PCP Characterizations of NP:
Towards a Polynomially-Small Error-Probability

Irit Dinur* Eldar Fischer* Guy Kindler* Ran Raz† Shmuel Safra*

Abstract

This paper strengthens the low-error PCP characterization of NP, coming closer to the upper limit of the BGLR conjecture. Namely, we prove that witnesses for membership in any NP language can be verified with a constant number of accesses, and with an error probability exponentially small in the number of bits accessed, where this number is as high as $\log^{\beta} n$, for *any* constant $\beta < 1$. (The BGLR conjecture claims the same for any $\beta \leq 1$).

Our results are in fact stronger, implying the Gap-Quadratic-Solvability problem to be NP-hard even if the equations are restricted to having a constant number of variables. That is, given a system of quadratic-equations over a field $\mathcal{F}$ (of size up to $2^{\log^{\beta} n}$), where each equation depends on a constant number of variables, it is NP-hard to decide between the case where there is a common solution for all of the equations, and the case where any assignment satisfies no more than a $\frac{2}{|\mathcal{F}|}$ fraction of them.

At the same time, our proof presents a *direct* construction of a low-degree-test whose error-probability is exponentially small in the number of bits accessed. Such a result was previously known only relying on recursive applications of the entire PCP theorem.

Introduction

Each $L \in$ NP is reducible to 3-SAT, by Cook-Levin's characterization of NP. The reduction is a polynomial-time algorithm that accepts an input string I, and produces a set Ψ, of Boolean functions (local-tests), each depending on a constant number of variables, that are taken from some common set of variables. Ψ represents membership of I in L, in the sense that there exists an assignment satisfying all tests if and only if the input is in L.

* School of Mathematical Sciences, Tel Aviv University, ISRAEL

† Weizmann Inst. of Science , ISRAEL

A PCP characterization of NP differs from Cook's in regards to what is guaranteed in case the input is not in L: In Cook's characterization, one can only be sure that no assignment would satisfy all Ψ's local-tests. In a PCP characterization of NP, in contrast, it is guaranteed that no assignment satisfies even a *small fraction* of them.

A satisfying assignment to Ψ can be viewed as a witness to I's membership in L. According to the PCP characterization of NP, this witness can be verified by randomly picking one local-test of Ψ and checking whether it holds (hence the term PCP – Probabilistic Checking of Proofs). The *error probability* of a PCP characterization is a bound on the fraction of Ψ that can be satisfied in case the input is not in L.

For most applications of PCP, the characterization of NP with constant error-probability and variables of constant range [AS92, ALM+92] suffices. In order to prove NP-hardness of other problems, however, sub-constant error-probability has turned out to be essential. For example [LY94] and [BGLR93] were able to prove approximating SET-COVER to within logarithmic factors *almost* NP-hard, using the constant error-probability PCP characterization of NP. To improve this result to strict NP-hardness, [BGLR93] had suggested the 'sliding scale' conjecture.

The BGLR conjecture states that it is possible to keep the number of variables accessed by each local-test constant while increasing the variables' range, and to achieve error probability polynomially small in the size of the variables' range. In other words, a membership-proof can be verified by accessing a constant number of words, such that the error probability is exponentially small in the length of a word.

One cannot expect the error-probability to be less than polynomially small in the size of the variables' range, since a random assignment is expected to satisfy such a fraction, no matter what structure the local-tests take (recall that each test depends on a constant number of variables). Hence the BGLR conjecture is optimal in the sense of error-probability.

According to the BGLR conjecture, the variables' range may be up to polynomially large. It is unlikely to hold for variables' range larger than polynomial, since the error-probability would then become less than polynomially small in the size of Ψ, therefore in case the input is not in L, no local-test succeeds, making it easy to decide if the input is in L.

The conjecture was eventually shown true for a sizeable portion of the applicable range-size in [RS97], which showed a PCP characterization of NP where the error-probability is exponentially small in the size of the variables, as long as the variable size is up to $\log^{\beta} n$, for *some* positive constant β (see also [AS97]).

Our Main Results

In this paper, we improve the above result by increasing the range of β for which it holds, substituting "some constant" with "any constant smaller than 1", thus coming closer to proving the BGLR conjecture for the full applicable range. Our results are, in fact, stronger: we show that given a system of quadratic-equations over a field $\mathcal{F}$ ($|\mathcal{F}| \approx 2^{\log^{\beta} n}$ for any constant $\beta < 1$), where each equation depends on a constant number of variables, it is NP-hard to decide whether there is a common solution to all of the equations, or if every assignment satisfies no more than a $\frac{2}{|\mathcal{F}|}$ fraction.

Our proof also gives a direct construction of a low-degree-test whose error-probability is exponentially small in the number of bits accessed, as discussed below.

Related Results

We note that no PCP result obtains an exponentially small error probability for a constant number of accesses and $O(\log n)$ number of bits in each access, even when allowing the reduction to be *super-polynomial* - here n denotes the size of the generated system. The repetition lemma of [Raz98] shows that by two accesses to $O(\log n)$ bits, the error-probability is exponentially small, but only in the size of the original input, while the size of the construction is $n^{\log n}$. Similarly, the multi-linear extension of [BFL91] yields a system with a $\frac{1}{n}$ error-probability but whose size is $n^{\log n}$. In fact, in any known reduction there is always a factor of at least $\log^{\epsilon} n$ in the exponent that separates the error-probability from the size of the generated instance.

Reaching an error-probability polynomially small in the size of the generated instance is an important open problem. Such a characterization of NP would improve hardness results for several problems. For example, approximating the 'Monotone-Minimum-Satisfying-Assignment' problem (which is closely related to approximating the length of propositional proofs [ABMP98]) has been shown NP-hard in [DS98] via a reduction from PCP (such that the hardness of approximation ratio is preserved). Hence a polynomially small error-probability PCP characterization of NP would immediately imply that it is NP-hard to approximate the length of propositional proofs to within an n^{ϵ} factor for some constant $\epsilon > 0$.

[RS97] managed to keep the exponential relation between the number of bits accessed and the error-probability, thus showing the BGLR conjecture true for variables' range-size of up to $2^{\log^{\beta} n}$ for *some* constant $\beta < 1$. For larger β (any constant $\beta < 1$) [RS97] showed a system whose error probability is $2^{-\log^{\beta} n}$, yet without the exponential relation between the number of bits accessed and the error-probability (the number of bits accessed was $O(\log^{\beta} n \cdot \text{poly}\log\log n)$). This factor of $\text{poly}\log\log n$ is significant when viewing, for example, the result in terms of Gap-Quadratic-Solvability. The result of [RS97], if it were to be translated to Gap-Quadratic-Solvability terms, would at best give an equation system with each equation depending on $O(\text{poly}\log\log n)$ variables. In comparison, our result translates to a quadratic equation-system with the same error-probability, but where every equation depends on a *constant* $O(\frac{1}{1-\beta})$ number of variables.

An essential component of the known proofs for PCP characterizations of NP, involves a consistency test for low-degree-polynomials called low-degree-test. A low-degree-test of small error-probability seems necessary in order to show a PCP characterization of NP of small error-probability. Such low-degree-tests with small error-probability appeared in [RS97, AS97].

Technique

We use the general framework of [AS92, ALM+92, RS97] for our proof, however replace the generalized form of the composition paradigm utilized in previous PCP proofs, by a more concrete representation. Our result could have been obtained using the previous structure, however this representation simplifies our proof, and some of the techniques may be of independent interest.

Our proof takes the form of a series of linear (degree preserving) transformations of quadratic equation-systems. All of these equation-systems are over the same field. These two ideas – namely fixing the field throughout the proof and utilizing linear transformations – combined, overcome the barrier the proof technique of [RS97] couldn't, enabling the claimed NP-hardness of Gap-Quadratic-Solvability for a field of size $2^{\log^{\beta} n}$, for any constant $\beta < 1$, unlike *some* $\beta < 1$ in [RS97].

Instead of using recursive applications of the entire PCP theorem, we separate the proof into two parts. The first part consists of iterative applications of the *sum-check* technique. The second, *low-degree-test* part, shows a concrete representation of an LDF (LDF = low-degree function) and a direct low-degree-test whose error-probability is exponentially small in the number of bits accessed.

A low-degree-test of small (sub-constant) error-probability seems necessary in order to show a PCP characterization of NP of small (sub-constant) error-probability. Such low-degree-tests with small error-probability appeared in [RS97, AS97]. [RS97] introduced a new low-degree-test (plane-vs.-plane) and showed it to be of small error-probability. The previously used line-vs.-point test was shown by [AS97] to be of small error-probability too.

The error probability of all these tests is still not exponentially small in the number of bits accessed. In fact, in any such direct low-degree-test comparing subspaces (lines, planes, etc.) for consistency, the error-probability can be no less than polynomially small in the number of bits accessed. One can attain exponentially small error-probability by utilizing the composition technique, applying the entire PCP theorem to the known low-degree-test. Our proof, in contrast, makes the recursion concrete, utilizing an explicit tree-like representation of LDFs, yielding a direct low-degree-test.

The proof employs two techniques to obtain such an explicit representation for an LDF. The first technique, called *cube-representation* (section 3.1), specifies how an LDF over a multi-dimensional space can be represented by its restrictions to cubes (affine subspaces of constant dimension). The second is a new technique, called *embedding* *extension* (section 3.2), which embeds an LDF over a space of constant dimension as a function of significantly lower degree over a space of higher dimension. The alternating application of the cube-representation and the embedding techniques enables a concrete representation of an LDF, thus eliminating the need for recursive application of the entire PCP theorem.

