Electronic Colloquium on Computational Complexity, Report No. 26 (2007)

Sub-Constant Error Probabilistically Checkable Proof of Almost-Linear Size

Dana Moshkovitz * Ran Raz †

November 19, 2006

Abstract

We show a construction of a PCP with both sub-constant error and almost-linear size. Specifically, for some constant $0 < \alpha < 1$, we construct a PCP verifier for checking satisfiability of Boolean formulas that on input of size n uses $\log n + O((\log n)^{1-\alpha})$ random bits to query a constant number of places in a proof of size $n \cdot 2^{O((\log n)^{1-\alpha})}$ over symbols consisting of $O((\log n)^{1-\alpha})$ bits and achieves error $2^{-\Omega((\log n)^{\alpha})}$.

The construction is by a new *randomness-efficient version of the aggregation through curves technique* [1]. Its main ingredients are a recent *low degree test* with both sub-constant error and almost-linear size [12] and a new method for constructing a short list of *balanced curves*.

*dana.moshkovitz@weizmann.ac.il. Department of Computer Science and Applied Mathematics, The Weizmann Institute, Rehovot, Israel.

†ran.raz@weizmann.ac.il. Department of Computer Science and Applied Mathematics, The Weizmann Institute, Rehovot, Israel.

ISSN 1433-8092

1 Introduction

The notion of *proofs* is of fundamental interest to the theory of Computer Science. One defines $\mathcal{NP}$ to be the class of all languages L in which membership can be proved efficiently. That is, there is an algorithm (called a *verifier*) that given input x of size n, as well as a proof π of size polynomial in n for the statement "$x \in L$", runs in time polynomial in n and behaves as follows: (i) if indeed $x \in L$, then there exists a proof π for which the verifier accepts; while (ii) if $x \notin L$, then for any alleged proof π, the verifier rejects.

The PCP Theorem [2, 1] (PCP for Probabilistically Checkable Proofs) states that all languages L in $\mathcal{NP}$ have proofs of polynomial size that can be verified *probabilistically* by querying only a *constant* number of places in the proof. A PCP verifier, when given access to an input x and to a proof π, tosses r coins, uses them to pick only a *constant* number of queries q to π and satisfies the following: (i) if $x \in L$, then, as before, there exists a proof π that the verifier always accepts; while (ii) if $x \notin L$, then for any alleged proof π, the verifier accepts with probability at most ε. The term ε is called the *error* of the PCP.

The formal definition of a PCP verifier is as follows:

Definition 1 (PCP Verifier). *A PCP verifier for a language L with* size $s : \mathbb{N} \to \mathbb{N}$, randomness complexity $r : \mathbb{N} \to \mathbb{N}$, query complexity $q : \mathbb{N} \to \mathbb{N}$, answer size $a : \mathbb{N} \to \mathbb{N}$, perfect completeness *and* error $\varepsilon : \mathbb{N} \to (0, 1)$, *is a probabilistic polynomial time Turing machine, that given access to an input $x \in \{0,1\}^n$, as well as an oracle access to a proof π of size $s(n)$ containing symbols from $\{0,1\}^{a(n)}$, (i) tosses $r(n)$ random coins; (ii) picks $q(n)$ indices in π; (iii) queries each index to obtain the corresponding answer of length $a(n)$; (iv) decides whether to accept or reject; and satisfies the following properties:*

- ***Completeness:*** *if $x \in L$, then there exists a proof π, on which the verifier always accepts.*
- ***Soundness:*** *if $x \notin L$, then given access to any proof π, the verifier accepts with probability at most $\varepsilon(n)$.*

Other than being a surprising result about the power of proofs, the PCP Theorem has a tight connection to hardness of approximation. This connection enabled a vast body of results showing that approximating many optimization problems to within various factors is $\mathcal{NP}$-hard. The PCP Theorem and the techniques used for its proof also motivated and inspired many influential notions such as *local testing* and *local decoding*.

Given the importance of the PCP Theorem, there has been a long line of research trying to improve it.

1.1 Decreasing The Error

The original PCP Theorem [2, 1] gave error $\varepsilon = \frac{1}{2}$. A natural goal is to reduce the error, preferably to sub-constant error $o(1)$.

However, the error of a verifier that makes q queries to the proof, where each symbol of the proof is a single bit, is at least 2^{-q}. Thus, to allow sub-constant error, we consider proofs in which each symbol consists of a bits, where a is non-constant. For applications (e.g., to hardness of approximation [4]), many times even a logarithmic answer size a (corresponding to a polynomial sized alphabet) is permissible, while it is desired to keep the number of queries q constant.

This path of research is taken in several works [4, 14, 3, 9]. The state of the art is by [9], who construct, for *any* constant $0 < \alpha < 1$, a PCP verifier that on input of size n uses $O(\log n)$ random bits to query a constant number of places in a proof of size polynomial in n over symbols consisting of at most $O((\log n)^{\alpha})$ bits and achieves sub-constant error $2^{-\Omega((\log n)^{\alpha})}$.

1.2 Decreasing The Size

The original PCP Theorem [2, 1] gave proofs of size polynomial in the input size, n^c, for a large constant c. A natural goal is to reduce the size, preferably to almost-linear in the input size $n^{1+o(1)}$.

This path of research is taken in works such as [13, 11, 7, 6, 5, 8]. The state of the art is by Dinur [8] who constructs a PCP verifier for checking satisfiability of Boolean formulas of size n with almost-linear size $n \cdot \operatorname{poly} \log n$, randomness $\log n + O(\log \log n)$, constant number of queries, constant answer size and constant error.

1.3 Sub-Constant Error and Almost-Linear Size

The question of whether there exist PCPs with both sub-constant error and almost-linear size was open. A step in this direction is taken in [12], where a low degree test of sub-constant error and almost-linear size is constructed. Low degree tests are an important ingredient in most PCP constructions, and it seemed that the construction of [12] should also yield PCPs with sub-constant error and almost-linear size. This work confirms this conjecture. Our main theorem is as follows.

Theorem 2 (Main). *There exists a constant $0 < \alpha < 1$, for which there is a PCP verifier for checking satisfiability of Boolean formulas that on input of size n uses $\log n + O((\log n)^{1-\alpha})$ random bits to query 7 places in a proof of size $n \cdot 2^{O((\log n)^{1-\alpha})}$ over symbols consisting of $O((\log n)^{1-\alpha})$ bits. The verifier has perfect completeness and error $2^{-\Omega((\log n)^{\alpha})}$.*

We note that the error and the size achieved by our construction are not as good as the best error and size known for each parameter separately. We achieve size $n \cdot 2^{O((\log n)^{1-\alpha})}$ for a small constant α, which is indeed almost-linear $n^{1+o(1)}$, however, the size of Dinur's construction [8] is only $n \cdot 2^{O(\log \log n)}$. We achieve error $2^{-\Omega((\log n)^{\alpha})}$ for *some* small constant $0 < \alpha < 1$, which is indeed sub-constant $o(1)$, however, the error of the construction by [9] can be made as low as $2^{-\Omega((\log n)^{\alpha})}$ for *any* constant $0 < \alpha < 1$. The question of whether one can optimize the two parameters, size and error, simultaneously remains open.

1.4 Overview

Our construction consists of three conceptual steps:

1. **Amplification:** this step takes a PCP verifier that has constant error and produces a PCP verifier that has sub-constant error $2^{-\Omega((\log n)^{\alpha})}$, but makes a non-constant number of queries $O((\log n)^{\alpha})$ to the proof.

2. **Aggregation:** this step takes a PCP verifier that makes a non-constant number of queries $O((\log n)^{\alpha})$ to the proof and produces a PCP verifier that makes only a constant number of queries, but to a proof with large answer size $2^{(\log n)^{O(\alpha)}}$.

3. **Reduction of answer size:** this step reduces the answer size of the PCP verifier constructed in the previous step to $O((\log n)^{1-\alpha})$.

The input to the amplification step is a PCP verifier with constant error that has low randomness complexity and almost-linear size, i.e., it uses only $(1 + o(1)) \log n$ random bits to query a proof of size $n^{1+o(1)}$. Any one of a number of existing constructions can be used for this purpose. We use the construction of Dinur [8].

The important point is that we implement the three steps in a way that *maintains low randomness complexity and small size*.

The amplification step is standard and uses random walks on expanders. The aggregation step is the main new step and we discuss it below. For the reduction of answer size we use ideas from [14, 9] for testing and reading of low degree polynomials. Notably, differing from previous constructions, this step is done without recursive composition of PCP verifiers.

Aggregation. The aggregation step is the main new step of the construction, and it is done via a new randomness-efficient version of the aggregation through curves technique of [1].

This technique assumes a PCP verifier V that makes q queries to a proof of size s, and constructs a new PCP verifier V' making a constant number of queries to a somewhat larger proof with a large answer size. Let us give a short description of the basic technique and the difficulty in using it in a way that maintains low randomness complexity and almost-linear size.

For sufficiently large finite field $\mathbb{F}$ and dimension m, a proof for V can be encoded by a low degree polynomial in $\mathbb{F}^m$ (its so-called *low degree extension*). The indices $\{1, \ldots, s\}$ are identified with points in $\mathbb{F}^m$ and the evaluation of the polynomial on these points gives the value of the proof in the corresponding positions.

A valid proof π for V' contains the evaluation of a low degree extension of a proof for V on all points in $\mathbb{F}^m$, as well as its restriction to certain subspaces and curves in $\mathbb{F}^m$. The verifier V' simulates V using a constant number of queries to the proof as follows:

1. V' tosses coins for V. Suppose $\vec{x}_1, \ldots, \vec{x}_q \in \mathbb{F}^m$ are the q points corresponding to the proof positions that V queries upon the outcome of the coin tosses.

2. V' passes a curve c of degree q through $\vec{x}_1, \ldots, \vec{x}_q$ and through a point $\vec{x}_{q+1}$ chosen uniformly at random from $\mathbb{F}^m$. V' queries the proof for the restriction $\pi(c)$ of the low degree extension to the curve c. This is a polynomial of degree at most q times the degree of the low degree extension.

3. V' chooses a uniformly random point $\vec{x}$ on the curve c among those that were not set to be $\vec{x}_1, \ldots, \vec{x}_q$. V' uses low degree testing to check that the evaluation of $\pi(c)$ on $\vec{x}$ is consistent with the low degree extension.

   [Note that if $\pi(c)$ is consistent with a low degree polynomial over $\mathbb{F}^m$ on many points on c, then it is consistent with the polynomial on *all* points on c, and, in particular, on $\vec{x}_1, \ldots, \vec{x}_q$.]

4. V' uses the evaluation of $\pi(c)$ on $\vec{x}_1, \ldots, \vec{x}_q$ to simulate V.

The verifier V' makes only a constant number of queries: one query to $\pi(c)$ and a constant number of queries required for low degree testing. Those queries are made to a proof with a large answer size: containing restrictions to subspaces and curves.

V' is not randomness-efficient, since, in addition to the coin tosses for V, it chooses another independent uniformly distributed point $\vec{x}_{q+1} \in \mathbb{F}^m$ to pass a curve through. The resulting proof size is at least quadratic, since per fixing of the coin tosses for V, it contains entries for $|\mathbb{F}^m|$ curves. [recall that $|\mathbb{F}^m| \geq s$].

The reason that the additional uniformly distributed point $\vec{x}_{q+1} \in \mathbb{F}^m$ is required is that the low degree testing works well only *on average*. To have low error probability, the verifier should apply low degree testing on a point $\vec{x}$ that is uniformly distributed in $\mathbb{F}^m$ (or close to such). When $\vec{x}_{q+1}$ is uniformly distributed, it indeed holds that all the points on the curve, but those set to be $\vec{x}_1, \ldots, \vec{x}_q$, are uniformly distributed in $\mathbb{F}^m$. The challenge is to have $\vec{x}$ uniformly distributed in $\mathbb{F}^m$ (or close to such) without using an additional independent uniformly distributed point.

Balancing Curves. We provide an efficient deterministic algorithm for constructing a short list of curves through the (arbitrary) query points of V, such that the distribution of a random point on *a curve chosen uniformly at random from the list* is ε-close (in the l_1-norm) to uniform over $\mathbb{F}^m$. For approximation parameter $\varepsilon > 0$, assuming there are N tuples of query points, the algorithm produces a list of size at most $O(N + \frac{|\mathbb{F}^m|}{\varepsilon^2})$. This enables randomness-efficient verification and almost-linear proof size.

The algorithm iteratively goes over all the tuples of query points. For each tuple, it chooses in a greedy manner curves through this tuple. For this purpose, it maintains, for each point $\vec{x} \in \mathbb{F}^m$, the number $d(\vec{x})$ of times that curves constructed thus far hit $\vec{x}$. The algorithm is as follows:

1. For every $\vec{x} \in \mathbb{F}^m$, set $d(\vec{x}) = 0$
2. For each tuple of query points $\vec{x}_1, \ldots, \vec{x}_q$, do the following $O(\eta/\varepsilon^2)$ times, where η denotes the ratio $|\mathbb{F}^m| / N$:
   - Go over the curves through $\vec{x}_1, \ldots, \vec{x}_q$ and through an additional point $\vec{x}_{q+1} \in \mathbb{F}^m$ ($|\mathbb{F}^m|$ curves). Choose a curve c that minimizes the sum $\sum_{\vec{x} \in C} d(\vec{x})$, where C is the multi-set of all points $\vec{x}$ on c, but those set to one of $\vec{x}_1, \ldots, \vec{x}_q$.
   - For every point $\vec{x}$ on c, increase $d(\vec{x})$ by the multiplicity of $\vec{x}$ in the multi-set C.

Note that the algorithm runs in time polynomial in N, $|\mathbb{F}^m|$ and $\frac{1}{\varepsilon}$.

1.5 Organization

The first sections provide the means for reduction of answer size and serve as an introduction to low degree testing and reading [in our terminology, *low degree reading* is the process of obtaining evaluations consistent with a low degree polynomial on arbitrary points $\vec{x}_1, \ldots, \vec{x}_q$].

The low degree tester of [12] and its adaptation to our setting are described in section 3. A low degree reader (i.e., an algorithm for low degree reading) using standard (not randomness-efficient) aggregation through curves is required as a procedure and described in section 4. An adaptation of it using an idea from [9] is presented in section 5. This machinery allows us to construct a sub-constant error low degree tester of almost-linear size *with small answer size* in section 6.

The aggregation step is enabled by a low degree reader presented in section 7. This low degree reader is implemented in a randomness-efficient manner, along the lines of the above description, and uses the techniques developed in the previous sections to reduce answer size. Section 7 also contains the algorithm for constructing balanced curves and its analysis.

The amplification step and finally the PCP verifier promised in our main theorem are presented in section 8.

2 Preliminaries

2.1 Polynomials

An *m-variate polynomial* over a field $\mathbb{F}$ is a function $Q : \mathbb{F}^m \to \mathbb{F}$ of the form

$$Q(x_1, \ldots, x_m) = \sum_{i_1,\ldots,i_m} a_{i_1,\ldots,i_m} x_1^{i_1} \cdots x_m^{i_m}$$

where all the *coefficients* $a_{i_1,\ldots,i_m}$ are in $\mathbb{F}$. The expressions $a_{i_1,\ldots,i_m} x_1^{i_1} \cdots x_m^{i_m}$ are called the *monomials* of the polynomial.

The *degree* of Q is $\deg Q \stackrel{def}{=} \max\left\{\sum_{j=1}^m i_j \mid a_{i_1,\ldots,i_m} \neq 0\right\}$, where the degree of the *identically zero* polynomial is defined to be 0.

Proposition 2.1. *Fix a field $\mathbb{F}$ and a dimension m. Let $Q_1, Q_2 : \mathbb{F}^m \to \mathbb{F}$ be two polynomials. Then, $Q_1 + Q_2$ and $Q_1 \cdot Q_2$ are m-variate polynomials over $\mathbb{F}$, such that*

1. $\deg(Q_1 + Q_2) \leq \max\{\deg Q_1, \deg Q_2\}$.
2. $\deg(Q_1 \cdot Q_2) \leq \deg Q_1 + \deg Q_2$.

[and thus for polynomials $Q_1, \ldots, Q_k : \mathbb{F}^m \to \mathbb{F}$, we have $\deg(\sum_{i=1}^k Q_i) \leq \max\{\deg Q_1, \ldots, \deg Q_k\}$ and $\deg(\prod_{i=1}^k Q_i) \leq \sum_{i=1}^k \deg Q_i$]

For a dimension m, a degree d and a field $\mathbb{F}$, we let $\mathcal{P}_{m,d,\mathbb{F}}$ denote the family of all m-variate polynomials of degree at most d over $\mathbb{F}$.

The Schwartz-Zippel lemma shows that different low degree polynomials differ on most points,

Proposition 2.2 (Schwartz-Zippel). *Fix a finite field $\mathbb{F}$, a dimension m and a degree d. For two different polynomials $Q_1, Q_2 : \mathbb{F}^m \to \mathbb{F}$ of degree at most d,*

$$\Pr_{\vec{x} \in \mathbb{F}^m}[Q_1(\vec{x}) = Q_2(\vec{x})] \leq \frac{d}{|\mathbb{F}|}$$

The Schwartz-Zippel lemma immediately implies a list decoding property,

Proposition 2.3 (list decoding). *Fix a finite field $\mathbb{F}$ and a dimension m. Let $f : \mathbb{F}^m \to \mathbb{F}$ be some function and consider some degree $d \leq |\mathbb{F}|$. Then, for any $\delta \geq 2\sqrt{\frac{d}{|\mathbb{F}|}}$, if $Q_1, \ldots, Q_l : \mathbb{F}^m \to \mathbb{F}$ are* different *polynomials of degree at most d, and for every $1 \leq i \leq l$, the polynomial Q_i agrees with f on at least δ fraction of the points, i.e., $\Pr_{\vec{x} \in \mathbb{F}^m}[Q_i(\vec{x}) = f(\vec{x})] \geq \delta$, then $l \leq \frac{2}{\delta}$.*

Proof. Let $\delta \geq 2\sqrt{\frac{d}{|\mathbb{F}|}}$, and assume on way of contradiction that there exist $l = \lfloor \frac{2}{\delta} \rfloor + 1$ different polynomials $Q_1, \ldots, Q_l : \mathbb{F}^m \to \mathbb{F}$ as stated.

For every $1 \leq i \leq l$, let $A_i \stackrel{def}{=} \{\vec{x} \in \mathbb{F}^m \mid Q_i(\vec{x}) = f(\vec{x})\}$. By inclusion-exclusion,

$$|\mathbb{F}^m| \geq \left|\bigcup_{i=1}^l A_i\right| \geq \sum_{i=1}^l |A_i| - \sum_{i \neq j} |A_i \cap A_j|$$

By Schwartz-Zippel, for every $1 \leq i \neq j \leq l$, $|A_i \cap A_j| \leq \frac{d}{|\mathbb{F}|} \cdot |\mathbb{F}^m|$. Therefore, by the premise,

$$|\mathbb{F}^m| \geq l\delta \, |\mathbb{F}^m| - \binom{l}{2} \frac{d}{|\mathbb{F}|} \, |\mathbb{F}^m|$$

On one hand, since $l > \frac{2}{\delta}$, we get $l\delta > 2$. On the other hand, since $\frac{2}{\delta} \leq \sqrt{\frac{|\mathbb{F}|}{d}}$ and $d \leq |\mathbb{F}|$, we get $\binom{l}{2} \leq \frac{|\mathbb{F}|}{d}$. This results in a contradiction. ■

Interpolation. Fix a finite field $\mathbb{F}$. For univariate polynomials over $\mathbb{F}$, for every fixing of values to k points, there exists a polynomial of degree at most $k-1$ that agrees with this fixing (and this polynomial is unique by Schwartz-Zippel). Let us develop some machinery to argue that.

Proposition 2.4. *For a finite subset $T \subseteq \mathbb{F}$ and $t_0 \in T$, let $I_{T,t_0} : \mathbb{F} \to \mathbb{F}$ be as follows: for every $x \in \mathbb{F}$,*

$$I_{T,t_0}(x) = \frac{\prod_{t \in T - \{t_0\}} (x - t)}{\prod_{t \in T - \{t_0\}} (t_0 - t)}$$

Then, I_{T,t_0} is a univariate polynomial of degree at most $|T|-1$ over $\mathbb{F}$, such that (i) $I_{T,t_0}(t_0) = 1$; (ii) for every $t_1 \in T - \{t_0\}$, it holds that $I_{T,t_0}(t_1) = 0$; (iii) for every $x \in \mathbb{F} - T$, it holds that $I_{T,t_0}(x) \neq 0$.

Proof. First note that I_{T,t_0} is of degree at most $|T| - 1$, since $t_0 \in T$, and by proposition 2.1.

1. $I_{T,t_0}(t_0) = \frac{\prod_{t \in T - \{t_0\}} (t_0 - t)}{\prod_{t \in T - \{t_0\}} (t_0 - t)} = 1$.
2. Let $t_1 \in T - \{t_0\}$. Then $\prod_{t \in T - \{t_0\}} (t_1 - t) = (t_1 - t_1) \cdot \prod_{t \in T - \{t_0, t_1\}} (t_1 - t) = 0$. So, $I_{T,t_0}(t_1) = 0$.
3. If $I_{T,t_0}(x) = 0$, then $\prod_{t \in T - \{t_0\}} (x - t) = 0$. Thus, for some $t \in T - \{t_0\}$, it holds that $(x - t) = 0$. This does not hold if $x \notin T$.

