Quantum Distributed Complexity of Set Disjointness on a Line *

Frédéric Magniez †

Université de Paris, IRIF, CNRS, France

Ashwin Nayak ‡

University of Waterloo, Canada

February 17, 2021

Abstract

Given $x, y \in \{0,1\}^n$, Set Disjointness consists in deciding whether $x_i = y_i = 1$ for some index $i \in [n]$. We study the problem of computing this function in a distributed computing scenario in which the inputs x and y are given to the processors at the two extremities of a path of length d. Each vertex of the path has a quantum processor that can communicate with each of its neighbours by exchanging $\mathrm{O}(\log n)$ qubits per round. We are interested in the number of rounds required for computing Set Disjointness with constant probability bounded away from $1/2$. We call this problem "Set Disjointness on a Line".

Set Disjointness on a Line was introduced by Le Gall and Magniez [LM18] for proving lower bounds on the quantum distributed complexity of computing the diameter of an arbitrary network in the CONGEST model. However, they were only able to provide a lower bound when the local memory used by the processors on the intermediate vertices of the path is severely limited. More precisely, their bound applies only when the local memory of each intermediate processor consists of $\mathrm{O}(\log n)$ qubits.

In this work, we prove an unconditional lower bound of $\widetilde{\Omega}\big(\sqrt[3]{nd^2} + \sqrt{n}\,\big)$ rounds for Set Disjointness on a Line with $d+1$ processors. This is the first non-trivial lower bound when there is no restriction on the memory used by the processors. The result gives us a new lower bound of $\widetilde{\Omega}\big(\sqrt[3]{n\delta^2} + \sqrt{n}\,\big)$ on the number of rounds required for computing the diameter δ of any n-node network with quantum messages of size $\mathrm{O}(\log n)$ in the CONGEST model.

We draw a connection between the distributed computing scenario above and a new model of query complexity. In this model, an algorithm computing a bi-variate function f (such as Set Disjointness) has access to the inputs x and y through two separate oracles $\mathcal{O}_x$ and $\mathcal{O}_y$, respectively. The restriction is that the algorithm is required to alternately make d queries to $\mathcal{O}_x$ and d queries to $\mathcal{O}_y$, with input-independent computation in between queries. The model reflects a "switching delay" of d queries between a "round"

*A preliminary version of this article appeared in the proceedings of ICALP 2020 [MN20]. Among other improvements, this article corrects an error in the statement of a result from prior work (Theorem 3.5) in the conference version, and makes corresponding changes in the rest of the article.

†IRIF, Université de Paris and CNRS, 75205 Paris Cedex 13, France. Email: magniez@irif.fr . Research supported in part by the ERA-NET Cofund in Quantum Technologies project QuantAlgo and the French ANR Blanc project RDAM.

‡Department of Combinatorics and Optimization, and Institute for Quantum Computing, University of Waterloo, 200 University Ave. W., Waterloo, ON, N2L 3G1, Canada. Email: ashwin.nayak@uwaterloo.ca . Research supported in part by NSERC Canada.

of queries to x and the following "round" of queries to y. The information-theoretic technique we use for deriving the round lower bound for Set Disjointness on a Line also applies to the number of rounds in this query model. We provide an algorithm for Set Disjointness in this query model with round complexity that matches the round lower bound stated above, up to a polylogarithmic factor. This presents a barrier for obtaining a better round lower bound for Set Disjointness on the Line, but leaves open the possibility of better communication protocols for the problem.

1 Introduction

1.1 Context

The field of Distributed Computing aims to model a collection of processors or computers communicating with each other over some network with the goal of collectively solving a global computational task. This task may depend on the structure of the network and on some additional data distributed among the computers. For instance, one may want to compute the distance between two nodes of the network, or its diameter, a proper colouring, a spanning tree, or even all-pairs shortest paths. In the context of cloud computing, data centres serve as special nodes of the network where data are stored. These centres are usually spread all over the world in order to minimise access time by clients. Since some operations need to be performed in order to synchronise the centres, the distance between these centres influence the quality of the network. For instance, one may want to decide if there is any inconsistency between two or more remote databases, or check for the availability of a common slot for booking some service.

In this work, we focus on the case of two remote data centres deployed on two nodes of a distributed network, and consider the problem of computing Set Disjointness. This fundamental problem, which we denote by D_n, consists in deciding whether two n-bit input strings x and y modelling two remote databases have the bit 1 at the same position. (This may indicate a schedule conflict, for instance.) The problem has been studied extensively in Communication Complexity [Yao79], due to its many applications in other contexts (see, for example, the survey by Chattopadhyay and Pitassi [CP10]). In the most basic setting, two remote parties, Alice and Bob, hold the inputs x and y, respectively. They communicate with each other directly in order to solve the problem, while minimising the total length of the messages exchanged. Depending upon the model of computation and the type of communication channel connecting the players, the messages may be deterministic, randomised, or quantum.

The two-party communication model may be too simplistic in some scenarios, since it assumes instantaneous communication and full access to the input (by the party that is "given" the input). To address the first issue, we may include the communication delay as a multiplicative factor in the communication complexity. However, this would not account for a potentially more sophisticated use of the communication channel between the two parties. Consider the case when the channel consists of a chain of d devices, say, repeaters. One could use the channel as a network of processors in order to minimise the communication delay, for instance using cached memories. With regard to the second issue—pertaining to access to the input—the standard two-party model may not be suitable when the inputs are massive, and may only be accessed in small parts. Such access is better modelled as in Query Complexity, in which inputs are accessed by querying *oracles* (see, e.g., Refs. [BdW02, dW19, Amb19]).

Motivated by a concrete problem in distributed computing, we define a new model of query complexity, two-oracle query complexity with a "switching delay". In this model we consider a single computer with access to two oracles, one for each input x or y, such that switching between queries to the two inputs involves a time delay d. The delay accounts for the lag in communication between the parties holding the inputs, for instance when the inputs are not physically at the same place. It might be advantageous to balance

this delay by making several accesses to the same input, say x, before switching to the other input y; we also incorporate this feature in the model. The new model attempts to address both the issues discussed above, and is described more precisely in Section 4.1.

There are several bridges between query complexity and communication complexity, but we are not aware of any previous work in a query model such as the one above. The two models—communication through a chain of d devices, and two-oracle query algorithms with a switching delay of d—share some similarities but also have fundamental differences. In the first model, one node has full access to half of the input. In the second model, all the information obtained so far from the inputs x and y is kept in the same memory registers, even when the algorithm switches between inputs.

In this work, we show that the above refinements of the two-party communication model and the query model differ significantly from their standard versions for solving Set Disjointness in the quantum setting. Such a difference does not occur in the setting of deterministic or randomised computing, and we do not know whether such a difference arises for another "natural" problem.

1.2 Application to quantum distributed computing

This study was initially motivated by a problem left open by Le Gall and Magniez [LM18] in the context of distributed computing with congestion (CONGEST model). They demonstrated the superiority of quantum computing for computing the diameter δ of a network with p nodes (Diameter problem). They designed a quantum distributed algorithm using $\widetilde{\mathrm{O}}(\sqrt{p\delta}\,)$ synchronised rounds, where simultaneous messages of $\mathrm{O}(\log p)$ qubits are exchanged at each round between neighbouring nodes in the network. They also established a lower bound of $\widetilde{\Omega}(\sqrt{p}+\delta)$ rounds.

Classically the congested distributed complexity of Diameter is well understood, and requires $\widetilde{\Theta}(p)$ rounds [HW12, PRT12, FHW12]. The lower bound is based on the construction of a two-party communication protocol for Set Disjointness from any distributed algorithm for Diameter. From n-bit inputs x, y, two pieces of a $\widetilde{\Theta}(n)$-node network are constructed by the two players. Then the pieces are connected by $\mathrm{O}(\log n)$ paths of length d. The diameter of the resulting network is either $d+4$ or $d+5$ depending on the solution to Set Disjointness with inputs (x, y). Finally, the classical lower bound of $\Omega(n)$ for the communication complexity of Set Disjointness implies the same lower bound on the number of rounds for any distributed algorithm for computing Diameter.

In the quantum setting, the situation is much more complex since Set Disjointness has communication complexity $\Theta(\sqrt{n}\,)$ [Raz03, AA03] for n-bit inputs. This leads to the lower bound of $\widetilde{\Omega}(\sqrt{p}+\delta)$ rounds for computing the diameter of a quantum congested network, which is significantly smaller than the upper bound stated above. Nonetheless, Le Gall and Magniez improved the lower bound for a restricted set of protocols in which each node has memory of size at most $\mathrm{poly}(\log p)$ qubits. For this, they used a more refined lower bound for Set Disjointness for bounded-round protocols.

Recall that the number of rounds in a two-party protocol is the number of messages exchanged, where the length of the messages may vary. Braverman, Garg, Ko, Mao, and Touchette [BGK$^+$18] showed that the communication complexity of r-round two-party quantum protocols for Set Disjointness on n-bit inputs is $\widetilde{\Omega}(n/r+r)$. Using this, Le Gall and Magniez showed that any quantum distributed protocol for Diameter with congestion $\mathrm{O}(\log p)$ and memory-size $\mathrm{poly}(\log p)$ per node requires $\widetilde{\Omega}(\sqrt{p\delta}\,)$ rounds. However, without any restriction on the memory size of the nodes, no better bound than $\widetilde{\Omega}(\sqrt{p}+\delta)$ was known.

1.3 Contributions

We prove that solving Set Disjointness with the two n-bit inputs given to the processors at the extremities of a line of $d+1$ quantum processors requires $\widetilde{\Omega}(\sqrt[3]{nd^2})$ rounds of communication of messages of size $\mathrm{O}(\log n)$ (**Theorem 3.1**). As a corollary, we get a new lower bound of $\widetilde{\Omega}(\sqrt[3]{p\delta^2})$ rounds for quantum distributed protocols computing the diameter δ of an p-node network with congestion $\mathrm{O}(\log p)$ (**Corollary 3.2**). This bound improves on the previous bound of $\widetilde{\Omega}(\sqrt{p})$ rounds when $\delta \in \widetilde{\Omega}(\sqrt[4]{p})$. The improvement is obtained by a more refined, information-theoretic analysis of a reduction similar to one due to Le Gall and Magniez [LM18].

We observe that the information-theoretic technique used to derive the above round lower bound for Set Disjointness on the Line also applies to two-oracle query algorithms with switching delay d (**Theorem 4.1**). We show that this bound, and the bound of $\Omega(\sqrt{n})$ coming from the standard query complexity model, are tight to within polylogarithmic factors in different ranges of the parameters n and d (**Theorem 4.2**). This presents a barrier for obtaining a better round lower bound for Set Disjointness on the Line, but leaves open the possibility of better communication protocols for the problem. We hope that these results and, more generally, the models we study also provide a better understanding of quantum distributed computing.

