Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 73 (2013)

On the Communication Complexity Methodology for Proving Lower Bounds on the Query Complexity of Property Testing

Oded Goldreich
Department of Computer Science
Weizmann Institute of Science
oded.goldreich@weizmann.ac.il

June 8, 2013

Abstract

A couple of years ago, Blais, Brody, and Matulef put forward a methodology for proving lower bounds on the query complexity of property testing via communication complexity. They provided a restricted formulation of their methodology (via "simple combining operators") and also hinted towards a more general formulation, which we spell out in this paper.

A special case of the general formulation proceeds as follows: In order to derive a lower bound on testing the property Π, one presents a mapping F of pairs of inputs $(x,y) \in \{0,1\}^{n+n}$ for a two-party communication problem Ψ to $\ell(n)$-bit long inputs for Π such that $(x,y) \in \Psi$ implies $F(x,y) \in \Pi$ and $(x,y) \notin \Psi$ implies that $F(x,y)$ is far from Π. Let $f_i(x,y)$ be the $i^{\rm th}$ bit of $F(x,y)$, and suppose that B is an upper bound on the (deterministic) communication complexity of each f_i and that C is a lower bound on the randomized communication complexity of Ψ. Then, testing Π requires at least C/B queries.

The foregoing formulation is generalized by considering randomized protocols (with small error) for computing the f_i's. In contrast, the restricted formulation (via "simple combining operators") requires that each $f_i(x,y)$ be a function of x_i and y_i only, and uses $B = 2$ for the straightforward computation of f_i.

We show that the general formulation cannot yield significantly stronger lower bounds than those that can be obtained by the restricted formulation. Nevertheless, we advocate the use of the general formulation, because we believe that it is easier to work with. Following Blais *et al.*, we also describe a version of the methodology for nonadaptive testers and one-way communication complexity.

Keywords: Property Testing, Communication Complexity, Locally Testable Codes, Locally Decodable Codes, Nonadaptive queries.

ISSN 1433-8092

Contents

1 Introduction

In the last couple of decades, the area of property testing has attracted much attention (see, e.g., [7, 19, 20]). Loosely speaking, property testing typically refers to sub-linear time probabilistic algorithms for deciding whether a given object has a predetermined property or is far from any object having this property. Such algorithms, called testers, obtain local views of the object by performing queries; that is, the object is seen as a function and the testers get oracle access to this function (and thus may be expected to work in time that is sub-linear in the length of the object).

A couple of years ago, Blais, Brody, and Matulef enriched the study of property testing by presenting a connection between property testing and communication complexity [4]. Specifically, they presented a methodology for proving lower bounds on the query complexity of property testing by relying on lower bounds on two-party communication complexity problems.

Encountering their work [4], we were quite surprised. Firstly, this connection seems unexpected, since property testing problems have no topology that can be naturally 2-partitioned to fit the two-party setting of communication complexity. Nevertheless, using this methodology, the authors of [4] were able to resolve a fair number of open problems, some of which have escaped our own attempts in the past (cf., e.g., [4, Thms. 1.1-1.3], which resolve open problems in [9]).

While Blais, Brody, and Matulef hint towards the formulation that we will present here (see a few lines before [4, Def. 2.3]), they preferred to present a more restricted formulation, which is pivoted at "simple combining operators" (see [4, Def. 2.3]). Furthermore, it seems that this restricted formulation is the one that is disseminating in the literature. The main purpose of this paper is to explicitly present a more general and flexible formulation, and demonstrate the ease of using it (in comparison to the use of the restricted formulation).

We also show that the restricted formulation is actually not significantly weaker than the general one. In other words, we show that any lower bound that can be derived by the general formulation, can also be derived (possibly with a small quantitative loss) by the restricted formulation. Nevertheless, we advocate the use of the general formulation, because we believe that it is easier to work with. This is demonstrated by using it to derive some (known and new) results regarding the hardness of codeword testing for some codes. Furthermore, we believe that the statement of the general formulation and its proof reveals better what is actually going on.

Organization. After recalling the standard definitions (in Section 2), we provide a general formulation of the communication complexity methodology for proving query complexity lower bounds on property testing (see Theorem 3.1 in Section 3). The (relative) ease of using this methodology is demonstrated in Section 4, and further discussed in the conclusion section. The version for nonadaptive testers and one-way communication complexity is presented in Section 5, and other ramifications are discussed in Section 6. In Section 7, we consider the relation between the general formulation and the restricted one (as presented in [4]). Indeed, Theorem 7.1 (i.e., the "emulation theorem") is the main technical contribution of this work.

We augment the current version with a suggestion by David Woodruff, which relates nonadaptive testers and simultaneous communication (see Theorem 5.2). This led us to further generalize the methodology (for general testers) by considering multi-party communication complexity (see Appendix).

2 Preliminaries

For sake of simplicity, we focus on problems that correspond to the binary representation (i.e., to objects that are represented as sequences over a binary alphabet). We shall discuss the general case of non-binary alphabets at a later stage (i.e., in Section 6).[1]

Also, our main presentation refers to finite problems that correspond to bit strings of fixed length. One should think of these lengths as generic (or varying), and interpret the O-notation (as well as similar notions) as hiding universal constants (which do not depend on any parameter of the problems discussed).

We refer to the standard setting of communication complexity, and specifically to randomized two-party protocols in the model of shared randomness (cf. [17, Sec. 3]). We denote by $\langle A(x), B(y)\rangle(r)$ the (joint) output of the two parties, when the first party uses strategy A and gets input x, the second party uses strategy B and gets input y, and both parties have free access to the shared randomness r. Since many of the known reductions that use the methodology of [4] actually reduce from promise problems, we present communication problems in this more general setting. The standard case of decision problems is obtained by using a trivial promise (i.e., $P = \{0,1\}^{2n}$).

Definition 2.1 (two-party communication complexity): *Let $\Psi = (P, S)$ such that $P, S \subseteq \{0,1\}^{2n}$, and $\eta \geq 0$. A two-party* protocol that solves Ψ with error at most η *is a pair of strategies (A, B) such that the following holds* (w.r.t. some $\rho = \rho(n)$):

1. *If $(x, y) \in P \cap S$, then* $\Pr_{r\in\{0,1\}^\rho}[\langle A(x), B(y)\rangle(r)=1] \geq 1 - \eta$.
2. *If $(x, y) \in P \setminus S$, then* $\Pr_{r\in\{0,1\}^\rho}[\langle A(x), B(y)\rangle(r)=0] \geq 1 - \eta$.

The communication complexity of this protocol *is the maximum number of bits exchanged between the parties when the maximization is over all $x, y \in \{0,1\}^n$ and $r \in \{0,1\}^\rho$. The* η-error communication complexity of Ψ, *denoted* $\mathtt{CC}_\eta(\Psi)$, *is the minimum communication complexity of all protocols that solve Ψ with error at most η.*

For a Boolean function $f : \{0,1\}^{2n} \to \{0,1\}$, the two-party communication problem of computing f is the promise problem $\Psi_f \stackrel{\rm def}{=} (\{0,1\}^{2n}, \{(x,y) : f(x,y)=1\})$. Abusing notation, we let $\mathtt{CC}_\eta(f)$ denote $\mathtt{CC}_\eta(\Psi_f)$.

Note that randomized complexity with zero error (i.e., $\eta = 0$) collapses to deterministic complexity.[2] This is one reason that we kept η as a free parameter rather than setting it to a small constant (e.g., $\eta = 1/3$), as is the standard. Another reason for our choice is to allow greater flexibility in our presentation. For the same reason, we take the rather unusual choice of making the error probability explicit also in the context of property testing (where we also denote it by η). In the next definition, as in most work on *lower bounds* in property testing (cf. [10, 11, 7, 20]), we fix the proximity parameter (denoted ϵ).

Definition 2.2 (property testing): *Let $\Pi \subseteq \{0,1\}^\ell$, and $\epsilon, \eta > 0$. An* ϵ-tester with error η for Π *is a randomized oracle machine T that satisfies the following two conditions.*

[1] Jumping ahead, we note that, with respect to the general formulation, little is lost by considering only the binary representation.

[2] Note that $\mathtt{CC}_0(\cdot)$ is different from the *standard* notion of zero-error randomized communication complexity, since in the latter one considers the expected number of bits exchanged on the worst-case pair of inputs (whereas we considered the worst-case over both the shared randomness and the pair of inputs). Note that the difference between the expected complexity and the worst-case complexity is not very significant in the case of $\Theta(1)$-error communication complexity, but it is crucial in the case of zero-error.

1. *If $z \in \Pi$, then* $\Pr[T^z(\ell)=1] \geq 1-\eta$.
2. *If $z \in \{0,1\}^\ell$ is ϵ-far from Π, then* $\Pr[T^z(\ell)=0] \geq 1-\eta$*, where the distance between z and Π is* $\min_{z' \in \Pi}\{|\{i \in [\ell] : z_i \neq z'_i\}|/\ell\}$.

The query complexity of *T is the maximum number of queries that T makes, when the maximization is over all $z \in \{0,1\}^\ell$ and the coin tosses of T. The* η-error query complexity of ϵ-testing Π*, denoted* $\mathtt{Q}_\eta(\epsilon,\Pi)$*, is the minimum query complexity of all ϵ-testers with error η for Π.*

For any property Π and any $\eta > 0$, it holds that $\mathtt{Q}_\eta(\epsilon,\Pi) = O(\mathtt{Q}_{1/3}(\epsilon,\Pi))$, where the O-notation hides a $\log(1/\eta)$ factor. Thus, establishing a lower bound on the ϵ-testing query complexity of Π for any constant error, yields the same asymptotic lower bound for the (standard) error level of 1/3. In light of this fact, we may omit the constant error from our discussion; that is, when we say the query complexity of ϵ-testing Π we mean the 1/3-error query complexity of ϵ-testing Π. Likewise, $\mathtt{Q}(\epsilon,\Pi) = \mathtt{Q}_{1/3}(\epsilon,\Pi)$

3 The general formulation of the methodology

With the above preliminaries in place, we are ready to state the main result, which is proved by a straightforward adaptation of the ideas of [4].

Theorem 3.1 (property testing lower bounds via communication complexity): *Let $\Psi = (P,S)$ be a promise problem such that $P,S \subseteq \{0,1\}^{2n}$, and let $\Pi \subseteq \{0,1\}^\ell$ be a property, and $\epsilon,\eta > 0$. Suppose that the mapping $F : \{0,1\}^{2n} \to \{0,1\}^\ell$ satisfies the following two conditions:*

1. *For every $(x,y) \in P \cap S$, it holds that $F(x,y) \in \Pi$.*
2. *For every $(x,y) \in P \setminus S$, it holds that $F(x,y)$ is ϵ-far from Π.*

Then, $\mathtt{Q}_\eta(\epsilon,\Pi) \geq \mathtt{CC}_{2\eta}(\Psi)/B$*, where* $B = \max_{i\in[\ell]}\{\mathtt{CC}_{\eta/n}(f_i)\}$ *and $f_i(x,y)$ is the i^{th} bit of $F(x,y)$. Furthermore, if* $B = \max_{i\in[\ell]}\{\mathtt{CC}_0(f_i)\}$*, then* $\mathtt{Q}_\eta(\epsilon,\Pi) \geq \mathtt{CC}_\eta(\Psi)/B$.

The main result in [4] refers to a mapping F such that each $f_i(x,y)$ is a function of the i^{th} bit of x and the i^{th} bit of y (i.e., x_i and y_i). Indeed, in that case, $\ell = n$ and $B = 2$ (by the straightforward protocol in which the two parties exchange the relevant bits (i.e., x_i and y_i)).

Proof: Given an ϵ-tester with error η for Π and communication protocols for the f_i's, we present a two-party protocol for solving Ψ. The key idea is that, using their shared randomness, the two parties (holding x and y, respectively) can emulate the execution of the ϵ-tester, while providing it with virtual access to $F(x,y)$. Specifically, when the tester queries the i^{th} bit of the oracle, the parties provide it with the value of $f_i(x,y)$ by first executing the corresponding communication protocol.

The protocol for Ψ proceeds as follows: On local input x (resp., y) and shared randomness $r = (r_0, r_1, ..., r_\ell) \in (\{0,1\}^*)^{\ell+1}$, the first (resp. second) party invokes the ϵ-tester on randomness r_0, and answers the tester's queries by interacting with the other party. That is, each of the two parties invokes a local copy of the tester's program, but both copies are invoked on the same randomness, and are fed with identical answers to their (identical) queries. When the tester issues a query $i \in [\ell]$, the parties compute the value of $f_i(x,y)$ by using the corresponding communication protocol, and feed $f_i(x,y)$ to (their local copy of) the tester. Specifically, denoting the latter protocol (i.e., pair of strategies) by (A_i, B_i), the parties answer with $\langle A_i(x), B_i(y)\rangle(r_i)$. When the tester halts, each party outputs the output it has obtained from (its local copy of) the tester.

