Communication Memento: Memoryless Communication Complexity

Srinivasan Arunachalam
IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
srinivasan.arunachalam@ibm.com

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

Abstract

We study the communication complexity of computing functions $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ 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. We show that some of these different variants of our memoryless communication model capture the garden-hose model of computation by Buhrman et al. (ITCS'13), space-bounded communication complexity by Brody et al. (ITCS'13) and the overlay communication complexity by Papakonstantinou et al. (CCC'14). Thus the memoryless communication complexity model provides a unified framework to study all these space-bounded communication complexity models.

We establish the following main results: (1) We show that the memoryless communication complexity of F equals the logarithm of the size of the smallest bipartite branching program computing F (up to a factor 2); (2) We show that memoryless communication complexity equals garden-hose model of computation; (3) We exhibit various exponential separations between these memoryless communication models.

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 [33].

2012 ACM Subject Classification Theory of computation → Communication complexity

Keywords and phrases Communication complexity, space complexity, branching programs, garden-hose model, quantum computing

Digital Object Identifier 10.4230/LIPIcs.ITCS.2021.61

Related Version A full version of the paper is available at [2], `https://arxiv.org/abs/2005.04068`.

Funding *Srinivasan Arunachalam*: Supported in part by the Army Research Laboratory and the Army Research Office under grant number W1975117.

Acknowledgements We thank Florian Speelman for the proof of the first inequality of Theorem 21 and letting us include it in our paper. We would like to thank Mika Göös, Robert Robere, Dave Touchette and Henry Yuen for helpful pointers. SP also thanks Anne Broadbent and Alexander Kerzner for discussions during the early phase of this project as a part of discussing gardenhose model. We also thank anonymous reviewers for their helpful comments and positive feedback. 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.

12th Innovations in Theoretical Computer Science Conference (ITCS 2021).
Editor: James R. Lee; Article No. 61; pp. 61:1–61:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Yao [44] introduced the model of *communication complexity* in 1979 and ever since its 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 $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ 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 $\mathsf{D}(F)$) 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 have been studies on non-deterministic communication complexity [42], quantum communication complexity [43] (wherein Alice and Bob are allowed to share *quantum bits* and possibly have shared entanglement), *unbounded error communication complexity* [36] 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 [34], circuit lower bounds [22], algorithms [1], data structures [32], property testing [7], streaming algorithms [6], computational complexity [8], extended formulations [18].[1]

1.1 Background

Space-bounded communication complexity. In the context of our current understanding of computation, the study of space required to solve a problem is a central topic in complexity theory. Several space-bounded models such as width-bounded branching programs [28], limited depth circuits, straight line protocols [29] 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 [23, 26, 28]. In particular, the relation between space-bounded computation and communication complexity was formally initiated by Brody et al. [11] who considered the following question: what happens if we change the standard communication model such that, in each step of communication, Alice and Bob are 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 is possible.[2] Brody et al. mainly studied the one-way communication complexity variant of this limited memory model in which Bob can have two

[1] For more on communication complexity and its applications, we refer the interested reader to the standard textbooks for communication complexity [27, 31].

[2] We remark that the separations obtained by [11] were for non-Boolean functions.

types of memory: an oblivious memory (depends only on Alice's message) and a non-oblivious memory (for computation). With these definitions, they obtained memory hierarchy theorems for such communication models analogous to the space hierarchy theorem in the Turing machine world.

Overlay communication complexity. Subsequently, Papakonstantinou, et al. [35] 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 consist of at most $t(n)$ bits and the complexity of computing F is the maximum $t(n)$ required over all inputs to F. They characterized the complexity in this 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 [3]. Papakonstantinou et al. [35] showed that the message length in their model corresponds to the oblivious memory in a variant of space-bounded model, introduced by Brody et al. [11], where Bob only has access to an oblivious memory.

Garden-hose model. Another seemingly unrelated complexity model, the garden-hose complexity was introduced by Buhrman et al. [15] to understand quantum attacks on position-based cryptographic schemes (see Section 5.1 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. A long-standing open question in this area is, is there an explicit function on n bits whose garden-hose complexity is super-linear in n?

Branching programs. Another unrelated computation model is the branching program. Understanding the size of De Morgan formulas that compute Boolean functions has a long history. In particular, there has been tremendous research in understanding lower bounds on size of De Morgan formulas computing a Boolean function. Similar to formulas, branching programs have also been well-studied in complexity theory. For both branching programs and formulas, we have explicit functions which achieve quadratic (in input size) lower bounds on the size of the branching program/formula computing them. A few years ago, Tal [40] considered *bipartite formulas* for $F : X \times Y \to \{0, 1\}$ (where each internal node computes an arbitrary function on either X or Y, but not both) and showed that the inner product function requires quadratic-sized formulas to compute. In the same spirit as Tal's result, a natural open question is, is there an explicit bipartite function which requires super-linear sized bipartite branching programs to compute?

Given these different models of computation, all exploring the effects on computation under various restrictions, a natural question is, can we view all of them in a unified way:

Is there a model of communication that captures all the above computational models?