2 Preliminaries

We assume that the reader is familiar with the basic notions of quantum information and computation. We recommend the texts by Nielsen and Chuang [NC00] and Watrous [Wat18], and the lecture notes by de Wolf [dW19] for a good introduction to these topics. We briefly describe some notation, conventions, and the main concepts that we encounter in this work.

We write pure quantum states using the ket notation, for example as $|\psi\rangle$. By a quantum register, we mean a sequence of quantum bits (qubits). We assume for simplicity (and without loss of generality) that the computation in the models we study do not involve any intermediate measurements, i.e., they are unitary until the measurement that is made to obtain the output.

We use the notation $\widetilde{\mathrm{O}}(\cdot)$ to indicate that we are suppressing factors that are poly-logarithmic in the stated expression. For a positive integer k, we denote the set $\{1, 2, \ldots, k\}$ by $[k]$. In the sequel, we consider the computation of Boolean bi-variate functions $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ in several models of computation.

2.1 Quantum distributed computing in the CONGEST model

We consider the quantum analogue of the standard CONGEST communication model. We give a brief overview here, and refer the reader to Ref. [Pel00] for a more detailed discussion of the model and its variants. The topology of the network is given by some graph $G := (V, E)$. Each node in the network has a distinct identifier and represents a processor. Initially, the nodes know nothing about the topology of the network except the set of edges incident on them, and a polynomial upper bound $\mathrm{O}(|V|^c)$ (for some constant c) on the total number of nodes $|V|$.

There are a number of subtleties in the use of shared entanglement in this model, such as what shared states are allowed, how they are distributed, and what knowledge the processors have about the states. These considerations gain more importance in the design of algorithms in the model. That said, we are concerned with *lower bounds* in this article, and we prove them in a model which is not necessarily realistic (or well-motivated), but in which the processors are *more powerful*. The lower bounds so obtained are thus stronger.

We assume that the processors initially share an arbitrary entangled *pure* state that depends only on the number of nodes (but not on the topology of the graph). Further, each processor knows the shared state and how it is partitioned amongst the processors in the network. Thus each processor initially also knows the precise number of nodes $|V|$, but not the set of communication links beyond those with its neighbours.

Communication protocols in the CONGEST model are executed with round-based synchrony. In each round, each node may perform some quantum computation on its local memory and the message registers it uses to communicate with its neighbours. Then each node transfers one message with b qubits to each adjacent node to complete that round. The parameter b is called the *congestion* or *bandwidth* of the communication channels. Unless explicitly mentioned, we assume that the congestion b is of order $\log |V|$. All links and nodes in the network (corresponding to the edges and vertices of G, respectively) are reliable and do not suffer any faults.

In this paper we consider the special case of a d-line network, where G consists in a single path of length d. The nodes/processors at the extremities receive inputs $x, y \in \{0,1\}^n$, respectively, and the intermediate nodes get no input. The $d+1$ processors also share an arbitrary quantum state as described above. In this setting, the quantum distributed complexity of f on a d-line is the minimum number of rounds of any quantum protocol that computes f with probability at least $2/3$ and congestion $\mathrm{O}(\log n)$. The complexity of any non-trivial function f of both its arguments is $\Omega(d)$. We assume that $d \leq n$; otherwise the complexity of such a function would be $\Theta(d)$. Note that even in the model without entanglement, we may assume that d is known to each node. Otherwise, d can be computed at the cost of $\Theta(d)$ rounds, which does not affect the asymptotic complexity of such functions.

2.2 Quantum information theory

We refer the reader to the texts by Nielsen and Chuang [NC00] and Watrous [Wat18] for the basic elements of quantum information theory.

Unless specified, we take the base of the logarithm function to be 2. Whenever we consider information-theoretic quantities involving quantum registers, we assume they are in a quantum state that is implied by the context. For ease of notation, we identify the register with the quantum state.

For a register X in state ρ the *von Neumann entropy* of X is defined as $\mathrm{S}(X)_\rho \coloneqq -\operatorname{Tr}(\rho \log \rho)$. We omit the subscript ρ when the state of the register is clear from the context. If the state space of X has dimension k, then $\mathrm{S}(X) \leq \log k$.

Suppose that the registers $WXYZ$ are in some joint quantum state ρ. The *mutual information* $\mathrm{I}(X:Y)_\rho$ of X and Y is defined as

$$\mathrm{I}(X:Y)_\rho \quad \coloneqq \quad \mathrm{S}(X) + \mathrm{S}(Y) - \mathrm{S}(XY) \ .$$

The *conditional mutual information* $\mathrm{I}(X:Y \mid Z)_\rho$ of X and Y given Z is defined as

$$\mathrm{I}(X:Y \mid Z)_\rho \quad \coloneqq \quad \mathrm{I}(X:YZ)_\rho - \mathrm{I}(X:Z)_\rho \ .$$

We omit the subscript ρ when the state of the registers is clear from the context.

Conditional mutual information is invariant under the application of an isometry to any of its three arguments. The quantity also satisfies the following important property.

Lemma 2.1 (Data Processing Inequality)**.** $\mathrm{I}(X:WY \mid Z) \ \geq \ \mathrm{I}(X:Y \mid Z)$.

We may bound conditional mutual information as follows.

Lemma 2.2. $\mathrm{I}(X:WY \mid Z) \ \leq \ 2\,\mathrm{S}(W) + \mathrm{I}(X:Y \mid Z)$.

The quantity simplifies if the register on which we condition is "classical".

Lemma 2.3. *Let σ be a possible state of the registers XYZ given by*

$$\sigma \quad \coloneqq \quad \sum_z \lambda_z \, \sigma_z^{XY} \otimes |z\rangle\langle z|^Z \ ,$$

where $(|z\rangle)$ is an orthonormal basis of the state space of register Z, λ is a probability distribution on this basis, and (σ_z) are possible states of the registers XY. Then

$$\mathrm{I}(X : Y \mid Z)_\sigma \quad = \quad \mathbb{E}_{z\sim\lambda} \, \mathrm{I}(X : Y)_{\sigma_z} \ .$$

2.3 Quantum communication complexity

We informally describe a two-party quantum communication protocol *with shared entanglement* (also called an *entanglement-assisted* two-party quantum communication protocol) for computing a bi-variate Boolean function $f(x, y)$ of n-bit inputs x, y. For a formal definition, we refer the reader to an article by Touchette [Tou15]. In such a protocol, we have two parties, Alice and Bob, each of whom gets an input in registers X and Y, respectively. In the protocols we consider, the inputs are *classical*, i.e., the joint quantum state in the input registers XY is diagonal in the basis $(|x, y\rangle \ : \ x, y \in \{0, 1\}^n)$. Alice and Bob's goal is to compute the value of the function on the pair of strings in the input registers by interacting with each other.

The protocol proceeds in some number $m \geq 1$ of *rounds*. At the cost of increasing the number of rounds by 1, we assume that Alice sends the message in the first round, after which the parties alternate in sending messages. Each party holds a *work* register in addition to the input register. Initially, Alice has work register A_0, Bob has B_0. We denote the work register with Alice at the end of round $k \in [m]$ by A_k and that with Bob by B_k.

The qubits in the work registers A_0B_0 are initialised to a fixed pure state that may be entangled across the partition across A_0 and B_0, but is independent of the inputs x, y. This is called *shared entanglement*. (In the model *without* shared entanglement, the registers A_0B_0 are initialised to $|\bar{0}\rangle$.) Suppose that Alice is supposed to send the message in the k-th round, for some $k \in [m]$. Alice applies an isometry controlled by her input register X to the work register A_{k-1} to obtain registers A_kM_k. She then sends the message register M_k to Bob. Bob's work register at the end of the k-th round is then $B_k \coloneqq M_kB_{k-1}$. After the m-th round (the last round), the recipient of the last message, say Bob, measures his work register B_k, possibly controlled by his input register Y, to produce the binary output of the protocol.

The length of a message is the number of qubits in the message register for that round. The *entanglement-assisted communication complexity* of the protocol is the sum of the lengths of the m messages in it. We say the protocol computes the function f with success probability α if for all inputs x, y, the probability that the protocol outputs $f(x, y)$ is at least α. The goal of the two parties is to compute the function while minimising the communication between themselves. The *entanglement-assisted quantum communication complexity* of f is the minimum communication complexity of a quantum protocol with shared entanglement that computes f with success probability at least $2/3$.

We analyse a subtle variant of the *conditional information loss* of two-party protocols, a notion introduced by Jain, Radhakrishnan, and Sen [JRS03b] (see Appendix A). We call the variant *conditional information leakage* to distinguish it from conditional information loss. This variant is implicit in Ref. [JRS03b], and turns out to be the quantity of interest for us. We define this notion following the convention and notation given above. In particular, we assume that Alice sends the messages in the odd rounds and Bob sends the messages in the even rounds. Moreover, the only measurement in the protocol is the one for producing

the output, so that the joint state of the two parties is pure for any fixed pair of inputs. Let μ be a joint distribution over the input set $\{0,1\}^n \times \{0,1\}^n$ and an auxiliary sample space. We initialise registers $\hat{X}\hat{Y}\hat{Z}XYZ$ to the canonical purification

$$\sum_{x,y,z} \sqrt{\mu(x,y,z)}\ |xyz\rangle^{\hat{X}\hat{Y}\hat{Z}} |xyz\rangle^{XYZ}$$

of the distribution, where Z corresponds to the auxiliary random variable. We use the register labels X, Y, Z to also refer to the two input random variables (X, Y) and the auxiliary random variable (Z), respectively. We then run the two-party protocol Π using the input registers X, Y respectively, and additional work registers as described above. We imagine that the purification register $\hat{X}$ is given to Alice, the register $\hat{Y}$ is given to Bob, and that the registers $\hat{Z}Z$ are held by a third party.

The *conditional information leakage* $\widetilde{\mathrm{IL}}(\Pi \,|\, XYZ)$ of the protocol Π is defined as

$$\widetilde{\mathrm{IL}}(\Pi \,|\, XYZ) \quad := \sum_{i \in [m],\ i \text{ odd}} \mathrm{I}(X : B_i Y \hat{Y} \,|\, Z) + \sum_{i \in [m],\ i \text{ even}} \mathrm{I}(Y : A_i X \hat{X} \,|\, Z)\ ,$$

where the registers are implicitly assumed to be in the state given by the protocol. Since Alice sends the messages in the odd rounds, and Bob in the even rounds, this quantity measures the cumulative information about the inputs "leaked" to the other party, over the course of the entire protocol.