■

Proposition 2.5 (univariate interpolation). *For different scalars $t_1, \ldots, t_k \in \mathbb{F}$, as well as $x_1, \ldots, x_k \in \mathbb{F}$, let $P_{t_1,\ldots,t_k,x_1,\ldots,x_k} : \mathbb{F} \to \mathbb{F}$ be as follows: for every $t \in \mathbb{F}$,*

$$P_{t_1,\ldots,t_k,x_1,\ldots,x_k}(t) = \sum_{i=1}^{k} I_{\{t_1,\ldots,t_k\},t_i}(t) \cdot x_i$$

Then, $P_{t_1,\ldots,t_k,x_1,\ldots,x_k}$ is a univariate polynomial of degree at most $k-1$ over $\mathbb{F}$, such that for every $1 \leq i \leq k$, it holds that $P_{t_1,\ldots,t_k,x_1,\ldots,x_k}(t_i) = x_i$.

Proof. Let $1 \leq i_0 \leq k$. Then, by proposition 2.4, (i)+(ii),

$$P_{t_1,\ldots,t_k,x_1,\ldots,x_k}(t_{i_0}) = \sum_{i=1}^{k} I_{\{t_1,\ldots,t_k\},t_i}(t_{i_0}) \cdot x_i = I_{\{t_1,\ldots,t_k\},t_{i_0}}(t_{i_0}) \cdot x_{i_0} + \sum_{i \in \{1,\ldots,k\} - \{i_0\}} I_{\{t_1,\ldots,t_k\},t_i}(t_{i_0}) \cdot x_i = x_{i_0}$$

■

Lemma 2.6 (multivariate interpolation). *Let m be a dimension. Consider a finite subset $H \subseteq \mathbb{F}$. Then, any function $f : H^m \to \mathbb{F}$ can be extended into a polynomial $Q_f : \mathbb{F}^m \to \mathbb{F}$ of degree at most $m(|H|-1)$ in a way that for every $\vec{x} \in H^m$, it holds that $Q_f(\vec{x}) = f(\vec{x})$.*

Proof. We use the notation from proposition 2.4 used for univariate interpolation. For every $(x_1, \ldots, x_m) \in \mathbb{F}^m$, let

$$Q_f(x_1, \ldots, x_m) = \sum_{h_1, \ldots, h_m \in H} I_{H,h_1}(x_1) \cdots I_{H,h_m}(x_m) f(h_1, \ldots, h_m)$$

Then $Q_f : \mathbb{F}^m \to \mathbb{F}$ is a polynomial of degree at most $m(|H|-1)$, since, by proposition 2.4, for every $1 \leq i \leq m$, the polynomial $I_{H,h_i}(x_i)$ is of degree at most $|H|-1$.

Moreover, for every $(h'_1, \ldots, h'_m) \in H^m$, it holds that

$$Q_f(h'_1, \ldots, h'_m) = \sum_{h_1, \ldots, h_m \in H} I_{H,h_1}(h'_1) \cdots I_{H,h_m}(h'_m) f(h_1, \ldots, h_m) = f(h'_1, \ldots, h'_m)$$

since for every $h_1, \ldots, h_m \in H$ such that $(h_1, \ldots, h_m) \neq (h'_1, \ldots, h'_m)$, there exists $1 \leq i \leq m$ such that $h_i \neq h'_i$, and thus, by proposition 2.4, $I_{H,h_i}(h'_i) = 0$, while $I_{H,h'_1}(h'_1) \cdots I_{H,h'_m}(h'_m) = 1$. ■

2.2 Curves

Fix a finite field $\mathbb{F}$ and a dimension m.

Definition 3 (curve). *A* curve *in $\mathbb{F}^m$ is a function $c : \mathbb{F} \to \mathbb{F}^m$, such that there exist univariate polynomials $c_1, \ldots, c_m : \mathbb{F} \to \mathbb{F}$, for which, for every $t \in \mathbb{F}$, $c(t) = (c_1(t), \ldots, c_m(t))$. The* degree *of the curve is the maximal degree of the polynomials:* $\deg c \stackrel{def}{=} \max\{\deg c_i \mid 1 \leq i \leq m\}$.

We let $\mathcal{C}_k^{m,\mathbb{F}}$ denote the family of all curves of degree at most k in $\mathbb{F}^m$.

The restriction of a polynomial $Q : \mathbb{F}^m \to \mathbb{F}$ to a curve $c : \mathbb{F} \to \mathbb{F}^m$ is $Q_{|c} : \mathbb{F} \to \mathbb{F}$ which is defined, for every $t \in \mathbb{F}$, by $Q_{|c}(t) = Q(c(t))$.

Proposition 2.7 (polynomial restricted to curve). *Fix a field $\mathbb{F}$ and a dimension m. For any polynomial $Q : \mathbb{F}^m \to \mathbb{F}$ and any curve $c : \mathbb{F} \to \mathbb{F}^m$, the restriction $Q_{|c}$ is a univariate polynomial of degree at most $\deg c \cdot \deg Q$.*

Proof. Denote $Q(x_1, \ldots, x_m) = \sum_{i_1, \ldots, i_m} a_{i_1, \ldots, i_m} x_1^{i_1} \cdots x_m^{i_m}$. Let $c_1, \ldots, c_m : \mathbb{F} \to \mathbb{F}$ be polynomials such that for every $t \in \mathbb{F}$, it holds that $c(t) = (c_1(t), \ldots, c_m(t))$. Then, for every $t \in \mathbb{F}$, we have $Q_{|c}(t) = Q(c(t)) = Q(c_1(t), \ldots, c_m(t)) = \sum_{i_1, \ldots, i_m} a_{i_1, \ldots, i_m} (c_1(t))^{i_1} \cdots (c_m(t))^{i_m}$. By proposition 2.1, for every $1 \leq l \leq m$, the degree of $(c_l(t))^{i_l}$ is at most $\deg c_l \cdot i_l \leq \deg c \cdot i_l$, and the degree of $(c_1(t))^{i_1} \cdots (c_m(t))^{i_m}$ is at most $\sum_{l=1}^m \deg c \cdot i_l \leq \deg c \cdot \sum_{l=1}^m i_l \leq \deg c \cdot \deg Q$. Hence, by proposition 2.1, the degree of $Q_{|c}$ is at most $\deg c \cdot \deg Q$. ■

Interpolation. For different scalars $t_1, \ldots, t_k \in \mathbb{F}$ and (not necessarily different) points $\vec{x}_1, \ldots, \vec{x}_k \in \mathbb{F}^m$, we will define a curve $c_{t_1, \ldots, t_k, \vec{x}_1, \ldots, \vec{x}_k} : \mathbb{F} \to \mathbb{F}^m$, such that for every $1 \leq i \leq k$, the curve evaluates to $\vec{x}_i$ at t_i, i.e., $c_{t_1, \ldots, t_k, \vec{x}_1, \ldots, \vec{x}_k}(t_i) = \vec{x}_i$.

For every $1 \leq i \leq k$, denote $\vec{x}_i = (x_{i,1}, \ldots, x_{i,m})$.

Proposition 2.8 (curve interpolation). *Let* $c_{t_1,\ldots,t_k,\vec{x}_1,\ldots,\vec{x}_k} : \mathbb{F} \to \mathbb{F}^m$ *be such that for every* $t \in \mathbb{F}$*,*

$$c_{t_1,\ldots,t_k,\vec{x}_1,\ldots,\vec{x}_k}(t) = \left(P_{t_1,\ldots,t_k,x_{1,1},\ldots,x_{k,1}}(t), \ldots, P_{t_1,\ldots,t_k,x_{1,m},\ldots,x_{k,m}}(t)\right)$$

[Recall that the definition of $P_{t_1,\ldots,t_k,x_{1,l},\ldots,x_{k,l}}$ *appeared in proposition 2.5] Then,*

1. $c_{t_1,\ldots,t_k,\vec{x}_1,\ldots,\vec{x}_k}$ *is a curve of degree at most* $k-1$*.*
2. *For every* $1 \leq i \leq k$*,* $c_{t_1,\ldots,t_k,\vec{x}_1,\ldots,\vec{x}_k}(t_i) = \vec{x}_i$*.*

Proof. Item 1 follows since for every $1 \leq l \leq m$, $P_{t_1,\ldots,t_k,x_{1,l},\ldots,x_{k,l}}$ is of degree at most $k-1$ by proposition 2.5. Item 2 follows since for every $1 \leq i \leq k$, for every $1 \leq l \leq m$, we have, again from proposition 2.5, $P_{t_1,\ldots,t_k,x_{1,l},\ldots,x_{k,l}}(t_i) = x_{i,l}$. ∎

3 Randomness-Efficient Low Degree Tester With Large Answer Size

In this section we describe a low degree tester based on the sub-constant error low degree test of almost-linear size from [12]. The tester is essentially the same as in [12], only we formulate it in a way more convenient for us and prove a slightly different soundness guarantee.

The purpose of the tester is to verify that a function $f : \mathbb{F}^m \to \mathbb{F}$ evaluates to (one of few) low degree polynomials, by making only a constant number of queries to f and to an additional proof. The tester gets as input a point $\vec{x} \in \mathbb{F}^m$, makes some probabilistic test in which it queries f and the additional proof, and decides whether to accept or reject. If f is indeed a low degree polynomial, the tester, when provided an adequate proof, should accept with probability 1 any input. Moreover, no matter which function f and proof are provided, for *almost all* points $\vec{x} \in \mathbb{F}^m$, *with high probability* over the randomness of the tester, whenever the tester accepts, it is guaranteed that $f(\vec{x})$ is the evaluation on $\vec{x}$ of one of few low degree polynomials associated with f.

How is the testing done? If f is of low degree, then its restriction to any affine subspace in $\mathbb{F}^m$ is a polynomial of low degree. Thus, as a proof, low degree tests usually ask for low degree polynomials that are supposedly the restrictions of f to some family of low dimensional affine subspaces. Then, a subspace s that contains $\vec{x}$ is picked from the family in some randomized manner, and $f(\vec{x})$ is compared against the evaluation of the polynomial assigned to s at $\vec{x}$.

The family of affine subspaces used in [12] is the family of all three-dimensional linear subspaces in $\mathbb{F}^m$ that are spanned by a vector over $\mathbb{F}$ and two vectors over a subfield $\mathbb{K}$ of $\mathbb{F}$. It is shown there that it is enough to take the subfield to be of small size: $|\mathbb{K}| = \text{poly}(m)$, resulting in a family (and a proof) of size lower than $|\mathbb{F}|^m \cdot |\mathbb{K}|^{2m} = |\mathbb{F}^m| \cdot m^{O(m)}$. For the field $\mathbb{F}$ and dimension m we use, this is indeed a small family.

The answer size of the proof is the number of bits required to describe a 3-variate low degree polynomial over $\mathbb{F}$. For the degree parameter we use, this answer size will be too large. Subsequent sections will allow us to reduce it.

Remark 3.1. *It will be more convenient for us [here and throughout the paper] to address the randomness complexity of the algorithm, rather than its proof size. Note that effectively the size of the proof needed for an algorithm that uses* r *random bits to query* q *positions of its proof is at most* $q \cdot 2^r$*. Thus, for algorithms making a constant number of queries to their proof (like all algorithms presented here), the size is bounded (up to a constant factor) by* 2^r*.*

Canonical Representations. We would like to use canonical representations for the three dimensional linear subspaces, so the restriction of a function $f : \mathbb{F}^m \to \mathbb{F}$ to a subspace s will be defined uniquely as a function $f_{|s} : \mathbb{F}^3 \to \mathbb{F}$. For this purpose, let us recall a few facts from linear algebra.

For a matrix M over a field $\mathbb{F}$, the *row space* of M is the linear subspace spanned by its rows. The rank of M is the dimension of its row space. Two matrices of the same dimensions over a field $\mathbb{F}$ are said to be *row-equivalent*, if they have the same row space.

Definition 4 (row canonical form). *We will say that a matrix over a field is in* row canonical form*, if it satisfies the following requirements:*

- *All nonzero rows are above any rows of all zeros.*
- *The leading coefficient of a nonzero row is always to the right of the leading coefficient of the row above it (if such row exists).*
- *The leading coefficients are the only nonzero entries in their column.*
- *All leading coefficients are* 1*.*

Fact 3.2. *The rank of a matrix in row canonical form is exactly the number of nonzero rows in it.*

Fact 3.3. *Every matrix over a field has a* unique *row-equivalent matrix in row canonical form. Moreover, the Gaussian elimination algorithm finds it using a polynomial (in the dimensions of the matrix) number of field operations.*

For dimensions k, m and field $\mathbb{F}$, let $\mathcal{M}_{k,m,\mathbb{F}}$ be the family of all $k \times m$ matrices over $\mathbb{F}$ that are in row canonical form.

Since every linear subspace $s \subseteq \mathbb{F}^m$ of dimension at most 3 is the row space of some $3 \times m$ matrix over $\mathbb{F}$, by the above discussion, for every linear subspace $s \subseteq \mathbb{F}^m$ of dimension at most 3, there is a unique matrix in $\mathcal{M}_{3,m,\mathbb{F}}$ whose row space is s. We denote this matrix by M_s and use it to represent s.

This allows us to define the restriction of a function $f : \mathbb{F}^m \to \mathbb{F}$ to s *uniquely* to be $f_{|s} : \mathbb{F}^3 \to \mathbb{F}$, where for every $\vec{t} \in \mathbb{F}^3$, $f_{|s}(\vec{t}) = f(M_s^T \vec{t})$.

Note that given vectors $\vec{v}_1, \vec{v}_2, \vec{v}_3 \in \mathbb{F}^m$ that span s, one can obtain the matrix M_s by applying Gaussian elimination on a matrix whose rows are $\vec{v}_1, \vec{v}_2, \vec{v}_3$. Moreover, for every $\vec{x} \in s$, one can find a linear combination $\vec{t} \in \mathbb{F}^3$ for which $M_s^T \vec{t} = \vec{x}$ (which will be unique if s is of dimension exactly 3) using $\mathrm{poly}(m)$ field operations, again using Gaussian elimination.

The algorithm is as follows.

Algorithm `Randomness-Efficient-Low-Degree-Tester-With-Large-Answer-Size`$_{m,d,\mathbb{F},\mathbb{K}}$.

Requirements. $m \geq 3$ is a dimension parameter. d is a degree parameter. $\mathbb{F}$ is a finite field and $\mathbb{K} \subseteq \mathbb{F}$ is a subfield of it, such that all field and subfield operations (addition, multiplication, sampling of a field element, etc.) can be done in time $\mathrm{poly} \log |\mathbb{F}|$.

Oracles. The algorithm has oracle access to the following:

1. A function $f : \mathbb{F}^m \to \mathbb{F}$.
2. An auxiliary proof oracle $\pi : \mathcal{M}_{3,m,\mathbb{F}} \to \mathcal{P}_{3,d,\mathbb{F}}$, supposedly assigning each matrix the polynomial of degree at most d which is the restriction of f to the row space of the matrix.

Input. A point $\vec{x} \in \mathbb{F}^m$ on which we wish to test f.

Output. Either *accept*, or *reject*.

Guarantee (to be proven below).

- **Completeness:** For every function f which is a polynomial of degree at most d, there exists an auxiliary proof oracle π, such that for every input $\vec{x}$, the tester accepts with probability 1.
- **Soundness:** Let $\varepsilon_{m,d,\mathbb{F},\mathbb{K}} = 2^7 m \left(\sqrt[8]{\frac{1}{|\mathbb{K}|}} + \sqrt[4]{\frac{md}{|\mathbb{F}|}} \right)$. For any function f, for any $\delta \geq 2\sqrt{\frac{d}{|\mathbb{F}|}}$, there are $l \leq 2/\delta$ polynomials $Q_1, \ldots, Q_l : \mathbb{F}^m \to \mathbb{F}$ of degree at most d, such that for every auxiliary proof oracle π, when $\vec{x}$ is uniformly distributed in $\mathbb{F}^m$, the probability – over $\vec{x}$ and over the randomness of the tester – that the tester accepts although $f(\vec{x}) \notin \{Q_1(\vec{x}), \ldots, Q_l(\vec{x})\}$ is at most $\delta + 3\varepsilon_{m,d,\mathbb{F},\mathbb{K}} + \frac{d+1}{|\mathbb{F}|}$.

Process.

1. **Pick three-dimensional subspace through $\vec{x}$.** Pick uniformly at random $\vec{y}_1, \vec{y}_2 \in \mathbb{K}^m$. Obtain the matrix $M \in \mathcal{M}_{3,m,\mathbb{F}}$ in row canonical form whose row space is the linear subspace over $\mathbb{F}$ spanned by $\vec{x}, \vec{y}_1, \vec{y}_2$. If $\vec{x}, \vec{y}_1, \vec{y}_2$ are linearly dependent, *accept*.
2. **Query subspace.** Query π on M and obtain its evaluation on the point $\vec{x}$ by computing $\vec{t} \in \mathbb{F}^3$ such that $M^T \vec{t} = \vec{x}$ and evaluating $y = \pi(M)(\vec{t})$.
3. **Compare to point.** Query f on $\vec{x}$ and compare the two evaluations, if $y = f(\vec{x})$, *accept*; otherwise, *reject*.

Running Time. Step 1: Picking $\vec{y}_1, \vec{y}_2$ can be done in time $\text{poly}(m, \log|\mathbb{F}|)$. Obtaining the matrix M in row canonical form associated with the linear subspace over $\mathbb{F}$ spanned by $\vec{x}, \vec{y}_1, \vec{y}_2$ can be done in time $\text{poly}(m, \log|\mathbb{F}|)$ using Gaussian elimination. Checking the linear dependence between $\vec{x}, \vec{y}_1, \vec{y}_2$ amounts to checking if the last row in M is zero, which can be checked in time $\text{poly}(m, \log|\mathbb{F}|)$.

Step 2: Computing $\vec{t} \in \mathbb{F}^3$ such that $M^T \vec{t} = \vec{x}$ can be done using Gaussian elimination in time $\text{poly}(m, \log|\mathbb{F}|)$. Evaluating the polynomial $\pi(M)$ on $\vec{t}$ can be done in time $\text{poly}(d, \log|\mathbb{F}|)$.

Step 3: Comparing the field elements can be done in time $\text{poly}\log|\mathbb{F}|$.

The total running time of the algorithm is hence $\text{poly}(m, d, \log|\mathbb{F}|)$.

Randomness. The tester requires $2m \log|\mathbb{K}|$ random bits to pick $\vec{y}_1, \vec{y}_2$ in step 1. Note that we do not count the randomness that may be required to pick the point $\vec{x}$, which is given as input.

Query Complexity. The tester makes one query to the function f on the point $\vec{x}$ and one query to the auxiliary proof oracle π, which is a total of two queries.

Answer Size. The answer size is $\text{poly}(d, \log|\mathbb{F}|)$.

Correctness.

Lemma 3.4 (Completeness). *For every function f which is a polynomial of degree at most d, there exists an auxiliary proof oracle π, such that for every input $\vec{x}$, the tester accepts with probability* 1.

Proof. Fix a function f which is a polynomial of degree at most d. For every matrix $M \in \mathcal{M}_{3,m,\mathbb{F}}$, define $\pi(M)$ to be the restriction of f to the subspace of dimension at most three over $\mathbb{F}$ represented by M, namely, let, for every $\vec{t} \in \mathbb{F}^3$, $\pi(M)(\vec{t}) = f(M^T\vec{t})$. Note that indeed $\pi(M) \in \mathcal{P}_{3,d,\mathbb{F}}$. For every input $\vec{x}$, in the only event in which the tester rejects, there exist a matrix $M \in \mathcal{M}_{3,m,\mathbb{F}}$ and a scalar $\vec{t} \in \mathbb{F}^3$, such that $M^T\vec{t} = \vec{x}$ and $\pi(M)(\vec{t}) \neq f(\vec{x})$, but this cannot happen since $\pi(M)(\vec{t}) = f(M^T\vec{t})$. ■

Lemma 3.5 (Soundness). *Let $\varepsilon_{m,d,\mathbb{F},\mathbb{K}} = 2^7 m \left(\sqrt[8]{\frac{1}{|\mathbb{K}|}} + \sqrt[4]{\frac{md}{|\mathbb{F}|}} \right)$. For any function f, for any $\delta \geq 2\sqrt{\frac{d}{|\mathbb{F}|}}$, there are $l \leq 2/\delta$ polynomials $Q_1, \ldots, Q_l : \mathbb{F}^m \to \mathbb{F}$ of degree at most d, such that for every auxiliary proof oracle π, when $\vec{x}$ is uniformly distributed in $\mathbb{F}^m$, the probability – over $\vec{x}$ and over the randomness of the tester – that the tester accepts although $f(\vec{x}) \notin \{Q_1(\vec{x}), \ldots, Q_l(\vec{x})\}$ is at most $\delta + 3\varepsilon_{m,d,\mathbb{F},\mathbb{K}} + \frac{d+1}{|\mathbb{F}|}$.*

Proof. Fix a function $f : \mathbb{F}^m \to \mathbb{F}$ and $\delta \geq 2\sqrt{\frac{d}{|\mathbb{F}|}}$. Let $Q_1, \ldots, Q_l$ be *all* the different polynomials Q_i of degree at most d that agree with f on fraction of at least δ of the points, $\Pr_{\vec{x}\in\mathbb{F}^m}[Q_i(\vec{x}) = f(\vec{x})] \geq \delta$. By proposition 2.3, there are few such polynomials: $l \leq 2/\delta$.

Assume on way of contradiction that there exists an auxiliary proof oracle π, such that, when $\vec{x}$ is uniformly distributed in $\mathbb{F}^m$, the probability – over $\vec{x}$ and over the randomness of the tester – that the tester accepts although $f(\vec{x}) \notin \{Q_1(\vec{x}), \ldots, Q_l(\vec{x})\}$ is more than $\delta' \stackrel{def}{=} \delta + 3\varepsilon_{m,d,\mathbb{F},\mathbb{K}} + \frac{d+1}{|\mathbb{F}|}$.

