

# On Interactive Proofs of Proximity with Proof-Oblivious Queries

Oded Goldreich

Weizmann Institute of Science, Rehovot, Israel

Guy N. Rothblum<sup>1</sup>

Apple, Cupertino, CA, USA

Tal Skverer

Weizmann Institute of Science, Rehovot, Israel

---

## Abstract

---

Interactive proofs of proximity (IPPs) offer ultra-fast approximate verification of assertions regarding their input, where ultra-fast means that only a small portion of the input is read and approximate verification is analogous to the notion of approximate decision that underlies property testing. Specifically, in an IPP, the prover can make the verifier accept each input in the property, but cannot fool the verifier into accepting an input that is far from the property (except for with small probability).

The verifier in an IPP system engages in two very different types of activities: interacting with an untrusted prover, and querying its input. The definition allows for arbitrary coordination between these two activities, but keeping them separate is both conceptually interesting and necessary for important applications such as addressing temporal considerations (i.e., at what time is each of the services available) and facilitating the construction of zero-knowledge schemes. In this work we embark on a systematic study of IPPs with proof-oblivious queries, where the queries should not be affected by the interaction with the prover. We assign the query and interaction activities to separate modules, and consider different limitations on their coordination.

The most strict limitation requires these activities to be totally isolated from one another; they just feed their views to a separate deciding module. We show that such systems can be efficiently emulated by standard testers.

Going to the other extreme, we only disallow information to flow from the interacting module to the querying module, but allow free information flow in the other direction. We show that extremely efficient one-round (i.e., two-message) systems of such type can be used to verify properties that are extremely hard to test (without the help of a prover). That is, the complexity of verifying can be polylogarithmic in the complexity of testing. This stands in contrast the MAPs (viewed as 1/2-round systems) in which proof-oblivious queries are as limited as our isolated model.

Our focus is on an intermediate model that allows shared randomness between the querying and interacting modules but no information flow between them. In this case we show that 1-round systems are efficiently emulated by standard testers but 3/2-round systems of extremely low complexity exist for properties that are extremely hard to test. One additional result about this model is that it can efficiently emulate any IPP for any property of low-degree polynomials.

**2012 ACM Subject Classification** Theory of computation → Complexity classes; Theory of computation → Interactive proof systems

**Keywords and phrases** Complexity Theory, Property Testing, Interactive Proofs, Interactive Proofs of Proximity, Proof-Oblivious Queries

**Digital Object Identifier** 10.4230/LIPIcs.ITCS.2023.59

**Related Version** *Full Version*: ECCC TR22-124 [15]

---

<sup>1</sup> Part of this work was done while the author was at the Weizmann Institute of Science



**Funding** *Oded Goldreich*: Partially supported by the Israel Science Foundation (grant No. 1041/18) and by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 819702).

*Guy N. Rothblum*: Received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 819702), from the Israel Science Foundation (grant number 5219/17), and from the Simons Foundation Collaboration on the Theory of Algorithmic Fairness.

*Tal Skverer*: Partially supported by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness.

**Acknowledgements** We are grateful to Ron Rothblum for helpful discussions.

## 1 Introduction

This paper initiates a systematic study of a natural type of interactive proofs of proximity, a notion which is a hybrid of interactive proofs and property testing. Specifically, interactive proofs of proximity (IPPs) combine the paradigm of verifying delegated computation with the paradigm of ultra-fast computation that refers to an approximate version of the actual input. Since any proof system is defined in terms of its verification procedure, which in turn presumes a model of computation, we start from the model of computation underlying property testing (see, e.g., the textbook [14]).

Loosely speaking, property testing typically refers to sub-linear time probabilistic algorithms for deciding whether a given object has a predetermined property or is far from any object having this property. Such algorithms, called testers, obtain local views of the object by making adequate queries; that is, the object is seen as a function and the testers get oracle access to this function (and thus may be expected to work in time that is sub-linear in the size of the object). In particular, one often seeks testers of extremely low query complexity (e.g., query complexity that is polylogarithmic in the size of the object).

Needless to say, such low level of complexity for testing cannot be achieved for all problems of interest, which leads to the natural question of whether traditional (non-interactive) proofs or interactive proofs can assist us in such cases. That is, viewing low query complexity as our base line, we are considering non-interactive and interactive proof systems in which the verifier makes few queries and reads short proofs (resp., short messages from the prover). Indeed, we are talking about “NP” and “IP” analogs of property testing (viewed as an analogue of “P”). Such analogs, termed *MA-proofs of proximity* (MAPs)<sup>2</sup> and *interactive proofs of proximity* (IPPs), were introduced and studied in [19] and [21], respectively.<sup>3</sup>

The seemingly innocuous analogy between the well-known complexity classes  $\mathcal{MA}$  and  $\mathcal{IP}$  and the “property testing versions” termed MAP and IPP ignores the question of orchestrating the two activities of the verifier (i.e., querying the oracle and interacting with the prover). The trivial answer is that these two activities are performed by the same entity concurrently with free information flow from one activity to the other. But *is this concurrency and free information flow essential?*

The foregoing question is not merely interesting *per se*. There may be settings in which such concurrency is not possible or that a free information flow is not desirable. For example, it may be that query access to the input is only available at one time, whereas the interaction

<sup>2</sup> Note that in a randomized setting, as is inherent for algorithms that read small parts of their input, alleged (non-interactive) proofs are verified probabilistically. Hence, such verification is actually analogous to MA rather than to NP.

<sup>3</sup> Actually, the work of [21] predated [19], and both were predated by [9], which presents an extremely general framework that contains both IPPs and PCPPs as a special case.

with the prover is only available afterwards. Alternatively, as outlined at the end of Section 2, having queries that are oblivious of the prover’s messages seems a first step towards obtaining a natural notion of zero-knowledge IPPs (cf. [4]). Indeed, our focus is on IPPs that employ *proof-oblivious queries*.

