Sound 3-Query PCPPs Are Long

ELI BEN-SASSON and PRAHLADH HARSHA
Technion, Haifa, Israel
ODED LACHISH
University of Warwick, Coventry, UK
and
ARIE MATSLIAH
CWI, Amsterdam, Netherlands

We initiate the study of the trade-off between the length of a probabilistically checkable proof of proximity (PCPP) and the maximal soundness that can be guaranteed by a 3-query verifier with oracle access to the proof. Our main observation is that a verifier limited to querying a short proof cannot obtain the same soundness as that obtained by a verifier querying a long proof. Moreover, we quantify the soundness deficiency as a function of the proof-length and show that any verifier obtaining "best possible" soundness must query an exponentially long proof.

In terms of techniques, we focus on the special class of inspective verifiers that read at most 2 proof-bits per invocation. For such verifiers, we prove *exponential* length-soundness trade-offs that are later on used to imply our main results for the case of general (i.e., not necessarily inspective) verifiers. To prove the exponential trade-off for inspective verifiers, we show a connection between PCPP proof length and property-testing query complexity that may be of independent interest. The connection is that any linear property that can be verified with proofs of length ℓ by linear inspective verifiers must be testable with query complexity $\approx \log \ell$.

Categories and Subject Descriptors: F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

General Terms: Theory, Algorithms

Additional Key Words and Phrases: PCP, PCP of proximity

A preliminary version of this article appeared in *Proceedings of the 35th International Colloquium on Automata, Languages, and Programming (ICALP'08)* [Ben-Sasson et al. 2008].
The work of E. Ben-Sasson, P. Harsha, and O. Lachish was supported in part by the European Community (International Reintegration Grant) and by the Israel Science Foundation (Grant No. 2009054).
Authors' addresses: E. Ben-Sasson and P. Harsha, Computer Science Department, Technion—Israel Institute of Technology, Haifa, Israel; email: {eli, prahladh}@cs.technion.ac.il; O. Lachish, Centre for Discrete Mathematics and its Applications (DIMAP), University of Warwick, Coventry, United Kingdom; email: oded@cs.warwick.ac.uk; A. Matsliah, CWI, Amsterdam, Netherlands. email: arie@cwi.nl.

ACM Reference Format:
Ben-Sasson, E., Harsha, P., Lachish, O., and Matsliah, A. 2009. Sound 3-query PCPPs are long. ACM Trans. Comput. Theor. 1, 2, Article 7 (September 2009), 49 pages.
DOI = 10.1145/1595391.1595394. http://doi.acm.org/10.1145/1595391.1595394.

1. INTRODUCTION

In this article we discuss the relationship between two basic parameters of Probabilistically Checkable Proofs of Proximity (PCPPs)—their *proof length* and *soundness*. PCPPs were simultaneously introduced by Ben-Sasson et al. [2006] and (under the name assignment testers) by Dinur and Reingold [2006], and a similar notion also appeared earlier in Szegedy [1999] and Ergün et al. [2004]. The interest in PCPPs stems first and foremost from the role they play within the proof of the celebrated PCP Theorem of Arora and Safra [1998] and Arora et al. [1998]. All recent constructions of PCPs, starting with the work of Ben-Sasson et al. [2006] and Dinur and Reingold [2006], use PCPPs to simplify the proof of the PCP theorem and improve certain aspects of it, most notably, to decrease the length of proofs as in Ben-Sasson et al. [2006], Ben-Sasson and Sudan [2008], and Dinur [2007]. All previous proofs of the PCP theorem implicitly use PCPPs and can be augmented to yield them. (See, e.g., Ben-Sasson et al. [2006, Theorem 3.2] for a conversion of the original PCP system of Arora and Safra [1998] and Arora et al. [1998] into a PCPP). But PCPPs are also interesting beyond the scope of the PCP Theorem. They can be used to transform any error correcting code into a locally testable one and to construct "relaxed" locally decodable codes [Ben-Sasson et al. 2006]. Additionally, as shown by Fischer and Fortnow [2006], and Guruswami and Rudra [2005], they have applications to questions in the theory of "tolerant" property testing that was introduced by Parnas et al. [2006].

A *PCPP verifier*, (or, simply, verifier) for a property $P \subset \{0,1\}^n$ is a randomized, sublinear-time algorithm that distinguishes with high probability between inputs that belong to P and inputs that are far in relative Hamming distance from all members of P. In this respect a verifier is similar to a *property-tester* as defined by Goldreich et al. [1998]. However, in contrast to a tester, the verifier may query an auxiliary proof, called a *proof of proximity*. A PCPP system has four basic parameters of interest, described next—*length, query complexity, completeness* and a *soundness function*. The proof length is the length of the auxiliary proof that is queried by the verifier.[1] The query complexity is the maximal number of bits that can be read from *both* the input and the proof. The completeness parameter is the minimal probability with minimal probability with which inputs that belong to P are accepted when they are presented along with a "good" proof of proximity. Finally, the soundness function $s(\delta)$ is the minimal rejection probability of inputs that are δ-far (in relative Hamming distance) from (all members of) P, where the minimum

[1]In PCP literature one often encounters *randomness complexity* as a means for bounding proof-length. The two parameters are closely related, that is, proof-length $\approx 2^{\text{randomness}}$ and we stick to the former parameter.

is taken over all such δ-far inputs and all possible proofs that may accompany them.[2] (See Section 2 for a formal definition of PCPPs and further discussion of their parameters).

1.1 Context and Motivation

We are motivated by the attempt to understand the limitations of PCP constructions. One interesting open question related to our research is that of obtaining 3-query PCPs with quasilinear length, completeness $1 - \varepsilon$ and soundness $\frac{1}{2} - \varepsilon$ for any language in **NP**. For the sake of reference, we informally call such a construction a "super-PCP." The celebrated result of Håstad [2001] obtains three out of four of these parameters (the proof length there is a (very large) polynomial). Numerous other works, such as Guruswami et al. [1998], Håstad and Khot [2005], Samorodnitsky and Trevisan [2000], Engebretsen and Holmerin [2008], Khot and Saket [2006], and Samorodnitsky and Trevisan [2006], to name a few, investigate optimal or nearly optimal trade-offs between the three parameters of query complexity, completeness and soundness, while settling for polynomial length proofs. A different line of research focused on optimizing the trade-off between proof length and query complexity [Polishchuk and Spielman 1994; Harsha and Sudan 2000; Goldreich and Sudan 2006; Ben-Sasson et al. 2003, 2006; Ben-Sasson and Sudan 2008; Dinur 2007; Moshkovitz and Raz 2008a, 2007] and all of these constructions obtain perfect completeness. Several of these works, most notably Harsha and Sudan [2000], Goldreich and Sudan [2006], Moshkovitz and Raz [2008a], and Moshkovitz and Raz [2007], also strive to simultaneously optimize the fourth parameter, soundness, but have stopped short of constructing a "super-PCP."

Our results show why a certain natural class of PCP constructions will not be suitable for reaching our goal. Most constructions of "short" PCPs (i.e., with proof length $n^{1+o(1)}$ for **NP** instances of size n) start by encoding a witness for an **NP**-instance by some good linear error correcting code, usually based on univariate or multivariate polynomials. These codes are inherently hard to test because they have relatively high degree and are converted into locally testable codes by appending a PCPP to each individual codeword. Moreover, all known PCPP constructions are linear, that is, can be obtained by applying a linear transformation to the codeword (see Section 2.3.2). Our results show, for instance, that no 3-query linear PCPP appended to such a code can achieve anything close to the best possible, unless the proof is exponentially long.

This work can also be placed within the larger context of the study of limitations of PCPs and objects related to them. There are preciously few results that give nontrivial trade-offs between the basic parameters of a PCP system. One notable example presented by Zwick [1998] shows that the soundness of a 3-query PCP verifier with perfect-completeness cannot exceed 3/8 unless

[2]Often, in literature on PCPs, the term "soundness" refers to "soundness-error" which is defined to be the maximal acceptance probability of a "bad" input. The connection between soundness (used here) and soundness-error, denoted s_{error}, is given by $s = 1 - s_{error}$.

NP $\subseteq$ **BPP**. A larger number of works try to understand the limitations of PCP systems by either *(i)* showing limitations of specific techniques used in PCP constructions, or *(ii)* proving limitations on computational and combinatorial objects that are closely related to PCPs. Along the first line of research one can mention the work of Feige and Kilian [1995] that shows limitations on derandomizing the parallel repetition method of Raz [1998] and that of Bogdanov [2005] that shows upper bounds on the soundness that can be obtained from the gap amplification technique of Dinur [2007]. The second line of research includes the study of the limits of various basic parameters of locally decodable codes [Katz and Trevisan 2000; Kerenidis and de Wolf 2004; Woodruff 2008], locally testable codes [Ben-Sasson et al. 2003; Guruswami 2006], unique games [Khot 2002; Trevisan 2005; Charikar et al. 2006] and a large number of results regarding the limits of property testing (see the survey [Fischer 2001] for further information). Our work resonates with both of these lines of research because PCPPs are computational objects that are closely related to PCPs and constitute the method of choice for constructing them. We also hope that the research initiated here will contribute to a better understanding of the inherent limits of the magical PCP theorem.

Last but not least, the actual soundness parameter one obtains from a small query PCPP (and the PCPs and LTCs resulting from it) may someday in the future deem whether such objects can be put to practical use in proof checking (a la Babai et al. [1991]), communication and cryptography (as in Kilian [1992], Micali [2000]). Therefore, the study of trade-offs between soundness and proof length is of inherent importance.

1.2 Informal Description of Main Results

To describe our results, let us discuss the range of parameters we can expect from a verifier for a *linear* property over the binary alphabet, that is, a property that is closed under addition modulo 2. (This amounts to saying P is a linear subspace of $\mathbb{F}_2^n$ where $\mathbb{F}_2$ denotes the two-element field.) We look at nonadaptive 3-query verifiers with perfect completeness, thereby fixing two of the four basic parameters, and look at the trade-off between proof length and soundness. We point out that all known constructions of PCPPs naturally yield nonadaptive 3-query verifiers with perfect completeness (see, e.g., Lemma 8.1), so the results described next apply to all of them.

Suppose we are interested in minimizing proof length. The results of Ben-Sasson and Sudan [2008] and Dinur [2007] give constructions with proofs of length at most $m \cdot \text{polylog}\, n$ where m is the minimal size of circuit deciding P. (Notice the linearity of P implies $m = O(n^2)$.) Regarding the soundness function, consider linear properties P having minimal distance $\sim n/2$ and vanishing rate. In this case, a random word can be shown to have, with high probability, distance $\delta \approx \frac{1}{2}$ from P. The "short PCPP" construction mentioned earlier gives $s(\delta) > \varepsilon$ for some small and unspecified constant $\varepsilon > 0$ that depends only on δ and neither on P, nor on n.

Next, let us try to increase the soundness. We show in Theorem 2.8 that soundness can be boosted to $s(\delta) \geq \delta$ and this soundness is obtained by a *linear* verifier. A verifier is called linear if it checks a linear constraint on the symbols it queries. (For $\mathbb{F}_2$ this amounts to saying the verifier accepts iff the sum (mod 2) of the queried bits is 0.) For such verifiers, it can be shown that $s(\delta)$ is at most $\frac{1}{2}$ and thus the soundness of our construction is optimal. On the down side, the length of the proof used by this verifier is exponential in n. (We note in passing that this soundness-optimal construction can be carried out over any finite field of prime size. See Theorem 2.8 for details.)

To sum up the situation so far, we have constructions that are nearly optimal in length, but are deficient in soundness and we have constructions that are optimal in soundness but deficient in length. One could have conjectured (as we did before embarking on this research project) that a "super-PCPP" with short proofs and optimal soundness exists. Our first main result, stated in Theorem 2.9 and Corollary 2.10, rules this out. We show a trade-off between proof length and soundness that essentially matches our soundness-optimal construction. In plain words, for some properties (discussed later) any PCPP verifier that queries a short proof of length ℓ must incur a *soundness deficiency*, and this deficiency increases as ℓ decreases (informally, soundness deficiency measures how much the rejection probability of wrong inputs is reduced when moving from long proofs to short ones; see Definition 2.5 for a formal definition of soundness deficiency).

Our next main result, stated in Theorem 2.11 and Corollary 2.12, proves a tighter trade-off similar to the one mentioned for the case of $\mathbb{F}_p$-linear verifiers for $\mathbb{F}_p$-linear properties over a finite field of (prime) size p. Our results in this case are stronger even though the query complexity, when measured in bits, is greater than 3 (however, the bits are read from three "blocks", where each block encodes a field element). Finally, our third main result, stated in Theorem 2.13 and Corollary 2.14, presents essentially the same kind of exponential trade-off between soundness and proof length for a natural generalization of linear verifiers, called *unique* verifiers (see Definition 2.2).

So far we have not specified which properties cause this kind of trade-off to arise, that is, which properties are hard to verify. The culprits are properties that are "hard to test." Informally, we say that $P \subset \{0, 1\}^n$ is hard to test if any property-tester for P (as defined in Goldreich et al. [1998]) that rejects (say) $\frac{1}{3}$-far inputs with probability greater than (say) $1/100$ requires query complexity $q \gg 3$. Our main theorems (Theorems 2.9, 2.11, and 2.13) show an exponential trade-off between the property-testing query complexity q and the minimal length of a 3-query verifier with large soundness (say, achieving soundness function $s(\delta) \geq \delta - 1/100$). In a certain sense we show that any property that is hard to test is also hard to verify. Next, we briefly explain why we believe our results are interesting.

Later Results. We would like to add that, after the publication of this work, Moshkovitz and Raz [2008b] constructed 2-query projection PCPs of almost linear size with perfect completeness and soundness $o(1)$ over a nonbinary alphabet. An immediate consequence of this result is the construction of 3-query PCP with almost-linear size, completeness $1 - o(1)$ and soundness $1/2 - o(1)$ for binary alphabet. Their construction, however, does not imply existence of PCPPs of similar parameters.

1.3 Proof Techniques

Inspective PCPPs. Consider a 3-query verifier that rejects inputs that are δ-far from P with probability $\approx \delta$. At first sight it may seem that reaching soundness $s(\delta) \geq \delta$ is impossible because such high soundness forces the verifier to make at least one out of three queries to the input, leaving only two queries for checking the proof. Indeed, a verifier that seldom queries the input can easily be fooled to accept with high probability a legitimate proof accompanying an input that is δ-far from P. The need to look at the input naturally leads us to define an *inspective* verifier as one that *inspects* the input on every invocation. Formally, an inspective verifier is one that makes at most two queries to the proof; all other queries are to the input.[3] Our main *positive* result, Theorem 2.8, says that every $\mathbb{F}_p$-linear property over a prime field of size p has a 3-query $\mathbb{F}_p$-linear inspective verifier with soundness function $s(\delta) \geq \delta$ and proof length at most $p^{\dim(P)}$. "Good" proofs for inputs $w \in P$ turn out to be certain "folded" Hadamard codewords and we analyze soundness using the Fourier analytic approach to linearity testing that was introduced by Bellare et al. [1996]. (See Section 3 for more details.) The soundness obtained by the verifier of Theorem 2.8 is the benchmark against which we measure all other 3-query verifiers, and next we describe how we prove that short proofs lead to soundness-deficiency with respect to this benchmark.

Exponential Trade-offs between Soundness and Proof Length for Inspective PCPPs. All our results about the soundness deficiency of short PCPPs are based on exponential trade-offs between soundness and proof length for *inspective* PCPPs. Since these results are similar in spirit, let us describe how we obtain them in the simplest setting—that of $\mathbb{F}_2$-linear verifiers. The actual proofs have a few additional subtle details that we brush aside in the following informal description.

Roughly speaking, we show that if the linear property $P \subset \mathbb{F}_2^n$ has a linear inspective verifier that makes q queries[4] to a proof of length ℓ and achieves soundness function $s(\delta)$, then for every $\varepsilon > 0$ the property P has a *tester*, i.e., a proofless verifier that queries only input bits, with query complexity $O((q \log \ell)/\varepsilon)$ and soundness function $s(\delta) - \varepsilon$. The contrapositive formulation for $\delta \approx 1/2$ and $\varepsilon = 0.01$ gives the following statement. Suppose P is hard to test, that is, any tester for P with large soundness requires large query complexity. Then any inspective linear verifier for P with small query complexity must use proofs of exponential length. Examples of hard to test properties include most random Low Density Parity Check (LDPC) codes as defined by Gallager [1963] and linear spaces P for which the dual space, denoted $P^{\perp}$, has no elements of small support (in coding terminology, P is a linear code

[3] Alternatively, an inspective verifier could be defined as one that makes at least one query to the input. For query complexity 3 the two definitions coincide, but for larger query complexity there is a big difference. In particular, our main technical lower bound can be extended to any q-query inspective PCPP, as long as we limit the number of proof-queries to be at most two.
[4] Our trade-offs for inspective PCPPs hold for query complexity larger than 3, even though for the proof of our three main theorems query complexity 3 suffices.

with large dual distance). As mentioned earlier, most error correcting codes actually used as the starting point for constructing PCPs, PCPPs, and LTCs (including Reed-Solomon/Reed-Muller codes) fall within this latter class.

From Inspective to General PCPP Trade-offs. Given the exponential trade-off between soundness and proof length for inspective verifiers, the proof of our main results (stated in Section 2) goes along the following lines. A verifier is forced to choose between two bad options. Either the probability that it reads only proof-bits is large. In this case we fool it by presenting a legitimate proof for some word and capitalize on the fact that the verifier seldom looks at the input (that is δ-far from P). Otherwise, the probability that the verifier makes an inspective query is large. In this case we use the trade-off for the inspective case to fool verifiers that use short proofs. In either of these two cases we manage to fool the verifier into accepting words that are δ-far from P with probability $\approx 1 - \delta/2$, that is, the soundness-deficiency of short-proof verifiers when compared to the exponential length verifier of Theorem 2.8 is $\approx \delta/2$. To complete the overview of our proof techniques we describe next how we obtain exponential length-soundness trade-offs for inspective verifiers.

Proving Trade-off Theorems for Inspective Verifiers. Informally, we convert a q-query *inspective verifier* for P that uses a proof of length ℓ and obtains soundness function s into a proofless *tester* with query complexity $O(q \log \ell)/\varepsilon$ and soundness $s - \varepsilon$. We start by noticing that an inspective verifier gives rise to a natural induced labeled multigraph. The vertices of this graph are indices of proof bits, so the number of vertices equals the length of the proof. For simplicity assume each query-tuple reads exactly two bits of the proof. Thus, each query-tuple defines an edge whose endpoints are the proof bits read, and we label this edge by the set of indices of input bits read when making the query. (The resulting graph may have multiple edges between two vertices, and these edges may have different labels.). Notice the induced graph is actually a representation of the verifier in the sense that a single invocation of the verifier corresponds to picking a random edge in the graph and making the set of queries given by the names of the end-vertices and the edge-label. More to the point, the labeled graph also constitutes a partially defined *constraint graph*, meaning that if all input bits are read then the resulting set of constraints (over proof bits) forms a constraint satisfaction problem with two-variables per constraint.

We apply a *decomposition lemma* (Lemma 4.5) due to Leighton and Rao [1999] to the constraint graph and remove some of its edges. The decomposition lemma guarantees that if the graph was small to start with (i.e., the proof was short), then after removing a tiny fraction of edges we are left with disconnected components of small radius.[5] The "decomposed" graph corresponds to a new linear inspective verifier whose soundness has not decreased significantly because it makes pretty much the same queries as the original verifier. Our

[5]The radius of a connected graph is the minimum maximal distance between any vertex and any other vertex (i.e., $rad(G) = \min_v \max_u d(u, v)$, where $d(u, v)$ denotes the distance between the vertices u and v.

analysis is completed (in Lemma 6.3) by showing that inspective PCPPs whose induced graph has radius R can be converted with no loss in soundness into (proofless) testers with query complexity $O(R)$. Summing up, if the proof is short to start with, then its decomposed graph has small radius, hence P has a (proofless) tester with small query complexity and good soundness.

The decomposition lemma mentioned above was previously used in a closely related context by Trevisan [2005] to provide algorithms for approximating unique games. We use it for similar purposes, namely, for analyzing constraint graphs, but our setting differs from that of Trevisan [2005] in three important aspects. First, in our setting the constraints that label edges of the constraint graph are not given to the verifier. Only the structure of the graph itself is known in advance. This difference also explains why the techniques relying on linear and semidefinite programming that were used in Khot [2002], Trevisan [2005], Charikar et al. [2006], and Gupta and Talwar [2006] do not seem appropriate for our setting. The second difference is that for our constraint graphs that are induced by 3-query verifiers, perfect completeness can be assumed. In the context of the unique games conjecture, assuming perfect completeness makes the problem trivial to solve. Finally, we use the decomposition lemma to construct a tester for the constraint graph rather than just decide if the constraint graph is close to be satisfiable.

We end our discussion of the proof techniques by pointing out Lemma 5.2, a generalization of the decomposition lemma to the case of nonunique constraint graphs. This lemma, which is required for obtaining our main result for general verifiers (Theorem 2.9), may be of independent interest. It says that any 2-CSP with ℓ constraints over the binary alphabet that is ε-far from being satisfiable, must contain a contradiction with $O(\log \ell/\varepsilon)$ constraints.

Organization. In Section 2, we give formal definitions and statements of our main results. We then in Section 3 construct (exponentially long) PCPPs with optimal soundness. We then prove the deficiency in soundness when one restricts to polynomially long PCPPs in Sections 4–7. Section 4 serves as warmup to latter sections for the purpose of illustrating the main idea of the proofs. As such, we prove only a weaker version of one of the main theorems in Section 4 and then prove the full theorems in the latter sections. In Section 8, we sketch how the nearly linear PCPs to [Ben-Sasson and Sudan 2008; Dinur 2007] can be adapted to yield nonadaptive 3-query linear verifiers with perfect completeness for linear properties.

2. DEFINITIONS AND MAIN RESULTS

We start by recalling the basic definitions and parameters of a PCPP system. Then, in Section 2.2 we introduce and define the *best soundness* and the *soundness deficiency* which are the quantities we use to measure the trade-off between proof length and soundness. In Section 2.3 we summarize our main results for the three cases of *(i)* general PCPPs over the binary alphabet, *(ii)* linear PCPPs over finite fields, and *(iii)* unique PCPPs.

Finally, in Section 2.4 we formally define *inspective* PCPPs and state the trade-offs for these PCPPs.

