

On the complexities of Interactive Proofs of Proximity

Noga Amit*

Oded Goldreich†

October 29, 2026

Abstract

The plural in the title hints that there are several complexities involved here. The most traditional one is the computational complexity of the verifier; but, in the context of Interactive Proofs of Proximity (IPPs), the query complexity of the verifier and the communication complexity of the system should take the front seat. Hence, we avoid limiting the computational complexity of the verifier, since this limitation blurs the role of the main complexity measures that should be of interest in the case of IPPs.

Specifically, we consider IPPs in which the verifier strategy is only restricted by its query complexity and its communication complexity. While we do not limit the computational complexity of the verifier, we do care about the computational complexity of the properties for which we prove a query+communication lower bound. Our results present a trade-off between the computational complexity of the properties for which query+communication lower bounds are asserted and complexity assumptions that condition these results. Specifically, letting q and c denote the query and communication complexities of IPPs (and letting n denote the input length), we essentially prove

1. An unconditional lower bound of $q(n) + c(n) = \Omega(n/\log n)$ for properties that are decidable in double-exponential time.
2. Assuming that exponential-time (resp., $\mathcal{NP}$) contains a set that does not have (non-uniform) small-exponential circuit size, we obtain a $q(n) + c(n) = \Omega(n/\log n)$ lower bound for properties that are decidable in exponential-time (resp., in $\mathcal{NP}$).
3. Assuming the existence of pseudorandom generators that fool small-exponential size circuits, we obtain a lower bound of $q(n) + c(n) = \Omega(\sqrt{n/\log n})$ for properties that are decidable in $\mathcal{P}$.

In contrast, recall that any property in uniform- $\mathcal{NC}$ has an IPP with $q(n) + c(n) = \tilde{O}(\sqrt{n})$. Hence, the almost linear lower bounds of the first two items are unlikely to be proven unconditionally for properties in $\mathcal{NP}$.

In the foregoing text, $c(n)$ denotes the total number of bits exchanged between the two parties (in both directions). One may wonder whether similar bounds hold when considering only the number of bits communicated by the prover, denoted $p(n)$. We show that this is essentially the case, by proving that the verifier gains almost nothing by sending messages of length exceeding $O(q(n) \cdot \log n)$. Specifically, we show that if Π has an $r(n)$ -round IPP in which the prover sends at most $p(n)$ bits and the verifier makes at most $q(n)$ queries, then Π has an $r(n)$ -round IPP of query complexity $O(q(n))$ and communication complexity $O(p(n) + r(n) \cdot q(n) \cdot \log n)$.

*MIT, Cambridge, MA, USA.

†Weizmann Institute of Science, Rehovot, ISRAEL.

Contents

1	Introduction	1
1.1	Overview	1
1.2	Discussion	3
1.3	Techniques	4
1.4	Organization	5
2	Preliminaries: IPPs and their complexity measures	5
3	Unconditional results	7
4	Conditional results using properties in either $\mathcal{NP}$ or $\mathcal{E}$	9
5	Conditional results using properties in $\mathcal{P}$ (Part 1)	13
5.1	Reproving the lower bound of [18] and going beyond	13
5.2	A stronger result for tolerant IPPs	15
6	Conditional results using properties in $\mathcal{P}$ (Part 2)	17
7	Compressing the verifier's messages	18
7.1	Compressing distributions	18
7.2	Proof of Theorem 1.4	23
	Appendix A: An alternative proof of Part of Theorem 4.1	28
	Appendix B: Claims 7.2 and 7.3 imply one another	28
	References	31

1 Introduction

Interactive Proofs of Proximity (IPPs), introduced by Rothblum, Vadhan, and Wigderson [21], are a hybrid of property testing and interactive proofs. In these systems, the tester is replaced by a verifier, who can communicate with an untrusted prover, who (unlike the verifier) has free access to the input. Analogously to testers in the context of property testing, for a property (or set) Π , the verifier is required to accept inputs in Π and reject inputs that are *far from* Π (see detailed definitions in Section 2).

The plural in the title hints at the fact that several complexities are involved in the study of Interactive Proofs of Proximity (IPPs). The most traditional complexity measure is the computational complexity of the verifier (i.e., its running-time), but in the context of IPPs the query complexity of the verifier and the communication complexity of the system should take front seat (see further discussion in Section 1.2). Hence, we avoid limiting the computational complexity of the verifier, since this limitation blurs the role of the main complexity measures that should be of interest in the case of IPPs (i.e., the query complexity of the verifier and the communication complexity of the system). The latter point is articulated in Section 3.

1.1 Overview

We consider IPPs in which the verifier strategy is only restricted by its query complexity and its communication complexity. For simplicity, we fix the proximity parameter (i.e., let $\epsilon = \Omega(1)$). Letting q and c denote the query and communication complexities of IPPs (and letting n denote the input length), our first result asserts the existence of properties that do not have IPPs satisfying $q(n) + c(n) = o(n/\log n)$. (Indeed, following the common tradition in the area, we state all lower bounds as a function of n , while referring to a sufficiently small constant value of the proximity parameter.)

Theorem 1.1 (unconditional lower bound): *There exists a property Π such that every IPP for Π must satisfy $q(n) + c(n) = \Omega(n/\log n)$. Furthermore, Π is decidable in double-exponential-time.*

Recall that every property Π has two trivial IPPs: The first IPP is actually a tester that reads the entire input (i.e., $q(n) = n$ and $c(n) = 0$), whereas the second IPP consists of having the prover send the entire input to the verifier who also checks it for consistency with the actual input (i.e., $c(n) = n$ and $q(n) = O(1)$). Thus, Theorem 1.1 asserts that there are cases in which no IPP can significantly improve over these two trivial IPPs.

We admit that Theorem 1.1 has two drawbacks: First, it asserts $q(n) + c(n) = \Omega(n/\log n)$ rather than $q(n) + c(n) = \Omega(n)$; second, it refers to a “very complex” property (i.e., a property in double-exponential-time). The second drawback seems unavoidable to some extent, since every property in uniform- $\mathcal{NC}$ has an IPP with $q(n) + c(n) = \tilde{O}(\sqrt{n})$ (see [21, 22]). Nevertheless, lower bounds on $q(n) + c(n)$ may be obtained for “less complex” properties. We were able to obtain such lower bounds under some standard computational assumptions.

Theorem 1.2 (a conditional lower bound): *Suppose that $\mathcal{E}$ (resp., $\mathcal{NP}$) contains a set that does not have (non-uniform) small-exponential circuit size (i.e., size $2^{\alpha n}$ for some small constant $\alpha > 0$). Then, there exists a property $\Pi \in \mathcal{E}$ (resp., $\Pi \in \mathcal{NP}$) such that every IPP for Π must satisfy $q(n) + c(n) = \Omega(n/\log n)$.*

Using a seemingly stronger assumption yields a (weaker) lower bound for a set in $\mathcal{P}$.

Theorem 1.3 (another conditional lower bound): *Suppose that there exists a pseudorandom generator that fools small-exponential size circuits. Then, there exists a property $\Pi \in \mathcal{P}$ such that every IPP for Π must satisfy $q(n) + c(n) = \Omega(\sqrt{n}/\log n)$.*

Recall that every property in uniform- $\mathcal{NC}$ has IPPs with $q(n) + c(n) = \tilde{O}(\sqrt{n})$ (see [21, 22]), whereas an unconditional separation between uniform- $\mathcal{NC}$ and $\mathcal{P}$ is not known (to say the least). Still, a natural question is whether the lower bound of Theorem 1.3 (or even just $q(n) + c(n) \geq n^{\Omega(1)}$) can be established (for some properties in $\mathcal{P}$) without using any computational assumptions.

We mention that Theorem 1.3 is proved by combining the ideas that underlie Theorems 1.1 and 1.2 with an idea of Kalai and Rothblum [18], which refer to computationally restricted verifiers. In fact, we re-establish and improve their result (see Theorem 5.2): *assuming the existence of pseudorandom generators that fool small-exponential time, we present $\Pi \in \mathcal{P}$ such that for every q such that $q(n) < n/O(\log n)$, every IPP with a PPT verifier for Π must satisfy $c(n) = \Omega(n/q(n))$.* (Our improvement is in the order of quantifiers: in [18] it is $\forall q \exists \Pi$, whereas here it is $\exists \Pi \forall q$.) We also show that the same assumption (or alternatively the “random ETH” assumption) yields a lower bound of $c(n) = \Omega(n)$ (regardless of q) for *tolerant IPPs* (with PPT verifiers, see Theorem 5.3).

Prior work. Interactive Proofs of Proximity were introduced by Rothblum, Vadhan, and Wigderson [21]. While their main result is an IPP for every set in uniform- $\mathcal{NC}$ such that $q(n) \cdot c(n) = n^{1+o(1)}$ (cf., a recent exposition [6]), they also provided a lower bound of $q(n) + c(n) = \sqrt{n/\log n}$ for a special case of 1-round IPPs (i.e., AM-type IPPs) for some natural sets in uniform- $\mathcal{NC}$. A stronger lower bound of the form $q(n) + c(n) = \Omega(n)$ was shown in the special case of MAPs (i.e., 0.5-round IPPs) in [13], albeit for an “artificial” and “complex” property. Lower bounds on $q(n) \cdot c(n)$ were also provided in the special case of MAPs (see [3, 13]) and in the special case of “prover-oblivious queries” (see [10] for a general study of various models of proof-oblivious queries IPPs (which extends their study in the context of MAPs [13])). We stress that, in contrast to [18] and our extension of it (presented in Section 5.1), these works do not restrict the computational complexity of the verifier.

In a somewhat different vein, Gur and Rothblum [14] proved a hierarchy theorem for IPPs, which essentially trades-off the number of rounds for verification time. A later work with Liu shows an exponential separation between MAP and AM-type IPPs [15].

Bi-directional vs uni-directional communication. We stress that, unless explicitly stated otherwise, communication complexity means the total number of bits exchanged by the two parties rather than the number of bits sent by the prover. We recall that in the case of public-coin IPPs, one may assume (w.l.o.g) that the total length of the verifier’s messages is at most $O(\log n)$ times the total length of the prover’s messages; this fact is implicit in [21, Lem 4.6], which follows [11, Apdx. B.3]. We observe that such a result, which is *oblivious of the query complexity*, is unlikely for general IPPs, because it would imply that interactive proofs with laconic provers exist only for sets in $\mathcal{BPP}$ (which is unlikely given that such interactive proofs are known for all problems in $\mathcal{SZK}$ [23] (which is widely conjectured to extend beyond $\mathcal{BPP}$)).¹ Nevertheless, one may wonder

¹Starting with a laconic interactive proof (i.e., one in which the prover sends $O(1)$ bits), one would first obtain an interactive proof in which the total communication complexity is logarithmic, and then apply [7, Thm. 1]. Note that

whether the total length of the verifier’s messages can be related to its query complexity (and to the total length of the prover’s messages). We present a result in this direction.

Theorem 1.4 (compressing the verifier’s communication): *Suppose that Π has an r -round IPP in which the prover sends at most p bits and the verifier makes at most q queries. Then, Π has an r -round IPP of query complexity $O(q)$ and communication complexity $O(p + r \cdot q \cdot \log n)$. Furthermore, the randomness complexity of the verifier is $O(r \cdot q \cdot \log n)$, and perfect completeness is preserved.*

Intuitively, unlike in the case of interactive proof in which the verifier messages may depend on the entire input (and have length that is proportional to it),² in the case of IPPs the verifier only knows $q(n)$ bits of the input. Hence, it seems that the verifier gains nothing by sending messages of length exceeding $O(q(n) \cdot \log n)$. We prove that this is indeed the case (while stressing that our transformation does not preserve the computational complexity of the verifier). This allows to derive lower bounds on uni-directional communication (and query complexity) from the lower bounds that refer to bidirectional communication.

1.2 Discussion

Viewing interactive proofs of proximity (IPPs) from the perspective of interactive proofs (IPs) leads to postulating that the verifier strategy in IPPs is represented by a probabilistic-polynomial-time interacting machine. In contrast, viewing IPPs as an extension of the property testing model leads to neglecting the computational complexity of the verifier (or, at least, to viewing it as secondary to the query complexity).

In any case, the query complexity of the verifier is the aspect that prevents a reduction of the study of IPPs to the study of IPs. (In other words, such a reduction is trivial when the verifier is allowed linear query complexity.) Analogously, sublinear communication complexity prevents an alternative simple reduction (in which the prover sends the entire input to the verifier who checks that it fits the real input by making $O(1/\epsilon)$ queries (cf. [13]).³

In light of the foregoing, we view the query and communication complexities of the IPPs as the primary complexity measures. This focus is further justified by the fact that we intend to study these complexity measures. From that perspective, the computational complexity of the verifier is secondary. Furthermore, insisting on probabilistic polynomial-time verifiers may obscure the picture; for example, sets that are not in $\mathcal{PSPACE}$ yield properties that do not have IPPs (with polynomial-time verifiers), regardless of the query and communication complexities. Specifically, for any $S \notin \mathcal{PSPACE}$, consider the property $\Pi = \{C(x) : x \in S\}$, where C is a good error-correcting code.

We stress that our main results (i.e., Theorem 1.1–1.4) refer only to the query and communication complexities of the IPPs; they do not restrict the computational complexity of the verifiers. Recall that prior to our work, such lower bounds were known only for restricted classes of IPPs (i.e., constant-round IPPs [21] and IPPs with proof-oblivious queries [10]).

the transformation of general IPPs into public-coin IPPs (which follows the analogous transformation for IP [12, 8]) comes at the cost of increasing their query and communication complexities by a factor that is related to their randomness complexity. Hence, in general, this transformation may yield public-coin IPPs with at least linear complexities.

²For example, consider the laconic interactive proof systems for Quadratic Non-Residuosity and Graph Non-Isomorphism.

³For a property in $\mathcal{BPP}$, the verifier also checks that the string sent by the prover is in the property. For a property in $\mathcal{PSPACE} \setminus \mathcal{BPP}$, the verifier also invokes the relevant interactive proof on this string.

1.3 Techniques

The pivot of our main lower bounds is the observation that the circuit-size of verifier strategies can be upper-bounded in terms of the sum of their query and communication complexities (which yields Theorem 1.1). This observation is rooted in the fact that the verifier’s actions are determined by a sequence of (conditional) probability distributions, which can be approximated by distributions of small support.⁴ Hence, we can derive alternative verifier strategies of size $\text{poly}(n)^{q(n)+c(n)}$; see Theorem 3.2.

Combining the resulting verifier strategies with the optimal prover strategy yields a procedure that distinguishes between inputs in the property and inputs that are far from the property (or set). Hence, sets of larger circuit-size yield (when encoded with a good error-correcting code) properties that cannot have IPPs of small $q(n) + c(n)$, which establishes Theorem 1.2. Combining this idea with a construction presented in [18] (and using an additional idea), we derive Theorem 1.3.

We stress that the foregoing results do not refer to the computational complexity of the verifier’s strategy.⁵ However, they either yield query+communication lower bounds on IPPs for “relatively complex” properties (see Theorem 1.1 at one extreme) or rely on some (reasonable) complexity theoretic assumptions (see Theorem 1.3 at the other extreme, whereas Theorem 1.2 is in-between). Indeed, we wonder whether it is possible to obtain query+communication lower bounds on IPPs for properties in $\mathcal{P}$ without relying on complexity theoretic assumptions.

