Low-depth Clifford circuits approximately solve MaxCut

Manuel H. Muñoz-Arias,[1,*] Stefanos Kourtis,[1] and Alexandre Blais[1,2]
[1]*Institut Quantique and Département de Physique, Université de Sherbrooke, Sherbrooke, Quebec J1K 2R1, Canada*
[2]*Canadian Institute for Advanced Research, Toronto, Ontario M5G 1M1, Canada*

(Received 19 November 2023; revised 4 December 2023; accepted 28 May 2024; published 20 June 2024)

We introduce a quantum-inspired approximation algorithm for MaxCut based on low-depth Clifford circuits. We start by showing that the solution unitaries found by the adaptive quantum approximation optimization algorithm (ADAPT-QAOA) for the MaxCut problem on weighted fully connected graphs are (almost) Clifford circuits. Motivated by this observation, we devise an approximation algorithm for MaxCut, ADAPT-Clifford, that searches through the Clifford manifold by combining a minimal set of generating elements of the Clifford group. Our algorithm finds an approximate solution of MaxCut on an N-vertex graph by building a depth $O(N)$ Clifford circuit. The algorithm has runtime complexity $O(N^2)$ and $O(N^3)$ for sparse and dense graphs, respectively, and space complexity $O(N^2)$, with improved solution quality achieved at the expense of more demanding runtimes. We implement ADAPT-Clifford and characterize its performance on graphs with positive and signed weights. The case of signed weights is illustrated with the paradigmatic Sherrington-Kirkpatrick model, for which our algorithm finds solutions with ground-state mean energy density corresponding to ∼94% of the Parisi value in the thermodynamic limit. The case of positive weights is investigated by comparing the cut found by ADAPT-Clifford with the cut found with the Goemans-Williamson (GW) algorithm. For both sparse and dense instances we provide copious evidence that, up to hundreds of nodes, ADAPT-Clifford finds cuts of lower energy than GW.

DOI: 10.1103/PhysRevResearch.6.023294

I. INTRODUCTION

Near-term quantum processors have found a niche in the hybrid quantum-classical model of computation with variational quantum algorithms (VQAs) [1–3]. These algorithms perform *classical* optimization of a problem-specific objective function that is evaluated by measuring the output of a parametrized *quantum* circuit. Through variational search over circuit parameters, a VQA thus seeks a solution circuit that transforms a simple input state in the Hilbert space of problem variables to a superposition of approximate solutions, i.e., configurations that yield near-optimal values of the objective function. In attempts to determine whether they can lead to a speedup over classical algorithms for any useful task, VQAs have been applied to a variety of combinatorial optimization problems.

Quantum approximate optimization algorithms (QAOAs) [4,5] have been in a constant tug of war with classical solvers, with initial indications of putative quantum speedups [6–9], followed by experimental claims [10] and rebuttals [11], and then further proposals for possible quantum speedup [12]. A byproduct of this large effort has been the definition and construction of quantum-inspired algorithms [13–18]. These are "dequantized" classical versions of quantum or hybrid quantum-classical algorithms that unveil and exploit previously unrecognized properties or structures in a problem, leading to solution strategies that outperform the best known classical algorithm.

In this work we introduce a quantum-inspired approximation algorithm for the MaxCut problem. The algorithm, which we dub ADAPT-Clifford, is motivated by the observation that the solution circuits found by an adaptive QAOA variant for MaxCut on weighted complete graphs are (almost) Clifford circuits, a well-known restricted class of quantum circuits that are easy to simulate classically [19–23]. ADAPT-Clifford builds an entangled state with a number of unitary operations that is equal to the number of nodes N, adding at every step a known two-qubit gate to the circuit. The algorithm is polynomial in both time and space, with worst case runtime complexity $O(N^4)$ and space complexity $O(N^2)$. We characterize the performance of the algorithm in several families of graphs. For graph sizes up to $N = 30$ nodes we report the exact approximation ratios. For larger problem sizes up to hundreds of nodes and depending on graph family, we assess the performance by either direct comparison with the solution found by the best classical algorithm for MaxCut [24,25], or by comparing with the known value of the mean energy density in the thermodynamic limit.

The rest of this paper is organized as follows. In Sec. II we present a short summary of the MaxCut problem, quantum approximate optimization, and its adaptive variant, Clifford circuits, and introduce the tools we use to characterize the

*Contact author: munm2002@usherbrooke.ca

structure of solution circuits. In Sec. III we present the origin of the algorithm by analyzing the structure of solution circuits to MaxCut on weighted complete graphs obtained with QAOA and ADAPT-QAOA. In Sec. IV we present the ADAPT-Clifford algorithm, discuss its details, and analyze its runtime and space complexities in different types of graphs. In Sec. V we present numerical results of the algorithm performance on weighted complete graphs with positive and signed weights, using the Sherrington-Kirkpatrick model as a specific example of the latter. In Sec. VI we explore the performance of the algorithm beyond complete graphs, including weighted and unweighted K-regular graphs and Erdős-Rényi graphs. Finally in Sec. VII we conclude with a discussion of our results in the context of near-term quantum optimization algorithms and present an outlook for future work.

II. BACKGROUND AND METHODS

A. The MaxCut problem

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the vertex set and $\mathcal{E} \subseteq \mathcal{V}^2$ is the set of edges ($\mathcal{E} = \mathcal{V}^2$ is a complete graph), and edge weights $\omega_{i,j} \in \mathbb{R}$ for $(i, j) \in \mathcal{E}$, the MaxCut problem asks to partition $\mathcal{V}$ into two complementary subsets $\mathcal{A}, \overline{\mathcal{A}} \subseteq \mathcal{V}$, such that the total weight of the edges between $\mathcal{A}$ and $\overline{\mathcal{A}}$ is maximized. We use binary variables $z_i \in \{0, 1\}$, $i \in \mathcal{V}$ to help us identify each subset, so that $z_i = 1$ if vertex $i \in \mathcal{A}$ and $z_i = 0$ if $i \in \overline{\mathcal{A}}$. The maximal cut can then be formally expressed as the assignment $\mathbf{z}'$ that maximizes the cost function

$$C(\mathbf{z}) = \sum_{(i,j)\in\mathcal{E}} \omega_{i,j} z_i (1 - z_j), \tag{1}$$

where $\mathbf{z} = z_1 \dots z_N$ is a N-bit binary string and $\omega_{i,j} = \omega_{j,i}$ $\forall (i, j) \in \mathcal{E}$.

The MaxCut problem on general graphs is known to be NP-hard [26,27]. However, MaxCut can be solved in polynomial time in some special cases, such as graphs without long odd cycles [28], weakly bipartite graphs [29], planar graphs both weighted [30,31] and unweighted [32,33], 1-planar graphs [34], and graphs with k crossings [35]. Finally, when all the edge weights are negative, MaxCut becomes a equivalent to MinCut and admits a polynomial time algorithm [36].

Beyond the special cases mentioned above and due to the difficulty of solving the problem exactly, one often aims instead to find approximation algorithms that yield reasonably good solutions in polynomial time for all problem instances. That is, we search for an algorithm that outputs an assignment $\mathbf{z}^*$, such that the approximation ratio

$$\alpha(\mathbf{z}^*) = \frac{C(\mathbf{z}^*)}{\max_{\mathbf{z}}[C(\mathbf{z})]} \tag{2}$$

equals some desired value, ideally as close to 1 as possible, on all instances of MaxCut. However, in some cases the gap between approximate an optimal solutions cannot be reduced arbitrarily in polynomial time [37], a phenomenon known as hardness of approximation. For MaxCut on general graphs, the best known approximation algorithm is that of Goemans and Williamson (GW) [24], which has a performance guarantee (worst case) of $\alpha \simeq 0.878$ [24,25,38,39]. Below we will present extensive performance comparisons between ADAPT-Clifford and the GW algorithm; in order to be as self-contained as possible we review the details of the GW algorithm in Appendix A.

One might hope to achieve better approximation ratios by focusing on specific families of graphs. For unweighted graphs Ref. [40] showed that finding an algorithm yielding an approximation ratio better than 16/17 is NP-hard. Nearly optimal algorithms both for cubic graphs [41], guaranteeing $\alpha = 0.9326$, and for K-regular graphs of large degree [42] are known. Another interesting example is the case of dense graphs, i.e., graphs with $O(N^2)$ edges. Polynomial time approximation schemes (PTASs) are known for both unweighted [43,44] and weighted [45] graphs, although for the latter there is only an existence result. A PTAS guarantees an approximate solution whose cost is $1 - \epsilon$ away from the optimal. Although these schemes have a provable polynomial runtime in N, it might not be polynomial in ϵ; see, for example, Ref. [44]. We will come back to this point in Sec. VII.

In quantum approximate optimization, the objective function of a combinatorial optimization problem defined on binary variables z_i, such as MaxCut, is expressed as an Ising Hamiltonian through the mapping $\sigma_i = 2z_i - 1$, with connectivity dictated by the graph $\mathcal{G}$ [46]; that is, the entries $\omega_{i,j}$ of the adjacency matrix reflect the coupling between the ith and jth spin. In this setting the optimum $\mathbf{z}'$ is encoded as the ground state of the Ising Hamiltonian. The Ising Hamiltonian is then promoted to a Hamiltonian operator via the identification $\sigma_i \to Z_i$, with Z_i a Pauli-z operator acting on the qubit that corresponds to the ith spin. For the MaxCut problem, the corresponding Hamiltonian is

$$H_C = \frac{1}{2} \sum_{i<j} \omega_{i,j} Z_i Z_j. \tag{3}$$

In writing Eq. (3) we have dropped a constant factor equal to $\sum_{i<j} \frac{\omega_{i,j}}{2}$ and added a minus sign to turn the maximization problem defined by Eq. (1) into a minimization one.

In analogy with classical approximation algorithms, quantum approximate optimization yields approximate solutions in the form of a state $|\phi\rangle$, whose energy expectation is as close as possible to the ground-state energy of the Ising Hamiltonian. Thus, the approximation ratio, Eq. (2), takes the form

$$\alpha = \frac{\langle\phi|H_C|\phi\rangle}{E^C_{\min}}, \tag{4}$$

where $E^C_{\min}$ is the smallest eigenvalue of H_C. To achieve advantageous performance, a quantum algorithm must produce an approximate solution with a desired α faster than any classical algorithm.

B. The quantum approximate optimization algorithm and its adaptive variant

The Quantum Approximate Optimization Algorithm (QAOA) is a type of variational algorithm [2] that aims to solve combinatorial optimization problems [4]. It is defined by a parametrized quantum circuit with a periodic structure. Each layer of the circuit is given by a product of two unitaries, time evolution under H_C, followed by time evolution under a

mixer Hamiltonian

$$H_{\mathrm{M}} = \sum_{j=1}^{N} X_j, \tag{5}$$

where X_j is a Pauli-x on the jth qubit. For p layers QAOA prepares the state

$$|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle_p = \left[\prod_{l=1}^{p} e^{-i\beta_l H_{\mathrm{M}}} e^{-i\gamma_l H_{\mathrm{C}}}\right] \mathrm{H}^{\otimes N} |0\rangle^{\otimes N}, \tag{6}$$

where $\boldsymbol{\gamma} = \gamma_1, \ldots, \gamma_p$, $\boldsymbol{\beta} = \beta_1, \ldots, \beta_p$, and H is the Hadamard gate. In order to find approximate solutions the set of $2p$ parameters is optimized so as to minimize $\langle\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})|H_{\mathrm{C}}|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle_p$. After executing the circuit with optimized parameters a measurement in the computational basis returns a candidate solution in the form of a bit string $\mathbf{z}^*$. Ideally, one would sample with high probability a good approximate solution. We will denote the optimal parameters found by numerical experiments as $(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$ and the associated solution unitary $U(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*) = \prod_{l=1}^{p} e^{-i\beta_l^* H_{\mathrm{M}}} e^{-i\gamma_l^* H_{\mathrm{C}}}$.

Not much is known regarding performance guarantees and hardness of approximation for QAOA. The case of constant p has so far been the main focus, as it is the regime of interest for current quantum devices [47]. Reference [48] gives evidence for a possible quantum advantage for MaxCut on 3-regular graphs with shallow QAOA. References [49,50] provide evidence that both $p = 1$ and large-depth QAOA output states with bit string probabilities following Boltzman distributions, rendering sampling classically hard. At the same time, it is known that constant p QAOA is bounded away from optimality in sparse graphs [51–53], as well as in some dense problems where the overlap gap property [54] is known to exist [55]. These results were recently extended to the case of $p \sim \log(N)$ [56]. However these results do not apply directly to the case of $p \sim \mathrm{poly}(N)$. As a consequence there are no conclusive results on the runtime required for $p \sim \mathrm{poly}(N)$ QAOA to reach a given approximation ratio, with only loose lower bounds appearing recently [57]. Most studies of QAOA so far have been numerical experiments on different families of problem instances; two examples are Erdős-Rényi graphs [6] and 3- and 4-regular graphs (weighted and unweighted) [58]. Importantly, any indication of a putative advantage in this type of studies has been inconclusive due to the small problem sizes accessible to either quantum implementations or classical simulation [59].

To alleviate some of the roadblocks explored above, variants to the original QAOA ansatz have been developed; see Ref. [60] for a review. Of interest to us here is the ADAPT [61] variant, which was proposed as a way to find ansätze which are tailored to the specifics of the problem under consideration. ADAPT-QAOA is an iterative variational algorithm which replaces the fixed mixer Hamiltonian in Eq. (6), by a suitably chosen one, A_l, at each layer $l \leqslant p$. Thus, p-layer ADAPT-QAOA prepares the state

$$|\psi(\gamma, \beta)\rangle_p^{\mathrm{ADAPT}} = \left[\prod_{l=1}^{p} e^{-i\beta_l A_l} e^{-i\gamma_l H_{\mathrm{C}}}\right] \mathrm{H}^{\otimes N} |0\rangle^{\otimes N}. \tag{7}$$

The lth mixer Hamiltonian is chosen as the one which maximizes the energy gradient, that is,

$$A_l = \max_{A_s \in \mathrm{P_{OP}}} [-i\langle\psi_{l-1}|e^{i\gamma_l H_{\mathrm{C}}}[H_{\mathrm{C}}, \hat{A}_s]e^{-i\gamma_l H_{\mathrm{C}}}|\psi_{l-1}\rangle], \tag{8}$$

where the new variational parameter γ_l is set to a predefined small positive value $\gamma_0 \sim 0$ [61], $\mathrm{P_{OP}}$ is an operator pool, and $|\psi_{l-1}\rangle$ is the state resulting from the application of the ADAPT-QAOA solution circuit with only $l-1$ layers. The choice of pool is not unique, with different pools being advantageous in different situations [62,63]. Below we restrict ourselves to the pool

$$\mathrm{P_{OP}} = \left\{\sum_i X_i, \sum_i Y_i\right\} \cup \{X_j, Y_j\}_{j=1,\ldots,N} \cup \{X_jX_k, Y_jY_k, Y_jZ_k, Z_jY_k\}_{j,k=1,\ldots,N, j\neq k}, \tag{9}$$

which is sufficient for our purposes.

In contrast to QAOA, ADAPT-QAOA grows the circuit layer by layer, until the desired number p. As such, we begin with a single layer, find the corresponding mixer according to Eq. (8), then optimize to find the best parameters. We then add a second layer, find the corresponding mixer according to Eq. (8), initialize the new pair of parameters to zero [64] and the rest of the parameters to the best values already found, and optimize all of them. This procedure is repeated until p layers are added. For a fair comparison between QAOA and ADAPT-QAOA in our numerical simulations we construct the QAOA solution circuit following the same iterative strategy, but with a fixed mixer.

C. Clifford circuits and their efficient simulation

In this subsection we review some concepts of the stabilizer formalism which will be used later in the paper. For a general presentation see Ref. [65].

The single-qubit Pauli group is given by the operators $\{\mathbb{I}, X, Y, Z\}$ together with multiplicative factors $\pm 1, \pm i$. The N-qubit Pauli group $\tilde{\mathcal{P}}_N$ is given by all the N-tensor products of these operators together with multiplicative factors. Given a pure state on N qubits $|\psi\rangle$, we say $\tilde{P}_i \in \tilde{\mathcal{P}}_N$ stabilizes $|\psi\rangle$ if the state is an eigenvector of $\tilde{P}_i$ with eigenvalue $+1$: $\tilde{P}_i|\psi\rangle = |\psi\rangle$. A n-qubit pure state is a stabilizer state if it can be completely specified, up to a global phase, by its N stabilizers.

Quantum circuits which map stabilizer states to stabilizer states define a large class of nontrivial quantum circuits—stabilizer circuits—which can be simulated in polynomial time on a classical computer [22,23]. This is the content of the celebrated Gottesman-Knill theorem [20,21]. These quantum circuits can be completely written in terms of controlled-NOT, Hadamard, and phase gates and single-qubit measurements. Importantly, the efficient classical simulability does not imply these circuits are not interesting. On the contrary, they have extensive applications in quantum information science, for instance, encoding and decoding in quantum error correction [19,21,66,67], dense quantum coding [68], quantum teleportation [69], quantum simulation [70,71], and proof of principle of quantum advantage with nonlocal games [72,73], as well as in quantum many-body physics [74–78].

In absence of measurements, stabilizer circuits are referred to as Clifford circuits or Clifford unitaries. They form a group

$\mathcal{C}$, defined as the unitaries which normalize the Pauli group, that is, the unitaries which map Pauli operators to Pauli operators. Following from the Gottesman-Knill theorem, this group has three generators, the controlled-NOT, Hadamard, and phase gates. Naturally the Pauli operators are elements of the group, as they are generated by Hadamard and phase.

Both the QAOA and ADAPT-QAOA ansätze are defined as products of unitaries generated by Pauli strings. When are unitary transformations generated by Pauli strings Clifford unitaries? To answer this question, take $\tilde{P}_i, \tilde{P}_j \in \tilde{\mathcal{P}}$, two distinct Pauli strings that either commute or anticommute by definition. Further consider the unitary $W(\theta) = e^{-i\theta\tilde{P}_j}$, then $W^\dagger \tilde{P}_i W = \tilde{P}_i$ if $[\tilde{P}_i, \tilde{P}_j] = 0$, and $W^\dagger \tilde{P}_i W = i\tilde{P}_i\tilde{P}_j$ if $\{\tilde{P}_i, \tilde{P}_j\} = 0$ and $\theta = \pm m\frac{\pi}{4}$ with $m \in \mathbb{N}$. We thus see that when a quantum circuit is composed of products of unitaries generated by Pauli strings that do not necessarily commute, it is a Clifford circuit if and only if the parameters of these transformations are integer multiples of $\pm\pi/4$. Therefore, if QAOA solution circuits $U(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$ are to be Clifford, then the circuit parameters $\boldsymbol{\gamma}^*$ and $\boldsymbol{\beta}^*$ must be integer multiples of $\pm\pi/4$.

