Electronic Colloquium on Computational Complexity, Report No. 119 (2010)

Distance Estimators with Sublogarithmic Number of Queries

Michal Moshkovitz *

July 24, 2010

Abstract

A distance estimator is a code together with a randomized algorithm. The algorithm approximates the distance of any word from the code by making a small number of queries to the word. One such example is the Reed-Muller code equipped with an appropriate algorithm. It has polynomial length, polylogarithmic alphabet size, and polylogarithmic number of queries. In our work we present two results. First, we construct a distance estimator with arbitrary small alphabet size, polynomial length, and polylogarithmic number of queries. Second, we construct a distance estimator with sublogarithmic number of queries, almost linear length, and polylogarithmic alphabet size.

Distance estimators are the coding theoretical analog of two-query low-error PCP. A recent work by Moshkovitz and Raz [FOCS'08] established two-query low-error PCP for the first time. In this work we examine whether we can construct a distance estimator via the new technique for PCP. Perhaps surprisingly, the new technique illuminates the difference between codes and PCP; there is an inherent problem with using the technique in the same way that was done for PCP. However, as we see in this work, the technique can be used to construct a distance estimator (up to a point).

To prove our results, we develop a general scheme for showing that a combinatorial operation preserves the distance estimator property.

*michali.mos@gmail.com. Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. Supported by the Israel Science Foundation, by the Wolfson Family Charitable Trust, and by a European Research Council (ERC) Starting Grant.

ISSN 1433-8092

Contents

1 Introduction

A *local algorithm* is an algorithm that given an input does not query the input entirely, and yet is able to return an approximate correct answer with high probability. Local algorithms can run in sublinear time. Hence, an extremely fast running time is possible, even for large input size. Constructing local algorithms received a lot of attention in the past few years. A few examples of known local algorithms that are related to codes/verification are:

1. locally testable codes (LTC) [12]: the algorithm *accepts* if the given word is in the code and *rejects* words that are far from the code.
2. locally decodable codes (LDC) [16]: given a corrupted codeword, the algorithm decodes any required symbol of the message.
3. probabilistically checkable proofs (PCP) [3, 2]: the verifier *accepts* if the given word is in the language and given a proof that is correct. It *rejects* words that are not in the language, no matter what the proof is.

In this paper we consider the natural question of constructing a code and a local algorithm that estimates the distance of any given word from the code. The Reed-Muller code is an example of a distance estimator. Each codeword in the Reed-Muller code is the truth table of a multivariate low degree polynomial. The code has polynomial length and polylogarithmic alphabet size. The *low degree theorem* (LDT for short; see [22, 4, 19]) proves that it is a distance estimator with polylogarithmic number of queries using the following local algorithm: given a truth table of an arbitrary function f, choose a random line and return the distance of the closest low degree polynomial to f on this line. The notion of a distance estimator is defined explicitly for the first time in this work.

1.1 PCP of Low Error and Distance Estimators

Locally Testable Codes (LTC) and Probabilistically Checkable Proofs (PCP) are closely related. In both, a local test is performed in order to answer a global question (is the word in the code or far from the code? is the proof correct or there is no correct proof?). Constructions of PCP usually yield LTC [12, 7, 8, 5]. More than that, in [5] it is proven that a stronger notion of PCP (called PCP of proximity or assignment testers) can be transformed into LTC. However, in general, the two notions are different. Notably, in PCP the algorithm may accept (with high probability) if there *exists* a correct proof even if the given proof is far from being correct, while in LTC the algorithm accepts with high probability only if *the given word* is close to a codeword.

The *error* of PCP is defined to be the probability that the algorithm accepts an input that is not in the language being decided, for the best proof. We want the error to be as low as possible. In a recent work [21] two-query PCP with low error was constructed. The analog of low-error two-query PCP for codes are distance estimators. One could hope that a distance estimator can be constructed from the PCP techniques, but this turn out to be more difficult than expected.

In this paragraph we describe some relevant aspects of the techniques of [21] with the simplification of [10] and explain the problem of constructing LTC using the same approach (in particular, we cannot construct a distance estimator, since distance estimator is a stronger notion). The core of the

technique of [21] is a new composition operation. It takes PCP with a large number of queries and produces PCP with a smaller number of queries. Analogously, in the coding language, it takes an algorithm with a large number of queries and a code with a large distance. It creates a new algorithm with a smaller number of queries and a new code with a large distance. The operation takes each codeword and divides it into overlapping blocks. Then, it encodes each block using some code with large distance. We do not describe the new algorithm, but only mention that the word generated from the following steps is accepted by the algorithm with high probability. Take a codeword, replace each block with a close string on this block, and encode it. This word can be proven to be far from the code. For PCP this is not a problem, since the algorithm may accept if there only *exists* a valid proof (and indeed there exists — the one we started with). On the other hand, this example shows that the algorithm and code resulted from the composition do not form LTC.

To the best of our knowledge, composition is the first operation with inherent difficulties translating from the PCP world to the codes world. For more on this subject see Section 1.4.

1.2 Our Results

Before formalizing our results we need some basic definitions and notations. A code is defined by an encoding function, $E : \Sigma^k \to \Sigma^n$ (we call k the *message length*). We sometimes refer to a code as the image of E. A local algorithm gets an oracle access to a word $w \in \Sigma^n$. We now list the standard parameters of a code and a local algorithm.

Code parameters:

(a) *length*: n. We want to minimize the length. Linear length, $n = O(k)$, is preferable.

(b) *(relative) distance*: the minimal fraction of symbols on which two codewords differ. It is denoted by $\triangle$. We wish to maximize this parameter. Close to 1 is preferable.

(c) *alphabet size*: $|\Sigma|$. We wish to minimize this parameter.

Algorithm parameters:

(a) *number of queries*: number of coordinates, of $\{1, \ldots, n\}$, that the algorithm queries. This parameter measures how local the algorithm is. We want to minimize this parameter.

(b) *error*: probability that the output of the algorithm deviates significantly from the distance of the word form the code, see Definition 1.1. We wish to minimize the error. Sub-constant is preferable.

The queries of a (non-adaptive) algorithm can be represented as a bipartite graph $G = (A, B, E)$ (with some ordering of the edges), where B corresponds to the n coordinates, A corresponds to the randomness of the algorithm, and an edge (a, b) corresponds to querying a coordinate b when the randomness is a. For a word $w \in \Sigma^n$ and a vertex $a \in A$ of degree $\deg(a)$, we denote by $w_a \in \Sigma^{\deg(a)}$ the word w restricted to the neighbors of a. Similarly, $C_a = \{c_a \,|\, c \in C\} \subseteq \Sigma^{\deg(a)}$ is the code C restricted to the neighbors of a.

In this work we examine the estimation version of the LTC problem. We want to locally find the distance of any given word from a code that we construct. We are mostly interested in estimating

the distance of words that are far from the code. The case of words that are close to the code is typically easier. This case was already been investigated quite a lot in the literature, see Section 1.3.

Instead of estimating the distance of a word from the code, it is more convenient for us to estimate a related notion — (relative) *agreement* (for words that are far from the code the distance is large and the agreement is low). For a word w and a code C we denote by $\mathrm{agr}(w, C)$ the fraction of coordinates on which w and its closest codeword in C agree on. Notice that the distance of the word w from the code C is equal to $1 - \mathrm{agr}(w, C)$.

A graph G gives a natural test: pick a random vertex $a \in A$ and return $\mathrm{agr}(w_a, C_a)$ (we call the last term the *local agreement*). This gives rise to the definition of a *tester*. A pair (C, G) of a code $C \subseteq \Sigma^n$ and a bipartite graph $G = (A, B, E)$ with $|B| = n$ is called a tester.

Next we formalize the definition of a *distance estimator*.

Definition 1.1. *A tester $(C \subseteq \Sigma^n, (A, B, E))$ is a (γ, β, M)-high probability distance estimator if for every word $w \in \Sigma^n$ it holds that*

$$\Pr_{a \in A}[\mathrm{agr}(w_a, C_a) > M \cdot \mathrm{agr}(w, C) + \gamma] \leq \beta.$$

In words, the probability that the output of the algorithm, $\mathrm{agr}(w_a, C_a)$, is more than an additive factor γ and a multiplicative factor M from the desired answer, $\mathrm{agr}(w, C)$, is at most β. We wish to minimize M, γ, and β; $M = 1$ is preferable. Notice that $0 \leq \gamma, \beta \leq 1$.

Notice that Definition 1.1 only bounds the probability that the output of the algorithm is much more than the correct answer (and not the probability that the output of the algorithm is much less than the correct answer). The reason is that the direction that appears in this definition is the harder one. As for the other direction, we can bound this probability more easily. We can do this in two ways: first, it is easy to prove that for regular graphs it holds that $\mathbb{E}_{a \in A}[\mathrm{agr}(w_a, C_a)] \geq \mathrm{agr}(w, C)$ (see Claim 2.3 for the proof); we can then use Markov's inequality to get the required bound. Second, if the underlying graph is a sampler (see Definition 2.4), then $\Pr_{a \in A}[\mathrm{agr}(w_a, C_a) < \mathrm{agr}(w, C) + \gamma] \leq \beta$, for small β, γ. In a future version we prove that the underlying graph that we construct is indeed a sampler.

In this work we construct two distance estimators. The first construction (in Theorem 1.2) is of a tester that the alphabet size can be made arbitrary small. The distance of the code is nearly optimal, the additive error is small, and the multiplicative error is 1. The downside is a somewhat large number of queries (at least polylogarithmic). The second construction (in Theorem 1.3) is of a tester with sublogarithmic number of queries and the code is of an almost linear length. However, this construction has a large multiplicative factor. For both of the results we start with the Reed-Muller code and perform a sequence of combinatorial operations on it.

Theorem 1.2. *For any $k \geq 1$ and $1/\log^{1/17} k < \varepsilon < 1$ there is an $(\varepsilon, \varepsilon, 1)$-high probability distance estimator with message length k, alphabet size $\mathrm{poly}(\frac{1}{\varepsilon})$, and distance $1 - O(\varepsilon^2)$. The other parameters are either (i) polynomial length and $\mathrm{poly}(\log k)$ number of queries, or (ii) almost linear length (i.e., $k^{1+o(1)}$) and $k^{o(1)}$ number of queries.*

Theorem 1.3. *For any $k \geq 1$ and any constant $0 < c < 1$ there exists $c' = O(c)$ such that for any $(\log k)^{-c'} < \varepsilon < 1$ there is an $(\varepsilon, \varepsilon, \mathrm{poly}(\frac{1}{\varepsilon}))$-high probability distance estimator with message length k, almost linear length, $\log^c k$ number of queries, $\mathrm{poly}(\log k)$ alphabet size, and distance $1 - 1/\mathrm{poly}(\log k)$.*

Notice that the second theorem is interesting only for words w that are far from the code (i.e., $\mathrm{agr}(w, C) \leq \mathrm{poly}(\varepsilon)$). The first theorem is interesting for a larger set of words: for all words w such that $\mathrm{agr}(w, C) \leq 1 - \varepsilon$.

1.3 Comparison with Previous Work

Reed-Muller Code. A code that was studied in the context of LTC is the Reed-Muller code, where the codewords correspond to polynomials of low degree compared to the field size. There are many ways to set the parameters of such a code while keeping its length polynomial (see Section 6 for details). One possibility is to have a polynomial length and $\mathrm{poly}(\log k)$ number of queries and alphabet size. Another possibility is to have an almost linear length ($k^{1+o(1)}$) and $k^{o(1)}$ number of queries and alphabet size. Using the low degree theorem (LDT) we know that in both cases the tester is a $(o(1), o(1), 1)$-distance estimator. Due to the large alphabet size a near optimal distance $(1 - o(1))$ is achievable. We emphasize that if we require the length to be polynomial, the number of queries and alphabet size must be at least polylogarithmic, no matter how we set the parameters.

In contrast to the large alphabet size in the Reed-Muller code, Theorem 1.2 gives an arbitrary small alphabet size at the price of increasing the additive error. The second result, Theorem 1.3, breaks the barrier of polylogarithmic number of queries of the Reed-Muller code by constructing a distance estimator with sublogarithmic number of queries, at the price of increasing the error.

A different variant of the Reed-Muller code is the case that the alphabet size is constant and the dimension and degree are large. This case is of less interest to us, since we require the distance of the code to be at least (some) constant and the length to be polynomial and these cannot be achieved for that variant of the Reed-Muller code.

Robust Locally Testable Codes. In the work of Ben-Sasson and Sudan [6] a construction of a distance estimator with the property $\mathbb{E}_{a \in A}[\triangle(w_a, C_a)] \geq \alpha \triangle(w, C)$ is described, where $\triangle$ is the relative distance and $0 < \alpha < 1$ is not arbitrarily close to 1. This notion is called Robust Locally Testable Codes to reflect the connection to a variant of PCP introduced in a prior work to theirs [5] called Robust PCP. In terms of agreement, they prove that $\mathbb{E}_{a \in A}[\mathrm{agr}(w_a, C_a)] \leq \alpha \cdot \mathrm{agr}(w, C) + (1 - \alpha)$. For words w that are extremely close to the code (namely, $\mathrm{agr}(w, C) > 1 - (\varepsilon/(1-\alpha))$) their result is better than Theorem 1.2. While, for the case that w is farther from the code, Theorem 1.2 gives a better result.

The number of queries in their construction is k^c for some constant $c < 1$. In both of our results the query size is much better. Namely, the number of queries in our construction is $k^{o(1)}$ or even sublogarithmic.

Dinur-Goldenberg [9] and Impagliazzo-Kabanets-Wigderson [15]. They construct a tester that test if a given word is far from the code. Their construction has two disadvantages. First, they are solving the decision problem (testing if the word is far) and not the estimation problem. Second, they work with a weaker notion of agreement (for example, two codewords can be different in every coordinate but still be considered close according to their definition). Their multiplicative factor is 1. The distance of their code is poor: polynomially small.

They had two results. First, the tradeoff between the alphabet and the error is good but then

if the error is sub-constant the length is not even polynomial. In contrast to the polynomial length in our construction, despite the small error. Their second result gives a polynomial length code but with large alphabet size compared to the error. In contrast to our almost linear length and inverse polynomial relation between error and alphabet. On the bright side, they query only 2 or 3 coordinates, while we query much more[1].

Tolerant Locally Testable Codes. In [14] the notion of *Tolerant Locally Testable Codes* was introduced. It is a strengthening of the decision problem (the LTC problem); the added requirement is that words that are close to the code should be accepted with high probability. (Notice that the notion of tolerant LTC is still a weaker notion than estimating the distance.) In [14] the authors modify the construction of [5, 6] to be tolerant LTC and not just LTC.

1.4 Proof Outline

To prove Theorems 1.2 and 1.3 we apply a series of combinatorial operations on the Reed-Muller code, along the lines of the recent low-error PCP construction of [21, 10]. The combinatorial operations we use are: query reduction, right degree reduction, and alphabet reduction (see Section 5 for the description of the operations). Next we describe how to (partly) solve the problem that was presented in Section 1.1 and highlights a difference between PCP and LTC.

Differences Between LTC and PCP: Query Reduction. We remind the reader that the query reduction operation (in the PCP world it is called *composition*) takes as an input a tester $(C, (A, B, E))$ and translates each codeword to a new codeword as follows. First, it divides the codeword into overlapping blocks. Then, it encodes each block using some inner code. As we already discussed in Section 1.1, in general the query reduction operation does not necessarily work for LTC. We gave there a word that is far from the code but it is accepted with high probability. To construct this word we used the fact that C_a have low distance for all $a \in A$. We overcome this difficulty by first recalling that the Reed-Muller code has a high local distance (i.e., for every $a \in A$, C_a has high distance). Second, we show that high local distance is enough to prove that the resulting tester (from the query reduction operation) is a distance estimator with a large multiplicative factor (recall that a small multiplicative factor is preferable).

It is more convenient to use a definition that encapsulates the two requirements: distance estimator and a high local distance. This notion was stated in the LDT literature [4, 22, 19] (see Section 3 for a proof of the equivalence). We call this property *locally list explainable tester* (LLE). Roughly speaking, a tester $(C, (A, B, E))$ is an LLE if for every word w there is a short list of codewords $L(w) \subseteq C$ that "explains" all somewhat close local codewords. I.e., for most $a \in A$ all local codewords $c^a \in C_a$ are either in $L(w)_a$ or $\mathrm{agr}(w_a, c^a)$ is small. It is clear that a tester that is an LLE does not have the problem presented in Section 1.1.