The proof of Theorem 1.4 is pivoted at the observation that the verifier’s messages carry partial information on the input bits that the verifier read (but not on the bits it did not read). We note that there are only $(2n)^{q(n)}$ possibilities for the location and value of these bits, and that, each such possibility, denoted $\omega \in \Omega$, leads the original verifier to send a message according to a distribution denoted X_ω . We “compress” these distributions into corresponding Y_ω ’s such that (1) Y_ω does not improve the (cheating) prover’s ability to convince the verifier (beyond what is possible based on X_ω), and (2) the honest prover can operate as well when given Y_ω (rather than X_ω). Specifically, while the X_ω ’s may range over a huge domain, the Y_ω ’s range over a relatively small number of values (i.e., over $m = \text{poly}(|\Omega|)$ values), regardless of the original domain, denoted $[M]$. Intuitively, the first condition is related to the requirement that the total variation distance (TVD) between pairs of X_ω ’s is preserved by the corresponding Y_ω ’s. The second condition follows by showing that for every value function $v : [M] \rightarrow [0, 1]$ there exists $v' : [m] \rightarrow [0, 1]$ such that $\mathbb{E}[v'(Y_\omega)] \approx \mathbb{E}[v(X_\omega)]$ for every $\omega \in \Omega$.

Hence, the core of the proof of Theorem 1.4 is a compression of sets of distributions (i.e., the X_ω ’s). The compression scheme is based on a sample of size m that is drawn by selecting each sample-point at random from one of the X_ω ’s. Fixing this sample, denoted $(s_1, \dots, s_m)$, the distribution Y_ω is generated by outputting $j \in [m]$ with probability that is essentially proportional to $\frac{\Pr[X_\omega = s_j]}{\sum_{\omega' \in \Omega} \Pr[X_{\omega'} = s_j]}$. Needless to say, we define $v'(j) \stackrel{\text{def}}{=} v(s_j)$. We believe that the compression scheme is of independent interest. We stress that the distinctive characteristic of this compression scheme is that it is applied to n different distributions and compresses the description of individual items in these distributions.

⁴Such approximations were used in [7, 11] (and we follow them).

⁵In contrast, results that refer to IPPs with polynomial-time verifiers are presented in Section 5. We don’t discuss these results here.

1.4 Organization

We start, in Section 2, with a brief review of the basic definition of Interactive Proof of Proximity (IPPs) and their complexity measures. For wider perspectives see [21] and [5].

Our unconditional results are presented in Section 3. In particular, we upper-bound the circuit size of verifier strategies in terms of the sum of their query and communication complexities. This result implies the existence of properties for which any IPP must have query and communication complexities that sum-up to be almost linear (i.e., Theorem 1.1). Interestingly, the foregoing unconditional upper bound on the circuit size of verifier strategies will also be used towards establishing the conditional results of Sections 4 and 6.

In Sections 4 and 6 we present our main conditional results. In Section 4 we present lower bounds on IPPs for properties in $\mathcal{NP}$ and $\mathcal{E}$, while relying on common conjectures regarding the size of circuits for these classes; this establishes Theorem 1.2. In Section 6 we use stronger assumptions, but derive lower bounds on IPPs for properties in $\mathcal{P}$, establishing Theorem 1.3.

Section 5, where we re-establish and extend the lower bound of [18], is an outlier, since it refers to IPPs with computationally bounded verifiers (i.e., the commonly assumed probabilistic polynomial-time verifiers). This section appears before Section 6, because the former offers a good warm-up towards the latter.

Lastly, in Section 7, we establish Theorem 1.4. That is, we transform IPPs in which the query complexity and the length of the prover's messages are upper-bounded into ones in which also the length of the verifier's messages is upper-bounded.

2 Preliminaries: IPPs and their complexity measures

The notion of an interactive proof of proximity is a hybrid of the notions of property testing and interactive proof systems. The basic mind-frame (i.e., approximate decision and sub-linear complexity) is inherited from the former, whereas the actual setting (i.e., verification via an interaction between a powerful but untrusted prover and a probabilistic verifier) is taken from the latter. In other words, a verifier is a tester that is assisted by interaction with an untrusted prover. In the context of properties of functions, which is the focus of this work, we denote by $(\tilde{P}, V^f)(z)$ the output of V , on input z and oracle access to f , after interacting with an arbitrary prover $\tilde{P}$.

Definition 2.1 (verifier for property Π):⁶ *Let $\Pi = \bigcup_{n \in \mathbb{N}} \Pi_n$ such that Π_n contains Boolean functions over $[n]$ (i.e., functions of the form $f : [n] \rightarrow \{0, 1\}$). An interactive verifier of proximity for Π is an interactive and probabilistic oracle machine, denoted V , that, on input parameters n and ϵ and oracle access to a function $f : [n] \rightarrow \{0, 1\}$, interacts with a potential prover, and outputs a binary verdict that satisfies the following two conditions.*

1. *Completeness: V accepts inputs in Π : For every $n \in \mathbb{N}$ and $\epsilon > 0$, and for every $f \in \Pi_n$, there exists a prover strategy P that makes the verifier accept (w.h.p); that is,*

$$\Pr[(P, V^f)(n, \epsilon) = 1] \geq 2/3.$$

If $\Pr[(P, V^f)(n, \epsilon) = 1] = 1$ always holds, then we say that V has perfect completeness.

⁶Here the property Π is viewed as a set of Boolean functions; in other parts of the paper the property Π is viewed as a set of bit strings.

2. Soundness: V rejects inputs that are ϵ -far from Π : For every $n \in \mathbb{N}$ and $\epsilon > 0$, and for every $f : [n] \rightarrow \{0, 1\}$ such that $\delta_\Pi(f) > \epsilon$, no prover strategy $\tilde{P}$ can make the verifier accept (w.h.p); that is,

$$\Pr[(\tilde{P}, V^f)(n, \epsilon) = 0] \geq 2/3,$$

where $\delta(f, g) \stackrel{\text{def}}{=} |\{i \in [n] : f(i) \neq g(i)\}|/n$ and $\delta_\Pi(f) \stackrel{\text{def}}{=} \min_{g \in \Pi_n} \{\delta(f, g)\}$.

Indeed, we say that f is ϵ -far from Π if $\delta_\Pi(f) > \epsilon$, and otherwise (i.e., $\delta_\Pi(f) \leq \epsilon$) we say that f is ϵ -close to Π .

An alternative formulation specifies the (honest) prover P strategy that is used in the completeness condition. This is called for when wishing to discuss the complexity of such honest prover strategies, and in that case one provides the honest prover with the function f (or with oracle access to it). In such a case, one refers to the pair (P, V) as an **interactive proof of proximity (IPP)** system.

We comment that there is no need to provide the (potentially cheating) provers considered in the soundness condition with the function f , since these provers may depend arbitrarily of f (i.e., have f “hard wired”).

Complexity measures. The following complexity measures of IPPs are of common concern:

- The **query complexity** of V is a function (of n and ϵ) that specifies the maximal number of queries made by V on input parameters n and ϵ , when given oracle access to any function $f : [n] \rightarrow \{0, 1\}$.
- The **communication complexity** of the system is an analogous function that specifies the maximal number of bits sent and received by V .

(Hence, the communication complexity of the system accounts for the communication in both directions; in contrast, one may also consider the communication in each direction.)

- The **computational complexity** of V is an analogous function that specifies the total running time of V (in case it is a uniform machine, and its size as a circuit otherwise).

As stated upfront, we shall focus on the first two complexity measures (i.e., number of query and the total length of the messages exchanged). We are interested in IPPs in which both these measures are sublinear. In contrast, *ignoring the computational complexity of the verifier*, every property has an IPP in which the verifier makes a linear number of queries and does not communicate with the prover at all. Likewise, every property has an IPP in which the prover sends the entire input (i.e., f) to the verifier (which means linear communication complexity), who queries the actual input at $O(1/\epsilon)$ random locations.

An IPP, as any two-party protocol, proceeds in **rounds** such that in each (communication) round a pair of messages is exchanged between the two parties. The common convention is that the verifier sends the first message and the prover sends the very last message. An exceptional case is of MAPs (i.e., “MA proofs of proximity”), in which the communication consists of a single message (sent from the prover to the verifier). Indeed, MAPs may be viewed as 0.5-round IPPs, whereas in 1-round IPPs a single message of the verifier is sent and answered by a single message of the prover.

An IPP is said to be of the **public-coin** type if at each round the verifier selects uniformly at random a string of a predetermined length and sends it to the prover. In the case of public-coin

IPPs, we may assume (w.l.o.g.) that the verifier queries the oracle after completing its interaction with the prover. We stress that the queries may depend on the prover’s messages, unless specified differently such as in models of **prover-oblivious queries**, which were studied extensively in [10].

Convention for lower bounds. In the context of proving lower bounds that depend of the size parameter n , one often neglect the dependence on the proximity parameter ϵ , setting it to a sufficiently small constant. We follow this convention.

3 Unconditional results

We first observe that even if one only insist that IPPs should employ recursive verifier strategies, then there exist properties that have no such IPPs *regardless of the query and communication complexities*.⁷ This observation indicates that, if we seek to study the query and communication complexities of IPPs, then we must avoid even the postulation that the verifiers strategies can be described by algorithms.

The foregoing observation is merely a consequence of the fact that such a verifier, when combined with the corresponding optimal prover strategy (which selects messages that maximize the acceptance probability of the residual interaction), yields a (single) recursive procedure that distinguishes strings in the property from strings far from the property. Hence, we get a countable set of potential distinguishers, but such a set can be easily fooled (by a adequately selected property), which means that there exists a property that has no IPP. Specifically, for every $t : \mathbb{N} \rightarrow \mathbb{N}$, there exists a $t(n)$ -subset $S_n \subset \{0, 1\}^n$ such that none of the first $t(n) - 1$ machines (in the enumeration) can distinguish some distribution over S_n from the uniform distribution over $\{0, 1\}^n$ (see [9]), whereas (assuming, say, $t(n) < 2^{n/2}$) almost all n -bit strings are far from S_n .

In light of the above, we place no computational restrictions on the verifier strategies employed by the IPP, and focus on their query and communication complexities. Specifically, we consider non-uniform verifiers of query complexity $q(n)$ and communication complexity $c(n)$.

We shall show that these verifiers can be (“approximately”) described using $\text{poly}(n)^{q(n)+c(n)}$ bits, which means that the corresponding IPPs can be fooled by almost all $2^{n/2}$ -subsets of $\{0, 1\}^n$, unless $q(n) + c(n) = \Omega(n/\log n)$. (In other words, Lemma 3.1 implies Theorem 1.1; see details below.)

Lemma 3.1 (the description length of verification strategies for IPPs): *Suppose that Π has an IPP with query complexity q and communication complexity c . Then, Π has an IPP of query complexity $O(q)$ and communication complexity $O(c)$ that uses a verification strategy (for n -bit long inputs) that can be described using $\text{poly}(n)^{q(n)+c(n)}$ bits.*

Proof: Starting with a generic IPP, we can convert this IPP into an IPP that proceeds in $O(q(n) + c(n))$ steps such that in each step the IPP proceeds as follows:

1. The verifier sends a (one-bit) message to the prover;
2. the prover responds with a (one-bit) message;

⁷Indeed, this stands in contrast to the existence of IPPs (with either linear query complexity or linear communication complexity) for any property (see Section 2). The point is that these trivial IPPs may use non-recursive verifier strategies.

3. the verifier makes a query to the input oracle;
4. the verifier obtains an answer from the oracle.

That is, each steps consists of four **moves**, two moves performed by the verifier (i.e., the choice of a (bit) message and a query (in $[n]$)) and two moves by external entities (i.e., the prover's and the oracle's answers). Thus, the verifier's behavior in each step is fully specified by the distribution on the verifier's message (which depends on the moves perform so far) and the distribution on the verifier's query (which again depends on the corresponding moves).

Each of these two distributions can be approximated (upto a deviation of $o(1/n)$)⁸ by a sample of $\text{poly}(n)$ elements (drawn from $\{0, 1\}$ and $[n]$, respectively), whereas the number of distributions is $O(n)^{O(q(n)+c(n))}$. Ignoring an additive $o(1)$ term in the completeness and soundness errors, this yields an alternative verifier for Π such that its strategy on n -bit long inputs can be described using $\text{poly}(n)^{q(n)+c(n)}$ bits, which describe the various samples (used by the alternative verifier) for each of the possible sequence of moves performed so far.⁹

The foregoing almost establishes the claim. One deficiency is that the error probability bound increased from $1/3$ to $(1/3) + o(1)$. This can be corrected by employing error reduction. The second deficiency is that we obtained an IPP in which both query and communication complexity is $O(q(n) + c(n))$. This can be corrected by revising the initial four-move presentation such that, in each of the $O(q(n) + c(n))$ steps, only one pair of moves are taken (i.e., either the two communication moves or the query-answer moves). ■

Proof of Theorem 1.1: As noted at the beginning of this section, combining a verifier strategy (for an IPP for Π) with the corresponding optimal prover procedure (which selects messages that maximize the acceptance probability of the residual interaction) yields a Boolean function that distinguishes strings in Π from strings far from Π . On the other hand, a random set of $2^{n/2}$ strings of length n fools any fixed Boolean function with overwhelmingly high probability. Hence, an upper bound on the description length of the verifier's strategy (as provided by Lemma 3.1) yields an upper bound on the number of Boolean functions that we should fool, which in turn implies that a random $2^{n/2}$ -subset as above fools all of them. Details follow.

We use the fact that, for every set of $N = \exp(2^{n/3})$ Boolean functions (over $\{0, 1\}^n$), there exists a $2^{n/2}$ -subset of $\{0, 1\}^n$ that is pseudorandom with respect to each of these functions; that is, each of these functions cannot distinguish the uniform distribution over this $2^{n/2}$ -subset from the uniform distribution over $\{0, 1\}^n$. The latter fact is implicitly shown in [4, Exer. 8.1], which actually refers to the number of potential distinguishers (rather than to their "complexity"). Specifically, the key observation is that, when fixing any potential distinguisher (viewed as a function), with probability $1 - \exp(-\Omega(2^{0.4n}))$, a random $2^{n/2}$ -subset fools this distinguisher (i.e., this distribution is pseudorandom wrt this distinguisher (i.e., the latter cannot distinguish between the uniform distribution on this $2^{n/2}$ -subset and the uniform distribution over n -bit strings)). Trying all possible

⁸We may assume, w.l.o.g., that $q(n) + c(n) = o(n)$, because the lemma holds trivially otherwise (i.e., every property has an IPP with $q(n) = n$ and $c(n) = 0$ (resp., with $c(n) = n$ and $q(n) = O(1/\epsilon)$)).

⁹To upper-bound the deviation of the alternative verifier from the original verifier we consider a t -step game such that at each step a choice is made based on one of two distributions that are determined adversarially based on the choices made so far. (The choice represent either the verifier's message and the prover's response to it or the verifier's query and the oracle's answer.) If the total variation distance between each pair of distributions is at most ϵ , then the total variation distance between a play (of the game) in which the first distribution is always used and a play in which the second distribution is always used is upper-bounded by $t \cdot \epsilon$.

$2^{n/2}$ -subsets (and all candidate distinguishers), and computing the acceptance probability of each n -bit string, we can find (in time 2^{2^n}) a set Π_n that is pseudorandom with respect to all N candidate distinguishers. Now, using Lemma 3.1, it follows that any IPPs of query complexity q and communication complexity c for $\Pi = \bigcup_{n \in \mathbb{N}} \Pi_n$ must satisfy $\text{poly}(n)^{q(n)+c(n)} > \log_2 N = \Omega(2^{n/3})$, because otherwise we reach contradiction to the pseudorandomness of Π_n . Hence, $q(n) + c(n) = \Omega(n/\log n)$ must hold. ■

On the circuit complexity of verification strategies for IPPs: The proof of Lemma 3.1 also implies an upper bound on the circuit complexity of implementing the strategy for the verifier of the corresponding IPP. We state this result for further use.

