Dynamic adaptive quantum approximate optimization algorithm for shallow, noise-resilient circuits

Nikola Yanakiev,[1,2] Normann Mertig,[1] Christopher K. Long,[1,2] and David R. M. Arvidsson-Shukur[1]
[1]*Hitachi Cambridge Laboratory, J. J. Thomson Ave., Cambridge, CB3 0HE, United Kingdom*
[2]*Cavendish Laboratory, Department of Physics, University of Cambridge, Cambridge, CB3 0HE, United Kingdom*

(Received 26 September 2023; accepted 9 February 2024; published 20 March 2024)

The quantum approximate optimization algorithm (QAOA) is an appealing proposal to solve NP problems on noisy intermediate-scale quantum (NISQ) hardware. Making NISQ implementations of the QAOA resilient to noise requires short ansatz circuits with as few controlled-NOT (CNOT) gates as possible. Here we present the dynamic adaptive quantum approximate optimization algorithm (Dynamic-ADAPT-QAOA). Our algorithm significantly reduces the circuit depth and the CNOT count of standard ADAPT-QAOA, a leading proposal for near-term implementations of the QAOA. Throughout our algorithm, the decision to apply CNOT-intensive operations is made dynamically, based on algorithmic benefits. Using density-matrix simulations, we benchmark the noise resilience of ADAPT-QAOA and Dynamic-ADAPT-QAOA. We compute the gate-error probability $p^{\star}_{\text{gate}}$ below which these algorithms provide, on average, more accurate solutions than the classical, polynomial-time approximation algorithm by Goemans and Williamson. For small systems with six to ten qubits, we show that $p^{\star}_{\text{gate}} > 10^{-3}$ for Dynamic-ADAPT-QAOA. Compared to standard ADAPT-QAOA, this constitutes an order-of-magnitude improvement in noise resilience. This improvement should make Dynamic-ADAPT-QAOA viable for implementations on superconducting NISQ hardware, even in the absence of error mitigation.

DOI: 10.1103/PhysRevA.109.032420

I. INTRODUCTION

NP problems are ubiquitous in computer science, occurring frequently in combinatorial optimization, and machine learning [1,2]. Finding their solutions is computationally hard. One strategy to solve NP problems, relies on the Ising model [3–5]. An NP problem is encoded in the real and symmetric matrix W_{ij}. The (approximate) solution is then found by approximating the ground-state energy E_0 of an Ising Hamiltonian

$$\widehat{H} = \frac{1}{4}\sum_{i,j=1}^{N} W_{ij}\widehat{Z}_i\widehat{Z}_j, \tag{1}$$

where $\widehat{Z}_i$ denotes the Pauli-z operator acting on qubit $i = 1,\ldots,N$. Approximate solutions are usually found using heuristics [6–9] or adiabatic quantum computers [10–13]. The quality of these solutions can be compared to the average-case-solution accuracy of the Goemans and Williamson (GW) algorithm [14], which, in the worst case, provides approximate solutions within $87.8\ldots\%$ of the true ground-state energy in polynomial time (using an alternative representation of the NP problem).

Recent works [15,16] have proposed solving NP problems on gate-based quantum computers, using the quantum approximate optimization algorithm (QAOA). The QAOA identifies approximate solutions to NP problems by creating upper bounds to the ground-state energy E_0 of H via the Rayleigh-Ritz variational principle:

$$E_0 \leqslant E(\vec{\beta},\vec{\gamma}) = \langle\Psi(\vec{\beta},\vec{\gamma})|\widehat{H}|\Psi(\vec{\beta},\vec{\gamma})\rangle. \tag{2}$$

The classically hard-to-represent trial state is prepared on a quantum computer by evolving an initial state $|\Psi_0\rangle$:

$$|\Psi(\vec{\beta},\vec{\gamma})\rangle = \widehat{U}_P(\vec{\beta},\vec{\gamma})|\Psi_0\rangle, \tag{3}$$

using a parametrized ansatz circuit

$$\widehat{U}_P(\vec{\beta},\vec{\gamma}) = \prod_{p=1}^{P}\left[e^{-i\beta_p\widehat{A}_p}e^{-i\gamma_p\widehat{H}}\right]. \tag{4}$$

The QAOA then optimizes the parameters to minimize the energy expectation value $E(\vec{\beta},\vec{\gamma})$.

In the original proposal of the QAOA [15], the form of the ansatz circuit [Eq. (4)] is inspired by a Trotterized form of the adiabatic theorem [17]. By setting the mixer Hamiltonian to $\widehat{A}_p = \prod_{i=1}^{N}\widehat{X}_i$ for all p, and the initial state to $|\Psi_0\rangle = |+\rangle\cdots|+\rangle$, the QAOA finds the ground state exactly as the number of Trotter steps tends to infinity ($P\to\infty$). Unfortunately, large values of P lead to intractably deep ansatz circuits. In the presence of noise, the need for deep circuits precludes the implementation of the QAOA on existing quantum hardware [18,19].

To reduce the intractably deep quantum circuits, adaptive QAOA (ADAPT-QAOA) [20] was developed. The algorithm improves the ansatz circuit in P iterations. Further, it allows the mixer Hamiltonian $\widehat{A}_p$ to vary in each iteration p, by choosing it from a mixer pool $\mathcal{P}$. In noiseless numerical simulations, ADAPT-QAOA has been demonstrated to

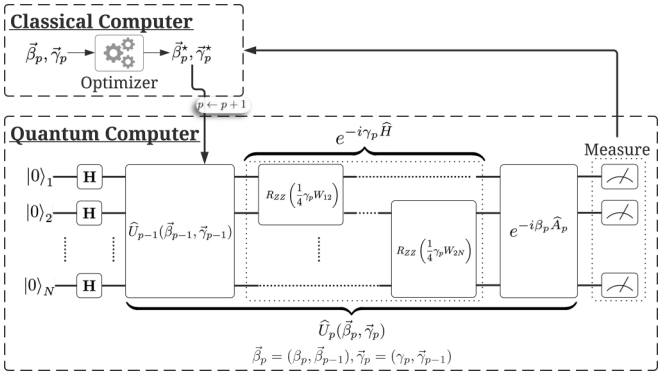

FIG. 1. The pth iteration of ADAPT-QAOA: After initialization, the ansatz circuit from the previous iteration $\widehat{U}_{p-1}$ is augmented by appending unitary evolutions generated by $\widehat{H}$ and $\widehat{A}_p$. The optimal circuit parameters $\vec{\beta}_p^\star, \vec{\gamma}_p^\star$ are identified by minimizing the measured energy expectation.

generate shallower circuits than the QAOA. Despite these improvements, ADAPT-QAOA lies outside the scope of current hardware. Moreover, the resilience of ADAPT-QAOA to noise has never been quantified.

In this paper, we benchmark ADAPT-QAOA in the presence of noise. Using density-matrix simulations, we compute the gate-error probability $p^\star_{\text{gate}}$ below which the quantum algorithm outputs, on average, better approximate solutions than the classical GW algorithm. For small systems of six to ten qubits, we find that ADAPT-QAOA requires $p^\star_{\text{gate}}$ comparable to or smaller than the gate-error probabilities available on current hardware. To reduce the hardware requirements of ADAPT-QAOA further, we develop Dynamic-ADAPT-QAOA. This algorithm removes redundant components from the ansatz circuits. For the problems we study, Dynamic-ADAPT-QAOA reduces the circuit depths significantly. For instance, in noiseless simulations of six-qubit systems, Dynamic-ADAPT-QAOA surpasses the average-case-solution accuracy of the GW algorithm with approximately 80% fewer controlled-NOT (CNOT) gates than the original ADAPT-QAOA. This reduction in CNOT gates leads to improved noise resilience, with $p^\star_{\text{gate}}$ being approximately an order of magnitude better than that of the original ADAPT-QAOA. Dynamic-ADAPT-QAOA may thus be implementable on current superconducting hardware, even in the absence of error mitigation.

II. DYNAMIC-ADAPT-QAOA

In this section, we introduce Dynamic-ADAPT-QAOA. Our presentation strategy is to first review the standard ADAPT-QAOA template. Subsequently, we describe its improvement via Dynamic-ADAPT-QAOA.

A. ADAPT-QAOA

As depicted in Fig. 1, ADAPT-QAOA grows the ansatz circuit in P steps. In each step p, unitary evolutions generated by $\widehat{H}$ and $\widehat{A}_p$ are appended to the circuit from the previous step:

$$\widehat{U}_p(\vec{\beta}_p, \vec{\gamma}_p) = e^{-i\beta_p \widehat{A}_p} e^{-i\gamma_p \widehat{H}} \widehat{U}_{p-1}(\vec{\beta}_{p-1}, \vec{\gamma}_{p-1}). \tag{5}$$

The process starts from $\widehat{U}_0 = \widehat{1}$. Concurrently, the real parameter vectors are updated as

$$\vec{\beta}_p = (\beta_p, \vec{\beta}_{p-1}) \quad \text{and} \quad \vec{\gamma}_p = (\gamma_p, \vec{\gamma}_{p-1}), \tag{6}$$

starting from empty vectors $\vec{\beta}_0 = ()$ and $\vec{\gamma}_0 = ()$. In each step, an optimal mixer Hamiltonian $\widehat{A}_p$ is picked from a pool $\mathcal{P}$ such that the energy gradient is maximized (see below). The circuit parameters are then optimized,

$$\vec{\beta}_p^\star, \vec{\gamma}_p^\star = \text{argmin}_{\vec{\beta}_p, \vec{\gamma}_p} [E_p(\vec{\beta}_p, \vec{\gamma}_p)], \tag{7}$$

to minimize the energy expectation value

$$E_p(\vec{\beta}_p, \vec{\gamma}_p) = \langle \Psi_0 | \, \widehat{U}_p^\dagger(\vec{\beta}_p, \vec{\gamma}_p) \widehat{H} \widehat{U}_p(\vec{\beta}_p, \vec{\gamma}_p) \, | \Psi_0 \rangle \, . \tag{8}$$

This yields an upper bound $E_p^\star = E_p(\vec{\beta}_p^\star, \vec{\gamma}_p^\star)$ on the ground-state energy E_0, and an optimal trial state $|\Psi_p^\star\rangle \equiv \widehat{U}_p(\vec{\beta}_p^\star, \vec{\gamma}_p^\star)|\Psi_0\rangle$. Iterating this process provides a hierarchy of bounds $E_0^\star > E_1^\star > \cdots > E_p^\star > \cdots \geqslant E_0$. The algorithm terminates when $p = P$ or if $|E_{p-1}^\star - E_p^\star|$ falls below a predefined threshold ε.

To accelerate convergence, ADAPT-QAOA picks the mixer Hamiltonian which maximizes the energy gradient. To evaluate this gradient, the optimal trial state is augmented by appending a cost and a mixer unitary:

$$|\Psi_p(\beta_p, \gamma_p; \widehat{A})\rangle = e^{-i\beta_p \widehat{A}} e^{-i\gamma_p \widehat{H}} |\Psi_{p-1}^\star\rangle. \tag{9}$$

The energy variation due to the added parameters

$$\delta E_p(\beta_p, \gamma_p; \widehat{A}) = \langle \Psi_p(\beta_p, \gamma_p; \widehat{A}) | \widehat{H} | \Psi_p(\beta_p, \gamma_p; \widehat{A}) \rangle \tag{10}$$

enables the definition of a corresponding energy gradient:

$$\begin{aligned} G_p(\gamma_p; \widehat{A}) &\equiv \frac{\partial}{\partial \beta_p} \delta E_p(\beta_p, \gamma_p; \widehat{A}) \Big|_{\beta_p = 0} \\ &= \langle \Psi_{p-1}^\star | e^{i\gamma_p \widehat{H}} i[\widehat{A}, \widehat{H}] e^{-i\gamma_p \widehat{H}} | \Psi_{p-1}^\star \rangle. \end{aligned} \tag{11}$$

Evaluating this gradient for each $\widehat{A} \in \mathcal{P}$ allows for selecting the optimal mixer:

$$\widehat{A}_p = \text{argmax}_{\widehat{A} \in \mathcal{P}} [|G_p(\gamma_p; \widehat{A})|]. \tag{12}$$

Throughout this work, we use the same mixer pool as in ADAPT-QAOA [20],

$$\mathcal{P} = \mathcal{P}_0 \cup \mathcal{P}_1 \cup \mathcal{P}_2, \tag{13}$$

comprising QAOA mixers ($\mathcal{P}_0$) as well as Pauli strings of length one ($\mathcal{P}_1$) and length two ($\mathcal{P}_2$):

$$\begin{aligned} \mathcal{P}_0 &= \left\{ \sum_{n=1}^N \widehat{X}_n, \sum_{n=1}^N \widehat{Y}_n \right\}, \\ \mathcal{P}_1 &= \bigcup_{n=1}^N \{\widehat{X}_n, \widehat{Y}_n\}, \\ \mathcal{P}_2 &= \bigcup_{\substack{n,n'=1 \\ n \neq n'}}^N \{\widehat{X}_n, \widehat{Y}_n\} \times \{\widehat{X}_{n'}, \widehat{Y}_{n'}, \widehat{Z}_{n'}\}. \end{aligned} \tag{14}$$

B. Dynamic-ADAPT-QAOA

Motivation. Our motivation for developing Dynamic-ADAPT-QAOA comes from two observations. First, in each step p, the quantum circuit representing the cost unitary $e^{-i\gamma_p \widehat{H}}$ requires $O(N^2)$ CNOT gates (see Appendix A). On the other hand, the quantum circuit representing the mixer unitary $e^{-i\beta_p \widehat{A}_p}$ requires only $O(1)$ CNOT gates [21]. As CNOT gates induce noise, minimizing the number of cost unitaries in the ansatz circuit could be valuable [22]. Second, in standard ADAPT-QAOA, the vector of optimal parameters $\vec{\gamma}_p^\star$ tends to be sparse, with many parameters taking values close to zero (see Sec. III B). As cost unitaries $e^{-i\gamma_p \widehat{H}}$ with $\gamma_p \approx 0$ hardly affect the final quantum circuit, it could be advantageous to exclude them altogether.

Idea. In general, the energy expectation value in Eq. (8) is a nontrivial function of the circuit parameters. Hence, it is not obvious how to predict which entries in $\vec{\gamma}_p^\star$ would take optimal values close to zero. Yet, in ADAPT-QAOA, optimal circuit parameters of the pth iteration are usually well approximated by the circuit parameters of the previous iteration:

$$\vec{\beta}_p^\star \approx (\beta_p^\star, \vec{\beta}_{p-1}^\star) \quad \text{and} \quad \vec{\gamma}_p^\star \approx (\gamma_p^\star, \vec{\gamma}_{p-1}^\star). \tag{15}$$

Thus, we can estimate the optimal circuit parameters $\beta_p^\star, \gamma_p^\star$ of the pth iteration, by studying the minima of

$$\delta E_p(\beta_p, \gamma_p) \equiv \delta E_p(\beta_p, \gamma_p; \widehat{A}_p). \tag{16}$$

As explained in Appendix B, for Pauli-string mixers $\widehat{A}_p$, we can identify whether $\delta E_p(\beta_p, \gamma_p)$ has minima near $\gamma_p^\star = 0$. To this end, we split the cost Hamiltonian into two parts $\widehat{H} = \widehat{H}_- + \widehat{H}_+$, such that $\widehat{H}_-$ commutes and $\widehat{H}_+$ anticommutes with $\widehat{A}_p$. This enables the evaluation of three additional expectation values:

$$B_p = \langle \Psi_{p-1}^\star | i\widehat{A}_p \widehat{H}_+ | \Psi_{p-1}^\star \rangle \equiv G_p(0; \widehat{A}_p)/2, \tag{17a}$$

$$C_p = \langle \Psi_{p-1}^\star | \widehat{A}_p \widehat{H}_+^2 | \Psi_{p-1}^\star \rangle, \tag{17b}$$

$$D_p = \left\langle \Psi_{p-1}^\star \middle| i\widehat{A}_p \widehat{H}_+^3 \middle| \Psi_{p-1}^\star \right\rangle. \tag{17c}$$

As shown in Appendix B, $\delta E_p(\beta_p, \gamma_p)$ has a local minimum at $\gamma_p^\star = 0$ if

$$C_p = 0 \quad \text{and} \quad B_p D_p > 0. \tag{18}$$

Algorithm. Dynamic-ADAPT-QAOA excludes the cost unitary of the pth iteration, if $\widehat{A}_p$ is a Pauli string and condition (18) holds. Otherwise, the algorithm follows the standard mixer-selection procedure of ADAPT-QAOA. That is, if $\widehat{A}_p$ is not a Pauli-string but a standard QAOA mixer, the ansatz circuit and parameter vectors are grown as described in Eqs. (5) and (6). On the other hand, if $\widehat{A}_p$ is a Pauli string that does not satisfy condition (18) the gradients for all $\widehat{A} \in \mathcal{P}$ are reevaluated at some given offset $\gamma_p = \pm\tilde{\gamma}$, and the optimal mixer is redetermined:

$$\widehat{A}_p = \text{argmax}_{\widehat{A}\in\mathcal{P}}[\max(|G_p(+\tilde{\gamma}; \widehat{A})|, |G_p(-\tilde{\gamma}; \widehat{A})|)]. \tag{19}$$

After redetermining $\widehat{A}_p$, the ansatz circuit and parameter vectors are grown as described in Eqs. (5) and (6). Pseudocode summarizing Dynamic-ADAPT-QAOA is given in Algorithm 1.

ALGORITHM 1. Dynamic-ADAPT-QAOA.

