# Benchmark Study of Quantum Algorithms for Combinatorial Optimization: Unitary versus Dissipative

Krishanu Sankar,[1] Artur Scherer,[2] Satoshi Kako,[3] Sam Reifenstein,[3] Navid Ghadermarzy,[1] Willem B. Krayenhoff,[1] Yoshitaka Inui,[3] Edwin Ng,[3,4] Tatsuhiro Onodera,[3,5] Pooya Ronagh,[1,6,7,*] and Yoshihisa Yamamoto[3,*]

[1] *1QB Information Technologies (1QBit), Vancouver, BC, Canada*
[2] *1QB Information Technologies (1QBit), Waterloo, ON, Canada*
[3] *Physics & Informatics Laboratories, NTT Research Inc, Sunnyvale, CA, USA*
[4] *E. L. Ginzton Laboratory, Stanford University, Stanford, CA, USA*
[5] *School of Applied and Engineering Physics, Cornell University, Ithaca, NY, USA*
[6] *Institute for Quantum Computing, University of Waterloo, Waterloo, ON, Canada*
[7] *Department of Physics & Astronomy, University of Waterloo, Waterloo, ON, Canada*

(Dated: May 11, 2021)

We study the performance scaling of three quantum algorithms for combinatorial optimization: measurement-feedback coherent Ising machines (MFB-CIM), discrete adiabatic quantum computation (DAQC), and the Dürr–Høyer algorithm for quantum minimum finding (DH-QMF) that is based on Grover's search. We use MaxCut problems as our reference for comparison, and time-to-solution (TTS) as a practical measure of performance for these optimization algorithms. We empirically observe a $\Theta(2^{\sqrt{n}})$ scaling for the median TTS for MFB-CIM, in comparison to the exponential scaling with the exponent $n$ for DAQC and the provable $\widetilde{\mathcal{O}}\left(\sqrt{2^n}\right)$ scaling for DH-QMF. We conclude that these scaling complexities result in a dramatic performance advantage for MFB-CIM in comparison to the other two algorithms for solving MaxCut problems.

## I. INTRODUCTION

Combinatorial optimization problems are ubiquitous in modern science, engineering, and medicine. These problems are often NP-hard, so the runtime of classical algorithms for solving them is expected to scale exponentially. One approach for tackling such hard optimization problems is to map them to the Ising spin glass model [1],

$$\mathcal{H} = -\sum_{i<j} J_{ij} S_i S_j - \sum_i h_i S_i.$$

Here, each $S_i$ represents a classical Ising spin attaining a value of $\pm 1$, $[J_{ij}]$ is an Ising coupling matrix and $[h_i]$ is a vector of local field biases on the spin sites. When all $h_i$ are zero, the Ising model is equivalent to a (weighted) MaxCut problem on a graph with vertices corresponding to the spin sites and edge weights corresponding to the Ising couplings between the spin sites. Various mathematical programming problems, such as partitioning problems, binary integer linear programming, covering and packing problems, satisfiability problems, colouring problems, Hamiltonian cycles, tree problems, and graph isomorphisms can be formulated in the Ising model, with the required number of spins scaling at most cubically with respect to the problem size [2]. This has been a primary motivation for the recent extensive study of various Ising solvers. Several potential areas of industrial application of Ising solvers include drug discovery and biocatalyst development (e.g., in lead optimization or virtual screening), compressed sensing, deep learning (e.g., in the synaptic pruning of deep neural network), scheduling (e.g., resource allocation and traffic control), computational finance, and social networks (e.g., community detection).

Approximate algorithms and heuristics, such as semidefinite programming (SDP) [3], simulated annealing (SA) [4, 5] and its variants [6, 7], and breakout local search (BLS) [8] have been widely used as practical tools for solving MaxCut problems. However, even problem instances of moderate size require substantial computation time and, in the worst cases, solutions cannot be found with such approximate algorithms and heuristics. To overcome these shortcomings, a search for alternative solutions using various forms of quantum computing has been actively pursued. Adiabatic quantum computation [9], quantum annealing [10, 11], and the quantum approximate optimization algorithm (QAOA) [12] using circuit model quantum computers have been proposed. A coherent Ising machine (CIM) using networks of quantum optical oscillators has also been studied and implemented [13, 14].

Given that the present circuit model quantum computers suffer from short coherence times, gate errors, and limited connectivity among qubits, a fair comparison between them and modern heuristics is not yet possible [15–17]. This situation raises the important question of whether quantum devices can, even in principle, provide sensible solutions to combinatorial optimization problems, assuming all sources of noise and imperfections can be overcome and ideal quantum processors are built in the future. In order to address this pressing question, we perform a comparative numerical study on three dis-

* Corresponding authors:
pooya.ronagh@1qbit.com, and
yoshihisa.yamamoto@ntt-research.com

tinct quantum approaches, ignoring the effects of noise, gate errors, and decoherence, that is, we compare the ultimate theoretical limits of three quantum approaches.

The first approach is based on the effects of constructive and destructive quantum interference of amplitudes in a circuit model quantum computer that utilizes only unitary evolution of pure states and projective (exact) measurement of qubits. The approach uses Grover's search algorithm [18, 19] as a key computational primitive. We call this approach "DH-QMF" in reference to Dürr and Høyer's "quantum minimum finding" algorithm [20].

The second approach is based on adiabatic quantum state preparation implemented on a circuit model quantum computer. The underlying concept, the quantum adiabatic theorem, goes back as far as to the seminal work by Born and Fock [21]. Its application to quantum computing and solving optimization problems was introduced by Farhi et al. [9]. A Trotterized approximation to adiabatic evolution gives rise to a discrete implementation suitable for the circuit model. We refer to this approach as "discrete adiabatic quantum computation" (DAQC). A variant of DAQC is the "quantum approximate optimization algorithm" (QAOA) [12, 22]. This algorithm uses an iterative unitary evolution of pure states in a quantum circuit according to a mixing Hamiltonian and a problem Hamiltonian, which in the framework of adiabatic quantum computation correspond to the initial and final Hamiltonians of evolution, respectively. QAOA is considered a promising candidate for solving combinatorial optimization problems on noisy, intermediate-scale quantum (NISQ) devices (although its original formulation is not restricted to implementations on such devices). As such, QAOA has become associated with NISQ-type algorithms, that is, it is commonly characterized by its use of shallow quantum circuits of short depth along with a method for optimizing the set of parameters specifying the unitary gates in those circuits. Motivated by our interest in exploring the capabilities of QAOA for solving optimization problems, we study two schemes for optimizing its quantum gate parameters. The first scheme treats gate parameters as hyperparameters that follow a tuned schedule for a Trotterized adiabatic evolution, very much in accordance with the ordinary DAQC approach. The second scheme uses a variational hybrid quantum–classical protocol to optimize the gate parameters. We find a performance advantage for the tuned adiabatic schedules of DAQC over the variational method commonly used in hybrid QAOA schemes. For this reason, we use pre-tuned DAQC schedules for our benchmarking analysis. Moreover, to obtain the ultimate theoretical performance limit (for a fair comparison with the other approaches), we drop the requirement of having to use only low-depth circuits that are necessary in the case of NISQ devices. That is, our reported benchmarking results pertain to the implementation of DAQC using quantum circuits of arbitrary depth.

The third approach is based on a measurement-feedback coherent Ising machine (MFB-CIM) [23, 24]. This algorithm utilizes a quantum-to-classical transition in an open-dissipative, non-equilibrium network of quantum oscillators. A critical phenomenon known as pitchfork bifurcation realizes the transition of squeezed vacuum states to coherent states in the optical parametric oscillator. The measurement-feedback circuit plays several important roles. It continually reduces entropy and sustains a quasi-pure state in the quantum oscillator network in a controlled manner using repeated approximate measurements. It, additionally, implements the Ising coupling matrix $[J_{ij}]$ and local field vector $[h_i]$ in an iterative fashion. Finally, it removes the amplitude heterogeneity among the oscillators and destabilizes the machine state out of local minima. Table I summarizes the differences among the three approaches studied in this paper.

When studying quantum algorithms, it is important to consider the effect of noise and control errors, and the overhead needed to overcome them. Several previous studies have been investigating these effects on the performance of QAOA (here viewed as a NISQ-type variant of DAQC). In references [25, 26], various Pauli noise channels, namely the dephasing, bit-flip, and the depolarizing noise channels, are considered. These two papers report on the fidelity of the state prepared by a noisy QAOA circuit to the state prepared by an ideal QAOA circuit, for varying amounts of physical noise affecting the circuit. In contrast, [22] models noise via single-qubit rotations by an angle chosen from a Gaussian distribution with variance values of $T_{\rm G}/T_2$, where $T_{\rm G}$ is the gate time and $T_2$ is the decoherence time of the qubits. All three papers provide results on how noise affects the expected energy of the prepared state.

DH-QMF circuits are much deeper than QAOA circuits and, therefore, their performance is significantly hampered by various sources of noise unless the algorithm is run on a fault-tolerant quantum computer with quantum error correction [27–34]. Different noise models have been used to study the sensitivity of Grover's search by simulating small quantum circuits that apply it to simple functions. [27] introduces random Gaussian noise on each step of Grover's search. [28] studies the effect of gate imperfections on the probability of success of the algorithm. [31] examines the effect of unbiased and isotropic unitary noise resulting from small perturbations of Hadamard gates. [29] models the effect of decoherence by introducing phase errors in each qubit and time step and using a perturbative method. [32] conducts a numerical analysis on the effects of single-qubit and two-qubit gate errors and memory errors, modelling decoherence using a depolarizing channel. The impact of using a noisy oracle is examined in [30], wherein noise is modelled by introducing random phase errors. The effects of localized dephasing are studied in [34]. Finally, [33] investigates the effects of various noise channels using trace-preserving, completely positive maps applied to density matrices.

| | DH-QMF | DAQC | MFB-CIM |
|---|---|---|---|
| Quantum dynamics | Closed-unitary | Closed-unitary | Open-dissipative |
| Operational principle | Amplitude amplification by quantum interference | Adiabatic quantum evolution | Quantum-to-classical transition |
| Information carrier | Digital (spin-1/2 particle) | Digital (spin-1/2 particle) | Analog (harmonic oscillator) |
| Decoherence time | $T_2 \to \infty$ | $T_2 \to \infty$ | $T_2 \to \infty$ |
| Dissipation time | $T_1 \to \infty$ | $T_1 \to \infty$ | $T_1$ : finite |
| Gate error | None | None | Vacuum noise limited |
| Spin-spin coupling | all-to-all | all-to-all | all-to-all |

TABLE I: Three approaches studied for MaxCut problems: the Dürr–Høyer algorithm for quantum minimum finding (DH-QMF) based on Grover's search, the discretized adiabatic quantum computation algorithm (DAQC), and the measurement-feedback coherent Ising machine (MFB-CIM).

We have evaluated the wall-clock time-to-solution (TTS) of the three algorithms introduced above for solving MaxCut problems, and empirically found exponential scaling laws for them already in the relatively small problem size range of 4 to 800 spins. In order to elucidate the ultimate performance limits of these solvers, we assume no extrinsic noise, gate errors, or connectivity limitations exist in the hardware. That is, we assume that phase decoherence ($T_2$) and energy dissipation ($T_1$) times are infinite and gate errors are absent. Consequently, there is no overhead arising from the need to perform quantum error correction and to build fault-tolerant architectures. We also assume that all spins (represented by qubits in the circuit model) can be coupled to each other via (non-local) spin–spin interaction with a universal gate time of 10 nanoseconds. Therefore, there is no need to implement expensive sequences of swap gates or other bus techniques for transferring quantum information across the hardware. However, since energy dissipation and stochastic noise both constitute important computational resources for the MFB-CIM, we allow a finite energy dissipation time $T_1$, as well as a finite gate error limited by vacuum noise, for the MFB-CIM.

From a fundamental viewpoint, such a comparative study is of interest but the outcome is difficult to predict, because the three algorithms are based on completely different computational principles, as shown in Table I. The DH-QMF algorithm iteratively deploys Grover's search, which uses a unitary evolution of a superposition of computation basis states in order to amplify the amplitude of a target state by successive constructive interference, while the amplitudes of all the other states are attenuated by destructive interference. The DAQC algorithm attempts to prepare a pure state that has a large overlap with the ground state of the optimization problem through an approximation of the adiabatic quantum evolution. Finally, the ground state search mechanism of the MFB-CIM employs a collective phase transition at the threshold of an optical parametric oscillator (OPO) network. The correlations formed among the squeezed vacuum states in OPOs below the threshold guide the network toward oscillating at a ground state.

It is worth noting that all the algorithms in our study in various ways rely on hybrid quantum–classical architectures for computation. In a closed-loop CIM with self-diagnosis and dynamical feedback control, a classical processor plays an important role by detecting when the OPO network is trapped in local minima, and destabilizes it out of those states. The DH-QMF algorithm also relies on comparing the values of an objective function with a (classical) threshold value. This threshold value is updated in a classical coprocessor as DH-QMF proceeds. Finally, DAQC relies on tuning a set of parameters (e.g., the rotation angles of quantum gates). These parameters can be treated as hyperparameters of a predefined approximate adiabatic evolution and tuned for the problem type solved by the algorithm. Alternatively, the quantum circuit can be viewed as a variational ansatz, in which case the gate parameters are optimized using a classical optimizer. In the latter case, the algorithm can be considered as a variational quantum algorithm [35]. QAOA is commonly viewed as such an algorithm. In previous studies, the contribution of the variational optimization of QAOA parameters to the TTS has often been ignored. In fact, while both approaches (i.e., hyperparameter tuning and variational optimization) have been adopted for solving MaxCut problems using QAOA [36, 37], our investigation makes it clear that the variational approach hurts the TTS scaling significantly. The optimization landscape for such a variational quantum algorithm is ill-behaved, which results in a poor and unstable scaling for TTS with respect to the size of the MaxCut

instances (refer to Appendix C). As a result, the TTS scalings reported in this paper rely on pre-tuned DAQC schedules rather than variational optimization.

## II. SCALING OF THE MFB-CIM

A CIM is a non-equilibrium, open-dissipative computing system based on a network of degenerate OPOs to find a ground state of Ising problems [13, 38–41]. The Ising Hamiltonian is mapped to the loss landscape of the OPO network formed by the dissipative coupling rather than the standard Hamiltonian coupling. By providing a sufficient gain to compensate for the overall network loss, a ground state of the target Hamiltonian is expected to build up spontaneously as a single oscillation mode [14]. However, the mapping of the cost function to the OPO network loss landscape often fails in the case of a frustrated spin problem due to the OPO amplitude inhomogeneity [13, 23]. In addition, with an increasing number of local minima occurring as problem sizes become larger, the machine state is trapped in those minima for a substantial amount of time, thereby causing the machine to report suboptimal solutions [14, 24]. Recently, self-diagnosis and dynamical feedback mechanisms have been introduced by a measurement-feedback CIM (MFB-CIM) to overcome these problems [23, 24]. This is achieved by a mutual coupling field dynamically modulated for each OPO to suppress the amplitude inhomogeneity and simultaneously to destabilize the machine's state out of local minima.

### A. Principle of Operation

A schematic diagram of two MFB-CIMs with predefined feedback control (hereafter referred to as "open-loop CIM") and with self-diagnosis and dynamical feedback control (hereafter referred to as "closed-loop CIM"), is shown in Fig. 1 (a). If the fibre ring resonator has high finesse, both CIMs are modelled via the Gaussian quantum theory [42, 43]. The dynamics captured by the master equation for the density operator (i.e., the Liouville–von Neumann equation) is driven by the parametric interaction Hamiltonian, $\hat{\mathcal{H}} = i\hbar\frac{S}{2}\sum_i\left(\hat{a}_i^{\dagger 2} - \hat{a}_i^2\right)$, the measurement-induced state reduction (the third term on the right-hand side in Eq. (1)), the coherent injection (the fourth term on the right-hand side in Eq. (1)), as well as three Liouvillians. The Liouvillians pertain to the linear loss due to measurement and injection couplings, $\hat{\mathcal{L}}_{\rm c}^{(i)} = \sqrt{J}\hat{a}_i$, two-photon absorption loss (i.e., parametric back conversion) in a degenerate parametric amplifying device, $\hat{\mathcal{L}}_2^{(i)} = \sqrt{B/2}\,\hat{a}_i^2$, and background linear losses, $\hat{\mathcal{L}}_1^{(i)} = \sqrt{\gamma_{\rm s}}\hat{a}_i$, respectively [43]. The master equation is thus given by

$$\frac{d}{dt}\hat{\rho} = -\frac{i}{\hbar}\left[\hat{\mathcal{H}},\hat{\rho}\right] + \sum_{i=1}^{n}\sum_{k=1,2,{\rm c}}\left(\left[\hat{\mathcal{L}}_k^{(i)},\hat{\rho}\hat{\mathcal{L}}_k^{(i)\dagger}\right] + h.c.\right) + \sqrt{J}\sum_{i=1}^{n}\left(\hat{a}_i\hat{\rho} + \hat{\rho}\hat{a}_i^\dagger - \langle\hat{a}_i + \hat{a}_i^\dagger\rangle\hat{\rho}\right)w_i + \frac{J}{2}\sum_{i,k=1}^{n} e_i(t)J_{ik}\left(\langle\hat{a}_k + \hat{a}_k^\dagger\rangle + \frac{w_k}{\sqrt{J}}\right)[\hat{a}_i^\dagger - \hat{a}_i,\hat{\rho}]. \quad (1)$$

In general, the numerical integration of Eq. (1) requires exponentially growing resources as the problem size $n$ (i.e., the number of spins) increases. Generally speaking, the size of the density matrix scales as $\mathcal{O}({n_0}^n \times {n_0}^n)$, where $n_0 \gg 1$ is the maximum number of photons possible for each OPO pulse. In MFB-CIMs, however, there is no entanglement between the OPO pulses, that is, the OPO states are separable. Therefore, the simulation's memory requirements reduce to $\mathcal{O}(n \times {n_0}^2)$. However, this reduction still yields too many c-number differential equations due to the large upper bounds on the number of photons $n_0 \lesssim 10^7$ and the number of spins $n \leq 1000$. The Gaussian quantum model has been introduced to overcome this difficulty [24, 43].