Theorem 3.2 (the size of circuits for verification strategies of IPPs): *Suppose that Π has an IPP with query complexity q and communication complexity c . Then, Π has an IPP of query complexity $O(q)$ and communication complexity $O(c)$ that uses a verification strategy (for n -bit long inputs) that can be implemented by a family of probabilistic $\text{poly}(n)^{O(c(n)+q(n))}$ -sized circuits.*

4 Conditional results using properties in either $\mathcal{NP}$ or $\mathcal{E}$

Here we present lower bounds that refer to properties that can be decided in exponential-time (rather than in double-exponential time). An alternative instantiation yields properties in $\mathcal{NP}$. Alas, these lower bounds rely on computational complexity conjectures, where the conjecture about $\mathcal{E}$ is quite standard in the area of derandomization. More generally, if $\mathcal{C}$ does not have small circuits (i.e., circuits of size s), then $\mathcal{C}$ has a property that does not have an IPP of small query+communication complexity (i.e., $q(n) + c(n) = o(\log_n s(n))$).

Theorem 4.1 (hardness vs lower bounds for IPP): *Let $\mathcal{C}$ be a complexity class that is closed under polynomial-time computations. Suppose that there exists a set S in $\mathcal{C}$ such that for some constant $\alpha > 0$ and all but finitely many $n \in \mathbb{N}$ every family of $2^{\alpha n}$ -sized circuits fails to correctly decide S on some n -bit strings. Then, there exists a property Π in $\mathcal{C}$ such that for every IPP for Π it holds that $q(n) + c(n) = \Omega(n/\log n)$, where q (resp., c) denotes the query (resp., communication) complexity of the IPP.*

We stress that the bound on $q + c$ holds for a constant value of the proximity parameter ϵ (e.g., $\epsilon = 0.1$ will do). As stated upfront, both $\mathcal{NP}$ and $\mathcal{E}$ are examples of complexity classes to which the theorem can be applied. More generally, if the hypothesis refers to $s(n)$ -sized circuits, then we obtain $q(n) + c(n) = \Omega(\log(s(n))/\log n)$.

We comment that our initial proof of a special case of Theorem 4.1, which refers to the case of $\mathcal{C} = \mathcal{E}$ only, actually implies that even an IPP-like system that (only) distinguishes random elements of the property from random strings must satisfy $q(n) + c(n) = \Omega(n/\log n)$. The proof, which appears in Appendix A, uses the pseudorandom generators of [20], while relying on [1, 17] (or alternatively [24]).

Proof: The basic idea is that an IPP of query complexity q and communication complexity c for a property Π yields a family of $\text{poly}(n)^{O(q(n)+c(n))}$ -sized circuits that distinguishes between n -bit strings in Π and n -bit strings that are ϵ -far from Π . Unless $q(n) + c(n) = \Omega(n/\log n)$, this will contradict the hypothesis, since the hypothesis implies the existence of a set Π , which is in $\mathcal{C}$, for

which such distinguishing is impossible. (Hint: The set Π is obtained by encoding the elements of the set S under a good error-correcting code; intuitively, this encoding bridges the gap between deciding S and the approximate decision that underlies an IPP for Π .) We start by constructing the claimed distinguisher.

Before doing so, we stress that the notion of a distinguisher used here is actually a worst-case notion; that is, this distinguisher is actually a decision procedure that accepts (whp) every input in Π and rejects (whp) every input that is ϵ -far from Π . (This worst-case notion should be contrasted with the notion of average-case distinguishers used in the theory of pseudorandomness; the latter are only required to distinguish between distributions of inputs (i.e., an adequate pseudorandom distribution and the uniform distribution).)

Towards constructing the distinguisher. Applying Theorem 3.2, we obtain a verifier, denoted V , of communication complexity $O(c(n))$ that can be implemented by a family of probabilistic $\text{poly}(n)^{t(n)}$ -sized circuits, where $t(n) = O(c(n) + q(n))$.

We stress that the rest of our proof does not refer to the query complexity of V ; it only refers to the size of the circuit implementing V (i.e., $\exp(O(t(n) \cdot \log n))$) and to the total communication complexity (i.e., $O(c(n))$). In fact, for simplicity, we assume that V has free access to the input (to be denoted x). (Hence, V behaves like a verifier in a standard interactive proof, except that it is not necessarily implemented in probabilistic polynomial-time. Furthermore, V is only required to decide correctly on inputs that are either in Π or are ϵ -far from Π .) Furthermore, for simplicity, we bound the communication complexity by $t(n)$.

Fixing any n , let $t = t(n)$, and assume, w.l.o.g., that the IPP proceeds for t rounds such that in each round the verifier sends a single bit and the prover responds with a single bit. For $\gamma \in \bigcup_{i \in [t]} \{0, 1\}^{2(i-1)}$, we denote by $V(x, r, \gamma)$ the message that the verifier V sends on input x , random-pad r , and a sequence γ that represents all messages exchanged so far with the prover (i.e., the bits exchanged in the previous $i - 1$ rounds).¹⁰

For every partial (communication) transcript $\gamma \in \bigcup_{i \in \{0, 1, \dots, 2t\}} \{0, 1\}^i$, let $p_V^x(\gamma)$ denote the residual probability that the verifier accepts the input x when interacting with the optimal prover on input x , and when γ represents the first i messages exchanged with the prover (i.e., $\lfloor i/2 \rfloor$ messages of the verifier and $\lfloor i/2 \rfloor$ messages of the prover).

In particular, for the full transcript $\gamma \in \{0, 1\}^{2t}$, the value of $p_V^x(\gamma)$ is determined by x, γ and r , where r is distributed uniformly over all possibilities that are consistent with $\gamma = (\alpha_1, \beta_1, \dots, \alpha_t, \beta_t)$ and x ; that is, for every $i \in [t]$, the bit α_i equals the i^{th} message sent by V on (input x) random-pad r , and prior prover messages $\beta_1, \dots, \beta_{i-1}$ (i.e., $\alpha_i = V(x, r, \alpha_1, \beta_1, \dots, \alpha_{i-1}, \beta_{i-1})$). Hence, letting $C_V^x(\gamma)$ denote the set of consistent r 's, we have

$$p_V^x(\gamma) = \mathbb{E}_{r \in C_V^x(\gamma)} [D(x, r, \beta_1, \dots, \beta_t)] \quad (1)$$

where $D : \{0, 1\}^* \times \{0, 1\}^t \rightarrow \{0, 1\}$ represents the final decision predicate of the verifier (and $\gamma = (\alpha_1, \beta_1, \dots, \alpha_t, \beta_t)$).

We note that the other $p_V^x(\gamma)$'s can be defined recursively. (This refers to strictly partial transcripts of length ranging between 0 and $2t-1$, where odd and even length are treated differently.) In particular, for a partial transcript $\gamma \in \bigcup_{i \in [t]} \{0, 1\}^{2i-1}$, which ends with a verifier message, it holds that

¹⁰Indeed, for $\gamma = (\alpha_1, \beta_1, \dots, \alpha_{i-1}, \beta_{i-1})$, the value of $V(x, r, \gamma)$ is actually determined by x, r , and $\beta_1, \dots, \beta_{i-1}$.

$$p_V^x(\gamma) = \max_{\sigma \in \{0,1\}} \{p_V^x(\gamma\sigma)\}. \quad (2)$$

(That is, the current transcript is extended by a prover message that maximizes the residual probability of success.)

For a strictly partial transcript $\gamma \in \bigcup_{i \in [t]} \{0,1\}^{2^{i-1}}$, which ends with a prover message, it holds that

$$p_V^x(\gamma) = \sum_{\sigma \in \{0,1\}} \frac{|C_V^x(\gamma\sigma)|}{|C_V^x(\gamma)|} \cdot p_V^x(\gamma\sigma), \quad (3)$$

where $C_V^x(\gamma)$ (resp., $C_V^x(\gamma\sigma)$) is defined analogously; that is, it equals the set of r 's that are consistent with γ (resp., $\gamma\sigma$). (That is, the current transcript is extended by a verifier message that is selected at random according to the residual probability space that consist of all random-pads that are consistent with γ . Indeed, the probability that σ is selected equals $|C_V^x(\gamma\sigma)|/|C_V^x(\gamma)|$.)

Lastly, letting λ denote the empty string, note that $p_V^x(\lambda)$ equals the probability that the verifier accepts the input x when interacting with the optimal prover (on x). Hence, if V constitute a valid verifier for $\Pi = \bigcup_{n \in \mathbb{N}} \Pi_n$, where $\Pi_n = \Pi \cap \{0,1\}^n$, then $p_V^x(\lambda) \geq 2/3$ for every $x \in \Pi_n$ and $p_V^x(\lambda) \leq 1/3$ for every x that is ϵ -far from Π_n .

The distinguisher. The key observation is that, given x , the value of $p_V^x(\lambda)$ can be well approximated by an $\text{poly}(n)^{t(n)}$ -sized circuit. Essentially, this is done by recursive procedure that uses Eq. (1) in its basis and either Eq. (2) or Eq. (3) in recursive steps.

Claim 4.1.1 (approximating p_V^x): *There exists a Boolean circuit of size $\exp(O(t(n) \cdot \log n))$ that, on input $x \in \{0,1\}^n$, approximates the value of $p_V^x(\lambda)$ upto 0.01.*

We stress that we do not claim that this circuit can be constructed efficiently. Its mere existence suffices. We mention that the following proof also provides circuits (of the same size) for computing an optimal prover strategy.

Proof: We first assume, for simplicity, that the randomness complexity of the verifier is $O(t(n))$. In this case, the circuit first constructs the sets $C_V^x(\gamma)$ for all $\gamma \in \bigcup_{i \in \{0,1,\dots,2t\}} \{0,1\}^i$. (Indeed, by the simplifying assumption (and the upper bound on the computational complexity of V), this construction can be carried out by a circuit of size $\exp(O(t(n))) \cdot \text{Size}_V(n) = \exp(O(t(n) \cdot \log n))$.)¹¹ Using these sets and going from longer γ 's to shorter ones, the circuit now computes all $p_V^x(\gamma)$'s, by first using Eq. (1), and then using Eq. (2) and Eq. (3).

Turning to the general case, note that we can only upper-bound the randomness complexity of V by its size (as a non-uniform circuit), which is $\exp(O(t(n) \cdot \log n))$. Nevertheless, using Newman's idea [19], we can derive an alternative verifier (of similar communication and size complexities), that has randomness complexity $\max(t(n), \log_2 n) + O(1)$, by considering all possible n -bit inputs and all possible prover strategies. Details follow.

First, fixing any input and a potential prover strategy, we observe that, with probability greater than $1 - 2^{-(n+2^{t(n)+1})}$, a random set of $O(n + 2^{t(n)})$ random-pads performs as well as the original set of random-pads (i.e., we consider a verifier that selects a random-pad uniformly in the small

¹¹Specifically, scanning all possible random-pads r and all sequences of potential prover messages $\beta = (\beta_1, \dots, \beta_t)$, we emulate the execution of V on input x and each r and β , placing r in the adequate $C_V^x(\gamma)$'s.

set instead of selecting it uniformly in the original set). Next, applying a union bound (over the $2^n \cdot 2^{2^{t(n)+1}-1}$ possible inputs and prover-strategies), we infer that there exists a set R_n of size $O(n + 2^{t(n)})$ such that the original verifier V can be emulated by a verifier V' that has R_n hard-wired and emulates V using a random-pad that is uniformly selected in R_n . Indeed, this emulation incurs an added error of (say) 0.01 in the completeness and soundness error. (Note that the bound on the randomness complexity of V is only used in order to bound the length of the strings in the hard-wired set R_n .)

Lastly, applying the reasoning of the simplified case to the foregoing V' (rather than to the original V), the claim follows. ■

Deriving a conflict with the hypothesis (unless $t(n) = \Omega(n/\log n)$). Restating Claim 4.1.1, we have shown that there exists a Boolean circuit of size $\exp(O(t(n) \cdot \log n))$ that distinguishes between n -bit strings in Π and n -bit strings that are ϵ -far from Π . That is, this circuit accepts every n -bit string in Π and rejects every n -bit string that is ϵ -far from Π . Using the main hypothesis, we shall show a set Π in $\mathcal{C}$ for which $2^{\alpha n/10}$ -size circuits cannot distinguish inputs in the set Π from inputs that are ϵ -far from the set Π . This implies that $c(n) + q(n) = \Omega(t(n)) = \Omega(n/\log n)$. Intuitively, the claimed set is obtained by encoding the elements of hard set using a good error-correcting code. Details follow.

Claim 4.1.2 (hardness of deciding S implies hardness of distinguishing between Π and far-from- Π): *Suppose that there exists a set S in $\mathcal{C}$ such that for some constant $\alpha > 0$ and all but finitely many $k \in \mathbb{N}$ every family of $2^{\alpha \cdot k}$ -sized circuits fails to correctly decide S on some k -bit strings. Then, there exist a constant $\alpha' > 0$ and a set Π in $\mathcal{C}$ such for all but finitely many $n \in \mathbb{N}$ no $2^{\alpha' \cdot n}$ -size circuit can decide Π correctly on all n -bit strings that are each promised to be either in Π or ϵ -far from Π . In other words, the promise problem of distinguishing inputs in Π from inputs that are ϵ -far from Π has no $2^{\alpha' \cdot n}$ -size circuits.*

Proof: Let $C : \{0, 1\}^* \rightarrow \{0, 1\}^*$ be a good error-correcting code; that is, C constitutes a code of constant rate (i.e., $|C(x)| = O(|x|)$ for every x) and constant relative distance (i.e., for every $x \neq x'$ of equal length it holds that $C(x)$ and $C(x')$ differ on $\Omega(|C(x)|)$ bits).¹² In addition, we assume that C is computable and invertible in polynomial-time. Letting $\Pi = \{C(x) : x \in S\}$, the closure of $\mathcal{C}$ to polynomial-time computations implies that Π is in $\mathcal{C}$. As shown next, the indistinguishability claim is proved by the contrapositive.

The key observation is that all codewords of C are either in Π or are ϵ -far from Π (where $\epsilon > 0$ is slightly smaller than the relative distance of C). Hence, given a decision procedure D that is correct on codewords of C , we obtain a decision procedure for S ; specifically, on input x , the latter decision procedure computes $C(x)$, invokes D on $C(x)$, and outputs $D(C(x))$. Specifically, if D can be implemented by $2^{\alpha' \cdot n}$ -size circuits, then deciding S on k -bit long inputs can be performed by circuits of size $\text{poly}(k) + 2^{\alpha' \cdot k/\rho}$, where ρ is the rate of C . Using any $\alpha' < \rho \cdot \alpha$, this contradicts the hypothesis regarding S . ■

Conclusion. For Π as guaranteed by Claim 4.1.2, we have conflicting conclusions. On the one hand, by Claim 4.1.2, distinguishing between n -bit strings in Π and n -bit strings that are far from Π requires circuits of size $\exp(\Omega(n))$. On the other hand, by Claim 4.1.1, such distinguishing can be performed by $\exp(O(c(n) + q(n)) \cdot \log n)$ -sized circuits. Hence, $c(n) + q(n) = \Omega(n/\log n)$ must hold. ■

¹²Indeed, we also assume that $|C(x)| = |C(x')|$ for every two equal-length strings.

Proof of Theorem 1.2: As stated upfront, Theorem 1.2 follows as a special case of Theorem 4.1.

5 Conditional results using properties in $\mathcal{P}$ (Part 1)

In this section, we consider IPPs with a probabilistic polynomial-time verifier strategy. Under suitable computational assumptions, we show lower bounds on the query and communication complexity of IPPs for some properties in $\mathcal{P}$. In Section 5.1, extending a result Kalai and Rothblum [18], we present a lower bound on the trade-off between the query complexity and the communication complexity of such IPPs. In Section 5.2, we present a lower bound on communication complexity of “tolerant IPPs” (regardless of their query complexity).

