Learning High Dimensional Multi-response Linear Models with Non-oracular Quantum Search

1st Jinyang Chen
Department of Statistics
The University of Georgia
Athens, USA
jinyangchen40519@gmail.com

2nd Cheolwoo Park
Department of Mathematical Sciences
KAIST
Daejeon, South Korea
parkcw2021@kaist.ac.kr

3rd Yuan Ke
Department of Statistics
The University of Georgia
Athens, USA
yuan.ke@uga.edu, ORCID 0000-0001-7291-8302

***Abstract*—This paper studies linear regression models for high dimensional multi-response data with a hybrid quantum computing algorithm. We propose an intuitively appealing estimation method based on identifying the linearly independent columns in the coefficient matrix. Our method relaxes the low rank constraint in the existing literature and allows the rank to diverge with dimensions. The linearly independent columns are selected by a novel non-oracular quantum search (NQS) algorithm which is significantly faster than classical search methods implemented on electronic computers. Besides, NQS achieves a near optimal computational complexity as existing quantum search algorithms but does not require any oracle information of the solution state. We prove the proposed estimation procedure enjoys desirable theoretical properties. Intensive numerical experiments are also conducted to demonstrate the finite sample performance of the proposed method, and a comparison is made with some popular competitors. The results show that our method outperforms all of the alternative methods under various circumstances.**

***Index Terms*—high dimensional data, Grover's algorithm, multi-response linear model, non-oracular quantum search, quantum machine learning**

I. Introduction

High dimensional multi-response datasets are ubiquitous in machine learning applications. The component-wise analysis is not desirable as it does not fully make use of the information available. For example, the observations of one component may contain the information for the others. Overlooking this information will result in the loss of estimation efficiency. To that end, it is necessary to take multivariate approaches for multi-response data analysis. Over the past two decades, the multi-response linear models in the high dimensional setting have been attracting more and more attention than ever before. In particular, many interesting developments in low rank high dimensional multi-response linear models have appeared in literature, see [1]–[7] and references therein. Let $\mathbf{A} \in \mathbb{R}^{p\times q}$ be a coefficient matrix with rank r. A commonly assumed low rank approximation is to decompose $\mathbf{A}$ to $\mathbf{CDQ}$, where $\mathbf{C}$ and $\mathbf{Q}$ are two matrices of size $p \times \hat{r}$ and $\hat{r} \times q$ with $\hat{r}$ being an estimator of r, and $\mathbf{D}$ is a diagonal matrix of size $\hat{r}$. Different approaches may result in different ways to estimate r, see [1], [5]. Although the existing approaches for estimating r enjoy nice asymptotic properties, when the low rank assumption is violated, we often come up against a dilemma: the estimation of $\mathbf{C}$, $\mathbf{Q}$ and $\mathbf{D}$ involves at least $(p+q)r$ unknown parameters, which is more than the unknown parameters in $\mathbf{A}$ without using the decomposition when $r > pq/(p+q)$. Besides, low rank approaches can only determine $\mathbf{A}$ up to a rotation matrix. Let us explain this with a toy example. Suppose we want to apply SVD to a p by q matrix $\mathbf{A}$ and its randomly perturbed version $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{E}$, where $\mathbf{E}$ is an estimation error matrix. Then, the first (in terms of the descending order of singular values) left singular vector of $\tilde{\mathbf{A}}$ may accidentally estimate the second left singular vector of $\mathbf{A}$ due to the presence of random errors and/or the multiplicity of singular values. Further, we refer to Theorem 4 in [8] for theoretical discussions.

Recently, a new line of studies [9], [10] propose to recover the rank and estimate the multi-response linear models without decomposing the coefficient matrix $\mathbf{A}$. The intuition is to represent $\mathbf{A}$ by its linearly independent columns whose size equals the rank r. Statistically, this new learning procedure is more efficient than both least-squares approach and decomposition based approaches as it only needs to estimate $r(p+q) - r^2$ unknown parameters which is clearly less than pq. Despite the statistical advantage of the new learning procedure, the rank of $\mathbf{A}$ is estimated by selecting a set of linearly independent columns that minimize a loss function. When r is unknown and possibly diverges with p and q, identifying linearly independent columns in $\mathbf{A}$ involves a combinatorial search over all subsets of sizes $\{1, \ldots, \min(p, q)\}$ and hence is an NP-hard problem [11]–[13]. Therefore, solving this problem with electronic computers is computationally expensive if not infeasible in high dimensional and large rank scenarios.

Unlike electronic computers, a quantum computer operates on quantum processing units, or qubits, which can take values 0, 1, or both simultaneously due to the superposition property. The left panel of Fig. 1 gives a visualized example of a qubit. The number of complex numbers required to characterize quantum states usually grows exponentially with the size of the system. For example, a quantum system with q qubits can be in infinite many superpositions of 2^q orthonormal states simultaneously, while a classical system can only be in one state at a time [14]. Such a paradigm change has motivated

Park's research is supported by Basic Science Research Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2021R1A2C1092925). Ke acknowledges the support of National Science Foundation (NSF) of the United States grant DMS-2210468.

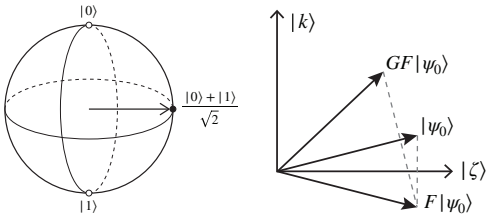

Fig. 1: Left: Visualization of a quantum bit; Right: Geometric interpretation for Grover's operation

significant developments of scalable quantum algorithms in many areas, see [15]–[21] and references therein.

However, existing quantum search algorithms [22]–[26] are, in general, oracular algorithms that rely on an oracle to decide if an item is a solution or not. For example, Grover's algorithm [22] requires an oracle function that can map all solution states to 1 and all non-solution states to 0 with one operation. However, such a piece of oracle information is usually not available in machine learning applications as the solution is generally a function of random observations. When the oracle function is inaccurate, Grover's algorithm may rotate the initial superposition towards the wrong direction and can perform as bad as a random guess. To overcome the limitations, we propose a novel method named non-oracular quantum search (NQS). For a combinatorial search over $D = 2^q$ models, an electrical computing algorithm requires $O(D)$ queries to find the target model. In contrast, within $O(\log_2 D)$ iterations, NQS converges to a superposition that heavily weighs on the solution state and hence outputs the target model with a high probability. The complexity of NQS is upper bounded by the order $O(\sqrt{D}\log_2 D)$ which is only a $\log_2 D$ factor larger than the theoretical lower bound for oracular quantum search [27]. Though the NP-hardness has not been fully conquered, NQS has made a steady step to downscale the complexity of the combinatorial search problem.

Our contributions. (1) The proposed estimation method is statistically more efficient when the dimensions (p and q) and the rank (r) are large. Besides, our approach avoids the identifiability issue in low rank methods which can only determine $\mathbf{A}$ up to a rotation matrix. (2) We propose a non-oracular search algorithm (NQS) that is free of oracle information. To the best of our knowledge, this is the first study of a non-oracular quantum search algorithm. Besides, NQS provides a general non-oracular quantum search framework as the state loss function can be tailored for various machine learning problems, such as best subset selection, support vector machine, clustering, and optimal design problems.

II. Model and estimation method

Denote $Y \in \mathbb{R}^q$ a vector of all response variables and $X \in \mathbb{R}^p$ a vector of all covariates. The multi-response linear model follows

$$Y = \mathbf{A}^\top X + \boldsymbol{\epsilon}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{p\times q}$ is a coefficient matrix with an unknown rank r and $r \le q < p$, and $\boldsymbol{\epsilon} \in \mathbb{R}^q$ is a vector of random errors. Further, we assume

$$E(\boldsymbol{\epsilon}|X) = \mathbf{0} \quad \text{and} \quad \text{cov}(\boldsymbol{\epsilon}|X) = \Sigma.$$

When a p by q matrix $\mathbf{A}$ is of rank r, each column of $\mathbf{A}$ can be written as a linear combination of the r linearly independent columns of $\mathbf{A}$. Motivated by this observation, we propose to estimate $\mathbf{A}$ and rank r by recovering the linearly independent columns of $\mathbf{A}$.

Let $\{\mathbf{x}_i, \mathbf{y}_i\}$, $i = 1, \ldots, n$, be a sample drawn from $\{X, Y\}$, with $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top$ and $\mathbf{y}_i = (y_{i1}, \ldots, y_{iq})^\top$. We propose to estimate $\mathbf{A}$ by minimizing

$$L_n(s; j_1, \ldots, j_s) = \sum_{i=1}^{n} \Bigg\{ \sum_{l=1}^{s} \left(y_{ij_\ell} - \mathbf{x}_i^\top \mathbf{a}_{j_\ell}\right)^2 + \sum_{k \notin \{j_1,\ldots,j_s\}} \left(y_{ik} - \mathbf{x}_i^\top \sum_{\ell=1}^{s} b_{k\ell}\mathbf{a}_{j_\ell} \right)^2 \Bigg\}, \tag{2}$$

with respect to $\mathbf{a}_{j_\ell}$'s, $b_{k\ell}$'s, s and $\{j_1, \ldots, j_s\}$, where $1 \le j_1 < \cdots < j_s \le q$. For ease of presentation, the dependency of $\mathbf{x}_i$'s, $\mathbf{y}_i$'s, $\mathbf{a}_{j_\ell}$'s and $b_{k\ell}$'s has been suppressed in $L_n(\cdot)$.