Turning to the analysis of this protocol, we note that the two local executions of the tester are identical, since they are fed with the same randomness and the same answers (to the same queries). The total number of bits exchanged by the two parties is at most B times the query complexity of ϵ-tester; that is, the communication complexity of this protocol is at most $B \cdot q$, where q denotes the query complexity of the ϵ-tester.

Let us consider first the furthermore clause; that is, assume that $B = \max_{i\in[\ell]}\{\mathtt{CC}_0(f_i)\}$. In this case, the parties always provide the ϵ-tester, denoted T, with the correct answers to its queries. Now, if $(x, y) \in P \cap S$, then $F(x, y) \in \Pi$, which implies that $\Pr[T^{F(x,y)}(\ell) = 1] \geq 1 - \eta$, which in turn implies that the parties output 1 with probability at least $1 - \eta$. On the other hand, if $(x, y) \in P \setminus S$, then $F(x, y)$ is ϵ-far from Π, which implies that $\Pr[T^{F(x,y)}(\ell) = 0] \geq 1 - \eta$, which in turn implies that the parties output 0 with probability at least $1 - \eta$. Hence, in this case (and assuming that T has query complexity $\mathtt{Q}_\eta(\epsilon, \Pi)$), we get $\mathtt{CC}_\eta(\Psi) \leq B \cdot \mathtt{Q}_\eta(\epsilon, \Pi)$.

Turning to the main claim, we may assume that $q \leq n$, since otherwise we can just use the trivial communication protocol for Ψ (which has complexity n). Recall that if $(x, y) \in P \cap S$, then $\Pr[T^{F(x,y)}(\ell) = 1] \geq 1 - \eta$. However, the emulation of T is given access to bits that are each correct only with probability $1 - (\eta/n)$, and hence the probability that the protocol outputs 1 is at least $1 - \eta - (q\eta/n) \geq 1 - 2\eta$. On the other hand, if $(x, y) \in P \setminus S$, then $\Pr[T^{F(x,y)}(\ell) = 0] \geq 1 - \eta$. Taking account of the errors in computing the f_i's, we conclude that the probability that the protocol outputs 0 in this case is at least $1 - 2\eta$. The claim follows. ■

Corollary 3.2 (a special case of Theorem 3.1): *Let $\Psi = (P, S)$, Π, $\epsilon, \eta > 0$, F, and the f_i's be as in Theorem 3.1. Suppose that each $f_i(x, y)$ either depends on at most one bit of x* (and possibly some bits of y) *or depends on at most one bit of y* (and possibly some bits of x). *Then, $\mathtt{Q}_\eta(\epsilon, \Pi) \geq \mathtt{CC}_\eta(\Psi)/2$.*

Proof: In this case $\mathtt{CC}_0(f_i) \leq 2$ for each $i \in [\ell]$. The claim follows by the furthermore clause of Theorem 3.1. ■

Corollary 3.3 (the special case of "simple combining operator" [4]): *Let $\Psi = (P, S)$, Π, $\epsilon, \eta > 0$, F, and the f_i's be as in Corollary 3.2. Suppose that each $f_i(x, y)$ depends only on the i^{th} bit of x and the i^{th} bit of y. Then, $\mathtt{Q}_\eta(\epsilon, \Pi) \geq \mathtt{CC}_\eta(\Psi)/2$.*

Corollary 3.3 is stated merely for sake of reference. We note that the methodology as presented in [4] is slightly more general than Corollary 3.3, since it refers to sequences over an arbitrary alphabet Σ (rather than to bit strings).[3] For further discussion, see Section 6.

4 Application to codeword testing

The applications presented in this section are (of course) negative ones: They are families of codes for which codeword testing is extremely hard. Such families were known before (cf., e.g., [3]).[4] The following results can also be proved by using the restricted methodology as presented in [4] and Corollary 3.3 (see discussion following the proof of Theorem 4.1), but we believe that deriving them

[3] In that case, $f_i : \Sigma^{2n} \to \Sigma$, and simple combining operators correspond to the case that each $f_i(x, y)$ depends only on the i^{th} symbol of x and the i^{th} symbol of y. The assertion then is that $\mathtt{Q}_\eta(\epsilon, \Pi) \geq \mathtt{CC}_\eta(\Psi)/2\lceil\log_2 |\Sigma|\rceil$.

[4] In contrast, for locally testable codes (cf., e.g., [12, 8]), codeword testing is very easy.

via the general methodology (i.e., using either Corollary 3.2 or Theorem 3.1) is simpler. Recall that the rate of a code $C : \{0,1\}^n \to \{0,1\}^\ell$ is n/ℓ, and its relative distance is d/ℓ such that every two different codewords differ on at least d positions (i.e., for every $x, y \in \{0,1\}^n$ such that $x \neq y$. it holds that $C(x)$ and $C(y)$ disagree on at least d positions).

Theorem 4.1 (on the hardness of testing codewords in some codes): *Let $\{\Psi_n = (P_n, S_n)\}_{n\in\mathbb{N}}$ be a family of communication problems such that $P_n, S_n \subseteq \{0,1\}^{2n}$ and for some constant $\eta > 0$ it holds that $\mathtt{CC}_\eta(\Psi_n) = \Omega(n)$. Let $\{C_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{n\in\mathbb{N}}$ be a family of codes of constant relative distance. Then, for some constant $\epsilon > 0$, the query complexity of ϵ-testing the property $\Pi = \{\Pi_n\}_{n\in\mathbb{N}}$, where*

$$\Pi_n \overset{\text{def}}{=} \{C_n(x)C_n(y) : (x,y) \in P_n \cap S_n\}, \tag{1}$$

is $\Omega(n)$. That is, $\mathtt{Q}(\epsilon, \Pi_n) = \Omega(n)$.

Note that the elements of Π are codewords of a code C' that has constant relative distance; that is, each $(x,y) \in P_n \cap S_n$ is encoded by $C'_n(xy) = C_n(x)C_n(y)$. Also note that if C has constant rate, then so does C', because $\mathtt{CC}_\eta(\Psi_n) = \Omega(n)$ implies that $\log|P_n \cap S_n| = \Omega(n)$. In any case, Theorem 4.1 asserts that, for the code C', codeword testing requires $\Omega(n)$ queries. Recall that such codes were known before (cf., e.g., [3]), but the code C' is definitely different. An appealing example of such a code C' can be obtained by using the inner product (mod 2) in the role of Ψ (and using the communication complexity lower bound of [6]); that is, $P_n = \{0,1\}^{2n}$ and $(x,y) \in S_n$ iff the inner product of x and y (mod 2) equals 0.

Proof: We invoke Corollary 3.2 while using $F(x,y) = C_n(x)C_n(y)$ and noting that each bit in $F(x,y)$ either depends only on bits of x or depends only on bits of y. By Eq. (1), for every $(x,y) \in P_n \cap S_n$ it holds that $F(x,y)$ is in Π_n. On the other hand, if $(x,y) \in P_n \setminus S_n$, then by the distance of C it holds that $F(x,y)$ is $\Omega(1)$-far from Π. Specifically, if the relative distance of C is δ, then this $F(x,y)$ must be $\delta/2$-far from Π_n (since at least one of the two codewords in $F(x,y)$ must be replaced). Indeed, Corollary 3.2 ("only") implies $\mathtt{Q}_\eta(\delta/2, \Pi_n) = \Omega(n)$, but using $\mathtt{Q}(\delta/2, \Pi_n) = \mathtt{Q}_{1/3}(\delta/2, \Pi_n) = \Omega(\mathtt{Q}_\eta(\delta/2, \Pi_n)/\log(1/\eta))$, we are done. ■

An alternative proof of Theorem 4.1. As stated up-front, Theorem 4.1 can be proved by applying the communication complexity methodology as formulated in [4] (cf. Corollary 3.3). In order to do this, we need to introduce an auxiliary communication complexity problems, which is related to Ψ. Specifically, let $\Psi'_n = (P'_n, S'_n)$ be such that

$$\begin{aligned} P'_n &\overset{\text{def}}{=} \{(C_n(x)0^{\ell(n)}, 0^{\ell(n)}C_n(y)) : (x,y) \in P_n\} \\ S'_n &\overset{\text{def}}{=} \{(C_n(x)0^{\ell(n)}, 0^{\ell(n)}C_n(y)) : (x,y) \in S_n\}. \end{aligned}$$

(That is, x is replaced by $C_n(x)0^\ell$, whereas y is replaced by $0^\ell C_n(y)$.) First, note that $\mathtt{CC}_\eta(\Psi'_n) \geq \mathtt{CC}_\eta(\Psi_n)$, since a communication protocol for Ψ is obtained by a straightforward emulation of any communication protocol for Ψ'. Next, we shall reduce the communication problem Ψ'_n to $\delta/2$-testing Π, by using $F'(u,v) \overset{\text{def}}{=} u \oplus v$, where $\oplus$ denotes the bit-by-bit XOR of strings (which indeed is a simple combining operator). Indeed, for every $(u,v) \in P'_n$ it holds that $u = C_n(x)0^{\ell(n)}$ and $v = 0^{\ell(n)}C_n(y)$ for some $x, y \in \{0,1\}^n$, and so $F'(u,v) = u \oplus v = C_n(x)C_n(y)$, which equals the value of $F(x,y)$ as defined in the proof of Theorem 4.1. Since F' falls within the restricted framework of [4] (cf. Corollary 3.3), by their result $\mathtt{Q}_\eta(\delta/2, \Pi_n) \geq \mathtt{CC}_\eta(\Psi'_n)/2$. A similar comment applies also to the following result.

Theorem 4.2 (more on the hardness of testing codewords in some codes): *Let* $\{C_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{n\in\mathbb{N}}$ *be a family of* linear *codes* (i.e., $C_n(x \oplus y) = C_n(x) \oplus C_n(y)$) *of constant relative distance. Let* $\mathrm{wt}(z)$ *denote the Hamming weight of* z*; that is,* $\mathrm{wt}(z) = |\{i\in[|z|] : z_i=1\}|$*. Then, for some constant* $\epsilon > 0$ *and any function* $k : \mathbb{N} \to \mathbb{N}$ *such that* $k(n)$ *is even and* $k(n) < n/2$*, the query complexity of* ϵ*-testing the property*

$$\Pi_n \stackrel{\text{def}}{=} \{C_n(z) : z\in\{0,1\}^n \wedge \mathrm{wt}(z)=k(n)\} \tag{2}$$

is $\Omega(k(n))$*. That is,* $\mathsf{Q}(\epsilon, \Pi_n) = \Omega(k(n))$*.*

Note that Π_n is a code; actually, it is a sub-code of the (linear) code C. In the special case that C is the Hadamard code, the property Π_n is $k(n)$-linearity; that is, the codewords of the Hadamard code corresponds to linear functions (from $\mathrm{GF}(2)^n$ to $\mathrm{GF}(2)$) and the codewords of Π_n are $k(n)$-linear (i.e., they are linear functions that depend on exactly $k(n)$ variables). This special case of Theorem 4.2 was proved in [4].

Proof: We reduce from the communication problem SET DISJOINTNESS, while using Theorem 3.1. Specifically, we consider the $k/2$-disjointness problem, denoted $\{\mathrm{DISJ}_n^{(k)} = (P_n, S_n)\}_{n\in\mathbb{N}}$, where $P_n, S_n \subseteq \{0,1\}^{2n}$ such that $(x, y) \in P_n$ if $\mathrm{wt}(x) = \mathrm{wt}(y) = k(n)/2$, and $(x, y) \in S_n$ if (the "intersection" set) $I(x, y) \stackrel{\text{def}}{=} \{i \in [n] : x_i = y_i = 1\}$ is empty. Thus, for every $(x, y) \in P_n$ it holds that $\mathrm{wt}(x \oplus y) = k(n) - 2|I(x, y)|$. We use $F(x, y) = C_n(x \oplus y)$ and note that (by the linearity of C) the i^{th} bit of $C_n(x \oplus y) = C_n(x) \oplus C_n(y)$ can be computed by exchanging the i^{th} bits of $C_n(x)$ and $C_n(y)$. The claimed lower bound follows by combining the celebrated result $\mathsf{CC}_{1/3}(\mathrm{DISJ}_n^{(k)}) = \Omega(k(n))$, which is implicit in [14] (see also [4, Lem. 2.6]), with Theorem 3.1 (while using the hypothesis that C has constant relative distance). ■

Digest. While both Theorems 4.1 and 4.2 can be proved by applying the restricted methodology of [4] (cf. Corollary 3.3) after introducing suitable auxiliary communication complexity problems, our proofs avoid the introduction of such auxiliary problems. Instead, our proofs are based on the existence of simple protocols for exchanging bits in the encoding of the inputs under error correcting codes. Although these bits may depend on a linear number of bits in the original input, each party can compute the relevant bit by itself. Indeed, exactly the same computations take place when using the restricted methodology of [4] (cf. Corollary 3.3), but there these computations take place in the reduction of the original communication problem to an auxiliary one (which must be introduced in order to use the restricted methodology). When using the general methodology, the foregoing computation take place in the communication protocols that demonstrate that each bit in $F(x, y)$ has low communication complexity, and this demonstration is performed without introducing any auxiliary problem. We stress that the issue is not with these simple computations, but rather with whether the lower bound proof requires the (explicit) introduction of auxiliary communication problems.