In this work we introduce a very simple and new framework called *the memoryless communication complexity* which captures all the computational models mentioned above.

1.2 Memoryless Communication Models

We introduce a *natural* model of communication complexity which we call *the 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) sufficient to compute a function F on all inputs x and y, wherein Alice and Bob do not have any additional memory to remember information between rounds. We denote the *memoryless communication cost of computing* F as $\mathsf{NM}(F)$ (where 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.

Being more formal, we say $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ can be computed in the memoryless communication model with complexity t, if the following is true. For every $x, y \in \{0,1\}^n$ there exists functions $\{f_x, g_y : \{0,1\}^t \to \{0,1\}^t\}$ such that, on input x, y, Alice and Bob use f_x and g_y respectively to run the following protocol: the first message in the protocol is $f_x(0^t)$ from Alice to Bob and thereafter, for every message m_B Bob receives, he replies with deterministic $m' = g_y(m_B)$ and similarly for every message m_A Alice receives she replies with $m'' = f_x(m_A)$. The protocol terminates when the transcript is $(1^{t-1}b)$ at which point they output b as their guess for $F(x, y)$; and we say the protocol computes F if for every x, y, the output b equals $F(x, y)$. $\mathsf{NM}(F)$ is defined as the smallest t that suffices to compute F (using the protocol above) for every $x, y \in \{0,1\}^n$.

It is worth noting that in the memoryless communication protocol, Alice and Bob do not even have access to clocks and hence cannot tell in which round they are in (without looking at the message register). Hence, every memoryless protocol can be viewed as Alice and Bob applying *deterministic* functions (depending on their inputs) which map incoming messages to out-going messages. Also note that unlike the standard communication complexity, where a single bit-message register suffices for computing all functions (since everyone has memory), in the NM model because of the memoryless-ness we need more than a single bit register for computing almost all functions.

For better understanding, let us look at a protocol for the standard *equality* function defined as $\mathsf{EQ}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ where $\mathsf{EQ}_n(x, y) = 1$ if and only if $x = y$. It is well-known that $\mathsf{D}(\mathsf{EQ}_n) = n$. In our model, we show that $\mathsf{NM}(\mathsf{EQ}_n) \leq \log n + 1$: for $i = 1, \ldots, n$, at the ith round, Alice sends the $(\log n + 1)$-bit message (i, x_i) and Bob returns

$(i, [x_i = y_i])$,[3] 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.

Variants of the memoryless model. Apart from the memoryless communication complexity, we will also look at the "memory-nomemory communication complexity" wherein 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 $\mathsf{M}(F)$) 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 analogues 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 $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, let

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

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

1.3 Our Contributions

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 emphasize that we view our main contribution as a new *simple* communication model that provides a conceptual – rather than technical – contribution to studying space complexity, bipartite branching programs and garden-hose complexity under a single model. Given the vast amount of research in the field of communication complexity, we believe that our memoryless model is a very natural model of computation. We now state of various connections between our memoryless communication model and other computational models.

1. Characterization in terms of branching programs. It is well-known that standard models of communication complexity are characterized by the so-called (monochromatic) "rectangles" that partition the communication matrix of the function Alice and Bob are trying to compute. In the study of the memoryless model, Papakonstantinou et al. [35] specifically 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

[3] Here $[\cdot]$ is the indicator of an event in the parenthesis.

$\mathsf{RO}(F)$) of the memory-no memory communication model.[4] One natural idea is to improve the $\mathsf{RO}(F)$ complexity to a more fine-grained rectangle measure that could potentially also characterize $\mathsf{NM}(F)$, but this doesn't seem to be true. The fact that both Alice and Bob do not have memory, doesn't allow them to "narrow" down into a rectangle allowing them to compute the function, instead they narrow down into a set of rectangles. This motivates the question, is there a natural characterization of even the memoryless communication model, in which both Alice and Bob do not have memory? Here we answer this in the positive. We provide a characterization of memoryless communication complexity using branching programs. In particular, we show that for every $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, the memoryless complexity $\mathsf{NM}(F)$ is (up to a factor 2) equal to the logarithm of the size of the smallest bipartite branching program computing F.[5] We give a proof of this statement in Theorem 14.

2. Characterization in terms of garden-hose complexity. The garden-hose model of computation was introduced by Buhrman et al. [15] to understand quantum attacks on position-based cryptographic schemes. It is a playful communication model where two players compute a function with set of pipes, hoses and water going back-and-forth through them. Alice and Bob start with s pipes and based on their private inputs "match" some of the openings of the pipes on their respective sides. Alice also connects a water tap to one of the open pipes. Then based on which side the water spills they decide on the function value. Naturally they want to minimize the number of pipes required over all possible inputs and the garden-hose complexity $\mathsf{GH}(F)$ is defined to be the minimum *number of pipes* required to compute F this way. Given its puzzle-like structure, there have been several works to understand this model and various connections between the garden-hose model and other branches of theoretical computer science were established [15, 25, 39, 14, 16, 38, 17].