Init pool $\mathcal{P}$; accuracies $\varepsilon, \delta_1, \delta_2$; and offset $\tilde{\gamma}$.
Init $p \leftarrow 0$.
Set initial state $|\Psi_0^\star\rangle \leftarrow |+\rangle \cdots |+\rangle$;
Init parameters $\vec{\beta}_0 \leftarrow ()$, $\vec{\gamma}_0 \leftarrow ()$.
Set optimal parameters $\vec{\beta}_0^\star \leftarrow ()$, $\vec{\gamma}_0^\star \leftarrow ()$ and unitary $\widehat{U}_0 \leftarrow \widehat{I}$.
while not converged **do**
  $p \leftarrow p + 1$
  //Select mixer with maximal gradient
  $\forall \widehat{A} \in \mathcal{P}$ evaluate $G_p(0; \widehat{A}) \leftarrow \langle \Psi_{p-1}^\star | [i\widehat{A}, \widehat{H}] | \Psi_{p-1}^\star \rangle$
  Select optimal mixer: $\widehat{A}_p \leftarrow \text{argmax}_{\widehat{A}\in\mathcal{P}}[|G_p(0; \widehat{A})|]$
  if $\widehat{A}_p$ is a Pauli string in $\mathcal{P}_1 \cup \mathcal{P}_2$ **then**
    //Test, if the cost unitary is needed
    Evaluate B_p, C_p, D_p in Eq. (17)
    if $|C_p| \leqslant \delta_1$ **and** $B_p \cdot D_p > \delta_2$ **then**
      //Append only mixer unitary to ansatz circuit
      Update $\vec{\gamma}_p \leftarrow \vec{\gamma}_{p-1}$; $\vec{\beta}_p \leftarrow (\beta_p, \vec{\beta}_{p-1})$
      $\widehat{U}_p(\vec{\beta}_p, \vec{\gamma}_p) \leftarrow e^{-i\beta_p \widehat{A}_p} \widehat{U}_{p-1}(\vec{\beta}_{p-1}, \vec{\gamma}_{p-1})$
    else *//Default back to ADAPT-QAOA*
      Update $|\tilde{\Psi}_p^\pm\rangle \leftarrow \exp(\mp i\tilde{\gamma}\widehat{H})|\Psi_{p-1}^\star\rangle$
      $\forall \widehat{A} \in \mathcal{P}$ redo $G_p(\pm\tilde{\gamma}, \widehat{A}) \leftarrow \langle \tilde{\Psi}_p^\pm | [i\widehat{A}, \widehat{H}] | \tilde{\Psi}_p^\pm \rangle$
      Select $\widehat{A}_p \leftarrow \text{argmax}_{\widehat{A}\in\mathcal{P}}[|G_p(\pm\tilde{\gamma}, \widehat{A})|]$
      //Append mixer and cost unitary to ansatz
      Update $\vec{\gamma}_p \leftarrow (\gamma_p, \vec{\gamma}_{p-1})$; $\vec{\beta}_p \leftarrow (\beta_p, \vec{\beta}_{p-1})$
      $\widehat{U}_p(\vec{\beta}_p, \vec{\gamma}_p) \leftarrow e^{-i\beta_p \widehat{A}_p} e^{-i\gamma_p \widehat{H}} \widehat{U}_{p-1}(\vec{\beta}_{p-1}, \vec{\gamma}_{p-1})$
  else if $\widehat{A}_p$ is a QAOA mixer in $\mathcal{P}_0$ **then**
    //Append standard QAOA mixer to ansatz circuit
    Update $\vec{\gamma}_p \leftarrow (\gamma_p, \vec{\gamma}_{p-1})$; $\vec{\beta}_p \leftarrow (\beta_p, \vec{\beta}_{p-1})$
    $\widehat{U}_p(\vec{\beta}_p, \vec{\gamma}_p) \leftarrow e^{-i\beta_p \widehat{A}_p} e^{-i\gamma_p \widehat{H}} \widehat{U}_{p-1}(\vec{\beta}_{p-1}, \vec{\gamma}_{p-1})$
  //Optimize ansatz circuit and update bound
  Optimize parameters $\vec{\beta}_p^\star, \vec{\gamma}_p^\star \leftarrow \text{argmax}_{\vec{\beta}_p, \vec{\gamma}_p}[E_p(\vec{\beta}_p, \vec{\gamma}_p)]$
  Update ansatz state $|\Psi_p^\star\rangle \leftarrow \widehat{U}_p(\vec{\beta}_p^\star, \vec{\gamma}_p^\star)|\Psi_0\rangle$
  Repeatedly create $|\Psi_p^\star\rangle$ and sample bit strings
  Compute $E_p^\star \leftarrow E_p(\vec{\beta}_p^\star, \vec{\gamma}_p^\star)$ from bit strings
  //Check convergence
  if $p = P$ **or** $|E_{p-1}^\star - E_p^\star| < \varepsilon$ **then**
    converged $\leftarrow$ True
Return bit strings, $E_p^\star$, circuit $\widehat{U}_p$, params $\vec{\beta}_p^\star, \vec{\gamma}_p^\star$

Remarks I (Alternative versions). In Appendix C, we discuss two alterations of Dynamic-ADAPT-QAOA. These are alterations that could save quantum resources. However, they are detrimental to the performance of our algorithm. In the first alteration, all cost unitaries are, *a priori*, removed from the ansatz circuit. In the second alteration, the algorithm does not reevaluate the optimal mixer $\widehat{A}_p$ at $\gamma_p = \pm\tilde{\gamma}$ if condition (18) fails. As shown in Appendix C, both of these alterations prevent the algorithm from converging towards highly accurate solutions.

Remarks II (Barren plateaus). Common worries regarding variational quantum algorithms concern barren plateaus (vanishing gradients) and the presence of bad local minima [23–30]. A promising way to mitigate these issues is to reduce the circuit depths [30,31], which is precisely what our algorithm does. Moreover, since the gates of adaptive variational

quantum algorithms are tailored to the optimization problem itself, there are indications that these algorithms avoid such issues better than other variational quantum algorithms [30,32–35]. In the instances studied below, Dynamic-ADAPT-QAOA efficiently implements the variational optimization.

Remarks III (Shot-based implementation). In Algorithm 1 we formulate Dynamic-ADAPT-QAOA, based on expectation values. However, on a real quantum computer, expectation values must be evaluated by sampling. In order to bridge this gap (along with other intricacies, such as dimensionless units), we describe a more practical shot-based implementation of Dynamic-ADAPT-QAOA in Appendix D, Algorithm 2.

Remarks IV (Sampling overhead). For Dynamic-ADAPT-QAOA one may worry that evaluating C_p and D_p in each iteration of the algorithm may add significant sampling overhead. In Appendix D, we show that this is not the case. By analyzing various sampling strategies [36–39] we calculate upper bounds for the number of samples needed to sample expectations corresponding to energies, gradients, C_p and D_p, with a given precision. Order estimates for Max-Cut problems are presented in Appendix D 5. Our results indicate that $O(f(N))$ samples are required to estimate the energy parameters line 29 of Algorithm 1. [Here $O(f(N))$ indicates the number of times an optimizer needs to evaluate the energy expectation.] Our results further show that measuring all the energy gradients in lines 8 and 19 of Algorithm 1, will require $O(N^2)$ samples. As compared to these estimates the need for $O(1)$ samples to measure C_p and D_p should be negligible.

III. BENCHMARKING

In this section, we benchmark Dynamic- and standard ADAPT-QAOA in numerical simulations. Our investigation will demonstrate that Dynamic-ADAPT-QAOA can remove redundant components from the ansatz circuits of standard ADAPT-QAOA. We show that this leads to a reduced CNOT count and an increased noise resilience.

A. Benchmarking methodology

Max-Cut. In what follows, we benchmark ADAPT-QAOAs on random instances of weighted Max-Cut problems. Consider allocating weights to the edges of an N-vertex graph. In this work, we consider complete, i.e., fully connected, graphs. The edge weights between vertices $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, N\}$ form a real symmetric matrix W_{ij} with zeros on its diagonal. A binary vector $\vec{b} \in \{0, 1\}^N$ defines a *cut*, a splitting of all vertices into two disjoint sets. A cut value is defined as the sum of edge weights between the two partitions:

$$V(\vec{b}) = \sum_{i,j=1}^{N} W_{ij} b_i (1 - b_j). \tag{20}$$

The weighted Max-Cut problem is to find the binary vector $\vec{b}^\star$ that maximizes the cut value: $\vec{b}^\star = \mathrm{argmax}_{\vec{b}} V(\vec{b})$. $\vec{b}^\star$ corresponds to the optimal partition, which yields the maximal cut value $V_{\max} = V(\vec{b}^\star)$. By mapping binary variables $b_i = (1 + z_i)/2$ to the eigenvalues $z_i \in \{-1, 1\}$ of $\widehat{Z}_i$, the weighted Max-Cut problem becomes equivalent to finding the ground state of the Ising model, Eq. (1). We create random Max-Cut instances by uniformly sampling edge weights $W_{ij} \in [0, 1]$. This is known to generate NP-hard problems [40,41]. For a visualization of Max-Cut, see Fig. 2.

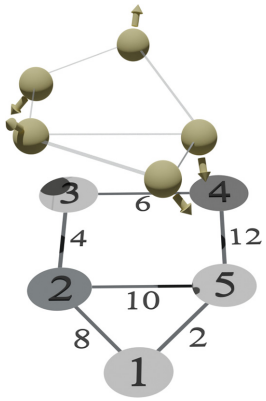

FIG. 2. Diagramatic representation of a five-vertex weighted graph. The vertices are labeled 1–5. The weights are shown next to the corresponding edges. The partition resulting in a Max-Cut, (135)(24), is depicted using different shades of gray. The Max-Cut value is 40. Directly above the graph we illustrate how the problem maps onto a qubit system. The qubits' spins point in different vertical half-planes, corresponding to which set of the Max-Cut partition they are in.

Approximation ratio. Our benchmarks compare the average-case solution accuracy of three algorithms: The average-case solution accuracy of Dynamic- and standard ADAPT-QAOA, as well as the average-case solution accuracy of the classical, polynomial-time approximation algorithm by Goemans and Williamson (GW). Rather than solving Max-Cut exactly, all three algorithms sample a collection of bit strings [42]. This leads to a distribution of approximate cut values, Eq. (20), with average cut values V_{d}, V_{s}, and V_{GW}, respectively. Algorithms providing a higher average cut value tend to provide better-quality solutions. Further, normalizing the average cut value by the maximal achievable value $V_{\max}$ allows for averaging various instances of Max-Cut. This defines our key performance metric—the average-case approximation ratio:

$$\alpha_{\mathrm{d}} \equiv \frac{V_{\mathrm{d}}}{V_{\max}}, \alpha_{\mathrm{s}} \equiv \frac{V_{\mathrm{s}}}{V_{\max}}, \text{ and } \alpha_{\mathrm{GW}} \equiv \frac{V_{\mathrm{GW}}}{V_{\max}}. \tag{21}$$

The GW algorithm is the classical, polynomial-time algorithm that achieves the best worst-case approximation ratio of $87.8\ldots\%$ [14]. Its average-case approximation ratio, which will be considered in this paper, typically reaches even better values: $\alpha_{\mathrm{GW}} > 87.8\ldots\%$ [14]. Below, we will compare the average-case approximation ratio of the GW algorithm α_{GW} to numerically computed values of α_{d} and α_{s}. In our simulations, we average the results over 100 random instances of the Max-Cut problem. In real applications of the QAOA, one would return the cut corresponding to the sampled bit string with minimum cost, not the average. However, in the small problem sizes studied here, the final wavefunction has substantial overlap with all bit strings. Thus, for a relatively small number of shots the true solution will always be obtained. Therefore, we compare the average approximation ratios. Further, we emphasize that our comparison between QAOAs and the GW

algorithm focuses on comparing solution accuracy in terms of average-case approximation ratios, as opposed to comparing their computational time complexity.

Simulations. To assess the average-case approximation ratios of Dynamic- and standard ADAPT-QAOA in the presence of noise, we use full density-matrix simulations, as previously described in Ref. [34]. First, the unitaries in Eq. (5) are compiled to standard circuit representations [21]. To simulate the effect of noise, we work with density matrices. In the evolution of the quantum states, we apply a depolarizing channel after each CNOT gate:

$$\mathcal{D}(i, p_{\text{gate}})[\rho] : (1 - p_{\text{gate}})\rho + \frac{p_{\text{gate}}}{3} \sum_{\widehat{P}} \widehat{P}_i \rho \widehat{P}_i. \tag{22}$$

Here ρ is the density matrix after the CNOT gate, i denotes the target qubit of the CNOT gate, $p_{\text{gate}} \in [0, 1]$ denotes the gate-error probability, and the $\widehat{P}_i$-summation is over the three Pauli matrices acting on qubit i. Owing to the diverse nature of current quantum hardware, a noise model cannot be both platform agnostic and realistically detailed. Nevertheless, our noise model captures the depolarizing effect of two-qubit gates, which is the dominant noise source across several platforms [43,44]. We deem our model a reasonably hardware-agnostic compromise, which should be sufficient to assess fundamental quantitative features.

Since full density-matrix simulations require extensive computing time, we apply an approximation similar to that outlined in Ref. [34]. In more detail, we simulate ADAPT-QAOAs by growing their ansatz circuits in the absence of noise. We store the optimal ansatz circuits U_p at each iteration step p. Subsequently, we investigate the effect of noise by simulating the preoptimized circuit U_p at various noise levels p_{gate} on our density matrix simulator. As demonstrated in Appendix E, the noiseless-growth approximation has little effect on our results.

Parameters. Before presenting our findings, we specify the hyperparameters used in our simulations. By setting $\varepsilon = 0$, we ensure that the convergence criterion corresponds to having reached a certain circuit depth. The depth is determined by the number of iterations, which we set to $P = 12$. For Dynamic-ADAPT-QAOA, the cost-unitary offset (see Algorithm 1) was set to $\tilde{\gamma} = 0.1$, following the settings used in [20]. In Algorithm 1, $\delta_1 > 0$ would mitigate some experimental errors in the identification of a local minimum where, in ideal scenarios, $C_p = 0$. Similarly, $\delta_2 > 0$ would mitigate some experimental errors in establishing whether $B_p \cdot D_p$ is positive. In our simulations, we set $\delta_1 = 0$. To emulate practical implementations, we choose $\delta_2 \in (0,\ 10^{-4})$ after performing a hyperparameter search for each separate graph.

B. Vanishing cost parameters

As mentioned in Sec. II B, our motivation to develop Dynamic-ADAPT-QAOA stems from the observation that standard ADAPT-QAOA appends cost unitaries to the quantum circuit in cases where they do not lead to any significant improvement in convergence. In Fig. 3 we show data which support this conclusion. The histogram of optimal cost parameters $\gamma^\star$ of standard ADAPT-QAOA exhibits a well-defined peak at $\gamma^\star = 0$. A majority ($\approx$70%) of the cost unitaries do not contribute to the algorithm's convergence. This peak is absent in the corresponding histogram for Dynamic-ADAPT-QAOA: Our algorithm successfully removes redundant cost unitaries from the ansatz circuits.

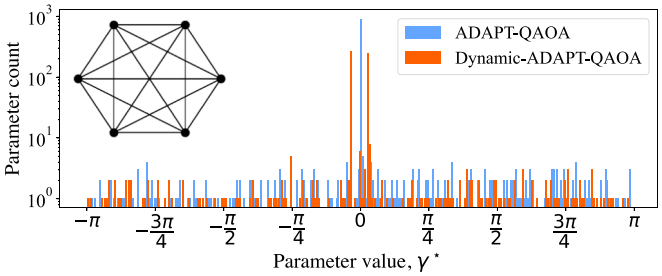

FIG. 3. Histogram of optimized circuit parameters $\gamma_p^\star$, taken from the cost unitaries from all layers of the ansatz circuits grown with Dynamic- and standard ADAPT-QAOA. The data were acquired in noiseless simulations of 100 instances of Max-Cut on six-vertex graphs. The algorithms were run until a maximum circuit depth of $P = 12$.

C. Benchmarking the CNOT-count reduction

Now, we show that Dynamic-ADAPT-QAOA significantly reduces the number of CNOT gates needed to reach a certain algorithmic precision. In Sec. II we described how Dynamic-ADAPT-QAOA prunes unnecessary circuit elements. To investigate the effect on the CNOT count, we consider how the average-case approximation ratio α, averaged over 100 instances of Max-Cut, improves as the algorithm grows the quantum circuit. Our results are shown in Fig. 4. We plot data from both noiseless and noisy simulations of Dynamic- and standard ADAPT-QAOA. In both scenarios, Dynamic-ADAPT-QAOA uses significantly fewer CNOT gates to reach a fixed average-case approximation ratio. For a fixed gate-error probability this CNOT reduction allows Dynamic-ADAPT-QAOA to calculate more accurate average-case approximation ratios than standard ADAPT-QAOA. In noiseless simulations, we see that Dynamic-ADAPT-QAOA needs approximately 80% fewer CNOT gates than ADAPT-QAOA to calculate average-case approximation ratios that outperform the average-case approximation ratios achievable with the classical GW algorithm for six-vertex complete graphs. Moreover, at a gate-error probability of $p_{\text{gate}} = 0.122\%$, the Dynamic-ADAPT-QAOA can achieve better average-case approximation ratios than the average-case approximation ratios of the GW algorithm, while the standard ADAPT-QAOA cannot. In the next section, we widen our analysis of how noise affects the quantum algorithms' achieved average-case approximation ratios.