5 Nonadaptive testers and one-way communication

Following [4], we also present a version of the method that relates the complexity of nonadaptive testers to the communication complexity of one-way protocols.

One-way communication complexity. In one-way communication protocols the first party sends a single message to the second party, who is the only party that produces an output. Thus, it is natural to denote the outcome of such a protocol by $B(y, r, A(x, r))$, where A and B are the algorithms employed by the two parties (and x, y, r are the private inputs and the shared randomness, respectively, as in Definition 2.1). For $\Psi = (P, S)$ as in Definition 2.1, the η-error one-way communication complexity of Ψ, denoted $\overrightarrow{\mathsf{CC}}_\eta(\Psi)$, is the minimum communication complexity of all *one-way* protocols that solve Ψ with error at most η.

Nonadaptive testers. A nonadaptive oracle machine is one that determines all its queries based solely on its explicit input and its internal coin tosses, as opposed to a general (adaptive) oracle machine that may select its queries based also on the answers to prior queries. The η-error nonadaptive query complexity of ϵ-testing Π, denoted $\mathsf{Q}^{\rm na}_\eta(\epsilon, \Pi)$, is the minimum query complexity of all *nonadaptive* ϵ-testers with error η for Π.

Theorem 5.1 (Theorem 3.1, revised for nonadaptive testers vs one-way communication): *Let $\Psi = (P, S)$, Π, $\epsilon, \eta > 0$, F, and the f_i's be as in Theorem 3.1. Then, $\mathsf{Q}^{\rm na}_\eta(\epsilon, \Pi) \geq \overrightarrow{\mathsf{CC}}_{2\eta}(\Psi)/B$, where $B = \max_{i\in[\ell]}\{\overrightarrow{\mathsf{CC}}_{\eta/n}(f_i)\}$. Furthermore, if $B = \max_{i\in[\ell]}\{\overrightarrow{\mathsf{CC}}_0(f_i)\}$, then $\mathsf{Q}^{\rm na}_\eta(\epsilon, \Pi) \geq \overrightarrow{\mathsf{CC}}_\eta(\Psi)/B$.*

Again, the main result in [4] uses a mapping F such that each $f_i(x, y)$ is a function of the $i^{\rm th}$ bits of x and y. Indeed, in that case, $\ell = n$ and $B = 1$, by the straightforward one-way protocol in which the first party sends the relevant bit (i.e., x_i) to the second party.

Proof: We merely adapt the proof of Theorem 3.1: Given a nonadaptive ϵ-tester with error η for Π and one-way communication protocols for the f_i's, we present a one-way protocol for solving Ψ. Again, using their shared randomness, each of the two parties determines the (nonadaptive) queries of the tester, and the first party communicates to the second party the information it needs in order to determine the oracle's answers to the tester. Specifically, if position i is included in the set of nonadaptive queries, then the first party employs the one-way communication protocol for f_i, which results in sending a message that allows the second party to determine $f_i(x, y)$. The rest of the analysis proceeds as in the proof of Theorem 3.1, under the obvious modifications. In particular, here each of the two parties locally determines the set of queries made on the given randomness, but only the second party obtains (via the one-way communication protocols) the answers to these (nonadaptive) queries, which it feeds to the tester, and only the second party obtains the final verdict of the tester.[5] (This is OK, because here only the second party needs to generate an output.) ■

Nonadaptive testers and simultaneous communication complexity. David Woodruff suggested to replace one-way communication in Theorem 5.1 by simultaneous communication. The model of simultaneous communication protocols consists of three parties such that only two parties obtain inputs, whereas (only) the third party (called the referee) produces the output. Communication in unidirectional from each of the two main parties (which obtain inputs) to the referee: Based on its own local input (and the shared randomness), each party sends a (single) message to the

[5] Formally, a nonadaptive tester T consists of a pair of algorithms, Q and D, which use the same randomness, such that Q determines the tester's query and D its decision given the corresponding answers; that is, $T^z(r) = D(r, z_{i_1}, ..., z_{i_q})$, where $(i_1, ..., i_q) = Q(r)$. In our one-way communication protocol each of the two parties locally determines $(i_1, ..., i_q) = Q(r_0)$, then, for each $j \in [q]$, the first party sends to the second party the message required for the computation of $f_{i_j}(x, y)$, and finally (after computing all the $f_{i_j}(x, y)$'s) the second party invokes D and outputs $D(r_0, f_{i_1}(x, y), ..., f_{i_q}(x, y))$.

referee, who then produces the output (based also on the joint randomness).[6] Thus, it is natural to denote the outcome of such a protocol by $R(r, A(x,r), B(y,r))$, where A and B are the algorithms employed by the two main parties and R is the algorithm employed by the referee. For $\Psi = (P,S)$ as in Definition 2.1, the η-error simultaneous communication complexity of Ψ, denoted $\ddot{\mathsf{CC}}_\eta(\Psi)$, is the minimum communication complexity of all *simultaneous* protocols that solve Ψ with error at most η. Note that $\vec{\mathsf{CC}}_\eta(\Psi) \leq \ddot{\mathsf{CC}}_\eta(\Psi)$, since the second party (in the one-way communication model) can emulate the referee (in the simultaneous communication model).

Theorem 5.2 (Theorem 3.1, revised for nonadaptive testers vs simultaneous communication): *Let $\Psi = (P,S)$, Π, $\epsilon, \eta > 0$, F, and the f_i's be as in Theorem 3.1. Then, $\mathsf{Q}^{\rm na}_\eta(\epsilon, \Pi) \geq \ddot{\mathsf{CC}}_{2\eta}(\Psi)/B$, where $B = \max_{i\in[\ell]}\{\ddot{\mathsf{CC}}_{\eta/n}(f_i)\}$. Furthermore, if $B = \max_{i\in[\ell]}\{\ddot{\mathsf{CC}}_0(f_i)\}$, then $\mathsf{Q}^{\rm na}_\eta(\epsilon, \Pi) \geq \ddot{\mathsf{CC}}_\eta(\Psi)/B$.*

Strictly speaking, Theorem 5.2 is not stronger than Theorem 5.1, but we do expect it to be more useful (since the possibility of $\ddot{\mathsf{CC}}_\eta(\Psi) \gg \vec{\mathsf{CC}}_\eta(\Psi)$ seems more promising than the potential cost of $\ddot{\mathsf{CC}}_\eta(f_i) \geq \vec{\mathsf{CC}}_\eta(f_i)$).

Proof: We merely adapt the proof of Theorem 5.1, replacing one-way protocols by simultaneous ones. Given a nonadaptive ϵ-tester with error η for Π and simultaneous communication protocols for the f_i's, we present a simultaneous protocol for solving Ψ. Again, using their shared randomness, each of the three parties determines the (nonadaptive) queries of the tester, and each of the two main parties communicates to the referee the information it needs in order to determine the oracle's answers to the tester. Specifically, if position i is included in the set of nonadaptive queries, then each main party employs the simultaneous communication protocol for f_i, which results in sending a message to the referee, who upon receiving these two messages determines the value of $f_i(x,y)$. The rest of the analysis proceeds as in the proof of Theorem 5.1, under the obvious modifications. In particular, here each of the parties locally determines the set of queries made on the given randomness, but only the referee obtains (via the simultaneous communication protocols) the answers to these (nonadaptive) queries, which it feeds to the tester, and only the referee obtains the final verdict of the tester. (This is OK, because here only the referee needs to generate an output.) $\blacksquare$

6 Ramifications: one-sided error and non-binary alphabets

In this section, we briefly comment on two ramifications, which have appeared in [4].

One-sided error versions. One-sided error testers are testers that are allowed no error when the object has the property; that is, if T is a one-sided error tester for Π, then for every $z \in \Pi$ it holds that $\Pr[T^z(\ell) = 1] = 1$. (Error probability is only allowed in the case that $z \notin \Pi$.) Deriving lower bounds for such testers via the communication complexity methodology requires referring to the corresponding one-sided error version of communication complexity. That is, we shall consider communication protocols for (P,S) such that for every $(x,y) \in P \cap S$ it holds that $\Pr_{r\in\{0,1\}^\rho}[\langle A(x), B(y)\rangle(r) = 1] = 1$. In this case we may only use *zero-error* communication protocols for computing the f_i's.[7] For sake of clarity, we state the (main) corresponding result.

[6] It is crucial that the two main parties have access to the same shared randomness, since they cannot communicate with one another. In contrast, it is less essential that the referee also has access to this shared randomness, since one of the main parties can send it along while relying on the fact that the randomness can be made logarithmic in the input length (cf. [17, Thm. 3.14]).

[7] The point is that in this case we cannot afford any error, regardless of the value of $f_i(x,y)$.

Theorem 6.1 (Theorem 3.1, revised for one-sided error): *Let $\Psi = (P, S)$, Π, $\epsilon, \eta > 0$, F, and the f_i's be as in Theorem 3.1. Then, the one-sided η-error query complexity of ϵ-testing Π is at least $1/B$ times the one-sided η-error communication complexity of Ψ, where $B = \max_{i\in[\ell]}\{\mathtt{CC}_0(f_i)\}$.*

By one-sided η-error query (resp., communication) complexity, we mean the complexity of one-sided error testers (resp., protocols) that have error probability at most η on the "no-instances".

Non-binary alphabets. Our treatment of the subject-matter referred to computational problems over binary strings. Clearly, any computational problem over other alphabets can be stated via binary alphabets, but sometimes the former formulation is more appealing. This holds, in particular, for property testing problems. Examples include property testing in the bounded-degree graph model (cf. [11]) and testing monotonicity over general range (cf. [5]). Thus, we may consider properties and communication problems that refer to sequences over some alphabet Σ, rather that over a binary alphabet. The problem, however, is that the communication protocols themselves need not respect the "integrity of the alphabet" (i.e., messages are arbitrary functions of the input, regardless of the alphabet in which the latter is encoded). (Things are, of course, different in the context of property testing: The tester's queries must respect the input format.)

Given this state of affairs, it seems that we gain little by a general treatment. Instead, when studying a property that refer to object that are encoded as sequences over Σ, we may consider their encoding as binary strings (which means that we lose a factor of $\log_2|\Sigma|$ in the query lower bounds that we derive).[8] When using the trivial encoding, we also lose in the value of the proximity parameter for which the lower bound hold, but this loss may be reduced to a constant by using encoding via a good error correcting code. Details follow.

Suppose that we wish to establish a lower bound for ϵ-testing a property Π of objects that are encoded as sequences over Σ, and suppose that we have a reduction F from some communication problem to testing $\Pi \subseteq \Sigma^\ell$. Then, we may consider the corresponding binary property Π' (i.e., $\Pi' = \{(C(z_1), ..., C(z_\ell)) : (z_1, ..., z_\ell) \in \Pi\}$, where $C : \Sigma \to \{0,1\}^{O(\log|\Sigma|)}$ is a good code), and the corresponding reduction F' (which encodes the F-values under this C). Now, if $F(x,y) \in \Pi$ then $F'(x,y) \in \Pi'$, whereas if $F(x,y)$ is ϵ-far from Π then $F'(x,y)$ is ϵ'-far from Π', where $\epsilon' = \Omega(\epsilon)$. Hence, we derive a lower bound on $\mathtt{Q}(\epsilon', \Pi')$, which yields a lower bound on $\mathtt{Q}(\epsilon', \Pi)$ (i.e., $\mathtt{Q}(\epsilon', \Pi) \geq \mathtt{Q}(\epsilon', \Pi')/O(\log|\Sigma|)$).