Gap-Quadratic-Solvability

Our theorem is in fact stronger than the PCP characterization claimed, imposing additional structure on the local-tests, namely requiring the tests to be quadratic-equations over some finite field. We define the gap version of the Maximum Quadratic-Solvability problem as follows:

Definition 1 (Gap-Quadratic-Solvability) *The gap-Quadratic-Solvability problem with parameters D, $\mathcal{F}$ and ϵ, denoted gap-QS$[D,\mathcal{F},\epsilon]$, is as follows. An instance of the problem is a set of n quadratic-equations over a finite field $\mathcal{F} = \mathbb{Z}_p$ for a prime p, where each equation has at most D variables. The problem is to distinguish between the following two cases:*

Yes. There is an assignment to the variables that satisfies all of the equations.

No. Every assignment to the variables satisfies at most an $\epsilon(n)$ fraction of the equations.

An instance which falls under one of the above criteria is said to have the gap property. Any outcome is acceptable for instances that do not have the gap property.

Our main theorem shows NP-hardness of gap-QS for a field of size $|\mathcal{F}| = 2^{c\log^{\beta} n}$ for some constant c, and an error parameter $\epsilon = \frac{2}{|\mathcal{F}|}$. Note that the requirement for a $\frac{2}{|\mathcal{F}|}$ error is almost optimal, since it is easy to satisfy a $\frac{1}{|\mathcal{F}|}$ fraction of any system by a random assignment. On the other hand, from any error $\epsilon = \frac{1}{|\mathcal{F}|^c}$ which is polynomially small in the size of the field, one can obtain a $\frac{2}{|\mathcal{F}|}$ error by a simple amplification technique.

We therefore abbreviate gap-QS$[D,\mathcal{F}]$ for the gap-QS problem where ϵ is fixed to be $\frac{2}{|\mathcal{F}|}$. This error probability is polynomially small in the size of the field, and therefore exponentially small in the size of each variable (namely $\log|\mathcal{F}|$).

Theorem 1 (Main Theorem) *For every constant $\beta < 1$ there exists a constant $c > 0$ such that gap-QS$[O(1),\mathcal{F}]$ is NP-hard, where $|\mathcal{F}| = 2^{c\log^{\beta} n}$.*

This theorem holds (and obtained via the same techniques) for any field size $|\mathcal{F}| < 2^{\log^{\beta} n}$. In this paper, however, we focus only on the largest size.

Gap-QS$[n,\mathcal{F}]$, where the number of variables in each equation is not bounded, is proven NP-hard in [HPS93] for any polynomially bounded field size, using simple linear codes:

Theorem 2 *([HPS93]) Gap-QS$[n,\mathcal{F}]$ is NP-hard, for any polynomially bounded field size.*

Hence the entire difference between Cook-Levin's characterization of NP (from which theorem 2 follows) and the PCP characterization boils down to reducing the number of variables each equation accesses to constant. The proof commences with $\Psi_0 \in$ gap-QS$[n,\mathcal{F}]$ and by a series of degree preserving (linear) transformations arrives at $\Psi_l \in$ gap-QS$[O(1),\mathcal{F}]$. Ψ_l is an equation-system that enjoys the same gap in the fraction of satisfiable equations yet each of its equations accesses a constant number of variables. Namely **either** all of the equations are satisfiable **or** no more than a $\frac{2}{|\mathcal{F}|}$ fraction. An essential ingredient of our proof is that the field remains fixed throughout this series of transformations. Writing the size of $\mathcal{F}$ in terms of the size of Ψ_l, m, yields $|\mathcal{F}| = 2^{c\log^{\beta} m}$ for some constant c as required by theorem 1, since m is polynomial in n.

Structure of the Reduction

The reduction is broken into two parts. The first part of the reduction, the sum-check part (summarized in lemma 1), transforms each equation of Ψ_0 by a series of reductions, to many new equations, each of which accesses a constant number of *encoding variables*. These variables encode Ψ_0's variables – as often done in proofs of PCP characterizations of NP – by their low-degree-extension (see definition 5). This means that the new set of encoding variables has one variable for each point in some geometric domain, and assignments are said to be *legal encodings* if these variables receive the point evaluation of some LDF (=low-degree-function, see definition 4) over the domain. The new equations interpolate the values of old variables by *linear* combinations of encoding variables, therefore the degree of the equation-system is maintained.

Given an assignment to the new variables which is a legal encoding, the 'sum-check' simulates (with a small error) an equation depending on up to n variables by many equations, each accessing fewer variables. The sum-check part performs this 'simulation' repeatedly, eventually arriving at a system of equations that each access a constant number of variables.

The resulting equation-system has certain subsets of variables designated each to some geometric domain, and whose values supposedly represent the low-degree extension (see definition 5) of the values assigned to variables of the old system. Assignments to the variables are considered *legal encodings* if they assign LDFs of a certain degree to these subsets. We prove that in the end of the sum-check part, every assignment that is a legal encoding either satisfies all of the equations or less than an ϵ fraction. The proof of lemma 1 relies mainly on previously known techniques, and is omitted due to space limitations.

The second part of the reduction, the low-degree-test part (section 3), is aimed at getting rid of the 'encoding' assumption. It transforms the equation-system generated by the sum-check into an equation-system as in theorem 1. This goal is achieved by explicitly representing the LDFs in a way that allows checking consistency while keeping the number of variables in each equations constant (lemma 2). This representation is accomplished by alternately applying the cube-representation method, that appears in subsection 3.1, and the embedding extension, appearing in subsection 3.2.

The cube-representation technique allows the representation of LDFs over a multi-dimensional domain $\mathcal{F}^d$, by their restriction to affine-subspaces of constant degree. We show that in such a representation, one can consistently read a value of a point, by reading it from a random affine-subspace (containing the point) and verifying the consistency of the affine subspace by comparing it with the global domain $\mathcal{F}^d$ on a random point.

The embedding extension is a new technique showing how to represent an LDF over a constant-dimensional space, by a significantly lower degree LDF over a higher dimensional space.

These two techniques, when alternately combined, repeatedly reduce the degree and the dimension of the representation, until eventually we obtain a representation of an LDF that enables consistent point evaluation by a constant number of accesses.

We begin in section 1, by giving preliminary notions and definitions. We then proceed in section 2 to show the reduction, based on lemma 1 and lemma 2. Section 3, as mentioned, contains the LDF tree-like representation.

1 Preliminaries

This separation is formalized by a mechanism called *assumptions*. Let $\text{Assign}(\mathcal{V}) \stackrel{def}{=} \{A \mid A : \mathcal{V} \rightarrow \mathcal{F}\}$ be the set of possible assignments to the variables $\mathcal{V}$ of an equation-system Ψ. For any assignment $A \in \text{Assign}(\mathcal{V})$ we define the satisfiability of Ψ by A as

$$s(\Psi, A) \stackrel{def}{=} \Pr_{\psi \in \Psi}[\psi \text{ is satisfied by } A]$$

and the satisfiability of Ψ as

$$s(\Psi) \stackrel{def}{=} \max_{A \in \textbf{Assign}(\mathcal{V})}[s(\Psi, A)]$$

The initial and final systems Ψ_0 and Ψ_l are a special kind of promise problems, namely gap problems. The promise over them is that **either** the equation-system is totally satisfiable, **or** no more than ϵ-satisfiable – $s(\Psi_0), s(\Psi_l) \notin (\epsilon, 1)$. Ψ_l is obtained from Ψ_0 by a series of transformations $\Psi_0 \Rightarrow \Psi_1 \Rightarrow \ldots \Rightarrow \Psi_l$. The intermediate systems Ψ_i are also promise problems, nevertheless, the structure of their promise is more complex than the fractional gap.

Each equation-system Ψ_{i+1} is obtained from Ψ_i by encoding Ψ_i's variables into new variables and then having Ψ_{i+1}'s equations access these new variables. For each assignment A to Ψ_i's variables there is an assignment A' to Ψ_{i+1}'s variables that encodes it - A' is called a *legal encoding* of A. If we were sure Ψ_{i+1} is only assigned legal encodings we could maintain the gap. The gap of the equation system now becomes more complex, namely, **either** there is an assignment satisfying all of the equations, **or** (and this is the more complex part) every assignment to the variables – such that the values on the new variables *form a legal encoding* – cannot satisfy more than ϵ of the equations. In case Ψ_i is totally satisfiable by an assignment A, then its encoding A' totally satisfies Ψ_{i+1}. This means that we know that in the 'yes' case, the system is totally satisfiable by a legal encoding.

Assumptions

The promise described above guarantees that the equation-system Ψ falls into one of the following two cases,

Yes. There is an assignment satisfying all of the equations in Ψ, *and such that the assignment to certain designated variables is from a restricted set of 'legal assignments'.*

No. No *legal assignment* can satisfy more than an ϵ fraction of Ψ.

Let $\mathsf{A}_{yes} \subseteq \text{Assign}(\mathcal{V})$ be a set of assignments for Ψ (corresponding to legal encodings of assignments to the previous system). When we reduce Ψ to a system Ψ', we show how an assignment in A_{yes} can be encoded as a legal assignment of Ψ'. Sometimes, however, a legal assignment for Ψ' cannot be decoded back to a legal assignment for Ψ, but only to an assignment which is 'very close' to being legal. Let $\mathsf{A}_{no} \supset \mathsf{A}_{yes}$ be a possibly larger set of assignments (containing decodings of legal assignments of the previous system). An *assumption* over $\mathcal{V}$ is defined as a pair $(\mathsf{A}_{yes}, \mathsf{A}_{no})$ of arbitrary subsets $\mathsf{A}_{yes} \subseteq \mathsf{A}_{no} \subseteq \text{Assign}(\mathcal{V})$. We define a gap-under-assumption as follows,

Definition 2 ($(\epsilon, 1)$-Gap Under Assumption) *Let Ψ be an equation-system over a variable set $\mathcal{V}$, and let $(\mathsf{A}_{yes}, \mathsf{A}_{no})$ be an assumption for Ψ. Ψ is said to have an $(\epsilon, 1)$-gap under the assumption $(\mathsf{A}_{yes}, \mathsf{A}_{no})$ if exactly one of the following holds:*

Yes. There exists an assignment A to $\mathcal{V}$, $A \in \mathsf{A}_{yes}$, that satisfies Ψ.

