Fast quantum circuit cutting with randomized measurements

Angus Lowe[1,2], Matija Medvidović[1,3,4], Anthony Hayes[1], Lee J. O'Riordan[1], Thomas R. Bromley[1], Juan Miguel Arrazola[1], and Nathan Killoran[1]

[1]Xanadu, Toronto, ON, M5G 2C8, Canada
[2]Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA
[3]Center for Computational Quantum Physics, Flatiron Institute, New York, NY, 10010, USA
[4]Department of Physics, Columbia University, New York, 10027, USA

We propose a method to extend the size of a quantum computation beyond the number of physical qubits available on a single device. This is accomplished by randomly inserting measure-and-prepare channels to express the output state of a large circuit as a separable state across distinct devices. Our method employs randomized measurements, resulting in a sample overhead that is $\widetilde{O}(4^k/\varepsilon^2)$, where ε is the accuracy of the computation and k the number of parallel wires that are "cut" to obtain smaller sub-circuits. We also show an information-theoretic lower bound of $\Omega(2^k/\varepsilon^2)$ for any comparable procedure. We use our techniques to show that circuits in the Quantum Approximate Optimization Algorithm (QAOA) with p entangling layers can be simulated by circuits on a fraction of the original number of qubits with an overhead that is roughly $2^{O(p\kappa)}$, where κ is the size of a known balanced vertex separator of the graph which encodes the optimization problem. We obtain numerical evidence of practical speedups using our method applied to the QAOA, compared to prior work. Finally, we investigate the practical feasibility of applying the circuit cutting procedure to large-scale QAOA problems on clustered graphs by using a 30-qubit simulator to evaluate the variational energy of a 129-qubit problem as well as carry out a 62-qubit optimization.

1 Introduction

In this work, we consider combining measurement outcomes from multiple quantum circuits to estimate the expectation value at the output of a larger quantum circuit. Computing such expectation values is a key step in many proposals for quantum algorithms such as the QAOA [23, 25, 30], VQE [48], or in the estimation of output probabilities, for example, for quantum classifiers [52, 36].

Given a circuit comprising m gates, the standard procedure to accomplish this task is to repeatedly run the desired circuit and measure in some basis, for a total quantum runtime on the order of m/ε^2. Alternatively, one may perform a full classical simulation of the circuit, which is believed to require a runtime exponential in the number of qubits n without strong assumptions about the structure of the circuit, for example, a bounded treewidth [39], low stabilizer-rank [7, 11, 9], or low negativity [45]. Based on current hardware limitations both approaches become infeasible for moderately large n: the time cost of the classical simulation is prohibitive, while on the other hand current quantum devices are limited to small numbers of qubits. This motivates considering a hybrid approach, where results obtained on smaller quantum devices are used to solve the estimation task.

For instance, Ref. [9] suggests constructing "virtual qubits" which enable the simulation of an otherwise intractable computation. Similarly, Ref. [46] presented a framework for simulating clustered quantum circuits. A third framework, presented in Refs. [43, 42], focuses on removing entangling gates between sub-circuits. For these methods, as well as in follow-up work [49, 47, 60, 3, 4, 40], one takes advantage of large, disconnected components of the quantum circuit which may be obtained by removing a subset of wires or gates. We refer to such methods collectively as *quantum circuit cutting*. Related work analyzes hybrid approaches to solving 3SAT [17] and

quantum simulation by combining tensor network methods with small quantum devices [5, 61].

Our main result is a fast quantum circuit cutting method based on separating weakly entangled circuits through randomized measurements. Compared to the approach in [46] in which single-qubit Pauli measurements are performed, our method offers a quadratic improvement in the overhead in certain cases of interest. As a consequence, the number of wires that can be cut under a fixed budget on the runtime is approximately doubled using this approach. Furthermore, our method likely outperforms all other proposed circuit cutting methods for simulating quantum computation using circuits with appropriate structure (see Table 1).

Our approach combines intuition which stems from quantum tomography as well as the notion of a matrix-product state (MPS). Specifically, the algorithm can be viewed as an attempt to classically simulate the entanglement between qudits in an MPS with low bond dimension using repeated mid-circuit measurements. This measurement procedure is described by *measure-and-prepare* channels (equivalent to entanglement-breaking channels [31]), enabling the expression of the post-measurement state as a separable state across distinct devices. The measurements we choose for this task are based on unitary 2-designs, which are related to the sample-optimal measurements in various quantum learning settings [28, 32, 14, 18].

A point of departure from previous work is that our procedure uses synchronized measurements and preparations. For circuits with an MPS-like structure — as considered in [33] for example — this requirement may be satisfied by repeatedly executing circuits on a single device. However, if the so-called communication graph [46] of the sub-circuits contains a cycle, then our method genuinely requires multiple circuits, separated in space rather than time.

To understand the source of our speedup, we turn to the form of the equation we employ in our proposal. In any dimension $d > 0$, one may consider a collection of linear maps $\Phi_i : \mathsf{L}(\mathbb{C}^d) \to \mathsf{L}(\mathbb{C}^d)$ such that

$$\mathrm{id} = \sum_i a_i \Phi_i \ , \tag{1}$$

where $\mathrm{id} : \mathsf{L}(\mathbb{C}^d) \to \mathsf{L}(\mathbb{C}^d)$ is the identity map. Throughout, we let $\mathsf{L}(\mathbb{C}^d)$ denote the set of linear operators acting on $\mathbb{C}^d$. See Appendix A for a comprehensive set of definitions. As suggested in Ref. [46], we choose an identity of this form such that the substitution of any individual map Φ_i in a suitable location within the circuit enables the execution of a quantum computation which would otherwise be intractable. We refer to this as *wire cutting*. A similar identity is employed to decompose 2-qubit operations into product operators in Refs. [9, 42, 43, 49], and we refer to this approach as *gate cutting*. In either case, each application of the identity results in a multiplicative overhead in the runtime scaling with the 1-norm of the coefficients, e.g., $\sum_i |a_i|$ in the case where the identity takes the form in Eq. (1). This is similar to the effect of the negativity quantity associated with fully classical simulation method of [45] based on quasiprobability sampling.

As previously observed in [43, 60, 49], the overhead in circuit cutting depends on the choice of linear maps used in the decomposition. In particular, Ref. [49] shows a lower bound on the overhead for gate cutting in several cases, while Ref. [60] proposed a simulated-annealing-based search for solutions to Eq. (1) to attempt to minimize the above 1-norm quantity. Inspired by recent work demonstrating the utility of randomized measurements [18], we give an explicit decomposition of the form in Eq. (1) whose 1-norm scales with the dimension of the subspace upon which the channel acts. We argue that this bound results in our method outperforming the state-of-the-art for a natural problem (see Table 1), and are able to prove that it is at most (roughly) quadratically worse than the best possible wire cutting method by deriving an information-theoretic lower bound.

We then apply our observations to derive sufficient conditions for the efficient hybrid simulation of circuits which arise in the Quantum Approximate Optimization Algorithm (QAOA) [23, 24, 25]. This continues a line of research where the goal is to identify restricted families of circuits which can be simulated using smaller devices than might naively be expected. For instance, Ref. [9] proposed a notion of sparse circuits for which a small number of qubits could be efficiently traded for classical computation, while Refs. [46, 15] show that weakly interacting, clustered Hamiltonians can be simulated ef-

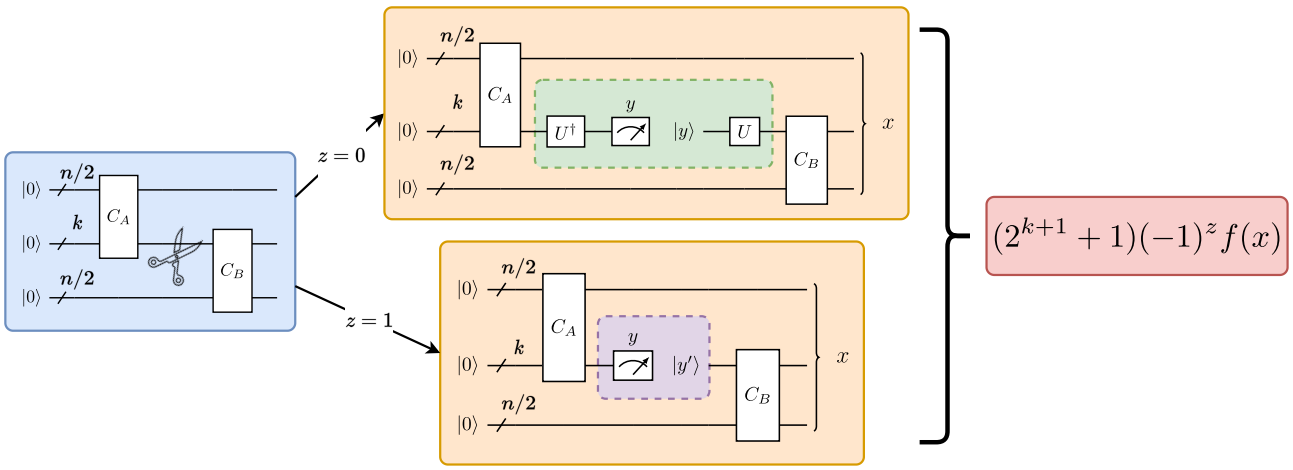

Figure 1: Example of our quantum circuit cutting scheme. One of two measure-and-prepare channels ($z = 0, 1$) is applied at random. If $z = 0$, a randomized measurement based on a unitary 2-design (e.g., a random Clifford) U is performed. If $z = 1$, the qubits are traced out, and a random basis state $|y'\rangle$ is prepared. The output is an unbiased estimator of the expectation value at the output of the original circuit.

ficiently for short times through wire cutting. We show that p-layer QAOA circuits can be efficiently simulated in a hybrid manner up to depth $p = O(\log n)$, provided the underlying graph encoding the optimization problem has a small balanced vertex separator. This addresses an open problem from Ref. [46] which asked for good ways to partition circuits based on the structure of the problem at hand. Our observations should also carry over to Trotter-Suzuki-based circuits for the Hamiltonian simulation of 2-local Hamiltonians whose interaction graph has small vertex separators, as opposed to the edge separators considered in [46]. We then show how to sample from the output distribution of a QAOA circuit to recover bitstrings encoding optimal solutions by combining measurement outcomes obtained from the smaller circuits.

We remark that prior work [35] proposed utilizing balanced vertex separators to split instances of the Max-Cut problems into smaller sub-problems for the QAOA, without considering circuit cutting methods. In addition, Refs. [50, 55] consider applying the circuit cutting method proposed in [46] to solve a modified version of the QAOA. However, these algorithms require estimating the measurement distribution of the subcircuits as a subroutine, which limits the practicality of such proposals.

We demonstrate practical speedups by classically simulating our quantum circuit cutting method applied to small instances of the QAOA and comparing to the procedure in [46]. These numerical experiments are enabled by leveraging open-source circuit cutting functionality available in PennyLane [6]. We then show that a circuit cutting procedure can be performed on large-scale QAOA problems. Using the tensor-contraction based method introduced in [46] and available in PennyLane, we break the full QAOA circuit into multiple circuit fragments of at most 30 qubits and execute them on a cluster of NVIDIA A100 40GB GPUs, acting as simulator substitutes for practical quantum hardware devices. As a result, the variational energy of a 2-layer QAOA circuit of 129 qubits is evaluated, as well as a full 1- and 2-layer QAOA optimization of a 62-qubit circuit.

2 Fast circuit cutting with randomized measurements

We first provide details for the computational model considered in this work. A quantum circuit is a directed acyclic network of gates (vertices) connected by wires (edges) which represent the qubits upon which the gates act. In general, gates represent quantum channels, i.e., completely positive and trace-preserving linear maps. The overall action of the circuit is some quantum channel acting on the input qubits, which throughout this paper we take to be $\rho_0^{\otimes n} := (|0\rangle\langle 0|)^{\otimes n}$ for a circuit with n input qubits. The *size* of the circuit is the total number of gates. As in Peng et al. [46], we adopt a model of

quantum computation in which the goal is to estimate the expectation value of some diagonal observable $O_f = \sum_{x\in\{0,1\}^n} f(x)|x\rangle\langle x|$, where $f : \{0,1\}^n \to [-1,1]$ is a classically efficiently computable function.

More specifically, given an input circuit on n qubits representing the action of some channel $\mathcal{N}$, the task is to estimate the value $\mathrm{Tr}(O_f \mathcal{N}(\rho_0^{\otimes n}))$ to within additive error ε. This is a more general version of the computational model adopted in [9], and — up to single-qubit rotations prior to measurement — encompasses the "quantum mean-value problem" defined in [8]. The model is motivated by the fact that measurement in an alternative, efficiently implementable basis, can be absorbed into the unitary operation of the circuit, and an observable whose spectral norm is larger by a polynomial factor only incurs a polynomial overhead in the runtime for the estimation task. In the remainder of this paper we further assume that f can be evaluated in constant time, for succinctness in the statement of our results.

