A Toolbox for Barriers on Interactive Oracle Proofs

Gal Arnon
gal.arnon@weizmann.ac.il
Weizmann Institute

Amey Bhangale
amey.bhangale@ucr.edu
UC Riverside

Alessandro Chiesa
alessandro.chiesa@epfl.ch
EPFL

Eylon Yogev
eylon.yogev@biu.ac.il
Bar-Ilan University

November 24, 2022

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

Contents

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) [BFLS91; FGLSS96] 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 [AS98; ALMSS98]: 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 [BS08; Din07], 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) [BCS16; RRR16]. 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 [BCGV16; Ben+17; BCGRS17; BBHR18; BCGGHJ17; XZZPS19; BCG20; BCL22; RR20; LSTW21; GLVTW21; BN22; RR22]. 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 [CFLS97; Dru11; ACY22a; ACY22b].

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 [BGLR93] states that for every β with $1/\mathrm{poly}(n) \leq \beta < 1$ there is a PCP system for NP that has perfect completeness, soundness error β, polynomial proof length over a $\mathrm{poly}(1/\beta)$-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* $\mathrm{polylog}(n)$*, alphabet size* $\mathrm{poly}(n)$*, proof length* $\mathrm{poly}(n)$*, and query complexity $O(1)$. Then under a derandomization assumption[1] (or alternatively by using a non-uniform verifier) R has a PCP with perfect completeness, soundness error $1/n$, alphabet size* $\mathrm{poly}(n)$*, proof length* $\mathrm{poly}(n)$*, and query complexity $O(1)$.*

Our full theorem in Section 7 allows for trade-offs 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 $\mathrm{polylog}(n)$-round IOPs with sliding-scale parameters is as hard as constructing sliding-scale PCPs.

Our theorem does leave open the question of constructing $\mathrm{poly}(n)$-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., [BCGRS17]). 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),[2] short IOPs for 3SAT have high soundness error.

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 [DHK15] 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 [DHK15] (even when significantly increasing alphabet size).

Corollary 1.1. *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 [GVW02] show that $\mathrm{IP}[\mathsf{k}] \neq \mathrm{IP}[o(\mathsf{k})]$ for every k, under reasonable complexity assumptions. In other words,

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

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

IPs with k rounds cannot be "compressed" to have $o(\mathsf{k})$ rounds. In contrast, Arnon, Chiesa, and Yogev [ACY22b] 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 [ACY22b]. 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 [ACY22b]), 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 [Zwi98] 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.[3] Håstad [Hås14] 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 $\mathsf{k} = \mathrm{polylog}(n)$, 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

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

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 [Has05].

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 [FGLSS91], given 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 [MNT22]. Assuming ETH, these limitations can be extended to PCPs with super-polynomial proof length and super-constant alphabet size and query complexity. For completeness, in Section 10.1 we show quantitatively 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.* [GH98] show that public-coin IPs with bounded prover communication complexity can be decided in non-trivial (probabilistic) time. [GVW02] 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, [CY20] extend to IOPs the limitations of [GH98], showing barriers for IOPs with small soundness error relative to query complexity.

  [NR22] 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 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.

- [ACY22a; ACY22b] 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).
- [NR22] 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.
- [AG21] 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. 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. Details are in Section 4.
2. **Tools for improving completeness:** Section 2.2 outlines transformations that improve the completeness errors of IOPs. Details are in Section 5.
3. **Tools for derandomization:** Section 2.3 outlines transformations that decrease the number of random bits used by the IOP verifier. Details are in Section 6.

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 $\boldsymbol{\ell}$**, 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 $\boldsymbol{\mathsf{k}'+1}$**, 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 2.1 (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 2.2 (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 [FGMSZ89]. Directly applying that transformation to IOPs increases the query complexity of the protocol significantly. We show a variant of the transformation in [FGMSZ89] 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 2.3. 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 Section 5.1 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 $\vec{\rho} \in S$. [FGMSZ89] 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 2.4. 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 $\mathsf{t_{PRG}}$. 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 $\boldsymbol{\ell}$**, and verifier running time* $\mathrm{poly}(\mathsf{vt}, \mathsf{t_{PRG}})$.

   (Such a PRG with seed length $\ell = O(\log |\mathbb{x}|)$, error $\varepsilon = 1/\mathrm{poly}(|\mathbb{x}|)$ and computation time $\mathsf{t_{PRG}} = \mathrm{poly}(|\mathbb{x}|)$ 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 $\boldsymbol{\Theta(\log((\mathsf{r} \cdot \mathsf{k} + |\mathbb{x}|)/\varepsilon))}$**, 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 pseudorandom 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 2.5 (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 Claim 2.6, 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 2.6. *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.

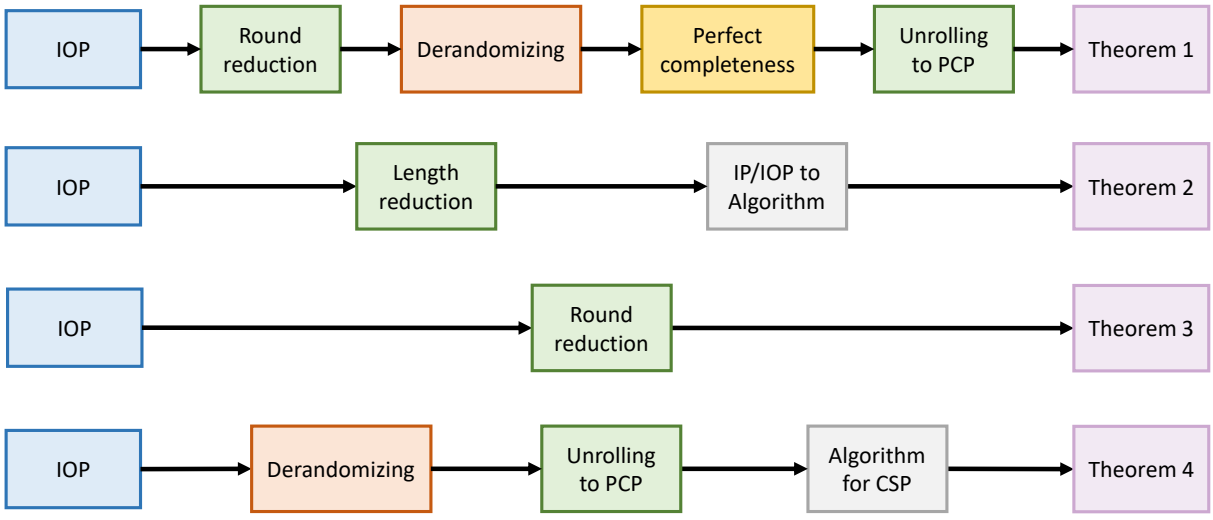

Figure 1: Summary of how our tools are used to derive each theorem. Grey boxes are prior work.

2.4.1 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

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

   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 $O(\log |\mathbb{x}|)$.

2.4.2 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 [CY20] 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 polylog(n), alphabet size λ, (total) proof length l, query complexity q, and verifier randomness poly(n). Suppose towards contradiction that $\mathsf{l} \geq n$ and $\left(\frac{\mathsf{l}\cdot\log\lambda}{n}\right)^{\mathsf{q}} \leq n^{\text{polylog}(n)}$ and that $\beta = \frac{1}{2}\cdot\left(\frac{2\cdot e\cdot\mathsf{l}\cdot\log\lambda}{c\cdot n}\right)^{-\mathsf{q}} \geq n^{-\text{polylog}(n)}$.[4]

We apply the following transformations.

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 $\mathsf{k}' :=$ 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 [CY20] 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 [CY20] 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.

2.4.3 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 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 $1/3$

[4] 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 $\beta \leq \frac{1}{2}\cdot\left(\frac{2\cdot e\cdot\mathsf{l}\cdot\log\lambda}{c\cdot n}\right)^{-\mathsf{q}}$ since we can always increase the soundness error without loss of generality.

and soundness error 1/3. This is done by using the standard technique of taking 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 $R \notin \mathrm{AM}[\mathsf{k}']$.

2.4.4 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 [Sch78], 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 [Zwi98], 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. □

3 Preliminaries

In some theorems we highlight important parameters with a yellow background.

3.1 Interactive oracle proofs

Interactive Oracle Proofs (IOPs) [BCS16; RRR16] are information-theoretic proof systems that combine aspects of Interactive Proofs [Bab85; GMR89] and Probabilistically Checkable Proofs [BFLS91; FGLSS91; AS98; ALMSS98], and also generalize the notion of Interactive PCPs [KR08]. Below we describe *public-coin* IOPs.

Recall that a k-round public-coin IOP works as follows. In every round $i \in [\mathsf{k}]$, the verifier sends a uniformly random message ρ_i to the prover; then the prover sends a proof string π_i to the verifier. After k rounds of interaction, the verifier makes some queries to the proof strings $\pi_1, \ldots, \pi_{\mathsf{k}}$ sent by the prover, and then decides if to accept or to reject.

In more detail, let $\mathsf{IOP} = (\mathbf{P}, \mathbf{V})$ be a tuple where $\mathbf{P}$ is an interactive algorithm, and $\mathbf{V}$ is an interactive oracle algorithm. We say that IOP is a *public-coin IOP* for an indexed relation R with k rounds, completeness error α, and soundness error β if the following holds.

- **Completeness.** For every $(\mathbb{x}, \mathbb{w}) \in R$,

$$\Pr_{\rho_1,\ldots,\rho_{\mathsf{k}}}\left[\mathbf{V}^{\pi_1,\ldots,\pi_{\mathsf{k}}}(\mathbb{x}, \rho_1, \ldots, \rho_{\mathsf{k}}) = 1 \;\middle|\; \begin{array}{r} \pi_1 \leftarrow \mathbf{P}(\mathbb{x}, \mathbb{w}, \rho_1) \\ \vdots \\ \pi_{\mathsf{k}} \leftarrow \mathbf{P}(\mathbb{x}, \mathbb{w}, \rho_1, \ldots, \rho_{\mathsf{k}}) \end{array}\right] \geq 1 - \alpha \ .$$

- **Soundness.** For every $\mathbb{x} \notin L(R)$ and unbounded malicious prover $\tilde{\mathbf{P}}$,

$$\Pr_{\rho_1,\ldots,\rho_{\mathsf{k}}}\left[\mathbf{V}^{\pi_1,\ldots,\pi_{\mathsf{k}}}(\mathbb{x}, \rho_1, \ldots, \rho_{\mathsf{k}}) = 1 \;\middle|\; \begin{array}{r} \pi_1 \leftarrow \tilde{\mathbf{P}}(\rho_1) \\ \vdots \\ \pi_{\mathsf{k}} \leftarrow \tilde{\mathbf{P}}(\rho_1, \ldots, \rho_{\mathsf{k}}) \end{array}\right] \leq \beta \ .$$

For interactive (oracle) algorithms $\mathbf{A}$ and $\mathbf{B}$, we denote by $\langle \mathbf{A}(a), \mathbf{B}(b) \rangle(c)$ the random variable describing the output of $\mathbf{B}$ following the interaction between $\mathbf{A}$ and $\mathbf{B}$, where $\mathbf{A}$ is given private input a, $\mathbf{B}$ is given private input b and both parties are given joint input c.

Efficiency measures. We study several efficiency measures. All of these complexity measures are implicitly functions of the instance $\mathbb{x}$.

- *Rounds* k: The IOP has k rounds of interaction.
- *Alphabet* Σ *and alphabet size* λ: the symbols of each π_i come from the alphabet Σ, of size λ.
- *Proof length (per round)* l: the number of bits in each proof π_i.
- *Queries* q: the number of bits read by the verifier from $\pi_1, \ldots, \pi_{\mathsf{k}}$.
- *Randomness (per round)* r: the number of bits in each verifier message ρ_i.
- *Verifier time* vt: $\mathbf{V}$ runs in time vt.
- *Decision complexity* dt: Following the choice of queries, $\mathbf{V}$ runs in time dt to decide whether to accept or reject.

PCPs, Round-query IOPs and IPs. A PCP is an IOP with no interaction rounds (only a single prover message). A round-query IOP with round-query complexity $\mathsf{q}_{\mathsf{round}}$ is an IOP in which the verifier reads symbols from at most $\mathsf{q}_{\mathsf{round}}$ rounds of the interaction. An interactive proof (IP) is an IOP where the verifier reads every symbol in the interaction.

3.2 Interaction trees of probabilistic proofs

An alternate way of viewing a probabilistic proof is that of an *interaction tree* that describes all possible conversations between a prover and verifier:

Definition 3.1. *The (complete)* **interaction tree** *of an IOP* $(\mathbf{P}, \mathbf{V})$ *on an instance* $\mathbb{x}$*, denoted by* $T_{\mathbb{x}}$*, is defined as follows:*

- *The root represents the empty transcript.*
- *Each node* v *represents a transcript* tr*, and moving from node* v *to a child* u *via edge* e *corresponds to adding the message* e *to the transcript* tr*. Thus, the node* u *represents the transcript* $\mathsf{tr}' = (\mathsf{tr}||e)$*.*
- *Each leaf* v*, representing a complete transcript* tr*, is labelled with "accept" or "reject" matching whether* $\mathbf{V}$ *accepts* tr *given instance* $\mathbb{x}$*.*

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, if $\mathbf{V}$ *runs in polynomial time, then* $\mathsf{val}(T_{\mathbb{x}})$ *can be computed in space that is polynomial in* $|\mathbb{x}|$ *and in the round complexity, the proof length, and the verifier randomness of the IOP. This can be done using the standard* PSPACE *algorithm for* IP*.*

3.3 IP to algorithm

Goldreich and Håstad [GH98] show that a public-coin IP with constant completeness and soundness errors for a relation implies a probabilistic algorithm that decides the same relation. Chiesa and Yogev [CY20] refine this result to apply to arbitrary completeness and soundness errors.

Lemma 3.2 ([GH98; CY20])**.** *Suppose that a relation* R *has a public-coin IP with completeness error* α*, soundness error* β*, round complexity* k*, and (binary) prover-to-verifier communication* l*. For* $d(n) := \mathsf{l}(n) + \mathsf{k}(n) \cdot \log \frac{\mathsf{k}(n)}{1-\alpha(n)-\beta(n)}$*, the relation* R *is in*

$$\mathrm{BPTIME}\left[2^{O(d)} \cdot \mathrm{poly}(n)\right] \ .$$

3.4 Constraint satisfaction problems

We introduce notation and facts for constraint satisfaction problems (CSPs). We denote by Σ a finite alphabet, l the number of variables, m the number of constraints, and q the constraint arity.

Definition 3.3. *A* $(\Sigma, \mathsf{q}, \mathsf{l})$**-constraint** C *is a tuple of indices* $(i_1, \ldots, i_{\mathsf{q}})$ *in* $[\mathsf{l}]$ *and a function* $f\colon \Sigma^{\mathsf{q}} \to \{0,1\}$*. An assignment* $a\colon [\mathsf{l}] \to \Sigma$ **satisfies** *the constraint* C *if* $f\big(a(i_1), \ldots, a(i_{\mathsf{q}})\big) = 1$*.*

Definition 3.4. *A* $(\Sigma, \mathsf{q}, \mathsf{l}, m)$**-CSP** *is a list* $\psi = (C_1, \ldots, C_m)$ *where each* C_i *is a* $(\Sigma, \mathsf{l}, \mathsf{q})$*-constraint. We say that* ψ *is a* (Σ, q)*-CSP if it is a* $(\Sigma, \mathsf{q}, \mathsf{l}, m)$*-CSP for some* l *and* m*.*

The CSP ψ *is* **satisfiable** *if there exists an assignment that satisfies each of its constraints. The* **value** *of the CSP* ψ *is the maximum fraction of satisfied constraints across any assignment.*

It is a well-known fact that (non-adaptive) PCPs can be translated into CSPs. In this paper we use an extension of this fact that works for PCPs where the verifier receives non-uniform advice. This extension follows straightforwardly from the standard PCP-to-CSP transformation.

Fact 3.5 (PCP to CSP)**.** *Let R be a relation with a non-adaptive PCP with alphabet Σ, proof length l, query complexity q, randomness complexity r, decision time d, completeness error α, and soundness error β where the verifier uses ℓ bits of non-uniform advice.*

There exists a deterministic reduction that, given ℓ bits of non-uniform advice, receives as input an instance $\mathbb{x}$ for R, runs in time $\mathrm{poly}(2^{\mathsf{r}}, \mathsf{d})$, and outputs a $(\Sigma, \mathsf{l}, \mathsf{q}, 2^{\mathsf{r}})$-CSP such that:
- *if $\mathbb{x} \in L(R)$ then the value of ψ is at least $1 - \alpha$;*
- *if $\mathbb{x} \notin L(R)$ then the value of ψ is at most β.*

Moreover, the size of each constraint in ψ is $\mathrm{poly}(\mathsf{d})$, and so $|\psi| = \mathrm{poly}(2^{\mathsf{r}}, \mathsf{d})$.

We use theorems for efficiently solving binary-alphabet CSPs with arity 2 and 3.

Theorem 3.6 ([Sch78])**.** *There exists an algorithm that decides in time $\mathrm{poly}(|\psi|)$ whether a $(\{0,1\}, 2)$-CSP instance ψ is satisfiable.*

Theorem 3.7 ([Zwi98])**.** *There exists an algorithm that decides in time $\mathrm{poly}(|\psi|)$ whether a $(\{0,1\}, 3)$-CSP instance ψ has value 1 or value at most $5/8$.*

3.5 Pseudorandom generators

We define pseudorandom generators that fool oracle circuits of a given size.

