

Clifford Circuits can be Properly PAC Learned if and only if $RP = NP$

Daniel Liang

Department of Computer Science
University of Texas at Austin

Given a dataset of input states, measurements, and probabilities, is it possible to efficiently predict the measurement probabilities associated with a quantum circuit? Recent work of Caro and Datta [19] studied the problem of PAC learning quantum circuits in an information theoretic sense, leaving open questions of computational efficiency. In particular, one candidate class of circuits for which an efficient learner might have been possible was that of Clifford circuits, since the corresponding set of states generated by such circuits, called stabilizer states, are known to be efficiently PAC learnable [44]. Here we provide a negative result, showing that proper learning of CNOT circuits with $1/\text{poly}(n)$ error is hard for classical learners unless $RP = NP$, ruling out the possibility of strong learners under standard complexity theoretic assumptions. As the classical analogue and subset of Clifford circuits, this naturally leads to a hardness result for Clifford circuits as well. Additionally, we show that if $RP = NP$ then there would exist efficient proper learning algorithms for CNOT and Clifford circuits. By similar arguments, we also find that an efficient proper quantum learner for such circuits exists if and only if $NP \subseteq RQP$. We leave open the problem of hardness for improper learning or $\mathcal{O}(1)$ error to future work.

1 Introduction

The goal of efficient learning of quantum states and the circuits that act on them, is to be able to predict the outcome of various measurements with some degree of accuracy. For example, given a quantum state ρ and a two-outcome measurement M can we predict the probability that the measurement accepts?

Naively, one can try and learn everything there is to know about the system, a technique known as tomography with versions for quantum states [29, 41, 42] and quantum processes [4, 21]. However, this requires exponential time in the number of qubits due to information theoretic reasons related to the exponential dimension of the system. This exponential bound remains when trying to find a state close in trace distance as shown by a combination of Flammia et al. [24] and Holevo's bound. To address this, one can choose to restrict the type of information one wanted to learn, which led to the ideas of shadow tomography [2] and classical shadows [31]. By only needing to predict the value of M observables $\{O_i\}$, one is able to use only a number of measurements that is polynomial in

Daniel Liang: dliang@cs.utexas.edu

the number of qubits and polylogarithmic in M . In a similar vein, Aaronson [1] proposed the idea of PAC learning quantum states, which is the idea of learning relative to some distribution over measurements, but only being given samples from that distribution as well (see Section 2.5 for details). Aaronson was then able to give a generalization theorem for this problem, showing that if one could find a hypothesis h that had small training error on $\mathcal{O}(n)$ samples that h would also perform well on future samples. However, the problem of efficiently finding h was left open.

An alternative direction was to restrict the class of objects being learned on, but allow one to choose what kind of measurements are taken. Montanaro [40] was able to learn stabilizer states and later Low [39] learned an unknown Clifford circuit. Lai and Cheng [38] built on these results in the case of actually recovering the circuit, as well as limited learning in the presence of a small amount of non-Clifford gates. Stabilizer states and Clifford circuits are of particular interest to the quantum information community because many quantum communication protocols and well known quantum query algorithms [10–13, 26, 46] utilize these states and circuits. Gottesman and Knill [27] (with improvements by Aaronson and Gottesman [3]) were able to give an efficient classical simulation of these objects, showing these class of objects to seemingly be much simpler than the set of all quantum states or circuits. Combined with the fact that stabilizer states are good approximations to Haar random states [37, 50], we get a set of circuits and states that are highly quantum with many interesting uses, but still have enough exploitable structure to be classically simulable, making them a prime candidate for learning.

Rocchetto [44] was able to combine the ideas of PAC learning with the structure provided by restricting to stabilizer states to give an efficient PAC algorithm for learning stabilizer states. Caro and Datta [19] extended the ideas of PAC learning to quantum circuits, giving an analogous generalization theorem to Aaronson [1]. As with Aaronson [1], the problem of efficiently finding such a good hypothesis was left open. A natural follow-up was whether or not Clifford circuits could be efficiently PAC learned in an analogous way to stabilizer states. Here, we are given inputs of the form $(\rho, \frac{I^{\otimes n} + P}{2})$ for some stabilizer state ρ and Pauli matrix P , with labels $\text{Tr}[\frac{I^{\otimes n} + P}{2} C \rho C^\dagger]$ corresponding to an unknown Clifford circuit C and asked to predict future labels. It is worth noting that we have slightly altered the definition of PAC learning a quantum circuit from that of Caro and Datta [19] to a setting we find more comparable to Aaronson’s original PAC learning result for quantum states [1]. In the setting introduced by Caro and Datta, the measurements were limited to being rank 1 projectors with product structure, rather than the rank 2^{n-1} projectors we use in our proof.

When one attempts to create a PAC learning algorithm a natural first step is to try an elimination method, i.e., eliminating options that don’t match the given training data and then outputting some option that does match the data well. Such algorithms are known as *proper* learning algorithms (see Section 2.5 for more details) and were the only kind of learning algorithms considered when the idea of PAC learning was first introduced by Valiant [47]. And while the learning theory community now considers things like improper learning algorithms, the original proper learning algorithms generally remain the most natural class of learning algorithms to consider first. We note for instance that Rocchetto [44] is a proper learning algorithm, as well as learning algorithms for parities and other well known learning problems [35]

To that extent, we show in this paper that an efficient proper learner for Clifford circuits that achieves $1/\text{poly}(n)$ error exists if and only if $\text{RP} = \text{NP}$, effectively ruling out “straight-forward” learning algorithms for Clifford circuits. More generally, these results apply to any proper learner that achieves arbitrary error (ϵ, δ) with runtime $\text{poly}(n, \epsilon^{-1}, \delta^{-1})$, which

is known as a *strong* learner. Furthermore, this is true even just for a learner of a subset of Clifford circuits called CNOT circuits. This subset essentially restricts to the set of Clifford circuits that map computational basis states to other computational basis states and these circuits are highly related to the complexity class L (see Section 2.4). We leave open the problem of showing $\mathcal{O}(1)$ hardness for proper learners using complexity theoretic means, such as in Guruswami and Raghavendra [28].

One can also imagine that the learning algorithm has access to a quantum computer. Since there exists problems like factoring [45] for which we have an efficient quantum algorithm but not an efficient classical algorithm, this learner may be able to efficiently learn more expressive concept classes. We also give results for this setting by relating NP to RQP, the quantum analogue of RP. We now informally state our main theorems regarding CNOT and Clifford circuits.

Theorem 1.1. *There exists an efficient randomized proper PAC learner for CNOT circuits if and only if $RP = NP$. Furthermore, an efficient quantum proper PAC learner for CNOT circuits exists if and only if $NP \subseteq RQP$.*

Corollary 1.2. *There exists an efficient randomized proper PAC learner for Clifford circuits if and only if $RP = NP$. Furthermore, an efficient quantum proper PAC learner for Clifford circuits exists if and only if $NP \subseteq RQP$.*

The proofs of these main results starts by realizing that finding a CNOT circuit with zero training error requires finding a full rank matrix in an affine subspace of matrices under matrix addition (so as to differentiate from a coset of a matrix group using matrix multiplication). This is known as the NONSINGULARITY problem [17] and is NP-complete. While this may seem like a backwards reduction, it turns out that the set of matrix affine subspaces used to show that NONSINGULARITY can solve 3SAT are a subset of the ones needed to learn CNOT circuits with zero training error. Thus, there exist a set of samples such that a CNOT circuit with zero training error exists if and only if the SAT instance is satisfiable. Finding such a CNOT circuit is what is known as the search version of the *consistency problem* and in turn the decision version of the consistency problem is also NP-complete.

To show that an efficient proper learner for CNOT circuits implies $RP = NP$, we follow the same proof structure as similar results for NP-hardness of the consistency problem for 2-clause CNF, 3-DNF, or the intersection of two halfspaces [14, 15, 30]. First, let S be some sample from the decision version of the consistency problem for CNOT circuits. Using the uniform distribution over each element in S , we will sample every element of S with high probability given enough queries. Since S contains at most a polynomial number of samples, we are able to show that an efficient learner with arbitrary $1/\text{poly}(n)$ error would necessarily also solve the consistency problem with high enough probability to create a solution in RP.

Completing the proof in the other direction, if $RP = NP$ we utilize search-to-decision reductions for NP-complete problems to get an efficient algorithm for the search problem of minimizing training error. We can treat this search algorithm as our means of generating a hypothesis circuit C with low training error. By the generalization theorem provided by Caro and Datta [19], assuming we have enough samples, this C will properly generalize and have low true error, thus completing the proof. The quantum forms of the proof essentially come for free by replacing RP with RQP everywhere and using learners capable of doing quantum computation.

1.1 Related Work

We also note that we are dealing with the problem of classically PAC learning a classical function (i.e., classical labels) derived from a quantum system. This is as opposed to quantum PAC learning of a classical function as in Arunachalam and de Wolf [6], Arunachalam and de Wolf [7], Arunachalam et al. [8], Quek et al. [43] where instead of a distribution over samples we receive access to copies of a quantum state. This state results in the same distribution classically when measured in the computational basis but can be measured in other basis to get different results. There is also the attempt to directly learn a quantum process with quantum labels, as in Caro [18], Chung and Lin [22]. Here, they do not choose to measure the output state, and have samples of the form $(\rho, \mathcal{M}(\rho))$ for quantum process \mathcal{M} . Other related quantum learning works, some of which are outside the PAC model, include Cheng et al. [20], Low [39], Yoganathan [49].

2 Preliminaries

2.1 Quantum States and Circuits

A quantum state ρ on n qubits is a $2^n \times 2^n$ PSD matrix with trace 1. If the matrix is rank 1 then we refer to ρ being a PURE STATE, since it can be decomposed as $\rho = |\psi\rangle\langle\psi|$ where $|\psi\rangle$ is a 2^n -dimensional column vector with norm 1 and $\langle\psi|$ is its complex conjugate. A two-outcome measurement E is then a projector such that $E^2 = E$ such that the probability of a ‘1’ outcome is $\text{Tr}[E\rho]$ and the probability of a ‘0’ outcome is $1 - \text{Tr}[E\rho]$, leaving the expectation value as simply $\text{Tr}[E\rho]$.

A quantum process is how one evolves a quantum state, and therefore it must preserve the trace 1 and the PSD condition. We will be primarily interested in quantum circuits, which are the subset of quantum processes that map pure states only to other pure states. These are constrained to be unitary operations, such that after acting on ρ with the circuit C , the state that we are left with is $C\rho C^\dagger$ where C^\dagger is the complex conjugate of C .

2.2 Paulis and Stabilizer States/Groups

We will start by giving the following matrices, known as the PAULI MATRICES.

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Noting that these are all unitaries that act on a single qubit, we can generalize to n qubits.

Definition 2.1. Let $\mathcal{P}_n = \{\pm 1, \pm i\} \times \{I, X, Y, Z\}^{\otimes n}$ be the matrix group consisting all n -qubit Paulis with phase ± 1 or $\pm i$.

We’ll also introduce some shorthand notation:

Definition 2.2. Let X_i and Z_i be the Pauli acting only on the i -th qubit with X or Z respectively and the identity matrix on all other qubits.

Definition 2.3. For $v \in \{0, 1\}^n$, let $X^v = \prod_{i=1}^n X_i^{v_i}$ and $Z^v = \prod_{i=1}^n Z_i^{v_i}$.

Note that $Z^v \cdot Z^w = Z^{v+w}$, assuming the dimensions of v and w match. It is easy to see that $v \neq w$ also implies that $Z^v \neq Z^w$.

A STABILIZER STATE ρ is any state that can be written as $\frac{1}{2^n} \sum_{g \in G} g$, where G is an abelian subgroup $G \subset \mathcal{P}_n \setminus \{-I^{\otimes n}\}$ without the negative identity. G is known as the STABILIZER GROUP of ρ . As it turns out, if G is of order 2^n then ρ will be a pure state. This leads to the alternative (and more popular definition) where $\rho = |\psi\rangle\langle\psi|$, is the unique state that is stabilized by G . That is, for all $g \in G$, $g|\psi\rangle = |\psi\rangle$. This definition shows why $-I^{\otimes n}$ isn't allowed to be in G , since $-I^{\otimes n}$ stabilizes nothing. It also shows why one must restrict the entries of G to only have real phase.