We mention that proof-oblivious queries were considered before, both in the context of MAPs (cf. [19, Def. 2.2]) and in the context of IPPs (cf. [21, Fn. 2] and [18, Sec. 5]). In particular, it was shown that MAPs with proof-oblivious queries can be efficiently emulated by a standard tester [19, Thm. 4.2]: If the former use proofs of length  $p$  and  $q$  queries, then the tester has query complexity  $O(p \cdot q)$ . This should be contrasted with the fact that there are MAPs (with proof-dependent queries) of extremely low complexity for properties that are extremely hard to test (see [19, Thm. 1.1]). We interpret these facts as saying that proof-oblivious queries severely limit the power of MAPs. But *is it so also for IPPs?*

Before answering the foregoing question, we observe that, in the context of IPPs, *the notion of proof-oblivious queries may have several different natural interpretations* (which all coincide in the context of MAPs).<sup>4</sup>

In fact, we initiate a systematic study of proof-oblivious queries (*PO-queries*) in the context of IPPs. Jumping ahead, we telegraphically highlight the following results (which will be properly reviewed in Section 3):

- Under the most strict interpretation of IPPs with proof-oblivious queries (i.e., the “isolated” model), these systems can be efficiently emulated by a standard tester.
- Under the most liberal interpretation of IPPs with proof-oblivious queries (i.e., the “general” model), there exist such systems of extremely low complexity for properties that are extremely hard to test. This holds even for 1-round systems (in which a single message by the verifier followed by a single message by the prover), in contrast to MAPs (which are 1/2-round IPPs).
- Under an intermediate (and natural) interpretation of IPPs with proof-oblivious queries (i.e., the “pre-coordinated” model), it holds that 1-round systems are efficiently emulated by testers but 3/2-round systems of extremely low complexity exist for properties that are extremely hard to test.

We now turn to a more detailed account of the underlying definitions and results.

## 2 Defining IPPs and Types of IPPs With PO-Queries

An interactive proof of proximity is a two-party protocol for parties called *verifier* and *prover*. The verifier has oracle access to a function  $f : [n] \rightarrow \Sigma$ , and also gets explicit inputs  $n$  and  $\epsilon > 0$ , where  $\epsilon$  is called the *proximity parameter*. The prover gets  $f$  as explicit input, and its aim is to convince the verifier that  $f$  is in some predetermined set  $\Pi_n$ , called the property. In analogy to the definition of interactive proof systems [16], we require that the prover can convince the verifier to accept any  $f$  in  $\Pi$  (w.h.p.), but cannot fool the verifier into accepting  $f$  that is  $\epsilon$ -far from  $\Pi_n$  (except for with low probability). Indeed, the main deviation from the definition of a (standard) interactive proof system is that *the soundness requirement is made only for inputs that are  $\epsilon$ -far from  $\Pi_n$* , where  $f$  is  $\epsilon$ -far from  $\Pi_n$  if for every  $g \in \Pi_n$  it holds that  $|\{i \in [n] : f(i) \neq g(i)\}| > \epsilon \cdot n$ . The actual definition refers to an optimal prover strategy, denoted  $P$ , and focuses on the communication and query complexities of the system.

<sup>4</sup> While [18] do not formally define proof-oblivious systems, their claims seem to refer to two different notions. See Section 6.

► **Definition 1.** (interactive proofs of proximity systems (IPPs) [21]):<sup>5</sup> Let  $\Pi = \bigcup_{n \in \mathbb{N}} \Pi_n$  such that  $\Pi_n$  is a set of functions over  $[n]$ . A randomized and interactive oracle machine, denoted  $V$ , constitutes a verifier for an interactive proof of proximity for (the property)  $\Pi$  if the following two conditions hold.

**(completeness):** On input  $n, \epsilon$  and oracle access to any  $f \in \Pi_n$ , after interacting with an optimal prover  $P$ , the verifier accepts with probability at least  $2/3$ .

**(soundness):** On input  $n, \epsilon$  and oracle access to any  $f : [n] \rightarrow \Sigma$  that is  $\epsilon$ -far from  $\Pi_n$ , after interacting with an optimal prover  $P$ , the verifier accepts with probability at most  $1/3$ .

We say that the system has **perfect completeness** if the verifier accepts each  $f \in \Pi$  with probability 1. The **query complexity** of  $V$  is  $q : \mathbb{N} \times [0, 1] \rightarrow \mathbb{N}$  if, on input  $n, \epsilon$  and oracle access to any  $f : [n] \rightarrow \Sigma$ , the verifier makes at most  $q(n, \epsilon)$  queries to  $f$ . The **communication complexity of the system** is  $c : \mathbb{N} \times [0, 1] \rightarrow \mathbb{N}$  if, on input  $n, \epsilon$  and oracle access to any  $f : [n] \rightarrow \Sigma$ , the parties exchange at most  $c(n, \epsilon)$  bits.

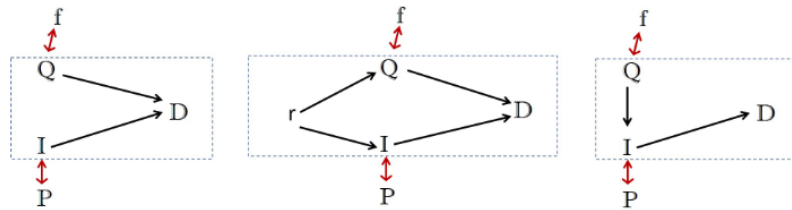
Standard testers can be viewed as a special case in which the communication complexity is zero; that is, effectively there is no prover. MAPs correspond to the special case in which the communication is unidirectional, with the prover sending a single message (of length at most  $c$ ).

The general notion of proof-oblivious queries merely asserts that the queries made by the verifier are oblivious of the messages received from the prover. In order to clarify what this means as well as discuss alternative notions, we decompose the verifier  $V$  into three (randomized) modules: A **querying module** denoted  $Q$ , which is in charge of querying the oracle, a **interacting module** denoted  $I$ , which is in charge of interacting with the prover, and a **deciding module** denoted  $D$ , which takes the final decision based on the information handed to it by the other two modules. Now, the **general notion (or model) of proof-oblivious queries** postulates that the interaction from the querying module  $Q$  to the other modules is unidirectional (from  $Q$  to  $I$  and  $D$ ). Actually, it suffices to have  $Q$  send a single message to  $I$ , which in turn sends a single message to  $D$ . (Actually, in this case, we may combine  $I$  and  $D$  into a single entity.)