On the other hand, space-bounded communication complexity was introduced by Brody et al. [11] to study the effects on communication complexity when they players are limited in their ability to store information from previous rounds. Here Alice and Bob each have at most $s(n)$ bits of memory. Based on their private inputs x, y they want to compute the function in a manner in which at each round Alice receives a single bit message from Bob and based on her input x, the incoming message m_B and her previous $s(n)$-bit register content, she computes a new $s(n)$-bit register and decides whether to stop and output 0/1 or to continue. Bob does the same. space-bounded communication complexity $\mathsf{SM}(F)$ of computing a function F is the minimum register size $s(n)$ required to compute the function on the hardest input.

It was already shown by [11] that for every function, the logarithm of the garden-hose complexity and the space-bounded communication complexity is factor 2 related. It is also easy to show that our newly defined memoryless communication complexity is also factor 2 related to the space-bounded communication complexity by [11]: $\mathsf{NM}(F) \leq 2 \cdot \mathsf{SM}(F) + 1$, and $\mathsf{SM}(F) \leq \mathsf{NM}(F) + \log \mathsf{NM}(F)$. We give a proof of this statement in Lemma 20. Thus it immediately follows that the logarithm of the garden-hose complexity and the memoryless communication complexity of any function is also at most factor 3 related. However we improve this relation using an elegant trick of [30] that allows one to make computations reversible; and thereby show that for every function F, $\mathsf{NM}(F)$ and $\mathsf{GH}(F)$ are equal up to an additive term 4.

$$\log \mathsf{GH}(F) - 4 \leq \mathsf{NM}(F) \leq \log \mathsf{GH}(F).$$

[4] This rectangle-overlay complexity is formally defined in Section 4.2.
[5] We defer the formal definition of such branching programs to Section 2 and Section 4.2.

We give a proof of this in Theorem 21. Hence, the memoryless communication complexity model provides a clean framework for studying all these apparently different looking computational models.

As an immediate application of this new characterization of the garden-hose model, we get a better upper bound for the garden-hose complexity of the indirect storage access function.

▶ **Definition 1** (Indirect Storage Access). *Let $n \geq 4$ and $m \geq 2$ be such that $n = 2m + \log(m/(\log m))$. The Indirect storage access function $\mathsf{ISA}_n : \{0,1\}^n \to \{0,1\}$ is defined on the input string $(\vec{x}, \vec{y}_1, \ldots, \vec{y}_{2^k}, \vec{z})$ where $k = \log\left(\frac{m}{\log m}\right)$, $\vec{x} \in \{0,1\}^m$, $\vec{y} \in \{0,1\}^{\log m}$, $\vec{z} \in \{0,1\}^k$. Then $\mathsf{ISA}_n(\vec{x}, \vec{y}_1, \cdots, \vec{y}_{2^k}, \vec{z})$ is evaluated as follows: compute $a = \mathsf{Int}(z) \in [2^k]$, then compute $b = \mathsf{Int}(\vec{y}_a) \in [m]$ and output x_b.*

For the communication complexity version Alice gets $\vec{x}, \vec{z}$ and Bob gets $\vec{y}_1, \ldots, \vec{y}_{2^k}$, they want to compute $\mathsf{ISA}_n(\vec{x}, \vec{y}_1, \ldots, \vec{y}_{2^k}, \vec{z})$.

It was conjectured in [25] that the Indirect Storage Access function has garden-hose complexity $\Omega(n^2)$. This function is known to have $\Omega(n^2)$ lower bound for the branching program [41] and thus is believed to be hard for garden-hose model in general.[6] But it is easy to see that $\mathsf{NM}(\mathsf{ISA}) \leq \log n$: Alice sends $\vec{z}$ to Bob who then replies with $\vec{y}_{\mathsf{Int}(z)}$. Finally Alice computes the output. Thus using the memoryless-garden-hose equivalence (in Theorem 21) we immediately get $\mathsf{GH}(\mathsf{ISA}) \leq 16n$ (thereby refuting the conjecture of [25]).

3. Separating these models. We then establish the following inequalities relating the various models of communication complexity.[7]

$$\begin{array}{ccccccc} \mathsf{M}(F) & \leq & \mathsf{NM}(F) & = \log \mathsf{GH}(F) \leq & \mathsf{M}^{\rightarrow}(F) & \leq & \mathsf{NM}^{\rightarrow}(F) \\ \rotatebox{-90}{$\geq$} & & \star\rotatebox{-90}{$\geq$} & \| & \rotatebox{-90}{$\geq$} & & \rotatebox{-90}{$\geq$} \\ \mathsf{QM}(F) & & \mathsf{QNM}(F) & 2 \cdot \mathsf{S}(F) & \mathsf{QM}^{\rightarrow}(F) & & \mathsf{QNM}^{\rightarrow}(F) \end{array}$$

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 every inequality is 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[8] is an extensively studied subject [13, 12, 21, 4, 20] and in this paper we show such separations can also be obtained in the memoryless models. We provide the proof in Theorems 17 and 18.