D. Benchmarking the noise resilience

In this section, we analyze how noise affects the quality of average-case approximation ratios of Dynamic- and standard ADAPT-QAOA. The convergence curves presented in Fig. 4 show that increasing the gate-error probability p_{gate} worsens the best attainable average-case approximation ratio $\alpha^\star$. More

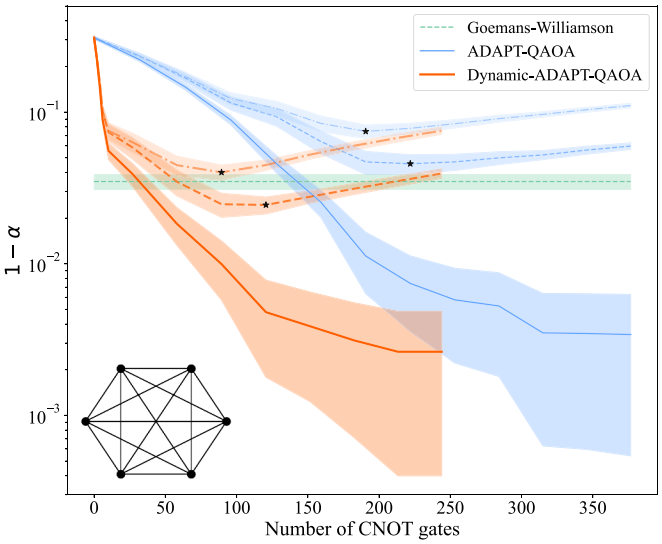

FIG. 4. Convergence curves for the Dynamic- and standard ADAPT-QAOA, applied to six-vertex complete graphs, with and without noise. $1-\alpha$ is plotted as a function of the number of CNOT gates present in the ansatz circuits U_P. The dashed horizontal curve corresponds to the average-case approximation ratio of the classical GW algorithm. The shaded regions correspond to the 95% confidence intervals. The convergence curves for three gate-error probabilities are shown: $p_{\text{gate}} = 0.0\%$, 0.122%, and 0.263%. These are depicted using solid, dashed, and dash-dotted line styles, respectively. Stars indicate the maximally attainable average-case approximation ratio $\alpha^\star$.

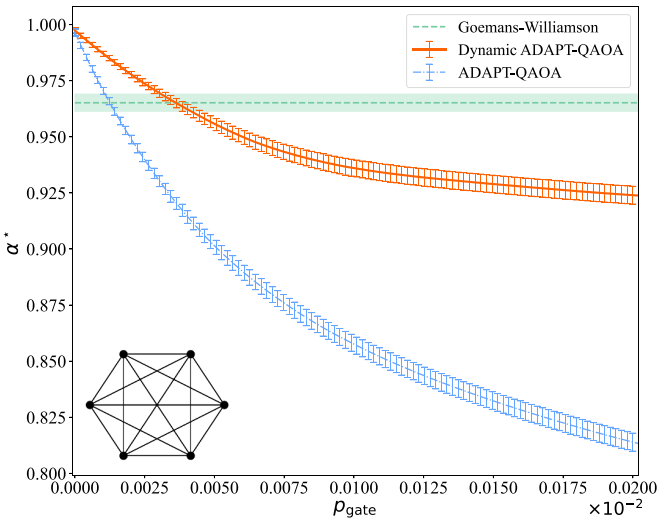

FIG. 5. Best attainable average-case approximation ratio $\alpha^\star$ as a function of the gate-error probability p_{gate}. The data were acquired in noisy simulations of six-vertex graphs. The error bars show the standard error in the mean average-case approximation ratio. The dashed curve corresponds to the mean average-case approximation ratio of the classical GW algorithm. The shaded regions correspond to the 95% confidence intervals.

specifically, as ADAPT-QAOA grows the circuit (leading to an increase of CNOT gates on the abscissa) the average-case approximation ratio improves initially. However, as the circuit acquires more CNOT gates, the effect of noise starts to dominate, leading to a subsequent deterioration of the average-case approximation ratio. This causes the characteristic "smirk" shape of the convergence curves in Fig. 4. The dip of each convergence curve marks the best attainable average-case approximation ratio $\alpha^\star$ at a certain gate-error probability p_{gate}.

Figure 4 indicates that Dynamic-ADAPT-QAOA outperforms the average-case solution quality of standard ADAPT-QAOA in the presence of noise. To quantify this benefit of our algorithm, we investigate $\alpha^\star$ as a function of p_{gate} in Fig. 5. For all values of p_{gate}, Dynamic-ADAPT-QAOA calculates better average-case approximation ratios than standard ADAPT-QAOA. Evidently, our algorithm exhibits better noise resilience.

As can be seen from the leftmost portion of Fig. 5, given sufficiently weak noise, both Dynamic- and standard ADAPT-QAOA can provide better average-case approximation ratios than the average-case approximation ratio of the GW algorithm. We now investigate the range of gate-error probabilities for which Dynamic- and standard ADAPT-QAOAs achieve such an improvement. To this end, we define the gate-error probability $p^\star_{\text{gate}}$, below which the quantum algorithms achieve a better average-case approximation ratio than the average-case approximation of the GW algorithm. In Fig. 6 we plot $p^\star_{\text{gate}}$ with respect to the number of graph vertices. Compared to standard ADAPT-QAOA, Dynamic-ADAPT-QAOA can achieve a better average-case approximation ratio than the average-case approximation ratio of the classical GW algorithm for Max-Cut at roughly an order of magnitude larger values of $p^\star_{\text{gate}}$. In particular, the critical probability at which Dynamic-ADAPT-QAOA achieves higher average-case approximation ratios than the average-case approximation ratio of the GW algorithm is $p^\star_{\text{gate}} = 1.3 \pm 0.2\%$ for six-vertex graphs and $p^\star_{\text{gate}} = 0.13 \pm 0.05\%$ for ten-vertex graphs. Both of these values are well above achieved gate-error probabilities [45], implying that for small Max-Cut problems with less than ten vertices Dynamic-ADAPT-QAOA may, on average, achieve better quality approximate solutions

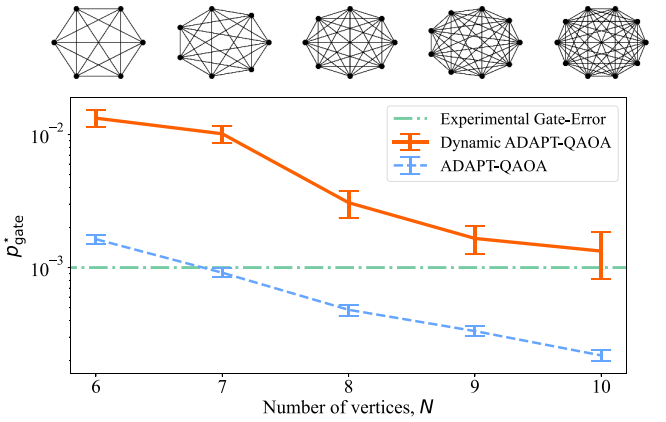

FIG. 6. $p^\star_{\text{gate}}$ with respect to different graph sizes. At gate-error probabilities below $p^\star_{\text{gate}}$ the quantum algorithms outperform the average-case solution quality of the classical GW algorithm. The horizontal line shows the experimentally achieved two-qubit gate-error probability in state-of-the-art superconducting hardware [45]. The error bars show the standard error.

than the GW algorithm on existing hardware. On the other hand, for standard ADAPT-QAOA, the critical probability to generate on average higher quality solutions than the GW algorithm on existing hardware is currently achievable only for graphs with less than seven vertices.

IV. DISCUSSION

We have introduced Dynamic-ADAPT-QAOA, a quantum algorithm for combinatorial optimization. Similar to the original ADAPT-QAOA algorithm, our algorithm variationally approximates the ground state of an Ising Hamiltonian. Thus, it can provide approximate solutions to NP problems. By dynamically assessing the importance of unitaries before they are added in the variationally grown algorithms, Dynamic-ADAPT-QAOA can operate with remarkably few CNOT gates. Above, we benchmarked the average (as opposed to the worst-case) solution accuracy of our algorithm. For example, in the idealized case of no noise, Dynamic-ADAPT-QAOA requires on average about 35 (350) CNOT gates to achieve better average-case solution accuracy than the GW algorithm on six-vertex (ten-vertex) graphs. Moreover, we have shown that for graphs with six to ten vertices, Dynamic-ADAPT-QAOA can, on average, provide more accurate solutions than the GW algorithm, even in the presence of noise levels comparable with current state-of-the-art hardware [45]. This should make Dynamic-ADAPT-QAOA an attractive candidate to showcase proof-of-principle computations on NISQ hardware. Finally, we conclude this work with a few comments.

Other QAOAs. There are plenty of promising QAOA and VQE-inspired algorithms in the literature [15,46–55]. However, this work focuses on ADAPT-QAOAs [20]—mainly due to their relatively shallow ansatz circuits. In the future, it would be of interest to expand the benchmarks of noise resilience to other types of QAOA. It would also be of interest to augment Dynamic-ADAPT-QAOA by warm starting [56] to get better than classical approximation ratios at even lower CNOT count with even higher noise resilience.

Other algorithms. This study focuses on investigating the utility of gate-based quantum computers for solving NP-problems. However, adiabatic quantum computers [10–13] and state-of-the-art annealing heuristics [6–9,57] can comfortably handle systems with up to 5000 and 100 000 spins, respectively, most likely at a higher solution accuracy. Moreover, other approximation algorithms [58,59] could also lead to high average solution accuracy. This shows that QAOA still has a long way to go before reaching practical quantum advantage.

Error mitigation. Applying error-mitigation techniques [60–63] to boost expectation values would straightforwardly improve the average-case approximation ratios of standard and Dynamic-ADAPT-QAOA; see Appendix F. However, to the best of our knowledge, error-mitigation methods have never been used to improve the underlying bit strings. Consequently, error-mitigation methods would not improve the cut value provided by the experimentally accessible bit strings. An interesting direction of future research is to consider how error-mitigation techniques could be used to improve not only the cut value, but also the bit strings provided by a QAOA.

Code to run the algorithms and reproduce the plots of this paper is publicly available at [64].

ACKNOWLEDGMENTS

We thank Kieran Dalton, Yordan Yordanov, Bobak Kiani, Nicholas Mayhall, Sophia Economou, Edwin Barnes, and members of the Hitachi QI team for useful discussions.

APPENDIX A: CNOT GATE COUNT IN COST UNITARIES

In this Appendix we show that the number of CNOT gates included in implementations of the cost unitaries in QAOA scales quadratically with the number of vertices for complete graphs. This forms part of the motivation for our development of Dynamic-ADAPT-QAOA.

The general form of a parameterized cost unitary is $e^{-i\gamma\widehat{H}}$, where the Ising Hamiltonian is defined in Eq. (1). For complete graphs, $\widehat{H}$ is a summation of $O(N^2)$ two-qubit Pauli strings, where N equals the number of vertices. As all of these Pauli strings are of the same general form, namely with two z-measurements, they all commute with one another. Hence, we can rewrite the cost unitary as a product of $O(N^2)$ RZZ gates:

$$e^{-i\gamma\widehat{H}} = \prod_{i,j=1}^{N} \exp\left(-i\frac{\gamma W_{ij}}{4}\widehat{Z}_i\widehat{Z}_j\right). \tag{A1}$$

Each of these RZZ gates can be implemented using two CNOT gates. Therefore, we conclude that each cost unitary included in the parameterized unitary of the algorithm contributes $O(N^2)$ CNOT gates to the quantum circuit decomposition.

APPENDIX B: THEORETICAL ASPECTS UNDERLYING DYNAMIC-ADAPT-QAOA

In this Appendix we provide further details on the analysis of minima in the energy variation, Eq. (16).

1. Splitting of cost Hamiltonians

To begin, we note that any cost Hamiltonian $\widehat{H}$, Eq. (1), can be decomposed into two parts, $\widehat{H}_-$ and $\widehat{H}_+$, which commute and anticommute with a given Pauli-string mixer $\widehat{A}$, respectively. To show this, denote the qubit indices as a vertex set $\mathcal{V} := \{1, \ldots, N\}$ with the corresponding edge set given as $\mathcal{E} = \{(i, j) \in \mathcal{V} \times \mathcal{V} \,|\, i < j\}$. This allows for writing the cost Hamiltonian, Eq. (1), as

$$\widehat{H} = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} W_{ij}\widehat{Z}_i\widehat{Z}_j. \tag{B1}$$

To split the Hamiltonian, we split the edge set $\mathcal{E}$ into two disjoint subsets $\mathcal{E}_-$ and $\mathcal{E}_+$, such that

$$\mathcal{E} = \mathcal{E}_- \cup \mathcal{E}_+ \quad \text{and} \quad \mathcal{E}_- \cap \mathcal{E}_+ = \emptyset. \tag{B2}$$

This defines two Hamiltonians

$$\widehat{H}_- = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}_-} W_{ij}\widehat{Z}_i\widehat{Z}_j \quad \text{and} \quad \widehat{H}_+ = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}_+} W_{ij}\widehat{Z}_i\widehat{Z}_j, \tag{B3}$$

such that

$$\widehat{H} = \widehat{H}_- + \widehat{H}_+. \tag{B4}$$

For later reference, we note that $\widehat{H}$, $\widehat{H}_-$, and $\widehat{H}_+$ commute.

For mixer Hamiltonians $\widehat{A} \in \mathcal{P}$, which are Pauli strings of length one, acting on qubit $n = 1, \ldots, N$ via $\widehat{X}_n$ or $\widehat{Y}_n$, we partition the edge set of $\widehat{H}$ as follows:

$$\mathcal{E}_-(n) = \{(i, j) \in \mathcal{E} \quad | \quad \text{if neither } i \text{ nor } j \text{ is } n\}, \tag{B5}$$

$$\mathcal{E}_+(n) = \{(i, j) \in \mathcal{E} \quad | \quad \text{if either } i \text{ or } j \text{ is } n\}. \tag{B6}$$

For mixer Hamiltonians $\widehat{A} \in \mathcal{P}$, which are Pauli strings of length two, acting on qubits $n, n' = 1, \ldots, N$, we partition the edge sets as follows: (1) If the Pauli string takes the form $\widehat{Z}_n \cdot \widehat{Z}_{n'}$, the partition of edges is trivial $\mathcal{E}_- = \mathcal{E}$ and $\mathcal{E}_+ = \emptyset$. (2) If the Pauli string is of the form $\widehat{X}_n \cdot \widehat{Z}_{n'}$, we use the partition of single Pauli strings acting on n. (3) If the Pauli string is of the form $\widehat{X}_n \cdot \widehat{X}_{n'}$, $\widehat{X}_n \cdot \widehat{Y}_{n'}$, or $\widehat{Y}_n \cdot \widehat{Y}_{n'}$, respectively, we partition the edge set of $\widehat{H}$ as follows:

$$\mathcal{E}_-(n, n') = \{(i, j) \in \mathcal{E} \quad | \quad \text{if } i \text{ and } j \text{ differ from } n \text{ and } n', \text{ or if } i = n \text{ and } j = n', \text{ or } j = n \text{ and } i = n'\}, \tag{B7}$$

$$\mathcal{E}_+(n, n') = \{(i, j) \in \mathcal{E} \quad | \quad \text{if either } i \text{ or } j \text{ is equal to } n \text{ or } n', \text{ but not both}\}. \tag{B8}$$

These partitions ensure that $\widehat{H}_-$ and $\widehat{H}_+$ commute or anticommute with Pauli strings $\widehat{A}$ of length one or two, respectively.

2. Energy variation

Next, we analyze the energy variation, Eq. (16), with respect to a mixer $\widehat{A}$, as defined by Eqs. (9) and (10):

$$\delta E_p(\beta_p, \gamma_p; \widehat{A}) = \left\langle \Psi^\star_{p-1} \middle| e^{i\gamma_p \widehat{H}} e^{i\beta_p \widehat{A}} \widehat{H} e^{-i\beta_p \widehat{A}} e^{-i\gamma_p \widehat{H}} \middle| \Psi^\star_{p-1} \right\rangle. \tag{B9}$$

To analyze this expression further we rewrite it as

$$\begin{aligned} \delta E_p(\beta_p, \gamma_p; \widehat{A}) = {} & \langle \Psi^\star_{p-1} | \widehat{H}_- | \Psi^\star_{p-1} \rangle \\ & + \cos(2\beta_p) \langle \Psi^\star_{p-1} | \widehat{H}_+ | \Psi^\star_{p-1} \rangle \\ & + \sin(2\beta_p) \langle \Psi^\star_{p-1} | i\widehat{A}\widehat{H}_+ e^{-i2\gamma_p \widehat{H}_+} | \Psi^\star_{p-1} \rangle \end{aligned} \tag{B10}$$

To derive this expression we used the following properties of Pauli strings $\widehat{A}$:

$$\widehat{1} = \widehat{A}^2 \quad \text{and} \quad e^{-i\beta_p \widehat{A}} = \widehat{1} \cos(\beta_p) + i\widehat{A} \sin(\beta_p). \tag{B11}$$

Moreover, we use that $\widehat{H} = \widehat{H}_- + \widehat{H}_+$ and the established commutation relations implying

$$\widehat{H}_- \widehat{A} = \widehat{A}\widehat{H}_-, \quad \text{and} \quad \widehat{H}_+ \widehat{A} = -\widehat{A}\widehat{H}_+, \tag{B12}$$

as well as

$$e^{-i\gamma_p \widehat{H}} = e^{-i\gamma_p \widehat{H}_+} e^{-i\gamma_p \widehat{H}_-} = e^{-i\gamma_p \widehat{H}_-} e^{-i\gamma_p \widehat{H}_+}. \tag{B13}$$

Next, defining the following quantities:

$$E_0^- = \langle \Psi^\star_{p-1} | \widehat{H}_- | \Psi^\star_{p-1} \rangle, \tag{B14a}$$

$$E_0^+ = \langle \Psi^\star_{p-1} | \widehat{H}_+ | \Psi^\star_{p-1} \rangle, \tag{B14b}$$

$$B(\gamma_p) = \langle \Psi^\star_{p-1} | i\widehat{A}\widehat{H}_+ e^{-i2\gamma_p \widehat{H}_+} | \Psi^\star_{p-1} \rangle, \tag{B14c}$$

$$C(\gamma_p) = \langle \Psi^\star_{p-1} | \widehat{A}\widehat{H}_+^2 e^{-i2\gamma_p \widehat{H}_+} | \Psi^\star_{p-1} \rangle = \tfrac{1}{2} \partial_{\gamma_p} B(\gamma_p), \tag{B14d}$$

$$D(\gamma_p) = \langle \Psi^\star_{p-1} | i\widehat{A}\widehat{H}_+^3 e^{-i2\gamma_p \widehat{H}_+} | \Psi^\star_{p-1} \rangle = -\tfrac{1}{2} \partial_{\gamma_p} C(\gamma_p), \tag{B14e}$$

one rewrites the energy fluctuation, Eq. (B10), as

$$\delta E_p(\beta_p, \gamma_p; \widehat{A}) = E_0^- + E_0^+ \cos(2\beta_p) + \sin(2\beta_p) B(\gamma_p). \tag{B15}$$

