On the Power of Computationally Sound Interactive Proofs of Proximity

Hadar Strauss ✉
Weizmann Institute of Science, Rehovot, Israel

Abstract

Interactive proofs of proximity (IPPs) are a relaxation of interactive proofs, analogous to property testing, in which soundness is required to hold only for inputs that are ϵ-*far* from the property being verified, where $\epsilon > 0$ is a proximity parameter. In such proof systems, the verifier has oracle access to the input, and it engages in two types of activities before making its decision: querying the input oracle and communicating with the prover. The main objective is to achieve protocols where both the query and communication complexities are extremely low.

In this work, we focus on *computationally sound* IPPs (cs-IPPs). We study their power in two aspects:

- *Query complexity:* We show that, assuming the existence of collision-resistant hashing functions (CRHFs), any public-coin cs-IPP that has query complexity q can be transformed into a cs-IPP that makes only $O(1/\epsilon)$ queries, while increasing the communication complexity by roughly q. If we further assume the existence of a good computational PIR (private information retrieval) scheme, then a similar transformation holds for *general* (i.e., possibly private-coin) cs-IPPs.
- *Coordination:* Aside from the low query complexity, the resulting cs-IPP has only minimal coordination between the verifier's two activities. The general definition of IPPs allows the verifier to fully coordinate its interaction with the prover and its queries to the input oracle. Goldreich, Rothblum, and Skverer (ITCS 2023) introduced two restricted models of IPPs that are *minimally coordinated*: The pre-coordinated model, where no information flows between the querying and interacting activities, but they may use a common source of randomness, and the isolated model, where the two activities are fully independent, each operating with a separate source of randomness.
  Our transformation shows that (under the aforementioned computational assumptions) any cs-IPP can be made to be in the pre-coordinated model, while preserving its efficiency. Hence, pre-coordinated cs-IPPs are essentially as powerful as general cs-IPPs.
  In contrast, we show that cs-IPPs in the *isolated* model are extremely limited, offering almost no advantage over property testers. Specifically, extending on a result shown by Goldreich et al. for *unconditionally sound* IPPs in the isolated model, we show that if a property has a cs-IPP in the isolated model that makes q queries and uses $c > 0$ bits of communication, then it has a tester with query complexity $O(c \cdot q)$.

2012 ACM Subject Classification Theory of computation → Interactive proof systems

Keywords and phrases Interactive Proofs of Proximity, Computational Soundness

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.117

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2025/195/

Acknowledgements I am deeply grateful to my advisor, Oded Goldreich, for his invaluable guidance throughout the development of this work, and, in particular, for his feedback during the writing stages. I also wish to thank Tom Gur, Ron Rothblum, and Guy Rothblum for helpful discussions, and anonymous reviewers for helpful comments.

1 Introduction

Interactive proofs of proximity (IPPs) are the property testing analog of interactive proofs. The focus of this work is on studying the power of *computationally sound* IPPs. We begin with some background.

Property testing. The field of property testing [8, 17] studies a relaxed notion of decision problems. Rather than deciding exact membership, a property tester is only required to distinguish (w.h.p.) between objects in the property and objects that are ϵ-*far* from the property, where $\epsilon > 0$ is a proximity parameter. An object $x \in \{0,1\}^n$ is considered ϵ-far from a property $\Pi_n \subseteq \{0,1\}^n$ if it differs from any object in the property on more than $\epsilon \cdot n$ locations.

Standard decision problems generally require reading the entire input, since flipping a single bit can change the decision. In contrast, relaxed decision opens the possibility of algorithms that (probabilistically) read only a sub-linear portion of the input. Thus, in property testing, the input is viewed as a huge object to which the tester gets only oracle access, and the goal is to obtain testers with extremely low query complexity (e.g., query complexity that is poly-logarithmic in the input size).