Throughout this section, we use a pseudorandom generator (PRG) that is computed in polynomial-time and stretches its input seed to an output sequence that is twice as long. We shall assume that this PRG fools probabilistic algorithms that run in time T , where T ranges between being larger than any polynomial and being some small exponential function (i.e., a function of the form $n \mapsto 2^{\alpha \cdot n}$, for some constant $\alpha > 0$). (Note that since the PRG in use stretches its seed by a factor of 2, a time bound stated in terms of the PRG’s output is closely related to one stated in terms of the PRG’s seed (i.e., $\text{poly}(2k)$ is $\text{poly}(k)$, whereas $\exp(O(2k))$ is $\exp(O(k))$.)

We shall also use an efficiently computable error correcting code C that has constant rate and constant relative distance; that is, the function C is polynomial-time computable, $|C(x)| = O(|x|)$ and for every $x \neq x'$ of equal length $C(x)$ and $C(x')$ differ on $\Omega(|x|)$ bits.

5.1 Reproving the lower bound of [18] and going beyond

For $k, m \in \mathbb{N}$ and a pseudorandom generator $G : \{0, 1\}^k \rightarrow \{0, 1\}^{2k}$, define

$$\Pi^{(k,m)} = \{(C(G(s_1)), \dots, G(s_m)), (s_1, \dots, s_m) : s_1, \dots, s_m \in \{0, 1\}^k\}, \quad (4)$$

where $C : \{0, 1\}^{2mk} \rightarrow \{0, 1\}^{n-mk}$ is a good error correcting code (i.e., $n = O(mk)$ whereas every two C -codewords are far apart). In general, we shall replace k by $k(n)$ and m by $m(n)$ such that $n = \Theta(m(n) \cdot k(n))$.

Theorem 5.1 (Part of [18, Thm. 4]): *Suppose that G is pseudorandom with respect to probabilistic algorithms that run in time T , and that $\Pi^{(k,m)}$ has an IPP with a polynomial-time verifier of query complexity q and communication complexity c . Then, either $q(n) > m(n)/4$ or $c(n) = \Omega(\log \frac{T(k(n))}{\text{poly}(n)})$.*

Note that when $T(k) = \exp(\Omega(k))$ and $k(n) = \omega(\log n)$, we get $c(n) = \Omega(\Omega(k(n)) - O(\log n)) = \Omega(k(n))$. In this case, either $q(n) > m(n)/4$ or $c(n) = \Omega(n/m(n))$. Hence, for $q(n) = m(n)/4$, we get $c(n) = \Omega(n/q(n))$. We stress that $\Pi^{(k,m)}$ is in $\mathcal{P}$; furthermore, if both G and C can be computed in $\mathcal{NC}$, then $\Pi^{(k,m)} \in \mathcal{NC}$.

Proof: Assuming towards the contradiction that both $q(n) \leq \frac{m(n)}{4}$ and $c(n) = o(\log \frac{T(k(n))}{\text{poly}(n)})$, we shall show that this contradicts the pseudorandomness hypothesis regarding G . Specifically, letting $m = m(n)$ and $k = k(n)$, we consider the following distinguisher. On input $x \in \{0, 1\}^{2k}$, the distinguisher proceeds as follows.

1. It selects uniformly at random $i \in [m]$, and lets $x_i \leftarrow x$ and $s_i = 0^k$.

2. For every $j \in [m] \setminus \{i\}$, it selects s_j uniformly at random in $\{0, 1\}^k$ and computes $x_j \leftarrow G(s_j)$.
3. Letting $c = c(n)$, it selects a random set, R , of $O(2^c)$ possible random-pads for the verifier, and computes the acceptance probability of the resulting verifier (when interacting with an optimal prover) on input $z \stackrel{\text{def}}{=} (C(x_1, \dots, x_m), (s_1, \dots, s_m))$.

The distinguisher can compute the foregoing probability in $|R| \cdot 2^c \cdot \text{poly}(n)$ -time, by using the game-tree of this IPP (on input z), since the distinguisher knows z .

Recall that using a random set R yields a residual verifier that is almost as good as the original one (cf. the second part of the proof of Claim 4.1.1).¹³

Note that the foregoing algorithm runs in time $\exp(O(c)) \cdot \text{poly}(n) < T(k)/O(1)$. We shall show that this algorithm distinguishes between random inputs in the image of G and uniformly distributed $2k$ -bit long strings. (This contradicts the hypothesis regarding G .)

On the one hand, if x is not in the image of G , then z is $\Omega(1)$ -far from $\Pi^{(k,m)}$, because $C(x_1, \dots, x_m)$ is $\Omega(1)$ -far from $\{C(G(s_1), \dots, G(s_m)) : s_1, \dots, s_m \in \{0, 1\}^k\}$. It follows that the original verifier accepts z with probability at most $\eta \leq 1/3$, whereas the residual verifier (which uses a random-pad in R) accepts z with probability at most $\eta + 0.01 < 0.35$.

On the other hand, if $x = G(s)$ for some $s \in \{0, 1\}^k$, then for every $i \in [m]$ and every $s_1, \dots, s_m$, letting $s'_i = s$ and $s'_j = s_j$ otherwise, it holds that $z' \stackrel{\text{def}}{=} (C(G(s'_1), \dots, G(s'_m)), (s'_1, \dots, s'_m))$ is in $\Pi^{(k,m)}$. Hence, the probability that the residual verifier accepts z' is at least $1 - \eta - 0.01 > 0.65$. The key observation is that, *when $s, s_1, \dots, s_m$ are uniformly and independently distributed in $\{0, 1\}^k$, and when i is selected uniformly at random in $[m]$, the probability that the verifier examines the i^{th} block of $(s'_1, \dots, s'_m)$ is at most $q/m \leq 0.25$* . (We stress that this assertion holds for a uniformly distributed $z' \in \Pi$, and relies on the fact that this distribution yields no information about i).¹⁴ Recalling that z and z' differ only on s'_i , it follows that the probability that this verifier accepts a random z (which matched the foregoing random $z' \in \Pi$), *when the (optimal) prover behaves according to z'* , is greater than $0.65 - 0.25 = 0.4$. Thus, the probability that the verifier accepts a random z , when the (optimal) prover behaves according to z , is greater than 0.4, because a cheating prover may just replace z by z' when determining its actions (i.e., replace $s_i = 0^k$ by a uniformly distributed G -preimage of x).

Recall that the foregoing algorithm computes the acceptance probability of a residual verifier on input z constructed based on its own input x . If x is not in the range of G , then the computed value is at most 0.35, whereas if $x = G(s)$ for a uniformly distributed $s \in \{0, 1\}^k$, then (on the average) the computed value is at least 0.4. Hence, this algorithm, which runs in time $\exp(O(c)) \cdot \text{poly}(n) < T(k)/O(1)$, distinguishes between random $2k$ -bit strings and pseudorandom strings generated by G , in contradiction to the hypothesis regarding G . ■

Digest: Strings in the property $\Pi^{(k,m)}$ consist of two parts: The first part is an error-correcting version of the concatenation of m pseudorandom sequences, whereas the second part is the concatenation of the m corresponding seeds. The first part is the actual core of $\Pi^{(k,m)}$, whereas the second part is there in order to reduce the computational complexity of $\Pi^{(k,m)}$ (i.e., placed $\Pi^{(k,m)}$

¹³ Actually, the argument here is more similar to the one in [7, Sec. 2.5]: We apply a union bound over all prover strategies (only).

¹⁴ In contrast, suppose that the m -long sequence encoded by C would have yielded information about i . In such a case, even single bits in the encoding could have leaked such information.

in $\mathcal{P}$ or even in $\mathcal{NC}$).¹⁵ The hypothesis that the query complexity is at most $m/4$ implies that the verifier cannot examine most seeds, which essentially implies an interactive proof of communication complexity $c'(2k) = c(2n/3m)$ for the set $\{G(s) : s \in \{0, 1\}^*\}$, which in turn yields a distinguisher of time complexity $\text{poly}(n) \cdot \exp(c')$.

Perspective: Intuitively, we capitalized on the fact that an IPP of query complexity $m(n)/4$ for $\Pi^{(k,m)}$ yields a “weakly tolerant” IPP for $\Pi^{(k,m)}$, where the tolerance is at a very low rate (i.e., $\frac{k(n)}{n}$).¹⁶ (Considering a standard notion of tolerant IPPs, we can obtain a stronger result: See Section 5.2.) Note that we took the unconventional step of considering a cheating prover for inputs that are close to $\Pi^{(k,m)}$, while relying on the fact that our decision procedure emulates the optimal prover.

Beyond Theorem 5.1: Assuming the existence of G that is pseudorandom with respect to probabilistic algorithms that run in time T , Theorem 5.1 implies that, *for every q , there exists a property Π in $\mathcal{P}$ such that every IPP for Π of query complexity q (and polynomial-time verifier) has communication complexity $c(n) = \Omega(\log \frac{T(\Omega(n/q(n)))}{\text{poly}(n)})$.* Here we switch the order of quantifiers, showing

Theorem 5.2 (Theorem 5.1, revisited): *Suppose that G is pseudorandom with respect to probabilistic algorithms that run in time T . Then, there exists a property Π in $\mathcal{P}$ such that if Π has an IPP with a polynomial-time verifier of query complexity q , then the communication complexity of this IPP is $c(n) = \Omega(\log \frac{T(\Omega(n/q(n)))}{\text{poly}(n)})$.*

Note that when $T(k) = \exp(\Omega(k))$ and $q(n) < n/O(\log n)$, we get $c(n) = \Omega(n/q(n))$. Again, if G can be computed in $\mathcal{NC}$, then $\Pi \in \mathcal{NC}$.

Proof Sketch: Employing a good error correcting code C that can be computed in $\mathcal{NC}$, we use the following variant of Eq. (4):

$$\Pi = \bigcup_{m,k \in \mathbb{N}} \{(C(m, k, G(s_1), \dots, G(s_m)), (s_1, \dots, s_m)) : s_1, \dots, s_m \in \{0, 1\}^k\}. \quad (5)$$

Observe that for any $(m, k) \neq (m', k')$, either $|C(m, k, z)| \neq |C(m', k', z')|$ or $C(m, k, z)$ is far from $C(m', k', z')$. Hence, for any IPP of query complexity $q : \mathbb{N} \rightarrow \mathbb{N}$, we may set $m = 4q$ and $k = \Theta(n/q)$ and follow the strategy of the proof of Theorem 5.1. ■

5.2 A stronger result for tolerant IPPs

We shall prove the existence of a set in $\mathcal{P}$ such that every *tolerant* IPP for it has to satisfy $c(n) = \Omega(\log T(n))$, where T is as in Theorem 5.1, regardless of the query complexity q of the IPP. Recall that, for $0 < \epsilon' < \epsilon < 1$, an ϵ' -tolerant ϵ -IPP for Π should accept (w.h.p) every input that is

¹⁵We use two features of the pseudorandom generator. The first feature is that it yields a pair of distributions that are hard to distinguish (in the average-case sense). The second feature is that the instances in one of these two distributions can be efficiently generated along with corresponding NP-witnesses (of linear length).

¹⁶Essentially, we showed that the original IPP preserves its completeness and soundness probabilities (up to a small deviation) also when one tampers with $k(n)$ of the input bits.

ϵ' -close to Π and reject (w.h.p) every input that is ϵ -far from Π . We shall prove the result for some positive constants $\epsilon' < \epsilon < 1/3$.

Using a pseudorandom generator $G : \{0, 1\}^k \rightarrow \{0, 1\}^{2k}$, we define

$$\Pi = \{(C(G(s)), s) : s \in \{0, 1\}^k\}, \quad (6)$$

where $C : \{0, 1\}^{2k} \rightarrow \{0, 1\}^{n-k}$ is a polynomial-time computable code of relative distance $\delta > 0$ and constant rate $\rho > 0$ satisfying $\rho < \delta$. We make the following observations:

- Any input of the form $(C(G(s)), \cdot)$ is $\frac{k}{n}$ -close to Π .
- Any input of the form $(C(x), \cdot)$ such that $x \notin \text{Image}(G)$ is $\frac{\delta \cdot (n-k) - 1}{n}$ -far from Π .
- $n = \frac{2k}{\rho} + k \in \left(\frac{2k}{\rho}, \frac{3k}{\rho}\right)$.

Observing that $k < \rho n/2$ (since $n > 2k/\rho$) and $n > 3k$, we set $\epsilon' = \rho/2 > k/n$ and $\epsilon = 2\delta/3$, noting that $\epsilon > \epsilon'$ (since $\delta > \rho$) and $\epsilon < \delta \cdot (n-k)/n$ (since $2/3 < (n-k)/n$). Thus, an ϵ' -tolerant ϵ -IPP for Π should accept any input of the form $(C(G(s)), \cdot)$ and reject any input of the form $(C(x), \cdot)$ such that x is not in the image of G .

Theorem 5.3 (conditional lower bounds on tolerant IPPs for properties in $\mathcal{P}$): *Suppose that G is pseudorandom with respect to probabilistic algorithms that run in time T , and that Π has an ϵ' -tolerant ϵ -IPP with a polynomial-time verifier of communication complexity c . Then, $c(n) = \Omega(\log \frac{T(k(n))}{\text{poly}(n)})$.*

We stress that Π is in $\mathcal{P}$. Note that when $T(k) = \exp(\Omega(k))$, we get $c(n) = \Omega(k(n)) = \Omega(n)$.

Proof: Suppose that the tolerant IPP has error probability $1/3$, and consider the following potential distinguisher. On input $x \in \{0, 1\}^{2k(n)}$, the distinguisher selects a random set, R , of $O(2^{c(n)})$ possible random pads for the verifier, and computes the acceptance probability of the resulting verifier (when interacting with an optimal prover) on input $z \leftarrow (C(x), 0^{k(n)})$. (Recall that this computation can be performed in $\exp(c(n)) \cdot \text{poly}(n)$ -time; see the proof of Theorem 5.1.)

On the one hand, if x is in the image of G , then z is ϵ' -close to Π . It follows that the probability that the residual verifier accepts z is at least $(2/3) - 0.01 > 0.65$. On the other hand, if x is not in the image of G , then z is ϵ -far from Π . It follows that the probability that the residual verifier accepts z is at most $(1/3) + 0.01 < 0.35$.

Thus, avoiding a contradiction to the pseudorandomness of G requires $\exp(c(n)) \cdot \text{poly}(n) > T(k(n))$ (equiv., $c(n) = \Omega(\log(T(k(n))/\text{poly}(n)))$). ■

An alternative hypothesis yielding the same conclusion. Observing that the proof of Theorem 5.3 actually yields a decision procedure for the NP-set $\{G(s) : s \in \{0, 1\}^*\}$, we conclude that we can use any NP-set that is conjectured to be hard in the worst-case. In particular, here we derive the conclusion of Theorem 5.3 based on a worst-case hypothesis regarding SAT. Specifically, we may use the randomized Exponential-Time Hypothesis (rETH), introduced in [2] as a strengthening of ETH [16]. More generally, we shall assume that SAT is not solvable in $\text{BPTIME}(T)$, for some super-polynomial function T .

Theorem 5.4 (Theorem 5.3, revisited using rETH): *Suppose that SAT is not solvable by probabilistic algorithms that run in time T , and let Π consist of pairs of the form $(C(\phi), \tau)$, where ϕ is a satisfiable CNF formula, τ is a satisfying assignment to it, and C is a good error correcting code that is polynomial-time computable and invertible. If Π has a tolerant IPP with a polynomial-time verifier of communication complexity c , then $c(n) = \Omega(\log \frac{T(n)}{\text{poly}(n)})$.*

Again, Π is in $\mathcal{P}$, and under rETH (i.e., when $T(n) = \exp(\Omega(n))$), we get $c(n) = \Omega(n)$.