In the case of a small saturation parameter, $g^2 = B/\gamma_{\rm s} \ll 1$, we can split the $i$-th OPO's pulse amplitude operator, $\hat{a}_i = \frac{1}{\sqrt{2}}(\hat{X}_i + i\hat{P}_i)$, into the mean field and small fluctuation operators, $\hat{X}_i = \langle\hat{X}_i\rangle + \Delta\hat{X}_i$ and $\hat{P}_i = \langle\hat{P}_i\rangle + \Delta\hat{P}_i$. The saturation parameter $g^2$ corresponds to the inverse photon number at twice the threshold pump rate of a solitary OPO. With an appropriate choice of the pump phase, each OPO mean-field is generated only in an $\hat{X}$-quadrature, that is, $\langle\hat{P}_i\rangle = 0$. The equation of motion for the mean field $\mu_i = \langle\hat{X}_i\rangle/\sqrt{2}$ and the variances $\sigma_i = \langle\Delta\hat{X}_i^2\rangle$ and $\eta_i = \langle\Delta\hat{P}_i^2\rangle$ obey the following equations [43]:

$$\frac{d}{dt}\mu_i = \left[-(1+j)+p-g^2\mu_i^2\right]\mu_i + je_i(t)\sum_k J_{ik}\tilde{\mu}_k + \sqrt{j}\,(\sigma_i - 1/2)\,w_i\,, \tag{2}$$

$$\frac{d}{dt}\sigma_i = 2\left[-(1+j)+p-3g^2\mu_i^2\right]\sigma_i - 2j(\sigma_i-1/2)^2 + \left[(1+j)+2g^2\mu_i^2\right], \tag{3}$$

$$\frac{d}{dt}\eta_i = 2\left[-(1+j)-p-g^2\mu_i^2\right]\eta_i + \left[(1+j)+2g^2\mu_i^2\right]. \tag{4}$$

Here, $t = \gamma_s T$ is a normalized and dimensionless time, where $T$ is physical (or wall-clock) time, and $\gamma_s$ is the loss rate of the cavity. The time $t$ is normalized so that the background linear loss (with a signal amplitude decay rate of $1/e$) is 1. The term $-(1+j)$ in Eqs. (2) to (4) represents a background linear loss ($-1$) and an out-coupling loss ($-j$) for optical homodyne measurement and feedback injection, where $j = J/\gamma_s$ is a normalized out-coupling rate (see Fig. 1(a)). The parameter $p = S/\gamma_s$ is a normalized linear gain coefficient provided by the parametric device. The term $g^2\mu_i^2$ represents two-photon absorption loss (i.e., back conversion from signal to pump fields). The second and third terms on the right-hand side of Eq. (2), respectively, represent the Ising coupling term and the measurement-induced shift of the mean-field $\mu_i$. The inferred mean-field amplitude, $\tilde{\mu}_k = \mu_k + \sqrt{\frac{1}{4j}}w_k$, deviates from the internal mean-field amplitude $\mu_k$ by a finite measurement uncertainty in the optical homodyne detection. The random variable $w_k\sqrt{\Delta t}$ attains values drawn from the standard normal distribution, where $\Delta t$ is a time step for the numerical integration of Eqs. (2) to (4). The $k$-th Ising spin $S_k = \pm 1$ is determined by the sign of the inferred mean-field amplitude, $S_k = \tilde{\mu}_k/|\tilde{\mu}_k|$. $J_{ik}$ is the Ising coupling coefficient and $e_i(t)$ is a dynamically modulated feedback-field amplitude. The second term on the right-hand side of Eq. (3) represents the measurement-induced partial state reduction of the OPO field. The last terms of Eqs. (3) and (4), respectively, represent the variance increase by the incident (fresh) vacuum field fluctuations via linear loss and the pump noise coupled to the OPO field via gain saturation.

The dynamically modulated feedback-field amplitude $e_i(t)$ is introduced to reduce the amplitude inhomogeneity [23], which is determined by the inferred signal amplitude $\tilde{\mu}_i$:

$$\frac{d}{dt}e_i(t) = -\beta\left[g^2\tilde{\mu}_i^2 - a\right]e_i(t). \tag{5}$$

Here, $\beta$ is a positive constant representing the rate of change for the exponentially growing or attenuating feedback amplitude $e_i(t)$, and $a$ is a target squared amplitude. Both $a$ and the pump rate $p$ are dynamically determined by the difference of the current Ising energy $\mathcal{E}(t) = -\sum_{i<k} J_{ik}S_iS_k$ and the lowest Ising energy $\mathcal{E}_{\rm opt}$ visited previously:

$$a(t) = \alpha + \rho_a \tanh\left(\frac{\mathcal{E}(t)-\mathcal{E}_{\rm opt}}{\Delta}\right), \tag{6}$$

$$p(t) = \pi - \rho_p \tanh\left(\frac{\mathcal{E}(t)-\mathcal{E}_{\rm opt}}{\Delta}\right). \tag{7}$$

Here, $\pi$, $\alpha$, $\rho_a$, $\rho_p$, and $\Delta$ are predetermined positive parameters which characterize the self-diagnosis and dynamic feedback control.

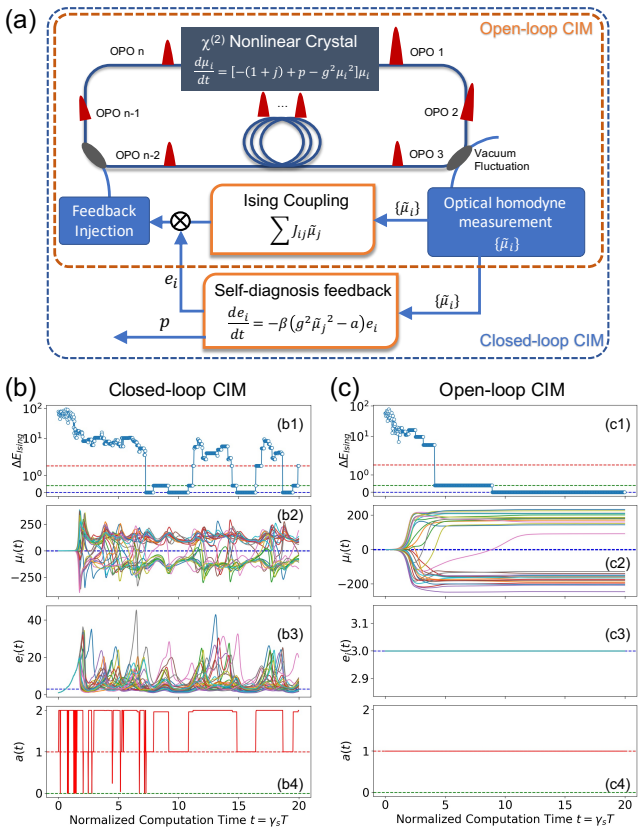



FIG. 1: (a) Schematic diagram of the measurement-feedback coupling CIMs with and without the self-diagnosis and dynamic feedback control (closed-loop and open-loop CIMs) indicated using dashed blue and orange lines, respectively. (b) and (c) Dynamical behaviour of the closed-loop and open-loop CIMs, respectively. (b1) and (c1) Inferred Ising energy (the dashed horizontal lines are the lowest three Ising eigen-energies). (b2) and (c2) Mean-field amplitude $\mu_i(t)$. (b3) and (c3) Feedback-field amplitude $e_i(t)$. (b4) Target squared amplitude $a(t)$. (c4) Pump rate $p(t)$.

The machine can distinguish the following three modes of operation from the energy measurements. When $\mathcal{E}(t) - \mathcal{E}_{\rm opt} < -\Delta$, the machine is in a gradient descent mode and moving toward a local minimum, in which case the pump is set to a positive value of $\pi + \rho_p$ (leading to parametric amplification). When $|\mathcal{E}(t) - \mathcal{E}_{\rm opt}| \ll \Delta$, the machine is close to, or trapped in, a local minimum, in which case the pump is switched off (i.e., there is no parametric amplification) so as to destabilize the current spin configuration. When $\mathcal{E}(t) - \mathcal{E}_{\rm opt} > \Delta$, the machine is attempting to escape from a previously visited local minimum, in which case the pump is set to a negative value of $\pi - \rho_p$ (i.e., there is parametric de-amplification) to increase the rate of spin flips.

Fig. 1(b) shows the time evolution of a closed-loop CIM to demonstrate its inherent exploratory behaviour from one local minimum to another. We solve a MaxCut problem with randomly generated discrete edge-weights $J_{ij} \in \{-1, -0.9, \ldots, 0.9, 1\}$ over $n = 30$ vertices, for which an exact solution is obtained by performing an exhaustive search. The dynamical behaviour of the inferred Ising energy measured from the ground state energy, $\Delta\mathcal{E}(t) = \mathcal{E}(t) - \mathcal{E}_{\rm G}$, the mean amplitude, $\mu(t)$, the feedback-field amplitude, $e(t)$, and the target squared amplitude, $a(t)$, are shown in Fig. 1 (b) and (c). The results shown in Fig. 1(b) are taken from a single trial for one particular problem instance and a particular set of noise amplitudes $w_i\sqrt{\Delta t}$. The feedback parameters are set to $\alpha = 1.0$, $\pi = 0.2$, $\rho_a = \rho_p = 1.0$, $\Delta = 1/5$, and $\beta = 1.0$ [24]. The saturation parameter is chosen as $g^2 = 10^{-4}$. The time step $\Delta t$ for the numerical integration of Eqs. (2) to (4) is identical to the normalized round-trip time $\Delta t_{\rm c} = \gamma_{\rm s}\Delta T_{\rm c} = 0.025$. This means the signal-field lifetime $1/\gamma_{\rm s}$ is 40 times greater than the round-trip time.

As shown in Fig. 1(b1), the inferred Ising energy $\mathcal{E}(t)$ fluctuates up and down during the search for a solution even after the machine finds one of the degenerate ground states. As shown in Fig. 1(b2), the measured squared amplitude $g^2\tilde{\mu}_i{}^2$ is stabilized to the target squared amplitude $a$ through the dynamically modulated feedback mean-field $e_i(t)$. Several OPO amplitudes, however, flipped their signs followed by an exponential increase in $e_i(t)$, while most other OPOs maintained a target amplitude. During this spin-flip process, the feedback-field amplitude $e_i(t)$ increases exponentially and then decreases exponentially after the OPO's squared amplitude $g^2\tilde{\mu}_i^2$ exceeds the target squared amplitude $a(t)$. The mutual coupling strength $\sum_k J_{ik}\tilde{\mu}_k$ is adjusted in order to decrease the energy continuously by flipping the "wrong" spins and preserving the "correct" ones. If the machine reaches local minima, which may also include global minima (in which case there are degenerate ground states), the current Ising energy $\mathcal{E}(t) = -\sum_{i<k} J_{ik}S_iS_k$ is roughly equal to the minimum Ising energy $\mathcal{E}_{\rm opt}$ previously visited ($\mathcal{E}(t) \simeq \mathcal{E}_{\rm opt}$). The machine then decreases the target squared amplitude $a$, which helps it to escape from the local minimum. During this escape, the current Ising energy $\mathcal{E}(t)$ becomes greater than the minimum Ising energy $\mathcal{E}_{\rm opt}$. The machine then switches the pump rate $p$ to a negative value and deamplifies the signal amplitude, which results in further destabilization of the local minimum. As a consequence of such dynamical modulation of the pump rate $p$ and the target squared amplitude $a$, the machine continually escapes local minima, migrating from one local minimum to another as the computation carries on.

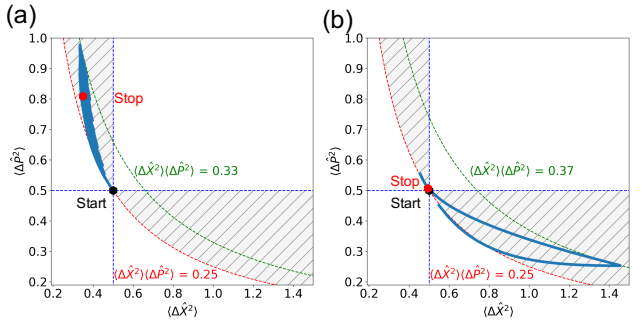



FIG. 2: Variances $\langle\Delta\hat{X}^2\rangle$ and $\langle\Delta\hat{P}^2\rangle$ for (a) a closed-loop CIM and (b) an open-loop CIM. The shaded areas show the quantum domains ($\langle\Delta\hat{X}^2\rangle < 1/2$ or $\langle\Delta\hat{P}^2\rangle < 1/2$). Note that these are the results for one particular OPO, i.e., for one of the trajectories shown in Fig. 1(b) and (c).

Fig. 1(c) shows the time evolution of an open-loop CIM, in which both the pump rate $p$ and the feedback-field amplitude $e_i(t)$ are predetermined constants.

As shown in Fig. 2(a) and (b), the quantum states of the OPO fields satisfy the minimum uncertainty product, $\langle\Delta\hat{X}^2\rangle\langle\Delta\hat{P}^2\rangle = 1/4$, with a small excess factor of $\sim 30\%$ despite the open-dissipative nature of the machine. We note that each OPO state is in a quantum domain ($\langle\Delta\hat{X}^2\rangle < 1/2$ or $\langle\Delta\hat{P}^2\rangle < 1/2$), which is shown by the shaded area in Fig. 2. This is a consequence of the repeated homodyne measurements performed during the computation, which iteratively reduces the entropy in the machine and partially collapses the OPO state such that it comes close to being a minimum-uncertainty state. In a closed-loop CIM, parametric amplification with a positive pump rate ($p > 0$) is employed only in the initial stage, but parametric deamplification with a negative pump rate ($p < 0$) is used later on. The resulting squeezing ($\langle\Delta\hat{X}^2\rangle < 1/2$) rather than anti-squeezing ($\langle\Delta\hat{X}^2\rangle > 1/2$) is favourable for exploration when using repetitive spin flips. In contrast, parametric amplification with a positive pump rate is used in an open-loop CIM throughout the computation.

### B. Time-to-Solution

Figures 3(a) and (b) show the median of the success probability $P_{\rm s}$ and time-to-solution (TTS) $t_{\rm s}$ of the closed-loop CIM as a function of problem size $n = 4, 5, \ldots, 30$ with varying runtime $t_{\rm max}$. We perform

1000 trials, with a trial considered successful if the machine finds an exact solution within $t_{\max}$. The success probability $P_s$ decreases exponentially with respect to $n$, especially for $t_{\max} \leq 5$. For a greater value of $t_{\max}$, the slope of the decay improves as shown in Fig. 3(a). The TTS is defined as the expected computation time required to find a ground state for a particular problem instance with 99% confidence. As such, it is defined via

$$t_s = R_{99} \cdot t_{\max}\,, \tag{8}$$

where $R_{99} = \frac{\log(0.01)}{\log(1-P_s)}$ is the number of trials required to achieve a 99% probability of success. We solve 1000 instances for each problem size ($n = 4, \ldots, 30$) to evaluate the median $P_s$ and TTS. Note that $t_s$ refers to the *normalized* and dimensionless TTS, while the actual wall-clock TTS (in seconds) is denoted by $T$. These two notions of TTS are related via the equation $t_s = \gamma_s T$. The wall-clock time $T$ is estimated by assuming a cavity round-trip time of $\Delta T_c = 10$ nanoseconds (all-to-all spin coupling is implemented in 10 nanoseconds), and a $1/e$ signal amplitude decay time of 400 nanoseconds ($\gamma_s \Delta T_c = 0.025$). An important observation from Fig. 3(b) is that the optimal median TTS scales as an exponential function of the square root of the problem size, that is, an exponential of $\sqrt{n}$ rather than $n$. This unique trend was first noticed in [17].

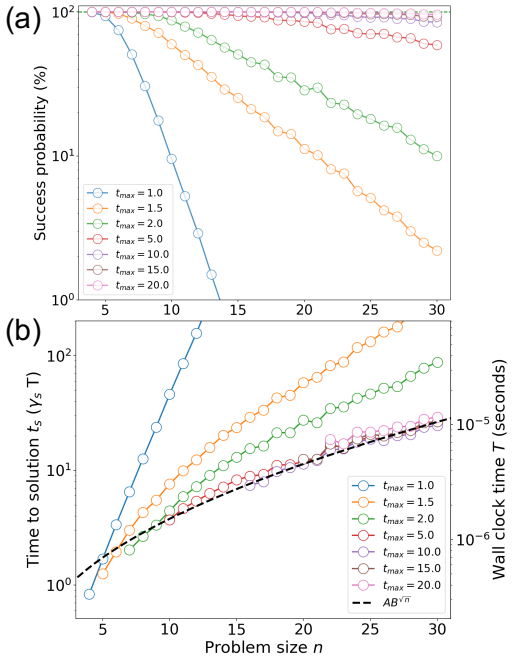



FIG. 3: (a) Success probability $P_s$ and (b) time-to-solution (in units of signal field decay time $1/\gamma_s$) as a function of problem size $n$ for various runtimes $t_{\max}$. The black dotted line shows the best-fit TTS curve of the form $AB^{\sqrt{n}}$.