Proposition 2.4. *Any abelian subgroup of $G \subseteq \mathcal{P}_n \setminus \{-I^{\otimes n}\}$ cannot contain any Paulis with an imaginary phase.*

Proof. Given a Pauli with an imaginary phase, its square would be equal to $-I^{\otimes n}$, making the group not closed. This is a contradiction. \square

One of the reasons stabilizer states are so important is this bijection between the stabilizer group of a stabilizer state and the state itself; by simply knowing the generators of the group, one can easily reconstruct the state. And since there are at most n generators, if one can efficiently write down the generators themselves then there is a polynomial size representation of a stabilizer state. We now show how one can write down any member of a stabilizer group as follows. Given, $P \in \mathcal{P}_n$ with real phase such that $P = \pm \otimes_i P_i$, define a function $N : \mathcal{P} \rightarrow \{0, 1\}^2$ for each qubit $N(I) = 00$, $N(X) = 10$, $N(Z) = 11$, and concatenate to make $N(P) = (N(P_1), N(P_2), \dots, N(P_n))$. Additionally, have an extra bit for the sign for whether the sign is -1 or 1 . This results in a $2n + 1$ bit string for each generator, so writing down a stabilizer state requires only $\mathcal{O}(n^2)$ bits to write down classically.

2.3 Clifford Circuits

Informally, a Clifford circuit maps stabilizer states to other stabilizer states.

Definition 2.5. A CLIFFORD CIRCUIT is a unitary U such that $U\mathcal{P}_nU^\dagger = \mathcal{P}_n$, while ignoring global phase on the unitary. More formally, consider the normalizer $\mathcal{N}(\mathcal{P}_n) = \{U \in U(2^n) \mid U\mathcal{P}_nU^\dagger = \mathcal{P}_n\}$, and let $\mathcal{C}_n = \mathcal{N}(\mathcal{P}_n)/U(1)$ be the CLIFFORD GROUP.

Like stabilizer states, generators are an important part of how we deal with Clifford circuits. How a given Clifford circuit U acts on the generators of the Pauli matrices completely characterizes the unitary U [39]. To borrow the notation of Koenig and Smolin [36], this relationship can be efficiently described via:

$$UX_jU^\dagger = (-1)^{p_j} \prod_{i=1}^n X_i^{\alpha_{ij}} Z_i^{\beta_{ij}} \quad UZ_jU^\dagger = (-1)^{q_j} \prod_{i=1}^n X_i^{\gamma_{ij}} Z_i^{\theta_{ij}} \quad (1)$$

where p_i , q_i , α_{ij} , β_{ij} , γ_{ij} , and θ_{ij} are all $\{0, 1\}$ values. It will sometimes be useful to view α_{ij} , β_{ij} , γ_{ij} , and θ_{ij} as the $n \times n$ boolean matrices A , B , Γ , and Θ respectively. This gives us a simple upper-bound on the number of Clifford circuits.

Proposition 2.6. *There are at most $2^{\mathcal{O}(n^2)}$ Clifford circuits.*

Proof. The total number of bits we use to represent p , q , A , B , Γ , and Θ is $4n^2 + 2n = \mathcal{O}(n^2)$. There can then be at most $2^{\mathcal{O}(n^2)}$ Clifford circuits. \square

However, because commutation relations are preserved, not all possible values of $\alpha, \beta, \gamma, \theta$ are allowed (the p and q values can be arbitrary). This leads us to the idea of symplectic matrices. We note that a Clifford circuit can be encoded as a $(2n + 1) \times 2n$ boolean matrix S where column $2j - 1$ is equal to $(\alpha_{1j}, \beta_{1j}, \dots, \alpha_{nj}, \beta_{nj}, p_j)$ and column $2j$ is equal to $(\gamma_{1j}, \theta_{1j}, \dots, \gamma_{nj}, \theta_{nj}, q_j)$. We will call this the *full encoding* of the Clifford circuit.

Definition 2.7. A symplectic matrix over \mathbb{F}_2^{2n} is a $2n \times 2n$ matrix S with entries in \mathbb{F}_2 such that

$$S^T \Lambda(n) S = \Lambda(n) \equiv \bigoplus_{i=1}^n \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \quad (2)$$

These matrices form the symplectic group $\text{Sp}(2n, \mathbb{F}_2)$.

The symplectic matrices preserve the symplectic inner product $\omega(v, w) = v^T \Lambda(n) w$ on \mathbb{F}_2^{2n} . It turns out that if we consider the submatrix defined by the first $2n$ rows of our full encoding S , a necessary and sufficient condition to preserve the commutation relations of the generators is for this submatrix to be symplectic, as $\{X_i\} \cup \{Z_i\}$ form what is known a symplectic basis. Formally, $\mathcal{C}_n/\mathcal{P}_n \cong \text{Sp}(2n, \mathbb{F}_2)$.

2.4 CNOT circuits and $\oplus\mathbb{L}$

It is a well known fact that every Clifford circuit can be generated using only H , P , and CNOT gates as defined below:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad \text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

We note that $X = HP^2H$. If we restrict to the subset of circuits that are generated by only X and CNOT, we get what are known as CNOT circuits [3], which are a clear subset of Clifford circuits.

Definition 2.8 (Aaronson and Gottesman [3]). The complexity class $\oplus\mathbb{L}$ is the class of problems that reduce to simulating a polynomial-size CNOT circuit.

A perhaps more familiar definition for complexity theorists is the class of problems that are solvable by a nondeterministic logarithmic-space Turing machine that accepts if and only if the total number of accepting paths is odd.

Let us now consider the set of all Clifford circuit that map computational basis states to other computational basis states, thereby stabilizing the subgroup $\{\pm 1\} \times \{I, Z\}^{\otimes n}$. Very briefly, we will call these *classical Clifford circuits* as we will now prove that they are largely equivalent to CNOT circuits. The following lemmas will be useful.

Lemma 2.9. Let Θ be the matrix form of the θ_{ij} from Eq. (1). Any CNOT circuit C must have Θ be full rank.

Proof. Let us first consider what happens to a computational basis state when acted upon by C and let S be the full encoding of C . Referencing Eq. (1), the γ_{ij} must be 0 for all i and j . Since every member of $\text{Sp}(2n, \mathbb{F}_2)$ is full rank, the even columns of S must be as well. Since the γ terms are all zero, the even columns of S are full rank if and only if Θ is full rank. \square

Lemma 2.10. *Let Θ be the matrix form of the θ_{ij} from Eq. (1) for some Clifford circuit C . If Θ is full rank then there exists a CNOT circuit with the same Θ .*

Proof. One can verify that the Θ matrix of the circuit that does nothing, which is a valid CNOT circuit as well, is the identity matrix. We note that a CNOT from qubit i to qubit j performs the rowsum operation of adding row j to row i of Θ . Thus it is possible to efficiently construct a circuit with matching Θ using rowsum operations via CNOT gates. \square

We can now prove our desired goal leveraging these two lemmas.

Proposition 2.11. *Let C be an arbitrary classical Clifford circuit. It can be efficiently generated using solely X , Z , and CNOT gates. Moreover, its effect on the computational basis states can be entirely simulated using only X and CNOT.*

Proof. Let us first consider what happens to a computational basis state when acted upon by C . Referencing Eq. (1), the γ_{ij} must be 0 for all i and j , and we will essentially ignore α_{ij} , β_{ij} , and p_j for now leaving us with θ_{ij} and q_j . By Lemma 2.9, Θ must full rank. By Lemma 2.10, there exists a CNOT matrix that achieves the same Θ as well. To get a matching q_j , one can simply apply an X gate at the beginning of each qubit that has $q_j = 1$, since $XZX = -Z$, and the following CNOT gates will not itself introduce any negative phases. From here, we have already proved the *moreover* statement.

To prove the full result, we return to the α_{ij} and β_{ij} . We will show that there exists a single unique solution. Similar to Θ we will define the corresponding $n \times n$ matrices A and B for the α_{ij} and β_{ij} respectively. Based on Eq. (2), to form a symplectic basis we find that $A^T\Theta = I$ and $A^TB = 0$, since $\gamma_{ij} = 0$. Clearly $A^T = \Theta^{-1}$, which is guaranteed to exist, and $B = 0$ since A will also be full rank. To match the p_j values we simply place Z gates in front of the qubits where $p_j = 1$, similar to the X gates for q_j . \square

This means we do not lose any kinds of interactions by only considering CNOT circuits, since the only differentiating factors (i.e., the Z gates) do not actually affect the outcome when fed with a computational basis state. As such, all given results will be given in terms of simply CNOT circuits.

2.5 PAC Learning

The goal of PAC learning is to learn a function relative to a certain distribution of inputs, rather than in an absolute sense. Let's say we want to learn an arbitrary f from some concept class \mathcal{C} . If a hypothesis function h matches the true function f on many of the high probability inputs, then we can say that we have *approximately learned* f . If we can do this with high probability for arbitrary f , then we *probably* approximately (PAC) learned \mathcal{C} .

Definition 2.12. Let Ω be some domain of inputs and let \mathcal{C} be a set of functions $f : \Omega \rightarrow [0, 1]$. We say that \mathcal{C} is (ϵ, δ) -PAC-learnable if there exists a learner that, when given samples of the form $(x, f(x))$ for $x \sim \mathcal{D}$ for arbitrary f and unknown distribution \mathcal{D} , outputs with probability at least $1 - \delta$, over both the samples and the learning algorithm,

a hypothesis h with error¹ satisfying

$$\mathbb{E}_{x \sim \mathcal{D}} [(f(x) - h(x))^2] \leq \epsilon.$$

The number of samples used is referred to as the sample complexity, and we refer to the learner being *efficient* if it can find such an h in time $\text{poly}(n, \epsilon^{-1}, \delta^{-1})$ for arbitrary ϵ and δ .

From here, one can define two types of learning, based on where h comes from. If h is allowed to be any function that meets the PAC constraints, we refer to this as IMPROPER LEARNING. If instead $h \in \mathcal{C}$, we get what is known as PROPER LEARNING, which will be the focus of this paper. With proper learning, we can then begin to talk about the consistency problem formally.

Definition 2.13. Let S be a set of labeled samples such that $|S| < s$. Let $\text{CONSISTENTSEARCH}(\mathcal{C}, s)$ be the problem of finding a function $h \in \mathcal{C}$ that is *consistent* with all of S (i.e., for all $(x, f(x)) \in S$, $f(x) = h(x)$) if such an h exists, otherwise reject.

2.5.1 Generalization

Intuitively, given a set of samples the best one can really hope to do is find such an h that gets zero training error and hope that the true error for h is also low. This leads to the idea of *generalization*, which aims to show that doing well on a large enough set of training data (i.e., the consistency problem) allows one to give the PAC guarantee as well with high probability. In terms of computational efficiency, this effectively reduces the problem of proper learning to the consistency problem, or an approximation of the consistency problem. The most common approach to guarantee generalization is to bound the “expressiveness” of the concept class, such as with the VC-dimension [16]. Since VC-dimension is defined for $\{0, 1\}$ labels, we will now give a generalization of VC-dimension in the regression setting.

Definition 2.14. Let Ω be some domain of inputs, $\eta > 0$, and let \mathcal{C} be a set of functions $f : \Omega \rightarrow [0, 1]$. We say that a set of inputs $\{x_1, x_2, \dots, x_m\} \subseteq \Omega$ is η -*fat-shattered* by \mathcal{C} if there exists a set $y_1, y_2, \dots, y_m \in [0, 1]$ such that for any vector $b = \{\pm 1\}^m$ there is an $f_C \in \mathcal{C}$ that satisfies $b_i \cdot (f_C(x_i) - y_i) \geq \eta$.

Definition 2.15. The η -*fat-shattering dimension* of a concept class \mathcal{C} is the size of the largest set of inputs that is fat-shattered by \mathcal{C} . We denote this as $\text{fat}_{\mathcal{C}}(\eta)$.

At a high-level, the η parameter provides a buffer such that each $f_C(x_i)$ is robustly bounded away from y_i by η in the appropriate direction. We now give a result saying that bounded fat-shattering dimension implies generalization from the training data to the actual learning task.