2.1 Probabilistically Checkable Proofs of Proximity (PCPPs)

Recall the basic task of *property testing*. Let Σ be a finite alphabet. A set $P \subseteq \Sigma^n$ is called a *property* of length n over Σ. We are interested in deciding the promise problem whose set of YES instances is P and whose set of NO instances is $\mathrm{NO}_{\delta_0} = \{w \in \Sigma^n \mid \delta(w, P) > \delta_0\}$, where $\delta(\cdot)$ denotes fractional Hamming distance and δ_0 is called the *proximity parameter*. The decision should be made after making a small number of queries into the input *word* $w \in \Sigma^n$ and the decision should be correct with high probability. (More information on property testing can be found in Goldreich et al. [1998] and in the survey Fischer [2001].)

In the context of *proximity testing* we try to decide the very same promise problem but the difference is that we allow oracle access to an additional *proof of proximity* $\pi \in \Sigma^\ell$ of length ℓ, and restrict the total number of queries that can be made to both w and π. A randomized query-restricted algorithm deciding the property testing problem is called a *tester* and when we allow oracle access to a proof we call it a *verifier*. The formal definition follows. (See Ben-Sasson et al. [2006] for more information on PCPPs.)

To simplify exposition we view w, π as functions from $[n] = \{1, \ldots, n\}$ and from $[n+1, n+\ell] = \{n+1, \ldots, n+\ell\}$ respectively to Σ and define the *word-proof pair* as the function $(w \circ \pi) : [n+\ell] \to \Sigma$ that is the concatenation of w and π. We call $(w \circ \pi)[i]$ a *word-symbol* whenever $i \leq n$ and a *proof symbol* when $i \in \{n+1, \ldots, n+\ell\}$. For a set of indices $I \subseteq [n+\ell]$ let $(w \circ \pi)|_I : I \to \Sigma$ denote the restriction of $w \circ \pi$ to I.

Definition 2.1 *(Verifier, Tester)*. A *query* of size q into a word of length n and proof of length ℓ is a pair $Q = (I, C)$ where $I \subseteq [n+\ell]$, $0 < |I| \leq q$ denotes the query's *index-set* and $C : \Sigma^I \to \{\mathsf{accept}, \mathsf{reject}\}$ is the query's *constraint*. Given word w and proof π let $Q(w \circ \pi) = C((w \circ \pi)|_I)$. A (q, n, ℓ)-*verifier* for a property of length n is a pair $\mathcal{V} = \langle \mathcal{Q}, D \rangle$ where

—$\mathcal{Q}$ is a finite set of queries of size at most q into a word of length n and proof of length ℓ.
—D is a distribution over $\mathcal{Q}$. We use $Q \sim_D \mathcal{Q}$ to denote that Q is sampled from $\mathcal{Q}$ according to distribution D.

A q-*tester* is a $(q, n, 0)$-verifier, that is, a verifier that queries only the input.

Often we will restrict our attention to a subclass of verifiers that use special kinds of constraints. In particular, we will be interested in *unique* and *linear* verifiers, defined next.

Definition 2.2 *(Linear and Unique verifiers)*. A query $Q = (I, C)$ is called *unique* if for every set of $|I| - 1$ answers to $|I| - 1$ queries, there exists a unique answer to the missing query that satisfies the constraint. Formally, for all $i_0 \in I$ and $a_{i_j} \in \Sigma, i_j \in I \setminus \{i_0\}$ there exists a unique $b \in \Sigma$ such that $C(a_{i_1}, \ldots, a_{i_0 - 1}, b, a_{i_0+1}, \ldots, a_{i_{|I|}}) = \mathsf{accept}$. A verifier is called *unique* if all its queries are unique. Let **uniqV** denote the set of unique verifiers.

A query is called $\mathbb{F}$-linear if $\Sigma = \mathbb{F}$ is a finite field and the set of assignments accepted by the query-constraint C are solutions to a linear constraint of the form $\sum a_i x_i = 0$ where $a_i \in \mathbb{F}$. A verifier is called $\mathbb{F}$-*linear* if all its queries are $\mathbb{F}$-linear. Let $\mathbb{F}$-**linV** denote the set of $\mathbb{F}$-linear verifiers.

Notice that without loss of generality, $\mathbb{F}$-linear verifiers are unique (this assumption is justified by removing from each query's index-set the set of indices upon which the query-constraint does not depend). The use of the term *unique* is justified by noticing that if we assign all but two indices of a unique constraint, the restricted binary constraint is "unique" according to the definition of this term by Khot [2002].

Informally, if a (q, n, ℓ)-verifier solves the promise problem associated with P "with high probability" then we say P "has a PCPP" (with query complexity q and length ℓ). The *completeness* and *soundness* parameters quantify the success probability of the verifier. The formal definition follows.

Definition 2.3 *(PCPP, Testability)*. A property $P \subset \Sigma^n$ is said to have a PCPP of length ℓ, query complexity q, completeness parameter c and soundness function $s : (0, 1] \to [0, 1]$ if there exists a (q, n, ℓ)-verifier for the property satisfying the following pair of requirements.

—*Completeness*. For all $w \in P$,

$$\max_{\pi \in \Sigma^\ell} \Pr_{Q \sim_D \mathcal{Q}} [Q(w \circ \pi) = \text{accept}] \geq c.$$

If $c = 1$, we say the verifier has *perfect* completeness.

—*Soundness*. For all $w \in \Sigma^n \setminus P$,

$$\min_{\pi \in \Sigma^\ell} \Pr_{Q \sim_D \mathcal{Q}} [Q(w \circ \pi) = \text{reject}] \geq s(\delta(w, P)),$$

where $\delta(w, P)$ denotes the minimal fractional Hamming distance between w and an element of P.

If P has a PCPP of length 0, query complexity q, completeness parameter c and soundness function s, we say that P is q-*testable* with completeness c and soundness s.

A verifier is said to be *adaptive* if its query indices depend on answers given to previous queries. The verifier defined above is *nonadaptive*. All results in this paper refer to nonadaptive verifiers with perfect completeness. We point out that all known PCPP constructions use nonadaptive verifiers and achieve perfect completeness so our deficiency bounds, stated next, apply to all of them (see Section 8 for further discussion).

2.2 Soundness Deficiency

We study the trade-off between proof length and soundness. Our aim is to show that short PCPPs cannot attain the same soundness as long ones. To quantify this trade-off we start by defining the *best soundness* that can be obtained by a class of verifiers with restricted proof length.

Definition 2.4 *(Best Soundness)*. Let $P \subseteq \Sigma^n$ be a property. For integers q, ℓ and $\delta \in [0, 1]$, define the *best soundness* $\mathcal{S}^P(q, \ell, \delta)$ to be the maximum—taken over all (q, n, ℓ)-verifiers $\mathcal{V}$—of the soundness of $\mathcal{V}$ with respect to inputs that are δ-far from P. Formally,

$$\mathcal{S}^P(q, \ell, \delta) = \max_{(q,n,\ell)\text{-verifiers}} \ \min_{w \circ \pi \in \Sigma^{n+\ell},\ \delta(w,P)=\delta} \ \Pr_{Q \sim_D \mathcal{Q}}[Q(w \circ \pi) = \text{reject}].$$

The *best tester soundness* is $\mathcal{S}^P(q, 0, \delta)$.

The best soundness with respect to a class of verifiers $\mathbf{V}$, denoted $\mathcal{S}^P_{\mathbf{V}}(q, \ell, \delta)$, is defined by taking the maximum above over all (q, n, ℓ)-verifiers in $\mathbf{V}$. Notice that $\mathcal{S}^P_{\mathbf{V}}(q, \ell, \delta) \leq \mathcal{S}^P(q, \ell, \delta)$.

The *soundness-deficiency*, defined next, is the reduction in best soundness incurred by 3-query verifiers limited to using short proofs.[6] As customary in computational complexity, we measure the asymptotic deficiency over a family of properties of increasing length. In the remark following the definition, we further explain the need for complexity assumptions.

Definition 2.5 *(Soundness deficiency)*. For $\mathcal{P} = \{P \subseteq \Sigma^n \mid n \in \mathbb{Z}^+\}$ a family of properties, $\mathbf{V}$ a class of verifiers and $\ell : \mathbb{Z}^+ \to \mathbb{Z}^+$ a function measuring proof length, let the *soundness-deficiency* be the function measuring the decrease in soundness due to limited proof length. Formally, it is a function from $(0, 1]$ to $[0, 1]$ defined by

$$\text{s-Def.}_{\mathbf{V}}[\mathcal{P}, \ell](\delta) = \liminf_{n\to\infty} \mathcal{S}^{P_n}_{\mathbf{V}}(3, \infty, \delta) - \mathcal{S}^{P_n}_{\mathbf{V}}\left(3, \ell(n), \delta\right).$$

For $\mathcal{C}$ a complexity class and $\mathcal{L}$ a family of complexity functions, we denote by s-Def.$_{\mathbf{V}}[\mathcal{C}, \mathcal{L}]$ the set of soundness deficiency functions, containing one function for each $\mathcal{P} \in \mathcal{C}$ and $\ell \in \mathcal{L}$ and by s-Def.$_{\mathbf{V}}[\mathcal{C}, \mathcal{L}](\delta)$ the set of evaluations of these functions at the point δ. Let in addition

$$\text{max-s-Def.}_{\mathbf{V}}[\mathcal{C}, \mathcal{L}] = \sup_{\delta\in(0,1]} \text{s-Def.}_{\mathbf{V}}[\mathcal{C}, \mathcal{L}](\delta)$$

be the supremum value that these functions obtain over all $\delta \in (0, 1]$. As before, whenever there is no restriction to a specific class of verifiers, the subscript $\mathbf{V}$ is omitted.

Remark 2.6 *(Complexity restrictions)*. If no restriction is placed on the complexity of $\mathcal{P}$, then one may end up with trivial and uninteresting results. For instance, if $P_n \subset \{0, 1\}^n$ is random, then with high probability any nondeterministic circuit deciding the promise problem associated with P_n requires size $2^{\Omega(n/\log n)}$. This implies that there are no constant query PCPPs with positive soundness and proof length $2^{o(n/\log n)}$. Thus, to get meaningful results,

[6]The definition could be naturally generalized to query complexity greater than 3. However, since all our results are limited to $q = 3$ we omit the query complexity parameter to simplify notation.

we focus on properties $\mathcal{P} \in \mathbf{P}$/poly for which the existence of polynomial-length PCPPs is guaranteed.

2.3 Summary of Results

In this section, we summarize our main results bounding the maximum soundness deficiency for three different classes of 3-query verifiers

—*general (binary) verifiers*. verifiers with general constraints over $\{0, 1\}$;
—*linear verifiers*. verifiers with linear constraints over any field $\mathbb{F}$;
—*unique verifiers*. verifiers with unique constraints over any Σ.

We will refer to linear properties and their duals in all our results.

Definition 2.7. Let $\mathbb{F}$ be a finite field and $n \in \mathbb{Z}^+$. A property $P \subseteq \mathbb{F}^n$ is said to be $\mathbb{F}$-linear if P is a linear space with respect to the field $\mathbb{F}$. We will usually refer to an $\mathbb{F}$-linear property as an $\mathbb{F}$-linear code and its elements as codewords.

For an $\mathbb{F}$-linear property $P \subseteq \mathbb{F}^n$, the dual property to P, denoted by $P^\perp \subseteq \mathbb{F}^n$ is defined as follows: $P^\perp = \{w \in F^n | \langle u, w\rangle = 0, \forall u \in P\}$. We will usually refer to $P^\perp$ as the dual code and its elements as dual codewords.

We will frequently refer to the following testing question (parametrized by δ): distinguishing codewords from words that are δ-far from the code. This question is not interesting when there are no words which are δ-far from the code. In this case, we will say that the code is *trivially δ-testable*. Thus, the testing question is interesting only for codes that are *not* trivially δ-testable. We say that a family $P = \{P_n | n \in \mathbb{Z}^+\}$ is not trivially δ-testable if there exists n_0 such that for all $n > n_0$, P_n is not trivially δ-testable.

Deficiency bounds are obtained by bounding from below the soundness of inspective verifiers that have access to long proofs and then bounding from above the soundness obtained by verifiers limited to short proofs. The next theorem shows the first bound, namely, that large soundness is obtainable if no restriction is placed on proof length. Its proof is based on the Fourier analytic approach introduced by Bellare et al. [1996] and appears in Section 3.

THEOREM 2.8 (BEST SOUNDNESS WITH UNBOUNDED PROOF LENGTH). *Let $\mathbb{F}_p$ be a prime field. Every $\mathbb{F}_p$-linear property $P \subseteq \mathbb{F}_p^n$ has a 3-query $\mathbb{F}_p$-linear verifier using a proof of length $\leq |\mathbb{F}|^{\dim(P)} \leq |\mathbb{F}|^n$ that achieves soundness function $s(\delta) \geq \delta$. Formally,*

$$\mathcal{S}^P_{\mathbf{linV}}\left(3, |\mathbb{F}_p|^{\dim(P)}, \delta\right) \geq \delta.$$

2.3.1 *Deficiency of Short PCPPs.* Our first main theorem says that for some properties, proofs of subexponential length incur constant soundness-deficiency. This deficiency can be reduced, but only at the expense of using exponentially long proofs.

THEOREM 2.9 (MAIN). *Let $\alpha \in (0, 1)$ be a positive constant and let $\mathcal{P} \triangleq \{P_n \subseteq \mathbb{F}_2^n : n \in \mathbb{Z}^+\}$ be a family of binary linear properties (codes) with dual* *distance*[7] *at least* αn *and such that for some* $\delta_0 \in (0,1)$ *they are not trivially* δ_0*-testable. The properties in* $\mathcal{P}$ *have no subexponential* 3*-query PCPP's achieving soundness larger than* $1/3$*. Namely, for every* $\varepsilon > 0$ *there are* $\beta > 0$ *and* $n_0 \in \mathbb{N}$ *such that for any property* $P_n \in \mathcal{P}$, $n > n_0$ *the following is satisfied for all* $\delta \in [0, \delta_0]$:

$$\mathcal{S}^{P_n}\left(3, 2^{\beta n}, \delta\right) \leq \frac{1}{3} + \varepsilon.$$

As a special case of Theorem 2.8, we show that every binary linear property $P \subseteq \mathbb{F}_2^n$ of dimension $k \leq n$ has a $(3, n, 2^k)$-verifier with soundness function $s(\delta) \geq \delta$. This implies constant deficiency for short PCPPs over the binary alphabet as formalized in the following corollary. To obtain it take $\mathcal{P}$ to be any family of linear properties $\mathcal{P} = \{P_n \subset \mathbb{F}_2^n\}$ satisfying $|P_n| \leq 2^{\varepsilon n}$ for $\varepsilon > 0$ and having dual distance $\Omega(n)$.

COROLLARY 2.10 (SOUNDNESS DEFICIENCY). *Let* **SUBEXP** *denote the set of subexponential functions, that is, functions satisfying* $f(n) = 2^{o(n)}$*. There exists a family* $\mathcal{P}$ *of linear properties over* $\mathbb{F}_2$ *such that*

$$\text{s-Def.}[\mathcal{P}, \mathbf{SUBEXP}](\delta) \geq \delta - \frac{1}{3}.$$

Consequently, since there are words that are roughly $\frac{1}{2}$*-far from the property* $\mathcal{P}$*, the maximal deficiency with sub-exponential proof length is at least* $\frac{1}{6}$*, that is,*

$$\text{max-s-Def.}[\mathbf{P}/poly, \mathbf{SUBEXP}] \geq \frac{1}{6}.$$

2.3.2 *Deficiency of Short Linear PCPPs.* Our next main theorem presents stronger deficiency bounds for linear PCPPs and states the following intuitively appealing implication: Every $\mathbb{F}$-linear property that is untestable—in the sense that testers with small query complexity for it have low soundness—is also unverifiable, that is, 3-query $\mathbb{F}$-linear verifiers with short proofs must incur a large loss in soundness. Limiting our attention to linear verifiers seems natural in light of the fact that all current PCPP constructions produce linear verifiers for linear properties, as argued in Section 8.

THEOREM 2.11 (MAIN, LINEAR CASE). *Let* $P \subseteq \mathbb{F}^n$ *be a* $\mathbb{F}$*-linear property. Let* $s[\ell](\delta)$ *denote the best soundness of a* $(3, n, \ell)$*-linear verifier for P, that is,* $s[\ell](\delta) = \mathcal{S}^P_{\mathbf{linV}}(3, \ell, \delta)$*. Let* $t[q](\delta)$ *denote the best soundness of a q-tester for P, that is,* $t[q](\delta) = \mathcal{S}^P(q, 0, \delta)$*. Then*

$$s[\ell](\delta) \leq \inf_{\varepsilon > 0}\left\{ t\left[\frac{36 \log \ell}{\varepsilon}\right](\delta) + \frac{1}{2} \cdot \left(1 - \frac{1}{|\mathbb{F}|} + \varepsilon\right)\right\}.$$

[7] The dual distance of a linear property P is the distance of the dual code $P^\perp$ which is equivalently the minimal support-size of a nonzero vector in the space dual to P. The dual distance of P is a lower bound for the query complexity of one-sided error local testing, since it is also equal to the lightest parity check vector for P.

Using Theorem 2.8 again for arbitrary prime p we get the following bound on the deficiency of linear verifiers.

COROLLARY 2.12 (SOUNDNESS DEFICIENCY, LINEAR CASE). *Let* **SUBEXP** *denote the set of subexponential functions, that is, functions satisfying* $f(n) = 2^{o(n)}$. *For every prime field* $\mathbb{F}_p$ *there exists a family of* $\mathbb{F}_p$*-linear properties* $\mathcal{P}$ *such that*

$$\text{s-Def.}_{\mathbb{F}_p-\mathbf{linV}}[\mathcal{P}, \mathbf{SUBEXP}](\delta) \geq \delta - \frac{1}{2} \cdot \left(1 - \frac{1}{p}\right).$$

Consequently, the maximal deficiency of linear verifiers with subexponential proofs is at least $\frac{1}{2} \cdot (1 - 1/p)$. *In other words,*

$$\text{max-s-Def.}_{\mathbb{F}_p-\mathbf{linV}}[\mathbb{F}_p - \textit{linear}, \mathbf{SUBEXP}] \geq \frac{1}{2} \cdot \left(1 - \frac{1}{p}\right).$$

We point out that even if we restrict our attention to families of linear properties with constant dual distance and not trivially testable, the soundness deficiency can be very large. This last point is explained in detail in the proof of Corollary 2.12.

2.3.3 *Deficiency of Short Unique PCPPs.* Our last main theorem bounds the soundness of arbitrary unique verifiers (of which linear verifiers are a special case).

THEOREM 2.13 (MAIN—UNIQUE CASE). *Let* $\alpha \in (0, 1)$ *be a positive constant and let* $\mathcal{P} \triangleq \{P_n \subseteq \mathbb{F}^n : n \in \mathbb{N}\}$ *be a family of* $\mathbb{F}$*-linear properties (codes) with dual distance at least* αn *and such that for some* $\delta_0 \in (0, 1)$ *they are not trivially* δ_0*-testable. For every* $\varepsilon > 0$*, there exists a* $\beta > 0$ *and* $n_0 \in \mathbb{N}$ *such that for any property* $P_n \in \mathcal{P}$, $n > n_0$ *the following is satisfied for all* $\delta \in (0, \delta_0]$:

$$\mathcal{S}^{P_n}_{\mathbf{uniqV}}\left(3, 2^{\beta n}, \delta\right) \leq \frac{2(1+\varepsilon)}{3} \cdot \left(1 - \frac{1}{|\mathbb{F}|}\right).$$

As before, we use the fact that for prime p, every $\mathbb{F}_p$-linear property has a high-soundness linear (hence unique) verifier, as long as proof length is unlimited. This implies the following bound on deficiency of unique verifiers.

COROLLARY 2.14 (SOUNDNESS DEFICIENCY, UNIQUE CASE).*Let* **SUBEXP** *denote the set of sub-exponential functions, that is, functions satisfying* $f(n) = 2^{o(n)}$. *For every prime field* $\mathbb{F}_p$ *there exists a family of* $\mathbb{F}_p$*-linear properties* $\mathcal{P}$ *such that*

$$\text{s-Def.}_{\mathbb{F}_p-\mathbf{uniqV}}[\mathcal{P}, \mathbf{SUBEXP}](\delta) \geq \delta - \frac{2}{3} \cdot \left(1 - \frac{1}{p}\right).$$

Consequently, the maximal deficiency of unique verifiers with subexponential proofs is at least $\frac{1}{3} \cdot (1 - 1/p)$*, or formally,*

$$\text{max-s-Def.}_{\mathbb{F}_p-\mathbf{uniqV}}[\mathbb{F}_p - \textit{linear}, \mathbf{SUBEXP}] \geq \frac{1}{3} \cdot \left(1 - \frac{1}{p}\right).$$

2.4 Inspective PCPPs

The deficiency bounds stated above follow from much stronger bounds on the soundness achieved by a special family of *inspective* verifiers, defined next. Informally, inspective verifiers are called so because every 3-query they make *inspects* the word w in at least one location.

Definition 2.15 *(Inspective PCPP)*. A query $Q = (I, C)$ is called *inspective* if its index-set involves at most two symbols of the proof, that is, $\left|I \cap [n+1, n+\ell]\right| \leq 2$. We refer to the above quantity as the inspective size (i-size) of the query Q.

A verifier $\mathcal{V} = \langle \mathcal{Q}, D\rangle$ is said to be *inspective* if all its queries are inspective. We denote by $\mathbf{V_i}$ the set of inspective verifiers, by $\mathbf{linV_i}$ the set of inspective linear verifiers and by $\mathbf{uniqV_i}$ the set of inspective unique verifiers.

A property $P \subset \Sigma^n$ is said to have an inspective PCPP of length ℓ, query complexity q, and soundness function $s : (0, 1] \to [0, 1]$ if there exists a (q, n, ℓ)-inspective verifier with soundness function s. Inspective linear PCPPs and inspective unique PCPPs are similarly defined.

Remark 2.16. We note that the linear verifier mentioned in Theorem 2.8 is in fact an inspective verifier that makes inspective queries of size exactly two. Thus, $\mathcal{S}^P_{\mathbf{linV_i}}\left(3, |\mathbb{F}_p|^{\dim(P)}, \delta\right) \geq \delta$.

