Statistical limits of supervised quantum learning

Carlo Ciliberto,[1] Andrea Rocchetto,[2,3] Alessandro Rudi,[4] and Leonard Wossnig[5,6]

[1]*Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2BT, United Kingdom*
[2]*Department of Computer Science, University of Texas at Austin, Austin, Texas 78712, USA*
[3]*Kavli Institute for Theoretical Physics, University of California, Santa Barbara, California 93106, USA*
[4]*INRIA - Sierra Project Team, 75012 Paris, France*
[5]*Department of Computer Science, University College London, London WC1E 6EA, United Kingdom*
[6]*Rahko Limited, London N4 3JP, United Kingdom*

(Received 11 March 2020; accepted 13 October 2020; published 28 October 2020)

Within the framework of statistical learning theory it is possible to bound the minimum number of samples required by a learner to reach a target accuracy. We show that if the bound on the accuracy is taken into account, quantum machine learning algorithms for supervised learning—for which statistical guarantees are available—cannot achieve polylogarithmic runtimes in the input dimension. We conclude that, when no further assumptions on the problem are made, quantum machine learning algorithms for supervised learning can have at most polynomial speedups over efficient classical algorithms, even in cases where quantum access to the data is naturally available.

DOI: 10.1103/PhysRevA.102.042414

*Introduction.*A wide class of quantum algorithms for supervised learning problems (where the goal is to infer a mapping given examples of an input-output relation) exploit fast quantum linear algebra subroutines to achieve runtimes that are exponentially faster than their classical counterparts [1,2]. Examples of these algorithms are quantum support vector machines [3], quantum linear regression [4,5], and quantum least squares [6,7].

A careful analysis of these algorithms identified a number of caveats that limit their practical applicability such as the need for a strong form of quantum access to the input data, restrictions on structural properties of the data matrix (such as condition number or sparsity), and modes of access to the output [8]. Furthermore, if one assumes that it is efficient to (classically) sample elements of the training data in a way proportional to their norm, then it is possible to show that classical algorithms are only polynomially slower (albeit the scaling of the quantum algorithms can be considerably better) [9–13].

In this paper, we continue to investigate the limitations of quantum algorithms for supervised learning problems. Our analysis focuses on the dependency on the size of the data set that is introduced when considering the statistical guarantees of the estimators. The key elements of our work are a series of well-known results in statistical learning theory that show how the accuracy parameter of a supervised learning problem scales inverse polynomially with the number of samples in the training set. We leverage on these insights to show that quantum learning algorithms must have at least a polynomial runtime in the dimension of the training data and therefore cannot achieve exponential speedups over classical polynomial time machine learning algorithms. We remark that our results do not rule out exponential advantages for the learning problem where no efficient classical algorithms are known. In fact, in this regime, there exist learning problems for which quantum algorithms have a superpolynomial advantage [14,15].

Our results are independent of the modes of access to the training data, that is, even if the data set is naturally stored in a quantum structure, quantum machine learning algorithms can have at most a polynomial advantage over their classical variants.

Finally, we note that our results do not assume any prior knowledge on the function to be learned. This allows us to make statements on virtually every possible learning algorithm, including neural networks. Using stronger assumptions on the target function it is possible to improve the dependency of the accuracy in the number of samples (consider the limit case where the function is known; in this case, zero samples can determine the function with maximum accuracy).

Statistical learning theory. The field of statistical learning theory investigates how to quantify the statistical resources required to solve a learning problem [16]. In this work, we consider supervised learning settings where the goal is to find a model that fits well a set of input-output training examples but that, more importantly, guarantees good prediction performance on new observations. This latter property, also known as the *generalization capability* of the learned model, is a key aspect separating machine learning from the standard optimization literature. Indeed, while data fitting is often approached as an optimization problem in practice, the focus of machine learning is to design statistical estimators able to "fit" well future examples.