For $\ell \in \{1, \ldots, \hat{s}\}$, denote the minimizer of (2) by

$$\hat{s},\ \hat{\mathcal{D}} = \{\hat{j}_1, \ldots, \hat{j}_{\hat{s}}\},\ \hat{\mathbf{a}}_j,\ j \in \hat{\mathcal{D}},\ \text{and}\ \hat{b}_{k\ell},\ k \in \hat{\mathcal{D}}^c,$$

where $\hat{\mathcal{D}}^c$ is the complement of $\hat{\mathcal{D}}$. Then, the j-th column of $\mathbf{A}$ is estimated by $\hat{\mathbf{a}}_j$, if $j \in \hat{\mathcal{D}}$ or $\sum_{\ell=1}^{\hat{s}} \hat{b}_{j\ell}\hat{\mathbf{a}}_{\hat{j}_\ell}$, if $j \in \hat{\mathcal{D}}^c$. The collection of these column-wise estimators, denoted as $\hat{\mathbf{A}}$, gives the estimator of $\mathbf{A}$.

When s and $\{j_1, \ldots, j_s\}$ are fixed, the minimization of $L_n(s; j_1, \ldots, j_s)$ is a quadratic optimization problem which has been studied in [9]. However, minimizing $L_n(s; j_1, \ldots, j_s)$ with respect to s and $\{j_1, \ldots, j_s\}$ is non-convex and involves a combinatorial search over $\sum_{s=1}^{q}\binom{q}{s} = 2^q - 1$ options. Without a convex relaxation or a stage-wise approximation, the optimization is computationally equivalent to the best subset selection, which is a well known NP-hard problem [11]. We will study this combinatorial optimization problem with the help of quantum computing in Section III.

III. Non-oracular quantum search

A. Preliminaries

To facilitate the discussion in the paper, we review some essential notations and definitions for quantum computing. The typical vector space of interest in quantum search is a Hilbert space $\mathcal{H}$ of dimension $D = 2^q$ with a positive integer q. For a vector $|a\rangle \in \mathcal{H}$, we denote its dual vector as $\langle a|$, which is an element in the dual Hilbert space $\mathcal{H}^*$. Besides, $\mathcal{H}$ and $\mathcal{H}^*$ together naturally induce an inner product $\langle a|b\rangle = \langle \mathbf{a}, \mathbf{b}\rangle$, which is also known as a 'bra-ket'. We say $|a\rangle$ is a unit vector if $\langle a|a\rangle = 1$. A set of D vectors $\mathcal{D} = \{|0\rangle, \ldots, |D-1\rangle\}$ is called an orthonormal basis of $\mathcal{H}$ if $\langle i|j\rangle = \delta_{i,j}$, $\forall\ i, j \in \mathcal{D}$, where $\delta_{i,j} = 1$ when $i = j$ and $\delta_{i,j} = 0$ otherwise.

The framework of quantum computing resides in a state-space postulate which describes a state of a system by a unit vector in a Hilbert space. For example, a quantum computer of q qubits can represent a state of a system by a unit vector $|\psi\rangle$ in a $D = 2^q$ dimensional Hilbert space $\mathcal{H}$. Let $\mathcal{D} = \{|i\rangle\}_{i=0}^{D-1}$ be an orthonormal basis of $\mathcal{H}$. Every state $|\psi\rangle \in \mathcal{H}$ can be decomposed as

$$|\psi\rangle = \sum_{i=0}^{D-1} \phi_i |i\rangle , \tag{3}$$

where $\phi_0, \ldots, \phi_{D-1}$ is a set of coefficients with $\phi_i = \langle i|\psi\rangle$ and $\sum_{i=0}^{D-1} |\phi_i|^2 = 1$. Another salient feature of quantum computing is the measurement of a quantum state yields a probabilistic outcome rather than a deterministic one. When $|\psi\rangle$ in (3) is measured, it collapses to a random state in $\mathcal{D}$. In addition, the probability we observe $|i\rangle$ is $|\phi_i|^2$ for $i = 0, \ldots, D-1$.

B. Grover's algorithm and its limitations

Suppose we want to search a unique solution state $|k\rangle$ for some $k \in \{0, \ldots, D-1\}$, say the smallest real number from a set of D real numbers, or a word from a dictionary of D words. In Algorithm 1 below, we summarize a seminal quantum search method named Grover's algorithm [22]. Grover's algorithm assumes that there exists an oracle function $S(\cdot)$, such that $S(|k\rangle) = 1$ and $S(|i\rangle) = 0$ for $i \neq k$. Grover's algorithm is initialized with a superposition as the equally weighted average of an orthonormal basis $\mathcal{D} = \{|i\rangle\}_{i=0}^{D-1}$. To be specific, the initial superposition is defined as

$$|\psi_0\rangle = \frac{1}{\sqrt{D}} \sum_{i=0}^{D-1} |i\rangle \equiv c_0 |k\rangle + d_0 \sum_{i \neq k} |i\rangle ,$$

where $c_0 = d_0 = \frac{1}{\sqrt{D}}$. Let θ be the angle that satisfies $\sin^2 \theta = \frac{1}{D}$. After the j-th iteration, the coefficients are updated to c_j and d_j which admit a closed form [14],

$$\begin{cases} c_j = \sin\big((2j+1)\theta\big), \\ d_j = \frac{1}{\sqrt{D-1}} \cos\big((2j+1)\theta\big). \end{cases}$$

The closed form provides an intuitive geometric interpretation of Grover's algorithm. Let $|\zeta\rangle = \frac{1}{\sqrt{D-1}} \sum_{i \neq k} |i\rangle$ be the average of all non-solution states which is orthogonal to the solution state $|k\rangle$. In the j-th iteration, the operator F mirrors $|\psi_{j-1}\rangle$ with respect to $|\zeta\rangle$ and the operator G mirrors $F|\psi_{j-1}\rangle$ with respect to $|\psi_0\rangle$. The right panel of Fig. 1 provides a visualization of the two steps in a Grover's operation. Thus, each iteration in Grover's algorithm is equivalent to rotating the superposition $|\psi_j\rangle$ towards the solution state $|k\rangle$ by 2θ. When D is large, i.e., θ is small, we can approximate the angle by $\theta \approx \sin\theta = \frac{1}{\sqrt{D}}$. Then, a natural stopping criterion for Grover's algorithm is to choose the number of iterations τ by $(2\tau+1)/\sqrt{D} = \pi/2$, which yields τ is approximately $\lceil \sqrt{D}\pi/4 \rceil$, where $\lceil \cdot \rceil$ is the ceiling function.

Grover's algorithm is an oracular quantum algorithm as it depends on an oracle evaluation function that maps the solution state to 1 and all other states to 0. However, such a piece of oracle information is usually not available in machine learning problems as the states are measured over random samples. When we have partial or inaccurate oracle information of the solution state, say we may only identify the solution state up to a subset of states, i.e. $|k\rangle \in \mathcal{M} \subset \{|0\rangle, \ldots, |D-1\rangle\}$, the best oracle evaluation function that we can construct is

$$\begin{cases} S(|i\rangle) = 1, & \text{when } i \in \mathcal{M}, \\ S(|i\rangle) = 0, & \text{when } i \in \mathcal{M}^c. \end{cases}$$

Then, each iteration of Grover's algorithm rotates the current superposition towards the hyperplane spanned by the states in $\mathcal{M}$ instead of the true solution state $|k\rangle$. As a result, Grover's algorithm creates a biased estimator and fails to converge. The bias is lower bounded by the difference between $|k\rangle$ and the projection of the initial superposition on the hyperplane spanned by the states in $\mathcal{M}$. When we have no oracle information of the solution state at all, the oracle evaluation function can only be constructed by a randomly selected solution state. Then, Grover's algorithm is highly likely to rotate the initial superposition in the wrong direction and the output of the algorithm can be as bad as a random guess. We empirically demonstrate this phenomenon in Section V-D.

Algorithm 1 Grover's algorithm [22]

Input: A set $\mathcal{D} = \{|i\rangle\}_{i=0}^{D-1}$ with $D = 2^q$; a binary evaluation function S associated with the oracle state $|k\rangle$, such that $S(|k\rangle) = 1$ and $S(|i\rangle) = 0$ for $i \neq k$; number of iterations $\tau = \lceil \pi\sqrt{D}/4 \rceil$.
Initialization: Prepare a superposition $|\psi_0\rangle = \frac{1}{\sqrt{D}} \sum_{i=0}^{D-1} |i\rangle$ on a quantum register of p-qubits.
for $j = 1, \ldots, \tau$; **do**
  Grover's operation: Let $|\psi_j\rangle = GF|\psi_{j-1}\rangle$, where $F|i\rangle = (1 - 2S(|i\rangle))|i\rangle$, $G = 2|\psi_0\rangle\langle\psi_0| - \mathbf{I}_D$, and $\mathbf{I}_D$ is a $D \times D$ identity matrix.
end for
Output: Measure the latest superposition $|\psi_\tau\rangle$ on the quantum register.

C. Non-oracular quantum search algorithm

To solve the linearly independent column selection problem on a quantum computer, we first encode all subsets of $\{1, \ldots, q\}$ as quantum states in an orthonormal basis of $\mathcal{H}$, i.e. $\mathcal{D} = \{|i\rangle\}_{i=0}^{D-1}$ with $D = 2^q$. Further, we define a state loss function $g(\cdot) : \mathcal{H} \to \mathbb{R}$ as follows

$$g(|i\rangle) \equiv \widehat{L}_n(s; j_1, \ldots, j_s),$$

where the state $|i\rangle \in \mathcal{D}$ is a vector in $\mathcal{H}$ that corresponds to the subset $\{j_1, \ldots, j_s\}$, and $\widehat{L}_n$ is the minimum of (2) with respect to fixed s and $\{j_1, \ldots, j_s\}$. Intuitively, we want to find the best subset of columns corresponding to the quantum state that minimizes the state loss function.