Proof Sketch: Following the proof strategy of Theorem 5.3, we observe that the distinguisher presented there is actually a decision procedure. Hence, the existence of a tolerant IPP of communication complexity c implies a randomized decision procedure for SAT that runs in time $\exp(c(n)) \cdot \text{poly}(n)$. Thus, avoiding a contradiction to the worst-case hardness of SAT requires $\exp(c(n)) \cdot \text{poly}(n) > T(n)$ (equiv., $c(n) = \Omega(\log(T(n)/\text{poly}(n)))$). ■

Variants of Theorems 5.3 and 5.4. By omitting the NP-witness from the properties Π considered in Theorems 5.3 and 5.4, we get results about standard IPPs (rather than tolerant IPPs), but these are for properties in $\mathcal{NP}$.

Theorem 5.5 (conditional lower bounds on IPPs for properties in $\mathcal{NP}$): *Let $C : \{0, 1\}^m \rightarrow \{0, 1\}^n$ be a polynomial-time computable code of relative distance $\delta > 0$ and constant rate $\rho > 0$.*

1. *Suppose that G is pseudorandom with respect to probabilistic algorithms that run in time T , and let $\Pi' = \{C(G(s)) : s \in \{0, 1\}^*\} \in \mathcal{NP}$. If Π' has an IPP with a polynomial-time verifier of communication complexity c , then $c(n) = \Omega(\log \frac{T(n)}{\text{poly}(n)})$.*
2. *Suppose that SAT is not solvable by probabilistic algorithms that run in time T , and let $\Pi' = \{C(\phi) : \phi \in \text{SAT}\} \in \mathcal{NP}$, where SAT is the set of satisfiable CNF formula. If Π' has an IPP with a polynomial-time verifier of communication complexity c , then $c(n) = \Omega(\log \frac{T(n)}{\text{poly}(n)})$.*

6 Conditional results using properties in $\mathcal{P}$ (Part 2)

Using the ideas presented in Section 5, we reconsider our basic setting in which no computational restrictions are imposed on the verifier. Specifically, we first establish a result analogous to Theorem 5.2.

Theorem 6.1 (Theorem 5.2 adapted for computationally unbounded verifiers): *Suppose that G is pseudorandom with respect to non-uniform circuits of size S . Then, there exists a property Π in $\mathcal{P}$ such that if Π has an IPP of query complexity q and communication complexity c , then $c(n) + q(n) = \Omega(\frac{\log S(\Omega(n/q(n)))}{O(\log n)})$.*

Note that when $S(k) = \exp(\Omega(k))$, we get $q(n) + c(n) = \Omega(n/(q(n) \cdot \log n))$ (equiv., $q(n) \cdot (q(n) + c(n)) = \Omega(n/\log n)$), which implies $q(n) + c(n) = \Omega(\sqrt{n/\log n})$. This establishes Theorem 1.3. Again, if G can be computed in $\mathcal{NC}$, then $\Pi \in \mathcal{NC}$.

Proof Sketch: We start with establishing a non-uniform version of Theorem 5.1, which refers to the property $\Pi^{(k,m)}$ as defined in Eq. (4). Specifically, we shall show that the existence of an IPP

with query complexity $q(n) < m(n)/O(1)$ and communication complexity $c(n) = o(\frac{\log S(k(n))}{O(\log n)}) - q(n)$ contradicts the pseudorandomness hypothesis regarding G .

As in the proof of Theorem 4.1, we start by using Theorem 3.2 and concluding that $\Pi^{(k,m)}$ has an IPP of query complexity $O(q(n))$ and communication complexity $O(c(n))$ such that the verifier strategy can be implemented by circuits of size $s(n) = \exp(O((q(n) + c(n)) \cdot \log n))$. Proceeding as in the proof of Theorem 5.1, we infer that there exists a distinguisher of size $\exp(O(c(n))) \cdot s(n) = \text{poly}(s(n)) < S(k(n))/O(1)$ that distinguishes between $2k(n)$ -bit long pseudorandom sequences generated by G and uniformly distributed $2k(n)$ -bit long strings.

Next, defining Π as in Eq. (5) (in the proof of Theorem 5.2), we conclude that, for every IPP (for Π) of query complexity q , the communication complexity is $\Omega(\frac{\log S(\Omega(n/q(n)))}{O(\log n)}) - q(n)$. ■

Theorem 6.2 (Theorem 5.3 adapted for computationally unbounded verifiers): *Suppose that G is pseudorandom with respect to non-uniform circuits of size S , and let Π be as in Eq. (6). If Π has an ϵ' -tolerant ϵ -IPP of query complexity q and communication complexity $c(n)$, then $c(n) + q(n) = \Omega(\frac{\log S(\Omega(n))}{O(\log n)})$.*

Recall that Π is in $\mathcal{P}$, and that for $S(k) = \exp(\Omega(k))$ we get $c(n) + q(n) = \Omega(n/\log n)$. We mention that an analogous non-uniform version of Theorem 5.4 can be proved using the analogous adaptations.

Proof Sketch: As in the proof of Theorem 6.1, the key observation is that the hypothesis implies that Π has an IPP of query complexity $O(q(n))$ and communication complexity $O(c(n))$ such that the verifier strategy can be implemented by circuits of size $s(n) = \exp(O((q(n) + c(n)) \cdot \log n))$. This yields a distinguisher (for G) of size $\exp(O(c(n))) \cdot s(n) = \text{poly}(s(n))$ and contradiction is avoided only if $q(n) + c(n) = \Omega(\frac{\log S(\Omega(n))}{O(\log n)})$. ■

7 Compressing the verifier's messages

As stated in Section 1.3, the proof of Theorem 1.4 reduces to “compressing distributions” in the sense defined in Section 7.1. Hence, we first study this general problem, which is of independent interest. Using this solution, we apply it to the (distribution of the) verifier's messages (see Section 7.2).

Before starting, we stress that the error parameter ε used in the current section is not the proximity parameter of the IPP (i.e., ϵ). Furthermore, the parameter n used in Section 7.1 is not the length of the input to the IPP.

7.1 Compressing distributions

Given n distributions, $X_1, \dots, X_n$, over a huge domain $[M]$, we wish to compress them so to obtain distributions $Y_1, \dots, Y_n$ over a smaller domain $[m]$, where $m = \text{poly}(n/\varepsilon)$, such that the following two conditions hold:¹⁷

1. Pairwise TVDs are approximately preserved; that is, for every $i_1, i_2 \in [n]$,

$$\sum_{j \in [m]} |\Pr[Y_{i_1} = j] - \Pr[Y_{i_2} = j]| \approx \sum_{e \in [M]} |\Pr[X_{i_1} = e] - \Pr[X_{i_2} = e]|,$$

¹⁷The variable n is different from the one in our application (i.e., Section 7.2). In the latter, $[n]$ is replaced by Ω .

where here and below approximation is up to an additive error of ε .

2. Functionality w.r.t a given value function $v : [M] \rightarrow [0, 1]$ is preserved; that is, there exists $v' : [m] \rightarrow [0, 1]$ such that for every $i \in [n]$ it holds that $\mathbb{E}[v'(Y_i)] \approx \mathbb{E}[v(X_i)]$.

We construct the desired distributions by fixing an adequate sequence $\bar{s} = (s_1, \dots, s_m) \in [M]^m$, and defining $v'(j) \stackrel{\text{def}}{=} v(s_j)$ and $Y_1, \dots, Y_n$ such that $\Pr[Y_i = j]$ is *proportional to* $p_i(s_j)/p(s_j)$, where $p_k(e) \stackrel{\text{def}}{=} \Pr[X_k = e]$ and $p(e) \stackrel{\text{def}}{=} \sum_{k \in [n]} p_k(e)/n$. Indeed, since v' and the Y_i 's depends on $\bar{s}$, it is prudent to denote them $v^{\bar{s}}$ and $Y_i^{\bar{s}}$.

In the analysis, we shall consider selecting $\bar{s} = (s_1, \dots, s_m)$ such that each s_j is identically and independently distributed according to p . We will show that, w.v.h.p (over the choice of $\bar{s}$), the pairwise TVDs and the functionality (w.r.t a fixed v) are preserved. Let us first detail the construction of the Y_i 's.

The distribution $Y_i^{\bar{s}}$. For fixed $\bar{s} = (s_1, \dots, s_m)$ and $i \in [n]$, the distribution $Y_i^{\bar{s}}$ is defined via the rejection sampling paradigm such that, in each iteration, we select $j \in [m]$ uniformly at random and output j with (“acceptance”) probability

$$\alpha_i^{\bar{s}}(j) \stackrel{\text{def}}{=} \frac{p_i(s_j)}{n \cdot p(s_j)} = \frac{p_i(s_j)}{\sum_{k \in [n]} p_k(s_j)} \quad (7)$$

Note that $\alpha_i(j) \stackrel{\text{def}}{=} \alpha_i^{\bar{s}}(j) \in [0, 1]$. The probability that a single iteration produces any output is $\sum_{j \in [m]} \alpha_i(j)/m$, which is typically $\Theta(1/n)$ (i.e., the bound holds w.v.h.p. over the choice of $\bar{s}$).¹⁸ Repeating these iterations till an output is produced, we obtain the distribution $Y_i = Y_i^{\bar{s}}$ such that $q_i(j) = q_i^{\bar{s}}(j) \stackrel{\text{def}}{=} \Pr[Y_i^{\bar{s}} = j]$ equals

$$\frac{\alpha_i(j)}{\sum_{k \in [m]} \alpha_i(k)} = \frac{p_i(s_j)/p(s_j)}{\sum_{k \in [m]} (p_i(s_k)/p(s_k))} \quad (8)$$

which (as we shall see) is typically $\approx p_i(s_j)/(m \cdot p(s_j))$.

We shall assume that $\varepsilon > n/m$ (equiv., $m > n/\varepsilon$). Actually, we shall use $m > \tilde{O}(n/\varepsilon)^2$. We shall first establish the following key fact, which will be used in proving the two foregoing conditions (i.e., “preservation of pairwise TVD” and “preservation of functionality”).

Claim 7.1 (the distribution of $\sigma_i(\bar{s})$): *For every $\bar{s} = (s_1, \dots, s_m) \in [M]^m$, let*

$$\sigma_i(s_1, \dots, s_m) \stackrel{\text{def}}{=} \sum_{k \in [m]} (p_i(s_k)/p(s_k)). \quad (9)$$

Then, with probability at least $1 - \exp(-\Omega((\varepsilon/n)^2 \cdot m))$ over the choice of $\bar{s} \sim p^m$, it holds that $\sigma_i(\bar{s}) = (1 \pm \varepsilon) \cdot m$.

¹⁸For every $j \in [m]$

$$\mathbb{E}_{\bar{s} \sim p^m} [\alpha_i^{\bar{s}}(j)] = \mathbb{E}_{s_j \sim p} \left[\frac{p_i(s_j)}{n \cdot p(s_j)} \right] = \sum_{e \in [M]} \frac{p_i(e)}{n}$$

which equals $1/n$. As shown in the proof of Claim 7.1, for any fixed $i \in [n]$ and random $\bar{s} \sim p^m$, the random variables $\alpha_i^{\bar{s}}(1), \dots, \alpha_i^{\bar{s}}(m)$ are IID random variables. Hence, w.v.h.p., their sum is approximately m/n .

Hence, with probability at least $1 - \exp(-\Omega((\varepsilon/n)^2 \cdot m))$ over the choice of $\bar{s} = (s_1, \dots, s_m) \sim p^m$, it holds that $q_i^{\bar{s}}(j) = \frac{p_i(s_j)}{(1 \pm \varepsilon) \cdot m \cdot p(s_j)}$.

Proof: The key observation is that $\sigma_i(\bar{s})$ is the sum of m IIDs that each ranges in $[0, n]$ and has expectation 1; specifically, the k^{th} random variable is $p_i(s_k)/p(s_k)$ and $\mathbb{E}_{\bar{s} \sim p^m}[p_i(s_k)/p(s_k)]$ equals

$$\mathbb{E}_{e \sim p}[p_i(e)/p(e)] = \sum_{e \in [M]} p_i(e) = 1.$$

Using Chernoff bound, the claim follows. $\blacksquare$

Claim 7.2 (preservation of TVD of pairs of distributions): *For every pair $\{i_1, i_2\} \in \binom{[n]}{2}$, with probability at least $1 - \exp(-\Omega((\varepsilon/n)^2 \cdot m))$ over the choice of $\bar{s}$, it holds that $\|q_{i_1}^{\bar{s}} - q_{i_2}^{\bar{s}}\|_1 = \|p_{i_1} - p_{i_2}\|_1 \pm O(\varepsilon)$.*

The claim is quite intuitive in the special case of $n = 2$, but our focus is on the general case (and more so on $n \gg 2$).

Proof: Without loss of generality, we consider $i_1 = 1$ and $i_2 = 2$. We shall show that, w.v.h.p (over the choice of $\bar{s} = (s_1, \dots, s_m)$), it holds that $\|q_1^{\bar{s}} - q_2^{\bar{s}}\|_1 \approx \|p_1 - p_2\|_1$. Specifically, for a random choice of $\bar{s}$, the value $\|q_1^{\bar{s}} - q_2^{\bar{s}}\|_1$, which equals $\sum_{j \in [m]} |q_1^{\bar{s}}(j) - q_2^{\bar{s}}(j)|$, is the sum of m identical (alas somewhat dependent) random variables such that each ranges in $[0, 1]$. That is, for each $j \in [m]$, we let $\zeta_j = \zeta_j(\bar{s})$ denote $|q_1^{\bar{s}}(j) - q_2^{\bar{s}}(j)|$, where $\bar{s} = (s_1, \dots, s_m)$. We first observe that the expected value of $\zeta_j(\bar{s})$ equals

$$\mathbb{E}_{\bar{s} \sim p^m} [|q_1^{\bar{s}}(j) - q_2^{\bar{s}}(j)|] = \mathbb{E}_{s_1, \dots, s_m \sim p} \left[\left| \frac{p_1(s_j)/p(s_j)}{\sum_{k \in [m]} (p_1(s_k)/p(s_k))} - \frac{p_2(s_j)/p(s_j)}{\sum_{k \in [m]} (p_2(s_k)/p(s_k))} \right| \right] \quad (10)$$

$$= \mathbb{E}_{s_m \sim p} \mathbb{E}_{s_1, \dots, s_{m-1} \sim p} \left[\left| \frac{p_1(s_m)/p(s_m)}{\sigma_1(s_1, \dots, s_m)} - \frac{p_2(s_m)/p(s_m)}{\sigma_2(s_1, \dots, s_m)} \right| \right] \quad (11)$$

where the first equality uses Eq. (8) and $\sigma_i(\bar{s})$ is as defined in Claim 7.1. Next, we simplify Eq. (11) using Claim 7.1.

