A single T-gate makes distribution learning hard

M. Hinsche,[1,*] M. Ioannou,[1,*] A. Nietner,[1,*] J. Haferkamp,[1]
Y. Quek,[1] D. Hangleiter,[2] J.-P. Seifert,[3] J. Eisert,[1,4,5] and R. Sweke[1,*]

[1]*Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany*
[2]*Joint Center for Quantum Information and Computer Science (QuICS), University of Maryland & NIST, College Park, MD 20742, USA*
[3]*Department of Electrical Engineering and Computer Science, TU Berlin, 10587 Berlin, Germany*
[4]*Helmholtz-Zentrum Berlin für Materialien und Energie, 14109 Berlin, Germany*
[5]*Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany*

(Dated: July 8, 2022)

The task of learning a probability distribution from samples is ubiquitous across the natural sciences. The output distributions of local quantum circuits form a particularly interesting class of distributions, of key importance both to quantum advantage proposals and a variety of quantum machine learning algorithms. In this work, we provide an extensive characterization of the learnability of the output distributions of local quantum circuits. Our first result yields insight into the relationship between the efficient learnability and the efficient simulatability of these distributions. Specifically, we prove that the density modelling problem associated with Clifford circuits can be efficiently solved, while for depth $d = n^{\Omega(1)}$ circuits the injection of a single T-gate into the circuit renders this problem hard. This result shows that efficient simulatability does not imply efficient learnability. Our second set of results provides insight into the potential and limitations of quantum generative modelling algorithms. We first show that the generative modelling problem associated with depth $d = n^{\Omega(1)}$ local quantum circuits is hard for *any* learning algorithm, classical or quantum. As a consequence, one *cannot* use a quantum algorithm to gain a practical advantage for this task. We then show that, for a wide variety of the most practically relevant learning algorithms – including hybrid-quantum classical algorithms – even the generative modelling problem associated with depth $d = \omega(\log(n))$ Clifford circuits is hard. This result places limitations on the applicability of near-term hybrid quantum-classical generative modelling algorithms.

Deep generative models have recently empowered many impressive scientific feats, ranging from predicting protein structure to atomic accuracy [1] to achieving human-level language comprehension [2]. Consequently, there has been much interest in architecture and algorithm development for probabilistic modelling. Ideally one would like to obtain a rigorous theoretical understanding of these emerging state-of-the-art models, which requires a suitable theoretical framework. Such a framework is provided by the problem of *distribution learning*: Given samples from an unknown distribution, output some suitable representation of that distribution. Significant effort has been devoted to characterizing the complexity of learning various classes of structured distributions [3–5], including mixture models [6, 7], output distributions of restricted Boolean circuits [5, 8] and Poisson binomial distributions [9]. However, these classes of distributions are still somewhat removed from those of most interest to machine learning practitioners, such as those governing movements in the stock market, or the outputs of deep generative models.

Simultaneously, the last years have witnessed significant interest in the potential of exploiting quantum devices for machine learning tasks [10–12]. Of particular interest are hybrid quantum-classical schemes, in which parameterized quantum circuits are used as a model class, whose parameters are optimized via classical algorithms [13, 14]. In the context of generative modelling, the output distributions of quantum circuits are a particularly natural model class, referred to as *quantum circuit Born machines* (QCBMs) [15, 16]. In particular, it is known that this model class is expressive enough to contain many probabilistic graphical models [17, 18], while not being classically simulatable [19–21]. These facts, along with a growing body of numerical experiments [22–25], suggest that hybrid quantum-classical algorithms using QCBMs as a model class may offer concrete advantages over state-of-the-art classical generative modelling techniques. However, to date, there are no rigorous results on the learnability of this model class which support this intuition.

In order to address this, we provide in this letter a comprehensive study of the learnability of the output distributions of local quantum circuits – i.e., QCBMs. This allows us to resolve a variety of open questions. Firstly, we provide two hardness results for the generative modelling problem associated with these distributions. The first shows that the output distributions of n qubit quantum circuits of depth $n^{\Omega(1)}$ are not efficiently learnable by any learning algorithm with access to samples from the unknown distribution. The second shows that the output distributions of quantum circuits of depth $\omega(\log(n))$ are not efficiently learnable by algorithms which use only statistical averages with respect to the unknown distribution. Most practically relevant algorithms are indeed of this type. To date, the output distributions of local quantum circuits are considered the most promising candidate for demonstrating a rigorous complexity theoretic separation between the power of QCBM-based hybrid quantum-classical algorithms and purely classical generative modelling techniques. However, our hardness results show that this is not possible, and, therefore, place strong limitations on the advantages one might hope to achieve in this setting with near-term quantum devices.

* M. H. , M. I. , A. N. and R. S. have contributed equally.
Corresponding authors: m.hinsche@fu-berlin.de, marios.ioannou@fu-berlin.de, a.nietner@fu-berlin.de, rsweke@gmail.com

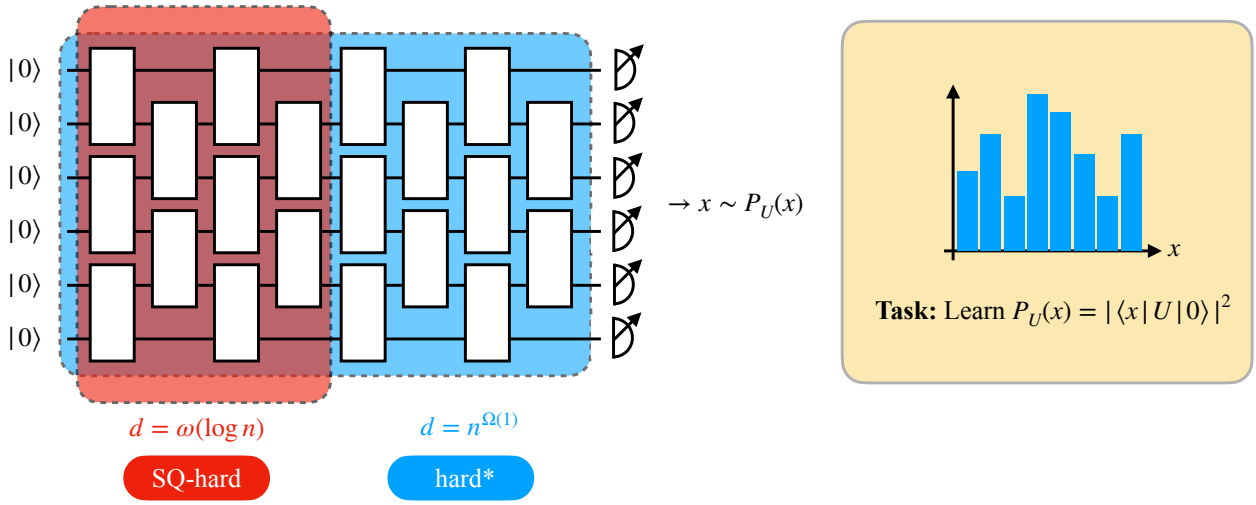

FIG. 1. How hard is the task of generator- or evaluator-learning the output distributions of local quantum circuits on n qubits of depth d? In accord with the intuition that deeper circuits generate more complex distributions, our answer depends on how d scales with n. We find that for $d = \omega(\log n)$, even the output distributions of Clifford circuits are not efficiently learnable when given *statistical query* access to P_U (Theorem 4). When given the *sample* access, the output distributions of generic local quantum circuits cease to be efficiently learnable at linear depths $d = n^{\Omega(1)}$ and beyond, up to standard cryptographic assumptions (Corollary 1 and Theorem 3).

Secondly, we show clearly that, within the context of distribution learning, classical simulatability of a class of quantum circuits *does not* imply efficient learnability. This is in strong contrast to existing conjectures and known results in other related settings [22, 26–28]. To do this, we prove that the output distributions of Clifford circuits are efficiently learnable, while the addition of a single T-gate to the circuit renders the learning problem hard. As such, while the complexity of the classical simulation scales with the number of T-gates, we find that the addition of a single T-gate induces a striking complexity transition in the corresponding distribution learning problem.

Setting. — In this Letter, we are concerned with learning distributions promised to be from a *distribution class* $\mathcal{D}$. In particular, we are interested in the properties of learning algorithms that solve the following problem [1]:

Problem 1 (Distribution learning) *Given a distribution class $\mathcal{D}$, samples from an unknown distribution $P \in \mathcal{D}$, and $\varepsilon, \delta \in (0,1)$, output with probability at least $1-\delta$, a representation of a distribution Q satisfying* $\mathrm{TV}(P,Q) \leq \varepsilon$.

We will be concerned with two types of *representations*, namely *generators* and *evaluators*:

- An evaluator for a distribution Q is a computationally efficient algorithm which, when given some x, outputs the probability $Q(x)$.
- A generator for a distribution Q is a computationally efficient algorithm for generating samples from Q.

We note that the problem of distribution learning with respect to an evaluator is often referred to as *density modelling*, while the problem of learning with respect to a generator is often referred to as *generative modelling*. Additionally, we stress that in the case of generative modelling it is *not* sufficient for the learning algorithm to store and later reproduce the samples it received during the learning phase, or to output a larger but still bounded set of samples [29]. Indeed, the learning algorithm is required to output another algorithm – a generator – which can output as many as samples as desired, from a distribution which is close in total variation distance to the unknown target distribution.

We are concerned here exclusively with discrete distributions over $\{0,1\}^n$, and denote the set of all such distributions by $\mathcal{D}_n$. Given some $\mathcal{D} \subseteq \mathcal{D}_n$, we say that an algorithm is a computationally (sample) efficient algorithm for learning $\mathcal{D}$ with respect to a particular representation (either generators or evaluators) if it solves the above problem for all $P \in \mathcal{D}$, using $O(\mathrm{poly}(n, 1/\varepsilon, 1/\delta))$ computational time (samples). If there exists a computationally efficient learning algorithm for $\mathcal{D}$ with respect to a particular representation, then we say that $\mathcal{D}$ is efficiently learnable with respect to that representation. If there does not exist a computationally efficient learning algorithm for some class $\mathcal{D}$ with respect to a particular representation, then we say that $\mathcal{D}$ is hard to learn with respect to that representation.

Our particular focus in this work is on the output distributions of quantum circuits. More specifically, to any unitary U we have the associated probability distribution P_U, with probabilities

$$P_U(x) := \left|\langle x|U|0^{\otimes n}\rangle\right|^2. \tag{1}$$

We then consider sets of distributions obtained from all unitaries generated by quantum circuits of a specific depth, with

[1] TV denotes here the total variation distance between two probability distributions, see also the appendix.

gates from a specific gate set. Unless otherwise specified, we consider one-dimensional circuits consisting only of nearest-neighbour gates, which for convenience we refer to as *local* quantum circuits. We are particularly interested in how the complexity of learning depends on both the gate set, and the circuit depth. We note that our results generalize and extend seminal work on learning the output distributions of classical circuits [5].