Fig. 4(a) and (b) show the optimum TTS of the closed-loop CIM and the open-loop CIM with respect to the problem size $n$. We solve two types of MaxCut problems. The first type are randomly generated instances with edge weights $J_{ij} \in \{-1, -0.9, \ldots, 0.9, 1\}$. We refer to these instances as 21-weight MaxCut problem instances. The second type are randomly generated Sherrington–Kirkpatrick (SK) spin glass instances with $J_{ij} = \pm 1$. We study the open-loop CIM with the same Gaussian quantum model without dynamical modulation of $e_i(t)$, $a_i(t)$, and $p_i(t)$, but with measurement-induced state reduction (the third term of Eq. (2) and the second term of Eq. (3)) [43]. We set the feedback parameters $\beta = 0$, $\rho_a = 0$, and $\rho_p = 0$ for the open-loop CIM in order to have a constant feedback field strength $e_i(t) = e_i(0) = 1.0$. The pump rate $p$ is linearly increased from $p = 0.5$ at $t = 0$ (below threshold) to $p = 1.0$ at $t_{\max}$ (above threshold). As shown in Fig. 4(a) and (b), the performance of the closed-loop CIM is superior to that of the open-loop CIM for both types of MaxCut problems.

Table II summarizes the best-fitting parameters $A$ and $B$ for a function of the form $t_s = AB^{\sqrt{n}}$ in both the closed-loop and open-loop CIMs. The smaller coefficient values for $B$ for the closed-loop CIM than those for the open-loop CIM highlight the superior scaling of the closed-loop CIM compared to the open-loop variant. We note that $A$ is expressed in units of a normalized time $t_s = \gamma_s T$, where $T$ is the wall-clock time.

TABLE II: Parameters $A$ and $B$ found by regression of a function of the form $AB^{\sqrt{n}}$ to the TTS curves of the closed-loop and open-loop CIMs for the two types of MaxCut instances.

| | 21-weight random $J_{ij}$ | | Binary random $J_{ij}$ | |
|---|---|---|---|---|
| | $n = 4, \ldots, 30$ | | $n = 4, \ldots, 30$ | |
| | Closed loop | Open loop | Closed loop | Open loop |
| A | 0.26 | 0.32 | 0.16 | 0.13 |
| B | 2.32 | 4.12 | 2.33 | 3.92 |

### C. Discrete-Time Model

The previous section presented the results of our study of the performance of closed-loop and open-loop CIMs with a high-finesse cavity. Nevertheless, it is obvious that a low-finesse cavity with a larger signal decay rate $\gamma_s$ is favourable in terms of the runtime of the algorithm. This is because the wall-clock time $T$ scales as $T = t_s/\gamma_s$. However, it appears that the continuous-time Gaussian quantum theory based on the master equation [Eq. (1)] breaks down in the case of a low-finesse cavity. Here, we describe a new discrete-time Gaussian quantum model [44].

We treat the MFB-CIM as an $n$-mode bosonic system with $2n$ quadrature operators, $\hat{X}_1, \hat{P}_1, \ldots, \hat{X}_n, \hat{P}_n$, satisfying $[\hat{X}_k, \hat{P}_{k'}] = i\delta_{kk'}$. If the system is in a Gaussian state, it is fully characterized by a mean-field vector $\mu$ and a covariance matrix $\Sigma$. In other words, the density operator of each OPO pulse can be written as $\hat{\rho}_i(\mu_i, \Sigma_i)$,

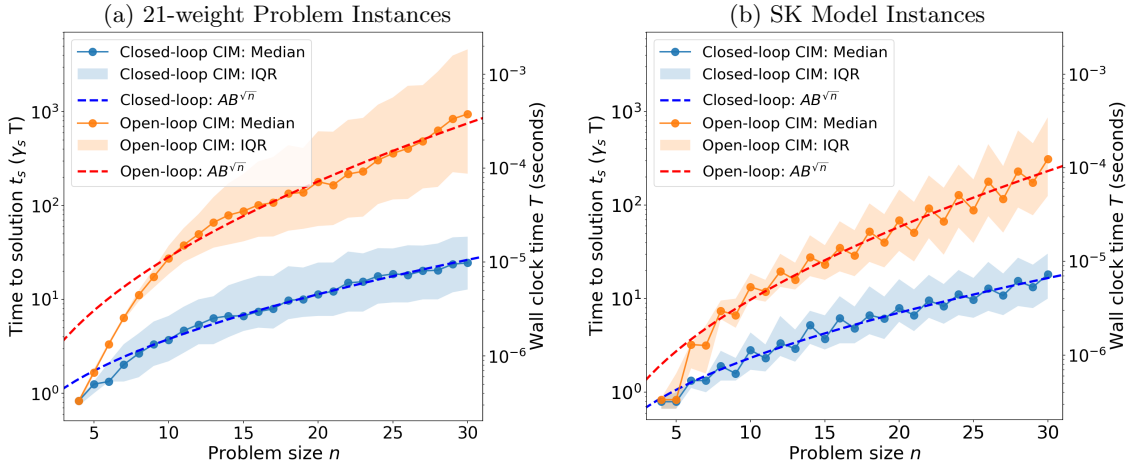



FIG. 4: The optimal (median) time-to-solution of the closed-loop CIM and open-loop CIM on (a) 21-weight randomly generated $J_{ij}$ and (b) binary-weight randomly generated instances ($J_{ij} = \pm 1$, SK model). The shaded regions represent the interquartile range (IQR), showing the region between the 25th and 75th percentiles obtained from the 1000 instances. The dashed blue and red lines are fitted curves of the form $AB^{\sqrt{n}}$.

where

$$\mu_i = (\langle \hat{X}_i \rangle, \langle \hat{P}_i \rangle), \tag{9}$$

$$\Sigma_i = \begin{pmatrix} \langle \hat{X}_i^2 \rangle & \frac{1}{2}\langle \Delta\hat{X}_i \Delta\hat{P}_i + \Delta\hat{P}_i \Delta\hat{X}_i \rangle \\ \frac{1}{2}\langle \Delta\hat{X}_i \Delta\hat{P}_i + \Delta\hat{P}_i \Delta\hat{X}_i \rangle & \langle \hat{P}_i^2 \rangle \end{pmatrix}. \tag{10}$$

We let $\hat{\rho}\left(\mu_i(\ell), \Sigma_i(\ell)\right)$ denote the state of the $i$-th OPO pulse just before it starts its $\ell$-th round trip through the cavity. To propagate the state of the $i$-th signal pulse from $\hat{\rho}(\mu_i(\ell), \Sigma_i(\ell))$ to $\hat{\rho}(\mu_i(\ell+1), \Sigma_i(\ell+1))$, we perform the following five discrete maps iteratively: the background linear-loss map $\mathcal{B}$, the OPO crystal propagation map $\chi$, the out-coupling loss map $\mathcal{B}_{\text{out}}$, the homodyne detection map $H$, and the feedback injection map $\mathcal{D}$. These discrete maps are defined in Appendix A.

In order to see how the wall-clock TTS of the closed-loop and open-loop CIMs are decreased by increasing the cavity loss rate, $\gamma_s$, we solved the 21-weight MaxCut instances and the SK model instances for $n = 30$ to explore the TTS as a function of the normalized loss rate $\gamma_s \Delta T_c$. The results are shown in Fig. 5.

As expected, the TTS (expressed in terms of the number of round trips) decreases monotonically for both problem types and for both the closed-loop CIM and the open-loop CIM as long as $\gamma_s \Delta T_c \lesssim 0.1$ (i.e., in the case of a high-finesse cavity). However, if $\gamma_s \Delta T_c \gtrsim 1$ (i.e., in the case of a very-low-finesse cavity), the TTS increases for both the closed-loop and the open-loop CIMs. This is because one homodyne measurement per round-trip loss does not provide sufficiently accurate information about the internal OPO pulse state and, therefore, the measurement-feedback circuit fails to implement the Ising Hamiltonian and self-diagnosis feedback properly. At $n = 30$, the optimum normalized loss rate is $\gamma_s \Delta T_c \sim 1$ for both the closed-loop and the open-loop CIMs.

## III. SCALING OF DAQC

We now analyze discrete adiabatic quantum computation (DAQC) for solving MaxCut problems. DAQC is associated with the first-order Suzuki—Trotter expansion of the adiabatic Hamiltonian evolution in this paper. We consider an interesting variant of DAQC to be the quantum approximate optimization algorithm (QAOA) [12, 45]. This algorithm attempts to prepare the ground state of a target Hamiltonian $H_P$. QAOA is considered to be an interesting candidate for solving combinatorial optimization problems on NISQ devices, and its performance as a NISQ algorithm is being studied [22, 36, 37, 46]. Similar to DAQC, the circuit ansatz of QAOA is a Trotterized analogue of quantum adiabatic evolution. The state $|+\rangle^{\otimes n}$ is prepared on $n$ qubits, and is evolved through a sequence of $p$ "layers". Each layer consists of an evolution according to a target Hamiltonian $H_P$ along a computational basis, here chosen to be the Pauli-$Z$ eigenbasis, followed by an evolution under a mixing Hamiltonian $H_M = \sum_i X_i$. A vector of tunable parameters $\gamma = (\gamma_1, \ldots, \gamma_p)$ is chosen, where each entry $\gamma_i$ corresponds to the angle of rotation along $H_P$ in the $i$-th layer. Similarly, a vector $\beta = (\beta_1, \ldots, \beta_p)$ is chosen for the $H_M$ evolutions. Finally, the qubits undergo projective measurements in the computational basis, and the measurement results are used to compute the energy of the Hamiltonian $H_P$. The circuit for QAOA is represented in Fig. 6.

A "shot" of the circuit with parameters $(\gamma, \beta)$ is defined as a single execution of the circuit from preparation to measurement, and returns a single energy measurement. Multiple shots performed with the same parameters $(\gamma, \beta)$ can return different results, as they are taken from independent copies of the same prepared state $|\psi(\gamma, \beta)\rangle$. For the weighted MaxCut problem, we use the target Hamiltonian $H_P = \sum_{i,j} J_{ij} Z_i Z_j$, which is

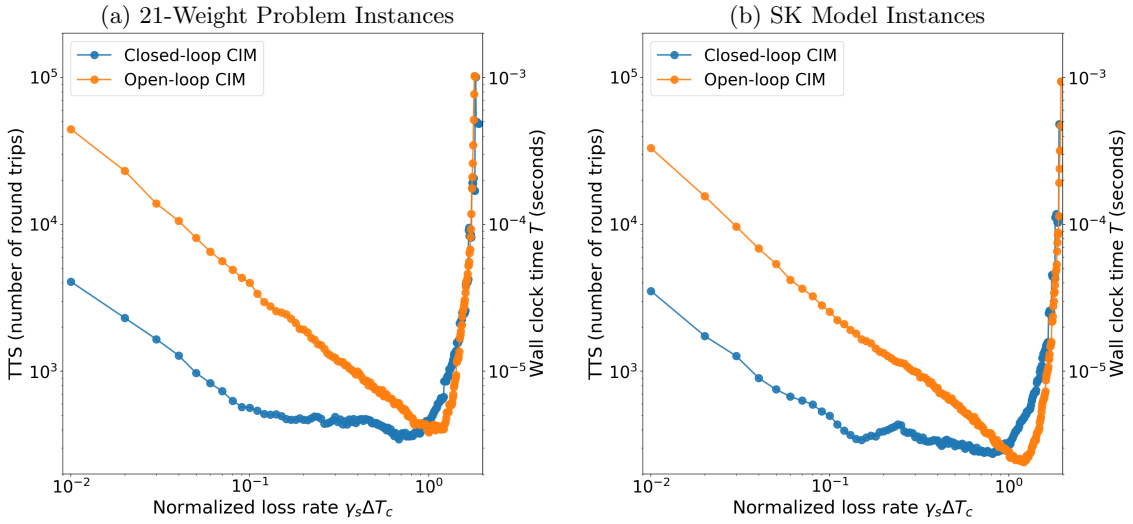



FIG. 5: Median TTS in units of the number of round trips (left y-axis) and corresponding wall-clock time (right y-axis) of the closed-loop CIM and the open-loop CIM versus the normalized loss rate $\gamma_s \Delta T_c$ for (a) 21-weight problem instances and (b) SK model instances, of size $n = 30$ in both cases.

diagonal in the computational basis and whose ground states correspond to the largest cuts of the complete $n$-vertex graph with edge weights $J_{ij}$.

We study two schemes for optimizing the gate parameters of the QAOA algorithm. As mentioned in the introduction, the first scheme treats gate parameters as hyper-parameters that follow a tuned DAQC schedule, in accordance with the ordinary DAQC approach. The second scheme uses a variational hybrid quantum–classical protocol to optimize the gate parameters. We find the first scheme to be superior to the second.

### A. Time-to-Solution Scaling of DAQC

To study the time-to-solution of DAQC in the solving of MaxCut problems, we analyze the QAOA algorithm using pre-tuned Trotterized adiabatic scheduling. We use randomly generated graphs of size $n \in \{10, \ldots, 20\}$. Our test set consists of 1000 graphs of each size, with edge weights $J_{k\ell} = \pm 0.1j$, where $j \in \{0, 1, \ldots, 10\}$.

Given a parameter vector $(\gamma, \beta)$, we evaluate the TTS of QAOA as a product of two terms [6],

$$\mathrm{TTS}(\gamma, \beta) = R_{99}(\gamma, \beta) \cdot t_{\mathrm{ss}}\,, \tag{11}$$

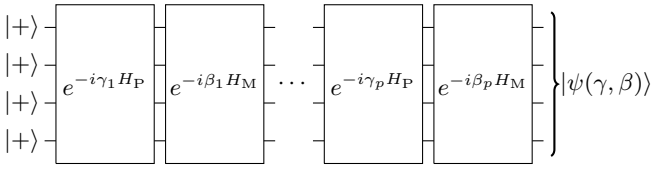



FIG. 6: Quantum approximate optimization algorithm (QAOA) circuit with $p$ layers. The rotation parameters satisfy $\gamma_i \in [0, \pi)$ and $\beta_i \in [0, \pi/2)$. This ansatz is analogous to the Hamiltonian simulation circuits implementing a discretized adiabatic evolution in terms of a first-order Suzuki—Trotter expansion, which we refer to as DAQC.

where $t_{\mathrm{ss}}$ is the time taken for a single shot.

The $R_{99}$ is the number of shots that must be performed to ensure a 99% probability of observing the ground state of $H_P$. It is a metric commonly used to benchmark the success of heuristic optimization algorithms. If the state $|\psi(\gamma, \beta)\rangle$ has a probability $p$ of being projected onto the ground state, then

$$R_{99}(\gamma, \beta) = \frac{\log(0.01)}{\log(1 - p)}. \tag{12}$$

We estimated the time required for a single shot using the following assumptions for an ideal, highly performant quantum computer with access to arbitrary-angle, single-qubit $X$-rotations and two-qubit $ZZ$-rotations.

**Assumption 1.** The preparation and measurements of qubits collectively take 1.0 microseconds. The processor performs any single-qubit or two-qubit gate operations in 10 nanoseconds. Gate operations may be performed simultaneously if they do not act on the same qubit. In addition, all components of the circuit are noise-free and, therefore, there is no overhead for quantum error correction or fault-tolerant quantum computation.

For each problem size varying from 10 to 20 vertices, Fig. 7 shows a plot of the median TTS, suggesting that the TTS scales exponentially with respect to problem size. With more layers, QAOA has a lower potential $R_{99}$, but a single shot takes more time. We found the best scaling was achieved with $p \approx 20$ layers. However, near-term hardware will suffer from various sources of noise, such as decoherence and control noise, which will restrict us to employing shallow QAOA circuits with only a few layers, for example, $p = 4$.