Interactive proofs of proximity. Interactive proofs of proximity (IPPs) [6, 16] extend the relaxation considered in property testing to the realm of proof systems, analogously to the extension of standard decision algorithms to standard interactive proofs (IPs). Specifically, an interactive proof of proximity for a property is a protocol between two parties, called a verifier and a prover. The verifier has oracle access to the input, and it interacts with an (untrusted) prover that tries to convince it to accept the input. The goal is for the verifier to be convinced to accept inputs that satisfy the property when interacting with an honest prover ("completeness"), and to not be fooled into accepting inputs that are far from the property, no matter what strategy is employed by the prover ("soundness"). The prover is assumed to have explicit access to the input and is computationally unbounded.

The main complexity measures considered in IPPs are the verifier's query complexity and the communication complexity (i.e., the total number of bits exchanged during the interaction with the prover). Like the query complexity, the communication complexity should be sublinear in the input length. With linear communication complexity, the prover could simply send the entire input, and the verifier could verify that (a) the alleged input is indeed in the property (which requires no queries to the oracle); and (b) the alleged input is ϵ-close to the actual input, by checking for consistency with $O(1/\epsilon)$ random locations in the actual input. Another complexity measure of interest is the round complexity (the number of back-and-forth communication rounds). The goal is to obtain proof systems with significantly lower query complexity than a tester for the property can achieve, while also minimizing the communication and round complexities.

The *computational* complexity of the verifier is also an important complexity measure, although it is often a secondary consideration in the property testing literature, where the focus is on query complexity. In this work, we require that the verifier is implementable in probabilistic polynomial time. IPPs where also the honest prover is implementable in probabilistic polynomial-time are called doubly-efficient (see, e.g., [15]).[1]

The isolated and pre-coordinated models. The verifier in an IPP performs two distinct activities: querying the input oracle and interacting with the prover. The general definition of IPPs allows the verifier to fully coordinate these two activities; that is, it can choose where to query the input based on its communication with the prover, and likewise, it can send challenges to the prover based on values seen in the input.

[1] Doubly-efficient IPPs should not be confused with doubly-sublinear IPPs [1], which refer to the *query* complexity of the honest prover.

Goldreich, Rothblum, and Skverer [10] considered highly restricted models of IPPs where the querying and interacting activities are assigned to separate modules such that no information can flow between them. The two modules feed their final views to a separate deciding module that decides whether to accept or reject based on the combined views.

They introduced two versions of this model. In the first model, called the isolated model, the querying and the interacting modules each get a separate and independent source of randomness, making them completely independent. The second model, called the pre-coordinated model, provides both modules with a shared source of randomness, which allows for some amount of coordination.[2]

Goldreich et al. showed that the isolated model is extremely weak; that is, it can only offer a very limited advantage over property testers. Specifically, they showed that IPPs in the isolated model that use q queries and $c > 0$ bits of communication can be emulated by property testers with query complexity $O(c \cdot q)$.

In contrast, they showed that the pre-coordinated model is much more powerful. In particular, they demonstrated that there are pre-coordinated IPPs of extremely low complexity for properties that are extremely hard to test. Still, they also showed that the pre-coordinated model is considerably limited compared to general IPPs. They showed that public-coin $O(1)$-round IPPs in the pre-coordinated model can be efficiently emulated by standard property testers.[3]

2 This work – computationally sound IPPs

In this work, we focus on *computationally sound* interactive proofs of proximity (cs-IPPs). That is, we consider IPPs in which the soundness condition is relaxed to hold only against computationally bounded provers.[4] Note that for the definition to be meaningful, cs-IPPs require that also the *honest* prover is computationally bounded; otherwise, any property may have a trivial cs-IPP in which the verifier simply checks whether the prover succeeds in solving a computationally hard task.[5] Hence, we actually consider computationally sound *doubly-efficient* IPPs.

2.1 Positive result in the pre-coordinated model

We exhibit the power of cs-IPPs, both for achieving low query complexity and for achieving pre-coordination. We show that, assuming the existence of collision-resistant hashing functions (CRHFs), any public-coin cs-IPP (and hence any public-coin *unconditionally* sound IPP with an efficient honest prover) can be efficiently emulated by a cs-IPP that makes only $O(1/\epsilon)$ queries, and is in the *pre-coordinated* model. The communication complexity of the resulting system is roughly $c + q$, where c and q are the communication and query complexity of the original system, respectively. Furthermore, the emulation adds only 2 rounds of interaction.

