Quantum algorithm for linear regression

Guoming Wang

Phys. Rev. A **96**, 012335 — Published 31 July 2017

DOI: 10.1103/PhysRevA.96.012335

New Quantum Algorithm for Linear Regression

Guoming Wang*
Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, MD 20742, USA
(Dated: July 10, 2017)

We present a quantum algorithm for fitting a linear regression model to a given data set using the least squares approach. Different from previous algorithms which yield a quantum state encoding the optimal parameters, our algorithm outputs these numbers in the classical form. So by running it once, one completely determines the fitted model and then can use it to make predictions on new data at little cost. Moreover, our algorithm works in the standard oracle model, and can handle data sets with nonsparse design matrices. It runs in time $\mathrm{poly}(\log(N), d, \kappa, 1/\epsilon)$, where N is the size of the data set, d is the number of adjustable parameters, κ is the condition number of the design matrix, and ϵ is the desired precision in the output. We also show that the polynomial dependence on d and κ is necessary. Thus, our algorithm cannot be significantly improved. Furthermore, we also give a quantum algorithm that estimates the quality of the least-squares fit (without computing its parameters explicitly). This algorithm runs faster than the one for finding this fit, and can be used to check whether the given data set qualifies for linear regression in the first place.

I. INTRODUCTION

Curve fitting, also known as *regression analysis* in statistics, is the process of constructing a mathematical function that has the best fit to a series of data points according to some criterion. This procedure is widely used in many scientific fields, including physics, astronomy, chemistry, biology, medicine, agriculture, geology, engineering, economics, etc. It can help us to understand the relationship among variables, to predict the unknown value of a variable from the known values of other variables, to compress data, and to aid data visualization. In practice, one often needs to fit a concise theoretical model to a huge amount of experimental data, and it is highly desirable to have an efficient algorithm for this task.

Linear regression is one of the most common forms of curve fitting. It assumes that the relationship between a *dependent variable* (or *response*) and one or more *explanatory variables* (or *predictors*) is linear. So it fits a function which is linear in some adjustable parameters to the given data set. These parameters are usually determined using the *(ordinary) least squares* approach, which minimizes the sum of the squared deviations of the data from the model function. This optimization problem turns out to be closely related to a matrix inversion problem, which is time-consuming for large data sets.

With the rise of quantum computation, one naturally asks whether quantum algorithms can perform linear regression faster than their classical counterparts. Wiebe, Braun and Llyod (WBL) [1] first studied this problem and answered it affirmatively. Building upon the quantum algorithm for solving linear systems of equations by Harrow, Hassidim and Lloyd (HHL) [2], they developed a quantum algorithm for estimating the quality of the least-squares fit for a given data set. Under the assumption that there exist two fast procedures for specifying the nonzero entries of the *design matrix* and for preparing a quantum state proportional to the *response vector*, respectively (see Section II B for the definition of this matrix and vector), their algorithm has complexity $\mathrm{poly}(\log(N), s, \kappa, 1/\epsilon)$, where N is the size of the data set, s and κ are the sparsity and condition number of the design matrix, respectively, and ϵ is the desired precision in the output. WBL also gave an algorithm with similar complexity for preparing a quantum state approximately proportional to the optimal parameters. Furthermore, they proposed to use statistical sampling and quantum state tomography to find a concise representation for this state. WBL's algorithms are mainly suited for data sets whose design matrices are sparse and well-conditioned.

Recently, Schuld, Sinayskiy and Petruccione (SSP) [3] reapproached the problem of linear regression on a quantum computer from a machine learning perspective. Building upon HHL's strategy for matrix inversion and Lloyd, Mohseni and Rebentrost (LMR)'s *density matrix exponentiation* technique [4], they developed a quantum algorithm for *pattern recognition*, in which one only needs to make a prediction on a new data point based on a linear regression model trained on a given data set and

* wgmcreate@berkeley.edu

does not need to find this model explicitly. Their algorithm takes as input multiple copies of three quantum states encoding the design matrix of the training set, the response vector of the training set, and the new data point, respectively, and outputs a scalar value which is the predicted response for the new data point. Excluding the costs of preparing these states and assuming the design matrix is close to a low-rank matrix, this algorithm has complexity $\mathrm{poly}(\log(d), \kappa, 1/\epsilon)$, where d is the number of adjustable parameters, κ is the condition number of the design matrix, and ϵ is the desired precision in the output. SSP's algorithm is mainly suited for data sets whose design matrices are well-conditioned and have low-rank approximations.

Both WBL and SSP have focused on the scenario where both the size N of the data set and the number d of adjustable parameters are exponentially large. Thus, they do not attempt to find the optimal parameters explicitly (which is time-consuming), but only encode these parameters in a quantum state (which can be used to make predictions on new data via swap test). While this scenario is useful in some applications (e.g. estimation of the output state of a quantum device), we believe that it is equally important to consider the scenario where d is much smaller than N. Namely, N is exponentially large, but d is only polynomially large. One often encounters this situation when dealing with a classical data set and wanting to compress a large amount of data into a concise model (with few parameters). Once such a model is found explicitly, one can use it to make predictions on new data at little cost. Furthermore, saving the optimal parameters is much easier than storing the quantum state encoding these numbers, as quantum resources are fragile.

For the above reasons, in this paper, we present a new quantum algorithm for fitting a linear regression model to a given data set using the least squares approach. Our algorithm works in the standard oracle model, and outputs the optimal parameters in the classical form. It runs in time $\mathrm{poly}(\log(N), d, \kappa, 1/\epsilon)$, where N is the size of the data set, d is the number of adjustable parameters, κ is the condition number of the design matrix, and ϵ is the desired precision in the output. Note that the polynomial dependence on d is inevitable, because simply writing down all the optimal parameters takes $\Omega(d)$ time. We show that the polynomial dependence on κ is also necessary, by proving a lower bound on the quantum query complexity of this problem. These facts imply that our algorithm cannot be significantly improved. Furthermore, we also give a quantum algorithm that estimates the quality of the least-squares fit (without computing its parameters explicitly). This algorithm runs faster than the one for finding this fit, and can be used to check whether the given data set qualifies for linear regression in the first place.

We make use of two recent results in designing our algorithms. The first one is Low and Chuang's method for Hamiltonian simulation based on *qubitization* [5] and *quantum signal processing* [6]. This method allows us to simulate a nonsparse Hamiltonian, provided that this Hamiltonian can be embedded into a larger unitary operator in certain way. The second one is Childs, Kothari and Somma (CKS)'s approach to matrix inversion [7]. This approach differs from HHL's in that it does not use phase estimation, but relies on a techique for implementing a *linear combination of unitaries* (LCU) and a suitable Fourier or Chebyshev series representation of the matrix inverse function. Consequently, it has exponentially better dependence on the precision than HHL's approach. We combine these results with traditional techniques (such as amplitude estimation [8]) to find the optimal parameters and to estimate the quality of the least-squares fit for a given data set.

As mentioned before, WBL have suggested a sampling-based algorithm for learning the optimal parameters in Ref. [1]. Our algorithm for computing the optimal parameters differs from their algorithm in several ways. First, as mentioned above, our algorithm uses the approach of Ref. [7] for matrix inversion, which has better dependence on the desired precision in the output than HHL's approach (which was used by Ref. [1]). Second, we compute the pseudoinverse of the design matrix by considering its singular value decomposition (SVD), while Ref. [1] achieved this by following a step-by-step approach (see the end of Section IV for more discussion on this). Third, as mentioned above, our algorithm is based on the method of Ref. [5] for simulating a large class of Hamiltonians, while Ref. [1] was based on an old method for simulating sparse Hamiltonians. As a consequence, our algorithm can handle data sets with nonsparse design matrices. Fourth, we assume that the data set is given via standard oracles (see Section II C for more details), and explicitly address the issue of preparing a quantum state proportional to the response vector. By contrast, Ref. [1] ignored the cost of this step. Finally, our algorithm uses amplitude estimation to estimate the optimal parameters, which has quadratically better dependence on the desired accuracy in the output than statistical sampling (which was used by Ref. [1]).

The remainder of this paper is organized as follows. In Section II, we provide some requisite back-

ground information, and formally state the problems studied in this work. In Section III, we describe an efficient procedure for simulating a nonsparse Hamiltonian related to the design matrix, which is a key component of our algorithms. In Section IV, we present a quantum algorithm for fitting a linear regression model to a given data set using the least squares approach. In Section V, we propose a quantum algorithm that estimates the quality of the least-squares fit (without computing its parameters explicitly). In Section VI, we prove a lower bound on the quantum query complexity of linear regression. Finally, we conclude in Section VII with some comments and future research directions.

II. PRELIMINARIES

In this section, we provide the necessary background information to understand this paper. In Section II A, we introduce the notation used in this paper. In Section II B, we review some basic facts about linear regression. In Section II C, we formally state the problems studied in this work.

A. Notation

Given a real number x, we define its sign as $\mathrm{sgn}(x) = 1$ if $x \geq 0$, and $\mathrm{sgn}(x) = -1$ otherwise. Given two real numbers a, b and a real number $\delta > 0$, we say that a is a δ-additive approximation of b if $|a - b| \leq \delta$. Moreover, we say that an algorithm estimates a quantity x up to additive error δ if it outputs a δ-additive approximation of x.

Given a vector $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T \in \mathbb{C}^N$, we use $\|\mathbf{x}\|_\infty$ and $\|\mathbf{x}\|$ to denote the l^∞ and l^2 norms of $\mathbf{x}$, respectively, i.e.

$$\|\mathbf{x}\|_\infty := \max_{1 \leq i \leq N} |x_i|, \tag{1}$$

and

$$\|\mathbf{x}\| := \sqrt{\sum_{i=1}^{N} |x_i|^2}. \tag{2}$$

Moreover, we define

$$\rho(\mathbf{x}) := \frac{\sqrt{N}\|\mathbf{x}\|_\infty}{\|\mathbf{x}\|} = \frac{\max_{1 \leq i \leq N} |x_i|}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} |x_i|^2}}. \tag{3}$$

The smaller $\rho(\mathbf{x})$ is, the more *balanced* $\mathbf{x}$ is, in the sense that the no entry of $\mathbf{x}$ has significantly larger norm than the quadratic mean norm of $\mathbf{x}$'s entries. In particular, we say that $\mathbf{x}$ is *balanced* if $\rho(\mathbf{x}) = O(1)$ (e.g. at most 100).

Given a matrix $\mathbf{A} = (a_{i,j}) \in \mathbb{C}^{N \times M}$, we define $\mathbf{a}_i := (a_{i,1}, a_{i,2}, \ldots, a_{i,M})^T$, for each $i \in \{1, 2, \ldots, N\}$. We also use $\|\mathbf{A}\|$ and $\|\mathbf{A}\|_{\mathrm{F}}$ to denote the spectral and Frobenius norms of $\mathbf{A}$, respectively, i.e.

$$\|\mathbf{A}\| := \max_{\mathbf{x} \in \mathbb{C}^M,\ \mathbf{x} \neq 0} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}, \tag{4}$$

and

$$\|\mathbf{A}\|_{\mathrm{F}} := \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{M} |a_{i,j}|^2}. \tag{5}$$

In addition, we define

$$\|\mathbf{A}\|_{2,\infty} := \max_{\mathbf{x} \in \mathbb{C}^M,\ \mathbf{x} \neq 0} \frac{\|\mathbf{A}\mathbf{x}\|_\infty}{\|\mathbf{x}\|} \tag{6}$$

$$= \max_{1 \leq i \leq N} \|\mathbf{a}_i\| \tag{7}$$

$$= \max_{1 \leq i \leq N} \sqrt{\sum_{j=1}^{M} |a_{i,j}|^2} \tag{8}$$

and

$$\sigma(\mathbf{A}) := \frac{\sqrt{N}\|\mathbf{A}\|_{2,\infty}}{\|\mathbf{A}\|_{\mathrm{F}}} = \frac{\max_{1 \leq i \leq N} \|\mathbf{a}_i\|}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{a}_i\|^2}}. \tag{9}$$

The smaller $\sigma(\mathbf{A})$ is, the more *balanced* $\mathbf{A}$ is, in the sense that no row of $\mathbf{A}$ has significantly larger norm than the quadratic mean norm of $\mathbf{A}$'s rows. In particular, we say that $\mathbf{A}$ is *balanced* if $\sigma(\mathbf{A}) = O(1)$ (e.g. at most 100).

For the above $\mathbf{A}$, we also use $\mathrm{Range}(\mathbf{A})$ to denote the range (i.e. column space) of $\mathbf{A}$, and use $\Pi(\mathbf{A})$ to denote the projection onto $\mathrm{Range}(\mathbf{A})$. We also use $s_j(A)$ to denote j-th smallest singular value of A (counted with multiplicity), and use $\lambda_j(A)$ to denote the j-th smallest eigenvalue of A (counted with multiplicity), starting with $j = 1$. The *condition number* of $\mathbf{A}$, denoted by $\kappa(\mathbf{A})$, is defined as the ratio of largest to smallest singular value of $\mathbf{A}$. Futhermore, we use $\mathbf{A}^+$ to denote the *Moore-Penrose pseudoinverse* of $\mathbf{A}$. That is, if $\mathbf{A}$ has the singular value decomposition $\mathbf{A} = \sum_k s_k \mathbf{u}_k \mathbf{v}_k^\dagger$, where $s_k > 0$, $\mathbf{u}_k \in \mathbb{C}^N$ and $\mathbf{v}_k \in \mathbb{C}^M$ are unit vectors, then $\mathbf{A}^+ := \sum_k s_k^{-1} \mathbf{v}_k \mathbf{u}_k^\dagger$.

Given a matrix $\mathbf{A} \in \mathbb{C}^{N \times M}$ and a vector $\mathbf{x} \in \mathbb{C}^N$, we define

$$\tau(\mathbf{A}, \mathbf{x}) := \frac{\|\Pi(\mathbf{A})\mathbf{x}\|^2}{\|\mathbf{x}\|^2}. \tag{10}$$

In words, $\tau(\mathbf{A}, \mathbf{x})$ measures how much "fraction" of $\mathbf{x}$ lies in the range of $\mathbf{A}$. In particular, we say that $(\mathbf{A}, \mathbf{x})$ is *well-behaved* if $\tau(\mathbf{A}, \mathbf{x}) = \Omega(1)$ (e.g. at least $2/3$).

Given a vector $\mathbf{x} \in \mathbb{C}^N$, we say that $\mathbf{x}$ is d-sparse if it contains at most d nonzero entries. Given a matrix $\mathbf{A} \in \mathbb{C}^{N \times M}$, we say that $\mathbf{A}$ is d-sparse if it contains at most d nonzero entries in each row and column. In particular, if $d = \text{poly}(\log(L))$ where $L = \max\{N, M\}$, then we simply say that $\mathbf{A}$ is sparse.

Given a state $|\varphi\rangle \in \mathbb{C}^d$ and a real number $\epsilon > 0$, we say that a procedure prepares $|\varphi\rangle$ with precision ϵ if this procedure prepares a state $|\psi\rangle \in \mathbb{C}^d$ satisfying $\||\varphi\rangle - |\psi\rangle\| \leq \epsilon$.