Let us say that $\vec{x} \in \mathbb{F}^m$ is *explained* if $f(\vec{x}) \in \{Q_1(\vec{x}), \ldots, Q_l(\vec{x})\}$. We will construct a new function $f' : \mathbb{F}^m \to \mathbb{F}$ that identifies with f on all points, except that it assigns explained points values that do not correspond to a polynomial of degree at most d. We can do that by choosing an arbitrary polynomial $Q' : \mathbb{F}^m \to \mathbb{F}$ of degree *precisely* $d+1$, and letting f' assign explained points $\vec{x} \in \mathbb{F}^m$ the value $Q'(\vec{x})$.

By the contradicting assumption, for a uniformly distributed $\vec{x} \in \mathbb{F}^m$, the probability – over $\vec{x}$ and over the randomness of the tester – that it accepts, given oracle access to f' and to π, on input $\vec{x}$, is more than δ'.

By [[12], Theorem 2, Decoding], there exists a polynomial $Q : \mathbb{F}^m \to \mathbb{F}$ of degree at most d, such that

$$\Pr_{\vec{x}\in\mathbb{F}^m}\left[Q(\vec{x}) = f'(\vec{x})\right] \geq \delta' - 3\varepsilon_{m,d,\mathbb{F},\mathbb{K}}$$

The polynomials Q and Q' are necessarily different, as they have different degrees. Moreover, they both have degrees at most $d+1$. Thus, by the Schwartz-Zippel lemma,

$$\Pr_{\vec{x}\in\mathbb{F}^m}\left[Q(\vec{x}) = Q'(\vec{x})\right] \leq \frac{d+1}{|\mathbb{F}|}$$

Therefore, the fraction of points $\vec{x} \in \mathbb{F}^m$ on which $Q(\vec{x}) = f'(\vec{x})$ and $Q(\vec{x}) \neq Q'(\vec{x})$ is at least $\delta' - 3\varepsilon_{m,d,\mathbb{F},\mathbb{K}} - \frac{d+1}{|\mathbb{F}|} = \delta$. Whenever $Q(\vec{x}) = f'(\vec{x})$ and $Q(\vec{x}) \neq Q'(\vec{x})$, we also have $f'(x) \neq Q'(\vec{x})$, and thus $f'(\vec{x}) = f(\vec{x})$. Hence, there is a fraction of at least δ of the points $\vec{x} \in \mathbb{F}^m$ for which (i) $Q(\vec{x}) = f(\vec{x})$; (ii) $f(\vec{x}) \notin \{Q_1(\vec{x}), \ldots, Q_l(\vec{x})\}$. But this cannot happen: the fact that (i) happens for a fraction of at least δ of the points implies that there exists $1 \leq i \leq l$, such that Q identifies with Q_i. On the other hand, the fact that (i) and (ii) occur for a positive fraction of the points implies the existence of a point $\vec{x} \in \mathbb{F}^m$ on which $Q_i(\vec{x}) \notin \{Q_1(\vec{x}), \ldots, Q_l(\vec{x})\}$, which is a contradiction. ■

Remark 3.6 (comparison with list-decoding version of [12]). *Note that the soundness claim does not follow directly from the list decoding version appearing in [[12], Theorem 2, List decoding], because (in one sense) the assertion there is weaker: it only implies that for every function f and auxiliary proof oracle π, there exists a list decoding. Here we assert that for every function f, there exists a list-decoding [the same for all π].*

On the other hand, the notion of list decoding appearing in [[12], Theorem 2, List decoding] is stronger than what is required here: there it is shown that the probability of accepting, although the polynomial assigned to the tested subspace is not the restriction of one of $Q_1, \ldots, Q_l$ to the subspace, is small. We only wish to argue this for the tested point.

4 Low Degree Reader With Large Answer Size

In this section we present an algorithm called a *low degree reader*. The algorithm has oracle access to π, supposedly encoding a low degree polynomial $Q : \mathbb{F}^m \to \mathbb{F}$. The purpose of the algorithm is to evaluate the low degree polynomial on points $\vec{x}_1, \ldots, \vec{x}_k \in \mathbb{F}^m$ it gets as input by making only a constant number of queries to π. If the algorithm detects that π is not a valid encoding of a low degree polynomial, it is allowed to reject.

The encoding and the reader should have the following two properties:

- **Every polynomial is realizable:** For every polynomial $Q : \mathbb{F}^m \to \mathbb{F}$, there exists π that encodes it. Given access to that π, the reader, on input $\vec{x}_1, \ldots, \vec{x}_k \in \mathbb{F}^m$, outputs $Q(\vec{x}_1), \ldots, Q(\vec{x}_k)$ (with probability 1).
- **A low degree polynomial is evaluated with high probability:** With every π, a few low degree polynomials $Q_1, \ldots, Q_l : \mathbb{F}^m \to \mathbb{F}$ are associated (*list decoding*). For *every* tuple $\vec{x}_1, \ldots, \vec{x}_k \in \mathbb{F}^m$, with high probability, whenever the reader does not reject and does output $a_1, \ldots, a_k$, for some $1 \leq i \leq l$ we must have $a_1 = Q_i(\vec{x}_1), \ldots, a_k = Q_i(\vec{x}_k)$ [the same i for the entire tuple!].

The algorithm uses the low degree tester of section 3 and applies the technique of aggregation through curves. It is not the final reader we construct, since it has two flaws:

1. The answer size of π is too large: it depends polynomially on the degree, which will be too large for the degree parameter we use.
2. Its randomness complexity is too high: it needs more than $m \log |\mathbb{F}|$ random bits per input tuple $\vec{x}_1, \ldots, \vec{x}_k$.

4.1 Random Curves

The algorithm passes a curve through the points $\vec{x}_1, \ldots, \vec{x}_k$ it gets as input and through an additional uniformly distributed point in $\mathbb{F}^m$. The following proposition shows that each of the points on the curve, but those forced to be $\vec{x}_1, \ldots, \vec{x}_k$, is uniformly distributed in $\mathbb{F}^m$.

We use the notation of proposition 2.8.

Proposition 4.1 (random curve). *Let $t_1, \ldots, t_k \in \mathbb{F}$ be different scalars, and let $\vec{x}_1, \ldots, \vec{x}_k \in \mathbb{F}^m$. Let $t_{k+1} \in \mathbb{F}$ be some scalar different than $t_1, \ldots, t_k$. Let X_{k+1} be uniformly distributed in $\mathbb{F}^m$. Then, for every $t \in \mathbb{F} - \{t_1, \ldots, t_k\}$, the distribution of $c_{t_1,\ldots,t_k,t_{k+1},\vec{x}_1,\ldots,\vec{x}_k,X_{k+1}}(t)$ is uniform in $\mathbb{F}^m$.*

Proof. Fix $t \in \mathbb{F} - \{t_1, \ldots, t_k\}$. Denote $X_{k+1} = (X_{k+1,1}, \ldots, X_{k+1,m})$, where $X_{k+1,1}, \ldots, X_{k+1,m}$ are uniformly and independently distributed in $\mathbb{F}$.

Recall that

$$c_{t_1,\ldots,t_k,t_{k+1},\vec{x}_1,\ldots,\vec{x}_k,X_{k+1}}(t) = \left(P_{t_1,\ldots,t_k,t_{k+1},x_{1,1},\ldots,x_{k,1},X_{k+1,1}}(t), \ldots, P_{t_1,\ldots,t_k,t_{k+1},x_{1,m},\ldots,x_{k,m},X_{k+1,m}}(t)\right)$$

Let $1 \leq l \leq m$. Let $a \in \mathbb{F}$. We will show that there exists precisely one value $x \in \mathbb{F}$ for $X_{k+1,l}$, such that $P_{t_1,\ldots,t_k,t_{k+1},x_{1,l},\ldots,x_{k,l},x}(t) = a$, and thus $P_{t_1,\ldots,t_k,t_{k+1},x_{1,l},\ldots,x_{k,l},X_{k+1,l}}(t)$ is uniformly distributed in $\mathbb{F}$. The proposition follows.

Recall that as in proposition 2.5,

$$P_{t_1,\ldots,t_k,t_{k+1},x_{1,l},\ldots,x_{k,l},x}(t) = \sum_{i=1}^{k} I_{\{t_1,\ldots,t_k,t_{k+1}\},t_i}(t) \cdot x_{i,l} + I_{\{t_1,\ldots,t_k,t_{k+1}\},t_{k+1}}(t) \cdot x$$

Denote $A \stackrel{def}{=} \sum_{i=1}^{k} I_{\{t_1,\ldots,t_k,t_{k+1}\},t_i}(t) x_{i,l}$. Let $B \stackrel{def}{=} I_{\{t_1,\ldots,t_k,t_{k+1}\},t_{k+1}}(t)$. Both A and B are scalars in the field. Note that, by proposition 2.4, (i)+(iii), $B \neq 0$, since $t \notin \{t_1, \ldots, t_k\}$. Then, $P_{t_1,\ldots,t_k,t_{k+1},x_{1,l},\ldots,x_{k,l},x}(t) = A + Bx$. The equation $A + Bx = a$ in the variable $x \in \mathbb{F}$ has exactly one solution $x = (a - A)/B \in \mathbb{F}$. ∎

4.2 Low Degree Reader

Algorithm `Low-Degree-Reader-With-Large-Answer-Size`$_{m,d,\mathbb{F},\mathbb{K},k}$**.**

Requirements. $m \geq 3$ is a dimension parameter. $d \geq 1$ is a degree parameter. $\mathbb{F}$ is a finite field and $\mathbb{K} \subseteq \mathbb{F}$ is a subfield of it, such that all field and subfield operations (addition, multiplication, sampling of a field element, retrieval of the i'th element in the field, etc.) can be done in time $\operatorname{poly}\log|\mathbb{F}|$. k is the number of points to be read. We assume that $k \leq |\mathbb{F}|/2$.

Oracles. The algorithm has oracle access to $\pi = (\pi_1, \pi_2, \pi_3)$ for

1. A function $\pi_1 : \mathbb{F}^m \to \mathbb{F}$, supposedly representing a polynomial of degree at most d.
2. An auxiliary proof oracle $\pi_2 : \mathcal{M}_{3,m,\mathbb{F}} \to \mathcal{P}_{3,d,\mathbb{F}}$ as needed for the algorithm `Randomness-Efficient-Low-Degree-Tester-With-Large-Answer-Size`$_{m,d,\mathbb{F},\mathbb{K}}$ to test π_1.
3. An oracle for curves $\pi_3 : \mathcal{C}_k^{m,\mathbb{F}} \to \mathcal{P}_{1,kd,\mathbb{F}}$, supposedly assigning each curve the restriction of π_1 to it.

Input. Points $\vec{x}_1, \ldots, \vec{x}_k \in \mathbb{F}^m$.

Output. Either *reject*, or k scalars $a_1, \ldots, a_k \in \mathbb{F}$.

Guarantee (to be proven below).

- **Completeness:** For every polynomial $Q : \mathbb{F}^m \to \mathbb{F}$ of degree at most d, there exist π_1, π_2, π_3, such that for every input $\vec{x}_1, \ldots, \vec{x}_k$, the reader outputs $Q(\vec{x}_1), \ldots, Q(\vec{x}_k)$ with probability 1.
- **Soundness:** Let $\varepsilon_{m,d,\mathbb{F},\mathbb{K}} = 2^7 m \left(\sqrt[8]{\frac{1}{|\mathbb{K}|}} + \sqrt[4]{\frac{md}{|\mathbb{F}|}} \right)$ as in the soundness of the algorithm $\texttt{Randomness-Efficient-Low-Degree-Tester-With-Large-Answer-Size}_{m,d,\mathbb{F},\mathbb{K}}$. For any π_1, π_2, π_3 and any $\delta \geq 2\sqrt{\frac{d}{|\mathbb{F}|}}$, there are $l \leq 2/\delta$ polynomials $Q_1, \ldots, Q_l : \mathbb{F}^m \to \mathbb{F}$ of degree at most d, such that the following holds: for every input $\vec{x}_1, \ldots, \vec{x}_k$, the probability that the reader outputs $a_1, \ldots, a_k$ such that there is no $1 \leq i \leq l$, for which $a_1 = Q_i(\vec{x}_1), \ldots, a_k = Q_i(\vec{x}_k)$, is at most $\delta + 4\varepsilon_{m,d,\mathbb{F},\mathbb{K}} + k \cdot 2\sqrt{\frac{d}{|\mathbb{F}|}}$.

Process.

1. **Pick curve through $\vec{x}_1, \ldots, \vec{x}_k$.** Pick uniformly at random $\vec{x}_{k+1}$ in $\mathbb{F}^m$. Generate a curve through $\vec{x}_1, \ldots, \vec{x}_k, \vec{x}_{k+1}$ by picking the first $k+1$ scalars in the field $t_1, \ldots, t_k, t_{k+1} \in \mathbb{F}$ [recall that the first scalars can be retrieved by our requirement from the field], and letting $c = c_{t_1,\ldots,t_k,t_{k+1},\vec{x}_1,\ldots,\vec{x}_k,\vec{x}_{k+1}}$ [recall that $c_{t_1,\ldots,t_k,t_{k+1},\vec{x}_1,\ldots,\vec{x}_k,\vec{x}_{k+1}}$ is defined in proposition 2.8, where it is asserted that it is of degree at most k and for every $1 \leq i \leq k+1$, it holds that $\vec{x}_i = c(t_i)$].
2. **Pick point on curve.** Pick uniformly at random $t \in \mathbb{F} - \{t_1, \ldots, t_k\}$.
3. **Low degree test.** Run the algorithm $\texttt{Randomness-Efficient-Low-Degree-Tester-With-Large-Answer-Size}_{m,d,\mathbb{F},\mathbb{K}}$ on oracle access to π_1 and π_2 and input $c(t)$ [recall that by proposition 4.1, $c(t)$ is uniformly distributed in $\mathbb{F}^m$]. If the algorithm rejects, *reject*.
4. **Query curve.** Query π_3 for the evaluations of $\vec{x}_1 = c(t_1), \ldots, \vec{x}_k = c(t_k), c(t)$, that is, let $a_1 = \pi_3(c)(t_1), \ldots, a_k = \pi_3(c)(t_k), a_{k+1} = \pi_3(c)(t)$.
5. **Check curve.** If $a_{k+1} \neq \pi_1(c(t))$, *reject*.
6. **Output.** Return $a_1, \ldots, a_k$.

Running Time. Step 1: Picking $\vec{x}_{k+1}$ can be done in time $\text{poly}(m, \log|\mathbb{F}|)$. Picking $t_1, \ldots, t_k, t_{k+1} \in \mathbb{F}$ can be done in time $\text{poly}(k, \log|\mathbb{F}|)$. Generating the curve can be done in time $\text{poly}(m, k, \log|\mathbb{F}|)$.

Step 2: Picking $t \in \mathbb{F} - \{t_1, \ldots, t_k\}$ can be done in time $\text{poly}\log|\mathbb{F}|$.

Step 3: Evaluating $c(t)$ can be done in time $\text{poly}(m, k, \log|\mathbb{F}|)$. Running the low degree tester can be done in time $\text{poly}(m, d, \log|\mathbb{F}|)$.

Step 4: Evaluating $\pi_3(c)$ on $t_1, \ldots, t_k, t$ can be done in time $\text{poly}(d, k, \log|\mathbb{F}|)$.

Step 5: Comparing the two field elements can be done in time $\text{poly}\log|\mathbb{F}|$ [recall that $c(t)$ was evaluated in an earlier stage of the algorithm].

The total running time of the algorithm is hence $\text{poly}(m, d, k, \log|\mathbb{F}|)$.

Randomness. The reader requires $m \log |\mathbb{F}|$ random bits to pick $\vec{x}_{k+1} \in \mathbb{F}^m$ in step 1. The reader requires additional $\log |\mathbb{F}|$ random bits to pick $t \in \mathbb{F} - \{t_1, \ldots, t_k\}$ in step 2. The low degree testing in step 3 requires $2m \log |\mathbb{K}|$ random bits.

The total randomness complexity of the reader is $(m+1) \log |\mathbb{F}| + 2m \log |\mathbb{K}|$ random bits.

Query Complexity. The reader queries π_1 on the point $c(t)$ [the value is required for step 3 and for step 5]. The low degree testing in step 3 requires one more query of π_2. The reader also queries π_3 on the curve c in step 4. The total number of queries made is 3.

Answer Size. The answer size required for π_1 is $\log |\mathbb{F}|$. The answer size required for π_2 is $\text{poly}(d, \log |\mathbb{F}|)$. The answer size required for π_3 is $\text{poly}(d, k, \log |\mathbb{F}|)$.

Thus, the answer size is $\text{poly}(d, k, \log |\mathbb{F}|)$.

Correctness.

Lemma 4.2 (Completeness). *For every polynomial $Q : \mathbb{F}^m \to \mathbb{F}$ of degree at most d, there exist π_1, π_2, π_3, such that for every input $\vec{x}_1, \ldots, \vec{x}_k$, the reader outputs $Q(\vec{x}_1), \ldots, Q(\vec{x}_k)$ with probability 1.*

Proof. Let π_1 be Q. Let π_2 be the oracle guaranteed in the completeness of algorithm `Randomness-Efficient-Low-Degree-Tester-With-Large-Answer-Size`$_{m,d,\mathbb{F},\mathbb{K}}$ (lemma 3.4) for π_1. Let π_3 assign every curve $c \in \mathcal{C}_k^{m,\mathbb{F}}$ the polynomial $Q_{|c}$ [which is indeed a univariate polynomial of degree at most kd over $\mathbb{F}$, by proposition 2.7].

Let $\vec{x}_1, \ldots, \vec{x}_k$ be an input to the reader. Fix some randomness for the reader.

The reader does not reject in step 3, by the completeness of the low degree tester. The reader does not reject in step 5, since $a_{k+1} = \pi_3(c)(t) = Q_{|c}(t) = Q(c(t)) = \pi_1(c(t))$. Thus, the reader outputs $a_1, \ldots, a_k$. Moreover, for every $1 \leq j \leq k$, it holds that $a_j = \pi_3(c)(t_j) = Q_{|c}(t_j) = Q(c(t_j)) = Q(\vec{x}_j)$.

Therefore, the reader outputs $Q(\vec{x}_1), \ldots, Q(\vec{x}_k)$ with probability 1. ∎

Lemma 4.3 (Soundness). *Let $\varepsilon_{m,d,\mathbb{F},\mathbb{K}} = 2^7 m \left(\sqrt[8]{\frac{1}{|\mathbb{K}|}} + \sqrt[4]{\frac{md}{|\mathbb{F}|}} \right)$ as in the soundness of the algorithm `Randomness-Efficient-Low-Degree-Tester-With-Large-Answer-Size`$_{m,d,\mathbb{F},\mathbb{K}}$. For any π_1, π_2, π_3 and any $\delta \geq 2\sqrt{\frac{d}{|\mathbb{F}|}}$, there are $l \leq 2/\delta$ polynomials $Q_1, \ldots, Q_l : \mathbb{F}^m \to \mathbb{F}$ of degree at most d, such that the following holds: for every input $\vec{x}_1, \ldots, \vec{x}_k$, the probability that the reader outputs $a_1, \ldots, a_k$ such that there is no $1 \leq i \leq l$, for which $a_1 = Q_i(\vec{x}_1), \ldots, a_k = Q_i(\vec{x}_k)$, is at most $\delta + 4\varepsilon_{m,d,\mathbb{F},\mathbb{K}} + k \cdot 2\sqrt{\frac{d}{|\mathbb{F}|}}$.*

Proof. Fix oracles π_1, π_2, π_3. Let $\delta \geq 2\sqrt{\frac{d}{|\mathbb{F}|}}$. Let $Q_1, \ldots, Q_l : \mathbb{F}^m \to \mathbb{F}$ be the $l \leq 2/\delta$ polynomials of degree at most d corresponding to π_1 and δ that are guaranteed in the soundness of the algorithm `Randomness-Efficient-Low-Degree-Tester-With-Large-Answer-Size`$_{m,d,\mathbb{F},\mathbb{K}}$ (lemma 3.5).

If the reader does not reject, it does not reject in step 3 and in step 5. By proposition 4.1, $c(t)$ is uniformly distributed in $\mathbb{F}^m$. Thus, by the guarantee made on $Q_1, \ldots, Q_l$ in the soundness of the low degree tester, the probability that the reader does not reject in step 3, though $\pi_1(c(t)) \notin \{Q_1(c(t)), \ldots, Q_l(c(t))\}$ is at most $\delta + 3\varepsilon_{m,d,\mathbb{F},\mathbb{K}} + \frac{d+1}{|\mathbb{F}|} \leq \delta + 4\varepsilon_{m,d,\mathbb{F},\mathbb{K}}$ (where the last inequality follows from the definition of $\varepsilon_{m,d,\mathbb{F},\mathbb{K}}$ and the fact that $m \geq 3$ and $d \geq 1$).

If the reader does not reject in step 5, then $a_{k+1} = \pi_1(c(t))$. We have $a_{k+1} = \pi_3(c)(t)$. Thus, we obtain the following statement, which we will mark by (*): the probability that the reader does not reject, though $\pi_3(c)(t) \notin \{Q_1(c(t)), \ldots, Q_l(c(t))\}$ is at most $\delta + 4\varepsilon_{m,d,\mathbb{F},\mathbb{K}}$.