The main technical components in the proofs of Theorems 2.9, 2.11, and 2.13 are the following respective upper bounds on the soundness of inspective verifiers limited to querying only short proofs. The proof of these theorems, which are deferred to the appendix, rely on defining a natural *inspective graph* (Definition 5.5) and applying a decomposition lemma to it. In the case of general PCPPs over the binary alphabet we use Lemma 5.2 and in the remaining two cases we apply Lemma 4.5 which is very similar to the original decomposition lemma of Leighton and Rao [1999].

Definition 2.17 *(d-Universal Properties)*. A property $P \subseteq \Sigma^n$ is *d-universal* if for all subsets $I \subset [n], |I| \leq d$, the restriction of P to I equals Σ^I, that is, $\{w|_I \mid w \in P\} = \Sigma^I$. Observe that any linear property P with dual distance d is $(d-1)$-universal.[8]

THEOREM 2.18 (BEST SOUNDNESS WITH INSPECTIVE VERIFIERS). *Let $P \subseteq \{0,1\}^n$ be a d-universal property, and let $q \in \mathbb{Z}^+$. Let s_i denote the best soundness of a (q, n, ℓ)-inspective verifier for P, that is, $s_i(\delta) = \mathcal{S}^P_{\mathbf{V_i}}(q, \ell, \delta)$. Then for every $\delta \in [0, 1]$,*

$$s_i(\delta) \leq \inf_{\varepsilon > 0}\left\{\frac{4\log(\varepsilon^{-2}(n+\ell))}{\frac{d}{q-1} - 2} + \varepsilon\right\}.$$

[8]Recall (from Footnote 7) that the dual distance of a linear property P is d if the minimal support-size of a nonzero vector in the space dual to P is exactly d. This, in turn, implies that the restriction of the property to any $d-1$ coordinates is Σ^{d-1}.

THEOREM 2.19 (BEST SOUNDNESS WITH INSPECTIVE LINEAR VERIFIERS). *Let $P \subseteq \mathbb{F}^n$ be a $\mathbb{F}$-linear property. Let $s_{\mathrm{i}}(\delta)$ denote the best soundness of a $(3, n, \ell)$-linear inspective verifier for P, that is, $s_{\mathrm{i}}(\delta) = \mathcal{S}^P_{\mathbf{linV_i}}(3, \ell, \delta)$. Let $t[q](\delta)$ denote the best soundness of a q-tester for P, that is, $t[q](\delta) = \mathcal{S}^P(q, 0, \delta)$. Then*

$$s_{\mathrm{i}}(\delta) \leq \inf_{\varepsilon>0}\left\{ t\left[\frac{36\log\ell}{\varepsilon}\right](\delta) + \varepsilon \right\}.$$

THEOREM 2.20 (BEST SOUNDNESS WITH INSPECTIVE UNIQUE VERIFIERS). *Let $P \subseteq \Sigma^n$ be a property. Let s_{i} denote the best soundness of a $(3, n, \ell)$-unique inspective verifier for P, that is, $s_{\mathrm{i}}(\delta) = \mathcal{S}^P_{\mathbf{uniqV_i}}(3, \ell, \delta)$. Let $t[q](\delta)$ denote the best soundness of a q-tester for P, that is, $t[q](\delta) = \mathcal{S}^P(q, 0, \delta)$. Then for any $s_{\mathrm{i}}(\delta) > \varepsilon$*

$$s_{\mathrm{i}}(\delta) \leq \inf_{\varepsilon>0}\left\{ 4t\left[\frac{10\log\ell}{\left(s_{\mathrm{i}}(\delta)-\varepsilon\right)\varepsilon}\cdot \ln(2|\Sigma|)\right](\delta) + \varepsilon \right\}.$$

3. LONG PCPPS WITH BEST POSSIBLE SOUNDNESS

In this section, we will prove that any $\mathbb{F}_p$-linear property $P \subseteq \mathbb{F}_p^n$ over a prime field $\mathbb{F}_p$ has a 3-query linear inspective PCPP of length at most $p^{\dim(P)}$. Furthermore, the soundness of this verifier on words that are δ-far from P satisfies $s(\delta) \geq \delta$, thereby proving Theorem 2.8. We point out that if P is "nontrivial", meaning there is no $i \in [n]$ such that $w_i = 0$ for all $w \in P$, then the soundness of linear verifiers can be shown to be bounded from above by $1-1/p$. This shows that for δ approaching $1-1/p$ the term "best possible" aptly describes the soundness function of our verifier.

3.1 Fourier Transform—Preliminaries

We interpret $\mathbb{Z}_p$ as the multiplicative group of p^{th} complex roots of unity. Let $\omega \triangleq e^{\frac{2\pi i}{p}}$, and let $\mu_p = \{\omega^0, \omega^1, \ldots, \omega^{p-1}\}$ be the p^{th} complex roots of unity. For every $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}_p^n$ we define the function $\chi_\alpha : \mathbb{Z}_p^n \to \mathbb{C}$ as

$$\chi_\alpha(x_1, \ldots, x_n) = \omega^{(x\cdot\alpha)} = \omega^{\sum_i x_i\alpha_i}.$$

For two functions $f : \mathbb{Z}_2^n \to \mathbb{C}$ and $g : \mathbb{Z}_p^n \to \mathbb{C}$, we define their *inner product* as

$$\langle f, g\rangle \triangleq \frac{1}{p^n}\sum_{x\in\mathbb{Z}_p^n} f(x)\overline{g(x)} = \mathbb{E}_{x\in\mathbb{Z}_p^n}\left[f(x)\overline{g(x)}\right].$$

It is easy to verify that the functions $\chi_\alpha : \mathbb{Z}_p^n \to \mathbb{C}$ are *orthonormal* with respect to this inner product. Namely, that for every $\alpha \in \mathbb{Z}_p^n$,

$$\langle \chi_\alpha, \chi_\alpha\rangle = 1$$

and for every $\alpha, \beta \in \mathbb{Z}_p^n$, $\alpha \neq \beta$,

$$\langle \chi_\alpha, \chi_\beta \rangle = 0.$$

Therefore, the functions $\{\chi_\alpha\}_{\alpha \in \mathbb{Z}_p^n}$ form a *basis* for the space of functions $f : \mathbb{Z}_p^n \to \mathbb{C}$ (the dimension of which is exactly p^n). Hence, every function $f : \mathbb{Z}_p^n \to \mathbb{C}$ can be written as a linear combination of the elements of this basis

$$f(x) = \sum_\alpha \hat{f}_\alpha \chi_\alpha(x),$$

where the coefficients $\hat{f}_\alpha$ (called the *Fourier coefficients* of f) are defined as follows:

$$\hat{f}_\alpha = \langle f, \chi_\alpha \rangle.$$

We have the following equality (*Parseval's identity*)

$$\sum_{\alpha \in \mathbb{Z}_p^n} |\hat{f}_\alpha|^2 = \langle f, f \rangle = \mathbb{E}_{x \in \mathbb{Z}_p^n}\left[|f(x)|^2\right]$$

and in particular, if $f : \mathbb{Z}_p^n \to \mu_p$, then $\sum_{\alpha \in \mathbb{Z}_p^n} |\hat{f}_\alpha|^2 = 1$ and for all α, $|\hat{f}_\alpha| \leq 1$.

We also have the following useful lemma.

LEMMA 3.1. *Let $\eta \in \mu_p$ be a p^{th} root of unity. Then the sum $\sum_{i \in [p] \setminus \{0\}} \eta^i$ equals $p - 1$ if $\eta = 1$, and it equals -1 for any $\eta \neq 1$.*

3.2 Proof of Theorem 2.8

Let $P \subseteq \mathbb{Z}_p^n$ be a $\mathbb{Z}_p$-linear space of dimension k. Fix $G \in \mathbb{Z}_p^{n \times k}$ to be a matrix such that P equals the span of columns of G so that

$$P = \{w : \exists x \in \mathbb{Z}_p^k \text{ such that } w = Gx\}.$$

Let $g_i \in \mathbb{Z}_p^k$ denote the i^{th} row of G. Thus, if $w = Gx$, we have that $w_i = (g_i \cdot x)$ for all i. In the terminology of error correcting codes G is a *generating matrix* for the $[n, k]_p$-code P and so we refer to elements $w \in P$ as "codewords."

For every $x \in \mathbb{Z}_p^k$ we denote by $H_x : \mathbb{Z}_p^k \to \mathbb{C}$ the *Hadamard* encoding of x, which is defined as $H_x(y) = \omega^{(x \cdot y)} = \omega^{\sum_i x_i y_i}$. The function H_x can be explicitly written as a vector of values (of the exponents) in $\mathbb{Z}_p^{p^k}$. However, the following *folded* representation of H_x will be simpler to analyze. We partition the set $\mathbb{Z}_p^k \setminus \{0\}$ into disjoint classes of the form $\left\{jy : j \in \{1, \ldots, p-1\}\right\}$, each of size $p - 1$ (each class corresponds to some element $y \in \mathbb{Z}_p^k \setminus \{0\}$). Then for each of these classes we choose one of its elements as a representative, and eventually we keep the values of H_x only for these representative elements. Now we can extract the value of $H_x(y)$ for every $y \in \mathbb{Z}_p^k$ as follows.

—If $y = 0$ then $H_x(y) = \omega^0 = 1$.

—If y is one of the representatives, then we read the appropriate value according to the folded encoding.

—Otherwise, we find a representative u and j such that $y = ju$, we read $H_x(u)$ by the previous rule, and set $H_x(y) = \left(H_x(u)\right)^j$.

Since H_x is a linear function, these extraction rules are consistent with the original function.

For every codeword $w \in P$, we denote by $x_w \in \mathbb{Z}_p^k$ the vector that satisfies $w = Gx_w$, and we denote by $\pi_w : \mathbb{Z}_p^k \to \mathbb{C}$ the Hadamard encoding of x_w, that is, $\pi_w = H_{x_w}$. We assume that π_w is represented in its folded form, so the actual representation of π_w takes $\frac{p^k-1}{p-1}$ values in $\mathbb{Z}_p$. Note that the value of π_w on 0 is not kept in the folded representation.

Consider the following 3-query linear inspective verifier V for P

INSPECTIVE VERIFIER V
INPUT (AS ORACLES): $w \in \mathbb{Z}_p^n$, $\pi : \mathbb{Z}_p^k \to \mathbb{C}$

(1) Choose $y \in \mathbb{Z}_p^k$ and $i \in [n]$ uniformly at random
(2) Output accept if and only if $\pi(y)\omega^{w_i} = \pi(y + g_i)$.

CLAIM 3.2. *The inspective verifier V satisfies the following properties*:

—Completeness: *If* $w \in P$ *and* $\pi = \pi_w$ *then* $\Pr\left[V^{(w,\pi_w)} = \text{accept}\right] = 1$

—Soundness: *For any* $w \in \mathbb{Z}_p^n$ *and any (folded)* $\pi \in \mathbb{Z}_p^{\frac{p^k-1}{p-1}}$, $\Pr\left[V^{(w,\pi)} = \text{reject}\right] \geq \delta(w, P)$

Before proceeding to the proof of Claim 3.2, we first observe that Theorem 2.8 follows immediately from the above claim.

PROOF. For a codeword $w = G \cdot x_w \in P$ and a legal proof $\pi_w = H_{x_w}$ we have $w_i = (g_i \cdot x_w)$, and together with the fact that H_{x_w} is linear we have

$$\pi_w(y + g_i) = \pi_w(y) \cdot \pi_w(g_i) = \pi_w(y) \cdot \omega^{(g_i \cdot x_w)} = \pi(y) \cdot \omega^{w_i}$$

thus, the completeness condition is satisfied. Now we have to prove that the soundness of V is as required.

In the following, we use the fact that the function π is represented in folded form, and hence for every $y \in \mathbb{Z}_p^k$ and $j \in [p]$ we have $\pi(jy) = \left(\pi(y)\right)^j$. Denote by s the soundness of V, i.e., the probability it rejects a word-proof pair. We are going to express s in terms of $\delta(w, P)$ by making some manipulations on the Fourier expansion of π. According to the description of algorithm V,

$$1 - s = \Pr_{y,i}[\pi(y)\omega^{w_i}\overline{\pi(y + g_i)} = 1]$$

and according to Lemma 3.1, if η is a p^{th} root of unity, then the sum $\sum_{j\in[p]\setminus\{0\}} \eta^j$ equals $p-1$ when $\eta = 1$, and it equals -1 otherwise. Thus for all pairs (w, π) we have

$$(p-1)(1-s) - s = \mathbb{E}_{y,i}\Big[\sum_{j\in[p]\setminus\{0\}} \Big(\pi(y)\omega^{w_i}\overline{\pi(y+g_i)}\Big)^j\Big] =$$

$$\mathbb{E}_{y,i}\Big[\sum_{j\in[p]\setminus\{0\}} \pi(jy)\omega^{jw_i}\overline{\pi(jy+jg_i)}\Big] =$$

$$\mathbb{E}_{y,i}\Big[\sum_{j\in[p]\setminus\{0\}} \omega^{jw_i}\big(\sum_\alpha \hat{\pi}_\alpha\chi_\alpha(jy)\big)\big(\sum_\beta \overline{\hat{\pi}_\beta\chi_\beta(jy)\chi_\beta(jg_i)}\big)\Big] =$$

$$\sum_{\alpha,\beta}\hat{\pi}_\alpha\overline{\hat{\pi}_\beta}\sum_{j\in[p]\setminus\{0\}}\mathbb{E}_i\Big[\omega^{jw_i}\overline{\chi_\beta(jg_i)}\Big]\mathbb{E}_y\Big[\chi_\alpha(jy)\overline{\chi_\beta(jy)}\Big] =$$

by the orthonormality of the character functions

$$\sum_\alpha |\hat{\pi}_\alpha|^2\sum_{j\in[p]\setminus\{0\}}\mathbb{E}_i\Big[\omega^{jw_i}\overline{\chi_\alpha(jg_i)}\Big] =$$

$$\sum_\alpha |\hat{\pi}_\alpha|^2\mathbb{E}_i\Big[\sum_{j\in[p]\setminus\{0\}}\omega^{jw_i}\overline{\chi_\alpha(jg_i)}\Big] =$$

$$\sum_\alpha |\hat{\pi}_\alpha|^2\mathbb{E}_i\Big[\sum_{j\in[p]\setminus\{0\}}\Big(\omega^{w_i}\overline{\chi_\alpha(g_i)}\Big)^j\Big] =$$

$$\sum_\alpha |\hat{\pi}_\alpha|^2\mathbb{E}_i\Big[\sum_{j\in[p]\setminus\{0\}}\Big(\omega^{w_i-\alpha\cdot g_i}\Big)^j\Big] =$$

by Lemma 3.1, for every i such that $w_i = \alpha g_i$ (the agreeing indices) the sum $\sum_{j\in[p]\setminus\{0\}}\left(\omega^{w_i-\alpha\cdot g_i}\right)^j$ evaluates to $p-1$, and for all other indices i, this sum evaluates to -1; therefore, the expression equals

$$\sum_\alpha |\hat{\pi}_\alpha|^2\Big(\big(1-\delta(w, G\alpha)\big)(p-1) - \delta(w, G\alpha)\Big) \le$$

$$\Big(\big(1-\delta(w, P)\big)(p-1) - \delta(w, P)\Big)\sum_\alpha |\hat{\pi}_\alpha|^2 \le$$

$$p - 1 - p\delta(w, P).$$

The last inequality is due to Parseval's identity. To conclude, we have $(p-1) - ps \le (p-1) - p\delta(w, P)$, or simply $s \ge \delta(w, P)$ as required. □

4. PROOF OF LENGTH-SOUNDNESS TRADE-OFF FOR LINEAR PCPPS OVER $\mathbb{F}_2$

In this section, we sketch the proof of length-soundness trade-off for linear PCPPs over the binary field. Although we consider only a special case, this section captures most of the ideas that are used for proving our main results. As mentioned earlier, this section serves only as a warm-up to the latter sections and the reader may skip the section.

THEOREM 4.1 (SPECIAL CASE OF THEOREMS 2.9 AND 2.11). *Given any $0 < \varepsilon < 1$ and a subexponential function $\ell \in 2^{o(n)}$, let $\mathcal{C} \subseteq \mathbb{F}_2^n$ be an $\mathbb{F}_2$-linear property (code) with dual distance at least $\left(\frac{c \log \ell}{\varepsilon}\right)$ (c large enough constant) and such that for some $\delta_0 \in (0, 1/2)$ they are not trivially δ_0-testable. Let $s[\ell](\delta)$ denote the best soundness of a $(3, n, \ell)$-linear verifier for $\mathcal{C}$, that is, $s[\ell](\delta) = \mathcal{S}^{\mathcal{C}}_{\mathbf{linV}}(3, \ell, \delta)$. Then for every $\delta \leq \delta_0$,*

$$s[\ell](\delta) \leq \frac{1}{3} + \varepsilon.$$

Consequently, the soundness deficiency of linear verifiers with subexponential proofs is $1/6 - o(1)$.

The following lemma is the main ingredient in the proof of Theorem 4.1.

LEMMA 4.2 (INSPECTIVE VERIFIABILITY IMPLIES TESTABILITY). *Let $\mathcal{C} \subseteq \mathbb{F}_2^n$ be any linear code. For every $q, \ell \in \mathbb{Z}^+$, $\varepsilon \in (0, 1]$ and $s : (0, 1] \to [0, 1]$, the following holds. If $\mathcal{C}$ has a linear q-query inspective PCPP of length ℓ and soundness function s, then $\mathcal{C}$ has a $O(\frac{q \log \ell}{\varepsilon})$-tester with soundness function $s - \varepsilon$.*

Before proving the lemma, we outline the proof of Theorem 4.1. Fix a $(3, n, \ell)$-linear verifier $\mathcal{V} = \langle \mathcal{Q}, D \rangle$ for $\mathcal{C}$, and denote by s the soundness function of $\mathcal{V}$. Let μ denote the probability that $\mathcal{V}$ makes an inspective query, namely, the probability (with respect to distribution D) that $\mathcal{V}$ probes at least one word bit.

Now, for two possible ranges of μ and for any $\delta \leq 1/2$ we design a "fooling" word-proof pair $(w \circ \pi)$, such that: (i) $\delta(w, \mathcal{C}) \geq \delta$; and (ii) $\mathcal{V}$ accepts $(w \circ \pi)$ with probability $2/3 - \varepsilon$, concluding the proof.

Case $\mu \leq 2/3$. Fix any $\delta \leq 1/2$ and consider the distribution over word-proof pairs $(w \circ \pi)$, where π is a legitimate proof of some fixed codeword $w' \in \mathcal{C}$ and $w = (w_\delta + r)$ is a sum of a fixed δ-far (from $\mathcal{C}$) word w_δ and a randomly chosen codeword $r \in \mathcal{C}$. It is not hard to show that

(1) w is δ-far from $\mathcal{C}$;

(2) for any subset $I \in [n]$ of at most 3 coordinates of w, the $2^{|I|}$ possible values of w_I are distributed uniformly (over the choices of r).

Clearly, all noninspective queries are accepted with probability 1 since π is a legitimate proof. In addition, item 2 implies that the inspective queries are accepted with probability 1/2, since the constraints are all linear. Therefore, $s(\delta) \leq \mu/2 \leq 1/3$.

Case $\mu > 2/3$. Let $\mathcal{V}_i$ be a $(3, n, \ell)$-linear inspective verifier derived from $\mathcal{V}$ as follows: $\mathcal{V}_i$ picks a query $Q \in \mathcal{Q}$ according to distribution D. If Q is inspective then $\mathcal{V}_i$ proceeds exactly as $\mathcal{V}$. Otherwise, $\mathcal{V}_i$ immediately accepts (without making the query Q). Let s_i denote the soundness function of $\mathcal{V}_i$. Since $\mathcal{V}_i$ is inspective, by Lemma 4.2 we get that for any $\left(\frac{3 \log \ell}{\varepsilon}\right)$-query tester T for $\mathcal{C}$

having soundness function s_T, $s_i \leq s_T + \varepsilon$. We also know (by definition of $\mathcal{C}$'s dual distance and non-trivial testability) that any such tester (with perfect completeness) has soundness $s_T \equiv 0$. Therefore, $s(\delta) \leq \mu \cdot (0+\varepsilon)+1-\mu \leq 1/3+\varepsilon$ for any $\delta \leq 1/2$.

4.1 Proof of Lemma 4.2

As mentioned in the introduction, we are going to view an inspective q-query linear verifier as a graph. An inspective graph (defined below) is a representation of an inspective verifier in the sense that a single invocation of the verifier corresponds to picking a random edge in the graph and making the set of queries given by the names of the end-vertices and the edge-label. The definition that follows is a special case of the more general definition in Section 5.3. This definition is only for this warm-up section.

Definition 4.3 *(Inspective Graph—special case for* $\mathbb{F}_2$*).* A (q, n, ℓ)-inspective graph is a triplet $\mathcal{G} = (V, E, L_E)$ where $(V = \{0, 1, \ldots, l\}, E)$ is an undirected multigraph graph and $L_E : E \to (\mathbb{F}_2)^n_{\leq q}$ is a mapping of the edges to $\mathbb{F}_2$-vectors of dimension n and weight at most q.

A word $w \in \mathbb{F}_2^n$ induces a labeling $L_E^{(w)} : V \to \mathbb{F}_2$ as follows: $L_E^{(w)}(e) = \langle L_E(e), w\rangle = \sum_{i=1}^n L_E(e)[i] \cdot w[i]$.

A labeling $\pi : V \to \mathbb{F}_2$ is said to satisfy edge $e = (u, v)$ with respect to w if $\pi(u) + \pi(v) + L_E^{(w)} = \pi(u) + \pi(v) + \langle L_E(e), w\rangle = 0$. A labeling $\pi : V \to \mathbb{F}_2$ is said to α-satisfy $\mathcal{G}$ with respect to w if it satisfies an α-fraction of the edges with respect to w.

$\mathcal{G}$ is said to be a (q, n, ℓ)-inspective proof graph for property P with soundness function $s : (0, 1] \to [0, 1]$ if the following two conditions are satisfied.

Perfect Completeness. For all $w \in P$ there exists a a labeling $\pi : V \to \mathbb{F}_2$ such that π 1-satisfies $\mathcal{G}$ with respect to w.

Soundness. For all $w \in \Sigma^n$, no labeling $\pi : V \to \mathbb{F}_2$ $(1 - s(\delta(w, P))$-satisfies $\mathcal{G}$ with respect to w where $\delta(w, P)$ denotes the minimal fractional Hamming distance between w and an element of P.