We propose to establish a non-oracular quantum search (NQS) algorithm which consists of the following three steps.

(1) INITIALIZATION: We randomly choose an initial benchmark state $|w\rangle$ from $\mathcal{D} = \{|i\rangle\}_{i=0}^{D-1}$. Also, we pre-specify a learning rate $\lambda \in (0,1)$ with a recommendation $\lambda = 0.5$.
(2) UPDATING: We run Algorithm 1 over $\mathcal{D}$ by inputting $|w\rangle$ as the oracle state and $\tau = \lceil \pi\lambda^{-m/2}/4 \rceil$ as the number of iterations, where m is a positive integer. Denote the output of Algorithm 1 as $|w^{new}\rangle$ which is a state in $\mathcal{D}$. Then, we compare $|w^{new}\rangle$ with $|w\rangle$ in terms of the state loss function $g(\cdot)$. If $g(|w^{new}\rangle) < g(|w\rangle)$, we update the current benchmark state $|w\rangle$ to $|w^{new}\rangle$, otherwise we do not update $|w\rangle$.
(3) ITERATION AND OUTPUT: Start with $m = 1$ and repeat the updating step. After each updating step, set $m = m + 1$. NQS stops when $m > C(\lambda) \ln D$, where $C(\lambda)$ is a positive constant that depends on the learning rate λ. In our numerical experiments, we set $C(\lambda) = -6 \log_\lambda 10$, which works well. Then, we measure the quantum register with the latest superposition. The output is the observed state in $\mathcal{D}$ and its corresponding subset of columns.

Unlike Grover's algorithm, NQS randomly selects a state in $\mathcal{D}$ as the benchmark state which does not require any oracle information about the solution. Then, NQS iteratively updates the benchmark state towards the direction that reduces the state loss function. Besides, NQS is data-adaptive in the sense that it starts with a conservative learning step size (e.g. $m = 1$) and gradually increases the learning step size as the benchmark state has been updated towards the truth. We summarize NQS in Algorithm 2 below. The selection of λ and the sensitivity analysis will be discussed in Section V-C.

Algorithm 2 Non-oracular quantum search (NQS)

Input: An orthonormal basis $\mathcal{D} = \{|i\rangle\}_{i=0}^{D-1}$ of size $D = 2^q$, a state loss function $g(\cdot)$ that maps a state in $\mathcal{D}$ to a real number, and a learning rate $\lambda \in (0,1)$.
Initialization Set $m = 1$. Randomly select a state in $\mathcal{D}$ as the initial benchmark state $|w\rangle$. Define a local evaluation function $S(\,\cdot\,, |w\rangle, g)$ such that $S(|i\rangle, |w\rangle, g) = 1$ if $g(|i\rangle) \le g(|w\rangle)$ and $S(|i\rangle, |w\rangle, g) = 0$ if $g(|i\rangle) > g(|w\rangle)$.
repeat
(1) Run Algorithm 1 by inputting $\mathcal{D}$, $S(\,\cdot\,, |w\rangle, g)$ and $\tau(m) = \lceil \pi\lambda^{-m/2}/4 \rceil$.
(2) Measure the quantum register and denote the readout by $|w^{new}\rangle$.
(3) If $g(|w^{new}\rangle) < g(|w\rangle)$, set $|w\rangle = |w^{new}\rangle$ and update $S(\,\cdot\,, |w\rangle, g)$ accordingly.
(4) $m = m + 1$.
until $m > C(\lambda) \ln D$, where $C(\lambda)$ is a positive constant depends on λ.
Output: The latest benchmark state $|w\rangle$.

D. Intuition of non-oracular quantum search

Next, we discuss the intuition of NQS. Suppose that we implement NQS on a set $\mathcal{D} = \{|i\rangle\}_{i=0}^{D-1}$ with a pre-specified state loss function $g(\cdot)$ and a learning rate $\lambda \in (0,1)$. We assume there is a sole solution state in $\mathcal{D}$ that minimizes $g(\cdot)$. Without loss of generality, we number the states in $\mathcal{D}$ as the ascending rank of their state loss function values, i.e.

$$g(|0\rangle) < g(|1\rangle) \le g(|2\rangle) \le \cdots \le g(|D-1\rangle), \tag{4}$$

where $|0\rangle$ is the sole solution state.

If the initialization step luckily selects the sole solution state $|0\rangle$ as the initial benchmark state, NQS will never update the benchmark state and hence reduces to Grover's algorithm with a known oracle state and $\tau \approx \sqrt{D}\pi/4$. Therefore, NQS can recover the true state with a high success probability. A more interesting discussion would be considering the initial benchmark state $|w\rangle$ does not coincide with the truth, i.e. $|w\rangle \neq |0\rangle$.

Given the rank in (4), the local evaluation function $S(|i\rangle, |w\rangle, g)$ can be simplified as

$$\begin{cases} S(|i\rangle, |w\rangle, g) = 1, & \text{if } i \le w, \\ S(|i\rangle, |w\rangle, g) = 0, & \text{if } i > w. \end{cases}$$

In the m-th iteration, NQS calls Algorithm 1 by inputting $\mathcal{D}$, $S(\,\cdot\,, |w\rangle, g)$ (suppose that $w \neq 0$) and $\tau(m) = \lceil \pi\lambda^{-m/2}/4 \rceil$. Algorithm 1 initializes an equally weighted superposition as

$$|\psi_0\rangle = \frac{1}{\sqrt{D}} \sum_{i=0}^{D-1} |i\rangle \equiv \alpha_0 \sum_{i=0}^{w} |i\rangle + \beta_0 \sum_{j=w+1}^{D-1} |j\rangle,$$

with $\alpha_0 = \beta_0 = \frac{1}{\sqrt{D}}$.

Then, Algorithm 1 applies $\tau(m)$ times of Grover's operation to $|\psi_0\rangle$ which updates $|\psi_0\rangle$ to $|\psi_{\tau(m)}\rangle$ with coefficients satisfying

$$\begin{cases} \alpha_{\tau(m)} = \frac{1}{\sqrt{w+1}} \sin\big((2\tau(m)+1)\theta\big), \\ \beta_{\tau(m)} = \frac{1}{\sqrt{D-w-1}} \cos\big((2\tau(m)+1)\theta\big), \end{cases}$$

where the angle θ satisfies $\sin^2\theta = (w+1)/D$. After the m-th iteration, Algorithm 1 outputs a random state $|w_{new}\rangle \in \mathcal{D}$ with the following probability mass function

$$P(|w_{new}\rangle = |i\rangle) = \begin{cases} \alpha_{\tau(m)}^2, & \text{if } i \le w, \\ \beta_{\tau(m)}^2, & \text{if } i > w. \end{cases}$$

Since the learning rate $\lambda \in (0,1)$, we have $\alpha_{\tau(m)}^2 > 1/D > \beta_{\tau(m)}^2$ for some positive m. After the m-th iteration, NQS amplifies the probability of drawing the states whose state loss function values are smaller or equal to $g(|w\rangle)$. Meanwhile, NQS suppresses the probability of drawing the states whose state loss function values are greater than $g(|w\rangle)$. Geometrically, NQS rotates the initial superposition $|\psi_0\rangle$ towards $\frac{1}{\sqrt{w+1}} \sum_{i=0}^{w} |i\rangle$, which is an average of the states that can reduce the state loss function from $|w\rangle$. If the output of the m-th iteration is $|w_{new}\rangle = |i\rangle$ for some $i \ge w$, NQS will not update $|w\rangle$. On the other hand, if $i < w$, NQS will update $|w\rangle$ with $|w_{new}\rangle$, which is equivalent to descending $|w\rangle$ to a state with a smaller state loss function value. In Fig. 2, we visually illustrate the mechanism of NQS when $q = 2$.

Intuitively, one can think of NQS as a "quantum elevator"

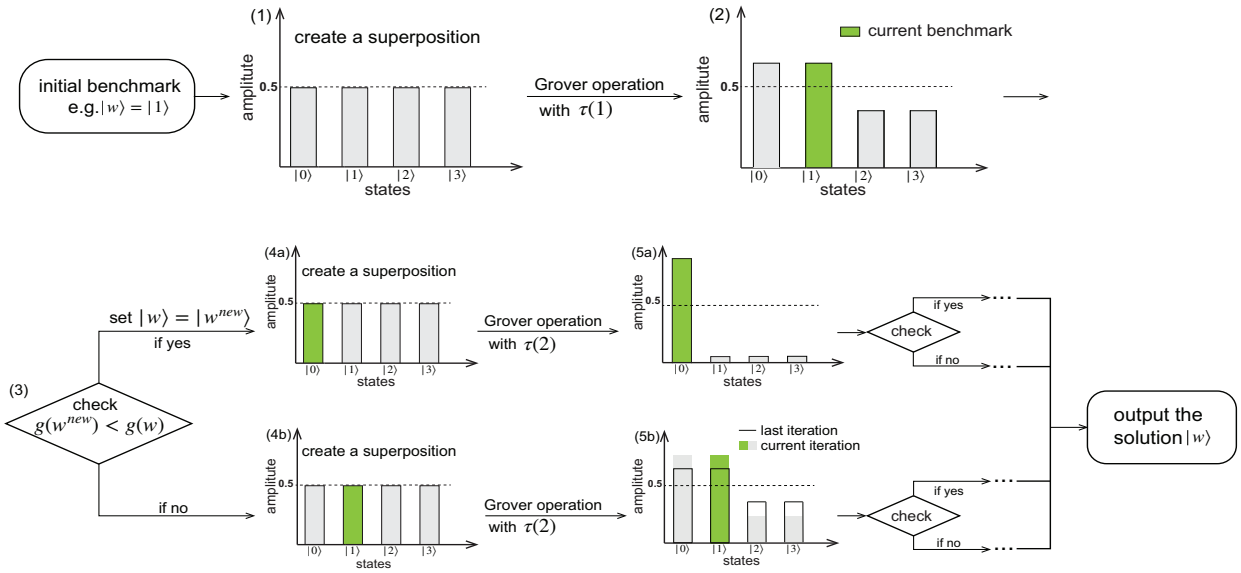

Fig. 2: An illustrative example ($q = 2$) for non-oracular quantum search.

that starts at a random floor of a high tower and aims to descent to the ground floor. Each operation, a quantum machine randomly decides if this elevator stays at the current floor or goes down to a random lower-level floor. The probability of going down will gradually increase as the number of operations increases. After a large enough amount of operations, it is not hard to imagine that the "quantum elevator" can descend to the ground floor with a high success probability.