Specifically, recalling that $p_i(s_j)/p(s_j) \in [0, n]$, we use $\sigma_i(s_1, \dots, s_m) = \sigma_i(s_1, \dots, s_{m-1}) \pm n$ and apply Claim 7.1 to $\sigma_i(s_1, \dots, s_{m-1})$. Hence, for every value of s_m , with probability $1 - \exp(-\Omega((\varepsilon/n)^2 \cdot m))$, it holds that $\sigma_i(s_1, \dots, s_m) = (1 \pm \varepsilon) \cdot m$, whereas $\sigma_i(s_1, \dots, s_m)$ is always at most mn . Hence, for every value of s_m it holds that

$$\mathbb{E}_{s_1, \dots, s_{m-1} \sim p} \left[\left| \frac{p_1(s_m)/p(s_m)}{\sigma_1(s_1, \dots, s_m)} - \frac{p_2(s_m)/p(s_m)}{\sigma_2(s_1, \dots, s_m)} \right| \right] \quad (12)$$

$$\begin{aligned} &= \left| \frac{p_1(s_m)/p(s_m)}{(1 \pm \varepsilon) \cdot m} - \frac{p_2(s_m)/p(s_m)}{(1 \pm \varepsilon) \cdot m} \right| \pm \exp(-\Omega((\varepsilon/n)^2 \cdot m)) \cdot mn \\ &= \left| \frac{(1 \pm 2\varepsilon) \cdot p_1(s_m)}{m \cdot p(s_m)} - \frac{(1 \pm 2\varepsilon) \cdot p_2(s_m)}{m \cdot p(s_m)} \right| \end{aligned} \quad (13)$$

Combining Eq. (10)&(11) and Eq. (12)&(13), and recalling that $\zeta_j(\bar{s}) = |q_1^{\bar{s}}(j) - q_2^{\bar{s}}(j)|$, we get

$$\begin{aligned}
\mathbb{E}_{\bar{s} \sim p^m} [\zeta_j(\bar{s})] &= \mathbb{E}_{s_m \sim p} \left[\left| \frac{(1 \pm 2\varepsilon) \cdot p_1(s_m)}{m \cdot p(s_m)} - \frac{(1 \pm 2\varepsilon) \cdot p_2(s_m)}{m \cdot p(s_m)} \right| \right] \\
&= \sum_{e \in [M]} \left| \frac{(1 \pm 2\varepsilon) \cdot p_1(e)}{m} - \frac{(1 \pm 2\varepsilon) \cdot p_2(e)}{m} \right| \\
&= \frac{\|p_1 - p_2\|_1 \pm 4\varepsilon}{m}
\end{aligned}$$

Ignoring for a moment the fact that the ζ_j 's are not independent, the claim essentially follows. Unfortunately, the $\zeta_j(\bar{s})$'s are not independent (cf., Eq. (10)), but their dependence only affects the denominators of the r.h.s. of Eq. (10). Intuitively, since these random quantities are concentrated around m , we can replace them by $(1 \pm \varepsilon) \cdot m$ and obtain IIDs that are easy to analyze. Details follow.

We introduce auxiliary random variables ξ_j 's such that $\xi_j(\bar{s}) = \frac{|p_1(s_j) - p_2(s_j)|}{m \cdot p(s_j)} \in [0, n/m]$, which are independent and are “coupled” to the ζ_j 's. First, observe that $\sum_{j \in [m]} \xi_j = \|p_1 - p_2\|_1 \pm \varepsilon$ with probability $1 - \exp(-\Omega((\varepsilon/n)^2 \cdot m))$, since $\mathbb{E}[\xi_j] = \|p_1 - p_2\|_1/m$ for every $j \in [m]$. Next, observe that, when $\sigma_i(\bar{s}) = (1 \pm \varepsilon) \cdot m$ for both $i \in \{1, 2\}$, it holds that $\zeta_j(\bar{s}) = \xi_j(\bar{s}) \pm \frac{(p_1(s_j) + p_2(s_j)) \cdot O(\varepsilon)}{m \cdot p(s_j)}$.¹⁹ Recalling that $\mathbb{E}_{e \sim p}[p_i(e)/p(e)] = 1$, it follows that $\mathbb{E}_{\bar{s}}[\zeta_j(\bar{s})] = \mathbb{E}_{\bar{s}}[\xi_j(\bar{s})] \pm O(\varepsilon/m)$. Lastly, we view $\zeta(\bar{s}) \stackrel{\text{def}}{=} \sum_{j \in [m]} \zeta_j(\bar{s})$ as a deterministic modification of $\xi(\bar{s}) \stackrel{\text{def}}{=} \sum_{j \in [m]} \xi_j(\bar{s})$ such that $\zeta(\bar{s})$ is reset to an arbitrary value in $[\xi(\bar{s}) \pm O(\varepsilon)]$ if $\sigma_i(\bar{s}) = (1 \pm \varepsilon) \cdot m$ for both $i \in \{1, 2\}$, where the condition holds with probability $1 - \exp(-\Omega((\varepsilon/n)^2 \cdot m))$. Hence, with probability $1 - \exp(-\Omega((\varepsilon/n)^2 \cdot m))$ over the choice of $\bar{s}$, it holds that

$$\begin{aligned}
\|q_1^{\bar{s}} - q_2^{\bar{s}}\|_1 &= \sum_{j \in [m]} \zeta_j(\bar{s}) \\
&= \sum_{j \in [m]} \xi_j(\bar{s}) \pm O(\varepsilon) \\
&= \|p_1 - p_2\|_1 \pm O(\varepsilon).
\end{aligned}$$

Observing that the foregoing applies to any $\{i_1, i_2\} \in \binom{[n]}{2}$, this establishes the claim. $\blacksquare$

Claim 7.3 (preservation of functionality): *For every $v : [M] \rightarrow [0, 1]$ and every $i \in [n]$, with probability at least $1 - \exp(-\Omega((\varepsilon/n)^2 \cdot m))$ over the choice of $\bar{s} = (s_1, \dots, s_m)$, it holds that $\mathbb{E}_{j \sim q_i^{\bar{s}}}[v^{\bar{s}}(j)] = \mathbb{E}_{e \sim p_i}[v(e)] \pm O(\varepsilon)$, where $v^{\bar{s}}(j) = v(s_j)$.*

¹⁹When $\sigma_i(\bar{s}) = (1 \pm \varepsilon) \cdot m$ for both $i \in \{1, 2\}$, analogously to Eq. (10)&(11) and Eq. (12)&(13), we have

$$\begin{aligned}
\zeta_j(\bar{s}) &= \left| \frac{p_1(s_j)/p(s_j)}{\sigma_1(\bar{s})} - \frac{p_2(s_j)/p(s_j)}{\sigma_2(\bar{s})} \right| \\
&= \left| \frac{(1 \pm \varepsilon)^{-1} \cdot p_1(s_j)}{m \cdot p(s_j)} - \frac{(1 \pm \varepsilon)^{-1} \cdot p_2(s_j)}{m \cdot p(s_j)} \right| \\
&= \xi_j(\bar{s}) \pm 2\varepsilon \cdot \frac{p_1(s_j) + p_2(s_j)}{m \cdot p(s_j)}
\end{aligned}$$

The following proof is analogous to the proof of Claim 7.2. An alternative proof appears in Appendix B.

Proof Sketch: W.l.o.g., we consider $i = 1$, and proceed analogously to the proof of Claim 7.2. Specifically, using Eq. (8), we observe that $\mathbb{E}_{j \sim q_1^{\bar{s}}}[v^{\bar{s}}(j)]$, which equals $\sum_{j \in [m]} q_1^{\bar{s}}(j) \cdot v(s_j)$, is the sum of m identical (alas somewhat dependent) random variables such that each ranges in $[0, 1]$. For each $j \in [m]$, the expectation of the j^{th} variable equals

$$\begin{aligned} \mathbb{E}_{\bar{s} \sim p^m} [q_1^{\bar{s}}(j) \cdot v(s_j)] &= \mathbb{E}_{s_1, \dots, s_m \sim p} \left[\frac{p_1(s_j)/p(s_j)}{\sum_{k \in [m]} (p_1(s_k)/p(s_k))} \cdot v(s_j) \right] \\ &= \mathbb{E}_{s_m \sim p} \mathbb{E}_{s_1, \dots, s_{m-1} \sim p} \left[\frac{p_1(s_m)/p(s_m)}{\sigma_1(s_1, \dots, s_m)} \cdot v(s_m) \right] \end{aligned}$$

Hence, for every value of s_m , it holds that

$$\mathbb{E}_{s_1, \dots, s_{m-1} \sim p} \left[\frac{p_1(s_m)/p(s_m)}{\sigma_1(s_1, \dots, s_m)} \cdot v(s_m) \right] = (1 \pm 2\varepsilon) \cdot \frac{p_1(s_m)/p(s_m)}{m} \cdot v(s_m)$$

which implies (for every $j \in [m]$)

$$\begin{aligned} \mathbb{E}_{\bar{s} \sim p^m} [q_1^{\bar{s}}(j) \cdot v(s_j)] &= (1 \pm 2\varepsilon) \cdot \mathbb{E}_{s_m \sim p} \left[\frac{p_1(s_m)/p(s_m)}{m} \cdot v(s_m) \right] \\ &= (1 \pm 2\varepsilon) \cdot \sum_{e \in [M]} \frac{p_1(e) \cdot v(e)}{m} \\ &= (1 \pm 2\varepsilon) \cdot \mathbb{E}_{e \sim p_1} [v(e)]/m. \end{aligned}$$

Using the same analysis as in the proof of Claim 7.2, the current claim follows. $\blacksquare$

Preservation of several functionalities. Note Claim 7.3 implies that the functionalities of $\exp(\text{poly}(n/\varepsilon))$ value functions can be preserved too. For each such value function $v : [M] \rightarrow [0, 1]$, we define the corresponding $v^{\bar{s}}$ and apply a union bound.

Reduction to the Boolean case. We note that a special case of Claim 7.3 in which the value function is binary implies the general case. This is the case because we can reduce the approximation of any value function $v : [M] \rightarrow [0, 1]$ to the approximation of the value functions $v_i : [M] \rightarrow \{0, 1\}$ (for $i \in [1/\varepsilon]$) such that $v_i(e) = 1$ if $v(e) \in ((i-1) \cdot \varepsilon, i \cdot \varepsilon]$ and $v_i(e) = 0$ otherwise. (Indeed, approximating a general value function up-to deviation ε requires approximating the Boolean functions up to ε^2 (or so)).²⁰

Claims 7.2 and 7.3 imply one another. Actually, as shown in Appendix B, these two claims imply one another. Furthermore, only Claim 7.3 is used in Section 7.2 (along with Claim 7.1). Nevertheless, we included the preservation of pairwise TVDs (i.e., Claim 7.2) in our exposition because we find it very appealing.

²⁰ A tighter result can be obtained by using threshold functions (i.e., $v_i(e) = 1$ if $v(e) > (i-0.5) \cdot \varepsilon$, and observing that $v(e) = \varepsilon \cdot \sum_{i \in [1/\varepsilon]} v_i(e) \pm \varepsilon$).

7.2 Proof of Theorem 1.4

The transformation presented in Section 7.1 preserves the complexity of the given distributions. However, in our application, the distributions in question are conditional distributions (i.e., they are conditioned on the verifier making certain queries and sending certain messages), which means that these conditional distributions may be more complex than the verifier's strategy (which determine its actions given the answers to prior queries and previous messages). For this reason, the verifier derived by our proof of Theorem 1.4 does not preserve the computational complexity of the original verifier.

We shall first present a slightly simplified version of the transformation (denoted “take 1”). Its analysis will expose a gap (see discussion following the statement of Claim 7.5), which will be addressed later (in “take 2”). Throughout this section we may assume that $r(n) \cdot q(n) = o(n/\log n)$, since Theorem 1.4 holds trivially otherwise.

The derived verifier, take 1. Recall that, on input $x \in \{0, 1\}^n$, the original verifier proceeds in $r(n)$ rounds such that in each round it makes some queries, sends a message to the prover, and receives a message from the prover.²¹ For each partial transcript γ of the communication in the first $i - 1$ rounds (of the original protocol), we consider all possible queries and answers that the (original) verifier might have made in the first i rounds (including the queries it makes in the i^{th} round). Denoting such a possible Q&A sequence by $\omega \in \Omega \stackrel{\text{def}}{=} \bigcup_{q' \leq q(n)} ([n] \times \{0, 1\})^{q'}$, we let $X_\omega^{(\gamma)}$ represent the distribution of the i^{th} message sent by the (original) verifier, stressing that this distribution depends on γ and ω . For each γ , applying the distribution compression scheme of Section 7.1 to the sequence $(X_\omega^{(\gamma)})_{\omega \in \Omega}$, we obtain a corresponding sequence of compressed distributions $(Y_\omega^{(\gamma, \bar{s}^{(\gamma)})})_{\omega \in \Omega}$, where $\bar{s}^{(\gamma)} \in [M]^m$ is the sample used by the compression scheme. These sequences of compressed distributions are hard-wired into the strategy of the derived verifier (by hard-wiring the corresponding $\bar{s}^{(\gamma)}$'s).²² The i^{th} message sent by the new verifier will be drawn from $Y_\omega^{(\gamma, \bar{s}^{(\gamma)})}$, where γ and ω are as above. Using $\varepsilon = 1/n = o(1/r(n))$, recall that $Y_\omega^{(\gamma, \bar{s}^{(\gamma)})}$ has support size $m = \text{poly}(|\Omega|/\varepsilon) = \text{poly}(n^{q(n)})$, which means that the length of the verifier's message (in each round) is $O(q(n) \cdot \log n)$.

Needless to say, upon receiving a message $j \in [m]$ (from the new verifier), the prover first recovers the corresponding s_j (i.e., the j^{th} element in the sequence $\bar{s}^{(\gamma)}$) and then invokes the original prover. Actually, this description refers to the honest prover used when establishing the completeness requirement; the cheating prover that is analyzed towards establishing the soundness requirement need not behave so.

Analysis of the derived verifier. In the analysis, we shall consider a fictitious protocol in which the compressed messages are not elements of $[m]$ but are rather elements of the corresponding sample (which is an m -sequence over $[M]$); that is, if the message $j \in [m]$ of the derived protocol is selected from $Y_\omega^{(\cdot, \bar{s})}$, where $\bar{s} = (s_1, \dots, s_m) \in [M]^m$, then the fictitious verifier sends the message s_j (rather than j). Accordingly, for $\bar{s}^{(\gamma)} = (s_1, \dots, s_m)$, we let $Z_\omega^{(\gamma, \bar{s}^{(\gamma)})}$ denote the distribution obtained by sampling j according to $Y_\omega^{(\gamma, \bar{s}^{(\gamma)})}$, and outputting s_j . Note that we may assume that,

²¹We may assume that $r(n) = o(n)$, since Theorem 1.4 holds trivially otherwise.

²²We do not care of the length of the description of the verifier's strategy, let alone that we have Lemma 3.1 at our disposal.

w.v.h.p., the elements in $\bar{s}^{(\gamma)}$ are distinct,²³ and so there is a 1-1 correspondance between $[m]$ and the elements in $\bar{s}^{(\gamma)}$. The foregoing convention facilitates a uniform presentation of hybrid distributions (as defined next).

We consider hybrid verifiers in which the first i messages are generated according to the compressed distributions and the remaining $r(n) - i$ messages are generated as in the original verifier (i.e., according to the original message-distributions), where $i \in \{0, 1, \dots, r(n)\}$. We first present notation for the last $r(n) - i$ rounds.

Fixing the input $x \in \{0, 1\}^n$, for each $i \in \{0, 1, \dots, r(n)\}$ and each γ (resp., ω) that represents the transcript of the communication (resp., the Q&A sequence) in the first i rounds, we consider the residual verifier $V_{\gamma, \omega}^x$ that describes the operation of the verifier in the $r(n) - i$ remaining rounds. Likewise, $P_\gamma(x)$ describes the operation of the prover in these $r(n) - i$ remaining rounds. The difference in notation reflects the fact that the verifier knows ω whereas the prover has only partial information about ω (via γ), and that the verifier has query access to the input x whereas the prover has direct access to it. Hence, we let $(P_\gamma(x), V_{\gamma, \omega}^x)$ denote the distribution of the verdict of V , on input x , where γ and ω are as above (i.e., the i -round communication transcript and the i -round Q&A sequence). Furthermore, we let $(P_\gamma(x), V_\gamma^x)$ denote the distribution of $(P_\gamma(x), V_{\gamma, \omega}^x)$ when $\omega \in \Omega$ is distributed conditioned on the communication transcript γ .

