



This is a repository copy of *Learning quantum graph states with product measurements*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/198734/>

Version: Accepted Version

---

**Proceedings Paper:**

Ouyang, Y. [orcid.org/0000-0003-1115-0074](https://orcid.org/0000-0003-1115-0074) and Tomamichel, M. (2022) Learning quantum graph states with product measurements. In: 2022 IEEE International Symposium on Information Theory (ISIT) Proceedings. 2022 IEEE International Symposium on Information Theory (ISIT), 26 Jun - 01 Jul 2022, Espoo, Finland. Institute of Electrical and Electronics Engineers (IEEE) , pp. 2963-2968. ISBN 9781665421607

<https://doi.org/10.1109/isit50566.2022.9834440>

---

© 2022, IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works. Reproduced in accordance with the publisher's self-archiving policy.

**Reuse**

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.



[eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk)  
<https://eprints.whiterose.ac.uk/>

# Learning quantum graph states with product measurements

Yingkai Ouyang

Department of Electrical and Computer Engineering,  
Centre of Quantum Technologies,  
National University of Singapore  
Email: oyingkai@gmail.com

Marco Tomamichel

Department of Electrical and Computer Engineering,  
Centre of Quantum Technologies,  
National University of Singapore

**Abstract**—We consider the problem of learning  $N$  identical copies of an unknown  $n$ -qubit quantum graph state with product measurements. These graph states have corresponding graphs where every vertex has exactly  $d$  neighboring vertices. Here, we detail an explicit algorithm that uses product measurements on multiple identical copies of such graph states to learn them. When  $n \gg d$  and  $N = O(d \log(1/\epsilon) + d^2 \log n)$ , this algorithm correctly learns the graph state with probability at least  $1 - \epsilon$ . From channel coding theory, we find that for arbitrary joint measurements on graph states, any learning algorithm achieving this accuracy requires at least  $\Omega(\log(1/\epsilon) + d \log n)$  copies when  $d = o(\sqrt{n})$ . We also supply bounds on  $N$  when every graph state encounters identical and independent depolarizing errors on each qubit.

## I. INTRODUCTION

Learning of quantum states has been investigated in a multitude of settings. In the most traditional setting, quantum tomography [1] studies this learning problem, and this topic still attracts plenty of attention [2], [3], [4], [5]. In quantum tomography, we learn a description of the quantum state to a prescribed degree of accuracy given multiple identical copies. Studies in quantum tomography are concerned with obtaining bounds on the minimum number  $N$  of such copies for different families of quantum states. Suppose that  $\rho = |\psi\rangle\langle\psi|$  is an  $n$ -qubit pure state, and that the estimate of  $\rho$  given by  $\hat{\rho}$  is close to  $\rho$  (for some constant precision, for example in trace distance) with probability at least  $1 - \epsilon$ . For pure states, [6, Sec. IIA] and [7] showed that for any measurement strategy, even when entangling operations are applied across multiple copies of  $\rho$ , we have  $N = \tilde{\Theta}(2^n + \log \frac{1}{\epsilon})$ , omitting terms linear in  $n$ . This implies that the optimal learning strategy for determining an arbitrary  $n$ -qubit pure state requires  $N$  to be exponential in  $n$ .

Imposing additional structure (apart from purity) on quantum states allows substantial reduction of the number of copies  $N$  required to learn  $\rho$ . For instance, learning an unknown stabilizer state from the set of all stabilizer states using collective measurements can be achieved with  $N$  linear in  $n$  [8], [9], [10]. Bounds on  $N$  for the learning of subsets of stabilizer states [11] or quantum states that are stabilizer pseudomixtures [12] have also recently been obtained.

Learning a quantum state using measurements that act on a single or multiple copies of an  $n$ -qubit state  $|\psi\rangle$  can be challenging to implement. This is because the entangling operations across multiple qubits that these measurements require can be difficult to implement in an error-free way in practice. It would be highly beneficial if we could learn quantum states by simply performing product measurements. A simple way

to perform product measurements on  $N$  copies of  $|\psi\rangle$  is to measure each qubit of  $|\psi\rangle$  either in the computation basis  $B_0 = \{|0\rangle, |1\rangle\}$  or in the Hadamard basis  $B_1 = \{|+\rangle, |-\rangle\}$ , where  $|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}$  and  $|0\rangle$  and  $|1\rangle$  form an orthonormal basis of a qubit. If we measure a qubit in the basis  $B_0$ , we denote the measurement outcomes corresponding to  $|0\rangle$  and  $|1\rangle$  to be 0 and 1 respectively. If we measure a qubit in the basis  $B_1$ , we denote the measurement outcomes corresponding to  $|+\rangle$  and  $|-\rangle$  to be 0 and 1 respectively. While product measurements are easy to describe, it is unclear how many copies  $N$  of  $|\psi\rangle$  a learning algorithm that uses product measurements would need.

