Electronic Colloquium on Computational Complexity, Report No. 188 (2024)

Catalytic Communication

Edward Pyne[1], Nathan S. Sheffield[1], and William Wang[1]

epyne@mit.edu, *shefna@mit.edu*, *wmwang@mit.edu*
[1]Massachusetts Institute of Technology

Abstract

The study of space-bounded computation has drawn extensively from ideas and results in the field of communication complexity. Catalytic Computation (Buhrman, Cleve, Koucký, Loff and Speelman, STOC 2013) studies the power of bounded space augmented with a pre-filled hard drive that can be used non-destructively during the computation. Presently, many structural questions in this model remain open. Towards a better understanding of catalytic space, we define a model of *catalytic communication complexity* and prove new upper and lower bounds.

In our model, Alice and Bob share a blackboard with a tiny number of free bits, and a larger section with an arbitrary initial configuration. They must jointly compute a function of their inputs, communicating only via the blackboard, and must always reset the blackboard to its initial configuration. We prove several upper and lower bounds:

i) We characterize the simplest nontrivial model, that of one bit of free space and three rounds, in terms of $\mathbb{F}_2$ rank. In particular, we give natural problems that are solvable with a minimal-sized blackboard that require near-maximal (randomized) communication complexity, and vice versa.

ii) We show that allowing constantly many free bits, as opposed to one, allows an exponential improvement on the size of the blackboard for natural problems. To do so, we connect the problem to existence questions in extremal graph theory.

iii) We give tight connections between our model and standard notions of non-uniform catalytic computation. Using this connection, we show that with an arbitrary constant number of rounds and bits of free space, one can compute all functions in TC^0.

We view this model as a step toward understanding the value of filled space in computation.

1 Introduction

Communication complexity has proven an essential tool in analyzing the power of *space* in computation. For the well-studied problem of derandomizing space-bounded computation, i.e. proving $\mathsf{BPL} = \mathsf{L}$, the frontier pseudorandom generators of [Nis92; INW94] are analyzed by considering the space-bounded algorithm as a communication protocol. There has been extensive work analyzing restricted classes of space-bounded algorithms, again relying on this connection [Bra+14; PV21; Coh+21]. Other works have tightly characterized the space required to solve fundamental problems such as estimating the bias of a coin [BV10; BGW20; BGZ21], using sophisticated measures of information complexity.

Concurrently, a new model of bounded-space computation known as *catalytic* computation was introduced by Buhrman, Cleve, Koucký, Loff and Speelman [Buh+14], and used to solve fundamental computational problems more efficiently. In the Catalytic Logspace (CL) model, an algorithm receives an n-bit input, $O(\log n)$ bits of standard working space, and an auxiliary $\mathsf{poly}(n)$ bit catalytic tape τ. This tape has an arbitrary initial configuration, and must be reset to that starting configuration at the end of the computation. Despite a possible intuition that such a tape would not be useful, they showed that CL is likely to be *strictly* stronger than L — in particular, it contains logspace-uniform TC^1, and thus nondeterministic logspace (NL). Recently, Cook and Mertz used catalytic algorithms to show that the tree-evaluation problem, a candidate problem for separating L and P, can in fact be solved by an algorithm running in space $O(\log n \log\log n)$,

ISSN 1433-8092

contradicting long-standing prior beliefs [CM20; CM21; CM23]. Due to the striking power of this model, there have been many followup papers studying its structure [Dul15; Buh+18; Gup+19; Pyn24]. In the other direction, [Coo+24] used tools from the space-bounded literature (in particular, communication bottlenecks) to unconditionally *derandomize* CL. However, many basic questions remain open — it is consistent with current knowledge that $\mathsf{P} \subseteq \mathsf{CL}$, or that $\mathsf{CL} \subseteq \mathsf{NC}^2$, and there is no conditional evidence in either direction.

We aim to understand the power of access to a full memory by developing and studying the phenomenon in the setting of *communication complexity* — to what extent can such un-erasable extra memory be useful in exchanging information between two (computationally unbounded) parties?

1.1 Our Contribution: Catalytic Communication Complexity

We develop a natural model of catalytic communication complexity and prove several results, including maximal separations from the standard model of (randomized) communication complexity, a characterization of efficient protocols for the equality function in terms of extremal graph theory, and a strong equivalence between certain settings of the model and (nonuniform) catalytic computation itself.

Our model is defined as follows. There are two parties, Alice and Bob (perhaps graduate students coming to work on alternating days), sharing a single blackboard. Each party is computationally unbounded, but has no persistent memory between days, except what is written on the blackboard. Unfortunately, all but a tiny corner of the blackboard is currently full of someone else's important research, with instructions not to erase. Are Alice and Bob limited to sharing information through the tiny blank space, or can they do better by temporarily modifying the "Do Not Erase" section in such a way that still allows them to restore it when they're done? If they want to compute some function together, how big does the blackboard need to be, and how many days will they need? Such a model naturally captures the flow of information in a catalytic algorithm. The formal definition is as follows:

Definition 1 (Catalytic Communication Protocol)**.** Fix a function $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$.[1] A ***catalytic protocol computing f with r rounds, s bits of clean space, and c bits of catalytic space*** consists of r many transition functions and one output function,

$$A_1 : \{0,1\}^{n_a} \times \{0,1\}^c \times \{0,1\}^s \to \{0,1\}^c \times \{0,1\}^s$$

$$B_2 : \{0,1\}^{n_b} \times \{0,1\}^c \times \{0,1\}^s \to \{0,1\}^c \times \{0,1\}^s$$

$$\cdots$$

$$A_r : \{0,1\}^{n_a} \times \{0,1\}^c \times \{0,1\}^s \to \{0,1\}^c \times \{0,1\}^s$$

$$B_{\text{out}} : \{0,1\}^{n_b} \times \{0,1\}^c \times \{0,1\}^s \to \{0,1\}.$$

(If r is even, we will instead have $B_r : \{0,1\}^{n_b} \times \{0,1\}^c \times \{0,1\}^s \to \{0,1\}^c \times \{0,1\}^s$ and $A_{\text{out}} : \{0,1\}^{n_a} \times \{0,1\}^c \times \{0,1\}^s \to \{0,1\}$.) For the protocol to be valid, for any $\tau \in \{0,1\}^c$, $x \in \{0,1\}^{n_a}$, $y \in \{0,1\}^{n_b}$, letting $A_i^{(x)} = A_i(x, \cdot\,)$ and $B_i^{(y)} = B_i(y, \cdot\,)$, these functions must satisfy

$$A_r^{(x)} \circ \cdots \circ B_2^{(y)} \circ A_1^{(x)}(\tau, 0^s) = (\tau, w)$$

for some w with

$$B_{\text{out}}^{(y)}(\tau, w) = f(x, y).$$

In words, Alice and Bob's blackboard consists of c bits of arbitrarily-initialized catalytic space and s bits of initially-empty clean space. In a given round, the active party modifies the blackboard according to some arbitrary function of its current contents, the input visible to the active party, and the current timestep, then passes the blackboard to the other party. At the end of the protocol, the catalytic part of the blackboard must have been reset to its starting configuration, and the party with the blackboard must be

[1] We will typically be concerned with the case $n_a = n_b = n$, but will also consider functions with asymmetric input lengths

able to announce the answer. Readers familiar with catalytic computing may observe that such a protocol could be described as an ***amortized bipartite branching program***; this interpretation is the motivation for the particulars of our definition, and will be discussed in more detail in Section 6.

This model induces a corresponding measure of communication complexity:

Definition 2 (Catalytic Communication Complexity)**.** The ***r-round s-clean catalytic communication complexity*** of a function $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$, denoted $\mathsf{CC}_{r,s}(f)$, is the minimum value of c such that f has an r-round catalytic protocol with s bits of clean space and c bits of catalytic space.

The following three propositions represent simple observations about catalytic communication, demonstrating that this complexity measure is both well-defined and interesting. We give only the statements here; the associated proofs can be found in Appendix A. First note that with fewer than 3 rounds, the catalytic space is provably useless — in particular, this means that $\mathsf{CC}_{1,s}(f)$ and $\mathsf{CC}_{2,s}(f)$ are not well-defined for every f, since there may be no amount of catalytic space that suffices:

Proposition 1. Any function computable by a catalytic protocol with 1 or 2 rounds is computable by such a protocol with no catalytic space (i.e. there exists a memoryless protocol with the same amount of clean space, so without loss of generality only the clean space is used).

For $r \geq 3$, $s \geq 1$, however, with enough catalytic space it is possible to compute any function, so $\mathsf{CC}_{r,s}(f)$ is always well-defined:

Proposition 2. Every function $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ has a 3-round catalytic protocol with 1 bit of clean space and $2^{\min(n_a,n_b)}$ bits of catalytic space.

Proposition 2 represents an upper bound on catalytic communication complexity in general, but it is far from tight for some functions. Recall the inner product function (over $\mathbb{F}_2$) $\mathsf{IP}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, defined as $\mathsf{IP}_n(x,y) = \langle x, y\rangle$. We show that IP can be computed with one bit of clean space and n bits of catalytic space:

Proposition 3. We have $\mathsf{CC}_{3,1}(\mathsf{IP}_n) \leq n$, i.e. the inner product function has a 3-round catalytic protocol with 1 bit of clean space and n bits of catalytic space.

Note that this protocol is roughly as efficient as a catalytic protocol can possibly be: as long as f is left-injective (i.e. no two values of x result in the same function $f(x,\cdot)$), it is clear for information theoretic reasons that the catalytic space cannot be made much less than n bits, and certainly the clean space required for any non-constant f must be at least 1 bit — so inner product is essentially the *easiest possible* function for this model. This is in contrast to standard notions of communication complexity: inner product has, up to constant factors, maximal deterministic and randomized communication complexity [RY20]. Such a separation even for a single bit of clean space indicates that the "picture" of catalytic communication can diverge widely from that of standard models.

Furthermore, unlike deterministic and randomized communication complexity, there does not appear to be a direct counting argument showing that a random function requires any amount of catalytic space larger than the trivial bound of n. Despite this barrier, we give a tight characterization of protocols with one bit of clean space and three rounds. As our first main result, we show that $\mathsf{CC}_{3,1}$ is characterized up to a constant factor by $\mathbb{F}_2$ rank:

Theorem 1. For any $f \colon \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$,

$$\frac{\mathrm{rk}(f)}{6} - o(1) \leq \mathsf{CC}_{3,1}(f) \leq \mathrm{rk}(f),$$

where $\mathrm{rk}(f)$ denotes the rank of the communication matrix over $\mathbb{F}_2$.

The proof of the upper bound of Theorem 1 is an explicit protocol, while the lower bound involves an application of Harper's vertex-isoperimetric theorem on an appropriate subspace of the Boolean hypercube.

We remark that several standard communication complexity measures are suspected to be closely controlled by appropriate notions of the rank of the communication matrix, but this case is unusual in that it is the rank over $\mathbb{F}_2$, as opposed to $\mathbb{R}$, that is relevant.

This establishes catalytic communication complexity as a very different measure from standard ones. In particular, if one compares catalytic communication complexity against standard randomized communication complexity[2], there are maximal separations for natural functions in *both* directions. Inner product has a maximally efficient 3-round 1-clean catalytic communication protocol but no standard randomized communication protocol with communication less than the trivial $\Omega(n)$. On the other hand, the equality function EQ_n has $\mathsf{CC}_{3,1}(\mathsf{EQ}_n) = \Omega(2^n)$ but randomized communication complexity $O(1)$ [RY20].

The lower bound approach we use for $\mathsf{CC}_{3,1}$ does not extend to $\mathsf{CC}_{3,2}$, i.e. protocols with a single extra bit of clean space. This is not simply a consequence of technical limitations — we show that for equality, allowing a single additional bit of clean space enables an exponentially more efficient protocol:

Theorem 2. We have that $\mathsf{CC}_{3,2}(\mathsf{EQ}_n) \leq O(n \log n)$.

Our protocol for equality follows from a connection to a new variant we propose of the well-known Ruzsa–Szemerédi problem in extremal graph theory. We demonstrate that efficient 3-round catalytic protocols for equality can be obtained generically from constructions of "full Ruzsa–Szemerédi graphs", a special case of Ruzsa–Szemerédi graphs. We show that the current frontier Ruzsa–Szemerédi construction can be modified to satisfy our stronger condition, and hence obtain an efficient protocol. We also show a reverse implication — that is, that efficient protocols imply dense Ruzsa–Szemerédi constructions — enabling us to obtain nontrivial lower bounds for the complexity of equality:

Theorem 3. For any constant s, we have $\mathsf{CC}_{3,s}(\mathsf{EQ}_n) \geq n + \Omega(\log^*(n))$.

In addition to this unconditional lower bound, the connection to Ruzsa–Szemerédi graphs also gives a barrier result: improving our upper bound to $\mathsf{CC}_{3,O(1)}(\mathsf{EQ}_n) = O(n)$ would yield (among other things) a polynomial query *lower* bound for testing monotonicity of Boolean functions over general posets.

Towards obtaining lower bounds stronger than $n + \Omega(\log^* n)$, we propose considering protocols for the ***indexing problem*** $\mathsf{IND}_n : \left([2^n] \times \{0,1\}^{2^n}\right) \to \{0,1\}$, $\mathsf{IND}_n(x,y) = y[x]$. Here, Alice's input is a length-n bitstring representing an index, and Bob's input is a length-2^n bitstring, with the output of the function being the bit of Bob's input corresponding to Alice's index. Another description of this problem is that Alice is given some n-bit input, and Bob is given the truth table of some arbitrary boolean function on n-bit inputs, and they must compute the evaluation of Bob's function on Alice's input. It is clear from this phrasing that proving a complexity lower bound on any function in terms of Alice's input length would require proving at least as strong a lower bound on IND_n, as any other function is a special case. We find a graph theoretic characterization of protocols for this problem, which allows us to show a non-trivial lower bound via the Kővári-Sós-Turán theorem.

Theorem 4. We have $\mathsf{CC}_{3,s}(\mathsf{IND}_n) \geq (1+\varepsilon)n$, for some constant ε depending on s.

While we suspect Theorem 4 is far from tight, it remains open to show super-linear lower bounds even for 2 clean bits.