Learning Clifford distributions. — We start by studying the learnability of the output distributions of Clifford circuits. Our primary motivation for doing so is to better understand the relation between the complexity of classical simulation of quantum circuits and their learnability: It is well-known that by virtue of the Gottesman-Knill theorem, Clifford circuits can be efficiently classically simulated [30, 31]. Similarly, it has been found previously that the algebraic structure of the Clifford group also facilitates efficient learning of an unknown stabilizer state [32] or Clifford circuit [27] from few copies of the unknown quantum state. Furthermore, stabilizer states have been found to be efficiently PAC-learnable [26, 33] in Aaronson's framework for PAC-learning quantum states [34]. In this setting, Ref. [28] finds a sufficient condition under which the complexity of simulatibility and learnability are aligned. Here, we ask whether the alignment in the complexity of classical simulation and learning holds also in the distribution learning setting. Indeed, when studying Clifford circuits, we find that our learning model is no exception.

Theorem 1 *The set $\mathcal{D}_{\mathrm{Cl}}$ of Clifford circuit distributions, for any depth, is efficiently learnable with respect to generators and evaluators.*

Proof (sketch): Clifford circuit output distributions are uniform over affine subspaces of the finite n dimensional vector space $\mathbb{F}_2^n$. Hence using Gaussian elimination on $O(n)$ samples recovers the correct affine subspace, and from this the correct distribution representation, with success rate $1-\exp(-\Omega(n))$. □

Hardness of learning Clifford+T-distributions. — Next, we ask whether this alignment of complexity extends even to slightly non-Clifford circuits. In particular, on the simulation side, the run-time of the best-known classical algorithms for simulating T-enriched Clifford circuits will grow exponentially with the number of T gates [12, 35–37]. On the learning side, a first result for learning output states of unknown Clifford+T circuits, from copies of the unknown state, has been obtained in Ref. [27]. They also find an exponential scaling in the number of T gates provided all T gates are applied in a single layer.

Let us now return to the distribution learning setting. We consider the class of output distributions arising from T-enriched Clifford circuits. The following result relies on the *learning parities with noise* (LPN) assumption. It posits that there does not exist an efficient algorithm, quantum or classical, for learning from classical samples the class of Boolean parity functions under the uniform distribution when subject to any constant-rate random classification noise. We note that this is a canonical assumption for many cryptographic schemes [38, 39].

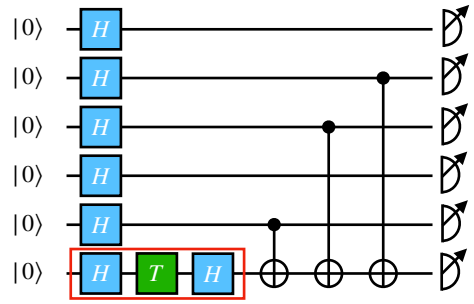

FIG. 2. Example of a circuit used in the proof of Theorem 2. Without the red box, samples from this circuit are of the form $(x, f(x))$ where x is uniformly random and f is the parity function supported on bits $2, 3, 5$. With the red box, the samples are of the form (x, y) where $y = f(x)$ with probability $1-\eta$ and $y = \neg f(x)$ with probability η.

Theorem 2 *Under the LPN assumption, the output distributions of local Clifford circuits of depth $d = n^{\Omega(1)}$ enriched with a single T-gate are not efficiently learnable with respect to an evaluator.*

Proof (sketch): Ref. [5] gives a class of distributions such that LPN reduces to evaluator-learning this class. Specifically, for each parity function, there is a corresponding distribution. Each such distribution can be realized as the output distribution of Clifford circuit enriched with a single T gate (see e.g. Fig. 2). We obtain the stated depth dependence by recompiling the circuit into local gates and using a rescaling argument to trade circuit-depth for learning complexity. □

We note that a similar hardness result based on the LPN assumption can be obtained for output distributions of Clifford circuits subject to single-qubit depolarizing noise. The key insight underlying the proof of Theorem 2 is that the LPN noise can be realized by a single T gate. Moreover, it can be seen that, if one relaxes the nearest-neighbour requirement on the Clifford gates, i.e., allowing instead for arbitrary connectivity between qubits, then one obtains the above hardness result in Theorem 2 already for depth $d = \Omega(1)$.

The sharp transition in complexity between Theorem 1 and Theorem 2 stands in interesting contrast to the smooth increase in the complexity of classically simulating T-enriched Clifford circuits: In particular, while T-enriched Clifford circuits can be simulated efficiently for up to $O(\log n)$ many T gates [37], a single T gate is enough to make distribution-learning with an evaluator at least as hard as LPN.

The class of T-enriched local Clifford circuits is a subclass of the class of all local quantum circuits. Hence, the conditional hardness result of Theorem 2 also applies to this more general class:

Corollary 1 *Under the LPN assumption, the output distributions of local quantum circuits of depth $d = n^{\Omega(1)}$ are not efficiently learnable with respect to an evaluator.*

Hardness of learning generators. — In the previous sections we have seen how adding a single T gate can make the task of learning an evaluator for Clifford distributions at least as hard

as LPN. This leaves open the question of the complexity of learning the output distributions of non-Clifford circuits with respect to a generator. As discussed in the introduction, the complexity of generator learning is interesting not only from a purely theoretical standpoint. It also allows us to gain insight into the potential of quantum generative models (QCBMs).

In Ref. [5], it has been shown that the output distributions of polynomially sized classical circuits are not efficiently classically learnable with respect to a generator. In this section, we establish an analogous result for the output distributions of quantum circuits by adapting the proof strategy of Ref. [5]. Our result applies to both quantum and classical learning algorithms. In particular, we show that one can embed *pseudo-random functions* (PRFs) into the output distributions of local quantum circuits. In order to establish hardness for quantum learning algorithms, we use "standard-secure" PRFs – i.e., PRFs secure against quantum adversaries with classical membership queries [40].

Theorem 3 *Assuming the existence of classical-secure (standard-secure) pseudorandom functions, there is no efficient classical (quantum) algorithm for learning the output distributions of depth $d = n^{\Omega(1)}$ local quantum circuits, with gates from any universal gate set.*

Proof (sketch): Instantiating the proof of Theorem 17 in Ref. [5] with a standard-secure PRF yields the following: the output distributions of polynomially sized classical circuits are not efficiently generator learnable, even by quantum learning algorithms. Polynomially sized classical circuits can be realized by polynomially sized local quantum circuits. Therefore, the output distributions of polynomially sized local quantum circuits can also not be learned efficiently with respect to a generator. This result can be extended to any universal gate set by virtue of the Solovay-Kitaev theorem. We obtain the stated depth dependence by use of a rescaling argument trading complexity for depth. □

Previous work has suggested, and provided numerical evidence, that learning a generator for quantum circuit output distributions is hard for classical learning algorithms [22–25]. Theorem 3 provides a rigorous proof for this and, interestingly, shows that these distributions are also hard to learn using quantum algorithms – including QCBM based learners. As such, one cannot hope to use the output distributions of local quantum circuits to prove a probabilistic modelling separation between QCBM based algorithms and classical algorithms.

We note that our proof technique shares similarities with that of Ref. [41], where it was shown that learning Boolean functions generated by constant depth classical circuits is hard for quantum algorithms, even with quantum examples. However classes of Boolean functions which are hard to learn cannot be generically used to create distribution classes which are hard to learn with respect to a generator [42]. As such, our results do not follow directly from theirs, despite similarities in the proof strategies.

Hardness of learning with statistical query algorithms. — In the previous sections we have established the hardness of learning the output distributions of polynomial depth circuits. However, the efficient learnability of shorter circuits remains open. In this section we show that the hardness results of the previous sections can be strengthened to hold for the output distributions of super-logarithmic depth circuits, if one considers a restricted – but practically highly relevant – class of learning algorithms.

To understand this restriction recall that in Theorem 1 we have seen an example of a distribution class – namely the output distributions of Clifford circuits – whose intrinsic algebraic structure allowed us to devise an efficient learning algorithm. In particular, this algorithm is able to exploit individual samples from the target distribution, by using the promise that the target distribution is the uniform distribution over some affine subspace of $\mathbb{F}_2^n$. However, in the absence of a strong promise on the structure of the unknown distribution to be learned, it is a-priori unclear how a learning algorithm should utilize individual samples from the target distribution. As such, most *generic* distribution learning algorithms – i.e., algorithms which are not designed specifically for one particular distribution class – work by using samples from the unknown distribution to estimate statistical averages with respect to that distribution [43]. Indeed, this is the case for almost all gradient based algorithms used in practice, both for classical neural network model classes (such as RBMs and GANs) [43] as well as quantum circuit based model classes such as QCBMs [16, 22].

In order to formally study the properties of such learning algorithms, we assume that the learning algorithm does not have access to samples from the unknown distribution P, but only to approximate statistical averages with respect to P. More specifically, we assume that the algorithm has access to a *statistical query oracle*, which when queried with some efficiently computable function $\phi : \{0,1\}^n \to [-1,1]$ returns some v such that $|\mathbb{E}_{x\sim P}[\phi(x)] - v| \leq \tau$ - i.e., an approximation of the expectation value of ϕ with respect to P, up to accuracy τ [44]. While in principle one could consider any accuracy parameter τ, we consider at most inverse polynomial accuracy – i.e., $\tau = \Omega(1/\text{poly}(n))$ – as in this regime the statistical query oracle can be efficiently simulated from samples, and query-complexity lower bounds with respect to statistical queries yield computational complexity lower bounds with respect to sample queries [45].

Theorem 4 *There is no query efficient algorithm for learning from inverse polynomially accurate statistical queries*

- *$\mathcal{D}_{\text{Cl}}$ at depth $\omega(\log(n))$,*
- *$\mathcal{D}_{\mathcal{G}}$ at depth $\omega(\log^k(n))$ where k is a constant depending on the universal gate set $\mathcal{G}$ (which can be as small as 2),*

with respect to either generators or evaluators.

Proof (sketch): As shown in Refs. [46, 47] learning parities in the statistical query model is hard. From this, one can prove that the output distributions of parity functions on uniformly random inputs are also hard to learn from statistical queries. We have already shown in the proof of Theorem 2

that the output distributions of parity functions can be realized by linear depth Clifford circuits. Combining these two facts yields the hardness result for linear depth Clifford circuits. We then obtain the first claim by applying a rescaling argument which trades circuit depth for complexity. We obtain the second claim by using robustness properties of the statistical query oracle, coupled with the Solovay-Kitaev theorem to approximate Clifford circuits. □

A first immediate consequence of the above result is that one cannot hope to use the output distributions of super-logarithmic depth local circuits to prove a practical separation between the power of classical learning algorithms and QCBM's, provided one uses previously-proposed QCBM based learning algorithms based on statistical queries [16, 22]. Additionally, Theorem 3 leaves open the possibility that there exists some efficient learning algorithm for circuits with depth less than $n^{\Omega(1)}$. However, as hardness in the statistical query model is often taken as evidence for hardness in the sample model [44], the above result provides evidence that Theorem 3 could potentially be strengthened to hold for the output distributions of super-logarithmic depth circuits. At least, any efficient learning algorithm for such circuits must utilize individual samples in a non-trivial way.

Conclusions. — In this letter, we have provided an extensive characterization of the complexity of learning the output distributions of local quantum circuits. Apart from being of fundamental interest in its own right, this characterization also contributes to our understanding of the relationship between the learnability and simulatibility of local quantum circuit output distributions.

