Classically Simulating Quantum Supremacy IQP Circuits through a Random Graph Approach

Julien Codsi[1,*] and John van de Wetering[2,†]

[1] *University of Montréal*
[2] *University of Amsterdam*

(Dated: November 13, 2024)

Quantum Supremacy is a demonstration of a computation by a quantum computer that can not be performed by the best classical computer in a reasonable time. A well-studied approach to demonstrating this on near-term quantum computers is to use random circuit sampling. It has been suggested that a good candidate for demonstrating quantum supremacy with random circuit sampling is to use *IQP circuits*. These are quantum circuits where the unitary it implements is diagonal. In this paper, we introduce improved techniques for exactly classically simulating random IQP circuits. We find a simple algorithm to calculate an amplitude of an n-qubit IQP circuit with dense random two-qubit interactions in time $O(\frac{\log^2 n}{n}2^n)$, which for sparse circuits (where each qubit interacts with $O(\log n)$ other qubits) runs in $o(2^n/\text{poly}(n))$ for any given polynomial. Using a more complicated stabiliser decomposition approach we improve the algorithm for dense circuits to $O\left(\frac{(\log n)^{4-\beta}}{n^{2-\beta}}2^n\right)$ where $\beta \approx 0.396$. We further discuss how our techniques also lead to improved simulation times for IQP circuits on restricted architectures. We benchmarked our main algorithm and found that we can simulate up to 50-qubit circuits in a couple of minutes on a laptop, with 68-qubit sparse circuits taking a couple of hours. We estimate dense 70-qubit circuits are in range for large computing clusters.

Recent years have seen the development of noisy quantum computers that have enough qubits and coherence to start to probe the limits of classical simulation. In fact, in 2019 Arute et al. [1] already claimed to have reached *quantum supremacy*: a quantum computation that cannot be simulated by any classical computer in a reasonable time frame. This was done by sampling from a random quantum circuit, and computing a metric called the linear cross-entropy benchmark (XEB). Their claim was that it would take the best supercomputer in the world 10.000 years to simulate the computation they did. However, soon after that, improvements in tensor contraction techniques reduced this number to just days [2], and even hours on a moderately sized GPU cluster [3]. By allowing the simulation to produce correlated bitstrings, much higher XEB scores can be reached with fewer resources [4], and bypassing directly simulating the computation entirely, non-trivial XEB scores turned out to also be generatable in mere seconds on a single GPU [5].

This progress shows that claims of quantum supremacy should be made carefully, as improvements in classical algorithms can quickly gain orders of magnitudes in improvement. In this paper, we will consider the classical simulation of a different type of random quantum circuit that has been proposed as a good candidate for quantum supremacy experiments.

Instantaneous Quantum Polynomial (IQP) circuits are quantum circuits where the input is prepared in the all-zero state $|0\cdots 0\rangle$, the unitary is of the form $H^{\otimes n}DH^{\otimes n}$ where $H^{\otimes n}$ is a Hadamard gate applied to all the qubits, and D is a unitary consisting of polynomially many diagonal gates [6]. The name 'instantaneous' comes from the fact that all the diagonal gates commute so that there is no time order encoded into the circuit. IQP circuits were originally introduced in [6] as a simplified model of quantum computation where interesting, and hard to classically simulate, problems could be formulated. Indeed, it was proven in [7] that the ability to efficiently simulate IQP circuits would imply a collapse of the polynomial hierarchy to the third level, which is considered very unlikely. This was improved in [8] to hardness under a more reasonable additive error bound. Then in [9], it was shown that even random IQP circuits consisting of just powers of the $T = \text{diag}(1, e^{i\pi/4})$ gate and $O(n \log n)$ $CS = \text{diag}(1, 1, 1, i)$ gates are likely to be hard to simulate, and that furthermore they can be compiled onto a 2D architecture within a reasonable depth, and that they can be constructed in such a way to be resilient to some noise. These properties make these circuits an interesting candidate for quantum supremacy experiments and raise the question of where the boundary of classical simulability lies: even though the simulation is likely to be asymptotically hard, it might still be that in practical regimes, the results can still be efficiently simulated.

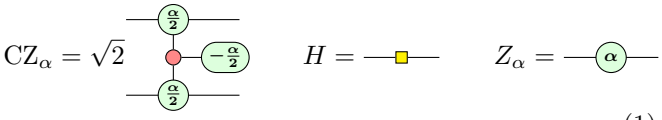
In this paper, we find better algorithms for simulating random $\{T, CS\}$ IQP circuits. We do this by realising that such circuits follow the structure of Erdős-Rényi random graphs. Such graphs have relatively large independent vertex sets. This allows us to use techniques from the stabiliser decomposition technique of simula-