Definition 3.8. *For a boolean function f, an* **f-oracle circuit** *is a boolean circuit that includes f gates (in addition to standard gates). The size of the circuit is the total number of wires and gates.*

Definition 3.9. *A function $\mathsf{G}\colon \{0,1\}^{\ell_{\mathsf{PRG}}(n)} \to \{0,1\}^n$ is a* **pseudorandom generator (PRG)** *with seed length ℓ_{PRG}, computation time $\mathsf{t}_{\mathsf{PRG}}$, and error ϵ_{PRG} against f-circuits of size $\mathsf{s}_{\mathsf{PRG}}(n)$ if (i) G is computable by a Turing machine in time $\mathsf{t}_{\mathsf{PRG}}$; and (ii) for every f-circuit C of size at most $\mathsf{s}_{\mathsf{PRG}}(n)$ it holds that:*

$$|\Pr[C(U_n) = 1] - \Pr[C(\mathsf{G}(U_{\ell_{\mathsf{PRG}}})) = 1]| \leq \epsilon_{\mathsf{PRG}} \ .$$

Above U_n denotes the uniform distribution over binary strings of length n.

Results in [NW94; IW97] imply that, under certain complexity assumptions, PRGs with logarithmic seed length exist. It was later observed in [AIKS16] that these results can be extended to more powerful circuits.

Theorem 3.10 ([AIKS16])**.** *Let f be a boolean function. Assume that there exists a function in $\mathsf{E} = \mathrm{DTIME}(2^{O(n)})$ with f-circuit complexity $2^{\Omega(n)}$. Then, for every polynomial $\mathsf{s}_{\mathsf{PRG}}$, there exists a PRG $\mathsf{G}\colon \{0,1\}^{\ell_{\mathsf{PRG}}(n)} \to \{0,1\}^n$ against f-circuits of size $\mathsf{s}_{\mathsf{PRG}}(n)$, where $\ell_{\mathsf{PRG}}(n) = O(\log n)$ and $\epsilon_{\mathsf{PRG}} = 1/\mathsf{s}_{\mathsf{PRG}}(n)$.*

The assumption underlying the above theorem is a worst-case assumption that can be seen as a natural generalization of the assumption that $\mathsf{E} \not\subseteq \mathrm{NP}$. For a further discussion about this type of assumptions see [AIKS16; AASY16].

3.6 Exponential-time hypotheses

The exponential time hypothesis, introduced in [IP01], is a strengthening of the P $\neq$ NP conjecture roughly stating that 3SAT on n variables using deterministic algorithms requires exponential time to decide in the worst case.

Conjecture 3.11. *The* **exponential time hypothesis (ETH)** *states that there exists a constant $c > 0$ such that* 3SAT $\notin$ DTIME$[2^{c \cdot n}]$.

A strengthening of the exponential time hypothesis introduced in [DHMTW14] says that deciding 3SAT requires exponential time even for probabilistic algorithms:

Conjecture 3.12. *The* **randomized exponential time hypothesis (RETH)** *states that there exists a constant $c > 0$ such that* 3SAT $\notin$ BPTIME$[2^{c \cdot n}]$.

3.7 Probabilistic inequalities

We use McDiarmid's Inequality:

Theorem 3.13 (McDiarmid's Inequality)**.** *Let $X_1, X_2, \ldots, X_n$ be independent random variables such that $X_i \in \mathcal{K}_i$, for some measurable set $\mathcal{K}_i$. Suppose $f : \prod_{i=1}^{n} \mathcal{K}_i \to \mathbb{R}$ is 'Lipschitz' in the following sense: for each $k \leq n$ and any two input sequence $x, x' \in \prod_i \mathcal{K}_i$, that differ only in the k-th coordinate,*

$$|f(x) - f(x')| \leq \sigma_k.$$

Let $Y = f(X_1, X_2, \ldots, X_n)$. Then for every $\gamma > 0$,

$$\Pr[|Y - \mathbb{E}[Y]| \geq \gamma] \leq 2 \cdot \exp\left(-\frac{2\gamma^2}{\sum_{i=1}^{n} \sigma_i^2}\right).$$

We use the following fact:

Fact 3.14. *Let $n, k, q \in \mathbb{N}$ be parameters such that $q \leq k \leq n$. Then:*

$$\binom{n-q}{k-q} / \binom{n}{k} > (k/(e \cdot n))^q \ .$$

Proof.

$$\begin{aligned}
\binom{n-q}{k-q} / \binom{n}{k} &= \frac{(n-q)!}{(k-q)! \cdot (n-k)!} \cdot \frac{k! \cdot (n-k)!}{n!} \\
&= \frac{(n-q)!}{n!} \cdot \frac{k!}{(k-q)!} \\
&= \binom{k}{q} / \binom{n}{q} \\
&> \left(\frac{k}{q}\right)^q \cdot \left(\frac{q}{e \cdot n}\right)^q \\
&= (k/(e \cdot n))^q \ .
\end{aligned}$$

□

4 Tools for length and round reduction

We show several transformations for reducing the length of IOPs, either by reducing the number of rounds, or by reducing each round's proof length.

- Section 4.1: a length-reduction lemma for IOPs with small query complexity.
- Section 4.2: a round-reduction lemma for IOPs that do not query each of their rounds.
- Section 4.3: a statement that captures how IOPs can be "unrolled" into PCPs.

4.1 Length reduction

The lemma below shows that the length an IOP can be reduced, albeit with an increase in the completeness error.

Lemma 4.1. *Let R be a relation with a public-coin IOP $(\mathbf{P}, \mathbf{V})$ with total proof length $\mathsf{l}_{\mathsf{tot}}$ and query complexity q. Let ℓ be a parameter satisfying $\mathsf{q} \leq \ell \leq \mathsf{l}_{\mathsf{tot}}$. Then Construction 4.2 yields a public-coin IOP $(\mathbf{P}', \mathbf{V}')$ for R with the following parameters:*

IOP $(\mathbf{P}, \mathbf{V})$ for R	
Completeness error	α
Soundness error	β
Rounds	k
Alphabet size	λ
Proof length (total)	$\mathsf{l}_{\mathsf{tot}}$
Queries	q
Randomness (per round)	r
Verifier running time	vt

$\longrightarrow$

Round-reduced IOP $(\mathbf{P}', \mathbf{V}')$ for R	
Completeness error	$1 - (1-\alpha) \cdot \binom{\mathsf{l}_{\mathsf{tot}} - \mathsf{q}}{\ell - \mathsf{q}} / \binom{\mathsf{l}_{\mathsf{tot}}}{\ell} \leq 1 - (1-\alpha) \cdot (\ell / (e \cdot \mathsf{l}_{\mathsf{tot}}))^{\mathsf{q}}$
Soundness error	β
Rounds	k
Alphabet size	λ
Proof length (total)	ℓ
Queries	q
Randomness (per round)	$\mathsf{r} + \log \binom{\mathsf{l}_{\mathsf{tot}}}{\ell} \leq \mathsf{r} + \ell \cdot \log \mathsf{l}_{\mathsf{tot}}$
Verifier running time	$\mathrm{poly}(\mathsf{vt}, \ell, \log \mathsf{l}_{\mathsf{tot}})$

Moreover, if $(\mathbf{P}, \mathbf{V})$ is non-adaptive, then so is $(\mathbf{P}', \mathbf{V}')$.

Construction 4.2. The 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. $\mathbf{V}'$: Sample a random set of ℓ indices $I := \{i_k\}_{k \in [\ell]} \in \binom{[\mathsf{l}_{\mathsf{tot}}]}{\ell}$ with $i_1 < \cdots < i_\ell$. Note that these indices span all of the prover messages. Send I to the prover.

2. For $j = 1, \ldots, \mathsf{k}$:

   (a) $\mathbf{V}'$: Randomly sample and send $\rho_j \leftarrow \{0,1\}^{\mathsf{r}}$.

   (b) $\mathbf{P}'$: Compute $\pi_j := \mathbf{P}(\mathbb{x}, \mathbb{w}, \rho_1, \ldots, \rho_j)$. Send π'_j equal to π_j restricted to the indices appropriate to π_j in I.

3. $\mathbf{V}'$: Given direct access to interaction randomness $\rho_1, \ldots, \rho_{\mathsf{k}}$ and oracle access to the prover messages $\pi'_1, \ldots, \pi'_{\mathsf{k}}$:
   (a) Simulate $\mathbf{V}^{\pi_1,\ldots,\pi_{\mathsf{k}}}(\mathbb{x}, \rho_1, \ldots, \rho_{\mathsf{k}})$ where, for any query made by the verifier to a location $j \in I$, the query is forwarded to the appropriate location proof oracle. If $\mathbf{V}$ queries some location $j \notin I$, then immediately reject.
   (b) Output the same answer as $\mathbf{V}$.

Proof of Lemma 4.1. We prove completeness, then soundness, and finally analyze complexity measures.

Completeness. Fix $(\mathbb{x}, \mathbb{w}) \in R$. Consider the following algorithms $\mathbf{A}'$ and $\mathbf{A}$.

- $\mathbf{A}'(\mathbb{x}, \mathbb{w})$:
  1. Choose a random set $I \in \binom{[\mathsf{l}_{\mathsf{tot}}]}{\ell}$.
  2. For $i \in [\mathsf{k}]$ run $\mathbf{V}$ to receive ρ_i and compute $\pi_j := \mathbf{P}(\mathbb{x}, \mathbb{w}, \rho_1, \ldots, \rho_j)$.
  3. Run $\mathbf{V}^{\pi_1,\ldots,\pi_{\mathsf{k}}}(\mathbb{x}, \rho_1, \ldots, \rho_{\mathsf{k}})$, where if $\mathbf{V}$ queries a location $j \in I$ then answer according to the appropriate proof. If $\mathbf{V}$ queries at a location $j \notin I$ then output 0 and quit.
  4. If the algorithm has not quit, output whatever $\mathbf{V}$ output.

- $\mathbf{A}(\mathbb{x}, \mathbb{w})$:
  1. Choose a random set $I \in \binom{[\mathsf{l}_{\mathsf{tot}}]}{\ell}$.
  2. For $i \in [\mathsf{k}]$ run $\mathbf{V}$ to receive ρ_i and compute $\pi_j := \mathbf{P}(\mathbb{x}, \mathbb{w}, \rho_1, \ldots, \rho_j)$.
  3. Run $\mathbf{V}^{\pi_1,\ldots,\pi_{\mathsf{k}}}(\mathbb{x}, \rho_1, \ldots, \rho_{\mathsf{k}})$. For every $j \in [\mathsf{k}]$, let Q be the set of locations queries by $\mathbf{V}$.
  4. If $Q \subseteq I$ then output whatever $\mathbf{V}$ output. Otherwise, output 0.

Notice that $\mathbf{A}'$ can be viewed as an emulation of $(\mathbf{P}', \mathbf{V}')$, and so:

$$\Pr\left[\langle \mathbf{P}'(\mathbb{w}), \mathbf{V}' \rangle(\mathbb{x}) = 1\right] = \Pr\left[\mathbf{A}'(\mathbb{x}, \mathbb{w}) = 1\right] \ .$$

Moreover, there is no difference between quitting whenever $\mathbf{V}$ queries outside of the set I and actually generating the result of $\mathbf{V}$ and only then quitting if $\mathbf{V}$ queried outside of I:

$$\Pr\left[\mathbf{A}'(\mathbb{x}, \mathbb{w}) = 1\right] = \Pr\left[\mathbf{A}(\mathbb{x}, \mathbb{w}) = 1\right] \ .$$

Finally, notice that the simulated prover and verifier in $\mathbf{A}(\mathbb{x}, \mathbb{w})$ constitute a random execution of $(\mathbf{P}(\mathbb{w}), \mathbf{V})$ on input $\mathbb{x}$. Fix Q of size q. The probability that I contains Q is equal to $\binom{\mathsf{l}_{\mathsf{tot}}-\mathsf{q}}{\ell-\mathsf{q}} / \binom{\mathsf{l}_{\mathsf{tot}}}{\ell}$. Therefore:

$$\Pr\left[\mathbf{A}(\mathbb{x}, \mathbb{w}) = 1\right] \geq \frac{\binom{\mathsf{l}_{\mathsf{tot}}-\mathsf{q}}{\ell-\mathsf{q}}}{\binom{\mathsf{l}_{\mathsf{tot}}}{\ell}} \cdot \Pr\left[\langle \mathbf{P}(\mathbb{w}), \mathbf{V} \rangle(\mathbb{x}) = 1\right] \ .$$

Putting all of this together with the fact that the IOP $(\mathbf{P}, \mathbf{V})$ has completeness error α, we have that

$$\begin{aligned}
\Pr\left[\langle \mathbf{P}'(\mathbb{w}), \mathbf{V}' \rangle(\mathbb{x}) = 1\right] &\geq \frac{\binom{\mathsf{l}_{\mathsf{tot}}-\mathsf{q}}{\ell-\mathsf{q}}}{\binom{\mathsf{l}_{\mathsf{tot}}}{\ell}} \cdot \Pr\left[\langle \mathbf{P}(\mathbb{w}), \mathbf{V} \rangle(\mathbb{x}) = 1\right] \\
&\geq (1-\alpha) \cdot \frac{\binom{\mathsf{l}_{\mathsf{tot}}-\mathsf{q}}{\ell-\mathsf{q}}}{\binom{\mathsf{l}_{\mathsf{tot}}}{\ell}} \\
&> (1-\alpha) \cdot (\ell/(e \cdot \mathsf{l}_{\mathsf{tot}}))^{\mathsf{q}} \ ,
\end{aligned}$$

where the final inequality follows from Fact 3.14.

Soundness. Fix $\mathbb{x} \notin L(R)$ and a malicious prover $\tilde{\mathbf{P}}'$ for the IOP $(\mathbf{P}', \mathbf{V}')$. Consider the prover $\tilde{\mathbf{P}}$ for the IOP $(\mathbf{P}, \mathbf{V})$ defined as follows:

- $\tilde{\mathbf{P}}$:
  1. Choose a random set $I \in \binom{[\mathsf{l}_{\mathsf{tot}}]}{\ell}$.
  2. For $j \in [\mathsf{k}]$:
     (a) Receive ρ_j from the verifier $\mathbf{V}$.
     (b) Compute $\pi'_j := \tilde{\mathbf{P}}'(\rho_1, \ldots, \rho_j)$ and let I_j be I restricted to the indices related to the j-th message of $\mathbf{P}$. Define π_j as follows: for every $i \in I_j$, set $\mathbf{P}_j[i]$ to be equal to the appropriate location in π'_j. For any $i \notin I_j$, set $\mathbf{P}_j[i] := 0$. Send π_i to $\mathbf{V}$.

The sets I chosen by $\mathbf{V}'$ and $\tilde{\mathbf{P}}$ are chosen with the same probability. For every fixed $I = I^*$, the probability that $\mathbf{V}$ simulated by $\mathbf{V}'$ queries only in I is identical to the probability that the real $\mathbf{V}$ decides to query only in I in the interaction with $\tilde{\mathbf{P}}$. When there are only queries in I, the verifiers in both cases have an identical view, and so accept with the same probability. When the $\mathbf{V}$ simulated by $\mathbf{V}'$ queries outside of I, it immediately rejects, and, the real verifier $\mathbf{V}$ may or may not reject when it samples outside of I, since it reads from proofs padded by zeroes. Therefore

$$\Pr\left[\langle \tilde{\mathbf{P}}', \mathbf{V}' \rangle(\mathbb{x}) = 1\right] \leq \Pr\left[\langle \tilde{\mathbf{P}}, \mathbf{V} \rangle(\mathbb{x}) = 1\right] \ .$$

Finally, since the IOP $(\mathbf{P}, \mathbf{V})$ has soundness error β, we conclude that

$$\Pr\left[\langle \tilde{\mathbf{P}}', \mathbf{V}' \rangle(\mathbb{x}) = 1\right] \leq \Pr\left[\langle \tilde{\mathbf{P}}, \mathbf{V} \rangle(\mathbb{x}) = 1\right] \leq \beta \ .$$

Complexity parameters. We analyze the complexity parameters of the new IOP.

- *Rounds.* The protocol has k rounds. (The message I and the first set of randomness sent by the verifier can be merged into a single message.)
- *Alphabet size and proof length.* The alphabet remains unchanged. The total proof length of the IOP is ℓ.
- *Queries.* The verifier makes q queries while simulating $\mathbf{V}$.
- *Randomness.* The verifier begins by sending $I \in \binom{[\mathsf{l}_{\mathsf{tot}}]}{\ell}$, which takes $\log \binom{\mathsf{l}_{\mathsf{tot}}}{\ell}$ bits. In the same round it sends its first random message of r bits. In each subsequent round, $\mathbf{V}'$ sends r bits of randomness. Thus the per-round interaction-randomness complexity is $\mathsf{r} + \log \binom{\mathsf{l}_{\mathsf{tot}}}{\ell} \leq \mathsf{r} + \ell \cdot \log \mathsf{l}_{\mathsf{tot}}$.
- *Verifier running time.* The running time of the verifier is $\mathrm{poly}(\mathsf{vt}, \ell, \log \mathsf{l}_{\mathsf{tot}})$.
- *Adaptivity.* The IOP is non-adaptive if the original IOP was non-adaptive.