Finally, we study protocols with more rounds. We note that our measure of catalytic communication complexity corresponds directly to the minimum amount of amortization required in an amortized bipartite branching program of bounded length. Since an ordinary branching program is in particular a bipartite branching program, this immediately lets us translate known results from nonuniform catalytic computing to show statements such as $\mathsf{CC}_{\mathsf{poly}(n),O(1)}(f) \leq O(n)$ for all f. By a more specialized analysis of Buhrman, Cleve, Koucký, Loff and Speelman's algebraic proof of $\mathsf{TC}^1 \subseteq \mathsf{CL}$, we are also able to show the following:

[2] Comparing against *deterministic* standard communication complexity, we would instead find that catalytic communication is strictly stronger, in the sense that $\log(\mathsf{CC}_{3,1}(f))$ is always at most the deterministic communication complexity of f.

Theorem 5. Let $f\colon \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ be any function computable by a size S, depth d majority circuit with arbitrary input preprocessing — that is, there exists a depth-d circuit C composed of S many majority gates, and arbitrary functions g, h, such that $f(x,y) = C(g(x), h(y))$. Then, $\mathsf{CC}_{4^d,1}(f) \leq \mathsf{poly}(S, 2^d)$.

This implies in particular that any TC^0 function family has constant-round, 1-clean catalytic protocols with $\mathsf{poly}(n)$ catalytic space.

We view these results as indicating that catalytic communication complexity is both an interesting model in its own right, and one that can shed light on reusing space more generally. We remark that our model addresses a question raised in prior work on variants of communication complexity, which we now discuss.

1.2 Related Work

Identifying an interesting communication analogue of catalytic computing was posed as an open problem by Arunachalam and Podder in a paper on space-bounded communication [AP21]. Space-bounded communication complexity was first studied by Brody, Chen, Papakonstantinou, Song and Sun, who proposed a model in which the two parties in a standard communication protocol are in addition required to compress each of their states into a small number of bits after each message is sent [Bro+13]. Subsequent work by Papakonstantinou, Scheder, and Song considered a one-way model, in which Alice (a party with unbounded memory) sends messages of length s to Bob (a party with zero, or only constant, memory) — they showed that when Bob is memoryless, this model can be characterized in terms of a combinatorial notion of rectangle overlays, and when Bob has only 5 available memory states and $s = \mathsf{polylog}(n)$ this model computes exactly PSPACE^{cc} [PSS14]. Arunachalam and Podder suggested instead letting both Alice and Bob be memoryless, and measuring complexity in terms of the size of a block of memory they pass back and forth [AP21]. They noted that this model aligns closely with bipartite branching program complexity, and that variants could capture the aforementioned models of Brody, Chen, Papakonstantinou, Song and Sun [Bro+13] and Papakonstantinou, Scheder and Song[PSS14], as well as the "garden hose" model of Buhrman, Fehr, Schaffner and Speelman [Buh+13]. Our model can be thought of as a version of Arunachalam and Podder's memoryless framework where the size of the block of memory passed back and forth is *amortized*, in a sense analogous to the way workspace usage is amortized across inputs in a catalytic algorithm.

2 Preliminaries

We will by convention denote a setting of the catalytic portion of the blackboard by τ (for "transparent registers"), a setting of the initially-clean portion by ω (for "work registers"), and a setting of the entire blackboard by γ (for no particular reason). We will let $x \in \{0,1\}^{n_a}$ be the input given to Alice, and $y \in \{0,1\}^{n_b}$ be the input given to Bob. We now define a few terms and notations which appear in the statements and proofs.

Definition 3. For any n, we denote by IP_n the inner product function over $\mathbb{F}_2^n$. That is,

$$\mathsf{IP}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\},$$

$$\mathsf{IP}(x,y) = \sum_{i \in [n]} x_i y_i \mod 2.$$

Definition 4. For any n, we denote by EQ_n the equality function on n-bit inputs. That is,

$$\mathsf{EQ}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\},$$

$$\mathsf{EQ}(x,y) = \begin{cases} 1 \text{ if } x = y \\ 0 \text{ if } x \neq y \end{cases}.$$

Definition 5. For any n, we denote by IND_n the function taking an n-bit index and a 2^n-bit bitstring to the value of that bitstring at that index. That is,

$$\mathsf{IND}_n : [2^n] \times \{0,1\}^{2^n} \to \{0,1\},$$

$$\mathsf{IND}_n(x, y) = y[x].$$

Definition 6. For a function $f : \{0,1\}^{n_a} \to \{0,1\}^{n_b} \to \{0,1\}$, let the ***communication matrix*** of f be the $2^n \times 2^n$ matrix M where $M_{x,y} = f(x, y)$, and let $\mathrm{rk}(f)$ be the rank of M over $\mathbb{F}_2$.

For a function on two inputs $f \colon X, Y \to Z$, we will use $f(x, \cdot)$ to denote the application $f(x, \cdot) \colon Y \to Z$. We say a function $f \colon X, Y \to Z$ is ***left-injective*** if there do not exist $x \neq x' \in X$ such that $f(x, \cdot)$ and $f(x', \cdot)$ are identical functions. For integers n, we will write $[n]$ to denote $\{1, \ldots, n\}$. To denote to a multiset with elements $a_1, \ldots, a_n$, we will use the "bag" notation $\lbag a_1, \ldots, a_n \rbag$.

3 Characterizing 3-Round 1-Clean Complexity

In this section, we prove Theorem 1.

Theorem 1. For any $f \colon \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$,

$$\frac{\mathrm{rk}(f)}{6} - o(1) \leq \mathsf{CC}_{3,1}(f) \leq \mathrm{rk}(f),$$

where $\mathrm{rk}(f)$ denotes the rank of the communication matrix over $\mathbb{F}_2$.

We first give the upper bound, showing that any function with small rank has an efficient catalytic protocol.

Lemma 1. For any $f \colon \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$, we have $\mathsf{CC}_{3,1}(f) \leq \mathrm{rk}(f)$.

Proof. By definition of rank, the rows of f's communication matrix span a $\mathrm{rk}(f)$-dimensional subspace of $\mathbb{F}_2^n$. Both parties fix ahead of time a basis $v_1, \ldots v_{\mathrm{rk}\, f}$ for this subspace, and define a protocol as follows:

i) On the first round, Alice finds $T \subseteq [\mathrm{rk}\, f]$ such that $\bigoplus_{i \in T} v_i$ is the xth row of the communication matrix (i.e., is equal to the truth table of the function $y \mapsto f(x, y)$), and sets the catalytic portion of the blackboard to $\tau \oplus 1_{i \in T}$.

ii) Bob computes $\bigoplus_{i \in [\mathrm{rk}\, f]} \tau_i v_i$, and writes the yth entry of the result to the clean portion.

iii) Alice XORs out the same string she XORed in, resetting the catalytic portion.

iv) Bob finds the new yth entry of $\bigoplus_{i \in [\mathrm{rk}\, f]} \tau_i v_i$, and outputs its XOR with the clean bit.

The correctness of this protocol is straightforward. After the second round of the protocol, Bob sets the clean bit to

$$b_1 = \left(\bigoplus_{i \in [\mathrm{rk}\, f]} (\tau_i \oplus 1_{i \in T}) v_i \right)_y,$$

and so his final output will be

$$b_1 \oplus \left(\bigoplus_{i \in [\mathrm{rk}\, f]} \tau_i v_i \right)_y = \left(\bigoplus_{i \in T} v_i \right)_y = f(x, y),$$

as desired. The resource consumption is immediate from the protocol definition. ■

We would like to show that it is impossible to do substantially better than the above protocol. Note that the protocol described in Lemma 1 follows a simple pattern: Alice modifies only the catalytic portion of the blackboard, performing some bijective map on all possible settings, Bob uses only the clean portion of the blackboard to "remember" one bit about the result, and then Alice undoes her bijection and Bob makes his output decision based on his remembered bit in addition to that initial catalytic configuration. For the purposes of showing lower bounds, it would be helpful first to claim that without loss of generality all 3-round protocol must be of that form. However, this is not *quite* true. For one, Alice could use the clean portion of the blackboard to send Bob an additional bit of information about her input on the first round, which is provably helpful sometimes. Another rather more substantial concern is that there's room for "amortization" in the amount of information that Bob remembers: it could, for instance, be possible that, on some particular settings of Alice's input x and the initial catalytic setting C, Alice can on her first round encode all of the catalytic information in a small prefix of the tape — in such cases, Bob could use the rest to send Alice a large amount of information about *his* input, which would then inform her in choosing a single bit send him back in the clean space of the final setting. However, we expect that this latter sort of behaviour should only happen rarely, since most initial catalytic settings can't be compressed. In Appendix B, we address these concerns formally, guaranteeing that any 3-round 1-clean catalytic protocol can be converted into a protocol that *almost* follows the simple pattern we noted. Specifically, we show the following:

Lemma 2. Every left-injective function $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ with $\mathsf{CC}_{3,1}(f) = c$ has a protocol of the following form, which we'll call a ***mostly-one-way-catalytic protocol***:

i) For every $x \in \{0,1\}^{n_a}$, Alice has an injective function $\alpha^{(x)} : \{0,1\}^c \to \{0,1\}^c \times \{0,1\}^3$.

ii) For every $y \in \{0,1\}^{n_b}$, Bob has two functions, $\beta^{(y)}_{\text{rem}} : \{0,1\}^c \times \{0,1\}^3 \to \{0,1\}$ and $\beta^{(y)}_{\text{out}} : \{0,1\}^c \times \{0,1\} \to \{0,1\}$.

iii) Call a pair $(x, \tau) \in \{0,1\}^{n_a} \times \{0,1\}^c$ ***bad*** if, for some $y \in \{0,1\}^{n_b}$, we have $\beta^{(y)}_{\text{out}}(\tau, \beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau))) \neq f(x, y)$. Then, at most 2^{c+1} many pairs are bad.

Additionally, we may assume that, for every τ, y, we have $\beta^{(y)}_{\text{out}}(\tau, 0) \neq \beta^{(y)}_{\text{out}}(\tau, 1)$.

This observation will allow us to associate a communication protocol with a structured collection of vectors in $\mathbb{F}_2^{2^{n_b}}$, whose size we can bound using standard combinatorial facts.

Proof of Theorem 1. The upper bound was shown in Lemma 1; it remains to show the lower bound. Note that removing duplicate rows from the communication matrix of a function can change neither its rank, nor the catalytic communication complexity (since a protocol for the function on the larger domain could simply treat several of its inputs identically). So, it suffices to show the claim for left-injective functions. Also, note that, given the $o(1)$ term in the statement, it suffices to only consider the case where $\text{rk}(f) \geq 1000$. We fix an arbitrary left-injective $f \colon \{0,1\}^{n_a} \times \{0,1\}^{n_b}$ with $\text{rk}(f) \geq 1000$, and assume for contradiction that $\mathsf{CC}_{3,1}(f) < \text{rk}(f)/6$.

By Lemma 2, we have a mostly-one-way-catalytic protocol for f with $c < \text{rk}(f)/6$. We will use this protocol to define three multi-subsets $U, V, W \subseteq \{0,1\}^{n_b}$. For every $\gamma \in \{0,1\}^{c+3}$, let $u_\gamma \in \{0,1\}^{n_b}$ be the truth table of $y \mapsto \beta^{(y)}_{\text{rem}}(\gamma)$, and let $U = \Lbag u_\gamma \colon \gamma \in \{0,1\}^{c+3} \Rbag$. For every $\tau \in \{0,1\}^c$, let $v_\tau \in \{0,1\}^{n_b}$ be the truth table of $y \mapsto \beta^{(y)}_{\text{out}}(\tau, 0)$, and let $V = \Lbag v_\tau \colon \tau \in \{0,1\}^c \Rbag$. Finally, for every bad $(x, \tau) \in \{0,1\}^{n_a} \times \{0,1\}^c$, let $w_{(x,\tau)}$ be the truth table of $y \mapsto f(x, y) \oplus \beta^{(y)}_{\text{out}}(\tau, 0)$, and let $W = \Lbag w_{(x,\tau)} \colon (x, \tau) \text{ is bad} \Rbag$.

Note that $|U| = 2^{c+3}$, $|V| = 2^c$, and $|W| \leq 2^{c+1}$. The idea of the proof will be to show that $U \cup W$ contains the neighbours in an appropriate Boolean hypercube of every element of V (with appropriate multiplicity) — since small subsets of the hypercube have large vertex boundary compared to their volume, but we know that $|U| + |W|$ is only a constant factor larger than $|V|$, this will allow us to give a lower bound on c. This approach is formalized as follows.

Imagine placing red and blue pebbles on $\mathbb{F}_2^{n_b}$, such that each element of $\mathbb{F}_2^{n_b}$ gets a number of red pebbles equal to the number of times it appears in $U \cup W$, and a number of blue pebbles equal to the number of times it appears in V. We claim that the following must hold:

Claim 1. For any such pebbling constructed from a valid protocol, for any $k \in \mathbb{N}$, if $v \in \mathbb{F}_2^{n_b}$ has at least k blue pebbles and $r \in \mathbb{F}_2^{n_b}$ is a row of f's communication matrix (i.e. r is the truth table of $y \mapsto f(x, y)$ for some x), then $v \oplus r$ has at least k red pebbles.

Proof of Claim 1. Because v has at least k blue pebbles, there are at least k distinct initial catalytic configurations $\tau_1, \dots, \tau_k \in \{0,1\}^c$ such that $v_{\tau_i} = v$. Let x be the input such that r is the truth table of $y \mapsto f(x, y)$. For every bad τ_i, we have $w_{(x,\tau_i)} = r \oplus v_{\tau_i} = v \oplus r$, so each of these will contribute a red pebble to $v \oplus r$. Then, since $\alpha^{(x)}$ is injective, each of $\alpha^{(x)}(\tau_1), \dots, \alpha^{(x)}(\tau_k)$ must be distinct. So, to obtain the remainder of the red pebbles, we'll show that whenever (x, τ_i) is not bad, $u_{\alpha^{(x)}(\tau_i)} = v \oplus r$.