D. Characterizing the structure of solution unitaries

Here we introduce the tools we use in the next section to characterize the structure of QAOA solution circuits $U(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$. Consider the Hilbert space $\mathcal{H}$ of N qubits with dimension $d = 2^N$, and define the N-qubit Pauli basis as $\mathcal{P}_N = \tilde{\mathcal{P}}_N/\langle \pm i\mathbb{I}\rangle$, the quotient group containing, $D = 4^N - 1$, Pauli strings with all multiplicative factors equal to $+1$. Furthermore any pair of Pauli strings obey $\mathrm{Tr}[P_iP_j] = d\delta_{ij}$. Therefore, $\mathcal{P}_N$ defines a basis for all Hermitian operators in $\mathcal{H}$.

Consider some Hermitian operator O acting on $\mathcal{H}$. If O evolves under some unitary transformation V, we write

$$O' = V^\dagger O V = \sum_{j=1}^{D} f[P_j; O']P_j, \tag{10}$$

with $P_j \in \mathcal{P}_N$. Noticing that $\sum_j |f[P_j; O']|^2 = \mathrm{Tr}[O'^2] = \mathrm{Tr}[O^2]$, we define

$$p_j(O; V) = \frac{1}{\mathrm{Tr}[O^2]}|f[P_j; O']|^2. \tag{11}$$

It is easy to see that $\sum_j p_j = 1$. Equations (11) thus denotes the probability of finding O' to be the jth Pauli string P_j. In the case of $O = P_l$, the normalization factor in Eq. (11) is $\sum_j |f[P_j; O']|^2 = d$.

We analyze the transformation V as an "input-output" channel, with O the input and O' the output, and we are interested in characterizing the locality, in the Pauli basis, of the output. This can be inferred from the the localization properties of $p_j(O; V)$, which we investigate with the second Rényi entropy (see Ref. [79] and Sec. 2.7 of Ref. [80]):

$$\mathcal{S}(O; V) = -\log\left(\sum_{j=0}^{4^N} \frac{|f[P_j; O']|^4}{d^2}\right). \tag{12}$$

Equation (12) has a resemblance to the stabilizer Rényi entropy [81]. Although the latter quantifies the nonstabilizerness of a multiqubit state, the expression in Eq. (12) directly looks at non-Cliffordness of the transformation. As such, one might interpret it as the operator space counterpart to the stabilizer Rényi entropy, and we expect both quantities to have similar behaviors, that is, if for a multiqubit state $V|\psi\rangle^{\otimes N}$ the stabilizer Rényi entropy is high/low, then Eq. (12) for some input Pauli string O and the same unitary V will be high/low.

The $P_l \in \mathcal{P}_N$ can be ordered by their "weight," i.e., the number of nonidentity elements in the Pauli string. This ordering allows us to systematically study the Clifford character of the transformation V on Pauli strings. Naturally, the first step will be to check it for strings of weight one, which is done by setting $O = Y_n$, where Y_n denotes a Pauli operator with a Pauli-y on the nth qubit position and identity everywhere else. In particular we denote $\mathcal{S}(Y_n; V) = \mathcal{S}_n(V)$. Since we can place the initial Pauli-y at any of the N positions representing the nodes of the graph, we consider the node-averaged Renyi entropy of the operator distribution

$$\overline{\mathcal{S}}(V) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{S}_n(V) \tag{13}$$

as our figure of merit. Since Clifford unitaries map Pauli strings to Pauli strings, then $\mathcal{S}(O; V) = 0$ for all $O \in \mathcal{P}_N$. Since we are checking only the behavior of V as a "channel" for Pauli strings localized on one qubit, a vanishing $\overline{\mathcal{S}}$ is necessary (but not sufficient) for V to be Clifford. We thus use $\overline{\mathcal{S}}$ as evidence of Cliffordness.

We supplement this evidence with an examination of the optimal parameters $(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$. Observation of $\boldsymbol{\gamma}^*, \boldsymbol{\beta}^* = \pm m\frac{\pi}{4}$ with $m \in \mathbb{N}$ then provides the sufficient condition for V to be Clifford. This observation is made quantitative via the distance of the vector of parameters $\mathbf{v}$ to the discrete set of interest. We define this distance as

$$\mathcal{D}(\mathbf{v}) = \sum_{v_i \in \mathbf{v}} \frac{\min_{l\in\mathbb{Z}}\left[\left|v_i - l\frac{\pi}{4}\right|\right]}{\pi/8}, \tag{14}$$

where the normalization ensures that each term in the sum is bounded to the interval [0,1], thus we have $0 \leqslant \mathcal{D}(\mathbf{v}) \leqslant |\mathbf{v}|$. Then the instance-averaged distances $\mathbb{E}[\mathcal{D}(\gamma^*)] \to 0$ and $\mathbb{E}[\mathcal{D}(\beta^*)] \to 0$ will disclose solution circuits which are close to Clifford.

III. ORIGIN OF THE ADAPT-CLIFFORD ALGORITHM

To understand the origin of the ADAPT-Clifford algorithm, it is instructive to examine the solution circuits obtained with QAOA and ADAPT-QAOA for MaxCut on small weighted complete graphs. We implemented both variational algorithms in the extensible Julia framework Yao.jl [82] and use the COBYLA optimizer. The analysis of the operator distribution in the Pauil basis was implemented using QuantumOptics.jl [83].

We consider first the case of graphs with positive weights with $\omega_{i,j}$ from either U[0, 1], where by U[a, b] we denote the uniform distribution in the interval [a, b], or Exp(1), the exponential distribution with mean 1. In Fig. 1(a) we show the mean approximation ratios for 50 problem instances with $N = 6$ for circuits up to $p = 10$ layers. Similar to the observation in Ref. [61], ADAPT-QAOA [green diamonds and circles in Fig. 1(a)] finds a solution arbitrarily close to the exact solution at sufficiently high but finite p, away from the $p \to \infty$ limit

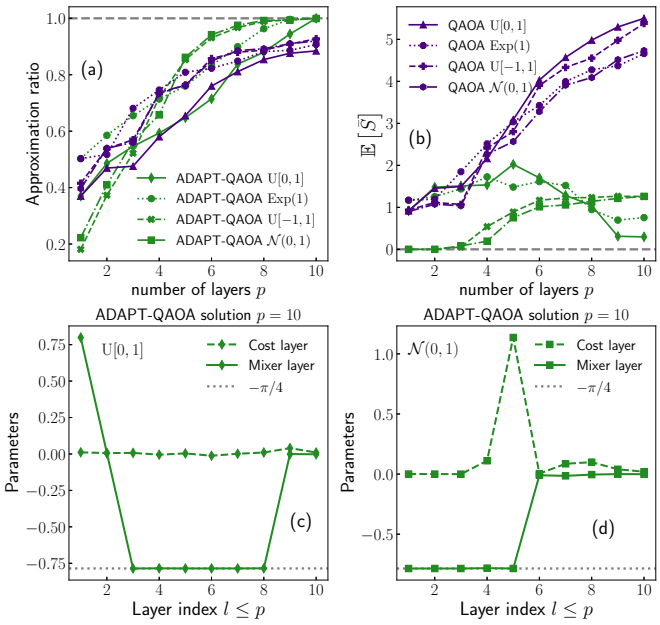

FIG. 1. QAOA and ADAPT-QAOA results for MaxCut on weighted complete graphs with $N = 6$. (a) Instance average of α for QAOA (purple) and ADAPT-QAOA (green). (b) Instance average of $\overline{\mathcal{S}}$ in Eq. (13) of solution circuits with p layers fore QAOA (purple) and ADAPT-QAOA (green). For both (a) and (b) the distributions of the weights used are indicated in the figure. (c), (d) Examples of the parameters $\boldsymbol{\gamma}^*$ (cost, solid line) and $\boldsymbol{\beta}^*$ (mixer, dashed line) of the solution unitary $U(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$ for an instance with weights drawn from U[0, 1] (c) and $\mathcal{N}(0, 1)$ (d).

where QAOA is guaranteed to reach the exact solution. In Fig. 1(b) we show the expectation value over instances of $\overline{\mathcal{S}}$, $\mathbb{E}[\overline{\mathcal{S}}]$. The ADAPT-QAOA solution circuits that lead to $\alpha \to 1$ in Fig. 1(a) display $\mathbb{E}[\overline{\mathcal{S}}] \to 0$, indicating they might be Clifford circuits. In contrast, the QAOA solution unitaries show $\mathbb{E}[\overline{\mathcal{S}}] > 0$ with a tendency towards the typical value, $\log(4^{-N})$, with increasing depth, in agreement with previous works using other indicators [84–86].

To verify the Cliffordness of the ADAPT-QAOA solution circuits we examine the optimized parameters, $(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$, at $p = 10$. An example is shown in Fig. 1(c) where dashed and solid lines correspond to $\boldsymbol{\gamma}^*$ and $\boldsymbol{\beta}^*$, respectively. We observe $\boldsymbol{\gamma}^* \to 0$ and $\boldsymbol{\beta}^* \to -\pi/4$ in all layers, and the mixer Hamiltonians selected by the adaptive step are almost always $Y_l Z_m$ for some pair of qubits (l, m). Furthermore the distances of the optimal parameters for $p = 10$ to $\pm s\frac{\pi}{4}$ with $s \in \mathbb{N}$ averaged over all instances with $\alpha \to 1$ ($\omega_{i,j} \in$ U[0, 1]) are $\mathbb{E}[\mathcal{D}(\gamma^*)] = 0.522 \pm 0.270$ and $\mathbb{E}[\mathcal{D}(\beta^*)] = 0.247 \pm 0.322$, indicating the optimized parameters are closed, on average, to the Clifford values. This is to be contrasted with $\mathbb{E}[\mathcal{D}(\gamma^*)] = 3.53 \pm 0.94$ and $\mathbb{E}[\mathcal{D}(\beta^*)] = 3.41 \pm 0.62$ for the optimized parameters of the QAOA solution circuits with $p = 10$. Finally, we extensively checked that the properties of the ADAPT-QAOA solution unitary discussed here do not change as long as the edge weights are all positive.

Next we consider the case of signed weights with $\omega_{i,j}$ sampled either from U[−1, 1] or $\mathcal{N}(0, 1)$, the normal distribution with mean 0 and variance 1. In Fig. 1(a) we compare the averaged approximation ratio of the ADAPT-QAOA solutions with that of the QAOA solutions for the same problem instances. Similar to the case of strictly positive weights, ADAPT-QAOA solutions get arbitrarily close to the exact solution when enough layers are considered. As seen in Fig. 1(b) $\mathbb{E}[\overline{\mathcal{S}}] \neq 0$ for the ADAPT-QAOA solution (green crosses and squares). Although the circuits found are therefore not Clifford, $\mathbb{E}[\overline{\mathcal{S}}] \sim 1$ at $p = 10$ for the small problem size under study, in contrast to QAOA solution circuits (purple crosses and hexagons), for which $\mathbb{E}[\overline{\mathcal{S}}]$ tends towards the typical value.

The small value of $\mathbb{E}[\overline{\mathcal{S}}]$ for the ADAPT-QAOA solutions raises the question: how far is this solution from the Clifford manifold? To answer this, in Fig. 1(d) we show the optimized parameters $(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$ found for one of the problem instances solved. The $\boldsymbol{\beta}^*$'s are either 0 or $-\pi/4$, indicating the mixer unitaries are Clifford, with mixer Hamiltonians almost always $Y_l Z_m$ for some pair of qubits (l, m), and most of the $\boldsymbol{\gamma}^*$'s are zero with only few, ∼2, being nonzero. Furthermore, the distances of the optimized parameters for $p = 10$ to $\pm s\frac{\pi}{4}$ with $s \in \mathbb{N}$ averaged over all instances with $\alpha \to 1$ ($\omega_{i,j} \in \mathcal{N}(0, 1)$), are $\mathbb{E}[\mathcal{D}(\boldsymbol{\gamma}^*)] = 1.51 \pm 0.72$ and $\mathbb{E}[\mathcal{D}(\boldsymbol{\beta}^*)] = 0.83 \pm 0.96$, indicating that, on average, the solution circuits are farther from the Clifford manifold than in the case of $\omega_{i,j} > 0$. This is to be contrasted with $\mathbb{E}[\mathcal{D}(\boldsymbol{\gamma}^*)] = 2.77 \pm 0.78$ and $\mathbb{E}[\mathcal{D}(\boldsymbol{\beta}^*)] = 2.75 \pm 0.64$ for the optimal parameters of the QAOA solution circuits with $p = 10$. Therefore, the overall structure of the mixer unitaries of the ADAPT-QAOA $U(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$ found for positive $\omega_{i,j}$ is still there when $\omega_{i,j}$ are signed, complemented with a nontrivial non-Clifford action of a few of the cost layers. We have checked that this structure is common to all ADAPT-QAOA solutions reaching $\alpha \to 1$.

We summarize the observations of this section:

(i) The mixer part of all layers is Clifford with parameters either 0 or $-\pi/4$. The mixer Hamiltonian at a given step is of the form $Y_l Z_m$ for some pair of qubits (l, m).

(ii) The cost part of most layers acts trivially with parameters equal to 0.

(iii) Only N steps are required to find an approximated solution. Consequently, only N mixer layers of the form described in the first point are needed.

IV. ADAPT-CLIFFORD APPROXIMATION ALGORITHM FOR MAXCUT

A bit string $\mathbf{z}^*$ is a good approximate solution to MaxCut if $\alpha(\mathbf{z}^*)$ is as close to 1 as possible. Thus, finding good approximate solutions to this problem using only Clifford circuits means to prepare a stabilizer state $|\Psi\rangle$ whose energy expectation satisfies $|\langle\Psi|H_C|\Psi\rangle - E_{\min}^C| \leqslant \epsilon$, with ϵ a small positive constant ideally equal to 0. A measurement in the computational basis then returns $\mathbf{z}^*$ with the desired value of α.

Consider the bit string $\mathbf{z}'$ which maximizes the cost in Eq. (1). A stabilizer state satisfying the conditions discussed above is

$$|\Psi'\rangle = \frac{1}{\sqrt{2}}(|\mathbf{z}'\rangle - |\overline{\mathbf{z}'}\rangle), \tag{15}$$

where $\overline{\mathbf{z}'}$ is the complement of $\mathbf{z}'$, and we have chosen the state to be antisymmetric under the Ising symmetry $[H_C, X^{\otimes N}] = 0$, of the cost Hamiltonian. The state $|\Psi'\rangle$ is completely determined by its N stabilizers. One of them is $-X_1X_2X_3\ldots X_N$, while the remaining $N-1$ ones are of ZZ type and their signs encode the maximal cut of the graph. In this setting, an approximation algorithm based on Clifford circuits must be able to determine an assignment of the signs of the ZZ stabilizers leading to either $\mathbf{z}'$ or a $\mathbf{z}^*$ with $\alpha(\mathbf{z}^*)$ as close to one as possible.

A. Details of the algorithm

We design ADAPT-Clifford so as to exploit the observations summarized at the end of Sec. III to prepare a stabilizer state $|\Psi\rangle$ with the general form given in Eq. (15). This is done in a greedy manner, where after a choice of an initial seed, at every step the best local update is performed. As such, at an intermediate step $0 < r \leqslant N$ we label qubits as active and inactive. A qubit is active if a Pauli gate has been applied to it, otherwise it is inactive, and $a^{(r)} \in \mathbf{a}^{(r)}$ and $b^{(r)} \in \mathbf{b}^{(r)}$ are indices denoting the positions of "active" and "inactive" qubits, respectively, and $\mathbf{a}^{(r)}$ and $\mathbf{b}^{(r)}$ are vectors storing the positions of all the active and inactive qubits at step r.

ADAPT-Clifford prepares $|\Psi\rangle$ starting from the kth qubit and growing this entangled state qubit by qubit, in such a way that at step r the state is a product of two parts: an entangled state of all the $|\mathbf{a}^{(r)}|$ active qubits and all the $|\mathbf{b}^{(r)}|$ inactive qubits in the product state $|+\rangle^{\otimes|\mathbf{b}^{(r)}|}$. To specify the pair $(a^{(r)}, b^{(r)})$ of qubit indices at each step, we use a "gradient" criterion similar to that of ADAPT-QAOA. Specifically, at step $r > 2$ we compute

$$\begin{aligned} g^{(r)}_{a^{(r-1)},b^{(r-1)}} &= -i\langle[H_C, Z_{a^{(r-1)}}Y_{b^{(r-1)}}]\rangle_{r-1} \\ &= -\sum_l \omega_{l,b^{(r-1)}}\langle Z_l X_{b^{(r-1)}} Z_{a^{(r-1)}}\rangle_{r-1}, \end{aligned} \tag{16}$$

where $\langle . \rangle_{r-1} = \langle\psi_{r-1}|.|\psi_{r-1}\rangle$ is taken on the state at step $r-1$. Then we choose the pair of qubits $(a^{(r-1)}, b^{(r-1)})$ that maximizes $g^{(r)}_{a^{(r-1)},b^{(r-1)}}$. The case of $r = 1$ is special, and we discuss it below alongside the steps of the algorithm.

ADAPT-Clifford returns a candidate maximal cut $\mathbf{z}^*$ of a graph $\mathcal{G}$ after completing the following N steps:

(0) At step $r = 0$ we begin by selecting a position k and preparing the product state

$$|\psi_0\rangle = Z_k \mathrm{H}^{\otimes N}|0\rangle^{\otimes N}. \tag{17}$$

At this point the active and inactive qubits are $\mathbf{a}^{(0)} = \{k\}$ and $\mathbf{b}^{(0)} = \{1,\ldots,N\}\backslash\{k\}$.

