Probabilistic Checking Against Non-Signaling Strategies from Linearity Testing

Alessandro Chiesa
UC Berkeley, Berkeley, CA, USA
alexch@berkeley.edu

Peter Manohar
UC Berkeley, Berkeley, CA, USA
manohar@berkeley.edu

Igor Shinkar
Simon Fraser University, Vancouver, Canada
igor.shinkar@sfu.ca

Abstract

Non-signaling strategies are a generalization of quantum strategies that have been studied in physics over the past three decades. Recently, they have found applications in theoretical computer science, including to proving inapproximability results for linear programming and to constructing protocols for delegating computation. A central tool for these applications is probabilistically checkable proofs (PCPs) that are *sound against non-signaling strategies*.

In this paper we prove that the exponential-length constant-query PCP construction due to Arora et al. (JACM 1998) is sound against non-signaling strategies.

Our result offers a new length-vs-query tradeoff when compared to the non-signaling PCP of Kalai, Raz, and Rothblum (STOC 2013 and 2014) and, moreover, may serve as an intermediate step to a proof of a non-signaling analogue of the PCP Theorem.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases probabilistically checkable proofs, linearity testing, non-signaling strategies

Digital Object Identifier 10.4230/LIPIcs.ITCS.2019.25

Related Version Full version is available on the Electronic Colloquium on Computational Complexity as TR18-123, `https://eccc.weizmann.ac.il/report/2018/123/`.

Funding This work was supported by the UC Berkeley Center for Long-Term Cybersecurity.

Acknowledgements We are grateful to Thomas Vidick for suggestions that have improved the presentation in this paper.

1 Introduction

Probabilistically Checkable Proofs (PCPs) [3, 11, 2, 1] are proofs whose validity can be checked by a probabilistic verifier that accesses only a few locations of the proof. PCPs have numerous applications across the theory of computing, including to hardness of approximation [11] and delegation of computation [19, 21].

10th Innovations in Theoretical Computer Science (ITCS 2019).
Editor: Avrim Blum; Article No. 25; pp. 25:1–25:17

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A seminal result, known as the PCP Theorem [2, 1], states that every language in $\mathsf{NTIME}(T)$ can be probabilistically checked by a verifier that uses $O(\log T)$ random bits and makes $O(1)$ queries to a proof of length $\mathsf{poly}(T)$.[1]

In this paper we study PCPs that are sound against *non-signaling strategies* (nsPCPs). These have recently found applications that appear out of the reach of (standard) PCPs, including 1-round delegation of computation from falsifiable assumptions [15, 17] and hardness of approximation for linear programming [16]. The efficiency measures achieved in known nsPCPs appear suboptimal, which affects the quality of the corresponding applications. We thus ask whether a non-signaling analogue of the PCP Theorem holds.

Below we explain the aforementioned notions, and then present our results in this direction.

Non-signaling strategies. Non-signaling strategies are a class of "non-local" correlations that strictly generalize quantum strategies, and capture the minimal condition that spatially-isolated parties cannot communicate instantaneously. They have been studied in physics for over three decades [24, 18, 23] in order to better understand quantum entanglement.

There are two definitions, corresponding to whether the strategy is meant to represent a function or isolated parties; the former is the relevant one for nsPCPs [15, 17].[2] Given a locality parameter $k \in \mathbb{N}$, a *k-non-signaling function* $\mathcal{F}$ extends the notion of a function $f\colon D \to \{0,1\}$ as follows: it is a collection $\{\mathcal{F}_S\}_{S\subseteq D,|S|\leq k}$ where each $\mathcal{F}_S$ is a *distribution* over $\{0,1\}^S$ and, for every two subsets S_1 and S_2 each of size at most k, the restrictions of $\mathcal{F}_{S_1}$ and $\mathcal{F}_{S_2}$ to $S_1 \cap S_2$ are equal as distributions.[3] Note that if $k = |D|$ then $\mathcal{F}$ is a distribution over functions $f\colon D \to \{0,1\}$.

Note that k-non-signaling functions are solutions to the linear program arising from the k-relaxation in the Sherali–Adams hierarchy [25]. The variables are of the form $X_{S,\vec{b}}$ (for all $S \subseteq D$ of size at most k and $\vec{b} \in \{0,1\}^S$) and express the probability of $\vec{b}$ in the distribution $\mathcal{F}_S$; consistency across subsets S and T is expressed using the natural linear constraints.[4]

Non-signaling PCPs. Recall that a classical PCP verifier is given oracle access to a proof represented as a function $f\colon D \to \{0,1\}$. The verifier uses random bits, makes a few queries to f, and then accepts or rejects. Completeness requires that if the statement being checked is true then there is a function f that makes the verifier always accept. Soundness requires that if the statement being checked is false then every function f makes the verifier reject with high probability.

In the non-signaling setting, "proofs" are non-signaling functions rather than (classical) functions. Soundness is correspondingly stronger: given a locality parameter $k \in \mathbb{N}$, soundness requires that every *k-non-signaling function* $\mathcal{F}$ makes the nsPCP verifier reject with high probability.

---

[1] In particular, for every language in $\mathsf{NEXP} = \cup_{c\in\mathbb{N}}\mathsf{NTIME}(2^{n^c})$, the verifier uses $\mathsf{poly}(n)$ random bits and makes $O(1)$ queries to a proof of length $2^{\mathsf{poly}(n)}$.

[2] The other definition underlies the notion of multi-prover interactive proofs that are sound against non-signaling strategies (nsMIPs). Any nsPCP gives rise to an nsMIP with similar parameters. See [15, 17] for details.

[3] A common relaxation of this condition only requires that the marginals $\mathcal{F}_{S_1}|_{S_1\cap S_2}$ and $\mathcal{F}_{S_2}|_{S_1\cap S_2}$ are statistically close, rather than equal; a further relaxation only requires these to be computationally close. While we only consider the standard definition (the marginals must equal) we note that [9] shows that this is almost without loss of generality, as every statistically or computationally non-signaling strategy is close to an (exact) non-signaling strategy.

[4] In fact it suffices to only have variables of the form $X_{S,1^s}$ since all other probabilities can be computed from these.

Efficiency measures of a nsPCP include familiar notions such as proof length (defined as $|D|$) and the verifier's randomness and query complexity. In addition, the locality parameter k controls how hard it is to attain soundness: the smaller k is, the larger the set of non-signaling functions that the verifier could face. (Note that k-non-signaling implies $(k-1)$-non-signaling.)

There is a qualitative difference between the complexity classes captured by PCPs and by nsPCPs; namely, while PCPs capture *non-deterministic* time languages, nsPCPs capture *deterministic* ones. Indeed, the aforementioned PCP Theorem implies that it is NEXP-hard to approximate the maximum acceptance probability of a PCP verifier (that uses polynomial randomness). In contrast, computing the maximum acceptance probability of an nsPCP verifier that uses r random bits reduces to a linear program with $2^{\mathsf{poly}(rk)}$ variables and constraints, a problem solvable in $\mathsf{EXP} = \cup_{c\in\mathbb{N}}\mathsf{DTIME}(2^{n^c})$.

If $k = 2$, the linear program is solvable in PSPACE [13], which is a tight upper bound [14]. For $k > 2$ little is known, except for a seminal result of Kalai, Raz, and Rothblum [15, 17], which shows that for $k = \mathsf{poly}(n)$ it is EXP-hard to approximate a nsPCP verifier's maximum acceptance probability. In more detail, every language in $\mathsf{DTIME}(T)$ has a verifier that uses $\mathsf{poly}(\log T)$ random bits and makes $\mathsf{poly}(\log T)$ queries to a proof of length $\mathsf{poly}(T)$; soundness holds against $\mathsf{poly}(\log T)$-non-signaling functions; the verifier runs in time $n \cdot \mathsf{poly}(\log T)$ and space $\mathsf{poly}(\log T)$.[5]

The nsPCP Conjecture. The nsPCP construction behind the above result is a whitebox modification of early PCP constructions [4, 3], and achieves efficiency similar to those. However, modern "PCP technology" goes well beyond these early constructions, via tools such as proof composition [2] and proofs of proximity [10, 6], and enables better efficiency, including the PCP Theorem. Yet, current "nsPCP technology" is limited to the above results, and the question of whether a non-signaling analogue of the PCP Theorem holds remains open.

