Communication memento: Memoryless communication complexity

Srinivasan Arunachalam  
IBM Research, USA  
Srinivasan.Arunachalam@ibm.com

Supartha Podder  
University of Ottawa, Canada  
spodder@uottawa.ca

Abstract

We study the communication complexity of computing functions $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ in the memoryless communication model. Here, Alice is given $x \in \{0, 1\}^n$, Bob is given $y \in \{0, 1\}^n$ and their goal is to compute $F(x, y)$ subject to the following constraint: at every round, Alice receives a message from Bob and her reply to Bob solely depends on the message received and her input $x$ (in particular, her reply is independent of the information from the previous rounds); the same applies to Bob. The cost of computing $F$ in this model is the maximum number of bits exchanged in any round between Alice and Bob (on the worst case input $x, y$). In this paper, we also consider variants of our memoryless model wherein one party is allowed to have memory, the parties are allowed to communicate quantum bits, only one player is allowed to send messages and the relationship between our communication model and the garden-hose model of computation. Restricted versions of our communication model were studied before by Brody et al. (ITCS’13) and Papakonstantinou et al. (CCC’14), in the context of space-bounded communication complexity.

In these models, we establish the following main results: (1) We show that the memoryless communication complexity of $F$ characterizes the logarithm of the size of the smallest bipartite branching program computing $F$ (up to a factor 2); (2) We give exponential separations between various classical variants of memoryless communication models; (3) We exhibit exponential quantum-classical separations in the four variants of the memoryless communication model;

We end with an intriguing open question: can we find an explicit function $F$ and universal constant $c > 1$ for which the memoryless communication complexity is at least $c \log n$? Note that $c \geq 2 + \varepsilon$ would imply a $\Omega(n^{2+\varepsilon})$ lower bound for general formula size, improving upon the best lower bound by Nečiporuk [Nec66].

1 Introduction

Yao [Yao79] introduced the model of communication complexity in 1979 and ever since it’s introduction, communication complexity has played a pivotal role in understanding various problems in theoretical computer science. In its most general form in this model, the goal is the following: there are two separated parties usually referred to as Alice and Bob, Alice is given an $n$-bit string $x \in \{0, 1\}^n$ and similarly Bob is given $y \in \{0, 1\}^n$ and together they want to compute $F(x, y)$ where $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ is a function known to both of them. Here Alice and Bob are given unlimited computational time and memory and the cost of any communication protocol between Alice and Bob is the total number of bits exchanged between them. Clearly a trivial protocol is Alice sends her input $x$ to Bob who can then compute $F(x, y)$, which takes $n$ bits of communication. Naturally, the goal in communication complexity is to minimize the number of bits of communication.
between them before computing $F(x, y)$. The deterministic communication complexity of a function $F$ (denoted $D(F)$) is defined as the total number of bits of communication before they can decide $F(x, y)$ on the worst-case inputs $x, y$.

Since its introduction there have been various works that have extended the standard deterministic communication model to the setting where Alice and Bob are allowed to share randomness and need to output $F(x, y)$ with high probability (probability taken over the randomness in the protocol). Apart from this there has been studies on non-deterministic communication complexity [Wal03], quantum communication complexity [Yao93] (wherein Alice and Bob are allowed to share quantum bits and possibly have shared entanglement), unbounded error communication complexity [PS86] and their variants. One-way variants have also been considered where only Alice sends messages to Bob. Study of these different models of communication complexity and their variants have provided many important results in the fields of VLSI [Pal99], circuit lower bounds [GH92], algorithms [AMS99], data structures [MNSW98], property testing [BBM12], streaming algorithms [BYJKS04], computational complexity [BW16], extended formulations [FMP+15].

1.1 Background

In the context of our current understanding of computation, the study of space required to solve any problem is one of the central topics in complexity theory. Several space bounded models such as width-bounded branching programs, limited depth circuits, straight line programs have been widely studied in this context. In this direction variants of communication complexity have also been analyzed to better understand communication-space trade-offs [IW10, Kla04, LTT89]. In particular, the relation between space-bounded computation and communication complexity was initiated by Brody et al. [BCP+13] who considered the following question: what happens if we change the standard communication model such that, between each step of communication, Alice and Bob limited in their ability to store the information from the previous rounds (which includes their private memory and messages exchanged). In this direction, they introduced a new model wherein Alice and Bob each are allowed to store at most $s(n)$ bits of memory and showed that unlike the standard communication complexity, in this model super-linear lower bounds on the amount of communication are possible\(^2\). Brody et al. mainly studied one-way communication complexity variant of this limited memory model in which Bob can have two types of memory: an oblivious memory (depends only on Alice’s message) and a non-oblivious memory (for computation). With these definitions, among other results they obtained memory hierarchy theorems for such communication models analogous to the space hierarchy theorem in the Turing machine world.

Subsequently, Papakonstantinou et al. [PSS14] defined a similar space-bounded one-way communication model wherein Alice has unlimited memory and Bob has either no memory or constant-sized memory. At each round, messages from Alice to Bob consists of at most $t(n)$ bits and the complexity of computing any function is the maximum $t(n)$ required over all inputs. They characterized the complexity in their no-memory one-way model by an elegant combinatorial object called the rectangle overlay (which is defined in Section 4.2). They also managed to establish connections between their model and the well-known communication complexity polynomial hierarchy, introduced by Babai, Frankl and Simon [BFS86]. Papakonstantinou et al. [PSS14] showed that the

\(^1\)For more on communication complexity and its applications, we refer the interested reader to the standard textbooks for communication complexity [KN97, LS09].

\(^2\)A separation were proven for a non-Boolean function.
message length in their model corresponds to the oblivious memory in a variant of space bounded model, introduced by Brody et al. [BCP\textsuperscript{+}13], where Bob only has access to an oblivious memory.

Another seemingly unrelated complexity model, the garden-hose complexity was introduced by Buhrman et al. [BFSS\textsuperscript{13}] to understand quantum attacks on position-based cryptographic schemes (see Section 5.2 for a formal definition). Polynomial size garden-hose complexity is known to be equivalent to Turing machine log-space computations with pre-processing. In the garden-hose model two distributed players Alice and Bob use several pipes to send water back and forth and compute Boolean functions based on whose side the water spills. Garden-hose model was shown to have many connections to well-established complexity models like formulas, branching programs and circuits and it provides new techniques to prove lower bounds for these complexity models.

In this work, we introduce a new general framework of memoryless communication complexity which captures all the above variants of the space-bounded models.

1.2 Memoryless Communication Models

Memoryless communication models. We introduce a natural model of communication complexity which we call memoryless communication complexity. Here, like the standard communication complexity, there are two parties Alice and Bob given \(x, y\) respectively and they need to compute \(F(x, y)\), where \(F : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}\) is known to both of them. However, we tweak the standard communication model in the following two ways: The first change is that Alice is “memoryless”, i.e., at every round Alice computes the next message to send solely based on only her input \(x\) and the message received from Bob in this round. She does not remember the entire transcript of messages that were communicated in the previous rounds and also forgets all the private computation she did in the previous rounds. Similarly Bob computes a message which he sends to Alice, based only on his input \(y\) and the message received from Alice in the current round. After Bob sends his message, he also forgets the message received and all his private computations. Alice and Bob repeat this procedure for a certain number of rounds before one of them outputs \(F(x, y)\).

The second crucial change in the memoryless communication model is that the cost of computing \(F\) in this model is the size of the largest message communicated between Alice and Bob in any round of the protocol (here size refers to the number of bits in the message). Intuitively, we are interested in knowing what is the size of a re-writable message register (passed back and forth between Alice and Bob) required to compute a function \(F\) on all distributed inputs \(x\) and \(y\), wherein Alice and Bob do not have any additional memory to remember information between rounds.\(^3\) We denote the memoryless communication cost of computing \(F\) as \(\text{NM}(F)\) (where \(\text{NM}\) stands for “no-memory”). We believe this communication model is very natural and as far as we are aware this memoryless communication model wasn’t defined and studied before in the classical literature.

It is worth noting that in the memoryless communication model, Alice and Bob do not even have access to clocks and hence cannot tell in which round they are in (without possibly looking at the message register). Hence, every memoryless protocol can be viewed as Alice and Bob applying deterministic functions (depending on their respective inputs) which map incoming messages to out-going messages.

\(^3\)Note that unlike the standard communication complexity, where a single bit-message register suffices for computing all functions, here because of the memoryless-ness constraint we need more than a single bit register for computing most of the functions.
In order to get a feel of this model, let us look at a protocol for the standard equality function defined as $\text{EQ}_n : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$ where $\text{EQ}_n(x,y) = 1$ if and only if $x = y$. It is well-known that $D(\text{EQ}_n) = n$. In our model, we show that $\text{NM}(\text{EQ}_n) \leq \log n + 1$: for $i = 1, \ldots, n$, at the $i$th round, Alice sends $(i, x_i)$ and Bob returns $(i, [x_i = y_i])$. Alice increments $i$ and repeats this protocol for $n$ rounds. In case Bob finds an $i$ for which $x_i \neq y_i$, he outputs 0, if not after $n$ rounds they output 1. Note that this protocol didn’t require Alice and Bob to have any memory and the length of the longest message in this protocol was $\log n + 1$. We discuss more protocols later in the paper and formally describe the memoryless communication model in Section 3.

Apart from memoryless communication complexity, we will also look at the “memory-nomemory communication” protocols where Alice is allowed to have memory (i.e., Alice can know which round she is in, can remember the entire transcript and her private computations of each round) whereas Bob doesn’t have any memory during the protocol. The goal of the players remains to compute a function $F$ and the cost of these protocols (denoted by $M(F)$) is still defined as the smallest size of a message register required between them on the worst inputs. Apart from this, we will also consider the quantum analogous of these two communication models wherein the only difference is that Alice and Bob are allowed to send quantum bits. We formally describe these models of communication in Section 3. In order to aid the reader we first set up some notation which we use to describe our results: for $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$, let

1. $\text{NM}(F)$ be the memoryless communication complexity of computing $F$ wherein Alice and Bob both do not have any memory.
2. $M(F)$ be the memory-nomemory communication complexity of computing $F$ where Alice has memory and Bob doesn’t have memory.
3. $\text{GH}(F)$ be the garden-hose complexity of computing $F$.

Apart from these, we will also allow quantum bits of communication between Alice and Bob and the complexities in these models are denoted by $\text{QNM}(F)$ and $\text{QM}(F)$. Additionally, we will consider the one-way communication models wherein only Alice can send messages to Bob and the complexities in these models are denoted by $\text{NM}^{-}(F), M^{-}(F), \text{QNM}^{-}(F), \text{QM}^{-}(F)$.

### 1.3 Our Contribution

**Defining and characterizing the model.** The main contribution in this paper is to first define the model of the memoryless communication complexity and consider various variants of this model (only some of which were looked at before in the literature). We provide a characterization of memoryless communication complexity using branching programs. In particular, we show that for every $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$, the memoryless complexity $\text{NM}(F)$ is (up to a factor 2) equal to the logarithm of the size of the smallest bipartite branching program computing $F$. We defer the definition of such branching programs to Section 2 and Section 4.2.

**Separating these models.** We then establish the following inequalities relating the various models of communication complexity.\(^5\)

\(^4\)Here $[\cdot]$ is the indicator of an event in the parenthesis.

\(^5\)Some of the inequalities are straightforward but we explicitly state it for completeness.
\[
\begin{align*}
M(F) & \leq \text{NM}(F) < \log(\text{GH}(F)) \leq M^{-}(F) \leq \text{NM}^{-}(F) \\
\text{QM}(F) & \leq \text{QNM}(F) \leq \text{QM}^{-}(F) \leq \text{QNM}^{-}(F)
\end{align*}
\]

Furthermore, except the inequality marked by $\star$, we show the existence of various functions $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ for which all the inequalities are exponentially weak. In order to prove these exponential separations we use various variants of well-known functions such as inner product, disjointness, Boolean hidden matching problem, gap-hamming distance problem. Giving exponential separations between quantum and classical communication complexity$^6$ is an extensively studied subject [BCW98, BCWW01, GKK$^+$08, BJK08, Gav19] and in this paper we show such separations can also be obtained in the memoryless models.

