Quantum learning robust against noise

Andrew W. Cross,[*] Graeme Smith,[†] and John A. Smolin[‡]

IBM T. J. Watson Research Center, 1101 Kitchawan Road, Yorktown Heights, New York 10598, USA
(Received 23 July 2014; revised manuscript received 30 January 2015; published 27 July 2015)

Noise is often regarded as anathema to quantum computation, but in some settings it can be an unlikely ally. We consider the problem of learning the class of n-bit parity functions by making queries to a quantum example oracle. In the absence of noise, quantum and classical parity learning are easy and almost equally powerful, both information-theoretically and computationally. We show that in the presence of noise this story changes dramatically. Indeed, the classical learning problem is believed to be intractable, while the quantum version remains efficient. Depolarizing the qubits at the oracle's output at any constant nonzero rate does not increase the computational (or query) complexity of quantum learning more than logarithmically. However, the problem of learning from corresponding classical examples is the learning parity with noise problem, for which the best known algorithms have superpolynomial complexity. This creates the possibility of observing a quantum advantage with a few hundred noisy qubits. The presence of noise is essential for creating this quantum-classical separation.

I. INTRODUCTION

A theory of quantum fault tolerance has been erected to overcome pervasive decoherence [1–3]. Without such fault-tolerant machinery, large classes of quantum algorithms can fail to give any significant improvement over classical algorithms [4]. This may lead one to suppose that noise can only rob quantum algorithms of their supremacy or at best increase the cost of running them. However, Burhman, Newman, Rohrig, and de Wolf presented a model of fault-tolerant decision trees where one is given oracle access to input bits perturbed by coherent noise [5]. Among other results, they established a logarithmic separation between quantum and classical query complexity, showing that quantum computers can be more robust than classical computers when computing with noisy inputs. Here we exhibit a problem for which noise is an even more significant classical foe and is crucial to achieving a quantum speed-up, or rather, a classical slow-down. Our problem is believed to be superpolynomially more difficult classically when noise is added, but only logarithmically harder in the quantum case. This challenges the conventional wisdom that quantum computations are inherently delicate while classical computation is more robust.

We consider the problem of learning a class of Boolean functions by making queries to a quantum example oracle [6]. Such an oracle provides a quantum state that encodes a hidden function, and the goal is to discover the function efficiently, meaning with a number of queries and an amount of postprocessing that scales polynomially in the number of input bits. In the quantum setting, we are permitted to apply coherent operations to the quantum state, whereas in the classical setting we must first measure the state in the computational basis before further computation. This model of quantum learning differs from other attempts to use quantum computers to perform machine learning tasks [7–11].

Information-theoretically, quantum learning from queries to ideal oracles is only polynomially more powerful than classical learning [12,13]. Computationally, however, there is a class of functions that is polynomial time learnable from quantum coherent queries but not from classical queries, under the assumption that factoring Blum integers is intractable [12].

In this work, we exhibit a learning problem with a superpolynomial quantum computational speed-up only in the presence of noise. The physical implementation of any oracle on bare qubits will inevitably be noisy. To fairly assess the performance of a quantum algorithm given access to such an oracle, we must compare it to a classical algorithm given access to a noisy classical oracle with similar noise characteristics. To do this, we imagine constructing a noisy classical oracle by completely dephasing the inputs and outputs of a noisy quantum oracle in the computational basis.

For the class of parity functions, we show that depolarizing the example oracle's output at any nonzero rate has a small (logarithmic) effect on the computational complexity of learning from quantum coherent examples. However, the function cannot be learned from classical examples provided by the corresponding noisy classical oracle, as this is equivalent to a problem called learning parity with noise (LPN), for which the best known algorithm has superpolynomial complexity [14]. Both problem settings are tractable without noise, so a quantum advantage is not merely retained; it occurs because of the noise.