2.4 Quantum query complexity

For a thorough introduction to the quantum query model, see, for example, the lecture notes by de Wolf [dW19] and the survey by Ambainis [Amb19]. In this work, we study algorithms for computing a bi-variate Boolean function $f(x, y)$ as above, using *two* unitary operators $\mathcal{O}_x$ and $\mathcal{O}_y$ that provide access to the n-bit inputs x and y, respectively. For any $z \in \{0,1\}^n$, the operator $\mathcal{O}_z$ acts as $\mathcal{O}_z|i,b\rangle = |i, b \oplus z_i\rangle$ on the Hilbert space $\mathcal{H}$ spanned by the orthonormal basis $\{|i,b\rangle : i \in [n], b \in \{0,1\}\}$. We call operators of the form $\mathcal{O}_z$ an *oracle*, and each application of such an operator a *query*.

A query algorithm $\mathcal{A}$ with access to two oracles $\mathcal{O}_x$ and $\mathcal{O}_y$ is an alternating sequence of unitary operators $U_0, V_1, U_1, V_2, U_2, V_3, U_3, \ldots, V_t, U_t$, where the operators U_i act on a Hilbert space of the form $\mathcal{H} \otimes \mathcal{W}$ and are independent of the inputs x, y, and the $V_i \in \{\mathcal{O}_x, \mathcal{O}_y\}$. The computation starts in a fixed state $|\bar{0}\rangle \in \mathcal{H} \otimes \mathcal{W}$, followed by the sequence of unitary operators to get the final state $U_t V_t \cdots U_3 V_3 U_2 V_2 U_1 V_1 U_0 |\bar{0}\rangle$. Finally, we measure the first qubit in the standard basis to obtain the output $\mathcal{A}(x, y)$ of the algorithm. We say the algorithm computes f with success probability α if for all inputs x, y, we have $\mathcal{A}(x, y) = f(x, y)$ with probability at least α.

As in the standard quantum query model, we focus on the number of applications of the operators $\mathcal{O}_x$ and $\mathcal{O}_y$ in an algorithm, and ignore the cost of implementing unitary operators that are independent of x and y. The *query complexity* of an algorithm is the number of queries made by the algorithm (t in the definition above). The *quantum query complexity* of a function f is the minimum query complexity of any quantum algorithm that computes f with probability at least $2/3$.

3 Set Disjointness on a Line

3.1 The problem and results

The Set Disjointness problem $\mathsf{L}_{n,d}$ on a line was introduced recently by Le Gall and Magniez [LM18] in the context of distributed computing. It is a communication problem involving $d + 1$ communicating

parties, $\mathsf{A}_0, \mathsf{A}_1, \ldots, \mathsf{A}_d$, arranged on the vertices of a path of length d. The edges of the path denote two-way quantum communication channels between the players. Parties A_0 and A_d receive n-bit inputs $x, y \in \{0,1\}^n$, respectively. The $d+1$ parties also share an arbitrary entangled state as described in Section 2.1. The communication protocol proceeds in rounds. In each round, parties A_{i-1} and A_i may exchange b qubits in each direction, for each $i \in [d]$, i.e., the *bandwidth* of each communication channel is b. The goal of the parties is to determine if the sets x and y intersect or not. I.e., they would like to compute the Set Disjointness function $\mathsf{D}_n(x,y) \coloneqq \bigvee_{i=1}^{n}(x_i \wedge y_i)$.

We are interested in the number of rounds required to solve $\mathsf{L}_{n,d}$. We readily get a quantum protocol Π_d for this problem with $\mathrm{O}(\sqrt{nd}\,)$ rounds by following an observation due to Zalka [Zal99] on black-box algorithms that make "parallel" queries. Let Π denote the optimal two-party quantum communication protocol for Set Disjointness due to Aaronson and Ambainis [AA03]. In Π_d, we partition the n-bit inputs into d parts of length n/d each. Parties A_0 and A_d then simulate Π on each of the d corresponding pairs of inputs independently. The protocol Π runs in $\sqrt{n/d}$ rounds with $\mathrm{O}(1)$ qubits of communication per instance of length n/d, per round. So the total communication to or from A_0 due to one round of the d runs of Π is $\mathrm{O}(d)$. Since $\mathrm{O}(d)$ qubits can be transmitted across the path of length d in $\mathrm{O}(d)$ rounds of the multi-party protocol, the protocol Π_d simulates the d parallel runs of Π in $\mathrm{O}(\sqrt{nd}\,)$ rounds. Since Π finds an intersection with probability at least $3/4$ whenever there is one, and does not err when there is no intersection, the protocol Π_d also has the same correctness probability.

Le Gall and Magniez observed that a lower bound of $\Omega(\sqrt{n}/b)$ for the number of rounds follows from the $\Omega(\sqrt{n}\,)$ lower bound due to Razborov [Raz03] on the quantum communication complexity of Set Disjointness in the two-party communication model. This is because two parties, Alice and Bob, may use any r-round protocol for $\mathsf{L}_{n,d}$ to solve Set Disjointness with $2rb$ qubits of communication: Alice simulates A_0 and Bob simulates the actions of the remaining parties $\mathsf{A}_1, \mathsf{A}_2, \ldots, \mathsf{A}_d$. An $\Omega(d)$ lower bound is also immediate due to the need for communication between A_0 and A_d.

Le Gall and Magniez devised a more intricate simulation of a protocol for $\mathsf{L}_{n,d}$ by two parties, thereby obtaining a two-party protocol for Set Disjointness. Using this, they obtained a round lower bound of $\widetilde{\Omega}(\sqrt{nd}\,)$ for $\mathsf{L}_{n,d}$ when the bandwidth b of each communication channel (in each round) and the local memory of the players $\mathsf{A}_1, \mathsf{A}_2, \ldots, \mathsf{A}_{d-1}$ are both $\mathrm{O}(\log n)$ qubits. We show that a similar simulation leads to an unconditional round lower bound of $\Omega(nd^2/b)^{1/3}$ by studying the conditional information leakage of the resulting two-party protocol (see Section 2.3).

Theorem 3.1. *Any entanglement-assisted quantum communication protocol with error probability at most $1/3$ for the Set Disjointness problem $\mathsf{L}_{n,d}$ on the line requires $\Omega(\sqrt[3]{nd^2/b}\,)$ rounds.*

This bound dominates the straightforward bound of $\Omega(\sqrt{n}/b)$ mentioned above when $d \geq \sqrt[4]{n}/b$, i.e., when $d \geq \sqrt[4]{n}/\log n$ when $b \coloneqq \log n$. However, we do not know if either bound is achievable in the respective parameter regimes. We study the optimality of the above bound via a related query model in Section 4.

Using the reduction from $\mathsf{L}_{n,d}$ to the problem of computing the diameter described in the proof of Theorem 1.3 in Ref. [LM18], we get a new lower bound for quantum distributed protocols for the diameter problem in the CONGEST model.

Corollary 3.2. *Any distributed protocol for computing the diameter δ of n-node networks with congestion $\mathrm{O}(\log n)$ in the quantum CONGEST model (possibly with shared entanglement), requires $\widetilde{\Omega}(\sqrt[3]{n\delta^2}\,)$ rounds.*

In more detail, the reduction mentioned above is based on a construction due to Abboud, Censor-Hillel, and Khoury [MN16], and involves a network $G_{n,d}(x,y)$ in which the number of nodes is determined by n

and d, but the edges may also depend on the inputs x, y to $\mathsf{L}_{n,d}$. Given an instance of $\mathsf{L}_{n,d}$, the $d+1$ parties (A_i) locally construct parts of the network $G_{n,d}(x,y)$. The parts are such that each party holds a disjoint subset of the vertices of $G_{n,d}(x,y)$, and each edge of $G_{n,d}(x,y)$ is between vertices held either by adjacent parties or by the same party. Only the edges between the vertices held by A_0 may depend on the input x given to it, and only the edges between the vertices held by A_d may depend on the input y given to it. The remaining edges are determined by n, d. Given an algorithm for the Diameter Problem, the parties (A_i) are then able to simulate the it on the graph $G_{n,d}(x,y)$. In particular, these observations imply that shared entanglement in the network $G_{n,d}(x,y)$ of the type described in Section 2.1 translates to shared entanglement between the parties (A_i) of the same type. We refer the reader to Refs. [MN16, LM18] for the remaining details of the reduction.

3.2 Overview of the proof

We begin by giving an overview of the proof of Theorem 3.1. It rests on a simulation of a protocol for $\mathsf{L}_{n,d}$ by a two-party protocol for Set Disjointness similar to one designed by Le Gall and Magniez [LM18, Theorem 6.1]. (In fact, the simulation works for any multi-party protocol over the path of length d that computes some bi-variate function $g(x,y)$ of the inputs given to A_0 and A_d.) The idea underlying the simulation is the following. Suppose we have a protocol Π_d for the problem $\mathsf{L}_{n,d}$. In the two-party protocol Π, Alice begins by holding the registers used by parties $\mathsf{A}_0, \mathsf{A}_1, \ldots, \mathsf{A}_{d-1}$. She then simulates all the actions—local operations and communication—of the parties $\mathsf{A}_0, \mathsf{A}_1, \ldots, \mathsf{A}_{d-1}$ from the first round in Π_d, except for the communication between A_{d-1} and A_d. This is possible because these actions do not depend on the input y held by A_d. She can continue simulating the actions of $\mathsf{A}_0, \mathsf{A}_1, \ldots, \mathsf{A}_{d-2}$ from the second round, except the communication between A_{d-2} and A_{d-1}, as these do not depend on the message from A_d from the first round in Π_d. Continuing this way, Alice can simulate the actions of $\mathsf{A}_0, \mathsf{A}_1, \ldots, \mathsf{A}_{d-i}$ from round i of Π_d, except the communication between A_{d-i} and A_{d-i+1}, for all $i \in [d]$, all in one round of Π. These actions constitute Alice's local operations in the first round of Π.

Alice then sends Bob the local memory used by parties $\mathsf{A}_1, \ldots, \mathsf{A}_{d-1}$ in Π_d, along with the qubits sent by A_{i-1} to A_i in round i, for each $i \in [d]$. (Alice retains the input x and the memory used by party A_0.) This constitutes the first message from Alice to Bob in Π.

Given the first message, Bob can simulate the remaining actions of $\mathsf{A}_1, \mathsf{A}_2, \ldots, \mathsf{A}_d$ from the first d rounds of Π_d, except for the communication from A_1 to A_0. These constitute his local operations in the second round of Π. He then sends Alice the qubits sent by A_1 to A_0 in round d of Π_d along with the local memory used by the parties A_i, for $i \in [d-1]$. (Bob retains the input y and the local memory used by party A_d.) This constitutes the second message in Π.