Along the way, we establish various other results. In order to understand the limitations of the memoryless communication model, we show that the logarithm of the (standard) non-deterministic communication complexity of $F$ is a lower bound on $\text{NM}(F)$. In particular, this provides tight bounds for many interesting problems like equality (a matching upper bound was described at the start of this section), inner product, disjointness, majority.

**Relevance to garden-hose complexity and its implications.** We show a relation between the garden-hose complexity to memoryless communication complexity. In particular we show that the logarithm of the garden-hose complexity of computing $F$ is sandwiched between $\text{NM}(F)$ and $M^{-}(F)$. Moreover, using the results of Papakonstantinou et al. [PSS14] that characterized $M^{-}(F)$ in terms of rectangle overlays, we present a new upper bound technique for the garden-hose model. Using this method we obtain a sub-quadratic garden-hose protocol for computing the function $\text{Disjointness with quadratic universe}$ which was conjecture to have a quadratic complexity in [KP14].

An additional consequence of this is the following: in the work of Papakonstantinou, et al. [PSS14] it was stated that the message length in their one-way memory-no memory model corresponds to oblivious memory setting considered by Brody et al. [BCP$^+$13]. However, Brody et al. [BCP$^+$13] pointed out that if the memory required to compute a function $F$ in their general model is $\text{OM}(f)$ and the garden hose complexity is $\text{GH}(f)$ then $\text{OM}(f) \leq \log \text{GH}(f) \leq 2\text{OM}(f)$. We exhibit a function for which $\log \text{GH}(f)$ and $M^{-}(f)$ are exponentially separated. Thus together with the result of [BCP$^+$13] we show that $\text{OM}(f)$ are $M^{-}(f)$ are exponentially separated.

**Towards obtaining better formula bounds.** Finally, it was shown by Klauck and Podder [KP14] that any formulae of size $s$ consisting of arbitrary fan-in 2 gates (i.e., formulae over the binary basis of fan-in 2 gates) can be simulated by a garden-hose protocol of size $s^{1+\varepsilon}$ for any arbitrary $\varepsilon > 0$. In this work, we show that an arbitrary garden-hose protocol can be simulated by a memoryless protocol without any additional loss, i.e., a size $s$ garden-hose protocol can be turned into a memoryless protocol of size $\log s$. In particular, putting together these two connections, it implies that a size $s$ formula can be turned into a memoryless protocol of size $(1 + \varepsilon) \log s$. Thus our result provides a new way of proving formulae size lower bound for arbitrary function $F$ by analyzing the memoryless protocol of $F$.$^7$ The best known lower bound for formulae size (over the basis of all fan-in 2 gate)

---

$^6$These exponential separations are in the standard communication model where the communication complexity is the total number of bits or qubits exchanged between Alice and Bob.

$^7$Here, the inputs $x, y$ are distributed among two players and their goal is to compute $(F \circ g)(x, y)$ where $g$ is a constant-sized gadget.
is $\Omega(n^2/\log n)$, due to Nečiporuk [Nec66]. Analogous to the Karchmer-Wigderson games [KW90] and Goldman and Håstad [GH92] techniques which uses communication complexity framework to prove circuit lower bounds our new communication complexity framework is a new tool for proving formulae size lower bounds.

1.4 Further remarks

Directions for future works. As a main open problem, we mention a seemingly-simple intriguing open question: Is there any explicit function $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$ for which $\text{NM}(F) \geq c \cdot \log n$ for an arbitrary constant $c > 1$ (we discuss further implications of this result in Section 6). As mentioned earlier, proving $c \geq 2 + \varepsilon$ for some explicit function $F$ would imply new lower bounds for general formulae size, improving the best-known $\Omega(n^2/\log n)$ lower bounds by Nečiporuk [Nec66] from 1966.\(^8\)

Also using our characterized to branching programs, $c \geq 1+\varepsilon$ would result in the first super-linear lower bound for bipartite branching programs (analogous to Tal’s first super-linear lower bound on bipartite formula size of inner-product [Tal16]). One of the possible candidates for such lower bound could be distributed 3-clique function. We discuss these and several other open problems for future research in Section 6.

Other related works. Finally here we discuss some more related works. Impagliazzo and Williams [IW10] considered two variants of the standard communication complexity model where players have access to a synchronized clock. They then studied their relationship with the standard communication complexity and the polynomial hierarchy. In the quantum setting, Ablayev et al. [AAKK18] consider the memoryless communication model we consider and their focus was on discussing its connections to proving lower bounds on automata and ordered binary decision diagrams and streaming algorithms. Chailloux et al. [CKL17] study the quantum memoryless communication complexity wherein Alice and Bob do not have private memory but are allowed to adaptively apply unitaries to the quantum states they exchange in each round. They study the information cost of memoryless quantum protocols and prove a tight lower bound on the information cost of the AND function for k-round quantum memoryless protocols. Buhrman et al. [BCG+16] studied quantum memoryless protocols and established a connection between memoryless protocols and Bell inequality violations. Jeffery [Jef20] recently related the space complexity of quantum query algorithms with approximate span programs.

Organization. In Section 2 we describe the basic communication model and branching programs. In Section 3 we describe the memoryless communication as well as other variants of this model. In Section 4 we characterize the complexity of memoryless communication in terms of bipartite branching programs. In Section 5 we present various algorithms, lower bounds and separations between the memoryless models and we finally conclude with some open questions in Section 6.

\(^8\)Additionally, proving $c \geq 3 + \varepsilon$ would improve the best-known $\Omega(n^3)$ De Morgan formula size lower bounds by Gal et al. [GTN19].
2 Preliminaries

Notation. Let \([n] = \{1, \ldots, n\}\). For \(x \in \{0, 1\}^n\), let \(\text{Int}(x) \in \{0, \ldots, 2^n - 1\}\) be the integer representation of the \(n\)-bit string \(x\). We now define a few standard functions which we use often in this paper. The equality function \(\text{EQ}_n : \{0, 1\}^n \to \{0, 1\}^n \to \{0, 1\}\) is defined as \(\text{EQ}_n(x, y) = 1\) if and only if \(x = y\). The disjointness function \(\text{DISJ}_n\) defined as \(\text{DISJ}_n(x, y) = 0\) if and only if there exists \(i\) such that \(x_i = y_i = 1\). The inner product function \(\text{IP}_n\) is defined as \(\text{IP}(x, y) = \sum_i x_i \cdot y_i \mod 2\) (where \(\cdot\) is the standard bit-wise product).

Quantum information. We briefly review the basic concepts in quantum information theory. Here a qubit \(|\psi\rangle\) is a unit vector in \(\mathbb{C}^2\) and the basis for this space is denoted by \(|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}\) and \(|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}\). An arbitrary \(|\psi\rangle\) is in a superposition of \(|0\rangle, |1\rangle\), i.e., \(|\psi\rangle = \alpha|0\rangle + \beta|1\rangle\) for \(\alpha, \beta \in \mathbb{C}\) satisfying \(|\alpha|^2 + |\beta|^2 = 1\). In order to obtain a quantum state on \(n\) qubits, one can take the tensor product of single-qubit states, hence an arbitrary \(n\)-qubit state \(|\phi\rangle\) is a unit vector in \(\mathbb{C}^{2^n}\) and can be expressed as \(|\psi\rangle = \sum_{x \in \{0, 1\}^n} \alpha_x |x\rangle\) where \(\alpha_x \in \mathbb{C}\) and \(\sum_x |\alpha_x|^2 = 1\).

We now define formulae, branching programs and refer the interested reader to Wegener’s book [Weg87] for more on the subject.

Definition 2.1 (De Morgan Formulae) A De Morgan formula is a binary tree whose internal nodes are marked with AND gates or OR gates and the leaves are marked with input variables \(x_1, x_2, \cdots, x_n\) or their negations. We say a depth-\(d\) De Morgan formula computes a function \(f : \{0, 1\}^n \to \{0, 1\}\), if there is a binary tree of depth \(d\) such that on input \(x = (x_1, x_2, \cdots, x_n)\), the root of the tree outputs \(f(x)\). The De Morgan formula size of a function \(f\) is the size of the smallest De Morgan formula computing \(f\).

Definition 2.2 (General Formulae) General formulas are same as De Morgan formulae except that the nodes could consist of gates corresponding to any arbitrary 2-bit functions \(f : \{0, 1\}^2 \to \{0, 1\}\). Note that De Morgan formulae consisted only of AND, OR gates whereas in general formulae there could be 16 such possible gates in the tree.

Definition 2.3 (Branching programs (BP)) A branching program for computing a Boolean function \(f : \{0, 1\}^n \to \{0, 1\}\) is a directed acyclic graph with a source node labelled \(S\) and two sink nodes labelled 0 and 1. Every node except the source and sink nodes are labelled by an input variable \(x_i\). The out-degree of every node is two and the edges are labelled by 0 and 1. The source node has in-degree 0 and the sink nodes have out-degree 0. The size of a branching program is the number of nodes in it. We say a branching program computes \(f\) if for all \(x \in f^{-1}(1)\) (resp. \(x \in f^{-1}(0)\)) the algorithm starts from the source, and depending on the value of \(x_i \in \{0, 1\}\) at each node the algorithm either moves left or right and eventually reaches the 1-sink (resp. 0-sink) node. We denote \(BP(f)\) as the size (i.e., the number of nodes) of the smallest branching program that computes \(f\) for all \(x \in \{0, 1\}^n\).

We now define the standard communication complexity model defined by Yao [Yao79].

Definition 2.4 (Standard communication complexity) Let \(F : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}\). Here two players Alice and Bob want to compute \(F\) in the following manner: Alice receives \(x \in\)
\{0,1\}^n$, Bob gets $y \in \{0,1\}^n$ and they are allowed to exchange bits before computing $F(x,y)$. We say a protocol computes $F$ if for every $x,y$, Alice and Bob compute $F(x,y)$ with probability 1. The models we describe below vary in how the communication protocol proceeds between Alice and Bob and measures the complexity of the protocols in a different way.

1. **Standard one-way communication complexity:** Here we restrict only Alice to send bits to Bob. The complexity of the protocol in this model is the total number of communication between Alice and Bob and finally Bob needs to output $F(x,y)$. The classical complexity of this model is denoted $D^+(F)$. Suppose they exchange quantum bits, then due to the inherent randomness in quantum states, we allow them to output $F(x,y)$ with probability at least 2/3. The quantum complexity of computing $F$ in this model is denoted by $Q^+(F)$.

2. **Standard two-way communication complexity:** This is exactly the same as $D^+(F)$, except that both Alice and Bob are allowed to send bits to one another and the classical complexity is denoted $D(F)$. Suppose they are allowed to exchange quantum bits, then the quantum complexity is denoted $Q(F)$.

### 3 Memoryless Communication Complexity

In this section we define memoryless communication complexity model and its variants.

#### 3.1 Deterministic Memoryless Communication Model

The crucial difference between the memoryless communication model and standard communication model is that, at any round of the communication protocol Alice and Bob do not have memory to remember previous transcripts and their private computations from the previous rounds. We now make this formal.

**Definition 3.1 (Two-way Deterministic memoryless communication complexity)** Let $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$. Here there are two parties Alice and Bob whose goal is to compute $F$. Every $s$-bit memoryless protocol is defined by a set of functions $\{f_x\}_{x \in \{0,1\}^n}$ and $\{g_y\}_{y \in \{0,1\}^n}$ wherein $f_x,g_y : \{0,1\}^s \rightarrow \{0,1\}^s$. On input $x,y$ to Alice and Bob respectively a memoryless protocol is defined as follows: at every round Alice obtains a message $m_B \in \{0,1\}^s$ from Bob, she computes $m_A = f_x(m_B) \in \{0,1\}^s$ and sends $m_A$ to Bob. On receiving $m_A$, Bob computes $m'_B = g_y(m_A)$ and replies with $m'_B \in \{0,1\}^s$ to Alice. They alternately continue doing this for every round until the protocol ends. Without loss of generality we assume the protocol ends once $m_A,m_B \in \{1^{s-1}0,1^{s-1}1\}$, then the function output is given by the last bit. So, once the transcript is $1^{s-1}b$, Alice and Bob output $F(x,y) = b$.\(^9\)

We say a protocol $P_F$ computes $F$ correctly if for every $(x,y)$, Bob outputs $F(x,y)$. We let $\text{cost}(P_F,x,y)$ be the smallest $s$ for which $P_F$ computes $F$ on input $(x,y)$. Additionally, we let $\text{cost}(P_F) = \max_{x,y} \text{cost}(P_F,x,y)$ and the memoryless communication complexity of computing $F$ in this model is defined as $\text{NM}(F) = \min_{P_F} \text{cost}(P_F)$.