In contrast, the most strict interpretation of proof-oblivious queries means that the querying and interacting modules are totally uncoordinated; in particular, they do not communicate with one another nor do they receive any message from the deciding module (since otherwise  $D$  may serve as a subliminal communication channel between  $Q$  and  $I$ ). In the corresponding model, called the **isolated model**, the two main modules (i.e.,  $Q$  and  $I$ ) are totally isolated from one another. The only communication between the three modules amount to  $Q$  and  $I$  sending their view to the deciding module  $D$ ; specifically, the querying module sends the answers that it has obtained from the oracle along with its internal coin tosses, whereas the interacting module sends the messages that it has obtained from the prover along with its internal coin tosses. Without loss of generality, each module sends a single message to  $D$ . See illustration on the l.h.s of Figure 1.

Note that in the isolated model the two messages received by  $D$  (from  $Q$  and  $I$ , respectively) are statistically independent. In particular,  $Q$  and  $I$  each have their own source of randomness, and these sources are independent of one another. In contrast, in the **pre-coordinated model**, these two parties are fed by the same source of randomness, although they are free to use disjoint parts of it. See illustration at the center of Figure 1. Hence, although the

<sup>5</sup> As discussed in Section 5, we avoid the good convention of specifying a (“honest”) prover strategy for the completeness condition.



■ **Figure 1** The isolated, pre-coordinated, and general models of PO-queries.

communication pattern (in the pre-coordinated model) is the same as in the isolated model, the messages that  $D$  receives may be correlated, since  $Q$  and  $I$  are using the same source of randomness. In particular,  $I$  may send to the prover messages that are related to the queries that  $Q$  made to the oracle, but unlike in the general model (see illustration on the r.h.s of Figure 1) these messages may not depend on the answers provided by the oracle.

In Section 7 we provide a more formal description of the three models (of IPPs with PO-queries) that are studied in the paper.

### Public-coin IPPs

Recall that public-coin interactive proofs (of proximity) are ones in which each message sent by the verifier is a predetermined subsequence of the randomness of the verifier. Note that any public-coin IPP that uses PO-queries is, by definition, in the pre-coordinated model, because its interacting module is not allowed to use the information it might have received from the querying module in determining its messages to the prover. On the other hand, any IPP of the isolated model can be emulated by a public-coin IPP of the isolated model.

### Obtaining zero-knowledge IPPs

Loosely speaking, IPPs that use proof-oblivious queries, even in the general sense, may be easily converted into zero-knowledge IPPs (e.g., as defined in [4]). In case these IPPs (with PO-queries) are of the public-coin type, we just have the (zero-knowledge) prover send commitments to its original messages, and prove at the very end (in zero-knowledge manner) that the original verifier would have accepted the corresponding (de-committed) information. This does not disrupt the new verifier's actions, since its queries are proof-oblivious and its messages to the prover are random. When using general (i.e., private-coin) IPPs (with PO-queries), one may use secure two-party computation to enable the verifier to ("blindly") deliver messages to the prover; these messages are computed based on the verifier's randomness and the previously committed messages of the prover, while the prover provides (in secrecy) the corresponding de-committed values, and the verifier remains ignorant about its own messages. Alternatively, one may use fully-homomorphic encryptions instead of the commitment, and have the verifier send encryptions of its own messages.

## 3 Main Results

As stated upfront, our focus is on interactive proofs of proximity (IPP) that use proof-oblivious queries (PO-queries). We initiate a systematic study of such systems, obtaining various results that distinguish the three models that were presented in Section 2 and relating them to MAPs and to standard testers.

When relating the various models, we 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. In contrast, separation results assert at least a sub-exponential gap between the corresponding complexities. Our first result asserts that IPPs in the isolated model can be efficiently emulated by standard testers.

► **Theorem 2** (efficient emulation of the isolated model by testers). *Suppose that  $\Pi$  is a property that can be verified in the isolated model (of IPPs with proof-oblivious queries) using  $q$  queries and  $c$  bits of communication. Then,  $\Pi$  has a standard tester of query complexity  $O((c + 1) \cdot q)$ .*

Theorem 2 significantly extends [19, Thm. 4.2], which establishes an analogous emulation of MAPs with proof-oblivious queries. Recall that MAPs are 1/2-round IPPs (i.e., the “interaction” is unidirectional with the prover sending a single message to the verifier), and in this case all three models of proof-oblivious queries coincide. In contrast, in Theorem 2 the communication between the prover and the verifier is only restricted by the total amount of bits communicated.

While the isolated model offers limited advantage over a standard tester, we show that the general model (of IPPs with proof-oblivious queries) is significantly stronger. In fact, not only is it stronger than standard testers, it is even not efficiently emulated by general MAPs (i.e., ones that are *not* restricted to proof-oblivious queries).

► **Theorem 3** (the general model of PO-queries cannot be efficiently emulated by MAPs). *There exists a property  $\Pi$  that can be verified in the general model (of IPPs with proof-oblivious queries) using  $O(1/\epsilon)$  queries and  $O(\epsilon^{-1} \log n)$  bits of communication, but any MAP for  $\Pi$  that uses a proof of length  $p$  must use at least  $\Omega(n^{1/2}/(p + 1))$  queries. Furthermore, the IPP with proof-oblivious queries uses a single round of communication.*

Theorem 3 is proved for the property consisting of all permutations over  $[n]$ , while using the lower bound established in [18, Lem. 4.3]. For the upper bound we use a proof system different than the (1-round) IPP presented in [18, Sec. 4.1], since the latter uses proof-dependent queries (i.e., it does not satisfy the condition of proof-oblivious queries).

The results regarding the model of pre-coordinated (PO-queries) IPPs bridge the two extremes captured by Theorems 2 and 3. On the one hand, 1-round IPPs in the pre-coordinated model can be efficiently emulated by standard testers. On the other hand, 3/2-round IPPs in the pre-coordinated model are significantly stronger than standard testers. Recall that in 1-round IPP the communication consists of two messages (i.e., the first message is sent from the verifier to the prover who responds with a single message), whereas in 3/2-round IPP the communication consists of three messages (i.e., the prover sends a single message, which is followed by a communication round as in a 1-round IPP).