In effect, the simulation implements the first d rounds of Π_d in two rounds of Π (see Figure 2). The same idea allows Alice and Bob to simulate the rest of the protocol Π_d while implementing each successive block of d rounds of Π_d in two rounds of Π, with communication per round of the order of $d(b+s)$, where b is the bandwidth of the communication channels in Π_d, and s is a bound on the number of qubits of local memory used by any of the parties $\mathsf{A}_1, \ldots, \mathsf{A}_{d-1}$. Building on the detailed description of protocols on the line in Section 3.3, we describe the simulation formally in Section 3.4, and show the following.

Lemma 3.3. *Given any r-round entanglement-assisted quantum protocol Π_d for $\mathsf{L}_{n,d}$ over communication channels with bandwidth b in which each party uses local memory at most s, there is an entanglement-assisted two-party quantum protocol Π for Set Disjointness D_n that has $2\lceil r/d \rceil$ rounds, total communication of order $r(b+s)$, and has the same probability of success. Further, if the protocol Π_d does not use shared entanglement, the protocol Π also does not.*

The communication required by a k-round bounded-error two-party protocol for Set Disjointness is $\Omega(n/(k \log^8 k))$ [BGK+18, Theorem A]. This gives us the lower bound of $\widetilde{\Omega}(\sqrt{nd}\,)$ due to Le Gall and Magniez on the number of rounds r in Π_d, when $b + s$ is of order $\log n$. More precisely, the bound with the logarithmic factors is

$$\Omega\left(\frac{\sqrt{nd}}{(\log n)^{1/2} \log^4 \frac{n}{d \log n}}\right) .$$

In fact, in the case the protocol Π_d does not use shared entanglement, we may derive an unconditional lower bound on the number of rounds from the same reduction, one that holds without any restriction on the local memory used by the parties in Π_d. This is because the state of the registers of any party A_i, with $i \in [d-1]$, in an r-round protocol without entanglement has support on a fixed subspace of dimension at most 2^{4br}, independent of the inputs, at any moment in the protocol. This follows from an argument due to Yao [Yao93], by considering a two party protocol obtained by grouping all parties except A_i together (see Appendix B). So the state of party A_i at any point in the protocol can be mapped to one over $4br$ qubits. Using this for the bound s on the local memory, bandwidth $b \in \mathrm{O}(\log n)$, and the same reasoning as before, we get a lower bound of $\widetilde{\Omega}(nd)^{1/3}$ on the number of rounds r in Π_d. The precise expression for the bound with the logarithmic terms is

$$\Omega\left(\frac{(nd)^{1/3}}{(\log n)^{1/3} \log^{8/3} \frac{n}{d^2 \log n}}\right) .$$

We refine the analysis further to obtain Theorem 3.1, by appealing to an information-theoretic argument. The key insight is that regardless of the size of the local memory maintained by the parties A_i, for $i \in [d-1]$, the new *information* they get about either input x or y in one round is bounded by b, the length of the message from A_0 or A_d, respectively. Thus, the total information contained in the memory and messages of these parties about the inputs may be bounded by rb at any point in the protocol (see Lemma 3.6). This carries over to the information contained in the messages between Alice and Bob in the two-party protocol Π derived from Π_d. The conditional information leakage of the two-party protocol may then be bounded by rbm, where m is the number of rounds in Π (for suitable distributions over the inputs).

Lemma 3.4. *Let XYZ be jointly distributed random variables such that $X, Y \in \{0,1\}^n$, and X and Y are independent given Z. The conditional information leakage of the two-party protocol Π for Set Disjointness D_n mentioned in Lemma 3.3 is bounded as $\widetilde{\mathrm{IL}}(\Pi \,|\, XYZ) \in \mathrm{O}(r^2 b/d)$.*

We derive this as Corollary 3.7 in Section 3.5.

We now appeal to the following result due to Jain *et al.* [JRS03b] on the conditional information leakage of bounded-round protocols for Set Disjointness. This result is implicit in the proof of the $\Omega(n/m^2)$ lower bound on the communication required by m-round quantum protocols for Set Disjointness. (See Appendix A for the details, including the significance of the auxiliary random variable Z.)

Theorem 3.5 (Jain, Radhakrishnan, Sen [JRS03b])**.** *There is a choice of distribution for XYZ such that $X, Y \in \{0,1\}^n$, the random variables X and Y are independent given Z, and for any bounded-error entanglement-assisted two-party quantum communication protocol Γ for Set Disjointness D_n with m rounds, the conditional information leakage $\widetilde{\mathrm{IL}}(\Gamma \,|\, XYZ)$ is at least $\Omega(n/m)$.*

Since the number of rounds m in the two-party protocol Π is at most $2\lceil r/d \rceil$, we conclude the $\Omega(nd^2/b)^{1/3}$ lower bound stated in Theorem 3.1.

3.3 Formal description of protocols on the line

In order to establish the lemmas stated in Section 3.2, we introduce some conventions and notation associated with multi-party protocols on the line of the sort we study for $\mathsf{L}_{n,d}$. By using unitary implementations of measurements, we assume that all the local operations in the protocol, except the final measurement to obtain the outcome of the protocol, are unitary. We also that the input registers X with A_0 and Y with A_d are read-only. I.e., the input registers may only be used as control registers during the protocol, and are retained by the respective parties throughout.

For ease of exposition, we use subscripts on the registers held by all the parties to implicitly specify the state of the register and the party which last modified the state of the register. At the beginning of round $t+1$, for $t \in \{0, 1, \ldots, r-1\}$, party A_0 holds registers $X A_{0,t} L_{1,t}$, party A_d holds registers $R_{d-1,t} A_{d,t} Y$, and for $i \in [d-1]$, party A_i holds registers $R_{i-1,t} A_{i,t} L_{i+1,t}$. The registers $L_{i,t}$ and $R_{i,t}$, for $i \in [0,d]$ and $t \in [0,r]$, all have b qubits. Except in the first round, the first subscript at the beginning of the round, say i, indicates that party A_i held the register in the previous round, and sent the register to the neighbour that holds it in the current round.

At the beginning of the first round, registers X and Y are initialized to the input to the protocol. The qubits in the remaining registers are all initialised to a pure shared state that is independent of the inputs. Note that this shared state also includes any "work" qubits in state $|\bar{0}\rangle$ required by the parties.

In round $t+1$, each party A_i applies a unitary operation to the registers they hold. We view the unitary operation as an isometry $U_{i,t+1}$ that maps the registers to another sequence of registers with the same dimensions. The registers $X A_{0,t} L_{1,t}$ with A_0 are mapped to $X A_{0,t+1} R_{0,t+1}$. The registers $R_{d-1,t} A_{d,t} Y$ with A_d are mapped to $L_{d,t+1} A_{d,t+1} Y$. For $i \in [d-1]$, the registers $R_{i-1,t} A_{i,t} L_{i+1,t}$ with A_i are mapped to $L_{i,t+1} A_{i,t+1} R_{i,t+1}$. So for $t \geq 1$, the subscripts i, t on a register indicate that the register was an "output" of the isometry applied by party A_i in round t, and that it is in the corresponding state.

As the final action in round $t+1$, for $t < r$, if $i > 0$, party A_i sends $L_{i,t+1}$ to the party on the left (i.e., to A_{i-1}) and receives $R_{i-1,t+1}$ from her; and if $i < d$, she sends $R_{i,t+1}$ to the party on the right (i.e., to A_{i+1}) and receives register $L_{i+1,t+1}$ from her.

After the r rounds of the protocol have been completed, party A_0 makes a two-outcome measurement, possibly depending on her input, on the registers $A_{0,r} L_{1,r}$. The outcome is the output of the protocol. Figure 1 depicts such a protocol.

3.4 The two-party simulation

We now prove Lemma 3.3, by giving a formal description of the two-party protocol Π for Set Disjointness D_n derived from a protocol Π_d for $\mathsf{L}_{n,d}$. We use the notation and convention defined in Section 3.3 in our description below. For simplicity, we assume that the number of rounds r in Π_d is a multiple of d, by adding dummy rounds with suitable local operations, if necessary. Since D_n depends non-trivially on *both* inputs, the number of rounds r required to compute the function over a path of length d is at least d. So the addition of dummy rounds may at most double the number of rounds.

In the protocol Π, Alice initially holds all the registers with parties A_i for $i < d$ at the beginning of the first round, and Bob holds the registers with A_d. All of the registers are initialized as in Π_d. The simulation implements blocks of d successive rounds of Π_d with two rounds in Π, with Alice sending the message in the first of the two rounds and Bob in the second. See Figure 2 for a depiction of the simulation.

Assume that k blocks of d rounds each of Π_d have been implemented with $2k$ rounds in Π, for some $k \in [0, r/d - 1]$. We describe how the $(k+1)$-th block is implemented. Let $t := kd$. We maintain the invariant that at the beginning of the $(2k+1)$-th round in Π, Alice holds the registers $X A_{0,t} L_{1,t}$, and the

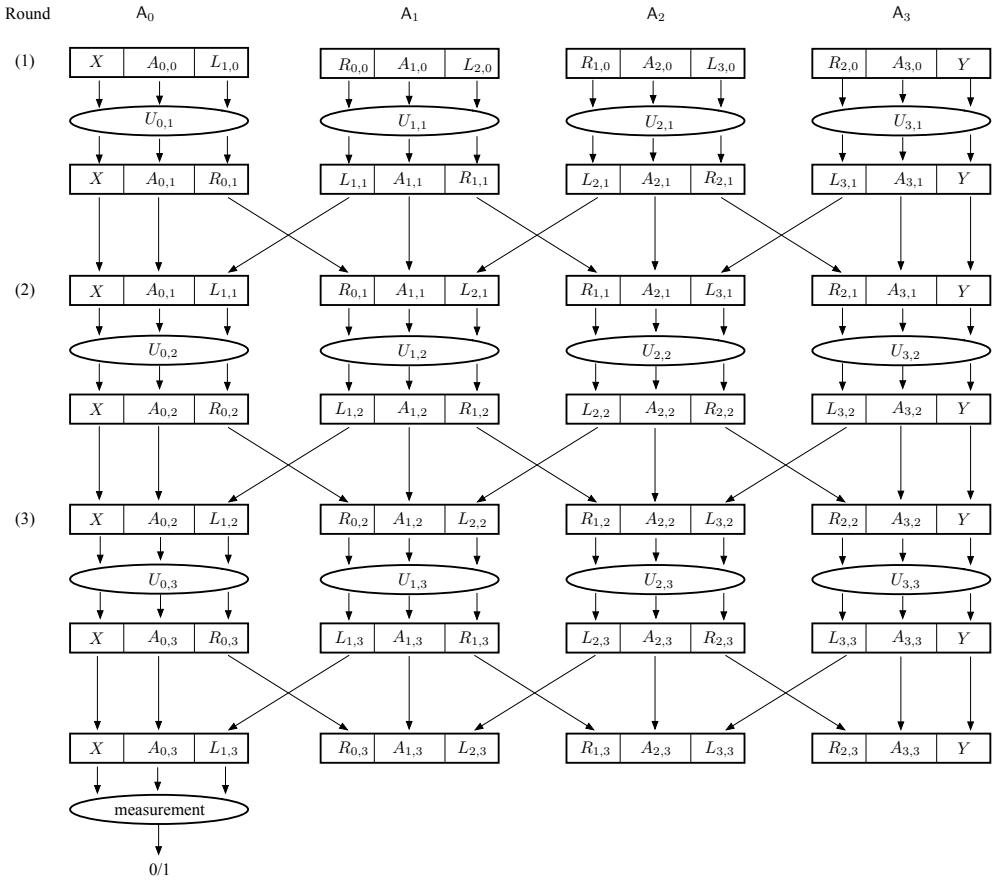

Figure 1: A multi-party communication protocol on the line with 4 parties and 3 rounds, of the type we study for $\mathsf{L}_{n,d}$. For $t \geq 1$, the subscripts i, t on a register indicate that the register was an "output" of the isometry applied by party A_i in round t, and that it is in the corresponding state. For example, the register $R_{1,3}$ was produced by the isometry applied by A_1 in the third round.

