arXiv:2402.15686v3 [quant-ph] 25 Dec 2024

Lower bounds for quantum-inspired classical algorithms via communication complexity

Nikhil S. Mande[1] and Changpeng Shao[2]

[1]Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK
[2]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100190 China

Quantum-inspired classical algorithms provide us with a new way to understand the computational power of quantum computers for practically-relevant problems, especially in machine learning. In the past several years, numerous efficient algorithms for various tasks have been found, while an analysis of lower bounds is still missing. Using communication complexity, in this work we propose the first method to study lower bounds for these tasks. We mainly focus on lower bounds for solving linear regressions, supervised clustering, principal component analysis, recommendation systems, and Hamiltonian simulations. For those problems, we prove a quadratic lower bound in terms of the Frobenius norm of the underlying matrix. As quantum algorithms are linear in the Frobenius norm for those problems, our results mean that the quantum-classical separation is at least quadratic. As a generalisation, we extend our method to study lower bounds analysis of quantum query algorithms for matrix-related problems using quantum communication complexity. Some applications are given.

1 Introduction

Initiated by Tang in the seminal work [1], quantum-inspired classical algorithms now provide a new way for us to understand the power and limitations of quantum computers for some practically relevant problems, especially in machine learning [2]. These algorithms are very different from standard classical algorithms that usually output vector solutions. Instead, they are more comparable to quantum algorithms and can be used as a tool to study quantum advantage.

Roughly, in a quantum-inspired classical algorithm for a matrix-relevant problem, such as a linear regression problem $\arg\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|$, we are given access to a matrix A and vector $\mathbf{b}$ via some data structures (similar to QRAM access), and the goal is to output a data structure to the solution $\mathbf{x}$. The requirements of this data structure are that we must be able to query entries of the solution and sample from the distribution defined by the solution under the ℓ_2-norm. This data structure can be thus be viewed as a classical analog of the quantum state of the solution.

The importance of quantum-inspired classical algorithms lies in the finding that quantum computers only achieve polynomial speedups rather than exponential speedups as we originally expected for some matrix problems when the input matrices have low rank. A typical example is the linear regression problem. Assuming the quantum random access memory (QRAM) data structure, the best quantum algorithm known so far has complexity linear in $\|A\|_{\mathtt{F}}$, which is believed to be optimal [3]. Here $\|A\|_{\mathtt{F}}$ is the Frobenius norm of A, which is defined as the square root of the sum of the absolute squares of all entries. Using a similar data structure to QRAM, some efficient quantum-inspired classical algorithms were also proposed [2, 4–6]. The best algorithms known so far have complexity quartic in $\|A\|_{\mathtt{F}}$ [4, 5]. This implies that the quantum speedup is at most quartic in terms of the Frobenius norm. We remark that the efficiency of quantum/quantum-inspired algorithms is also influenced by other parameters, however, the Frobenius norm is usually the dominating one.

Nikhil S. Mande: nikhil.mande@liverpool.ac.uk
Changpeng Shao: changpeng.shao@amss.ac.cn

Problems	Upper bounds (Q)	Upper bounds (QIC)	Lower bounds (QIC, this paper)
Linear Regression	$\widetilde{O}(\|A\|_{\mathrm{F}})$ [3]	$\widetilde{O}(\|A\|_{\mathrm{F}}^4)$ [4]	$\widetilde{\Omega}(\|A\|_{\mathrm{F}}^2)$
Supervised clustering	$\widetilde{O}(\|A\|_{\mathrm{F}}^2\|\mathbf{b}\|^2/\varepsilon)$ [7]	$\widetilde{O}(\|A\|_{\mathrm{F}}^4\|\mathbf{b}\|^4/\varepsilon^2)$ [2]	$\widetilde{\Omega}(\|A\|_{\mathrm{F}}^4\|\mathbf{b}\|^4/\varepsilon)$
Principal component analysis	$\widetilde{O}(\|A\|_{\mathrm{F}})$ [3]	$\widetilde{O}(\|A\|_{\mathrm{F}}^6)$ [2]	$\widetilde{\Omega}(\|A\|_{\mathrm{F}}^2)$
Recommendation systems	$\widetilde{O}(\|A\|_{\mathrm{F}})$ [8]	$\widetilde{O}(\|A\|_{\mathrm{F}}^4)$ [4]	$\widetilde{\Omega}(\|A\|_{\mathrm{F}}^2)$
Hamiltonian simulation	$\widetilde{O}(\|A\|_{\mathrm{F}})$ [9]	$\widetilde{O}(\|A\|_{\mathrm{F}}^4)$ [4]	$\widetilde{\Omega}(\|A\|_{\mathrm{F}}^2)$

Table 1: Upper and lower bounds for quantum (Q) and quantum-inspired classical (QIC) algorithms for some other problems. We mainly show the dependence on $\|A\|_{\mathrm{F}}$ here. The explicit statements of these problems are given in Section 4.

To better understand quantum-classical separations and also to see how far these quantum-inspired classical algorithms are far from optimal, we provide the first approach for proving lower bounds of quantum-inspired classical algorithms.

1.1 Main results

Our main focus of this work is on the lower bounds analysis of quantum-inspired classical algorithms for five problems: linear regressions, supervised clustering, principal component analysis, recommendation systems, and Hamiltonian simulation. Algorithms for these problems are well-studied both in the quantum case (e.g., [3, 7–11], to name a few) and in the quantum-inspired classical case (e.g., [2, 4, 12–14], to name a few). There are lots of results on the lower bounds of quantum algorithms [9, 11] (also see relevant references in [15]). However, lower bounds for quantum-inspired classical algorithms are missing.

Our main technique for lower bounds analysis is communication complexity. We will establish a connection between quantum-inspired classical algorithms and communication complexity, then use some well-known results in communication complexity (such as lower bounds for the Set-Disjointness) to obtain some lower bounds. As a by-product, we also extend our method to study lower bounds analysis of quantum query algorithms for functions of matrices (see Section 5).

We now give more details. In quantum-inspired classical algorithms, we assume that we are given access to inputs via certain data structures called "SQ". Here S and Q stand for Sampling and Query, respectively, and the rigorous definition of "SQ" is given in Section 2. For example, for a vector $\mathbf{v} = (v_1, \ldots, v_n)$, $S(\mathbf{v})$ allows us to sample from the distribution defined by $\mathbf{v}$ under the ℓ_2 norm (i.e., the probability of seeing i is $|v_i|^2/\|\mathbf{v}\|^2$), and $Q(\mathbf{v})$ allows us to query an entry and/or the norm of $\mathbf{v}$.

Our main results are briefly summarised in Table 1. From the table, we can claim that for these tasks the quantum speedup is at least quadratic. Apart from the supervised clustering problem, all other bounds are not tight so far with respect to $\|A\|_{\mathrm{F}}$. So it is interesting to know for these problems (and/or other problems not included), whether the lower bounds can be improved or more efficient quantum-inspired classical algorithms can be found.

1.2 Summary of main ideas

The form of access to input data in quantum-inspired classical algorithms is very similar to that in the setting of *query complexity*. However, a quantum-inspired classical algorithm has more power than a conventional query algorithm, since, for example, it can sample from the input data in a way that query algorithms can not. Thus, showing lower bounds query algorithms does not directly imply lower bounds for quantum-inspired classical algorithms. A standard model of computation that is studied, and richer than the query model, is that of *communication complexity*. As it turns out, quantum-inspired classical algorithms can indeed be efficiently simulated by communication protocols. Thus, showing communication complexity lower bounds for a task implies lower bounds

for quantum-inspired classical algorithms solving the same task. This approach is inspired by [16].

Below we sketch the main ideas behind our lower bounds for the linear regression problem. Consider two players Alice and Bob, who are individually computationally unbounded. Alice has as input a matrix $A^{(1)}$ and a vector $\mathbf{b}^{(1)}$. Bob has another matrix $A^{(2)}$ and another vector $\mathbf{b}^{(2)}$. They can see their own inputs, but neither of the players can see the other player's input. Their goal is to solve the linear regression $\mathbf{x}_* := \arg\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|$, where $A = \begin{pmatrix} A^{(1)} \\ A^{(2)} \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} \mathbf{b}^{(1)} \\ \mathbf{b}^{(2)} \end{pmatrix}$. To solve this problem, they are allowed to communicate with each other via a pre-determined communication protocol. The communication complexity is the total number of bits used in the communication on a worst-case input. Without loss of generality, we assume the existence of a coordinator such that the communication is 2-way between each player and the coordinator. With a coordinator, it is easy to see how to generalise this idea to the multi-player model. Also, we assume that each of the entries of $A, \mathbf{b}$ are specified by $O(\log q)$ bits.

In the model of quantum-inspired classical algorithms, we are given $SQ(A)$ and $SQ(\mathbf{b})$, and the goal is to output $SQ(\mathbf{x}_*)$ up to certain relative error. A quantum-inspired classical algorithm of complexity T is a classical algorithm that uses T applications of $SQ(A), SQ(\mathbf{b})$ and an arbitrary number of other arithmetic operations independent of "SQ". We show below that a quantum-inspired classical algorithm of complexity T for the above linear regression implies a communication protocol of complexity $O(T \log(qmn))$ for the same problem, where m and n are the total number of rows and columns of A, respectively. To see this, it suffices to demonstrate a protocol for the coordinator to implement $SQ(A)$ and $SQ(\mathbf{b})$.

Let us first consider how the coordinator implements $SQ(\mathbf{b})$. Recall that $SQ(\mathbf{b})$ means sampling from the distribution defined by $\Pr(i) = |b_i|^2/\|\mathbf{b}\|^2$. The protocol is as follows:

- Firstly, Alice sends norm $\|\mathbf{b}^{(1)}\|$ and Bob sends norm $\|\mathbf{b}^{(2)}\|$ to the coordinator. At this point the coordinator knows $\|\mathbf{b}\|^2 = \|\mathbf{b}^{(1)}\|^2 + \|\mathbf{b}^{(2)}\|^2$.
- Secondly, the coordinator samples an index from the distribution defined by $\Pr(i) = \frac{\|\mathbf{b}^{(i)}\|^2}{\|\mathbf{b}\|^2}$.
- If the output is 1, then the coordinator asks Alice to sample an index from the distribution defined by $\mathbf{b}^{(1)}$ under the ℓ_2 norm. Otherwise, the coordinator asks Bob to sample an index from the distribution defined by $\mathbf{b}^{(2)}$ under the ℓ_2 norm.

Now, it is not hard to check that the result is a desired sample. Next, let us consider how the coordinator implements $Q(\mathbf{b})$, i.e., query the ith entry of $\mathbf{b}$, say. This task is simpler. Firstly, Alice sends the dimension, say m_1, of $\mathbf{b}^{(1)}$ to the coordinator. Secondly, if the index $i \leq m_1$, then the coordinator asks Alice to send back the i-th entry of $\mathbf{b}^{(1)}$. Otherwise, the coordinator asks Bob to send back the $(i - m_1)$-th entry of $\mathbf{b}^{(2)}$. The protocol for the coordinator to use $SQ(A)$ is similar.

As a result, if we know the lower bounds of communication complexity of solving (say) linear regressions, we then can obtain lower bounds of quantum-inspired classical algorithms for the same task. With this idea, we then only need to focus on the lower bounds analysis of the communication complexity of linear regressions. For this, we reduce some well-studied problems like the Set-Disjointness problem to some matrix inversion problems derived from the original linear regression problem.

1.3 Outline of the paper

The rest of the paper is organised as follows: Section 2 contains some preliminary results on quantum-inspired classical algorithms and communication complexity. Section 3 builds the formal connection between quantum-inspired classical algorithms and communication complexity. In Section 4 we show lower bounds for quantum-inspired classical algorithms. Finally, as a generalisation of the above idea, Section 5 studies the connection between quantum query complexity and quantum communication complexity of functions of matrices. Some applications are introduced. Appendix A describes another connection between quantum-inspired classical algorithms and communication complexity, which might be useful in future applications.

2 Preliminaries

2.1 The model for quantum-inspired classical algorithms

In this section, we recall definitions of the model used in quantum-inspired classical algorithms. For more details, we refer the reader to [2].

