Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 83 (2018)

An Exponential Separation Between MA and AM Proofs of Proximity

Tom Gur[*] UC Berkeley

Yang P. Liu MIT

Ron D. Rothblum[†] MIT and Northeastern University

May 8, 2018

Abstract

Interactive proofs of proximity allow a sublinear-time verifier to check that a given input is *close* to the language, using a small amount of communication with a powerful (but untrusted) prover. In this work we consider two natural *minimally interactive* variants of such proofs systems, in which the prover only sends a single message, referred to as the proof.

The first variant, known as MA-proofs of Proximity (MAP), is fully *non-interactive*, meaning that the proof is a function of the input *only*. The second variant, known as AM-proofs of Proximity (AMP), allows the proof to additionally depend on the verifier's (entire) random string. The complexity of both MAPs and AMPs is the total number of bits that the verifier observes – namely, the sum of the proof length and query complexity.

Our main result is an exponential separation between the power of MAPs and AMPs. Specifically, we exhibit an explicit and natural property Π that admits an AMP with complexity $O(\log n)$, whereas any MAP for Π has complexity $\tilde{\Omega}(n^{1/4})$, where n denotes the length of the input in bits. Our MAP lower bound also yields an alternate proof, which is more general and arguably much simpler, for a recent result of Fischer *et al.* (ITCS, 2014).

Lastly, we also consider the notion of *oblivious* proofs of proximity, in which the verifier's queries are oblivious to the proof. In this setting we show that AMPs can only be quadratically stronger than MAPs. As an application of this result, we show an exponential separation between the power of public and private coin for oblivious interactive proofs of proximity.

[*] Email: tom.gur@berkeley.edu. Research supported in part by the UC Berkeley Center for Long-Term Cybersecurity.

[†] Email: ronr@csail.mit.edu. Research supported in part by NSF Grants CNS-1350619 and CNS-1414119, Alfred P. Sloan Research Fellowship, Microsoft Faculty Fellowship and in part by the Defense Advanced Research Projects Agency (DARPA), the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236 and by the Cybersecurity and Privacy Institute at Northeastern University.

ISSN 1433-8092

1 Introduction

The field of property testing [RS96, GGR98] deals with sublinear algorithms for deciding whether a given object has a predetermined property or is far from any object having this property. Such algorithms, called testers, obtain local views of the object by performing queries; that is, the object is seen as a function and the tester receives oracle access to this function. The goal of the tester is to ascertain a global property of the function based only on its local view.

In the last couple of decades, the area of property testing has attracted much attention (see surveys [Ron08, Ron09, Can15] and recent textbook [Gol17]). However, while much success was found in designing testers for a myriad of natural properties, which only make a small number of queries, many other natural properties were shown to require a very large number of queries to test (often linear in the length of the input).

Proofs of proximity, first considered by Ergün, Kumar and Rubinfeld [EKR04], are both intrinsically interesting as a natural notion of proof systems for sublinear algorithms, as well as provide means to significantly reduce the number of queries that the tester needs to make in order to verify, rather than decide. These probabilistic proof systems can be viewed as augmenting testers with a help from a powerful, yet untrusted prover. In a recent line of works [RVW13, GR16, FGL14, GGR15, KR15, GG16b, RRR16, GR17, BRV17, CG17] various types of interactive [RVW13] and non-interactive proofs of proximity [GR18] were studied, including arguments of proximity [KR15], zero-knowledge proofs of proximity [BRV17], and proofs of proximity for distribution testing [CG17].

In this work we study the relation between two types of proofs of proximity that are *minimally interactive*; namely, MA and AM proofs of proximity, which can be viewed as the property testing analogue of the class MA (i.e., "randomized NP") and AM, respectively, and are described in more detail next.

Informally speaking, an MA proof of proximity (MAP) protocol consists of a tester (or rather a verifier) that receives oracle access to an input function f but also receives explicit access to a short purported proof w. Based on the proof string and a few oracle queries to f, the verifier should decide whether f has some property Π (i.e., whether $f \in \Pi$). More specifically, after reading the proof w, the verifier tosses random coins, makes queries to the oracle f, and decides whether to accept or reject. We require the following completeness and soundness conditions: if $f \in \Pi$, then there exists a proof w that the verifier accepts with high probability, and if f is "far" (in Hamming distance) from any function in Π, then the verifier rejects with high probability. Following the literature, the complexity of an MAP is the total number of bits that the verifier observes - namely, the sum of its proof length and query complexity.[1]

The reason that the foregoing model is referred to as a "Merlin-Arthur" protocol is that we think of the prover as being Merlin (the all powerful magician) and the verifier as Arthur

[1] Alternatively, one could view the running time of the verifier (which serves as an upper bound on the query and communication complexities) as the main resource to be minimized. However, for simplicity (and following the property testing literature), we focus on combinatorial resources. This only makes our lower bounds stronger, whereas for all of our upper bounds the verifier is also computationally efficient.

(a mere mortal). Then in the MAP model Merlin "speaks" first (i.e., sends the proof) and Arthur "speaks" second (i.e., tosses his random coins).

It is natural to ask what happens if we switch the order - letting Arthur toss his coins first and Merlin send his proof after seeing Arthur's coin tosses. This type of protocol is typically referred to as an "Arthur-Merlin" protocol. More precisely, an AM proof of proximity (AMP) is defined similarly to an MAP, except that now the proof oracle is a function of the verifier's entire random string. We view AMPs as *minimally interactive*, since given a common random string, the protocol consists of a single message (i.e., the proof). Analogously to MAPs, the complexity of an AMP is the sum of its proof length and query complexity. We emphasize that the prover's message can depend on all of the verifier's coin tosses. Namely, the verifier cannot toss an additional coins after receiving its message from the prover.[2]

While the difference between these two proof systems may appear minor, MA-type and AM-type proofs naturally admit very different types of strategies. In particular, note that AM proofs provide the additional power of allowing the prover and verifier to jointly restrict their attention to a random subset of the input function's domain. On the other hand, the AM model also significantly hampers the power of the verifier to detect malicious prover strategies, since the prover knows the entire randomness of the verifier, and in particular the prover knows which queries the verifier will make.

At first glance, it may seem that AMPs are extremely limited, since the prover can predict exactly what the verifier will check (knowing the verifier's *entire* random string allows the prover to deduce which queries the verifier will make). However, it turns out that a straightforward adaptation of the classical MA $\subseteq$ AM inclusion [BM88] implies that any MAP can be emulated by an AMP at a quadratic cost. (More precisely, an MAP with proof complexity p and query complexity q can be emulated by an AMP with proof length p and query complexity $O(p \cdot q)$.[3])

It is natural to ask the following converse question:

Can any AMP *protocol be emulated by an* MAP*, or is there a gap between the power of these two models?*

Note that any AMP can be easily emulated by an MAP with at most an exponential overhead.[4] Thus, the question that we would really like to answer is whether such an exponential overhead is inherent.

[2] In contrast, the complexity class AM is sometimes defined as any constant-round public-coin interactive proof-system. Indeed, if one does not care about polynomial factors, then by a result of Babai and Moran [BM88], any public-coin constant-round interactive proof can be reduced to just 2 messages.

[3] The idea is to first reduce the soundness error of the MAP to $2^{-O(p)}$ (by repetition). Now suppose that the verifier reveals its randomness to the prover before receiving the proof-string. For soundness, observe that when f is far from the having the property, for any fixed proof-string the probability that the verifier would accept is at most $2^{-\Omega(p)}$ and so by a union bound, with high probability there simply does not exist a proof-string that will make the verifier accept.

[4] Any AMP with proof complexity p and query complexity q can be emulated by a *tester* (i.e., an MAP which does not use a proof at all) with query complexity $q \cdot 2^p$ by simply trying all possible candidate proof strings.

1.1 Our results

Our main result shows that it is indeed the case that AMPs can be *exponentially* stronger than MAPs and so the foregoing emulation strategy is optimal, up to polynomial factors:

Theorem 1. *There exists a property* $\Pi \subseteq \{f : [n] \to [n]\}$ *such that:*

- Π *has an* AMP *of complexity* $O(\log(n)/\varepsilon)$*, with respect to proximity parameter* $\varepsilon > 0$*; and*
- *Every* MAP *for* Π*, with respect to proximity parameter* $\varepsilon \leq \frac{1}{10}$*, must have complexity* $\Omega(n^{\frac{1}{4}})$*.*

The property Π that we use to prove Theorem 1 is actually very simple and natural. Specifically, Π is the set of all permutations over $[n]$; the goal of the verifier is to check whether a given function $f : [n] \to [n]$ is close to being a permutation by querying the function in a few locations and with a short interaction with the prover.

The AMP protocol for deciding whether a given function $f : [n] \to [n]$ is a permutation is extremely simple. The idea is that the random string specifies some random element $y \in [n]$ and the prover should specify an inverse x of y (under f). If f is a permutation such an element must exist whereas if f is ε-far from being a permutation, then with probability ε it holds that y simply does not have an inverse. We can repeat the base protocol $O(1/\varepsilon)$ times to get constant soundness error. This protocol can actually be traced back to a result of Bellare and Yung [BY96] who used it resolve a gap in the [FLS99] construction of non-interactive zero-knowledge proofs for NP based on trapdoor permutations.

Our MAP lower bound is the technically more challenging part of this work, and is actually a special case of a more general MAP lower bound that we prove. We show that any property that satisfies a relaxed notion of k-wise independence requires MAPs with complexity roughly $\sqrt{k}$. This result generalizes a recent result of Fischer, Goldhirsh and Lachish [FGL14] which can be interpreted as an MAP lower bound of $\sqrt{k}$ for properties that are exactly k-wise independent.[5] Our proof is also (arguably) significantly simpler than that of [FGL14] and in particular uses only elementary arguments, see further discussion in Section 1.2.

1.1.1 Oblivious Proofs of Proximity

Having established Theorem 1, we revisit the MA versus AM problem within the context of proofs of proximity *with prover-oblivious queries* (a notion first considered in [RVW13] and further explored in [GR18]), or in short *oblivious proofs of proximity*. These are proofs of proximity that have the special feature that the queries that the verifier makes are independent of the proof. Viewed from a temporal perspective, in these proof systems the verifier *first* makes its queries to the input function, and only after making all of its queries does it receive the proof. One reason that makes this feature appealing is because it allows the verifier to probe the object and obtain a certificate, which can then be used later

[5] More precisely, [FGL14] show that any linear code with large dual distance requires MAPs of complexity that is roughly square root of the code's blocklength.

when interacting with a prover, even if the object is no longer accessible. Another reason is that many of the interactive proof systems from the literature (e.g., the sumcheck protocol of [LFKN92]) are *oblivious*.