We now define the two different measure-and-prepare channels which are used in our scheme. Let $\{|j\rangle\}_{j=1}^d$ denote the standard basis for a d-dimensional complex Euclidean space. In any dimension $d > 0$ we may consider a *random* POVM $\{U|j\rangle\langle j|U^\dagger\}_{j=1}^d$ comprising rank-1 measurement operators where $U \in \mathsf{U}(\mathbb{C}^d)$ is a random unitary operator (matrix-valued random variable) which forms a unitary 2-design (see Appendix A). For a fixed choice of such a random POVM, the first channel we consider $\Psi_0 : \mathsf{L}(\mathbb{C}^d) \to \mathsf{L}(\mathbb{C}^d)$ is then defined as

$$\Psi_0(X) = \mathbb{E}_U \left[\sum_{j=1}^d \langle j| U^\dagger X U |j\rangle \, U|j\rangle\langle j|U^\dagger \right], \quad (2)$$

for every $X \in \mathsf{L}(\mathbb{C}^d)$. This mapping is a measure-and-prepare channel, i.e., it describes a measurement performed on the d-dimensional register (using the random POVM described above) followed by preparing a new register in the state corresponding to the measurement outcome that was observed. It would also suffice for our purposes to consider a fixed rank-1 POVM whose measurement operators are based on *state* 2-designs, since the resulting measure-and-prepare channel would be identical. However, we focus on randomized basis measurements to make explicit how such a channel could be implemented in practice.

The second measure-and-prepare channel we consider is the completely depolarizing channel $\Psi_1 : \mathsf{L}(\mathbb{C}^d) \to \mathsf{L}(\mathbb{C}^d)$ defined as

$$\Psi_1(X) = \mathrm{Tr}(X)\frac{\mathbb{1}}{d}, \quad (3)$$

for every $X \in \mathsf{L}(\mathbb{C}^d)$. The following lemma implies that measure-and-prepare channels of the above form can be used to develop an improved approach to circuit cutting. The claim follows straightforwardly by evaluating the action of the channel Ψ_0, as done previously in [28, 32, 18], for example. A proof is provided in Appendix B for completeness.

Lemma 2.1. *Let d be a positive integer and Ψ_0, Ψ_1 be the channels defined in Eqs. (2) and (3), respectively, acting on d-dimensional states. Define the Bernoulli random variable $z \in \{0,1\}$ to be equal to 1 with probability $d/(2d+1)$. It holds that*

$$\mathrm{id} = (2d+1)\,\mathbb{E}_z\left[\,(-1)^z\ \Psi_z\right]. \quad (4)$$

Eq. (4) suggests a sampling-based approach to wire cutting by randomly inserting channels on the wires to be "cut". This idea is depicted in Fig. 1, and is realized by the procedure described in Algorithm 1 in Appendix B.2. The following theorem then bounds the overhead incurred from using this method to perform wire-cutting on a natural example of a clustered circuit. Once again, we ignore the classical time to compute the post-processing function f in the statement of this theorem.

Theorem 2.2 (Bipartitioning circuits)**.** *Let C be a size-m quantum circuit acting on n qubits which is a composition of circuits C_A, C_B acting non-trivially on sets of qubits $A, B \subseteq [n]$, respectively. If $|A \cap B| \leq k$, then quantum computation using C can be simulated to within accuracy ε in time $O(4^k(m+k^2)/\varepsilon^2)$ by a quantum circuit acting on at most $\max\{|A|,|B|\}$ qubits using the procedure described in Algorithm 1.*

The proof of the theorem is given in Appendix B.3. The theorem follows from Lemma 2.1 by using the fact that random k-qubit Clifford operators: (i) form a unitary 3-design [34, 59, 62], (ii) can be sampled efficiently [56], and (iii) can be implemented using poly-depth circuits [2]. We remark that while

	Type	Overhead	Communication
Thm. 1 in [9]	Gate cutting	$2^{O(nD)}$	None
Thm. 1 in [46]	Wire cutting	16^k	None
Thm. 5 in [42]	Gate cutting	$\min\{9^{kD}, 16^k\}$	None
Eq. (5.1) in [49]	Gate cutting	4^{kD}	LOCC
Thm. 2.2	Wire cutting	4^k	LOCC

Table 1: Upper bounds on the runtime of circuit cutting methods applied to circuits of the form given in Fig. 1, omitting polynomial factors in n, k, D, and ε. It is assumed that the task is to simulate a quantum computation with this circuit using a device comprising at most $n/2+k$ qubits ("doubling" the number of qubits). The "Type" column identifies whether the procedure exploits decompositions of gates or of the identity operation applied to wires. Here D is a sparseness parameter, denoting the assumption that the qubits being cut participate in at most D multi-qubit gates. Note that in this case, classical communication may be achieved by recycling qubits on a single quantum device.

random Clifford operations on many qubits can be challenging to implement in practice, we expect they may be feasible for small values of k, which is the regime suitable for efficient circuit cutting. Additionally, though we appeal to Hoeffding's inequality to bound the sample complexity in our proof, it may be possible to achieve a better scaling in terms of k for certain applications where the variance of the observables is non-trivially bounded, which we leave as a possible direction for future work.

When $|A| \approx |B| \approx n/2$ and k is a small constant, this technique effectively doubles the number of qubits which can be simulated given a quantum device. In Table 1 we compare the overheads of several circuit cutting methods assuming tightness of the upper bounds shown in each respective work (see Sec. 3.3 for numerical evidence that this is approximately the case for the wire cutting method in Ref. [46]). In the table, the setting being considered is a special case of that in Theorem 2.2: we have a composition of two circuits acting on wires belonging to overlapping sets A and B with $|A| = |B| = n/2 + k$, and the goal is to run the quantum computation using a device limited to $n/2 + k$ wires. The scenario is depicted in the left-hand side of Fig. 1. To enable the comparison with gate-cutting methods, it is necessary that we assume each circuit satisfies a certain sparseness property, introduced in Ref. [7]. Namely, we assume that the qubits in $A \cap B$ participate in at most D multi-qubit gates. Otherwise, gate-cutting methods may incur an unbounded overhead to accomplish the simulation task.

A natural question is whether the method described in Algorithm 1 saturates the best possible overhead for wire cutting. For example, one might wonder whether a subexponential runtime, or runtime on the order of 2^{ck} for some $0 < c < 1$ can be achieved. We rule out these possibilities using an information-theoretic lower bound on the sample complexity of a task that reduces to wire-cutting, which we describe in more detail in Appendix C. Roughly speaking, a procedure for wire cutting enables one to succeed at a difficult quantum state discrimination task by cutting n wires, similar to the lower bounds on classical shadows which appear in [32].

Theorem 2.3 (Informal)**.** *Any procedure for bipartitioning the circuit C through wire cutting necessarily incurs a worst-case overhead of $\Omega(2^k/\varepsilon^2)$.*

We now explain how the procedure generalizes to multiple applications of the identity in Lemma 2.1. Consider a depth-L, n-qubit quantum circuit comprising rounds $1, \ldots, L$ of commuting quantum gates which implements the channel $\mathcal{N}$. We say that a subset of the wires in the circuit are *parallel* if and only if there exists an $i \in \{1, \ldots, L-1\}$ such that each wire connects a gate applied in round $j \leq i$ to a gate in round $j \geq i+1$. In a circuit diagram, this corresponds to a set of wires which may be bisected by a single vertical line. Suppose that the circuit may be separated into disconnected sub-circuits by removing ℓ subsets of parallel wires, and let $k_1, \ldots, k_\ell$ denote the number of wires in each subset. In Fig. 2, for example, a depth-4 circuit has

$\ell = 2$ subsets of k_1 and k_2 parallel wires, respectively, and is partitioned into three disconnected sub-circuits by removing these wires (depicted as inserting a measure-and-prepare operation). Then by repeated application of Lemma 2.1, we have

$$\mathrm{Tr}(O_f \mathcal{N}(\rho_0^{\otimes n})) = \mathop{\mathbb{E}}_{z_1,\ldots,z_\ell} \left[\mathrm{Tr}(O_f \mathcal{N}'_{z_1,\ldots,z_\ell}(\rho_0^{\otimes n})) \prod_{j=1}^{\ell} (2d_j + 1)(-1)^{z_j} \right]. \tag{5}$$

Here, $z_1, \ldots, z_\ell$ are independent Bernoulli random variables equal to 1 with probability $d_j/(2d_j+1)$ where $d_j = 2^{k_j}$, for each $j \in [\ell]$, and $\mathcal{N}'_{z_1,\ldots,z_\ell}$ is the modified quantum channel which is implemented by substituting the channel Ψ_{z_j} for the identity map at the j^{th} location in the original circuit, for each $j \in [\ell]$. For fixed $z_1, \ldots, z_\ell$, an unbiased estimator of the quantity $\mathrm{Tr}(O_f \mathcal{N}'_{z_1,\ldots,z_\ell}(\rho_0^{\otimes n}))$ can be obtained by measuring the output wires of the modified circuit in the computational basis. Furthermore, if removal of the wires in each of the ℓ subsets separates the circuit into r disconnected sub-circuits, *fragments*, then the circuit can be executed using r quantum devices, each possibly with fewer qubits than the original.

This setting introduces the following drawbacks: (i) the overhead is comparatively large when $k_j = 1$ for every $j \in [\ell]$ (as opposed to when ℓ is small and each k_j is large) and (ii) classical communication between multiple separate devices is required. The latter issue is avoided in Algorithm 1 since the first circuit in the sequence of circuits has no dependence on the results from the second circuit, and so its output wires could be measured simultaneously with the wires to be cut. These measurement results could then be stored and used at a later time to construct the estimator, while qubits on that same device are recycled in order to execute the second circuit. But consider the communication graph [46] representing the flow of communication between different circuits. Then this recycling procedure is not possible whenever the graph contains a cycle, which may be the case in general. In this case, the circuit sends its partial measurement results to another, and has to wait until receiving a measurement outcome before proceeding with its remaining gates.

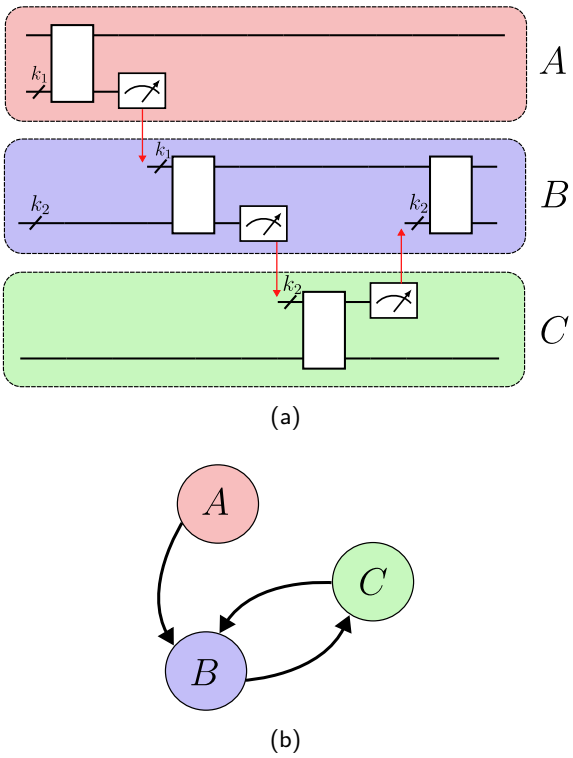

Figure 2: (a) Example of a 4-qubit circuit with 3 wire cuts. Red lines represent classical communication between distinct devices. (b) Communication graph for circuit in Fig. 2a. Since the communication graph has a cycle, classical communication between at least two distinct devices is required to execute the wire cutting procedure.

3 Circuit cutting for QAOA

In this section, we combine the above results with observations related to the structure of QAOA circuits, and perform numerical simulations of quantum circuit cutting applied to the QAOA.

3.1 Structure of QAOA circuits for circuit cutting

Here, we establish sufficient conditions for the efficient application of quantum circuit cutting to QAOA circuits for the Max-Cut problem. A key aspect to consider is the wires which must be cut to separate a single layer of the QAOA circuit for Max-Cut into multiple smaller fragments. For a given input graph $G = (V, E)$ encoding an instance of the Max-Cut problem, the cost operator takes the form

$$H_C = \sum_{(i,j)\in E} Z_i Z_j \,, \tag{6}$$

where Z_i is the single-qubit Pauli-Z operator act-

ing on the i-th qubit. The Max-Cut solution corresponds to the state with the minimal expectation value of this cost operator. One may alternatively view this operator as a 2-local Hamiltonian whose *interaction graph* is precisely the input graph. The QAOA consists of applying alternating layers of mixing unitaries based on single-qubit rotations and cost unitaries based on the above cost operator. Since the mixing unitary comprises single-qubit gates which do not influence qubit connectivity, this part of the circuit can safely be ignored for the purposes of this section. We may therefore focus on the cost unitary,

$$U_C(\gamma) = e^{-i\gamma H_C}, \tag{7}$$

which is implemented by a product of commuting two-qubit ZZ rotations, one for each edge in E. The fact that these operations commute also implies that the cost unitary can be implemented using multiple different circuits, corresponding to permutations of the terms in the product. This observation allows us to make the simplifying assumption that gates in the QAOA circuit are applied in an order which respects a chosen partition of the gates for the purpose of circuit cutting, i.e., all gates in a given partition of the edges are applied contiguously in the circuit.