Definition 1 (Query access)**.** *For a vector* $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{C}^n$*, we have* $Q(\mathbf{v})$*,* query access to $\mathbf{v}$*, if for all* $i \in [n]$*, we can query for* v_i*. Likewise, for a matrix* $A = (A_{ij}) \in \mathbb{C}^{m\times n}$*, we have* $Q(A)$ *if for all* $(i, j) \in [m] \times [n]$*, we can query for* A_{ij}*.*

Definition 2 (Sampling and query access to a vector)**.** *For a vector* $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{C}^n$*, we have* $SQ(\mathbf{v})$*,* sampling and query access to $\mathbf{v}$*, if we can*

- *query for entries of* $\mathbf{v}$ *as in* $Q(\mathbf{v})$*;*
- *obtain independent samples of indices* $i \in [n]$*, each distributed as* $\Pr(i) = |v_i|^2/\|\mathbf{v}\|^2$*;*
- *query for* $\|\mathbf{v}\|$*.*

Definition 3 (Sampling and query access to a matrix)**.** *For a matrix* $A \in \mathbb{C}^{m\times n}$*, we have* $SQ(A)$ *if we have* $SQ(A_{i*})$ *for all* $i \in [m]$ *and* $SQ(\mathbf{a})$ *for* $\mathbf{a} = (\|A_{1*}\|, \ldots, \|A_{m*}\|)$*. Here* A_{i*} *refers to the* i*-th row of* A*.*

By a quantum-inspired classical algorithm of (query) complexity T for solving a matrix-related problem, e.g., solving a linear regression $\min \|A\mathbf{x}-\mathbf{b}\|$, we mean we "solve" the problem by obtaining the "SQ" data structure of the solution, with T applications of $SQ(A), SQ(\mathbf{b})$ and an arbitrary number of other arithmetic operations that are independent of "SQ".

2.2 Communication complexity

Communication complexity was first introduced by Andrew Yao more than four decades ago in the study of distributed computation [17]. It has wide applications in many areas, especially in showing lower bounds, which has extensively been studied both in the classical and quantum fields [18,19]. There are many models to study communication complexity in. In this work, we will focus on the coordinator model [20].

In the coordinator model, there are $k \geq 2$ players $\mathcal{P}_1, \ldots, \mathcal{P}_k$ and a coordinator $\mathcal{C}$. Each player holds some private information, and their goal is to solve some problem using as little communication as possible. The communication occurs between a player and the coordinator via a 2-way private channel. The computation is in terms of rounds: at the beginning of each round, the coordinator sends a message to one of the k players, and then that player sends a message back to the coordinator. In the end, the coordinator returns an answer. Unless mentioned otherwise, the protocol must output a correct answer with probability at least $2/3$. The communication complexity is defined to be the total number of bits sent through the channels, on a worst-case input and worst-case outcomes of the internal randomness of the protocol.

A fundamental example is the Set-Disjointness problem. When $k = 2$, player $\mathcal{P}_1$ receives an n-bit string $\mathbf{x} = (x_1, \ldots, x_n) \in \{0, 1\}^n$ and player $\mathcal{P}_2$ receives another n-bit string $\mathbf{y} = (y_1, \ldots, y_n) \in \{0, 1\}^n$, their goal is to determine if there is an i such that $x_i = y_i = 1$. It is well-known that the naive protocol ($\mathcal{P}_1$ has to send all $\mathbf{x}$ to the coordinator $\mathcal{C}$ who then sends $\mathbf{x}$ to $\mathcal{P}_2$, and finally $\mathcal{P}_2$ sends the right answer to $\mathcal{C}$) is asymptotically optimal [**?**, 21]. Namely, the communication complexity of the Set-Disjointness problem is $\Theta(n)$.

Another fundamental problem is the Gap-Hamming problem. In this problem, Alice has a n-bit string $\mathbf{x}$, Bob has a n-bit string $\mathbf{y}$, and they wish to compute

$$G(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{if } \Delta(\mathbf{x}, \mathbf{y}) \geq n/2 + \varepsilon n, \\ 0 & \text{if } \Delta(\mathbf{x}, \mathbf{y}) \leq n/2 - \varepsilon n, \\ * & \text{otherwise.} \end{cases} \tag{1}$$

In the above, $\Delta(\mathbf{x}, \mathbf{y})$ computes the hamming distance of $\mathbf{x}, \mathbf{y}$. As shown in [22] the communication complexity for this problem is $\Theta(\min(n, 1/\varepsilon^2))$.

For those problems, there are also some multiplayer versions. We state them as follows.

Definition 4 (k-player Set-Disjointness problem)**.** *For $i \in [k]$, player $\mathcal{P}_i$ receives a bit string $T_i = (T_{i1}, \ldots, T_{in}) \in \{0,1\}^n$. Their goal is to determine if there is a $j \in \{2, \ldots, k\}$ such that $T_{1\ell} = T_{j\ell} = 1$ for some $\ell \in [n]$.*

Proposition 5 (Theorem 3.3 of [20], Theorem 1 of [23])**.** *When $n \geq 3200k$, for any classical protocol that succeeds with probability $1 - 1/k^3$ for solving k-player Set-Disjointness problem, the randomised communication complexity is $\Theta(kn)$.*

The above result holds even if all T_i have Hamming weight $\Theta(n)$. This indeed follows from the proof of the above proposition in the cited references. Here we only highlight the main idea. More details can be found in [20, 23]. We use the symmetrization argument to reduce the k-player Set-Disjointness problem to a standard 2-player Set-Disjointness problem. In the symmetrization argument, Alice plays the role of one player and Bob plays the role of all other players as well as the coordinator. The 2-player Set-Disjointness problem is well studied and is known hard even if the Hamming weight of the inputs has order n. By the symmetrization argument, the complexity of the original k-player Set-Disjointness problem is $\Omega(kn)$ when $|T_j| = \Theta(n)$ for all j. Indeed, in the symmetrization argument, all inputs have Hamming weight $\Theta(n)$, see the reduction at [20, page 8]. Additionally, similar to the 2-player Set-Disjointness problem, the k-player Set-Disjointness problem is also hard when there is at most one j such that $T_{1\ell} = T_{j\ell} = 1$.

Definition 6 (k-Gap-Hamming problem)**.** *For $i \in \{1, 2, \ldots, k+1\}$, player $\mathcal{P}_i$ receives a bit string $T_i \in \{\pm 1\}^n$ with the promise that $T := \sum_{i=1}^k T_i \in \{\pm 1\}^n$ and $\sum_{j=1}^k T_j \cdot T_{k+1} \in [-c_2\sqrt{n}, c_2\sqrt{n}]$. Their goal is to determine if $\sum_{i=1}^k T_i \cdot T_{k+1} \geq c_1\sqrt{n}$ or $\leq -c_1\sqrt{n}$. If $\sum_{i=1}^k T_i \cdot T_{k+1}$ is in between then any output is allowed.*

Proposition 7 (Theorem 3.3 of [24])**.** *Any protocol that solves the k-Gap-Hamming problem with large constant probability requires $\Omega(kn)$ bits of communication.*

Let $P = (p_1, \ldots, p_n)$ be a distribution. When we say that we *approximately sample* from P, we mean that we output a sample from any distribution $\widetilde{P} = (\tilde{p}_1, \ldots, \tilde{p}_n)$ such that $\|\widetilde{P} - P\|_1 := \sum_i |p_i - \tilde{p}_i| \leq \varepsilon$ for some constant inaccuracy ε.

Definition 8 (Distributed sampling problem)**.** *Assume that Alice has as input a function $f : \{0,1\}^n \to \{\pm 1\}$ and Bob has as input a function $g : \{0,1\}^n \to \{\pm 1\}$. Their goal is to approximately sample from the distribution defined by*

$$\Pr(y) := \left(\frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x) g(x) (-1)^{x \cdot y} \right)^2 .$$

Proposition 9 (Theorem 1 of [25])**.** *There exists a universal constant ε such that, for sufficiently large n, any 2-way classical communication protocol for Distributed Fourier Sampling with shared randomness and inaccuracy ε must communicate at least $\Omega(2^n)$ bits.*

2.3 Some background on quantum computing

In this paper, we often use the Dirac notation to represent unit vectors. For a unit column vector $\mathbf{v}$, we sometimes write it as $|\mathbf{v}\rangle$. In the quantum case, it will be called a quantum state. Its complex conjugate is denoted as $\langle \mathbf{v}|$. The standard basis of $\mathbb{C}^n$ will be denoted as $|1\rangle, |2\rangle, \cdots, |n\rangle$. Let $|\mathbf{a}\rangle = \sum_i a_i |i\rangle$ be a quantum state. The coefficients a_i must satisfy $\sum_{i=1}^n |a_i|^2 = 1$. When we measure it in the standard basis, then we will observe i with probability $|a_i|^2$. So measurement naturally corresponds to the operation of sampling. In this paper, the quantum notation $|0\rangle^n$ will also be used. This is the same as $|1\rangle$ in $\mathbb{C}^{2^n}$. For more about quantum computation, we refer the reader to the book [26].

3 The connection

We below discuss the connection between quantum-inspired classical algorithms and communication complexity. Our main result is the following theorem, which shows that there is an efficient simulation of a quantum-inspired classical algorithm using a classical communication protocol.

Theorem 10. *In the multi-player coordinator model, for each $i \in \{1, \ldots, k\}$, assume that player $\mathcal{P}_i$ holds a matrix $A^{(i)} \in \mathbb{R}^{\ell_i \times n}$ and a vector $\mathbf{b}^{(i)} \in \mathbb{R}^{m_i}$ with $m := \sum_i \ell_i = \sum_i m_i$. Assume that all entries are specified by $O(\log q)$ bits. Let*

$$A = \begin{pmatrix} A^{(1)} \\ \vdots \\ A^{(k)} \end{pmatrix}_{m \times n}, \quad \mathbf{b} = \begin{pmatrix} \mathbf{b}^{(1)} \\ \vdots \\ \mathbf{b}^{(k)} \end{pmatrix}_{m \times 1}. \tag{2}$$

Then we have the following:

- *The coordinator $\mathcal{C}$ can use $SQ(A)$ $O(T)$ times, using $O((T+k)\log(qmn))$ bits of communication.*
- *The coordinator $\mathcal{C}$ can use $SQ(\mathbf{b})$ $O(T)$ times, using $O((T+k)\log(qm))$ bits of communication.*

Proof. We will prove the second claim first. For the coordinator $\mathcal{C}$ to use $SQ(\mathbf{b})$, i.e., sample from and query for $\mathbf{b}$, the player $\mathcal{P}_i$ first sends $\|\mathbf{b}^{(i)}\|$ to $\mathcal{C}$. This requires $O(k\log(qm))$ bits of communication. Here $O(\log(qm))$ is the number of bits used to specify $\|\mathbf{b}^{(i)}\|$. Notice that this only needs to be done once for the whole protocol, rather than once for each access. At this point, the coordinator knows $\|\mathbf{b}\|$, and also can sample from the distribution

$$\mathcal{D}_{\mathbf{b}}(i) = \frac{\|\mathbf{b}^{(i)}\|^2}{\|\mathbf{b}\|^2}, \quad i \in \{1, 2, \ldots, k\}.$$

To sample from $\mathbf{b}$, the coordinator first uses the distribution $\mathcal{D}_{\mathbf{b}}$ to obtain a random index i, then communicates with $\mathcal{P}_i$ and asks $\mathcal{P}_i$ to sample from the distribution defined by $\mathbf{b}^{(i)}$ under the ℓ_2-norm. For an arbitrary index $j \in [m]$, suppose the j'th entry of $\mathbf{b}$ is held by player i, and say $b_k^{(i)} = b_j$. Then it is easy to see that

$$\Pr(j) = \frac{\|\mathbf{b}^{(i)}\|^2}{\|\mathbf{b}\|^2} \frac{|b_k^{(i)}|^2}{\|\mathbf{b}^{(i)}\|^2} = \frac{|b_j|^2}{\|\mathbf{b}\|^2},$$

which is the same as the distribution defined in $SQ(\mathbf{b})$.

To query the j-th entry of $\mathbf{b}$, the coordinator can communicate with the player who holds index j. To this end, player $\mathcal{P}_i$ only needs to send m_i to $\mathcal{C}$, which costs $O(\log m)$ bits of communication. So in total $O(k \log m)$ bits of communication. In conclusion, for the coordinator $\mathcal{C}$ to use $SQ(\mathbf{b})$ $O(T)$ times, they use $O(T\log(qm) + k\log(qm))$ bits of communication. Here $O(T\log(qm))$ counts the total number of bits of receiving indices or entries from all players, and $O(k\log(qm))$ counts the total number of bits of receiving the norm $\|\mathbf{b}^{(i)}\|$ and m_i from player $\mathcal{P}_i$ for all i.