Our key point is the following. Consider a combinatorial operation that takes as input an LLE and returns a tester. If the combinatorial operation satisfies a certain condition (to be described below), then the returned tester is a distance estimator. In particular we can show that the query reduction operation and the other combinatorial operations we use satisfy the condition. However,

[1] For our stronger notion (i.e., distance estimator), the number of queries must depend on ε.

it seems inherent that the tester resulting from the query reduction operation is not an LLE. This means that we cannot use recursion as [10] do to reduce the query size from sublogarithmic to a constant, since the problem mentioned in Section 1.1 arises after one step.

Another difference between LTC and PCP is revealed in the construction of the inner code used in the query reduction operation. The inner code is required to be an LLE (though we allow poor parameters compared to our constructions in Theorems 1.2,1.3) and also must satisfy an additional property that we do not describe here. One might hope that we would be able to transform the Reed-Muller code to an inner code using the construction outlined in [10], but this is not the case. We devote Section 4 to the purpose of constructing an inner code.

Condition for Proving Correctness of a Combinatorial Operation. Each of the combinatorial operations that we consider takes as an input a tester $(C \subseteq \Sigma^{|B|}, G = (A, B, E))$ and outputs a new tester $(C' \subseteq \Delta^{|B'|}, (A', B', E'))$. An operation can change the tester (C, G) in many different ways. For example, replace each vertex in B with a few vertices in B' (e.g., this is how alphabet reduction and right degree reduction look). Another example is replacing the vertex set B with a few copies of the vertex set A (e.g., this is how query reduction looks). The combinatorial operation also translates each codeword in C to a unique codeword in C'. We denote the translation function by $Tran : C \to C'$. The inverse function of $Tran$ defines the source of any codeword in C'. We can also define the sources of a local codeword $c^{a'} \in C'_{a'}$ (for any $a' \in A'$) as the codewords that agree on this local codeword (i.e., all $c \in C$ such that $Tran(c)_{a'} = c^{a'}$). Since $c^{a'}$ is a *local* codeword, most likely there is more than one source. However, our combinatorial operations have the property that for every $a' \in A'$ there is a set of coordinates $S(a') \subseteq B$ such that the restriction of the sources to those coordinates is unique. I.e., for any two completions of $c^{a'}$ to global codewords $c^1, c^2 \in C'$, their sources have the same value in the coordinates of $S(a')$. We define the *source of a local codeword* $c^{a'}$ as the sources of $c^{a'}$ restricted to the coordinates in $S(a')$. For example, if the combinatorial operation replaces each coordinate $b \in B$ with s coordinates $b'_1, \ldots, b'_s$ then the sources of any local codeword that contains the s coordinates $b'_1, \ldots, b'_s$ have the same value in their b-th coordinate.

We are now ready to loosely state the condition: for every word $w' \in \Delta^{|B'|}$ and for every coordinate in B, there is a short list of symbols in Σ with the following property. For most $a' \in A'$, the source of every (somewhat close) local codeword is in the short list.

We can prove that if the latter condition holds then the resulting tester is a *punctured estimator* (see Definition 3.7); this is a weaker notion than LLE. In Lemma 3.8 we prove that any punctured estimator is a distance estimator with a large multiplicative factor. In this paper we also give a stronger condition such that any combinatorial operation satisfying it results with an LLE (and not just a punctured estimator). The alphabet reduction and the right degree reduction both satisfy this stronger condition. The query reduction operation does not satisfy this stronger condition.

This work also has the benefit of presenting a unified, and hopefully simpler, framework for all combinatorial operations. This might help to prove that other natural combinatorial operations (e.g., applying code repetition or the distance amplification of [1]) preserve the LLE property. We do not show the details of this proofs since our interests lie only in the three operations needed for our constructions.

1.5 Organization

In Section 2 we give the basic definitions and lemmas we use. In Section 3 we define different notions of distance estimators and discuss their relations. To achieve Theorem 1.3 we need an inner component that we construct in Section 4. In Section 5 we present combinatorial operations on testers and prove they preserve the distance estimator property. We put all the pieces together in Section 6. In that section we describe the processes that create the testers of Theorems 1.2 and 1.3 and analyze their parameters.

2 Preliminaries

General notation. Given a graph G and a vertex v in G, we denote by $\Gamma_G(v)$ the set that consists of all the neighbors of v. If the graph G is clear from the context we write $\Gamma(v)$. We say that G is *regular* if the degrees of all the vertices in G are equal. We say that a bipartite graph (A, B, E) is *right regular* if the degrees of all the vertices $a \in A$ are equal. We denote by $[k]$ the set $\{1, \ldots, k\}$. We denote by $\circ$ the string concatenation operation. Given a predicate P, we denote by I_P the indicator function for P, i.e., $I_P = 1$ if P is true and else $I_P = 0$. For a set Σ and an integer s we denote by $\binom{\Sigma}{\leq s}$ the set that contains all subsets of Σ of size at most s. For any set A we denote by $\mathcal{P}(A)$ the power set of A, i.e., the set that contains all the subsets of A. We use $\mathbb{F}$ to denote a finite field and $\deg p$ to denote the total degree of a polynomial p. For a vector $v \in \mathbb{F}^m$, a coordinate $i \in \{1, \ldots, m\}$, and $a \in \mathbb{F}$ we denote by $(v_1, \ldots, v_m)|_{i \to a}$ the vector $(v_1, \ldots, v_{i-1}, a, v_{i+1}, \ldots, v_m)$. In Sections 4 and 6, we use the symbol $\delta_{m,d,k,f}$ to denote the value $384 \cdot m \left(\sqrt[8]{\frac{1}{k}} + \sqrt[4]{\frac{md}{f}} \right)$, for integers $m, d, k \neq 0$, and $f \neq 0$.

Codes and agreement. A *word* w is a tuple over an alphabet Σ, i.e., $w \in \Sigma^n$ for some integer $n \geq 1$. For a coordinate $i \in \{1, \ldots, n\}$, we denote by w_i the i th coordinate of w. A *code* C is a set of words, i.e., $C \subseteq \Sigma^n$. A word in C is called a *codeword*. The *message length* of a code $C \subseteq \Sigma^n$ is defined as $\log_{|\Sigma|} |C|$. For two words $u, w \in \Sigma^n$ we define their (relative) *agreement* as the fraction of coordinates in which u and v have the same symbol, i.e., $\mathrm{agr}(u, w) = \frac{1}{n} |\{i \,|\, u_i = w_i\}|$. We define the (relative) *distance* of a code C as $\triangle(C) = 1 - \max_{c_1 \neq c_2 \in C}(\mathrm{agr}(c_1, c_2))$. The closest codeword to w in C (ties are broken arbitrarily) is denoted by $\tau(w, C)$. We define the agreement of w with C as the agreement of w with the closest codeword in C, i.e., $\mathrm{agr}(w, C) = \mathrm{agr}(w, \tau(w, C))$. We say that a code $C \subseteq \Sigma^n$ is *systematic* for a set $C' \subseteq \Sigma^k$, if for every $c \in C$ there is a unique $c' \in C'$ such that $c = c' \circ u$ for some $u \in \Sigma^{n-k}$.

Extended words. We say that a tuple W is an s-extended word, for some integer s, if $W \in \binom{\Sigma}{\leq s}^n$, i.e., each coordinate W_i in W is a set of at most s symbols from Σ. We, naturally, identify every word $w \in \Sigma^n$ with the 1-extended word $(\{w_i\})_{i=1}^n$. We can alternatively define an s-extended word using s words: W is *created from a set of s words* $\{w^1, \ldots, w^s\}$ (and denoted as $W =< w_1, \ldots, w_s >$) if for every $i \in \{1, \ldots, n\}$, $W_i = \{w_i^1, \ldots, w_i^s\}$. Notice that an extended word can be created from many different sets.

We can extend the definition of agreement to extended words in the following way: for two extended words $W, U \in \mathcal{P}(\Sigma)^n$ we define their agreement as $\mathrm{agr}(W, U) = \frac{1}{n} |\{i \,|\, W_i \cap U_i \neq \phi\}|$. (For

a word $w \in \Sigma^n$ and an extended word $U \in \mathcal{P}(\Sigma)^n$, $\mathrm{agr}(w, U)$ is well-defined since w is identified with a 1-extended word.) Mind the fact that for two words $u, w \in \Sigma^n$, if $\mathrm{agr}(w, u) = 1$ then $w = u$, while for two extended words, this is false. Notice that for an extended word $W =< w^1, \ldots, w^s >$, for some words $w^1, \ldots, w^s \in \Sigma^n$, it holds that for any extended word $U \in \mathcal{P}(\Sigma)^n$,

$$\mathrm{agr}(W, U) \leq \sum_{i \in [s]} \mathrm{agr}(w^i, U). \tag{1}$$

For a code C and an extended word $W \in \mathcal{P}(\Sigma)^n$ we define the closest codeword in C to W (denoted as $\tau(W, C)$) and the agreement of W with C (denoted as $\mathrm{agr}(W, C)$) similarly to the case where W is a word. As an aside note, we remark that the problem of list-recovery [13] can be formulated, using our terminology, as the problem of finding all close codewords in C to a given s-extended word.

For a code C and an integer s we denote by $C^{\odot s}$ the set of all s-extended words created by any set of at most s codewords in C. I.e., $C^{\odot s} = \{< c^1, \ldots, c^s > \mid c^1, \ldots, c^s \in C\}$. It follows from Equation 1 that $\mathrm{agr}(W, C^{\odot s}) \leq s \cdot \mathrm{agr}(W, C)$.

For two extended words $W, U \in \mathcal{P}(\Sigma)^n$, we define the difference $V = U \setminus W$ as the extended word in which the i th coordinate is $V_i = U_i \setminus W_i$ (this operation will be used in Definition 3.7 and Section 5.4).

The next lemma is known as the *Johnson Bound* (for a proof, see, e.g., [10], Fact 5.5).

Lemma 2.1 (Johnson Bound)**.** *For a code C with distance $1 - \delta$, any $\eta \geq 2\sqrt{\delta}$, and any word w it holds that $|\{c \in C \mid \mathrm{agr}(w, c) > \eta\}| \leq 2/\eta$.*

Reed-Muller Code. The Reed Muller code, denoted as $RM^{\mathbb{F},m,d}$, is the evaluation, on $\mathbb{F}^m$, of all m-variate polynomials with degree at most d over a field $\mathbb{F}$. I.e., for $\mathbb{F}^m = \{a_1, \ldots, a_{|\mathbb{F}|^m}\}$

$$RM^{\mathbb{F},m,d} = \{p(a_1) \circ \ldots \circ p(a_{|\mathbb{F}|^m}) \mid \deg p \leq d\}.$$

The following lemma shows us that the Reed-Muller code has a large distance.

Lemma 2.2 (Schwartz-Zippel Lemma)**.** *For any m-variate polynomial p of degree at most d over a field $\mathbb{F}$, that is not the zero polynomial, it holds that*

$$\Pr_{x \in \mathbb{F}^m}(p(x) = 0) \leq \frac{d}{|\mathbb{F}|}.$$

The lemma can be easily proved by induction. An easy proof of a slightly weaker form of this lemma ($d/(|\mathbb{F}|-1)$ instead of $d/|\mathbb{F}|$), resembling a proof given in [18], appears in appendix A.

The code $RM^{\mathbb{F},m,m(|H|-1)}$ is *systematic* for $\mathbb{F}^{|H|^m}$, for any set $H \subseteq \mathbb{F}$, using the *low degree extension*. It shows a mapping of any $f : H^m \to \mathbb{F}$ to a (unique) m-variate polynomial with individual degree at most $|\mathbb{H}| - 1$, see Lemma 2.6 in [20].

Testers. A pair (C, G) where $C \subseteq \Sigma^n$ is a code of length n over alphabet Σ and $G = (A, B, E)$ is a bipartite graph with $|B| = n$ is called a *tester*. For $a \in A$ we call $\Gamma(a)$ the *window* a. We use the notation w_a to denote the restriction of a word $w \in \Sigma^n$ to the $\deg(a)$ coordinates that are the

neighbors of a. Similarly, we denote by $C_a = \{c_a \,|\, c \in C\} \subseteq \Sigma^{\deg(a)}$ the code C restricted to $\Gamma(a)$. We use the notation c^a for a local codeword, i.e., $c^a \in C_a$, for some $a \in A$.

Recall that our goal is to estimate $\mathrm{agr}(w, C)$ for any given word w. A natural way to do this is to take a tester and return $\mathrm{agr}(w_a, C_a)$ for a random $a \in A$. The next claim shows that this way we do not return less than $\mathrm{agr}(w, C)$.

Claim 2.3. *Assume that the graph of a tester $(C \subseteq \Sigma^n, G = (A, B, E))$ is regular. Then for all $w \in \Sigma^n$, it holds that*

$$\mathrm{agr}(w, C) \leq \mathbb{E}_{a \in A}[\mathrm{agr}(w_a, C_a)].$$

Proof. Denote by $c = \tau(w, C)$ the closest codeword in C to w. Then it holds that

$$\mathrm{agr}(w, C) = \mathrm{agr}(w, c) = \mathbb{E}_{a \in A, b \in \Gamma(a)}[I_{w_b = c_b}] \leq \mathbb{E}_{a \in A}[\mathrm{agr}(w_a, C_a)],$$

where the second equality follows from the fact that G is regular. □

Sampler Graphs. We say that a graph $G = (A, B, E)$ is a *sampler* if for every $T \subseteq B$ it holds that most vertices in A have roughly $\frac{|T|}{|B|}$ of their neighbors inside T. More formally,

Definition 2.4 (sampler)**.** *For an accuracy parameter $0 \leq \varepsilon \leq 1$ and a confidence parameter $0 \leq \varepsilon' \leq 1$, we say that a regular bipartite graph $G = (A, B, E)$ with left degree d_A is a $(\varepsilon, \varepsilon')$-*sampler *if for every $T \subseteq B$ it holds that,*

$$\Pr_{a \in A}\left(\left|\frac{|\Gamma(a) \cap T|}{d_A} - \frac{|T|}{|B|}\right| \geq \varepsilon\right) \leq \varepsilon'.$$

We also need to define a stronger notion of a sampler that will be helpful in Section 5. The stronger notion requires the maximum fraction of vertices in A that do not approximate T correctly to also depend multiplicatively on the size of T.

Definition 2.5 (strong-sampler)**.** *For an accuracy parameter $0 \leq \varepsilon \leq 1$ and a confidence parameter $0 \leq \varepsilon' \leq 1$, we say that a regular bipartite graph $G = (A, B, E)$ with left degree d_A is a $(\varepsilon, \varepsilon')$-*strong-sampler *if for every $T \subseteq B$ it holds that*

$$\Pr_{a \in A}\left(\left|\frac{|\Gamma(a) \cap T|}{d_A} - \frac{|T|}{|B|}\right| \geq \varepsilon\right) \leq \varepsilon' \frac{|T|}{|B|}.$$

To show that strong-samplers exist, we recall the definition of an *expander graph* and prove that an expander graph is a strong-sampler.

Definition 2.6 (expander)**.** *For a regular bipartite graph $G = (A, B, E)$ with left degree d_A, we say that G is a λ-expander if for every $S \subseteq A$ and $T \subseteq B$ it holds that*

$$\left|\frac{|E(S, T)|}{d_A\,|S|} - \frac{|T|}{|B|}\right| \leq \frac{\lambda}{d_A}\sqrt{\frac{|T|}{|S|}}.$$

Claim 2.7. *Assume that a regular graph $G = (A, B, E)$ is a λ-expander with left degree d_A, and right degree d_B. Then for any $\varepsilon > 0$, G is a $(\varepsilon, \frac{2\lambda^2}{\varepsilon^2 d_A d_B})$-strong-sampler.*

Proof. Let δ denote $\frac{|T|}{|B|}$. Define the set $S_1 \subseteq A$ to be all the vertices in A that have at least $(\delta+\varepsilon)d_A$ neighbors in T, and $S_2 \subseteq A$ to be all the vertices in A that have at most $(\delta-\varepsilon)d_A$ neighbors in T. Our goal is to prove that these sets are small.

To prove that S_1 is small, note that

$$|E(S_1,T)| \geq (\delta+\varepsilon)d_A\,|S_1|\,.$$

On the other hand, Definition 2.6 implies that

$$|E(S_1,T)| \leq \delta d_A\,|S_1| + \lambda\sqrt{|S_1|\,|T|}.$$

Since $\frac{1}{d_A} = \frac{|A|}{|B|d_B}$ we get that $|S_1| \leq \frac{\lambda^2|A|}{\varepsilon^2 d_A d_B}\delta$.

Similarly, one can prove that S_2 is small. □

There is a constant $\alpha < 1$ such that for any n, d there is a $\Theta(d^\alpha)$-expander (A, B, E) with $|A| = |B| = n$ and degree $\Theta(d)$; see, for example, [21, Lemma 5.3]. Using Claim 2.7 we have the following corollary.