Surprisingly, it turns out that the gap between the power of *oblivious* AMPs and MAPs is dramatically smaller than the one exhibited in Theorem 1. Loosely speaking, we show that oblivious AMPs can only be *quadratically* stronger than oblivious MAPs, and in fact, standard testers (that do not use a proof).

Theorem 2. *For any property Π, if there exists an* oblivious AMP *for Π with proof complexity p and query complexity q, then there also exists a tester (i.e.,* MAP *with proof complexity 0) for Π with query complexity $O(p \cdot q)$.*

As an application, we use Theorem 2 to derive lower bounds on public-coin oblivious interactive proofs of proximity, and show an exponential separation between public-coin and private-coin protocols in this setting. See further discussion in Section 5.

1.2 Related works

The notion of proofs of proximity was originally proposed by Ergün, Kumar and Rubinfeld. Ben Sasson *et al.* [BGH$^+$06] and Dinur and Reingold [DR06] considered such proofs in the context of PCPs. Rothblum, Vadhan and Wigderson [RVW13], considered *interactive* proofs of proximity and showed that every language computable by a low-depth circuit has an interactive proof of proximity (IPP) with a sublinear time verifier. Reingold, Rothblum, and Rothblum [RRR16] showed *constant-round* IPPs for any language computable in polynomial-time and bounded polynomial-space. Goldreich and Gur [GG16a, GG16b] showed general-purpose IPP with only 3 rounds, albeit for a much smaller class.

Proofs of proximity were further studied by [GGR15] who showed more efficient constructions for certain restricted complexity classes, such as functions accepted by small read-once branching programs and context-free languages. Gur and Rothblum proved a round hierarchy theorem for IPPs [GR17], showing that the power of IPPs gradually increases with the number of rounds of interaction. Several works focused on studying non-interactive (MA) proofs of proximity [GR18, FGL14, GGK15] (see also [Gur17]). In addition, recent works studied (computationally sound) interactive *arguments* of proximity [KR15], zero-knowledge proofs of proximity [BRV17], and proofs of proximity for distribution testing [CG17]. Proofs of proximity have also found applications to property testing and related models [GR16, FLV15, GR17]. We remark that a concurrent work of Berman *et al.* [BRV17], utilizes our results (specifically Theorem 1) to derive a separation between the power of MAPs and zero-knowledge IPPs.

The notion of MA and AM proofs plays a central role in the study of proofs system in various computational models, other than in the setting of polynomial-time Turing machines in which they were originally conceived [BM88]. For example, in quantum computation, the class QMA (quantum MA proofs) captures the most fundamental type of quantum proof systems (since quantum algorithms are inherently randomized) and it has been extensively studied in the last couple of decades (see survey [AN02]). Of particular relevance, Aaronson

[Aar12] considered the problem of deciding whether a function is close to a permutation to derive a quantum query complexity separation between the class QMA and the class of statistical zero knowledge SZK, showing that every QMA query complexity algorithm with a w-qubit witness and query complexity q must satisfy $q + w = \Omega(n^{1/6})$.

In addition, MA and AM proof systems received much attention in the setting of communication complexity [BFS86, Kla03, RS04, Kla11, GPW15, She16] and streaming algorithms [CMT10, CTY11, CCMT14, GR15, CCM+15, Tha16]. The former also has an interesting connection to the algebrization barrier [AW09] and recently found important applications to distributed PCPs and hardness of approximation [ARW17].[6] The latter can be viewed as the property testing analogue of online annotated data streams (there, instead of oracle access to the input, the algorithm has one-pass sequential access to the input, and the goal is to minimize *space* complexity rather than *query* complexity). Indeed, part of our results concerning oblivious proofs of proximity are inspired by the techniques for online annotated data streams in [CCM+15].

Perhaps most relevant to us, the notion of MA and AM proofs for decision tree complexity (or the "query complexity model"), which can be thought of as property testing for *exact* (rather than approximate) decision problems, is closely related to proofs of proximity, though the query complexity model is much simpler to analyze than property testing. We remark that the high-level approach of our main lower bound for MAPs is inspired by the work of Raz *et al.* [RTVV98].

Comparison with the techniques in [FGL14]. As we discussed above, our MAP lower bound generalizes the main result of Fischer, Goldhirsh, and Lachish [FGL14]. The latter result can be interpreted as an MAP lower bound for any k-wise independent property. Our lower bound extends to a natural generalization of this family. We stress that this extension is *crucial* for our main result, as the permutation property (with respect to which we prove Theorem 1) is *not* k-wise independent, but does satisfy our more general notion.[7]

The proof in [FGL14] is technically quite involved and includes several subtle and nontrivial arguments. For example, while typically property testing lower bounds are shown by exhibiting two distributions that are chosen only as a function of the property, the argument in [FGL14] crucially relies on distributions that are functions of both the property and the description of the specific analyzed algorithm. This entails the usage of several complex mechanisms. For example, they rely on an involved treatment of adaptivity, which consists

[6] We remark that there are several similarities between MA and AM proof systems in the setting of property testing and communication complexity. In particular, simulating MA communication complexity protocols by their AM counterparts can also be done while only incurring a quadratic blow-up in complexity, and on the other hand AM protocols can also be exponentially more powerful than MA protocols [Kla11]. In addition, oblivious MA proofs of proximity can be viewed as analogous to *online* MA communication complexity protocols [CCM+15].

[7] Jumping ahead, we remark that our relaxed notion of k-wise independence refers to distributions for which the probability that any subset of k indices is equal to any given sequence of k values is upper bounded by the same probability given the uniform distribution *up to a multiplicative constant* (whereas the standard (i.e., non-relaxed) notion requires exact equality). See further details in Section 3.

of procedures for "grafting" decision trees, and use a special type of algorithms (called "readers") that expose low-entropy portions. Perhaps the most significant complication is that their argument uses a delicate information theoretic analysis to handle MAPs that have a two-sided error.

In contrast, our proof is much shorter and consists purely of a combinatorial argument, which does not require any special treatment of adaptivity and two-sided error, and does not use information theory.

1.3 Organization

In Section 2, we introduce the notations and definitions that we use throughout this work. In Section 3, we prove our main technical contribution, which is an MAP lower bound for relaxed k-wise independent properties. In Section 4, we derive our main result: an exponential separation between MAPs and AMPs. In Section 5, we present and prove our results regarding oblivious proofs of proximity. Finally, in Section 6, we conclude with a discussion and raise open problems.

2 Preliminaries

In this section we establish the definitions and notions that we will need throughout this work.

2.1 Properties and Distance

We focus on testing properties of *functions* and identify a "property" with the set of functions having that property. More accurately, for each $n \in \mathbb{N}$, let D_n and R_n be sets. Let $\mathcal{F}_n$ be the set of functions from D_n to R_n. We define a property as an ensemble $\Pi = \bigcup_n \Pi_n$, where $\Pi_n \subseteq \mathcal{F}_n$ for all n.

For an alphabet Σ, we denote the Hamming distance between two strings $x, y \in \Sigma^n$ by $\Delta(x,y) \coloneqq |\{x_i \neq y_i : i \in [n]\}|$. If $\Delta(x,y) \leq \varepsilon \cdot n$, we say that x is ε-close to y, otherwise we say that x is ε-far from y. For a non-empty set $S \subseteq \Sigma^n$, we similarly define $\Delta(x,S) \coloneqq \min_{y \in S} \Delta(x,y)$. Again, if $\Delta(x,S) \leq \varepsilon \cdot n$, we also say that x is ε-close to S and otherwise x is ε-far from S. We extend these definitions to functions by identifying functions with their truth tables (viewed as strings).

Integrality. Throughout this work, for simplicity of notation, we use the convention that all (relevant) integer parameters that are stated as real numbers are implicitly rounded to the closest integer.

2.2 Proofs of Proximity

We recall the definitions of MA and AM proofs of proximity (i.e., MAPs and AMPs), following [GR18]. Throughout, for an algorithm V we denote by $V^f(n, \varepsilon, w)$ the output of V given oracle access to a function f and explicit access to inputs n, ε, and w; if V is a probabilistic algorithm, we write $\Pr[V^f(n, \varepsilon, w) = z]$ to represent the probability *over the internal randomness of* V that this outcome is z.

Definition 2.1 (MAP)**.** *A* Merlin-Arthur proof of proximity (MAP) *for a property* $\Pi = \bigcup_n \Pi_n$ *consists of a probabilistic algorithm* V*, called the* verifier, *that is given as explicit inputs an integer* $n \in \mathbb{N}$*, a proximity parameter* $\varepsilon > 0$*, and a proof string* $w \in \{0,1\}^*$*; in addition, it is given oracle access to a function* $f \in \mathcal{F}_n$*. The verifier satisfies the following conditions.*

1. **Completeness:** *For every* $n \in \mathbb{N}$ *and* $f \in \Pi_n$*, there exists a string* w *(the proof) such that for every* $\varepsilon > 0$ *the verifier accepts with high probability; that is,*

$$\Pr\left[V^f(n, \varepsilon, w) = 1\right] \geq \frac{2}{3}.$$

2. **Soundness:** *For every* $n \in \mathbb{N}$*, function* $f \in \mathcal{F}_n$*, string* w*, and proximity parameter* $\varepsilon > 0$*, if* f *is* ε*-far from* Π_n*, then the verifier rejects with high probability; that is,*

$$\Pr\left[V^f(n, \varepsilon, w) = 0\right] \geq \frac{2}{3}.$$

A MAP is said to have query complexity $q : \mathbb{N} \times \mathbb{R}^+ \to \mathbb{N}$ if for every $n \in \mathbb{N}, \varepsilon > 0$, $f \in \mathcal{F}_n$, and string $w \in \{0,1\}^*$, the verifier reads at most $q(n, \varepsilon)$ bits in its queries to f. We say that a MAP has proof complexity $p : \mathbb{N} \to \mathbb{N}$ if for every $n \in \mathbb{N}$, there always exists a $w \in \{0,1\}^{p(n)}$ satisfying the conditions of Definition 2.1. We define the complexity of the MAP to be $t(n, \varepsilon) = q(n, \varepsilon) + p(n)$.