In this paper, we consider learning a family of graph states using product measurements. Graph states [13], [14], [15], [16] are quantum states that correspond directly to undirected simple graphs, and the set of all graph states is equivalent under local Clifford operations to stabilizer states [17], [18], [19]. Recently Montanaro and Shao evaluated an upper bound on  $N$  the number of copies of  $|\psi\rangle$  required to learn the underlying graph of degree  $d$ , and found that  $N = O(d \log n)$  when collective measurements are performed on pairs of graph states [11].

In this paper, we detail an explicit algorithm that uses product measurements to learn  $n$ -qubit graph states of degree  $d$  when  $n \gg d$ . Product measurements are much simpler to perform than independent measurements; while independent measurements act on a single copy of the  $n$ -qubit state  $|\psi\rangle$ , product measurements measure every qubit in every  $|\psi\rangle$  individually. For this algorithm to correctly learn what the graph state is with probability at least  $1 - \epsilon$ , it suffices to require that the number of copies  $N$  satisfies

$$N \geq 4ed \log(n/\epsilon) + 4ed^2 \log(ne), \quad (1)$$

where  $\log$  denotes the natural logarithm and  $e$  is Euler's constant. From channel coding theory, we find that for arbitrary measurements on the graph states, we must have

$$N \geq d \log_4(nd), \quad (2)$$

thereby also establishing the asymptotic optimality of Montanaro and Shao's scheme. We also supply bounds on  $N$  when using our algorithm and also for any algorithm when every graph state  $|\psi\rangle$  encounters a qubit depolarizing channel that acts independently and identically on each of its qubits.

## II. PRELIMINARIES

Let  $G_{n,d}$  denote the set of graphs with  $n$  vertices, and with vertex degree equal to  $d$  [20]. These graphs are called  $d$ -regular

graphs on  $n$  vertices. Here, we require that  $d = o(\sqrt{n})$ . For any graph  $g \in G_{n,d}$ , we use  $V = \{1, \dots, n\}$  denote the set of vertices, and  $E(g)$  to denote the edge set of  $g$ . Given any vertex  $v$ , let  $N_g(v) = \{i \in V : \{v, i\} \in E(g)\}$  denote the neighbor set of  $v$ . Note that  $N_g(v) \subset V \setminus \{v\}$ . The graph state  $|g\rangle$  is defined to be the unique stabilizer state stabilized by the stabilizer generated by the generators

$$W_v = X_v \prod_{j \in N_g(v)} Z_j, \quad (3)$$

for  $v \in V$ , where  $X_k$  and  $Z_j$  denote the Pauli  $X$  operator (a bit-flip) acting on the  $v$ th qubit and identity everywhere else, and the Pauli operator  $Z$  (a phase-flip) acting on the  $j$ th qubit respectively. This means that  $|g\rangle$  is the unique state satisfying the equations

$$W_v |g\rangle = |g\rangle, \quad \forall v \in V. \quad (4)$$

We define a function Oracle() which is function that always returns as its output the graph state  $|g\rangle$ .

In a graph state, each qubit corresponds to a vertex in a graph. In the preparation of a graph state, each qubit is first initialized as a  $|+\rangle$  state. Second, for every edge in the graph, a controlled phase gate is applied between the corresponding qubits. Since the controlled phase gate is invariant under swapping of the pair of qubits it acts on, the distinction between the control and target qubit is not important. Moreover, since all controlled phase gates are diagonal in the computation basis, they must commute. This implies that the order in which controlled phase gates are applied is not important.

Given any binary vector  $\mathbf{v}$ , we let  $\text{wt}(\mathbf{v})$  denote its Hamming weight. Given that an  $n$ -bit binary vector  $\mathbf{v}$  we define

$$\text{Maj}(\mathbf{v}) = \begin{cases} 1 & \text{if } \text{wt}(\mathbf{v}) > n/2 \\ \hat{Q} & \text{if } \text{wt}(\mathbf{v}) = n/2 \\ 0 & \text{if } \text{wt}(\mathbf{v}) < n/2 \end{cases}, \quad (5)$$

where  $\hat{Q}$  is a random variable such that  $\Pr[\hat{Q} = j] = 1/2$  for  $j = 0, 1$ .

### III. LEARNING WITHOUT NOISE

#### A. Algorithm

We could learn what  $g$  is, by applying Algorithm MeasQbits which performs product measurements on  $N$  copies of an unknown graph state  $|g\rangle$ . This graph state is obtained by querying Oracle() a total of  $N = mr$  times where  $m$  and  $r$  are positive integers.