The rest of the paper is organized as follows. Section II reviews definitions relevant to quantum learning. Sections III and IV consider the learning problem without and with noise, respectively. Finally, Sec. V concludes.

II. DEFINITIONS

We begin by reviewing relevant definitions. A *membership oracle* for a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ is an oracle that, when queried with input x, outputs the result $f(x)$. It is so called because we can think of $f(x)$ as telling us whether input x belongs to a set associated with the function (namely, the set of inputs that evaluate to 1). A query to a *uniform random example oracle* for a Boolean function f returns an ordered

—————
[*]awcross@us.ibm.com
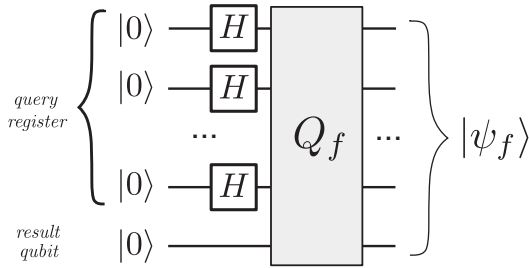[†]gsbsmith@gmail.com
[‡]smolin@us.ibm.com

FIG. 1. One can construct a uniform quantum example oracle from a quantum membership oracle for f. H denotes a Hadamard gate. The quantum learner then performs a quantum computation to identify the function f. The corresponding classical example oracle is obtained from the quantum oracle by measuring its output qubits in their computational bases. The classical learner uses this output together with classical computation to learn the function.

pair $(x, f(x))$ where x is drawn uniformly at random from the set of all possible inputs of f. The membership oracle gives an agent freedom to choose the input, whereas the example oracle merely allows one to "push a button" and request an output.

The problem of learning a class of Boolean functions by querying such oracles can be generalized to a quantum coherent setting [6]. A *quantum membership oracle* Q_f is a unitary transformation that acts on the computational basis states as

$$Q_f : |x, b\rangle \mapsto |x, b \oplus f(x)\rangle, \qquad (1)$$

where $x \in \{0,1\}^n$ and $b \in \{0,1\}$. A *uniform quantum example oracle* for f outputs the quantum state

$$|\psi_f\rangle \equiv \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x, f(x)\rangle. \qquad (2)$$

This oracle only gives the learner freedom to request some number of quantum states, each at unit cost. For both oracles, the *query register* comprises the qubits containing x, and the *result qubit* is the auxiliary qubit containing $f(x)$ (Fig. 1).

Given any quantum oracle, we define a corresponding classical oracle by completely dephasing every interface to the quantum oracle, passing each input or output qubit through a channel $\mathcal{E}_Z(\rho) = (\rho + Z\rho Z)/2$, where

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Any quantum membership oracle becomes a classical membership oracle, and any uniform quantum example oracle becomes a uniform random example oracle. This definition allows us to begin with a noisy quantum oracle and identify a corresponding noisy classical oracle with similar noise characteristics. Equivalently, one can instead completely dephase the *learner's* interface by moving the dephasing channels outside the quantum oracle. One can then define a *classical learner* as a learner who interacts with oracles through dephased interfaces, whereas a *quantum learner* has no such restriction. Classical and quantum learners can now be given access to the same noisy quantum oracle. Note that a classical learner interacting

with a quantum oracle is equivalent to the traditional classical problem of learning from the corresponding classical oracle.

In the domain of learning theory, a *concept* f is a Boolean function $f : \{0,1\}^n \to \{0,1\}$. A *concept class* $\mathfrak{C} = \cup_{n \geqslant 1} C_n$ (hereafter, *class*) is a collection of concepts, and each C_n contains the concepts whose domain is $\{0,1\}^n$. Given a *target concept* $f \in \mathfrak{C}$, a typical goal is to construct a *hypothesis* function $h : \{0,1\}^n \to \{0,1\}$ that agrees with f on at least a $1 - \epsilon$ fraction of the inputs in $\{0,1\}^n$, i.e.,

$$\Pr_x[h(x) = f(x)] \geqslant 1 - \epsilon, \qquad (3)$$

where x is drawn from the uniform distribution. Such a function is called an ϵ *approximation of* f.