Moreover, our results have multiple implications for the emerging field of *quantum machine learning*. In particular, a major focus of current research efforts in this direction is the identification of problems for which one can rigorously prove a separation between the power of quantum and classical learning algorithms [48]. Previous work has leveraged cryptographic assumptions to construct highly fine-tuned learning problems for which fault-tolerant quantum computers can obtain an exponential advantage [49–51]. The output distributions of quantum circuits were a primary candidate for establishing a separation for a natural learning problem. However, our work establishes that this is not possible, and, therefore, implies the need to identify new strategies for proving practically relevant quantum advantages in machine learning. In particular, our work complements existing results [52] that place limitations on the applicability of near-term hybrid quantum-classical learning algorithms, including QCBMs.

There remain many exciting questions. Firstly, are our worst-case bounds tight? In particular, can one exhibit efficient learning algorithms for the circuit depths not covered by our hardness results? Secondly, can one characterize the *sample* complexity of the learning tasks we have considered. Thirdly, in order to gain insight into the performance of heuristic learning algorithms, it is important to understand the *average-case* complexity of learning the output distributions of local quantum circuits. Additionally, it is interesting to study the learnability of other physically-motivated distributions, such as those arising from free-fermionic evolutions [53, 54]. Finally, to fully characterize the relationship between simulatability and learnability, it is of interest to understand whether hardness of simulation implies hardness of learning. In particular, are there circuit distributions which are hard to classically simulate, while being efficiently learnable?

Acknowledgments. We are thankful for excellent discussions with Matthias Caro, Hakop Pashayan and feedback of an unknown peer reviewer. This work has been funded by the Cluster of Excellence MATH+ (EF1-11), the BMWK (PlanQK), the BMBF (Hybrid, QPIC-1), the DFG (CRC183, EI 519 20-1), the QuantERA (HQCC), the Munich Quantum Valley (K8), and the Alexander von Humboldt Foundation.

[1] J. Jumper et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596:583–589, 2021. 1

[2] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, Katie Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre. Training compute-optimal large language models. *arXiv:2203.15556*, 2022. 1

[3] C. L. Canonne. A short note on learning discrete distributions. *arXiv:2002.11457*, 2020. 1

[4] I. Diakonikolas. Learning structured distributions. *Handbook of Big Data*, 267, 2016.

[5] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R. E. Schapire, and L. Sellie. On the learnability of discrete distributions. In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 273–282, New York, NY, USA, 1994. ACM. 1, 3, 4, 8, 14, 15

[6] I. Diakonikolas, D. M. Kane, and A. Stewart. Statistical query lower bounds for robust estimation of high-dimensional Gaussians and Gaussian mixtures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 73–84, 2017. 1

[7] S.-O. Chan, I. Diakonikolas, X. Sun, and R. A. Servedio. Learning mixtures of structured distributions over discrete domains. In *Proceedings of the 2013 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 1380–1394. Society for Industrial and Applied Mathematics, 2013. 1

[8] A. De, I. Diakonikolas, and R. A. Servedio. Learning from satisfying assignments. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 478–497. Society for Industrial and Applied Mathematics, 2015. 1

[9] C. Daskalakis, I. Diakonikolas, and R. A. Servedio. Learning poisson binomial distributions. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 709–728, New York, NY, USA, 2012. Association for Computing Machinery. 1

[10] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. Quantum machine learning. *Nature*, 549:195–202, 2017. 1

[11] G. Carleo, J. I. Cirac, K. Cranmer, L. Daudet, M. Schuld,

N. Tishby, L. Vogt-Maranto, and L. Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91:045002, 2019.

[12] S. Bravyi and D. Gosset. Improved classical simulation of quantum circuits dominated by Clifford gates. *Phys. Rev. Lett.*, 116:250501, 2016. 1, 3

[13] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik. Noisy intermediate-scale quantum (NISQ) algorithms. *arXiv:2101.08448*, 2021. 1

[14] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini. Parameterized quantum circuits as machine learning models. *Quant. Sc. Tech.*, 4:043001, 2019. 1

[15] M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz. A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Information*, 5, 2019. 1

[16] J.-G. Liu and L. Wang. Differentiable learning of quantum circuit Born machine. *Phys. Rev. A*, 98:062324, 2018. 1, 4, 5

[17] I. Glasser, R. Sweke, N. Pancotti, J. Eisert, and J. I. Cirac. Expressive power of tensor-network factorizations for probabilistic modeling. *Advances in Neural Information Processing Systems*, 32:1498–1510, 2019. 1

[18] Sandesh Adhikary, Siddarth Srinivasan, Jacob Miller, Guillaume Rabusseau, and Byron Boots. Quantum tensor networks, stochastic processes, and weighted automata. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2080–2088. PMLR, 13–15 Apr 2021. 1

[19] M. J. Bremner, R. Jozsa, and D. J. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proc. Roy. Soc. A*, 467:459–472, 2010. 1

[20] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven. Characterizing quantum supremacy in near-term devices. *Nature Phys.*, 14:595–600, 2018.

[21] D. Hangleiter and J. Eisert. Computational advantage of quantum random sampling. *arXiv:2206.04079*, 2022. 1

[22] B. Coyle, D. Mills, V. Danos, and E. Kashefi. The Born supremacy: Quantum advantage and training of an Ising Born machine. *npj Quant. Inf.*, 6:60, 2020. 1, 2, 4, 5

[23] K. Gili, M. Mauri, and A. Perdomo-Ortiz. Evaluating generalization in classical and quantum generative models. *arXiv:2201.08770*, 2022.

[24] M. S. Rudolph, N. Bashige Toussaint, A. Katabarwa, S. Johri, B. Peropadre, and A. Perdomo-Ortiz. Generation of high-resolution handwritten digits with an ion-trap quantum computer. *arXiv:2012.03924*, 2020.

[25] M. Y. Niu, A. M. Dai, L. Li, A. Odena, Z. Zhao, V. Smelyanskyi, H. Neven, and S. Boixo. Learnability and complexity of quantum samples. *arXiv:2010.11983*, 2020. 1, 4

[26] A. Rocchetto. Stabiliser states are efficiently PAC-learnable. *Quantum Info. Comput.*, 18:541–552, 2018. 2, 3

[27] C.-Y. Lai and H.-C. Cheng. Learning quantum circuits of some t gates. *IEEE Trans. Inf. Th.*, pages 1–1, 2022. 3

[28] Mithuna Yoganathan. A condition under which classical simulability implies efficient state learnability. *arXiv:1907.08163 [quant-ph]*, 2019. 2, 3

[29] B. Axelrod, S. Garg, V. Sharan, and G. Valiant. Sample amplification: Increasing dataset size even when learning is impossible. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 442–451. PMLR, 2020. 2

[30] D. Gottesman. The Heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006*, 1998. 3

[31] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70, 2004. 3

[32] A. Montanaro. Learning stabilizer states by Bell sampling. *arXiv:1707.04012 [quant-ph]*, 2017. 3, 13

[33] Aravind Gollakota and Daniel Liang. On the Hardness of PAC-learning Stabilizer States with Noise. *Quantum*, 6:640, 2022. 3

[34] S. Aaronson. The learnability of quantum states. *Proc. Roy. Soc. A*, 463:3089–3114, 2007. 3

[35] H. Pashayan, J. J. Wallman, and S. D. Bartlett. Estimating outcome probabilities of quantum circuits using quasiprobabilities. *Phys. Rev. Lett.*, 115:070501, 2015. 3

[36] H. Pashayan, S. D. Bartlett, and D. Gross. From estimation of quantum probabilities to simulation of quantum circuits. *Quantum*, 4:223, 2020.

[37] Sergey Bravyi, Dan Browne, Padraic Calpin, Earl Campbell, David Gosset, and Mark Howard. Simulation of quantum circuits by low-rank stabilizer decompositions. *Quantum*, 3:181, 2019. 3

[38] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56, 2009. 3

[39] K. Pietrzak. Cryptography from learning parity with noise. In *SOFSEM 2012: Theory and Practice of Computer Science*, pages 99–114, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 3

[40] M. Zhandry. How to construct quantum random functions. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 679–687. IEEE, 2012. 4, 15

[41] S. Arunachalam, A. B. Grilo, and H. Yuen. Quantum statistical query learning. *arXiv:2002.08240*, 2020. 4

[42] D. Xiao. Learning to create is as hard as learning to appreciate. In A. T. Kalai and M. Mohri, editors, *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 516–528. Omnipress, 2010. 4

[43] S. Mohamed and B. Lakshminarayanan. Learning in Implicit Generative Models. *arXiv:1610.03483 [cs, stat]*, 2017. 4

[44] V. Feldman. A General Characterization of the Statistical Query Complexity. In *Proceedings of the 2017 Conference on Learning Theory*, pages 785–830. PMLR, 2017. 4, 5

[45] I. Diakonikolas, D. M. Kane, and A. Stewart. Statistical query lower bounds for robust estimation of high-dimensional Gaussians and Gaussian mixtures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 73–84, 2017. 4

[46] M. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45:983–1006, 1998. 4, 16

[47] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing - STOC '94*, pages 253–262. ACM Press, 1994. 4, 16

[48] S. Arunachalam and R. de Wolf. Guest column: A survey of quantum learning theory. *ACM SIGACT News*, 48:41–67, 2017. 5

[49] Y. Liu, S. Arunachalam, and K. Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Phys.*, pages 1–5, 2021. 5

[50] R. Sweke, J.-P. Seifert, D. Hangleiter, and J. Eisert. On the quantum versus classical learnability of discrete distributions. *Quantum*, 5:417, 2021. 15

[51] Sofiene Jerbi, Casper Gyurik, Simon Marshall, Hans Briegel,

and Vedran Dunjko. Parametrized quantum policies for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 28362–28375. Curran Associates, Inc., 2021. 5

[52] Daniel Stilck França and Raul García-Patrón. Limitations of optimization algorithms on noisy quantum devices. *Nature Physics*, 17(11):1221–1227, November 2021. 5

[53] S. Aaronson and S. Grewal. Efficient learning of non-interacting fermion distributions. *arXiv:2102.10458*, 2021. 5

[54] S. Aaronson. Shtetl-optimized: Yet more mistakes in papers. https://www.S.aaronson.com/blog/?p=5706, 2021. Accessed: 2021-09-27. 5

[55] F. G. S. L. Brandão, W. Chemissany, N. Hunter-Jones, R. Kueng, and J. Preskill. Models of quantum complexity growth. *PRX Quantum*, 2:030316, 2021. 12

[56] C. M. Dawson and M. A. Nielsen. The Solovay-Kitaev algorithm. *arXiv:quant-ph/0505030*, 2005. 13

[57] A. W. Harrow, B. Recht, and I. L. Chuang. Efficient discrete approximations of quantum gates. *Journal of Mathematical Physics*, 43:4445–4451, 2002. 13, 17

[58] J. Dehaene and B. De Moor. The Clifford group, stabilizer states, and linear and quadratic operations over GF(2). *Phys. Rev. A*, 68, 2003. 13

[59] Paulo J. S. G. Ferreira, Bruno Jesus, Jose Vieira, and Armando J. Pinho. The rank of random binary matrices and distributed storage applications. *IEEE Communications Letters*, 17(1):151–154, 2013. 13

[60] D. Helmbold, R. Sloan, and M. K. Warmuth. Learning integer lattices. *SIAM J. Comp.*, 21:240–266, 1992. 13

[61] P. Auer and N. Cesa-Bianchi. On-line learning with malicious noise and the closure algorithm. In *Algorithmic Learning Theory*, volume 872, pages 229–247. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994. 13

[62] A. Blum, A. T. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50:506–519, 2003. 13

[63] S. Bravyi and D. Maslov. Hadamard-free circuits expose the structure of the clifford group. *IEEE Trans. Inf. Th.*, 67:4546–4563, 2021. 15, 16

[64] A. Bogdanov and A. Rosen. Pseudorandom functions: Three decades later. In *Tutorials on the Foundations of Cryptography*, pages 79–158. Springer, 2017. 15

[65] O. Goldreich. *Foundations of cryptography: volume 2, basic applications*. Cambridge University Press, 2009. 15

[66] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011. 15

Appendix A: Preliminaries

We start by giving formal definitions of the objects and problems considered in this work. Throughout we denote by $\mathcal{F}_n$ the set of Boolean functions from $\{0,1\}^n$ to $\{0,1\}$, by $\mathcal{D}_n$ the set of probability distributions over $\{0,1\}^n$. A subset $\mathcal{D} \subset \mathcal{D}_n$ is referred to as a distribution class. For two discrete probability distributions $P, Q : \{0,1\}^n \to [0,1]$, we denote by $\mathrm{TV}(P,Q) := \frac{1}{2}\sum_{x\in\{0,1\}^n} |P(x) - Q(x)|$ the total variation distance between them. The trace distance of two quantum states ρ and σ is given by $\frac{1}{2}\|\rho - \sigma\|_{\mathrm{tr}}$, where $\|\cdot\|_{\mathrm{tr}}$ denotes the trace norm. Access to distributions is formalized by assuming access to some oracle that has a specific operational structure. In particular, we use the sample and the statistical query oracle which are defined as follows.