Note that r is the truth table of $y \mapsto f(x, y) \oplus v$, that $v = v_{\tau_i}$ is the truth table of $y \mapsto \beta^{(y)}_{\text{out}}(\tau_i, 0)$, and that $u_{\alpha^{(x)}(\tau_i)}$ is the truth table of $y \mapsto \beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau_i))$. So, to conclude that $r = v \oplus u_{\alpha^{(x)}(\tau_i)}$, we need to show that, for all y, we have $f(x, y) = \beta^{(y)}_{\text{out}}(\tau_i, 0) \oplus \beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau_i))$.

Since (x, τ_i) is not bad, we have $f(x, y) = \beta^{(y)}_{\text{out}}(\tau, \beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau)))$. If $\beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau)) = 0$, this gives $f(x, y) = \beta^{(y)}_{\text{out}}(\tau, 0) = \beta^{(y)}_{\text{out}}(\tau, 0) \oplus 0$, so the claim holds. If, on the other hand, $\beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau)) = 1$, we have $f(x, y) = \beta^{(y)}_{\text{out}}(\tau, 1)$. But then, since we always have $\beta^{(y)}_{\text{out}}(\tau, 0) \neq \beta^{(y)}_{\text{out}}(\tau, 1)$, we know that $\beta^{(y)}_{\text{out}}(\tau, 1) = \beta^{(y)}_{\text{out}}(\tau, 0) \oplus 1$. ■

This will be sufficient to prove that there must be many blue pebbles, contradicting the assumption that c is small. Consider the graph on $\mathbb{F}_2^{n_b}$ where an edge (u, v) exists whenever $u \oplus v$ is a row of f's communication matrix. This graph consists of $2^{n_b}/2^{\text{rk}(f)}$ many identical connected components. If we fix some subset $X \subseteq \{0,1\}^{n_a}$ such that the corresponding rows of the communication matrix form a basis for the row-space, and consider only the edges generated by those rows, each of these connected components will be isomorphic to the $\text{rk}(f)$-dimensional hypercube. At least one of these connected components must contain a blue pebble; we will restrict our attention to one such connected component.

Claim 1 ensures that each vertex of this hypercube has at least as many red pebbles as the maximum number of blue pebbles among its neighbours. Consider the set S of vertices with at least one blue pebble. All vertices adjacent to a vertex of S must have at least one red pebble; we claim that there are many such vertices.

Claim 2. For any subset $S \subseteq \mathbb{F}_2^{\text{rk}(f)}$ with $|S| \leq 2^{\text{rk}(f)/6}$, we have $|N(S)| > 10 \cdot |S|$, where $N(S)$ denotes the set of all Hamming neighbours of elements of S.

Proof of Claim 2. Take S to be a set of the smallest possible size such that $|N(S)| \leq 10 \cdot |S|$. Harper's theorem states that, among all subsets of the Boolean hypercube of a given size, the vertex boundary is minimized by a Hamming ball. That is, for any ℓ, if $|S| = \sum_{i=0}^{\ell} \binom{\text{rk}(f)}{i}$, then $|N(S) \setminus S| \geq \binom{\text{rk}(f)}{\ell+1}$ [Har66; Bol86]. Let ℓ be the radius of the smallest Hamming sphere larger than our fixed set S; i.e., the smallest integer such that $|S| \leq \sum_{i=0}^{\ell} \binom{\text{rk}(f)}{i}$. Since $\min_{S\colon |S|=s} \left(\frac{|N(S)|}{|S|}\right)$ is monotonically nonincreasing in s for small s[3], the surface-area-to-volume ratio of this Hamming sphere lower bounds that of S:

$$\frac{|N(S)|}{|S|} \geq \frac{\binom{\text{rk}(f)}{\ell+1}}{\sum_{i=0}^{\ell} \binom{\text{rk}(f)}{i}}.$$

By minimality of S, we know that $\sum_{i=0}^{\ell'} \binom{\text{rk}(f)}{i} \leq \binom{\text{rk}(f)}{\ell'+1}$ for all $\ell' < \ell$, since otherwise the Hamming ball of radius ℓ' would be a smaller example. So, $\sum_{i=0}^{\ell} \binom{\text{rk}(f)}{i} \leq \binom{\text{rk}(f)}{\ell} + \sum_{i=0}^{\ell-1} \binom{\text{rk}(f)}{i} \leq 2\binom{\text{rk}(f)}{\ell}$. Thus,

$$\frac{|N(S)|}{|S|} \geq \frac{\binom{\text{rk}(f)}{\ell+1}}{2\binom{\text{rk}(f)}{\ell}} = \frac{\frac{\text{rk}(f)!}{(\ell+1)!(\text{rk}(f)-\ell-1)!}}{\frac{2\,\text{rk}(f)!}{\ell!(\text{rk}(f)-\ell)!}} = \frac{(\text{rk}(f) - \ell)}{2(\ell + 1)}.$$

[3] As long as s is small enough that a Hamming ball of that size can't have vertex boundary less than twice the volume (which, as we show, is true here), one can observe that adding a new point at the boundary will always increase the surface-area-to-volume ratio.

Since we assumed that $\frac{|N(S)|}{|S|} \leq 10$, this gives $\ell \geq \frac{\mathrm{rk}(f)-19}{20}$. Now, we have

$$|S| \geq \sum_{0}^{\ell-1} \binom{\mathrm{rk}(f)}{i} \geq \binom{\mathrm{rk}(f)}{\ell-1} \geq 2^{\mathrm{rk}(f)\cdot H\left(\ell/\,\mathrm{rk}(f)\right)-\log\left(\mathrm{rk}(f)\right)} \geq 2^{\mathrm{rk}(f)\cdot H\left(1/20-1/\,\mathrm{rk}(f)\right)-\log\left(\mathrm{rk}(f)\right)},$$

where $H\colon p \mapsto -p\log(p)-(1-p)\log(1-p)$ is the binary entropy function. For all values of $\mathrm{rk}(f) \geq 1000$, we have $\mathrm{rk}(f)H\big(1/20-1/\,\mathrm{rk}(f)\big)-\log\big(\mathrm{rk}(f)\big) > \frac{\mathrm{rk}(f)}{6}$, so this implies that $|S| > 2^{\left(\mathrm{rk}(f)/6\right)}$. ■

We can't immediately a derive contradiction from Claim 1 and Claim 2, because V is a multiset (i.e. some vertices may have more than one red pebble). However, this is not a serious obstacle.

Claim 3. Suppose a pebbling satisfies the conditions of Claim 1 —i.e., that $v \oplus r$ always has at least as many blue pebbles as v has red pebbles for any row r of the communication matrix. Then, if we remove one blue pebble from every vertex with at least one blue pebble, and one red pebble from any vertex with at least one red pebble, that condition still holds.

Proof. Consider any v and any r. Before removal, $v \oplus r$ had at least as many blue pebbles as v had red pebbles. If v had 0 blue pebbles, then after removal this will remain the case, so $v \oplus r$ will still be at least as pebbled. Otherwise, we will remove the same number of blue pebbles from v as we remove red pebbles from $v \oplus r$. ■

If we remove a single blue pebble from every vertex with at least one blue pebble, and a single red pebble from every vertex with at least one red pebble, Claim 2 ensures that we will remove strictly more than ten times as many red pebbles as blue pebbles. After removing those pebbles, there are still at most 2^c vertices with blue pebbles, and we've just shown that Claim 1 still holds — so, we can repeatedly perform such removals until all pebbles are gone, always removing strictly more than ten times as many red pebbles as blue pebbles. This means that the graph overall has *strictly* more than ten times as many red pebbles as blue pebbles, contradicting the fact that $|U| + |W| = 2^{c+3} + 2^{c+1} = 10 \cdot 2^c = 10 \cdot |V|$. ■

One interesting consequence of Theorem 1 is that the equality function, which has constant randomized communication complexity, requires the maximal $\Omega(2^n)$ blackboard size in this catalytic model. On the other hand, as noted, the inner product function can be communicated with only n bits of catalytic space, despite requiring (up to constant factors) maximal standard communication complexity even with randomness [RY20]. Catalytic communication complexity seems to be quantifying a very different notion from standard communication complexity measures.

A natural question at this stage is whether our choice to restrict to only a single bit of clean space was roughly without loss of generality, or whether protocols with more clean space can look substantially different. Having more free space is definitely at least moderately helpful: for instance, equality can be computed with 2 bits of free space and $2 \cdot 2^{n/2}$ bits of catalytic space, by simply running two copies of the 1-bit protocol on the two halves of the inputs simultaneously, and having Bob output the AND of the answers. But one might suspect that, with only a little more clean space, there is not too much more that can be done — perhaps for any constant s it's possible to show that $\mathsf{CC}_{3,s}(\mathsf{EQ}_n) \geq 2^{\Omega(n)}$. As we will show in the next section, this intuition turns out to be false — even with two just bits of clean space, it's possible to communicate equality very efficiently.

4 The 3-Round Complexity of Equality

In this section, we prove the following upper bound on 3-round 2-clean catalytic complexity of equality.

Theorem 2. We have that $\mathsf{CC}_{3,2}(\mathsf{EQ}_n) \leq O(n \log n)$.

The key to the proof of Theorem 2 comes from a long line of research on graphs representable as the union of many large induced matchings, known as Ruzsa–Szemerédi graphs. We will provide background on the Ruzsa–Szemerédi problem in Section 4.1, where we will also introduce a variant important for our application. In Section 4.2, we will show that a known construction can be modified to satisfy the requirements of that variant definition. Then, in Section 4.3, we will demonstrate how any such construction can be generically converted into a catalytic protocol for equality, which will in particular prove Theorem 2.

In Section 4.4 we will demonstrate a reverse implication — that is, that efficient 3-round equality protocols give dense Ruzsa–Szemerédi protocols — which will allow us to state catalytic communication complexity lower bounds. We will show unconditionally that there exists no 3-round $O(1)$-clean protocol for equality using $n+O(1)$ bits of catalytic space, and show that finding any stronger upper bound than $O(n \log n)$ would require improved graph theoretic constructions that would in turn prove new *lower* bounds in areas such as property testing and streaming algorithms.

4.1 Background on the Ruzsa–Szemerédi Problem

We give the following definition:

Definition 7. A graph G is an (r,t)-***Ruzsa–Szemerédi graph*** if there exists a partition of its edges into t sets of size r, such that each set constitutes an induced matching in G.

In 1976, motivated by a problem of Brown, Erdős and Sós on 3-uniform hypergraphs containing no 6 vertices sharing 3 edges [BET73], Ruzsa and Szemerédi proposed studying such graphs, with the goal of finding examples where both r and t are large compared to number of vertices of G. They showed that Behrend's construction of sets without 3-term arithmetic progression [Beh46] gives $\left(\frac{n}{2^{O(\sqrt{\log(n)})}}, \frac{n}{3}\right)$-Ruzsa–Szemerédi graphs, and used regularity methods to show that r and t cannot both be made $\Omega(n)$ [RS76]. The best known upper bounds still go through graph regularity: Fox's improved bounds for triangle removal imply that if $r = \Theta(n)$, then $t \leq \frac{n}{2^{\Omega(\log^*(n))}}$ [Fox11].

Ruzsa–Szemerédi graphs have since seen several applications in computational and communication complexity [HW03; BLM06; LPS19]. In particular, constructions where $r = \Theta(n)$ and t is large have been used to obtain a number of lower bounds in streaming algorithms and property testing [Fis+02; GKK; KN21; Kap21; AS23]. The best known lower bound on t when $r = \Theta(n)$ is due to Fischer, Lehman, Newman, Raskhodnikova, Rubinfeld and Samorodnitsky, giving a construction where $r = n/3$ and $t \geq n^{\Omega\left(\frac{1}{\log\log(n)}\right)}$ — eliminating the gap between this bound and the $t \leq \frac{n}{2^{\Omega(\log^*(n))}}$ upper bound is a major open problem in combinatorics [Fis+02].

For our application, we also give the following more restrictive definition:

Definition 8. A graph G is a (k,t)-***full Ruzsa–Szemerédi graph*** if there exists a partition of its edges into t perfect matchings, and a partition of each of those matchings into k partial matchings, such that each partial matching is induced in G.

Note that a (k,t)-full Ruzsa–Szemerédi graph is in particular a $(n/2k, kt)$-Ruzsa–Szemerédi graph, but that here we are imposing an additional constraint by requiring that the induced matchings can be grouped together to form perfect matchings in G. The construction presented by Fischer, Lehman, Newman, Raskhodnikova, Rubinfeld and Samorodnitsky is not a full Ruzsa–Szemerédi graph, but we will show in the next section that it can easily be made so.

4.2 Dense Construction of a Full Ruzsa–Szemerédi Graph

In this section, we give a construction of a full Ruzsa–Szemerédi graph with a large number of edges, following the same framework as Fischer, Lehman, Newman, Raskhodnikova, Rubinfeld and Samorodnitsky [Fis+02].

Lemma 3. As long as $n = 2\ell^\ell$ for some integer ℓ, there exists a $\left(3, n^{\Omega\left(\frac{1}{\log\log n}\right)}\right)$-full Ruzsa–Szemerédi graph.

Proof. By the Gilbert-Varshamov bound in coding theory for constant-weight codes [Lev71], for any ℓ there exists a family $\mathcal{S} \subseteq \mathcal{P}([\ell])$ such that $|\mathcal{S}| \geq 2^{\Omega(\ell)}$, $|S| = \ell/3$ for all $S \in \mathcal{S}$, and $|S \cap T| \leq \ell/6$ for all $S \neq T \in \mathcal{S}$. We will use $\mathcal{S}$ to determine the edge set of a bipartite graph on $2\ell^\ell$ vertices, where the vertices are identified with two copies of $\{0, \ldots, \ell-1\}^\ell$. For each $S \in \mathcal{S}$, let M_S be the perfect matching obtained by connecting each left vertex x with the right vertex $(x + 1_S \mod \ell)$, where use the identification of the vertices with vectors in $\{0, \ldots, \ell-1\}^\ell$ and add coordinatewise. We let the edge set of the graph be the union of all these perfect matchings.