Next, we define AM proofs of proximity (AMPs) similarly to MAPs, except that here the proof is also a function of the inner randomness of the verifier (alternatively, the verifier first sends the prover its entire random string).

Definition 2.2 (AMP)**.** *An* Arthur-Merlin proof of proximity (AMP) *for a property* $\Pi = \bigcup_n \Pi_n$ *consists of a probabilistic algorithm* V*, called the* verifier, *that is given as explicit inputs an integer* $n \in \mathbb{N}$*, a proximity parameter* $\varepsilon > 0$*, and a proof string* w *that* depends on the verifier's random string r*, as well as oracle access to a function* $f \in \mathcal{F}_n$*. The verifier must also be deterministic given the random string* r*. The protocol satisfies the following conditions.*

1. **Completeness:** *For every* $n \in \mathbb{N}$ *and* $f \in \Pi_n$*,*

$$\Pr_r\left[\exists w = w(r) \text{ such that } V^f(n, \varepsilon, w; r) = 1\right] \geq \frac{2}{3}.$$

2. **Soundness:** *For every $n \in \mathbb{N}$, function $f \in \mathcal{F}_n$, and proximity parameter $\varepsilon > 0$, if f is ε-far from Π_n, then:*

$$\Pr_r \left[\exists w \text{ such that } V^f(n, \varepsilon, w; r) = 1\right] \leq \frac{1}{3}.$$

Analogously to MAPs, an AMP is said to have *query complexity* $q : \mathbb{N} \times \mathbb{R}^+ \to \mathbb{N}$ if for every $n \in \mathbb{N}, \varepsilon > 0$, $f \in \mathcal{F}_n$, and string $w \in \{0,1\}^*$, the verifier reads at most $q(n, \varepsilon)$ bits in its queries to f; and *proof complexity* $p : \mathbb{N} \times \mathbb{R}^+ \to \mathbb{N}$ if for every $n \in \mathbb{N}$ and $f \in \mathcal{F}_n$, with probability at least $\frac{2}{3}$ over coin tosses in the first round, there exists a $w \in \{0,1\}^{p(n,\varepsilon)}$ satisfying the completeness condition of Definition 2.2. We define the complexity of the AMP to be $t(n, \varepsilon) = q(n, \varepsilon) + p(n, \varepsilon)$.

We note that we do not include the randomness complexity of the verifier in the complexity of the protocol (although the randomness complexity in all the protocols described in this work is not large). This is a similar choice to what is done in similar contexts such as AM query and communication complexities. Moreover, we show in Appendix A (see Theorem 6) that if a property Π of functions $f : D_n \to R_n$, such that $|R_n|^{|D_n|} = O(\exp(\text{poly}(n)))$, admits an AMP verifier with query complexity q and proof complexity p, then it also admits an AMP verifier with query complexity $O(q)$, proof complexity $O(p)$, and randomness complexity $O(\log n)$.[8] This transformation is similar to known results of Newman [New91] in the context of communication complexity, Goldreich and Sheffet [GS10] in the context of property testing, and Gur and Rothblum [GR18] for MAPs. Its main disadvantage however is that it does not preserve the *computational* complexity of the verifier.

3 MAP Lower Bound for (Relaxed) k-wise Independence

In this section we show a general MAP lower bound for a large class of properties. More specifically, we show that any MAP for a (non-degenerate) property that is k-wise independent, must have complexity $\Omega(\sqrt{k})$. By a k-wise independent property we mean that if we sample a random element having the property, than its restriction to any k coordinates looks uniform. As mentioned in the introduction, this generalizes a result due to Fischer *et al.* [FGL14].

We would like to apply this lower bound to the permutation property. However, the permutation property is not k-wise independent and so we cannot apply it directly.[9]. Rather, we give a relaxed notion of k-wise independence that does capture the permutation property and for which we can similarly derive an MAP lower bound.

We proceed to define our relaxed notion of k-wise independence. Recall that we use $\mathcal{F}_n$ to denote the set of all functions from D_n to R_n (see Section 2).

[8] For most properties, we have that both the domain and range have size that is polynomial in n. Indeed, the case that $|R_n|^{|D_n|} = \omega(\exp(\text{poly}(n)))$ seems quite pathological.

[9] Indeed, it is not even pairwise independent: the chance of seeing the same element twice is *zero*.

Definition 3.1 (Relaxed k-wise Independence)**.** *Let $\Pi = \bigcup_{n\geq 1} \Pi_n$ be a property, where $\Pi_n \subset \mathcal{F}_n$ for every n. We say that Π is* relaxed k-wise independent*, for $k = k(n)$, if there exists a constant $C \geq 1$ such that for all positive integers n, all pairwise distinct k-tuples $(i_1, i_2, \ldots, i_k) \in (D_n)^k$ and arbitrary $(t_1, t_2, \ldots, t_k) \in (R_n)^k$, we have that*

$$\Pr_{f\in\Pi_n}\Big[f(i_j) = t_j \textit{ for all } j \in [k]\Big] \leq \frac{C}{|R_n|^k}. \tag{1}$$

Note that standard definition of a k-wise independence corresponds to the special case of Definition 3.1 when $C = 1$ (in which case the inequality in Eq. (1) can be replaced with an equality).

At first glance it may seem that the relaxation that we allow in Definition 3.1 is relatively minor and any lower bound that holds for the full-fledged definition should easily be extendable to our relaxed variant. We argue that it is not the case. For example, in a seminal work, Braverman [Bra11] showed that any k-wise independent distribution (for k that is poly-logarithmic) fools AC_0 circuits. Now consider the permutation property (to be defined formally in Section 4) which as noted above is not even pairwise independent but does satisfy our relaxed variant (with $k = \sqrt{n}$). It is not too hard to see that there is a very simple AC_0 circuit for checking whether a function is a permutation: simply by checking whether there exist a pair of entries in the truth table that are identical - thus, our seemingly minor relaxation completely sidesteps Braverman's result. As a matter of fact, a similar situation occurs in the context of AMPs: Rothblum *et al.* [RVW13] showed an AMP lower bound for exact k-wise independent distribution, whereas we show a protocol for the permutation property with logarithmic complexity.

Having defined our notion of relaxed k-wise independence, we proceed to describe a second important condition that we require: namely, that the property is *sparse*, in the sense that a random function is far from the property. Sparsity is essential for our result since there are trivial properties that are k-wise independent but are testable with very few queries (e.g., the property that consists of all functions).

Definition 3.2 (Sparse Property)**.** *Fix the proximity parameter $\varepsilon = \frac{1}{10}$. We say that a property $\Pi_n = \bigcup_{n\in\mathbb{N}} \Pi_n$ is $t(n)$-sparse if:*

$$\Pr_{f\in\mathcal{F}_n}[f \textit{ is } \varepsilon\textit{-far from } \Pi_n] \geq 1 - |R_n|^{-t(n)}.$$

We can now state our main theorem for this section.

Theorem 3. *Let Π be a relaxed k-wise independent and k-sparse property. Then, any* MAP *for Π, with respect to proximity parameter $\varepsilon = 1/10$, with proof complexity p and query complexity q must satisfy $p \cdot q = \Omega(k)$.*

The intuition and high level approach for the proof are as follows. First, we use the duality of an MAP as a collection of partial testers [FGL14]. More specifically, the existence of an MAP for a property Π implies that there is some large "sub-property" $\Pi' \subseteq \Pi$ and a *tester* T that distinguishes between inputs in Π' from those that are far from Π.

This simple observation reduces lower bounding MAPs for Π to lower bounding a partial tester for an arbitrary, but large, sub-property. To show such a lower bound, consider the uniform distribution on Π' vs. the uniform distribution over functions that are far from Π. We would like to argue that these two distributions look the same to T, which therefore cannot distinguish between them.

As a matter of fact, we will argue that both these distributions are "close" to being k-wise independent, which suffices as long as k is larger than the tester's query complexity. First, by the sparsity condition we have that the uniform distribution over functions that are far from Π is close to the uniform distribution over *all* functions. Clearly the latter is k-wise independent.

As for the uniform distribution over Π', we would like to argue that since Π' covers a substantial part of Π, which is relaxed k-wise independent, then also Π' is relaxed k-wise independent. The problem with this argument is that Π' only consists of a 2^{-p} fraction of Π, and so it could be quite far from being even relaxed k-wise independent (e.g., it could be that the value of functions in Π' on some fixed elements of R_n is constant over all functions in Π').

This seems like a significant difficulty and was overcome using highly elaborate techniques in [FGL14]. In contrast, we suggest a much simpler argument. The idea is that we first reduce the soundness error of the MAP to $2^{-O(p)}$ by repetition. This increases the query complexity of the tester to $O(p \cdot q)$ but now that the soundness error is so small, that the fact that Π' covers a 2^{-p} fraction of Π is sufficient to make the argument go through.

We proceed to the actual proof.

3.1 Proof of Theorem 3

Let C be a constant such that Π_n satisfies the constraints of Definition 3.1.

Let V be an MAP verifier, with respect to proximity parameter ε, for Π_n, and denote its proof complexity by p and query complexity by q. Note that any MAP with standard $2/3$ completeness and soundness probability (as in Definition 2.1) can be amplified, via $O(p)$ repetitions, to have completeness and soundness errors $\frac{1}{10C} \cdot 2^{-p}$ at the cost of increasing the query (but not the proof) complexity by a multiplicative factor of $O(p)$, to $O(p \cdot q)$. For concreteness, let us fix a constant C' such that a $(C' \cdot p)$-fold repetition of V has completeness and soundness errors $\frac{1}{10C} \cdot 2^{-p}$ (while having proof complexity p and query complexity $C' \cdot p \cdot q$). Assume towards a contradiction that $p \cdot q \leq \frac{k}{10C'}$.