* julien.codsi@umontreal.ca
† john@vdwetering.name; http://vdwetering.name

tion [10–14] to cut the circuit into a sum of smaller instances. In particular, we find we can exactly calculate amplitudes of random dense Clifford+T IQP circuits in time $O(\frac{\log^2 n}{n}2^n)$, and with a more complicated algorithm in time $O\left(\frac{(\log n)^{4-\beta}}{n^{2-\beta}}2^n\right)$ where $\beta \approx 0.396$ is the stabiliser decomposition constant of [12, 14]. For the random sparse circuits of [9] we find we can calculate an amplitude in time $O\left(\frac{n\log\log(n)}{\log(n)}2^{n\left(1-O\left(\frac{\log\log(n)}{\log(n)}\right)\right)}\right)$. Note that this bound is faster than $O(2^n/\text{poly}(n))$ for any given polynomial. We can boost the calculation of amplitudes to a procedure for sampling from the circuit, by using the 'gate-by-gate' simulation technique of [15] that avoids calculating marginals. This technique turns out to be particularly suited to IQP circuits, as it only requires an additional sample per non-diagonal gate, of which there are $O(n)$ (corresponding to the layers of Hadamard gates). Our algorithms can hence weakly sample from the dense, respectively sparse, circuits in time $O\left(\frac{(\log n)^{4-\beta}}{n^{1-\beta}}2^n\right)$, respectively $O\left(\frac{n^2\log\log(n)}{\log(n)}2^{n\left(1-O\left(\frac{\log\log(n)}{\log(n)}\right)\right)}\right)$. This should be compared to the cost $O(n^2 2^n)$, respectively $O(n\log n 2^n)$ of doing a state vector simulation, or $O(n^4 2^{O(n^2)})$, respectively $O(n^2\log^2 n 2^{O(n\log n)})$ of using stabiliser decompositions directly. Instead using the technique of [15] directly on the IQP circuit gives a scaling of $O(n2^n)$, but note that this does not allow for strong simulation (calculating of amplitudes). As far as we are aware, no other non-trivial algorithm for generic random IQP circuits has been implemented that would give a comparable benchmark. The closest is [16], which shows that for a large portion of random IQP circuits with a density of $1/n$, meaning any qubit interacts in expectation with one other qubit, we can calculate an amplitude in polynomial time using a tensor contraction algorithm (we show later in the paper how to prove this result in a significantly more straightforward way using a variation on our methods). The more recent [17] finds $O(n^3 2^{n/3})$ for a particularly structured type of IQP circuits.

Our results give asymptotic polynomial improvements over the previous best, but these improvements are also of practical significance. We implemented the simpler algorithm and found that, depending on the density of the circuit, we can calculate an amplitude of 30- to 50-qubit IQP circuits on a single CPU core on a laptop in a couple of minutes; see Figure 1. As our algorithm is easily distributed in parallel (except for the calculation of the independent set, but this cost is negligible at the size of graphs we are considering), we estimate that a 100.000 CPU core cluster could calculate an amplitude of a dense 60-qubit circuit in about an hour and that 70-qubit circuits should be within reach of the world's best supercomputers.

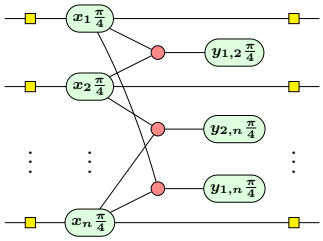
ZX-diagrams.—Our algorithm was found by representing IQP computations as ZX-diagrams. As it will be used throughout this paper, we give a brief overview of the ZX-calculus [18, 19] in the appendix. For an in-depth reference see [20]. We will be using controlled Z-phase gates, Hadamard gates, and Z phase gates, each which have a simple representation as a ZX-diagram:



(1)

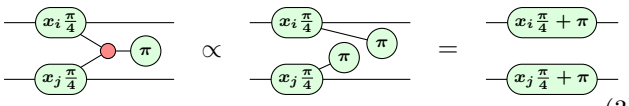
Here, the way we represent the CZ_α gates is as a *phase gadget* [21], a particularly useful type of subdiagram that features heavily in ZX-calculus-based optimisation routines [21–23]. Since the gates of Eq. (1) form a universal gate set, by composing them we can represent any quantum circuit as a ZX-diagram. Note that we will be particularly interested in the $CS = \text{CZ}_{\frac{\pi}{2}}$ gate and the $T = Z_{\frac{\pi}{4}}$ gate.

IQP circuits as ZX-diagrams.—As a ZX-diagram, an IQP circuit can be represented, up to some known global non-zero scalar as



(2)

Where the phases $x_i, y_{i,j} \in \{0, \ldots, 7\}$ arise from the powers of the T and CS gates in the circuit. We note that $y_{i,j} = 4$ corresponds to having four CS-gates in a row between the qubit i and j which is equivalent to the identity. We can see this in the ZX-diagram as:

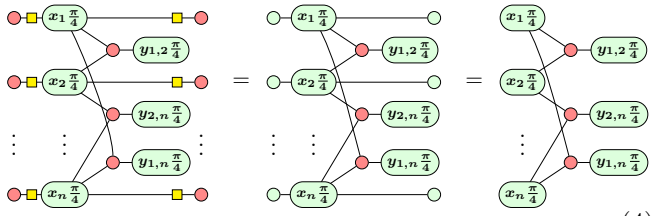


(3)

The extra π phase on the qubits is compensated by the other phases in the definition of CZ_α in Eq. (1) (this can also be seen by the fact that $CS = diag(1, 1, 1, i)$). A similar derivation can be done for $y_{i,j} = 0$. We can hence assume that the diagram is written in such a way that the trivial phase gadgets are removed. There is then a connection via a phase gadget between an x_i and x_j pair when $y_{i,j} \neq 0$ and $y_{i,j} \neq 4$.

Calculating amplitudes.—We first address the case of *strong* simulation of an IQP circuit, i.e. calculating amplitudes of the circuit. We will show later how we can derive *weak* simulation, i.e. sampling from strong simulation with linear overhead in the number of qubits. Without loss of generality, we can assume that we want to know the amplitude of observing 0^n from an IQP circuit

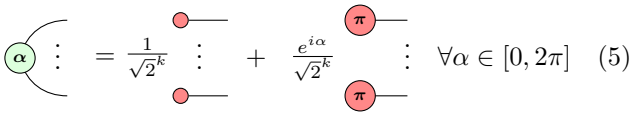
C. We can represent $\langle 0^n|C|0^n\rangle$, up to some known power of $\sqrt{2}$, as a ZX-diagram, and simplify it as follows:



(4)

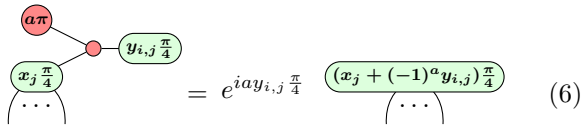
To calculate the value of such diagrams, we will use a stabiliser decomposition approach [11, 24]. We will show that it is possible to remove a vertex labelled by x_i, and all its adjacent phase gadgets from a ZX-diagram at the cost of having to solve two (smaller) instances instead of one.