Theorem 2.16 (Anthony and Bartlett [5] Corollary 3.3). *Let \mathcal{C} be a concept class from Ω to $[0, 1]$ and let \mathcal{D} be some distribution on Ω . Let $\delta, \epsilon, \alpha, \beta \in (0, 1)$ be parameters such*

¹Since this problem is a regression problem rather than binary classification, we claim that this squared-loss is a more natural notion of error than the kind used in Aaronson [1], Caro and Datta [19], and other older papers [5]. It also allows us to express PAC learning using only 2 error parameters, rather than 3. We can also still recover some form of the 3 parameter bound using $\Pr_{x \sim \mathcal{D}} [|f(x) - h(x)| > t] \leq \frac{1}{t^2} \mathbb{E}_{x \sim \mathcal{D}} [(f(x) - h(x))^2]$ via Markov’s inequality.

that $\beta > \alpha$. Furthermore, let $\{x_1, x_2, \dots, x_m\}$ and $\{y_1, y_2, \dots, y_m\}$ be a set of m samples drawn i.i.d. from some distribution \mathcal{D} where $y_i = f(x_i)$ for some $f \in \mathcal{C}$. If $h \in \mathcal{C}$ satisfies $|h(x_i) - y_i| \leq \alpha$ for all $1 \leq i \leq m$ then

$$m = \mathcal{O}\left(\frac{1}{\epsilon} \left(\text{fat}_{\mathcal{C}}\left(\frac{\beta - \alpha}{8}\right) \log^2\left(\frac{\text{fat}_{\mathcal{C}}\left(\frac{\beta - \alpha}{8}\right)}{(\beta - \alpha)\epsilon}\right) + \log\frac{1}{\delta}\right)\right)$$

number of samples suffices to achieve

$$\mathbb{E}_{x \sim \mathcal{D}}[(h(x) - f(x))^2] \leq (1 - \epsilon)\beta^2 + \epsilon$$

with probability at least $1 - \delta$ over the samples.

The following folklore bound on fat-shattering dimension is very loose, but still sufficient for our purposes of complexity-theoretic hardness in Section 6.

Lemma 2.17 (Folklore). *Given a concept class \mathcal{C} such that $|\mathcal{C}|$ is finite, then for all $\eta > 0$, $\text{fat}_{\mathcal{C}}(\eta) \leq \log_2|\mathcal{C}|$.*

Proof. Assume for the sake of contradiction that \mathcal{C} η -fat-shatters the set of points $\{x_1, x_2, \dots, x_m\}$ for $m > \log_2|\mathcal{C}|$. Then \mathcal{C} must be able to properly match each $b \in \{\pm 1\}^m$. There are $2^m > |\mathcal{C}|$ possible b vectors, and only one $f \in \mathcal{C}$ can be used per shattering attempt, since no f can ever satisfy two different b vectors. This is a contradiction since we don't have enough $f \in \mathcal{C}$ to go around to satisfy every b vector. \square

2.5.2 Decision Problems

One can also define the decision version of the consistency problem, which is deciding if there even exists an $h \in \mathcal{C}$ that is consistent with all of S . We show that the existence of efficient learning algorithms can imply efficient one-sided error algorithms for the decision version of the consistency problem.

Definition 2.18. Let $\text{CONSISTENTDECIDE}(\mathcal{C}, s)$ be decision version of the consistency problem for \mathcal{C} using at most s samples.

Proposition 2.19. *An efficient randomized $(\epsilon < \frac{\alpha^2}{s^2}, \delta < \frac{1}{2} + \frac{1}{2s})^2$ proper learning algorithm implies $\text{CONSISTENTDECIDE}(\mathcal{C}, s) \in \text{RP}$ where $\alpha = \inf_{x \in \Omega, f(x) \neq g(x)} |f(x) - g(x)|$ is the minimum non-zero error any hypothesis function can make on a single input.*

Proof. For every set of samples S such that $|S| \leq s$, we can define the \mathcal{D}_S to be the uniform distribution over all $x \in \chi$ such that $(x, f(x)) \in S$. By coupon collector, if we draw $\mathcal{O}(s \log s)$ many samples then with probability at least $1 - \frac{1}{s}$ we will have drawn every item from S . Now imagine that there exists some hypothesis $h \in \mathcal{C}$ that is not consistent with S . Then our error must be at least $\frac{\alpha^2}{s^2}$ by the definition of α .

Now assume we have some efficient randomized (ϵ, δ) proper learning algorithm for $\epsilon < \frac{\alpha^2}{s^2}$ and $\delta < \frac{1}{2} + \frac{1}{2s}$. When running the learner on an arbitrary \mathcal{D}_S , it will see samples S with probability at least $1 - \frac{1}{s}$. To get error less than $\frac{\alpha^2}{s^2}$ the learner must then be able to solve the search version consistency problem with probability p such that $\frac{1}{s} + (1 - \frac{1}{s})p \geq 1 - \delta$. Solving for p we find $p \geq \frac{1}{2}$ on accepting instances.

²We abuse notation to signify that ϵ is a value less than $\frac{\alpha^2}{s^2}$ and likewise for $\delta < \frac{1}{2s}$.

This gives rise to the following algorithm in RP for solving $\text{CONSISTENTDECIDE}(\mathcal{C}, s)$. Given samples S with $|S| \leq s$, we can run our learning algorithm and pretend that S is what we sampled from \mathcal{D}_S to get hypothesis h . If h is consistent with S then accept, otherwise reject. On an accepting instance h will be consistent with probability at least $\frac{1}{2}$ while on rejecting instances it will never be consistent so the algorithm will always reject. \square

Informally, if there exists enough structure on the concept class, it can be possible to go the other way and show that an efficient algorithm for $\text{CONSISTENTDECIDE}(\mathcal{C}, s)$ implies an efficient proper learner for \mathcal{C} . Namely, if a search-to-decision reduction exists for the consistency problem on \mathcal{C} and $\text{fat}_{\mathcal{C}}$ is finite then we can also expect to show that an efficient algorithm for the decision problem would imply an efficient proper learner for \mathcal{C} . Of particular interest are NP-complete problems, which always admit search-to-decision reductions [32]. We can now give a formal proof of this commonly used technique to show proper PAC learning if $\text{RP} = \text{NP}$.

Lemma 2.20. *Let \mathcal{C} be a concept class and let*

$$m = \Theta\left(\frac{1}{\epsilon} \left(\text{fat}_{\mathcal{C}}\left(\frac{\beta}{8}\right) \log^2\left(\frac{\text{fat}_{\mathcal{C}}\left(\frac{\beta}{8}\right)}{\beta\epsilon}\right) + \log\frac{1}{\delta}\right)\right)$$

be the parameter from Theorem 2.16 with $\alpha = 0$. If $\text{CONSISTENTDECIDE}(\mathcal{C}, s)$ for $s > m$ is NP-complete and $\text{RP} = \text{NP}$ then for $c = \mathcal{O}(\text{poly}(n))$ there exists an efficient algorithm to $\left((1 - \epsilon)\beta^2 + \epsilon, \delta + \frac{1}{2^c}\right)$ proper learn \mathcal{C} .

Proof. Because search-to-decision reductions exist for all NP-complete problems [32], a zero-error oracle for $\text{CONSISTENTDECIDE}(\mathcal{C}, s)$ can be used to efficiently solve $\text{CONSISTENTSEARCH}(\mathcal{C}, s)$. Let us run our algorithm for $\text{CONSISTENTSEARCH}(\mathcal{C}, s)$ on a sample S such that $s \geq |S| \geq m$. We now have an $h \in \mathcal{C}$ such that

$$|h(x_i) - f(x_i)| = 0 \quad \forall 1 \leq i \leq k$$

and so by Theorem 2.16

$$\mathbb{E}_{x \sim \mathcal{D}} \left[(h(x) - f(x))^2 \right] \leq (1 - \epsilon)\beta^2 + \epsilon$$

with probability at most δ over the samples.

Finally, let γ be the number of calls to $\text{CONSISTENTDECIDE}(\mathcal{C}, s)$ used in the search-to-decision reduction. In order for the reduction to be efficient, $\gamma = \mathcal{O}(\text{poly}(n))$. Since $\text{CONSISTENTDECIDE}(\mathcal{C}, s)$ is in NP and therefore RP, we have an efficient one-sided constant error algorithm \mathcal{A} for $\text{CONSISTENTDECIDE}(\mathcal{C}, s)$. Using $\mathcal{O}(c + \log \gamma) = \mathcal{O}(\text{poly}(n))$ many calls to \mathcal{A} and taking the majority, we can get error at most $\frac{1}{\gamma \cdot 2^c}$. Call this new algorithm \mathcal{A}' and use it in place of the zero-error oracle for $\text{CONSISTENTDECIDE}(\mathcal{C}, s)$. By the union bound over all γ calls to \mathcal{A}' , the probability that any query to \mathcal{A}' differs from the zero-error oracle is at most $\frac{1}{2^c}$.

By the union bound over both the samples and the error in \mathcal{A}' , the total error probability is at most $\delta + \frac{1}{2^c}$. \square

3 PAC Learning Applied to Clifford Circuits

Because of the works of Rocchetto [44] and Lai and Cheng [38], Clifford circuits are a prime candidate for an efficiently PAC-learnable class of circuits. We give a very loose bound on the fat-shattering dimension of Clifford circuits that is sufficient for our purposes.

Lemma 3.1. *Let \mathcal{C} be the set of Clifford circuits. For all $\eta > 0$, $\text{fat}_{\mathcal{C}}(\eta) \leq \mathcal{O}(n^2)$.*

Proof. By Proposition 2.6 there are at most $2^{\mathcal{O}(n^2)}$ Clifford circuits. By Lemma 2.17, the fat-shattering dimension for all $\eta > 0$ is at most $\log_2(2^{\mathcal{O}(n^2)}) = \mathcal{O}(n^2)$. \square

Because CNOT circuits are a subset of Clifford circuits, we can also upper-bound the fat-shattering dimension of CNOT circuits by $\mathcal{O}(n^2)$.

3.1 Consistency Problem of Clifford Circuits

We now turn to the consistency problem. Noting that each Pauli matrix is Hermitian, a very natural way to measure a stabilizer state is in a product basis where we measure each qubit with respect to a Pauli.

Definition 3.2. If $P \in \mathcal{P}_n$ is a Pauli operator, then the two-outcome measurement associated with P is $\frac{I^{\otimes n} + P}{2}$, and is referred to as a PAULI MEASUREMENT.

Definition 3.3. Let the problem of PAC learning Clifford circuits with respect to Pauli measurements be defined as follows. Let C be an unknown Clifford circuit and let \mathcal{D} be an unknown joint distribution over both stabilizer states and Pauli measurements. Finally, let samples to C be given as

$$\left(\rho, E, \text{Tr}\left[EC\rho C^\dagger\right]\right)$$

where $\rho, E \sim \mathcal{D}$ are a stabilizer state and Pauli measurement jointly drawn from \mathcal{D} and represented as classical bit strings using the stabilizer formalism. The goal is to then learn the measurements $\text{Tr}\left[EC\rho C^\dagger\right]$ up to error ϵ under the distribution \mathcal{D} .

A critical part of Rocchetto [44] was noting that the measurement results with Pauli measurements could only have three distinct values:

Lemma 3.4 (Rocchetto [44] Lemma 1). *Let $E^P = \frac{I^{\otimes n} + P}{2}$ be a Pauli measurement associated to a Pauli operator $P \in \mathcal{P}_n$ and ρ be an n -qubit stabiliser state. Then $\text{Tr}\left[E^P C\rho C^\dagger\right]$ can only take on the values $\left\{0, \frac{1}{2}, 1\right\}$, and:*

$$\begin{cases} \text{Tr}\left[E^P C\rho C^\dagger\right] = 1 \text{ iff } P \text{ is a stabilizer of } C\rho C^\dagger; \\ \text{Tr}\left[E^P C\rho C^\dagger\right] = 1/2 \text{ iff neither } P \text{ nor } -P \text{ is a stabilizer of } C\rho C^\dagger; \\ \text{Tr}\left[E^P C\rho C^\dagger\right] = 0 \text{ iff } -P \text{ is a stabilizer of } C\rho C^\dagger. \end{cases}$$

What information does a single sample tell us? Let G_i be the stabilizer group of ρ_i . From this, we can gather that if $\text{Tr}\left[E^P C\rho_i C^\dagger\right] = 1$ then $C^\dagger P C \in G_i$, and if $\text{Tr}\left[E^P C\rho_i C^\dagger\right] = 0$ then $C^\dagger P C \in -G_i$ where $-G_i = \{-P \mid P \in G_i\}$. Finally, if $\text{Tr}\left[E^P C\rho_i C^\dagger\right] = \frac{1}{2}$ then $C^\dagger P C$ is in the complement of $G_i \cup -G_i$.