Similarly, for the coordinator to use $SQ(A)$, the player $\mathcal{P}_i$ first sends $\|A^{(i)}\|_{\mathtt{F}}$ to the coordinator $\mathcal{C}$. This costs $O(k\log(qmn))$ bits of communication, where $O(\log(qmn))$ is the number of bits used to specify $\|A^{(i)}\|_{\mathtt{F}}$. The coordinator now has access to the distribution

$$\mathcal{D}_A(i) = \frac{\|A^{(i)}\|_{\mathtt{F}}^2}{\|A\|_{\mathtt{F}}^2}, \quad i \in \{1, 2, \ldots, k\}.$$

As in Definition 3, let $\mathbf{a} = (\|A_{1*}\|, \cdots, \|A_{m*}\|)$, where A_{i*} is the i'th row of A. The coordinator can use $SQ(\mathbf{a})$ in a similar way to $SQ(\mathbf{b})$. Now each entry is specified by $O(\log(qn))$ bits. To use $SQ(A_{i*})$, the coordinator only needs to communicate with the player who has the row index i. In summary, for the coordinator to use $SQ(A)$ $O(T)$ times, they use $O((T+k)\log(qmn))$ bits of communication. □

Recall that a quantum-inspired classical algorithm of complexity $O(T)$ means an algorithm that uses $O(T)$ applications of the data structure "SQ" and an arbitrary number of other arithmetic operations. As a result of Theorem 10, if there is a quantum-inspired classical algorithm of complexity $O(T)$, then there is a protocol for the same problem with communication complexity $O((T+k)\log(qmn))$. So we can use this to prove lower bounds of T. To obtain a non-trivial

lower bound from this result, it is useful to consider the regime where k and q are small. This is usually easy to enforce in the multi-player model when $k \ll T$. Indeed, for practical applications below, we can design appropriate models such that $q = O(m+n)$ and $k, \log(q) \ll T$. Thus the communication complexity is mainly dominated by T.

Remark 11. In Theorem 10, it is possible that $A^{(i)}$ or $\mathbf{b}^{(i)}$ is empty or public for some i. The above argument still works. A very special case is that the whole matrix A or the whole vector $\mathbf{b}$ belongs to a particular player, say player i. In this case, for the coordinator to use $SQ(A)$ or $SQ(\mathbf{b})$, it suffices to communicate with player i.

Note that there are already many efficient quantum-inspired classical algorithms for a wide range of problems, e.g., see [2,4], so we can use these algorithms to construct efficient communication protocols for the same problems based on Theorem 10. One big advantage of this is that when studying communication complexity, the assumptions on quantum-inspired classical algorithms can be greatly relaxed. More precisely, one costly step of quantum-inspired classical algorithms is the preprocessing step of building the data structure "SQ" for the input matrices or vectors. However, in communication complexity, local costs are not considered since the individual players are considered to be all-powerful. So each player can build the data structure "SQ" to their own inputs independently, and this cost will not be counted when analyzing the communication complexity (note that the only power that we needed the players to have in the proof of Theorem 10 was SQ access to their own inputs). Namely, the preprocessing step can be resolved easily in the model of communication complexity. This also means that quantum-inspired classical algorithms can play important roles in distributed computation. To support this point, we introduce another connection in Appendix A, which we expect to be more useful for applications. However, we will not consider specific applications of this in the current paper.

4 Lower bounds for quantum-inspired classical algorithms

In this section, we use Theorem 10 to prove lower bounds for quantum-inspired classical algorithms for five problems: linear regression, supervised clustering, principal component analysis, recommendation systems, and Hamiltonian simulation. Below, the statements of these problems come from [2].

Problem (Linear regression)**.** *Let A be a matrix and $\mathbf{b}$ be a vector, given $SQ(A)$ and $SQ(\mathbf{b})$, output $SQ(\tilde{\mathbf{x}}_*)$ such that $\|\tilde{\mathbf{x}}_* - \mathbf{x}_*\| \leq \varepsilon \|\mathbf{x}_*\|$, where $\mathbf{x}_* = A^+\mathbf{b}$.*

From [4], the complexity of the best quantum-inspired classical algorithm known so far for linear regression is $\widetilde{O}(\kappa_{\mathtt{F}}^4 \kappa^{10}/\varepsilon^2\gamma^2)$, where $\kappa_{\mathtt{F}}, \kappa, \gamma$ are defined as follows: Let $\sigma_{\min}$ be the minimal nonzero singular value of A, then

$$\kappa_{\mathtt{F}} := \|A\|_{\mathtt{F}}/\sigma_{\min}, \quad \kappa := \|A\|/\sigma_{\min}, \quad \gamma := \|A\mathbf{x}_*\|/\|\mathbf{b}\|. \tag{3}$$

These are the scaled condition number, condition number and the overlap of $\mathbf{b}$ is the column space of A, respectively. In the special case when A is row-sparse, the complexity is only $O(s\kappa_{\mathtt{F}}^2 \log(1/\varepsilon))$ when $\gamma = \Theta(1)$ [5], where s is the row sparsity (i.e., the maximal number of nonzero entries of each row). Thus there are five parameters we have to consider in the lower bounds analysis, i.e., $\kappa_{\mathtt{F}}, \kappa, \varepsilon, \gamma, s$. In the lower bounds analysis below, we will mainly focus on the analysis with respect to $\kappa_{\mathtt{F}}$, which relates the rank of A and is usually the dominating term in the complexity. Indeed, if the rank of A is r, then it is obvious that $\kappa_{\mathtt{F}} \geq \sqrt{r}$ because $\kappa_{\mathtt{F}}^2 = \|A\|_{\mathtt{F}}^2/\sigma_{\min}^2$ and $\|A\|_{\mathtt{F}}^2$ equals the square sum of all singular values. So $\kappa_{\mathtt{F}}$ can be very large even if the matrix is well-conditioned.

With Theorem 10, it suffices to give lower bounds on the communication complexity of linear regressions where $A, \mathbf{b}$ are defined in the form of (2) in the coordinator model. As our main focus will be on the dependence on $\kappa_{\mathtt{F}}$, in our reductions we try to ensure that all the other parameters $\kappa, \varepsilon, \gamma, s$ are $O(1)$, to obtain non-trivial lower bounds in terms of $\kappa_{\mathtt{F}}$. We use the notation $\widetilde{\Omega}(\cdot)$ to hide all factors that are polylogarithmic in the input size. Specifically, we will ignore the logarithmic factors in Theorem 10 for the sake of readability. Below, we focus on the sampling task in SQ in two cases: row-sparse and dense cases.

Proposition 12 (Sampling in the row-sparse case)**.** *Assume that A is row sparse and $\varepsilon \in (0,1)$ is a constant. Then ε-approximately sampling from $A^{+}\mathbf{b}$ requires making $\widetilde{\Omega}((\kappa^2+\kappa_{\mathrm{F}})/\gamma)$ calls to $SQ(A), SQ(\mathbf{b})$.*

Proof. We give a reduction from the k-player Set-Disjointness problem, defined in Definition 4. Suppose the k players $\mathcal{P}_1,\ldots,\mathcal{P}_k$ are given inputs to the Set-Disjointness problem (using the notation given in Definition 4). Player $\mathcal{P}_j$ constructs the following vector:

$$\mathbf{t}_j = \sum_{\ell=1}^{n} T_{j\ell}|\ell\rangle, \quad |\mathbf{t}_j\rangle = \mathbf{t}_j/\alpha_j,$$

where $\alpha_j = \|\mathbf{t}_j\| = \Theta(\sqrt{n})$ for all j. Here we used the ket notation $\{|1\rangle,\ldots,|n\rangle\}$ to represent the standard basis of $\mathbb{C}^n$, see Subsection 2.3. So $|\mathbf{t}_j\rangle$ is a unit vector.

Let β_A, β_b be parameters that will be fixed later. Let

$$A = \beta_A|1\rangle_k \otimes |1\rangle_n\langle 1|_k + \sum_{j=2}^{k} |j\rangle_k \otimes |\mathbf{t}_j\rangle\langle j|_k, \quad \mathbf{b} = \beta_b|1\rangle_k \otimes |1\rangle_n + n\sum_{j=2}^{k} |j\rangle_k \otimes |\mathbf{t}_1\rangle,$$

where the subindex k or n refers to the dimension of base vectors. In matrix form,

$$A = \begin{pmatrix} \beta_A|1\rangle_n & & & \\ & |\mathbf{t}_2\rangle & & \\ & & \ddots & \\ & & & |\mathbf{t}_k\rangle \end{pmatrix}_{kn\times k}, \quad \mathbf{b} = \begin{pmatrix} \beta_b|1\rangle_n \\ n|\mathbf{t}_1\rangle \\ \vdots \\ n|\mathbf{t}_1\rangle \end{pmatrix}_{kn\times 1}.$$

In this construction, the first-row block of A and $\mathbf{b}$ are public for all players. The second-row block of A belongs to player 2, and the last-row block of A belongs to player k. The whole vector of $\mathbf{b}$ except the first-row block belongs to player 1. This satisfies the conditions made in Theorem 10, also see Remark 11.

For the above matrix, the row sparsity is 1, and it is easy to check that

$$A^{\mathrm{T}}A = \beta_A^2|1\rangle_k\langle 1|_k + \sum_{j=2}^{k}|j\rangle_k\langle j|_k, \quad A^{\mathrm{T}}\mathbf{b} = \beta_A\beta_b|1\rangle_k + n\sum_{j=2}^{k}\langle \mathbf{t}_1|\mathbf{t}_j\rangle|j\rangle_k.$$

The optimal solution of the linear regression $\arg\min\|A\mathbf{x}-\mathbf{b}\|$ is

$$\mathbf{x}_* = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}\mathbf{b} = \frac{\beta_b}{\beta_A}|1\rangle_k + n\sum_{j=2}^{k}\langle \mathbf{t}_1|\mathbf{t}_j\rangle|j\rangle_k.$$

If there is no intersection for the k-player Set-Disjointness problem, then $\mathbf{x}_* = \frac{\beta_b}{\beta_A}|1\rangle_k$. Measuring this state only returns index 1 with probability 1. If there is an intersection with one common index, then there is a $j\in\{2,\ldots,k\}$ such that

$$\mathbf{x}_* = \frac{\beta_b}{\beta_A}|1\rangle_k + \frac{n}{\alpha_1\alpha_j}|j\rangle_k.$$

Note that $\alpha_1\alpha_j = \Theta(n)$, so if we choose $\beta_A = \Theta(\beta_b)$, then with a constant probability we will see index j by sampling from the distribution defined by $\mathbf{x}_*$. In conclusion, if we can sample from the solution, we then can solve the k-player Set-Disjointness problem using a constant number of such samples.

In the remaining part of the proof we analyse the parameters of this problem and conclude the lower bound using Theorem 10 and Proposition 5.

The overlap of $\mathbf{b}$ in the column space of A is

$$\gamma^2 = \frac{\|A\mathbf{x}_*\|^2}{\|\mathbf{b}\|^2} = \frac{\beta_b^2 + n^2\sum_{j=2}^{k}\langle \mathbf{t}_1|\mathbf{t}_j\rangle^2}{\beta_b^2+(k-1)n^2} = \frac{\beta_b^2+c}{\beta_b^2+(k-1)n^2}$$

for some constant $c = \Theta(1)$. We also have

$$\|A\|_{\mathrm{F}}^2 = \beta_A^2 + k - 1, \quad \sigma_{\min} = \min\{\beta_A, 1\}.$$

So

$$\kappa^2 = \frac{\max(\beta_A^2, 1)}{\min(\beta_A^2, 1)}, \quad \kappa_{\mathrm{F}}^2 = \frac{\|A\|_{\mathrm{F}}^2}{\sigma_{\min}^2} = \frac{\beta_A^2 + k - 1}{\min(\beta_A^2, 1)}, \quad \gamma^2 = \frac{\beta_b^2 + c}{\beta_b^2 + (k-1)n^2}.$$

If we choose $\beta_A^2 = \beta_b^2 = k$, then $\kappa^2, \kappa_{\mathrm{F}}^2 = \Theta(k), \gamma^2 = \Theta(1/n^2)$. Now we cannot determine the dependence on κ_{F}. But note that $\kappa_{\mathrm{F}} \geq \kappa$, so by Theorem 10 a lower bound we can claim is $\widetilde{\Omega}(\kappa^2/\gamma)$ because the communication complexity of the k-player Set-Disjointness problem is $\Theta(kn)$ by Proposition 5. If we choose $\beta_A^2 = \beta_b^2 = 1$, then $\kappa = 1, \kappa_{\mathrm{F}}^2 = \Theta(k), \gamma^2 = \Theta(1/kn^2)$, now we obtain a lower bound of $\widetilde{\Omega}(\kappa_{\mathrm{F}}/\gamma)$.