No. Every assignment A to $\mathcal{V}$, $A \in \mathsf{A}_{no}$, satisfies no more than an ϵ fraction of Ψ.

In other words, if we generalize the satisfiability function for restricted subsets of assignments by

$$s_{\mathsf{A}}(\Psi) \stackrel{def}{=} \max_{A \in \mathsf{A}}(s(\Psi, A))$$

then a gap-under-assumption is a stronger promise on Ψ, stating that in the yes case $s_{\mathsf{A}_{yes}}(\Psi) = 1$ rather than $s(\Psi) = 1$, and in the no case $s_{\mathsf{A}_{no}}(\Psi) \leq \epsilon$ rather than $s(\Psi) \leq \epsilon$.

Let us go back to the series of reductions of equation-systems, $\Psi_0 \Rightarrow \Psi_1 \Rightarrow \cdots \Rightarrow \Psi_l$. In each reduction the variables $\mathcal{V}_i$ of Ψ_i are encoded by the variables $\mathcal{V}_{i+1}$ of Ψ_{i+1}. We will need to show that the gap is maintained by the reduction. The easy part will always be to show how a legal assignment that totally satisfies Ψ_i can be encoded into a totally satisfying assignment for Ψ_{i+1}. The harder part will be to show that any assignment for Ψ_{i+1} can be 'decoded-back' into an assignment satisfying roughly the same fraction of Ψ_i. This is the counter-positive way of saying that no-instances are mapped into no-instances.

Definition 3 (Gap-Maintaining Reduction) *Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two sets of equation-systems, and let R be a polynomial algorithm that on input $\Psi \in \mathcal{E}_1$ outputs an equation-system $\Psi' \in \mathcal{E}_2$.*

For a system $\Psi \in \mathcal{E}_1$, we denote its assumption by $\mathsf{A} = (\mathsf{A}_{yes}, \mathsf{A}_{no})$, and denote the assumption of the system R generates from Ψ, Ψ', by $\mathsf{A}' = (\mathsf{A'}_{yes}, \mathsf{A'}_{no})$. R is said to be a gap-maintaining reduction between $\mathcal{E}_1$ and $\mathcal{E}_2$ if there exists a constant $c > 0$ such that Ψ and Ψ' are assured to have the following properties:

1. *If* $(s_{\mathsf{A}_{yes}}(\Psi) = 1)$ *then also* $(s_{\mathsf{A}'_{yes}}(\Psi') = 1)$.

2. *For any assignment* $\mathsf{A}' \in \mathsf{A}'_{no}$ *there exists an assignment* $A \in \mathsf{A}_{no}$ *that satisfies almost the same fraction of equations, i.e.* $s(\Psi, \mathsf{A}) \geq s(\Psi', \mathsf{A}') - |\mathcal{F}|^{-c}$

In that case we write $\Psi \Rightarrow \Psi'$, *meaning that* (1) *and* (2) *hold for* Ψ *and* Ψ'. *If there exists a gap-maintaining reduction between* $\mathcal{E}_1$ *and* $\mathcal{E}_2$, *we write* $\mathcal{E}_1 \Rightarrow \mathcal{E}_2$

If Ψ_{i+1} is generated from Ψ_i by a gap-maintaining reduction, by property 1 the gap is indeed maintained - if Ψ_i is satisfiable then so is Ψ_{i+1}. On the other hand, if it is not even ϵ satisfiable then Ψ_{i+1} is at most $\epsilon + |\mathcal{F}|^{-c}$ satisfiable, by property 2 in the definition (we later take care of the additional $|\mathcal{F}|^{-c}$).

During the first half of the construction (the sum-check part), we repeatedly reduce an equation-system Ψ with a gap under assumption to a new equation-system by adding to its variables $\mathcal{V}$ new variables $\mathcal{V}'$ that serve as some encoding of the old variables. We then prove that the new system has a gap under the old assumption $(\mathsf{A}_{yes}, \mathsf{A}_{no})$ (that only restricts the assignments to $\mathcal{V}$) *and* under a new assumption $(\mathsf{A}'_{yes}, \mathsf{A}'_{no})$ over the new variables $\mathcal{V}'$. In other words, the assumption of the new system will be the intersection of the old and the new assumptions,

$$(\mathsf{A}_{yes}, \mathsf{A}_{no}) \cap (\mathsf{A}'_{yes}, \mathsf{A}'_{no}) \stackrel{def}{=} (\mathsf{A}_{yes} \cap \mathsf{A}'_{yes}, \mathsf{A}_{no} \cap \mathsf{A}'_{no})$$

When we say that an equation system has a gap under several assumptions, it is implied that the system has a gap under the intersection of the assumptions.

Encodings and Extensions

In each step of our series of reductions, we generate a new system from an old one by *extending* some subsets of the variables, such that the extended variables *encode* the old ones. In the intermediate steps of the reduction, we set *assumptions* allowing only assignments that are legal encodings over the extensions. The most frequently used extension technique is the *low-degree extension* which extends assignments via low-degree functions. An additional type of extension, called the *multiplicative-extension*, is used exactly once in the proof.

Low–Degree Extension

Before going into the definition of LDF extension, let us first define an LDF formally,

Definition 4 (Low Degree Function - LDF) *A function* $f : \mathcal{F}^d \to \mathcal{F}$ *is said to have degree* r, *and called an* $[r, d]$*-LDF, if its values are the point evaluation of an* r *degree polynomial on* $\mathcal{F}^d$.

A Low-Degree extension of a set $\mathcal{V}$ of variables, is done by identifying each variable with a point in $\mathcal{F}^d$, and then assigning new variables for the rest of the points in $\mathcal{F}^d$. An assignment is now viewed as a function from $\mathcal{F}^d$ to $\mathcal{F}$. The assignment of $\mathcal{V}$ can then be encoded as an LDF over the extended set.

Definition 5 (Low-Degree Extension) *Let* $h < |\mathcal{F}|^{\frac{1}{2}}$, *and define* $\mathcal{H} = \{0, 1, \ldots, h\} \subseteq \mathcal{F}$. *Let* $\mathcal{V}$ *be an arbitrary set of variables, and take* $d \stackrel{def}{=} \log_{h+1}(|\mathcal{V}|)$. [1] *Identify every point* $x \in \mathcal{H}^d$ *with a distinct arbitrary variable* $\mathcal{V}[x]$.

Now add a new variable $\mathcal{V}[x]$, *for each point* $x \in \mathcal{F}^d \setminus \mathcal{H}^d$. *The extended set of variables* $\widehat{\mathcal{V}} \stackrel{def}{=} \{\mathcal{V}[x] \mid x \in \mathcal{F}^d\}$, *together with the identification* $x \to \mathcal{V}[x]$, *is called the low-degree extension of* $\mathcal{V}$ *with parameter* h.

Given an arbitrary assignment $A : \mathcal{V} \to \mathcal{F}$, *we say that the assignment* $\widehat{A} : \widehat{\mathcal{V}} \to \mathcal{F}$ *is a* **proper-extension** *of* A *of degree* r *if* $\widehat{A}$ *coincides with* A, *and also the correspondence* $x \to \widehat{A}(\mathcal{V}[x])$ *is an* $[r, d]$*-LDF.*

To ensure that the extended variable set is assigned a legal encoding, we introduce an LDF-assumption over it.

Definition 6 (LDF-Assumption) *Let* $r_{yes} \leq r_{no} \ll |\mathcal{F}|$. *Let* $\mathcal{D}$ *be a variable set, containing a distinct variable* $\mathcal{D}[x]$ *for each* $x \in \mathcal{F}^d$. *The assumption* $(\mathsf{A}_{yes}, \mathsf{A}_{no})$ *defined by*

$$\mathsf{A}_{yes} = \{A \mid \textit{The mapping } x \to A[\mathcal{D}[x]] \textit{is an } [r_{yes}, d] - LDF \textit{ on } \mathcal{F}^d\}$$

$$\mathsf{A}_{no} = \{A \mid \textit{The mapping } x \to A[\mathcal{D}[x]] \textit{is an } [r_{no}, d] - LDF \textit{ on } \mathcal{F}^d\}$$

is called an $[r_{yes}, r_{no}, d]$*-LDF-assumption over* $\mathcal{D}$, *and denoted by* $\mathcal{L}_{\mathcal{D}}[r_{yes}, r_{no}, d]$. $\mathcal{D}$ *is called the domain of the assumption, or its assumption-domain (the assumption only restricts the assignments to the variables in its domain).*

LDF-assumptions can be joined by intersection, like every assumption. However, we will only join LDF-assumptions over *pairwise disjoint* domains, so a variable will belong to the domain of at most one LDF-assumption. The reason for this is that we will need, at a later stage of the proof, to get rid of these assumptions - and we do it by adding variables to the equations that verify them. If a variable were to participate in *many* LDF-assumptions, it would over-increase the number of variables in the equations of the final equation-system.

If $\mathcal{D}_1, \ldots, \mathcal{D}_s$ are disjoint variable sets, each equipped with a correspondence between its variables and the points of $\mathcal{F}^d$, then the *compound* LDF-assumption $\mathcal{L}_{\mathcal{D}_1,\ldots,\mathcal{D}_s}[r_{yes}, r_{no}, d]$ is defined as the intersection

$$\mathcal{L}_{\mathcal{D}_1}[r_{yes}, r_{no}, d] \cap \mathcal{L}_{\mathcal{D}_2}[r_{yes}, r_{no}, d] \cap \ldots \cap \mathcal{L}_{\mathcal{D}_s}[r_{yes}, r_{no}, d]$$

We sometimes omit the list of domains and refer to a compound LDF-assumption by $\mathcal{L}[r_{yes}, r_{no}, d]$.