Recall that for $\Pi' \subseteq \Pi$, a (Π, Π')-partial tester (a notion due to [FGL14]) is a tester that is required to accept functions in the subset Π' and reject functions that are ε-far from the superset Π. As pointed out by Fischer *et al.* [FGL14] an MAP as we assumed above, implies a covering of the property by partial testers as follows. For every possible proof string $w \in \{0,1\}^p$, let

$$S_w = \Big\{ f \in \Pi_n \ : \ \Pr\left[V^f(n, \varepsilon, w) = 1\right] \geq 1 - \frac{1}{10C} \cdot 2^{-p} \Big\}.$$

By the completeness requirement of an MAP, these sets cover the property Π_n. That is, $\bigcup_w S_w = \Pi_n$.

Since the number of sets S_w is at most 2^p, there exists a proof w that corresponds to a large S_w. Namely, such that $|S_w| \geq |\Pi_n| \cdot 2^{-p}$. We fix such a proof w and argue that the corresponding (Π_n, S_w)-partial tester must make $\Omega(k)$ queries, which would contradict our assumption, thereby proving Theorem 3. Hence, we have reduced proving an MAP lower bound for Π_n to proving a partial testing lower bound for (Π_n, S_w).

Let $V_w^f(n,\varepsilon) \coloneqq V^f(n,\varepsilon,w)$ be the (S_w,Π_n)-partial tester that is induced by V when we fix the proof string w (and with respect to parameters n and ε). We use the notation $V_w^f(n,\varepsilon;r)$ to denote the *deterministic* output V_w^f when its random string is set to r.

Let $B_\varepsilon = \{f \in \mathcal{F}_n : f \text{ is } \varepsilon\text{-far from } \Pi_n\}$ (i.e., the no-instances). As standard in the property testing literature, we prove a lower bound on the query complexity q' of a tester by presenting a distribution over YES-instances ($f \in S_w$) and a distribution over NO-instances ($f \in B_\varepsilon$) and bounding away from 1 the distinguishing probability for every *deterministic* algorithm making q' queries. Specifically, we give distributions over S_w and B_ε such that any deterministic algorithm making q' queries to f has at most a $1 - \frac{1}{4C} \cdot 2^{-p}$ probability of distinguishing between them, which is sufficient for our purposes. In our case, we simply consider the uniform distributions over S_w and B_ε.

More formally, we first observe that

$$\begin{aligned}&\mathbb{E}_{f\in S_w}\left[\Pr_r\left[V_w^f(n,\varepsilon;r)=1\right]\right] - \mathbb{E}_{f\in B_\varepsilon}\left[\Pr_r\left[V_w^f(n,\varepsilon;r)=1\right]\right]\\ &= \mathbb{E}_r\left[\Pr_{f\in S_w}\left[V_w^f(n,\varepsilon;r)=1\right] - \Pr_{f\in B_\varepsilon}\left[V_w^f(n,\varepsilon;r)=1\right]\right], \qquad (2)\end{aligned}$$

By Eq. (2), it suffices to bound the distinguishing probably for any deterministic verifier. We do this via the following lemma.

Lemma 3.3. *For any deterministic verifier W with query complexity at most $\frac{k}{10}$, we have that*

$$\Pr_{f\in S_w}\left[W^f(n,\varepsilon)=1\right] - \Pr_{f\in B_\varepsilon}\left[W^f(n,\varepsilon)=1\right] \leq 1 - \frac{1}{4C}\cdot 2^{-p}.$$

Proof. We first show that

$$\Pr_{f\in B_\varepsilon}\left[W^f(n,\varepsilon)=1\right] \geq \frac{1}{2C}\cdot 2^{-p}\cdot \Pr_{f\in S_w}\left[W^f(n,\varepsilon)=1\right]. \qquad (3)$$

We can view the verifier W as a decision tree of depth $q' = k/10$. Each leaf of the decision tree is associated with indices $i_1, i_2, \ldots, i_{q'} \in D_n$ and values $t_1, t_2, \ldots, t_{q'} \in R_n$ such that a function $f \in \mathcal{F}_n$ is accepted at that leaf if and only if $f(i_j) = t_j$ for all $j \in [q']$. We may assume without loss of generality that the sets of indices $i_1, \ldots, i_{q'}$ for all paths in the decision tree are pairwise distinct. Fix such a sequence of indices $i_1, \ldots, i_{q'} \in D_n$ and values $t_1, \ldots, t_{q'} \in R_n$. Then,

$$\begin{aligned}\Pr_{f\in B_\varepsilon}\left[f(i_j)=t_j \text{ for all } j\in[q']\right] &\geq \frac{|\{f\in\mathcal{F}_n : f(i_j)=t_j \text{ for all } j\in[q']\}|-|\mathcal{F}_n\backslash B_\varepsilon|}{|B_\varepsilon|}\\ &\geq \frac{1}{|R_n|^{q'}}-\frac{1}{|R_n|^k-1}\\ &\geq \frac{1}{2|R_n|^{q'}}. \end{aligned} \tag{4}$$

Here we have used k-sparsity to note that $\frac{|\mathcal{F}_n\backslash B_\varepsilon|}{|B_\varepsilon|}\leq\frac{1}{|R_n|^k-1}$, and we used that $q'=\frac{k}{10}$. On the other hand, we also have that:

$$\Pr_{f\in S_w}\left[f(i_j)=t_j \text{ for all } j\in[q']\right]\leq\frac{\Pr_{f\in\Pi_n}\left[f(i_j)=t_j \text{ for all } j\in[q']\right]}{\Pr_{f\in\Pi_n}[f\in S_w]}\leq\frac{C\cdot 2^p}{|R_n|^{q'}} \tag{5}$$

by relaxed q'-wise independence and the lower bound on the size of S_w.

Dividing Eq. (4) by Eq. (5), we obtain that

$$\Pr_{f\in B_\varepsilon}[f(i_j)=t_j \text{ for all } j\in[q']]\geq\frac{1}{2C}\cdot 2^{-p}\cdot\Pr_{f\in S_w}[f(i_j)=t_j \text{ for all } j\in[q']]\,.$$

Now, summing the above equation over all leaves of the decision tree corresponding to W (since these correspond to disjoint events) gives us Eq. (3).

Given Eq. (3), we now consider two cases. First, if

$$\Pr_{f\in S_w}\left[W^f(n,\varepsilon)=1\right]\leq 1-\frac{1}{2C}\cdot 2^{-p}$$

we are obviously done. Otherwise, we can assume that $\Pr_{f\in S_w}\left[W^f(n,\varepsilon)=1\right]>1-\frac{1}{2C}\cdot 2^{-p}$ and so:

$$\begin{aligned}\Pr_{f\in S_w}\left[W^f(n,\varepsilon)=1\right]-\Pr_{f\in B_\varepsilon}\left[W^f(n,\varepsilon)=1\right] &\leq 1-\frac{1}{2C}\cdot 2^{-p}\cdot\Pr_{f\in S_w}\left[W^f(n,\varepsilon)=1\right]\\ &\leq 1-\frac{1}{2C}\cdot 2^{-p}\cdot\left(1-\frac{1}{2C}\cdot 2^{-p}\right)\\ &\leq 1-\frac{1}{4C}\cdot 2^{-p},\end{aligned}$$

where the first inequality is by Eq. (3). The lemma follows.

□

Now we are ready to use Lemma 3.3 to complete our proof of Theorem 3. Because V_w^f has completeness and soundness errors $\frac{1}{10C}\cdot 2^{-p}$, we have that

$$\begin{aligned}\mathbb{E}_{f\in S_w}\left[\Pr_r\left[V_w^f(n,\varepsilon;r)=1\right]\right]-\mathbb{E}_{f\in B_\varepsilon}\left[\Pr_r\left[V_w^f(n,\varepsilon;r)=1\right]\right] &\geq 1-\frac{1}{10C}\cdot 2^{-p}-\frac{1}{10C}\cdot 2^{-p}\\ &=1-\frac{1}{5C}\cdot 2^{-p}.\end{aligned}$$

On the other hand, by Eq. (2) and Lemma 3.3, it holds that

$$\mathbb{E}_{f\in S_w}\left[\Pr_r\left[V_w^f(n,\varepsilon;r)=1\right]\right]-\mathbb{E}_{f\in B_\varepsilon}\left[\Pr_r\left[V_w^f(n,\varepsilon;r)=1\right]\right]\leq 1-\frac{1}{4C}\cdot 2^{-p},$$

which is a contradiction. Therefore, we can conclude that $p\cdot q\geq \frac{k}{10C'}$, as desired.

4 An Exponential Gap Between MAP and AMP

In this section we prove Theorem 1, by exhibiting a property with an exponential gap between its AMP and MAP complexities. In fact, we show this separation result with respect to the permutation property, which we define next:

Definition 4.1 (The Permutation Property)**.** *Let* $\mathsf{Perm}=\bigcup_n \mathsf{Perm}_n$ *be the property, where* Perm_n *consists of all functions* $f:[n]\to[n]$ *that are permutations.*

To prove Theorem 1, we prove the following upper and lower bounds for the permutation property.

Lemma 4.2 (AMP upper bound)**.** *There exists an* AMP*, with respect to proximity parameter* $\varepsilon>0$*, for* Perm *with proof complexity* $O(\log(n)/\varepsilon)$ *and query complexity* $O(1/\varepsilon)$*.*

Lemma 4.3 (MAP lower bound)**.** *Any* MAP*, with respect to proximity parameter* $\varepsilon\leq 1/10$*, for* Perm *with proof complexity* p *and query complexity* q *must satisfy* $p\cdot q=\Omega(\sqrt{n})$*.*

Note that $p\cdot q=\Omega(\sqrt{n})$ implies that $p+q=\Omega(n^{1/4})$, and so combining Lemmas 4.2 and 4.3 shows that Perm has an AMP with logarithmic complexity, whereas any MAP for Perm has complexity $\Omega(n^{1/4})$. This proves Theorem 1.

Section Organization. The rest of this section is devoted to the proofs of Lemmas 4.2 and 4.3. We first prove the AMP upper bound (Lemma 4.2) in Section 4.1 and then the MAP lower bound (Lemma 4.3) in Section 4.2.

4.1 AMP Upper Bound - Proof of Lemma 4.2

The AMP for checking whether $f:[n]\to[n]$ is a permutation proceeds as follows. First, the verifier randomly selects $O(1/\varepsilon)$ integers from $[n]$ (with repetition). Denote this (multi-)set of integers by S. The verifer sends S to the prover and expects to get in response a sequence of integers $(t_s)_{s\in S}$. After receiving these, the verifier simply checks whether $f(t_s)=s$ for every $s\in S$. If so, it accepts, and otherwise, it rejects.