In this paper, we are not been able to give a large separation between QNM and NM, primarily because all lower bound techniques we have for NM seem to apply for QNM as well, e.g., the deterministic one-way communication complexity and the non-deterministic communication complexity[9]. The only "trivial" separation we can give is a factor-2 gap

---

[6] In a typical garden-hose protocol for computing ISA_n, Alice uses m pipes to describe $\vec{z}$ to Bob (each pipe for a different value of $\vec{z}$). Bob can then use another set of m pipes to send $\vec{y}_{\mathsf{Int}(z)}$ to Alice. But since $\vec{y}_i$s need to be unique it appears that Bob needs m set of such m pipes in the worst case. This many-to-one mapping seems unavoidable and hard to tackle in the garden-hose model in general. Hence ISA_n appears to have an $\Omega(n^2)$ garden-hose complexity.

[7] Some of the inequalities are straightforward but we explicitly state it for completeness.

[8] 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.

[9] We note that [15] defined a quantum version of the garden-hose model which differs from the classical model only by the ability of the players to have pre-shared entanglements. They used it to exhibit an exponential classical-quantum separation. Our definition of the quantum version of the memoryless model is a more natural generalization which involves exchanging quantum registers. Thus their exponential separation does not imply a similar separation in our model.

between QNM and NM using the standard idea of super-dense coding (which in fact applies to *all* classical memoryless protocols). Observe that by our our garden-hose characterization earlier, this factor-2 separation translates to a quadratic separation between "quantum-garden-hose" model and the classical garden-hose model.

Since $\mathsf{NM}(F)$ is at most $\mathsf{M}^{\rightarrow}(F)$ for any F, memory-no memory communication complexity can be use to design garden-hose protocols. Using this method we obtain a sub-quadratic garden-hose protocol for computing the function *Disjointness with quadratic universe* which was conjecture to have a quadratic complexity in [25]. We discuss this protocol in Section 5.1.

4. Towards obtaining better formula bounds. Finally, it was shown by Klauck and Podder [25] 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.[10] The best known lower bound for formulae size (over the basis of all fan-in 2 gate) is $\Omega(n^2/\log n)$, due to Nečiporuk from 1966 [33]. Analogous to the Karchmer-Wigderson games [24] and Goldman and Håstad [22] 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. We note that in the memoryless model, constants really matter, e.g., a lower bound of $\log n$ is not same as a lower bound of $2\log n$ as the former would give an n lower bound, whereas the latter will yield an n^2 lower bound for the formula size. This is similar, in flavour, to the circuit depth lower bound where it took several decades of research to get from the trivial $\log n$ lower bound to the sophisticated $3\log n$ lower bound by Håstad [22]. In formula size terminology this translate to going from n to n^3.

Brody et al. [11] conjectured that the problem of reachability in a graph requires $\Omega(\log^2 n)$ non-oblivious memory. However as we have mentioned earlier the space-bounded communication complexity and the memoryless communication complexity of any function are equal up to a constant factor. Thus proving this conjecture would imply the same lower bound on the memoryless communication complexity and in turn imply an $n^{\log n}$ formula-size lower bound for reachability, which would be a break-through in complexity theory. In fact, because of the same general formula to memoryless communication simulation, showing even a $(2+\varepsilon)\log n$ lower bound for reachability would be very interesting.

Finally an additional benefit to our characterization is the following: information theory has been used extensively to understand communication complexity [5, 6, 19, 10, 9] (just to cite a few references). As far as we are aware, usage of information theoretic techniques haven't been explored when understanding the models of computation such as formula size, branching programs and garden-hose model. We believe our characterization using memoryless communication model might be an avenue to use information-theoretic ideas to prove stronger lower bounds in these areas.

---

[10] 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.

Open questions. In this work, our main contribution has been to describe a seemingly-simple model of communication complexity and characterize its 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 the natural open question (referring the reader to the full version for more open questions): Is there a function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ and a universal constant $c > 1$ for which we have $\mathsf{NM}(F) \geq c \log n$. In particular, there are two consequences of such a result: (a) Using our relation to garden-hose model in Section 5.1, 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[11] of inner-product [40]). 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 [33]. 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?

2 Preliminaries

Notation. Let $[n] = \{1, \ldots, n\}$. For $x \in \{0,1\}^n$, let $\mathsf{Int}(x) \in \{0, \ldots, 2^n - 1\}$ 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 $\mathsf{EQ}_n : \{0,1\}^n \to \{0,1\}^n \to \{0,1\}$ is defined as $\mathsf{EQ}_n(x, y) = 1$ if and only if $x = y$. The disjointness function DISJ_n defined as $\mathsf{DISJ}_n(x, y) = 0$ if and only if there exists i such that $x_i = y_i = 1$. The inner product function IP_n is defined as $\mathsf{IP}(x, y) = \sum_i x_i \cdot y_i \pmod 2$ (where $\cdot$ is the standard bit-wise product).

We now define formulae, branching programs and refer the interested reader to Wegener's book [41] for more on the subject.