Multiplicative Extension

In the reduction we transform Ψ_0 from quadratic to linear (before applying the sum-check technique) using the multiplicative extension, that adds new variables instead of quadratic terms. We take the set of variables, make an

[1] w.l.o.g. we assume that every degree, dimension, etc. is a natural number.

LDF extension, and then add a new variable for each product of two variables.

As the variables of the LDF extension are identified with the points of a geometric domain $\mathcal{F}^d$, the variables of the multiplicative extension will be naturally identified with $\mathcal{F}^{2d}$ - the variable representing a product of $\mathcal{V}[x]$ and $\mathcal{V}[y]$, will be identified with the point (x, y).

Definition 7 (Multiplicative-Extension) *Let $h < |\mathcal{F}|^{\frac{1}{2}}$, and let $\mathcal{V}$ be an arbitrary set of variables. Let v_e be a new variable (which will represent the value 1), and take $\widehat{\mathcal{V}}$ to be the low-degree-extension of $\mathcal{V} \cup \{v_e\}$, with parameter h. We denote, as in definition 5, $\mathcal{H} = \{0, .., h\} \subset \mathcal{F}$ and $d \stackrel{def}{=} \log_{h+1}(|\mathcal{V}| + 1)$, so $\left|\widehat{\mathcal{V}}\right| = |\mathcal{F}|^d$. Every point $x \in \mathcal{H}^d$ now corresponds to a variable in $\mathcal{V} \cup \{v_e\}$, denoted $\mathcal{V}[x]$, and denote by $e \in \mathcal{H}^d$ the point such that $v_e = \mathcal{V}[e]$.*

The variables in $\widehat{\mathcal{V}}$ are associated with the points of $\mathcal{F}^d$. Let us now associate them with points in $\mathcal{F}^{2d}$ instead, by identifying each $(x, e) \in \mathcal{F}^{2d}$ with $x \in \mathcal{F}^d$. We add new variables, corresponding to the rest of the points in $\mathcal{F}^{2d}$, and denote the entire variable set by $\mathcal{V}_\times$. $\mathcal{V}_\times$, together with the correspondence between its variables and the points of $\mathcal{F}^{2d}$, is called the multiplicative-extension of $\mathcal{V}$ with parameter h.

An assignment for a variable set $\mathcal{V}$ may be extended to an LDF over the LDF extension of $\mathcal{V}$. If in addition we have the multiplicative extension of $\mathcal{V}$, the assignment may be naturally extended further - the assignment for each product variable $\mathcal{V}_\times[x, y]$ will be assigned the product of the assignments to $\mathcal{V}[x]$ and $\mathcal{V}[y]$. Such an extension yields an LDF over the extension variables.

Definition 8 (Proper Multiplicative-Extension) *Let $A : \mathcal{V} \to \mathcal{F}$ be any assignment. An assignment $A_\times : \mathcal{V}_\times \to \mathcal{F}$ is called a proper multiplicative-extension of A, if the function f defined by*

$$\forall x \in \mathcal{F}^{2d} \qquad f(x) = A_\times(\mathcal{V}_\times[x])$$

is a $[2hd, 2d]$-LDF that obeys

1. $f(e, e) = 1$
2. $\forall x \in \mathcal{H}^d \quad f(x, e) = A(\mathcal{V}[x])$*, i.e. $A_\times$ coincides with A.*
3. $\forall x, y \in \mathcal{F}^d \quad f(x, y) = f(x, e) \cdot f(y, e)$.

To enforce assignments to $\mathcal{V}_\times$ to be proper multiplicative-extensions of the assignment of $\mathcal{V}$, we use an LDF-assumption over $\mathcal{V}_\times$, and accompany it by a multiplicative-assumption over $\mathcal{V}_\times$.

Definition 9 (Multiplicative-Assumption) *Let $\mathcal{V}_\times$ be a multiplicative extension with parameter h, of a variable-set $\mathcal{V}$. The multiplicative-assumption $\mathcal{X}$ over $\mathcal{V}_\times$ is defined by $\mathcal{X} \stackrel{def}{=} (\mathsf{A}, \mathsf{A})$ where A is the set of all assignments $A : \mathcal{V}_\times \to \mathcal{F}$ satisfying*

$$\forall (x, y) \in \mathcal{F}^{2d} \quad A(\mathcal{V}_\times[(x, y)]) = A(\mathcal{V}_\times[(x, e)]) \cdot A(\mathcal{V}_\times[(y, e)])\}$$

$$\textit{and} \qquad A(\mathcal{V}_\times[(e, e)]) = 1$$

1.1 Additional Definitions

In the proof of theorem 1 we define a series of equation-systems, as mentioned above. The intermediate systems in the series will be allowed to consist of conjunctions of equations (- *conjunctions* for short) rather than plain equations. The degree of such a conjunction is defined as the maximum of its equations' degrees, and its depend parameter is the total number of variables that appear in it.

An equation-system is said to be of *degree d*, (esp. *linear* or *quadratic*) if all of its conjunctions are of degree at most d. The *depend* of a system is the maximum depend of its conjunctions.

We deal with equations systems of varying parameters. Let us introduce a notation for an equation-system, that contains most of its relevant parameters.

Definition 10 *Let $\mathcal{V}$ be a set of variables, let $\mathcal{A}$ be a set of assumptions on the assignments to $\mathcal{V}$, and let $\epsilon > 0$ and $d, D \geq 1$. Denote by $\mathbf{EQ}^d_{\epsilon,D}(\mathcal{A}|\mathcal{V})$, the set of all degree-d equation-systems over variables $\mathcal{V}$ with depend D, that have an $(\epsilon, 1)$-gap under the assumptions in $\mathcal{A}$. All the above parameters may be functions of n, the size of the system, which is implicit here. Throughout the proof we only use $\mathbf{EQ}^d_{\ldots}(\ldots)$ where d is either 1 or 2.*

2 Main Theorem - the Reduction

In this section we give the reduction from gap-QS$[n, \mathcal{F}]$ to gap-QS$[O(1), \mathcal{F}]$, starting with Ψ_0 and generating a quadratic-equation-systems that maintains the gap of Ψ_0, but whose equations depend only on a constant number of variables each.

Reduction Sketch. We first give a sketch of the series of reductions we intend to apply. As already mentioned, the first part of the reduction is the sum-check - the sum-check lemma (lemma 1) reduces the depend of a given equation-system to constant, but it can only be applied to *linear* equation-systems. So we first transform Ψ_0 to a linear equation-system Ψ_1, using the multiplicative extension, that has a new variable for each quadratic term, and introduce a multiplicative-assumption to ensure consistency. Then we apply the sum-check lemma, obtaining an equation-system Ψ_2 with a constant depend parameter, and some additional LDF-assumptions.

Before we continue with the second part of the proof - the LDF-reader, we eliminate the multiplicative-assumption by substituting each variable back for its multiplicative-term (Ψ_3). We then use the LDF-reader (lemma 2) to eliminate the LDF-assumptions (Ψ_4). Finally, we amplify the error probability (Ψ_5) and use a simple technique to transform conjunctions of equations back into equations (Ψ_6).

We now proceed with the details of the reduction.

Eliminating Multiplicative Terms. We construct a *linear homogeneous* equation-system Ψ_1, from the quadratic Ψ_0. Let $\mathcal{V}_\times$ be the multiplicative-extension (see definition 7) of $\mathcal{V}$ with parameter h, where $h = |\mathcal{F}|^{c'}$ for some $0 < c' < \frac{1}{2}$ (e.g. take $c' = \frac{1}{4}$). For every $\psi \in \Psi_0$ we shall have a linear equation in Ψ_1, obtained by substituting each quadratic term $\mathcal{V}[x] \cdot \mathcal{V}[y]$ in ψ with a new variable $\mathcal{V}_\times[(x,y)]$. We also multiply the constants in ψ by the variable $\mathcal{V}_\times[(e,e)]$ to make it homogeneous.

Let $\mathcal{X}$ be the multiplicative-assumption (see definition 9) over $\mathcal{V}_\times$. We introduce $\mathcal{X}$ into Ψ_1, together with the LDF-assumption $\mathcal{L}_{\mathcal{V}_\times}[2hd, 2hd(\log n)^4, 2d]$ where $d \stackrel{def}{=} \log_{h+1}(|\mathcal{V}|+1)$. Note that every assignment to $\mathcal{V}$ can be extended to $\mathcal{V}_\times$ by the proper-extension, such that both $\mathcal{X}$ and the 'yes' part of the LDF-assumption are satisfied. Therefore Ψ_1 satisfies

$$\Psi_1 \in \mathbf{EQ}^1_{\epsilon_1, |\mathcal{V}_\times|}(\mathcal{X}, \mathcal{L}_{\mathcal{V}_\times}[2hd, 2hd(\log n)^3, 2d] \mid \mathcal{V}_\times)$$

where $\epsilon_1 = \frac{O(1)}{|\mathcal{F}|}$ and $\Psi_0 \Rightarrow \Psi_1$.

We use the next lemma to replace Ψ_1 with an equation-system of *constant* depend, in the price of adding new LDF-assumptions. Since Ψ_1 is linear homogeneous, it can be viewed as a set of weighted sums of variables, that are supposed to evaluate to zero. The sum-check lemma shows how to verify this by only accessing a constant number of variables.