Definition 2 (Distribution oracles) *Given $P \in \mathcal{D}_n$, some $\tau \in (0,1)$, we define:*

1. *The sample oracle $\mathtt{Samp}(P)$ as the oracle which, when queried, provides a sample $x \sim P$.*

2. *The statistical query oracle $\mathtt{Stat}_\tau(P)$ as the oracle which, when queried with a function $\phi : \{0,1\}^n \to [-1,1]$, responds with some v such that $|\mathbf{E}_{x\sim P}[\phi(x)] - v| \leq \tau$.*

Let us next define generators and evaluators, the central objects of this work, whose learnability we study. Informally, a generator for a given distribution P is an algorithm that generates samples from P. Likewise, an evaluator for P is an algorithm that computes $P(x)$ for all x in the support of P. More precisely:

Definition 3 (Generators) *Given some probability distribution $P \in \mathcal{D}_n$, we say that a probabilistic (or quantum) algorithm $\mathtt{Gen}_P$ is a generator for P if $\mathtt{Gen}_P$ produces samples according to P.*

Definition 4 (Evaluators) *Given some probability distribution $P \in \mathcal{D}_n$, we say that an algorithm $\mathtt{Eval}_P : \{0,1\}^n \to [0,1]$ is an evaluator for $P \in \mathcal{D}_n$ if on input $x \in \{0,1\}^n$ the algorithm outputs $\mathtt{Eval}_P(x) = P(x)$.*

We are interested in learning the output distributions of quantum circuits. To formalize this, we use the framework for *learning a distribution* as introduced in Ref. [5]. This definition is analogous to the definition of probably-approximately correct (PAC) *function* learning, in that it introduces parameters ε and δ to quantify approximation error and probability of successful approximation, respectively.

Problem 2 ((ε,δ)-distribution-learning) *Let $\varepsilon, \delta \in (0,1)$ and let $\mathcal{D}$ be a distribution class. Let $\mathcal{O}$ be a distribution oracle. The following task is called (ε,δ)-distribution-learning $\mathcal{D}$ from $\mathcal{O}$ with respect to a generator (evaluator): Given access to oracle $\mathcal{O}(P)$ for any unknown $P \in \mathcal{D}$, output with probability at least $1-\delta$ an efficient generator (evaluator) of a distribution Q such that $\mathrm{TV}(P,Q) < \varepsilon$.*

Definition 5 (Efficiently learnable distribution classes) *Let $\mathcal{D}$ be a distribution class, and let $\mathcal{O}$ be a distribution oracle. We say that $\mathcal{D}$ is computationally (query) efficiently learnable from $\mathcal{O}$ with respect to a generator/evaluator, if there exists an algorithm $\mathcal{A}$ which for all $(\varepsilon,\delta) \in (0,1)$ solves the problem of (ε,δ)-distribution learning $\mathcal{D}$ from $\mathcal{O}$ with respect to a generator/evaluator, using $O(\mathrm{poly}(n, 1/\varepsilon, 1/\delta))$ computational steps (oracle queries).*

As we are most often concerned with computational efficiency and with the sample oracle, we often omit these qualifiers in this case, and simply say "$\mathcal{D}$ is efficiently learnable". If a distribution class is not efficiently learnable, then we say it is hard to learn.

We are particularly interested in distribution classes induced by quantum circuit classes by measuring each corresponding quantum circuit in the computational basis. We denote such classes in the following fashion:

Definition 6 ($\mathcal{D}_{\mathcal{G}}(n,d)$) *Let $\mathcal{G}$ be a gate set and let $n, d \in \mathbb{N}$. We denote by $\mathcal{D}_{\mathcal{G}}(n,d)$ the set of output distributions of n-qubit nearest neighbor quantum circuits with gates from the gate set $\mathcal{G}$ at depth d. In particular, $\mathcal{D}_{\mathcal{G}}(n,d)$ contains those distributions $P \in \mathcal{D}_n$ that can be written as*

$$P(x) = |\langle x|U|0^n\rangle|^2 , \tag{A1}$$

where U can be written as a depth d nearest neighbor quantum circuit in one dimension on n qubits composed of gates from $\mathcal{G}$.

Appendix B: Useful reductions

In this section we provide a variety of lemmata, used in the proofs of our main theorems. We start with an embedding lemma which, at a high level, allows us to trade circuit depth for computational complexity of learning. More specifically, this lemma allows us to take a lower bound for learning the output distributions of a class of quantum circuits of a given depth, and obtain a new *smaller* lower bound for learning *shorter* quantum circuits. This allows us to take existing lower bounds for some class of circuits, and identify the shortest circuit depth which admits a super-polynomial lower bound. The intuition behind this lemma is illustrated in Fig. 3, and is as follows: Assume learning the output distributions of a given class of quantum circuits takes at least a certain number of computational steps (or oracle queries). Now consider the class of circuits one obtains by embedding the original circuits into wider circuits, which act trivially on the extra qubits. Intuitively, learning the output distributions of the wider quantum circuits should take at least the same number of steps (oracle queries) as for the original circuits. However, as a function of the number of qubits, both the depth of the wider quantum circuits, and the computational time (number of oracle queries) required for learning their output distributions, is reduced. We formalize this below:

Lemma 7 (Embedding reduction) *Let $n \in \mathbb{N}$, $\varepsilon, \delta \in (0,1)$ and let $\tau(n) > 0$ be a function depending on n. Let $f, g : \mathbb{N} \to \mathbb{N}$ be functions where f is monotonous and g is strictly monotonous with $n \leq g(n)$. We call g the stretch. Assume (ε, δ)-learning $\mathcal{D}_{\mathcal{G}}(n, f(n))$*

- *with respect to a generator from samples requires at least time $t(n, \varepsilon, \delta)$, and $g = O(\mathrm{poly}(n))$, or*
- *with respect to any representation requires at least $q(n, \varepsilon, \delta)$ statistical queries with tolerance $\tau(n)$.*

Then it requires at least time $t(g^{-1}(n), \varepsilon, \delta) - O(\mathrm{poly}(n))$ (respectively $q(g^{-1}(n), \varepsilon, \delta)$ statistical queries with tolerance at least $\tau \circ g^{-1}(n)$) to (ε, δ)-learn $\mathcal{D}_{\mathcal{G}}(n, f \circ g^{-1}(n))$ with respect to the corresponding representation.

Proof: To begin, we consider the first claim. Let $\mathcal{A}$ be an algorithm that (ε, δ)-learns $\mathcal{D}_{\mathcal{G}}(n, f \circ g^{-1}(n))$ from samples with respect to a generator in time $t^*(n, \varepsilon, \delta)$. We now define an algorithm $\mathcal{B}$ that makes use of $\mathcal{A}$ as a subroutine to (ε, δ)-learn $\mathcal{D}_{\mathcal{G}}(n, f(n))$ from samples with respect to a generator. As we will show, its runtime is bounded by $t^*(g(n), \varepsilon, \delta) + O(\mathrm{poly}(n))$.

Let $k \in \mathbb{N}$, denote $n = g(k)$ and let $P \in \mathcal{D}_{\mathcal{G}}(k, f(k))$ be a distribution to which we are given sample access via $\mathtt{Samp}(P)$. We define algorithm $\mathcal{B}$ as follows: $\mathcal{B}$ first emulates a sample oracle $\mathtt{Samp}(Q)$ to a distribution $Q \in \mathcal{D}_{\mathcal{G}}(n, f \circ g^{-1}(n))$ defined as

$$Q(x_1, \ldots, x_k, x_{k+1}, \ldots, x_n) = \begin{cases} P(x_1, \ldots, x_k)\,, \text{ if } x_{k+1} = \cdots = x_n = 0 \\ 0\,, \text{ else} \end{cases} \tag{B1}$$

by appending $n - k$ zeros to any bit string $(x_1, \ldots, x_k)$ output by $\mathtt{Samp}(P)$. Then $\mathcal{B}$ invokes $\mathcal{A}$ with access to $\mathtt{Samp}(Q)$ which returns a generator $\mathtt{Gen}_{Q'}$ for a $Q' \in \mathcal{D}_{\mathcal{G}}(n, f \circ g^{-1}(n))$. $\mathcal{B}$ then returns the generator $\mathtt{Gen}_{P'}$ which is defined as follows: Run $\mathtt{Gen}_{Q'}$ and receive a sample $(x_1, \ldots, x_n)$. Return $(x_1, \ldots, x_k)$ discarding the remaining $n - k$ bits.

Let us now analyze the correctness of $\mathcal{B}$: By the tensorial structure of quantum circuits, $\mathtt{Samp}(Q)$ is a valid sample oracle to some $Q \in \mathcal{D}_{\mathcal{G}}(n, f \circ g^{-1}(n)))$. Therefore, $\mathcal{A}$ will with probability at least $1 - \delta$ return a generator $\mathtt{Gen}(Q')$, efficient in n, to some Q' that is at least $1 - \varepsilon$ close to Q in TV-distance. Now we observe that $\mathtt{Gen}_{P'}$ is a generator for the marginal distribution P' of Q' on the first k bits. Hence, assuming that Q' is a correct ε-approximation to Q, by the contractivity of the TV-distance, we find that P' is a valid ε-approximation to P. Moreover, since $g(k) = O(\mathrm{poly}(k))$ by assumption, we find that $\mathtt{Gen}_{P'}$ is also efficient in k. Hence, with probability $1 - \delta$ our algorithm $\mathcal{B}$ will find an efficient generator for a distribution that is ε close to the original distribution P, thus proving the correctness.

We now observe that all steps in the reduction can be implemented with an at most polynomial overhead. Hence, learning $\mathcal{D}_{\mathcal{G}}(k, f(k))$ takes time at most $t^*(n, \varepsilon, \delta) + O(\mathrm{poly}(n)) = t^*(g(k), \varepsilon, \delta) + O(\mathrm{poly}(k))$, proving the first claim.