\(^9\)Without loss of generality, we assume that the first message is between Alice and Bob and she sends $f_x(0^s) \in \{0,1\}^s$ to Bob.
where is the minimum is taken over all protocols \( P_F \) that compute \( F \) correctly.

We crucially remark that in the memoryless model, the players do not even have access to a clock and hence they cannot tell which round of the protocol they are in. At every round they just compute their local functions \( \{f_x\}_x, \{g_y\}_y \) on the message they received and proceed according to the output of these functions.

**One-way Deterministic Memoryless Model.** Similar to the definition above, one can define the one-way memoryless communication complexity wherein only Alice is allowed to send messages to Bob and the remaining aspects of this model is the same as Definition 3.1. We denote the complexity in this model by \( \text{NM} \rightarrow (F) \). It is easy to see that since Alice does not have any memory she cannot send multi-round messages to Bob as there is no way for her to remember in which round she is in. Also Bob cannot send messages back to Alice for her to keep a clock. Hence all the information from Alice to Bob has to be conveyed in a single round. Thus one-way memoryless communication complexity is equal to the standard deterministic one-way communication complexity\(^\text{10}\).

**Fact 3.2** For all function \( F \) we have \( \text{NM} \rightarrow (F) = \text{D} \rightarrow (F) \).

### 3.2 Deterministic Memory-No Memory Communication Model

We now consider another variant of the memoryless communication model wherein one party is allowed to have a memory but the other party doesn’t. In this paper, we always assume that Alice has a memory and call this setup the memory no-memory model. In this work, we will not consider the other case wherein Bob has a memory and Alice doesn’t have a memory. Note that this setting is asymmetric i.e., there exists functions for which the complexity of the function can differ based on whether Alice or Bob has the memory.

**Two-way Memory-No Memory Communication Model.** Here the players are allowed to send messages in both directions. For a function \( F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\} \), we denote the complexity in this model as \( \text{M}(F) \). Observe that \( \text{M}(F) \) is trivially upper bounded by \( \log n \) for every \( F \): for every \( i \in [n] \), Alice can send \( i \) and Bob replies with \( y_i \). Since Alice has memory, after \( n \) rounds she has complete knowledge of \( y \in \{0,1\}^n \) and computes \( F(x,y) \) locally and sends it to Bob. Additionally, observe that in this model it doesn’t matter which party has memory.

**One-way Memory-No Memory Communication Model.** Here we allow only Alice to send messages to Bob. Since Alice has a memory she can send multiple messages one after another, but Bob cannot reply to her messages. Hence, after receiving any message Bob computes the function \( g_y(\cdot) \in \{0,1,\bot\} \) and if he obtains \( \{0,1\} \), he outputs \( 0 \) or \( 1 \), and continues if he obtains \( \bot \). We denote the communication complexity in this model by \( \text{M} \rightarrow (F) \). This model was formally studied by Papakonstantinou et al. [PSS14] as overlay communication complexity (we discuss their main contributions in Section 4).

\(^{10}\)Without loss of generality, in any one-way standard communication complexity protocol of cost \( c \) Alice can send all the \( c \) bits in a single round.
Finally, we can also have a model where both players have memory and hence both players can remember the whole transcript of the computation. This is exactly the widely-studied standard communication complexity except that the complexity measure here is the size of the largest transcript (so the complexity in our model is just 1 since they could exchange a single bit for $n$ rounds and compute an arbitrary function on $2n$ bits) and the latter counts the total number of bits exchanged in a protocol.

**Quantum memoryless Models.** Here we introduce the quantum memoryless communication model. There are a few ways one can define the quantum extension of the classical memoryless model. We find the following exposition the simplest to explain. This quantum communication model is defined exactly as the classical memoryless model except that Alice and Bob are allowed to communicate quantum states. A $T$ round quantum protocol consists of the following: Alice and Bob have local $k$-qubit memories $A, B$ respectively, they share a $m$-qubit message register $M$ and for every round they perform a $q$-outcome POVM $\mathcal{P} = \{P_1, \ldots, P_q\}$ for $q = 2^m$ (which could potentially depend on their respective inputs $x$ and $y$). Let $\{U_x\}_{x \in \{0,1\}^n}, \{V_y\}_{y \in \{0,1\}^n}$ be the set of $(m+k)$-dimensional unitaries acting on $(A, M)$ and $(B, M)$ respectively (this is analogous to the look-up tables $\{f_x, q_y : \{0,1\}^m \rightarrow \{0,1\}^m\}_{x,y \in \{0,1\}^n}$ used by Alice and Bob in the classical memoryless protocol). Let $\psi_0 = (A, M)$ be the all-0 mixed state. Then, the quantum protocol between Alice and Bob can be written as follows: on input $x, y$ to Alice and Bob respectively, on the $i$th round (for $i \geq 1$) Alice sends $\psi_i$ for odd $i$ and Bob replies with $\psi_{i+1}$ defined as follows:

$$
\psi_i = \text{Tr}_A(\mathcal{P} \circ U_x \psi_{i-1} \otimes |0\rangle \langle 0|_B),
$$

where $\mathcal{P} \circ U_x \psi_{i-1}$ is the post-measurement state after performing the POVM $\mathcal{P}$ on the state $U_x \psi_{i-1}$ and $\text{Tr}_A(\cdot)$ refers to taking the partial trace of register $A$. Similarly, define

$$
\psi_i = |0\rangle \langle 0|_A \otimes \text{Tr}_B(\mathcal{P} \circ U_y \psi_i),
$$

where $\text{Tr}_B(\cdot)$ takes the partial trace of register $B$. Intuitively, the states $\psi_i$ (similarly $\psi_{i+1}$) can be thought of as follows: after applying unitaries $U_x$ to the registers $(A, M)$, Alice applies the $q$-outcome POVM $\mathcal{P}$ which results in a classical outcome and post-measurement state on the registers $(A, M)$ and she discards her private memory register and initializes the register $B$ in the all-0 state. The quantum communication protocol terminates at the $i$th round once the $q$-outcome POVM $\mathcal{P}$ results in the classical outcome $\{(1^{m-1}, b)\}_{b \in \{0,1\}}$. After they obtain this classical output, Alice and Bob output $b$. We say a protocol computes $F$ if for every $x, y \in \{0,1\}^n$, with probability at least $2/3$ (probability taken over the randomness in the protocol), after a certain number of rounds the POVM measurement results in $(1^{m-1}, F(x,y))$. The complexity of computing $F$ in the quantum memoryless model, denoted $\text{QNM}(F)$ is the smallest $m$ such that there is a $m$-qubit message protocol that computes $F$. As defined before, we also let $\text{QNM}^+(F)$ (resp. $\text{QNM}^{-}(F)$) be the model in which Alice has a memory (has no memory) and Bob doesn’t have a memory and the communication happens from Alice to Bob.

**Notation.** For the remaining part of the paper we abuse notation by letting $\text{NM}(F)$, $\text{QNM}(F)$ denote the memoryless complexity of computing $F$ and we let $\text{NM model}$ (resp. $\text{QNM model}$) be the memoryless communication model (resp. quantum memoryless communication model).

---

11 After each round of communication, these registers are set to the all-0 register.

12 We remark that a good quantum communication protocol should be such that for every $i \in [T]$, the probability of obtaining $(1^{m-1}, 1 \oplus F(x,y))$ when measuring $\psi_i$ using the POVM $\mathcal{P}$ should be $\leq 1/3$. 

---

10
4 Understanding and characterization of memoryless models

We now state a few observations and relations regarding the memoryless communication models.

Fact 4.1 For every $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$, we have

$$M(F) \leq NM(F) \leq 2M^\rightarrow(F) \leq 2NM^\rightarrow(F).$$

Proof. The proof of the first and last inequality is straightforward since the complexity of a protocol only increases when we force a party to not have memory. We now show the second inequality, suppose $M^\rightarrow(F) = k$. Then there are at most $2^k$ messages Alice sends (suppose at round $i$, she sends a message $m$ and Bob didn’t output $\{0,1\}$, then she knows that $g_y(m) = \perp$, so she need not repeat sending $m$ using her memory) and suppose these $2^k$ messages Alice sends are $\{f_x(m_1), \ldots, f_x(m_{2^k})\}$. The NM protocol for $f$ goes as follows, at the $i$th round Alice sends Bob $(f_x(m_i), i)$ which uses $2k$ bits. If Bob does not output $\{0,1\}$ he simply increments $i$ to $i + 1$ and sends back $i + 1$, which takes at most $k$ bits. Since Alice doesn’t have a memory but has received $i + 1$ and hence has the information of the round $i + 1$, she runs the $M^\rightarrow(F)$ protocol for $(i + 1)$th round and her next message to Bob is $(f_x(m_{i+1}), i + 1)$. Alice and Bob continue the protocol this way until Bob outputs 0 or 1. Since the original $M^\rightarrow(F)$ protocol computes $F$, the NM protocol computes $F$ as well. \hfill \square

As we mentioned earlier, our main contribution in this paper is the memoryless NM model of communication. We saw in Fact 3.2 that $NM^\rightarrow(F)$ is equal to the standard one-way deterministic communication complexity of computing $F$. The $M^\rightarrow(F)$ model was introduced and studied by Papakonstantinou et al. [PSS14]. Additionally observe that the strongest model of communication complexity $M(F)$ is small for every function $F$.

Fact 4.2 For every $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$, we have $M(F) \leq \log n$.

To see this, observe that in the M model (i.e., two-way memory-no memory model), on the $i$th round, Alice sends $i \in [n]$ and Bob (who doesn’t have memory) sends the message $y_i$ to Alice. Alice stores $y_i$ and increments $i$ to $i + 1$ and repeats. After $n$ rounds Alice simply has the entire $y$ and computes $F(x, y)$ on her own (note that $F$ is known to both Alice and Bob).

Below we give few protocols in the NM model to give more intuition of this model.

Algorithms in the memoryless model: In the introduction we described a $\log n + 1$ protocol for the equality function. Below we describe a protocol for the inner product function. For the inner product function IP$_n$, a simple protocol is as follows: For $i = 1, \ldots, n$, on the $i$th round, Alice sends

$$\left(i, x_i, \sum_{j=0}^{i-1} x_i \cdot y_i \pmod{2}\right)$$

which takes $\log n + 2$ bits and Bob replies with

$$\left(i, x_i, \sum_{j=0}^{i-1} x_i \cdot y_i + x_i \cdot y_i \pmod{2}\right) = \left(i, x_i, \sum_{j=0}^{i} x_i \cdot y_i \pmod{2}\right).$$
They repeat this protocol for \( n \) rounds and after the \( n \)th round, they have computed \( \IP_n(x, y) \). Hence \( \NM(\IP_n) \leq \log n + 2 \). Now we describe a protocol for the disjointness function \( \DISJ_n \). Here a \( \log n \) protocol is as follows: Alice sends the first coordinate \( i \in [n] \) for which \( x_i = 1 \) and Bob outputs 0 if \( y_i = 1 \), if not Bob replies with the first \( j \) after \( i \) for which \( y_j = 1 \) and they repeat this procedure until \( i \) or \( j \) equals \( n \). It is not hard to see that \( \DISJ_n(x, y) = 0 \) if and only if there exists \( k \) for which \( x_k = y_k = 1 \) in which case Alice and Bob will find such (smallest) \( k \) in the protocol above, if not the protocol will run for at most \( n \) rounds and they decide that \( \DISJ_n(x, y) = 1 \).