Next, for $i \in [r(n)]$, we consider the distribution of pairs (γ', ω) such that γ' represents the transcript of the communication in the first $i - 1$ rounds and ω representing a Q&A sequence for the first i rounds. In $i - 1^{\text{st}}$ hybrid, the i^{th} communication round consists by the verifier-message $\alpha \sim X_\omega^{(\gamma')}$ followed by the prover message $\beta \leftarrow B(x, \gamma', \alpha)$, where B is an optimal next-message strategy (for the prover in the $i - 1^{\text{st}}$ hybrid). In i^{th} hybrid, the i^{th} communication round consists by the verifier-message $\alpha \sim Z_\omega^{(\gamma', \bar{s}^{(\gamma')})}$, followed by the prover message $\beta \leftarrow B^{\bar{s}^{(\gamma')}}(x, \gamma', \alpha)$, where $B^{\bar{s}^{(\gamma')}}$ is an optimal next-message strategy (for the prover in the i^{th} hybrid). In both cases, we let $\gamma = (\gamma', \alpha, \beta)$, and consider the interaction of the residual verifier $V_{\gamma, \omega}^x$ with the residual prover $P_\gamma(x)$. We wish to relate these two hybrids when conditioned on a fixed value of γ' . We shall show that, w.v.h.p over the choice of $\bar{s}^{(\gamma')}$, the probability of accepting x in these two cases is approximately the same.

Claim 7.4 (completeness – lower bound on the i^{th} hybrid): *For any fixed $x \in \{0, 1\}^n$ and any fixed γ' that represents a transcript of the communication in the first $i - 1$ rounds, let ζ and ζ^+ denote the distributions generated for the $i - 1^{\text{st}}$ and i^{th} hybrids, respectively. Specifically, in both cases $\omega \in \Omega$, which represents a Q&A sequence for the first i rounds, is distributed conditioned on the communication transcript γ' , whereas $\zeta = (\gamma', \alpha, B(x, \gamma', \alpha))$, where $\alpha \sim X_\omega^{(\gamma')}$, and $\zeta^+ = (\gamma', \alpha, B^{\bar{s}^{(\gamma')}}(x, \gamma', \alpha))$ where $\alpha \sim Z_\omega^{(\gamma', \bar{s}^{(\gamma')})}$. Then, with probability at least $1 - \exp(-\Omega(m/n^{O(q(n))}))$ over the choice of $\bar{s}^{(\gamma')}$,*

$$\Pr[(P_{\zeta^+}(x), V_{\zeta^+}^x) = 1] \geq \Pr[(P_\zeta(x), V_\zeta^x) = 1] - O(\varepsilon). \quad (14)$$

We stress that both probabilities depend on the choice of ω that is conditioned by the same γ' .

Since the claim is stated for fixed x and γ' , we need to employ a union bound over all $x \in \{0, 1\}^n$ (but not on γ' because an independent sample $\bar{s}^{(\gamma')}$ is used for each γ'). This is not a problem, because $2^n \ll \exp(m/n^{O(q(n))})$, by our choice of m .

²³Specifically, we may augment all verifier's messages in the original protocol by long random strings (e.g., a distribution X over $[M]$ is replaced by a distribution X' over $[M] \times [M']$ such that $\Pr[X' = (j_1, j_2)] = \Pr[X = j_1]/M'$, where M' is huge even w.r.t M).

Proof: Intuitively, the current claim follows by the fact that the compression of distributions preserves functionality (i.e., Claim 7.3). This requires defining adequate value functions that can be related to the two sides of Eq. (14). For the i^{th} hybrid distribution the relation is only in one direction; this is the reason that the current claim only establishes an inequality in one direction.

For every $\alpha \in [M]$ and $\omega \in \Omega$ (which is possible when conditioning on γ'), let $v_{x,\gamma',\omega}(\alpha)$ denote the probability that $(P_{\gamma',\alpha,\beta}(x), V_{(\gamma',\alpha,\beta),\omega}^x) = 1$ where $\beta = B(x, \gamma', \alpha)$. Then,

$$v_{x,\gamma',\omega}(X_{\omega}^{(\gamma')}) = \Pr[(P_{\zeta}(x), V_{\zeta,\omega}^x) = 1]. \quad (15)$$

Letting $v_{x,\gamma',\omega}^{\bar{s}(\gamma')}(j) \stackrel{\text{def}}{=} v_{x,\gamma',\omega}(s_j)$, where s_j is the j^{th} element of $\bar{s}(\gamma')$, and using Claim 7.3, we infer that (for every $\omega \in \Omega$, w.v.h.p over the choice of $\bar{s}(\gamma')$) it holds that

$$v_{x,\gamma',\omega}^{\bar{s}(\gamma')}(Y_{\omega}^{(\gamma',\bar{s}(\gamma'))}) = v_{x,\gamma',\omega}(X_{\omega}^{(\gamma')}) \pm O(\varepsilon). \quad (16)$$

Lastly, for ω that is distributed as in the claim (i.e., the distribution of a Q&A sequence for the first i rounds that is conditioned on the communication transcript γ'), it holds that

$$\Pr[(P_{\zeta^+}(x), V_{\zeta^+}^x) = 1] \geq \mathbb{E}_{\omega}[v_{x,\gamma',\omega}^{\bar{s}(\gamma')}(Y_{\omega}^{(\gamma',\bar{s}(\gamma'))})] \quad (17)$$

because ζ^+ is defined by using an i^{th} prover message (i.e., $\beta = B^{\bar{s}}(x, \gamma', \alpha)$) that is optimized for $\bar{s}(\gamma')$ (and (x, γ', α)), whereas $v_{x,\gamma',\omega}^{\bar{s}(\gamma')} : [m] \rightarrow [0, 1]$ (which is based on $v_{x,\gamma',\omega} : [M] \rightarrow [0, 1]$) uses an i^{th} prover message that is optimized for (x, γ', α) obviously of $\bar{s}(\gamma')$. Combining Eq. (15)–(17), the claim follows. $\blacksquare$

Claim 7.5 (soundness – upper bound on the i^{th} hybrid, take 1): *For any fixed $x \in \{0, 1\}^n$ and γ' as in Claim 7.4, let ζ and ζ^+ denote the distributions generated for the $i - 1^{\text{st}}$ and i^{th} hybrids, respectively (as in Claim 7.4).²⁴ Then, with probability at least $1 - \eta$ over the choice of $\bar{s}(\gamma')$,*

$$\Pr[(P_{\zeta^+}(x), V_{\zeta^+}^x) = 1] \leq \Pr[(P_{\zeta}(x), V_{\zeta}^x) = 1] + O(\varepsilon/\eta) \quad (18)$$

(We will set $\eta = 1/n$ and $\varepsilon = 1/n^2$.)

Note that the probability that Eq. (18) holds is not as high as the probability that Eq. (14) holds. In particular, it will not suffice for a union bound on all possible $x \in \{0, 1\}^n$.

Proof: As hinted in the proof of Claim 7.4, the problem we face here is that the i^{th} prover message (in the i^{th} hybrid) may be selected depending on $\bar{s}(\gamma')$ (i.e., $\beta = B^{\bar{s}(\gamma')}(x, \gamma', \alpha)$). Specifically, the concern is that, due to $\bar{s}(\gamma')$, the prover in the i^{th} hybrid may have more information about ω than the prover in the $i - 1^{\text{st}}$ hybrid. Claim 7.2 seems to indicate that this is not the case, but it is not clear to us whether this route can work. An alternative approach, which we follow, is to convert a cheating prover for the i^{th} hybrid into a cheating prover for the $i - 1^{\text{st}}$ hybrid.

Specifically, recall that $\zeta^+ = (\gamma', \alpha, \beta)$ (which determines the optimal strategy P_{ζ^+} for the i^{th} hybrid) is determined by letting $\alpha \sim Z_{\omega}^{(\gamma', \bar{s}(\gamma'))}$ and $\beta \leftarrow B^{\bar{s}(\gamma')}(x, \gamma', \alpha)$. We consider a cheating prover, denoted $\tilde{P}$, for the $i - 1^{\text{st}}$ hybrid, that, given x, γ' and α (where $\alpha \sim X_{\omega}^{(\gamma')}$), selects $j \in [m]$

²⁴Recall that in both cases $\omega \in \Omega$ is distributed conditioned on the communication transcript γ' , whereas $\zeta = (\gamma', \alpha, B(x, \gamma', \alpha))$, where $\alpha \sim X_{\omega}^{(\gamma')}$, and $\zeta^+ = (\gamma', \alpha, B^{\bar{s}(\gamma')}(x, \gamma', \alpha))$ where $\alpha \sim Z_{\omega}^{(\gamma', \bar{s}(\gamma'))}$.

uniformly at random and $\bar{s} \sim p^m$ such that $s_j = \alpha$, and then uses $B^{\bar{s}}(x, \gamma', \alpha)$ as its message. (In later rounds, $\tilde{P}$ behaves like P_ζ and P_{ζ^+} .)

Clearly, the probability that $\tilde{P}$ convinces the verifier cannot exceed the probability that P_ζ (which is optimal for the $i - 1^{\text{st}}$ hybrid) does so. On the other hand, as shown next, the success probability of $\tilde{P}$ approximately equals the average of the success probabilities of P_{ζ^+} that correspond to different choices of $\bar{s}^{(\gamma')} \sim p^m$. Specifically, for a fixed ω , we consider the probability that the sample $\bar{s}^{(\gamma')}$ equals a specific $\bar{s} = (s_1, \dots, s_m)$ and that $Y_\omega^{\bar{s}} = j$ (equiv., $Z_\omega^{\bar{s}} = s_j$). In the straightforward process, which is reflected in the average (from P_{ζ^+} 's point of view), this event occurs with probability

$$\left(\prod_{k \in [m]} p(s_k) \right) \cdot \frac{p_\omega(s_j)/p(s_j)}{\sigma_\omega(\bar{s})} = \frac{p_\omega(s_j)}{\sigma_\omega(\bar{s})} \cdot \prod_{k \in [m] \setminus \{j\}} p(s_k) \quad (19)$$

whereas in the process underlying $\tilde{P}$ (where s_j should equal the verifier's message $\alpha \sim X_\omega$) this event occurs with probability

$$p_\omega(s_j) \cdot \frac{1}{m} \cdot \prod_{k \in [m] \setminus \{j\}} p(s_k). \quad (20)$$

Recalling that Claim 7.1 asserts that $\sigma_\omega(\bar{s}) = (1 \pm \varepsilon) \cdot m$ (w.v.h.p), it follows that Eq. (19) and Eq. (20) are a factor of at most $1 \pm \varepsilon$ apart. This means that, on the average (over the choice of $\bar{s}^{(\gamma')}$), the success probability of P_{ζ^+} is at most $O(\varepsilon)$ larger than the success probability of $\tilde{P}$.

The next observation is that, w.v.h.p over the choice of $\bar{s}^{(\gamma')}$, the success probability of P_{ζ^+} is at least $\Pr[(P_\zeta(x), V_\zeta^x) = 1] - O(\varepsilon)$, because (by Claim 7.4) almost the entire probability mass resides on choices of $\bar{s}^{(\gamma')}$ that satisfy this inequality. Applying Markov inequality, it follows that, with probability at least $1 - \eta$ over the choice of $\bar{s}^{(\gamma')}$, the success probability of P_{ζ^+} , where ζ^+ is determined based on $\bar{s}^{(\gamma')}$ (as recalled above), is at most $\Pr[(P_\zeta(x), V_\zeta^x) = 1] + O(\varepsilon/\eta)$. The claim follows. ■

The derived verifier, take 2. For a fixed γ' , we say that $\bar{s}^{(\gamma')}$ is *good* for $x \in \{0, 1\}^n$ if it satisfies Eq. (18) for this x . Claim 7.5 asserts that, for every $x \in \{0, 1\}^n$, a random $\bar{s}^{(\gamma')}$ is good with probability at least $1 - \eta$. However, we need an $\bar{s}^{(\gamma')}$ that is good for all x 's. The gap is bridged by “error reduction” to a level that can support a union bound; that is, the key observation is that, for every $x \in \{0, 1\}^n$, with probability greater than $1 - 2^{-2n}$, a random sequence of $n' \stackrel{\text{def}}{=} O(n/\eta)$ choices of $\bar{s}^{(\gamma')}$'s contains at least an $1 - 2\eta$ fraction of $\bar{s}^{(\gamma')}$'s that are good for x . Consequently, we can fix (and hard-wire) an n' -long sequence of $\bar{s}^{(\gamma')}$'s such that, for every $x \in \{0, 1\}^n$, at least an $1 - 2\eta$ fraction of the $\bar{s}^{(\gamma')}$'s are good for x . Furthermore, we may select this n' -long sequence such that all $\bar{s}^{(\gamma')}$'s in it satisfy Eq. (14). We call such an n' -long sequence of $\bar{s}^{(\gamma')}$'s *excellent* for γ' .

Now, we associate with each γ' an excellent n' -long sequence of m -sequences, and let the verifier select (in execution time) one of these m -sequences uniformly at random. Specifically, denoting this n' -long sequence by $\bar{s} = (\bar{s}^{(1)}, \dots, \bar{s}^{(n')})$, the verifier message in round i is a pair $(k, j) \in [n'] \times [m]$ such that k is uniformly distributed in $[n']$ and $j \sim Y_\omega^{(\gamma', \bar{s}^{(k)})}$, where ω and γ' are as above. Consequently, for every $x \in \{0, 1\}^n$, the sequence $\bar{s}^{(k)}$ selected by the verifier satisfies Eq. (18) with probability at least $1 - 2\eta$. Hence, for every x , with probability at least $1 - 2\eta$, the success probability of the prover increases by at most $O(\varepsilon/\eta)$ (and decreases by at most $O(\varepsilon)$).

Lastly, setting $\eta = 1/n = o(1/r(n))$ and $\varepsilon = 1/n^2 = o(\eta/r(n))$, we conclude that, for every $x \in \{0, 1\}^n$, the derived verifier has error probability that is at most $r(n) \cdot O(\varepsilon/\eta) = o(1)$ larger than the error probability of the original verifier. This establishes the main claim of Theorem 1.4. Furthermore, perfect completeness is preserved. Observing that the distributions $Y_{\omega}^{(\gamma, \bar{s}^{(\gamma)})}$'s can be approximated using $O(\log m) = O(q(n) \cdot \log n)$ random bits, we also establish the furthermore claim of Theorem 1.4.

An afterthought. We note that our analysis uses the preservation of value functions (i.e., Claim 7.3), but does not use the preservation of pairwise TVDs (i.e., Claim 7.2).²⁵ This may feel odd, since one may expect soundness to rely on the fact that the $Y_{\omega}^{\bar{s}}$'s do not reveal more information about ω than the X_{ω} 's, which seems related to Claim 7.2. Instead, Claim 7.5 uses Claim 7.4 (and Claim 7.1), whereas Claim 7.4 relies on Claim 7.3 (but not on Claim 7.2). However, as noted at the end of Section 7.1, it is actually the case that Claim 7.3 implies Claim 7.2. (Actually, the two claims imply one another.) Nevertheless, we included the preservation of pairwise TVDs in our exposition because we find it very appealing.

²⁵Claim 7.1 is also explicitly used, let alone that it is used in the proof of Claim 7.3 (as well as in the proof of Claim 7.2).

Appendix A: An alternative proof of Part of Theorem 4.1

For sake of potential interest, we reproduce here our initial proof of a special case of Theorem 4.1 that refers to the case of $\mathcal{C} = \mathcal{E}$.

This proof used Claim 4.1.1 (as is), which we restate as asserting that *there exists a Boolean circuit of size $\exp(O(t(n) \cdot \log n))$ that distinguishes between n -bit strings in Π and n -bit strings that are ϵ -far from Π* . In this alternative proof, Claim 4.1.2 is replaced by an alternative property that is the image of an adequate pseudorandom generator.²⁶

Specifically, using the ideas in [20] and [1, 17] (where [24] can replace [1, 17]),²⁷ we use the main hypothesis to present a pseudorandom set Π in $\mathcal{E}$, where pseudorandomness is wrt circuits of size $2^{\alpha' \cdot n}$ (for some constant $\alpha' > 0$, which will be determined in Claim A.1). This implies that $c(n) + q(n) = \Omega(t(n)) = \Omega(n/\log n)$. Details follow.

