A Toolbox for Barriers on Interactive Oracle Proofs

Gal Arnon[1⋆], Amey Bhangale[2], Alessandro Chiesa[3⋆⋆], and Eylon Yogev[4⋆⋆⋆]

[1] gal.arnon@weizmann.ac.il
Weizmann Institute
[2] amey.bhangale@ucr.edu
UC Riverside
[3] alessandro.chiesa@epfl.ch
EPFL
[4] eylon.yogev@biu.ac.il
Bar-Ilan University

Abstract. Interactive oracle proofs (IOPs) are a proof system model that combines features of interactive proofs (IPs) and probabilistically checkable proofs (PCPs). IOPs have prominent applications in complexity theory and cryptography, most notably to constructing succinct arguments.

In this work, we study the limitations of IOPs, as well as their relation to those of PCPs. We present a versatile toolbox of IOP-to-IOP transformations containing tools for: (i) length and round reduction; (ii) improving completeness; and (iii) derandomization.

We use this toolbox to establish several barriers for IOPs:

- Low-error IOPs can be transformed into low-error PCPs. In other words, interaction can be used to construct low-error PCPs; alternatively, low-error IOPs are as hard to construct as low-error PCPs. This relates IOPs to PCPs in the regime of the sliding scale conjecture for inverse-polynomial soundness error.
- Limitations of quasilinear-size IOPs for 3SAT with small soundness error.
- Limitations of IOPs where query complexity is much smaller than round complexity.
- Limitations of binary-alphabet constant-query IOPs.

We believe that our toolbox will prove useful to establish additional barriers beyond our work.

Keywords: probabilistically checkable proofs; interactive oracle proofs; lower bounds

⋆ Supported in part by a grant from the Israel Science Foundation (no. 2686/20) and by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness.
⋆⋆ Supported in part by the Ethereum Foundation.
⋆⋆⋆ Supported in part by the BIU Center for Research in Applied Cryptography and Cyber Security in conjunction with the Israel National Cyber Bureau in the Prime Minister's Office, and by the Alter Family Foundation.

1 Introduction

Probabilistic proof systems have enabled breakthroughs in complexity theory and cryptography in areas such as zero-knowledge, delegation of computation, hardness of approximation, and more.

A probabilistically checkable proof (PCP) [6, 24] is a proof system in which a polynomial-time probabilistic verifier has query access to a proof string. The power of PCPs is often exemplified by the celebrated PCP theorem [5, 4]: every language in NP can be decided, with constant soundness error, by probabilistically examining a constant number of bits in a polynomial-size proof. Decades of PCP research have achieved many other goals and applications.

Yet challenging open problems about PCPs remain. For example, the shortest PCPs known to date have quasi-linear length [13, 20], and efforts to achieve linear length have not succeeded. As another example, it remains open to construct a PCP for NP with soundness error $1/n$, alphabet size $\mathrm{poly}(n)$, query complexity $O(1)$, and randomness complexity $O(\log n)$. The existence of such "low-error" PCPs is known as the "sliding-scale conjecture".

Interactive oracle proofs. Due to the lack of progress on these and other open problems, researchers introduced an interactive variant of PCPs called *interactive oracle proofs* (IOP) [12, 34]. A k-round IOP is a k-round IP where the verifier has PCP-like access to each prover message (the verifier may read a few symbols from any prover message).

A rich line of work constructs IOPs that provide significant efficiency improvements over known PCPs [11, 8, 10, 9, 14, 38, 15, 16, 35, 31, 28, 17, 36]. In particular, known IOPs achieve desirable properties such as linear proof length, fast provers, added properties such as zero-knowledge, and even good concrete efficiency. In turn, these IOPs have led to breakthroughs in the construction of highly-efficient cryptographic proofs, which have been widely deployed in real-world applications.

Another line of work shows that IOPs can also be used to prove hardness of approximation results for certain stochastic problems [19, 22, 3, 2].

What is the power of IOPs?. Since IOPs were invented to bypass open problems of PCPs, it is crucial to understand the limitations of IOPs, and the relation to the limitations of PCPs.

What are the limitations of IOPs, and how do they compare to PCPs?

For example: What trade-offs are there between round complexity, query complexity, and soundness error in IOPs? How small can the soundness error of an IOP be if we require constant query complexity but allow increasing the alphabet size (as in a sliding-scale PCP)?

In this paper, we explore these and other questions.

1.1 Our results

We show several results for IOPs in different regimes: (1) low-error IOPs imply low-error PCPs; (2) limitations of short IOPs; (3) limitations of high-round low-query IOPs; and (4) limitations of binary-alphabet constant-query IOPs. All these results follow from combining various tools from a new toolbox of transformations for IOPs. We discuss this toolbox in more detail in Section 2. We believe that our toolbox will prove useful to establish additional barriers beyond our work.

(1) Low-error IOPs imply low-error PCPs. The "sliding scale" conjecture [7] states that for every β with $1/\text{poly}(n) \leq \beta < 1$ there is a PCP system for NP that has perfect completeness, soundness error β, polynomial proof length over a $\text{poly}(1/\beta)$-size alphabet, constant query complexity, and logarithmic randomness complexity. A major open problem is constructing such PCPs when β is an inverse polynomial.

We show that (under a complexity assumption or using non-uniformity), a polylog-round IOP with inverse-polynomial soundness error and constant query complexity can be transformed into a sliding-scale PCP with inverse-polynomial soundness error.

Theorem 1 (informal)**.** *Let R be a relation with a public-coin IOP with perfect completeness, soundness error $1/n$, round complexity $\text{polylog}(n)$, alphabet size $\text{poly}(n)$, proof length $\text{poly}(n)$, and query complexity $O(1)$. Then under a derandomization assumption*[5] *(or alternatively by using a non-uniform verifier) R has a PCP with perfect completeness, soundness error $1/n$, alphabet size $\text{poly}(n)$, proof length $\text{poly}(n)$, and query complexity $O(1)$.*

Our full theorem, described in the full version of this paper, allows for tradeoffs between the parameters of the IOP and PCP.

Theorem 1 can be interpreted as a positive result or a negative result. The positive viewpoint is that efforts towards constructing sliding-scale PCPs can rely on interaction as an additional tool. The negative viewpoint is that constructing $\text{polylog}(n)$-round IOPs with sliding-scale parameters is as hard as constructing sliding-scale PCPs.

Our theorem does leave open the question of constructing $\text{poly}(n)$-round IOPs with constant query complexity and small soundness error.

(2) Limitations of short IOPs. While the shortest PCPs known have quasi-linear proof length, constructing linear-size PCPs remains a major open problem. In contrast, interaction has enabled IOPs to achieve linear proof length (e.g., [10]). Yet, we do not have a good understanding of the relation between proof length and soundness error for IOPs. We show that, under the randomized exponential-time hypothesis (RETH),[6] short IOPs for 3SAT have high soundness error.

[5] There exists a function in E with circuit complexity $2^{\Omega(n)}$ for circuits with PSPACE gates.

[6] RETH states that there exists a constant $c > 0$ such that $3\text{SAT} \notin \text{BPTIME}[2^{c \cdot n}]$.

Theorem 2 (informal)**.** *Assume* RETH *and suppose that there exists a public-coin IOP for n-variate* 3SAT *with the following parameters: perfect completeness, soundness error β, round complexity* $\mathrm{polylog}(n)$*, alphabet size λ, (total) proof length l, and query complexity q.*

If $\left(\frac{\mathsf{l}\cdot\log\lambda}{n}\right)^{\mathsf{q}} \leq n^{\mathrm{polylog}(n)}$, then $\beta > \Omega\left(\frac{n}{\mathsf{l}\cdot\log\lambda}\right)^{\mathsf{q}}$.

The theorem provides a barrier to improving some state-of-the-art PCPs. Dinur, Harsha, and Kindler [21] come close to a sliding-scale PCP in the inverse-polynomial regime: they construct a PCP for NP with perfect completeness, soundness error $1/\mathrm{poly}(n)$, alphabet size $n^{1/\mathrm{polyloglog}(n)}$, proof length $\mathrm{poly}(n)$, and query complexity $\mathrm{polyloglog}(n)$. While IOPs have been useful in improving proof length over PCPs, Theorem 2 implies that IOPs are unlikely to help achieving nearly-linear proof length in the parameter regime of [21] (even when significantly increasing alphabet size).

Corollary 1. *Assuming* RETH*, there is no public-coin IOP for n-variate* 3SAT *with perfect completeness, soundness error $1/n$, round complexity* $\mathrm{polylog}(n)$*, alphabet size $n^{\mathrm{polylog}(n)}$, proof length $n\cdot\mathrm{polylog}(n)$, and query complexity* $\mathrm{polyloglog}(n)$*.*

We leave open the question of whether IOPs in this parameter regime can be made to have linear proof length by using $O(n)$ rounds of interaction.