▶ **Question 1.** *Is it true that every language in* $\mathsf{DTIME}(T)$ *has an nsPCP verifier that uses* $O(\log T)$ *random bits, makes* $O(1)$ *queries, and is sound against* $O(1)$*-non-signaling functions?*
(As above, we also require that the verifier runs in time $n \cdot \mathsf{poly}(\log T)$ *and space* $\mathsf{poly}(\log T)$*.)*

An affirmative answer to the above question would, e.g., improve the hardness result for linear programming in [16], by yielding a reduction that outputs a linear program of polynomial, rather than a quasipolynomial, size. While we do not know if an affirmative answer exists (and we cannot prove that it does not exist), it is clear that the (very few) tools that we have to construct and analyze nsPCPs are far from this goal. In this paper we make headway towards this goal.

1.1 Towards a nsPCP Theorem

In [1] a key step towards the PCP Theorem is to prove a weaker result in which the proof has *exponential*, rather than *polynomial*, size (and so the randomness complexity of the verifier is polynomial rather than logarithmic). Namely, one proves that every language in $\mathsf{NTIME}(T)$ has a PCP verifier that uses $\mathsf{poly}(T)$ random bits and makes $O(1)$ queries to a proof of length $2^{\mathsf{poly}(T)}$.

[5] Achieving time and space complexities that are $o(T)$ is important for applications. This is not surprising as every language in $\mathsf{DTIME}(T)$ has a trivial nsPCP verifier that runs in time T: the verifier that simply decides the language, without asking any queries. This is unlike the case of PCPs for $\mathsf{NTIME}(T)$, where time complexity is less critical.

In this paper we ask whether a non-signaling analogue of this result holds for the class $\mathsf{DTIME}(T)$.

▶ **Question 2.** *Is it true that every language in* $\mathsf{DTIME}(T)$ *has an nsPCP verifier that uses* $\mathsf{poly}(T)$ *random bits, makes* $O(1)$ *queries, and is sound against* $O(1)$*-non-signaling functions?*

We propose this question as a relaxation that, not only is interesting in its own right, but is likely to shed light on Question 1. However, one must be careful with the precise formulation of Question 2. If the verifier can use $\mathsf{poly}(T)$ random bits then it can simply decide the language by running in time T, without making any queries. To recover a nontrivial question, we *require* that in order to decide whether an instance x is in a language $L \in \mathsf{DTIME}(T)$ the nsPCP verifier first generates queries via a $\mathsf{poly}(T)$-time sampler that is *input oblivious* (knows the length of x but not x itself), and then rules according to a $o(T)$-time decision predicate that knows x. We stress that all PCP/nsPCP verifiers discussed in this paper are input oblivious.

In this paper we study Question 2 by analyzing a natural candidate construction, and ask:

Is the exponential-length $O(1)$-query PCP of [1] sound against $O(1)$-non-signaling functions?

Hereafter, we consider the complexity class $\mathsf{DSIZE}(S)$ (languages decidable by uniform circuits of size $S(n)$) instead of the class $\mathsf{DTIME}(T)$ (languages decidable by machines in time $T(n)$) because our results, like their classical counterparts, are most easily stated in terms of uniform circuits. This change is only for simplicity, as $\mathsf{DTIME}(T) \subseteq \mathsf{DSIZE}(\mathsf{poly}(T))$.

1.2 Main theorem

In this paper we prove that the exponential-length constant-query PCP construction of [1] (without modifications) is sound against non-signaling functions. We obtain the following theorem.

▶ **Theorem 3** (main theorem)**.** *Every language* $L \in \mathsf{DSIZE}(S)$ *has an input-oblivious nsPCP verifier that uses* $O(S^2)$ *random bits, makes* 11 *queries, and is sound against* $O(\log^2 S)$*-non-signaling functions. The query sampler runs in time* $O(S^2)$*, and the decision predicate runs in time* $O(n)$*.*

The theorem is close to answering Question 2, which asks for soundness against $O(1)$-non-signaling functions. (See Table 1 for a comparison with the classical result on nondeterministic languages.) At the same time, some may consider Ito's algorithm [13] as evidence that soundness against $O(1)$-non-signaling functions is too much to hope for. Understanding this gap needs further research.

Our result is *incomparable* to the nsPCP of [15, 17], where the nsPCP verifier uses $\mathsf{poly}(\log S)$ random bits to make $\mathsf{poly}(\log S)$ queries. The fact that we prove soundness only against $O(\log^2 S)$-non-signaling functions (rather than $O(1)$-non-signaling functions) is somewhat undesirable, as this implies that the corresponding nsMIP requires $O(\log^2 S)$ provers. That said, the nsMIP of [15, 17] requires many more provers: $\mathsf{poly}(\log S)$ with the degree in the polynomial much larger than 2. Another feature of our result is that we have "room" to achieve smaller soundness error *without* using additional provers; for example, by asking more queries to the $O(\log^2 S)$ provers, we can achieve a sub-constant soundness error of $2^{-O(\log S)}$.

Finally, our result is the first to demonstrate that a classical PCP construction is secure against non-signaling functions, *without any modifications*. This should be compared to the construction considered in [15, 17] that, while modeled after the PCP in [4, 3], includes several notable modifications that are needed in the soundness proof.

Table 1 The (linear) ALMSS verifier in different PCP settings.

construction	reference	complexity class	type of PCP	soundness error	proof length	randomness	queries	locality
ALMSS verifier	[1]	NSIZE(S)	PCP	$1-1/36$	$2^{O(S^2)}$	$O(S^2)$	11	n/a
+ linearity test	Theorem 3	DSIZE(S)	ns PCP	$1-1/10^7$				$O(\log^2 S)$
ALMSS verifier	[1]	NSIZE(S)	LPCP	3/4	$O(S^2)$	$O(S)$	4	n/a
	Theorem 4	DSIZE(S)	ns LPCP	39/40				$O(\log S)$

1.3 Main lemmas

We outline the ideas behind our theorem in Section 2. Concretely, we highlight several statements, which we deem of independent interest, that we prove on the way to the theorem.

Recall that the exponential-length constant-query PCP in [1] is obtained in two steps. First, construct a constant-query verifier where soundness holds as long as the proof string is a *linear function*; this is known as a *linear PCP*. Second, use a linearity test [8] and self-correction to *compile* this linear PCP into a (standard) PCP, where soundness holds against arbitrary proofs.

Our approach follows the same two steps, but adapted to the non-signaling setting. This also departs from the approach in [17], which does not make use of any property testing results.

Note, however, that it is a priori not clear what is the non-signaling analogue of a linear function. A natural attempt would be to say that a non-signaling function $\mathcal{F}$ is linear iff it passes the BLR linearity test with probability 1 (where the probability is over the test and $\mathcal{F}$). But this attempt is awkward, because the definition depends on a local test, and avoids discussing "global" structure.

A recent work [9] tells us that the right definition is to say that $\mathcal{F}$ is linear iff it corresponds to a *quasi-distribution* over linear functions. A quasi-distribution is a probability distribution where the weights can be any real number and are not restricted to be in $[0,1]$. Quasi-distributions over functions arise in this context because they are an equivalent description of non-signaling functions.

In light of the above, the notion of a *non-signaling linear PCP* (nsLPCP) is immediate: the definition requires soundness to hold against all *linear* non-signaling functions.

The first step in our proof is showing that the linear PCP verifier of [1] (the "ALMSS verifier"), when used for deterministic computations, is sound against linear non-signaling functions.

▶ **Theorem 4.** *The (input oblivious) ALMSS verifier, for a given language $L \in \mathsf{DSIZE}(S)$, uses $O(S)$ random bits, makes 4 queries, and is sound against linear $O(\log S)$-non-signaling functions.*

See Table 1 for a comparison with the classical result showing soundness against linear functions.

In order to "lift" Theorem 4 to Theorem 3, we need a suitable linearity test.