□

4.2 Round reduction

The lemma below shows that the round complexity of an IOP can be reduced, albeit with an increase in the completeness error.

Lemma 4.3. *Let R be a relation with a public-coin k-round round-query IOP $(\mathbf{P}, \mathbf{V})$ with round-query complexity $\mathsf{q}_{\mathsf{round}}$. Let ℓ be a parameter such that $\mathsf{q}_{\mathsf{round}} \leq \ell \leq \mathsf{k}$. Then Construction 4.4 yields a public-coin IOP $(\mathbf{P}', \mathbf{V}')$ for R with parameters related*

IOP $(\mathbf{P}, \mathbf{V})$ for R	
Completeness error	α
Soundness error	β
Rounds	k
Alphabet size	λ
Proof length (per round)	l
Round queries	$\mathsf{q}_{\mathsf{round}}$
Queries (in total)	q
Randomness (per round)	r
Verifier running time	vt

$\longrightarrow$

Round-reduced IOP $(\mathbf{P}', \mathbf{V}')$ for R	
Completeness error	$1 - (1-\alpha) \cdot \binom{\mathsf{k}-\mathsf{q}_{\mathsf{round}}}{\ell-\mathsf{q}_{\mathsf{round}}} / \binom{\mathsf{k}}{\ell} \leq 1 - (1-\alpha) \cdot (\ell/(e \cdot \mathsf{k}))^{\mathsf{q}_{\mathsf{round}}}$
Soundness error	β
Rounds	$\ell + 1$
Alphabet size	λ
Proof length (per round)	l
Round queries	$\mathsf{q}_{\mathsf{round}}$
Queries (in total)	q
Randomness (per round)	$\mathsf{k} \cdot \mathsf{r} + \log \binom{\mathsf{k}}{\ell} \leq \mathsf{k} \cdot \mathsf{r} + \ell \cdot \log \mathsf{k}$
Verifier running time	$\mathrm{poly}(\mathsf{vt})$

Moreover, if $(\mathbf{P}, \mathbf{V})$ is non-adaptive, then so is $(\mathbf{P}', \mathbf{V}')$.

The goal of the construction is to decrease the number of rounds in the protocol to be equal to the number of prover messages that the verifier reads. The construction follows from the intuition that if the verifier does not read all of the prover's messages in an IOP, then the prover might as well not send them. Unfortunately, we cannot know the beginning of the protocol execution which of the prover's messages the verifier will read (in order to not send messages that will not be read). Instead, we have the verifier *guess* which of the prover's messages it will read and send this guess to the prover. The prover now skips over sending these messages.

If it turns out that the verifier chose correctly, then it has access to everything it needs to read. If it chose incorrectly, then it rejects. This behavior causes a degradation in completeness error, but the soundness error remains unchanged.

Construction 4.4. The 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. $\mathbf{V}'$: Sample a random set of ℓ indices $I := \{i_j\}_{j \in [\ell]} \in \binom{[\mathsf{k}]}{\ell}$ with $i_1 < \cdots < i_\ell$. For notational convenience, set $i_0 := 0$. Send I to the prover.

2. For $j = 1, \ldots, \ell$:

   (a) $\mathbf{V}'$: Randomly sample and send $\rho_{i_{(j-1)}+1}, \ldots, \rho_{i_j} \leftarrow \{0,1\}^{\mathsf{r}}$.

   (b) $\mathbf{P}'$: Compute and send $\pi_{i_j} := \mathbf{P}(\mathbb{x}, \mathbb{w}, \rho_1, \ldots, \rho_{i_j})$.

3. $\mathbf{V}'$: Given direct access to interaction randomness $\rho_1, \ldots, \rho_{i_\ell}$ and oracle access to the prover messages $\pi_{i_1}, \ldots, \pi_{i_\ell}$:
   (a) Randomly sample $\rho_{(i_\ell+1)}, \ldots \rho_{\mathsf{k}} \leftarrow \{0,1\}^{\mathsf{r}}$.
   (b) Simulate $\mathbf{V}^{\pi_1,\ldots,\pi_{\mathsf{k}}}(\mathbb{x}, \rho_1, \ldots, \rho_{\mathsf{k}})$ where if the verifier queries the proof π_j for $j \in I$, the query is forwarded to the appropriate oracle. If $\mathbf{V}$ queries some $j \notin I$, then immediately reject.
   (c) Output the same answer as $\mathbf{V}$.

Remark 4.5. As written in Construction 4.4, the protocol $(\mathbf{P}', \mathbf{V}')$ 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.

Proof of Lemma 4.3. We analyze completeness, soundness, and complexity measures.

Completeness. Fix $(\mathbb{x}, \mathbb{w}) \in R$. Consider the following algorithms $\mathbf{A}'$ and $\mathbf{A}$.

- $\mathbf{A}'(\mathbb{x}, \mathbb{w})$:
  1. Choose a random set $I \in \binom{[\mathsf{k}]}{\ell}$.
  2. For $i \in [\mathsf{k}]$ run $\mathbf{V}$ to receive ρ_i and compute $\pi_i := \mathbf{P}(\mathbb{x}, \mathbb{w}, \rho_1, \ldots, \rho_i)$.
  3. Run $\mathbf{V}^{\pi_1,\ldots,\pi_{\mathsf{k}}}(\mathbb{x}, \rho_1, \ldots, \rho_{\mathsf{k}})$, where if $\mathbf{V}$ queries $j \in I$ then answer using π_j. If $\mathbf{V}$ queries $j \notin I$ then output 0 and quit.
  4. If the algorithm has not quit, output whatever $\mathbf{V}$ outputs.

- $\mathbf{A}(\mathbb{x}, \mathbb{w})$:
  1. Choose a random set $I \in \binom{[\mathsf{k}]}{\ell}$.
  2. For $i \in [\mathsf{k}]$ run $\mathbf{V}$ to receive ρ_i and compute $\pi_i := \mathbf{P}(\mathbb{x}, \mathbb{w}, \rho_1, \ldots, \rho_i)$.
  3. Run $\mathbf{V}^{\pi_1,\ldots,\pi_{\mathsf{k}}}(\mathbb{x}, \rho_1, \ldots, \rho_{\mathsf{k}})$. Let Q_{rnd} be the set of rounds queried by $\mathbf{V}$.
  4. If $Q_{\mathsf{rnd}} \subseteq I$ then output whatever $\mathbf{V}$ output. Otherwise output 0.

Notice that $\mathbf{A}'$ can be viewed as an emulation of $(\mathbf{P}', \mathbf{V}')$, and so:

$$\Pr\left[\langle \mathbf{P}'(\mathbb{w}), \mathbf{V}' \rangle(\mathbb{x}) = 1\right] = \Pr\left[\mathbf{A}'(\mathbb{x}, \mathbb{w}) = 1\right] \ .$$

Moreover, there is no difference between quitting whenever $\mathbf{V}$ queries outside of I and actually generating the result of $\mathbf{V}$ and only then quitting if $\mathbf{V}$ queried outside of I:

$$\Pr\left[\mathbf{A}'(\mathbb{x}, \mathbb{w}) = 1\right] = \Pr\left[\mathbf{A}(\mathbb{x}, \mathbb{w}) = 1\right] \ .$$

Finally, notice that the simulated prover and verifier in $\mathbf{A}(\mathbb{x}, \mathbb{w})$ constitute a random execution of $(\mathbf{P}(\mathbb{w}), \mathbf{V})$ on input $\mathbb{x}$. For a fixed Q_{rnd}, the probability that I contains Q_{rnd} is equal to $\binom{\mathsf{k}-\mathsf{q}_{\mathsf{round}}}{\ell-\mathsf{q}_{\mathsf{round}}} / \binom{\mathsf{k}}{\ell}$. Therefore:

$$\Pr\left[\mathbf{A}(\mathbb{x}, \mathbb{w}) = 1\right] > \frac{\binom{\mathsf{k}-\mathsf{q}_{\mathsf{round}}}{\ell-\mathsf{q}_{\mathsf{round}}}}{\binom{\mathsf{k}}{\ell}} \cdot \Pr\left[\langle \mathbf{P}(\mathbb{w}), \mathbf{V} \rangle(\mathbb{x}) = 1\right] \ .$$

Putting all of this together with the fact that the IOP $(\mathbf{P}, \mathbf{V})$ has completeness error α, we have

that

$$\begin{aligned}\Pr\left[\langle \mathbf{P}'(\mathbb{w}), \mathbf{V}'\rangle(\mathbb{x}) = 1\right] &> \frac{\binom{\mathsf{k}-\mathsf{q}_{\mathsf{round}}}{\ell-\mathsf{q}_{\mathsf{round}}}}{\binom{\mathsf{k}}{\ell}} \cdot \Pr\left[\langle \mathbf{P}(\mathbb{w}), \mathbf{V}\rangle(\mathbb{x}) = 1\right] \\ &\geq (1-\alpha) \cdot \frac{\binom{\mathsf{k}-\mathsf{q}_{\mathsf{round}}}{\ell-\mathsf{q}_{\mathsf{round}}}}{\binom{\mathsf{k}}{\ell}} \\ &> (1-\alpha) \cdot (\ell/(e \cdot \mathsf{k}))^{\mathsf{q}_{\mathsf{round}}} \ ,\end{aligned}$$

where the final inequality follows from Fact 3.14.

Soundness. Fix $\mathbb{x} \notin L(R)$ and a malicious prover $\tilde{\mathbf{P}}'$ for the IOP $(\mathbf{P}', \mathbf{V}')$. Consider the prover for the IOP $(\mathbf{P}, \mathbf{V})$ defined as follows:

- $\tilde{\mathbf{P}}$:
  1. Choose a random set $I = \{i_j\}_{j \in [\ell]}$.
  2. For $i \in [\mathsf{k}]$:
     (a) Receive ρ_i from the verifier $\mathbf{V}$.
     (b) If $i \in I$, compute $\pi_i := \tilde{\mathbf{P}}'(\rho_1, \ldots, \rho_i)$. Otherwise, let π_i be the all-zeroes proof. Send π_i to $\mathbf{V}$.

The sets I chosen by $\mathbf{V}'$ and $\tilde{\mathbf{P}}$ are chosen with the same probability. For every fixed I, the probability that $\mathbf{V}$ simulated by $\mathbf{V}'$ queries only in I is identical to the probability that the real $\mathbf{V}$ decides to query only in I in the interaction with $\tilde{\mathbf{P}}$. When there are only queries in I, the verifiers in both cases have an identical view, and so accept with the same probability. When the $\mathbf{V}$ simulated by $\mathbf{V}'$ queries outside of I, it immediately rejects, and, the real verifier $\mathbf{V}$ may or may not reject when it samples outside of I, since it reads from the all-zeroes proofs. Therefore

$$\Pr\left[\langle \tilde{\mathbf{P}}', \mathbf{V}'\rangle(\mathbb{x}) = 1\right] \leq \Pr\left[\langle \tilde{\mathbf{P}}, \mathbf{V}\rangle(\mathbb{x}) = 1\right] \ .$$

Finally, since the IOP $(\mathbf{P}, \mathbf{V})$ has soundness error β, we conclude that

$$\Pr\left[\langle \tilde{\mathbf{P}}', \mathbf{V}'\rangle(\mathbb{x}) = 1\right] \leq \Pr\left[\langle \tilde{\mathbf{P}}, \mathbf{V}\rangle(\mathbb{x}) = 1\right] \leq \beta \ .$$

Complexity parameters. We analyze the complexity parameters of the new IOP.

- *Rounds.* The protocol has $\ell + 1$ rounds. The message I and the first set of randomness sent by the verifier can be merged into a single message, but the final choice of randomness cannot be merged into a previous round.
- *Alphabet size and proof length.* The alphabet and proof length of the new IOP are identical to that of the original IOP.
- *Round queries.* The verifier reads queries from $\mathsf{q}_{\mathsf{round}}$ rounds.
- *Queries.* The verifier makes q queries while simulating $\mathbf{V}$.
- *Randomness.* The verifier first sends a random $I \in \binom{[\mathsf{k}]}{\ell}$, which takes $\log \binom{\mathsf{k}}{\ell}$ random bits. Following the padding of the interaction randomness (see Remark 4.5), in each subsequent round $\mathbf{V}'$ sends $\mathsf{k} \cdot \mathsf{r}$ bits of randomness. Overall, the first round is the longest, in which the verifier sends $\mathsf{k} \cdot \mathsf{r} + \log \binom{\mathsf{k}}{\ell} \leq \mathsf{k} \cdot \mathsf{r} + \ell \cdot \log \mathsf{k}$ bits.

- *Verifier running time.* The running time of the verifier is $\mathrm{poly}(\mathsf{vt}, \mathsf{k}) = \mathrm{poly}(\mathsf{vt})$.
- *Adaptivity.* The IOP is non-adaptive if the original IOP was non-adaptive.

□

4.3 Unrolling IOPs to PCPs

The lemma below captures the folklore idea that any (public-coin) IOP can be unrolled into a PCP, whose proof length depends exponentially in the number of rounds and randomness.

Lemma 4.6. *Let R be a relation with a public-coin IOP system $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$. Then R has a PCP system $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$ with parameters related as below.*

IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ for R	
Completeness error	α
Soundness error	β
Rounds	k
Alphabet size	λ
Proof length (per round)	l
Queries	q
Randomness (per round)	r
Verifier running time	vt
Advice length	ℓ

$\longrightarrow$

PCP $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$ for R	
Completeness error	α
Soundness error	β
Alphabet size	λ
Length	$\mathsf{l} \cdot 2^{O(\mathsf{k} \cdot \mathsf{r})}$
Queries	q
Randomness	$\mathsf{k} \cdot \mathsf{r}$
Verifier running time	$\mathrm{poly}(\mathsf{vt})$
Advice length	ℓ

If $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ is non-adaptive, then so is $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$. Moreover, any good advice string for $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ is also a good advice string for $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$.

Remark 4.7. A similar statement can be made for *private-coin* IOPs. For such IOPs the verifier may not sample random coins in each round, and as such we cannot reason about randomness *per round*, but rather consider the total number of random coins used by the verifier, $\mathsf{r}_{\mathsf{tot}}$. The PCP after unrolling has length $\mathsf{l} \cdot 2^{O(\mathsf{r}_{\mathsf{tot}})}$ and its verifier uses $\mathsf{r}_{\mathsf{tot}}$ random bits. A shorter length may be achieved by keeping track of the space of possible verifier messages rather than its randomness.

5 Tools for improving completeness

We show two transformations for reducing the completeness error of IOPs.

- Section 5.1: reducing the completeness error while increasing the alphabet size, and increasing the query complexity by 1.
- Section 5.2: achieving perfect completeness while increasing the alphabet size, and increasing the query complexity by 2.

5.1 Completeness amplification

The lemma below shows that the completeness error of an IOP can be reduced via a variation of parallel repetition, while the query complexity increases by 1.

Lemma 5.1. *Let $t \in \mathbb{N}$ be a parameter. Let R be a relation with a public-coin IOP $(\mathbf{P}, \mathbf{V})$. Then Construction 5.2 yields a public-coin IOP $(\mathbf{P}', \mathbf{V}')$ for R with parameters related as below.*

IOP $(\mathbf{P}, \mathbf{V})$ for R	
Completeness error	α
Soundness error	β
Rounds	k
Alphabet size	λ
Proof length (per round)	l
Queries	q
Randomness (per round)	r
Verifier running time	vt

$\longrightarrow$

Completeness-amplified IOP $(\mathbf{P}', \mathbf{V}')$ for R	
Completeness error	α^t
Soundness error	$t \cdot \beta$
Rounds	$\mathsf{k} + 1$
Alphabet size	$\max\{\lambda, t\}$
Proof length (per round)	$t \cdot \mathsf{l}$
Queries	$\mathsf{q} + 1$
Randomness (per round)	$t \cdot \mathsf{r}$
Verifier running time	$\mathrm{poly}(\mathsf{vt}, t)$

Construction 5.2. The 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. $\mathbf{P}'$ and $\mathbf{V}'$ run t independent parallel executions of the interaction between $\mathbf{P}$ and $\mathbf{V}$ (excluding the decision stage) on instance $\mathbb{x}$ with witness $\mathbb{w}$. Let $\vec{\pi}_i$ and $\vec{\rho}_i$ be the proofs and interaction random strings corresponding to the i-th execution.

2. $\mathbf{P}'$:
   (a) If there exists an index $i \in [t]$, such that $\mathbf{V}^{\vec{\pi}_i}(\mathbb{x}, \vec{\rho}_i) = 1$, then send i to the verifier $\mathbf{V}'$ as a non-oracle message (choosing arbitrarily if there is more than one such index).
   (b) Otherwise, send an arbitrary value for i as a non-oracle message (or, alternatively, quit the execution).

3. $\mathbf{V}'$: Receive the value i (as a non-oracle message) from the prover. Check that $\mathbf{V}^{\vec{\pi}_i}(\mathbb{x}, \vec{\rho}_i)=1$, querying the appropriate proofs in the i-th execution as required by $\mathbf{V}$.

Notice that the verifier in this construction is adaptive.

Proof of Lemma 5.1. We analyze completeness, soundness, and complexity measures.

