Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 134 (2026)

A Simple Algebraic Proof of the PCP Theorem

Prashanth Amireddy[*] Amik Raj Behera[†] Srikanth Srinivasan [‡] Madhu Sudan[§]
Sophus Valentin Willumsgaard [¶]

October 22, 2026

Abstract

We give the simplest known algebraic proof of the PCP theorem, involving only ingredients like code concatenation, polynomial interpolation, and polynomial multiplication. Specifically, we prove that graph 3-coloring has a polynomial-sized proof that can be verified by a verifier tossing logarithmically many coins and querying a constant number of bits in the proof. In particular, our proof does not involve any PCP compositions; notably, it does not invoke the NP-completeness of any fixed problem, such as SAT or 3-coloring, in the construction of the verifier. The main innovation in our work is a clean, coding theoretic, way to encode univariate polynomials that allows us to implement "low-degree testing" using just a constant number of bits of queries. Insights from recent attempts to simplify the PCP proof by the authors (STOC 2026) and Goldreich (ECCC 2025) allow us to observe that low-degree testing was the key bottleneck in converting previous algebraic constructions of the PCP verifier into a constant query PCP. Thus, by overcoming this bottleneck, we get the full PCP verifier using elementary and self-contained steps. As concrete support for the claimed simplicity, we include the full pseudocode of the PCP verifier, assuming finite field arithmetic, and a full description of the completeness (aka "honest") prover, assuming multivariate polynomial arithmetic including interpolation and evaluation, that fit in about a page each.

[*]School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts, USA. Supported in part by a Simons Investigator Award and NSF Award CCF 2152413 to Madhu Sudan and a Simons Investigator Award to Salil Vadhan. Part of this work was done during a visit to the University of Copenhagen supported by the European Research Council (ERC) under grant agreement no. 101125652 (ALBA). Email: pamireddy@g.harvard.edu

[†]Department of Computer Science, University of Copenhagen, Denmark. Supported by Srikanth Srinivasan's start-up grant from the University of Copenhagen. Email: ambe@di.ku.dk

[‡]Department of Computer Science, University of Copenhagen, Denmark. Supported by the European Research Council (ERC) under grant agreement no. 101125652 (ALBA). Email: srsr@di.ku.dk

[§]School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts, USA. Supported in part by a Simons Investigator Award, NSF Award CCF 2152413 and AFOSR award FA9550-25-1-0112. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights. Email: madhu@cs.harvard.edu

[¶]Department of Computer Science, University of Copenhagen, Denmark. Supported by the European Research Council (ERC) under grant agreement no. 101125652 (ALBA). Email: sophus.willumsgaard@di.ku.dk

ISSN 1433-8092

Contents

1 Introduction **1**

2 Preliminaries **5**
- 2.1 PCPs . . . 5
- 2.2 Low-Degree Test and Local Correction over Large Alphabet . . . 9

3 Two Encodings **11**
- 3.1 Encoding Univariate Polynomials using Set-Multilinear Encoding . . . 11
- 3.2 Syntactic Test and Self-Correction of Set-Multilinear Encoding . . . 12
- 3.3 Encoding Field Elements using Low-Degree Hadamard Codes . . . 14
- 3.4 Syntactic Test and Self-Correction of Low-Degree Hadamard Code . . . 16

4 Low-Degree Test over Binary Alphabet **19**
- 4.1 Warmup: Low-Degree Test over alphabet $\mathbb{F}_q$ . . . 19
- 4.2 Low-Degree Test over $\mathbb{F}_2$ . . . 21
- 4.3 Proof of Lemma 4.3 . . . 28

5 Local Correction over Binary Alphabet **30**

6 Zero-on-Grid Test over Binary Alphabet **33**

7 The PCP Verifier **34**

References **42**

A Proofs from Section 2 and Section 3 **44**

B Analysis of HadTest **47**

C Proof of Local Correction **49**

D Proof of Zero-on-Grid Test **51**

E Honest Prover and Full PCP Verifier **53**

1 Introduction

Ever since the discovery of the PCP theorem in the early 90s [AS98; ALMSS98], a "simple proof" of the PCP theorem has been an object of desire in the CS community. Indeed, this quest has led to many improvements to the original algebraic proof, while also launching radically novel directions as in the proof of the PCP theorem by gap amplification by Dinur [Din07] (see also Dinur and Reingold [DR06]). The algebraic proofs and gap amplification proofs are quite far from each other, and the relative simplicity is hard to compare. In this work, we focus on the algebraic direction and further simplify it to be able to achieve two concrete milestones: Specifically, we give pseudocodes (assuming finite field and polynomial arithmetic) for the PCP verifier, as well as the completeness (aka "honest") prover, each of which fits in about a page (see Appendix E). Our constructions have also been implemented in Python [ABSSW26b], and our proof has been formalized in Lean by Lee [Lee26]. One notable aspect of our proof is that there is no composition of proofs in our verifier, and in particular we do not rely on NP-completeness proofs of some canonical problem to recursively verify proofs. To elaborate further, we give some background on past algebraic proofs of the PCP theorem and the ingredients therein.

Before delving into the details, let us recall that PCPs are typically parameterized by two main parameters — the *randomness* complexity of the verifier, ideally logarithmic in the length of the classical proof; and the *query* complexity, ideally constant. Achieving both ideal limits yields the PCP theorem, which is the goal of this paper; however, the constructions involve ingredients that are weak in one or both parameters. Of course our objective is to achieve these ideal parameters with a "simple" proof. As indicated above, we use this term informally to capture two ingredients, namely (1) the PCP verifier who, given an instance of say graph 3-coloring, determines the length of the proof and the queries to be performed and the acceptance condition and (2) the "honest" prover (formally the completeness prover), who given a 3-coloring of a 3-colorable graph, computes the bits of the PCP proof to supply to the verifier (which the verifier should accept with probability one). Indeed, one could argue that the complexity of this honest prover is perhaps the most natural measure of complexity of a PCP proof in that it explains what the PCP proof looks like; and this has been a complex transformation in all previous proofs and can be viewed as the primary motivation of this work.

The Ingredients in an Algebraic PCP: Algebraic PCPs, starting with the works of Arora and Safra [AS98] have relied on three kinds of ingredients: (1) "Atomic PCPs", which are self-contained, simple constructions of PCPs for some natural problem (like SAT or Graph-Coloring), which are deficient in at least one of the two parameters. (2) "Composition" — a technique that combines two PCP-like objects that we will call outer and inner verifiers to get a new object that has randomness complexity close to that of the outer verifier and query complexity closer to that of the inner verifier. Applying this operator judiciously can get both parameters closer to the ideal limit. (3) "Composition-preparation" — we coin this term and use it to describe a variety of steps involved to convert the atomic PCPs into outer and inner verifiers suitable for composition. In some cases, like in the original proof of the PCP theorem in the work of Arora, Lund, Motwani, Sudan and Szegedy [ALMSS98], this can be a technically complex operation. But in all cases, the preparations are needed to make the atomic verifier into a "robust" one suitable for use as an outer verifier (involving operations termed "parallelization" (see [ALMSS98]) or "bundling" (see [BGHSV06]) or "robustification") as well as into a "PCP of proximity" or "assignment tester" (see [BGHSV06; DR06]) which makes them suitable for use as an inner verifier. (Notably, the use of the NP completeness of the "natural problem" for which the PCP was originally designed to get a verifier for less natural and harder-to-specify problems.) Together, "composition" with "composition-preparation" contribute to the bulk of the conceptual

complexity of algebraic PCPs, making them hard to describe and even harder to teach. In this work, our goal is to build a PCP whose complexity is not much more than that of the atomic PCP, foregoing composition-preparation altogether and simplifying "composition" to the level of "code-concatenation" which we expand on next.

Composition of PCPs vs. Concatenation of Codes (see also Goldreich [Gol25, Section 4]): Concatenation of codes is an elementary operation that converts codes over large alphabets into codes over smaller ones. Concatenation uses two ingredients - an outer code over a large alphabet, where the large alphabet makes it easy to achieve a large distance between codewords; and an inner code that shows how to encode elements of the large alphabet as, say, binary strings. Concatenation simply converts the codewords of the outer code into binary strings by writing each element of the alphabet in binary using the inner code. Notationally, concatenation is as simple as composition of functions (viewed appropriately).

Composition of PCPs is an extension of this concept to PCPs but becomes significantly more complex. Viewed in terms of what the honest prover would do, composition takes an outer prover who builds a PCP proof over a large alphabet (in which the verifier queries a constant number of symbols) and then converts this proof into a PCP proof on bits. But this translation involves a number of complexities beyond concatenation. (1) In the simplest form, the bits of the composed proof are encodings of all c-tuples of symbols of the outer code for some constant c. (2) In a slightly more complex form, a special family of c-tuples of the symbols of the outer codes is chosen to be re-encoded. (3) In the most complex form, the encoding of each c-tuple depends on the acceptance condition that the outer verifier intends to apply, and thus varies from c-tuple to c-tuple. (Typically this step involves use of the Cook-Levin theorem to express arbitrary outer verifiers in some canonical form using an NP-complete problem.)[1]

Our result and ideas: The PCP we present in this paper eliminates the need for compositions in the sense above.[2] As a consequence the honest prover as well as the verifier become significantly simpler and can be described in a relatively straightforward manner using standard polynomial manipulations. Broadly, our PCP is obtained by taking the simplest known atomic outer PCP, namely the one from [ABSSW26a]; identifying the key bottleneck to a simple composition there, namely the "low-degree test", and then giving a new encoding and implementation of the low-degree test that allows for a simple composition. We expand on these steps by first giving a quick overview of the atomic test.

The atomic test we use goes back to the work of Babai, Fortnow, and Lund [BFL91], which was first converted to a "large alphabet" constant query test in [ALMSS98]. These works effectively show that verifying graph 3-coloring on n vertex graphs "reduces" to verifying that a constant number of m-variate polynomials over a field $\mathbb{F}_q$ are of degree at most d for $m, d, q = \mathrm{poly}(\log n)$ and verifying that these polynomials are zero on some subset of the domain of the form H^m for $H \subseteq \mathbb{F}_q$.[3] The latter task used to involve the famed "sum-check protocol" but this aspect (while elegant) is cleaned up significantly in the work of Ben-Sasson and Sudan [BS08] who show that the final task reduces to m low-degree tests on m additional polynomials, and further cleaned up by previous work of the authors [ABSSW26a] who reduce this to one low-degree test on just one additional $2m$-variate polynomial. The complexity hidden by the

[1] In our estimate, all previous algebraic proofs including [ALMSS98; BGHSV06; BS08; ABSSW26a] use (3) though all but [ALMSS98] don't rely on (2). The PCP in [Din07] relies crucially on (2) but does not need (3).

[2] One could argue that the use of the low-degree test, described below, which necessitates a suitable 'proof oracle', is a non-trivial mechanism that might count for '0.5' compositions in the sense of Goldreich [Gol25].

[3] A similar reduction also works for SAT but the reduction is cleaner and more straightforward for 3-coloring.

"reductions" alluded to above involves querying a constant number of polynomials at a constant number of locations and verifying that the query responses are zeroes of a constant number of constant-degree polynomials.

Thus, apart from the "low-degree test" (still to be discussed), all aspects of the proof thus far require querying the proof in a constant number of places, where every element of the proof is from $\mathbb{F}_q$ and thus requires $\Theta(\log \log n)$ bits to describe. The query complexity here can be reduced to $\mathcal{O}(1)$ bits by simple concatenation with the "low-degree long code" (see [BGHMRS15])[4] which involves writing down the evaluation of every $\mathcal{O}(1)$-degree polynomial of these $\mathcal{O}(\log \log n)$ bits. This encoding is known to be locally testable and sufficient to perform the tests used in the reductions mentioned above; and their encoding takes exponential length, but on strings of length $\log \log n$ we can afford this blow-up! This leads us to the key bottleneck, namely the low-degree test.

The low-degree test: The low-degree testing problem is a simple one to specify: We are given oracle access to a function $f : \mathbb{F}_q^m \to \mathbb{F}_q$, a degree parameter d and we would like a test that queries f in $\mathcal{O}(1)$ locations and accepts degree d polynomials while rejecting functions that are, say 0.01-far (in relative Hamming distance) from degree d polynomials with probability say 0.0001. Unfortunately, as stated, this problem is not solvable, and so one allows an auxiliary "proof" oracle f_1. Now we would like that the test is allowed $\mathcal{O}(1)$-queries into f and f_1, such that for every degree d polynomial f there exists f_1 such that the verifier always accepts (f, f_1) while if f is 0.01-far, then for every f_1 the pair (f, f_1) is rejected with probability at least 0.0001. Such tests are well-known — we use the line-point test of Rubinfeld and Sudan [RS96] that suffices for our purpose. Here the function $f_1 : \mathbb{F}_q^{2m} \to \mathbb{F}_q^{d+1}$ where the domain represents a pair of points in $\mathbb{F}_q^m$ and the $f_1(a, b)$ purportedly represents the degree d univariate polynomial that describes f restricted to a line containing a with slope b. The test picks $a, b \in \mathbb{F}_q^m$ and a non-zero $\lambda \in \mathbb{F}_q$ randomly and verifies that $f_1[a, b](\lambda) \stackrel{?}{=} f(a + \lambda b)$ which is an $\mathbb{F}_q$-linear test involving just one query each to f and f_1.

Unfortunately, we cannot reduce the alphabet size of the prover in this test by using a "low-degree long code". In particular evaluations of f_1 are $d \log q = \text{poly}(\log n)$ bit strings (specifically the strings are $\mathcal{O}(\log n)^{c_1}$ bits long for some $c_1 > 1$.) Low-degree long-code encodings map k-bit strings to $2^{k^{c_2}}$-bit strings for some constant $c_2 > 1$. Composing the two gives a quasi-polynomial-sized PCP (equivalently, the resulting randomness complexity would be $\mathcal{O}(\log n)^{c_1 c_2}$) which does not meet our ideal goal. This leads us to our key problem and solution described next.

A new multivariate encoding: Our goal is to come up with some encoding $E : \mathbb{F}_q^d \to \mathbb{F}_2^N$ for some N, such that this encoding is easy to test with $\mathcal{O}(1)$ queries and so that given $\lambda \in \mathbb{F}_q$, the encoding $E[f_1(a, b)]$ allows us to "locally retrieve" $f_1(a, b)(\lambda)$ (specifically we need a "self-corrector", one that, even if the encoding is slightly corrupted, for every λ correctly reports $f_1(a, b)(\lambda)$ for the nearest codeword). For our application achieving $N = 2^{d^\epsilon}$ for arbitrarily small $\varepsilon > 0$ works and we can allow a locality of $\mathcal{O}_\varepsilon(1)$. Such a task has been considered in the literature in the past, specifically in the work of Dinur, Fischer, Kindler, Raz, and Safra [DFKRS11], though possibly with additional requirements on the nature of the information that is locally retrievable from the information and with different requirements on the locality and N.

[4] A low-degree long code encodes a field element $\alpha \in \mathbb{F}_q$, where $\mathbb{F}_q \cong \mathbb{F}_2^t$, by evaluations $R(\alpha)$ of all $\mathcal{O}(1)$-degree t-variate polynomials R.

We start by getting a non-binary encoding, specifically an $E_0 : \mathbb{F}_q^d \to \mathbb{F}_q^{N'}$ with $N' = 2^{O(d^\varepsilon)}$. Concatenation with a Hadamard code converts this to a binary code in a standard way, and we omit the details here. To get this encoding E_0, we represent a univariate degree $< d$ polynomial $P(X)$ by a multivariate polynomial of lower degree. We call this the *Set-Multilinear encoding* of P as it is essentially a "set-multilinearization" of the well-known Inverse Kronecker map that encodes a high-degree univariate polynomial as a low-degree multivariate polynomial.

The class of multivariate polynomials we use in the encoding is set-multilinear[5] polynomials in c blocks of variables (we will choose c to be an absolute constant), with each block containing $d^{1/c}$ variables. Denoting these blocks of variables by $\{Y_{i,j}\}_{i\in[c],j\in[\ell]}$ for $\ell = \lceil d^{1/c} \rceil$, we think of the variable Y_{ij} as representing the monomial $X^{(j-1)\ell^{i-1}}$. Writing every exponent in base ℓ we get a way to represent every monomial X^e as a set multilinear form in the Y_{ij}'s. Thus every polynomial $P(X)$ can be represented as a set-multilinear polynomial, and hence degree $\leqslant c$, polynomial $\widetilde{P}$ in the variables Y_{ij}. Our encoding of P is simply the evaluation of $\widetilde{P}$ over $\mathbb{F}_q^{c\ell}$. We test this encoding loosely by simply testing that it is a degree $\leqslant c$ polynomial. Since c is a constant, this only requires constantly many queries into the encoding. Every honest encoding of a univariate polynomial passes the test. But if a function passes the test, we only get that it is close to the evaluation of some $\widetilde{P}$ that has total degree $\leqslant c$. But this is sufficient to argue that $\widetilde{P}((X^{(j-1)\ell^{i-1}})_{i,j})$ is a (univariate) polynomial of degree $\mathcal{O}(cd)$ and furthermore this encoding can be locally self-corrected. The degree guarantee is weaker than strictly needed in a low-degree test[6] but this is completely fine for the soundness proof. We note that the encoding proposed in [DFKRS11] also seems to start with the inverse Kronecker map, but does not do the "set multilinearization". As a result their encoding is more efficient, with $N = \mathrm{poly}_\varepsilon(d)$, but achieves locality of $\mathcal{O}(d^\epsilon)$ for local information retrieval.

Combining this encoding of univariate low-degree polynomials with the other ingredients listed above (the atomic PCP from [ABSSW26a] and the low-degree long code) now suffices to give us our PCP.

Our new multivariate encoding (along with the replacement for the sum-check protocol based on [BS08; ABSSW26a]) allows us to extend this technique to get the randomness complexity down to $\mathcal{O}(\log n)$, at the expense of one more code-concatenation.

Comparison with the work of Goldreich [Gol25]: This result is motivated by a result of Goldreich [Gol25], who shows how to obtain a PCP theorem with randomness complexity n^ε (for arbitrarily small ε) and $\mathcal{O}(1)$ queries using only code-concatenation. ([Gol25] describes the use of code concatenation as a "half-composition". Using this PCP (or a variant thereof) as an inner PCP that can be composed with the outer PCP from [ALMSS98] yields the full PCP theorem with one (and a half) composition steps.)

Goldreich's main result, namely the $\mathcal{O}(1)$-query PCP with randomness complexity n^ε, is limited by the use of a version of the sum-check protocol, which has constant query complexity only for a restricted range of parameters. One can view our construction as modifying the construction from [Gol25] with a replacement for the sum-check protocol based on [BS08; ABSSW26a]. This allows us to reduce the query complexity to a constant while keeping randomness $\mathcal{O}(\log n)$, while paying a price in the alphabet size of the PCP which is now $\mathcal{O}(\mathrm{poly}\log n)$ bits. This answer size could be reduced to $\mathcal{O}(1)$ by encoding answers using a Hadamard PCP and applying a recursive verification strategy as in [Gol25]. This would bring

[5] A multivariate polynomial in c disjoint sets of variables $Y_1, \ldots, Y_c$ is *set-multilinear* if every monomial contains exactly one variable from each Y_i.

[6] This could be rectified with a few more tests, but we do not do so in the interest of keeping the verifier simple

down the query complexity to constant, but would increase the randomness to $\mathcal{O}(\operatorname{poly}\log n)$ and thus fall short of our goal of reducing the randomness down to $\mathcal{O}(\log n)$. To bring down the randomness to $\mathcal{O}(\log n)$, we need our set-multilinear encoding of univariate polynomials, which allows us to reduce the answer size more randomness-efficiently and thus get the full PCP theorem at the expense of two levels of code-concatenation.

Organization of the paper

In Section 2, we give the necessary definitions, background results, and state the PCP Theorem (Theorem 2.2). In Section 3, we describe the two encodings that we will use: set-multilinear encoding (Section 3.1) and low-degree Hadamard encoding (Section 3.3). We also discuss their syntactic test (i.e., a test to decide whether the oracle is close to the expected form) and self-correctors in Section 3.2 and Section 3.4, respectively. We provide the proofs from Section 3 in Appendix A and Appendix B. Using these two encodings, we give our low-degree test in Section 4 and analyze it. In Section 5, we give our local corrector over alphabet $\mathbb{F}_2$ and analyze it in Appendix C. In Section 6, we give our zero-on-grid test over alphabet $\mathbb{F}_2$ and analyze it in Appendix D. Using these three algorithms (low-degree test, local corrector, and zero-on-grid test), we describe our PCP verifier in Section 7. In Appendix E, we give a complete standalone description of the PCP verifier and an honest prover.

2 Preliminaries

Probabilistic Algorithms and Oracles For an algorithm $\mathcal{V}$ and a string Π over alphabet Σ, we will use $\mathcal{V}^{\Pi}$ to denote the algorithm $\mathcal{V}$ with oracle access to string Π. By oracle access, we mean $\mathcal{V}$ can query $\Pi[i]$ for any $1 \leqslant i \leqslant |\Pi|$. For a probabilistic algorithm $\mathcal{V}$, we will use the notation $\mathcal{V}^{\Pi}(x; R)$ to say that the algorithm $\mathcal{V}$ has oracle access to Π, has input x, and access to a random string R. In this notation, $\mathcal{V}^{\Pi}(x; R)$ is a deterministic algorithm, and the randomness is in the choice of R. For a string Π and a subset $S \subseteq [|\Pi|]$, $\Pi|_S$ refers to the substring formed by entries of Π corresponding to S.

2.1 PCPs

Throughout this article, whenever we mention a verifier, we mean a *standard verifier*, which we define next. We will always use a standard verifier, both in PCP and in the subroutines such as low-degree test.

Definition 2.1 (Standard Verifier)**.** *For functions $r, \ell : \mathbb{Z}^{\geqslant 0} \to \mathbb{Z}^{\geqslant 0}$, define an (r, ℓ)-standard verifier $\mathcal{V}$ as follows: On input $\mathbf{x} \in \{0,1\}^n$ of length n, a string[7] $R \in \{0,1\}^{r(n)}$, and oracle access to a string[8] $\Pi \in \{0,1\}^{\operatorname{size}(n)}$,*

- *$\mathcal{V}$ computes a subset $Q = Q(\mathbf{x}, R) \subseteq [\operatorname{size}(n)]$ of cardinality $\ell(n)$. These are the* queries *of $\mathcal{V}$ on input $\mathbf{x}$ and randomness R.*
- *$\mathcal{V}$ then either returns* Accept *or* Reject *depending on $\Pi|_Q$.*

Furthermore, the running time of $\mathcal{V}$ has to be $\operatorname{poly}(n \cdot 2^{r(n)})$.

[7] This string R is the random string fed into $\mathcal{V}$.
[8] called a *proof*

Observe that in the above definition, $\mathcal{V}$ makes $\ell(n)$ queries to Π using the $r(n)$ coin tosses. In particular, $\mathcal{V}$ can only query a coordinate within range of $[0, \ell(n) \cdot 2^{r(n)} - 1]$. So without loss of generality, we can always assume that the proof size $|\Pi|$ is $\mathrm{size}(n) = \mathcal{O}(\ell(n) \cdot 2^{r(n)})$.

In this paper, we will focus on the language 3-COLOR of 3-colorable graphs, which is known to be NP-complete. This is only because 3-COLOR fits quite nicely with our PCP proof. We could always construct a PCP verifier similar to ours for other NP problems. In particular, we show that certifying membership in 3-COLOR admits $(\mathcal{O}(\log n), \mathcal{O}(1))$-standard verifiers:

Theorem 2.2 (The PCP Theorem [ALMSS98])**.** *There exists a randomized oracle algorithm $\mathcal{V}$, a constant $\gamma > 0$, and a polynomially growing function $\mathrm{size}(n)$ such that for every n and for every input graph $G = (V, E)$ over n vertices, we have the following.*

- *If $G \in$ 3-COLOR, then there exists $\Pi \in \{0,1\}^{\mathrm{size}(n)}$ such that*

$$\Pr[\mathcal{V}^{\Pi}(G) \text{ returns } \textsc{Accept}] = 1.$$

- *If $G \notin$ 3-COLOR, then for every $\widetilde{\Pi} \in \{0,1\}^{\mathrm{size}(n)}$, we have*

$$\Pr[\mathcal{V}^{\widetilde{\Pi}}(G) \text{ returns } \textsc{Reject}] \geqslant \gamma.$$

Furthermore, for every $\Pi \in \{0,1\}^{\mathrm{size}(n)}$, $\mathcal{V}^{\Pi}(G)$ runs in time $\mathrm{poly}(n)$, uses $\mathcal{O}(\log n)$ bits of randomness, and makes $\mathcal{O}(1)$ oracle queries.