In order to demonstrate that this is a full Ruzsa–Szemerédi graph, we must partition each M_S into three partial matchings, such that each is an induced matching in the graph. To do so, we define for each S a vertex 3-colouring π_S, where

$$\pi_S(x) = \begin{cases} \left\lfloor \frac{6\sum_{i \in S} x_i}{\ell} \right\rfloor \mod 3 & \text{if } x \text{ belongs to the left part} \\ \left\lfloor \frac{6\sum_{i \in S} x_i}{\ell} \right\rfloor + 1 \mod 3 & \text{if } x \text{ belongs to the right part} \end{cases}.$$

The 3 partial matchings of M_S will be the subgraphs induced by the vertices of each of the 3 colours. We need to show that these 3 partial matchings contain between them all the edges of M_S, and that none of the three vertex sets contain an edge from M_T for any $T \neq S$. To show the first of these properties, we note that, for any x,

$$\left\lfloor \frac{6\sum_{i \in S}(x + 1_S)_i}{\ell} \right\rfloor + 1 = \left\lfloor \frac{6\left(|S| + \sum_{i \in S} x_i\right)}{\ell} \right\rfloor + 1 = \left\lfloor \frac{6\sum_{i \in S} x_i}{\ell} \right\rfloor + 3.$$

Also, since adding a multiple of ℓ to a coordinate of x changes $6\sum_{i \in S} x_i$ by a multiple of 6ℓ, modular overflow of individual coordinates doesn't change this value mod 3. So, the endpoints of each edge of M_S have equal colours under π_S, meaning that the partial matchings contain all edges. To show that each of these partial matchings is induced, we need to show that, for any $T \neq S$, the endpoints of any edge of M_T are assigned non-equal colours under π_S. This follows because, for any x,

$$\left\lfloor \frac{6\sum_{i \in S} x_i}{\ell} \right\rfloor + 1 \leq \left\lfloor \frac{6\sum_{i \in S}(x + 1_T)_i}{\ell} \right\rfloor + 1 = \left\lfloor \frac{6\left(|S \cap T| + \sum_{i \in S} x_i\right)}{\ell} \right\rfloor + 1 \leq \left\lfloor \frac{6\sum_{i \in S} x_i}{\ell} \right\rfloor + 2. \qquad \blacksquare$$

4.3 Equality Protocols from Full Ruzsa–Szemerédi Graphs

We now show that full Ruzsa–Szemerédi graphs can be generically converted to catalytic protocols.

Lemma 4. If there exists a (k, t)-full Ruzsa–Szemerédi graph on 2^c vertices, then there exists a 3-round, $\lceil \log(k) \rceil$-clean catalytic protocol for equality on $\lfloor \log(t) \rfloor$-bit inputs with c bits of catalytic space[4].

We first note that this will immediately give us our desired upper bounds on $\mathsf{CC}_{3,2}(\mathsf{EQ}_n)$.

Proof of Theorem 2 given Lemma 4. For some constant $0 < \delta < 1$, by Lemma 3, there exists a $(3, N^{\frac{\delta}{\log \log N}})$-full Ruzsa–Szemerédi graph on N vertices. Plugging this in to Lemma 4, we obtain a $\lceil \log(3) \rceil = 2$-clean catalytic protocol for equality on $\lfloor \log(t) \rfloor = \lfloor \delta \log(N)/\log\log(N) \rfloor$ bit inputs, requiring $\log(N)$ bits of catalytic space. Defining $n = \lfloor \log(t) \rfloor$, this is a 3-round 2-clean protocol with $O(n \log n)$ bits of catalytic space[5]. $\blacksquare$

[4] Observe though, if one cares about such things, that unless that Ruzsa–Szemerédi graph can be constructed explicitly, the resulting protocol may not be computationally efficient. In order to get a computationally effective protocol from our construction, one would have to plug in an explicit code as opposed to just citing Gilbert-Varshanov.

[5] Note that this construction is only defined when both N is a power of two, and $N = 2 \cdot \ell^\ell$ for some integer ℓ. However, if we let ℓ be the smallest power of two such that $2 \cdot \ell^\ell \geq N$, and let $N' = 2 \cdot \ell^\ell$, then note that $\log(N') \leq 4 \cdot \log(N)$. So, for other values of n, Alice and Bob can simply pad their inputs with zeroes and run the equality protocol for a value of n where this is defined, losing only a constant factor in the amount of catalytic space required.

Proof of Lemma 4. Let G be such a graph, with vertices $V(G) = v_1, \ldots, v_{2^c}$, composed of perfect matchings $M_1 \sqcup \cdots \sqcup M_t = E(G)$. The full-Ruzsa–Szemerédi property gives, for each $i \in [t]$, a vertex partition $M_i^{(1)} \sqcup \cdots \sqcup M_i^{(k)} = V(G)$ such that the edges of M_i are the union over j of the edges induced by $M_i^{(j)}$. Fix an arbitrary bijection $\pi\colon \{0,1\}^c \to \{v_1, \ldots, v_N\}$, and arbitrary injections $\sigma\colon \{0,1\}^{\lfloor \log(t) \rfloor} \to \{M_1, \ldots, M_t\}$, $\mu\colon [k] \to \{0,1\}^{\lceil \log k \rceil}$. The protocol is as follows.

i) $A_1(x, \tau, \omega) = (\pi^{-1}(v), \omega)$, where v is the neighbour of $\pi(\tau)$ in the matching $\sigma(x)$.

ii) $B_2(y, \tau, \omega) = (\tau, \mu(w))$, where w is the unique value such that $\pi(\tau) \in \sigma(y)^{(w)}$.

iii) $A_3(x, \tau, \omega) = (\pi^{-1}(v), \omega)$, where v is the neighbour of $\pi(\tau)$ in the matching $\sigma(x)$.

iv) $B_{\text{out}}(y, \tau, \omega)$ accepts if and only if $\omega = \mu(w)$, where w is the unique value such that $\pi(\tau) \in \sigma(y)^{(w)}$.

Correctness of this protocol follows from the definition of a full Ruzsa–Szemerédi graph. If $x = y$, then the matchings $\sigma(x)$ and $\sigma(y)$ are the same. Since every edge of $\sigma(x)$ is induced by some part of the partition $\sigma(x)^{(1)}, \ldots, \sigma(x)^{(k)}$, this means that both endpoints of that edge belong to the same $\sigma(y)^{(w)}$, and so Bob will accept. If, on the other hand, $x \neq y$, then the two endpoints of an edge in $\sigma(x)$ cannot belong to the same $\sigma(y)^{(k)}$, because $\sigma(y)^{(k)}$ induces only the edges belonging to $\sigma(y)$. ■

4.4 Lower Bounds on Equality Protocols

In this section, we will show that, as long as we are content with a not-necessarily-full Ruzsa–Szemerédi graph, the conversion in Lemma 4 can be done in reverse. That is, the existence of dense Ruzsa–Szemerédi graphs is necessary for efficient equality protocols. This will allow us to deduce both conditional and unconditional communication lower bounds.

Lemma 5. For any s, if there exists a 3-round, s-clean catalytic protocol for equality on n-bit inputs with c bits of catalytic space, there exists an $(\Omega(2^{c-s}), \Omega(2^{n-s}))$-Ruzsa–Szemerédi graph on $O(2^{c+4^s})$ vertices.

For the purposes of proving this lemma, it would simplify matters to know that Alice only ever modifies the catalytic portion of the blackboard, and Bob only ever modifies the clean portion[6]. As in Section 3, although we cannot claim such behaviour without loss of generality, we can at least claim that, a substantial fraction of the time, Bob only remembers s bits.

Lemma 6. For any $s > 0$, any function $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ with $\mathsf{CC}_{3,s}(f) = c$ has a protocol of the following form, which we'll call a ***sometimes-one-way-catalytic protocol***:

i) For every $x \in \{0,1\}^{n_a}$, Alice has an injective function $\alpha^{(x)} : \{0,1\}^c \to \{0,1\}^c \times \{0,1\}^{s \cdot (2^s+1)}$.

ii) For every $y \in \{0,1\}^{n_b}$, Bob has two functions, $\beta^{(y)}_{\text{rem}} : \{0,1\}^c \times \{0,1\}^{s \cdot (2^s+1)} \to \{0,1\}^s$ and $\beta^{(y)}_{\text{out}} : \{0,1\}^c \times \{0,1\}^s \to \{0,1\}$.

iii) Call a pair $(x, \tau) \in \{0,1\}^{n_a} \times \{0,1\}^c$ ***bad*** if, for some $y \in \{0,1\}^{n_b}$, we have $\beta^{(y)}_{\text{out}}(\tau, \beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau))) \neq f(x, y)$. Then, at most a $2^s/(2^s+1)$ fraction of all pairs are bad.

Proof of Lemma 5. By Lemma 6, we have a sometimes-one-way catalytic protocol for f with c bits of catalytic space. From this protocol, we will construct a bipartite graph G whose 2^c many left vertices are identified with $\{0,1\}^c$, and whose $2^{c+s\cdot(2^s+1)}$ many right vertices are identifed with $\{0,1\}^c \times \{0,1\}^{s\cdot(2^s+1)}$. The edge set will consist of, for every non-bad $(x, \tau) \in \{0,1\}^n \times \{0,1\}^c$, an edge from τ on the left to $\alpha^{(x)}(\tau)$ on the right.

Now, for every $z \in \{0,1\}^n$, $\omega \in \{0,1\}^s$, we'll define a vertex subset $M_z^{(\omega)} \subseteq V(G)$. A left vertex $\tau \in \{0,1\}^c$ will be included in $M_z^{(\omega)}$ if and only if both (z, τ) is non-bad, and $\beta^{(z)}_{\text{rem}}(\alpha^{(z)}(\tau)) = \omega$. A right vertex $\gamma \in \{0,1\}^c \times \{0,1\}^{s\cdot(2^s+1)}$ will be included in $M_z^{(\omega)}$ if and only if $\beta^{(z)}_{\text{rem}}(\gamma) = \omega$.

[6] In fact, in this case we note that an analysis similar to the one we present here would give a *full*-Ruzsa–Szemerédi graph, as opposed to just a Ruzsa–Szemerédi graph, demonstrating (up to some quantitative loss) an equivalence between full Ruzsa–Szemerédi graphs and this kind of "well-behaved" 3-round catalytic protocol for equality. We suspect that such an equivalence should also exist without the well-behavedness condition — perhaps even that full Ruzsa–Szemerédi graphs can be generically constructed from Ruzsa–Szemerédi graphs — however we do not know a proof.

We claim that every edge of G is induced by one of these subsets, and that each subset induces a matching. The first part of this is immediate: both endpoints of the edge generated by a pair (x, τ) must belong to $M_x^{(\beta_{\mathrm{rem}}^{(x)}(\alpha^{(x)}(\tau)))}$. To see the second part, consider some edge $(\tau, \gamma) \in E(G)$, generated by a non-bad (x, τ), and suppose it belongs to some M_z^{ω} with $z \neq x$. (Note that, by definition of $M_x^{(\omega)}$, this edge it cannot belong to $M_x^{(\omega)}$ for any $\omega \neq \beta_{\mathrm{rem}}^{(x)}(\alpha^{(x)}(\tau))$.) But now, suppose the protocol is run with Alice given input x, Bob given input z, and the catalytic tape initialized to τ. The output of the protocol will be $\beta_{\mathrm{out}}^{(z)}(\tau, \beta_{\mathrm{rem}}^{(z)}(\alpha^{(x)}(\tau))) = \beta_{\mathrm{out}}^{(z)}(\tau, \beta_{\mathrm{rem}}^{(z)}(\gamma)) = \beta_{\mathrm{out}}^{(z)}(\tau, \omega) = \beta_{\mathrm{out}}^{(z)}(\tau, \beta_{\mathrm{rem}}^{(z)}(\alpha^{(z)}(\tau)))$. However, since both (x, τ) and (z, τ) are non-bad, we know $\beta_{\mathrm{out}}^{(z)}(\tau, \beta_{\mathrm{rem}}^{(z)}(\alpha^{(x)}(\tau))) = \mathsf{EQ}_n(x, z) = 0$ and $\beta_{\mathrm{out}}^{(z)}(\tau, \beta_{\mathrm{rem}}^{(z)}(\alpha^{(z)}(\tau))) = \mathsf{EQ}_n(z, z) = 1$, so this is contradiction.

Now, oberve that since there are at least $(2^c \cdot 2^n)/(2^s + 1)$ many non-bad pairs, G has at least this many edges. Since we've partitioned these edges into $2^n \cdot 2^s$ many induced matchings, each of which has at most 2^c many edges, we must have at least $2^n/2^{s+1}$ many induced matchings each with at least $2^c/2^{s+1}$ many edges. Letting $G' \subseteq G$ be the graph on only those edges, this is a $2^c + 2^{c+s\cdot(2^s+1)} = O(2^c)$-vertex $(\Omega(2^c), \Omega(2^n))$-Ruzsa–Szemerédi graph. ■

This characterization immediately allows us to show that no 3-round $\Theta(1)$-clean catalytic protocol for equality can use only $n + O(1)$ bits of catalytic space — although we have found a surprisingly efficient protocol, equality is at least somewhat more difficult for this model than, for instance, inner product.

Corollary 1. For any constant s, we have $\mathsf{CC}_{3,s}(\mathsf{EQ}_n) \geq n + \Omega(\log^*(n))$.

Proof. It is known by the triangle removal lemma that for any constant k, a $(N/k, t)$-Ruzsa–Szemerédi graph on N vertices must satisfy $t \leq \frac{N}{2^{\Omega(\log^*(N))}}$ [Fox11]. Since, by Lemma 5, a 3-round s-clean protocol on n-bit inputs with c bits of catalytic space gives a $(\Omega(2^c), \Omega(2^n))$-Ruzsa–Szemerédi graph on $O(2^c)$ vertices, this means that $2^n \leq O\left(\frac{2^c}{2^{\Omega(\log^*(2^c))}}\right)$. Rearranging gives $c \geq n + \Omega(\log^* n)$. ■

Of course, this doesn't rule out a protocol with linear catalytic space — and in fact, the second author believes that such a protocol likely exists. However, it does provide evidence that constructing such a protocol may be difficult: doing so would improve known Ruzsa–Szemerédi constructions, yielding improvements to state-of-the-art bounds in property testing, streaming algorithms, and information theory. We mention a few such implications; the reader is referred to the cited works for definitions of terms.