► **Theorem 4** (the pre-coordinated model of PO-queries – 1 round vs 1.5 rounds). *The following dichotomy holds regarding IPPs (with proof-oblivious queries) in the pre-coordinated model.*

1. *Any property that can be verified in the pre-coordinated model, using a 1-round IPP with  $q$  queries and  $c$  bits of communication, has a standard tester of query complexity  $O((c + 1) \cdot q)$ .*
2. *There exists a property  $\Pi$  that can be verified in the pre-coordinated model using a 3/2-round IPP with  $O(1/\epsilon)$  queries and  $O(\epsilon^{-1} \log n)$  bits of communication, but any tester for  $\Pi$  uses at least  $\Omega(n^{1/2})$  queries.*

■ **Table 1** Can standard testers efficiently emulate IPPs that use PO-queries?

	1/2-round	1-round	3/2-round	$O(1)$ -round	$\omega(1)$ -round
isolated	Yes [19, Thm 4.2]	←			Yes (Thm 2)
public-coin		←			Yes (Thm 8)   No (Thm 5)
pre-coordinated	↓	Yes (Thm 4.1)	No (Thm 4.2)	→	
general POQ		No (Thm 3)		→	

Indeed, Part 1 asserts an efficient emulation at a complexity level that matches the bound in Theorem 2 (which refers to the isolated model), whereas Part 2 provides a separation analogous to Theorem 3 (which refers to the general model). Note that Part 1 implies a separation between 1-round IPPs in the general model (of proof-oblivious queries) and 1-round IPPs in the pre-coordinated model. On the other hand, the separation in Part 2 is only with respect to testers (rather than MAPs as in Theorem 3). In order to prove the hardness of the emulation of the pre-coordinated model by general MAPs, we use more rounds of the pre-coordinated model.

► **Theorem 5** (the pre-coordinated model of PO-queries cannot be efficiently emulated by MAPs). *There exists a property  $\Pi$  that can be verified in the pre-coordinated model (of IPPs with proof-oblivious queries) using  $\text{poly}(\epsilon^{-1} \log n)$  queries and  $\text{poly}(\log n)$  bits of communication, but any MAP for  $\Pi$  that uses a proof of length  $p$  must use at least  $\Omega(n^{0.999}/(p+1))$  queries. Furthermore, the IPP with proof-oblivious queries uses  $O(\log n)$  rounds of public-coin communication.*

Theorem 5 is proved by observing that the proof of [19, Thm. 3.28] uses an IPP that can be implemented in the pre-coordinated model. Specifically, we observe that the celebrated sum-check protocol of [20] can be implemented in the pre-coordinated model (see Section 4.3 in our technical report [15]). The general picture that emerges from the results reviewed so far (and the following Theorem 8) is summarized in Table 1.

### A focus on the pre-coordinated model

We find the pre-coordinated model especially appealing, since it allows the two main modules to operate independently of one another (but based on the same randomness, which is essential in light of Theorem 2). Theorems 4 and 5 frame our study of the pre-coordinated model, which is aimed at a finer understanding of what these proof systems can achieve. We loosely state two positive results and one negative result. The first result refers to a natural class of (general) MAPs.

► **Theorem 6** (3/2-rounds of IPPs in the pre-coordinated model can efficiently emulate a natural class of MAPs). *Suppose that  $\Pi$  is a property of functions from  $[n]$  to  $[m]$  that can be verified by a MAP that uses a proof of length  $\ell$  and makes  $q$  non-adaptive and uniformly distributed queries. Then,  $\Pi$  can be verified in the pre-coordinated model by a 3/2-round IPP that uses  $\tilde{q} = \tilde{O}(q)$  queries and total communication  $O(\tilde{q} \cdot \ell + \tilde{q}^2 \cdot \log(nm))$ .*

We stress that the hypothesis only requires that each query of the MAP verifier is uniformly distributed; their joint distribution, which may also depend on the given proof-string, is arbitrary (beyond this requirement).<sup>6</sup>

<sup>6</sup> Indeed, we are interested in the case that the original MAP uses proof-dependent queries. In this case, the dependence on the proof-string is manifested in the dependent between the queries.

One example of a property in the class is the set  $\{uuvv : uv \in \{0, 1\}^{n/2}\}$ , which is hard to test. We comment that the 3/2-round (pre-coordinated model) IPP of Theorem 4 (Part 2) is essentially obtained as a special case of Theorem 6. On the other hand, the result we actually obtain (see Theorem 4.7 in our technical report [15]) is much more general than Theorem 6. The next result refers to a much wider class of IPPs, but restricts the class of properties.

► **Theorem 7** (IPP in the pre-coordinated model can efficiently emulate general IPPs for any property of low-degree polynomials). *Let  $\Pi$  be a property (equiv., a subset) of  $m$ -variate polynomials of total degree  $d$  over a finite field  $\mathcal{F}$ , and suppose that  $\Pi$  can be verified by an  $r$ -round public-coin IPP of query complexity  $q$  and communication complexity  $c$ , and that  $d \leq |\mathcal{F}|/O(1 + \log_{|\mathcal{F}|} q)$ . Then,  $\Pi$  can be verified in the pre-coordinated model by a  $(r+q)$ -round IPP that uses  $O(q+d)$  queries and total communication  $O(c) + q \cdot (d+m) \cdot O(\log(q+|\mathcal{F}|))$ . Furthermore, if the original IPP uses non-adaptive queries, then the emulation can be done in  $r+1$  rounds.*

Recall that general  $r$ -round IPPs can be efficiently emulated by  $O(r)$ -round public-coin IPPs (see [21], following [17]); hence the result stated in Theorem 7 extends to the case that  $\Pi$  can be verified by any IPP, but the resulting round and communication complexities are increased in a suitable manner (so to account for the public-coin emulation). On the other hand, the verifier in the resulting IPP (i.e., in the pre-coordinated model) is not public-coin, and this is inherent because any property (including ones that have an efficient MAP but are hard to test) can be embedded in a property of polynomials whereas the following Theorem 8 limits the power of public-coin system.<sup>7</sup>

Indeed, it follows that *IPP of the pre-coordinated model cannot be efficiently emulated by public-coin IPP of the pre-coordinated model* (cf. Corollary 9).