(3) Limitations of high-round low-query IOPs. Goldreich, Vadhan, and Wigderson [27] show that $\mathrm{IP}[\mathsf{k}] \neq \mathrm{IP}[o(\mathsf{k})]$ for every k, under reasonable complexity assumptions. In other words, IPs with k rounds cannot be "compressed" to have $o(\mathsf{k})$ rounds. In contrast, Arnon, Chiesa, and Yogev [2] show that k-round IPs can be modified so that the verifier *reads* $o(\mathsf{k})$ rounds. We show that reading $o(\mathsf{k})$ rounds comes at the price of a large soundness error.

Theorem 3. *Let $L \in \mathrm{AM}[\mathsf{k}]\backslash\mathrm{AM}[\mathsf{k}']$ be a language for $\mathsf{k}' < \mathsf{k}$ and suppose that L has a public-coin IOP with perfect completeness, soundness error β, round complexity k, alphabet size $2^{\mathrm{poly}(n)}$, proof length* $\mathrm{poly}(n)$*, and query complexity $\mathsf{q} \leq \mathsf{k}'$. Then $\beta \geq \Omega\left(\frac{\mathsf{k}'}{\mathsf{k}}\right)^{\mathsf{q}} - n^{-c}$ for every constant $c > 0$.*

This provides a barrier to improving the parameters of IOPs in [2]. They show that any language in $\mathrm{IP}[\log(n)]$ has an IOP with perfect completeness, soundness error $1/\mathrm{polylog}(n)$, round complexity $\mathrm{polylog}(n)$, alphabet size $2^{\mathrm{poly}(n)}$, and query complexity $O(1)$. By Theorem 3 the soundness error $1/\mathrm{polylog}(n)$ is tight unless $\mathrm{IP}[\log(n)] = \mathrm{IP}[O(1)]$. Moreover, since the soundness error of IOPs is closely related to the approximation factor for the value of stochastic constraint satisfaction problems (SCSP) (see [2]), our theorem additionally provides barriers to proving hardness of approximation results for SCSPs using IOPs.

(4) Limitations of binary-alphabet constant-query IOPs. PCPs with a binary alphabet and small query complexity cannot have good soundness. In more detail, assuming the randomized exponential-time hypothesis, any binary-alphabet PCP with perfect completeness, soundness error β, and query complexity q satisfies the following.

- If $\mathsf{q} = 2$ then $\beta = 1$ (i.e., no such PCPs exist). This follows from the fact that we have linear time algorithms to check satisfiability of every binary-alphabet 2-ary constraint satisfaction problem.
- If $\mathsf{q} = 3$ then $\beta > 5/8$. Zwick [39] gives a polynomial-time algorithm that, on input a satisfiable CSP with binary alphabet and arity 3, distinguishes whether the CSP is satisfiable or whether every assignment satisfies at most a $5/8$ fraction of the constraints. This implies that, unless P = NP, every PCP for NP with binary alphabet, polynomial size, and query complexity 3 must have soundness error greater than $5/8$.[7] Håstad [30] shows that this lower bound on soundness error is essentially optimal: for every $\varepsilon > 0$, he constructs a PCP for NP with perfect completeness, soundness error $5/8 + \varepsilon$, binary alphabet, polynomial proof length, and query complexity 3.

We ask whether interaction can help in further reducing the soundness error in the constant-query regime. Our next result shows that this is unlikely if the number of rounds is not large.

Theorem 4. *Assume* RETH *and suppose that there exists a non-adaptive public-coin IOP for n-variate* 3SAT *with the following parameters: perfect completeness, soundness error β, round complexity k, alphabet size 2, proof length $2^{o(n)}$, query complexity q, verifier randomness r, and verifier running time $2^{o(n)}$.*
- *If $\mathsf{q} = 2$ then $\beta > 1 - \varepsilon$ for every ε satisfying $\mathsf{k} \cdot \log(\mathsf{r} \cdot n/\varepsilon) = o(n)$.*
- *If $\mathsf{q} = 3$ then $\beta > 5/8 - \varepsilon$ for every ε satisfying $\mathsf{k} \cdot \log(\mathsf{r} \cdot n/\varepsilon) = o(n)$.*

For example, assuming RETH, there is no public-coin IOP with perfect completeness, soundness error $\beta = 1 - 2^{-o(n)}$, round complexity $\mathsf{k} = \text{polylog}(n)$, alphabet size 2, proof length $2^{o(n)}$, query complexity 2, and verifier randomness $\mathsf{r} = 2^{o(n)}$.

The bound on the query complexity of PCPs can be extended to q queries for any $\mathsf{q} = O(1)$ for which there is a polynomial-time algorithm that decides q-ary CSPs. Theorem 4 generalizes similarly to match the soundness error for PCPs. However, for $\mathsf{q} > 3$, we do not know the exact optimal soundness error for PCPs with perfect completeness [29].

Constructing an IOP for 3SAT with polynomial round complexity, binary alphabet, constant query complexity, and small soundness error remains an open problem.

1.2 Related work

Barriers on probabilistic proofs. We describe known limitations about PCPs, IPs, and IOPs.

- *PCPs.* If P $\neq$ NP then, for every $\mathsf{q} = o(\log n)$ and $\mathsf{r} = o(\log n)$, NP has no non-adaptive PCP with alphabet size $\lambda = O(1)$, query complexity q, and randomness complexity r. Indeed, the PCP-to-CLIQUE reduction in [23], given

[7] Assuming ETH, the proof length of the PCP can be $2^{o(n)}$.

an instance $\mathbb{x}$ for the language L of the PCP, produces, in polynomial time, a graph of size $\lambda^{\mathsf{q}} \cdot 2^{\mathsf{r}} \ll n$ whose maximum clique size is either large (if $\mathbb{x} \in L$) or small (if $\mathbb{x} \notin L$), where the gap between these sizes depends on the PCP's completeness and soundness errors. By iteratively applying that reduction a polynomial number of times, one can (in polynomial time) reduce $\mathbb{x}$ to a graph G of size $O(\log n)$, while preserving the large-or-small property of the maximum clique. Since the size of G is logarithmic, one can then determine in polynomial time whether the largest clique in G is large or small, and thereby decide membership for the original instance $\mathbb{x}$.

Moreover, if P $\neq$ NP then NP does not have non-adaptive PCPs with alphabet size $\lambda = O(1)$, query complexity $\mathsf{q} = O(1)$, and randomness complexity $\mathsf{r} = O(\log n)$ with soundness error $\beta < \frac{\log \lambda}{\lambda^{\mathsf{q}-1}}$. Indeed, such a non-adaptive PCP can be converted into a CSP of size $\mathrm{poly}(n)$, and any efficient algorithm for approximating the CSP's number of satisfied constraints imposes a limitation on the soundness error β. For example, the bound $\frac{\log \lambda}{\lambda^{\mathsf{q}-1}}$ follows from the approximation algorithm in [32]. Assuming ETH, these limitations can be extended to PCPs with super-polynomial proof length and super-constant alphabet size and query complexity. See the full version of this paper for a quantitative proof of how to combine PCPs with small soundness error for 3SAT and polynomial-time approximation algorithms for CSPs in order to decide 3SAT faster than is possible under ETH.

Notice that an adaptive PCP with alphabet size λ and query complexity q can be converted into a non-adaptive PCP with query complexity λ^{q}, which is constant when $\lambda = O(1)$ and $\mathsf{q} = O(1)$. Hence the above discussion applies to adaptive PCPs in this regime as well.

- *IPs.* [26] show that public-coin IPs with bounded prover communication complexity can be decided in non-trivial (probabilistic) time. [27] strengthen these results for the case of private-coin IPs, showing that similar bounds on communication imply that the complement of the language can be decided in non-trivial non-deterministic time. Such results are limitations on IPs for languages believed to be hard, such as SAT.
- *IOPs.* In order to derive barriers for succinct arguments, [18] extend to IOPs the limitations of [26], showing barriers for IOPs with small soundness error relative to query complexity.

  [33] show limitations for *succinct* IOPs for circuit SAT (CSAT), where the proof length is polynomial in the number n of circuit inputs. The results cover different parameters, depending on the "plausibility" of the complexity assumption used. For example (on the most probable end), suppose that the satisfiability of a circuit C cannot be decided by a $\mathrm{poly}(n)$-space algorithm following $\mathrm{poly}(|C|)$-time preprocessing. Then there is no succinct IOP for CSAT with constant round complexity and logarithmic query complexity.

IOP-to-IOP transformations. Our toolbox (outlined in Section 2) contains IOP-to-IOP transformations that include round reduction, achieving perfect completeness, and derandomization.

– [3, 2] provide IOP-to-IOP transformations for round reduction and achieving perfect completeness, but we cannot use them because those transformations do *not* preserve query complexity of the IOP (a key property for us).
– [33] show that any public-coin IOP can be transformed into one with less interaction randomness at the cost of introducing a "common reference string" (CRS) and satisfying only non-adaptive soundness. Their main goal is to achieve randomness complexity that depends (logarithmically) only on the prover-to-verifier communication complexity (but not the instance length) and on an error parameter over the choice of the CRS. They also show that the CRS can be replaced with non-uniform advice for the verifier at the cost of increasing the randomness complexity to also depend (logarithmically) on the instance length. Our derandomization lemma focuses on IOPs with a non-uniform verifier and allows choosing the target randomness complexity, rather than optimizing with regards to the prover-to-verifier communication complexity.
– [1] show how to derandomize *private-coin IPs* via non-uniform advice or PRGs. Our derandomization lemma applies to public-coin IOPs.