For a prime power q, we denote the finite field of size q by $\mathbb{F}_q$. Throughout this article, we will have $q = 2^t$ for some $t \in \mathbb{N}$. We will use $\mathcal{P}_d(m, \mathbb{F})$ to denote the space of functions that can be represented by m-variate polynomials of degree at most d (or simply *degree-d*) over the field $\mathbb{F}$. For a finite string Π and an algorithm $\mathcal{A}$, we will use $\mathcal{A}^{\Pi}$ to denote that the algorithm $\mathcal{A}$ has oracle access to Π. In particular, $\mathcal{A}$ can query the string Π at any coordinate/entry. For us, Π will be evaluation of functions on some domain D, and so $\mathcal{A}^{\Pi}$ means that $\mathcal{A}$ can access the value of $\Pi[\mathbf{x}]$ for any point $\mathbf{x} \in D$. When $D = D_1 \times D_2$, for $\mathbf{x} \in D_1$, we will use $\Pi[\mathbf{x}, \cdot]$ to denote the restriction of Π corresponding to fixing the first coordinate to $\mathbf{x}$. We define $\Pi[\mathbf{x}, \cdot, \cdot]$ similarly when $D = D_1 \times D_2 \times D_3$. For a probabilistic algorithm $\mathcal{A}$, we will use $\mathcal{A}(\mathbf{x}; \mathbf{r})$ to denote the output of the algorithm on input $\mathbf{x}$ and random string $\mathbf{r}$.

For two functions $f, g : S \to T$, we use $\delta(f, g)$ to denote their relative Hamming distance, i.e., the fraction of points in the domain S where they differ. We say two functions $f, g : S \to T$ are δ-close if $\delta(f, g) \leqslant \delta$; else, we say they are δ-far. For a family $\mathcal{F}$ of functions from S to T and a function $f : S \to T$, we use $\delta(f, \mathcal{F})$ to denote the minimum relative distance of f from a function in $\mathcal{F}$, i.e., $\delta(f, \mathcal{F}) = \min_{g \in \mathcal{F}} \delta(f, g)$.

For a subset $S \subseteq \mathbb{F}_q^m$, we will use the notation $\mathbf{x} \sim S$ to denote that $\mathbf{x}$ is uniformly distributed over the subset S. Let $q = 2^t$ for an even integer t, and $\mathbf{Z} = (Z_1, \ldots, Z_t)$ denote t many variables. Let $\mathbb{F}_2[\mathbf{Z}]^{\leqslant i}$ denote the set of multilinear polynomials in $\mathbb{F}_2[\mathbf{Z}]$ of degree at most i. Also, we will use $\mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}$ to denote the family of homogeneous degree 1 polynomials. For a set of polynomials $\{G_1, \ldots, G_k\}$, we will denote the ideal generated by them as $\mathbb{I}(G_1, \ldots, G_k)$. Let $\omega \in \mathbb{F}_q \backslash \{1\}$ be a cube root of the multiplicative identity. We also state a few standard facts about finite fields and low-degree extensions.

Fact 2.3 (Construction of finite fields, see e.g. [Sho05, Chapter 20]). *Let $q = 2^t$ for some $t \in \mathbb{N}$. A description of the finite field $\mathbb{F}_q$ along with a generator can be computed*[9] *in time* $\mathrm{poly}(q)$.

Fact 2.4 (Low-Degree Extension). *Let $H \subseteq \mathbb{F}_q$ be an arbitrary subset and $m \in \mathbb{N}$ denote the number of variables. For every function $f : H^m \to \mathbb{F}_q$, we denote by $\mathrm{LDE}(f) \in \mathbb{F}_q[X_1, \ldots, X_m]$, the* low-degree extension *of f which is the unique polynomial satisfying:*

- *Individual degree*[10] *of $\mathrm{LDE}(f)$ is at most $|H| - 1$.*
- *For every $\mathbf{a} \in H^m$, $\mathrm{LDE}(f)(\mathbf{a}) = f(\mathbf{a})$.*

Furthermore, $\mathrm{LDE}(f)$ can be constructed in time $\mathrm{poly}(|H|^m, \log q)$ via solving a system of linear equations (coefficients of $\mathrm{LDE}(f)$ are the variables and evaluations of f on H^m are the constraints).

We start by recalling polynomial distance lemmas, which are the crucial reason why low-degree polynomials are so useful in constructing PCPs, and they will be the backbone of our analysis.

Theorem 2.5 (Polynomial Distance Lemmas, [Ore22; DL78; Sch80; Zip79]). *Fix a field $\mathbb{F}_q$, degree parameter $d \in \mathbb{N}$, and a non-zero polynomial $P \in \mathbb{F}_q[X_1, \ldots, X_m]$ of degree at most d.*

- *If $d \leqslant q$, we have*

$$\Pr_{\mathbf{a} \sim \mathbb{F}_q^m}[P(\mathbf{a}) \neq 0] \geqslant 1 - \frac{d}{q}.$$

- *If $q = 2$ and P is multilinear, we have*

$$\Pr_{\mathbf{a} \sim \mathbb{F}_2^m}[P(\mathbf{a}) \neq 0] \geqslant \frac{1}{2^d}.$$

An immediate and useful corollary of Theorem 2.5 is the following: If two degree $\leqslant d$ polynomials P and Q agree on strictly more than d/q-fraction of $\mathbb{F}_q^m$, then $P = Q$.

Lines Table For every $m \in \mathbb{N}$, field $\mathbb{F}_q$, and points $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^m$, let $\ell_{\mathbf{a},\mathbf{b}} : \mathbb{F}_q \to \mathbb{F}_q^m$ (read as "line passing through $\mathbf{a}$ with slope $\mathbf{b}$") be defined as $\ell_{\mathbf{a},\mathbf{b}}(\lambda) := \mathbf{a} + \lambda\mathbf{b}$.

Definition 2.6 (Lines Table). *Fix a field $\mathbb{F}_q$. Let $d \in \mathbb{N}$ be the degree parameter and $m \in \mathbb{N}$ be the number of variables. For every degree $\leqslant d$ polynomial $f : \mathbb{F}_q^m \to \mathbb{F}_q$, we define the lines table of f as the function $f_{\mathrm{lines}} : \mathbb{F}_q^{2m} \longrightarrow \mathbb{F}_q^{d+1}$ that maps an input $(\mathbf{a}, \mathbf{b}) \in \mathbb{F}_q^{2m}$ to $f(\ell_{\mathbf{a},\mathbf{b}}(X))$, where X is an indeterminate. We note that $f(\ell_{\mathbf{a},\mathbf{b}}(X))$ is indeed a univariate polynomial of degree at most d in the indeterminate X and can be specified by the $d + 1$ coefficients of $X^0, X^1, \ldots, X^d$.*

[9] i.e., given t as input, output the "truth tables" of the field addition and multiplication. Consequently, we can also find an element of any given order (if it exists).

[10] Individual degree is defined to be the maximum degree of any variable appearing in the polynomial. Hence, the total degree of $\mathrm{LDE}(f)$ is at most $(|H| - 1) \cdot m$.

Throughout the paper, whenever we have a function F that has range in $\mathbb{F}_q^{d+1}$, we will interpret the output as a univariate polynomial of degree at most d.

Local characterization of low-degree polynomials Another reason why low-degree polynomials are useful in PCPs is the fact that the family of low-degree polynomials is the same as the family of functions that are low-degree on every line. We state it formally in the following fact. The fact can be proved using a closely related characterization in [FS95, Lemma 4] and univariate polynomial interpolation. We give a proof for the sake of completeness in Appendix A.

Fact 2.7 (Characterization of Low-Degree Polynomials)**.** *Fix a degree parameter $c \in \mathbb{N}$ and fix a field $\mathbb{F}_q$ with $q > 2c$. Let $\zeta \in \mathbb{F}_q^\times$ be a generator of $\mathbb{F}_q^\times$ and consider the set $\{\zeta_0, \ldots, \zeta_{c+1}\}$ where $\zeta_0 = 0$ and $\zeta_i = \zeta^i$ for $i \in [c+1]$. Let $\eta_0 = -1$ and for $1 \leqslant i \leqslant c+1$ let:*

$$\eta_i \;=\; \left[\prod_{\substack{1 \leqslant j \leqslant c+1 \\ j \neq i}} \frac{-\zeta_j}{\zeta_i - \zeta_j} \right].$$

Then, a function $P : \mathbb{F}_q^m \to \mathbb{F}_q$ is a degree-c polynomial if and only if

$$\sum_{i=0}^{c+1} \eta_i \cdot P(\mathbf{u} + \zeta_i \cdot \mathbf{v}) \;=\; 0, \qquad \text{for every } \mathbf{u}, \mathbf{v} \in \mathbb{F}_q^m,$$

where for scalar $\alpha \in \mathbb{F}_q$, the point $\alpha \cdot \mathbf{y}$ is defined as $(\alpha y_1, \ldots, \alpha y_m)$.

A simple corollary of the above characterization is that we can evaluate a low-degree polynomial at any given point using its evaluations on a line passing through that point. We record it formally below.

Fact 2.8. *Fix a degree-parameter $c \in \mathbb{N}$ and fix a field $\mathbb{F}_q$ of characteristic 2 with $q > 2c$. Let ζ be a generator of $\mathbb{F}_q^\times$ and $\{\zeta_0, \ldots, \zeta_{c+1}\}$ and $\{\eta_0, \ldots, \eta_{c+1}\}$ be the same as in Fact 2.7. For every m-variate degree-c polynomial $P \in \mathbb{F}_q[X_1, \ldots, X_m]^{\leqslant c}$, we have,*

$$P(\mathbf{u}) \;=\; \sum_{i=1}^{c+1} \eta_i \cdot P(\mathbf{u} + \zeta_i \cdot \mathbf{v}) \qquad \text{for every } \mathbf{u}, \mathbf{v} \in \mathbb{F}_q^m.$$

Vanishing of Low-Degree Polynomials on a Subset For a subset $H \subseteq \mathbb{F}_q$ and for every $i \in [m]$, we will define the polynomial $Z_H^{(i)}(\mathbf{X})$ as the vanishing polynomial on H for the i^{th} variable X_i, i.e.,

$$Z_H^{(i)}(\mathbf{X}) \;:=\; \prod_{\lambda \in H} (X_i - \lambda).$$

We will use $Z_H(X_1, \ldots, X_m)$ to denote the tuple of polynomials $(Z_H^{(1)}, \ldots, Z_H^{(m)})$. The next lemma gives an equivalent condition for when a polynomial vanishes on the subset $H^m \subseteq \mathbb{F}_q^m$.

Lemma 2.9 (Combinatorial Nullstellensatz)**.** *Fix any polynomial $P \in \mathbb{F}_q[\mathbf{X}]$. The polynomial P vanishes on*

H^m, *i.e.* $P|_{H^m} \equiv 0$, *if and only if there exist polynomials* $Q_1(\mathbf{X}), \ldots, Q_m(\mathbf{X})$ *with* $\deg(Q_1), \ldots, \deg(Q_m) \leqslant \deg(P) - |H|$ *such that*

$$P(\mathbf{X}) \;=\; \sum_{i=1}^{m} Q_i(\mathbf{X}) \cdot Z_H^{(i)}(\mathbf{X}).$$

The above lemma can be interpreted as giving a certificate for when a low-degree polynomial P vanishes. In particular, the certificate consists of low-degree polynomials $Q_1, \ldots, Q_m$. Similar to the approach in [ABSSW26a], we will define a "vanishing certificate polynomial" consisting of the Q_i's and use that as a certificate of $P|_{H^m} \equiv 0$. Essentially, we will consider the polynomial we get after replacing $Z_H^{(i)}(\mathbf{x})$ by a new variable y_i.

Corollary 2.10 (Vanishing Certificate of a Polynomial)**.** *Fix any polynomial* $P \in \mathbb{F}_q[\mathbf{X}]$*. The polynomial* P *vanishes on* H^m*, if and only if there exists a vanishing certificate polynomial* $\mathcal{M}_P \in \mathbb{F}_q[\mathbf{X}, Y_1, \ldots, Y_m]$ *satisfying the following properties:*

1. *Degree of* $\mathcal{M}_P$ *is at most* $\deg(P)$*.*
2. *The polynomial* $\mathcal{M}_P(\mathbf{X}, \mathbf{0})$ *is identically the zero polynomial. In other words,* $\mathcal{M}_P \in \mathbb{I}(\mathbf{Y})$*.*
3. *The following identity holds:*

$$P(\mathbf{X}) \;=\; \mathcal{M}_P\left(\mathbf{X}, Z_H^{(1)}(\mathbf{X}), \ldots, Z_H^{(m)}(\mathbf{X})\right).$$

3-colorability. We state the 3-colorability language below, which is an NP-complete problem. Note that the choice of 3-coloring as an NP-complete problem instead of one of many others is a choice for simplicity.

Definition 2.11. *The decision problem* 3-COLOR *is the following problem:*
Given a graph $G = (V, E)$ *with* n *vertices, decide whether there exists a proper coloring of* G *using* 3 *colors, i.e. for every edge* $(u, v) \in E$*, the vertices* u *and* v *are assigned different colors.*

2.2 Low-Degree Test and Local Correction over Large Alphabet

In this subsection, we discuss the standard point-vs-line test for low-degree testing from [ALMSS98]. The analysis of the test is a *robust* version of Fact 2.7, i.e., if a function is low-degree on most of the lines in $\mathbb{F}_q^m$, then it is close to a global low-degree polynomial.

We start by recalling the test and state its properties in Theorem 2.12. Recall that for a function with output in $\mathbb{F}_q^{d+1}$, we interpret the output as a degree-d univariate polynomial.

Algorithm 1: Low-Degree Test (with lines table): TabLDT

Input: Degree parameter d and oracle access to (f, f') where $f : \mathbb{F}_q^m \to \mathbb{F}_q$ and $f' : \mathbb{F}_q^{2m} \to \mathbb{F}_q^{d+1}$

Sample $\mathbf{a}, \mathbf{b} \sim \mathbb{F}_q^m$, $\lambda \sim \mathbb{F}_q^\times$

if $f'[(\mathbf{a}, \mathbf{b})](\lambda) \neq f[\mathbf{a} + \lambda \mathbf{b}]$ **then return** *Reject*

return *Accept*

Theorem 2.12 (Low-Degree Test via Lines Table, [ALMSS98, Theorem 45])**.** *There exist absolute positive constants δ_0, C such that for every $\delta < \delta_0$, for every $d, q \in \mathbb{N}$ with $q > Cd^3$, the following holds over $\mathbb{F}_q$.*

1. *If $f \in \mathcal{P}_d(m, \mathbb{F}_q)$, then $\mathsf{TabLDT}_d^{f, f_{\text{lines}}}(; \mathbf{a}, \mathbf{b}, \lambda)$ returns Accept with probability 1, over the randomness of $(\mathbf{a}, \mathbf{b}, \lambda)$.*

2. *For every $f : \mathbb{F}_q^m \to \mathbb{F}_q$ and for every $f' : \mathbb{F}_q^{2m} \to \mathbb{F}_q^{d+1}$, we have*

$$\Pr_{\mathbf{a},\mathbf{b},\lambda}[\mathsf{TabLDT}_d^{f,f'}(; \mathbf{a}, \mathbf{b}, \lambda) \text{ returns } \textit{Reject}] \leqslant \delta \implies \delta(f, \mathcal{P}_d(m, \mathbb{F}_q)) \leqslant 4\delta.$$

Furthermore, $\mathsf{TabLDT}_d^{f,f'}(; \mathbf{a}, \mathbf{b}, \lambda)$ makes 2 oracle queries, uses $\mathcal{O}(m \log q)$ bits of randomness, and runs in time $\mathrm{poly}(m, d, \log q)$.

Remark 2.13. *For low-degree testing, there has been a long line of work on achieving better parameters in terms of field size and soundness guarantee. We refer the interested reader to [HKSS24, Section 1] for a detailed overview of the results of low-degree testing and also for the state-of-the-art parameters (see [HKSS24, Theorem 1.2]). We use the low-degree testing from [ALMSS98] because the algorithm and analysis are stated in the language of the lines table.*

Now we discuss the local correction algorithm for degree d polynomials over $\mathbb{F}_q^m$ from [ALMSS98]. We first describe the local corrector and then analyze it in Theorem 2.14. Recall that for a function with output in $\mathbb{F}_q^{d+1}$, we interpret the output as a degree-d univariate polynomial.

Algorithm 2: Local Corrector (with lines table): TabLC

Input: Degree parameter d, evaluation point $\mathbf{a} \in \mathbb{F}_q^m$, and oracle access to (f, f') where $f : \mathbb{F}_q^m \to \mathbb{F}_q$ and $f' : \mathbb{F}_q^{2m} \to \mathbb{F}_q^{d+1}$

Sample $\mathbf{b} \sim \mathbb{F}_q^m$, $\lambda \sim \mathbb{F}_q^\times$

if $f'[(\mathbf{a}, \mathbf{b})](\lambda) \neq f[\mathbf{a} + \lambda \mathbf{b}]$ **then return** *Reject*

return $f'[(\mathbf{a}, \mathbf{b})](0)$

Theorem 2.14 (Local Correction via Lines Table). *(see [ALMSS98, Proposition 7.2.2.1]) There exist absolute positive constants $\delta_0, C > 0$ such that for every $\delta < \delta_0$, for every $d, q \in \mathbb{N}$ satisfying $q > Cd$, the following holds.*

1. *If f is a degree-d polynomial, then for every $\mathbf{a} \in \mathbb{F}_q^m$, $\mathsf{TabLC}_d^{f, f_{\text{lines}}}(\mathbf{a}; \mathbf{b}, \lambda)$ returns $f(\mathbf{a})$ with probability 1, over the randomness $(\mathbf{b}, \lambda)$.*
2. *Let $f : \mathbb{F}_q^m \to \mathbb{F}_q$ be any function with the condition that there exists a degree-d polynomial P such that $\delta(f, P) \leqslant \delta$. Then for every $f' : \mathbb{F}_q^{2m} \to \mathbb{F}_q^{d+1}$, and every $\mathbf{a} \in \mathbb{F}_q^m$, we have:*
   Either TabLC returns Reject or computes $P(\mathbf{a})$ exactly with high probability, i.e.

$$\Pr_{\mathbf{b},\lambda}\left[\mathsf{TabLC}_d^{f,f'}(\mathbf{a}; \mathbf{b}, \lambda) = P(\mathbf{a}) \quad \textit{OR} \quad \mathsf{TabLC}_d^{f,f'}(\mathbf{a}; \mathbf{b}, \lambda) \ \textit{returns Reject}\right] \geqslant 1 - 2\sqrt{\delta} - \frac{d}{q-1}.$$

Furthermore, $\mathsf{TabLC}_d^{f,f'}$ makes 2 oracle queries, uses $\mathcal{O}(m \log q)$ bits of randomness, and runs in time $\mathrm{poly}(m, d, \log q)$.

3 Two Encodings

In this section, we will explain two different encodings that we will use to encode our PCP proofs. The first encoding will map a degree-d univariate polynomial (for us d will be $\mathrm{poly}(\log n) = \mathrm{poly}(q)$) to a multivariate polynomial of $\mathcal{O}(1)$-degree in, say d^ε variables, for some small constant ε. As an example, say we want to encode a degree-d univariate as a degree-2 polynomial in $2\sqrt{d}$ variables. The idea is to express every degree from 1 to d in its base-$(\sqrt{d}+1)$ representation (the number of digits will be 2). Thus, we use variables $X_{i,j}$ where $i \in \{0, 1\}$ denotes the digit and $j \in \{0, \ldots, \sqrt{d}\}$ denotes the value at that digit. We describe this formally in Section 3.1. In Section 3.2, we discuss its syntactic test and self-correction algorithm.

The second encoding will map an element in $\mathbb{F}_q$ to an $\mathbb{F}_2$-vector of length $2^{\mathrm{poly}(\log q)}$. This encoding will be done via constant-degree Hadamard encoding (higher-degree generalization of the well-known quadratic Hadamard encoding from [ALMSS98]). This then naturally extends to an encoding of a $\mathbb{F}_q$-valued function. We describe this formally in Section 3.3. In Section 3.4, we discuss its syntactic test and self-correction algorithm.

3.1 Encoding Univariate Polynomials using Set-Multilinear Encoding

In this subsection, we discuss our first encoding that encodes a "high"-degree (for us, it $\mathrm{poly}(\log n)$-degree) univariate polynomial to a constant-degree multivariate polynomial, which we refer to as *set-multilinear* encoding. As discussed in the introduction, this transformation allows us to perform a low-degree test with constant queries over an alphabet of size $\mathrm{poly}(\log n)$. In this and following subsections, think of $d = \mathrm{poly}(\log n)$ and c as a small absolute constant.

Definition 3.1 (Set-Multilinear Encoding). *Fix any field $\mathbb{F}$. Let $d, m_1, c \in \mathbb{N}$ with $d < m_1^c$. Consider the*

following $\mathbb{F}$-linear map Ψ_{d,c,m_1}:

$$\Psi_{d,c,m_1} : \mathbb{F}[Y]^{\leqslant d} \to \mathbb{F}[X_{0,0}, \ldots, X_{0,m_1-1}, \ldots, X_{c-1,0}, \ldots, X_{c-1,m_1-1}]$$

$$\Psi_{d,c,m_1}(Y^k) \;=\; X_{0,k_0} \cdots X_{c-1,k_{c-1}}, \qquad \textit{where } 0 \leqslant k_j < m_1 \textit{ are such that } \; k \;=\; \sum_{j=0}^{c-1} k_j \cdot m_1^j.$$

The map Ψ_{d,c,m_1} is defined on all of $\mathbb{F}[Y]^{\leqslant d}$ by linearity. In simple words, Ψ_{d,c,m_1} sends Y^k to the monomial encoding the m_1-ary representation of the integer k (which uses c digits because $d < m_1^c$).
Note that every monomial in $\Psi_{d,c,m_1}(Y^k)$ is set-multilinear with respect to the partition $\{X_{0,\cdot}\} \sqcup \ldots \sqcup \{X_{c-1,\cdot}\}$, i.e., it is multilinear and contains exactly one variable from each $\{X_{j,\cdot}\}$.

We also extend this definition to a family of univariate polynomials of degree d, i.e. if $F : \mathbb{F}^k \to \mathbb{F}[Y]^{\leqslant d}$, then the $\mathbb{F}$-linear map Ψ^*_{d,c,m_1} takes a function F from $\mathbb{F}^k$ to $\mathbb{F}[Y]^{\leqslant d}$, and outputs a function from $\mathbb{F}^k$ to $\mathbb{F}[X_{0,0}, \ldots, X_{c-1,m_1-1}]$ as follows:

$$\Psi^*_{d,c,m_1}(F)(\mathbf{u}) = \Psi_{d,c,m_1}(F(\mathbf{u})).$$

Next, we define the following map that will let us evaluate the univariate polynomial via evaluation of the multivariate polynomial:

$$\Phi_{c,m_1} : \mathbb{F} \to (\mathbb{F}^{m_1})^c$$

$$\lambda \;\mapsto\; \left((1, \lambda, \lambda^2, \ldots, \lambda^{m_1-1}), \ldots, \left(1, \lambda^{m_1^{c-1}}, \lambda^{2m_1^{c-1}}, \ldots, \lambda^{(m_1-1)m_1^{c-1}}\right)\right). \tag{1}$$

From the definition of Ψ_{d,c,m_1} and Φ_{c,m_1}, the following claim is immediate (proved in Appendix A).

Claim 3.2. *Let $d, c, m_1 \in \mathbb{N}$ with $d < m_1^c$. Let Ψ_{d,c,m_1} and Φ_{c,m_1} be as defined in Definition 3.1 and Equation (1). Then for every degree-d univariate polynomial P and for every $\lambda \in \mathbb{F}$, we have,*

$$P(\lambda) \;=\; (\Psi_{d,c,m_1}(P))(\Phi_{c,m_1}(\lambda)).$$

In simple words, evaluation of the cm_1-variate degree-c polynomial $\Psi_{d,c,m_1}(P)$ on the point $\Phi_{c,m_1}(\lambda)$ equals $P(\lambda)$.

Hence Ψ_{d,c,m_1} sends degree-d univariate polynomials to cm_1-variate polynomials of degree-c (in fact, these are set-multilinear polynomials in c parts). For our final PCP construction, we will have $d = \widetilde{\mathcal{O}}(\log^2 n)$ and we will choose c to be an absolute constant.