If $\pi_3(c)$ is not one of $Q_{1|c}, \ldots, Q_{l|c}$, then, for every $1 \leq i \leq l$, since both $\pi_3(c)$ and $Q_{i|c}$ are different polynomials of degree at most kd over $\mathbb{F}$ (see proposition 2.7), it follows from the Schwartz-Zippel lemma that on at most kd scalars $t_0 \in \mathbb{F}$, it holds that $\pi_3(c)(t_0) = Q_{i|c}(t_0)$. So, $\pi_3(c)(t_0) \in \{Q_1(c(t_0)), \ldots, Q_l(c(t_0))\}$ on at most lkd scalars $t_0 \in \mathbb{F}$. Since t is uniformly distributed in $\mathbb{F} - \{t_1, \ldots, t_k\}$, the probability that $\pi_3(c)(t) \in \{Q_1(c(t)), \ldots, Q_l(c(t))\}$, but $\pi_3(c)$ is not one of $Q_{1|c}, \ldots, Q_{l|c}$ is at most $\frac{lkd}{|\mathbb{F}|-k} \leq k \cdot 2\sqrt{\frac{d}{|\mathbb{F}|}}$ (where the last inequality follows since $k \leq |\mathbb{F}|/2$, and, hence, $\frac{d}{|\mathbb{F}|-k} \leq \frac{2d}{|\mathbb{F}|}$, and $l \leq 2/\delta \leq \sqrt{\frac{|\mathbb{F}|}{d}}$). Let us mark this statement by (**).

Therefore, from (*) and (**), we get that the probability that the reader does not reject, but $\pi_3(c)$ is not one of $Q_{1|c}, \ldots, Q_{l|c}$ is at most $(\delta + 4\varepsilon_{m,d,\mathbb{F},\mathbb{K}}) + k \cdot 2\sqrt{\frac{d}{|\mathbb{F}|}}$ (where the first addend bounds the probability that this happens and $\pi_3(c)(t) \notin \{Q_1(c(t)), \ldots, Q_l(c(t))\}$, and the second addend bounds the probability that this happens and $\pi_3(c)(t) \in \{Q_1(c(t)), \ldots, Q_l(c(t))\}$).

If there exists $1 \leq i \leq l$, such that $\pi_3(c)$ is $Q_{i|c}$, then for every $1 \leq j \leq k$, it holds that $a_j = \pi_3(c)(t_j) = Q_{i|c}(t_j) = Q_i(c(t_j)) = Q_i(\vec{x}_j)$. We conclude that the probability that the reader outputs $a_1, \ldots, a_k$ such that there is no $1 \leq i \leq l$, for which $a_1 = Q_i(\vec{x}_1), \ldots, a_k = Q_i(\vec{x}_k)$, is at most $\delta + 4\varepsilon_{m,d,\mathbb{F},\mathbb{K}} + k \cdot 2\sqrt{\frac{d}{|\mathbb{F}|}}$. ■

5 Low Degree Reader For Small Dimension

The purpose of this section is to adapt the low degree reader of section 4, whose answer size depends *polynomially* on d, into a low degree reader whose answer size has only a *logarithmic* dependence on d.

This adaptation allows to read from dimension m by applying the reader of section 4 on dimension md^* for d^* which is logarithmic in d. This adds a factor which is logarithmic in d into the randomness complexity, thus obtaining a very randomness-wasteful low degree reader. Nevertheless, we will only use the reader for a very small dimension m, and this way be able to maintain a low randomness-complexity.

The adaptation is by applying the *power substitution* technique from [9]. To explain the idea, let us focus on the case $m = 1$. Consider a univariate polynomial of degree $d \geq 1$ over a field $\mathbb{F}$, and let us denote it by $Q(x) = \sum_{i=0}^{d} a_i x^i$ for some coefficients $a_0, \ldots, a_d \in \mathbb{F}$. We can obtain from Q a polynomial of much lower degree as follows. Let $d^* = \lceil \log(d+1) \rceil$ denote the number of bits required to represent a number bounded by d in binary representation. Introduce d^* new variables $x_0, x_1, \ldots, x_{d^*-1}$ representing x raised to powers $2^0, 2^1, \ldots, 2^{d^*-1}$, i.e., let $x_0 = x^{2^0}, x_1 = x^{2^1}, \ldots, x_{d^*-1} = x^{2^{d^*-1}}$. Now, for every $0 \leq i \leq d$, we can express x^i by writing i in binary representation $i = \sum_{j=0}^{d^*-1} b_{i,j} 2^j$ for $b_{i,j} \in \{0,1\}$, and letting $x^i = x_0^{b_{i,0}} \cdots x_{d^*-1}^{b_{i,d^*-1}}$. This substitution gives a new polynomial Q^* in as many as d^* variables, but of degree at most d^* (rather than d) $Q^*(x_0, \ldots, x_{d^*-1}) = \sum_{i=0}^{d} a_i x_0^{b_{i,0}} \cdots x_{d^*-1}^{b_{i,d^*-1}}$. Thus, the idea is to read a polynomial of degree at most d^* on $\mathbb{F}^{d^*}$ on points of the form $(x^{2^0}, x^{2^1}, \ldots, x^{2^{d^*-1}})$ for $x \in \mathbb{F}$. As we saw, any univariate polynomial of degree at most d over $\mathbb{F}$ can be read this way. Moreover, any polynomial of degree at most d^* on $\mathbb{F}^{d^*}$ translates into a polynomial of degree at most d^*d on $\mathbb{F}$.

In subsection 5.1, we describe a mapping of $\mathbb{F}^m$ to $\mathbb{F}^{md^*}$ along the lines of the above description

for $m \geq 1$. In subsection 5.2, we describe the adaptation of the low degree reader using this mapping.

5.1 Degree-Reducing Mapping

Let $\mathbb{F}$ be a finite field. Let m be a dimension and let $d \geq 1$ be a degree. Let $d^* = \lceil \log(d+1) \rceil$. We will define a mapping $\varphi : \mathbb{F}^m \to \mathbb{F}^{md^*}$ as follows: for every $(x_1, \ldots, x_m) \in \mathbb{F}^m$, let

$$\varphi(x_1, \ldots, x_m) = (x_1^{2^0}, x_1^{2^1}, \ldots, x_1^{2^{d^*-1}}, \cdots, x_m^{2^0}, x_m^{2^1}, \ldots, x_m^{2^{d^*-1}})$$

Proposition 5.1 (degree reducing mapping). *The mapping $\varphi : \mathbb{F}^m \to \mathbb{F}^{md^*}$ has the following properties:*

1. *Given $\vec{x} \in \mathbb{F}^m$, one can compute $\varphi(\vec{x})$ using $\mathrm{poly}(m, \log d)$ field operations.*
2. *For every polynomial $Q : \mathbb{F}^m \to \mathbb{F}$ of degree at most d, there exists a polynomial $Q^* : \mathbb{F}^{md^*} \to \mathbb{F}$ of degree at most md^*, such that for every $\vec{x} \in \mathbb{F}^m$, it holds that $Q(\vec{x}) = Q^*(\varphi(\vec{x}))$.*
3. *For every polynomial $Q^* : \mathbb{F}^{md^*} \to \mathbb{F}$, it holds that $Q \stackrel{def}{=} Q^* \circ \varphi : \mathbb{F}^m \to \mathbb{F}$ is a polynomial of degree at most $d \cdot \deg Q^*$.*

Proof. Let us prove the properties:

(1) Given $(x_1, \ldots, x_m) \in \mathbb{F}^m$, in order to compute $\varphi(x_1, \ldots, x_m)$, we need, for every $1 \leq i \leq m$, to raise x_i to the the power of two $(d^* - 1)$ times. Thus the number of required field operations is at most $\mathrm{poly}(m, \log d)$.

(2) Assume that $Q : \mathbb{F}^m \to \mathbb{F}$ is a polynomial of degree at most d. Write $Q(x_1, \ldots, x_m) = \sum_{i_1,\ldots,i_m} a_{i_1,\ldots,i_m} x_1^{i_1} \cdots x_m^{i_m}$ for coefficients $a_{i_1,\ldots,i_m} \in \mathbb{F}$. For every $0 \leq i \leq d$, let $b_{i,d^*-1} \cdots b_{i,0}$ denote the binary representation of i, that is, $i = \sum_{j=0}^{d^*-1} b_{i,j} 2^j$, where $b_{i,j} \in \{0,1\}$ [note that the binary representation of i can be written this way since $2^{d^*} - 1 = 2^{\lceil \log(d+1) \rceil} - 1 \geq (d+1) - 1 = d$]. Define $Q^* : \mathbb{F}^{md^*} \to \mathbb{F}$ as follows:

$$Q^*(x_{1,0}, \ldots, x_{1,d^*-1}, \cdots, x_{m,0}, \ldots, x_{m,d^*-1}) = \sum_{i_1,\ldots,i_m} a_{i_1,\ldots,i_m} x_{1,0}^{b_{i_1,0}} \cdots x_{1,d^*-1}^{b_{i_1,d^*-1}} \cdot \cdots \cdot x_{m,0}^{b_{i_m,0}} \cdots x_{m,d^*-1}^{b_{i_m,d^*-1}}$$

Note that the polynomial Q^* is of degree at most md^*. Let $(x_1, \ldots, x_m) \in \mathbb{F}^m$. Then,

$$\begin{aligned}
Q^*(\varphi(x_1, \ldots, x_m)) &= Q^*(x_1^{2^0}, x_1^{2^1}, \ldots, x_1^{2^{d^*-1}}, \cdots, x_m^{2^0}, x_m^{2^1}, \ldots, x_m^{2^{d^*-1}}) \\
&= \sum_{i_1,\ldots,i_m} a_{i_1,\ldots,i_m} \left(x_1^{2^0}\right)^{b_{i_1,0}} \cdots \left(x_1^{2^{d^*-1}}\right)^{b_{i_1,d^*-1}} \cdot \cdots \cdot \left(x_m^{2^0}\right)^{b_{i_m,0}} \cdots \left(x_m^{2^{d^*-1}}\right)^{b_{i_m,d^*-1}} \\
&= \sum_{i_1,\ldots,i_m} a_{i_1,\ldots,i_m} x_1^{\sum_{j=0}^{d^*-1} b_{i_1,j} 2^j} \cdot \cdots \cdot x_m^{\sum_{j=0}^{d^*-1} b_{i_m,j} 2^j} \\
&= \sum_{i_1,\ldots,i_m} a_{i_1,\ldots,i_m} x_1^{i_1} \cdot \cdots \cdot x_m^{i_m} \\
&= Q(x_1, \ldots, x_m)
\end{aligned}$$

(3) Let $Q^* : \mathbb{F}^{md^*} \to \mathbb{F}$ be a polynomial. Write

$$Q^*(x_{1,0},\ldots,x_{1,d^*-1},\cdots,x_{m,0},\ldots,x_{m,d^*-1}) = \sum_{i_{1,0},\ldots,i_{m,d^*-1}} a_{i_{1,0},\ldots,i_{m,d^*-1}} x_{1,0}^{i_{1,0}} \cdots x_{1,d^*-1}^{i_{1,d^*-1}} \cdot \ldots \cdot x_{m,0}^{i_{m,0}} \cdots x_{m,d^*-1}^{i_{m,d^*-1}}$$

for coefficients $a_{i_{1,0},\ldots,i_{m,d^*-1}} \in \mathbb{F}$.

Let $(x_1,\ldots,x_m) \in \mathbb{F}^m$. Then,

$$\begin{aligned}
(Q^* \circ \varphi)(x_1,\ldots,x_m) &= Q^*(x_1^{2^0}, x_1^{2^1}, \ldots, x_1^{2^{d^*-1}}, \cdots, x_m^{2^0}, x_m^{2^1}, \ldots, x_m^{2^{d^*-1}}) \\
&= \sum_{i_{1,0},\ldots,i_{m,d^*-1}} a_{i_{1,0},\ldots,i_{m,d^*-1}} \left(x_1^{2^0}\right)^{i_{1,0}} \cdots \left(x_1^{2^{d^*-1}}\right)^{i_{1,d^*-1}} \cdots \left(x_m^{2^0}\right)^{i_{m,0}} \cdots \left(x_m^{2^{d^*-1}}\right)^{i_{m,d^*-1}} \\
&= \sum_{i_{1,0},\ldots,i_{m,d^*-1}} a_{i_{1,0},\ldots,i_{m,d^*-1}} x_1^{\sum_{j=0}^{d^*-1} 2^j i_{1,j}} \cdots x_m^{\sum_{j=0}^{d^*-1} 2^j i_{m,j}}
\end{aligned}$$

For every $0 \le j \le d^*-1$, we have $2^j \le 2^{d^*-1} = 2^{\lceil \log(d+1) \rceil - 1} \le d$ [where the last inequality holds since $\lceil \log(d+1) \rceil < \log(d+1)+1$, and thus $2^{\lceil \log(d+1) \rceil - 1} < d+1$. Note that for $d \ge 1$, $2^{\lceil \log(d+1) \rceil - 1}$ is integer, and thus it is at most d]. Hence, for every $i_{1,0},\ldots,i_{m,d^*-1}$, the degree of the corresponding monomial is bounded by $\sum_{l=1}^{m} \sum_{j=0}^{d^*-1} 2^j i_{l,j} \le d \cdot \sum_{l=1}^{m} \sum_{j=0}^{d^*-1} i_{l,j} \le d \cdot \deg Q^*$. Therefore, the degree of $Q^* \circ \varphi$ is at most $d \cdot \deg Q^*$. ■

5.2 Low Degree Reader

Algorithm `Low-Degree-Reader-For-Small-Dimension`$_{m,d,\mathbb{F},\mathbb{K},k}$**.**

Requirements. $m \ge 1$ is a dimension parameter. $d \ge 4$ is a degree parameter. $\mathbb{F}$ is a finite field and $\mathbb{K} \subseteq \mathbb{F}$ is a subfield of it, such that all field and subfield operations (addition, multiplication, sampling of a field element, retrieval of the i'th element in the field, etc.) can be done in time $\operatorname{poly}\log|\mathbb{F}|$. k is the number of points to be read. We assume that $k \le |\mathbb{F}|/2$.

Notation. $d^* = \lceil \log(d+1) \rceil$.

Oracles. The algorithm has oracle access to the following:

1. An oracle π which is the concatenation of all oracles required for the algorithm `Low-Degree-Reader-With-Large-Answer-Size`$_{md^*,md^*,\mathbb{F},\mathbb{K},k}$. [Note that the setting of parameters, $md^*, md^*, \mathbb{F}, \mathbb{K}, k$, satisfies all the requirements set for this algorithm: $d \ge 4$, and thus $d^* \ge 3$ and $md^* \ge 3$; $\mathbb{F}$ is a finite field and $\mathbb{K} \subseteq \mathbb{F}$ is a subfield of it as required there; $k \le |\mathbb{F}|/2$].

Input. Points $\vec{x}_1,\ldots,\vec{x}_k \in \mathbb{F}^m$.

Output. Either *reject*, or k scalars $a_1,\ldots,a_k \in \mathbb{F}$.

Guarantee (to be proven below).

- **Completeness:** For every polynomial $Q : \mathbb{F}^m \to \mathbb{F}$ of degree at most d, there exists π, such that for every input $\vec{x}_1, \ldots, \vec{x}_k$, the reader outputs $Q(\vec{x}_1), \ldots, Q(\vec{x}_k)$ with probability 1.
- **Soundness:** For any π and any $\delta \geq 2\sqrt{\frac{md^*}{|\mathbb{F}|}}$, there are $l \leq 2/\delta$ polynomials $Q_1, \ldots, Q_l : \mathbb{F}^m \to \mathbb{F}$ of degree at most $md^* \cdot d$, such that the following holds: for every input $\vec{x}_1, \ldots, \vec{x}_k$, the probability the reader outputs $a_1, \ldots, a_k$ such that there is no $1 \leq i \leq l$, for which $a_1 = Q_i(\vec{x}_1), \ldots, a_k = Q_i(\vec{x}_k)$, is at most $\delta + 4\varepsilon_{md^*,md^*,\mathbb{F},\mathbb{K}} + k \cdot 2\sqrt{\frac{md^*}{|\mathbb{F}|}}$. [recall that $\varepsilon_{md^*,md^*,\mathbb{F},\mathbb{K}} = 2^7 md^* \left(\sqrt[8]{\frac{1}{|\mathbb{K}|}} + \sqrt[4]{\frac{(md^*)^2}{|\mathbb{F}|}} \right)$]

Process.

1. Compute $\vec{y}_1 = \varphi(\vec{x}_1), \ldots, \vec{y}_k = \varphi(\vec{x}_k)$.
2. Apply the algorithm $\texttt{Low-Degree-Reader-With-Large-Answer-Size}_{md^*,md^*,\mathbb{F},\mathbb{K},k}$ with oracle access to π on input $\vec{y}_1, \ldots, \vec{y}_k \in \mathbb{F}^{md^*}$. If the algorithm rejects, *reject*; otherwise, if the algorithm outputs $a_1, \ldots, a_k \in \mathbb{F}$, output $a_1, \ldots, a_k$.

Running Time. Step 1: By proposition 5.1, computing $\vec{y}_1 = \varphi(\vec{x}_1), \ldots, \vec{y}_k = \varphi(\vec{x}_k)$ can be done in time $\text{poly}(m, \log d, k, \log |\mathbb{F}|)$.

Step 2: Running the algorithm $\texttt{Low-Degree-Reader-With-Large-Answer-Size}_{md^*,md^*,\mathbb{F},\mathbb{K},k}$ can be done in time $\text{poly}(m, \log d, k, \log |\mathbb{F}|)$.

The total running time is thus $\text{poly}(m, \log d, k, \log |\mathbb{F}|)$.

Randomness. The randomness complexity of the algorithm is $(md^* + 1) \log |\mathbb{F}| + 2md^* \log |\mathbb{K}|$ random bits.

Query Complexity. The number of queries made is 3.

Answer Size. The answer size is $\text{poly}(m, \log d, k, \log |\mathbb{F}|)$.

Correctness.

Lemma 5.2 (Completeness). *For every polynomial $Q : \mathbb{F}^m \to \mathbb{F}$ of degree at most d, there exists π, such that for every input $\vec{x}_1, \ldots, \vec{x}_k$, the reader outputs $Q(\vec{x}_1), \ldots, Q(\vec{x}_k)$ with probability 1.*

Proof. Fix a polynomial $Q : \mathbb{F}^m \to \mathbb{F}$ of degree at most d. By proposition 5.1, there exists a polynomial $Q^* : \mathbb{F}^{md^*} \to \mathbb{F}$ of degree at most md^*, such that for every $\vec{x} \in \mathbb{F}^m$, it holds that $Q(\vec{x}) = Q^*(\varphi(\vec{x}))$.

By the completeness of the algorithm $\texttt{Low-Degree-Reader-With-Large-Answer-Size}_{md^*,md^*,\mathbb{F},\mathbb{K},k}$ (lemma 4.2), for this Q^*, there exists π, such that for every input $\vec{y}_1, \ldots, \vec{y}_k \in \mathbb{F}^{md^*}$, the reader outputs $Q^*(\vec{y}_1), \ldots, Q^*(\vec{y}_k)$ with probability 1.

Consider this oracle π. In particular, for every $\vec{x}_1, \ldots, \vec{x}_k \in \mathbb{F}^m$, on input $\varphi(\vec{x}_1), \ldots, \varphi(\vec{x}_k)$, the reader outputs $Q^*(\varphi(\vec{x}_1)), \ldots, Q^*(\varphi(\vec{x}_k))$, that is, $Q(\vec{x}_1), \ldots, Q(\vec{x}_k)$, with probability 1. ■

Lemma 5.3 (Soundness). *For any π and any $\delta \geq 2\sqrt{\frac{md^*}{|\mathbb{F}|}}$, there are $l \leq 2/\delta$ polynomials $Q_1, \ldots, Q_l : \mathbb{F}^m \to \mathbb{F}$ of degree at most $md^* \cdot d$, such that the following holds: for every input $\vec{x}_1, \ldots, \vec{x}_k$, the probability the reader outputs $a_1, \ldots, a_k$ such that there is no $1 \leq i \leq l$, for which $a_1 = Q_i(\vec{x}_1), \ldots, a_k = Q_i(\vec{x}_k)$, is at most $\delta + 4\varepsilon_{md^*,md^*,\mathbb{F},\mathbb{K}} + k \cdot 2\sqrt{\frac{md^*}{|\mathbb{F}|}}$. [recall that $\varepsilon_{md^*,md^*,\mathbb{F},\mathbb{K}} = 2^7 md^* \left(\sqrt[8]{\frac{1}{|\mathbb{K}|}} + \sqrt[4]{\frac{(md^*)^2}{|\mathbb{F}|}} \right)$]*

Proof. Fix oracle π and $\delta \geq 2\sqrt{\frac{md^*}{|\mathbb{F}|}}$.

Let $Q_1^*, \ldots, Q_l^* : \mathbb{F}^{md^*} \to \mathbb{F}$ be the $l \leq 2/\delta$ polynomials of degree at most md^* guaranteed in the soundness of the algorithm $\texttt{Low-Degree-Reader-With-Large-Answer-Size}_{md^*,md^*,\mathbb{F},\mathbb{K},k}$ (lemma 4.3) for π and δ.

For every $1 \leq i \leq l$, let $Q_i : \mathbb{F}^m \to \mathbb{F}$ be $Q_i^* \circ \varphi$. By proposition 5.1, the polynomials $Q_1, \ldots, Q_l$ are of degree at most $md^* \cdot d$.

Let $\vec{x}_1, \ldots, \vec{x}_k \in \mathbb{F}^m$ be an input to the algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{m,d,\mathbb{F},\mathbb{K},k}$. By the design of the algorithm and the guarantee on $Q_1^*, \ldots, Q_l^*$, with probability at least $1 - (\delta + 4\varepsilon_{md^*,md^*,\mathbb{F},\mathbb{K}} + k \cdot 2\sqrt{\frac{md^*}{|\mathbb{F}|}})$, either the algorithm rejects, or it outputs $a_1, \ldots, a_k \in \mathbb{F}$, for which there exists $1 \leq i \leq l$ with $a_1 = Q_i^*(\varphi(\vec{x}_1)) = Q_i(\vec{x_1}), \ldots, a_k = Q_i^*(\varphi(\vec{x}_k)) = Q_i(\vec{x}_k)$. ■

6 Randomness-Efficient Low Degree Tester

In this section we use the low degree reader for small dimension of section 5 to construct a low degree tester with relatively small answer size. The tester is obtained from the tester with large answer size of section 3, when instead of querying polynomials for subspaces, we use low degree reading.