2 Techniques

We describe our tools for IOPs and sketch their proofs, and then show how they can be applied to achieve our main results. Further details on how these tools are constructed can be found in the full version of this paper. The tools are divided into three groups.

1. **Tools for length and round reduction:** Section 2.1 outlines transformations that decrease the length and round complexity of IOPs with low query complexity.
2. **Tools for improving completeness:** Section 2.2 outlines transformations that improve the completeness errors of IOPs.
3. **Tools for derandomization:** Section 2.3 outlines transformations that decrease the number of random bits used by the IOP verifier.

Following the presentation of our toolbox, in Section 2.4 we explain how we use the tools (in conjunction with additional arguments) to derive the theorems described in Section 1.1.

2.1 Tools for length and round reduction

We describe how to decrease the length and round complexity of IOPs.

Lemma 1 (informal)**.** *Let R be a relation with a public-coin IOP $(\mathbf{P}, \mathbf{V})$ with completeness error α, soundness error β, round complexity k, alphabet size λ, per-round proof length l, query complexity q, per-round verifier randomness r, and verifier running time vt.*

1. ***Length reduction:*** *Let ℓ be a parameter with $\mathsf{q} \leq \ell \leq \mathsf{k} \cdot \mathsf{l}$. Then R has a public-coin IOP with completeness error $1-(1-\alpha)\cdot(\ell/(e\cdot\mathsf{k}\cdot\mathsf{l}))^{\mathsf{q}}$, soundness error β, round complexity k, alphabet size λ,* ***total proof length ℓ****, query complexity q, per-round verifier randomness $\mathsf{r}+\ell\cdot\log(\mathsf{k}\cdot\mathsf{l})$, and verifier running time $\mathrm{poly}(\mathsf{vt},\ell)$.*
2. ***Round reduction:*** *Let k' be a parameter with $\mathsf{q} \leq \mathsf{k}' \leq \mathsf{k}$. Then R has a public-coin IOP with completeness error $1-(1-\alpha)\cdot(\mathsf{k}'/(e\cdot\mathsf{k}))^{\mathsf{q}}$, soundness error β,* ***round complexity $\mathsf{k}'+1$****, alphabet size λ, per-round proof length l, query complexity q, per-round verifier randomness $\mathsf{k}\cdot(\mathsf{r}+\log\mathsf{k})$, and verifier running time $\mathrm{poly}(\mathsf{vt})$.*
3. ***Unrolling to PCP:*** *R has a PCP with completeness error α, soundness error β, alphabet size λ, proof length $\mathsf{l}\cdot 2^{O(\mathsf{k}\cdot\mathsf{r})}$, query complexity q, randomness $\mathsf{k}\cdot\mathsf{r}$, and verifier running time $\mathrm{poly}(\mathsf{vt})$.*

Below we sketch the proofs of Items 1 and 2. Item 3 is folklore and follows by setting the PCP to equal the interaction tree of the IOP.

Length reduction. The length of low-query IOPs can be reduced while incurring an increase in the completeness error. The intuition is that if the IOP has query complexity $\mathsf{q} \ll \mathsf{k}\cdot\mathsf{l}$, then each symbol in the proof is read by the verifier with small probability. Hence, if the prover omits a random subset of the proof symbols, the verifier is unlikely to require these missing symbols.

Construction 1 (informal)**.** The new prover $\mathbf{P}'$ receives as input an instance $\mathbb{x}$ and a witness $\mathbb{w}$, while the verifier $\mathbf{V}'$ receives as input the instance $\mathbb{x}$. They interact as follows.

1. $\mathbf{V}'$ *guesses the locations that* $\mathbf{V}$ *will query.* $\mathbf{V}'$ samples and sends a random set $I \subseteq [\mathsf{k}\cdot\mathsf{l}]$ of ℓ indices from among all the prover message symbols.
2. *The original IOP is simulated with prover messages omitted according to I.* For every $j \in [\mathsf{k}]$:
   (a) $\mathbf{V}'$ sends $\rho_j \leftarrow \{0,1\}^{\mathsf{r}}$.
   (b) $\mathbf{P}'$ computes $\pi_j := \mathbf{P}(\mathbb{x},\mathbb{w},\rho_1,\ldots,\rho_j)$ and sends π'_j equal to π_j with symbols outside of I omitted.
3. $\mathbf{V}'$ *simulates* $\mathbf{V}$*, and rejects if any queries are made outside of I.* $\mathbf{V}'$ simulates the decision stage of $\mathbf{V}$ given input $\mathbb{x}$. Whenever an index $i \in I$ is queried, return the appropriate symbol from the prover messages. If an index $i \notin I$ is queried, then immediately reject. Output the same answer as $\mathbf{V}$.

The *total* proof length is ℓ since the prover $\mathbf{P}'$ sends only those symbols whose index is in I (which has size ℓ). The per-round verifier randomness at most $\mathsf{r}+\ell\cdot\log(\mathsf{k}\cdot\mathsf{l})$ because in the first round the verifier sends I (which can be described with $\ell\cdot\log(\mathsf{k}\cdot\mathsf{l})$ random bits) and then it sends its first message of r bits. The rest of the complexity parameters follow straightforwardly from the construction.

Soundness follows from the fact that the changes made to the IOP can only increase the chance that the verifier rejects. We sketch the proof of completeness. Fix some $\mathbb{x} \in L$. The locations read by $\mathbf{V}$ are independent of the set I. Therefore,

the probability that $\mathbf{V}$ queries outside the set I is $\binom{\mathsf{k}\cdot\mathsf{l}-\mathsf{q}}{\ell-\mathsf{q}}/\binom{\mathsf{k}\cdot\mathsf{l}}{\ell} \geq (\ell/(e\cdot\mathsf{k}\cdot\mathsf{l}))^{\mathsf{q}}$. Conditioned on $\mathbf{V}$ querying only inside I, $\mathbf{V}$ accepts with probability at least $1-\alpha$. Hence the probability that the new verifier $\mathbf{V}'$ accepts is at least $(1-\alpha)\cdot(\ell/(e\cdot\mathsf{k}\cdot\mathsf{l}))^{\mathsf{q}}$.

Round reduction. We sketch how the round-complexity of low-query IOPs can be reduced. The intuition behind this lemma is similar to that described for length reduction: if $\mathsf{q} \ll \mathsf{k}$, then the verifier is unlikely to need most of the rounds, so removing a random subset of the rounds does not harm completeness by much. Below we describe the transformation for IOP round reduction.

Construction 2 (informal)**.** The new prover $\mathbf{P}'$ receives as input an instance $\mathbb{x}$ and a witness $\mathbb{w}$, while the verifier $\mathbf{V}'$ receives as input the instance $\mathbb{x}$. They interact as follows.

1. $\mathbf{V}'$ *guesses the rounds that* $\mathbf{V}$ *will query.* $\mathbf{V}'$ samples and sends a random set $I \subseteq [\mathsf{k}]$ of k' indices. Denote $I := (i_1, \ldots, i_{\mathsf{k}'})$ with $i_j < i_{j+1}$ and let $i_0 := 1$.
2. *The original IOP is simulated with rounds omitted according to* I. For every $j \in [\mathsf{k}']$:
   (a) $\mathbf{V}'$ sends $\rho_{i_{(j-1)}+1}, \ldots, \rho_{i_j} \leftarrow \{0,1\}^{\mathsf{r}}$.
   (b) $\mathbf{P}'$ computes and sends $\pi_j := \mathbf{P}(\mathbb{x}, \mathbb{w}, \rho_1, \ldots, \rho_{i_j})$.
3. $\mathbf{V}'$ *simulates* $\mathbf{V}$*, and rejects if any queries are made outside of* I. $\mathbf{V}'$ samples $\rho_{i_{\mathsf{k}'}+1}, \ldots, \rho_{\mathsf{k}} \leftarrow \{0,1\}^{\mathsf{r}}$ simulates the decision stage of $\mathbf{V}$ given input $\mathbb{x}$ and verifier messages $\rho_1, \ldots, \rho_{\mathsf{k}}$. Whenever an index in round $i \in I$ is queried, return the appropriate symbol in the prover messages. If a round $i \notin I$ is queried, then immediately reject. Output the same answer as $\mathbf{V}$.

A technical remark: as written above, the protocol is not public-coin because the verifier's first message I dictates the length of subsequent verifier messages. Nevertheless, the protocol can be made public-coin by padding verifier messages to $\mathsf{k}\cdot\mathsf{r}$ bits. The prover and verifier act as in the protocol description, ignoring the padding bits. The verifier additionally sends $\mathsf{k}'\cdot\log\mathsf{k}$ bits as the choice of the set I. Thus, the per-round randomness of the verifier is $\mathsf{k}\cdot\mathsf{r}+\mathsf{k}'\cdot\log\mathsf{k} \leq \mathsf{k}\cdot(\mathsf{r}+\log\mathsf{k})$.