(1) At step $r = 1$, given that $a^{(0)} = k$ we can estimate the largest gradient analytically. In fact, $\max_{b^{(0)}}[g^{(1)}_{k,b^{(0)}}] = \max_{b^{(0)}}[\omega_{k,b^{(0)}}]$, thus the pair we are looking for is the edge (k, j) of $\mathcal{G}$ with

$$j = \underset{b^{(0)}}{\mathrm{argmax}}[\omega_{k,b^{(0)}}]. \tag{18}$$

After applying the gate $e^{i\frac{\pi}{4}Y_kZ_j}$, the state is

$$|\psi_1\rangle = e^{i\frac{\pi}{4}Z_jY_k}Z_k\mathrm{H}^{\otimes N}|0\rangle^{\otimes N}. \tag{19}$$

The vectors of active and inactive qubits are updated to $\mathbf{a}^{(1)} = \{k, j\}$ and $\mathbf{b}^{(1)} = \{1,\ldots,N\}\backslash\{k, j\}$, respectively.

(2) For $r = 2,\ldots,N-1$, we find the pair of qubits $(\tilde{l}, b^{(r-1)})$, with $\tilde{l} \in \{k, j\}$, which maximizes $g^{(r)}_{\tilde{l},b^{(r-1)}}$, apply the gate $e^{i\frac{\pi}{4}Z_{\tilde{l}}Y_{b^{(r-1)}}}$, and update the vectors of active and inactive qubits. In the case of more than one pair $(\tilde{l}, b^{(r-1)})$ leading to the same largest value of $g^{(r)}_{a^{(r-1)},b^{(r-1)}}$ we break the tie arbitrarily.

(3) After all N steps are completed, we perform a measurement in the computational basis. From the output bit string, $\mathbf{z}_{\mathrm{out}}$, we read out the approximate maximal cut of the graph as $(\mathcal{A}, \overline{\mathcal{A}})$ with $\mathcal{A} = \{z_i \in \mathbf{z}_{\mathrm{out}} | z_i = 0, i = 1,\ldots,N\}$ and $\overline{\mathcal{A}} = \{z_i \in \mathbf{z}_{\mathrm{out}} | z_i = 1, i = 1,\ldots,N\}$.

After the above N steps are completed, the resulting stabilizer state $|\Psi\rangle$ encoding the solution has the form

$$|\Psi\rangle = \left[\prod_{r=2}^{N-1} e^{i\frac{\pi}{4}Z_{\tilde{l}}Y_{b^{(r)}}}\right] e^{i\frac{\pi}{4}Z_jY_k}Z_k\mathrm{H}^{\otimes N}|0\rangle^{\otimes N}. \tag{20}$$

While it may seem that restricting the search to pairs of the form $(\tilde{l}, b^{(r-1)})$ in step 2 may lead to missing the true largest gradient, in Appendix B we show that this is not the case. Furthermore, this restriction has a simple interpretation. After step $r = 1$, we have effectively selected the edge (k, j) as a reference with respect to which we are going to partition the graph. Nodes k and j are thus representatives of the disjoint subsets of the cut. Thus, from that step onward, we can pick $a^{(r-1)} \in \{k, j\}$ without loss of generality in order to decide which qubit to move into the active set, i.e., to include in the entangled state.

Some further comments are in order: (i) Given the type of two-qubit gate we are considering, the form of the initial product state $|\psi_0\rangle$ is chosen as to guarantee that $\max_{b^{(0)}}[g^{(1)}_{k,b^{(0)}}]$ will be positive. (ii) For $r > 1$, and independently of the graph connectivity, not all the terms in the sum in Eq. (16) are nonzero; in fact, the expectation values in $g^{(r)}_{a^{(r-1)},b^{(r-1)}}$ become $\langle Z_lX_{b^{(r-1)}}Z_{a^{(r-1)}}\rangle_{r-1} = \langle Z_lZ_{a^{(r-1)}}\rangle_{r-1}$ and are nonzero only for those values of l for which either $\pm Z_lZ_{a^{(r-1)}}$ is a stabilizer of $|\psi_{r-1}\rangle$. This observation allows us to find the largest gradient without explicitly computing the expectation values in Eq. (16), which we show in Appendix C. At the same time, this observation establishes a direct connection between ADAPT-Clifford and a family of existing MaxCut euristics [87,88], as was recently pointed out in Ref. [89]. (iii) The relevant two-qubit gate can be written in terms of Clifford gates as

$$e^{i\frac{\pi}{4}Y_lZ_m} = \mathrm{S}_l\mathrm{H}_m\mathrm{CNOT}_{l,m}R^{(l)}_x(-\pi/2)\mathrm{CNOT}_{l,m}\mathrm{S}^\dagger_l\mathrm{H}_m, \tag{21}$$

where the S_l, H_l, are the phase and Hadamard gates acting on the lth qubit, $\mathrm{CNOT}_{l,m}$ is the controlled-NOT gate, with qubit l and qubit m as control and target qubits, respectively. Furthermore one can write $R^{(l)}_x(-\pi/2) = \mathrm{H}^{YZ}_l Z_l$ with H^{YZ}_l a variant of the Hadamard gate which swaps the y and z axes. We work through the operations of our algorithm for two small examples in Appendix F.

1. A stabilizer perspective on the algorithm

We can gain further understanding of the inner workings of the algorithm by looking at the way the stabilizers

of the state change from step $r = 0$ to step $r = N - 1$. At step $r = 0$, the product state $|\psi_0\rangle$ has $N - 1$ stabilizers equal to X_l, $l = 1, \ldots, N$, $l \neq k$ and the remaining stabilizer equal to $-X_k$. At step $r = 1$ the action of the gate between qubits (k, j), with j found as described previously, increases the weight of the $-X$ stabilizer by one and changes one of the $+X$ stabilizers by a ZZ stabilizer. The state $|\psi_1\rangle$ is hence stabilized by $-\mathbb{I}_1 \ldots X_k \mathbb{I}_{k+1} \ldots \mathbb{I}_{j-1} X_j \ldots \mathbb{I}_N$ and $-\mathbb{I}_1 \ldots Z_k \mathbb{I}_{k+1} \ldots \mathbb{I}_{j-1} Z_j \ldots \mathbb{I}_N$ while the remaining $N - 2$ are still X_l with $l \neq k, j$. This process continues until $r = N - 1$; with every new gate the weight of the $-X$ stabilizer increases by one, and one of the $+X$ stabilizers gets replaced by a ZZ stabilizer. We see then that the Clifford gate $e^{i\frac{\pi}{4} Y_l Z_m}$ was not chosen arbitrarily. In fact, ADAPT-QAOA finds it because it is the gate that maps X_l to $Z_l Z_m$.

As such we can phrase goal of the algorithm to be the correct assignment of the signs of the ZZ stabilizers. After all $N - 1$ steps are completed, the state $|\psi_{N-1}\rangle$ has one stabilizer equal to $-X_1 X_2 X_3 \ldots X_N$ and the remaining $N - 1$ stabilizers are ZZ with signs that were determined in the previous steps. If this sign assignment is done correctly, it encodes the approximate maximal cut produced by the algorithm. One can read it out directly by setting the value of any spin to either $+1$ or -1 arbitrarily and use the measured signs of the ZZ stabilizers to fix the values of the spins at the other $N - 1$ positions relative to the first one.

B. Runtime and space complexities

Although finding the qubit with the largest gradient at a given step does not require the explicit computation of expectation values of Pauli strings (see Appendix C for details), it is defined based on a double sum with indices running on portions of the vertex set. As such, this part of the algorithm incurs the leading runtime cost.

At step $r > 1$ and before applying the two-qubit gate, there are $r - 1$ active qubits and $N - r + 1$ inactive qubits. In order to decide on which pair of qubits we act the gate, we compute Eq. (16) via Eq. (C3) for all pairs $(\tilde{l}, b^{(r-1)})$ where $\tilde{l} \in \{k, j\}$ and $b^{(r-1)} \in \mathbf{b}^{(r-1)}$. There are $2(N - r + 1)$ of those pairs. For a given pair the sum in Eq. (16) is $\forall_l$ such that $(l, \tilde{l}) \in \mathcal{E}$. However only when $l \in \mathbf{a}^{(r-1)}$ is the expectation value $\langle Z_l X_b^{(r-1)} Z_{\tilde{l}} \rangle_{r-1}$ nonzero. Hence, at step r, there are $\delta = \min(r - 1, K)$ with K the maximum degree of the graph, nonzero terms in the sum. For bounded-degree graphs, such as K-regular graphs, $\delta = O(K)$ at most, whereas for dense graphs with $K = O(N)$, $\delta = O(N)$.

For a fixed initial position k, the algorithm executes $N - 1$ steps before reaching a candidate solution. The total number of nonzero terms involved in the computation of the largest gradients is $\sum_{r=2}^{N-1} 2(N - r + 1)\delta$, so we have

$$2K \sum_{r=2}^{N-1} (N - r + 1) = K(N^2 - N + 2)$$

or

$$2 \sum_{r=2}^{N-1} (N - r + 1)(r - 1) = \frac{2}{3}(N^3 - 7N + 6)$$

for bounded-degree and dense graphs, respectively. Leading to a run time complexity of $O(N^2)$ for bounded degree graphs and $O(N^3)$ for dense graphs.

Since in general the initial position k leading to the best approximate solution is not known, we propose and explore two complementary approaches. In the first approach, we choose the initial position at random. This algorithm, to which we refer as randomized ADAPT-Clifford, leads to run time complexities of $O(N^2)$ and $O(N^3)$ for bounded-degree and dense graphs, respectively, as described above. Second, we introduce a deterministic version—deterministic ADAPT-Clifford—where the best initial position k^* is determined by exhaustive search. That is, we run ADAPT-Clifford N times, each with a different initial position k, and return the cut of minimal energy found. The runtime complexity of this deterministic approach is thus $O(N^3)$ and $O(N^4)$ for bounded-degree and dense graphs, respectively. Naturally, the deterministic approach is guaranteed to return solutions of equal or smaller energy expectation that the randomized approach, at the cost of a more limiting runtime. Whether there exist graph families for which any initial position is as good as any other is a question for future work. Finally, it is easy to see that for both randomized and deterministic approaches the space complexity of the algorithm is $O(N^2)$, corresponding to the memory required to store the Tableau.

V. ALGORITHM PERFORMANCE ON WEIGHTED COMPLETE GRAPHS

We have implemented the ADAPT-Clifford algorithm using the fast stabilizer circuit simulator Stim [23]. Our implementation is available at [90]. Although we have chosen this simulator to implement our algorithm, any stabilizer circuit simulator which supports interactivity, that is, where expectation values of Pauli strings can be computed and the circuit modified according to the results, could be used to implement the algorithm.

We follow the presentation of Sec. III and discuss separately our algorithm's performance for MaxCut on weighted complete graphs with positive and signed weights. For the latter case, we will focus on the Sherrington-Kirkpatrick model.

A. The case of positive weights

The results of Sec. III indicate that the precise choice of positive weight distribution may be immaterial. We have verified numerically that this is indeed the case for a few different weight distributions. In this subsection, we focus the discussion to $\omega_{i,j}$ sampled from U[0, 1] and leave an exhaustive investigation for future work.

We begin studying the performance of the randomized approach. We draw a parallel between the random initialization of ADAPT-Clifford and the rounding step of GW, and thus assess the performance of the randomized ADAPT-Clifford by direct comparison with GW. We solved 100 different problem instances for graph sizes up to $N = 1000$ with both algorithms. In Fig. 2(a) we show the normalized mean minimum energy, $\mathbb{E}[E_{\min}]/N$, of the solutions obtained with randomized ADAPT-Clifford (green circles) and the ones obtained with GW (light gray circles). Notice that our randomized

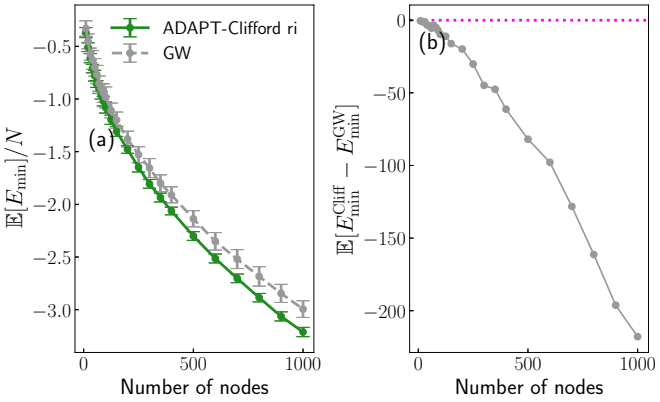

FIG. 2. Performance of randomized ADAPT-Clifford on weighted complete graphs. (a) Normalized energy found by randomized ADAPT-Clifford (green circles) and GW (light gray circles) averaged over 100 instances. (b) Instance-averaged minimum energy difference between the solution found with randomized ADAPT-Clifford and GW as a function of problem size. Notice that randomized ADAPT-Clifford is almost always superior to GW. The magenta dotted line indicates a mean energy difference of zero. We have omitted the error bars for the sake of clarity.

ADAPT-Clifford almost always produces a solution of lower energy expectation than GW. These observations can be further verified with the mean difference of the minimum energy found, $\mathbb{E}[E_{\min}^{\text{Cliff}} - E_{\min}^{\text{GW}}]$, which we show in Fig. 2(b). Since our randomized ADAPT-Clifford consistently beats GW, we expect it to have a performance guarantee for typical instances of positively weighted complete graphs above that of GW for the general problem. We discuss the methodology used to estimate it in Appendix G. We find $\overline{\alpha}^{\text{r}} \approx 0.8986$ a value which confirms our intuition and sets a lower bound for the expected performance of the deterministic ADAPT-Clifford.

We now focus on the deterministic approach. First, we benchmark this algorithm for instances with size up to $N = 30$ for which the exact solution can be found exhaustively. Figure 3(a) shows the exact approximation ratios α, obtained for 100 problem instances. For these small problems, our algorithm performs, on average, above $\alpha = 0.997$, with the value of the minimum α increasing as $N \to 30$. We notice that the number of instances for which our algorithm finds the exact ground state slightly decreases with the problem size. The success rate, defined as the number of times the algorithm finds a cut with energy $E_{\min}^{\text{Cliff}} - E_{\min}^{\text{C}} < 10^{-10}$, is shown in Fig. 3(b) as a function of the problem size. We observe a success rate ~80% for $N = 30$.

For problem sizes beyond $N = 30$, when we cannot access the exact value of the ground-state energy, we resort to a direct comparison with the GW algorithm. We find that the cuts obtained with deterministic ADAPT-Clifford are of superior quality to those found with the standard GW algorithm. To obtain a comparison, we thus systematically increase the number of times $\mathcal{I}$ the rounding step is performed in GW and return the best cut found; see Appendix A for details. The standard GW algorithm thus corresponds to $\mathcal{I} = 1$. In Fig. 3(c) we show the normalized mean energies $\mathbb{E}[E_{\min}]/N$ for 60 problem instances up to a problem size of $N = 200$ produced by our algorithm (orange circles), standard GW

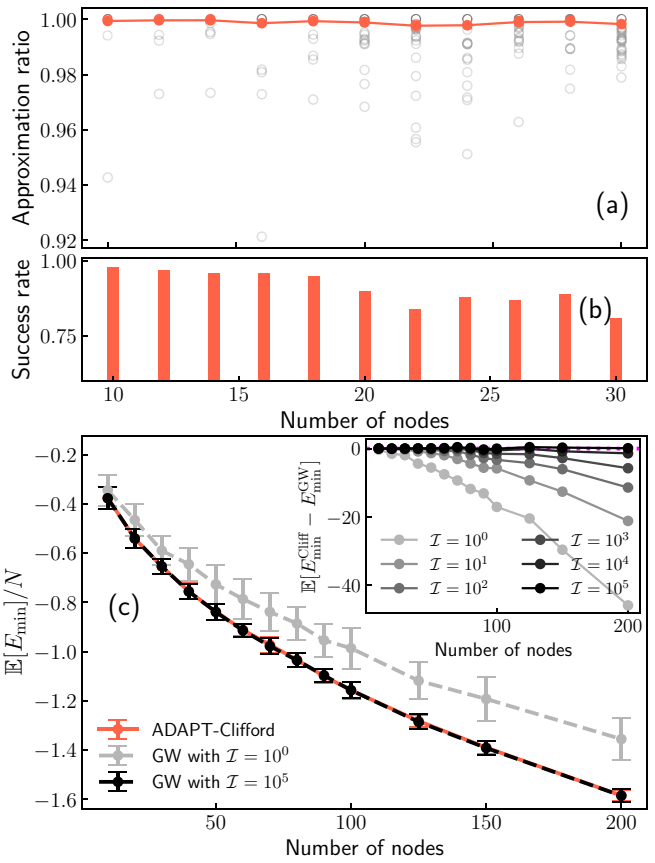

FIG. 3. (a) Approximation rations α (empty circles) and mean approximation ratios (orange full circles) of solutions to MaxCut on 100 weighted complete graphs per graph size found with deterministic ADAPT-Clifford. (b) Success rate on the 100 problem instances per graph size considered in (a). (c) Instance-averaged minimum energy over 60 problems up to graphs with 200 nodes, both with our algorithm (orange solid line), with Goemans-Williamson algorithm (light gray dashed line) and Goemans-Williamson with $\mathcal{I} = 10^5$ (black dashed line). The inset shows the mean difference in the minimum energies found by our algorithm and the GW algorithm as a function of the problem size and for different values of $\mathcal{I}$. The magenta dotted line indicates a mean energy difference of zero. In the inset we have omitted the error bars for the sake of clarity.

(light gray circles), and GW with $\mathcal{I} = 10^5$ (black circles). Notice that our algorithm produces cuts which are *always*, not merely on average, better than those produced with standard GW, and only when we reach $\mathcal{I} = 10^5$ does the GW algorithm begin to produce a cut whose quality is, on average, superior to that of the cut produced by our algorithm.

To further verify this observation, the inset of Fig. 3(c) shows the mean energy difference, $\mathbb{E}[E_{\min}^{\text{Cliff}} - E_{\min}^{\text{GW}}]$, between the solution found with our algorithm and the one found with GW, with the magenta dotted line indicating $\mathbb{E}[E_{\min}^{\text{Cliff}} - E_{\min}^{\text{GW}}] = 0$, that is, equal quality cuts on average. It is seen that ADAPT-Clifford performs increasingly better than GW with fixed $\mathcal{I}$ as problem size is increased. To quantify the approximation quality of ADAPT-Clifford, we estimate the average approximation ratio of the deterministic ADAPT-Clifford on this family of graphs to be $\overline{\alpha} = 0.9686$; see Appendix G for details.