Corollary 2.8. *There is a constant $\alpha < 1$ such that for any $\varepsilon > 0$ and for any two integers d, n there is a regular graph $G = (A, B, E)$ with $|A| = |B| = n$, degree $\Theta(d)$, and G is a $(\varepsilon, \Theta(\frac{1}{\varepsilon^2 d^{2(1-\alpha)}}))$-strong-sampler.*

3 Variants of Distance Estimators and their Relations

In this section we define the notion of a distance estimator, which is a weaker notion than high probability distance estimator (introduced in Section 1). We choose to use this weaker definition because one of the components we use (low degree theorem from [19]) uses distance estimator and not high probability distance estimator. We then introduce three more definitions (locally list explainable tester, extended locally list explainable tester, and punctured estimator) and show relations among all four definitions. It is important to note that we construct a locally list explainable tester (Definition 3.2) and punctured estimator (Definition 3.7) that imply the stronger notion of a high probability distance estimator (see Lemmas 3.3 and 3.8). In Figure 1 we summarize the relations. We finish this section with a related definition, called Locally Decodable List Explainable Tester (LDLE).

Definition 3.1 (distance estimator)**.** *We say that a tester (C, G), where $C \subseteq \Sigma^n$ and $G = (A, B, E)$ is a* (γ, M)-distance estimator *if for every $w \in \Sigma^n$ it holds that*

$$\mathbb{E}_{a\in A}[\mathrm{agr}(w_a, C_a)] \leq M \cdot \mathrm{agr}(w, C) + \gamma.$$

We would like γ, β to be close to 0, preferably sub-constant. If $M = 1$ we omit it and write γ-distance estimator.

We now state the second definition of distance estimator. The definition requires that all close local codewords come from a short global list. This notion appears implicitly in the LDT literature.

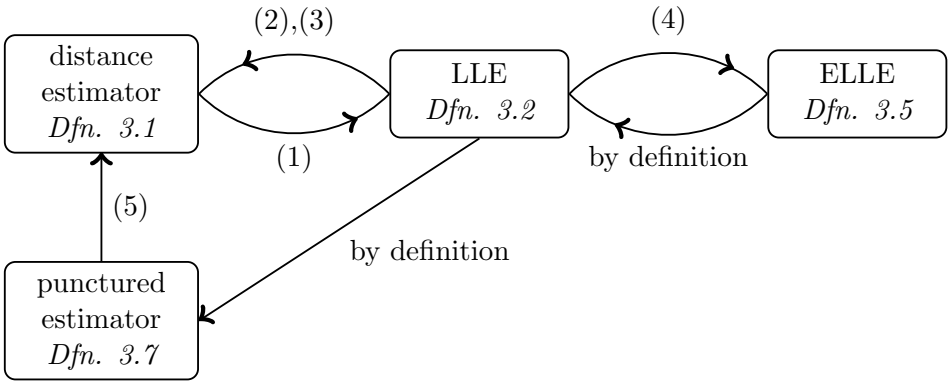

Figure 1: Four definitions of distance estimator and their relations. (1) Using Lemma 3.4 with the requirement that the local codes has distance close to 1. (2) Any (γ, L)-LLE is a (γ, L)-distance estimator; (3) if the graph is a sampler, it is a $(\gamma', 1)$-distance estimator (Lemma 3.3). (4) Any LLE is an ELLE, with slightly worse parameters (Lemma 3.6). (5) Any (γ, M)-punctured estimator is a (γ, M)-distance estimator (Lemma 3.8).

Definition 3.2 (locally list explainable tester)**.** *For any $0 \leq \gamma \leq 1$ we say that a tester (C, G), where $C \subseteq \Sigma^n$ and $G = (A, B, E)$ is a (γ, L)-*locally list explainable tester *(LLE) if for every word $w \in \Sigma^n$ there exists a list of codewords $L(w) \subseteq C$ of size $|L(w)| \leq L$ satisfying that*

$$\mathbb{E}_{a \in A}[\mathrm{agr}(w_a, C_a \setminus L(w)_a)] \leq \gamma.$$

The next two lemmas establish connections between the two definitions above. The proofs are adaptation of existing proofs in the LDT literature (e.g., [4, Theorems 6,7]) from the Reed-Muller code to general codes.

Lemma 3.3. *Let $(C \subseteq \Sigma^n, G = (A, B, E))$ be a (β, L)-LLE and assume that G is a $(\varepsilon, \varepsilon')$-sampler for some $\varepsilon, \varepsilon' > 0$. Then, (C, G) is a $(\beta + \varepsilon' L + \varepsilon)$-distance estimator.*

Proof. Fix an arbitrary word $w \in \Sigma^n$ and denote by $L(w) = \{c^1, \ldots, c^L\} \subseteq C$ the set of codewords given to us by the assumption that (C, G) is a (β, L)-LLE. Then, the following inequalities hold:

$$\begin{aligned}\mathbb{E}_{a \in A}[\mathrm{agr}(w_a, C_a)] &\leq \mathbb{E}_{a \in A}[\mathrm{agr}(w_a, C_a \setminus L(w)_a)] + \mathbb{E}_{a \in A}[\mathrm{agr}(w_a, L(w)_a)] \\ &\leq \beta + \mathbb{E}_{a \in A}[\mathrm{agr}(w_a, L(w)_a)],\end{aligned}$$

where the second inequality follows from our assumption on (C, G). To bound the latter expectation by $\mathrm{agr}(w, C) + \varepsilon' L + \varepsilon$ we use the assumption that G is a sampler. For each codeword c in $L(w)$ define the set $T = \{i \in B \mid w_i = c_i\}$ and notice that $\frac{|T|}{|B|} = \mathrm{agr}(w, c)$. By the definition of a sampler we get that for all but at most an ε' fraction of the vertices $a \in A$ it holds that $\mathrm{agr}(w_a, C_a) \leq \mathrm{agr}(w, c) + \varepsilon \leq \mathrm{agr}(w, C) + \varepsilon$. Hence for all but at most an $\varepsilon' L$ fraction of the vertices $a \in A$ it holds that $\mathrm{agr}(w_a, L(w)_a) \leq \mathrm{agr}(w, C) + \varepsilon$, and since $\mathrm{agr}(\cdot) \leq 1$ we are done. □

We note that using Markov's inequality, the last proof in fact proves that an LLE is a high probability distance estimator and not just a distance estimator. Notice that we can prove that any (β, L)-LLE is a (β, L)-distance estimator, without using any assumption on G.

Lemma 3.4. *Let (C,G), with $\triangle(C) \geq 1-\gamma^2/2$, be a (γ,M)-distance estimator and assume that for every $a \in A$ it holds that $\triangle(C_a) \geq 1-\frac{\gamma^2}{2}$. Then, (C,G) is a $(O(\gamma(M+1)),\frac{2}{\gamma})$-LLE.*

Proof. Let $w \in \Sigma^n$ be any word, and define $L(w) = \{c \in C \,|\, \mathrm{agr}(w,c) \geq \gamma\}$. Note that by the Johnson bound (Lemma 2.1) and the assumption that $\triangle(C) \geq 1-\frac{\gamma^2}{2}$ we get that $|L(w)| \leq \frac{2}{\gamma}$.

We now show the existence of a word u that is sufficiently far from the code, namely, $\mathrm{agr}(u,C) \leq 2\gamma$. Consider an arbitrary set of codewords $J \subseteq C$ of size $\frac{1}{\gamma}$ (if $|C| < \frac{1}{\gamma}$ the lemma follows immediately). Define the word $u \in \Sigma^n$ whose first γn coordinates are from the first codeword in J the next γn coordinates are from the second codeword in J, and so on. Clearly, for every codeword $c \in C$, $\mathrm{agr}(c,u) \leq \gamma + (1-\triangle(C))\frac{1}{\gamma} \leq 2\gamma$.

Denote by $L(w)_i$ the set of symbols $\{c_i \,|\, \exists c \in L(w)\}$. We define the word $w' \in \Sigma^n$ as

$$w'_i = \begin{cases} w_i & \text{if } w_i \notin L(w)_i \\ u_i & \text{else.} \end{cases}$$

It is now easy to see that for all $c \in C$, $\mathrm{agr}(w',C) \leq 3\gamma$: for $c \in L(w)$, $\mathrm{agr}(w',C) \leq \mathrm{agr}(u,C) \leq 2\gamma$ and for $c \in C \setminus L(w), \mathrm{agr}(w',c) \leq \mathrm{agr}(w,C) + \mathrm{agr}(u,C) \leq \gamma + 2\gamma$.

For every $a \in A$ and $c^a \in C_a \setminus L(w)_a$ it holds that

$$\begin{aligned} \mathrm{agr}(w_a, c^a) &\leq \Pr_{b \in \Gamma(a)}(w_b = c^a_b \wedge c^a_b \notin L(w)_b) + \Pr_{b \in \Gamma(a)}(c^a_b \in L(w)_b) \\ &\leq \Pr_{b \in \Gamma(a)}(w'_b = c^a_b) + \frac{2}{\gamma}(1-\triangle(C_a)) \\ &\leq \mathrm{agr}(w'_a, C_a) + \gamma. \end{aligned}$$

Hence, $\mathrm{agr}(w_a, C_a \setminus L(w)_a) \leq \mathrm{agr}(w'_a, C_a) + \gamma$. Taking expectation,

$$\mathbb{E}_a[\mathrm{agr}(w_a, C_a \setminus L(w)_a)] \leq \mathbb{E}_a[\mathrm{agr}(w'_a, C_a)] + \gamma \leq M\mathrm{agr}(w', C) + 2\gamma \leq 3\gamma M + 2\gamma,$$

where the second inequality is from the assumption that the tester (C,G) is a (γ,M)-distance estimator. □

We extend the definition of an LLE to the case that w is an extended word (i.e., each coordinate in w is a list), and the local codewords are lists too (i.e., the local code is $(C_a \setminus L(w)_a)^{\odot s}$).

Definition 3.5 (extended locally list explainable tester)**.** *We say that a tester (C,G), where $C \subseteq \Sigma^n$ and $G = (A,B,E)$ is a (γ, s_1, s_2, L)*-extended locally list explainable tester *(ELLE) if for every s_1-extended word W there exists a list of codewords $L(W) \subseteq C$ of size $|L(W)| \leq L$ satisfying that*

$$\mathbb{E}_{a \in A}\left[\mathrm{agr}\left(W_a, (C_a \setminus L(W)_a)^{\odot s_2}\right)\right] \leq \gamma.$$

Notice that any $(\gamma,1,1,L)$-ELLE is equivalent to a (γ,L)-LLE. Also notice that, by definition, any ELLE is also an LLE. The next lemma states that any LLE is also an ELLE, with slightly worse parameters.

Lemma 3.6. *Assume a tester $(C, G = (A,B,E))$ is a (γ,L)-LLE. Then, for every integers $s_1, s_2 > 0$, (C,G) is a $(\gamma s_1 s_2, s_1, s_2, s_1 L)$-ELLE.*

Proof. Fix an arbitrary s_1-extended word W. For $j = 1, \ldots, s_1$ define the *word* $w^j \in \Sigma^n$ by $w^j_b = (W_b)_j$. Here, $(W_b)_j$ denotes the j th symbol in the set W_b under some arbitrary order (we can assume without loss of generality that each W_b is of size exactly s_1). We define $L(W)$ to be $\cup_j L(w^j)$ and notice that $|L(W)| \leq s_1 L$. This completes the proof since

$$\begin{aligned}
\mathbb{E}_{a\in A}\left[\mathrm{agr}\left(W_a, (C_a \setminus L(W)_a)^{\odot s_2}\right)\right] &\leq s_2 \mathbb{E}_{a\in A}\left[\mathrm{agr}\left(W_a, C_a \setminus L(W)_a\right)\right] \\
&\leq s_2 \sum_{j=1}^{s_1} \mathbb{E}_a[\mathrm{agr}(w^j_a, C_a \setminus L(W)_a)] \\
&\leq s_2 \sum_{j=1}^{s_1} \mathbb{E}_a[\mathrm{agr}(w^j_a, C_a \setminus L(w^j)_a)] \\
&\leq s_1 s_2 \gamma,
\end{aligned}$$

where we used the simple fact stated as Equation 1. □

Next, we define a *punctured estimator*. This notion appeared (in different words) in the LDT literature. A tester is punctured estimator if for every word w there is a short list of codewords $L(w)$ that "cover" close local codewords. I.e., for every local codeword $c^a \in C_a$ that is close to w_a, the agreement of c^a with the extended word created by $L(w)_a$ is large. More formally,

Definition 3.7 (punctured estimator)**.** *We say that a tester (C, G), where $C \subseteq \Sigma^n$ and $G = (A, B, E)$ is a* (γ, M)-punctured estimator *if for every word $w \in \Sigma^n$ there exists an M-extended word c in $C^{\odot M}$ such that*

$$\mathbb{E}_{a\in A}[\mathrm{agr}(w_a \setminus c_a, C_a)] \leq \gamma,$$

where the minus sign denotes coordinate-wise set difference and we think of w_a as an extended word.

Notice that any (γ, L)-LLE, is also an (γ, L)-punctured estimator. The next lemma shows that any punctured estimator is also a distance estimator.

Lemma 3.8. *Assume a tester $(C, G = (A, B, E))$ with $C \subseteq \Sigma^n$ is right regular and a (γ, M)-punctured estimator. Then, (C, G) is a (γ, M)-distance estimator.*

Proof. Fix an arbitrary word $w \in \Sigma^n$. Denote by $c \in C^{\odot M}$ the M-extended word created from the M *codewords* $\{c^1, \ldots, c^M\}$ as promised to us by the definition of punctured estimator. For every $a \in A$ we know that

$$\begin{aligned}
\mathrm{agr}(w_a, C_a) &\leq \mathrm{agr}(w_a \setminus c_a, C_a) + \mathrm{agr}(w_a, c_a) \\
&\leq \mathrm{agr}(w_a \setminus c_a, C_a) + \sum_{j\in[M]} \mathrm{agr}(w_a, c^j_a).
\end{aligned}$$

From the assumption that G is right regular we know that $\mathbb{E}_{a\in A}[\mathrm{agr}(w_a, c^j_a)] = \mathrm{agr}(w, c^j) \leq \mathrm{agr}(w, C)$. Using the assumption that (C, G) is a punctured estimator the proof is completed. □

We note that using Markov's inequality, the last proof in fact proves that a punctured estimator is a high probability distance estimator and not just a distance estimator.

To prove theorem 1.3, we need an LLE that is able to decode. Our definition is related but stronger than the recently introduced notion of Locally Decode/Reject Code [21] and the closely related notion of dPCP [10]. Namely, our robustness property is stronger than [21, 10].

First, we extend the definition of a tester to the case there is an input to the local algorithm (recall that the graph of a tester is the queries of a local algorithm). I.e., a tester is $(C, \{G_i\}_{i\in[k]})$, where $[k]$ consists of all the possible values for the new input. We can now define a Locally Decodable List Explainable Tester (LDLE).

Definition 3.9 (Locally Decodable List Explainable Tester)**.** *A tester $(C, \{G_i = (A_i, B, E_i)\}_{i\in[k]})$ is a (γ, L)-LDLE for some set $C' \subseteq \Sigma^k$, if the following two conditions hold.*

1. *(decodability) C is systematic for C' and for every $i \in [k]$, $a \in A_i$ and for every two codewords $c, c' \in C$, if c and c' agree locally on a, i.e., $c_a = c'_a$, then these codewords also agree on the coordinate i, i.e., $c_i = c'_i$.*
2. *(robustness) for every word $w \in \Sigma^{|B|}$ there exists a short list of codewords $L(w) \subseteq C$, $|L(w)| \leq L$ such that*
$$\mathbb{E}_{i,a\in A_i}[\mathrm{agr}(w_a, C_a \setminus L(w)_a)] \leq \gamma.$$

In Section 4 we show how to construct an LDLE for every set for which we can check membership efficiently.

4 Construction of a Locally Decodable List Explainable Tester

In this section we construct an LDLE for every C to which we can check membership (see Theorem 4.2). For that, we recall the definition of a circuit as a model that allows us to check membership to a set.

Definition 4.1 (circuit)**.** *A circuit over a field $\mathbb{F}$ is an acyclic graph with in-degree 2 or 0 for each vertex. The vertices of in-degree 0 are called inputs, and are labeled with either a variable or a constant in $\mathbb{F}$, and vertices of in-degree 2 are labeled with $+$ (sum) or $\times$ (multiplication), and called gates. One special vertex is designated as the output vertex. The* size *of the circuit is the number of inputs and gates.*

We say that a circuit *checks membership* in a set $C \subseteq \mathbb{F}^k$ if for every input $x \in \mathbb{F}^k$, the output gate is 0 if and only if $x \in C$. For example, we can construct a polynomial size circuit to check if the input is an evaluation of a low degree polynomial.