Corollary 2. If, for some constant s, $\mathsf{CC}_{3,s}(\mathsf{EQ}_n) \leq O(n)$, then

i) Any non-adaptive tester for monotonicity over general N-element posets requires query complexity $\Omega(N^c)$ for some $c > 0$.

ii) Any randomized semi-streaming algorithm for $(1 - \varepsilon)$-approximate maximum bipartite matching requires $\Omega(\log(1/\varepsilon))$ passes over the stream.

iii) There exist constant-rate centralized coding caching schemes, in which each file is divided into a number of pieces polynomial in the number of participants.

Proof. By Lemma 5, if $\mathsf{CC}_{3,s}(\mathsf{EQ}_n) \leq O(n)$, then for some $c > 0$ there exist $(\Theta(N), N^c)$-Ruzsa–Szemerédi graphs on N vertices. The property testing lower bounds of Fischer, Lehman, Newman, Raskhodnikova, Rubinfeld and Samorodnitsky [Fis+02], the streaming lower bounds of Assadi and Sundaresan [AS23], and the centralized coding caching scheme construction of Shangguan, Zhang and Ge [SZG18] all rely on constructions of Ruzsa–Szemerédi graphs where the partial matchings are of linear size. Plugging in a construction with $t = N^c$ to those arguments would immediately yield the strengthened bounds in the statement of the corollary. ■

A more optimistic interpretation of Corollary 2 is that designing catalytic protocols for equality could provide an approach towards improved Ruzsa-Semerédi constructions. There is some evidence that the language of communication complexity can be useful in this area: Linial, Pitassi, and Shraibman have shown a close relationship between Ruzsa–Szemerédi graphs and protocols for high-dimensional permutations in the Number On the Forehead model, which Alon and Shraibman used to give simple communication theoretic descriptions of a couple of known Ruzsa–Szemerédi constructions [LPS19; AS20]. 3-round catalytic protocols for equality offer another distinct communication-theoretic interpretation of the problem that may prove useful.

5 The 3-Round Complexity of Indexing

The fact that $\mathsf{CC}_{3,2}(\mathsf{EQ}_n)$ is exponentially smaller than $\mathsf{CC}_{3,1}$ raises the question of whether perhaps *every* function has an efficient 3-round s-clean protocol for some constant s. One might suspect that most functions should require exponential complexity, but there is no clear counting argument to this effect. In fact, even for functions of unbalanced input lengths — i.e. when $n_b \gg n_a$ — there's no obvious way to rule out a protocol with catalytic space close to n_a [7]. We propose considering the following function:

Definition 5. For any n, we denote by IND_n the function taking an n-bit index and a 2^n-bit bitstring to the value of that bitstring at that index. That is,

$$\mathsf{IND}_n : [2^n] \times \{0,1\}^{2^n} \to \{0,1\},$$

$$\mathsf{IND}_n(x,y) = y[x].$$

This indexing function is often considered in one-way communication complexity contexts, where the party holding the index cannot send messages, so all information must be sent by the party with the longer input. Note that in our case, however, we consider the reverse: we're giving Alice the index, and by Lemma 6 our protocol can be largely thought of as one-way from Alice to Bob. The reason we're particularly interested in this setting is that a protocol for IND_n can be thought of as a protocol for all functions simultaneously: Bob has to be able to determine any Boolean function on Alice's input. So, for fixed value of Alice's input length $n_a = n$, IND_n is the hardest possible function, in the sense that a protocol for IND_n gives a protocol with the same parameters for every other function with the same n_a.

By Proposition 2, we know $\mathsf{CC}_{3,s}(\mathsf{IND}_n) \leq 2^n$. We suspect that this is close to the correct answer for any constant s, but the results we've shown thus far haven't ruled out the possibility that $\mathsf{CC}_{3,2}(\mathsf{IND}_n) = n + O(\log^*(n))$. In this section, we will prove a somewhat stronger lower bound — still far away from the upper bound, but enough to suggest for instance that it's unlikely that some sort of Ruzsa–Szemerédi construction as in Section 4 will work directly. We show the following:

Theorem 4. We have $\mathsf{CC}_{3,s}(\mathsf{IND}_n) \geq (1+\varepsilon)n$, for some constant ε depending on s.

As in our lower bounds for equality, our proof comes from a graph theoretic interpretation. We will show that a protocol for IND_n corresponds to a graph composed of a union of matchings, with the property that any subset of the matchings are separable from the rest of the graph. We will then show that such a graph cannot have too high density, bounding the efficiency of the catalytic protocol. We make the following definitions:

Definition 9. Let E_1 and E_2 be disjoint edge sets on a common set of vertices, and let $k \in \mathbb{N}$. We say E_1 and E_2 are ***k-separable*** if there exists a k-edge colouring of $E_1 \cup E_2$ such that

- each colour appears in only one of E_1 or E_2, and
- for every 3-edge path, if the first and last edges share the same colour, so does the middle edge.

Definition 10. We call an n-vertex graph ***k-divisive*** if its edges can be partitioned into matchings $M_1, \ldots, M_m$ such that for all $S \subseteq [m]$, the edge sets $\bigcup_{i \in S} M_i$ and $\bigcup_{j \notin S} M_j$ are k-separable.

Lemma 7. For any constant s, there is a family $\{G_n\}_{n \in \mathbb{N}}$, with $|V(G_n)| \leq O(2^c)$ and $|E(G_n)| \geq \Omega(2^{c+n})$ for $c = \mathsf{CC}_{3,s}(\mathsf{IND}_n)$, and where each G_n is 2^{2^s+s}-divisive.

[7] The other direction of asymmetry — that is, when $n_a \gg n_b$ — is less interesting for 3-round protocols, since Bob can without loss of generality communicate very little information to Alice, and so there are easily functions requiring at least n_a bits of catalytic space for information theoretic reasons.

Proof. As in Section 4.4, we begin by invoking Lemma 6 to obtain a sometimes-one-way-catalytic protocol. In fact, we'll define the same graph: let G consist of 2^c many left vertices identified with $\{0,1\}^c$, $2^{c+s\cdot(2^s+1)}$ many right vertices identified with $\{0,1\}^c \times \{0,1\}^{s\cdot(2^s+1)}$, and an edge $(\tau, \alpha^{(x)}(\tau))$ for each x whenever (x,τ) is non-bad. This graph has $2^c + 2^{c+s\cdot(2^s+1)} \leq O(2^c)$ many vertices, and $2^{n+c}/(2^s+1) \geq \Omega(2^{c+n})$ many edges.

For any $x \in [2^n]$, let $M_x \subseteq E(G_n)$, $M_x = \{(\tau, \alpha^{(x)}(\tau)) \colon (x,\tau) \text{ is not bad}\}$. Note that each M_x is a matching, since $\alpha^{(x)}$ is injective. We claim that, for any $S \subseteq [2^n]$, the edge sets $\bigcup_{i\in S}$ and $\bigcup_{i\notin S} M_i$ are $\left(2^{2^s+s}\right)$-separable. Fix any $S \subseteq [2^n]$, and let $1_S \in \{0,1\}^{2^n}$ denote the indicator vector of S. Now, to separate $\bigcup_{i\in S}$ and $\bigcup_{i\notin S} M_i$, we will "colour" each edge (τ,γ) with the pair $\left(\beta^{(1_S)}_{\text{out}}(\tau,\cdot), \beta^{(1_S)}_{\text{rem}}(\gamma)\right) \in (\{0,1\}^s \to \{0,1\}) \times \{0,1\}^s$. If (τ,γ) and (τ',γ') have the same colour, it means that $\beta^{(1_S)}_{\text{out}}(\tau,\cdot)$ and $\beta^{(1_S)}_{\text{out}}(\tau',\cdot)$ are the same function, and $\beta^{(1_S)}_{\text{rem}}(\gamma)$ and $\beta^{(1_S)}_{\text{rem}}(\gamma')$ are the same bitstring — so, (τ,γ') and (τ',γ) will also share this colour.

We now just need to show that no edge $(\tau,\gamma) \in \bigcup_{i\in S} M_i$ can share a colour with an edge $(\tau',\gamma') \in \bigcup_{i\notin S} M_i$. Fix any $x, x' \in [2^n]$, and any edges $(\tau,\gamma) \in M_x$ and $(\tau',\gamma') \in M_{x'}$ with the same colour. In order for these edges to be present, both (x,τ) and (x',τ') must be good. So,

$$\begin{aligned}
f(x, 1_S) &= \beta^{(1_S)}_{\text{out}}(\tau, \beta^{(1_S)}_{\text{rem}}(\alpha^{(x)}(\tau))) \\
&= \beta^{(1_S)}_{\text{out}}(\tau, \beta^{(1_S)}_{\text{rem}}(\gamma)) \\
&= \beta^{(1_S)}_{\text{out}}(\tau', \beta^{(1_S)}_{\text{rem}}(\gamma')) \\
&= \beta^{(1_S)}_{\text{out}}(\tau', \beta^{(1_S)}_{\text{rem}}(\alpha^{(x')}(\tau'))) \\
&= f(x', 1_S),
\end{aligned}$$

meaning that either both x and x' belong to S, or neither do.

■

We now demonstrate that any such graph cannot contain a large complete bipartite subgraph, which will allow us to bound its density by the KST theorem.

Lemma 8. For any $k \in \mathbb{N}$, if a graph G is k-divisive, then G does not contain the complete bipartite graph $K_{3k^2,3k^2}$ as a subgraph.

Proof. We proceed by contradiction: suppose we've found a copy H of $K_{3k^2,3k^2}$ in G. Let $M_1 \sqcup \cdots \sqcup M_m = E(G)$ be the partition into matchings guaranteed by the divisiveness condition. We claim that we can find some set of M_i and some smaller complete bipartite subgraph of H whose intersection with the union of those M_i is a perfect matching.

Claim 4. There exists a set $S \subseteq [m]$, and a subgraph $H' \subseteq H \subseteq G$, such that H' is isomorphic to $K_{k+1,k+1}$, and $\bigcup_{i\in S}(H \cap M_i)$ is a matching with $k+1$ edges.

Proof of Claim 4. First, suppose some single M_i contains at least $k+1$ edges of H. Then, the claim follows immediately by taking $S = \{i\}$, and letting H' be the subgraph induced by the edges of $H \cap M_i$. So, we will assume each M_i contains at most k edges of H.

We construct S greedily, starting with $S = \emptyset$ and adding indices one-at-a-time. Suppose $\bigcup_{i\in S}(H \cap M_i)$ currently consists of a matching with $\ell < k+1$ many edges. The number of edges of H that contain any endpoint of an edge in $\bigcup_{i\in S}(H \cap M_i)$ is therefore no more than $2 \cdot \ell \cdot (3k^2)$. Since no M_i contains more than k edges of H, and H has $(3k^2)^2$ edges, there must be at least $\frac{(3k^2)^2}{k} = 9k^3 > 2 \cdot \ell \cdot (3k^2)$ distinct M_i with at least one edge in H. Thus, we can find some i^* such that $M_{i^*} \cap H$ contains at least one edge, but contains no edge sharing an endpoint with an edge of $\bigcup_{i\in S}(H \cap M_i)$ — this implies that $\bigcup_{i\in(S\cup\{i^*\})}(H \cap M_i)$ is a matching with at least $\ell+1$ many edges. The claim follows by repeating this procedure until $|\bigcup_{i\in S}(H \cap M_i)| \geq k+1$, and then taking H' to be the subgraph induced by those edges. ■

Now, we use H' and S to contradict the k-separability assumption. By assumption on G, the edge sets $\bigcup_{i \in S} M_i$ and $\bigcup_{i \notin S} M_i$ are k-separable — fix an edge colouring that k-separates them. Since $\bigcup_{i \in S}(H \cap M_i)$ contains $k+1$ edges, by pigeonhole principle it must contain two edges (u_1, u_2) and (v_1, v_2) that are given the same colour. Since H' is a complete bipartite graph, the edge (u_1, v_2) must also be present, and since $\bigcup_{i \in S}(H \cap M_i)$ is a matching it cannot belong to any M_i, $i \in S$. Since (u_1, u_2) and (v_1, v_2) have the same colour, the definition of a k-separation requires (u_1, v_2) to share that colour — but (u_1, v_2) does not belong to $\bigcup_{i \in S} M_i$, so that colour cannot be used on (u_1, v_2). This is contradiction. ∎

Theorem 4 now follows directly from Lemma 7 and Lemma 8.

Proof of Theorem 4. Lemma 7 gives a family $\{G_n\}_{n \in \mathbb{N}}$ of 2^{2^s+s}-divisive graphs, where each G_n has $N = O(2^{\mathsf{CC}_{3,s}(\mathsf{IND}_n)})$ many vertices and $\Omega(2^{\mathsf{CC}_{3,s}(\mathsf{IND}_n)+n}) = \Omega(N^{1+n/\mathsf{CC}_{3,s}(\mathsf{IND}_n)})$ many edges. Then, Lemma 8 ensures that no G_n in this family can contain a copy of $K_{t,t}$, for $t = 3\left(2^{2^s+s}\right)^2$. The Kővári–Sós–Turán theorem guarantees that an N-vertex graph without $K_{t,t}$ as a subgraph must have at most $O(N^{2-1/t})$ many edges [KST54; Zha23], so we must have $n/\mathsf{CC}_{3,s}(\mathsf{IND}_n) \leq 1 - 1/t$ for sufficiently large n. Rearranging, this gives that $\mathsf{CC}_{3,s}(\mathsf{IND}_n) \geq n + \left(\frac{1}{t-1}\right) n$ for all sufficiently large n. ∎

This is not an especially strong lower bound — it seems plausible that $\mathsf{CC}_{3,s}(\mathsf{IND}_n)$ is exponential in n for any constant s, but do not even know that $\mathsf{CC}_{3,s}(\mathsf{IND}_n) \geq 2n$. However, the proof does at least give some evidence that the sort of direct application of Ruzsa–Szemerédi constructions we saw in Section 4 is unlikely to work here — Ruzsa–Szemerédi graphs can contain complete bipartite graphs of any constant size, and we expect that bounds of the form $t \leq n^{1-\Omega(1)}$ for $r = \Theta(n)$ on (r,t)-Ruzsa–Szemerédi graphs, if true, will be difficult to prove [FHS17]. We note also that the graph theoretic property of k-divisiveness, in which any subset of matchings must be k-separable from the rest of the graph, is a strictly stronger property than that of being a Ruzsa–Szemerédi graph, and may be interesting to study in its own right.