We consider two types of collision-resistance: weak collision-resistance, which requires that finding collisions is hard for polynomial-size circuits, and strong collision-resistance, which requires that finding collisions is hard for *sub-exponential* size circuits (see Definition 2.3 in the full version).

---

[2] The isolated and pre-coordinated models were originally studied in [10] as restricted versions of IPPs with proof-oblivious queries. Proof-oblivious queries restrict only the information flow from the interacting module to the querying module.

[3] We usually consider an emulation of one model in a second model to be efficient if the communication and query complexity in the second model are polynomial in the complexities in the first model.

[4] This notion was first discussed in passing in [16], and served as the focus of [13].

[5] Recall that we already assume that the *verifier* is computationally efficient.

▶ **Theorem 1** (public-coin cs-IPPs can be efficiently emulated in the pre-coordinated model)**.** *Let $\mathcal{H}$ be a family of* CRHFs*. Suppose that a property Π has a* public-coin *cs-*IPP *with query complexity q, communication complexity c, and round complexity r. Then Π has a cs-*IPP in the pre-coordinated model *with the following complexities:*

- Query complexity: $O(1/\epsilon)$. *Furthermore, the queries are uniformly and independently distributed.*[6]
- Communication complexity: $O\big(c(n) + \big(q(n) + \epsilon^{-1}\big) \cdot \tau(n) + \rho(n)\big)$*, where:*
  - $\tau(n) = \text{polylog}(n)$ *if $\mathcal{H}$ is strong collision-resistant, and $\tau(n) = O(n^{\gamma})$ for any constant $\gamma > 0$ if $\mathcal{H}$ is weak collision-resistant.*
  - ρ *is the randomness complexity of the original verifier.*
- Round complexity: $r(n) + 2$.

Furthermore, perfect completeness is preserved, and if $\mathcal{H}$ is a public-coin CRHF *family,*[7] *then the resulting cs-*IPP *is public-coin.*

The emulation also roughly preserves the *running time* of the verifier: If the running time of the original verifier is t_{V}, then the resulting verifier runs in time $O\big(t_{\mathsf{V}}(n) + \big(q(n) + \epsilon^{-1}\big) \cdot \tau(n)\big)$ (where $\tau(n)$ is as in the statement of Theorem 1). In addition, the running time of the resulting honest prover is $O(t_{\mathsf{P}}(n) + t_{\mathsf{V}}(n) + n \cdot \tau(n))$, where t_{P} is the running time of the original honest prover.

The term $\rho(n)$ appearing in Theorem 1 actually represents only the additional coins that the verifier uses to generate its queries, beyond the coins sent during the interaction (which contribute at most c coins). By standard techniques, this ρ term can always be reduced to $O(\log n)$, at the cost of introducing non-uniformity; see Remark 3.9 in the full version. Alternatively, we show that the ρ term can be avoided altogether, at the cost of $q(n)$ additional rounds of communication in the general case, and at no extra cost when the original verifier uses non-adaptive queries. However, the resulting system will not be public-coin (see Theorem 3.5 in the full version).

While Theorem 1 is stated for *public-coin* cs-IPPs, we show a similar transformation for *general* cs-IPPs when using a computational PIR (private information retrieval) scheme (see Theorem 3.10 in the full version).

Application to construction of zero-knowledge IPPs. IPPs in the pre-coordinated model are a special type of IPPs with *proof oblivious queries* (cf. [10]). As noted by [10], one of the motivations for obtaining IPPs with proof oblivious queries is their use in facilitating the construction of *zero-knowledge* IPPs [4]. Loosely speaking, in the context of IPPs, zero-knowledge means that anything that can be obtained from the prover can also be obtained without access to the prover in roughly the same time and with roughly the same number of queries.