registers $R_{i-1,t}\, A_{i,t}\, L_{i+1,t}$, for all $i \in [d-1]$. Alice's local operations in round $2k+1$ are as follows. For each $j \in \{t+1, t+2, t+3, \ldots, t+d\}$ in increasing order (where j denotes a round in Π_d),

1. Alice applies the isometry $U_{0,j}$ to the registers $X\, A_{0,j-1}\, L_{1,j-1}$ to get registers $X\, A_{0,j}\, R_{0,j}$.
2. For each l with $1 \leq l \leq d-(j-t)$ (denoting a party from Π_d), Alice applies the isometry $U_{l,j}$ to the registers $R_{l-1,j-1}\, A_{l,j-1}\, L_{l+1,j-1}$ to get registers $L_{l,j}\, A_{l,j}\, R_{l,j}$.
3. For each l with $1 \leq l \leq d-(j-t)$, Alice swaps registers $R_{l-1,j}$ and $L_{l,j}$.

At this point, Alice has implemented the left upper triangular "space-time slice" of the $(k+1)$-th block of d rounds of Π_d. She holds the registers

$$\begin{aligned} &X\, A_{0,t+d}\, R_{0,t+d}\;\; R_{0,t+d-1}\, A_{1,t+d-1}\, R_{1,t+d-1}\;\; R_{1,t+d-2}\, A_{2,t+d-2}\, R_{2,t+d-2} \\ &R_{2,t+d-3}\, A_{3,t+d-3}\, R_{3,t+d-3}\;\cdots\; R_{d-3,t+2}\, A_{d-2,t+2}\, R_{d-2,t+2}\;\; R_{d-2,t+1}\, A_{d-1,t+1}\, R_{d-1,t+1}\;\;, \end{aligned} \tag{3.1}$$

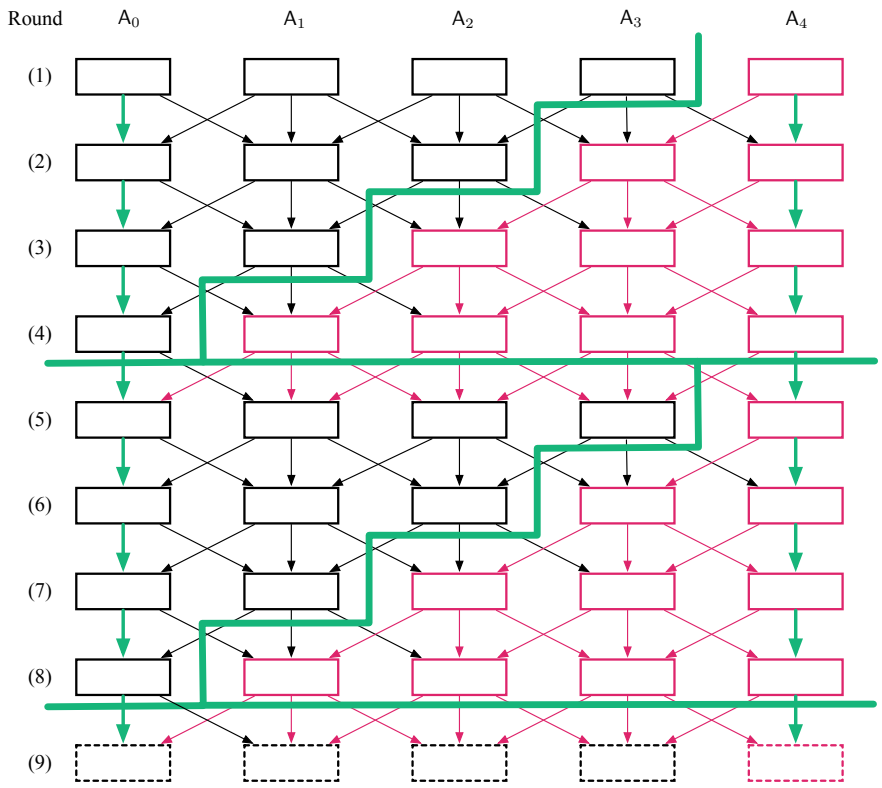

Figure 2: A depiction of the two-party simulation of a multi-party communication protocol of the type we study for $\mathsf{L}_{n,d}$. Here, we have 5 parties and show the simulation of the first 8 rounds of the original protocol. Each round in the two-party protocol is delineated by thick green lines. The black rectangular boxes represent the isometries implemented by Alice, and the black arrows going across the thick green lines represent the communication from her to Bob. The red rectangular boxes represent the isometries implemented by Bob, and the red arrows going across the thick green lines represent the communication from him to Alice. The green arrows indicate that the input register and the local memory of the parties at the extremities are retained by them throughout.

in the state implicitly specified by the subscripts. (The registers have been grouped into threes, in the order of the parties that hold them in Π_d.) She sends all the registers *except* $XA_{0,t+d}$ to Bob. This concludes the $(2k+1)$-th round of Π.

We also maintain the invariant that at the beginning of the $(2k+2)$-th round of Π, Bob holds all the registers in Eq. (3.1) except $XA_{0,t+d}$, in addition to the registers $R_{d-1,t}\, A_{d,t}Y$, where $t = kd$. Bob's local operations in round $2k+2$ are as follows. For each $j \in \{t+1, t+2, t+3, \ldots, t+d\}$ in increasing order (where j denotes a round in Π_d that Bob intends to complete),

1. Bob applies the isometry $U_{d,j}$ to the registers $R_{d-1,j-1}\, A_{d,j-1}Y$ to get registers $L_{d,j}\, A_{d,j}Y$.
2. For each l with $d-(j-t-1) \leq l \leq d-1$ (denoting a party from Π_d), Bob applies the isometry $U_{l,j}$ to the registers $R_{l-1,j-1}\, A_{l,j-1}\, L_{l+1,j-1}$ to get registers $L_{l,j}\, A_{l,j}\, R_{l,j}$.
3. For each l with $d-(j-t-1) \leq l \leq d$, Bob swaps registers $R_{l-1,j}$ and $L_{l,j}$.

At this point, Bob holds the registers

$$\begin{array}{c} L_{1,t+d}\ R_{0,t+d}\,A_{1,t+d}\,L_{2,t+d}\ R_{1,t+d}\,A_{2,t+d}\,L_{3,t+d}\ R_{2,t+d}\,A_{3,t+d}\,L_{4,t+d} \\ \cdots\ R_{d-2,t+d}\,A_{d-1,t+d}\,L_{d,t+d}\ R_{d-1,t+d}\,A_{d,t+d}Y\ , \end{array} \tag{3.2}$$

in the state implicitly specified by the subscripts. The registers are thus all in the state at the end of the $(kd+d)$-th round in Π_d. Bob sends all the registers *except* $R_{d-1,t+d}\,A_{d,t+d}Y$ to Alice. This concludes the $(2k+2)$-th round of Π, and the simulation of the $(k+1)$-th block of rounds of Π_d.

At the end of the simulation of the (r/d)-th block of rounds of Π_d, Alice measures the registers $A_{0,r}\,L_{1,r}$ as in Π_d to obtain the output. (As in Π_d, this measurement may be controlled by the input register X.) This completes the description of the two-party simulation. The correctness of the simulation follows by induction, by observing that Alice and Bob implement all the local operations and communication in Π_d in the correct order and with the correct registers. Lemma 3.3 thus follows.

3.5 Conditional information leakage of the two-party protocol

We are now ready to bound the conditional information leakage of the two-party protocol Π derived from the multi-party protocol Π_d, with respect to a distribution μ on the inputs and an auxiliary random variable. We initialise registers $\hat{X}\hat{Y}\hat{Z}XYZ$ to the canonical purification

$$\sum_{x,y,z} \sqrt{\mu(x,y,z)}\ |xyz\rangle^{\hat{X}\hat{Y}\hat{Z}}|xyz\rangle^{XYZ}\ ,$$

and run the protocol Π_d (and therefore Π) on the input registers X and Y, along with the other registers they need. We use X, Y, Z to also refer to the input and auxiliary random variables. Suppose that μ is such that X and Y are independent given Z. We imagine that the purification register $\hat{X}$ is given to party A_0 in Π_d (or to Alice in Π), the register $\hat{Y}$ is given to party A_d in Π_d (or to Bob in Π), and the registers $\hat{Z}Z$ are held by a party not involved in either protocol.

We first bound the information contained about any input in the registers held by all other parties in Π_d, conditioned on Z. For ease of notation, for $t \geq 0$, we denote by D_t the entire sequence of registers (including $\hat{Y}$) held by the parties A_i, with $i \geq 1$, in the state at the end of the t-th round of Π_d. Similarly, we denote by C_t the entire sequence of registers (including $\hat{X}$) held by the parties A_i, with $i \leq d-1$, in the state at the end of the t-th round of Π_d.

Lemma 3.6. *For all $t \geq 0$, we have $\mathrm{I}(X : D_t \,|\, Z) \leq 2tb$, and $\mathrm{I}(Y : C_t \,|\, Z) \leq 2tb$.*

Proof: We prove the bound on $\mathrm{I}(X : D_t \,|\, Z)$ by induction. The second bound is obtained similarly.