While the original construction of [20] is aimed at obtaining pseudorandom generators of large stretch (e.g., small exponential stretch), here we use the same ideas to obtain a pseudorandom generator that stretches its seed to a sequence that is (only) twice as long. The point in doing so is that the resulting generator $G : \{0, 1\}^{n/2} \rightarrow \{0, 1\}^n$ is pseudorandom wrt circuits of size $2^{\alpha' \cdot n}$. (In contrast, the generator in [20] is typically presented (cf. [4, Sec. 8.3.2]) as stretching $O(\log n)$ -bit long seeds into n -bit sequences that are pseudorandom wrt circuits of size n^2 (or n).) Specifically, we get –

Claim A.1 (implicit in [20], when combined with [1, 17]): *Suppose that there exists a set S in $\mathcal{E}$ such that for some constant $\alpha > 0$ and all but finitely many $n \in \mathbb{N}$ every family of $2^{\alpha \cdot n}$ -sized circuits fails to correctly decide S on some n -bit strings. Then, there exist a constant $\alpha' > 0$ and a function $G : \{0, 1\}^{n/2} \rightarrow \{0, 1\}^n$ that is computable in exponential-time such that no $2^{\alpha' \cdot n}$ -size circuit can distinguish between the uniform distribution over the multi-set $\{G(s) : s \in \{0, 1\}^{n/2}\}$ and the uniform distribution over $\{0, 1\}^n$.*

Letting $\Pi_n = \{G(s) : s \in \{0, 1\}^{n/2}\}$ and $\Pi = \bigcup_{n \in \mathbb{N}} \Pi_n$, the theorem follows. Specifically, by Claim 4.1.1, there exist $\exp(O(c(n) + q(n)) \cdot \log n)$ -sized circuits that distinguish between random strings in Π_n and uniformly distributed n -bit strings (since, w.v.h.p, the latter strings are ϵ -far from Π_n),²⁸ whereas by Claim A.1 these distributions are indistinguishable by $2^{\Omega(n)}$ -sized circuits. Hence, $c(n) + q(n) = \Omega(n/\log n)$ must hold.

Appendix B: Claims 7.2 and 7.3 imply one another

We start by showing that Claim 7.2 implies Claim 7.3, because the proof of the opposite direction is more technical and uses also Claim 7.1.

²⁶Here we use a pseudorandom generator (PRG) that is computed in exponential-time (i.e., a PRG of the type used for canonical derandomization [4, Sec. 8.3]), in contrast to the PRG that is computed in polynomial-time (i.e., a PRG of the type used in cryptography [4, Sec. 8.2]) that is used in Section 5.

²⁷See presentation in [4, Sec. 7.2].

²⁸This uses the fact that $2^{n/2} \cdot 2^{H_2(\epsilon) \cdot n} = o(2^n)$ for a sufficiently small $\epsilon > 0$ (e.g., $H_2(0.1) < 0.47$). We can actually support any $\epsilon < 1/2$ if we use a pseudorandom generator of larger (linear) stretch, which can (also) be constructed under the hypothesis of Claim A.1.

Claim 7.2 implies Claim 7.3. Using the fact that it suffices to consider Boolean value functions, we consider an arbitrary $v : [M] \rightarrow \{0, 1\}$ and the corresponding $v^{\bar{s}}(j) = v(s_j)$. Given the n distributions $X_1, \dots, X_n$ (over $[M]$), we consider also the distributions $X_{n+1}, \dots, X_{2n}$ such that X_{n+i} is the conditional distribution of X_i conditioned on $v(X_i) = 0$. Then, using $v(X_{n+i}) \equiv 0$, we have

$$\begin{aligned} \|X_{n+i} - X_i\|_1 &= 2 \cdot (\Pr[v(X_{n+i})=0] - \Pr[v(X_i)=0]) \\ &= 2 \cdot (1 - \Pr[v(X_i)=0]) \\ &= 2 \cdot \mathbb{E}[v(X_i)]. \end{aligned}$$

We now consider the distributions $Y_1^{\bar{s}}, \dots, Y_{3n}^{\bar{s}}$ that are defined by the compression scheme when applied to the distributions $X_1, \dots, X_{3n}$ (where $X_{2n+1}, \dots, X_{3n}$ are defined below). Using $v^{\bar{s}}(Y_{n+i}^{\bar{s}}) \equiv 0$, which holds by definition of $Y_{n+i}^{\bar{s}}$, we have

$$\begin{aligned} \|Y_{n+i}^{\bar{s}} - Y_i^{\bar{s}}\|_1 &\geq 2 \cdot (\Pr[v^{\bar{s}}(Y_{n+i}^{\bar{s}})=0] - \Pr[v^{\bar{s}}(Y_i^{\bar{s}})=0]) \\ &= 2 \cdot \mathbb{E}[v^{\bar{s}}(Y_i^{\bar{s}})]. \end{aligned}$$

Now, applying Claim 7.2 (i.e., $\|Y_{n+i}^{\bar{s}} - Y_i^{\bar{s}}\|_1 = \|X_{n+i} - X_i\|_1 \pm O(\varepsilon)$), we get

$$\mathbb{E}[v^{\bar{s}}(Y_i^{\bar{s}})] \leq \mathbb{E}[v(X_i)] + O(\varepsilon). \quad (21)$$

The opposite inequality is proved by considering also the distributions $X_{2n+1}, \dots, X_{3n}$ such that X_{2n+i} is the conditional distribution of X_i conditioned on $v(X_i) = 1$. In this case, we get

$$\begin{aligned} \|X_{2n+i} - X_i\|_1 &= 2 \cdot (\Pr[v(X_{2n+i})=1] - \Pr[v(X_i)=1]) \\ &= 2 \cdot (1 - \mathbb{E}[v(X_i)]) \end{aligned}$$

and

$$\begin{aligned} \|Y_{2n+i}^{\bar{s}} - Y_i^{\bar{s}}\|_1 &\geq 2 \cdot (\Pr[v^{\bar{s}}(Y_{2n+i}^{\bar{s}})=1] - \Pr[v^{\bar{s}}(Y_i^{\bar{s}})=1]) \\ &= 2 \cdot (1 - \mathbb{E}[v^{\bar{s}}(Y_i^{\bar{s}})]). \end{aligned}$$

Applying Claim 7.2, we get

$$\mathbb{E}[v^{\bar{s}}(Y_i^{\bar{s}})] \geq \mathbb{E}[v(X_i)] - O(\varepsilon). \quad (22)$$

Note that the distributions $Y_1, \dots, Y_n$ that are defined by the compression scheme when applied to the distributions $X_1, \dots, X_{3n}$ are typically different from the distributions $Y_1, \dots, Y_n$ that are defined by the compression scheme when applied to the distributions $X_1, \dots, X_n$.

Claim 7.3 implies Claim 7.2. As in the proofs of Claims 7.2 and 7.3, we use Claim 7.1, which asserts that, with probability at least $1 - \exp(-\Omega(\varepsilon/n)^2 \cdot m)$ over the choice of $\bar{s}$, for every $i \in [n]$, it holds that $\sigma_i(\bar{s}) = (1 \pm \varepsilon) \cdot m$. For every such $\bar{s} = (s_1, \dots, s_m)$, and for every $i \in [n]$ and $j \in [m]$, it holds that

$$q_i^{\bar{s}}(j) = \frac{p_i(s_j)/p(s_j)}{\sigma_i(\bar{s})} = \frac{1}{(1 \pm \varepsilon) \cdot m} \cdot \frac{p_i(s_j)}{p(s_j)} \quad (23)$$

where the p_i 's and the $q_i^{\bar{s}}$'s are as defined above (i.e., $p_i(e) \stackrel{\text{def}}{=} \Pr[X_i = e]$ and $q_i^{\bar{s}}(j) \stackrel{\text{def}}{=} \Pr[Y_i^{\bar{s}} = j]$). Now, for every $i_1, i_2 \in [n]$, we consider the value function $v_{i_1, i_2} : [M] \rightarrow [0, 1]$ such that

$$v_{i_1, i_2}(e) \stackrel{\text{def}}{=} \frac{|p_{i_1}(e) - p_{i_2}(e)|}{p_{i_1}(e) + p_{i_2}(e)}$$

Then,

$$\begin{aligned} \sum_{i \in \{i_1, i_2\}} \mathbb{E}[v_{i_1, i_2}(X_i)] &= \sum_{i \in \{i_1, i_2\}} \sum_{e \in [M]} p_i(e) \cdot \frac{|p_{i_1}(e) - p_{i_2}(e)|}{p_{i_1}(e) + p_{i_2}(e)} \\ &= \sum_{e \in [M]} |p_{i_1}(e) - p_{i_2}(e)| \\ &= \|p_{i_1} - p_{i_2}\|_1 \end{aligned}$$

Analogously, for $v_{i_1, i_2}^{\bar{s}}(j) \stackrel{\text{def}}{=} v_{i_1, i_2}(s_j)$, using Eq. (23), we have

$$\begin{aligned} \sum_{i \in \{i_1, i_2\}} \mathbb{E}[v_{i_1, i_2}^{\bar{s}}(Y_i^{\bar{s}})] &= \sum_{i \in \{i_1, i_2\}} \sum_{j \in [m]} q_i^{\bar{s}}(j) \cdot \frac{|p_{i_1}(s_j) - p_{i_2}(s_j)|}{p_{i_1}(s_j) + p_{i_2}(s_j)} \\ &= \sum_{i \in \{i_1, i_2\}} \sum_{j \in [m]} q_i^{\bar{s}}(j) \cdot \frac{|(1 \pm \varepsilon) \cdot q_{i_1}^{\bar{s}}(j) - (1 \pm \varepsilon) \cdot q_{i_2}^{\bar{s}}(j)|}{(1 \pm \varepsilon) \cdot q_{i_1}^{\bar{s}}(j) + (1 \pm \varepsilon) \cdot q_{i_2}^{\bar{s}}(j)} \\ &= \sum_{j \in [m]} (q_{i_1}^{\bar{s}}(j) + q_{i_2}^{\bar{s}}(j)) \cdot \frac{|q_{i_1}^{\bar{s}}(j) - q_{i_2}^{\bar{s}}(j)| \pm \varepsilon \cdot (q_{i_1}^{\bar{s}}(j) + q_{i_2}^{\bar{s}}(j))}{(1 \pm \varepsilon) \cdot (q_{i_1}^{\bar{s}}(j) + q_{i_2}^{\bar{s}}(j))} \\ &= \sum_{j \in [m]} \left(|q_{i_1}^{\bar{s}}(j) - q_{i_2}^{\bar{s}}(j)| \pm O(\varepsilon) \cdot (q_{i_1}^{\bar{s}}(j) + q_{i_2}^{\bar{s}}(j)) \right) \\ &= \|q_{i_1}^{\bar{s}} - q_{i_2}^{\bar{s}}\|_1 \pm O(\varepsilon). \end{aligned}$$

Using Claim 7.3 (i.e., $\mathbb{E}[v_{i_1, i_2}^{\bar{s}}(Y_i^{\bar{s}})] = \mathbb{E}[v_{i_1, i_2}(X_i)] \pm O(\varepsilon)$), we obtain $\|q_{i_1}^{\bar{s}} - q_{i_2}^{\bar{s}}\|_1 = \|p_{i_1} - p_{i_2}\|_1 \pm O(\varepsilon)$.

References

- [1] Laszlo Babai, Lance Fortnow, Noam Nisan and Avi Wigderson. BPP has Subexponential Time Simulations unless EXPTIME has Publishable Proofs. *Complexity Theory*, Vol. 3, pages 307–318, 1993.
- [2] Holger Dell, Thore Husfeldt, Daniel Marx, Nina Taslaman, and Martin Wahlen. Exponential time complexity of the permanent and the Tutte polynomial. *ACM Trans. Algor.*, Vol. 10 (4), 2014.
- [3] Eldar Fischer, Yonatan Goldhirsh, and Oded Lachish. Partial tests, universal tests and decomposability. In *5th ITCS*, pages 483–500, 2014.
- [4] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008.
- [5] Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.
- [6] Oded Goldreich. A digest of the work of Rothblum, Vadhan, and Wigderson (2013). *ECCC*, TR26-088, 2026.
- [7] Oded Goldreich and Johan Hastad. On the Complexity of Interactive Proofs with Bounded Communication. *Information Processing Letters*, Vol. 67 (4), pages 205–214, 1998.
- [8] Oded Goldreich and Maya Leshkowitz. On Emulating Interactive Proofs with Public Coins. *ECCC*, TR16-066, 2016.
- [9] Oded Goldreich and Bernd Meyer. Computational Indistinguishability – Algorithms vs Circuits. *Theoretical Computer Science*, Vol. 191, pages 215–218, 1998.
- [10] Oded Goldreich, Guy Rothblum, and Tal Skverer. On Interactive Proofs of Proximity with Proof-Oblivious Queries. *ECCC*, TR22-124, 2022.
- [11] Oded Goldreich, Salil Vadhan, and Avi Wigderson. On Interactive Proofs with a Laconic Prover. *Computational Complexity*, Vol. 11, pages 1–53, 2002.
- [12] Shafi Goldwasser and Michael Sipser. Private Coins versus Public Coins in Interactive Proof Systems. In *18th STOC*, pages 59–68, 1986.
- [13] Tom Gur and Ron Rothblum. Non-Interactive Proofs of Proximity. *Computational Complexity*, Vol. 27 (1), pages 99–207, 2018. Preliminary version in *ECCC*, TR13-078, 2013.
- [14] Tom Gur and Ron Rothblum. A Hierarchy Theorem for Interactive Proofs of Proximity. In *8th ITCS*, LIPIcs 67, pages 39:1–39:43, 2017.
- [15] Tom Gur, Yang Liu, and Ron Rothblum. An Exponential Separation Between MA and AM Proofs of Proximity. *Computational Complexity*, Vol. 30 (2), Art. 12, 2021.
- [16] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Science*, Vol. 62 (2), , pages 367–375, 2001.

- [17] Russell Impagliazzo and Avi Wigderson. P=BPP if E requires exponential circuits: Derandomizing the XOR Lemma. In *29th ACM Symposium on the Theory of Computing*, pages 220–229, 1997.
- [18] Yael Tauman Kalai and Ron D. Rothblum. Arguments of Proximity (Extended Abstract). In *35th CRYPTO*, pages 422–442, 2015.
- [19] Ilan Newman. Private vs Common Random Bits in Communication Complexity. *Information Processing Letters*, Vol. 39 (2), pages 67–71, 1991.
- [20] Noam Nisan and Avi Wigderson. Hardness vs Randomness. *Journal of Computer and System Science*, Vol. 49, No. 2, pages 149–167, 1994.
- [21] Guy N. Rothblum, Salil Vadhan, and Avi Wigderson. Interactive Proofs of Proximity: Delegating Computation in Sublinear Time. In *45th ACM Symposium on the Theory of Computing*, pages 793–802, 2013.
- [22] Guy Rothblum and Ron Rothblum. Batch Verification and Proofs of Proximity with Polylog Overhead. In *18th TCC (Part II)*, LNCS (Vol. 12551), Springer, pages 108–138, 2020.
- [23] Amit Sahai and Salil Vadhan. A Complete Problem for Statistical Zero Knowledge. *Journal of the ACM*, Vol. 50 (2), pages 196–249, 2003.
- [24] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR Lemma. *Journal of Computer and System Science*, Vol. 62, No. 2, pages 236–266, 2001.