The idea is to observe that the definition of a green vertex, i.e. *Z-spiders*, as a linear map (11) means we can decompose it as a sum of diagrams containing *X-spiders* (red vertices) via (13):



(5)

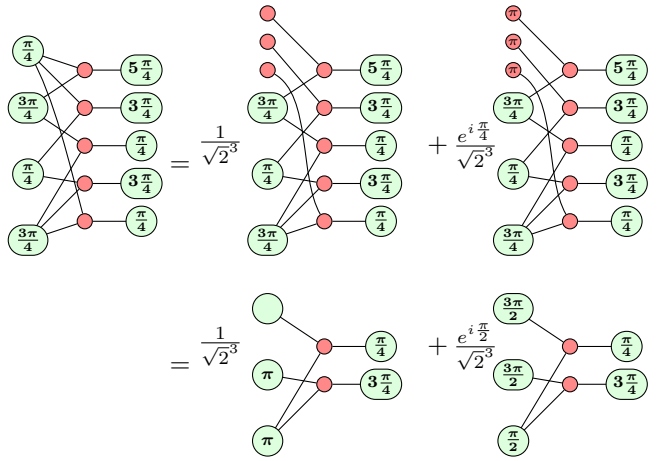
Applying this to one of the x_i vertices in Eq. (4), we can then remove its previously adjacent phase gadgets, using the rules of Eq. (15):



(6)

We can view Eq. (5) as a stabiliser decomposition, which then propagates to remove additional T-like phases that are adjacent. In this sense it can be seen as a special case of the stabiliser decomposition of many $|\text{cat}_3\rangle$ states connected together; see [14, 25].

The following example illustrates this process.

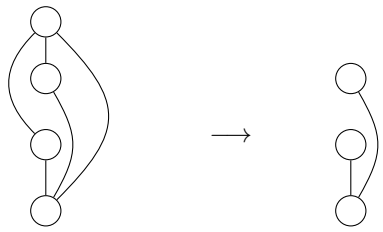


This cutting procedure takes linear time and creates two new diagrams representing IQP amplitudes with one less qubit. We could continue this process until nothing is left but a complex number, but this would require summing up $O(2^n)$ terms. However, it is possible to do better by removing qubits up until we are left with a fully disconnected diagram of some size k. This fully disconnected diagram can then be contracted in linear time as it is just the product of k complex numbers. This leads to an algorithm that runs in time $O(k2^{n-k})$. Thus, it is fruitful to find a set of qubits to remove that maximizes the value of k, i.e. the largest set of qubits that are not connected to each other. In the next section, we will consider how the strong simulation of IQP circuits can be represented by random graphs which will give us a lower bound on the value of k.

An algorithm for random IQP circuits.—We are considering two random distributions over IQP circuits. First, the random distribution for *dense* IQP circuits is obtained by uniformly and independently choosing a power of T gates x_i on every qubit and a random power of the CS gate $y_{i,j}$ for every pair of qubits. These were shown in [8], under mild assumptions, to be hard to classically sample from in the average case. Note that as $CS^4 = \text{id}$, in the dense case, there is a 3/4 chance of a non-trivial interaction between a given pair of qubits. Furthermore, since $CS^2 = CZ$, in this case, the interaction is Clifford.

Second, the random distribution for γ-*sparse* IQP circuits is obtained in a similar manner, but now, every pair of qubits only has a probability of $p = \gamma\frac{\ln(n)}{n}$ to have a power of a CS gate between them, so that each qubit interacts with $O(\ln n)$ other qubits. It has been shown that, under slightly different hardness assumptions, for γ large enough it is also hard to sample from these circuits [9].

We can define the *interaction graph* of an IQP circuit as the graph where we have one vertex per qubit and where there is an edge between two vertices iff the qubits they represent are connected by a phase gadget. For instance, here is the interaction graph from the previous example and the graph it is simplified to:



We see then that the maximal size of a disconnected graph k we can produce using this vertex-removing procedure corresponds to the *independence number* $\alpha(G)$ of the interaction graph G. We note that finding the largest independent set of a graph can be done in $\tilde{O}(1.1996^n)$ [26]. Since the search of the maximal independent set need only to be done once and its time complexity is a lot

lower than the one for the algorithm we will construct, we will omit it in the rest of the complexity analysis.

Interestingly, interaction graphs of random IQP circuits are random graphs under the Erdős–Rényi model. Specifically, random dense n-qubit IQP circuits have interaction graphs distributed like $G(n, 3/4)$. This 3/4 comes from the fact that a uniformly random power of CS has a 1/4 chance to be the identity. Similarly, random γ-sparse IQP circuit give rise to interaction graphs distributed like $G(n, \frac{3\gamma \ln(n)}{4n})$. We can hence use tools from the random graphs literature to bound the independence number obtained, which gives us a guarantee on the time complexity of our algorithm. The following classical result will be useful in particular:

Theorem 1 (Matula, 1972 [27]). *For $p \in (0,1)$, $\alpha(G(n,p))$ is tightly concentrated around $2\log_{1/(1-p)} n$. More precisely, let $b = \frac{1}{1-p}$, $\varepsilon > 0$ and $d = 2\log_b n - 2\log_b(\log_b(n)) + 2\log_b e/2 + 1$, then*

$$\lim_{n\to\infty} \mathbb{P}\left(\lfloor d-\varepsilon\rfloor \leq \alpha(G(n,p)) \leq \lfloor d+\varepsilon\rfloor\right) = 1.$$

Corollary 2. *Let $p \in (0,1)$, $b = \frac{1}{1-p}$ then $\alpha(G(n,p)) \geq 2\log_b n - 2\log_b(\log_b(n))$ with high probability.*

For random dense IQP circuits we have $p = 3/4$ and hence $b = 4$, so that $\log_b n = \frac{1}{2}\log_2 n$. This implies that the independence number of the interaction graph is with high probability bigger than $\log_2 n - \log_2\log_2 n$. Hence, our strategy for calculating an amplitude runs in $O\left((\log_2 n - \log_2\log_2 n)2^{n-\log_2 n+\log_2\log_2 n}\right) = O\left(\frac{\log^2 n}{n}2^n\right)$.