More formally, let ρ be a distribution over $X \times Y$, with X a so-called *domain (or input) set* and Y a *label (or output) set*. The goal of supervised learning is to produce a hypothesis

$f : X \to Y$ such that the *expected risk* or *expected error*

$$\mathcal{E}(f) := \mathbb{E}_{\rho}[\ell(y, f(x))] \tag{1}$$

is small with respect to a suitable *loss function* $\ell : Y \times Y \to \mathbb{R}$ measuring prediction errors. However, in practice, the target distribution ρ is unknown and only accessible by means of a finite *training set* $S_n = \{(x_i, y_i), i = 1, \dots, n\}$ of independent and identically distributed (i.i.d.) points sampled from it. Depending on whether the label set Y is dense or discrete the task is called *regression* (dense) or *classification* (discrete). Typical loss functions are the quadratic loss $\ell_{\text{sq}}(f(x), y) = (f(x) - y)^2$ over $Y = \mathbb{R}$ for regression and the $0 - 1$ loss $\ell_{0-1}(f(x), y) = \mathbf{1}_{f(x) \neq y}$ over $Y = \{-1, 1\}$ for classification.

Different machine learning frameworks have different prescriptions on how to choose the hypothesis f. The *empirical risk minimization* (ERM) framework prescribes to choose a hypothesis that minimizes the empirical risk

$$\inf_{f \in \mathcal{H}} \hat{\mathcal{E}}(f), \quad \hat{\mathcal{E}}(f) = \frac{1}{n} \sum_{(x_i, y_i) \in T} \ell(y_i, f(x_i)), \tag{2}$$

over a suitable *hypotheses space* $\mathcal{H}$. Under weak assumptions on $\mathcal{H}$ (for instance, a bounded subset of a Hilbert space [16]), it is possible to guarantee the existence of a minimizer for Eq. (2) that we denote $\hat{f} = \arg\min_{f \in \mathcal{H}} \hat{\mathcal{E}}(f)$.

The difference between risk and empirical risk is called the *generalization error* and plays a central role in statistical learning theory. Indeed, when Eq. (1) admits a minimizer in $\mathcal{H}$, we have

$$\mathcal{E}(\hat{f}) - \inf_{f \in \mathcal{H}} \mathcal{E}(f) \leqslant 2 \sup_{f \in \mathcal{H}} |\hat{\mathcal{E}}(f) - \mathcal{E}(f)|. \tag{3}$$

In other words, the *excess risk* incurred by the empirical risk minimizer is controlled by the worse generalization error over $\mathcal{H}$. A fundamental result in statistical learning theory [16–18], often referred to in the literature as the fundamental theorem of statistical learning, is that for every $n \in \mathbb{N}$, $\delta \in (0, 1)$, and every distribution ρ, the following holds with probability larger than $1 - \delta$,

$$\sup_{f \in \mathcal{H}} |\hat{\mathcal{E}}(f) - \mathcal{E}(f)| \leqslant \Theta\left(\sqrt{\frac{c(\mathcal{H}) + \log(1/\delta)}{n}}\right), \tag{4}$$

where $c(\mathcal{H})$ is a measure of the complexity of $\mathcal{H}$ [such as the Vapnik-Chervonenkis (VC) dimension, covering numbers, the Rademacher complexity, to name a few [16,19]]. Intuitively, the dependency on $c(\mathcal{H})$ in Eq. (4) models the phenomenon known as *overfitting* in which a large hypothesis space incurs in low training (empirical) error but performs poorly on the true risk. This problem can be addressed with so-called *regularization techniques*, which essentially limit the expressive power of the learned estimator in order to avoid overfitting the training data set.

Different regularization strategies have been proposed in the literature (see Refs. [17,20,21] for an introduction to the main ideas), and one of the well-established approaches which directly imposes constraints on the hypotheses class of candidate predictors is the Tikhonov regularization. Regularization ideas have led to popular machine learning approaches which are widely used in practice, such as regularized least squares [19], Gaussian process (GP) regression and classification [22], logistic regression [20], and support vector machines (SVMs) [17]. All these algorithms can be studied within the framework of kernel methods [23].

From a computational perspective, these approaches compute a solution for the learning problem by optimizing over the constraint objective, which typically consists of a sequence of standard linear algebra operations such as matrix multiplication and inversion. For most classical algorithms, such as GP or SVM, the computational time is therefore $O(n^3)$, which is similar to the time it requires to invert a square matrix that has a size equal to the number n of examples in the training set. Notably this can be improved depending on the sparsity and the conditioning of the specific optimization problem.