One thing we did not analyse is the error. In the above constructions, we either have $\mathbf{x}_* = |1\rangle_k$ or $\mathbf{x}_* = |1\rangle_k + d|j\rangle_k$ with $d = \Theta(1)$ depending on the result of the Set-Disjointness problem. The error will not change the sampling result too much when the inaccuracy $\varepsilon = O(1)$ is a small constant because the two solutions are far from each other. □

Proposition 13 (Sampling in the dense case)**.** *Assume that $\varepsilon \in (0, 1)$ is a constant. Then ε-approximately sampling from $A^+\mathbf{b}$ requires making $\widetilde{\Omega}(\kappa_{\mathrm{F}}^2)$ calls to $SQ(A), SQ(\mathbf{b})$.*

Proof. Here we give a reduction from the distributed sampling problem. We below use the notations stated in Definition 8. Let $D_f = \sum_x f(x)|x\rangle\langle x|$ be the diagonal matrix such that the x-th diagonal entry is $f(x)$ and let $|g\rangle = \frac{1}{\sqrt{2^n}} \sum_x g(x)|x\rangle$ be a unit column vector whose entries are $g(x)$, then the distribution in the distributed sampling problem is defined by the state $(D_f H^{\otimes n})^{-1}|g\rangle$, where $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ is the Hadamard gate. Now Alice has the whole matrix $A := D_f H^{\otimes n}$ and Bob has the whole vector $\mathbf{b} := |g\rangle$. This clearly satisfies the conditions in Theorem 10. It is also easy to check that for matrix $D_f H^{\otimes n}$ we have $\kappa_{\mathrm{F}}^2 = 2^n$, $\kappa = \gamma = 1$. So we obtain a lower bound of $\widetilde{\Omega}(\kappa_{\mathrm{F}}^2)$ by Proposition 9 and Theorem 10. □

Problem (Supervised clustering)**.** *Let $\mathbf{p}, \mathbf{q}_1, \ldots, \mathbf{q}_n \in \mathbb{R}^d$ and $\varepsilon \in (0, 1)$, denote*

$$A = \begin{pmatrix} -\mathbf{p}/\|\mathbf{p}\| \\ \mathbf{q}_1/\|\mathbf{q}_1\|\sqrt{n} \\ \vdots \\ \mathbf{q}_n/\|\mathbf{q}_n\|\sqrt{n} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \|\mathbf{p}\| \\ \|\mathbf{q}_1\|/\sqrt{n} \\ \vdots \\ \|\mathbf{q}_n\|/\sqrt{n} \end{pmatrix}.$$

Given $SQ(A), SQ(\mathbf{b})$, compute $\|\mathbf{b}^{\mathrm{T}} A\|^2 \pm \varepsilon$.

If we denote $\mathbf{q} = \frac{1}{n} \sum_{i=1}^n \mathbf{q}_i$, then $\|\mathbf{b}^{\mathrm{T}} A\|^2 = \|\mathbf{p} - \mathbf{q}\|^2$, which is the distance between $\mathbf{p}$ and the center of $\{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$. This task is widely used in supervised cluster assignments. In this problem, we are given a vector $\mathbf{p}$ and some sets of vectors $Q_1, \ldots, Q_k$. The goal is to assign $\mathbf{p}$ to one of the sets. In supervised clustering, the criterion relies on the distance between $\mathbf{p}$ and the centers. We will assign $\mathbf{p}$ to Q_i if the distance between $\mathbf{p}$ and the center of Q_i is the smallest one.

For supervised clustering, by [2, Corollary 6.10], there is a quantum-inspired classical algorithm with complexity $\widetilde{O}(\|A\|_{\mathrm{F}}^4 \|\mathbf{b}\|^4 \varepsilon^{-2})$. In the quantum case, by [7], there is a quantum algorithm of complexity $\widetilde{O}(\|A\|_{\mathrm{F}}^2 \|\mathbf{b}\|^2 \varepsilon^{-1})$. Below we show a near-tight lower bound of quantum-inspired classical algorithms, which implies that the quantum-classical separation for supervised clustering is quadratic.

Proposition 14. *For supervised clustering, any quantum-inspired classical algorithms must use $\widetilde{\Omega}(\|A\|_{\mathrm{F}}^4 \|\mathbf{b}\|^4 \varepsilon^{-1} + \varepsilon^{-2})$ applications of $SQ(A), SQ(\mathbf{b})$.*

Proof. We obtain the first term in the complexity via a reduction from the n-Gap-Hamming problem, see Definition 6. Let $\mathbf{x}, \mathbf{y}_i \in \{\pm 1\}^d$ and α be a scaling parameter, which will be determined later. Let $\mathbf{p} = \alpha \mathbf{x}, \mathbf{q}_i = \alpha \mathbf{y}_i$. This construction coincides with the setting stated in Theorem 10. Now we can compute that $\|A\|_{\mathrm{F}}^2 = 2, \|\mathbf{b}\|^2 = 2\alpha^2 d$. Moreover, $\|\mathbf{p} - \mathbf{q}\|_2^2 = 2\alpha^2 d - \frac{2\alpha^2}{n} \sum_{i=1}^n \mathbf{x} \cdot \mathbf{y}_i$. In

the n-Gap-Hamming problem, the promise is that $\sum_{i=1}^n \mathbf{x}\cdot\mathbf{y}_i \geq \sqrt{d}$ or $\leq -\sqrt{d}$. To distinguish them it suffices to choose $\varepsilon = \alpha^2\sqrt{d}/n$. We now set α such that $\alpha^2\sqrt{d} = 1$, then $\varepsilon = 1/n$ and $\|\mathbf{b}\|^2 = 2\sqrt{d}$. By Proposition 7, the lower bound of the n-Gap-Hamming problem is $\Omega(nd) = \Omega(\|A\|_{\mathrm{F}}^4\|\mathbf{b}\|^4\varepsilon^{-1})$, so we obtain the first part of the claimed lower bound using Theorem 10.

In particular, when $n = 1$, the reduction is from the (2-player) Gap-Hamming problem, see Equation (1). We now set $\mathbf{p} = \mathbf{x}/\sqrt{d}, \mathbf{q} = \mathbf{y}/\sqrt{d}$ as unit vectors. Then $\|\mathbf{p}-\mathbf{q}\|_2^2 = \frac{4}{d}\Delta(\mathbf{x},\mathbf{y})$. If we can approximate this quantity up to additive error ε, we then can solve the Gap-Hamming problem. So we obtain a lower bound of $\widetilde{\Omega}(\varepsilon^{-2})$. □

Problem (Principal component analysis)**.** *Let $A \in \mathbb{C}^{m\times n}$ be a matrix such that $A^\dagger A$ has top k eigenvalues $\{\lambda_i : i \in [k]\}$ and corresponding eigenvectors $\{\mathbf{v}_i : i \in [k]\}$. Given $SQ(A)$, with probability $\geq 1-\delta$, compute eigenvalues $\{\tilde{\lambda}_i : i \in [k]\}$ such that $\sum_{i\in[k]} |\tilde{\lambda}_i - \lambda_i| \leq \varepsilon\|A\|_{\mathrm{F}}^2$ and eigenvectors $\{SQ(\mathbf{v}_i) : i \in [k]\}$ such that $\|\mathbf{v}_i - \tilde{\mathbf{v}}_i\| \leq \varepsilon$.*

About this problem, by [2, Corollary 6.12] there is a quantum-inspired classical algorithm whose complexity is $\widetilde{O}(\frac{\|A\|_{\mathrm{F}}^6}{\lambda_k^2\|A\|^2}\eta^{-6}\varepsilon^{-6})$, where $\eta = \min_{i\in[k]} |\lambda_i - \lambda_{i+1}|/\|A\|^2$. In particular, when $k = 1$, the complexity becomes $\widetilde{O}(\frac{\|A\|_{\mathrm{F}}^6\|A\|^6}{|\lambda_1-\lambda_2|^6}\varepsilon^{-6})$. In comparison, the quantum complexity is $\widetilde{O}(\|A\|_{\mathrm{F}}/\varepsilon)$ when $k = 1$ [3]. In this particular case, we can prove the following lower bound for quantum-inspired classical algorithms. It indicates the quantum-classical separation is at least quadratic.

Proposition 15. *Let A be a matrix with maximal singular value σ and corresponding right singular vector $\mathbf{v}$, then computing $\sigma \pm \varepsilon$ or ε-approximately sampling from $\mathbf{v}$ requires $\widetilde{\Omega}(\|A\|_{\mathrm{F}}^2)$ applications of $SQ(A)$.*

Proof. We show a reduction from the Set-Disjointness problem. In the Set-Disjointness problem, Alice has $(a_1,\ldots,a_n) \in \{0,1\}^n$ and Bob has $(b_1,\ldots,b_n) \in \{0,1\}^n$. The goal is to determine if there is an i such that $a_i = b_i = 1$. If i exists, we can assume that there is only one such i. For this problem, the lower bound is $\Omega(n)$. To use this result, Alice constructs a diagonal matrix $D_a = \mathrm{diag}(a_1,\ldots,a_n)$, and Bob constructs another diagonal matrix $D_b = \mathrm{diag}(b_1,\ldots,b_n)$. Let $A = \begin{pmatrix} D_a \\ D_b \end{pmatrix}$. Then we have $A^\dagger A = D_a^2 + D_b^2$. If there is an intersection, say $a_i = b_i = 1$, then $\sigma = \sqrt{2}$ and $\mathbf{v} = |i\rangle$. Otherwise $\sigma = 1$. The corresponding right singular vectors are $\mathbf{v} = |j\rangle$ for any j with $a_j = 0, b_j = 1$ or $a_j = 1, b_j = 0$. So if we can approximate σ or sampling from $\mathbf{v}$, we then can solve the Set-Disjointness problem. In this construction, we have $\|A\|_{\mathrm{F}}^2 = \Theta(n), \|A\| = \Theta(1), \varepsilon = \Theta(1)$. So we obtain the claimed lower bound by Theorem 10. □

Recall that if the SVD of A is $\sum_i \sigma_i|u_i\rangle\langle v_i|$, then we define $A_{\geq\delta} := \sum_{i:\sigma_i\geq\delta} \sigma_i|u_i\rangle\langle v_i|$.

Problem (Recommendation systems)**.** *Let $A \in \mathbb{C}^{m\times n}$ be a matrix and $i \in [m]$. Let $\varepsilon, \delta > 0$. Given $SQ(A)$, the goal is to ε-approximately sample from the i-th row of $A_{\geq\delta}$.*

For the problem of recommendation systems, by [4, Corollary 1.2], there is a quantum-inspired classical algorithm of complexity $O(\|A\|_{\mathrm{F}}^4/\sigma^8\varepsilon^2)$. In comparison, there is a quantum algorithm of complexity $O(\|A\|_{\mathrm{F}}/\sigma)$ [8]. As a corollary of Proposition 15, we have the following lower bound.