The second claim follows from a similar reasoning replacing computational time with oracle queries. Since the claim is in terms of the query complexity and as such inherently information theoretic, we do not need to impose the stretch g to be polynomial. Similarly, as the reduction itself does not make any statistical queries we will get the direct mapping of the query complexity $q(n, \varepsilon, \delta) \mapsto q(g^{-1}(n), \varepsilon, \delta)$ when applying $\mathcal{A}$ as a subroutine. Moreover, due to the information theoretic nature of the statement it applies to both generators and evaluators. In particular, it suffices to show the claim for generators, as we can, at least in a computationally inefficient way, obtain the corresponding evaluators without additional statistical queries.

This means, we only need to adapt the oracle emulation: Assume $\phi : \{0,1\}^n \to [-1, 1]$ to be some function queried by $\mathcal{A}$ and let P and Q be as before. To emulate $\mathtt{Stat}_{\tau \circ g^{-1}(n)}(Q)$ when queried with ϕ we query $\mathtt{Stat}_{\tau(k)}(P)$ with θ and return the corresponding value, where

$$\theta(x_1, \ldots, x_k) = \phi(x_1, \ldots, x_k, 0, \ldots, 0)\,. \tag{B2}$$

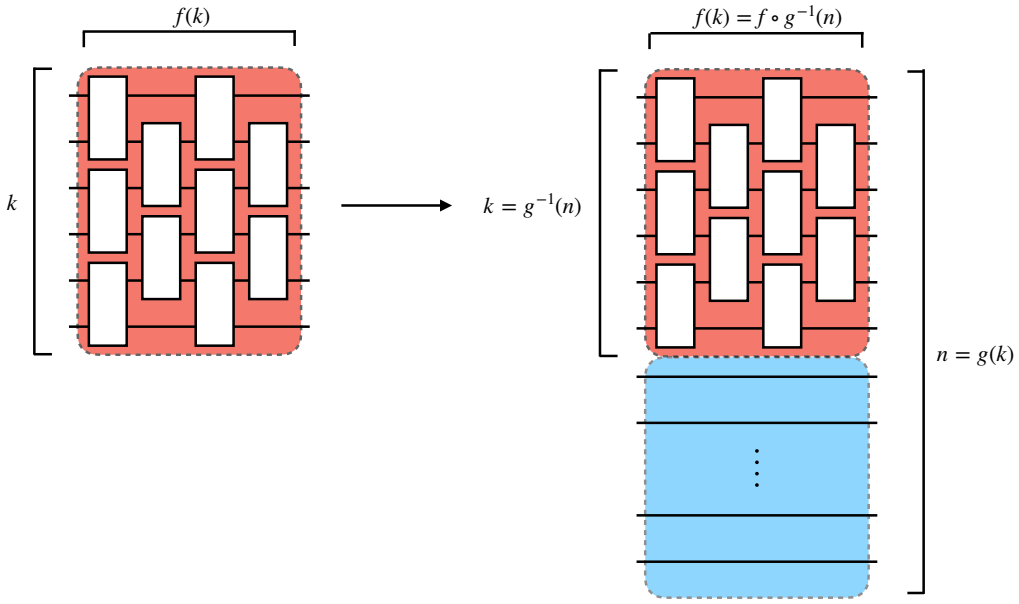

FIG. 3. Illustration of the embedding reduction used in the proof of Lemma 7. Given a class of circuits on k qubits, we can define a new class of circuits on $n = g(k)$ qubits by embedding the original circuits onto the first k qubits. Intuitively, the number of computational steps (oracle queries) required to learn the output distributions of the wider circuits, should be at least as many as that required for the original circuits. However, as a function of the number of qubits in the wider circuits, both the depth and the learning complexity are reduced by the inverse of the "stretch-factor" g.

We complete the proof by noting that

$$\mathop{\mathbf{E}}_{x_1,\ldots,x_k\sim Q}[\phi(x_1,\ldots,x_n)] = \mathop{\mathbf{E}}_{x_1,\ldots,x_k\sim P}[\phi(x_1,\ldots,x_k,0,\ldots,0)] = \mathop{\mathbf{E}}_{x_1,\ldots,x_k\sim P}[\theta(x_1,\ldots,x_k)]\,, \tag{B3}$$

and $\tau \circ g^{-1}(n) = \tau(k)$ such that this prescription is indeed a valid emulation. The correctness proof is identical to that of the first claim. □

The same trade-off of depth for complexity also applies to learning with respect to evaluators. In particular, in the special case of $\varepsilon = 0$, we immediately obtain the following corollary.

Corollary 8 *Let n, δ, g, f as before, $g = O(\mathrm{poly}(n))$ and assume that $(0,\delta)$-learning $\mathcal{D}_{\mathcal{G}}(n, f(n))$ from samples with respect to an evaluator requires at least time $t(n,\delta)$. Then it requires at least time $t(g^{-1}(n),\delta) - O(\mathrm{poly}(n))$ to $(0,\delta)$-learn $\mathcal{D}_{\mathcal{G}}(n, f\circ g^{-1}(n))$ with respect to an evaluator.*

Proof: The proof is identical to the first part of the proof of Lemma 7 only that the output of $\mathcal{A}$ is, with probability $1-\delta$, the evaluator of Q. Note, as $\varepsilon = 0$ it holds that $Q' = Q$ and $P' = P$. In order to transform $\mathtt{Eval}_Q$ to the evaluator of the original P we simply map

$$\mathtt{Eval}_P(x_1,\ldots,x_k) = \mathtt{Eval}_Q(x_1,\ldots,x_k,0,\ldots,0)\,. \tag{B4}$$

The correctness follows from the correctness of $\mathcal{A}$ together with $\varepsilon = 0$. □

In principle, the proof above also works in the case of non-zero ε. However, the output of the algorithm will in general not be an evaluator in the exact sense of Definition 4. This is because the mapping in Eq. (B4) does not preserve the normalization of the probability distribution. For practical purposes, however, one can just relax the definition of an evaluator to also apply to non-negative vectors instead of normalized probability distributions and replacing the TV distance by the ℓ_1-norm. Then, the above proof goes through for $\varepsilon \neq 0$.

We have stated Lemma 7 in its most general form as we believe that it might be of use on its own. In order to give a concrete example, we provide a corollary that will also be of use in the proof of Theorem 2.

Corollary 9 *Let $n > 0$ and assume $d = O(\mathrm{poly}(n))$. If there is no efficient algorithm for learning $\mathcal{D}_{\mathcal{G}}(n, d(n))$ with respect to a generator, then there is no efficient algorithm for learning $\mathcal{D}_{\mathcal{G}}(n, d'(n))$ with respect to a generator for any $d' = n^{\Omega(1)}$.*

Proof: Let $r \in \mathbb{N}$ be such that $d(n) = O(n^r)$. Then, via Lemma 7 with $g(n) = n^{rs}$ for some $s \in \mathbb{N}$, we find that there is no efficient algorithm for learning $\mathcal{D}_{\mathcal{G}}(n, d'(n))$ with $d'(n) = \Omega(n^{1/s})$. The claim then follows since $s \in \mathbb{N}$ is arbitrary. □

Next we will clarify in which way hardness results for one learning problem can be leveraged to obtain hardness results for a different distribution class which only approximates the former. This is a crucial tool for lifting lower bounds for circuits with some specific gate set, to generic universal quantum circuits, since the latter are known to efficiently approximate the former due to the Solovay-Kitaev theorem. Let us start by introducing some notation.

Definition 10 *Let $\mathcal{D}$ and $\mathcal{H}$ be distribution classes over the domain X and let $\sigma \in [0, 1)$. We say $\mathcal{D}$ is σ-approximately contained in $\mathcal{H}$ (with respect to the total variation distance), and write*

$$\mathcal{D} \subseteq_\sigma \mathcal{H}\,, \tag{B5}$$

if for every $P \in \mathcal{D}$ it exists a $Q \in \mathcal{H}$ such that $\mathrm{TV}(P, Q) \leq \sigma$.

Given this we find the following reduction from the learnability of a class $\mathcal{H}$ to the learnability of a approximately contained class $\mathcal{D}$. Alternatively, this implies that a harndess result on $\mathcal{D}$ translates to a corresponding hardness result on $\mathcal{H}$.

Lemma 11 (Approximation reduction) *Let $\mathcal{D}$, $\mathcal{H}$ and σ as before and $\mathcal{D} \subseteq_\sigma \mathcal{H}$. Assume that $\mathcal{H}$ is (ε, δ)-learnable from s samples with respect to a representation. Then $\mathcal{D}$ is $(\varepsilon + \sigma, \delta + s\sigma)$-learnable from s samples with respect to the same representation.*

Proof: Let $\mathcal{A}$ be an algorithm that (ε, δ)-learns $\mathcal{H}$ with respect to a representation from s samples. Then, applying $\mathcal{A}$ to $\mathcal{D}$ directly yields an $(\varepsilon+\sigma, \delta+s\sigma)$-learner with respect to the same representation. We first show this assuming $\mathcal{A}$ to be deterministic before switching to the general case. Assume $\mathcal{A}$ to be deterministic. For any $P \in \mathcal{H}$ define the event

$$\mathcal{E}(P, \varepsilon, \mathcal{A}) := \{(x_1, \ldots, x_s) \mid \mathrm{TV}(\mathcal{A}(x_1, \ldots, x_s), P) < \varepsilon\} \subseteq X^s\,. \tag{B6}$$

We interpret the characteristic function $\mathcal{E}(x_1, \ldots, x_s) := \mathbb{1}_{\mathcal{E}(P,\varepsilon,\mathcal{A})}(x_1, \ldots, x_s)$ as a random variable with respect to a distribution over X^s.

Since $\mathcal{D} \subseteq_\sigma \mathcal{H}$ we know that for every $Q \in \mathcal{D}$ there exists a $P \in \mathcal{H}$ such that $\mathrm{TV}(P, Q) \leq \sigma$. This implies

$$\mathrm{TV}(P^{\otimes s}, Q^{\otimes s}) \leq s\sigma\,. \tag{B7}$$

Therefore, it must hold

$$\mathop{\mathbf{Pr}}_{\mathcal{A}}\Big[\mathrm{TV}\Big(\mathcal{A}^Q, P\Big) < \varepsilon\Big] = \mathop{\mathbf{Pr}}_{(x_1,\ldots,x_s)\sim Q^{\otimes s}}[\mathcal{E}] \geq \mathop{\mathbf{Pr}}_{(x_1,\ldots,x_s)\sim P^{\otimes s}}[\mathcal{E}] - s\sigma = \mathop{\mathbf{Pr}}_{\mathcal{A}}\Big[\mathrm{TV}\Big(\mathcal{A}^P, P\Big) < \varepsilon\Big] - s\sigma = 1 - \delta - s\sigma\,, \tag{B8}$$

where $\mathcal{A}^Q$ ($\mathcal{A}^P$) is short hand notation for the output of the algorithm $\mathcal{A}$ with oracle access to $\mathtt{Samp}(Q)$ ($\mathtt{Samp}(P)$). The inequality is due to Eq. (B7) and the variational characterization of the TV-distance. Hence, running $\mathcal{A}$ on any $Q \in \mathcal{D}$ will, with probability $1 - (\delta + s\sigma)$ return a representation of some P' with

$$\mathrm{TV}(P', Q) \leq \mathrm{TV}(P', P) + \mathrm{TV}(P, Q) \leq \varepsilon + \sigma\,, \tag{B9}$$

proving the deterministic case.