Given a unitary operation $V \in \mathbb{U}(d)$ and a real number $\epsilon > 0$, we say that a procedure implements V with precision ϵ and failure probability $O(\epsilon)$ if there exists an integer $l \geq 0$ such that, on any input state $|\psi\rangle \in \mathbb{C}(d)$, this procedure first appends an l-qubit ancilla system in state $|0^l\rangle$, then performs a unitary operation $U \in \mathbb{U}(2^l \times d)$ on the joint system such that

$$U|0^l\rangle|\psi\rangle = |0^l\rangle A|\psi\rangle + \sum_{j \neq 0^l} |j\rangle B_j|\psi\rangle, \tag{11}$$

where A and the B_j's are linear operators satisfying $\|A - V\| \leq \epsilon$ and $A^\dagger A + \sum_{j \neq 0^l} B_j^\dagger B_j = I$, and finally measures the ancilla system and postselects on the outcome being 0^l. Note that since V is unitary and $\|V - A\| \leq \epsilon$, we get $\|V|\psi\rangle - A|\psi\rangle\| \leq \epsilon$, and

$$\begin{aligned} |\||A|\psi\rangle\| - 1| &= |\||A|\psi\rangle\| - \||V|\psi\rangle\|| && (12)\\ &\leq \|(A - V)|\psi\rangle\| && (13)\\ &\leq \epsilon, && (14) \end{aligned}$$

and

$$\begin{aligned} \left\| V|\psi\rangle - \frac{A|\psi\rangle}{\||A|\psi\rangle\|} \right\| &\leq \|V|\psi\rangle - A|\psi\rangle\| \\ &\quad + \left\| A|\psi\rangle - \frac{A|\psi\rangle}{\||A|\psi\rangle\|} \right\| && (15)\\ &\leq \epsilon + |\||A|\psi\rangle\| - 1| && (16)\\ &\leq 2\epsilon. && (17) \end{aligned}$$

Thus, on any input state $|\psi\rangle$, this procedure succeeds with probability $\|A|\psi\rangle\|^2 = 1 - O(\epsilon)$ (with a flag indicating success), and when it succeeds, it outputs the state $\frac{A|\psi\rangle}{\||A|\psi\rangle\|}$ which is $O(\epsilon)$-close to $V|\psi\rangle$ in l^2 norm.

Now consider a quantum circuit consisting of a sequence of unitary operations $V_1 \to V_2 \to \cdots \to V_{m-1} \to V_m$. Suppose P_i is a procedure that implements V_i with precision ϵ_i and failure probability $O(\epsilon_i)$, for each $i \in \{1, 2, \ldots, m\}$. Let P be the concatenation of these procedures (i.e. $P_1 \to P_2 \to \cdots \to P_{m-1} \to P_m$). Then by a standard hybrid argument, one can show that P implements the unitary operation $V := V_m V_{m-1} \ldots V_2 V_1$ with precision $\epsilon := \sum_{i=1}^m \epsilon_i$ and failure probability $O(\epsilon)$. Thus, on any input state $|\psi\rangle$, the procedure P succeeds with probability $1 - O(\epsilon)$ (with a flag indicating success), and when it succeeds, it outputs a state $O(\epsilon)$-close to $V|\psi\rangle$ in l^2 norm. This fact will be useful in the design of our algorithms.

B. Linear Regression

Given a data set $\{y_i, x_{i,1}, x_{i,2}, \ldots, x_{i,d}\}_{i=1}^N$ of N statistical units (where $N \geq d$), a linear regression model assumes that the relationship between the *response* (or *regressand*, *dependent variable*) y_i and the *predictors* (or *regressors*, *explanatory variables*) $x_{i,1}, x_{i,2}, \ldots, x_{i,d}$ is linear. That is, there exist some unknown parameters β_1, β_2, $\ldots$, β_d and residual terms ϵ_i such that

$$y_i = \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_d x_{i,d} + \epsilon_i, \ 1 \leq i \leq N. \tag{18}$$

In the matrix form, it can be written as

$$\mathbf{y} = \mathbf{X}\beta + \epsilon, \tag{19}$$

where

$$\mathbf{X} := \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix} = \begin{pmatrix} x_{1,1} & x_{1,2} & \ldots & x_{1,d} \\ x_{2,1} & x_{2,2} & \ldots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \ldots & x_{N,d} \end{pmatrix}, \tag{20}$$

$$\mathbf{y} := \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}, \quad \beta := \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_d \end{pmatrix}, \quad \epsilon := \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{pmatrix}. \tag{21}$$

We usually call $\mathbf{X}$ the *design matrix*, $\mathbf{y}$ the *response vector*, β the *parameter vector*, and ϵ the *residual vector*. Here we assume that the $x_{i,j}$'s and y_i's are real numbers. This is actually without loss of generality, because any linear regression model with complex variables can be reduced to a (slightly larger) linear regression model with real variables. Moreover, we assume that the design matrix $\mathbf{X}$ has full rank d. In other words, the d columns of $\mathbf{X}$ are linearly independent. This is a necessary condition for linear regression to have a unique solution.

We emphasize that the predictors can be nonlinear functions of some "baseline" variables. This allows linear regression to fit a nonlinear relationship between the response and the baseline variables. For example, suppose we are interested in learning how the yield y_i of a chemical synthesis is related to the temperature t_i at which the synthesis takes place. We propose a quadratic model of the form:

$$y_i = a_0 + a_1 t_i + a_2 t_i^2 + \epsilon_i, \ 1 \le i \le N. \tag{22}$$

This model is linear in the parameters a_0, a_1 and a_2, but nonlinear in the baseline variable t_i. In the matrix form, it can be written as

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ \vdots & \vdots & \vdots \\ 1 & t_N & t_N^2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{pmatrix} \tag{23}$$

Here the design matrix is a Vandermonde matrix, and it has full rank as long as there are at least three distinct t_i's. Furthermore, this design matrix is not sparse. This is a generic phenomenon in linear regression, because we often include the constant 1 as one of the predictors, and consequently the design matrix often contains a dense column of all 1's.

Linear regression models are usually fitted using the *least squares* approach, which minimizes the sum of the squared residuals. Namely, it finds

$$\hat{\beta} := \operatorname*{argmin}_{\beta \in \mathbb{R}^d} \|\mathbf{X}\beta - \mathbf{y}\|^2 \tag{24}$$

This optimization problem has the following closed-form solution [9]

$$\hat{\beta} = \mathbf{X}^+ \mathbf{y} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \tag{25}$$

Noting that

$$\Pi(\mathbf{X}) = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T, \tag{26}$$

we obtain

$$\mathbf{X}\hat{\beta} = \Pi(\mathbf{X})\mathbf{y}. \tag{27}$$

Namely, $\mathbf{X}\hat{\beta}$ is exactly the projection of $\mathbf{y}$ onto the range of $\mathbf{X}$. This is the geometric interpretation of least-squares linear regression.

Although Eq. (25) gives the solution of linear regression, it is not computationally convenient, because $\mathbf{X}$ is a rectangular matrix and $\mathbf{X}^+$ is not easy to implement physically. To overcome this issue, we adopt the strategy of Ref. [2] (which was also used in Ref. [1]) and embed $\hat{\beta}$ into the solution of a larger linear system. Specifically, let

$$\mathbf{A} := \begin{pmatrix} 0 & \mathbf{X} \\ \mathbf{X}^T & 0 \end{pmatrix}, \ \mathbf{b} := \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix}, \ \mathbf{z} := \begin{pmatrix} 0 \\ \hat{\beta} \end{pmatrix}. \tag{28}$$

Then we have

$$\begin{aligned} \mathbf{A}^+ \mathbf{b} &= \begin{pmatrix} 0 & \mathbf{X} \\ \mathbf{X}^T & 0 \end{pmatrix}^+ \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} && (29) \\ &= \begin{pmatrix} 0 & (\mathbf{X}^T)^+ \\ \mathbf{X}^+ & 0 \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} && (30) \\ &= \begin{pmatrix} 0 \\ \mathbf{X}^+ \mathbf{y} \end{pmatrix} && (31) \\ &= \begin{pmatrix} 0 \\ \hat{\beta} \end{pmatrix} && (32) \\ &= \mathbf{z}. && (33) \end{aligned}$$

The fact that $\mathbf{A}$ is a real symmetric matrix facilitates the implementation of $\mathbf{A}^+$. Once we have a procedure for preparing a quantum state proportional to $\mathbf{A}^+\mathbf{b} = \mathbf{z}$, we can utilize this procedure to get useful information about $\hat{\beta}$.

A statistical model fits a data set well only if the discrepancy between the observed response and the response predicted by this model is small. Here we measure the quality of the least-squares fit $\mathbf{y} \approx \mathbf{X}\hat{\beta}$ using the quantity

$$\tau := \frac{\|\hat{\mathbf{y}}\|^2}{\|\mathbf{y}\|^2} = 1 - \frac{\|\hat{\epsilon}\|^2}{\|\mathbf{y}\|^2}, \tag{34}$$

where

$$\hat{\mathbf{y}} := \mathbf{X}\hat{\beta} = \Pi(\mathbf{X})\mathbf{y} \tag{35}$$

and

$$\hat{\epsilon} := \mathbf{y} - \hat{\mathbf{y}} = (I - \Pi(\mathbf{X}))\mathbf{y}. \tag{36}$$

Namely, $1 - \tau$ is the ratio of the squared norm of the residual vector $\hat{\epsilon}$ to that of the response vector $\mathbf{y}$. It turns out that $\tau = \tau(\mathbf{X}, \mathbf{y})$. So a data set $(\mathbf{X}, \mathbf{y})$ can be explained well by a linear regression model only if it is well-behaved, i.e. $\tau(\mathbf{X}, \mathbf{y}) = \Omega(1)$ (i.e. at least 2/3). This kind of data sets will be the main focus of our study. We will also give an efficient quantum algorithm for testing whether a given data set is well-behaved or not.

It is worth noting that Wiebe, Braun and Lloyd (WBL) [1] have used the quantity $E = \||\mathbf{y}\rangle - \mathbf{F}|\boldsymbol{\lambda}\rangle\|^2$ (according to their notation) to measure the error of the least-squares fit. Their $|\mathbf{y}\rangle$, $\mathbf{F}$ and $|\boldsymbol{\lambda}\rangle$ correspond to our $\mathbf{y}/\|\mathbf{y}\|$, $\mathbf{X}$ and $\hat{\beta}/\|\mathbf{y}\|$, respectively. Then one can see that their $1 - E$ is equivalent to our τ. So WBL essentially measured the quality of the least-squares fit in the same way as we do.

In practice, after one collects the raw data from the experiements, one does not immediately fit a

mathematical function to these data. Instead, one needs to preprocess the raw data to make them well-suited for data fitting. This preprocessing usually consists of imputation of missing data, data normalization or standadization, and elimination of influential *outliers* which have detrimental effect on the estimated regression function (an outlier is a data point whose response y does not follow the general trend of the rest of the data). The last step is important, because we want the fitted model to capture the *typical* relationship among the response and predictors so that it can be generalized to new data. This requires that the loss function

$$\|\mathbf{X}\beta - \mathbf{y}\|^2 = \sum_{i=1}^{N} \left|\mathbf{x}_i^T \beta - y_i\right|^2 \tag{37}$$

$$= \sum_{i=1}^{N} \left| \left(\sum_{j=1}^{d} \beta_j x_{i,j} \right) - y_i \right|^2, \tag{38}$$

should not be dominated by only a few data points. An outlier has the potential to do so, especially if it has high *leverage* (i.e. it has "extreme" predictor x values). However, we emphasize that not all outliers are influential and should be eliminated. The identification of influential outliers is an important and complicated topic, and many techniques have been developed for this task, such as difference in fits (DFFITS) and Cook's distance. In this paper, we assume that the given data set has already been preprocessed, and the harmful outliers have been removed. Since there is no general characterization of such data points, we assume for simplicity that no $\mathbf{x}_i$ or y_i has extremely large norm (compared to the average norm of the $\mathbf{x}_i$'s or y_i's, respectively). In other words, both $\mathbf{X}$ and $\mathbf{y}$ are balanced, i.e. $\sigma(\mathbf{X}) = O(1)$ (e.g. at most 100) and $\rho(\mathbf{y}) = O(1)$ (e.g. at most 100). These assumptions ensure that no data point has significantly larger contribution to the loss function than the others, and are useful in practice. However, we acknowledge that these assumptions might be too stringent for some applications, and standard preprocessing techniques do not always guarantee them, and it remains future work to extend our results to the most general case of linear regression.

C. Problem Statement

In this paper, we assume that the data set $\{y_i, x_{i,1}, x_{i,2}, \ldots, x_{i,d}\}_{i=1}^N$ is given via two black-box subroutines. For $\mathbf{X} = (x_{i,j}) \in \mathbb{R}^{N\times d}$, we assume there exists a procedure $\mathcal{P}_x$ that allows us to perform the map

$$|i\rangle|j\rangle|z\rangle \mapsto |i\rangle|j\rangle|z \oplus x_{i,j}\rangle \tag{39}$$

for any $i \in \{1, 2, \ldots, N\}$ and $j \in \{1, 2, \ldots, d\}$, where the third register holds a bit string representing an entry of $\mathbf{X}$. For $\mathbf{y} = (y_1, y_2, \ldots, y_N)^T \in \mathbb{R}^N$, we assume there exists a procedure $\mathcal{P}_y$ that allows us to perform the map

$$|i\rangle|z\rangle \mapsto |i\rangle|z \oplus y_i\rangle \tag{40}$$

for any $i \in \{1, 2, \ldots, N\}$, where the second register holds a bit string representing an entry of $\mathbf{y}$. We assume that both $\mathcal{P}_x$ and $\mathcal{P}_y$ are efficient, in the sense that they run in time $\mathrm{poly}(\log(N))$. This requires that either each entry of $\mathbf{X}$ and $\mathbf{y}$ can be quickly computed by an algorithm (given its position), or these entries are stored in a quantum random access memory (QRAM) beforehand. Our algorithms work well in both cases.

Given access to $\mathcal{P}_x$ and $\mathcal{P}_y$, our primary goal is to fit a linear regression model to the data set $\{y_i, x_{i,1}, x_{i,2}, \ldots, x_{i,d}\}_{i=1}^N$ using the least squares approach. Our secondary goal is to estimate the quality of the fitted model (without computing its parameters explicitly).

Formally, we define our linear regression (LR) problems as follows:

Problem 1 (LR-P). *Let $\mathbf{X} = (x_{i,j}) \in \mathbb{R}^{N\times d}$ be a balanced matrix such that its singular values are in the range $[1/\kappa, 1]$. Let $\mathbf{y} = (y_1, y_2, \ldots, y_N)^T \in \mathbb{R}^N$ be a balanced unit vector. Suppose $(\mathbf{X}, \mathbf{y})$ is well-behaved. Given $\epsilon > 0$ and access to the procedures $\mathcal{P}_x$ and $\mathcal{P}_y$ described above, the goal is to output a vector $\beta := (\beta_1, \beta_2, \ldots, \beta_d)^T \in \mathbb{R}^d$ satisfying $\|\beta - \hat{\beta}\|_\infty \le \epsilon$, where $\hat{\beta} := \mathbf{X}^+\mathbf{y}$, succeeding with high probability (e.g. at least 2/3).*

Problem 2 (LR-Q). *Let $\mathbf{X} = (x_{i,j}) \in \mathbb{R}^{N\times d}$ be a balanced matrix such that its singular values are in the range $[1/\kappa, 1]$. Let $\mathbf{y} = (y_1, y_2, \ldots, y_N)^T \in \mathbb{R}^N$ be a balanced unit vector. Given $\epsilon > 0$ and access to the procedures $\mathcal{P}_x$ and $\mathcal{P}_y$ described above, the goal is output an ϵ-additive approximation of $\tau := \|\Pi(\mathbf{X})\mathbf{y}\|^2 / \|\mathbf{y}\|^2$, succeeding with high probability (e.g. at least 2/3).*

Although in the above problems we assume that the singular values of $\mathbf{X}$ lie in the range $[1/\kappa, 1]$ and $\|\mathbf{y}\| = 1$, this is without loss of generality. Suppose instead that the singular values of $\mathbf{X}$ lie in the range $[a/\kappa, a]$ and $\|\mathbf{y}\| = b$, for some constants $a, b > 0$.

Namely, $\mathbf{X}$ and $\mathbf{y}$ are rescaled by a factor of a and b, respectively. Then $\hat{\beta} = \mathbf{X}^{+}\mathbf{y}$ is rescaled by a factor of b/a. So we only need to multiply the result of LR-P by this factor. On the other hand, $\tau = \|\Pi(\mathbf{X})\mathbf{y}\|^2/\|\mathbf{y}\|^2$ is immune to this rescaling. So we do not need to make any change to the result of LR-Q.