To complement our benchmarks we performed a time to solution (TTS) study of GW with variable $\mathcal{I}$, randomized ADAPT-Clifford, and deterministic ADAPT-Clifford, the details are shown in Appendix D. While Fig. 3(c) shows that $\mathcal{I} = 10^5$ rounding steps are needed for the GW algorithm to match the approximation quality of the deterministic ADAPT-Clifford for problem sizes up to $N = 200$, the data in the inset imply that $\mathcal{I}$ may in fact need to scale with N for the GW algorithm to compete with ADAPT-Clifford. However, in Appendix D we show that the TTS of GW remains constant with I up to $I = 10^3$ and only slightly increased for larger values of $\mathcal{I}$, and even at $\mathcal{I} = 10^5$ the TTS of GW is faster than deterministic ADAPT-Clifford. In fact, we empirically verify the $O(N^3)$ runtime scaling of GW (see Appendix D and [24,91] and references therein).

However, we also find an empiric runtime scaling for randomized ADAPT-Clifford of $O(N^{2.7})$, indicating an advantage. The comparison here is more involved, as GW with $\mathcal{I} > 10^1$ already produces a better solution than randomized ADAPT-Clifford. However, the Cholesky decomposition performed as part of GW is usually executed as a multicore operation in most numerical linear algebra libraries. On the contrary, our implementation of ADAPT-Clifford runs on a single core, thus one could then easily improve the quality of solution without sacrificing the TTS or runtime scaling by executing the algorithm for different initial position in parallel. We thus believe this variant of the algorithm does offer an advantage over GW.

B. Signed weights: The Sherrington-Kirkpatrick model

The Sherrington-Kirkpatrick (SK) model [92] has played a fundamental role in the advancement of the understanding of the physics of spin glasses and disordered systems [93–96]. It describes N classical spins with all-to-all couplings of both ferromagnetic and antiferromagnetic character. The Hamiltonian is given by

$$H_{\rm sk} = \frac{1}{\sqrt{N}} \sum_{i<j} \omega_{i,j}\sigma_i\sigma_j, \tag{22}$$

where $\sigma_i \in \{-1, 1\}$ is a classical spin and the couplings $\omega_{i,j}$ are sampled from a distribution with zero mean and unit variance, for instance, the normal distribution $\mathcal{N}(0, 1)$. A milestone result by Parisi [97,98] gave an explicit expression for the ground-state energy density of this model in the thermodynamic limit, which we refer to as the Parisi value,

$$\lim_{N\to\infty} \mathbb{E}\left[\frac{E^{\rm sk}_{\rm min}}{N}\right] = \Pi^* = -0.763166\ldots, \tag{23}$$

where the expectation value is over realizations of the random couplings, and $E^{\rm sk}_{\rm min}$ refers to the ground-state energy of Hamiltonian in Eq. (22). The most accurate numerical value of Eq. (23) to date was computed in Ref. [99]. The limit in the left-hand side of Eq. (23) has been formally shown to both exist and be equal to the Parisi value [100,101].

Recently the SK model has been used as a benchmark in the study of quantum approximate optimization algorithms [9,102,103]. Motivated by these works, we focus our attention on this model to characterize the performance of our algorithm on complete graphs with signed weights. A word of caution: The ADAPT-QAOA solution circuits for the signed case, including small instances of the SK model, are not completely Clifford; see Figs. 1(b) and 1(d) and Sec. III. As such, we do not expect our algorithm to match the solution quality of the best classical algorithm due to Montanari [104,105], which produces a $\boldsymbol{\sigma}^*$ with energy below $(1-\epsilon)$ times the lowest energy for typical instances, with ϵ a small positive constant [106]. Nevertheless, we are interested in seeing how close the $\boldsymbol{\sigma}^*$'s produced by our algorithm get to the Parisi value, for both the randomized and deterministic variants of ADAPT-Clifford.

In order to utilize ADAPT-Clifford we promote the classical spin in Eq. (22) to $\sigma_i \to Z_i$ and use the resulting Hamiltonian as our cost. Following the presentation of the previous subsection, we discuss first the performance of the randomized ADAPT-Clifford. The green circles in the inset of Fig. 4(c) show $\mathbb{E}[\frac{E_{\rm min}}{N}]$ for this algorithm with problem sizes up to $N = 1000$. To obtain its value in the thermodynamic limit we fit the data for $N \in [40, 1000]$ to a model of the form $qN^{-2/3} + \Pi^{\rm Cliff}_{\rm ri}$ [107] where $\Pi^{\rm Cliff}_{\rm ri}$ corresponds to the mean energy density in the thermodynamic limit obtained with the randomized ADAPT-Clifford. We find $\Pi^{\rm Cliff}_{\rm ri} \approx -0.682$, which corresponds to ∼89% of the Parisi value [black star in inset of Fig. 4(c)]. This value is below what is obtained with convex relaxation methods, for instance, semidefinite programming, which is known to give $\mathbb{E}[\frac{E_{\rm min}}{N}] = -\frac{2}{\pi} + o(1) \approx -0.6366$ with $o(1)$, a number which vanishes for $N \to \infty$ [108,109]. For comparison we display $\mathbb{E}[\frac{E_{\rm min}}{N}] = -\frac{2}{\pi}$ as the horizontal dotted line both in the inset and in Fig. 4(c).

Let us now consider the deterministic ADAPT-Clifford. For small problems $N \in [10, 30]$ we computed the exact approximation ratios α over 100 problem instances, and show them as empty circles in Fig. 4(a). Notably we do not observe $\alpha < 0.94$ for any instance, and the average over instances is always above $\alpha > 0.997$. To complement this observation we compute the success rate, defined as the number of instances for which the difference $E^{\rm SK}_{\rm min} - E^{\rm Cliff}_{\rm min} < 10^{-10}$. These are shown in Fig. 4(b) with the smallest one being ∼82% at $N = 30$.

To fully explore the performance of the deterministic ADAPT-Clifford algorithm, we solve 100 instances for problems up to $N = 200$. The normalized energies $E^{\rm Cliff}_{\rm min}/N$ are shown as empty circles in Fig. 4(c) for all the instances considered, the red full circles show the respective $\mathbb{E}[E^{\rm Cliff}_{\rm min}]/N$, and the error bars correspond to the standard deviation of the normalized energies. To assess the quality of the solutions found we consider the data in the interval $N \in [40, 200]$ and fit it to a model of the form $qN^{-2/3} + \Pi^{\rm Cliff}$ with $\Pi^{\rm Cliff}$ the estimated mean energy density in the thermodynamic limit of the solutions found by our algorithm. In particular for $N = 200$ we find $\mathbb{E}[\frac{E^{\rm Cliff}_{\rm min}}{N}] \approx -0.727\ldots$ and from the linear fit we find $\Pi^{\rm Cliff} \approx -0.7409\ldots$, shown by a red star in Fig. 4(c). These values correspond to ∼94% and ∼97% of the Parisi value, respectively (the latter is shown with a black star in Fig. 4 c). These values are below what is obtained with convex relaxation methods [horizontal dotted line in Fig. 4(c)]. Notably, the value reached by our algorithm for $N = 200$ is already better than what can be obtained with zero-temperature simulated annealing which gives $\mathbb{E}[\frac{E_{\rm min}}{N}] \sim -0.71$

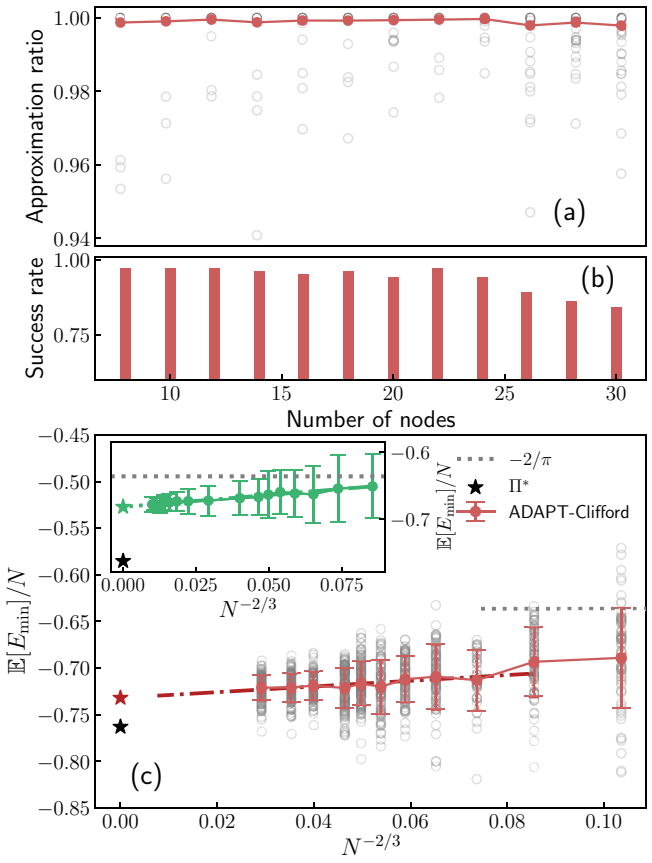

FIG. 4. (a) Approximation rations α (empty circles) and mean approximation ratios (red full circles) of deterministic ADAPT-Clifford for 100 different disorder realizations of the SK model per system size. (b) Success rate on the 100 problem instances per graph size considered in (a). (c) Ground-state energy density for each of the 100 problem instances (empty circles) per problem size up to $N = 200$, and their mean (full circles). The dashed-dotted line shows the best linear fit and the red star the respective mean energy density in the thermodynamic limit. The inset shows the average ground-state energy density up to $N = 1000$ as obtained with randomized ADAPT-Clifford, with its corresponding linear fit (see main text) and the value in the thermodynamic limit (green star). The gray dotted line shows the mean energy density obtained with semidefinite programing, and the black star shows the Parisi value Π^*.

(as quoted in Ref. [102]), and Π^{Cliff} is comparable to what is achievable with simulated annealing on large problem instances.

VI. ALGORITHM PERFORMANCE ON OTHER FAMILIES OF GRAPHS

In this section, we characterize the performance of ADAPT-Clifford in both its variants for the MaxCut problem on K-regular graphs (unweighetd and weighted) and unweighted Erdős-Rényi graphs with various edge probabilities. For the randomized ADAPT-Clifford, we directly compare the quality of the cuts found with standard GW, while for determinisitic ADAPT-Clifford we discuss the exact approximation ratios for small problems and compare against GW with variable $\mathcal{I}$, the number of time the rounding step is performed.

A. Performance on K-regular graphs

We consider 3-regular and 8-regular graphs, unweighted and weighted. In all cases edge weights are sampled from U[0, 1].

In Figs. 5(a)–5(c) we show the normalized instance-averaged minimum energy of the solutions found with randomized ADAPT-Clifford and standard GW. For large ($N > 200$) unweighted 3-regular graphs, GW finds better solutions, on average, than randomized ADAPT-Clifford; see Fig. 5(e). The situation is markedly reversed with the inclusion of nontrivial edge weights, with randomized ADAPT-Clifford outperforming standard GW [see Fig. 5(b)], and the performance margin widens with increased connectivity [see Fig. 5(c)]. These observations are verified with the averaged minimum energy differences shown in Figs. 5(f) and 5(g) for weighted 3- and 8-regular graphs, respectively. Thus, GW performs better than randomized ADAPT-Clifford only for unweighted 3-regular graphs, while the comparative performance of our algorithm consistently improves with both the inclusion of edge weights and higher connectivity.

We now move to the performance of deterministic ADAPT-Clifford. In Fig. 6(a) we show the mean approximation ratios over 100 problem instances for each of these types of graphs. For the unweighted 3-regular graphs we consider problem sizes $N \in [10, 28]$ and for the weighted problems we consider problem sizes $N \in [12, 28]$. We have omitted the error bars from the figure for the sake of clarity. Deterministic ADAPT-Clifford shows the poorest performance for unweighted 3-regular graphs, circles in Fig. 6(a), with a decreasing mean α as N increases. Interestingly the comparative performance of ADAPT-Clifford improves upon inclusion of edge weights, diamonds in Fig. 6(a), with a mean α above 0.995 for all problem sizes considered. Further improved performance is observed with higher edge connectivity, as evidence by the mean approximation ratio for weighted 8-regular graphs [squares in Fig. 6(a)]. In Fig. 6(b) we show the success rate of the algorithm, i.e., the number of times ADAPT-Clifford found the maximal cut. The 3-regular graphs (unweighted and weighted) show a success rate which consistently decay with problem size. On the contrary for weighted 8-regular graphs our algorithm shows a success rate above 90% up to $N = 28$.

For larger problem sizes, we compare the solution quality of deterministic ADAPT-Clifford with that of GW with variable $\mathcal{I}$ ($\mathcal{I} = 1$ is the standard GW algorithm). Figures 7(a)–7(c) show the normalized mean minimum energy $\mathbb{E}[E_{\min}]/N$ for the K-regular graphs studied. Notably, deterministic ADAPT-Clifford produces solutions of lower energy than GW for all three graph ensembles under consideration; compare the colorful markers with the light gray markers in Figs. 7(a)–7(c). For the unweighted 3-regular graphs, Fig. 7(a), already at $\mathcal{I} = 10$ GW consistently finds a cut of lower energy than deterministic ADAPT-Clifford, signaling a reduced performance of the latter method compared to the case of weighted complete graphs. This observation can be

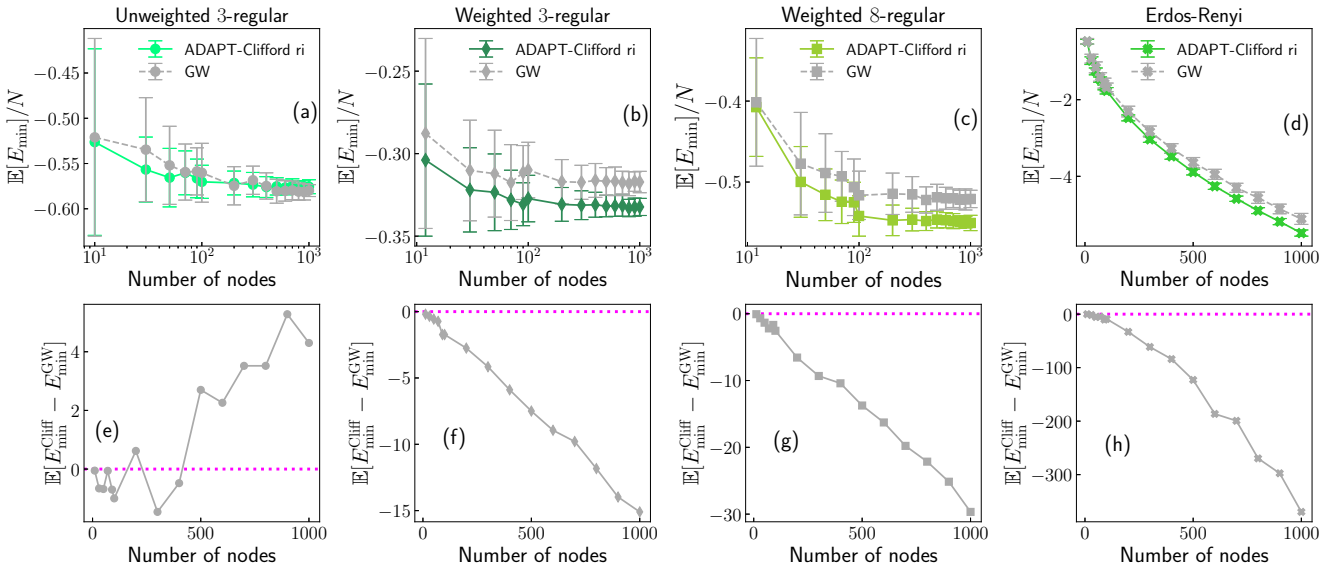

FIG. 5. Performance of randomized ADAPT-Clifford vs GW. (a)–(d) Normalized instance-averaged minimum energy found with randomized ADAPT-Clifford (colorful markers and solid lines) and standard GW (light gray markers and dashed lines). The different graph types studied are (a) unweighted 3-regular graphs, (b) weighted 3-regular graphs, (c) weighted 8-regular graphs, and (d) unweighted Erdős-Rényi graphs with edge probability 1/2. For the weighted case we take $\omega_{i,j}$ in U[0, 1]. (e)–(h) Instance-averaged minimum energy differences between the solutions found with ADAPT-Clifford and standard GW for (e) unweighted 3-regular (circles), (f) weighted 3-regular (diamonds), (g) weighted 8-regular (squares), and (h) Erdős-Rényi with edge probability 1/2 (x's). The magenta dotted line indicates equal energy of the solutions found on average. We have omitted the error bars to avoid saturating the figure. All averages were computed over 100 randomly generated instances.

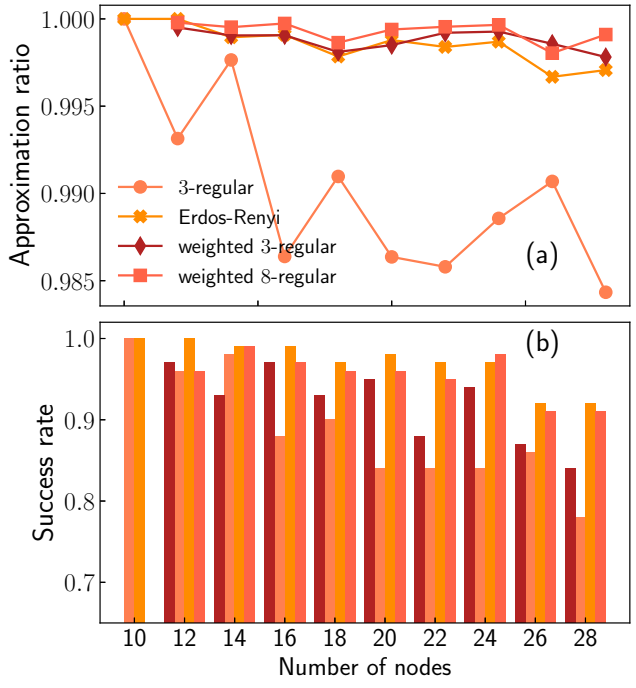

FIG. 6. (a) Instance-averaged approximation ratios up to $N = 28$ for 100 different instances of: unweighted 3-regular graphs (circles), weighted 3- (diamonds) and 8-regular graphs (squares) with $\omega_{i,j} \in$ U[0, 1], and Erdős-Rényi graphs with edge probability 1/2 (x's). We have omitted the error bars for clarity. (b) Success rate for each of the problem sizes and graph types considered in (a).

further verified with the mean difference of the minimum energy found, $\mathbb{E}[E_{\min}^{\text{Cliff}} - E_{\min}^{\text{GW}}]$, shown in Fig. 7(e).