6 More Rounds and Connections to Catalytic Computing

We've restricted attention thus far to protocols of only 3 rounds, as the design and analysis of such protocols has proven to already be quite theoretically rich. In this section, we consider what can be said for protocols with more rounds. We observe a close relationship between standard models of nonuniform catalytic computation and our model of catalytic communication, and show that by a communication analogue of Buhrman, Cleve, Koucký, Loff and Speelman's results on CL [Buh+14], there exist constant-round constant-clean protocols with polynomial catalytic space for all of TC^0.

6.1 Amortized Bipartite Branching Programs

The notion of catalytic space introduced by Buhrman, Cleve, Koucký, Loff and Speelman is a uniform model of computation: it consists of a space-bounded algorithm, with an additional resource of catalytic space that must be reset at the end of computation [Buh+14]. One can also define a natural *nonuniform* catalytic model:

Definition 11. An ***amortized branching program***[8] of amortized width w and length ℓ computing m copies of a function f is a directed acyclic multigraph consisting of ℓ layers, where

- The first layer has m vertices, the last layer has 2 vertices, and all other layers have mw vertices.
- Each layer except the last is labeled with an index $i \in [n]$.
- Each vertex except in the last layer has exactly two outgoing edges, labeled 0 and 1 respectively.
- The two vertices in the last layer are labeled "accept" and "reject", and the vertices in the second-to-last layer are labeled with the names of vertices in the first layer, such that each name appears exactly w times.

[8] We remark that our definition is slightly nonstandard, as the last layer is usually defined to have width $2m$ with labels (v, b), where $v \in [m]$ and $b \in \{0, 1\}$. However, the definitions are easily seen to be equivalent (up to a unit change in length).

To compute the function on an input x, the program starts at an arbitrary vertex in the first layer, and then for each layer reads the associated index of x and follows the edge labeled with the resulting value. Correctness of the program entails that, for every input and every starting vertex, the program ends in the vertex with the correct acceptance behaviour, and the second-to-last vertex visited has the same label as the starting vertex.

Amortized branching programs were introduced by Girard, Koucký and McKenzie, who showed examples where direct sum theorems succeeded and failed for the model [GKM15]. Since then, there has been interest in determining the minimum amount of amortization needed to compute a function with small amortized width and length. Potechin showed that every function has an amortized branching program of length $O(n)$ and amortized width $O(1)$ computing 2^{2^n} copies [Pot17]. Robere and Zuiddam showed that this amount of amortization could be reduced for bounded-degree functions over $\mathbb{F}_2$ [RZ22]. Cook and Mertz showed a trade-off in both of these results between length and amortization, allowing them to show in the former case that any function has an amortized branching program of length $O(n)$ and amortized width $O(1)$ computing $2^{2^{\varepsilon n}}$ copies, where the constant in the program length depends on ε [CM22]. In a recent breakthrough work on the tree evaluation problem, Cook and Mertz showed that the ideas of this tradeoff could be strengthened to give length $\mathsf{poly}(n)$, amortized width $O(1)$ branching programs computing $2^{O(n)}$ copies of any function [CM23].

In their work on memoryless communication complexity, Arunachalam and Podder observed that the model was closely related to a notion of ***bipartite*** branching programs [AP21]. We define a similar notion in the amortized sense, which we note exactly describes our model of catalytic communication, and can be seen as an alternate definition.

Definition 12. A ***bipartite amortized branching program*** of amortized width w and length ℓ computing m copies of a function f is a directed acyclic multigraph consisting of ℓ layers, where

- The first layer has m vertices, the last layer has 2 vertices, and all other layers have mw vertices.
- Each layer except the last is labeled either x or y.
- Each vertex except in the last layer has exactly 2^n outgoing edges, labeled with the values of $\{0,1\}^n$.
- The two vertices in the last layer are labeled "accept" and "reject", and the vertices in the second-to-last layer are labeled with the names of vertices in the first layer, such that each name appears exactly w times.

To compute the function on an input (x, y), the program starts at an arbitrary vertex in the first layer, and then for each layer follows the edge labeled with the value of the associated input. Correctness of the program entails that, for every input and every starting vertex, the program ends in the vertex with the correct acceptance behaviour, and the second-to-last vertex visited has the same label as the starting vertex.

The equivalence between such programs and catalytic communication protocols is immediate.

Proposition 4. $\mathsf{CC}_{r,s}(f) \leq c$ if and only if there exists a bipartite amortized branching program of amortized width 2^s and length $r + 2$ computing 2^c copies of f.

Proof. Given such a branching program, we define a catalytic protocol by thinking of the initial catalytic setting as giving the index of a starting vertex of the program, and then on subsequent steps having the blackboard store the index of the current node within its layer. The fact that vertices in the second-to-last layer are labeled by vertices in the first layer means that, with an appropriate ordering of the indexing, the condition of second-to-last vertex having the same label as the starting vertex corresponds exactly to the catalytic portion of the blackboard being reset after r rounds. The final layer corresponds to the output function.

Given a r-round s-clean catalytic protocol with catalytic space c, we can perform the reverse transformation. Each layer except the first and last contains one vertex for each blackboard state, and the transition functions between pairs of layers are determined by how the active player would update the blackboard (with the transition function to the last layer determined by the output function). ■

Note that any amortized branching program is also a bipartite amortized branching program, so a consequence of this equivalence is that any result known for amortized branching programs trivially extends to catalytic communication. For instance, the results of [CM22] and [CM23] give the following, respectively:

Corollary 3. $\mathsf{CC}_{O(n),O(1)}(f) \leq 2^{\varepsilon n}$ for all $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, $\varepsilon > 0$.

Corollary 4. $\mathsf{CC}_{\mathsf{poly}(n),O(1)}(f) \leq O(n)$ for all $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$.

However, there are cases where bipartite amortized branching programs can be much more efficient than amortized branching programs. Note that an amortized branching program of length less than n is uninteresting (it is effectively solving a problem on a smaller input length), whereas we've seen that bipartite amortized branching programs of length 5 can be quite powerful. In the next section, we give a result making use of this power: Buhrman, Cleve, Koucký, Loff and Speelman's CL algorithm for threshold circuit evaluation gives an amortized branching program of length $\mathsf{poly}(n)$ even when the circuit has constant depth, but we note that it can be converted to a bipartite amortized branching program of only constant length.

6.2 Constant-Round Protocols for TC^0

We recall an outline of Buhrman, Cleve, Koucký, Loff and Speelman's results on transparent programs for TC^1.

Definition 13. A ***transparent program*** of length ℓ, with registers $r_1, \ldots, r_m$ over a ring R, on input $x_1, \ldots, x_n \in R$, is a sequence of instructions of the form $r_i \leftarrow r_i + (m_1 \times m_2 \times \cdots \times m_k)$, where the m_i are all either input variables, registers other than r_i, or constants (i.e. elements of R). We say a transparent program ***transparently computes*** a function $f(x_1, \ldots, x_n)$ into a register r_i if, no matter the initial settings $\tau_1, \ldots, \tau_m$ of the registers, the setting of register r_i at the end of the program holds value $\tau_i + f(x_1, \ldots, x_n)$.

Note that a transparent program is inherently reversible: by performing the sequence of instructions in reverse, with $(m_1 \times m_2 \times \cdots \times m_k)$ replaced by $(-1 \times m_1 \times m_2 \times \cdots \times m_k)$ in every instruction, all registers are returned to their initial states. Buhrman, Cleve, Koucký, Loff and Speelman's proof of $\mathsf{TC}^1 \subseteq \mathsf{CL}$ proceeded by an explicit transparent program for computing the majority of several transparently computed values; we present here the transparent subroutines used in that procedure. For explanation of correctness of these algorithms, the reader is referred to [Buh+14]; we present them for the purpose of making structural observations. In each algorithm, r^* is the output register, and all other named registers are unique to the program (i.e. new registers not used in any subroutines).

Algorithm 1 $\texttt{Sum}(r^*, P_1, \ldots, P_k)$

```
for $i = 1, \ldots, k$ do
    $r^* \leftarrow r^* - r_i$
for $i = 1, \ldots, k$ do
    $P_i(r_i)$                ▷ Transparently computes $f_i(x)$ into the $i$th register.
for $i = 1, \ldots, k$ do
    $r^* \leftarrow r^* + r_i$                ▷ The program ends with $\sum_i f_i(x)$ computed into $r^*$.
```

If the programs $P_1, \ldots, P_k$ transparently compute some functions $f_1(x), \ldots, f_k(x)$ respectively, this new program transparently computes the function $\sum_i f_i(x)$, giving us a way to compose transparent operations. The following shows how to compose a transparent program with exponentiation.

Algorithm 2 `Power`(r^*, P, k)

```
for $i = 1, \ldots, k$ do
    $r^* \leftarrow r^* - (-1)^i \binom{k}{i} \times r_i \times r_{k+1}^{k-i}$
$P(r_{k+1})$                                    ▷ Transparently computes $f(x)$ into $r_{k+1}$.
for $i = 1, \ldots, k$ do
    $r_i \leftarrow r_i + r_{k+1}^i$
$r^* \leftarrow r^* + r_{k+1}^k$
$P^{-1}(r_{k+1})$                               ▷ Uncomputes $f(x)$ from $r_{k+1}$, resetting it.
for $i = 1, \ldots, k$ do
    $r^* \leftarrow r^* + (-1)^i \binom{k}{i} \times r_i \times r_{k+1}^{k-i}$    ▷ The program ends with $f(x)^k$ computed into $r^*$.
```

The next algorithm requires the ring R to be the finite field $\mathbb{F}_p$ for some prime $p > s$.

Algorithm 3 `Exact-Value`$(r^*, P_1, \ldots, P_k, s)$

```
Power($r^*$, $r \mapsto$ [Sum($r, P_1, \ldots, P_k$); $r \leftarrow r - s$], $p - 1$)    ▷ The program ends with $(\sum_{i=1}^k f_i(x) - s)^{p-1}$
computed into $r^*$; by Fermat's little theorem, this is the indicator of $\sum_{i=1}^k f_i(x) \neq s$
```

Buhrman, Cleve, Koucký, Loff and Speelman observe that any size-S, depth-d circuit of majority gates has an equivalent size-$2dS^2$, depth $2d$ layered circuit of exact value gates, letting them use this `Exact-Value` procedure to evaluate majority circuits. The number of instructions required is exponential in the depth of the circuit, and polynomial in the size, allowing them to give polynomial-length programs for evaluating any TC^1 circuit. This gives polynomial-length amortized branching programs with amortized width 2 computing $2^{\mathsf{poly}(n)}$ many copies of any TC^1 function family; Buhrman, Cleve, Koucký, Loff and Speelman show additionally that for uniform TC^1 families, the catalytic computation can be done uniformly.

We now show that by an appropriate arrangement of this transparent program, we can get *bipartite* amortized branching programs whose length depends only on the circuit depth, and not on the circuit size. This will imply in particular that any TC^0 function family f_n has a constant-length amortized branching program of width 2 computing $2^{\mathsf{poly}(n)}$ many copies, meaning that for some constant r, $\mathsf{CC}_{r,1}(f_n) \leq \mathsf{poly}(n)$.

Theorem 5. Let $f\colon \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ be any function computable by a size S, depth d majority circuit with arbitrary input preprocessing — that is, there exists a depth-d circuit C composed of S many majority gates, and arbitrary functions g, h, such that $f(x, y) = C(g(x), h(y))$. Then, $\mathsf{CC}_{4^d,1}(f) \leq \mathsf{poly}(S, 2^d)$.

Proof. As shown in [Buh+14], any size-S, depth-d majority circuit has an equivalent size-$\mathsf{poly}(S, d)$, depth-$2d$ exact value circuit. So, it suffices to give a 2^d-round, 1-clean catalytic protocol with $\mathsf{poly}(S, 2^d)$ catalytic space to compute the output of a size-S, depth-d exact value circuit C each of whose inputs depend on only one of x and y. We fix a prime $S < p < 2S$, and make the following claim:

Definition 14. Say that an index i ***belongs*** to x (resp y.) if the largest index $j \leq i$ depending on either x or y depends on x (resp. y). We define the number of ***alternations*** of a transparent program to be the number of indices i such that i and $i+1$ belong to different inputs.

Claim 5. For any gate at height h in C, there exists a transparent program over $\mathbb{F}_p$ computing the output of that gate, such that the first instruction of the program belongs to x, and the program makes at most 2^h many alternations.

Proof of Claim 5. We proceed inductively. Everything at the bottom layer of the circuit depends on only one of x or y, so can be transparently computed with a single alternation by simply adding the appropriate function of x or y. Now, suppose every gate at height $h-1$ can be computed using only 2^{h-1} alternations. Observe that, crucially, the `Exact-Value` algorithm presented never simultaneously requires some input registers to transparently store values while others are in their original state. That is, when the subroutine calls are unfolded, the `Exact-Value` algorithm follows the following sequence of steps:

i) Perform computation independent of x and y.

ii) Transparently compute all of the inputs of the algorithm.

iii) Perform computation independent of x and y.

iv) Transparently uncompute all of the inputs of the algorithm.

v) Perform computation independent of x and y.

So, we can use 2^{h-1} alternations to simultaneously compute all input gates, perform computation that doesn't require alternations, and then perform 2^{h-1} alternations to simultaneously uncompute all of them, giving 2^h many alternations in total. ■

Thus, the overall output of the circuit is computable by a transparent program with at most 2^d many alternations. Note that, as observed by Buhrman, Cleve, Koucký, Loff and Speelman, this program also only requires $\mathsf{poly}(S, 2^d)$ many instructions, and hence at most that many registers. So, the state of all registers used in the program can be maintained with $\mathsf{poly}(S, 2^d)$ many bits.