We will develop quantum algorithms for solving the above problems. We quantify the resource requirements of these algorithms using two measures. The *query complexity* is the number of uses of the procedures $\mathcal{P}_x$ and $\mathcal{P}_y$ in the algorithm. The *gate complexity* is the number of 2-qubit gates used in the algorithm. An algorithm is *gate-efficient* is if it is gate complexity is larger than its query complexity only by a logarithmic factor. Formally, an algorithm with query complexity Q is gate-efficient if its gate complexity is $O(Q \cdot \mathrm{poly}(\log(QN)))$. All the algorithms presented in this paper will be gate-efficient.

III. HAMILTONIAN SIMULATION

Hamiltonian simulation is an important topic that has received a lot of attention in the past years [5, 6, 10–21]. Recently, Low and Chuang [6] proposed a techinque named *quantum signal processing*, and showed how to use this technique and Childs' quantum walk [14] to simulate sparse Hamiltonians nearly optimally. Later, in Ref. [5], they proposed another technique called *qubitization*, and demonstrated how to use this technique and quantum signal processing to simulate a larger class of Hamiltonians efficiently. In this paper, we will use the following variant of their result:

Theorem 1 (Adapted from Theorem 1 of Ref. [5])**.** *Let $\hat{U}$ and $\hat{G}$ be unitary operators on n and k ($< n$) qubits, respectively, such that $\langle G|\hat{U}|G\rangle = \hat{H}$ is a Hermitian operator on $n-k$ qubits, where $|G\rangle := \hat{G}|0^k\rangle$. Then there exists a gate-efficient algorithm that simulates $e^{-i\hat{H}t}$ with precision ϵ and failure probability $O(\epsilon)$ by making $O(t + \log(1/\epsilon))$ uses of controlled-$\hat{G}$ and controlled-$\hat{U}$.*

Theorem 1 provides a way to simulate a nonsparse Hamiltonian, provided that this Hamiltonian can be embedded into a larger unitary operator in the way described above. Using this fact, we develop an efficient procedure for simulating $e^{-i\mathbf{A}t}$, which will be a crucial component of our algorithms for solving the LR-P and LR-Q problems. Recall that $\mathbf{A}$ is defined by Eq. (28) and is not sparse in general.

Lemma 1. *Let $\mathbf{X}$ be defined as in LR-P or LR-Q. Let $\mathbf{A} := |1\rangle\langle 0| \otimes \mathbf{X}^T + |0\rangle\langle 1| \otimes \mathbf{X}$. Then there exists a gate-efficient procedure that simulates $e^{-i\mathbf{A}t}$ with precision ϵ and failure probability $O(\epsilon)$ by making*

$$O\left(d\left(\sqrt{d}t + \log\left(\frac{1}{\epsilon}\right)\right)\right)$$

uses of $\mathcal{P}_x$.

Proof. Let $\tilde{\mathbf{X}} = (\tilde{x}_{i,j}) := \mathbf{X} \cdot \sqrt{N}/(\sigma\sqrt{d})$, where $\sigma := \sigma(\mathbf{X}) = O(1)$. Then we claim

$$\left\|\tilde{\mathbf{X}}\right\|_{2,\infty} = \max_{1\le i\le N} \|\tilde{\mathbf{x}}_i\| \tag{41}$$

$$= \max_{1\le i\le N} \sqrt{\sum_{j=1}^{d} |\tilde{x}_{i,j}|^2} \tag{42}$$

$$\le 1. \tag{43}$$

To see this, recall that the singular values of $\mathbf{X}$ are in the range $[1/\kappa, 1]$. So

$$\|\mathbf{X}\|_{\mathrm{F}}^2 = \mathrm{tr}\left(\mathbf{X}^T\mathbf{X}\right) = \sum_{j=1}^{d} (s_j(\mathbf{X}))^2 \le d. \tag{44}$$

This implies that

$$\|\mathbf{X}\|_{2,\infty} = \max_{1\le i\le N} \|\mathbf{x}_i\| \tag{45}$$

$$= \frac{\sigma\|\mathbf{X}\|_{\mathrm{F}}}{\sqrt{N}} \tag{46}$$

$$\le \frac{\sigma\sqrt{d}}{\sqrt{N}}. \tag{47}$$

Using this fact and $\tilde{\mathbf{X}} = \mathbf{X} \cdot \sqrt{N}/(\sigma\sqrt{d})$, we obtain Eq. (43), as desired.

Now let $\hat{V}$ be a unitary operator such that

$$\hat{V}|0,i\rangle_1|0,0^m\rangle_2|0\rangle_3 = |0,i\rangle_1|\varphi_i\rangle_{2,3},\ 1 \le i \le N, \tag{48}$$

$$\hat{V}|1,j\rangle_1|0,0^m\rangle_2|0\rangle_3 = |1,j\rangle_1|\psi\rangle_{2,3},\ 1 \le j \le d, \tag{49}$$

where $m = \Theta(\log(N))$,

$$|\varphi_i\rangle_{2,3} := \sum_{j=1}^{d} \tilde{x}_{i,j}|1,j\rangle_2|0\rangle_3 + \sqrt{1 - \|\tilde{\mathbf{x}}_i\|^2}|1,1\rangle_2|1\rangle_3 \tag{50}$$

and

$$|\psi\rangle_{2,3} := \frac{1}{\sqrt{N}} \sum_{i=1}^{N} |0,i\rangle_2|0\rangle_3. \tag{51}$$

Let $\mathrm{SWAP}_{1,2}$ be the swap operator on the first two registers, i.e. $\mathrm{SWAP}_{1,2}|\varphi\rangle_1|\psi\rangle_2 = |\psi\rangle_1|\varphi\rangle_2$ for all states $|\varphi\rangle$ and $|\psi\rangle$. Then we define

$$\hat{W} := (\mathrm{SWAP}_{1,2} \otimes I_3) \cdot \hat{V} \tag{52}$$

and

$$\hat{U} := \hat{W}^\dagger \hat{V}. \tag{53}$$

In addition, let $|G\rangle := |0, 0^m\rangle_2 |0\rangle_3$. Then by a direct calculation, one can verify that

$$\hat{H} := \langle G|\hat{U}|G\rangle = \frac{\mathbf{A}}{\sigma\sqrt{d}}. \tag{54}$$

We will show below that $\hat{U}$ can be implemented by a gate-efficient procedure that makes use $O(d)$ uses of $\mathcal{P}_x$. Then by Theorem 1,

$$e^{-i\mathbf{A}t} = e^{-i\hat{H}\sigma\sqrt{d}t} \tag{55}$$

can be implemented with precision ϵ and failure probability $O(\epsilon)$ by a gate-efficient procedure that makes

$$O\left(d\left(\sigma\sqrt{d}t + \log\left(\frac{1}{\epsilon}\right)\right)\right) \tag{56}$$

$$= O\left(d\left(\sqrt{d}t + \log\left(\frac{1}{\epsilon}\right)\right)\right) \tag{57}$$

uses of $\mathcal{P}_x$ (recall that $\sigma = O(1)$), as claimed.

Clearly, $\mathrm{SWAP}_{1,2}$ can be implemented in time $\mathrm{poly}(\log(N))$. So it remains to show that $\hat{V}$ can be implemented by a gate-efficient procedure that makes use $O(d)$ uses of $\mathcal{P}_x$. To prove this, first note that the mapping

$$|1, j\rangle_1 |0, 0^m\rangle_2 |0\rangle_3 \to |1, j\rangle_1 |\psi\rangle_{2,3} \tag{58}$$

can be implemented in time $O(\log(N))$, since $|\psi\rangle_{2,3} = \frac{1}{\sqrt{N}} \sum_{i=1}^N |0, i\rangle_2 |0\rangle_3$ is easy to prepare. Meanwhile, we can accomplish the transformation

$$|0, i\rangle_1 |0, 0^m\rangle_2 |0\rangle_3 \to |0, i\rangle_1 |\varphi_i\rangle_{2,3} \tag{59}$$

as follows. First, we learn $x_{i,1}$, $x_{i,2}$, ..., $x_{i,d}$ by making $O(d)$ uses of $\mathcal{P}_x$, and obtain the state

$$|0, i\rangle_1 |0, 0^m\rangle_2 |0\rangle_3 \left(\bigotimes_{j=1}^d |x_{i,j}\rangle\right)_4. \tag{60}$$

Then, we perform a unitary operation on the second and third registers depending on the content of the last register, and convert $|0, 0^m\rangle_2 |0\rangle_3$ into $|\varphi_i\rangle_{2,3}$. This step can be achieved in time $O(d \cdot \log(N))$, since $|\varphi_i\rangle_{2,3} = \sum_{j=1}^d \tilde{x}_{i,j} |1, j\rangle_2 |0\rangle_3 + \sqrt{1 - \|\tilde{\mathbf{x}}_i\|^2} |1, 1\rangle_2 |1\rangle_3$ is a $O(d)$-sparse vector in an $O(N)$-dimensional Hilbert space [22]. Finally, we uncompute the $x_{i,j}$'s in the last register by making $O(d)$ uses of $\mathcal{P}_x$, and obtain the desired state $|0, i\rangle_1 |\varphi_i\rangle_{2,3}$. This process requires $O(d)$ uses of $\mathcal{P}_x$ and is gate-efficient. Combining the above facts, we know that $\hat{V}$ can be implemented by a gate-efficient procedure that makes $O(d)$ uses of $\mathcal{P}_x$, as claimed. □

We remark that our embedding construction in the proof of Lemma 1 looks similar to the construction in Refs. [14, 18]. However, we emphasize that our high-level strategy for simulating the Hamiltonian $\mathbf{A}$ is very different from that of Refs. [14, 18]. Specifically, Refs. [14, 18] simulate a Hamiltonian by embedding it into a quantum walk operator and then performing phase estimation on this operator. By contrast, we simulate $e^{-i\mathbf{A}t}$ by embedding $\mathbf{A}$ into a unitary operator $\hat{U}$ (which is not a quantum walk) in certain way and then invoking the method of Ref. [5] for Hamiltonian simulation (which is arguably more advanced than phase-estimation-based methods). So the similarity between the proof of Lemma 1 and the construction in Refs. [14, 18] is superficial rather than essential.

IV. FINDING THE LEAST-SQUARES FIT

In this section, we present a quantum algorithm for solving the LR-P problem, i.e. finding the parameters $\hat{\beta} = \mathbf{X}^+\mathbf{y}$ of the least-squares fit $\mathbf{y} \approx \mathbf{X}\hat{\beta}$ for a given data set $(\mathbf{X}, \mathbf{y})$. Roughly speaking, this algorithm computes $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_d)^T$ in three stages. The first stage estimates the absolute values of the $\hat{\beta}_i$'s. The second stage determines the signs of these parameters, up to a global sign ± 1. That is, up to this stage, we obtain a vector $\beta \in \mathbb{R}^d$ which is close to either $\hat{\beta}$ or $-\hat{\beta}$. The final stage decides which of the two cases holds. This algorithm relies on several subroutines (besides the one for Hamiltonian simulation in Lemma 1). One is the following procedure for preparing the state $|\mathbf{y}\rangle = \sum_{i=1}^N y_i |i\rangle$ (recall that $\|\mathbf{y}\| = 1$).

Lemma 2. *Let $\mathbf{y}$ be defined as in LR-P or LR-Q. Then the state $|\mathbf{y}\rangle = \sum_{i=1}^N y_i |i\rangle$ can be prepared with precision δ by a gate-efficient procedure that makes $O(\log(1/\delta))$ uses of $\mathcal{P}_y$.*

Proof. Consider the following procedure which transforms $|0^n\rangle$ into $|\mathbf{y}\rangle$ probabilistically, where $n = \Theta(\log(N))$. First, we map $|0^n\rangle$ to $\frac{1}{\sqrt{N}} \sum_{i=1}^N |i\rangle$ in time $O(\log(N))$. Then, we convert this state into

$\frac{1}{\sqrt{N}}\sum_{i=1}^{N}|i\rangle|y_i\rangle$ by making $O(1)$ uses of $\mathcal{P}_y$. Next, we append an ancilla qubit in state $|0\rangle$, and perform the controlled-rotation

$$|y_i\rangle|0\rangle \to |y_i\rangle\left(\frac{y_i}{\|\mathbf{y}\|_\infty}|0\rangle + \sqrt{1-\frac{|y_i|^2}{\|\mathbf{y}\|_\infty^2}}|1\rangle\right) \quad (61)$$

on the last two registers, where $\|\mathbf{y}\|_\infty = \max_i|y_i| = \Theta\left(\frac{1}{\sqrt{N}}\right)$ (since $\mathbf{y}$ is a balanced unit vector). After that, we measure the ancilla qubit, and with probability $\frac{\|\mathbf{y}\|^2}{N\|\mathbf{y}\|_\infty^2} = \Omega(1)$, the outcome is 0 and we obtain the state $\sum_{i=1}^{N} y_i|i\rangle|y_i\rangle$. Finally, we uncompute y_i in the second register by making $O(1)$ uses of $\mathcal{P}_y$, and obtain the desired state $|\mathbf{y}\rangle = \sum_{i=1}^{N} y_i|i\rangle$.

The above procedure, denoted by $\mathcal{A}$, makes $O(1)$ uses of $\mathcal{P}_y$, is gate-efficient, and has success probability $\Omega(1)$. We use Grover's $\pi/3$-amplitude amplification (i.e. the generalization of fixed-point quantum search) [23] to raise the success probability to $1-O(\delta^2)$. This boosted procedure, denoted by $\mathcal{A}'$, requires $O(\log(1/\delta))$ repetitions of $\mathcal{A}$, and satisfies

$$\mathcal{A}'\big|0^l\big\rangle|0^n\rangle = \sqrt{1-\delta'}\big|0^l\big\rangle|\mathbf{y}\rangle + \sqrt{\delta'}\big|\Phi^\perp\big\rangle, \quad (62)$$

where l is a positive integer, $\delta' = O(\delta^2)$, and $\big|\Phi^\perp\big\rangle$ is a normalized state satisfying $(\big|0^l\big\rangle\big\langle 0^l\big| \otimes I)\big|\Phi^\perp\big\rangle = 0$. This implies that

$$\big\|\mathcal{A}'\big|0^l\big\rangle|0^n\rangle - \big|0^l\big\rangle|\mathbf{y}\rangle\big\|^2 = (1-\sqrt{1-\delta'})^2 + \delta' \quad (63)$$
$$= O(\delta^2). \quad (64)$$

Furthermore, $\mathcal{A}'$ makes $O(\log(1/\delta))$ uses of $\mathcal{P}_y$, and is gate-efficient. So $\mathcal{A}'$ satisfies all the desired properties. This concludes the proof. □

Our algorithm for solving the LR-P problem also requires the following procedures for computing $|\hat{\beta}_i|$ and $|\hat{\beta}_i - \hat{\beta}_j|$.

Lemma 3. *Let $\mathbf{X}$, $\mathbf{y}$ and $\hat{\beta}$ be defined as in LR-P. Then there exists a gate-efficient quantum algorithm that makes*

$$O\left(\frac{d^{1.5}\kappa^3}{\epsilon^2}\cdot \mathrm{poly}\left(\log\left(\frac{\kappa}{\epsilon\delta}\right)\right)\right)$$

uses of $\mathcal{P}_x$ and $\mathcal{P}_y$, and outputs an ϵ-additive approximation of $|\hat{\beta}_i|$, for any given $i \in \{1,2,\ldots,d\}$, succeeding with probability at least $1-\delta$.

