Catalytic Communication

Edward Pyne ✉ 🆔
MIT, Cambridge, MA, USA

Nathan S. Sheffield ✉ 🏠 🆔
MIT, Cambridge, MA, USA

William Wang ✉
MIT, Cambridge, MA, USA

────── **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:

1. 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.

2. 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.

3. 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 [36, 31] 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 [9, 40, 16]. Other works have tightly characterized the space required to solve fundamental problems such as estimating the bias of a coin [10, 7, 8], 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 [13], 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 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 $O(\log n \log \log n)$, contradicting long-standing prior beliefs [18, 19, 21]. Due to the striking power of this model, there have been many followup papers studying its structure [22, 15, 28, 39]. In the other direction, [17] 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,*

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

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

---

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

...

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

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

(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 $A_{out} : \{0,1\}^{n_a} \times \{0,1\}^c \times \{0,1\}^s \to \{0,1\}$.) 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

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

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 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 Section 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 3.** *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 4.** *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 4 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 5.** *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 [41]. 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 6.** *For any* $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$,

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

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

The proof of the upper bound of Theorem 6 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 $O(1)$ [41].

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:

---

[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 7.** *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 8.** *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 9.** *We have* $\mathsf{CC}_{3,s}(\mathsf{IND}_n) \geq (1+\varepsilon)n$, *for some constant ε depending on s.*

While we suspect Theorem 9 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:

▶ **Theorem 10.** *Let $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ 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 [2]. 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 [11]. 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} [37]. 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 [2]. 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 [11] and Papakonstantinou, Scheder and Song [37], as well as the "garden hose" model of Buhrman, Fehr, Schaffner and Speelman [14]. 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 11.** *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 12.** *For any n, we denote by EQ_n the equality function on n-bit inputs. That is,*

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

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

▶ **Definition 13.** *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 14.** *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 $\wr a_1, \ldots, a_n \wr$.

3 Characterizing 3-Round 1-Clean Complexity

In this section, we prove Theorem 6.

▶ **Theorem 6.** *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 15.** *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:

1. 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 $\tau \oplus 1_{i \in T}$.
2. Bob computes $\bigoplus_{i \in [\mathrm{rk}\, f]} \tau_i v_i$, and writes the yth entry of the result to the clean portion.
3. Alice XORs out the same string she XORed in, resetting the catalytic portion.
4. 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 15 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 the full version, 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 16.** *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**:*
1. *For every* $x \in \{0,1\}^{n_a}$, *Alice has an injective function* $\alpha^{(x)} : \{0,1\}^c \to \{0,1\}^c \times \{0,1\}^3$.
2. *For every* $y \in \{0,1\}^{n_b}$, *Bob has two functions,* $\beta_{rem}^{(y)} : \{0,1\}^c \times \{0,1\}^3 \to \{0,1\}$ *and* $\beta_{out}^{(y)} : \{0,1\}^c \times \{0,1\} \to \{0,1\}$.
3. *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*

$$\beta_{out}^{(y)}(\tau, \beta_{rem}^{(y)}(\alpha^{(x)}(\tau))) \neq f(x,y).$$

Then, at most 2^{c+1} *many pairs are bad.*

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

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 6. The upper bound was shown in Lemma 15; 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 $\mathrm{rk}(f) \geq 1000$. We fix an arbitrary left-injective $f \colon \{0,1\}^{n_a} \times \{0,1\}^{n_b}$ with $\mathrm{rk}(f) \geq 1000$, and assume for contradiction that $\mathsf{CC}_{3,1}(f) < \mathrm{rk}(f)/6$.

By Lemma 16, we have a mostly-one-way-catalytic protocol for f with $c < \mathrm{rk}(f)/6$. 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 $y \mapsto \beta_{rem}^{(y)}(\gamma)$, and let $U = \langle u_\gamma \colon \gamma \in \{0,1\}^{c+3} \rangle$. For every $\tau \in \{0,1\}^c$, let $v_\tau \in \{0,1\}^{n_b}$ be the truth table of $y \mapsto \beta_{out}^{(y)}(\tau, 0)$, and let $V = \langle v_\tau \colon \tau \in \{0,1\}^c \rangle$. Finally, for every bad $(x, \tau) \in \{0,1\}^{n_a} \times \{0,1\}^c$, let $w_{(x,\tau)}$ be the truth table of $y \mapsto f(x,y) \oplus \beta_{out}^{(y)}(\tau, 0)$, and let $W = \langle w_{(x,\tau)} \colon (x, \tau)$ is bad$\rangle$.