Now assume $\mathcal{A}$ to be a random algorithm. Thus, the randomness in the second part of Eq. (B8), $(x_1, \ldots, x_s) \sim Q^{\otimes s}$, gets replaced by $(x_1, \ldots, x_s, r_1, \ldots, r_k) \sim Q^{\otimes s} \otimes D$ (and similarly for P) where D is the distribution on k bits induced by derandomizing $\mathcal{A}$. The claim then follows from the same argument replacing Eq. (B7) by

$$\mathrm{TV}(Q^{\otimes s} \otimes D, P^{\otimes s} \otimes D) \leq s\sigma\,, \tag{B10}$$

which follows from Eq. (B7) and the factorization of the ℓ_1-norm.

The argument is the same in spirit for quantum algorithms, only the formulation of the 'derandomization' procedure changes due to the quantum nature of the algorithm. Recall, that any quantum algorithm $\mathcal{A}$ which makes s queries to $\mathtt{Samp}(Q)$ can be written as a quantum circuit acting on a suitable input density matrix encoding the s queries to the oracle and the internal quantum resources of the algorithm in terms of an auxiliary density matrix. We, therefore, replace $Q^{\otimes s} \otimes D$ from the previous reasoning by $\rho_Q^{\otimes s} \otimes \rho_{\mathcal{A}}$ (and likewise for P) where $\rho_Q = \sum_{i\in X} Q(i)|i\rangle\langle i|$ is the diagonal mixed state corresponding to Q and $\rho_{\mathcal{A}}$ is the density matrix corresponding to $\mathcal{A}$'s auxiliary space. Using the factorization of the trace norm and the fact that both ρ_Q and ρ_P are diagonal in the computational basis we find

$$\frac{1}{2}\left\|\rho_Q^{\otimes s} \otimes \rho_{\mathcal{A}} - \rho_P^{\otimes s} \otimes \rho_{\mathcal{A}}\right\|_{\mathrm{tr}} = \frac{1}{2}\left\|\rho_Q^{\otimes s} - \rho_P^{\otimes s}\right\|_{\mathrm{tr}} = \mathrm{TV}(P^{\otimes s}, Q^{\otimes s}) \leq s\sigma\,. \tag{B11}$$

Hence, the claim follows from

$$\mathop{\mathbf{Pr}}_{\mathcal{A}}[\mathrm{TV}(\mathcal{A}^Q, P) < \varepsilon] = \mathrm{Tr}\Big[\Pi \cdot \rho_Q^{\otimes s} \otimes \rho_{\mathcal{A}}\Big] \leq \mathrm{Tr}\Big[\Pi \cdot \rho_P^{\otimes s} \otimes \rho_{\mathcal{A}}\Big] - s\sigma = \mathop{\mathbf{Pr}}_{\mathcal{A}}[\mathrm{TV}(\mathcal{A}^P, P) < \varepsilon] - s\sigma\,, \tag{B12}$$

where Π is the POVM encoding the application of the quantum algorithm $\mathcal{A}$ to the input $\rho^{\otimes s} \otimes \rho_{\mathcal{A}}$ and then projecting onto the valid solutions. □

For the proof of Theorem 4 we will need a slightly adjusted version of Lemma 11 which applies to the setting of statistical query learning.

Lemma 12 (Statistical query approximation reduction) *Let $\mathcal{D}, \mathcal{H}$ be distribution classes over the same domain X , let $\tau > \sigma > 0$ and let $0 < \gamma < \tau - \sigma$. Assume $\mathcal{D} \subseteq_\sigma \mathcal{H}$ and that $\mathcal{H}$ is (ε, δ)-learnable from q statistical queries with tolerance τ. Then $\mathcal{D}$ is $(\varepsilon + \sigma, \delta)$-learnable from at most q statistical queries with tolerance γ.*

Proof: The proof idea is similar to that of Lemma 11 though, by properties of statistical query learning, is technically much simpler. To begin with assume $\mathcal{A}$ to be an algorithm that (ε, δ)-learns $\mathcal{H}$ with q statistical queries of tolerance τ. Then, applying $\mathcal{A}$ to $\mathcal{D}$ directly yields an $(\varepsilon + \sigma, \delta)$-learner for $\mathcal{D}$ which uses at most q statistical queries of tolerance $\gamma \leq \tau - \sigma$. To see this, we first note that by assumption, for any $P \in \mathcal{D}$ there exists a $Q \in \mathcal{H}$ such that $\mathrm{TV}(P, Q) < \sigma$. By the variational characterization of the total variation distance and the triangle inequlity we hence find that, for any $v \in [-\gamma, \gamma]$

$$\left|\mathop{\mathbf{E}}_{x\sim P}[\phi(x)] + v - \mathop{\mathbf{E}}_{x\sim Q}[\phi(x)]\right| \leq \left|\mathop{\mathbf{E}}_{x\sim P}[\phi(x)] - \mathop{\mathbf{E}}_{x\sim Q}[\phi(x)]\right| + |v| < \sigma + \tau - \sigma = \tau\,. \tag{B13}$$

Thus, any oracle $\mathtt{Stat}_\gamma(P)$ can be interpreted as a $\mathtt{Stat}_\tau(Q)$ oracle. This implies, that when run with access to $\mathtt{Stat}_\gamma(P)$ algorithm $\mathcal{A}$ will, with probability at least $1 - \delta$ return a representation for some distribution D that is ε close to Q in total variation distance. By the triangle inequality this is at most $\varepsilon + \sigma$ far from P completing the proof. □

As we are exclusively concerned here with distribution classes associated with local quantum circuits, the following additional standard results will be useful to us to quantify the extent to which the output distributions of one class of quantum circuits can be approximated by the output distributions of another class of quantum circuits.

Lemma 13 *Let $\rho = |\psi\rangle\langle\psi|$ and $\sigma = |\phi\rangle\langle\phi|$ be pure quantum states. Then it holds*

$$\|\rho - \sigma\|_{\mathrm{tr}} = 2\sqrt{1 - |\langle\psi|\phi\rangle|^2}\,. \tag{B14}$$

Proof: (From the proof of Theorem 10 in Ref. [55].) Denote $X = \rho - \sigma$. Then X is self-adjoint and $\mathrm{tr}[X] = 0$. Hence X has eigenvalues λ and $-\lambda$. Moreover $\mathrm{tr}\left[X^2\right] = 2\lambda^2 = 2(1 - |\langle\psi|\phi\rangle|^2)$. Hence, $\lambda = \sqrt{1 - |\langle\psi|\phi\rangle|^2}$ and the claim follows from $\|X\|_{\mathrm{tr}} = 2|\lambda|$. □

Lemma 14 *Let $n \in \mathbb{N}$. Let U and W be unitary circuits on n qubits and let P and Q be the Born distributions corresponding to $U|0^n\rangle$ and respectively $W|0^n\rangle$. Assume $\|U - W\|_{op} < \varepsilon$. Then it holds $\mathrm{TV}(P, Q) < \varepsilon$.*

Proof: Denote by $\rho = U|0^n\rangle\langle 0^n|U^\dagger$, $\sigma = Q|0^n\rangle\langle 0^n|Q^\dagger$ and for any $M \subseteq \{0,1\}^n$ let $\Pi_M = \sum_{i\in M}|i\rangle\langle i|$. Then by the variational characterization of the trace- and total variation distances it holds

$$\mathrm{TV}(P, Q) = \sup_M |P(M) - Q(M)| = \sup_M \left|\mathrm{tr}[\rho\Pi_M] - \mathrm{tr}[\sigma\Pi_M]\right| \leq \frac{1}{2}\|\rho - \sigma\|_{\mathrm{tr}}\,. \tag{B15}$$

To estimate the last expression we write

$$\||U|0^n\rangle - Q|0^n\rangle\|_2 = \sqrt{2 - 2\mathrm{Re}\Big(\langle 0^n|Q^\dagger U|0^n\rangle\Big)} < \varepsilon\,, \tag{B16}$$

such that

$$\sqrt{2 - 2\left|\langle 0^n|Q^\dagger U|0^n\rangle\right|} \leq \sqrt{2 - 2\mathrm{Re}\Big(\langle 0^n|Q^\dagger U|0^n\rangle\Big)} < \varepsilon\,. \tag{B17}$$

We can now combine this with Lemma 13 to obtain

$$\|\rho - \sigma\|_{\mathrm{tr}} = 2\sqrt{1 - \left|\langle 0^n|Q^\dagger U|0^n\rangle\right|^2} < 2\sqrt{\varepsilon^2 - \varepsilon^4/4} \leq 2\varepsilon\,, \tag{B18}$$

and hence $\mathrm{TV}(P, Q) < \varepsilon$ □

Corollary 15 (Solovay-Kitaev reduction) *Let $n, d \in \mathbb{N}$, let $\varepsilon > 0$ and let $\mathcal{G}$ be a universal gate set. Then there exists a constant c such that*

$$\mathcal{D}_{\mathrm{U}(4)}(n,d) \subseteq_{\varepsilon} \mathcal{D}_{\mathcal{G}}(n,d') \quad \textit{with} \quad d' = d \cdot \log^{c}\left(\frac{n \cdot d}{\varepsilon}\right). \tag{B19}$$

Proof: By the Solovay-Kitaev theorem [56] there exists for any depth d circuit U with at most $n \cdot d$ gates a depth d' circuit Q consisting of gates from $\mathcal{G}$ that approximates U in operator norm $\|U - Q\|_{\mathrm{op}} < \varepsilon$. Hence, applying Lemma 14 yields the claim. □

Note that there exist universal gate sets for which $c = 1$ in the statement of Corollary 15 [57].

Appendix C: Proof of Theorem 1

As the proof of Theorem 1 will be based on the algebraic structure of Clifford circuits let us review the following properties first.

Definition 16 (Affine subspace) *An affine subspace $A \subseteq \mathbb{F}_2^n$ is a set such that for every $a, b, c \in A$ and $\lambda \in \mathbb{F}_2$ it holds*

$$a + (b - a) + \lambda \cdot (c - a) \in A\,, \tag{C1}$$

where all operations are with respect to $\mathbb{F}_2^n$.

In other words, for every $a \in A$ the set $A - a$ forms a linear subspace L and A is the set resulting from shifting L by a. This is, there exists an integer $m \leq n$ such that for any $t \in A$ there exists a full-rank matrix $\mathbf{R} \in \mathbb{F}_2^{m \times n}$, such that

$$A = \{\mathbf{R}b + t \,|\, b \in \mathbb{F}_2^m\}\,. \tag{C2}$$

We say A has dimension m. The choice of $\mathbf{R}$ is not unique.

The output states of Clifford circuits are called stabilizer states. As shown in Refs. [32, 58], up to a global phase, all n-qubit stabilizer state vectors $|\psi\rangle$ can be written in the computational basis as

$$|\psi\rangle = \frac{1}{\sqrt{|A|}} \sum_{x \in A} (-i)^{l(x)} (-1)^{q(x)} |x\rangle, \tag{C3}$$

where A is some affine subspace of $\mathbb{F}_2^n$ and l, q are linear and quadratic functions on $\mathbb{F}_2^n$, respectively. Thus, we find the following corollary.