Proof. Let $\mathbf{A} := |1\rangle\langle 0| \otimes \mathbf{X}^T + |0\rangle\langle 1| \otimes \mathbf{X}$ and $|\mathbf{b}\rangle := |0\rangle|\mathbf{y}\rangle$. Suppose $\mathbf{X}$ has the singular value decomposition

$$\mathbf{X} = \sum_{j=1}^{d} s_j|\mathbf{u}_j\rangle\langle\mathbf{v}_j|, \quad (65)$$

where $s_j \in [1/\kappa, 1]$, $|\mathbf{u}_j\rangle \in \mathbb{R}^N$ and $|\mathbf{v}_j\rangle \in \mathbb{R}^d$ are unit vectors, for all $j \in \{1,2,\ldots,d\}$. Then $\mathbf{A}$ has the spectral decomposition

$$\mathbf{A} = \sum_{j=1}^{d} s_j|+_j\rangle\langle+_j| - \sum_{j=1}^{d} s_j|-_j\rangle\langle-_j|, \quad (66)$$

where

$$|\pm_j\rangle := \frac{1}{\sqrt{2}}(|0\rangle|\mathbf{u}_j\rangle \pm |1\rangle|\mathbf{v}_j\rangle). \quad (67)$$

So $\mathbf{A}$ is a Hermitian matrix whose nonzero eigenvalues are in the range $D_\kappa := [-1,-1/\kappa]\cup[1/\kappa,1]$. Moreover, $|\mathbf{b}\rangle = |0\rangle|\mathbf{y}\rangle$ is a unit vector, and $\mathbf{A}^+|\mathbf{b}\rangle = |1\rangle|\hat{\beta}\rangle$ by Eq. (33).

We will use a recent technique proposed by Childs, Kothari and Somma [7] to approximately invert the matrix $\mathbf{A}$. Let the function $h(x)$ be defined as

$$h(x) := \sum_{j=0}^{J-1}\sum_{k=-K}^{K}\alpha(j,k)e^{-ix\eta(j,k)}, \quad (68)$$

where

$$\alpha(j,k) := \frac{i}{\sqrt{2\pi}}k\delta_y\delta_z^2 e^{-k^2\delta_z^2/2}, \quad (69)$$
$$\eta(j,k) := jk\delta_y\delta_z, \quad (70)$$

for some $J = \Theta((\kappa/\epsilon)\cdot\log(\kappa/\epsilon))$, $K = \Theta(\kappa\cdot\log(\kappa/\epsilon))$, $\delta_y = \Theta\left(\epsilon/\sqrt{\log(\kappa/\epsilon)}\right)$ and $\delta_z = \Theta\left(1/(\kappa\sqrt{\log(\kappa/\epsilon)})\right)$. Then $h(x)$ is ϵ-close to $1/x$ on the domain D_κ [7], i.e.

$$\big|h(x) - x^{-1}\big| \le \epsilon, \quad \forall x \in D_\kappa. \quad (71)$$

Then since $\mathbf{A}$ is a Hermitian matrix whose nonzero eigenvalues are in the range D_κ, we have

$$\big\|h(\mathbf{A}) - \mathbf{A}^+\big\| \le \epsilon. \quad (72)$$

This implies that

$$\big\|h(\mathbf{A})|\mathbf{b}\rangle - \mathbf{A}^+|\mathbf{b}\rangle\big\| \le \epsilon, \quad (73)$$

as $|\mathbf{b}\rangle$ is a unit vector. Moreover, Ref. [7] shows that

$$\alpha := \sum_{j=0}^{J-1}\sum_{k=-K}^{K}|\alpha(j,k)| = \Theta\left(\kappa\sqrt{\log(\kappa/\epsilon)}\right), \quad (74)$$

and

$$|\eta(j,k)| \le JK\delta_y\delta_z = \Theta(\kappa \cdot \log(\kappa/\epsilon)), \tag{75}$$

for all j, k.

Now let $|\mathbf{z}\rangle := \mathbf{A}^+|\mathbf{b}\rangle = |1\rangle|\hat{\beta}\rangle$ and $|\mathbf{z}'\rangle := h(\mathbf{A})|\mathbf{b}\rangle$. Then $\||\mathbf{z}\rangle - |\mathbf{z}'\rangle\| = O(\epsilon)$ by Eq. (73). Thus, for any $i \in \{1, 2, \ldots, d\}$, we have

$$\left|\langle 1, i|\mathbf{z}'\rangle - \hat{\beta}_i\right| = |\langle 1, i|\mathbf{z}'\rangle - \langle 1, i|\mathbf{z}\rangle| \tag{76}$$

$$\le \||\mathbf{z}'\rangle - |\mathbf{z}\rangle\| \tag{77}$$

$$= O(\epsilon). \tag{78}$$

So in order to estimate $|\hat{\beta}_i|$ up to additive error $O(\epsilon)$, we only need to obtain an $O(\epsilon)$-additive approximation of $|\langle 1, i|\mathbf{z}'\rangle|$. This can be achieved as follows.

Let V be a unitary operator such that

$$V|0^m\rangle = \frac{1}{\sqrt{\alpha}} \sum_{j=0}^{J-1} \sum_{k=-K}^{K} \sqrt{|\alpha(j,k)|}|j,k\rangle, \tag{79}$$

where $m = O(\log(JK)) = O(\log(\kappa/\epsilon))$, and let U be defined as

$$U := i \sum_{j=0}^{J-1} \sum_{k=-K}^{K} |j,k\rangle\langle j,k| \otimes \mathrm{sgn}(k) e^{-i\mathbf{A}\eta(j,k)}. \tag{80}$$

Then we define

$$W := V^\dagger U V. \tag{81}$$

A direct calculation shows that

$$W|0^m\rangle|\mathbf{b}\rangle = \frac{1}{\alpha}|0^m\rangle h(\mathbf{A})|\mathbf{b}\rangle + \left|\Phi^\perp\right\rangle \tag{82}$$

$$= \left(\frac{\|h(\mathbf{A})|\mathbf{b}\rangle\|}{\alpha}\right)|0^m\rangle \frac{h(\mathbf{A})|\mathbf{b}\rangle}{\|h(\mathbf{A})|\mathbf{b}\rangle\|} + \left|\Phi^\perp\right\rangle, \tag{83}$$

where $\left|\Phi^\perp\right\rangle$ is an unnormalized state satisfying $(|0^m\rangle\langle 0^m| \otimes I)\left|\Phi^\perp\right\rangle = 0$. Next, let R be a unitary operator such that

$$R|0\rangle|1,i\rangle = |1\rangle|1,i\rangle, \tag{84}$$

$$R|0\rangle|1,i'\rangle = |0\rangle|1,i'\rangle, \ 1 \le i' \le N, \ i' \ne i, \tag{85}$$

$$R|0\rangle|0,j\rangle = |0\rangle|0,j\rangle, \ 1 \le j \le N. \tag{86}$$

Then by Eqs. (83), (84), (85) and (86), we obtain

$$RW|0^m\rangle_1|0\rangle_2|\mathbf{b}\rangle_3 = \frac{\langle 1, i|\mathbf{z}'\rangle}{\alpha}|0^m\rangle_1|1\rangle_2|1,i\rangle_3 + \sum_{i' \ne i} \frac{\langle 1, i'|\mathbf{z}'\rangle}{\alpha}|0^m\rangle_1|0\rangle_2|1,i'\rangle_3 + \sum_{j} \frac{\langle 0, j|\mathbf{z}'\rangle}{\alpha}|0^m\rangle_1|0\rangle_2|0,j\rangle_3 + \left|\Xi^\perp\right\rangle_{1,2,3}, \tag{87}$$

where W acts on the first and third registers, R acts on the second and third registers, and $\left|\Xi^\perp\right\rangle$ is an unnormalized state satisfying $(|0^m\rangle\langle 0^m| \otimes I)\left|\Xi^\perp\right\rangle = 0$. If we measure the first $m+1$ qubits of this state in the standard basis, the probability of getting outcome 0^m1 is

$$p' := \frac{|\langle 1, i|\mathbf{z}'\rangle|^2}{\alpha^2}. \tag{88}$$

We use amplitude estimation [8] to obtain an ϵ''-additive approximation $\hat{p}'$ of p', where

$$\epsilon'' := \Theta\left(\frac{\epsilon^2}{\alpha^2}\right) = \Theta\left(\frac{\epsilon^2}{\kappa^2 \log(\kappa/\epsilon)}\right), \tag{89}$$

succeeding with probability at least $3/4$. Then $\sqrt{\hat{p}'}$ is an $O\left(\sqrt{\epsilon''}\right)$-additive approximation of $\sqrt{p'}$ (note that $\sqrt{a} - \sqrt{\gamma} \le \sqrt{a-\gamma} \le \sqrt{a+\gamma} \le \sqrt{a} + \sqrt{\gamma}$ for all $a \ge \gamma \ge 0$). As a result,

$$\left||\langle 1, i|\mathbf{z}'\rangle| - \alpha\sqrt{\hat{p}'}\right| = \left|\alpha\sqrt{p'} - \alpha\sqrt{\hat{p}'}\right| \tag{90}$$

$$= O\left(\alpha\sqrt{\epsilon''}\right) \tag{91}$$

$$= O(\epsilon). \tag{92}$$

Namely, $\alpha\sqrt{\hat{p}'}$ is an $O(\epsilon)$-additive approximation of $|\langle 1, i|\mathbf{z}'\rangle|$, as desired.

The above basic algorithm has success probability at least $3/4$. To boost the success probability to at least $1 - \delta$, we repeat this algorithm $O(\log(1/\delta))$ times, and take the median of the estimates from these runs. A standard Chernoff's bound ensures that the failure probability is at most δ.

Let us analyze the complexity of this algorithm. Since we want to estimate p' up to additive error ϵ'', amplitude estimation requires

$$O\left(\frac{1}{\epsilon''}\right) = O\left(\frac{\alpha^2}{\epsilon^2}\right) = O\left(\frac{\kappa^2 \log(\kappa/\epsilon)}{\epsilon^2}\right) \tag{93}$$

repetitions of R, $W = V^\dagger U V$ and the procedure for preparing $|\mathbf{b}\rangle = |0\rangle|\mathbf{y}\rangle$. This means that we need to implement U with precision $O(\epsilon'')$ and failure probability $O(\epsilon'')$. We also need to prepare $|\mathbf{y}\rangle$ with precision $O(\epsilon'')$. By Lemma 1, Eqs. (75) and (80), and Lemma 8 of Ref. [7], U can be implemented with precision $O(\epsilon'')$ and failure probability $O(\epsilon'')$ by a gate-efficient procedure that makes $O\big(d^{1.5}\kappa \cdot \mathrm{poly}(\log(\kappa/\epsilon))\big)$ uses of $\mathcal{P}_x$. Meanwhile, by Lemma 2, $|\mathbf{y}\rangle$ can be prepared with precision $O(\epsilon'')$ by a gate-efficient procedure that makes $O(\log(\kappa/\epsilon))$ uses of $\mathcal{P}_y$. Furthermore, V can be implemented in time $O(\kappa \cdot \mathrm{poly}(\log(\kappa/\epsilon)))$ [7], and clearly R can be

implemented in time $\mathrm{poly}(\log(N))$. As a result, this algorithm makes

$$O\left(\frac{d^{1.5}\kappa^3}{\epsilon^2}\cdot \mathrm{poly}\left(\log\left(\frac{\kappa}{\epsilon\delta}\right)\right)\right) \tag{94}$$

uses of $\mathcal{P}_x$ and $\mathcal{P}_y$, and is gate-efficient, as claimed. □

Lemma 4. *Let $\mathbf{X}$, $\mathbf{y}$ and $\hat{\beta}$ be defined as in LR-P. Then there exists a gate-efficient quantum algorithm that makes*

$$O\left(\frac{d^{1.5}\kappa^3}{\epsilon^2}\cdot \mathrm{poly}\left(\log\left(\frac{\kappa}{\epsilon\delta}\right)\right)\right)$$

uses of $\mathcal{P}_x$ and $\mathcal{P}_y$, and outputs an ϵ-additive approximation of $|\hat{\beta}_i - \hat{\beta}_j|$, for any given $i, j \in \{1, 2, \ldots, d\}$, succeeding with probability at least $1-\delta$.

Proof. Let us use the same notation as in the proof of Lemma 3. The proof of this lemma is quite similar to that one. The main difference is that here we replace R with a unitary operator Q satisfying

$$Q|0\rangle|1,-_{i,j}\rangle = |1\rangle|1,-_{i,j}\rangle, \tag{95}$$
$$Q|0\rangle|1,+_{i,j}\rangle = |0\rangle|1,+_{i,j}\rangle, \tag{96}$$
$$Q|0\rangle|1,l\rangle = |0\rangle|1,l\rangle,\ 1\le l\le N,\ l\ne i,j, \tag{97}$$
$$Q|0\rangle|0,k\rangle = |0\rangle|0,k\rangle,\ 1\le k\le N, \tag{98}$$

where

$$|1,\pm_{i,j}\rangle := |1\rangle\otimes\frac{|i\rangle \pm |j\rangle}{\sqrt{2}}. \tag{99}$$

Then by Eqs. (83), (95), (96), (97) and (98), we get

$$\begin{aligned} QW|0^m\rangle_1|0\rangle_2|\mathbf{b}\rangle_3 = {} & \frac{\langle 1,-_{i,j}|\mathbf{z}'\rangle}{\alpha}|0^m\rangle_1|1\rangle_2|1,-_{i,j}\rangle_3 \\ & + \frac{\langle 1,+_{i,j}|\mathbf{z}'\rangle}{\alpha}|0^m\rangle_1|0\rangle_2|1,+_{i,j}\rangle_3 \\ & + \sum_{l\ne i,j}\frac{\langle 1,l|\mathbf{z}'\rangle}{\alpha}|0^m\rangle_1|0\rangle_2|1,l\rangle_3 \\ & + \sum_{k}\frac{\langle 0,k|\mathbf{z}'\rangle}{\alpha}|0^m\rangle_1|0\rangle_2|0,k\rangle_3 \\ & + |\Xi^\perp\rangle_{1,2,3}, \end{aligned} \tag{100}$$

where W acts on the first and third registers, Q acts on the second and third registers, and $|\Xi^\perp\rangle$ is an unnormalized state satisfying $(|0^m\rangle\langle 0^m|\otimes I)|\Xi^\perp\rangle = 0$. If we measure the first $m+1$ qubits of this state, then the probability of getting outcome $0^m 1$ is

$$p'' := \frac{|\langle 1,-_{i,j}|\mathbf{z}'\rangle|^2}{\alpha^2}. \tag{101}$$

Recall that $|\mathbf{z}\rangle = \mathbf{A}^+|\mathbf{b}\rangle = |1\rangle|\hat{\beta}\rangle$ and $|\mathbf{z}'\rangle = h(\mathbf{A})|\mathbf{b}\rangle$ satisfy $\||\mathbf{z}\rangle - |\mathbf{z}'\rangle\| = O(\epsilon)$. As a result, $|\langle 1,-_{i,j}|\mathbf{z}'\rangle| = \alpha\sqrt{p''}$ is an $O(\epsilon)$-additive approximation of $|\langle 1,-_{i,j}|\mathbf{z}\rangle| = |\hat{\beta}_i - \hat{\beta}_j|/\sqrt{2}$. So in order to estimate $|\hat{\beta}_i - \hat{\beta}_j|$ up to additive error $O(\epsilon)$, we only need to obtain an $O(\epsilon)$-additive approximation of $\alpha\sqrt{p''}$. To achieve this, we use amplitude estimation to obtain an ϵ''-additive approximation $\hat{p}''$ of p'', where $\epsilon'' = \Theta(\epsilon^2/\alpha^2)$, succeeding with probability at least $3/4$. Then $\sqrt{\hat{p}''}$ is an $O(\epsilon/\alpha)$-additive approximation of $\sqrt{p''}$, and hence $\alpha\sqrt{\hat{p}''}$ is an $O(\epsilon)$-additive approximation of $\alpha\sqrt{p''}$, as desired.