We stress that the foregoing discussion refers to the general formulation as presented in Theorem 3.1. In the context of the special case presented in [4] there is a benefit in directly treating arbitrary alphabet (as indeed done in [4]), since this allows a less restricted notion of simple combining operators. Recall that the formulation in [4] requires that each symbol in $F(x,y)$ can be computed as a function of the corresponding symbols of x and y. When we view $F(x,y)$ as a binary string, this means that the i^{th} bit of $F(x,y)$ is a function of x_i and y_i only. But if we view $F(x,y)$ as a sequence over Σ (or as binary strings partitioned into $O(\log|\Sigma|)$-bit long blocks), then this means that each bit in the description of the i^{th} symbol (or i^{th} block) of $F(x,y)$ may depend on all bits in the description of the i^{th} symbol of x and the i^{th} symbol of y (or on all bits in the i^{th} blocks of x and y).

[8] When we derive a query lower bound regarding testing an object under the binary encoding, we lose a factor of $\log_2|\Sigma|$ when inferring from it a lower bound for testing this object encoded as a sequence over Σ.

7 On emulating the general formulation by the restricted one

In this section we show that the restricted formulation of [4] (via simple combination operators, cf. Corollary 3.3) can emulate the general formulation as captured by Theorem 3.1. The emulations we present come at the cost of some degradation in the parameters obtained, but this degradation is relatively small. Furthermore, as will become apparent throughout our proof, the degradation is even smaller in some special cases. In fact, we find it useful to present the proof by going from special cases to more general cases.

Before stating our most general result (i.e., Theorem 7.1), we make a couple of tedious comments. First, we assume for simplicity that $\eta \geq \epsilon$. While this seems the most relevant case (e.g., typically one considers $\eta = 1/3$), the result generalizes to arbitrary $\eta > 0$ (while replacing some $\log(1/\epsilon)$ factors by $\log(1/\eta)$ factors). Second, we restrict ourselves to $\epsilon' = \widetilde{\Omega}(1/n)$, and note that the case of $\epsilon' \in (0, \widetilde{\Omega}(1/n)]$ is uninteresting (in light of the application of Theorem 7.1).[9] Lastly, we stress that the constants hidden in the $\widetilde{O}$-notation are universal constants, which are independent of all the parameters that appear in the statement of the result.

Theorem 7.1 (emulating Theorem 3.1 via simple combining operators): *Let $\Psi = (P, S)$, $\Pi \subseteq \{0,1\}^\ell$, $\eta \geq \epsilon > 0$, $F : \{0,1\}^{2n} \to \{0,1\}^\ell$, and the f_i's be as in Theorem 3.1.*[10] *Suppose that $B = \max_{i\in[\ell]}\{\mathtt{CC}_{\eta/n}(f_i)\}$, which implies $\mathtt{Q}_\eta(\epsilon, \Pi) \geq \mathtt{CC}_{2\eta}(\Psi)/B$* (by the main part of Theorem 3.1). *Then, there exists a communication problem $\Psi' = (P', S')$ such that $\mathtt{CC}_\eta(\Psi') \geq \mathtt{CC}_\eta(\Psi)$ and a property Π' such that $\mathtt{Q}_\eta(\epsilon', \Pi) \geq \frac{\mathtt{Q}_\eta(\epsilon'/2,\Pi') - \widetilde{O}(B/\epsilon')}{\widetilde{O}(B)\cdot\log(\mathtt{Q}_\eta(\epsilon'/2,\Pi')/\eta)}$* (for every $\epsilon' = \widetilde{\Omega}(1/n)$), *whereas Ψ' and Π' are related as follows:*

1. *For every $(u, v) \in P' \cap S'$, it holds that $u \oplus v \in \Pi'$.*
2. *For every $(u, v) \in P' \setminus S'$, it holds that $u \oplus v$ is $\Omega(\epsilon)$-far from Π'.*

This means that whenever $\mathtt{Q}_\eta(\epsilon, \Pi) \geq \mathtt{CC}_{2\eta}(\Psi)/B$ is established by Theorem 3.1, when using $B = \max_{i\in[\ell]}\{\mathtt{CC}_{\eta/n}(f_i)\}$, we can (roughly) establish $\mathtt{Q}_\eta(\Omega(\epsilon), \Pi) = \widetilde{\Omega}(\mathtt{CC}_\eta(\Psi))/B$ by using the formulation as presented in [4] (cf. Corollary 3.3). This alternative derivation uses the mapping $F'(u, v) = u \oplus v$, and proceeded via

$$\begin{aligned}
\mathtt{Q}_\eta(\Omega(\epsilon), \Pi) &\geq \frac{\mathtt{Q}_\eta(\Omega(\epsilon), \Pi') - \widetilde{O}(B/\epsilon)}{\widetilde{O}(B)\cdot\log(\mathtt{Q}_\eta(\Omega(\epsilon), \Pi')/\eta)} \\
&\geq \frac{\mathtt{CC}_\eta(\Psi') - \widetilde{O}(B/\epsilon)}{\widetilde{O}(B)\cdot\log(\mathtt{CC}_\eta(\Psi')/\eta)} \\
&\geq \frac{\mathtt{CC}_\eta(\Psi) - \widetilde{O}(B/\epsilon)}{\widetilde{O}(B)\cdot\log(\mathtt{CC}_\eta(\Psi)/\eta)}
\end{aligned}$$

[9] In that application, we derive a lower bound on $\mathtt{Q}_\eta(\Omega(\epsilon), \Pi)$. This lower bound is smaller than $\mathtt{CC}_\eta(\Psi) - \widetilde{O}(1/\epsilon)$, which in turn is negative in the case of $\epsilon \leq \mathrm{poly}(\log n)/n$.

[10] Recall that this means that $\Psi = (P, S)$ is a promise problem such that $P, S \subseteq \{0,1\}^{2n}$, that $\Pi \subseteq \{0,1\}^\ell$ is a property, and that the mapping $F : \{0,1\}^{2n} \to \{0,1\}^\ell$ satisfies the following two conditions:

1. For every $(x, y) \in P \cap S$, it holds that $F(x, y) \in \Pi$.
2. For every $(x, y) \in P \setminus S$, it holds that $F(x, y)$ is ϵ-far from Π.

Lastly, $f_i(x, y)$ denotes the i^{th} bit of $F(x, y)$.

where the second inequality is by [4] (cf. Corollary 3.3).[11] The lower bound derived this way is quantitatively inferior to the one derived by Theorem 3.1. In particular, in the denominator B is replaced by $\widetilde{O}(B) \cdot \log \mathtt{CC}_\eta(\Psi)$, and the lower bound refers to a smaller value of the proximity parameter (i.e., $\Omega(\epsilon)$ rather than ϵ). However, when we aim at large values of $\mathtt{CC}_{2\eta}(\Psi)/B$, the loss of factors of the form of $\log \mathtt{CC}_{2\eta}(\Psi)$ (and $\log B$) seems relatively small. In any case, the additive loss of $\widetilde{O}(B/\epsilon)$ in the numerator is typically insignificant, since we typically aim at much higher lower bounds.

Organization of the proof. Theorem 7.1 is proved in three steps. We shall start with the special case in which each f_i can be expressed as $d_i(g_i(x), h_i(y))$, where $|g_i(x)|, |h_i(y)| \leq B/2$. Indeed, in this case $\mathtt{CC}_0(f_i) \leq B$, via the straightforward protocol in which the parties exchange $g_i(x)$ and $h_i(y)$. We shall then move to the special case where $\mathtt{CC}_0(f_i) \leq B$ (i.e., the case of arbitrary deterministic protocols), and end with the general case (i.e., $\mathtt{CC}_{\eta/n}(f_i) \leq B$).

In each step, we shall introduce an auxiliary communication problem Ψ' and an auxiliary property Π', and establish three relations of the type asserted in the theorem: (1) a relation between the communication complexity problems (i.e., Ψ' and Ψ), (2) a relation between the property testing problems (i.e., Π' and Π), and (3) a relation between the auxiliary problems (i.e., Ψ' and Π').

Step 1: a syntactic special case

We start by considering the special case in which for every $i \in [\ell]$ it holds that $f_i(x,y) = d_i(g_i(x), h_i(y))$, where $|g_i(x)|, |h_i(y)| \leq B/2$ and $d_i : \bigcup_{j,k \leq B/2} \{0,1\}^{j+k} \to \{0,1\}$, which implies $\mathtt{Q}_\eta(\epsilon, \Pi) \geq \mathtt{CC}_\eta(\Psi)/B$ (by the furthermore clause of Theorem 3.1). In this case we shall present *a communication problem $\Psi' = (P', S')$ such that $\mathtt{CC}_\eta(\Psi') \geq \mathtt{CC}_\eta(\Psi)$ and a property Π' such that $\mathtt{Q}_\eta(\epsilon', \Pi) \geq \mathtt{Q}_\eta(\epsilon', \Pi')/B$, whereas Ψ' and Π' are related as follows*:

1. *For every $(u, v) \in P' \cap S'$, it holds that $u \oplus v \in \Pi'$.*

2. *For every $(u, v) \in P' \setminus S'$, it holds that $u \oplus v$ is ϵ/B-far from Π'.*

This means that whenever $\mathtt{Q}_\eta(\epsilon, \Pi) \geq \mathtt{CC}_\eta(\Psi)/B$ is established by Theorem 3.1, when using f_i's of the above form, we can establish $\mathtt{Q}_\eta(\epsilon/B, \Pi) \geq \mathtt{CC}_\eta(\Psi)/2B$ by using the restricted formulation via simple combining operators as presented in [4] (cf. Corollary 3.3). Note that there is some degradation in the parameters also in this special case: The main issue is not that B is replaced by $2B$, but rather than the lower bound refers to a smaller value of the proximity parameter (i.e., ϵ/B rather than ϵ). We shall refer to this issue when discussing the case of general deterministic protocols.

We start with a few simplifying assumptions, which hold without loss of generality up to some degradation in some parameters.

1. F is non-shrinking; that is, $\ell \geq n$.

   Otherwise, for $m \stackrel{\rm def}{=} \lceil n/\ell \rceil$, consider the property $\Pi^{(m)} \stackrel{\rm def}{=} \{z^m : z \in \Pi\}$ and the mapping $(x, y) \mapsto F(x, y)^m$, which satisfy the conditions of the theorem (and of the current case). Lower bounds on the query complexity of testing $\Pi^{(m)}$ imply similar bounds for Π, because $\mathtt{Q}_\eta(2\epsilon, \Pi^{(m)}) \leq \mathtt{Q}_\eta(\epsilon, \Pi) + O(\epsilon^{-1} \log(1/\eta))$ (by using also a repetition test).

[11] In addition, this inequality uses the relation between F', Ψ' and Π' established in Theorem 7.1. The other two inequalities use the relation between Π and Π' and the relation between Ψ' and Ψ, respectively.

2. The mappings $x \mapsto (g_1(x), ..., g_\ell(x))$ and $y \mapsto (h_1(y), ..., h_\ell(y))$ are one-to-one.

   Using the first assumption, for each $i \in [n]$, we append x_i to $g_i(x)$. Ditto for y_i and $h_i(y)$.

3. For each $i \in [\ell]$, it holds that $|g_i(x)| = |h_i(y)| = B/2$. Furthermore, not all $B/2$-bit long strings are in the range of g_i, and ditto for h_i.

   We use a standard encoding of $\bigcup_{j\in[B']}\{0,1\}^i$ by $(B'+1)$-bit long strings (e.g., encoding the string s by $s10^{B'-|s|}$).

4. For each $i \in [\ell]$, the predicate $d_i : \{0,1\}^B \to \{0,1\}$ is onto.

   Using the assumption that not all $B/2$-bit long strings are in the range of g_i and ditto for h_i, we can modify d_i on a pair that is *not* in the range of (h_i, g_i) without affecting the conditions of the theorem (and of the current case).

We now turn to the construction of Ψ' and Π'. First, define $\Psi' = (P', S')$ such that

$$P' \stackrel{\rm def}{=} \{(g_1(x)\cdots g_\ell(x)0^{\ell B/2}, 0^{\ell B/2}h_1(y)\cdots h_\ell(y)) : (x,y) \in P\} \quad (3)$$

$$S' \stackrel{\rm def}{=} \{(g_1(x)\cdots g_\ell(x)0^{\ell B/2}, 0^{\ell B/2}h_1(y)\cdots h_\ell(y)) : (x,y) \in S\}. \quad (4)$$

Note that $\mathtt{CC}_\eta(\Psi) \leq \mathtt{CC}_\eta(\Psi')$, since a protocol for Ψ can proceed by emulating the protocol for Ψ'. Specifically, on input x the first party computes $g_1(x)\cdots g_\ell(x)0^{\ell B/2}$, and likewise the second party computes $0^{\ell B/2}h_1(y)\cdots h_\ell(y)$. By the one-to-one feature of these mappings, the answer obtained for Ψ' is valid for Ψ.