Proposition 16. *Approximately sample from the i-th row of $A_{\geq\delta}$ requires making $\widetilde{\Omega}(\|A\|_{\mathrm{F}}^2)$ calls to $SQ(A)$.*

Proof. We use the same construction as in the proof of Proposition 15. Now we choose $\delta \in (1,\sqrt{2})$, then $A_{\geq\delta} = \sigma|\mathbf{u}\rangle\langle\mathbf{v}|$ for some $\mathbf{u}$ if there is an intersection, and $A_{\geq\delta} = 0$ if no intersection. So if we can sample from the i-th row of $A_{\geq\delta}$ approximately, we then can solve the Set-Disjointness problem. □

Problem (Hamiltonian simulation)**.** *Let $A \in \mathbb{C}^{n\times n}$ be a Hermitian matrix with $\|A\| \leq 1$ and $\mathbf{v} \in \mathbb{C}^{n\times n}$ be a unit vector. Let $\varepsilon, \delta \in (0,1]$. Given $SQ(A), SQ(\mathbf{v})$, output $SQ(\mathbf{u})$ with probability at least $1-\delta$ such that $\|\mathbf{u} - e^{iAt}\mathbf{v}\| \leq \varepsilon$.*

From [4, Corollary 1.6], we know that there is a quantum-inspired classical algorithm for Hamiltonian simulation with complexity $\widetilde{O}(\|A\|_{\mathrm{F}}^4 t^8/\varepsilon^2)$. In comparison, the quantum algorithm has complexity $\widetilde{O}(\|A\|_{\mathrm{F}}t)$. Regarding this problem, we have the following lower bound:

Proposition 17. *Assume that $\varepsilon \in (0,1)$ is a constant. Then ε-approximately sampling from $e^{iAt}\mathbf{v}$ requires making $\widetilde{\Omega}(\|A\|_{\mathrm{F}}^2)$ calls to $SQ(A), SQ(\mathbf{v})$.*

Proof. The result follows via a reduction from the distributed Fourier sampling problem. In this problem, Alice has a function $f : \{0,1\}^n \to \{\pm 1\}$ and Bob has a function $g : \{0,1\}^n \to \{\pm 1\}$. Let D_f, D_g be the diagonal matrices defined by $f(x), g(x)$ respectively. Let H_2 be the Hadamard gate. Then their goal is to sample from $D_f H_2^{\otimes n} D_f D_g H_2^{\otimes n} |0\rangle^n$. By Proposition 9, the communication complexity of this problem is $\Theta(2^n)$.

Let

$$A = D_f \left(\frac{1}{2n} \sum_{j=1}^{n} I_2^{\otimes(j-1)} \otimes (I_2 - H_2) \otimes I_2^{\otimes(n-j)} \right) D_f,$$

then it is easy to check that $D_f H_2^{\otimes n} D_f = e^{iAt}$, where $t = n\pi$. This matrix is in Alice's hand. For this matrix, it is not hard to show that $\|A\|_{\mathrm{F}}^2 = \Theta(2^n)$, $\|A\| = 1$. In the Hamiltonian simulation, we set $\mathbf{v} = D_g H_2^{\otimes n} |0\rangle^n$. This vector is in Bob's hand. Finally, by Proposition 9 and Theorem 10, we obtain a lower bound of $\widetilde{\Omega}(\|A\|_{\mathrm{F}}^2)$. □

5 A connection between quantum query and communication complexities of matrix problems

In this section, we generalise Theorem 10 to the quantum case. We focus on two types of quantum algorithms for matrix-based problems.

The query model: For a matrix $A = (A_{ij}) \in \mathbb{C}^{m \times n}$, we assume that there is a quantum oracle that can query its entries in the form of

$$\mathcal{O}_A : |i,j\rangle|b\rangle \mapsto |i,j\rangle|b \oplus A_{ij}\rangle. \tag{4}$$

For sparse matrices, two additional oracles that reveal the information of positions of nonzero entries are given, i.e.,

$$\mathcal{O}_R : |i\rangle|j\rangle \mapsto |i\rangle|r_{ij}\rangle, \tag{5}$$
$$\mathcal{O}_C : |i\rangle|j\rangle \mapsto |c_{ij}\rangle|j\rangle, \tag{6}$$

where r_{ij} is the index for the j-th non-zero entry of the i-th row of A, and if there are less than s non-zero entries, then it is $j+n$; c_{ij} are defined similarly according to columns.

In this setting, we focus on quantum query algorithms of the following form:

$$U_T \mathcal{O} U_{T-1} \mathcal{O} \cdots U_2 \mathcal{O} U_1 \, |0..0\rangle, \tag{7}$$

where $\mathcal{O} \in \{I \otimes \mathcal{O}', \mathcal{O}' \otimes I : \mathcal{O}' \in \{\mathcal{O}_A, \mathcal{O}_A^{-1}, \mathcal{O}_R, \mathcal{O}_R^{-1}, \mathcal{O}_C, \mathcal{O}_C^{-1}\}\}$, and $U_1, U_2, \ldots, U_T$ are unitaries independent of the oracles.

The block-encoding model: For a matrix A, we assume that we are given a block-encoding of A, i.e., we are given the following unitary for some $\alpha \geq \|A\|$

$$W = \begin{pmatrix} A/\alpha & \cdot \\ \cdot & \cdot \end{pmatrix}. \tag{8}$$

In this setting, we focus on quantum algorithms of the following form:

$$U_T \widetilde{W} U_{T-1} \widetilde{W} \cdots U_2 \widetilde{W} U_1 |0..0\rangle, \tag{9}$$

where $\widetilde{W} \in \{I \otimes W', W' \otimes I : W' \in \{W, W^\dagger, c\text{-}W, c\text{-}W^\dagger\}\}$ and $U_1, U_2, \ldots, U_T$ are unitaries independent of W. Here, c-W means control-W.

Let $|\psi_A\rangle$ be the final state of (7) or (9). Regarding the output of quantum algorithms, if the goal is to produce a quantum state, then we assume that $|\psi_A\rangle = \sqrt{1-\varepsilon}\,|0\rangle|\phi_0\rangle + \sqrt{\varepsilon}\,|1\rangle|\phi_1\rangle$ for some $\varepsilon \in [0, 1/3]$. There is flag qubit (i.e., $|0\rangle$) that tells us the desired state (i.e., $|\phi_0\rangle$). If the output state is a number, say a bit, then we assume that $|\psi_A\rangle = p|0\rangle|\phi_0\rangle + \sqrt{1-p^2}|1\rangle|\phi_1\rangle$. If the output is 0 then $|p|^2 \geq 2/3$, otherwise $|p|^2 \leq 1/3$. For the above two tasks, we only need to measure the first qubit. With high probability, we will obtain the desired result.

There are some connections between the above two models. To see this, recall that the block-encoding of a sparse matrix is constructed as follows [9]: assume that $\max_{i,j} |A_{ij}| \leq 1$ for simplicity, define unitary

$$U_L : |0\rangle|i\rangle \to |i\rangle|0\rangle \to \frac{1}{\sqrt{s}} \sum_{j=1}^{s} |i\rangle|j\rangle \xrightarrow{\mathcal{O}_R} \frac{1}{\sqrt{s}} \sum_{j=1}^{s} |i\rangle|r_{ij}\rangle. \tag{10}$$

And define unitary

$$\begin{aligned} U_R : \quad & |0\rangle|j\rangle|0\rangle|0\rangle \to \frac{1}{\sqrt{s}} \sum_{i=1}^{s} |i\rangle|j\rangle|0\rangle|0\rangle \xrightarrow{\mathcal{O}_C \otimes I \otimes I} \frac{1}{\sqrt{s}} \sum_{i=1}^{s} |c_{ij}\rangle|j\rangle|0\rangle|0\rangle \\ & \xrightarrow{\mathcal{O}_A \otimes I} \frac{1}{\sqrt{s}} \sum_{i=1}^{s} |c_{ij}\rangle|j\rangle|A_{c_{ij},j}\rangle|0\rangle \\ & \to \frac{1}{\sqrt{s}} \sum_{i=1}^{s} |c_{ij}\rangle|j\rangle|A_{c_{ij},j}\rangle \left(A_{c_{ij},j}|0\rangle + \sqrt{1 - A_{c_{ij},j}^2}|1\rangle\right) \\ & \xrightarrow{\mathcal{O}_A^{-1} \otimes I} \frac{1}{\sqrt{s}} \sum_{i=1}^{s} |c_{ij}\rangle|j\rangle|0\rangle \left(A_{c_{ij},j}|0\rangle + \sqrt{1 - A_{c_{ij},j}^2}|1\rangle\right). \end{aligned} \tag{11}$$

Then

$$\begin{aligned} W &= (U_L \otimes I \otimes I)^\dagger U_R \\ &= U_1(\mathcal{O}_R^{-1} \otimes I \otimes I)(\mathcal{O}_A^{-1} \otimes I)U_2(\mathcal{O}_A \otimes I)(\mathcal{O}_C \otimes I \otimes I)U_3 \\ &= \begin{pmatrix} A/s & \cdot \\ \cdot & \cdot \end{pmatrix} \end{aligned}$$

is a block encoding of A, where U_1, U_2, U_3 are some unitaries independent of the oracles. As a result, if using this W, then the algorithm described in (9) is in the query model.

5.1 Known results in these models

It is possible that the oracles $\mathcal{O}_A, \mathcal{O}_R, \mathcal{O}_C$ or the block-encoding W can be used in some ways that are very different from (7), (9) in a quantum algorithm, but most quantum algorithms for matrix problems known so far are in these two models. We below present some examples to support this.

5.1.1 Quantum query algorithms for Boolean functions

Let f be a Boolean function. Given oracle access to $(x_1, \ldots, x_n) \in \{-1, 1\}^n$, the goal is to compute $f(x_1, \ldots, x_n)$. A commonly used oracle is $\mathcal{O} : |i\rangle|b\rangle \mapsto |i\rangle|b \oplus x_i\rangle$, and a quantum query algorithm for evaluating $f(x_1, \ldots, x_n)$ works as the form $U_T \mathcal{O} U_{T-1} \mathcal{O} \cdots U_2 \mathcal{O} U_1 |0\rangle$. This fits into the model described in (7). Here we only need to view $(x_1, \ldots, x_n)$ as the diagonal matrix $\mathrm{diag}(x_1, \ldots, x_n)$.

5.1.2 Quantum eigenvalue transformation

Let A be a p-qubit operator, $\alpha, \varepsilon \in \mathbb{R}^+$ and $q \in \mathbb{N}$, then we say that the $(p+q)$-qubit unitary W is an (α, q, ε)-block-encoding of A, if $\|A - \alpha(\langle 0|^p \otimes I) W (|0\rangle^p \otimes I)\| \leq \varepsilon$. Namely, up to some error ε, W has the form (8).

Block-encoding is a building block of the quantum singular value transformation [9]. We below state it for Hermitian matrices. Let $P(x)$ be a polynomial of degree d. By [9, Theorems 17 and 56], we can construct a $(1, q+2, 4d\sqrt{\varepsilon/\alpha})$ block encoding $\widetilde{W}$ of $P(A/\alpha)$ as follows.

- If $P(x)$ is an even polynomial, then there exist unitaries $U_1, \ldots, U_d$ at most depending on $P(x)$ such that

$$\widetilde{W} = \prod_{k=1}^{d/2} \Big(U_{2k-1} W^\dagger U_{2k} W \Big). \tag{12}$$

- If $P(x)$ is an odd polynomial, then there exist unitaries $W_1, \ldots, W_d$ at most depending on $P(x)$ such that

$$\widetilde{W} = U_1 W \prod_{k=1}^{(d-1)/2} \Big(U_{2k} W^\dagger U_{2k+1} W \Big). \tag{13}$$

- Generally, denote $\ell = \lfloor d/2 \rfloor$. Then there exist unitaries $U_1, U_2, \ldots, U_{2\ell+2}$ at most depending on $P(x)$ such that

$$\widetilde{W} = U_1 \begin{pmatrix} I & 0 \\ 0 & W \end{pmatrix} \left(\prod_{k=1}^{\ell} U_{2k} (I \otimes W^\dagger) U_{2k+1} (I \otimes W) \right) U_{2\ell+2}. \tag{14}$$

As a block-encoding, we have

$$\widetilde{W} \approx \begin{pmatrix} P(A/\alpha) & \cdot \\ \cdot & \cdot \end{pmatrix}.$$

For any given state $|\mathbf{b}\rangle = V|0\cdots 0\rangle$, applying $\widetilde{W}$ to $|0\rangle|\mathbf{b}\rangle$ gives $|\psi\rangle = |0\rangle \otimes P(A/\alpha)|\mathbf{b}\rangle + \beta|1\rangle \otimes |G\rangle$, where β is the amplitude and $|G\rangle$ is some garbage state. To obtain the state $|P(A/\alpha)\mathbf{b}\rangle$, we can apply amplitude amplification. A key operator that will be used many times to increase the success probability is the reflection $2|\psi\rangle\langle\psi| - I = \widetilde{W}(2|0\rangle|\mathbf{b}\rangle\langle 0|\langle \mathbf{b}| - I)\widetilde{W}^\dagger$. Now it is not hard to see that the above quantum algorithm that prepares the state $|P(A/\alpha)\mathbf{b}\rangle$ lies in the model described in (9). The uniatries $U_1, U_2, \ldots, U_T$ are independent of W but may depend on unitary V that prepares the state $|\mathbf{b}\rangle$.