The linearity test of [8] was recently analyzed in the non-signaling setting by [9], who proved that any k-non-signaling function $\mathcal{F}$ that passes the linearity test with probability $1-\varepsilon$ can be self-corrected to a $\lfloor k/2 \rfloor$-non-signaling function $\hat{\mathcal{F}}$ that is $2^{O(k)}\varepsilon$-close to a linear $\lfloor k/2 \rfloor$-non-signaling function $\mathcal{L}$. (Self-correction and closeness have precise meanings, discussed later.) However, we cannot directly use [9]'s result, because in our theorem the locality parameter k is required to be super-constant ($k = O(\log S)$ in Theorem 4), and thus

the bound on the distance between $\hat{\mathcal{F}}$ and $\mathcal{L}$ is too large, even when considering only query sets of small size. Specifically, we need the distance to be a sufficiently small constant on query sets of size 4 (the number of queries in Theorem 4).

We solve this problem by extending the result in [9] in a black-box way and proving that the distance between $\hat{\mathcal{F}}$ and $\mathcal{L}$ on a query set Q is only $O(|Q|\sqrt{\varepsilon})$, provided that the error ε and $\mathcal{L}$'s locality are sufficiently small. Crucially, if $|Q|$ is constant, so is the distance between $\hat{\mathcal{F}}$ and $\mathcal{L}$. The proof of this statement involves analyzing the *repeated* linearity test, whose behavior in the non-signaling setting is quite subtle when compared to the classical setting (see Section 2.6).

▶ **Theorem 5.** *Let $k, \bar{k}\mathbb{N}$ and $\varepsilon \in (0, 1/400]$ be such that $k = \Omega((\bar{k} + \log \frac{1}{\varepsilon}) \cdot \bar{k})$. If a k-non-signaling function $\mathcal{F}\colon \{0,1\}^n \to \{0,1\}$ passes the linearity test with probability at least $1 - \varepsilon$ then there exists a linear $\bar{k}$-non-signaling function $\mathcal{L}\colon \{0,1\}^n \to \{0,1\}$ such that for all query sets $Q \subseteq \{0,1\}^n$ with size $|Q| \leq \bar{k}$ and for all events $E \subseteq \{0,1\}^Q$ it holds that*

$$\left|\Pr[\hat{\mathcal{F}}(Q) \in E] - \Pr[\mathcal{L}(Q) \in E]\right| \leq O(|Q|\sqrt{\varepsilon}) \ .$$

The above result on linearity testing enables us to transform our nsLPCP, and more generally *any* nsLPCP, into a corresponding nsPCP with minimal changes in parameters (the transformation is exactly the classical compiler). This is the last key statement in the proof of our main theorem.

▶ **Lemma 6.** *For every $\varepsilon \in [0,1]$, if a language L has an nsLPCP where the verifier uses r random bits, makes q queries, and has soundness error $1 - \varepsilon$ against linear k-non-signaling functions $\mathcal{L}\colon \{0,1\}^\ell \to \{0,1\}$, then L has an nsPCP where the verifier uses $r + O(q\ell)$ random bits, makes $O(q)$ queries, and has soundness error $1 - O_q(\varepsilon^2)$ against $O_\varepsilon(k^2)$-non-signaling functions $\mathcal{F}\colon \{0,1\}^\ell \to \{0,1\}$. (Furthermore, if the former is input oblivious, so is the latter.)*

1.4 Enriching the toolkit for non-signaling PCPs

Progress in our understanding of PCPs has typically moved hand in hand with progress in our understanding of low-degree testing. In particular, many PCP constructions follow this blueprint:

(1) a low-degree test that, via only a few queries, ensures that a given proof conforms to a specified algebraic encoding;

(2) a probabilistic test that, assuming the proof is (essentially) given in this encoding, ensures that the statement being checked is true with high probability.

In contrast, while the nsPCP in [17] is reminiscent of this blueprint, its analysis does not follow it, despite the fact that the construction is modeled after the PCP in [4, 3], for which the two-step analysis *is* possible (in the classical setting). The lack of such general paradigms means that we lack general design principles to construct better nsPCPs.

This state of affairs raises the intriguing question of whether a theory of low-degree testing (and, more generally, property testing) is feasible in the non-signaling setting and, moreover, whether one can build on it to construct nsPCPs in order to make further progress towards Question 1.

An additional contribution of our work is to *enrich* the current "non-signaling toolkit", by demonstrating an example where the aforementioned blueprint is both possible and useful.

Namely, building on the work of [9] on linearity testing, our results provide a modular paradigm that not only simplifies the overall analysis, thereby enabling us to assert that the construction of [1] *with no modifications* is sound against non-signaling strategies, but also (as discussed later) clarifies the technical barriers that separate us from answering Question 2. All this suggests that our techniques will be helpful for constructing more efficient nsPCPs.

1.5 Concurrent work

In a concurrent work, Kiyoshima [20] studies the soundness of ALMSS-type PCPs against non-signaling strategies. Kiyoshima proves that, for a sufficiently large security parameter t (at least logarithmic in the circuit size), the t-repetition of a $O(t)$-query modification of the ALMSS-verifier has soundness error $\mathrm{negl}(t)$ against $O(t^2)$-non-signaling functions. In comparison, we prove that the *unmodified* 11-query ALMSS PCP has soundness error $O(1)$ against $O(\log^2 S)$-non-signaling strategies (and also that a modification of its t-repetition has soundness error $\exp(-t)$ for every $t = \Omega(\log S)$). While both our analysis and Kiyoshima's analysis avoid the use of an augmented circuit (necessarily so as it would have had exponential size), our techniques differ. Kiyoshima conducts a direct analysis of the PCP verifier, while we adopt a modular approach in which we first prove soundness against *linear* non-signaling strategies (a simpler task), and then, building on a recent analysis of the linearity test [9], we deduce soundness against *all* non-signaling strategies. We consider the modular and simple analysis in our work to be of independent interest. Kiyoshima additionally proves that soundness holds against *computational* non-signaling strategies, a relaxation where the marginal distributions on intersections are only required to be computationally close. Our results directly extend to computational non-signaling strategies as every computational non-signaling strategy is close to an exact non-signaling strategy (as proved in [9]).

In another concurrent work, Holmgren and Rothblum [12] study the problem of constructing PCPs/MIPs in which the prover is very efficient in time and space [7], in the non-signaling setting. While they consider a construction that is more closely related to the PCP in [4, 3] (honest proofs are encoded via low-degree polynomials rather than linear functions), their soundness analysis also has the feature that it avoids the use of an augmented circuit.

1.6 Open problems

The question of whether the exponential-length constant-query PCP of [1] is sound against $O(1)$-non-signaling functions remains open. A concrete approach to affirmatively answer this question is to prove that the *linear* PCP verifier of [1] is sound against k-non-signaling functions for $k = O(1)$, rather than $k = O(\log S)$ as in Theorem 4. (Our generic compiler from Theorem 6 would then take care of the rest.) Another intriguing possibility is that an affirmative answer to Question 2 could come from a *different* exponential-size constant-query PCP. However, the result due to [13] shows that the class of nsMIPs with 2 provers equals PSPACE, which possibly suggests that soundness against $O(1)$-non-signaling functions is too much to hope for.

Moreover, while our results can be interpreted as progress towards a non-signaling analogue of the PCP Theorem (Question 1), it remains unclear whether such an analogue holds, and more investigations in nsPCPs are needed. We believe that our work and our new techniques can inform such investigations.

2 Techniques

We outline the techniques used to prove our results. First, in Section 2.1, we explain the transformation from a nsLPCP to a corresponding nsPCP. Next, in Sections 2.2 to 2.5 we discuss the nsLPCP on which we apply this transformation, namely, the ALMSS verifier [1]. Finally, in Section 2.6, we discuss linearity testing with low error, which underlies the transformation.

2.1 From nsLPCP to nsPCP

We discuss the transformation from nsLPCP to nsPCP (Theorem 6). We first recall the classical transformation from LPCP to PCP, and then explain how to achieve its non-signaling analogue.

The classical case. The classical transformation from LPCP to PCP relies on the following tools.