Algorithm $\texttt{Randomness-Efficient-Low-Degree-Tester}_{m,d,\mathbb{F},\mathbb{K}}$.

Requirements. $m \geq 3$ is a dimension parameter. $d \geq 4$ is a degree parameter. $\mathbb{F}$ is a finite field and $\mathbb{K} \subseteq \mathbb{F}$ is a subfield of it, such that all field and subfield operations (addition, multiplication, sampling of a field element, retrieval of the i'th element in the field, etc.) can be done in time $\text{poly} \log |\mathbb{F}|$.

Oracles. The algorithm has oracle access to the following:

1. A function $f : \mathbb{F}^m \to \mathbb{F}$.
2. For every matrix $M \in \mathcal{M}_{3,m,\mathbb{F}}$ (representing a subspace of dimension at most 3 in $\mathbb{F}^m$; see section 3), an auxiliary proof oracle π_M as needed for the algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{3,d,\mathbb{F},\mathbb{K},1}$ [note that the setting of parameters satisfies all the requirements made for the algorithm: $3 \geq 1$, $d \geq 4$, $\mathbb{F}$ is a finite field and $\mathbb{K} \subseteq \mathbb{F}$ is a subfield of it as required there; $1 \leq |\mathbb{F}|/2$].

Input. A point $\vec{x} \in \mathbb{F}^m$ on which we wish to test f.

Output. Either *accept*, or *reject*.

Guarantee (to be proven below).

- **Completeness:** For every function f which is a polynomial of degree at most d, there exist auxiliary proof oracles $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$, such that for every input $\vec{x}$, the tester accepts with probability 1.
- **Soundness:** Let $d^* = \lceil \log(d+1) \rceil$ and $\varepsilon^{final-tester}_{m,d,\mathbb{F},\mathbb{K}} = 8\sqrt{\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}}} + 5\varepsilon_{3d^*,3d^*,\mathbb{F},\mathbb{K}}$ [recall that for every dimension m, degree d, field $\mathbb{F}$ and subfield $\mathbb{K}$, we define $\varepsilon_{m,d,\mathbb{F},\mathbb{K}} = 2^7 m \left(\sqrt[8]{\frac{1}{|\mathbb{K}|}} + \sqrt[4]{\frac{md}{|\mathbb{F}|}} \right)$].

  For any function f, for any $\delta \geq 3\sqrt{\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}}}$, there are $l \leq 18/\delta^2$ polynomials $Q_1, \ldots, Q_l : \mathbb{F}^m \to \mathbb{F}$ of degree at most $3d^* \cdot d$, such that for every auxiliary proof oracles $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$, when $\vec{x}$ is uniformly distributed in $\mathbb{F}^m$, the probability – over $\vec{x}$ and over the randomness of the tester – that the tester accepts although $f(\vec{x}) \notin \{Q_1(\vec{x}), \ldots, Q_l(\vec{x})\}$ is at most $\delta + \varepsilon^{final-tester}_{m,d,\mathbb{F},\mathbb{K}}$.

Process.

1. **Pick three-dimensional subspace through $\vec{x}$.** Pick uniformly at random $\vec{y}_1, \vec{y}_2 \in \mathbb{K}^m$. Obtain the matrix $M \in \mathcal{M}_{3,m,\mathbb{F}}$ in row canonical form whose row space is the linear subspace over $\mathbb{F}$ spanned by $\vec{x}, \vec{y}_1, \vec{y}_2$. If $\vec{x}, \vec{y}_1, \vec{y}_2$ are linearly dependent, *accept*.
2. **Query subspace.** Compute $\vec{t} \in \mathbb{F}^3$ such that $M^T\vec{t} = \vec{x}$. Invoke the algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{3,d,\mathbb{F},\mathbb{K},1}$ on oracle access to π_M and input $\vec{t}$. If the algorithm rejects, *reject*; otherwise, let y denote its output.
3. **Compare to point.** Query f on $\vec{x}$ and compare the two evaluations, if $y = f(\vec{x})$, *accept*; otherwise, *reject*.

Running Time. Step 1: As in step 1 of the algorithm $\texttt{Randomness-Efficient-Low-Degree-Tester-With-Large-Answer-Size}_{m,d,\mathbb{F},\mathbb{K}}$, this step can be done in time $\text{poly}(m, \log|\mathbb{F}|)$.

Step 2: Computing $\vec{t} \in \mathbb{F}^3$ such that $M^T\vec{t} = \vec{x}$ can be done using Gaussian elimination in time $\text{poly}(m, \log|\mathbb{F}|)$. The running time of the algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{3,d,\mathbb{F},\mathbb{K},1}$ is at most $\text{poly}(\log d, \log|\mathbb{F}|)$.

Step 3: Comparing the field elements can be done in time $\text{poly}\log|\mathbb{F}|$.

The total running time of the algorithm is hence $\text{poly}(m, \log d, \log|\mathbb{F}|)$.

Randomness. The tester requires $2m\log|\mathbb{K}|$ random bits to pick $\vec{y}_1, \vec{y}_2$ in step 1. The algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{3,d,\mathbb{F},\mathbb{K},1}$ requires additional $(3d^*+1)\log|\mathbb{F}| + 6d^*\log|\mathbb{K}|$ random bits.

Thus, the randomness complexity of the algorithm is $(3d^*+1)\log|\mathbb{F}| + (2m + 6d^*)\log|\mathbb{K}|$ [recall that we do not count the randomness that may be needed to generate $\vec{x}$].

Query Complexity. The tester makes one query to the function f on the point $\vec{x}$. The algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{3,d,\mathbb{F},\mathbb{K},1}$ makes additional 3 queries. Thus, the query complexity of the algorithm is 4 queries.

Answer Size. The answer size is $\text{poly}(\log d, \log|\mathbb{F}|)$, which is required for the algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{3,d,\mathbb{F},\mathbb{K},1}$.

Correctness.

Lemma 6.1 (Completeness). *For every function f which is a polynomial of degree at most d, there exist auxiliary proof oracles $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$, such that for every input $\vec{x}$, the tester accepts with probability* 1*.*

Proof. Fix a function f which is a polynomial of degree at most d. Consider some matrix $M \in \mathcal{M}_{3,m,\mathbb{F}}$ representing a subspace of dimension at most 3 in $\mathbb{F}^m$. Let $Q_M : \mathbb{F}^3 \to \mathbb{F}$ be the restriction of f to the subspace, i.e., for every $\vec{t} \in \mathbb{F}^3$, let $Q_M(\vec{t}) = f(M^T\vec{t})$. Then, $Q_M : \mathbb{F}^3 \to \mathbb{F}$ is a polynomial of degree at most d. Let π_M be the auxiliary proof guaranteed in the completeness of the algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{3,d,\mathbb{F},\mathbb{K},1}$ for Q_M (lemma 5.2).

Let $\vec{x}$ be an input to the algorithm $\texttt{Randomness-Efficient-Low-Degree-Tester}_{m,d,\mathbb{F},\mathbb{K}}$. Fix some randomness for the algorithm.

The algorithm cannot reject in step 2. Moreover, it cannot reject in step 3, since this would imply the existence of $M \in \mathcal{M}_{3,m,\mathbb{F}}$ and $\vec{t} \in \mathbb{F}^3$ for which $M^T\vec{t} = \vec{x}$ and $Q_M(\vec{t}) \neq f(\vec{x})$, but $Q_M(\vec{t}) = f(M^T\vec{t}) = f(\vec{x})$.

Thus, the algorithm accepts with probability 1. ∎

Lemma 6.2 (Soundness). *Let $d^* = \lceil\log(d+1)\rceil$ and $\varepsilon^{final-tester}_{m,d,\mathbb{F},\mathbb{K}} = 8\sqrt{\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}}} + 5\varepsilon_{3d^*,3d^*,\mathbb{F},\mathbb{K}}$ [recall that for every dimension m, degree d, field $\mathbb{F}$ and subfield $\mathbb{K}$, we define $\varepsilon_{m,d,\mathbb{F},\mathbb{K}} = 2^7 m\left(\sqrt[8]{\frac{1}{|\mathbb{K}|}} + \sqrt[4]{\frac{md}{|\mathbb{F}|}}\right)$].*

For any function f, for any $\delta \geq 3\sqrt{\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}}}$, there are $l \leq 18/\delta^2$ polynomials $Q_1,\ldots,Q_l : \mathbb{F}^m \to \mathbb{F}$ of degree at most $3d^*\cdot d$, such that for every auxiliary proof oracles $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$, when $\vec{x}$ is uniformly distributed in $\mathbb{F}^m$, the probability – over $\vec{x}$ and over the randomness of the tester – that the tester accepts although $f(\vec{x}) \notin \{Q_1(\vec{x}),\ldots,Q_l(\vec{x})\}$ is at most $\delta + \varepsilon^{final-tester}_{m,d,\mathbb{F},\mathbb{K}}$.

Proof. Denote the algorithm $\texttt{Randomness-Efficient-Low-Degree-Tester}_{m,d,\mathbb{F},\mathbb{K}}$ by T. Its analysis is via comparison to the behavior of the algorithm $\texttt{Randomness-Efficient-Low-Degree-Tester-With-Large-Answer-Size}_{m,3d^*d,\mathbb{F},\mathbb{K}}$, which we will denote by $T^{(0)}$.

Let $f : \mathbb{F}^m \to \mathbb{F}$ be a function, and let $\delta \geq 3\sqrt{\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}}}$. Define $\delta_{tester} = \delta^2/9$ and $\delta_{reader} = \delta/3$.

Let $Q_1,\ldots,Q_l : \mathbb{F}^m \to \mathbb{F}$ be the $l \leq 2/\delta_{tester} = 18/\delta^2$ polynomials of degree at most $3d^*d$ guaranteed for f and $\delta_{tester} \geq 2\sqrt{\frac{3d^*d}{|\mathbb{F}|}}$ in the soundness of $T^{(0)}$ (lemma 3.5).

Let $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$ be auxiliary proof oracles. For every $M \in \mathcal{M}_{3,m,\mathbb{F}}$, let $Q_{M,1},\ldots,Q_{M,l_M} : \mathbb{F}^3 \to \mathbb{F}$ be $l_M \leq 2/\delta_{reader}$ polynomials of degree at most $3d^*d$ guaranteed for π_M and $\delta_{reader} \geq 2\sqrt{\frac{3d^*}{|\mathbb{F}|}}$ in the soundness of the algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{3,d,\mathbb{F},\mathbb{K},1}$ (lemma 5.3). Let us assume without loss of generality that all the lists (for all $M \in \mathcal{M}_{3,m,\mathbb{F}}$) are of the same length $l_M = l'$ for some $l' \leq 2/\delta_{reader}$ [as we can add dummy polynomials of degree at most $3d^*d$ to shorter lists].

Assume on way of contradiction that on input $\vec{x}$ uniformly distributed in $\mathbb{F}^m$, the probability – over $\vec{x}$ and over the randomness of T – that T accepts although $f(\vec{x}) \notin \{Q_1(\vec{x}),\ldots,Q_l(\vec{x})\}$ is more than $\delta + \varepsilon^{final-tester}_{m,d,\mathbb{F},\mathbb{K}}$.

Consider a new (imaginary) tester $T^{(1)}$ that behaves the same as T, except that we add an additional check to step 2:

2. **Query subspace.** Compute $\vec{t} \in \mathbb{F}^3$ such that $M^T\vec{t} = \vec{x}$. Invoke the algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{3,d,\mathbb{F},\mathbb{K},1}$ on oracle access to π_M and input $\vec{t}$. If the algorithm rejects, *reject*; otherwise, let y denote its output.
   If $y \notin \{Q_{M,1}(\vec{t}), \ldots, Q_{M,l'}(\vec{t})\}$, *reject*.

When invoked with oracle access to f and $\{\pi_M\}_{M \in \mathcal{M}_{3,m,\mathbb{F}}}$, on the same input $\vec{x}$ and randomness, the probability that T accepts, but $T^{(1)}$ rejects, is at most $\delta_{reader} + 4\varepsilon_{3d^*,3d^*,\mathbb{F},\mathbb{K}} + 2\sqrt{\frac{3d^*}{|\mathbb{F}|}} \leq \delta_{reader} + 5\varepsilon_{3d^*,3d^*,\mathbb{F},\mathbb{K}}$.

Thus, on oracle access to f and to $\{\pi_M\}_{M \in \mathcal{M}_{3,m,\mathbb{F}}}$, and input $\vec{x}$ uniformly distributed in $\mathbb{F}^m$, the probability – over $\vec{x}$ and over the randomness of $T^{(1)}$ – that $T^{(1)}$ accepts although $f(\vec{x}) \notin \{Q_1(\vec{x}), \ldots, Q_l(\vec{x})\}$, is more than $\delta + \varepsilon^{final-tester}_{m,d,\mathbb{F},\mathbb{K}} - \delta_{reader} - 5\varepsilon_{3d^*,3d^*,\mathbb{F},\mathbb{K}} = 2\delta/3 + 8\sqrt{\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}}}$.

Now consider another (imaginary) tester $T^{(2)}$ that behaves the same as $T^{(1)}$, except that we add an additional random decision to step 3:

3. **Compare to point.** Pick $i \in [l']$ uniformly at random. Query f on $\vec{x}$ and compare the two evaluations, if $y = f(\vec{x})$ **and** $y = Q_{M,i}(\vec{t})$, *accept*; otherwise, *reject*.

Assume that $T^{(1)}$ and $T^{(2)}$ are invoked on oracle access to f and to $\{\pi_M\}_{M \in \mathcal{M}_{3,m,\mathbb{F}}}$, on the same input $\vec{x}$ uniformly distributed in $\mathbb{F}^m$ and on the same randomness (let us think of $T^{(1)}$ as simply ignoring the additional randomness required for $T^{(2)}$). Let Bad denote the event that $f(\vec{x}) \notin \{Q_1(\vec{x}), \ldots, Q_l(\vec{x})\}$.

$$\Pr\left[T^{(1)}, T^{(2)} \text{ accept} \wedge Bad\right] = \Pr\left[T^{(1)}, T^{(2)} \text{ accept in step } 1 \wedge Bad\right] + \Pr\left[T^{(1)}, T^{(2)} \text{ accept in step } 3 \wedge Bad\right]$$

We have $\Pr\left[T^{(1)}, T^{(2)} \text{ accept in step } 1 \wedge Bad\right] = \Pr\left[T^{(1)} \text{ accepts in step } 1 \wedge Bad\right]$. Moreover,
$\Pr\left[T^{(1)}, T^{(2)} \text{ accept in step } 3 \wedge Bad\right] =$
$\Pr\left[T^{(1)} \text{ accepts in step } 3 \wedge Bad\right] \cdot \Pr\left[T^{(2)} \text{ accepts in step } 3 | T^{(1)} \text{ accepts in step } 3 \wedge Bad\right] \geq$
$\Pr\left[T^{(1)} \text{ accepts in step } 3 \wedge Bad\right] \cdot \frac{1}{l'}$. Hence,

$$\begin{aligned}\Pr\left[T^{(1)}, T^{(2)} \text{ accept} \wedge Bad\right] &\geq \Pr\left[T^{(1)} \text{ accepts in step } 1 \wedge Bad\right] + \Pr\left[T^{(1)} \text{ accepts in step } 3 \wedge Bad\right] \cdot \frac{1}{l'} \\ &\geq \Pr\left[T^{(1)} \text{ accepts} \wedge Bad\right] \cdot \frac{1}{l'}\end{aligned}$$

In particular, on oracle access to f and to $\{\pi_M\}_{M \in \mathcal{M}_{3,m,\mathbb{F}}}$, and input $\vec{x}$ uniformly distributed in $\mathbb{F}^m$, the probability – over $\vec{x}$ and over the randomness of $T^{(2)}$ – that $T^{(2)}$ accepts although $f(\vec{x}) \notin \{Q_1(\vec{x}), \ldots, Q_l(\vec{x})\}$, is more than $\left(2\delta/3 + 8\sqrt{\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}}}\right) \cdot \frac{1}{l'}$. Recalling that $l' \leq 2/\delta_{reader} = 6/\delta$, this probability is at least $\left(2\delta/3 + 8\sqrt{\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}}}\right) \cdot \frac{\delta}{6} = \frac{\delta^2}{9} + 4\frac{\delta}{3}\sqrt{\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}}}$. Recalling that $\delta \geq 3\sqrt{\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}}}$ and $\delta_{tester} = \delta^2/9$, the last probability is at least $\delta_{tester} + 4\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}}$.

For every $i \in [l']$, let $T^{(2)}_i$ denote an algorithm that behaves the same as $T^{(2)}$, only that it deterministically picks i in step 3:

3. **Compare to point.** Query f on $\vec{x}$ and compare the two evaluations, if $y = f(\vec{x})$ **and** $y = Q_{M,i}(\vec{t})$, *accept*; otherwise, *reject*.

We conclude that there exists $i_0 \in [l']$, such that on input $\vec{x}$ uniformly distributed in $\mathbb{F}^m$, the probability – over $\vec{x}$ and over the randomness of $T^{(2)}_{i_0}$ – that $T^{(2)}_{i_0}$ accepts although $f(\vec{x}) \notin \{Q_1(\vec{x}), \ldots, Q_l(\vec{x})\}$, is more than $\delta_{tester} + 4\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}}$.

Define $\pi : \mathcal{M}_{3,m,\mathbb{F}} \to \mathcal{P}_{3,3d^*d,\mathbb{F}}$ as follows: for every $M \in \mathcal{M}_{3,m,\mathbb{F}}$, let $\pi(M)$ be $Q_{M,i_0} : \mathbb{F}^3 \to \mathbb{F}$. Recall that indeed Q_{M,i_0} is of degree at most $3d^*d$.

Now observe the behavior of $T^{(0)}$ when given oracle access to f and to π and of $T^{(2)}_{i_0}$ when given oracle access to f and to $\{\pi_M\}_{M \in \mathcal{M}_{3,m,\mathbb{F}}}$. Assume that both are invoked on the same input $\vec{x}$ uniformly distributed in $\mathbb{F}^m$ and on the same randomness (where $T^{(0)}$ ignores the additional randomness required for $T^{(2)}_{i_0}$). Whenever $T^{(2)}_{i_0}$ accepts, $T^{(0)}$ accepts.

Thus, on input $\vec{x}$ uniformly distributed in $\mathbb{F}^m$, the probability – over $\vec{x}$ and over the randomness of $T^{(0)}$ – that $T^{(0)}$ accepts although $f(\vec{x}) \notin \{Q_1(\vec{x}), \ldots, Q_l(\vec{x})\}$, is more than $\delta_{tester} + 4\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}} \geq \delta_{tester} + 3\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}} + \frac{3d^*d+1}{|\mathbb{F}|}$. This contradicts the guarantee on $Q_1, \ldots, Q_l$.

The lemma follows. ■

7 Randomness Efficient Low Degree Reader By Balancing Curves

In this section we construct our final low degree reader. The algorithm follows that of the low degree reader with large answer size from section 4 but differs from it in two places: (i) it reduces the answer size by low degree reading. (ii) it saves in the randomness complexity by guaranteeing a weaker property (which is still sufficient for our purposes) using a new technique of balancing curves.

The weaker property is as follows. Rather than getting as input a single tuple, the reader gets as input a collection of N tuples $\vec{x}^1_1, \ldots, \vec{x}^1_k, \ldots, \vec{x}^N_1, \ldots, \vec{x}^N_k \in \mathbb{F}^m$ and an index $i_0 \in [N]$ of a tuple. The reader should return values $a_1, \ldots, a_k \in \mathbb{F}$ corresponding to the evaluation of a low degree polynomial on $\vec{x}^{i_0}_1, \ldots, \vec{x}^{i_0}_k$. However, the reader is only required to succeed with high probability over its randomness *and over the choice of a tuple* $i_0 \in [N]$. That is, the reader may always produce answers that do not correspond to any low degree polynomial for a few tuples among the N.

7.1 Balanced Curves

The following lemma is behind the technique of balancing curves. It shows how to pass curves through each of the tuples in $\vec{x}^1_1, \ldots, \vec{x}^1_k, \ldots, \vec{x}^N_1, \ldots, \vec{x}^N_k$, such that for a uniformly distributed curve among the N, the distribution of a uniformly distributed point on the curve (other than those forced to be points from a tuple) is close to uniform in $\mathbb{F}^m$. The choice of the curves is done *deterministically* by a greedy algorithm.

Lemma 7.1 (balancing curves). *Assume that $\mathbb{F}$ is a field in which all field operations can be done in time* $\operatorname{poly}\log|\mathbb{F}|$. *Let m be a dimension and let $k < |\mathbb{F}|$. Fix different scalars $t_1, \ldots, t_{k+1} \in \mathbb{F}$. Let $0 < \epsilon \leq 1$.*

Given a collection of size N of k-tuples of points, $\vec{x}^1_1, \ldots, \vec{x}^1_k, \cdots, \vec{x}^N_1, \ldots, \vec{x}^N_k \in \mathbb{F}^m$, where $N \geq |\mathbb{F}^m|/\epsilon^2$, one can find, in time polynomial in N, points $\vec{x}^1_{k+1}, \ldots, \vec{x}^N_{k+1} \in \mathbb{F}^m$, such that the distribution $\mathcal{D}$ obtained by picking uniformly and independently at random $i \in [N]$ and $t \in \mathbb{F} - \{t_1, \ldots, t_k\}$ and computing $c_{t_1,\ldots,t_{k+1},\vec{x}^i_1,\ldots,\vec{x}^i_{k+1}}(t)$ is ϵ-close (in the l_1-norm) to the uniform distribution over $\mathbb{F}^m$.