Similarly to its randomized counterpart, deterministic ADAPT-Clifford performs more competitively when edge weights are included. In Fig. 7(b) we show the $\mathbb{E}[E_{\min}]/N$ obtained with our algorithm (red diamonds), GW (light gray diamonds), and GW with $\mathcal{I} = 10^3$ (black diamonds), for the weighted 3-regular graphs. Further inspection of the corresponding $\mathbb{E}[E_{\min}^{\text{Cliff}} - E_{\min}^{\text{GW}}]$, shown in Fig. 7(f), shows that at least $\mathcal{I} = 10^2$ are necessary for the GW solution to be, on average, superior to that found by our algorithm. The performance margin widens as we move to regular graphs with higher connectivity. Figure 7(c) shows $\mathbb{E}[E_{\min}]/N$ obtained with deterministic ADAPT-Clifford (orange squares), standard GW (light gray squares), and GW with $\mathcal{I} = 10^3$ (black squares), for weighted 8-regular graphs. After inspecting the $\mathbb{E}[E_{\min}^{\text{Cliff}} - E_{\min}^{\text{GW}}]$ in Fig. 7(g) we observe that $\mathcal{I} = 10^4$ is necessary for the GW solution to be consistently better than the deterministic ADAPT-Clifford solution. Thus, for sparse graphs the performance of both randomized and deterministic ADAPT-Clifford improves with the inclusion of edge weights and/or higher node connectivity.

B. Performance on unweighted Erdős-Rényi graphs

We now wish to characterize the performance of ADAPT-Clifford for MaxCut on dense graphs with variable density. For this task we will focus on unweighted Erdős-Rényi graphs.

First, we fixed the edge probability to 1/2 and study the performance with respect to the problem size. In Fig. 5(d)

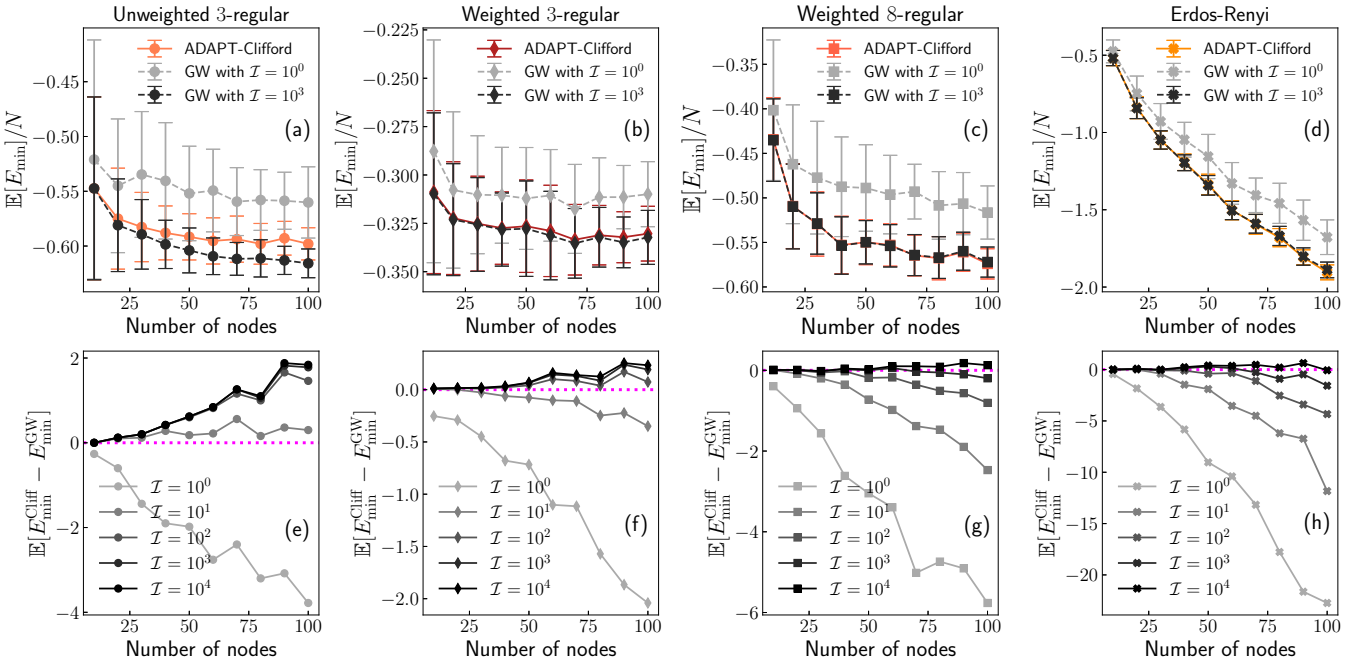

FIG. 7. (a)–(d) Normalized instance-averaged minimum energy found over 100 instances for problem sizes up to $N = 100$ and different graph types obtained with deterministic ADAPT-Clifford (colored markers and solid lines), with standard GW (light gray markers and dashed lines), and GW with $\mathcal{I} = 10^3$ (dark gray markers and dashed lines). The different graph types studied are (a) unweighted 3-regular graphs, (b) weighted 3-regular graphs, (c) weighted 8-regular graphs, and (d) unweighted Erdős-Rényi graphs with edge probability 1/2. For the weighted case we always take $\omega_{i,j}$ in U[0, 1]. (e)–(h) Instance-averaged minimum energy differences between the solutions found with our algorithm and the solution found with GW with different values of $\mathcal{I}$. For the graph types: (e) unweighted 3-regular (circles), (f) weighted 3-regular (diamonds), (g) weighted 8-regular (squares), and (h) Erdős-Rényi with edge probability 1/2 (x's). As a reference the the magenta dotted line indicates equal energy of the solutions found on average. We have omitted the error bars for clarity.

we show the instance-averaged minimum energy of solutions obtained with randomized ADAPT-Clifford (green) and standard GW (light gray). For graphs up to $N = 1000$. The randomized version of our algorithm produces better solutions, on average, than GW, an observation that is verified by the instance-averaged minimum energy differences shown in Fig. 5(h).

Next, we analyze the performance of the deterministic ADAPT-Clifford on small problems $N \leqslant 28$. Figures 6(a) and 6(b) show mean approximation ratios x's), which are above $\alpha \sim 0.997$, and success rates, respectively. Notably, deterministic ADAPT-Clifford shows a higher success rate for this family of graphs, finding the maximal cut on all instances considered for the sizes $N = 10, 12$ (whereas it only achieves the same for the 19 nonisomorphic 3-regular graphs at $N = 10$). For larger problems, in Fig. 7(d) we compare the normalized instance-averaged minimum energy found by our algorithm (orange solid line), standard GW (light gray dashed line), and GW with $\mathcal{I} = 10^3$ (black dashed line). Our algorithm (orange) finds a solution of lower energy, on average, than that found with GW (light gray). We explore the extent of this advantage by inspecting the mean difference of the minimum energy found, $\mathbb{E}[E^{\text{Cliff}}_{\min} - E^{\text{GW}}_{\min}]$, as a function of N and with $\mathcal{I}$ as a control parameter. The results are shown in Fig. 7(h). It is seen that only at $\mathcal{I} = 10^4$ are the GW solutions consistently of lower energy than those found by deterministic ADAPT-Clifford.

Now we turn our attention to benchmarking both the randomized and deterministic ADAPT-Clifford on Erdős-Rényi graphs with varying edge inclusion probability. We focus on problems with $N = 120$ and consider edge probabilities in $[0.1, 0.9]$. We solve 100 problem instances of MaxCut per edge inclusion probability. In Fig. 8(a) we show the normalized mean energies found with the randomized ADAPT-Clifford (green) and with standard GW (light gray). Our randomized algorithm returns, on average, a cut of better quality than GW [see also instance-averaged minimum energy differences in Fig. 8(b)]. The normalized mean energies of the solutions found with deterministic ADAPT-Clifford (orange) are shown in Fig. 8(c), alongside those for standard GW (light gray), and GW with $\mathcal{I} = 10^4$ (black). With the exception of edge probabilities smaller than 0.15 and larger than 0.85, the solutions found by our algorithm are *always*, not merely on average, better than the ones found with GW, with the largest advantage observed for edge probabilities around 1/2. Only at $\mathcal{I} \sim 10^4$ does GW produce solutions on average comparable to those found by ADAPT-Clifford. This is seen more clearly in the instance-averaged energy difference of the solutions found $\mathbb{E}[E^{\text{Cliff}}_{\min} - E^{\text{GW}}_{\min}]$, shown in Fig. 8(b). Only at $\mathcal{I} = 10^4$ do we find $\mathbb{E}[E^{\text{Cliff}}_{\min} - E^{\text{GW}}_{\min}] \in (0, 0.5]$ for all edge probabilities, indicating our algorithm no longer offers an advantage over GW.

The results discussed here suggest that ADAPT-Clifford offers an advantage over GW on the quality of the cut found

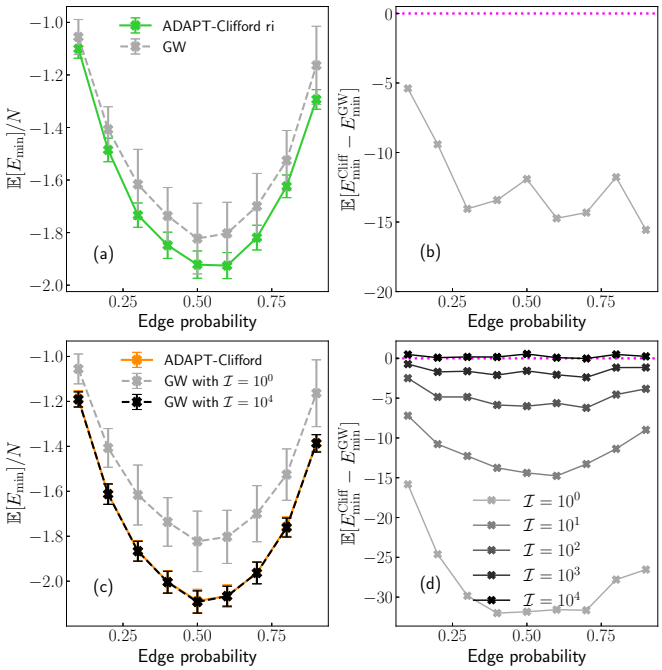

FIG. 8. (a) Normalized instance-averaged minimum energy of the solutions found with randomized ADAPT-Clifford (green) and GW (light gray). (b) Instance-averaged minimum energy differences between the solutions found with our algorithm and the solution found with GW. (c) Normalized instance-averaged minimum energy of the solutions found with the deterministic ADAPT-Clifford (orange), standard GW (light gray), and GW with $\mathcal{I} = 10^4$ (black), as a function of the edge probability which we take in $[0.1, 0.9]$. (d) Instance-averaged minimum energy differences between the solutions found with our algorithm and the solution found with GW with different values of $\mathcal{I}$, as a function of edge probability. The magenta dotted line indicates mean energy difference of zero. The averages are taken over 100 different problem instances and for $N = 120$.

for dense graphs, with the largest gap for graphs with density $\sim 1/2$.

VII. DISCUSSION AND OUTLOOK

We introduce ADAPT-Clifford, a quantum-inspired *classical* approximation algorithm for MaxCut. For each problem instance, ADAPT-Clifford builds a low-depth Clifford circuit to prepare a stabilizer state that encodes an approximate solution. The algorithm was inspired by observation of the (almost) Clifford character of the ADAPT-QAOA solution circuits for MaxCut on weighted fully connected graphs. A comparison between the two methods and resource estimates for ADAPT-Clifford is given in Appendix E. We introduce a randomized and a deterministic variant of this algorithm. Their respective runtime complexities are $O(N^2)$ and $O(N^3)$ for sparse graphs, and $O(N^3)$ and $O(N^4)$ for dense graphs, and in all cases the space complexity is $O(N^2)$. Naturally, the deterministic variant *always* outperforms the randomized variant, albeit at the cost of an increased runtime.

We have studied the performance of ADAPT-Clifford on MaxCut for various families of graphs, both dense and sparse, and both unweighted and weighted. On weighted complete graphs with positive weights, ADAPT-Clifford finds very high quality cuts, reaching the absolute maximum in the majority of small instances. Moreover, the algorithm is scalable, allowing us to easily find solutions for instances with up to 1000 nodes. ADAPT-Clifford also performs well for signed weights, finding good approximations to the ground state of the SK model with an energy that extrapolates to 97% of the Parisi value in the thermodynamic limit. To investigate performance as a function of density, we applied ADAPT-Clifford to MaxCut on unweighted Erdős-Rény graphs with variable edge inclusion probability. We again find that ADAPT-Clifford finds the absolute maximum cut for the majority of small instances and easily scales to hundreds of nodes. Finally, we study the performance of ADAPT-Clifford for sparse graphs. Even though these graphs are far from the context that gave rise to the algorithm, we find that ADAPT-Clifford still performs well, producing the absolute maximum cut with high probability for small instances and easily scaling to 1000 nodes. Only for the case of 3-regular graphs, the sparsest category of graphs we studied, do we observe a noticeable deterioration in solution quality with increasing size. Counter-intuitively, performance improves somewhat with the inclusion of edge weights.

To assess the performance of ADAPT-Clifford for large problem instances whose exact solution is intractable, we compare its performance with the GW algorithm, which represents the state of the art in approximate solution of MaxCut. For all graph families studied, ADAPT-Clifford outperforms the standard GW algorithm in the quality of the cut found. Only for very sparse unweighted graphs, such as 3-regular graphs, does the performance of the GW algorithm become comparable to that of ADAPT-Clifford, but even in this case the inclusion of edge weights favors the latter. Finally, ADAPT-Clifford solves problems to which the GW algorithm is not directly applicable, as exemplified by our results on the SK model.

The Clifford or near Clifford character of the ADAPT-QAOA solution circuits is a key observation which was missed in previous work [61]. This observation, as laid out in Sec. III, allowed us to devise a quantum-inspired, polynomial-time approximation algorithm for MaxCut. While it is known that MaxCut on dense graphs admits polynomial time approximation schemes (PTASs), leading to approximated solutions which are $1 - \epsilon$ away from the optimum in time polynomial in N [44,45], the scaling of the runtime as a function of ϵ may render these algorithms impractical. In contrast, in this work we showed empirically that ADAPT-Clifford performs better than an algorithm that offers a guaranteed approximation ratio. Notably, based on the gradient criteria used as update rule in ADAPT-Clifford a connection between this algorithm and a family of heuristics for the MaxCut problem, known as Sahni-Gonzales algorithms [87,88], can be established, as was recently pointed out in Ref. [89]. It is remarkable to see that when ADAPT-QAOA performs best, the adaptive approach builds solution circuits which share this property with well-known classical heuristics.

We hope the results reported here will help delimit the subset of graphs where a quantum speedup could be expected and thus where the current efforts should focus, in similar spirit to previous results obtained with a different subuniversal family

of gates [110]. While our work indicates that ADAPT-Clifford has a guaranteed approximation ratio, we do not yet have a proof. Our algorithm showed the poorest performance on fully connected graphs with signed weights. This was anticipated in Sec. III since the ADAPT-QAOA solution circuits are not fully Clifford. However, they are *near*-Clifford, motivating then a resource-centered design of variational ansätze, with a Clifford mixer part constructed following a scheme like the one introduced in this work, similar in spirit to the optimal mixers restricted to subspaces [111], and a cost part with *few* variational parameters adding just the right amount of non-Cliffordness necessary to approximate the problem up to a desired ratio. Furthermore our algorithm could aid in reducing the cost of parameter optimization in QAOA when used to warm start [112] it. More broadly, Clifford circuits can be leverage to construct a framework for the efficient state initialization in variational quantum algorithms beyond the product state paradigm. An example of this applied to quantum chemistry problems was introduced in Ref. [113]. Finally, our Clifford algorithm was tailored to solve the Max-Cut problem. It remains an open question to what extent other combinatorial optimization problems admit Clifford approximation algorithms with practical polynomial runtimes.

ACKNOWLEDGMENTS

The authors are grateful to Othmane Benhayoune-Khadraoui for helpful discussions, to Pablo Poggi for his comments in the early stages of the project, to Camille Le Calonnec for her insights on the implementation of adaptive variational quantum algorithms, to Maxime Dion for general discussion about the workings of the algorithm, and to Fen Zuo for pointing out the connection between ADAPT-Clifford and the Sahni-Gonzales family of MaxCut heuristics. This material is partially based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Quantum Systems Accelerator (QSA). Additional support is acknowledge from the Canada First Research Excellence Fund and the Ministère de l'Économie et de l'Innovation du Québec. S.K. is supported by a Research Chair in Quantum Computing by the Ministère de l'Économie, de l'Innovation et de l'Énergie du Québec.

APPENDIX A: THE GOEMANS-WILLIAMSON ALGORITHM

Suppose we are interested in solving the MaxCut problem for some given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of N nodes and edge weights $\omega_{i,j}$ using the Goemans-Williamson algorithm [24,25]. To do so one proceeds as follows:

(1) Relax the binary character of the variables in the optimization problem defined by the cost function in Eq. (1), that is, replace the $z_i \in \{0, 1\}$ with unit vectors $y_i \in \mathbb{R}^N$ and the product $z_i z_j$ with the inner product $y_i^T y_j$ with T the transpose. The new cost function $\sum_{i,j<i} \omega_{i,j}(1 - y_i^T y_j)$ with the constraints $y_i^T y_i = 1$, $\tilde{Y} = [y_i^T y_j]$ is positive semidefinite and defines a semidefinite program.

(2) Solve the semidefinite program using a polynomial time algorithm, and find an optimal solution $\tilde{Y}^*$ for the relaxed problem.