- *Testing linearity.* Given a boolean function $f\colon \{0,1\}^\ell \to \{0,1\}$, the linearity test draws random $x, y \in \{0,1\}^\ell$ and checks that $f(x) + f(y) = f(x+y)$ [8]. If the test passes with probability $1-\varepsilon$, then f is ε-close to a linear function $f^*\colon \{0,1\}^\ell \to \{0,1\}$ [8, 5].
- *Self-correction.* Given f that is ε-close to a linear function f^*, one can create a probabilistic oracle $\mathcal{O}$ that, given *any* $x \in \{0,1\}^\ell$, returns $f^*(x)$ with probability $1-2\varepsilon$. Namely, $\mathcal{O}$ samples a random $z \in \{0,1\}^\ell$, queries f on $z+x$ and z, and answers with $f(z+x) - f(z)$.

The above tools imply a transformation from LPCP to PCP: given access to an arbitrary function $f\colon \{0,1\}^\ell \to \{0,1\}$, the PCP verifier runs the linearity test and then runs the LPCP verifier by self-correcting each of its queries. If the LPCP verifier makes q queries and has soundness error γ, then the resulting PCP verifier makes $3+2q$ queries and has soundness error $\max\{1-\varepsilon, \gamma + 2q\varepsilon\}$, where ε is (a bound on) the distance of f to linear functions. This soundness error is bounded by $1 - \frac{1-\gamma}{2q+1}$ (the maximum is when the two terms equal), which is bounded away from 1.

If desired, the soundness error can be made arbitrarily close to γ by repeating the linearity test. Given a parameter t, the repeated linearity test samples $x_i, y_i \in \{0,1\}^\ell$ for each $i \in [t]$ and checks that $f(x_i) + f(y_i) = f(x_i + y_i)$ for all $i \in [t]$. Now, the PCP verifier makes $3t + 2q$ queries and has soundness error $\max\{(1-\varepsilon)^t, \gamma + 2q\varepsilon\}$, which for suitable ε and $t = O_{\gamma,\varepsilon}(q)$ is arbitrarily close to γ.

The non-signaling case. We follow the structure of the classical transformation. However, the non-signaling case not only calls for a different analysis but also raises a problem that we must solve.

The linearity test in the non-signaling setting has the following guarantee [9]: if $\mathcal{F}$ is a k-non-signaling function such that $\Pr_{x,y,\mathcal{F}}[\mathcal{F}(x) + \mathcal{F}(y) = \mathcal{F}(x+y)] \geq 1-\varepsilon$ then $\mathcal{F}$ can be self-corrected (in the natural way) to a $(k/2)$-non-signaling function $\hat{\mathcal{F}}$ that is $2^{O(k)}\varepsilon$-close to a *linear* non-signaling function $\mathcal{L}$. Note that self-correction is already part of the conclusion.

The above result appears sufficient for compiling a nsLPCP verifier into a corresponding nsPCP verifier. Namely, given a k-non-signaling function $\mathcal{F}\colon \{0,1\}^\ell \to \{0,1\}$, the nsPCP verifier checks that $\mathcal{F}(x) + \mathcal{F}(y) = \mathcal{F}(x+y)$ for random $x, y \in \{0,1\}^\ell$ and also checks that the nsLPCP verifier accepts $\hat{\mathcal{F}}$. Analogously to before, if the nsLPCP verifier makes q queries and has soundness error γ against *linear* $(\frac{k-3}{2})$-non-signaling functions, then the resulting PCP verifier makes $3+2q$ queries and has soundness error $\max\{1-\varepsilon, \gamma + 2^{O(k)}\varepsilon\}$ against *arbitrary* k-non-signaling functions.

However, our analysis of the ALMSS verifier (the nsLPCP that we use) will require locality $k = \Omega(\log N)$, which means that the additive term $2^{O(k)}\varepsilon$ grows with N. This precludes achieving a constant soundness error with constant query complexity.

The foregoing motivates the problem of testing linearity of non-signaling functions *with low error*: how do we ensure that $\hat{\mathcal{F}}$ is sufficiently close to a linear non-signaling function $\mathcal{L}$? We stress that while in the classical case improving the "quality" of the self-correction has a straightforward solution (repeat the linearity test, and do self-correction), in the non-signaling case this problem is quite involved. Moreover, *we do not wish to modify in any way the classical compiler*, and thus relying on additional queries (even if only a constant number depending on q and ε) is not an option.

We discuss our solution to this problem later on in Section 2.6, thereby providing the missing ingredient of our compiler from nsLPCP to nsPCP. In the meantime, in Sections 2.2 to 2.5, we discuss how we prove that the ALMSS verifier is secure against linear non-signaling functions.

2.2 The linear ALMSS verifier against linear non-signaling functions

Our goal is to establish that the linear PCP verifier of [1] (the "ALMSS verifier") is sound against linear non-signaling functions, and thus prove that every language $L \in \mathsf{DSIZE}(S)$ has a constant-query nsLPCP verifier that is sound against linear $O(\log S)$-non-signaling functions. Note that we invoke the ALMSS verifier on *deterministic* (DSIZE) computations, rather than on *nondeterministic* (NSIZE) computations as in the classical case. We now recall the ALMSS verifier.

Let $L \in \mathsf{DSIZE}(S)$ be a language, and let $\{C_n\}_{n\mathbb{N}}$ be a uniform boolean circuit family of size $N := S(n)$ that decides L (for all $x \in \{0,1\}^n$, $x \in L$ iff $C_n(x) = 1$). Hereafter we omit the subscript in C_n as it is clear from context. Given an input x, one can express the condition "$C(x) = 1$" as a system of simple equations over C's wires W; the variables are $\mathbf{w} = (w_1, \ldots, w_N)$, one per wire. We use the convention that the input wires are $w_1, \ldots, w_n$ and the output wire is w_N. To ensure input consistency we need that $w_j = x_j$ for every $j \in \{1, \ldots, n\}$; to ensure correct gate computations we need that, for every $j \in \{n+1, \ldots, N\}$, w_j is the correct combination of the variables used to compute it (e.g., if w_j is the output of an AND gate with inputs w_{j_1} and w_{j_2} then the equation is $w_j = w_{j_1} \cdot w_{j_2}$); to ensure that the output is 1 we need that $w_N = 1$. This can be summarized as a system of $M := N + 1$ equations $\{P_j(\mathbf{w}) = c_j\}_{j\in[M]}$, where $P_1, \ldots, P_M$ are quadratic polynomials (each involving at most three variables in $\mathbf{w}$) and $c_1, \ldots, c_M$ are boolean constants.

The ALMSS verifier is given below. We overload notation and use P_j to also denote the upper triangular matrix in $\{0,1\}^{N^2}$ such that $P_j(\mathbf{w}) = \langle P_j, \mathbf{w} \otimes \mathbf{w}\rangle$; that is, if $P_j(\mathbf{w}) = \sum_{i=1}^{N} a_i w_i + \sum_{1\le i<i'\le N} a_{i,i'} w_i w_{i'}$, then P_j has a_i in the diagonal entry (i,i) and $a_{i,i'}$ in the entry (i,i'), for $1 \le i < i' \le N$. Also, for $a \in \{0,1\}^N$, D_a is the diagonal matrix in $\{0,1\}^{N^2}$ whose diagonal is a.

The ALMSS verifier, given input $x \in \{0,1\}^n$ and oracle access to a linear non-signaling function $\mathcal{L}\colon \{0,1\}^{N^2} \to \{0,1\}$, works as follows:

1. Use the circuit C and input x to construct the matrices $P_1, \ldots, P_M \in \{0,1\}^{N^2}$ and constants $c_1, \ldots, c_M \in \{0,1\}$, which represent the computation of C on x.
2. Draw random $s \in \{0,1\}^M$, $u, v, \in \{0,1\}^N$, and query $\mathcal{L}$ on the set $\{\sum_{j=1}^{M} s_j P_j, D_u, D_v, u \otimes v\}$.
3. Check that $\mathcal{L}(\sum_{j=1}^{M} s_j P_j) = \sum_{j=1}^{M} s_j c_j$ and that $\mathcal{L}(D_u)\mathcal{L}(D_v) = \mathcal{L}(u \otimes v)$.

If $C(x) = 1$, the honest proof is the linear function $\pi\colon \{0,1\}^{N^2} \to \{0,1\}$ where $\pi(Z) := \langle \mathbf{w} \otimes \mathbf{w}, Z\rangle = \sum_{i,i'\in[N]} w_i w_{i'} \cdot Z_{i,i'}$ where w_i is now the value of the i-th wire in the computation of C on x.