We now mention a non-trivial protocol in the \( \NM \) model for the majority function defined as \( \MAJ_n(x, y) = \lceil \sum_i x_i \cdot y_i \geq n/2 + 1 \rceil \). A trivial protocol for \( \MAJ \) is similar to the \( \IP_n \) protocol, on the \((i + 1)\)th round, Alice sends \((i, x_i, \sum_{i=1}^n x_i \cdot y_i) \) (without the \((\mod 2)\)) and Bob replies with \((i, x_i, \sum_{i=1}^n x_i \cdot y_i) \). Note that this protocol takes \( 2 \log n + 1 \) bits (\( \log n \) for sending the index \( i \in [n] \) and \( \log n \) to store \( \sum_{i=1}^n x_i \cdot y_i \in [n]\)). Apriori this seems the best one can do, but interestingly using intricate ideas from number theory there exists a \( n \log^3 n \) [ST97, KP14] garden-hose protocol for computing \( \MAJ_n \). Plugging this in with Lemma 5.3 we get a protocol of cost \( \log n + 3 \log \log n \) for computing \( \MAJ_n \) in the \( \NM \) model.

An interesting question is, are these protocols for \( \IP_n, \EQ_n, \DISJ_n, \MAJ_n \) optimal? Are there more efficient protocols possibly with constant bits of communication in each round? In order to understand this, in the next section we show that the memoryless communication complexity is lower bounded by the standard non-deterministic communication complexity. Using this connection, we can show the tightness of the first three protocols. Additionally, we show that \( \NM(\MAJ_n) \geq \log n \), thus the exact status of \( \NM(\MAJ_n) \in \{ \log n, \ldots, \log n + 3 \log \log n \} \) remains an intriguing open question.

### 4.1 Lower bounds on memoryless communication complexity

In the introduction, we mentioned that it is an interesting open question to find an explicit function \( F \) for which \( \NM(F) \geq 2 \log n \). Unfortunately we do not even know of an explicit function for which we can prove lower bounds better than \( \log n + \omega(1) \) (we discuss more about this in the open questions). However, it is not hard to show that for a random function \( F \), the memoryless communication complexity of \( F \) is large.

**Lemma 4.3** Let \( F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\} \) be a random function. Then, \( \NM(F) = \Omega(n) \).

**Proof.** The proof is via a simple counting argument. There are \( 2^{2^n} \) distinct functions \( F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\} \). Consider an arbitrary \( s \)-bit \( \NM \) protocol. Let Alice and Bob’s local functions be given by \( \{f_x : \{0,1\}^s \to \{0,1\}^s\}_{x \in \{0,1\}^n} \) and similarly \( \{g_y\}_y \). First observe that there are at most \( 2^n \) distinct messages that Alice can receive from Bob and there are at most \( 2^n \) distinct inputs \( x \), so the total number distinct messages that Alice can send to Bob (recall that Alice’s messages are given by \( f_x(\cdot) \in \{0,1\}^s \)) is at most \( (2^n)^{2^n} = (2^n)^{2^n} \). Similarly Bob can send at most \( (2^n)^{2^s + n} \) distinct messages to Alice. In total there are at most \( (2^n)^{2^s + n} \) distinct protocols that can arise from an \( s \)-bit \( \NM \) protocol. If we have an \( \NM \) protocol that computes an arbitrarily random function \( F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\} \), then we need that \( (2^n)^{2^s + n + 1} \geq 2^{2^n} \), which implies \( s \geq n - \log n - 1 \). \( \square \)

We remark that similar ideas used in this lemma can be used to show that for all \( s < s' \), there exists functions that can be computed using \( s' \) bits of communication in each round but not \( s \) bits of communication. This gives rise to a space hierarchy theorem for the \( \NM \) model.

12
4.1.1 Non-deterministic complexity and memoryless complexity.

In this section we show that non-deterministic communication complexity can be used to provide lower bounds on the memoryless communication model. In the non-deterministic model, Alice is given \( x \), Bob is given \( y \), they need to compute \( F(x,y) \) where \( F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\} \). In a non-deterministic protocol for \( F \), on input \( x \) Alice guesses a proof \( P \) that can convince Alice and Bob that \( F(x,y) \) is either 0 or 1 i.e., she guess a proof \( P \) that is consistent with her input \( x \). If consistent she sends \( P \) to Bob who receives \( y \) and checks the consistency of \( P \) with \( y \). If both player’s inputs are consistent with \( P \) they output 0 or 1 respectively. The cost of the protocol is the size of \( P \). Non-deterministic communication complexity \( \text{Ncc}(F) \) of any function \( F \) is the size of the smallest proofs for the worst case inputs of \( F \). For more on this subject, we refer the interested reader to [Wol03]. We now prove our main lemma.

**Lemma 4.4** \( \text{NM}(F) \geq \log \text{Ncc}(F) - 1 \)

**Proof.** Let \( \text{NM}(F) = s \) and for the rest of the proof we fix the inputs \( x,y \) for Alice and Bob respectively. First observe that the NM-protocol runs for at most \( 2^{s+1} \) rounds. In particular we argue that two messages of any given player in an NM protocol cannot be the same. We prove this fact by contradiction. Suppose for contradiction the protocol on input \( x,y \) runs for \( t > 2^{s+1} \) rounds. Then let us assume that Alice sends the same message \( m \) in both \( i \)th and \( j \)th round for \( i < j \leq t \). Recall that Bob’s message involves computing the function \( g_y(\cdot) \) at each round and on the \( i \) and \( j \)th round his message \( m' \in \{0,1\}^g \) will be identical \( m' = g_y(m) \). Thus Alice receives \( m' \) in both \((i+1)\)th round and \((j+1)\)th round, and she replies with the same message on the \((i+2)\)nd and \((j+2)\)nd round (since she applies the same function \( f_x(\cdot) \) in both rounds to the same message) and so on. It is not hard to see that this protocol will be on an infinite loop without being able to perform the \( t - i - 1 \) rounds of communication to compute \( F \). This contradicts the fact that the protocol ran for \( t \) rounds. Thus any \( s \)-bit memoryless protocol can have at most \( t \leq 2^{s+1} \) rounds.

Now consider a \( 2^{s+1} \)-bit string \( T \in \{0,1\}^{2^{s+1}} \) given by

\[
T = ( a_1, \ldots, a_{2^s}, a_{2^s+1}, \ldots, a_{2^{s+1}} ).
\]

Furthermore, suppose \( T_1 \in \{0,1\}^{2^s} \) is an indicator string for the messages that Alice sends to Bob in the NM protocol (i.e., \( a_i = 1 \) if and only if the \( i \)th message was sent to Bob in the \( 2^s \) rounds in the \( s \)-bit NM protocol, where we implicitly are representing every \( s \)-bit message with its integer value in \( [2^s] \)). Similarly, let \( T_2 \) be an indicator string which indicates which messages were sent from Bob to Alice in the \( 2^s \) rounds. Then we design a non-deterministic protocol in the following way: for every \( (x,y) \), suppose Alice guesses the string \( T \). Recall that in the \( \text{NM}(F) \) protocol, Alice applies the function \( f_x(\cdot) \) on the messages she received. Now, in the non-deterministic communication protocol, Alice uses the same function \( f_x \) and verifies if \( T \) is valid, i.e., she checks if one of the two conditions hold: 1) for every message of Bob \( m_B \) (which she knows because they are present in \( T_2 \)) there exists a message \( m_A \) (indexed in the first half of \( T \)) such that \( f_x(m_B) = m_A \), and 2) \( f_x(m_B) = 1^{s-1}b, b \in \{0,1\} \), for a unique message \( m_B \in \{0,1\}^s \). Suppose Alice observes that \( T \) does not satisfy this verification, then she aborts the protocol, otherwise if \( T \) passes this verification, she sends \( T \) to Bob. Bob performs a similar verification. If Alice and Bob both do not abort, then one of them knows the output bit \( b \) and outputs \( F(x,y) \).

Observe that for every input \( (x,y) \), Alice can always guess a valid transcript \( T \). In order to check the validity of this transcript \( T \), she communicates \( T \) to Bob. Once \( T \) has passed the the
verification test of both Alice and Bob, one player outputs \( F(x, y) \). Hence \( \text{Ncc}(F) \leq |T| = 2^{s+1} \) which gives the lemma statement.

Using this lemma, we immediately get the following corollary.

**Corollary 4.5** Let \( n \geq 2 \). Then \( \text{NM}(\text{EQ}_n), \text{NM}(\text{IP}_n), \text{NM}(\text{DISJ}_n), \text{NM}(\text{MAJ}_n) \geq \log n \).

This corollary follows immediately from Lemma 4.4 because the non-deterministic communication complexity of these functions are at least \( n \), thereby showing that the \((\log n)\)-bit protocols we described in the beginning of this section for the first three of these functions are optimal. However one drawback of Lemma 4.4 is that one cannot hope to prove a lower bound that is better than \( \log n \) since \( \text{Ncc}(F) \leq n \) for every function \( F : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\} \).

### 4.2 Characterization of memoryless communication

Papakonstantinou et al. [PSS14] consider the memory-nomemory model of communication complexity wherein Alice has a memory and Bob doesn’t and they are restricted to one-way communication from Alice to Bob. They show a beautiful combinatorial rectangle-overlay characterization (denoted \( \text{RO}(F) \)) of the \( M^\to \) model. We briefly define \( \text{RO}(F) \) below.

**Definition 4.6 (Rectangle overlay complexity [PSS14])** Let \( n \geq 1, F : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\} \). A length-\( k \) rectangle overlay for \( F \) is a collection \( \{R_1, b_1\}, \ldots, \{R_k, b_k\} \) satisfying the following:

- \( R_i \subseteq \{0, 1\}^n \times \{0, 1\}^n \) is a \( b_i \)-monochromatic rectangle (i.e., \( R_i = X_i \times Y_i \) where \( X_i, Y_i \subseteq \{0, 1\}^n \)) and \( b_i \in \{0, 1\} \).
- \( \{R_1, \ldots, R_k\} \) covers \( \{0, 1\}^n \times \{0, 1\}^n \).
- For every \( (x, y) \in \{0, 1\}^n \times \{0, 1\}^n \), suppose \( R_\ell \) was the first rectangle that contains \((x, y)\), then \( F(x, y) = b_\ell \).

Then, define \( \text{RO}(F) \) is the smallest \( k \) for which there exists a length-\( k \) rectangle overlay for \( F \).

One of the main results of [PSS14] was the following characterization.

**Theorem 4.7 ([PSS14])** For every \( F : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\} \), we have

\[
\log \text{RO}(F) \leq M^\to(F) \leq 2\log \text{RO}(F).
\]

A natural question following their work is, can we even characterize our new general framework of communication complexity wherein both Alice and Bob do not have memory and the communication can be two-way. Generalizing the rectangle-based characterization of [PSS14] to our setting seemed non-trivial because in our communication model the memoryless-ness of the protocol doesn’t seem to provide any meaningful way to split the communication matrix into partitions or overlays (as far as we could analyze). Instead we characterize our communication model in terms of bipartite branching programs, which we define below.\(^{13}\)

\(^{13}\)For a definition of general branching program (BP), refer to Section 2.
Definition 4.8 (Bipartite Branching Program (BBP)) Let $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$. A bipartite branching program is a BP that computes $F$ in the following way: for every $(x, y)$, each node in the branching program is either labelled by a function $f_i \in \mathcal{F} = \{f_i : \{0, 1\}^n \rightarrow \{0, 1\}\}$, or by $g_j \in \mathcal{G} = \{g_j : \{0, 1\}^n \rightarrow \{0, 1\}\}$; the output of the note is given by $f_i(x)$ or $g_j(y)$. We define $\text{BBP}(F)$ as the size of the smallest program that computes $F$ for all $(x, y) \in \{0, 1\}^{2n}$.

Observe that in a BBP every node no longer just queries $x \in \{0, 1\}^n$ at an arbitrary index $i$ (like in the standard BP), but instead is allowed to compute an arbitrary Boolean function on $x$ or $y$. Of course, another natural generalization of BBP is, why should the nodes of the program just compute Boolean-valued functions? We now define the generalized BBP wherein each node can have out-degree $k$ (instead of out-degree 2 in the case of BBP and BP).