To reduce the computational cost, instead of considering the optimization problem as a separate process from the statistical one, more recent methods hinge on the intuition that reducing the computational burden of the learning algorithm can be interpreted as a form of regularization on its own. For instance, *early stopping* approaches are now widely used in practice, and perform only a limited number of steps of an iterative optimization algorithm, to avoid overfitting the training set. They thereby entail less operations, while provably maintaining the same generalization error of approaches such as Tikhonov regularization [21]. More specifically, prototypical results (such as Ref. [21]) show that the number of iterations required are of the order of $1/\lambda$ where λ is the ideal regularization parameter that one would use for ERM. Therefore, if in the worst-case scenario $\lambda = O(1/\sqrt{n})$, early stopping would attain (up to constants) the same generalization error of regularized ERM by performing only $\sqrt{n}$ iterations.

A different approach, also known as *divide and conquer*, is based on the idea of distributing portions of the training data onto separate machines, each solving a smaller learning problem, and then combining individual predictors into a joint one. This computation hence benefits from both the parallelization and the reduced dimension of distributed data sets while similarly maintaining statistical guarantees [24].

A third approach that has recently received significant attention from the machine learning community, along with the quantum computing community, is based on random subsampling, a form of dimensionality reduction. Depending on how such sampling is performed, different methods have been proposed, the most well known being random features [25] and Nyström approaches [26,27]. Here, the computational advantage is clearly given by the smaller dimensionality of the hypotheses space, and it has also recently been shown that it is possible to obtain an equivalent generalization error to classical methods in these settings [28].

For all these methods, training times can be typically reduced from the $O(N^3)$ of standard approaches to $\widetilde{O}(N^2)$ or $\widetilde{O}(Nz)$, where z is the number of nonzero entries, while keeping the statistical performance of the learned estimator essentially unaltered.

Quantum learning algorithms. Linear algebra subroutines are a central computational element of learning algorithms. A large class of quantum algorithms for supervised learning problems claims exponential speedups compared to classical algorithms by making use of fast quantum linear algebra subroutines [3–7,29,30]. One widely used algorithm is the quantum linear system solver [31] (also known as HHL after the three authors Harrow, Hassidim, and Lloyd). The algorithm takes as input a quantum encoding of the vector $b \in \mathbb{R}^n$ and an s-sparse matrix $A \in \mathbb{R}^{n\times n}$, with $\|A\| \leqslant 1$, and outputs an approximation $|\tilde{w}\rangle$ of the solution $|w = A^{-1}b\rangle$ of the linear system such that

$$\||\tilde{w}\rangle - |w\rangle\| \leqslant \gamma \tag{5}$$

for an error parameter $\gamma > 0$. The current best implementation of the algorithm runs in time [7]

$$O(\|A\|_F\, \kappa \,\mathrm{polylog}(\kappa, n, 1/\gamma)), \tag{6}$$

where $\|A\|_F$ is the Frobenius norm of A and κ its condition number. Note that the HHL algorithm requires us to access the data matrix $A \in \mathbb{R}^{n\times d}$ in $O(\mathrm{polylog}(nd))$ time. All the quantum learning algorithms we discuss in this paper inherit this assumption. Recently, it was proven that such strong oracular assumptions (when the data matrix is low rank) also lead to exponentially faster classical algorithms [9,10,12,13]. We recommend Refs. [2,8] for more detailed discussions of the limits of quantum learning algorithms based on fast linear algebra subroutines.

Before proceeding to the statistical analysis of quantum learning algorithms we review some quantum algorithms for the least-squares problem. These will serve as the main examples in our analysis.

Quantum least squares. Least squares is an algorithm for minimizing the empirical risk, with respect to the squared loss, for the hypothesis class of linear functions. More specifically, let $X = \mathbb{R}^d$ and $Y = \mathbb{R}$, and let $\mathcal{H} := \{f : X \to Y \mid \exists\, w \in \mathbb{R}^d : f(x) = w^T x\}$ be the hypothesis class of linear functions. The empirical risk is

$$\hat{\mathcal{E}}(f) := \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - y_i)^2. \tag{7}$$

We can minimize the empirical risk by setting its gradient to zero. Using $X := \sum_i x_i x_i^T$ and $b := \sum_i y_i x_i$ one can write a closed-form solution to the least-squares problem as $w = X^{-1}b$.