Completeness. Fix $(\mathbb{x}, \mathbb{w}) \in R$. By completeness of the original IOP, the verifier rejects in a parallel execution with probability at most α. Therefore, the probability that the verifier rejects in

all executions is at most α^t. If there is any execution for which $\mathbf{V}$ accepts, then the prover sends its index as i, and $\mathbf{V}'$ accepts when emulating $\mathbf{V}$ on the i-th execution. We conclude that

$$\Pr\left[\langle \mathbf{P}'(\mathbb{w}), \mathbf{V}' \rangle(\mathbb{x}) = 0\right] \leq \alpha^t \ .$$

Soundness. Fix $\mathbb{x} \notin L(R)$ and a malicious prover $\tilde{\mathbf{P}}'$. By soundness of the original IOP, each of the parallel executions accepts with probability at most β. By applying the union bound over all t executions, we have that

$$\Pr\left[\langle \tilde{\mathbf{P}}', \mathbf{V}' \rangle(\mathbb{x}) = 1\right] \leq t \cdot \beta \ .$$

Complexity parameters. We analyze the complexity parameters of the resulting IOP.

- *Rounds.* The protocol has $\mathsf{k}+1$ rounds.
- *Alphabet size and proof length.* The alphabet of the new IOP is identical to that of the original IOP, with the addition of needing to send the index $i \in [t]$. Therefore the alphabet size is $\max\{\lambda, t\}$. The proof length is $t \cdot \mathsf{l}$, as it contains t executions of the original IOP.
- *Queries.* The verifier makes q queries while simulating $\mathbf{V}$, and one to read i. Therefore, the query complexity is $\mathsf{q}+1$.
- *Randomness per-round.* In each round, $\mathbf{V}'$ sends $t \cdot \mathsf{r}$ bits of randomness corresponding to each parallel execution. Therefore the total randomness sent per round is $t \cdot \mathsf{r}$.
- *Verifier running time.* The running time of the verifier is at most $\mathrm{poly}(\mathsf{vt}, t)$.
- *Adaptivity.* The IOP is adaptive since the verifier's queries depend on the index i.

□

5.2 Perfect completeness

The lemma below shows that, for every IOP with good enough soundness error, perfect completeness can be achieved while preserving the query complexity up to constants.

Lemma 5.3. *Let R be a relation with a public-coin IOP $(\mathbf{P}, \mathbf{V})$. Then Construction 5.4 yields a public-coin IOP $(\mathbf{P}', \mathbf{V}')$ for R with parameters related as below.*

IOP $(\mathbf{P}, \mathbf{V})$ for R	
Completeness error	α
Soundness error	β
Rounds	k
Alphabet size	λ
Proof length (per round)	l
Queries	q
Randomness (per round)	r
Verifier running time	vt

$\longrightarrow$

Perfectly-complete IOP $(\mathbf{P}', \mathbf{V}')$ for R	
Completeness error	0
Soundness error	$O\left(\beta \cdot \left(\frac{\mathsf{r}\cdot\mathsf{k}}{\log(1/\alpha)}\right)\right)$
Rounds	$\mathsf{k}+1$
Alphabet size	$\max\{\lambda, 2^{\mathsf{r}\cdot\mathsf{k}}\}$
Proof length (per round)	$O\left(\mathsf{l} \cdot \left(\frac{\mathsf{r}\cdot\mathsf{k}}{\log(1/\alpha)}\right)\right)$
Queries	$\mathsf{q}+2$
Randomness (per round)	r
Verifier running time	$\mathrm{poly}\left(\mathsf{vt}, \log\left(\frac{\mathsf{r}\cdot\mathsf{k}}{\log(1/\alpha)}\right)\right)$

Moreover, if the verifier $\mathbf{V}$ uses non-uniform advice, then $\mathbf{V}'$ uses advice of the same length.

Construction 5.4. Let $t := 2 \cdot \left(\frac{\mathsf{r} \cdot \mathsf{k}}{\log(1/\alpha)}\right)$. The 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.

- $\mathbf{P}'$: Send $\vec{z}_1, \ldots, \vec{z}_t = (z_{1,1}, \ldots, z_{1,\mathsf{k}}), \ldots, (z_{t,1}, \ldots z_{t,\mathsf{k}}) \in \{0,1\}^{\mathsf{r} \cdot \mathsf{k}}$ to the verifier.
- 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 $\pi_{j,i} := \mathbf{P}(\mathbb{x}, \mathbb{w}, \rho_1 \oplus z_{j,1}, \ldots, \rho_i \oplus z_{j,i})$.
- $\mathbf{P}'$:
  1. If there exists an index $j \in [t]$, such that $\mathbf{V}^{\pi_{j,1},\ldots,\pi_{j,\mathsf{k}}}(\mathbb{x}, \rho_1 \oplus z_{j,1}, \ldots, \rho \oplus z_{j,\mathsf{k}}) = 1$, then send j to the verifier $\mathbf{V}'$ as a non-oracle message (choosing arbitrarily if there is more than one possible value for j).
  2. Otherwise, send $\perp$.
- $\mathbf{V}'$: Receive j as a non-oracle message.
  1. If $j = \perp$, then reject.
  2. 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 $\mathbf{V}$.

While this construction (and subsequent analysis) assumes that $\mathbf{V}$ uses no non-uniform advice, the non-uniform case is identical, except that whenever any party invokes $\mathbf{V}$, it also passes along the advice string. Notice that the verifier in this construction is adaptive.

Proof of Lemma 5.3. We analyze completeness, soundness, and complexity measures. Completeness and soundness are identical to those in [FGMSZ89], and are repeated here for convenience.

Completeness. Fix $(\mathbb{x}, \mathbb{w}) \in R$. For convenience of notation, we denote $\vec{\rho} := (\rho_1, \ldots, \rho_{\mathsf{k}})$. Thus, $\vec{\rho} \oplus \vec{z}_j = (\rho_1 \oplus z_{j,1}, \ldots, \rho_{\mathsf{k}} \oplus z_{j,\mathsf{k}})$ for any j. By completeness of the original IOP, the verifier rejects in a parallel execution with probability at most α. We show the following claim:

Claim 5.5. *There exist strings $\vec{z}_1, \ldots, \vec{z}_t = (z_{1,1}, \ldots, z_{1,\mathsf{k}}), \ldots, (z_{t,1}, \ldots z_{t,\mathsf{k}})$ such that for every choice of verifier messages $\vec{\rho} = (\rho_1, \ldots, \rho_{\mathsf{k}})$ there is some $j \in [t]$ satisfying*

$$\mathbf{V}^{\pi_{j,1},\ldots,\pi_{j,\mathsf{k}}}(\mathbb{x}, \vec{\rho} \oplus \vec{z}_j) = 1 \ ,$$

where the prover messages $\pi_{j,i}$ are generated as in Construction 5.4.

This fact completes the proof, as the honest prover will be able to send these values $\vec{z}_1, \ldots, \vec{z}_t$, and, following the interaction with the verifier, is able to send the correct j to the verifier.

Proof of Claim 5.5. Let S be the set of verifier randomness $\vec{\rho}$ for which the original prover $\mathbf{P}$ is able to convince the verifier $\mathbf{V}$ on instance $\mathbb{x}$ and witness $\mathbb{w}$. By completeness of the protocol:

$$\Pr_{\vec{z}}[\vec{z} \notin S] < \alpha \ .$$

We say that a sequence $\vec{z}_1, \ldots, \vec{z}_t$ is *good* if for every $\vec{\rho}$, there exists $j \in [t]$ with $\vec{\rho} \oplus \vec{z}_j \in S$. We show that the probability that a uniformly random $\vec{z}_1, \ldots, \vec{z}_t$ is good is non-zero, which implies that

the prover $\mathbf{P}'$ can always find such a sequence and manages to convince the verifier $\mathbf{V}'$:

$$\begin{aligned}
&\Pr_{\vec{z}_1,\ldots,\vec{z}_t}[\ \vec{z}_1,\ldots,\vec{z}_t \textit{ are not good}\ ] \\
&= \Pr_{\vec{z}_1,\ldots,\vec{z}_t}[\ \exists\,\vec{\rho}\ \forall\, j \in [t]\ \vec{\rho} \oplus \vec{z}_j \notin S\ ] \\
&\leq 2^{\mathsf{r}\cdot\mathsf{k}} \cdot \Pr_{\vec{z}_1,\ldots,\vec{z}_t}[\ \forall j \in [t]\ \vec{z}_j \notin S\ ] \qquad (1) \\
&< 2^{\mathsf{r}\cdot\mathsf{k}} \cdot \alpha^t \\
&< 1\ , \qquad (2)
\end{aligned}$$

where Equation 1 holds since $\vec{\rho}$ is $\mathsf{r}\cdot\mathsf{k}$ bits long, and by the fact that for every fixed string $\vec{\rho}$, the probability that $\vec{\rho} \oplus \vec{z}_j \in S$ for a uniformly random $\vec{z}_j$ is equal to the probability that $\vec{z}_j \in S$. Equation 2 holds since $t > \frac{\mathsf{r}\cdot\mathsf{k}}{\log(1/\alpha)}$. □

Soundness. Fix $\mathbb{x} \notin L(R)$ and a malicious prover $\tilde{\mathbf{P}}'$. Suppose towards contradiction that $\tilde{\mathbf{P}}'$ is able to convince $\mathbf{V}'$ with probability greater than

$$\beta' := \beta \cdot t = 2 \cdot \beta \cdot \left(\frac{\mathsf{r}\cdot\mathsf{k}}{\log(1/\alpha)}\right)\ .$$

We construct a prover $\tilde{\mathbf{P}}$ that manages to convince $\mathbf{V}$ to accept on $\mathbb{x}$ with probability greater than β, which contradicts the soundness error of the original IOP.

$\tilde{\mathbf{P}}$ begins by using $\tilde{\mathbf{P}}'$ on $\mathbb{x}$ to generate a sequence of strings

$$\vec{z}_1,\ldots,\vec{z}_t = (z_{1,1},\ldots,z_{1,\mathsf{k}}),\ldots,(z_{t,1},\ldots z_{t,\mathsf{k}})\ .$$

It then chooses a random $j \in [t]$. During interaction with the verifier, upon receiving a string ρ_i, $\tilde{\mathbf{P}}$ passes $\rho'_i := \rho_i \oplus z_{i,j}$ to $\tilde{\mathbf{P}}'$ and receives proofs $\pi_{1,i},\ldots,\pi_{t,i}$.

Whenever $\tilde{\mathbf{P}}'$ convinces $\mathbf{V}'$ to accept by telling $\mathbf{V}'$ to read the shift string $\vec{z}_j$, this means that $\mathbf{V}$ accepts given proofs $\pi_{j,1},\ldots,\pi_{j,\mathsf{k}}$ and randomness $(\rho'_1 \oplus z_{j,1}),\ldots,(\rho'_{\mathsf{k}} \oplus z_{j,\mathsf{k}})$. Since $\rho'_i := \rho_i \oplus z_{i,j}$, this means that $\mathbf{V}$ accepts given proofs $\pi_{j,1},\ldots,\pi_{j,\mathsf{k}}$ and randomness $\rho_1,\ldots,\rho_{\mathsf{k}}$. Hence, $\tilde{\mathbf{P}}$ is able to cause the verifier to accept whenever $\tilde{\mathbf{P}}'$ manages to do so with the same index j chosen by $\tilde{\mathbf{P}}$. Therefore, the probability that $\tilde{\mathbf{P}}$ causes the verifier to accept is greater than $\beta'/t = \beta$, which contradicts the soundness error of the original IOP $(\mathbf{P},\mathbf{V})$.

Complexity parameters. We analyze the complexity parameters of the resulting IOP.

- *Rounds.* The protocol has $\mathsf{k}+1$ rounds as the prover's final message can be merged with its previous message.
- *Alphabet size and proof length.* The alphabet of the new IOP is identical to that of the original IOP, with the addition of needing to send the strings $\vec{z}_1,\ldots,\vec{z}_t$ (which are each read as a single block by the verifier) and the index $j \in [t] \cup \{\perp\}$. Therefore the alphabet size is $\max\{\lambda, 2^{\mathsf{r}\cdot\mathsf{k}}\}$. The proof length of the first prover message (the vectors $\vec{z}$) is t, and the length of the rest of the messages is $\mathsf{l}\cdot t$ since the prover sends t different proofs of the original IOP to the verifier. The proof length is, therefore, $\mathsf{l}\cdot t = 2\cdot\mathsf{l}\cdot\left(\frac{\mathsf{r}\cdot\mathsf{k}}{\log(1/\alpha)}\right)$ symbols.
- *Queries.* The verifier makes one query to read j, one query to read $\vec{z}_j$ and q queries while simulating $\mathbf{V}$. Therefore, the query complexity is $\mathsf{q}+2$.

- *Randomness per-round.* The randomness complexity of the IOP is r.
- *Verifier running time.* The running time of the verifier is $\mathrm{poly}(\mathsf{vt}, \log t) = \mathrm{poly}\left(\mathsf{vt}, \log\left(\frac{\mathsf{r}\cdot\mathsf{k}}{\log(1/\alpha)}\right)\right)$.
- *Adaptivity.* The IOP is adaptive since the verifier's queries depend of the index j.

□

6 Tools for derandomization

We show how to derandomize a public-coin IOP based on non-uniform advice (Section 6.1) or based on pseudorandom generators (Section 6.2). In both cases, the verifier's randomness is reduced by having the IOP verifier sample each round's randomness from a smaller set; in Section 6.1 this set is the non-uniform advice, and in Section 6.2 this set is the PRG's seeds.

6.1 Derandomization using non-uniform advice

Theorem 6.1. *Let R be a relation with a public-coin IOP $(\mathbf{P}, \mathbf{V})$ and $\varepsilon = \varepsilon(|\mathbb{x}|) \in (0, 1]$ be a noticeable function. Then Construction 6.2 is a public-coin IOP $(\mathbf{P}', \mathbf{V}')$ for R with parameters related as below.*

IOP $(\mathbf{P}, \mathbf{V})$ for R	
Completeness error	α
Soundness error	β
Rounds	k
Alphabet size	λ
Proof length (per round)	l
Queries	q
Randomness (per round)	r
Verifier running time	vt

$\longrightarrow$

Non-uniform IOP $(\mathbf{P}', \mathbf{V}')$ for R	
Completeness error	$\alpha + \mathsf{k} \cdot \varepsilon$
Soundness error	$\beta + \mathsf{k} \cdot \varepsilon$
Rounds	k
Alphabet size	λ
Proof length (per round)	l
Queries	q
Randomness (per round)	$\Theta\left(\log\left((\mathsf{r} \cdot \mathsf{k} + \|\mathbb{x}\|)/\varepsilon\right)\right)$
Verifier running time	$O(\mathsf{vt} + \mathsf{r} \cdot (\mathsf{r} \cdot \mathsf{k} + \|\mathbb{x}\|)/\varepsilon^2)$
Advice length	$\Theta(\mathsf{r} \cdot (\mathsf{r} \cdot \mathsf{k} + \|\mathbb{x}\|)/\varepsilon^2)$

Moreover, if $\alpha = 0$ then $(\mathbf{P}', \mathbf{V}')$ has completeness error 0. Finally, for every instance $\mathbb{x}$, a uniformly random string of length $\Theta(\mathsf{r} \cdot (\mathsf{r} \cdot \mathsf{k} + |\mathbb{x}|)/\varepsilon^2)$ is good advice with probability at least $1 - 2^{-|\mathbb{x}|}$.

Construction 6.2. On instance $\mathbb{x}$, witness $\mathbb{w}$, and non-uniform advice $Y \subseteq \{0,1\}^{\mathsf{r}}$ where $|Y| = \Theta((\mathsf{r} \cdot \mathsf{k} + |\mathbb{x}|)/\varepsilon^2)$, the IOP $(\mathbf{P}', \mathbf{V}')$ proceeds as follows.

1. For $j = 1, \ldots, \mathsf{k}$:
   (a) $\mathbf{V}'$: Sample a random string $\rho_j \leftarrow Y$ and send it to the prover. This involves sampling and sending $O(\log |Y|)$ bits.
   (b) $\mathbf{P}'$: Given verifier messages $\rho_1, \ldots, \rho_j$, send a message π_j.
2. $\mathbf{V}'$: Given oracle access to $\pi_1, \ldots, \pi_{\mathsf{k}}$, accept if and only if

$$\mathbf{V}^{\pi_1, \ldots, \pi_{\mathsf{k}}}(\mathbb{x}, \rho_1, \ldots, \rho_{\mathsf{k}}) = 1 \ .$$

In order to prove Theorem 6.1, we show a generic lemma that bounds the effect of the sub-sampling on the average value of the interaction tree of a the protocol.

Lemma 6.3. *Let $T_{\mathbb{x}}$ be an interaction tree of IOP $(\mathbf{P}, \mathbf{V})$ on input $\mathbb{x}$. For a set $Y \subseteq \{0,1\}^{\mathsf{r}}$ of size $t = \Theta((\mathsf{r} \cdot \mathsf{k} + |\mathbb{x}|)/\varepsilon^2)$, let $T_{\mathbb{x},Y}$ be the sub-tree of $T_{\mathbb{x}}$ where all verifier messages are chosen only from Y. Then*

$$\Pr_Y \left[|\mathsf{val}(T_{\mathbb{x},Y}) - \mathsf{val}(T_{\mathbb{x}})| \geq \mathsf{k} \cdot \varepsilon \right] \leq 1 - 2^{-|\mathbb{x}|}/3 \ .$$

In particular (by applying the union bound), there exists a set Y such that for every $\mathbb{x}$:

$$|\mathsf{val}(T_{\mathbb{x},Y}) - \mathsf{val}(T_{\mathbb{x}})| \geq \mathsf{k} \cdot \varepsilon \ .$$

First, we use Lemma 6.3 to prove Theorem 6.1, and then prove Lemma 6.3.