Next, we introduce the property $\Pi' \subseteq \{0,1\}^{B\ell}$. For every $a_1, ..., a_\ell, b_1, ..., b_\ell \in \{0,1\}^{B/2}$ the string $a_1\cdots a_\ell b_1\cdots b_\ell$ is in Π' if and only if it holds that $d_1(a_1,b_1)\cdots d_\ell(a_\ell,b_\ell) \in \Pi$. That is:

$$\Pi' \stackrel{\rm def}{=} \{a_1\cdots a_\ell b_1\cdots b_\ell : d_1(a_1,b_1)\cdots d_\ell(a_\ell,b_\ell) \in \Pi\}. \quad (5)$$

Claim 7.1.1 (relating Π' to Π): *Let* Π' *be as in* Eq. (5). *Then,* $\mathtt{Q}_\eta(\epsilon', \Pi') \leq B \cdot \mathtt{Q}_\eta(\epsilon', \Pi)$.

Proof: Basically, an ϵ'-tester for Π' can just emulate the execution of an ϵ'-tester for Π while answering each query $i \in [\ell]$ by reading the two corresponding $B/2$-bit long blocks in its oracle. Specifically, using an ϵ'-tester T for Π, we construct an tester for Π' that emulates the virtual oracle $d_1(a_1,b_1)\cdots d_\ell(a_\ell,b_\ell)$ for T by accessing its own oracle $a_1\cdots a_\ell b_1\cdots b_\ell$. Hence, each query of T is answered by making B oracle queries. Now, if $a_1\cdots a_\ell b_1\cdots b_\ell \in \Pi'$, then it must be that $d_1(a_1,b_1)\cdots d_\ell(a_\ell,b_\ell) \in \Pi$, and so our tester accepts (with probability at least 2/3). On the other hand, if $a_1\cdots a_\ell b_1\cdots b_\ell$ is ϵ'-far from Π', then $d_1(a_1,b_1)\cdots d_\ell(a_\ell,b_\ell)$ is ϵ'-far from Π, since otherwise it suffices to change less than $\epsilon\ell$ of the (a_i,b_i)-pairs in order to obtain a string in Π' (where here we use the hypothesis that d_i is onto). The claim follows. ■

Finally, consider $F' : \{0,1\}^{2B\ell} \to \{0,1\}^{B\ell}$ such that $F'(u,v) = u \oplus v$.

Claim 7.1.2 (relating Ψ' to Π'): *Let* Ψ' *be as in* Eq. (3)&(4) *and* Π' *be as in* Eq. (5). *Then:*

1. *For every* $(u,v) \in P' \cap S'$ *it holds that* $F'(u,v) \in \Pi'$.

2. *For every* $(u,v) \in P' \setminus S'$ *it holds that* $F'(u,v)$ *is* ϵ/B*-far from* Π'.

Proof: The key observation is that for every $(u,v) \in P'$, it holds that $u = u'0^{\ell B/2}$ and $v = 0^{\ell B/2}v'$, and so $F'(u,v) = u \oplus v = u'v'$. Furthermore, in that case there exists $(x,y) \in P$ such that $u' = g_1(x) \cdots g_\ell(x)$ and $v' = h_1(y) \cdots h_\ell(y)$. Using the hypothesis that the mapping $(x,y) \to (u,v)$ is one-to-one, we infer that this (x,y) is unique.

Now, if $(u,v) \in P' \cap S'$, then the aforementioned (x,y) must be in $P \cap S$, and it follows that $F(x,y) \in \Pi$ (by the theorem's hypothesis regarding F, see Footnote 10). It follows that $F'(u,v) = u'v' \in \Pi'$, because $u'v' = g_1(x) \cdots g_\ell(x)h_1(y) \cdots h_\ell(y)$ whereas for each $i \in [\ell]$ it holds that $d_i(g_i(x), h_i(y))$ is the i^{th} bit in $F(x,y)$.

Having established Item 1, we turn to Item 2: We observe that if $(u,v) \in P' \setminus S'$, then the aforementioned (x,y) must be in $P \setminus S$, and it follows that $F(x,y)$ is ϵ-far from Π (by the theorem's hypothesis regarding F). In this case, $F'(u,v) = g_1(x) \cdots g_\ell(x)h_1(y) \cdots h_\ell(y)$ such that for each $i \in [\ell]$ it holds that $d_i(g_i(x), h_i(y))$ is the i^{th} bit in $F(x,y)$. Recalling that $F(x,y)$ is ϵ-far from Π (i.e., $F(x,y)$ differs in at least $\epsilon\ell$ positions from any ℓ-bit string in Π), it follows that for at least $\epsilon\ell$ of the $i \in [\ell]$ at least one of the corresponding strings (i.e., $g_i(x)$ and $h_i(y)$) must be modified to place the $B\ell$-bit long string in Π'. Hence, $F'(u,v)$ is $(\epsilon\ell/B\ell)$-far from Π'. ■

This completes the proof for the aforementioned special case (i.e., of f_i's of the form $d_i(g_i(x), h_j(y))$), and we now turn to the more general case in which the bound B is guaranteed by arbitrary deterministic protocols (i.e., $B = \max_{i \in [\ell]}\{\mathtt{CC}_0(f_i)\}$).

Step 2: The case of deterministic protocols

Here we consider the case that $B = \max_{i \in [\ell]}\{\mathtt{CC}_0(f_i)\}$. We shall present a *communication problem $\Psi' = (P', S')$ such that $\mathtt{CC}_\eta(\Psi') \geq \mathtt{CC}_\eta(\Psi)$ and a property Π' such that $\mathtt{Q}_\eta(\epsilon', \Pi) \geq \mathtt{Q}_\eta(\epsilon', \Pi')/B$, whereas Ψ' and Π' are related as follows*:

1. *For every $(u,v) \in P' \cap S'$, it holds that $u \oplus v \in \Pi'$.*

2. *For every $(u,v) \in P' \setminus S'$, it holds that $u \oplus v$ is $\epsilon/2^B$-far from Π'.*

This means that whenever $\mathtt{Q}_\eta(\epsilon, \Pi) \geq \mathtt{CC}_\eta(\Psi)/B$ is established by Theorem 3.1, when using $B = \max_{i \in [\ell]}\{\mathtt{CC}_0(f_i)\}$, we can establish $\mathtt{Q}_\eta(\epsilon/2^B, \Pi) \geq \mathtt{CC}_\eta(\Psi)/2B$ by using the formulation as presented in [4] (cf. Corollary 3.3). Again, the parameters of the derived lower bound are somewhat weaker: The main issue is not that B is replaced by $2B$, but rather than the lower bound refers to a smaller value of the proximity parameter (i.e., $\epsilon/2^B$ rather than ϵ). The decrease in the value of the proximity parameter is far more acute than in the previous special case. In particular, in the case of $B > \log_2 \ell$ the alternative derivation only yields a result that refers to the query complexity of exact decision (since the value of the proximity parameter is smaller than $1/\ell$, where ℓ is the input length). This deficiency can be fixed by an idea that is presented in the treatment of the general case, which will follow (cf. Step 3).[12] But here we focus on the construction of Ψ' and Π' that satisfy the above (somewhat deficient) claim.

By the hypothesis, for every $i \in [\ell]$, there exists a deterministic two-party protocol of communication complexity at most B for computing f_i. Let A_i and B_i denote the corresponding strategies of the two parties, and let $A_i^x = A_i(x)$ and $B_i^y = B_i(y)$ denote the residual strategies for local inputs x and y, respectively. That is, $A_i^x(\gamma)$ denotes the answer of the first party, holding input x, to a message sequence γ sent by the second party (ditto for B_i^y).[13]

[12] Specifically, we refer to the use of the encoding of the parties' strategies by suitable error correcting codes.

[13] It is standard to assume that the parties interact by sending single-bit messages and that the first party starts. In such a case, A_i^x will be defined for strings of length at most $(B-1)/2$, including the empty string, while B_i^y will

We make the simplifying assumption that the mappings $x \mapsto (A_1^x, ..., A_\ell^x)$ and $y \mapsto (B_1^y, ..., B_\ell^y)$ are one-to-one, where the justification is that A_i^x may start by sending x_i (and ditto for B_i^y, with $\ell \geq n$ justified as in Step 1). Let $\langle A_i^x \rangle$ (resp., $\langle B_i^y \rangle$) denote a canonical 2^{B-1}-bit long description of the strategy A_i^x (resp., B_i^y) such that the value of $A_i^x(\gamma)$ (resp., $B_i^y(\gamma)$) appears in a specific bit location in $\langle A_i^x \rangle$ (resp., $\langle B_i^y \rangle$), where this location only depends on γ. Now, define $\Psi' = (P', S')$ such that

$$P' \stackrel{\rm def}{=} \{(\langle A_1^x \rangle \cdots \langle A_\ell^x \rangle 0^{2^{B-1}\ell}, 0^{2^{B-1}\ell} \langle B_1^y \rangle \cdots \langle B_\ell^y \rangle) : (x,y) \in P\} \tag{6}$$

$$S' \stackrel{\rm def}{=} \{(\langle A_1^x \rangle \cdots \langle A_\ell^x \rangle 0^{2^{B-1}\ell}, 0^{2^{B-1}\ell} \langle B_1^y \rangle \cdots \langle B_\ell^y \rangle) : (x,y) \in S\}. \tag{7}$$

Note that $\mathtt{CC}_\eta(\Psi) \leq \mathtt{CC}_\eta(\Psi')$, since a protocol for Ψ can proceed by emulating the protocol for Ψ'.

Let (α, β) be a pair of residual strategies (as considered above) for a two-party communication protocol. We say that (α, β) produce the bit σ if emulating the interaction between these strategies yields the (joint) outcome σ. The emulation proceeds by determining the first message sent according to α, then determining the response according to β, and so on.

Next, we introduce the property $\Pi' \subseteq \{0,1\}^{2^B \ell}$. For every $a_1, ..., a_\ell, b_1, ..., b_\ell \in \{0,1\}^{2^{B-1}}$ the string $a_1 \cdots a_\ell b_1 \cdots b_\ell$ is in Π' if and only if for every $i \in [\ell]$ it holds that (a_i, b_i) describes a pair of strategies that produce the output bit w_i and $w = w_1 \cdots w_\ell \in \Pi$. Denoting the bit produced by these descriptions by $\mathtt{P}(a_i, b_i)$, we have

$$\Pi' \stackrel{\rm def}{=} \{a_1 \cdots a_\ell b_1 \cdots b_\ell : \mathtt{P}(a_1, b_1) \cdots \mathtt{P}(a_\ell, b_\ell) \in \Pi\}. \tag{8}$$

Claim 7.1.3 (relating Π' to Π): *Let* Π' *be as in* Eq. (8). *Then,* $\mathtt{Q}_\eta(\epsilon', \Pi') \leq B \cdot \mathtt{Q}_\eta(\epsilon', \Pi)$.

Proof: Using an ϵ'-tester T for Π, we construct an ϵ'-tester for Π' by emulating the execution of T. Specifically, if T makes the query $i \in [\ell]$, then we access the $i^{\rm th}$ pair of strategies included in our own oracle, denoted z (i.e., for $z = a_1 \cdots a_\ell b_1 \cdots b_\ell$, this means accessing a_i and b_i). By making B queries to these strategies, we emulate the computation of the $i^{\rm th}$ bit in a virtual ℓ-bit string tested by T (i.e., the string $\mathtt{P}(a_1, b_1) \cdots \mathtt{P}(a_\ell, b_\ell)$). Specifically, we need only determine the value of the B bits that are exchanged in the interaction between a_i and b_i, rather than the full description of a_i and b_i. (Recall that each of these communicated bits appears as an explicit bit in the corresponding full description.)

Note that when given oracle access to $z = a_1 \cdots a_\ell b_1 \cdots b_\ell$, we emulate a computation of T by providing it with oracle access to the virtual string $\mathtt{P}(a_1, b_1) \cdots \mathtt{P}(a_\ell, b_\ell)$. Now, if $z \in \Pi'$, then (by definition) the corresponding virtual string is in Π. On the other hand, if z is ϵ'-far from Π', then the virtual string must be ϵ'-far from Π, because otherwise it suffices to modify less than $\epsilon' \ell$ pairs of strategies in order to produce a string in Π (which contradicts the hypothesis that z is ϵ'-far from Π'). ■

Finally, consider $F' : \{0,1\}^{2^{B+1}\ell} \to \{0,1\}^{2^B \ell}$ such that $F'(u,v) = u \oplus v$.