Lemma 1 (Sum-Check) *Let $0 < c_0 < \frac{1}{2}$ be a constant. There exist constants $c_1, c_2, c_3 > 0$, and a polynomial-time algorithm, that, given an equation-system*

$$\Psi \in \mathbf{EQ}^1_{\epsilon, |\mathcal{V}|}(\mathcal{X}, \mathcal{L}_{\mathcal{V}}[r_{yes}, r_{no}, d] \mid \mathcal{V})$$

where $\mathcal{X}$ is an arbitrary assumption over $\mathcal{V}$, $r_{yes} \le |\mathcal{F}|^{c_0}$ and $r_{no} = r_{yes} \cdot (\log n)^4$ and where every conjunction $\psi \in \Psi$ is singleton (i.e. is actually an equation*); constructs an equation-system*

$$\widetilde{\Psi} \in \mathbf{EQ}^1_{\widetilde{\epsilon}, c_2}(\mathcal{X}, \mathcal{L}_{\mathcal{D}_1,..,\mathcal{D}_s}[\widetilde{r_{yes}}, r_{no}, c_3 d])$$

where

- $\Psi \Rightarrow \widetilde{\Psi}$
- $\widetilde{\epsilon} = \epsilon + |\mathcal{F}|^{-c_1}$*, and* $\widetilde{r_{yes}} = 2d(r_{yes}+1)$.

The proof of this lemma appears in the full version of the paper.

We apply the sum-check lemma to Ψ_1 and obtain Ψ_2, whose depend parameter is the constant c_2.

Removing the multiplicative-assumption. We will now bring the multiplicative-terms back into Ψ_2, and discard the multiplicative-assumption $\mathcal{X}$ without loosing the gap property. In every conjunction $\psi \in \Psi_2$, we replace every occurrence of a variable $\mathcal{V}_\times[(x,y)]$ where $y \neq e$, with the product $\mathcal{V}_\times[(x,e)] \cdot \mathcal{V}_\times[(y,e)]$. We also add to every conjunction the equation $(\mathcal{V}_\times[(e,e)] = 1)$. $\mathcal{X}$ may now be discarded without affecting the gap.

This yields a *quadratic* equation-system Ψ_3 such that $\Psi_2 \Rightarrow \Psi_3$ and Ψ_3 is in

$$\mathbf{EQ}^2_{\epsilon_3, 2c_2+1}(\mathcal{L}_{\mathcal{D}_1,..,\mathcal{D}_s}[2d(2hd+1),\ 2hd(\log n)^4,\ O(d)])$$

where $\epsilon_3 = \epsilon_2$.

Eliminating the LDF-assumptions. We next state the LDF-reader lemma that eliminates all the LDF-assumptions, and yields an equation-system with a gap in the conventional sense.

Lemma 2 (LDF-Reader) *Let $c_1, c_2 > 0$ and $0 < c_3, c_4 < 1$ be arbitrary constants. Let $r_{yes} < |\mathcal{F}|^{c_3}$ and let $\epsilon = |\mathcal{F}|^{-c_4}$. There exist $\omega < (\log n)^3$ and constants $c_5, c_6 > 0$ such that*

$$\mathbf{EQ}^2_{\epsilon, c_1}\left(\{\mathcal{L}_{\mathcal{D}_1,..,\mathcal{D}_s}[r_{yes}, \omega r_{yes}, c_2 \cdot \log_{|\mathcal{F}|} n]\}\right)$$

$$\Downarrow$$

$$\mathbf{EQ}^2_{\widetilde{\epsilon}, c_6}(\phi)$$

where $\widetilde{\epsilon} < \epsilon + \mathcal{F}^{c_5}$ and ϕ denotes the empty set.

Before we can apply the LDF-Reader lemma to Ψ_3 we need to adjust its parameters. We note that the family of equation-systems with a gap under certain LDF-assumptions, is monotone decreasing in the r_{no} parameter. This means that we may decrease the r_{no} parameter in the assumptions of an equation-system, and the gap will be maintained.

The ratio between the r_{no} and r_{yes} in the assumptions of Ψ_3 is $\Omega\left(\frac{\log^4 n}{d}\right)$, and thus bigger than $\log^3 n$. Therefore we may decrease the r_{no} parameter, and write that Ψ_3 is in

$$\mathbf{EQ}^2_{\epsilon_3, 2c_2+1}(\mathcal{L}_{\mathcal{D}_1,..,\mathcal{D}_s}[2d(2hd+1),\ \omega \cdot 2d(2hd+1),\ O(d)])$$

where ω is the parameter that appears in the LDF-reader lemma. Now we can apply the lemma to Ψ_3, obtaining Ψ_4. Ψ_4 is almost the desired gap-QS instance, except for two small differences: It is a system of conjunctions rather than equations, and its error-probability is some $|\mathcal{F}|^{-c}$ rather than $\frac{2}{|\mathcal{F}|}$.

We amplify the error-probability by setting $k = \lceil \frac{1}{c} \rceil$, and transforming $\Psi_4 = \{\psi_1, .., \psi_m\}$ into

$$\Psi_5 = \{(\psi_{i_1}) \wedge \cdots \wedge (\psi_{i_k}) \mid 1 \le i_1, \ldots, i_k \le m\}$$

The error probability is raised to the power k, thus obtaining a $(\frac{1}{|\mathcal{F}|}, 1)$-gap.

Ψ_5 is still a system of *conjunctions*, rather than equations. This is solved by the following (final) transformation. We replace every conjunction $\psi = \varphi_1 \wedge \cdots \wedge \varphi_s \in \Psi_5$ with all possible $(|\mathcal{F}|^s)$ linear combinations of the equations φ_i:

$$\Psi_\psi \stackrel{def}{=} \{\sum_{i=1}^{s} \alpha_i \cdot \varphi_i \ :\ \alpha_1, .., \alpha_s \in \mathcal{F}\}$$

where a linear combination of equations is defined in the obvious way.

It is easy to see that the system of equations $\Psi_6 = \cup_{\psi \in \Psi_5} \Psi_\psi$ obeys $\Psi_5 \Rightarrow \Psi_6$, since any assignment that did not satisfy a conjunction ψ, cannot satisfy more than $\frac{1}{|\mathcal{F}|}$ of the equations in Ψ_ψ. The error probability increases by no more than $\frac{1}{|\mathcal{F}|}$, hence $\Psi_6 \in$ gap-QS$[O(1), \mathcal{F}]$.

The reduction we have described implies that deciding whether $s(\Psi_6) = 1$ or $s(\Psi_6) \le \frac{2}{|\mathcal{F}|}$ enables deciding whether $s(\Psi_0) = 1$ or $s(\Psi_0) \le \frac{2}{|\mathcal{F}|}$; hence gap-QS$[O(1), \mathcal{F}]$ is NP-hard. ∎

3 Reading LDFs Consistently

In this section we prove the following lemma, showing how to eliminate LDF-assumptions, and remain with a gap in the conventional sense. This lemma can be alternatively viewed as a low-degree-test whose error-probability is exponentially small in the number of bits accessed.

Lemma 2 (LDF-Reader) *Let $c_1, c_2 > 0$ and $0 < c_3, c_4 < 1$ be arbitrary constants. Let $r_{yes} < |\mathcal{F}|^{c_3}$ and let $\epsilon = |\mathcal{F}|^{-c_4}$. There exist $\omega < (\log n)^3$ and constants $c_5, c_6 > 0$ such that*

$$\mathbf{EQ}^2_{\epsilon,c_1}\left(\{\mathcal{L}_{\mathcal{D}_1,..,\mathcal{D}_s}[r_{yes}, \omega r_{yes}, c_2 \cdot \log_{|\mathcal{F}|} n]\}\right)$$

$$\Downarrow$$

$$\mathbf{EQ}^2_{\widetilde{\epsilon},c_6}(\phi)$$

where $\widetilde{\epsilon} < \epsilon + \mathcal{F}^{c_5}$ and ϕ denotes the empty set.

We prove this lemma via three sub-lemmas. Let us first state the lemmas and sketch the proof, and then proceed to the proof itself.

The main idea is to take an equation-system Ψ and repeatedly replace its assumptions, until ultimately they can be easily verified (i.e. by accessing a constant number of additional variables). Then, by adding the appropriate verification tests to the conjunctions, we eliminate the need for assumptions altogether.

We employ two main techniques for achieving such equation-system reductions. One technique reduces the dimension of the LDFs in the LDF-assumptions, and the other reduces the degree (r_{no} and r_{yes}). Then, when the LDF-assumptions are of very low degree and dimension, (namely, they are $[O(\log\log n), O(\log\log n), O(1)]$ assumptions) we use a third linearization technique (similar to [ALM+92]) that eliminates the assumptions altogether (by verifying them in the body of the conjunctions).

The first technique, named *cube representation*, uses geometric properties of domains, and their sub-cubes, to replace $[r_{yes}, r_{no}, d]$-assumptions by $[r_{yes}, r_{no}, O(1)]$-assumptions hence the dimension of the LDF becomes constant, while the total degree remains the same:

Lemma 3 (Cube-Representation) *Fix a constant $0 < c < 1$, and let $r_{yes} \le |\mathcal{F}|^c$, and $r_{no} = \omega \cdot r_{yes}$ where $1 \le \omega < (\log n)^3$. Also assume $\epsilon = |\mathcal{F}|^{-c_2}$ for some $0 < c_2 < 1$, $d \le \log_{|\mathcal{F}|} n$ and $D = O(1)$. There exists $\alpha = |\mathcal{F}|^{-c_3}$, for some constant $0 < c_3 < 1$, such that*

$$\mathbf{EQ}^2_{\epsilon,D}(\{\mathcal{L}[r_{yes}, r_{no}, d]\})$$

$$\Downarrow$$

$$\mathbf{EQ}^2_{\epsilon+\alpha,3D}(\{\mathcal{L}[r_{yes}, r_{no}, D+3]\})$$

Moreover, there exists a reduction algorithm between the above sets with the following property. If K bounds, for an input system Ψ, the number of different assumption-domains that have variables in a single conjunction of Ψ, then

- *The depend parameter of the output system is only $D + 2K$.*
- *The bound, K, is maintained in the output system.*

The second technique, named *embedding extension*, replaces $[r_{yes}, r_{no}, O(1)]$ assumptions by $[\widetilde{r_{yes}}, \widetilde{r_{no}}, d]$-assumptions where $\widetilde{r_{no}}$ and $\widetilde{r_{yes}}$ are significantly smaller than r_{yes} and r_{no}, while the dimension d is only slightly increased.