Recall that an LDLE is a stronger notion than Locally Decode/Reject Code [21] and the closely related notion of dPCP [10]. To construct an LDLE we need to modify the construction outlined in [10]. They used the *sum check protocol* (see [17, 3] for more information); we instead use the *zero propagation test* from [5], as the sum check protocol seems not to apply in our setting.

Theorem 4.2. *Assume $C' \subseteq \mathbb{F}^k$ has a circuit of size S that checks membership in it and let $h, m > 1$ be such that $h^m \geq S + |\mathbb{F}|$ and $|\mathbb{F}| > (mh)^2$. Then, for any $\delta \geq \delta_{m,(mh)^2,|\mathbb{F}|,|\mathbb{F}|}$ there is an $(C, \{G_i\}_{i\in[k]})$*

that is $(O(\delta + {}^{(mh)^2}/_{|\mathbb{F}|\delta}), \frac{2}{\delta})$*-LDLE for* C'*. The length of* C *is* $|\mathbb{F}|^{4m+1}$*, the number of queries is at most* $|\mathbb{F}|^4$*, the alphabet is* $\mathbb{F}$*, and the distance is at least* $1 - O({}^{mh}/_{|\mathbb{F}|})$*.*

Recall that $\delta_{m,(mh)^2,|\mathbb{F}|,|\mathbb{F}|}$ was defined in Section 2.

4.1 Definitions

Individual degree, Curves, and Manifolds. For any polynomial p, we define $\deg_{\text{indv}}(p)$ as the maximal individual degree, over all variables in p. For example, $\deg_{\text{indv}}(x^2 + xyzw) = 2$.

A *curve* c is a function $c : \mathbb{F} \to \mathbb{F}^m$ such that $c(x) = (c_1(x), \ldots, c_m(x))$ for some m univariate polynomials $c_1, \ldots, c_m : \mathbb{F} \to \mathbb{F}$. We define the degree of a curve c as $\deg c = \max_i(\deg c_i)$. An easy fact, that follows from univariate interpolation, is that given any k different elements $t_1, \ldots, t_k \in \mathbb{F}$, and any k points $x_1, \ldots, x_k \in \mathbb{F}^m$ there exists a unique curve $c = c_{t_1,\ldots,t_k,x_1,\ldots,x_k}$ such that for every $i \in \{1, \ldots, k\}$, $c(t_i) = x_i$ and $\deg(c) \leq k - 1$; see [20, Proposition 2.8] for the proof. We define the *restriction* of a polynomial $p : \mathbb{F}^m \to \mathbb{F}$ to the curve $c : \mathbb{F} \to \mathbb{F}^m$ as the univariate polynomial $p|_c : \mathbb{F} \to \mathbb{F}$ that is defined by $p|_c(x) = p(c(x))$. Another easy fact is that $\deg(p|_c) \leq \deg(p)\deg(c)$ (see [20, Proposition 2.7] for the proof). A useful theorem on curves is the following.

Claim 4.3 ([20], Proposition 4.1)**.** *For different* $k + 1$ *scalars* $t_1, \ldots, t_{k+1} \in \mathbb{F}$*, and* k *points* $x_1, \ldots, x_k \in \mathbb{F}^m$*. Let* x_{k+1} *be a uniformly distributed point in* $\mathbb{F}^m$*. Then, for every* $t \in \mathbb{F} \setminus \{t_1, \ldots, t_k\}$*, the distribution* $c_{t_1,\ldots,t_{k+1},x_1,\ldots,x_{k+1}}(t)$ *is uniform in* $\mathbb{F}^m$*.*

A *manifold* υ is a function $\upsilon : \mathbb{F}^k \to \mathbb{F}^m$ with $\upsilon(x_1, \ldots, x_k) = (\upsilon_1(x_1, \ldots, x_k), \ldots, \upsilon_m(x_1, \ldots, x_k))$ for some m k-variate polynomials $\upsilon_1, \ldots, \upsilon_m : \mathbb{F}^k \to \mathbb{F}$. We define the degree of a manifold υ as $\deg \upsilon = \max_i(\deg \upsilon_i)$. For a polynomial $p : \mathbb{F}^m \to \mathbb{F}$ and a manifold $\upsilon : \mathbb{F}^k \to \mathbb{F}^m$ we define the restriction of p to the manifold υ as a multivariate polynomial $p|_\upsilon : \mathbb{F}^k \to \mathbb{F}$. It is defined by $p|_\upsilon(x_1, \ldots, x_k) = p(\upsilon(x_1, \ldots, x_k))$. Notice that for any polynomial p and a manifold υ, $\deg(p|_\upsilon) \leq \deg(p)\deg(\upsilon)$.

Checking Membership using Polynomials. We present an alternative definition for *checking membership*, using polynomials.

Definition 4.4 (checking membership using polynomials)**.** *Let* $\mathbb{F}$ *be a field and let* $C \subseteq \mathbb{F}^k$*. We say that a set of polynomials* P *over* $\ell \geq k$ *variables* checks membership *in* C *if for every* $a \in \mathbb{F}^k$ *it holds that* $a \in C$ *if and only if there exists (a unique)* $a' \in \mathbb{F}^{\ell-k}$ *such that* $a \circ a'$ *is a common zero of all polynomials in* P*.*

Every circuit can be transformed into a set of polynomials that have the latter property, as the next lemma shows. Notice that these polynomials are very local in the sense that each polynomial depends only on a constant number of variables.

Claim 4.5. *Let* $C \subseteq \mathbb{F}^k$ *be a set, and assume that there is a circuit of size* S *that checks membership in* C*. Then, there is a set of polynomials that check membership in* C *over* $|\mathbb{F}| + S$ *variables. Moreover, all polynomials are of the form* $y_{i_1} - y_{i_2}y_{i_3} - y_{i_4}$ *for some four variables* $y_{i_1}, y_{i_2}, y_{i_3}, y_{i_4}$ *with the exception of two polynomials. One being* y_0*, the other being* $y_1 - g$*, where* g *is a multiplicative generator of* $\mathbb{F}$*.*

Proof. First, we assign a new variable y_i to each element $g^i \in \mathbb{F}$ for $i \in \{1, \ldots, |\mathbb{F}| - 1\}$ and y_0 to 0. We also create a set of $|\mathbb{F}|$ polynomials such that in any common zero of these polynomials the value of y_i is its matched element in $\mathbb{F}$. The set of polynomials we create are: y_0, $y_1 - g$, and for $i \in \{2, \ldots, |\mathbb{F}| - 1\}$ the polynomial $y_i - y_{i-1}y_1 - y_0$.

Second, we assign a new variable to each gate. Now every gate and input is matched to a variable. Loosely speaking, two variables enter every gate. For every gate that is matched to the variable y_{i_1} and the variables that enter it are y_{i_2} and y_{i_3} we add a new polynomial. If this is a + gate, add the polynomial $y_{i_1} - y_{i_2}y_{|\mathbb{F}|-1} - y_{i_3}$ (recall that $y_{|\mathbb{F}|-1}$ is 1 for a common zero). Else, if this is a $\times$ gate, add the polynomial $y_{i_1} - y_{i_2}y_{i_3} - y_0$ (recall that y_0 is 0 for a common zero).

Third, if the variable matched with the output gate is y_{i_1}, add the polynomial $y_{i_1} - y_0y_0 - y_0$. □

Low Degree Theorem. We start with the fact that the Reed-Muller code is a distance estimator due to the well known Low Degree Theorem (LDT) (see [4, 19, 22]). Briefly, the LDT states that it is enough to query a 3-dimensional linear subspace to get a distance estimator. More formally,

Definition 4.6. *For any field $\mathbb{F}$ and a subfield $\mathbb{K} \subseteq \mathbb{F}$ we define the graph $G^{LDT,\mathbb{K}}$ as $(\mathbb{F}^m \times \mathbb{K}^m \times \mathbb{K}^m, \mathbb{F}^m, E)$ where $((x, y, z), w) \in E$ if and only if $w = t_1x + t_2y + t_3z$ for some $t_1 \in \mathbb{F} \setminus \{0\}, t_2, t_3 \in \mathbb{F}$.*

Lemma 4.7 (Low Degree Testing Theorem)**.** *For every subfield $\mathbb{K} \subseteq \mathbb{F}$ the tester $(RM^{\mathbb{F},m,d}, G^{LDT,\mathbb{K}})$ is a $\delta_{m,d,|\mathbb{K}|,|\mathbb{F}|}$-distance estimator.*

Recall that $\delta_{m,d,|\mathbb{K}|,|\mathbb{F}|}$ was defined in Section 2. To see that this lemma follows from [19, Theorem 2] note that the test in [19] is the same as choosing a random 3-dimensional linear sub-space and a point in it. Second, the tester in Lemma 4.7 includes all 3-dimensional sub-space the same number of times.

Using Lemma 3.4, the fact that $RM^{\mathbb{F},m,d}$ on any subspace is also a $RM^{\mathbb{F},m',d}$ for some m', and that for $\delta \geq \delta_{m,d,|\mathbb{K}|,|\mathbb{F}|}$ it holds that $\delta \geq \sqrt{2d/|\mathbb{F}|}$, we have the simple corollary that the latter is also an LLE.

Corollary 4.8 (Low Degree Testing Theorem-II)**.** *For every field $\mathbb{F}$, a subfield $\mathbb{K} \subseteq \mathbb{F}$, and $\delta \geq \delta_{m,d,|\mathbb{K}|,|\mathbb{F}|}$ the tester $(RM^{\mathbb{F},m,d}, G^{LDT,\mathbb{K}})$ is a $(O(\delta), \frac{2}{\delta})$-LLE.*

Notice that $G^{LDT,\mathbb{K}}$ is right regular with degree $|\mathbb{K}|^{2m} |\mathbb{F}|^2 (|\mathbb{F}| - 1)$. The reason is that for every $w \in \mathbb{F}^m$ its neighbors are $(w - t_2y - t_3z/t_1, y, z)$ for every $y, z \in \mathbb{K}^m, t_2, t_3 \in \mathbb{F}, t_1 \in \mathbb{F} \setminus \{0\}$.

Bundled Polynomial. We need a helpful claim that shows how to bundle a few low-degree polynomials into one low-degree polynomial. Specifically, we can bundle k^l polynomials of degree d into one polynomial of degree only $kl + d$ and the dimension increases by only l.

Claim 4.9. *For any field $\mathbb{F}$, a subset $H \subseteq \mathbb{F}$, and m-variate polynomials $p_1, \ldots, p_{|H^\ell|}$ over $\mathbb{F}$ of degree at most d, the following holds. There is an $(m + \ell)$-variate polynomial $Q_{p_1,\ldots,p_{|H^\ell|}}$ of degree at most $d + \ell(|H| - 1)$ such that for every $\vec{h} = (h_1, \ldots, h_\ell) \in H^\ell$ the following equality holds*

$$Q_{p_1,\ldots,p_{|H^\ell|}}(h_1, \ldots, h_\ell, x_1, \ldots, x_m) = p_{\vec{h}}(x_1, \ldots, x_m).$$

Proof. Let $h \in H$, and define

$$q_h(x) = \frac{\prod_{h' \in H \setminus \{h\}} (x - h')}{\prod_{h' \in H \setminus \{h\}} (h - h')}.$$

Notice that $q_h(x)$ is of degree $|H|-1$, is 1 on h and 0 on any point in $H \setminus \{h\}$. Let $\vec{h} = (h_1, \ldots, h_\ell) \in H^\ell$, and define the polynomial

$$q_{\vec{h}} = q_{h_1}(x_1) \cdots q_{h_\ell}(x_\ell).$$

Notice that the latter polynomial is of degree $\ell(|H|-1)$ and is 1 on $\vec{h}$ and 0 on any point in $H^\ell \setminus \{\vec{h}\}$. Finally, we define our desired Q as

$$Q_{p_1,\ldots,p_{|H^\ell|}}(h_1, \ldots, h_\ell, x_1, \ldots, x_m) = \sum_{\vec{h'} \in H^\ell} [q_{\vec{h'}}(h_1, \ldots, h_\ell) p_{\vec{h'}}(x_1, \ldots, x_m)].$$

□

Relative Locally Testable Code. We also need one last definition in which we weaken the definition of "locally testable codes". Instead of requiring to reject *all* far words, we require only to reject words that are in some set $\tilde{C}$. More formally,

Definition 4.10. *We say that a tester $(C, (A, B, E))$ is a γ-Locally Testable Code (LTC) relative to a set $\tilde{C}$ if for every $c \in \tilde{C} \setminus C$ it holds that $\Pr_{a \in A}(c_a \in C_a) \leq \gamma$.*

In the following two claims, we present simple testers that are LTC relative to the Reed-Muller code. Both are based on Schwartz-Zippel Lemma (see Lemma 2.2). First, define the set $Indv^{d,m,\mathbb{F}} = \{f : \mathbb{F}^m \to \mathbb{F} \mid \deg_{\text{indv}}(f) \leq d\}$.

Claim 4.11. *For any field $\mathbb{F}$ and integers $m, d' > d$, the following holds. There is a graph G, with left degree $m(d+2)$, such that the tester $(Indv^{d,m,\mathbb{F}}, G)$ is a $d'/|\mathbb{F}|$-LTC relative to $RM^{\mathbb{F},m,d'}$.*

Proof. We construct a graph G with left side $\mathbb{F}^{m+d+1}$, right side $\mathbb{F}^m$, and the neighbors of a vertex $(a_1, \ldots, a_{m-1}, a'_1, \ldots, a'_{d+2}) \in \mathbb{F}^{m+d+1}$ are defined in the following way. For every $j \in \{1, \ldots, m\}$ choose $d+2$ coordinates in $\mathbb{F}^m$ by taking the vector $(a_1, \ldots, a_{m-1}) \in \mathbb{F}^{m-1}$ and in the j th coordinate put one of the $d+2$ values $a'_1, \ldots, a'_{d+2}$. Namely,

$$\Gamma_G(a_1, \ldots, a_{m-1}, a'_1, \ldots, a'_{d+2}) = \{(a_1, \ldots, a_{j-1}, a'_{j'}, a_j \ldots, a_{m-1}) \mid j \in \{1, \ldots, m\}, j' \in \{1, \ldots, d+2\}\}.$$

Notice that the left degree of G is $m(d+2)$.

Fix an arbitrary polynomial $p \in RM^{\mathbb{F},m,d'} \setminus Indv^{d,m,\mathbb{F}}$. We know that there are $i \in \{1, \ldots, m\}$, $k > d$ and a $(m-1)$-variate polynomial $p_j : \mathbb{F}^{m-1} \to \mathbb{F}$ with $\deg(p_j) \leq d' - j$, for every $j \in \{0, \ldots, k\}$ such that

$$p(x_1, \ldots, x_m) = \sum_{j=0}^{k} x_i^j p_j(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m),$$

where p_k is not the zero polynomial. For every $a_1, \ldots, a_{m-1} \in \mathbb{F}$, define the univariate polynomial

$$p'_{a_1,\ldots,a_{m-1}}(x) = p(a_1, \ldots, a_{i-1}, x, a_i, \ldots, a_{m-1}).$$

Notice that $\deg(p'_{a_1,\ldots,a_{m-1}}) \leq k$. We bound the term $\Pr_{a\in|\mathbb{F}|^{m+d+1}}(p_a \in Indv_a^{d,m,\mathbb{F}})$ by

$$\begin{aligned}
&\Pr_a(p_k(a_1,\ldots,a_{m-1})=0) + \Pr_a(p_k(a_1,\ldots,a_{m-1})\neq 0 \wedge p_a \in Indv_a^{d,m,\mathbb{F}}) \\
&\leq \frac{d'-k}{|\mathbb{F}|} + \Pr_a(\deg(p'_{a_1,\ldots a_{m-1}})=k \wedge (p'_{a_1,\ldots,a_{m-1}}(a'_{j'}))_{j'=1}^{d+2} \in RM^{\mathbb{F},1,d}_{a'_1,\ldots,a'_{d+2}}) \\
&\leq \frac{d'-k}{|\mathbb{F}|} + \frac{k}{|\mathbb{F}|} = \frac{d'}{|\mathbb{F}|}.
\end{aligned}$$

□

For any field $\mathbb{F}$ and integers $m' > m$, we define $LessDim^{m,m',\mathbb{F}}$ as the set of all m'-variate polynomials that are influenced only by the first m variables. I.e., it is the set

$$\{f : \mathbb{F}^{m'} \to \mathbb{F} \mid \forall x_1,\ldots,x_{m'},y_{m+1},\ldots,y_{m'} \in \mathbb{F}, f(x_1,\ldots,x_{m'}) = f(x_1,\ldots,x_m,y_{m+1},\ldots,y_{m'})\}.$$

Notice that $LessDim^{m,m',\mathbb{F}}$ is *systematic* for $\mathbb{F}^{|\mathbb{F}|^m}$, by mapping any $f : \mathbb{F}^m \to \mathbb{F}$ to the m'-variate polynomial $f'(x_1,\ldots,x_{m'}) = f(x_1,\ldots,x_m) \in LessDim^{m,m',\mathbb{F}}$. In the following claim we prove that $LessDim^{m,m',\mathbb{F}}$ is an LTC relative to $RM^{\mathbb{F},m',d}$.