Proof. We will show how to find $\vec{x}^1_{k+1}, \ldots, \vec{x}^N_{k+1}$ iteratively. For every $1 \leq i \leq N+1$, for every $\vec{x} \in \mathbb{F}^m$, we let $d_{i-1}(\vec{x})$ denote the number of times $\vec{x}$ is hit by the curves constructed until the i'th iteration, that is, $d_{i-1}(\vec{x}) = \left|\left\{1 \leq j < i,\ t \in \mathbb{F} - \{t_1, \ldots, t_k\} \;\middle|\; c_{t_1,\ldots,t_{k+1},\vec{x}^j_1,\ldots,\vec{x}^j_{k+1}}(t) = \vec{x}\right\}\right|$.

For every $i \in [N]$, in the i'th iteration, we find $\vec{x}^i_{k+1} \in \mathbb{F}^m$ that minimizes

$$\sum_{t \in \mathbb{F} - \{t_1,\ldots,t_k\}} d_{i-1}(c_{t_1,\ldots,t_{k+1},\vec{x}^i_1,\ldots,\vec{x}^i_{k+1}}(t))$$

The running time of this algorithm is polynomial in N, $|\mathbb{F}^m|$, $|\mathbb{F}|$, m, k, and, hence, polynomial in N.

Denote $F = |\mathbb{F}| - k$.

The crucial observation is the following:

Claim 7.1.1. *In every iteration $i \in [N]$, we have*

$$\sum_{t \in \mathbb{F} - \{t_1,\ldots,t_k\}} d_{i-1}(c_{t_1,\ldots,t_{k+1},\vec{x}^i_1,\ldots,\vec{x}^i_{k+1}}(t)) \leq \frac{(i-1)F^2}{|\mathbb{F}^m|}$$

Proof. Let X^i_{k+1} be a random variable uniformly distributed in $\mathbb{F}^m$. By proposition 4.1, for every $t \in \mathbb{F} - \{t_1, \ldots, t_k\}$, the random variable $c_{t_1,\ldots,t_{k+1},\vec{x}^i_1,\ldots,\vec{x}^i_k,X^i_{k+1}}(t)$ is uniformly distributed in $\mathbb{F}^m$.

Using the linearity of the expectation, we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{X^i_{k+1}}\left[\sum_{t \in \mathbb{F} - \{t_1,\ldots,t_k\}} d_{i-1}(c_{t_1,\ldots,t_{k+1},\vec{x}^i_1,\ldots,\vec{x}^i_k,X^i_{k+1}}(t))\right] &= \sum_{t \in \mathbb{F} - \{t_1,\ldots,t_k\}} \mathop{\mathbf{E}}_{X^i_{k+1}}\left[d_{i-1}(c_{t_1,\ldots,t_{k+1},\vec{x}^i_1,\ldots,\vec{x}^i_k,X^i_{k+1}}(t))\right] \\
&= \sum_{t \in \mathbb{F} - \{t_1,\ldots,t_k\}} \frac{1}{|\mathbb{F}^m|} \sum_{\vec{x} \in \mathbb{F}^m} d_{i-1}(\vec{x}) \\
&= \sum_{t \in \mathbb{F} - \{t_1,\ldots,t_k\}} \frac{(i-1)F}{|\mathbb{F}^m|} \\
&= \frac{(i-1)F^2}{|\mathbb{F}^m|}
\end{aligned}$$

Therefore, there exists $\vec{x} \in \mathbb{F}^m$, for which $\sum_{t \in \mathbb{F} - \{t_1,\ldots,t_k\}} d_{i-1}(c_{t_1,\ldots,t_{k+1},\vec{x}^i_1,\ldots,\vec{x}^i_k,\vec{x}}(t)) \leq \frac{(i-1)F^2}{|\mathbb{F}^m|}$. In particular, this holds for $\vec{x}^i_{k+1}$ that minimizes the sum. ■[of claim 7.1.1]

Applying the claim we can prove the following:

Claim 7.1.2. *For every $0 \leq i \leq N$,*

$$\sum_{\vec{x} \in \mathbb{F}^m} d_i(\vec{x})^2 \leq F^2 i \cdot \left(\frac{i-1}{|\mathbb{F}^m|} + 1\right)$$

Proof. We will prove the claim by induction on i. Note that for every $\vec{x} \in \mathbb{F}^m$, we have $d_0(\vec{x}) = 0$, so the claim holds for $i = 0$. Assume that the claim holds for $i - 1 \geq 0$ and let us prove it for $i \in [N]$.

Let us denote the set of the points that are hit by the curve constructed in the i'th iteration by $C_i = \left\{ c_{t_1,\ldots,t_{k+1},\vec{x}_1^i,\ldots,\vec{x}_{k+1}^i}(t) \mid t \in \mathbb{F} - \{t_1,\ldots,t_k\} \right\}$. Denote the multiplicity of a point $\vec{x} \in C_i$, i.e., the number of $t \in \mathbb{F} - \{t_1,\ldots,t_k\}$ for which $\vec{x} = c_{t_1,\ldots,t_{k+1},\vec{x}_1^i,\ldots,\vec{x}_{k+1}^i}(t)$, by $m_{i,\vec{x}}$. Then,

$$\begin{aligned}
\sum_{\vec{x}\in\mathbb{F}^m} d_i(\vec{x})^2 &= \sum_{\vec{x}\in\mathbb{F}^m - C_i} d_i(\vec{x})^2 + \sum_{\vec{x}\in C_i} d_i(\vec{x})^2 \\
&= \sum_{\vec{x}\in\mathbb{F}^m - C_i} d_{i-1}(\vec{x})^2 + \sum_{\vec{x}\in C_i} \left(d_{i-1}(\vec{x}) + m_{i,\vec{x}}\right)^2 \\
&= \sum_{\vec{x}\in\mathbb{F}^m} d_{i-1}(\vec{x})^2 + \sum_{\vec{x}\in C_i} 2 d_{i-1}(\vec{x}) m_{i,\vec{x}} + \sum_{\vec{x}\in C_i} m_{i,\vec{x}}^2 \\
&\leq \sum_{\vec{x}\in\mathbb{F}^m} d_{i-1}(\vec{x})^2 + 2\sum_{\vec{x}\in C_i} d_{i-1}(\vec{x}) m_{i,\vec{x}} + F \sum_{\vec{x}\in C_i} m_{i,\vec{x}} \\
&= \sum_{\vec{x}\in\mathbb{F}^m} d_{i-1}(\vec{x})^2 + 2\sum_{\vec{x}\in C_i} d_{i-1}(\vec{x}) m_{i,\vec{x}} + F^2
\end{aligned}$$

By claim 7.1.1,

$$\sum_{\vec{x}\in\mathbb{F}^m} d_i(\vec{x})^2 \leq \sum_{\vec{x}\in\mathbb{F}^m} d_{i-1}(\vec{x})^2 + \frac{2(i-1)F^2}{|\mathbb{F}^m|} + F^2$$

By the induction hypothesis,

$$\begin{aligned}
\sum_{\vec{x}\in\mathbb{F}^m} d_i(\vec{x})^2 &\leq F^2(i-1)\cdot\left(\frac{i-2}{|\mathbb{F}^m|} + 1\right) + \frac{2(i-1)F^2}{|\mathbb{F}^m|} + F^2 \\
&= F^2 i \cdot \left(\frac{i-1}{|\mathbb{F}^m|} + 1\right)
\end{aligned}$$

It follows that the claim holds for every $0 \leq i \leq N$. ■[of claim 7.1.2]

Recall that $\mathcal{D}$ is the distribution obtained by picking uniformly at random $i \in [N]$ and $t \in \mathbb{F} - \{t_1,\ldots,t_k\}$ and computing $c_{t_1,\ldots,t_{k+1},\vec{x}_1^i,\ldots,\vec{x}_{k+1}^i}(t)$. Note that for every $\vec{x} \in \mathbb{F}^m$ the probability that $\mathcal{D}$ assigns to $\vec{x}$ is exactly $\frac{d_N(\vec{x})}{NF}$. From claim 7.1.2, we can bound the square of the distance in the l_2-norm of $\mathcal{D}$ from uniform:

$$\begin{aligned}
\sum_{\vec{x}\in\mathbb{F}^m} \left(\frac{d_N(\vec{x})}{NF} - \frac{1}{|\mathbb{F}^m|}\right)^2 &= \frac{1}{(NF)^2}\sum_{\vec{x}\in\mathbb{F}^m} d_N(\vec{x})^2 - \frac{2}{|\mathbb{F}^m|}\sum_{\vec{x}\in\mathbb{F}^m}\frac{d_N(\vec{x})}{NF} + \sum_{\vec{x}\in\mathbb{F}^m}\frac{1}{|\mathbb{F}^m|^2} \\
&\leq \frac{F^2 N}{(NF)^2}\cdot\left(\frac{N-1}{|\mathbb{F}^m|} + 1\right) - \frac{2}{|\mathbb{F}^m|} + \frac{1}{|\mathbb{F}^m|} \\
&= \frac{1}{N} - \frac{1}{|\mathbb{F}^m|}\cdot\left(1 - \frac{N-1}{N}\right) \\
&\leq \frac{1}{N}
\end{aligned}$$

We now use Jensen's inequality to bound the distance in the l_1-norm. Since the square function is convex, we have

$$\left(\frac{1}{|\mathbb{F}^m|}\sum_{\vec{x}\in\mathbb{F}^m}\left|\frac{d_N(\vec{x})}{NF} - \frac{1}{|\mathbb{F}^m|}\right|\right)^2 \leq \frac{1}{|\mathbb{F}^m|}\sum_{\vec{x}\in\mathbb{F}^m}\left|\frac{d_N(\vec{x})}{NF} - \frac{1}{|\mathbb{F}^m|}\right|^2$$

Therefore, we can bound the distance in the l_1-norm by

$$\sum_{\vec{x}\in\mathbb{F}^m}\left|\frac{d_N(\vec{x})}{NF}-\frac{1}{|\mathbb{F}^m|}\right| \;\leq\; \sqrt{|\mathbb{F}^m|\sum_{\vec{x}\in\mathbb{F}^m}\left(\frac{d_N(\vec{x})}{NF}-\frac{1}{|\mathbb{F}^m|}\right)^2} \leq \sqrt{\frac{|\mathbb{F}^m|}{N}} \leq \epsilon$$

■

7.2 Low Degree Reader

Algorithm `Randomness-Efficient-Low-Degree-Reader`$_{m,d,\mathbb{F},\mathbb{K},k,N}$.

Requirements. $m \geq 3$ is a dimension parameter. $4 \leq d \leq |\mathbb{F}|$ is a degree parameter. $\mathbb{F}$ is a finite field and $\mathbb{K} \subseteq \mathbb{F}$ is a subfield of it, such that all field and subfield operations (addition, multiplication, sampling of a field element, retrieval of the i'th element in the field, etc.) can be done in time $\operatorname{poly}\log|\mathbb{F}|$. $k \geq 1$ is the number of points to be read. We assume that $(k+1) \leq |\mathbb{F}|/2$. N is the number of k-tuples from which the tuple to be read is chosen. We assume that $N \geq |\mathbb{F}^m|$.

Oracles. The algorithm has oracle access to the following:

1. A function $\pi : \mathbb{F}^m \to \mathbb{F}$, supposedly representing a polynomial of degree at most d.
2. Auxiliary proof oracles $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$ as needed for the algorithm `Randomness-Efficient-Low-Degree-Tester`$_{m,d,\mathbb{F},\mathbb{K}}$, allowing us to test π.
3. For every curve $c \in \mathcal{C}_k^{m,\mathbb{F}}$, a proof oracle π_c as required for the algorithm `Low-Degree-Reader-For-Small-Dimension`$_{1,kd,\mathbb{F},\mathbb{K},k+1}$, allowing us to read from polynomials on c.

Input. A collection of N tuples of points $\vec{x}_1^1,\ldots,\vec{x}_k^1,\cdots,\vec{x}_1^N,\ldots,\vec{x}_k^N \in \mathbb{F}^m$ and an index $i_0 \in [N]$, supposedly uniformly distributed in $[N]$.

Output. Either *reject*, or k scalars $a_1,\ldots,a_k \in \mathbb{F}$.

Guarantee (to be proven below).

- **Completeness:** For every polynomial $Q : \mathbb{F}^m \to \mathbb{F}$ of degree at most d, there exist π, $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$, $\{\pi_c\}_{c\in\mathcal{C}_k^{m,\mathbb{F}}}$, such that for every input $\vec{x}_1^1,\ldots,\vec{x}_k^1,\cdots,\vec{x}_1^N,\ldots,\vec{x}_k^N$ and every $i_0 \in [N]$, the reader outputs $Q(\vec{x}_1^{i_0}),\ldots,Q(\vec{x}_k^{i_0})$ with probability 1.
- **Soundness:** There exist explicit constants $0 < c_1, c_2 \leq 1$ and explicit polynomials $p_1(\cdot,\cdot,\cdot), p_2(\cdot,\cdot,\cdot)$, such that for

  $$\varepsilon = p_1(m,\log d,\log k)\left(\frac{1}{|\mathbb{K}|}\right)^{c_1} + p_2(m,\log d,k)\left(\frac{d}{|\mathbb{F}|}\right)^{c_2}$$

  the following holds.

  For any π, $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$, $\{\pi_c\}_{c\in\mathcal{C}_k^{m,\mathbb{F}}}$, for any $\delta \geq \varepsilon$, there are $l \leq O(1/\delta^2)$ polynomials $Q_1,\ldots,Q_l : \mathbb{F}^m \to \mathbb{F}$ of degree at most $d_{poly} = O(d\log d)$, such that for every input

$\vec{x}_1^1, \ldots, \vec{x}_k^1, \cdots, \vec{x}_1^N, \ldots, \vec{x}_k^N$, for i_0 uniformly and independently distributed in $[N]$, the probability – over i_0 and over the randomness of the reader – that the reader outputs $a_1, \ldots, a_k$ such that there is no $1 \leq i \leq l$ for which $a_1 = Q_i(\vec{x}_1^{i_0}), \ldots, a_k = Q_i(\vec{x}_k^{i_0})$, is at most $\delta + \varepsilon + \sqrt{\frac{|\mathbb{F}^m|}{N}}$.

Process.

1. **Pick curve through $\vec{x}_1^{i_0}, \ldots, \vec{x}_k^{i_0}$.**

   Let $t_1, \ldots, t_k, t_{k+1} \in \mathbb{F}$ denote the first $k+1$ scalars in the field $\mathbb{F}$.

   For the collection $\vec{x}_1^1, \ldots, \vec{x}_k^1, \cdots, \vec{x}_1^N, \ldots, \vec{x}_k^N \in \mathbb{F}^m$ given as input, find, as in lemma 7.1, $\vec{x}_{k+1}^1, \ldots, \vec{x}_{k+1}^N \in \mathbb{F}^m$, such that the distribution $\mathcal{D}$ obtained by picking uniformly and independently at random $i \in [N]$ and $t \in \mathbb{F} - \{t_1, \ldots, t_k\}$ and computing $c_{t_1,\ldots,t_{k+1},\vec{x}_1^i,\ldots,\vec{x}_{k+1}^i}(t)$ is $\sqrt{\frac{|\mathbb{F}^m|}{N}}$-close (in the l_1-norm) to the uniform distribution over $\mathbb{F}^m$.

   Let $c = c_{t_1,\ldots,t_k,t_{k+1},\vec{x}_1^{i_0},\ldots,\vec{x}_k^{i_0},\vec{x}_{k+1}^{i_0}}$ [recall that by proposition 2.8, c is a curve in $\mathbb{F}^m$ of degree at most k].

2. **Pick point on curve.** Pick uniformly at random $t \in \mathbb{F} - \{t_1, \ldots, t_k\}$.

3. **Low degree test.** Run the algorithm $\texttt{Randomness-Efficient-Low-Degree-Tester}_{m,d,\mathbb{F},\mathbb{K}}$ on oracle access to π and $\{\pi_M\}_{M \in \mathcal{M}_{3,m,\mathbb{F}}}$ and input $c(t)$. If the algorithm rejects, *reject*.

4. **Query curve.** Run the algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{1,kd,\mathbb{F},\mathbb{K},k+1}$ on oracle access to π_c and input $t_1, \ldots, t_k, t$. If the algorithm rejects, *reject*; otherwise, let $a_1, \ldots, a_k, a_{k+1}$ denote its output.

5. **Check curve.** If $a_{k+1} \neq \pi(c(t))$, *reject*.

6. **Output.** Return $a_1, \ldots, a_k$.

Running Time. Step 1: Picking $t_1, \ldots, t_k, t_{k+1} \in \mathbb{F}$ can be done in time $\text{poly}(k, \log|\mathbb{F}|)$. Finding $\vec{x}_{k+1}^1, \ldots, \vec{x}_{k+1}^N$ can be done in time polynomial in N, by lemma 7.1. Generating the curve c can be done in time $\text{poly}(m, k, \log|\mathbb{F}|)$.

Step 2: Picking $t \in \mathbb{F} - \{t_1, \ldots, t_k\}$ can be done in time $\text{poly}\log|\mathbb{F}|$.

Step 3: Evaluating $c(t)$ can be done in time $\text{poly}(m, k, \log|\mathbb{F}|)$. Running the algorithm $\texttt{Randomness-Efficient-Low-Degree-Tester}_{m,d,\mathbb{F},\mathbb{K}}$ can be done in time $\text{poly}(m, \log d, \log|\mathbb{F}|)$.

Step 4: Running the algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{1,kd,\mathbb{F},\mathbb{K},k+1}$ can be done in time $\text{poly}(\log d, k, \log|\mathbb{F}|)$.

Step 5: Comparing the two field elements can be done in time $\text{poly}\log|\mathbb{F}|$ [recall that $c(t)$ was evaluated in an earlier stage of the algorithm].

Recalling that $N \geq |\mathbb{F}^m|$ and $k, d \leq |\mathbb{F}|$, the running time of the algorithm is $\text{poly}N$.

Randomness. We denote $(kd)^* = \lceil \log(kd+1) \rceil$.

The algorithm requires $\log|\mathbb{F}|$ random bits to pick $t \in \mathbb{F} - \{t_1, \ldots, t_k\}$ in step 2. The low degree testing in step 3 requires $(3d^*+1)\log|\mathbb{F}| + (2m+6d^*)\log|\mathbb{K}|$ random bits. The low degree reading in step 4 requires $((kd)^*+1)\log|\mathbb{F}| + 2(kd)^*\log|\mathbb{K}|$ random bits.

The total randomness complexity of the algorithm is $((kd)^* + 3d^* + 3)\log|\mathbb{F}| + 2(m + (kd)^* + 3d^*)\log|\mathbb{K}|$ random bits [Note that we do not count here the randomness required to generate $i_0 \in [N]$ given as input].

Query Complexity. The algorithm queries π on the point $c(t)$ [the value is required for step 3 and for step 5]. The low degree tester in step 3 makes additional 3 queries. The low degree reader in step 4 makes 3 more queries. The total number of queries made is 7.

Answer Size. The answer size required for π is $\log|\mathbb{F}|$. The answer size required for $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$ is $\text{poly}(\log d, \log|\mathbb{F}|)$. The answer size required for $\{\pi_c\}_{c\in\mathcal{C}_k^{m,\mathbb{F}}}$ is $\text{poly}(\log d, k, \log|\mathbb{F}|)$. Thus, we have a bound of $\text{poly}(\log d, k, \log|\mathbb{F}|)$ on the answer size.

Correctness.

Lemma 7.2 (Completeness). *For every polynomial $Q:\mathbb{F}^m\to\mathbb{F}$ of degree at most d, there exist π, $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$, $\{\pi_c\}_{c\in\mathcal{C}_k^{m,\mathbb{F}}}$, such that for every input $\vec{x}_1^1,\ldots,\vec{x}_k^1,\cdots,\vec{x}_1^N,\ldots,\vec{x}_k^N$ and $i_0\in[N]$, the reader outputs $Q(\vec{x}_1^{i_0}),\ldots,Q(\vec{x}_k^{i_0})$ with probability 1.*

Proof. Let $Q:\mathbb{F}^m\to\mathbb{F}$ be a polynomial of degree at most d. Take π to be Q. Let $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$ be the auxiliary proof oracles guaranteed in the completeness of algorithm $\texttt{Randomness-Efficient-Low-Degree-Tester}_{m,d,\mathbb{F},\mathbb{K}}$ (lemma 6.1) for π.

By proposition 2.7, for every curve $c\in\mathcal{C}_k^{m,\mathbb{F}}$, the polynomial $Q_{|c}:\mathbb{F}\to\mathbb{F}$ is a polynomial of degree at most kd. Let π_c be the oracle guaranteed by the completeness of the algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{1,kd,\mathbb{F},\mathbb{K},k+1}$ (lemma 5.2) for $Q_{|c}$.

Consider some $\vec{x}_1^1,\ldots,\vec{x}_k^1,\cdots,\vec{x}_1^N,\ldots,\vec{x}_k^N$ and $i_0\in[N]$ given as input to the algorithm $\texttt{Randomness-Efficient-Low-Degree-Reader}_{m,d,\mathbb{F},\mathbb{K},k,N}$. Fix some randomness for the algorithm.