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 17.** 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 17. Because v has at least k blue pebbles, there are at least k distinct initial catalytic configurations $\tau_1, \ldots, \tau_k \in \{0, 1\}^c$ 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 $\alpha^{(x)}(\tau_1), \ldots, \alpha^{(x)}(\tau_k)$ 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 $y \mapsto \beta_{\text{out}}^{(y)}(\tau_i, 0)$, and that $u_{\alpha^{(x)}(\tau_i)}$ is the truth table of $y \mapsto \beta_{\text{rem}}^{(y)}(\alpha^{(x)}(\tau_i))$. So, to conclude that $r = v \oplus u_{\alpha^{(x)}(\tau_i)}$, we need to show that, for all y, we have $f(x, y) = \beta_{\text{out}}^{(y)}(\tau_i, 0) \oplus \beta_{\text{rem}}^{(y)}(\alpha^{(x)}(\tau_i))$.

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

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 17 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 18.** 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 18. 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 $|N(S) \setminus S| \geq \binom{\text{rk}(f)}{\ell+1}$ [29, 6]. 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 $\min_{S: |S|=s} \left(\frac{|N(S)|}{|S|} \right)$ is monotonically nonincreasing in s for small s[3], the surface-area-to-volume ratio of this Hamming sphere lower bounds that of S:

---

[3] As long as s is small enough that a Hamming ball of that size can't have vertex boundary less than

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

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

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

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 $\mathrm{rk}(f)H\left(1/20-1/\mathrm{rk}(f)\right) - \log\left(\mathrm{rk}(f)\right) > \frac{\mathrm{rk}(f)}{6}$, so this implies that $|S| > 2^{\left(\mathrm{rk}(f)/6\right)}$.   ◁

We can't immediately a derive contradiction from Claim 17 and Claim 18, because V is a multiset (i.e. some vertices may have more than one red pebble). However, this is not a serious obstacle.

▷ **Claim 19.**   Suppose a pebbling satisfies the conditions of Claim 17 – 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 18 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 17 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 6 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

---

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.

be communicated with only n bits of catalytic space, despite requiring (up to constant factors) maximal standard communication complexity even with randomness [41]. 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 7.** *We have that* $\mathsf{CC}_{3,2}(\mathsf{EQ}_n) \leq O(n \log n)$.

The key to the proof of Theorem 7 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 demonstrate how any such construction can be generically converted into a catalytic protocol for equality, which will in particular prove Theorem 7.

In Section 4.3 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 20.** *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 [12], 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 [4] 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 $\Omega(n)$ [43]. The best known upper bounds still go through graph regularity: Fox's improved bounds for triangle removal imply that if $r = \Theta(n)$, then $t \leq \frac{n}{2^{\Omega(\log^*(n))}}$ [24].

Ruzsa–Szemerédi graphs have since seen several applications in computational and communication complexity [30, 5, 35]. 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 [23, 27, 33, 32, 3]. 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 [23].

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

▶ **Definition 21.** *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 show in the full version that it can easily be made so:

▶ **Lemma 22.** *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.*

4.2 Equality Protocols from Full Ruzsa–Szemerédi Graphs

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

▶ **Lemma 23.** *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 7 given Lemma 23. For some constant $0 < \delta < 1$, by Lemma 22, there exists a $(3, N^{\frac{\delta}{\log\log N}})$-full Ruzsa–Szemerédi graph on N vertices. Plugging this in to Lemma 23, 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]. ◀

Proof of Lemma 23. 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

---

[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.

over j of the edges induced by $M_i^{(j)}$. Fix an arbitrary bijection $\pi \colon \{0,1\}^c \to \{v_1, \dots, v_N\}$, and arbitrary injections $\sigma \colon \{0,1\}^{\lfloor \log(t) \rfloor} \to \{M_1, \dots, M_t\}$, $\mu \colon [k] \to \{0,1\}^{\lceil \log k \rceil}$. The protocol is as follows.