The correspondence between linear inspective PCPPs and inspective graphs is as follows. First, assume without loss of generality that the verifier $\mathcal{V} = \langle \mathcal{Q}, D\rangle$ has uniform distribution D over $\mathcal{Q}$. This can be assumed by replacing $\mathcal{Q}$ with a multiset of queries where the number of copies of a query Q reflects the probability with which the Q is performed. The vertices $V \setminus \{0\}$ correspond to the ℓ locations of the proof. The vertex 0 is a special vertex which corresponds to the bit 0. Any labeling of the vertices V (that satisfies $\pi(0) = 0$) corresponds to a proof. However, we may assume $\pi(0) = 0$ without loss of generality for the following reason. If a labeling π α-satisfies $\mathcal{G}$, so does the labeling $\pi + b$ defined as follows: $(\pi + b)(v) = \pi(v) + b$. Hence, we might assume that the labeling π satisfies $\pi(0) = 0$.

Recall that for a given query, i-size denotes the number of proof bits read by that query. The edges of the graph correspond to the (inspective) tests of the verifier. Non-self-loop edges in $E \cap (V \setminus \{0\} \times V \setminus \{0\})$ correspond to inspective queries of i-size 2, non-self-loop edges in $E \cap (V \times \{0\})$ to inspective queries of i-size 1 while the self-loop edges correspond to inspective queries of i-size 0. On input w, the verifier chooses an edge of the graph uniformly at random and checks if the labeling π satisfies the edge with respect to w. The multiplicity of an edge is proportional to the probability with which the PCPP verifier chooses the corresponding test.

CLAIM 4.4. *Let $\mathcal{G} = (V, E, L_E)$ be a (q, n, ℓ)-inspective proof graph for some linear code $\mathcal{C}$. Suppose the vertices $v_1, v_2, \ldots, v_k, v_{k+1} = v_1$ form a cycle in the graph (V, E), then the vector $\sum_{i=1}^{k} L_E(v_i, v_{i+1})$ is a member of the dual code $\mathcal{C}^{\perp}$.*

Before proceeding we need some notation. For any graph G, let $V(G)$ and $E(G)$ denote the set of vertices and set of edges respectively of the graph G. For any subset $V' \subseteq V$ of vertices, let $G(V')$ denote the induced subgraph of G on the vertex set V'. Also, let $E(V') = E(G(V'))$. Similarly, let $E(V', V \setminus V')$ denote the set of edges between V' and $V \setminus V'$ (i.e., $E(V', V \setminus V') = E \cap (V' \times (V \setminus V'))$. For any connected graph G, define the radius of G (denoted by $\mathrm{r}ad(G)$) as follows:

$$\mathrm{r}ad(G) = \min_{v \in V} \max_{u \in V} d(u, v),$$

where $d(u, v)$ denotes the length of the shortest path between vertices u and v. Notice that for any connected graph, the distance between any two vertices is at most twice the radius of the graph.

Lemma 4.2 is proved by first showing that the inspective proof graph can be decomposed into components with small radii, and then transforming any inspective proof graph with components of small radii into a tester (Lemma 4.6).

LEMMA 4.5 (DECOMPOSITION LEMMA [LEIGHTON AND RAO 1999]). *For every $\varepsilon \in (0, 1)$ and every multigraph $G = (V, E)$, there exists a subset of edges $E' \subseteq E$ of size at most $\varepsilon|E|$, such that every component of the graph $G_{\mathsf{Decomp.}} = (V, E \setminus E')$ has radius strictly less than $\log |V|/\varepsilon$. The graph $G_{\mathsf{Decomp.}}$ is said to be an ε-*decomposition *of G.*

The proof of Lemma 4.5 is deferred to Section 6.2. Now we show how to convert an inspective graph into a tester. The query complexity of the tester will be bounded by the length of the cycles in the graph. Thus, a graph with small radius will result in a tester of low query complexity.

LEMMA 4.6 LOW RADIUS IMPLIES TESTABILITY. *Let $\mathcal{C} \subseteq \mathbb{F}_2^n$ be a linear code and $\mathcal{G} = (V, E, L)$ be a (q, n, ℓ)-inspective proof graph for the code $\mathcal{C}$ with soundness function s. If each of the components of the graph (V, E) has radius smaller than r, then $\mathcal{C}$ is $2qr$-testable with soundness function s.*

PROOF. Let $\mathcal{G} = (V, E, L)$ be a (q, n, ℓ)-inspective proof graph for $\mathcal{C}$ with soundness s such that each component of the graph $G = (V, E)$ has radius smaller than r. Having radius smaller than r implies that there exists a spanning forest $F = (V, E')$ of G such that the height of each tree in F is less than r.

Consider the mapping $\tau : E \to \mathbb{F}_2^n$ of the edges to $\mathbb{F}_2$-vectors of length n defined as follows: If $e \in E(F)$, then $\tau(e) = 0$. Otherwise, $E(F) \cup \{e\}$ contains a unique cycle C. Then, define $\tau(e) = \sum_{e \in C} L(e)$. Since each tree of F is of height less than r, any such cycle C is of length at most $2r$. Also, from the definition of inspective proof graphs we have that $L(e)$ is a vector of weight at most q. Hence, $\tau(e)$ is a vector of weight at most $2qr$ and τ is a mapping from E to $(\mathbb{F}_2^n)_{\leq 2qr}$.

We define a tester $\mathcal{T}_{\mathcal{G}}$ based on the graph $\mathcal{G}$ as follows: On input $w \in \mathbb{F}^n$, it selects an edge e uniformly at random from $E(G)$ and checks if $\langle \tau(e), w \rangle = 0$. If yes it accepts, e it rejects. We now prove that $\mathcal{T}_{\mathcal{G}}$ is a $2qr$-tester for $\mathcal{C}$ with soundness function s.

Query Complexity. Since each $\tau(e)$ is of weight at most $2qr$, the tester queries at most $2qr$ locations of the word w.

Completeness. Suppose $w \in \mathcal{C}$. We have by Claim 4.4, that for each cycle C in G, we have $\sum_{e \in C} L(e)$ is a member of the dualcode $\mathcal{C}^{\perp}$. Therefore, $\langle \tau(e), w \rangle = 0$ for all edges e in G. In other words, the tester $\mathcal{T}_{\mathcal{G}}$ accepts with probability 1.

Soundness. Let w be any word. Consider the labeling $\pi : V \to \mathbb{F}_2$ defined as follows: For each tree T in the forest F, choose an arbitrary vertex v in T and set $\pi(v) = 0$. For any other vertex u in the tree, let $v = v_0, v_1, \ldots, v_k = u$ be the unique path in the tree T from v to u. Define $\pi(u) = \langle \sum L(v_i, v_{i+1}), w \rangle$. It is easy to check that if the tester $\mathcal{T}_{\mathcal{G}}$ accepts w with probability α, then the labeling π α-satisfies $\mathcal{G}$ with respect to π. The soundness of the tester now follows from the soundness of the inspective proof graph $\mathcal{G}$. □

Lemma 4.2 now easily follows from the Decomposition Lemma and Lemma 4.6.

5. PROOF OF LENGTH-SOUNDNESS TRADE-OFF

The proof is organized as follows. In Section 5.1 we define constraint graphs, which are later used to analyze inspective verifiers. In Section 5.2 we prove an auxiliary lemma that allows us to convert any verifier $\mathcal{V} = \langle \mathcal{Q}, D \rangle$ into a verifier $\mathcal{V}' = \langle \mathcal{Q}', D' \rangle$ such that $\mathcal{V}'$ achieves almost the same soundness as $\mathcal{V}$, but the size of $\mathcal{Q}$ is linear in the length of the proof, and the distribution D' is uniform over $\mathcal{Q}$. In Section 5.3 we prove that the soundness of inspective verifiers goes to zero as long as the proof length is subexponential. Based on these, we prove Theorem 2.9 in Section 5.4 and complete several missing proofs in Section 5.5.

5.1 Constraint Graphs and the Generalized Decomposition Lemma

Definition 5.1 *(Constraint Graphs).* A constraint graph is a pair $\phi = (G, C)$, where $G = (V, E)$ is a directed multigraph and $C = \Big\{c_e : \{0,1\}^2 \to \{\text{accept}, \text{reject}\} \mid e \in E\Big\}$ is a set of binary constraints associated with the edges of G.

If an assignment $\pi : V \to \{0, 1\}$ satisfies a δ-fraction of the constraints in ϕ then we say that π δ-satisfies ϕ. Namely, π is δ-satisfying if $\Big|\Big\{e = (u, v) \in E : c_e\Big(\pi(u), \pi(v)\Big) = \text{accept}\Big\}\Big| = \delta|E|$.

A constraint graph ϕ is unsatisfiable if there is no assignment that 1-satisfies it. We also say that ϕ is ε-far from being satisfiable if there is no assignment $\pi : V \to \{0, 1\}$ that δ-satisfies ϕ, for any $\delta \geq 1 - \varepsilon$.

For abbreviation, we say that a constraint graph $\phi' = (G', C')$ is a subgraph of $\phi = (G, C)$ if G' is a subgraph of G, and in addition, for every $e \in E(G')$ the corresponding constraints $c_e \in C$ and $c'_e \in C'$ are identical.

The following main lemma is a natural generalization of the decomposition lemma of Leighton and Rao [1999], which is useful when analyzing graphs with general edge-constraints (rather than linear ones). The lemma states that any constraint graph which is far from being satisfiable has a small unsatisfiable subgraph (witness of unsatisfiability).

LEMMA 5.2. *Let $\phi = (G, C)$ be a constraint graph which is ε-far from being satisfiable. Then ϕ has an unsatisfiable subgraph ϕ' with at most $\frac{4 \log |E(G)|}{\varepsilon} + 2$ edges.*

Observe that an immediate corollary of Lemma 5.2 is that if a 2-CSP formula with m constraints is ε-far from being satisfiable (meaning that any assignment falsifies at least εm constraints) then it has an unsatisfiable subset of at most $\frac{4 \log m}{\varepsilon} + 2$ constraints.

Before proving the lemma we need some definitions.

Definition 5.3 *(Forcing)*. Let $\phi = (G, C)$ be a constraint graph, and let $u \in V(G)$ and $b_u \in \{0, 1\}$ be a vertex of G and a value assigned to it, respectively. For every vertex $v \in V(G) \setminus \{u\}$ and any value $b_v \in \{0, 1\}$, we say that $(u \leftarrow b_u)$ *forces* $(v \leftarrow b_v)$ if

—the partial assignment $\pi : \{u, v\} \to \{0, 1\}$ defined as $\pi(u) = b_u$ and $\pi(v) = b_v$ does not violate any constraint in C

—the partial assignment $\pi' : \{u, v\} \to \{0, 1\}$ defined as $\pi'(u) = b_u$ and $\pi'(v) = 1 - b_v$ violates at least one constraint $c_e \in C$ (and the violated constraints are called the *forcing constraints*).

We can naturally extend the notion of forcing for subsets of vertices as follows. Let $U \subset V(G)$ be a subset of G's vertices, and let $\pi_U : U \to \{0, 1\}$ be a partial assignment on U. For every vertex $v \in V(G) \setminus U$ and every value $b_v \in \{0, 1\}$ we say that π_U forces $(v \leftarrow b_v)$ if there exists a vertex $u \in U$ such that $(u \leftarrow \pi_U(u))$ forces $(v \leftarrow b_v)$.

In some cases there is no immediate forcing between assignments, but there is an indirect implication. We say that $(u \leftarrow b_u)$ *implies* $(v \leftarrow b_v)$ if there are $k > 0$ vertices $x_1, x_2, \ldots, x_k \in V \setminus \{u, v\}$ and k values $b_1, b_2, \ldots, b_k \in \{0, 1\}$ such that:

—$(u \leftarrow b_u)$ forces $(x_1 \leftarrow b_1)$

—for all $1 \leq i < k$, $(x_i \leftarrow b_i)$ forces $(x_{i+1} \leftarrow b_{i+1})$

—$(x_k \leftarrow b_k)$ forces $(v \leftarrow b_v)$.

We also define the *implication path* from $(u \leftarrow b_u)$ to $(v \leftarrow b_v)$ as the corresponding path of $k + 1$ forcing edges from u to v.

If for some pair of vertices $u, v \in V$ and a value $b_u \in \{0, 1\}$ the assignment $(u \leftarrow b_u)$ implies both $(v \leftarrow 0)$ and $(v \leftarrow 1)$, it means that $(u \leftarrow b_u)$ leads to contradiction, and hence any assignment π for which $\pi(u) = b_u$ cannot satisfy ϕ. In this case we call the pair of corresponding implication paths a *contradiction cycle*. Furthermore, if both $(u \leftarrow 0)$ and $(u \leftarrow 1)$ lead to contradiction, then clearly the constraint graph is unsatisfiable. In this case, we call the pair of corresponding contradiction cycles a *witness of unsatisfiability*.

Given a subset $U \subset V$, a partial assignment $\pi_U : U \to \{0, 1\}$ has no consistent extensions if one of the following holds.

—π_U forces two different values on some $v \in V \setminus U$.
—there exists an edge $e = (v_1, v_2) \in E(V \setminus U)$ such that π_U forces the values b_1, b_2 on v_1, v_2 respectively, and $c_e(b_1, b_2) = \mathsf{reject}$.

Notice that in both cases there is a contradiction cycle witnessing the inextensibility of π_U.

If π_U has consistent extensions, then we denote by $f(U) \triangleq \{v_1, \ldots, v_k\} \subseteq V \setminus U$ the set of all vertices that are forced by π_U to have the values $b_{v_1}, \ldots, b_{v_k}$ respectively, and we define the *forced extension* of π_U which is an assignment $\pi_{U\cup f(U)} : U \cup f(U) \to \{0, 1\}$ given by

$$\pi_{U\cup f(U)}(v) = \begin{cases} \pi_U(v) & , \quad v \in U \\ b_v & , \quad v \in f(U) \end{cases}.$$

PROOF OF LEMMA 5.2. Assume for the sake of contradiction that $\phi = (G, C)$ is the smallest constraint graph that violates the conditions of Lemma 5.2. Namely, ϕ is ε-far from being satisfiable, but it has no unsatisfiable subgraph with at most $\frac{4\log|E(G)|}{\varepsilon} + 2$ edges. Pick an arbitrary vertex $r \in V(G)$ and consider the executions **FindContradiction**$(r, 0)$ and **FindContradiction**$(r, 1)$ of the following algorithm, which is basically a BFS algorithm starting from vertex r that proceeds along forcing edges.

FindContradiction(r,b):

(1) Set $U = \{r\}$, $i = 0$, and define a partial assignment π_U as $\pi_U(r) = b$.
(2) $i = i + 1$.
(3) If $i > \frac{\log|E(G)|}{\varepsilon}$ output FAIL.
(4) If π_U has a consistent extension $\pi_{U\cup f(U)}$ to the set $f(U)$ of the forced neighbors of U:
  (a) If $|E(f(U), U)| \geq \varepsilon|E(U)|$ then set $U = U \cup f(U)$, set $\pi_U = \pi_{U\cup f(U)}$ and go to step 2.
  (b) E output FAIL.
(5) E there must be a contradiction cycle $\mathcal{W}$ of length[9] at most $2i + 1 \leq \frac{2\log|E(G)|}{\varepsilon} + 1$ for the assignment $(r \leftarrow b)$. Output $\mathcal{W}$.

[9]The bound on the cycle length is due to the fact that every implication in U has a corresponding implication path of length at most i that follows the iterative extension of π_U.

If both executions **FindContradiction**$(r, 0)$ and **FindContradiction**$(r, 1)$ reached step 5 then we have a pair of contradiction cycles (each of length at most $\frac{2\log|E(G)|}{\varepsilon} + 1$) for both $(r \leftarrow 0)$ and $(r \leftarrow 1)$. Joined together, these cycles form a witness of unsatisfiability of length at most $\frac{4\log|E(G)|}{\varepsilon} + 2$, contradicting our assumption that ϕ has no unsatisfiable subgraphs with at most $\frac{4\log|E(G)|}{\varepsilon} + 2$ edges. Therefore, one of the executions must output FAIL either in step 3 or in step 4b.

Since in every iteration of the algorithm $|E(U)|$ grows by a multiplicative factor of at least $(1+\varepsilon)$, after $\frac{\log|E(G)|}{\varepsilon} > \log_{(1+\varepsilon)}|E(G)|$ iterations we get $|E(U)| > |E(G)|$, which is of course impossible. This completely rules out the possibility of outputting FAIL in step 3.

Finally, assume towards a contradiction that one of the executions outputs FAIL in step 4b. Consider the induced subgraphs $G_U = G(U)$ and $G_{V\setminus U} = G(V \setminus U)$, and the corresponding induced constraint graphs $\phi_U = (G_U, C_U)$ and $\phi_{V\setminus U} = (G_{V\setminus U}, C_{V\setminus U})$ where C_U and $C_{V\setminus U}$ are the sets of all original constraints associated with $E(U)$ and $E(V \setminus U)$ respectively.

According to Algorithm **FindContradiction(r,b)**, the set U is enlarged only when the assignment π_U has a consistent extension. This fact preserves the invariant that the constraints $\{c_e : e \in E(U)\}$ are always satisfied by π_U. Therefore π_U completely satisfies the subgraph ϕ_U. On the other hand, by the minimality condition on ϕ, $\phi_{V\setminus U}$ must be $1-\varepsilon$ satisfiable by some assignment $\pi_{V\setminus U}$. Let $\pi : V(G) \to \{0, 1\}$ be the union of π_U and $\pi_{V\setminus U}$, defined as

$$\pi(v) = \begin{cases} \pi_U(v) & , \quad v \in U \\ \pi_{V\setminus U}(v) & , \quad v \in V \setminus U \end{cases}.$$

Since the execution was terminated at step 4b, π falsifies at most $\varepsilon|E(U)|$ of the constraints on $E(U, V \setminus U)$. So the total number of unsatisfied constraints by π is bounded by $\varepsilon|E(V \setminus U)| + \varepsilon|E(U, V \setminus U)| \le \varepsilon|E(G)|$, contradicting our initial assumption. □

5.2 The Uniform (Sparse) Verifier Lemma

In this section we claim that without loss of generality we can concentrate on (q, n, ℓ)-verifiers that make roughly $O(n + \ell)$ uniformly distributed queries. This assumption eases the application of Lemma 5.2, which bounds the size of contradiction witnesses as a function of number of edges (rather than number of vertices as in Lemma 4.5).

We note that a similar lemma was already proved by Goldreich and Sudan [2006] for $(q, n, 0)$-verifiers (property testers).

LEMMA 5.4. *For every $\gamma > 0$ and property $P \subset \Sigma^n$, if P has a (q, n, ℓ)-verifier $\mathcal{V} = \langle \mathcal{Q}, D\rangle$ with perfect completeness and soundness function $s : (0, 1] \to [0, 1]$ then P also has a (q, n, ℓ)-verifier $\mathcal{V}' = \langle \mathcal{Q}', U\rangle$ with the following properties.*

(1) *$\mathcal{V}'$ has perfect completeness.*
(2) *$\mathcal{V}'$ has soundness function s' that for all δ satisfies $s'(\delta) \ge s(\delta) - \gamma$.*
(3) *The number of queries in $\mathcal{Q}'$ is $\lceil \gamma^{-2}(n + \ell)\log|\Sigma|\rceil$.*
(4) *U is the uniform distribution over $\mathcal{Q}'$.*

PROOF. We prove the lemma by the following probabilistic argument. Construct a multiset $\mathcal{Q}'$ by choosing independently at random $\gamma^{-2}(n+\ell)\log|\Sigma|$ queries $Q \in \mathcal{Q}$ according to distribution D. Given $\mathcal{Q}'$, the new verifier $\mathcal{V}'$ operates similarly to $\mathcal{V}$, but instead of choosing queries from $\mathcal{Q}$ according to distribution D, it chooses them from $\mathcal{Q}'$ according to the uniform distribution.

Since the original verifier $\mathcal{V}$ had perfect completeness and since $\mathcal{Q}' \subseteq \mathcal{Q}$, $\mathcal{V}'$ has perfect completeness too. Conditions 3 and 4 of the lemma follow from the definition of $\mathcal{Q}'$ and $\mathcal{V}'$. We only need to show that the soundness function s' of $\mathcal{V}'$ satisfies $s'(\delta) \geq s(\delta) - \gamma$ for all $\delta > 0$. Clearly, this is satisfied for all δ for which $s(\delta) \leq \gamma$ because the rejection probability is always nonnegative. Therefore, to complete the proof it is enough to show that with positive probability the set $\mathcal{Q}'$ satisfies the following: For every word w such that $s\Big(\delta(w,P)\Big) > \gamma$ and every proof π, at least a $\Big(s\Big(\delta(w,P)\Big) - \gamma\Big)$-fraction of the queries in $\mathcal{Q}'$ reject the pair $w \circ \pi$ (we say that the query $Q = (I, C)$ *rejects* the pair $w \circ \pi$ if $C(w \circ \pi|_I) = \mathsf{reject}$).

Fix a word $w \in \Sigma^n$ such that $s\Big(\delta(w,P)\Big) > \gamma$ and a proof $\pi \in \Sigma^\ell$. For every $Q \in \mathcal{Q}$, we define the indicator variable $x_{Q,w\circ\pi}$ which is equal to 1 if Q rejects the pair $w \circ \pi$. Notice that once w is fixed, for any proof π we have $\mathbb{E}_{Q\sim_D \mathcal{Q}}[x_{Q,w\circ\pi}] \geq s\Big(\delta(w,P)\Big)$.