We can derive a similar bound for γ-sparse random graphs, which we prove in the Appendix.

Theorem 3. *There exists a constant $C > 0$ such that with high probability*

$$\alpha\left(G\left(n, \frac{3\gamma\ln(n)}{4n}\right)\right) \geq C\frac{n\log\log(n)}{\log(n)}.$$

This bound implies that for random γ-sparse IQP circuits, our simulation method has a time complexity of $O\left(\frac{n\log\log(n)}{\log(n)}2^{n\left(1-\frac{C\log\log(n)}{\log(n)}\right)}\right)$. Note that this bound is faster than $O(2^n/\mathrm{poly}(n))$, for any choice of polynomial (but slower than $O(2^{cn})$ for any $c < 1$).

Benchmarking.—We implemented the algorithm for computing amplitudes described above and tested it on several sizes of circuits and with different sparsities. The language used was Rust and the benchmarks ran on a single thread on a consumer laptop (Intel Core i7-10750H CPU 2.60GHz). For circuits that we could simulate in under 100 seconds, we repeated the simulation 100 times. For larger circuits, we reduced the number of repetitions to 5. Our results are shown in Figure 1. We were able to calculate amplitudes from circuits with up to 68 qubits depending on the density.

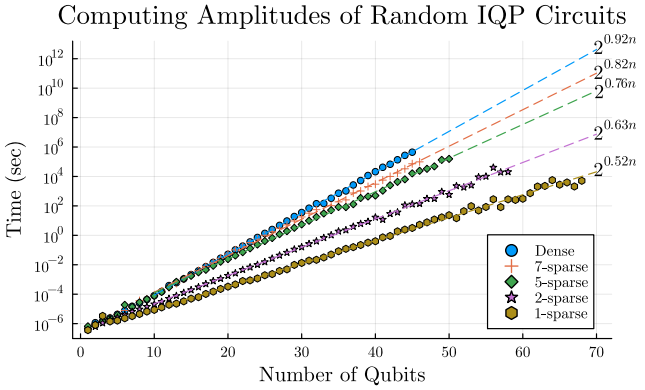


FIG. 1: Average time taken to compute a single amplitude using our algorithm. The dashed lines show exponential fits starting from $n = 10$. Note that for 7-sparse, the graphs only start to be different from the dense ones at $n = 22$. The increased variance in the datapoints for simulation times above 10^2 is because of a reduced number of instances making up the average.

Note that the data fits remarkably well to an exponential fit $c2^{\alpha n}$ where α ranges from 0.92 to 0.52. This suggests that our bound on the complexity of our algorithm might not be tight.

As the algorithm is easily parallelisable (since each term in the decomposition can be treated independently, and the calculating of the independent graph only occurs once and takes negligible time), we see that simulating circuits well into 60, or even 70, qubits should be possible with a sizable computing cluster even in the densest case. Indeed, assuming the scaling of these experiments extrapolates, 1-sparse circuits take roughly $2.52\cdot 10^{-7}\cdot 2^{0.518n}$ seconds, meaning for $n = 80$ we find a time of 209 hours ($\approx$ 1.24 weeks). For the dense circuits, we extrapolate $1.48\cdot 10^{-7}\cdot 2^{0.924n}$ seconds, which for $n = 70$ gives about $1.18\cdot 10^9$ hours, or about 11800 hours (1.34 years) on a 100.000 CPU core cluster. The computations can quite likely also be sped up significantly by running instances on a GPU instead of a CPU, for example, using the parametric ZX rewriting described in [28].

Weak simulation.—The above only describes how to calculate amplitudes of IQP circuits. To sample from circuits, we can use the strong simulation procedure described above as a subroutine of the 'gate-by-gate' simulation technique of [15] that avoids calculating marginals. This technique requires the computation of an amplitude for every non-diagonal gate in the circuit. Each of these amplitudes is based on a subcircuit of the original. We note that calculating such an amplitude is at most as hard as calculating an amplitude of the full circuit (and in fact, will often be much easier, with just the last few amplitudes involving most of the gates of the circuit). Since IQP circuits only have $2n$ non-diagonal

gates (corresponding to the layers of Hadamard gates), bootstrapping our strong simulation algorithm to a weak simulation one only adds a linear overhead $O(n)$, which in practice could be negligible.