► **Theorem 8** (efficient emulation by testers of public-coin  $O(1)$ -round IPPs in the pre-coordinated model). *Let  $\Pi$  be a property that can be verified by an  $r$ -round public-coin IPP in the pre-coordinated model using  $q$  queries and  $c$  bits of communication. Then,  $\Pi$  has a standard tester of query complexity  $(\text{poly}(r) \cdot c)^r \cdot q$ .*

Recall that any public-coin IPP that uses PO-queries is, by definition, in the coordinated model. The public-coin restriction in Theorem 8 is inherent in light of Theorem 4 (Part 2). Indeed, combining Theorems 8 and 4, we get.

► **Corollary 9** (private-coin 3/2-round IPPs in the pre-coordinated model cannot be efficiently emulated by  $O(1)$ -round public-coin IPPs in the pre-coordinated model). *There exists a property  $\Pi$  that can be verified in the pre-coordinated model using a 3/2-round IPP with  $O(1/\epsilon)$  queries and  $O(\epsilon^{-1} \log n)$  bits of communication, but any  $r$ -round public-coin IPP in the pre-coordinated model for  $\Pi$  of query complexity  $q$  uses at least  $\Omega((n/q^2)^{1/2r})$  bits of communication.*

<sup>7</sup> For example, starting from [19, Thm. 1.1], we have a property of functions  $f : [n] \rightarrow \{0, 1\}$  that has a MAP that uses a proof of length  $O(\log n)$  and makes  $O(1/\epsilon)$  queries, but cannot be tested with  $n^{0.999}$  queries. Associating  $[n]$  with  $H^m$  such that  $|H| = \log_2 n$  and considering the low-degree extensions of these functions over a field  $\mathcal{F}$  of size  $O(m \cdot |H|)$ , we obtain a property of degree  $m \cdot |H|$  polynomials that has a MAP that uses a proof of length  $O(\log n)$  and  $O(m \cdot |H|/\epsilon) = O(\epsilon^{-1} \log^2 n)$  non-adaptive queries, and is at least as hard to test as the original property. Note that the domain of the new functions has size  $s \stackrel{\text{def}}{=} |\mathcal{F}^m| = o(n^2)$ ; hence, hardness of testing holds for query complexity  $s^{0.49}$ .



Indeed, Corollary 9 provides yet another natural example of the gap between public-coin and general interactive proof systems.<sup>8</sup>

Turning back to Theorem 8, we mention that it extends (from public-coin IPPs of the pre-coordinated model) to IPPs of the pre-coordinated model that use *proof-oblivious messages* (i.e., the messages sent by the verifier are oblivious of the prover’s messages): See Theorem 4.10 in our technical report [15]. On the other hand, the emulation provided by Theorem 8 is relatively tight, since the proof system of [19, Lem. 3.29] has an  $r$ -round version that yields (via this emulation) a tester with almost optimal query complexity (see Appendix A.3 in our technical report [15]).

## 4 Techniques

Theorems 2, 4(1), 6, 7, and 8 are proved by emulating one model on another. In particular, we indicate the limitation of various types of IPPs that use PO-queries by showing that they can be efficiently emulated by standard testers. In contrast, we illustrate the power of IPPs that use PO-queries by showing that they can efficiently emulate general IPPs of certain types. We now review these two different types of emulations.

### 4.1 Emulating IPPs That Use PO-Queries by Standard Testers

We indicate the limitation of some types of IPPs (of the isolated and coordinated models) by showing that they can be efficiently emulated by standard testers. Such emulations are used in the proof of Theorem 2, Part 1 of Theorem 4, and Theorem 8. These emulations are best illustrated in the simplest contents of Theorem 2, which refers to the isolated model (of IPPs with PO-queries).

The starting point is the representation of all possible executions of a standard interactive proof system by a “game-tree” in which internal vertices represents execution prefixes and their children represent possible moves of the relevant party (cf. [3, Sec. 4] and [12, Apdx C.1]). The value of a leaf in the tree is the probability that the verifier accepts conditioned on the corresponding sequence of messages sent during the execution, where the value is either 0 or 1 in the special case that the sequence of messages fully determines the verifier’s randomness (e.g., when the system is of the public-coin type). The value of an internal vertex associated with the verifier equals the expected value of its children, when the expectation is according to the (conditional) probability space that determines the next message of the verifier.<sup>9</sup>

The value of an internal vertex associated with the prover equals the maximal value among the values assigned to its children.

Hence, the value of the root of the game-tree represents the probability that the verifier accepts, which in turn represents the expected value of the leaf reached in a random execution (with an optimal prover).

We stress that the foregoing description refers to standard interactive proof systems (not to IPPs). Turning to the isolated model, we observe that the value of a leaf in the tree may be defined as the probability that the (decision module of the) verifier accepts in a

<sup>8</sup> The best-known gaps were demonstrated in the context of zero-knowledge (cf. [13, Thm. 4.5.11] vs [13, Sec. 4.9.1]) and relatively efficient proving (cf. [22]). Closer to our context are the gaps shown for IPPs in the distribution testing setting [7, Sec. 6] and in the sample-based setting [11, Sec. 4.2].

<sup>9</sup> Note that this selection from a conditional probability space is not necessarily how the real (possibly private-coin) verifier acts, but this is how we view the effects of its actions in the analysis.

random execution of the querying module, when (the deciding module is) also fed with the corresponding output of the interacting module (i.e., the corresponding sequence of messages sent during the execution of the interacting module). The key observation is this value is a function of the identity of the specific leaf (corresponding to a specific interaction transcript) and *the outcome of a random process that does not depend on the identity of this leaf*. In other words, the same random process (representing the querying module) is used in all leaves. Hence, all that we need is (constant additive error) approximations of the values of all leaves, and all these approximations can be obtained based on the same repeated invocations of the random process. Thus, it suffices to invoke the random process for a number of times that is logarithmic (rather than linear) in the number of leaves. Using these approximated values of all leaves, we can compute the approximate value of the root of the tree, and Theorem 2 follows (i.e., the tester invokes the querying module  $O(c)$  times, where each invocation makes  $q$  queries, where  $c$  denotes the communication complexity of the IPPs and  $q$  its query complexity).