3. Gradients, stationary points, and Hessian of the energy variation

The gradient of the energy fluctuation are then given as

$$\partial_{\gamma_p} \delta E_p(\beta_p, \gamma_p; \widehat{A}) = 2 \sin(2\beta_p) C(\gamma_p), \tag{B16}$$

$$\partial_{\beta_p} \delta E_p(\beta_p, \gamma_p; \widehat{A}) = 2[-\sin(2\beta_p) E_0^- + \cos(2\beta_p) B(\gamma_p)]. \tag{B17}$$

Evaluating the gradient at $\gamma_p = 0$, $\beta_p = 0$ gives

$$|G_p(\widehat{A})| = |2B(0)| = |\langle \Psi^\star_{p-1} | i\widehat{A}\widehat{H}_+ | \Psi^\star_{p-1} \rangle|. \tag{B18}$$

We further note that the stationary points $\bar{\beta}_p$, $\bar{\gamma}_p$ where the gradient vanishes, fulfill the following conditions:

$$0 = \sin(2\bar{\beta}_p) C(\bar{\gamma}_p) \quad \text{and} \quad \sin(2\bar{\beta}_p) E_0^- = \cos(2\bar{\beta}_p) B(\bar{\gamma}_p). \tag{B19}$$

This implies stationary points of two types:

$$\text{Type 1: } \sin(2\bar{\beta}_{p,1}) = 0 \quad \text{and} \quad B(\bar{\gamma}_{p,1}) = 0, \tag{B20}$$

$$\text{Type 2: } C(\bar{\gamma}_{p,2}) = 0 \quad \text{and} \quad E_0^- \sin(2\bar{\beta}_{p,2}) = B(\bar{\gamma}_{p,2}) \cos(2\bar{\beta}_{p,2}). \tag{B21}$$

Further computing the second derivatives

$$\partial^2_{\gamma_p} \delta E_p(\beta_p, \gamma_p; \widehat{A}) = -4 \sin(2\beta_p) D(\gamma_p), \tag{B22}$$

$$\partial_{\beta_p} \partial_{\gamma_p} \delta E_p(\beta_p, \gamma_p; \widehat{A}) = 4 \cos(2\beta_p) C(\gamma_p), \tag{B23}$$

$$\partial_{\gamma_p} \partial_{\beta_p} \delta E_p(\beta_p, \gamma_p; \widehat{A}) = 4 \cos(2\beta_p) C(\gamma_p), \tag{B24}$$

$$\partial^2_{\beta_p} \delta E_p(\beta_p, \gamma_p; \widehat{A}) = -4[\cos(2\beta_p) E_0^- + \sin(2\beta_p) B(\gamma_p)] \tag{B25}$$

gives the Hessian

$$\mathrm{Hess}(\beta_p, \gamma_p) = \begin{pmatrix} -4\sin(2\beta_p)D(\gamma_p) & 4\cos(2\beta_p)C(\gamma_p) \\ 4\cos(2\beta_p)C(\gamma_p) & -4[\cos(2\beta_p)E_0^- + \sin(2\beta_p)B(\gamma_p)] \end{pmatrix}. \tag{B26}$$

The determinant of the Hessian is given as

$$\begin{aligned}\mathrm{Det}[\mathrm{Hess}(\beta_p, \gamma_p)] = 16[&\sin(2\beta_p)\cos(2\beta_p)D(\gamma_p)E_0^- \\ &+ \sin^2(2\beta_p)B(\gamma_p)D(\gamma_p) \\ &- \cos^2(2\beta_p)C(\gamma_p)].\end{aligned} \tag{B27}$$

Evaluating the determinant at the stationary points results in

$$\begin{aligned}&\mathrm{Det}[\mathrm{Hess}(\bar{\beta}_{p,1}, \bar{\gamma}_{p,1})] = -16\cos^2(2\bar{\beta}_{p,1})C^2(\bar{\gamma}_{p,1}) \text{ and} \\ &\mathrm{Det}[\mathrm{Hess}(\bar{\beta}_{p,2}, \bar{\gamma}_{p,2})] = 16B(\bar{\gamma}_{p,2})D(\bar{\gamma}_{p,2}).\end{aligned} \tag{B28}$$

The determinant of type 1 stationary points is always negative, indicating a positive and a negative eigenvalue of the Hessian matrix. This implies that type 1 stationary points are saddle point of the energy variation. On the other hand, the determinant of type 2 stationary points is positive if

$$B(\bar{\gamma}_{p,2})D(\bar{\gamma}_{p,2}) > 0. \tag{B29}$$

This implies that both eigenvalues of the Hessian matrix have the same sign, and thus indicates the existence of a minimum or a maximum of the energy variation.

Next, we analyze the trace of the Hessian, which is identical to the sum of its eigenvalues

$$\begin{aligned}\mathrm{Tr}[\mathrm{Hess}(\beta_p, \gamma_p)] = -4[&\sin(2\beta_p)D(\gamma_p) + \cos(2\beta_p)E_0^- \\ &+ \sin(2\beta_p)B(\gamma_p)].\end{aligned} \tag{B30}$$

Evaluating the trace at type-2 stationary points, Eq. (B21), and using its properties results in

$$\begin{aligned}\mathrm{Tr}[\mathrm{Hess}(\bar{\beta}_{p,2}, \bar{\gamma}_{p,2})] &= -4\{\sin(2\bar{\beta}_{p,2})[D(\bar{\gamma}_{p,2}) + B(\bar{\gamma}_{p,2})] \\ &\quad + \cos(2\bar{\beta}_{p,2})E_0^-\}, \qquad \text{(B31)} \\ &= -4\sin(2\bar{\beta}_{p,2})\Big[D(\bar{\gamma}_{p,2}) + B(\bar{\gamma}_{p,2}) \\ &\quad + \frac{(E_0^-)^2}{B(\bar{\gamma}_{p,2})}\Big]. \qquad \text{(B32)}\end{aligned}$$

To analyze the sign of the trace further, note that there are two distinct type-2 stationary points

$$\bar{\beta}_{p,2,-} \in [-\pi/2, 0] \quad \text{and} \quad \bar{\beta}_{p,2,+} \in [0, \pi/2]. \tag{B33}$$

Depending on the sign of $D(\bar{\gamma})_{p,2} + B(\bar{\gamma})_{p,2} + (E_0^-)^2/B(\bar{\gamma}_{p,2})$ one of these solutions will lead to a positive and the other one to a negative trace. Thus, as long as Eq. (B29) is fulfilled, the energy variation, will always have a unique minimum.

4. Assessment criteria for Dynamic-ADAPT-QAOA

Summarizing the aforementioned analysis, Eq. (B21) and Eq. (B29) imply that the energy variation, Eq. (16), has a unique minimum at $\bar{\gamma}_{p,2} = 0$ if

$$C(0) = 0 \quad \text{and} \quad B(0)D(0) > 0. \tag{B34}$$

Substituting $\widehat{A} = \widehat{A}_p$ into these equations, allows for identifying $B(0) \equiv B_p$, $C(0) \equiv C_p$, and $D(0) \equiv D_p$ [as defined in Eq. (17)] and results in condition (18). Dynamic-ADAPT-QAOA tests this condition to confirm whether the energy fluctuation has a unique minimum at $\bar{\gamma}_{p,2} = 0$, indicating an optimal value at $\gamma_p^\star \approx 0$.

APPENDIX C: COMPARISON OF DYNAMIC-ADAPT-QAOA VERSIONS

In this Appendix we provide numerical evidence supporting the two details presented at the end of Sec. II B related to the execution of Dynamic-ADAPT-QAOA. In particular, we analyze the convergence of three versions of Dynamic-ADAPT-QAOA:

(1) The full version which we consider in all other parts of this paper

(2) A version in which we remove the cost unitaries from all layers of the quantum circuit

(3) A version in which we do not reevaluate the energy gradients $G_p(\gamma_p; \widehat{A})$, and keep using the same mixer $\widehat{A}_p$ in the case where condition (18) is not met.

The data are presented in Fig. 7. We find that the full version of Dynamic-ADAPT-QAOA produces a much better average-case approximation ratio when compared to the other two tested versions. We conclude that the two aforementioned subtleties to our algorithm are justified, and contribute to its improved average-case solution accuracy.

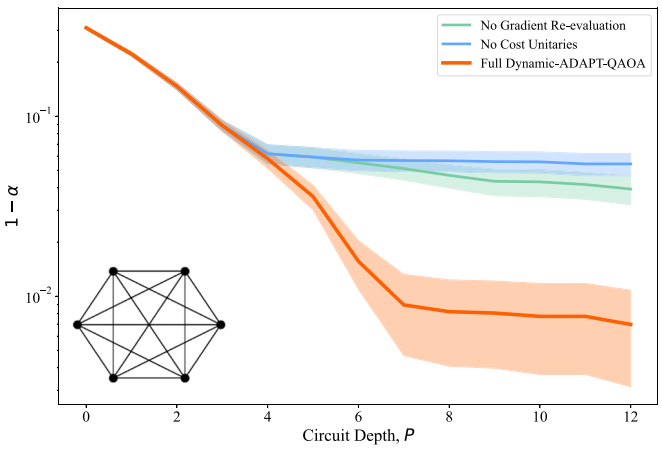

FIG. 7. Convergence curves for three versions of Dynamic-ADAPT-QAOA, applied to six-vertex complete graphs, without noise. $1-\alpha$ is plotted as a function of the depth P of the parameterized unitary U_P. The shaded regions depict 95% confidence intervals for the mean average-case approximation ratio, averaged over 100 randomized graphs.

APPENDIX D: SHOT-BASED IMPLEMENTATION AND SAMPLING COMPLEXITY OF DYNAMIC-ADAPT-QAOA

In the body of the paper, we present our algorithm based on expectation values. However, on a real quantum computer, expectation values must be evaluated by sampling. In this section we describe a sample-based implementation of Dynamic-ADAPT-QAOA and assess its sample complexity. To arrive at a sample-based implementation, we require several considerations. These include the following:

(1) Introducing appropriate dimensionless units in Appendix D 1.

(2) Explicitly expressing the gradients in Eq. (11) and additinal parameters in Eq. (17) in terms of a linear combination of Pauli strings, given by the weights W_{ij} of $\widehat{H}$ in Appendix D 2.

(3) Replacing expectation values with sample means and summarizing the sample-based implementation of Dynamic-ADAPT-QAOA, as described in Appendix D 3.

(4) Estimating the sample count of energy measurements, gradient measurements as well as measuring the linear response and curvature coefficient C_p and D_p under standard sampling strategies [36–39] in Appendix D 4.

(5) Estimating sample complexity for Max-Cut problems in Appendix D 5.

(6) Considering parallel gradient sampling in Appendix D 6.

(7) Fixing the relative precisions as described in Appendix D 7.

The aforementioned considerations will support the following conclusion, stated in the main text. In each iteration of (Dynamic-)ADAPT-QAOA the largest number of samples is required for estimating expectation values of the energy, when optimizing circuit parameters as well as measuring gradients of $O(N^2)$ mixer operators. Additionally, evaluating C_p and D_p to decide as to whether or not a cost Hamiltonian can be saved, represents a mild overhead. Specific order estimates for MaxCut problems are summarized in Appendix D 5.

1. Dimensionless units

Scaling. To motivate dimensionless units, consider a linear scaling of the weight factors in Eq. (1), given by

$$W_{ij} \mapsto \lambda W_{ij}. \tag{D1}$$

Under this transformation the energy operators scale linearly

$$\widehat{H} \mapsto \lambda \widehat{H}, \quad \widehat{H}_+ \mapsto \lambda \widehat{H}_+, \quad \widehat{H}_- \mapsto \lambda \widehat{H}_-. \tag{D2}$$

Additionally, expectation values related to energy and gradients change linearly

$$E_p \mapsto \lambda E_p, \quad G_p \mapsto \lambda G_p, \quad B_p \mapsto \lambda B_p, \tag{D3}$$

while the expectation of the linear response and curvature parameter scale quadratically or cubically, respectively, as

$$C_p \mapsto \lambda^2 C_p \quad \text{and} \quad D_p \mapsto \lambda^3 D_p. \tag{D4}$$

This is undesirable, as a change in the energy scale should not change the operation of Dynamic-ADAPT-QAOA. To overcome this issue we now formulate our algorithm in dimensionless units.

Energy scale. To introduce appropriate dimensionless units we define an energy scale $V_t > 0$. The energy scale should also scale linearly as

$$V_t \mapsto \lambda V_t, \tag{D5}$$

upon mapping the weights W_{ij} to λW_{ij}. A suitable choice for the energy scale V_t is the norm of the minimal eigenvalue of $\widehat{H}$. Alternatively, for Max-Cut the largest eigenvalue of the cost operator which represents the cut values, given by

$$\widehat{V} = \frac{1}{4}\sum_{i,j=1}^{N} W_{ij}(1 - \widehat{Z}_j\widehat{Z}_j), \tag{D6}$$

that is, the cut value $V_{\max}$ is also a suitable choice for the energy scale V_t. Since neither the minimal energy of H nor $V_{\max}$ are generally know, one could also use the cut value V_{GW} achieved by the GW algorithm to set the energy scale V_t. Note that scaling energies with V_t implies we will measure accuracies relative to that energy scale. This is precisely what we do, when comparing approximation ratios.

Dimensionless units. To define dimensionless units we introduce dimensionless weights for the cost Hamiltonian relative to the energy scale V_t as

$$\bar{W}_{ij} = \frac{W_{ij}}{V_t}. \tag{D7}$$

This leads to dimensionless energy operators given as

$$\widehat{H} = \frac{\widehat{H}}{V_t}, \quad \widehat{H}_+ = \frac{\widehat{H}_+}{V_t}, \quad \widehat{H}_- = \frac{\widehat{H}_-}{V_t}. \tag{D8}$$

Next, we scale the circuit parameters as

$$\vec{\bar{\beta}}_p = \vec{\beta}_p \quad \text{and} \quad \vec{\bar{\gamma}}_p = V_t\vec{\gamma}_p, \tag{D9}$$

respectively. This ensures that the unitary evolutions in Eq. (5) remain identical as

$$\widehat{\bar{U}}_p(\vec{\bar{\beta}}_p, \vec{\bar{\gamma}}_p) = \widehat{U}_p(\vec{\beta}_p, \vec{\gamma}_p). \tag{D10}$$

This defines the dimensionless energy expectation value

$$\bar{E}_p(\vec{\beta}_p, \vec{\gamma}_p) = \langle\Psi_0|\,\widehat{\bar{U}}^{\dagger}_p(\vec{\bar{\beta}}_p, \vec{\bar{\gamma}}_p)\frac{\widehat{H}}{V_t}\widehat{\bar{U}}_p(\vec{\bar{\beta}}_p, \vec{\bar{\gamma}}_p)\,|\Psi_0\rangle\,, \tag{D11}$$

and its optimal parameters

$$\vec{\bar{\beta}}^{\star}_p, \vec{\bar{\gamma}}^{\star}_p = \mathrm{argmin}_{\vec{\bar{\beta}}_p, \vec{\bar{\gamma}}_p}[\bar{E}_p(\vec{\bar{\beta}}_p, \vec{\bar{\gamma}}_p)]. \tag{D12}$$

The dimensionless optimal parameters define the optimal state after $p-1$ iterations

$$|\Psi^{\star}_{p-1}\rangle = \widehat{\bar{U}}_{p-1}(\vec{\bar{\beta}}^{\star}_{p-1}, \vec{\bar{\gamma}}^{\star}_{p-1})\,|\Psi_0\rangle\,, \tag{D13}$$

which defines dimensionless gradients of Pauli strings $\widehat{A} \in \mathcal{P}$ (which are already dimensionless) in the pth iteration as

$$\bar{G}_p(\gamma; \widehat{A}) = \langle\Psi^{\star}_{p-1}|\,e^{i\bar{\gamma}\widehat{\bar{H}}_+}\frac{i2\widehat{A}\widehat{H}_+}{V_t}e^{-i\bar{\gamma}\widehat{\bar{H}}_+}\,|\Psi^{\star}_{p-1}\rangle\,, \tag{D14}$$

and the additional dimensionless expectation values of the pth iteration for the Pauli string $\widehat{A}_p$ of maximal gradient as

$$\bar{B}_p = \langle\Psi^{\star}_{p-1}|\frac{i\widehat{A}_p\widehat{H}_+}{V_t}|\Psi^{\star}_{p-1}\rangle \equiv \bar{G}_p(0; \widehat{A}_p)/2, \tag{D15a}$$

$$\bar{C}_p = \langle\Psi^{\star}_{p-1}|\frac{\widehat{A}_p\widehat{H}^2_+}{V_t^2}|\Psi^{\star}_{p-1}\rangle, \tag{D15b}$$

$$\bar{D}_p = \langle\Psi^{\star}_{p-1}|\frac{i\widehat{A}_p\widehat{H}^3_+}{V_t^3}|\Psi^{\star}_{p-1}\rangle. \tag{D15c}$$

For notational convenience, we will drop the overbar from symbols indicating dimensionless units. Nonetheless, going forward all quantities are to be understood as considered in dimensionless units.