2.2 Tools for improving completeness

A transformation for achieving perfect completeness for IPs is shown in [25]. Directly applying that transformation to IOPs increases the query complexity of the protocol significantly. We show a variant of the transformation in [25] that preserves query complexity up to a small additive constant.

Lemma 2 (informal)**.** *Let* R *be a relation with a public-coin IOP* $(\mathbf{P}, \mathbf{V})$ *with completeness error* α*, soundness error* β*, round complexity* k*, alphabet size* λ*, per-round proof length* l*, query complexity* q*, per-round verifier randomness* r*, and verifier running time* vt*.*

Then R *has a public-coin IOP with* ***perfect completeness****, soundness error* $O\left(\frac{\beta\cdot\mathsf{k}\cdot\mathsf{r}}{\log(1/\alpha)}\right)$*, round complexity* $\mathsf{k}+1$*, alphabet size* $\max\{\lambda, 2^{\mathsf{k}\cdot\mathsf{r}}\}$*, per-round proof*

length $O\left(\frac{\mathsf{l}\cdot\mathsf{k}\cdot\mathsf{r}}{\log(1/\alpha)}\right)$*, query complexity* $\mathsf{q}+2$*, per-round verifier randomness* r*, and verifier running time* $\mathrm{poly}(\mathsf{vt})$.

Remark 1. If only small completeness error is desired (rather than completeness error 0), then this can be achieved with similar query complexity but smaller overhead to the alphabet size. See the full version of this paper for more details.

Review: perfect completeness for IPs. Consider the set S of verifier random coins $\vec{\rho} = (\rho_1, \ldots, \rho_{\mathsf{k}})$ (over the entire protocol) where the honest prover has a strategy to make the verifier accept if it is sent these strings while interacting with the verifier. Given the matching prover messages, the verifier can efficiently check whether $\vec{\rho} \in S$. [25] shows that for large enough t there exist "shifts" $\vec{z}_1, \ldots, \vec{z}_t$ such that for *every* choice of verifier randomness $\vec{\rho}$ there exists j such that $(\vec{z}_j \oplus \vec{\rho}) \in S$. It follows that the honest prover needs only to send these shifts, and then run the protocol with the verifier, giving answers matching each shift. At the end of the protocol, the verifier accepts if and only if $\vee_{j=1}^{t}\left((\vec{z}_j \oplus \vec{\rho}) \stackrel{?}{\in} S\right) = 1$. The soundness error degrades by a multiplicative factor of t since a malicious prover only needs to convince the verifier in one execution.

Perfect completeness for IOPs. The aforementioned verifier computes the "OR" of t expressions. We observe that, in order to prove the claim $\vee_{j=1}^{t}\left((\vec{z}_j \oplus \vec{\rho}) \stackrel{?}{\in} S\right) = 1$, it suffices for the prover to send the verifier a *single* index j where $(\vec{z}_j \oplus \vec{\rho}) \in S$, which is then checked by the verifier. The verifier only needs to check *a single execution* of the IOP, rather than t, and so the query complexity of the protocol is preserved up to reading the index j and shift $\vec{z}_j$.

Construction 3. Let $t := 2 \cdot \left(\frac{\mathsf{r}\cdot\mathsf{k}}{\log(1/\alpha)}\right)$. The new prover $\mathbf{P}'$ receives as input an instance $\mathbb{x}$ and a witness $\mathbb{w}$, while the verifier $\mathbf{V}'$ receives as input the instance $\mathbb{x}$. They interact as follows.

1. $\mathbf{P}'$ *sends* t *"shifts" for the verifier randomness.* $\mathbf{P}'$ sends
$$\vec{z}_1, \ldots, \vec{z}_t = (z_{1,1}, \ldots, z_{1,\mathsf{k}}), \ldots, (z_{t,1}, \ldots z_{t,\mathsf{k}}) \in \{0,1\}^{\mathsf{r}\cdot\mathsf{k}} \ ,$$
to the verifier such that for every $\vec{\rho}$ there exists j where $(\vec{z}_j \oplus \vec{\rho}) \in S$ (i.e., the original prover $\mathbf{P}$ has an accepting strategy for verifier randomness $(\vec{z}_j \oplus \vec{\rho})$).
2. *Original IOP is simulated, where for every verifier message, prover replies with a message for each shifted randomness.* For $i = 1, \ldots, \mathsf{k}$:
   - $\mathbf{V}'$: Choose $\rho_i \leftarrow \{0,1\}^{\mathsf{r}}$ uniformly and send to the prover.
   - $\mathbf{P}'$: Send $\{\pi_{j,i}\}_{j\in[t]}$ where $\pi_{j,i} := \mathbf{P}(\mathbb{x}, \mathbb{w}, \rho_1 \oplus z_{j,1}, \ldots, \rho_i \oplus z_{j,i})$.
3. *Prover sends index* j *of shift where its messages succeed in convincing the verifier.* $\mathbf{P}'$: If there exists an index $j \in [t]$, such that $\mathbf{V}^{\pi_{j,1},\ldots,\pi_{j,\mathsf{k}}}(\mathbb{x}, \rho_1 \oplus z_{j,1}, \ldots, \rho_{\mathsf{k}} \oplus z_{j,\mathsf{k}}) = 1$, then send j to the verifier $\mathbf{V}'$ as a non-oracle message. Otherwise, send $\perp$.
4. $\mathbf{V}'$ *checks that* $\mathbf{V}$ *accepts the "shifted"* j*-th execution.* $\mathbf{V}'$: Receive j as a non-oracle message.

(a) If $j = \perp$, then reject.
(b) Otherwise, query $\vec{z}_j = (z_{j,1}, \ldots, z_{j,\mathsf{k}})$ and check that

$$\mathbf{V}^{\pi_{j,1},\ldots,\pi_{j,\mathsf{k}}}(\mathbb{x}, \rho_1 \oplus z_{j,1}, \ldots, \rho_{\mathsf{k}} \oplus z_{j,\mathsf{k}}) = 1 \ ,$$

querying the appropriate proofs as required by $\mathbf{V}$.

2.3 Tools for derandomization

We show how to derandomize public-coin IOPs based on non-uniform advice or based on pseudorandom generators (PRGs), while preserving the use of public-coins. Both transformations achieve logarithmic randomness complexity but slightly increase completeness and soundness error. Round complexity, proof length, and query complexity are preserved.

Lemma 3 (informal)**.** *Let R be a relation with a public-coin IOP $(\mathbf{P}, \mathbf{V})$ with completeness error α, soundness error β, round complexity k, alphabet size λ, per-round proof length l, query complexity q, per-round verifier randomness r, and verifier running time vt.*

1. ***Derandomization using PRGs:*** *Suppose that there exists a PRG against polynomial-size* PSPACE *circuits with seed length ℓ, error ε and evaluation time $\mathsf{t}_{\mathsf{PRG}}$. Then R has a public-coin IOP with completeness error $1 - O((1-\alpha) - \varepsilon \cdot \mathsf{k}^2)$, soundness error $O(\beta + \varepsilon \cdot \mathsf{k}^3)$, round complexity k, alphabet size λ, per-round proof length l, query complexity q, **per-round verifier randomness ℓ**, and verifier running time* $\mathrm{poly}(\mathsf{vt}, \mathsf{t}_{\mathsf{PRG}})$.
   (Such a PRG with seed length $\ell = O(\log |\mathbb{x}|)$, error $\varepsilon = 1/\mathrm{poly}(|\mathbb{x}|)$ and computation time $\mathsf{t}_{\mathsf{PRG}} = \mathrm{poly}(|\mathbb{x}|)$ exists if there exists a function in E with circuit complexity $2^{\Omega(n)}$ for circuits with PSPACE *gates.)*
2. ***Derandomization using non-uniformity:*** *Let $\varepsilon \in (0,1)$ be a parameter. Then R has a public-coin IOP with completeness error $\alpha + \mathsf{k} \cdot \varepsilon$, soundness error $\beta + \mathsf{k} \cdot \varepsilon$, round complexity k, alphabet size λ, per-round proof length l, query complexity q, **per-round verifier randomness $\Theta(\log((\mathsf{r} \cdot \mathsf{k} + |\mathbb{x}|)/\varepsilon))$**, and verifier running time* $\mathrm{poly}(\mathsf{vt}, \mathsf{k}, \mathsf{l}, \mathsf{r}, 1/\varepsilon)$*, where the verifier receives* $\mathrm{poly}(|\mathbb{x}|, \mathsf{k}, \mathsf{r}, 1/\varepsilon)$ *bits of non-uniform advice. Moreover, a random string constitutes good advice with probability $1 - 2^{-|\mathbb{x}|}$.*

We focus the overview below on Item 1. Item 2 can be shown in a similar manner.