Proof of Theorem 6.1. We analyze completeness, soundness, and the efficiency parameters.

Completeness. Fix $(\mathbb{x}, \mathbb{w}) \in R$. Let $T_{\mathbb{x}}$ be the interaction tree of $(\mathbf{P}, \mathbf{V})$ on input $\mathbb{x}$ and $T_{\mathbb{x},\mathsf{G}}$ be the interaction tree of $(\mathbf{P}', \mathbf{V}')$ on input $\mathbb{x}$ and given PRG G. We have $\mathsf{val}(T_{\mathbb{x}}) \geq 1 - \alpha$. By Lemma 6.3:

$$|\mathsf{val}(T_{\mathbb{x}}) - \mathsf{val}(T_{\mathbb{x},\mathsf{G}})| \leq \mathsf{k} \cdot \varepsilon \ ,$$

and so $\mathsf{val}(T_{\mathbb{x},\mathsf{G}}) \geq 1 - (\alpha + \mathsf{k} \cdot \varepsilon)$. It follows that there exists a prover strategy that causes the $\mathbf{V}'$ to accept with probability at least $1 - (\alpha + \mathsf{k} \cdot \varepsilon)$.

Soundness. Fix some instance $\mathbb{x} \notin L(R)$. Let $T_{\mathbb{x}}$ be the interaction tree of $(\mathbf{P}, \mathbf{V})$ on input $\mathbb{x}$ and $T_{\mathbb{x},\mathsf{G}}$ be the interaction tree of $(\mathbf{P}', \mathbf{V}')$ on input $\mathbb{x}$ and given PRG G. We have $\mathsf{val}(T_{\mathbb{x}}) \leq \beta$. By Lemma 6.3:

$$|\mathsf{val}(T_{\mathbb{x}}) - \mathsf{val}(T_{\mathbb{x},\mathsf{G}})| \leq \mathsf{k} \cdot \varepsilon \ ,$$

and so $\mathsf{val}(T_{\mathbb{x},\mathsf{G}}) \leq \beta + \mathsf{k} \cdot \varepsilon$. It follows that there no malicious prover strategy can cause $\mathbf{V}'$ to accept with probability greater than $\beta + \mathsf{k} \cdot \varepsilon$.

Complexity measures. We analyze the efficiency parameters of the resulting IOP:

- *Rounds.* The protocol has k rounds, as in the original IOP $(\mathbf{P}, \mathbf{V})$.
- *Alphabet size and proof length.* The alphabet and proof length of the new IOP are identical to that of the original IOP.
- *Advice length.* The advice $Y \subseteq \{0,1\}^{\mathsf{r}}$ contains $\Theta((\mathsf{r} \cdot \mathsf{k} + |\mathbb{x}|)/\varepsilon^2)$ strings of length r. Therefore the total length of the advice is $\Theta(\mathsf{r} \cdot (\mathsf{r} \cdot \mathsf{k} + |\mathbb{x}|)/\varepsilon^2)$.
- *Queries.* The IOP verifier reads at most q symbols, as in the original IOP $(\mathbf{P}, \mathbf{V})$.
- *Randomness per-round.* The verifier, in each round, samples a random element from Y. Since $|Y| = \Theta((\mathsf{r} \cdot \mathsf{k} + |\mathbb{x}|)/\varepsilon^2)$, the randomness complexity is $\Theta(\log\left(\Theta((\mathsf{r} \cdot \mathsf{k} + |\mathbb{x}|)/\varepsilon^2)\right))$.
- *Verifier running time.* The verifier runs in time $O(\mathsf{vt} + \mathsf{r} \cdot (\mathsf{r} \cdot \mathsf{k} + |\mathbb{x}|)/\varepsilon^2)$ since it must read all of its advice.
- *Adaptivity.* The IOP is non-adaptive if the original IOP was non-adaptive.

□

Proof of Lemma 6.3. First, we prove that, for every node on the tree, if in this node the verifier samples its messages from the set Y (rather than $\{0,1\}^{\mathsf{r}}$) then with high probability (over the choice of Y) the value of the subtree following the node does not change by much.

Claim 6.4. *Let T be an interaction (sub-)tree whose root corresponds to the verifier's turn to send a message, and, for a set $Y \subseteq \{0,1\}^{\mathsf{r}}$ of size t, let T_Y be the tree T when the verifier message corresponding to the root is chosen only from Y. Then:*

$$\Pr_{Y}\left[|\mathsf{val}(T_Y) - \mathsf{val}(T)| \geq \varepsilon\right] \leq 2 \cdot \exp(-2\varepsilon^2 \cdot t) \ .$$

Proof. For $\rho \in \{0,1\}^{\mathsf{r}}$, let $T(\rho)$ be the subtree of T when ρ is chosen as the verifier's message. Then by definition:

$$\mathsf{val}(T) := \mathbb{E}_{\rho \leftarrow \{0,1\}^{\mathsf{r}}}[\mathsf{val}(T(\rho))] \ .$$

Similarly, for every Y: $\mathsf{val}(T_Y) = \mathbb{E}_{\rho \leftarrow Y}[\mathsf{val}(T(\rho))]$. Notice that for every Y, Y' of size t that are identical apart for only one string,

$$|\mathsf{val}(T_Y) - \mathsf{val}(T_{Y'})| \leq 1/t \ ,$$

since the largest difference occurs if $\mathsf{val}(T(\rho)) = 1$ and $\mathsf{val}(T(\rho')) = 0$, where ρ and ρ' are the elements that differ between the sets. We can therefore apply Theorem 3.13 with $f(Y) := \mathsf{val}(T_Y)$ and value $\sigma_j = 1/t$ for every j, noting that $\mathsf{val}(T) := \mathbb{E}_{\rho \leftarrow \{0,1\}^{\mathsf{r}}}[\mathsf{val}(T(\rho))] = \mathbb{E}_Y[f(Y)]$ to get

$$\Pr_Y[|\mathsf{val}(T_Y) - \mathsf{val}(T)| \geq \varepsilon] \leq 2 \cdot \exp(-2\varepsilon^2 \cdot t) \ .$$

□

We now apply Claim 6.4 to the entire tree $T_{\mathbb{x}}$ to turn it into the tree $T_{\mathbb{x},Y}$, showing that with high probability that the value of the tree does not change by much. We do this in a layer-by-layer fashion from the bottom up. For $i \in \{0, \ldots, \mathsf{k}\}$, let T_i be the tree $T_{\mathbb{x}}$ where nodes in layers $1, \ldots, i$ have the verifier sampling uniformly, and nodes in layers $i+1, \ldots, \mathsf{k}$ are changed to subsample from Y.

Claim 6.5. *For every $i \in [\mathsf{k}]$:*

$$\Pr_Y[|\mathsf{val}(T_{i-1}) - \mathsf{val}(T_i)| \geq \varepsilon] \leq 2 \cdot \ell_i \cdot \exp(-2\varepsilon^2 \cdot t) \ ,$$

where ℓ_i is the number of nodes in layer i of T for which the next message is a verifier message.

Proof. By Claim 6.4, switching a single node changes the value of its subtree by a value more than ε with probability at most $2 \cdot \exp(-2\varepsilon^2 \cdot t)$. By applying the union bound over all ℓ_i nodes in layer i, we have that there is no node in layer i whose subtree changes value by more than ε with probability at least $2 \cdot \ell_i \cdot \exp(-2\varepsilon^2 \cdot t)$.

The claim follows by noticing that since all of the nodes that have been changed are in the same layer, the expected value of the tree also has a change of at most ε. □

By applying Claim 6.5 and noticing that $T_0 = T_{\mathbb{x},Y}$ and $T_{\mathsf{k}} = T_{\mathbb{x}}$, we have that:

$$\Pr_Y[|\mathsf{val}(T_{\mathbb{x}}) - \mathsf{val}(T_{\mathbb{x},Y})| \geq \mathsf{k} \cdot \varepsilon] \leq \Pr_Y[\exists i \text{ s.t. } |\mathsf{val}(T_{i-1}) - \mathsf{val}(T_i)| \geq \varepsilon] \tag{3}$$

$$\leq 2 \cdot \exp(-2\varepsilon^2 \cdot t) \cdot \sum_i \ell_i \tag{4}$$

$$\leq 2^{\mathsf{k} \cdot \mathsf{r} + 1} \cdot \exp(-2\varepsilon^2 \cdot t) \ . \tag{5}$$

Equation 3 can be verified by a simple counting argument, since there are k layers. Equation 4 follows by applying Claim 6.5 and the union bound, and Equation 5 follows from the fact that since in every one of the k rounds the verifier sends r random bits, and so the tree T contains at most $2^{\mathsf{k} \cdot \mathsf{r}}$ nodes.

By setting $t = \Theta((\mathsf{r} \cdot \mathsf{k} + |\mathbb{x}|)/\varepsilon^2)$, we have that:

$$\Pr_Y[|\mathsf{val}(T_{\mathbb{x},Y}) - \mathsf{val}(T_{\mathbb{x}})| < \mathsf{k} \cdot \varepsilon] < 1 - 2^{-|\mathbb{x}|}/3 \ .$$

□

6.2 Derandomization using pseudorandom generators

Theorem 6.6. *Let R be a relation with a public-coin IOP $(\mathbf{P}, \mathbf{V})$ with (per round) randomness complexity r, and communication complexity l. Let $\mathsf{G}\colon \{0,1\}^{\ell_{\mathsf{PRG}}} \to \{0,1\}^{\mathsf{r}}$ be a PRG against circuits of size $\mathsf{s}_{\mathsf{PRG}} = \mathrm{poly}(|\mathbb{x}|, \mathsf{k}, \mathsf{l}, \mathsf{r})$ with* PSPACE *gates. Then Construction 6.7 yields an IOP $(\mathbf{P}', \mathbf{V}')$ for R with parameters:*

IOP $(\mathbf{P}, \mathbf{V})$ for R	
Completeness error	α
Soundness error	β
Rounds	k
Alphabet size	λ
Proof length (per round)	l
Queries	q
Randomness (per round)	r
Verifier running time	$\mathsf{vt} = \mathrm{poly}(\lvert\mathbb{x}\rvert, \mathsf{k}, \mathsf{l}, \mathsf{r})$

$+$

PRG G	
Seed length	ℓ_{PRG}
Error	ϵ_{PRG}
Security against size	$\mathrm{poly}(\lvert\mathbb{x}\rvert, \mathsf{k}, \mathsf{l}, \mathsf{r})$
Running time	$\mathsf{t}_{\mathsf{PRG}}$

$\longrightarrow$

IOP $(\mathbf{P}', \mathbf{V}')$ for R	
Completeness error	$1 - ((1-\alpha)/3 + 4\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^2)$
Soundness error	$3\beta + 54\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^3$
Rounds	k
Alphabet size	λ
Proof length (per round)	l
Queries	q
Randomness (per round)	ℓ_{PRG}
Verifier running time	$O(\mathsf{vt} + \mathsf{k} \cdot \mathsf{t}_{\mathsf{PRG}})$

Moreover, if $\alpha = 0$ then $(\mathbf{P}', \mathbf{V}')$ has completeness error 0.

Construction 6.7. Suppose without loss of generality that the honest prover $\mathbf{P}$ maximizes the probability that $\mathbf{V}$ accepts given $\mathbb{x} \in L(R)$. On instance $\mathbb{x}$ and witness $\mathbb{w}$, the protocol $(\mathbf{P}', \mathbf{V}')$ proceeds as follows:

1. For $j = 1, \ldots, \mathsf{k}$:
   (a) $\mathbf{V}'$: Choose a uniformly random string $\rho_j \leftarrow \{0,1\}^{\ell_{\mathsf{PRG}}}$ and send it to the prover.
   (b) $\mathbf{P}'$: Send some message π_j to the verifier.
2. $\mathbf{V}'$: Given oracle access to $\pi_1, \ldots, \pi_{\mathsf{k}}$, accept if and only if $\mathbf{V}^{\pi_1, \ldots, \pi_{\mathsf{k}}}(\mathbb{x}, \mathsf{G}(\rho_1), \ldots, \mathsf{G}(\rho_{\mathsf{k}})) = 1$, querying the oracles $\pi_1, \ldots, \pi_{\mathsf{k}}$ appropriately.

In order to prove Theorem 6.6, we show a generic lemma that bounds the effect of the subsampling on the average value of the interaction tree of a the protocol.

Lemma 6.8. *Let $T_{\mathbb{x}}$ be an interaction tree of IOP $(\mathbf{P}, \mathbf{V})$ on input $\mathbb{x}$. For a PRG G against circuits of size $\mathsf{s}_{\mathsf{PRG}} = \mathrm{poly}(|\mathbb{x}|, \mathsf{k}, \mathsf{l}, \mathsf{r})$ with* PSPACE *gates, let $T_{\mathbb{x},\mathsf{G}}$ be the tree $T_{\mathbb{x}}$ when the verifier messages are chosen only using G. Then*

$$\mathsf{val}(T_{\mathbb{x}})/3 - 4\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^2 \leq \mathsf{val}(T_{\mathbb{x},\mathsf{G}}) \leq 3\mathsf{val}(T_{\mathbb{x}}) + 54\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^3 \ .$$

We first use Lemma 6.8 to prove Theorem 6.6, and later give a proof of Lemma 6.8.

Proof of Theorem 6.6. We analyze completeness, then soundness, and finally the efficiency parameters of the resulting IOP.

Completeness. Fix $(\mathbb{x}, \mathbb{w}) \in R$. Let $T_{\mathbb{x}}$ be the interaction tree of $(\mathbf{P}, \mathbf{V})$ on input $\mathbb{x}$ and $T_{\mathbb{x},\mathsf{G}}$ be the interaction tree of $(\mathbf{P}', \mathbf{V}')$ on input $\mathbb{x}$ and given PRG G. We have $\mathsf{val}(T_{\mathbb{x}}) \geq 1 - \alpha$. By Lemma 6.8:

$$\mathsf{val}(T_{\mathbb{x},\mathsf{G}}) \geq \mathsf{val}(T_{\mathbb{x}})/3 - 4\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^2 \ ,$$

and so $\mathsf{val}(T_{\mathbb{x},\mathsf{G}}) \geq (1-\alpha)/3 + 4\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^2$. It follows that there exists a prover strategy that causes the $\mathbf{V}'$ to accept with probability at least $(1-\alpha)/3 + 4\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^2$.

Soundness. Fix some instance $\mathbb{x} \notin L(R)$. Let $T_{\mathbb{x}}$ be the interaction tree of $(\mathbf{P}, \mathbf{V})$ on input $\mathbb{x}$ and $T_{\mathbb{x},\mathsf{G}}$ be the interaction tree of $(\mathbf{P}', \mathbf{V}')$ on input $\mathbb{x}$ and given PRG G. We have $\mathsf{val}(T_{\mathbb{x}}) \leq \beta$. By Lemma 6.8:

$$\mathsf{val}(T_{\mathbb{x},\mathsf{G}}) \leq 3\mathsf{val}(T_{\mathbb{x}}) + 54\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^3 \ ,$$

and so $\mathsf{val}(T_{\mathbb{x},\mathsf{G}}) \leq 3\beta + 54\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^3$. It follows that there no malicious prover strategy can cause $\mathbf{V}'$ to accept with probability greater than $3\beta + 54\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^3$.

Complexity measures. We analyze the efficiency parameters of the resulting IOP:

- *Rounds.* The protocol has k rounds, as in the original IOP $(\mathbf{P}, \mathbf{V})$.
- *Alphabet size and proof length.* The alphabet and proof length of the new IOP are identical to that of the original IOP.
- *Queries.* The IOP verifier reads at most q symbols, as in the original IOP $(\mathbf{P}, \mathbf{V})$.
- *Randomness per-round.* The verifier, in each round, samples a random seed for G of length ℓ_{PRG}.
- *Verifier running time.* The verifier runs in time $O(\mathsf{vt} + \mathsf{k} \cdot \mathsf{t}_{\mathsf{PRG}})$.
- *Adaptivity.* The IOP is non-adaptive if the original IOP was non-adaptive.

□

Proof of Lemma 6.8. First, we show that, for every node in the tree, if in this node the verifier samples its messages using the PRG then the value of the subtree following the node does not change by much.

Claim 6.9. *Let T' be an interaction (sub-)tree corresponding to an instance $\mathbb{x}$ and transcript prefix tr whose root corresponds to the verifier's turn to send a message. Furthermore, let T'_{G} be the tree T' when the root verifier message is chosen only using the PRG G. Then:*

$$(1 + 1/\mathsf{k})^{-1} \cdot \left(\mathsf{val}(T') - 12\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}\right) \leq \mathsf{val}(T'_{\mathsf{G}}) \leq (1 + 1/\mathsf{k}) \cdot \left(\mathsf{val}(T') + 18 \cdot \epsilon_{\mathsf{PRG}} \cdot \mathsf{k}^2\right) \ .$$

Proof. For the verifier's message ρ denote $T'(\rho)$ be the sub-tree of T' where ρ is chosen as the verifier's first message. Then by definition: $\mathbb{E}_\rho[\mathsf{val}(T'(\rho))] = \mathsf{val}(T')$.