**Function** MeasQbits( $m, r, w$ )

**Input:** Positive integers  $m$  and  $r$  and integer  $w \in \{1, \dots, n\}$ .

**Output:** Bits  $x_{k,j}, m_{t,k,j}$  for  $j \in V, k = 1, \dots, m$ , and  $t = 1, \dots, r$ .

1. **For**  $k = 1:m$
2. Pick  $\mathbf{x} = (x_{k,1}, \dots, x_{k,n})$  uniformly at random from all  $n$ -bit strings where  $\text{wt}(\mathbf{x}) = w$
3. **For**  $t = 1:r$
4. Set  $|g\rangle \leftarrow \text{Oracle}()$
5. **For**  $j = 1:n$
6. Measure qubit  $j$  of  $|g\rangle$  in basis  $B_{x_{k,j}}$ .
7. Set  $m_{t,k,j}$  to be the measurement outcome.

8. **EndFor**
9. **EndFor**
10. **EndFor**

The output of MeasQbits is encoded in the binary matrices  $(X)_{k,j}$  and  $(M_t)_{k,j}$ , where  $x_{k,j}$  and  $m_{t,k,j}$  denote the matrix elements in the  $k$ th row and  $j$ th columns of  $(X)_{k,j}$  and  $(M_t)_{k,j}$  respectively. Based on this information, we estimate the most likely neighbor set of each vertex  $v$  using the following algorithm.

**Function** FindNbs( $(X)_{k,j}, (M_1)_{k,j}, \dots, (M_r)_{k,j}$ )

**Input:** Binary matrices  $(X)_{k,j}, (M_1)_{k,j}, \dots, (M_r)_{k,j}$ .

**Output:**  $\mathcal{S}_1, \dots, \mathcal{S}_n$ , where each  $\mathcal{S}_v$  is a tuple with each component that are subsets of  $V \setminus \{v\}$  and with  $|\alpha_i| = d$ .

1. Set  $\mathcal{S}_v = \{\alpha \subset V \setminus \{v\} : |\alpha| = d\}$  for  $v \in V$
2. **For**  $k = 1 : m$
3. Define the index set  $W = \{j : x_{k,j} = 1\}$
4. **For**  $v \in W$
5. **For**  $\alpha = \{a_1, \dots, a_d\} \subset V \setminus W$
6. **For**  $t = 1 : r$
7. Set  $s_{t,k,v,\alpha} = \text{mod}(m_{t,k,v} + \sum_{j=1}^d m_{t,k,a_j}, 2)$ .
9. **EndFor**
8. Set  $\mathbf{s}_{k,v,\alpha} = (s_{1,k,v,\alpha}, \dots, s_{r,k,v,\alpha})$ .
10. **If**  $\text{Maj}(\mathbf{s}_{k,v,\alpha}) = 1$ , delete  $\alpha$  from  $\mathcal{S}_v$ .
11. **EndFor**
12. **EndFor**
13. **EndFor**

**Function** LearnGraphState( $m, r, w$ )

**Input:** Positive integers  $m$  and  $r$  and integer  $w \in \{1, \dots, n\}$ .

**Output:**  $(\eta_1, \dots, \eta_m)$ .

1. Set  $((X)_{k,j}, (M_1)_{k,j}, \dots, (M_r)_{k,j}) = \text{MeasQbits}(m, r, w)$
2. Set  $(\mathcal{S}_1, \dots, \mathcal{S}_n) = \text{FindNbs}((X)_{k,j}, (M_1)_{k,j}, \dots, (M_r)_{k,j})$

The algorithm LearnGraphState succeeds if for all  $v = 1, \dots, n$ , we have  $\mathcal{S}_v = \{N_g(v)\}$ . This algorithm also uses  $N = mr$  copies of  $|g\rangle$  to learn what  $g$  is.

#### B. Analysis

Here, we analyze the algorithms introduced in Section III-A assuming that we obtain noiseless copies of  $|g\rangle$  from querying Oracle().

Lemma 1 gives the probability that for a random  $v \in V$  belongs to a fixed  $d$ -set  $\alpha$  does not contain  $v$ , and also is a subset of the complement of a random  $w$ -set  $W$ . This corresponds to the probability that a given  $\alpha$  is sampled at the  $k$ th iteration of Algorithm FindNbs.

**Lemma 1.** Now let  $\alpha$  be any fixed  $d$ -set  $\alpha$  where  $\alpha \subset V$ . Let  $v \in V$  be random and let  $W$  be a random subset of  $V$  with cardinality  $w$ . Then  $\Pr[\alpha \cap W = \emptyset \wedge v \in W] = p_{\text{samp}}$  where

$$p_{\text{samp}} = \frac{w \binom{n-d}{w}}{n \binom{n}{w}}. \quad (6)$$

*Proof.* Now  $\Pr[v \in W] = w/n$  and  $\Pr[\alpha \cap W = \emptyset] = \binom{n-d}{w} / \binom{n}{w}$ . The events  $v \in W$  and  $\alpha \cap W = \emptyset$  are independent

because  $v$  is a random variable that is independent of the non-random  $\alpha$ . Hence the joint probability is the product of their individual probabilities.  $\square$

The sampling probability  $p_{\text{samp}}$  is used later in our analysis, and we will need bounds on it.