Derandomization using PRGs. We show that IOPs can be derandomized using a pseudo-random generator. In this transformation, the verifier samples seeds for the PRG rather than uniform random messages. Thus the verifier randomness per-round is as small as a seed of the PRG.

Construction 4 (informal)**.** On instance $\mathbb{x}$ and witness $\mathbb{w}$, the protocol $(\mathbf{P}', \mathbf{V}')$ proceeds as follows:

1. *Simulate original IOP where verifier messages are chosen using the PRG.* For $j = 1, \ldots, \mathsf{k}$:
   (a) $\mathbf{V}'$: Sample and send a random $\rho_j \leftarrow \{0,1\}^{\ell}$.
   (b) $\mathbf{P}'$: Compute and send the prover message π_j that maximizes the probability that $\mathbf{V}$ accepts where all of the verifier messages are chosen using the PRG G.
2. $\mathbf{V}'$: Accept if and only if $\mathbf{V}^{\pi_1,\ldots,\pi_{\mathsf{k}}}(\mathbb{x}, \mathsf{G}(\rho_1), \ldots, \mathsf{G}(\rho_{\mathsf{k}})) = 1$.

The verifier sends ℓ_{PRG} bits of randomness in each round, since it sends a seed for the PRG. The rest of the complexity parameters follow straightforwardly from the construction.

Interaction trees. The *interaction tree* of a protocol on input $\mathbb{x}$, denoted $T_{\mathbb{x}}$ is the full tree of all possible transcripts corresponding to each choice of prover and verifier messages. The leaves are labelled as accepting or rejecting corresponding to whether the verifier accepts or rejects the full transcript represented by the leaf.

The *value* of an interaction tree $T_{\mathbb{x}}$, denoted by $\mathsf{val}(T_{\mathbb{x}})$, is the probability of reaching an accepting leaf from the root of the tree in a walk on the tree where verifier messages are chosen uniformly at random and prover messages are chosen so as to maximize the probability of reaching an accepting node. The notion of value extends to sub-trees as well, where the value is the probability of reaching an accepting leaf when beginning on the root of the sub-tree. Notice that $\mathsf{val}(T_{\mathbb{x}}) = \max_{\tilde{\mathbf{P}}}\{\Pr[\langle \tilde{\mathbf{P}}, \mathbf{V}\rangle(\mathbb{x}) = 1]\}$. Moreover, $\mathsf{val}(T_{\mathbb{x}})$ can be computed in space that is polynomial in $|\mathbb{x}|$, the round complexity, the proof length, and the verifier randomness of the IOP.

Completeness and soundness. Completeness and soundness follow straightforwardly from Section 2.3, which says that the value of the interaction tree of the IOP does not change by much when the verifier messages are sampled via a PRG.

Claim. Let G be a PRG against circuits of size $\mathrm{poly}(|\mathbb{x}|)$ with PSPACE gates. Then for every instance $\mathbb{x}$:

$$O(\mathsf{val}(T) - \epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^2) \leq \mathsf{val}(T_{\mathsf{G}}) \leq O(\mathsf{val}(T) + \epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^3) \ ,$$

where T is the interaction tree of the IOP and T_{G} is the interaction tree of $(\mathbf{P}', \mathbf{V}')$, which is identical to T except verifier randomness is always sampled using the PRG G.

We give a simplified sketch of the proof of the claim. Let $T^{(0)} := T_{\mathsf{G}}$ and for $i = 1, \ldots, \mathsf{k}$ let $T^{(i)}$ be the tree of an intermediate protocol where the messages $\rho_1, \ldots, \rho_i$ are chosen uniformly at random and $\rho_{i+1}, \ldots, \rho_{\mathsf{k}}$ are chosen from the PRG. Notice that $T^{(\mathsf{k})} = T$.

We show that, under a simplifying assumption to be described later, there exist circuit families $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(\mathsf{k})}$ each comprised of circuits of size $\mathrm{poly}(|\mathbb{x}|, \mathsf{k}, \mathsf{l}, \mathsf{r})$

that have PSPACE gates, such that if G fools $\mathcal{C}^{(i)}$ then

$$|\mathsf{val}(T^{(i-1)}) - \mathsf{val}(T^{(i)})| \leq \epsilon_{\mathsf{PRG}} \cdot \mathsf{k} \ .$$

Letting $\mathcal{C} := \cup_i \mathcal{C}_i$, we have that if G fools $\mathcal{C}$ (i.e., fools circuits of size $\max_{C \in \mathcal{C}} |C| = \mathrm{poly}(|\mathbb{x}|, \mathsf{k}, \mathsf{l}, \mathsf{r})$), then

$$|\mathsf{val}(T_{\mathbb{x}}) - \mathsf{val}(T_{\mathbb{x},\mathsf{G}})| \leq \epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^2 \ .$$

Fix some i. We show a family $\mathcal{C}^{(i)}$ such that if G fools $\mathcal{C}^{(i)}$ then $|\mathsf{val}(T^{(i-1)}) - \mathsf{val}(T^{(i)})| \leq \epsilon_{\mathsf{PRG}} \cdot \mathsf{k}$. Consider a fixed node in $T^{(i)}$ corresponding to the transcript prefix $\mathsf{tr} = (\rho_1, m_1, \ldots, \rho_{i-1}, m_{i-1})$ (which is empty if $i = 1$). For ρ_i let $T^{(i,\mathsf{tr})}(\rho_i)$ be the sub-tree of $T^{(i)}$ whose root corresponds to the transcript $(\mathsf{tr}||\rho_i)$.

Define

$$S := \left\{ \left(1 + \frac{1}{3\mathsf{k}}\right)^{-1}, \ldots, \left(1 + \frac{1}{3\mathsf{k}}\right)^{-O(\mathsf{k})}, 0 \right\} \ .$$

We make the simplifying assumption that $\mathsf{val}(T^{(i,\mathsf{tr})}(\rho_i)) \in S$ and $\mathsf{val}(T^{(i-1,\mathsf{tr})}(\rho_i)) \in S$ for every ρ_i. In the full proof of the claim we achieve this by discretizing the functions $\mathsf{val}(T^{(i,\mathsf{tr})}(\cdot))$ and $\mathsf{val}(T^{(i-1,\mathsf{tr})}(\cdot))$, which incurs additional errors. For simplicity, we ignore these errors in this overview.

For every transcript tr, let $\mathcal{C}^{(i,\mathsf{tr})} := \{C_p^{(i,\mathsf{tr})}\}_{p \in S}$ where each circuit $C_p^{(i,\mathsf{tr})}$, on input ρ_i, outputs 1 if and only if $\mathsf{val}(T^{(i,\mathsf{tr})}(\rho_i)) = p$. We observe that a careful implementation of $C_p^{(i,\mathsf{tr})}$ (computing the value of a tree can be done space proportional to its depth) has size at most $\mathrm{poly}(|\mathbb{x}|, \mathsf{k}, \mathsf{l}, \mathsf{r})$ using PSPACE gates. Thus, if G fools every circuit in the family $\mathcal{C}^{(i,\mathsf{tr})}$ we get that

$$\begin{aligned}
\mathsf{val}(T^{(i-1,\mathsf{tr})}) &= \sum_{p \in S} p \cdot \Pr_s[C_p^{(i,\mathsf{tr})}(\mathsf{G}(s)) = 1] \\
&\leq \sum_{p \in S} p \cdot \left(\Pr_{\rho_i}[C_p^{(i,\mathsf{tr})}(\rho_i) = 1] + \epsilon_{\mathsf{PRG}} \right) \\
&= \mathsf{val}(T^{(i,\mathsf{tr})}) + \sum_{p \in S} p \cdot \epsilon_{\mathsf{PRG}} \\
&\leq \mathsf{val}(T^{(i,\mathsf{tr})}) + O(\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}) \ ,
\end{aligned}$$

where $T^{(i-1,\mathsf{tr})}$ is the sub-tree of $T^{(i-1)}$ whose root corresponds to the transcript tr. The final inequality follows by the fact that $\sum_{p \in S} p = \sum_{i=1}^{O(\mathsf{k})} (1 + 1/3\mathsf{k})^{-i}$ is a geometric series bounded by $O(\mathsf{k})$.