A class $\mathfrak{C}$ is *efficiently PAC (probably approximately correct [15]) learnable under the uniform distribution* if given a uniform example oracle for any target concept $f \in \mathfrak{C}$, there is an algorithm that

(1) for any $\epsilon, \delta \in (0, 1/2)$, outputs an ϵ approximation h of f with probability $1 - \delta$,

(2) runs in time and uses a number of queries, that is poly$(n, 1/\epsilon, 1/\delta)$.

The definition of learning is identical in the quantum setting except that the uniform example oracle is replaced by a uniform quantum example oracle, and the allowed computations may be coherent. We will also consider example oracles corrupted by noise of constant rate $\eta < 1/2$ in a way that is defined later. The definition of learning is unchanged in this case, although one may require the algorithm to run in time poly$(1/(1/2 - \eta))$ as well.

We now restrict the discussion to the class of parity functions,

$$f_a(x) = \langle a, x \rangle = \sum_{j=1}^n a_j x_j \bmod 2, \qquad (4)$$

where $a \in \{0,1\}^n$ and a_j (x_j) denotes the jth bit of a (x). We are given access to a uniform quantum example oracle for the unknown concept $f_a(x) = \langle a, x \rangle$. If we incorrectly guess even a single bit of a, our hypothesis function is a $1/2$ approximation to f_a and remains so for any number of incorrect bits. Therefore, we must with high probability find a exactly (i.e., probably *exactly* correct learning), and this is what we require hereafter.

III. LEARNING FROM IDEAL QUERIES

First, consider the noiseless case where each query returns a pure quantum state. This case is tractable for both quantum and classical queries as we now review.

The classical oracle provides an example $(x, f_a(x))$ where x is uniformly random over $\{0,1\}^n$. Since $f_a(x)$ is a linear function, it is clear that n queries are sufficient to learn f_a exactly with a constant probability of success. The probability that n queries produce linearly independent examples is

$$\prod_{j=0}^{n-1} (1 - 2^{j-n}), \qquad (5)$$

which is greater than $1/4$ for any $n > 1$. Any algorithm that detectably fails with constant probability less than p and

otherwise succeeds can be repeated no more than $\log_{1/p}(1/\delta)$ times to reduce the failure probability below δ. The value of a is obtained from the examples by Gaussian elimination.

In the quantum setting, f_a can be learned with constant probability from a single query. Given $|\psi_f\rangle$, apply Hadamard gates

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad (6)$$

to each of the $n + 1$ output qubits. A simple calculation shows that the output state becomes

$$\frac{1}{\sqrt{2}}(|0^n,0\rangle + |a,1\rangle). \qquad (7)$$

Therefore, with probability $1/2$, measurement reveals the value of a directly in the query register whenever the result qubit is 1. Again the probability of success can be amplified with $O(\log(1/\delta))$ queries.

Note that this is very similar to the Bernstein-Vazirani algorithm [16] but adapted to use an example oracle rather than a membership oracle. The only difference is our treatment of the final qubit. In the Bernstein-Vazirani algorithm, the result qubit is input as a $|-\rangle$ state and will, with certainty in the noiseless case, end up as a $|1\rangle$. It then does not even need to be measured. For the example oracle we have considered, we do not have the luxury of choosing the input, so we simply check that the result qubit is $|1\rangle$, which collapses the output state of the other n qubits to the result of the Bernstein-Vazirani oracle.

IV. LEARNING IN THE PRESENCE OF NOISE

Now we consider how the situation changes when we add noise to the output of the example oracle. We will see that learning parity from a noisy example oracle seems to become computationally intractable, while the same task with a noisy quantum example oracle can be solved efficiently on a quantum computer. We will first consider a simple case that is easy to analyze followed by the more realistic case of depolarizing noise.