An improved algorithm for calculating amplitudes.— It is possible to pick a different set of qubits to decompose with Eq. (5) which leads to a better asymptotic complexity in the dense case and might also give practical benefit in the sparse setting. The idea is to stop cutting vertices before completely disconnecting the diagram and then use a general stabiliser decomposition algorithm. This two-step process allows us to bring down the number of T gates from $O(n^2)$ to a more manageable $O(n)$ before using a more efficient stabiliser decomposition algorithm. To do so, let us consider the *non-Clifford interaction graph* of an IQP circuit. In this graph, there is an edge between two vertices only if they are connected by a non-Clifford phase gadget (i.e. when $y_{i,j}$ in Eq. (4) is odd). Finding the largest independent set of this graph and removing all the other qubits using Eqs. (5) and (6) then results in a diagram where all the interactions between two qubits are Clifford. This diagram can then be given to a stabiliser decomposition algorithm such as that in [14]. On average, half of the qubits in the diagram will have a non-Clifford phase that comes from the initial layer of powers of T gates in the construction of the IQP circuit. Therefore, this algorithm runs in time $O\left(2^{n-k} f(k/2)\right)$ where k is the size of the largest independent set and $f(k/2)$ is the time taken to calculate the amplitude of a diagram with $k/2$ T gates by a dedicated stabiliser decomposition algorithm. At the time of this writing, the best general-purpose stabiliser decomposition algorithm is from [14] and has a time complexity of $f(t) = O\left(t^2 2^{\beta t}\right)$ where $\beta = \log_2(3)/4 \approx 0.396$.

The main advantage of using this modified approach is that the non-Clifford interaction graph is less dense than the standard interaction graphs, while still being Erdős–Rényi random. More precisely, the non-Clifford interaction graph is distributed as $G(n, 1/2)$. By Corollary 2, this graph has with high probability an independent set of size $2\log_2 n - 2\log_2 \log_2 n$. Using this approach, calculating an amplitude then runs in $O\left(\frac{(\log n)^{4-\beta}}{n^{2-\beta}} 2^n\right)$. Since $2-\beta \approx 1.604$, this is an improvement over our first approach. Using this improved algorithm for sparse circuits results in the same asymptotic complexity as we found before, but might still be better in practice. But on the other hand, this method does introduce significant complexity in the implementation, which might, in fact, result in enough slowdown to cancel out the asymptotic benefit for relevant parameters.

Simulating circuits with low density.—We have been considering random IQP circuits with relatively high connectivity. In [16] the authors considered random IQP circuits with an interaction probability of $p = 1/n$. They showed that a large proportion of such circuits can be simulated in polynomial time. Using our technique of removing vertices, we can also prove these circuits can be efficiently simulated, but in a more elementary way. In particular, it turns out that with high probability such interaction graphs will have $O(\log n)$ cycles. By removing one vertex in each cycle, the resulting graph will be a tree, which can be contracted in linear time using standard tensor-network techniques. The proof that such graphs have $O(\log n)$ cycles is straightforward. A given set of k vertices can include a k-cycle in $k!/(2k)$ ways: the $k!$ comes from picking an ordering, the $1/k$ comes from identifying cycles that 'start' at a different vertex, and the $1/2$ comes from identifying cycles going in opposite directions. Each of these cycles has a probability of p^k of being part of the graph. Finally, there are $\binom{n}{k}$ ways to choose k vertices to include a k-cycle. Hence, the expected number of cycles in the graph is:

$$\mathbf{E}[\#\text{cycles}] = \sum_{k=3}^{n} \binom{n}{k} \frac{k!}{2k} p^k \tag{7}$$

$$\leq \frac{1}{2} \sum_{k=3}^{n} \frac{(np)^k}{k} \tag{8}$$

$$= \frac{1}{2} \sum_{k=3}^{n} \frac{1}{k} \tag{9}$$

$$\leq \frac{\log(n)}{2} \tag{10}$$

We conclude that by Markov's inequality, the interaction graph doesn't have more than $O(\log(n))$ cycles with arbitrarily high probability. We can delete one vertex per cycle so that the remaining $O(2^{\#\text{cycles}}) = O(\text{poly}(n))$ interaction graph will be trees. Trees of course have a treewidth of 1, which can hence be contracted using standard techniques in linear time [29]. The total running time will hence be $O(2^{\#\text{cycles}}) = O(\text{poly}(n))$.

Other IQP architectures.—Our method is based on the fact that the tensor network of an IQP computation can be simplified to the interaction graph, whose size only depends on the number of qubits and not on the number of gates, and on the fact that we can delete a vertex from this graph in a way that the rest of the diagram simplifies nicely. This avoids dealing with a tensor network with $O(n^2)$ tensors that a naive tensor representation would get.

We have been working with a model where any two qubits can interact. For certain architectures we however would expect interactions to occur locally between qubits that are physically close to each other. When the underlying graph has more structure than an Erdős-Rényi random graph, it is possible to derive faster algorithms. For instance, if the interactions occur on a square grid of size $\sqrt{n} \times \sqrt{n}$, then there will always be an independent set of size $n/4$ which can be found by picking vertices at a distance 2 of each other. Our algorithm would then give

a scaling of $O(n2^{3/4n})$. However, in this setting, we can in fact achieve subexponential scaling, as we will outline now.

Whenever the interaction graph belongs to a family that has a separator theorem (e.g. graphs that are planar [30], have bounded range [31], bounded genus [32], or have certain forbidden minors [33]), we can use the separators of size $O(\sqrt{n})$ to recursively decompose the diagram into smaller instances, each of which only contains at most $\frac{2}{3}n$ vertices. This method gives a computational cost of $O(\text{poly}(n)2^{O(\sqrt{n})})$. A quantum circuit simulation scheme based on graph separator theorems is described in [34], where the authors also consider IQP circuits. However, they construct the graph/tensor network directly from the circuit, which contains $O(n^2)$ gates, which means they lose the sub-exponential scaling. Instead calling their method with our description of the interaction graph of an IQP circuit does give a scaling of $O(\text{poly}(n)2^{O(\sqrt{n})})$.