(3) Rounding: Choose a random vector $\mathbf{r} \in \mathbb{R}^N$ from a Gaussian distribution, and for all i define $h_i = \mathrm{sgn}(\mathbf{r}^T y_i^*)$, where $\mathrm{sgn}(x)$ is the sign function. This assignment defines a partition of the nodes in two disjoint sets $\mathcal{A} = \{i|h_i = 1\}$ and $\overline{\mathcal{A}} = \{i|h_i = -1\}$.

(4) Return the cut $(\mathcal{A}, \overline{\mathcal{A}})$.

In this form the algorithm only performs the rounding, step (3), a single time based on a single random vector $\mathbf{r}$. As such, a simple improvement consists of repeating this step $\mathcal{I}$ times for different random vectors and then returning the cut of largest cost among all the cuts found. We have used this approach in comparing our algorithm with GW.

All the results for the GW algorithm reported in this paper have been obtained using a freely available Julia implementation [114].

APPENDIX B: VALIDITY OF THE SEARCH THROUGH A RESTRICTED SET OF PAIRS

In step (2) of ADAPT-Clifford in Sec. IV, we restricted our search to pairs of the form $(\tilde{l}, b^{(r-1)})$ with $\tilde{l} \in \{k, j\}$ and (k, j) is the edge where the first two-qubit gate was applied, and $b^{(r-1)} \in \mathbf{b}^{(r-1)}$. In this Appendix we show than in doing so we do not miss the value of the largest gradient.

At step $r > 1$ the gradient is of the form

$$\begin{aligned} g^{(r)}_{a^{(r-1)},b^{(r-1)}} &= -\sum_l \omega_{l,b^{(r-1)}} \langle Z_l X_{b^{(r-1)}} Z_{a^{(r-1)}} \rangle_{r-1} \\ &= -\sum_l \omega_{l,b^{(r-1)}} \langle Z_l Z_{a^{(r-1)}} \rangle_{r-1}, \end{aligned} \tag{B1}$$

where we have used the fact that $X_{b^{(r-1)}}|\psi_{r-1}\rangle = |\psi_{r-1}\rangle$ since $b^{(r-1)}$ is inactive. The maximum of Eq. (B1) happens at the pair $(a^{(r-1)}, b^{(r-1)})$ such that the number of l's, with $l \in \mathbf{a}^{(r-1)}$, for which $-Z_l Z_{a^{(r-1)}}|\psi_{r-1}\rangle = |\psi_{r-1}\rangle$ is the largest.

Now consider the situation of interest where we search for the pair to apply the gate among those of the form $(\tilde{l}, b^{(r-1)})$, and suppose we know that the maximum of Eq. (B1) occurs at the pair $(\tilde{a}, \tilde{b})$ with $\tilde{a} \in \mathbf{a}^{(r-1)}$ and $\tilde{b} \in \mathbf{b}^{(r-1)}$. Then

$$g^{(r)}_{\tilde{a},\tilde{b}} = \sum_l \omega_{l,\tilde{b}} \langle -Z_l Z_{\tilde{a}} \rangle_{r-1} = \sum_l \omega_{l,\tilde{b}} \langle -Z_l Z_{\tilde{l}} Z_{\tilde{l}} Z_{\tilde{a}} \rangle_{r-1}, \tag{B2}$$

where we introduced an identity $\mathbb{I}_{\tilde{l}} = Z_{\tilde{l}} Z_{\tilde{l}}$.

Since $\tilde{a}$ is active and $-Z_k Z_j|\psi_{r-1}\rangle = |\psi_{r-1}\rangle$ we can always pick the value of $\tilde{l} \in \{k, j\}$ such that $Z_{\tilde{a}} Z_{\tilde{l}}|\psi_{r-1}\rangle = |\psi_{r-1}\rangle$. Thus, we can extend Eq. (B2) to the following chain of equalities:

$$g^{(r)}_{\tilde{a},\tilde{b}} = \sum_l \omega_{l,\tilde{b}} \langle -Z_l Z_{\tilde{l}} Z_{\tilde{l}} Z_{\tilde{a}} \rangle_{r-1} = \sum_l \omega_{l,\tilde{b}} \langle -Z_l Z_{\tilde{l}} \rangle_{r-1} = g^{(r)}_{\tilde{l},\tilde{b}}. \tag{B3}$$

We see then that the largest gradient does live within the restricted set of pairs of the form $(\tilde{l}, b^{(r-1)})$.

APPENDIX C: GRADIENT COMPUTATION WITHOUT EXPLICIT EVALUATION OF THE EXPECTATION VALUES

As mentioned in step 2 of our algorithm description, from $r > 1$ onwards we select the qubit $b^{(r)} \in \mathbf{b}^{(r)}$ with the largest gradient with either $\{k, j\}$ and move it to the active set. The

gradient is thus

$$g^{(r)}_{\tilde{l},b^{(r-1)}} = -\sum_{l} \omega_{l,b^{(r-1)}} \langle Z_l X_{b^{(r-1)}} Z_{\tilde{l}} \rangle_{r-1}, \tag{C1}$$

where $\tilde{l} \in \{k, j\}$. Since $b^{(r-1)}$ is an inactive qubit $X_{b^{(r-1)}}|\psi_{r-1}\rangle = |\psi_{r-1}\rangle$. Further, as was pointed out in Sec. IV only those terms for which $l \in \mathbf{a}^{(r-1)}$ are nonzero. We write the gradient as

$$g^{(r)}_{\tilde{l},b^{(r-1)}} = -\sum_{\substack{l\in\mathbf{a}^{(r-1)} \\ (l,b^{(r-1)})\in\mathcal{E}}} \omega_{l,b^{(r-1)}} \langle Z_l Z_{\tilde{l}} \rangle_{r-1}. \tag{C2}$$

Further note that the active qubits carry an additional label, indicating whether the qubit became active after being entangled with k or with j. This label allows us to write the active qubits as $\mathbf{a}^{(r)} = V^{(r)}_k \cup V^{(r)}_j$ with $V^{(r)}_k \cap V^{(r)}_j = \emptyset$. Consider an inactive qubit at step $r-2$; importantly if it becomes active after being entangled with k, then the stabilizer generator at its position, $X_{b^{(r-2)}}$, changes to $e^{-i\frac{\pi}{4}Y_{b^{(r-2)}}Z_k} X_{b^{(r-2)}} e^{i\frac{\pi}{4}Y_{b^{(r-2)}}Z_k} = Z_k Z_{b^{(r-2)}}$. Since $-Z_k Z_j$ is a stabilizer of the state, then $-Z_j Z_{b^{(r-2)}}$ also stabilizes the state. Similarly, if the qubit becomes active after being entangled with j, then after application of the respective two-qubit gate, the state is stabilized by both $Z_j Z_{b^{(r-2)}}$ and $-Z_k Z_{b^{(r-2)}}$. With these expressions we can immediately write the gradient as

$$g^{(r)}_{k,b^{(r-1)}} = -\sum_{\substack{l\in\mathbf{V}^{(r-1)}_k \\ (l,b^{(r-1)})\in\mathcal{E}}} \omega_{l,b^{(r-1)}} + \sum_{\substack{l\in\mathbf{V}^{(r-1)}_j \\ (l,b^{(r-1)})\in\mathcal{E}}} \omega_{l,b^{(r-1)}}, \tag{C3}$$

and $g^{(r)}_{j,b^{(r-1)}} = -g^{(r)}_{k,b^{(r-1)}}$. We thus see that the computation of the gradient does not require the explicit computation of the expectation values seen in Eq. (16).

We would like to point out that the explicit form of the gradient in Eq. (C3) establishes a connection with a family of heuristic algorithms for MaxCut known by the name of Sahni-Gonzalez [87,88]. This connection was recently studied in Ref. [89].

APPENDIX D: TIME TO SOLUTION ANALYSIS: ADAPT-CLIFFORD VS GOEMANS-WILLIAMSON

In order to complete the comparison between the performance of ADAPT-Clifford and GW, we conducted a time-to-solution (TTS) analysis. This study was carried out on an Apple M1 Pro laptop with eight cores. We focused on the case of weighted complete graphs studied in Sec. V A, and compared our Python-based implementation of ADAPT-Clifford available at [90] against the freely available Julia implementation of GW for the MaxCut problem [114]. Since the TTS of deterministic ADAPT-Clifford can be easily determined as one order of magnitude larger than that of randomized ADAPT-Clifford, we restricted the TTS study to the latter variant of our algorithm. For the GW we restricted ourselves to consider the situations of only $\mathcal{I} = 10^0$ and $\mathcal{I} = 10^5$, the two extreme values taken for the results in Sec. V A.

The results are shown in Fig. 9. Several observations are in order. The TTS of GW shows a relative slow increase with system size for small problems, until about $N \sim 300$ after which the familiar $O(N^3)$ scaling is observed. Additionally,

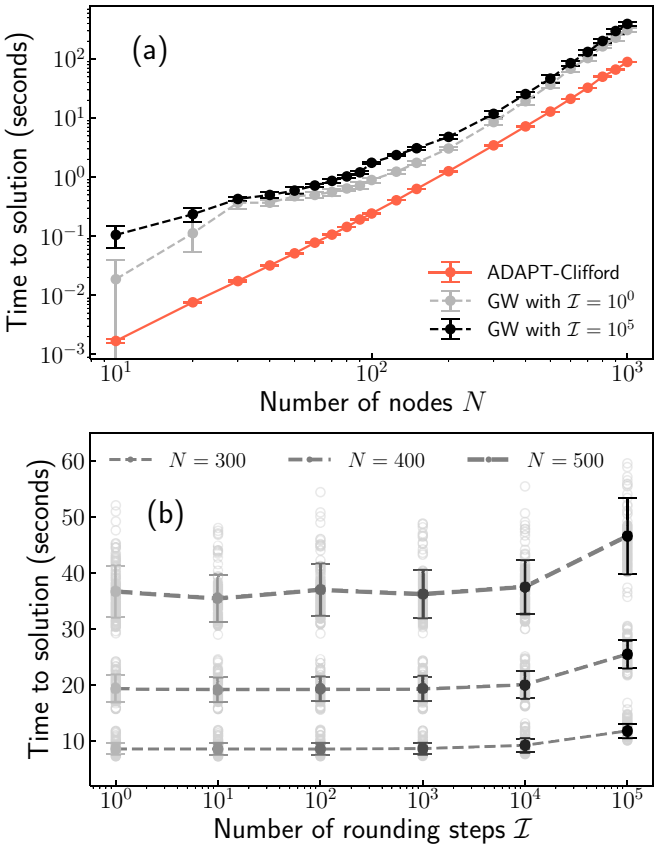

FIG. 9. (a) Problem instance-averaged time to solution for randomized ADAPT-Clifford (orange solid line) and GW with $\mathcal{I} = 10^0, 10^5$ (dashed light gray and black), for the weighted complete graphs considered in Fig. 2. (b) Time to solution of GW as a function of the number of times the rounding step I is performed. Results are shown for three different system sizes, $N = 300, 400, 500$ from bottom to top, respectively. The empty circles correspond to all the individual TTS for the different 100 problem instances solved per problem size, the full circle to the mean TTS, and the error bars showing the standard deviation. The dashed lines are guides for the eye.

we do not find a strong dependence of the TTS with $\mathcal{I}$; in fact, for the problem sizes considered in this work, i.e., up to $N = 1000$, the TTS does not increase with $\mathcal{I}$ as this is not bigger than $O(10^2)$; see Fig. 9(b). When the value of I exceeds this threshold, we do observe an increase in the runtime albeit not a considerable one. As such, deterministic ADAPT-Clifford will be superior in both solution quality and TTS only up to $N = 30$.

For randomized ADAPT-Clifford the TTS is always better than GW; see Fig. 9(a). In fact, we empirically find a more advantageous scaling of $O(N^{2.7})$ when contrasted to the $O(N^3)$ found in the same manner for GW. However, this makes the comparison considerably more subtle. Although the quality of solution found by GW is already better than randomized ADAPT-Clifford when $\mathcal{I} > 10$, GW relies on a Cholesky decomposition which has highly optimized numerical subroutines that exploit the multicore nature of modern CPUs. However, randomized ADAPT-Clifford is based on STIM and

thus runs on a single core. One could then easily improve the quality of solution without sacrificing the TTS or runtime scaling by executing the algorithm for different initial position in parallel. We thus believe that this variant of the algorithm does offer an advantage over GW.

APPENDIX E: RESOURCE ESTIMATES FOR ADAPT-CLIFFORD

As mentioned in Sec. VII one might wonder how ADAPT-Clifford compares with ADAPT-QAOA, and what would be the resource demands to implement ADAPT-Clifford on a near-term device. In this Appendix we explore these two questions.

Given the computational cost of directly simulating ADAPT-QAOA we restrict the comparison to small problem instances, such as those solved in Fig. 1. As we observed in Sec. III, the best performing ADAPT-QAOA solution circuits are very close to Clifford. In this sense, the performance, quantified by the approximation ratio, of both ADAPT-QAOA and ADAPT-Clifford, for these small problem instances, must be the same. Now, if the comparison is extended to other metrics, for instance, number of layers or steps, cost of parameter optimization, or cost of gradient evaluation, then ADAPT-Clifford will be best.

In order to see this we will show that ADAPT-Clifford can be directly obtained from ADAPT-QAOA by systematically reducing the ansatz expressivity. We start from the full ADAPT-QAOA ansatz in Eq. (7), and use the observations summarized at the end of Sec. III to systematically reduced the ansatz expressivity. First, we start from the state $Z_k \mathrm{H}^{\otimes N}|0\rangle^{\otimes N}$ where k is chosen randomly. Second, we set all $\gamma_l = 0$ and the $\beta_l = -\pi/4$, this guarantees the resulting circuit to be Clifford. Third, we restrict the operator pool in Eq. (9) to $\mathrm{P}_{\mathrm{OP}} = \{Y_l Z_m, Z_l Y_m\}_{j,k=1,\ldots,N,j\neq k}$,, where one of the two indices is fixed by our choice of k. Fourth, we use the gradient Criteria to select the mixer Hamiltonian at any intermediate step.

The above algorithm is a direct simplification of the full ADAPT-QAOA ansatz, and although it is different from the algorithm presented in Sec. IV, it already allows us to see why ADAPT-Clifford is a more efficient algorithm. First, the gradient evaluation is highly simplified by the reduction of the operator pool size. Second, the need for parameter optimization is bypassed. Third, the restriction to Clifford unitaries enables efficient classical simulation. Finally, for ADAPT-Clifford we showed in Appendix C that the gradient evaluation can be done without the need for explicit evaluation of expectation values, which further speeds up the algorithm.

Let us now focus on the second question. We based the resource estimation on the CNOT count. First, let us consider a qubit chip with all-to-all connectivity. In this case, and given the gate shown in Eq. (21), it is easy to see that the total number of CNOT gates required to run the circuit is $2N$, with N the number of nodes in the problem graph.

Consider now the opposite case, a qubit architecture with only linear connectivity. In order to apply a CNOT between two qubits at arbitrary positions, say, l and m, we bring the second qubit m to $l+1$ using a swap network, then apply the Clifford gate, and swap the second qubit back to its original position m. In the following we use the fact that a swap can be implemented with three CNOT gates. Further, we assume that the two reference qubits, k and j, are on opposite ends of the chain, thus $|k-j| = N-1$, and implementing the entangling Clifford gate between these two qubits requires $6(N-2)+2$ CNOT gates. Finally, we assume the worst scenario in terms of separation between the qubits, where the entangling Clifford gates need to be applied between the reference qubit and the qubit which is farthest from it, among those inactive. Thus, the rest of the $N-2$ steps can be implemented using $2(N-2) + 6\sum_{s=0}^{N-3} s = 3(N-3)(N-2) + 2(N-2)$ CNOT gates, which leads to a total CNOT count of $2 + (N-2)[3(N-3)+8]$.

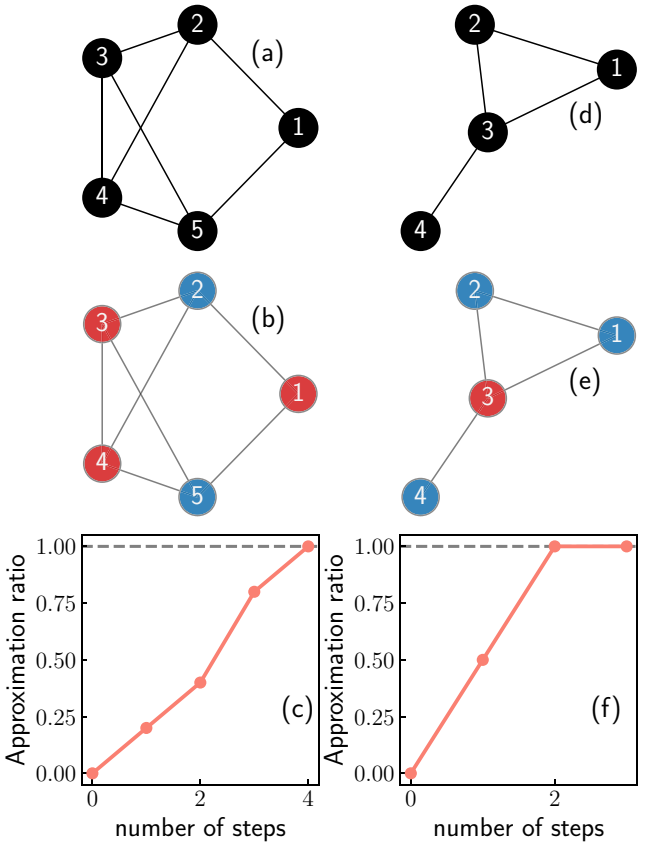

FIG. 10. (a), (d) Example graphs with $N = 5$ and $N = 4$ nodes, respectively. (b), (e) Partitioned graphs according to the cuts produced by our algorithm, different colors denote the nodes in each of the disjoint subsets. (c), (f) Approximation ratios of the states produced by our algorithm in the search process for the maximal cut of the graphs shown in (a) and (d). In both cases the algorithm reaches approximation ratio of 1, indicating a maximal cut has been found.

APPENDIX F: SOME EXPLICIT EXAMPLES OF ADAPT-CLIFFORD SOLVING MaxCut

In this Appendix, we go over the full analytical calculation of the steps involved in solving MaxCut using the algorithm introduced in Sec. IV for two small graphs with $N = 4, 5$

nodes. In order to keep the expressions clean we have decided to focus on the case of unweighted graphs.