We also define an indicator variable $I_{w\circ\pi}$ which equals 1 if the fraction of queries in $\mathcal{Q}'$ that reject the pair $w \circ \pi$ is at least $s\Big(\delta(w,P)\Big) - \gamma$. Since the queries in $\mathcal{Q}'$ were chosen independently (according to distribution D), by Chernoff's bound for any w and any π we have

$$\Pr_{\mathcal{Q}'}[I_{w,\pi} = 0] = \Pr_{\mathcal{Q}'}\left[\left(\frac{1}{|\mathcal{Q}'|}\sum_{Q\in\mathcal{Q}'} x_{Q,w\circ\pi}\right) < s\Big(\delta(w,P)\Big) - \gamma\right]$$

$$\leq \exp(-2\gamma^2|\mathcal{Q}'|) = \exp(-2\gamma^2\gamma^{-2}(n+\ell)\log|\Sigma|) < |\Sigma|^{-n-\ell}$$

and if we apply the union bound over all word-proof pairs $w \circ \pi$ we get

$$\Pr_{\mathcal{Q}'}[I_{w,\pi} = 0 \text{ for some pair } w \circ \pi \text{ as above}] < |\Sigma|^{n+\ell} \cdot |\Sigma|^{-n-\ell} < 1.$$

We conclude that there must be a query set $\mathcal{Q}'$ that satisfies the required soundness condition. □

5.3 Best Soundness for Inspective Verifiers (Proof of Theorem 2.18)

THEOREM 2.18 (RESTATED BEST INSPECTIVE SOUNDNESS WITH SHORT PROOFS). *Let $P \subseteq \{0,1\}\varepsilon\{0,1\}^n$ be a d-universal property, and let $q \in \mathbb{Z}^+$. Let s_{i} denote the best soundness of a (q,n,ℓ)-inspective verifier for P, i.e., $s_{\mathrm{i}}(\delta) = \mathcal{S}^P_{\mathbf{V}_{\mathrm{i}}}(q,\ell,\delta)$. Then for every $\delta \in [0,1]$,*

$$s_{\mathrm{i}}(\delta) \leq \inf_{\varepsilon>0}\left\{\frac{4\log(\varepsilon^{-2}(n+\ell))}{\frac{d}{q-1} - 2} + \varepsilon\right\}.$$

Before proceeding to the proof, we need to define an inspective verifier, which is basically the graph that is induced by a verifier (recall that we had defined a special case of inspective graphs (Definition 4.3) in the warm-up section). These graphs play a crucial role in the proofs of Lemma 6.3 and Theorem 2.20.

Definition 5.5 *(Inspective Graph)*. Let $\mathcal{V} = \langle \mathcal{Q}, D\rangle$ be a (q, n, ℓ)-verifier. For $Q = (I, C)$ of i-size 2 we say Q *generates* the pair $I \cap [n+1, n+\ell]$. Similarly, if Q is of i-size 1 we say it *generates* the pair $(0, I \cap [n+1, n+\ell])$. A query of i-size different than 1, 2 generates no pair. The inspective graph of $\mathcal{V}$, denoted $G_{\mathcal{V}}$, is the multigraph with vertex set $V = \{0\} \cup [n+1, n+\ell]$ and edge set E being the multiset of pairs generated by $\mathcal{Q}$.

PROOF. Let $P \subset \{0, 1\}^n$ be a d-universal property, and let us fix $\varepsilon \in (0, 1)$ and $\delta \in (0, 1)$. Let $\mathbf{V_i}$ be an inspective (q, n, ℓ) verifier for P and let $\mathbf{V_i}' = \langle \mathcal{Q}', U\rangle$ be the corresponding "sparse" verifier (which is also inspective) described in Lemma 5.4 for $\gamma = \varepsilon$.

Fixing a δ-far word w defines a constraint graph $\phi_w = (G, C)$ over ℓ+1 vertices as follows.

—G is the inspective graph induced by $\mathbf{V_i}'$ as per Definition 5.5.
—For every $e = (u, v) \in E(G)$, the constraint c_e evaluates to accept whenever the valuation $\pi(u), \pi(v)$ and the word w satisfy the query in $\mathcal{Q}'$ (with i-size 2) that generates the edge e.
—For every $e = (0, v) \in E(G)$, the (unary) constraint c_e evaluates to accept whenever the valuation $\pi(v)$ and the word w satisfy the query in $\mathcal{Q}'$ (with i-size 1) that generates the edge e.

Notice that according to Lemma 5.4, the number of edges in $E(G)$ is bounded by $\varepsilon^{-2}(n + \ell)$. In addition, every constraint c_e depends on at most $q - 1$ word bits.

Since the minimal rejection probability of δ-far words by $\mathbf{V_i}'$ is $s_{\mathrm{i}}(\delta) - \varepsilon$, the constraint graph ϕ_w must be $(s_{\mathrm{i}}(\delta) - \varepsilon)$-far from being satisfiable. Hence by Lemma 5.2, ϕ_w has an unsatisfiable subgraph ϕ with at most

$$\frac{4 \log |E(G)|}{s_{\mathrm{i}}(\delta) - \varepsilon} + 2 \le \frac{4 \log \left(\varepsilon^{-2}(n + \ell)\right)}{s_{\mathrm{i}}(\delta) - \varepsilon} + 2$$

edges. Let $i_1, i_2, \ldots, i_k \in [n]$ be the word bits associated with the constraints (edges) of the unsatisfiable subgraph ϕ, where $k \le (q - 1) \cdot \left(\frac{4 \log(\varepsilon^{-2}(n+\ell))}{s_{\mathrm{i}}(\delta)-\varepsilon} + 2\right)$. It is clear that any word $w' \in \{0, 1\}^n$ that agrees with w on indices $i_1, i_2, \ldots, i_k$ cannot be in the property P. Therefore, because of the universality condition k must be larger than d, implying

$$(q - 1) \cdot \left(\frac{4 \log \left(\varepsilon^{-2}(n + \ell)\right)}{s_{\mathrm{i}}(\delta) - \varepsilon} + 2\right) > d$$

or equivalently

$$s_{\mathrm{i}}(\delta) < \frac{4\log\left(\varepsilon^{-2}(n+\ell)\right)}{\frac{d}{q-1}-2} + \varepsilon.$$

□

COROLLARY 5.6. *Let $\alpha \in (0,1)$ be a positive constant and let $\mathcal{P} \triangleq \{P_n \subseteq \{0,1\}^n : P_n$ is αn−universal$\}$ be a family of αn-universal properties. The properties in $\mathcal{P}$ have no subexponential* **inspective** *PCPP's achieving constant soundness. Namely, for every $\varepsilon' \in (0,1]$ there are $\beta > 0$ and $n_0 \in \mathbb{N}$ such that for any property $P_n \in \mathcal{P}$, $n > n_0$ the following is satisfied for all $\delta \in [0,1]$:*

$$\mathcal{S}^{P_n}_{\mathbf{V_i}}\left(3, 2^{\beta n}, \delta\right) \leq \varepsilon'.$$

PROOF. Fix an arbitrary $\varepsilon' > 0$, and set $\beta > 0$ and $n_0 \in \mathbb{N}$ such that all $n > n_0$ satisfy the inequality

$$2^{\beta n} < 2^{\frac{\varepsilon'}{8}(\frac{\alpha n}{2}-2)+2\log\varepsilon'-2} - n.$$

Since P_n is a αn-universal property, we can apply Theorem 2.18 (with $q = 3$ and $\varepsilon = \varepsilon'/2$) and get that for every $\delta \in [0,1]$:

$$\mathcal{S}^{P_n}_{\mathbf{V_i}}\left(3, 2^{\beta n}, \delta\right) \leq \frac{4\Big(\log(n+2^{\beta n}) - 2\log\varepsilon' + 2\Big)}{\frac{\alpha n}{2}-2} + \varepsilon'/2,$$

additionally, according to our choice of β and n_0 we also have:

$$\frac{4\Big(\log(n+2^{\beta n}) - 2\log\varepsilon' + 2\Big)}{\frac{\alpha n}{2}-2} \leq \varepsilon'/2,$$

completing the proof. □

5.4 Proof of Theorem 2.9

THEOREM 2.9 (RESTATED). *Let $\alpha \in (0,1)$ be a positive constant and let $\mathcal{P} \triangleq \{P_n \subseteq \mathbb{F}_2^n : n \in \mathbb{N}\}$ be a family of linear properties (codes) with dual distance at least αn and such that for some $\delta_0 \in (0,1)$ they are not trivially δ_0-testable. The properties in $\mathcal{P}$ have no 3-query sub-exponential PCPP's achieving soundness larger than $1/3$. Namely, for every $\varepsilon \in (0,1]$ there are $\beta > 0$ and $n_0 \in \mathbb{N}$ such that for any property $P_n \in \mathcal{P}$, $n > n_0$ the following is satisfied for all $\delta \in [0,\delta_0]$:*

$$\mathcal{S}^{P_n}\left(3, 2^{\beta n}, \delta\right) \leq \frac{1}{3} + \varepsilon.$$

Before proceeding to the proof of Theorem 2.9 we need the following lemma, which is proved in the next section.

LEMMA 5.7. *Let $\mathcal{V}$ be a $(3, n, \ell)$ verifier for a $\mathbb{F}_p$-linear property $P \subseteq \mathbb{F}_p^n$ with dual distance at least 4. Let μ be the probability that $\mathcal{V}$ makes an inspective query (i.e., one that makes at most two queries into the proof). Then, using $s^{\mathcal{V}}$ to denote the soundness function of $\mathcal{V}$, we have for any $\delta < 1/2$*

$$s^{\mathcal{V}}(\delta) \leq \min\left\{1 - \mu + \mathcal{S}^{P}_{\mathbf{V_i}}(3, \ell, \delta), \left(1 - \frac{1}{p}\right)\mu\right\}.$$

PROOF OF THEOREM 2.9. Fix any $\varepsilon \in (0, 1]$, and let $\beta > 0$ and n_0 be the parameters promised by Corollary 5.6, so that $\mathcal{S}^{P_n}_{\mathbf{V_i}}\left(3, 2^{\beta n}, \delta\right) < \varepsilon$ for every $n > n_0$.

Notice that the right hand side of the inequality in Lemma 5.7 ($p = 2$ in our case) is maximized when the two terms are equal, that is, when $\mu = \frac{2}{3}\Big(1 + \mathcal{S}^{P}_{\mathbf{V_i}}(3, \ell, \delta)\Big)$. Therefore, for $n > n_0$ and proofs of length $2^{\beta n}$,

$$s^{\mathcal{V}}(\delta) \leq \frac{1}{3}\left(1 + \mathcal{S}^{P_n}_{\mathbf{V_i}}\left(3, 2^{\beta n}, \delta\right)\right) < \frac{1}{3} + \varepsilon,$$

where the second inequality follows from Corollary 5.6. □

5.5 Proof of Lemma 5.7

PROOF. To see why $s^{\mathcal{V}}(\delta) \leq 1 - \mu + \mathcal{S}^{P}_{\mathbf{V_i}}(3, \ell, \delta)$ convert $\mathcal{V} = \langle \mathcal{Q}, D \rangle$ into an inspective verifier $\mathcal{V}'$ as follows. $\mathcal{V}'$ picks $Q \sim D$ in the same manner that $\mathcal{V}$ does. If Q is an inspective query, $\mathcal{V}'$ performs it. Otherwise, $\mathcal{V}'$ performs the trivial (inspective) query that always accepts (without reading any information). Since $\mathcal{V}'$ is inspective, we conclude $s^{\mathcal{V}'} \leq \mathcal{S}^{P}_{\mathbf{V_i}}(3, \ell, \delta)$, that is, there exists some input w that is δ-far from $\mathcal{C}$ and a proof π such that $(w \circ \pi)$ is rejected by $\mathcal{V}'$ with probability at most $\mathcal{S}^{P}_{\mathbf{V_i}}(3, \ell, \delta)$. Even if $\mathcal{V}$ rejects all noninspective queries on this particular pair, this can only increase the soundness by an additive factor $1 - \mu$, implying the first inequality.

To show that $s^{\mathcal{V}}(\delta) \leq (1 - \frac{1}{p})\mu$ we need the following two lemmas, which we prove in Sections 5.5.1 and 5.5.2.

LEMMA 5.8. *Let $\mathcal{C} \subset \mathbb{F}_p^n$ be a linear code. For any $x \in \mathbb{F}_p^n$ and any codeword $w \in \mathcal{C}$,*

$$\delta(x + w, \mathcal{C}) \geq \delta(x, \mathcal{C}).$$

LEMMA 5.9. *Let $\mathcal{C} \subset \mathbb{F}_p^n$, and let $I \subset [n]$ be a subset of indices such that there does not exist a non-zero dual codeword $u \in \mathcal{C}^{\perp} \setminus \{0\}$ such that $\mathrm{supp}(u) \subseteq I$. Then, for any $x \in \mathbb{F}_p^n$ and any $y \in \mathbb{F}_p^{|I|}$,*

$$\Pr_{w \sim_U \mathcal{C}}[(x + w)|_I = y] = p^{-|I|},$$

and in particular, for any $y \in \mathbb{F}_p^{|I|}$,

$$\Pr_{w \sim_U \mathcal{C}}[w|_I = y] = p^{-|I|}.$$

One example of a set I (as stated in the hypothesis of Lemma 5.9) is any set of at most d indices when the dual distance of $\mathcal{C}$ is $d + 1$.

The proof proceeds as follows. First we fix a δ-far word $x \in \mathbb{F}_p^n$, and pick $\hat{w} \in \mathcal{C}$ uniformly at random. Let π denote the legitimate proof for the codeword $\hat{w}$. Then, we pick another codeword $w' \in \mathcal{C}$ uniformly at random, and set $w \triangleq x + w'$. Recall that according to Lemma 5.8, w is δ-far from $\mathcal{C}$. We use the word-proof pair $(w \circ \pi)$ to fool the verifier $\mathcal{V} = \langle \mathcal{Q}, D \rangle$, that is, to make it reject with probability at most $(1 - \frac{1}{p})\mu$.

Let $\mathcal{Q}_0, \mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3$ be a partition of $\mathcal{Q}$, where $\mathcal{Q}_i$ contains all queries that read i bits from the proof. Since the verifier $\mathcal{V}$ has perfect completeness, all queries in $\mathcal{Q}_3$ must be satisfied because π is a legitimate proof and all queries in $\mathcal{Q}_0$ (tester queries) must be satisfied because the dual distance of $\mathcal{C}$ is larger than three. In addition, the queries in $\mathcal{Q}_2$ are satisfied with probability at least $1/p$, since according to Lemma 5.9 for every $i \in [n]$, $w_i = \hat{w}_i$ with probability $1/p$. To complete the proof, it is enough to show that every query $Q \in \mathcal{Q}_1$ is satisfied with probability at least $1/p$ over the choice of $\hat{w}$ and w'.

Let $Q = (I, C)$ be a query in $\mathcal{Q}_1$. Let i_1, i_2 be the indices in $I \cap [n]$ and let j be the index in $I \cap [n+1, n+\ell]$, so that the query Q is satisfied whenever $C(w_{i_1}, w_{i_2}, \pi_j) = \mathsf{accept}$. For every $\beta \in \mathbb{F}_p$, let k_β denote the number of assignments $(\alpha_1, \alpha_2) \in \mathbb{F}_p^2$ for which $C(\alpha_1, \alpha_2, \beta) = \mathsf{accept}$.

Recall that we chose π by the following distribution: pick a codeword $\hat{w} \in \mathcal{C}$ uniformly at random, and set π to be the legitimate proof for codeword $\hat{w}$. Let η_β denote the probability that when $\hat{w} \in \mathcal{C}$ is picked uniformly at random, in the legitimate proof π of $\hat{w}$ we have $\pi_j = \beta$. Perfect completeness and the assumption that the dual distance of $\mathcal{C}$ is larger than two imply that for any of the p^2 pairs $w_{i_1}, w_{i_2} \in \mathbb{F}_p$ there must be at leat one $\pi_j \in \mathbb{F}_p$ for which $C(w_{i_1}, w_{i_2}, \pi_j) = \mathsf{accept}$. Therefore, the average value of k_β is at least p. That is, $\sum_\beta \eta_\beta k_\beta \geq p$.

Now let us fix $x \in \mathbb{F}_p^n$, $\hat{w} \in \mathcal{C}$ and the corresponding $\pi_j \in \mathbb{F}_p$, and analyze what happens when we pick a random $w' \in \mathcal{C}$. Recall that $w = x + w'$, and hence by Lemma 5.9 the values w_{i_1} and w_{i_2} are distributed uniformly and independently of each other. Therefore, for any fixed $\pi_j \in \mathbb{F}_p$ we have

$$\Pr_{w'}[C(w_{i_1}, w_{i_2}, \pi_j) = \mathsf{accept}] = \frac{k_\beta}{p^2}.$$

So, the overall acceptance probability is

$$\Pr_{\hat{w}, w'}[C(w_{i_1}, w_{i_2}, \pi_j) = \mathsf{accept}] = \sum_\beta \eta_\beta \cdot \frac{k_\beta}{p^2} \geq 1/p$$

as required.

We constructed a distribution of word-proof pairs $(w \circ \pi)$ in which all words are δ-far from $\mathcal{C}$, and all proofs are legitimate proofs. Any query from $\mathcal{Q}_3$ is satisfied with probability 1 under this distribution, and all other queries are satisfied with probability at least $1/p$. So by linearity of expectation, we conclude that there must be a pair $(w \circ \pi)$ (where w is δ-far from $\mathcal{C}$) that is accepted by the verifier $\mathcal{V}$ with probability at least $(1-\mu) \cdot 1 + \mu \cdot \frac{1}{p} = 1 - (1 - \frac{1}{p})\mu$. □

5.5.1 *Proof of Lemma 5.8.*

PROOF. Assume towards a contradiction that for some $x \in \mathbb{F}_p^n$ and $w \in \mathcal{C}$ we have $\delta(x + w, \mathcal{C}) < \delta(x, \mathcal{C})$. Let $w' \in \mathcal{C}$ be the closest codeword to $x + w$, i.e. a codeword for which $\delta(x + w, w') = \delta(x + w, \mathcal{C})$. Observe that $\delta(x + w, w') = \delta(x, w' + (-w))$, and $w' + (-w) \in \mathcal{C}$. This, together with our initial assumption, leads to the following contradiction,

$$\delta(x, \mathcal{C}) > \delta(x + w, \mathcal{C}) = \delta(x + w, w') = \delta(x, w' + (-w)) \geq \delta(x, \mathcal{C}).$$

□

5.5.2 *Proof of Lemma 5.9.*

PROOF. We only need to prove the second part of the lemma since the first part follows from the second part, as a constant shift of the uniform distribution yields the uniform distribution. Now to the proof of the second part. Let H be the parity check matrix for the code $\mathcal{C}$, that is, $\mathcal{C} = \{w|Hw = 0\}$. Since there does not exist any nonzero dual codeword u such that $\text{supp}(u) \subseteq I$, we have that the system of equations $Hw = 0$ and $\{w_i = y_i | i \in I\}$ for any $y \in \mathbb{F}_p^{|I|}$ is consistent. Denote by $\mathcal{C}_{I=y}$ the solution set to this system of equations. It is easy to see that $\mathcal{C}_{I=y} = w_y + \mathcal{C}_{I=\bar{0}}$ where $w_y \in \mathcal{C}_{I=y}$. Thus, the sets $\mathcal{C}_{I=y}$, as y varies over $\mathbb{F}_p^{|I|}$ is an equipartition of the code $\mathcal{C}$. Thus, $\Pr_{w \sim_U \mathcal{C}}[w \in \mathcal{C}_{I=y}] = p^{-|I|}$ for all y. This proves the second part. □

6. PROOF OF LENGTH-SOUNDNESS TRADE-OFF FOR LINEAR VERIFIERS

We start by restating our main theorem regarding linear verifiers and its main corollary. In Section 6.1 we reduce both of these results to our main technical lemma, Lemma 6.3. To prove the lemma we need (a variant of) the decomposition lemma of Leighton and Rao [1999] and this is proved in Section 6.2. After setting the ground with the decomposition lemma, we complete our proof by proving the main lemma in Section 6.3.

THEOREM 2.11 (RESTATED). *Let $P \subseteq \mathbb{F}^n$ be a $\mathbb{F}$-linear property. Let $s[\ell](\delta)$ denote the best soundness of a $(3, n, \ell)$-linear verifier for P, i.e., $s[\ell](\delta) = \mathcal{S}^P_{\mathbf{linV}}(3, \ell, \delta)$. Let $t[q](\delta)$ denote the best soundness of a q-tester for P, i.e., $t[q](\delta) = \mathcal{S}^P(q, 0, \delta)$. Then*

$$s[\ell](\delta) \leq \inf_{\varepsilon > 0} \left\{ t\left[\frac{36 \log \ell}{\varepsilon}\right](\delta) + \frac{1}{2} \cdot \left(1 - \frac{1}{|\mathbb{F}|} + \varepsilon\right) \right\}.$$

COROLLARY 2.12 (RESTATED). *Let* **SUBEXP** *denote the set of subexponential functions, i.e., functions satisfying $f(n) = 2^{o(n)}$. For every prime field $\mathbb{F}_p$ there exists a family of $\mathbb{F}_p$-linear properties $\mathcal{P}$ such that*

$$\text{s-Def.}_{\mathbb{F}_p-\mathbf{linV}}[\mathcal{P}, \mathbf{SUBEXP}](\delta) \geq \delta - \frac{1}{2} \cdot \left(1 - \frac{1}{p}\right),$$

Consequently, the maximal deficiency of linear verifiers with subexponential proofs is at least $\frac{1}{2} \cdot (1 - 1/p)$:

$$\text{max-s-Def.}_{\mathbb{F}_p-\mathbf{linV}}[\mathbb{F}_p\textit{-linear}, \mathbf{SUBEXP}] \geq \frac{1}{2} \cdot \left(1 - \frac{1}{p}\right).$$

We start by proving that the main theorem implies the corollary.

PROOF OF COROLLARY 2.12. Take $\mathcal{P} = \{P_n \mid n \in \mathbb{Z}^+\}$ to be a family of linear properties satisfying both *(a)* $(\dim(P_n)/n)\xrightarrow[n\to\infty]{}0$ and *(b)* the best soundness of an $o(n)$-tester for P_n goes to 0 as n goes to ∞. One construction of such a family is based on properties that are not trivially testable and have vanishing rate (i.e., satisfy *(a)*) and linear dual distance, that is, the minimal weight of a nonzero element in $P_n^\perp$ is $\Omega(n)$. Any $o(n)$-tester with perfect completeness for such a property must have soundness function 0. A different construction is obtained by taking $\mathcal{P}$ to be a family of random Low Density Parity Check (LDPC) codes that satisfy *(a)*. These codes were shown by Ben-Sasson et al. [2005] to satisfy *(b)*. Let $w_n \in \mathbb{F}^n$ be δ-far from P_n. The verifier in Theorem 2.8 achieves soundness $\geq \delta$ on w when the proof-length is exponential in n. On the other hand, take ε_n to be a sequence approaching 0 when n approaches ∞ while satisfying $\frac{36 \log \ell(n)}{\varepsilon_n} = o(n)$. Such a sequence exists because $\ell(n) = 2^{o(n)}$. In this case Theorem 2.11 shows that the upper bound on soundness of $(3, n, \ell(n))$-verifiers approaches $\frac{1}{2} \cdot \left(1 - \frac{1}{p}\right)$ as n approaches ∞. This proves the first part of the corollary. To get the second part notice that *(a)* implies that a random $w' \in \mathbb{F}_p^n$ has distance $\delta = ((1 - 1/p) - o(1))$ from P_n[10]. This completes the proof. $\square$

6.1 Proof of Theorem 2.11

Overview. Given a verifier $\mathcal{V}$ and a word w that is δ-far from P we need to describe a proof π such that $\mathcal{V}$ accepts $w \circ \pi$ with relatively high probability. We divide this into two cases. If a large fraction of the queries of $\mathcal{V}$ are inspective, we try to satisfy these queries and care little about the rejection probability on the other queries. This part is argued in Lemma 6.3. On the other hand, if $\mathcal{V}$ rarely queries w, we present a proof that is good for some codeword $w' \in P$ and hope that $\mathcal{V}$ doesn't notice the difference between w and w'. Details follow.