In essence r_{yes} and r_{no} are of approximately the same size. For technical reasons, the embedding extension 'works' only if the ratio $\frac{r_{no}}{r_{yes}}$ is large enough. Every iterative application of the embedding extension technique 'eats up' some of this ratio. This is the reason we needed the sum-check lemma and the multiplication lemma to generate assumptions where $\frac{r_{no}}{r_{yes}} = (\log n)^{O(1)}$.

Lemma 4 (Embedding Extension) *Fix a constant t. For any $k = O(\log_{|\mathcal{F}|} n)$, if $\frac{r_{no}}{r_{yes}} \ge kt$ then*

$$\mathbf{EQ}^2_{\epsilon,D}(\{\mathcal{L}_{\mathcal{D}_1,..,\mathcal{D}_s}[r_{yes}, r_{no}, t]\})$$

$$\Downarrow$$

$$\mathbf{EQ}^2_{\epsilon,D}\left(\{\mathcal{L}_{\widetilde{\mathcal{D}}_1,..,\widetilde{\mathcal{D}}_s}[kt \cdot \sqrt[k]{r_{yes}},\ \frac{r_{no}}{r_{yes}} \cdot \sqrt[k]{r_{yes}},\ kt]\}\right)$$

Moreover, there is a linear reduction between the above, that maintains the bound on the number of different assumption-domains that appear in each conjunction.

This lemma holds, in fact, for any $t = O(\log_{|\mathcal{F}|} n)$. We state it for $t = O(1)$ since this how we use it.

The lemma can only be applied for $k = O(\log_{|\mathcal{F}|} n)$, hence the dominating term in the new degree will usually be $\sqrt[k]{r_{yes}}$. Since the k parameter determines how fast the degree decreases, choosing k as large as possible gives the greatest reduction of the degree, well worth the increase in the dimension.

We alternate the use of these two techniques a constant (at most $\lfloor \frac{1}{1-\beta} \rfloor + 2$) number of iterations until we reach an equation-system with $[O(\log\log n), O(\log\log n), O(1)]$-assumptions. We then use a linearization technique that by transforming the LDF-assumptions into *linear* assumptions, and then actually assuring the linear assumptions by interpolation, eliminates the need for assumptions altogether:

Lemma 5 (Linearization) *Fix $c > 0$ and let $\epsilon = |\mathcal{F}|^{-c}$. Then there exists some constant $0 < c' < 1$ such that*

$$\mathbf{EQ}^2_{\epsilon,O(1)}(\{\mathcal{L}[O(\log\log n), O(\log\log n), O(1)]\})$$

$$\Downarrow$$

$$\mathbf{EQ}^2_{\widetilde{\epsilon},O(1)}(\phi)$$

where $\widetilde{\epsilon} = \epsilon + |\mathcal{F}|^{c'}$

Having stated the lemmas, we turn to prove the LDF-reader lemma (lemma 2). The proofs of sub-lemmas 3 and 4 follow in the next subsections. Due to space limitations, the proof of lemma 5 only appears in the full version of the paper.

Proof: **(of lemma 2)** We begin with an equation-system

$$\Psi_0 \in \mathbf{EQ}^2_{\epsilon,O(1)}\left(\{\mathcal{L}[|\mathcal{F}|^{c_3}\ ,\ \omega \cdot |\mathcal{F}|^{c_3}\ ,\ O(\log_{|\mathcal{F}|} n)]\}\right)$$

where $\omega < \log^3 n$ is a parameter soon to be stated. We shall show a sequence of equation-reductions

$$\Psi_0 \Rightarrow \Psi_1 \Rightarrow \ldots \Rightarrow \Psi_l$$

such that $l = O(\frac{1}{1-\beta})$ and

$$\Psi_l \in \mathbf{EQ}^2_{\epsilon^{O(1)}, O(1)}(\phi)$$

Since the reduction relation '$\Rightarrow$' is transitive (as long as it is composed a constant number of times), we get $\Psi_0 \Rightarrow \Psi_l$ and end the proof. The transition from each system to the next is done via one of the above three lemmas. We begin by applying lemmas 3 and 4 iteratively, and end with one application of lemma 5.

Choosing Parameters. For convenience we denote $\delta \stackrel{def}{=} 1 - \beta$ and $\Delta \stackrel{def}{=} \lfloor \frac{1}{\delta} \rfloor$. We apply lemma 3 to Ψ_0, and then lemma 4 to the result, with parameter $k \stackrel{def}{=} \log_{|\mathcal{F}|} n$. The outcome is denoted Ψ_1. We will then apply this double transformation to Ψ_1 and get Ψ_2, and continue iteratively until we have Ψ_Δ. Then we will do it twice more, now with smaller k parameters, getting $\Psi_{\Delta+1}$ and $\Psi_{\Delta+2}$. Over all we apply the double iterations $\Delta + 2$ times at the most.

We next compute the initial ratio ω, between the r_{yes} and r_{no} parameters of Ψ_0, that we need in order to legally apply the sequence of transformations. Combining the application of lemma 3 and then lemma 4 with parameter k affects the parameters of a system Ψ as follows:

$$\mathbf{EQ}^2_{\epsilon,D}(\{\mathcal{L}[r_{yes}, r_{no}, d]\})$$

by lemma 3, $\Downarrow$

$$\mathbf{EQ}^2_{\epsilon+\alpha,D+K}(\{\mathcal{L}[r_{yes}, r_{no}, D+3]\})$$

by lemma 4, $\Downarrow$

$$\mathbf{EQ}^2_{\epsilon+\alpha,D+K}\left(\mathcal{L}\left[k(D+3)\cdot\sqrt[k]{r_{yes}}\ ,\ \frac{r_{no}}{r_{yes}}\cdot\sqrt[k]{r_{yes}}\ ,\ k(D+3)\right]\right)$$

For the first transformation to be legal we need $\frac{r_{no}}{r_{yes}} \leq \log^3 n$. We also want the final r_{no} parameter to be no smaller than the r_{yes} parameter. The double lemma application *decreases* the ratio between r_{no} and r_{yes} by a factor of $k(D+3)$, so for the iterative application we need the first $\frac{r_{no}}{r_{yes}}$ ratio to be smaller than $\log^3 n$, and the last ratio to be at least 1.

For each Ψ_i in the sequence, denote

$$\Psi_i \in \mathbf{EQ}^2_{\epsilon+i\alpha,D+2iK}(\mathcal{L}[r_{yes,i}\ ,\ r_{no,i}\ ,\ d_i])$$

Then $d_i = k(D_{i-1}+3) = k(D+2K(i-1)+3) \leq (D+2K(\Delta+2)+3)k \leq 10D\Delta k$ and therefore

$$\frac{r_{no,i}}{r_{yes,i}} = \frac{r_{no,i-1}}{d_i r_{yes,i-1}} \geq \frac{1}{10D\Delta k}\cdot\frac{r_{no,i-1}}{r_{yes,i-1}}$$

so each double transformation 'eats up' at most $10D\Delta k$ of the $\frac{r_{no}}{r_{yes}}$ ratio, where k is the parameter chosen for the embedding. The k's we use do not exceed $\log_{|\mathcal{F}|} n \equiv \log^\delta n$, and we make at most $\Delta + 2$ double transformations, so taking the initial ω to be

$$\omega \stackrel{def}{=} (10D\Delta\log^\delta n)^{\Delta+2} = O(\log^{\delta(\Delta+2)} n) < \log^3 n$$

ensures that all the transformations are legal, and that the final r_{no}/r_{yes} ratio is at least 1.

We apply the double transformation iteratively on Ψ_0, with parameter $k = \log_{|\mathcal{F}|} n$. By induction on the double transformation parameters, the size of $r_{no,i}, r_{yes,i}$, is

$$\forall 0 \leq i < \Delta \qquad r_{yes,i}, r_{no,i} \leq 2^{O((\log n)^{1-(i+1)\delta})}$$

specifically, for some $i \leq \Delta$,

$$r_{yes,\Delta}, r_{no,\Delta} \leq 2^{O(\log^\delta n)}$$

We stop iterating at the first i which satisfies the above. The obtained system is denoted by Ψ_Δ for simplicity.
The degree of the LDF-assumptions in Ψ_Δ is significantly smaller than what we started with, but it is still not small enough. We therefore repeat the double transformation twice more, now choosing smaller k parameters (since the other parameters have shrunk, k is now dominant in determining the assumptions degree of the result of the transformation).

We choose $k = \log_2(r_{yes,\Delta})$ for the first transformation, thus getting a system $\Psi_{\Delta+1}$, and then take $k = \log_2(r_{yes,\Delta+1})$ for the second transformation, which obtains $\Psi_{\Delta+2}$. By doing the calculations, which we omit, we get that $r_{yes,\Delta+2} \leq r_{no,\Delta+2} = O(\log\log n)$, and that also $d_{\Delta+2} = O(\log\log n)$.