3.2 Syntactic Test and Self-Correction of Set-Multilinear Encoding

In this subsection, we will discuss the algorithms to do low-degree test and self-correction for degree-c multivariate polynomials, *over the alphabet* $\mathbb{F}_q$. Note that every Ψ_{d,c,m_1} encoding is a degree-c multivariate polynomial. We have already seen algorithms for a very similar task in Algorithm 1 and Algorithm 2. However, their alphabet was "large", i.e. $\mathbb{F}_q^{d+1}$. A large alphabet allowed us to get a constant query

complexity even for $d = \mathrm{poly}(\log n)$. In the degree-$c$ setting, we think of c as a constant, and in that case we do not have to use the lines table. We still get constant many queries (depending on c) and over the alphabet $\mathbb{F}_q$. Before we describe these algorithms, it will be useful to recall Fact 2.7. We start with a syntactic test.

Lemma 3.3 (Standard Low-Degree Test)**.** *There exists an absolute positive constant C such that for every $\delta < 1/2(c+2)^2$, every $c, q \in \mathbb{N}$ where q is a power of 2 larger than Cc^3, the following holds.*
Let ζ be a generator of $\mathbb{F}_q^\times$ and let $\{\zeta_0, \ldots, \zeta_{c+1}\}$ and $\{\eta_0, \ldots, \eta_{c+1}\}$ be the same as in Fact 2.7. Given a function $f : \mathbb{F}_q^{m'} \to \mathbb{F}_q$ and a degree parameter c, we will use $\mathsf{StdLDT}_c^f(;\mathbf{u},\mathbf{v})$ to denote the expression

$$\mathsf{StdLDT}_c^f(;\mathbf{u},\mathbf{v}) \;:=\; \sum_{i=0}^{c+1} \eta_i \cdot f(\mathbf{u} + \zeta_i \cdot \mathbf{v}).$$

1. *If $f \in \mathcal{P}_c(m', \mathbb{F}_q)$ is a polynomial of degree at most c, then*

$$\Pr_{\mathbf{u},\mathbf{v}}\left[\mathsf{StdLDT}_c^f(;\mathbf{u},\mathbf{v}) \;=\; 0\right] \;=\; 1.$$

2. *For every function $f : \mathbb{F}_q^{m'} \to \mathbb{F}_q$,*

$$\Pr_{\mathbf{u},\mathbf{v}}\left[\mathsf{StdLDT}_c^f(;\mathbf{u},\mathbf{v}) \;\neq\; 0\right] \;\leqslant\; \delta \;\implies\; \delta(f, \mathcal{P}_c(m', \mathbb{F}_q)) \;\leqslant\; 2\delta.$$

The proof of Lemma 3.3 follows roughly the same proof as that of [RS96, Theorem 4.1] (by using the characterization from Fact 2.7). This is also a special case of a more general theorem, [KS08, Theorem 2.9].

We now turn to local correction in the same setting.

Lemma 3.4 (Standard Local Correction)**.** *There exists an absolute positive constant C such that for every $\delta < 1/2(c+2)^2$, every $c, q \in \mathbb{N}$ where q is a power of 2 larger than Cc^3, the following holds.*
Let ζ be a generator of $\mathbb{F}_q^\times$ and let $\{\zeta_0, \ldots, \zeta_{c+1}\}$ and $\{\eta_0, \ldots, \eta_{c+1}\}$ be the same as in Fact 2.7. For a function $f : \mathbb{F}_q^{m'} \to \mathbb{F}_q$ with oracle access and a degree parameter c, we will use $\mathsf{StdLC}_c^f(\mathbf{u};\mathbf{v})$ to denote

$$\mathsf{StdLC}_c^f(\mathbf{u};\mathbf{v}) \;:=\; \sum_{i=1}^{c+1} \eta_i \cdot f(\mathbf{u} + \zeta_i \cdot \mathbf{v}).$$

1. *If $f \in \mathcal{P}_c(m', \mathbb{F}_q)$ is a polynomial of degree c, then for every point $\mathbf{u} \in \mathbb{F}_q^{m'}$,*

$$\Pr_{\mathbf{v}}\left[\mathsf{StdLC}_c^f(\mathbf{u};\mathbf{v}) \;=\; f(\mathbf{u})\right] \;=\; 1.$$

2. *Let $f : \mathbb{F}_q^{m'} \to \mathbb{F}_q$ be any function with the condition that there exists a degree c polynomial P such*

that $\delta(f, P) \leqslant \delta$. Then for every point $\mathbf{u} \in \mathbb{F}_q^{m'}$

$$\Pr_{\mathbf{v}}\left[\mathsf{StdLC}_c^f(\mathbf{u};\mathbf{v}) \;=\; P(\mathbf{u})\right] \;\geqslant\; 1-(c+1)\delta.$$

Proof of Lemma 3.4. This is simply due to Fact 2.8 and a union bound. ■

3.3 Encoding Field Elements using Low-Degree Hadamard Codes

In this subsection, we will discuss an encoding of $\mathbb{F}_q$ elements to $2^{\text{poly}\log q}$ long vectors over $\mathbb{F}_2$. This encoding also naturally extends to $\mathbb{F}_q$-valued functions (for our PCP construction, these $\mathbb{F}_q$-valued functions will be low-degree polynomials over $\mathbb{F}_q$). Let $q = 2^t$ (recall that throughout this article, we will always have q a power of 2). In our PCP construction, we will have $t = \mathcal{O}(\log\log n)$. We move to the definition of the encoding now.

Firstly, the inclusion $\mathbb{F}_2 \subseteq \mathbb{F}_q$ gives $\mathbb{F}_q$ the structure of an $\mathbb{F}_2$-vector space, and throughout the paper, let $\rho : \mathbb{F}_q \to \mathbb{F}_2^t$ be an arbitrary but fixed $\mathbb{F}_2$-linear bijection that identifies $\mathbb{F}_q$ as the vector space $\mathbb{F}_2^t$. We will be dealing with the polynomial ring $\mathbb{F}_2[Z_1, \ldots, Z_t]$, which we will denote by $\mathbb{F}_2[\mathbf{Z}]$. We define the *degree-r Hadamard encoding* below. This is also known as degree-r long code in the literature. In this and following subsections, think of r as an absolute constant, and in fact, we will use $r = 3$, which is sufficient for our construction.

Definition 3.5 (Degree-r Hadamard encoding of an $\mathbb{F}_q$ element)**.** *Let $q = 2^t$ and fix a degree parameter $r \in \mathbb{N}$. For an element $u \in \mathbb{F}_q$, its* degree-r Hadamard encoding *is a function $\mathrm{Had}^{(r)}(u) : \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$, defined as follows:*

$$\mathrm{Had}^{(r)}(u)(R) \;:=\; R(\rho(u)), \qquad \text{for every } R \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant r}.$$

The encoding $\mathrm{Had}^{(r)}(u)$ is then the evaluation table of the above function. The length of $\mathrm{Had}^{(r)}(u)$ is equal to the number of degree-r polynomials in t variables over $\mathbb{F}_2$, which is equal to $2^{\binom{t}{\leqslant r}}$.

For $r = 1$ and $r = 2$, we get the usual Hadamard code and the quadratic Hadamard code, respectively. We now extend the previous definition to degree-r Hadamard encoding of $\mathbb{F}_q$-valued functions.

Definition 3.6 (Degree-r Hadamard encoding of an $\mathbb{F}_q$-function)**.** *Let $q = 2^t$, fix a degree parameter $r \in \mathbb{N}$, and fix a variable parameter $k \in \mathbb{N}$. For a function $f : \mathbb{F}_q^k \to \mathbb{F}_q$, the degree-$r$ Hadamard encoding of f is a combined degree-r Hadamard encoding of each of f's evaluations. More formally, it's given by a function as follows:*

$$\mathrm{Had}^{(r)}(f) : \mathbb{F}_q^k \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \;\to \mathbb{F}_2$$
$$\mathrm{Had}^{(r)}(f)(\mathbf{a}, R) \;:=\; \mathrm{Had}^{(r)}(f(\mathbf{a}))(R) \;=\; R(\rho(f(\mathbf{a}))).$$

We also extend this definition to a family of functions as follows. For a family $\mathcal{F}$ of $\mathbb{F}_q$-functions, its degree-r

Hadamard encoding is a set of degree-r Hadamard encodings, defined as:

$$\mathrm{Had}^{(r)}_{\mathcal{F}} \;:=\; \left\{ \mathrm{Had}^{(r)}(f) \;\middle|\; f \in \mathcal{F} \right\}.$$

In other words, degree-r Hadamard encoding of an $f : \mathbb{F}_q^m \to \mathbb{F}_q$ is obtained by applying degree-r long code on every evaluation of f. For every $1 \leqslant j \leqslant r$, we will denote the restriction of a function $H : \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$ to degree-j polynomials by $H|_{\leqslant j} : \mathbb{F}_2[\mathbf{Z}]^{\leqslant j} \to \mathbb{F}_2$.

Why degree-r? From the definition, we see that even degree-1 Hadamard encodings are enough to go from alphabet $\mathbb{F}_q$ to $\mathbb{F}_2$. So why do we need higher degree Hadamard encodings? The reason is that we are constructing PCPs for NP and degree-1 relations do not seem to be strong enough to capture NP. Let us understand this. While constructing a PCP, we will have oracle access to some functions, and we have to check whether these functions satisfy a degree > 1 relation. For example, given a function f as an oracle, we might be interested in knowing whether $(f^3 - 1)$ satisfies some algebraic property. Here we are checking whether a specific degree 3 relation (i.e., $Y^3 - 1$) of f has some algebraic property. Such relations are typical when expressing an NP-complete problem using polynomials. A classic example of an NP-complete problem that looks like this is QUAD-EQ, which asks whether a given set of quadratic equations has a common Boolean solution. Note that degree-1 relations are probably not strong enough to capture NP-completeness, because a system of linear equations can be solved in polynomial time.

Let us imagine an abstract step that occurs in our PCP construction. Suppose $f : \mathbb{F}_q^m \to \mathbb{F}_q$ is a function, $P(Y)$ is a univariate polynomial of degree-r (for example $Y^3 - 1$ for $r = 3$), and we want to test whether $P(f)$ (which is a polynomial in f) satisfies some algebraic property. To test whether $P(f)$ satisfies the algebraic property or not, we might want to query the evaluation of $P(f(\mathbf{a}))$. However, we will only have access to the degree-r Hadamard encoding of f. Instead of $P(f(\mathbf{a}))$, we can consider $\mathrm{Had}^{(1)}(P \circ f)(\mathbf{a}, L)$ for some $L \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$. Therefore, we need to be able to access $\mathrm{Had}^{(1)}(P \circ f)(\mathbf{a}, L)$ using constant many oracle queries to $\mathrm{Had}^{(r)}(f)$ (we want constant many oracle queries so the final query complexity is constant). Claim 3.7 tells us exactly how to do this. Recall that $q = 2^t$ and $\rho : \mathbb{F}_q \to \mathbb{F}_2^t$ is an $\mathbb{F}_2$-linear bijection. The proof is via a straightforward algebraic argument, so we skip the proof here and provide it in Appendix A.

Claim 3.7. *Let $r \in \mathbb{N}$ be a degree parameter, and fix an arbitrary degree-r univariate polynomial $P \in \mathbb{F}_q[Y]$.*
Then for every degree-1 polynomial $L \in \mathbb{F}_2[Z_1, \ldots, Z_t]^{\leqslant 1}$, there exists a degree-$r$ polynomial $\Lambda_{P,L} \in \mathbb{F}_2[Z_1, \ldots, Z_t]^{\leqslant r}$ such that the following holds: For every $\lambda \in \mathbb{F}_q$, we have

$$\mathrm{Had}^{(1)}(P(\lambda))(L) \;=\; \mathrm{Had}^{(r)}(\lambda)(\Lambda_{P,L}).$$

Consequently, for every function $f : \mathbb{F}_q^m \to \mathbb{F}_q$ and every $\mathbf{a} \in \mathbb{F}_q^m$, we have

$$\mathrm{Had}^{(1)}(P \circ f)(\mathbf{a}, L) \;=\; \mathrm{Had}^{(r)}(f)(\mathbf{a}, \Lambda_{P,L}).$$

Finally, if $P(Y) = \eta \cdot Y$ and L is homogeneous, then $\Lambda_{P,L}$ is also homogeneous.

We further note that given P and L, the above polynomial $\Lambda_{P,L}$ can be computed (in its coefficient

representation) in time $\mathrm{poly}(\log^r q)$ by interpolation. In the special case where $P(Y) = \eta \cdot Y$, we write $\Lambda_{\eta,L}$ for $\Lambda_{P,L}$. We also note that $\Lambda_{\eta,-} : \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}} \to \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}$ is a bijection, if η is non-zero.

We will make an observation that will be quite useful later for analyzing different subroutines of our PCP. The following observation says that the distance between two $\mathbb{F}_q$-functions and the distance between their respective low-degree Hadamard encodings are the same up to a constant factor.

Observation 3.8. *Let $f, g : \mathbb{F}_q^k \to \mathbb{F}_q$ be two arbitrary functions and fix any degree parameter $r \in \mathbb{N}$. The following two equalities will come in handy in our calculations later. The first one is:*

$$\delta\left(\mathrm{Had}^{(r)}(f), \mathrm{Had}^{(r)}(g)\right) = \frac{1}{2} \cdot \delta(f, g).$$

This follows because if $f(\mathbf{a}) = g(\mathbf{a})$ for some $\mathbf{a} \in \mathbb{F}_q^k$, then $\mathrm{Had}^{(r)}(f(\mathbf{a})) \equiv \mathrm{Had}^{(r)}(g(\mathbf{a}))$, while if $f(\mathbf{a}) \neq g(\mathbf{a})$, then $\rho(f(\mathbf{a})) \in \mathbb{F}_2^t$ and $\rho(g(\mathbf{a}))$ disagree on at least one coordinate Z_i. So for every polynomial $Q \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant r}$, we have

$$(Q + Z_i)(\rho(f(\mathbf{a}))) - (Q + Z_i)(\rho(g(\mathbf{a}))) \neq Q(\rho(f(\mathbf{a}))) - Q(\rho(g(\mathbf{a}))),$$

which implies that exactly half of the polynomials in $\mathbb{F}_2[\mathbf{Z}]^{\leqslant r}$ disagree on $f(\mathbf{a}), g(\mathbf{a})$. For the same reason, we get:

$$\Pr_{\mathbf{u} \sim \mathbb{F}_q^k, L \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}}\left[\mathrm{Had}^{(r)}(f)(\mathbf{u}, L) \neq \mathrm{Had}^{(r)}(g)(\mathbf{u}, L)\right] = \frac{1}{2} \cdot \delta(f, g)$$

where L is a random linear homogeneous polynomial.

Observation 3.8 is quite useful for us because it allows us to go back and forth between checking whether $f \stackrel{?}{=} g$ and checking whether $\mathrm{Had}^{(r)}(f) \stackrel{?}{=} \mathrm{Had}^{(r)}(g)$. The reader can interpret Observation 3.8 as follows:

- If a test over alphabet $\mathbb{F}_q$ checks if $f \stackrel{?}{=} g$, then its analogue test over alphabet $\mathbb{F}_2$ checks if $\mathrm{Had}^{(r)}(f) \stackrel{?}{=} \mathrm{Had}^{(r)}(g)$.
- If $\mathrm{Had}^{(r)}(f)$ and $\mathrm{Had}^{(r)}(g)$ agree for a random pair in $\mathbb{F}_q^m \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$, then f and g also agree on a random pair.

We will use Observation 3.8 repeatedly in our analysis.

3.4 Syntactic Test and Self-Correction of Low-Degree Hadamard Code

In this subsection, we will discuss two algorithms for degree-r Hadamard encoding of a function: *Syntactic Test* and *Self-Correction*. In the syntactic test, given oracle access to a function, we want to test whether it is close to degree-r Hadamard encoding of an element in $\mathbb{F}_q$. In self-correction, we are promised that the given oracle is close to degree-r Hadamard encoding, and we want to compute it correctly for a specific degree-r polynomial. We start by discussing the syntactic test.

Syntactic Test We are interested in the following task:

Given oracle access to a function $H : \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$, decide whether H is close to degree-r Hadamard encoding of some element in $\mathbb{F}_q$.

For $r = 1$, [BLR93] gave a syntactic test for the Hadamard code (see Theorem B.1). For $r = 2$, [ALMSS98, Proposition 34] gave a syntactic test for the quadratic Hadamard code. We extend the test of [ALMSS98] to higher degrees.
Note that a function $H : \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$ is a degree-r Hadamard encoding of an element in $\mathbb{F}_q$ if and only if it is a restriction of a ring homomorphism $\mathbb{F}_2[\mathbf{Z}] \to \mathbb{F}_2$. We state it formally in the following claim.

Claim 3.9 (Local characterization of degree-r Hadamard encoding)**.** *Let $H : \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$. There exists a $u \in \mathbb{F}_q$ such that $H = \mathrm{Had}^{(r)}(u)$ if and only if H satisfies the following three properties:*

1. *The map H preserves addition, i.e., for any two polynomials $P_1, P_2 \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant r}$, we have,*

$$H(P_1 + P_2) \;=\; H(P_1) \;+\; H(P_2).$$

2. *The map H preserves the multiplicative unit, i.e.,*

$$H(1) \;=\; 1.$$

3. *The map H preserves multiplication, i.e., for any two polynomials $P, R \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant r}$ satisfying $\deg(P) + \deg(R) \leqslant r$, we have,*

$$H(P \cdot R) \;=\; H(P) \cdot H(R).$$

   Note that this condition only matters for $r \geqslant 2$.

Proof of Claim 3.9. For any $u \in \mathbb{F}_q$, it is easy to verify that $\mathrm{Had}^{(r)}(u)$ satisfies these three properties. Now suppose a function H satisfies these three properties. This implies H is fully determined by its values on $1, z_1, \ldots, z_t$ (since they are the generators for $\mathbb{F}_2[\mathbf{Z}]$). Define $u \in \mathbb{F}_q$ as $\rho^{-1}(H(z_1), \ldots, H(z_t))$. Then it is easy to verify that $H = \mathrm{Had}^{(r)}(u)$. This finishes the proof of Claim 3.9. ■

The above three properties, especially the first and the third properties, give us a local characterization of degree-r Hadamard encodings. We design a syntactic test in Algorithm 3, which is a *robust* version of the above local characterizations. We also need an additional property from our syntactic test: if a function H passes the syntactic test with high probability, then not only is it close to a degree-r Hadamard encoding $\mathrm{Had}^{(r)}(u)$, but H's restriction to degree $\leqslant 1$ polynomials is also close to a $\mathrm{Had}^{(1)}(u)$. This will be useful in designing our low-degree test and local correction algorithm over $\mathbb{F}_2$. We now describe the syntactic test.

Algorithm 3: Syntactic Test for Low-Degree Hadamard Codes: HadTest

Input: Oracle access to $H : \mathbb{F}_2[Z_1, \ldots, Z_t]^{\leqslant r} \to \mathbb{F}_2$ and degree parameter r

1 Sample $R, R' \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant r}$, $P \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant (r-1)}$ and $L \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$

2 **if** $H[R + R'] \neq H[R] + H[R']$ **then return** *Reject*

3 **if** $H[1 + R] - H[R] \neq 1$ **then return** *Reject*

4 **if** $H[L + R] - H[R] \neq H[L]$ **then return** *Reject*

5 **if** $H[L \cdot P + R] - H[R] \neq (H[L + R] - H[R]) \cdot (H[P + R] - H[R])$ **then return** *Reject*

6 **return** *Accept*

Lemma 3.10 (Syntactic Test for Generalized Hadamard Code). *Fix a degree parameter $r \in \mathbb{N}$. There exists a δ_0, such that for every $\delta < \delta_0$ and for every $H : \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$, the following holds.*

1. ***Completeness:*** *If there exists a $u \in \mathbb{F}_q$ such that H is $\mathrm{Had}^{(r)}(u)$, then $\mathsf{HadTest}_r^H$ returns Accept with probability 1 over the internal randomness of HadTest.*

2. ***Soundness:*** *Suppose that*

$$\Pr[\mathsf{HadTest}_r^H = \textit{Reject}] \leqslant \delta.$$

Then there exists an element $u \in \mathbb{F}_q$, such that:

- $\delta(H, \mathrm{Had}^{(r)}(u)) \leqslant \delta$
- $\delta(H|_{\leqslant 1}, \mathrm{Had}^{(1)}(u)) \leqslant 3\delta$, *where $H|_{\leqslant 1}$ is the restriction of H to $\mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$.*

Furthermore, $\mathsf{HadTest}_r^H$ makes 8 oracle queries, uses $\mathcal{O}\left(\binom{t}{\leqslant r}\right)$ bits of randomness, and runs in time $t^{\mathcal{O}(r)}$.

The proof of Lemma 3.10 is analogous to the analysis of the syntactic test for the quadratic Hadamard code in [ALMSS98, Proposition 34]. We present a proof for the sake of completeness in Appendix B.

Self-Correction. We are interested in the following task.

Given oracle access to a function $H : \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$ that is close to degree-r Hadamard encoding of $u \in \mathbb{F}_q$, compute $\mathrm{Had}^{(r)}(u)(P)$ for a particular P.

Firstly, for the task to make sense, δ should be small enough to guarantee the existence of a unique u. Using Observation 3.8, we get that for any two distinct $u, u' \in \mathbb{F}_q$, their respective degree-r Hadamard encodings differ on at least $1/2$-fraction, i.e., $\delta(\mathrm{Had}^{(r)}(u), \mathrm{Had}^{(r)}(u')) \geqslant 1/2$. Thus if $\delta < 1/4$, there can be at most a single u whose degree-r Hadamard encoding is δ-close to H.
Suppose H is δ-close to $\mathrm{Had}^{(r)}(u)$, for some $u \in \mathbb{F}_q$ (u is unknown) and we are interested in computing $\mathrm{Had}^{(r)}(u)(P)$ using queries to H. For this, we have the following simple algorithm:

Sample a uniformly random polynomial $R \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant r}$ and compute $H[P+R]-H[R]$.

Since R is uniformly random, $P+R$ is also uniformly random in $\mathbb{F}_2[\mathbf{Z}]^{\leqslant r}$. Thus we have,

$$\Pr_{R\sim\mathbb{F}_2[\mathbf{Z}]^{\leqslant r}}[H[R] \neq \mathrm{Had}^{(r)}(u)(R)] \leqslant \delta \quad \text{and} \quad \Pr_{R\sim\mathbb{F}_2[\mathbf{Z}]^{\leqslant r}}[H[P+R] \neq \mathrm{Had}^{(r)}(u)(P+R)] \leqslant \delta.$$

By union bound, H agrees with $\mathrm{Had}^{(r)}(u)$ on both R and $P+R$ with probability at least $1-2\delta$. Thus $H[P+R]-H[R]$ equals $\mathrm{Had}^{(r)}(u)(P)$ with probability at least $1-2\delta$.

4 Low-Degree Test over Binary Alphabet

In this section, we will discuss our new low-degree test that works over the alphabet $\mathbb{F}_2$ (described later in Algorithm 5). In the following subsection, Section 4.1, we develop some intuition behind our final low-degree test. At the beginning of Section 4.2, we give more intuition behind our test, before giving the formal algorithm in Algorithm 5. We state its guarantee in Theorem 4.1 and then prove it.
As mentioned in the introduction, we note that the low-degree test is the bottleneck in getting a PCP without proof compositions. We overcome this by using the two encodings from the previous section. Before stating the algorithm, which might seem cryptic at first glance, we explain the idea behind it. As a warmup, we give a low-degree test over *alphabet* $\mathbb{F}_q$ (instead of directly going to alphabet $\mathbb{F}_2$). Throughout this section, degree parameter d should be thought of as $\mathrm{poly}(\log n)$, i.e., it's a growing parameter.

4.1 Warmup: Low-Degree Test over alphabet $\mathbb{F}_q$

The goal of this subsection is to *design a low-degree test with constant query complexity over alphabet* $\mathbb{F}_q$. Recall that TabLDT (see Algorithm 1) is a low-degree test with constant query complexity but over the alphabet $\mathbb{F}_q^{d+1}$. The reason the alphabet is $\mathbb{F}_q^{d+1}$ is that every entry of the lines table is a degree-d univariate polynomial, written as a single element in $\mathbb{F}_q^{d+1}$ (and that's why the query complexity is constant). Observe that in TabLDT, after a line is queried, we only need evaluation of that degree-d univariate polynomial at a single point.

The idea is to encode degree-d univariate polynomials such that any evaluation of the polynomial can be inferred by reading constant-many locations of the encoding (even in presence of few errors).