IV. Theoretical results

This section presents the major theoretical results of the paper. Due to the space limitation, we defer some regularity conditions, technical lemmas, and detailed proofs to Appendix A.

First, we justify the intuition of NQS in Theorem 1 below. Theorem 1 states that within $O(\log_2 D)$ iterations, NQS finds the sole solution state with any arbitrary success probability greater than 50%.

Theorem 1. *Let $\kappa \in (0.5, 1)$ be a constant. With probability at least κ, NQS (e.g. Algorithm 2) finds the sole solution state within $C_\kappa \log_2 D$ iterations, where C_κ is a positive constant that depends on κ.*

Suppose that, in the m-th iteration of Algorithm 2, the current benchmark state is $|w_m\rangle$ with $g(|w_m\rangle)$ being the s_m-th smallest among $\{g(|i\rangle)\}_{i=0}^{D-1}$. Theorem 2 below shows the expected number of Grover's operations that Algorithm 2 needs to update $g(|w_m\rangle)$ is of order $O(\sqrt{D/s_m})$.

Theorem 2. *Let $|w_m\rangle$ be the current benchmark state in the m-th iteration of Algorithm 2. Let s_m be the rank of $g(|w_m\rangle)$ in the sorted sequence of $\{g(|i\rangle)\}_{i=0}^{D-1}$ in ascending order, $s_m = 1, \ldots, D$. The expected time for Algorithm 2 to update $|w_m\rangle$ is of order $O(\sqrt{D/s_m})$.*

Theorems 1 and 2 together imply the computational complexity upper bound for NQS is of order $O(\sqrt{D}\log_2 D)$, which is only a $\log_2 D$ factor larger than the theoretical lower bound for oracular quantum search algorithms [27]. Therefore, NQS achieves a near optimal computational efficiency for oracular quantum search algorithms without using any oracle information.

In Theorem 3 below, we establish a consistency property for the rank estimator and an oracle property for the coefficient matrix estimator $\widehat{\mathbf{A}} \equiv \widehat{\mathbf{A}}(\widehat{r}; \widehat{j}_1, \ldots, \widehat{j}_r)$, where $\widehat{r}$ and $\{\widehat{j}_1, \ldots, \widehat{j}_r\}$ are the rank and the linearly independent columns selected by NQS.

Theorem 3. *Suppose conditions 1–4 in Appendix A.1 hold. With a probability approaching 1, we have*

$$\widehat{r} = r \quad \textit{and} \quad \|\widehat{\mathbf{A}} - \mathbf{A}\|_{\mathrm{F}} = O_p(\sqrt{r(p+q-r)\log p/n}),$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm.

Theorem 3 implies that we can correctly identify the linearly independent columns in $\mathbf{A}$ with an overwhelming probability when n is large. The proposed coefficient matrix estimator is consistent and satisfies an oracle property since the convergence rate is in line with existing results [2] as if the rank and the location of linearly independent columns were known.

V. Numerical studies

In this section, we use several numerical experiments to assess the finite sample performance of the proposed learning procedure, which is denoted as Hybrid Quantum Estimation (HQE). To be specific, HQE uses Algorithm 2 to select linearly independent columns in the coefficient matrix and then estimate the coefficient matrix. We also compare HQE with five alternative methods: ES and FA are similar to HQE except that the linearly independent columns are selected by Exhaustive Search and Forward Adding, respectively; STRS

TABLE I: Simulation results over 200 replications with standard errors shown in parentheses.

ρ		0.1		0.5	
b		0.4	0.7	0.4	0.7
RANK	ES	3.055(0.268)	3.025(0.211)	3.260(0.577)	3.110(0.385)
	HQE	3.065(0.284)	3.025(0.211)	3.280(0.601)	3.115(0.390)
	FA	3.050(0.240)	3.020(0.140)	3.390(0.713)	3.170(0.501)
	STRS	5.465(0.929)	5.485(0.930)	5.550(0.944)	5.550(0.950)
	TR	14.39(0.932)	14.94(0.326)	14.85(0.497)	14.98(0.140)
	OLS	15.00(0.000)	15.00(0.000)	15.00(0.000)	15.00(0.000)
EST	ES	0.100(0.015)	0.096(0.013)	0.154(0.026)	0.147(0.024)
	HQE	0.101(0.015)	0.096(0.013)	0.155(0.026)	0.147(0.024)
	FA	0.101(0.015)	0.097(0.014)	0.143(0.012)	0.149(0.026)
	STRS	0.111(0.013)	0.111(0.013)	0.141(0.016)	0.141(0.016)
	TR	0.113(0.014)	0.148(0.024)	0.162(0.023)	0.238(0.040)
	OLS	0.156(0.006)	0.156(0.006)	0.199(0.008)	0.199(0.008)
PRED	ES	1.222(0.056)	1.206(0.049)	1.295(0.074)	1.271(0.070)
	HQE	1.223(0.058)	1.206(0.049)	1.295(0.074)	1.271(0.070)
	FA	1.225(0.060)	1.207(0.052)	1.304(0.082)	1.279(0.080)
	STRS	1.268(0.056)	1.269(0.056)	1.271(0.056)	1.271(0.056)
	TR	1.274(0.065)	1.441(0.128)	1.330(0.079)	1.604(0.163)
	OLS	1.482(0.036)	1.482(0.036)	1.482(0.036)	1.482(0.036)

stands for Self-Tuning Rank Selection, which is a low rank matrix decomposition approach [5]; TR stands for Trace-norm Regularization, which is a penalized matrix estimation approach [28]; and the Ordinary Least Squares estimator (OLS). Notice that ES is computationally infeasible in high dimension. Hence, we implement ES with oracle information that the rank of the coefficient matrix is upper bounded by twice the truth. To that end, we consider ES as an oracle upper bound and OLS as a naive lower bound for the numerical experiments. Replication codes and data will be released in a Github repository upon acceptance.

A. Implementation details

We implement the quantum experiments on IBM Quantum Experience (www.ibm.com/quantum-computing), a publicly available cloud-based quantum computing system. This platform has developed a Qiskit Python development kit (https://qiskit.org/), which allows users to perform both quantum computing and classical computing in a single project. According to the rule of thumb recommendations in Section V-C, we choose $\lambda = 0.5$ and $C(\lambda) = -6\log_\lambda 10$ in Algorithm 2 throughout the paper (as the proposed algorithm is not sensitive to these choices).

B. Simulation experiments

Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ and $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)^\top$ be a random sample that follows the multi-response linear model (1). We generate $\mathbf{x}_i \sim N_p(\mathbf{0}, \mathbf{\Sigma})$ with $\mathbf{\Sigma} = (\sigma_{jk})_{j,k=1}^p$ and $\sigma_{jk} = \rho^{|j-k|}$ for some $\rho \in [0, 1)$. The errors are drawn from i.i.d. $N(0, 1)$. The coefficient matrix satisfies $\mathbf{A} = b\mathbf{\Gamma}_0\mathbf{\Gamma}_1$, with $b > 0$, $\mathbf{\Gamma}_0 \in \mathbb{R}^{p\times r}$, $\mathbf{\Gamma}_1 \in \mathbb{R}^{r\times q}$ and $r \le \min(p,\ q)$. The entries of $\mathbf{\Gamma}_0$ and $\mathbf{\Gamma}_1$ are independently drawn from $N(0, 1)$. The rank r and parameter b together control the signal to noise ratio. Further, we independently generate $\{\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}\}$ as a testing sample of size n_t. We set the training and testing sample sizes as $n = 100$ and $n_t = 200$, and the dimensions as $p = 50$, $q = 15$ and $r = 3$. Besides, we choose $\rho = 0.1, 0.5$ and $b = 0.4, 0.7$. For each setting, we simulate 200 replications.

TABLE II: Mean (SD) of selection accuracy over 100 replications.

$\lvert\mathcal{D}\rvert$	NQS	Grover
256	0.998(0.004)	0.012(0.011)
512	0.997(0.006)	0.008(0.006)
1024	0.998(0.005)	0.004(0.006)

The numerical performance is assessed by the following three criteria: (1) estimated rank (RANK) $\hat{r}$; (2) scaled estimation error (EST) $\|\widehat{\mathbf{A}} - \mathbf{A}\|_{\mathrm{F}}/\sqrt{pq}$; and (3) scaled prediction error (PRED) $\|\widetilde{\mathbf{Y}} - \widetilde{\mathbf{X}}\widehat{\mathbf{A}}\|_{\mathrm{F}}/\sqrt{n_t q}$. The simulation results are presented in Table I. It can be seen that HQE performs almost identically to the oracle method ES in all scenarios, which implies the proposed non-oracular quantum search is a successful quantum alternative for exhaustive search. FA and STRS perform less promising than HQE and ES. OLS performs worst in all scenarios as it completely ignores the low rank structure.