If the measurement E^P appears multiple times across multiple samples, we can gather further information. For instance, have

$$S_P = \left\{ \left(\rho_i, E^P, \text{Tr}\left[E^P C\rho_i C^\dagger\right]\right) \right\}$$

be the set of all samples such that E^P is the measurement taken and let G_i be the stabilizer group of each ρ_i . Based on each label $\text{Tr}[E^P C \rho_i C^\dagger]$, we know that $C^\dagger P C$ must lie in H_i , which is one of G_i , $-G_i$ or $G_i \cup -G_i$. We then deduce that $C^\dagger P C$ must lie in $\bigcap_i H_i$. To actually be a Clifford circuit, we must also add the constraint that $C^\dagger P C \neq I^{\otimes n}$, giving us

$$C^\dagger P C \in \left(\bigcap_i H_i \right) \setminus \{I^{\otimes n}\}.$$

The problem of finding a Clifford circuit with zero training error then reduces to the search problem of finding a set of $\alpha, \beta, \gamma, \theta, p, q$ from Eq. (1) representing a C^\dagger that is consistent with all of these constraints while remaining symplectic according to Eq. (2). Let \mathcal{C} be the set of Clifford circuits. We will call this problem $\text{CLIFFORDSEARCH}(s) = \text{CONSISTENTSEARCH}(\mathcal{C}, s)$ ³.

Due to Gottesman-Knill [3, 27] showing that Clifford circuits are classically simulable, the act of verifying that we have a circuit that has zero training error is efficient, meaning that the decision version $\text{CLIFFORDDECIDE}(s) = \text{CONSISTENTDECIDE}(\mathcal{C}, s)$ of the problem is in NP.

Proposition 3.5. *The decision problem, $\text{CLIFFORDDECIDE}(\text{poly}(n))$, of deciding if there exists a Clifford circuit consistent with polynomially sized sample S is in NP.*

Proof. Given a set of $\alpha, \beta, \gamma, \theta, p, q$, it is easy to check that the γ, θ, p, q form a symplectic matrix by checking with respect to $\Lambda(n)$. Checking that they are consistent with the samples in S can be done by iterating through S since the trace can be computed efficiently using Gottesman-Knill [3]. \square

Knowing this, we find that $\text{CLIFFORDSEARCH}(\text{poly}(n)) \in \text{FNP}$. This property extends to the analogous problems for CNOT circuits, $\text{CNOTSEARCH}(\text{poly}(n))$ and $\text{CNOTDECIDE}(\text{poly}(n))$, since one can also efficiently verify that $\gamma_{ij} = 0$ and $p_j = 0$ for all i and j .

4 Generating Samples with Certain Constraints

We will now show how we can use samples from PAC learning to generate certain kinds of constraints. It will suffice to only consider CNOT circuits with computational basis state measurements and measurements of the form $\{I, Z\}^{\otimes n}$. The net effect of this is that from a PAC learning standpoint, for unknown CNOT circuit C we only need to figure a set of $C^\dagger Z_i C$ that is consistent with the samples as described in Section 3. Since we will never be tested on a measurement with some component of X_i involved, this is equivalent to finding the θ_{ij} and q_j values from Eq. (1) of C^\dagger . We will again choose to view the θ_{ij} as the matrix Θ , such that Θ must be full rank.

Definition 4.1. Given a set of abelian generators $\{P_i\}$, let

$$\rho(P_1, P_2, \dots, P_n) = \frac{1}{2^n} \sum_{P \in \langle P_1, \dots, P_n \rangle} P$$

be the stabilizer state that is formed from that stabilizer group.

³We stress that via Definition 3.3, the input states are assumed to be stabilizer states.

The following observation will notationally make the proceeding theorem states and proofs easier to follow.

Observation 4.2. *Any one-dimensional affine subspace $v + \langle w \rangle$ can be represented as $\{v, v + w\}$ and any set of two vectors/matrices $\{v, w\}$ represents the one-dimensional affine subspace $v + \langle v + w \rangle$. Thus we can freely move between the two representations.*

Lemma 4.3. *Let C be a CNOT circuit on n qubits and have $\{v, v + w\} \subset \{0, 1\}^n$ be a one-dimensional affine subspace of column vectors such that $v \neq w$ and $w, v \neq 0$. Given an arbitrary pauli P there exists a set of n samples that constrains $C^\dagger PC$ to only have consistent solutions lying in $\{Z^v, Z^{v+w}\}$. Furthermore these n samples can be efficiently generated.*

Proof. Let (v, w, v_3, \dots, v_n) be an arbitrary basis for $\{0, 1\}^n$ containing v and w . This can be found with $\mathcal{O}(n)$ random vectors and the use of Gaussian elimination. Recalling Definition 2.3, let us start by creating the sample

$$\left(\left(\rho(Z^v, Z^w, Z^{v_3}, Z^{v_4}, \dots, Z^{v_n}), \frac{I^{\otimes n} + P}{2} \right), 1 \right),$$

which limits $C^\dagger PC$ to be in $\{I, Z\}^{\otimes n}$ with positive phase. We can create the set of samples:

$$\begin{aligned} & \left(\left(\rho(Z^v, Z^w, -Z^{v_3}, Z^{v_4}, \dots, Z^{v_n}), \frac{I^{\otimes n} + P}{2} \right), 1 \right), \\ & \left(\left(\rho(Z^v, Z^w, Z^{v_3}, -Z^{v_4}, \dots, Z^{v_n}), \frac{I^{\otimes n} + P}{2} \right), 1 \right), \\ & \quad \vdots \\ & \left(\left(\rho(Z^v, Z^w, Z^{v_3}, Z^{v_4}, \dots, -Z^{v_n}), \frac{I^{\otimes n} + P}{2} \right), 1 \right). \end{aligned}$$

By construction $C^\dagger PC$ cannot have any component of Z^{v_3} because of the first sample of this set, nor any Z^{v_i} for $i > 3$ due to the remaining samples. This leaves $C^\dagger PC$ to be one of Z^v, Z^w , or Z^{v+w} (since it cannot be identity). To remedy this, we can introduce the final sample:

$$\left(\left(\rho(-Z^v, Z^w, Z^{v_3}, Z^{v_4}, \dots, Z^{v_n}), \frac{I^{\otimes n} + P}{2} \right), 0 \right),$$

which then eliminates Z^w (and identity, due to the negative sign). The total number of samples is n and the whole process takes polynomial in n time to find the basis and create said samples. \square

We can easily extend this to the 0-dimensional case by simply treating w as v_2 , using an extra sample to remove the last dimension. More importantly, let's say we've constrained $C^\dagger Z^x C$ to lie in $\{Z^v, Z^{v+w}\}$. The effect of this on Θ is that if we sum the columns i where $x_i = 1$ then the sum must lie in $\{v, v + w\}$.

Corollary 4.4. *Let*

$$\{v, v + w\} = \left\{ \left[\begin{array}{c|c|c|c} | & | & | & | \\ \hline v_1 & v_2 & \dots & v_k \\ \hline | & | & | & | \end{array} \right], \left[\begin{array}{c|c|c|c} | & | & | & | \\ \hline v_1 + w_1 & v_2 + w_2 & \dots & v_k + w_k \\ \hline | & | & | & | \end{array} \right] \right\}$$

be a one-dimensional affine subspace of $n \times k$ matrices over $\{0, 1\}$ such that for all i , $v_i \neq w_i$ and $v_i, w_i \neq 0$. Finally, let Θ' be an arbitrary $n \times k$ submatrix of Θ . Then there exists a set of $(2k - 1)n$ samples that constrain Θ' to only have consistent solutions lying in $\{v, v + w\}$ for CNOT circuit C . Furthermore these samples can be efficiently generated.

Proof. WLOG, we will let the set of k different columns we choose for Θ' to be columns 1 through k . We will use induction on k to prove this corollary, with the base case covered by Lemma 4.3. Now let us assume that we have samples that constrain columns 2 through k to be either

$$\left[\begin{array}{c|c|c|c} | & | & | & | \\ v_2 & v_3 & \dots & v_k \\ | & | & | & | \end{array} \right] \text{ or } \left[\begin{array}{c|c|c|c} | & | & | & | \\ v_2 + w_2 & v_2 + w_3 & \dots & v_k + w_k \\ | & | & | & | \end{array} \right].$$

The goal will be to generate constraints such that if column 2 is v_2 then column 1 must be v_1 . Otherwise, if column 2 is $v_2 + w_2$ then column 1 is constrained to be $v_1 + w_1$.

To start us off, we can use Lemma 4.3 to constrain the sum of columns 1 and 2 to be either $v_1 + w_1$ or $v_1 + w_1 + v_2 + w_2$. If we focus on columns 1 and 2, the solutions to this constraint lie in an affine subspace defined by:

$$\left[\begin{array}{c|c} | & | \\ v_1 + v_2 + u & u \\ | & | \end{array} \right] \text{ or } \left[\begin{array}{c|c} | & | \\ v_1 + w_1 + v_2 + w_2 + u & u \\ | & | \end{array} \right]$$

for arbitrary vector u . We then apply Lemma 4.3 again to constrain column 1 to be either v_1 or $v_1 + w_1$. Thus the first two columns must either be

$$\left[\begin{array}{c|c} | & | \\ v_1 & v_2 \\ | & | \end{array} \right] \text{ or } \left[\begin{array}{c|c} | & | \\ v_1 + w_1 & v_2 + w_2 \\ | & | \end{array} \right]$$

Finally, to lie in the intersection from the inductive hypothesis, we note that if the second column is v_2 or $v_2 + w_2$ then columns 3 through k must be

$$\left[\begin{array}{c|c|c|c} | & | & | & | \\ v_3 & v_4 & \dots & v_k \\ | & | & | & | \end{array} \right] \text{ or } \left[\begin{array}{c|c|c|c} | & | & | & | \\ v_3 + w_3 & v_4 + w_4 & \dots & v_k + w_k \\ | & | & | & | \end{array} \right].$$

respectively.

Collectively, we achieve our goal of constraining the entire solution to lie in $v + \langle w \rangle$. We used n samples at the first step and $2n$ for every inductive step after (one set of n samples for each call of Lemma 4.3), giving us a total number of samples of $2n(k - 1) + n = 2kn - 2n + n = (2k - 1)n$. Since each step was efficient, the whole process takes polynomial in n time to generate all of the samples. \square

5 On the NP-completeness of NONSINGULARITY

Definition 5.1. Given $n \times n$ matrices M_0, M_1, \dots, M_m over some field \mathbb{F} , NONSINGULARITY is the problem of deciding if there exists $\alpha_1, \alpha_2, \dots, \alpha_m \in \mathbb{F}$ such that $M_0 + \sum_i \alpha_i M_i$ results in a non-singular matrix.

Theorem 5.2 (Buss et al. [17] Corollary 10). *NONSINGULARITY over \mathbb{F}_2 is NP-complete.*

Formula F	The series-parallel s - t graph G_F with edge weights
Constant c	
Variable x	
$F = F_1 \cdot F_2$	
$F = F_1 + F_2$	

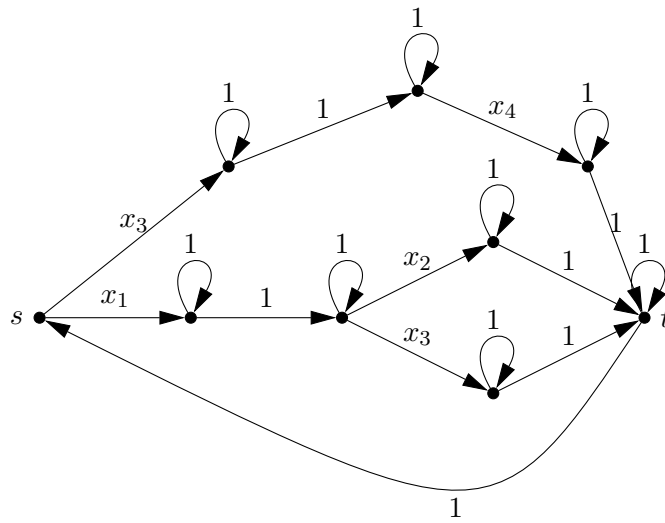
Figure 1: Inductive Construction from Formula to Graph

The high level idea of the proof is to first reduce a 3SAT instance over variables $\{x_i\}$ to solving an arithmetic formula F . The formula is then turned into a weighted directed graph whose adjacency matrix $M(x)$ has a determinant that is equal to the formula F , where $M(x)$ has entries from $\{0, 1\} \cup \{x_i\}$, and can thus be viewed as an affine subspace over $\mathbb{F}_2^{(|F|+2) \times (|F|+2)}$.