Corollary 17 *For any $P \in \mathcal{D}_{\mathrm{Cl}}$ there exists an affine subspace $A \subseteq \mathbb{F}_2^n$ such that $P = U_A$, where U_A is the uniform distribution on A*

$$U_A(x) = \begin{cases} 2^{-d}\,, & d = \dim(A),\ x \in A \\ 0\,, & \textit{else.} \end{cases} \tag{C4}$$

For the proof of Theorem 1, we can make use of the following fact (c.f. Ref. [59]).

Lemma 18 *Let $L \subseteq \mathbb{F}_2^n$ be a m-dimensional linear subspace with $m \leq n$. Let $x_1, \ldots, x_k$ be $k \geq m$ vectors sampled uniformly at random from L. Then it holds*

$$\mathbf{Pr}[\mathrm{span}\{x_1, \ldots, x_k\} = L] \geq 1 - 2^{m-k}\,. \tag{C5}$$

Lemma 18 can be exploited to learn the affine subspace A from U_A as explained in Algorithm 1. This algorithm is a variant of the more general *closure algorithm*, which has previously been used to efficiently solve on-line learning problems such as learning parity functions and integer lattices [60, 61] and which is used as a subroutine for subexponentially learning parities with noise [62]. The guarantees of Algorithm 1 are as follows.

Input: $\delta \in (0, 1)$ and access to $\mathtt{Samp}(U_A)$ for some affine subspace $A \subseteq \mathbb{F}_2^n$,

1: Let $k := n + \lceil \log(1/\delta) \rceil$. Obtain samples $\{x_1, \ldots, x_k\} \sim U_A$ by querying $\mathtt{Samp}(U_A)$.
2: Transform the samples $x_1, \ldots, x_k$ to $y_1, \ldots, y_k$ via $y_i = x_i + x_1$.
3: Use Gaussian elimination to determine from $y_1, \ldots, y_k$ a maximal linearly independent subset of vectors $V := \{y_{i_1}, \ldots, y_{i_m}\}$.
4: Form the full rank $n \times m$ matrix $\mathbf{R}$ by placing vectors from V as columns.
5: Output $(\mathbf{R}, x_1)$.

Algorithm 1: Affine subspace recovery from samples.

Lemma 19 (Efficient recovery of affine subspaces) *Let $A \subseteq \mathbb{F}_2^n$, $\delta \in (0,1)$ be as stated above. Algorithm 1 runs in time $O(\text{poly}(n, 1/\delta))$ and uses $O(\text{poly}(n, 1/\delta))$ samples, and outputs, with probability at least $1-\delta$, a tuple $(\mathbf{R}, t)$ which parametrizes A.*

Proof: The sample complexity is as stated in Algorithm 1. The time complexity follows from the fact that Gaussian elimination on an $n \times m$ matrix, $m < n$, takes time polynomial in n. It remains to prove the correctness of this algorithm.

Let $\mathbf{R}'$ be such that $(\mathbf{R}', x_1)$ parametrizes A. Line 2 transforms each $x_i \in A$ into a vector $y_i \in L$ where $L := \{\mathbf{R}'b \,|\, b \in \mathbb{F}_2^m\}$ is the linear subspace in A shifted by x_1. By assumption the original samples $\{x_1, \dots x_k\}$ are uniform on A. A linear transformation of a uniform distribution is another uniform distribution, such that the new samples $\{y_1, \dots, y_k\}$ are uniform on L. From Lemma 18 we obtain

$$\Pr[\text{span}\{y_1, \dots, y_k\} = L] \geq 1 - 2^{m-k} \geq 1 - 2^{n-k} \geq 1 - 2^{n-(n+\log(1/\delta))} = 1 - \delta. \tag{C6}$$

Hence, with probability at least $1-\delta$, the columns of $\mathbf{R}$ defined in Step 4 provide a basis for L.

To finish the proof assume that $\mathbf{R}$ is full rank and denote by A' the affine subspace parametrized by $(\mathbf{R}, x_1)$. Then for every $b \in \mathbb{F}_2^m$ it holds

$$\mathbf{R} \cdot b + x_1 \in \text{span}\{x_1, \dots, x_k\} \subseteq A \tag{C7}$$

and hence $A' \subseteq A$. Contrarily, since $\mathbf{R}$ has full rank $|A| = |A'|$. Thus $A = A'$, which completes the proof. □

We now combine these insights to prove the actual statement.

Theorem 1 *The set $\mathcal{D}_{\text{Cl}}$ of Clifford circuit distributions, for any depth, is efficiently learnable with respect to generators and evaluators.*

Proof: By Corollary 17, all distributions in $\mathcal{D}_{Cl}$ take the form of U_A for some affine subspace $A \subseteq \mathbb{F}_2^n$. Using Algorithm 1 in conjunction with Lemma 19 we obtain, with probability $1-\delta$, a parametrization $(\mathbf{R}, t)$ of A in time $\text{poly}(n, 1/\delta)$ using $\text{poly}(n, 1/\delta)$ many samples from U_A.

We now get an efficient generator for U_A by uniformly at random sampling $b \sim \mathbb{F}_2^m$ and outputting $\mathbf{R} \cdot b + t$. An efficient evaluator that computes $U_A(x)$ on input x is defined as follows: use Gaussian elimination in order to decide whether $x - t \in \mathbf{R}\mathbb{F}_2^m$. If this is the case return $2^- d$ with $d = \dim(A)$. Else return 0. Thus it is sample- and computationally-efficient to (ε, δ)-learn $\mathcal{D}_{\text{Cl}}$ with respect to a generator and evaluator. □

Appendix D: Proof of Theorem 2

Theorem 2 *Under the LPN assumption, the output distributions of local Clifford circuits of depth $d = n^{\Omega(1)}$ enriched with a single T-gate are not efficiently learnable with respect to an evaluator.*

Proof: For each string $s \in \{0,1\}^k$ let $\chi_{(s,k)} \in \mathcal{F}_k$ be the associated parity function on k bits – i.e. $\chi_{(s,k)}(x) = x \cdot s$ for all $x \in \{0,1\}^k$. For any $\eta \in (0, 1/2)$ we define the "noisy parity distribution on $k+1$ bits" $P_{(s,\eta,k)} \in \mathcal{D}_{k+1}$ via

$$P_{(s,\eta,k)}(x,y) = \begin{cases} 2^{-k} \cdot (1-\eta), & \text{if} \quad y = \chi_{s,k}(x) \\ 2^{-k} \cdot \eta, & \text{else}, \end{cases} \tag{D1}$$

for all $s, x \in \{0,1\}^k$. Define the distribution T_l as the trivial distribution on l bits - i.e. the distribution with $T_l(0^l) = 1$ and for any $k \leq n$ define $\mathcal{D}_\eta(n,k) \subseteq \mathcal{D}_{n+1}$ as the set of "noisy parity distributions on the first $k+1$ bits" – i.e. $\mathcal{D}_\eta(n,k) = \{P_{(s,\eta,k)} \otimes T_{n-k} \,|\, s \in \{0,1\}^k\}$.

In the proof of Theorem 16 in Ref. [5], the authors show that, under the LPN assumption, there is no efficient algorithm for learning the noisy parity distributions $\mathcal{D}_\eta(n,n)$ with respect to an evaluator, for any $\eta \in (c, 1/2 - c)$ where $c \in \Omega(1)$. In other words, in this parameter range, all algorithms for learning $\mathcal{D}_\eta(n,n)$ with respect to an evaluator require $\omega(\text{poly}(n))$ time. By using similar reasoning to that used in the proof of Lemma 7 – i.e. embedding the noisy parity distributions onto a subset of bits – one can extend this result to show that, assuming the LPN assumption, any algorithm for learning $\mathcal{D}_\eta(n,k)$ with respect to an evaluator requires $\omega(\text{poly}(k))$ time . As $\omega(\text{poly}(n^{\Omega(1)})) = \omega(\text{poly}(n))$ we can conclude that, assuming the LPN assumption, there exists no efficient algorithm for learning $\mathcal{D}_\eta(n, n^{\Omega(1)})$ with respect to an evaluator.

Next, we note that for any $s \in \{0,1\}^k$, when $\eta = \sin^2(\pi/8) \approx 0.146$, the distribution $P_{(s,\eta,k)}$ is the output distribution of the quantum circuit on $k+1$ qubits given in Fig. 2 with the CNOT gates between the i^{th} and the $(k+1)^{st}$ qubit for all $s_i = 1$. As such, for any $k \leq n$ and any $s \in \{0,1\}^k$, when $\eta = \sin^2(\pi/8)$ the distribution $P_{(s,\eta,k)} \otimes T_{n-k}$ is the output distribution of the quantum circuit on $n+1$ qubits, with the above mentioned circuit from Fig. 2 on the first $k+1$ qubits, and no gates on the remaining $n-k$

wires. While this circuit contains non-local two-qubit gates, we note that *any* Clifford unitary $U \in \mathrm{Cl}(2^k)$ can be implemented exactly using a depth $d = O(k)$ nearest-neighbour Clifford circuit [63]. By recompiling the circuit on the first $k+1$ qubits in this way, we obtain an $O(k)$ depth local "Clifford + one T" circuit whose output distribution is $P_{(s,\eta,k)} \otimes T_{n-k}$. Using this, the theorem statement follows from the previously established hardness of learning $\mathcal{D}_\eta(n, n^{\Omega(1)})$ with respect to an evaluator, when $\eta = \sin^2(\pi/8)$. □

Appendix E: Proof of Theorem 3

We start by defining the notion of a *pseudorandom* function whose existence is the primary assumption used for Theorem 3. For more detailed definitions and discussion of these objects, see Refs. [40, 50].

Definition 20 (Classical-secure and standard-secure pseudorandom functions) *Let $C \subseteq \mathcal{F}_n$ be a set of efficiently computable functions. We say that C is a classical-secure (standard-secure) pseudorandom function if for all classical-probabilistic (quantum) polynomial time algorithms $\mathcal{A}$, all polynomials p, and all sufficiently large n, it holds that*

$$\left| \Pr_{f\sim C}\left[\mathcal{A}^{\mathtt{MQ}(f)} = 1\right] - \Pr_{g\sim \mathcal{F}_n}\left[\mathcal{A}^{\mathtt{MQ}(g)} = 1\right] \right| < \frac{1}{p(n)}, \tag{E1}$$

where $\mathtt{MQ}(f)$ denotes the membership query oracle, which, when queried with some $x \in \{0,1\}^n$ returns $f(x)$.

At a high level, the above definition says that a set of functions C is classical-secure (standard-secure) if no classical (quantum) algorithm can, with non-negligible probability, distinguish functions drawn uniformly from C from functions drawn uniformly from $\mathcal{F}_n$. We note that the assumed existence of both classical-secure and standard-secure pseudorandom functions is standard in cryptography [64, 65]. We can now recollect the statement of Theorem 3.