Several quantum algorithms for least squares (or, more generally, linear regression problems) have been proposed [4,6,7,29,30]. A common feature is that they use a fast quantum linear system algorithm to find a quantum encoding $|w\rangle$ of the solution vector $w = X^{-1}b$. The fastest known algorithm in the class [7], which improves the dependency on the error from polynomial to logarithmic, solves the (regularized) least squares or linear regression problem in time,

$$O(\|X\|_F\, \kappa \,\mathrm{polylog}(n, \kappa, 1/\gamma)), \tag{8}$$

where κ is the condition number of X and $\gamma > 0$ is an approximation parameter. As for every other quantum algorithm discussed in this paper the quantum least-squares solver requires a quantum-accessible data structure. The dependency on the Frobenius norm implies that it is possible to obtain a speedup only when X is low rank (but nonsparse). Due to approximation errors, the output of the algorithm is not $|w\rangle$ but a quantum state $|\tilde{w}\rangle$, such that $\||\tilde{w}\rangle - |w\rangle\| \leqslant \gamma$.

It is possible to get rid of the dependency on the Frobenius norm using the sample-based Hamiltonian simulation method [32,33]. Leveraging this technique, Ref. [5] proposed a least-squares algorithm whose scaling does not depend on the Frobenius norm but requires a higher number of copies (with respect to Ref. [7]) of the input density matrix. Note that, because the algorithm in Ref. [5] is posed in the query model, i.e., the computational complexity is given in the number of calls to the oracle which returns the data already encoded in the form of a quantum state, it is not possible to make a direct comparison between the two algorithms. The computational complexity of the algorithm given in Ref. [5] is

$$O(\kappa^2\gamma^{-3}\mathrm{polylog}(n)), \tag{9}$$

and the dependency on the error is polynomial.

Quantum speedups and statistical bounds. In this section we analyze the speedup claims of quantum machine learning algorithms using the framework of statistical learning theory. Our main point is that if one considers the $\Theta(n^{-1/2})$ scaling of the generalization error—see Eq. (4)—quantum learning algorithms cannot achieve a polylogarithmic runtime in n.

The starting point of our discussion is the following standard error decomposition. Consider a hypothesis f. We want to bound how far the generalization error of f is from the best possible generalization error; this is known as the *Bayes risk* and is indicated by $\mathcal{E}^* := \inf_{f\in\mathcal{F}} \mathcal{E}(f)$, where $\mathcal{F}$ denotes the set of all measurable functions $f : X \to Y$. We want to decompose this general error into different components and for this reason we introduce $\mathcal{E}_\mathcal{H} := \inf_{f\in\mathcal{H}} \mathcal{E}(f)$, that is, the best risk attainable by a function in the hypothesis space $\mathcal{H}$. In order to simplify our discussion let us assume that $\mathcal{E}_\mathcal{H}$ always admits a minimizer $f_\mathcal{H} \in \mathcal{H}$ (it is possible to levy this assumption using the theory of regularization). Recalling that $\hat{\mathcal{E}}(\hat{f}) := \inf_{f\in\mathcal{H}} \hat{\mathcal{E}}(f)$, we can decompose the total error as

$$\mathcal{E}(f) - \mathcal{E}^* = \underbrace{\mathcal{E}(f) - \mathcal{E}(\hat{f})}_{\text{Optimization error}} + \underbrace{\mathcal{E}(\hat{f}) - \mathcal{E}_\mathcal{H}}_{\text{Estimation error}} + \underbrace{\mathcal{E}_\mathcal{H} - \mathcal{E}^*}_{\text{Irreducible error}} \tag{10}$$

$$= \xi + \Theta(1/\sqrt{n}) + \mu. \tag{11}$$

The first term in Eq. (10) is the *optimization error* and measures how good is the optimization that generated f with respect to the ideal minimization of the empirical risk. This error is related to the approximation error of the algorithm. The second term is the *estimation error* and models the error that we make by estimating the true risk using samples from the distribution ρ. This is the generalization bound we discussed in Eq. (4). The third term is the *irreducible error* and measures how well the hypothesis space models the problem. It is an irreducible source of error that we indicate with the letter μ. If the irreducible error is zero, then we say that $\mathcal{H}$ is universal. For simplicity, we assume throughout the paper that $\mu = 0$.

From the error decomposition in Eq. (10) we see that in order to have an algorithm with optimal statistical performance we must make sure that the optimization error is not larger than the estimation error. Therefore the optimization error must scale at most as the best estimation error. If it does, we say that the optimization error matches the bound of the estimation error.