Notation. When discussing $\mathbb{F}$-linear verifiers, we view a word-proof pair as a vector $w \circ \pi \in \mathbb{F}^{n+\ell}$ by setting $(w \circ \pi)_i = (w \circ \pi)[i]$. A q-query constraint $Q = (I, C)$ can be represented by a vector $v_Q \in \mathbb{F}^{n+\ell}$ such that the support of v_Q, denoted $\mathrm{supp}(v_Q)$, is I and

$$C(w \circ \pi|_I) = \mathsf{accept} \Leftrightarrow \langle v_Q, w \circ \pi \rangle = \sum_{i=1}^{n+\ell} (v_Q)_i (w \circ \pi)_i = 0.$$

Abusing notation, we identify Q with its representing vector and say "$(w \circ \pi)$ satisfies Q" whenever $\langle Q, (w \circ \pi) \rangle = 0$. For $I' \subset [n + \ell]$ we denote $\mathrm{supp}(Q) \cap I'$ by $\mathrm{supp}_{I'}(Q)$. Similarly, let $\langle Q, w \circ \pi \rangle_{I'} = \sum_{i \in I'} Q_i \cdot (w \circ \pi)_i$, where Q_i denotes the i^{th} entry of the vector Q. Finally, for P a linear space we denote its dual space by $P^\perp$.

To simplify the proof of Theorem 2.11 we assume our verifier makes no *redundant* queries according to the following definition and claim.

Definition 6.1. A query $Q \in \mathbb{F}^{n+\ell}$, $|\mathrm{supp}(Q)| \leq 3$ is called *redundant* for the property P if $|\mathrm{supp}_{[n]}(Q)| > 0$, $|\mathrm{supp}_{[n+1,n+\ell]}(Q)| > 0$ and there exists $u \in P^\perp$, $u \neq 0$ with $\mathrm{supp}(u) \subseteq \mathrm{supp}_{[n]}(Q)$.

[10] This follows from the fact that the expected distance of a random word w from a fixed codeword $c \in P_n$ is exactly $1 - 1/p$. Applying Chernoff, we obtain that $\Pr[\delta(w, c) \geq (1 - 1/p) - o(1)] = 1 - o(1)$. Applying union bound (over all the codewords in P_n) we obtain $\delta(w, P_n) = (1 - 1/p) - o(1)$ with high probability (since $(\dim(P_n)/n) \xrightarrow[n\to\infty]{} 0$).

If the dual distance of P is greater than 2 then all queries are nonredundant. The next claim says that even if the dual distance of P is 2, we may assume without loss of generality that its verifier makes no redundant queries. Its proof comes after the proof of Theorem 2.11.

CLAIM 6.2. *If P has a $(3, n, \ell)$-linear verifier with soundness function s, then P has a $(3, n, \ell)$-linear verifier that makes no redundant query and has soundness function s.*

PROOF OF THEOREM 2.11. Let $\mathcal{V} = \langle \mathcal{Q}, D\rangle$ be a 3-query linear verifier. Let $\mu = \Pr_{Q\sim_D \mathcal{Q}}[\mathrm{supp}_{[n]}(Q) \neq \emptyset]$. Fix $\varepsilon > 0$. We prove the following bound:

$$s[\ell](\delta) \leq \min \left\{ t\left[\frac{36\log \ell}{\varepsilon}\right](\delta) + \varepsilon + (1-\mu)\cdot\left(1 - \frac{1}{|\mathbb{F}|}\right), \right. \\ \left. t\left[\frac{36\log \ell}{\varepsilon}\right](\delta) + \mu\cdot\left(1 - \frac{1}{|\mathbb{F}|}\right)\right\}. \quad (1)$$

The right-hand side attains its maximal value when

$$\mu = \frac{1}{2} + \frac{\varepsilon}{2\left(1 - \frac{1}{|\mathbb{F}|}\right)}.$$

Plugging this value of μ back into (1) completes the proof of Theorem 2.11.

Now we argue Equation (1). The first element on the right hand side of Equation (1) is given by the following lemma that is proved in the next subsection.

LEMMA 6.3. *Let $\mathcal{V} = \langle \mathcal{Q}, D\rangle$ be a $\mathbb{F}$-linear verifier for the $\mathbb{F}$-linear property $P \subseteq \mathbb{F}^n$ with soundness function s, let $\varepsilon > 0$ and let $\mu = \Pr_{Q\sim_D \mathcal{Q}}[\mathrm{supp}_{[n]}(Q) \neq \emptyset]$. Then*

$$s(\delta) \leq t\left[\frac{36\log \ell}{\varepsilon}\right](\delta) + \varepsilon + (1-\mu)\cdot\left(1 - \frac{1}{|\mathbb{F}|}\right).$$

To complete the proof we only need to show

$$s[\ell](\delta) \leq t\left[\frac{36\log \ell}{\varepsilon}\right](\delta) + \mu\cdot\left(1 - \frac{1}{|\mathbb{F}|}\right). \quad (2)$$

Let w_0 be δ-far from P. By linearity, the all-zero proof $\pi_0 = \mathbf{0}$ is a legitimate proof (accompanying the zero codeword). Consider the soundness of $\mathcal{V}$ when presented with $w \circ \pi_0$ where w is the sum of w_0 and a random word $w' \in P$. Every query Q, $\mathrm{supp}_{[n]}(Q) = \emptyset$ is satisfied by the legitimate proof π_0. Additionally, every query Q, $\mathrm{supp}_{[n+1,n+\ell]}(Q) = \emptyset$ corresponds to a test, so the accumulated rejection probability of such tests is at most $t\left[\frac{36\log\ell}{\varepsilon}\right](\delta)$ because increasing query complexity does not decrease soundness. Finally, consider a query Q such that both $\mathrm{supp}_{[n]}(Q)$ and $\mathrm{supp}_{[n+1,n+\ell]}(Q)$ are not empty. By Claim 6.2 we may assume $\mathcal{V}$ is nonredundant, so there is no $u \in P^{\perp}, u \neq 0$ such that $\mathrm{supp}(u) \subseteq \mathrm{supp}_{[n]}(Q)$. Since P is linear, by Lemma 5.9 for a random $w' \in P$ we know that $\langle Q, w'\rangle_{[n]}$ is a random element of $\mathbb{F}$. This implies the rejection probability over such tests is at most $\mu\cdot(1 - 1/|\mathbb{F}|)$. This gives Equation (2), and Theorem 2.11 follows. □

PROOF OF CLAIM 6.2. Let $\mathcal{V}$ be $(3, n, \ell)$-linear verifier for P using redundant queries. We replace these queries, one at a time, without increasing query complexity and length and without decreasing soundness.

Let Q be redundant. Since $\left|\mathrm{supp}_{[n]}(Q)\right| \leq 2$ and there exists $u \in P^{\perp}$, $\mathrm{supp}(u) \subseteq \mathrm{supp}_{[n]}(Q)$ there exists a nonzero vector $Q' \in \mathrm{span}((P^{\perp}, 0^{l}), Q)$ such that $\left|\mathrm{supp}_{[n]}(Q')\right| < \left|\mathrm{supp}_{[n]}(Q)\right|$ and $\mathrm{supp}_{[n+1,n+\ell]}(Q') = \mathrm{supp}_{[n+1,n+\ell]}(Q)$ where by $(P^{\perp}, 0^{l})$ we refer to the space $\{(p, 0^{l}) | p \in P^{\perp}\}$, i.e., elements of $P^{\perp}$ appended by ℓ zeroes. Replace Q by Q'. It is easy to check that the completeness and the soundness are unaltered. Thus, each time there exists a redundant query, we can iteratively reduce its support in $[n]$ (i.e., $\left|\mathrm{supp}_{[n]}(Q')\right| < \left|\mathrm{supp}_{[n]}(Q)\right|$) till the size of the support in $[n]$ reduces to 0, in which case it is no longer a redundant query. □

We end this subsection with the formal proof of Theorem 2.19.

PROOF OF THEOREM 2.19. Follows from Lemma 6.3 by noticing that in the case of an inspective verifier we have $\mu = 1$. □

6.2 The Decomposition Lemma

In the proof of Lemma 6.3 and later on in the proof of Theorem 2.20 we use the decomposition lemma of Leighton and Rao [1999], stated next. The proof is included here because we use a stronger version than the one appearing in Leighton and Rao [1999] and Trevisan [2005]. Our version deals with multigraphs yet bounds the radius of the decomposed graph as a function of the number of vertices. The proof is along the lines of Leighton and Rao [1999]. We will use the same notation as in Section 4.1 (introduced after Claim 4.4).

LEMMA 4.5 (RESTATED DECOMPOSITION LEMMA [LEIGHTON AND RAO 1999]). *For every $\varepsilon \in (0, 1)$ and every multigraph $G = (V, E)$, there exists a subset of edges $E' \subseteq E$ of size at most $\varepsilon|E|$, such that every component of the graph $G_{\mathsf{Decomp.}} = (V, E \setminus E')$ has radius strictly less than $\log|V|/\varepsilon$. The graph $G_{\mathsf{Decomp.}}$ is said to be an ε-*decomposition *of G.*

PROOF. Assume for contradiction that for some $0 < \varepsilon < 1$, there exists a graph G which cannot be decomposed into components of radius less than $\log|V|/\varepsilon$ by removing at most ε-fraction of the edges. Let G be such a graph with the minimum number of vertices.

Let v be a vertex of maximum degree in V. Hence, $\deg(v) \geq 2|E|/|V|$. Now, consider the set of vertices V' defined by the following sequence of operations. In the following, $\Gamma(V')$ denotes the neighborhood of V' (i.e., $\Gamma(V') = \{u \in V | (u, v) \in E \text{ for some } v \in V'\}$).

(1) Set $V' \leftarrow \{v\} \cup \Gamma(v)$

(2) While $|E(V', V \setminus V')| > \varepsilon|E(V')|$ do
  Set $V' \leftarrow V' \cup \Gamma(V')$

(3) Output V'

Clearly, $|E(V', V \setminus V)| \leq \varepsilon|E(V')|$. Let t be the number of iterations of the while loop in the above procedure. Clearly, $t + 1$ upper bounds the radius of the induced subgraph $G(V')$ because $d(v, u) \leq t + 1$ for all $u \in G(V')$. Furthermore, each iteration of the while loop increases the number of edges in $G(V')$ by a multiplicative factor of at least $(1 + \varepsilon)$. Hence,

$$|E(V')| > (1+\varepsilon)^t \deg(v) \geq (1+\varepsilon)^{(rad(G(V'))-1)} \left(\frac{2|E|}{|V|}\right) \geq (1+\varepsilon)^{rad(G(V'))} \cdot \frac{|E|}{|V|}$$

where in the last inequality we have used the fact $2 > (1 + \varepsilon)$. However, since $E(V') \subseteq E$, we have that $rad(G(V')) < \log |V| / \log(1 + \varepsilon) < \log |V|/\varepsilon$. Here, we have used the fact that $\log_2(1 + \varepsilon) > \varepsilon$ for all $\varepsilon \in (0, 1)$.

Now, consider the induced subgraph $G' = G(V \setminus V')$. Since $|V \setminus V'| < |V|$, by the minimality condition we have that there exists a set of edges $E'' \subseteq E(G')$ of size at most $\varepsilon|E(G')|$, such that every component of the graph $G'_{\mathsf{Decomp.}} = (V \setminus V', E(G') \setminus E'')$ has radius strictly less than $\log |V \setminus V'|/\varepsilon$.

Let $E' = E(V', V \setminus V') \cup E''$. We first observe that $|E'| \leq \varepsilon|E(V')| + \varepsilon|E(G')| \leq \varepsilon|E|$. Furthermore, the components of the graph $G_{\mathsf{Decomp.}} = (V, E \setminus E')$ are $G(V')$ and the components of $G'_{\mathsf{Decomp.}}$. Hence, their radius is strictly less than $\log |V|/\varepsilon$. This contradicts the assumption that G is a counterexample to the lemma. Hence, proved. □

6.3 Proof of Lemma 6.3

Overview. Given verifier $\mathcal{V} = \langle \mathcal{Q}, D\rangle$ we construct a tester $\mathcal{V}' = \langle \mathcal{Q}', D\rangle$ with a one-to-one correspondence between the queries of $\mathcal{V}$ and those of $\mathcal{V}'$. The query complexity of $\mathcal{V}'$ is $O\left(\frac{\log \ell}{\varepsilon}\right)$. Additionally, we construct a set of proofs Π such that for every proof $\pi \in \Pi$, a $(1 - \varepsilon)$-fraction of inspective queries Q satisfy $\langle Q, w \circ \pi\rangle = \langle Q', w \circ \pi\rangle$, where Q' is the test of $\mathcal{V}'$ corresponding to Q. Finally, we show that if π is a random proof from Π then the expected acceptance probability of a noninspective query is $\geq 1/|\mathbb{F}|$. Summing up, the difference between the rejection probability of the tester $\mathcal{V}'$ and that of the verifier $\mathcal{V}$ is at most $\varepsilon + (1 - 1/|\mathbb{F}|)(1 - \mu)$ and this completes our proof. The construction of $\mathcal{V}'$ and Π uses *(i)* the $\mathbb{F}$-linearity of the constraints and *(ii)* the ε-decomposition of the inspective graph of $\mathcal{V}$ given in Lemma 4.5. We now focus on these two aspects.

Decomposed $\mathbb{F}$-Linear Verifiers. Let $\mathcal{V}$ be a $\mathbb{F}$-linear verifier and let $G = G(\mathcal{V})$ be its inspective graph from Definition 5.5. Recall that if $\left|\mathrm{supp}_{[n+1,n+\ell]}(Q)\right| = 1$ then Q generates an edge between 0 and a vertex $i \in [n + 1, n + \ell]$ whereas if $\left|\mathrm{supp}_{[n+1,n+\ell]}(Q)\right| = 2$ both vertices of the edge generated by Q lie in $[n + 1, n + \ell]$. (If $\left|\mathrm{supp}_{[n+1,n+\ell]}(Q)\right| \neq 1, 2$ then Q generates no edge.)

Let G' be an ε-decomposition of G as per Lemma 4.5 with E' being the set of removed edges, $\left|E'\right| \leq \varepsilon\,|E|$. Let $V_0, V_1, \ldots, V_m$ be the set of connected components of G', where V_0 is the component to which the vertex 0 belongs. Let $F_0, \ldots, F_m$ be spanning trees of $V_0, V_1, \ldots, V_m$ respectively, of radius at most $\frac{\log \ell}{\varepsilon}$ each. (The existence of these trees is guaranteed by Lemma 4.5.) Let $r_1, \ldots, r_m$ be arbitrary roots for $F_1, \ldots, F_m$ and set $r_0 = 0$ to be the root of F_0. To describe $\mathcal{V}'$ and Π we define two types of constraints that belong to $\mathrm{span}(\mathcal{Q})$. They are described next.

Vertex Constraints. For $i \in V_j \setminus \{r_j\}$ let $\mathcal{Q}(i)$ be the set of constraints that generate the edges along the unique path in F_j leading from r_j to i. Let $Q(i)$ be the unique nonzero vector in $\mathrm{span}(\mathcal{Q}(i))$ satisfying

$$(Q(i))_{i'} = \begin{cases} -1 & i' = i \\ 0 & i' \in [n+1, n+\ell] \setminus \{r_j, i\} \end{cases} \tag{3}$$

Such a constraint can be shown to exist by performing Gaussian elimination to remove the variables appearing in internal nodes along the path from r_j to i. Formally, we can obtain $Q(i)$ as follows: Let $i_0 = r_j, i_i \ldots, i_t = i$ be the internal nodes along the path form r_j to i and let $Q_1, Q_2, \ldots, Q_t$ be the queries that generate the edges $(i_0, i_1), (i_1, i_2), \ldots, (i_{t-1}, i_t)$. We can assume wlog that query Q_k is of the form $a_k \pi_{i_{k-1}} - \pi_{i_k} + b_k w_{j_k} = 0$ for some $a_k, b_k \in \mathbb{F}$ and $j_k \in [n]$. We can now perform Gaussian elimination to eliminate $\pi_{i_k}, k = 1, \ldots, t-1$ from the constraints $Q_1, \ldots, Q_t$ to obtain $\left(\prod_{k=1}^t a_k\right) \pi_{i_0} - \pi_{i_t} + \sum_k c_k w_{j_k} = 0$ for some $c_k \in \mathbb{F}$ or equivalently $\left(\prod_{k=1}^t a_k\right) \pi_{r_j} - \pi_i + \sum_{c_k} w_{j_k} = 0$. This is the constraint $Q(i)$. We call $Q(i)$ the *vertex constraint* corresponding to i and record for future reference its basic properties.

CLAIM 6.4 (BASIC PROPERTIES OF VERTEX CONSTRAINT). *For $i \in V_j \setminus \{r_j\}$ we have*

(a) $\{i\} \subseteq \mathrm{supp}_{[n+1,n+\ell]}(Q(i)) \subseteq \{i, r_j\}$,
(b) $|\mathrm{supp}_{[n]}(Q(i))| \leq \frac{4 \log \ell}{\varepsilon}$ *and*
(c) $r_j \in \mathrm{supp}_{[n+1,n+\ell]}(Q(i))$ *iff* $j \neq 0$.

PROOF. Part *(a)* follows by construction. Part *(b)* holds because a query Q that generates an edge has $\left|\mathrm{supp}_{[n]}(Q)\right| \leq 2$ and $Q(i)$ lies in the span of at most $\frac{2 \log \ell}{\varepsilon}$ constraints. Regarding part *(c)*, clearly $j = 0$ implies $r_j \notin \mathrm{supp}_{[n+1,n+\ell]}(Q(i))$ because 0 is not in the support of any query. For the other direction, if $j \neq 0$ notice every constraint has precisely two vertices in its support. Additionally, every internal vertex along the path from r_j to i, but for i and r_j, appears in the support of exactly two constraints. Thus, any $Q \in \mathrm{span}(\mathcal{Q}(i))$ satisfying Equation (3) must have r_j in its support. □

Edge Constraints. For $e = (i, i') \in V_j \times V_j$ an edge in G' generated by Q, let

$$\hat{Q}(e) = \begin{cases} Q + Q_i \cdot Q(i) & i' = r_j \\ Q + Q_{i'} \cdot Q(i') & i = r_j \\ Q + Q_i \cdot Q(i) + Q_{i'} \cdot Q(i') & i, i' \neq r_j \end{cases} \text{ and } Q(e) = \begin{cases} \hat{Q}(e) & (\hat{Q}(e))_{r_j} = 0 \\ \frac{-1}{\hat{Q}_{r_j}} \cdot \hat{Q}(e) & (\hat{Q}(e))_{r_j} \neq 0 \end{cases}.$$

In words, $Q(e)$ is the unique linear combination of Q and $Q(i), Q(i')$ (if one or both of the latter two are defined) that satisfies

$$Q(e)_{r_j} \in \{-1, 0\} \text{ and } Q(e)_{i''} = 0 \text{ for } i'' \in [n+1, n+\ell] \setminus \{r_j\}. \tag{4}$$

We call $Q(e)$ the *edge constraint* corresponding to e and record for future reference its basic properties.

CLAIM 6.5. *For $e = (i, i') \in V_j \times V_j$ we have (a)* $\mathrm{supp}_{[n+1,n+\ell]}(Q(e)) \subseteq \{r_j\}$, *(b)* $|\mathrm{supp}_{[n]}(Q)| \leq \frac{8 \log \ell}{\varepsilon}$ *and (c) if $j = 0$ then* $\mathrm{supp}_{[n+1,n+\ell]}(Q(e)) = \emptyset$.

PROOF. Let Q be the constraint that generates e and notice $\text{supp}_{[n+1,n+\ell]}(Q) = \{i, i'\}$. For part *(a)* assume both i and i' are not r_j. Recall from Claim 6.4 that $\text{supp}_{[n+1,n+\ell]}(Q(k)) \subseteq \{r_j, k\}$ and $Q(k)_k = -1$ for all $k \in V_j \setminus \{r_j\}$. This implies $\text{supp}_{[n+1,n+\ell]}(Q(e)) = \text{supp}_{[n+1,n+\ell]}(\hat{Q}(e)) = \text{supp}_{[n+1,n+\ell]}(Q + Q_i \cdot Q(i) + Q_{i'} \cdot Q(i')) \subseteq \{r_j\}$. The case when one of i, i' is equal to r_j is handled similarly and this proves part *(a)*. Part *(b)* follows because $Q(e)$ lies in the span of at most $\frac{4 \log \ell}{\varepsilon}$ constraints and each constraint has $\left|\text{supp}_{[n]}(Q)\right| \le 2$. Part *(c)* follows from part *(a)* by observing that 0 is not in the support of any constraint. □

Forced Components. The construction of the tester $\mathcal{V}'$ and the corresponding proofs Π depend on a partition of the components of G' into *forced* and *unforced* components, defined next.

Definition 6.6 *(Forced Component)*. If $e \in V_j \times V_j$ satisfies $\text{supp}_{[n+1,n+\ell]}(Q(e)) = \{r_j\}$ we say *e forces* V_j. If V_j contains an edge that forces it we say V_j is *forced*. Pick an arbitrary ordering of edges and set the *designated* forcing edge of V_j to be the smallest edge that forces it. If a component V_j is not forced, it is said to be *unforced*.