The proof complexity of this protocol is $O(\log(n)/\varepsilon)$, as the prover uses $O(\log n)$ to specify each t_s. The query complexity is $O(1/\varepsilon)$.

Completeness. Suppose $f\in\mathsf{Perm}_n$, i.e. f is indeed a permutation. The prover can respond with $(t_s)_{s\in S}$, where $t_s=f^{-1}(s)$ for every $s\in S$. Given this response the verifier accepts with probability 1.

Soundness. Assume that $f : [n] \to [n]$ is ε-far from Perm_n. Note that the range of f has size at most $(1-\varepsilon)\cdot n$, i.e. $|\{f(i) : i \in [n]\}| \le (1-\varepsilon)\cdot n$.[10] The only way in which the prover can convince the verifier to accept is if for all $s \in S$ it holds that s is in the range of f. Therefore, the probability that the verifier accepts is at most $(1-\varepsilon)^{O(1/\varepsilon)}$ which is smaller than $1/3$ (by setting the constant in the big-Oh notation to be sufficiently large).

4.2 MAP Lower Bound - Proof of Lemma 4.3

We proceed to show that any MAP for Perm must have complexity $\Omega(n^{\frac{1}{4}})$. More accurately, for a fixed proximity parameter $\varepsilon = \frac{1}{10}$, we show that any MAP for Perm, with respect to proximity parameter ε, with proof complexity p and query complexity q must satisfy $p \cdot q = \Omega(\sqrt{n})$.

To show this, we simply show that Perm is relaxed $\frac{\sqrt{n}}{10}$-wise independent and $\sqrt{n}$-sparse (as defined in Section 3). The MAP lower bound then follows from Theorem 3. This is done in the two lemmas below. Recall that for our property Perm, we have that $R_n = [n]$.

Lemma 4.4. Perm *is relaxed* $\frac{\sqrt{n}}{10}$*-wise independent.*

Proof. Define $q := \frac{\sqrt{n}}{10}$. Fix pairwise distinct indices $i_1, i_2, \ldots, i_q \in [n]$ and values $t_1, t_2, \ldots, t_q \in [n]$. If the t_j's are not distinct then clearly it holds that:

$$\Pr_{f \in \mathsf{Perm}}\left[f(i_j) = t_j \text{ for all } j \in [q]\right] = 0.$$

If the t_j's are distinct, then:

$$\begin{aligned}\Pr_{f \in \mathsf{Perm}}\left[f(i_j) = t_j \text{ for all } j \in [q]\right] &= \frac{1}{\prod_{i=0}^{q-1}(n-i)} \\ &= \frac{1}{n^q} \cdot \prod_{i=0}^{q-1}\left(1 - \frac{i}{n}\right)^{-1} \\ &\le \frac{1}{n^q} \cdot \left(1 - \frac{q^2}{2n}\right)^{-1} \\ &\le \frac{2}{n^q},\end{aligned}$$

where the last inequality uses the fact that $q = \frac{\sqrt{n}}{10}$. □

We still need to show that Perm is sparse. Recall that $B_\varepsilon = \{f \in \mathcal{F}_n : f \text{ is } \varepsilon\text{-far from } \mathsf{Perm}\}$ (i.e., the no-instances).

[10] Otherwise, by changing the repeated outputs of f to different outputs, we can construct a permutation g that is ε-close to f.

Claim 4.5. *For any constant $\varepsilon \leq \frac{1}{10}$, we have that*

$$\Pr_{f \in \mathcal{F}_n}[f \in B_\varepsilon] \geq 1 - e^{-\frac{n}{10}}$$

.

Proof. Recall that $f : [n] \to [n]$ is ε-far from Perm_n if and only if the image of f has size at most $(1-\varepsilon) \cdot n$. Thus, we need to upper bound the probability that a random function f has an image of size greater than $(1-\varepsilon) \cdot n$.

Let $k = \frac{n}{2} + \varepsilon n$ and consider the set $S = \{f(i) : i \in [k]\}$. If f is to have image with size greater than $(1-\varepsilon) \cdot n$, then certainly we must have $|S| \geq \frac{n}{2}$. Now each of the values $f(k+1), \dots, f(n)$ has probability at least $\frac{|S|}{n} \geq \frac{1}{2}$ of colliding with a previous value. By the Chernoff bound, the probability that we get at most εn collisions among these last $n-k$ values is at most

$$\exp\left(-\frac{(1-2\varepsilon)^2 \cdot (n-k)}{2}\right) \leq e^{-\frac{n}{10}}$$

for any $\varepsilon \leq \frac{1}{10}$, as desired. □

Corollary 4.6. Perm *is $\sqrt{n}$-sparse.*

Proof. We clearly have that $e^{-\frac{n}{10}} \leq n^{-\sqrt{n}}$ for sufficiently large n. □

This completes the proof of Lemma 4.3.

5 MA vs AM Revisited: Oblivious Proofs of Proximity

We revisit the MA versus AM problem within the context of *oblivious proofs of proximity*. These are proofs of proximity that have the special feature that the queries that the verifier makes are independent of the proof (or prover messages in the interactive setting). Such oblivious proofs should be thought of as a two phase process. First, there is a query phase in which the verifier can makes its queries but is not allowed to interact with the prover. In the communication phase, the verifier interacts with the prover (or just receives the proof in the non-interactive setting) but is not allowed to make any more queries.

We thus define (equivalently to the definition in [RVW13, GR18]) an oblivious MAP as an MAP in which a verifier V operates in the following stages: (1) V queries the input f, (2) V receives a proof w, and (3) V decides whether to accept or reject according to w and the queries that it made in the first stage.

The definition of oblivious AMPs is analogous, except for one subtle point. Namely, it is crucial that the query phase is *decoupled* from the proof phase; that is, the verifier first queries the input f, then it engages in a public-coin interaction in which it sends a *fresh* random string r, receives a proof w that depends on f and r, and rules according to w and the queries it made. In particular, the randomness that was used for the first step is not revealed to the prover.[11] While we find this definition to be the most natural one, we do remark that it has the unfortunate consequence that an oblivious AMP according to our definition is not necessarily an AMP. The reason is that the verifier in an oblivious AMP is allowed to toss coins that are not revealed to the prover. Still, an oblivious AMP can be viewed as an an AMAP (i.e., an AMP that is allowed to toss coins after receiving the proof).

In the next subsection, we show that the gap between the power of *oblivious* AMPs and *oblivious* MAPs is much smaller than the one exhibited in Theorem 1. Subsequently, we use this result to derive an exponential separation between the power of public and private coin oblivious proofs of proximity.

5.1 A Generic Lower Bound

In this subsection we prove Theorem 2, which shows that oblivious AMPs can only be *quadratically* stronger than oblivious MAPs, and in fact, standard testers. We remark that this result generalizes the lower bound on oblivious MAPs in [GR18].

Fix $\varepsilon > 0$ and input length $n \in \mathbb{N}$, and let Π be any property of inputs of length n. Suppose that there exists an AMP verifier V, with respect to proximity parameter ε, for Π with proof complexity p and query complexity q. We show that this implies that there exists a tester (i.e., MAP with proof complexity 0), also with respect to proximity parameter ε, for Π with query complexity $O(p \cdot q)$. We begin with a high-level overview, followed by a complete proof.

Overview. Recall that an AMP protocol has the following structure. The tester makes (possibly adaptive) queries $a_1, \dots, a_q$ to the input function f, using randomness ρ_{queries}. Then, the verifier samples fresh randomness ρ_{msg} and sends it to the prover. In return, the prover replies with a proof w, which may arbitrarily depend on f, the proximity parameter ε, and the verifier message ρ_{msg}. Finally the verifier reads w and decides according to it and the queries $a_1, \dots, a_q$.

The high-level idea is that since the query phase is independent of the proof phase (i.e., the verifier's message ρ_{msg} and the prover's message w), a tester can emulate all possible proofs, while using the *same* samples for *all* invocations. To support a union bound over all possible proofs, we wish to perform standard parallel repetition. Indeed, this is the simple argument used in the lower bound on oblivious MAPs in [GR18].

However, the situation is more involved when dealing with oblivious AMPs. Specifically, we cannot perform standard parallel repetition, as this would increase the *proof* complexity.[12] In addition, it is not clear a priori how the verifier should identify an (emulated) valid proof,

[11] Indeed, note that if the verifier also sends the old randomness that was used to determine its queries, then the resulting proof system is rendered completely degenerate. Namely, the verifier has all the information necessary to fully emulate the optimal prover (by say, enumerating over all possible proofs), and thus this model is equivalent to standard property testing (without a proof or a prover).

[12] This causes a "circular" argument, since we want to reduce the soundness to be exponentially small in the proof complexity, but the amplification itself increases the proof complexity.

since it is possible that for some verifier message there exists a proof that always fools the verifier.

The way these difficulties are dealt with is by observing that since the query phase and the proof phase are decoupled, then each oblivious AMP induces a family of testers that are determined by the verifier and prover messages. In particular, this allows us to perform soundness amplification on the induced testers, rather than on the protocol, and hence this does *not* increase the proof complexity. This implies that, with high probability over verifier's message ρ_{msg}, each of the corresponding induced testers decides correctly, with only an exponentially small probability of error.

Thus, we can invoke all the testers that are induced by all proofs that correspond to a *particular* verifier message ρ_{msg}, while reusing the queries for all invocations. Finally, the completeness and soundness of the original AMP assert that with high probability we choose a good verifier message ρ_{msg}, and so we can rule according to the induced tester. A complete proof follows.

Proof of Theorem 2. We follow the notation used in the overview. Assume, without loss of generality, that the AMP verifier V has soundness error at most $1/100$ (this can be obtained via parallel repetition, while only increasing the proof and query complexity by a constant). For a verifier message ρ_{msg} and prover message w denote by $T_{\rho_{\mathsf{msg}},w}$ the tester that is induced by ρ_{msg} and $w = w(f, \rho_{\mathsf{msg}}, \varepsilon)$ (i.e., the decision procedure that the oblivious AMP verifier invokes after receiving the proof). Denote by $T_{\rho_{\mathsf{msg}},w}(a_1, \ldots, a_q; \varepsilon)$ the random variable that represents the output of $T_{\rho_{\mathsf{msg}},w}$ with respect to queries $a_1, \ldots, a_q$ drawn according to V's distribution of queries. Let $T'_{\rho_{\mathsf{msg}},w}$ be the induced tester that is obtained by amplifying the soundness of $T_{\rho_{\mathsf{msg}},w}$ via $O(p)$ repetitions; denote its query complexity by $q' = O(p \cdot q)$, to be determined precisely later. Note that all (amplified) induced testers perform queries according to the exact same distribution.