In order to make the notion of matching the bound more concrete, let us consider again the case of least squares. The

TABLE I. Summary of time complexities for training and testing of different classical and quantum algorithms when statistical guarantees are taken into account. We omit polylog(n, d) dependencies for the quantum algorithms. We assume that the generalization error scales as $\Theta(1/\sqrt{n})$ and count the effects of measurement errors. The acronyms in the table refer to support vector machines (SVM), kernel ridge regression (KRR), quantum kernel least squares (QKLS), quantum kernel linear regression (QKLR), and quantum support vector machines (QSVM). Note that for quantum algorithms the state obtained after training cannot be maintained or copied and the algorithm must be retrained after each test round. This brings a factor proportional to the train time in the test time of quantum algorithms. Because the condition number may also depend on n and for quantum algorithms this dependency may be worse, the overall scaling of the quantum algorithms may be slower than the classical.

	Algorithm	Train time	Test time
Classical	SVM/KRR	n^3	n
	KRR [34–38]	n^2	n
	Divide and conquer [24]	n^2	n
	Nyström [27,28]	n^2	$\sqrt{n}$
	FALKON [39]	$n\sqrt{n}$	$\sqrt{n}$
Quantum	QKLS/QKLR [7]	$\sqrt{n}$	$n\sqrt{n}$
	QSVM [3]	$n\sqrt{n}$	$n^2\sqrt{n}$

closed-form solution $w = X^{-1}b$ requires $O(n^3)$ time to be computed and attains essentially zero optimization error. Because the total error is dominated by the $1/\sqrt{n}$ term of the estimation error, one may wonder about the convenience of paying a cost of order $O(n^3)$ to achieve zero optimization error. A careful analysis shows that this is indeed not a convenient choice and it is possible to design algorithms that are less accurate but converge faster to estimators that, albeit not attaining zero optimization error, achieve an error that matches the bound—this is the approach taken by early stopping, divide and conquer, and random subsampling methods. For many quantum algorithms, such as some of the quantum linear regression and least-squares algorithms we discussed in the previous section (e.g., Refs. [3,5]), the time complexity depends inverse polynomially on the error and the matching procedure has important consequences. In the next section we discuss these implications and show that, in order to obtain an optimization error that scales at most as the best estimation error, one should expect to pay a computational price which is polynomial in n.

For quantum algorithms with polylogarithmic error dependency, such as Ref. [7], the optimization error is lower than the estimation error and therefore there are no bounds to be matched. In this case, we show that the quantum algorithms argument cannot achieve a polylogarithmic runtime in the dimension of the training set based on an argument that analyzes the error dependency introduced via the finite sampling process that is required to extract a classical output from the algorithm. This will be discussed in a later section.

We begin by discussing the dependency on the error and then proceed to discuss the dependency on the measurement errors. We summarize the results of our analysis in Table I.

Error dependency of the quantum algorithms. In this section we show that in order to have a total error [see Eq. (10)] that scales as $1/\sqrt{n}$ we must introduce a polynomial n dependency in the quantum algorithm. For simplicity, we present our argument by discussing the case of quantum least-squares algorithms with inverse polynomial dependency on the error [4,5,29]. Our results generalize easily for all kernel methods.

For a γ error guarantee on the final output state, the quantum algorithms we consider have a time complexity that scales as $O(\kappa^c\gamma^{-\beta}\text{polylog}(n))$ for some $\beta, c > 0$. For example, $\beta = 3$ in Ref. [5] and $\beta = 4$ in Ref. [40].

Since for the quantum algorithm the data matrix needs either to be Hermitian or encoded in a larger Hermitian matrix such that the dimensionality of the matrix is $n \times d$ for n data points in $\mathbb{R}^d$, we assume here for simplicity that the data are given by an $n \times n$ Hermitian matrix, i.e., n points in $\mathbb{R}^n$.

In order give a precise bound to the optimization error term in Eq. (10) in terms of the approximation error of the quantum algorithm, we consider the following decomposition between the ideal minimizer of the empirical risk $\hat{f}$ and the approximate minimizer $\hat{f}_\gamma$, the output of the learning algorithm

$$\begin{aligned}\mathcal{E}(\hat{f}_\gamma) &- \mathcal{E}(\hat{f})\\ &= \underbrace{\mathcal{E}(\hat{f}_\gamma) - \hat{\mathcal{E}}(\hat{f}_\gamma)}_{\text{Generalization error}} + \underbrace{\hat{\mathcal{E}}(\hat{f}_\gamma) - \hat{\mathcal{E}}(\hat{f})}_{\text{Algorithmic error}} + \underbrace{\hat{\mathcal{E}}(\hat{f}) - \mathcal{E}(\hat{f})}_{\text{Generalization error}}\end{aligned} \tag{12}$$