The QAOA parameters $(\gamma, \beta)$ used in Fig. 7 were produced using the formula explained in what follows. Recall the setup for quantum adiabatic evolution [47]. Given an

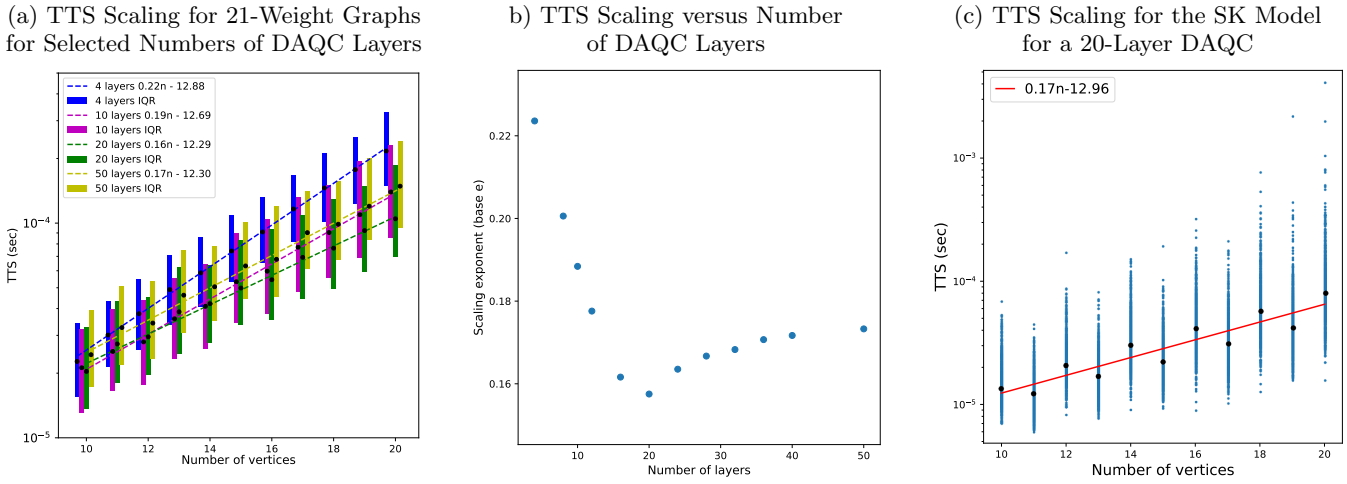



FIG. 7: Scaling of DAQC in solving MaxCut problems. The TTS results are obtained by simulating the QAOA algorithm, using pre-tuned adiabatic scheduling rather than optimizing its parameters variationally. (a) TTS scaling for a 4-, 10-, 20-, and 50-layer DAQC as the problem size grows from 10 to 20 vertices. A best-fit line (dashed) is drawn to the median of the TTSs of the 1000 instances of each size, whose IQR ranges are represented using coloured bars. The equation of this linear regression is given by $\ln(TTS) = mn + b$, where $n$ is the problem size. (b) Slope of the linear regression for a range of layers. The best scaling for DAQC on these 21-weight MaxCut instances is observed at 20 layers. (c) TTS scaling for the SK model, when using a 20-layer DAQC. A best-fit linear-regression is drawn to the median of the TTSs of the 1000 instances for each size.

initial Hamiltonian $H_0$ and a target Hamiltonian $H_1$, we consider the time-dependent Hamiltonian

$$H(t) = s(t)H_1 + (1 - s(t))H_0, \quad t \in [0, T]$$

over a total annealing time $T$, where the function $s(t)$ is an increasing schedule satisfying $s(0) = 0$ and $s(T) = 1$. The time-dependent Hamiltonian $H(t)$ is then applied to the ground state of $H_0$. Let $\psi(t)$ denote the wavefunction at time $t$, so that $\psi(0)$ is the ground state of $H_0$ and $\psi$ evolves according to the Schrödinger equation

$$\dot{\psi} = -i\Big(s(t)H_1 + (1 - s(t))H_0\Big)\psi.$$

We use Trotterization to approximate the prepared state $\psi(T)$. Let

$$c_k := \int_{(k-1)T/p}^{kT/p} s(t)\,dt \quad \text{and} \quad b_k := \int_{(k-1)T/p}^{kT/p} (1 - s(t))\,dt.$$

Then,

$$\psi(T) \approx e^{-ib_pH_0}e^{-ic_pH_1}\cdots e^{-ib_1H_0}e^{-ic_1H_1}\psi(0), \qquad (13)$$

and this approximation becomes exact in the limit as $p \to \infty$.

The Hamiltonians $H_0$ and $H_1$ are both chosen to have a Frobenius norm equal to 1. We divide both $H_M$ and $H_P$ by their corresponding norms, which can easily be calculated, as each Hamiltonian is a sum of the orthogonal Pauli terms

$$H_0 = \frac{1}{\|H_M\|}H_M = -\frac{1}{\sqrt{n}}\sum_i X_i$$

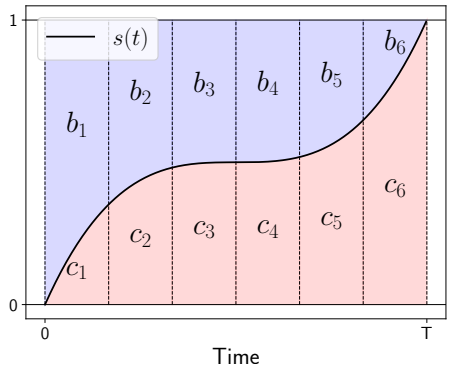



FIG. 8: Trotterization of adiabatic evolution into $p = 6$ layers. The integrals computing $b_k$ and $c_k$ yield the coefficients for $H_0$ and $H_1$, respectively.

and

$$H_1 = \frac{1}{\|H_P\|}H_P = \frac{1}{\sqrt{\sum_{i,j} J_{ij}^2}}\sum_{i,j} J_{ij}Z_iZ_j.$$

Thus,

$$\gamma_k = \int_{(k-1)T/p}^{kT/p} \frac{s(t)}{\|H_P\|}\,dt \quad \text{and} \quad \beta_k = \int_{(k-1)T/p}^{kT/p} \frac{1 - s(t)}{\|H_M\|}\,dt.$$

Empirically, we found that enforcing this Frobenius normalization has yielded a very well-performing schedule for QAOA for multiple problem types. The theoretical basis for this is yet to be fully understood.

The schedule $s(t)$ should have an "inverted S" shape [48, 49] in order to handle the squeezed energy gap in the middle. We take $s(t)$ to be a cubic function

with the general form

$$s(t) = \frac{t}{T} + a \cdot \frac{t}{T}\left(\frac{t}{T} - \frac{1}{2}\right)\left(\frac{t}{T} - 1\right) \qquad (14)$$

for a free parameter $a$. When $a = 0$, $s(t)$ is a straight linear path. When $a = 4$, $s(t)$ is a curved path with a slope of 0 at $t = T/2$. We found by empirical means that $a = 4$ and $T = p(1.6 + 0.1n)$ are the best hyperparameters. See Appendix B for more details.

We also compare the TTS for DAQC to the TTS for Breakout-Local Search (BLS), a classical search algorithm. For each graph instance, 20 runs of BLS were performed, and runtimes were averaged rto obtain the TTS. The algorithm's runtime for each run was capped at 0.1 seconds, although the minimum value was almost always found within that time. Fig. 9 demonstrates that the TTS for DAQC shows no significant correlation with the TTS for BLS.

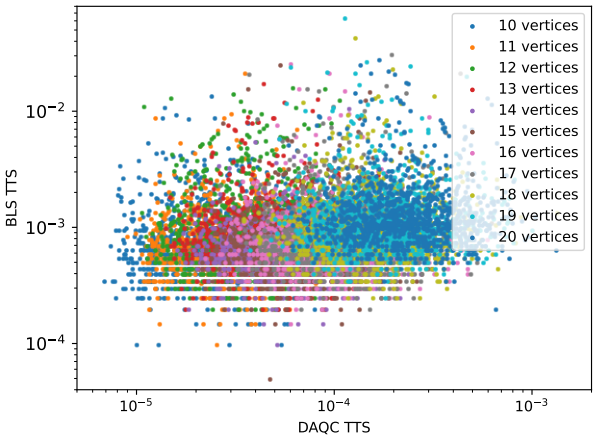



FIG. 9: Scatter plot of DAQC-TTS versus BLS-TTS indicates there is no significant correlation between the difficulty of an instance for DAQC versus the difficulty of an instance for Breakout-Local Search.

### B. Challenges Encountered when Using the Variational Approach

The QAOA protocol usually includes an optimization loop which learns better parameters $(\gamma, \beta)$ by using the data from already-performed shots. However, we found that including an optimization step did not improve the total TTS for the following reasons, and therefore did not include the step in our analysis. The $R_{99}$ is impossible to measure without knowledge of the ground state, and therefore any optimization routine must instead rely on energy measurements. A common approach is to use the "expected energy", $\langle\psi(\gamma,\beta)|\, H_{\mathrm{P}}\, |\psi(\gamma,\beta)\rangle$, which is estimated by averaging over the multiple shots taken with the parameters $(\gamma, \beta)$. This approach suffers from two limitations. First, we must use a large number of shots to accurately estimate the expected energy, which makes the optimization step costly. Second, the expected energy is an imperfect stand-in for $R_{99}$, and therefore optimization typically offers little to no improvement upon the annealing-inspired parameter schedule. See Appendix C for more details.

## IV. SCALING OF DH-QMF

We now consider using Dürr and Høyer's algorithm for quantum minimum finding (DH-QMF) [20] to find the ground state of an Ising Hamiltonian corresponding to a MaxCut problem. Given a real-valued function $E : S \to \mathbb{R}$ on a discrete domain $S$ of size $N = |S|$, DH-QMF finds a minimizer of $E$ (out of the possibly many) using $\mathcal{O}(\sqrt{N})$ queries to $E$. In our case, the domain $S$ is the set of all spin configurations of a classical Ising Hamiltonian on $n$ sites ($N = 2^n$), and the function $E$ maps each spin configuration to its energy. The DH-QMF algorithm is a randomized algorithm, that is, it succeeds in finding the optimal solution only up to a (high) probability. The probability of failure of DH-QMF can be made arbitrarily small without changing the mentioned complexity. A schematic illustration of DH-QMF is shown in Fig. 10, and additional technical details can be found in Appendix D.

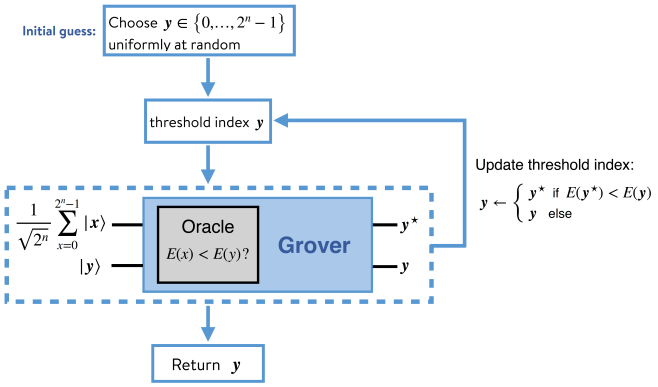



FIG. 10: Schematic illustration of the Dürr–Høyer algorithm for quantum minimum finding /mbox(DH-QMF) applied to searching for a spin configuration corresponding to the energy minimum (ground state). The possible spin configurations are labelled by the indices $y \in \{0, \ldots, 2^n - 1\}$. The algorithm starts by choosing uniformly at random an *initial guess* for the "threshold index" $y$, whose energy $E(y)$ serves as a threshold: solutions to the problem cannot have an energy value larger than this threshold. The main step of the algorithm is a loop consisting of Grover's search for a spin configuration with an energy value strictly smaller than the threshold energy, followed by a threshold-index update. This loop needs to be repeated many times until the threshold index eventually holds the solution with a probability of success higher than a given target lower bound, say, e.g., $p_{\mathrm{succ}} = 0.99$. The final step returns the threshold index as output. A key element of the Grover's search subroutine is an oracle which marks all states whose energies are strictly smaller than the threshold energy. Note that Grover's search may fail to output a marked state.

Given an $n$-spin Ising Hamiltonian

$$H = -\sum_{0\le i<j\le n-1} J_{ij} Z_i Z_j \tag{15}$$

corresponding to an undirected weighted graph of size $n$, its $N = 2^n$ energy eigenstates can be labelled by the integer indices $0 \le y \le N-1$, with the corresponding energy eigenvalues $E(y)$. The index $y$ associated with a computational basis state $|y\rangle = |\eta_0\rangle \otimes \cdots \otimes |\eta_{n-1}\rangle$ represented by the classical bits $\eta_j \in \{0, 1\}$ is the binary representation $y = \sum_{j=0}^{n-1} \eta_j 2^j$ of the bit string $(\eta_0, \ldots, \eta_{n-1})$.

The algorithm starts by choosing uniformly at random an index $y \in \{0, \ldots, N-1\}$ as the initial "threshold index". The threshold index is used to initiate a Grover's search [19, 50]. The Grover subroutine searches for a label $y^\star$ whose energy is strictly smaller than the threshold value $E(y)$. We measure the output of Grover's search and (classically) ascertain whether the search has been successful, $E(y^\star) < E(y)$, in which case we (classically) update the threshold index from $y$ to $y^\star$, and then continue by performing the next Grover's search using the new threshold. The threshold is not updated if Grover's search fails to find a better threshold.

In this paper, we assume a priori knowledge of a hyperparameter we call the number of "Grover iterations" (see Section IV B) inside every Grover's search subroutine that guarantees a sufficiently small failure probability. However, the practical scheme for using DH-QMF consists of multiple trials of Grover's search and iterative updates to the threshold index. We terminate this loop when the Grover subroutine repeatedly fails to provide any further improvement to $y$ and the probability of the existence of undetected improvements drops below a sufficiently small value. Finally, we return the last threshold index as the solution. As shown in [20], the overall required number of Grover iterations needed to find the ground state with sufficiently high probability, say 1/2, is in $\mathcal{O}(\sqrt{N})$.

### A. Time-to-Solution Benchmark for DH-QMF

We investigate the scaling of the time required by DH-QMF to find a solution of weighted MaxCut instances with a 0.99 success probability, assuming an optimistic scenario that is explained in Section IV B. This runtime is analogous to the TTS measure defined in previous sections for the heuristic algorithms of the MFB-CIM and DAQC and we therefore call this runtime a TTS as well. For each instance of the problem we have estimated an *optimistic lower bound* on the runtime of the quantum algorithm with numbers of Grover's iterations in DH-QMF set (ahead of any trials) to achieve an at least 0.99 success probability. As this optimal number of Grover's iterations is dependent on the specific MaxCut instance, we consider this an optimistic bound on performance of DH-QMF. We use the same test set of randomly generated 21-weight MaxCut instances as in previous sections.

Our results are illustrated in Fig. 11. The optimistic values for the TTS are in the range of orders of magnitude of 1.0 milliseconds – 1.0 seconds for the considered range of the number of vertices, $10 \le n \le 20$, using the same set of assumptions for the quantum processor as in Assumption 1.

Our estimates for the runtime of the quantum algorithm are obtained as follows. We note that DH-QMF consists of a sequence of Grover's search algorithms. The total runtime of DH-QMF is therefore the sum of the runtimes of the quantum circuits, each of which corresponds to a Grover's search. The runtime of each such circuit is calculated using the *depth* of that circuit, which is the length of the longest sequence of native operations on the quantum processor (i.e., qubit preparations, single-qubit and two-qubit gates, and qubit measurements) in that circuit, assuming maximum parallelism between independent operations. This path is also known as the "critical path" of a circuit. The runtime of the circuit is therefore identical to the sum of the runtimes of the operations along the critical path, with a contribution of 1.0 microsecond in total for both qubit initialization and measurement, and 10 nanoseconds for any quantum gate operation along the critical path.

The asymptotic scaling of the TTS is identical to the scaling of the circuit depth, which is

$$\Theta\left(\sqrt{2^n}\left(n^2 \log\log n + (\log n)^2 + n\right)\right), \tag{16}$$

as shown in Appendix D. Here the $\Theta\left(\sqrt{2^n}\right)$ contribution is that of the number of Grover iterations (identical to the query complexity of Grover's search), while the $\mathrm{poly}(n, \log n, \log\log n)$ factors are the contribution of each single Grover iteration consisting of an oracle query with implementation cost $\Theta\left(n^2 \log\log n + (\log n)^2\right)$ and the Grover diffusion with cost $\Theta(n)$. A nonlinear least-squares regression toward this scaling is shown in Fig. 11 for both the 21-weight and the SK model problem instances, respectively. Note that the contributions of logarithmic terms are significant only for small problem sizes.

Alongside the optimistic runtime, we have also computed lower bounds on the number of quantum gates, including concrete counts for the overall number of single-qubit gates, two-qubit Cnot gates, and $T$ gates (see Fig. 12). Our circuit analysis in Appendix D yields the gate complexity

$$\Theta\left(\sqrt{2^n}\left(n^2 \log n \log\log n + (\log n)^2 + n\right)\right). \tag{17}$$

Our resource estimates have been generated using ProjectQ [51].

### B. The Optimal Number of Grover Iterations

In what follows, we explain how the algorithm can always be designed such that the output is indeed a ground

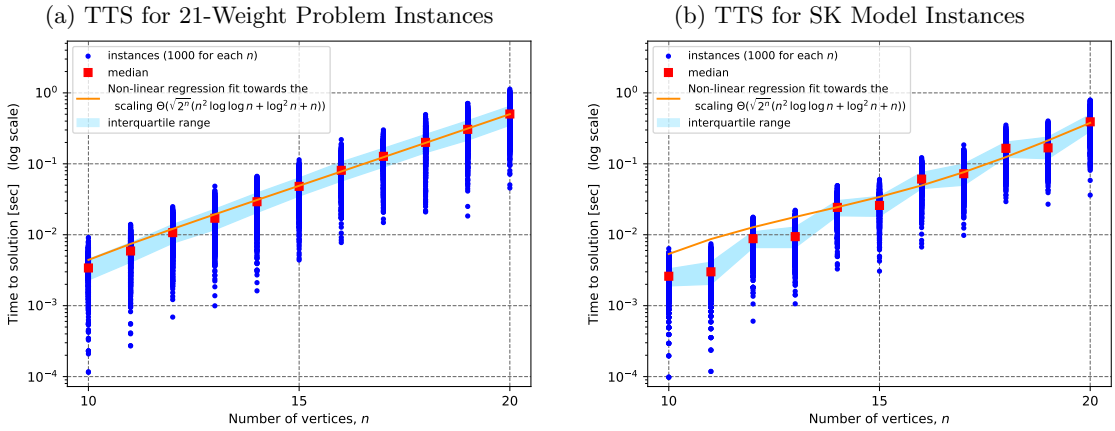



FIG. 11: Scaling of Dürr and Høyer's algorithm for quantum minimum finding (DH-QMF) in solving MaxCut. (a) Time-to-solution (TTS) for 21-weight problem instances. (b) TTS for the SK model instances. In both cases, for each value of the number of vertices in the range $10 \leq n \leq 20$, DH-QMF has been emulated for 1000 (dark blue data) MaxCut instances (see main text). A non-linear least-squares regression (orange curve) has been performed to fit the expected runtime scaling in Eq. (16), respectively, resulting in a sum of squared residuals approximately $1.2 \times 10^{-4}$ seconds$^2$ for 21-weighted instances and $3.30 \times 10^{-3}$ seconds$^2$ for the SK model instances. A logarithmic scale has been used to display the data and the regression fits. Note that the contributions from the logarithmic factors become more (less) significant for smaller (larger) problem sizes.