Consider the following tester T, which operates as follows.

1. *Query step:* Make the queries $a_1, \ldots, a_{q'}$ that the (amplified) induced testers perform.

2. *Emulation step:* Choose a uniform random string ρ_{msg}, and invoke the induced amplified testers $\{T'_{\rho_{\mathsf{msg}},w}\}_{w \in \{0,1\}^p}$ with respect to proximity parameter ε and queries $a_1, \ldots, a'_q$.

3. *Decision step:* Accept if and only if there exists $w \in \{0,1\}^p$ such that $T'_{\rho_{\mathsf{msg}},w}(a_1, \ldots, a_q; \varepsilon) = 1$.

Clearly the query complexity of T is as stated. We proceed to prove its correctness. For a function f (purportedly in the property Π), consider the quantity

$$\mathsf{hard}_{f,\Pi}(\rho_{\mathsf{msg}}) = \max_{w} \Pr_{a_1,\ldots,a_q}\left[T_{\rho_{\mathsf{msg}},w}(a_1, \ldots, a_q; \varepsilon) = 1\right] ,$$

which can be viewed as measuring the hardness of the random "challenge" ρ_{msg} that is posed by the verifier V; that is, the probability of the optimal prover strategy to convince the verifier given the random message ρ_{msg}.

For completeness, suppose $f \in \Pi$, and observe that $\mathbb{E}_{\rho_{\mathsf{msg}}}[\mathsf{hard}_{f,\Pi}(\rho_{\mathsf{msg}})] \geq 99/100$. Thus, by an averaging argument, we have that $\Pr_{\rho_{\mathsf{msg}}}[\mathsf{hard}_{f,\Pi}(\rho_{\mathsf{msg}}) \geq 9/10] \geq 9/10$, which correspond to the "good" event that the verifier chose a random challenge ρ_{msg} that admits a convincing prover strategy. In which case, there exists a proof $w \in \{0,1\}^p$ such that

$$\Pr[T^f(\varepsilon) = 1] \geq \Pr_{\rho_{\mathsf{msg}}}[\mathsf{hard}_{f,\Pi}(\rho_{\mathsf{msg}})] \cdot \Pr[T'_{\rho_{\mathsf{msg}},w}(a_1,\ldots,a_q;\varepsilon) = 1] \geq \frac{9}{10} \cdot \Pr[T_{\rho_{\mathsf{msg}},w}(a_1,\ldots,a_q;\varepsilon) = 1] \geq \frac{2}{3} \ ,$$

as required.

The soundness argument is similar, only that now we need to rely on the amplification to tolerate a union bound over all possible proofs. More precisely, suppose f is ε-far from Π, and observe that $\mathbb{E}_{\rho_{\mathsf{msg}}}[\mathsf{hard}_{f,\Pi}(\rho_{\mathsf{msg}})] < 1/100$. Thus, by an averaging argument, we have that $\Pr_{\rho_{\mathsf{msg}}}[\mathsf{hard}_{f,\Pi}(\rho_{\mathsf{msg}}) \geq 9/10] \leq 1/10$, which correspond to the "good" event that the verifier chose a random challenge ρ_{msg} that does *not* admit a convincing prover strategy. Thus,

$$\Pr[T^f(\varepsilon) = 1] \geq \Pr_{\rho_{\mathsf{msg}}}[\mathsf{hard}_{f,\Pi}(\rho_{\mathsf{msg}})] \cdot \Pr[\exists w \in \{0,1\}^p \text{ such that } T'_{\rho_{\mathsf{msg}},w}(a_1,\ldots,a_q;\varepsilon) = 1] \geq \frac{9}{10} \cdot 2^p \cdot 2^{O(-p)} \ .$$

We conclude the argument by choosing a sufficiently large $q' = O(pq)$ such that $\Pr[T^f(\varepsilon) = 1] \geq 2/3$. □

5.2 Application: a Chasm Between Public and Private Coin

As an application, we use Theorem 2 to derive lower bounds on public-coin oblivious interactive proofs of proximity, and show an exponential separation between public-coin and private-coin protocols in this setting.

More specifically, (private-coin) oblivious interactive proofs of proximity (oblivious IPPs) [RVW13] are a natural extension of oblivious MAPs, in which the proof is replaced with a prover with whom the verifier can interact. More accurately, an r-round oblivious IPP consists of a verifier V and prover P that interact as follows: (1) V queries the input f, (2) V and P exchange messages in r rounds, where each round contains one message from each of the parties, and (3) at the end of the interaction, V rules according to the transcript of the interaction and the queries it made.

A public-coin oblivious interactive proof of proximity is simply referred to as an r-round oblivious AMP. Here the definition is the same as with oblivious IPPs, except that now the verifier is only allowed to send a (fresh) random string in each round. Note that standard oblivious AMPs (to which we referred in the previous subsection) are simply 1-round oblivious AMPs. The complexity of an oblivious IPP is the sum of its communication complexity and query complexity.

We remark that in polynomial time computation, as well as in non-oblivious proofs of proximity, the aforementioned models are roughly equivalent (see, e.g., [RVW13]). In stark contrast, as the following theorem shows, it turns out that oblivious IPPs can be much more powerful than their public-coin counterparts.

Theorem 4. *There exists a property Π for which there exists a 2-round oblivious* IPP *whose complexity is exponentially smaller than that of any r-round oblivious* AMP*, for constant $r \in \mathbb{N}$.*

We prove Theorem 4 with respect to the following natural property that consists of all low-degree polynomials that have a root in a predetermined subset. More precisely, let $\mathbb{F}$ be a finite field, let $m, d \in \mathbb{N}$ such that $d \cdot m < |\mathbb{F}|/10$ and let H be an arbitrary subset of $\mathbb{F}$ of size $d+1$. Consider the following property.

Definition 5.1. *The* Tensor Root *property, denoted* $\mathsf{TensorRoot}_{\mathbb{F},m,d,H}$*, is parameterized by a field $\mathbb{F}$, a dimension $m \in \mathbb{N}$, a degree $d \in \mathbb{N}$ and a subset $H \subset \mathbb{F}$, and contains all polynomials $P : \mathbb{F}^m \to \mathbb{F}$ of individual degree d that takes a root in H^m; that is,*

$$\mathsf{TensorRoot}_{\mathbb{F},m,d,H} = \{P : \mathbb{F}^m \to \mathbb{F} \textit{ of individual degree } 2d \;\; : \exists z \in H^m \textit{ such that } P(z) = 0\} \; .$$

Throughout this section we fix a fairly standard parameterization of low degree extension from the PCP literature; namely, we fix a size parameter $n \in \mathbb{N}$, degree $d = \log(n) - 1$, dimension $m = \log(n)/\log\log(n)$, a finite field $\mathbb{F}$ of size $10 \cdot dm$, and a subset H of $\mathbb{F}$ of size d (note that $|H|^m = n$). Then, we denote $\mathsf{TensorRoot} = \mathsf{TensorRoot}_{\mathbb{F},m,d,H}$, with the respect to the parameters that we fixed.[13]

We prove Theorem 4 via the following two lemmas.

Lemma 5.2. *Fix $\varepsilon < 1/10$ and a constant $r \in \mathbb{N}$. Any oblivious r-round oblivious* AMP*, with respect to proximity parameter ε, for* $\mathsf{TensorRoot}$*, with communication complexity c and query complexity q must satisfy $c + q = n^{\Omega(1/r)}$.*

Lemma 5.3. *For any $\varepsilon > 0$, there exists a 2-round oblivious* IPP *for for* $\mathsf{TensorRoot}$*, with communication complexity $O(\log^2 n)$ and query complexity $O(\log(n))$.*

Before we proceed to prove the lemmas above, we shall first need the following immediate corollary of Theorem 2, which follows by a straightforward application of the Babai-Moran [BM88] speedup theorem, as worked out in the setting of proofs of proximity in [RVW13], while noting that this transformation preserves the oblivious feature.

Corollary 5.4. *For any property Π and constant $r \in \mathbb{N}$, if there exists an* oblivious *r-round* AMP *for Π with communication complexity c and query complexity q, then there also exists a tester for Π with query complexity $c^{O(r)} \cdot q$.*

In particular, Corollary 5.4 implies that for a property Π that requires t queries to test, it holds that any oblivious AMP for Π with communication complexity c and query complexity q must satisfy $c + q = t^{\Omega(1/r)}$.

[13] Note that the input is of size $|\mathbb{F}|^m$, which is larger than $|H|^m$.

Comparison with proofs of proximity for distributions. In a recent work [CG17], the question of private versus public coin interaction was studied in the context of proofs of proximity for *distribution testing*. These proof systems differ from standard proofs of proximity in three respects: (1) the input is a distribution, rather than a function; (2) query access is replaced with independent samples from the input distribution; and (3) proximity is measured by total variation distance, rather than Hamming distance.

Proofs of proximity for distribution testing can be thought of as oblivious in a very strong sense (because the verifier cannot query its input, but rather passively receives a set of samples). This allows for a very strong lower bound on r-round AM proofs of proximity for distribution testing. Namely, for *any* number of rounds r (including super-constant values) these proofs of proximity can be emulated by standard distribution testers (i.e., without proof or prover) at only a quadratic cost.

In contrast, our lower bound on r-round oblivious AMPs *degrades* as the round complexity increases. Hence, it is natural to inquire whether our lower bound can be improved to match the stronger lower bound that holds in the distribution testing setting.

However, we remark that the above strengthening is *impossible*, and in fact any lower bound for oblivious AMPs must degrade with the round complexity. This follows from the bounds on the *Tensor Sum* property that appear in [GR18]. More specifically, the argument in [GR18] shows that for every r, there exist an r-round AMP protocol for the *Tensor Sum* property with complexity $\tilde{O}(n^{1/r})$.[14] Moreover, their sumcheck-based protocol is actually an *oblivious* AMP. Since in [GR18] it is shown that the Tensor Sum property requires $\tilde{\Omega}(n)$ queries to test, then together with our Theorem 4, it holds that any r-round oblivious AMP must have complexity $n^{\Omega(1/r)}$. So the complexity of r-round oblivious AMPs for the Tensor Sum property decreases with as the number of rounds increases.