$$= \Theta(n^{-1/2}) + \underbrace{\hat{\mathcal{E}}(\hat{f}_\gamma) - \hat{\mathcal{E}}(\hat{f})}_{\text{Algorithmic error}}, \tag{13}$$

where the first and third contributions result from the generalization error bounds and the second is the approximation error of the quantum algorithm. In order to achieve the best statistical performance the algorithmic error must scale at worst as the worst statistical error, that is, $\hat{\mathcal{E}}(\hat{f}_\gamma) - \hat{\mathcal{E}}(\hat{f}) = O(n^{-1/2})$.

Let us analyze the algorithmic error term for the problem of linear regression and least-squares problem. Assuming that the output of the quantum algorithm is a state $|\tilde{w}\rangle$ while the exact minimizer of the empirical risk is $|w\rangle$, with $\||\tilde{w}\rangle - |w\rangle\| \leqslant \gamma$, we find that (assuming $|X|$ and $|Y|$ are bounded)

$$|\hat{\mathcal{E}}(\hat{f}_\epsilon) - \hat{\mathcal{E}}(\hat{f})| \leqslant \frac{1}{n}\sum_{i=1}^{n}|(\tilde{w}^T x_i - y_i)^2 - (w^T x_i - y_i)^2| \tag{14}$$

$$\leqslant \frac{1}{n}\sum_{i=1}^{n} L\big|(\tilde{w} - w)^T x_i\big| \tag{15}$$

$$\leqslant \frac{1}{n}\sum_{i=1}^{n} L\|\tilde{w} - w\|\,\|x_i\| \leqslant k\cdot\gamma, \tag{16}$$

where $k > 0$ is a constant and the inequality comes from Cauchy-Schwarz and the fact that, because $|X|$ and $|Y|$ are bounded, we have that, for the square loss ℓ_{sq}, the following inequality holds, $|\ell_{\text{sq}}(f(x_1), y_1) - \ell_{\text{sq}}(f(x_2), y_2)| \leqslant L|(f(x_1) - y_1) - (f(x_2) - y_2)|$ for some $L > 0$.

In order to have an algorithm that achieves the best possible statistical accuracy, we need the algorithmic error to scale at worst as the statistical error—this can be obtained by setting $\gamma = n^{-1/2}$. In this case, the time complexity of quantum least squares becomes

$$O(\kappa^c n^{\beta/2} \log(n)), \tag{17}$$

for some constant c.

Measurement errors in quantum algorithms. So far we have ignored the error introduced by the measurement process used to compute a classical estimate of the output of the quantum algorithm. In practice, this corresponds to the estimation of expected values of quantum operators. With a classical statistical analysis of the errors—and assuming the measurements are statistically independent—it is possible to show, using the central limit theorem, that the estimation error for a quantum expected value scales as $1/\sqrt{m}$, where m is the number of measurements [41]. This is known as the *standard quantum limit* or the *shot-noise limit*. Using techniques developed within the field of quantum metrology it is often possible to overcome this limit—using the same physical resources and the addition of quantum effects such as entanglement—and obtain a precision that scales as $1/m$. It is possible to show that this is the ultimate limit to measurement precision and follows directly from the Heisenberg uncertainty principle [41,42].

In this section we analyze the contribution of the measurement error to the time complexity of quantum learning algorithms. Let us consider again the case of quantum least squares. The (quantum) output of the algorithm is the state $|\tilde{w}\rangle$, an approximation (due to algorithmic errors) of the ideal output $|w\rangle$. Using techniques such as quantum state tomography we can produce a classical estimate $\hat{w}$ of the vector $\tilde{w}$ with accuracy,

$$\|\tilde{w} - \hat{w}\| \leqslant \tau = \Omega(1/m), \tag{18}$$

where m is the number of measurements performed for the estimation of the expected values on $|\tilde{w}\rangle$.

Let y be the ideal prediction. We have two sources of error, the algorithmic error and the error coming from the estimation process,

$$|y - \hat{y}| = |w^T x - \hat{w}^T x| \tag{19}$$

$$\leqslant \|w - \tilde{w} + \tau\| \, \|x\| \tag{20}$$

$$\leqslant (\gamma + \tau)\|x\|, \tag{21}$$

where we used Cauchy-Schwarz and $\|w - \tilde{w}\| \leqslant \gamma$.