While we will not prove the correctness of this statement, we will want to ascertain exactly what kind of M_i are formed through the reduction. We now describe the construction of the graph (see Fig. 1 and Fig. 2 for relevant illustrations):

- For each atomic formula F' , create vertices s and t .
For each constant c create a unique node v_c with a path from s to v_c with weight c and a path from v_c to t with weight 1.
For each variable x_i create a unique node v_{x_i} with a path from s to v_{x_i} with weight x_i and a path from v_{x_i} to t with weight 1.
- For multiplication of F_i and F_j , place the graphs of F_i and F_j in series.
- For addition of F_i and F_j , place the graphs of F_i and F_j in parallel.

$$F : x_1(x_2 + x_3) + x_3 \cdot x_4$$



$$\begin{pmatrix} 0 & x_1 & 0 & 0 & 0 & x_3 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & x_2 & x_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 2: Example of constructing the adjacency matrix with a specific determinant

- Once all of this is done, create a path of weight 1 from the global t vertex to the global s vertex.
- Create self loops at every vertex besides the global s vertex.

Let M be the resulting adjacency matrix of this graph. For every entry that is a constant, we can assign that to M_0 . Then for each variable x_i , we can set M_i to be the matrix that is zero everywhere except where x_i appears in M . As an example, for a matrix

$$M(x) = \begin{pmatrix} x_1 & 1 \\ x_2 & x_1 \end{pmatrix}$$

we can describe it using $M(x) = M_0 + x_1M_1 + x_2M_2$ where

$$M_0 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad M_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad M_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

For a more succinct reduction later in Section 6, we want to isolate the kinds of matrix affine subspaces over \mathbb{F}_2 that are hard to solve (i.e., are used in the reduction from 3SAT). The following notation will be beneficial for that.

Definition 5.3. For a $n \times n$ matrix M over a field \mathbb{F} , let $NZ(M) \subseteq \{1, 2, \dots, n\}$ be the columns of M that are non-zero (i.e., are not the all zeros vector).

Definition 5.4. Let M and W be $n \times n$ matrices over a field \mathbb{F} and let $k := |NZ(W)|$ such that $NZ(W) = \{i_1, i_2, \dots, i_k\}$. If

$$M = \begin{bmatrix} | & | & | & | \\ m_1 & m_2 & \dots & m_n \\ | & | & | & | \end{bmatrix}$$

then the restriction of the columns of M to the non-zero columns of W is defined as

$$R_W(M) = \begin{bmatrix} | & | & | & | \\ m_{i_1} & m_{i_2} & \dots & m_{i_k} \\ | & | & | & | \end{bmatrix}.$$

Likewise, for a set of matrices S , $R_W(S) = \{R_W(M) \mid M \in S\}$.

We now define the modified version of NONSINGULARITY and show that it is still NP-complete.

Definition 5.5. Let $M = M_0 + \langle M_1, M_2, \dots, M_m \rangle$ be an matrix affine subspace of $n \times n$ matrices over some field \mathbb{F} . In addition, require the M_i have the property that $NZ(M_i) \cap NZ(M_j) = \emptyset$ for all $i > 0$ and $j > 0$. Finally, for all $i > 0$ with $k_i := |NZ(M_i)|$, constrain M_0 and M_i such that the restriction of M to the non-zero columns of M_i ,

$$R_{M_i}(M) = R_{M_i}(\{M_0, M_0 + M_i\}),$$

can be represented as:

$$\left\{ \left[\begin{array}{c|c|c|c|} | & | & | & | \\ v_1 & v_2 & \dots & v_{k_i} \\ | & | & | & | \end{array} \right], \left[\begin{array}{c|c|c|c|} | & | & | & | \\ v_1 + w_1 & v_2 + w_2 & \dots & v_{k_i} + w_{k_i} \\ | & | & | & | \end{array} \right] \right\}$$

for some v and w such that $v_j \neq w_j$ and $v_j, w_j \neq 0$. The problem of deciding if there exists $\alpha_1, \alpha_2, \dots, \alpha_m \in \mathbb{F}$ such that $M_0 + \sum_i \alpha_i M_i$ results in a non-singular matrix will be called the MODIFIED-NONSINGULARITY problem.

Corollary 5.6. *The MODIFIED-NONSINGULARITY problem is NP-complete over \mathbb{F}_2 .*

Proof. Rather than solve NONSINGULARITY using MODIFIED-NONSINGULARITY, we merely show that the reduction from 3SAT naturally leads to instances of MODIFIED-NONSINGULARITY.

Let G be the graph produced by the reduction from 3SAT with adjacency matrix $M(x)$. By the construction of $M(x)$ given by Fig. 1, we see that every instance of a variable will create its own unique subgraph such that the instance of each variable connects to a unique vertex. Because this unique vertex can never be used as an s or t vertex from Fig. 1, that vertex also necessarily has in-degree 1 so that no other edges connect to it. Because these are the only vertices that have incoming edges assigned with variable weight, this conversely means that the columns of $M(x)$ contain at most one variable.

Recall that for variable x_i with $i > 0$, the matrix M_i we form from the decomposition of $M(x)$ is the entries associated with x_i . Since each column only contains at most one x_i , then $NZ(M_i) \cap NZ(M_j)$ must be disjoint for $i > 0$ and $j > 0$. Due to the constraint being met, we know that the $R_{M_i}(M)$ is confined to a one-dimensional affine subspace. As such, it can be represented as:

$$\left\{ \left[\begin{array}{c|c|c|c} | & | & | & | \\ \hline v_1 & v_2 & \dots & v_{k_i} \\ \hline | & | & | & | \end{array} \right], \left[\begin{array}{c|c|c|c} | & | & | & | \\ \hline v_1 + w_1 & v_2 + w_2 & \dots & v_{k_i} + w_{k_i} \\ \hline | & | & | & | \end{array} \right] \right\}$$

for some v and w , where the v is a sub-matrix of M_0 and w is the corresponding sub-matrix for M_i . Due to the definition of $NZ(M_i)$ it is clear that the w_j are non-zero. Furthermore, since the weight of an edge will never be $x_i + 1$ then an entry of w_j being one implies the corresponding entry of v_j is zero. This implies $v_j \neq w_j$ for all j . Finally, to ensure that $v_j \neq 0$, we note that each vertex besides s receives a self loop with weight 1. s instead receives an edge from t with weight 1. These self loops and the edge from t to s ensures that each column of M_0 has at least entry with 1 in it such that the $v_j \neq 0$.

We have now shown that every matrix affine subspace produced from the reduction in Buss et al. [17] also meets the requirements for MODIFIED-NONSINGULARITY, thus showing that this problem is also NP-complete. \square

6 PAC Learning CNOT Circuits and NP

Lemma 6.1. *The decision problem, CNOTDECIDE($2n^2$), of deciding whether or not there exists a CNOT circuit consistent with at most $2n^2$ samples is NP-complete.*

Proof. By Corollary 5.6 and the Cook-Levin theorem [34], if a problem can be used to solve MODIFIED-NONSINGULARITY then it is NP-hard. Looking at each individual M_i for $i > 0$ from Definition 5.5, they are all non-zero on disjoint columns. If we restrict to the non-zero columns of a particular M_i for $i > 0$, we find that $R_{M_i}(M)$ meet the requirements for Corollary 4.4. Since the M_i act on disjoint columns, if we apply Corollary 4.4 for each $R_{M_i}(M_i)$ then we have efficiently created a set of samples S_i that restricts those columns of Θ , the matrix form of the θ_{ij} values, to lie in $R_{M_i}(\{M_0, M_0 + M_i\})$. To fix the columns not touched by the M_i , it is not hard to show that the ideas of Lemma 4.3 can also create a 0-dimensional affine space over these columns, thereby setting the whole matrix to lie in $M_0 + \langle M_1, M_2, \dots, M_m \rangle$. Let such samples be called T . Altogether, by taking $S = T \cup (\bigcup_i S_i)$ we are able to use Lemma 4.3 and Corollary 4.4 to set Θ to lie in $M(x)$ from MODIFIED-NONSINGULARITY.

If $M(x)$ is an accepting instance of MODIFIED-SINGULARITY then there must be a full rank Θ that is consistent with S . Since Lemma 2.10 ensures us that CNOT circuits can instantiate any full rank Θ , there must also exist a CNOT circuit consistent with S . Alternatively, if $M(x)$ does not contain a non-singular matrix, then there does not exist a full rank Θ that is consistent with S . By Lemma 2.9 there cannot exist a CNOT circuit that is consistent with the data. This gives us that $M(x)$ contains a non-singular matrix if and only if there is a CNOT circuit consistent S , which can be produced efficiently.

We now count the number of samples used. For each all $1 \leq i \leq m$, we use n samples to constrain the first column of $NZ(M_i)$. For the remaining columns, we either use $2n$ if that column is contained in some $NZ(M_j)$, otherwise, we use $n + 1$ samples from the generalization of Lemma 4.3. Since $2n \geq n + 1$ for $n \geq 1$, we use at most $2n(n - m)$ samples for the remainder, giving us at most $2n(n - m) + nm = 2n^2 - mn \leq 2n^2$ samples. This shows CNOTDECIDE($2n^2$) is NP-hard. Combined with Proposition 3.5 we find that CNOTDECIDE($2n^2$) to be NP-complete. \square

Since CNOTDECIDE($2n^2$) \subseteq CNOTDECIDE(poly(n)) then CNOTDECIDE(poly(n)) is also NP-hard.

Theorem 6.2 (Formal Statement of Theorem 1.1). *There exists an efficient randomized (ϵ, δ) proper PAC learner for CNOT circuits with arbitrary ϵ and δ as arbitrary $\frac{1}{\text{poly}(n)}$ values if and only if $\text{RP} = \text{NP}$. Furthermore, an efficient quantum (ϵ, δ) proper PAC learner for CNOT circuits with arbitrary ϵ and δ as arbitrary $\frac{1}{\text{poly}(n)}$ values exists if and only if $\text{NP} \subseteq \text{RQP}$.*

Proof. We will start by proving the $\text{NP} \subseteq \text{RP}$ version for classical randomized learners. The quantum version will follow trivially by replacing the learner with a quantum algorithm and therefore RP with RQP. The only change then is that $\text{NP} \subseteq \text{RQP}$ does not necessary imply $\text{NP} = \text{RQP}$ like it does with RP.

Because CNOT circuits with classical inputs and measurements only has labels 0 and 1, the smallest non-zero error is 1. By Proposition 2.19 with $\alpha = 1$, an efficient

$$\left(\epsilon < \frac{1}{4n^4}, \delta < \frac{1}{2} + \frac{1}{4n^2} \right)^4$$

randomized proper learner for CNOT circuits will imply CNOTDECIDE($2n^2$) \in RP. Since CNOTDECIDE($2n^2$) is NP-complete by Lemma 6.1, efficient randomized learners for CNOT circuits with arbitrary $\epsilon = \frac{1}{\text{poly}(n)}$ and $\delta = \frac{1}{\text{poly}(n)}$ are only possible if $\text{NP} \subseteq \text{RP}$.

Conversely, by Lemma 3.1 and Lemma 2.20 with $\beta = \frac{1}{2}$, $c = \text{poly}(n)$, $\epsilon = \frac{4\epsilon' - 1}{3}$, and $\delta = \delta' - 2^{-c}$ if $\text{NP} \subseteq \text{RP}$ then there must exist an efficient (ϵ', δ') proper learner for CNOT circuits as long as the number of samples m is polynomial in n . For arbitrary $\epsilon' = \frac{1}{\text{poly}(n)}$ and $\delta' = \frac{1}{\text{poly}(n)}$ our required number of samples becomes

$$m = \Theta \left(\frac{1}{\epsilon} \left(\text{fatc} \left(\frac{\beta}{8} \right) \log^2 \left(\frac{\text{fatc} \left(\frac{\beta}{8} \right)}{\beta \epsilon} \right) + \log \frac{1}{\delta} \right) \right) = \mathcal{O}(\text{poly}(n))$$

This is sufficiently small as desired, completing the proof. \square

Corollary 6.3 (Formal Statement of Corollary 1.2). *There exists an efficient randomized (ϵ, δ) proper PAC learner for Clifford circuits with arbitrary ϵ and δ as arbitrary $\frac{1}{\text{poly}(n)}$*

⁴We again abuse notation to signify that ϵ is a value less than $\frac{1}{4n^4}$ and likewise for $\delta < \frac{1}{2} + \frac{1}{4n^2}$.

values if and only if $RP = NP$. Furthermore, an efficient quantum (ϵ, δ) proper PAC learner for Clifford circuits with arbitrary ϵ and δ as arbitrary $\frac{1}{\text{poly}(n)}$ values exists if and only if $NP \subseteq RQP$.