5.2.1 Proof of Lemma 5.2

By Corollary 5.4 (and the discussion that follows it), to prove Lemma 5.2 it suffices to prove that any (standard) property tester for $\mathsf{TensorRoot}$ must make $\tilde{\Omega}(n)$ queries. We prove this via the framework of Blais, Brody, and Matulef [BBM12] for showing property testing lower bounds via communication complexity lower bounds. To this end, we assume basic familiarity with communication complexity (for a comprehensive introduction to communication complexity, see [KN97]).

The basic approach of [BBM12] is to reduce a hard communication complexity problem to the property testing problem for which we want to show a lower bound. We follow [BBM12] by showing a reduction from the well-known communication complexity problem of *set-disjointness*. The aforementioned framework allows us to obtain a lower bound on the query complexity of testing the *Tensor Root* property.

Recall that the $\mathsf{set\text{-}disjointness}$ problem is the communication complexity problem wherein Alice gets an n-bit string x, Bob gets an n-bit string y, and their goal is to decide whether there exists $i \in [n]$ such that $x_i = y_i = 1$. Equivalently, Alice and Bob's inputs can be viewed

[14] Actually, in [GR18] this result is only stated for $r = 1, 2, 3, \log(n)$. However, straightforward inspection shows that it generalizes to any value of r.

as indicator vectors of sets $A, B \subseteq [n]$. In this case, the goal of the players is to decide if the sets corresponding to their inputs intersect or not. We denote this problem by DISJ_n.

It is well-known that the randomized communication complexity of the *set-disjointness* problem is linear in the size of the inputs, even under the promise that A and B intersect in at most one element.

Theorem 5 ([KS92]). *For every $n \in \mathbb{N}$, every randomized communication complexity protocol for DISJ_n must use $\Omega(n)$ bits of communication.*

We are now ready to prove the property testing lower bound on the $\mathsf{TensorRoot}$ property via reduction from the set-disjointness problem DISJ_n. Let T be a tester, with respect to proximity parameter $\varepsilon = 1/10$, for $\mathsf{TensorRoot}$ with query complexity q. We prove that $q = \tilde{\Omega}(n)$.

Consider the following randomized communication complexity protocol. There are two parties, Alice and Bob. Alice receives an input $x \in \{0,1\}^n$, and Bob receives an input $y \in \{0,1\}^n$. The goal is to decide whether $(x, y) \in \mathsf{DISJ}_n$, i.e., there exists an $i \in [n]$ such that $x_i = y_i = 1$; or whether $(x, y) \notin \mathsf{DISJ}$, i.e., for every $i \in [n]$ either $x_i = 0$ or $y_i = 0$.[15] Recall that we fixed $d = \log(n) + 1$, $m = \log(n)/\log\log(n)$, and $H \subseteq \mathbb{F}$ of size d (so $|H|^m = n$), and that the $\mathsf{low\text{-}degree\ extension}$ of a string $x \in \{0,1\}^n$ is the unique m-variate polynomial P_x of individual degree d that agrees with f_x on H^m, where $f_x \colon H^m \to \{0,1\}$ is an arbitrary predetermined embedding of x in H^m. The protocol takes places as follows:

1. *Creating distance:* Alice computes the (unique) low-degree extension of x, denoted P_x; Bob computes the (unique) low-degree extension of y, denoted P_y.

2. *Emulating the Tensor Root tester:* Alice invokes the tester T, which requires (possibly adaptive) queries to a function $P : \mathbb{F}^m \to \mathbb{F}$. For each query $z \in \mathbb{F}^m$, Alice evaluates $P_x(z)$, sends z to Bob, and Bob sends back $P_y(z)$ to Alice.

3. *The combining operator:* For each query $z \in \mathbb{F}^m$ asked by the tester and values $P_x(z)$ and $P_y(z)$ computed by Alice and Bob, Alice feeds the tester T with the evaluation of the m-variate individual degree $2d$ polynomial $P(z) \coloneqq 1 - P_x(z) \cdot P_x(z)$.

For completeness, note that if $(x, y) \in \mathsf{DISJ}_n$, then there exists an $i \in [n]$ such that $x_i = y_i = 1$. Hence, there exists a $t \in H^m$ such that $P_x(t) = P_y(t) = 1$, and so $1 - P_x(t) \cdot P_x(t) = 0$. Therefore, the (combined) polynomial P has a root in H^m, and so $P \in \mathsf{TensorRoot}$.

For soundness, note that if $(x, y) \in \mathsf{DISJ}_n$, then for every $i \in [n]$ either $x_i = 0$ or $y_i = 0$. Hence, for every $t \in H^m$ it holds that either $P_x(t) = 0$ or $P_y(t) = 0$, and so $1 - P_x(t) \cdot P_x(t) = 1$. Therefore, the (combined) polynomial P does *not* have a root in H^m. By the distance of the Reed-Muller code (i.e., the Schwartz-Zippel Lemma), this implies that P is ε-far from $\mathsf{TensorRoot}$.

[15] More accurately, this is actually the set-intersection problem, i.e., the complement of DISJ_n; however, since standard randomized communication complexity is closed under complement, the same lower bound applies.

Since Alice can obtain the answer to each query of the tester T to the function P using $O(\log n)$ bits of communication, then the reduction above implies a communication complexity protocol for DISJ_n with communication complexity $O(q \cdot \log(n))$, where recall that q denotes the query complexity of T. Plugging the $\Omega(n)$ lower bound on the (randomized) communication complexity of DISJ_n, we obtain that $q = \tilde{\Omega}(n)$, which concludes the proof of Lemma 5.2.

5.2.2 Proof of Lemma 5.3

We show a 2-round oblivious IPP for for TensorRoot, with communication complexity $O(\log^2 n)$ and query complexity $O(\log(n))$. The main idea underlying this protocol can be traced back to a work on quantum information and the PCP theorem by Raz [Raz09], and was also used in the study of streaming interactive proofs [CCM+15].

Note that the TensorRoot problem is trivial for non-oblivious proofs of proximity: the prover can simply indicate the location of the root in H^m, and then the verifier checks that this is the case. However, in oblivious proofs of proximity, the query is performed *prior* to the interaction with the prover, and so even if the prover tells the verifier the location of the root, the verifier cannot "go back in time" and make the query.

However, we show that by capitalizing on the algebraic structure of low-degree polynomials and on *private-coin* communication (which we know is required, by Lemma 5.2), we can in a sense "go back in time" and recover the value pointed by the prover, even though we can no longer make queries to the input function.

Consider the following 2-round oblivious IPP, with respect to proximity parameter $\varepsilon > 0$, for TensorRoot, where, assume without loss of generality that $\varepsilon < 1/10$.

1. **Query phase:**
   (a) *Low-degree test:* The verifier performs a standard low (individual) degree test on the input function $P : \mathbb{F}^m \to \mathbb{F}$, and accepts if and only if the low-degree test accepts. (See [GR18] for details of applying low-degree tests in the setting of proofs of proximity.)

   (b) *Secret query:* The verifier draws uniformly at random a point $r \in \mathbb{F}^m$ and queries $P(r)$.

2. **Interaction phase:**
   (a) *Pointing to the root:* The prover sends a purported root $z \in H^m$ (i.e., such that allegedly $P(z) = 0$).

   (b) *A line through the root:* The verifier sends (some canonical representation) of the line ℓ that passes through its secret query r and the purported root z. (Note that this does *not* reveal the full identity of r to the prover).

   (c) *Restricting the polynomial to the line:* the prover sends the purported restriction of P to the line ℓ, denoted $\tilde{P|}\ell$, specified by its $2d$ coefficients.

(d) *Line-versus-point consistency:* The verifier checks that $\widetilde{P|\ell}$, restricted to the point r, agrees with the value $P(r)$ that it had originally queried.

Since low-degree testing can be performed via $O(\log(n))$ queries, the protocol above clearly satisfies the query complexity requirement. In terms of communication, the verifier sends a line (which can be characterized by two canonically chosen points), and the prover sends $2d$ coefficients, thus the communication complexity is $O(\log^2 n)$.

Completeness is immediate by construction. For soundness, suppose that P is ε-far from TensorRoot. If P is ε-far from being an individual degree $2d$ polynomial, then the low-degree test rejects with high probability. Otherwise, by the distance of the Reed-Muller code (i.e., the Schwartz-Zippel Lemma) we can assume that P is ε-close to an individual degree $2d$ polynomial $\hat{P}$ such that $\hat{P}(x) \neq 0$ for all $x \in H^m$. Note that with probability $1 - \varepsilon$ (and by our assumption we have $1 - \varepsilon > 9/10$) the verifier queries a "good" point r such that $P(r) = \hat{P}(r)$. Suppose hereafter that this is the case.

The prover receives the line ℓ that passes through the secret query r and the purported root z. However, since this line is canonically represented, the prover does not know the location of the point r on the line ℓ. Since a low-degree polynomial restricted to a line is a Reed-Solomon codeword, then $\widetilde{P|\ell}$ must disagree with P except on a negligible number of points, and so the verifier rejects with probability at least $9/10 \cdot (1 - o(1)) \geq 2/3$.

6 Discussion and Open Problems

The complexity of the permutation property for testers, which do not use a proof, is $\tilde{\Theta}(\sqrt{n})$. In this work we showed a lower bound of $\tilde{\Omega}(n^{\frac{1}{4}})$ for MAPs for Perm. Thus, the MAP complexity of Perm is somewhere between $\tilde{\Omega}(n^{\frac{1}{4}})$ and $\tilde{O}(\sqrt{n})$ - resolving the exact complexity is an interesting open problem:

Problem 6.1. *Does every* MAP *for* Permutation *have complexity* $\tilde{\Omega}(\sqrt{n})$*?*

Second, our work shows that AMPs can be exponentially more efficient than MAPs. It is natural to ask whether the converse also holds - can MAPs be much more efficient than AMPs? A partial answer to this question is known. As mentioned in Footnote 3, every MAP with complexity c can be emulated by an AMP with complexity (roughly) c^2.