Turning to the pre-coordinated model, which is the focus of Theorems 4 and 8, we note that it is no longer the case that the random execution of the querying module is the same in all leaves (corresponding to all interaction transcripts). Indeed, the PO-queries condition implies that these executions do not depend on the prover messages, but they are conditioned by the choices made by the interacting module, since the querying and interacting modules use the same source of randomness. In particular, different conditional spaces may lead to different sequences of queries, and so our goal is to show that it suffices to use much fewer conditional spaces than the number of leaves. In the case of one-round IPPs (of the pre-coordinated model) this is achieved by observing that the conditional spaces are oblivious of the last prover message, and that it suffices to sample a constant number of verifier messages. This leads to establishing Part 1 of Theorem 4. In the case of public-coin IPPs, the verifier messages are also oblivious of the prover messages, and so the relevant parameter is the product of the number of verifier messages used in each round. Proving that it suffices to sample  $\text{poly}(r) \cdot c$  verifier messages for each of the  $r$  rounds, allows to establish Theorem 8. In other words, in this case, we prune the game-tree, leaving only  $\text{poly}(r) \cdot c$  children in each vertex that corresponds to a verifier move.

## 4.2 Emulating Some Types of IPPs by IPPs That Use PO-Queries

We illustrate the power of IPPs in the coordinated model by showing that they can efficiently emulate general IPPs of certain types. Such emulations are used in the proofs of Theorems 6 and 7.

Recall that Theorem 6 refers to any MAP that uses a proof of length  $\ell$  and makes  $q$  non-adaptive and uniformly distributed queries to a function  $f : [n] \rightarrow [m]$ , where these queries may depend on the proof given to the verifier (i.e., the individual queries may be related in a way that depends on the given proof). The key idea is to query  $f$  at a uniformly distributed point  $u \in [n]$ , and place  $u$  as the  $i^{\text{th}}$  query of a random execution of the MAP, where  $i \in [q]$  is selected uniformly at random, then ask the prover to provide the corresponding execution of the MAP, and accept if and only if the provided transcript is accepting and matches  $f(u)$ . That is, our 3/2-round IPP makes the random query  $u$ , and enters an interaction with the prover, who is supposed to send the MAP-proof as its first message. Denoting this message by  $\pi$ , our verifier selects  $i \in [q]$  uniformly at random, and selects a *random  $r$  such that on randomness  $r$ , upon given the MAP-proof  $\pi$ , the  $i^{\text{th}}$  query of MAP-verifier equals  $u$* . Our verifier sends  $r$  (or the corresponding query sequence) to the prover, who is supposed to respond with the corresponding answers. Letting  $\bar{a} = (a_1, \dots, a_q)$

denote the actual prover response, the deciding module accepts the value  $f(u)$  provided by the querying module and the transcript  $(\pi, r, \bar{a})$  provided by the interacting module if and only if  $a_i = f(u)$  and the MAP-verifier would have accepted the proof  $\pi$ , when using randomness  $r$  and getting the oracle answer-sequence  $(a_1, \dots, a_q)$ . The error is reduced to a constant by repeating the foregoing system for  $O(q)$  times, in parallel.

Turning to Theorem 7, recall that we are given an  $r$ -round public-coin IPP of query complexity  $q$  and communication complexity  $c$  for some property of low degree polynomials. Here, we make a random query per each query of the original prover, and ask our prover for (the univariate polynomial that describes) the values of the tested polynomial on the line that connects our random query and the real query. Specifically, the interacting module first emulates the original public-coin IPP, while relying on the fact that the verifier’s messages are independent of the verifier’s queries and the answers to these queries, and later it tries to obtain the answers to these queries as determined by the corresponding univariate (“line”) polynomials. (Actually, we use low-degree curves rather than lines, and, in addition, we also check that the tested function is a low-degree polynomial.)<sup>10</sup>

If our prover provides the wrong univariate polynomial (for a line), then the deciding module catches it with high probability (since we know the value of a random point on this line), and otherwise it uses the value (of the corresponding query) as determined by this univariate polynomial. We comment that the foregoing idea was applied quite extensively in the study of PCPs, starting with [10, 1], but it seems that its first appearance in the context of IPPs with proof-oblivious queries is due to [18, Lem. 5.4].

## 5 Some Comments About Our Conventions

This section contains brief comments about some of our conventions. We first note that although many results are stated in terms of properties of functions over  $[n]$ , one should view  $n$  as a generic (or varying) parameter rather than as fixed; formally, the results should be restated as in Definition 1. Also, whenever we write 0.999 (or 0.99), we actually mean any constant smaller than 1.

### On the computational complexity of the strategies

In this work we focus on the communication and query complexities of IPPs, while ignoring their computational complexities (both for the verifier and for the prescribed prover). This makes the negative results, which establish the limited power of certain IPPs (vis-a-vis testers), stronger. We mention that our positive results, which establish the power of certain IPPs, are actually obtained using relatively low computational complexity. In particular, with the exception of Theorem 6, the verifier strategy we present runs in time that is almost-linear in its query and communication complexities. In the exceptional case, the computational complexity of the verifier depends on the complexity of “reversed sampling” (i.e., sampling a random-pad that generates a given query), which is low in natural cases. In general, the computational complexity of IPPs and property testers is a secondary consideration, which is left for follow-up studies.

<sup>10</sup>We cannot rely on the fact that this test is conducted by the original IPPs, since we have to verify that the tested function is close to a low-degree polynomial in order to establish the soundness of the emulation.

### Using optimal prover strategies

The foregoing discussion justifies not specifying a prescribed proof-strategy for the completeness conditions, but rather referring to an optimal prover strategy both in the completeness and soundness condition. As far as the soundness condition is concerned, nothing is lost by this convention, but not specifying a prover (i.e., a honest prover) strategy for the completeness condition hinders the desire to have strategies that possess additional features such as relative efficiency or zero-knowledge. The gain in the convention adopted in the current work is that it slightly simplifies the definitions and facilitates some of the proofs; specifically, the accepting probability of a verifier  $V$  on any input can be expressed as a function  $p_V : \{0, 1\}^* \rightarrow [0, 1]$  that depends solely on  $V$ .