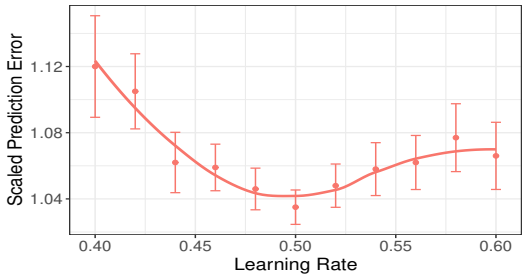

Fig. 3: Scaled prediction error versus the learning rate λ. The mean and standard errors of scaled prediction error over 200 replications are plotted as red dots and vertical error bars, respectively. The solid line is a smoothed curve.

C. Sensitivity analysis for learning rate

The proposed non-oracular quantum search algorithm involves a learning rate $\lambda \in (0, 1)$ which controls the "step size" of each iteration. In this subsection, we use the low dimensional setting introduced in Section V-B to assess the sensitivity of Algorithm 2 with respect to the choice of λ. To be specific, we let λ be a sequence of grid points between 0.40 and 0.60 with a step size of 0.02. For each λ, we simulate the low dimensional setting for 200 replications. The scaled prediction error versus the values of λ is reported in Fig. 3. We observe that Algorithm 2 is not very sensitive to the choice of λ. Further, the smallest prediction error is attained when λ is around 0.5, which aligns with our rule-of-thumb recommendation.

D. NQS versus Grover's algorithm

We provide an empirical comparison between NQS and Grover's algorithm in Table II to select the smallest number in a random set $\mathcal{D}$ whose elements are i.i.d. drawn from $U[0, 1]$. For all three sets of cardinalities, NQS achieves near-perfect results while Grover behaves like random guesses. This observation, which is in line with the discussions in Section III-B, shows that oracular quantum search algorithms

cannot be effectively applied to statistical and machine learning problems.

E. Real data example

Particulate matter up to $2.5\mu m$ (PM2.5) is a complex mixture of solid particles, chemicals (e.g., sulfates, nitrates), and liquid droplets in the air, which include inhalable particles that are small enough to penetrate the thoracic region of the respiratory system. Long term exposure to PM2.5 may lead to an increase in hospital admissions related to respiratory and cardiovascular morbidity, such as aggravation of asthma, respiratory symptoms, and cardiovascular disorders [29]–[33].

We investigate the relationship between the concentration of PM2.5 and four air pollutants: ozone (O_3), sulfur dioxide (SO_2), carbon monoxide (CO), and nitrogen dioxide (NO_2). The dataset consists of 728 daily observations between January 2017 and April 2019 collected from 15 outdoor monitoring sites across the United States. We have taken the first-order difference for each variable to remove the non-stationarity. The original dataset is publicly available at https://www.epa.gov/outdoor-air-quality-data.

In the experiment, we compare the prediction performance of our method (HQE) with the other five competitors. We use the first 600 observations as the training set and the rest 128 observations as the testing set. Scaled prediction errors and estimated ranks are presented in Table III. According to Table III, ES and HQE perform identically and outperform the other methods. FA, STRS and TR suffer from high prediction errors as they overestimate the rank. As expected, OLS performs the worst since it ignores the low rank structure. Further, we visualize the site level association between the concentrations of PM2.5 and the other four air pollutants at the 15 outdoor monitoring sites in Fig. 4. The plots in Fig. 4 show clear geographical clustering structures, which may lead to some interesting scientific findings.

TABLE III: Results for the real data example.

	PRED	RANK
ES	0.925	1
HQE	0.925	1
FA	0.932	2
STRS	0.942	4
TR	0.927	15
OLS	0.966	15

VI. CONCLUDING REMARKS

Multi-response datasets are commonly encountered in machine learning problems. Existing literature [1], [5] proposes to estimate the coefficient matrix by low rank decompositions. However, when the low rank assumption is not valid, the decomposition scheme can create more unknown parameters than estimating the original matrix directly. To address this challenge, we propose to estimate the coefficient matrix by selecting linearly independent columns, which provides a mathematically elegant solution. A novel non-oracular quantum search algorithm tackles the computationally challenging column selection problem. Theoretical justifications and numerical studies support the advantages of the proposed estimation approach.

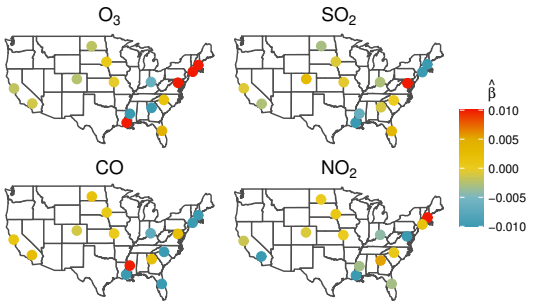

Fig. 4: Associations between PM2.5 and the other four air pollutants at the 15 outdoor monitoring sites in the United States.

APPENDIX

A. Proofs for theoretical results in Section IV

The appendix provides regularity conditions, technical lemmas and detailed proofs for the theoretical results in Section IV.

A.1 Regularity conditions and tchnical lemmas

First, we impose the regularity conditions which are required to establish the asymptotic properties in theorems.

Condition 1. *There exist positive constants* c_1 *and* c_2 *such that with probability one* $c_1 \leq \lambda_{\min}(n^{-1}\mathbf{X}^\top\mathbf{X}) < \lambda_{\max}(n^{-1}\mathbf{X}^\top\mathbf{X}) \leq c_2$.

Condition 2. *For some positive constants* c_3 *and* c_4, $\mathbb{E}\{\exp(c_3\epsilon_j^2)\} < c_4$ *for* $j = 1, \ldots, q$.

Condition 3. *The elements of* $\mathbf{A}$ *and* $\mathbf{XA}$ *are bounded. The matrix* $\mathbf{A}$ *is* Ω_n*-sparse in the sense that* $\max_{1\leq\ell\leq q}\sum_{j=1}^{p}\mathbb{I}(a_{j\ell}\neq 0) \leq \Omega_n$, *where* $\mathbb{I}(\cdot)$ *is an indicator function.*

Condition 4. *We assume* $r \leq q \leq p$ *and* $rp\log p/n \to 0$ *when* $n \to \infty$.

Condition 1 is a restricted eigenvalue condition, which is widely used in the literature, e.g., [34]. It requires the the sample matrix of covariates to behave reasonably well. Condition 2 requires the errors to be sub-Gaussian, and hence their tail probabilities decay exponentially. Condition 3 facilitates our derivation but can be much relaxed so that the true signal strength depends on n as well. Condition 4 imposes a requirement on the diverging rates of p, q and r in order to obtain consistent estimations when $n \to \infty$.

Next, we provide two technical lemmas to pave the way for the proof of main theorems. We omit the proof of Lemma 1 as it can be found in the literature.

Lemma 1 (c.f. Lemma 1 in [23])**.** *For any real numbers α and β and any positive integer m, we have*

$$\sum_{j=0}^{m-1} \cos(\alpha + 2\beta j) = \frac{\sin(m\beta)\cos(\alpha + (m-1)\beta)}{\sin\beta}.$$

In particular, when $\alpha = \beta$, the above equality can be simplified as

$$\sum_{j=0}^{m-1} \cos(\alpha + 2\beta j) = \frac{\sin(2m\alpha)}{2\sin\alpha}.$$

Lemma 2. *Let s_0 be the (unknown) number of solutions states among D states. Let θ be such that $\sin^2\theta = s_0/D$. Let γ be an arbitrary positive integer. Let j be an integer sampled uniformly between 0 and $\tau - 1$. If we observe the register after applying j Grover's operations to the initial state $|\psi_0\rangle = \sum_{i=0}^{D-1} \frac{1}{\sqrt{D}}|i\rangle$, the probability of obtaining a solution is exactly*

$$P_\tau = \frac{1}{2} - \frac{\sin(4\tau\theta)}{4\tau\sin(2\theta)}.$$

Further, we have $P_\tau \geq \frac{1}{4}$ when $\tau \geq \frac{1}{\sin(2\theta)}$.

Proof of Lemma 2. The probability of finding one solution state among s_0 if we perform j Grover's operations is $s_0 k_j^2 = \sin^2((2j+1)\theta)$. When we choose $0 \leq j < \tau$ randomly, the average success probability follows

$$\begin{aligned} P_\tau &= \sum_{j=0}^{\tau-1} \frac{1}{\tau}\sin^2((2j+1)\theta) \\ &= \frac{1}{2\tau}\sum_{j=0}^{\tau-1}\{1 - \cos((2j+1)2\theta)\} \\ &= \frac{1}{2} - \frac{\sin(4\tau\theta)}{4\tau\sin 2\theta}. \end{aligned}$$

If $\tau \geq \frac{1}{\sin(2\theta)}$, we complete the proof as the following inequality holds

$$\frac{\sin(4\tau\theta)}{4\tau\sin 2\theta} \leq \frac{1}{4\tau\sin 2\theta} \leq \frac{1}{4}.$$

□

A.2 Proof of Theorem 1

For the ease of presentation and without loss of generality, we re-number the states in descending order in this proof, i.e.

$$g(|0\rangle) > g(|1\rangle) \geq g(|2\rangle) \geq \cdots \geq g(|D-1\rangle) > g(|D\rangle),$$

where $|D\rangle$ is the unique solution state and $|0\rangle$ is an added initial state to facilitate the discussion with $g(|0\rangle)$ being an arbitrary large value.