Thus, MAPs can be at most *quadratically* more efficient than AMPs. However, we do not know a property for which this gap is tight. In particular, the following problem is open:

Problem 6.2. *Does this exist a property* Π *that has an* MAP *with complexity* $O(\sqrt{n})$ *but every* AMP *for* Π *must have complexity* $\Omega(n)$*?*

Acknowledgments

We thank Oded Goldreich and Justin Thaler for very helpful discussions.

References

[Aar12] Scott Aaronson. Impossibility of succinct quantum proofs for collision-freeness. *Quantum Information & Computation*, 12(1-2):21–28, 2012.

[AN02] Dorit Aharonov and Tomer Naveh. Quantum np-a survey. *arXiv preprint quant-ph/0210077*, 2002.

[ARW17] Amir Abboud, Aviad Rubinstein, and R. Ryan Williams. Distributed PCP theorems for hardness of approximation in P. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 25–36, 2017.

[AW09] Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *TOCT*, 1(1):2:1–2:54, 2009.

[BBM12] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012.

[BFS86] László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 337–347, 1986.

[BGH$^+$06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust pcps of proximity, shorter pcps, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006.

[BM88] László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988.

[Bra11] Mark Braverman. Poly-logarithmic independence fools bounded-depth boolean circuits. *Commun. ACM*, 54(4):108–115, 2011.

[BRV17] Itay Berman, Ron D. Rothblum, and Vinod Vaikuntanathan. Zero-knowledge proofs of proximity. *IACR Cryptology ePrint Archive*, 2017:114, 2017.

[BY96] Mihir Bellare and Moti Yung. Certifying permutations: Noninteractive zero-knowledge based on any trapdoor permutation. *J. Cryptology*, 9(3):149–166, 1996.

[Can15] Clément L. Canonne. A survey on distribution testing: Your data is big. but is it blue? *Electronic Colloquium on Computational Complexity (ECCC)*, 22:63, 2015.

[CCM$^+$15] Amit Chakrabarti, Graham Cormode, Andrew McGregor, Justin Thaler, and Suresh Venkatasubramanian. Verifiable stream computation and arthur-merlin communication. In *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, pages 217–243, 2015.

[CCMT14] Amit Chakrabarti, Graham Cormode, Andrew McGregor, and Justin Thaler. Annotations in data streams. *ACM Trans. Algorithms*, 11(1):7:1–7:30, 2014.

[CG17] Alessandro Chiesa and Tom Gur. Proofs of proximity for distribution testing. *ECCC*, 24:155, 2017.

[CMT10] Graham Cormode, Michael Mitzenmacher, and Justin Thaler. Streaming graph computations with a helpful advisor. In *Algorithms - ESA 2010, 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part I*, pages 231–242, 2010.

[CTY11] Graham Cormode, Justin Thaler, and Ke Yi. Verifying computations with streaming interactive proofs. *PVLDB*, 5(1):25–36, 2011.

[DR06] Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. *SIAM J. Comput.*, 36(4):975–1024, 2006.

[EKR04] Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. Fast approximate probabilistically checkable proofs. *Inf. Comput.*, 189(2):135–159, 2004.

[FGL14] Eldar Fischer, Yonatan Goldhirsh, and Oded Lachish. Partial tests, universal tests and decomposability. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 483–500, 2014.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs under general assumptions. *SIAM Journal on Computing*, 1999. Preliminary version in *FOCS'90*.

[FLV15] Eldar Fischer, Oded Lachish, and Yadu Vasudev. Trading query complexity for sample-based testing and multi-testing scalability. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1163–1182, 2015.

[GG16a] Oded Goldreich and Tom Gur. Universal locally testable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:42, 2016.

[GG16b] Oded Goldreich and Tom Gur. Universal locally verifiable codes and 3-round interactive proofs of proximity for CSP. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:192, 2016.

[GGK15] Oded Goldreich, Tom Gur, and Ilan Komargodski. Strong locally testable codes with relaxed local decoders. In *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, pages 1–41, 2015.

[GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.

[GGR15] Oded Goldreich, Tom Gur, and Ron D. Rothblum. Proofs of proximity for context-free languages and read-once branching programs - (extended abstract). In *International Colloquium on Automata, Languages and Programming ICALP*, 2015.

[Gol17] Oded Goldreich. *Introduction to Property Testing.* Cambridge University Press, 2017.

[GPW15] Mika Göös, Toniann Pitassi, and Thomas Watson. Zero-information protocols and unambiguity in arthur-merlin communication. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 113–122, 2015.

[GR15] Tom Gur and Ran Raz. Arthur-merlin streaming complexity. *Inf. Comput.*, 243:145–165, 2015.

[GR16] Tom Gur and Ron D. Rothblum. Non-interactive proofs of proximity. *Computational Complexity*, June 2016.

[GR17] Tom Gur and Ron D. Rothblum. A hierarchy theorem for interactive proofs of proximity. In *Innovations in Theoretical Computer Science ITCS*, 2017.

[GR18] Tom Gur and Ron D. Rothblum. Non-interactive proofs of proximity. *Computational Complexity*, 27(1):99–207, 2018.

[GS10] Oded Goldreich and Or Sheffet. On the randomness complexity of property testing. *Computational Complexity*, 19(1):99–133, 2010.

[Gur17] Tom Gur. *On Locally Verifiable Proofs of Proximity.* PhD thesis, Weizmann Institute, 2017.

[Kla03] Hartmut Klauck. Rectangle size bounds and threshold covers in communication complexity. In *18th Annual IEEE Conference on Computational Complexity (Complexity 2003), 7-10 July 2003, Aarhus, Denmark*, pages 118–134, 2003.

[Kla11] Hartmut Klauck. On arthur merlin games in communication complexity. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, June 8-10, 2011*, pages 189–199, 2011.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity.* Cambridge University Press, 1997.

[KR15] Yael Tauman Kalai and Ron D. Rothblum. Arguments of proximity - [extended abstract]. In *CRYPTO*, 2015.

[KS92] Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM J. Discrete Math.*, 5(4):545–557, 1992.

[LFKN92] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[New91] Ilan Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, 39(2):67–71, 1991.

[Raz09] Ran Raz. Quantum information and the PCP theorem. *Algorithmica*, 55(3):462–489, 2009.

[Ron08] Dana Ron. Property testing: A learning theory perspective. *Foundations and Trends in Machine Learning*, 1(3):307–402, 2008.

[Ron09] Dana Ron. Algorithmic and analysis techniques in property testing. *Foundations and Trends in Theoretical Computer Science*, 5(2):73–205, 2009.

[RRR16] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 49–62, 2016.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

[RS04] Ran Raz and Amir Shpilka. On the power of quantum proofs. In *19th Annual IEEE Conference on Computational Complexity (CCC 2004), 21-24 June 2004, Amherst, MA, USA*, pages 260–274, 2004.

[RTVV98] Ran Raz, Gábor Tardos, Oleg Verbitsky, and Nikolai K. Vereshchagin. Arthur-merlin games in boolean decision trees. In *Proceedings of the 13th Annual IEEE Conference on Computational Complexity, Buffalo, New York, USA, June 15-18, 1998*, pages 58–67, 1998.

[RVW13] Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In *Symposium on Theory of Computing, STOC*, 2013.

[She16] Alexander A. Sherstov. The multiparty communication complexity of set disjointness. *SIAM J. Comput.*, 45(4):1450–1489, 2016.

[Tha16] Justin Thaler. Semi-streaming algorithms for annotated graph streams. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 59:1–59:14, 2016.

A Randomness Reduction in AMPs

In this section, we prove the following theorem, which shows that for reasonable properties, we can reduce the randomness of any AMP for the property down to $O(\log n)$. Specifically, we have the following theorem.

Theorem 6. *Let Π be a property of functions $f : D_n \to R_n$, where $|R_n|^{|D_n|} \leq \exp(\mathrm{poly}(n))$. If Π has an AMP verifier that makes q queries, uses a proof of length p, and uses r random bits, then Π has an AMP that makes $O(q)$ queries, uses a proof of length $O(p)$, and uses $O(\log n)$ random bits.*

We note that most properties that are considered have $|R_n| = O(\mathrm{poly}(n))$ and $|D_n| = O(\mathrm{poly}(n))$, so properties that have $|R_n|^{|D_n|} = \omega(\exp(\mathrm{poly}(n)))$ seem quite pathological.

Proof. Our proof follows closely to the proof of MAP randomness reduction in [GR18]. Let $\mathcal{F}_n$ denote the set of all functions $D_n \to R_n$. Let V be an AMP verifier for Π_n. For a function $f \in \mathcal{F}_n$, let $V^f(n, \varepsilon, w; t)$ denote the output of V when running with oracle access to f with random string t of length r. For every function $f \in \mathcal{F}_n$ and subset $S \subseteq \{0,1\}^r$, define

$$\beta_f(S) = \left| \Pr_{t \in \{0,1\}^r} \left[\exists w \text{ such that } V^f(n, \varepsilon, w; t) = 1 \right] - \Pr_{t \in S} \left[\exists w \text{ such that } V^f(n, \varepsilon, w; t) = 1 \right] \right| .$$

We show, using the probabilistic method, that there exists a multiset S of strings in $\{0,1\}^r$ of size $\mathrm{poly}(n)$ such that for all functions $f \in \mathcal{F}_n$ it holds that $\beta_f(S) \leq 1/7$.

Fix a function $f \in \mathcal{F}_n$. Sample k strings from $\{0,1\}^r$ uniformly at random and let these be the set S. By the Chernoff bound, we have that $\beta_f(S) \leq 1/7$ with probability $2^{-\Omega(k)}$ over our random choice of S. Thus by setting $k = O(\log |\mathcal{F}_n|) = O\left(\log(|R_n|^{|D_n|})\right) = O(\mathrm{poly}(n))$ and applying the union bound over all $f \in \mathcal{F}_n$, we obtain that there exists a multiset S as desired.

Now, we can obtain an AMP verifier using only $O(\log |S|) = O(\log n)$ random bits by simply running the original verifier V but with respect to random strings selected uniformly from S (instead of $\{0,1\}^r$). We can amplify the completeness and soundness to $\frac{2}{3}$ with $O(1)$ repetitions.

□

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il