In [4, Thm. 6.3] it was shown that assuming the existence of one-way functions, any public-coin cs-IPP with proof-oblivious queries can be efficiently transformed to a zero-knowledge cs-IPP. Hence, combining Theorem 1 with [4, Thm. 6.3], it follows that assuming the existence of a public-coin CRHF family, *any public-coin cs-*IPP *can be efficiently transformed to a zero-knowledge cs-*IPP. In addition, as stated in [10], it seems that also general (i.e., private-coin) cs-IPPs with proof oblivious queries can be transformed to zero-knowledge ones by using

[6] I.e., the resulting system is actually sample-based (cf. [9, 7]).

[7] $\mathcal{H}$ is a *public-coin* CRHF family if the collision resistance property still holds when the collision finder is given access to the random coins used by the sampler of $\mathcal{H}$.

secure two-party computation (or, alternatively, fully-homomorphic encryption). Combined with our general emulation in the pre-coordinated model (i.e., Theorem 3.10), it follows that (under the corresponding assumptions) *general* cs-IPPs can be converted into zero-knowledge cs-IPPs.

2.2 Negative result for the isolated model

Having shown the power of computational soundness for the pre-coordinated model, we turn to consider the *isolated* model. In contrast to the pre-coordinated model, we show that cs-IPPs in the isolated model can be efficiently emulated by testers, as in the unconditional soundness case. In particular, this demonstrates the necessity of pre-coordination for the result of Theorem 1.[8]

▶ **Theorem 2** (cs-IPPs in the isolated model can be efficiently emulated by testers)**.** *If a property* Π *has a cs-*IPP *in the isolated model with query complexity* q *and communication complexity* $c > 0$*, then* Π *has a tester with query complexity* $O(c \cdot q)$*.*

While Theorem 2 rules out an emulation in the isolated model with the efficiency of Theorem 1, in Section 3.3 of the full version we show that our pre-coordinated model emulation can be transformed to the isolated model at the cost of increasing the *product* of the query and communication complexities by roughly n/ϵ. More precisely, we can get a general tradeoff between the resulting query and communication complexities, such that for any $q' > O(1/\epsilon)$, we can obtain query complexity q' while increasing the communication complexity (over that obtained in Theorem 1) by an additive $O\left(\frac{n}{\epsilon \cdot q'}\right) \cdot \tau(n)$ term, where $\tau(n)$ is as in the statement of Theorem 1. Note that this tradeoff is nearly optimal given the negative result of Theorem 2, since there exist properties that have very efficient cs-IPPs but require nearly linear query complexity for testing (see details in Section 3.3 of the full version).

3 Technical Overview

3.1 Emulation of cs-IPPs by cs-IPPs in the pre-coordinated model

Our emulation in the pre-coordinated model is based on the well-known technique of Kilian [14]. Recall that Kilian showed how to transform PCPs into computationally sound interactive proofs that have very low communication complexity, assuming the existence of collision-resistant hashing functions. Kilian's protocol uses a special commitment scheme, called a tree commitment scheme (also known as Merkle-hashing), that allows one to commit to an n-bit string such that individual bits in the string can be revealed (and verified as correct) with very low communication cost (e.g., poly-logarithmic in n). In Kilian's protocol, the prover commits to the PCP-proof, and the verifier asks the prover to reveal the locations that the original PCP-verifier would query in the proof oracle. The commitment scheme ensures that a computationally bounded prover must respond consistently with a fixed proof string (i.e., it cannot cheat by answering differently based on the set of queries it has received).

[8] In [19] we showed this necessity through a direct lower bound on cs-IPPs in the isolated model for the property PERM (the set of all permutations over $[n]$). More precisely, we showed that any isolated cs-IPP for PERM with query complexity $q > 0$ and communication complexity $c > 0$ must satisfy $q^5 \cdot c = \Omega(n/\log(n))$. The emulation of isolated cs-IPPs by testers implies a similar lower bound for PERM that follows from a simpler argument. Specifically, since testing PERM requires $\Omega(\sqrt{n})$ queries (see [18, Apdx. A] following [11, Lem. 4.3]), it follows from Theorem 2 that any isolated cs-IPP for PERM with query complexity $q > 0$ and communication complexity $c > 0$ must satisfy $q \cdot c = \Omega(\sqrt{n})$.