**Lemma 2.** Suppose that  $d \geq 2$  and  $n \geq 2d^2$ . Then

$$\frac{1}{2ed} \leq \left( \frac{1}{ed} - \frac{1}{4ed(1-d/n)} \right) (1-d/n) \leq p_{\text{samp}} \leq \frac{1}{ed}. \quad (7)$$

*Proof.* Let  $w = \lceil \frac{n-d}{d} \rceil$ . We can see that  $\frac{n}{d} - 1 \leq w \leq \frac{n}{d}$ . Note that

$$\begin{aligned} p_{\text{samp}} &= \frac{w}{n} \prod_{j=0}^{w-1} \left( 1 - \frac{d}{n-j} \right) \geq \frac{w}{n} \left( 1 - \frac{d}{n-w+1} \right)^w \\ &\geq \frac{n-d}{dn} \left( 1 - \frac{d}{n(1-1/d)+1} \right)^{n/d}. \end{aligned} \quad (8)$$

Since  $n/d \geq 2$  we have

$$p_{\text{samp}} \geq \frac{n-d}{dn} \left( 1 - \frac{d}{n} \right)^{n/d} = \frac{1}{d} \left( 1 - \frac{d}{n} \right)^{n/d+1}. \quad (9)$$

Now, note that

$$(1-x)^{1/x} = 1/e - x/(2e) + O(x^2). \quad (10)$$

Using Taylor's theorem, for  $0 < x < 1$ , we can show that

$$(1-x)^{1/x} \geq \frac{1}{e} - \frac{x}{2e(1-x)}. \quad (11)$$

Therefore

$$\left( 1 - \frac{d}{n} \right)^{n/d} \geq \frac{1}{e} - \frac{d/n}{2e(1-d/n)}. \quad (12)$$

Substituting (12) into (9) and using  $d/n \geq 1/(2d)$  gives the first lower bound for the lemma.

For the second lower bound, note that  $1-d/n \geq 1-1/(2d) \geq 3/4$ , and from the first lower bound we get  $p_{\text{samp}} \geq \frac{3}{4ed} \left( 1 - \frac{1}{2(3/4)} \right) = \frac{3}{4ed} \left( 1 - \frac{1}{4(3/4)} \right) = \frac{1}{2ed}$ .

For the upper bound, note that

$$p_{\text{samp}} \leq \frac{w}{n} \left( 1 - \frac{d}{n} \right)^w. \quad (13)$$

The upper bound is a continuous function of  $w$ , and its derivative is monotone decreasing on  $1 \leq w \leq n$ . The derivative is positive when  $w = 1$  and negative when  $w = n$ . Hence this upper bound is maximized in the interval  $[1, n]$ , and is attained when  $w = -1/\log(1-d/n)$  with optimal value

$$\frac{-1}{en \log(1-d/n)} \leq \frac{1}{ed}. \quad (14)$$

$\square$

Note that when  $d$  grows and  $d = o(n)$ , Lemma 2 implies that

$$p_{\text{samp}} \geq \frac{1}{ed} + O(1/n). \quad (15)$$

Note that for a fixed value of  $t, k$  and  $v$ , we have that

$$\Pr[s_{t,k,v,\alpha} = 1] = \begin{cases} 0 & \alpha = N_g(v) \\ 1/2 & \alpha \neq N_g(v) \end{cases}. \quad (16)$$

This is because of two reasons. First, if  $\alpha = \{a_1, \dots, a_d\} = N_g(v)$ , the parity of the measured bits  $m_{t,k,v}, m_{t,k,a_1}, \dots, m_t$  must be even, which means that  $s_{t,k,v,\alpha}$  is always equal to zero. Second, if  $\alpha = \{a_1, \dots, a_d\} \neq N_g(v)$ , the parity of the measured bits  $m_{t,k,v}, m_{t,k,a_1}, \dots, m_t$  is even and odd with equal probability, which means that  $s_{t,k,v,\alpha}$  is always equal to 1 with probability  $1/2$ . Next, it is easy to see that

$$\Pr[\text{Maj}(s_{k,v,\alpha}) = 1] = \begin{cases} 0 & \alpha = N_g(v) \\ 1/2 & \alpha \neq N_g(v) \end{cases}. \quad (17)$$

We now specify conditions under which LearnGraphStates fails with probability at most  $\epsilon$ .

**Theorem 3.** Let the conditions of Lemma 2 hold, and that  $w = \lceil \frac{n-d}{d} \rceil$ . Suppose that

$$m \geq 4ed \log(n/\epsilon) + 4ed^2 \log(ne/d). \quad (18)$$

Then, using  $N = mr$  copies of  $|g\rangle$ , the probability that Algorithm LearnGraphState gives the correct output is at least  $1 - \epsilon$ .