Conclusion.—We found a new algorithm for exactly calculating amplitudes of random IQP circuits that, both in the dense and sparse setting, improves upon the previous asymptotic complexity and allows us to simulate large circuits in practice. Our results show that current and near-term hardware is probably not yet at a level where quantum supremacy could definitively be shown using random IQP circuit sampling. Our benchmarks suggest that it might be possible to derive better asymptotic bounds for the cost of simulating sparse circuits. We find that calculating amplitudes of dense 70-qubit random IQP circuits is within reach of a reasonably sized computational cluster. Adapting the algorithm to run on a GPU instead would probably push this number even higher. In [35] the authors argue that, assuming reasonable computational hardness assumptions, a definitive demonstration of quantum supremacy using IQP requires 208 qubits. They based this on the worlds best supercomputer (doing 10^{18} flops) taking at least 100 years to calculate it. Doing a back-of-the-envelope extrapolation (assuming our benchmark machine was doing 10^{10} flops), we estimate our algorithm could calculate an 86-qubit dense IQP amplitude, or a 130-qubit 1-sparse IQP amplitude, in that same time. This puts the cross-over point of quantum supremacy based on random IQP circuits (defined as the best current supercomputer taking 100 years to calculate an amplitude) somewhere between around 100 and 200 qubits.

Since the appearance of the original preprint of this paper, a number of other results on classical simulation of IQP circuits have appeared. In [36] the authors show that beyond some critical minimal depth, it is possible to efficiently sample from a *noisy* IQP circuit, where each applied gate applies some fixed Pauli noise channel to each qubit. The minimum depth depends on the amount of noise, and for a realistic noise probability of 1%, lies around 270. This means this algorithm is not applicable to the circuit sizes we consider. Note that this result however does suggest quantum supremacy experiments for larger numbers of qubits are also not feasible (although a step of error correction like in [9] might remedy that problem). In [17] the authors showed that the specific IQP supremacy experiments of [37] can be exactly strongly simulated in time $O(n^3 2^{n/3})$. In this result they exploited specific structure present in the circuits of [37] that is not present in the random circuits we consider.

Acknowledgements.—We would like to thank Tuomas Laakkonen for his fruitful comments and his help with the implementation of the algorithm. We would also like to thank Oliver Riordan for his help in the analysis of the independence number of γ-sparse random graphs. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[2] C. Huang, F. Zhang, M. Newman, J. Cai, X. Gao, Z. Tian, J. Wu, H. Xu, H. Yu, B. Yuan, M. Szegedy, Y. Shi, and J. Chen, Classical Simulation of Quantum Supremacy Circuits (2020), arXiv:2005.06787 [quant-ph].

[3] F. Pan, K. Chen, and P. Zhang, Solving the sampling problem of the sycamore quantum circuits, Phys. Rev. Lett. **129**, 090502 (2022).

[4] F. Pan and P. Zhang, Simulating the sycamore quantum supremacy circuits, arXiv preprint arXiv:2103.03074 (2021).

[5] X. Gao, M. Kalinowski, C.-N. Chou, M. D. Lukin, B. Barak, and S. Choi, Limitations of linear cross-entropy as a measure for quantum advantage, arXiv preprint arXiv:2112.01657 (2021).

[6] D. Shepherd and M. J. Bremner, Temporally unstructured quantum computation, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **465**, 1413 (2009).

[7] M. J. Bremner, R. Jozsa, and D. J. Shepherd, Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **467**, 459 (2011), publisher: Royal Society.

[8] M. J. Bremner, A. Montanaro, and D. J. Shepherd, Average-Case Complexity Versus Approximate Simulation of Commuting Quantum Computations, Physical Review Letters **117**, 080501 (2016), publisher: American Physical Society.

[9] M. J. Bremner, A. Montanaro, and D. J. Shepherd, Achieving quantum supremacy with sparse and noisy commuting quantum computations, Quantum **1**, 8 (2017).

[10] S. Bravyi and D. Gosset, Improved classical simulation of quantum circuits dominated by Clifford gates, Physical Review Letters **116**, 10.1103/PhysRevLett.116.250501 (2016).

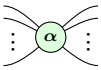
[11] S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset,