With a final application of lemma 3 we obtain an equation-system $\Psi_{\Delta+2\frac{1}{2}}$ with the same degrees $r_{yes,\Delta+2\frac{1}{2}} = r_{yes,\Delta+2}$, and $r_{no,\Delta+2\frac{1}{2}} = r_{no,\Delta+2}$ but with $d_{\Delta+2\frac{1}{2}} = O(1)$. We are now ready to apply lemma 5. We thus obtain an equation-system Ψ_3, $\Psi_{\Delta+2\frac{1}{2}} \Rightarrow \Psi_{\Delta+3}$, where

$$\Psi_{\Delta+3} \in \mathbf{EQ}^2_{\epsilon^*,O(\Delta)}(\phi)$$

for $\epsilon^* = \epsilon_0 + (\Delta+3)\cdot\alpha + \frac{1}{|\mathcal{F}|} = |\mathcal{F}|^{-c^*}$ for some $0 < c^* < 1$. ■

3.1 Cube Representation

In this subsection we show a general algorithm that, given a system with a constant depend and a gap under $[r_{yes}, r_{no}, d]$-LDF-assumptions, generates a system with LDF-assumptions of the same degree, but of *constant* dimension. The generated system will maintain the gap of the original system. The depend parameter will not be increased by much.

Lemma 3 (Cube-Representation) *Fix a constant* $0 < c < 1$*, and let* $r_{yes} \leq |\mathcal{F}|^c$*, and* $r_{no} = \omega \cdot r_{yes}$ *where* $1 \leq \omega < (\log n)^3$*. Also assume* $\epsilon = |\mathcal{F}|^{-c_2}$ *for some* $0 < c_2 < 1$ *,* $d \leq \log_{|\mathcal{F}|} n$ *and* $D = O(1)$*. There exists* $\alpha = |\mathcal{F}|^{-c_3}$*, for some constant* $0 < c_3 < 1$*, such that*

$$\mathbf{EQ}^2_{\epsilon,D}(\{\mathcal{L}[r_{yes}, r_{no}, d]\})$$

$$\Downarrow$$

$$\mathbf{EQ}^2_{\epsilon+\alpha,3D}(\{\mathcal{L}[r_{yes}, r_{no}, D+3]\})$$

Moreover, there exists a reduction algorithm between the above sets with the following property. If K bounds, for an input system Ψ, the number of different assumption-domains that have variables in a single conjunction of Ψ, then

- *The depend parameter of the output system is only $D+2K$.*
- *The bound, K, is maintained in the output system.*

We begin with an equation-system $\Psi \in \mathbf{EQ}^2_{\epsilon,D}(\{\mathcal{L}[r_{yes}, r_{no}, d]\})$. Ψ has a polynomial number of LDF-assumptions, each over a variable-set corresponding to $\mathcal{F}^d$ (these variable-sets are pairwise disjoint). We would like to substitute these d-dimensional LDF-assumptions with LDF-assumptions of constant dimension, namely $(D+3)$.

Suppose we substitute an assumption $\mathcal{L}$ over a variable-set $\mathcal{D}$, with many LDF-assumptions, one for each $(D+3)$-dimensional affine subspace of $\mathcal{D}$ (Since the variables of $\mathcal{D}$ correspond to points of $\mathcal{F}^d$, we regard $\mathcal{D}$ both as a variable-set and as a geometric domain). The intersection of these new assumptions is exactly equal to $\mathcal{L}$.

However, the $(D+3)$-dimensional affine subspaces of $\mathcal{D}$ ($(D+3)$-cubes for short) are far from being disjoint, and we cannot have LDF-assumptions over intersecting domains (see explanation following definition 6 in subsection 1). We overcome this problem by duplicating the variables: we add a *new* variable-set for each $(D+3)$-dimensional affine subspace of $\mathcal{D}$.

Then, for each $\psi \in \Psi$, we construct a set Ψ'_ψ of conjunctions that simulate ψ by replacing the original variables with duplicates, and then somehow verifying consistency between the duplicates and the original variables.

The Construction of Ψ'

We will describe the construction of an equation-system $\Psi' \in \mathbf{EQ}^2_{\epsilon+\alpha,D+2K}(\{\mathcal{L}[r_{yes}, r_{no}, D+3]\})$ from Ψ, such that $\Psi \Rightarrow \Psi'$.

Variables. Ψ' will have the variables of Ψ, and also additional variables. Let $\mathcal{D}$ be an assumption-domain in Ψ. For each $(D+3)$-cube $\mathcal{C} \subset \mathcal{D}$, we add a new set $\mathcal{C}[\cdot]$ of *cube-variables*, with a distinct duplicate $\mathcal{C}[x]$ for each variable $\mathcal{D}[x] \in \mathcal{C}$. Note that now each point in $\mathcal{D}$ has many variables representing it, one for each $(D+3)$-cube containing it.

Assumptions. Let $\mathcal{D}$ be an assumption-domain in Ψ. For each cube $\mathcal{C} \subset \mathcal{D}$, Ψ' will have an $[r_{yes}, r_{no}, D+3]$-assumption on $\mathcal{C}[\cdot]$. These assumptions *replace* the $[r_{yes}, r_{no}, d]$-assumption on $\mathcal{D}$. The assumption replacement can be summarized by

$$\mathcal{L}_{\mathcal{D}}[r_{yes}, r_{no}, d]$$

$$\downarrow$$

$$\{\mathcal{L}_{\mathcal{C}[\cdot]}[r_{yes}, r_{no}, D+3] \ : \ \mathcal{C} \text{ is a } (D+3)\text{-cube in } \mathcal{D}\}$$

The Conjunctions. For every $\psi \in \Psi$ we shall have a set of conjunctions Ψ'_ψ, and set

$$\Psi' \stackrel{def}{=} \bigcup_{\psi \in \Psi} \Psi'_\psi$$

The conjunctions in Ψ'_ψ will simulate ψ, only they will refer not to the original variables, but to cube-variable duplicates. In addition, the conjunctions in Ψ'_ψ will have consistency equations verifying the consistency of the cube-variables with the original variables.

Take a conjunction $\psi \in \Psi$. It depends on variables from K assumption-domains[2], $\mathcal{D}_1, \ldots, \mathcal{D}_K$, called the assumption domains of ψ. ψ has at most D variables from each $\mathcal{D}_i$, since the total depend is D. For each i, take $\mathbf{x}_i \subseteq \mathcal{D}_i$ to be an arbitrary D-cube that contains the variables of ψ in $\mathcal{D}_i$ - $\mathbf{x}_i$ is called the *base* of ψ in $\mathcal{D}_i$.

Definition 11 *Let $\mathcal{D} = \mathcal{F}^d$ and let $\mathbf{x} \subseteq \mathcal{D}$ be a subset. We denote by $\mathcal{S}_{\mathbf{x}}$ the set of $(D+3)$-cubes in $\mathcal{D}$ that contain $\mathbf{x}$.*

The set Ψ'_ψ is defined as follows. For every choice of points $y_i \in \mathcal{D}_i$ and cubes $\mathcal{C}_i \in \mathcal{S}_{\mathbf{x}_i \cup \{y_i\}}$, $i = 1, .., K$, Ψ'_ψ has a conjunction $\psi[y_1, \ldots, y_K, \mathcal{C}_1, \ldots, \mathcal{C}_K] \in \Psi'_\psi$ that consists of:

- The conjunction ψ, where every variable of the form $\mathcal{D}_i[x]$ is substituted by $\mathcal{C}_i[x]$ ($\mathcal{C}_i$ contains base of ψ in $\mathcal{D}_i$, therefore it contains the necessary variables).
- The equations $(\mathcal{C}_i[y_i] = \mathcal{D}_i[y_i])$, for $1 \leq i \leq K$. These are called the *consistency test equations.*

The variables of $\psi[y_1, \ldots, y_K, \mathcal{C}_1, \ldots, \mathcal{C}_K] \in \Psi'$, unlike those of ψ in Ψ, are not guaranteed to have values of global LDFs on the $\mathcal{D}_i$'s. However, we will show that the gap is maintained in Ψ', because in almost all the conjunctions of Ψ'_ψ, *if the consistency tests succeed*, the variables have values of *permissible* LDFs on the $\mathcal{D}_i$'s. We will elaborate on this when we prove that Ψ' maintains the gap.

Remarks About the Construction. Ψ' may contain new variable domains and LDF-assumptions, that no conjunction really has variables from. These domains may as well be discarded. We also note that every set Ψ'_ψ has the same number of conjunctions - this fact is used later.

The Parameters of Ψ'

The assumptions of Ψ' are $[r_{yes}, r_{no}, D+3]$-LDF assumptions. The depend of each conjunction equals the depend of ψ, plus the depend of the K consistency test equations. The depend is thus bounded above by $D+2K$.

We therefore have $\Psi' \in \mathbf{EQ}^2_{1,D+2K}(\{\mathcal{L}[r_{yes}, r_{no}, D+3]\})$. i.e. Ψ' has the properties claimed in the lemma, except for the gap property, which remains to be verified.

The Gap

Due to space limitations we only give a general description of the proof of the gap property.

[2] ψ may depend on variables from less than K assumption-domains, in which case we associate it with arbitrary domains.

In the yes case, where Ψ is completely satisfiable, it is easy to take a satisfying assignment for Ψ and extend it to a satisfying assignment of Ψ'. Ψ' has the variables of Ψ, which remain with the same assignment, and also new variables. In the construction, each new variable was associated with an old one, so it is natural to assign it with the value of the variable it represents. It is easy to verify that this extension of the assignment indeed yields a satisfying (legal) assignment for Ψ'.

In the no case, we show that each assignment of Ψ' can be (randomly) translated to an assignment of Ψ that satisfies almost the same fraction. In the proof we use the low-degree-test lemma of [RS97], to show that the LDF-assumptions on the cubes in Ψ', force the assignments to the domains to consist of so called *permissible* LDFs. The assignment of Ψ' is translated to an assignment for Ψ by assigning each domain with one of its permissible LDFs.