**Remark 4.** To use LearnGraphState in the noiseless setting, we can set  $r = 1$ , so that  $N \geq 4ed \log(n/\epsilon) + 4ed^2 \log(ne/d)$  copies of  $|g\rangle$  suffices to learn  $|g\rangle$  with probability at least  $1 - \epsilon$ .

*Proof of Theorem 3.* It suffices to show that the probability that LearnGraphState finds some  $v \in V$  for which  $\mathcal{S}_v \neq \{N_g(v)\}$  is at most  $\epsilon$ .

For any  $v \in V$ , suppose that an  $\alpha \neq N_g(v)$  has been sampled  $s$  times by LearnGraphState. Using (17), the probability that a random  $\mathcal{S}_v$  contains  $\alpha$  is  $2^{-s}$ . The number  $s$  ranges from 0 to  $m$ . Hence, at the conclusion of LearnGraphState,

$$\begin{aligned} &\Pr[\alpha \in \mathcal{S}_v] \\ &= \sum_{s=0}^m \binom{m}{s} p_{\text{samp}}^s (1-p_{\text{samp}})^{m-s} 2^{-s} \\ &= \sum_{s=0}^m \binom{m}{s} (p_{\text{samp}}/2)^s (1-p_{\text{samp}})^{m-s} \\ &= (p_{\text{samp}}/2 + 1 - p_{\text{samp}})^m = (1 - p_{\text{samp}}/2)^m. \end{aligned} \quad (19)$$

Applying the union bound on all of the vertices  $v$ , and on all  $d$ -sets that are subsets of  $V \setminus \{v\}$ , we must have

$$n \binom{n-1}{d} (1 - p_{\text{samp}}/2)^m \leq \epsilon. \quad (20)$$

The above inequality is equivalent to

$$m \log \left( \frac{1}{1 - p_{\text{samp}}/2} \right) \geq \log \left( \frac{n \binom{n-1}{d}}{\epsilon} \right). \quad (21)$$

Now  $\log \left( \frac{1}{1 - p_{\text{samp}}/2} \right) \geq p_{\text{samp}}/2$ . Choosing  $w = \frac{n-d}{d}$ , from Lemma 2, we get  $p_{\text{samp}} \geq 1/(2ed)$ . Since we also have  $\binom{n-1}{d} \leq \binom{n}{d} \leq (ne/d)^d$ , for (21) to hold, it suffices to require the following inequality to hold

$$m \geq (p_{\text{samp}}/2)^{-1} (\log(n/\epsilon) + d \log(ne/d)). \quad (22)$$

$\square$

#### IV. LEARNING WITH NOISE

Now consider errors that afflict our quantum graph state. We model noise using the the qubit depolarizing channel  $\mathcal{D}_p$ , which applies the identity operator on a qubit with probability  $1 - p$ , and with probability  $p/3$ , applies a bit-flip  $X$ , a phase flip  $Z$ , or both a bit-flip and a phase-flip  $Y = iXZ$ . On every  $|g\rangle$  obtained from each query of Oracle(), the  $n$ -qubit depolarizing channel  $\mathcal{D}_p^{\otimes n}$  acts on  $|g\rangle$  before the product measurements are performed.

Depolarizing noise affects measurement outcomes in the bases  $B_0$  and  $B_1$  in a simple way. If we measure a qubit with density matrix  $\tau$  in the basis  $B_0$ , the probability of obtaining the outcome  $|0\rangle$  and  $|1\rangle$  is  $\text{Tr}(\frac{I+Z}{2}\tau)$  and  $\text{Tr}(\frac{I-Z}{2}\tau)$  respectively. Since  $\frac{I+Z}{2}Z = \frac{I+Z}{2}$ , a  $Z$  error does not change measurement outcomes in the basis  $B_0$ . Similarly an  $X$  does not change the measurement outcome in the basis  $B_1$ . On the other hand, an  $X$  error flips the measurement outcome in the  $B_0$  basis, a  $Z$  error flips the measurement outcome in the  $B_1$  basis, and a  $Y$  error flips the measurement outcome in both bases. Hence, for both bases  $B_0$  and  $B_1$ , the probability that a qubit measurement outcome is flipped is  $2p/3$ .

Here, Lemma 5 gives the probabilities that the majority function in FindNBs evaluates to 1.

**Lemma 5.** Suppose that for some real  $p$  where  $0 \leq p < 3/4$ , errors modeled by  $\mathcal{D}_p^{\otimes n}$  occur on every oracle output  $|g\rangle$ . Then for a fixed  $v$  and  $k$ ,

$$\Pr[\text{Maj}(\mathbf{s}_{k,v,\alpha}) = 1] = \begin{cases} \eta & , \alpha = N_g(v) \\ 1/2 & , \alpha \neq N_g(v) \end{cases}, \quad (23)$$

where  $\eta = \sum_{t > n/2} \binom{r}{t} (\frac{1}{2} - \gamma)^t (\frac{1}{2} + \gamma)^{r-t}$  and

$$\gamma = \frac{(1 - 4p/3)^{d+1}}{2}. \quad (24)$$

*Proof.* The second result of Lemma 5 follows directly from (17), because the presence of depolarizing errors does not affect the probability of the measurement outcomes in both bases  $B_0$  and  $B_1$  when  $\alpha \neq N_g(v)$ . Hence only the first result of the lemma is non-trivial.