2. Pauli-string representation of measurement operators

In this sections we show that gradient, linear response, and curvature operators can be expressed as linear combinations of Pauli strings $\widehat{S}$. This is useful to assess their sampling complexity:

$$\widehat{G}(\widehat{A}) := i\widehat{A}\widehat{H}_+ = \sum_{i=1}^{N} g_i(\widehat{A})\widehat{S}_i(\widehat{A}), \tag{D16}$$

$$\widehat{C}(\widehat{A}) := \widehat{A}\widehat{H}_+^2 = \sum_{i,j=1}^{N} c_{i,j}(\widehat{A})\widehat{S}_{i,j}(\widehat{A}), \tag{D17}$$

$$\widehat{D}(\widehat{A}) := i\widehat{A}\widehat{H}_+^3 = \sum_{i,j,k=1}^{N} d_{i,j,k}(\widehat{A})\widehat{S}_{i,j,k}(\widehat{A}). \tag{D18}$$

Pauli mixers of length 1. First, we consider mixer operators of length one, i.e., $\widehat{A} \in \mathcal{P}_1$. In what follows let $n = n(\widehat{A})$ denote the qubit which the Pauli operator $\widehat{A} = \widehat{P}_n$ acts on. The corresponding anticommuting part of $\widehat{H}$ is then given as

$$\widehat{H}_{n,+} = \widehat{Z}_n \sum_{j=1}^{N} \frac{W_{n,j}}{2}\widehat{Z}_j. \tag{D19}$$

Recall that diagonal weights are zero, such that $\forall i = 1, \ldots, N$ we have $W_{ii} = 0$. Further defining the multiplicative part that depends only on Pauli Z operators as

$$\widehat{h}_n = \sum_{j=1}^{N} \frac{W_{n,j}}{2}\widehat{Z}_j, \tag{D20}$$

one shows

$$\widehat{H}_{n,+} = \widehat{Z}_n\widehat{h}_n, \quad \widehat{H}_{n,+}^2 = \widehat{h}_n^2, \quad \text{and} \quad \widehat{H}_{n,+}^3 = \widehat{Z}_n\widehat{h}_n^3. \tag{D21}$$

Next, defining the z-modified Pauli string

$$\widehat{\bar{P}}_n := i\widehat{P}_n\widehat{Z}_n, \tag{D22}$$

which maps $\widehat{P}_n = \widehat{Y}_n$ to $\widehat{\bar{P}}_n = -\widehat{X}_n$ and $\widehat{P}_n = \widehat{X}_n$ to $\widehat{\bar{P}}_n = \widehat{Y}_n$, one shows that

$$\widehat{G}(\widehat{A}) = \widehat{\bar{P}}_n\widehat{h}_n = \widehat{\bar{P}}_n \sum_i \frac{W_{n,i}}{2}\widehat{Z}_i, \tag{D23}$$

$$\widehat{C}(\widehat{A}) = \widehat{P}_n\widehat{h}_n^2 = \widehat{P}_n \sum_{i,j=1}^{N} \frac{W_{n,i}W_{n,j}}{2}\widehat{Z}_i\widehat{Z}_j, \tag{D24}$$

$$\widehat{D}(\widehat{A}) = \widehat{\bar{P}}_n\widehat{h}_n^3 = \widehat{\bar{P}}_n \sum_{i,j,k=1}^{N} \frac{W_{n,i}W_{n,j}W_{n,k}}{2}\widehat{Z}_i\widehat{Z}_j\widehat{Z}_k. \tag{D25}$$

From these expression we can read off the terms $g_i(\widehat{A})$, $c_{i,j}(\widehat{A})$, $d_{i,j,k}(\widehat{A})$, as well as the corresponding Pauli strings $\widehat{S}_i(\widehat{A})$, $\widehat{S}_{i,j}(\widehat{A})$, and $\widehat{S}_{i,j,k}(\widehat{A})$, respectively. It is further worth noting that (1) While $\widehat{P}_n$ and $\widehat{\bar{P}}_n$ act only on qubit n, the operators in $\widehat{h}_n$ act on all qubits, but n. Thus, $\widehat{P}_n$ and $\widehat{\bar{P}}_n$, commute with $\widehat{h}_n$. (2) Further, all Pauli strings in the above operator representation commute with each other and can therefore be measured simultaneously. (3) When evaluating the expectation values of the above operators for a trial state we use standard methods [36–39]. First, one rotates the operators $\widehat{P}_n$ (or $\widehat{\bar{P}}_n$) into the z-basis, by adding a single qubit rotation on the nth qubit to the ansatz circuit [38]. After that, one determines the eigenvalue of the Pauli string, by sampling in the z-basis.

Pauli mixers of length II with one $\widehat{Z}$. Next, we consider mixer operators of length two, given as $\widehat{A} = \widehat{P}_n\widehat{Z}_{n'}$, where $\widehat{P}_n$ is an $\widehat{X}$ or a $\widehat{Y}$ operator acting on qubit n and the second operator a Pauli $\widehat{Z}$ acting on a different qubit $n' \neq n$. The corresponding anticommuting part of $\widehat{H}$ is again given by $\widehat{H}_{n,+}$. Following the same steps as above one shows that

$$\widehat{G}(\widehat{A}) = \widehat{\bar{P}}_n\widehat{Z}_{n'}\widehat{h}_n = \widehat{\bar{P}}_n\widehat{Z}_{n'} \sum_i \frac{W_{n,i}}{2}\widehat{Z}_i, \tag{D26}$$

$$\widehat{C}(\widehat{A}) = \widehat{P}_n\widehat{Z}_{n'}\widehat{h}_n^2 = \widehat{P}_n\widehat{Z}_{n'} \sum_{i,j=1}^{N} \frac{W_{n,i}W_{n,j}}{2}\widehat{Z}_i\widehat{Z}_j, \tag{D27}$$

$$\widehat{D}(\widehat{A}) = \widehat{\bar{P}}_n\widehat{Z}_{n'}\widehat{h}_n^3 = \widehat{\bar{P}}_n\widehat{Z}_{n'} \sum_{i,j,k=1}^{N} \frac{W_{n,i}W_{n,j}W_{n,k}}{2}\widehat{Z}_i\widehat{Z}_j\widehat{Z}_k. \tag{D28}$$

From these expression we can again read off the terms $g_i(\widehat{A})$, $c_{i,j}(\widehat{A})$, $d_{i,j,k}(\widehat{A})$, as well as the corresponding Pauli strings $\widehat{S}_i(\widehat{A})$, $\widehat{S}_{i,j}(\widehat{A})$, and $\widehat{S}_{i,j,k}(\widehat{A})$, respectively. It is again worth noting that (1) While $\widehat{P}_n$ and $\widehat{\bar{P}}_n$ act only on qubit n, the operators in $\widehat{Z}_{n'}\widehat{h}_n$ act on all qubits, but n. Thus, $\widehat{P}_n$ and $\widehat{\bar{P}}_n$, commute with $\widehat{Z}_{n'}\widehat{h}_n$. (2) Further, all Pauli strings in the above operator representation commute with each other and can therefore be measured simultaneously. (3) When evaluating the expectation values of the above operators for a trial state we use standard methods [36–39]. First, one rotates the operators $\widehat{P}_n$ (or $\widehat{\bar{P}}_n$) into the z-basis, by adding a single qubit rotation on the nth qubit to the ansatz circuit [38]. After that, one determines the eigenvalue of the Pauli string, by sampling in the z-basis.

Pauli mixers of length II without $\widehat{Z}$. Finally, we consider mixer operators of length two, i.e., $\widehat{A} \in \mathcal{P}_2$, where none of the two operators is a $\widehat{Z}$ operator. In what follows let $n = n(\widehat{A})$, $n' = n'(\widehat{A})$ with $n \neq n'$ denote the two qubits which the Pauli operator $\widehat{A} = \widehat{P}_n\widehat{P}'_n$ acts on. Next, we define the coefficients

$$h_j^{n,n'} := \begin{cases} \frac{1}{2}W_{n,j} & \text{if } j \neq n, n' \\ 0 & \text{if } j = n, n' \end{cases}. \tag{D29}$$

This allows defining an operator of Pauli $\widehat{Z}$ operators, which acts on neither n nor n' as

$$\widehat{h}_{n,n'} = \sum_{j=1}^{N} h_j^{n,n'}\widehat{Z}_j. \tag{D30}$$

Using this operator, we can express the part of $\widehat{H}$ which anticommutes with $\widehat{A}$ as

$$\widehat{H}_{n,+} = (\widehat{Z}_n + \widehat{Z}_{n'}) \sum_{j=1}^{N} h_j^{n'n'}\widehat{Z}_j = (\widehat{Z}_n + \widehat{Z}_{n'})\widehat{h}_{n,n'}, \tag{D31}$$

and its second and third power as

$$\widehat{H}^2_{n,+} = 2(\widehat{1} + \widehat{Z}_n\widehat{Z}_{n'})\widehat{h}^2_{n,n'}, \tag{D32}$$

$$\widehat{H}^3_{n,+} = 4(\widehat{Z}_n + \widehat{Z}_{n'})\widehat{h}^3_{n,n'}. \tag{D33}$$

Again, one shows that for mixers $\widehat{A} = \widehat{P}_n\widehat{P}_{n'}$ with $n \neq n'$ and $\widehat{P}_n \in \{X_n, Y_n\}$ and $\widehat{P}_{n'} \in \{X_{n'}, Y_{n'}\}$ one has

$$\widehat{G}(\widehat{A}) = (\widehat{\bar{P}}_n\widehat{P}_{n'} + \widehat{P}_n\widehat{\bar{P}}_{n'})\widehat{h}_{n,n'}, \tag{D34}$$

$$\widehat{C}(\widehat{A}) = 2(\widehat{P}_n\widehat{P}_{n'} - \widehat{\bar{P}}_n\widehat{\bar{P}}_{n'})\widehat{h}^2_{n,n'}, \tag{D35}$$

$$\widehat{D}(\widehat{A}) = 4(\widehat{\bar{P}}_n\widehat{P}_{n'} + \widehat{P}_n\widehat{\bar{P}}_{n'})\widehat{h}^3_{n,n'}. \tag{D36}$$

From these expression we can read off the terms $g_i(\widehat{A})$, $c_{i,j}(\widehat{A})$, $d_{i,j,k}(\widehat{A})$, as well as the corresponding Pauli strings $\widehat{S}_i(\widehat{A})$, $\widehat{S}_{i,j}(\widehat{A})$, and $\widehat{S}_{i,j,k}(\widehat{A})$, respectively. It is further worth noting that (2) While the operators $\widehat{P}_n\widehat{P}_{n'}$, $\widehat{\bar{P}}_n\widehat{P}_{n'}$, $\widehat{P}_n\widehat{\bar{P}}_{n'}$, and $\widehat{\bar{P}}_n\widehat{\bar{P}}_{n'}$ act only on qubits n or n', the operators in $\widehat{h}_{n,n'}$ act on all qubits, but n, n'. Thus, $\widehat{P}_n\widehat{P}_{n'}$, $\widehat{\bar{P}}_n\widehat{P}_{n'}$, $\widehat{P}_n\widehat{\bar{P}}_{n'}$, and $\widehat{\bar{P}}_n\widehat{\bar{P}}_{n'}$ commute with $\widehat{h}_{n,n'}$. (2) Further, all Pauli strings in the above operator representation commute with each other and can therefore be measured simultaneously. (3) When evaluating the expectation values of the above operators for a trial state we use standard methods [36–39]. First, one rotates the operators $\widehat{P}_n\widehat{P}_{n'}$, $\widehat{\bar{P}}_n\widehat{P}_{n'}$, $\widehat{P}_n\widehat{\bar{P}}_{n'}$, and $\widehat{\bar{P}}_n\widehat{\bar{P}}_{n'}$ into the z-basis. This requires adding single and two-qubit qubit rotations on the nth and the n'th qubit after executing the circuit for ansatz preparation. After that, one determines the eigenvalue of the Pauli string, by sampling in the z-basis.

3. Replacing expectation value with sample mean

As a next step towards making Dynamic-ADAPT-QAOA more realistic for deployment on a quantum computer, we will replace (dimensionless) expectation values by their corresponding sample mean. In general, any expectation value (in dimensionless units) X, will be replaced by a sample-based estimator $\mathcal{X}$. For convenience, we will denote the sample-based estimator by the same symbol in a calgraphic font. In the algorithm, instead of making reference to a certain (dimensionless) expectation value given by

$$X = \langle\Psi_0^\star|\, U^\dagger \widehat{X} U\, |\Psi_0^\star\rangle\,, \tag{D37}$$

we will state "Sample X with precision ϵ to get $\mathcal{X}$." To achieve this one applies a specified circuit U to the state $|\Psi_0\rangle = |+\rangle\cdots|+\rangle$ to generate the state $U|\Psi_0\rangle$ several times. This is repeated until the probability that the estimator $\mathcal{X}$ deviates from $\bar{X}$ by less then ϵ is below a small given probability δ:

$$\mathbb{P}[|\bar{X} - \mathcal{X}| > \epsilon] < \delta. \tag{D38}$$

To achieve this, one will generally have to repeat the sampling process M times, where M will depend on the operator X, the precision ϵ, and the failure probability δ. Specific bounds for $M(X, \epsilon, \delta)$ to sample an expectation value X will be discussed further below.

In Dynamic-ADAPT-QAOA, we will sample the following expectation values: To estimate the dimensionless energy expectation value E_p we sample $\mathcal{E}_p$ with precision ϵ. To estimate dimensionless gradients $G_p(\bar{\gamma}; A)$ we sample $\mathcal{G}_p(\bar{\gamma}; A)$ with precision ϵ. And finally, to estimate $\bar{C}_p$ and $\bar{D}_p$ we sample $\mathcal{C}_p$ with precision ϵ' and $\mathcal{D}_p$ with precision ϵ'', respectively. A pseudocode summary of the sample-based algorithm is given in Algorithm 2.

4. Sampling requirements

a. Standard sampling strategies

To assess the sample requirements of Dynamic-ADAPT-QAOA, let us first recall standard strategies [36–39] for estimating expectation values of a (dimensionless) operator $\widehat{X}$ which consists of L Pauli strings $\widehat{S}_1, \ldots, \widehat{S}_L$ of arbitrary length as follows:

$$\widehat{X} = \sum_{l=1}^{L} h_l \widehat{S}_l. \tag{D39}$$

String-by-string sampling. A common strategy to sample an expectation value X of $\widehat{X}$ is to sample M_l eigenvalues $\{S_l^m : m = 1, \ldots, M_l\}$ for each Pauli string $\widehat{S}_l$ resulting in an estimator

$$\mathcal{X}(M_1, \ldots, M_l) = \sum_{l=1}^{L} \frac{h_l}{M_l} \sum_{m=1}^{M_l} S_l^m. \tag{D40}$$

By linearity of expectation, the expectation of the estimator is identical to the expectation of the operator

$$\mathbb{E}[\mathcal{X}(M_1, \ldots, M_l)] = \mathbb{E}[\widehat{X}]. \tag{D41}$$

Moreover, assuming all eigenvalues S_l^m for all Pauli strings have been sampled independently, it follows that

$$\mathrm{Var}[\mathcal{X}(M_1, \ldots, M_l)] = \sum_{l=1}^{L} \frac{h_l^2 \mathrm{Var}[\widehat{S}_l]}{M_l} \leqslant \sum_{l=1}^{L} \frac{h_l^2}{M_l}. \tag{D42}$$

Here the later inequality follows from $\mathrm{Var}[\widehat{S}_l] \leqslant 1$. Then minimizing the upper bound for a fixed number of samples $M = \sum_l M_l$, gives the optimal measurement strategy [36,37], where the lth Pauli string gets allocated the number of samples given by

$$M_l = \frac{|h_l|\sqrt{\mathrm{Var}[\widehat{S}_l]}}{\sum_l^{\prime L} |h_l'|\sqrt{\mathrm{Var}[\widehat{S}_l]}} M, \tag{D43}$$

leading to a variance, given by

$$\mathrm{Var}[\mathcal{X}(M_1, \ldots, M_l)] = \frac{1}{M}\left(\sum_{l=1}^{L} |h_l|\sqrt{\mathrm{Var}[\widehat{S}_l]}\right)^2. \tag{D44}$$

Using Chebyshev's inequality, one shows that

$$\mathbb{P}[|\mathcal{X}(M_1, \ldots, M_l) - \mathbb{E}[\widehat{X}]| > \epsilon] \leqslant \frac{\left(\sum_{l=1}^{L} |h_l|\sqrt{\mathrm{Var}[\widehat{S}_l]}\right)^2}{\epsilon^2 M}. \tag{D45}$$

ALGORITHM 2. Dynamic-ADAPT-QAOA implementation.