▶ **Definition 2** (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 $x \in \{0,1\}^n$.*

[11] Note that no super-linear lower bound is known for bipartite formulas that use all gates with fan-in 2 (in particular XOR gates).

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** (Two-way Deterministic memoryless communication complexity). *Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. 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 $f_x, g_y : \{0,1\}^s \to \{0,1\}^s$. 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 $F(x,y) = b$.*[12]

We say a protocol P_F computes F correctly *if for every (x,y), Bob outputs $F(x,y)$. We let $\mathsf{cost}(P_F, x, y)$ be the smallest s for which P_F computes F on input (x,y). Additionally, we let*

$$\mathsf{cost}(P_F) = \max_{x,y} \mathsf{cost}(P_F, x, y)$$

and the memoryless communication complexity *of computing F in this model is defined as*

$$\mathsf{NM}(F) = \min_{P_F} \mathsf{cost}(P_F),$$

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. We denote the complexity in this model by $\mathsf{NM}^{\to}(F)$. 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.[13]

▶ **Fact 4.** *For all function F we have $\mathsf{NM}^{\to}(F) = \mathsf{D}^{\to}(F)$.*

[12] 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.

[13] 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.

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 $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, we denote the complexity in this model as $\mathsf{M}(F)$. Observe that $\mathsf{M}(F)$ 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.

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 $g_y(\cdot) \in \{0,1,\perp\}$ and if he obtains $\{0,1\}$, he outputs 0 or 1, and continues if he obtains $\perp$. We denote the communication complexity in this model by $\mathsf{M}^{\to}(F)$. This model was formally studied by Papakonstantinou et al. [35] as *overlay communication complexity* (we discuss their main contributions in Section 4).

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,[14] they share a m-qubit message register M and for every round they perform a q-outcome POVM $\mathscr{P} = \{P_1, \ldots, P_q\}$ 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 $\{f_x, g_y : \{0,1\}^m \to \{0,1\}^m\}_{x,y\in\{0,1\}^n}$ used by Alice and Bob in the classical memoryless protocol). Let $\psi_0 = (\mathsf{A},\mathsf{M})$ 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 ith round (for $i \geq 1$) Alice sends ψ_i for odd i and Bob replies with ψ_{i+1} defined as follows:

$$\psi_i = \mathrm{Tr}_{\mathsf{A}}(\mathscr{P} \circ U_x \psi_{i-1}) \otimes |0\rangle\langle 0|_{\mathsf{B}},$$

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

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

$\psi_i = |0\rangle\langle 0|_{\mathsf{A}} \otimes \mathrm{Tr}_{\mathsf{B}}(\mathscr{P} \circ U_y \psi_i)$,

where $\mathrm{Tr}_{\mathsf{B}}(\cdot)$ takes the partial trace of register B. Intuitively, the states ψ_i (similarly ψ_{i+1}) can be thought of as follows: after applying unitaries U_x to the registers (A, M), Alice applies the q-outcome POVM $\mathscr{P}$ 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 ith round once the q-outcome POVM $\mathscr{P}$ results in the classical outcome $\{(1^{m-1}, b)\}_{b\in\{0,1\}}$.[15] 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 $\mathsf{QNM}(F)$ is the smallest m such that there is a m-qubit message protocol that computes F. As defined before, we also let $\mathsf{QM}^{\rightarrow}(F)$ (resp. $\mathsf{QNM}^{\rightarrow}(F)$) to 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.

Note that unlike the classical case, with quantum messages there is no apparent way for the players to know with certainty if they have received a designated terminal state (and whether they should stop and output 0/1) without disturbing the message content. Thus a natural choice is to integrate a partial measurement of the message register at each round into the definition.

Notation. For the remaining part of the paper we abuse notation by letting $\mathsf{NM}(F)$, $\mathsf{QNM}(F)$ denote the memoryless *complexity* of computing F and we let NM *model* (resp. QNM *model*) be the memoryless communication model (resp. quantum memoryless communication model). Additionally, we omit *explicitly* writing that Alice and Bob exchange the final message $1^{s-1}f(x,y)$ once either parties have computed $f(x,y)$ (on input x, y respectively).

4 Understanding and characterization of memoryless models

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

▶ **Fact 5.** *For every* $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$, *we have* $\mathsf{M}(F) \le \mathsf{NM}(F) \le 2\mathsf{M}^{\rightarrow}(F) \le 2\mathsf{NM}^{\rightarrow}(F)$.

We refer the reader to the full version of the paper for the proof. As we mentioned earlier, our main contribution in this paper is the memoryless NM model of communication. We saw in Fact 4 that $\mathsf{NM}^{\rightarrow}(F)$ is equal to the standard one-way deterministic communication complexity of computing F. The $\mathsf{M}^{\rightarrow}(F)$ model was introduced and studied by Papakonstantinou et al. [35]. Additionally observe that the strongest model of communication complexity $\mathsf{M}(F)$ is small for every function F.