A. Classification noise

Just about the most trivial model of noise one can imagine is this one: flip the result qubit with probability η by applying the Pauli $\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ operator. Classically, learning f_a from such corrupted results is called *learning parity with noise* (LPN). The LPN problem is equivalent to decoding a random linear code in the presence of stochastic noise [14]. In the worst case, decoding linear codes is NP hard [17] and also hard to solve approximately [18]. The LPN problem setting is reminiscent of the Goldreich-Levin (GL) theorem [19], but there one is free to choose the queries to an oracle that lies on the answers to some fixed fraction of the queries, but must lie consistently on repeated queries. This restriction allows efficient solution of the GL problem. On the other hand, the LPN problem is believed to be computationally intractable [20] and potential cryptographic applications have been proposed for this problem and its generalizations [21]. The best known algorithms for LPN are subexponential (but superpolynomial)

in n [14,22]. Problem instances with hundreds of bits may be impractical to solve [23].

However, the quantum case remains easy. With noise, the output of the oracle transformed by Hadamards becomes the mixture of (7) with probability $1 - \eta$ and

$$\frac{1}{\sqrt{2}}(|0^n,1\rangle + |a,0\rangle) \qquad (8)$$

with probability η. The probability that the query register contains a remains $1/2$, independent of η. Thus, after k queries, the probability of observing a is $1 - (1/2)^k$. This suggests the simple strategy of reporting either $a = $ "whatever nonzero result is seen" or 0^n otherwise. It fails with a probability that is exponentially small in k and independent of n.

This strategy is strictly suboptimal since it ignores the information contained in the result qubit. The fact that it works so well, regardless, suggests that our noise model is rather unfair to the classical case by degrading the result qubit, which the quantum algorithm hardly even needs. Indeed, in the Bernstein-Vazirani algorithm, the output bit is not measured at all. Still, this simple case serves to illustrate how noise can more severely impact the classical learner. Even in the more realistic noise model we consider next, the quantum algorithm survives the addition of significant amounts of noise because the quantum queries reveal so very much.

B. Depolarizing noise

Now we consider the case where the output of the oracle is subject to independent depolarizing noise $D_\eta^{\otimes(n+1)}$ where $D_\eta(\rho) = (1 - 2\eta)\rho + 2\eta I/2$ and $\eta < 1/2$ is a constant known noise rate. This noise process is an idealization of realistic independent noise and corrupts the ideal output of the oracle with probability proportional to η.

Classically, in the presence of this noise, we obtain examples $(x \oplus e_{1:n}, f_a(x) \oplus e_{n+1})$ where x is uniformly random over $\{0,1\}^n$ and each bit of the noise $(e_{1:n}, e_{n+1})$ is 1 with probability η. Here $e_{1:n} \in \{0,1\}^n$ and $e_{n+1} \in \{0,1\}$. Since x is uniformly random, $x' = x \oplus e_{1:n}$ is uniformly random as well and

$$(x \oplus e_{1:n}, f_a(x) \oplus e_{n+1}) = (x', f_a(x' \oplus e_{1:n}) \oplus e_{n+1}). \qquad (9)$$

The probability that $f_a(x' \oplus e_{1:n}) \neq f_a(x')$ depends on the value of a and is given by

$$\zeta_a = \sum_{w=1}^{n} \sum_{k=1,\text{odd}}^{w} \binom{|a|}{k} \binom{n-|a|}{w-k} \eta^w (1-\eta)^{n-w}, \qquad (10)$$

where $|a|$ denotes the number of 1's in a (also called the Hamming weight). The total probability of error on the result bit is simply $\eta' := \eta(1 - \zeta_a) + \zeta_a(1 - \eta)$. Therefore, learning from these examples is the LPN problem with noise rate $\eta' \in [\eta, 1 - \eta]$.