The algorithm does not reject in step 3, by the completeness of the low degree tester. The algorithm does not reject in step 4, by the completeness of the low degree reader. Moreover, $a_1=Q_{|c}(t_1),\ldots,a_k=Q_{|c}(t_k),a_{k+1}=Q_{|c}(t)$. Thus, $a_{k+1}=Q_{|c}(t)=Q(c(t))=\pi(c(t))$, so the algorithm does not reject in step 5. Finally, by the choice of c and from proposition 2.8, the algorithm returns $a_1=Q_{|c}(t_1)=Q(c(t_1))=Q(\vec{x}_1^{i_0}),\ldots,a_k=Q_{|c}(t_k)=Q(c(t_k))=Q(\vec{x}_k^{i_0})$.

We conclude that the algorithm outputs $Q(\vec{x}_1^{i_0}),\ldots,Q(\vec{x}_k^{i_0})$ with probability 1. ∎

The guarantee on the soundness of the algorithm follows from the following lemma.

Lemma 7.3 (Soundness). *Let $d^*=\lceil\log(d+1)\rceil$; $(kd)^*=\lceil\log(kd+1)\rceil$. Define a degree bound for polynomials $d_{poly}=3d^*d$ and a degree bound on curves $d_{curves}=3(kd)^*kd$. Recall that for every dimension m, degree d, field $\mathbb{F}$ and subfield $\mathbb{K}$, we define $\varepsilon_{m,d,\mathbb{F},\mathbb{K}}=2^7m\left(\sqrt[8]{\frac{1}{|\mathbb{K}|}}+\sqrt[4]{\frac{md}{|\mathbb{F}|}}\right)$ and $\varepsilon_{m,d,\mathbb{F},\mathbb{K}}^{final-tester}=8\sqrt{\varepsilon_{m,3d^*d,\mathbb{F},\mathbb{K}}}+5\varepsilon_{3d^*,3d^*,\mathbb{F},\mathbb{K}}$ (see in lemma 6.2).*

For any π, $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$, $\{\pi_c\}_{c\in\mathcal{C}_k^{m,\mathbb{F}}}$ and any $\delta\geq\max\left\{6\sqrt{\varepsilon_{m,d_{poly},\mathbb{F},\mathbb{K}}},9\sqrt[6]{\frac{d_{curves}}{|\mathbb{F}|}}\right\}$, there are $l\leq 72/\delta^2$ polynomials $Q_1,\ldots,Q_l:\mathbb{F}^m\to\mathbb{F}$ of degree at most d_{poly}, such that the following holds: for every input $\vec{x}_1^1,\ldots,\vec{x}_k^1,\cdots,\vec{x}_1^N,\ldots,\vec{x}_k^N$, for i_0 uniformly and independently distributed in $[N]$, the probability – over i_0 and over the randomness of the reader – that the reader outputs $a_1,\ldots,a_k$ such that there is no $1\leq i\leq l$, for which $a_1=Q_i(\vec{x}_1^{i_0}),\ldots,a_k=Q_i(\vec{x}_k^{i_0})$, is at most $\delta+e_{balance}+e_{tester}+e_{reader}+e_{agree}$, where $e_{balance}=\sqrt{|\mathbb{F}^m|/N}$, $e_{tester}=\varepsilon_{m,d,\mathbb{F},\mathbb{K}}^{final-tester}$, $e_{reader}=4\varepsilon_{(kd)^*,(kd)^*,\mathbb{F},\mathbb{K}}+(k+1)2\sqrt{\frac{(kd)^*}{|\mathbb{F}|}}$ and $e_{agree}=\sqrt{\frac{d_{curves}}{|\mathbb{F}|}}$.

Proof. Fix oracles π, $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$, $\{\pi_c\}_{c\in\mathcal{C}_k^{m,\mathbb{F}}}$ and let $\delta\geq\max\left\{6\sqrt{\varepsilon_{m,d_{poly},\mathbb{F},\mathbb{K}}},9\sqrt[6]{\frac{d_{curves}}{|\mathbb{F}|}}\right\}$.

Let $Q_1,\ldots,Q_l:\mathbb{F}^m\to\mathbb{F}$ be $l\leq 18/(\frac{\delta}{2})^2=72/\delta^2$ polynomials of degree at most d_{poly} guaranteed for π and $\frac{\delta}{2}\geq 3\sqrt{\varepsilon_{m,d_{poly},\mathbb{F},\mathbb{K}}}$ in the soundness of $\texttt{Randomness-Efficient-Low-Degree-Tester}_{m,d,\mathbb{F},\mathbb{K}}$ (lemma 6.2).

Consider input $\vec{x}_1^1,\ldots,\vec{x}_k^1,\cdots,\vec{x}_1^N,\ldots,\vec{x}_k^N$ to $\texttt{Randomness-Efficient-Low-Degree-Reader}_{m,d,\mathbb{F},\mathbb{K},k,N}$. Let $\vec{x}_{k+1}^1,\ldots,\vec{x}_{k+1}^N$ be the corresponding points as being computed in step 1.

Claim 7.3.1. *For i_0 uniformly distributed in $[N]$, the probability – over i_0 and the randomness of the algorithm – that the algorithm does not reject in step 3, although $\pi(c(t))\notin\{Q_1(c(t)),\ldots,Q_l(c(t))\}$, is at most $\frac{\delta}{2}+e_{balance}+e_{tester}$.*

Proof. Denote the algorithm $\texttt{Randomness-Efficient-Low-Degree-Tester}_{m,d,\mathbb{F},\mathbb{K}}$ invoked with oracle access to π and $\{\pi_M\}_{M\in\mathcal{M}_{3,m,\mathbb{F}}}$ by T. We define an error function for T, denoted $\mu:\mathbb{F}^m\to[0,1]$, as follows: for every $\vec{x}\in\mathbb{F}^m$, let $\mu(\vec{x})$ be the probability over the randomness of T that T accepts when given input $\vec{x}$, although $\pi(\vec{x})\notin\{Q_1(\vec{x}),\ldots,Q_l(\vec{x})\}$.

By the choice of $Q_1,\ldots,Q_l$ and e_{tester}, we have $\frac{1}{|\mathbb{F}^m|}\sum_{\vec{x}\in\mathbb{F}^m}\mu(\vec{x})\leq\frac{\delta}{2}+e_{tester}$.

For every $\vec{x}\in\mathbb{F}^m$, let $\mathcal{D}(\vec{x})$ denote the probability, when picking $i\in[N]$ and $t\in\mathbb{F}-\{t_1,\ldots,t_k\}$ uniformly and independently at random, of getting $\vec{x}=c_{t_1,\ldots,t_{k+1},\vec{x}_1^i,\ldots,\vec{x}_{k+1}^i}(t)$. By lemma 7.1, $\sum_{\vec{x}\in\mathbb{F}^m}\left|\mathcal{D}(\vec{x})-\frac{1}{|\mathbb{F}^m|}\right|\leq e_{balance}$.

Therefore,

$$\begin{aligned}\sum_{\vec{x}\in\mathbb{F}^m}\mathcal{D}(\vec{x})\mu(\vec{x}) &= \sum_{\vec{x}\in\mathbb{F}^m}\frac{1}{|\mathbb{F}^m|}\mu(\vec{x})+\sum_{\vec{x}\in\mathbb{F}^m}\left(\mathcal{D}(\vec{x})-\frac{1}{|\mathbb{F}^m|}\right)\mu(\vec{x})\\ &\leq \frac{1}{|\mathbb{F}^m|}\sum_{\vec{x}\in\mathbb{F}^m}\mu(\vec{x})+\sum_{\vec{x}\in\mathbb{F}^m}\left|\mathcal{D}(\vec{x})-\frac{1}{|\mathbb{F}^m|}\right|\\ &\leq \frac{\delta}{2}+e_{tester}+e_{balance}\end{aligned}$$

By the design of the algorithm, when i_0 is uniformly and independently distributed in $[N]$, the probability that the algorithm accepts in step 3, although $\pi(c(t))\notin\{Q_1(c(t)),\ldots,Q_l(c(t))\}$ is $\sum_{\vec{x}\in\mathbb{F}^m}\mathcal{D}(\vec{x})\mu(\vec{x})\leq\frac{\delta}{2}+e_{balance}+e_{tester}$. ■[of claim 7.3.1]

For every $c\in\mathcal{C}_k^{m,\mathbb{F}}$, let $Q_{c,1},\ldots,Q_{c,l_c}:\mathbb{F}\to\mathbb{F}$ denote $l_c\leq 4/\delta$ polynomials of degree at most $(kd)^*kd\leq d_{curves}$ as guaranteed for π_c and $\frac{\delta}{2}\geq 2\sqrt{\frac{(kd)^*}{|\mathbb{F}|}}$ [recall that $\delta\geq 9\sqrt[6]{\frac{d_{curves}}{|\mathbb{F}|}}$ and $(kd)^*\leq d_{curves}$] by the soundness of the algorithm $\texttt{Low-Degree-Reader-For-Small-Dimension}_{1,kd,\mathbb{F},\mathbb{K},k+1}$ (lemma 5.3).

Assume on way of contradiction that the probability – over i_0 uniformly distributed in $[N]$ and over the randomness of the algorithm – that the algorithm outputs $a_1,\ldots,a_k$, such that there is no $1\leq i\leq l$ for which $a_1=Q_i(\vec{x}_1^{i_0}),\ldots,a_k=Q_i(\vec{x}_k^{i_0})$ is larger than $\delta+e_{balance}+e_{tester}+e_{reader}+e_{agree}$.

Then, by claim 7.3.1 and the soundness of $\texttt{Low-Degree-Reader-For-Small-Dimension}_{1,kd,\mathbb{F},\mathbb{K},k+1}$ in step 4, the probability – over i_0 and over the randomness of the algorithm – that the following events happen simultaneously is larger than e_{agree}:

1. The algorithm outputs $a_1,\ldots,a_k$, such that there is no $1\leq i\leq l$ for which $a_1=Q_i(\vec{x}_1^{i_0}),\ldots,a_k=Q_i(\vec{x}_k^{i_0})$.
2. $\pi(c(t))\in\{Q_1(c(t)),\ldots,Q_l(c(t))\}$.

3. There exists $1 \leq i \leq l_c$, for which $a_1 = Q_{c,i}(t_1), \ldots, a_k = Q_{c,i}(t_k), a_{k+1} = Q_{c,i}(t)$.

   In addition, when the above events happen, by the check in step 5 $[a_{k+1} = \pi(c(t))]$, we also have for the i from the previous item:

4. $Q_{c,i}(t) \in \{Q_1(c(t)), \ldots, Q_l(c(t))\}$.

Hence, there exists $i_0 \in [N]$, such that the probability over the randomness of the algorithm that the above events happen is larger than e_{agree}. Fixing i_0 fixes the choice of the curve c in step 1.

For $1 \leq i \leq l_c$, let us say that a polynomial $Q_{c,i}$ is *bad* if it is not one of $Q_{1|c}, \ldots, Q_{l|c}$.

By proposition 2.7, the degree of $Q_{1|c}, \ldots, Q_{l|c}$ is at most $kd_{poly} \leq d_{curves}$ [since $d^* \leq (kd)^*$ for $k \geq 1$]. The polynomials $Q_{c,1}, \ldots, Q_{c,l_c}$ are all of degree at most d_{curves}. Hence, by the Schwartz-Zippel lemma, for every $1 \leq i \leq l_c$ such that $Q_{c,i}$ is bad, for every $1 \leq j \leq l$, the polynomial $Q_{c,i}$ agrees with $Q_{j|c}$ on at most d_{curves} scalars in $\mathbb{F}$. Thus, there are at most $l_c \cdot l \cdot d_{curves} \leq \frac{4}{\delta} \cdot \frac{72}{\delta^2} \cdot d_{curves}$ scalars $t \in \mathbb{F} - \{t_1, \ldots, t_k\}$ for which there exists $1 \leq i \leq l_c$ such that $Q_{c,i}$ is bad, but $Q_{c,i}(t) \in \{Q_1(c(t)), \ldots, Q_l(c(t))\}$. Let us denote this bad set of scalars by B.

Hence, the probability that t chosen in step 2 uniformly at random in $\mathbb{F} - \{t_1, \ldots, t_k\}$ falls into B is at most $\frac{288}{\delta^3} \cdot \frac{d_{curves}}{|\mathbb{F}|-k} \leq \sqrt{\frac{d_{curves}}{|\mathbb{F}|}} = e_{agree}$ [where we use $k \leq |\mathbb{F}|/2$ and $\delta \geq 9\sqrt[6]{\frac{d_{curves}}{|\mathbb{F}|}}$].

Therefore there exists a fixing of $i_0 \in [N]$ and randomness for the algorithm for which $t \notin B$ and all the events above occur. In particular,

1. The algorithm outputs $a_1, \ldots, a_k$, such that there is no $1 \leq i \leq l$ for which $a_1 = Q_i(\vec{x}_1^{i_0}), \ldots, a_k = Q_i(\vec{x}_k^{i_0})$.

2. There exists $1 \leq i \leq l_c$, for which $a_1 = Q_{c,i}(t_1), \ldots, a_k = Q_{c,i}(t_k), a_{k+1} = Q_{c,i}(t)$.

3. For the i from the previous item, $Q_{c,i}(t) \in \{Q_1(c(t)), \ldots, Q_l(c(t))\}$, and hence (since $t \notin B$), $Q_{c,i}$ is not bad, i.e., there exists $1 \leq j \leq l$ such that $Q_{c,i}$ is $Q_{j|c}$.

But from items 2 and 3 we get that $a_1 = Q_{j|c}(t_1) = Q_j(c(t_1)) = Q_j(\vec{x}_1^{i_0}), \ldots, a_k = Q_{j|c}(t_k) = Q_j(c(t_k)) = Q_j(\vec{x}_k^{i_0})$, which is a contradiction to item 1. The lemma follows. ■

8 The Final Verifier

In this section we present the PCP verifier implying Theorem 2.

Theorem 2 (Main). *There exists a constant $0 < \alpha < 1$, for which there is a PCP verifier for checking satisfiability of Boolean formulas that on input of size n uses $\log n + O((\log n)^{1-\alpha})$ random bits to query 7 places in a proof of size $n \cdot 2^{O((\log n)^{1-\alpha})}$ over symbols consisting of $O((\log n)^{1-\alpha})$ bits. The verifier has perfect completeness and error $2^{-\Omega((\log n)^{\alpha})}$.*

The verifier uses the low degree reader from section 7.

8.1 Initial Verifier

Our starting point is the work of Dinur [8] that constructs a randomness-efficient PCP with *constant* error.

Theorem 5 ([8]). *There exist constants q and a, such that there is a PCP verifier for checking satisfiability of Boolean formulas of size n with size $s(n) = n \cdot \operatorname{poly}\log n$, randomness complexity $r(n) = \log n + O(\log\log n)$, query complexity $q(n) = q$, answer size $a(n) = a$, perfect completeness and error $\varepsilon(n) = \frac{1}{2}$.*

We can amplify the error of the PCP by enlarging its query complexity. This can be done at a reasonable cost in the randomness complexity via the use of randomness efficient *hitters* (cf. [10]).

Lemma 8.1 (randomness-efficient amplification). *If a language L has a PCP verifier with size $s : \mathbb{N} \to \mathbb{N}$, randomness complexity $r : \mathbb{N} \to \mathbb{N}$, query complexity $q : \mathbb{N} \to \mathbb{N}$, answer size $a : \mathbb{N} \to \mathbb{N}$, perfect completeness and error $\frac{1}{2}$, then, for any error $\varepsilon : \mathbb{N} \to (0,1)$, L has a PCP verifier with the same size s and the same answer size a, randomness complexity $r + O(\log \frac{1}{\varepsilon})$, query complexity $q \cdot O(\log \frac{1}{\varepsilon})$, perfect completeness and error ε. The running time of the verifier is polynomial in n and in $\log \frac{1}{\varepsilon}$.*

Proof. Let V be the PCP verifier for L ensured in the premise of the lemma. Consider some $\varepsilon : \mathbb{N} \to \mathbb{N}$, and let us describe a new PCP verifier V' for L with error at most ε.

As V, V' is given an input x of size n and an oracle access to a proof π of length $s(n)$ with answer size $a(n)$. V' invokes the efficient Expander-Walk Hitter described in [10], that uses $r(n) + O(\log \frac{1}{\varepsilon(n)})$ random bits to sample $k = O(\log \frac{1}{\varepsilon(n)})$ elements $w_1, \ldots, w_k \in \{0,1\}^{r(n)}$ in time polynomial in n and in $\log \frac{1}{\varepsilon(n)}$. Those elements have the property that for every set $A \subseteq \{0,1\}^{r(n)}$ whose size is at least half of the size of $\{0,1\}^{r(n)}$, at least one of the elements $w_1, \ldots, w_k$ hits A, with probability larger than $1 - \varepsilon(n)$, i.e., $\Pr\left[\bigwedge_{i=1}^{k} (w_i \notin A)\right] < \varepsilon(n)$.

V' then simulates V on random strings $w_1, \ldots, w_k$ and accepts if V accepts for every random string w_i for $i = 1, \ldots, k$.

If $x \in L$, then there exists a proof π that V always accepts, and hence, for this proof π, V' always accepts. Moreover, if $x \notin L$, then for any proof π, for at least half of the random strings in $\{0,1\}^{r(n)}$ V rejects, hence the probability that V' does not encounter a random string on which V rejects, is less than $\varepsilon(n)$.

Thus, L has a PCP verifier with the same size s and the same answer size a, randomness complexity $r + O(\log \frac{1}{\varepsilon})$, query complexity $q \cdot O(\log \frac{1}{\varepsilon})$, perfect completeness and error ε, whose running time is polynomial in n and in $\log \frac{1}{\varepsilon}$. ∎

Corollary 8.2 (initial verifier). *There exists a constant a, such that for any error $\varepsilon : \mathbb{N} \to (0,1)$, there is a PCP verifier for checking satisfiability of Boolean formulas of size n with size $s(n) = n \cdot \operatorname{poly}\log n$, randomness complexity $r(n) = \log n + O(\log\log n) + O(\log \frac{1}{\varepsilon(n)})$, query complexity $q(n) = O(\log \frac{1}{\varepsilon(n)})$, answer size $a(n) = a$, perfect completeness and error $\varepsilon(n)$. The running time of the verifier is polynomial in n and in $\log \frac{1}{\varepsilon(n)}$.*

8.2 Simulating The Initial Verifier

We now describe a PCP verifier for checking satisfiability of Boolean formulas that achieves sub-constant error by making only 7 queries to a proof of almost-linear size.

Specifically, for input size n, the verifier achieves error $\varepsilon \stackrel{def}{=} 2^{-(\log n)^{\frac{c}{2}}}$, for a constant $0 < c \leq \frac{2}{5}$ to be determined later. The verifier simulates the initial verifier from corollary 8.2 for error ε^3. This verifier makes a *non-constant* number of queries. To simulate it using a constant number of queries, the final verifier makes use of the low degree reader of section 7.

Let us denote the verifier of corollary 8.2 for error ε^3 by V. Let φ be an input of size n. Let s be the size of the proof that V queries. Let r be the number of random bits that V uses. Let q be the number of queries that V makes, and let a be the answer size of the proof. For $R \in \{0,1\}^r$, denote by $Q(\varphi, R, 1), \ldots, Q(\varphi, R, q) \in [s]$ the queries made by the verifier on input φ and randomness R. For answers $a_1, \ldots, a_q \in \{0,1\}^a$, denote by $V(\varphi, R, a_1, \ldots, a_q)$ the verdict of the verifier (*accept*/*reject*) when, on queries $Q(\varphi, R, 1), \ldots, Q(\varphi, R, q)$, the verifier receives answers $a_1, \ldots, a_q$.

The new verifier chooses appropriate finite field $\mathbb{F}$ and dimension m. It identifies the indices $\{1, \ldots, s\}$ of a proof for V with s points in $\mathbb{F}^m$ by choosing a set $H \subseteq \mathbb{F}$ of size $h \approx s^{\frac{1}{m}}$ and defining an injection $E : [s] \to H^m$. With the proof itself it identifies a low degree polynomial $Q : \mathbb{F}^m \to \mathbb{F}$ by identifying proof symbols in $\{0,1\}^a$ with field elements. Specifically, by multivariate interpolation as in lemma 2.6, a proof extends to a polynomial of degree at most $d \stackrel{def}{=} m(h-1)$.

Let us give a short overview of the new verifier. To simulate V, the new verifier chooses a random $R \in \{0,1\}^r$ and reads the value of a low degree polynomial (that supposedly encodes a proof for V) in the $k \stackrel{def}{=} q$ points corresponding to V's queries on randomness R, $E(Q(\varphi, R, 1)), \ldots, E(Q(\varphi, R, q))$. This is done using the low degree reader from section 7. If the verifier receives answers $a_1, \ldots, a_q$ that indeed correspond to elements of $\{0,1\}^a$, it decides according to $V(\varphi, R, a_1, \ldots, a_q)$.

For the low degree reader, the verifier needs to choose a finite field $\mathbb{F}$ that has a subfield $\mathbb{K}$. To this end, the verifier picks two natural numbers g_1, g_2 (of appropriate sizes) such that $g_1 | g_2$, and takes $\mathbb{F}$ to be $GF(2^{g_2})$ and $\mathbb{K}$ to be the subfield of $\mathbb{F}$ of order 2^{g_1}.

Computations in the field. Before formally presenting the new verifier, let us discuss the issue of efficient computation in the field and subfield.

To explicitly handle $\mathbb{F}$, we pick an irreducible polynomial $r(x) \in GF(2)[x]$ of degree g_2 (such exists and can be found in a preprocessing stage by exhaustive search in time polynomial in 2^{g_2}). We let $\mathbb{F} = GF(2)[x]/\langle r(x) \rangle$, or, equivalently, let $\mathbb{F}$ be the set of all formal polynomials in x of degree less than g_2 over $GF(2)$, where addition and multiplication are done modulo $r(x)$. Note that the field elements can be represented as binary strings of length g_2 and addition and multiplication can be done in time polynomial in g_2.

