6761 (2024).
- [39] Gurobi Optimization LLC, <https://www.gurobi.com>.
- [40] IBM ILOG CPLEX. V20.1: User's manual for cplex, <https://www.ibm.com/docs/en/icos/22.1.2?topic=optimizers-users-manual-cplex>.
- [41] A. Lucas, Ising formulations of many NP problems, *Front. Phys.* **2**, 5 (2014).
- [42] F. B. Maciejewski, S. Hadfield, B. Hall, M. Hodson, M. Dupont, B. Evert, J. Sud, M. S. Alam, Z. Wang, S. Jeffrey, *et al.*, Design and execution of quantum circuits using tens of superconducting qubits and thousands of gates for dense Ising optimization problems, *Phys. Rev. Appl.* **22**, 044074 (2024).
- [43] L. Zhu, H. L. Tang, G. S. Barron, F. A. Calderon-Vargas, N. J. Mayhall, E. Barnes, and S. E. Economou, Adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer, *Phys. Rev. Res.* **4**, 033029 (2022).
- [44] F. G. Fuchs, K. O. Lye, H. M. Nilsen, A. J. Stasik, and G. Sartor, Constraint preserving mixers for the quantum approximate optimization algorithm, *Algorithms* **15**, 202 (2022).
- [45] Z. He, R. Shaydulin, S. Chakrabarti, D. Herman, C. Li, Y. Sun, and M. Pistoia, Alignment between initial state and mixer improves QAOA performance for constrained optimization, *npj Quantum Inf.* **9**, 121 (2023).

- [46] M. Y. Niu, S. Lu, and I. L. Chuang, Optimizing QAOA: Success probability and runtime dependence on circuit depth, [arXiv:1905.12134](#).
- [47] We only consider two-qubit gates in gate counts as these are the main source of errors. For instance, the median error rates of the CZ gates is 8.30×10^{-3} , compared to 2.23×10^{-4} for single-qubit gates on *ibm_fez*. Similarly, when computing circuit depth, we refer only to the depth required to implement two-qubit gates.
- [48] J. Weidenfeller, L. C. Valor, J. Gacon, C. Tornow, L. Bello, S. Woerner, and D. J. Egger, Scaling of the quantum approximate optimization algorithm on superconducting qubit based hardware, *Quantum* **6**, 870 (2022).
- [49] A. Matsuo, S. Yamashita, and D. J. Egger, A SAT approach to the initial mapping problem in SWAP gate insertion for commuting gates, *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **E106.A**, 1424 (2023).
- [50] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo, *et al.*, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor, *Nat. Phys.* **17**, 332 (2021).
- [51] M. Golay, Sieves for low autocorrelation binary sequences, *IEEE Trans. Inf. Theory* **23**, 43 (1977).
- [52] J. Bernasconi, Low autocorrelation binary sequences : Statistical mechanics and configuration space analysis, *J. Phys. France* **48**, 559 (1987).
- [53] 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](#).
- [54] V. Rosenhaus, An introduction to the SYK model, *J. Phys. A: Math. Theor.* **52**, 323001 (2019).
- [55] Z. Luo, Y.-Z. You, J. Li, C.-M. Jian, D. Lu, C. Xu, B. Zeng, and R. Laflamme, Quantum simulation of the non-Fermi-liquid state of Sachdev-Ye-Kitaev model, *npj Quantum Inf.* **5**, 53 (2019).
- [56] B. Kobrin, Z. Yang, G. D. Kahanamoku-Meyer, C. T. O'Connell, J. E. Moore, D. Stanford, and N. Y. Yao, Many-body chaos in the Sachdev-Ye-Kitaev model, *Phys. Rev. Lett.* **126**, 030602 (2021).
- [57] A. Mandal, A. Roy, S. Upadhyay, and H. Ushijima-Mwesigwa, Compressed quadratization of higher order binary optimization problems, in *Proceedings of the 17th ACM International Conference on Computing Frontiers* (Association for Computing Machinery, New York, 2020), pp. 126–131.