The challenge is to prove that the ALMSS verifier is sound against *linear non-signaling functions*. Namely, we must show that if there is a linear non-signaling function $\mathcal{L}$ that is accepted with good probability then $x \in L$, or equivalently that $C(x) = 1$. We discuss this in the next sub-sections.

2.3 A linear local assignment generator suffices

The first step in our soundness analysis shows that, to establish that $C(x) = 1$, it suffices to construct a *linear local assignment generator* with sufficiently small error.

A linear k-local assignment generator for (C, x) with error ε is a linear k-non-signaling function $\mathcal{A}\colon \{0,1\}^N \to \{0,1\}$ that individually satisfies each of the M constraints with probability $1 - \varepsilon$ (over the randomness of $\mathcal{A}$). Namely,

(a) for each $i \in \{1, \ldots, n\}$, $\Pr[\mathcal{A}(e_i) = x_i] \geq 1 - \varepsilon$;

(b) for each $i \in \{n+1, \ldots, N\}$, if w_i is the output of a unary gate g with input w_j then $\Pr[\mathcal{A}(e_i) = g(\mathcal{A}(e_j))] \geq 1 - \varepsilon$, else if w_i is the output of a binary gate g with inputs w_{j_1}, w_{j_2} then $\Pr[\mathcal{A}(e_i) = g(\mathcal{A}(e_{j_1}), \mathcal{A}(e_{j_2}))] \geq 1 - \varepsilon$;

(c) $\Pr[\mathcal{A}(e_N) = 1] \geq 1 - \varepsilon$.

(Here e_i is the i-th vector in the standard basis.)

▶ **Lemma 7** (informal). *If there exists a k-local assignment generator for (C, x) with error ε for $k = \Omega(\log N)$ and $\varepsilon = O(\frac{1}{N \log N})$, then $C(x) = 1$.*

We sketch the proof of this lemma. The transcript of the computation of C on x is the *unique* correct assignment to all the wires. We say that a wire $w_i \in W$ of C is *correct* whenever $\mathcal{A}(e_i)$ equals the value contained in this transcript; more generally, we say that a vector $z \in \{0,1\}^N$ is correct if $\mathcal{A}(z)$ equals the value of z in the linear extension of the transcript. Below, we partition C's wires W into layers $W_1, \ldots, W_H$ according to depth. (We assume layered circuits.)

As a warmup, suppose for now that $k \geq N$. The probability that all wires in W_1 are correct is at least $1 - |W_1|\,\varepsilon$, and the probability that all the gates are correct is at least $1 - \sum_{h=2}^{H} |W_h|\,\varepsilon$. Therefore by union bound, the probability that all wires in the circuit are correct is at least $1 - \sum_{h=1}^{H} |W_h|\,\varepsilon$, because if all the input wires are correct and all the gates are computed correctly, then all the wires in the circuit are correct. In particular, we deduce that the output wire is correct with probability $1 - \sum_{h=1}^{H} |W_h|\,\varepsilon = 1 - |W|\,\varepsilon = 1 - N\varepsilon$. Since the output wire is 1 with probability $1 - \varepsilon$, and $\varepsilon = O(\frac{1}{N})$, we conclude that $\Pr[C(x) = 1] > 0$, and thus $C(x) = 1$.

The above argument requires that $k \geq N$, because we have to simultaneously "view" assignments to all wires in the circuit. While the argument can be easily modified so that we only require k to be at least twice the width of C, the latter may still be much larger than $O(\log N)$.

Using the linearity of $\mathcal{A}$, however, we can modify the argument to merely require $k = \Omega(\log N)$. For each layer h, we define an event E_h such that if E_h holds, then any wire in layer h is correct with high probability. In the warmup above E_h is the event "all wires in layer $h - 1$ are correct"; in our proof E_h is the event "t random linear combinations of wires in layer h are correct". Given a wire w_i in layer h, we can bound the event "$\mathcal{A}(e_i)$ is incorrect and E_h holds" as follows. If $\mathcal{A}(e_i)$ is incorrect, then all linear combinations of wires in layer h can be split into pairs z and $z + e_i$, and exactly one of $\mathcal{A}(z)$ and $\mathcal{A}(z + e_i)$ is incorrect. Hence, the probability that a random linear combination of wires in layer h is correct, given that $\mathcal{A}(e_i)$ is incorrect, is at most $1/2$, and so $\Pr[E_h \mid \mathcal{A}(e_i) \text{ is incorrect}] \leq 2^{-t}$, since the t random linear combinations are independent. Using Bayes's rule (and an additional assumption that $\Pr[E_h] \geq 1/2$), we deduce that $\Pr[\mathcal{A}(e_i) \text{ is incorrect} \mid E_h]$ is small. We then proceed inductively on the layers as before.

The argument above requires that $\varepsilon = O(\frac{1}{N \log N})$. One may wonder whether a similar result could be proved with, say, $\varepsilon = O(1)$. We additionally prove that our analysis is almost tight, in that an error of $\varepsilon = O(\frac{\log N}{N})$ is necessary, *regardless* of how large the locality k is.

See the full version of this paper for details.

Local assignment generators in prior works. Local assignment generators appear in prior works on nsPCPs [17, 22], but our notion is qualitatively different, as we now explain.

Prior works consider local assignment generators for an *augmented* circuit C_{aug} rather than for C itself. (See Section 1.5 for a discussion of concurrent work that avoids augmented circuits.) Informally, C_{aug} not only contains C as a sub-circuit but also low-degree extensions of C's layers as well as subcircuits computing all low-degree tests on these. The wires contained in these additional subcircuits are what enables defining an event E_h on which to condition for each layer.

The analogue of the augmented circuit C_{aug} in our setting, however, has *exponential* size, and thus *we cannot use it*. Namely, we would have to encode each layer of C via the Hadamard code (all linear combinations of wires in the layer) and then compute all possible linear tests on these.

Instead, our assumption that the local assignment generator is a *linear* non-signaling function implies that we *do not have to construct an augmented circuit*. Namely, the linear combinations that we use to define the event E_h are implicitly available due this linearity, and so there is no need to augment C (nor, in particular, to introduce any gates that evaluate linearity tests).

The assumption that the local assignment generator is linear is justified by the fact that a different part of our construction (the linearity test in our generic compiler) ensures the non-signaling function is (close to) linear. Overall, this separation not only avoids the aforementioned issues of using augmented circuits, but also simplifies the analysis of the local assignment generator.

2.4 Constructing the linear local assignment generator

Given a k-non-signaling function $\mathcal{L}\colon \{0,1\}^{N^2} \to \{0,1\}$ that is accepted by the ALMSS verifier with probability at least $1-\varepsilon$, we can obtain a linear k-local assignment generator $\mathcal{A}\colon \{0,1\}^N \to \{0,1\}$ with error $O(\varepsilon)$ by "restricting $\mathcal{L}$ to its diagonal". Namely, in order to query $\mathcal{A}$ at $v \in \{0,1\}^N$, we query $\mathcal{L}$ at $D_v \in \{0,1\}^{N^2}$, where D_v is the diagonal matrix that has v as its diagonal.

We show that, since $\mathcal{L}$ is accepted with probability at least $1-\varepsilon$, $\mathcal{L}$ must satisfy any *individual* constraint $P_j(\mathbf{w}) = c_j$ with probability at least $1-O(\varepsilon)$, and this directly implies that the linear local assignment generator $\mathcal{A}$ has error $O(\varepsilon)$. (See the full version of this paper for details.)

The discussion so far already gives us a weak bound on the soundness error of the ALMSS verifier, namely $1-O(\frac{1}{N \log N})$. Indeed, for $k = O(\log N)$ and $\varepsilon = O(\frac{1}{N \log N})$, we can apply the lemma above (in Section 2.3) to conclude that $C(x) = 1$.

However, our goal is to show that the ALMSS verifier (as is) has *constant* soundness error, and doing so requires more technical work, which we discuss next.

▶ Remark. We stress that proving a soundness error of even $1-O(\frac{1}{N \log N})$ is a non-trivial statement. This is in contrast to the classical setting, where if an assignment satisfies an $1-\varepsilon$ fraction of the $M = N+1$ constraints for $\varepsilon < 1/M$, then, trivially, *all* constraints are satisfied.