Our emulation of public-coin cs-IPPs uses the tree commitment scheme to have the prover commit to the *input*. Recall that we are given a public-coin cs-IPP for some property Π, and aim to efficiently emulate it in the pre-coordinated model while making only $O(1/\epsilon)$ queries. The emulation will begin by executing the commitment phase of the tree commitment scheme, where the prover commits to a string that allegedly equals to the input. Next, we emulate the original interaction with the prover. Note that this can be done without access to the input oracle since the original interaction is public-coin. The emulation is done with respect to proximity parameter $\epsilon/2$ (where ϵ is our target proximity parameter). At the end of the emulated interaction, instead of making the queries that the original verifier would have made to the input oracle, we ask that the prover provide the answer to these queries by revealing the corresponding locations in the alleged input via the tree-commitment scheme. We check that the original verifier would have accepted given the answers the prover provided, along with the interaction transcript generated in the emulated interaction. In addition, we query the actual input oracle on $O(1/\epsilon)$ randomly chosen locations, ask the prover to reveal the same locations in the alleged (committed) input, and check for consistency. Note that the interacting and querying modules of the resulting system need only share the random consistency-check locations, and otherwise operate independently from one another.

To see why soundness holds, consider the foregoing emulation given an arbitrary input $x \in \{0,1\}^n$ that is ϵ-far from the property Π, when interacting with an arbitrary computationally bounded prover. At a high level, once the tree commitment is established, the prover's answers to the queries must be consistent with some fixed string x'. If x' is $(\epsilon/2)$-close to x, then x' is $(\epsilon/2)$-far from Π, and so the prover will fail to fool the original verifier with high probability. Otherwise, x' is $(\epsilon/2)$-far from x, and then by checking for consistency between x and x' on $O(1/\epsilon)$ random locations, we will detect a mismatch with high probability.

Note that it is important to first emulate the original interaction and only after ask the prover to provide the answer to the queries (which is possible due to the public-coin hypothesis). This is because, by asking the prover to provide the answers to the queries, we reveal to the prover their locations, whereas the soundness of the original verifier may rely on keeping these locations secret.

To extend the emulation to general cs-IPPs that are not necessarily public-coin, we interleave the emulation of the original interaction with the query requests, since the verifier's messages may depend on responses to its prior queries. To withhold the actual queries from the prover, we use a computational PIR scheme for providing answers to these queries.

While the technique of using the tree commitment scheme to force a computationally bounded prover to respond consistently with a fixed string is well-known, to the best of our knowledge, there is a lack of a source that presents this technique formally and in a general manner (with the exception of the concurrent [12]; see Section 4).[9] In this work, we provide a general lemma (see Lemma 3.3 in the full version) that captures this property of the tree commitment scheme. The lemma asserts that in any interaction that uses the tree commitment scheme, with overwhelmingly high probability, at the end of the commitment phase there exists a fixed string, such that any location that is subsequently opened will either be opened consistently with this string or will be detected as faulty.

[9] One notable source that provides a formal treatment of this technique is [2]. The setting treated in [2] is a more complex one, but it is less generic.

3.2 Emulation of cs-IPPs in the isolated model by testers

Recall that in the isolated model, the verifier's querying and interacting activities are totally independent from one another, each operating with a separate source of randomness. Specifically, the verifier is decomposed into a querying, interacting, and deciding modules Q, I, and D, respectively, such that its decision when interacting with a prover strategy P on input x is expressed as: $D(Q^x(R_Q), \langle P, I(R_I)\rangle)$, where R_Q and R_I are independent random variables representing the randomness of the querying and the interacting modules, respectively.

In [10] it was shown that *unconditionally sound* IPPs in the isolated model can be efficiently emulated by property testers: If a property Π has an IPP in the isolated model that makes q queries and uses $c > 0$ bits of communication, then Π has a tester with query complexity $O(c \cdot q)$. We show the same is true for *computationally sound* IPPs.