1: *//All quantities in dimensionless units*
2: Init pool $\mathcal{P}$
3: Init energy precision ε with desired accuracy of approximation ratio.
4: Set precisions ϵ, ϵ', ϵ'', and thresholds δ', δ''. *//For example, using Eq. (D107)*
5: Init $p \leftarrow 0$.
6: Init the initial state $|\Psi_0^\star\rangle = |+\rangle \cdots |+\rangle$.
7: Init circuit parameters $\vec{\beta}_0 \leftarrow ()$, $\vec{\gamma}_0 \leftarrow ()$ and its optimal values $\vec{\beta}_0^\star \leftarrow ()$, $\vec{\gamma}_0^\star \leftarrow ()$.
8: Init circuit $\widehat{U}_0(\vec{\beta}_p, \vec{\gamma}_p) \leftarrow \widehat{I}$.
9: Sample $E_p(\vec{\beta}_p^\star, \vec{\gamma}_p^\star) = \langle\Psi_0^\star| \widehat{U}_p^\dagger(\vec{\beta}_p^\star, \vec{\gamma}_p^\star)\widehat{H}\widehat{U}_p(\vec{\beta}_p^\star, \vec{\gamma}_p^\star)|\Psi_0^\star\rangle$ with precision ϵ to get $\mathcal{E}_p(\vec{\beta}_p^\star, \vec{\gamma}_p^\star)$.
10: **while** not converged **do**
11: $p \leftarrow p + 1$
12: *//Select mixer with maximal gradient*
13: $\forall \widehat{A} \in \mathcal{P}$ sample gradient $G_p(0;\widehat{A}) = \langle\Psi_0^\star| \widehat{U}_{p-1}^\dagger(\vec{\beta}_{p-1}^\star, \vec{\gamma}_{p-1}^\star) i[\widehat{A}, \widehat{H}]\widehat{U}_{p-1}(\vec{\beta}_{p-1}^\star, \vec{\gamma}_{p-1}^\star)|\Psi_0^\star\rangle$ with precision ϵ to get $\mathcal{G}_p(0;\widehat{A})$.
14: Select mixer with maximal gradient $\widehat{A}_p \leftarrow \text{argmax}_{\widehat{A}\in\mathcal{P}}[|\mathcal{G}_p(0;\widehat{A})|]$.
15: *//Grow circuit*
16: **if** A_p is a Pauli string **then**
17: *//Test, if the cost unitary is needed*
18: Sample $C_p = \langle\Psi_0^\star| \widehat{U}_{p-1}^\dagger(\vec{\beta}_{p-1}^\star, \vec{\gamma}_{p-1}^\star)[\widehat{A}_p\widehat{H}_+^2(\widehat{A}_p)]\widehat{U}_{p-1}(\vec{\beta}_{p-1}^\star, \vec{\gamma}_{p-1}^\star)|\Psi_0^\star\rangle$ with precision ϵ' to get $\mathcal{C}_p$.
19: Sample $D_p = \langle\Psi_0^\star| \widehat{U}_{p-1}^\dagger(\vec{\beta}_{p-1}^\star, \vec{\gamma}_{p-1}^\star)[i\widehat{A}_p\widehat{H}_+^3(\widehat{A}_p)]\widehat{U}_{p-1}(\vec{\beta}_{p-1}^\star, \vec{\gamma}_{p-1}^\star)|\Psi_0^\star\rangle$ with precision ϵ'' to get $\mathcal{D}_p$.
20: **if** $|\mathcal{G}_p(0;A_p)| > \varepsilon$ **and** $|\mathcal{C}_p| \leqslant \delta'$ **and** $|\mathcal{D}_p| > \delta''$ **and** $\mathcal{G}_p(0;A_p)\mathcal{D}_p > 0$ **then**
21: *//Add only mixer to parameters and circuits*
22: $\vec{\gamma}_p \leftarrow \vec{\gamma}_{p-1}$
23: $\vec{\beta}_p \leftarrow (\beta_p, \vec{\beta}_{p-1})$
24: $\widehat{U}_p(\vec{\beta}_p, \vec{\gamma}_p) \leftarrow e^{-i\beta_p\widehat{A}_p}\widehat{U}_{p-1}(\vec{\beta}_{p-1}, \vec{\gamma}_{p-1})$
25: **else** *//Do standard ADAPT-QAOA*
26: *//Preadd cost unitary to circuit*
27: $\widehat{U}_p(\tilde{\gamma}) \leftarrow e^{-i\tilde{\gamma}\widehat{H}}\widehat{U}_{p-1}(\vec{\beta}_{p-1}^*, \vec{\gamma}_{p-1}^*)$
28: *//Reselect mixer with maximal gradient*
29: $\forall \widehat{A} \in \mathcal{P}$ sample $G_p(\pm\tilde{\gamma};\widehat{A}) = \langle\Psi_0^\star| \widehat{U}_p^\dagger(\tilde{\gamma}) i[\widehat{A}, \widehat{H}]\widehat{U}_p(\tilde{\gamma})|\Psi_0^\star\rangle$ with precision ϵ to get $\mathcal{G}_p(\pm\tilde{\gamma};\widehat{A})$.
30: Select $\widehat{A}_p \leftarrow \text{argmax}_{\widehat{A}\in\mathcal{P}}[\max(|\mathcal{G}_p(\pm\tilde{\gamma}, \widehat{A})|)]$
31: *//Add mixer & cost to parameters & circuits*
32: $\vec{\gamma}_p \leftarrow (\gamma_p, \vec{\gamma}_{p-1})$
33: $\vec{\beta}_p \leftarrow (\beta_p, \vec{\beta}_{p-1})$
34: $\widehat{U}_p(\vec{\beta}_p, \vec{\gamma}_p) \leftarrow e^{-i\beta_p\widehat{A}_p}e^{-i\gamma_p\widehat{H}}\widehat{U}_{p-1}(\vec{\beta}_{p-1}, \vec{\gamma}_{p-1})$
35: **else if** A_p is a QAOA mixer **then** *//standard QAOA*
36: *//Add mixer & cost to parameters and circuits*
37: $\vec{\gamma}_p \leftarrow (\gamma_p, \vec{\gamma}_{p-1})$
38: $\vec{\beta}_p \leftarrow (\beta_p, \vec{\beta}_{p-1})$
39: $\widehat{U}_p(\vec{\beta}_p, \vec{\gamma}_p) \leftarrow e^{-i\beta_p\widehat{A}_p}e^{-i\gamma_p\widehat{H}}\widehat{U}_{p-1}(\vec{\beta}_{p-1}, \vec{\gamma}_{p-1})$
40: *//Optimize ansatz circuit and update bound*
41: Optimize params $\vec{\beta}_p^\star, \vec{\gamma}_p^\star \leftarrow \text{argmax}_{\vec{\beta}_p, \vec{\gamma}_p}[E_p(\vec{\beta}_p, \vec{\gamma}_p)]$
42: Sample to estimate $E_p(\vec{\beta}_p^\star, \vec{\gamma}_p^\star) = \langle\Psi_0^\star| \widehat{U}_p^\dagger(\vec{\beta}_p^\star, \vec{\gamma}_p^\star)\widehat{H}\widehat{U}_p(\vec{\beta}_p^\star, \vec{\gamma}_p^\star)|\Psi_0^\star\rangle$ with precision ϵ to get $\mathcal{E}_p(\vec{\beta}_p^\star, \vec{\gamma}_p^\star)$.
43: *//Check convergence*
44: **if** $p = P$ **or** $|\mathcal{E}_{p-1}(\vec{\beta}_{p-1}^\star, \vec{\gamma}_{p-1}^\star) - \mathcal{E}_p(\vec{\beta}_p^\star, \vec{\gamma}_p^\star)| < \varepsilon$ **then**
45: converged $\leftarrow$ True
46: **Return** $\mathcal{E}_p$, circuit U_p, params $\vec{\beta}_p^\star, \vec{\gamma}_p^\star$

Thus, to guarantee that the sample mean $\mathcal{X}(M_1, \ldots, M_l)$ deviates from the true expectation $\mathbb{E}[\widehat{X}]$ by less than epsilon with probability less than δ one requires $M(X, \epsilon, \delta)$ samples as given by

$$M(X, \epsilon, \delta) = \frac{\left(\sum_{l=1}^L |h_l|\sqrt{\text{Var}[\widehat{S}_l]}\right)^2}{\epsilon^2\delta} \leqslant \frac{\left(\sum_{l=1}^L |h_l|\right)^2}{\epsilon^2\delta}. \tag{D46}$$

Using the fact that each Pauli string $h_l\widehat{S}_l$ has a spectrum bounded in $[-|h_l|, |h_l|]$ one can further derive a tighter bound by using Hoeffding's inequality. This would result in a sample count given by

$$M(X, \epsilon, \delta) = \frac{\left(\sum_l |h_l|\right)^2}{\epsilon^2} \log\left(\frac{2}{\delta}\right) \tag{D47}$$

to guarantee that

$$\mathbb{P}[|\mathcal{X}(M_1, \ldots, M_l) - \mathbb{E}[\widehat{X}]| > \epsilon] \leqslant \delta. \tag{D48}$$

All-strings-at-once sampling. Finally, for all operators relevant to Dynamic-ADAPT-QAOA we showed above that all Pauli strings in the respective expansions commute. This

means we can measure all their expectation values simultaneously as discussed above. Thus, we can define an estimator of the expectation value X based on M eigenvalues $\{X^m : m = 1, \ldots, M\}$ of $\widehat{X}$ given as

$$\mathcal{X}(M) = \frac{1}{M}\sum_{m=1}^{M} X^m. \tag{D49}$$

For this estimator one shows that

$$\mathbb{E}[\mathcal{X}(M)] = \mathbb{E}[\widehat{X}], \tag{D50}$$

and assuming all eigenvalues X^m to have been sampled independently, it is easy to show that

$$\mathrm{Var}[\mathcal{X}(M)] = \frac{\mathrm{Var}[\widehat{X}]}{M}. \tag{D51}$$

Thus, using Chebychev's inequality, one shows that

$$\mathbb{P}[|\mathcal{X}(M) - \mathbb{E}[\widehat{X}]| > \epsilon] \leqslant \frac{\sqrt{\mathrm{Var}[\widehat{X}]}}{\epsilon^2 M}. \tag{D52}$$

To guarantee that the sample mean $\mathcal{X}(M)$ deviates from the true expectation $\mathbb{E}[\widehat{X}]$ by less than epsilon with probability at most δ one requires $M(X, \epsilon, \delta)$ samples as given by

$$M(X, \epsilon, \delta) = \frac{\mathrm{Var}[\widehat{X}]}{\epsilon^2 \delta}. \tag{D53}$$

In general, we expect sampling all commuting strings at once to be the more sample efficient strategy to estimate expectation values. Therefore, we recommend measuring all Pauli strings at once, whenever possible. On the other hand, using the fact that an operator X will have a spectrum bounded in $[X_l, X_r]$ with

$$|X_r - X_l| \leqslant 2\sum_{l=1}^{L} |h_l| \tag{D54}$$

we can use Popoviciu's inequality

$$\mathrm{Var}[\widehat{X}] \leqslant \frac{|X_r - X_l|^2}{4} \leqslant \left(\sum_{l=1}^{L} |h_l|\right)^2, \tag{D55}$$

to bound the sample count of sampling all strings at once as

$$M(X, \epsilon, \delta) = \frac{\mathrm{Var}[\widehat{X}]}{\epsilon^2\delta} \leqslant \frac{|X_r - X_l|^2}{4\epsilon^2\delta} \leqslant \frac{\left(\sum_{l=1}^{L} |h_l|\right)^2}{\epsilon^2\delta}. \tag{D56}$$

This implies that when deriving worst case sample estimate below, we will often not make a difference between all-strings-at-once and string-by-string-based sampling. It will, however, allow a tighter bound for energy expectation values.

b. Sample counts for Dynamic-ADAPT-QAOA

We will now give (worst-case) sample counts for Dynamic-ADAPT-QAOA. This concerns the expectation values of the energy, gradient, linear response, and curvature parameters. Note that all bounds will be given by coefficients which derive from weights W_{ij} in units with dimension. For that reason, the energy rescaling V_t will appear as an explicit factor.

Energy. First, we consider energy expectation values, which we sample with precision ϵ and failure probability less than δ. To begin with note that the spectral width of $\widehat{H}$ is identical to the spectral width of the cost operator $\widehat{V}$. Thus the spectral width of the dimensionless $\widehat{H}$ is given by

$$|E_r - E_l| \leqslant \frac{V_{\max}}{4V_t} \leqslant \sum_{i,l=1}^{N} \frac{|W_{il}|}{4V_t}, \tag{D57}$$

and the sample count is bounded as

$$M(E, \epsilon, \delta) \leqslant \frac{(V_{\max}/4V_t)^2}{\epsilon^2\delta} \leqslant \frac{\left(\sum_{i,l=1}^{N} |W_{il}|/4V_t\right)^2}{\epsilon^2\delta}, \tag{D58}$$

where the inner estimate holds for all-strings-at-once sampling and the outer bound for both Pauli-string-based sampling and all-strings-at-once sampling.

Additional parameters. Finally, when sampling a single gradient with precision ϵ, the linear response with precision ϵ', or curvature parameter with precision ϵ'' (either string-by-string or all-strings-at-once), we can upper bound their sample count as

$$M(G(\gamma; \widehat{A}), \epsilon, \delta) = \frac{\left(\sum_{i=1}^{N} |g_i(\widehat{A})|/V_t\right)^2}{\epsilon^2\delta}, \tag{D59}$$

$$M(C_p, \epsilon', \delta) = \frac{\left(\sum_{i,j=1}^{N} |c_{i,j}(\widehat{A}_p)|/V_t^2\right)^2}{\epsilon'^2\delta}, \tag{D60}$$

$$M(D_p, \epsilon'', \delta) = \frac{\left(\sum_{i,j,k=1}^{N} |d_{i,j,k}(\widehat{A}_p)|/V_t^3\right)^2}{\epsilon''^2\delta}. \tag{D61}$$

5. Sampling complexity for Max-Cut

In this section we evaluate the per iteration sampling complexity for the Max-Cut problem defined in the body of our paper. To this end recall that the weights W_{ij} with $i < j$ in dimensional units are sampled from a uniform distribution on the interval [0,1]. Weights along the diagonal $W_{ii} = 0$. Further, we now set the typical energy scale to be the Max-Cut value $V_{\max}$ of the cost operator as

$$V_t = V_{\max} = \Omega(N). \tag{D62}$$

Energy. With the above assumption one shows that the dimensionless energy expectation can be sampled all-strings-at-once with precision ϵ and failure probability less than δ using only

$$M(E, \epsilon, \delta) = \frac{1}{4\epsilon^2\delta}. \tag{D63}$$

Further, assuming the classical optimizer scales as $O(f(N))$ optimizing the dimensionless energy in each iteration with precision ϵ will require samples of the order of

$$M_E = O\left(\frac{f(N)}{\epsilon^2\delta}\right). \tag{D64}$$

In contrast, using that $\sum_{i,l=1}^{N} W_{ij} < N^2$ sampling the dimensionless energy expectation with precision ϵ with a string-by-string base strategy will require samples of the order of

$$M(E, \epsilon, \delta) = O\left(\frac{N^2}{\epsilon^2\delta}\right). \tag{D65}$$

This is worse and all-strings-at-once sampling should be applied.

Additional parameters. Finally, when assessing the sampling complexity of individual gradients, linear response, and curvature parameters per iteration, we can use that the coefficients of $\widehat{h}_n$ are bounded as

$$\sum_{j=1}^{N} \frac{|W_{n,j}|}{2} \leqslant \frac{N}{2} \tag{D66}$$

as well as that the coefficients of $\widehat{h}_{n,n'}$ are bounded as

$$\sum_{j=1, j\neq n'}^{N} \frac{|W_{n,j}|}{2} \leqslant \frac{N}{2} \tag{D67}$$

to show that

$$M(G(\gamma;\widehat{A}),\epsilon,\delta) = \frac{(N/V_{\max})^2}{\epsilon^2\delta} = O\left(\frac{1}{\epsilon^2\delta}\right), \tag{D68}$$

$$M(C_p,\epsilon',\delta) = \frac{\left(N^2/V_{\max}^2\right)^2}{\epsilon'^2\delta} = O\left(\frac{1}{\epsilon'^2\delta}\right), \tag{D69}$$

$$M(D_p,\epsilon'',\delta) = \frac{\left(N^3/V_{\max}^3\right)^2}{\epsilon''^2\delta} = O\left(\frac{1}{\epsilon''^2\delta}\right). \tag{D70}$$

Thus, by naively sampling the gradients for all $O(N^2)$ mixer operators independently, the total sampling cost per iteration for gradients is at most

$$M_G = O\left(\frac{N^2}{\epsilon^2\delta}\right). \tag{D71}$$

6. Parallel gradient sampling

Next, we consider a method for reducing the overhead of gradient sampling which has recently been presented for ADAPT-VQE [39]. Instead of sampling $|\mathcal{P}_1 \cup \mathcal{P}_2| = O(N^2)$ gradients $G(\gamma;\widehat{A})$ for each Pauli-string mixer $\widehat{A} \in \mathcal{P}_1 \cup \mathcal{P}_2$, it samples Pauli-string components of various gradients in parallel. Here we show that applying this method for Dynamic-ADAPT-QAOA will unfortunately not improve the worst-case estimate for the sample count.

At first, note that gradient operators result from a commutator of a mixer operator $\widehat{A} \in \mathcal{P}_1 \cup \mathcal{P}_2$ and the Hamiltonian $\widehat{H}$ as follows:

$$\widehat{G}(\widehat{A}) = [\widehat{A},\widehat{H}] = \sum_{i,j=1}^{N} \frac{W_{ij}}{2V_t}[\widehat{A},\widehat{Z}_i\widehat{Z}_j]. = \sum_{i,j\in\mathcal{E}_+(\widehat{A})} \frac{W_{ij}}{V_t}\widehat{A},\widehat{Z}_i\widehat{Z}_j. \tag{D72}$$

Here the last line follows from the following facts. If the commutator $[\widehat{A},\widehat{Z}_i\widehat{Z}_j]$ vanishes (i.e., if $\widehat{Z}_i\widehat{Z}_j$ is in the commuting set of $\widehat{A}$) then the term vanishes. Else, if the $[\widehat{A},\widehat{Z}_i\widehat{Z}_j]$ does not vanish, then $\widehat{Z}_i\widehat{Z}_j$ is in the anticommuting set of $\widehat{A}$ and we have $[\widehat{A},\widehat{Z}_i\widehat{Z}_j] = 2\widehat{A}\widehat{Z}_i\widehat{Z}_j$.

Next, consider two commuting Pauli-string mixer operators $\widehat{A}$ and $\widehat{A}'$. Following [39] we show

$$\begin{aligned}\forall\widehat{A},\widehat{A}' \in \mathcal{P}_1\cup\mathcal{P}_2 \text{ s.t. } [\widehat{A},\widehat{A}'] = 0\\ \Rightarrow [[\widehat{A},\widehat{Z}_i\widehat{Z}_j],[\widehat{A}',\widehat{Z}_i\widehat{Z}_j]] = 0.\end{aligned} \tag{D73}$$

To proof this note that, if neither $[\widehat{A},\widehat{Z}_i\widehat{Z}_j] = 0$ nor $[\widehat{A}',\widehat{Z}_i\widehat{Z}_j] = 0$, then both $\widehat{A}$ and $\widehat{A}'$ anticommute with $\widehat{Z}_i\widehat{Z}_j$ as

$$\begin{aligned}[[\widehat{A},\widehat{Z}_i\widehat{Z}_j],[\widehat{A}',\widehat{Z}_i\widehat{Z}_j]] &= 4[\widehat{A}\widehat{Z}_i\widehat{Z}_j,\widehat{A}'\widehat{Z}_i\widehat{Z}_j]\\ &= 4\widehat{A}\widehat{Z}_i\widehat{Z}_j\widehat{A}'\widehat{Z}_i\widehat{Z}_j - 4\widehat{A}'\widehat{Z}_i\widehat{Z}_j\widehat{A}\widehat{Z}_i\widehat{Z}_j\\ &= -4\widehat{A}\widehat{A}' + 4\widehat{A}'\widehat{A} = 4[\widehat{A}',\widehat{A}] = 0.\end{aligned} \tag{D74}$$

This shows that Pauli strings originating from nonvanishing commutators $[\widehat{A},\widehat{Z}_i\widehat{Z}_j]$ and $[\widehat{A}',\widehat{Z}_i\widehat{Z}_j]$ in distinct gradient operators $\widehat{G}(\widehat{A})$ and $\widehat{G}(\widehat{A}')$, induced by commuting Pauli mixers $[\widehat{A},\widehat{A}'] = 0$, can be measured simultaneously.