In contrast, coherent manipulation of the noisy output state of the example oracle allows a quantum learner to learn f_a in a number of queries that is logarithmic in n. The algorithm is as follows. Make $k = O(\log n)$ queries to the example oracle, and for each query, Hadamard transform all $n + 1$ noisy output qubits and measure them to obtain an outcome. Each outcome has the form (m,b) where $m \in \{0,1\}^n$ is a result string in the

query register and the result bit b is uniformly random. Discard the outcome if $b = 0$ and otherwise retain the result string m. We are left with k' result strings $m_1, m_2, \ldots, m_{k'}$ on which we perform a bit-wise majority vote to obtain an estimate $\hat{a}$ of a.

We will now argue that the estimate $\hat{a}$ obtained from this protocol is equal to a for appropriately chosen parameters. Given any constant $\delta > 0$, the algorithm must find an estimate $\hat{a}$ such that $\Pr[\hat{a} \neq a] < \delta$. We make repeated use of a loose form of the Chernoff bound,

$$\Pr[|X - \eta k| < \delta \eta k] > 1 - 2e^{-\delta^2 \eta k/3} \equiv 1 - B_k(\eta, \delta), \quad (11)$$

where X is the sum of k independent Bernoulli random variables with $\Pr(1) = \eta$ and $0 < \delta < 1$. Clearly we can query the oracle until we retain a total of k' result strings. This takes $2k'$ expected queries. By performing, say, $3k'$ queries, we are guaranteed via Eq. (11) to retain fewer than k' with probability exponentially small in k'. Now, let D_q^η be the probability distribution over $\{0,1\}^n$ that corresponds to the bit string q corrupted by independent bit-flip noise of rate η. The retained strings are drawn from the distribution $(1 - \eta)D_a^\eta + \eta D_{0^n}^\eta$. Let s be the random variable giving the unknown number of strings drawn from D_a^η. These successful queries, which we take to be $m_1, m_2, \ldots, m_s$ without loss of generality, contain information about the hidden function. The expected value of s is $\mu_s = (1 - \eta)k'$, and its variation from this mean is controlled by

$$\Pr[|s - \mu_s| < \delta' \mu_s] > 1 - B_{k'}(1 - \eta, \delta'). \quad (12)$$

Our algorithm votes independently on the jth bits of the strings for each $j = 1, 2, \ldots, n$. Let M_j be the random variable corresponding to the sum $(m_1)_j + (m_2)_j + \cdots + (m_{k'})_j$ of the jth bits. The worst case occurs when $a_j = 1$ and we assume this. Define random variables $M_j^{(a)} = m_1 + \cdots + m_s$ and $M_j^{(0)} = M_j - M_j^{(a)}$ with means $\mu_j^{(a)} = (1 - \eta)s$ and $\mu_j^{(0)} = \eta(k' - s)$, respectively. The mean of M_j is $\mu_j = \mu_j^{(a)} + \mu_j^{(0)}$. Conditioned on obtaining a typical value of s, the probability of a successful vote on the jth bit is

$$\gamma_j \geqslant \Pr\left[M_j > \frac{k'}{2}\right] \geqslant \Pr[|M_j - \mu_j| < \delta' \mu_j] \quad (13)$$

$$\geqslant \Pr\left[|M_j^{(a)} - \mu_j^{(a)}| < \delta' \mu_j^{(a)}\right] \quad (14)$$

$$\times \Pr\left[|M_j^{(0)} - \mu_j^{(0)}| < \delta' \mu_j^{(0)}\right] \quad (15)$$

$$> 1 - 2B_{k'}(\tilde{\eta}, \delta'). \quad (16)$$

We have defined $\tilde{\eta} = \eta[1 - (1 + \delta')(1 - \eta)]$ and chosen $\delta' < \eta/(1 - \eta)$. The second inequality of (13) follows by further choosing $1 - \delta' > \frac{1}{2}[(1 - 2\eta)(1 - \eta) + \eta]^{-1}$. To find (16), we used $\tilde{\eta} \leqslant (1 - \delta')(1 - \eta)^2$. This gives us an upper bound