2.5 The ALMSS verifier has constant soundness error

Our goal is to prove that the ALMSS verifier has constant soundness error. In a first step (Section 2.5.1), we use the soundness error proved above (Section 2.4) to show that the t-repeated ALMSS verifier has soundness error γ when $t = \Omega(\log N + \log \frac{1}{\gamma})$. In a second

step (Section 2.5.2), we prove that the basic ALMSS verifier (no repetitions) has constant soundness error. The second step is *generic* and of independent interest: we prove that if a t-repeated verifier has soundness error $\exp(-t)$, then the corresponding basic verifier has soundness error $O(1)$.

2.5.1 The t-repeated ALMSS verifier has soundness error $\exp(-t)$

While in the classical setting reducing soundness error via simple repetition is straightforward (t-wise repetition reduces soundness error from δ to δ^t), in the non-signaling setting simple repetition *does not work*.[6] Indeed, consider the non-signaling function (in fact, distribution) that, with probability $1-\varepsilon$, answers the verifier's queries in an accepting way, and otherwise answers randomly. This non-signaling function is accepted by the t-repeated verifier with probability $\approx 1-\varepsilon$, which is about the same as the probability that it is accepted by a single verifier.

However, this example provides intuition for how one circumvents this issue. Informally, we would like to extract the "$1-\varepsilon$ good part" that satisfies the verifier, and drop the "ε bad part". We follow a technique used in [17] and, instead of arguing about the probability that $\mathcal{L}$ passes the t-repeated verifier, we argue that the non-signaling function $\mathcal{L}$ *conditioned on passing the t-repeated verifier* passes the basic verifier with high probability. Indeed, in the aforementioned example, conditioning on at least one test passing removes the "ε bad part" injected by the distribution, and intuitively *extracts* the part of $\mathcal{L}$ that is passing the verifier. An interesting feature of our analysis of the verifier is that our conclusion is about the basic verifier, not the relaxed t-repeated verifier, which plays a major role in the analysis in [17].[7] This is a qualitative difference in our analysis arising from our use of property testing (not present in [17]), which also simplifies the analysis.

In more detail, let $\mathcal{L}'$ denote the linear non-signaling function that equals $\mathcal{L}$ when conditioned on passing the t-repeated verifier. Namely, if E is the (random) event that $\mathcal{L}$ passes the t-repeated verifier, then for any $S \subseteq \{0,1\}^n$ (of some maximal size) and $\vec{b} \in \{0,1\}^S$, we define

$$\Pr\left[\mathcal{L}'(S) = \vec{b}\right] := \Pr\left[\mathcal{L}(S) = \vec{b} \mid E\right] = \frac{\Pr[\mathcal{L}(S) = \vec{b} \wedge E]}{\Pr[E]} \quad . \tag{1}$$

We then prove that $\mathcal{L}'$ passes the basic verifier with probability at least $1 - \frac{1/\Pr[E]}{\exp(t)}$.

The proof uses a generic lemma stating that, if we run $t+d$ independent tests, then the probability that at most r out of the first d tests pass and all of the last t tests pass is at most $(\frac{d}{t+d})^{r+1}$. A naive application of this lemma (with $r=0$ and $d=1$) shows that $\mathcal{L}'$ passes the basic verifier with probability at least $1 - \frac{1/\Pr[E]}{(t+1)}$. This is not enough, because (using $\Pr[E] \geq \gamma$) we would require $t = \Omega(N \log N \cdot \frac{1}{\gamma})$ to prove soundness, which is again far too many repetitions.

However, we leverage the linearity of $\mathcal{L}$ to deduce the stronger guarantee, as we now explain. We want to bound the probability that $\mathcal{L}'$ does not pass the basic verifier, which means we need to bound the probability that $\mathcal{L}$ fails exactly the first test of $t+1$ independent tests. We do this by arguing this individually for each of the two types of tests made by

[6] Even if simple repetition were to reduce soundness error from δ to δ^t, then to get $\delta^t = \gamma$ when $\delta = 1 - O(\frac{1}{N \log N})$ we would need to repeat $t = \Omega(N \log N + \log \frac{1}{\gamma})$ times, which requires too large of a locality k for the analysis.

[7] The relaxed t-repeated verifier runs t tests and accepts if a large fraction of them pass.

the ALMSS verifier: the tensor test "$\mathcal{L}(D_u)\mathcal{L}(D_v) = \mathcal{L}(u \otimes v)$" and the satisfiability test "$\mathcal{L}(\sum_{j=1}^{M} s_j P_j) = \sum_{j=1}^{M} s_j c_j$". We will explain our techniques in the case of the satisfiability test; the same techniques work for the tensor test, but the algebra is messier.

In the case of the satisfiability test, we split the "special" test (i.e., the first one) into d pairs of tests, such that each individual test is random, but each pair is correlated so that if both tests in some pair pass, then the original test passes. Specifically, we draw d random vectors $s^{(1)}, \ldots, s^{(d)} \in \{0,1\}^M$, and then we split the test "$\mathcal{L}(\sum_{j=1}^{M} s_j P_j) = \sum_{j=1}^{M} s_j c_j$" into the d pairs of tests

$$\text{“}\mathcal{L}\left(\sum_{j=1}^{M}(s_j + s_j^{(i)})P_j\right) = \sum_{j=1}^{M}(s_j + s_j^{(i)})c_j\text{”} \quad \text{and} \quad \text{“}\mathcal{L}\left(\sum_{j=1}^{M} s_j^{(i)} P_j\right) = \sum_{j=1}^{M} s_j^{(i)} c_j\text{”} \ .$$

This allows us to apply the lemma with $d = O(t)$, and $r = O(t)$, which shows that $\mathcal{L}'$ passes the basic verifier with probability at least $1 - \frac{1/\Pr[E]}{\exp(t)}$, an exponential decrease in t.

The above analysis shows soundness error of γ for the t-repeated verifier, for $t = O(\log N + \log \frac{1}{\gamma})$. Indeed, by the above argument, the conditioned function $\mathcal{L}'$ passes the basic verifier with probability $1 - \frac{1}{\gamma}\exp(-t) = 1 - O(\frac{1}{N \log N})$, by choice of t. The analysis in the previous section (Section 2.4) then implies that $C(x) = 1$, proving soundness of the t-repeated verifier.

Setting $\gamma = \exp(-t)$, the discussion so far merely shows that the t-wise repetition of the ALMSS verifier, which makes $4t$ queries, has soundness error $\exp(-t)$ when $t = \Omega(\log N)$; moreover, we get no conclusions for $t = o(\log N)$. But we still did not conclude anything about the soundness of a *single* invocation of the 4-query ALMSS verifier. We next discuss how to handle this case.

2.5.2 Back to the 4-query ALMSS verifier

We establish that the ALMSS verifier has constant soundness error by proving a generic lemma. The lemma states that, for *any* PCP verifier V, if the t-repeated verifier V^t has soundness error $\exp(-t)$, then V has soundness error $O(1)$. Since we have already argued that the t-repeated ALMSS verifier has soundness error $\exp(-t)$ for $t = \Omega(\log N)$, we can conclude that the basic ALMSS verifier has soundness error $O(1)$, against $O(\log N)$-non-signaling linear functions.

In the classical case, the proof of this generic fact is trivial: a (classical) function passes a PCP verifier V with probability δ if and only if it passes the t-repeated verifier V^t with probability δ^t. However, in the non-signaling case, it is not clear what one can say because a non-signaling function can provide correlated answers across repetitions. Nevertheless, we are able to *lower bound* the probability that V^t accepts by a quantity that is almost δ^t (which is, in particular, almost tight).

To our knowledge, we are the first to relate the soundness of V to the soundness of V^t. Generic statements in prior works (starting with [15]) have related the soundness of V^t to the soundness of the t-repeated relaxed verifier (which accepts if a vast majority of the t tests pass), but did not provide conclusions about the basic verifier V.

2.6 Testing linearity with low error

Below we discuss linearity testing *with low error* (Theorem 5) in more detail.

Warmup: distributions. We have discussed (in Section 2.1) how to test linearity with low error in the classical setting. In order to illustrate some of the difficulties that arise in the non-signaling setting, we first discuss a special case of it: testing linearity against a *distribution* over functions.