▶ **Fact 6.** *For every* $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$, *we have* $\mathsf{M}(F) \le \log n$.

[15] 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 ψ_i using the POVM $\mathscr{P}$ should be $\le 1/3$.

To see this, observe that in the M model (i.e., two-way memory-no memory model), on the ith 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 ith round, Alice sends $\Big(i, x_i, \sum_{j=0}^{i-1} x_i \cdot y_i \pmod 2\Big)$ which takes $\log n + 2$ bits and Bob replies with $(i, x_i, \sum_{j=0}^{i-1} x_i \cdot y_i + x_i \cdot y_i \pmod 2\Big) = \Big(i, x_i, \sum_{j=0}^{i} x_i \cdot y_i \pmod 2\Big)$.[16] They repeat this protocol for n rounds and after the nth round, they have computed $\mathsf{IP}_n(x, y)$. Hence $\mathsf{NM}(\mathsf{IP}_n) \leq \log n + 2$. 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 $\mathsf{DISJ}_n(x, y) = 0$ 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 $\mathsf{DISJ}_n(x, y) = 1$. We now mention a non-trivial protocol in the NM model for the majority function defined as $\mathsf{MAJ}_n(x, y) = [\sum_i x_i \cdot y_i \geq n/2 + 1]$. A trivial protocol for MAJ_n 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 $\pmod 2$) and Bob replies with $(i, x_i, \sum_{i=1}^{n+1} x_i \cdot y_i)$. 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 $n \log^3 n$ [37, 25] garden-hose protocol for computing MAJ_n. Plugging this in with Theorem 21 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 $\mathsf{IP}_n, \mathsf{EQ}_n$, 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 deterministic one-way communication complexity. Using this connection, we can show the tightness of the first three protocols. Additionally, we show that $\mathsf{NM}(\mathsf{MAJ}_n) \geq \log n$, thus the exact status of $\mathsf{NM}(\mathsf{MAJ}_n) \in \{\log n, \ldots, \log n + 3\log\log n\}$ 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 $\mathsf{NM}(F) \geq 2\log n$. 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 7.** *Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a random function. Then,* $\mathsf{NM}(F) = \Omega(n)$.

[16] Technically Bob need not send back the bit x_i.

The proof is via a simple counting argument. We refer the reader to the full version of the paper for the definition. 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.

4.1.1 Deterministic one-way communication complexity and memoryless complexity

We now give a very simple lower bound technique for the memoryless communication model in terms of deterministic one-way communication. Although this lower bound is "almost immediate", as we mentioned in the introduction, it already gives us non-trivial lower bounds on the NM complexity of certain functions.

▶ **Fact 8.**
$\mathsf{NM}(F) \geq \log\Big(D^{\rightarrow}(F)/\log D^{\rightarrow}(F)\Big)$, *and* $\mathsf{QNM}(F) \geq \Omega\Big(\log\Big(D^{\rightarrow}(F)/\log D^{\rightarrow}(F)\Big)\Big)$.

Using this lemma, we immediately get the following corollary.

▶ **Corollary 9.** *Let* $n \geq 2$. *Then* $\mathsf{NM}(\mathsf{EQ}_n), \mathsf{NM}(\mathsf{IP}_n), \mathsf{NM}(\mathsf{DISJ}_n), \mathsf{NM}(\mathsf{MAJ}_n), \mathsf{NM}(\mathsf{Index})$, $\mathsf{NM}(\mathsf{BHM})$ *is* $\Omega(\log n)$. *Similarly, we have* QNM *complexity of these functions are* $\Omega(\log n)$.

This corollary follows immediately from Fact 8 because the detereministic-one way communication complexity of these functions are at least n (by a simple adverserial argument), thereby showing that the $(\log n)$-bit protocols we described in the beginning of this section for the first three of these functions is close-to-optimal. However one drawback of Fact 8 is it cannot be used to prove a lower bound that is better than $\log n$ since $\mathsf{D}^{\rightarrow}(F) \leq n$ for every function $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$.

4.2 Characterization of memoryless communication

Papakonstantinou et al. [35] 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 $\mathsf{RO}(F)$) of the $\mathsf{M}^{\rightarrow}$ model. We refer the reader to the full version of the paper for the definition.

One of the main results of [35] was the following characterization.

▶ **Theorem 10** ([35]). *For every* F, *we have* $\log \mathsf{RO}(F) \leq \mathsf{M}^{\rightarrow}(F) \leq 2 \log \mathsf{RO}(F)$.

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 [35] 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.[17]

[17] For a definition of general branching program (BP), refer to Section 2.