This transparent program directly gives a catalytic communication protocol (which we can in turn think of as a bipartite amortized branching program). Alice and Bob treat the catalytic portion of the blackboard as a description of the state of all the registers used in this transparent program. At the start of the protocol, Alice records in the clean space the parity of the output register (i.e. the parity of its unique representative in $\{0, \dots, p-1\}$, since registers store elements of $\mathbb{F}_p$, which we can think of as modular equivalence classes). Then, they execute the transparent program one instruction at a time, passing the blackboard at each alternation to ensure that the participant with the blackboard always has access to the appropriate input. Since the program has at most 2^d many alternations, this communication protocol has at most 2^d many rounds. To compute the output, Alice determines the new parity of the output register (where this now means parity of the representative in $\{1, \dots, p\}$, to prevent issues with modular overflow), and accepts if and only if it differs from the clean bit. Correctness of this protocol follows from correctness of the transparent program. ■

7 Conclusion and Open Directions

This work naturally suggests a number of open directions; we mention a few here explicitly as a conclusion.

i) It seems particularly of interest to improve the bounds on protocols for IND_n. There is a line of work in the catalytic literature aiming to understand how much amortization is needed to allow an amortized catalytic branching to compute every function with linear amortized size. There, it is known that $2^{2^{\varepsilon n}}$ amortization suffices for every ε, but we have no nontrivial lower bounds [Pot17; CM20; RZ22; CM22; CM23]. One can view bounding $\mathsf{CC}_{r,s}(\mathsf{IND}_n)$ for constant r as a simplified analogue of this problem: we're still asking to be able to compute any function (now even with the second input arbitrarily large), but instead of considering linear-length amortized branching programs, we're allowing the branching programs to be bipartite but only of constant length. It seems that length-5 bipartite amortized branching programs already capture much of the information-theoretical behaviour of linear-length amortized bipartite branching programs (our Proposition 2 is a direct analogue of Potechin's branching program [Pot17]), but in this setting we actually *were* able to obtain a non-trivial lower bound. It would be interesting to either improve that (rather weak) lower bound, or to find a better upper bound for constant-length bipartite amortized branching programs.

ii) Relatedly, the graph-theoretic notion of k-divisiveness introduced in Section 5 seems potentially interesting in its own right, as a strictly stronger condition than being a Ruzsa–Szemerédi graph[9]. We upper bounded the density of an k-divisive graph by noting that such graphs forbid large complete

[9] A Ruzsa–Szemerédi graph can be decomposed into matchings such that each is induced — or in other words, each matching is 2-separable from the union of all the other matchings. If we knew that our graph was k-divisive, we could divide each constitutent matching into k submatchings, corresponding to the colours in the k-separation of that matching from the rest of the graph. The result would be a Ruzsa–Szemerédi partion with only k times as many constituent matchings. The k-divisiveness condition is much stronger since it requires not just separability of a matching from the rest of the edges, but separability of every set of matchings from every other.

bipartite subgraphs, but it's possible that there exist much stronger ways to exploit the structure of such a graph (and hence obtain stronger lower bounds on $\mathsf{CC}_{3,s}(\mathsf{IND}_n)$). A particularly clean special case to study would be to require each M_i in the matching partition of G to be an individual edge — that is, to require that *every* subset of G's edges be k-separable from the rest of G's edges. It seems plausible that one could get very strong control over what such graphs can look like.

iii) Obtaining lower bounds on protocols of more than three rounds seems difficult — we've shown that constant-round protocols with $\mathsf{poly}(n)$ catalytic space can compute all of TC^0, so to have any hope of showing superpolynomial lower bounds on constant-round catalytic protocols without a breakthrough in circuit complexity one would have to consider a non-explicit function. An intermediate object to consider would be a ***one-way catalytic protocol***, where we mandate that Bob is not allowed to modify the initially-catalytic portion of the blackboard. Recall that our analysis of 3-round protocols involved proving that without loss of generality a 3-round protocol can be made *almost* one-way; this was crucial for the analysis, but seems unlikely to be true for protocols with more rounds (in particular, the protocol in Theorem 5 that allows computing TC^0 does not have this property). One-way catalytic protocols, even of more than 3 rounds, may be tractable to analyze combinatorially — for instance, we suspect that our results on equality protocols in terms of Ruzsa–Szemerédi graphs could be extended to give characterizations of one-way equality protocols with more rounds in terms of larger Steiner systems.

iv) Given the connection between 3-round equality protocols and Ruzsa–Szemerédi graphs outlined in Section 4, one might wonder whether thinking in terms of catalytic communication is useful approach for trying to improve state-of-the-art Ruzsa–Szemerédi constructions. We note that this is perhaps not an implausible idea: there are known constructions of Ruzsa–Szemerédi graphs for certain parameter regimes with natural constructions described by (a different form of) communication protocol [AS20]. Another graph-theoretic question to consider would be whether our notion of "full Ruzsa–Szemerédi graphs" are a stonger condition than standard Ruzsa–Szemerédi graphs, or if perhaps there's a generic conversion.

v) A structural question about this model that remains open is the power of randomness. It has been recently shown that catalytic logspace can be derandomized via compression arguments — one might wonder whether one can use similar approaches to determine how much smaller randomized catalytic communication complexity can be than deterministic catalytic communication complexity.

8 Acknowledgements

We thank Ryan Williams for guidance and suggesting the question of catalytic communication complexity, and Carl Schildkraut for helpful discussion about Ruzsa-Szemerédi graphs.

References

[AP21] Srinivasan Arunachalam and Supartha Podder. "Communication Memento: Memoryless Communication Complexity". In: *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Ed. by James R. Lee. Vol. 185. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 61:1–61:20. ISBN: 978-3-95977-177-1. DOI: 10.4230/LIPIcs.ITCS.2021.61. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2021.61.

[AS20] Noga Alon and Adi Shraibman. "Number on the forehead protocols yielding dense ruzsa–szemerédi graphs and hypergraphs". In: *Acta Mathematica Hungarica* 161.2 (2020), pp. 488–506.

[AS23] Sepehr Assadi and Janani Sundaresan. "Hidden Permutations to the Rescue: Multi-Pass Streaming Lower Bounds for Approximate Matchings". In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. 2023, pp. 909–932. DOI: 10.1109/FOCS57990.2023.00058.

[Beh46] Felix A Behrend. "On sets of integers which contain no three terms in arithmetical progression". In: *Proceedings of the National Academy of Sciences* 32.12 (1946), pp. 331–332.

[BET73] William G Brown, Pál Erdős, and Vera T Sós. "On the existence of triangulated spheres in 3-graphs and related problems". In: *Periodica Mathematica Hungarica* 3 (1973), pp. 221–229.

[BGW20] Mark Braverman, Sumegha Garg, and David P. Woodruff. "The Coin Problem with Applications to Data Streams". In: *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*. Ed. by Sandy Irani. IEEE, 2020, pp. 318–329.

[BGZ21] Mark Braverman, Sumegha Garg, and Or Zamir. "Tight Space Complexity of the Coin Problem". In: *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*. IEEE, 2021, pp. 1068–1079.

[BLM06] Y. Birk, N. Linial, and R. Meshulam. "On the uniform-traffic capacity of single-hop interconnections employing shared directional multichannels". In: *IEEE Trans. Inf. Theor.* 39.1 (2006), pp. 186–191. ISSN: 0018-9448. DOI: 10.1109/18.179355. URL: https://doi.org/10.1109/18.179355.

[Bol86] Béla Bollobás. *Combinatorics: set systems, hypergraphs, families of vectors, and combinatorial probability*. Cambridge University Press, 1986.

[Bra+14] Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. "Pseudorandom Generators for Regular Branching Programs". In: *SIAM J. Comput.* 43.3 (2014), pp. 973–986.

[Bro+13] Joshua E. Brody, Shiteng Chen, Periklis A. Papakonstantinou, Hao Song, and Xiaoming Sun. "Space-Bounded Communication Complexity". In: *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*. ITCS '13. Berkeley, California, USA: Association for Computing Machinery, 2013, pp. 159–172. ISBN: 9781450318594. DOI: 10.1145/2422436.2422456. URL: https://doi.org/10.1145/2422436.2422456.

[Buh+13] Harry Buhrman, Serge Fehr, Christian Schaffner, and Florian Speelman. "The garden-hose model". In: *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*. ITCS '13. Berkeley, California, USA: Association for Computing Machinery, 2013, pp. 145–158. ISBN: 9781450318594. DOI: 10.1145/2422436.2422455. URL: https://doi.org/10.1145/2422436.2422455.

[Buh+14] Harry Buhrman, Richard Cleve, Michal Koucký, Bruno Loff, and Florian Speelman. "Computing with a Full Memory: Catalytic Space". In: STOC '14 (2014), pp. 857–866. DOI: 10.1145/2591796.2591874. URL: https://doi.org/10.1145/2591796.2591874.

[Buh+18] Harry Buhrman, Michal Koucký, Bruno Loff, and Florian Speelman. "Catalytic Space: Nondeterminism and Hierarchy". In: *Theory Comput. Syst.* (2018).

[BV10] Joshua Brody and Elad Verbin. "The Coin Problem and Pseudorandomness for Branching Programs". In: *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*. IEEE Computer Society, 2010, pp. 30–39.

[CM20] James Cook and Ian Mertz. "Catalytic approaches to the tree evaluation problem". In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2020. Chicago, IL, USA: Association for Computing Machinery, 2020, pp. 752–760. ISBN: 9781450369794. DOI: 10.1145/3357713.3384316. URL: https://doi.org/10.1145/3357713.3384316.

[CM21] James Cook and Ian Mertz. "Encodings and the tree evaluation problem". In: *Electron. Colloquium Comput. Complex.* 2021, p. 54.

[CM22] James Cook and Ian Mertz. "Trading time and space in catalytic branching programs". In: *Proceedings of the 37th Computational Complexity Conference*. CCC '22. Philadelphia, Pennsylvania: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2022. ISBN: 9783959772419. DOI: 10.4230/LIPIcs.CCC.2022.8. URL: https://doi.org/10.4230/LIPIcs.CCC.2022.8.

[CM23] James Cook and Ian Mertz. "Tree Evaluation is in Space O(log n · log log n)." In: *Electronic Coloquium Comput. Complex.* TR23 (2023). URL: https://eccc.weizmann.ac.il/report/2023/174/.

[Coh+21] Gil Cohen, Dean Doron, Oren Renard, Ori Sberlo, and Amnon Ta-Shma. "Error Reduction for Weighted PRGs Against Read Once Branching Programs". In: *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*. Ed. by Valentine Kabanets. Vol. 200. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 22:1–22:17.

[Coo+24] James Cook, Jiatu Li, Ian Mertz, and Edward Pyne. "The Structure of Catalytic Space: Capturing Randomness and Time via Compression". In: *Electron. Colloquium Comput. Complex.* TR24-106 (2024). ECCC: TR24-106. URL: https://eccc.weizmann.ac.il/report/2024/106.

[Dul15] Yfke Dulek. *Catalytic space: on reversibility and multiple-access randomness.* 2015.

[FHS17] Jacob Fox, Hao Huang, and Benny Sudakov. "On graphs decomposable into induced matchings of linear sizes". In: *Bulletin of the London Mathematical Society* 49.1 (2017), pp. 45–57.

[Fis+02] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. "Monotonicity testing over general poset domains". In: *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing.* STOC '02. Montreal, Quebec, Canada: Association for Computing Machinery, 2002, pp. 474–483. ISBN: 1581134959. DOI: 10.1145/509907.509977. URL: https://doi.org/10.1145/509907.509977.

[Fox11] Jacob Fox. "A new proof of the graph removal lemma". In: *Annals of Mathematics* (2011), pp. 561–579.

[GKK] Ashish Goel, Michael Kapralov, and Sanjeev Khanna. "On the communication and streaming complexity of maximum bipartite matching". In: *Proceedings of the 2012 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 468–485. DOI: 10.1137/1.9781611973099.41. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611973099.41. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611973099.41.

[GKM15] Vincent Girard, Michal Kouckỳ, and Pierre McKenzie. "Nonuniform catalytic space and the direct sum for space". In: *Electronic Colloquium on Computational Complexity (ECCC).* Vol. 138. 2015.

[Gup+19] Chetan Gupta, Rahul Jain, Vimal Raj Sharma, and Raghunath Tewari. "Unambiguous Catalytic Computation". In: *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019.* Vol. 150. LIPIcs. 2019, 16:1–16:13.

[Har66] L.H. Harper. "Optimal numberings and isoperimetric problems on graphs". In: *Journal of Combinatorial Theory* 1.3 (1966), pp. 385–393. ISSN: 0021-9800. DOI: https://doi.org/10.1016/S0021-9800(66)80059-5. URL: https://www.sciencedirect.com/science/article/pii/S0021980066800595.

[HW03] Johan Håstad and Avi Wigderson. "Simple analysis of graph tests for linearity and PCP". In: *Random Structures & Algorithms* 22.2 (2003), pp. 139–160.

[INW94] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. "Pseudorandomness for network algorithms". In: *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada.* Ed. by Frank Thomson Leighton and Michael T. Goodrich. ACM, 1994, pp. 356–364.

[Kap21] Michael Kapralov. "Space lower bounds for approximating maximum matching in the edge arrival model". In: *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA).* SIAM. 2021, pp. 1874–1893.

[KN21] Christian Konrad and Kheeran K Naidu. "On two-pass streaming algorithms for maximum bipartite matching". In: *arXiv preprint arXiv:2107.07841* (2021).

[KST54] P Kővári, Vera Sós, and Pál Turán. "On a problem of Zarankiewicz". In: *Colloquium Mathematicum.* Vol. 3. Polska Akademia Nauk. 1954, pp. 50–57.

[Lev71] VI Levenšteın. "Upper bounds for codes with a fixed weight of vectors". In: *Problemy Peredaci Informacii* 7 (1971), pp. 3–12.

[LPS19] Nati Linial, Toniann Pitassi, and Adi Shraibman. "On the Communication Complexity of High-Dimensional Permutations". In: *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*. Ed. by Avrim Blum. Vol. 124. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019, 54:1–54:20. ISBN: 978-3-95977-095-8. DOI: 10.4230/LIPIcs.ITCS.2019.54. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2019.54.

[Nis92] Noam Nisan. "Pseudorandom generators for space-bounded computation". In: *Comb.* 12.4 (1992), pp. 449–461.