Claim 7.1.4 (relating Ψ' to Π'): *Let* Ψ' *be as in* Eq. (6)&(7) *and* Π' *be as in* Eq. (8). *Then:*

1. *For every* $(u,v) \in P' \cap S'$ *it holds that* $F'(u,v) \in \Pi'$.
2. *For every* $(u,v) \in P' \setminus S'$ *it holds that* $F'(u,v)$ *is* $\epsilon/2^B$*-far from* Π'.

be defined for $\bigcup_{j \in [B/2]} \{0,1\}^j$. In general, the situation may be more complex, but in all cases the length of the description of each of the two strategies is at most 2^{B-1}.

Proof: As in the proof of Claim 7.1.2, for every $(u,v) \in P'$ it holds that $u = u'0^{2^{B-1}\ell}$ and $v = 0^{2^{B-1}\ell}v'$, and so $F'(u,v) = u \oplus v = u'v'$. Also, in this case, there exists a unique $(x,y) \in P$ such that $u' = g_1(x) \cdots g_\ell(x)$ and $v' = h_1(y) \cdots h_\ell(y)$.

If $(u,v) \in P' \cap S'$, then the aforementioned (x,y) must be in $P \cap S$, and it follows that $F(x,y) \in \Pi$ (by the theorem's hypothesis regarding F). It follows that $F'(u,v) = u'v' \in \Pi'$, because $u' = \langle A_1^x \rangle \cdots \langle A_\ell^x \rangle$ and $v' = \langle B_1^y \rangle \cdots \langle B_\ell^y \rangle$, whereas for each $i \in [\ell]$ it holds that A_i^x and B_i^y produce the $i^{\rm th}$ bit in $F(x,y) \in \Pi$.

Having established Item 1, we turn to Item 2: If $(u,v) \in P' \setminus S'$, then the aforementioned (x,y) must be in $P \setminus S$, and it follows that $F(x,y)$ is ϵ-far from Π (by the theorem's hypothesis regarding F). In this case $F'(u,v) = \langle A_1^x \rangle \cdots \langle A_\ell^x \rangle \langle B_1^y \rangle \cdots \langle B_\ell^y \rangle$, where for each $i \in [\ell]$ it holds that A_i^x and B_i^y produce the $i^{\rm th}$ bit in $F(x,y)$. Recalling that $F(x,y)$ is ϵ-far from Π (i.e., $F(x,y)$ differs in at least $\epsilon\ell$ positions from any ℓ-bit string in Π), it follows that for at least $\epsilon\ell$ of the $i \in [\ell]$ at least one of the corresponding strategies (i.e., A_i^x and B_i^y) must be modified to place the $2^B\ell$-bit long string in Π'. Hence, $F'(u,v)$ is $(\epsilon\ell/2^B\ell)$-far from Π'. ■

This completes the proof for the special case of deterministic communication protocols for the f_i's. We now turn to the general case in which the bound B is guaranteed by arbitrary (randomized) protocols.

Step 3: The general case

Finally, we turn the general case, in which we are only guaranteed that $B = \max_{i\in[\ell]}\{\mathtt{CC}_{\eta/n}(f_i)\}$; that is, we have to deal with randomized protocols (of error probability at most η/n). The basic idea is to proceed as in Step 2, while using descriptions of residual randomized strategies, where a description of a residual randomized strategy consists of a sequence of descriptions of the corresponding residual deterministic strategies. This raises a difficulty, because not all possible descriptions (i.e., sequences) correspond to legitimate residual randomized strategies (since the descriptions may correspond to strategies that have higher error probability). Hence, some additional tests will be required when reducing the ϵ'-testing of the (modified) auxiliary property Π' to the ϵ'-testing of the (original) property Π. Specifically, we shall test that at least a $1 - \eta/n$ fraction of the pairs in the sequence produce the same bit.

Given the fact that additional tests are used, we seize the opportunity to also address a deficiency we have neglected in Steps 1 and 2 – the fact that we derived lower bounds for testing Π with smaller proximity parameters (i.e., ϵ/B and $\epsilon/2^B$, respectively). Our solution is to encode the aforementioned descriptions using an error correcting code that is locally testable (cf. [12, Def. 2.2]) and locally decodable (cf. [15]), where local decodability is essential for the emulation of the tester of the original property by a tester for the auxiliary property (because the original property refers to strings that appear in encoded form in the auxiliary property). Needless to say, local testability is essential for the testing of the modified Π', because this property contains certain sequences of codewords. Lastly, it is important that this code has constant relative distance, but its rate does not matter, and so we may just use the Hadamard code. We shall denote this code by C, and denote its relative distance by δ_C.

For starters, by the hypothesis, for every $i \in [\ell]$, there exists a randomized two-party protocol of communication complexity at most B for computing f_i. This protocol is in the shared randomness model, and we denote by ρ the length of the random string in use.[14] Let A_i and B_i denote the

[14] Indeed, we may assume (w.l.o.g., cf. [17, Thm. 3.14]) that $\rho \stackrel{\rm def}{=} O(\log n/\eta)$, but this is not needed for our argument.

corresponding strategies of the two parties, and let $A^x_{i,r} = A_i(x;r)$ and $B^y_{i,r} = B_i(y;r)$ denote the residual strategies for local inputs x and y and shared randomness $r \in \{0,1\}^\rho$. That is, $A^x_{i,r}(\gamma)$ denotes the answer of the first party, holding input x and viewing the shared randomness r, to a message sequence γ sent by the second party (ditto for $B^y_{i,r}$).

Let $\langle A^x_{i,r}\rangle$ (resp., $\langle B^y_{i,r}\rangle$) denote a canonical 2^{B-1}-bit long description of the strategy $A^x_{i,r}$ (resp., $B^y_{i,r}$), and let $\langle A^x_i\rangle$ (resp., $\langle B^y_i\rangle$) denote the 2^ρ-long sequence of corresponding codewords; that is, $\langle A^x_i\rangle \overset{\rm def}{=} (C(\langle A^x_{i,0^\rho}\rangle), ..., C(\langle A^x_{i,1^\rho}\rangle))$ and $\langle B^y_i\rangle \overset{\rm def}{=} (C(\langle B^y_{i,0^\rho}\rangle), ..., C(\langle B^y_{i,1^\rho}\rangle))$. We make the simplifying assumption (with justifications as in Step 2) that the mappings $x \mapsto (\langle A^x_1\rangle, ..., \langle A^x_\ell\rangle)$ and $y \mapsto (\langle B^y_1\rangle, ..., \langle B^y_\ell\rangle)$ are one-to-one. Using $L \overset{\rm def}{=} |\langle A^x_i\rangle| = 2^\rho \cdot |C(1^{2^{B-1}})|$, define $\Psi' = (P', S')$ such that

$$P' \overset{\rm def}{=} \{(\langle A^x_1\rangle \cdots \langle A^x_\ell\rangle 0^{\ell\cdot L}, 0^{\ell\cdot L}\langle B^y_1\rangle \cdots \langle B^y_\ell\rangle) : (x,y) \in P\} \tag{9}$$

$$S' \overset{\rm def}{=} \{(\langle A^x_1\rangle \cdots \langle A^x_\ell\rangle 0^{\ell\cdot L}, 0^{\ell\cdot L}\langle B^y_1\rangle \cdots \langle B^y_\ell\rangle) : (x,y) \in S\}. \tag{10}$$

Note that $\mathtt{CC}_\eta(\Psi) \le \mathtt{CC}_\eta(\Psi')$, since a protocol for Ψ can proceed by emulating the protocol for Ψ'.

As in Step 2, we say that the (residual) deterministic strategies α and β produce the bit $\sigma = \mathtt{P}(\alpha,\beta)$ if emulating the interaction between these strategies yields the (joint) outcome σ. We say that a sequence of such pairs safely produce the bit σ if at least a $1-\eta/n$ fraction of the pairs in the sequence produce this bit; that is, $\mathtt{SP}((\alpha_{0^\rho},\beta_{0^\rho}), ..., (\alpha_{1^\rho},\beta_{1^\rho})) = \sigma$ if $|\{r \in \{0,1\}^\rho : \mathtt{P}(\alpha_r,\beta_r) = \sigma\}| \ge (1-\eta/n)\cdot 2^\rho$.

Next, we introduce the property $\Pi' \subseteq \{0,1\}^{2\ell L}$. Loosely speaking, Π' will contain sequences of C-codewords that each encode ℓ sequences of pairs such that the $i^{\rm th}$ sequence *safely* produces the $i^{\rm th}$ bit of an ℓ-bit string in Π. Namely, for every sequence $(a_1, ..., a_\ell, b_1, ..., b_\ell)$ such that $a_i = (a_{i,0^\rho}, ..., a_{i,1^\rho}) \in \{0,1\}^{2^\rho\cdot 2^{B-1}}$ and $b_i = (b_{i,0^\rho}, ..., b_{i,1^\rho}) \in \{0,1\}^{2^\rho\cdot 2^{B-1}}$, the corresponding $2\ell\cdot L$-bit long string $c_1 \cdots c_{2\ell}$ is in Π' if and only if the following conditions hold:

1. For every $i \in [\ell]$, it holds that $c_i = C(a_{i,0^\rho}) \cdots C(a_{i,1^\rho})$ and $c_{\ell+i} = C(b_{i,0^\rho}) \cdots C(b_{i,1^\rho})$.
2. For every $i \in [\ell]$, the sequence of pairs $(a_{i,r}, b_{i,r})_{r\in\{0,1\}^\rho}$ safely produce a bit w_i such that $w_1 \cdots w_\ell \in \Pi$.

That is:

$$\Pi' \overset{\rm def}{=} \left\{ c_{1,0^\rho}\cdots c_{2\ell,1^\rho} : \begin{array}{c} \exists a_{1,0^\rho}, ..., a_{\ell,1^\rho}, b_{1,0^\rho}, ..., b_{\ell,1^\rho} \in \{0,1\}^{2^{B-1}} \text{ s.t.} \\ (1)\quad \forall i\in[\ell]\,\forall r\in\{0,1\}^\rho \;\; C(a_{i,r}) = c_{i,r} \wedge C(b_{i,r}) = c_{\ell+i,r} \\ (2)\quad \exists w_1\cdots w_\ell \in \Pi \;\; \forall i\in[\ell] \;\; \mathtt{SP}((a_{i,0^\rho}, b_{i,0^\rho}), ..., (a_{i,1^\rho}, b_{i,1^\rho})) = w_i \end{array} \right\}, \tag{11}$$

Claim 7.1.5 (relating Π' to Π): *Let Π' be as in* Eq. (11). *For every $\eta \ge \epsilon' = \widetilde{\Omega}(1/n)$ it holds that $\mathtt{Q}_\eta(\epsilon', \Pi') = \widetilde{O}(B/\epsilon') + \widetilde{O}(B\cdot \mathtt{Q}_\eta(\epsilon'/2, \Pi))$, where the polylogarithmic factor hidden in the second $\widetilde{O}$-notation is $O(\log(B\cdot \mathtt{Q}_\eta(\epsilon'/2,\Pi)/\eta))\cdot \log B)$.*

Proof: Unlike the proofs of Claims 7.1.1 and 7.1.3, here ϵ'-testing Π' does not reduce to merely emulating an ϵ'-tester for Π, because here strings in Π' have additional structure – they are sequences of codewords that encode pairs that safely produce some bits. Thus, in addition to emulating an $\epsilon'/2$-tester for Π, we would also perform codeword tests and consistency (i.e., "safe production") tests. We start by describing these new testing activities, while letting $n_C = |C(1^{2^{B-1}})|$ denote the length of the codewords in C.

On input $z = (c_1, ..., c_{2\ell \cdot 2^\rho})$, with each $c_i \in \{0,1\}^{n_C}$, we first check that this sequence is $\epsilon'/4$-close to a sequence of codewords of C. This can be done at a cost of $\widetilde{O}(1/\epsilon')$ queries, by selecting, for each $j \in [\lceil \log_2(8/\epsilon') \rceil]$, a random sample of $O(2^j \log(1/\epsilon'))$ indices $I \subseteq [2\ell \cdot 2^\rho]$ and performing an $2^j\epsilon'$-test (with error probability $\mathrm{poly}(\epsilon')$) on c_i for each $i \in I$. The (strong) local testablity of the code C asserts that ϵ''-testing its codewords with error probability 2^{-k} can be done by using $O(k/\epsilon'')$ queries.