Let us assume Algorithm 2 is initialized with the least favorable state $|0\rangle$. Let Z denote the number of iterations the algorithm takes to arrive at the solution state $|D\rangle$. The rule that Algorithm 2 moves from $|0\rangle$ to $|D\rangle$ can be abstracted as the following mathematical process.

ITERATION 1: Draw an integer X_1 uniformly from 0 to D, then the algorithm moves from $|0\rangle$ to $|X_1\rangle$.

ITERATION 2: Draw an integer X_2 uniformly from 0 to $D - X_1$, then the algorithm moves from $|X_1\rangle$ to $|X_1 + X_2\rangle$.

$\vdots$

ITERATION z: Draw an integer X_z uniformly from 0 to $D - \sum_{i=1}^{z-1} X_i$, then the algorithm moves from $|\sum_{i=1}^{z-1} X_i\rangle$ to $|\sum_{i=1}^{z} X_i\rangle$. If $\sum_{i=1}^{z} X_i = D$, then the algorithms stops as at $Z = z$ as it finds the solution state $|D\rangle$. Otherwise, the algorithm goes to the $(z+1)$th iteration.

As we can see, the total number of iterations Z is a discrete random variable that can take any positive integer values. To prove Theorem 1, it is equivalent to show that the κ-th quantile of the discrete random variable Z is upper bounded by $C_\kappa \ln D$ for a positive constant C_κ. The proof will be unveiled by three steps.

Step 1: A partial sum process.

To investigate the probability distribution of Z, we first study the partial sum of X_i. Define a partial sum process

$$S_z = \sum_{i=1}^{z} X_i \text{ for } z = 1, 2, \ldots.$$

Notice that $(S_z | S_{z-1} = s_{z-1}, \ldots, S_1 = s_1) \overset{\mathcal{D}}{=} (S_z | S_{z-1} = s_{z-1}) \sim \text{unif}\{s_{z-1}, D\}$. Also, we can show $\{S_z\}_{z=1,2,\ldots}$ is a submartingale, i.e. $\mathbb{E}[S_z|S_1, \ldots, S_{z-1}] = \mathbb{E}[S_z|S_{z-1}] \geq S_{z-1}$. Thus, $\{S_z\}_{z=1,2,\ldots}$ is a discrete-time Markov chain with a finite state space $\{0, 1, 2, \ldots, D\}$.

Moreover, the expectation of S_z satisfies

$$\begin{aligned} \mathbb{E}[S_z] &= \mathbb{E}\left[\mathbb{E}[S_z|S_{z-1}]\right] = \mathbb{E}\left[\frac{S_{z-1} + D + 1}{2}\right] \\ &= \mathbb{E}[S_{z-1}]/2 + (D+1)/2 \\ &= \mathbb{E}[X_1]/2^{z-1} + (D+1)\big(1 - 1/2^{z-1}\big) \\ &= (D+1)/2^z + +(D+1)\big(1 - 1/2^{z-1}\big) \\ &= (D+1)\big(1 - 1/2^z\big). \end{aligned}$$

Step 2: The probability mass function of Z.

Next, we derive the probability mass function of Z. The derivation is based on the Markov property of the partial sum process $\{S_z\}_{z=1,2,\ \ldots}$. Define a transition matrix P as

$$P = \begin{pmatrix} \frac{1}{D+1} & \frac{1}{D+1} & \frac{1}{D+1} & \cdots & \frac{1}{D+1} & \frac{1}{D+1} \\ 0 & \frac{1}{D} & \frac{1}{D} & \cdots & \frac{1}{D} & \frac{1}{D} \\ 0 & 0 & \frac{1}{D-1} & \cdots & \frac{1}{D-1} & \frac{1}{D-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Let π_z be a row vector with dimension $D+1$ that denotes the distribution of the chain $\{S_z\}_{z=1,2,\ldots}$ at time z. By the definition of $\{S_z\}_{z=1,2,\ \ldots}$ and P, we have

$$\pi_{z+1} = \pi_z P, \quad \text{for} \quad z = 1, 2, \ldots.$$

Hence, we can write out π_1, π_2 and π_3 as

$$\pi_1 = \Big(\frac{1}{D+1}, \frac{1}{D+1}, \frac{1}{D+1}, \ldots, \frac{1}{D+1}, \frac{1}{D+1}\Big),$$

$$\pi_2 = \frac{1}{D+1}\Big(\frac{1}{D+1}, \sum_{i_1=D}^{D+1}\frac{1}{i_1}, \ \ldots\ , \sum_{i_1=1}^{D+1}\frac{1}{i_1}\Big), \quad \text{and}$$

$$\pi_3 = \frac{1}{D+1}\Big(\frac{1}{(D+1)^2}, \sum_{i_2=D}^{D+1}\frac{1}{i_2}\sum_{i_1=i_2}^{D+1}\frac{1}{i_1}, \ldots, \sum_{i_2=1}^{D+1}\frac{1}{i_2}\sum_{i_1=i_2}^{D+1}\frac{1}{i_1}\Big).$$

In general, we can summarize the expression of π_z for $z = 1, 2, \ldots$ as

$$\pi_z = \Big(\mathbb{P}(S_z = 0), \ldots, \mathbb{P}(S_z = D)\Big),$$

where for $j = 0, 1, \ldots, D$,

$$\mathbb{P}(S_z = j) = \frac{1}{D+1}\sum_{i_{z-1}=D+1-j}^{D+1}\frac{1}{i_{z-1}}\sum_{i_{z-2}=i_{z-1}}^{D+1}\frac{1}{i_{z-2}}\cdots\sum_{i_1=i_2}^{D+1}\frac{1}{i_1},$$

Therefore, we can write out the probability mass function of $Z = z$ for $z = 1, 2, \ldots$ as

$$\mathbb{P}(Z = z) = \mathbb{P}\big(S_z = D\big) = \frac{1}{D+1}\sum_{i_{z-1}=1}^{D+1}\frac{1}{i_{z-1}}\cdots\sum_{i_1=i_2}^{D+1}\frac{1}{i_1}. \tag{5}$$

Notice that the last summation in (5), i.e. $\sum_{i_1=i_2}^{D+1}\frac{1}{i_1}$, is a finite partial sum of a harmonic series. Hence, we can write it as

$$\sum_{i_1=i_2}^{D+1}\frac{1}{i_1} = \ln(D+1) + \eta_{D+1} - \ln(i_2) - \eta_{i_2} \asymp \ln(D+1) - \ln(i_2), \text{ where } \eta_z \asymp \frac{1}{2z}.$$

Let us move on to the next level of summation in (5), which is upper bounded by

$$\sum_{i_2=i_3}^{D+1}\frac{1}{i_2}\sum_{i_1=i_2}^{D+1}\frac{1}{i_1} \asymp \sum_{i_2=i_3}^{D+1}\frac{1}{i_2}\big[\ln(D+1) - \ln(i_2)\big] \lesssim \ln(D+1)\big[\ln(D+1) - \ln(i_3)\big] \lesssim \ln^2(D+1),$$

and lower bounded by

$$\sum_{i_2=i_3}^{D+1}\frac{1}{i_2}\sum_{i_1=i_2}^{D+1}\frac{1}{i_1} \asymp \sum_{i_2=i_3}^{D+1}\frac{1}{i_2}\big[\ln(D+1) - \ln(i_2)\big] \asymp \ln(D+1)\big[\ln(D+1) - \ln(i_3)\big] - \sum_{i_2=i_3}^{D+1}\frac{\ln(i_2)}{i_2} \gtrsim \ln^2(D+1) - \ln(D+1)\ln(i_3),$$

since $\sum_{i_2=i_3}^{D+1}\frac{\ln(i_2)}{i_2} \asymp \frac{1}{2}\ln^2(D+1) - \frac{1}{2}\ln^2(i_3)$.

Similarly, we can go through all the summations in (5) and show

$$\mathbb{P}(Z = 0) = \frac{1}{D+1} \quad \text{and} \quad \mathbb{P}(Z = z) \asymp \frac{\ln^{z-1}(D+1)}{D+1}. \tag{6}$$

Step 3: The κ-th quantile of Z.

With the results in (6), we summarize the cumulative distribution function of Z as

$$F_Z(z) = \mathbb{P}(Z \le z) = \sum_{j=0}^{z}\mathbb{P}(Z = j) \asymp \frac{1}{D+1} + \sum_{j=1}^{z}\frac{\ln^{j-1}(D+1)}{D+1} \asymp \frac{\ln^{z-2}(D+1)}{D+1}.$$

Let $z = c\ln(D+1)$ with a positive constant $c \in (0, 1)$, we have

$$F_Z(z) \asymp \frac{\ln^{z-2}(D+1)}{D+1} \asymp \frac{(D+1)^{c\ln\ln(D+1)}}{D+1} \asymp 1.$$

Therefore, for any $\kappa \in (\frac{1}{2}, 1)$, there exists a positive constant C_κ such that the κ-th quantile of Z is upper bounded by $C_\kappa \log_2 D$, which completes the proof.