$$\Pr[\hat{a}_j \neq a_j] = 1 - \gamma_j < 2B_{k'}(\tilde{\eta}, \delta'), \quad (17)$$

on the probability of the jth bit being computed incorrectly, which we can use together with a union bound to find

$$\Pr[\hat{a} \neq a] \leqslant \sum_j \Pr[\hat{a}_j \neq a_j] < 2n B_{k'}(\tilde{\eta}, \delta'). \quad (18)$$

Choosing

$$k' > \frac{3}{(\delta')^2 \tilde{\eta}} \ln\left(\frac{4n}{\delta}\right) \quad (19)$$

ensures that $\Pr[\hat{a} \neq a] < \delta$. For sufficiently large η, one can verify easily that $(\delta')^2 \tilde{\eta}$ is a polynomial in $(1/2 - \eta)$, and therefore $k' = O(\text{poly}(1/(1/2 - \eta)))$.

C. More general noise

Finally, we consider the case of general noise on the output of the example oracle. Because the noise is the last thing that happens before the output is measured, one can think of the situation as the result of a noiseless oracle as viewed by a noisy measurement. For any noise processes, the classical learner will instead of getting results $x, f_a(x)$ get $x', f_a'(x)$ where $x \to x'$ and $f_a(x) \to f_a'(x)$ depend on the details of the noise. For any noise where $f_a'(x) \neq f_a(x')$ this results in classification noise which is, as we have said above, believed to be inefficient in the number of queries. Almost any noise will be of this form unless it is extremely small or consists only of dephasing, which is noise the classical learner performs by definition in any event.

For the quantum learner, the voting algorithm given for depolarizing noise will still work for a wide variety of noise processes. Again, each example results in a string (m, b). So long as b is 1 with probability scaling no worse than $1/n$, and whenever it is m differs from the correct result on fewer than $n/2$ bits, the vote on a logarithmic number of queries will still give the correct answer with high probability.

This is a rather operational specification of which types of noise will remain easy for the quantum learner, and indeed would need to be analyzed for the actual noise present in any particular experiment. To see that this specification includes a wide range of noise processes, consider any independent noise which can be "twirled" [24] into a depolarizing channel of $\eta < 1/2$. All such channels are no worse for the quantum learner than the corresponding depolarizing channel.

V. CONCLUSION

We have defined the problem of quantum learning from a noisy quantum example oracle and shown that the class of parity functions can be learned in logarithmic time from corrupted quantum queries. In contrast, it appears to be intractable to learn this class in polynomial time from classical queries to the corresponding classical noisy oracle [25]. If the oracle is ideal, the problem is tractable for both quantum and classical learners, so the noise plays an essential role in the exhibited behavior. For this problem at least, decoherence is an ally of quantum computation.

The example oracle for parity can be implemented in practice with $O(n)$ one- and two-qubit gates. The quantum learner then needs only single-qubit gates and measurements, or even just measurements in a nonstandard basis. This suggests that an experimental demonstration may be quite practicable. The independent depolarizing noise model we use is an idealization of realistic decoherence. Although a more detailed study of actual experimental noise would be needed, it should be possible to demonstrate quantum supremacy for

learning using several hundred *noisy* qubits, i.e., without the use of quantum error correction. In the meantime, since a classical learner requires at least n queries in the noiseless case while the quantum learner needs only $O(\log n)$ queries with noise, a quantum advantage for query complexity, while small, could be shown experimentally in existing systems.

[1] D. Aharonov and M. Ben-Or, Fault tolerant quantum computation with constant error, in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, 1998), pp. 176–188.