We can implement operations such as retrieval of the i'th element in the field or sampling of an element in the field via the representation of field elements as binary strings. We, however, take a different approach, which will allow us to implement these operations for the subfield as well.

A *primitive element* of a finite field $\mathbb{F}$ is an element $\alpha \in \mathbb{F}$, which is a generator of the cyclic multiplicative group $\mathbb{F} - \{0\}$, namely, $\alpha, \alpha^2, \alpha^3, \ldots$ are all the field elements but 0.

We pick a primitive element α of the field (such exists and can be found in a preprocessing stage in time polynomial in 2^{g_2}), and regard α^i as the i'th element of the field for $1 \leq i \leq 2^{g_2} - 1$. We consider the zero element of the field to be the 2^{g_2}'th element in it.

Proposition 8.3. *Let $\mathbb{F}$ be a finite field and let $\alpha \in \mathbb{F}$. Then, given natural $i \geq 1$, one can compute α^i in time polynomial in $\log i$ and in the time required for multiplication in the field.*

Proof. Given natural $i \geq 1$, compute $\alpha^{2^0}, \ldots, \alpha^{2^{\lfloor \log i \rfloor}}$ by repeated squaring starting from $\alpha^{2^0} = \alpha$. This can be done in time polynomial in $\log i$ and in the time required for multiplication in the field.

Now compute a binary representation of i, namely, $b_0, \ldots, b_{\lfloor \log i \rfloor} \in \{0,1\}$, such that $i = \sum_{j=0}^{\lfloor \log i \rfloor} b_j 2^j$. This can be done in time polynomial in $\log i$.

We have $\alpha^i = (\alpha^{2^0})^{b_0} \cdots (\alpha^{2^{\lfloor \log i \rfloor}})^{b_{\lfloor \log i \rfloor}}$. Computing this requires at most $\lfloor \log i \rfloor$ multiplications in the field. ■

Hence, retrieving the i'th element in the field, for every $1 \leq i \leq 2^{g_2}$ can be done in time polynomial in g_2.

Now, let us discuss the handling of the subfield. It is known from the theory of finite fields that the (unique) subfield of $\mathbb{F}$ of order 2^{g_1} is given by $\mathbb{K} \stackrel{def}{=} \left\{ x \in \mathbb{F} \;\middle|\; x^{2^{g_1}} = x \right\}$. Let $\beta = \alpha^{\frac{2^{g_2}-1}{2^{g_1}-1}}$. Note that since $g_1 | g_2$, the power $\frac{2^{g_2}-1}{2^{g_1}-1}$ is natural, and equals $1 + 2^{g_1} + (2^{g_1})^2 + \ldots + (2^{g_1})^{\frac{g_2}{g_1}-1}$, so $\beta \in \mathbb{F}$. Moreover, β is a primitive element of $\mathbb{K}$, because:

1. $\beta \in \mathbb{K}$, since $\beta^{2^{g_1}} = \beta^{2^{g_1}-1} \cdot \beta = \left(\alpha^{\frac{2^{g_2}-1}{2^{g_1}-1}} \right)^{2^{g_1}-1} \cdot \beta = \alpha^{2^{g_2}-1} \cdot \beta = \beta$.

2. $\beta^1, \beta^2, \ldots, \beta^{2^{g_1}-1}$ are all distinct elements, since for every $1 \leq i \leq 2^{g_1} - 1$, we have $\beta^i = \left(\alpha^{\frac{2^{g_2}-1}{2^{g_1}-1}} \right)^i = \alpha^{\frac{i}{2^{g_1}-1} \cdot 2^{g_2}-1}$, where $\frac{i}{2^{g_1}-1} \cdot (2^{g_2}-1)$ is a natural number between 1 and $2^{g_2} - 1$, and since $\alpha^1, \alpha^2, \ldots, \alpha^{2^{g_2}-1}$ are all distinct.

So we can also retrieve the i'th element of the subfield for every $1 \leq i \leq 2^{g_1}$, and sample an element in the subfield, in time polynomial in g_2 (again we use the convention that the 2^{g_1}'th element of the subfield is the zero element).

The Verifier

Given as input a Boolean formula φ of size n.

Parameter Setting and Preprocessing. Set the error to be $\varepsilon \stackrel{def}{=} 2^{-(\log n)^{\frac{c}{2}}}$, for a constant $0 < c \leq \frac{2}{5}$ to be determined later.

The rest of the parameters are set with respect to the parameters of the verifier V from corollary 8.2 for input size n and error ε^3, which are: size $s = n \cdot \operatorname{poly} \log n$, randomness $r = \log n + O((\log n)^{\frac{c}{2}})$, number of queries $q = O((\log n)^{\frac{c}{2}})$ and answer size $a = O(1)$.

$m \stackrel{def}{=} \max\left\{ \lceil (\log s)^{1-c} \rceil, 3 \right\}$; $h \stackrel{def}{=} \max\left\{ \lceil 2^{(\log s)^c} \rceil, 3 \right\}$; $d \stackrel{def}{=} m(h-1)$; $k \stackrel{def}{=} q$.

For the constants $0 < c_1, c_2 \leq 1$ and polynomials $p_1(\cdot,\cdot,\cdot), p_2(\cdot,\cdot,\cdot)$ from the guarantee on the soundness of the reader in section 7, set lower bounds on the size of the field and subfield:

$$K_0 \stackrel{def}{=} \max\left\{ 2, \left(\frac{p_1(m, \log d, \log k)}{\varepsilon} \right)^{\frac{1}{c_1}} \right\} \;\; ; \;\; F_0 \stackrel{def}{=} d \cdot \max\left\{ 2(k+1), 2^a, \left(\frac{p_2(m, \log d, k)}{\varepsilon} \right)^{\frac{1}{c_2}} \right\}$$

These are chosen with the intent of bounding the error terms from the low degree reading that depend on $\frac{1}{|\mathbb{K}|}$ and $\frac{d}{|\mathbb{F}|}$ by ε. Set $g_1 \stackrel{def}{=} \lceil \log K_0 \rceil$ and $g_2 \stackrel{def}{=} g_1 \cdot \lceil \frac{\log F_0}{g_1} \rceil$ [note that g_1 and g_2 are natural (non-zero) numbers satisfying $g_1 | g_2$]. Let $\mathbb{F} = GF(2^{g_2})$ and let $\mathbb{K} = \left\{ x \in \mathbb{F} \;\middle|\; x^{2^{g_1}} = x \right\}$ be the subfield of $\mathbb{F}$ of order 2^{g_1}. Find an irreducible polynomial of degree g_2 over $GF(2)$ and

a primitive element α of the field $\mathbb{F}$ (see the above discussion about computations in the field). Let $N \stackrel{def}{=} 2^r \cdot l$ for a multiplier $l = \lceil \frac{1}{2^r} |\mathbb{F}^m| / \varepsilon^2 \rceil$. This is chosen with the intent of bounding the error term from the low degree reading that depends on N by ε.

We have the following bounds:

- $m \leq (\log n)^{1-c} + O(\log \log n)$
- $h \leq 2^{(\log n)^c + O(\log \log n)}$
- $d \leq 2^{(\log n)^c + O(\log \log n)}$
- $k \leq O((\log n)^{\frac{c}{2}})$
- $|\mathbb{K}| \leq 2^{O((\log n)^{\frac{c}{2}})}$
- $|\mathbb{F}| \leq 2^{(\log n)^c + O((\log n)^{\frac{c}{2}})}$
- $|\mathbb{F}^m| \leq 2^{\log n + O((\log n)^{1-\frac{c}{2}})}$
- $N \leq 2^{\log n + O((\log n)^{1-\frac{c}{2}})}$

Mapping $[s]$ to $\mathbb{F}^m$. Let H be the first h elements in the field $\mathbb{F}$ [note the $|\mathbb{F}| \geq d \geq h$]. We define an injection $E : [s] \to H^m$ that is computable in time polynomial in $\log n$ as follows: For $i \in [s]$, consider a representation of $i-1$ in base h, $b_{m-1}, \ldots, b_0 \in \{0, \ldots, h-1\}$, where $i - 1 = \sum_{j=0}^{m-1} b_j h^j$ [note that since $h^m - 1 \geq s - 1$, such representation exists]. Let $E(i)$ be the m-tuple composed of the $(b_0 + 1)$'th element in the field, the $(b_1 + 1)$'th element in the field, etc. Note that E indeed maps $[s]$ to H^m.

- E **is injective:** since the representation in base h is unique and the ordering of the field elements is fixed.
- E **is computable in time polynomial in** $\log n$**:** given $i \in [s]$, computing the representation of $i - 1$ in base $h \geq 2$ can be done in time polynomial in $\log i$ and in $\log h$. Retrieving the appropriate m elements in the field can be done in time $\mathrm{poly}(m, \log |\mathbb{F}|)$. For our parameters, the computation can be done in time polynomial in $\log n$.

Proof Oracle. The verifier has oracle access to the following:

1. An oracle π which is a concatenation of all oracles required for the algorithm $\texttt{Randomness-Efficient-Low-Degree-Reader}_{m,d,\mathbb{F},\mathbb{K},k,N}$ [Note that the setting of parameters satisfies the requirements for the algorithm: $m \geq 3$; $4 \leq d \leq |\mathbb{F}|$; $\mathbb{F}$ is a finite field and $\mathbb{K} \subseteq \mathbb{F}$ is a subfield of it, such that all field and subfield operations can be done in time $\mathrm{poly} \log |\mathbb{F}|$; $k \geq 1$; $(k+1) \leq |\mathbb{F}|/2$; $N \geq |\mathbb{F}^m|$].

Process.

1. **Generate Collection of Queries.** For every $R \in \{0,1\}^r$, make l copies of the q-tuple $E(Q(\varphi, R, 1)), \ldots, E(Q(\varphi, R, q)) \in \mathbb{F}^m$ corresponding to the queries of the verifier V on input φ and randomness R. Denote by $\vec{x}_1^1, \ldots, \vec{x}_q^1, \ldots, \vec{x}_1^N, \ldots, \vec{x}_q^N$ the resulting $N = 2^r \cdot l$ tuples.

2. **Make Queries.** Pick uniformly and independently at random $R \in \{0,1\}^r$ and $i \in [l]$. Let $i_0 \in [N]$ index the i'th copy of the q-tuple associated with R, and invoke algorithm $\texttt{Randomness-Efficient-Low-Degree-Reader}_{m,d,\mathbb{F},\mathbb{K},k,N}$ on oracle access to π and on input $\vec{x}_1^1, \ldots, \vec{x}_q^1, \ldots, \vec{x}_1^N, \ldots, \vec{x}_q^N$ and i_0. If the algorithm rejects, *reject*. Otherwise, let $a_1, \ldots, a_q \in \mathbb{F}$ denote its output.

3. **Verify.** If at least one of $a_1, \ldots, a_q$ does not correspond to an element in $\{0,1\}^a$, *reject*. Otherwise, accept or reject as $V(\varphi, R, a_1, \ldots, a_q)$.

Running Time. Setting parameters, preprocessing: Setting the parameters can be done in time polynomial in $\log n$. Finding an irreducible polynomial and a primitive element for the field can be done in time polynomial in the size of the field [see the discussion regarding computations in the field]. Thus, this phase can be done in time polynomial in n.

Step 1: Generating the collection of queries can be done in time polynomial in N, the running time of V and the time needed to compute the mapping E on the q queries. Thus, this step can be done in time polynomial in n.

Step 2: Picking $R \in \{0,1\}^r$ and $i \in [l]$ can be done in time $\log N$. The running time of the algorithm $\texttt{Randomness-Efficient-Low-Degree-Reader}_{m,d,\mathbb{F},\mathbb{K},k,N}$ is polynomial in N. Thus, this step can also be done in time polynomial in n.

Step 3: Checking the q elements can be done in time polynomial in q and in $\log |\mathbb{F}|$. Simulating the verifier V can be done in time polynomial in n.

The total running time is polynomial in n.

Randomness. The algorithm requires $\log N = \log n + O((\log n)^{1-\frac{c}{2}})$ random bits for step 2.

The algorithm $\texttt{Randomness-Efficient-Low-Degree-Reader}_{m,d,\mathbb{F},\mathbb{K},k,N}$, requires $O((\log k + \log d)\log |\mathbb{F}|) + O((m + \log k + \log d)\log |\mathbb{K}|)$ random bits. For the above parameter setting, this becomes $O((\log n)^{1-\frac{c}{2}})$ [recall that $c \leq \frac{2}{5}$].

The total number of random bits the algorithm uses is $\log n + O((\log n)^{1-\frac{c}{2}})$.

Query Complexity. The query complexity of the verifier is the same as the query complexity of the algorithm $\texttt{Randomness-Efficient-Low-Degree-Reader}_{m,d,\mathbb{F},\mathbb{K},k,N}$, which is 7 queries.

Answer Size. The answer size of the proof oracle is the same as the answer size for the algorithm $\texttt{Randomness-Efficient-Low-Degree-Reader}_{m,d,\mathbb{F},\mathbb{K},k,N}$, which is $\text{poly}(\log d, k, \log |\mathbb{F}|)$. For the above parameter setting, this becomes $\text{poly}((\log n)^c)$.

Correctness.

Lemma 8.4 (Completeness). *If φ is satisfiable, then there exists π, such that the verifier accepts with probability* 1.

Proof. Assume that φ is satisfiable. Let $\pi_0 : [s] \to \{0,1\}^a$ denote a proof on which V accepts with probability 1. Let us define a function $f : H^m \to \mathbb{F}$ representing this proof, by letting, for every $i \in [s]$, $f(E(i)) = \pi_0(i)$ [recall that E is injective and that we identify elements in $\{0,1\}^a$ with field elements (note that indeed $g_2 \geq a$)]. Let f assign other elements in its domain arbitrary values in the field $\mathbb{F}$. Extend f into a polynomial $Q_f : \mathbb{F}^m \to \mathbb{F}$ of degree at most d as in lemma 2.6. Let π be the concatenation of all oracles guaranteed in the completeness of the low degree reader (lemma 7.2) for Q_f.

Fix some randomness for the verifier. This fixes the choice of the randomness R and the index i_0 in step 2. Denote the queries that V makes on randomness R by $i_1 = Q(\varphi, R, 1), \ldots, i_q = Q(\varphi, R, q) \in [s]$. Recall that $\vec{x}_1^{i_0} = E(i_1), \ldots, \vec{x}_q^{i_0} = E(i_q) \in H^m$ are the points corresponding to these queries.

By the choice of π, the verifier does not reject while reading, i.e., in step 2. Moreover, reading necessarily returns $a_1 = Q_f(\vec{x}_1^{i_0}), \ldots, a_q = Q_f(\vec{x}_q^{i_0})$, i.e., $a_1 = Q_f(E(i_1)), \ldots, a_q = Q_f(E(i_q))$. Since Q_f extends f and $E(i_1), \ldots, E(i_q) \in H^m$, we have $a_1 = f(E(i_1)), \ldots, a_q = f(E(i_q))$, and by the choice of f, we have $a_1 = \pi_0(i_1), \ldots, a_q = \pi_0(i_q)$. Hence, by the choice of the proof π_0, the elements $a_1, \ldots, a_q$ are all in $\{0,1\}^a$ and V necessarily accepts, and so does the new verifier. ■

Lemma 8.5 (Soundness). *If φ is not satisfiable, then for any π, the verifier accepts with probability at most $\varepsilon' = O(\varepsilon)$.*

Proof. Let π be such that on oracle access to π and input φ, the verifier accepts with probability larger than ε'. Let us construct a proof π_0 on which the verifier V accepts φ with probability at least $\Omega((\varepsilon' - 5\varepsilon)\varepsilon^2)$. For some $\varepsilon' = O(\varepsilon)$, this probability is larger than ε^3. By the soundness of V, having error at most ε^3, it follows that φ is satisfiable.

Note that by the choice of parameters

$$p_1(m, \log d, \log k) \left(\frac{1}{|\mathbb{K}|}\right)^{c_1} + p_2(m, \log d, k) \left(\frac{d}{|\mathbb{F}|}\right)^{c_2} \leq 2\varepsilon$$

Let $Q_1, \ldots, Q_l : \mathbb{F}^m \to \mathbb{F}$ be $l = O(1/\varepsilon^2)$ polynomials guaranteed for π and $\delta = 2\varepsilon$ in the soundness of the algorithm $\texttt{Randomness-Efficient-Low-Degree-Reader}_{m,d,\mathbb{F},\mathbb{K},k,N}$ (lemma 7.3).

For $j \in [l]$, let us denote by $Read_j$ the event that the low degree reading in step 2 results in answers agreeing with Q_j, namely, that the verifier does not reject in this step and $a_1 = Q_j(\vec{x}_1^{i_0}), \ldots, a_q = Q_j(\vec{x}_q^{i_0})$. Let us denote by $Read$ the event that at least one of $Read_1, \ldots, Read_l$ occurs. Let $Accept$ denote the event that the verifier accepts.

By our assumption on π and the choice of $Q_1, \ldots, Q_l$, noting that i_0 is uniformly distributed in $[N]$, the probability that both $Accept$ and $Read$ occur is larger than $\varepsilon' - (\delta + 2\varepsilon + \sqrt{|\mathbb{F}^m| / N}) \geq \varepsilon' - 5\varepsilon$. We also have

$$\Pr[Accept \wedge Read] \leq \sum_{j=1}^{l} \Pr[Accept \wedge Read_j]$$

Therefore, there exists $j_0 \in [l]$ such that

$$\Pr[Accept \wedge Read_{j_0}] \geq \frac{1}{l} \cdot \Pr[Accept \wedge Read] \geq \Omega((\varepsilon' - 5\varepsilon)\varepsilon^2) \tag{1}$$

Let us define π_0 such that for every $i \in [s]$, the symbol in the i'th position is $Q_{j_0}(E(i))$, if it corresponds to an element in $\{0,1\}^a$, and an arbitrary dummy symbol in $\{0,1\}^a$ otherwise.

By the construction of the verifier and (1), with probability at least $\Omega((\varepsilon' - 5\varepsilon)\varepsilon^2)$ over the choice of randomness $R \in \{0,1\}^r$, if we denote the positions $Q(\varphi, R, 1), \ldots, Q(\varphi, R, q)$ by $i_1, \ldots, i_q \in [s]$ and denote $a_1^{j_0} = Q_{j_0}(E(i_1)), \ldots, a_q^{j_0} = Q_{j_0}(E(i_q))$:

- $a_1^{j_0}, \ldots, a_q^{j_0}$ all correspond to elements in $\{0,1\}^a$. Hence, $\pi_0(i_1) = a_1^{j_0}, \ldots, \pi_0(i_q) = a_q^{j_0}$.
- $V(\varphi, R, a_1^{j_0}, \ldots, a_q^{j_0})$ accepts.

Therefore, the verifier V accepts when given oracle access to π_0 on input φ with probability at least $\Omega((\varepsilon' - 5\varepsilon)\varepsilon^2)$. The lemma follows. ■

Main Theorem Follows. Theorem 2 is now proven by choosing $0 < c \leq \frac{2}{5}$ to be a small enough constant so that the answer size becomes $O((\log n)^{1-\frac{c}{2}})$, and taking $\alpha = \frac{c}{2}$. The randomness of the verifier is $\log n + O((\log n)^{1-\alpha})$ and the verifier makes 7 queries to its proof. Thus, the size of the proof required for the verifier is at most $n \cdot 2^{O((\log n)^{1-\alpha})}$. The verifier has perfect completeness and its error is at most $\varepsilon' = 2^{-\Omega((\log n)^{\alpha})}$.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[2] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[3] S. Arora and M. Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003.

[4] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximations. In *Proc. 25th ACM Symp. on Theory of Computing*, pages 294–304, 1993.

[5] E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, and S. Vadhan. Robust PCPs of proximity, shorter PCPs and applications to coding. In *Proc. 36th ACM Symp. on Theory of Computing*, pages 1–10, 2004.

[6] E. Ben-Sasson and M. Sudan. Simple PCPs with poly-log rate and query complexity. In *Proc. 37th ACM Symp. on Theory of Computing*, pages 266–275, 2005.

[7] E. Ben-Sasson, M. Sudan, S. P. Vadhan, and A. Wigderson. Randomness-efficient low degree tests and short PCPs via epsilon-biased sets. In *Proc. 34th ACM Symp. on Theory of Computing*, pages 612–621, 2003.

[8] I. Dinur. The PCP theorem by gap amplification. In *Proc. 38th ACM Symp. on Theory of Computing*, pages 241–250, 2006.

[9] I. Dinur, E. Fischer, G. Kindler, R. Raz, and S. Safra. PCP characterizations of NP: Towards a polynomially-small error-probability. In *Proc. 31st ACM Symp. on Theory of Computing*, pages 29–40, 1999.

[10] O. Goldreich. *Lecture Notes for the Course Randomized Methods in Computation; Lecture 12, Hitters and Samplers, available at http://www.wisdom.weizmann.ac.il/∼oded/rnd.html*.

[11] O. Goldreich and M. Sudan. Locally testable codes and PCPs of almost-linear length. *Journal of the ACM*, 53(4):558–655, 2006.

[12] D. Moshkovitz and R. Raz. Sub-constant error low degree test of almost-linear size. In *Proc. 38th ACM Symp. on Theory of Computing*, pages 21–30, 2006.

[13] A. Polishchuk and D. A. Spielman. Nearly-linear size holographic proofs. In *Proc. 26th ACM Symp. on Theory of Computing*, pages 194–203, 1994.

[14] 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.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/