The above basic algorithm has success probability at least $3/4$. To raise the success probability to at least $1-\delta$, we repeat this algorithm $O(\log(1/\delta))$ times, and take the median of the estimates from these runs. A standard Chernoff's bound ensures that the failure probability is at most δ.

To analyze the complexity of this algorithm, note that all the parameters are on the same order as in the proof of Lemma 3. Moreover, Q can be implemented in time $\mathrm{poly}(\log(N))$. Therefore, this algorithm makes

$$O\left(\frac{d^{1.5}\kappa^3}{\epsilon^2}\cdot \mathrm{poly}\left(\log\left(\frac{\kappa}{\epsilon\delta}\right)\right)\right) \tag{102}$$

uses of $\mathcal{P}_A$ and $\mathcal{P}_b$, and is gate-efficient, as claimed. □

Our algorithm for solving the LR-P problem also requires the following procedure for determining whether a given vector $\beta \in \mathbb{R}^d$ is close to $\hat{\beta}$ or $-\hat{\beta}$, under the promise that one of these cases holds.

Lemma 5. *Let $\mathbf{X}$, $\mathbf{y}$ and $\hat{\beta}$ be defined as in LR-P. Suppose $\beta \in \mathbb{R}^d$ is given such that either $\|\beta - \hat{\beta}\| \le \delta$ or $\|\beta + \hat{\beta}\| \le \delta$, for some $\delta < \tau/(2\sigma\rho\sqrt{d})$, where $\tau := \tau(\mathbf{X},\mathbf{y})$, $\sigma := \sigma(\mathbf{X})$, and $\rho := \rho(\mathbf{y})$. Then there exits a gate-efficient quantum algorithm that makes $O(d^{1.5}\kappa)$ uses of $\mathcal{P}_x$ and $\mathcal{P}_y$, and determines which case holds, succeeding with high probability (e.g. at least $3/4$).*

Proof. Let $\hat{\mathbf{y}} := \Pi(\mathbf{X})\mathbf{y} = \mathbf{X}\hat{\beta}$. Then since $\tau = \tau(\mathbf{X},\mathbf{y}) = \|\hat{\mathbf{y}}\|^2/\|\mathbf{y}\|^2 = \|\hat{\mathbf{y}}\|^2$ (recall that $\|\mathbf{y}\| = 1$), we have $\|\hat{\mathbf{y}}\| = \sqrt{\tau}$. Meanwhile, recall that the singular values of $\mathbf{X}$ are in the range $[1/\kappa, 1]$. Thus, we have

$$\sqrt{\tau} \le \|\hat{\beta}\| = \left\|\mathbf{X}^+\mathbf{y}\right\| = \left\|\mathbf{X}^+\hat{\mathbf{y}}\right\| \le \kappa\sqrt{\tau}. \tag{103}$$

Note that $\|\hat{\beta} - (-\hat{\beta})\| = 2\|\hat{\beta}\| \ge 2\sqrt{\tau}$. So by the triangle inequality, at least one of $\|\beta - \hat{\beta}\| \ge \sqrt{\tau}$ and $\|\beta - (-\hat{\beta})\| \ge \sqrt{\tau}$ must hold. Then since $\delta <$

$\tau/(2\sigma\rho\sqrt{d}) \le \sqrt{\tau}$ (note that $\tau \le 1$ and $\sigma, \rho, d \ge 1$), the two cases $\|\beta - \hat{\beta}\| \le \delta$ and $\|\beta + \hat{\beta}\| \le \delta$ cannot happen simultaneously.

Recall that we have shown in the proof of Lemma 1 that

$$\|\mathbf{x}_i\| \le \frac{\sigma\sqrt{d}}{\sqrt{N}}, \quad 1 \le i \le N \tag{104}$$

(see Eq. (47)). Combining Eqs. (103) and (104) yields

$$\left|\mathbf{x}_i^T \hat{\beta}\right| \le \frac{\sigma\kappa\sqrt{\tau d}}{\sqrt{N}}, \quad 1 \le i \le N. \tag{105}$$

Moreover, by $\|\mathbf{y}\| = 1$ and $\rho(\mathbf{y}) = \rho$, we obtain

$$|y_i| \le \frac{\rho}{\sqrt{N}}, \quad 1 \le i \le N. \tag{106}$$

Now let $\hat{q}_i := y_i \cdot \mathbf{x}_i^T \hat{\beta}$ for $i \in \{1, 2, \ldots, N\}$. Then Eqs. (105) and (106) imply that

$$|\hat{q}_i| \le \frac{\sigma\rho\kappa\sqrt{\tau d}}{N}, \quad 1 \le i \le N, \tag{107}$$

Furthermore, we have

$$\begin{aligned}
\sum_{i=1}^N \hat{q}_i &= \sum_{i=1}^N y_i \cdot \mathbf{x}_i^T \hat{\beta} && (108)\\
&= \mathbf{y}^T \mathbf{X} \hat{\beta} && (109)\\
&= \mathbf{y}^T \Pi(\mathbf{X}) \mathbf{y} && (110)\\
&= \|\hat{\mathbf{y}}\|^2 && (111)\\
&= \tau. && (112)
\end{aligned}$$

Now let $q_i := y_i \cdot \mathbf{x}_i^T \beta$ for $i \in \{1, 2, \ldots, N\}$. We claim that we can distinguish the cases $\|\beta - \hat{\beta}\| \le \delta$ and $\|\beta + \hat{\beta}\| \le \delta$ by estimating the quantity $\sum_{i=1}^N q_i$ up to additive error $\tau/2$. To prove this, let us consider these two cases separately:

- Case 1: $\|\beta - \hat{\beta}\| \le \delta < \tau/(2\sigma\rho\sqrt{d})$. Using Eqs. (104) and (106), we get

$$\begin{aligned}
|q_i - \hat{q}_i| &= \left|y_i \cdot \mathbf{x}_i^T \left(\beta - \hat{\beta}\right)\right| && (113)\\
&\le |y_i| \|\mathbf{x}_i\| \|\beta - \hat{\beta}\| && (114)\\
&\le \frac{\rho}{\sqrt{N}} \cdot \frac{\sigma\sqrt{d}}{\sqrt{N}} \cdot \delta && (115)\\
&< \frac{\tau}{2N}, && (116)
\end{aligned}$$

  for all $i \in \{1, 2, \ldots, N\}$. Then by Eqs. (107) and (116), we find that

$$\begin{aligned}
|q_i| &< |\hat{q}_i| + |q_i - \hat{q}_i| && (117)\\
&\le \frac{\sigma\rho\kappa\sqrt{\tau d}}{N} + \frac{\tau}{2N} && (118)\\
&\le \frac{2\sigma\rho\kappa\sqrt{d}}{N}, && (119)
\end{aligned}$$

  for all $i \in \{1, 2, \ldots, N\}$ (note that $\rho, \sigma, \kappa, d \ge 1$ and $\tau \le 1$). Furthermore, Eqs. (112) and (116) imply that

$$\begin{aligned}
\sum_{i=1}^N q_i &\ge \sum_{i=1}^N \hat{q}_i - \sum_{i=1}^N |q_i - \hat{q}_i| && (120)\\
&> \tau - \frac{\tau}{2} && (121)\\
&= \frac{\tau}{2}. && (122)
\end{aligned}$$

- Case 2: $\|\beta + \hat{\beta}\| \le \delta < \tau/(2\sigma\rho\sqrt{d})$. Using Eqs. (104) and (106), we get

$$\begin{aligned}
|q_i + \hat{q}_i| &= \left|y_i \cdot \mathbf{x}_i^T \left(\beta + \hat{\beta}\right)\right| && (123)\\
&\le |y_i| \|\mathbf{x}_i\| \|\beta + \hat{\beta}\| && (124)\\
&\le \frac{\rho}{\sqrt{N}} \cdot \frac{\sigma\sqrt{d}}{\sqrt{N}} \cdot \delta && (125)\\
&< \frac{\tau}{2N}, && (126)
\end{aligned}$$

  for all $i \in \{1, 2, \ldots, N\}$. Then by Eqs. (107) and (126), we find that

$$\begin{aligned}
|q_i| &< |\hat{q}_i| + |q_i + \hat{q}_i| && (127)\\
&\le \frac{\sigma\rho\kappa\sqrt{\tau d}}{N} + \frac{\tau}{2N} && (128)\\
&\le \frac{2\sigma\rho\kappa\sqrt{d}}{N}, && (129)
\end{aligned}$$

  for all $i \in \{1, 2, \ldots, N\}$ (note that $\rho, \sigma, \kappa, d \ge 1$ and $\tau \le 1$). Furthermore, Eqs. (112) and (126) imply that

$$\begin{aligned}
\sum_{i=1}^N q_i &\le -\sum_{i=1}^N \hat{q}_i + \sum_{i=1}^N |q_i + \hat{q}_i| && (130)\\
&< -\tau + \frac{\tau}{2} && (131)\\
&= -\frac{\tau}{2}. && (132)
\end{aligned}$$

Comparing Eqs. (122) and (132), we know that we can distinguish the two cases $\|\beta - \hat{\beta}\| \le \delta$ and $\|\beta + \hat{\beta}\| \le \delta$ by estimating $\sum_{i=1}^N q_i$ up to additive error $\tau/2$, as claimed.

We obtain a $\tau/2$-additive approximation of $\sum_{i=1}^N q_i$ as follows. Let U be a unitary operator such that

$$U|i\rangle|0\rangle = |i\rangle|\psi_i\rangle, \quad 1 \le i \le N, \tag{133}$$

where

$$|\psi_i\rangle := \sqrt{\frac{1}{2} + \frac{Nq_i}{2\Delta}}|0\rangle + \sqrt{\frac{1}{2} - \frac{Nq_i}{2\Delta}}|1\rangle \tag{134}$$

in which $\Delta := 2\sigma\rho\kappa\sqrt{d}$. Note that U is a valid unitary operator, since $N|q_i| \le \Delta$ by Eqs. (119) and (129). Then we have

$$U\left(\frac{1}{\sqrt{N}}\sum_{i=1}^N |i\rangle\right)|0\rangle = \frac{1}{\sqrt{N}}\sum_{i=1}^N |i\rangle|\psi_i\rangle. \tag{135}$$

If we measure the second register of this state in the standard basis, then the probability of obtaining outcome 0 is

$$p := \frac{1}{2} + \frac{\sum_{i=1}^N q_i}{\Delta}. \tag{136}$$

We use amplitude estimation to obtain an $\tau/(2\Delta)$-additive approximation $\hat{p}$ of p, succeeding with high probability (e.g. at least 3/4). Then $(\hat{p} - 1/2)\Delta$ is a $\tau/2$-additive approximation of $\sum_{i=1}^N q_i$, as desired.

The unitary operator U can be implemented as follows. For any $i \in \{1, 2, \ldots, N\}$, given the state $|i\rangle|0\rangle$, we first transform it into $|i\rangle|0\rangle|q_i\rangle$, where $q_i = y_i(\sum_{j=1}^d x_{i,j}\beta_j)$ can be computed by making $O(d)$ uses of $\mathcal{P}_x$ and $\mathcal{P}_y$. Then we perform the controlled-rotation

$$|0\rangle|q_i\rangle \to |\psi_i\rangle|q_i\rangle \tag{137}$$

on the last two registers. After that, we uncompute q_i in the last register by making $O(d)$ uses of $\mathcal{P}_x$ and $\mathcal{P}_y$, and get the desired state $|i\rangle|\psi_i\rangle$. This implemenation of U requires $O(d)$ uses of $\mathcal{P}_x$ and $\mathcal{P}_y$, and is gate-efficient.

Since we want to estimate p up to additive error $\tau/(2\Delta)$, amplitude estimation requires

$$O\left(\frac{\Delta}{\tau}\right) = O\left(\frac{\sigma\rho\kappa\sqrt{d}}{\tau}\right) = O\left(\kappa\sqrt{d}\right) \tag{138}$$

repetitions of U (recall that $\sigma = O(1)$, $\rho = O(1)$ and $\tau = \Omega(1)$). As a result, this algorithm makes $O\big(d^{1.5}\kappa\big)$ uses of $\mathcal{P}_x$ and $\mathcal{P}_y$, and is gate-efficient, as claimed.

□

Now we are ready to state our algorithm for solving the LR-P problem.

Theorem 2. *The LR-P problem can be solved by a gate-efficient quantum algorithm that makes*

$$O\left(\frac{d^{2.5}\kappa^3}{\delta^2} \cdot \mathrm{poly}\left(\log\left(\frac{d\kappa}{\delta}\right)\right)\right)$$

uses of $\mathcal{P}_x$ and $\mathcal{P}_y$, where $\delta := \min\{\epsilon, 1/d\}$.

Proof. **Algorithm:** Let $\mathbf{X}$ and $\mathbf{y}$ be defined as in LR-P. Let $\tau := \tau(\mathbf{X}, \mathbf{y}) = \Omega(1)$, $\sigma := \sigma(\mathbf{X}) = O(1)$ and $\rho := \rho(\mathbf{y}) = O(1)$. We use the following algorithm to obtain a vector $\beta = (\beta_1, \beta_2, \ldots, \beta_d)^T \in \mathbb{R}^d$ satisfying $\|\beta - \hat{\beta}\|_\infty \le \epsilon$, succeeding with probability at least 2/3:

1. Let $\epsilon' := \min\{\tau/(2\sigma\rho d), \epsilon\}$.
2. For each $j \in \{1, 2, \ldots, d\}$, we run the algorithm in Lemma 3 to obtain an $\epsilon'/6$-additive approximation μ_j of $|\hat{\beta}_j|$, succeeding with probability at least $1 - 1/(25d)$.
3. Let $S := \{j \in \{1, 2, \ldots, d\} : \mu_j > 2\epsilon'/3\}$. If $S = \emptyset$, then this algorithm fails; otherwise, we continue as follows.
4. Pick arbitrary $j_0 \in S$. For each $j \in S$, $j \ne j_0$, we run the algorithm in Lemma 4 to obtain an $\epsilon'/6$-additive approximation γ_j of $|\hat{\beta}_{j_0} - \hat{\beta}_j|$, succeeding with probability at least $1 - 1/(25d)$.
5. For each $j \in \{1, 2, \ldots, d\}$, we define $s_j \in \{-1, 0, 1\}$ as follows:
   - If $j \notin S$, then $s_j = 0$.
   - If $j = j_0 \in S$, then $s_j = 1$.
   - Otherwise, we have $j \in S$ and $j \ne j_0$. If $||\mu_{j_0} - \mu_j| - \gamma_j| \le \epsilon'/2$, then $s_j = 1$; otherwise, $s_j = -1$.
6. Let $\beta' = (\beta'_1, \beta'_2, \ldots, \beta'_d)^T \in \mathbb{R}^d$ be defined as $\beta'_j := s_j\mu_j$ for each $j \in \{1, 2, \ldots, d\}$. We will prove below that, with high probability, either $\|\beta' - \hat{\beta}\| < \tau/(2\sigma\rho\sqrt{d})$ or $\|\beta' + \hat{\beta}\| < \tau/(2\sigma\rho\sqrt{d})$. We run the algorithm in Lemma 5 to determine which case holds, succeeding with probability at least 3/4. If the first case holds, then we return $\beta := \beta'$ as our estimate of $\hat{\beta}$; otherwise, we return $\beta := -\beta'$ as our estimate of $\hat{\beta}$.

Correctness: Let us call the case where all the instances of the algorithms in Lemmas 3, 4

and 5 succeed the *typical* case. By union bound, the probability of this case happening is at least $1 - 2d/(25d) - 1/4 > 2/3$. We will prove that in the typical case, our algorithm outputs a correct β (i.e. $\|\beta - \hat{\beta}\|_\infty \le \epsilon$) with certainty.