Let σ denote the state of the registers XD_tZ, so that

$$\sigma \quad = \quad \sum_z \lambda_z\ \sigma_z^{XD_t} \otimes |z\rangle\langle z|^Z\ ,$$

where λ is the marginal distribution of Z, and σ_z is the state of the registers XD_t, conditioned on the event $Z = z$. By Lemma 2.3,

$$\mathrm{I}(X : D_t \,|\, Z)_\sigma \quad = \quad \mathbb{E}_{z\sim\lambda}\,\mathrm{I}(X : D_t)_{\sigma_z}\ .$$

The base case $t = 0$ is then immediate from the following observations. The state of the registers of the parties A_i, for $i \in [d]$, except $Y\hat{Y}$, is independent of the inputs, i.e., is in tensor product with the state

of $XY\hat{Y}$. Further, for every z, the state of the registers $Y\hat{Y}$ is in tensor product with that of X, conditioned on $Z = z$.

Assume that the bound holds for $t = j$, with $j \geq 0$. Let G_{j+1} denote the sequence of registers with all the parties A_i, for $i \geq 2$, after the isometry in round $j + 1$ has been applied. Then we have

$$\mathrm{I}(X : L_{1,j+1}\, A_{1,j+1}\, R_{1,j+1} G_{j+1} \,|\, Z) \quad = \quad \mathrm{I}(X : D_j \,|\, Z) \quad \leq \quad 2jb \ ,$$

by the invariance of conditional mutual information under isometries and the induction hypothesis. Let H_{j+1} denote all the registers of the parties A_i, for $i \geq 2$, after the communication in round $j+1$. Then $L_{2,j+1}\, H_{j+1}$ and $R_{1,j+1}\, G_{j+1}$ consist of the same set of registers, but in different order. By the properties of entropy and conditional mutual information mentioned below,

$$\begin{aligned}
\mathrm{I}(X : D_{j+1} \,|\, Z) &= \mathrm{I}(X : R_{0,j+1}\, A_{1,j+1}\, L_{2,j+1}\, H_{j+1} \,|\, Z) \\
&= \mathrm{I}(X : R_{0,j+1}\, A_{1,j+1}\, R_{1,j+1}\, G_{j+1} \,|\, Z) \\
&\leq 2\,\mathrm{S}(R_{0,j+1}) + \mathrm{I}(X : A_{1,j+1}\, R_{1,j+1}\, G_{j+1} \,|\, Z) \\
&\leq 2b + \mathrm{I}(X : L_{1,j+1}\, A_{1,j+1}\, R_{1,j+1} G_{j+1} \,|\, Z) \\
&\leq 2b + 2jb \ .
\end{aligned}$$

The first inequality follows from Lemma 2.2, the second by the property that $\mathrm{S}(B)$ is bounded from above by the number of qubits in the register B and the data processing inequality (Lemma 2.1), and the final one by the induction hypothesis. ∎

For $l \in [2r/d]$, denote the message registers in the l-th round of the two-party protocol Π in the corresponding state together by M_l. Denote the registers with Alice at the end of the l-th round (including $\hat{X}$), in the corresponding state, by E_l, and the registers with Bob at the end of the l-th round (including $\hat{Y}$), in the corresponding state, by F_l.

Consider $k \in [r/d]$. We observe from the definition of the protocol Π, that for the odd numbered round $2k - 1$, the state given by register D_{kd} is obtained by an isometry on the registers $M_{2k-1}F_{2k-2}$. The registers $M_{2k-1}F_{2k-2}$ (in the state implicitly specified by their definition) are precisely the registers Bob holds at the end of round $2k - 1$ of Π. Moreover, for the even numbered round $2k$, the state given by the registers $E_{2k-1}M_{2k}$ is precisely the state given by the register C_{kd} in Π_d. The registers $E_{2k-1}M_{2k}$ are precisely the registers Alice holds at the end of round $2k$ of Π. Therefore, by Lemma 3.6 and the definition of conditional information leakage, we have:

Corollary 3.7. *For all $k \in [r/d]$, we have*

$$\begin{aligned}
\mathrm{I}(X : M_{2k-1}F_{2k-2} \,|\, Z) &= \mathrm{I}(X : D_{kd} \,|\, Z) \leq 2kdb \ , \qquad \textit{and} \\
\mathrm{I}(Y : E_{2k-1}M_{2k} \,|\, Z) &= \mathrm{I}(Y : C_{kd} \,|\, Z) \leq 2kdb \ .
\end{aligned}$$

Consequently, the conditional information leakage of Π is bounded as $\widetilde{\mathrm{IL}}(\Pi \,|\, XYZ) \leq 4r^2 b/d$.

4 Two-oracle query algorithms with a switching delay

In this section, we define a new model of query complexity, two-oracle query complexity with a "switching delay", motivated by the study of Set Disjointness on a Line $\mathsf{L}_{n,d}$. The lower bound technique involving

conditional information leakage that we use to establish Theorem 3.1 extends to the analogue of Set Disjointness D_n in this model, with a switching delay of d queries. As a consequence, it yields the same lower bound on *query* complexity. Furthermore, we design a quantum algorithm that matches this bound up to a polylogarithmic factor. This shows that the lower bound on conditional information leakage for D_n stated in Theorem 3.5 is optimal up to a logarithmic factor. Therefore, if the lower bound for $\mathsf{L}_{n,d}$ is not optimal, we would require different ideas to improve it.

4.1 The new query model

Turning to the definition of the query model, we consider query algorithms for computing bi-variate functions $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. We define the quantum version of the model; the classical versions may be defined analogously. The inputs x, y to the algorithm are provided indirectly, through oracles $\mathcal{O}_x$ and $\mathcal{O}_y$, as defined in Section 2.4. The query algorithm is defined in the standard manner, as an alternating sequence of unitary operators independent of the inputs x, y, and queries $\mathcal{O}_x$ or $\mathcal{O}_y$, applied to a fixed initial state (that is also independent of the inputs). Thus, the sequence of queries to the inputs is pre-determined. However, we define the complexity of the algorithm differently. In addition to the queries, we charge the algorithm for switching between a query to x and a query to y. We include a cost of d in the complexity whenever the algorithm switches between a query to x and a query to y. This cost parallels the cost of accessing the inputs in the distributed computing scenario in which the inputs are physically separated by distance d.

We may simplify the above model as follows, at the expense of increasing the complexity by a factor of at most 2. In the simplified model, we require that the queries be made in *rounds*. In each round, the algorithm makes d queries, but exclusively to one of the inputs x or y. Further, the algorithm alternates between the two oracles $\mathcal{O}_x$ and $\mathcal{O}_y$ in successive rounds. The complexity of the algorithm is now defined in the standard manner, as the total number of queries in the algorithm. Thus the complexity equals d times the number of rounds.

It is straightforward to verify that any algorithm with complexity q in the first model has complexity at most $2q$ in the second model, for computing any function f that depends on both inputs (i.e., when the algorithm in the first model queries both oracles). Furthermore, any algorithm with complexity q in the second model has complexity at most $2q$ in the first model. The two models are thus equivalent up to a factor of two in complexity.

The second model is also relevant in a "semi-parallel" scenario, where a sequence of d queries are made to x independently of the answers to d other queries made to y during the same time steps. Up to a factor of 2 in complexity, this semi-parallel model can be simulated by the second model above. We thus adopt the second model in the definition below.

Definition 4.1. *A* two-oracle delay-d quantum query algorithm *is a query algorithm $\mathcal{A}$ with (predetermined) access to two oracles $\mathcal{O}_1, \mathcal{O}_2$, which may be decomposed into some number of contiguous sequences of unitary operators called rounds such that each round contains d queries to the same oracle, and the algorithm alternates between the two oracles in successive rounds. The* round complexity *of $\mathcal{A}$ is the number r of rounds in a decomposition of $\mathcal{A}$ as above. The* delay-d query complexity *of $\mathcal{A}$ is $d \times r$.*

We define the *quantum two-oracle delay-d round complexity* of a bi-variate function f as the minimum round-complexity of any two-oracle delay-d quantum query algorithm computing f with probability of error at most $1/3$, given oracles $\mathcal{O}_x, \mathcal{O}_y$ for the inputs x, y. We define the *quantum two-oracle delay-d query complexity* of f similarly. We may assume that $d \leq n$, as otherwise, an algorithm can learn x and y in two rounds.

Adapting the tools developed in Section 3 we get the following lower bound.

Theorem 4.1. *Let $d \leq n$. The quantum two-oracle delay-d round complexity of Set Disjointness D_n is $\Omega(\sqrt{n}/d)$ and $\Omega(\sqrt[3]{n/(d \log n)})$. The quantum two-oracle delay-d query complexity is $\Omega(\sqrt{n})$ and $\Omega(\sqrt[3]{nd^2/\log n})$.*

Note that the first expression for either bound dominates when $d^4 \in \mathrm{O}(n \log^2 n)$.

We briefly sketch the proof of Theorem 4.1. The query lower bound follows from the one on rounds. The $\Omega(\sqrt{n}/d)$ lower bound on rounds follows by observing that Set Disjointness D_n simplifies to the unordered search problem (OR function on n bits) in the standard quantum query model when we set y to be the all 1s string. For the second lower bound, we view a query to an oracle $\mathcal{O}_x$ or $\mathcal{O}_y$ as the exchange of $2(\log n + 1)$ qubits between the algorithm and the oracle. So we can use any r-round algorithm for computing f in the two-oracle delay-d query model to derive a two-party communication protocol for computing f also with r rounds. The two parties run the query algorithm, each party sending all its registers to the corresponding player, whenever the algorithm switches between queries to x and queries to y. In each round, the state of the algorithm (therefore the corresponding message) accumulates at most $2d(\log n + 1)$ qubits of *additional* information about either input. This is a consequence of the same kind of reasoning as in Lemma 3.6. Thus the conditional information leakage of the resulting two-party protocol may be bounded by $2r^2 d(\log n + 1)$. By Theorem 3.5, this is $\Omega(n/r)$, so we get the $\Omega(\sqrt[3]{n/(d \log n)})$ lower bound for the number of rounds stated in Theorem 4.1.

4.2 Algorithm for Set Disjointness

Finally, we present an algorithm in the two-oracle model that matches the lower bounds stated in Theorem 4.1, up to polylogarithmic factors.

Theorem 4.2. *Let $d \leq n$. The quantum two-oracle delay-d round and query complexity of Set Disjointness D_n are*

- $\mathrm{O}(\sqrt{n \log n}/d)$ *and* $\mathrm{O}(\sqrt{n \log n})$*, respectively, when $d^4 \leq n \log^3 n$; and*
- $\mathrm{O}(\sqrt[3]{n/d})$ *and* $\mathrm{O}(\sqrt[3]{nd^2})$*, respectively, when $d^4 \geq n \log^3 n$.*

Proof: We present a quantum two-oracle delay-d query algorithm with a parameter $t \in [n]$, which gives the round and query bounds for suitable choices of t depending on how large d is as compared with n.