Definition 4.9 (Generalized Bipartite Branching Program (GBBP)) Let $k \geq 1$. A generalized bipartite branching program is a BBP that computes $F$ in the following way: for every $(x, y)$, each node in the branching program can have out-degree $k$ and labelled by the node $f_i \in \mathcal{F} = \{f_i : \{0, 1\}^n \rightarrow \{[k]\}\}$, or by $g_j \in \mathcal{G} = \{g_j : \{0, 1\}^n \rightarrow \{[k]\}\}$; the output of each node is given by $f_i(x)$ or $g_j(y)$. We define $\text{GBBP}(F)$ as the size of the smallest program that computes $F$ for all $(x, y) \in \{0, 1\}^{2n}$.

We now show that the generalized bipartite branching programs are not much more powerful than bipartite branching programs, in fact these complexity measures are quadratically related.

Fact 4.10 For $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$, we have $\text{GBBP}(F) \leq \text{BBP}(F) \leq \text{GBBP}(F)^2$.

Proof. The first inequality is obvious as GBBPs are generalized version of BBPs and thus can simulate BBPs. Let $\text{GBBP}(F) = s$. In order to see the second inequality, we show that every node in the GBBP can be computed using a BBP with at most $s$ nodes: observe that if a node in GBBP has $k$ outputs then we can express this node using a binary tree of depth $\log k$ and size $k$ such that each node in this binary tree is indexed by a Boolean function. Hence we can replace every node in the GBBP using the argument above, and we get $\text{BBP}(F) \leq s \cdot k \leq s^2$. \hfill $\square$

It is not clear if the quadratic factor loss in the simulation above is necessary and we leave it as an open question. We are now ready to prove our main theorem relating NM communication model and bipartite branching programs.

Theorem 4.11 For every $F : \{0, 1\}^{n \times n} \rightarrow \{0, 1\}$, we have $\frac{1}{2} \log \text{BBP}(F) \leq \text{NM}(F) \leq \log \text{BBP}(F)$.

Proof. We in fact prove something stronger here, i.e., $\text{NM}(F) = \log \text{GBBP}(F)$ for all $F$. Using Fact 4.10 we get the theorem statement.

We first prove $\log \text{GBBP}(F) \leq \text{NM}(F)$. Let $\text{NM}(F) = s$. Given an $s$-bit NM protocol that computes $F$, one can label all possible messages from Alice to Bob by the set $\mathcal{M}_A = \{m_1^A, m_2^A, \ldots, m_s^A\}$ and similarly all the messages from Bob to Alice by $\mathcal{M}_B = \{m_1^B, m_2^B, \ldots, m_s^B\}$.

Also let us suppose in the NM protocol, Alice and Bob have functions $\{f_x\}_{x \in \{0, 1\}^n}$ and $\{g_y\}_{y \in \{0, 1\}^n}$ respectively. Now we construct a generalized bipartite branching program that contains $2^{s+1}$ nodes, each node

\footnote{Technically the number of messages exchanged between them should be a parameter $T$, but for notational simplicity we assume they communication for $2^s$ rounds, where we used $T \leq 2^s$ by the argument in the first paragraph of the proof of Lemma 4.4.}
labelled by one of the messages \( \mathcal{M}_A \cup \mathcal{M}_B \) which were exchanged between Alice and Bob in the NM protocol. It remains to establish the edge-connections between the respective nodes as well as associate each node with a function \( \{f_j^i\}_{j=1}^{2^k} \times \{g_j^i\}_{j=1}^{2^\ell} \) each mapping \( \{0,1\}^s \) to \( \{0,1\}^s \). For any two pair of nodes \((m^A, m^B) \in \mathcal{M}_A \times \mathcal{M}_B\) we put an edge from \(m^A\) to \(m^B\) if and only if there exists some \( y \in \{0,1\}^s \) such that \( g_y(m^A) = m^B \) in the NM protocol computing \( F \). Similarly for any two pair of nodes \((m^B, m^A) \in \mathcal{M}_B \times \mathcal{M}_A\) we put an edge from \(m^B\) to \(m^A\) if and only if there exists some \( x \in \{0,1\}^n \) such that \( f_x(m^B) = m^A \) in the NM(\( F \)) protocol. We now associate each node with a function \( f^i_j, g^i_j \). On every node \( m^B_j \in \mathcal{M}_B\), we associate the function \( f^i_j : \{0,1\}^n \to \{0,1\}^s \) defined as \( f^i_j(x) \triangleq f_x(m^B_j) \). Similarly on every node \( m^A_i \in \mathcal{M}_A\), we associate a function \( g^i_j : \{0,1\}^n \to \{0,1\}^s \) defined as \( g^i_j(y) \triangleq g_y(m^A_i) \). These functions dictate how the branching program routes between every two nodes. Finally, we make \( m^A_1 \) as the source node and we glue the nodes \( m^A_{1-1_0} \) and \( m^B_{1-1_0} \) together and designate it as a sink node \( b \). This completes the construction of the branching program. Note that all the edges are either from \( \mathcal{M}_A \) nodes to \( \mathcal{M}_B \) nodes or vice versa but not both. It now follows from the construction that for every \((x,y)\) the sequence of message (from \( \mathcal{M}_A \cup \mathcal{M}_B \)) exchanged between Alice and Bob is exactly the sequence of nodes traversed in the branching program when the functions \( f^i_j, g^i_j \) are evaluated on the inputs \( x,y \). By the promise of the NM protocol, we have that the branching program computes \( F(x,y) \), hence \( \text{GBBP}(F) \leq 2^s \).

We now show the other direction, \( \log \text{GBBP}(F) \geq \text{NM}(F) \). Consider an arbitrary generalized bipartite branching program of size \( s \). Note that one can view a \( \text{GBBP} \) computing \( F \) as a bipartite graph. Let us assume that \( s = k + \ell \), where the bi-partition sizes are \( k \) and \( \ell \). We now construct an NM protocol of cost at most \( \log s \). Suppose the nodes are labelled by \([k] \cup [\ell]\). We now let \( a_i \) (resp. \( b_j \)) be the binary number representations of the node label \([k]\) (resp. \([\ell]\)). By the definition of a \( \text{GBBP} \), there are no edges within the nodes labelled by \([k]\) and within nodes labelled by \([\ell]\) and we only have edges going between \([k]\) and \([\ell]\). Moreover, by definition every node labelled by \([k]\) (resp. \([\ell]\)) has a function \( f_i \) (resp. \( g_j \)) associated with it, where \( f_i, g_j : \{0,1\}^n \to \{0,1\}^s \). We now construct our NM protocol as follows: we define two sets of function \( \{f^i_x\}_{x \in \{0,1\}^n} \) and \( \{g^j_y\}_{y \in \{0,1\}^n} \) where \( f^i_x, g^j_y : \{0,1\}^{\log s} \to \{0,1\}^{\log s} \) and these will serve as the functions for the NM protocol. We define Alice’s functions \( \{f^i_x\}_{x} \) in the following way: for every \( x \in \{0,1\}^n, a_i \in \{0,1\}^{\log k}, b_j \in \{0,1\}^{\log \ell} \) if on the node \( i \in [k]\), \( f_i(x) = b_j \) then we define \( f^i_x(b_i) = a_j \). Similarly we define Bob’s functions \( \{g^j_y\}_{y} \) as follows: for every \( y \in \{0,1\}^n, a_i \in \{0,1\}^{\log k}, b_j \in \{0,1\}^{\log \ell} \) if on the node \( j \in [\ell]\), \( g_j(y) = a_i \) then we define \( g^j_y(a_j) = b_i \). From the construction of the \( \text{GBBP} \), it is not too hard to see that the number of different messages used in the NM protocol is at most the number of nodes in the bi-partitions. Hence, each message size at most \( \log s \).

Earlier we saw that \( \text{GBBP} \) is polynomially related to \( \text{BBP} \). We now observe that both these measures can be exponentially smaller than standard branching program size.\(^{15}\)

**Lemma 4.12** The parity function \( \text{PARITY}_n(x,y) = \sum_i x_i \oplus y_i \) (mod 2) gives an exponential separation between generalized bipartite branching programs and branching programs.

**Proof.** In order to see that bipartite branching program can compute \( \text{PARITY}_n \) efficiently, we use the characterization in Theorem 4.11: in the memoryless communication model, Alice can simply compute \( b_1 = \sum_i x_i \) (mod 2) and send it to Bob who computes \( b_2 = \sum_i y_i \) (mod 2) and outputs \( b_1 \oplus b_2 \). However, using Néchiporuk [Nec66] it is possible to show that an arbitrary branching program computing \( \text{PARITY}_n \) is at least \( n \). The idea of the proof is that setting any variable \( x_i \) to a constant \( b \), would give us two different functions on the rest of the bits depending on \( b \): either

\(^{15}\)The function we use here is the standard function that separates bipartite formula size from formula size.
PARITY \( n - 1 \) or \( 1 \oplus \text{PARITY}_n \). Thus fixing each bit gives us two different sub-functions and using Nečiporuk we obtain a lower bound of \( n \) on the branching program size.

**Time Space Trade-off for Memoryless.** Finally, we mention a connection between our communication model and time-space trade-offs. In particular, what are the functions that can be computed if we limit the number of rounds in the memoryless protocol? Earlier we saw that, an arbitrary memoryless protocol of cost \( s \) for computing a function \( F \) could consist of at most \( 2^{s+1} \) rounds of message exchanges. If sending one message takes one unit of time, we can ask whether it is possible to simultaneously reduce the message size \( s \) and the time \( t \) required to compute a function.

The fact below gives a time-space trade-off in terms of deterministic communication complexity.

**Fact 4.13** For every \( k \geq 1 \) and function \( F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\} \), we have \( \text{NM}_k(F) \geq D(F)/k \), where \( \text{NM}_k(F) \) is the NM communication complexity of computing \( F \) with at most \( k \) rounds of communication, and \( D(F) \) is the standard deterministic communication complexity.

**Proof.** Observe that an arbitrary deterministic communication protocol for computing any function \( F \) can be obtained by simulating a memoryless protocol of cost \( \text{NM}_k(F) = s \): at each of the \( k \) rounds, Alice or Bob send an \( s \)-bit message to one another. Hence \( D(f) \leq \text{NM}_k(F) \cdot k \). \( \square \)

It is not hard to now see that the number of rounds in an \( \text{NM}(F) \) protocol corresponds to the depth of the generalized bipartite branching program computing \( F \). So an immediate corollary of the fact above is, even for simple functions such as equality, inner product, if we restrict the depth of GBBP to be \( o(n) \), then we can show exponential-size lower bounds on such GBBPs computing these functions.

## 5 Relations between memoryless communication models

### 5.1 Separations between memoryless models

In this section, we show that there exists exponential separations between the four memoryless communication models defined in Section 3 (and in particular, Fact 4.1).

**Theorem 5.1** There exists functions \( F \) for which the following inequalities (as shown in Fact 4.1) is exponentially weak\(^{16}\)

\[
M(F) \leq \text{NM}(F) \leq 2M^{\rightarrow}(F) \leq 2\text{NM}^{\rightarrow}(F).
\]

**Proof.** The third inequality is exponentially weak for the Disjointness function \( \text{DISJ}_n \) defined as: for every \( x, y \in \{0,1\}^n \), \( \text{DISJ}(x, y) = 0 \) if and only if there exists \( i \in [n] \) such that \( x_i = y_i = 1 \). In this model, we have \( \text{NM}^{\rightarrow}(F) \geq \Omega(n) \). In order to see this, we saw in Fact 3.2 that \( \text{NM}^{\rightarrow}(F) = D^{\rightarrow}(F) \) which is just the standard one-way communication model. It is well known \([\text{KN97}]\) that \( D^{\rightarrow}(\text{DISJ}_n) = \Omega(n) \). However, when Alice has a memory they can perform the following protocol: Alice sends an index with the value \( (i, x_i) \) which takes \( \log n + 1 \) bits and Bob gets a symbol in \( \{0, \perp\} \): 0 if \( x_i = y_i = 1 \) and \( \perp \) if \( x_i \neq y_i \). Bob outputs 0 if he gets 0, otherwise for \( \perp \) Bob continues.