Claim 4.12. *For any field $\mathbb{F}$ and integers $d, m' > m$, the following holds. There is a graph G, with left degree 2, such that the tester $(LessDim^{m,m',\mathbb{F}}, G)$ is a $d/|\mathbb{F}|$-LTC relative to $RM^{\mathbb{F},m',d}$.*

Proof. We construct a graph G with left side $\mathbb{F}^{2m'-m}$, right side $\mathbb{F}^{m'}$, and the two neighbors of the vertex $(a_1,\ldots,a_{m'},a'_{m+1},\ldots,a'_{m'}) \in |\mathbb{F}|^{2m'-m}$ are

$$(a_1,\ldots,a_{m'}),(a_1,\ldots,a_m,a'_{m+1},\ldots,a'_{m'}).$$

Fix an arbitrary polynomial $p \in RM^{\mathbb{F},m',d} \setminus LessDim^{m,m',\mathbb{F}}$. We define the polynomial

$$p'(x_1,\ldots,x_{m'},y_{m+1},\ldots,y_{m'}) = p(x_1,\ldots,x_{m'}) - p(x_1,\ldots,x_m,y_{m+1},\ldots,y_{m'}).$$

Notice that $\deg(p') \leq \deg(p)$. Using the assumption that $p \notin LessDim^{m,m',\mathbb{F}}$, we have that p' is not the zero polynomial. Thus, from Schwartz-Zippel Lemma, we are done. □

4.2 The Construction

The proof of Theorem 4.2 is composed of two steps.

1. First, in Lemma 4.13 we construct, for any set $C' \subseteq \mathbb{F}^k$ that has polynomials that check membership in it over h^m variables, a tester (C, G') with the following properties. The code C is a subset of $RM^{\mathbb{F},4m+1,O(hm)}$ and systematic for C', the left degree of G is $O(mh)$, and the tester (C, G) is LTC relative to the RM code. This code is based on the zero propagation test [5].
2. Second, in Lemma 4.14 we assume a tester $(C \subseteq RM^{\mathbb{F},m,d}, G')$ is LTC relative to $RM^{\mathbb{F},m,d}$, systematic for $C' \subseteq \mathbb{F}^k$, and $|\mathbb{F}| > d_L(G')$ ($d_L(G)$ is the left degree of G'). Then, we construct graphs $\{G_i\}_{i\in[k]}$ such that $(C, \{G_i\}_{i\in[k]})$ is an LDLE for C'. This is a modification of Lemma 8.1 from [21].

Now we prove the first step in the construction of an LDLE.

Lemma 4.13. *For any field $\mathbb{F}$ and a subset $H \subseteq \mathbb{F}$ the following holds. Assume a set P of polynomials on $|H^m|$ variables, $m \leq (|\mathbb{F}|-2)/4$, each of the form $y_{i_1} - y_{i_2}y_{i_3} - y_{i_4}$, that check membership in a set $C' \subseteq \mathbb{F}^k$. Then, there is a tester $(C \subseteq RM^{\mathbb{F},4m+1,d}, G')$, with $d = O(m\,|H|)$, that is $O(d/|\mathbb{F}|)$-LTC relative to $RM^{\mathbb{F},4m+1,d}$, has left degree $O(m\,|H|)$, and C is systematic for C'.*

Proof. The motivation of the proof is the following. Recall that our goal is to show a subcode $C \subseteq RM$ systematic for C' and a graph (A', B, E) such that for any low degree polynomial $p \notin C$, $\Pr_{v\in A'}(p_v \in C_v)$ is small. First, for any $a \in \mathbb{F}^k$ we define a function h_a that is zero if and only if $a \in C'$. Then we use the low degree extension (see Section 2) to transform the problem to checking if a low-degree polynomial is zero on a subcube. Then, we use the zero-propagation test to create a short sequence of polynomials such that $a \in C'$ if and only if the last polynomial is the zero one. Our code is obtained by bundling all those polynomials, using Claim 4.9.

Identify the variables with elements of H^m. This allows us to think of any function $a : H^m \to \mathbb{F}$ as an assignment to the variables. We say that a function $a : H^m \to \mathbb{F}$ is a *satisfying assignment* if a is a common zero of P.

Denote by P^{all} the set of all the polynomials of the form $y_{i_1} - y_{i_2}y_{i_3} - y_{i_4}$ over the $|H^m|$ variables. Notice that any polynomial in P^{all} can be represented by a member in $(H^m)^4$ (corresponding to 4 variables). We denote by $I_P : H^{4m} \to \mathbb{F}$ the indicator function of P, mapping each polynomial in P^{all} to 0 or 1, depending on whether it is in P. We remark that sometimes we refer to an element in $\mathbb{F}^{4m}$ as four elements in $\mathbb{F}^m$ and sometimes as $4m$ elements in $\mathbb{F}$. When we write $\hat{x}$ we mean $\hat{x} \in \mathbb{F}^m$.

For every function $a : H^m \to \mathbb{F}$ define the function $h_a : H^{4m} \to \mathbb{F}$ in the following way,

$$h_a(\hat{x_1}, \hat{x_2}, \hat{x_3}, \hat{x_4}) = I_P(\hat{x_1}, \hat{x_2}, \hat{x_3}, \hat{x_4}) \cdot (a(\hat{x_1}) - a(\hat{x_2})a(\hat{x_3}) - a(\hat{x_4})). \tag{2}$$

Notice that for every $a : H^m \to \mathbb{F}$ it holds that a is a satisfying assignment if and only if h_a is the zero function (on H^{4m}).

The next step is to transform the function $I_P : H^{4m} \to \mathbb{F}$ to a low degree polynomial that agrees with I_P on H^{4m} (and similarly for a). We do this by using the *low degree extension* described in Section 2. We denote by $\bar{a} : \mathbb{F}^m \to \mathbb{F}, \bar{I_P} : \mathbb{F}^{4m} \to \mathbb{F}$ the low degree extensions of a and I_P with degrees $\deg(\bar{a}) \leq |H|\,m$ and $\deg(\bar{I_P}) \leq 4\,|H|\,m$. We define $\bar{h_a} : \mathbb{F}^{4m} \to \mathbb{F}$ as in Equation 2 with $\bar{I_P}$ and $\bar{a}$ instead of I_P and a. Note that $\deg(\bar{h_a}) \leq 6\,|H|\,m$. Notice that for every $a : H^m \to \mathbb{F}$ it holds that a is a satisfying assignment if and only if $\bar{h_a}$ is zero on H^{4m}.

Let π be an arbitrary bijection from H to $\{0, \ldots, |H|-1\}$. We define recursively $4m+1$ polynomials $p_0, \ldots, p_{4m} : \mathbb{F}^{4m} \to \mathbb{F}$ by $p_0 = \bar{h_a}$ and for $1 \leq i \leq 4m$ as

$$\begin{aligned} p_i(x_1, \ldots, x_{4m}) &= \sum_{a\in H} p_{i-1}(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_{4m}) x_i^{\pi(a)} \\ &= \sum_{a_1,\ldots,a_i \in H} x_1^{\pi(a_1)} \cdots x_i^{\pi(a_i)} \bar{h_a}(a_1, \ldots, a_i, x_{i+1}, \ldots, x_{4m}). \end{aligned}$$

Notice that for every $i \in \{0, \ldots, 4m\}$ it holds that $\deg p_i \leq 6\,|H|\,m + i(|H|-1)$, and a is a satisfying assignment if and only if p_{4m} is the zero polynomial.

The code. We map $a : H^m \to \mathbb{F}$, $a \in C'$, to the bundled polynomial of $p_0 = \bar{h_a}, p_1, \ldots, p_{4m} : \mathbb{F}^{4m} \to \mathbb{F}$ and $\bar{a} : \mathbb{F}^m \to \mathbb{F}$, using Claim 4.9. Here we extend $\bar{a}$ to a polynomial on $\mathbb{F}^{4m}$ that is in $LessDim^{m,4m,\mathbb{F}}$. Therefore, a is mapped to a $(4m+1)$-variate polynomial of degree at most $d = (6\,|H|\,m + 4\,|H|\,m) + (4m+1) = O(|H|\,m)$. Note that if we permute over the coordinates we get a systematic code for C'.

The graph. Fix an arbitrary polynomial $Q \in RM^{\mathbb{F},4m+1,d}$. We denote by $Q(j,\cdot) : \mathbb{F}^{4m} \to \mathbb{F}$, for some $j \in \mathbb{F}$, the polynomial Q with its first variable evaluated to j. We expand arbitrarily π to a bijection from $\mathbb{F}$ to $\{0,\ldots,|\mathbb{F}|-1\}$. For readability, when it is understood from the context, we omit π^{-1}.

We interpret $Q(4m+1,\cdot)$ as $\bar{a}$ and $Q(0,\cdot),\ldots,Q(4m,\cdot)$ as $p_0 = \bar{h_a}, p_1, \ldots, p_{4m}$. Notice that Q is in our code if and only if the following holds for any $x = (\hat{x_1},\hat{x_2},\hat{x_3},\hat{x_4}) \in \mathbb{F}^{4m}$,

$$
\begin{aligned}
Q(0,\cdot) &\in LessDim^{m,4m,\mathbb{F}} \\
Q(0,\cdot) &\in Indv^{|H|-1,4m,\mathbb{F}} \\
Q(0,x) &= \bar{I_P}(x)(Q(4m+1,\hat{x_1},\hat{0}^3) \\
&\quad - Q(4m+1,\hat{x_2},\hat{0}^3)Q(4m+1,\hat{x_3},\hat{0}^3) - Q(4m+1,\hat{x_4},\hat{0}^3)) \\
\forall\, 1 \le i \le 4m.\ Q(i,x) &= \sum_{a\in H} Q(i-1, x|_{i\to a}) x_i^{\pi(a)} \\
Q(4m,x) &= 0 \\
Q(x) &= Q_{Q(0,\cdot),\ldots,Q(4m+1,\cdot)}(x),
\end{aligned}
\tag{3}
$$

where $\hat{0}^3 = (\hat{0},\hat{0},\hat{0})$ and $Q_{Q(0,\cdot),\ldots,Q(4m+1,\cdot)}$ is the polynomial constructed in Claim 4.9. The first two conditions make sure that $Q(4m+1,\cdot)$ is indeed the low degree extension of some $a : H^m \to \mathbb{F}$. The first equality states that $Q(0,\cdot)$ is indeed $p_0 = \bar{h_a}$. The next $4m$ equalities make sure that $Q(1,\cdot),\ldots,Q(4m,\cdot)$ are $p_1,\ldots,p_{4m}$. The penultimate equality makes sure that $Q(4m,\cdot)$ is the zero function (i.e., a is a satisfying assignment). The last equality ensure us that Q is indeed the bundled polynomial, as defined in Claim 4.9.

Using Claims 4.11,4.12, and Schwartz-Zippel Lemma we can construct a tester that is $O(d/|\mathbb{F}|)$-LTC relative to Reed-Muller with left degree $O(m\,|H|)$.

□

Note that two of the polynomials promised to us in Lemma 4.5 are not of the form required for Lemma 4.13. We can easily fix this problem by increasing the left degree by only 2. Specifically, we query also the coordinates $(4m+1,\hat{y_0},\hat{0}^3),(4m+1,\hat{y_1},\hat{0}^3)$, where $\hat{y_0},\hat{y_1}$ are the appropriate values in H^m for the variables y_0, y_1. Now, for $p \in RM \setminus C$ one of the polynomials $\{y_0, y_1 - \gamma\} \cup P$ are not zero. It the polynomial that is not zero is y_0 or $y_1 - \gamma$ then for every $v \in A'$ it holds that $p_v \notin C_v$. Else, by Lemma 4.13, $\Pr_{v\in A'}(p_v \in C_v)$ is small.

Notice that we used the zero propagation test instead of the sum check protocol as in [10]. The latter requires bundling of *many* polynomials and therefore the last condition in Equation 3 cannot be tested with $O(m\,|H|)$ queries.

We continue with the second step in the construction of an LDLE. Recall that $\delta_{m,d,|\mathbb{F}|,|\mathbb{F}|}$ was defined in Section 2.

Lemma 4.14. *Assume $C \subseteq RM^{\mathbb{F},m,d}$ is a systematic code for $C' \subseteq \mathbb{F}^k$, and assume a tester $(C, G' = (A', B, E'))$, with left degree q where $(q+2)(d+3) < |\mathbb{F}|$, is γ-LTC relative to $RM^{\mathbb{F},m,d}$. Then, for any $\delta \geq \delta_{m,d(q+2),|\mathbb{F}|,|\mathbb{F}|}$ there exists a set of regular graphs $\{G_i = (A_i, B, E_i)\}_{i\in[k]}$ such that $(C, \{G_i\}_{i\in[k]})$ is a $(O(\delta + \gamma/\delta), \frac{2}{\delta})$-LDLE for C' with left degree at most $|\mathbb{F}|^4$ and $|A_i| = |A'|\,|\mathbb{F}|^{3m}$ for every i.*

Proof. We first describe the idea of the proof. Corollary 4.8 shows a construction of an LLE for the RM code that queries a random 3-dimensional subspace. This construction is, unfortunately, not an LDLE as above, for the following two reasons. First, the list guaranteed by the robustness condition contains Reed-Muller codewords but these are not necessarily codewords of C. Second, it does not decode. We reduce the first problem to the second one by using the assumption that we can locally check if a word is in the code. Hence, all we need is to decode while preserving the robustness condition.

To perform decoding we pass a random curve through the points we wish to decode, as in Claim 4.3. Then, by that claim, we know that each point on the curve (except the fixed ones) is uniform. Finally, through each point in the curve we query a random 3-dimensional subspace. This means we are choosing uniformly at random a 3-dimensional subspace and from Corollary 4.8 we are done.

We proceed with the formal proof. Define arbitrarily a bijection π from $\{0, \ldots, |\mathbb{F}| - 1\}$ to $\mathbb{F}$.

The Graphs. Fix an arbitrary index $i \in [k] \subseteq \mathbb{F}^m$. For every $a \in A', x \in \mathbb{F}^m$ denote by $c_{i,a,x}$ the unique curve of degree at most $q+1$ that when evaluated on $\pi(0), \ldots, \pi(q+1)$ attains $\Gamma_{G'}(a), i, x$, where the elements of $\Gamma_{G'}(a)$ appear in some fixed order. We denote the set $Q = \{\pi(0), \ldots, \pi(q+1)\}$. We define $A_i = A' \times \mathbb{F}^{3m}$. For any $a \in A', x, y, z \in \mathbb{F}^m$, consider the manifold $\mu_{i,a,x,y,z} : \mathbb{F}^4 \to \mathbb{F}^m$ of degree at most $q+2$,

$$\mu_{i,a,x,y,z}(t_0, t_1, t_2, t_3) = t_1 c_{i,a,x}(t_0) + t_2 y + t_3 z.$$

We define the multiset of neighbors of a vertex $(a, x, y, z) \in A_i$ as a subset of the latter manifold. Namely,

$$\Gamma_G(a, x, y, z) = \{\mu_{i,a,x,y,z}(t_0, t_1, t_2, t_3) \,|\, t_0 \in \mathbb{F} \setminus Q, t_1 \in \mathbb{F} \setminus \{0\}, t_2, t_3 \in \mathbb{F}\}.$$

Notice that $|\Gamma_G(a, x, y, z)| = (|\mathbb{F}| - (q+2))(|\mathbb{F}| - 1)\,|\mathbb{F}|^2$.