Let $\delta := \frac{1}{3\mathsf{k}}$. We define ranges of values for $\mathsf{val}(T'(\rho))$. Let $m \in \mathbb{N}$ be the minimal integer such that $(1+\delta)^m \cdot \delta \cdot \mathsf{val}(T') \geq 1$. For every $i \in \{0, 1, \ldots, m\}$ let $R_i = (1+\delta)^{-i}$, and set $R_{m+1} = 0$. For every $i \in \{0, 1, \ldots, m\}$, let $p_i = \Pr_\rho[R_i \leq \mathsf{val}(T'(\rho)) \leq R_{i-1}]$. Notice that:

$$\sum_{i=1}^{m+1} p_i \cdot R_i \leq \mathsf{val}(T') \leq \sum_{i=1}^{m+1} p_i \cdot R_{i-1} \ .$$

Additionally, notice that since $R_{i-1} = (1+\delta) \cdot R_i$ for every $i \neq m+1$:

$$\sum_{i=1}^{m+1} p_i \cdot R_i = \sum_{i=1}^{m} p_i \cdot R_i = (1+\delta)^{-1} \cdot \sum_{i=1}^{m} p_i \cdot R_{i-1} \ .$$

By noting that $\sum_{i=1}^{m+1} p_i \cdot R_{i-1} \leq (1+\delta)^{-m} + \sum_{i=1}^{m} p_i \cdot R_{i-1}$, we conclude the following bounds on $\mathsf{val}(T')$:

$$(1+\delta)^{-1} \cdot \sum_{i=1}^{m+1} p_i \cdot R_i - (1+\delta)^{-m-1} \leq \mathsf{val}(T') \leq (1+\delta) \cdot \sum_{i=1}^{m} p_i \cdot R_{i-1} \ . \tag{6}$$

Let C_i be a circuit that has $\mathbb{x}$ and the transcript tr hardwired such that $C_i(\rho) = 1$ if and only if $R_i \leq \mathsf{val}(T'(\rho)) \leq R_{i-1}$. Therefore, $\Pr_\rho[C_i(\rho)] = p_i$. We observe that $\mathsf{val}(T'(\rho))$ can be computed in space $\mathrm{poly}(|\mathbb{x}|, \mathsf{k}, \mathsf{l}, \mathsf{r})$ (see Section 3.2) and so the circuit C_i can be implemented in size $\mathrm{poly}\left(|\mathbb{x}|, \mathsf{k}, \mathsf{l}, \mathsf{r}, \log\left((1+\delta)^i\right)\right) = \mathrm{poly}(|\mathbb{x}|, \mathsf{k}, \mathsf{l}, \mathsf{r})$ with PSPACE gates.

Then, by the pseudo-randomness of the PRG against $\{C_i\}_{i\in[m]}$ we get that for every $i \in [m]$:

$$|\Pr_s[C_i(\mathsf{G}(s)) = 1] - p_i| = |\Pr_s[C_i(\mathsf{G}(s)) = 1] - \Pr_\rho[C_i(\rho) = 1]| \leq \epsilon_{\mathsf{PRG}} \ .$$

Using these definitions we have that:

$$\sum_{i=1}^{m+1} \Pr_s[C_i(\mathsf{G}(s)) = 1] \cdot R_i \leq \mathbb{E}_s[\mathsf{val}(T'(\mathsf{G}(s)))] \leq \sum_{i=1}^{m+1} \Pr_s[C_i(\mathsf{G}(s)) = 1] \cdot R_{i-1} \ .$$

Since $\mathbb{E}_s[\mathsf{val}(T'(\mathsf{G}(s)))] = \mathsf{val}(T'_{\mathsf{G}})$, and by combining the pseudo-randomness of the PRG, we have

$$\sum_{i=1}^{m+1} (p_i - \epsilon_{\mathsf{PRG}}) \cdot R_i \leq \mathsf{val}(T'_{\mathsf{G}}) \leq \sum_{i=1}^{m+1} (p_i + \epsilon_{\mathsf{PRG}}) \cdot R_{i-1} \ .$$

By applying Equation 6, we get:

$$(1+\delta)^{-1} \cdot \mathsf{val}(T') - \epsilon_{\mathsf{PRG}} \cdot \sum_{i=1}^{m+1} R_i \leq \mathsf{val}(T'_{\mathsf{G}}) \leq (1+\delta)^{-m} + (1+\delta) \cdot \mathsf{val}(T') + \epsilon_{\mathsf{PRG}} \cdot \sum_{i=1}^{m+1} R_{i+1} \ .$$

Finally, we note that the sums $\sum_{i\in[m]} R_i = \sum_{i=1}^{m} (1+\delta)^{-i}$ and $\sum_{i\in[m]} R_{i-1} = \sum_{i=1}^{m} (1+\delta)^{-i+1}$ are geometric series bounded from above by $\frac{1+\delta-(1+\delta)^{-m}}{\delta} \leq \frac{2}{\delta}$. Therefore:

$$(1+\delta)^{-1} \cdot \mathsf{val}(T') - \frac{2\epsilon_{\mathsf{PRG}}}{\delta} \leq \mathsf{val}(T'_{\mathsf{G}}) \leq (1+\delta)^{-m} + (1+\delta) \cdot \mathsf{val}(T') + \frac{2\epsilon_{\mathsf{PRG}}}{\delta} \ .$$

We begin by refining the lower-bound:

$$\begin{aligned}
(1+\delta)^{-1} \cdot \mathsf{val}(T') - \frac{2\epsilon_{\mathsf{PRG}}}{\delta} &= (1+\delta)^{-1} \cdot \left(\mathsf{val}(T') - \frac{2\epsilon_{\mathsf{PRG}}(1+\delta)}{\delta}\right) \\
&\geq (1+\delta)^{-1} \cdot \left(\mathsf{val}(T') - \frac{4\epsilon_{\mathsf{PRG}}}{\delta}\right) \\
&\geq (1+1/\mathsf{k})^{-1} \cdot \left(\mathsf{val}(T') - 12\epsilon_{\mathsf{PRG}} \cdot \mathsf{k}\right) \ ,
\end{aligned}$$

where the final equality follows by recalling that $\delta = \frac{1}{3\mathsf{k}}$. We now refine the upper-bound, recalling that $(1+\delta)^m \cdot \delta \cdot \mathsf{val}(T') \geq 1$:

$$\begin{aligned}(1+\delta)^{-m} + (1+\delta)\cdot\mathsf{val}(T') + \frac{2\epsilon_{\mathsf{PRG}}}{\delta} &\leq \delta\cdot\mathsf{val}(T') + (1+\delta)\cdot\mathsf{val}(T') + \frac{2\epsilon_{\mathsf{PRG}}}{\delta} \\ &\leq (1+3\delta)\cdot\left(\mathsf{val}(T') + \frac{2\epsilon_{\mathsf{PRG}}}{\delta^2}\right) \\ &= (1+1/\mathsf{k})\cdot\left(\mathsf{val}(T') + 18\cdot\epsilon_{\mathsf{PRG}}\cdot\mathsf{k}^2\right) \quad .\end{aligned}$$

Putting all of this together, we conclude that

$$(1+1/\mathsf{k})^{-1}\cdot\left(\mathsf{val}(T') - 12\epsilon_{\mathsf{PRG}}\cdot\mathsf{k}\right) \leq \mathsf{val}(T'_{\mathsf{G}}) \leq (1+1/\mathsf{k})\cdot\left(\mathsf{val}(T') + 18\cdot\epsilon_{\mathsf{PRG}}\cdot\mathsf{k}^2\right) \quad .$$

□

We now use Claim 6.9 to show that the value of the interaction tree does not grow significantly over the entire interaction. Let T_i be the tree T_{x} after changing the nodes in layers $i,\ldots,\mathsf{k}$ to subsample from G. Notice that $T_0 = T_{\mathsf{x},\mathsf{G}}$ and $T_{\mathsf{k}} = T_{\mathsf{x}}$. Then:

$$(1+1/\mathsf{k})^{-1}\cdot(\mathsf{val}(T_{i+1}) - 12\epsilon_{\mathsf{PRG}}\cdot\mathsf{k}) \leq \mathsf{val}(T_i) \leq (1+1/\mathsf{k})\cdot\left(\mathsf{val}(T_{i+1}) + 18\cdot\epsilon_{\mathsf{PRG}}\cdot\mathsf{k}^2\right) \quad .$$

This follows since, by Claim 6.9, this is the variation in value of changing a node in layer i to sub-sample from G. Since all the nodes changed are in the same layer, the expected value of the tree also changes by the same value.

We now combine the bounds over all the layers to achieve the required bounds. We begin with the lower-bound:

$$\begin{aligned}\mathsf{val}(T_{\mathsf{x},\mathsf{G}}) &\geq (1+1/\mathsf{k})^{-1}\cdot(\mathsf{val}(T_{\mathsf{k}-1}) - 12\epsilon_{\mathsf{PRG}}\cdot\mathsf{k}) \\ &\geq \cdots \\ &\geq (1+1/\mathsf{k})^{-\mathsf{k}}\cdot(\mathsf{val}(T_0) - \mathsf{k}\cdot 12\epsilon_{\mathsf{PRG}}\cdot\mathsf{k}) \\ &\geq \mathsf{val}(T_{\mathsf{x}})/3 - 4\epsilon_{\mathsf{PRG}}\cdot\mathsf{k}^2 \quad .\end{aligned}$$

Similarly, we give the upper-bound:

$$\begin{aligned}\mathsf{val}(T_{\mathsf{x},\mathsf{G}}) &\leq (1+1/\mathsf{k})\cdot\left(\mathsf{val}(T_{\mathsf{k}-1}) + 18\cdot\epsilon_{\mathsf{PRG}}\cdot\mathsf{k}^2\right) \\ &\leq \cdots \\ &\leq (1+1/\mathsf{k})^{\mathsf{k}}\cdot\left(\mathsf{val}(T_0) + \mathsf{k}\cdot 18\cdot\epsilon_{\mathsf{PRG}}\cdot\mathsf{k}^2\right) \\ &\leq 3\mathsf{val}(T_{\mathsf{x}}) + 54\epsilon_{\mathsf{PRG}}\cdot\mathsf{k}^3 \quad .\end{aligned}$$

Therefore, we conclude that:

$$\mathsf{val}(T_{\mathsf{x}})/3 - 4\epsilon_{\mathsf{PRG}}\cdot\mathsf{k}^2 \leq \mathsf{val}(T_{\mathsf{x},\mathsf{G}}) \leq 3\mathsf{val}(T_{\mathsf{x}}) + 54\epsilon_{\mathsf{PRG}}\cdot\mathsf{k}^3 \quad .$$

□

7 Low-error IOPs to low-error PCPs

We show that IOPs with small soundness error imply PCPs with small soundness error. In particular (under complexity assumptions) in order to construct a sliding-scale PCP for NP, it suffices to construct an IOP for NP, with polynomial round complexity, with similar parameters.

We first show that this can be done using PRGs, meaning that, under complexity assumptions, IOPs can be transformed into PCPs.

Theorem 7.1. *Let R be a relation with a public-coin IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ with (per round) interaction-randomness r. Suppose that there exists a PRG $\mathsf{G}\colon \{0,1\}^{\ell_{\mathsf{PRG}}} \to \{0,1\}^{\mathsf{r}}$ against circuits of size $\mathsf{s}_{\mathsf{PRG}} = \mathrm{poly}(|\mathbb{x}|, \mathsf{k}, \mathsf{l}, \mathsf{r})$ with* PSPACE *gates. Then R has a PCP $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$ with the following parameters:*

IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ for R	
Completeness error	α
Soundness error	β
Rounds	k
Alphabet size	λ
Proof length (per round)	l
Queries	q
Randomness (per round)	r
Verifier running time	vt

$+$

PRG G	
Seed length	ℓ_{PRG}
Error	ϵ_{PRG}
Security against size	$\mathrm{poly}(\lvert\mathbb{x}\rvert, \mathsf{k}, \mathsf{l}, \mathsf{r})$
Running time	$\mathsf{t}_{\mathsf{PRG}}$

$\longrightarrow$

PCP $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$ for R where $\gamma := 1/3 \cdot (1-\alpha) \cdot (\mathsf{q}/(e \cdot \mathsf{k}))^{\mathsf{q}} - 4\epsilon_{\mathsf{PRG}} \cdot \mathsf{q}^2$	
Completeness error	0
Soundness error	$O\left(\left(\beta + \epsilon_{\mathsf{PRG}} \cdot \mathsf{q}^3\right) \cdot \left(\frac{\mathsf{q} \cdot \ell_{\mathsf{PRG}}}{\gamma}\right)\right)$
Alphabet size	$\max\{\lambda, 2^{\mathsf{q} \cdot \ell_{\mathsf{PRG}}}\}$
Length	$O\left(\mathsf{l} \cdot 2^{\mathsf{q} \cdot \ell_{\mathsf{PRG}}} \cdot \mathsf{q} \cdot \ell_{\mathsf{PRG}}/\gamma\right)$
Queries	$\mathsf{q} + 2$
Randomness	$O(\mathsf{q} \cdot \ell_{\mathsf{PRG}})$
Verifier running time	$\mathrm{poly}(\mathsf{vt}, \mathsf{q}, \mathsf{t}_{\mathsf{PRG}}, \log(\frac{\ell_{\mathsf{PRG}}}{\gamma}))$

We now show an analogue of the above theorem in which, rather than use a PRG, the verifier uses non-uniform advice.

Theorem 7.2. *Let R be a relation with a public-coin IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ and let $\varepsilon = \varepsilon(|\mathbb{x}|) \in (0,1]$ be a noticeable function. Then R has a PCP $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$ with the following parameters:*

IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ for R	
Completeness error	α
Soundness error	β
Rounds	k
Alphabet size	λ
Proof length (per round)	l
Queries	q
Randomness (per round)	r
Verifier running time	vt

$\longrightarrow$

PCP $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$ for R	
Completeness error	0
Soundness error	$O\left(\left(\beta + \mathsf{q} \cdot \varepsilon\right) \cdot \left(\frac{\mathsf{q} \cdot \ell}{\gamma}\right)\right)$
Alphabet size	$\max\{\lambda, 2^{\mathsf{q} \cdot \ell_{\mathsf{PRG}}}\}$
Length	$O\left(\mathsf{l} \cdot 2^{\mathsf{q} \cdot \ell} \cdot \mathsf{q} \cdot \ell/\gamma\right)$
Queries	$\mathsf{q} + 2$
Randomness	$O(\mathsf{q} \cdot \ell)$
Verifier running time	$\mathrm{poly}(\mathsf{vt}, \mathsf{k}, \mathsf{r}, 1/\varepsilon, \log(1/\gamma))$
Advice length	$O(\mathsf{q} \cdot \mathsf{k} \cdot \mathsf{r} \cdot \varepsilon^2)$

(Where $\gamma := (1-\alpha) \cdot (\mathsf{q}/(e \cdot \mathsf{k}))^{\mathsf{q}} - \mathsf{q} \cdot \varepsilon$ and $\ell := O\left(\log\left(\frac{\mathsf{q} \cdot \mathsf{r} \cdot \mathsf{k} \cdot |\mathbb{x}|}{\varepsilon}\right)\right) = O(\log(|\mathbb{x}|/\varepsilon))$.)

Corollary 7.3. *Let R be a relation with a public-coin IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ with parameters as below. Then R has an* adaptive *PCP $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$ with the following parameters:*

IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ for R	
Completeness error	0
Soundness error	$1/\lvert\mathbb{x}\rvert$
Rounds	$\text{polylog}(\lvert\mathbb{x}\rvert)$
Alphabet size	$\text{poly}(\lvert\mathbb{x}\rvert)$
Proof length (per round)	$\text{poly}(\lvert\mathbb{x}\rvert)$
Queries	$O(1)$
Randomness (per round)	$\text{poly}(\lvert\mathbb{x}\rvert)$
Verifier running time	$\text{poly}(\lvert\mathbb{x}\rvert)$

$\longrightarrow$

PCP with non-uniform advice $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$ for R	
Completeness error	0
Soundness error	$1/\lvert\mathbb{x}\rvert$
Alphabet size	$\text{poly}(\lvert\mathbb{x}\rvert)$
Length	$\text{poly}(\lvert\mathbb{x}\rvert)$
Queries	$O(1)$
Randomness	$O(\log \lvert\mathbb{x}\rvert)$
Verifier running time	$\text{poly}(\lvert\mathbb{x}\rvert)$
Advice length	$\text{poly}(\lvert\mathbb{x}\rvert)$

Moreover, the non-uniform advice can be removed under the assumption that there exists a function in E *with circuit complexity* $2^{\Omega(n)}$ *for circuits with* PSPACE *gates.*

Proof. We first run amplify the IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ twice so that the soundness error is $1/|\mathbb{x}|^2$, and all other parameters are related by at most a factor of 2. We then apply Theorem 7.2 with $\varepsilon = 1/|\mathbb{x}|^2$. Notice that here $\gamma = 1/\text{polylog}(|\mathbb{x}|)$ and $\ell := O(\log |\mathbb{x}|)$. Therefore, the length of the PCP is equal to $\text{poly}(|\mathbb{x}| \cdot 2^{O(\log |\mathbb{x}|)} \cdot \text{polylog}(|\mathbb{x}|)) = \text{poly}(|\mathbb{x}|)$, and it has soundness error $(1/|\mathbb{x}|^2 + 1/\text{poly}(|\mathbb{x}|)) \cdot \text{polylog}(|\mathbb{x}|) \leq 1/|\mathbb{x}|$.