\(^{16}\)We remark that the functions exhibiting these exponential separations are different for the three inequalities.
Note that Alice doesn’t know Bob’s output since it’s a one way $M^{-}(F)$ protocol. Regardless of Bob’s output, Alice repeats the protocol above for $n$ different $(i, x_i)$ (this is where we use the fact that Alice has a memory, hence she doesn’t repeat sending the same $i$ twice). At the $n$th round after Alice sends $(n, x_n)$, she sends an one-bit ‘output’ all-1 string. Note that if Bob didn’t output 0 in the first $n$ rounds, he will output 1 once he gets the all-1 message.

The second inequality is exponentially weak for the inner-product function $IP_n$, defined as $IP(x, y) = \sum_i x_i \oplus y_i \pmod{2}$ for $x, y \in \{0, 1\}^n$. Papakonstantinou et al. [PSS14] showed that $M^{-}(IP_n) \geq n/4$. In the memoryless communication model, we showed $NM(IP_n) \leq \log n + 1$ at the start of this section.

The first inequality is weak for a random function. Let $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ be a random function, then we showed in Lemma 4.3 that $NM(F) \geq \Omega(n)$. Also $M(F) \leq \log n + 1$ follows immediately from Fact 4.2.

We now exhibit exponential separations between the quantum and classical memoryless models of communication complexity.

**Theorem 5.2** There exist functions $F : D \rightarrow \{0, 1\}$ where $D \subseteq \{0, 1\}^n \times \{0, 1\}^n$ for which the following inequalities are exponentially weak: (i) $QNM^{-}(F) \leq NM^{-}(F)$, (ii) $QM^{-}(F) \leq M^{-}(F)$, (iii) $QNM(F) \leq NM(F)$, (iv) $QM(F) \leq M(F)$.

**Proof.** In order to prove that the first inequality is exponentially weak, we use the the standard Boolean Hidden matching problem introduced in [BJK08, GKK+08]: Alice is given $x \in \{0, 1\}^n$, Bob is given a matching $M \in \{0, 1\}^{n/2 \times n}$ on an $n$-vertex bipartite graph and they need to decide if $MX = y$ or $MX = y \oplus 1^n$. The quantum protocol in the memoryless setting is the following: Alice prepares copies of $|\psi\rangle = \frac{1}{\sqrt{n}} \sum_{x \in \{0, 1\}^n} (-1)^{x_i} |i\rangle$ and sends them to Bob. Bob performs the two-outcome measurement $\{ \frac{1}{\sqrt{2}} (|k\rangle \pm |q\rangle) : (k, q) \in M \}$ where $(k, q) \in [n]^2$ refers to the two vertices of an edge in the matching. Observe that probability of obtaining a basis state outcome $\frac{1}{\sqrt{2}} (|k\rangle + |q\rangle)$ is given by

$$\frac{1}{\sqrt{2}} \langle \psi | (|k\rangle + |q\rangle) \rangle = \frac{1}{2n}((-1)^{x_k} + (-1)^{x_q})^2,$$

which equals 0 if $x_k \oplus x_q = 1$. So if Bob obtains $\frac{1}{\sqrt{2}} (|k\rangle + |q\rangle)$, he knows with certainty that $x_k \oplus x_q = 0$ and similarly if he obtains the state $\frac{1}{\sqrt{2}} (|k\rangle - |q\rangle)$ he is sure that $x_k \oplus x_q = 1$. Now, Bob looks at which row in the matching corresponds to the edge $(k, q)$ and suppose it is the $i$th row, then Bob outputs $x_k \oplus x_q \oplus y_i$. Note that this bit equals 0 if and only if $MX = y$ and is 1 otherwise. In order to prove the memoryless lower bound, we immediately get $NM^{-}(F) = D^{-}(F) = \Omega(\sqrt{n})$, where the first equality follows from Fact 3.2 and the second equality was proven in [GKK+08].

In order to prove that the second inequality is exponentially weak, we first define a partial function *gap-hamming distance* $GHD_n : D \rightarrow \{0, 1\}$ where $D = \{(x, y) \in \{0, 1\}^{2n} : d(x, y) \leq n/3$ or $d(x, y) \geq 2n/3\}$ and $GHD(x, y) = 0$ if $d(x, y) \leq n/3$ and 1 if $d(x, y) \geq 2n/3$. We then define the total function $g$ as $g(x, y) = GHD(x, y)$ if $(x, y)$ satisfy the promise of $GHD$ and

---

17 Again, the functions exhibiting these separations are different for the four inequalities.

18 Think of $M$ as the incident matrix where the rows are labelled by the $n/2$ edges and the columns are indexed by the $n$ vertices. Since $M$ is a matching every row has Hamming weight at most 2.
otherwise \( g(x, y) = 0 \). In order to compute \( g \), Alice first sends \( O(1) \) copies of the state \( |\psi_x\rangle = \frac{1}{\sqrt{n}} \sum_i (-1)^{x_i} |i\rangle \) followed by the all-0 string. Bob first performs the swap test between \( |\psi_x\rangle \) and \( |\psi_y\rangle = \frac{1}{\sqrt{n}} \sum_i (-1)^{y_i} |i\rangle \).\(^{19}\) The swap test outputs 1 with probability

\[
\Pr[1] = \frac{1}{2} + \frac{1}{2} \langle \psi_x | \psi_y \rangle^2 = \frac{1}{2} + \frac{1}{2} \left( \frac{1}{n} \sum_i (-1)^{x_i \oplus y_i} \right)^2.
\]

In the case where \( d(x, y) \leq n/3 \), observe that \( \Pr[0] \geq 2/3 \) and in the case where \( d(x, y) \geq 2n/3 \) we have \( \Pr[1] \geq 2/3 \). So Bob runs \( O(1) \) swap tests and suppose he obtains \( \geq 2/3 \)-fraction of 1-outcomes, he outputs 1 and suppose he obtains \( \geq 2/3 \)-fraction of 0-outcomes, he outputs 0. If this were not the case, then Bob simply sees the all-zero string and outputs 0. Observe that with constant error probability (which can be reduced by performing more swap tests) Bob’s output equals \( g(x, y) \) for all \( x, y \). In particular, the overall communication cost to compute \( g \) was \( O(\log n) \), so we have \( \text{QM}^{-}(g) = O(\log n) \). However, Song [Son14] (in particular, Theorem 4.11 in his PhD thesis) shows that every total-function extension \( g \) of the \( \text{GHD}_n \) problem satisfies \( \text{M}^{-}(g) \geq \Omega(n) \).\(^{20}\) Hence \( g \) gives an exponential separation for the second inequality.

The third inequality is weak for the equality function defined as \( \text{EQ}_n(x, y) = 1 \) if and only if \( x = y \). We saw earlier in Corollary 4.5 that \( \text{NM}(\text{EQ}_n) \geq \log n \). We now show that \( \text{QNM}(\text{EQ}_n) \leq O(1) \). Consider the following protocol: Alice first sends a single qubit state \( U_x|0\rangle = \sin \left( \frac{\pi}{2} \cdot \text{Int}(x)/N \right)|0\rangle + \cos \left( \frac{\pi}{2} \cdot \text{Int}(x)/N \right)|1\rangle \) and Bob applies the corresponding phase rotation \( U_y \) to obtain \( |\psi_1\rangle = \sin \left( \frac{\pi}{2} \cdot \text{Int}(x) - \text{Int}(y)/N \right)|0\rangle + \cos \left( \frac{\pi}{2} \cdot \text{Int}(x) - \text{Int}(y)/N \right)|1\rangle \). Suppose \( x = y \), then measuring \( |\psi_1\rangle \) will result in \( |1\rangle \) with certainty, however the problematic case is when \( |\text{Int}(x) - \text{Int}(y)| = 1 \), in which case distinguishing between \( x = y \) and \( |\text{Int}(x) - \text{Int}(y)| = 1 \) seems quantum-hard. For the rest of the proof, we restrict to the case where \( |\text{Int}(x) - \text{Int}(y)| = 1 \) (the same analysis also works if the difference is larger and in fact makes the problem easier to solve). In this hard-instance, Alice additionally sends a qubit \( |\psi_0\rangle = \frac{1}{\sqrt{N}}|0\rangle + \sqrt{\left( 1 - \frac{1}{\sqrt{N}} \right)}|1\rangle \). First, Bob measures \( |\psi_0\rangle \) and if he obtains a 0-outcome, then he measures \( |\psi_1\rangle \). The probability of obtaining a 0-outcome is \( p = 1/N \), so with overwhelming probability he will obtain the 1-outcome, and in this case Bob returns \( |\psi_1\rangle \) to Alice. Alice applies \( U_x \) to \( |\psi_1\rangle \) and sends \( U_x|\psi_1\rangle \) along with \( |\psi_0\rangle \) to Bob. Bob first applies \( U_y \) to \( U_x|\psi_1\rangle \) to obtain

\[
|\psi_2\rangle = U_y U_x|\psi_1\rangle = \sin \left( \frac{\pi}{2} \cdot \frac{2}{N} \cdot (\text{Int}(x) - \text{Int}(y)) \right)|0\rangle + \cos \left( \frac{\pi}{2} \cdot \frac{2}{N} \cdot (\text{Int}(x) - \text{Int}(y)) \right)|1\rangle.
\]

Bob then measures \( |\psi_0\rangle \). If he obtains the 1-outcome, he sends back \( |\psi_2\rangle \) to Alice, and if he obtains a 0-outcome, he measures \( |\psi_2\rangle \). They repeat this process for \( R \) rounds and at the end of \( (R + 1) \)-th round, if Bob never obtained a 0-outcome in any of the \( R \) rounds, then Alice and Bob have prepared the state

\[
|\psi_R\rangle = (U_y U_x)^R|\psi_0\rangle = \sin \left( \frac{\pi}{2} \cdot \frac{R}{N} \cdot (\text{Int}(x) - \text{Int}(y)) \right)|0\rangle + \cos \left( \frac{\pi}{2} \cdot \frac{R}{N} \cdot (\text{Int}(x) - \text{Int}(y)) \right)|1\rangle.
\]

Observe that after an expected number of \( R = \Theta(N) \) rounds of this protocol, Bob would obtain a 0-outcome while measuring \( |\psi_0\rangle \). But when \( R = \Theta(N) \), observe that the probability of measuring \( |0\rangle \)

\(^{19}\)The swap test [BCWW01] is a well-known quantum protocol takes in two quantum states \( |\phi\rangle, |\psi\rangle \) as inputs and outputs a bit \( b \in \{0, 1\} \) such that \( \Pr[b = 1] = 1/2 + |\langle \phi | \psi \rangle|^2/2 \).

\(^{20}\)Here total-function extension of \( \text{GHD}_n \) means that: for every \( g : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\} \) defined as \( g(x, y) = \text{GHD}_n(x, y) \) if \( d(x, y) \leq n/3 \) or \( d(x, y) \geq 2n/3 \) and \( g(x, y) \) is defined arbitrarily to take values in \( \{0, 1\} \) when \( d(x, y) \in \{n/3, \ldots, 2n/3\} \).
when measuring $|\psi_R\rangle$ is a constant $O(1)$. Hence, if Bob measures $|\psi_R\rangle$ after $R$ rounds, he will obtain $|0\rangle$ outcome with constant probability and he can decide that $E_{\text{EQ}}(x,y) = 0$ (note that in the yes-instance when $x = y$, Bob will always obtain the outcome $|1\rangle$ regardless of $R$). The goal of repeating this $R$ times to boost the probability of detecting the instances when $|\text{Int}(x) - \text{Int}(y)| = 1$ versus the case $x = y$.

It remains to take care of the expectation and bound the probability of failure. So the final QNM protocol for equality is the following: Alice initially prepares $C = O(1)$ copies of $|\psi_1\rangle$ and $T = O(1)$ copies of $|\psi_0\rangle$. In the $i$th round for $i \geq 2$, Alice applies $U_x$ on each one of the $C$ copies of $|\psi_{i-1}\rangle$, prepares another $T$ copies of $|\psi_0\rangle$ and sends it to Bob. Bob applies $U_y$ to the $C$ copies of $U_x|\psi_{i-1}\rangle$ and measures the $T$ copies of $|\psi_0\rangle$. If Bob obtains a single 0 in the $T$ measurements, he measures the $C$ copies of $|\psi_i\rangle$. If not, he sends back the $C$ copies of $|\psi_i\rangle$ to Alice and they continue. Earlier we saw that an expected $R = \Theta(N)$ rounds suffice in order for Bob to obtain a 0 outcome. By Markov’s inequality, with probability at least $\Omega(2^{3n})$ (probability taken over the randomness of this protocol), after $3R$ rounds the protocol will terminate with Bob obtaining a 0 outcome. After terminating the protocol, Bob measures the $C$ copies of $|\psi_R\rangle$ which will result in a majority of 0-outcomes with high probability. The overall failure probability of Bob can be made an arbitrary constant by picking appropriate constants $C,T = O(1)$.