1. $A_1(x, \tau, \omega) = (\pi^{-1}(v), \omega)$, where v is the neighbour of $\pi(\tau)$ in the matching $\sigma(x)$.
2. $B_2(y, \tau, \omega) = (\tau, \mu(w))$, where w is the unique value such that $\pi(\tau) \in \sigma(y)^{(w)}$.
3. $A_3(x, \tau, \omega) = (\pi^{-1}(v), \omega)$, where v is the neighbour of $\pi(\tau)$ in the matching $\sigma(x)$.
4. $B_{\mathrm{out}}(y, \tau, \omega)$ 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 $\sigma(x)^{(1)}, \dots, \sigma(x)^{(k)}$, 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 $\sigma(y)$. ◀

4.3 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 23 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 24.** *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 25.** *For any $s > 0$, any function $f \colon \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ with $\mathsf{CC}_{3,s}(f) = c$ has a protocol of the following form, which we'll call a **sometimes-one-way-catalytic protocol**:*

1. *For every $x \in \{0,1\}^{n_a}$, Alice has an injective function $\alpha^{(x)} \colon \{0,1\}^c \to \{0,1\}^c \times \{0,1\}^{s \cdot (2^s + 1)}$.*
2. *For every $y \in \{0,1\}^{n_b}$, Bob has two functions, $\beta_{rem}^{(y)} \colon \{0,1\}^c \times \{0,1\}^{s \cdot (2^s + 1)} \to \{0,1\}^s$ and $\beta_{out}^{(y)} \colon \{0,1\}^c \times \{0,1\}^s \to \{0,1\}$.*
3. *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 $\beta_{out}^{(y)}(\tau, \beta_{rem}^{(y)}(\alpha^{(x)}(\tau))) \neq f(x, y)$. Then, at most a $2^s/(2^s + 1)$ fraction of all pairs are bad.*

---

[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.

Proof of Lemma 24. By Lemma 25, 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 $\beta_{\mathrm{rem}}^{(z)}(\alpha^{(z)}(\tau)) = \omega$. 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 $\beta_{\mathrm{rem}}^{(z)}(\gamma) = \omega$.

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 26.** *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 $t \leq \frac{N}{2^{\Omega(\log^*(N))}}$ [24]. Since, by Lemma 24, 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 27.** *If, for some constant s, $\mathsf{CC}_{3,s}(\mathsf{EQ}_n) \leq O(n)$, then*
1. *Any non-adaptive tester for monotonicity over general N-element posets requires query complexity $\Omega(N^c)$ for some $c > 0$.*

2. *Any randomized semi-streaming algorithm for $(1 - \varepsilon)$-approximate maximum bipartite matching requires $\Omega(\log(1/\varepsilon))$ passes over the stream.*
3. *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 24, 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 [23], the streaming lower bounds of Assadi and Sundaresan [3], and the centralized coding caching scheme construction of Shangguan, Zhang and Ge [44] 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 27 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 [35, 1]. 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 13.** *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 25 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.

---

[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.

By Proposition 4, 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 9.** *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 28.** *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 29.** *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 30.** *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.*

Proof. As in Section 4.3, we begin by invoking Lemma 25 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 $M_x \subseteq E(G_n)$, $M_x = \{(\tau, \alpha^{(x)}(\tau)) : (x, \tau) \text{ is not bad}\}$. 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 (2^{2^s+s})-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 $\left(\beta_{\mathrm{out}}^{(1_S)}(\tau, \cdot), \beta_{\mathrm{rem}}^{(1_S)}(\gamma)\right) \in (\{0,1\}^s \to \{0,1\}) \times \{0,1\}^s$. If (τ, γ) and (τ', γ') have the same colour, it means that $\beta_{\mathrm{out}}^{(1_S)}(\tau, \cdot)$ and $\beta_{\mathrm{out}}^{(1_S)}(\tau', \cdot)$ are the same function, and $\beta_{\mathrm{rem}}^{(1_S)}(\gamma)$ and $\beta_{\mathrm{rem}}^{(1_S)}(\gamma')$ 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,

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

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 31.** *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 32.** 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 32. 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 9 now follows directly from Lemma 30 and Lemma 31.

Proof of Theorem 9. Lemma 30 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 31 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 [34, 45], 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 [25]. 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 [13], 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 [13]. One can also define a natural *nonuniform* catalytic model:

▶ **Definition 33.** *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.*

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 [26]. 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

---

[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).

function has an amortized branching program of length $O(n)$ and amortized width $O(1)$ computing 2^{2^n} copies [38]. Robere and Zuiddam showed that this amount of amortization could be reduced for bounded-degree functions over $\mathbb{F}_2$ [42]. 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 [20]. 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 [21].