Next, we organize the Pauli strings mixers $(\mathcal{P}_1 \cup \mathcal{P}_2)$ into $3N+2$ disjoint subsets of commuting Pauli-string mixers. First, define the set of qubit indices with the nth index removed as

$$\mathcal{Q}(n) = \{1,\ldots,N\}\setminus\{n\}. \tag{D75}$$

Use this to collect all pairs XY, (YZ and ZX), respectively, into $n = 1,\ldots,N$ subsets, respectively, where X, (Y and Z), respectively, is pinned on the nth qubit, as

$$\mathcal{P}(n) := \{\widehat{X}_n\widehat{Y}_{n'} : n' \in \mathcal{Q}(n)\}, \tag{D76}$$

$$\mathcal{P}(N+n) := \{\widehat{Y}_n\widehat{Z}_{n'} : n' \in \mathcal{Q}(n)\}, \tag{D77}$$

$$\mathcal{P}(2N+n) := \{\widehat{Z}_n\widehat{X}_{n'} : n' \in \mathcal{Q}(n)\}, \tag{D78}$$

respectively. Finally, gather all remaining mixers composed of X- or XX-type into one commuting set and all mixers composed of Y- or YY-type into a second commuting set:

$$\begin{aligned}\mathcal{P}(3N+1) &:= \{\widehat{X}_n : n = 1,\ldots,N\}\\ &\bigcup\{\widehat{X}_n\widehat{X}_{n'} : n,n' = 1,\ldots,N \wedge n \neq n'\},\end{aligned} \tag{D79}$$

$$\begin{aligned}\mathcal{P}(3N+2) &:= \{\widehat{Y}_n : n = 1,\ldots,N\}\\ &\bigcup\{\widehat{Y}_n\widehat{Y}_{n'} : n,n' = 1,\ldots,N \wedge n \neq n'\}.\end{aligned} \tag{D80}$$

For each of the above $3N+2$ subpools $\mathcal{P}(\alpha)$ (with $\alpha = 1,\ldots,3N+2$) we know that for any two operators $\widehat{A},\widehat{A}' \in \mathcal{P}_\alpha$ we have $[\widehat{A},\widehat{A}'] = 0$. Hence, Pauli strings $\widehat{A}\widehat{Z}_i\widehat{Z}_j$, induced by nonvanishing commutators $[\widehat{A},\widehat{Z}_i\widehat{Z}_j]$ in gradient operators $\widehat{G}(\widehat{A})$ with $\widehat{A}$ in the same pool $\mathcal{P}_\alpha$, can be sampled simultaneously. (This requires applying single qubit rotations on up to N qubits, to simultaneously rotate all Pauli strings $\widehat{A}\widehat{Z}_i\widehat{Z}_j$ with $\widehat{A} \in \mathcal{P}_\alpha$ into the z-basis. This is actually always possible.)

Next, using the above structure we acquire samples as follows:

(1) For all $\widehat{A} \in \mathcal{P}(\alpha = 1)$ simultaneously sample M_{ij} eigenvalues $\{S^m(\widehat{A}\widehat{Z}_i\widehat{Z}_j)|m = 1,\ldots,M_{ij}\}$, corresponding to the Pauli strings $\widehat{A}\widehat{Z}_i\widehat{Z}_j$.

(2) Repeat, the previous step, for all $3N+2$ pools.

(3) Repeat, the previous two steps, for all combinations i,j with $i = 1,\ldots,N$ and $j = 1,\ldots,N$.

Completing the above sample acquisition, we construct gradient estimators as follows:

$$\mathcal{G}(\widehat{A};M_{ij}) = \sum_{i,j\in\mathcal{E}_+(\widehat{A})} \frac{W_{ij}}{V_t M_{ij}} \sum_{m=1}^{M_{ij}} S^m(\widehat{A}\widehat{Z}_i\widehat{Z}_j). \tag{D81}$$

Since all eigenvalues in the aforementioned estimator have been sampled independently, one proves

$$\mathbb{E}[\mathcal{G}(\widehat{A};M_{ij})] = \mathbb{E}[\widehat{G}(\widehat{A})] \tag{D82}$$

as well as

$$\mathrm{Var}[\mathcal{G}(\widehat{A};M_{ij})] = \sum_{i,j\in\mathcal{E}_+(\widehat{A})} \frac{W_{ij}^2 \mathrm{Var}[\widehat{A}\widehat{Z}_i\widehat{Z}_j]}{V_t^2 M_{ij}} \leqslant \sum_{i,j\in\mathcal{E}_+(\widehat{A})} \frac{W_{ij}^2}{V_t^2 M_{ij}}. \tag{D83}$$

Again, fixing the total number of samples in the above sample acquisition to $M_G^P = (3N+2)\sum_{i,j=1}^N M_{ij}$ and choosing

$$M_{ij} = \frac{M_G^P}{3N+2} \frac{|W_{ij}|/V_t}{\sum_{i,j}^N |W_{ij}|/V_t} \tag{D84}$$

leads to a variance given by

$$\mathrm{Var}[\mathcal{G}(\widehat{A};M_{ij})] = \frac{3N+2}{M_G^P} \left(\sum_{i,j\in\mathcal{E}_+(\widehat{A})} \frac{|W_{ij}|}{V_t} \sqrt{\mathrm{Var}[\widehat{A}\widehat{Z}_i\widehat{Z}_j]} \right) \times \left(\sum_{i,j=1}^N \frac{|W_{ij}|}{V_t} \right). \tag{D85}$$

Using Chebychev's inequality, one shows that

$$\mathbb{P}[|\mathcal{G}(\widehat{A};M_{ij}) - \mathbb{E}[\widehat{G}(\widehat{A})]| > \epsilon] \leqslant \frac{(3N+2)\Big(\sum_{i,j\in\mathcal{E}_+(\widehat{A})} |W_{ij}|/V_t \sqrt{\mathrm{Var}[\widehat{A}\widehat{Z}_i\widehat{Z}_j]}\Big)\big(\sum_{i,j=1}^N |W_{ij}|/V_t\big)}{\epsilon^2 M_G^P}. \tag{D86}$$

Thus, to guarantee that the sample means $\mathcal{G}(\widehat{A};M_{ij})$ of all $\widehat{A}\in\mathcal{P}_1\cup\mathcal{P}_2$ deviates from their true expectation $\mathbb{E}[\widehat{G}(\widehat{A})]$ by less than ϵ with probability less than δ one requires M_G^P samples as given by

$$M_G^P = \frac{(3N+2)\Big(\sum_{i,j\in\mathcal{E}_+(\widehat{A})} |W_{ij}|/V_t \sqrt{\mathrm{Var}[\widehat{A}\widehat{Z}_i\widehat{Z}_j]}\Big)\big(\sum_{i,j=1}^N |W_{ij}|/V_t\big)}{\epsilon^2\delta} \leqslant (3N+2)\frac{\big(\sum_{i,j=1}^N |W_{ij}|/V_t\big)^2}{\epsilon^2\delta}. \tag{D87}$$

As discussed in Ref. [39] this shows that sampling all gradients in parallel is indeed only $O(N)$ larger than a string-by-string-based sampling of $\widehat{H}$. [Recall that the latter would require $M(E,\epsilon,\delta) < (\sum_{i,j=1}^N |W_{ij}|/V_t)^2/(\epsilon^2\delta)$ samples.] Nevertheless, as we can measure the expectation of the energy of the Ising model with all strings at once, we were able to derive a tighter bound on the energy variance, resulting in fewer samples $M(E,\epsilon,\delta) = 1/(\epsilon^2\delta)$.

Finally, we evaluate the sample count of parallel gradient sampling for the Max-Cut problem. Specifically, we use

$$\sum_{i,j\in\mathcal{E}_+(\widehat{A})} |W_{ij}| \sqrt{\mathrm{Var}[\widehat{A}\widehat{Z}_i\widehat{Z}_j]} \leqslant N \quad \text{and} \quad \sum_{i,j=1}^N |W_{ij}| \leqslant N^2 \tag{D88}$$

to show that sampling all gradients of one iteration in parallel will require a number of shots on the order of

$$M_G^P \leqslant \frac{(3N+2)N^3}{\epsilon^2 V_{\max}^2 \delta} = O\left(\frac{N^2}{\epsilon^2\delta}\right). \tag{D89}$$

Notably, this worst case bound is identical to the worst case bound of Eq. (D71), which summarizes the shot count for sampling all gradients induced by Pauli-string mixers $\widehat{A}\in\mathcal{P}_1\cup\mathcal{P}_2$ individually. Thus, it remains to be seen in practice, whether sampling all gradients in parallel is as beneficial for QAOA as it was found to be for VQE in Ref. [39].

7. Suggesting relative precisions

We now propose relative sampling precisions for energies, gradients, linear response, and curvature operators. The result is given by Eq. (D107).

a. Further analysis of the energy fluctuation

To begin with, we return to the energy fluctuation defined in Eq. with parameters defined in Eq. :

$$\delta E_p(\beta_p,\gamma_p;\widehat{A}) = E_0^- + E_0^+\cos(2\beta_p) + B(\gamma_p)\sin(2\beta_p). \tag{D90}$$

Note that this function depends on γ_p only via the parameter $B(\gamma_p)$. Next, rewrite this function as

$$\delta E_p(\beta_p,\gamma_p;\widehat{A}) = E_0^- - \sqrt{|E_0^+|^2 + |B(\gamma_p)|^2} \times \cos(2\beta_p - 2\tilde{\beta}_p(B(\gamma_p))), \tag{D91}$$

where

$$\tilde{\beta}_p(B(\gamma_p)) := \tfrac{1}{2}\mathrm{Atan2}(-B(\gamma_p), E_0^+). \tag{D92}$$

This shows that for each fixed value of γ_p the energy fluctuation $\delta E_p(\beta_p,\gamma_p;\widehat{A})$ as a function of $\beta_p\in[-\pi/2,\pi/2]$ has a single unique minimum at $\tilde{\beta}_p(B(\gamma_p))$. In other words, $\tilde{\beta}_p(B(\gamma_p))$ has a valley, running "parallel" to the γ_p direction with its deepest point located at $\tilde{\beta}_p(B(\gamma_p))$. Around this minimum, the curvature in β-direction is always positive. Hence, to trace minima in $\delta E_p(\beta_p,\gamma_p;\widehat{A})$ it is sufficient to trace along $\tilde{\beta}_p(B(\gamma_p))$ in the gamma direction. This is achieved by setting $\beta = \tilde{\beta}_p(B(\gamma_p))$ and analyzing the reduced energy fluctuation

along the valley parameterized by $\beta = \tilde{\beta}_p(B(\gamma_p))$ as

$$\delta E_p(\gamma_p) = E_0^- - \sqrt{|E_0^+|^2 + |B(\gamma_p)|^2}. \tag{D93}$$

This simplification allows us to determine whether the energy fluctuation has a minimum, maximum or a saddle at $\gamma_p = 0$, by considering the above simplified function.

To determine whether the energy fluctuation has a minimum near $\gamma_p = 0$ we further analyze the reduced energy fluctation at $\gamma_p = 0$. To this end we subtract the energy at $\gamma_p = 0$ and define

$$\begin{aligned}\Delta(\gamma_p) = \delta(\gamma_p) - \delta(0) = -\sqrt{|E_0^+|^2 + |B(\gamma_p)|^2} \\ + \sqrt{|E_0^+|^2 + |B(0)|^2} \leqslant |B(\gamma_p) - B(0)|.\end{aligned} \tag{D94}$$

(The upper bound results from the triangle inequality.) Next, Taylor expanding $\Delta(\gamma_p)$ to quadratic order in γ_p gives

$$\begin{aligned}\Delta(\gamma_p) = &-\frac{B(0)B'(0)}{\sqrt{|E_0^+|^2 + |B(0)|^2}}\gamma_p \\ &- \frac{1}{2}\left[\frac{B(0)B''(0)}{\sqrt{|E_0^+|^2 + |B(0)|^2}} + \frac{|E_0^+|^2|B'(0)|^2}{\sqrt{|E_0^+|^2 + |B(0)|^2}^3}\right]\gamma_p^2 \\ &+ O(\gamma_p^3).\end{aligned} \tag{D95}$$

At this point recall Eqs. (B14) and (17), which imply that

$$B(0) = G_p(\gamma_p; \widehat{A}_p)/2, \tag{D96}$$

$$B'(0) = 2C_p, \tag{D97}$$

$$B''(0) = -4D_p. \tag{D98}$$

Since the optimization of gradients ensures that Dynamic-ADAPT-QAOA selects $2|B(0)| = |G_p(\gamma_p; \widehat{A}_p)| > 0$, the above Taylor expansion implies that the energy fluctuation $\Delta(\gamma_p)$ will have a minimum at $\gamma_p = 0$, if

$$\begin{aligned}B'(0) = 2C_p = 0 \quad &\text{and} \\ B(0)B''(0) = -4B_pD_p \\ = -2G_p(\gamma_p; \widehat{A})D_p < 0.\end{aligned} \tag{D99}$$

Otherwise, either $B'(0)$ is nonzero, implying that the energy fluctuation has a significant slope along the γ_p-direction and a better minimum is located at a different location of $\gamma_p \neq 0$. Or $B'(0) = 0$ but $B(0)B''(0) > 0$, implying that the energy fluctuation has a saddle at $\gamma_p = 0$. Again, this implies a lower energy fluctuation should be located at a different location of $\gamma_p \neq 0$. Both cases, are covered by the algorithm stepping away to $\gamma = \pm\tilde{\gamma}$.

b. Setting accuracy parameters

Finally, we turn our attention to assessing the conditions in Eq. (D99) in the presence of finite sampling errors. Recall, in each iteration we assess

(1) $\mathcal{G}_p(\gamma_p; \widehat{A}_p)$ estimates $G_p(\gamma_p; \widehat{A}_p) = 2B(0) = 2B_p$ s.t. $\mathbb{P}[|G_p(\gamma_p; \widehat{A}_p) - \mathcal{G}_p(\gamma_p; \widehat{A}_p)| > \epsilon] < \delta$.

(2) $\mathcal{C}_p$ estimates $C_p = C(0) = 2B'(0)$ s.t. $\mathbb{P}[|C_p - \mathcal{C}_p| > \epsilon'] < \delta$.

(3) $\mathcal{D}_p$ estimates $D_p = D(0) = -B''(0)/4$ s.t. $\mathbb{P}[|D_p - \mathcal{D}_p| > \epsilon''] < \delta$.

Further, recall that dimensionless units are scaled with the typical energy scale V_t of the system. Thus, dimensionless energies behave similar to approximation ratio. Thus we recommend to fix ε on the order of an acceptable error in an approximation ratio. Next, we recommend sampling energies and gradients with an energy precision ϵ which is better than the desired energy scale ε, for example using $\epsilon \approx \frac{\varepsilon}{10}$. Next, we discuss how to ensure the conditions in Eq. (D99) with high probability.

Convexity condition. To ensure the convexity condition $B(0)B''(0) < 0$ in Eq. (D99) with high probability, we will implement the following tests:

$$|\mathcal{G}_p(\gamma_p; \widehat{A})| \geqslant \varepsilon \quad \text{and} \quad |\mathcal{D}_p| \geqslant \delta'' \quad \text{and} \quad \mathcal{G}_p(\gamma_p; \widehat{A})\mathcal{D}_p \geqslant 0. \tag{D100}$$

The first test ensures that $B_p > (\varepsilon - \epsilon)/2$ with probability larger $(1 - \delta)$. The second test ensures that $D_p > (\delta'' - \epsilon'')$ with probability larger $(1 - \delta)$. Further keeping $\varepsilon > \epsilon$ and $\delta'' > \epsilon''$ ensures that

$$\mathbb{P}[\text{sign}(\mathcal{G}_p(\gamma_p; \widehat{A})\mathcal{D}_p) = \text{sign}(G_p(\gamma_p; \widehat{A})D_p)] > (1 - 2\delta). \tag{D101}$$

In other words, the product of estimators $\mathcal{G}_p(\gamma_p; \widehat{A})\mathcal{D}_p$ has the same sign as the expectation value product $G_p(\gamma_p; \widehat{A})D_p$ with probability larger than $1 - 2\delta$. And thus, the third test $\mathcal{G}_p(\gamma_p; \widehat{A})\mathcal{D}_p \geqslant 0$ ensures that the convexity condition $G_p(\gamma_p; \widehat{A})D_p \geqslant 0$ is assessed correctly with high probability larger than $1 - 2\delta$.