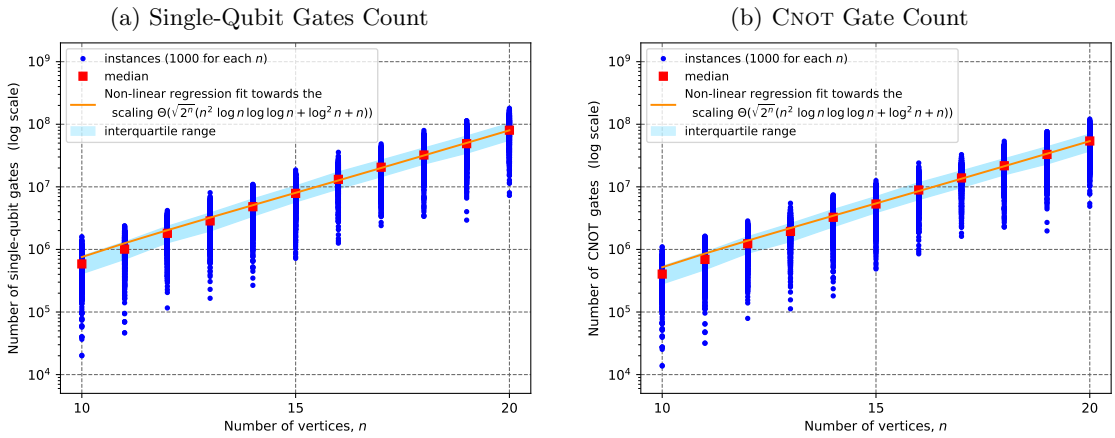



FIG. 12: Optimistic gate counts for DH-QMF in solving the MaxCut problem. For each value of the number of vertices in the range $10 \leq n \leq 20$, the DH-QMF algorithm was emulated for 1000 (blue data) 21-weight MaxCut instances, see main text. Concrete counts were conducted for the (a) overall number of single-qubit gates, and (b) two-qubit CNOT gates. A non-linear least-squares regression (orange curve) has been performed to fit the expected gate complexity given in Eq. (17), respectively. A logarithmic scale has been used to display the data and the regression fits.

state with a probability higher than any target lower bound for the probability of success, for example, 0.99.

A key component of Grover's search as part of QMF is an oracle that marks every input state $|x\rangle$ whose energy is strictly smaller than the energy corresponding to the threshold index $y$ (see Fig. 10). We call it the "QMF oracle" and denote it by $O_{\mathrm{QMF}}$ to distinguish it from the "energy oracle" $O_E$ which computes the energy of a state under the problem Hamiltonian. The oracle $O_{\mathrm{QMF}}$ uses an ancilla qubit initialized in the state $|z\rangle$ to store its outcome

$$O_{\mathrm{QMF}} : |x\rangle\,|z\rangle \longmapsto |x\rangle\,|z \oplus f(x)\rangle\,, \tag{18}$$

where $f(x) = 1$ if, and only if, $E(x) < E(y)$, and $f(x) = 0$ otherwise. Here, $\oplus$ represents a bitwise XOR. The QMF oracle is constructed from multiple uses of the energy oracle $O_E$ and an operation that compares the values held by two registers. Details of this construction can be found in Appendix D 2. The combined effect of querying $O_{\mathrm{QMF}}$ followed by the Grover diffusion (together forming the *Grover iteration* to be repeated $\mathcal{O}\left(\sqrt{2^n}\right)$ times) results in constructively amplifying the amplitudes of the marked items while diminishing the amplitudes of the unmarked ones.

When there are multiple solutions to a search problem, as is frequently the case in the Grover subroutine of QMF, the *optimal number of Grover iterations* needed to maximize the success probability depends on the number of marked items as well. Indeed, suppose we were to have knowledge of the number of marked items $t$ ahead of time. Then, the optimal number of Grover iterations could be obtained from the closed formulae provided in

[50]:

$$\wp_{\text{succ}} = \sin^2\left((2m+1)\theta\right), \\ \wp_{\text{fail}} = \cos^2\left((2m+1)\theta\right). \tag{19}$$

Here, $m$ is the number of Grover iterations, and $\theta$ is defined by $\sin^2\theta = t/N$. Hence, the success probability is maximized for the optimal number of Grover iterations $m_{\text{opt}} = \lfloor \pi/4\theta \rfloor$. We also observe that after exactly $m_{\text{opt}}$ iterations the failure probability obeys

$$\wp_{\text{fail}} \leq \sin^2\theta = t/N,$$

which is negligible when $t \ll N$.

In practice, $t$ and, consequently, $m_{\text{opt}}$ are often unknown. Nevertheless, [50, Sec. 4 and Theorem 3] propose a method to find a marked item with query complexity $\mathcal{O}\left(\sqrt{N/t}\right)$ even when no knowledge of the number of solutions is assumed.

To simplify the analysis for our benchmark in this paper, we examine each MaxCut instance and assume $t$ is known every time Grover's search is invoked. This assumption provides a lower bound on the performance of DH-QMF. In view of the previous discussion, having knowledge of $t$ allows us to compute $m_{\text{opt}}$, $\wp_{\text{succ}}$, and $\wp_{\text{fail}}$.

We then boost the overall success probability of Grover's search to any target success probability $p_{\text{G}}$ by repeating it $K$ times, where $K$ satisfies

$$p_{\text{G}} \leq 1 - \wp_{\text{fail}}^K. \tag{20}$$

Moreover, if DH-QMF requires $J$ non-trivial threshold index updates in total, we must succeed in every boosted Grover search (each including $K$ Grover searches). The probability of this event is thus at least $p_{\text{G}}^J$. Finally, let us denote the target lower bound for the probability of success of the overall DH-QMF algorithm by $p_{\text{succ}}$. We then must have

$$p_{\text{succ}} \leq p_{\text{G}}^J. \tag{21}$$

We achieve a lower bound for $K$ using Eqs. (20) and (21):

$$K \geq \frac{\log\left(1 - p_{\text{succ}}^{\frac{1}{J}}\right)}{\log \wp_{\text{fail}}}. \tag{22}$$

Note that this number still depends on the optimal number $m_{\text{opt}}$ of Grover iterations. The remainder of this section explains how the latter number is sampled for each MaxCut instance via Monte Carlo simulation.

Given a weighted graph, we first generate the histogram of the sizes of all cuts in the graph. Examples of such histograms are provided in Fig. 13. This cut-size histogram allows us to perform a Monte Carlo simulation of the progression of DH-QMF as follows. The DH-QMF algorithm starts by choosing uniformly at random an initial cut $C$ as the threshold index. The resulting energy threshold is therefore sampled according to the cut-size histogram. Grover search then attempts to find a larger

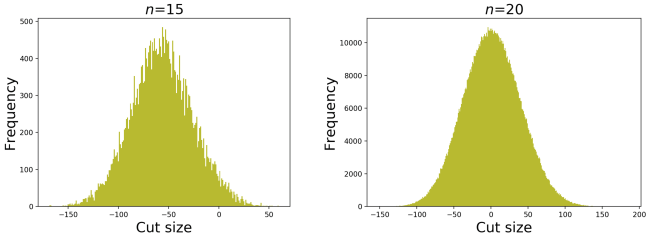



FIG. 13: Typical cut-size histograms of undirected random weighted graphs with weights $w_{k\ell} = \pm 0.1j$, where $j \in \{0, 1, \ldots, 10\}$. Two instances are shown for random graphs with $n = 15$ (left) and $n = 20$ (right) vertices. Note that, for a fully connected graph with $n$ vertices, the overall number of edges is $n(n-1)/2$.

cut. The number of these cuts is $t$ in the notation above, and can be found if the cut-size histogram is known. Using Eq. (19), we can also compute the optimal number $m_{\text{opt}}$ of Grover iterations needed to achieve the highest possible success rate $\wp_{\text{succ}}$ in that search. We furthermore can now use Eq. (22) to predict the number $K$ of Grover searches needed to boost the success probability to at least $p_{\text{G}}$. The cut $C$ is now replaced with a larger cut also selected at random using the cut-size histogram, and this simulation is repeated for the next iteration in DH-QMF.

We repeatedly sample and update the threshold until we find a maximum cut (i.e., at an iteration where $t = 0$). At this point, we stop our Monte Carlo simulation (even though in practice it will not be known that $t$ has become zero). For each sampling step $j$, we count the total number $t_j$ of states contributing to strictly greater cuts and use it to calculate the optimal number $m_{\text{opt}}^{(j)}$ of Grover iterations as well as the number of boosting iterations $K_j$ via Eq. (22).

We now obtain an optimistic TTS as well as an optimistic gate count estimate using the formulae

$$\text{TTS} = \sum_{j=1}^{J} K_j m_{\text{opt}}^{[j]} \times \textsc{Runtime}, \tag{23}$$

$$\#\,\text{gates} = \sum_{j=1}^{J} K_j m_{\text{opt}}^{[j]} \times \textsc{GateCount}. \tag{24}$$

Here, Runtime denotes the running time and GateCount indicates the gate count for a single Grover iteration. In Section IV A we provided optimistic estimates for the number of single-qubit gates, Cnot gates, and $T$ gates. The quantum circuit implementation of a single Grover iteration is presented in Appendix D.

## V. COMPARISON OF THE THREE ALGORITHMS

A direct comparison of the three algorithms for solving MaxCut is illustrated in Fig. 14. In Fig. 14 (a), the median wall-clock TTS of DH-QMF, DAQC, and the

closed-loop MFB-CIM are plotted as a function of problem size $n$ for randomly generated 21-weight MaxCut instances. The solid blue line indicates a best-fitting curve, $f_{\rm CIM}(n) = AB^{\sqrt{n}}$, for the closed-loop MFB-CIM, where $A = 121$ nanoseconds and $B = 2.21$; the solid orange line represents a best-fitting curve, $f_{\rm DAQC}(n) = A'B'^n$, for a 20-layer DAQC, where $A' = 4.6$ microseconds and $B' = 1.17$; and the solid green curve represents a best-fitting curve, $f_{\rm QMF}(n) = \left(\tilde{A}n^2 \log\log n + \tilde{C}(\log n)^2 + \tilde{D}n\right)\tilde{B}^n$, for DH-QMF, where $\tilde{B} = \sqrt{2}$, and $\tilde{A}$, $\tilde{C}$, and $\tilde{D}$ are equal to 17.3, $2.87\times10^3$, and $-1.65\times10^3$ microseconds, respectively.

In order to see how the performance of a closed-loop MFB-CIM scales with increasing problem size, we solved MaxCut problems with SK instances of problem sizes $n = 100, 200, \ldots, 800$. A total of 100 instances of the SK model for each problem size were randomly generated. Using a closed-loop MFB-CIM, we solved each instance 100 times to evaluate the success probability $P_{\rm s}$ of finding a ground state and compute a wall-clock time to achieve a success probability of $\geq 0.99$. It is assumed that all-to-all spin coupling is implemented in 10 nanoseconds, which corresponds to a cavity round-trip time. The signal field lifetime is 100 nanoseconds, that is, $N_{decay} = 10$. We used the discrete map Gaussian model to simulate such a low-finesse machine. The results are shown in Fig. 14(b), along with the predicted performance of DAQC and DH-QMF for the SK model instances.

The minimum wall-clock TTS for the closed-loop MFB-CIM at the optimized runtime $t_{\rm max}$ scales as an exponential function of $\sqrt{n}$, while those for DH-QMF and DAQC scale as exponential functions of $n$. At a problem size of $n = 800$, the wall-clock TTS for the closed-loop MFB-CIM is $\sim$ 10 milliseconds , while those for DH-QMF and DAQC are $\sim 10^{120}$ seconds and $\sim 10^{50}$ seconds, respectively.

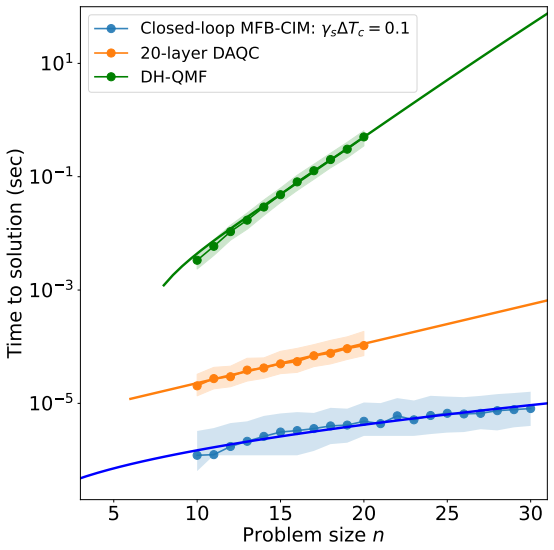
## VI. CONCLUSION

In this paper, we have studied the scaling of two types of measurement-feedback coherent Ising machines (MFB-CIM) and compared this scaling to that of the discrete adiabatic quantum computation (DAQC) and the Dürr–Høyer algorithm for quantum minimum finding (DH-QMF). We performed this comparative study by testing numerical simulations of these algorithms on 21-weight MaxCut problems, that is, weighted MaxCut problems with randomly generated edge weights attaining 21 equidistant values from $-1$ to 1.

The MFB-CIM of the first type is an open-loop MFB-CIM with predefined feedback control parameters and the second is a closed-loop MFB-CIM with self-diagnosis and dynamically modulated feedback control parameters. The open-loop MFB-CIM utilizes the anti-squeezed $\hat{X}$ amplitude near threshold under a positive pump amplitude for finding a ground state but at larger problem sizes the machine is often trapped in local minima. The closed-loop MFB-CIM employs the squeezed $\hat{X}$ amplitude under a negative pump amplitude, in which a finite internal energy is sustained through an external feedback injection signal rather than through parametric amplification. This second machine self-diagnoses its current state by performing Ising energy measurement and comparison with the previously attained minimum energy. The machine continues to explore local minima without getting trapped even in a ground state. We observed that for both the 21-weight MaxCut problems and the SK Ising model, the closed-loop MFB-CIM outperforms the open-loop MFB-CIM. One remarkable result is that a low-finesse cavity machine realizes a shorter TTS than a high-finesse one. This fact clearly demonstrates that the dissipative coupling of the machine to external reservoirs is a crucial computational resource for MFB-CIMs. The wall-clock TTS of the closed-loop MFB-CIM closely follows TTS $\approx 4.32 \times (1.34)^{\sqrt{n}}$ microseconds for the SK model instances of size $n$ ranging from 100 to 800, assuming a cavity round-trip time of 10 nanoseconds and a $1/e$ signal amplitude decay time of 100 nanoseconds ($\gamma_{\rm s}\Delta T_{\rm c} = 0.1$). The performance of the MFB-CIM shown in Fig. 14 is already competitive against various heuristic solvers implemented on advanced digital platforms such as CPUs, GPUs, and FPGAs, in which massive parallel computation is performed over many, many billions of transistors [6, 7, 52–55]. Note that the results shown in Fig. 14 assume one and only one OPO implemented in the MFB-CIM as an active element. If a future MFB-CIM were to implement multiple OPOs, parallel computation would become possible and its performance could be greatly improved.