We can similarly show that $\mathsf{val}(T^{(i-1,\mathsf{tr})}) \geq \mathsf{val}(T^{(i,\mathsf{tr})}) - O(\epsilon_{\mathsf{PRG}} \cdot \mathsf{k})$. Notice that $\mathsf{val}(T^{(i)}) = \mathbb{E}_{\mathsf{tr}}[\mathsf{val}(T^{(i,\mathsf{tr})})]$ and $\mathsf{val}(T^{(i-1)}) = \mathbb{E}_{\mathsf{tr}}[\mathsf{val}(T^{(i-1,\mathsf{tr})})]$ (where the expectation is over the verifier's random coins). Therefore, if the G fools the entire circuit family $\mathcal{C}^{(i)} := \cup_{\mathsf{tr}} \mathcal{C}^{(i,\mathsf{tr})}$ then we have

$$\begin{aligned}
|\mathsf{val}(T^{(i-1)}) - \mathsf{val}(T^{(i)})| &= |\mathbb{E}_{\mathsf{tr}}[\mathsf{val}(T^{(i-1,\mathsf{tr})})] - \mathbb{E}_{\mathsf{tr}}[\mathsf{val}(T^{(i,\mathsf{tr})})]| \\
&\leq \left| \mathbb{E}_{\mathsf{tr}} \left[\mathsf{val}(T^{(i,\mathsf{tr})}) + O(\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}) \right] - \mathbb{E}_{\mathsf{tr}} \left[\mathsf{val}(T^{(i,\mathsf{tr})}) \right] \right| \\
&= O(\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}) \ .
\end{aligned}$$

2.4 Deriving our results using the tools

We use the toolbox developed in the previous sections to derive the theorems in Section 1.1. Each theorem is proved by applying a carefully chosen sequence of tools (along with other arguments). Figure 1 summarizes which tools are used to derive each theorem and the order of their use.

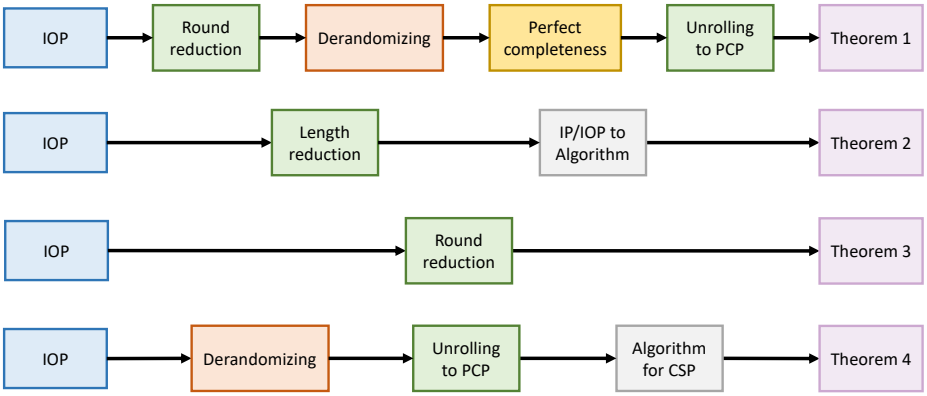

Fig. 1. Summary of how our tools are used to derive each theorem. The "IP/IOP to algorithm" and "Algorithm for CSP" boxes are due to prior work.

Low-error IOPs to low-error PCPs We sketch the proof of Theorem 1, which shows that low-error IOPs can be transformed into low-error PCPs. The proof is a sequence of transformations from our toolbox, whose goal is to transform the IOP into one that is efficient enough to be unrolled into a PCP via Item 3 of Lemma 1. This unrolling has an exponential dependency on the round complexity and on the verifier randomness complexity of the IOP, so we seek to decrease these without increasing the soundness error.

Decreasing the round complexity is done using the round-reduction transformation of Lemma 1, and decreasing the verifier randomness is done using either one of our derandomization lemmas (Lemma 3). Since both transformations degrade completeness, prior to applying the unrolling lemma (Item 3 of Lemma 1), we restore the IOP back to having perfect completeness using Lemma 2. Since the transformation for perfect completeness increases the soundness error, we counterbalance it by beginning the sequence of transformations with a small number of parallel repetitions.

In somewhat more detail, the sequence of transformations is as follows.

1. **Initial IOP.** We begin with an IOP with the following parameters: perfect completeness, soundness error $1/|\mathbb{x}|$, round complexity $\text{polylog}(|\mathbb{x}|)$, alphabet size $\text{poly}(|\mathbb{x}|)$, proof length $\text{poly}(|\mathbb{x}|)$, query complexity $O(1)$, and per-round randomness $\text{poly}(|\mathbb{x}|)$.

2. **Parallel repetition.** Repeat the protocol twice in parallel, and have the verifier accept if and only if both executions are accepted. This yields a public-coin IOP for R with: perfect completeness, **soundness error $1/|\mathbb{x}|^2$**, round complexity $\mathsf{k} = \text{polylog}(|\mathbb{x}|)$, alphabet size $\text{poly}(|\mathbb{x}|)$, query complexity $\mathsf{q} = O(1)$, and per-round randomness $\text{poly}(|\mathbb{x}|)$.
3. **Round reduction.** Reduce the number of rounds of the IOP via Item 2 of Lemma 1 with $\ell := \mathsf{q}$ where $\mathsf{q} = O(1)$ is the query complexity of the IOP verifier. This transformation results in a public-coin IOP for R with: completeness error $1-(\mathsf{q}/(e\cdot\mathsf{k}))^{\mathsf{q}} = 1-1/\text{polylog}(|\mathbb{x}|)$, soundness error $1/|\mathbb{x}|^2$, **round complexity $O(1)$**, alphabet size $\text{poly}(|\mathbb{x}|)$, query complexity $O(1)$, and per-round randomness $\text{poly}(|\mathbb{x}|)$.
4. **Derandomization.** Derandomize the IOP verifier using either item of Lemma 3. This results in a public-coin IOP for R with: completeness error $1-1/\text{polylog}(|\mathbb{x}|)$, soundness error $O(1/|\mathbb{x}|^2)$, round complexity $O(1)$, alphabet size $\text{poly}(|\mathbb{x}|)$, query complexity $O(1)$, and **per-round randomness $O(\log|\mathbb{x}|)$**.
5. **Perfect completeness.** Improve the IOP to have perfect completeness using Lemma 2. The resulting IOP has the following parameters: **perfect completeness**, soundness error

$$O(1/|\mathbb{x}|^2)\cdot\left(\frac{\mathsf{q}\cdot O(\log|\mathbb{x}|)}{-\log(1-1/\text{polylog}(|\mathbb{x}|))}\right) \le 1/|\mathbb{x}| \ ,$$

   round complexity $O(1)$, alphabet size $\text{poly}(|\mathbb{x}|)$, query complexity $O(1)$, and randomness $O(\log|\mathbb{x}|)$.
6. **Unrolling to PCP.** Unroll the IOP with perfect completeness into a PCP via Item 3 of Lemma 1. This gives us our final PCP with parameters: perfect completeness, soundness error $1/|\mathbb{x}|$, alphabet size $\text{poly}(|\mathbb{x}|)$, proof length $\text{poly}(|\mathbb{x}|)$, query complexity $O(1)$, and randomness complexity $O(\log|\mathbb{x}|)$.

Limitations of short IOPs We sketch the proof of Theorem 2, which shows that short IOPs with small soundness contradict RETH, the hypothesis that 3SAT $\notin$ BPTIME$[2^{c\cdot n}]$ for a constant $c > 0$. First, we convert the IOP into a short IP, and then apply a transformation from [18] that converts short IPs into fast probabilistic algorithms. This leads to a fast algorithm for 3SAT, contradicting RETH.

Consider a public-coin IOP for n-variate 3SAT with parameters as in Theorem 2: perfect completeness, soundness error β, round complexity $\text{polylog}(n)$, alphabet size λ, (total) proof length l, query complexity q, and verifier randomness $\text{poly}(n)$. Suppose towards contradiction that $\mathsf{l} \ge n$ and $\left(\frac{\mathsf{l}\cdot\log\lambda}{n}\right)^{\mathsf{q}} \le n^{\text{polylog}(n)}$ and that $\beta = \frac{1}{2}\cdot\left(\frac{2\cdot e\cdot\mathsf{l}\cdot\log\lambda}{c\cdot n}\right)^{-\mathsf{q}} \ge n^{-\text{polylog}(n)}$.[8]

We apply the following transformations.

[8] It is sufficient to assume that $\beta = \frac{1}{2}\cdot\left(\frac{2\cdot e\cdot\mathsf{l}\cdot\log\lambda}{c\cdot n}\right)^{-\mathsf{q}}$ to find contradiction in $\beta \le \frac{1}{2}\cdot\left(\frac{2\cdot e\cdot\mathsf{l}\cdot\log\lambda}{c\cdot n}\right)^{-\mathsf{q}}$ since we can always increase the soundness error without loss of generality.

1. **Length reduction.** Apply Item 1 of Lemma 1 with parameter $\ell := e \cdot \mathsf{l} \cdot (2\beta)^{1/\mathsf{q}}$. This results in an IOP with: completeness error $\alpha' := 1 - 2\beta$, soundness error β, round complexity $\mathsf{k}' := \text{polylog}(n)$, alphabet size $\lambda' := \lambda$, and **proof length $\mathsf{l}' := e \cdot \mathsf{l} \cdot (2\beta)^{1/\mathsf{q}}$**.
2. **IOP to algorithm.** Convert the IOP into an algorithm using a lemma from [18] that says that if a relation R has a public-coin IP with completeness error α', soundness error β', round complexity k', and prover-to-verifier communication length l' of symbols of size λ', then there is a probabilistic algorithm for deciding R in time $2^{O(d)+o(n)}$ for $d := \mathsf{l}' \cdot \log \lambda' + \mathsf{k}' \cdot \log \frac{\mathsf{k}'}{1-\alpha'-\beta'}$. Notice that while the result from [18] applies to IPs rather than IOPs, one can straightforwardly convert an IOP into an IP by having the verifier read the prover's messages in their entirety.