In the specific case of circuit bipartitions, we make use of the idea of a *balanced vertex separator* of an undirected graph G, which is a subset of vertices whose removal leaves two sets of disconnected vertices of roughly equal size, for example, of size at most $2|V|/3$.

Theorem 3.1 (Imprecise, see restatement in Appendix D)**.** *Suppose the graph $G = (V, E)$ with $|V| = n$ has some known balanced vertex separator $S \subseteq V$ such that $|S| = \kappa$. Then quantum computation with a p-layer QAOA circuit for Max-Cut on G can be simulated to within accuracy ε using a pair of (non-entangled) quantum devices with $n - \Omega(n)$ qubits in time $\widetilde{O}(2^{\alpha p \kappa}/\varepsilon^2)$ for some universal constant α.*

We once again ignore classical time to compute the objective function of a given bit-string in the statement of this theorem. The proof is in Appendix D. The theorem follows by combining observations about the structure of single-layer QAOA circuits with Eq. (5) as well as an intermediate result concerning how the number of wires to be cut scales with the number of layers p. We are thus able to introduce $\Omega(n)$ "virtual qubits" in polynomial time on up to log-depth circuits, so long as the underlying graph for the QAOA has a certain structure. In particular, this is a regime for which certain negative results on the overall effectiveness of the QAOA do not apply. For example, it is known that for $o(\log n)$-depth QAOA both the average- and worst-case performance is limited compared to the best classical algorithms [10, 22, 21].

Based on the above observations, choosing suitable instances of the Max-Cut problem for circuit cutting is related to finding graphs with small vertex separators. Concretely, we should pick a class of graphs whose vertices are clustered, such that the graph can be (vertex-)separated into roughly equal-sized components. The number of vertices in the separator relates to the number of cuts required in the overall circuit, and thus the overhead in performing circuit cutting. We therefore suggest the class of graphs depicted in Fig. 3 for benchmarking QAOA circuit cutting. These graphs are defined to be those which can be written as r sets of n vertices connected through $r - 1$ groups of $k = O(1)$ vertices. These connecting groups of vertices form a vertex separator of size $k(r - 1)$, and the overhead for computing expectation values scales exponentially with this quantity.

We may also consider the more general case in which there is no prior information about the structure of the graphs available. While the problem of finding the minimal balanced vertex separator is known to be NP-hard [12], there is a polynomial-time classical algorithm for finding an $O(\sqrt{\log \kappa})$ approximation to the optimal solution based on semi-definite programming relaxation [26]. Here, κ is the size of the minimal vertex separator. Hence, even without prior knowledge of the minimal vertex separator, one may employ such classical algorithms to search for approximate solutions on the given input graph, and the overhead in Theorem 3.1 holds with κ the minimal vertex separator, up to a factor of $O(\sqrt{\log \kappa})$ in the exponent.

3.2 Sampling from cut circuits

In this section, we establish that it is possible to recover optimal solutions from circuit fragments which have been used to perform the QAOA. We emphasize that this *sampling* aspect of the al-

gorithm has not been addressed in any previous works on circuit cutting known to the authors. Our analysis leads to a simple suggestion that helps bridge the gap between procedures for circuit cutting and their application to problems of practical interest.

In the Max-Cut problem, the objective function maps bitstrings (partitions of vertices) to their cost (number of edges cut), $f : \{0,1\}^n \to [0, M]$. Here and throughout, we consider the graph $G = (V, E)$ with $n = |V|$ and $M = |E| \leq n^2$. The goal of the QAOA is to return a bitstring x such that $f(x)$ is close to maximal, with high probability. The optimized QAOA circuit produces the state $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$ according to parameters $\boldsymbol{\gamma}, \boldsymbol{\beta}$. This results in the measurement distribution given by $q(x) = |\langle x|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle|^2$ such that $\mathbb{E}_{x\sim q} f(x)$ is ideally close to the maximum. Let us denote this expectation by μ from now on, and note that $\mu \in [0, M]$.

Given access to samples from q, it is straightforward to show that

$$\Pr_{x\sim q}[x : f(x) \geq \mu] \geq \frac{1}{M}. \tag{8}$$

In other words, after around M trials one obtains a bit string x which has cost at least μ. Now consider the case where the QAOA circuit is being partitioned into disconnected fragments by removing k wires in total. Using Eq. (5) in the case where $\ell = k$ (individual wires are being cut), this expression implies

$$q(x) = \mathrm{Tr}(|x\rangle\langle x|\mathcal{N}(\rho_0^{\otimes n})) \tag{9}$$

$$= 5^k \mathop{\mathbb{E}}_{z_1,\ldots,z_k} \left[\widetilde{q}(x|z_1,\ldots,z_k) \prod_{j=1}^{k} (-1)^{z_j}\right] \tag{10}$$

for every $x \in \{0,1\}^n$, where $z_1, \ldots, z_k$ are independent Bernoulli random variables which determine the settings in a modified quantum circuit, as described in Sec. 2, and $\widetilde{q}(x|z_1,\ldots,z_k)$ is the probability of obtaining the outcome x from the modified circuit conditioned on fixed settings $z_1, \ldots, z_k$. Considering the absolute value of each of the terms in the expectation, the expectation in the right-hand side of Eq. (10) is at most

$$\widetilde{q}(x) := \mathop{\mathbb{E}}_{z_1,\ldots,z_k} \left[\widetilde{q}(x|z_1,\ldots,z_k)\right]. \tag{11}$$

This is the marginal probability of observing the outcome $x \in \{0,1\}^n$ in the modified circuit.

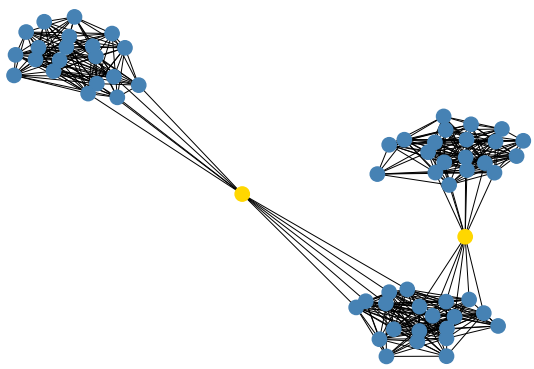

Figure 3: An example problem graph used as input for QAOA simulations. This graph contains $r = 3$ clusters, $n = 20$ nodes within each cluster and $k = 1$ vertex separators between each cluster. Note that the cluster graphs are not complete; an edge exists between any two nodes in a cluster with probability of 0.7. An edge exists between a node in a cluster and its neighbouring vertex separator with probability 0.3.

Therefore, it holds that

$$\widetilde{q}(S) \geq q(S)/5^k \quad \forall S \subseteq \{0,1\}^n, \tag{12}$$

where $q(S) := \sum_{x\in S} q(x)$ and likewise for the distribution $\widetilde{q}$. Hence, using Eq. (8) we get

$$\Pr_{x\sim\widetilde{q}}[x : f(x) \geq \mu] \geq \frac{1}{5^k M}, \tag{13}$$

which in turn implies one would have to wait at most 5^k times as long in expectation to sample a bit string x such that $f(x) \geq \mu$ using the smaller circuit fragments. This is analogous to the overhead incurred for expectation values: in both cases, the cost of circuit cutting is a need to repeatedly execute the circuit by an amount that grows exponentially in the number of wires cut.

We remark that repeating the above analysis using the wire cutting procedure derived in the proof of Theorem 1 in Ref. [46] results in an improved overhead of 4^k for the sampling task, and as mentioned previously does not require synchronizing preparations to measurement outcomes in the modified circuits.

3.3 Numerical comparison for circuit bipartitions

Here we numerically benchmark our circuit cutting method against the methods given in Ref. [46], and compare to the theoretical bounds.

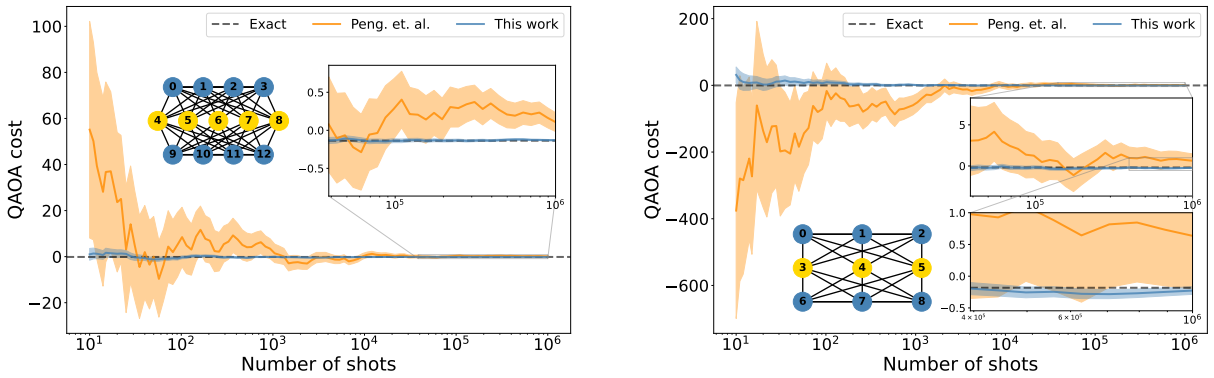

Figure 4: Convergence of the QAOA cost expectation as a function of the number of shots. In both insets, yellow nodes indicate the wires which are cut. Shaded intervals represent one standard deviation from the mean. **Left:** $p = 1$ layer, larger graph (13 qubits). **Right:** $p = 2$ layers, medium-sized graph (9 qubits).

The supporting code used to generate these numerical results can be found in [1]. We focus on a family of graphs described in Sec. 3.1, with specific examples depicted in Fig. 4. These graphs can later be generalized, as seen on Fig. 3. The cost variances for these graphs are examined at QAOA depth $p = 1$ and $p = 2$ as functions of the number of required device shots. The results can be seen on Fig. 4. All reference cost values have been calculated numerically exactly due to relatively small test graphs (maximum 13 qubits).

For all graphs considered, QAOA parameters [23] $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ have been fixed to their optimal values

$$\boldsymbol{\gamma}^*, \boldsymbol{\beta}^* = \arg\min \langle \boldsymbol{\gamma}, \boldsymbol{\beta} | H_C | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle , \tag{14}$$

where $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$ is the QAOA output state given labeled parameters.

One of the main advantages of the circuit cutting method proposed in this work is the faster convergence to the correct value of the target observable. To meet the requirement of the cost function being in the interval $[-1, 1]$, we rescale the expression given in Eq. (6) by $M = |E|$, the number of edges in the graph. Simulations were performed using PennyLane [6], an open-source Python software framework for quantum differentiable programming, for both cutting methods.

As discussed in Appendix D, a QAOA circuit with p layers will generally require at least p separate cuts, one per QAOA layer. For $p \geq 2$, the measure-and-prepare nature of both methods introduces mid-circuit measurements that are not commonly supported on qubit simulators. To accommodate mixed quantum states that would be generated by circuit cutting on a real quantum device, one can either employ a mixed-state simulator or introduce auxiliary qubits. Our implementation of the randomized channel method uses the former, while the PennyLane implementation of the Pauli-based method of Peng et al. [46] employs the latter.

The randomized method introduced in Sec. 2 outperforms the method based on Pauli measurements [46] in terms of convergence speed and accuracy for all considered graph sizes, cut sizes, and QAOA depths. Improvements range from marginal to orders of magnitude. For larger cut sizes k, and especially at QAOA depth $p = 2$, at 10^6 shots, the randomized method offers reliable cost estimates, while the Pauli method still exhibits standard deviations larger than the target interval $[-1, 1]$ as shown in Fig. 4. We observe that the randomized channel approach converges faster in all QAOA instances checked, as shown on Fig. 5.

3.4 Large-scale simulations

Having demonstrated the faster convergence of our circuit cutting method over existing approaches for small-sized QAOA problems, we now switch focus toward showing the applicability of a circuit cutting procedure to larger problem sizes of above 50 qubits, a scale that is challenging for direct execution on near-term quantum hardware devices. Our objective is to provide a proof-of-principle evaluation and optimization of a large-scale QAOA problem when cut into smaller fragments of at most 30 qubits. To overcome the limitations of existing 30-qubit quantum hardware,

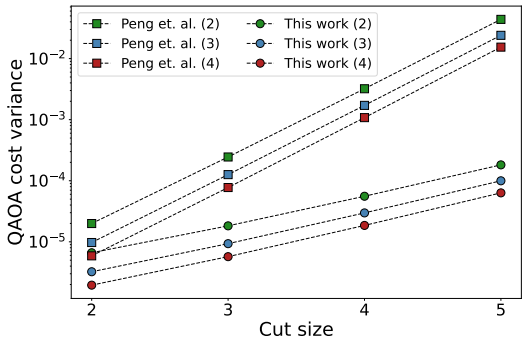

Figure 5: Numerical experiment results for QAOA cost variance as a function of the cut size at 10^6 shots. Exponential scaling agrees with the bounds presented in Sec. 2. The parenthesized number indicates the cut size for a given method at $p = 1$.

we use a cluster of GPU-based simulators as a substitute.