5.1.3 Sparse Hamiltonian simulation and its applications

The oracles (4), (5), (6) for sparse matrices were used by Berry and Childs [27] to do Hamiltonian simulation. Their quantum algorithm fits into the model (7).[1] The key idea of their quantum algorithm is using quantum phase estimation to a discrete-time quantum walk described by a unitary $V = iS(2TT^\dagger - I)$, where $S|j,k\rangle = |k,j\rangle$ and $T = \sum_j |j\rangle|\phi_j\rangle\langle j|$. Here if the Hamiltonian is $H = (H_{jk})$, then $|\phi_j\rangle = \sqrt{\frac{\varepsilon}{\Lambda_1}} \sum_k \sqrt{H^*_{jk}}|k\rangle|0\rangle + \sqrt{1 - \frac{\varepsilon\sigma_j}{\Lambda_1}}|\zeta_j\rangle|1\rangle$, where Λ_1 is a known upper bound of $\|H\|_1$, $\sigma_j = \sum_k |H_{jk}|$, $|\zeta_j\rangle$ is some garbage state, and $\varepsilon \in (0,1]$ is a small parameter that ensures a lazy quantum walk. The state $|\phi_j\rangle$ can be prepared using $O(1)$ calls to the oracles $\mathcal{O}_A, \mathcal{O}_R, \mathcal{O}_C$, which is similar to the process (11). This is indeed how the oracles are used in their whole quantum algorithm for Hamiltonian simulation. So it fits into the model (7). Hamiltonian simulation is a useful subroutine of many quantum algorithms for linear algebra problems. A famous example is the HHL algorithm for linear systems of equations [11]. As a result, all these algorithms found so far based on [27] also fit into the model (7).

5.2 Connections to quantum communication complexity

Similar to the classical setting, we can also build connections between quantum protocols and quantum query algorithms for matrix problems. In the quantum case, we will mainly focus on the case that $k = 2$, i.e., the Alice-Bob model. It turns out that this is enough for us. One key reason is that in the quantum case, the cost will be roughly kT when there are k players. This is in sharp contrast to $k + T$ in the classical case (see Theorem 10). So we focus on the following simplified setting: Suppose Alice has a matrix $A^{(1)} \in \mathbb{R}^{m_1 \times n}$ and a vector $\mathbf{b}^{(1)} \in \mathbb{R}^{m_1}$, Bob has

[1] Although not in the same model, the quantum algorithm for recommendation systems also has a similar structure [28].

another matrix $A^{(2)} \in \mathbb{R}^{m_2 \times n}$ and another vector $\mathbf{b}^{(2)} \in \mathbb{R}^{m_2}$. Let $f(x)$ be a univariate function. Their goal is for Alice or Bob to output the state $|f(A)\mathbf{b}\rangle$, where

$$A = \begin{pmatrix} A^{(1)} \\ A^{(2)} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \mathbf{b}^{(1)} \\ \mathbf{b}^{(2)} \end{pmatrix}. \tag{15}$$

Proposition 18. *Assume that $A \in \mathbb{C}^{m \times n}$ has the decomposition (15). For Alice or Bob to use the oracle $\mathcal{O}_A, \mathcal{O}_R$ or $\mathcal{O}_C$ once, they need to communicate $O(\log(mn))$ qubits.*

Proof. Let $\sum_{i,j} \alpha_{ij}|i,j\rangle|0\rangle$ be a quantum state in Alice's hand. To implement $\mathcal{O}_A$, she first applies $\mathcal{O}_{A^{(1)}}$ in her hand to the state to obtain $\sum_{i \le m_1} \sum_j \alpha_{ij}|i,j\rangle|A_{ij}\rangle + \sum_{i > m_1} \sum_j \alpha_{ij}|i,j\rangle|0\rangle$. She then sends the state to Bob who applies $\mathcal{O}_{A^{(2)}}$ to it to obtain $\sum_{i,j} \alpha_{ij}|i,j\rangle|A_{ij}\rangle$. Bob then sends the state back to Alice. This costs $2\log(mn)$ qubits of communication. The protocol for using $\mathcal{O}_R$ is similar.

Next, we consider how to implement $\mathcal{O}_C$. For $i \in \{1,2\}$, denote the number of nonzero entries of j-th column of $A^{(i)}$ as $s_j^{(i)}$. So Alice knows $\{s_1^{(1)}, \ldots, s_n^{(1)}\}$ and Bob knows $\{s_1^{(2)}, \ldots, s_n^{(2)}\}$. For any state $\sum_{i,j} \alpha_{ij}|i\rangle|j\rangle$ in Alice's hand, she first queries the indices in superposition as follows $\sum_{i,j} \alpha_{ij}|i\rangle|j\rangle|s_j^{(1)}\rangle = \sum_{i \le s_j^{(1)}} \alpha_{ij}|i\rangle|j\rangle|s_j^{(1)}\rangle + \sum_{i > s_j^{(1)}} \alpha_{ij}|i\rangle|j\rangle|s_j^{(1)}\rangle$. Next, she applies $\mathcal{O}_C$ for $A^{(1)}$ to this state to obtain $\sum_{i \le s_j^{(1)}} \alpha_{ij}|c_{ij}\rangle|j\rangle|s_j^{(1)}\rangle + \sum_{i > s_j^{(1)}} \alpha_{ij}|i + m_1\rangle|j\rangle|s_j^{(1)}\rangle$. Alice then sends the state to Bob and asks him to do similar operations for the second term. Here for the second term, Bob has to query the $(i - s_j^{(1)})$-th nonzero entry based on the oracle he has. In the end, he sends the state back to Alice who undoes the query of $s_j^{(1)}$. This uses $2(2\log m + \log n)$ qubits of communication in total. □

From the above proof, if there are k-players and a coordinator, then for the coordinator to use the oracles once, they need $O(k\log(mn))$ qubits of communication. The following result follows from step 1 of [15, Proof of Proposition 14].

Proposition 19. *For each $i \in \{1,2\}$, let W_i be an $(\alpha_i, 1, 0)$ block-encoding of $A^{(i)} \in \mathbb{R}^{m_i \times n}$. Then there is a quantum protocol for Alice or Bob to use an $(\sqrt{\alpha_1^2 + \alpha_2^2}, 2, 0)$ block-encoding of A once with quantum communication complexity $O(\log(mn))$.*

With the above two propositions, it is not hard to obtain the following result.

Theorem 20. *If there is a quantum algorithm of the form (7) or (9) of complexity $O(T)$, then there is a quantum protocol for the same task with quantum communication complexity $O(T\log(mn))$.*

5.3 Applications: Lower bounds analysis of quantum query algorithms

In this part, we present some applications of Theorem 20. We will focus on a further simplified case than the setting (15). Here suppose Alice has a matrix A and Bob has a state $|\mathbf{b}\rangle$, their goal is to prepare the state $|f(A)\mathbf{b}\rangle$ by Alice or Bob, where f is a public function. In the model (7) or (9), we now assume that the unitaries $U_1, U_2, \ldots, U_T$ can depend on the unitary that prepares the state $|\mathbf{b}\rangle$. We below use Theorem 20 to prove certain lower bounds in the quantum query model, either the number of queries or the number of block-encodings we must use. We can even obtain lower bounds with respect to the number of copies of $|\mathbf{b}\rangle$. To be more exact, if it is Alice to output $|f(A)\mathbf{b}\rangle$, then there is no need to communicate the oracles or the block-encoding of A. What she needs is $U_1, U_2, \ldots, U_T$, which may depend on $|\mathbf{b}\rangle$. So we will obtain a lower bound on the number of copies of $|\mathbf{b}\rangle$. If Bob is the one to output $|f(A)\mathbf{b}\rangle$, then we will obtain a lower bound on the number of calls of the block-encoding of A, or the number of calls of queries to A.

Proposition 21 (Lower bounds for matrix inversion)**.** *Any quantum algorithm in the form (7) or (9) that prepares $|A^+\mathbf{b}\rangle$ requires making $\Omega(\kappa/\log(mn))$ calls to the oracles or block-encoding of A, and requires $\Omega(\kappa/\log(mn))$ copies of the state $|\mathbf{b}\rangle$.*

Proof. The proof is inspired by [16, Theorem 10 (3)]. Let Alice have $(a_1, \ldots, a_n) \in \{0,1\}^n$ and Bob have $(b_1, \ldots, b_n) \in \{0,1\}^n$. Alice constructs a diagonal matrix A by setting $A_{ii} = 1/\sqrt{n}$ if

$a_i = 1$ and 1 if $a_i = 0$. Bob constructs a quantum state $|\mathbf{b}\rangle \propto \sum_{b_i=1} |i\rangle$. Then the quantum state $|A^{-1}\mathbf{b}\rangle \propto \sqrt{n}\sum_{a_i=b_i=1} |i\rangle + \sum_{a_i=0,b_i=1} |i\rangle$. When there is only one intersection, then we will set index i with $a_i = b_i = 1$ with probability close to 1/2. So if we have the state $|A^{-1}\mathbf{b}\rangle$, we then can solve the Set-Disjointness problem using constantly many invocations of the above, whose communication complexity is $\Omega(\sqrt{n}) = \Omega(\kappa)$. By Theorem 20, we obtain the claimed lower bound. □

Apart from the $\log(mn)$ factor, the lower bound obtained above coincides with known lower bound for matrix inversion [11]. Note that in the query model, there is a similar proof based on Grover's algorithm [29]. Let $(x_1, \ldots, x_n) \in \{0,1\}^n$. Assume that there is an i_0 such that $x_{i_0} = 1$ and $x_j = 0$ for all $j \neq i_0$. It is known that the quantum query complexity of finding i_0 is $\Theta(\sqrt{n})$. Let A be a diagonal matrix with $A_{ii} = 1/\sqrt{n}$ if $i = i_0$ and $A_{ii} = 1$ if $i \neq i_0$. Let $|\mathbf{b}\rangle = \frac{1}{\sqrt{n}}\sum_i |i\rangle$. Then the quantum state $|A^{-1}\mathbf{b}\rangle = \frac{1}{\sqrt{2n-1}}(\sqrt{n}|i_0\rangle + \sum_{j\neq i_0} |j\rangle)$. If we can prepare this state, then we can solve the search problem by measuring $|A^{-1}\mathbf{b}\rangle$. For this problem, the condition number $\kappa = \sqrt{n}$. So it gives a lower bound $\Omega(\kappa)$ of query complexity for the matrix inversion problem. Although the lower bound in Proposition 21 is weaker by a factor of $\log(mn)$, it gives a lower bound on the number of copies of the state $|\mathbf{b}\rangle$ at the same time. Moreover, the lower bounds also hold in the block-encoding model.

Proposition 22 (Lower bounds for matrix exponential)**.** *Any quantum algorithm in the form (7) or (9) that prepares $|e^{At}\mathbf{b}\rangle$ requires making $\Omega(e^t/\log(mn))$ calls to the oracles or block-encoding of A, and requires $\Omega(e^t/\log(mn))$ copies of the state $|\mathbf{b}\rangle$.*

Proof. The proof is similar to the previous one. Imagine that Alice has $\mathbf{a} \in \{0,1\}^n$ and Bob has $\mathbf{b} \in \{0,1\}^n$. Alice constructs a diagonal matrix A using $\mathbf{a}$. Then

$$e^{At}\mathbf{b} = \sum_{i=1}^{n} e^{a_i t} b_i |i\rangle = \sum_{a_i=b_i=1} e^t |i\rangle + \sum_{a_i=0,b_i=1} |i\rangle.$$

Let $t = \frac{1}{2}\ln n$. If we can prepare $|e^{At}\mathbf{b}\rangle$, then sampling from it returns i such that $a_i = b_i = 1$ with probability close to 1/2. So, just like in the previous proof, we can solve the Set-Disjointness problem using this routine constantly many times. As the lower bound of the Set-Disjointness problem is $\Omega(\sqrt{n})$ and $e^t = \sqrt{n}$, by Theorem 20, we obtain a lower bound of $\Omega(e^t/\log(mn))$ for preparing the state $|e^{At}\mathbf{b}\rangle$. □

Recall that in [30, Proposition 14], An et al. showed that the number of copies of $|\mathbf{b}\rangle$ required to preparing state $|e^{At}\mathbf{b}\rangle$ is roughly $\Omega(e^t)$. In the above, we gave a simple proof of this and also proved the same lower bound on the query complexity with respect to A at the same time.