First, suppose that $\mathcal{D}$ is a distribution over functions $f\colon \{0,1\}^n \to \{0,1\}$ that passes the linearity test with probability $1-\varepsilon$. The self-correction $\hat{\mathcal{D}}$ that on input $x \in \{0,1\}^n$ samples a random $z \in \{0,1\}^n$ and outputs $\hat{\mathcal{D}}(x) = \mathcal{D}(z+x) - \mathcal{D}(z)$ is 2ε-close to a distribution over *linear* functions $\mathcal{D}^*$, namely, for every $x \in \{0,1\}^n$ it holds that $\left|\Pr[\hat{\mathcal{D}}(x)=1] - \Pr[\mathcal{D}^*(x)=1]\right| \leq 2\varepsilon$. Indeed, consider the distribution $\mathcal{D}^*$ that samples $f \leftarrow \mathcal{D}$ and outputs any linear function f^* closest to f.[8] Then, for every function f and $x \in \{0,1\}^n$, the probability over a random $z \in \{0,1\}^n$ that $f^*(z) = f(z)$ and $f^*(z+x) = f(z+x)$ is at least $1 - 2\varepsilon_f$, where $\varepsilon_f := 1 - \Pr_{x,y}[f(x)+f(y) = f(x+y)]$. Denoting by d_f denotes the probability that $\mathcal{D}$ samples the function f, we conclude that $\left|\Pr[\mathcal{D}^*(x)=1] - \Pr[\hat{\mathcal{D}}(x)=1]\right| \leq \sum_f 2\varepsilon_f \cdot d_f = 2\varepsilon$.

Next, suppose that we seek a self-correction of $\mathcal{D}$ that is δ-close to a distribution over linear functions, for $\delta \ll 2\varepsilon$. One idea is to follow the same strategy as in the case of a single function: repeat the linearity test and then do self-correction. This idea, however, does not work now.

Consider the distribution $\mathcal{D} = (1-\varepsilon)\cdot \mathbf{0} + \varepsilon \cdot \mathbf{1}$, i.e., the distribution that with probability $1-\varepsilon$ answers according to the all-zeros function (a linear function), and with probability ε according to the all-ones function (a function maximally far from linear functions). While $\mathcal{D}$ passes the linearity test with probability $1-\varepsilon$, $\mathcal{D}$ also passes the t-repeated linearity test with probability $1-\varepsilon$. In other words, if $\mathcal{D}$ passes the t-repeated linearity test with probability $1-\varepsilon$, we can still only conclude that $\hat{\mathcal{D}}$ is 2ε-close to linear, independent of t.

While repeating the test does not increase the rejection probability, it can still be used to improve the quality of self-correction, by considering a *different* notion of self-correction that penalizes functions in the support of $\mathcal{D}$ that are far from linear. Concretely, consider the distribution $\mathcal{D}_t$ that equals $\mathcal{D}$ *when conditioned on the event that the t-repeated linearity test passes*, and then define $\hat{\mathcal{D}}_t$ to be the self-correction of $\mathcal{D}_t$. That is, $\hat{\mathcal{D}}_t$ samples f from $\mathcal{D}_t$ and answers any query $x \in \{0,1\}^n$ by sampling $z \in \{0,1\}^n$ and returning $f(z+x) - f(z)$. We claim that $\hat{\mathcal{D}}_t$ is very close to linear.

Indeed, suppose that $\mathcal{D}$ passes the t-repeated test with probability $\gamma > 0$, and let $c > 1$ be a parameter. A function f sampled from $\mathcal{D}_t$ is $\frac{\ln c}{t}$-close to linear with probability at least $\frac{\gamma - 1/c}{\gamma} = 1 - \frac{1}{\gamma c}$.[9] Setting $c := t/\log t$, the probability that $\mathcal{D}_t$ outputs a function f that is $\frac{\log t - \log\log t}{t}$-far from linear is at most $\frac{\log t}{\gamma t}$. Therefore, by applying the argument from the beginning of this subsection, we conclude that $\hat{\mathcal{D}}_t$ is $O_\gamma(\frac{\log t}{t})$-close to a distribution over linear functions.

We can further reduce the distance to be exponentially small in t by performing self-correction t times: $\hat{\mathcal{D}}_t(x)$ now samples $z_1, \ldots, z_t \in \{0,1\}^n$, and outputs the majority of $\{\mathcal{D}(z_i + x) - \mathcal{D}(z_i)\}_{i\in[t]}$ conditioned on the event that the t-repeated linearity test passes. By setting $c := 2^{t/10}$ in the discussion above, we conclude that if we sample f from $\mathcal{D}_t$, then f is 0.1-close to a linear function f^* with probability $1 - \frac{1}{\gamma 2^{t/10}}$. In particular, for every $x \in \{0,1\}^n$ it holds that $\Pr_{z_i}[f^*(z_i) = f(z_i) \wedge f^*(z_i + x) = f(z_i + x)] \geq 0.8$, and so the probability that the majority value of $\{\mathcal{D}(z_i + x) - \mathcal{D}(z_i)\}_{i\in[t]}$ is not equal to $f^*(x)$ is $2^{-O(t)}$. In sum, the t-repeated self-correction conditioned on the event that the t-repeated linearity test passes yields us a distribution that is $\frac{1}{\gamma}2^{-O(t)}$-close to linear.

---

[8] Recall that if f is 0.25-close to linear functions then f^* is unique. We do not rely on uniqueness.

[9] The t-repeated linearity test accepts a function f that is $\frac{\ln c}{t}$-far from linear with probability at most $(1 - \frac{\ln c}{t})^t \leq \frac{1}{c}$.

The non-signaling case. The case of non-signaling strategies is similar to the case of distributions in that the analysis of the self-correction involves conditioning over a certain event. Yet, the conclusions and steps of the proof are quite different. Informally, this is because non-signaling functions are quasi-distributions (probabilities can be negative), which prevents us from doing a straightforward analysis such as the one above. We now discuss how we address this.

Suppose that we have a k-non-signaling function $\mathcal{F}\colon \{0,1\}^n \to \{0,1\}$ that passes the linearity test with probability $1-\varepsilon$. The result of [9] proves that the self-correction $\hat{\mathcal{F}}$ defined as $\hat{\mathcal{F}}(x) := \mathcal{F}(z+x)-\mathcal{F}(z)$ (where z is chosen randomly from $\{0,1\}^n$) is $2^{O(k)}\varepsilon$-close to linear. This is too large in our setting as we have $k = O(\log N)$, and we would like the distance to be $O(\varepsilon)$. Instead, we prove a slightly different guarantee from [9]. Namely, we show that there is a linear non-signaling function $\mathcal{L}$, such that on every set S, $\hat{\mathcal{F}}$ is $O(|S|\sqrt{\varepsilon})$-close to $\mathcal{L}$. Unlike in the result of [9], our distance now decays with $|S|$, and is in particular independent of k. This is sufficient for our purposes, since we set $|S| = 4$, the number of queries made by the ALMSS verifier.

In our proof, we consider a *different* self-correction $\overline{\mathcal{F}}_t$ that, unlike $\hat{\mathcal{F}}$, is *only used in the analysis* and is not used by the compiler. First, we show that $\overline{\mathcal{F}}_t$ passes the linearity test with probability $1-\exp(-t)$, and so the result of [9] implies that $\overline{\mathcal{F}}_t$ is very close to a linear non-signaling function. Then, we relate $\overline{\mathcal{F}}_t$ and $\hat{\mathcal{F}}$ to show that $\hat{\mathcal{F}}$ is $O(\sqrt{\varepsilon})$-close to a linear non-signaling function.