For the last inequality we use a promise Equality $p_{\text{EQ}}$ function defined as $p_{\text{EQ}}(x,y) = 1$ for all $(x,y)$ such that $|\text{Int}(x) - \text{Int}(y)| \leq N^{1/3}$ (i.e., they differ in at most the $n^{1/3}$ least significant digits), and $p_{\text{EQ}}(x,y) = 0$ for $(x,y)$ satisfying $|\text{Int}(x) - \text{Int}(y)| \geq N^{2/3}$ (i.e., they differ in at least a constant fraction of the $n^{1/3}$ most significant digits), promised the inputs to $p_{\text{EQ}}$ satisfy this condition. We first show that for $\text{QNM}(\text{EQ}_n) = 1$. Here, Bob sends the same 1-qubit state $\sqrt{1 - \text{Int}(y)/N}|0\rangle + \sqrt{\text{Int}(y)/N}|1\rangle$ for $T = \Theta(2^{3n})$ many rounds (although Bob doesn’t have the memory to store which round he is in, Alice can send him bits in $\{0,1\}$ asking him to send these $T$ qubits). Alice stores each of the qubits, and after receiving $T$ such qubits she measures all these qubits. Suppose she obtained the string $\tilde{y} \in \{0,1\}^T$, it is not hard to see by a Hoeffding bound that, with high probability, $\text{Int}(\tilde{y})$ is close to $\text{Int}(y)$.\footnote{Measuring this qubit $T$ times is equivalent to flipping $T \in \{0,1\}$-valued coins where the probability of 1 is $\text{Int}(y)/N$. Suppose we flip $T$ coins and observe the number of 1s in these flips, let us call this number $M$. Observe that expectation of $M = T \cdot \text{Int}(y)/N$, then the Hoeffding bound implies that $\Pr[M \leq (1 - \delta)T \cdot \text{Int}(y)/N] \leq \exp(-\delta^2 T \cdot \text{Int}(y)/2N)$. By letting $T = \Theta(N/\delta^2)$ and choosing $\delta = 1/N$, we can ensure that $M \cdot N/T$ is arbitrarily close to $\text{Int}(y)$, in fact it is close up to a constant difference.} Hence with success probability arbitrarily close to 1, Alice obtains a $\tilde{y} \in \{0,1\}^n$ such that $|\text{Int}(\tilde{y}) - \text{Int}(y)| \leq O(1)$. Once Alice obtains such a strong approximation of $y$, she compares the $n^{1/3}$ least significant bits of $x,y$ and decides the function value $p_{\text{EQ}}(x,y)$. We now show that $M(p_{\text{EQ}}) \geq \Omega(\log n)$. To see this, we first observe that if $M(p_{\text{EQ}}) = s$, then the number of rounds in this protocol is at most $2^s$: note that Alice has memory and Bob has no memory, so there is no point in Alice sending the same message $m \in \{0,1\}^s$ twice, on both instances Bob would return $g_y(m)$ and Alice would have known $g_y(m)$ the first time she sent $m$ to Bob. In order to show the lower bound, let us fix an arbitrary $y \in \{0,1\}^n$, then after $2^s$ rounds of communication, Alice has obtained at most $s \cdot 2^s$ bits of information about $y$. In order to solve the $p_{\text{EQ}}$ problem, Alice needs at least $\Omega(n^{1/3})$ bits of information before she can compute $p_{\text{EQ}}(x,y)$. Hence we have $s \cdot 2^s \geq \Omega(n^{1/3})$, which gives us $s \geq \Omega(\log n)$. \hfill \Box

One drawback in the exponential separations above is that we allow a quantum protocol to err with constant probability but require the classical protocols to be correct with probability 1. We remark that except the second and third inequalities, the remaining inequalities also show
exponential separations between the randomized memoryless model (wherein Alice and Bob have public randomness and are allowed to err in computing the function) versus the corresponding quantum memoryless model.

5.2 Relating the Garden-hose model and Memoryless complexity

In this section we show an interesting connection between the memoryless communication models and the so-called garden-hose complexity. The garden-hose model of computation was introduced by Buhrman et al. [BFSS13] to understand quantum attacks on position-based cryptographic schemes. In this section, we show that the logarithm of the garden-hose complexity of a Boolean function $F$ is sandwiched between memoryless communication complexity $\text{NM}(F)$ and one-way memory communication complexity (wherein Bob doesn’t have a memory) $M^{-\alpha}(F)$. We now briefly define the garden-hose model.

Garden-hose model. In the garden-hose model of computation, Alice and Bob are neighbours (who cannot communicate) and have few pipes going across the boundary of their gardens. Based on their private inputs $x, y$ and a function $F : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$ known to both, the players connect some of the opening of the pipes on their respective sides with garden-hoses. Additionally, Alice connects a tap to one of the pipes on her side. Naturally, based on the garden-hose connections, water travels back and forth through some of the pipes and finally spills on either Alice’s or Bob’s side, based on which they decide if a function $F$ on input $x, y$ evaluates to 0 or 1. It is easy to show that Alice and Bob can compute every function using this game. The garden-hose complexity $\text{GH}(F)$ is defined to be the minimum number of pipes required to compute $F$ this way for all possible inputs $x, y$ to Alice and Bob. For more on garden hose complexity, we refer the interested reader to [BFSS13, KP14, Spe12, Spe16].

The first observation relating the garden-hose model and memoryless communication complexity is that, the garden-hose model is exactly the $\text{NM}$ communication model, except that in addition to the memoryless-ness of Alice and Bob, there is a bijection between the incoming and the outgoing messages of both players (i.e., the local functions Alice and Bob apply $\{f_x : \{0, 1\}^s \to \{0, 1\}^s\}_x, \{g_y : \{0, 1\}^s \to \{0, 1\}^s\}_y$ are bijective functions. We now state and prove our main lemma which shows how $\text{GH}$ is related to the standard memoryless communication model.

Lemma 5.3 For $F : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$, we have $\text{NM}(F) \leq \log(\text{GH}(F)) \leq M^{-\alpha}(F) + 2$.

Proof. We first prove the first inequality. Let $x, y \in \{0, 1\}^n$ and $\text{GH}(F, x, y) = s$, i.e., there exists $s$ pipes that Alice and Bob place which form the garden-hose protocol to compute $F(x, y)$. Let the pipe openings on Alice’s side be indexed by $A_1, \ldots, A_s$ and Bob’s side be $B_1, \ldots, B_s$. We now define an $\text{NM}$ protocol as follows: let $f_x : \{0, 1\}^{\log s} \to \{0, 1\}^{\log s}$ and $g_y : \{0, 1\}^{\log s} \to \{0, 1\}^{\log s}$ be defined as, for $u, v \in [s]$, we $f_x(u) = v$ if and only if there exists a pipe between $A_u$ and $A_v$ on Alice’s side. Similarly, for Bob’s side we let $g_y(w) = z$ iff $B_w$ is connected to $B_z$ for $w, z \in [s]$. So, in the $\text{NM}$ protocol, Alice and Bob exchange the name of the pipes (which takes $\log s$ bits) through which water flows back and forth in the garden-hose protocol. This continues until one of the players sees a pipe opening in which case Alice or Bob output either 0 or 1 respectively. It is clear that the message size is at most $\log s$ since it takes at most $\log s$ bits to provide the name of one of the $s$ pipes. One can easily observe that as long as the garden-hose protocol computes $F$, the $\text{NM}$ protocol also computes $F$. 

21
We next prove the second inequality. Let \( M \rightarrow (F) = \ell \), hence there are at most \( 2^\ell \) messages that Alice can send to Bob (each message corresponds to an overlay rectangle, see Definition 4.6). We now construct an \( (3 \cdot 2^\ell) \)-pipe garden-hose protocol for \( F \) as follows. For each possible message in the \( M \rightarrow (F) \) protocol, we use blocks of 3 pipes (corresponding to the 3 states \{0, 1, \perp\} of Bob in an \( M \rightarrow \) protocol) designed as follows: for every one of the \( 2^\ell \) blocks \( B \) of 3-pipes, a flow on the first pipe of \( B \) signifies Alice sending the message \( i \) to Bob, the second pipe is labelled by \( \perp \) and a flow in it signifies Bob continues the protocol after receiving message \( i \), and the third pipe is denoted by output \( 0 \). A flow on it would signify Bob wants to output \( 0 \) after receiving message \( i \) (note that Bob doesn’t bother about the 1 outcome because in the garden-hose model, by convention, if its a 1 outcome the spill occurs in Alice’s side).

Now consider a sequence of \( 2^\ell \) overlay rectangles provided by the \( M \rightarrow (F) = \ell \) protocol. On input \( x \), suppose Alice chooses a subset of \( k \) rectangles \( R_1, R_2, \ldots, R_k \). Alice then does the following: she connects the tap to the first pipe of block \( B_1 \) and for all \( i \in [k - 1] \), she connects the second pipe of \( B_i \) to the first pipe of \( B_{i+1} \). For each block Bob does the following: on input \( y \), on any message \( i \) if Bob wants to output \( 0 \) he connects the first pipe of the block \( B_i \) with the third pipe of that block. If he wants to output \( 1 \), he does nothing on the block \( B_i \), and if he wants to continue, he connects the first pipe to the second pipe of that block. This completes the construction of the garden-hose protocol. Since the \( M \rightarrow \) protocol computes \( F \), just by construction observe that once the GH protocol between Alice and Bob starts, the water in the pipes will flow and spill on either one of the sides, depending on the output of the \( M \rightarrow (F, x, y) \). Now, the correctness of the garden-hose protocol is straightforward from the rectangle overlay construction. This proves our desired inequality.

We remark that the second inequality in the lemma above is exponentially weak for the inner product function \( \text{IP}_n \). Papakonstantinou [PSS14] showed that \( M \rightarrow (\text{IP}_n) \geq \Omega(n) \) but we know that \( \text{GH}(\text{IP}_n) = \Theta(n) \) [KP14]. Hence this gives an exponential separation between \( \log(\text{GH}(\text{IP}_n)) \) and \( M \rightarrow (\text{IP}_n) \).

Interestingly, the second inequality in Lemma 5.3 gives us a way to construct a garden-hose protocol using an \( M \rightarrow \) protocol and, as we will see below, this could result in potentially stronger upper bound on the garden-hose model. In an earlier work of Klauck and Podder [KP14], it was conjectured that the disjointness function with input size \( m = n \cdot 2 \log n \) (i.e., with set size \( n \) and universe size \( n^2 \)) has a quadratic lower bound \( \Omega(m^2) \) in the garden-hose model. Here, we show that GH protocol for this problem has cost \( O(m^2/\log^2 m) \). Although the improvement is only by a logarithmic-factor, we believe that this complexity can be reduced further which we leave as an open question.

**Disjointness with quadratic universe:** Alice and Bob are given \( n \) numbers each from \([n^2]\) as a \( m = n \cdot 2 \log n \) long bit strings. Their goal is to check if all of their \( 2n \) numbers are unique. Without loss we can assume that the \( n \) numbers on the respective sides of Alice and Bob are unique, if not they can check it locally and output \( 0 \) without any communication. Then an \( M \rightarrow \) protocol for computing this function is as follows: Alice keeps sending all her numbers to Bob one by one (using her local memory to keep track of which numbers she has already sent). This requires \( 2 \log n \) size message register on every round. Bob upon receiving any number from Alice, checks if any number of his side matches the number received. If there is a match he outputs 0, else he continues. For the last message Alice sends the number along with a special marker. Bob performs his usual check and output 1 if the check passes and the marker is present. Clearly the cost of this protocol is \( 2 \log n \)
and thus from Lemma 5.3 the garden-hose protocol for computing this function has cost $n^2$. Since the input size is $m = n \cdot 2\log n$, the cost of the garden-hose protocol is $O(m^2/\log^2 m)$.