Let $a_{1,0^\rho}, ..., a_{\ell,1^\rho}, b_{1,0^\rho}, ..., b_{\ell,1^\rho} \in \{0,1\}^{2^{B-1}}$ be such that $C(a_{1,0^\rho}) \cdots C(a_{\ell,1^\rho}) C(b_{1,0^\rho}) \cdots C(b_{\ell,1^\rho})$ is closest to z. We now check that the sequence of $a_{i,r}$'s and $b_{i,r}$'s is $\epsilon'/4$-close to a sequence that safely produces ℓ bits (i.e., one bit per each value of $i \in [\ell]$), by selecting a sample of i's, taking a sample of r's for each i, and checking that the pairs $(a_{i,r}, b_{i,r})$ produce the same value for each such i. (For each $j \in [\lceil \log_2(8/\epsilon') \rceil]$, we select a random sample of $O(2^j \log(1/\epsilon'))$ indices $I \subseteq [\ell]$ and take a sample of $O(1/(2^j\epsilon'))$ choices of $r \in \{0,1\}^\rho$ for each $i \in I$.)[15] The aforementioned checking is performed while employing local decodability of the relevant bits (in the description of the strategy). We use local decodability with error probability $\mathrm{poly}(\epsilon'/B)$ (which is guranteed to work up to relative distance $\delta_C/3$, where δ_C denotes the relative distance of the code C). Furthermore, each of these invocations of the local decodability procedure will also run an $(\delta_C/3)$-tester for C-codewords (again, with error probability $\mathrm{poly}(\epsilon'/B)$), and the tester (for Π') will reject whenever any invocation of the codeword tester rejects. Hence, each pair (i, r) that we check generates $O(B \log(B/\epsilon'))$ queries, whereas we check $\widetilde{O}(1/\epsilon')$ such pairs.

Finally, we get to emulate the execution of the $\epsilon'/2$-tester for Π, denoted T. Specifically, if T makes the query $i \in [\ell]$, then we access the i^{th} pair of sequences (which is typically close to $(C(a_{i,r}))_{r\in\{0,1\}^\rho}$ and $(C(b_{i,r}))_{r\in\{0,1\}^\rho}$), and try to recover the answer by self-correction with error probability $\eta/O(Bq)$, where q is the query complexity of T. This self-correction procedure combines a self-correction for the bit produced by the pairs $(a_{i,r}, (b_{i,r})_{r\in\{0,1\}^\rho}$, which in turn relies on local decodability of the relevant bits in the descriptions of the sampled $(a_{i,r}, b_{i,r})$-pairs. We also check whether these sequences are $1/4$-close to safely produce this answer (bit), and each such check is also performed with error probability $\eta/O(Bq)$. This means that each query of T is emulated by using $O(B \cdot \log(Bq/\eta) \cdot \log B)$ queries, since we use the codeword tester and decoder with error probability $1/O(B)$ (while using constant proximity parameter in the testing).

If $z \in \Pi'$, then (by definition, cf. Eq. (11)) the string z is a concatenation of codewords that encode pairs that safely produce the bits of some $w \in \Pi$. Noting that when T is invoked, all its queries are answered (with high probability) by the corresponding bits of this w, it follows that our tester accepts (with high probability).[16] On the other hand, if z is ϵ'-far from Π', then at least one of the following cases must hold:

1. Either z is $\epsilon'/4$-far from a sequence of C-codewords;

2. or z is C-decodable to a sequence $(a_{1,0^\rho}, ..., a_{\ell,1^\rho}, b_{1,0^\rho}, ..., b_{\ell,1^\rho})$ that is $\epsilon'/4$-far from safely producing bits of some ℓ-bit long;

3. or the string w that the foregoing sequence (safely) produces is $\epsilon'/2$-far from Π.

As argued next, in each of these cases, we reject with high probability. For Case 1 this follows from the various codeword tests that are performed, since in this case there exists an integer

[15] Since $\epsilon' = \widetilde{\Omega}(1/n)$, we do not expect to see pairs that produces the opposite value, which is quite rare (i.e., appears in at most a η/n fraction of the pairs).

[16] Note that we may also reject, with very small probability, due to encoutering pairs that produce different values (within a sequence of pairs that safely produces a value). But since the fraction of exceptional pairs is at most η/n, this event occurs with very small probability..

$j \in [\lceil \log_2(4/\epsilon') \rceil]$ such that at least a $1/O(2^j \log(1/\epsilon'))$ fraction of the (n_C-bit long) blocks are $2^j\epsilon'$-far from the code C. Assuming that Case 1 does not hold, we consider the foregoing sequence $(a_{1,0^\rho}, ..., a_{\ell,1^\rho}, b_{1,0^\rho}, ..., b_{\ell,1^\rho})$, and what happens when Case 2 holds. In this case, with very high probability, we either detect pairs $(a_{i,r}, b_{i,r})$ and $(a_{i,r'}, b_{i,r'})$ that produce different values (via the self-correction) or detect corresponding blocks that are $\delta_C/3$-far from the code C. Finally, assuming that Cases 1 and 2 do not hold, we consider the foregoing ℓ-bit string w that the said sequence produces. In this case, we either detect a problem when emulating T (i.e., indices $i \in [\ell]$ that correspond to bits that are 1/4-far from being safely produced, or blocks that are $\delta_C/3$-far from C-codewords) or we complete an emulation of T^w, which rejects (with high probability). The claim follows. ■

Finally, consider $F' : \{0,1\}^{4\ell L} \to \{0,1\}^{2\ell L}$ such that $F'(u, v) = u \oplus v$.

Claim 7.1.6 (relating Ψ' to Π'): *Let* Ψ' *be as in* Eq. (9)&(10) *and* Π' *be as in* Eq. (11). *Then:*

1. *For every* $(u, v) \in P' \cap S'$ *it holds that* $F'(u, v) \in \Pi'$.

2. *For every* $(u, v) \in P' \setminus S'$ *it holds that* $F'(u, v)$ *is* $\Omega(\epsilon)$*-far from* Π'.

Proof: As in the proofs of Claims 7.1.2 and 7.1.4, for every $(u, v) \in P'$ it holds that $u = u'0^{\ell L}$ and $v = 0^{\ell L}v'$, and so $F'(u, v) = u \oplus v = u'v'$. Also, in that case there exists a unique $(x, y) \in P$ such that $u' = \langle A_1^x \rangle \cdots \langle A_\ell^x \rangle$ and $v' = \langle B_1^y \rangle \cdots \langle B_\ell^y \rangle$.

If $(u, v) \in P' \cap S'$, then the aforementioned (x, y) must be in $P \cap S$, and it follows that $F(x, y) \in \Pi$ (by the theorem's hypothesis regarding F). It follows that $F'(u, v) = u'v' \in \Pi'$, because for each $i \in [\ell]$ it holds that $\langle A_i^x \rangle$ and $\langle B_i^y \rangle$ encode a sequence of pairs that safely produce the i^{th} bit in $F(x, y) \in \Pi$.

Having established Item 1, we turn to Item 2: If $(u, v) \in P' \setminus S'$, then the aforementioned (x, y) must be in $P \setminus S$, and it follows that $F(x, y)$ is ϵ-far from Π (by the theorem's hypothesis regarding F). In this case for each $i \in [\ell]$ it holds that $\langle A_i^x \rangle$ and $\langle B_i^y \rangle$ encode a sequence of pairs that safely produce the i^{th} bit in $F(x, y)$. Recalling that $F(x, y)$ is ϵ-far from Π (i.e., $F(x, y)$ differs in at least $\epsilon\ell$ positions from any ℓ-bit string in Π), it follows that for at least $\epsilon\ell$ of the $i \in [\ell]$ either $\langle A_i^x \rangle$ or $\langle B_i^y \rangle$ should be modified such that the encoded sequences safely produce a different value for the i^{th} bit. Recalling that each of the above is a sequence of 2^ρ codewords and that a vast majority of the C-decodable pairs produce the current value (of this bit), it follows that we need to change more than half of these codewords. Since the code C has (constant) relative distance δ_D, this means that we need to change at least $2^{\rho-1} \cdot \delta_C n_C = \Omega(L)$ bits per each such i, which implies that $F'(u, v)$ is $\Omega(\epsilon\ell L/2\ell L)$-far from Π'. The claim follows. ■

Comment: Step 3 can be carried out for $B = \max_{i\in[\ell]}\{\mathtt{CC}_{1/3}(f_i)\}$, at the cost of an additive overhead of $\widetilde{O}(B/(\epsilon')^2)$ (rather than $\widetilde{O}(B/\epsilon')$) in Claim 7.1.5. In light of this fact, it seems fair to reconsider the comparison made right after stating Theorem 7.1. In this case (i.e., starting with $B = \max_{i\in[\ell]}\{\mathtt{CC}_{1/3}(f_i)\}$), applying Theorem 3.1 requires to perform error-reduction first (i.e., use $\mathtt{CC}_{\eta/n}(f_i) = O(\mathtt{CC}_{1/3}(f_i) \cdot \log(n/\eta))$). Actually, for $C = \mathtt{CC}_{2\eta}(\Psi)$, we can use $\mathtt{CC}_{\eta/C}(f_i) = O(\mathtt{CC}_{1/3}(f_i) \cdot \log(C/\eta))$, since the proof of Theorem 3.1 holds also for $B = \max_{i\in[\ell]}\{\mathtt{CC}_{\eta/C}(f_i)\}$. In this case, for every fixed $\eta > 0$, we get $\mathtt{Q}_\eta(\epsilon, \Pi) \geq C/O(B \log C)$ by using the general formulation, which is closer to the *rough bound*[17] of $\mathtt{Q}_\eta(\Omega(\epsilon), \Pi) \geq C/O(B \log BC)$ that we get by the restricted formulation.

[17] Indeed, this rough bound neglects the aforementioned additive terms, which are insignificant for constant $\epsilon > 0$.

8 Conclusions

As demonstrated in Section 4, using the general formulation provided in Theorem 3.1 frees the user from the need to introduce *auxiliary* communication complexity problems as a bridge between known communication complexity problems and property testing problems. Recall that these auxiliary problems are needed because it is not clear how to directly reduce the original communication complexity problems (for which lower bounds are known) to the targeted property testing problems when using simple combining operators (as in [4], cf. Corollary 3.3).[18] In contrast, such direct reductions are easy to design when using the general formulation of Theorem 3.1. This phenomenon is not specific to the examples presented Section 4: In fact, it seem to arise in all known applications of the communication complexity methodology (starting from [4] itself).

We believe that the simpler it is to apply a methodology, the more useful the methodology becomes. Work should be shifted from the user (of the methodology) to the methodology itself (or rather to the proof of its validity). We believe that this is done by moving from the restricted formulation of [4] (cf. Corollary 3.3) to the general formulation of Theorem 3.1. The shifting of work is evident when trying to emulate results obtained via the general formulation by the restricted one, as done in Section 7. Indeed, we believe that the results of Section 7 demonstrate that while the general formulation is not much more powerful (as far as the obtainable lower bounds are concerned), it may be far easier to use (e.g., since the emulations that we found are quite imposing).

Acknowledgments

Part of this work is based on joint research with Dana Ron, who refused to co-author it. The constructions presented in Section 4 are partially inspired by some constructions in [13]. We are grateful to David Woodruff for suggesting Theorem 5.2 and allowing us to present it here. Ditto with respect to Tom Gur for Theorem A.2. Lastly, we thank Tom Gur and Ron Rothblum for comments on an early draft of this paper.

References

[1] N. Alon, S. Dar, M. Parnas, and D. Ron. Testing of clustering. SIAM Journal on Discrete Math., Vol. 16 (3), pages 393–417, 2003.

[2] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Science*, Vol. 58 (1), pages 137–147, 1999.

[3] E. Ben-Sasson, P. Harsha, and S. Raskhodnikova. Some 3CNF Properties are Hard to Test. *SIAM Journal on Computing*, Vol. 35 (1), pages 1–21, 2005.

[4] E. Blais, J. Brody, and K. Matulef. Property Testing Lower Bounds via Communication Complexity. *Computational Complexity*, Vol. 21 (2), pages 311–358, 2012.

[5] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron and A. Samorodnitsky. Improved Testing Algorithms for Monotonicity. In the proceedings of *3rd RANDOM*, Springer LNCS, Vol. 1671, pages 97–108.

[18] Instead, one reduces the original communication complexity problem to the auxiliary one, and then reduces the auxiliary communication problem to the property testing problem. The first reduction is performed within the setting of communication complexity, whereas the second reduction is the one in which simple combing operators are used.

[6] B. Chor and O. Goldreich. Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity. *SIAM Journal on Computing*, Vol. 17 (2), pages 230–261, 1988.

[7] O. Goldreich, editor. *Property Testing – Current Research and Surveys.* Lecture Notes in Computer Science, Vol. 6390, Springer, 2010.

[8] O. Goldreich. Short Locally Testable Codes and Proofs: A Survey in Two Parts. In [7].