Informally, the self-correction $\overline{\mathcal{F}}_t$ equals $\mathcal{F}$ with the standard self-correction procedure repeated t times, conditioned on $\mathcal{F}$ passing $(1-\sqrt{\varepsilon})t$ of t repetitions of the linearity test. In more detail, given a subset $S \subseteq \{0,1\}^n$, $\overline{\mathcal{F}}_t(S)$ is the following distribution. For each $x \in S$, sample uniform and independent $z_x^{(1)}, \ldots, z_x^{(t)} \in \{0,1\}^n$ conditioned on satisfying the same linear dependencies as in S; for instance, if $S = \{x, y, x+y\}$, then $z_x^{(i)} + z_y^{(i)} = z_{x+y}^{(i)}$ holds for all i. Then $\overline{\mathcal{F}}_t$ assigns to each $x \in S$ the value $\mathrm{MAJ}_{i\in[t]}\{\mathcal{F}(z_x^{(i)}+x) - \mathcal{F}(z^{(i)})\}$ *conditioned on the event* that $\mathcal{F}$ passes at least $(1-\sqrt{\varepsilon})t$ of t repetitions of the basic linearity test. We note that if $\mathcal{F}$ is linear, then $\overline{\mathcal{F}}_t \equiv \hat{\mathcal{F}} \equiv \mathcal{F}$.

The first part of the analysis uses a lemma that informally states that by conditioning on $\mathcal{F}$ passing most of the t-repeated linearity tests, we force the conditioned $\mathcal{F}$ to behave "close" to linear. Specifically, letting $b_x^{(i)} = \mathcal{F}(z_i + x) - \mathcal{F}(z_i)$, we get that with probability $1-\exp(-t)$ there is a bit b_x that equals $b_x^{(i)}$ for at least $\frac{3t}{4}$ of the i's (so the majority is a vast majority), which implies that $\overline{\mathcal{F}}_t(x) = b_x$, and analogously for y and $x+y$. Then, via a similar argument, we show that with probability $1-\exp(-t)$ for at least $\frac{3t}{4}$ of the i's it holds that $b_x^{(i)} + b_y^{(i)} = b_{x+y}^{(i)}$. By union bound, these events hold simultaneously, and so we conclude that $\overline{\mathcal{F}}_t$ satisfies $\overline{\mathcal{F}}_t(x) + \overline{\mathcal{F}}_t(y) = \overline{\mathcal{F}}_t(x+y)$ with probability $1-\exp(-t)$. We then invoke the result of [9] and conclude that $\hat{\mathcal{F}}$ is very close to some linear non-signaling function $\mathcal{L}$.

In the second step, we relate $\hat{\mathcal{F}}$ to $\overline{\mathcal{F}}_t$ by claiming that if $\Pr[\mathcal{F}(x)+\mathcal{F}(y) = \mathcal{F}(x+y)] \geq 1-\varepsilon$, then $\hat{\mathcal{F}}$ and $\overline{\mathcal{F}}_t$ are close in some precise sense. We first observe that if we run the t-repeated linearity test, i.e., choose $x^{(1)}, y^{(1)}, \ldots, x^{(t)}, y^{(t)}$ and check that $\mathcal{F}(x^{(i)}) + \mathcal{F}(y^{(i)}) = \mathcal{F}(x^{(i)} + y^{(i)})$ for every i, then a simple Markov argument shows that with high probability, most of the linearity tests are satisfied. For instance, with probability $1-\sqrt{\varepsilon}$ at least $(1-\sqrt{\varepsilon})t$ of the i's satisfy the linear constraint. This means that the event conditioned on in the definition of $\overline{\mathcal{F}}_t$ is a large event. We also know from the first part of the analysis that, with high probability, the conditioning causes most of the evaluations of $\mathcal{F}(z_x^{(i)}+x) - \mathcal{F}(z_x^{(i)})$ to output the same value. Intuitively, this implies that $\hat{\mathcal{F}}$ is close to $\overline{\mathcal{F}}_t$, via the following reasoning. Since $\overline{\mathcal{F}}_t$ conditions on a large event, it is close to the corresponding self-correction that does not condition at all. Since the majority taken over the evaluations of $\mathcal{F}(z_x^{(i)} + x) - \mathcal{F}(z_x^{(i)})$

when computing $\overline{\mathcal{F}}_t$ is a vast majority, with high probability $\hat{\mathcal{F}}$ (which is a sample from one of the elements the majority is over) will agree with the vast majority. This allows us to conclude that for any set S, $\hat{\mathcal{F}}$ will be $O(|S|\sqrt{\varepsilon})$-close to $\overline{\mathcal{F}}_t$.

See the full version of this paper for details.

References

1. Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. Preliminary version in FOCS '92.
2. Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. Preliminary version in FOCS '92.
3. László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd ACM Symposium on Theory of Computing*, STOC '91, pages 21–32, 1991.
4. László Babai, Lance Fortnow, and Carsten Lund. Non-Deterministic Exponential Time has Two-Prover Interactive Protocols. *Computational Complexity*, 1:3–40, 1991. Preliminary version appeared in FOCS '90.
5. Mihir Bellare, Don Coppersmith, Johan Håstad, Marcos A. Kiwi, and Madhu Sudan. Linearity testing in characteristic two. *IEEE Transactions on Information Theory*, 42(6):1781–1795, 1996.
6. Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of Proximity, Shorter PCPs, and Applications to Coding. *SIAM Journal on Computing*, 36(4):889–974, 2006.
7. Nir Bitansky and Alessandro Chiesa. Succinct Arguments from Multi-Prover Interactive Proofs and their Efficiency Benefits. In *Proceedings of the 32nd Annual International Cryptology Conference*, CRYPTO '12, pages 255–272, 2012.
8. Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993.
9. Alessandro Chiesa, Peter Manohar, and Igor Shinkar. Testing Linearity against Non-Signaling Strategies. In *Proceedings of the 33rd Annual Conference on Computational Complexity*, CCC '18, pages 17:1–17:37, 2018.
10. Irit Dinur and Omer Reingold. Assignment Testers: Towards a Combinatorial Proof of the PCP Theorem. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science*, FOCS '04, pages 155–164, 2004.
11. Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996. Preliminary version in FOCS '91.
12. Justin Holmgren and Ron Rothblum. Delegation With (Nearly) Optimal Time/Space Overhead. In *Proceedings of the 59th IEEE Symposium on Foundations of Computer Science*, FOCS '18, pages ???–???, 2018.
13. Tsuyoshi Ito. Polynomial-Space Approximation of No-Signaling Provers. In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming*, ICALP '10, pages 140–151, 2010.
14. Tsuyoshi Ito, Hirotada Kobayashi, and Keiji Matsumoto. Oracularization and Two-Prover One-Round Interactive Proofs against Nonlocal Strategies. In *Proceedings of the 24th IEEE Annual Conference on Computational Complexity*, CCC '09, pages 217–228, 2009.
15. Yael Kalai, Ran Raz, and Ron Rothblum. Delegation for Bounded Space. In *Proceedings of the 45th ACM Symposium on the Theory of Computing*, STOC '13, pages 565–574, 2013.

16 Yael Tauman Kalai, Ran Raz, and Oded Regev. On the Space Complexity of Linear Programming with Preprocessing. In *Proceedings of the 7th Innovations in Theoretical Computer Science Conference*, ITCS '16, pages 293–300, 2016.

17 Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In *Proceedings of the 46th ACM Symposium on Theory of Computing*, STOC '14, pages 485–494, 2014. Full version available at `https://eccc.weizmann.ac.il/report/2013/183/`.

18 Leonid A Khalfin and Boris S Tsirelson. Quantum/classical correspondence in the light of Bell's inequalities. *Foundations of physics*, 22(7):879–948, 1992.

19 Joe Kilian. A note on efficient zero-knowledge proofs and arguments. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, STOC '92, pages 723–732, 1992.

20 Susumu Kiyoshima. No-signaling Linear PCPs. In *Proceedings of the 16th Theory of Cryptography Conference*, TCC '18, pages 67–97, 2018.

21 Silvio Micali. Computationally Sound Proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000. Preliminary version appeared in FOCS '94.

22 Omer Paneth and Guy Rothblum. On Zero-Testable Homomorphic Encryption and Publicly Verifiable Non-Interactive Arguments. In *Proceedings of the 15th Theory of Cryptography Conference*, TCC '17, 2017.

23 Sandu Popescu and Daniel Rohrlich. Quantum nonlocality as an axiom. *Foundations of Physics*, 24(3):379–385, 1994.

24 Peter Rastall. Locality, Bell's theorem, and quantum mechanics. *Foundations of Physics*, 15(9):963–972, 1985.

25 Hanif D. Sherali and Warren P. Adams. A Hierarchy of Relaxations between the Continuous and Convex Hull Representations for Zero-One Programming Problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.