1. Example with $N = 5$ nodes

Consider the unweighted graph with five nodes shown in Fig. 10(a). Its adjacency matrix is given by

$$[\omega_{i,j}] = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}. \tag{F1}$$

We will solve MaxCut on this graph using our algorithm. We begin by flipping the state of the qubit at $k = 2$; thus we have

$$|\psi_0\rangle = |+-+++\rangle, \tag{F2}$$

where $|+\rangle = \mathrm{H}|0\rangle$ and $|-\rangle = Z\mathrm{H}|0\rangle$ are the eigenstates of the Pauli-x operator corresponding to eigenvalues $+1$ and -1, respectively. At this point we have to initialize the records of active and innactive qubits, which we identify by their respective indices. The active qubits are [2] and the inactive qubits are [1,3,4,5].

Given our choice of initial position, we have that $g^{(1)}_{1,2} = g^{(1)}_{2,3} = g^{(1)}_{2,4} = 1$ and they are the largest "gradients." We break this tie arbitrarily and chose the pair of qubits (2,4). Then

$$|\psi_1\rangle = e^{i\frac{\pi}{4}Y_2Z_4}|\psi_0\rangle = \frac{1}{\sqrt{2}}[|+-+++\rangle - |+++-+\rangle], \tag{F3}$$

and the records of the active and inactive qubits are updated to be [2,4] and [1,3,5], respectively. The second set of gradients is given by

$$\begin{aligned} g^{(2)}_{1,2} &= -\sum_{l=2,5}\langle Z_l X_1 Z_2\rangle \\ &= -\langle X_1\rangle - \langle Z_5 X_1 Z_2\rangle \\ &= -1 + 0 = -1, \end{aligned} \tag{F4a}$$

$$\begin{aligned} g^{(2)}_{3,2} &= -\sum_{l=2,4,5}\langle Z_l X_3 Z_2\rangle \\ &= -\langle X_3\rangle - \langle Z_4 X_3 Z_2\rangle - \langle Z_5 X_3 Z_2\rangle \\ &= -1 + 1 + 0 = 0, \end{aligned} \tag{F4b}$$

$$\begin{aligned} g^{(2)}_{5,2} &= -\sum_{l=1,3,4}\langle Z_l X_5 Z_2\rangle \\ &= -\langle Z_1 X_5 Z_2\rangle - \langle Z_3 X_5 Z_2\rangle - \langle Z_4 X_5 Z_2\rangle \\ &= 0 + 0 + 1 = 1, \end{aligned} \tag{F4c}$$

$$\begin{aligned} g^{(2)}_{1,4} &= -\sum_{l=2,5}\langle Z_l X_1 Z_4\rangle \\ &= -\langle Z_2 X_1 Z_4\rangle - \langle Z_5 X_1 Z_4\rangle \\ &= 1 + 0 = 1, \end{aligned} \tag{F4d}$$

$$\begin{aligned} g^{(2)}_{3,4} &= -\sum_{l=2,4,5}\langle Z_l X_3 Z_4\rangle \\ &= -\langle Z_2 X_3 Z_4\rangle - \langle X_3\rangle - \langle Z_5 X_3 Z_4\rangle \\ &= 1 - 1 + 0 = 0, \end{aligned} \tag{F4e}$$

$$\begin{aligned} g^{(2)}_{5,4} &= -\sum_{l=1,3,4}\langle Z_l X_5 Z_4\rangle \\ &= -\langle Z_1 X_5 Z_4\rangle - \langle Z_3 X_5 Z_4\rangle - \langle X_5\rangle = -1, \end{aligned} \tag{F4f}$$

and the largest gradients are $g^{(2)}_{5,2}$ and $g^{(2)}_{1,4}$. Since they are equal, we break the tie arbitrarily and chose the pair of qubits (1,4). Thus, the state at step $r = 2$ is given by

$$\begin{aligned} |\psi_2\rangle &= e^{i\frac{\pi}{4}Y_1Z_4}|\psi_1\rangle \\ &= \tfrac{1}{2}[|+-++++\rangle - |--+-+\rangle \\ &\quad - |+++-+\rangle - |-+++\!+\rangle]. \end{aligned} \tag{F5}$$

After the application of the gate we update the records of active and inactive qubits, which now are [1,2,4] and [3,5], respectively. The third set of gradients is given by

$$\begin{aligned} g^{(3)}_{3,1} &= -\sum_{l=2,4,5}\langle Z_l X_3 Z_1\rangle \\ &= -\langle Z_2 X_3 Z_1\rangle - \langle Z_4 X_3 Z_1\rangle - \langle Z_5 X_3 Z_1\rangle \\ &= 1 - 1 + 0 = 0, \end{aligned} \tag{F6a}$$

$$\begin{aligned} g^{(3)}_{5,1} &= -\sum_{l=1,3,4}\langle Z_l X_5 Z_1\rangle \\ &= -\langle X_5\langle -\langle Z_3 X_5 Z_1\rangle - \langle Z_4 X_5 Z_1\rangle \\ &= -1 + 0 - 1 = -2, \end{aligned} \tag{F6b}$$

$$\begin{aligned} g^{(3)}_{3,2} &= -\sum_{l=2,4,5}\langle Z_l X_3 Z_2\rangle \\ &= -\langle X_3\rangle - \langle Z_4 X_3 Z_2\rangle - \langle Z_5 X_3 Z_2\rangle \\ &= -1 + 1 + 0 = 0, \end{aligned} \tag{F6c}$$

$$\begin{aligned} g^{(3)}_{5,2} &= -\sum_{l=1,3,4}\langle Z_l X_5 Z_2\rangle \\ &= -\langle Z_1 X_5 Z_2\rangle - \langle Z_3 X_5 Z_2\rangle - \langle Z_4 X_5 Z_2\rangle \\ &= 1 + 0 + 1 = 2, \end{aligned} \tag{F6d}$$

$$\begin{aligned} g^{(3)}_{3,4} &= -\sum_{l=2,4,5}\langle Z_l X_3 Z_4\rangle \\ &= -\langle Z_2 X_3 Z_4\rangle - \langle X_3\rangle - \langle Z_5 X_3 Z_4\rangle \\ &= 1 - 1 + 0 = 0, \end{aligned} \tag{F6e}$$

$$\begin{aligned} g^{(3)}_{5,4} &= -\sum_{l=1,3,4}\langle Z_{l X_5 Z_4}\rangle \\ &= -\langle Z_1 X_5 Z_4\rangle - \langle Z_3 X_5 Z_4\rangle - \langle X_5\rangle \\ &= -1 + 0 - 1 = -2. \end{aligned} \tag{F6f}$$

The largest gradient is $g^{(3)}_{5,2} = 2$, and the gate is applied at the pair of qubits (2,5), where 2 is an active qubit and 5 is inactive. The state at step $r = 3$ is given by

$$\begin{aligned} |\psi_3\rangle &= e^{i\frac{\pi}{4}Z_2Y_5}|\psi_2\rangle \\ &= \frac{1}{2\sqrt{2}}[|+-++\!+\rangle + |+++\!+-\rangle \\ &\quad + |--+-+\rangle + |-++--\rangle \\ &\quad - |+++-+\rangle - |+-+--\rangle \\ &\quad - |-+++\!+\rangle - |--++-\rangle], \end{aligned} \tag{F7}$$

with the records of active and inactive qubits updated to [1,2,4,5] and [3], respectively. From this state we can compute the set of gradients of step $r = 4$. They are given by

$$\begin{aligned} g^{(4)}_{1,3} &= -\sum_{l=2,4,5} \langle Z_l X_3 Z_1 \rangle \\ &= -\langle Z_2 X_3 Z_1 \rangle - \langle Z_4 X_3 Z_1 \rangle - \langle Z_5 X_3 Z_1 \rangle \\ &= 1 - 1 + 1 = 1, \end{aligned} \tag{F8a}$$

$$\begin{aligned} g^{(4)}_{2,3} &= -\sum_{l=2,4,5} \langle Z_l X_3 Z_2 \rangle \\ &= -\langle X_3 \rangle - \langle Z_4 X_3 Z_2 \rangle - \langle Z_5 X_3 Z_2 \rangle \\ &= -1 + 1 - 1 = -1, \end{aligned} \tag{F8b}$$

$$\begin{aligned} g^{(4)}_{4,3} &= -\sum_{l=2,4,5} \langle Z_l X_3 Z_4 \rangle \\ &= -\langle Z_2 X_3 Z_4 \rangle - \langle X_3 \rangle - \langle Z_5 X_3 Z_4 \rangle \\ &= +1 - 1 + 1 = 1, \end{aligned} \tag{F8c}$$

$$\begin{aligned} g^{(4)}_{5,3} &= -\sum_{l=2,4,5} \langle Z_l X_3 Z_5 \rangle \\ &= -\langle Z_2 X_3 Z_5 \rangle - \langle Z_4 X_3 Z_5 \rangle - \langle X_3 \rangle \\ &= -1 + 1 - 1 = -1. \end{aligned} \tag{F8d}$$

There are two largest gradients, $g^{(4)}_{1,3} = g^{(4)}_{4,3} = 1$. We break the tie arbitrarily and take the pair of qubits (3,4). Thus, the state at step $r = 4$ is given by

$$\begin{aligned} |\psi_4\rangle &= e^{i\frac{\pi}{4} Y_3 Z_4} |\psi_3\rangle \\ &= \tfrac{1}{4}[|+-++{+}\rangle + |+--{-}{+}\rangle + |+++{+}{-}\rangle \\ &\quad + |++{-}{-}{-}\rangle + |--+{-}{+}\rangle \\ &\quad + |---{+}{+}\rangle + |-++{-}{-}\rangle + |-+-{+}{-}\rangle \\ &\quad - |+++{-}{+}\rangle - |++{-}{+}{+}\rangle - |+-+{-}{-}\rangle \\ &\quad - |+--{+}{-}\rangle - |-++{+}{+}\rangle - |-+-{-}{+}\rangle \\ &\quad - |--+{+}{-}\rangle - |----\rangle]. \end{aligned} \tag{F9}$$

To extract the cut found by our algorithm we should write $|\psi_4\rangle$ in the computational basis. In order to do this we use its stabilizers, which are

$$-XXXXX, \quad -Z\mathbb{I}\mathbb{I}\mathbb{I}Z, \quad +\mathbb{I}Z\mathbb{I}\mathbb{I}Z, \quad -\mathbb{I}\mathbb{I}Z\mathbb{I}Z, \quad -\mathbb{I}\mathbb{I}\mathbb{I}ZZ,$$

which correspond to the state

$$|\psi_4\rangle = \frac{1}{\sqrt{2}}(|10110\rangle - |01001\rangle), \tag{F10}$$

in the computational basis. This state upon a measurement in this basis returns the cut $(\mathcal{A}, \overline{\mathcal{A}}) = ([1, 3, 4], [2, 5])$, which is a maximal cut of the graph under consideration. We illustrate this partitioning of the graph by coloring the nodes in $\mathcal{A}$ red and those in $\overline{\mathcal{A}}$ blue, and show the resulting partitioned graph in Fig. 10(b). Additionally in Fig. 10(c) we show the approximation ratio of the states $|\psi_r\rangle$ computed in this section, notice that at $r = 4$ we have approximation ratio equal to 1, indicating the algorithm found a state composed of strings encoding maximal cuts.

2. Example with $N = 4$ nodes

We consider now the graph with $N = 4$ nodes shown in Fig. 10(d). Its adjacency matrix is given by

$$[\omega_{i,j}] = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{F11}$$

Let us start the algorithm with the state $|\psi_0\rangle = |+-++\rangle$. For this state there are two largest gradients at step $r = 1$ given by $g^{(1)}_{2,3} = g^{(1)}_{2,1} = 1$. We break the tie arbitrarily and pick the pair of qubits (2,3). Thus, the state at step $r = 1$ is given by

$$|\psi_1\rangle = e^{i\frac{\pi}{4} Y_2 Z_3} |\psi_0\rangle = \frac{1}{\sqrt{2}}[|+-++\rangle - |++-+\rangle]. \tag{F12}$$

Now the gradients at step $r = 2$ are given by

$$\begin{aligned} g^{(2)}_{1,2} &= -\sum_{l=2,3} \langle Z_l X_1 Z_2 \rangle = -\langle X_1 \rangle - \langle Z_3 X_1 Z_2 \rangle \\ &= -1 + 1 = 0, \end{aligned} \tag{F13a}$$

$$\begin{aligned} g^{(2)}_{1,3} &= -\sum_{l=2,3} \langle Z_l X_1 Z_3 \rangle = -\langle Z_2 X_1 Z_3 \rangle - \langle X_1 \rangle \\ &= 1 - 1 = 0, \end{aligned} \tag{F13b}$$

$$g^{(2)}_{4,2} = -\langle Z_3 X_4 Z_2 \rangle = 1, \tag{F13c}$$

$$g^{(2)}_{4,3} = -\langle X_4 \rangle = -1. \tag{F13d}$$

Thus, the largest gradient is $g^{(2)}_{4,2} = 1$. We apply the next gate to the pair (2,4) leading to a state at step $r = 2$ of the form

$$\begin{aligned} |\psi_2\rangle &= e^{i\frac{\pi}{4} Z_2 Y_4} |\psi_1\rangle = \tfrac{1}{2}[|+-++\rangle + |+++{-}\rangle \\ &\quad - |++{-}{+}\rangle - |+--{-}\rangle]. \end{aligned} \tag{F14}$$

For the next step we find all three gradients $g^{(3)}_{1,2} = g^{(3)}_{1,3} = g^{(3)}_{1,4} = 0$, thus no gate needs to be added in this last step. We verify this by looking at the approximation ratio of the states produced by the algorithm, shown in Fig. 10(f), and we observe that only after two steps does the algorithm reach an approximation ratio of 1, indicating a maximal cut has been found. In order to extract this cut we write $|\psi_2\rangle$ in the computational basis as

$$|\psi_2\rangle = \tfrac{1}{2}[|0010\rangle + |1101\rangle - |1010\rangle - |0101\rangle]; \tag{F15}$$

notice that the algorithm prepares a state which encodes two distinct maximal cuts for the graph under consideration: one of the form $(\mathcal{A}, \overline{\mathcal{A}}) = ([1, 2, 4], [3])$ which we illustrate in Fig. 10(e), and one of the form $(\mathcal{A}, \overline{\mathcal{A}}) = ([1, 3], [2, 4])$.

APPENDIX G: ESTIMATION OF THE MEAN APPROXIMATION RATIOS FOR THE CASE OF POSITIVE WEIGHTS

In this Appendix we present the method used to estimate $\overline{\alpha}$ and $\overline{\alpha}^{\mathrm{r}}$ reported in Sec. V A. Recall that our algorithm solves the problem N times, each time starting from a different

position $k \in [1, N]$. As such, for the same problem we might have up to N different α's.

We begin by fixing a threshold value $\alpha_{\rm tr}$ for a given graph ensemble. For $N \in [10, 30]$, we count how many initial positions k, $\mathrm{Num}(N; \alpha_{\rm tr})$ lead to a solution with $\alpha > \alpha_{\rm tr}$. We repeat this process for all problem instances considered and obtain $\mathbb{E}[\mathrm{Num}(N; \alpha_{\rm tr})]$. At this point, we perform a linear fit to the data $(N, \mathbb{E}[\mathrm{Num}(N; \alpha_{\rm tr})])$ and obtain the slope $\mathcal{M} = \mathcal{M}(\alpha_{\rm tr})$. We then vary the threshold $\alpha_{\rm tr} \in [0.88, 1]$ and repeat the above procedure. Once all the data $(\alpha_{\rm tr}, \mathcal{M}(\alpha_{\rm tr}))$ have been obtained we identify $\overline{\alpha} = \alpha_{\rm tr}|_{\mathcal{M}(\alpha_{\rm tr})=0}$, the last threshold value before the slope becomes negative. The largest approximation ratio we can guarantee is thus the one for which no initial position $k \leqslant N$ leads to $\alpha = \alpha_{\rm tr}$. To account for fluctuations among instances, the linear fit is done to the data $(N, \mathbb{E}[\mathrm{Num}(N; \alpha_{\rm tr})] - \sigma[\mathrm{Num}(N; \alpha_{\rm tr})])$, with $\sigma[\mathrm{Num}(N; \alpha_{\rm tr})]$ one standard deviation. As reported in the main text, this procedure leads to $\overline{\alpha} = 0.9686$ for the case of positive-weighted complete graphs with $N \in [10, 30]$ and 100 instances per N.

For the case of randomly chosen initial condition we identify $\overline{\alpha}^{\rm r} = \alpha_{\rm tr}|_{\mathcal{M}(\alpha_{\rm tr})=0.5}$, that is, the threshold for which at least half of the possible initial conditions will lead to, on average, an approximation ratio equal to the threshold. As reported in the main text, this procedure leads to $\overline{\alpha}^{\rm r} = 0.8986$.

[1] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, New J. Phys. **18**, 023023 (2016).

[2] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[3] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke *et al.*, Noisy intermediate-scale quantum algorithms, Rev. Mod. Phys. **94**, 015004 (2022).

[4] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

[5] S. Hadfield, Z. Wang, B. O'gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, Algorithms **12**, 34 (2019).

[6] G. E. Crooks, Performance of the quantum approximate optimization algorithm on the maximum cut problem, arXiv:1811.08419.

[7] J. Golden, A. Bärtschi, D. O'Malley, and S. Eidenbenz, Numerical evidence for exponential speed-up of QAOA over unstructured search for approximate constrained optimization, *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (Bellevue, WA, USA, 2023), pp. 496–505.

[8] S. Boulebnane and A. Montanaro, Solving Boolean satisfiability problems with the quantum approximate optimization algorithm, arXiv:2208.06909.

[9] C. Carlson, Z. Jorquera, A. Kolla, and S. Kordonowy, Comparing a classical and quantum one round algorithm on LocalMaxCut, arXiv:2304.08420.

[10] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar *et al.*, Quantum optimization of maximum independent set using Rydberg atom arrays, Science **376**, 1209 (2022).

[11] R. S. Andrist, M. J. Schuetz, P. Minssen, R. Yalovetzky, S. Chakrabarti, D. Herman, N. Kumar, G. Salton, R. Shaydulin, Y. Sun *et al.*, Hardness of the maximum independent set problem on unit-disk graphs and prospects for quantum speedups, Phys. Rev. Res. **5**, 043277 (2023).