In their work on memoryless communication complexity, Arunachalam and Podder observed that the model was closely related to a notion of **bipartite** branching programs [2]. 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 34.** *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 35.** $\mathsf{CC}_{r,s}(f) \leq c$ *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 [20] and [21] give the following, respectively:

▶ **Corollary 36.** $\mathsf{CC}_{O(n),O(1)}(f) \leq 2^{\varepsilon n}$ *for all* $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, $\varepsilon > 0$.

▶ **Corollary 37.** $\mathsf{CC}_{\mathsf{poly}(n),O(1)}(f) \leq O(n)$ *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 full version, 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 by appropriate rearranging it can be converted to a bipartite amortized branching program of only constant length, yielding the following:

▶ **Theorem 10.** *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,* $\mathsf{CC}_{4^d,1}(f) \leq \mathsf{poly}(S, 2^d)$.

References

1   Noga Alon and Adi Shraibman. Number on the forehead protocols yielding dense ruzsa–szemerédi graphs and hypergraphs. *Acta Mathematica Hungarica*, 161(2):488–506, 2020.

2   Srinivasan Arunachalam and Supartha Podder. Communication Memento: Memoryless Communication Complexity. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 61:1–61:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ITCS.2021.61`.

3   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)*, pages 909–932, 2023. `doi:10.1109/FOCS57990.2023.00058`.

4   Felix A Behrend. On sets of integers which contain no three terms in arithmetical progression. *Proceedings of the National Academy of Sciences*, 32(12):331–332, 1946.

5   Y. Birk, N. Linial, and R. Meshulam. On the uniform-traffic capacity of single-hop interconnections employing shared directional multichannels. *IEEE Trans. Inf. Theor.*, 39(1):186–191, 2006. `doi:10.1109/18.179355`.

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

7   Mark Braverman, Sumegha Garg, and David P. Woodruff. The coin problem with applications to data streams. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 318–329. IEEE, 2020. `doi:10.1109/FOCS46700.2020.00038`.

8   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*, pages 1068–1079. IEEE, 2021. `doi:10.1109/FOCS52979.2021.00106`.

9   Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. Pseudorandom generators for regular branching programs. *SIAM J. Comput.*, 43(3):973–986, 2014. `doi:10.1137/120875673`.

10   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*, pages 30–39. IEEE Computer Society, 2010. `doi:10.1109/FOCS.2010.10`.

11   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, pages 159–172, New York, NY, USA, 2013. Association for Computing Machinery. `doi:10.1145/2422436.2422456`.

12   William G Brown, Pál Erdős, and Vera T Sós. On the existence of triangulated spheres in 3-graphs and related problems. *Periodica Mathematica Hungarica*, 3:221–229, 1973.

13   Harry Buhrman, Richard Cleve, Michal Koucký, Bruno Loff, and Florian Speelman. Computing with a full memory: catalytic space. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 857–866. ACM, 2014. `doi:10.1145/2591796.2591874`.

14   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, pages 145–158, New York, NY, USA, 2013. Association for Computing Machinery. `doi:10.1145/2422436.2422455`.

15   Harry Buhrman, Michal Koucký, Bruno Loff, and Florian Speelman. Catalytic space: Non-determinism and hierarchy. *Theory Comput. Syst.*, 2018. `doi:10.1007/S00224-017-9784-7`.

16   Gil Cohen, Dean Doron, Oren Renard, Ori Sberlo, and Amnon Ta-Shma. Error reduction for weighted prgs against read once branching programs. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 22:1–22:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPICS.CCC.2021.22`.

17   James Cook, Jiatu Li, Ian Mertz, and Edward Pyne. The structure of catalytic space: Capturing randomness and time via compression. *Electron. Colloquium Comput. Complex.*, TR24-106, 2024. URL: `https://eccc.weizmann.ac.il/report/2024/106`, `arXiv:TR24-106`.

18   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, pages 752–760, New York, NY, USA, 2020. Association for Computing Machinery. `doi:10.1145/3357713.3384316`.

19   James Cook and Ian Mertz. Encodings and the tree evaluation problem. In *Electron. Colloquium Comput. Complex*, page 54, 2021.

20   James Cook and Ian Mertz. Trading time and space in catalytic branching programs. In *Proceedings of the 37th Computational Complexity Conference*, CCC '22, Dagstuhl, DEU, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.CCC.2022.8`.

21   James Cook and Ian Mertz. Tree evaluation is in space o(log n · log log n). *Electronic Coloquium Comput. Complex.*, TR23, 2023. URL: `https://eccc.weizmann.ac.il/report/2023/174/`.