If we assume that there exists a function in E with circuit complexity $2^{\Omega(n)}$ for circuits with PSPACE gates, then the non-uniform advice can be removed as follows: By Theorem 3.10, there exists a PRG $\mathsf{G}\colon \{0,1\}^{O(\log |\mathbb{x}|)} \to \{0,1\}^{\mathsf{r}}$ against $\text{poly}(|\mathbb{x}|)$-sized circuits with PSPACE-gates and error $1/\text{poly}(|\mathbb{x}|) \leq 1/|\mathbb{x}|^2$ (where $\mathsf{r} = \text{poly}(|\mathbb{x}|)$ is the maximum number of random bits sent by the verifier of the amplified IOP in a round). We can then apply Theorem 7.1 in place of Theorem 7.2. □

Proof of Theorems 7.1 and 7.2. We first prove Theorem 7.1. We apply a sequence of transformations:

1. reduce the number of rounds of the IOP via Lemma 4.3 with $\ell := \mathsf{q}$;
2. derandomize the IOP verifier via Theorem 6.6;
3. transform the IOP to have perfect completeness via Lemma 5.3; and
4. unroll the IOP into a PCP via Lemma 4.6.

This sequence of transformations and their effects on the parameters of the resulting proof systems is described in the list of tables below. The proof of Theorem 7.1 uses an identical series of transformations, except that Theorem 6.6 is replaced with Theorem 6.1 (and as a result, the parameters change slightly).

IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ *for* R	
Completeness error	α
Soundness error	β
Rounds	k
Alphabet size	λ
Proof length (per round)	l
Queries	q
Randomness (per round)	r
Verifier running time	vt

↓ *(Item 1)*

Round-reduced IOP for R via Lemma 4.3 with $\ell := \mathsf{q}$	
Completeness error	$1-(1-\alpha)\cdot(\mathsf{q}/(e\cdot\mathsf{k}))^{\mathsf{q}}$
Soundness error	β
Rounds	q
Alphabet size	λ
Proof length (per round)	l
Queries	q
Randomness (per round)	$\mathsf{k}\cdot\mathsf{r}+\mathsf{q}\cdot\log\mathsf{k}<(\mathsf{q}+1)\cdot\mathsf{k}\cdot\mathsf{r}$
Verifier running time	$\mathrm{poly}(\mathsf{vt})$

↓*(Item 2)*

Derandomized IOP for R via Theorem 6.6 with PRG G with $\gamma := 1/3\cdot(1-\alpha)\cdot(\mathsf{q}/(e\cdot\mathsf{k}))^{\mathsf{q}}-4\epsilon_{\mathsf{PRG}}\cdot\mathsf{q}^2$	
Completeness error	$1-\gamma$
Soundness error	$O(\beta+\varepsilon\cdot\mathsf{q}^3)$
Rounds	q
Alphabet size	λ
Proof length (per round)	l
Queries	q
Randomness (per round)	ℓ_{PRG}
Verifier running time	$\mathrm{poly}(\mathsf{vt},\mathsf{q},\mathsf{t}_{\mathsf{PRG}})$

↓ *(Item 3)*

Perfectly complete IOP for R via Lemma 5.3	
Completeness error	0
Soundness error	$O\left(\left(\beta+\epsilon_{\mathsf{PRG}}\cdot\mathsf{q}^3\right)\cdot\left(\frac{\mathsf{q}\cdot\ell_{\mathsf{PRG}}}{-\log(1-\gamma)}\right)\right)$
Rounds	$\mathsf{q}+1$
Alphabet size	$\max\{\lambda,2^{\mathsf{q}\cdot\ell_{\mathsf{PRG}}}\}$
Proof length (per round)	$O\left(\mathsf{l}\cdot\frac{\mathsf{q}\cdot\ell_{\mathsf{PRG}}}{-\log(1-\gamma)}\right)$
Queries	$\mathsf{q}+2$
Randomness (per round)	ℓ_{PRG}
Verifier running time	$\mathrm{poly}\left(\mathsf{vt},\mathsf{q},\mathsf{t}_{\mathsf{PRG}},\log\left(\frac{\mathsf{q}\cdot\ell_{\mathsf{PRG}}}{-\log(1-\gamma)}\right)\right)$

↓*(Item 4)*

PCP $(\mathbf{P}_{\mathsf{PCP}},\mathbf{V}_{\mathsf{PCP}})$ for R via Lemma 4.6	
Completeness error	0
Soundness error	$O\left(\left(\beta+\epsilon_{\mathsf{PRG}}\cdot\mathsf{q}^3\right)\cdot\left(\frac{\mathsf{q}\cdot\ell_{\mathsf{PRG}}}{-\log(1-\gamma)}\right)\right)$
Alphabet size	$\max\{\lambda,2^{\mathsf{q}\cdot\ell_{\mathsf{PRG}}}\}$
Length	$O\left(\mathsf{l}\cdot 2^{\mathsf{q}\cdot\ell_{\mathsf{PRG}}}\cdot\frac{\mathsf{q}\cdot\ell_{\mathsf{PRG}}}{-\log(1-\gamma)}\right)$
Queries	$\mathsf{q}+2$
Randomness	$\mathsf{q}\cdot\ell_{\mathsf{PRG}}$
Verifier running time	$\mathrm{poly}\left(\mathsf{vt},\mathsf{q},\mathsf{t}_{\mathsf{PRG}},\log\left(\frac{\mathsf{q}\cdot\ell_{\mathsf{PRG}}}{-\log(1-\gamma)}\right)\right)$

Finally, notice that $-\log(1-\gamma)\geq\gamma$ for $0<\gamma<1$, and so $\frac{1}{-\log(1-\gamma)}\leq\frac{1}{\gamma}$. □

8 Limitations of short IOPs

We show that, under the randomized exponential time hypothesis, there are no short IOPs for 3SAT with polylog(n) rounds and small soundness error.

Theorem 8.1. *Assume that* 3SAT $\notin$ BPTIME$[2^{c\cdot n}]$ *for a constant* $c > 0$ *and let* $(\mathbf{P}, \mathbf{V})$ *be a public-coin IOP for* 3SAT *with the following parameters:*

Public-coin IOP $(\mathbf{P}, \mathbf{V})$ for n-variate 3SAT	
Completeness error	0
Soundness error	β
Rounds	polylog(n)
Alphabet size	λ
Proof length (total)	$\mathsf{l}_{\mathsf{tot}}$
Queries	q
Randomness (per round)	poly(n)
Verifier running time	poly(n)

If $\mathsf{l}_{\mathsf{tot}} \geq n$ *and* $\left(\frac{\mathsf{l}_{\mathsf{tot}} \cdot \log \lambda}{n}\right)^{\mathsf{q}} \leq n^{\text{polylog}(n)}$ *then* $\beta > \frac{1}{2} \cdot \left(\frac{2 \cdot e \cdot \mathsf{l}_{\mathsf{tot}} \cdot \log \lambda}{c \cdot n}\right)^{-\mathsf{q}}$.

Proof of Theorem 8.1. We reduce the length of the IOP $(\mathbf{P}, \mathbf{V})$ using Lemma 4.1. Let $m = \mathsf{l}_{\mathsf{tot}}/n$. The parameters of the resulting IOP $(\mathbf{P}', \mathbf{V}')$ are described in the following tables:

IOP $(\mathbf{P}, \mathbf{V})$ *for* n*-variate* 3SAT	
Completeness error	0
Soundness error	β
Rounds	polylog(n)
Alphabet size	λ
Proof length (total)	$n \cdot m$
Queries	q
Randomness (per round)	poly(n)
Verifier running time	poly(n)

$\downarrow$

Length-reduced IOP $(\mathbf{P}', \mathbf{V}')$ *for* n*-variate* 3SAT *via Lemma 4.1 with* $\ell := e \cdot n \cdot m \cdot (2\beta)^{1/\mathsf{q}}$	
Completeness error	$\alpha' := 1 - 2\beta$
Soundness error	$\beta' := \beta$
Rounds	$\mathsf{k}' :=$ polylog(n)
Alphabet size	λ
Proof length (total)	$\mathsf{l}'_{\mathsf{tot}} := e \cdot n \cdot m \cdot (2\beta)^{1/\mathsf{q}}$
Queries	q
Randomness (per round)	poly(n)
Verifier running time	poly(n)

We now apply Lemma 3.2 with the IOP $(\mathbf{P}', \mathbf{V}')$ (viewed as an IP) to construct an algorithm for 3SAT. When considering the IOP with a binary alphabet (by writing each symbol in its binary representation), Lemma 3.2 says that for

$$\begin{aligned} d &= \mathsf{l}'_{\mathsf{tot}} \cdot \log \lambda + \mathsf{k}' \log \mathsf{k}' - \mathsf{k}' \cdot \log\left(1 - \alpha' - \beta'\right) + O(\log n) \\ &= e \cdot n \cdot m \cdot \log \lambda \cdot (2\beta)^{1/\mathsf{q}} + O(\text{polylog}(n) \cdot \log 1/\beta) + o(n) \ , \end{aligned}$$

there exists an algorithm that decides 3SAT in probabilistic time $2^{O(d)}$. Suppose towards contradiction (of RETH) that $\beta \leq \frac{1}{2} \cdot (2 \cdot e \cdot m \cdot \log \lambda / c)^{-\mathfrak{q}}$. Notice that $\frac{1}{2} \cdot (2 \cdot e \cdot m \cdot \log \lambda / c)^{-\mathfrak{q}} \geq n^{-\text{polylog}(n)}$, thus we can assume (without loss of generality) that $\beta \geq n^{-\text{polylog}(n)}$, and, as a result, that $\text{polylog}(n) \cdot \log 1/\beta = o(n)$. Then we have that:

$$\begin{aligned} d &= e \cdot n \cdot m \cdot \log \lambda \cdot (2\beta)^{1/\mathfrak{q}} + o(n) \\ &= c \cdot n/2 + o(n) \\ &< c \cdot n \ . \end{aligned}$$

As a result, we have an algorithm that decides 3SAT in time $2^{c \cdot n}$ in contradiction to the (RETH) assumption that 3SAT $\notin$ BPTIME$[2^{c \cdot n}]$. □

9 Limitations of high-round low-query IOPs

We show that if a relation cannot be decided by IPs with small round complexity, then any round-query IOP with small round-query complexity for the relation cannot have small soundness error.

Theorem 9.1. *Let* k *and* ℓ *be parameters and* R *be a relation where* $R \in \mathrm{AM}[\mathsf{k}] \backslash \mathrm{AM}[\ell]$. *Suppose that* R *has a* k*-round public-coin round-query IOP* $(\mathbf{P}, \mathbf{V})$ *with completeness error* α*, soundness error* β*, and round-query complexity* $\mathsf{q}_{\mathsf{round}} \leq \ell$.

Then $\beta \geq (1-\alpha) \cdot (\ell/(e \cdot \mathsf{k}))^{\mathsf{q}_{\mathsf{round}}} - |\mathbb{x}|^{-c}$ *for every constant* $c > 0$.

Proof of Theorem 9.1. Apply Lemma 4.3 to the k-round IOP $(\mathbf{P}, \mathbf{V})$ with parameter ℓ. This results in a ℓ-round IOP $(\mathbf{P}', \mathbf{V}')$ with completeness error $\alpha' := 1 - (1-\alpha) \cdot (\ell/(e \cdot \mathsf{k}))^{\mathsf{q}_{\mathsf{round}}}$ and soundness error β. Suppose towards contradiction that $\beta < (1-\alpha) \cdot (\ell/(e \cdot \mathsf{k}))^{\mathsf{q}_{\mathsf{round}}} - |\mathbb{x}|^{-c}$ for some $c \in \mathbb{N}$. Then the 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 polynomial $(\mathbf{P}', \mathbf{V}')$ can be transformed into a ℓ-round public-coin IP $(\mathbf{P}'', \mathbf{V}'')$ for R with completeness error $1/3$ and soundness error $1/3$. This is done by using the standard technique of taking $\mathrm{poly}(|\mathbb{x}|)$ 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 $(\mathbf{P}'', \mathbf{V}'')$, then, contradicts the assumption that $R \notin \mathrm{AM}[\ell]$.

□

10 Limitations of binary-alphabet constant-query IOPs

We prove lower bounds on the soundness error of 2-query and 3-query IOPs for 3SAT over the binary alphabet.

Theorem 10.1. *Assume that* RETH *holds and suppose that there exists a non-adaptive public-coin IOP* $(\mathbf{P}, \mathbf{V})$ *for* 3SAT *with the following parameters:*

Non-adaptive public-coin IOP $(\mathbf{P}, \mathbf{V})$ for n-variate 3SAT	
Completeness error	0
Soundness error	β
Rounds	k
Alphabet size	2
Proof length (per round)	$2^{o(n)}$
Queries	q
Randomness (per round)	r
Verifier running time	$2^{o(n)}$

Then:

- *If* $\mathsf{q} = 2$ *then* $\beta > 1 - \varepsilon$ *for every* ε *with* $\mathsf{k} \cdot \log(\mathsf{r} \cdot n/\varepsilon) = o(n)$.
- *If* $\mathsf{q} = 3$ *then* $\beta > 5/8 - \varepsilon$ *for every* ε *with* $\mathsf{k} \cdot \log(\mathsf{r} \cdot n/\varepsilon) = o(n)$.

The theorem follows from a generic lemma that we prove in two steps. In Section 10.1 we formalize the folklore idea that, in certain parameter regimes, it is unlikely that there simultaneously exist CSP solvers and PCPs. In Section 10.2 we build on this and prove an analogous lemma for IOPs.

When compared to the following analogous (folklore) theorem for binary-alphabet PCPs, Theorem 10.1 shows that interaction does not grant additional power when restricted to 2 or 3 queries. The soundness beyond which RETH is contradicted is nearly identical to that of PCPs (in order to contradict ETH). Moreover, the fewer rounds the IOP has, the smaller this gap.

Theorem 10.2 (folklore)**.** *Assume that the* ETH *conjecture holds and suppose that there exists a non-adaptive PCP* $(\mathbf{P}, \mathbf{V})$ *for* 3SAT *with the following parameters:*

Non-adaptive PCP $(\mathbf{P}, \mathbf{V})$ for n-variate 3SAT	
Completeness error	0
Soundness error	β
Alphabet size	2
Length	$2^{o(n)}$
Queries	q
Randomness	$o(n)$
Verifier running time	$2^{o(n)}$

Then:

- *If* $\mathsf{q} = 2$ *then* $\beta = 1$.
- *If* $\mathsf{q} = 3$ *then* $\beta \geq 5/8$.

The proof of Theorem 10.1 uses the fact that there exist efficient algorithms for deciding gap problems for CSPs with arities 2 and 3. This proof is generic in the sense that, given an efficient algorithm for deciding gap problems for CSPs with arity q, our proof gives a lower-bound on the soundness error of any IOP for 3SAT with query complexity q.

10.1 Algorithms for 3SAT from CSP-solvers and PCPs

We prove a generic lemma that states that, for certain parameters, the simultaneous existence of a PCP for 3SAT and an algorithm for solving CSPs implies a fast algorithm for 3SAT.

Lemma 10.3. *Suppose that the following exist:*

- *A non-adaptive PCP* $(\mathbf{P}, \mathbf{V})$ *for* n*-variable* 3SAT *with perfect completeness, soundness error* β*, alphabet* Σ*, proof length* $\mathsf{l} = 2^{o(n)}$ *, query complexity* q*, randomness complexity* $\mathsf{r} = o(n)$*, and verifier running time* $\mathsf{vt} = 2^{o(n)}$ *given* $\ell(n)$ *bits of non-uniform advice.*
- *A deterministic algorithm* $\mathbf{A}$ *that decides in time* $\mathrm{poly}(|\psi|)$ *whether an input* (Σ, q)*-CSP instance* ψ *has value* 1 *or value at most* β*.*

Then there exists an algorithm $\mathbf{A}'$ *that decides whether a* 3SAT *formula* ϕ *over* n *variables is satisfiable in time* $2^{o(n)}$ *given* $\ell(n)$ *bits of non-uniform advice. (Moreover, any advice string that is good for the PCP is also good for the resulting algorithm.)*

Proof. The algorithm $\mathbf{A}'$ receives as input a 3SAT formula ϕ on n variables and a non-uniform advice string $z \in \{0,1\}^{\ell}$, and works as follows.

1. Use Fact 3.5 to transform the PCP $(\mathbf{P}, \mathbf{V})$ on input the 3SAT formula ϕ and advice string z into a (Σ, q)-CSP instance ψ.
2. Run $\mathbf{A}(\psi)$ and output that ϕ is satisfiable if and only if $\mathbf{A}$ outputs that the value of ψ is 1.

We analyze the correctness and running time of the algorithm $\mathbf{A}'$.