and M. Howard, Simulation of quantum circuits by low-rank stabilizer decompositions, Quantum **3**, 181 (2019).
[12] H. Qassim, H. Pashayan, and D. Gosset, Improved upper bounds on the stabilizer rank of magic states, Quantum **5**, 606 (2021).
[13] A. Kissinger and J. van de Wetering, Simulating quantum circuits with ZX-calculus reduced stabiliser decompositions, Quantum Science and Technology **7**, 044001 (2022).
[14] A. Kissinger, J. van de Wetering, and R. Vilmart, Classical Simulation of Quantum Circuits with Partial and Graphical Stabiliser Decompositions, in *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 232, edited by F. Le Gall and T. Morimae (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2022) pp. 5:1–5:13.
[15] S. Bravyi, D. Gosset, and Y. Liu, How to simulate quantum measurement without computing marginals, Phys. Rev. Lett. **128**, 220503 (2022).
[16] C.-Y. Park and M. J. Kastoryano, Complexity phase transitions in instantaneous quantum polynomial-time circuits, arXiv preprint arXiv:2204.08898 (2022).
[17] D. Maslov, S. Bravyi, F. Tripier, A. Maksymov, and J. Latone, Fast classical simulation of harvard/quera iqp circuits, arXiv preprint arXiv:2402.03211 (2024).
[18] B. Coecke and R. Duncan, Interacting quantum observables, in *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP)*, Lecture Notes in Computer Science (2008).
[19] B. Coecke and R. Duncan, Interacting quantum observables: categorical algebra and diagrammatics, New Journal of Physics **13**, 043016 (2011).
[20] J. van de Wetering, ZX-calculus for the working quantum computer scientist, Preprint (2020), 2012.13966.
[21] A. Kissinger and J. van de Wetering, Reducing the number of non-Clifford gates in quantum circuits, Physical Review A **102**, 022406 (2020).
[22] N. de Beaudrap, X. Bian, and Q. Wang, Techniques to Reduce $\pi/4$-Parity-Phase Circuits, Motivated by the ZX Calculus, in *Proceedings 16th International Conference on Quantum Physics and Logic, Chapman University, Orange, CA, USA., 10-14 June 2019*, Electronic Proceedings in Theoretical Computer Science, Vol. 318, edited by B. Coecke and M. Leifer (Open Publishing Association, 2020) pp. 131–149.
[23] M. Backens, H. Miller-Bakewell, G. de Felice, L. Lobski, and J. van de Wetering, There and back again: A circuit extraction tale, Quantum **5**, 421 (2021).
[24] S. Bravyi, G. Smith, and J. A. Smolin, Trading classical and quantum computational resources, Physical Review X **6**, 021043 (2016).
[25] H. Qassim, H. Pashayan, and D. Gosset, Improved upper bounds on the stabilizer rank of magic states, Quantum **5**, 606 (2021).
[26] M. Xiao and H. Nagamochi, Exact algorithms for maximum independent set, Information and Computation **255**, 126 (2017).
[27] D. W. Matula, The employee party problem, Notices Of The American Mathematical Society **19** (1972).
[28] M. Sutcliffe and A. Kissinger, Fast classical simulation of quantum circuits via parametric rewriting in the ZX-calculus, arXiv preprint arXiv:2403.06777 (2024).
[29] I. L. Markov and Y. Shi, Simulating Quantum Computation by Contracting Tensor Networks, SIAM Journal on Computing **38**, 10.1137/050644756 (2008).
[30] R. J. Lipton and R. E. Tarjan, A Separator Theorem for Planar Graphs, SIAM Journal on Applied Mathematics **36**, 177 (1979), publisher: Society for Industrial and Applied Mathematics.
[31] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis, Separators for sphere-packings and nearest neighbor graphs, Journal of the ACM **44**, 1 (1997).
[32] J. R. Gilbert, J. P. Hutchinson, and R. E. Tarjan, A separator theorem for graphs of bounded genus, Journal of Algorithms **5**, 391 (1984).
[33] N. Alon, P. Seymour, and R. Thomas, A separator theorem for graphs with an excluded minor and its applications, in *Proceedings of the twenty-second annual ACM symposium on Theory of Computing*, STOC '90 (Association for Computing Machinery, New York, NY, USA, 1990) pp. 293–299.
[34] T. B. Wahl and S. Strelchuk, Simulating quantum circuits using efficient tensor network contraction algorithms with subexponential upper bound, Physical Review Letters **131**, 180601 (2023), arXiv:2208.01498 [cond-mat, physics:hep-th, physics:quant-ph].
[35] A. M. Dalzell, A. W. Harrow, D. E. Koh, and R. L. La Placa, How many qubits are needed for quantum computational supremacy?, Quantum **4**, 264 (2020).
[36] J. Rajakumar, J. D. Watson, and Y.-K. Liu, Polynomial-time classical simulation of noisy iqp circuits with constant depth, arXiv preprint arXiv:2403.14607 (2024).
[37] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, *et al.*, Logical quantum processor based on reconfigurable atom arrays, Nature **626**, 58 (2024).
[38] J. B. Shearer, A note on the independence number of triangle-free graphs, Discrete Mathematics **46**, 83 (1983).
[39] R. Penrose, Applications of negative dimensional tensors, in *Combinatorial Mathematics and its Applications* (Academic Press, 1971) pp. 221–244.
[40] M. Backens, The ZX-calculus is complete for stabilizer quantum mechanics, New Journal of Physics **16**, 093021 (2014).
[41] K. F. Ng and Q. Wang, Completeness of the ZX-calculus for Pure Qubit Clifford+T Quantum Mechanics, arXiv:1801.07993 (2018).
[42] E. Jeandel, S. Perdrix, and R. Vilmart, A Complete Axiomatisation of the ZX-calculus for Clifford+T Quantum Mechanics, in *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science* (ACM, 2018) pp. 559–568.

Appendix: An Introduction to the ZX-Calculus

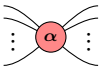
The ZX-calculus is a diagrammatic language similar to the familiar quantum circuit notation. A *ZX-diagram* (or simply *diagram*) consists of *wires* and *spiders*. Wires entering the diagram from the left are *inputs*; wires exiting to the right are *outputs*. Given two diagrams we can compose them by joining the outputs of the first to the inputs of the second, or form their tensor product by simply stacking the two diagrams.

Spiders are linear operations which can have any number of input or output wires. There are two varieties, Z spiders depicted as green dots:



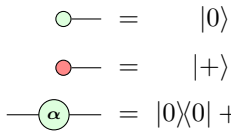
$$:= |0...0\rangle\langle 0...0| + e^{i\alpha}|1...1\rangle\langle 1...1| \tag{11}$$

and X spiders depicted as red dots:



$$:= |+...+\rangle\langle +...+| + e^{i\alpha}|-...-\rangle\langle -...-| \tag{12}$$

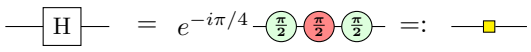
When the phase α is zero, we will omit it from the notation. The diagram as a whole corresponds to a linear map built from the spiders (and permutations) by the usual composition and tensor product of linear maps. As a special case, diagrams with no inputs represent (unnormalised) state preparations. For instance:



$$\begin{aligned} &= |0\rangle + |1\rangle &&= \sqrt{2}|+\rangle \\ &= |+\rangle + |-\rangle &&= \sqrt{2}|0\rangle \\ &= |0\rangle\langle 0| + e^{i\alpha}|1\rangle\langle 1| &&= Z_\alpha \end{aligned} \tag{13}$$

Here the last one is the Z_α phase gate.