[12] R. Shaydulin, P. C. Lotshaw, J. Larson, J. Ostrowski, and T. S. Humble, Parameter transfer for quantum approximate optimization of weighted MaxCut, ACM Trans. Quantum Comput. **4**, 1 (2023).

[13] A. Gilyén, S. Lloyd, and E. Tang, Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension, arXiv:1811.04909.

[14] E. Tang, A quantum-inspired classical algorithm for recommendation systems, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, Phoenix, AZ, New York, NY, US, 2019), pp. 217–228

[15] A. Gilyén, Z. Song, and E. Tang, An improved quantum-inspired algorithm for linear regression, Quantum **6**, 754 (2022).

[16] J. M. Arrazola, A. Delgado, B. R. Bardhan, and S. Lloyd, Quantum-inspired algorithms in practice, Quantum **4**, 307 (2020).

[17] Y. Tene-Cohen, T. Kelman, O. Lev, and A. Makmal, A variational qubit-efficient MaxCut heuristic algorithm, arXiv:2308.10383.

[18] A. Misra-Spieldenner, T. Bode, P. K. Schuhmacher, T. Stollenwerk, D. Bagrets, and F. K. Wilhelm, Mean-field approximate optimization algorithm, PRX Quantum **4**, 030335 (2023).

[19] D. Gottesman, Class of quantum error-correcting codes saturating the quantum Hamming bound, Phys. Rev. A **54**, 1862 (1996).

[20] D. Gottesman, Stabilizer codes and quantum error correction, Ph.D. thesis, California Institute of Technology (1997).

[21] D. Gottesman, The Heisenberg representation of quantum computers, arXiv:quant-ph/9807006.

[22] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, Phys. Rev. A **70**, 052328 (2004).

[23] C. Gidney, Stim: A fast stabilizer circuit simulator, Quantum **5**, 497 (2021).

[24] M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, J. ACM **42**, 1115 (1995).

[25] M. X. Goemans, Worst-case comparison of valid inequalities for the TSP, Math. Program.: Serias A and B **69**, 335 (1995).

[26] R. M. Karp, Reducibility among combinatorial problems, in *Complexity of Computer Computations*, edited by R. E. Miller, J. W. Thatcher, and J. D. Bohlinger (Springer US, New York, 1972), pp. 85–103.

[27] There is no polynomial time algorithm to solve the problem exactly in the worst case. It has also shown to be APX-hard [115] no polynomial time approximation scheme exists unless P=NP.

[28] M. Grötschel and G. L. Nemhauser, A polynomial algorithm for the Max-Cut problem on graphs without long odd cycles, Math. Program. **29**, 28 (1984).

[29] M. Grötschel and W. R. Pulleyblank, Weakly bipartite graphs and the Max-Cut problem, Oper. Res. Lett. **1**, 23 (1981).

[30] W.-K. Shih, S. Wu, and Y. Kuo, Unifying maximum cut and minimum cut of a planar graph, IEEE Trans. Comput. **39**, 694 (1990).

[31] F. Liers and G. Pardella, Partitioning planar graphs: A fast combinatorial approach for Max-Cut, Comput. Optim. Appl. **51**, 323 (2012).

[32] F. Hadlock, Finding a maximum cut of a planar graph in polynomial time, SIAM J. Comput. **4**, 221 (1975).

[33] G. Orlova, Finding the maximum cut in a graph, Eng. Cybern. **10**, 502 (1972).

[34] C. Dahn, N. M. Kriege, and P. Mutzel, A fixed-parameter algorithm for the Max-Cut problem on embedded 1-planar graphs, in *International Workshop on Combinatorial Algorithms* (Springer, 2018), pp. 141–152.

[35] M. Chimani, C. Dahn, M. Juhnke-Kubitzke, N. M. Kriege, P. Mutzel, and A. Nover, Maximum cut parameterized by crossing number, J. Graph Alg. Appl. **24**, 155 (2020).

[36] M. Stoer and F. Wagner, A simple min-cut algorithm, J. ACM **44**, 585 (1997).

[37] S. Khot, Inapproximability of NP-complete problems, discrete Fourier analysis, and geometry, in *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010)*, edited by R. Bathia, A. Pal, V. Srinivas, and M. Vanninathan (World Scientific, Hyderabad, India, 2010), pp. 2676–2697.

[38] H. Karloff, How good is the Goemans–Williamson MAX CUT algorithm? SIAM J. Comput. **29**, 336 (1999).

[39] We point out that several heuristic algorithms do perform well; see Ref. [116] for a comparison. However, at the moment these do not have known formal guarantees.

[40] J. Håstad, Some optimal inapproximability results, J. ACM **48**, 798 (2001).

[41] E. Halperin, D. Livnat, and U. Zwick, MAX CUT in cubic graphs, J. Alg. **53**, 169 (2004).

[42] A. El Alaoui, A. Montanari, and M. Sellke, Local algorithms for maximum cut and minimum bisection on locally treelike regular graphs of large degree, Random Struct. Alg. **63**, 689 (2021).

[43] W. Fernandez de la Vega, MAX-CUT has a randomized approximation scheme in dense graphs, Random Struct. Alg. **8**, 187 (1996).

[44] S. Arora, D. Karger, and M. Karpinski, Polynomial time approximation schemes for dense instances of NP-hard problems, J. Comput. Syst. Sci. **58**, 193 (1999).

[45] W. Fernandez de la Vega and M. Karpinski, Polynomial time approximation of dense weighted instances of MAX-CUT, Random Struct. Alg. **16**, 314 (2000).

[46] A. Lucas, Ising formulations of many NP problems, Front. Phys. **2**, 5 (2014).

[47] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[48] J. Wurtz and P. Love, MaxCut quantum approximate optimization algorithm performance guarantees for $p > 1$, Phys. Rev. A **103**, 042612 (2021).

[49] P. Díez-Valle, D. Porras, and J. J. García-Ripoll, Quantum approximate optimization algorithm pseudo-Boltzmann states, Phys. Rev. Lett. **130**, 050601 (2023).

[50] P. C. Lotshaw, G. Siopsis, J. Ostrowski, R. Herrman, R. Alam, S. Powers, and T. S. Humble, Approximate Boltzmann distributions in quantum approximate optimization, Phys. Rev. A **108**, 042411 (2023).

[51] C.-N. Chou, P. J. Love, J. S. Sandhu, and J. Shi, Limitations of local quantum algorithms on random MAX-k-XOR and beyond, in *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, edited by M. Bojańczyk, E. Merelli, and David P. Woodruff, Leibniz International Proceedings in Informatics (LIPIcs) (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2021).

[52] E. Farhi, D. Gamarnik, and S. Gutmann, The quantum approximate optimization algorithm needs to see the whole graph: A typical case, arXiv:2004.09002.

[53] E. Farhi, D. Gamarnik, and S. Gutmann, The quantum approximate optimization algorithm needs to see the whole graph: Worst case examples, arXiv:2005.08747.

[54] The overlap gap property (OGP) is a property of the geometry of the space of solutions which (very) roughly speaking states that nearly optimal solutions are either close or far apart from each other. The OGP is currently understood as an algorithmic barrier as it implies a large family of algorithms have their best solutions bounded away from optimality. For an extended discussion see Ref. [117].

[55] J. Basso, D. Gamarnik, S. Mei, and L. Zhou, Performance and limitations of the qaoa at constant levels on large sparse hypergraphs and spin glass models, in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, 2022), pp. 335–343.

[56] A. Chen, N. Huang, and K. Marwaha, Local algorithms and the failure of log-depth quantum advantage on sparse random CSPs, arXiv:2310.01563.

[57] N. Benchasattabuse, A. Bärtschi, L. P. García-Pintos, J. Golden, N. Lemons, and S. Eidenbenz, Lower bounds on number of QAOA rounds required for guaranteed approximation ratios, arXiv:2308.15442.

[58] 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).

[59] With further putative evidence coming from numerical experiments relying on techniques to bypass the classical optimization of the variational parameters [12,58,102,118] or to steer the optimization process in a more efficient and effective manner [119].

[60] 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).

[61] 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).
[62] Y. Chen, L. Zhu, N. J. Mayhall, E. Barnes, and S. E. Economou, How much entanglement do quantum optimization algorithms require? in *Quantum 2.0* (Optica Publishing Group, Boston, MA, 2022), pp. QM4A-2.
[63] C. Feniou, B. Claudon, M. Hassan, A. Courtat, O. Adjoua, Y. Maday, and J.-P. Piquemal, Adaptive variational quantum algorithms on a noisy intermediate scale quantum computer, arXiv:2306.17159.
[64] In practice the structure of the cost landscape forces us to initialize γ_l to a small positive value, which we fix to be $\gamma_l = 0.01$; see Ref. [61] for details.
[65] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, UK, 2010).
[66] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, Mixed-state entanglement and quantum error correction, Phys. Rev. A **54**, 3824 (1996).
[67] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, Quantum error correction and orthogonal geometry, Phys. Rev. Lett. **78**, 405 (1997).
[68] C. H. Bennett and S. J. Wiesner, Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states, Phys. Rev. Lett. **69**, 2881 (1992).
[69] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels, Phys. Rev. Lett. **70**, 1895 (1993).
[70] T. Begušić, K. Hejazi, and G. K. Chan, Simulating quantum circuit expectation values by Clifford perturbation theory, arXiv:2306.04797.
[71] T. Begušić and G. K. Chan, Fast classical simulation of evidence for the utility of quantum computing before fault tolerance, arXiv:2306.16372.
[72] A. K. Daniel and A. Miyake, Quantum computational advantage with string order parameters of one-dimensional symmetry-protected topological order, Phys. Rev. Lett. **126**, 090505 (2021).
[73] A. K. Daniel, Y. Zhu, C. H. Alderete, V. Buchemmavari, A. M. Green, N. H. Nguyen, T. G. Thurtell, A. Zhao, N. M. Linke, and A. Miyake, Quantum computational advantage attested by nonlocal games with the cyclic cluster state, Phys. Rev. Res. **4**, 033068 (2022).
[74] A. Kitaev, Quantum error correction with imperfect gates, in *Quantum Communication, Computing, and Measurement*, edited by O. Hirota, A. S. Holevo, and C. M. Caves (Springer, Boston, MA, 1997), Vol. 1.
[75] Y. Li, X. Chen, and M. P. A. Fisher, Quantum Zeno effect and the many-body entanglement transition, Phys. Rev. B **98**, 205136 (2018).
[76] A. Chan, R. M. Nandkishore, M. Pretko, and G. Smith, Unitary-projective entanglement dynamics, Phys. Rev. B **99**, 224307 (2019).
[77] J. Côté and S. Kourtis, Entanglement phase transition with spin glass criticality, Phys. Rev. Lett. **128**, 240601 (2022).
[78] S. P. Kelly, U. Poschinger, F. Schmidt-Kaler, M. Fisher, and J. Marino, Coherence requirements for quantum communication from hybrid circuit dynamics, SciPost Phys. **15**, 250 (2023).
[79] A. Rényi, On measures of entropy and information, in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, edited by J. Neyman (University of California Press, 1961), Vol. 4.1, pp. 547–562.
[80] I. Bengtsson and K. Życzkowski, *Geometry of Quantum States: An Introduction to Quantum Entanglement* (Cambridge University Press, Cambridge, MA, 2017).
[81] L. Leone, S. F. E. Oliviero, and A. Hamma, Stabilizer Fényi entropy, Phys. Rev. Lett. **128**, 050402 (2022).
[82] X.-Z. Luo, J.-G. Liu, P. Zhang, and L. Wang, Yao.jl: Extensible, efficient framework for quantum algorithm design, Quantum **4**, 341 (2020).
[83] S. Krämer, D. Plankensteiner, L. Ostermann, and H. Ritsch, QuantumOptics.jl: A Julia framework for simulating open quantum systems, Comput. Phys. Commun. **227**, 109 (2018).
[84] P. Díez-Valle, D. Porras, and J. J. García-Ripoll, Quantum variational optimization: The role of entanglement and problem hardness, Phys. Rev. A **104**, 062426 (2021).
[85] M. Dupont, N. Didier, M. J. Hodson, J. E. Moore, and M. J. Reagor, Entanglement perspective on the quantum approximate optimization algorithm, Phys. Rev. A **106**, 022423 (2022).
[86] C. Qian, W.-F. Zhuang, R.-C. Guo, M.-J. Hu, and D. E. Liu, Information scrambling and entanglement in quantum approximate optimization algorithm circuits, Eur. Phys. J. Plus **139**, 14 (2024).
[87] S. Sahni and T. Gonzalez, P-complete approximation problems, J. ACM **23**, 555 (1976).
[88] S. Kahruman, E. Kolotoglu, S. Butenko, and I. V. Hicks, On greedy construction heuristics for the MAX-CUT problem, Int. J. Comput. Sci. Eng. **3**, 211 (2007).
[89] J. Wang, C. Wu, and F. Zuo, More on greedy construction heuristics for the MAX-CUT problem, arXiv:2312.10895.
[90] https://github.com/manuelmz/adapt-clifford.
[91] F. Alizadeh, Interior point methods in semidefinite programming with applications to combinatorial optimization, SIAM J. Opt. **5**, 13 (1995).
[92] D. Sherrington and S. Kirkpatrick, Solvable model of a spin-glass, Phys. Rev. Lett. **35**, 1792 (1975).
[93] K. Binder and A. P. Young, Spin glasses: Experimental facts, theoretical concepts, and open questions, Rev. Mod. Phys. **58**, 801 (1986).
[94] D. Panchenko, The Sherrington-Kirkpatrick model: An overview, J. Stat. Phys. **149**, 362 (2012).
[95] D. Panchenko, *The Sherrington-Kirkpatrick Model* (Springer Science & Business Media, New York, NY, 2013).
[96] P. Charbonneau, E. Marinari, M. Mézard, G. Parisi, F. Ricci-Tersenghi, G. Sicuro, and F. Zamponi, *Spin Glass Theory and Far Beyond* (World Scientific, Singapore, 2023).
[97] G. Parisi, Infinite number of order parameters for spin-glasses, Phys. Rev. Lett. **43**, 1754 (1979).
[98] M. Mezard, G. Parisi, and M. Virasoro, *Spin Glass Theory and Beyond* (World Scientific, Singapore, 1986).
[99] M. J. Schmidt, Replica symmetry breaking at low temperatures, Ph.D. thesis, Universität Würzburg (2008).
[100] F. Guerra, Broken replica symmetry bounds in the mean field spin glass model, Commun. Math. Phys. **233**, 1 (2003).

[101] M. Talagrand, The Parisi formula, Ann. Math. **163**, 221 (2006).

[102] E. Farhi, J. Goldstone, S. Gutmann, and L. Zhou, The quantum approximate optimization algorithm and the Sherrington-Kirkpatrick model at infinite size, Quantum **6**, 759 (2022).

[103] M. Dupont and B. Sundar, Extending relax-and-round combinatorial optimization solvers with quantum correlations, Phys. Rev. A **109**, 012429 (2024).

[104] A. Montanari, Optimization of the Sherrington-Kirkpatrick Hamiltonian, in *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, Baltimore, MD, 2019), pp. 1417–1433.

[105] A. El Alaoui, A. Montanari, and M. Sellke, Optimization of mean-field spin glasses, Ann. Probab. **49**, 2922 (2021).

[106] The run time of this algorithm is $\mathrm{c}(\epsilon)O(N^2)$ with $\mathrm{c}(\epsilon)$ an inverse polynomial of ϵ.

[107] We point out that formal proof of the scaling $\mathbb{E}[\frac{E^{\mathrm{sk}}_{\mathrm{min}}}{N}] \sim N^{-2/3}$ exists only above the critical temperature, but it is believed that it holds for the whole spin glass phase. This assumption is supported with extensive numerical results; see, for instance, Ref. [120,121] and references therein. This is why we feel confident in its use in our model to fit the data.

[108] M. Aizenman, J. L. Lebowitz, and D. Ruelle, Some rigorous results on the Sherrington-Kirkpatrick spin glass model, Commun. Math. Phys. **112**, 3 (1987).

[109] A. Montanari and S. Sen, Semidefinite programs on sparse random graphs and their application to community detection, in *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, NY, 2016), pp. 814–827.

[110] G. Weitz, L. Pira, C. Ferrie, and J. Combes, Sub-universal variational circuits for combinatorial optimization problems, arXiv:2308.14981.

[111] F. G. Fuchs, Optimal mixers restricted to subspaces and the stabilizer formalism, arXiv:2306.17083.

[112] D. J. Egger, J. Mareček, and S. Woerner, Warm-starting quantum optimization, Quantum **5**, 479 (2021).

[113] G. S. Ravi, P. Gokhale, Y. Ding, W. Kirby, K. Smith, J. M. Baker, P. J. Love, H. Hoffmann, K. R. Brown, and F. T. Chong, Cafqa: A classical simulation bootstrap for variational quantum algorithms, in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (Association for Computing Machinery, New York, NY, 2022), Vol. 1, pp. 15–29.

[114] https://github.com/ericproffitt/maxcut.jl.

[115] C. Papadimitriou and M. Yannakakis, Optimization, approximation, and complexity classes, in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, NY, 1988), pp. 229–234.

[116] I. Dunning, S. Gupta, and J. Silberholz, What works best when? A systematic evaluation of heuristics for Max-Cut and QUBO, INFORMS J. Comput. **30**, 608 (2018).

[117] D. Gamarnik, C. Moore, and L. Zdeborová, Disordered systems insights on computational hardness, J. Stat. Mech.: Theory Exp. (2022) 114015.

[118] F. G. Brandao, M. Broughton, E. Farhi, S. Gutmann, and H. Neven, For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances, arXiv:1812.04170.

[119] D. Amaro, C. Modica, M. Rosenkranz, M. Fiorentini, M. Benedetti, and M. Lubasch, Filtering variational quantum algorithms for combinatorial optimization, Quantum Sci. Technol. **7**, 015021 (2022).

[120] C. Giberti and C. Vernia, Numerical study of ground state energy fluctuations in spin glasses, arXiv:0807.1197.

[121] M. Palassini, Ground-state energy fluctuations in the Sherrington-Kirkpatrick model, J. Stat. Mech.: Theory Exp. (2008) P10005.