This section uses the circuit cutting method introduced in Theorem 2 of Peng et al. [46, 47], which involves performing process tomography for each circuit fragment and contracting the resulting tensors. The tensor-contraction-based method is fully supported in PennyLane [6] and is compatible with both simulator- and hardware-based devices. When performed using simulation, this method is analogous to existing tensor network methods that aim to optimize the order of contractions in the network [27].

As before, we focus on instances from the class of graphs introduced in Sec. 3.1 for the Max-Cut problem, consisting of r clusters of n nodes connected in a chain by smaller clusters of k nodes, as exemplified in Fig. 3. The choice of problem parameters r, n and k as well as the number of QAOA layers p determines the total number of unique sub-circuits N required for circuit cutting as well as the maximum number of qubits m among all sub-circuits, as discussed in Appendix E.

The simulations detailed here were performed on the NERSC Perlmutter supercomputer using a cluster of NVIDIA A100 40GB GPUs running PennyLane's cuQuantum-enabled [19] Lightning-GPU simulator device. Each GPU node in Perlmutter is equipped with 4 NVIDIA A100 40GB GPUs, an AMD EPYC 7763 CPU, and networked with the HPE Slingshot-10 interconnect. Each GPU supports executing circuits of up to $m = 30$ qubits. The N circuit fragments generated due to cutting were distributed within the cluster using the Ray parallel execution library [44]. Supporting utility functions, data and methods used for this section are available in [1].

We first detail the execution of QAOA circuits with parameters $p = 2$, $n = 25$ and $k = 1$ for a range of cluster numbers $r \in \{3, 4, 5\}$, resulting in circuits with 77, 103, and 129 qubits, respectively. Such circuits could not be executed directly on a 30-qubit device, but by using circuit cutting we can break the circuit up into multiple smaller circuit fragments of number $N(r)$, depending on the choice of r. In this case, $N(3) = 1812$, $N(4) = 3540$ and $N(5) = 5268$, growing linearly with r as can be seen in Appendix E. To confirm the linear scaling numerically, we used simulation-based circuit cutting to evaluate the expectation value of the QAOA circuit with a randomly selected set of circuit parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ for all choices of r and recorded the execution time, resulting in the plot shown in Fig. 6. An analysis of the classical efficiency of our distributed GPU-based simulation is provided in Appendix F.

We also investigate the ability to optimize a large-scale QAOA problem using simulated circuit cutting. Given the ability to evaluate a circuit, it is straightforward to extend access to gradients using methods like parameter shift [53] or finite difference, which require repeated circuit evaluation. We optimize a 62-qubit QAOA problem using the gradient descent algorithm, providing gradients by combining circuit-cutting-inspired parallelization with the multi-parameter finite-difference method. Using the class of clustered graphs with parameters $n = 20$, $r = 3$ and $k = 1$, we consider QAOA circuits with $p = 1$ and $p = 2$ layers.

In the case of $p = 1$, an exact expression for the QAOA cost is available [57, 41], from which we are able to obtain the analytic optimal value to use as a baseline comparison. It is shown in Fig. 7 that an optimization using simulated circuit cutting is able to achieve this optimal cost value. For $p = 2$ there is no analytic expression known for the cost function, making the global minimum harder to find. Nevertheless, the cutting-based optimization results in a decreasing cost function and tends towards a value of -0.51. Although the result does not achieve a global minimum, since the $p = 1$ case is able to reach a lower value of -33.8, this provides a demonstration of the abil-

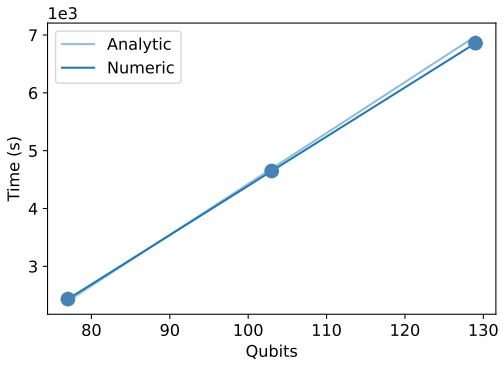

Figure 6: Execution time vs. the number of qubits for an evaluation of a QAOA circuit using circuit cutting. The simulation is distributed over 10 GPU-nodes (4 GPUs per node) and the input problem graph uses the fixed parameters $p = 2$, $n = 25$, and $k = 1$ while the number of clusters r is varied to increase the number of qubits. An analytic line is also included based upon Eq. (66), where the number of unique sub-circuits $N(r)$ has been scaled by an inferred constant execution time of 1.325 seconds per circuit.

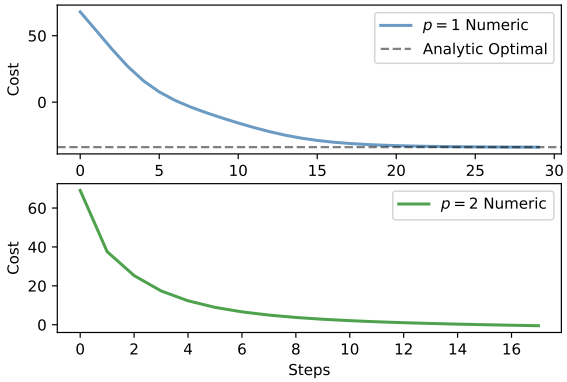

Figure 7: Cost vs. steps for a full 62-qubit QAOA optimization using circuit cutting. The input problem graph parameters are fixed at $n = 20$, $r = 3$, and $k = 1$. For $p = 1$ QAOA layers, the optimization was run over 2 GPU-nodes and took around 30 minutes, converging to the global minimum of -33.8. For $p = 2$, the optimization was run over 10 GPU-nodes and took around 12 hours. In this case, a local minimum is obtained, although this may be improved upon by careful choice of initial circuit parameters [38].

ity to perform optimization using circuit cutting methods at larger depths p.

4 Conclusion and outlook

In this work, we presented a circuit cutting method based on randomized measurements that provides a quadratic runtime improvement over the current state-of-the-art for circuits where multiple neighbouring wires are cut simultaneously. Our method requires classical communication between circuit fragments to coordinate measurement outcomes and state preparation. For circuits with matrix-product-state structure, the synchronization between circuit fragments can be carried out by repeatedly executing smaller circuits, even on a single device. For general circuits, our algorithm requires multiple circuit executions across many devices, separated in space rather than time.

Our results raise the question of whether additional improvements may be possible in this setting. With this in mind, we derived an information-theoretic lower bound that is quadratically lower than our method. It is an open question whether such a lower bound is in fact achievable or whether it can be further tightened.

We chose the QAOA algorithm as a testbed for the new method. An emphasis was put on the practical and operational aspects for real devices – we give an algorithm based on randomized measure-and-prepare channels as well as a way to sample the cut circuit. To the best of our knowledge, the latter has not been addressed in any previous works, which instead focus on estimating expectation values. Numerical simulations indicate a large speedup over previous state-of-the-art for circuit cutting.

Finally, as a more general exploration of circuit cutting methods, we give results on large-scale simulations of QAOA circuits over many GPUs acting as proxies for individual $\sim$ 30-qubit devices. We performed forward passes of up to 129 qubits and full optimization procedures for 62-qubit circuits. These results demonstrate that our software implementation of circuit cutting methods, built on top of PennyLane, indeed enables small-scale quantum devices to successfully emulate the results of a large circuit. Although additional difficulties may arise when employing quantum hardware instead of simulators, these numerical experiments are a testament to the practicality of large-scale circuit cutting workflows.

Acknowledgements

We thank Zeyue Niu, Zain H. Saleem, Michael A. Perlin, and Yuri Alexeev for useful discussions.

This work was supported under DARPA project HR0011-21-9-0073, Quantum Compilation of Unitaries and Tensors and used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award DDR-ERCAP0022246 under the QIS@Perlmutter program. MM acknowledges support from the CCQ graduate fellowship in computational quantum physics. The Flatiron Institute is a division of the Simons Foundation.

References

[1] https://github.com/XanaduAI/randomized-measurements-circuit-cutting. (2022).

[2] Scott Aaronson and Daniel Gottesman. "Improved simulation of stabilizer circuits". In: *Phys. Rev. A* 70 (5 Nov. 2004), p. 052328. DOI: 10.1103/PhysRevA.70.052328. URL: https://link.aps.org/doi/10.1103/PhysRevA.70.052328.

[3] J. Avron, Ofer Casper, and Ilan Rozen. "Quantum advantage and noise reduction in distributed quantum computing". In: *Phys. Rev. A* 104 (5 Nov. 2021), p. 052404. DOI: 10.1103/PhysRevA.104.052404. URL: https://link.aps.org/doi/10.1103/PhysRevA.104.052404.

[4] Thomas Ayral et al. "Quantum Divide and Compute: Hardware Demonstrations and Noisy Simulations". In: *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. 2020, pp. 138–140. DOI: 10.1109/ISVLSI49217.2020.00034.

[5] F. Barratt et al. "Parallel quantum simulation of large systems on small NISQ computers". In: *npj Quantum Information* 7.1 (May 2021), p. 79. ISSN: 2056-6387. DOI: 10.1038/s41534-021-00420-3. URL: https://doi.org/10.1038/s41534-021-00420-3.

[6] Ville Bergholm et al. *PennyLane: Automatic differentiation of hybrid quantum-classical computations*. 2018. DOI: 10.48550/ARXIV.1811.04968. URL: https://arxiv.org/abs/1811.04968.

[7] Sergey Bravyi and David Gosset. "Improved Classical Simulation of Quantum Circuits Dominated by Clifford Gates". In: *Phys. Rev. Lett.* 116 (25 June 2016), p. 250501. DOI: 10.1103/PhysRevLett.116.250501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.116.250501.

[8] Sergey Bravyi, David Gosset, and Ramis Movassagh. "Classical algorithms for quantum mean values". In: *Nature Physics* 17.3 (Jan. 2021), pp. 337–341. DOI: 10.1038/s41567-020-01109-8. URL: https://doi.org/10.1038/s41567-020-01109-8.

[9] Sergey Bravyi, Graeme Smith, and John A. Smolin. "Trading Classical and Quantum Computational Resources". In: *Phys. Rev. X* 6 (2 June 2016), p. 021043. DOI: 10.1103/PhysRevX.6.021043. URL: https://link.aps.org/doi/10.1103/PhysRevX.6.021043.

[10] Sergey Bravyi et al. "Obstacles to Variational Quantum Optimization from Symmetry Protection". In: *Phys. Rev. Lett.* 125 (26 Dec. 2020), p. 260505. DOI: 10.1103/PhysRevLett.125.260505. URL: https://link.aps.org/doi/10.1103/PhysRevLett.125.260505.

[11] Sergey Bravyi et al. "Simulation of quantum circuits by low-rank stabilizer decompositions". In: *Quantum* 3 (Sept. 2019), p. 181. ISSN: 2521-327X. DOI: 10.22331/q-2019-09-02-181. URL: https://doi.org/10.22331/q-2019-09-02-181.

[12] Thang Nguyen Bui and Curt Jones. "Finding good approximate vertex and edge partitions is NP-hard". In: *Information Processing Letters* 42.3 (1992), pp. 153–159. ISSN: 0020-0190. DOI: https://doi.org/10.1016/0020-0190(92)90140-Q. URL: https://www.sciencedirect.com/science/article/pii/002001909290140Q.

[13] Francesco Buscemi and Nilanjana Datta. "The Quantum Capacity of Channels With Arbitrarily Correlated Noise". In: *IEEE Transactions on Information Theory* 56.3 (2010), pp. 1447–1460. DOI: 10.1109/TIT.2009.2039166.

[14] Senrui Chen et al. "Robust Shadow Estimation". In: *PRX Quantum* 2 (3 Sept. 2021), p. 030348. DOI: 10.1103/PRXQuantum.2.030348. URL: https://link.aps.org/doi/10.1103/PRXQuantum.2.030348.

[15] Andrew M. Childs et al. "Theory of Trotter Error with Commutator Scaling". In: *Physical Review X* 11.1 (Feb. 2021). DOI: 10.1103/physrevx.11.011020. URL: https://doi.org/10.1103%2Fphysrevx.11.011020.

[16] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, Apr. 2005. DOI: 10.1002/047174882x. URL: https://doi.org/10.1002/047174882x.

[17] Vedran Dunjko, Yimin Ge, and J. Ignacio Cirac. "Computational Speedups Using Small Quantum Devices". In: *Phys. Rev. Lett.* 121 (25 Dec. 2018), p. 250501. DOI: 10.1103/PhysRevLett.121.250501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.121.250501.