Construction of the Tester $\mathcal{V}'$. We construct $\mathcal{V}' = \langle \mathcal{Q}', D\rangle$ from $\mathcal{V} = \langle \mathcal{Q}, D\rangle$ in three consecutive steps. Assume without loss of generality that $V_1, \ldots, V_k$ are the forced components of G' for some $k \le m$ and let $e_1, \ldots, e_k$ be the corresponding designated forcing edges. (Notice that Claim 6.5*(c)* implies that V_0 is unforced.) First we convert each query Q into a query $Q^{(1)}$ with $\text{supp}_{[n+1,n+\ell]}(Q^{(1)}) \subseteq \{r_1, \ldots, r_m\}$. Then we convert $Q^{(1)}$ into a $Q^{(2)}$ with $\text{supp}_{[n+1,n+\ell]}(Q^{(2)}) \subseteq \{r_{k+1}, \ldots, r_m\}$. Finally, we replace $Q^{(2)}$ by Q' with $\text{supp}_{[n+1,n+\ell]}(Q') = \emptyset$, i.e., Q' is a test. All the time we keep the same distribution over tests, that is, $D(Q') = D(Q^{(2)}) = D(Q^{(1)}) = D(Q)$. The detailed construction follows.

(1) For every query Q set

$$Q^{(1)} = Q + \sum_{i \in [n+1,n+\ell] \setminus \{r_1,\ldots,r_m\}} Q_i \cdot Q(i).$$

(2) For every query $Q^{(1)}$ set

$$Q^{(2)} = Q^{(1)} + \sum_{j=1}^{k} (Q^{(1)})_{r_j} \cdot Q(e_j).$$

(3) For every query $Q^{(2)}$ set

$$Q' = \begin{cases} 0 & |\text{supp}_{[n+1,n+\ell]}(Q^{(2)})| > 0 \\ Q^{(2)} & \text{otherwise} \end{cases}$$

Next we bound all of the important parameters of $\mathcal{V}'$ except for its soundness function.

CLAIM 6.7 (BASIC PROPERTIES OF $\mathcal{V}'$). *$\mathcal{V}'$ is a tester with perfect completeness and query complexity* $\le \frac{36 \log \ell}{\varepsilon}$.

PROOF. $\mathcal{V}'$ is a tester because the last conversion step enforces $\text{supp}(Q') \subseteq [n]$ for all $Q' \in \mathcal{Q}'$. Perfect completeness of $\mathcal{V}'$ follows from the perfect completeness of $\mathcal{V}$ by $\mathbb{F}$-linearity because $\mathcal{Q}' \subseteq \text{span}(\mathcal{Q})$.

Finally, the bound on query complexity follows from Claims 6.4*(b)*, 6.5*(b)* by noting that Q' lies in the span of Q and at most 3 vertex constraints and 3 edge constraints. Indeed,

$$Q^{(1)} \in \text{span}\left(Q, \left\{Q(i) \mid i \in \text{supp}_{[n+1,n+\ell]}(Q) \setminus \{r_1, \dots, r_m\}\right\}\right),$$

and since $\left|\text{supp}_{[n+1,n+\ell]}(Q)\right| \leq 3$ we conclude $Q^{(1)}$ is in the span of Q and at most 3 vertex constraints. By Claim 6.4*(a)* and Equation (3) we have

$$\text{supp}_{[n+1,n+\ell]}\left(Q^{(1)}\right) \subseteq \left\{r_j \mid \exists i \in \text{supp}_{[n+1,n+\ell]}(Q) \cap V_j\right\}, \tag{5}$$

so $\left|\text{supp}_{[n+1,n+\ell]}(Q)\right| \leq 3$ also implies $|\text{supp}_{[n+1,n+\ell]}(Q^{(1)})| \leq 3$. This implies $Q^{(2)}$ lies in the span of $Q^{(1)}$ and at most 3 edge constraints and our proof is complete. □

Construction of Proof-Set Π. To argue soundness of $\mathcal{V}'$ we introduce a family of proofs designed to fool inspective verifiers. Recall that $F_0, \dots, F_m$ are spanning trees of the components $V_0, \dots, V_m$. Let $F = \cup_j F_j$ be the spanning forest formed by the union of these spanning trees.

Definition 6.8. Let $V_1, \dots, V_k$ be the forced components of G' and let $e_1, \dots, e_k$ be their respective designated forcing edges. A proof π is called *F-compliant for w* if $w \circ \pi$ satisfies every constraint that generates an edge in $F \cup \{e_1, \dots, e_k\}\varepsilon\{e_1, \dots, e_k\}$. Let $\Pi = \Pi(w)$ denote the set of F-compliant proofs for w.

The next claim shows that F-compliant proofs exist for any word and describes the structure of these proofs. This structure will be used to analyze the soundness of $\mathcal{V}'$.

CLAIM 6.9. *For every $w \in \mathbb{F}^n$ and $\alpha_{k+1}, \dots, \alpha_m \in \mathbb{F}$ there exists a unique F-compliant proof for w such that $\pi_{r_j} = \alpha_j$ for $k < j \leq m$.*

PROOF. We first observe that the set of constraints that generate the edges of F, denoted $\mathcal{Q}(F)$, are linearly independent.

We will first show that for any $\alpha_1, \alpha_2, \dots, \alpha_m \in \mathbb{F}$, there exists a unique proof π such that $\pi_{r_j} = \alpha_j$ for $j = 1, \dots, m$ and π satisfies all the constraints in $\mathcal{Q}(F)$. We need to consider component V_0 (whose root is r_0) separately. Let $e = (0, i) \in F_0$ be generated by Q. There is a unique setting of π_i that satisfies Q because $\left|\text{supp}_{[n+1,n+\ell]}(Q)\right| = 1$. Once all vertices at distance 1 from r_0 have been fixed, there is a unique assignment to π_j, $j \in V_0$ that satisfies $\mathcal{Q}(F_0)$ — the set of constraints that generate edges in F_0. Now, set the values of the roots of the remaining trees $\pi_{r_1}, \dots, \pi_{r_m}$ to $\alpha_1, \dots, \alpha_m$ respectively. By linear independence of the constraints in each of F_i, we have that the above partial proof setting can be uniquely extended to a proof that satisfies all the constraints in $\mathcal{Q}(F)$ (This can be proved by induction on the length of the paths in F).

But we need to satisfy not only the constraints in $\mathcal{Q}(F)$, but also the constraints that generate the forcing edges $e_1, \ldots, e_k$. We will show that for each $1 \leq j \leq k$, the constraint that generates the forcing edge e_j, forces a particular value α_j for the corresponding root r_j (hence, the name forcing edge).

Let $1 \leq j \leq k$. Consider $e_j = (i, i')$ — generated by Q — that is the designated forcing edge of V_j. By Definition 6.6 and Equation (4), we have $\mathrm{supp}_{[n+1,n+\ell]}(Q(e)) = \{r_j\}$, so there is a unique setting for π_{r_j} that satisfies Q and $\mathcal{Q}(F_j)$. Let α_j be this unique value.

Given any $\alpha_{k+1}, \ldots, \alpha_m$, set the value of the roots $\pi_{r_j} = \alpha_j$ for $j = 1$ to m where α_j, $j = 1, \ldots, k$ are as defined above and α_j, $j = k+1, \ldots, m$ are the given values. From before, we have that there exists a unique proof π such that $\pi_{r_j} = \alpha_j$ for $j = 1, \ldots, m$ and π satisfies all the constraints in $\mathcal{Q}(F)$. But this proof also satisfies the constraints that generate the forcing edges $e_1, \ldots, e_k$ due to way we chose the α_j's for $j = 1, \ldots, k$. This proves the claim. □

F-compliant proofs are important because on "typical" queries the output of Q on $w \circ \pi$ is equal to the output of the test Q' performed on w. This is argued in our next claim.

CLAIM 6.10. *If π is F-compliant for w and $Q \in \mathcal{Q}$ has one of the following properties*:

(*1*) $\mathrm{supp}_{[n+1,n+\ell]}(Q) = \emptyset$, *or*
(*2*) *every* $i \in \mathrm{supp}_{[n+1,n+\ell]}(Q)$ *belongs to a forced component, or*
(*3*) Q *generates an edge* $e \in E \setminus E'$.

Then

$$\langle Q', w \circ \pi \rangle = \langle Q, w \circ \pi \rangle.$$

PROOF. We prove each case separately.

(1) By construction $Q' = Q^{(2)} = Q^{(1)} = Q$ and the claim follows.
(2) By assumption and Claim 6.4*(a)* and (5), we have $\mathrm{supp}_{[n+1,n+\ell]}(Q^{(1)}) \subseteq \{r_1, \ldots, r_k\}$. Suppose $r_j \in \mathrm{supp}_{[n+1,n+\ell]}(Q^{(1)})$. Definition 6.6 and Equation (4) imply $(Q(e_j))_{r_j} = -1$, so by construction $r_j \notin \mathrm{supp}_{[n+1,n+\ell]}(Q^{(2)})$. This is argued for each $r_j \in \mathrm{supp}_{[n+1,n+\ell]}(Q^{(1)})$ and shows $\mathrm{supp}_{[n+1,n+\ell]}(Q^{(2)}) = \emptyset$. By construction this implies $Q' = Q^{(2)}$. Notice $Q^{(2)} = Q + Q''$ where Q'' is a linear combination of constraints that generate edges in $F \cup \{e_1, \ldots, e_k\}$. We conclude

$$\langle Q', w \circ \pi \rangle = \langle Q^{(2)}, w \circ \pi \rangle = \langle Q, w \circ \pi \rangle + \langle Q'', w \circ \pi \rangle = \langle Q, w \circ \pi \rangle, \qquad (6)$$

The last equality follows because π is F compliant for w.

(3) We may assume e belongs to component V_j that is not forced because the other case (of forced V_j) was argued in part 2. By construction $Q^{(1)} = \hat{Q}(e)$. By assumption e does not force V_j, so by Definition 6.6 we have $\mathrm{supp}_{[n+1,n+\ell]}(Q(e)) = \emptyset$. By construction $Q' = Q^{(2)} = Q^{(1)}$ and the F-compliancy of π implies as argued in Equation (6) that $\langle Q', w \circ \pi \rangle = \langle Q^{(2)}, w \circ \pi \rangle = \langle Q, w \circ \pi \rangle$. This completes the proof.

□

We are ready to argue the soundness of $\mathcal{V}'$ and complete the proof of Lemma 6.3.

CLAIM 6.11 (SOUNDNESS). *Let* $\sigma = \Pr_{Q \sim_D \mathcal{Q}}[|\mathrm{supp}_{[n+1,n+\ell]}(Q)|\varepsilon|\mathrm{supp}_{[n+1,n+\ell]}(Q)| \leq 3]$. *There exists an F-compliant proof π such that*

$$\Pr[\mathcal{V}'^{w\circ\pi} = \textsf{reject}] \geq \Pr[\mathcal{V}^{w\circ\pi} = \textsf{reject}] - \varepsilon - (1 - 1/|\mathbb{F}|) \cdot \sigma.$$

PROOF. If π is F-compliant for w then by Claim 6.10 the output of $\mathcal{V}$ and $\mathcal{V}'$ on $w \circ \pi$ may differ only if the query performed is one of two types. The first type is a query that generates an edge $e \in E'$. The fraction of these queries is at most ε. The second type is a query with $\left|\mathrm{supp}_{[n+1,n+\ell]}(Q)\right| = 3$ and there exists $i \in \mathrm{supp}_{[n+1,n+\ell]}(Q)$ such that i belongs to an unforced component V_j. Let σ' denote the fraction of queries of the second type and notice $\sigma' \leq \sigma$. We can already conclude

$$\Pr\left[\mathcal{V}'^{w\circ\pi} = \textsf{reject}\right] \geq \Pr\left[\mathcal{V}^{w\circ\pi} = \textsf{reject}\right] - \varepsilon - \sigma,$$

but to reach the stronger claim stated above we need one additional observation regarding constraints of the second type.

Let Q be such a constraint and suppose $i \in \mathrm{supp}_{[n+1,n+\ell]}(Q)$ belongs to the unforced component V_j. Consider the uniform distribution over F-compliant proofs obtained by randomly fixing values $\alpha_{k+1}, \ldots, \alpha_m$ for $\pi_{r_{k+1}}, \ldots, \pi_{r_m}$ and extending these values to an F-compliant proof for w by Claim 6.9. Notice the value assigned to π_i depends linearly on the value of π_{r_j}. Thus, assigning a uniformly random value to π_{r_j} implies $\langle Q, w \circ \pi \rangle$ is a random variable ranging uniformly over $\mathbb{F}$, that is, Q accepts $w \circ \pi$ with probability $1/|\mathbb{F}|$. This implies the expected fraction of constraints of the second type that are satisfied is $1/|\mathbb{F}|$. We conclude the existence of an F-compliant proof that is rejected by at most a $(1 - 1/|\mathbb{F}|)$-fraction of the queries of the second type. This completes our proof. □

PROOF OF LEMMA 6.3. Let w be δ-far from P. Let $\mathcal{V}'$ be the tester constructed from $\mathcal{V}$ as described earlier in this subsection. Let π be the F-compliant proof for w satisfying Claim 6.11. Notice $\sigma \leq 1 - \mu$ so this claim implies

$$s(\delta) \leq \Pr[\mathcal{V}^{w\circ\pi} = \textsf{reject}] \leq \Pr[\mathcal{V}'^{w\circ\pi} = \textsf{reject}] + \varepsilon + (1 - 1/|\mathbb{F}|)(1 - \mu).$$

The proof is completed by recalling from Claim 6.7 that $\mathcal{V}'$ is a $\left(\frac{36 \log \ell}{\varepsilon}\right)$-tester, hence $\Pr[\mathcal{V}'^{w\circ\pi} = \textsf{reject}] \leq t\left[\frac{36 \log \ell}{\varepsilon}\right](\delta)$. □

7. PROOF OF LENGTH-SOUNDNESS TRADE-OFF FOR UNIQUE VERIFIERS

In this section, we prove the length-soundness trade-off for 3-query unique verifiers (Theorem 2.13). As in the case of linear verifiers, we first prove a similar theorem for the special case of inspective unique verifiers (Theorem 2.20) and then extend this result to general 3-query unique verifiers.

7.1 Best Soundness for Inspective Unique Verifiers (Proof of Theorem 2.20)

THEOREM 2.20 (RESTATED BEST SOUNDNESS WITH UNIQUE INSPECTIVE VERIFIERS). *Let $P \subseteq \Sigma^n$ be a property. Let $s_{\mathrm{i}}(\delta)$ denote the best soundness of a $(3, n, \ell)$-unique inspective verifier for P, i.e., $s_{\mathrm{i}}(\delta) = \mathcal{S}^P_{\mathbf{uniqV_i}}(3, \ell, \delta)$. Let $t[q](\delta)$ denote the best soundness of a q-tester for P, i.e., $t[q](\delta) = \mathcal{S}^P(q, 0, \delta)$. Then for any $s_{\mathrm{i}}(\delta) > \varepsilon$*

$$s_{\mathrm{i}}(\delta) \leq \inf_{\varepsilon>0} \left\{ 4t \left[\frac{10 \log \ell}{\left(s_{\mathrm{i}}(\delta) - \varepsilon\right) \varepsilon} \cdot \ln\left(2|\Sigma|\right) \right](\delta) + \varepsilon \right\}.$$

The conclusion of Theorem 2.20 has $s(\delta)$ on both sides of the inequality, which makes it rather cumbersome to deal with. So, we obtain the following corollary of Theorem 2.20, which is a more convenient form to work with (for instance to derive Theorem 2.13).

COROLLARY 7.1. *Let $\alpha \in (0, 1)$ and let $P \triangleq \{P_n \subseteq \mathbb{F}_n : n \in \mathbb{N}\}$ be a family of $\mathbb{F}$-linear properties (codes) with dual distance at least αn and such that for some $\delta_0 \in (0, 1)$ they are not trivially δ_0-testable. For every $\varepsilon > 0$, there exists a $\beta > 0$ and $n_0 \in \mathbb{N}$, such that for any property $P_n, n > n_0$, the following is satisfied for all $\delta \in (0, \delta_0]$,*

$$\mathcal{S}^{P_n}_{\mathbf{uniqV_i}}\left(3, 2^{\beta n}, \delta\right) \leq 2\varepsilon.$$

PROOF. Set $\beta = \alpha\varepsilon^2/(10 \ln(2|\mathbb{F}|))$. Suppose the corollary is fa for this setting of β, i.e., there exists an inspective unique $(q, n, 2^{\beta n})$ verifier with soundness $s(\delta) > 2\varepsilon$. Now, since $s(\delta) > 2\varepsilon$, we have that $\frac{10\beta n}{(s(\delta)-\varepsilon)\varepsilon} \cdot \ln(2|\mathbb{F}|) < \frac{10\beta n}{\varepsilon^2} \cdot \ln(2|\mathbb{F}|) = \alpha n$. Define $l = 2^{\beta n}$. Since the dual distance of P_n is at least αn, we have $t\left[\frac{10 \log l}{(s(\delta)-\varepsilon)\varepsilon} \cdot \ln(2|\mathbb{F}|)\right](\delta) = 0$. Thus, it follows from Theorem 2.20 that $s(\delta) \leq \varepsilon$ contradicting our assumption that $s(\delta) > 2\varepsilon$. Hence, proved. □

PROOF OF THEOREM 2.20. The outline of the proof is similar to the linear case. Given an inspective unique verifier for some property P, we construct using the graph decomposition lemma (Lemma 4.5) a tester for P. The lower bound on the soundness of the tester implies a lower bound on that of the inspective verifier.

Let $P \subset \Sigma^n$ and let $\mathcal{V} = \langle \mathcal{Q}, D \rangle$ be an inspective unique (q, n, ℓ) verifier for P and let s denote the soundness of the verifier $\mathcal{V}$. We may assume without loss of generality that D is the uniform distribution by repeating queries in $\mathcal{Q}$ proportional to their probability. Let $G = G(\mathcal{V})$ be the inspective graph corresponding to $\mathbf{uniqV_i}$, as per Definition 5.5. For any ε, let G_ε be an ε-decomposition of G

as per Lemma 4.5. Note that the soundness of the verifier corresponding to G_ε is at least $s' = s - \varepsilon$. Let $V_0, V_1, \ldots, V_m$ be the components of G_ε, where V_0 is the component which contains the vertex 0. Let $F_1, F_2, \ldots, F_m$ be spanning trees of the components $V_0, V_1, \ldots, V_m$ respectively, of radius at most $\log \ell/\varepsilon$. Let $r_1, r_2, \ldots, r_m$ be arbitrary roots for $F_0, F_1, \ldots, F_m$ respectively and set 0 to be the root r_0 of F_0. Furthermore, let $p_0, p_1, \ldots, p_m$ be the normalized number of edges in components $V_0, V_1, \ldots, V_m$ respectively (i.e., $p_i = |E(V_i)|/|E(G_\varepsilon)|$).

Corresponding to every nontree edge $e = (u, v)$ in $E(V_i) \setminus F_i$, there exists a unique cycle in the graph $F_i + \{e\}$. Call this cycle c_e, the cycle completed by edge e.

For $i = 1, \ldots, m$ and any $\sigma \in \Sigma$ let $\pi_i^\sigma : V_i \to \Sigma$ be the unique labeling of the vertices of component V_i such that (a) the root r_i is labeled by σ and (b) all the edge constraints of the tree edges of F_i are satisfied by π_i^σ. Note that once the label of the root is fixed, it induces a labeling on all the vertices of the tree such that all tree-edge constraints are satisfied due to the uniqueness property of the verifier. π_i^σ is this induced labeling where the root vertex is labeled by σ. For the component V_0, note that there is a unique labeling of the vertices of V_0 that satisfies all tree-edge constraints. Let $\pi_0 : V_0 \to \Sigma$ be this unique labeling.

We are now ready to describe the tester $\mathcal{T}$ that distinguishes $w \in P$ from w that are δ-far from P. Recall that the soundness of the inspective verifier corresponding to G_ε is at least $s(\delta) - \varepsilon$. We call this quantity s'.

Tester $\mathcal{T}$
Oracle: $w : [n] \to \Sigma$

(1) Choose $i \leftarrow_R \{0, \ldots, m\}$ according to the probability distribution $(p_0, \ldots, p_m)$.
(2) Choose $k = \frac{2}{s'} \ln(2|\Sigma|)$ edges in $E(V_i) \setminus F_i$ (i.e., the nontree edges) uniformly at random (independently and with repetition).
(3) Let $\mathcal{C}$ be the set of all cycles completed by the above k nontree edges. Let $E_{\mathcal{C}}$ be the set of all edges contained in the cycles $\mathcal{C}$ (i.e., $E_{\mathcal{C}} = \{e | \exists c \in \mathcal{C}, e \in c\}$).
(4) Let $\mathcal{Q}_{\mathcal{C}}$ be the set of constraints of $\mathcal{V}$ that generate the set of edges $E_{\mathcal{C}}$. Let $I_{\mathcal{C}}$ be the set of indices in $[n]$ probed by the constraints $\mathcal{Q}_{\mathcal{C}}$ (i.e., $I_{\mathcal{C}} = \left(\bigcup_{(I,C)\in\mathcal{Q}_{\mathcal{C}}} I\right) \cap [n]$).
(5) Query the word w for all indices $j \in I_{\mathcal{C}}$
(6) If $i = 0$
    Accept if the partial assignments $w : I_{\mathcal{C}} \to \Sigma$ and $\pi_0 : V_0 \to \Sigma$ do not violate any constraint in $\mathcal{Q}_{\mathcal{C}}$
(7) E (i.e., $i \neq 0$)
    Accept if there exists a $\sigma \in \Sigma$ such that the partial assignments $w : I_{\mathcal{C}} \to \Sigma$ and $\pi_i^\sigma : V_i \to \Sigma$ do not violate any constraint in $\mathcal{Q}_{\mathcal{C}}$

The query complexity of the tester $\mathcal{T}$ is at most twice the number of edges E because each edge is labeled by at most 2 indices in $[n]$, so this query complexity is bounded above by $2k \cdot (2 \log \ell/\varepsilon + 1) \leq (10 \log \ell/s'\varepsilon) \cdot \ln(2|\Sigma|)$.

Clearly, this tester has perfect completeness. Consider any word $w : [n] \to \Sigma$ that is δ-far from P. We show below that the tester $\mathcal{T}$ rejects w with probability at least $(s(\delta) - \varepsilon)/4 = s'/4$. Given this fact, the theorem follows since $t\left[\frac{10\log\ell}{s'\varepsilon} \cdot \ln(2|\Sigma|)\right](\delta)$ upper bounds the rejection probability of any tester.

Since w is δ-far from P, it follows from the soundness of the inspective graph G_ε, that for any labeling $\pi : V(G_\varepsilon) \to \Sigma$, at least $s' = s(\delta) - \varepsilon$ fraction of the edge constraints are violated.