In the typical case, we have

$$\left|\mu_i - \left|\hat{\beta}_i\right|\right| \le \frac{\epsilon'}{6}, \quad 1 \le i \le d, \tag{139}$$

and

$$\left|\gamma_j - \left|\hat{\beta}_{j_0} - \hat{\beta}_j\right|\right| \le \frac{\epsilon'}{6}, \quad \forall j \in S, j \ne j_0. \tag{140}$$

Then using the definition of S, we get

$$\begin{aligned} \left|\hat{\beta}_j\right| &\ge |\mu_j| - \left|\mu_i - \left|\hat{\beta}_i\right|\right| && (141)\\ &> \frac{2\epsilon'}{3} - \frac{\epsilon'}{6} && (142)\\ &= \frac{\epsilon'}{2}, \quad \forall j \in S, && (143) \end{aligned}$$

and

$$\begin{aligned} \left|\hat{\beta}_j\right| &\le |\mu_j| + \left|\mu_i - \left|\hat{\beta}_i\right|\right| && (144)\\ &\le \frac{2\epsilon'}{3} + \frac{\epsilon'}{6} && (145)\\ &= \frac{5\epsilon'}{6}, \quad \forall j \notin S. && (146) \end{aligned}$$

Recall that we have shown in the proof of Lemma 5 that

$$\left\|\hat{\beta}\right\| = \sqrt{\sum_{i=1}^{d} |\hat{\beta}_i|^2} \ge \sqrt{\tau} \tag{147}$$

(see Eq. (103)). This implies that there exists some $i_0 \in \{1, 2, \ldots, d\}$ such that

$$\left|\hat{\beta}_{i_0}\right| \ge \sqrt{\frac{\tau}{d}} \ge \frac{\tau}{\sigma\rho d} \ge 2\epsilon' \tag{148}$$

(note that $\sigma, \rho, d \ge 1$ and $\tau \le 1$). Then by Eqs. (139) and (148), we obtain

$$\begin{aligned} \mu_{i_0} &\ge |\beta_{i_0}| - \left|\mu_{i_0} - \left|\hat{\beta}_{i_0}\right|\right| && (149)\\ &\ge 2\epsilon' - \frac{\epsilon'}{6} && (150)\\ &> \frac{2\epsilon'}{3}. && (151) \end{aligned}$$

Thus, we have $i_0 \in S$ and $S \ne \emptyset$. So our algorithm does not fail in the typical case.

Now we claim that $s_j = \mathrm{sgn}(\hat{\beta}_j) \cdot \mathrm{sgn}(\hat{\beta}_{j_0})$ for any $j \in S$. The proof is as follows.

- If $j = j_0$, then $s_j = 1$ by definition.
- If $j \ne j_0$ and $\mathrm{sgn}(\hat{\beta}_j) = \mathrm{sgn}(\hat{\beta}_{j_0})$, then we have

$$\left|\hat{\beta}_{j_0} - \hat{\beta}_j\right| = \left|\left|\hat{\beta}_{j_0}\right| - \left|\hat{\beta}_j\right|\right|. \tag{152}$$

Combining Eqs. (139), (140) and (152) gives

$$\begin{aligned} ||\mu_{j_0} - \mu_j| - \gamma_j| &\le \left|\mu_{j_0} - \left|\hat{\beta}_{j_0}\right|\right| \\ &\quad + \left|\mu_j - \left|\hat{\beta}_j\right|\right| \\ &\quad + \left|\gamma_j - \left|\hat{\beta}_{j_0} - \hat{\beta}_j\right|\right| && (153)\\ &\le \frac{\epsilon'}{6} + \frac{\epsilon'}{6} + \frac{\epsilon'}{6} && (154)\\ &= \frac{\epsilon'}{2}. && (155) \end{aligned}$$

This implies that $s_j = 1$ for this j.

- If $j \ne j_0$ and $\mathrm{sgn}(\hat{\beta}_j) = -\mathrm{sgn}(\hat{\beta}_{j_0})$, then we have

$$\begin{aligned} \left|\hat{\beta}_{j_0} - \hat{\beta}_j\right| &= \left|\hat{\beta}_{j_0}\right| + \left|\hat{\beta}_j\right| \\ &= \left|\left|\hat{\beta}_{j_0}\right| - \left|\hat{\beta}_j\right|\right| && (156)\\ &\quad + 2\min\left\{\left|\hat{\beta}_{j_0}\right|, \left|\hat{\beta}_j\right|\right\} \\ &> \left|\left|\hat{\beta}_{j_0}\right| - \left|\hat{\beta}_j\right|\right| + \epsilon', && (157) \end{aligned}$$

since $|\hat{\beta}_{j_0}|, |\hat{\beta}_j| > \epsilon'/2$ by Eq. (143). Combining Eqs. (139), (140) and (157) yields

$$\begin{aligned} ||\mu_{j_0} - \mu_j| - \gamma_j| &\ge \epsilon' - \left|\mu_{j_0} - \left|\hat{\beta}_{j_0}\right|\right| \\ &\quad - \left|\mu_j - \left|\hat{\beta}_j\right|\right| \\ &\quad - \left|\gamma_j - \left|\hat{\beta}_{j_0} - \hat{\beta}_j\right|\right| && (158)\\ &> \epsilon' - \frac{\epsilon'}{6} - \frac{\epsilon'}{6} - \frac{\epsilon'}{6} && (159)\\ &= \frac{\epsilon'}{2}. && (160) \end{aligned}$$

This implies that $s_j = -1$ for this j.

The fact that $s_j = \mathrm{sgn}(\hat{\beta}_j) \cdot \mathrm{sgn}(\hat{\beta}_{j_0})$ for all $j \in S$ implies that either

$$\mathrm{sgn}(\beta'_j) = \mathrm{sgn}(\hat{\beta}_j), \quad \forall j \in S, \tag{161}$$

or

$$\mathrm{sgn}(\beta'_j) = -\mathrm{sgn}(\hat{\beta}_j), \quad \forall j \in S. \tag{162}$$

Moreover, by Eq. (139), we know that

$$\left||\beta'_j| - |\hat{\beta}_j|\right| \leq \frac{\epsilon'}{6}, \quad \forall j \in S. \tag{163}$$

As a result, we have either

$$\left|\beta'_j - \hat{\beta}_j\right| \leq \frac{\epsilon'}{6}, \quad \forall j \in S, \tag{164}$$

or

$$\left|\beta'_j + \hat{\beta}_j\right| \leq \frac{\epsilon'}{6}, \quad \forall j \in S. \tag{165}$$

Meanwhile, for any $j \notin S$, we have $s_j = 0$ and $|\hat{\beta}_j| \leq 5\epsilon'/6$ by Eq. (146). It follows that $\beta'_j = 0$ and

$$\left|\beta'_j - \hat{\beta}_j\right| = \left|\beta'_j + \hat{\beta}_j\right| \leq \frac{5\epsilon'}{6}, \quad \forall j \notin S. \tag{166}$$

Combining the cases $j \in S$ and $j \notin S$, we know that either

$$\|\beta' - \hat{\beta}\|_\infty \leq \frac{5\epsilon'}{6} < \epsilon' \tag{167}$$

or

$$\|\beta' + \hat{\beta}\|_\infty \leq \frac{5\epsilon'}{6} < \epsilon'. \tag{168}$$

As a result, we have either

$$\|\beta' - \hat{\beta}\| < \sqrt{d}\epsilon' \leq \frac{\tau}{2\sigma\rho\sqrt{d}} \tag{169}$$

or

$$\|\beta' + \hat{\beta}\| < \sqrt{d}\epsilon' \leq \frac{\tau}{2\sigma\rho\sqrt{d}}. \tag{170}$$

In the typical case, our algorithm in Lemma 5 correctly determines which case holds. If the first case holds, then it outputs $\beta = \beta'$ which satisfies $\|\beta - \hat{\beta}\|_\infty < \epsilon' \leq \epsilon$; otherwise, it outputs $\beta = -\beta'$ which also satisfies $\|\beta - \hat{\beta}\|_\infty < \epsilon' \leq \epsilon$, as desired.

Complexity: Recall that $\epsilon' = \min\{\tau/(2\sigma\rho d), \epsilon\}$ and $\delta = \min\{1/d, \epsilon\}$, where $\tau = \Omega(1)$, $\sigma = O(1)$ and $\rho = O(1)$. So we have $\epsilon' = \Omega(\delta)$. Let us analyze the complexity of each step. Step 2 makes $O(d)$ uses of the algorithm in Lemma 3, so it requires

$$O\left(d \cdot \frac{d^{1.5}\kappa^3}{(\epsilon')^2} \cdot \mathrm{poly}\left(\log\left(\frac{d\kappa}{\epsilon'}\right)\right)\right) \tag{171}$$

$$= O\left(\frac{d^{2.5}\kappa^3}{\delta^2} \cdot \mathrm{poly}\left(\log\left(\frac{d\kappa}{\delta}\right)\right)\right) \tag{172}$$

uses of $\mathcal{P}_x$ and $\mathcal{P}_y$, and is gate-efficient. Step 4 makes $O(d)$ uses of the algorithm in Lemma 4, so it requires

$$O\left(d \cdot \frac{d^{1.5}\kappa^3}{(\epsilon')^2} \cdot \mathrm{poly}\left(\log\left(\frac{d\kappa}{\epsilon'}\right)\right)\right) \tag{173}$$

$$= O\left(\frac{d^{2.5}\kappa^3}{\delta^2} \cdot \mathrm{poly}\left(\log\left(\frac{d\kappa}{\delta}\right)\right)\right) \tag{174}$$

uses of $\mathcal{P}_x$ and $\mathcal{P}_y$, and is gate-efficient. Step 6 makes $O(1)$ uses of the algorithm in Lemma 5, so it requires $O\big(\kappa d^{1.5}\big)$ uses of of $\mathcal{P}_x$ and $\mathcal{P}_y$, and is gate-efficient. Furthermore, the classical computation in this algorithm takes $O(d)$ time. As a result, this algorithm makes

$$O\left(\frac{d^{2.5}\kappa^3}{\delta^2} \cdot \mathrm{poly}\left(\log\left(\frac{d\kappa}{\delta}\right)\right)\right) \tag{175}$$

uses of $\mathcal{P}_x$ and $\mathcal{P}_y$, and is gate-efficient, as claimed. □

Our algorithm for computing $\hat{\beta} = \mathbf{X}^+\mathbf{y}$ is more efficient than an alternative one in which one first creates multiple copies of the state proportional to $\hat{\beta}$ and then uses statistical sampling and quantum state tomography to determine the $\hat{\beta}_j$'s (as suggested by Ref. [1]). The main reason is that, in order to obtain an ϵ-additive approximation of $|\hat{\beta}_j|^2$, the sampling-based approach would require $O\big(1/\epsilon^2\big)$ copies of the state encoding $\hat{\beta}$, but amplitude estimation only needs $O(1/\epsilon)$ repetitions of the procedure for preparing this state. So it is more efficient to couple the state generation process with amplitude estimation (as we did in our algorithm) rather than statistical sampling.

We also remark that the algorithm in Lemma 3 can be modified to produce a quantum state approximately proportional to $\hat{\beta}$. Specifically, note that if we measure the first register of $W|0^m\rangle|\mathbf{b}\rangle$ (in Eq. (83)) in the standard basis, then conditioning on the outcome being 0^m, we would obtain the normalized version of $h(\mathbf{A})|\mathbf{b}\rangle$, which is close to the normalized version of $\mathbf{A}^+|\mathbf{b}\rangle = |1\rangle|\hat{\beta}\rangle$. The probablity of this event happening is $\|h(\mathbf{A})|\mathbf{b}\rangle\|^2/\alpha^2 = \Omega\big(1/\alpha^2\big)$. We can use amplitude amplification to raise this probability to $\Omega(1)$, which requires $O(\alpha)$ repetitions of W and the procedure for preparing $|\mathbf{b}\rangle$. This leads to a gate-efficient algorithm that makes

$$O\Big(d^{1.5}\kappa^2 \cdot \mathrm{poly}\Big(\log\Big(\frac{\kappa}{\epsilon}\Big)\Big)\Big) \tag{176}$$

uses of $\mathcal{P}_x$ and $\mathcal{P}_y$, and prepares a quantum state ϵ-close to $\frac{|\hat{\beta}\rangle}{\||\hat{\beta}\rangle\|}$ in l^2 norm, succeeding with probability $\Omega(1)$ (with a flag indicating success). By utilizing Ambainis' *variable-time amplitude amplification* [24], we can reduce the κ-dependence from quadratic to linear, as done in Section 5 of Ref. [7]. This leads to a gate-efficient algorithm with query complexity

$$O\Big(d^{1.5}\kappa \cdot \mathrm{poly}\Big(\log\Big(\frac{\kappa}{\epsilon}\Big)\Big)\Big) \tag{177}$$

for the same task.

One may compare this algorithm for preparing a quantum state approximately proportional to the optimal parameters

$$\hat{\beta} = \mathbf{X}^{+}\mathbf{y} = (\mathbf{X}^{T}\mathbf{X})^{-1}\mathbf{X}^{T}\mathbf{y} \tag{178}$$

with the one in Ref. [1] for the same task. Our algorithm is based on the singular value decomposition (SVD) of $\mathbf{X}$, and it applies $\mathbf{X}^{+}$ to $\mathbf{y}$ in a direct manner. Consequently, it has only linear dependence on the condition number κ of $\mathbf{X}$. By contrast, Ref. [1] needs to first apply $\mathbf{X}^{T}$ to $\mathbf{y}$, which incurs a κ factor in the complexity; then it needs to apply $(\mathbf{X}^{T}\mathbf{X})^{-1}$ to the output of the first step, which incurs another κ^2 factor in the complexity. So its overall complexity is at least cubic in κ. This means that our algorithm has polynomially better dependence on κ than the one in Ref. [1]. Furthermore, due to the fact we use the new strategy of Ref. [7] for matrix inversion, our algorithm also has exponential better dependence on the desired precision ϵ in the output state.

V. ESTIMATING THE QUALITY OF THE LEAST-SQUARES FIT

In this section, we describe a quantum algorithm for solving the LR-Q problem, i.e. estimating the quality $\tau = \|\mathbf{X}\hat{\beta}\|^2/\|\mathbf{y}\|^2$ of the least-squares fit $\mathbf{y} \approx \mathbf{X}\hat{\beta}$ for a given data set $(\mathbf{X}, \mathbf{y})$ (without computing the parameters $\hat{\beta}$ explicitly). This algorithm requires the following variant of phase estimation [25, 26], which decides whether the eigenphase corresponding to an eigenvector of a unitary operator is θ or far away from θ, for some given $\theta \in [0, 2\pi)$, succeeding with probability close to 1. (Similar procedures have been used in Refs. [7, 27, 28].)

Lemma 6. *Let U be a unitary operator with eigenvectors $|\psi_j\rangle$ such that $U|\psi_j\rangle = e^{i\theta_j}|\psi_j\rangle$ for some $\theta_j \in [0, 2\pi)$. Let $\theta \in [0, 2\pi)$ and let Δ, $\delta \in (0, 1)$. Then there is a unitary procedure $\mathcal{P}$ that makes $O((1/\Delta)\cdot\log(1/\delta))$ uses of U, and uses $\mathrm{poly}(\log(1/(\Delta\delta)))$ additional 2-qubit gates, and satisfies*

$$\mathcal{P}|0\rangle|0^l\rangle|\psi_j\rangle = (\alpha_{j,0}|0\rangle|\eta_{j,0}\rangle + \alpha_{j,1}|1\rangle|\eta_{j,1}\rangle)|\psi_j\rangle, \tag{179}$$

where $l = O(\log(1/\Delta)\log(1/\delta))$, $|\alpha_{j,0}|^2 + |\alpha_{j,1}|^2 = 1$, $|\eta_{j,0}\rangle$ and $|\eta_{j,1}\rangle$ are two normalized states, and

- *If $\theta_j = \theta$, then $|\alpha_{j,0}|^2 \geq 1 - \delta$.*
- *If $|\theta_j - \theta| \geq \Delta$, then $|\alpha_{j,1}|^2 \geq 1 - \delta$.*

Proof. We can get a $\Delta/2$-additive approximation of θ_j by using the standard phase estimation, which makes $O(1/\Delta)$ uses of U and uses $\mathrm{poly}(\log(1/\Delta))$ additional 2-qubit gates. This is sufficient to distinguish the two cases. However, it only succeeds with probability $\Omega(1)$. To raise this probability to at least $1-\delta$, we repeat this procedure $O(\log(1/\delta))$ times and check whether the median of the estimates is $\Delta/2$-close to θ. A standard Chernoff's bound ensures that the failure proability is at most δ. This boosted procedure, denoted by $\mathcal{P}$, makes $O((1/\Delta)\cdot\log(1/\delta))$ uses of U, and uses $\mathrm{poly}(\log(1/(\Delta\delta)))$ additional 2-qubit gates, and satisfies all the desired properties. □

Theorem 3. *The LR-Q problem can be solved by a gate-efficient quantum algorithm that makes*

$$O\left(\frac{d^{1.5}\kappa}{\epsilon}\cdot\mathrm{poly}\left(\log\left(\frac{\kappa}{\epsilon}\right)\right)\right)$$

uses of $\mathcal{P}_x$ and $\mathcal{P}_y$.