[18] Andreas Elben et al. *The randomized measurement toolbox*. 2022. DOI: 10.48550/ARXIV.2203.11374. URL: https://arxiv.org/abs/2203.11374.

[19] Leo Fang et al. *NVIDIA/cuQuantum: cuQuantum v22.05.0*. Version v22.05.0. May 2022. DOI: 10.5281/zenodo.6574510. URL: https://doi.org/10.5281/zenodo.6574510.

[20] Robert M. Fano. *Transmission of Information: a Statistical Theory of Communications*. MIT Press, 1966.

[21] Edward Farhi, David Gamarnik, and Sam Gutmann. *The Quantum Approximate Optimization Algorithm Needs to See the Whole Graph: A Typical Case*. 2020. DOI: 10.48550/ARXIV.2004.09002. URL: https://arxiv.org/abs/2004.09002.

[22] Edward Farhi, David Gamarnik, and Sam Gutmann. *The Quantum Approximate Optimization Algorithm Needs to See the Whole Graph: Worst Case Examples*. 2020. DOI: 10.48550/ARXIV.2005.08747. URL: https://arxiv.org/abs/2005.08747.

[23] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014. DOI: 10.48550/ARXIV.1411.4028. URL: https://arxiv.org/abs/1411.4028.

[24] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. *A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem*. 2014. DOI: 10.48550/ARXIV.1412.6062. URL: https://arxiv.org/abs/1412.6062.

[25] Edward Farhi and Aram W Harrow. *Quantum Supremacy through the Quantum Approximate Optimization Algorithm*. 2016. DOI: 10.48550/ARXIV.1602.07674. URL: https://arxiv.org/abs/1602.07674.

[26] Uriel Feige, MohammadTaghi Hajiaghayi, and James R. Lee. "Improved Approximation Algorithms for Minimum Weight Vertex Separators". In: *SIAM Journal on Computing* 38.2 (2008), pp. 629–657. DOI: 10.1137/05064299X. eprint: https://doi.org/10.1137/05064299X. URL: https://doi.org/10.1137/05064299X.

[27] Johnnie Gray and Stefanos Kourtis. "Hyper-optimized tensor network contraction". In: *Quantum* 5 (Mar. 2021), p. 410. ISSN: 2521-327X. DOI: 10.22331/q-2021-03-15-410. URL: https://doi.org/10.22331/q-2021-03-15-410.

[28] M Guţă et al. "Fast state tomography with optimal error bounds". In: *Journal of Physics A: Mathematical and Theoretical* 53.20 (Apr. 2020), p. 204001. DOI: 10.1088/1751-8121/ab8111. URL: https://doi.org/10.1088/1751-8121/ab8111.

[29] Jeongwan Haah et al. "Sample-Optimal Tomography of Quantum States". In: *IEEE Transactions on Information Theory* 63.9 (2017), pp. 5628–5641. DOI: 10.1109/TIT.2017.2719044.

[30] Stuart Hadfield et al. “From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz”. In: *Algorithms* 12.2 (2019). ISSN: 1999-4893. DOI: 10.3390/a12020034. URL: https://www.mdpi.com/1999-4893/12/2/34.

[31] Michael Horodecki, Peter W. Shor, and Mary Beth Ruskai. “Entanglement Breaking Channels”. In: *Reviews in Mathematical Physics* 15.06 (2003), pp. 629–641. DOI: 10.1142/S0129055X03001709. eprint: https://doi.org/10.1142/S0129055X03001709. URL: https://doi.org/10.1142/S0129055X03001709.

[32] Hsin-Yuan Huang, Richard Kueng, and John Preskill. “Predicting many properties of a quantum system from very few measurements”. In: *Nature Physics* 16.10 (June 2020), pp. 1050–1057. DOI: 10.1038/s41567-020-0932-7. URL: https://doi.org/10.1038/s41567-020-0932-7.

[33] William Huggins et al. “Towards quantum machine learning with tensor networks”. In: *Quantum Science and Technology* 4.2 (Jan. 2019), p. 024001. DOI: 10.1088/2058-9565/aaea94. URL: https://doi.org/10.1088/2058-9565/aaea94.

[34] Richard Kueng and David Gross. *Qubit stabilizer states are complex projective 3-designs*. 2015. DOI: 10.48550/ARXIV.1510.02767. URL: https://arxiv.org/abs/1510.02767.

[35] Junde Li, Mahabubul Alam, and Swaroop Ghosh. *Large-scale Quantum Approximate Optimization via Divide-and-Conquer*. 2021. DOI: 10.48550/ARXIV.2102.13288. URL: https://arxiv.org/abs/2102.13288.

[36] Seth Lloyd et al. *Quantum embeddings for machine learning*. 2020. DOI: 10.48550/ARXIV.2001.03622. URL: https://arxiv.org/abs/2001.03622.

[37] Angus Lowe and Ashwin Nayak. *Lower bounds for learning quantum states with single-copy measurements*. 2022. DOI: 10.48550/ARXIV.2207.14438. URL: https://arxiv.org/abs/2207.14438.

[38] Danylo Lykov et al. *Sampling Frequency Thresholds for Quantum Advantage of Quantum Approximate Optimization Algorithm*. 2022. DOI: 10.48550/ARXIV.2206.03579. URL: https://arxiv.org/abs/2206.03579.

[39] Igor L. Markov and Yaoyun Shi. “Simulating Quantum Computation by Contracting Tensor Networks”. In: *SIAM Journal on Computing* 38.3 (Jan. 2008), pp. 963–981. DOI: 10.1137/050644756. URL: https://doi.org/10.1137%2F050644756.

[40] Simon C. Marshall, Casper Gyurik, and Vedran Dunjko. *High Dimensional Quantum Machine Learning With Small Quantum Computers*. 2022. DOI: 10.48550/ARXIV.2203.13739. URL: https://arxiv.org/abs/2203.13739.

[41] Matija Medvidović and Giuseppe Carleo. “Classical variational simulation of the Quantum Approximate Optimization Algorithm”. In: *npj Quantum Information* 7.1 (June 2021). DOI: 10.1038/s41534-021-00440-z. URL: https://doi.org/10.1038/s41534-021-00440-z.

[42] Kosuke Mitarai and Keisuke Fujii. “Constructing a virtual two-qubit gate by sampling single-qubit operations”. In: *New Journal of Physics* 23.2 (Feb. 2021), p. 023021. DOI: 10.1088/1367-2630/abd7bc. URL: https://doi.org/10.1088/1367-2630/abd7bc.

[43] Kosuke Mitarai and Keisuke Fujii. “Overhead for simulating a non-local channel with local channels by quasiprobability sampling”. In: *Quantum* 5 (Jan. 2021), p. 388. DOI: 10.22331/q-2021-01-28-388. URL: https://doi.org/10.22331%2Fq-2021-01-28-388.

[44] Philipp Moritz et al. *Ray: A Distributed Framework for Emerging AI Applications*. 2017. DOI: 10.48550/ARXIV.1712.05889. URL: https://arxiv.org/abs/1712.05889.

[45] Hakop Pashayan, Joel J. Wallman, and Stephen D. Bartlett. “Estimating Outcome Probabilities of Quantum Circuits Using Quasiprobabilities”. In: *Phys. Rev. Lett.* 115 (7 Aug. 2015), p. 070501. DOI:

10.1103/PhysRevLett.115.070501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.115.070501.

[46] Tianyi Peng et al. "Simulating Large Quantum Circuits on a Small Quantum Computer". In: *Physical Review Letters* 125.15 (Oct. 2020). ISSN: 1079-7114. DOI: 10.1103/physrevlett.125.150504. URL: http://dx.doi.org/10.1103/PhysRevLett.125.150504.

[47] Michael A. Perlin et al. "Quantum circuit cutting with maximum-likelihood tomography". In: *npj Quantum Information* 7.1 (Apr. 2021). DOI: 10.1038/s41534-021-00390-6.

[48] Alberto Peruzzo et al. "A variational eigenvalue solver on a photonic quantum processor". In: *Nature Communications* 5.1 (July 2014). DOI: 10.1038/ncomms5213. URL: https://doi.org/10.1038/ncomms5213.

[49] Christophe Piveteau and David Sutter. *Circuit knitting with classical communication*. 2022. DOI: 10.48550/ARXIV.2205.00016. URL: https://arxiv.org/abs/2205.00016.

[50] Zain H. Saleem et al. *Quantum Divide and Conquer for Combinatorial Optimization and Distributed Computing*. 2021. arXiv: 2107.07532 [quant-ph].

[51] Igal Sason and Sergio Verdú. "f - Divergence Inequalities". In: *IEEE Transactions on Information Theory* 62.11 (2016), pp. 5973–6006. DOI: 10.1109/TIT.2016.2603151.

[52] Maria Schuld et al. "Circuit-centric quantum classifiers". In: *Physical Review A* 101.3 (Mar. 2020). DOI: 10.1103/physreva.101.032308. URL: https://doi.org/10.1103%2Fphysreva.101.032308.

[53] Maria Schuld et al. "Evaluating analytic gradients on quantum hardware". In: *Phys. Rev. A* 99 (3 Mar. 2019), p. 032331. DOI: 10.1103/PhysRevA.99.032331. URL: https://link.aps.org/doi/10.1103/PhysRevA.99.032331.

[54] Hayk Shoukourian et al. "Predicting the energy and power consumption of strong and weak scaling HPC applications". In: *Supercomputing Frontiers and Innovations* 1.2 (2014), pp. 20–41. DOI: 10.14529/jsfi140202.

[55] Wei Tang and Margaret Martonosi. *ScaleQC: A Scalable Framework for Hybrid Computation on Quantum and Classical Processors*. 2022. DOI: 10.48550/ARXIV.2207.00933. URL: https://arxiv.org/abs/2207.00933.

[56] Ewout Van Den Berg. "A simple method for sampling random Clifford operators". In: *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2021, pp. 54–59. DOI: 10.1109/QCE52317.2021.00021.

[57] Zhihui Wang et al. "Quantum approximate optimization algorithm for MaxCut: A fermionic view". In: *Phys. Rev. A* 97.2 (Feb. 2018), p. 022304. ISSN: 24699934. DOI: 10.1103/PhysRevA.97.022304.

[58] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018. DOI: 10.1017/9781316848142.

[59] Zak Webb. "The Clifford group forms a unitary 3-design". In: (2015). DOI: 10.48550/ARXIV.1510.02769. URL: https://arxiv.org/abs/1510.02769.

[60] Roeland Wiersema et al. *Circuit connectivity boosts by quantum-classical-quantum interfaces*. 2022. DOI: 10.48550/ARXIV.2203.04984. URL: https://arxiv.org/abs/2203.04984.

[61] Xiao Yuan et al. "Quantum Simulation with Hybrid Tensor Networks". In: *Phys. Rev. Lett.* 127 (4 July 2021), p. 040501. DOI: 10.1103/PhysRevLett.127.040501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.127.040501.

[62] Huangjun Zhu. "Multiqubit Clifford groups are unitary 3-designs". In: *Phys. Rev. A* 96 (6 Dec. 2017), p. 062336. DOI: 10.1103/PhysRevA.96.062336. URL: https://link.aps.org/doi/10.1103/PhysRevA.96.062336.

A Preliminaries

We explicitly define random variables throughout, including matrix-valued random variables where relevant. We let $x \in A$ denote a random variable x which takes values in the set A. For any positive integer N we let $[N]$ denote the set $\{1, \ldots, N\}$. We use sans-serif font to denote subspaces of complex finite-dimensional Euclidean vector spaces, or operators acting on these spaces. For any positive integer d we let $\mathsf{L}(\mathbb{C}^d)$ denote the set of square linear operators acting on $\mathbb{C}^d$, $\mathsf{H}(\mathbb{C}^d)$ the subset of operators in $\mathsf{L}(\mathbb{C}^d)$ which are Hermitian, $\mathsf{Psd}(\mathbb{C}^d)$ the subset of operators in $\mathsf{H}(\mathbb{C}^d)$ which are positive semidefinite, and $\mathsf{D}(\mathbb{C}^d)$ the subset of operators in $\mathsf{Psd}(\mathbb{C}^d)$ which have unit trace (quantum states). We also let $\mathsf{U}(\mathbb{C}^d)$ denote the set of unitary operators acting on $\mathbb{C}^d$ and $\mathsf{S}(\mathbb{C}^d)$ the set of unit-norm vectors in $\mathbb{C}^d$ (vector-representation of pure states). We distinguish between identity operators acting on $\mathbb{C}^d$ and on $\mathsf{L}(\mathbb{C}^d)$ by adopting the notation $\mathbb{1}$ for the former and id for the latter, using subscripts to denote the space on which the operators act if this is not clear from context. Our main theorem will make use of structured POVMs based on unitary t-designs (in the case where $t = 2$), which we define below.