By virtue of Eq. (12), we have that, if we want an algorithm that attains the best statistical accuracy for the number of samples contained in the training set, we need to make sure that the contribution coming from the measurement error scales at most as the worst possible generalization error. Recalling that the generalization error scales as $\Theta(1/\sqrt{n})$ we have that $\tau = O(1/\sqrt{n})$, from which it follows that $m = \Omega(\sqrt{n})$. This lower bound on the number of measurement required to extract a classical estimate of the output effectively sets a $\Omega(\sqrt{n})$ lower bound on the time complexity of all supervised quantum machine learning algorithms.

If we consider this lower bound, classical algorithms can have time complexities matching those of the quantum algorithms. For a more detailed comparison of the runtime of popular classical and quantum algorithms for supervised learning problems, see Table I.

Conclusions. Quantum machine algorithms promise to be exponentially faster than classical methods. In this paper, we use standard results from statistical learning theory to rule out quantum machine algorithms with a polylogarithmic time complexity in the input dimensions. Considering that almost any current and practically used machine learning algorithm has a polynomial runtime, our results warn against the possibility of superpolynomial advantages for supervised quantum machine learning. We remark two limitations of our analysis. First, our results do not rule out exponential advantages over classical algorithms with a superpolynomial runtime. Second, we do not make assumptions on the hypothesis space; using prior knowledge it is possible get error rates that converge faster than $1/\sqrt{n}$.

Our argument leverages the fact that the statistical error of the algorithm has a provable polynomial dependence on the number of samples in the training set. Since the statistical error and the approximation error of the algorithm are additive, in order to achieve the best possible error rate, the asymptotic scaling of the statistical error must match that of the approximation error. This matching forces the approximation error of quantum algorithms to scale polynomially with the number of samples. This effectively kills quantum speedups for algorithms that have polynomial dependence on the error.

For algorithms where the dependency on the error is logarithmic, this argument does not apply. In this case, we show that the sampling error coming from the measurement process also adds up additively to the total error and this introduces a polynomial dependency in the number of samples that kills the superpolynomial speedup.

Notably, our results hold even assuming that quantum algorithms can access a quantum data structure at no cost. In this respect, we prove a stronger "no-go" result for quantum learning than the one proved by Tang in Ref. [9]. Indeed, the latter relies on a classical data structure that mimics a quantum data structure but is unrealistic in practice.

As future directions, it is worth mentioning that it may be possible strengthen our results by analyzing the n dependency of the condition number. Previous results in this direction are discussed in Refs. [19,43].

Acknowledgments. The authors would like to thank Aram Harrow, Sathya Subramanian, and Maria Schuld for helpful feedback on an earlier draft of the article, Shantanav Chakraborty and Stacey Jeffrey for discussions on the Frobenius norm dependency of their quantum least-squares algorithm, and Alessandro Davide Ialongo for helpful comments. This research was supported in part by the National Science Foundation under Grant No. NSF PHY-1748958 and by the Heising-Simons Foundation. A.R. is supported by the Simons Foundation through *It from Qubit: Simons Collaboration on Quantum Fields, Gravity, and Information*. L.W. is supported by a Google Ph.D. Fellowship.

[1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature (London) **549**, 195 (2017).

[2] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Proc. R. Soc. A **474**, 20170551 (2018).

[3] P. Rebentrost, M. Mohseni, and S. Lloyd, Phys. Rev. Lett. **113**, 130503 (2014).

[4] N. Wiebe, D. Braun, and S. Lloyd, Phys. Rev. Lett. **109**, 050505 (2012).

[5] M. Schuld, I. Sinayskiy, and F. Petruccione, Phys. Rev. A **94**, 022342 (2016).

[6] I. Kerenidis and A. Prakash, Phys. Rev. A **101**, 022316 (2020).

[7] S. Chakraborty, A. Gilyén, and S. Jeffery, in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, edited by C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, Leibniz International Proceedings in Informatics (LIPIcs) (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019), pp. 33:1–33:14.

[8] S. Aaronson, Nat. Phys. **11**, 291 (2015).

[9] E. Tang, in *STOC 2019: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2019), pp. 217–228.

[10] N.-H. Chia, H.-H. Lin, and C. Wang, arXiv:1811.04852.