We want that any evaluation can be inferred from constant-many locations so that the query complexity remains constant. We would also need that the encoding is locally testable with constant queries. To do this, we use the set-multilinear encoding Ψ_{d,c,m_1} from Definition 3.1. Here, think of c as a constant and m_1 as $\Theta(d^{1/c})$. The encoding maps a degree-d univariate polynomial $L(Y)$ to a constant-degree multivariate polynomial $\Psi_{d,c,m_1}(L)$. Why can any evaluation of L be inferred from constant-many evaluations of $\Psi_{d,c,m_1}(L)$? This is because $L(\lambda)$ is equal to $\Psi_{d,c,m_1}(L)$ evaluated at the point $\Phi_{c,m_1}(\lambda)$ (see Claim 3.2) and the fact that any constant-degree polynomial's evaluation at a point can be recovered from its evaluations on constant-many other points, even in the presence of error (see Lemma 3.4). Since d, c, m_1 are clear in our context, we will simply denote these maps by Ψ, Φ, and Ψ^* (recall Ψ^* from Definition 3.1). Next, we describe the expected oracles and the queries.

Expected Oracles and Queries Suppose $P \in \mathcal{P}_d(m, \mathbb{F}_q)$ is a degree-d polynomial and let P_{lines} from $\mathbb{F}_q^{2m}$ to $\mathbb{F}_q^{d+1}$ be its lines table. Recall that $\Psi^*(P_{\text{lines}})$ is the following function:

$$\begin{aligned}\Psi^*(P_{\text{lines}}) : \mathbb{F}_q^{2m} \times \mathbb{F}_q^{cm_1} &\to \mathbb{F}_q \\ ((\mathbf{a}, \mathbf{b}), \mathbf{u}) &\mapsto \Psi(P_{\text{lines}}(\mathbf{a}, \mathbf{b}))(\mathbf{u}).\end{aligned}$$

By definition of Ψ, for every pair $(\mathbf{a}, \mathbf{b})$, $\Psi(P_{\text{lines}}(\mathbf{a}, \mathbf{b}))(\mathbf{X})$ is a degree-c polynomial in $\mathbf{X}$-variables. Suppose the verifier has oracle access to $f : \mathbb{F}_q^m \to \mathbb{F}_q$ and to $g : \mathbb{F}_q^{2m} \times \mathbb{F}_q^{cm_1} \to \mathbb{F}_q$ and wants to test whether f is a degree-d polynomial or far from every degree-d polynomial. The idea is to simulate TabLDT. Following TabLDT (see Algorithm 1) and Claim 3.2, the verifier samples a random pair $(\mathbf{a}, \mathbf{b}) \sim \mathbb{F}_q^{2m}$, samples a random $\lambda \sim \mathbb{F}_q^\times$ and would like to check whether

$$f[\mathbf{a} + \lambda \cdot \mathbf{b}] \text{ is equal to } g[(\mathbf{a}, \mathbf{b}), \Phi(\lambda)]. \tag{2}$$

To make the above check work, the verifier needs to make a couple of modifications.

- *Syntactic Test on g:* The above check is trying to simulate the TabLDT test. One of the reasons why TabLDT works is that the lines table consists of degree-d univariate polynomials, and not arbitrary functions. So here also, we need to first check that g is obtained from applying Ψ to a lines table. In other words, we have to test, for every pair $(\mathbf{a}, \mathbf{b})$, whether the function $g[(\mathbf{a}, \mathbf{b}), \cdot] : \mathbb{F}_q^{cm_1} \to \mathbb{F}_q$ is a degree-c polynomial. Equivalently, the verifier would like to do a low-degree test on $g[(\mathbf{a}, \mathbf{b}), \cdot]$ with degree parameter c, for uniformly random $(\mathbf{a}, \mathbf{b})$. Since c is a constant, we simply use StdLDT for this (see Lemma 3.3) and perform this test in constant queries. Note that here we are using the local testability of our encoding Ψ, which is essentially the local testability of Reed-Muller codes.

- *Query point is not uniformly random:* If $g[(\mathbf{a}, \mathbf{b}), \cdot]$ passes the above syntactic test, then we are only guaranteed that $g[(\mathbf{a}, \mathbf{b}), \cdot]$ is close to a degree-c polynomial $Q_{\mathbf{a},\mathbf{b}} : \mathbb{F}_q^{cm_1} \to \mathbb{F}_q$. To do the check in Equation (2), the verifier wants $Q_{\mathbf{a},\mathbf{b}}(\Phi(\lambda))$, but has oracle access to only $g[(\mathbf{a}, \mathbf{b}), \cdot]$. Since the point $\Phi(\lambda)$ is not a uniformly random point in $\mathbb{F}_q^{cm_1}$, querying $g[(\mathbf{a}, \mathbf{b}), \cdot]$ might not be equal to $Q_{\mathbf{a},\mathbf{b}}(\Phi(\lambda))$ with high probability. Thus the verifier self-corrects $g[(\mathbf{a}, \mathbf{b}), \cdot]$ at the point $\Phi(\lambda)$. Since we are self-correcting a function close to a constant-degree polynomial, we can use StdLC for this (see Lemma 3.4) and perform this self-correction with constant queries.

We are now ready to state our low-degree test over the alphabet $\mathbb{F}_q$.

Algorithm 4: Warmup: Low-Degree Test over $\mathbb{F}_q$ alphabet

Input: Degree parameter d, Oracles $f : \mathbb{F}_q^m \to \mathbb{F}_q$, $g : \mathbb{F}_q^{2m} \times \mathbb{F}_q^{cm_1} \to \mathbb{F}_q$

1 Sample $(\mathbf{a}, \mathbf{b}) \sim \mathbb{F}_q^{2m}$, sample $\mathbf{u}, \mathbf{v} \sim \mathbb{F}_q^{cm_1}$, sample $\lambda \sim \mathbb{F}_q^\times$

2 **if** $\text{StdLDT}_c^{g[(\mathbf{a},\mathbf{b}),\cdot]}(; \mathbf{u}, \mathbf{v}) \neq 0$ **then return** *Reject*

3 **if** $\text{StdLC}_c^{g[(\mathbf{a},\mathbf{b}),\cdot]}(\Phi(\lambda); \mathbf{u}) \neq f[\mathbf{a} + \lambda \cdot \mathbf{b}]$ **then return** *Reject*

4 **return** *Accept*

In the next subsection, we describe the low-degree test that emulates Algorithm 1 over the binary alphabet. For that, we will extend the above warm-up test by replacing the checks with their low-degree Hadamard encodings.

4.2 Low-Degree Test over $\mathbb{F}_2$

We are now ready to describe our *low-degree test with constant query complexity over the alphabet* $\mathbb{F}_2$. The warmup-test (Algorithm 4) in the previous subsection is over the alphabet $\mathbb{F}_q$, because the oracles are $\mathbb{F}_q$-functions. Note that the checks in the warmup-test are checking whether two elements in $\mathbb{F}_q$ agree or not.

The idea is to encode elements of $\mathbb{F}_q$ over alphabet $\mathbb{F}_2$ with good distance.

By good distance, we mean a constant relative distance. If the encoding has constant distance, then we can check whether two $\mathbb{F}_q$ elements are equal or not by reading constant-many bits of their respective encodings. To do this, we use degree-r Hadamard encoding[11] $\mathrm{Had}^{(r)}(\cdot)$ (see Definition 3.6). For us, r would always be an absolute constant[12]. As before, we would also need that this encoding is locally testable, and we know degree-r Hadamard encoding is locally testable (see Algorithm 3). The expected oracles will be degree-r Hadamard encoding applied to each of the oracles used in the warmup-test (i.e., Hadamard encoding applied to each evaluation of $\mathbb{F}_q$-functions) and the queries will be random entries of degree-r Hadamard encoding of appropriate $\mathbb{F}_q$ elements.

Expected Oracles and Queries Suppose $P \in \mathcal{P}_d(m, \mathbb{F}_q)$ is a degree-d polynomial and let $\Psi^*(P_{\text{lines}})$ be the function defined in the previous subsection. The expected oracles are:

$$\mathrm{Had}^{(r)}(P) : \mathbb{F}_q^m \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$$
$$\mathrm{Had}^{(r)}(\Psi^*(P_{\text{lines}})) : \mathbb{F}_q^{2m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2.$$

Suppose the verifier has access to oracles $f : \mathbb{F}_q^m \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$ and $f' : \mathbb{F}_q^{2m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$. As mentioned above, to check whether two $\mathbb{F}_q$ elements are equal or not, due to good distance, it is sufficient to check if their respective degree-r Hadamard encodings agree on a random coordinate or not. With this, the idea is to do as in the warmup-test, but at a random coordinate in the Hadamard encoding. In other words, if the verifier wants to check whether $g_1(\mathbf{u}) = g_2(\mathbf{u})$ for two $\mathbb{F}_q$-functions g_1, g_2, then it suffices to check whether $\mathrm{Had}^{(r)}(g_1(\mathbf{u}))(L) = \mathrm{Had}^{(r)}(g_2(\mathbf{u}))(L)$ for a random $L \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant r}$.

To make the above check work, the verifier needs to make a few modifications:

- *Hadamard Syntactic Test:* The above idea only works if the oracles are actual degree-r Hadamard encodings. So the verifier would like to do a degree-r Hadamard test on $f[\mathbf{a}, \cdot]$ and $f'[(\mathbf{a}, \mathbf{b}), \mathbf{u}, \cdot]$ for random $\mathbf{a}, \mathbf{b}, \mathbf{u}$. For this, we use HadTest (see Algorithm 3), and since r is a constant, this can be performed with a constant number of queries.

- *Linearity holds for restricted queries:* To do an analogue of StdLDT and StdLC, a natural step would be to choose a random $Q \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant r}$ and query with respect to Q's coordinate in the Hadamard encodings. That is, say if StdLDT queries $h(\mathbf{u})$ for some function h, then we query $\mathrm{Had}^{(r)}(h)(\mathbf{u}, Q)$. However, this does not work directly because in StdLDT and StdLC, we have to take a linear

[11] The reason we are using degree-r Hadamard encoding instead of Hadamard encoding is that in a PCP for an NP-complete problem, we would need to check degree-r relations on the queried bits, see Section 3.3.

[12] In fact, for our construction of PCP, $r = 3$ suffices.

combination of the queries, but for arbitrary $\alpha, \beta \in \mathbb{F}_q$, $\mathrm{Had}^{(r)}(\alpha)(Q) + \mathrm{Had}^{(r)}(\beta)(Q)$ is not necessarily equal to $\mathrm{Had}^{(r)}(\alpha+\beta)(Q)$. This however holds when Q is a homogeneous linear polynomial. Thus the queries have to be with respect to a random homogeneous linear polynomial.

- *Scalar Multiplication:* Similar to the above point, if we do StdLDT or StdLC for some function h, then we need to evaluate $\eta_i \cdot h(\mathbf{u})$ at some homogeneous linear polynomial L, even though we only have query access to the Hadamard encoding of h. We then use Claim 3.7, which gives us a homogeneous linear polynomial $\Lambda_{\eta_i,L}$ (with slight abuse of notation) such that $\mathrm{Had}^{(r)}(\alpha)(\Lambda_{\eta_i,L}) = \mathrm{Had}^{(r)}(\eta_i \cdot \alpha)(L)$ for all $\alpha \in \mathbb{F}_q$. We note that if L is uniformly random, then $\Lambda_{\eta_i,L}$ is also uniformly random.

We now give the low-degree test over $\mathbb{F}_2$.

Algorithm 5: Low-Degree Test over the alphabet $\mathbb{F}_2$: NewLDT

Input: Parameters d, r, Oracles $f : \mathbb{F}_q^m \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$, $f' : \mathbb{F}_q^{2m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$

1 Sample $(\mathbf{a}, \mathbf{b}) \sim \mathbb{F}_q^{2m}$, $\mathbf{u}, \mathbf{v} \sim \mathbb{F}_q^{cm_1}$, $\lambda \sim \mathbb{F}_q^{\times}$, and $L \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}$

2 Run $\mathsf{HadTest}_r^{f[\mathbf{a},\cdot]}$ and $\mathsf{HadTest}_r^{f'[(\mathbf{a},\mathbf{b}),\mathbf{u},\cdot]}$

3 Check if $f'[(\mathbf{a},\mathbf{b}),\mathbf{u},\Lambda_{\eta_0,L}] + \sum_{i=1}^{c+1} f'[(\mathbf{a},\mathbf{b}),\mathbf{u}+\zeta^i\mathbf{v},\Lambda_{\eta_i,L}] \stackrel{?}{=} 0$

4 Check if $\sum_{i=1}^{c+1} f'[(\mathbf{a},\mathbf{b}),\Phi(\lambda)+\zeta^i\mathbf{v},\Lambda_{\eta_i,L}] \stackrel{?}{=} f[\mathbf{a}+\lambda\mathbf{b},L]$, where Φ is as in Equation (1).

5 **if** *any of the above tests returns* REJECT **then return** REJECT

6 **return** ACCEPT

Theorem 4.1 (Low-Degree Test over $\mathbb{F}_2$)**.** *There exist absolute constants $C, \delta_0 > 0$ such that for every choice of parameters $m, m_1, c, d, d', q, r \in \mathbb{N}$ satisfying $d' = 4cd$, $m_1 \geqslant c \geqslant 2$, $(m_1-1)^c \leqslant d < m_1^c$, q is a power of 2 and $q > Cd'^3$ and $\delta < \delta_0$, the following holds:*

1. ***Completeness:*** *If $f = \mathrm{Had}^{(r)}(P)$ for a degree-d polynomial $P \in \mathcal{P}_d(m, \mathbb{F}_q)$ and $f' = \mathrm{Had}^{(r)}(P')$ where P' is $\Psi^*_{d,c,m_1}(P_{\mathrm{lines}})$ (recall Ψ^*_{d,c,m_1} from Definition 3.1), then $\mathsf{NewLDT}_{d,r}^{f,f'}$ returns* ACCEPT *with probability 1.*

2. ***Soundness:*** *For every f, f' the following holds. If $\mathsf{NewLDT}_{d,r}^{f,f'}$ returns* REJECT *with probability at most δ, then there exists a degree-d' polynomial $P \in \mathcal{P}_{d'}(m, \mathbb{F}_q)$ such that*

$$\delta\left(f,\ \mathrm{Had}^{(r)}(P)\right),\ \delta\left(f|_{\leqslant 1},\ \mathrm{Had}^{(1)}(P)\right) \leqslant \mathcal{O}_c(\delta).$$

Furthermore, $\mathsf{NewLDT}_{d,r}^{f,f'}$ makes $\mathcal{O}_r(c)$ queries to the oracles (f, f'), uses $\mathcal{O}_r(m \log q + cm_1 \log q + (\log q)^r)$ bits of randomness, and runs in time $\mathrm{poly}(m_1^c, \log q)$.

Remark 4.2. *In the soundness condition of Theorem 4.1, the guarantee is only that f is close to a degree-d' polynomial, where d'* is more than *d. Note that in Theorem 2.12, the soundness guarantee was that the oracle is close to a degree-d polynomial, which is stronger than the above guarantee. The reason is roughly as follows. In Theorem 2.12, every entry of the lines table is a degree-d univariate polynomial. In* NewLDT*, Line* 3 *checks that every entry of the supposed lines table is a degree-c polynomial in cm_1 variables, and then after composing it with Φ, we get degree-d' univariate polynomial. Thus while reducing Theorem 4.1 to Theorem 2.12, we are only able to guarantee that every entry of the lines table is a degree-d' univariate polynomial. We can get degree-d by having additional tests that also test that the degree-c polynomial is set-multilinear (i.e. the monomials have same degree pattern as expected). We skip these tests as they are not crucial for proving the PCP Theorem.*

Proof Overview As mentioned earlier, NewLDT is a simulation of TabLDT via encodings Ψ and $\mathrm{Had}^{(r)}(\cdot)$. Completeness follows directly from the definitions. The harder task is to show the soundness of NewLDT, and for that we will fall back on the soundness of TabLDT. In simple words, we will define two functions $\mathrm{Corr}_f : \mathbb{F}_q^m \to \mathbb{F}_q$ (read as "corrected f") and $F : \mathbb{F}_q^{2m} \to \mathbb{F}_q^{d'+1}$ using the oracles f and f'. Here Corr_f, F are proxies of f, f' on which we will apply TabLDT. We will argue that if NewLDT returns Reject with small probability, then $\mathsf{TabLDT}^{\mathrm{Corr}_f, F}$ also returns Reject with small probability. Then Theorem 2.12 guarantees the existence of a low-degree polynomial P that is close to Corr_f, and then we also get $\mathrm{Had}^{(r)}(P)$ is close to the input oracle f. We now discuss the proof in a bit more detail. Assume NewLDT returns Reject with probability $\leqslant \delta$.

- We define Corr_f pointwise, i.e., $\mathrm{Corr}_f(\mathbf{a})$ is chosen such that its Hadamard encoding minimizes the distance to $f[\mathbf{a}, \cdot]$. To define F, for every pair $(\mathbf{a}, \mathbf{b})$, we first choose a degree-c polynomial $Q_{\mathbf{a},\mathbf{b}}$ in cm_1 variables whose Hadamard encoding minimizes the distance to $f'[(\mathbf{a}, \mathbf{b}), \cdot, \cdot]$, and then apply the Φ map on $Q_{\mathbf{a},\mathbf{b}}$.

- $\mathsf{HadTest}_r^{f[\mathbf{a},\cdot]}$ returns Reject with small probability implies f is close to $\mathrm{Had}^{(r)}(\mathrm{Corr}_f)$. We prove this in Claim 4.6.

- $\mathsf{HadTest}_r^{f'[(\mathbf{a},\mathbf{b}),\cdot,\cdot]}$ and Line 3 together return Reject with small probability implies that for most of the pairs $(\mathbf{a}, \mathbf{b})$, Hadamard encoding of $Q_{\mathbf{a},\mathbf{b}}$ is close to $f'[(\mathbf{a}, \mathbf{b}), \cdot, \cdot]$. We prove this in Lemma 4.3. Since $Q_{\mathbf{a},\mathbf{b}}$ is of degree-c, $Q_{\mathbf{a},\mathbf{b}} \circ \Phi$ is a degree-d' univariate polynomial. So we get for every $(\mathbf{a}, \mathbf{b})$, $F(\mathbf{a}, \mathbf{b})$ is a degree-d' univariate polynomial.

- From the above two points and Observation 3.8, Line 4 is roughly the same as checking if $\mathrm{Corr}_f(\mathbf{a} + \lambda\mathbf{b}) \stackrel{?}{=} F(\mathbf{a}, \mathbf{b})(\lambda)$. Theorem 2.12 and the second item finish the soundness analysis.

Before we delve into the proof of Theorem 4.1, it will be convenient for us to prove the following lemma, which will be used as an intermediate step in this and later proofs.
Lemma 4.3 says that if a function $g : \mathbb{F}_q^k \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$ (i.e., the same domain and codomain as a degree-r Hadamard encoding of a function on $\mathbb{F}_q^k$) passes $\mathsf{HadTest}_r^g$ and an analogue of StdLDT with high probability, then the local correction of g at any point $\mathbf{u} \in \mathbb{F}_q^k$ will, with high probability, return the value of the nearest polynomial.

Lemma 4.3. *Let $g : \mathbb{F}_q^k \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$ be a function, and let $P : \mathbb{F}_q^k \to \mathbb{F}_q$ be the degree-c polynomial, whose degree-*1 *Hadamard encoding is closest to g. Let ζ_i, η_i be as in Fact 2.7. Suppose g satisfies the following*

two conditions:

$$\Pr_{\mathbf{u}}[\mathsf{HadTest}_r^{g[\mathbf{u},\cdot]} \text{ returns } \textsc{Reject}] \leqslant \delta_1 \text{ and } \Pr_{\mathbf{u},\mathbf{v},L\sim\mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}}\left[\sum_{i=0}^{c+1} g[\mathbf{u}+\zeta_i\mathbf{v},\Lambda_{\eta_i,L}] \neq 0\right] \leqslant \delta_1,$$

where the first probability is also over the internal randomness of HadTest*. Then for every* $\mathbf{u}\in\mathbb{F}_q^k$*, we have,*

$$\Pr_{\mathbf{v},L\sim\mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}}\left[\sum_{i=1}^{c+1} g[\mathbf{u}+\zeta_i\mathbf{v},\Lambda_{\eta_i,L}] \neq \mathrm{Had}^{(1)}(P)(\mathbf{u},L)\right] \leqslant \mathcal{O}_c(\delta_1).$$

We defer the proof of the above lemma to Section 4.3 and proceed with the proof of Theorem 4.1.

Proof of Theorem 4.1. We start by discussing the completeness of Algorithm 5.

Completeness Firstly, since both f, f' are the Hadamard encodings of functions, both the tests in Line 2 return Accept. Now fix an arbitrary $(\mathbf{a},\mathbf{b})\in\mathbb{F}_q^{2m}$. Let $g(Y)=P^{(d)}_{\mathrm{lines}}(\mathbf{a},\mathbf{b})(Y)$. Since $g(Y)$ is a degree-d univariate polynomial, $h(\mathbf{X}):=\Psi(g(Y))$ is a degree-c polynomial. Fix arbitrary $\mathbf{u},\mathbf{v}\in\mathbb{F}_q^{cm_1}$. Fact 2.7 then implies

$$\sum_{i=0}^{c+1}\eta_i h(\mathbf{u}+\zeta_i\mathbf{v}) = 0.$$

Then for every *homogeneous* linear polynomial $L\in\mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}$, we get,

$$\sum_{i=0}^{c+1}\Lambda_{\eta_i,L}(\rho(h(\mathbf{u}+\zeta_i\mathbf{v}))) = L\left(\sum_{i=0}^{c+1}\rho(\eta_i h(\mathbf{u}+\zeta_i\mathbf{v}))\right) = L\left(\rho\left(\sum_{i=0}^{c+1}\eta_i h(\mathbf{u}+\zeta_i\mathbf{v})\right)\right) = 0.$$

This shows that the test in Line 3 always returns Accept. Now it remains to show that Line 4 also always returns Accept. By an analogous argument, Line 4 corresponds to

$$L(\rho(h(\Phi(\lambda)))) \stackrel{?}{=} L(\rho(P(\mathbf{a}+\lambda\mathbf{b}))) = L(\rho(g(\lambda))).$$

Hence, by using the fact that $h(\Phi(\lambda))=g(\lambda)$ and the completeness part of TabLDT (Theorem 2.12), Line 4 also always returns Accept. This finishes the discussion for completeness of NewLDT.

Soundness We have the following events:

- Let $\mathcal{E}_1$ denote the event $\mathsf{HadTest}_r^{f[\mathbf{a},\cdot]}$ returns Reject.
- Let $\mathcal{E}_2$ denote the event $\mathsf{HadTest}_r^{f'[(\mathbf{a},\mathbf{b}),\mathbf{u},\cdot]}$ returns Reject.
- Let $\mathcal{E}_3$ denote the event $\sum_{i=0}^{c+1} f'[(\mathbf{a},\mathbf{b}),\mathbf{u}+\zeta_i\mathbf{v},\Lambda_{\eta_i,L}]$ is not equal to 0.
- Let $\mathcal{E}_4$ denote the event $\sum_{i=1}^{c+1} f'[(\mathbf{a},\mathbf{b}),\ \Phi(\lambda)+\zeta_i\mathbf{v},\ \Lambda_{\eta_i,L}]$ is not equal to $f[\mathbf{a}+\lambda\mathbf{b},\ L]$.

We want to show that if all the events happen with probability at most δ, then the soundness guarantee of Theorem 4.1 holds.

Defining the proxies We start by defining the proxy for f, and for f', which will represent the underlying function and its lines table respectively.

Definition 4.4 (Correction for f)**.** *For every $\mathbf{a} \in \mathbb{F}_q^m$, let $\mathbf{v_a} \in \mathbb{F}_q$ denote the $\mathbb{F}_q$-element whose degree-r Hadamard encoding is closest to $f[\mathbf{a}, \cdot]$, i.e.,*

$$\mathbf{v_a} \;:=\; \arg\min_{\mathbf{v}\in\mathbb{F}_q} \delta\left(f[\mathbf{a},\cdot],\ \mathrm{Had}^{(r)}(\mathbf{v})\right).$$

We then define $\mathrm{Corr}_f : \mathbb{F}_q^m \to \mathbb{F}_q$ as $\mathrm{Corr}_f(\mathbf{a}) = \mathbf{v_a}$.