## 6 Open questions and future directions

In this work, our main contribution has been to describe a seemingly-simple model of communication complexity and characterize it’s complexity using branching programs. We believe that our work could open up a new direction of research and results in this direction. Towards this, here we mention a few open questions:

1. Is there a function $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ and a universal constant $c > 1$ for which we have $\text{NM}(F) \geq c \log n$. In particular, there are two consequences of such a result: (a) Using our relation to garden-hose model in Section 5.2, such a function will lead to the first super-linear $n^c$ lower bound for garden-hose complexity, (b) using our characterized to branching programs, this would result in the first super-linear $n^c$ lower bound for bipartite branching programs (analogous to Tal’s first super-linear lower bound on bipartite formula size of inner-product [Tal16]). Also if we could show this for $c \geq 2 + \varepsilon$, this would imply a $\Omega(n^{2+\varepsilon})$ lower bound for general formula size, improving upon the best lower bound by Nečiporuk [Nec66].

2. One possible candidate function which we haven’t been to rule out is the distributed 3-clique function: suppose Alice is given $x \in \{0, 1\}^{\binom{n}{2}}$ and Bob is given $y \in \{0, 1\}^{\binom{n}{2}}$. We view their inputs as jointly labelling of the $\binom{n}{2}$ edges of a graph on $n$ vertices, then does the graph with edges labelled by $x \oplus y$ have a triangle? Also, what is the complexity of the $k$-clique problem?

3. Another possible generic technique to approach this question is through a lifting theorem: in particular, is there a gadget $g : \{0, 1\}^b \times \{0, 1\}^b \rightarrow \{0, 1\}$ such that for every function $F : \{0, 1\}^{nb} \times \{0, 1\}^{nb} \rightarrow \{0, 1\}$ and $F \circ g$ defined as

$$F \circ g(x^1, \ldots, x^n, y^1, \ldots, y^n) = F(g(x^1, y^1), \ldots, g(x^n, y^n)),$$

we have $\text{NM}(F \circ g) \geq \text{NM}_{\text{query}}(F) \cdot b$?\footnote{In this model, an $m$-bit memory query algorithm works in the space of $\{0, 1\}^{\log n + 1 + m}$ bits. Suppose the goal is to compute $f : \{0, 1\}^n \rightarrow \{0, 1\}$ on input $x$ and the state of the algorithm is $(i, b, w) \in \{0, 1\}^{\log n + 1 + m}$, then one step of the query algorithm corresponds to the following: one query to $i$ and obtain $b \oplus x_i$, operations on the workspace memory $(w, b)$ in order to produce a new index $i'$ to query the next turn. The cost of such a protocol is the smallest workspace memory $m$ required to compute a function $F$ (on the hardest input $x$). It is not hard to see that the complexity in this query model corresponds to the size of the smallest branching program computing $F$.}

4. For a random function $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$, do we have $\text{QNM}(F) \geq \Omega(n)$?

5. For every $k \geq 1$, can we find problems that are complete for the class of memoryless communication complexity when Alice and Bob share at most $k$ bits of memory?

6. In this work, we only looked at the “deterministic version” of memoryless communication complexity. One could also look at the model wherein Alice and Bob have private or public randomness and need to compute $F$ with probability at least $2/3$.\footnote{We remark that two of the four separations in Theorem 5.2 that we presented between quantum memoryless communication and classical memoryless communication might not hold in the setting where the classical players are allowed some error, but for the other two separations between the randomized version of memoryless and quantum memoryless model still hold.}
7. Similar to Papakonstantinou et al. [PSS14] can we show a relation between the NM model and the polynomial hierarchy in the communication complexity world? Papakonstantinou et al. showed that poly(log n) space in the M→ model corresponds to P^{NP_{CC}}. We believe that poly(log log n) space in the NM model corresponds to P^{NP_{CC}} in the communication complexity world.

8. Is there a notion of catalytic communication complexity (similar to the notion of catalytic computation introduced by Buhrman et al. [BCK+14]) wherein on top of memoryless-ness, the message register is filled with a pre-loaded data and at the end of the computation Alice and Bob needs to restore the data. It seems that in order to do this, the computation has to be reversible i.e., from Alice’s (Bob’s) perspective given an arbitrary input x (input y) two incoming messages cannot be mapped into a single message. We remark that this model looks similar to the garden-hose model of computation which is also reversible and memoryless [KP14].

Acknowledgements. We would like to thank Mika Göös, Robert Robere, Dave Touchette and Henry Yuen for helpful pointers. SP also thanks Alexander Kehrner for discussions during the early phase of this project as a part of discussing garden-hose model. We thank Henry Yuen for ideas on the QNM versus NM separation. Part of this work was done when SA was visiting University of Ottawa (hosted by Anne Broadbent) and University of Toronto (hosted by Henry Yuen) and thank them for their hospitality. SA was supported in part by the Army Research Laboratory and the Army Research Office under grant number W1975117.

References

[AAKK18] F. M. Ablayev, A. Ambainis, K. Khadiev, and A. Khadieva. Lower bounds and hierarchies for quantum memoryless communication protocols and quantum ordered binary decision diagrams with repeated test. In SOFSEM 2018: Theory and Practice of Computer Science - 44th International Conference on Current Trends in Theory and Practice of Computer Science, pages 197–211, 2018. 6

[AMS99] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. Journal on Computer Systems and Science, 58(1):137–147, 1999. Earlier in STOC’96. 2

[BBM12] E. Blais, J. Brody, and K. Matulef. Property testing lower bounds via communication complexity. Computational Complexity, 21(2):311–358, 2012. 2

[BCG+16] H. Buhrman, L. Czekaj, A. Grudka, M. Horodecki, P. Horodecki, M. Markiewicz, F. Speelman, and S. Strelchuk. Quantum communication complexity advantage implies violation of a bell inequality. Proceedings of the National Academy of Sciences, 113(12):3191–3196, 2016. 6

[BCK+14] H. Buhrman, R. Cleve, M. Koucký, B. Loff, and F. Speelman. Computing with a full memory: catalytic space. In Symposium on Theory of Computing, STOC, pages 857–866, 2014. 24

[BCP+13] J. E. Brody, S. Chen, P. A. Papakonstantinou, H. Song, and X. Sun. Space-bounded communication complexity. In Proceedings of the 4th conference on Innovations in Theoretical Computer Science, pages 159–172, 2013. 2, 3, 5

24
[BCW98] H. Buhrman, R. Cleve, and A. Wigderson. Quantum vs. classical communication and computation. In Proceedings of the thirtieth annual ACM symposium on Theory of computing (STOC), pages 63–68, 1998. 5

[BCWW01] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf. Quantum fingerprinting. Physical Review Letters, 87(16):167902, 2001. 5, 19

[BFS86] L. Babai, P. Frankl, and J. Simon. Complexity classes in communication complexity theory. In 27th Annual Symposium on Foundations of Computer Science (sfcs 1986), pages 337–347. IEEE, 1986. 2

[BFSS13] H. Buhrman, S. Fehr, C. Schaffner, and F. Speelman. The garden-hose model. In Innovations in Theoretical Computer Science, ITCS, pages 145–158, 2013. 3, 21

[BJK08] Z. Bar-Yossef, T. S. Jayram, and I. Kerenidis. Exponential separation of quantum and classical one-way communication complexity. SIAM Journal on Computing, 38(1):366–384, 2008. Earlier in STOC’04. 5, 18

[BW16] M. Braverman and O. Weinstein. A discrepancy lower bound for information complexity. Algorithmica, 76(3):846–864, 2016. 2

[BYJKS04] Z. Bar-Yossef, T. S Jayram, R. Kumar, and D Sivakumar. An information statistics approach to data stream and communication complexity. Journal of Computer and System Sciences, 68(4):702–732, 2004. 2

[CKL17] A. Chailloux, I. Kerenidis, and M. Lauriere. The information cost of quantum memoryless protocols. arXiv preprint arXiv:1703.01061, 2017. 6

[FMP+15] S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, and R. de Wolf. Exponential lower bounds for polytopes in combinatorial optimization. Journal of the ACM, 62(2):17:1–17:23, 2015. 2

[Gav19] D. Gavinsky. Bare quantum simultaneity versus classical interactivity in communication complexity. arXiv preprint arXiv:1911.01381, 2019. 5

[GH92] M. Goldmann and J. Håstad. A simple lower bound for monotone clique using a communication game. Information Processing Letters, 41(4):221–226, 1992. 2, 6

[GKK+08] D. Gavinsky, J. Kempe, I. Kerenidis, R. Raz, and R. de Wolf. Exponential separation for one-way quantum communication complexity, with applications to cryptography. SIAM Journal on Computing, 38(5):1695–1708, 2008. Previous in FOCS’07. 5, 18

[GTN19] A. Gál, A. Tal, and A. T. Núñez. Cubic formula size lower bounds based on compositions with majority. In 10th Innovations in Theoretical Computer Science Conference, ITCS, pages 35:1–35:13, 2019. 6

[IW10] R. Impagliazzo and R. Williams. Communication complexity with synchronized clocks. In 2010 IEEE 25th Annual Conference on Computational Complexity, pages 259–269. IEEE, 2010. 2, 6

[Jef20] S. Jeffery. Span programs and quantum space complexity. In 11th Innovations in Theoretical Computer Science Conference, ITCS, pages 4:1–4:37, 2020. 6
[Kla04] Hartmut Klauck. Quantum and classical communication-space tradeoffs from rectangle bounds. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 384–395. Springer, 2004.  

[KN97] E. Kushilevitz and N. Nisan. *Communication complexity*. Cambridge University Press, 1997.  

[KP14] H. Klauck and S. Podder. New bounds for the garden-hose model. In *34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS*, pages 481–492, 2014.  

[KW90] M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM Journal on Discrete Mathematics*, 3(2):255–265, 1990.  

[LS09] T. Lee and A. Shraibman. Lower bounds in communication complexity. *Foundations and Trends in Theoretical Computer Science*, 3(4):263–398, 2009.  

[LTT89] T. Lam, P. Tiwari, and M. Tompa. Tradeoffs between communication and space. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 217–226, 1989.  

[MNSW98] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication complexity. *Journal on Computer Systems and Science*, 57(1):37–49, 1998. Earlier in STOC’95.  

[Nec66] Edward I Nechiporuk. A boolean function. *Engl. transl. in Sov. Phys. Dokl.*, 10:591–593, 1966.  

[Pal99] M. Pal. Communication complexity and parallel computing. *IEEE Concurrency*, 7(2):81–83, 1999.  

[PS86] R. Paturi and J. Simon. Probabilistic communication complexity. *Journal of Computer and System Sciences*, 33(1):106–123, 1986.  

[PSS14] P. A. Papakonstantinou, D. Scheder, and H. Song. Overlays and limited memory communication. In *IEEE 29th Conference on Computational Complexity, CCC*, pages 298–308, 2014.  

[Son14] H. Song. *Space-Bounded Communication Complexity*. PhD thesis, Tsinghua University, 2014.  

[Spe12] F. Speelman. Position-based quantum cryptography and the garden-hose game, 2012. Masters Thesis.  

[Spe16] F. Speelman. Position-based quantum cryptography and catalytic computation, 2016. PhD Thesis.  

[ST97] R. K. Sinha and J. S. Thathachar. Efficient oblivious branching programs for threshold and mod functions. *Journal of Computer and System Sciences*, 55(3):373–384, 1997.  

[Tal16] A. Tal. The bipartite formula complexity of inner-product is quadratic, 2016. Earlier in STOC’17.  

[Weg87] I. Wegener. *The complexity of Boolean functions*. BG Teubner, 1987.  

26
[Wol03] R. de Wolf. Nondeterministic quantum query and communication complexities. *SIAM Journal on Computing*, 32(3):681–699, 2003. 2, 13

[Yao79] A. Chi-Chih Yao. Some complexity questions related to distributive computing. In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, pages 209–213, 1979. 1, 7

[Yao93] A. C. Yao. Quantum circuit complexity. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 352–361. IEEE, 1993. 2