Definition A.1 (Unitary t-design)**.** *For positive integers $t, d > 0$, a* unitary t-design *is a random unitary operator $U \in \mathsf{U}(\mathbb{C}^d)$ which for any $X \in \mathsf{L}(\mathbb{C}^d)^{\otimes t}$ satisfies*

$$\mathbb{E}_U \left[U^{\otimes t} X (U^\dagger)^{\otimes t} \right] = \int_{\mathsf{U}(\mathbb{C}^d)} V^{\otimes t} X (V^\dagger)^{\otimes t} \, \mathrm{d}\mu(V) \tag{15}$$

where μ is the Haar measure on $\mathsf{U}(\mathbb{C}^d)$.

In the case where U in the above definition is a discrete random variable taking values in $\{U_1, U_2 \ldots, U_m\} \subset \mathsf{U}(\mathbb{C}^d)$ with probabilities $p_1, p_2, \ldots, p_m$, respectively, one may associate with this unitary t-design the ensemble of unitaries $\{(p_i, U_i)\}_{i=1}^m$. We denote by $\mathrm{Tr}_j(X)$ the partial trace over the j^{th} space, where it should be made clear from context which subspace the index j corresponds to. We also use the notation $\mathrm{Tr}_S(\cdot)$ for a set $S \subset \mathbb{Z}_+$ to denote tracing out a subspace corresponding to multiple qubits on a quantum circuit.

B Proof of main result

B.1 Proof of Lemma 2.1

The lemma is restated for convenience. Recall that we have

$$\Psi_0(X) = \mathbb{E}_U \left[\sum_{j=1}^{d} \langle j | U^\dagger X U | j \rangle \, U | j \rangle \langle j | U^\dagger \right] \tag{2}$$

for every $X \in \mathsf{L}(\mathbb{C}^d)$, where $U \in \mathsf{U}(\mathbb{C}^d)$ is a unitary 2-design, and

$$\Psi_1(X) = \mathrm{Tr}(X) \mathbb{1}/d \tag{3}$$

for every $X \in \mathsf{L}(\mathbb{C}^d)$.

Lemma 2.1. *Let d be a positive integer and Ψ_0, Ψ_1 be the channels defined in Eqs. (2) and (3), respectively, acting on d-dimensional states. Define the Bernoulli random variable $z \in \{0, 1\}$ to be equal to 1 with probability $d/(2d+1)$. It holds that*

$$\mathrm{id} = (2d+1) \, \mathbb{E}_z \left[\, (-1)^z \, \Psi_z \right]. \tag{16}$$

Proof. Since we assume that $U \in \mathsf{U}(\mathbb{C}^d)$ is a unitary 2-design, for any $|v\rangle \in \mathsf{S}(\mathbb{C}^d)$ we have

$$\mathbb{E}_U \left[(U|v\rangle\langle v|U^\dagger)^{\otimes 2} \right] = \int_{\mathsf{U}(\mathbb{C}^d)} (V|v\rangle\langle v|V^\dagger)^{\otimes 2} \, \mathrm{d}\mu(V) \tag{17}$$

$$= \frac{1}{d(d+1)} \left(\mathbb{1} \otimes \mathbb{1} + W \right) \tag{18}$$

where $W \in \mathsf{U}((\mathbb{C}^d)^{\otimes 2})$ is the swap operator, and the second line is a well-known identity that follows by exploiting the permutation-invariance of Haar integrals of this form (cf. Eq. (7.179) in [58]). Using Eq. (18) as well as the linearity of trace, we may write the action of the first channel Ψ_0 on a linear operator $X \in \mathsf{L}(\mathbb{C}^d)$ as

$$\Psi_0(X) = \mathbb{E}_U \left[\sum_{j=1}^{d} \langle j| U^\dagger X U |j\rangle \, U|j\rangle\langle j|U^\dagger \right] \tag{19}$$

$$= \mathrm{Tr}_1 \left((X \otimes \mathbb{1}) \sum_{j=1}^{d} \mathbb{E}_U \left[(U|j\rangle\langle j|U^\dagger)^{\otimes 2} \right] \right) \tag{20}$$

$$= \frac{1}{d+1} \left[\mathrm{Tr}_1 \left((X \otimes \mathbb{1}) \right) + \mathrm{Tr}_1 \left((X \otimes \mathbb{1}) W \right) \right] \tag{21}$$

$$= \frac{1}{d+1} \left(\mathrm{Tr}(X)\mathbb{1} + X \right), \tag{22}$$

where the final line follows from the identity $\mathrm{Tr}_1((A \otimes \mathbb{1})W) = A$ for any square linear operator A. Also, $\Psi_1(X) = \mathrm{Tr}(X)\mathbb{1}/d$ for every $X \in \mathsf{L}(\mathbb{C}^d)$ by definition. Substituting into Eq. (22), we have

$$\Psi_0(X) = \frac{1}{d+1} \left[d\Psi_1(X) + X \right] \quad \forall X \in \mathsf{L}(\mathbb{C}^d) \tag{23}$$

which, upon rearranging, gives

$$X = (d+1)\Psi_0(X) - d\Psi_1(X) \tag{24}$$

$$= (2d+1) \left(\frac{d+1}{2d+1} \Psi_0(X) - \frac{d}{2d+1} \Psi_1(X) \right) \tag{25}$$

$$= (2d+1) \, \mathbb{E}_z \left[(-1)^z \Psi_z(X) \right] \tag{26}$$

for every $X \in \mathsf{L}(\mathbb{C}^d)$. This proves the claim. □

B.2 Bipartitioning pseudocode

We describe in full detail the circuit cutting procedure used in Theorem 2.2 using the pseudocode below, in Algorithm 1. As in the statement of the theorem, C is an n-qubit circuit composed of sub-circuits C_A, C_B acting on $A, B \subseteq [n]$ respectively, and $|A \cap B| = k$. Recalling our notation for the model of quantum computation introduced in Sec. 2, for any function $f : \{0,1\}^n \to [-1,1]$ we have $O_f = \sum_{x \in \{0,1\}^n} f(x)|x\rangle\langle x|$. The overall action of the quantum circuit C is to implement a quantum channel which we denote by $\mathcal{N}$, and the goal is to estimate $\mathrm{Tr}(O_f \mathcal{N}(\rho_0^{\otimes n}))$. Note also that we assume that $A \cup B = [n]$ for simplicity.

B.3 Proof of Theorem 2.2

We first restate the result for convenience.

Theorem 2.2. *Let C be a size-m quantum circuit acting on n qubits which is a composition of circuits C_A, C_B acting non-trivially on sets of qubits $A, B \subseteq [n]$, respectively. If $|A \cap B| \leq k$, then quantum computation using C can be simulated to within accuracy ε in time $O(4^k(m+k^2)/\varepsilon^2)$ by a quantum circuit acting on at most $\max\{|A|, |B|\}$ qubits using the procedure described in Algorithm 1.*

Algorithm 1 Circuit cutting procedure in Theorem 2.2.

Input: $f : \{0,1\}^n \to [-1,1]$, quantum circuit C as in Theorem 2.2, accuracy parameter ε
Output: Random variable Y s.t. $\mathbb{E}[Y] = \mathrm{Tr}(O_f \mathcal{N}(\rho_0^{\otimes n}))$ and $|Y| \leq O(2^k)$ with certainty

1: $z \leftarrow 1$ with probability $2^k/(2^{k+1}+1)$, 0 otherwise
2: Initialize wires A to $|0\rangle_A$
3: Apply circuit C_A to wires A
4: **if** $z = 0$ **then**
5: $\quad V \leftarrow$ random Clifford operator on k qubits
6: $\quad$ Apply circuit $V^\dagger$ to wires $A \cap B$
7: $\quad y \leftarrow$ measurement of wires $A \cap B$
8: **else**
9: $\quad y \leftarrow \mathrm{Unif}(\{0,1\}^k)$
10: **end if**
11: $x_A \leftarrow$ measurement of wires $[n] \backslash B$
12: Initialize wires B to $|y\rangle_{A\cap B} |0\rangle_{[n]\backslash A}$
13: Apply circuit C_B to wires B
14: $x_B \leftarrow$ measurement of wires B
15: $x \leftarrow x_A x_B$
16: $Y \leftarrow (2^{k+1}+1)(-1)^z f(x)$
17: **return** Y

Proof. Define $d := 2^k$. To prove the claim regarding time and space resources required, first note that the random variable $Y = (2^{k+1}+1)(-1)^z f(x)$ in Algorithm 1 is bounded in magnitude by $|Y| \leq 2d+1$ with certainty. Hence by Hoeffding's Inequality the sample mean of $N = O(d^2/\varepsilon^2) = O(4^k/\varepsilon^2)$ iterations of Algorithm 1 suffices to estimate the expectation value of Y to within accuracy ε with high probability. Now, suppose we have a device comprising $\max\{|A|, |B|\}$ qubits. If $z = 0$, the procedure up to and including Line 11 can be performed on this device in time $O(m + k^2)$. This follows from the fact that there exists an efficient procedure to sample a depth-$O(k \log(k))$ circuit which implements a random Clifford operator, running in time $O(k^2)$ [56]. If $z = 1$ then no sampling of random Cliffords is required, and the procedure yields x_A and y in time $O(m)$. In Line 12, the same device may then be re-initialized to the state $|y\rangle |0\rangle$. Finally, the application of the circuit C_B takes time $O(m)$ for a total runtime of $O(m + k^2)$ to produce a single sample.

It remains to show that Y is an unbiased estimator of $\mathrm{Tr}(O_f \mathcal{N}(\rho_0^{\otimes n}))$, where $\mathcal{N}$ is the channel implemented by the circuit C. Consider the channels Ψ_0 and Ψ_1 defined in Eqs. (2) and (3), respectively. Lines 5-7 and 12 perform the action of the channel Ψ_0 on qubits in $A \cap B$. This follows from the fact that the random Clifford operators comprise a 3-design [34, 62, 59]. Similarly, lines 9 and 12 perform the action of the channel Ψ_1 on qubits in $A \cap B$. Deferring the measurement in Line 11, the procedure therefore has the following action on the initial state for a fixed $z \in \{0,1\}$:

$$\rho_0^{\otimes n} \to (\mathcal{U}_A \otimes \mathrm{id}_{[n]\backslash A})(\rho_0^{\otimes n}) \qquad \text{(Line 3)} \qquad (27)$$

$$\to (\mathrm{id}_{[n]\backslash B} \otimes \Psi_z \otimes \mathrm{id}_{[n]\backslash A})(\mathcal{U}_A \otimes \mathrm{id}_{[n]\backslash A})(\rho_0^{\otimes n}) \qquad \text{(Lines 4-10)} \qquad (28)$$

$$\to \tau^{(z)} := (\mathrm{id}_{[n]\backslash B} \otimes \mathcal{U}_B)(\mathrm{id}_{[n]\backslash B} \otimes \Psi_z \otimes \mathrm{id}_{[n]\backslash A})(\mathcal{U}_A \otimes \mathrm{id}_{[n]\backslash A})(\rho_0^{\otimes n}). \qquad \text{(Line 13)} \qquad (29)$$

Here, $\mathcal{U}_A$ and $\mathcal{U}_B$ are unitary channels acting on wires (qubits) A and B respectively, such that the overall action of the circuit is given by

$$\mathcal{N}(\rho_0^{\otimes n}) = (\mathrm{id}_{[n]\backslash B} \otimes \mathcal{U}_B)(\mathcal{U}_A \otimes \mathrm{id}_{[n]\backslash A})(\rho_0^{\otimes n}). \qquad (30)$$

Let z now be the random variable defined in Line 1 of Algorithm 1, by Lemma 2.1 and the linearity

of channels we have

$$\mathbb{E}[(2d+1)(-1)^z \tau^{(z)}] = \mathcal{N}(\rho_0^{\otimes n}). \tag{31}$$

Therefore, by the linearity of trace it holds that

$$\begin{aligned}
\mathbb{E}[Y] &= \mathbb{E}\left[\mathbb{E}\left[(2d+1)(-1)^z f(x) \mid z\right]\right] && (32)\\
&= \mathrm{Tr}\left(O_f \, \mathbb{E}\left[(2d+1)(-1)^z \tau^{(z)}\right]\right) && (33)\\
&= \mathrm{Tr}\left(O_f \mathcal{N}(\rho_0^{\otimes n})\right) && (34)
\end{aligned}$$

as required. □

C An information-theoretic lower bound

Let us begin by formally defining what constitutes a procedure for wire cutting. Let $\mathcal{H}_A$, $\mathcal{H}_B$, and $\mathcal{H}_C$ be complex Euclidean spaces of dimension d_A, d_B, d_C corresponding to registers A, B, C, and let $\mathcal{H} := \mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_C$. We require that any procedure for wire cutting succeeds at the following task using a fixed measurement strategy on the register B. Given (i) a unitary channel $\mathcal{U}_2$ acting on states of the register (B, C), (ii) a diagonal observable $O_f \in \mathsf{H}(\mathcal{H})$ such that $\|O_f\| \leq 1$, and (iii) an unknown state of the register (A, B, C) of the form $\rho \otimes \sigma$ where $\rho \in \mathsf{D}(\mathcal{H}_A \otimes \mathcal{H}_B)$ and $\sigma \in \mathsf{D}(\mathcal{H}_C)$, a procedure for wire cutting first performs a measurement on the register B, leaving the register A in some reduced state ρ_A. Then, the register B is prepared in the state τ_y conditioned on the outcome y having been obtained. Next, the unitary channel $\mathcal{U}_2$ is applied to the state on the composite register (B, C). Finally all registers are measured in the standard basis, resulting in an outcome $z \in [d_A d_B d_C]$. A successful procedure for wire cutting outputs an ε-accurate estimate of the expectation value $\mathrm{Tr}(O_f(\mathbb{1}_A \otimes \mathcal{U}_2)(\rho \otimes \sigma))$ using N i.i.d. copies of z.