For convenience, special notation for the Hadamard gate is used:

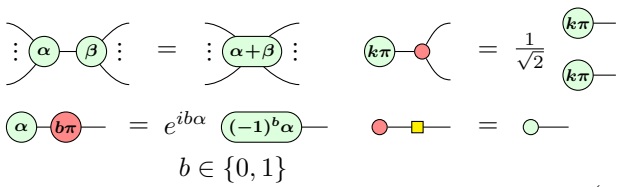


$$= e^{-i\pi/4} \quad =: \tag{14}$$

Two diagrams are considered *equal* when one can be deformed to the other by moving the vertices around in the plane, bending, unbending, crossing, and uncrossing wires, as long as the connectivity and the order of the inputs and outputs is maintained. Equivalently, a ZX-diagram can be considered as a graphical depiction of a tensor network, as in e.g. [39]. The Z- and X-spiders are symmetric tensors, and hence, like for other tensor networks of symmetric tensors, the interpretation of a ZX-diagram is unaffected by deformation.

Quantum circuits can be straightforwardly translated into ZX-diagrams. In fact, as we can also represent state preparations and post-selections, ZX-diagrams with arbitrary angles are expressive enough to represent any linear map [19]. When we restrict the angles to multiples of $\pi/2$, the maps it represents correspond to Clifford maps: linear maps that can be expressed as a combination of stabiliser state preparations, Clifford unitaries, and stabiliser post-selections [40]. Instead restricting the angles to multiples of $\pi/4$ gives us the Clifford+T *fragment*, which corresponds to those linear maps that can be constructed from Clifford+T unitaries together with state preparations and post-selections [41, 42].

In addition to this extra flexibility which allows us to represent arbitrary linear maps, the real utility for ZX-diagrams comes from the set of rewrite rules they satisfy. This set of equations is called the ZX-calculus. Diagrams that can be transformed into each other using the rules of the ZX-calculus correspond to equal linear maps. We will only need a small number of rules:



(15)

These are the *spider-fusion* rule—that adjacent spiders of the same colour fuse together (which also holds for the X-spider)—and special cases of the *colour-change* rule—that a Hadamard can be commuted through a spider to change its colour—and the π-*copy* rules—that a π phase can be commuted through the opposite colour [20].

Independence number of γ-sparse random graphs

We here restate theorem 3:

Theorem 3. *There exist a constant $C > 0$ such that with high probability*

$$\alpha\left(G\left(n, \frac{3\gamma\ln(n)}{4n}\right)\right) \geq C\frac{n\log\log(n)}{\log(n)}.$$

The idea behind this proof is to use a classic result from Shearer [38] about independent sets in triangle-free graphs.

Theorem 4 (Shearer 1983)**.** *Let G be a triangle-free graph on n points with average degree d, then*

$$\alpha(G) \geq n(d\ln(d) - d + 1)/(d-1)^2. \tag{16}$$

Even though γ-sparse random graphs aren't triangle-free, with high probability they contain rather few triangles. We utilise this fact by removing vertices from G until it is triangle-free.

Lemma 5. *With high probability $G\left(n, \frac{3\gamma\ln n}{4n}\right)$ has less than $\ln^4 n$ triangles.*

Proof. Let X be the random variable representing the number of triangles in $G\left(n, \frac{3\gamma\ln n}{4n}\right)$. Then

$$\begin{aligned} \frac{\mathbb{E}[X]}{\ln^4 n} &= \frac{1}{\ln^4 n}\binom{n}{3}\left(\frac{3\gamma\ln n}{4n}\right)^3 \\ &= \frac{O(\ln^3(n))}{\ln^4(n)} \\ &= o(1) \end{aligned}$$

Applying Markov's inequality gives us the result. □

By removing one vertex per triangle of $G\left(n, \frac{3\gamma \ln n}{4n}\right)$, we obtain a triangle free graph G'. By the lemma, G' contains with high probability more than $n - \ln^4 n$ vertices. Let us denote the number of vertices and the average degree of G' by n' and d' respectively. Assume that n is large enough that $\ln^4 n < \frac{1}{2}n$, so that $n' > \frac{1}{2}n$ with high probability. Notice that $d' \leq \frac{2|E(G)|}{n'}$. By Chernoff bound, $|E(G)| \leq \gamma n \ln(n)$ with high probability. Therefore, $d' \leq 2\gamma n \ln(n)/n' \leq 4\gamma n \ln(n)/n = 4\gamma \ln n$ with high probability. We can now use theorem 4 to prove the theorem.

Proof. (of Theorem 3) We calculate:

$$\begin{aligned}
\alpha(G) &\geq \alpha(G') \\
&\geq n' \frac{d' \ln(d') - d' + 1}{(d'-1)^2} \\
&= O\left(n' \frac{d' \ln(d')}{d'^2}\right) \\
&= O\left(n' \frac{\ln(d')}{d'}\right) \\
&\geq O\left(n' \frac{\ln(4\gamma \ln(n))}{4\gamma \ln(n)}\right) \\
&\geq O\left((n - \ln^4(n)) \frac{\ln(4\gamma \ln(n))}{4\gamma \ln(n)}\right) \\
&\geq O\left(n \frac{\ln(\ln(n))}{\ln(n)}\right)
\end{aligned}$$

Hence, $\alpha(G) \geq Cn\frac{\ln(\ln(n))}{\ln(n)}$ for some $C > 0$ (with high probability) when n is large enough. □

We note that the bounds used to derive this theorem are quite crude when n is small. For the circuit sizes we considered in our benchmarks, the independent sets were much larger than one could expect by simply looking at those asymptotic results.