### Alphabet (or range of the functions)

The properties that we consider are sets of functions over  $[n]$ . Typically, these functions are either Boolean or range over  $[n]$ , but we also use functions of the form  $f : \mathcal{F}^m \rightarrow \mathcal{F}$  for some finite field  $\mathcal{F}$ . In all cases, the emulations preserve the function, so the type of queries is the same in both models. Lastly, we mention that one can always represent function of the form  $f : [n] \rightarrow \Sigma$  by Boolean functions over  $[n']$  such that  $n' = n \cdot O(\log |\Sigma|)$ . In this case one encodes elements of  $\Sigma$  by codewords (taken from a code with constant relative distance); this is done in order to preserve relative distances up to a constant factor.

### Communication rounds

Our notion of a communication round refers to the exchange of a pair of messages; hence, for  $r \in \mathbb{N}$ , an  $r$ -round IPP refers to the case that each of the two parties sends  $r$  messages, and means that the first message is sent by the verifier. In contrast, an  $(r - 0.5)$ -round protocol starts with the prover sending a message, and the verifier sending only  $r - 1$  messages. In both cases, the last message is by the prover, who sends a total of  $r$  messages.

### Non-adaptive queries

When saying that a general IPP uses non-adaptive queries we mean that the queries are determined based on the verifier's randomness and on the message it has received from the prover, but do not depend (directly) on the answers to prior queries. (An indirect dependence may arise if the verifier leaks information on its queries and the prover's messages depend on this information and on the value of the tested function at these queries.) Needless to say, non-adaptive queries in IPPs that use proof-oblivious queries depend only on the randomness of the querying module.

## 6 Related Works

As mentioned upfront, proof-oblivious queries were considered before, both in the context of MAPs (cf. [19, Def. 2.2]) and in the context of IPPs (cf. [21, Fn. 2] and [18, Sec. 5]). However, the focus of these works was elsewhere. Specifically, the focus of [19] was on introducing the MAP model (of non-interactive proofs of proximity), which is viewed as an NP (or MA) version of property testing, and demonstrating its power (see, e.g., [19, Thm. 1.1]). Within this context, showing that MAPs with proof-oblivious queries can be efficiently emulated by standard testers [19, Thm. 1.5] was viewed as an indication that proof-dependent queries are essential to the power of MAPs.

Likewise, the focus of [21] was on defining IPPs in their full generality and on studying their power. They mention that in some applications it may be helpful if the verifier’s queries can be made in advance of interacting with the prover, and note that all protocols in their work have this “oblivious-queries” feature, but they do not formally define or further study this feature. The focus of [18] is on separating MAPs (viewed as 1/2-round IPPs) from *one-round* IPPs (see [18, Thm. 1.1]). Within this context, showing that one-round IPPs that use “proof oblivious” queries can be efficiently emulated by standard tester [18, Thm. 1.2] is viewed as indication that proof-dependent queries are essential for that separation. As noted in Footnote 4, the notion of “proof oblivious” queries is not formally defined in [18], but it seems that the proof of [18, Thm. 1.2], which is presented in [18, Sec. 5.1], refers to the isolated model. We mention that a comment made (in passing) in [18, Sec. 1.1.1], which asserts that the sum-check protocol uses “proof oblivious” queries, seems to refer to the pre-coordinated model. The same holds for the contents of [18, Sec. 5.2.2].<sup>11</sup>

The notion of sample-based IPPs (SIPPs), introduced and studied in [11], considers IPPs in which the verifier can only obtain the value of the input at a sample of uniformly and independent distributed locations. These SIPPs can be viewed as a special case of our general model (of IPPs with PO-queries), whereas public-coin SIPPs can be viewed as a special case of our pre-coordination model. We mention that [11, Thm. 3.1] asserts that a natural class of standard non-adaptive testers can be efficiently emulated by SIPPs [11, Thm. 3.1]. The transformation underlying the proof of [11, Thm. 3.1] is similar to our proof of Theorem 6, although the results and the context are different.

We briefly mention one other line of work that has considered limits to the coordination of the verifier’s activities in an interactive proof. We refer to the work of [6], which was motivated by the study streaming model of interactive proofs [5, 8], studied a model of *two-party unidirectional communication* in which the receiver – called Bob – can interact with a *prover*. In this model, they make the distinction between protocols in which Bob’s messages to the prover are independent of the input, protocols in which Bob’s messages to the prover may depend on its input but not on the (single) message it received from the other party (called Alice), and general protocols (in this model). Indeed, the restriction considered in [6] refers to the coordination between the interaction with prover and input-examination activities, but it is not of the type that we study in this work. We remark that our pre-coordinated and isolated models do guarantee that the verifier’s interaction with the prover does not depend on the input (whereas the general PO-query model does not provide this guarantee).

Despite the fundamental differences between the models considered in our work and the models considered in [6], there are similarities between some arguments used in our work and arguments used in [6]. In particular, the proof of Part 1 of Theorem 4 is related to the proof of [6, Thm. 5.3] in the sense that in both cases a prover is emulated by considering the effect of each of its possible messages on a sample of invocations of the rest of the system.<sup>12</sup>

In contrast, a corresponding analogue of [6, Thm. 5.15], which would have stated an exponential complexity gap between 1/2-round and 1-round IPPs in the pre-coordinated model, is false.

<sup>11</sup>Hence, the chasm between public and private coins claimed in [18, Sec. 5.2] seems unsubstantiated, since the limits of the public coin version seem to be proved only for the isolated model, whereas the protocol for the private coin version is in the pre-coordinated model.

<sup>12</sup>Actually, a similar emulation technique can be traced to [2], where a “simultaneous model” with private randomness is emulated by an analogous deterministic model.

## 7 Models and Notation

In order to define the various models of *proof-oblivious queries* (PO-queries) interactive proofs of proximity (IPP), we decompose the verifier into three modules, called *querying*, *interacting*, and *deciding*, and denoted  $Q$ ,  $I$ , and  $D$ , respectively. The querying module (i.e.,  $Q$ ) is the only part that queries the input function, and the interacting module (i.e.,  $I$ ) is the only part that interacts with the prover. The final decision is made by the deciding module (i.e.,  $D$ ), which is fed with the outputs of the two other modules. The three models differ by the restrictions imposed on the coordination between them, where in all models the querying module gets no information from the other modules.