Note that a successful procedure for this task using $N \leq O(\kappa^2/\varepsilon^2)$ iterations is implied by the existence of a decomposition of the identity of the form

$$\mathrm{id} = \sum_i a_i \Phi_i \tag{35}$$

for some collection of measure-and-prepare channels $\Phi_i : \mathsf{L}(\mathcal{H}_B) \to \mathsf{L}(\mathcal{H}_B)$ and real a_i satisfying

$$\sum_i |a_i| = \kappa. \tag{36}$$

This follows from the fact that such a decomposition could be used to perform the wire cutting task described above, resulting in an unbiased estimator of the expectation value which is bounded in magnitude by κ. Hence, our lower bound on N will also imply a lower bound on the quantity κ.

To prove our lower bound, let us consider the case where $d_A = d_C = 1$. Then we must be able to determine $\mathrm{Tr}(O_f \mathcal{U}_2(\rho))$ to within error ε using a procedure of the form described above. Namely, we measure $\rho \in \mathsf{D}(\mathcal{H}_B)$ using some POVM $\{M_y\}$ and prepare the state $\tau_y \in \mathsf{D}(\mathcal{H}_B)$ conditioned on receiving the outcome y. We then apply $\mathcal{U}_2$ on this state and measure it in the computational basis receiving outcome $z \in [d_B]$, and this procedure is repeated N times using N identical copies of the state ρ.

Since there is only one non-trivial register we will drop the subscript B from now on. Define

$$\rho_j := \frac{2\varepsilon}{d}\Pi_j + (1-\varepsilon)\frac{\mathbb{1}}{d}, \quad j = 0, 1 \tag{37}$$

where

$$\Pi_0 := \sum_{i=1}^{d/2} |i\rangle\langle i|, \qquad \Pi_1 := \mathbb{1} - \Pi_0. \tag{38}$$

These states will allow us to construct a difficult state discrimination problem which reduces to a successful procedure for wire cutting. The lower bound on the number of iterations required will then follow by bounding the information gained from the measurement statistics in a single iteration.

To accomplish this, we make use of elementary facts from information theory (see, for example, Ref. [16].) We adopt standard notation for the mutual information and conditional mutual information between two random variables, as well as the entropy and conditional entropy of a random variable. We let $D(P \parallel Q)$ denote the relative entropy between two distributions P, Q such that $\mathrm{supp}(P) \subseteq \mathrm{supp}(Q)$. We also make use of the χ^2-divergence between two discrete distributions P, Q over the sample space $\mathcal{X}$ defined through

$$D_{\chi^2}(P \parallel Q) := \sum_{x \in \mathcal{X}} Q(x) \left(\frac{P(x)}{Q(x)} - 1 \right)^2 = \left(\sum_{x \in \mathcal{X}} \frac{P(x)^2}{Q(x)} \right) - 1. \tag{39}$$

In addition to more standard facts from information theory, we require the following two results, which we state without proof. The first is a special case of Lemma 6 in Ref. [13], while the second is a looser version of Eq. (5) in Ref. [51].

Lemma C.1. *Let $P_{X,Y}$ be a discrete distribution over the sample space $\mathcal{X} \times \mathcal{Y}$, and let P_X, P_Y be the marginal distributions over $\mathcal{X}$, $\mathcal{Y}$, respectively. For any pair of discrete distributions Q_1 over $\mathcal{X}$ and Q_2 over $\mathcal{Y}$, it holds that*

$$D(P_{X,Y} \parallel P_X \otimes P_Y) \leq D(P_{X,Y} \parallel Q_1 \otimes Q_2). \tag{40}$$

Lemma C.2. *Let P, Q be discrete distributions over the sample space $\mathcal{X}$ such that* $\mathrm{supp}(P) \subseteq \mathrm{supp}(Q)$. *It holds that*

$$D(P \parallel Q) \leq \frac{1}{\ln(2)} D_{\chi^2}(P \parallel Q). \tag{41}$$

Let $U \in \mathsf{U}(\mathcal{H})$ be a Haar-random unitary operator and $x \in \{0, 1\}$ be uniformly random. Suppose we are given as input to our problem the unitary channel $\mathcal{U} : X \mapsto U^\dagger X U$, observable $\Pi_0 \in \mathsf{H}(\mathcal{H})$, and the state $U \rho_x U^\dagger \in \mathsf{D}(\mathcal{H})$. Then the wire cutting procedure produces standard basis outcomes $z_1, \ldots, z_N \in [d]$ which are independent given x and U, as well as intermediate measurement results $y_1, \ldots, y_N$ which are independent given x and U. The final output of the procedure after the N iterations enables one to correctly determine the value of $x \in \{0, 1\}$ with high probability. This follows since without loss of generality we can assume the output will be an $\varepsilon/3$-accurate estimate of the quantity $\mathrm{Tr}(\Pi_0 U^\dagger U \rho_x U^\dagger U) = \mathrm{Tr}(\Pi_0 \rho_x)$, with high probability. Now, because

$$\mathrm{Tr}(\Pi_0 \rho_0) = \frac{1}{2} + \frac{\varepsilon}{2} \qquad \text{and} \qquad \mathrm{Tr}(\Pi_0 \rho_1) = \frac{1}{2} - \frac{\varepsilon}{2} \tag{42}$$

such an $\varepsilon/3$-accurate estimate of $\mathrm{Tr}(\Pi_0 \rho_x)$ allows one to infer the value of x with certainty. By Fano's Inequality [20], the mutual information between the random variables obtained from identical iterations of this procedure $z := (z_1, \ldots, z_N)$ and the choice of state x satisfies

$$I(x : z) \geq \Omega(1). \tag{43}$$

On the other hand, we can upper bound the left-hand side of the above as follows. We have

$$
\begin{aligned}
I(x:z) &\leq I(x:z,U) && \text{(Data-Processing Inequality)} && (44)\\
&= I(x:U) + I(x:z|U) && \text{(chain rule for mut. inf.)} && (45)\\
&= I(x:z_1,\ldots,z_N|U) && (x \text{ and } U \text{ indep.}) && (46)\\
&= \sum_{i=1}^{N} I(x:z_i|U,z_{i-1},\ldots,z_1) && \text{(chain rule for mut. inf.)} && (47)\\
&= \sum_{i=1}^{N} H(z_i|U,z_{i-1},\ldots,z_1) - H(z_i|x,U,z_{i-1},\ldots,z_1) && \text{(defn. of mut. inf.)} && (48)\\
&\leq \sum_{i=1}^{N} H(z_i|U) - H(z_i|x,U) && && (49)\\
&= \sum_{i=1}^{N} I(x:z_i|U) && && (50)
\end{aligned}
$$

where the second-to-last line follows because conditioning reduces entropy, and z_i is independent of $z_{i-1},\ldots,z_1$ given x and U by the assumptions made in our criteria for what constitutes a wire cutting procedure. Hence, it remains to bound the individual mutual information terms.

Conditioning on U we find $x \to y_i \to z_i$ forms a Markov Chain, using the fact that for fixed y_i and U we defined z_i to be the outcome obtained from measuring the state $U^\dagger \tau_{y_i} U$ in the computational basis. By Data-Processing Inequality we have

$$I(x:z_i|U) \leq I(x:y_i|U). \tag{51}$$

Define $\tilde{y}$ to be the random variable y_i conditioned on U, which has conditional distribution given by $P_{\tilde{y}|x=x'}(y) = \mathrm{Tr}(M_y U \rho_{x'} U^\dagger)$ for each $x' \in \{0,1\}$ and marginal distribution $P_{\tilde{y}}(y) = \sum_{x'=0}^{1} P_{\tilde{y}|x=x'}(y)$. Also, let $P_x = (\frac{1}{2},\frac{1}{2})$, let $P_{x,\tilde{y}}$ be the joint distribution, and define a distribution Q given by $Q(y) = \mathrm{Tr}(M_y)/d$. Then the right-hand side of the above is

$$
\begin{aligned}
\mathbb{E}_U\left[D(P_{x,\tilde{y}} \,\|\, P_x \otimes P_{\tilde{y}})\right] &\leq \mathbb{E}_U\left[D(P_{x,\tilde{y}} \,\|\, P_x \otimes Q)\right] && \text{(by Lemma C.1)} && (52)\\
&= \frac{1}{2}\,\mathbb{E}_U\left[D(P_{\tilde{y}|x=0} \,\|\, Q) + D(P_{\tilde{y}|x=1} \,\|\, Q)\right] && && (53)\\
&= \mathbb{E}_U\left[D(P_{\tilde{y}|x=0} \,\|\, Q)\right]. && && (54)
\end{aligned}
$$

In the second line we made use of the unitary invariance of the Haar measure, which implies that the two terms in the sum in Eq. (53) are equal by the fact that there exits a unitary operator $V \in \mathsf{U}(\mathcal{H})$ such that $\rho_1 = V\rho_0 V^\dagger$. We then have

$$
\begin{aligned}
\mathbb{E}_U\left[D(P_{\tilde{y}|x=0} \,\|\, Q)\right] &\leq \frac{1}{\ln(2)}\,\mathbb{E}_U\left[D_{\chi^2}(P_{\tilde{y}|x=0} \,\|\, Q)\right] && \text{(by Lemma C.2)} && (55)\\
&= \frac{1}{\ln(2)}\left[\left(\sum_y \frac{\mathbb{E}_U\left[P_{\tilde{y}|x=0}(y)^2\right]}{Q(y)}\right) - 1\right]. && && (56)
\end{aligned}
$$

The Haar integral in the numerator of each of the terms above is evaluated implicitly in previous arguments for lower bounds of this type [29, 32], and explicitly as Lemma 3.2.6 in [37], by which it holds that

$$\mathbb{E}_U\left[P_{\tilde{y}|x=0}(y)^2\right] \leq Q(y)^2\left(1 + \frac{\varepsilon^2}{d+1}\right) \tag{57}$$

for every outcome y, and hence

$$\frac{1}{\ln(2)}\left[\left(\sum_y \frac{\mathbb{E}_U\left[P_{\tilde{y}|x=0}(y)^2\right]}{Q(y)}\right)-1\right] \leq \frac{\varepsilon^2}{\ln(2)(d+1)}. \tag{58}$$

In summary, we have

$$I(x:z_i|U) \leq \frac{\varepsilon^2}{\ln(2)(d+1)} \tag{59}$$

for every $i \in [N]$ so that, by Eq. (50) and the lower bound in Eq. (43) we have

$$\Omega(1) \leq I(x:z) \leq \frac{N\varepsilon^2}{\ln(2)(d+1)} \tag{60}$$

which holds if and only if $N = \Omega(d/\varepsilon^2)$. For a system comprising k qubits, this is $\Omega(2^k/\varepsilon^2)$. This proves the bound in Theorem 2.3.
It holds that $p^\star \geq \Omega(\sqrt{d})$.

D Proof of Theorem 3.1

In this appendix we offer proofs for Theorem 3.1 as well as some related results in the form of lemmas. We state a notational convenience at the outset: in order not to confuse undirected input graphs for the Max-Cut problem with the (multi-) graphs representing quantum circuits, we reserve the term *fragments* to refer to subgraphs of the latter, obtained by removing a subset of wires. Recall that we are interested in these fragments since applying the identity in Lemma 2.1 (or in the proof of Theorem 1 of Ref. [46]) to each of the removed wires allows us to simulate the circuit using smaller devices corresponding to the fragments.

The QAOA ansatz with p layers is a circuit whose structure is repeated p times in succession. To aid our discussion on the relationship between problem instances and the complexity of circuit cutting, it will be helpful to reduce the scope of our analysis to the case of a single layer. The following observation, which is not specific to the QAOA ansatz, motivates this choice. For an n-qubit circuit, we define the *support* of a set of its gates A to be the subset of qubits upon which they act non-trivially, $\mathrm{supp}(A) \subseteq [n]$. Suppose this circuit is separated into fragments $F_1, \ldots, F_r$ upon removing a subset of wires. Then we refer to the support of the subset of gates belonging to the fragment F_j as the support of the fragment itself, denoted $\mathrm{supp}(F_j) \subseteq [n]$, for each $j \in [r]$. This quantity represents the size of each of the sub-circuits formed in the circuit cutting procedure.