□

A.3 Proof of Theorem 2

Suppose we are in the m-th iteration of Algorithm 2. Let $|w_m\rangle$ be the current benchmark state and s_m be the rank of $g(|w_m\rangle)$ in the sorted sequence of $\{g|i\rangle\}_{i=0}^{D-1}$ in ascending order, $s_m = 1, \ldots, D$. Notice that Algorithm 2 implements $\tau(m) = \lceil \pi\lambda^{-m/2}/4 \rceil \equiv \lceil \frac{\pi}{4}\gamma^m \rceil$ Grover's operations, where $\gamma = \lambda^{-1/2}$. We are interested in finding the expected number of Grover's operations to update $|w_m\rangle$.

Let θ be the angle such that $\sin^2\theta = s_m/D$. Let

$$\tau_m^* = \frac{1}{\sin(2\theta)} = \frac{D}{2\sqrt{(D - s_m)s_m}} < \sqrt{\frac{D}{s_m}}.$$

We say that the algorithm reaches a phase transition for $|w_m\rangle$ if $\tau(s)$ exceeds $\tau^*(m)$ for some $s \ge m$.

The expected total number of Grover's operations needed to reach the phase transition for $|w_m\rangle$ is upper bounded by

$$\frac{\pi}{4}\sum_{j=0}^{\lceil \log_\gamma(\tau_m^*)\rceil}\gamma^j < \frac{\pi}{4}\frac{\gamma^m - 1}{\gamma - 1} < \frac{\tau_m^* + 1}{\gamma - 1}.$$

Thus, if the algorithm updates $|w_m\rangle$ before it reaches the phase transition, the expected number of Grover's operations is at most of order $O(\tau_m^*)$, which is further upper bounded by $O(\sqrt{D/s_m})$.

If the phase transition for $|w_m\rangle$ is reached, as proved by Lemma 2, every new iteration of Algorithm 2 will be able to update $|w_m\rangle$ with a probability at least 1/4 since $\tau(s) \ge \tau_m^*$. Then, the expected iterations to update $|w_m\rangle$ after the phase transition is upper bounded by a positive constant. Hence, the expected number of additional Grover's operations needed to update $|w_m\rangle$ after the phase transition is upper bounded by $C\tau_m^*$ for some positive constant C. These two scenarios together complete the proof.

□

A.4 Proof of Theorem 3

Let r be the the rank of $\mathbf{A}$ and $\mathcal{J} = \{j_1, \ldots, j_r\}$ be the indices set of linearly independent columns of $\mathbf{A}$. We denote $\widehat{r}$ and $\widehat{\mathcal{J}} = \{\hat{j}_1, \ldots, \hat{j}_{\hat{r}}\}$ the estimators of r and $\mathcal{J}$ obtained from Algorithm 2. Theorem 1 has shown that NQS can correctly identify the number and locations of linearly independent columns of $\mathbf{A}$ with any arbitrarily high probability. In this proof, we prove the second result in Theorem 3 over the event that $\mathcal{E} = \{\widehat{r} = r, \widehat{\mathcal{J}} = \mathcal{J}\}$ whose exception probability, i.e. $\Pr(\mathcal{E}^c)$, is negligible when n diverges.

Without loss of generality, we can column-wise permute $\mathbf{A}$ such that

$$\mathbf{A} = (\mathbf{A}_1,\ \mathbf{A}_2) \equiv (\mathbf{A}_1,\ \mathbf{A}_1\mathbf{B}),$$

where $\mathbf{A}_1 \in \mathbb{R}^{p\times r}$ has the r linearly independent columns of $\mathbf{A}$, $\mathbf{A}_2 \in \mathbb{R}^{p\times(q-r)}$ has the rest $q-r$ columns, and $\mathbf{B} \in \mathbb{R}^{r\times(q-r)}$ is a coefficient matrix such that $\mathbf{A}_2 = \mathbf{A}_1\mathbf{B}$. With a bit of violation of notations, we denote the matrices of responses, covariates and errors by their columns, i.e., $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_q)$, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)$ and $\mathbf{E} = (\mathbf{e}_1, \ldots, \mathbf{e}_q)$ in the proof. Then, we follow the column-wise permute of $\mathbf{A}$ and represent $\mathbf{Y}$ and $\mathbf{E}$ as

$$\mathbf{Y} = (\mathbf{Y}_1,\ \mathbf{Y}_2) \quad \text{and} \quad \mathbf{E} = (\mathbf{E}_1,\ \mathbf{E}_2).$$

Also, we can represent the loss function (2) as

$$L_n(\mathcal{J}, \mathbf{A}_1, \mathbf{B}) = G_n(\mathbf{A}_1) + H_n(\mathbf{A}_1, \mathbf{B}),$$

where

$$G_n(\mathbf{A}_1) = \|\mathbf{Y}_1 - \mathbf{X}\mathbf{A}_1\|_{\mathrm{F}}^2 \text{ and } H_n(\mathbf{A}_1, \mathbf{B}) = \|\mathbf{Y}_2 - \mathbf{X}\mathbf{A}_1\mathbf{B}\|_{\mathrm{F}}^2.$$

Given a full rank matrix $\mathbf{A}_1$, by minimizing the function $H_n(\mathbf{A}_1, \mathbf{B})$ with respect to $\mathbf{B}$, we obtain that

$$H_n(\mathbf{A}_1) \equiv H_n(\mathbf{A}_1, \widehat{\mathbf{B}}) = \mathrm{tr}\left\{\mathbf{Y}_2^\top(\mathbf{I} - \mathbf{H}_{\mathbf{Z}})\mathbf{Y}_2\right\},$$

where $\mathbf{H}_{\mathbf{Z}} = \mathbf{Z}(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top$ and $\mathbf{Z} = n^{-1/2}\mathbf{X}\mathbf{A}_1$. Further, for the ease of presentation, we denote

$$L_n(\mathbf{A}_1) \equiv L_n(\mathcal{J}, \mathbf{A}_1, \widehat{\mathbf{B}}).$$

Let $\delta_n = \sqrt{r(p+q-r)\log p/n}$ and $C_1 > 0$ be a large constant. We aim to show that as $n \to \infty$,

$$\Pr\left\{\inf_{\mathbf{W}\in\mathbb{R}^{p\times r};\|\mathbf{W}\|_{\mathrm{F}}=C_1} L_n(\mathbf{A}_1 + \delta_n\mathbf{W}) < L_n(\mathbf{A}_1)\right\} \to 0. \quad (7)$$

The condition in (7) implies that there exists a local minimum in the ball $\{\mathbf{A}_1 + \delta_n\mathbf{W} : \|\mathbf{W}\|_{\mathrm{F}} \le C_1\}$ with probability tending to one. Hence, there exists a local minimizer of $L_n(\mathbf{A}_1)$ such that $\|\widehat{\mathbf{A}}_1 - \mathbf{A}_1\|_{\mathrm{F}} = O_p(\delta_n)$.

First, we have the following decomposition

$$\begin{aligned}
&L_n(\mathbf{A}_1 + \delta_n\mathbf{W}) - L_n(\mathbf{A}_1)\\
=&\{G_n(\mathbf{A}_1 + \delta_n\mathbf{W}) - G_n(\mathbf{A}_1)\} + \{H_n(\mathbf{A}_1 + \delta_n\mathbf{W}) - H_n(\mathbf{A}_1)\}\\
\equiv&\Delta_1 + \Delta_2.
\end{aligned}$$

With some calculations, we can show that

$$\Delta_1 = n\delta_n^2\mathrm{tr}\{\mathbf{w}^\top(n^{-1}\mathbf{X}^\top\mathbf{X})\mathbf{w}\} - 2\delta_n\mathrm{tr}(\mathbf{E}_1^\top\mathbf{X}\mathbf{w}).$$

By Condition 1, we have $\mathrm{tr}\{\mathbf{w}^\top(n^{-1}\mathbf{X}^\top\mathbf{X})\mathbf{w}\} \ge c_1\|\mathbf{w}\|_{\mathrm{F}}^2$. Then, the first term of Δ_1 is lower bounded by $C_1^2c_1n\delta_n^2$, which is quadratic in C_1.

For the second term of Δ_1, using Cauchy-Schwartz inequality, we have

$$2\delta_n\mathrm{tr}(\mathbf{E}_1^\top\mathbf{X}\mathbf{w}) \le 2\delta_nC\left(\sum_{\ell\in\mathcal{D}}\sum_{j=1}^{p}(\mathbf{x}_j^\top\mathbf{e}_\ell)^2\right)^{1/2}.$$

By Condition 2 and the exponential tail probability of sub-Gaussian variables, we have

$$pq\Pr\left(|\mathbf{x}_j^\top\mathbf{e}_\ell| > C_2\sqrt{n\log p}\right) \to 0 \quad \text{as} \quad n\to\infty$$

for a sufficiently large $C_2 > 0$. Thus, $\mathbf{x}_j^\top\mathbf{e}_\ell = O_p(\sqrt{n\log p})$ for all j and ℓ. Consequently, the second term of Δ_1 is of order $O_p(\sqrt{rnp\log p}\delta_nC_1)$, which is linear in C_1. Therefore, by the definition of δ_n, as long as the constant C_1 is sufficiently large, the first term of Δ_1 dominates the second term with an arbitrarily large probability.

Next, we study Δ_2. To simplify the presentation, we denote $\tilde{\mathbf{A}}_1 = \mathbf{A}_1 + \delta_n\mathbf{w}$, $\tilde{\mathbf{Z}} = n^{-1/2}\mathbf{X}\tilde{\mathbf{A}}_1$, $\mathbf{H} = \mathbf{H}_{\mathbf{Z}}$ and $\tilde{\mathbf{H}} = \mathbf{H}_{\tilde{\mathbf{Z}}}$.