Proposition 23 (Lower bounds for matrix powers)**.** *Any quantum algorithm in the form (7) or (9) that prepares $|A^d\mathbf{b}\rangle$ requires making $\Omega((\kappa^d + 1/\gamma)/\log(mn))$ calls to the oracles or block-encoding of A, and requires $\Omega((\kappa^d + 1/\gamma)/\log(mn))$ copies of the state $|\mathbf{b}\rangle$.*

Proof. We will use similar constructions as the above proofs. Alice constructs a diagonal matrix A with diagonal entry $A_{ii} = 1$ if $a_i = 1$ and $A_{ii} = \varepsilon$ if $a_i = 0$. Bob uses this input to define the state $|\mathbf{b}\rangle$. Then the state $|A^d\mathbf{b}\rangle$ is proportional to $\sum_{a_i=b_i=1} |i\rangle + \varepsilon^d \sum_{a_i=0,b_i=1} |i\rangle$. We will choose ε such that $\varepsilon^{2d} n = 1$. So we can solve the Set-Disjointness problem with high probability by measuring this state. The condition number of A is $\kappa = 1/\varepsilon$ and so $\kappa^d = \sqrt{n}$. It equals the complexity of the Set-Disjointness problem. So we obtain a lower bound of $\Omega(\kappa^d)$. To prove $\Omega(1/\gamma)$, Alice sets $\varepsilon = 0$. Now $\kappa = 1$ and $\gamma = \|A^d\mathbf{b}\|/\|\mathbf{b}\| = 1/\sqrt{n}$. □

In [15], the authors studied the lower bound of quantum query complexity of computing $f(A)_{i,j}$, where A is sparse and f is a continuous function. They proved a low bound in terms of the approximate degree of $f(x)$. If $f(x) = x^d$, then the approximate degree is $\Theta(\sqrt{d})$ and so the lower bound of computing $A^d_{i,j}$ is $\Omega(\sqrt{d})$. However, this task is weaker than preparing the state $|A^d\mathbf{b}\rangle$. If we apply quantum singular value transform [9], then given the block-encoding of A, we can construct a block-encoding of A^d. To sample from $|A^d\mathbf{b}\rangle$, we need to do the postselection. The success probability depends on $\|A^d\mathbf{b}\| \geq 1/\kappa^d$ if assuming $\|A\| = 1$. This leads to a quantum algorithm of complexity $O(\kappa^d)$, which almost matches our lower bound.

6 Conclusions and open problems

In this work, we built a connection between quantum-inspired classical algorithms and communication complexity. In particular, we showed that quantum-inspired classical algorithms can be efficiently simulated using communication protocols. Using this connection, we proved some lower bounds of quantum-inspired classical algorithms for some matrix-relevant problems. We also generalised the idea to the quantum case.

The following are some open problems that we think merit further study.

1. The most exciting problem is to prove better lower bounds for the tasks mentioned in this work. We feel some new ideas are required. For example, the known results on communication complexity used in this paper are Set-Disjointness and Gap-Hamming problem in the coordinator model. If we have more candidate hard problems for communication in this model, we may use them to obtain better lower bounds. It is also possible that more efficient quantum-inspired classical algorithms exist such that the lower bounds obtained in this paper are close to tight. So the exploration of more efficient quantum-inspired classical algorithms is also very important.

2. The quantum query complexity of functions of matrices is widely studied in the past [9, 15], see the references in [15] for a summary. In this work, we provided a new method to prove lower bounds of quantum query complexity, so it is interesting to see if some nontrivial new lower bounds can be obtained.

3. In this paper, we only used the connection on one side, i.e., proving lower bounds. As discussed at the end of Section 3, via the connection, we can also propose efficient protocols for many matrix-relevant problems from quantum-inspired classical algorithms we know so far, e.g., [2, 4, 5]. So it is interesting to see some useful applications of this method in communication complexity. Theorem 28 might be helpful in this direction.

Acknowledgements

This research is supported by the National Key Research Project of China under Grant No. 2023YFA1009403.

A Another connection

Apart from the connection discussed in Section 3, another way to see the connection, which should be more general in practice, is considering $A = \sum_{i=1}^{k} \lambda_i A^{(i)}$ and $\mathbf{b} = \sum_{i=1}^{k} \mu_i \mathbf{b}^{(i)}$. First, we list some definitions.

Definition 24. *Let $p = (p_1, \ldots, p_n), q = (q_1, \ldots, q_n) \in \mathbb{R}^n_{\geq 0}$ be two distributions, i.e., $\sum p_i = \sum q_i = 1$ we say p ϕ-oversamples q if for all $i \in [n]$, $p_i \geq q_i/\phi$.*

Definition 25 (Oversampling and query access to a vector)**.** *Let $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{C}^n$ and $\phi \geq 1$, we have $SQ_\phi(\mathbf{v})$, ϕ-oversampling and query access to $\mathbf{v}$, if we have $Q(\mathbf{v})$ and $SQ(\tilde{\mathbf{v}})$ for some vector $\tilde{\mathbf{v}} = (\tilde{v}_1, \ldots, \tilde{v}_n) \in \mathbb{C}^n$ satisfying $\|\tilde{\mathbf{v}}\|^2 = \phi\|\mathbf{v}\|$ and $|\tilde{v}_i|^2 \geq |v_i|^2$ for all $i \in [n]$.*

Definition 26 (Oversampling and query access to a matrix)**.** *Let $A = (A_{ij}) \in \mathbb{C}^{m\times n}$ and $\phi \geq 1$, we have $SQ_\phi(\mathbf{v})$, ϕ-oversampling and query access to $\mathbf{v}$, if we have $Q(A)$ and $SQ(\widetilde{A})$ for some matrix $\widetilde{A} = (\widetilde{A}_{ij}) \in \mathbb{C}^{m\times n}$ satisfying $\|\widetilde{A}\|_{\mathrm{F}}^2 = \phi\|A\|_{\mathrm{F}}^2$ and $|\widetilde{A}_{ij}|^2 \geq |A_{ij}|^2$ for all $(i, j) \in [m] \times [n]$.*

Lemma 27 (Lemma 3.5 of [2])**.** *Suppose we are given $SQ_\phi(\mathbf{v})$ and $\delta \in (0, 1]$, then we can sample from $D_{\mathbf{v}}$ with probability $\geq 1 - \delta$ with $O(\phi \log(1/\delta))$ applications of $SQ_\phi(\mathbf{v})$. We can also estimate $\|\mathbf{v}\|$ to ε multiplicative error for $\varepsilon \in (0, 1]$ with probability $\geq 1 - \delta$ with $O(\varepsilon^{-2})$ applications of $SQ_\phi(\mathbf{v})$.*

Theorem 28. *Assume that player $\mathcal{P}_i$ holds a matrix $A^{(i)} \in \mathbb{R}^{m\times n}$ and a vector $\mathbf{b}^{(i)} \in \mathbb{R}^m$ whose entries are specified by $O(\log q)$ bits, where $i \in [k]$. Let $A = \sum_i \lambda_i A^{(i)}$ and $\mathbf{b} = \sum_i \mu_i \mathbf{b}^{(i)}$. Then in the coordinator model, we have the following.*

- *For the coordinator to use $SQ(\mathbf{b})$ $O(T)$ times, it costs $O(kT\phi_b \log(qm))$ bits of communication, where*
$$\phi_b = k\frac{\sum_i \|\mu_i \mathbf{b}^{(i)}\|^2}{\|\mathbf{b}\|^2}.$$
- *For the coordinator to use $SQ(A)$ $O(T)$ times, it costs $O(kT\phi_A \log(qmn))$ bits of communication, where*
$$\phi_A = k\frac{\sum_i \|\lambda_i A^{(i)}\|_{\mathrm{F}}^2}{\|A\|_{\mathrm{F}}^2}.$$

In comparison, Theorem 10 has a lower complexity than the one stated in Theorem 28. This is the main reason why we choose to use Theorem 10 to prove lower bounds. However, the case discussed in Theorem 28 seems more powerful since, for example, the linear regression $\arg\min \|A\mathbf{x}-\mathbf{b}\|$ with $A, \mathbf{b}$ given in (2) is equivalent to $\arg\min_{\mathbf{x}} \|\sum_i (A^{(i)})^{\mathrm{T}} A^{(i)}\mathbf{x} - \sum_i (A^{(i)})^{\mathrm{T}}\mathbf{b}^{(i)}\|$, which is a special case of the one discussed above. We feel Theorem 28 can play more important roles in designing efficient classical protocols for problems, such as machine learning problems or Hamiltonian simulation.

Proof. First, we show how the coordinator uses $SQ(\mathbf{b})$. We will use $b_j^{(i)}$ and b_j to denote the j-th entry of $\mathbf{b}^{(i)}$ and $\mathbf{b}$ respectively. Given $SQ(\mathbf{b}^{(i)})$ for all $i \in [k]$, we can construct $SQ_{\phi_b}(\mathbf{b})$ (see Definition 25 for SQ_{ϕ_b}) in the following way, which is based on the proof of [2, Lemma 3.6] and also provides a protocol for the coordinator to use $SQ(\mathbf{b})$.

1. Obtain $Q(\mathbf{b})$: For each $j \in [m]$, we can compute $b_j = \sum_{i=1}^k \mu_i b_j^{(i)}$ by querying all $b_j^{(i)}$.
2. Obtain $SQ(\tilde{\mathbf{b}})$: Define $\tilde{\mathbf{b}}$ via
$$\tilde{b}_j = \sqrt{k\sum_{i=1}^k |\mu_i b_j^{(i)}|^2}, \quad j \in [m].$$
It is easy to check that $\|\tilde{\mathbf{b}}\|^2 = \phi_b \|\mathbf{b}\|^2$ and $|\tilde{b}_j| \geq |b_j|$ for all $j \in [m]$.
   (a) Query for $\|\tilde{\mathbf{b}}\|^2$: We can compute $\|\tilde{\mathbf{b}}\|^2 = k\sum_i \|\mu_i \mathbf{b}^{(i)}\|^2$ directly by querying $\|\mathbf{b}^{(i)}\|^2$ for all $i \in [k]$.
   (b) Query for $\tilde{b}_j$: We can compute $\tilde{b}_j$ by querying $b_j^{(i)}$ for all $i \in [k]$.
   (c) Sample from $\tilde{\mathbf{b}}$: We sample an index $i \in [k]$ with probability $\|\mu_i \mathbf{b}^{(i)}\|^2 / \sum_j \|\mu_j \mathbf{b}^{(j)}\|^2$, then take a sample $j \in [m]$ from the distribution defined by $\mathbf{b}^{(i)}$. We can check that the probability is
$$\sum_{i=1}^k \frac{\|\mu_i \mathbf{b}^{(i)}\|^2}{\sum_j \|\mu_j \mathbf{b}^{(j)}\|^2} \frac{|b_j^{(i)}|^2}{\|\mathbf{b}^{(i)}\|^2} = \frac{\sum_{i=1}^k |\mu_i b_j^{(i)}|^2}{\sum_j \|\mu_j \mathbf{b}^{(j)}\|^2} = \frac{|\tilde{b}_j|^2}{\|\mathbf{b}\|^2}.$$

The above procedure provides a protocol for the coordinator to use $SQ_{\phi_b}(\mathbf{b})$, from which the coordinator can construct $SQ(\mathbf{b})$. Firstly, querying for entries of $\mathbf{b}$ is straightforward, which costs $O(k \log q)$ bits of communication. Secondly, query for $\|\tilde{\mathbf{b}}\|^2$ and $\tilde{b}_j$ are also straightforward, which costs $O(k \log q)$ bits of communication. Thirdly, to sample from $\tilde{\mathbf{b}}$, player $\mathcal{P}_i$ first sends $\|\tilde{\mathbf{b}}^{(i)}\|$ to the coordinator. This costs $O(k \log(qm))$ bits of communication. Now the coordinator can do step (2.c) discussed in the above procedure with $O(\log q)$ bits of communication with player $\mathcal{P}_i$. In summary, for the coordinator to use $SQ_{\phi_b}(\mathbf{b})$ $O(T)$ times, they use $O(T \log q + kT \log(qm)) = O(kT \log(qm))$ bits of communication. Combining Lemma 27, it costs $O(kT\phi_b \log(qm))$ bits of communication for the coordinator to use $SQ(\mathbf{b})$ $O(T)$ times.