Roughly speaking, the reason that the emulation of [10] does not carry over to the context of cs-IPPs, is that it emulates the verifier with the *optimal* prover strategy. In the context of cs-IPPs, the performance of the optimal strategy (on NO-instances) is not relevant, since the optimal strategy may not be efficiently implementable.

Instead of the optimal prover, we will consider the *honest prover*, which is guaranteed to be efficient. Note that we cannot directly emulate the interaction with the honest prover on the actual input, because doing so requires full access to the input. Thus, instead, we emulate 2^n interactions, each with a different possible n-bit input (actually, it will suffice to use only the YES-instances). Observe that if the actual input is a YES-instance, then there exists an emulation that will lead to accepting with probability at least $2/3$. On the other hand, if the actual input is a NO-instance, then each of these emulations leads to rejection with probability at least $2/3$.

However, we do not want to take a union bound over 2^n events. Instead, we observe that each of these emulations can be expressed as a (different) convex combination of the probabilities that the verifier accepts given each fixed interaction transcript $\tau \in \{0,1\}^c$. For simplicity, assume that the interaction transcript fully determines the randomness of the interacting module. Then, for any prover strategy P, we have:

$$\begin{aligned}&\Pr\left[D\big(Q^x(R_Q), \langle P, I(R_I)\rangle\big) = 1\right] \\ &= \sum_{\tau\in\{0,1\}^c} \Pr\left[\langle P, I(R_I)\rangle = \tau\right] \cdot \Pr\left[D\big(Q^x(R_Q), \tau\big) = 1\right] \end{aligned} \tag{1}$$

Thus, we can obtain an (additive constant) approximation of the acceptance probability of all 2^n emulations by obtaining an approximation of the 2^c fixed-transcripts probabilities:

$$p^x_\tau \stackrel{\text{def}}{=} \Pr\left[D\big(Q^x(R_Q), \tau\big) = 1\right].$$

As observed in [10],[10] an approximation of p^x_τ can be obtained by repeated invocations of the querying module Q^x, where the same invocations can be used towards approximating all 2^c values p^x_τ since the invocations are oblivious of the value of τ. Hence, by invoking the querying module $O(c)$ times, we can obtain for each p^x_τ a constant additive approximation with error probability $O(2^{-c})$, and by a union bound we get an approximation of all 2^c values, with high constant probability. Using these values, we can obtain approximations of

[10] The fixed-transcript probabilities p^x_τ are used in [10] to compute the acceptance probability of the optimal prover via the max-average game tree of interactive proofs (where the probabilities p^x_τ are viewed as the leaves of the game tree).

the success probability of the honest prover strategy on all 2^n inputs, by computing, for each such strategy, the corresponding coefficients (i.e., $\Pr[\langle P, I(R_I)\rangle = \tau]$) in Eq. (1). Indeed, this will be computationally expensive, but the computational complexity is irrelevant to us.

4 Further related works

As mentioned in Section 2, computationally sound IPPs were briefly discussed in the work of Rothblum, Vadhan, and Wigderson [16], and then became the focus of the work of Kalai and Rothblum [13]. In the former work [16], the authors noted that in a similar manner to Kilian's technique, PCPs of *Proximity* (PCPPs) [3, 5] can be transformed to computationally sound IPPs. Note that this refers to committing to the *proof* oracle of the PCPP, as in Kilian's original protocol, whereas we use the commitment scheme to commit to the *input* oracle (and use it for a distinct setting and purpose). In addition, the work of [16] establishes a separation between what can be achieved with unconditionally sound versus computationally sound IPPs. The subsequent work of [13] strengthened this separation, under a stronger computational assumption. In addition, [13] showed that assuming the existence of a sub-exponentially secure FHE scheme, any language in P has a *one-round* cs-IPP with sublinear complexities (more specifically, the query complexity is $n^{1-\gamma}$ for some $\gamma > 0$).