Proof. Since CNOT circuits are also a form of Clifford circuits, $\text{CNOTDECIDE}(2n^2) \subset \text{CLIFFORDDECIDE}(2n^2)$, and so $\text{CLIFFORDDECIDE}(2n^2)$ is also NP-hard. Combined with Proposition 3.5, we get that it is NP-complete. The proof of Theorem 6.2 continues but with $\alpha = \frac{1}{2}$ instead due to Lemma 3.4. This leads to slightly different constants, but the proof ideas all follow without major change. \square

7 Discussion and Open Problems

In this work, we prove a negative result in proper learning of one of the best candidates for efficient PAC learning of quantum circuits. However, it should be noted that in many cases there exist improper learners even in the case where proper learning is NP-hard, such as 2-clause CNF, 3-DNF, and intersection of half spaces [14, 15, 30]. This immediately leaves the problem of whether or not an improper learner exists for Clifford circuits. One way of showing hardness would be to leverage cryptographic hardness such as in Arunachalam et al. [9], Kharitonov [33]. Another approach would be to assume the hardness of random k-DNF, such as the work of Daniely and Shalev-Shwartz [23]. For upper bounds, the work of Caro and Datta [19] can also be used to get agnostic generalization results, providing a possible pathway to answering research questions in that direction.

Another thing worth considering is that the hardness results only apply for small errors (roughly $1/\text{poly}(n)$). And while this is sufficient to give complexity-theoretic hardness for the kinds of PAC learners (i.e., strong proper learners) originally introduced by Valiant [47], it would be nice to get hardness results for larger errors as in the work by Guruswami and Raghavendra [28]. This work involved using PCP/hardness-of-approximation ideas to show that even constant training error was NP-hard.

We also note that a single output bit of a CNOT circuit is simply an XOR, which is easy to learn efficiently if there is no noise. However, because we are dealing with reversible computation each output bit has to be a linearly independent XOR such that each input bit is recoverable. Finding a CNOT circuit that matches a single XOR function f can be done by sampling expected $\mathcal{O}(n)$ random XOR until we get n linearly independent XOR with one of them being f (see Appendix B.1). Thus, the entire difficulty of proper learning CNOT circuits is this linear independence of the output bits. As such, even though $\text{AC}^0 \subseteq \text{TC}^0 \subseteq \text{NC}^1 \subseteq \text{L} \subseteq \oplus\text{L}$ with the lower classes having improper hardness results based on cryptographic hardness [33], one cannot directly give an improper learning result for CNOT circuits despite the fact that simulating CNOT circuits is complete for $\oplus\text{L}$.

As noted previously, our PAC learning framework is slightly different from that of Caro and Datta [19], in that we use Pauli matrices, rather than rank 1 projectors as measurements. To the author's knowledge, there exists no proof showing that one framework is necessarily harder than the other. The author also do not see an obvious way of proving an analogous hardness theorem in the specific framework of Caro and Datta [19] for Clifford or CNOT circuits.

Finally, with everything from the input states to the circuits involved being classical, it is entirely possible to prove the technical results about CNOT circuits only talking about bit strings and parity functions. Namely, one can replace the entire problem with samples of the form $(x, s, s^T C x \pmod{2})$ where $(x, s) \sim \mathcal{D}$ are n -bit strings and C is a CNOT circuit. Since the stabilizer group of a computational basis state always lies in $\{\pm 1\} \times \{I, Z\}^{\otimes n}$,

we can uniquely define it by the subgroup that has positive phase. This is equivalent to the orthogonal complement of x , which is the subspace $W_x = \{x \in \{0, 1\}^n \mid w \cdot x = 0\}$. From there, a sample of the form $(x, s, 0)$ simply says that $Cx \in W_s$, and one can get an analogous proof by copying the lemmas and theorems in Section 4 and Section 6. However, this proof isn't anymore intuitive than the one given using stabilizer groups, and in fact is probably less intuitive to the average reader due to the lack of established formalism from stabilizers and paulis. It would be interesting if a more intuitive purely classical proof could be made to show hardness of learning CNOT circuits under this model.

Acknowledgements

The author would like to thank Aravind Gollakota for being a wealth of information with regard to statistical learning theory and William Kretschmer for insightful commentary and feedback. The author would also like to thank Patrick Rall, Justin Yirka, Sabee Grewal, Matthias Caro, and Scott Aaronson for useful discussions. The author was supported by Scott Aaronson's Paths to Quantum Supremacy award by the Office of Naval Research and Scott Aaronson's Simons Foundation It From Qubit award.

References

- [1] Scott Aaronson. The learnability of quantum states. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088):3089–3114, 2007. DOI: [10.1098/rspa.2007.0113](https://doi.org/10.1098/rspa.2007.0113). 2, 8
- [2] Scott Aaronson. Shadow tomography of quantum states. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 325–338, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355599. DOI: [10.1145/3188745.3188802](https://doi.org/10.1145/3188745.3188802). 1
- [3] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004. DOI: [10.1103/physreva.70.052328](https://doi.org/10.1103/physreva.70.052328). 2, 6, 12
- [4] J. B. Altepeter, D. Branning, E. Jeffrey, T. C. Wei, P. G. Kwiat, R. T. Thew, J. L. O'Brien, M. A. Nielsen, and A. G. White. Ancilla-assisted quantum process tomography. *Phys. Rev. Lett.*, 90:193601, May 2003. DOI: [10.1103/PhysRevLett.90.193601](https://doi.org/10.1103/PhysRevLett.90.193601). 1
- [5] Martin Anthony and Peter L. Bartlett. Function learning from interpolation. *Combinatorics, Probability and Computing*, 9(3):213–225, 2000. DOI: [10.1017/S0963548300004247](https://doi.org/10.1017/S0963548300004247). 8
- [6] Srinivasan Arunachalam and Ronald de Wolf. Guest column: A survey of quantum learning theory. *ACM SIGACT News*, 48(2):41–67, 2017. DOI: [10.1145/3106700.3106710](https://doi.org/10.1145/3106700.3106710). 4
- [7] Srinivasan Arunachalam and Ronald de Wolf. Optimal Quantum Sample Complexity of Learning Algorithms. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference (CCC 2017)*, volume 79 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 25:1–25:31, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-040-8. DOI: [10.4230/LIPIcs.CCC.2017.25](https://doi.org/10.4230/LIPIcs.CCC.2017.25). 4
- [8] Srinivasan Arunachalam, Alex B Grilo, and Henry Yuen. Quantum statistical query learning. *arXiv preprint arXiv:2002.08240*, 2020. DOI: <https://doi.org/10.48550/arXiv.2002.08240>. 4

- [9] Srinivasan Arunachalam, Alex Bredariol Grilo, and Aarthi Sundaram. Quantum hardness of learning shallow classical circuits. *SIAM Journal on Computing*, 50(3): 972–1013, 2021. DOI: [10.1137/20M1344202](https://doi.org/10.1137/20M1344202). 20
- [10] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*, 560:7–11, Dec 2014. ISSN 0304-3975. DOI: [10.1016/j.tcs.2014.05.025](https://doi.org/10.1016/j.tcs.2014.05.025). 2
- [11] Charles H. Bennett and Stephen J. Wiesner. Communication via one- and two-particle operators on einstein-podolsky-rosen states. *Phys. Rev. Lett.*, 69:2881–2884, Nov 1992. DOI: [10.1103/PhysRevLett.69.2881](https://doi.org/10.1103/PhysRevLett.69.2881).
- [12] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 70:1895–1899, Mar 1993. DOI: [10.1103/PhysRevLett.70.1895](https://doi.org/10.1103/PhysRevLett.70.1895).
- [13] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. DOI: [10.1137/S0097539796300921](https://doi.org/10.1137/S0097539796300921). 2
- [14] Avrim Blum. Computational Hardness of Learning. <http://www.cs.cmu.edu/~avrim/ML07/lect1007.pdf>, 2015. URL <http://www.cs.cmu.edu/~avrim/ML07/lect1007.pdf>. Lecture notes for CS 10-806 Foundations of Machine Learning and Data Science. 3, 20
- [15] Avrim L. Blum and Ronald L. Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992. ISSN 0893-6080. DOI: [https://doi.org/10.1016/S0893-6080\(05\)80010-3](https://doi.org/10.1016/S0893-6080(05)80010-3). 3, 20
- [16] Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, oct 1989. ISSN 0004-5411. DOI: [10.1145/76359.76371](https://doi.org/10.1145/76359.76371). 8
- [17] Jonathan F Buss, Gudmund S Frandsen, and Jeffrey O Shallit. The computational complexity of some problems of linear algebra. *Journal of Computer and System Sciences*, 58(3):572–596, 1999. ISSN 0022-0000. DOI: <https://doi.org/10.1006/jcss.1998.1608>. 3, 14, 18
- [18] Matthias C. Caro. Binary classification with classical instances and quantum labels. *Quantum Machine Intelligence*, 3(1), may 2021. DOI: [10.1007/s42484-021-00043-z](https://doi.org/10.1007/s42484-021-00043-z). 4
- [19] Matthias C. Caro and Ishaun Datta. Pseudo-dimension of quantum circuits. *Quantum Machine Intelligence*, 2(2), Nov 2020. ISSN 2524-4914. DOI: [10.1007/s42484-020-00027-5](https://doi.org/10.1007/s42484-020-00027-5). 1, 2, 3, 8, 20, 25, 26, 27, 28, 29
- [20] Hao-Chung Cheng, Min-Hsiu Hsieh, and Ping-Cheng Yeh. The learnability of unknown quantum measurements. *Quantum Info. Comput.*, 16(7–8):615–656, may 2016. ISSN 1533-7146. DOI: [10.26421/QIC16.7-8-4](https://doi.org/10.26421/QIC16.7-8-4). 4
- [21] Isaac L. Chuang and M. A. Nielsen. Prescription for experimental determination of the dynamics of a quantum black box. *Journal of Modern Optics*, 44(11-12):2455–2467, 1997. DOI: [10.1080/09500349708231894](https://doi.org/10.1080/09500349708231894). 1
- [22] Kai-Min Chung and Han-Hsuan Lin. Sample Efficient Algorithms for Learning Quantum Channels in PAC Model and the Approximate State Discrimination Problem. In Min-Hsiu Hsieh, editor, *16th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2021)*, volume 197 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:22, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-198-6. DOI: [10.4230/LIPIcs.TQC.2021.3](https://doi.org/10.4230/LIPIcs.TQC.2021.3). 4
- [23] Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning dnf’s. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual*

- Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 815–830, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL <https://proceedings.mlr.press/v49/daniely16.html>. 20
- [24] Steven T Flammia, David Gross, Yi-Kai Liu, and Jens Eisert. Quantum tomography via compressed sensing: error bounds, sample complexity and efficient estimators. *New Journal of Physics*, 14(9):095022, sep 2012. DOI: 10.1088/1367-2630/14/9/095022. 1
- [25] Paul W. Goldberg and Mark R. Jerrum. Bounding the vapnik-chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, 18:131–148, 1995. DOI: 10.1007/BF00993408. 25, 26, 27
- [26] Daniel Gottesman. Stabilizer Codes and Quantum Error Correction, 1997. 2
- [27] Daniel Gottesman. The heisenberg representation of quantum computers, 1998. 2, 12
- [28] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning half-spaces with noise. *SIAM Journal on Computing*, 39(2):742–765, 2009. DOI: 10.1137/070685798. 3, 20
- [29] Jeongwan Haah, Aram W. Harrow, Zhengfeng Ji, Xiaodi Wu, and Nengkun Yu. Sample-optimal tomography of quantum states. *IEEE Transactions on Information Theory*, pages 1–1, 2017. ISSN 1557-9654. DOI: 10.1109/tit.2017.2719044. 1
- [30] Nika Haghtalab. Lecture 9: Hardness of Learning. <https://www.cs.cornell.edu/courses/cs6781/2020sp/lectures/09-hardness1.pdf>, 2020. URL <https://www.cs.cornell.edu/courses/cs6781/2020sp/lectures/09-hardness1.pdf>. Lecture notes for CS6781 - Theoretical Foundations of Machine Learning. 3, 20
- [31] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, oct 2020. DOI: 10.1038/s41567-020-0932-7. 1
- [32] Jonathan Katz. Notes on Complexity Theory Lecture 3. <https://www.cs.umd.edu/~jkatz/complexity/f11/lecture3.pdf>, 2011. URL <https://www.cs.umd.edu/~jkatz/complexity/f11/lecture3.pdf>. Lecture notes for CS 652 — Complexity Theory. 10
- [33] Michael Kharitonov. Cryptographic hardness of distribution-specific learning. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, page 372–381, New York, NY, USA, 1993. Association for Computing Machinery. ISBN 0897915917. DOI: 10.1145/167088.167197. 20
- [34] J. Kleinberg and E. Tardos. *Algorithm Design*. Pearson Education, 2022. ISBN 9780132131087. URL <https://books.google.com/books?id=G0RecgAACAAJ>. 18
- [35] Adam Klivans. The PAC Learning Model. <https://www.cs.utexas.edu/~klivans/f06lec2.pdf>, 2005. URL <https://www.cs.utexas.edu/~klivans/f06lec2.pdf>. Lecture notes for CS 395T Computational Learning Theory. 2
- [36] Robert Koenig and John A. Smolin. How to efficiently select an arbitrary clifford group element. *Journal of Mathematical Physics*, 55(12):122202, Dec 2014. ISSN 1089-7658. DOI: 10.1063/1.4903507. 5
- [37] Richard Kueng and David Gross. Qubit stabilizer states are complex projective 3-designs, 2015. 2
- [38] Ching-Yi Lai and Hao-Chung Cheng. Learning quantum circuits of some t gates. *IEEE Transactions on Information Theory*, pages 1–1, 2022. DOI: 10.1109/TIT.2022.3151760. 2, 11
- [39] Richard A. Low. Learning and testing algorithms for the clifford group. *Phys. Rev. A*, 80:052314, Nov 2009. DOI: 10.1103/PhysRevA.80.052314. 2, 4, 5

- [40] Ashley Montanaro. Learning stabilizer states by Bell sampling, 2017. 2
- [41] Ryan O’Donnell and John Wright. Efficient quantum tomography. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC ’16, page 899–912, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341325. DOI: [10.1145/2897518.2897544](https://doi.org/10.1145/2897518.2897544). 1
- [42] Ryan O’Donnell and John Wright. Efficient quantum tomography ii. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 962–974, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450345286. DOI: [10.1145/3055399.3055454](https://doi.org/10.1145/3055399.3055454). 1
- [43] Yihui Quek, Srinivasan Arunachalam, and John A Smolin. Private learning implies quantum stability. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20503–20515. Curran Associates, Inc., 2021. URL <https://proceedings.neurips.cc/paper/2021/file/abdbeb4d8dbe30df8430a8394b7218ef-Paper.pdf>. 4
- [44] Andrea Rocchetto. Stabiliser states are efficiently PAC-learnable. *Quantum Information & Computation*, 18(7-8):541–552, 2018. DOI: [10.26421/qic18.7-8-1](https://doi.org/10.26421/qic18.7-8-1). 1, 2, 11, 30
- [45] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999. DOI: [10.1137/S0036144598347011](https://doi.org/10.1137/S0036144598347011). 3
- [46] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. DOI: [10.1137/S0097539796298637](https://doi.org/10.1137/S0097539796298637). 2
- [47] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. DOI: [10.1145/1968.1972](https://doi.org/10.1145/1968.1972). 2, 20
- [48] Ewout Van Den Berg. A simple method for sampling random clifford operators. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 54–59, 2021. DOI: [10.1109/QCE52317.2021.00021](https://doi.org/10.1109/QCE52317.2021.00021). 30
- [49] Mithuna Yoganathan. A condition under which classical simulability implies efficient state learnability, 2019. 4
- [50] Huangjun Zhu. Multiqubit clifford groups are unitary 3-designs. *Phys. Rev. A*, 96:062336, Dec 2017. DOI: [10.1103/PhysRevA.96.062336](https://doi.org/10.1103/PhysRevA.96.062336). 2

A Generalization of Caro and Datta [19]

We generalize the main results of Caro and Datta [19], which is itself analogous to Goldberg and Jerrum [25], to allow for projective measurements beyond rank 1, such as in the settings used in this paper. While not necessary for the main result of this paper, we hope to give this proof in a way that is also more black-box accessible for future work.

A.1 Quantum Circuits as Polynomials

The end goal will be to show that the outputs of our concept class can be described as a set of polynomials with bounded degree. Combined with an upper bound on the number of polynomials in that set, we can later arrive at an upper bound on the pseudo-dimension, which itself is an upper-bound on fat-shattering dimension.

We now show a more terse version of Lemma 3 from Caro and Datta [19].

Lemma A.1. *Consider a quantum circuit C with a fixed circuit structure (i.e., the location of the 2-qudit gate are in fixed positions, though the gates themselves can be arbitrary) comprised of at most Γ 2-qudit gates. Such circuit can be described using variables $c_1, c_2, \dots, c_k \in \mathbb{R}$ such that $k = 2\gamma d^4$. Then for every pair of quantum states $|\psi\rangle$ and $|\varphi\rangle$ there exists a polynomial $p_{(|\psi\rangle, |\varphi\rangle)}(c_1, c_2, \dots, c_k) = \text{Tr}[C|\psi\rangle\langle\psi|C^\dagger|\varphi\rangle\langle\varphi|]$ with degree at most 2Γ .*

Proof. Every 2-qudit gate U can be naïvely expressed as the d^4 complex values that make up the $d^2 \times d^2$ unitary. By splitting up the complex values into a real and imaginary part, we get $2d^4$ real values to describe each 2-qudit gate. If $|\psi\rangle = \sum_{s \in \{0,1\}^n} \alpha_s |s\rangle$ then applying a 2-qudit unitary to $|\psi\rangle$ leaves us with the amplitudes of this new state being a polynomial of degree 1 in terms of the entries of U . Note that the α_i , along with the circuit structure, are what determine the coefficients of this polynomial. By applying all γ 2-qudit gates that comprise C , the amplitudes of $C|\psi\rangle$ can be described as a polynomial of degree Γ in $2\gamma d^4$ variables. Finally, since we can write $|\varphi\rangle = \sum_{s \in \{0,1\}^n} \beta_s |s\rangle$, then the inner product $\langle\varphi|C|\psi\rangle$ is some weighted linear combination of the amplitudes of $C|\psi\rangle$, which is a again polynomial with degree at most Γ . To get

$$\text{Tr}[C|\psi\rangle\langle\psi|C^\dagger|\varphi\rangle\langle\varphi|] = |\langle\varphi|C|\psi\rangle|^2$$

we note that the degree at most doubles when we multiply a polynomial by itself. This leaves us with $p_{(|\psi\rangle, |\varphi\rangle)}(c_1, c_2, \dots, c_k)$ as polynomial of degree at most 2Γ and $m = 2\gamma d^4$. \square

Corollary A.2. *Consider a quantum circuit C with a fixed circuit structure (i.e., the location of the 2-qudit gate are in fixed positions, though the gates themselves can be arbitrary) comprised of at most γ 2-qudit gates. Such circuit can be described using variables $c_1, c_2, \dots, c_k \in \mathbb{R}$ such that $k = 2\gamma d^4$. Then for every pair of quantum state $|\psi\rangle$ and projector Π there exists a polynomial $p_{(|\psi\rangle, \Pi)}(c_1, c_2, \dots, c_k) := \text{Tr}[C|\psi\rangle\langle\psi|C^\dagger\Pi]$ with degree at most 2γ .*

Proof. We note that $\Pi = \sum_i |\varphi_i\rangle\langle\varphi_i|$. By linearity, of the trace

$$\begin{aligned} p_{(|\psi\rangle, \Pi)}(c_1, c_2, \dots, c_k) &= \text{Tr} \left[C |\psi\rangle\langle\psi| C^\dagger \Pi \right] \\ &= \text{Tr} \left[C |\psi\rangle\langle\psi| C^\dagger \sum_i |\varphi_i\rangle\langle\varphi_i| \right] \\ &= \sum_i \text{Tr} \left[C |\psi\rangle\langle\psi| C^\dagger |\varphi_i\rangle\langle\varphi_i| \right] \\ &= \sum_i p_{(|\psi\rangle, |\varphi_i\rangle)}(c_1, c_2, \dots, c_k m). \end{aligned}$$

By Lemma A.1, this is the sum of real polynomials in $2\Gamma d^4$ variables with degree at most 2γ . Since the sum does not increase the degree, we are done. \square

Since we fixed the circuit structure, we will want to know how many circuit structures there are, because this directly bounds the number of polynomials we need to consider. The following result was the main ingredient in the proof of Lemma 2 from Caro and Datta [19].

Lemma A.3 (Caro and Datta [19] Lemma 2). *There are at most $\frac{\gamma! \delta^{\gamma-\delta}}{(\gamma-\delta)!} (n!)^\delta$ ways to structure 2-qudit circuits with size γ and depth δ .*

A.2 Pseudo-Dimension of Concept Classes Described via Polynomials

The following is a generalization of Goldberg and Jerrum [25], which used the degree of polynomials to bound the pseudo-dimension of concept classes that could be defined using polynomials in the parameter space of the concepts.

Definition A.4. The *pseudo-dimension* of a concept class \mathcal{C} is the limit of the fat-shattering dimension parameter η as η goes to zero. Formally, the pseudo-dimension is $\lim_{\eta \rightarrow 0^+} \text{fat}_{\mathcal{C}}(\eta)$.

Because fat-shattering dimension increases as η decreases, fat-shattering dimension is always upper-bounded by the pseudo-dimension for all values of $\eta > 0$.

Definition A.5. Let $\{p_1, p_2, \dots, p_m\} \subseteq \mathbb{R}^k \rightarrow \mathbb{R}$ be a set of m polynomials on k variables. For $\eta > 0$, the η -*sign assignment* of $\{p_1, p_2, \dots, p_m\}$ on the input $(x_1, x_2, \dots, x_k) \in \mathbb{R}^k$ is the vector $b \in \{-1, 0, 1\}^m$ such that

$$b_i = \begin{cases} 1 & p_i(x_1, x_2, \dots, x_k) \geq \eta \\ -1 & p_i(x_1, x_2, \dots, x_k) \leq -\eta \\ 0 & \text{otherwise} \end{cases}$$

Lemma A.6 (Goldberg and Jerrum [25] Corollary 2.1). *Let $\{p_1, p_2, \dots, p_m\} \subseteq \mathbb{R}^k \rightarrow \mathbb{R}$ be a set of real polynomials in k variables with $m \geq k$, each of degree at most $d \geq 1$. Then the number of unique η -sign assignments that $\{p_1, p_2, \dots, p_m\}$ can create over all inputs in \mathbb{R}^k is at most $\left(\frac{8edm}{k}\right)^k$ in the limit as $\eta \rightarrow 0^+$.*

Lemma A.6 is a stronger notion than pseudo-dimension, since it upper bounds the number of sign assignments over arbitrarily large sets of inputs. Since pseudo-dimension requires that \mathcal{C} can achieve all (i.e., an exponential number of) sign assignments on some large set of samples, we can show that the pseudo-dimension cannot be too large. We

formalize that notion here. Note that the polynomials in question in the following proof are over the parameters of the concept class, not the inputs. The intuition is that if the output of the concept is some bounded-degree polynomial in the parameter space, there cannot be too many sign assignments.