The quantum algorithm searches for a subset $I \subseteq [n]$ of size t such that it contains an index $i \in [n]$ with $x_i = y_i = 1$. If it succeeds in finding such a subset I, we may also find an index $i \in [n]$ with $x_i = y_i = 1$ without increasing the asymptotic complexity of the algorithm (although this is not required for computing D_n). For this, the algorithm sequentially runs through the indices in I and checks if the requisite condition is satisfied. This second stage of the algorithm can thus be implemented in $\mathrm{O}(\max\{1, t/d\})$ rounds. The choice of t is such that the number of rounds in the first stage always dominates, and gives us the stated bounds.

We describe the first stage next. In order to identify a subset I containing an index i as above, if there is any, we implement a search algorithm based on a quantum walk on the Johnson Graph $J(n, t)$, following the framework due to Magniez, Nayak, Roland, and Santha [MNRS11]. The vertices of $J(n, t)$ are t-subsets of $[n]$. There is an edge between two vertices I, I' in $J(n, t)$ iff I and I' differ in exactly 2 elements: $(I \setminus I') \cup (I' \setminus I) = \{i, j\}$ for distinct elements $i, j \in [n]$.

The three building blocks of such an algorithm are as follows.

Set-up: Construct the following starting superposition:

$$\binom{n}{t}^{-1/2} \sum_{I \subseteq [n] \,:\, |I|=t} |(i, x_i) : i \in I\rangle \ .$$

Checking: Check whether $x_i = y_i = 1$ for some $i \in I$:

$$|(i, x_i) : i \in I\rangle \quad \mapsto \quad \begin{cases} -|(i, x_i) : i \in I\rangle, & \text{if } x_i = y_i = 1 \text{ for some } i \in I \ ; \\ \ \ |(i, x_i) : i \in I\rangle, & \text{otherwise.} \end{cases}$$

Update: Replace some index $j \in I$ by an index $k \notin I$, and update the corresponding bit x_j to x_k:

$$|(i, x_i) : i \in I\rangle|j\rangle|k\rangle \quad \mapsto \quad |(i, x_i) : i \in (I \setminus \{j\}) \cup \{k\}\rangle|k\rangle|j\rangle \ .$$

Let ε be the probability that a uniformly random t-subset of $[n]$ contains an index i such that $x_i = y_i = 1$, given that such an element i exists. We have $\varepsilon \in \Omega(t/n)$. Then, according to Theorem 1.4 in Ref. [MNRS11], there is an algorithm based on quantum walk that finds a subset I such that $x_i = y_i = 1$ for some $i \in I$, if there is any such subset, with constant probability $> 1/2$. The algorithm uses one instance of **Set-up**, and $\mathrm{O}(\sqrt{1/\varepsilon}\,)$ alternations of one instance of **Checking** with a sequence of $\mathrm{O}(\sqrt{t}\,)$ instances of **Update**, interspersed with other unitary operations that are independent of the inputs x, y. (The spectral gap of the Johnson graph needed in the analysis of the algorithm may be derived from the results in Ref. [Knu93], for example.)

Note that **Set-up** uses t queries to x, and thus can be implemented in $\max(1, 2\lceil t/d \rceil)$ rounds. **Update** only requires 2 queries to x. Thus a sequence of $\sqrt{t}$ sequential **Update** operations can be implemented in order $\max(1, 2\sqrt{t}/d)$ rounds. We would like to use the Grover algorithm for unordered search to implement the checking step. The Grover algorithm incurs non-zero probability of error in general, while the algorithm due to Magniez *et al.* assumes that the checking step is perfect. We therefore use an algorithm for unordered search with small error due to Buhrman, Cleve, de Wolf, and Zalka [BCdWZ99] to implement **Checking** with error at most $c\sqrt{t/n}$ for a suitable positive constant c with order $\sqrt{t \log(n/t)}$ queries to y. Using standard arguments, this only increases the error of the quantum walk algorithm by a small constant, say $1/10$. In effect, **Checking** (with the stated error) can be implemented in order $\max(1, \sqrt{t \log n}/d)$ rounds. Thus the bound on the round complexity of the quantum walk algorithm is of the order of

$$\max\left\{1, \frac{t}{d}\right\} + \sqrt{\frac{n}{t}} \left(\max\left\{1, \frac{\sqrt{t \log n}}{d}\right\} + \max\left\{1, \frac{\sqrt{t}}{d}\right\} \right) \ . \tag{4.1}$$

In order to derive the bounds stated in the theorem, we optimise over t. We consider intervals of values for t such that each of the expressions involving maximisation in Eq. (4.1) simplifies to one of the terms. The intervals are given by partitioning $[n]$ at the points $d, d^2/\log n, d^2$. (Note that d need not be smaller than $d^2/\log n$.) We optimise the number of rounds within each interval, which in turn gives us a relation between d and n for which the rounds are minimised.

We first consider $d \leq \log n$, so that $d^2/\log n \leq d$, and t in the intervals

$$[1, d^2/\log n], \quad [d^2/\log n, d], \quad [d, d^2], \quad \text{and } [d^2, n] \ .$$

We optimise the number of rounds with t in each of these intervals, to find that the number of rounds is $\mathrm{O}(\sqrt{n \log n}/d)$ when $t := d$. The optimal values of t in the other intervals also give the same bound, but we stay with $t = d$ so as to minimise the rounds in the (optional) second stage of the algorithm.

Next we consider $d \geq \log n$, so that $d \leq d^2/\log n \leq d^2$. We again optimise over t in four intervals, and get the following bounds:

1. $t \in [1, d]$: $\mathrm{O}(\sqrt{n/d}\,)$ when $t := d$.

2. $t \in [d, d^2/\log n]$: $\mathrm{O}(\sqrt[3]{n/d}\,)$ when $t := \sqrt[3]{nd^2}$, *provided* $d^4 \geq n \log^3 n$. If $d^4 \leq n \log^3 n$, we get $\mathrm{O}(\sqrt{n \log n}/d)$ when $t := d^2/\log n$.

3. $t \in [d^2/\log n, d^2]$: $\mathrm{O}(\sqrt{n \log n}/d)$ when $t := d^2/\log n$ *provided* $d^4 \leq n \log^3 n$. If $d^4 \geq n \log^3 n$, we get $\mathrm{O}(d/\log n)$ with the same value of t.

4. $t \in [d^2, n]$: $\mathrm{O}(\sqrt{n \log n}/d)$ when $t := d^2$ *provided* $d^4 \leq n \log n$. If $d^4 \geq n \log n$, we get $\mathrm{O}(d)$ with the same value of t.

Since $\sqrt{n/d} \geq \sqrt{n \log n}/d$ when $d \geq \log n$, $\sqrt{n \log n}/d \leq d$ when $d^4 \geq n \log n$, and $(n/d)^{1/3} \leq d/\log n$ when $d^4 \geq n \log^3 n$, we conclude the bounds on round complexity stated in the theorem:

- $\mathrm{O}(\sqrt{n \log n}/d)$ with $t := d$ when $d \leq \log n$, or with $t := d^2/\log n$ when $\log^4 n \leq d^4 \leq n \log^3 n$, and
- $\mathrm{O}(\sqrt[3]{n/d}\,)$ with $t := \sqrt[3]{nd^2}$ when $d^4 \geq n \log^3 n$.

The bounds on query complexity follow. ∎

Note that in the range of parameters such that $n \log^2 n \leq d^4 \leq n \log^3 n$, the upper bound $\sqrt{n \log n}/d$ is at most $\sqrt{\log n}$ times the lower bound $\sqrt[3]{n/d \log n}$. So the bounds in Theorems 4.1 and 4.2 are indeed within polylogarithmic factors of each other for all values of d, n (such that $d \leq n$).

5 Conclusion

In this work, we studied a fundamental problem, Set Disjointness, in two concrete computational models. Set Disjointness on the Line $\mathsf{L}_{n,d}$ reveals new subtleties in distributed computation with quantum resources. It again puts the spotlight on the "double counting" of information in conditional information loss (and leakage). One may think that the more sophisticated notion of *quantum information cost* introduced by Touchette [Tou15], along with the results due to Braverman *et al.* [BGK$^+$18], might help us overcome this drawback. Indeed, quantum information cost helps us overcome the limitations of the former quantity in the case of Set Disjointness in the standard two-party communication model. Surprisingly, these techniques do not seem to help in obtaining a better lower bound for $\mathsf{L}_{n,d}$. (An analysis of the quantum information cost of the two-party protocol derived in Lemma 3.3, along with the lower bound in this quantity given by Ref. [BGK$^+$18], gives us a bound that is a poly-logarithmic factor smaller than the one we derive in Theorem 3.1.) We believe that new ideas may be needed to characterise its asymptotic round complexity.

The two-oracle query model we introduce gives us a different perspective on Set Disjointness on a Line. It implies that any improvement to the round lower bound for $\mathsf{L}_{n,d}$ would necessarily go beyond the use of conditional information leakage for two-party protocols for Set Disjointness. More generally, the new query model is tailored towards the study of distributed algorithms on the line and could shed light on protocols for other similar problems. Moreover, the model could also be of relevance in other distributed computation scenarios.

References

[AA03] Scott Aaronson and Andris Ambainis. Quantum search of spatial regions. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2003)*, pages 200–209. IEEE Computer Society, October 2003.

[Amb19] Andris Ambainis. Understanding quantum algorithms via query complexity. In *Proceedings of the International Congress of Mathematicians (ICM 2018)*, pages 3265–3285. World Scientific, 2019.

[BCdWZ99] Harry Buhrman, Richard Cleve, Ronald de Wolf, and Christof Zalka. Bounds for small-error and zero-error quantum algorithms. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS 1999)*, pages 358–368, USA, October 1999. IEEE Computer Society.

[BdW02] Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002.

[BGK+18] Mark Braverman, Ankit Garg, Young Kun Ko, Jieming Mao, and Dave Touchette. Near-optimal bounds on the bounded-round quantum communication complexity of Disjointness. *SIAM Journal on Computing*, 47(6):2277–2314, 2018.

[CP10] Arkadev Chattopadhyay and Toniann Pitassi. The story of Set Disjointness. *SIGACT News*, 41(3):59–85, September 2010.

[dW19] Ronald de Wolf. Quantum computing: Lecture notes, 2019.

[FHW12] Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. Networks cannot compute their diameter in sublinear time. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012)*, pages 1150–1162, 2012.