We have also studied the scaling of QAOA in solving 21-weight and SK model MaxCut problem instances. We considered two schemes for optimizing the quantum gate parameters of QAOA, denoted in the paper as $(\gamma, \beta)$. In the first scheme, we treat $\gamma$ and $\beta$ as hyperparameters that follow a schedule inspired by the adiabatic theorem. In this case, QAOA can be viewed as a Trotterization of an adiabatic evolution from the ground state of a mixing Hamiltonian to the ground state of a problem Hamiltonian. The second scheme views QAOA as a variational (hybrid) quantum algorithm wherein a classical optimizer is tasked with optimization of the parameters $\gamma$ and $\beta$. The variational scheme must perform repeated state preparation and projection measurements to estimate the ensemble averaged energy, which makes the optimization step not only costly but vulnerable to the shot noise of these measurements. Another disadvantage of the variational scheme is that optimizing the ensemble average energy does not necessarily improve the TTS, which is the more practical measure of performance for the algorithm (see Appendix C for more details). As shown in Fig. 16, the adiabatic schedules achieve very low $R_{99}$ values, suggesting a challenging bound for the number of shots allowed by the variational scheme to outperform QAOA for this problem. In view of these considerations,

(a) TTS Scalings for 21-Weight Graphs




(b) TTS Scalings as Functions of $\sqrt{n}$ for the SK Model

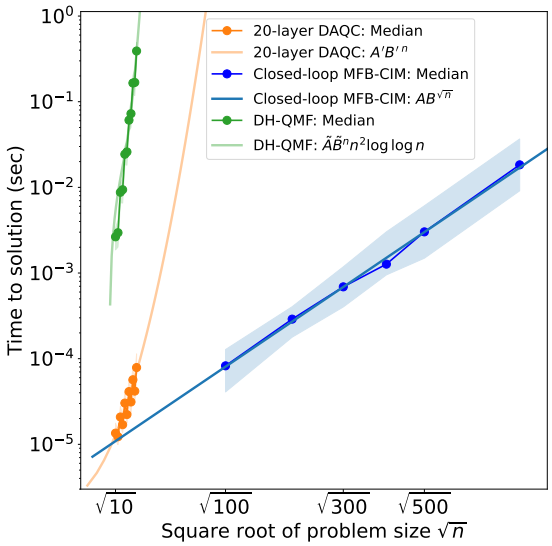



FIG. 14: Comparison of the time-to-solution (TTS) scalings for the MFB-CIM, DAQC, and DH-QMF in solving MaxCut. (a) Wall-clock time of a closed-loop CIM with a low-finesse cavity ($\gamma_s \Delta T_c = 0.1$), DAQC with an optimum number of layers ($p = 20$), and DH-QMF with an a priori known number of optimum iterations versus problem size $n$. (b) TTS of the closed-loop CIM on the fully connected SK model for problem sizes from $n = 100$ to $n = 800$, in steps of 100. For each problem size, the minimum TTS with respect to the optimization over $t_{\max}$ is plotted. In comparison, the SK model TTSs are shown for 20-layer DAQC and DH-QMF for problem sizes ranging from $n = 10$ to $n = 20$. The straight, lighter-blue line (a linear regression) for the CIM demonstrates a scaling according to $AB^{\sqrt{n}}$. The lighter-orange and lighter-green best-fit curves for DAQC and DH-QMF are extrapolated to larger problem instances, illustrating a scaling that is exponential in $n$ rather than in $\sqrt{n}$. In both figures, the shaded regions show the IQRs.

we used the adiabatic scheme to predict the performance of QAOA for large problem sizes, in which case QAOA is considered a pre-tuned DAQC algorithm. In contrast, we note that the quantum state in an MFB-CIM survives through repeated measurements, as the measurements performed on the OPO pulses are not direct projective measurements but indirect approximate measurements. These measurements perturb the internal quantum state of the OPO network but do not completely destroy it. As a result, the above drawback of a variational scheme for QAOA does not apply to the closed-loop MFB-CIM. The wall-clock TTS of QAOA with hypertuned adiabatic schedules (in this case viewed as a DAQC algorithm) is well-represented by TTS $\approx 4.6 \times (1.17)^n$ microseconds. As shown in Fig. 14, extrapolating this trend suggests that DAQC will perform poorly compared to MFB-CIM as the problem sizes increase due to an exponential dependence on the number, $n$, of vertices in the MaxCut problem compared to an exponential growth with a $\sqrt{n}$ exponent in the case of MFB-CIM.

Finally, we have also studied the scaling of DH-QMF for solving 21-weight and SK model MaxCut problems. As this algorithm is based on Grover's search, it performs $\tilde{\mathcal{O}}(\sqrt{2^n})$ Grover iterations, implying it makes a number of queries, of the same order, to its oracle. The algorithm also iterates on multiple values of a classical threshold index; however, this does not change the dominating factors in the scaling of the algorithm. We have shown that the wall-clock TTS of DH-QMF is well-approximated by TTS $\approx 17.3 \times 2^{n/2} n^2 \log\log n$ microseconds when extrapolated to larger problem sizes. As shown in Fig. 14, DH-QMF requires a computation time that is many orders of magnitude larger than that for either DAQC or MFB-CIM. This comparatively poor performance of DH-QMF can be traced back to the linear amplitude amplification in the Grover iteration in contrast to the exponential amplitude amplification at the threshold of the OPO network. Our study thus leaves open the question of whether there exist optimization tasks for which Grover-type speedups are of practical significance.

## ACKNOWLEDGMENTS

The authors thank Marko Bucyk for carefully reviewing and editing this manuscript. AS thanks Shengru Ren for helpful discussions. All authors acknowledge the support of the NSF CIM Expedition award (CCF-1918549). PR furthermore thanks the financial support of Mike and Ophelia Lazaridis, and Innovation, Science and Economic Development Canada.

# APPENDICES

## Appendix A: The Discrete Map Gaussian Model of CIM

In this Appendix, we summarize the discrete-map Gaussian model of the CIM presented in [44], and we adapt the feedback step to include the dynamic feedback control used for the closed-loop MFB-CIM. This discrete-map model is used to study the optimization performance of the MFB-CIM in Section II C. In the discrete Gaussian quantum model of MFB-CIM, the density operator of the $i$-th OPO pulse is fully characterized by the mean amplitude $\mu_i$ and covariance matrix $\Sigma_i$ defined by Eqs. (9) and (10). The total density operator before all pulses start their $\ell$-th round trip is expressed by $\otimes_{i=1}^{n}\hat{\rho}(\mu_i(\ell),\Sigma_i(\ell))$. Propagation of the state of the $i$-th pulse through $\ell$-th roundtrip from $\hat{\rho}(\mu_i(\ell),\Sigma_i(\ell))$ to $\hat{\rho}(\mu_i(\ell+1),\Sigma_i(\ell+1))$ is described by performing the following discrete maps consecutively.

1. *Background linear loss:* The lumped background linear loss transforms the density operator as

$$\hat{\rho}(\mu_i,\Sigma_i)\mapsto \mathrm{tr}_c\left(\mathcal{B}\left[\hat{\rho}(\mu_i,\Sigma_i)\otimes\hat{\rho}(0_c,\Sigma_c^0)\right]\right), \tag{A1}$$

where $\Sigma_c^0=\mathrm{diag}(1/2,1/2)$ is the covariance of a coherent state. The beamsplitter map $\mathcal{B}$ is defined by

$$\mathcal{B}\left[\hat{\rho}(\mu,\Sigma)\right]=\hat{\rho}(S\mu,S\Sigma S^{\mathrm{T}}), \tag{A2}$$

$$S=\begin{pmatrix} t & 0 & -r & 0\\ 0 & t & 0 & -r\\ r & 0 & t & 0\\ 0 & r & 0 & t\end{pmatrix}. \tag{A3}$$

Here, $t=\sqrt{1-r^2}$ is the amplitude transmission coefficient of a fictitious beamsplitter which represents background linear loss. Physically, $\hat{\rho}(0_c,\Sigma_c^0)$ is a reservoir vacuum state and it is traced out after mixing with the signal pulse at the beamsplitter.

2. *Parametric amplification/deamplification during OPO crystal propagation:* The propagation through a second-order nonlinear crystal with the pump pulse transforms the density operator as

$$\hat{\rho}(\mu_i,\Sigma_i)\mapsto \mathrm{tr}_b\left[\chi\Big(\hat{\rho}(\mu_i,\Sigma_i)\otimes\hat{\rho}(\mu_b,\Sigma_b^0)\Big)\right], \tag{A4}$$

where $\mu_b$ and $\Sigma_b^0=\mathrm{diag}(1/2,1/2)$ describe the initial condition of the (Gaussian) pump pulse, and the map $\chi$ abstractly represents their joint propagation through the crystal, that is,

$$\chi:\hat{\rho}(\mu_i,\Sigma_i)\otimes\hat{\rho}(\mu_b,\Sigma_b^0)\mapsto\hat{\rho}(\mu_{i,b},\Sigma_{i,b}), \tag{A5}$$

where $\hat{\rho}(\mu_{i,b},\Sigma_{i,b})$ is a joint two-mode Gaussian state of the signal and pump at the output. This joint output is determined by the equations of motion for the mean-field and covariance matrix:

$$\frac{d\langle\hat{X}_i\rangle}{dt}=\epsilon\langle\hat{X}_b\rangle\langle\hat{X}_i\rangle+\epsilon\langle\delta\hat{X}_b\delta\hat{X}_i+\delta\hat{P}_b\delta\hat{P}_i\rangle \tag{A6}$$

$$\frac{d\langle\hat{X}_b\rangle}{dt}=-\frac{\epsilon}{2}\langle\hat{X}_i{}^2\rangle-\frac{\epsilon}{2}\langle\delta\hat{X}_i{}^2-\delta\hat{P}_i{}^2\rangle \tag{A7}$$

$$\frac{d\langle\delta\hat{X}_i{}^2\rangle}{dt}=2\epsilon\langle\hat{X}_b\rangle\langle\delta\hat{X}_i{}^2\rangle+2\epsilon\langle\hat{X}_i\rangle\langle\delta\hat{X}_b\delta\hat{X}_i\rangle \tag{A8}$$

$$\frac{d\langle\delta\hat{P}_i{}^2\rangle}{dt}=-2\epsilon\langle\hat{X}_b\rangle\langle\delta\hat{P}_i{}^2\rangle+2\epsilon\langle\hat{X}_i\rangle\langle\delta\hat{P}_b\delta\hat{P}_i\rangle \tag{A9}$$

$$\frac{d\langle\delta\hat{X}_b{}^2\rangle}{dt}=-2\epsilon\langle\hat{X}_i\rangle\langle\delta\hat{X}_b\delta\hat{X}_i\rangle \tag{A10}$$

$$\frac{d\langle\delta\hat{P}_b{}^2\rangle}{dt}=-2\epsilon\langle\hat{X}_i\rangle\langle\delta\hat{P}_b\delta\hat{P}_i\rangle \tag{A11}$$

$$\frac{d\langle\delta\hat{X}_b\delta\hat{X}_i\rangle}{dt}=\epsilon\langle\hat{X}_i\rangle\langle\delta\hat{X}_b{}^2-\delta\hat{X}_i{}^2\rangle+\epsilon\langle\hat{X}_b\rangle\langle\delta\hat{X}_b\delta\hat{X}_i\rangle \tag{A12}$$

$$\frac{d\langle\delta\hat{P}_b\delta\hat{P}_i\rangle}{dt}=\epsilon\langle\hat{X}_i\rangle\langle\delta\hat{P}_b{}^2-\delta\hat{P}_i{}^2\rangle-\epsilon\langle\hat{X}_b\rangle\langle\delta\hat{P}_b\delta\hat{P}_i\rangle \tag{A13}$$

Here $\epsilon$ is the parametric coupling rate defined by the Hamiltonian $\mathcal{H}=i\frac{\hbar\epsilon}{2}(\hat{b}\hat{a}^{\dagger^2}-\hat{b}^{\dagger}\hat{a}^2)$, where $\hat{a}$ and $\hat{b}$ are signal and pump annihilation operators. We assume that the input state into the crystal satisfies $\langle\hat{P}_i\rangle=\langle\hat{P}_b\rangle=0$ (i.e., there is no coherent excitation along the quadrature-phase) and $\langle\{\delta\hat{X}_i,\delta\hat{P}_i\}\rangle=\langle\{\delta\hat{X}_b,\delta\hat{P}_b\}\rangle=0$ (both the signal and pump have no correlation between in-phase and quadrature-phase fluctuations). Note that $\langle\hat{P}_i\rangle=\langle\hat{P}_b\rangle=0$ is satisfied at all times under the above conditions. The defined map $\chi$ thus describes all such effects as linear parametric amplification/deamplification, signal-pump entanglement formation, and back conversion from the pump to the signal.

3. *Outcoupling and homodyne detection:* The outcoupling of the internal signal pulse is described by the map

$$\begin{aligned}\hat{\rho}\left(\mu_i,\Sigma_i\right)&\mapsto\hat{\rho}\left(\mu_{i,h},\Sigma_{i,h}\right)\\&=\mathcal{B}_{\mathrm{out}}\left[\hat{\rho}\left(\mu_i,\Sigma_i\right)\otimes\hat{\rho}\left(0_h,\Sigma_h^0\right)\right],\end{aligned} \tag{A14}$$

where the beamsplitter map $\mathcal{B}_{\mathrm{out}}$ is defined by Eqs. (A2) and (A3) with an outcoupling rate of $r_{\mathrm{out}}$. In Eq. (A14), a probe mode $h$ is prepared in a vacuum state and mixed with the signal pulse. This process creates a joint correlated state (entangled state) between the internal pulse and external (outcoupling) pulse. Suppose a homodyne measurement for the outcoupled pulse reports a result $m_i(l)$ for the $i$-th signal pulse at the $\ell$-th round trip. Such an indirect approximate measurement projects the internal state to a new state by the map

$$\begin{aligned}\hat{\rho}\left(\mu_{i,h},\Sigma_{i,h}\right)&\mapsto\mathcal{H}\left[\hat{\rho}\left(\mu_{i,h},\Sigma_{i,h}\right)\right]\\&=\hat{\rho}\left(\mu_i^{(m_i)},\Sigma_i^{(m_i)}\right),\end{aligned} \tag{A15}$$

where the homodyne detection map $\mathcal{H}$ is defined by

$$\mu_i^{(m_i)} = \mu_i + \left(\frac{w_i - \mu_h}{\Sigma_{XX}}\right) v_X \tag{A16}$$

$$\Sigma_i^{(m_i)} = \Sigma_i - \frac{v_X {v_X}^{\mathrm{T}}}{\Sigma_{XX}}. \tag{A17}$$

Here, $v_X$ is the $X$ off-diagonal component (the degree of signal-probe correlation) of the matrix $\Sigma_{i,h}$ and $\Sigma_{XX}$ is the $X$ diagonal element of $\Sigma_h$. The second terms of the right-hand sides of Eqs. (A16) and (A17) express the mean-field shift and variance reduction induced by the homodyne measurement.

4. *Feedback injection:* We finally implement the Ising coupling by applying the displacement operation for the internal pulse amplitude based on the measurement results of the $\ell$-th round trip, $m_j(\ell)$, for all pulses except for the $i$-th pulse. The displacement magnitude is given by

$$v_i(\ell) = e_i(\ell) \sum_{j \neq i} J_{ij} m_j(l), \tag{A18}$$

where $e_i(\ell)$ is the feedback-field amplitude of the $\ell$-th round trip which is determined by the equation of motion, Eq. (5), for the closed-loop CIM. The feedback injection map is thus determined by

$$\hat{\rho}\left(\mu_i, \Sigma_i\right) \mapsto \mathcal{D}_{v_i}\left[\hat{\rho}\left(\mu_i, \Sigma_i\right)\right] = \hat{\rho}\left(\mu_i + v_i, \Sigma_i\right). \tag{A19}$$

The above four steps are applied to all pulses ($i = 1, \ldots, n$), completing one round trip through the CIM cavity.

## Appendix B: Hyperparameter Tuning for QAOA Parameter Schedules

As described in Section III A, we have a recipe for generating DAQC parameter schedules for any problem Hamiltonian $H_{\mathrm{P}}$ and number of layers $p$. We consider two hyperparameters for these schedules:

- The number $L = T/p$ is the evolution time in each Trotterized layer of the associated annealing schedule. A larger value of $L$ corresponds to a slower and therefore better associated annealing schedule, but also brings along a greater Trotterization error;
- The number $a$ is the coefficient of the cubic term in the adiabatic schedule. When $a = 0$ the schedule is linear, and when $a = 4$ the schedule is cubic, with $f'(T/2) = 0$. We therefore only consider $a \in [0, 4]$, because for $a > 4$ the schedule would be decreasing at $t = T/2$.

Here, we compile our results on the performance of QAOA with cubic schedules for various values of the hyperparameters $a$, $L$, and $p$. In Figs. 15 to 17, the horizontal axis displays the number of vertices for the problem instance, and the vertical axis displays the $R_{99}$ or TTS (in logarithmic scale). Each blue dot represents a single problem instance. All plots depict a total of 11,000 problem instances varying from 10 to 20 nodes in size. Each black point represents the geometric mean of all values of $R_{99}$ or TTS for problem instances of a given size. Finally, the red line indicates the best linear fit to the black points. The equation corresponding to the best-fit line is written in each subplot, where $n$ is the number of vertices.

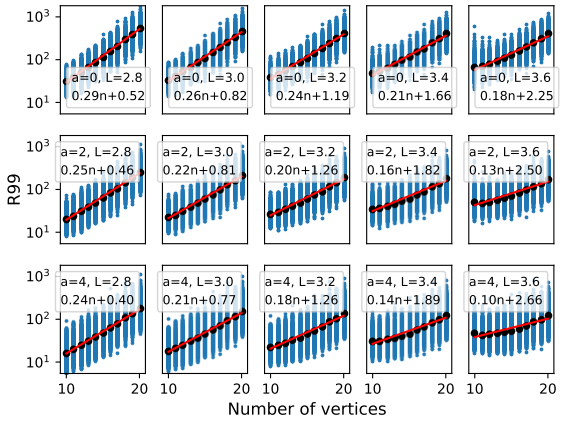



FIG. 15: $R_{99}$ of the good initial QAOA parameters at $p = 4$ layers for various values of $a$ and $L$, on all 1000 graph instances of each size ranging from 10 to 20.