Decodability. We now prove that each $v = (a, x, y, z) \in A_i$ decodes $\Gamma_{G'}(a) \cup \{i\}$. For any two codewords $p, q \in C \subseteq RM^{\mathbb{F},m,d}$ recall that the (total) degree of the polynomials $p|_{\mu_{i,a,x,y,z}}, q|_{\mu_{i,a,x,y,z}} : \mathbb{F}^4 \to \mathbb{F}$ is bounded by $(q+2)d$. Using our assumption on $|\mathbb{F}|$ it follows that $|\mathbb{F}|^3\,(q+2)d < |\Gamma_G(v)|$. By the Schwartz-Zippel Lemma (see Lemma 2.2) if $p_v = q_v$ then $p|_{\mu_{i,a,x,y,z}}$ and $q|_{\mu_{i,a,x,y,z}}$ must agree on the entire $\mathbb{F}^4$, and in particular on $\Gamma_{G'}(a) \cup \{i\}$.

Since the code C is systematic for C' we achieve the decodability requirement of the LDLE definition (see Definition 3.9).

For any $v = (a, x, y, z) \in A_i$, let $c^1, c^2 \in C_v$. Thus, from the assumption that $C \subseteq RM^{\mathbb{F},m,d}$, we have that there exist two polynomials $p^1, p^2 \in RM^{\mathbb{F},m,d}$ such that $p^1_v = c^1$ and $p^2_v = c^2$. Recall that $\deg(p^1|_{\mu_{i,a,x,y,z}}), \deg(p^2|_{\mu_{i,a,x,y,z}}) \leq (q+2)d$. Therefore, since $|\Gamma_G(v)| = (|\mathbb{F}| - (q+2))(|\mathbb{F}| - 1)\,|\mathbb{F}|^2 \geq 1/2\,|\mathbb{F}|^4$, it holds that $\mathrm{agr}(p^1_v, p^2_v) \leq 2 \cdot \mathrm{agr}(p^1|_{\mu_{i,a,x,y,z}}, p^2|_{\mu_{i,a,x,y,z}}) \leq 2 \cdot {(q+2)d}/{|\mathbb{F}|}$. As a result, we have that $\triangle(C_v) \geq 1 - {2(q+2)d}/{|\mathbb{F}|}$.

Robustness. For every word $w \in \mathbb{F}^m$ let $L^{\mathbb{P}}(w) \subseteq RM^{\mathbb{F},m,d}$ be the set given by Corollary 4.8 satisfying $\left|L^{\mathbb{P}}(w)\right| \leq \frac{2}{\delta}$. We will prove that $L(w) = L^{\mathbb{P}}(w) \cap C$ fulfills the robustness property. Since

for any two sets A, B it holds that $\mathrm{agr}(w, A \cup B) \leq \mathrm{agr}(w, A) + \mathrm{agr}(w, B)$, we have that

$$\begin{aligned}\mathbb{E}_{v\in A_i}[\mathrm{agr}(w_v, (C_v \setminus L(w)_v))] \leq\ & \mathbb{E}_{v\in A_i}[\mathrm{agr}(w_v, C_v \setminus L^{\mathbb{P}}(w)_v)] \\ & + \mathbb{E}_{v\in A_i}[\mathrm{agr}(w_v, (C_v \cap L^{\mathbb{P}}(w)_v) \setminus L(w)_v)].\end{aligned}$$

To bound the first expectation we denote by $c^v = \tau(w_v, C_v \setminus L^{\mathbb{P}}(w)_v)$ and by $v(s_0)$, for $s_0 \in \mathbb{F} \setminus Q$, the multiset $\{\mu_{i,a,x,y,z}(s_0, t_1, t_2, t_3) \mid t_1 \in \mathbb{F} \setminus \{0\}, t_2, t_3 \in \mathbb{F}\}$. Notice that $(v(s_0))_{s_0\in\mathbb{F}\setminus Q}$ is a partition of $\Gamma_G(v)$ and all the parts have the same size. The following inequalities hold,

$$\begin{aligned}\mathbb{E}_{v\in A_i}[\mathrm{agr}(w_v, (C_v \setminus L^{\mathbb{P}}(w)_v)] &= \mathbb{E}_{v\in A_i}[\mathrm{agr}(w_v, c^v)] \\ &= \mathbb{E}_{v\in A_i, s_0\in\mathbb{F}\setminus Q}[\mathrm{agr}(w_{v(s_0)}, c^v_{v(s_0)})] \\ &= \mathbb{E}_{v\in A_i, s_0\in\mathbb{F}\setminus Q}[\mathrm{agr}(w_{v(s_0)}, c^v_{v(s_0)}) I_{c^v_{v(s_0)}\in L^{\mathbb{P}}(w)_{v(s_0)}}] \\ &\quad + \mathbb{E}_{v\in A_i, s_0\in\mathbb{F}\setminus Q}[\mathrm{agr}(w_{v(s_0)}, c^v_{v(s_0)}) I_{c^v_{v(s_0)}\notin L^{\mathbb{P}}(w)_{v(s_0)}}] \\ &\leq \Pr_{v\in A_i, s_0\in\mathbb{F}\setminus Q}(c^v_{v(s_0)} \in L^{\mathbb{P}}(w)_{v(s_0)}) \\ &\quad + \mathbb{E}_{v\in A_i, s_0\in\mathbb{F}\setminus Q}[\mathrm{agr}(w_{v(s_0)}, C_{v(s_0)} \setminus L^{\mathbb{P}}(w)_{v(s_0)})] \\ &\leq \frac{2}{\delta} \cdot \frac{2(q+2)d}{|\mathbb{F}|} + O(\delta) \leq O(\delta),\end{aligned}$$

where the penultimate inequality follows from the bound on $\triangle(C_v)$, Corollary 4.8, and Claim 4.3.

To bound the second expectation we use the following two facts. First, recall that for every window $v = (a, x, y, z)$ we decode $\Gamma_{G'}(a)$. Second, the tester (C, G') is γ-LTC relative to RM,

$$\begin{aligned}\mathbb{E}_{v\in A_i}[\mathrm{agr}(w_v, (C_v \cap L^{\mathbb{P}}(w)_v) \setminus L(w)_v)] &\leq \Pr_{v\in A_i}((C_v \cap L^{\mathbb{P}}(w)_v) \setminus L(w)_v \neq \phi) \\ &\leq \Pr_{v\in A_i}(\exists p \in L^{\mathbb{P}}(w) \setminus C \text{ such that } p_v \in C_v) \\ &\leq \sum_{p\in L^{\mathbb{P}}(w)\setminus C} \Pr_{v\in A_i}[p_v \in C_v] \\ &\leq \frac{2}{\delta}\gamma.\end{aligned}$$

Notice that we prove a stronger notion of robustness. Namely, we prove the robustness for *every* index i instead of the required expectation on i.

Right Regularity. In the same way we proved the right regularity of $G^{LDT,\mathbb{K}}$ we prove that the LDLE we constructed is right regular. □

Theorem 4.2 now follows from Lemmas 4.14, 4.13, and 4.5.

Note that the sizes of the left and right sides depend only on k and not on the set C'.

5 Combinatorial Operations on Distance Estimators

In this section we describe three combinatorial operations on distance estimators: alphabet reduction (Section 5.2), right degree reduction (Section 5.3), and query reduction (Section 5.4). All three operations take as input an ELLE, and improve one parameter of it (while only slightly effecting the other parameters) by somehow combining it with a "small" tester. The resulting tester is an LLE in the first two operations and a punctured estimator in the case of query reduction.

5.1 Summary of Parameters

In Table 5.1 we summarize the parameters of all three operations. For any $\varepsilon > 0$, we perform the operations on a (γ, s_1, s_2, L)-ELLE and as a result we get an LLE (or in the query reduction operation we get a punctured estimator) with error $\gamma + \varepsilon$. In all three operations the list size remains the same (the list size changes when using the reduction from LLE to ELLE, Lemma 3.6).

In this paragraph we discuss the parameters of the inner code/graph/tester used in the three operations. For the alphabet reduction operation we use the code described in Claim 5.4 with distance $1 - \delta_{in} \geq 1 - (\varepsilon/2)^2$ (note that we can use a code with larger distance). For the right degree reduction operation we use the graph described in Corollary 2.8 with degree $1/\varepsilon^{O(1)}$. For the query reduction operation, we use the LDLE constructed in Section 4 (see Theorem 4.2). Note that for the construction of the LDLE we assumed that the alphabet is a field and every local code has an arithmetic circuit that checks membership in it with $h^m - |\mathbb{F}|$ variables.

(γ, s_1, s_2, L)-ELLE		$(\gamma + \varepsilon, L)$-LLE		$(\gamma + \varepsilon, L)$-punctured estimator
Parameters	Notations	alpha. red	degree red.	query reduction
length	b	bk	bD_r	$a\,\lvert\Sigma\rvert^{4m+1}$
# queries	q	qk	qd	$\lvert\Sigma\rvert^4 D_r$
alphabet size	$\lvert\Sigma\rvert$	$1/\varepsilon^4$	$\lvert\Sigma\rvert$	$\lvert\Sigma\rvert$
distance	δ	$\delta(1-(\varepsilon/2)^2)$	δ	$\delta(1 - mh/\lvert\Sigma\rvert)$
left size	a	a	a	$b\,\lvert\Sigma\rvert^{O(m)}$
right degree	D_r	D_r	d	$q\,\lvert\Sigma\rvert^{O(m)}$
conditions		$k = \log\lvert\Sigma\rvert/\varepsilon^4$ $s_1 \geq 2/\varepsilon$ $\delta_{in} \leq (\varepsilon/2)^2$	$d = 1/\varepsilon^{O(1)}$ $s_1 \geq 3/\varepsilon$	$\lvert\Sigma\rvert \geq (mh/\varepsilon)^{O(1)}$ $s_1 \geq 8/\varepsilon^2$ $s_2 \geq 4/\varepsilon$

Table 1: Parameters of the combinatorial operations. Some $O(\cdot)$ notations are omitted. The first row indicates the error and list sizes of the input/output testers.

5.2 Alphabet Reduction

Let $C^{out} \subseteq \Sigma_{out}^n$ and $C^{in} \subseteq \Sigma_{in}^{n'}$ be two codes and assume there exists a bijection $E^{in} : \Sigma_{out} \to C^{in}$. We define the *concatenation* (see [11]) of the *outer code* C^{out} with the *inner code* C^{in} as

$$C^{out} \diamond C^{in} = \{E^{in}(c_1) \circ \cdots \circ E^{in}(c_n) \,|\, (c_1, \ldots, c_n) \in C^{out}\} \subseteq \Sigma_{in}^{nn'}.$$

Notice that $C^{out} \diamond C^{in}$ is a code over the alphabet of the inner code.

The alphabet reduction operation takes as an input a tester (C, G) and inner code C^{in} of block length k, and outputs the tester $(C \diamond C^{in}, G')$, each query in G is replaced by k queries to the encoded symbol. More formally,

Definition 5.1 (alphabet reduction)**.** *The tester obtained by applying the alphabet reduction operation to a tester $(C, (A, B, E))$ using an inner code C^{in} of length k is the tester $(C \diamond C^{in}, (A, B \times [k], E^{targ}))$ where $(a, (b, l)) \in E^{targ} \Leftrightarrow (a, b) \in E$.*

Proposition 5.2 (correctness)**.** *Let $(C, G = (A, B, E))$ be a $(\gamma, s, 1, L)$-ELLE and C^{in} be a code of distance $1 - \delta_{in}$, and assume that $s \geq \frac{1}{\sqrt{\delta_{in}}}$. Then the tester (C', G') obtained by performing alphabet reduction on C using C^{in} is a $(2\sqrt{\delta_{in}} + \gamma, L)$-LLE.*

Proof. Denote by E_{in} the encoding function of the inner code. For every $S \subseteq B$ we can define a partial encoding function $E_{in}(C_S)$ and decoding function $E_{in}^{-1}(C'_S)$ in the natural way (i.e., encoding or decoding each coordinate in B separately). Fix an arbitrary word $w' \in \Sigma_{targ}^{n'}$. For every $b \in B$ we define W_b to be the decoding of all close codewords of C_{in} to w'_b. Formally, $W_b = \{c \in E_{in}^{-1}(C^{in}) \,|\, \mathrm{agr}(w'_b, E_{in}(c)) > 2\sqrt{\delta_{in}}\}$. By Lemma 2.1 we know that $|W_b| \leq \frac{1}{\sqrt{\delta_{in}}} \leq s$. We define $L(w') = E(L(W))$, where $E : C \to C'$ is the encoding function.

For every $a \in A$, $c^a \in C'_a$, and $b \in \Gamma_G(a)$ we denote by c^a_b the block of coordinates that are in the coordinate b. The following inequalities holds,

$$\begin{aligned}
\mathrm{agr}(w'_a, c^a) &= \mathbb{E}_{b \in \Gamma_G(a)}[\mathrm{agr}(w'_b, c^a_b)] \\
&= \mathbb{E}_{b \in \Gamma_G(a)}[\mathrm{agr}(w'_b, c^a_b) \cdot I_{E_{in}^{-1}(c^a_b) \notin W_b}] + \mathbb{E}_{b \in \Gamma_G(a)}[\mathrm{agr}(w'_b, c^a_b) \cdot I_{E_{in}^{-1}(c^a_b) \in W_b}] \\
&\leq 2\sqrt{\delta_{in}} + \Pr_{b \in \Gamma_G(a)}(E_{in}^{-1}(c^a_b) \in W_b) \\
&= 2\sqrt{\delta_{in}} + \mathrm{agr}(W_a, E_{in}^{-1}(c^a)),
\end{aligned}$$

where the inequality follows from the definition of W_b and the fact that $\mathrm{agr}(\cdot)$ is at most 1. To complete the proof notice that if $c^a \in C'_a \setminus L(w')_a$ then $E_{in}^{-1}(c^a) \in C_a \setminus L(W)_a$. □

Claim 5.3 (sampler property)**.** *Assume the graph $G = (A, B, E)$ is a $(\varepsilon, \varepsilon')$-sampler. Then, the graph $G^{targ} = (A, B^{targ}, E^{targ})$ obtained by performing alphabet reduction on (C, G) for some code C is a $(2\varepsilon, \frac{\varepsilon'}{\varepsilon})$-sampler.*

Proof. Denote by k the length of the inner code. Denote by d_A the left degree of G. Let $T \subseteq B \times [k] = B^{targ}$. For every $1 \leq i \leq k$ define $T_i = T \cap (B \times \{i\})$, and define the set of vertices in A that estimate correctly T_i,

$$\mathrm{estimate}(i) = \left\{ v \in A \,|\, \left(\frac{|T_i|}{|B|} - \varepsilon\right) d_A \leq |\Gamma_G(v) \cap T_i| \leq \left(\frac{|T_i|}{|B|} + \varepsilon\right) d_A \right\}.$$

Also define the vertices that do not estimate correctly at least ε of the T_i's,

$$\text{bad-estimate}_\varepsilon = \{v \in A \,|\, \Pr_{i \in [k]}(v \notin \mathrm{estimate}(i)) \geq \varepsilon\}.$$

We denote by $d_A^{targ} = kd_A$ the left degree of G^{targ}. For $v \notin$ bad-estimate

$$|\Gamma_{G^{targ}}(v) \cap T| \leq \sum_{i=1}^{k} \left(\frac{|T_i|}{|B|} + \varepsilon \right) d_A + (\varepsilon k) d_A = \left(\frac{|T|}{|B^{targ}|} + 2\varepsilon \right) d_A^{targ}.$$

In the same way we can lower bound the left term.

From the sampler property of (A, B, E) we know that for every i, $\Pr_{v \in A}(v \notin \text{estimate}(i)) \leq \varepsilon'$, which mean that $|\text{bad-estimate}_\varepsilon| \, \varepsilon k \leq \varepsilon'(k \, |A|)$. Hence, $\Pr_{v \in A}(v \in \text{bad-estimate}) \leq \frac{\varepsilon'}{\varepsilon}$. □

As an inner code for the alphabet reduction operation we can use the code stated in the next claim. We use this code as it has large distance and short length. See Remark 5.6 in [10] for the proof.

Claim 5.4. *For any $\delta \in (0,1)$ and alphabet Σ, there is a code $C : \Sigma \to \sigma^k$, $|\sigma| = O(\frac{1}{\delta^2})$, $k = O(\frac{\log|\Sigma|}{\delta^2})$. This code has relative distance $1 - \delta$.*

We can now state the following corollary.

Corollary 5.5. *For any $\delta_{in} \in (0,1)$, $\varepsilon \geq 2\sqrt{\delta_{in}}$ assume a tester $T = (C \subseteq \Sigma^n, (A, B, E))$ is a $(\gamma, \frac{2}{\varepsilon}, 1, L)$-ELLE and the distance of C is $1 - \delta$. Then, the tester (C^{targ}, G^{targ}) obtained by applying the alphabet reduction operation to T is a $(\gamma + \varepsilon, L)$-LLE. Also, C^{targ} has distance $(1 - \delta_{in})(1 - \delta)$ and the alphabet size is $O(1/\delta_{in}^2)$. The length and the number of queries increased by a factor of $O(\log|\Sigma|/\delta_{in}^2)$.*

5.3 Right Degree Reduction

Given a code $C \subseteq \Sigma^n$ we define the k *repetition code* as the set

$$\text{rep}_k(C) = \{\overbrace{c \cdots c}^{k} \mid c \in C\} \subseteq \Sigma^{nk}.$$

The distance of C is equal to the distance of $\text{rep}_k(C)$.