3.2 The Embedding Extension

In this section we state and prove lemma 4, showing how to replace $[r_{yes}, r_{no}, t]$-assumptions with considerably lower-degree assumptions, slightly increasing the dimension.

First we state a proposition that shows how an LDF over a constant dimensional domain $\mathcal{F}^t$ can be embedded as a much lower-degree LDF over a domain of larger dimension $\mathcal{F}^{tk}$. An LDF over $\mathcal{F}^t$, when embedded over $\mathcal{F}^{tk}$, yields an LDF of degree roughly the k'th root of the original degree.

The idea behind the representation is simple. If a variable x in a polynomial, is replaced by several variables which correspond to powers of x (say $y_1 = x$, $y_2 = x^3$, $y_3 = x^{27}$), then a polynomial of high degree in x such as $x^{28} + x^{10}$ can be represented by polynomial over more variables, but also of considerably lower degree $(y_1)(y_3) + (y_1)(y_2)^2$.

Proposition 6 *Let t be a constant and let k be some parameter. There exists an injective map $\mathcal{M} : \mathcal{F}^t \to \mathcal{F}^{tk}$ with the following properties:*

- *For any $[r,t]$-LDF $f : \mathcal{F}^t \to \mathcal{F}$, there is a $[btk, tk]$-LDF f_e on $\mathcal{F}^{tk}$ such that*

$$\forall x \in \mathcal{F}^t \ : \quad f(x) = f_e(\mathcal{M}(x))$$

  where $b \stackrel{def}{=} \sqrt[k]{r}$.

- *$\mathcal{M}$ is a polynomial manifold of degree $b^{k-1} = r_{yes}/b$ in $\mathcal{F}^{tk}$.*

  (Hence for any $[\widetilde{r}, tk]$-LDF f_e on $\mathcal{F}^{tk}$, its restriction to $\mathcal{M}$, $f_e \circ \mathcal{M} : \mathcal{F}^t \to \mathcal{F}$ is an $[\widetilde{r} \cdot b^{k-1}, t]$-LDF over $\mathcal{F}^t$.)

Proof: The embedding map $\mathcal{M}$ is defined as follows. For $\bar{x} = (x_1, \ldots, x_t) \in \mathcal{F}^t$, $\mathcal{M}(\bar{x})$ is the point $\bar{y} = (y_1, \ldots, y_t) \in \mathcal{F}^{tk}$, where for each i

$$y_i = (y_{i,0}, .., y_{i,k-1}) \stackrel{def}{=} ((x_i), (x_i)^b, (x_i)^{b^2}, \ldots, (x_i)^{b^{k-1}})$$

It is clear that $\mathcal{M}$ is an injective map and a manifold of degree b^{k-1}. This gives the second statement in the proposition. As for the first statement, consider an $[r,t]$-LDF f over $\mathcal{F}^t$. We construct the corresponding LDF f_e over $\mathcal{F}^{t \cdot k}$ by replacing the monomials of f, x_i^j, with $(y_{i,0})^{b_0}(y_{i,1})^{b_1} \ldots (y_{i,k-1})^{b_{k-1}}$ where $b_0 b_1 \ldots b_{k-1}$ is the base b representation of j, i.e. $j = \sum_{p=0}^{k-1} b_p \cdot b^p$.

It is easy to see that $\forall x,\ f(x) = f_e(\mathcal{M}(x))$ and evaluating f on any point $x \in \mathcal{F}^t$ can be done in polynomial time by evaluating f_e on $\mathcal{M}(x) \in \mathcal{F}^{tk}$. One may think of the embedding as adding redundancy in the representation of a point, in order to make the computation of the LDF 'easier'.

The dimension of f_e is the dimension of f times k, yet the degree in each variable is $< b$, hence the total degree is $b \cdot tk$. This completes the proof of the proposition. ∎

Lemma 4 (Embedding Extension) *Fix a constant t. For any $k = O(\log_{|\mathcal{F}|} n)$, if $\frac{r_{no}}{r_{yes}} \geq kt$ then*

$$\mathbf{EQ}^2_{\epsilon,D}(\{\mathcal{L}_{\mathcal{D}_1,..,\mathcal{D}_s}[r_{yes}, r_{no}, t]\})$$
$$\Downarrow$$
$$\mathbf{EQ}^2_{\epsilon,D}\left(\{\mathcal{L}_{\widetilde{\mathcal{D}}_1,..,\widetilde{\mathcal{D}}_s}[kt \cdot \sqrt[k]{r_{yes}},\ \frac{r_{no}}{r_{yes}} \cdot \sqrt[k]{r_{yes}},\ kt]\}\right)$$

Moreover, there is a linear reduction between the above, that maintains the bound on the number of different assumption-domains that appear in each conjunction.

Proof: Let $\Psi \in \mathbf{EQ}^2_{\epsilon,D}(\{\mathcal{L}_{\mathcal{D}_1,..,\mathcal{D}_s}[r_{yes}, r_{no}, t]\})$. We will show a general algorithm to construct $\widetilde{\Psi}$, such that $\widetilde{\Psi} \in \mathbf{EQ}^2_{\epsilon,D}\left(\{\mathcal{L}_{\widetilde{\mathcal{D}}_1,..,\widetilde{\mathcal{D}}_s}[kt \cdot \sqrt[k]{r_{yes}},\ \frac{r_{no}}{r_{yes}} \cdot \sqrt[k]{r_{yes}},\ kt]\}\right)$. Recall that the $\mathcal{D}_i$'s are pairwise disjoint, and each $\mathcal{D}_i$ is identified with a copy of $\mathcal{F}^t$ by the correspondence $x \to \mathcal{D}_i[x]$. For each i, we add a set of new variables $\widetilde{\mathcal{D}}_i$ that 'extends' $\mathcal{D}_i$ – $\widetilde{\mathcal{D}}_i$ is identified with an extended domain $\mathcal{F}^{tk}$. This accounts for the restriction $k = O(\log_{|\mathcal{F}|} n)$, since we want the size of $\widetilde{\Psi}$ to be polynomial in the size of Ψ.

Constructing $\widetilde{\Psi}$. We substitute the assumption $\mathcal{L}_{\mathcal{D}_i}[r_{yes}, r_{no}, t]$ for $\mathcal{L}_{\widetilde{\mathcal{D}}_i}[\widetilde{r_{yes}}, \widetilde{r_{no}}, kt]$, where

$$\widetilde{r_{no}} = \frac{r_{no}}{r_{yes}} \cdot b$$
$$\widetilde{r_{yes}} = kt \cdot b$$

and $b = \sqrt[k]{r_{yes}}$. Every occurrence of the variable $\mathcal{D}_i[x]$ in a conjunction is *replaced* by the new variable $\widetilde{\mathcal{D}}_i[\mathcal{M}(x)]$ where $\mathcal{M}$ is the map from proposition 6, hence $depend(\widetilde{\Psi}) = depend(\Psi)$.

The variables $\mathcal{D}_i$ no longer appear in any of the conjunctions and are discarded. Note that the variables from $\widetilde{\mathcal{D}}_i$ that appear in the conjunctions are only those on the manifold $\mathcal{M}(\mathcal{D}_i) \subset \widetilde{\mathcal{D}}_i$. The rest of the variables seem to have been artificially added, for the sake of the assumption. In fact, there will be many conjunctions that depend on these variables when we eliminate the LDF-assumptions (e.g. when the cube representation lemma is applied, and the new domains are broken into cubes).

Maintaining the Gap. Proving that $\widetilde{\Psi}$ maintains the gap of Ψ is both relatively easy. We only only give a description of it, due to space limitations.

In the yes case, we just use proposition 6 to extend a satisfying for Ψ, and get a satisfying assumption for $\widetilde{\Psi}$. In the no case, we use the fact that the embedding is a polynomial manifold, to translate an assignment for $\widetilde{\Psi}$ to an assignment for Ψ that satisfied the exact fraction of equations. This is done by restricting the assignments of the $\widetilde{\mathcal{D}}_i$'s, to the embedded $\mathcal{D}_i$'s.

■

4 Discussion

We would like to suggest the following conjecture, which can be viewed as the gap-QS parallel for the [BGLR93] conjecture:

Conjecture 3 $\exists c > 0$ *constant, such that gap-QS*$[O(1), \mathcal{F}]$ *is NP-hard for* $|\mathcal{F}| = \Theta(n^c)$.

As the BGLR conjecture states that proofs are verifiable with a polynomially small error by accessing a constant number of logarithmic sized variables, we claim that it is achievable even by tests formed as quadratic-equations.

References

[ABMP98] M. Alekhnovich, S. Buss, S. Moran, and T. Pitassi. Minimum propositional proof length is NP-hard to linearly approximate. Manuscript, 1998.

[ALM+92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *Proc. 33rd IEEE Symp. on Foundations of Computer Science*, pages 13–22, 1992.

[AS92] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. In *Proc. 33rd IEEE Symp. on Foundations of Computer Science*, pages 2–13, 1992.

[AS97] Sanjeev Arora and Madhu Sudan. Improved low degree testing and its applications. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 485–495, El Paso, Texas, 4–6 May 1997.

[BFL91] L. Babai, L. Fortnow, and C. Lund. Nondeterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

[BGLR93] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient multi-prover interactive proofs with applications to approximation problems. In *Proc. 25th ACM Symp. on Theory of Computing*, pages 113–131, 1993.

[DS98] I. Dinur and S. Safra. Monotone-minimum-satisfying assignment is NP-hard for almost polynomial factors. Manuscript, 1998.

[HPS93] J. Håstad, R. Phillips, and S. Safra. A well-characterized approximation problem. *Information Processing Letters*, 47:301–305, 1993.

[LY94] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.

[Raz98] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, June 1998.

[RS97] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proc. 29th ACM Symp. on Theory of Computing*, pages 475–484, 1997.