[HW12] Stephan Holzer and Roger Wattenhofer. Optimal distributed all pairs shortest paths and applications. In *Proceedings of the 2012 ACM Symposium on Principles of Distributed Computing (PODC 2012)*, pages 355–364, 2012.

[JRS03a] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A lower bound for bounded round quantum communication complexity of Set Disjointness. e-print arXiv:quant-ph/0303138v2, arXiv.org, April 2003.

[JRS03b] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A lower bound for the bounded round quantum communication complexity of Set Disjointness. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2003)*, pages 220–229. IEEE Computer Society Press, 2003.

[Knu93] Donald E. Knuth. Combinatorial matrices. Manuscript available at `http://www-cs-faculty.stanford.edu/~knuth/preprints.html#unpub`, 1993.

[LM18] François Le Gall and Frédéric Magniez. Sublinear-time quantum computation of the diameter in CONGEST networks. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing (PODC 2018)*, pages 337–346. Association for Computing Machinery, 2018.

[MN16] Frédéric Magniez and Ashwin Nayak. Near-linear lower bounds for distributed distance computations, even in sparse networks. In Cyril Gavoille and David Ilcinkas, editors, *Distributed Computing: 30th International Symposium, DISC 2016, Paris, France, September 27–29, 2016. Proceedings*, volume 9888 of *Theoretical Computer Science and General Issues*, pages 29–42. Springer-Verlag Berlin Heidelberg, 2016.

[MN20] Frédéric Magniez and Ashwin Nayak. Quantum distributed complexity of set disjointness on a line. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 82:1–82:18, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[MNRS11] Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via quantum walk. *SIAM Journal on Computing*, 40:142–164, 2011.

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2000.

[Pel00] David Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, USA, 2000.

[PRT12] David Peleg, Liam Roditty, and Elad Tal. Distributed algorithms for network diameter and girth. In *Proceedings of the 39th Internatinal Colloquium on Automata, Languages, and Programming (ICALP 2012)*, pages 660–672, 2012.

[Raz03] Alexander Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya: Mathematics*, 67(1):145–159, 2003. Russian version in *Izvestiya Rossiiskoi Academii Nauk (seriya matematicheskaya) 67* (2003), 1, 159–176.

[Tou15] Dave Touchette. Quantum information complexity. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing (STOC 2015)*, pages 317–326, New York, NY, USA, 2015. ACM.

[Wat18] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, May 2018.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing (STOC 1979)*, pages 209–213, New York, NY, USA, 1979. Association for Computing Machinery.

[Yao93] Andrew Chi-Chih Yao. Quantum circuit complexity. In *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1993)*, pages 352–361. IEEE Computer Society Press, 1993.

[Zal99] Christof Zalka. Grover's quantum searching algorithm is optimal. *Physical Review A*, 60:2746–2751, October 1999.

A Conditional information leakage of Set disjointness

Theorem 3.5, the lower bound on the conditional information leakage of bounded-round protocols for Set Disjointness due to Jain, Radhakrishnan, and Sen [JRS03b] is not stated explicitly in their article. In this section, we explain how the theorem may be inferred from their work.

Jain *et al.* implicitly analyse the conditional information leakage of protocols for Set Disjointness and relate it to the conditional information loss of a protocol for the two-bit AND function. For completeness, we define the latter quantity using the same notation as for the conditional information leakage of a quantum communication protocol Π introduced in Sec. 2.3.

The *conditional information loss* $\mathrm{IL}(\Pi \,|\, XYZ)$ of the protocol Π is defined as

$$\mathrm{IL}(\Pi \,|\, XYZ) \quad := \sum_{i\in[m],\ i \text{ odd}} \mathrm{I}(X : B_i Y \,|\, Z) + \sum_{i\in[m],\ i \text{ even}} \mathrm{I}(Y : A_i X \,|\, Z) \ ,$$

where the registers are implicitly assumed to be in the state given by the protocol. The difference between conditional information leakage and loss lies in the inclusion of the purification registers $\hat{X}$ and $\hat{Y}$ in the mutual information terms in the former quantity. By the Data Processing Inequality (Lemma 2.1), the conditional information leakage of a protocol is at least as large as its conditional information loss.

Jain *et al.* prove an $\Omega(n/m^2)$ lower bound on the communication required by any m-round entanglement-assisted two-party quantum communication protocol Γ for Set Disjointness. They show this in three steps. In the first step, they bound the conditional information leakage $\widetilde{\mathrm{IL}}(\Gamma|XYZ)$ by $2mc$, where c is the total number of qubits exchanged in Γ, and XYZ are any jointly distributed random variables such that $X, Y \in \{0,1\}^n$ and X and Y are independent given Z. In the second step, they show that there is a specific distribution for XYZ with the properties stated above, a protocol Γ' for the two-bit AND function (derived from Γ), and jointly distributed random variables $X'Y'Z'$ with $X', Y' \in \{0,1\}$ such that conditional information *loss* $\mathrm{IL}(\Gamma'|X'Y'Z')$ is at most $\widetilde{\mathrm{IL}}(\Gamma|XYZ)/n$. In the third step, they show that $\mathrm{IL}(\Gamma'|X'Y'Z')$ for AND is at least $\Omega(1/m)$. Theorem 3.5 follows by combining the last two steps. The auxiliary random variable Z and the conditioning on this random variable are what enable the *direct sum* property underlying the reduction in the second step.

The purification registers used in conditional information leakage are required so that the joint state of the two parties in the protocol Γ' for the AND function derived from Γ on any fixed input is pure until the measurement used for producing the output. This property is crucial for the third step of the proof described above. (It turns out, though, that for the distribution $X'Y'Z'$ they derive, the conditional information leakage of Γ' coincides with its conditional information loss.)

Jain *et al.* present their communication lower bound in more generality, for t-party protocols, for $t \geq 2$, and for a class of functions that includes Set Disjointness. The first two steps are proven together in Lemma 2 in Ref. [JRS03b], and the first step can be inferred from the derivation of Eq. (1) in the proof. The third step is proven in Lemma 3.

Theorem 3.5 may be easier to infer from the arXiv pre-print [JRS03a], as this version concerns two-party protocols. Conditional information loss is presented in Definition 5, the first two steps described above are proven together in Lemma 3, and the third step is proven in Lemma 4.

B States in two-party communication protocols

As it appears not to be well-known, we include a result on the structure of the joint states in a two-party quantum communication protocol *without* shared entanglement, and a consequence of relevance to us. We

state the result in the notation given in Section 2.3.

Lemma B.1 (Yao [Yao93])**.** *Consider a two-party quantum communication protocol without shared entanglement in which Alice gets input x and Bob gets input y. The joint state $|\psi_k\rangle$ of the work registers A_kB_k at the end of the k-th round of the protocol may be expressed as*

$$|\psi_k\rangle \quad = \quad \sum_{c \in \{0,1\}^{q_k}} |\phi(x,c)\rangle^{A_k} \otimes |\xi(y,c)\rangle^{B_k} \quad ,$$

where q_k is the sum of the lengths of the first k messages, and $|\phi(x,c)\rangle$ and $|\xi(y,c)\rangle$ are possibly non-normalised states of appropriate dimension that depend only on x, c and y, c, respectively.

Proof: We prove the statement by induction over k. For $k = 0$, we have $q_k := 0$, and we may write the state of A_0B_0 as

$$|\psi_0\rangle \quad = \quad |\bar{0}\rangle^{A_0} \otimes |\bar{0}\rangle^{B_0} \quad .$$

Assume that the statement holds for $k = j$, for some $j \geq 0$, so that

$$|\psi_j\rangle \quad = \quad \sum_{c \in \{0,1\}^{q_j}} |\phi(x,c)\rangle^{A_j} \otimes |\xi(y,c)\rangle^{B_j} \quad ,$$

with q_j, $|\phi(x,c)\rangle$, and $|\xi(y,c)\rangle$ as in the statement of the lemma.

Consider round $j+1$. Suppose that Alice sends the message in the $(j+1)$-th round; the other case is analogous. Suppose Alice applies the isometry U_x (depending on her input x) to the work register A_j to obtain registers $A_{j+1}M_{j+1}$, and $|\phi'(x,c)\rangle := U_x|\phi(x,c)\rangle$. She then sends the message register M_{j+1} to Bob. Bob's work register at the end of the $(j+1)$-th round is then $B_{j+1} := M_{j+1}B_j$. Suppose M_{j+1} consists of q qubits. We may express each state $|\phi'(x,c)\rangle$ as

$$|\phi'(x,c)\rangle \quad = \quad \sum_{c' \in \{0,1\}^{q}} |\phi(x,cc')\rangle^{A_{j+1}} \otimes |c'\rangle^{M_{j+1}} \quad ,$$

for suitable non-normalised states $|\phi(x,cc')\rangle$ of appropriate dimension. So we may write the state $|\psi_{j+1}\rangle$ as

$$|\psi_{j+1}\rangle \quad = \quad \sum_{c \in \{0,1\}^{q_j}} \sum_{c' \in \{0,1\}^{q}} |\phi(x,cc')\rangle^{A_{j+1}} \otimes |\xi(y,cc')\rangle^{B_{j+1}} \quad ,$$

where

$$|\xi(y,cc')\rangle \quad := \quad |c'\rangle^{M_{j+1}} \otimes |\xi(y,c)\rangle^{B_j} \quad .$$

This proves the lemma. ∎

The lemma implies that the state of the register B_m at the end of an m-round protocol has support in the linear span of the states $\{|\xi(y,c)\rangle : c \in \{0,1\}^m\}$, when the input given to Bob is y. When Bob has no input, the support has dimension at most 2^{q_m}, independent of Alice's input. We may thus define an isometry W such that $W|\xi(c)\rangle = |\xi'(c)\rangle \otimes |\bar{0}\rangle$, where $|\xi'(c)\rangle$ is a q_m-qubit state. Effectively, we need at most q_m qubits to store the final state. This argument may be extended to all the previous rounds, i.e., we may define suitable isometries to store the states in the previous rounds in q_m qubits. Let W_j be the isometry used for this purpose at the end of the j-th round, for $j \geq 0$. We may also modify the isometry V_j applied by Bob in the j-th round to $V_j' := W_jV_jW_{j-1}^*$. If Bob performs the measurement to produce the output

of the protocol, the measurement may be modified similarly. Thus, we get a protocol in which Bob uses at most q_m work qubits throughout.

In the context of the second paragraph after Lemma 3.3, for a fixed $i \in [d-1]$, suppose Bob simulates the actions of party A_i, while Alice simulates the actions of all the other parties. Then the protocol Π_d translates to a two-party protocol with $2r$ rounds and total communication of $4rb$. The isometries used by A_i are all independent of the inputs to Π_d. The claim made in the said paragraph then follows.