[2] A. Yu. Kitaev, Quantum computations: Algorithms and error correction, Russ. Math. Surv. **52**, 1191 (1997).

[3] P. Aliferis, D. Gottesman, and J. Preskill, Quantum accuracy threshold for concatenated distance-3 codes, Quantum Inf. Comput. **6**, 97 (2006).

[4] O. Regev and L. Schiff, Impossibility of a quantum speed-up with a faulty oracle, in *Proceedings of the 35th International Colloquium on Automata, Languages, and Programming (ICALP)* (Springer, Berlin, Heidelberg, 2008), pp. 773–781.

[5] H. Buhrman, I. Newman, H. Rohrig, and R. de Wolf, Robust polynomials and quantum algorithms, in *Proceedings of the 22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, Lecture Notes in Computer Science, Vol. 3404 (Springer, Berlin, Heidelberg, 2005), pp. 593–604.

[6] N. Bshouty and J. C. Jackson, Learning dnf over the uniform distribution using a quantum example oracle, SIAM J. Comput. **28**, 1136 (1999).

[7] D. Anguita, S. Ridella, F. Rivieccio, and R. Zunino, Quantum optimization for training support vector machines, Neural Networks **16**, 763 (2003).

[8] K. Pudenz and D. Lidar, Quantum adiabatic machine learning, Quantum Inf. Process. **12**, 2027 (2013).

[9] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, arXiv:1307.0411.

[10] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification, Phys. Rev. Lett. **113**, 130503 (2014).

[11] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum nearest-neighbor algorithms for machine learning, Quantum Inf. Comput. **15**, 0318 (2015).

[12] R. Servedio and S. Gortler, Equivalences and separations between quantum and classical learnability, SIAM J. Comput. **33**, 1067 (2004).

[13] A. Atici and R. Servedio, Improved bounds on quantum learning algorithms, Quantum Inf. Process. **4**, 355 (2005).

[14] V. Lyubashevsky, The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem, *Lecture Notes in Computer Science (8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX))*, Vol. 3624 (Springer, Berlin, Heidelberg, 2005), p. 378.

[15] L. G. Valiant, A theory of the learnable, Commun. ACM **27**, 1134 (1984).

[16] E. Bernstein and U. Vazirani, Quantum complexity theory, in *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC '93 (ACM, New York, 1993), pp. 11–20.

[17] E. R. Berlekamp, R. J. McEliece, and H. Van Tilborg, On the inherent intractability of certain coding problems, IEEE Trans. Inf. Theory **24**, 384 (1978).

[18] J. Hastad, Some optimal inapproximability results, in *Proceedings of the 29th Annual Symposium on Theory of Computing*, Vol. 48 (ACM, New York, NY, 2001), pp. 798–859.

[19] O. Goldreich and L. Levin, Hard-core predicates for any one-way function, in *Proceedings of the 21st Annual Symposium on Theory of Computing* (ACM, New York, NY, 1989), pp. 25–32.

[20] There is strong evidence that a generalization called learning with errors (LWE) is hard, but it remains open to establish such results for LPN [21].

[21] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, in *Proceedings of the 37th Annual Symposium on Theory of Computing*, Vol. 56 (ACM, New York, NY, 2005), pp. 84–93.

[22] A. Blum, A. Kalai, and H. Wasserman, Noise-tolerant learning, the parity problem, and the statistical query model, J. ACM **50**, 506 (2003).

[23] E. Levieil and P. Fouque, An improved lpn algorithm, in *Proceedings of the 5th International Conference on Security and Cryptography for Networks* (Springer, Berlin, Heidelberg, 2006), pp. 348–359.

[24] C. H. Bennett, D. P. Divincenzo, J. A. Smolin, and W. K. Wootters, Mixed-state entanglement and quantum error correction, Phys. Rev. A **54**, 3824 (1996).

[25] In fact, even a quantum computer given access to this classical noisy example oracle is not known to be able to efficiently learn this class.