Definition 4.5 (Correction for f')**.** *For every pair $(\mathbf{a}, \mathbf{b}) \in \mathbb{F}_q^{2m}$, let $Q_{\mathbf{a},\mathbf{b}} \in \mathcal{P}_c(cm_1, \mathbb{F}_q)$ be the degree-c polynomial whose degree-1 Hadamard encoding is closest to $f'[(\mathbf{a}, \mathbf{b}), \cdot, \cdot]$, i.e.,*

$$Q_{\mathbf{a},\mathbf{b}} \;:=\; \arg\min_{Q\in\mathcal{P}_c} \delta\left(f'[(\mathbf{a},\mathbf{b}),\cdot,\cdot]_{\leqslant 1},\ \mathrm{Had}^{(1)}(Q)\right). \tag{3}$$

Recall the map Φ_{c,m_1} from Equation (1). As the parameters c, m_1 are fixed, we will simply denote Φ_{c,m_1} by Φ. We then define the function F on $\mathbb{F}_q^{2m}$ as $F(\mathbf{a}, \mathbf{b}) = Q_{\mathbf{a},\mathbf{b}} \circ \Phi$, i.e., $F(\mathbf{a}, \mathbf{b})(Y) = Q_{\mathbf{a},\mathbf{b}}(\Phi(Y))$.

By definition of Φ, we know that $F(\mathbf{a}, \mathbf{b})(Y)$ is a univariate polynomial of degree at most

$$c(m_1 - 1)m_1^{c-1} \leqslant cm_1^c \leqslant \frac{cdm_1^c}{(m_1-1)^c} \leqslant 4cd,$$

using that $(m_1 - 1)^c \leqslant d$ and $c \leqslant m_1$. In simple words, F is a collection of univariate polynomials of degree at most $d' = 4cd$.

Let $\mathcal{P}_{d'}$ denote the class $\mathcal{P}_{d'}(m, \mathbb{F}_q)$. Via the triangle inequality, we have,

$$\delta\left(f, \mathrm{Had}^{(r)}(\mathcal{P}_{d'})\right) \;\leqslant\; \delta\left(f,\ \mathrm{Had}^{(r)}(\mathrm{Corr}_f)\right) \;+\; \delta\left(\mathrm{Had}^{(r)}(\mathrm{Corr}_f),\ \mathrm{Had}^{(r)}(\mathcal{P}_{d'})\right) \tag{4}$$

A similar triangle inequality holds for $r = 1$ too. In the above expression, we will bound the first summand in Claim 4.6 and will bound the second summand in Lemma 4.7, which is the key technical lemma in the soundness analysis. We first prove Claim 4.6, which can be seen as a generalization of Lemma 3.10 from points to functions.

Claim 4.6 (Proxy for f)**.** *Assuming $\Pr[\mathcal{E}_1] \leqslant \delta$, we have*

$$\delta\left(f,\ \mathrm{Had}^{(r)}(\mathrm{Corr}_f)\right), \delta\left(f|_{\leqslant 1},\ \mathrm{Had}^{(1)}(\mathrm{Corr}_f)\right) \;\leqslant\; \mathcal{O}(\delta).$$

Lemma 4.7. *Let Corr_f and F be as defined above. Assuming that* NewLDT *rejects with probability at most δ, we have,*

$$\Pr\left[\mathsf{TabLDT}_{d'}^{\mathrm{Corr}_f,F} \text{ returns } \textsc{Reject}\right] \;\leqslant\; \mathcal{O}_c(\delta).$$

Why are the above claim and lemma sufficient? Before discussing the proofs of Claim 4.6 and Lemma 4.7, let us see why they are sufficient to show the soundness guarantee. Let $\beta := \Pr[\mathsf{TabLDT}_{d'}^{\mathrm{Corr}_f,F} \text{ returns } \textsc{Reject}]$. If $\beta < \delta_0$, then Theorem 2.12 gives $\delta(\mathrm{Corr}_f, \mathcal{P}_{d'}) \leqslant 4\beta$, and otherwise trivially $\delta(\mathrm{Corr}_f, \mathcal{P}_{d'}) \leqslant 1 \leqslant \beta/\delta_0$. So in both cases

$$\delta(\mathrm{Corr}_f, \mathcal{P}_{d'}) \;\leqslant\; \max(4, 1/\delta_0) \cdot \Pr[\mathsf{TabLDT}_{d'}^{\mathrm{Corr}_f,F} \text{ returns } \textsc{Reject}].$$

From Observation 3.8, we have,

$$\delta\left(\mathrm{Had}^{(r)}(\mathrm{Corr}_f),\, \mathrm{Had}^{(r)}_{\mathcal{P}_{d'}}\right) \;=\; \frac{1}{2}\delta(\mathrm{Corr}_f, \mathcal{P}_{d'}) \;\leqslant\; \mathcal{O}_c(\delta).$$

Substituting this and Claim 4.6 in Equation (4), we get

$$\delta\left(f,\, \mathrm{Had}^{(r)}_{\mathcal{P}_{d'}}\right) \;\leqslant\; \mathcal{O}(\delta) + \mathcal{O}_c(\delta) = \mathcal{O}_c(\delta).$$

Going through the above calculation also gives us the same upper bound on the relative distance between $f|_{\leqslant 1}$ and $\mathrm{Had}^{(1)}(\mathcal{P}_{d'})$. Therefore, Claim 4.6 and Lemma 4.7 together are sufficient to get the desired soundness guarantee. For the rest of the proof, we discuss their proofs, starting with the proof of Claim 4.6.

Proof of Claim 4.6. Let

$$\mathrm{Rej}_{\mathbf{a}} := \Pr[\mathsf{HadTest}_r^{f[\mathbf{a},\cdot]} = \textsc{Reject}]$$

Then we have

$$\mathbb{E}_{\mathbf{a}}[\mathrm{Rej}_{\mathbf{a}}] = \Pr[\mathcal{E}_1] \leqslant \delta,$$

so by Markov's inequality, we have

$$\Pr_{\mathbf{a}}[\mathrm{Rej}_{\mathbf{a}} > \delta_0] \leqslant \frac{\delta}{\delta_0}.$$

For fixed $\mathbf{a} \in \mathbb{F}_q^m$ with $\mathrm{Rej}_{\mathbf{a}} \leqslant \delta_0$, we have from the soundness of Lemma 3.10, that there exists a $u \in \mathbb{F}_q$ such that

$$\begin{aligned}\delta(f[\mathbf{a},\cdot],\quad \mathrm{Had}^{(r)}(u)) &\leqslant\; \mathrm{Rej}_{\mathbf{a}}\\ \delta(f[\mathbf{a},\cdot]|_{\leqslant 1}, \mathrm{Had}^{(1)}(u)) &\leqslant 3\mathrm{Rej}_{\mathbf{a}}.\end{aligned}$$

Since $\mathrm{Corr}_f(\mathbf{a})$ is defined to be the element of $\mathbb{F}_q$ with closest degree-r Hadamard encoding to $f[\mathbf{a},\cdot]$, we have

$$\delta(f[\mathbf{a},\cdot], \mathrm{Had}^{(r)}(\mathrm{Corr}_f(\mathbf{a}))) \leqslant\; \mathrm{Rej}_{\mathbf{a}}$$

Averaging over every $\mathbf{a} \in \mathbb{F}_q^m$, we then get

$$\delta(f, \mathrm{Had}^{(r)}(\mathrm{Corr}_f)) \leqslant \delta + \frac{\delta}{\delta_0}.$$

giving us the desired bound. Following a similar argument, we get

$$\delta(f|_{\leqslant 1}, \mathrm{Had}^{(1)}(\mathrm{Corr}_f)) \leqslant 3\delta + \frac{\delta}{\delta_0}.$$

This finishes the proof of Claim 4.6. ■

Proof of Lemma 4.7. $\mathsf{TabLDT}_{d'}^{\mathrm{Corr}_f,F}$ returns Reject if $\mathrm{Corr}_f(\mathbf{a}+\lambda\mathbf{b}) \neq F(\mathbf{a},\mathbf{b})(\lambda)$, for the choice of $(\mathbf{a},\mathbf{b},\lambda)$ as random strings. So our goal will be to upper-bound the following quantity:

$$\Pr_{\mathbf{a},\mathbf{b},\lambda}\left[\mathrm{Corr}_f(\mathbf{a}+\lambda\mathbf{b}) \neq F(\mathbf{a},\mathbf{b})(\lambda)\right]$$

Using Observation 3.8, we get,

$$\Pr\left[\mathrm{Corr}_f(\mathbf{a}+\lambda\mathbf{b}) \neq F(\mathbf{a},\mathbf{b})(\lambda)\right] \;=\; 2\cdot\Pr\left[\mathrm{Had}^{(r)}(\mathrm{Corr}_f)(\mathbf{a}+\lambda\mathbf{b},L) \neq \mathrm{Had}^{(r)}(F)(((\mathbf{a},\mathbf{b}),\lambda),L)\right], \tag{5}$$

where the probabilities are over $\mathbf{a},\mathbf{b} \sim \mathbb{F}_q^m$, $\lambda \sim \mathbb{F}_q^\times$, and $L \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}$.
Thus, it suffices to upper-bound the term on the right in the above equality. For ease in writing, we will introduce a shorthand notation.
Notation: For every pair $(\mathbf{a},\mathbf{b}) \in \mathbb{F}_q^{2m}$, for every $\mathbf{u} \in \mathbb{F}_q^{cm_1}$, for every $L \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}$, we define:

$$\mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,L]}(\mathbf{u};\mathbf{v}) := \sum_{i=1}^{c+1} f'[(\mathbf{a},\mathbf{b}),\mathbf{u}+\zeta_i\mathbf{v},\Lambda_{\eta_i,L}].$$

To upper bound the term on the right, we will decompose it into three terms via a union bound:

$$\begin{aligned}
&\Pr\left[\mathrm{Had}^{(r)}(\mathrm{Corr}_f)(\mathbf{a}+\lambda\mathbf{b},L) \neq \mathrm{Had}^{(r)}(F)((\mathbf{a},\mathbf{b}),\lambda,L)\right]\\
\leqslant &\Pr\left[\mathrm{Had}^{(r)}(\mathrm{Corr}_f)(\mathbf{a}+\lambda\mathbf{b},\; L) \neq f[\mathbf{a}+\lambda\mathbf{b},L]\right]\\
+ &\Pr\left[f[\mathbf{a}+\lambda\mathbf{b},L] \neq \mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,L]}(\Phi(\lambda);\mathbf{v})\right]\\
+ &\Pr\left[\mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,L]}(\Phi(\lambda);\mathbf{v}) \neq \mathrm{Had}^{(r)}(F)((\mathbf{a},\mathbf{b})(\lambda),L)\right].
\end{aligned}$$

First and second summands For the first summand, we have

$$\begin{aligned}
&\Pr_{\mathbf{a},\mathbf{b},L}\left[\mathrm{Had}^{(r)}(\mathrm{Corr}_f)(\mathbf{a}+\lambda\mathbf{b},L) \neq f[\mathbf{a}+\lambda\mathbf{b},L]\right]\\
\leqslant\; 2\Pr_{\mathbf{a},\mathbf{b},L'\sim\mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}}&\left[\mathrm{Had}^{(r)}(\mathrm{Corr}_f)(\mathbf{a}+\lambda\mathbf{b},L') \neq f[\mathbf{a}+\lambda\mathbf{b},L']\right]\\
&\leqslant 2\cdot\delta\left(f|_{\leqslant 1}, \mathrm{Had}^{(1)}(\mathrm{Corr}_f)\right) \;\leqslant\; \mathcal{O}(\delta).
\end{aligned}$$

The first inequality is because the density of homogeneous $\mathbb{F}_2$-linear polynomials in degree-1 polynomials is $1/2$, and the final inequality is due to Claim 4.6.

The second summand is exactly $\Pr[\mathcal{E}_4]$, which is at most δ, as per our assumption made at the beginning of

the proof.

Third summand By the definition of F, the third summand is equivalent to

$$\Pr\left[\mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,L]}(\Phi(\lambda);\mathbf{v}) \neq \mathrm{Had}^{(r)}\left(Q_{\mathbf{a},\mathbf{b}}\right)(\Phi(\lambda),\, L)\right]. \tag{6}$$

For a fixed pair $(\mathbf{a},\mathbf{b}) \in \mathbb{F}_q^{2m}$, define $\mathrm{Rej}_{\mathbf{a},\mathbf{b}}$ as the probability that NewLDT rejects when $(\mathbf{a},\mathbf{b})$ is sampled. For fixed $(\mathbf{a},\mathbf{b})$, we then have by Lemma 4.3

$$\Pr\left[\mathrm{Had}^{(r)}\left(Q_{\mathbf{a},\mathbf{b}}\right)(\Phi(\lambda),L) \neq \mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,L]}(\Phi(\lambda);\mathbf{v})\right] \leqslant \mathcal{O}_c(\mathrm{Rej}_{\mathbf{a},\mathbf{b}}).$$

From our assumption that NewLDT rejects with low probability we have

$$\mathbb{E}_{\mathbf{a},\mathbf{b}}[\mathrm{Rej}_{\mathbf{a},\mathbf{b}}] \leqslant \delta,$$

so averaging over $(\mathbf{a},\mathbf{b})$ we get

$$\Pr\left[\mathrm{Had}^{(r)}\left(Q_{\mathbf{a},\mathbf{b}}\right)(\Phi(\lambda),L) \neq \mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,L]}(\Phi(\lambda);\mathbf{v})\right] \leqslant \mathcal{O}_c(\delta).$$

Hence, this gives us an upper bound on the third summand. Putting it together with Equation (5), we get

$$\begin{aligned}
&\Pr_{\mathbf{a},\mathbf{b},\lambda}\left[\mathrm{Corr}_f(\mathbf{a}+\lambda\mathbf{b}) \neq F(\mathbf{a},\mathbf{b})(\lambda)\right] \\
&= 2\Pr\left[\mathrm{Had}^{(r)}\left(\mathrm{Corr}_f\right)(\mathbf{a}+\lambda\mathbf{b},L) \neq \mathrm{Had}^{(r)}\left(F\right)((\mathbf{a},\mathbf{b}),\lambda,L)\right] \leqslant \mathcal{O}_c(\delta).
\end{aligned}$$

This finishes the proof of Lemma 4.7. ■

This finishes the proof of Theorem 4.1, modulo the proof of Lemma 4.3, which we prove in the next subsection. ■

4.3 Proof of Lemma 4.3

Proof of Lemma 4.3. Let $\mathcal{P}_c$ denote the family of degree-c polynomials and $\mathrm{Corr}_g : \mathbb{F}_q^k \to \mathbb{F}_q$ the correction function defined in Definition 4.4. Using the triangle inequality we get

$$\delta\left(g|_{\leqslant 1},\, \mathrm{Had}^{(1)}_{\mathcal{P}_c}\right) \;\leqslant\; \delta\left(g|_{\leqslant 1},\, \mathrm{Had}^{(1)}\left(\mathrm{Corr}_g\right)\right) \;+\; \delta\left(\mathrm{Had}^{(1)}\left(\mathrm{Corr}_g\right),\, \mathrm{Had}^{(1)}_{\mathcal{P}_c}\right).$$

From the first assumption of Lemma 4.3, along with Claim 4.6 we have

$$\delta\left(g|_{\leqslant 1},\, \mathrm{Had}^{(1)}\left(\mathrm{Corr}_g\right)\right) \;\leqslant\; \mathcal{O}(\delta_1). \tag{7}$$

We now focus on proving an upper bound on the distance between $\mathrm{Had}^{(1)}\left(\mathrm{Corr}_g\right)$ and $\mathrm{Had}^{(1)}_{\mathcal{P}_c}$. From Observation 3.8, we have,

$$\delta\left(\mathrm{Had}^{(1)}\left(\mathrm{Corr}_g\right),\, \mathrm{Had}^{(1)}_{\mathcal{P}_c}\right) \;=\; \frac{1}{2}\delta(\mathrm{Corr}_g,\, \mathcal{P}_c).$$

We will show the following claim.

Claim 4.8. *Let* Corr_g *be as defined above. Then,*

$$\delta(\mathrm{Corr}_g, \mathcal{P}_c) \leqslant \mathcal{O}_c(\delta_1).$$

Proof of Claim 4.8. To upper bound the distance between Corr_g and $\mathcal{P}_c$, we will use StdLDT and Lemma 3.3. In particular, we will upper bound the following probability:

$$\Pr_{\mathbf{u},\mathbf{v}}\left[\mathsf{StdLDT}_c^{\mathrm{Corr}_g}(;\mathbf{u},\mathbf{v}) \neq 0\right] = \Pr_{\mathbf{u},\mathbf{v}}\left[\sum_{i=0}^{c+1}\eta_i \mathrm{Corr}_g(\mathbf{u}+\zeta_i\mathbf{v}) \neq 0\right]. \tag{8}$$

Fix any pair $\mathbf{u}, \mathbf{v}$ for which we have:

$$\sum_{i=0}^{c+1}\eta_i \mathrm{Corr}_g(\mathbf{u}+\zeta_i\mathbf{v}) \neq 0.$$

Then for $L \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}$, we get,

$$\Pr_{L}\left[\sum_{i=0}^{c+1}\mathrm{Had}^{(r)}\left(\mathrm{Corr}_g\right)\left((\mathbf{u}+\zeta_i\mathbf{v}),\Lambda_{\eta_i,L}\right) \neq 0\right] = \Pr_{L}\left[L\left(\rho\left(\sum_{i=0}^{c+1}\eta_i \mathrm{Corr}_g(\mathbf{u}+\zeta_i\mathbf{v})\right)\right) \neq 0\right] = \frac{1}{2},$$

where for the final equality we use Observation 3.8. Thus,

$$\Pr_{\mathbf{u},\mathbf{v}}\left[\sum_{i=0}^{c+1}\eta_i \mathrm{Corr}_g(\mathbf{u}+\zeta_i\mathbf{v}) \neq 0\right] = 2\cdot\Pr_{\mathbf{u},\mathbf{v},L}\left[\sum_{i=0}^{c+1}\mathrm{Had}^{(r)}\left(\mathrm{Corr}_g\right)\left((\mathbf{u}+\zeta_i\mathbf{v}),\Lambda_{\eta_i,L}\right) \neq 0\right].$$

So our goal now reduces to getting an upper bound on the probability of the right-hand event.

For random $\mathbf{u}, \mathbf{v}$, the points $\mathbf{u}+\zeta_i\mathbf{v}$ are (non-independent) uniformly random, so by a union bound using Equation (7), along with the fact that homogeneous linear polynomials have density $1/2$ among linear polynomials, and $\Lambda_{\eta_i,L}$ is a random homogeneous linear polynomial, we get

$$\Pr_{\mathbf{u},\mathbf{v},L}\left[\sum_{i=0}^{c+1}\mathrm{Had}^{(r)}\left(\mathrm{Corr}_g\right)\left((\mathbf{u}+\zeta_i\mathbf{v}),\Lambda_{\eta_i,L}\right) \neq \sum_{i=0}^{c+1} g[(\mathbf{u}+\zeta_i\mathbf{v}),\Lambda_{\eta_i,L}]\right] \leqslant 2(c+2)\mathcal{O}(\delta_1).$$

From the second assumption of Lemma 4.3, we have

$$\Pr_{\mathbf{u},\mathbf{v},L}\left[\sum_{i=0}^{c+1} g[(\mathbf{u}+\zeta_i\mathbf{v}),\Lambda_{\eta_i,L}] \neq 0\right] \leqslant \delta_1.$$

Combining the last two inequalities, we get

$$\Pr_{\mathbf{u},\mathbf{v}}\left[\sum_{i=0}^{c+1}\eta_i \mathrm{Corr}_g(\mathbf{u}+\zeta_i\mathbf{v}) \neq 0\right] = 2\Pr_{\mathbf{u},\mathbf{v},L}\left[\sum_{i=0}^{c+1}\mathrm{Had}^{(r)}\left(\mathrm{Corr}_g\right)\left((\mathbf{u}+\zeta_i\mathbf{v}),\Lambda_{\eta_i,L}\right) \neq 0\right] \leqslant \alpha(c)\cdot\delta_1,$$

where $\alpha(c)$ is some constant depending on c. We will now apply Lemma 3.3, which holds when $\alpha(c) \cdot \delta_1$ is less than $1/2(c+2)^2$. However if $\alpha(c) \cdot \delta_1 \geqslant 1/2(c+2)^2$, then we have

$$\delta(\mathrm{Corr}_g, \mathcal{P}_c) \leqslant 1 \leqslant 2(c+2)^2\alpha(c) \cdot \delta_1,$$

So independent of the value of $\alpha(c) \cdot \delta_1$ we get

$$\delta(\mathrm{Corr}_g, \mathcal{P}_c) \;\leqslant\; 2(c+2)^2\alpha(c) \cdot \delta_1 = \mathcal{O}_c(\delta_1)$$

finishing the proof of Claim 4.8. ■

Combining Equation (7) and Claim 4.8, we get,

$$\delta\left(g|_{\leqslant 1}, \mathrm{Had}^{(1)}_{\mathcal{P}_c}\right) \;\leqslant\; \mathcal{O}(\delta_1) + \mathcal{O}_c(\delta_1) \;\leqslant \mathcal{O}_c(\delta_1).$$

Now let P be a degree-c polynomial such that $\delta\left(g|_{\leqslant 1}, \mathrm{Had}^{(1)}(P)\right) \;\leqslant\; \mathcal{O}_c(\delta_1)$. Since P is a degree-c polynomial, we have from Fact 2.8

$$\sum_{i=1}^{c+1} \mathrm{Had}^{(1)}(P)(\mathbf{u} + \zeta_i\mathbf{v}, \Lambda_{\eta_i,L}) \;=\; \mathrm{Had}^{(1)}\left(\sum_{i=1}^{c+1} \eta_i P(\mathbf{u} + \zeta_i\mathbf{v})\right)(L) \;=\; \mathrm{Had}^{(1)}(P)(\mathbf{u}, L).$$

The above relation says that if for every $1 \leqslant i \leqslant c+1$, the queried value $g[\mathbf{u} + \zeta_i\mathbf{v}, \Lambda_{\eta_i,L}]$ equals $\mathrm{Had}^{(1)}(P)(\mathbf{u} + \zeta_i\mathbf{v}, \Lambda_{\eta_i,L})$, then the sum returns $\mathrm{Had}^{(1)}(P)(\mathbf{u}, L)$.

For every $1 \leqslant i \leqslant (c+1)$, the point $\mathbf{u} + \zeta_i\mathbf{v}$, is uniformly distributed over $\mathbb{F}_q^k$, since $\mathbf{v}$ is uniformly distributed over $\mathbb{F}_q^k$ and $\zeta_i \neq 0$. Since $L \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}$, each $\Lambda_{\eta_i,L}$ is also uniformly random, so by a union bound we get

$$\Pr_{\mathbf{v},L}\left[\sum_{i=1}^{c+1} g[\mathbf{u} + \zeta_i\mathbf{v}, \Lambda_{\eta_i,L}] \;\neq\; \mathrm{Had}^{(r)}(P)(\mathbf{u}, L)\right] \;\leqslant\; (c+1) \cdot \Pr_{\mathbf{v},L}\left[g[\mathbf{v}, L] \neq \mathrm{Had}^{(r)}(P)(\mathbf{v}, L)\right].$$

Since homogeneous linear polynomials $\mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}$ have density $1/2$ in the space of degree-1 polynomials $\mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$, for every $\mathbf{u} \in \mathbb{F}_q^k$ we get

$$\Pr_{\mathbf{v},L}\left[\sum_{i=1}^{c+1} g[\mathbf{u} + \zeta_i\mathbf{v}, \Lambda_{\eta_i,L}] \;\neq\; \mathrm{Had}^{(r)}(P)(\mathbf{u}, L)\right] \;\leqslant\; 2(c+1) \cdot \delta\left(g|_{\leqslant 1}, \mathrm{Had}^{(1)}(P)\right) = \mathcal{O}_c(\delta_1)$$

which finishes the proof. ■

5 Local Correction over Binary Alphabet

In this section, we discuss our local correction algorithm for degree-1 Hadamard encoding of a low-degree $\mathbb{F}_q$-polynomial, at a random linear polynomial in $\mathbb{F}_2[\mathbf{Z}]$. We start by developing some intuition for our test. Then we give a formal description in Algorithm 6, state its guarantee in Theorem 5.1, and prove it in Appendix C. More particularly, we are interested in the following task:

Suppose we have oracle access to $H : \mathbb{F}_q^m \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$ that is close to $\mathrm{Had}^{(1)}(P)$ for a degree-d polynomial. For a given $\mathbf{a} \in \mathbb{F}_q^m$ and a random linear polynomial L, we want to output $\mathrm{Had}^{(1)}(P)(L)$ with high probability.

We show that using degree-1 Hadamard encoding of $\Psi^*(P_{\text{lines}})$, we get a local correction algorithm that has constant query complexity over the alphabet $\mathbb{F}_2$. The algorithm is completely described in Algorithm 6, its guarantee is stated in Theorem 5.1, and it is proved in Appendix C. Note that we are fixing $\mathbf{a} \in \mathbb{F}_q^m$ but allow the second coordinate to be random in $\mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\text{hom}}$. This might seem slightly unconventional for a local correction task. We start by explaining why this local correction suffices for us and give an intuition behind Algorithm 6, before describing the algorithm.