Let  $q = 2p/3$ . We now prove (23). Using the theory of generating functions, for any  $t = 1, \dots, r$ , we have that

$$\begin{aligned} & \Pr[s_{t,k,v,\alpha} = 1 | \alpha = N_g(v)] \\ &= \frac{(1 - q + q)^{d+1} - (1 - q - q)^{d+1}}{2} \\ &= \frac{1}{2} - \frac{(1 - 2q)^{d+1}}{2} = \frac{1}{2} - \gamma. \end{aligned} \quad (25)$$

The result follows from the independence of the random variables  $s_{1,k,v,\alpha}, \dots, s_{t,k,v,\alpha}$  and definition of the majority function in (5).  $\square$

**Theorem 6.** Let  $p, \epsilon \in \mathbb{R}$  such that  $0 \leq p \leq 3/4$  and  $\epsilon > 0$ . Suppose that every oracle evaluation in LearnGraphState returns a noisy graph state  $\mathcal{D}_p^{\otimes n}(|g\rangle\langle g|)$ . Then there exists some value of  $N$  for which LearnGraphState learns  $|g\rangle$  correctly with probability at least  $1 - \epsilon$  using  $N$  copies of  $|g\rangle$  where

$$N = O\left(\frac{1 - 4\gamma^2}{\gamma^2} f(\epsilon, n, d) g(\epsilon, n, d)\right), \quad (26)$$

where  $f(\epsilon, n, d) = \log(\epsilon^{-1}) + \log(d \log n)$ ,  $g(\epsilon, n, d) = d \log(\epsilon^{-1}) + d^2 \log n$ , and  $\gamma$  is as given in (24).

*Proof of Theorem 6.* The first part of the proof is to find a lower bound on  $m$  for which the probability that some  $\alpha \neq N_g(v)$  belongs to some  $\mathcal{S}_v$  is at most  $\epsilon/2$ . This lower bound

$$m \geq 4ed \log(n/(2\epsilon)) + 4ed^2 \log(ne/d) \quad (27)$$

can be obtained directly from Theorem 3.

The second part of the proof is to find an upper bound on  $m$  for which the probability that  $\alpha \notin \mathcal{S}_v$  is at most  $\epsilon/2$  if  $\alpha = N_g(v)$ . Now, if  $\alpha = N_g(v)$  is sampled  $s$  times, the probability that  $\alpha$  is eliminated from  $\mathcal{S}_v$  is  $1 - (1 - \eta)^s$ .

The overall probability that  $\alpha$  is eliminated from  $\mathcal{S}_v$  is

$$\begin{aligned} & \sum_{s=0}^m \binom{m}{s} p_{\text{samp}}^s (1 - p_{\text{samp}})^{m-s} (1 - (1 - \eta)^s) \\ &= 1 - (p_{\text{samp}}(1 - \eta) + 1 - p_{\text{samp}})^m \\ &= 1 - (1 - \eta p_{\text{samp}})^m. \end{aligned} \quad (28)$$

We require that  $1 - (1 - \eta p_{\text{samp}})^m \leq \frac{\epsilon}{2}$ . This is equivalent to requiring that

$$(1 - \eta p_{\text{samp}})^m \geq 1 - \frac{\epsilon}{2}. \quad (29)$$

For (29) to hold, since  $e^{-\epsilon/2} \geq 1 - \epsilon/2$  for  $0 \leq \epsilon \leq 1$ , it suffices to require that

$$(1 - \eta p_{\text{samp}})^m \geq e^{-\epsilon/2}. \quad (30)$$

Taking the logarithm on both sides of (30) shows that (30) is equivalent to

$$m \leq \frac{-\epsilon/2}{\log(1 - \eta p_{\text{samp}})}. \quad (31)$$

Since

$$-\log(1 - \eta p_{\text{samp}}) \leq \sum_{j \geq 1} \eta^j p_{\text{samp}}^j = \eta p_{\text{samp}} / (1 - \eta p_{\text{samp}}) \quad (32)$$

whenever  $\eta p_{\text{samp}} < 1$ , for (31) to hold, it suffices to require that

$$m \leq \frac{\epsilon(1 - \eta p_{\text{samp}})}{2\eta p_{\text{samp}}}. \quad (33)$$