22   Yfke Dulek. Catalytic space: on reversibility and multiple-access randomness, 2015.

23   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, pages 474–483, New York, NY, USA, 2002. Association for Computing Machinery. `doi:10.1145/509907.509977`.

24   Jacob Fox. A new proof of the graph removal lemma. *Annals of Mathematics*, pages 561–579, 2011.

25   Jacob Fox, Hao Huang, and Benny Sudakov. On graphs decomposable into induced matchings of linear sizes. *Bulletin of the London Mathematical Society*, 49(1):45–57, 2017.

26 Vincent Girard, Michal Koucký, and Pierre McKenzie. Nonuniform catalytic space and the direct sum for space. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 138, 2015.

27 Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 468–485. SIAM, 2012. `doi:10.1137/1.9781611973099.41`.

28 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*, volume 150 of *LIPIcs*, pages 16:1–16:13, 2019. `doi:10.4230/LIPICS.FSTTCS.2019.16`.

29 L.H. Harper. Optimal numberings and isoperimetric problems on graphs. *Journal of Combinatorial Theory*, 1(3):385–393, 1966. `doi:10.1016/S0021-9800(66)80059-5`.

30 Johan Håstad and Avi Wigderson. Simple analysis of graph tests for linearity and pcp. *Random Structures & Algorithms*, 22(2):139–160, 2003. `doi:10.1002/RSA.10068`.

31 Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In Frank Thomson Leighton and Michael T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 356–364. ACM, 1994. `doi:10.1145/195058.195190`.

32 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)*, pages 1874–1893. SIAM, 2021. `doi:10.1137/1.9781611976465.112`.

33 Christian Konrad and Kheeran K Naidu. On two-pass streaming algorithms for maximum bipartite matching. *arXiv preprint*, 2021. `arXiv:2107.07841`.

34 P Kővári, Vera Sós, and Pál Turán. On a problem of zarankiewicz. In *Colloquium Mathematicum*, volume 3, pages 50–57. Polska Akademia Nauk, 1954.

35 Nati Linial, Toniann Pitassi, and Adi Shraibman. On the Communication Complexity of High-Dimensional Permutations. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 54:1–54:20, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ITCS.2019.54`.

36 Noam Nisan. Pseudorandom generators for space-bounded computation. *Comb.*, 12(4):449–461, 1992. `doi:10.1007/BF01305237`.

37 Periklis Papakonstantinou, Dominik Scheder, and Hao Song. Overlays and limited memory communication. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 298–308, 2014. `doi:10.1109/CCC.2014.37`.

38 Aaron Potechin. A note on amortized branching program complexity. In *Proceedings of the 32nd Computational Complexity Conference*, pages 1–12, 2017. `doi:10.4230/LIPICS.CCC.2017.4`.

39 Edward Pyne. Derandomizing logspace with a small shared hard drive. In Rahul Santhanam, editor, *39th Computational Complexity Conference, CCC 2024, July 22-25, 2024, Ann Arbor, MI, USA*, volume 300 of *LIPIcs*, pages 4:1–4:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. `doi:10.4230/LIPICS.CCC.2024.4`.

40 Edward Pyne and Salil P. Vadhan. Pseudodistributions that beat all pseudorandom generators (extended abstract). In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 33:1–33:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPICS.CCC.2021.33`.

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

42 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)*, pages 759–769. IEEE, 2022. `doi:10.1109/FOCS52979.2021.00079`.

43   I. Ruzsa and E. Szemer'edi. Triple systems with no six points carrying three triangles. *Combinatorica*, 18, January 1976.

44   Chong Shangguan, Yiwei Zhang, and Gennian Ge. Centralized coded caching schemes: A hypergraph theoretical approach. *IEEE Transactions on Information Theory*, 64(8):5755–5766, 2018. `doi:10.1109/TIT.2018.2847679`.

45   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 3. 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 4. First, suppose $n_a \leq n_b$. In this case, the protocol is as follows:
1. Alice flips the xth bit of the catalytic portion.
2. 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.
3. Alice flips the xth bit of the catalytic portion again, resetting it.
4. 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 5 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 5. 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.
1. For the first round of the protocol, Alice replaces the catalytic portion of the blackboard with $\tau \oplus x$, where $\oplus$ denotes bitwise XOR.
2. 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.

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

4. 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 $\langle \tau, y \rangle \oplus \langle \tau \oplus x, y \rangle = \langle x, y \rangle$. ◀