Substituting the relevant parameters, we have that:

$$\begin{aligned} d &= \mathsf{l}' \cdot \log \lambda' + \mathsf{k}' \cdot \log \frac{\mathsf{k}'}{1-\alpha'-\beta'} \\ &= e \cdot \mathsf{l} \cdot (2\beta)^{1/\mathsf{q}} \cdot \log \lambda + \mathsf{k} \cdot \log(\mathsf{k}/\beta) \\ &= c \cdot n/2 + \text{polylog}(n) \ . \end{aligned}$$

Thus, 3SAT is decidable in probabilistic time $2^{c \cdot n/2 + o(n)} < 2^{c \cdot n}$ in contradiction to RETH.

Limitations of high-round low-query IOPs We sketch the proof of Theorem 3, showing that relations not decidable in few rounds do not have small-query IOPs with good soundness error. As in the theorem statement, let $R \in \text{AM}[\mathsf{k}] \backslash \text{AM}[\mathsf{k}']$ be a relation for $\mathsf{k}' < \mathsf{k}$ and suppose that R has a k-round public-coin IOP $(\mathbf{P}, \mathbf{V})$ with perfect completeness, soundness error β, alphabet size $2^{\text{poly}(|\mathbb{x}|)}$, proof length $\text{poly}(|\mathbb{x}|)$, and query complexity $\mathsf{q} \leq \mathsf{k}'$.

By applying the round-reduction lemma (Item 2 of Lemma 1) to the k-round IOP $(\mathbf{P}, \mathbf{V})$ with parameter k', we get a k'-round IOP $(\mathbf{P}', \mathbf{V}')$ with completeness error $\alpha' := 1 - (\mathsf{k}'/(e \cdot \mathsf{k}))^{\mathsf{q}}$ and soundness error β. Suppose towards contradiction that $\beta < (\mathsf{k}'/(e \cdot \mathsf{k}))^{\mathsf{q}} - |\mathbb{x}|^{-c}$ for some $c \in \mathbb{N}$. Then the (additive) gap between completeness and soundness error of $(\mathbf{P}', \mathbf{V}')$ is $1 - \alpha' - \beta > |\mathbb{x}|^{-c}$.

Since the gap between completeness and soundness error of $(\mathbf{P}', \mathbf{V}')$ is inverse polynomial, it can be transformed into a k'-round public-coin IP $(\mathbf{P}''_{\mathsf{IP}}, \mathbf{V}''_{\mathsf{IP}})$ for R with completeness error $1/3$ and soundness error $1/3$. This is done by using the standard technique of taking $\text{poly}(|\mathbb{x}|)$ parallel repetitions, computing the fraction of accepting transcripts, and accepting if the number of accepting transcripts is beyond some threshold that depends on α' and $|\mathbb{x}|^{-c}$. The IP $(\mathbf{P}''_{\mathsf{IP}}, \mathbf{V}''_{\mathsf{IP}})$, then, contradicts the assumption that $R \notin \text{AM}[\mathsf{k}']$.

Limitations of binary-alphabet constant-query IOPs We sketch the proof of Theorem 4, showing that assuming RETH there are no binary-alphabet IOPs

with 2 or 3 queries and small soundness error for 3SAT. We first discuss the following lemma which says that, assuming RETH, algorithms for solving constraint satisfaction problems (CSPs) cannot coexist with IOPs with a binary alphabet, constant query complexity, and small soundness error.

Lemma 4 (informal)**.** *Assume* RETH *and suppose that both of the following exist.*
- *An IOP with perfect completeness, soundness error β, round complexity k, alphabet size 2, proof length $2^{o(n)}$, query complexity q, verifier randomness r, and verifier running time $2^{o(n)}$.*
- *A polynomial-time algorithm $\mathbf{A}$ for deciding whether a binary-alphabet CSP with arity q has value 1 or value at most γ.*

Then $\beta > \gamma - \varepsilon$ for every ε satisfying $\mathsf{k} \cdot \log(\mathsf{r} \cdot n/\varepsilon) = o(n)$.

The proof of the theorem is concluded by relying on known algorithms for solving CSPs with appropriate arities q and decision bounds γ.

- For $\mathsf{q} = 2$, we rely on Schaefer's dichotomy theorem [37], which says that the satisfiability of a binary-alphabet CSP with arity 2 can be decided in polynomial time. In this case $\gamma = 1$.
- For $\mathsf{q} = 3$, we rely on Zwick's algorithm [39], which decides in polynomial time whether a binary-alphabet CSP with arity 3 has value 1 or value smaller than 5/8. In this case $\gamma = 5/8$.

Proof sketch of Lemma 4. Suppose towards contradiction that $\beta \leq \gamma - \varepsilon$ where ε satisfies $\mathsf{k} \cdot \log(\mathsf{r} \cdot n/\varepsilon) = o(n)$. The proof has two steps: (1) transform the IOP into a PCP for 3SAT that is "efficient-enough"; and (2) use the "efficient-enough" PCP and the algorithm $\mathbf{A}$ to decide 3SAT.

IOP to "efficient-enough" PCP. We apply these transformations from our toolbox.

1. **Derandomization using non-uniform advice.** Reduce the verifier randomness of the IOP using the non-uniform derandomization theorem (Lemma 3, Item 2) with error ε/k to get per-round randomness complexity of $O(\log(\mathsf{r} \cdot n/\varepsilon))$ bits. The new IOP uses $\mathrm{poly}(n, \mathsf{r}, 1/\varepsilon)$ bits of non-uniform advice, where a random string is good advice with overwhelming probability. The resulting IOP has perfect completeness, **soundness error $\boldsymbol{\beta+\varepsilon \leq \gamma}$**, round complexity k, alphabet size 2, proof length $2^{o(n)}$, query complexity q, **verifier randomness $\boldsymbol{O(\log(\mathsf{r} \cdot n/\varepsilon))}$**, and verifier running time $2^{o(n)} + \mathrm{poly}(n, \mathsf{r}, 1/\varepsilon) = 2^{o(n)}$.
2. **Unrolling to PCP.** Unroll the IOP into a PCP for 3SAT using Lemma 1, Item 3. This transformation preserves the number of advice bits, and also the fact that a random string is good advice with overwhelming probability. The resulting PCP has perfect completeness, soundness error γ, alphabet size 2, proof length $2^{O(\mathsf{k} \cdot \log(\mathsf{k} \cdot n/\varepsilon)) + o(n)} = 2^{o(n)}$, query complexity q, randomness complexity $O(\log(\mathsf{k} \cdot n/\varepsilon)) = o(n)$, and verifier running time $2^{o(n)}$.

Solving 3SAT *using the PCP and CSP solvers.* We use the PCP and the algorithm $\mathbf{A}$ to design a probabilistic algorithm $\mathbf{A}'$ that decides whether a 3SAT formula ϕ over n variables is satisfiable in time $2^{o(n)}$. The algorithm $\mathbf{A}'$, on input the 3SAT formula ϕ, works as follows.

1. **Sample random advice.** Sample a random advice string z for the PCP resulting from the previous transformation.
2. **Transform formula to CSP.** Transform the 3SAT formula ϕ into a binary-alphabet CSP ψ with arity q. This is done using the standard method of translating a PCP into a CSP; each constraint in the CSP is indexed by a choice of verifier randomness ρ and described by the verifier circuit with the input formula ϕ, randomness ρ, and advice z hard-coded. The CSP ψ has size $\text{poly}(2^{\mathsf{r}'}, \mathsf{vt}') = 2^{o(n)}$ where $\mathsf{r}' = o(n)$ and $\mathsf{vt}' = 2^{o(n)}$ are the randomness complexity and verifier running time of the PCP. Additionally, assuming that z is good advice, we have that if $\phi \in$ 3SAT then the value of ψ is 1, and if $\phi \notin$ 3SAT, then the value of ϕ is at most γ.
3. **Solve CSP.** Run $\mathbf{A}(\psi)$ and say that ϕ is satisfiable if and only if $\mathbf{A}$ says that ψ's value is 1.

The algorithm $\mathbf{A}'$ decides 3SAT with high probability: with overwhelming probability the choice of advice z is good, and deciding whether the value of the CSP instance ψ is 1 or γ, as $\mathbf{A}$ does, is equivalent to deciding whether ϕ is satisfiable.

Moreover, the algorithm $\mathbf{A}'$ runs in probabilistic time $2^{o(n)}$: the advice sampled in the first step is polynomial; the second step can be done in time $\text{poly}(2^{\mathsf{r}'}, \mathsf{vt}') = 2^{o(n)}$ where $\mathsf{r}' = o(n)$ and $\mathsf{vt}' = 2^{o(n)}$ are the randomness complexity and verifier running time of the PCP; the final step takes $\text{poly}(|\psi|) = 2^{o(n)}$, since $\mathbf{A}$ runs in polynomial time.