We empirically found that a value of $L$ between 2.6 and 3.6 worked best. In Fig. 15, we plot the $R_{99}$ values of the good parameter schedule with hyperparameters $a \in \{0, 2, 4\}$ and $L \in \{2.8, 3.0, 3.2, 3.4, 3.6\}$. Note that $a = 4$ (a cubic schedule with a derivative of 0 at the inflection point) outperforms $a = 0$ (a linear schedule). We observed that, as the number of vertices $n$ increases, the optimal value of the scaling constant $L$ increases. Therefore, our tuned hyperparameter value used in Figs. 16 and 17 is $L = 1.6 + 0.1n$.

In Figs. 16 and 17, we present the scaling of a linear schedule opposite to that of a cubic schedule. As the number of layers increases, performance as measured by $R_{99}$ improves, as expected. However, with more layers, more time is required to perform a single circuit shot, and therefore the scaling of TTS is actually *worse* at 50 layers than it is at 20 layers. For large numbers of layers, the linear schedule and cubic schedule perform similarly, which is expected because both are Trotterizations of a very slow adiabatic schedule.

## Appendix C: Challenges of Variational QAOA

When QAOA parameter schedules are tuned variationally, the energy measurements from the quantum device are used to decide the next parameters to try via a hybrid quantum–classical process. A single "shot" with parameters $(\gamma, \beta)$ consists of running the QAOA circuit

once with parameters $(\gamma, \beta)$, and measuring the energy of the prepared state $|\psi(\gamma,\beta)\rangle$, which destroys the prepared state and returns a single measurement outcome. We perform a large number of shots using $(\gamma, \beta)$, and the results are averaged to estimate the expected energy

$$EE(\gamma,\beta) := \langle\psi(\gamma,\beta)|H_{\mathrm{P}}|\psi(\gamma,\beta)\rangle. \tag{C1}$$

This expected energy is treated as a loss function which is minimized by a classical optimizer. This approach suffers from two major challenges.

Firstly, we want parameters $(\gamma, \beta)$ which minimize the $R_{99}$, rather than the expected energy. Although these two loss functions are related, they are not perfectly correlated, and this difference becomes more apparent as we move closer to the parameters which minimize $R_{99}$. Unfortunately, it is impossible to optimize the ansatz with respect to $R_{99}$, as this would require knowledge of the ground state.

Secondly, because projective measurements are stochastic, our estimate of the expected energy is approximate, and this makes parameter optimization difficult. To overcome this issue, we would need to use a large number of shots per point $(\gamma, \beta)$, which makes the variational algorithm costly.

In Fig. 18, we illustrate the implications of the first challenge. We consider a four-layer QAOA circuit on graphs of size 10, 15, and 20. For each graph $G$ the following analysis is performed. (i) The cubic schedule $\theta_G$ (see Section III A) is found and its $R_{99}$ is calculated. (ii) The Nelder-Mead method is used to optimize the expected energy, with its parameter schedule initialized as $\theta_G$ and given access to 100 perfect evaluations of expected energy (which ordinarily can only be approximated). The $R_{99}$ of the result is divided by the $R_{99}$ of the cubic schedule, and these ratios have been plotted in red. Finally, (iii) the Nelder–Mead method is used to optimize $R_{99}$, with a schedule initialized with $\theta_G$ and access to 100 perfect evaluations of $R_{99}$ (which is ordinarily impossible to calculate). The $R_{99}$ of the result is divided by the $R_{99}$ of the cubic schedule, and these ratios have been plotted in blue. For better visibility, the graph instances along the $x$-axis have been sorted by the $y$-values of the red points. We observe that even with *perfect* estimation of the expected energy, optimization results in a *worse* final $R_{99}$ in 15 to 40 percent of graph instances. This is the case despite the fact that the cost (in shots) of performing this optimization has been discarded. The effect of including the cost would have been substantial.

## Appendix D: Grover's Search as a Subroutine of DH-QMF

Grover's search algorithm [18, 19] has been extensively studied and applied since its invention more than twenty

FIG. 16: $R_{99}$ and TTS of a linear schedule for $10 \le n \le 20$, $p \in \{4, 10, 20, 50\}$, with hyperparameters $a = 0.0$ and $L = 1.6 + 0.1n$.

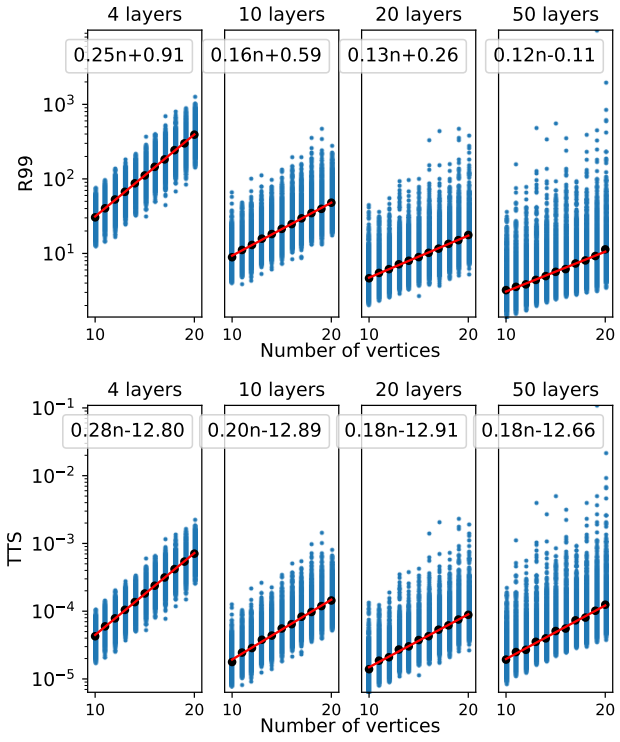
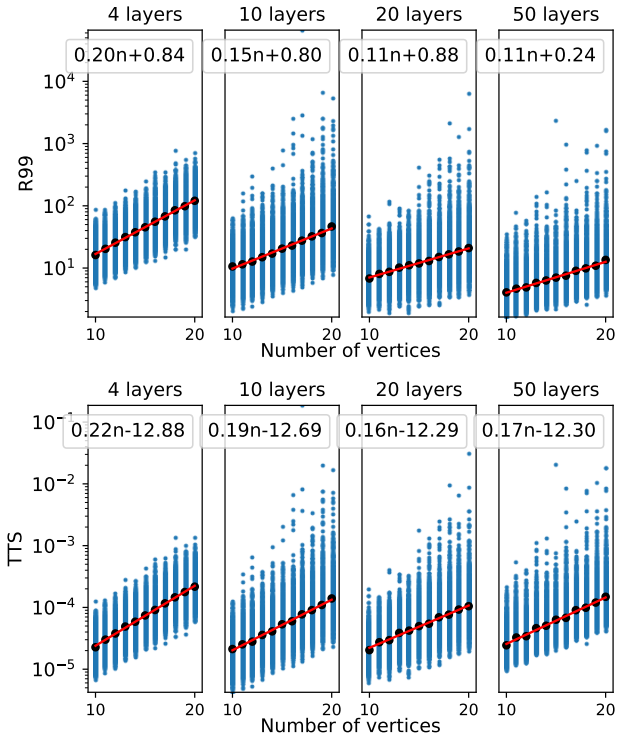



FIG. 17: $R_{99}$ and TTS of a cubic schedule for $10 \le n \le 20$, $p \in \{4, 10, 20, 50\}$, with hyperparameters $a = 4.0$ and $L = 1.6 + 0.1n$. The performance is better than that of the linear schedule for shallow circuits, but stops improving as the number of layers becomes larger.



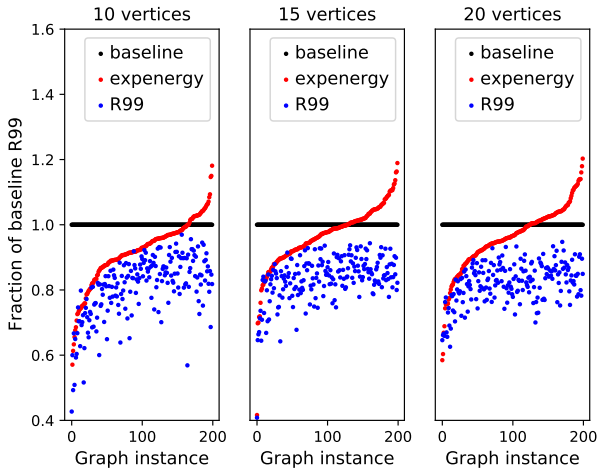



FIG. 18: Plot depicting the fraction of the baseline $R_{99}$ achieved when optimizing for expected energy with no shot noise (red) versus optimizing for $R_{99}$ (blue). Baseline $R_{99}$ (black) is given by the cubic parameter schedule, as described in Section III A. Even when shot noise is absent, optimizing for expected energy can *increase* the $R_{99}$ about a third of the time, as is evidenced by the fact that a third of the red points are above the black line. We performed this optimization using 100 function evaluations using the Nelder–Mead method, and due to imperfect optimization, a few blue points landed above the red curve. The $x$-axis is the graph instance number from 0 to 199, where graphs have been sorted according to the $y$-value of the red point.

years ago. This appendix provides more details with a focus on its implementation as a subroutine of DH-QMF. In Appendix D 1, we start with a brief review of how Grover's search algorithm works. In Appendix D 2, we expand on the quantum circuits used to implement the QMF oracle, which is required when Grover's search is employed as a subroutine of DH-QMF, and explain the contributions to its resource requirements.

### 1. A Brief Review of Grover's Search Algorithm

The circuit of Grover's search algorithm is illustrated in Fig. 19. The quantum circuit takes as inputs an $n$-qubit register `vertex` and a single-qubit register `flag`, where $n = \lceil \log_2 N \rceil$. The `vertex` register is used to encode the possible spin configurations (and any superpositions of them); it is initialized in the state $|0\rangle^{\otimes n}$ and transformed into a uniform superposition $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle_{\texttt{vertex}} \in \left(\mathbb{C}^2\right)^{\otimes n}$ by applying a Hadamard gate (denoted by $H$) to each qubit. The `flag` qubit is prepared in the state $|-\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = HX|0\rangle$. Grover's search is implemented by repeatedly applying the "Grover iterations" a number of times specified by $m$. After $m$ Grover iterations, the register `vertex` is measured in the computational basis. The measurement result ($n$ classical bits) is intended to yield a solution to the problem.

The effect of the Grover iteration is the combined effect of an oracle query followed by the Grover diffusion.

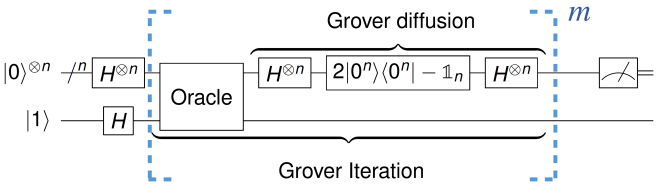



FIG. 19: Quantum circuit for Grover's search. The key components are an oracle, which marks the solution states, and the Grover diffusion, which implements a reflection about the mean amplitude. The composition of the oracle followed by Grover diffusion forms the so-called Grover iteration, which is repeated $m \in \mathcal{O}\left(\sqrt{2^n}\right)$ times. Here, $H$ denotes the Hadamard gate.

To explain the key role of the quantum oracle, it us useful to formulate the search problem as follows. Let $\{x_1, \ldots, x_N\}$ denote the set of the $N$ unordered items. We define a classical function $f : \{x_1, \ldots, x_N\} \to \{0, 1\}$ such that $f(x) = 1$ if and only if $x$ has the property we are looking for, and $f(x) = 0$ otherwise. The problem thus consists in finding an item $x \in \{x_1, \ldots, x_N\}$ such that $f(x) = 1$. The quantum oracle $O_f$ corresponding to the classical function $f$ is a unitary implementation of $f$. It is commonly defined as

$$O_f : |x\rangle_{\texttt{vertex}} |z\rangle_{\texttt{flag}} \mapsto |x\rangle_{\texttt{vertex}} |z \oplus f(x)\rangle_{\texttt{flag}}, \tag{D1}$$

where $z \in \{0, 1\}$ and $\oplus$ represents a bitwise XOR. If we choose $z = 0$, the `flag` qubit outputs the value 1 if and only if $x$ is a solution to the search problem. We say the oracle *marks* the solution states. The crucial property is that the oracle can be queried on a superposition of $N$ input states, and to compute the corresponding function values it needs to be queried only once:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle_{\texttt{vertex}} |0\rangle_{\texttt{flag}} \xmapsto{O_f} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle_{\texttt{vertex}} |f(x)\rangle_{\texttt{flag}}. \tag{D2}$$

In Grover's algorithm, we prepare the `flag` qubit in the $|-\rangle$ state. The resulting effect is a "phase kick-back", which gives rise to a minus sign as a phase whenever the input is a solution state:

$$|x\rangle_{\texttt{vertex}} |-\rangle_{\texttt{flag}} \xmapsto{O_f} (-1)^{f(x)} |x\rangle_{\texttt{vertex}} |-\rangle_{\texttt{flag}}. \tag{D3}$$

Observe that the state of the `flag` qubit remains unaffected and we effectively implement the transformation $|x\rangle_{\texttt{vertex}} \mapsto (-1)^{f(x)} |x\rangle_{\texttt{vertex}}$, which is the definition of a "phase oracle". However, the `flag` qubit plays a crucial role in inducing this transformation. While the factor $(-1)^{f(x)}$ seems like a global phase for a single term, it becomes a relative phase for a superposition of inputs:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle_{\texttt{vertex}} |-\rangle_{\texttt{flag}} \xmapsto{O_f} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle_{\texttt{vertex}} |-\rangle_{\texttt{flag}}. \tag{D4}$$

The following Grover diffusion implements a reflection about the mean amplitude. If $\alpha_x$ denotes the amplitude

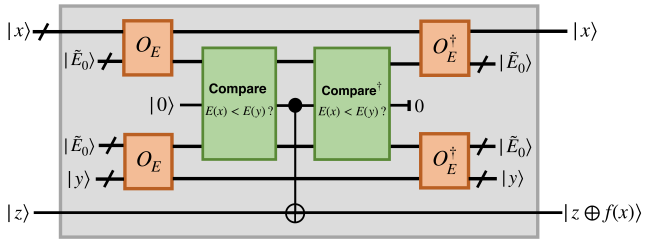
of the $|x\rangle$ component prior to applying the Grover diffusion, the effect of the latter is $\alpha_x \mapsto 2\bar{\alpha} - \alpha_x$, where $\bar{\alpha} := \frac{1}{N}\sum \alpha_x$. Observe that the amplitudes of the marked components (those that pick up a negative phase after the oracle query) are amplified while the amplitudes of all other components decrease. The combined effect of an oracle query followed by the Grover diffusion thus results in *amplitude amplification* of the solution states, while shrinking the amplitudes of all other states in the superposition. When repeated numerous times, the amplitudes of the solution states eventually become significantly larger than those of the non-solution states. The quadratic speedup with respect to classical search can be understood as coming about from adding amplitudes $\Omega\left(\frac{1}{\sqrt{N}}\right)$ to the marked items with each query, which results in an $\mathcal{O}\left(\sqrt{N}\right)$ convergence. This convergence rate was shown by Grover to be also optimal. Hence, the query complexity is actually $\Theta\left(\sqrt{N}\right)$.

### 2. The QMF Oracle

The search for a ground state of an Ising Hamiltonian $H = -\sum_{i<\ell} J_{i\ell} Z_i Z_\ell$ (corresponding to an undirected weighted graph with weights $w_{i\ell} = -J_{i\ell}$) requires an oracle which marks all states whose energies are strictly smaller than the energy corresponding to the latest updated threshold index value, respectively, which we refer to as the "QMF oracle" in this paper. Its quantum-circuit implementation is shown in Fig. 20. Note that here, instead of using the weights $w_{k\ell} = \pm 0.1j \in [-1, 1]$ for $j \in \{0, 1, \ldots, 10\}$, we take the weights to be the integers $-10 \leq w_{k\ell} \leq 10$; this facilitates the quantum circuit implementation of arithmetic operations without altering the underlying MaxCut problem.

In addition to the $n$-qubit register `vertex` for encoding the possible spin configurations and any superpositions of them and a single-qubit register `flag` for holding the result of the oracle, several other computational registers as well as ancillae are required to reversibly compute the energies $E(x)$ and $E(y)$ and compare their values. More concretely, we need another $n$-qubit register to encode the value $y$ of the threshold index as a quantum state $|y\rangle$. Furthermore, we need two computational registers to store the computed values $E(x)$ and $E(y)$; we call these registers "`data`$(H)$" to indicate that they hold the computed data related to the Hamiltonian. Both are initialized such that they initially hold an integer $\tilde{E}_0$ that is an upper bound on the maximum possible absolute value of an energy eigenvalue, $\tilde{E}_0 \geq \max_x |E(x)|$. This energy shift by a constant value allows us to have a nonnegative energy spectrum, which facilitates the implementation of the energy comparison. The maximum possible absolute energy eigenvalue, $\max_x |E(x)|$, is bounded by the product of the total number of edges in the graph times the maximum absolute edge weight in the weighted graph.