[9] O. Goldreich. On Testing Computability by Small Width OBDDs. In the proceedings of *14th RANDOM*, Springer LNCS, Vol. 6302, pages 574–586, 2010.

[10] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, pages 653–750, July 1998. Extended abstract in *37th FOCS*, 1996.

[11] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, pages 302–343, 2002.

[12] O. Goldreich and M. Sudan. Locally testable codes and PCPs of almost linear length. *Journal of the ACM*, Vol. 53, No. 4, July 2006, pp. 558–655.

[13] T. Gur and R. Rothblum. Non-Interactive Proofs of Proximity. *ECCC*, TR13-078, May 2013.

[14] B. Kalyanasundaram and G. Schintger. The probabilistic communication complexity of set intersection. SIAM Journal on Discrete Math., Vol. 5 (4), pages 545–557, 1992.

[15] J. Katz and L. Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proc. 32nd ACM Symposium on the Theory of Computing*, pages 80–86, 2000.

[16] M. Kearns and D. Ron. Testing problems with sub-learning sample complexity. *Journal of Computer and System Science*, Vol. 61 (3), pages 428–456, 2000.

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

[18] M. Parnas and D. Ron. Testing the diameter of graphs. *Random Structures and Algorithms*, Vol. 20 (2), pages 165–183, 2002.

[19] D. Ron. Property testing: A learning theory perspective. *Foundations and Trends in Machine Learning*, Vol. 1 (3), pages 307–402, 2008.

[20] D. Ron. Algorithmic and analysis techniques in property testing. *Foundations and Trends in TCS*, Vol. 5 (2), pages 73–205, 2009.

[21] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2), pages 252–271, 1996.

Appendix: Generalization to Multi-Party Communication

The formulation presented in Section 3 generalizes easily to the model of multi-party communication. The treatment is quite oblivious to the details of the model; for example, it does not matter if one considers the standard model of "input on the forehead" or to the more naive model in which each party gets a part of the input (with no overlap). (These variations can be captured by the promise problems that the parties wish to solve.) The exact way in which the parties communicate is also not crucial, at least as long as the number of parties (denoted m) is small. For simplicity, we consider here a broadcast model, where in each communication round there is a single designated sender (determined by the transcript of the communication so far).

In light of the above, we shall consider m-party communication protocols in which the local input of the j^{th} party is denoted $x^{(j)}$. We denote by $\langle A^{(1)}(x^{(1)}), ..., A^{(m)}(x^{(m)})\rangle(r)$ the (joint) output of the m parties, when the j^{th} party uses strategy $A^{(j)}$ and gets input $x^{(j)}$, and all parties have free access to the shared randomness r. Considering promise problems $\Psi = (P, S)$ such that $P, S \subseteq \{0,1\}^{m \cdot n}$, Definition 2.1 extends naturally; that is, the η-error communication complexity of Ψ, denoted $\texttt{CC}_\eta(\Psi)$, is the minimum communication complexity of all m-protocols that solve Ψ with error at most η.

Theorem A.1 (Theorem 3.1, generalized to m-party protocols): *Let $\Psi = (P, S)$ be a promise problem such that $P, S \subseteq \{0,1\}^{m \cdot n}$, and let $\Pi \subseteq \{0,1\}^\ell$ be a property, and $\epsilon, \eta > 0$. Suppose that the mapping $F : \{0,1\}^{m \cdot n} \to \{0,1\}^\ell$ satisfies the following two conditions:*

1. *For every $(x^{(1)}, ..., x^{(m)}) \in P \cap S$, it holds that $F(x^{(1)}, ..., x^{(m)}) \in \Pi$.*
2. *For every $(x^{(1)}, ..., x^{(m)}) \in P \setminus S$, it holds that $F(x^{(1)}, ..., x^{(m)})$ is ϵ-far from Π.*

Then, $\texttt{Q}_\eta(\epsilon, \Pi) \geq \texttt{CC}_{2\eta}(\Psi)/B$, where $B = \max_{i \in [\ell]}\{\texttt{CC}_{\eta/n}(f_i)\}$ and $f_i(x, y)$ is the i^{th} bit of $F(x, y)$. Furthermore, if $B = \max_{i \in [\ell]}\{\texttt{CC}_0(f_i)\}$, then $\texttt{Q}_\eta(\epsilon, \Pi) \geq \texttt{CC}_\eta(\Psi)/B$.

Theorem A.1 is proved by a straightforward generalization of the proof of Theorem 3.1; that is, we merely replace "two" by "m" (and everything goes through). We believe that this generalization further clarifies the ideas underlying the proof of Theorem 3.1 by presenting them in a slightly more abstract form.

Proof: The following description applies to any communication model in which all parties obtain the output produced by the protocol. Given an ϵ-tester with error η for Π and communication protocols for the f_i's, we present a protocol for solving Ψ. The key idea is that, using their shared randomness, the parties (holding the inputs $x^{(1)}, ..., x^{(m)}$, respectively) can emulate the execution of the ϵ-tester, while providing it with virtual access to $F(x^{(1)}, ..., x^{(m)})$. Specifically, when the tester queries the i^{th} bit of the oracle, the parties provide it with the value of $f_i(x^{(1)}, ..., x^{(m)})$ by first executing the corresponding communication protocol.

The protocol for Ψ proceeds as follows: On local input $x^{(j)}$ and shared randomness $r = (r_0, r_1, ..., r_\ell) \in (\{0,1\}^*)^{\ell+1}$, the j^{th} party invokes the ϵ-tester on randomness r_0, and answers the tester's queries by interacting with the other parties. That is, each of the parties invokes a local copy of the tester's program, but all copies are invoked on the same randomness, and are fed with identical answers to their (identical) queries. When the tester issues a query $i \in [\ell]$, the parties compute the value of $f_i(x^{(1)}, ..., x^{(m)})$ by using the corresponding communication protocol, and feed $f_i(x^{(1)}, ..., x^{(m)})$ to (their local copy of) the tester. Specifically, denoting the latter protocol (i.e., sequence of strategies) by $(A_i^{(1)}, ..., A_i^{(m)})$, the parties answer with $\langle A_i^{(1)}(x^{(1)}), ..., A_i^{(m)}(x^{(m)})\rangle(r_i)$.

When the tester halts, each party outputs the output it has obtained from (its local copy of) the tester.

We stress that the above description is oblivious to the details of the communication model, as long as in this model all parties obtain the output produced by the protocol.[19] Indeed, the description presented in the proof of Theorem 3.1 is merely a special case (which corresponds to the standard model of two-party computation), and the analysis of the general case (presented here) is identical to the analysis of the special case presented in the proof of Theorem 3.1. ■

On the potential usefulness of the generalization. Tom Gur has pointed out that the generalization to multi-party communication complexity allows additional flexibility for the design of reductions. To illustrate the point, he suggested the proof outlined below, which refers to a multi-party communication complexity model in which parties obtain non-overlapping inputs and communication is by individual point-to-point channels.

Theorem A.2 (a property testing (encoded) version of the frequency moment problem of [2]):[20] *For $k(n) = n/2$ and $\ell(n) = n^{1+o(1)}$, let Σ be a finite field of size n, and $C : \Sigma^{k(n)} \to \Sigma^{\ell(n)}$ be a Σ-linear code of constant relative distance, denoted δ. For any sequence $\sigma = (\sigma_1, ..., \sigma_k) \in \Sigma^k$ and $v \in \Sigma$, let $\#_v(\sigma)$ denote the number of occurrences of v in σ; that is, $\#_v(\sigma) = |\{i \in [k] : \sigma_i = v\}|$. For any constant $c > 1$, let*

$$\Pi = \left\{ C(x) : x \in \Sigma^{k(n)} \wedge \sum_{v \in \Sigma} \#_v(x)^c = k(n) \right\} \tag{12}$$

$$\Pi' = \left\{ C(x) : x \in \Sigma^{k(n)} \wedge \sum_{v \in \Sigma} \#_v(x)^c \leq 2k(n) \right\}. \tag{13}$$

Then, distinguishing inputs in Π from inputs that are δ-far from Π' requires $\Omega(\ell(n)^{1-(7/c)})$ queries.

Indeed, it follows that testing Π requires query complexity $\Omega(\ell(n)^{1-(7/c)})$, but this (and, in fact, a stronger $\Omega(n/\log n)$ lower bound) can be proved by reduction from a two-party communication complexity problem (i.e., DISJ).[21] In contrast, Theorem A.2 refers to a doubly-relaxed decision problem, where one level of relaxation is the approximation of the norm (captured by the gap between Π and Π') and the second is the standard property testing relaxation (captured by the gap between Π' and δ-far from Π'). Such doubly-relaxed problems have been often considered in

[19] If only a designated subset of the parties obtains the output, then we can emulate only nonadaptive testers (as done in Section 5).

[20] The following problem differs from the one in [2] in two aspects. Firstly, the computational model is different (i.e., we consider the query complexity of property testing, whereas [2] refers to the space complexity of streaming algorithms). Secondly, the problems are different: We consider an error-correcting encoding (i.e., $C(x)$) of the information (i.e., x) to which the frequency measure is applied. We stress, however, that the lower bound is not due to the complexity of codeword testing, since codeword testing may be easy for $\ell(n) = k(n)^{1+o(1)}$ (cf., e.g., [12]).

[21] Indeed, this follows from the proof of Theorem A.2, when setting $m = 2$, which correspond to the two-party case, and observing that NO-instances are mapped to instances having norm at least $m^c + (t-1)m > tm = k(n)$. Note that the same lower bound can be proved for Π', by padding the inputs to DISJ with $k(n)^{1/c}$ fixed elements (taken from a disjoint set of symbols). Note that these arguments rely on the fact that testing Π (or Π') requires distinguishing codewords that encode information (i.e., x) with a norm below some threshold from codewords that encode information with norm just above that threshold. In contrast, Theorem A.2 refers to a relaxation that captures an approximation of the corresponding norm, and a straightforward adaptation of the reduction from the two-party case does not seem to work here.

the property testing literature (cf., e.g., [18, 1]), starting with [16]. The following proof, which adapts a proof of [2] (which in turn refers to streaming algorithms), relies on a reduction from a multi-party communication problem. As is the case with its streaming original [2], it is not clear whether Theorem A.2 can be proved by reduction from a two-party communication problem.

Proof: We shall use a reduction from the following multi-party communication problem, denoted (m,t)-DISJ$_n$. In this problem, there are m parties, each holding a t-subset of $[n]$, and the problem is to distinguish the case that the subsets are pairwise disjoint from the case that the intersection of all subsets is non-empty. By [2], if $n \geq 2mt - m + 1$, then the communication complexity of (m,t)-DISJ$_n$ (in the point-to-point channels model) is $\Omega(t/m^3)$.[22]

We set $m = n^{1/c}$ and $t = n/2m$ (so that $k(n) = mt$), and represent the input of the $j^{\rm th}$ party by a sequence $x^{(j)} \in \Sigma^t$. Then, we let $F(x^{(1)},...,x^{(m)}) = C(x^{(1)} \cdots x^{(m)})$, which equals $\sum_{j\in[m]} C(0^{(j-1)t}x^{(j)}0^{(m-j)t})$. Hence, each bit of $F(x^{(1)},...,x^{(m)})$ can be computed (in this communication model) by communicating $m^2 \log_2 n$ bits (i.e., each party sends a single field elements to each of the other parties). Note that if $(x^{(1)},...,x^{(m)})$ is a YES-instance of (m,t)-DISJ$_n$ then $\sum_{v\in\Sigma} \#_v(C(x))^c = mt = k(n)$, which means that $F(x^{(1)},...,x^{(m)})$ is in Π, whereas if $(x^{(1)},...,x^{(m)})$ is a NO-instance of (m,t)-DISJ$_n$ then $\sum_{v\in\Sigma} \#_v(C(x))^c > m^c = n = 2k(n)$, which means that $F(x^{(1)},...,x^{(m)})$ is not in Π', and so is δ-far from any codeword in Π'. Applying Theorem A.1,[23] it follows that the query complexity of the promise problem of distinguishing Π from the set of $\ell(n)$-long sequences that are δ-far from Π' is lower bounded by $\Omega(t/m^3)/(m^2 \log n)$, which equals $\Omega(n/(m^6 \log n)) = \Omega(n^{1-6(1+o(1))/c})$. Using $n = \ell(n)^{1/(1+o(1))}$, the claim follows. ■

[22] The result of [2] is actually stronger, since it refers to the case that the NO-instances consist of subsets that have pairwise intersections that all equal the same singleton.

[23] Actually, we need to generalize Theorem A.1 so that it applies to doubly-relaxed problems. Such a generalization is straightforward.