We obtained an algorithm for deciding 3SAT in probabilistic time $2^{o(n)}$, contradicting RETH. □

References

1. Applebaum, B., Golombek, E.: On the randomness complexity of interactive proofs and statistical zero-knowledge proofs. In: Proceedings of the 2nd Conference on Information-Theoretic Cryptography. pp. 4:1–4:23. ITC '21 (2021)
2. Arnon, G., Chiesa, A., Yogev, E.: Hardness of approximation for stochastic problems via interactive oracle proofs. In: Proceedings of the 37th Annual IEEE Conference on Computational Complexity. pp. 24:1–24:16. CCC '22 (2022)
3. Arnon, G., Chiesa, A., Yogev, E.: A PCP theorem for interactive proofs. In: Proceedings of the 41st Annual International Conference on Theory and Application of Cryptographic Techniques. pp. 64–94. EUROCRYPT '22 (2022)
4. Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M.: Proof verification and the hardness of approximation problems. Journal of the ACM **45**(3), 501–555 (1998), preliminary version in FOCS '92.
5. Arora, S., Safra, S.: Probabilistic checking of proofs: a new characterization of NP. Journal of the ACM **45**(1), 70–122 (1998), preliminary version in FOCS '92.

6. Babai, L., Fortnow, L., Levin, L.A., Szegedy, M.: Checking computations in polylogarithmic time. In: Proceedings of the 23rd Annual ACM Symposium on Theory of Computing. pp. 21–32. STOC '91 (1991)
7. Bellare, M., Goldwasser, S., Lund, C., Russell, A.: Efficient probabilistically checkable proofs and applications to approximations. In: Proceedings of the 25th Annual ACM Symposium on Theory of Computing. pp. 294–304. STOC ?93 (1993)
8. Ben-Sasson, E., Bentov, I., Chiesa, A., Gabizon, A., Genkin, D., Hamilis, M., Pergament, E., Riabzev, M., Silberstein, M., Tromer, E., Virza, M.: Computational integrity with a public random string from quasi-linear pcps. In: Proceedings of the 36th Annual International Conference on Theory and Application of Cryptographic Techniques. pp. 551–579. EUROCRYPT '17 (2017)
9. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Fast Reed–Solomon interactive oracle proofs of proximity. In: Proceedings of the 45th International Colloquium on Automata, Languages and Programming. pp. 14:1–14:17. ICALP '18 (2018)
10. Ben-Sasson, E., Chiesa, A., Gabizon, A., Riabzev, M., Spooner, N.: Interactive oracle proofs with constant rate and query complexity. In: Proceedings of the 44th International Colloquium on Automata, Languages and Programming. pp. 40:1–40:15. ICALP '17 (2017)
11. Ben-Sasson, E., Chiesa, A., Gabizon, A., Virza, M.: Quasilinear-size zero knowledge from linear-algebraic PCPs. In: Proceedings of the 13th Theory of Cryptography Conference. pp. 33–64. TCC '16-A (2016)
12. Ben-Sasson, E., Chiesa, A., Spooner, N.: Interactive oracle proofs. In: Proceedings of the 14th Theory of Cryptography Conference. pp. 31–60. TCC '16-B (2016)
13. Ben-Sasson, E., Sudan, M.: Short PCPs with polylog query complexity. SIAM Journal on Computing **38**(2), 551–607 (2008), preliminary version appeared in STOC '05.
14. Bootle, J., Cerulli, A., Ghadafi, E., Groth, J., Hajiabadi, M., Jakobsen, S.K.: Linear-time zero-knowledge proofs for arithmetic circuit satisfiability. In: Proceedings of the 23rd International Conference on the Theory and Applications of Cryptology and Information Security. pp. 336–365. ASIACRYPT '17 (2017)
15. Bootle, J., Chiesa, A., Groth, J.: Linear-time arguments with sublinear verification from tensor codes. In: Proceedings of the 18th Theory of Cryptography Conference. pp. 19–46. TCC '20 (2020)
16. Bootle, J., Chiesa, A., Liu, S.: Zero-knowledge IOPs with linear-time prover and polylogarithmic-time verifier. In: Proceedings of the 41st Annual International Conference on Theory and Application of Cryptographic Techniques. pp. 275–304. EUROCRYPT '22 (2022)
17. Bordage, S., Nardi, J.: Interactive oracle proofs of proximity to algebraic geometry codes. In: Proceedings of the 37th Annual IEEE Conference on Computational Complexity. pp. 30:1–30:45. CCC '22 (2022)
18. Chiesa, A., Yogev, E.: Barriers for succinct arguments in the random oracle model. In: Proceedings of the 18th Theory of Cryptography Conference. pp. 47–76. TCC '20 (2020)
19. Condon, A., Feigenbaum, J., Lund, C., Shor, P.W.: Random debaters and the hardness of approximating stochastic functions. SIAM Journal on Computing **26**(2), 369–400 (1997)
20. Dinur, I.: The PCP theorem by gap amplification. Journal of the ACM **54**(3), 12 (2007)
21. Dinur, I., Harsha, P., Kindler, G.: Polynomially low error PCPs with polyloglog n queries via modular composition. In: Proceedings of the 47th Annual ACM Symposium on Theory of Computing. pp. 267–276. STOC '15 (2015)

22. Drucker, A.: A PCP characterization of AM. In: Proceedings of the 38th International Colloquium on Automata, Languages and Programming. pp. 581–592. ICALP '11 (2011)
23. Feige, U., Goldwasser, S., Lovász, L., Safra, S., Szegedy, M.: Approximating clique is almost NP-complete (preliminary version). In: Proceedings of the 32nd Annual Symposium on Foundations of Computer Science. pp. 2–12. SFCS '91 (1991)
24. Feige, U., Goldwasser, S., Lovász, L., Safra, S., Szegedy, M.: Interactive proofs and the hardness of approximating cliques. Journal of the ACM **43**(2), 268–292 (1996), preliminary version in FOCS '91.
25. Fürer, M., Goldreich, O., Mansour, Y., Sipser, M., Zachos, S.: On completeness and soundness in interactive proof systems. Advances in Computing Research **5**, 429–442 (1989)
26. Goldreich, O., Håstad, J.: On the complexity of interactive proofs with bounded communication. Information Processing Letters **67**(4), 205–214 (1998)
27. Goldreich, O., Vadhan, S., Wigderson, A.: On interactive proofs with a laconic prover. Computational Complexity **11**(1/2), 1–53 (2002)
28. Golovnev, A., Lee, J., V., S.S.T., Thaler, J., Wahby, R.S.: Brakedown: Linear-time and post-quantum snarks for R1CS. Cryptology ePrint Archive, Report 2021/1043 (2021)
29. Hast, G.: Beating a random assignment: Approximating constraint satisfaction problems. Ph.D. thesis, KTH (2005)
30. Håstad, J.: On the NP-hardness of Max-Not-2. SIAM Journal on Computing **43**(1), 179–193 (2014)
31. Lee, J., Setty, S.T.V., Thaler, J., Wahby, R.S.: Linear-time zero-knowledge snarks for R1CS. Cryptology ePrint Archive, Report 2021/30 (2021)
32. Manurangsi, P., Nakkiran, P., Trevisan, L.: Near-optimal NP-hardness of approximating MAX k-CSPR. Theory of Computing **18**(3), 1–29 (2022)
33. Nassar, S., Rothblum, R.D.: Succinct interactive oracle proofs: Applications and limitations. In: Proceedings of the 42nd Annual International Cryptology Conference. CRYPTO '22 (2022)
34. Reingold, O., Rothblum, R., Rothblum, G.: Constant-round interactive proofs for delegating computation. In: Proceedings of the 48th ACM Symposium on the Theory of Computing. pp. 49–62. STOC '16 (2016)
35. Ron-Zewi, N., Rothblum, R.: Local proofs approaching the witness length. In: Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science. pp. 846–857. FOCS '20 (2020)
36. Ron-Zewi, N., Rothblum, R.D.: Proving as fast as computing: Succinct arguments with constant prover overhead. In: Proceedings of the 54th ACM Symposium on the Theory of Computing. pp. 1353–1363. STOC '22 (2022)
37. Schaefer, T.J.: The complexity of satisfiability problems. In: Proceedings of the 10th Annual ACM Symposium on Theory of Computing. pp. 216–226. STOC '78 (1978)
38. Xie, T., Zhang, J., Zhang, Y., Papamanthou, C., Song, D.: Libra: Succinct zero-knowledge proofs with optimal prover computation. In: Proceedings of the 39th Annual International Cryptology Conference. pp. 733–764. CRYPTO '19 (2019)
39. Zwick, U.: Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In: Proceedings of the 9th Annual Symposium on Discrete Algorithms. pp. 201–210. SODA '98 (1998)