For (33) to hold, using the upper bound  $p_{\text{samp}} \leq 1/(ed)$  from Lemma 2, it suffices to require that

$$m \leq \frac{\epsilon ed(1 - \eta/(ed))}{2\eta}. \quad (34)$$

Using the additive form of the Chernoff bound on  $\eta$ , we get

$$\eta \leq \exp(-\gamma^2 r / (1 - 4\gamma^2)). \quad (35)$$

The trivial upper bound  $\eta \leq 1$  and the Chernoff upper bound on  $\eta$  in (35) imply that for (34) to hold, it suffices to require that

$$m \leq \frac{\epsilon}{2} (ed - 1) \exp(\gamma^2 r / (1 - 4\gamma^2)). \quad (36)$$

For the upper bound (36) on  $m$  to be larger than the lower bound (27) on  $m$ , it suffices to require that

$$r = \left\lceil \frac{1 - 4\gamma^2}{\gamma^2} \log \left( \frac{8ed \log(n/(2\epsilon)) + 8ed^2 \log(ne/d)}{\epsilon(ed - 1)} \right) \right\rceil, \quad (37)$$

and can set  $r$  to be slightly larger to ensure that the conditions in Lemma 2 hold. Evaluating the corresponding lower bound on  $N = rm$  asymptotically then gives the result.  $\square$

## V. CONVERSE BOUND

In this section, assuming  $d = o(\sqrt{n})$ , we derive a lower bound on the minimum number of copies  $N$  of noisy  $n$ -qubit graph states we require to learn it. We prove Theorem 7 by connecting the problem of learning a graph state with that of transmitting classical information over a quantum channel, and counting the asymptotic number of graphs in  $G_{n,d}$ .

**Theorem 7.** Let every query to Oracle() return a noisy graph state  $\mathcal{D}_p^{\otimes n}(|g\rangle\langle g|)$ , where  $0 \leq p < 3/4$ . Let  $0 \leq \epsilon < 1$ . Then as  $n$  becomes large, the minimum number of copies of  $|g\rangle$  required to learn  $|g\rangle$  with correctness at least  $1 - \epsilon$  satisfies the inequality

$$N \geq \frac{d \log_4(nd)}{(1 - H(2p/3))/(1 - \epsilon) + 1/n}, \quad (38)$$

where  $H$  denotes the binary entropy function.

*Proof.* The total number of qubits used to transmit information about our graph state is  $nN$ . The number of bits needed to describe a  $d$ -regular graph is  $|G_{n,d}|$ . Hence we can transmit  $\log_2 |G_{n,d}|$  bits of classical information using  $nN$  qubits, which corresponds to a transmission rate of

$$R = \frac{\log_2 |G_{n,d}|}{nN}. \quad (39)$$

From King's result [21], we know that the classical capacity of the depolarizing channel  $\mathcal{D}_p$  is

$$1 - H(2p/3). \quad (40)$$

From Fano's inequality, if  $\epsilon < 1$ , the optimal  $R$  is at most  $R \leq \frac{\epsilon}{1-\epsilon} + \frac{1}{n}$  for all  $n$ . Hence we must have

$$N \geq \frac{\log_2 |G_{n,d}|}{n(1 - H(2p/3))/(1 - \epsilon) + 1}. \quad (41)$$

From [22], whenever  $d = o(\sqrt{n})$ , we know that

$$|G_{n,d}| = \frac{(nd)!}{(nd/2)!2^{nd/2}(d!)^n} \exp \left( -\frac{d^2 - 1}{4} - \frac{d^3}{12n} + O(d^2/n) \right). \quad (42)$$

Since  $\log(n!) = n \log n - n + O(\log n)$ , we find that

$$\begin{aligned} \log |G_{n,d}| &= nd \log(nd) - nd + O(\log(nd)) \\ &\quad - (nd/2) \log(nd/2) + nd/2 + O(\log(nd/2)) \\ &\quad - (nd/2) \log 2 - n(d \log d - d + O(\log d)) \\ &\quad - \frac{d^2 - 1}{4} - \frac{d^3}{12n} + O(d^2/n). \end{aligned} \quad (43)$$

Simplifying this, we see that whenever  $d = o(\sqrt{n})$ , we have

$$\begin{aligned} \log |G_{n,d}| &= \frac{1}{2} nd \log(nd) + O(nd \log d) \\ &= \frac{1}{2} nd \log(nd) \left( 1 + O \left( \frac{\log d}{\log nd} \right) \right). \end{aligned} \quad (44)$$

Substituting (44) into (41) gives the result.  $\square$

Note that Theorem 7 implies that for constant  $p$  and when  $\epsilon < 1/2$ , we have a lower bound of  $N = \Omega(d \log n)$ .

The lower bound  $N = \Omega(\log(1/\epsilon))$  follows from state discrimination. Consider two graph states in the set that are as close as possible. The probability of making a discrimination error using an optimal strategy is exponentially small (with the quantum Chernoff exponent [23]). Solving for  $N$  gives this dependence.