▶ **Definition 11** (Bipartite Branching Program (BBP)). *Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. A bipartite branching program is a* BBP *that computes F in the following way: for every (x,y), each node in the branching program is* either *labelled by a function $f_i \in \mathscr{F} = \{f_i : \{0,1\}^n \to \{0,1\}\}_i$* or *by $g_j \in \mathscr{G} = \{g_j : \{0,1\}^n \to \{0,1\}\}_j$; the output edge is labelled by 0 or 1 and the output of the function in the node label decides which edge to follow. The size of a* BBP *is the number of nodes in it. We define* 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 12** (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 $f_i \in \mathscr{F} = \{f_i : \{0,1\}^n \to [k]\}_i$, or by $g_j \in \mathscr{G} = \{g_j : \{0,1\}^n \to [k]\}_j$; the output edges are labelled by $\{1,\ldots,k\}$ and the output of the function in the node label decides which edge to follow. The size of a* GBBP *is the number of nodes in it. We define* 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 13.** *For $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, we have $\mathsf{GBBP}(F) \leq \mathsf{BBP}(F) \leq \mathsf{GBBP}(F)^2$.*

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 14.** *For every F, we have $\frac{1}{2}\log \mathsf{BBP}(F) \leq \mathsf{NM}(F) \leq \log \mathsf{BBP}(F)$.*

We refer the reader to the full version of the paper for the proof. Earlier we saw that GBBP is polynomially related to BBP. We now observe that both these measures can be exponentially smaller than standard branching program size.[18]

▶ **Fact 15.** *The parity function $\mathsf{PARITY}_n(x,y) = \sum_i x_i \oplus y_i \pmod 2$ gives an exponential separation between generalized bipartite branching programs and branching programs.*

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.

[18] The function we use here is the standard function that separates bipartite formula size from formula size.

▶ **Fact 16.** *For every $k \geq 1$ and function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, we have* $\mathsf{NM}_k(F) \geq \mathsf{D}(F)/k$, *where* $\mathsf{NM}_k(F)$ *is the* NM *communication complexity of computing F with at most k rounds of communication, and* $\mathsf{D}(F)$ *is the standard deterministic communication complexity.*

It is not hard to now see that the number of rounds in an $\mathsf{NM}(F)$ 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. Similarly note that one can separate QNM and NM model of communication if we bound the number of rounds: consider the problem where Alice and Bob get $x, y \in \{0,1\}^n$ respectively promised that, $x = y$ or Hamming distance between x, y is $n/2$. In this case, clearly $\mathsf{NM}_k(F) \geq n/k$ (from the fact above), which in particular means that constant-round NM protocols need to send $\Omega(n)$ bits. In contrast, in the QNM model, Alice could simply send $O(1)$ copies of a fingerprint state $|\psi_x\rangle = \frac{1}{\sqrt{n}} \sum_i (-1)^{x_i} |i\rangle$ (in a *single round*) and due to the promise, Bob can perform swap test between $|\psi_x\rangle, |\psi_y\rangle$ and decide if $x = y$ or the Hamming distance is $n/2$ with probability 1.

5 Relations between memoryless communication 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 5).

▶ **Theorem 17.** *There exists functions F for which the following inequalities (as shown in Fact 5) is exponentially weak*[19] $\mathsf{M}(F) \leq \mathsf{NM}(F) \leq 2\mathsf{M}^{\rightarrow}(F) \leq 2\mathsf{NM}^{\rightarrow}(F)$.

We refer the reader to the full version of the paper for the definition. We now exhibit exponential separations between the quantum and classical memoryless models of communication complexity.

▶ **Theorem 18.** *There exist functions $F : D \to \{0,1\}$ where $D \subseteq \{0,1\}^n \times \{0,1\}^n$ for which the following inequalities are exponentially weak: (i)* $\mathsf{QNM}^{\rightarrow}(F) \leq \mathsf{NM}^{\rightarrow}(F)$, *(ii)* $\mathsf{QM}^{\rightarrow}(F) \leq \mathsf{M}^{\rightarrow}(F)$, *(iii)* $\mathsf{QM}(F) \leq \mathsf{M}(F)$.[20]

We refer the reader to the full version of the paper for the definition. 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 inequality, 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. A natural question is to extend these separations even when the classical model is allowed to err with probability at least $1/3$.

5.1 Relating the Garden-hose model, Space-bounded communication complexity and Memoryless complexity

In this section, we show that the memoryless communication complexity $\mathsf{NM}(F)$ of a Boolean function F is equal to the logarithm of the garden-hose complexity up to an additive constant and is equal to the space-bounded communication complexity up to factor 2. But first we briefly define the garden-hose model and the space-bounded communication complexity model.

[19] We remark that the functions exhibiting these exponential separations are different for the three inequalities.
[20] Again, the functions exhibiting these separations are different for the three inequalities.

Garden-hose model [15]. 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 $\mathsf{GH}(F)$ 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 [15, 25, 38, 39].

Space-bounded communication complexity [11]. Alice and Bob each have at most $s(n)$ bits of memory. Based on their private inputs x, y they want to compute the function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ in the following manner: At each round Alice receives a single bit message $m_B \in \{0,1\}$ from Bob and based on her input x, the incoming message m_B and her previous $s(n)$-bit register content, she computes a new $s(n)$-bit register and decides whether to stop and output 0/1 or to continue. Bob does the same. At the beginning of the game, the register contents of both players are set to the all-zero strings. The game then starts by Alice sending the first message and continues until one of players outputs 0/1. space-bounded communication complexity $\mathsf{SM}(F)$ of computing a function F is the minimum register size $s(n)$ required to compute F on the worst possible input (x, y). Brody et al. [11] claimed that space-bounded communication complexity is equal to the garden-hose communication complexity upto factor 2.