Theorem 3 *Assuming the existence of classical-secure (standard-secure) pseudorandom functions, there is no efficient classical (quantum) algorithm for learning the output distributions of depth $d = n^{\Omega(1)}$ local quantum circuits, with gates from any universal gate set.*

Proof: Let $C \subseteq \mathcal{F}_n$ be a classical-secure (standard-secure) pseudorandom-function. Define $\mathcal{D} = \{P_f \,|\, f \in C\}$ where

$$P_f(x,y) = \begin{cases} 2^{-n}, & \text{if} \quad y = f(x) \\ 0, & \text{else.} \end{cases} \tag{E2}$$

In Ref. [5] Theorem 17 the authors show that $\mathcal{D}$ is not efficiently classically learnable with respect to a generator, assuming that C is classical-secure. Their hardness result can be straightforwardly extended to apply to quantum learning algorithms as well by requiring C to be standard-secure. To leverage their result to show hardness for the output distributions of quantum circuits, we will show how to embed $\mathcal{D}$ into a suitable class of quantum circuits. To do so recall that any classical Boolean circuit can be implemented as a quantum circuit via the standard implementation of reversible classical gates together with uncomputation (see Chapter 3 of Ref. [66]). For a polynomial size classical circuit, this might incur at most a polynomial overhead in the number of ancilla qubits necessary. Hence, for all $f \in C \subseteq \mathcal{F}_n$ there exists a polynomial size quantum circuit C_f on $\mathrm{poly}(n)$ many qubits whose output distribution is $\tilde{P}_f$ with

$$\tilde{P}_f(x,y,z) = \begin{cases} 2^{-n}, & \text{if} \quad z = f(x) \text{ and } y = 0^m \\ 0, & \text{else}, \end{cases} \tag{E3}$$

where $m = O(\mathrm{poly}(n))$. Note that any such quantum circuit C_f can be turned into a nearest-neighbor circuit by qubit routing techniques such as using SWAP gates. This will again incur only a polynomial overhead in both size and depth of the circuit. Denote by $\tilde{\mathcal{D}} = \{\tilde{P}_f \mid f \in C\} \subset \mathcal{D}_{n+m+1}$ the class of all such distributions. Since $n + m + 1 = O(\mathrm{poly}(n))$ we find that $\tilde{\mathcal{D}}$ is hard to learn.

Lastly, note that $\tilde{\mathcal{D}}$ is a subset of the set of the output distributions of polynomial depth quantum circuits. As such, the output distributions of polynomial depth quantum circuits are not efficiently learnable with respect to a generator. We can see that this holds irrespective of the gate set used (as long as it is universal) by combining Corollary 15 and Lemma 11. Finally, using Corollary 9, we see that already for $n^{\Omega(1)}$ deep circuits there cannot exist any efficient classical (quantum) algorithm for learning the output distribution with respect to a generator. □

Appendix F: Proof of Theorem 4

Before proving Theorem 4, we recall a connection of distribution and Boolean function statistical query oracles.

Definition 21 (Boolean function statistical query oracle [46]) *Let $f \in \mathcal{F}_n$ be a Boolean function, $\tau \in (0,1)$ and let $P \in \mathcal{D}_n$ be a distribution. The Boolean function statistical query oracle of f with respect to P and tolerance τ is defined as the oracle $\mathtt{Stat}_{\tau,P}(f)$ that, when queried with a function $\phi : \{0,1\}^{n+1} \to [-1,1]$ returns some v such that $|\mathbf{E}_{x\sim P}[\phi(x, f(x))] - v| \leq \tau$.*

Corollary 22 *Let $f \in \mathcal{F}_n$ be a Boolean function and let $P \in \mathcal{D}_n$ be a distribution. Define the distribution $P_f \in \mathcal{D}_{n+1}$ as*

$$P_f(x,y) = \begin{cases} P(x)\,, & \text{if} \quad y = f(x) \\ 0\,, & \text{else}\,. \end{cases} \tag{F1}$$

Then, for any $\tau \in (0,1)$ any statistical query oracle $\mathtt{Stat}_\tau(P_f)$ is a Boolean function statistical query oracle $\mathtt{Stat}_{\tau,P}(f)$ and vice versa.

Now we are able to prove the theorem.

Theorem 4 *There is no query efficient algorithm for learning from inverse polynomially accurate statistical queries*

- *$\mathcal{D}_{\mathrm{Cl}}$ at depth $\omega(\log(n))$,*
- *$\mathcal{D}_{\mathcal{G}}$ at depth $\omega(\log^k(n))$ where k is a constant depending on the universal gate set $\mathcal{G}$ (which can be as small as 2),*

with respect to either generators or evaluators.

Proof: To prove the first claim we will reduce statistical query learning of parity functions to statistical query distribution learning of Clifford distributions. For each string $s \in \{0,1\}^n$ let $\chi_s \in \mathcal{F}_n$ be the associated parity function, and let $C = \{\chi_s \,|\, s \in \{0,1\}^n\}$ be the class of parity functions. For each $s \in \{0,1\}^n$ define $P_s = P_{\chi_s}$ and denote by $\mathcal{D} = \{P_s \mid s \in \{0,1\}^n\} \subset \mathcal{D}_{n+1}$ the class of parity distributions.

As shown in the seminal work of Refs. [46, 47], any algorithm with Boolean function statistical query access of tolerance $\Omega(2^{-n/3})$ to C, requires at least $\Omega(2^{n/3-1})$ queries for learning the class of parity functions with respect to the uniform distribution, for any failure probability less than $1/2 - O(2^{-3n})$.

We now show that, for any $\varepsilon < 1/2$, a statistical query algorithm for (ε,δ)-learning $\mathcal{D}$ with respect to an evaluator or generator from q many queries implies a statistical query algorithm for $(0,\delta)$-PAC learning C from q many queries. Assume there exists an algorithm $\mathcal{A}$ for (ε,δ)-learning $\mathcal{D}$ with respect to an evaluator or generator from q many queries to $\mathtt{Stat}_\tau(P)$ and with $\varepsilon < 1/2$. We then define the algorithm $\mathcal{A}'$ which when given access to $\mathtt{Stat}_{\tau,U}(\chi_s)$, for some unknown $s \in \{0,1\}^n$, does the following:

1. $\mathcal{A}'$ runs learning algorithm $\mathcal{A}$ where any query to $\mathtt{Stat}_\tau(P_s)$ is simulated by querying $\mathtt{Stat}_{\tau,U}(\chi_s)$. After at most q queries, $\mathcal{A}$ will output an evaluator $\mathtt{Eval}_Q$ (a generator $\mathtt{Gen}_Q$) for some distribution $Q \in \mathcal{D}_{n+1}$, which, with probability at least $1-\delta$, is at most ε far from P_s.

2. $\mathcal{A}'$ uses $\mathtt{Eval}_Q$ ($\mathtt{Gen}_Q$) to find s by brute force. This can be achieved by iterating through all strings $s \in \{0,1\}^n$, test $\mathrm{TV}(P_s, Q) < 1/2 - \varepsilon$ and return s if true. While this step is not computationally efficient, it requires no additional queries to the oracle.

To see that $\mathcal{A}'$ is correct we note that any two parity distributions are $\mathrm{TV}(P_s, P_t) = 1/2$ far apart. Now, assume $\mathcal{A}'$ runs with access to χ_s, then we find with probability $1-\delta$ that $\mathrm{TV}(Q, P_s) < \varepsilon$. Therefore, any t with $\mathrm{TV}(P_t, Q) < 1/2 - \varepsilon$ also fulfills $\mathrm{TV}(P_s, P_t) \leq \mathrm{TV}(P_s, Q) + \mathrm{TV}(Q, P_t) < 1/2$, where we used $\varepsilon < 1/2$. This implies $t = s$. Since all queries in $\mathcal{A}'$ are due to $\mathcal{A}$ we conclude that $\mathcal{A}'$ is an $(0,\delta)$ statistical query learner for C with respect to the uniform distribution, which requires at most q queries. We conclude that for any $\varepsilon < 1/2$ and $\delta < 1/2$, the problem of (ε,δ)-distribution learning $\mathcal{D}$ requires at least $\Omega(2^{n/3-1})$ queries.

Let us now turn to $\mathcal{D}_{\mathrm{Cl}}$. For any $s \in \{0,1\}^n$, the distribution P_s is the output distribution of the quantum circuit shown in Fig. 2, without the red box, and with the CNOT gates determined by the string s. Moreover, any Clifford unitary $U \in \mathrm{Cl}(2^n)$ can be realized by a depth $d = O(n)$ circuit consisting only of nearest neighbour two-qubit Clifford gates [63]. It therefore follows that the output distributions of linear depth local Clifford circuits are exponentially hard to learn from statistical queries.

Using Lemma 7 we will now trade the query complexity for the depth at which the hardness sets in. In the previous paragraph, we have shown that learning depth d Clifford distributions from statistical queries with tolerance $\tau(n) = \Omega(2^{-n/3})$ requires at least $q(n) = \Omega(2^{n/3})$ queries for any depth $d(n) = \Omega(n)$. Let $n \leq g(n) = o(2^n)$, $d = \Omega(n)$ and define $d'(n) = d \circ g^{-1}(n)$. Thus

$\omega(\log(n)) = g^{-1}(n) \leq n$ and $\omega(\log(n)) = d'(n)$. Lemma 7 then implies that learning $\mathcal{D}_{\mathrm{Cl}}(n, d'(n))$ from statistical queries with tolerance $\tau'(n) = \tau \circ g^{-1}(n)$ requires at least $q'(n) = q \circ g^{-1}(n)$ many queries with

$$\Omega(2^{-n/3}) = \tau'(n) = 2^{-\omega(\log(n))} \tag{F2}$$

$$q'(n) = 2^{\omega(\log(n))}\,. \tag{F3}$$

In particular, for any $\varepsilon, \delta < 1/2$ any statistical query algorithm for learning super logarithmic depth Clifford circuit distributions with inverse polynomial tolerance requires super polynomially many queries.

To obtain the second claim we first apply Corollary 15 in conjunction with Lemma 12 to find that the statistical query complexity for learning $\mathcal{D}_{\mathcal{G}}(n, d\log^c(n \cdot d/\sigma))$ with tolerance $\tau > \sigma$ is lower bounded by that of learning $\mathcal{D}_{\mathrm{Cl}}(n, d)$ with any tolerance $\gamma < \tau - \sigma$. Now fix any $\tau \in \Omega(1/\mathrm{poly}(n))$, let $\gamma = \tau/3$ and $\sigma = \tau/3$ such that $\gamma + \sigma < \tau$. As just shown, for any $\varepsilon, \delta < 1/2$ we know that (ε, δ)-learning $\mathcal{D}_{\mathrm{Cl}}(n, \omega(\log(n)))$ requires $\omega(\mathrm{poly}(n))$ statistical queries with tolerance at least $\gamma = \Omega(1/\mathrm{poly}(n))$. Hence, for any $\varepsilon < 1/2 - \sigma$ and $\delta < 1/2$, we find that (ε, δ)-learning $\mathcal{D}_{\mathcal{G}}(n, d)$ takes $\omega(\mathrm{poly}(n))$ many statistical queries of tolerance τ with

$$d = \omega\left(\log(n) \cdot \log^c\left(\frac{n \cdot \log(n)}{\tau}\right)\right) = \omega\left(\log^{c+1}(n)\right), \tag{F4}$$

where we have used $1 < \tau^{-1}$. Setting $k = c + 1$ completes the proof. Importantly, as noted earlier, for some gate sets $c = 1$ and hence $k = 2$ [57]. □