## VI. DISCUSSIONS

Our analysis of learning graph states using product measurements leaves a number of open problems. Since our converse bound applies for learning algorithms that use arbitrary measurements, this leaves open the question as to whether our algorithm is asymptotically optimal for product measurements. For instance, one question pertaining to our lower bound for  $m$  in Theorem 3 is, whether the quadratic scaling with respect to  $d$  is necessary. For future work, it would also be interesting to know if an adaptive algorithm can asymptotically outperform our randomized algorithm.

## VII. ACKNOWLEDGEMENTS

Y.O. is supported by the Quantum Engineering Programme grant NRF2021-QEP2-01-P06. M.T and Y.O are supported in part by NUS startup grants (R-263-000-E32-133 and R-263-000-E32-731).

## REFERENCES

- [1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, second ed., 2000.
- [2] S. Aaronson, "The learnability of quantum states," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 463, no. 2088, pp. 3089–3114, 2007.
- [3] S. Aaronson, "Shadow tomography of quantum states," *SIAM Journal on Computing*, vol. 49, no. 5, pp. STOC18–368, 2019.
- [4] S. Aaronson, X. Chen, E. Hazan, S. Kale, and A. Nayak, "Online learning of quantum states," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2019, no. 12, p. 124019, 2019.
- [5] H.-Y. Huang, R. Kueng, and J. Preskill, "Predicting many properties of a quantum system from very few measurements," *Nature Physics*, vol. 16, no. 10, pp. 1050–1057, 2020.
- [6] R. Kueng, H. Rauhut, and U. Terstiege, "Low rank matrix recovery from rank one measurements," *Applied and Computational Harmonic Analysis*, vol. 42, no. 1, pp. 88–116, 2017.
- [7] J. Haah, A. W. Harrow, Z. Ji, X. Wu, and N. Yu, "Sample-optimal tomography of quantum states," *IEEE Transactions on Information Theory*, vol. 63, no. 9, pp. 5628–5641, 2017.
- [8] D. Gottesman, "Identifying stabilizer states," aug 2008. <https://pirsa.org/08080052>.
- [9] L. Zhao, C. A. Pérez-Delgado, and J. F. Fitzsimons, "Fast graph operations in quantum computation," *Physical Review A*, vol. 93, no. 3, p. 032314, 2016.
- [10] A. Montanaro, "Learning stabilizer states by bell sampling," *arXiv preprint arXiv:1707.04012*, 2017.
- [11] A. Montanaro and C. Shao, "Quantum algorithms for learning a hidden graph and beyond," *arXiv preprint arXiv:2011.08611*, 2020.
- [12] C.-Y. Lai and H.-C. Cheng, "Learning quantum circuits of some  $t$  gates," *arXiv preprint arXiv:2106.12524*, 2021.
- [13] D. Schlingemann and R. F. Werner, "Quantum error-correcting codes associated with graphs," *Physical Review A*, vol. 65, no. 1, p. 012308, 2001.
- [14] W. Dür, H. Aschauer, and H.-J. Briegel, "Multiparticle entanglement purification for graph states," *Physical review letters*, vol. 91, no. 10, p. 107903, 2003.
- [15] R. Raussendorf, D. E. Browne, and H. J. Briegel, "Measurement-based quantum computation on cluster states," *Physical review A*, vol. 68, no. 2, p. 022312, 2003.

- [16] M. Hein, J. Eisert, and H. J. Briegel, "Multiparty entanglement in graph states," *Physical Review A*, vol. 69, no. 6, p. 062311, 2004.
- [17] D. Schlingemann, "Stabilizer codes can be realized as graph codes," *Quantum Information & Computation*, vol. 2, no. 4, pp. 307–323, 2002.
- [18] M. Van den Nest, J. Dehaene, and B. De Moor, "Local unitary versus local clifford equivalence of stabilizer states," *Physical Review A*, vol. 71, no. 6, p. 062323, 2005.
- [19] B. Zeng, H. Chung, A. W. Cross, and I. L. Chuang, "Local unitary versus local clifford equivalence of stabilizer and graph states," *Physical Review A*, vol. 75, no. 3, p. 032325, 2007.
- [20] C. Godsil and G. F. Royle, *Algebraic graph theory*. Springer, 2001.
- [21] C. King, "The capacity of the quantum depolarizing channel," *IEEE Transactions on Information Theory*, vol. 49, no. 1, pp. 221–229, 2003.
- [22] B. D. McKay and N. C. Wormald, "Asymptotic enumeration by degree sequence of graphs with degrees  $o(n^{1/2})$ ," *Combinatorica*, vol. 11, no. 4, pp. 369–382, 1991.
- [23] K. M. R. Audenaert, J. Calsamiglia, R. Muñoz Tapia, E. Bagan, L. Masanes, A. Acín, and F. Verstraete, "Discriminating states: The quantum chernoff bound," *Phys. Rev. Lett.*, vol. 98, p. 160501, Apr 2007.