▷ Claim 19 ([11]). For every function F there exists constants $c \in (0,1), d \in \mathbb{N}^+$ such that $c \cdot 2^{\mathsf{SM}(F)} \leq \mathsf{GH}(F) \leq 2^{2\mathsf{SM}(F)+2} + d$.

We show the following relation between

▶ **Lemma 20.** *For every function F,* $\mathsf{NM}(F) \leq 2\mathsf{SM}(F) + 1$, $\mathsf{SM}(F) \leq \mathsf{NM}(F) + \log \mathsf{NM}(F)$

We refer the reader to the full version of the paper for a proof. Using the Claim 19 and Lemma 20 we can conclude that the logarithm of the garden-hose complexity is equal to the memoryless NM complexity up to factor 2. This seems interesting already given we can connect these two models, but in the NM model, even factor-2s are important since they are related to formula lower bounds. Now we show that it is possible to further tighten the relation in the lemma above. Below we show that NM is actually equivalent to the logarithm of the garden-hose complexity up to an *additive term* of 4. The first observation relating the garden-hose model and memoryless communication complexity is that, the garden-hose model is exactly the 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 the theorem which shows how GH is related to the standard memoryless communication model.

▶ **Theorem 21.** *For F, we have* $\log \mathsf{GH}(F) - 4 \leq \mathsf{NM}(F) \leq \log \mathsf{GH}(F)$.

We refer the reader to the full version of the paper for a proof. Interestingly, Theorem 21 together with Theorem 17 gives us a way to construct a garden-hose protocol using an $\mathsf{M}^{\rightarrow}$ 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 [25], 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 $\mathsf{M}^{\rightarrow}$ 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 Theorem 21 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 $O(m^2/\log^2 m)$.

References

1. 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.
2. S. Arunachalam and S. Podder. Communication memento: Memoryless communication complexity. *arXiv*, 2000. `arXiv:2005.04068`.
3. 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.
4. 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. Ziv Bar-Yossef, Thathachar S Jayram, Ravi Kumar, and D Sivakumar. Information theory methods in communication complexity. In *Proceedings 17th IEEE Annual Conference on Computational Complexity*, pages 93–102. IEEE, 2002.
6. Ziv Bar-Yossef, Thathachar S Jayram, Ravi 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.
7. E. Blais, J. Brody, and K. Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012.
8. M. Braverman and O. Weinstein. A discrepancy lower bound for information complexity. *Algorithmica*, 76(3):846–864, 2016.
9. Mark Braverman. Interactive information complexity. *SIAM Journal on Computing*, 44(6):1698–1739, 2015.
10. Mark Braverman, Ankit Garg, Denis Pankratov, and Omri Weinstein. From information to exact communication. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 151–160, 2013.
11. 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.

12 H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16):167902, 2001.

13 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.

14 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.

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

16 Well Y Chiu, Mario Szegedy, Chengu Wang, and Yixin Xu. The garden hose complexity for the equality function. In *International Conference on Algorithmic Applications in Management*, pages 112–123. Springer, 2014.

17 Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum homomorphic encryption for polynomial-sized circuits. In *Annual International Cryptology Conference*, pages 3–32. Springer, 2016.

18 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.

19 Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication for boolean functions. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 557–566, 2015.

20 D. Gavinsky. Bare quantum simultaneity versus classical interactivity in communication complexity. *arXiv preprint*, 2019. `arXiv:1911.01381`.

21 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.

22 Mikael Goldmann and Johan Håstad. A simple lower bound for monotone clique using a communication game. *Information Processing Letters*, 41(4):221–226, 1992.

23 R. Impagliazzo and R. Williams. Communication complexity with synchronized clocks. In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 259–269. IEEE, 2010.

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

25 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.

26 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.

27 E. Kushilevitz and N. Nisan. *Communication complexity.* Cambridge University Press, 1997.

28 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.

29 Tak Lam, Prasoon Tiwari, and Martin Tompa. Trade-offs between communication and space. *Journal of Computer and System Sciences*, 45(3):296–315, 1992.

30 Klaus-Jörn Lange, Pierre McKenzie, and Alain Tapp. Reversible space equals deterministic space. *Journal of Computer and System Sciences*, 60(2):354–367, 2000.

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

32 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.

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

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

35 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.

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

37 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.

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

39 F. Speelman. Position-based quantum cryptography and catalytic computation, 2016. PhD Thesis.

40 A. Tal. The bipartite formula complexity of inner-product is quadratic, 2016. Earlier in STOC'17.

41 I. Wegener. *The complexity of Boolean functions*. BG Teubner, 1987.

42 R. de Wolf. Nondeterministic quantum query and communication complexities. *SIAM Journal on Computing*, 32(3):681–699, 2003.

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

44 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.