Lemma D.1. *If a circuit can be separated into r fragments $F_1, \ldots, F_r$ by removing ℓ sets of at most κ parallel wires (see Sec. 2), then a composition of p layers of that circuit can be separated into r fragments $F'_1, \ldots, F'_r$ by removing at most $(2p-1)\ell$ sets of at most κ parallel wires. Furthermore, $\mathrm{supp}(F_j) = \mathrm{supp}(F'_j)$ for each $j \in [r]$.*

Proof. First consider the (intact) single-layer circuit, where each gate is labelled by the fragment to which it belongs when it is separated. That is, we assign a label j to each gate if it is in the fragment F_j for some $j \in [r]$. We may construct a new directed multi-graph by contracting all the edges (wires) incident on gates with the same label. This is the *communication graph* in Ref. [46], so we adopt this terminology here as well. The edges in the communication graph represent those edges which separate the original circuit into r fragments when removed. We then choose an identical partition of the gates in each of the p repeated layers, resulting in a sequence of p identical circuits whose gates are labelled using the labels $1, \ldots, r$. The partition of the gates in the p-layer circuit is then defined by these labels, i.e., the fragment F'_j contains all gates labelled by j, for each $j \in [r]$. It is clear that $\mathrm{supp}(F_j) = \mathrm{supp}(F'_j)$ for all $j \in [r]$.

Let us now turn to the bound on the number and size of sets of parallel wires in the p-layer communication graph. There is a contribution of $p\ell$ sets of κ parallel wires in the communication graph since each layer adds ℓ sets of κ inter-fragment wires. It remains to bound the number of wires in the communication graph connecting gates in two different layers. Suppose the layers are labelled $1, \ldots, p$. For any $i \in [p-1]$, the wires corresponding to the a^{th} qubit connect gates with different labels between layers i and $i+1$ if only if the final gate acting upon the a^{th} qubit in the first layer has a different label from the first gate acting upon the a^{th} qubit. This happens only if there is a wire in the single-layer communication graph which corresponds to the a^{th} qubit. Repeating this reasoning for each qubit leads to a contribution to the total number of multi-graph edges that is bounded from above by $\ell\kappa$ for each $i \in [p-1]$. Furthermore, the wires connecting gates in layer i to layer $i+1$ are parallel by definition, and so these can be arbitrarily partitioned into at most ℓ sets of at most κ wires for each $i \in [p-1]$. In total, we have at most $p\ell + (p-1)\ell = 2p-1$ sets of at most κ parallel wires which, upon removal, yield the desired fragments $F_1, \ldots, F_r$. □

Lemma D.2. *Suppose there exists a partition of the edge set $E(G)$ into r disjoint subsets $E_1, \ldots, E_r$ and let $g_1, \ldots, g_r$ denote the subgraphs of G corresponding to these subsets of edges, respectively. Also, let $1 \leq \gamma \leq \binom{r}{2}$ denote the number of pairs of indices $i, j \in [r]$, $i < j$ such that $V(g_i) \cap V(g_j) \neq \emptyset$. Then there exists a single-layer QAOA circuit for Max-Cut on G that can be separated into fragments $F_1, \ldots, F_r$ by removing γ sets of at most κ parallel wires, where $\kappa := \max_{i \neq j} |V(g_i) \cap V(g_j)|$ and* $\text{supp}(F_j) = V(g_j)\ \forall j \in [r]$.

Proof. Note that in the single-layer QAOA circuit for Max-Cut, the vertex set V represents the qubits in the circuit. The edge set E represents the commuting 2-qubit gates acting between qubits. Consider the circuit in which all gates in E_1 are applied first, followed by E_2 and so on. Any pair of wires connecting gates in E_i to gates in E_j for some $i < j$ are then parallel by construction. Furthermore, the set of qubits acted upon by gates in both E_i and E_j is given by $V(g_i) \cap V(g_j)$, and for each element in this set there is at most one wire connecting a gate in E_i to a gate in E_j due to the ordering of gates we have chosen. Hence, there are at most γ nonempty sets of parallel wires which connect gates in E_i to E_j for some $i < j$, each of which has at most κ elements. Removing these wires produces the fragments $F_1, \ldots, F_r$ whose gates are given by $E_1, \ldots, E_r$, respectively, such that $\text{supp}(F_j) = V(g_j)$ for every $j \in [r]$ as claimed. □

We now prove the theorem using the above lemmas. For any $\delta \in [1/2, 1]$ a δ-balanced vertex separator of a graph $G = (V, E)$ is a subset of vertices $S \subseteq V$ whose removal leaves two sets of disconnected vertices each of size at most $\delta|V|$.

Theorem 3.1. *Suppose the graph $G = (V, E)$ with $|V| = n$ has some known 2/3-balanced vertex separator $S \subseteq V$ such that $|S| = \kappa$, and assume $\kappa \leq n/6$. Then quantum computation with a p-layer QAOA circuit for Max-Cut on G can be simulated using a pair of (non-entangled) quantum devices each with at most $5n/6$ qubits in time*

$$O\left(\frac{4^{(2p-1)(\kappa+2)}\text{poly}(p \cdot n)}{\varepsilon^2}\right). \tag{61}$$

Proof. We partition the edge set E in the following way. Let $\widetilde{g}_1$, $\widetilde{g}_2$ be the disconnected subgraphs which remain upon removing the vertices S, and assume without loss of generality that

$$|V(\widetilde{g}_2)| \leq \lfloor n/2 \rfloor \leq |V(\widetilde{g}_1)| \leq \lfloor 2n/3 \rfloor. \tag{62}$$

Let g_2 be the subgraph of G corresponding to the vertices $V(\widetilde{g}_2) \cup S$. We then define a partition of E into disjoint sets $E_1, E_2 \subseteq E$ by letting $E_2 := E(g_2)$ and $E_1 := E \backslash E_1$. Let g_1 be the subgraph of G corresponding to the edge set E_1. We have

$$|V(g_j)| \leq \lfloor 2n/3 \rfloor + \kappa \leq \lfloor 5n/6 \rfloor, \quad j = 1, 2 \tag{63}$$

$$\text{and} \quad V(g_1) \cap V(g_2) = S. \tag{64}$$

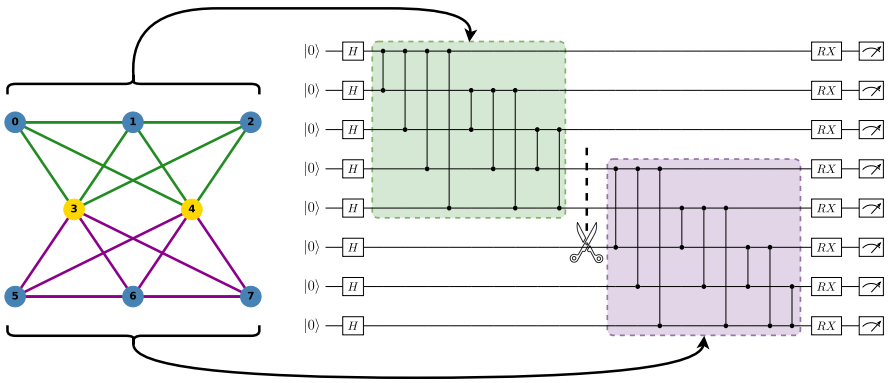

Figure 8: Schematic diagram of the idea behind Theorem 2.2. The graph on the left is the input to the Max-Cut problem, which has a balanced vertex separator given by the vertices in yellow. The two-qubit gates are applied in an order which respects the partition of the edges in the graph. We apply the resolution of the identity for circuit cutting on the two wires indicated by the dashed line.

By Lemma D.2 a single layer of the QAOA circuit ansatz can be separated into fragments F_1, F_2 by removing a single set of at most κ parallel wires, and $|\mathrm{supp}(F_j)| = |V(g_j)| \leq \lfloor 5n/6 \rfloor$ is a bound on the number of qubits in the support of fragment j. Also, by Lemma D.1, p layers of this circuit ansatz can be separated[1] into fragments F_1', F_2' by removing at most $2p-1$ sets of κ parallel wires, with $|\mathrm{supp}(F_j')| \leq \lfloor 5n/6 \rfloor$ for $j = 1, 2$. Then, since the cost operator $\mathcal{C}$ for the Max-Cut problem is a diagonal observable the expected cost $\langle \mathcal{C} \rangle$ can be written as

$$\langle \mathcal{C} \rangle = \mathop{\mathbb{E}}_{z_1,\ldots,z_{2p-1}} \left[\langle \mathcal{C}'_{z_1,\ldots,z_{2p-1}} \rangle \prod_{j=1}^{2p-1} (2^{\kappa+1}+1)(-1)^{z_j} \right] \tag{65}$$

by Eq. (5). Here, $\langle \mathcal{C}'_{z_1,\ldots,z_{2p-1}} \rangle$ is the expected value of a suitably modified circuit in which the channel Ψ_{z_i} is applied to the i^{th} group of parallel wires for each $i \in [2p-1]$, and $z_1, \ldots, z_{2p-1}$ are Bernoulli random variables as described in Lemma 2.1. The modified circuits can be executed on two quantum devices with at most $\max_j |\mathrm{supp}(F_j')| \leq \lfloor 5n/6 \rfloor$ qubits and classical communication. Using the random Clifford strategy described in the proof of Theorem 2.2, each of these circuits has at most $\mathrm{poly}(p\kappa|E|)$ gates. Measuring in the computational basis allows one to construct an unbiased estimator of the expectation value $\langle \mathcal{C} \rangle$ in a manner analogous to Line 7 in Algorithm 1. This estimator is bounded in magnitude by $(2^{\kappa+1}+1)^{(2p-1)}$ with certainty, so by Hoeffding's Inequality $(2^{\kappa+1}+1)^{2(2p-1)}/\varepsilon^2 \leq 4^{(2p-1)(\kappa+2)}/\varepsilon^2$ repetitions of this procedure suffices to estimate $\langle \mathcal{C} \rangle$ to within additive error ε, with high probability. □

E Circuit cutting complexity for QAOA with clustered graphs

We consider the execution of a multi-layered circuit resulting from encoding a problem graph of the type introduced in Sec. 3.1 for Max-Cut using QAOA. There are two key components of the simulation to consider: the number of unique sub-circuits N required for circuit cutting, and the number of qubits in each fragment circuit m. For a circuit of p layers and a problem graph of r clusters, n nodes within each cluster, and k vertex separators, we find:

$$N = N(p, r, k) = 3^{pk} 4^{(p-1)k} + (r-2) 12^{(2p-1)k} + 3^{(p-1)k} 4^{pk}, \tag{66}$$

The number of qubits required to simulate any fragment is upper bounded by $m = n + (3p-1)k$ (including additional auxiliary qubits due to mid-circuit measurements). In Sec. 3.4, by treating a GPU as a simulator analog of a quantum hardware device, we aim for one GPU per fragment execution, i.e., $m = n + (3p-1)k \leq 30$.

[1] That the wires to be removed in order to achieve this separation can be efficiently determined follows from the explicit procedure for doing so outlined in the proofs of Lemmas D.1 and D.2.

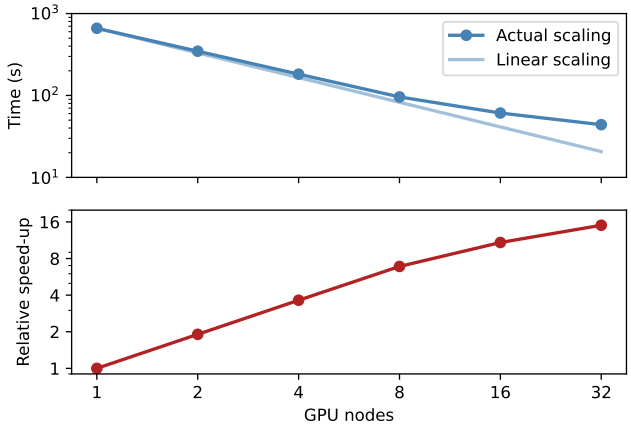

Figure 9: Execution time and relative speed-up vs. the number of GPU nodes for an execution of a 79-qubit QAOA circuit. The input problem graph parameters are fixed at $p = 1$, $r = 3$, $n = 25$, and $k = 2$, while the number of GPU nodes used in the distributed execution of the fragment circuits is varied. The ideal linear scaling relationship is also provided with reference to the single-node performance, and illustrates the divergence from ideal strong-scaling behaviour at large node numbers.

F Analyzing the efficiency of simulated circuit cutting

This appendix considers how the simulation run time scales for performing circuit cutting on QAOA circuits, for the problem graphs introduced in Sec. 3.1, as the number of classical resources is varied.

Increasing the number of GPUs used for the parallelized execution of the N fragment sub-circuits demonstrates reasonable strong-scaling behaviour [54], as it decreases the overall runtime for a QAOA circuit execution. However, given that the number of circuit executions per GPU reduces, we reach approximately 15-times speed-up using 32 GPU-nodes as depicted in Fig. 9, yielding an efficiency of approximately 47% at our largest scale evaluation. These inefficiencies are to be expected since increasing the number of GPU nodes also increases the time needed for scheduling, transmission, device setup overheads, and serial execution components which all contribute to the overall runtime.