- *Correctness.* By Fact 3.5, ψ is a (Σ, q)-CSP with size $\mathrm{poly}(2^{\mathsf{r}}, \mathsf{vt}) = 2^{o(n)}$ where: (i) if $\phi \in$ 3SAT then ψ has value 1; (ii) if $\phi \notin$ 3SAT then ψ has value at most β. By the correctness of the algorithm $\mathbf{A}$ we conclude that the algorithm $\mathbf{A}'$ correctly decides whether ϕ is satisfiable.
- *Running time.* The algorithm $\mathbf{A}'$ runs in time $2^{o(n)}$, because its first step runs in time $\mathrm{poly}(2^{\mathsf{r}}, \mathsf{vt}) = 2^{o(n)}$ and its second step runs in time $\mathrm{poly}(|\psi|) = 2^{o(n)}$.

□

10.2 Algorithms for 3SAT from CSP-solvers and IOPs

We prove a generic lemma that states that, for certain parameters, the simultaneous existence of an IOP for 3SAT and an algorithm for solving CSPs implies a fast *probabilistic* algorithm for 3SAT.

Lemma 10.4. *Let* k*,* r *and* ε *be parameters with* $\mathsf{k} \cdot \log(\mathsf{r}/\varepsilon) = o(n)$*. Suppose that the following exist:*

- *A non-adaptive public-coin IOP* $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ *for* n*-variable* 3SAT *with perfect completeness, soundness error* β*, round complexity* k*, alphabet* Σ*, per-round proof length* $\mathsf{l} = 2^{o(n)}$ *, query complexity* q*, per-round randomness complexity* r*, and verifier running time* $\mathsf{vt} = 2^{o(n)}$*.*
- *A deterministic algorithm* $\mathbf{A}$ *that decides in time* $\mathrm{poly}(|\psi|)$ *whether an input* (Σ, q)*-CSP instance* ψ *has value* 1 *or value at most* $\beta + \varepsilon$*.*

Then there exists a probabilistic algorithm $\mathbf{A}'$ *that decides whether a* 3SAT *formula* ϕ *over* n *variables is satisfiable in time* $2^{o(n)}$ *with probability at least* $2/3$*.*

Proof. First, we transform the IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ into a PCP $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$ for 3SAT as follows: (i) use Theorem 6.1 to derandomize the IOP using non-uniform advice; and (ii) use Lemma 4.6 to unroll the IOP into a PCP. Moreover, in Theorem 6.1 we may sample a random advice string rather than using non-uniformity: a random advice string is good with probability at least $1 - 2^{|\phi|}$. This same advice string can be used in the PCP $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$ derived by applying Lemma 4.6.

Next, we apply Lemma 10.3 with the PCP $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$ and the algorithm $\mathbf{A}$ to derive an algorithm $\mathbf{A}^*$ for 3SAT for whom the same advice strings as the PCP are good. Notice that Lemma 10.3 requires a PCP of length $2^{o(n)}$and that the PCP $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$ resulting from the series of transformations has proof length $2^{O(\mathsf{k}\cdot\log(\mathsf{r}\cdot n/\varepsilon))+o(n)}$. This length is sub-exponential in n since we require $\mathsf{k}\cdot\log(\mathsf{r}\cdot n/\varepsilon) = o(n)$.

Finally, we have that the algorithm $\mathbf{A}'$ that, on input a 3SAT formula ϕ first samples a random advice string z and then outputs the same as $\mathbf{A}^*(\phi, z)$ succeeds in deciding 3SAT with probability at least $1 - 2^{-|\phi|} \geq 2/3$.

In order to keep track of parameters, we describe the transformation as a series of tables :

IOP $(\mathbf{P}_{\mathsf{IOP}}, \mathbf{V}_{\mathsf{IOP}})$ *for* R	
Completeness error	0
Soundness error	β
Rounds	k
Alphabet size	λ
Proof length (per round)	$2^{o(n)}$
Queries	q
Randomness (per round)	r
Verifier running time	$2^{o(n)}$

↓

Derandomized IOP for R *via Theorem 6.1 with error* ε/k	
Completeness error	0
Soundness error	$\beta + \varepsilon$
Rounds	k
Alphabet size	λ
Proof length (per round)	$2^{o(n)}$
Queries	q
Randomness (per round)	$O(\log(\mathsf{r}\cdot n/\varepsilon))$
Verifier running time	$2^{o(n)} + \mathrm{poly}(n, \mathsf{r}, 1/\varepsilon)$
Advice length (chosen at random)	$\mathrm{poly}(n, \mathsf{r}, 1/\varepsilon)$

↓

PCP $(\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}})$ *for* R *via Lemma 4.6*	
Completeness error	0
Soundness error	$\beta + \varepsilon$
Alphabet size	λ
Length	$2^{O(\mathsf{k}\cdot\log(\mathsf{r}\cdot n/\varepsilon))+o(n)}$
Queries	q
Randomness	$O(\log(\mathsf{r}\cdot n/\varepsilon))$
Verifier running time	$2^{o(n)} + \mathrm{poly}(n, \mathsf{r}, 1/\varepsilon)$
Advice length (chosen at random)	$\mathrm{poly}(\mathsf{r}/\varepsilon)$

□

10.3 Proof of Theorems 10.1 and 10.2

We first prove Theorem 10.1. Consider an IOP with parameters as in the statement of Theorem 10.1. Suppose that there exist γ and ε with $\mathsf{k} \cdot \log(\mathsf{r}/\varepsilon) = o(n)$ such that: (i) the IOP soundness error β is at most $\gamma - \varepsilon$; and (ii) there exists an algorithm $\mathbf{A}$ that decides in time $\text{poly}(|\psi|)$ whether an input $(\{0,1\}, \mathsf{q})$-CSP instance ψ has value 1 or value at most γ. Then, by Lemma 10.4, there exists a probabilistic algorithm that decides 3SAT in time $2^{o(n)}$, in contradiction to RETH. Therefore, assuming RETH, the above items cannot both be true.

We conclude the proof of Theorem 10.1 by relying on known algorithms for solving CSPs with appropriate arities q and decision bounds γ, implying that (assuming RETH) it must be that $\beta > \gamma - \varepsilon$ for every ε such that $\mathsf{k} \cdot \log(\mathsf{r}/\varepsilon) = o(n)$.

- For $\mathsf{q} = 2$, we rely on Schaefer's dichotomy theorem (Theorem 3.6), which says that the satisfiability of a $(\{0,1\}, 2)$-CSP can be decided in polynomial time. In this case $\gamma = 1$.
- For $\mathsf{q} = 3$, we rely on Zwick's algorithm (Theorem 3.7), which decides whether a $(\{0,1\}, 3)$-CSP has value 1 or value smaller than $5/8$ in polynomial time. In this case $\gamma = 5/8$.

Finally, the proof of Theorem 10.2 is identical to the one of Theorem 10.1, except that Lemma 10.3 is used in place of Lemma 10.4. In more detail, under ETH, for every γ the following cannot be true simultaneously; (i) the PCP soundness error β is at most γ; and (ii) there exists an algorithm $\mathbf{A}$ that decides in time $\text{poly}(|\psi|)$ whether an input $(2, \mathsf{q})$-CSP instance ψ has value 1 or value at most γ. As in the proof of Theorem 10.1, we rely on Schaefer's dichotomy theorem and Zwick's algorithm to obtain the claimed lower bounds on β.

Acknowledgments

Gal Arnon is 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. Alessandro Chiesa is supported in part by the Ethereum Foundation. Eylon Yogev is 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.

References

[AASY16] Benny Applebaum, Sergei Artemenko, Ronen Shaltiel, and Guang Yang. "Incompressible Functions, Relative-Error Extractors, and the Power of Nondeterministic Reductions". In: *Computational Complexity* 25.2 (2016), pp. 349–418.

[ACY22a] Gal Arnon, Alessandro Chiesa, and Eylon Yogev. "A PCP Theorem for Interactive Proofs". In: *Proceedings of the 41st Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '22. 2022, pp. 64–94.

[ACY22b] Gal Arnon, Alessandro Chiesa, and Eylon Yogev. "Hardness of Approximation for Stochastic Problems via Interactive Oracle Proofs". In: *Proceedings of the 37th Annual IEEE Conference on Computational Complexity*. CCC '22. 2022, 24:1–24:16.

[AG21] Benny Applebaum and Eyal Golombek. "On the Randomness Complexity of Interactive Proofs and Statistical Zero-Knowledge Proofs". In: *Proceedings of the 2nd Conference on Information-Theoretic Cryptography*. ITC '21. 2021, 4:1–4:23.

[AIKS16] Sergei Artemenko, Russell Impagliazzo, Valentine Kabanets, and Ronen Shaltiel. "Pseudorandomness When the Odds are Against You". In: *Proceedings of the 31st Annual Conference on Computational Complexity*. CCC '16. 2016, 9:1–9:35.

[ALMSS98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. "Proof verification and the hardness of approximation problems". In: *Journal of the ACM* 45.3 (1998). Preliminary version in FOCS '92., pp. 501–555.

[AS98] Sanjeev Arora and Shmuel Safra. "Probabilistic checking of proofs: a new characterization of NP". In: *Journal of the ACM* 45.1 (1998). Preliminary version in FOCS '92., pp. 70–122.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. "Fast Reed–Solomon Interactive Oracle Proofs of Proximity". In: *Proceedings of the 45th International Colloquium on Automata, Languages and Programming*. ICALP '18. 2018, 14:1–14:17.

[BCG20] Jonathan Bootle, Alessandro Chiesa, and Jens Groth. "Linear-Time Arguments with Sublinear Verification from Tensor Codes". In: *Proceedings of the 18th Theory of Cryptography Conference*. TCC '20. 2020, pp. 19–46.

[BCGGHJ17] Jonathan Bootle, Andrea Cerulli, Essam Ghadafi, Jens Groth, Mohammad Hajiabadi, and Sune K. Jakobsen. "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*. ASIACRYPT '17. 2017, pp. 336–365.

[BCGRS17] Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. "Interactive Oracle Proofs with Constant Rate and Query Complexity". In: *Proceedings of the 44th International Colloquium on Automata, Languages and Programming*. ICALP '17. 2017, 40:1–40:15.

[BCGV16] Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, and Madars Virza. "Quasilinear-Size Zero Knowledge from Linear-Algebraic PCPs". In: *Proceedings of the 13th Theory of Cryptography Conference*. TCC '16-A. 2016, pp. 33–64.

[BCL22] Jonathan Bootle, Alessandro Chiesa, and Siqi Liu. "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*. EUROCRYPT '22. 2022, pp. 275–304.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. "Interactive Oracle Proofs". In: *Proceedings of the 14th Theory of Cryptography Conference*. TCC '16-B. 2016, pp. 31–60.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. "Checking computations in polylogarithmic time". In: *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*. STOC '91. 1991, pp. 21–32.

[BGLR93] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. "Efficient Probabilistically Checkable Proofs and Applications to Approximations". In: *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*. STOC ?93. 1993, pp. 294–304.

[BN22] Sarah Bordage and Jade Nardi. "Interactive Oracle Proofs of Proximity to Algebraic Geometry Codes". In: *Proceedings of the 37th Annual IEEE Conference on Computational Complexity*. CCC '22. 2022, 30:1–30:45.

[BS08] Eli Ben-Sasson and Madhu Sudan. "Short PCPs with Polylog Query Complexity". In: *SIAM Journal on Computing* 38.2 (2008). Preliminary version appeared in STOC '05., pp. 551–607.

[Bab85] László Babai. "Trading group theory for randomness". In: *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*. STOC '85. 1985, pp. 421–429.

[Ben+17] Eli Ben-Sasson et al. "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*. EUROCRYPT '17. 2017, pp. 551–579.

[CFLS97] Anne Condon, Joan Feigenbaum, Carsten Lund, and Peter W. Shor. "Random Debaters and the Hardness of Approximating Stochastic Functions". In: *SIAM Journal on Computing* 26.2 (1997), pp. 369–400.

[CY20] Alessandro Chiesa and Eylon Yogev. "Barriers for Succinct Arguments in the Random Oracle Model". In: *Proceedings of the 18th Theory of Cryptography Conference*. TCC '20. 2020, pp. 47–76.

[DHK15] Irit Dinur, Prahladh Harsha, and Guy Kindler. "Polynomially Low Error PCPs with polyloglog n Queries via Modular Composition". In: *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*. STOC '15. 2015, pp. 267–276.

[DHMTW14] Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlén. "Exponential time complexity of the permanent and the Tutte polynomial". In: *ACM Transactions on Algorithms* 10.4 (2014), 21:1–21:32.

[Din07] Irit Dinur. "The PCP theorem by gap amplification". In: *Journal of the ACM* 54.3 (2007), p. 12.

[Dru11] Andrew Drucker. "A PCP Characterization of AM". In: *Proceedings of the 38th International Colloquium on Automata, Languages and Programming*. ICALP '11. 2011, pp. 581–592.

[FGLSS91] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. "Approximating clique is almost NP-complete (preliminary version)". In: *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science*. SFCS '91. 1991, pp. 2–12.

[FGLSS96] Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. "Interactive proofs and the hardness of approximating cliques". In: *Journal of the ACM* 43.2 (1996). Preliminary version in FOCS '91., pp. 268–292.

[FGMSZ89] Martin Fürer, Oded Goldreich, Yishay Mansour, Michael Sipser, and Stathis Zachos. "On Completeness and Soundness in Interactive Proof Systems". In: *Advances in Computing Research* 5 (1989), pp. 429–442.

[GH98] Oded Goldreich and Johan Håstad. "On the complexity of interactive proofs with bounded communication". In: *Information Processing Letters* 67.4 (1998), pp. 205–214.

[GLVTW21] Alexander Golovnev, Jonathan Lee, Setty Srinath T. V., Justin Thaler, and Riad S. Wahby. *Brakedown: Linear-time and post-quantum SNARKs for R1CS*. Cryptology ePrint Archive, Report 2021/1043. 2021.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The knowledge complexity of interactive proof systems". In: *SIAM Journal on Computing* 18.1 (1989). Preliminary version appeared in STOC '85., pp. 186–208.

[GVW02] Oded Goldreich, Salil Vadhan, and Avi Wigderson. "On interactive proofs with a laconic prover". In: *Computational Complexity* 11.1/2 (2002), pp. 1–53.

[Has05] Gustav Hast. "Beating a random assignment: Approximating constraint satisfaction problems". PhD thesis. KTH, 2005.

[Hås14] Johan Håstad. "On the NP-hardness of Max-Not-2". In: *SIAM Journal on Computing* 43.1 (2014), pp. 179–193.

[IP01] Russell Impagliazzo and Ramamohan Paturi. "On the Complexity of k-SAT". In: *Journal of Computer and System Sciences* 62.2 (2001), pp. 367–375.

[IW97] Russell Impagliazzo and Avi Wigderson. "P=BPP if E Requires Exponential Circuits: Derandomizing the XOR Lemma". In: *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*. STOC '97. 1997, pp. 220–229.

[KR08] Yael Kalai and Ran Raz. "Interactive PCP". In: *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*. ICALP '08. 2008, pp. 536–547.

[LSTW21] Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. *Linear-time zero-knowledge SNARKs for R1CS*. Cryptology ePrint Archive, Report 2021/30. 2021.

[MNT22] Pasin Manurangsi, Preetum Nakkiran, and Luca Trevisan. "Near-Optimal NP-Hardness of Approximating MAX k-CSPR". In: *Theory of Computing* 18.3 (2022), pp. 1–29.

[NR22] Shafik Nassar and Ron D. Rothblum. "Succinct Interactive Oracle Proofs: Applications and Limitations". In: *Proceedings of the 42nd Annual International Cryptology Conference*. CRYPTO '22. 2022.

[NW94] Noam Nisan and Avi Wigderson. "Hardness vs Randomness". In: *Journal of Computer and System Sciences* 49.2 (1994), pp. 149–167.

[RR20] Noga Ron-Zewi and Ron Rothblum. "Local Proofs Approaching the Witness Length". In: *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science*. FOCS '20. 2020, pp. 846–857.

[RR22] Noga Ron-Zewi and Ron D. Rothblum. "Proving as Fast as Computing: Succinct Arguments with Constant Prover Overhead". In: *Proceedings of the 54th ACM Symposium on the Theory of Computing*. STOC '22. 2022, pp. 1353–1363.

[RRR16] Omer Reingold, Ron Rothblum, and Guy Rothblum. "Constant-Round Interactive Proofs for Delegating Computation". In: *Proceedings of the 48th ACM Symposium on the Theory of Computing*. STOC '16. 2016, pp. 49–62.

[Sch78] Thomas J. Schaefer. "The Complexity of Satisfiability Problems". In: *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*. STOC '78. 1978, pp. 216–226.

[XZZPS19] Tiancheng Xie, Jiaheng Zhang, Yupeng Zhang, Charalampos Papamanthou, and Dawn Song. "Libra: Succinct Zero-Knowledge Proofs with Optimal Prover Computation". In: *Proceedings of the 39th Annual International Cryptology Conference*. CRYPTO '19. 2019, pp. 733–764.

[Zwi98] Uri Zwick. "Approximation Algorithms for Constraint Satisfaction Problems Involving at Most Three Variables per Constraint". In: *Proceedings of the 9th Annual Symposium on Discrete Algorithms*. SODA '98. 1998, pp. 201–210.