The right degree reduction operation takes as an input a tester $(C, G = (A, B, E))$ with right degree d_B and uses a bipartite graph $G^{in} = (A^{in}, B^{in}, E^{in})$ with $|A^{in}| = d_B$, $|B^{in}| = k$, left degree d_A^{in} and right degree d_B^{in} to get the new tester $(\text{rep}_k(C), G')$ with right degree d_B^{in}. The graph G' is created by first defining for every $b \in B$ a matching between $\Gamma_G(b)$ and A^{in}. Then, each query in G is replaced by d_A^{in} queries according to G^{in}. More formally,

Definition 5.6 (right degree reduction)**.** *The tester obtained by applying right degree reduction operation to a tester $(C, G = (A, B, E))$ using a bipartite graph $G^{in} = (A^{in}, B^{in}, E^{in})$ with $|A^{in}| = d_B$ and $|B^{in}| = k$ is the tester $(rep_k(C), G^{targ} = (A, B \times B^{in}, E^{targ}))$ where $(a, (b, l)) \in E^{targ} \Leftrightarrow (a, b) \in E \wedge (\pi_b(a), l) \in G^{in}$ and π_b is a bijection between $\Gamma_G(b)$ and A^{in}. Each query is now replaced with d_A^{in} queries according to G^{in}.*

Proposition 5.7 (correctness)**.** *For any $0 < \alpha$, assume a tester $(C, G = (A, B, E))$ that is $(\gamma, s, 1, L)$-ELLE with $s \geq \frac{1}{\alpha}$. Then, after* right degree reduction *with $(\varepsilon, \varepsilon')$-strong-sampler the resulting tester $(C' \subseteq \Sigma^{n'}, G' = (A, B', E'))$ is a $(\alpha + \varepsilon' + \varepsilon + \gamma, L)$-LLE.*

Proof. Denote by Enc the encoding function of $\text{rep}_k(C)$. Fix an arbitrary word $w' \in \Sigma^{n'}$. We construct $\frac{1}{\alpha}$-extended word W, in the following way. For each $b \in B$ we define W_b to be all the α-popular symbols in w'_b. Formally, $W_b = \{\sigma \in \Sigma \mid \Pr_{i \in b}(w'_i = \sigma) \geq \alpha\}$. By a counting argument we get that $|W_b| \leq \frac{1}{\alpha} \leq s$. For every $a \in A$ and for every $c^a \in C_a$ and $b \in \Gamma_G(a)$ denote by c^a_b the coordinates of c^a that are in the block of b. Denote by $c^{a,b}$ their value for b. Denote by $w'_{a,b}$ the coordinates of block $b \in \Gamma_G(a)$. Notice that $Enc^{-1}(c^a)$ is defined and $Enc^{-1}(c^a)_b = c^{a,b}$. It holds that

$$\begin{aligned}
\mathbb{E}_a[\text{agr}(w'_a, c^a)] &= \mathbb{E}_{a,b\in\Gamma_G(a)}[\text{agr}(w'_b, c^a_b)] \\
&= \mathbb{E}_{a,b\in\Gamma_G(a)}[\text{agr}(w'_b, c^a_b) I_{c^{a,b}\notin W_b}] + \mathbb{E}_{a,b\in\Gamma_G(a)}[\text{agr}(w'_b, c^a_b) I_{c^{a,b}\in W_b}] \\
&\leq \alpha + \varepsilon + \varepsilon' + \mathbb{E}_{a,b\in\Gamma_G(a)}[\text{agr}(w'_b, c^a_b) I_{c^{a,b}\in W_b}] \\
&\leq \alpha + \varepsilon + \varepsilon' + \Pr_{a,b\in\Gamma_G(a)}(c^{a,b} \in W_b) \\
&= \alpha + \varepsilon + \varepsilon' + \text{agr}(W_a, Enc^{-1}(c^a)).
\end{aligned}$$

The first inequality follows from Claim 5.8 since we know that for every b and $\sigma \notin W_b$ if we denote by $T_\sigma = \{i \in b \mid w'_i = \sigma\}$ then $|T_\sigma| \leq d_A \alpha$ and for every $a \in A$ $\text{agr}(w'_b, c^a_b) = \left|\Gamma_{G^{targ}}(a) \cap T_{c^{a,b}}\right| / d_A$. We are done from the assumption that (C, G) is an ELLE. □

Claim 5.8. *For any $\alpha > 0$, assume a graph $G = (A, B, E)$ with left degree d_A is a $(\varepsilon, \varepsilon')$-strong-sampler. Also assume pairwise disjoint sets $\{T_i\}_i \subseteq B$ such that for every i it holds that $|T_i|/|B| \leq \alpha$. Then,*

$$\Pr_{a\in A}\left(\exists i. \frac{|\Gamma(a) \cap T_i|}{d_A} > \alpha + \varepsilon\right) \leq \varepsilon'.$$

Proof. For every set T_i define the set $A_i = \left\{a \in A \mid \frac{|\Gamma(a)\cap T_i|}{d_A} > \alpha + \varepsilon\right\}$, A_i contains the vertices in A that is connected to more than $(\alpha + \varepsilon)d_A$ vertices in T_i. Notice that

$$\sum_i \frac{|A_i|}{|A|} \leq \varepsilon' \sum_i \frac{|T_i|}{|B|} \leq \varepsilon'$$

where the first inequality follows from the assumption that G is a $(\varepsilon, \varepsilon')$-strong-sampler and the last inequality follows since $T'_i s$ are pairwise disjoint sets. □

5.4 Query Reduction

The query reduction operation, as the name suggests, takes as an input an ELLE tester $(C \subseteq \Sigma^n, G = (A, B, E))$ and constructs a punctured estimator with fewer queries (i.e., reduces the left degree of G). The operation uses for every $a \in A$ an LDLE for C_a with left side r^{in}, length k, and encoding function $Enc^{in,a}$. The resulting tester is $(C^{targ} \subseteq \Sigma^{n'}, G^{targ} = (B \times [r^{in}], A \times [k], E^{targ}))$. The code C^{targ} is created by taking every $c \in C$, encoding c_a according to the appropriate LDLE, for every $a \in A$, and concatenating all these encodings. I.e., $C^{targ} = \{Enc^{in,a_1}(c_{a_1}) \circ \ldots \circ Enc^{in,a_{|A|}}(c_{a_{|A|}}) \mid (c_{a_1}, \ldots, c_{a_{|A|}}) \in C\}$. The a th part of this concatenation is called the *block a*.

The neighbors of every $(b, r) \in B \times [r^{in}]$ in G^{targ} is a union of queries to a few blocks. In each block $a \in A$ we query according to the LDLE for C_a. This is done in order to guarantee that close

codewords to w'_a, for any word $w' \in \Sigma^{n'}$, be in the short list $L(w'_a)$— using the robustness property of an LDLE.

We want to be certain that all the short lists $L(w'_a)$ are a part of the same global short list $L(w') \subseteq C^{targ}$. We do this by showing that the lists $L(w'_a)$ are encoding of the short list $L(W)_a$, for an extended word $W \in \mathcal{P}(\Sigma)^n$. We set W_b to be the most frequent symbols among the multiset $\cup_{a\in\Gamma_G(b)} L(w'_a)_b$, where $L(w'_a)_b$ is the set of symbols $L(w'_a)$ assigns to b. We define the set of blocks that (b, r) reads to be $\Gamma_G(b)$. More formally,

Definition 5.9 (query reduction)**.** *Let $T = (C, G = (A, B, E))$ be a tester with left degree d_A. For every $a \in A$ let $T^{in,a} = (C^{in,a}, \{(A^{in}, B^{in}, E^{in,a}_b)\}_{b\in\Gamma_G(a)})$ be an LDLE for C_a with $\left|A^{in}\right| = r^{in}$, length $\left|B^{in}\right| = k$, and encoding function $Enc^{in,a}$. The tester T^{targ} obtained by applying query reduction operation to T using $(T^{in,a})_{a\in A}$ is defined as follows. $T^{targ} = (C^{targ}, (B \times [r^{in}], A \times [k], E^{targ}))$, where $((b, r), (a, l)) \in E^{targ} \Leftrightarrow (a, b) \in E \wedge (r, l) \in E^{in,a}_b$. The code is defined as $C^{targ} = \{Enc^{in,a_1}(c_{a_1}) \circ \ldots \circ Enc^{in,a_{|A|}}(c_{a_{|A|}}) \,|\, (c_{a_1}, \ldots, c_{a_{|A|}}) \in C\}$.*

Proposition 5.10 (correctness)**.** *Assume the following. A tester $T = (C, G = (A, B, E))$ is a (γ, s_1, s_2, L)-ELLE with $s_1 \geq \frac{L_{in}}{\gamma_{in}}$, and $s_2 \geq L_{in}$. The graph G is regular. For every $a \in A$, $T^{in,a} = (C^{in,a}, \{G^{in,a}_b\}_{b\in\Gamma_G(a)})$ is a (γ_{in}, L_{in})-LDLE for C_a. Then, the tester $(C^{targ} \subseteq \Sigma^{n'}, G^{targ} = (A^{targ}, B^{targ}, E^{targ}))$ obtained by applying* query reduction *to T using $(T^{in,a})_{a\in A}$ is a $(\gamma + 2\gamma_{in}, L)$-punctured estimator.*

Proof. Fix an arbitrary word $w' \in \Sigma^{n'}$ and define the following notations. For every $a \in A$,

- w'_a— the word w' restricted to block a.
- $w'_{b,r}$— the word w' restricted to the window (b, r).
- $(w'_{b,r})_a$— $w'_{b,r}$ restricted to block a (sometimes be written as $(w'_a)_{b,r}$).
- $L_{in}(w'_a)$— the promised list from the assumption that $T^{in,a}$ is an LDLE for C_a.
- $(L_{in}(w'_a))_b$— for $b \in \Gamma_G(a)$, the set of symbols $L(w'_a)$ assigns to b.
- $L(w'_a)_{b,r}$— for $b \in \Gamma_G(a), r \in [r^{in}]$, the set $L(w'_a)$, restricted to window (b, r).

The first three notations can be used also for any extended word $c \in \mathcal{P}(\Sigma)^{n'}$ and for the code C^{targ}. We denote by $Enc : C \to C^{targ}$ the encoding function from C to C^{targ}. Notice that for any completion of a local codeword $c^a \in C^{targ}_a$ to a global codeword $c \in C^{targ}$, $(Enc^{-1}(c))_a$ is the same. Thus, we allow ourselves to write $Enc^{-1}(c^a)$ for $c^a \in C^{targ}_a$.

We construct an s_1-extended word W in the following way. For each $b \in B$ define W_b to be the γ_{in}-popular symbols among $(L_{in}(w'_a))_b$, for $a \in \Gamma_G(b)$. Namely,

$$W_b = \{\sigma \in \Sigma \,|\, \Pr_{a\in\Gamma_G(b)}(\sigma \in (L_{in}(w'_a))_b) \geq \gamma_{in}\}.$$

By a counting argument we get that $|W_b| \leq \frac{L_{in}}{\gamma_{in}} \leq s_1$. Let $L(W)$ be the list defined by the assumption that (C, G) is an ELLE. Set $L(w')$ to be the set $Enc(L(W))$. Define $c \in (C^{targ})^{\odot L}$ to be the extended

word created from $L(w')$. We now want to show that this c fulfills the requirement of a punctured estimator.

We briefly describe the three steps of the proof. First, for a uniform random window, if a local codeword $c^{b,r} \in C^{targ}_{b,r}$ has high agreement with $w'_{b,r}$ (i.e., $\mathrm{agr}(w'_{b,r}, c^{b,r})$ is high), then for most $a \in \Gamma_G(b)$ it holds that $(c^{b,r})_a \in L(w'_a)_{b,r}$. The reason is that $c^{b,r}$ is close to $w'_{b,r}$ in blocks too (i.e., for most $a \in \Gamma_G(b)$, $\mathrm{agr}((w'_{b,r})_a, (c^{b,r})_a)$ is high) and we use the assumption that $T^{in,a}$ is an LDLE (second condition). Second, using the assumption that $T^{in,a}$ is an LDLE (first condition), any completion of $(c^{b,r})_a$ to a codeword in C^{targ}_a (and in particular to a codeword in $L(w'_a)$) gives a unique value and we can easily show that this value is in W_b. Third, the last step implies that W_a has high agreement with $Enc^{-1}(L(w'_a))$. Using the assumption that T is an ELLE, it follows that $(c^{b,r})_a \in (Enc(L(W))_{b,r})_a$. Notice that we only prove that the resulting tester is a punctured estimator and not LLE. This is due the fact that we prove that $(c^{b,r})_a \in (L(w')_{b,r})_a$, for most $a \in \Gamma_G(b)$, and not that $(c^{b,r}) \in L(w')_{b,r}$.

For the formal proof, note that, using the regularity of G, the following two distributions (that we define by a random process) are equal. First, choose uniformly at random $a \in A, b \in \Gamma_G(a)$. Second, choose uniformly at random $b \in B, a \in \Gamma_G(b)$. Denote by $c^{b,r} = \tau(w'_{b,r} \setminus c_{b,r}, C^{targ}_{b,r})$. We notice that for any $a \in \Gamma_G(b)$ if $c^{b,r}_a \in (L(w')_a)_{b,r}$ then $\mathrm{agr}((w'_{b,r})_a \setminus (c_{b,r})_a, (c^{b,r})_a) = 0$. Note that for any two completions of $c^{b,r}$ to two codewords c^1, c^2 it holds that $Enc^{-1}(c^1)_b = Enc^{-1}(c^2)_b$, by the definition of LDLE (first condition). We denote this value by $D_b(c^{b,r})$. By the definition of agreement, the regularity of G, and linearity of expectation, the following equalities holds,

$$\begin{aligned}
\mathbb{E}_{b,r}[\mathrm{agr}(w'_{b,r} \setminus c_{b,r}, C^{targ}_{b,r})] &= \mathbb{E}_{b,r}[\mathrm{agr}(w'_{b,r} \setminus c_{b,r}, c^{b,r})] \\
&= \mathbb{E}_{b,r,a\in\Gamma_G(b)}[\mathrm{agr}((w'_{b,r})_a \setminus (c_{b,r})_a, (c^{b,r})_a)] \\
&= \mathbb{E}_{a,r,b\in\Gamma_G(a)}[\mathrm{agr}((w'_{b,r})_a \setminus (c_{b,r})_a, (c^{b,r})_a)] \\
&= \mathbb{E}_{a,r,b\in\Gamma_G(a)}[\mathrm{agr}((w'_{b,r})_a) \setminus (c_{b,r})_a, (c^{b,r})_a) I_{(c^{b,r})_a \in L_{in}(w'_a)_{b,r} \wedge D_b(c^{b,r}) \in W_b}] \\
&\quad + \mathbb{E}_{a,r,b\in\Gamma_G(a)}[\mathrm{agr}((w'_{b,r})_a \setminus (c_{b,r})_a, (c^{b,r})_a) I_{(c^{b,r})_a \in L_{in}(w'_a)_{b,r} \wedge D_b(c^{b,r}) \notin W_b}] \\
&\quad + \mathbb{E}_{a,r,b\in\Gamma_G(a)}[\mathrm{agr}((w'_{b,r})_a \setminus (c_{b,r})_a, (c^{b,r})_a) I_{(c^{b,r})_a \notin L_{in}(w'_a)_{b,r}}].
\end{aligned}$$

The third expectation in the last equality is bounded by γ_{in}, using the second condition of the LDLE and the fact that $\mathrm{agr}((w'_{b,r})_a, (c^{b,r})_a) I_{(c^{b,r})_a \notin L_{in}(w'_a)_{b,r}} \leq \mathrm{agr}((w'_{b,r})_a, (C_a)_{b,r} \setminus L_{in}(w'_a)_{b,r})$. The second expectation is also bounded by γ_{in}, using the facts that $\mathrm{agr}(\cdot) \leq 1$, and that for every $(b,r) \in B \times [r^{in}]$ if $D_b(c^{b,r}) \notin W_b$ then $\Pr_{a\in\Gamma_G(b)}((c^{b,r})_a \in L_{in}(w'_a)_{b,r}) < \gamma_{in}$. To bound the first expectation, notice that

$$\begin{aligned}
&\mathrm{agr}((w'_{b,r})_a \setminus (c_{b,r})_a, (c^{b,r})_a) I_{(c^{b,r})_a \in L_{in}(w'_a)_{b,r} \wedge D_b(c^{b,r}) \in W_b} \\
&\qquad \leq I_{(c^{b,r})_a \in L_{in}(w'_a)_{b,r} \setminus (L(w')_a)_{b,r} \wedge D_b(c^{b,r}) \in W_b} \\
&\qquad \leq I_{W_b \cap c^{ext}_b \neq \phi},
\end{aligned}$$

where $c^{ext} \in (C_a \setminus L(W)_a)^{\odot s_2}$ is the s_2-extended word created from the set $Enc^{-1}(L_{in}(w'_a) \setminus L(w')_a)$. Hence, using the assumption that the tester (C, G) is (γ, s_1, s_2, L)-ELLE, the first expectation is bounded by $\mathbb{E}_a[\mathrm{agr}(W_a, Enc^{-1}(L_{in}(w'_a) \setminus L(w')_a))] \leq \gamma$.