Vanishing linear response. Finally, to approximately ensure vanishing linear response, i.e., $B'(0) = 0$ in Eq. (D99) with high probability, we will implement the following tests:

$$|\mathcal{C}_p| \leqslant \delta' \tag{D102}$$

with relative accuracies chosen as $\delta' = 2\epsilon'$, $\epsilon' = 2^{(4/3)}\epsilon^{2/3}/3$, $\delta'' = 2\epsilon''$, and $\epsilon'' = 2^{-(4/3)}\epsilon^{1/3}$ (as also summarized in Eq. (D107)). The reasoning behind these choices is as follows. $|\mathcal{C}_p| \leqslant \delta'$ guarantees that $|C_p| \leqslant \delta' + \epsilon'$ with high probability larger than $1 - \delta$. This may offset the true minimum of $\Delta(\gamma_p)$ to a different location $\gamma_{p,\min} \ll 1$ in the vicinity of $\gamma_p = 0$. In the worst case, this leads to an energy error upper bounded by

$$|\Delta(\gamma_p)| \leqslant |B(\gamma_p) - B(0)| = \left|B'(0)\gamma_p - \frac{B''(0)}{2}\gamma_p^2 + O(\gamma_p^3)\right|. \tag{D103}$$

This upper bound takes the maximal value at

$$\gamma_{p,\min} = -\frac{B'(0)}{B''(0)}. \tag{D104}$$

This would lead to the maximal possible energy error, given as

$$|\Delta(\gamma_p)| \leqslant \left|\frac{B'(0)^2}{2B''(0)}\right| + O(|\gamma_{p,\min}|^3)$$
$$= \left|\frac{B'(0)^2}{2B''(0)}\right| + O\left(\left|\frac{B'(0)}{B''(0)}\right|^3\right). \quad \text{(D105)}$$

With our sampling conditions we know that $B'(0)$ must have been smaller than $|B'(0)| < 3\epsilon'/2$ with high probability and $|B''(0)| > 4\epsilon''$ with large probability. Further using the proposed relations between accuracies, we find that the energy remains bounded to order ϵ with high accuracy as given by

$$|\Delta(\gamma_p)| \leqslant \frac{9\epsilon'^2}{2^5\epsilon''} + O\left(\frac{27\epsilon'^3}{2^9\epsilon''^3}\right) = \frac{\epsilon}{2} + O\left(\frac{\epsilon}{2}\right) = O(\epsilon). \quad \text{(D106)}$$

Thus, our sampling conditions are designed such that the maximal energy error induced by making a wrong decision is at most $O(\epsilon)$.

Summary of accuracies. This summarizes our recommended choice of accuracies as follows. Set ε as the desired accuracy of the approximation ratio and choose other ratios as follows:

$$\epsilon = \varepsilon/10, \quad \text{(D107a)}$$
$$\delta' = 2\epsilon', \quad \text{(D107b)}$$
$$\epsilon' = 2^{(4/3)}\epsilon^{2/3}/3, \quad \text{(D107c)}$$
$$\delta'' = 2\epsilon'', \quad \text{(D107d)}$$
$$\epsilon'' = 2^{-(4/3)}\epsilon^{1/3}. \quad \text{(D107e)}$$

APPENDIX E: NOISELESS AND NOISY CIRCUIT GROWTH

In this Appendix we present numerical evidence supporting our approach to evaluating the average-case solution accuracy of Dynamic- and standard ADAPT-QAOA. In Appendix A we mention two methods for analyzing the evolution of the average-case approximation ratio achieved by the quantum algorithms. In one, the effects of noise are implemented during the circuit growth. This is representative of how the algorithm would be run in an experiment using real hardware. However, simulating this approach using density matrices requires large computing times, making it impractical. Therefore, we make use of the alternative method, used previously in the context of variational quantum eigensolvers [34]. This alternative approach relies on the theoretical result that noise in variational quantum algorithms primarily flattens the parameter landscape, without altering its structure [65].

In particular, we grow the quantum circuits in the absence of noise, recording the optimal components and parameters found during each iteration. To investigate the effect of noise afterward, we simulate the preoptimized circuits while including the respective noise channel where necessary. This greatly reduces the computational time required for running the quantum algorithms, making it the preferable method for numerical investigations.

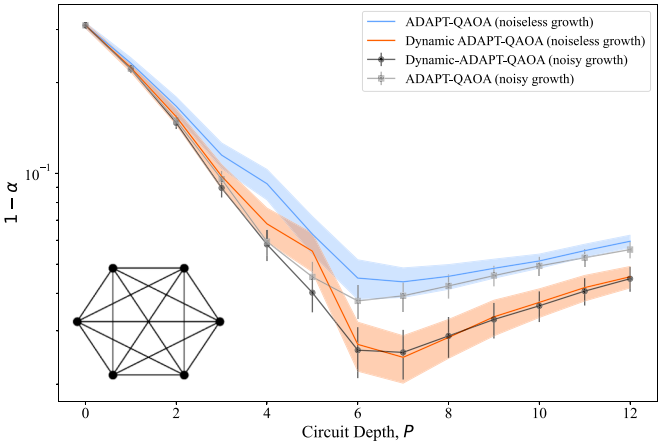

FIG. 8. Convergence curves for Dynamic- and standard ADAPT-QAOA, applied to six-vertex complete graphs, with noise. $1-\alpha$ is plotted as a function of the depth P of the parameterized unitary U_P. The data for both noiseless and noisy growth of the quantum circuit are presented. The error bars and shaded regions depict the standard error in the mean average-case approximation ratio, averaged over 100 randomized graphs.

It is important to check whether growing the quantum circuits in the noiseless regime significantly alters the results of our analysis. To address this, we compare the two methods described above in the context of 6-vertex graphs. The resulting data are shown in Fig. 8 for a gate-error probability of $p_{\text{gate}} = 0.122\%$. We find that both methods produce, on average, the same behavior for either algorithm. This justifies our use of the less computationally time-consuming approach for circuit growth simulation.

APPENDIX F: ERROR MITIGATION IN DYNAMIC-ADAPT-QAOA

In this Appendix we show that error mitigation techniques can be used to improve the mean average-case approximation ratio which can be achieved by Dynamic-ADAPT-QAOA for a given gate-error probability p_{gate}. In particular, we apply Richardson extrapolation as described in Ref. [61]. For a specific value of p_{gate} we consider the average-case approximation ratio outputted at the end of the algorithm when applied to six-vertex complete graphs. This is averaged over 100 randomized graph instances to produce a mean average-case approximation ratio.

Once we have collected the data for the average-case approximation ratio for each graph, we consider what these values are at a different gate-error probability, $c \times p_{\text{gate}}$. From the data at $c = 1$ and $c = 3$, we can extrapolate (according to the formula in Ref. [61]), and produce a hopefully improved average-case approximation ratio. This procedure is repeated for all randomized graphs separately, after which we average the results to produce a mean error-mitigated average-case approximation ratio.

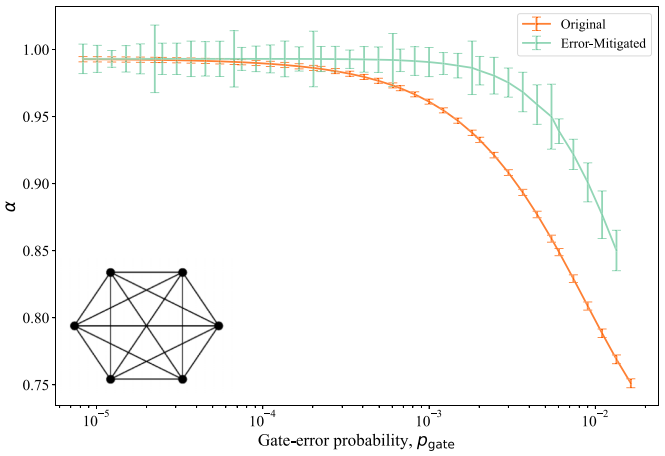

FIG. 9. Final layer average-case approximation ratio α as a function of the gate-error probability p_{gate}, both with and without error mitigation. The data were acquired in noisy simulations of six-vertex complete graphs. The error bars show the standard error in the mean average-case approximation ratio.

We produce curves of the mean average-case approximation ratio against the gate-error probability both with and without error mitigation. These are depicted in Fig. 9. The data shows that applying error mitigation, even if it is just to second-order, we can improve, on average, the outputted average-case approximation ratio by Dynamic-ADAPT-QAOA. Additionally, extending this technique to other versions of QAOA should be analogous. However, there is one important subtlety regarding the use of error mitigation to "improve" an algorithm's solution accuracy. In a practical setting, say in the context of Max-Cut, the important output is the final distribution of bit strings corresponding to partitions of the graph's vertices. The average-case approximation ratio is simply used as measure of the algorithm's average-case solution quality. Hence, although error mitigation, as applied above, can lead to improvements in the average-case approximation ratio as a performance measure of solution quality, it does not actually alter the average-case solution quality of the output bit strings which define the partition of vertices. In other words, it is not clear what practical advantage such a technique could offer.

[1] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness* (W. H. Freeman & Co., New York, 1990).

[2] G. Kochenberger, J.-K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang, and Y. Wang, The unconstrained binary quadratic programming problem: A survey, J. Combin. Optimiz. **28**, 58 (2014).

[3] A. Lucas, Ising formulations of many NP problems, Front. Phys. **2**, 1 (2014).

[4] F. Barahona, On the computational complexity of Ising spin glass models, J. Phys. A: Math. Gen. **15**, 3241 (1982).

[5] N. Mohseni, P. L. McMahon, and T. Byrnes, Ising machines as hardware solvers of combinatorial optimization problems, Nat. Rev. Phys. **4**, 363 (2022).

[6] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, Science **220**, 671 (1983).

[7] S. V. Isakov, I. N. Zintchenko, T. F. Rønnow, and M. Troyer, Optimised simulated annealing for Ising spin glasses, Comput. Phys. Commun. **192**, 265 (2015).

[8] H. Goto, K. Tatsumura, and A. R. Dixon, Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems, Sci. Adv. **5**, eaav2372 (2019).

[9] T. Okuyama, T. Sonobe, K.-I. Kawarabayashi, and M. Yamaoka, Binary optimization by momentum annealing, Phys. Rev. E **100**, 012111 (2019).

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

[11] T. Albash and D. A. Lidar, Adiabatic quantum computation, Rev. Mod. Phys. **90**, 015002 (2018).

[12] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk *et al.*, Quantum annealing with manufactured spins, Nature (London) **473**, 194 (2011).

[13] 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).

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

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

[16] Y.-H. Oh, H. Mohammadbagherpoor, P. Dreher, A. Singh, X. Yu, and A. J. Rindos, Solving multi-coloring combinatorial optimization problems using hybrid quantum algorithms, arXiv:1911.00595 (2019).

[17] M. Born and V. Fock, Beweis des Adiabatensatzes, Z. Phys. **51**, 165 (1928).

[18] D. Stilck França and R. García-Patrón, Limitations of optimization algorithms on noisy quantum devices, Nat. Phys. **17**, 1221 (2021).

[19] G. De Palma, M. Marvian, C. Rouzé, and D. S. França, Limitations of variational quantum algorithms: A quantum optimal transport approach, PRX Quantum **4**, 010309 (2023).

[20] L. Zhu, H. L. Tang, G. S. Barron, F. A. Calderon-Vargas, N. J. Mayhall, E. Barnes, and S. E. Economou, Adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer, Phys. Rev. Res. **4**, 033029 (2022).

[21] H. L. Tang, V. O. Shkolnikov, G. S. Barron, H. R. Grimsley, N. J. Mayhall, E. Barnes, and S. E. Economou, Qubit-ADAPT-VQE: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor, PRX Quantum **2**, 020310 (2021).

[22] C. K. Long, K. Dalton, C. H. W. Barnes, D. R. M. Arvidsson-Shukur, and N. Mertig, Layering and subpool exploration for

adaptive variational quantum eigensolvers: Reducing circuit depth, runtime, and susceptibility to noise, arXiv:2308.11708 (2023).

[23] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. **9**, 4812 (2018).

[24] R. Wiersema, C. Zhou, Y. de Sereville, J. F. Carrasquilla, Y. B. Kim, and H. Yuen, Exploring entanglement and optimization within the Hamiltonian variational ansatz, PRX Quantum **1**, 020319 (2020).

[25] B. T. Kiani, S. Lloyd, and R. Maity, Learning unitaries by gradient descent, arXiv:2001.11897 (2020).

[26] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, Nat. Commun. **12**, 1791 (2021).

[27] C. Ortiz Marrero, M. Kieferová, and N. Wiebe, Entanglement-induced barren plateaus, PRX Quantum **2**, 040316 (2021).

[28] X. You and X. Wu, Exponentially many local minima in quantum neural networks, in *International Conference on Machine Learning*, edited by M. Meila and T. Zhang (PMLR, 2021), pp. 12144–12155.

[29] E. R. Anschuetz, Critical points in quantum generative models, arXiv:2109.06957 (2022).

[30] E. R. Anschuetz and B. T. Kiani, Quantum variational algorithms are swamped with traps, Nat. Commun. **13**, 7760 (2022).

[31] A. Deshpande, P. Niroula, O. Shtanko, A. V. Gorshkov, B. Fefferman, and M. J. Gullans, Tight bounds on the convergence of noisy random circuits to the uniform distribution, PRX Quantum **3**, 040329 (2022).

[32] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, An adaptive variational algorithm for exact molecular simulations on a quantum computer, Nat. Commun. **10**, 3007 (2019).

[33] Y. S. Yordanov, V. Armaos, C. H. W. Barnes, and D. R. M. Arvidsson-Shukur, Qubit-excitation-based adaptive variational quantum eigensolver, Commun. Phys. **4**, 228 (2021).

[34] K. Dalton, C. K. Long, Y. S. Yordanov, C. G. Smith, C. H. W. Barnes, N. Mertig, and D. R. M. Arvidsson-Shukur, Quantifying the effect of gate errors on variational quantum eigensolvers for quantum chemistry, npj Quantum Inf. **10**, 1 (2024).

[35] H. R. Grimsley, G. S. Barron, E. Barnes, S. E. Economou, and N. J. Mayhall, Adaptive, problem-tailored variational quantum eigensolver mitigates rough parameter landscapes and barren plateaus, npj Quantum Inf. **9**, 19 (2023).

[36] D. Wecker, M. B. Hastings, and M. Troyer, Progress towards practical quantum variational algorithms, Phys. Rev. A **92**, 042303 (2015).

[37] N. C. Rubin, R. Babbush, and J. McClean, Application of fermionic marginal constraints to hybrid quantum algorithms, New J. Phys. **20**, 053020 (2018).

[38] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz, Quantum Sci. Technol. **4**, 014008 (2018).

[39] P. G. Anastasiou, N. J. Mayhall, E. Barnes, and S. E. Economou, How to really measure operator gradients in ADAPT-VQE, arXiv:2306.03227 (2023).

[40] C. W. Commander, Maximum cut problem, MAX-CUTMaximum Cut Problem, MAX-CUT, *Encyclopedia of Optimization*, edited by C. A. Floudas and P. M. Pardalos (Springer US, Boston, MA, 2009), pp. 1991–1999.

[41] S. T. McCormick, M. R. Rao, and G. Rinaldi, Easy and difficult objective functions for max cut, Math. Program. **94**, 459 (2003).

[42] D. P. Williamson and D. B. Shmoys, *The Design of Approximation Algorithms* (Cambridge University Press, Cambridge, 2011).

[43] C. J. Wood, Special session: Noise characterization and error mitigation in near-term quantum computers, in *2020 IEEE 38th International Conference on Computer Design (ICCD)* (2020), pp. 13–16.

[44] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, Superconducting qubits: Current state of play, Annu. Rev. Condens. Matter Phys. **11**, 369 (2020).

[45] L. Ding, M. Hays, Y. Sung, B. Kannan, J. An, A. Di Paolo, A. H. Karamlou, T. M. Hazard, K. Azar, D. K. Kim, B. M. Niedzielski, A. Melville, M. E. Schwartz, J. L. Yoder, T. P. Orlando, S. Gustavsson, J. A. Grover, K. Serniak, and W. D. Oliver, High-fidelity, frequency-flexible two-qubit fluxonium gates with a transmon coupler, Phys. Rev. X **13**, 031035 (2023).

[46] R. Shaydulin and S. M. Wild, Exploiting symmetry reduces the cost of training QAOA, IEEE Trans. Quantum Eng. **2**, 1 (2021).

[47] A. Bärtschi and S. Eidenbenz, Grover mixers for QAOA: Shifting complexity from mixer design to state preparation, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (2020), pp. 72–82.

[48] 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).

[49] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn *et al.*, Quantum optimization using variational algorithms on near-term quantum devices, Quantum Sci. Technol. **3**, 030503 (2018).

[50] 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).

[51] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo *et al.*, Quantum approximate optimization of nonplanar graph problems on a planar superconducting processor, Nat. Phys. **17**, 332 (2021).

[52] G. Pagano, A. Bapat, P. Becker, K. S. Collins, A. De, P. W. Hess, H. B. Kaplan, A. Kyprianidis, W. L. Tan, C. Baldwin *et al.*, Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator, Proc. Natl. Acad. Sci. USA **117**, 25396 (2020).

[53] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, Quantum approximate optimization algorithm for MaxCut: A fermionic view, Phys. Rev. A **97**, 022304 (2018).

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

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

[56] V. K. Sridhar, Y. Chen, B. Gard, E. Barnes, and S. E. Economou, Adapt-QAOA with a classically inspired initial state, arXiv:2310.09694 (2023).

[57] A. Das and B. K. Chakrabarti, Colloquium: Quantum annealing and analog quantum computation, Rev. Mod. Phys. **80**, 1061 (2008).

[58] A. Montanari, Optimization of the Sherrington-Kirkpatrick Hamiltonian, arXiv:1812.10897 (2019).

[59] A. El Alaoui, A. Montanari, and M. Sellke, Local algorithms for maximum cut and minimum bisection on locally treelike regular graphs of large degree, arXiv:2111.06813 (2023).

[60] S. Endo, S. C. Benjamin, and Y. Li, Practical quantum error mitigation for near-future applications, Phys. Rev. X **8**, 031027 (2018).

[61] K. Temme, S. Bravyi, and J. M. Gambetta, Error mitigation for short-depth quantum circuits, Phys. Rev. Lett. **119**, 180509 (2017).

[62] Y. Li and S. C. Benjamin, Efficient variational quantum simulator incorporating active error minimization, Phys. Rev. X **7**, 021050 (2017).

[63] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, Error mitigation extends the computational reach of a noisy quantum processor, Nature (London) **567**, 491 (2019).

[64] https://github.com/Hitachi-Cambridge/PaperDynamicADAPT-QAOA-Data.

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