By $\mathbf{Y}_2 = \mathbf{X}\mathbf{A}_1\mathbf{B} + \mathbf{E}_2$ and $\mathbf{Z}(\mathbf{I} - \mathbf{H}) = \mathbf{0}$, some simple algebra yields that

$$\begin{aligned}
\Delta_2 =\ &\delta_n^2\mathrm{tr}\left\{(\mathbf{X}\mathbf{w}\mathbf{B})^\top(\mathbf{I} - \tilde{\mathbf{H}})(\mathbf{X}\mathbf{w}\mathbf{B})\right\}\\
&+ 2\delta_n\mathrm{tr}\left\{\mathbf{E}_2^\top(\mathbf{I} - \tilde{\mathbf{H}})(\mathbf{X}\mathbf{w}\mathbf{B})\right\} \quad + \mathrm{tr}\left\{\mathbf{E}_2^\top(\mathbf{H} - \tilde{\mathbf{H}})\mathbf{E}_2\right\}. \quad (8)
\end{aligned}$$

Following the same arguments as in Δ_1, it can be shown that as long as the constant C_1 is sufficiently large, the first term on the right side of (8) dominates the second term with an arbitrarily large probability.

Consider the third term on the right side of (8). By Condition 1 again, we get

$$\begin{aligned}
\|\tilde{\mathbf{Z}} - \mathbf{Z}\|_{\mathrm{F}} \le c_2\|\tilde{\mathbf{A}}_{\mathcal{D}} - \mathbf{A}_{\mathcal{D}}\|_2 &= O(\delta_nC_1),\\
\|\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}} - \mathbf{Z}^\top\mathbf{Z}\|_2 &= O(\delta_nC_1),\\
\|(\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}})^{-1} - (\mathbf{Z}^\top\mathbf{Z})^{-1}\|_2 &= O(\delta_nC_1),\\
\text{and} \quad \|\tilde{\mathbf{H}} - \mathbf{H}\|_2 &= O(\delta_nC_1),
\end{aligned}$$

where $\|\cdot\|_2$ is the spectral norm. To that end, we can show the following result holds

$$\begin{aligned}
&\mathrm{tr}\left\{\mathbf{E}_2^\top(\mathbf{H} - \tilde{\mathbf{H}})\mathbf{E}_2\right\}\\
&= \mathrm{tr}\left[n^{-1}\mathbf{E}_2^\top\mathbf{X}\left\{\mathbf{A}_1(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{A}_1^\top - \tilde{\mathbf{A}}_1(\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}})^{-1}\tilde{\mathbf{A}}_1^\top\right\}\mathbf{X}^\top\mathbf{E}_2\right]\\
&= \mathrm{tr}\left[n^{-1/2}\mathbf{E}_2^\top\mathbf{X}\left\{\mathbf{A}_1(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{A}_1^\top - \tilde{\mathbf{A}}_1(\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}})^{-1}\tilde{\mathbf{A}}_1^\top\right\}n^{-1/2}\mathbf{X}^\top\mathbf{E}_2\right]\\
&\le O(\delta_nC_1)\|n^{-1/2}\mathbf{E}_2^\top\mathbf{X}\|_{\mathrm{F}}^2\\
&= O_p(\delta_nC_1rp\log p),
\end{aligned}$$

which is of smaller order of Δ_1.

By combining the arguments for Δ_1 and Δ_2, the result in (7) holds if we choose a sufficiently large positive constant C_1.

Further, we can verify that $\|\hat{\mathbf{B}} - \mathbf{B}\|_{\mathrm{F}} = O_p(\delta_n)$ in a similar fashion. Thus, $\|\hat{\mathbf{A}} - \mathbf{A}\|_{\mathrm{F}} = O_p(\delta_n)$ follows naturally.

□

REFERENCES

[1] M. Yuan, A. Ekici, Z. Lu, and R. Monteiro, "Dimension reduction and coefficient estimation in multivariate linear regression," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 3, pp. 329–346, 2007.

[2] S. Negahban and M. J. Wainwright, "Estimation of (near) low-rank matrices with noise and high-dimensional scaling," *The Annals of Statistics*, pp. 1069–1097, 2011.

[3] G. Obozinski, M. J. Wainwright, and M. I. Jordan, "Support union recovery in high-dimensional multivariate regression," *The Annals of Statistics*, vol. 39, no. 1, pp. 1–47, 2011.

[4] Y. Kong, D. Li, Y. Fan, and J. Lv, "Interaction pursuit in high-dimensional multi-response regression via distance correlation," *The Annals of Statistics*, vol. 45, no. 2, pp. 897–922, 2017.

[5] X. Bing, M. H. Wegkamp *et al.*, "Adaptive estimation of the rank of the coefficient matrix in high-dimensional multivariate response regression models," *The Annals of Statistics*, vol. 47, no. 6, pp. 3157–3184, 2019.

[6] G. Raskutti, M. Yuan, and H. Chen, "Convex regularization for high-dimensional multiresponse tensor regression," *The Annals of Statistics*, vol. 47, no. 3, pp. 1554–1584, 2019.

[7] Z. Zheng, M. T. Bahadori, Y. Liu, and J. Lv, "Scalable interpretable multi-response regression via seed," *Journal of Machine Learning Research*, vol. 20, no. 107, pp. 1–34, 2019.

[8] J. Fan, W. Wang, and Y. Zhong, "An ℓ_∞ eigenvector perturbation bound and its application to robust covariance estimation," *Journal of Machine Learning Research*, vol. 18, no. 207, pp. 1–42, 2018.

[9] C. Zou, Y. Ke, and W. Zhang, "Estimation of low rank high-dimensional multivariate linear models for multi-response data," *Journal of the American Statistical Association*, pp. 1–11, 2020.

[10] C. Li and R. Li, "Linear hypothesis testing in linear models with high-dimensional responses," *Journal of the American Statistical Association*, pp. 1–13, 2021.

[11] W. J. Welch, "Algorithmic complexity: three np-hard problems in computational statistics," *Journal of Statistical Computation and Simulation*, vol. 15, no. 1, pp. 17–25, 1982.

[12] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, 1995.

[13] D. Foster, H. Karloff, and J. Thaler, "Variable selection is hard," in *Conference on Learning Theory*, 2015, pp. 696–709.

[14] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[15] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.

[16] S. P. Jordan, "Fast quantum algorithm for numerical gradient estimation," *Physical Review Letters*, vol. 95, no. 5, p. 050501, 2005.

[17] T. Byrnes and Y. Yamamoto, "Simulating lattice gauge theories on a quantum computer," *Physical Review A*, vol. 73, no. 2, p. 022328, 2006.

[18] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical Review Letters*, vol. 103, no. 15, p. 150502, 2009.

[19] P. Wittek, *Quantum machine learning: what quantum computing means to data mining*. Academic Press, 2014.

[20] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.

[21] J. Hu and Y. Wang, "Quantum annealing via path-integral monte carlo with data augmentation," *Journal of Computational and Graphical Statistics*, pp. 1–13, 2020.

[22] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Physical Review Letters*, vol. 79, no. 2, p. 325, 1997.

[23] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, "Tight bounds on quantum searching," *Fortschritte der Physik: Progress of Physics*, vol. 46, no. 4-5, pp. 493–505, 1998.

[24] P. Kwiat, J. Mitchell, P. Schwindt, and A. White, "Grover's search algorithm: an optical approach," *Journal of Modern Optics*, vol. 47, no. 2-3, pp. 257–266, 2000.

[25] G.-L. Long, "Grover algorithm with zero theoretical failure rate," *Physical Review A*, vol. 64, no. 2, p. 022307, 2001.

[26] P. Høyer, J. Neerbek, and Y. Shi, "Quantum complexities of ordered searching, sorting, and element distinctness," *Algorithmica*, vol. 34, no. 4, pp. 429–448, 2002.

[27] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, "Strengths and weaknesses of quantum computing," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1510–1523, 1997.

[28] T. K. Pong, P. Tseng, S. Ji, and J. Ye, "Trace norm regularization: Reformulations, algorithms, and multi-task learning," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3465–3489, 2010.

[29] M. Riediker, W. E. Cascio, T. R. Griggs, M. C. Herbst, P. A. Bromberg, L. Neas, R. W. Williams, and R. B. Devlin, "Particulate matter exposure in cars is associated with cardiovascular effects in healthy young men," *American Journal of Respiratory and Critical Care Medicine*, vol. 169, no. 8, pp. 934–940, 2004.

[30] G. Polichetti, S. Cocco, A. Spinali, V. Trimarco, and A. Nunziata, "Effects of particulate matter (pm10, pm2. 5 and pm1) on the cardiovascular system," *Toxicology*, vol. 261, no. 1-2, pp. 1–8, 2009.

[31] U. Franck, S. Odeh, A. Wiedensohler, B. Wehner, and O. Herbarth, "The effect of particle size on cardiovascular disorders–the smaller the worse," *Science of the Total Environment*, vol. 409, no. 20, pp. 4217–4221, 2011.

[32] Y.-F. Xing, Y.-H. Xu, M.-H. Shi, and Y.-X. Lian, "The impact of pm2. 5 on the human respiratory system," *Journal of Thoracic Disease*, vol. 8, no. 1, p. E69, 2016.

[33] V. C. Pun, F. Kazemiparkouhi, J. Manjourides, and H. H. Suh, "Long-term pm2. 5 exposure and respiratory, cancer, and cardiovascular mortality in older us adults," *American Journal of Epidemiology*, vol. 186, no. 8, pp. 961–969, 2017.

[34] J. Fan and H. Peng, "Nonconcave penalized likelihood with a diverging number of parameters," *The Annals of Statistics*, vol. 32, no. 3, pp. 928–961, 2004.