Suppose V_i is the component chosen in step 1. Consider the inspective graphs $G(V_i)$ corresponding to the components V_i. Let s_i be the soundness of $G(V_i)$. Also assume $s_i \geq s'/2$. We will later show that it sufficient if we restrict our attention to those components that satisfy $s_i \geq s'/2$.

Assume $i \neq 0$. Consider any $\sigma \in \Sigma$. Since the soundness of $G(V_i)$ is s_i, the labeling π_i^σ violates at least s_i fraction of edge constraints. (note that only nontree edges are violated by π_i^σ). Hence, for a random nontree edge, the probability that it is not violated by π_i^σ is at most $1 - s_i$. Therefore, the probability that all k edges chosen in step 2 are not violated by π_i^σ is at most $(1-s_i)^k \leq e^{-s_i k}$. Hence, the probability that there exists a $\sigma \in \Sigma$ such that all k edges are not violated by π_i^σ is at most $|\Sigma|e^{-s_i k} \leq 2^{-2s_i/s'}$ since $s_i \geq s'/2$.

If $i = 0$, the analysis is similar to above except that we do not have the final union bound. Hence, the probability that all k edges are not violated by π_0 is at most $e^{-s_0 k} \leq 2^{-2s_0/s'}/|\Sigma| < 2^{-2s_0/s'}$ since $s_0 \geq s'/2$.

We now need to relate s_i to s'. Towards this end, observe that $\sum p_i s_i$ denotes the soundness of the entire graph which is at least $s' = s - \varepsilon$. Hence, with probability at least $s'/2$, the component i chosen in step 1 satisfies $s_i \geq s'/2$. Hence, with probability at least $s'/2$ over the choice of component in step 1 the tester rejects with probability at least $1 - 2^{-2(s'/2)/s'} \geq 1/2$. Hence, $\mathcal{T}$ rejects w with probability at least $(s'/2) \cdot (1/2) = s'/4 = (s(\delta) - \varepsilon)/4$. This completes the proof of the Theorem. □

7.2 Proof of Theorem 2.13

We are now ready to prove Theorem 2.13.

THEOREM 2.13 (RESTATED). *Let $\alpha \in (0, 1)$ be a positive constant and let $\mathcal{P} \triangleq \{P_n \subseteq \mathbb{F}^n : n \in \mathbb{N}\}$ be a family of $\mathbb{F}$-linear properties (codes) with dual distance at least αn and such that for some $\delta_0 \in (0, 1)$ they are not trivially δ_0-testable. For every $\varepsilon > 0$, there exists a $\beta > 0$ and $n_0 \in \mathbb{N}$ such that for any property $P_n \in \mathcal{P}$, $n > n_0$ the following is satisfied for all $\delta \in (0, \delta_0]$:*

$$\mathcal{S}^{P_n}_{\mathbf{uniqV}}\left(3, 2^{\beta n}, \delta\right) \leq \frac{2(1 + 2\varepsilon)}{3} \cdot \left(1 - \frac{1}{|\mathbb{F}|}\right).$$

PROOF. Let $\mathcal{V}$ be a unique verifier for P_n and let $s^{\mathcal{V}}(\delta)$ its soundness function. Let μ be the fraction of inspective queries made by $\mathcal{V}$. We have from Lemma 5.7 that

$$s^{\mathcal{V}}(\delta) \leq \min\left\{1 - \mu + \mathcal{S}^{P_n}_{\mathbf{uniqV_i}}\left(3, 2^{\beta n}, \delta\right), \mu\left(1 - \frac{1}{|\mathbb{F}|}\right)\right\}.$$

The above inequality is maximized when the two sides are equal, i.e.,

$$\mu = \left(1 + \mathcal{S}^{P_n}_{\mathbf{uniqV_i}}\left(3, 2^{\beta n}, \delta\right)\right)/(2 - 1/|\mathbb{F}|).$$

For this setting of μ, we have

$$\begin{aligned} s^{\mathcal{V}}(\delta) &\leq \left(1 + \mathcal{S}^{P_n}_{\mathbf{uniqV_i}}\left(3, 2^{\beta n}, \delta\right)\right) \cdot \frac{\left(1 - \frac{1}{|\mathbb{F}|}\right)}{\left(2 - \frac{1}{|\mathbb{F}|}\right)} \\ &\leq \left(1 + \mathcal{S}^{P_n}_{\mathbf{uniqV_i}}\left(3, 2^{\beta n}, \delta\right)\right) \cdot \frac{\left(1 - \frac{1}{|\mathbb{F}|}\right)}{\frac{3}{2}} \qquad \text{[Since } |\mathbb{F}| \geq 2] \end{aligned}$$

Corollary 7.1 implies that $\mathcal{S}^{P_n}_{\mathbf{uniqV_i}}\left(3, 2^{\beta n}, \delta\right) \leq 2\varepsilon$ which proves the theorem. □

8. SHORT LINEAR PCPPS

In this section, we show if we relax the restriction of soundness from optimal soundness to some soundness bounded away from 1, then, in fact, every $\mathbb{F}_2$-linear property has a 3-query linear verifier of quasilinear proof length and perfect completeness.

It has been shown [Dinur 2007; Ben-Sasson and Sudan 2008] that if $P \subset \{0, 1\}^n$ is a property that can be decided by a nondeterministic circuit of size t, then P has a $(3, t, t\text{polylog } t)$-verifier $\mathcal{V}$ with perfect completeness and *constant soundness*. Constant soundness means that for any δ there exists ε that depends only on δ, and is independent of n, such that the soundness function of $\mathcal{V}$ satisfies $s(\delta) > \varepsilon$. Next we claim that if P is $\mathbb{F}_2$-linear then $\mathcal{V}$ can be assumed without loss of generality to be $\mathbb{F}_2$-linear too.

In what follows, a $\mathbb{F}_2$-linear circuit is a multioutput circuit with fan-in and fan-out at most 2 comprised of gates that compute $\mathbb{F}_2$-addition. The property decided by a $\mathbb{F}_2$-linear circuit P is defined to be the space of inputs that cause all output gates to evaluate to 0. Notice that every $\mathbb{F}_2$-linear property $P \subset \mathbb{F}_2^n$ can be decided by such a circuit of size at most n^2.

LEMMA 8.1 (SHORT LINEAR PCPPS). *For every $\delta > 0$ there exists $\varepsilon = \varepsilon(\delta) > 0$ such that the following holds. Every $\mathbb{F}_2$-linear property $P \subseteq \mathbb{F}_2^n$ that can be decided by a $\mathbb{F}_2$-linear circuit of size m has a 3-query linear verifier accessing a proof of length $\ell = m \cdot \text{polylog}\,(n)$, that has perfect completeness and soundness function satisfying $s(\delta) \geq \varepsilon$.*

Moreover, the proof oracle is linear *in the input oracle, that is, there exists a $\mathbb{F}_2$-linear transformation $T : \mathbb{F}_2^n \to \mathbb{F}_2^\ell$ such that every $w \in P$ is accepted by the verifier in conjunction with the proof oracle $\pi_w = T(w)$.*

PROOF SKETCH. The results of Dinur [2007] and Ben-Sasson and Sudan [2008] imply all but the $\mathbb{F}_2$-linearity in the lemma stated above. It suffices to modify their PCPP construction so that the proof π_w for a word $w \in P$ will be given by a $\mathbb{F}_2$-linear transformation T. Then, consider the property

$$P' \subset \mathbb{F}_2^{n+\ell}, P' = \{w \circ \pi_w \mid w \in P\}.$$

By construction, P' is $\mathbb{F}_2$-linear. Hence, Ben-Sasson et al. [2005, Theorem 5.3] implies P' has a 3-query $\mathbb{F}_2$-linear tester and this tester is a $(3, n, \ell)$, $\mathbb{F}_2$-linear verifier for P with perfect completeness and soundness function as claimed.

Transforming the proof oracle of Ben-Sasson and Sudan [2008] into an $\mathbb{F}_2$-linear one involves inspecting the various steps in its construction and making sure each of them is $\mathbb{F}_2$-linear. This has already been argued for the closely related construction of Ben-Sasson et al. [2006] in Proposition 8.14 there. The key element in Ben-Sasson and Sudan [2008] that does not appear in Ben-Sasson et al. [2006] is the construction of PCPPs for Reed-Solomon codes. This construction can be verified to be given by a linear transformation by inspecting Section 6. In particular, let us follow the proof of Proposition 6.9 in Ben-Sasson and Sudan [2008] using the notation there. Let $\mathbb{F}$ be the finite field of characteristic 2 used there (and denoted there by GF(2^ℓ)). Let $p : \mathbb{F} \to \mathbb{F}$ be the evaluation of a polynomial P. The coefficients of the bivariate polynomial Q are obtained by a $\mathbb{F}$-linear transformation applied to the coefficients of P, because by construction (in Proposition 6.2) $Q = P \mod (y - q(x))$, and taking the remainder of P is a $\mathbb{F}$-linear operation. Hence, the function $f : S \to \mathbb{F}$ which is an evaluation of Q on a subset S of $\mathbb{F} \times \mathbb{F}$ is given by an $\mathbb{F}$-linear transformation applied to p. This implies that $f : S \cup T \to \mathbb{F}$ is also $\mathbb{F}$-linear in p. So, arguing inductively, the PCPP for an RS-codeword p is $\mathbb{F}$-linear in p and so it is also $\mathbb{F}_2$-linear in p. We assume p is itself obtained by a $\mathbb{F}_2$-linear transformation applied to w (by arguing along the lines of [Ben-Sasson et al. 2006, Proposition 8.14], details omitted). We conclude that the PCPP resulting from Ben-Sasson and Sudan [2008] is $\mathbb{F}_2$-linear in w.

We move on to the construction in Dinur [2007] and follow the proof of Dinur [2007, Theorem 9.1], using the notation given there. We assume we have at hand a proof of length $m \cdot \text{polylog}\, n$ obtained by applying a linear transformation to $w \in P$. This proof is viewed as a mapping $\sigma : V \to \mathbb{F}_2$ where V is the set of vertices of a constraint graph G. The first step in the proof of Dinur [2007, Theorem 9.1] is to construct $\sigma_1 : V_H \to \mathbb{F}_2$ where V_H replaces each vertex $v \in V$ by a "cloud" of vertices, denoted $[v]$, and σ_1 assigns the value $\sigma(v)$ to all vertices in $[v]$. Clearly, σ_1 is $\mathbb{F}_2$-linear in σ as it is obtained from σ by repetition. Next, an assignment $\sigma_2 : V_H \to \mathbb{F}_2^{d^{t/2}}$ is constructed from σ_1 by taking $\sigma_2(v)$ to be the value given by σ_1 to all vertices within distance $\leq t/2$ from v (d denotes the degree of the regular graph H). Being a repetition of σ_1, this transformation is also $\mathbb{F}_2$-linear. The final step is "alphabet reduction by composition" with an assignment tester, which is synonymous to a PCPP. In Dinur [2007], the long-code-based assignment tester is used. However, to maintain $\mathbb{F}_2$-linearity, we compose with the Hadamard based PCPP. In particular, for every $v \in V_H$ we replace $\sigma_2(v) \in \mathbb{F}_2^{d^{t/2}}$ with its Hadamard encoding which is an element of $\mathbb{F}_2^{2^{d^{t/2}}}$. Let us call the resulting assignment σ_3. Notice σ_3 is $\mathbb{F}_2$-linear in σ_2 because it is obtained by concatenation with a $\mathbb{F}_2$-linear code. We set $\sigma = \sigma_3$ and repeat this process ($\sigma \mapsto \sigma_1 \mapsto \sigma_2 \mapsto \sigma_3$) a number of times (see Dinur [2007, Section 8] for details), resulting in an $\mathbb{F}_2$-linear transformation that converts $w \in P$ into a proof of length m polylog n. This completes our proof-sketch. □

REFERENCES

ARORA, S., LUND, C., MOTWANI, R., SUDAN, M., AND SZEGEDY, M. 1998. Proof verification and the hardness of approximation problems. *J. ACM 45*, 3, 501–555. doi: 10.1145/278298.278306.

ARORA, S. AND SAFRA, S. 1998. Probabilistic checking of proofs: A new characterization of NP. *J. ACM 45*, 1, 70–122. doi: 10.1145/273865.273901.

BABAI, L., FORTNOW, L., LEVIN, L. A., AND SZEGEDY, M. 1991. Checking computations in polylogarithmic time. In *Proceedings of the 23rd ACM Symposium on Theory of Computing (STOC)*. ACM, 21–31. doi: 10.1145/103418.103428.

BELLARE, M., COPPERSMITH, D., HÅSTAD, J., KIWI, M. A., AND SUDAN, M. 1996. Linearity testing in characteristic two. *IEEE Trans. Inform. Theor. 42*, 6, 1781–1795. doi: 10.1109/18.556674.

BEN-SASSON, E., GOLDREICH, O., HARSHA, P., SUDAN, M., AND VADHAN, S. 2006. Robust PCPs of proximity, shorter PCPs and applications to coding. *SIAM J. Comput. 36*, 4, 889–974. doi: 10.1137/S0097539705446810.

BEN-SASSON, E., GOLDREICH, O., AND SUDAN, M. 2003. Bounds on 2-query codeword testing. In *Proceedings of the 7th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, S. Arora and A. Sahai, Eds. Lecture Notes in Computer Science, vol. 2764. Springer, 216–227. doi: 10.1007/b11961.

BEN-SASSON, E., HARSHA, P., LACHISH, O., AND MATSLIAH, A. 2008. Sound 3-query PCPPs are long. In *Proceedings of the 35th International Colloquium of Automata, Languages and Programming (ICALP), Part I*, L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, Eds. Lecture Notes in Computer Science, vol. 5125. Springer, 686–697. doi: 10.1007/978-3-540-70575-8_56.

BEN-SASSON, E., HARSHA, P., AND RASKHODNIKOVA, S. 2005. Some 3CNF properties are hard to test. *SIAM J. Comput. 35*, 1, 1–21. doi: 10.1137/S0097539704445445.

BEN-SASSON, E. AND SUDAN, M. 2008. Short PCPs with polylog query complexity. *SIAM J. Comput. 38*, 2, 551–607. doi: 10.1137/050646445.

BEN-SASSON, E., SUDAN, M., VADHAN, S., AND WIGDERSON, A. 2003. Randomness-efficient low degree tests and short PCPs via epsilon-biased sets. In *Proceedings of the 35th ACM Symposium on Theory of Computing (STOC)*. ACM, 612–621. doi: 10.1145/780542.780631.

BOGDANOV, A. 2005. Gap amplification fails below 1/2. Tech. rep. TR05-046, Electronic Colloquium on Computational Complexity. http://eccc.hpi-web.de/eccc-reports/2005/TR05-046/index.html.

CHARIKAR, M., MAKARYCHEV, K., AND MAKARYCHEV, Y. 2006. Near-optimal algorithms for unique games. In *Proceedings of the 38th ACM Symposium on Theory of Computing (STOC)*. ACM, 205–214. doi: 10.1145/1132516.1132547.

DINUR, I. 2007. The PCP theorem by gap amplification. *J. ACM 54*, 3, 12. doi: 10.1145/1236457.1236459.

DINUR, I. AND REINGOLD, O. 2006. Assignment testers: Towards a combinatorial proof of the PCP Theorem. *SIAM J. Comput. 36*, 975–1024. doi: 10.1137/S0097539705446962.

ENGEBRETSEN, L. AND HOLMERIN, J. 2008. More efficient queries in PCPs for NP and improved approximation hardness of maximum CSP. *Rand. Struct. Algor. 33*, 4, 497–514. doi: 10.1002/rsa.20226.

ERGÜN, F., KUMAR, R., AND RUBINFELD, R. 2004. Fast approximate probabilistically checkable proofs. *Inform. Computat. 189*, 2, 135–159. doi: 10.1016/j.ic.2003.09.005.

FEIGE, U. AND KILIAN, J. 1995. Impossibility results for recycling random bits in two-prover proof systems. In *Proceedings of the 27th ACM Symposium on Theory of Computing (STOC)*. ACM, 457–468. doi: 10.1145/225058.225183.

FISCHER, E. 2001. The art of uninformed decisions: A primer to property testing. *Bull. Eur. Assoc. Theoret. Comput. Sci. 75*, 97–126. The Computational Complexity Column. http://theorie.informatik.uni-ulm.de/Personen/toran/beatcs/.

FISCHER, E. AND FORTNOW, L. 2006. Tolerant versus intolerant testing for Boolean properties. *Theor. Comput. 2*, 1, 173–183. doi: 10.4086/toc.2006.v002a009.

GALLAGER, R. G. 1963. *Low Density Parity Check Codes*. MIT Press, Cambridge, MA.

GOLDREICH, O., GOLDWASSER, S., AND RON, D. 1998. Property testing and its connection to learning and approximation. *J. ACM 45*, 4, 653–750. doi: 10.1145/285055.285060.

GOLDREICH, O. AND SUDAN, M. 2006. Locally testable codes and PCPs of almost linear length. *J. ACM 53*, 4, 558–655. doi: 10.1145/1162349.1162351.

GUPTA, A. AND TALWAR, K. 2006. Approximating unique games. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 99–106. doi: 10.1145/1109557.1109569.

GURUSWAMI, V. 2006. On 2-query codeword testing with near-perfect completeness. In *Proceedings of the 17th International Symposium on Algorithms and Computation (ISAAC)*, T. Asano, Ed. Lecture Notes in Computer Science, vol. 4288. Springer, 267–276. doi: 10.1007/11940128_28.

GURUSWAMI, V., LEWIN, D., SUDAN, M., AND TREVISAN, L. 1998. A tight characterization of NP with 3-query PCPs. In *Proceedings of the 39th IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, 18–27. doi: 10.1109/SFCS.1998.743424.

GURUSWAMI, V. AND RUDRA, A. 2005. Tolerant locally testable codes. In *Proceedings of the 9th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, C. Chekuri, K. Jansen, J. D. P. Rolim, and L. Trevisan, Eds. Lecture Notes in Computer Science, vol. 3624. Springer, 306–317. doi: 10.1007/11538462_26.

HARSHA, P. AND SUDAN, M. 2000. Small PCPs with low query complexity. *Computat. Complex. 9*, 3–4, 157–201. doi: 10.1007/PL00001606.

HÅSTAD, J. 2001. Some optimal inapproximability results. *J. ACM 48*, 4, 798–859. doi: 10.1145/502090.502098.

HÅSTAD, J. AND KHOT, S. 2005. Query efficient pcps with perfect completeness. *Theor. Comput. 1*, 1, 119–148. doi: 10.4086/toc.2005.v001a007.

KATZ, J. AND TREVISAN, L. 2000. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC)*. ACM, 80–86. doi: 10.1145/335305.335315.

KERENIDIS, I. AND DE WOLF, R. 2004. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *J. Comput. Syst. Sci. 69*, 3, 395–420. doi: 10.1016/j.jcss.2004.04.007.

KHOT, S. 2002. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th ACM Symposium on Theory of Computing (STOC)*. ACM, 767–775. doi: 10.1145/509907.510017.

KHOT, S. AND SAKET, R. 2006. A 3-query nonadaptive PCP with perfect completeness. In *Proceedings of the 21st IEEE Conference on Computational Complexity*. IEEE, 159–169. doi: 10.1109/CCC.2006.5.

KILIAN, J. 1992. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *Proceedings of the 24th ACM Symposium on Theory of Computing (STOC)*. ACM, 723–732. doi: 10.1145/129712.129782.

LEIGHTON, F. T. AND RAO, S. 1999. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM 46*, 6, 787–832. doi: 10.1145/331524.331526.

MICALI, S. 2000. Computationally sound proofs. *SIAM J. Comput. 30*, 4, 1253–1298. doi: 10.1137/S0097539795284959.

MOSHKOVITZ, D. AND RAZ, R. 2007. Sub-constant error probabilistically checkable proof of almost linear size. Tech. rep. TR07-026, Electronic Colloquium on Computational Complexity. http://eccc.hpi-web.de/eccc-reports/2007/TR07-026/index.html.

MOSHKOVITZ, D. AND RAZ, R. 2008a. Sub-constant error low degree test of almost-linear size. *SIAM J. Comput. 38*, 1, 140–180. doi: 10.1137/060656838.

MOSHKOVITZ, D. AND RAZ, R. 2008b. Two query PCP with sub-constant error. In *Proceedings of the 49th IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, 314–323. doi: 10.1109/FOCS.2008.60.

PARNAS, M., RON, D., AND RUBINFELD, R. 2006. Tolerant property testing and distance approximation. *J. Comput. Syst. Sci. 72*, 6, 1012–1042. doi: 10.1016/j.jcss.2006.03.002.

POLISHCHUK, A. AND SPIELMAN, D. A. 1994. Nearly-linear size holographic proofs. In *Proceedings of the 26th ACM Symposium on Theory of Computing (STOC)*. ACM, 194–203. doi: 10.1145/195058.195132.

RAZ, R. 1998. A parallel repetition theorem. *SIAM J. Comput. 27*, 3, 763–803. doi: 10.1137/S0097539795280895.

SAMORODNITSKY, A. AND TREVISAN, L. 2000. A PCP characterization of NP with optimal amortized query complexity. In *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC)*. ACM, 191–199. doi: 10.1145/335305.335329.

SAMORODNITSKY, A. AND TREVISAN, L. 2006. Gowers uniformity, influence of variables, and PCPs. In *Proceedings of the 38th ACM Symposium on Theory of Computing (STOC)*. ACM, 11–20. doi: 10.1145/1132516.1132519.

SZEGEDY, M. 1999. Many-valued logics and holographic proofs. In *Proceedings of the 26th International Colloquium of Automata, Languages and Programming (ICALP)*, J. Wiedermann, P. van Emde Boas, and M. Nien, Eds. Lecture Notes in Computer Science, vol. 1644. Springer, 676–686. doi: 10.1007/3-540-48523-6.

TREVISAN, L. 2005. Approximation algorithms for unique games. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, 197–205. doi: 10.1109/SFCS.2005.22.

WOODRUFF, D. P. 2008. Corruption and recovery-efficient locally decodable codes. In *Proceedings of the 12th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, A. Goel, K. Jansen, J. D. P. Rolim, and R. Rubinfeld, Eds. Lecture Notes in Computer Science, vol. 5171. Springer, 584–595. doi: 10.1007/978-3-540-85363-3_46.

ZWICK, U. 1998. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 201–210. doi:10.1145/314613.314701.

Received June 2008; revised March 2009; accepted March 2009