Proof. **Algorithm:** Let $\mathbf{X}$ and $\mathbf{y}$ be defined as in LR-Q. We use the following algorithm to obtain an ϵ-additive approximation of $\tau = \|\Pi(\mathbf{X})\mathbf{y}\|^2/\|\mathbf{y}\|^2 = \|\Pi(\mathbf{X})\mathbf{y}\|^2$ (recall that $\|\mathbf{y}\| = 1$), succeeding with probability at least 2/3. Let $\mathbf{A} := |1\rangle\langle 0| \otimes \mathbf{X}^T + |0\rangle\langle 1| \otimes \mathbf{X}$ and $|\mathbf{b}\rangle := |0\rangle|\mathbf{y}\rangle$. Let $\mathcal{P}$ be the unitary procedure in Lemma 6 for $U = e^{-i\mathbf{A}}$, $\theta = 0$, $\Delta = 1/(2\kappa)$ and $\delta = \epsilon/2$. Suppose

$$\mathcal{P}|0\rangle_1|0^l\rangle_2|\mathbf{b}\rangle_3 = \mu_0|0\rangle_1|\varphi_0\rangle_{2,3} + \mu_1|1\rangle_1|\varphi_1\rangle_{2,3}, \tag{180}$$

where $l = O(\log(1/\Delta)\log(1/\delta))$, $|\mu_0|^2 + |\mu_1|^2 = 1$, and $|\varphi_0\rangle_{2,3}$ and $|\varphi_1\rangle_{2,3}$ are some normalized states on the second and third registers. We use amplitude estimation to get an $\epsilon/2$-additive approximation $\hat{r}$ of $r := |\mu_1|^2$, succeeding with probability at least 3/4. Then we return $\hat{r}$ as our estimate of τ. During this process, we use the procedure in Lemma 1 to implement $U = e^{-i\mathbf{A}}$ with precision $O(\epsilon^2/\kappa^2)$ (and failure probability $O(\epsilon^2/\kappa^2)$), and use the procedure in Lemma 2 to prepare $|\mathbf{y}\rangle$ with precision $O(\epsilon^2)$.

Correctness: Suppose $\mathbf{X}$ has the singular value decomposition

$$\mathbf{X} = \sum_{j=1}^{d} s_j|\mathbf{u}_j\rangle\langle\mathbf{v}_j|, \tag{181}$$

where $s_j \in [1/\kappa, 1]$, $|\mathbf{u}_j\rangle \in \mathbb{R}^N$ and $|\mathbf{v}_j\rangle \in \mathbb{R}^d$ are unit vectors, for all $j \in \{1, 2, \ldots, d\}$. Then $\mathbf{A}$ has

the spectral decomposition

$$\mathbf{A} = \sum_{j=1}^{d} s_j |+_j\rangle\langle+_j| - \sum_{j=1}^{d} s_j |-_j\rangle\langle-_j|, \quad (182)$$

where

$$|\pm_j\rangle := \frac{1}{\sqrt{2}}(|0\rangle|\mathbf{u}_j\rangle \pm |1\rangle|\mathbf{v}_j\rangle). \quad (183)$$

Meanwhile, we can write $|\mathbf{y}\rangle$ as

$$|\mathbf{y}\rangle = \sum_{j=1}^{d} \alpha_j |\mathbf{u}_j\rangle + \alpha\big|\Phi^{\perp}\big\rangle, \quad (184)$$

where $\sum_{j=1}^{d}|\alpha_j|^2 + |\alpha|^2 = 1$, and $\big|\Phi^{\perp}\big\rangle$ is some normalized state satisfying $\big\langle \mathbf{u}_j\big|\Phi^{\perp}\big\rangle = 0$ for all j. Note that

$$\tau = \|\Pi(\mathbf{X})\mathbf{y}\|^2 = \sum_{j=1}^{d}|\alpha_j|^2. \quad (185)$$

By Eqs. (183) and (184), we obtain

$$\begin{aligned} |\mathbf{b}\rangle &= |0\rangle|\mathbf{y}\rangle & (186)\\ &= \sum_{j=1}^{d} \alpha_j |0\rangle|\mathbf{u}_j\rangle + \alpha|0\rangle\big|\Phi^{\perp}\big\rangle & (187)\\ &= \sum_{j=1}^{d} \frac{\alpha_j}{\sqrt{2}}(|+_j\rangle + |-_j\rangle) + \alpha|0\rangle\big|\Phi^{\perp}\big\rangle. & (188) \end{aligned}$$

Note that $|0\rangle\big|\Phi^{\perp}\big\rangle$ is an eigenvector of $\mathbf{A}$ with eigenvalue 0, i.e. $\mathbf{A}|0\rangle\big|\Phi^{\perp}\big\rangle = 0$.

Now, since the eigenphase gap around 0 of $U = e^{-i\mathbf{A}}$ is at least $1/\kappa$, by Lemma 6 and our choice of parameters, we get

$$\mathcal{P}|0\rangle\big|0^l\big\rangle|+_j\rangle = \big(\gamma^+_{j,0}|0\rangle\big|\phi^+_{j,0}\big\rangle + \gamma^+_{j,1}|1\rangle\big|\phi^+_{j,1}\big\rangle\big)|+_j\rangle, \quad (189)$$

$$\mathcal{P}|0\rangle\big|0^l\big\rangle|-_j\rangle = \big(\gamma^-_{j,0}|0\rangle\big|\phi^-_{j,0}\big\rangle + \gamma^-_{j,1}|1\rangle\big|\phi^-_{j,1}\big\rangle\big)|-_j\rangle, \quad (190)$$

where $\big|\gamma^{\pm}_{j,1}\big|^2 \geq 1-\delta$, $\big|\gamma^{\pm}_{j,0}\big|^2 \leq \delta$, $\big|\phi^{\pm}_{j,0}\big\rangle$ and $\big|\phi^{\pm}_{j,1}\big\rangle$ are some normalized states, for all $j \in \{1,2,\ldots,d\}$, and

$$\mathcal{P}|0\rangle\big|0^l\big\rangle|0\rangle\big|\Phi^{\perp}\big\rangle = (\eta_0|0\rangle|\psi_0\rangle + \eta_1|1\rangle|\psi_1\rangle)|0\rangle\big|\Phi^{\perp}\big\rangle, \quad (191)$$

where $|\eta_0|^2 \geq 1-\delta$, $|\eta_1|^2 \leq \delta$, $|\psi_0\rangle$ and $|\psi_1\rangle$ are some normalized states. As a result, we have

$$\begin{aligned} \mathcal{P}|0\rangle\big|0^l\big\rangle|\mathbf{b}\rangle = &\sum_{j=1}^{d} \frac{\alpha_j}{\sqrt{2}}\big(\gamma^+_{j,0}|0\rangle\big|\phi^+_{j,0}\big\rangle + \gamma^+_{j,1}|1\rangle\big|\phi^+_{j,1}\big\rangle\big)|+_j\rangle\\ &+ \sum_{j=1}^{d} \frac{\alpha_j}{\sqrt{2}}\big(\gamma^-_{j,0}|0\rangle\big|\phi^-_{j,0}\big\rangle + \gamma^-_{j,1}|1\rangle\big|\phi^-_{j,1}\big\rangle\big)|-_j\rangle\\ &+ \alpha(\eta_0|0\rangle|\psi_0\rangle + \eta_1|1\rangle|\psi_1\rangle)|0\rangle\big|\Phi^{\perp}\big\rangle. \quad (192) \end{aligned}$$

It follows that

$$r = \frac{1}{2}\sum_{j=1}^{d}|\alpha_j|^2\Big(\big|\gamma^+_{j,1}\big|^2 + \big|\gamma^-_{j,1}\big|^2\Big) + |\alpha|^2|\eta_1|^2. \quad (193)$$

Note that since $|\gamma^{\pm}_{j,1}|^2 \approx 1$ and $|\eta_1| \approx 0$, we have $r \approx \tau$ by Eqs. (185) and (193). More precisely, the difference between r and τ can be bounded using the triangle inequality:

$$\begin{aligned} |r-\tau| &\leq \frac{1}{2}\sum_{j=1}^{d}|\alpha_j|^2\Big(1-\big|\gamma^+_{j,1}\big|^2\Big)\\ &\quad + \frac{1}{2}\sum_{j=1}^{d}|\alpha_j|^2\Big(1-\big|\gamma^-_{j,1}\big|^2\Big)\\ &\quad + |\alpha|^2|\eta_1|^2 & (194)\\ &\leq \frac{1}{2}\sum_{j=1}^{d}|\alpha_j|^2\cdot\delta + \frac{1}{2}\sum_{j=1}^{d}|\alpha_j|^2\cdot\delta\\ &\quad + |\alpha|^2\cdot\delta & (195)\\ &= \delta & (196)\\ &= \frac{\epsilon}{2}. & (197) \end{aligned}$$

Namely, r is an $\epsilon/2$-additive approximation of τ. Meanwhile, $\hat{r}$ is an $\epsilon/2$-additive approximation of r. It follows that $\hat{r}$ is an ϵ-additive approximation of τ, as desired.

In the above argument, we have ignored the error in the implementation of $U = e^{-i\mathbf{A}}$ and the error in the preparation of $|\mathbf{y}\rangle$. We will show below that our algorithm only makes $o(\kappa^2/\epsilon^2)$ uses of U and $o(1/\epsilon^2)$ uses of the procedure for preparing $|\mathbf{y}\rangle$. Thus, provided that U is implemented with precision $O(\epsilon^2/\kappa^2)$ (and failure probability $O(\epsilon^2/\kappa^2)$) and $|\mathbf{y}\rangle$ is prepared with precision $O(1/\epsilon^2)$, the error in the final state (compared to the ideal case) is only $o(1)$. Consequently, our algorithm outputs a correct $\hat{r}$ (i.e. $|\hat{r}-\tau| \leq \epsilon$) with probability at least $3/4 - o(1)$.

Complexity: Since we want to estimate r up to additive error $O(\epsilon)$, amplitude estimation requires $O(1/\epsilon)$ repetitions of the procedure $\mathcal{P}$ and the procedure for preparing $|\mathbf{y}\rangle$. Then by Lemma 6, our

algorithm makes

$$O\left(\frac{1}{\epsilon}\cdot\frac{1}{\Delta}\log\left(\frac{1}{\delta}\right)\right)=O\left(\frac{\kappa}{\epsilon}\cdot\log\left(\frac{1}{\epsilon}\right)\right) \quad (198)$$

uses of U. By Lemma 1, $U=e^{-i\mathbf{A}}$ can be implemented with precision $O\big(\epsilon^2/\kappa^2\big)$ (and failure probability $O\big(\epsilon^2/\kappa^2\big)$) by a gate-efficient procedure that makes $O\big(d^{1.5}\cdot\log(\kappa/\epsilon)\big)$ uses of $\mathcal{P}_x$. Meanwhile, by Lemma 2, $|\mathbf{y}\rangle$ can be prepared with precision $O\big(\epsilon^2\big)$ by a gate-efficient procedure that makes $O(\log(1/\epsilon))$ uses of $\mathcal{P}_y$. As a result, this algorithm makes

$$O\left(\frac{d^{1.5}\kappa}{\epsilon}\cdot\mathrm{poly}\left(\log\left(\frac{\kappa}{\epsilon}\right)\right)\right) \quad (199)$$

uses of $\mathcal{P}_x$ and $\mathcal{P}_y$, and is gate-efficient, as claimed. □

Comparing Theorem 2 and Theorem 3, one can see that it is easier to estimate the quality of the least-squares fit $\mathbf{y}\approx\mathbf{X}\hat{\beta}$ than to find its parameters $\hat{\beta}=\mathbf{X}^{+}\mathbf{y}$ explicitly. Thus, in practice, we can first run the algorithm in Theorem 3 to check whether a given data set is well-behaved (e.g. $\tau\geq 2/3$). If so, then we run the algorithm in Theorem 2 to fit a linear regression model to this data set. The total cost of this process is dominated by that of the second stage.

VI. LOWER BOUND ON THE COMPLEXITY OF LINEAR REGRESSION

Our quantum algorithm for computing $\hat{\beta}=\mathbf{X}^{+}\mathbf{y}$ has polynomial dependence on the condition number κ of the design matrix $\mathbf{X}$. In this section, we show that this dependence is indeed necessary. To prove this, we need the following lower bound on the quantum query complexity of a weaker version of *unstructured search*.

Lemma 7. *Let $f:\{1,2,\ldots,N\}\to\{0,1\}$ be a function such that $f(x)=1$ if and only if $x=z$ for some unknown $z\in\{1,2,\ldots,N\}$. Let $\mathcal{P}_f$ be a procedure that on input $x\in\{1,2,\ldots,N\}$, outputs the value of $f(x)$. Then one has to make $\Omega\Big(\sqrt{N}/\log(N)\Big)$ queries to $\mathcal{P}_f$ to determine whether the unknown z is larger than $\lfloor N/2\rfloor$ or not (succeeding with probability at least $2/3$).*

Proof. Suppose we can solve the given problem by making Q queries to $\mathcal{P}_f$. Then we can find the unknown z by making $O(Q\log(N))$ queries to $\mathcal{P}_f$. The idea is to use binary search. Namely, we first test whether z is in the range $[0,\lfloor N/2\rfloor]$ or $[\lfloor N/2\rfloor+1,N]$. If the first case holds, then we test whether z is in the range $[0,\lfloor N/4\rfloor]$ or $[\lfloor N/4\rfloor+1,\lfloor N/2\rfloor]$; otherwise, we test whether z is in the range $[\lfloor N/2\rfloor+1,\lfloor 3N/4\rfloor]$ or $[\lfloor 3N/4\rfloor+1,N]$, and so on. We only need $O(\log(N))$ such tests to locate z, since each test reduces the size of candidate set by a factor of 2. Furthermore, by assumption, each test can be accomplished by making at most Q queries to $\mathcal{P}_f$. Thus, we can find z by making $O(Q\log(N))$ queries to $\mathcal{P}_f$. On the other hand, it is known that unstructured search has quantum query complexity $\Omega\Big(\sqrt{N}\Big)$ [29, 30]. Combining these two facts, we know that $Q=\Omega\Big(\sqrt{N}/\log(N)\Big)$. □

Theorem 4. *The LR-P problem has quantum query complexity $\Omega(\kappa/\log(\kappa))$, where κ is the condition number of the design matrix $\mathbf{X}$.*

Proof. We prove this theorem by showing that for any positive integer N, there exists a balanced matrix $\mathbf{X}\in\mathbb{R}^{N\times 2}$ with singular values $s_1(\mathbf{X})=\Theta\Big(1/\sqrt{N}\Big)$ and $s_2(\mathbf{X})=\Theta(1)$ such that, for $\mathbf{y}=\frac{1}{\sqrt{N}}(1,1,\ldots,1)^T\in\mathbb{R}^N$, $\hat{\beta}=\mathbf{X}^{+}\mathbf{y}$ is either $(1,0)^T$ or $(0,1)^T$, but one has to make $\Omega\Big(\sqrt{N}/\log(N)\Big)$ queries to $\mathbf{X}$ to determine which case holds (succeeding with probability at least $2/3$).