Why does the above local correction suffice? In our PCP construction, we will have a situation similar to the following: Let P_1 and P_2 be degree-d polynomials on $\mathbb{F}_q^m$ and we have oracles Π_1 and Π_2 that are close to $\mathrm{Had}^{(1)}(P_1)$ and $\mathrm{Had}^{(1)}(P_2)$ respectively. We would like to check whether $P_1(\mathbf{a})$ equals $P_2(\mathbf{a})$ for a fixed $\mathbf{a}$, via querying the oracles Π_1 and Π_2. Similar to Observation 3.8, if $P_1(\mathbf{a}) \neq P_2(\mathbf{a})$, then $\mathrm{Had}^{(1)}(P_1)(\mathbf{a}, L)$ is not equal to $\mathrm{Had}^{(1)}(P_2)(\mathbf{a}, L)$ for a constant fraction of $L \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\text{hom}}$. In short, we are actually interested only in local correction of the underlying low-degree polynomial, and that's why the second coordinate L can be random. Also, the reason we want L to be homogeneous and not arbitrary degree-1 is the following (it is similar to the reason we used homogeneous linear polynomials in NewLDT, in Algorithm 5): We would like to perform analogues of StdLDT and StdLC, and $\mathrm{Had}^{(1)}(u+v)(Q)$ is not necessarily equal to $\mathrm{Had}^{(1)}(u)(Q) + \mathrm{Had}^{(1)}(v)(Q)$ for arbitrary $Q \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$, but it does hold for $Q \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\text{hom}}$.

Our local correction algorithm over $\mathbb{F}_2$ has a close resemblance to our constant-query low-degree test over $\mathbb{F}_2$ (see Algorithm 5). Recall the constant-query algorithm TabLC over alphabet $\mathbb{F}_q^{d+1}$ (see Theorem 2.14). The idea is to simulate TabLC using the two encodings, Ψ (see Definition 3.1) and $\mathrm{Had}^{(1)}(\cdot)$ (see Definition 3.6). Similar to NewLDT (Algorithm 5), we ask the prover to:

- Firstly, for every line in $\mathbb{F}_q^m$, encode its corresponding degree-d univariate polynomial in the lines table using Ψ_{d,c,m_1} for constant c.
- Secondly, now encode all the $\mathbb{F}_q$ functions using degree-1 Hadamard encoding.

We also want to emphasize that finally in our PCP construction, the local corrector will be applied after the low-degree test NewLDT. So the algorithm is designed while keeping in mind that "the oracles are close to the expected oracles", i.e., if we expected the oracle to be degree-1 Hadamard encoding on a degree-c polynomial, then it's close to such a function. So it will be helpful to recall the guarantees provided by Theorem 4.1 before going to the local corrector. We start by stating the expected oracles from an honest prover.

Expected Oracles and Queries Suppose $P \in \mathcal{P}_d(m, \mathbb{F}_q)$ is a degree-d polynomial and let $\Psi^*(P_{\text{lines}})$ be function defined in Section 4.2. The expected oracles are:

$$\mathrm{Had}^{(1)}(P) : \mathbb{F}_q^m \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$$
$$\mathrm{Had}^{(1)}(\Psi^*(P_{\text{lines}})) : \mathbb{F}_q^{2m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2.$$

Suppose the verifier has access to oracles $f : \mathbb{F}_q^m \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$ and $f' : \mathbb{F}_q^{2m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$. Following TabLC, the verifier wants to check if $f'[(\mathbf{a},\mathbf{b}), \Phi(\lambda), \cdot]$ equals $f[\mathbf{a} + \lambda\mathbf{b}, \cdot]$. One would like to do something similar to Line 4 of NewLDT. However, for this we need the guarantee that $f'[(\mathbf{a},\mathbf{b}), \cdot, \cdot]$ is Hadamard encoding of a degree-c polynomial in $\mathbf{X}$ variables. From NewLDT, we only have this guarantee for most of the lines (i.e., pairs $(\mathbf{a},\mathbf{b})$), but here $\mathbf{a}$ is fixed, and therefore the guarantee from NewLDT is not sufficient. Thus we first do a HadTest and then an analogue of StdLDT (similar to Line 3 in NewLDT) on the function $f'[(\mathbf{a},\mathbf{b}), \cdot, \cdot]$. After this, the verifier can return the corrected value of $f'[(\mathbf{a},\mathbf{b}), \Phi(0), L]$. This is similar to Line 4 in NewLDT, except there is no comparison against f. We now describe our algorithm.

Algorithm 6: Local Correction over $\mathbb{F}_2$ - NewLC

Input: Parameters d, c, Evaluation point $\mathbf{a} \in \mathbb{F}_q^m$, Oracles $f : \mathbb{F}_q^m \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$, $f' : \mathbb{F}_q^{2m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$, and $L \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\text{hom}}$

1 Sample $\mathbf{b} \sim \mathbb{F}_q^m$, $\mathbf{u}, \mathbf{v} \sim \mathbb{F}_q^{cm_1}$, $\lambda \in \mathbb{F}_q^\times$

2 Run $\text{HadTest}_1^{f'[(\mathbf{a},\mathbf{b}),\mathbf{u},\cdot]}$

3 Check if $f'[(\mathbf{a},\mathbf{b}), \mathbf{u}, \Lambda_{\eta_0,L}] + \sum_{i=1}^{c+1} f'[(\mathbf{a},\mathbf{b}), \mathbf{u} + \zeta^i\mathbf{v}, \Lambda_{\eta_i,L}] \stackrel{?}{=} 0$

4 Check if $\sum_{i=1}^{c+1} f'[(\mathbf{a},\mathbf{b}), \Phi(\lambda) + \zeta^i\mathbf{v}, \Lambda_{\eta_i,L}] \stackrel{?}{=} f[\mathbf{a} + \lambda\mathbf{b}, L]$

5 **if** *any of the above tests returns* REJECT **then return** REJECT

6 **else return** $\sum_{i=1}^{c+1} f'[(\mathbf{a},\mathbf{b}), \Phi(0) + \zeta^i\mathbf{v}, \Lambda_{\eta_i,L}]$

Theorem 5.1 (Local Correction over $\mathbb{F}_2$)**.** *There exists an absolute constant $C > 0$ such that for every set of parameters $m_1, c, d, d', q \in \mathbb{N}$ satisfying $m_1 \geqslant c \geqslant 2$, $d < m_1^c \leqslant d'/c$, q is a power of 2 and $q > Cd'^3$, the following holds:*

1. ***Completeness:*** *If there exists $P \in \mathcal{P}_d(m, \mathbb{F}_q)$ such that $f = \text{Had}^{(1)}(P)$ and $f' = \text{Had}^{(1)}(P')$, where $P' = \Psi^*_{d,c,m_1}(P_{\text{lines}})$, then for every $\mathbf{a}$ and L, $\text{NewLC}^{f,f'}_{d,c}(\mathbf{a}, L; \mathbf{b}, \mathbf{u}, \mathbf{v}, \lambda)$ returns $\text{Had}^{(1)}(P)(\mathbf{a}, L)$ with probability 1.*

2. ***Soundness:*** *Assume there exists a polynomial $P \in \mathcal{P}_{d'}(m, \mathbb{F}_q)$ such that f is δ-close to $\text{Had}^{(1)}(P)$. Then for every f' and $\mathbf{a} \in \mathbb{F}_q^m$, the following holds: If $\text{NewLC}^{f,f'}_{d,c}(\mathbf{a}, \cdot)$ rejects with probability at most η, then we have*

$$\Pr_{L,\mathbf{b},\mathbf{u},\mathbf{v},\lambda}[\text{NewLC}^{f,f'}_{d,c}(\mathbf{a}, L; \mathbf{b}, \mathbf{u}, \mathbf{v}, \lambda) \text{ returns } \text{Had}^{(1)}(P)(\mathbf{a}, L)] \geqslant 1 - \mathcal{O}_c\left(d'/q + \eta + \delta\right),$$

where in the above probability, $L \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\text{hom}}$.

Furthermore, $\text{NewLC}^{f,f'}_{d,c}(\mathbf{a}, L;)$ makes $\mathcal{O}(c)$ queries to the oracle, uses $\mathcal{O}(m \log q + cm_1 \log q)$ bits of randomness, and runs in time $\text{poly}(m_1^c, \log q)$.

We prove Theorem 5.1 in Appendix C.

6 Zero-on-Grid Test over Binary Alphabet

In this section, we will describe a test to decide whether a low-degree polynomial vanishes on a subset of its domain via its Hadamard encoding. We start by developing some intuition for our test. Then we give a formal description in Algorithm 7, state its guarantee in Theorem 6.1, and prove it in Appendix D. We are interested in the following task:

Fix a subset $H^m \subset \mathbb{F}_q^m$. Given oracle access to the Hadamard encoding of a low-degree m-variate polynomial P, decide whether P vanishes on H^m.

In our PCP construction, H will be of size $\mathcal{O}(\log n)$ and P will be of degree $\mathcal{O}(m|H|)$. Let us first understand a test for a simpler task. Assume we have oracle access to P and we want a test over the alphabet $\mathbb{F}_q^{d+1}$. This exact task was done in [ABSSW26a], using the vanishing certificate polynomial (see Corollary 2.10). Recall that P vanishes on H^m if and only if the vanishing certificate polynomial exists. The test is as follows. The verifier gets oracle access to P and to the vanishing certificate $\mathcal{M}_P$ (for simplicity, assume for now that both the polynomials are low-degree polynomials of the correct degree). The verifier samples $\mathbf{a} \sim \mathbb{F}_q^m$ and checks:

- Is $\mathcal{M}_P(\mathbf{a}, \mathbf{0})$ equal to 0?
- Is $\mathcal{M}_P(\mathbf{a}, Z_H(\mathbf{a}))$ equal to $P(\mathbf{a})$?

The above test has good soundness again because of the polynomial distance lemma over $\mathbb{F}_q$ (Theorem 2.5). In the construction of PCP in [ABSSW26a], they only had the guarantee that the oracle, which is supposed to be $\mathcal{M}_P$, is in fact only close to a low-degree polynomial. Since the two queried points $(\mathbf{a}, \mathbf{0})$ and $(\mathbf{a}, Z_H(\mathbf{a}))$ are not uniformly random, they cannot directly apply the polynomial distance lemma. Thus [ABSSW26a] used local correction on the oracle at these points. Note that $\mathcal{M}_P$ is a degree-d polynomial, which means the verifier does the local correction using the lines table, TabLC (see Algorithm 2).
Now we come back to our setting, where we only have access to Hadamard encodings and not the low-degree polynomials themselves. In this case, the verifier samples $\mathbf{a} \sim \mathbb{F}_q^m$, $L \sim \mathbb{F}_2[\mathbf{Z}]^{\leq 1}_{\text{hom}}$ and checks:

- Is $\text{Had}^{(1)}(\mathcal{M}_P)((\mathbf{a}, \mathbf{0}), L)$ equal to 0?
- Is $\text{Had}^{(1)}(\mathcal{M}_P)((\mathbf{a}, Z_H(\mathbf{a})), L)$ equal to $\text{Had}^{(1)}(P)(\mathbf{a}, L)$?

Finally, the verifier needs to do a local correction at these points. In the above discussion, the verifier uses TabLC. To make it work over $\mathbb{F}_2$, the verifier instead uses NewLC (see Algorithm 6). We now give the description of the test.

Algorithm 7: Zero-on-Grid Test over $\mathbb{F}_2$: ZERO

Input: Subset $H \subseteq \mathbb{F}_q$, Parameters d, c, Oracle access to $f : \mathbb{F}_q^m \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$, $\Pi : \mathbb{F}_q^{2m} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$, and $\Pi' : \mathbb{F}_q^{4m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$

1 Sample $\mathbf{a} \sim \mathbb{F}_q^m$, $\boldsymbol{\beta} \sim \mathbb{F}_q^{2m}$, $\mathbf{u}, \mathbf{v} \sim \mathbb{F}_q^{cm_1}$, $\lambda \sim \mathbb{F}_q^\times$, and $L \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\text{hom}}$

2 Check if $\text{NewLC}_{d,c}^{\Pi,\Pi'}((\mathbf{a}, \mathbf{0}), L; \boldsymbol{\beta}, \mathbf{u}, \mathbf{v}, \lambda) \stackrel{?}{=} 0$

3 Check if $\text{NewLC}_{d,c}^{\Pi,\Pi'}((\mathbf{a}, Z_H(\mathbf{a})), L; \boldsymbol{\beta}, \mathbf{u}, \mathbf{v}, \lambda) \stackrel{?}{=} f[\mathbf{a}, L]$

4 **if** *either of the above tests returns* REJECT **then return** REJECT

5 **else return** ACCEPT

Theorem 6.1 (Zero-on-Grid Test). *There exists an absolute constant $C > 0$ such that for every set of parameters $m, m_1, c, d, d', q \in \mathbb{N}$ satisfying $m_1 \geqslant c \geqslant 2$, $d < m_1^c \leqslant d'/c$, q is a power of 2 with $q > Cd'^3$, and $H \subseteq \mathbb{F}_q$, the following holds:*

1. ***Completeness:*** *Suppose $f = \mathrm{Had}^{(1)}(P)$ for a polynomial $P \in \mathcal{P}_d(m, \mathbb{F}_q)$ satisfying $P|_{H^m} \equiv 0$. Let $M \in \mathcal{P}_d(2m, \mathbb{F}_q)$ be the corresponding vanishing-certificate polynomial from Corollary 2.10. If $\Pi = \mathrm{Had}^{(1)}(M)$ and $\Pi' = \mathrm{Had}^{(1)}(M')$ where $M' = \Psi^*_{d,c,m_1}(M_{\text{lines}})$, then $\text{ZERO}_H^{f,\Pi,\Pi'}$ returns ACCEPT with probability 1.*

2. ***Soundness:*** *Assume there exists a polynomial $P \in \mathcal{P}_{d'}(m, \mathbb{F}_q)$ such that f is δ-close to $\mathrm{Had}^{(1)}(P)$ and there exists a polynomial $M \in \mathcal{P}_{d'}(2m, \mathbb{F}_q)$ such that Π is δ-close to $\mathrm{Had}^{(1)}(M)$.*
   If $P|_{H^m} \not\equiv 0$, then for all Π', $\text{ZERO}_H^{f,\Pi,\Pi'}$ returns REJECT with probability at least

$$\eta := \Omega_c(1) - \frac{2d'|H|}{q} - \delta.$$

   Furthermore, $\text{ZERO}_H^{f,\Pi,\Pi'}$ makes $\mathcal{O}(c)$ oracle queries, uses $\mathcal{O}(m \log q + cm_1 \log q)$ random bits, and runs in time $\mathrm{poly}(m_1^c, \log q)$.

We prove Theorem 6.1 in Appendix D.

7 The PCP Verifier

In this section, we give the description of a PCP verifier $\mathcal{V}$ and the expected proof for the language 3-COLOR.

Setup Throughout this section, we will use n to denote the number of vertices in an input graph $G = (V, E)$. We will identify (after adding a suitable number of isolated vertices) the vertices of the input graph V with H^m for suitable $m \in \mathbb{N}$ and a subset $H \subseteq \mathbb{F}_q$ (for an appropriate choice of the field size $q = 2^t$ and the set size $h = |H|$). Let $\rho : \mathbb{F}_q \to \mathbb{F}_2^t$ be an $\mathbb{F}_2$-linear bijection and $c \geqslant 2$ be a constant. Let $\omega, \zeta \in \mathbb{F}_q^\times$ where ω is of order 3 and ζ is a generator respectively (assume that $q - 1$ is divisible by 3). Let

$D, m_1 \in \mathbb{N}$ be given by $D = c_2 hm$ (for some large enough constant $c_2 \in \mathbb{N}$) and $(m_1 - 1)^c \leqslant D < m_1^c$. Let $\Psi = \Psi_{D,c,m_1}$ and $\Psi^* = \Psi^*_{D,c,m_1}$ be from Definition 3.1, $\Phi = \Phi_{c,m_1}$ be from Equation (1), and $\mathrm{Had}^{(r)}(\cdot)$ be from Definition 3.6. Recall that $\mathcal{P}_d(m, \mathbb{F}_q)$ denotes the family of m-variate degree-d polynomials over $\mathbb{F}_q$.

Expected Proof Assume $G = (V, E) \in$ 3-COLOR and let $\mathsf{Color} : H^m \to \{1, \omega, \omega^2\}$ be a proper 3-coloring of G. Before describing the PCP verifier, we will first describe the corresponding proof oracles. For this, we use the shorthand of $\widehat{f} = \mathrm{LDE}(f)$ to denote the *low-degree extension* of a function $f : H^{m'} \to \mathbb{F}_q$ (see Fact 2.4). There are $8 = 4 \times 2$ proof oracles:

1. Let $\chi := \widehat{\mathsf{Color}} \in \mathcal{P}_D(m, \mathbb{F}_q)$ and χ_{lines} denote the lines table of χ.

   The proof consists of $\mathrm{Had}^{(3)}(\chi) : \mathbb{F}_q^m \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3} \to \mathbb{F}_2$ and
   $\mathrm{Had}^{(3)}(\Psi^*(\chi_{\mathrm{lines}})) : \mathbb{F}_q^{2m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3} \to \mathbb{F}_2$.

2. Let $\mathsf{Val} \in \mathcal{P}_D(m, \mathbb{F}_q)$ denote the *valid-coloring polynomial*, defined as:

$$\mathsf{Val}(\mathbf{a}) := \prod_{i \in \{1,\omega,\omega^2\}} (\chi(\mathbf{a}) - i) = \chi(\mathbf{a})^3 - 1.$$

   Observation 7.1. *Suppose $\chi : V \to \{1, \omega, \omega^2\}$ is an extension of a (not necessarily proper) 3-coloring of G. Then for every $\mathbf{a} \in V$,*

$$\mathsf{Val}(\mathbf{a}) = 0.$$

   In other words, the polynomial Val *is identically zero on the subset $H^m \subseteq \mathbb{F}_q^m$. It is easy to see that the converse also holds, i.e., if* Val *is identically zero on the subset H^m, then χ gives a valid coloring of V.*

   Since the polynomial Val vanishes on H^m, Corollary 2.10 tells us there exists a polynomial $\mathcal{M}_{\mathsf{Val}} \in \mathbb{F}_q[X_1, \ldots, X_m, Y_1, \ldots, Y_m]$ satisfying the properties listed in Corollary 2.10. Let $\mathcal{M}_{\mathsf{Val},\mathrm{lines}}$ denote the lines table of $\mathcal{M}_{\mathsf{Val}}$.

   The proof consists of $\mathrm{Had}^{(1)}(\mathcal{M}_{\mathsf{Val}}) : \mathbb{F}_q^{2m} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$ and
   $\mathrm{Had}^{(1)}(\Psi^*(\mathcal{M}_{\mathsf{Val},\mathrm{lines}})) : \mathbb{F}_q^{4m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$.

3. Let $\chi' \in \mathcal{P}_D(2m, \mathbb{F}_q)$ denote the polynomial defined as

$$\chi'(\mathbf{a}, \mathbf{b}) = \chi(\mathbf{a}) - \chi(\mathbf{b}).$$

   The proof consists of $\mathrm{Had}^{(3)}(\chi') : \mathbb{F}_q^{2m} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3} \to \mathbb{F}_2$ and
   $\mathrm{Had}^{(3)}(\Psi^*(\chi'_{\mathrm{lines}})) : \mathbb{F}_q^{4m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3} \to \mathbb{F}_2$.

4. Let $\mathsf{Prop} \in \mathcal{P}_D(2m, \mathbb{F}_q)$ denote the *proper-coloring polynomial*, defined as:

$$\mathsf{Prop}(\mathbf{a}, \mathbf{b}) := \widehat{E}(\mathbf{a}, \mathbf{b}) \cdot \prod_{i \in \{1,\omega,\omega^2\}} (\chi'(\mathbf{a}, \mathbf{b}) - i) = \widehat{E}(\mathbf{a}, \mathbf{b}) \cdot (\chi'(\mathbf{a}, \mathbf{b})^3 - 1).$$

Observation 7.2. *Suppose $\chi : V \to \{1, \omega, \omega^2\}$ is an extension of a proper 3-coloring of G. Then for every $\mathbf{a}, \mathbf{b} \in V$,*

$$\mathsf{Prop}(\mathbf{a}, \mathbf{b}) \;=\; 0.$$

In other words, the polynomial Prop *is identically zero on H^{2m}. It is easy to see that the converse is also true, i.e., if* Prop *is identically zero on H^{2m}, then χ gives a proper 3-coloring of G.*

Since Prop vanishes on H^{2m}, Corollary 2.10 tells us there exists a polynomial $\mathcal{M}_{\mathsf{Prop}} \in \mathbb{F}_q[X_1, \ldots, X_{2m}, Y_1, \ldots, Y_{2m}]$ satisfying the properties listed in Corollary 2.10. Let $\mathcal{M}_{\mathsf{Prop},\mathrm{lines}}$ denote the lines table of $\mathcal{M}_{\mathsf{Prop}}$.
The proof consists of $\mathrm{Had}^{(1)}\left(\mathcal{M}_{\mathsf{Prop}}\right) : \mathbb{F}_q^{4m} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$ and
$\mathrm{Had}^{(1)}\left(\Psi^*(\mathcal{M}_{\mathsf{Prop},\mathrm{lines}})\right) : \mathbb{F}_q^{8m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$.

Expected Queries We now explain the queries our PCP verifier $\mathcal{V}$ is going to make while expecting the proof oracles as mentioned above. Firstly, $\mathcal{V}$ runs the low-degree test NewLDT on all the oracles. Guided by Observation 7.1, $\mathcal{V}$ does the following:

- Gets oracle access to a function that is supposed to be $\mathrm{Had}^{(1)}\left(\mathsf{Val}\right)$, using the oracle that is supposed to be $\mathrm{Had}^{(3)}\left(\chi\right)$. Here, by "gets oracle access" we mean that the function can be queried anywhere using constant many oracle queries to the oracle that is supposed to be $\mathrm{Had}^{(3)}\left(\chi\right)$ (we require constant many oracle queries so the final query complexity is a constant).
- Runs the zero-on-grid test ZERO on this new oracle.

In fact, the first step is slightly more subtle, which we elaborate on now. From Observation 7.1, we see that $\mathrm{Had}^{(1)}\left(\mathsf{Val}\right)$ is essentially a degree-3 version of $\mathrm{Had}^{(3)}\left(\chi\right)$, i.e. for every $\mathbf{a} \in \mathbb{F}_q^m$ and $L' \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$, we have, $\mathrm{Had}^{(1)}\left(\mathsf{Val}\right)(\mathbf{a}, L')$ is $\mathrm{Had}^{(3)}\left(\chi\right)(\mathbf{a}, \Lambda_{L'})$ where $\Lambda_{L'} \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3}$ is given by Claim 3.7. Observe that even for uniformly random $L' \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$, the distribution of $\Lambda_{L'}$ is far from uniform distribution[13] on $\mathbb{F}_2[\mathbf{Z}]^{\leqslant 3}$. From NewLDT, we only have the guarantee that we have an oracle that is close to $\mathrm{Had}^{(3)}\left(\chi\right)$, and thus we have to do self-correction of degree-3 Hadamard encoding.
Again, guided by Observation 7.2, $\mathcal{V}$ gets oracle access to a function that is supposed to be $\mathrm{Had}^{(1)}\left(\mathsf{Prop}\right)$, using the oracle that is supposed to be $\mathrm{Had}^{(3)}\left(\chi'\right)$, and then runs ZERO on it. Also, for the two ZERO tests to give a consistent result, $\mathcal{V}$ has to check that the oracles that are supposed to be $\mathrm{Had}^{(3)}\left(\chi\right)$ and $\mathrm{Had}^{(3)}\left(\chi'\right)$ are consistent.

Now we are ready to fully describe our PCP verifier $\mathcal{V}$ (Algorithm 8). As we saw above, the expected oracles come in pairs (Π_f, Π'_f), where Π is a degree-r Hadamard encoding of a function f and Π' is a degree-r Hadamard encoding of $\Psi^*(f_{\mathrm{lines}})$. We will use $(\widetilde{\Pi}_f, \widetilde{\Pi'}_f)$ to denote the corresponding received oracles. Finally, let Λ be as defined in Claim 3.7. All randomness used in Algorithm 8 is uniform and independent and $\mathbf{Z} = (Z_1, \ldots, Z_t)$.

[13] An informal way to believe this is by noting that the number of $\Lambda_{L'}$'s is at most the number of degree-1 polynomials in $\mathbb{F}_2[\mathbf{Z}]$, which is quite small compared to the number of polynomials in $\mathbb{F}_2[\mathbf{Z}]^{\leqslant 3}$.