[Pot17] Aaron Potechin. "A note on amortized branching program complexity". In: *Proceedings of the 32nd Computational Complexity Conference*. 2017, pp. 1–12.

[PSS14] Periklis Papakonstantinou, Dominik Scheder, and Hao Song. "Overlays and Limited Memory Communication". In: *2014 IEEE 29th Conference on Computational Complexity (CCC)*. 2014, pp. 298–308. DOI: 10.1109/CCC.2014.37.

[PV21] Edward Pyne and Salil P. Vadhan. "Pseudodistributions That Beat All Pseudorandom Generators (Extended Abstract)". In: *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*. Ed. by Valentine Kabanets. Vol. 200. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 33:1–33:15.

[Pyn24] Edward Pyne. "Derandomizing Logspace with a Small Shared Hard Drive". In: LIPIcs 300 (2024). Ed. by Rahul Santhanam, 4:1–4:20.

[RS76] I. Ruzsa and E. Szemer'edi. "Triple systems with no six points carrying three triangles". In: *Combinatorica* 18 (Jan. 1976).

[RY20] Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020.

[RZ22] Robert Robere and Jeroen Zuiddam. "Amortized circuit complexity, formal complexity measures, and catalytic algorithms". In: *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2022, pp. 759–769.

[SZG18] Chong Shangguan, Yiwei Zhang, and Gennian Ge. "Centralized Coded Caching Schemes: A Hypergraph Theoretical Approach". In: *IEEE Transactions on Information Theory* 64.8 (2018), pp. 5755–5766. DOI: 10.1109/TIT.2018.2847679.

[Zha23] Yufei Zhao. *Graph Theory and Additive Combinatorics: Exploring Structure and Randomness*. Cambridge University Press, 2023.

A Proof of Results In Introduction

First, catalytic protocols with two or fewer rounds cannot meaningfully make use of the catalytic space.

Proof of Proposition 1. In a one-round protocol, the blackboard is only modified once, so if the catalytic portion is ever changed it will end up in a different state from how it started. Since the catalytic state is an arbitrary string that can't be changed, Alice and Bob could just as well do without it.

In a two-round protocol, Alice and Bob each only get to modify the blackboard once. So, Bob must have a function (independent of x) that lets him recover the original state of the catalytic portion for every message Alice can send. There are 2^{s+c} many possible messages for Alice to send, each of which Bob returns to a single fixed catalytic state – so, by pigeonhole principle, the protocol has some catalytic state that's associated with at most 2^s many Alice messages. This gives a new protocol with no catalytic space: the two parties pretend the catalytic portion started in that state, Alice gives Bob the index of which of the 2^s associated messages she would have wanted to send in the original protocol, and Bob determines what bits he would have sent back in the clean space. ■

Next, we show that every function has a three-round protocol:

Proof of Proposition 2. First, suppose $n_a \le n_b$. In this case, the protocol is as follows:

i) Alice flips the xth bit of the catalytic portion.

ii) Based on his own input, Bob computes the truth table of $x \mapsto f(x, y)$, which he thinks of as a 2^{n_a}-entry bitvector. He takes the inner product of this truth table and the current catalytic space, and writes the result to the clean space.

iii) Alice flips the xth bit of the catalytic portion again, resetting it.

iv) Bob once again takes the inner product of his truth table and the catalytic space, and outputs the XOR of the result and the clean bit.

If, on the other hand, $n_a > n_b$, the roles are reversed: Alice XORs in the truth table of $y \mapsto f(x, y)$, and Bob reads off the yth bit. In either case, this can be seen as an application of the inner product protocol of Proposition 3 to compute the inner product of a truth table and the indicator vector of the index on which it's being evaluated. ■

Finally, we present our protocol for inner product.

Proof of Proposition 3. We will describe a protocol. Let x be Alice's input, y be Bob's input, and τ be the initial state of the catalytic space.

i) For the first round of the protocol, Alice replaces the catalytic portion of the blackboard with $\tau \oplus x$, where $\oplus$ denotes bitwise XOR.

ii) For the second round, Bob computes the inner product of the catalytic portion and his input, writing the resulting bit $\langle \tau \oplus x, y\rangle$ to the clean space.

iii) For the third round, Alice once again bitwise XORs her input into the catalytic portion, resetting it to $(\tau \oplus x) \oplus x = \tau$.

iv) To output the answer, Bob computes the inner product of the new catalytic portion and his input, and adds this to the bit stored in the clean space, resulting in $\langle \tau, y\rangle \oplus \langle \tau \oplus x, y\rangle = \langle x, y\rangle$.

■

B Controlling Information Flow in 3-Round Protocols

In this appendix, we will prove two lemmas demonstrating that any 3-round protocol can be converted into a new form of protocol in which, for a large fraction of inputs and catalytic settings, information flows only one-way from Alice to Bob. We first show that any constant-clean protocol can be turned into a protocol which is one-way at least a constant fraction of the time:

Lemma 6. For any $s > 0$, any function $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ with $\mathsf{CC}_{3,s}(f) = c$ has a protocol of the following form, which we'll call a ***sometimes-one-way-catalytic protocol***:

i) For every $x \in \{0,1\}^{n_a}$, Alice has an injective function $\alpha^{(x)} : \{0,1\}^c \to \{0,1\}^c \times \{0,1\}^{s\cdot(2^s+1)}$.

ii) For every $y \in \{0,1\}^{n_b}$, Bob has two functions, $\beta^{(y)}_{\text{rem}} : \{0,1\}^c \times \{0,1\}^{s\cdot(2^s+1)} \to \{0,1\}^s$ and $\beta^{(y)}_{\text{out}} : \{0,1\}^c \times \{0,1\}^s \to \{0,1\}$.

iii) Call a pair $(x, \tau) \in \{0,1\}^{n_a} \times \{0,1\}^c$ ***bad*** if, for some $y \in \{0,1\}^{n_b}$, we have $\beta^{(y)}_{\text{out}}(\tau, \beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau))) \ne f(x, y)$. Then, at most a $2^s/(2^s + 1)$ fraction of all pairs are bad.

Proof. Fix a catalytic protocol $(A_1, B_2, A_3, B_{\text{out}})$ using s bits of clean space and c bits of catalytic space. For tape settings $\gamma, \gamma' \in \{0,1\}^{c+s}$, we say γ' is ***Bob-reachable*** from γ if Bob maps γ to γ' on some input — that is, there exists some $y \in \{0,1\}^{n_b}$ such that $B_2(y, \gamma) = \gamma'$. Define the ***talkativity*** of γ, denoted $T(\gamma)$, to be the number of γ' that are Bob-reachable from γ.

Now, consider a modified version of the model where, instead of Bob modifying the tape, he's simply allowed to respond to Alice's message of γ with any number between 1 and $T(\gamma)$. They can still simulate the same protocol, since the set of Bob-reachable states from γ doesn't depend on x or y, and so Bob can just send Alice the index of $B_2(y,\gamma)$ in that set.

Alice must always reset the catalytic portion of the blackboard, so this message from Bob can only affect what she writes to the clean portion. For every value of Bob's message, she will write s bits to the clean portion — this behaviour is describable by $s \cdot T(\gamma)$ many bits. If she sent that information to Bob along with the output of A_1, then Bob could compute himself what value would end up written to the clean space.

Whenever $T(A_1(x,\tau,0^s)) \leq 2^s$, we'll let $\alpha^{(x)}(\tau)$ be $A_1(x,\tau,0^s)$ concatenated those with $s \cdot T(\gamma)$ many bits, and we will expect (x,τ) to be good. If the talkativity is larger, we'll just choose an arbitrary (maintaining injectivity) value for $\alpha^{(x)}(\tau)$, accepting that (τ, x) may be bad. Bob can compute $\beta^{(y)}_{\text{rem}}(\gamma, g)$ by determining, as described above, the s clean bits that Alice would end up writing to the clean space if they ran the original protocol. (Note that he will only necessarily correctly determine this if Alice was able to send her entire $s \cdot T(\gamma)$-bit function; i.e., if $T(\gamma) \leq 2^s$.) Finally, $\beta^{(y)}_{\text{out}}(\gamma)$ will simply be $B_{\text{out}}(y,\gamma)$.

Injectivity of $\alpha^{(x)}$ follows from correctness of the catalytic protocol: if $\alpha^{(x)}$ maps two distinct τ to the same output, the original protocol must reset both to the same final blackboard state. Since whenever $T(A_1(x,\tau,0^s)) \leq 2^s$, this new protocol agrees with the old, no such (x,τ) are bad. So it now suffices to show that at least a $1/(s+1)$ fraction of all (x,τ) pairs have $T(A_1(x,\tau,0^s)) \leq 2^s$.

Suppose that, for some x, we have $\sum_{\tau\in\{0,1\}^c} T(A_1(x,\tau,0^s)) > 2^{c+s}$. Then, by pigeonhole principle, there must be two distinct catalytic tape settings, $\tau \neq \tau'$, such that the sets of states Bob-reachable from $A_1(x,\tau,0^s)$ and $A_1(x,\tau',0^s)$ have non-empty intersection. That is, there exist $y, y' \in \{0,1\}^{n_b}$ such that $B_2(y, A_1(x,\tau,0^s)) = B_2(y', A_1(x,\tau',0^s))$. But this contradicts correctness of the catalytic protocol, because this ensures that inputs of (x,y,τ) and (x,y',τ') to the protocol will both result in the same final blackboard state, and hence at least one will fail to have catalytic portion reset correctly. Hence, for all x, $\sum_{\tau\in\{0,1\}^c} T(A_1(x,\tau,0^s)) \leq 2^{c+s}$. So, $\sum_{x,\tau\in\{0,1\}^{n_a}\times\{0,1\}^c} T(A_1(x,\tau,0^s)) \leq 2^{c+s+n_a}$. Since we always have $T(A_1(x,\tau,0^s)) \geq 0$, the number of (x,τ) such that $T(A_1(x,\tau,0^s)) \geq 2^s + 1$ can be at most $2^{c+s+n_a}/(2^s+1)$. ■

We now give a strengthening of Lemma 6 in the case $s = 1$, showing that 1-clean protocols can be converted into protocols that are one-way *almost all* of the time:

Lemma 2. Every left-injective function $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ with $\mathsf{CC}_{3,1}(f) = c$ has a protocol of the following form, which we'll call a ***mostly-one-way-catalytic protocol***:

i) For every $x \in \{0,1\}^{n_a}$, Alice has an injective function $\alpha^{(x)} : \{0,1\}^c \to \{0,1\}^c \times \{0,1\}^3$.

ii) For every $y \in \{0,1\}^{n_b}$, Bob has two functions, $\beta^{(y)}_{\text{rem}} : \{0,1\}^c \times \{0,1\}^3 \to \{0,1\}$ and $\beta^{(y)}_{\text{out}} : \{0,1\}^c \times \{0,1\} \to \{0,1\}$.

iii) Call a pair $(x,\tau) \in \{0,1\}^{n_a} \times \{0,1\}^c$ ***bad*** if, for some $y \in \{0,1\}^{n_b}$, we have $\beta^{(y)}_{\text{out}}(\tau, \beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau))) \neq f(x,y)$. Then, at most 2^{c+1} many pairs are bad.

Additionally, we may assume that, for every τ, y, we have $\beta^{(y)}_{\text{out}}(\tau,0) \neq \beta^{(y)}_{\text{out}}(\tau,1)$.

Proof. The proof begins identically to that of Lemma 6, noting that $1 \cdot (2^1 + 1) = 3$. For the main claim, it now suffices to show that, when $s = 1$, we can have $T(A_1(x,\tau,0)) > 2$ for at most 2^{c+1} many pairs (x,τ).

As before, we have have $\sum_{x,\tau\in\{0,1\}^{n_a}\times\{0,1\}^c} T(A_1(x,\tau,0^s)) \leq 2^{c+n_a+s} = 2^{c+n_a+1}$. Call a pair (x,τ) ***choiceless*** if $T(A_1(x,\tau,0)) = 1$. Since we always have $T(A_1(x,\tau,0)) \geq 1$, this implies that the number of choiceless (x,τ) is at least as large as the number of (x,τ) with $T(A_1(x,\tau,0)) > 2$. So, it suffices to show that there are at most 2^{c+1} many choiceless pairs. If there were more than 2^{c+1} many choiceless pairs, then by pigeonhole principle there would have to be at least three choiceless pairs using the same value of τ. Again, by pigeonhole principle, two of these three pairs must end the protocol with the same bit in the clean space — that is, $A_3(x, B_2(y, A_1(x,\tau))) = A_3(x', B_2(y, A_1(x',\tau)))$ for some $x \neq x'$ and (noting that $T(A_1(x,\tau)) = 1$ means that $B_2(\cdot, A_1(x,\tau))$ is constant) for all y. But then, this means

$f(x,y) = B_{\text{out}}(y, A_3(x, B_2(y, A_1(x,\tau)))) = B_{\text{out}}(y, A_3(x', B_2(y, A_1(x',\tau)))) = f(x',y)$ for all y, contradicting the assumption that f is left-injective.

We now show that this protocol can be modified to ensure that $\beta^{(y)}_{\text{out}}(\tau,0) \neq \beta^{(y)}_{\text{out}}(\tau,1)$ for all τ, y. Fix some y, and first suppose that $f(\cdot,y)$ is constant. In this case, we can modify the protocol to let $\beta^{(y)}_{\text{rem}}$ output that same constant for every τ, and let $\beta^{(y)}_{\text{out}}$ simply output the remembered bit, and our protocol will remain correct for that y and every x. Now, on the other hand, suppose that $f(\cdot,y)$ is not constant — in this case, we claim that the protocol must already have $\beta^{(y)}_{\text{out}}(\tau,0) \neq \beta^{(y)}_{\text{out}}(\tau,1)$ for all τ. If there were some τ such that $\beta^{(y)}_{\text{out}}(\tau,0) = \beta^{(y)}_{\text{out}}(\tau,1)$, by definition we would have $B_{\text{out}}(y,\tau,0) = B_{\text{out}}(y,\tau,1)$. But then, the original catalytic protocol, when the catalytic tape is initialized to τ and Bob is given input y, must always output the same value — contradicting the assumption that $f(\cdot,y)$ is non-constant. ■