(a) Coarse profile of the QMF oracle:




(b) Energy oracle $O_E$:

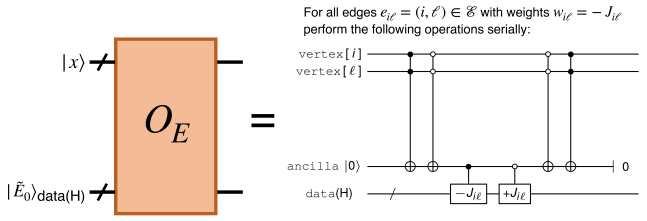



FIG. 20: Quantum oracle as a key component of the Grover step as part of DH-QMF. The oracle marks every state whose energy is strictly smaller than the threshold value $E(y)$, which is computed given the latest threshold index $y$. The result is recorded in a single-qubit `flag`: given its input state $|z\rangle$ (where $z \in \{0, 1\}$), the oracle outputs $|z \oplus f(x)\rangle$, where $f(x) = 1$ if, and only if, $E(x) < E(y)$, and $f(x) = 0$ otherwise. (a) The circuit consists of several queries to the energy oracle $O_E$, which reversibly computes the energy corresponding to a given input state, and applications of a unitary module called "Compare", which compares the values held by two registers and records the result (0 or 1) in a single-qubit ancilla. To infer if $E(x) < E(y)$ for a given input $|x\rangle$, we prepare the quantum state $|y\rangle$ corresponding to the known threshold index $y$, then independently compute $E(x)$ and $E(y)$ by separately employing $O_E$, respectively, and compare their values using "Compare". The computational registers for holding the energy values are initialized in $|\tilde{E}_0\rangle$, where $\tilde{E}_0$ is a constant energy shift chosen so as to avoid negative energies. If $E(x) < E(y)$ is TRUE, a 1 is recorded in an ancilla qubit that was initialized in $|0\rangle$; the ancilla remains unaltered otherwise. Using a Cnot gate, we copy out the result of the comparison to the single-qubit `flag` and reverse the whole circuit producing this result. (b) $O_E$ is implemented by serially executing the shown circuit template for every vertex pair $(i, \ell)$. Depending on whether `vertex`$[i]$ and `vertex`$[\ell]$ carry the same or different values, we respectively subtract or add the value $J_{i\ell}$ in the `data`$(H)$ register.

The registers `data`$(H)$ must thus be able to store a value twice as large as this bound. Since generic weighted graphs have full connectivity, the total number of edges in such graphs is $\binom{n}{2} = n(n-1)/2$, where $n$ is the number of vertices, while the maximum absolute edge weight in our analysis is $\max_{(i,\ell)} |w_{i\ell}| = 10$. Hence, we may use $\tilde{E}_0 := 10\binom{n}{2} = 5n(n-1)$ and choose the registers `data`$(H)$ to be of size $\lceil \log_2 (10n(n-1)) \rceil \in \mathcal{O}(\log n)$.

The energy values $E(x)$ and $E(y)$ are computed using two separate energy oracles, whose quantum circuit implementation is provided in Fig. 20(b). For a given input $|x\rangle = |\xi_0\rangle \otimes \cdots \otimes |\xi_{n-1}\rangle$ held by the `vertex` register, we serially execute the shown circuit template for every vertex pair $(i, \ell)$ in the graph whose edge $e_{i\ell}$ is nonzero. Each such circuit subtracts or adds the value $J_{i\ell}$ in the `data`$(H)$ register, depending on whether $\xi_i = \xi_\ell$ or $\xi_i \neq \xi_\ell$, respectively, effec-

tively contributing the term $(-1)^{\xi_i}(-1)^{\xi_\ell}(-J_{i\ell})$ to the overall energy. The series for all pairs of vertices accumulates the sum $\sum_{ij}(-1)^{\xi_i}(-1)^{\xi_\ell}(-J_{i\ell})$, which together with the initial value $\tilde{E}_0$ results in the value $E(x) = \tilde{E}_0 - \sum_{i\ell}(-1)^{\xi_i}(-1)^{\xi_\ell}J_{i\ell}$ held by the $\texttt{data}(H)$ register as output of the energy oracle $O_E$. Similarly, we obtain the value $E(y) = \tilde{E}_0 - \sum_{i\ell}(-1)^{\eta_i}(-1)^{\eta_\ell}J_{i\ell}$ for the quantum state $|y\rangle = |\eta_0\rangle \otimes \cdots \otimes |\eta_{n-1}\rangle$ corresponding to the threshold index $y$. For generic weighted graphs with full connectivity, this serial implementation contributes a factor $\mathcal{O}(n^2)$ to the overall circuit depth scaling. Moreover, there is an additional contribution from the arithmetic operations needed to implement addition and subtraction of the constant integer $J_{i\ell}$ within the $\texttt{data}(H)$ register. Our circuit implementations and resource estimates have been obtained using projectQ [56]. The implementation of addition or subtraction of a constant $c$, that is, $|E\rangle \mapsto |E \pm c\rangle$, in projectQ [51] is based on Draper's addition in Fourier space [57], which allows for optimization when executing several additions in sequence, which applies to our circuits. Due to cancellations of the quantum Fourier transform (QFT) and its inverse, $\mathrm{QFT}\,\mathrm{QFT}^{-1} = \mathbb{1}$, for consecutive additions or subtractions within the sequence given by the serial execution of circuits shown in Fig. 20(b), the overall sequence contributes a multiplicative factor scaling only as $\mathcal{O}(\log\log n)$ to depth, and a multiplicative factor in $\mathcal{O}(\log n\,\log\log n)$ to the gate complexity. To understand these contributions, recall that the registers $\texttt{data}(H)$ are of size $\mathcal{O}(\log n)$. The remaining initial QFT and the final inverse QFT, which transform into and out of the Fourier space in that scheme (cp. [51]), contribute an additional additive term $\mathcal{O}\left((\log n)^2\right)$ to both the depth and the gate complexity of the overall sequence. Hence, the implementation of the energy oracle $O_E$ contributes the factors $\mathcal{O}(n^2\log\log n + (\log n)^2)$ to the overall circuit depth and $\mathcal{O}(n^2\log n\log\log n + (\log n)^2)$ to the overall gate complexity.

The energy computation is followed by a unitary operation called "Compare", which compares the energies $E(x)$ and $E(y)$. Using methods developed in [58], we can implement this comparison by a circuit with a depth only logarithmic in the number of qubits, that is, with a depth in $\mathcal{O}(\log\log n)$, while its gate complexity is $\mathcal{O}(\log n)$. An additional single-qubit ancilla is used to store the result of the comparison. Concretely, initialized in state $|0\rangle$, the ancilla is output in the state $|f(x,y)\rangle$, where

$$f(x,y) = \begin{cases} 0, & \text{if } E(x) \geq E(y) \\ 1, & \text{if } E(x) < E(y)\ . \end{cases} \tag{D5}$$

Using a CNOT gate, we copy out this result to the single-qubit $\texttt{flag}$ (bottom wire) and reverse the whole circuit used to compute the result so as to uncompute the entanglement with the garbage generated along the way.

In summary, the QMF oracle is a quantum circuit of depth $\mathcal{O}\left(n^2\log\log n + (\log n)^2\right)$ and gate complexity $\mathcal{O}\left(n^2\log n\log\log n + (\log n)^2\right)$. The Grover diffusion requires an $n$-controlled NOT gate to implement the reflection, which is a circuit of depth and gate complexity both scaling as $\mathcal{O}(n)$ in terms of elementary gates. Putting all contributions together, a single Grover iteration in our implementation has a circuit of depth in $\mathcal{O}\left(n^2\log\log n + (\log n)^2 + n\right)$, while its gate complexity is $\mathcal{O}(n^2\log n\log\log n + (\log n)^2 + n)$. While we have not explicitly shown it, we note that the growth rates of circuit depth and gate counts are lower-bounded by the same scalings, meaning that in the above expressions we may replace the $\mathcal{O}(\cdot)$ notation by $\Theta(\cdot)$.

As an additional final remark, we note that it is possible to achieve a slightly better circuit depth scaling for the Grover iteration, namely as $\mathcal{O}\left(n + (\log n)^3 + \log\log n\right)$, by a parallel (instead of serial) execution of the circuit components shown in Fig. 20(b) pertaining to each vertex pair $(i,\ell)$ in the graph. However, this parallelization would come at an unreasonably high additional space cost, as it would necessitate the use of $n(n-1)$ computational registers of size $\mathcal{O}\left(\log n\right)$ instead of only two. The number of qubits required would scale as $\mathcal{O}\left(n + n^2\log n\right)$. In contrast, our serial implementation above requires only $\mathcal{O}\left(n + \log n\right)$ qubits.


[1] F. Barahona, On the computational complexity of Ising spin glass models, Journal of Physics A: Mathematical and General **15**, 3241 (1982).

[2] A. Lucas, Ising formulations of many NP problems, Frontiers in Physics **2**, 5 (2014).

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

[4] S. Kirkpatric, Optimization by simulated annealing, Science **220**, 671 (1983).

[5] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, Optimization by simulated annealing: An experimental evaluation; part I, graph partitioning, Operations Research **37**, 865 (1989).

[6] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, and H. G. Katzgraber, Physics-inspired optimization for quadratic unconstrained problems using a digital annealer, Frontiers in Physics **7**, 48 (2019).

[7] T. Takemoto, M. Hayashi, C. Yoshimura, and M. Yamaoka, 2.6 A 2× 30k-spin Multichip Scalable Annealing Processor based on a Processing-In-Memory Approach for Solving Large-Scale Combinatorial Optimization Problems, in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)* (IEEE, 2019) pp. 52–54.

[8] U. Benlic, E. K. Burke, and J. R. Woodward, Breakout local search for the multi-objective gate allocation problem, Computers & Operations Research **78**, 80 (2017).

[9] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem, Science **292**, 472 (2001).

[10] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse Ising model, Phys. Rev. E **58**, 5355 (1998).

[11] J. Brooke, D. Bitko, G. Aeppli, *et al.*, Quantum annealing of a disordered magnet, Science **284**, 779 (1999).

[12] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv preprint arXiv:1411.4028 (2014).

[13] Z. Wang, A. Marandi, K. Wen, R. L. Byer, and Y. Yamamoto, Coherent Ising machine based on degenerate optical parametric oscillators, Physical Review A **88**, 063853 (2013).

[14] Y. Yamamoto, K. Aihara, T. Leleu, K.-i. Kawarabayashi, S. Kako, M. Fejer, K. Inoue, and H. Takesue, Coherent Ising machines—Optical neural networks operating at the quantum limit, npj Quantum Information **3**, 1 (2017).

[15] V. Choi, Minor-embedding in adiabatic quantum computation: I. The parameter setting problem, Quantum Information Processing **7**, 193 (2008).

[16] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, Evidence for quantum annealing with more than one hundred qubits, Nature Physics **10**, 218 (2014).

[17] R. Hamerly, T. Inagaki, P. L. McMahon, D. Venturelli, A. Marandi, T. Onodera, E. Ng, C. Langrock, K. Inaba, T. Honjo, *et al.*, Experimental investigation of performance differences between coherent Ising machines and a quantum annealer, Science Advances **5**, eaau0823 (2019).

[18] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (1996) pp. 212–219.

[19] L. K. Grover, Quantum mechanics helps in searching for a needle in a haystack, Physical Review Letters **79**, 325 (1997).

[20] C. Dürr and P. Høyer, A quantum algorithm for finding the minimum, arXiv: quant-ph/9607014 (1996).

[21] M. Born and V. Fock, Beweis des Adiabatensatzes, Zeitschrift für Physik **51**, 165 (1928).

[22] G. G. Guerreschi and A. Y. Matsuura, QAOA for max-cut requires hundreds of qubits for quantum speed-up, Scientific Reports **9**, 6903 (2019).

[23] T. Leleu, Y. Yamamoto, P. L. McMahon, and K. Aihara, Destabilization of local minima in analog spin systems by correction of amplitude heterogeneity, Physical review letters **122**, 040607 (2019).

[24] S. Kako, T. Leleu, Y. Inui, F. Khoyratee, S. Reifenstein, and Y. Yamamoto, Coherent Ising machines with error correction feedback, Advanced Quantum Technologies , 2000045 (2020).

[25] C. Xue, Z.-Y. Chen, Y.-C. Wu, and G.-P. Guo, Effects of quantum noise on quantum approximate optimization algorithm, arXiv:1909.02196 (2019).

[26] J. Marshall, F. Wudarski, S. Hadfield, and T. Hogg, Characterizing local noise in QAOA circuits, IOP SciNotes **1**, 025208 (2020).

[27] B. Pablo-Norman and M. Ruiz-Altaba, Noise in Grover's quantum search algorithm, Phys. Rev. A **61**, 012301 (1999).

[28] G. L. Long, Y. S. Li, W. L. Zhang, and C. C. Tu, Dominant gate imperfection in Grover's quantum search algorithm, Phys. Rev. A **61**, 042305 (2000).

[29] H. Azuma, Decoherence in Grover's quantum algorithm: Perturbative approach, Phys. Rev. A **65**, 042311 (2002).

[30] N. Shenvi, K. R. Brown, and K. B. Whaley, Effects of a random noisy oracle on search algorithm complexity, Phys. Rev. A **68**, 052313 (2003).

[31] D. Shapira, S. Mozes, and O. Biham, Effect of unitary noise on Grover's quantum search algorithm, Phys. Rev. A **67**, 042301 (2003).

[32] P. J. Salas, Noise effect on Grover algorithm, The European Physical Journal D **46**, 365 (2008.).

[33] P. Gawron, J. Klamka, and R. Winiarczyk, Noise effects in the quantum search algorithm from the viewpoint of computational complexity, International Journal of Applied Mathematics and Computer Science **22**, 493 (01 Jun. 2012).

[34] D. Reitzner and M. Hillery, Grover search under localized dephasing, Phys. Rev. A **99**, 012339 (2019).

[35] 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, arXiv preprint arXiv:2012.09265 (2020).

[36] G. E. Crooks, Performance of the quantum approximate optimization algorithm on the maximum cut problem, arXiv preprint arXiv:1811.08419 (2018).

[37] 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).

[38] S. Utsunomiya, K. Takata, and Y. Yamamoto, Mapping of Ising models onto injection-locked laser systems, Optics Express **19**, 18091 (2011).

[39] A. Marandi, Z. Wang, K. Takata, R. L. Byer, and Y. Yamamoto, Network of time-multiplexed optical parametric oscillators as a coherent Ising machine, Nature Photonics **8**, 937 (2014).

[40] T. Inagaki, K. Inaba, R. Hamerly, K. Inoue, Y. Yamamoto, and H. Takesue, Large-scale Ising spin network based on degenerate optical parametric oscillators, Nature Photonics **10**, 415 (2016).

[41] P. L. McMahon, A. Marandi, Y. Haribara, R. Hamerly, C. Langrock, S. Tamate, T. Inagaki, H. Takesue, S. Utsunomiya, K. Aihara, *et al.*, A fully programmable 100-spin coherent Ising machine with all-to-all connections, Science **354**, 614 (2016).

[42] T. Shoji, K. Aihara, and Y. Yamamoto, Quantum model for coherent Ising machines: Stochastic differential equations with replicator dynamics, Phys. Rev. A **96**, 053833 (2017).

[43] Y. Inui and Y. Yamamoto, Noise correlation and success probability in coherent Ising machines, arXiv:2009.10328 (2020).

[44] E. Ng, T. Onodera, S. Kako, P. L. McMahon, H. Mabuchi, and Y. Yamamoto, Efficient sampling of ground and low-energy Ising spin configurations with a coherent Ising machine (2021), arXiv:2103.05629 [quant-ph].

[45] 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).

[46] M. Willsch, D. Willsch, F. Jin, H. De Raedt, and K. Michielsen, Benchmarking the quantum approximate

optimization algorithm, Quantum Information Processing **19**, 197 (2020).
[47] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, arXiv preprint quant-ph/0001106 (2000).
[48] G. B. Mbeng, R. Fazio, and G. E. Santoro, Optimal quantum control with digitized quantum annealing, arXiv preprint arXiv:1911.12259 (2019).
[49] J. Roland and N. J. Cerf, Quantum search by local adiabatic evolution, Physical Review A **65**, 042308 (2002).
[50] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, Tight bounds on quantum searching, Fortschritte der Physik: Progress of Physics **46**, 493 (1998).
[51] D. S. Steiger, T. Häner, and M. Troyer, Projectq: An open source software framework for quantum computing, Quantum **2**, 49 (2018).
[52] C. Yoshimura, M. Hayashi, T. Okuyama, and M. Yamaoka, Implementation and Evaluation of FPGA-based Annealing Processor for Ising Model by use of Resource Sharing, International Journal of Networking and Computing **7**, 154 (2017).
[53] T. Leleu, F. Khoyratee, T. Levi, R. Hamerly, T. Kohno, and K. Aihara, Scaling advantage of nonrelaxational dynamics for high-performance combinatorial optimization (2021), arXiv:2009.04084 [physics.comp-ph].
[54] H. Goto, K. Endo, M. Suzuki, Y. Sakai, T. Kanao, Y. Hamakawa, R. Hidaka, M. Yamasaki, and K. Tatsumura, High-performance combinatorial optimization based on classical mechanics, Science Advances **7**, 10.1126/sciadv.abe7953 (2021).
[55] K. Tatsumura, M. Yamasaki, and H. Goto, Scaling out Ising machines using a multi-chip architecture for simulated bifurcation, Nature Electronics **4**, 208 (2021).
[56] *ProjectQ*, `www.projectq.ch`.
[57] T. G. Draper, Addition on a quantum computer, arXiv: quant-ph/0008033 (2000).
[58] D. W. Berry, M. Kieferová, A. Scherer, Y. R. Sanders, G. H. Low, N. Wiebe, C. Gidney, and R. Babbush, Improved techniques for preparing eigenstates of fermionic Hamiltonians, npj Quantum Information **4**, 1 (2018).