Let $\mathbf{X}$ be an $N\times 2$ matrix such that its entries are all $1/\sqrt{N}$ except one entry 0 (whose location is unknown and arbitrary). Then we know that one column of $\mathbf{X}$ is equal to $\mathbf{y}=\frac{1}{\sqrt{N}}(1,1,\ldots,1)^T$, and the other column of $\mathbf{X}$ is linearly independent from $\mathbf{y}$. Consequently, using the definition

$$\hat{\beta}=\underset{\beta\in\mathbb{R}^2}{\mathrm{argmin}}\|\mathbf{X}\beta-\mathbf{y}\|, \quad (200)$$

we obtain that $\hat{\beta}$ is either $(0,1)^T$ or $(1,0)^T$, depending on whether the entry 0 is in the first or second column of $\mathbf{X}$, respectively. By Lemma 7, one must make $\Omega\Big(\sqrt{N}/\log(N)\Big)$ queries to $\mathbf{X}$ to determine which column contains the entry 0. This implies that one also needs to make $\Omega\Big(\sqrt{N}/\log(N)\Big)$ queries to $\mathbf{X}$ to determine whether $\hat{\beta}=(0,1)^T$ or $\hat{\beta}=(1,0)^T$.

It remains to show that $\mathbf{X}$ also satisfies the other desired properties. First, by a direct calculation, we get that $\|\mathbf{X}\|_{\mathrm{F}}=\Theta(1)$, $\|\mathbf{X}\|_{2,\infty}=\Theta\Big(1/\sqrt{N}\Big)$ and

hence $\sigma(\mathbf{X}) = \Theta(1)$. Second, note that either

$$\mathbf{X}^T\mathbf{X} = \begin{pmatrix} 1 - \frac{1}{N} & 1 - \frac{1}{N} \\ 1 - \frac{1}{N} & 1 \end{pmatrix} \tag{201}$$

or

$$\mathbf{X}^T\mathbf{X} = \begin{pmatrix} 1 & 1 - \frac{1}{N} \\ 1 - \frac{1}{N} & 1 - \frac{1}{N} \end{pmatrix}. \tag{202}$$

By a direct calculation, we find that $\lambda_1(\mathbf{X}^T\mathbf{X}) = \Theta(1/N)$ and $\lambda_2(\mathbf{X}^T\mathbf{X}) = \Theta(1)$. It follows that $s_1(\mathbf{X}) = \sqrt{\lambda_1(\mathbf{X}^T\mathbf{X})} = \Theta\left(1/\sqrt{N}\right)$ and $s_2(\mathbf{X}) = \sqrt{\lambda_2(\mathbf{X}^T\mathbf{X})} = \Theta(1)$, and hence $\kappa(\mathbf{X}) = \Theta\left(\sqrt{N}\right)$. This concludes the proof. □

Clearly, the LR-P problem has time complexity $\Omega(d)$, because simply writing down a d-dimensional vector $\beta \approx \hat{\beta}$ requires this amount of time. Combining this fact and Theorem 4, we know that the algorithm in Theorem 2 cannot be dramatically improved.

It is worth noting that Harrow, Hassidim and Lloyd (HHL) [2] have also given a lower bound on the quantum complexity of matrix inversion. They proved that unless BQP = PSPACE, one cannot solve the matrix inversion problem in quantum time $\kappa^{1-\delta} \cdot \mathrm{poly}(\log(N))$ for some constant $\delta > 0$, where κ and N are the condition number and dimension of the matrix to be inverted, respectively. We remark that this result and Theorem 4 are incomparable. At first glance, it may seem that Theorem 4 is stronger, since it has better dependence on κ and it does not rely on any complexity-theoretic assumption. But recall that in our LR-P problem, we allow the design matrix X to be nonspare, while HHL only allowed sparse matrices in their work. So we only obtain a stronger bound under a stronger assumption. Nevertheless, it may be possible to use our approach to improve HHL's bound, showing that our bound holds in the sparse case as well. This is left as an interesting open question.

VII. DISCUSSION

To summarize, we have presented an efficient quantum algorithm for fitting a linear regression model to a given data set using the least squares approach. Different from previous algorithms which produce a quantum state encoding the optimal parameters, our algorithm outputs these numbers in the classical form. So by running it once, one completely determines the fitted model and then can use it to make predictions on new data at little cost. The running time of this algorithm is polynomial in $\log(N)$, d, κ and $1/\epsilon$, where N is the size of the data set, d is the number of adjustable parameters, κ is the condition number of the design matrix, and ϵ is the desired precision in the output. We also show that the polynomial dependence on d and κ is necessary. Therefore, our algorithm cannot be greatly improved. Furthermore, we also give an efficient quantum algorithm that estimates the quality of the least-squares fit (without computing its parameters explicitly). This algorithm runs faster than the one for finding this fit, and can be used to check whether the given data set qualifies for linear regression in the first place.

One may have noticed that our algorithms actually solve two fundamental problems in linear algebra. One is to apply the pseudoinverse of a dense rectangular matrix to a vector, and the other is to estimate the norm of the projection of this vector onto the range of this matrix. Such problems frequently arise in many scenarios. So it is conceivable that our algorithms may find applications beyond linear regression.

Our algorithms might be improved in a few ways. Ambainis [24] proposed a technique called *variable-time amplitude amplification* and utilized it to enhance the κ-dependence of HHL's algorithm [2] for preparing a state encoding the solution of a linear system (this techique is also used in CKS's algorithm [7]). But it is unknown whether this technique leads to a more efficient algorithm for estimating an entry (or the difference between two entries) of this solution. If so, we would obtain a faster algorithm for fitting a linear regression model to a data set using the least squares approach. On the other hand, for estimating the quality of the fitted model, we still do not know whether the polynomial dependence on κ is necessary. We believe that this is the case, but could not prove it. This is left as an interesting open question.

In this paper, we have focused on linear regression with *ordinary least squares* optimization (which assumes that the errors for different observations are independent). It is also worth investigating the quantum complexity of linear regression with *generalized least squares* optimization (which allows the errors for different observations to be correlated). Furthermore, one might study how these complexities change when *regularization* is used. For ex-

ample, how hard is it to solve *ridge regression* [31] or *Lasso* [32] on a quantum computer? Finally, it would be worth exploring the power and limitation of quantum algorithms for *nonlinear regression*.

Our work is also a new contribution to the nascent field of *quantum machine learning*, which has made a lot of progress in the past years [1, 3, 33–57]. Here we briefly review this broad area, and position our work with the other works in this area (for an excellent review on quantum machine learning, see Ref. [50]). In fact, depending on the types of the learning device and the object to be learned, quantum machine learing can be divided into three branches. The first branch, which is also known as *quantum-enhanced machine learning*, uses quantum mechanics to improve the performance of classical machine learning methods (e.g. [1, 3, 41, 42, 52]). Conversely, the second branch applies classial machine learning methods to the study of quantum systems (e.g. [33, 43]). Finally, the third branch uses quantum approaches to study quantum systems (e.g. [46]). Clearly, our work is an instance of the first kind, i.e. quantum-enhanced machine learning.

Now let us look at quantum-enhanced machine learning more carefully. Traditional machine learning algorithms can be divided into three main groups based on their purpose: *supervised learning* (in which an algorithm learns from example data and associated target responses that can consist of numeric values or string labels), *unsupervised learning* (in which an algorithm learns from plain examples without any associated response), and *reinforcement learning* (in which an agent interacts with an environment and occasionally receives rewards for its actions, which allows the agent to adapt its behavior). There has beening exciting progress in all of these three paradigms. See Refs. [1, 3, 42, 47], Refs. [36, 37, 52] and Refs. [34, 41, 53, 54] for examples of the first, second and third kind, respectively. Our work belongs to the first category, as it concerns least-square linear regression – a typical supervised learning task.

Meanwhile, we can also classify the works on quantum-enhanced machine learning based on the techniques they use. It seems that most of these works fall into three groups according to this criterion. The first group use linear algebra methods (e.g. singular value decomposition), and are usually related to HHL's quantum algorithm for linear systems of equations somehow. Examples include Refs. [1, 3] and this work on least-squares linear regression, Ref. [42] on support vector machine, and Ref. [47] on Guassian processes. This approach could achieve exponential speedup (in some sense) over classical methods. The second group are based on amplitude amplification (including Grover's search and quantum walk). Examples include Ref. [37] on k-medians, Ref. [49] on k-nearest neighbors, Ref. [35] on Google's PageRank, and Ref. [41] on reinforcement learning. This approach usually achieves polynomial speedup over classial methods. Finally, the third group are based on quantum sampling techniques (e.g. quantum annealing). Examples include Refs. [48, 52, 55–57] on (deep) Boltzmann machines. We believe that the field of quantum(-enhanced) machine learning could benefit the most from the marriage of these different ideas, and look forward to seeing more novel quantum algorithms for solving machine learning tasks.

ACKNOWLEDGMENTS

The author thanks Scott Aaronson, Andrew Childs and Umesh Vazirani for helpful discussions and comments. The author also thanks the anonymous referee for providing many useful comments on an earlier version of this paper. Part of this work was done while the author was a graduate student at Computer Science Division, University of California, Berkeley. This research was supported by ARO Grant W911NF- 09-1-0440.

[1] N. Wiebe, D. Braun, and S. Lloyd, Phys. Rev. Lett. **109**, 050505 (2012).
[2] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).
[3] M. Schuld, I. Sinayskiy, and F. Petruccione, Phys. Rev. A **94**, 022342 (2016).
[4] S. Lloyd, M. Mohseni, and P. Rebentrost, Nature Physics **10**, 631 (2014).
[5] G. H. Low and I. L. Chuang, arXiv preprint arXiv:1610.06546 (2016).
[6] G. H. Low and I. L. Chuang, Phys. Rev. Lett. **118**, 010501 (2017).
[7] A. M. Childs, R. Kothari, and R. D. Somma, arXiv preprint arXiv:1511.02306 (2015).
[8] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Contemporary Mathematics **305**, 53 (2002).

[9] G. H. Golub and C. Reinsch, Numerische mathematik **14**, 403 (1970).
[10] S. Lloyd, Science **273**, 1073 (1996).
[11] D. Aharonov and A. Ta-Shma, in *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03 (ACM, New York, NY, USA, 2003) pp. 20–29.
[12] A. M. Childs, *Quantum information processing in continuous time*, Ph.D. thesis, Massachusetts Institute of Technology (2004).
[13] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Communications in Mathematical Physics **270**, 359 (2007).
[14] A. M. Childs, Communications in Mathematical Physics **294**, 581 (2010).
[15] A. M. Childs and R. Kothari, "Simulating sparse hamiltonians with star decompositions," in *Theory of Quantum Computation, Communication, and Cryptography: 5th Conference, TQC 2010, Leeds, UK, April 13-15, 2010, Revised Selected Papers*, edited by W. van Dam, V. M. Kendon, and S. Severini (Springer Berlin Heidelberg, Berlin, Heidelberg, 2011) pp. 94–103.
[16] D. Poulin, A. Qarry, R. Somma, and F. Verstraete, Phys. Rev. Lett. **106**, 170501 (2011).
[17] A. M. Childs and N. Wiebe, Quantum Info. Comput. **12**, 901 (2012).
[18] D. W. Berry and A. M. Childs, Quantum Info. Comput. **12**, 29 (2012).
[19] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, in *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14 (ACM, New York, NY, USA, 2014) pp. 283–292.
[20] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Phys. Rev. Lett. **114**, 090502 (2015).
[21] D. W. Berry, A. M. Childs, and R. Kothari, in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science* (2015) pp. 792–809.
[22] V. V. Shende, S. S. Bullock, and I. L. Markov, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **25**, 1000 (2006).
[23] L. K. Grover, Phys. Rev. Lett. **95**, 150501 (2005).
[24] A. Ambainis, in *29th International Symposium on Theoretical Aspects of Computer Science (STACS 2012)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 14, edited by C. Dürr and T. Wilke (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2012) pp. 636–647.
[25] A. Y. Kitaev, arXiv preprint quant-ph/9511026 (1995).
[26] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, Vol. 454 (The Royal Society, 1998) pp. 339–354.
[27] D. Nagaj, P. Wocjan, and Y. Zhang, Quantum Info. Comput. **9**, 1053 (2009).
[28] G. Wang, arXiv preprint arXiv:1311.1851 (2013).
[29] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, SIAM Journal on Computing **26**, 1510 (1997).
[30] M. Boyer, G. Brassard, P. Hyer, and A. Tapp, Fortschritte der Physik **46**, 493 (1998).
[31] A. E. Hoerl and R. W. Kennard, Technometrics **12**, 55 (1970).
[32] R. Tibshirani, Journal of the Royal Statistical Society. Series B (Methodological) **58**, 267 (1996).
[33] G. Sentís, J. Calsamiglia, R. Munoz-Tapia, and E. Bagan, Scientific reports **2**, 708 (2012).
[34] H. J. Briegel and G. De las Cuevas, Scientific reports **2**, 400 (2012).
[35] G. D. Paparo and M. A. Martin-Delgado, Scientific Reports **2**, 444 (2012).
[36] S. Lloyd, M. Mohseni, and P. Rebentrost, arXiv preprint arXiv:1307.0411 (2013).
[37] E. Aïmeur, G. Brassard, and S. Gambs, Machine Learning **90**, 261 (2013).
[38] S. Aaronson, Nature Physics **11**, 291 (2015).
[39] J. Adcock, E. Allen, M. Day, S. Frick, J. Hinchliff, M. Johnson, S. Morley-Short, S. Pallister, A. Price, and S. Stanisic, arXiv preprint arXiv:1512.02900 (2015).
[40] B. O'Gorman, R. Babbush, A. Perdomo-Ortiz, A. Aspuru-Guzik, and V. Smelyanskiy, The European Physical Journal Special Topics **224**, 163 (2015).
[41] G. D. Paparo, V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel, Phys. Rev. X **4**, 031002 (2014).
[42] P. Rebentrost, M. Mohseni, and S. Lloyd, Phys. Rev. Lett. **113**, 130503 (2014).
[43] N. Wiebe, C. Granade, C. Ferrie, and D. Cory, Phys. Rev. A **89**, 042314 (2014).
[44] M. Schuld, I. Sinayskiy, and F. Petruccione, Contemporary Physics **56**, 172 (2015).
[45] M. Schuld, I. Sinayskiy, and F. Petruccione, Physics Letters A **379**, 660 (2015).
[46] G. Sentís, M. Guţă, and G. Adesso, EPJ Quantum Technology **2**, 17 (2015).
[47] Z. Zhao, J. K. Fitzsimons, and J. F. Fitzsimons, arXiv preprint arXiv:1512.03929 (2015).
[48] S. H. Adachi and M. P. Henderson, arXiv preprint arXiv:1510.06356 (2015).
[49] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum Information and Computation **15** (2015).
[50] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, arXiv preprint arXiv:1611.09347 (2016).
[51] I. Kerenidis and A. Prakash, arXiv preprint arXiv:1603.08675 (2016).
[52] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum Information and Computation **16** (2016).
[53] V. Dunjko, J. M. Taylor, and H. J. Briegel, Phys. Rev. Lett. **117**, 130501 (2016).
[54] D. Crawford, A. Levit, N. Ghadermarzy, J. S. Oberoi, and P. Ronagh, arXiv preprint arXiv:1612.05695 (2016).

[55] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, arXiv preprint arXiv:1601.02036 (2016).

[56] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, arXiv preprint arXiv:1609.02542 (2016).

[57] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Phys. Rev. A **94**, 022308 (2016).