[11] N.-H. Chia, T. Li, H.-H. Lin, and C. Wang, arXiv:1901.03254.

[12] A. Gilyén, S. Lloyd, and E. Tang, arXiv:1811.04909.

[13] N.-H. Chia, A. Gilyén, T. Li, H.-H. Lin, E. Tang, and C. Wang, in *STOC 2020: Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2020), pp. 387–400.

[14] A. B. Grilo, I. Kerenidis, and T. Zijlstra, Phys. Rev. A **99**, 032314 (2019).

[15] V. Kanade, A. Rocchetto, and S. Severini, Quantum Inf. Comput. **19**, 1261 (2019).

[16] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms* (Cambridge University Press, Cambridge, U.K., 2014).

[17] V. N. Vapnik and V. Vapnik, *Statistical Learning Theory*, Vol. 1 (Wiley, New York, 1998).

[18] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, J. Assoc. Comput. Mach. **36**, 929 (1989).

[19] F. Cucker and S. Smale, Bull. Am. Math. Soc. **39**, 1 (2002).

[20] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, Berlin, 2006).

[21] F. Bauer, S. Pereverzev, and L. Rosasco, J. Complexity **23**, 52 (2007).

[22] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (MIT Press, Cambridge, MA, 2006).

[23] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis* (Cambridge University Press, Cambridge, U.K., 2004).

[24] Y. Zhang, J. Duchi, and M. Wainwright, in *Conference on Learning Theory*, Vol. 30 (PMLR, 2013), pp. 592–617.

[25] A. Rahimi and B. Recht, in *Advances in Neural Information Processing Systems (NIPS)*, Vol. 3 (Curran Associates, Red Hook, NY, 2007), p. 5.

[26] A. J. Smola and B. Schölkopf, *Sparse Greedy Matrix Approximation for Machine Learning* (Morgan Kaufmann, Burlington, MA, 2000), pp. 911–918.

[27] C. K. Williams and M. Seeger, in *Advances in Neural Information Processing Systems (NIPS)* (MIT Press, Cambridge, Massachusetts, 2001), pp. 682–688.

[28] A. Rudi, R. Camoriano, and L. Rosasco, in *Advances in Neural Information Processing Systems (NIPS)* (Curran Associates, Red Hook, NY, 2015), pp. 1657–1665.

[29] G. Wang, Phys. Rev. A **96**, 012335 (2017).

[30] D.-B. Zhang, S.-L. Zhu, and Z. Wang, arXiv:1808.09607.

[31] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[32] S. Lloyd, M. Mohseni, and P. Rebentrost, Nat. Phys. **10**, 631 (2014).

[33] S. Kimmel, C. Y.-Y. Lin, G. H. Low, M. Ozols, and T. J. Yoder, npj Quantum Inf. **3**, 13 (2017).

[34] Y. Yang, M. Pilanci, and M. J. Wainwright, Ann. Stat. **45**, 991 (2017).

[35] S. Ma and M. Belkin, in *Advances in Neural Information Processing Systems (NIPS)* (Curran Associates, Red Hook, NY, 2017), pp. 3778–3787.

[36] A. Gonen, F. Orabona, and S. Shalev-Shwartz, in *International Conference on Machine Learning* (Curran Associates, Red Hook, NY, 2016), pp. 1397–1405.

[37] H. Avron, K. L. Clarkson, and D. P. Woodruff, SIAM J. Matrix Anal. Appl. **38**, 1116 (2017).

[38] G. E. Fasshauer and M. J. McCourt, SIAM J. Sci. Comput. **34**, A737 (2012).

[39] A. Rudi, L. Carratino, and L. Rosasco, in *Advances in Neural Information Processing Systems (NIPS)* (Curran Associates, Red Hook, NY, 2017), pp. 3888–3898.

[40] T. Li, S. Chakrabarti, and X. Wu, in *Proceedings of the 36th International Conference on Machine Learning*, edited by K. Chaudhuri and R. Salakhutdinov (PMLR, Long Beach, CA, 2019), Vol. 97, pp. 3815–3824.

[41] V. Giovannetti, S. Lloyd, and L. Maccone, Science **306**, 1330 (2004).

[42] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **96**, 010401 (2006).

[43] H. Hochstadt, *Integral Equations* (Wiley, Hoboken, NJ, 2011), Vol. 91.