□

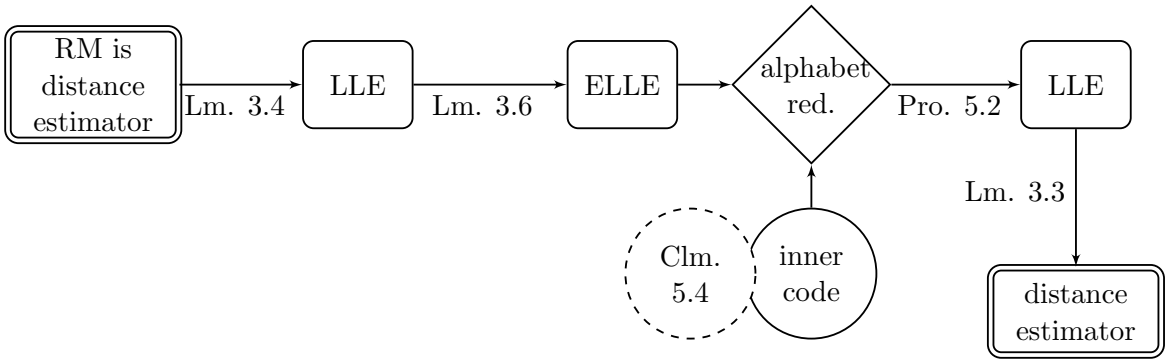

Figure 2: Structure of the proof of Theorem 1.2.

6 Putting the Pieces Together

In this final section we use what we constructed in previous sections to prove Theorems 1.2 and 1.3. See Figures 2 and 3 for the structure of the constructions.

In both theorems, given a message length k, we choose parameters m, d such that $k = \binom{m+d}{m}$. We start with the tester $(RM^{\mathbb{F},m,d}, G^{LDT,\mathbb{K}})$. Let $h = d/m + 1$ and notice that $k = \binom{mh}{m}$ and therefore $h^m \leq k \leq (eh)^m$. The distance of $RM^{\mathbb{F},m,d}$ is, by Lemma 2.2, $1 - \delta_{agr}$, where $\delta_{agr} = d/|\mathbb{F}|$. Number of queries is at most $|\mathbb{F}|^3 = (d/\delta_{agr})^3 \leq (mh/\delta_{agr})^3$. The length of the code is $|\mathbb{F}|^m = (d/\delta_{agr})^m \leq (m/\delta_{agr})^m \cdot k$. We will consider two choices of m: (i) $m = \log k / \log\log k$; this choice yields polynomial length. (ii) $m = (\log k)^\alpha$ for some constant $\alpha < 1$; this choice yields almost linear length.

We then transform the initial tester into an ELLE. Using Corollary 4.8 and Lemma 3.6 we can establish the next corollary.

Corollary 6.1 (Low Degree Testing Theorem-III)**.** *For every field $\mathbb{F}$, a subfield $\mathbb{K} \subseteq \mathbb{F}$, for all integers $s_1, s_2 > 0$, and $\delta \geq \delta_{m,d,|\mathbb{K}|,|\mathbb{F}|}$, the following holds. The tester $(RM^{\mathbb{F},m,d}, G^{LDT,\mathbb{K}})$ is a $(O(\delta s_1 s_2), s_1, s_2, s_1 \frac{2}{\delta})$-ELLE.*

We continue with proving each theorem separately. For ease of readability, we first repeat the statements of the theorems.

Theorem 1.2. *For any $k \geq 1$ and for any $1/\log^{1/17} k < \varepsilon < 1$ there is an ε-distance estimator with message length k, alphabet size $\mathrm{poly}(\frac{1}{\varepsilon})$, and distance $1 - O(\varepsilon^2)$. The other parameters are either (i) polynomial length and $\mathrm{poly}(\log k)$ number of queries, or (ii) almost linear length (i.e., $k^{1+o(1)}$) and $k^{o(1)}$ number of queries.*

Proof. Fix an arbitrary $1/\log^{1/17} k < \varepsilon < 1$. Choose the minimal $|\mathbb{F}|, |\mathbb{K}|$ such that $\delta_{m,d,|\mathbb{K}|,|\mathbb{F}|} \leq \varepsilon^2$. Note that, $|\mathbb{F}|, |\mathbb{K}| \leq m^{O(1)} h \cdot \varepsilon^{-16}$. Notice that $\delta_{agr} \leq \varepsilon^2$ and the length is at most $m^{O(m)} k \varepsilon^{-O(m)} \leq m^{O(m)} k^{1+o(1)}$. In this theorem we use the two choices of m mentioned above.

Apply Corollary 6.1 to $(RM^{\mathbb{F},m,d}, G^{LDT,\mathbb{K}})$ with $\delta = \varepsilon^2$ and $s_1 = 2/\varepsilon$ to obtain an $(O(\varepsilon), 2/\varepsilon, 1, 4/\varepsilon^3)$-ELLE. We perform alphabet reduction operation with $\delta_{in} = (\varepsilon/2)^2$ and get a $(O(\varepsilon), 4/\varepsilon^3)$-LLE (see

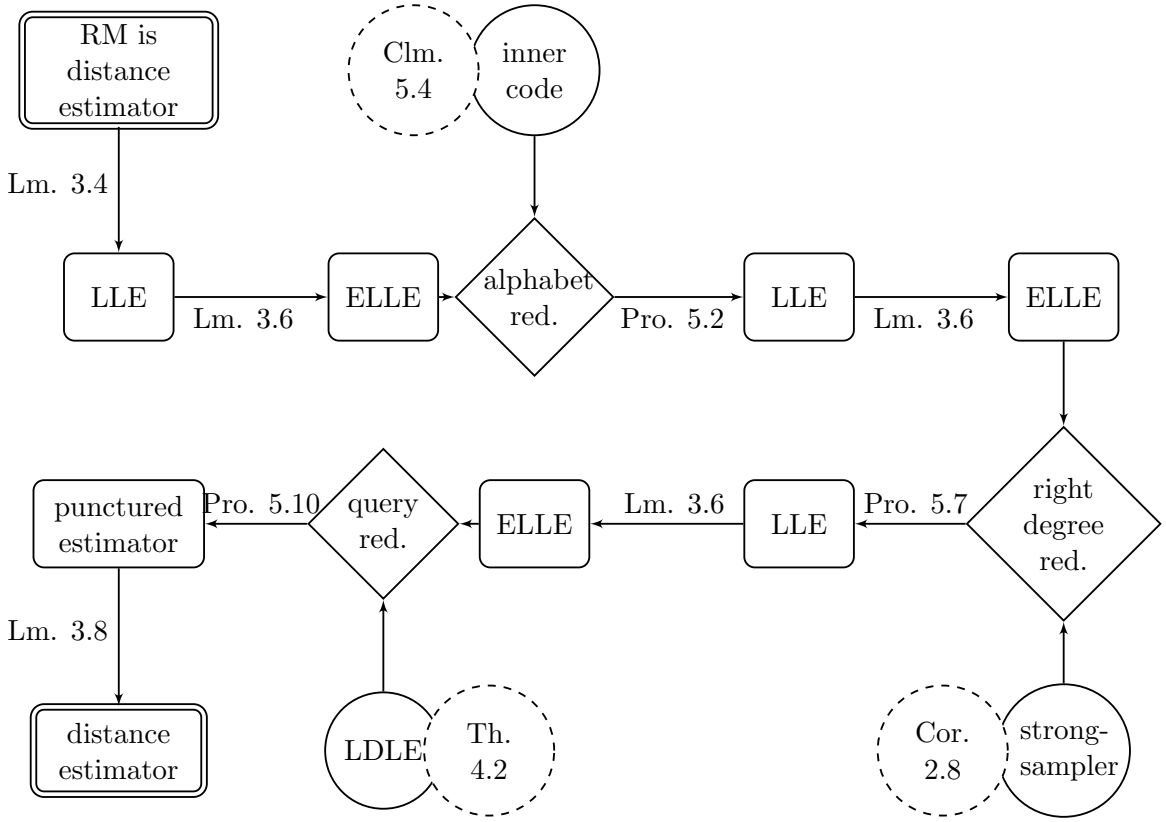

Figure 3: Structure of the proof of Theorem 1.3.

Corollary 5.5). The alphabet size is $O(1/\varepsilon^4)$ and distance is $1 - O(\varepsilon^2)$. The length and the number of queries increased by a factor of $O(\log|\mathbb{F}|/\varepsilon^4)$.

In [19, Corollary 5.9] we can find the proof that $G^{LDT,\mathbb{K}}$ is a sampler. More formally,

Claim 6.2. *For any $\beta > 0$, $G^{LDT,\mathbb{K}}$ is a $(O(\beta), \frac{1}{\beta^2} \cdot (\frac{1}{|\mathbb{K}|} + \frac{1}{|\mathbb{F}|}))$-sampler.*

Using the claim, with $\beta = \varepsilon$, and Claim 5.3 we know that the graph of the tester, after performing alphabet reduction, is a $(O(\varepsilon), \varepsilon^4)$-sampler. After using Lemma 3.3 we know that the tester we constructed is an $O(\varepsilon)$-distance estimator. We obtain the result by properly choosing ε. □

Theorem 1.3. *For any $k \geq 1$, for any constant $0 < c < 1$, exists $c' = O(c)$ such that for any $1/\log^{c'} k < \varepsilon < 1$ there is an $(\varepsilon, \mathrm{poly}(\frac{1}{\varepsilon}))$-distance estimator with message length k, almost linear length, $\log^c k$ number of queries, $\mathrm{poly}(\log k)$ alphabet size, and distance $1 - 1/\mathrm{poly}(\log k)$.*

Proof. Let $0 < c < 1$ be a constant and $1/\log^{O(c)} k < \varepsilon < 1$. We choose $m = (\log k)^\alpha$ for some constant α to be chosen latter, and we choose $|\mathbb{F}|$ and $|\mathbb{K}|$ such that $\delta_{m,d,|\mathbb{K}|,|\mathbb{F}|} \leq \varepsilon^{16}$. Note that $|\mathbb{F}| = m^{O(1)} h \cdot \varepsilon^{-4\cdot 16}$, and $|\mathbb{K}| = m^{O(1)} \varepsilon^{-8\cdot 16}$. Notice that $\delta_{agr} \leq 1/\mathrm{poly}\log k$, the length is $m^{O(m)} k \varepsilon^{-O(m)} \leq k^{1+o(1)}$ and the left side size is $|\mathbb{F}|^m |\mathbb{K}|^{2m} = k^{1+o(1)}$. By Corollary 4.8, we know that $(RM^{\mathbb{F},m,d}, G^{LDT,\mathbb{K}})$ is an $(\varepsilon_{start}, O(\varepsilon_{start}^{-1}))$-LLE, for large enough ε_{start}.

Then, perform the combinatorial operations: alphabet reduction with added error ε_{alpha} and alphabet size $\mathrm{poly}\log k$ (see Corollary 5.5), right degree reduction with added error ε_{deg} and query reduction with added error ε_{query} (see Table 5.1). Before using any of the operations, we use Lemma 3.6 to make the tester ELLE and not just LLE. The error, after performing all three operations, is (the $O(\cdot)$ notation is omitted):

$$\left(\left(\frac{\varepsilon_{start}}{\varepsilon_{alpha}} + \varepsilon_{alpha}\right)\frac{1}{\varepsilon_{deg}} + \varepsilon_{deg}\right)\frac{1}{\varepsilon_{query}^3} + \varepsilon_{query}.$$

We want the error to be at most $O(\varepsilon)$. For this, we choose $\varepsilon_{query} = \varepsilon$, $\varepsilon_{deg} = \varepsilon^4$, $\varepsilon_{alpha} = \varepsilon^8$, and $\varepsilon_{start} = \varepsilon^{16}$.

We investigate the values of the other parameters after performing the combinatorial operations. The list size has multiplied by $O(1/\varepsilon^4)$. The alphabet size is at most $\mathrm{poly}\log k$. The length is almost linear. The distance is $1 - 1/\mathrm{poly}\log k$. The query size is at most $\mathrm{poly}\frac{\log|\mathbb{F}|}{\varepsilon} \leq \log^c k$, for appropriate α. We get the result by properly choosing ε. We use Lemma 3.8 to get a distance estimator. □

Acknowledgements. I would like to thank my advisors. Oded Regev, for long talks on this subject that helped me gain another perspective and for his many comments making this paper more readable. Amir Shpilka, for his encouragement and helpful comments.

I would like to thank Irit Dinur and Prahladh Harsha for helpful discussions.

I am deeply indebted to my sister, Dana Moshkovitz, for introducing me to the PCP world, suggesting this problem to me, and for helping me overcome problems that arose through this paper. I would like to thank my brother, Guy Moshkovitz, for many illuminating conversations.

References

[1] N. Alon, J. Bruck, J. Naor, M. Naor, and R. M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38(2):509–516, 1992.

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

[3] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

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

[5] E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, and S. P. Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006.

[6] E. Ben-Sasson and M. Sudan. Robust locally testable codes and products of codes. *Random Struct. Algorithms*, 28(4):387–402, 2006.

[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 *STOC*, pages 612–621, 2003.

[8] I. Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007.

[9] I. Dinur and E. Goldenberg. Locally testing direct product in the low error range. In *FOCS*, pages 613–622, 2008.

[10] I. Dinur and P. Harsha. Composition of low-error 2-query PCPs using decodable PCPs. In *FOCS*, pages 472–481, 2009.

[11] G. D. Forney. *Concatenated Codes*. MIT Press, 1966.

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

[13] V. Guruswami. *List Decoding of Error-Correcting Codes*, volume 3282 of *Lecture Notes in Computer Science*. Springer, 2005.

[14] V. Guruswami and A. Rudra. Tolerant locally testable codes. In *APPROX-RANDOM*, pages 306–317, 2005.

[15] R. Impagliazzo, V. Kabanets, and A. Wigderson. New direct-product testers and 2-query PCPs. In *STOC*, pages 131–140, 2009.

[16] J. Katz and L. Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *STOC*, pages 80–86, 2000.

[17] C. Lund, L. Fortnow, H. J. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[18] D. Moshkovitz. Mini-course on projection PCPs, Lecture 4, 2009. [online] `http://www.math.ias.edu/~hdanam/courses/projection/index.html`.

[19] D. Moshkovitz and R. Raz. Sub-constant error low degree test of almost-linear size. *SIAM Journal on Computing*, 38(1):140–180, 2008.

[20] D. Moshkovitz and R. Raz. Sub-constant error probabilistically checkable proof of almost linear size. *Computational Complexity*, 2009.

[21] D. Moshkovitz and R. Raz. Two query PCP with sub-constant error. *J. ACM*, 57(5), 2010.

[22] 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.

A An alternative Proof of the Schwartz-Zippel Lemma

In this section we show a proof of a slightly weaker form of Lemma 2.2 ($d/(|\mathbb{F}|-1)$ instead of $d/|\mathbb{F}|$).

From the assumption that p is not the zero polynomial we know that there is a point $v \in \mathbb{F}^m$ such that $p(v) \neq 0$. Notice that the restriction of p to any line is a univariate polynomial of degree at most d. The $|\mathbb{F}|^m-1/|\mathbb{F}|-1$ lines that pass through v form a partition of $\mathbb{F}^m \setminus \{v\}$. Hence,

$$|\{x \in \mathbb{F}^m \mid p(x) = 0\}| \leq \frac{|\mathbb{F}|^m - 1}{|\mathbb{F}| - 1} d < |\mathbb{F}|^m \frac{d}{|\mathbb{F}| - 1}.$$

ECCC ISSN 1433-8092

http://eccc.hpi-web.de