More recently, independently and in parallel to us,[11] Herman and Rothblum [12] used an approach similar to ours in the context of *distribution testing*. Specifically, they use a tree commitment scheme to have the prover commit to an approximation of the entire input distribution. They use this to obtain a cs-IPP in which the verifier's complexities are all roughly $\sqrt{n}/\epsilon^2$ for any property of distributions decidable in polynomial time.

References

1 Noga Amir, Oded Goldreich, and Guy N. Rothblum. Doubly Sub-Linear Interactive Proofs of Proximity. In *16th Innovations in Theoretical Computer Science Conference*, volume 325 of *LIPIcs*, pages 6:1–6:25, 2025. doi:10.4230/LIPIcs.ITCS.2025.6.

2 Boaz Barak and Oded Goldreich. Universal arguments and their applications. In *Proceedings of the 17th IEEE Annual Conference on Computational Complexity*, CCC '02, page 194. IEEE Computer Society, 2002.

3 Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust pcps of proximity, shorter pcps and applications to coding. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 1–10. Association for Computing Machinery, 2004. doi:10.1145/1007352.1007361.

4 Itay Berman, Ron D. Rothblum, and Vinod Vaikuntanathan. Zero-Knowledge Proofs of Proximity. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:20, 2018. doi:10.4230/LIPIcs.ITCS.2018.19.

5 Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the pcp theorem. *SIAM J. Comput.*, 36(4):975–1024, 2006. doi:10.1137/S0097539705446962.

6 Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. Fast approximate probabilistically checkable proofs. *Information and Computation*, 189(2):135–159, 2004. doi:10.1016/J.IC.2003.09.005.

7 Guy Goldberg and Guy N. Rothblum. Sample-Based Proofs of Proximity. In *13th Innovations in Theoretical Computer Science Conference*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 77:1–77:19, 2022. doi:10.4230/LIPIcs.ITCS.2022.77.

[11] See preliminary version of this work [18].

8 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. doi:10.1145/285055.285060.

9 Oded Goldreich and Dana Ron. On sample-based testers. *ACM Trans. Comput. Theory*, 8(2), 2016. doi:10.1145/2898355.

10 Oded Goldreich, Guy N. Rothblum, and Tal Skverer. On Interactive Proofs of Proximity with Proof-Oblivious Queries. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251, pages 59:1–59:16, 2023. doi:10.4230/LIPIcs.ITCS.2023.59.

11 Tom Gur, Yang P. Liu, and Ron D. Rothblum. An Exponential Separation Between MA and AM Proofs of Proximity. *Comp. Complexity*, 30(2):12, 2021. doi:10.1007/S00037-021-00212-3.

12 Tal Herman and Guy N. Rothblum. How to verify any (reasonable) distribution property: Computationally sound argument systems for distributions. In *The Thirteenth International Conference on Learning Representations*, 2025.

13 Yael Tauman Kalai and Ron D. Rothblum. Arguments of proximity - [extended abstract]. In *CRYPTO*, pages 422–442. Springer, 2015. doi:10.1007/978-3-662-48000-7_21.

14 Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '92, pages 723–732, 1992. doi:10.1145/129712.129782.

15 Guy N. Rothblum and Ron D. Rothblum. Batch verification and proofs of proximity with polylog overhead. In *Theory of Cryptography: 18th International Conference, TCC, Proceedings, Part II*, pages 108–138. Springer-Verlag, 2020. doi:10.1007/978-3-030-64378-2_5.

16 Guy N. Rothblum, Salil Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 793–802. Association for Computing Machinery, 2013. doi:10.1145/2488608.2488709.

17 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996. doi:10.1137/S0097539793255151.

18 Hadar Strauss. Emulating computationally sound public-coin ipps in the pre-coordinated model. *Electronic Colloquium on Computational Complexity (ECCC)*, TR24-131, 2024. URL: https://eccc.weizmann.ac.il/report/2024/131.

19 Hadar Strauss. On the limits of computationally sound ipps in the isolated model. *Electronic Colloquium on Computational Complexity (ECCC)*, TR25-097, 2025. URL: https://eccc.weizmann.ac.il/report/2025/097.