Algorithm 8: PCP Verifier $\mathcal{V}$

Parameters: $n, h, m, H, \mathbb{F}_q, t, \rho, c, \omega, \zeta, D, m_1, c_1, c_2$ as discussed above.
Input: Graph $G = (V, E)$ where $V = H^m$.
Oracles:

$\widetilde{\Pi}_\chi : \mathbb{F}_q^m \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3} \to \mathbb{F}_2$,
$\widetilde{\Pi}_{\mathcal{M}_{\mathsf{Val}}} : \mathbb{F}_q^{2m} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$,
$\widetilde{\Pi}_{\chi'} : \mathbb{F}_q^{2m} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3} \to \mathbb{F}_2$,
$\widetilde{\Pi}_{\mathcal{M}_{\mathsf{Prop}}} : \mathbb{F}_q^{4m} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$,

$\widetilde{\Pi'}_\chi : \mathbb{F}_q^{2m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3} \to \mathbb{F}_2$,
$\widetilde{\Pi'}_{\mathcal{M}_{\mathsf{Val}}} : \mathbb{F}_q^{4m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$,
$\widetilde{\Pi'}_{\chi'} : \mathbb{F}_q^{4m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3} \to \mathbb{F}_2$,
$\widetilde{\Pi'}_{\mathcal{M}_{\mathsf{Prop}}} : \mathbb{F}_q^{8m} \times \mathbb{F}_q^{cm_1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \to \mathbb{F}_2$.

1 Sample $\mathbf{a}, \mathbf{b} \sim \mathbb{F}_q^m$, $R \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3}$, and $L \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\text{hom}}$
2 Given $\mathbf{a}', \mathbf{b}' \in \mathbb{F}_q^m$ and $L' \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$, let $\Lambda_{L'} \leftarrow \Lambda_{Y^3-1,L'}$ and $\Lambda_{\mathbf{a}',\mathbf{b}',L'} \leftarrow \Lambda_{\mathrm{LDE}(E)(\mathbf{a}',\mathbf{b}')\cdot(Y^3-1),L'}$ as defined in Claim 3.7
3 **Subroutine** $\mathsf{SC}(g : \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3} \to \mathbb{F}_2,\ \Lambda \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3})$ **return** $g[\Lambda + R] - g[R]$
4 Given $\mathbf{a}' \in \mathbb{F}_q^m$ and $L' \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$, let $\widetilde{\Pi}_{\mathsf{Val}}[\mathbf{a}', L'] \leftarrow \mathsf{SC}(\widetilde{\Pi}_\chi[\mathbf{a}', \cdot], \Lambda_{L'})$
5 Given $\mathbf{a}', \mathbf{b}' \in \mathbb{F}_q^m$ and $L' \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$, let $\widetilde{\Pi}_{\mathsf{Prop}}[(\mathbf{a}', \mathbf{b}'), L'] \leftarrow \mathsf{SC}(\widetilde{\Pi}_{\chi'}[(\mathbf{a}', \mathbf{b}'), \cdot], \Lambda_{\mathbf{a}',\mathbf{b}',L'})$
6 **for** *every pair of oracles* $(\widetilde{\Pi}, \widetilde{\Pi'})$ **do** Run $\mathsf{NewLDT}_D^{\widetilde{\Pi},\widetilde{\Pi'}}$
7 Run $\mathsf{ZERO}_H^{\widetilde{\Pi}_{\mathsf{Val}},\widetilde{\Pi}_{\mathcal{M}_{\mathsf{Val}}},\widetilde{\Pi'}_{\mathcal{M}_{\mathsf{Val}}}}$
8 Check if $\widetilde{\Pi}_{\chi'}[(\mathbf{a}, \mathbf{b}), L] \stackrel{?}{=} \widetilde{\Pi}_\chi[\mathbf{a}, L] - \widetilde{\Pi}_\chi[\mathbf{b}, L]$
9 Run $\mathsf{ZERO}_H^{\widetilde{\Pi}_{\mathsf{Prop}},\widetilde{\Pi}_{\mathcal{M}_{\mathsf{Prop}}},\widetilde{\Pi'}_{\mathcal{M}_{\mathsf{Prop}}}}$
10 **if** *any of the above tests return* REJECT **then return** REJECT
11 **else return** ACCEPT

Theorem 2.2 (The PCP Theorem [ALMSS98]). *There exists a randomized oracle algorithm $\mathcal{V}$, a constant $\gamma > 0$, and a polynomially growing function $\mathrm{size}(n)$ such that for every n and for every input graph $G = (V, E)$ over n vertices, we have the following.*

- *If $G \in$ 3-COLOR, then there exists $\Pi \in \{0,1\}^{\mathrm{size}(n)}$ such that*

$$\Pr[\mathcal{V}^{\Pi}(G) \text{ returns ACCEPT}] = 1.$$

- *If $G \notin$ 3-COLOR, then for every $\widetilde{\Pi} \in \{0,1\}^{\mathrm{size}(n)}$, we have*

$$\Pr[\mathcal{V}^{\widetilde{\Pi}}(G) \text{ returns REJECT}] \geqslant \gamma.$$

Furthermore, for every $\Pi \in \{0,1\}^{\mathrm{size}(n)}$, $\mathcal{V}^{\Pi}(G)$ runs in time $\mathrm{poly}(n)$, uses $\mathcal{O}(\log n)$ bits of randomness, and

makes $\mathcal{O}(1)$ oracle queries.

Proof of Theorem 2.2. We claim that Algorithm 8 is the desired PCP verifier for an appropriate setting of parameters as follows.

Choice of Parameters Let $c, c_1, c_2 \in \mathbb{N}$ be sufficiently large constants. Given $n \in \mathbb{N}$, let $h := \lceil \log n \rceil$, $m := \lceil \log n / \log\log n \rceil$ (so $h^m \geqslant n$)[14], $t := 2\lceil c_1 \log(hm) \rceil$ (so $2^t - 1$ is divisible by 3), $D = c_2 hm$, and $m_1 := \lfloor D^{1/c} \rfloor + 1$. Let $q := 2^t = \text{poly}\log n$ and $H \subseteq \mathbb{F}_q$ be an arbitrary subset of size h.

Proof size, Query complexity, and Running time We now analyze the resulting complexity of our PCP verifier: We first note that $m_1 = \mathcal{O}(m)$ assuming[15] $c \geqslant 3$. Hence the total proof size is $q^{\mathcal{O}(m)} \cdot q^{\mathcal{O}(cm_1)} \cdot 2^{\mathcal{O}(t^3)} = q^{\mathcal{O}(m)} \cdot \text{poly}(n) = \text{poly}(n)$. The number of random bits used by the verifier (including all the subroutines) is $\mathcal{O}(m \log q + cm_1 \log q + t^3) = \mathcal{O}(\log n)$. Since there are $\mathcal{O}(1)$ calls to $\textsc{sc}$, $\widetilde{\Pi}_{\mathsf{Val}}$, $\widetilde{\Pi}_{\mathsf{Prop}}$, NewLDT, and ZERO, the total number of oracle queries made by the verifier is also $\mathcal{O}(1)$ as required. Finally, the total running time is dominated by $q^{\mathcal{O}(m)} = \text{poly}(n)$ needed to compute $\text{LDE}(E)$ in Line 2.

Completeness Suppose $\mathsf{Color} : H^m \to \{1, \omega, \omega^2\}$ is a proper 3-coloring of G. Recall the expected proof oracles from the beginning of Section 7. In particular, $\chi = \widehat{\mathsf{Color}} = \text{LDE}(\mathsf{Color})$ and $\widehat{E} = \text{LDE}(E)$. Consider the proof oracles as

$$\begin{aligned}
\widetilde{\Pi}_{\chi} &= \text{Had}^{(3)}(\chi), \quad \widetilde{\Pi'}_{\chi} = \text{Had}^{(3)}(\Psi^*(\chi_{\text{lines}})), \\
\widetilde{\Pi}_{\mathcal{M}_{\mathsf{Val}}} &= \text{Had}^{(1)}(\mathcal{M}_{\mathsf{Val}}), \quad \widetilde{\Pi'}_{\mathcal{M}_{\mathsf{Val}}} = \text{Had}^{(1)}(\Psi^*(\mathcal{M}_{\mathsf{Val},\text{lines}})), \\
\widetilde{\Pi}_{\chi'} &= \text{Had}^{(3)}(\chi'), \quad \widetilde{\Pi'}_{\chi'} = \text{Had}^{(3)}(\Psi^*(\chi'_{\text{lines}})), \\
\widetilde{\Pi}_{\mathcal{M}_{\mathsf{Prop}}} &= \text{Had}^{(1)}(\mathcal{M}_{\mathsf{Prop}}), \quad \widetilde{\Pi'}_{\mathcal{M}_{\mathsf{Prop}}} = \text{Had}^{(1)}(\Psi^*(\mathcal{M}_{\mathsf{Prop},\text{lines}})).
\end{aligned}$$

We claim that with these oracles, the verifier $\mathcal{V}$ returns $\textsc{Accept}$ with probability 1. We first remark that these oracles are well-defined since all the polynomials considered above have degree at most $5hm \leqslant D$ (taking $c_2 \geqslant 5$). From the completeness part of Theorem 4.1, all the tests $\text{NewLDT}^{\widetilde{\Pi},\widetilde{\Pi'}}$ in Line 6 always return $\textsc{Accept}$. We now move to the zero tests. We first note that the $\textsc{sc}(g, \Lambda)$ calls in Lines 4 and 5 return $g[\Lambda]$ using the linearity of $\text{Had}^{(3)}(\cdot)$. We now claim that the intermediate oracle function defined in Line 4 is $\widetilde{\Pi}_{\mathsf{Val}} \equiv \text{Had}^{(1)}(\mathsf{Val})$: this follows because for all $\mathbf{a}' \in \mathbb{F}_q^m$ and $L' \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$, using Claim 3.7, we have

$$\widetilde{\Pi}_{\mathsf{Val}}[\mathbf{a}', L'] = \widetilde{\Pi}_{\chi}[\mathbf{a}', \Lambda_{Y^3-1,L'}] = L'(\rho(\chi(\mathbf{a}')^3 - 1)) = \text{Had}^{(1)}(\mathsf{Val})(\mathbf{a}', L').$$

Next, we observe that the check in Line 8 always passes since L is homogeneous degree-1:

$$\widetilde{\Pi}_{\chi'}[(\mathbf{a}, \mathbf{b}), L] = L(\rho(\chi'(\mathbf{a}, \mathbf{b}))) = L(\rho(\chi(\mathbf{a}) - \chi(\mathbf{b}))) = L(\rho(\chi(\mathbf{a}))) - L(\rho(\chi(\mathbf{b}))) = \widetilde{\Pi}_{\chi}[\mathbf{a}, L] - \widetilde{\Pi}_{\chi}[\mathbf{b}, L].$$

[14] We can add a suitable number of isolated vertices to make $|V| = h^m$.

[15] Indeed, for the same reason, we cannot set the value of c to be 1 as that would make m_1 at least $\Theta(hm)$, which is too large in our setting since we want the final oracle proof sizes $q^{O(m+m_1)}$ to be $\text{poly}(n)$.

Finally, we have $\widetilde{\Pi}_{\mathsf{Prop}} \equiv \mathrm{Had}^{(1)}(\mathsf{Prop})$ since for all $\mathbf{a}', \mathbf{b}' \in \mathbb{F}_q^m$ and $L' \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$:

$$\begin{aligned}\widetilde{\Pi}_{\mathsf{Prop}}[(\mathbf{a}', \mathbf{b}'), L'] &= \widetilde{\Pi}_{\chi'}[(\mathbf{a}', \mathbf{b}'), \Lambda_{\widehat{E}(\mathbf{a}', \mathbf{b}') \cdot (Y^3 - 1), L']] \\ &= L'(\rho(\widehat{E}(\mathbf{a}', \mathbf{b}') \cdot (\chi'(\mathbf{a}', \mathbf{b}')^3 - 1))) \\ &= \mathrm{Had}^{(1)}(\mathsf{Prop})((\mathbf{a}', \mathbf{b}'), L').\end{aligned}$$

Hence, using the completeness property of ZERO (Theorem 6.1) along with Observation 7.1 and Observation 7.2, we conclude that the tests in Lines 7 and 9 of $\mathcal{V}$ also always return Accept.

Soundness Suppose the rejection probability of $\mathcal{V}$ is at most γ, where $\gamma = \gamma(c, c_1, c_2) > 0$ is a sufficiently small constant that we will choose later, for an input graph G and some received proof oracles $\widetilde{\Pi}_{\chi}, \widetilde{\Pi'}_{\chi}, \widetilde{\Pi}_{\mathcal{M}_{\mathsf{Val}}}, \widetilde{\Pi'}_{\mathcal{M}_{\mathsf{Val}}}, \widetilde{\Pi}_{\chi'}, \widetilde{\Pi'}_{\chi'}, \widetilde{\Pi}_{\mathcal{M}_{\mathsf{Prop}}}, \widetilde{\Pi'}_{\mathcal{M}_{\mathsf{Prop}}}$. We will show that $G \in$ 3-COLOR. Indeed, we will show that $\widetilde{\Pi}_{\chi}$ is close to (the Hadamard encoding of a low-degree-extension of) a proper 3-coloring of G. Since $\mathcal{V}$ rejects if and only if any of the internal tests reject, we observe that each call of NewLDT in Line 6 rejects with probability at most γ. Hence, by the soundness guarantee of NewLDT (Theorem 4.1), we note that there exist polynomials $\widetilde{\chi} \in \mathcal{P}_{4cD}(m, \mathbb{F}_q)$, $\widetilde{\mathcal{M}_{\mathsf{Val}}} \in \mathcal{P}_{4cD}(2m, \mathbb{F}_q)$, $\widetilde{\chi'} \in \mathcal{P}_{4cD}(2m, \mathbb{F}_q)$, and $\widetilde{\mathcal{M}_{\mathsf{Prop}}} \in \mathcal{P}_{4cD}(4m, \mathbb{F}_q)$ such that they are $\mathcal{O}_c(\gamma)$-close to their respective oracles, i.e.,

$$\delta\left(\widetilde{\Pi}_{\chi},\ \mathrm{Had}^{(3)}(\widetilde{\chi})\right), \quad \delta\left(\widetilde{\Pi}_{\chi}|_{\leqslant 1}, \mathrm{Had}^{(1)}(\widetilde{\chi})\right) \leqslant \mathcal{O}_c(\gamma), \tag{9}$$

$$\delta\left(\widetilde{\Pi}_{\mathcal{M}_{\mathsf{Val}}},\ \mathrm{Had}^{(1)}\left(\widetilde{\mathcal{M}_{\mathsf{Val}}}\right)\right) \leqslant \mathcal{O}_c(\gamma), \tag{10}$$

$$\delta\left(\widetilde{\Pi}_{\chi'},\ \mathrm{Had}^{(3)}\left(\widetilde{\chi'}\right)\right), \quad \delta\left(\widetilde{\Pi}_{\chi'}|_{\leqslant 1},\ \mathrm{Had}^{(1)}\left(\widetilde{\chi'}\right)\right) \leqslant \mathcal{O}_c(\gamma), \tag{11}$$

$$\delta\left(\widetilde{\Pi}_{\mathcal{M}_{\mathsf{Prop}}},\ \mathrm{Had}^{(1)}\left(\widetilde{\mathcal{M}_{\mathsf{Prop}}}\right)\right) \leqslant \mathcal{O}_c(\gamma). \tag{12}$$

Now we build oracles based on the above polynomials and argue that they are close to the corresponding intermediate oracles constructed in Algorithm 8. In particular, let $\widetilde{\mathsf{Val}} \in \mathcal{P}_{12cD}(m, \mathbb{F}_q)$, $\widetilde{\chi''} \in \mathcal{P}_{4cD}(2m, \mathbb{F}_q)$, and $\widetilde{\mathsf{Prop}} \in \mathcal{P}_{20cD}(2m, \mathbb{F}_q)$ be the polynomials defined by

$$\widetilde{\mathsf{Val}}(\mathbf{a}') := \widetilde{\chi}(\mathbf{a}')^3 - 1, \quad \widetilde{\chi''}(\mathbf{a}', \mathbf{b}') := \widetilde{\chi}(\mathbf{a}') - \widetilde{\chi}(\mathbf{b}'), \quad \widetilde{\mathsf{Prop}}(\mathbf{a}', \mathbf{b}') := \widehat{E}(\mathbf{a}', \mathbf{b}') \cdot (\widetilde{\chi''}(\mathbf{a}', \mathbf{b}')^3 - 1).$$

We quickly argue that our constructed polynomial $\widetilde{\chi''}$ in fact is identical to the polynomial $\widetilde{\chi'}$ that was guaranteed from NewLDT (see Equation (11)).

Claim 7.3. *$\widetilde{\chi''}$ and $\widetilde{\chi'}$ are identical as polynomials.*

Proof of Claim 7.3. Since the check in Line 8 rejects with probability at most γ, we have

$$\Pr\left[\widetilde{\Pi}_{\chi'}[(\mathbf{a}, \mathbf{b}), L] \neq \widetilde{\Pi}_{\chi}[\mathbf{a}, L] - \widetilde{\Pi}_{\chi}[\mathbf{b}, L]\right] \leqslant \gamma.$$

From Equation (9) and Equation (11), we have

$$\Pr\left[\widetilde{\Pi}_{\chi}[\mathbf{a}, L] \neq \mathrm{Had}^{(1)}\left(\widetilde{\chi}\right)(\mathbf{a}, L)\right], \quad \Pr\left[\widetilde{\Pi}_{\chi'}[(\mathbf{a}, \mathbf{b}), L] \neq \mathrm{Had}^{(1)}\left(\widetilde{\chi'}\right)((\mathbf{a}, \mathbf{b}), L)\right] \leqslant 2 \cdot \mathcal{O}_c(\gamma).$$

Combining the above three inequalities by a union bound and Observation 3.8, we get

$$\Pr_{\mathbf{a},\mathbf{b}\sim\mathbb{F}_q^m}[\widetilde{\chi'}(\mathbf{a}, \mathbf{b}) \neq \widetilde{\chi}(\mathbf{a}) - \widetilde{\chi}(\mathbf{b})] \leqslant \mathcal{O}_c(\gamma),$$

which by Theorem 2.5 implies that $\widetilde{\chi'} \equiv \widetilde{\chi''}$ (taking γ to be sufficiently smaller than $1 - 4cD/q$). This finishes the proof of Claim 7.3. ■

In the rest of the proof, we will show that

$$\widetilde{\mathsf{Val}}|_{H^m} \equiv 0, \text{ and} \tag{13}$$

$$\widetilde{\mathsf{Prop}}|_{H^{2m}} \equiv 0. \tag{14}$$

If both Equation (13) and Equation (14) hold, then the converse direction of Observation 7.1 and Observation 7.2 imply that $\widetilde{\chi}|_V$ is a proper 3-coloring of G, which would give us the desired soundness guarantee. Thus it remains to show Equation (13) and Equation (14) hold.

Proof of Equation (13) Observe that for a uniformly random input in the domain of $\widetilde{\Pi}_{\mathsf{Val}}$, the two query points inside the SC call in Line 4 are marginally uniform in the domain of $\widetilde{\Pi}_{\chi}$ because $R \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3}$. Now, suppose $\mathbf{a}' \in \mathbb{F}_q^m$, $L' \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$, and $R \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3}$ are such that $\widetilde{\Pi}_{\chi}[\mathbf{a}', \Lambda_{L'} + R] = \mathrm{Had}^{(3)}\left(\widetilde{\chi}\right)(\mathbf{a}', \Lambda_{L'} + R)$ and $\widetilde{\Pi}_{\chi}[\mathbf{a}', R] = \mathrm{Had}^{(3)}\left(\widetilde{\chi}\right)(\mathbf{a}', R)$. Then, by the definition in Line 4 and using linearity of degree-3 Hadamard encoding and Claim 3.7, we have,

$$\widetilde{\Pi}_{\chi}[\mathbf{a}', \Lambda_{L'} + R] - \widetilde{\Pi}_{\chi}[\mathbf{a}', R] = \mathrm{Had}^{(1)}\left(\widetilde{\mathsf{Val}}\right)(\mathbf{a}', L').$$

Thus, by a union bound and Equation (9), we get

$$\mathop{\mathbb{E}}_{R\sim\mathbb{F}_2[\mathbf{Z}]^{\leqslant 3}}\left[\delta(\widetilde{\Pi}_{\mathsf{Val}}, \mathrm{Had}^{(1)}\left(\widetilde{\mathsf{Val}}\right))\right] \leqslant 2 \cdot \delta\left(\widetilde{\Pi}_{\chi}, \mathrm{Had}^{(3)}\left(\widetilde{\chi}\right)\right) \leqslant \mathcal{O}_c(\gamma). \tag{15}$$

Since $\mathcal{V}$ rejects with probability at most γ by assumption, we also have

$$\Pr_{R\sim\mathbb{F}_2[\mathbf{Z}]^{\leqslant 3}}\left[\mathsf{ZERO}_H^{\widetilde{\Pi}_{\mathsf{Val}}, \widetilde{\Pi}_{\mathcal{M}_{\mathsf{Val}}}, \widetilde{\Pi}'_{\mathcal{M}_{\mathsf{Val}}}} \text{ returns } \textsc{Reject}\right] \leqslant \gamma,$$

where the probability is also over the randomness of the test.

Combining the above two bounds, we observe that there exists $R \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3}$ such that both the following two conditions hold simultaneously for some $\gamma' = \mathcal{O}_c(\gamma)$:

$$\delta(\widetilde{\Pi}_{\mathsf{Val}}, \mathrm{Had}^{(1)}\left(\widetilde{\mathsf{Val}}\right)) \leqslant \gamma' \text{ and } \Pr\left[\mathsf{ZERO}_H^{\widetilde{\Pi}_{\mathsf{Val}}, \widetilde{\Pi}_{\mathcal{M}_{\mathsf{Val}}}, \widetilde{\Pi}'_{\mathcal{M}_{\mathsf{Val}}}} \text{ returns } \textsc{Reject}\right] \leqslant \gamma',$$

where we fix $\widetilde{\Pi}_{\mathsf{Val}}$ corresponding to the above R in the SC call used to define $\widetilde{\Pi}_{\mathsf{Val}}$.

Using the above bounds along with Equation (10) and applying the soundness part of ZERO (Theorem 6.1), we conclude that $\widetilde{\mathsf{Val}}|_{H^m} \equiv 0$ assuming that $\gamma' < \eta$, where $\eta := \Omega_c(1) - 100cDh/q - \gamma'$ is from Theorem 6.1. The inequality $\gamma' < \eta$ indeed holds true for some suitable $\gamma' = \mathcal{O}_c(1)$, by using $q = \omega(h^2 m)$. Thus we have established Equation (13).

Proof of Equation (14) The argument is similar to above, but we work with the polynomial $\widetilde{\chi'}$ instead of $\widetilde{\chi}$. Suppose $\mathbf{a}', \mathbf{b}' \in \mathbb{F}_q^m$, $L' \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}$, and $R \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3}$ are such that

$$\begin{aligned}
\widetilde{\Pi}_{\chi'}[(\mathbf{a}', \mathbf{b}'), \Lambda_{\mathbf{a}',\mathbf{b}',L'} + R] &= \mathrm{Had}^{(3)}\left(\widetilde{\chi'}\right)((\mathbf{a}', \mathbf{b}'), \Lambda_{\mathbf{a}',\mathbf{b}',L'} + R) \\
\widetilde{\Pi}_{\chi'}[(\mathbf{a}', \mathbf{b}'), R] &= \mathrm{Had}^{(3)}\left(\widetilde{\chi'}\right)((\mathbf{a}', \mathbf{b}'), R)
\end{aligned}$$

Then by the definition of $\widetilde{\Pi}_{\mathsf{Prop}}$ in Line 5 of Algorithm 8, we get that $\widetilde{\Pi}_{\mathsf{Prop}}[(\mathbf{a}', \mathbf{b}'), L']$ is equal to

$$\begin{aligned}
\mathrm{Had}^{(3)}\left(\widetilde{\chi'}\right)((\mathbf{a}', \mathbf{b}'), \Lambda_{\mathbf{a}',\mathbf{b}',L'}) &= \Lambda_{\mathbf{a}',\mathbf{b}',L'}(\rho(\widetilde{\chi'}(\mathbf{a}', \mathbf{b}'))) \\
&= L'(\rho(\widehat{E}(\mathbf{a}', \mathbf{b}') \cdot (\widetilde{\chi'}(\mathbf{a}', \mathbf{b}')^3 - 1))) && \text{(using Claim 3.7)} \\
&= L'(\rho(\widetilde{\mathsf{Prop}}(\mathbf{a}', \mathbf{b}'))) && \text{(using Claim 7.3)} \\
&= \mathrm{Had}^{(1)}\left(\widetilde{\mathsf{Prop}}\right)((\mathbf{a}', \mathbf{b}'), L').
\end{aligned}$$

Thus, using Equation (11) and following a similar argument as in the derivation of Equation (15), we obtain

$$\mathop{\mathbb{E}}_{R \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 3}}[\delta(\widetilde{\Pi}_{\mathsf{Prop}}, \mathrm{Had}^{(1)}\left(\widetilde{\mathsf{Prop}}\right))] \leqslant \mathcal{O}_c(\gamma).$$

Finally, using the fact that $\widetilde{\Pi}_{\mathcal{M}_{\mathsf{Prop}}}$ is $\mathcal{O}_c(\gamma)$-close to degree-$4cD$ (i.e., Equation (12)) and the soundness guarantee of $\mathsf{ZERO}_H^{\widetilde{\Pi}_{\mathsf{Prop}}, \widetilde{\Pi}_{\mathcal{M}_{\mathsf{Prop}}}, \widetilde{\Pi'}_{\mathcal{M}_{\mathsf{Prop}}}}$, we conclude that

$$\widetilde{\mathsf{Prop}}|_{H^{2m}} \equiv 0,$$

completing the proof of Equation (14) and Theorem 2.2. ■

Acknowledgments

We would like to thank Oded Goldreich and Prahladh Harsha for conversations leading to this work. We thank Troy Lee for pointing out a mistake in an earlier version of Lemma 3.3. We thank an anonymous reviewer of SOSA 2027 for pointing us to the encoding of [DFKRS11]. GPT-6.1 Sol was used for proofreading and formatting. No part of the article is AI-generated.