Similar results also hold for linear combinations of matrices. Given $SQ(A^{(i)})$ for all $i \in [k]$, we can construct $SQ_\phi(A)$ in the following way, which is based on the proof of [2, Lemma 3.9]. Below for any matrix M, we use $M_{i,j}$ to denote its (i,j)-th entry and use M_{i*} to denote the i-th row.

1. Obtain $Q(A)$: For each $(i,j) \in [m] \times [n]$, we can compute $A_{i,j} = \sum_{t=1}^{k} \lambda_t A_{i,j}^{(t)}$ by querying all $A_{i,j}^{(t)}$.

2. Obtain $SQ(\tilde{A})$: Define $\tilde{A}$ via

$$\tilde{A}_{i,j} = \sqrt{k \sum_{t=1}^{k} |\lambda_t A_{i,j}^{(t)}|^2},$$

It is easy to check that $\|\tilde{A}\|_{\mathrm{F}}^2 = \phi_A \|A\|_{\mathrm{F}}^2, |\tilde{A}_{ij}| \geq |A_{ij}|$ for all $(i,j) \in [m] \times [n]$.

   (a) Query for $\|\tilde{A}\|_{\mathrm{F}}^2$: We can compute $\|\tilde{A}\|_{\mathrm{F}}^2 = \sum_{t=1}^{k} \|\lambda_t A^{(t)}\|_{\mathrm{F}}^2$ directly by querying $\|A^{(i)}\|_{\mathrm{F}}^2$ for all $i \in [k]$.

   (b) Query for $\tilde{A}_{i,j}$: We can compute $\tilde{A}_{i,j}$ by querying $A_{i,j}^{(t)}$ for all $t \in [k]$.

   (c) Sample from i-th row $\tilde{A}_{i*}$: We first sample an index $s \in [k]$ with probability $\frac{\|\lambda_s A_{i*}^{(s)}\|^2}{\sum_{t=1}^{k} \|\lambda_t A_{i*}^{(t)}\|^2}$, then take a sample $j \in [m]$ from $A_{i*}^{(s)}$. The probability of seeing j is

$$\sum_{s=1}^{k} \frac{\|\lambda_s A_{i*}^{(s)}\|^2}{\sum_{t=1}^{k} \|\lambda_t A_{i*}^{(t)}\|^2} \frac{|A_{i,j}^{(s)}|^2}{\|A_{i*}^{(s)}\|^2} = \frac{\sum_{s=1}^{k} |\lambda_s A_{i,j}^{(s)}|^2}{\sum_{t=1}^{k} \|\lambda_t A_{i*}^{(t)}\|^2} = \frac{|\widetilde{A}_{ij}|^2}{\|\widetilde{A}_{i*}\|^2}.$$

   (d) Obtain $SQ(\mathbf{a})$ where $\tilde{\mathbf{a}} = (\|\tilde{A}_{1*}\|, \ldots, \|\tilde{A}_{m*}\|)$: Note that $\|\tilde{A}_{i*}\|^2 = k \sum_t \|\lambda_t \tilde{A}_{i*}^{(t)}\|^2$. So we can compute it by querying $\|\tilde{A}_{i*}^{(t)}\|^2$ for all $t \in [k]$. To sample from $\tilde{\mathbf{a}}$, we first sample $s \in [k]$ with probability $\|\lambda_s A^{(s)}\|_{\mathrm{F}}^2 / \sum_t \|\lambda_t A^{(t)}\|_{\mathrm{F}}^2$, then take a sample $i \in [m]$ from the distribution defined by $\mathbf{a}^{(s)} = (\|A_{1*}^{(s)}\|, \ldots, \|A_{m*}^{(s)}\|)$. The probability is

$$\sum_{s} \frac{\|\lambda_s A^{(s)}\|_{\mathrm{F}}^2}{\sum_t \|\lambda_t A^{(t)}\|_{\mathrm{F}}^2} \frac{\|A_{i*}^{(s)}\|^2}{\|A^{(s)}\|_{\mathrm{F}}^2} = \frac{\sum_s \|\lambda_s A_{i*}^{(s)}\|^2}{\sum_t \|\lambda_t A^{(t)}\|_{\mathrm{F}}^2} = \frac{\|\tilde{A}_{i*}\|^2}{\|\tilde{A}\|_{\mathrm{F}}^2}.$$

The analysis for the coordinator to use $SQ_\phi(A)$ is similar. For the coordinator to query $A_{i,j}, \tilde{A}_{i,j}$, it costs $O(k \log q)$ bits of communication. To query $\|\tilde{A}\|_{\mathrm{F}}^2 = k \sum_i \|\lambda_i A^{(i)}\|_{\mathrm{F}}^2$, it costs $O(k \log(qmn))$ bits of communication. Here $O(\log(qmn))$ is the number of bits used to specify $\|A^{(i)}\|_{\mathrm{F}}^2$. To query for $\|\tilde{A}_{i*}\|^2 = k \sum_t \|\lambda_t A_{i*}^{(t)}\|^2$, it costs $O(k \log(qn))$ bits of communication. To sample from $\widetilde{A}_{i*}$, each player $\mathcal{P}_i$ first sends $\|A_{i*}^{(s)}\|$ to the coordinator, this uses $O(k(\log q + \log n))$ bits of communication. To sample from $\tilde{\mathbf{a}}$, since the coordinator already has the information of $\|A^{(i)}\|_{\mathrm{F}}^2$ for all $i \in [k]$, this step now only costs $O(\log q)$ bits of communication (namely communicate with player $\mathcal{P}_i$ once for some i). Finally, to sample from $\tilde{A}_{i*}$, each player needs to send $\|A_{i*}^{(s)}\|$ to the coordinator, then the coordinator has the distribution of seeing s with probability $\|\lambda_s A_{i*}^{(s)}\|^2 / \sum_t \|\lambda_t A_{i*}^{(t)}\|^2$, this uses $O(\log(qn))$ bits of communication. Thus for the coordinator to use $SQ_{\phi_A}(A)$ $O(T)$ times, it costs $O(T \log(qn) + kT \log(qmn))$ bits of communication. By Lemma 27, it costs $O(kT\phi_A \log(qmn))$ bits of communication for the coordinator to use $SQ(A)$ $O(T)$ times. □

References

[1] Ewin Tang. "A quantum-inspired classical algorithm for recommendation systems". In Proceedings of the 51st annual ACM SIGACT Symposium on Theory of Computing. Pages 217–228. (2019).

[2] Nai-Hui Chia, András Pal Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang. "Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning". Journal of the ACM **69**, 1–72 (2022).

[3] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. "The Power of Block-Encoded Matrix Powers: Improved Regression Techniques via Faster Hamiltonian Simulation". In 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019). Volume 132, pages 33:1–33:14. (2019).

[4] Ainesh Bakshi and Ewin Tang. "An improved classical singular value transformation for quantum machine learning". In Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). Pages 2398–2453. SIAM (2024).

[5] Changpeng Shao and Ashley Montanaro. "Faster quantum-inspired algorithms for solving linear systems". ACM Transactions on Quantum Computing **3**, 1–23 (2022).

[6] András Gilyén, Zhao Song, and Ewin Tang. "An improved quantum-inspired algorithm for linear regression". Quantum **6**, 754 (2022).

[7] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. "Quantum algorithms for supervised and unsupervised machine learning" (2013). arXiv:1307.0411.

[8] Iordanis Kerenidis and Anupam Prakash. "Quantum recommendation systems". In 8th Innovations in Theoretical Computer Science Conference (ITCS 2017). Page 49:1–49:21. (2017).

[9] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics". In Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing. Pages 193–204. (2019).

[10] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. "Quantum principal component analysis". Nature Physics **10**, 631–633 (2014).

[11] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum algorithm for linear systems of equations". Physical Review Letters **103**, 150502 (2009).

[12] Ewin Tang. "Quantum principal component analysis only achieves an exponential speedup because of its state preparation assumptions". Physical Review Letters **127**, 060503 (2021).

[13] András Gilyén, Seth Lloyd, and Ewin Tang. "Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension" (2018). arXiv:1811.04909.

[14] Sevag Gharibian and Fran**c**cois Le Gall. "Dequantizing the quantum singular value transformation: hardness and applications to quantum chemistry and the quantum pcp conjecture". In Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing. Pages 19–32. (2022).

[15] Ashley Montanaro and Changpeng Shao. "Quantum and classical query complexities of functions of matrices". In Proceedings of the 56th Annual ACM Symposium on Theory of Computing. Pages 573–584. (2024).

[16] Ashley Montanaro and Changpeng Shao. "Quantum communication complexity of linear regression". ACM Transactions on Computation Theory **16**, 1–30 (2024).

[17] Andrew Chi-Chih Yao. "Some complexity questions related to distributive computing (preliminary report)". In Proceedings of the 11th annual ACM Symposium on Theory of Computing. Pages 209–213. (1979).

[18] Ronald de Wolf. "Quantum communication and complexity". Theoretical Computer Science **287**, 337–353 (2002).

[19] Anup Rao and Amir Yehudayoff. "Communication complexity: and applications". Cambridge University Press. Cambridge, U.K. (2020).

[20] Jeff M Phillips, Elad Verbin, and Qin Zhang. "Lower bounds for number-in-hand multiparty communication complexity, made easy". In Proceedings of the twenty-third annual ACM-SIAM Symposium on Discrete Algorithms (SODA). Pages 486–501. SIAM (2012).

[21] Bala Kalyanasundaram and Georg Schintger. "The probabilistic communication complexity of set intersection". SIAM Journal on Discrete Mathematics **5**, 545–557 (1992).

[22] Amit Chakrabarti and Oded Regev. "An optimal lower bound on the communication complexity of gap-hamming-distance". In Proceedings of the forty-third annual ACM Symposium on Theory of Computing. Pages 51–60. (2011).

[23] David P Woodruff and Qin Zhang. "When distributed computation is communication expensive". Distributed Computing **30**, 309–323 (2017).

[24] Yi Li, Honghao Lin, and David Woodruff. "ℓ_p-regression in the arbitrary partition model of communication". In Gergely Neu and Lorenzo Rosasco, editors, Proceedings of Thirty Sixth Conference on Learning Theory. Volume 195 of Proceedings of Machine Learning Research, pages 4902–4928. PMLR (2023). url: https://proceedings.mlr.press/v195/li23b.html.

[25] Ashley Montanaro. "Quantum states cannot be transmitted efficiently classically". Quantum **3**, 154 (2019).

[26] Michael A Nielsen and Isaac L Chuang. "Quantum Computation and Quantum Information". Cambridge University Press. Cambridge, U.K. (2010).

[27] Dominic W Berry and Andrew M Childs. "Black-box Hamiltonian simulation and unitary implementation". Quantum Information and Computation **12**, 29–62 (2012).

[28] Iordanis Kerenidis and Anupam Prakash. "Quantum Recommendation Systems". In Christos H. Papadimitriou, editor, 8th Innovations in Theoretical Computer Science Conference (ITCS 2017). Volume 67 of Leibniz International Proceedings in Informatics (LIPIcs), pages 49:1–49:21. Dagstuhl, Germany (2017). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[29] Lov K Grover. "A fast quantum mechanical algorithm for database search". In Proceedings of the twenty-eighth annual ACM Symposium on Theory of computing. Pages 212–219. (1996).

[30] Dong An, Jin-Peng Liu, Daochen Wang, and Qi Zhao. "A theory of quantum differential equation solvers: limitations and fast-forwarding" (2022). arXiv:2211.05246.