Theorem A.7 (Generalization of Goldberg and Jerrum [25] Theorem 2.2). *Let $\mathcal{C} \subseteq \Omega \rightarrow [0, 1]$ be a concept class such that every element of \mathcal{C} can be described via k different real variables $c_1, c_2, \dots, c_k \in \mathbb{R}$, as well as an index $l \in [s]$ for $s \geq 0$. Furthermore, for every $f_{c_1, c_2, \dots, c_k, l} \in \mathcal{C}$ and $x \in \Omega$, let $f_{c_1, c_2, \dots, c_k, l}(x) = p_{x, l}(c_1, c_2, \dots, c_k)$ where $p_{x, l}$ is one of s polynomials each with degree at most d for $d \geq 1$. Then the pseudo-dimension of \mathcal{C} is at most $2k \log_2(8eds)$.*

Proof. Let $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m) \subseteq \mathbb{R}$ be the largest set of points pseudo-shattered by \mathcal{C} . If $ms < k$, then there is no issue because the largest shattered set is smaller than k , which is smaller than $2k \log_2(8eds)$. Now assume that $ms \geq k$. By Definition A.4, there must exist some points $y_1, y_2, \dots, y_m \in \mathbb{R}$ and some (potentially arbitrarily small) value $\eta > 0$ such that for all $b_i \in \{\pm 1\}^m$, there is a $f_{c_1, c_2, \dots, c_k} \in \mathcal{C}$ with $b_i \cdot (f_{c_1, c_2, \dots, c_k}(x_i) - y_i) \geq \eta$. Because $f_{c_1, c_2, \dots, c_k}(x_i) = p_{x_i, l}(c_1, c_2, \dots, c_k)$, we can define the new set of polynomials $p'_{i, l} = p_{x_i, l} - y_i$. This means that $\bigcup_{i, l} \{p'_{i, l}\}$ is a set of ms polynomials that must be able to create at all 2^m different sign assignments that define b . However, we know from Lemma A.6 that the number of different sign assignments is at most $\left(\frac{8edms}{k}\right)^k$ as long as $ms \geq k$, which we have assumed to be true. Therefore, $2^m \leq \left(\frac{8edms}{k}\right)^k$. Taking the logarithm of both sides,

$$m \leq k \log_2 \left(\frac{8edms}{k} \right) = k \log_2(8eds) + k \log_2 \frac{m}{k}.$$

We divide the situation into two cases based on which of these two logarithms is bigger: $8eds \geq \frac{m}{k}$ and $8eds < \frac{m}{k}$. The first case is easy to analyze, since if $8eds \geq \frac{m}{k}$, then we directly get $m \leq 2k \log_2(8eds)$ via substitution on the right-hand side. The other case leads to $m < 2k \log_2 \frac{m}{k}$, also via substitution on the right-hand side. Solving this with the Lambert W -function tells us that if $k > 0$ then $m < ke^{-W_{-1}(-\frac{\ln 2}{2})} = 4k$. Because $d \geq 1$ and $s \geq 1$ then $\log_2(8eds) \geq \log_2(4e) > 2$, so $m < 4k < 2k \log_2(8eds)$ for this other case as well. \square

A.3 Pseudo-Dimension for Quantum Circuits

Proposition A.8 (Stirling's approximation).

$$\ln(n!) = n \ln n - n + \mathcal{O}(\ln n) = \mathcal{O}(n \ln n)$$

Theorem A.9 (Generalization of Caro and Datta [19] Theorem 3). *The pseudo-dimension of quantum circuits on n qudits comprised of at most γ 2-qudit gates with depth δ is upper bounded by $\mathcal{O}(d^4 \delta \gamma^2 \log \gamma)$.*

Proof. We want to apply Theorem A.7 to the concept class of quantum circuits. We know from Corollary A.2 that for fixed circuit structure with γ gates and depth δ that it can be described as a polynomial with degree at most 2γ in the $2\gamma d^4$ real variables that describe the entries of the circuit. Furthermore, Lemma A.3 tells us that there is at most $\frac{\gamma \delta^{\gamma-\delta}}{(\gamma-\delta)!} (n!)^\delta$ different circuit structures. We then apply Theorem A.7 with $k = 2\gamma d^4$, $d = 2\gamma$

and $s = \frac{\gamma! \delta^{\gamma-\delta}}{(\gamma-\delta)!} (n!)^\delta$. As a result we get that the pseudo-dimension is at most

$$2k \log_2(8eds) = 4\gamma d^4 \log_2 \left(8e(2\gamma) \frac{\gamma! \delta^{\gamma-\delta}}{(\gamma-\delta)!} (n!)^\delta \right) \quad (3)$$

We will now focus on giving an upper bound for the logarithmic term by showing that

$$\log_2 \left(8e(2\gamma) \frac{\gamma! \delta^{\gamma-\delta}}{(\gamma-\delta)!} (n!)^\delta \right) = \mathcal{O}(\delta\gamma \log \gamma).$$

Splitting up the logarithm into sums and applying Stirling's approximation to each factorial, we arrive at

$$\begin{aligned} \log_2 \left(8e(2\gamma) \frac{\gamma! \delta^{\gamma-\delta}}{(\gamma-\delta)!} (n!)^\delta \right) &= 4 + \log_2 e + \log_2 \gamma + (\gamma - \delta) \log_2 \gamma \\ &\quad + \mathcal{O}(\gamma \ln \gamma) + \delta \cdot \mathcal{O}(n \ln n) + \mathcal{O}((\delta - \gamma) \ln(\gamma - \delta)) \\ &= \mathcal{O}(\gamma \log \gamma + \delta n \log n + \delta \log \gamma) \end{aligned}$$

Due to the definition of circuit structure, we know that $\delta \leq \gamma$. WLOG, we can also assume that every qubit has been acted upon by at least one gate (even if it's just the identity gate) such that $n \leq \gamma$. Together, we arrive that the logarithmic term is at most $\mathcal{O}(\delta\gamma \log \gamma)$.

Since we have achieved our goal of upper-bounding the logarithmic term, from Eq. (3) we immediately get that the pseudo-dimension is at most $\mathcal{O}(d^4 \delta \gamma^2 \log \gamma)$. \square

We now state the generalization of the main result Caro and Datta [19] to projective measurements of arbitrary rank.

Corollary A.10 (Generalization of Caro and Datta [19] Corollary 3). *Let X be the set of quantum states on n qudits, and let Y be the set of all projectors on n qudits. Let U_* be a quantum circuit of 2-qudit quantum gates with size Γ and depth Δ . Let \mathcal{D} be a probability distribution on $X \times Y$ unknown to the learner. Let*

$$S = \left\{ \left((x^{(i)}, y^{(i)}), \text{Tr} \left[y^{(i)} U_* x^{(i)} U_*^\dagger \right] \right) \right\}_{i=1}^m$$

be corresponding training data where each $(x^{(i)}, y^{(i)})$ is drawn i.i.d according to \mathcal{D} . Let $\delta, \epsilon, \alpha, \beta \in (0, 1)$ where $\beta > \alpha$. Then, training data of size

$$m = \mathcal{O} \left(\frac{1}{\epsilon} \left(\Delta d^4 \Gamma^2 \log \Delta \log^2 \left(\frac{\Delta d^4 \Gamma^2 \log(\Gamma)}{(\beta - \alpha)\epsilon} \right) + \log \frac{1}{\delta} \right) \right)$$

suffice to guarantee that, with probability $\geq 1 - \delta$ with regard to choice of the training data, any quantum circuit U of size Γ and depth Δ that satisfies

$$\left| \text{Tr} \left[y^{(i)} U_* x^{(i)} U_*^\dagger \right] - \text{Tr} \left[y^{(i)} U x^{(i)} U^\dagger \right] \right| \leq \alpha \quad \forall 1 \leq i \leq m$$

also satisfies

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\left(\text{Tr} \left[y^{(i)} U_* x^{(i)} U_*^\dagger \right] - \text{Tr} \left[y^{(i)} U x^{(i)} U^\dagger \right] \right)^2 \right] \leq (1 - \epsilon) \beta^2 + \epsilon$$

Proof. We combine Theorem A.9 with Theorem 2.16, along with the fact that for all $\eta > 0$ the η -fat-shattering dimension is upper-bounded by pseudo-dimension. \square

As shown by Theorem 4 of Caro and Datta [19], a similar thing can be done with n -qudit quantum processes by simply changing the d^4 to d^8 in Lemma A.1 and Corollary A.2. This is because a quantum process is still a linear operation, but contains d^8 many entries now in parameter space. This propagates to Theorem A.9 and Corollary A.10 by again replacing every appearance of d^4 with d^8 .

B Special Cases with Efficient Proper Learners

Despite the results given, there still exist situations where it is possible to efficiently properly learn Clifford circuits and CNOT circuits. We give brief proof sketches of some of them here.

B.1 CNOT Circuits for a Distribution with Support over a Single Measurement

Let us try to learn CNOT circuits with regard to a distribution \mathcal{D} such that there exists some pauli $P \in \{I, Z\}^{\otimes n}$ with $\mathbb{P}_{(\rho, E) \sim \mathcal{D}}[E = \frac{I^{\otimes n} + P}] = 1$. Because we are dealing with CNOT circuit the labels will always be 0 and 1 so by Lemma 3.4 each label will tell us an affine subspace that $C^\dagger P C$ lies in. We can efficiently compute the intersection of this using Gaussian elimination with the generators to find a P' that is consistent with all of the labels. From there, let P, Q_2, Q_3, \dots, Q_n be a set of Paulis whose span is $\{I, Z\}^{\otimes n}$. Let $P', Q'_2, Q'_3, \dots, Q'_n$ also be a set of Paulis whose span is $\{I, Z\}^{\otimes n}$. It is clear that if we define our CNOT circuit such that $C^\dagger P C = P'$ and $C^\dagger Q_i C = Q'_i$ then we have a valid CNOT circuit. Efficiently finding such $\{Q_i\}$ and $\{Q'_i\}$ only takes $\mathcal{O}(n)$ expected samples of random Paulis in $\{I, Z\}^{\otimes n}$ and so can be done efficiently. Appealing to both Proposition 2.6 and Theorem 2.16 completes the proof.

B.2 Clifford circuits with the Uniform Distribution over Pauli Measurements

We note that if the distribution \mathcal{D} entails the measurements being uniform over the Paulis then the problem is trivially easy to properly learn with $\epsilon < \frac{1}{\exp(n)}$ and $\delta = 0$ by just outputting a random Clifford circuit. This is because the probability that a random Pauli is in a given state's stabilizer group is $\frac{2^n}{4^n}$ so we will almost always see the label $\frac{1}{2}$ regardless of the hypothesis circuit we choose.

B.3 CNOT Circuits with the Uniform Distribution over $\{I, Z\}^{\otimes n}$

Let Θ and Q be the matrix/vector forms of θ_{ij} and q_j values from Eq. (1). We note that if we have enough independent samples that $\Theta \oplus Q$ is confined to a $\mathcal{O}(\log n)$ dimensional affine subspace then we can simply iterate through all possible Θ and Q combinations to find one with a non-singular Θ in $\text{poly}(n)$ time. Let's say that we've restricted $\Theta \oplus Q$ to lie in a d dimensional affine subspace. Let M' be the true value of Θ . and let $M \neq M'$ be another arbitrary matrix. Likewise let x' be the true value of Q and x some arbitrary vector. Now let $Z^w \in \{I, Z\}^{\otimes n}$ be a Pauli selected uniformly at random, and $\rho = |v\rangle\langle v|$ a uniformly random computational basis state. The pair M and x will give the same label as the true label on the input $(\rho, \frac{I^{\otimes n} + Z^w}{2})$ if and only if

$$\left(v^T (M + M') + x^T + (x')^T \right) w = 0 \pmod{2}.$$

Because w is uniform random, as long as $v^T (M + M') + x^T + (x')^T$ is not the zero vector over \mathbb{F}_2 then this will only be 0 at most half the time. Since at least one of $M \neq M'$ or

$x \neq x'$ is true, then $(M + M')^T v = x + x'$ will only be true with probability at most $\frac{1}{2}$ as well. So the probability that any arbitrary M and x have different labels is at least $\frac{1}{4}$. Thus with $\mathcal{O}(n)$ expected samples uniformly drawn from arbitrary basis states and Z^w we will have constrained our system to something we can bruteforce to find a full rank Θ and corresponding q_j values that is consistent with all samples. From there we again apply both Proposition 2.6 and Theorem 2.16 to generalize with zero training error as long as the number of samples is also at least the parameter m from the theorem statement.

B.4 Clifford Circuits for a Distribution with Support over a Single State

In the converse of an earlier situation, let us try to learn Clifford circuits with regard to a distribution \mathcal{D} such that there exists some stabilizer state σ with $\mathbb{P}_{(\rho, E) \sim \mathcal{D}}[\rho = \sigma] = 1$. This situation effectively reduces to that of Rocchetto [44]. If we run that algorithm we will find a state σ' that is consistent with all of the labels. Let $\{g_i\}$ be the generators of σ and $\{g'_i\}$ the generators of σ' . If we let $Cg_iC^\dagger = g'_i$ we define the first part of a Clifford circuit that maps σ to σ' as desired. We can then run the algorithm from Van Den Berg [48] to fill in the remainder of the Clifford circuit. Appealing to both Proposition 2.6 and Theorem 2.16 once again completes the proof.