- [58] V. Suppakitpaisarn and J.-K. Hao, Utilizing graph sparsification for pre-processing in max cut QUBO solver, in *Metaheuristics International Conference* (Springer, Cham, 2024), pp. 219–233.
- [59] R. Herrman, P. C. Lotshaw, J. Ostrowski, T. S. Humble, and G. Siopsis, Multi-angle quantum approximate optimization algorithm, *Sci. Rep.* **12**, 6781 (2022).
- [60] S. H. Sack, R. A. Medina, R. Kueng, and M. Serbyn, Recursive greedy initialization of the quantum approximate optimization algorithm with guaranteed improvement, *Phys. Rev. A* **107**, 062404 (2023).
- [61] Noise sensitivity is computed as
- $$\frac{\langle E_{\text{QUBO}}(\lambda = 0.01) \rangle - \langle E_{\text{QUBO}}(\lambda = 0.001) \rangle}{\langle E_{\text{LABS}}(\lambda = 0.01) \rangle - \langle E_{\text{LABS}}(\lambda = 0.001) \rangle}.$$
- [62] S. H. Sack and D. J. Egger, Large-scale quantum approximate optimization on nonplanar graphs with machine learning noise mitigation, *Phys. Rev. Res.* **6**, 013223 (2024).
- [63] M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *J. ACM* **42**, 1115 (1995).
- [64] S. V. Barron, D. J. Egger, E. Pelofske, A. Bärttschi, S. Eidenbenz, M. Lehmkuehler, and S. Woerner, Provable bounds for noise-free expectation values computed from noisy samples, *Nat. Comput. Sci.* **4**, 865 (2024).
- [65] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, Improving variational quantum optimization using CVaR, *Quantum* **4**, 256 (2020).
- [66] D. C. McKay, I. Hincks, E. J. Pritchett, M. Carroll, L. C. G. Govia, and S. T. Merkel, Benchmarking quantum processor performance at scale, [arXiv:2311.05933](#).
- [67] A. Wack, H. Paik, A. Javadi-Abhari, P. Jurcevic, I. Faro, J. M. Gambetta, and B. R. Johnson, Quality, speed, and scale: Three key attributes to measure the performance of near-term quantum computers, [arXiv:2110.14108](#).
- [68] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, Barren plateaus in variational quantum computing, *Nat. Rev. Phys.* **7**, 174 (2025).
- [69] A. Arrasmith, Z. Holmes, M. Cerezo, and P. J. Coles, Equivalence of quantum barren plateaus to cost concentration and narrow gorges, *Quantum Sci. Technol.* **7**, 045015 (2022).
- [70] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, *Nat. Commun.* **9**, 4812 (2018).
- [71] L. Bittel and M. Kliesch, Training variational quantum algorithms is NP-hard, *Phys. Rev. Lett.* **127**, 120502 (2021).
- [72] H. Zou, M. Treinish, K. Hartman, A. Ivrii, and J. Lishman, LightSABRE: A lightweight and enhanced SABRE algorithm, [arXiv:2409.08368](#).
- [73] 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, NY, 2019), pp. 1001–1014.
- [74] Quantum optimization best practices, <https://github.com/qiskit-community/qopt-best-practices>.
- [75] J. Gray, quimb: A Python package for quantum information and many-body calculations, *J. Open Source Software* **3**, 819 (2018).
- [76] Y. Zhou, E. M. Stoudenmire, and X. Waintal, What limits the simulation of quantum computers? *Phys. Rev. X* **10**, 041038 (2020).
- [77] P. Secular, N. Goulianov, M. Lubasch, S. Dolgov, S. R. Clark, and D. Jaksch, Parallel time-dependent variational principle algorithm for matrix product states, *Phys. Rev. B* **101**, 235123 (2020).