Recall that in standard interactive proofs, one denotes the output of the verifier by  $\langle P, V \rangle(x)$ , where  $x$  is a common input. In extensions that allow private inputs (see, e.g., [13, Def. 4.2.10]), one uses the notation  $\langle P(y), V(z) \rangle(x)$ , where  $z$  and  $y$  are corresponding private inputs.<sup>13</sup>

Here we have no real common input, so the output of the *interacting module*  $I$ , which may include its entire view, is denoted  $\langle P(y), I(z) \rangle$ . By default (unless stated otherwise),  $P$  will denote the optimal prover strategy, so there is no need to quantify over all strategies. Note that the prover has free access to the tested function, denoted  $f$ .

### The most restricted model – “isolated” modules

Here the querying and interacting modules are isolated, and feed their respective outputs (which equals their entire views of the execution) to the deciding module. Each of these modules has its own randomness; actually, the deciding module may be deterministic (since it may use randomness provided to it by one of the other modules). In this case, we write the random variable that represents the decision of the verifier as

$$D(Q^f(R_Q), \langle P(f), I(R_I) \rangle), \quad (1)$$

where  $R_Q$  and  $R_I$  are independent random variables representing the randomness of each module. We refer to systems captured by (1) as belonging to the *isolated model*.

### The intermediate model – “pre-coordinated” modules

Here the main two modules are pre-coordinated by their shared randomness. In this case, we write the random variable representing the decision as

$$D(Q^f(R), \langle P(f), I(R) \rangle), \quad (2)$$

where  $R$  is a random variable representing the shared randomness of both module. Again, without loss of generality, the deciding module may be deterministic. We refer to systems captured by (2) as belonging to the *pre-coordinated model*.

### The least restricted model – general PO-queries

Here, the interacting module gets the outcome of the querying module. Hence, we can write the random variable representing the decision as  $D(\langle P(f), I(Q^f(R), R) \rangle)$ . Actually, we may omit  $R$  from the input to  $I$ , since the output of the querying module may include  $R$ . Hence, the random variable representing the decision of the verifier is written as

$$D(\langle P(f), I(Q^f(R)) \rangle). \quad (3)$$

<sup>13</sup>These extensions are for formulating additional features such as relatively-efficient proving and auxiliary-input zero-knowledge.

Indeed, in this case, we can combine the interacting and deciding module, or rather integrate the deciding module in the interacting module (and write the verifier’s decision as  $\langle P(f), I(Q^f(R)) \rangle$ ). We refer to systems captured by (3) as belonging to the general (PO-queries) model. An alternative definitional approach to the general PO-queries model is presented in Appendix A.1 in our technical report [15].

---

## References

- 1 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998. doi:10.1145/278298.278306.
- 2 László Babai and Peter G. Kimmel. Randomized simultaneous messages: Solution of a problem of yao in communication complexity. In *Proceedings of the Twelfth Annual IEEE Conference on Computational Complexity, Ulm, Germany, June 24-27, 1997*, pages 239–246. IEEE Computer Society, 1997. doi:10.1109/CCC.1997.612319.
- 3 Mihir Bellare, Oded Goldreich, and Shafi Goldwasser. Randomness in interactive proofs. *Comput. Complex.*, 3:319–354, 1993. doi:10.1007/BF01275487.
- 4 Itay Berman, Ron D. Rothblum, and Vinod Vaikuntanathan. Zero-knowledge proofs of proximity. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 19:1–19:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ITCS.2018.19.
- 5 Amit Chakrabarti, Graham Cormode, Andrew McGregor, and Justin Thaler. Annotations in data streams. *ACM Trans. Algorithms*, 11(1):7:1–7:30, 2014. doi:10.1145/2636924.
- 6 Amit Chakrabarti, Graham Cormode, Andrew McGregor, Justin Thaler, and Suresh Venkatasubramanian. Verifiable stream computation and arthur-merlin communication. *SIAM J. Comput.*, 48(4):1265–1299, 2019. doi:10.1137/17M112289X.
- 7 Alessandro Chiesa and Tom Gur. Proofs of proximity for distribution testing. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 53:1–53:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. See full version in *ECCC*, TR17-155, 2017. doi:10.4230/LIPIcs.ITCS.2018.53.
- 8 Graham Cormode, Justin Thaler, and Ke Yi. Verifying computations with streaming interactive proofs. *Proc. VLDB Endow.*, 5(1):25–36, 2011. doi:10.14778/2047485.2047488.
- 9 Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. Fast approximate peps. In Jeffrey Scott Vitter, Lawrence L. Larmore, and Frank Thomson Leighton, editors, *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 41–50. ACM, 1999. doi:10.1145/301250.301267.
- 10 Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 29(1):1–28, 1999. doi:10.1137/S0097539792230010.
- 11 Guy Goldberg and Guy N. Rothblum. Sample-based proofs of proximity. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 – February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 77:1–77:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.77.
- 12 Oded Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*, volume 17 of *Algorithms and Combinatorics*. Springer, 1998. doi:10.1007/978-3-662-12521-2.
- 13 Oded Goldreich. *The Foundations of Cryptography – Volume 1: Basic Techniques*. Cambridge University Press, 2001. doi:10.1017/CB09780511546891.
- 14 Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017. doi:10.1017/9781108135252.

- 15 Oded Goldreich, Guy N. Rothblum, and Tal Skverer. On interactive proofs of proximity with proof-oblivious queries. *Electron. Colloquium Comput. Complex.*, TR22-124, 2022. URL: <https://eccc.weizmann.ac.il/report/2022/124>, arXiv:TR22-124.
- 16 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989. doi:10.1137/0218012.
- 17 Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In Juris Hartmanis, editor, *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 59–68. ACM, 1986. doi:10.1145/12130.12137.
- 18 Tom Gur, Yang P. Liu, and Ron D. Rothblum. An exponential separation between MA and AM proofs of proximity. *Comput. Complex.*, 30(2):12, 2021. doi:10.1007/s00037-021-00212-3.
- 19 Tom Gur and Ron D. Rothblum. Non-interactive proofs of proximity. *Comput. Complex.*, 27(1):99–207, 2018. doi:10.1007/s00037-016-0136-9.
- 20 Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992. doi:10.1145/146585.146605.
- 21 Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 793–802. ACM, 2013. doi:10.1145/2488608.2488709.
- 22 Salil P. Vadhan. On transformation of interactive proofs that preserve the prover's complexity. In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 200–207. ACM, 2000. doi:10.1145/335305.335330.