References

[ABSSW26a] Prashanth Amireddy, Amik Raj Behera, Srikanth Srinivasan, Madhu Sudan, and Sophus Valentin Willumsgaard. "Ideals, Macaulay Bases, and PCPs". In: *Proceedings of the 58th Annual ACM Symposium on Theory of Computing, STOC 2026, Salt Lake City, UT, USA, June 22-26, 2026*. Ed. by Aditya Bhaskara and Artur Czumaj. ACM, 2026, pp. 1092–1103. DOI: 10.1145/3798129.3800821. URL: https://doi.org/10.1145/3798129.3800821 (cit. on pp. 2, 4, 9, 33).

[ABSSW26b] Prashanth Amireddy, Amik Raj Behera, Srikanth Srinivasan, Madhu Sudan, and Sophus Valentin Willumsgaard. "Implementation of a Simple Algebraic PCP". In: *Github survey* (2026). URL: https://github.com/SophusWald/Implementaiton-of-A-Simple-Algebraic-Proof-of-the-PCP-Theorem- (cit. on pp. 1, 53).

[ALMSS98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. "Proof Verification and the Hardness of Approximation Problems". In: *J. ACM* 45.3 (1998), pp. 501–555. DOI: 10.1145/278298.278306. URL: https://doi.org/10.1145/278298.278306 (cit. on pp. 1, 2, 4, 6, 9–11, 17, 18, 37).

[AS98] Sanjeev Arora and Shmuel Safra. "Probabilistic Checking of Proofs: A New Characterization of NP". In: *J. ACM* 45.1 (1998), pp. 70–122. DOI: 10.1145/273865.273901. URL: https://doi.org/10.1145/273865.273901 (cit. on p. 1).

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. "Non-Deterministic Exponential Time has Two-Prover Interactive Protocols". In: *Comput. Complex.* 1 (1991), pp. 3–40. DOI: 10.1007/BF01200056. URL: https://doi.org/10.1007/BF01200056 (cit. on p. 2).

[BGHMRS15] Boaz Barak, Parikshit Gopalan, Johan Håstad, Raghu Meka, Prasad Raghavendra, and David Steurer. "Making the Long Code Shorter". In: *SIAM J. Comput.* 44.5 (2015), pp. 1287–1324. DOI: 10.1137/130929394. URL: https://doi.org/10.1137/130929394 (cit. on p. 3).

[BCHKS96] M. Bellare, D. Coppersmith, J. Hastad, M. Kiwi, and M. Sudan. "Linearity testing in characteristic two". In: *IEEE Transactions on Information Theory* 42.6 (1996), pp. 1781–1795. DOI: 10.1109/18.556674 (cit. on p. 47).

[BGHSV06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. "Robust PCPs of Proximity, Shorter PCPs, and Applications to Coding". In: *SIAM J. Comput.* 36.4 (2006), pp. 889–974. DOI: 10.1137/S0097539705446810. URL: https://doi.org/10.1137/S0097539705446810 (cit. on pp. 1, 2).

[BS08] Eli Ben-Sasson and Madhu Sudan. "Short PCPs with Polylog Query Complexity". In: *SIAM J. Comput.* 38.2 (2008), pp. 551–607. DOI: 10.1137/050646445. URL: https://doi.org/10.1137/050646445 (cit. on pp. 2, 4).

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. “Self-Testing/Correcting with Applications to Numerical Problems”. In: *J. Comput. Syst. Sci.* 47.3 (1993), pp. 549–595. DOI: 10.1016/0022-0000(93)90044-W. URL: https://doi.org/10.1016/0022-0000(93)90044-W (cit. on pp. 17, 47).

[DL78] Richard A. DeMillo and Richard J. Lipton. “A probabilistic remark on algebraic program testing”. In: *Information Processing Letters* 7.4 (1978), pp. 193–195. DOI: 10.1016/0020-0190(78)90067-4 (cit. on p. 7).

[Din07] Irit Dinur. “The PCP theorem by gap amplification”. In: *J. ACM* 54.3 (2007), p. 12. DOI: 10.1145/1236457.1236459. URL: https://doi.org/10.1145/1236457.1236459 (cit. on pp. 1, 2).

[DFKRS11] Irit Dinur, Eldar Fischer, Guy Kindler, Ran Raz, and Shmuel Safra. “PCP Characterizations of NP: Toward a Polynomially-Small Error-Probability”. In: *Comput. Complex.* 20.3 (2011), pp. 413–504. DOI: 10.1007/S00037-011-0014-4. URL: https://doi.org/10.1007/s00037-011-0014-4 (cit. on pp. 3, 4, 41).

[DR06] Irit Dinur and Omer Reingold. “Assignment Testers: Towards a Combinatorial Proof of the PCP Theorem”. In: *SIAM J. Comput.* 36.4 (2006), pp. 975–1024. DOI: 10.1137/S0097539705446962. URL: https://doi.org/10.1137/S0097539705446962 (cit. on p. 1).

[FS95] Katalin Friedl and Madhu Sudan. “Some Improvements to Total Degree Tests”. In: *Third Israel Symposium on Theory of Computing and Systems, ISTCS 1995, Tel Aviv, Israel, January 4-6, 1995, Proceedings*. IEEE Computer Society, 1995, pp. 190–198. DOI: 10.1109/ISTCS.1995.377032. URL: https://doi.org/10.1109/ISTCS.1995.377032 (cit. on pp. 8, 45).

[Gol25] Oded Goldreich. “Proving the PCP Theorem with 1.5 proof compositions (or yet another PCP construction)”. In: *Electron. Colloquium Comput. Complex.* TR25, TR25-182 (2025). ECCC: TR25-182. URL: https://eccc.weizmann.ac.il/report/2025/182 (cit. on pp. 2, 4).

[HKSS24] Prahladh Harsha, Mrinal Kumar, Ramprasad Saptharishi, and Madhu Sudan. “An Improved Line-Point Low-Degree Test”. In: *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*. IEEE, 2024, pp. 1883–1892. DOI: 10.1109/FOCS61266.2024.00113. URL: https://doi.org/10.1109/FOCS61266.2024.00113 (cit. on p. 10).

[KS08] Tali Kaufman and Madhu Sudan. “Algebraic property testing: the role of invariance”. In: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*. Ed. by Cynthia Dwork. ACM, 2008, pp. 403–412. DOI: 10.1145/1374376.1374434. URL: https://doi.org/10.1145/1374376.1374434 (cit. on p. 13).

[Lee26] Troy Lee. *TCS formalizations*. Version pcp-v0.1.0. Lean 4 formalization of the PCP theorem, library PCP. Oct. 2026. DOI: 10.5281/zenodo.23445994. URL: https://doi.org/10.5281/zenodo.23445994 (cit. on p. 1).

[Ore22] Øystein Ore. "Über höhere kongruenzen". In: *Norsk Mat. Forenings Skrifter* 1.7 (1922), p. 15 (cit. on p. 7).

[RS96] Ronitt Rubinfeld and Madhu Sudan. "Robust Characterizations of Polynomials with Applications to Program Testing". In: *SIAM J. Comput.* 25.2 (1996), pp. 252–271. DOI: 10.1137/S0097539793255151. URL: https://doi.org/10.1137/S0097539793255151 (cit. on pp. 3, 13).

[Sch80] Jacob T. Schwartz. "Fast Probabilistic Algorithms for Verification of Polynomial Identities". In: *J. ACM* 27.4 (1980), pp. 701–717. DOI: 10.1145/322217.322225 (cit. on p. 7).

[Sho05] Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, 2005 (cit. on p. 7).

[Zip79] Richard Zippel. "Probabilistic algorithms for sparse polynomials". In: *Symbolic and Algebraic Computation*. Springer Berlin Heidelberg, 1979, pp. 216–226. DOI: 10.1007/3-540-09519-5_73 (cit. on p. 7).

A Proofs from Section 2 and Section 3

Proof of Fact 2.7. Before we prove this characterization for arbitrary m, it will be helpful to prove it for $m = 1$. We want to show that a function $f : \mathbb{F}_q \to \mathbb{F}_q$ is a degree-c polynomial if and only if

$$\sum_{i=0}^{c+1} \eta_i \cdot f(u + \zeta_i \cdot v) = 0, \qquad \text{for every } u, v \in \mathbb{F}_q.$$

Let us start with the forward direction. By linearity, it suffices to show for monomials of degree at most c, i.e., for $1, t, t^2, \ldots, t^c$. Fix any monomial t^k for $0 \leqslant k \leqslant c$. We will first show it for $u = 0$ and $v = 1$, i.e.,

$$\sum_{i=0}^{c+1} \eta_i \cdot \zeta_i^k = 0. \tag{16}$$

Note that t^k sends the points $\zeta_1, \ldots, \zeta_{c+1}$ to $\zeta_1^k, \ldots, \zeta_{c+1}^k$, so if we express t^k in Lagrange interpolation basis and substitute $t = \zeta_0 = 0$, we get

$$\sum_{i=1}^{c+1} \eta_i \cdot \zeta_i^k \;=\; \zeta_0^k.$$

Since $\eta_0 = -1$, we get $\sum_{i=0}^{c+1} \eta_i \zeta_i^k = 0$, thus showing Equation (16).
Now choose an arbitrary u and v. Note that for every $0 \leqslant i \leqslant c+1$, we have $(u + \zeta_i \cdot v)^k$ is a sum of $u^j v^{k-j} \cdot \zeta_i^{k-j}$ for different j. Thus we can use Equation (16) repeatedly for different j and this also handles

arbitrary u, v.

Now we will argue that if a function f satisfies the property for every $u, v \in \mathbb{F}_q$, then f is indeed a degree-c polynomial. The idea is to first construct an actual degree-c polynomial $P(t)$ by interpolation and then show that $P(x) = f(x)$ for every $x \in \mathbb{F}_q$. Consider the points $\{\zeta_0, \zeta_1, \ldots, \zeta_c\}$ (this corresponds to looking at the line where $u = 0$ and $v = 1$). Now interpolate a degree-c polynomial $P(t)$ through these points with evaluations $\{f(\zeta_0), f(\zeta_1), \ldots, f(\zeta_c)\}$. From the above argument and the assumption on f, we know the following two equalities hold:

$$\sum_{i=0}^{c+1} \eta_i \cdot P(\zeta_i) \;=\; 0 \quad \text{and} \quad \sum_{i=0}^{c+1} \eta_i \cdot f(\zeta_i) \;=\; 0.$$

Since $P(\zeta_i) = f(\zeta_i)$ for every $0 \leqslant i \leqslant c$, we also get that $P(\zeta_{c+1}) = f(\zeta_{c+1})$. By choice of ζ_i's, we have shown that P and f agree on $0, \zeta, \zeta^2, \ldots, \zeta^{c+1}$. To show that P and f agree on ζ^{c+2}, we repeat the above argument by interpolating a degree-c polynomial through points $\{0, \zeta^2, \ldots, \zeta^{c+1}\}$, which corresponds to setting $u = 0$ and $v = \zeta$. Note that this will agree with P from the previous step on $c + 1$ points namely $\{0, \zeta^2, \ldots, \zeta^{c+1}\}$, and two degree-$c$ polynomials agree on $c + 1$ points if and only if they are the same polynomial. Thus we will also get that $P(\zeta^{c+2}) = f(\zeta^{c+2})$. We keep repeating this argument till we reach ζ^{q-1}, and since ζ is a generator of $\mathbb{F}_q^\times$, we would have shown that P and f agree everywhere. Therefore, f is a degree-c polynomial.

Now we would argue that the above characterization holds for every m. We start by arguing for the forward direction. If $P : \mathbb{F}_q^m \to \mathbb{F}_q$ is a degree-c polynomial, then for every pair $(\mathbf{u}, \mathbf{v}) \in \mathbb{F}_q^m$, the restriction $P(\mathbf{u} + t \cdot \mathbf{v})$ is a univariate polynomial $P_{\mathbf{u},\mathbf{v}} : \mathbb{F}_q \to \mathbb{F}_q$ of degree-c. From the above discussion, we get that

$$\sum_{i=0}^{c+1} \eta_i \cdot P_{\mathbf{u},\mathbf{v}}(\zeta_i) = 0.$$

Now it remains to argue the reverse direction. Suppose $f : \mathbb{F}_q^m \to \mathbb{F}_q$ is a function that satisfies the property stated in Fact 2.7 for every pair $(\mathbf{u}, \mathbf{v})$. Fix an arbitrary $(\mathbf{u}, \mathbf{v})$ and let $f_{\mathbf{u},\mathbf{v}}(t) = f(\mathbf{u} + t \cdot \mathbf{v})$ be the univariate restriction to line $\ell_{\mathbf{u},\mathbf{v}}$. We will first show that $f_{\mathbf{u},\mathbf{v}}$ is degree-c, by relying on the above discussion. Choose any $\alpha, \beta \in \mathbb{F}_q$ and consider the pair $(\mathbf{u} + \alpha\mathbf{v}, \beta\mathbf{v})$. Then, we have

$$\sum_{i=0}^{c+1} \eta_i \cdot f_{\mathbf{u},\mathbf{v}}(\alpha + \zeta_i \cdot \beta) \;=\; \sum_{i=0}^{c+1} \eta_i \cdot f(\mathbf{u} + (\alpha + \zeta_i \cdot \beta) \cdot \mathbf{v}) \;=\; 0.$$

Therefore, $f_{\mathbf{u},\mathbf{v}}(t)$ is of degree-c. Since our choice of $(\mathbf{u}, \mathbf{v})$ was arbitrary, this holds for every pair. Now [FS95, Lemma 4] implies that f is indeed a degree-c polynomial. This finishes the reverse direction for arbitrary m and also finishes the proof of Fact 2.7. ■

Proof of Claim 3.2. For an integer $0 \leqslant k \leqslant d$, let $[k_{c-1}, \ldots, k_0]$ denote its m_1-ary representation. Letting

$P(Y) = \sum_{k=0}^{d} \alpha_k \cdot Y^k$, we have

$$\Psi_{d,c,m_1}(P) = \sum_{k=0}^{d} \alpha_k \cdot X_{0,k_0} \cdot X_{1,k_1} \cdots X_{c-1,k_{c-1}}$$

which then implies

$$\begin{aligned}(\Psi_{d,c,m_1}(P))(\Phi_{c,m_1}(\lambda)) &= \sum_{k=0}^{d} \alpha_k \cdot \lambda^{k_0} \cdot \lambda^{k_1 \cdot m_1} \cdots \lambda^{k_{c-1} \cdot m_1^{c-1}} \\ &= \sum_{k=0}^{d} \alpha_k \cdot \lambda^k = P(\lambda).\end{aligned}$$

This finishes the proof of Claim 3.2. ■

Proof of Claim 3.7. We first show it when L is homogeneous so $L(x+y) = L(x) + L(y)$. Let $v_i := \rho^{-1}(e_i)$, where (e_i) is the standard basis for $\mathbb{F}_2^t$. Let

$$P(Y) = \sum_{k=0}^{r} c_k Y^k.$$

In this case we have for any element $\lambda = \sum_{i \in [t]} \alpha_i v_i \in \mathbb{F}_q$,

$$\mathrm{Had}^{(1)}\left(P(\lambda)\right)(L) = L(\rho(P(\lambda))) = \sum_{k=0}^{r} L(\rho(c_k \lambda^k)) = \sum_{k=0}^{r} \sum_{i_1,\ldots,i_k \in [t]} L(\rho(c_k v_{i_1} \cdots v_{i_k}))\alpha_{i_1} \cdots \alpha_{i_k}.$$

We note that this expression is a degree r polynomial in the α_i's and define this to be $\Lambda_{P,L}(\alpha_1, \ldots, \alpha_t)$. We then have

$$\mathrm{Had}^{(r)}\left(\lambda\right)\left(\Lambda_{P,L}\right) = \Lambda_{P,L}(\rho(\lambda)) = \Lambda_{P,L}(\alpha_1, \ldots, \alpha_t),$$

showing it has the wanted property. If $P(Y) = \eta \cdot Y$, then

$$\Lambda_{P,L}(\alpha_1, \ldots, \alpha_t) = \sum_{i \in [t]} L(\rho(\eta v_i))\alpha_i,$$

showing that it is homogeneous in this case.

For $L' = L + b$ a sum of a homogeneous linear function and a constant $b \in \mathbb{F}_2$, we set $\Lambda_{P,L'} := \Lambda_{P,L} + b$. We then have

$$\mathrm{Had}^{(1)}\left(P(\lambda)\right)(L') = \mathrm{Had}^{(1)}\left(P(\lambda)\right)(L) + b = \mathrm{Had}^{(r)}\left(\lambda\right)\left(\Lambda_{P,L}\right) + b = \mathrm{Had}^{(r)}\left(\lambda\right)\left(\Lambda_{P,L'}\right).$$

This finishes the proof of Claim 3.7. ■

B Analysis of HadTest

To prove Lemma 3.10, we will use the local testability of the Hadamard code.

Theorem B.1 (BLR Linearity Testing [BLR93; BCHKS96])**.** *Let $r \in \mathbb{N}$ denote the degree parameter, and let $H : \mathbb{F}_2[Z_1, \ldots, Z_t]^{\leqslant r} \to \mathbb{F}_2$ be arbitrary. Let*

$$\varepsilon \;:=\; \Pr_{R,R' \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant r}}[H[R+R'] \;\neq\; H[R]+H[R']]$$

denote the rejection probability of the linearity test. Then the following holds:

1. *If H is $\mathrm{Had}^{(r)}(u)$ for some $u \in \mathbb{F}_q$, then $\varepsilon = 0$.*
2. *There is a linear function $\mathcal{L} : \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$ with $\delta(H, \mathcal{L}) \leqslant \varepsilon$.*

Theorem B.1 relates the rejection probability to distance for degree-1 Hadamard codes. Using this, we will also prove Lemma 3.10.

Proof of Lemma 3.10. For completeness of the algorithm, we observe that if $H = \mathrm{Had}^{(r)}(u)$ for some $u \in \mathbb{F}_q$, then H passes every test in HadTest and thus $\mathsf{HadTest}_r^H$ always returns Accept.

We now focus on the soundness of $\mathsf{HadTest}_r^H$. The following events can make the algorithm reject:

- Event $\mathcal{E}_{\mathrm{lin}}$ (line 2): $H[R+R'] \neq H[R]+H[R']$.
- Event $\mathcal{E}_{\mathrm{unit}}$ (line 3): $H[1+R]-H[R] \neq 1$.
- Event $\mathcal{E}_{\deg 1}$ (line 4): $H[L+R]-H[R] \neq H[L]$.
- Event $\mathcal{E}_{\mathrm{mult}}$ (line 5):

$$H[R+L\cdot P]-H[R] \;\neq\; (H[R+L]-H[R])\cdot(H[R+P]-H[R]).$$

We will first show that if $\mathcal{E}_{\mathrm{lin}}$, $\mathcal{E}_{\mathrm{unit}}$, and $\mathcal{E}_{\mathrm{mult}}$ all happen with probability less than δ, then H is δ-close to a degree-r Hadamard encoding $\mathrm{Had}^{(r)}(u)$ for some $u \in \mathbb{F}_q$.

First note that if event $\mathcal{E}_{\mathrm{lin}}$ happens with probability less than δ, then by Theorem B.1, there exists a linear function $\mathcal{L} : \mathbb{F}_2[Z_1, \ldots, Z_t]^{\leqslant r} \to \mathbb{F}_2$, such that $\delta(H, \mathcal{L}) \leqslant \delta$. By Claim 3.9, if $\mathcal{L}(1) = 1$ and $\mathcal{L}(P \cdot R) = \mathcal{L}(P) \cdot \mathcal{L}(R)$ for $\deg P + \deg R \leqslant r$, then $\mathcal{L}$ is the degree-r Hadamard encoding $\mathrm{Had}^{(r)}(u)$ for some $u \in \mathbb{F}_q$.

Unit condition By a union bound, we have

$$\Pr_{R \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant r}}[\mathcal{L}(1+R) \neq H[1+R] \;\vee\; \mathcal{L}(R) \neq H[R]] \leqslant 2\delta.$$

Since the event $\mathcal{E}_{\text{unit}}$ happens with probability less than δ, we also have

$$\Pr_{R \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant r}}[H[1+R] - H[R] \neq 1] \leqslant \delta.$$

Since $\delta < \frac{1}{3}$, this implies that $\mathcal{L}(1) = 1$.

Multiplicative condition For any function $f : \mathbb{F}_2[\mathbf{Z}]^{\leqslant r} \to \mathbb{F}_2$, define the function

$$\begin{aligned} A_f : \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1} \times \mathbb{F}_2[\mathbf{Z}]^{\leqslant r-1} &\to \mathbb{F}_2 \\ (L, P) &\mapsto f(L \cdot P) - f(L) \cdot f(P). \end{aligned}$$

If f is a linear function, then A_f is a bilinear form in the coefficients of L and P. By the polynomial distance lemma over small fields (Theorem 2.5), if $A_{\mathcal{L}}$ is non-zero, then it must be non-zero on at least $\frac{1}{4}$ of the inputs.

Using $\delta(\mathcal{L}, H) \leqslant \delta$ and union bound, we get,

$$\begin{aligned} \Pr_{L,R}[\mathcal{L}(L) \neq H(L+R) - H(R)] &\leqslant 2\delta \\ \Pr_{P,R}[\mathcal{L}(P) \neq H(P+R) - H(R)] &\leqslant 2\delta \\ \Pr_{L,P,R}[\mathcal{L}(L \cdot P) \neq H(L \cdot P + R) - H(R)] &\leqslant 2\delta, \end{aligned}$$

since all the points H is evaluated on are marginally uniformly random polynomials of degree at most r. It follows that

$$\begin{aligned} &\Pr_{L,P}[\mathcal{L}(L \cdot P) \neq \mathcal{L}(L) \cdot \mathcal{L}(P)] \\ &\leqslant 6 \cdot \delta + \Pr_{L,P,R}[\underbrace{H[R + L \cdot P] + H[R] \;\neq\; (H[R+L] + H[R]) \cdot (H[R+P] + H[R])}_{\mathcal{E}_{\text{mult}}}]. \end{aligned}$$

In the above inequality, the second summand is exactly the probability of the event $\mathcal{E}_{\text{mult}}$, which we assume to be at most δ. If we then assume $A_{\mathcal{L}}$ is non-zero, then we have

$$\frac{1}{4} \;\leqslant\; \Pr_{L,P}[\mathcal{L}(L \cdot P) - \mathcal{L}(L) \cdot \mathcal{L}(P) \neq 0] \;\leqslant\; 7\delta.$$

If we assume that $\delta < \frac{1}{28}$, we get a contradiction, so we have $\mathcal{L}(L \cdot P) = \mathcal{L}(L) \cdot \mathcal{L}(P)$ for any $L \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}, P \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant r-1}$. We can then extend this to $\mathcal{L}(P_1 \cdot P_2) = \mathcal{L}(P_1) \cdot \mathcal{L}(P_2)$ for any $P_1, P_2 \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant r-1}$ with $\deg P_1 + \deg P_2 \leqslant r$, by using that any polynomial P_1 can be written as a sum of products of linear polynomials. This shows that $\mathcal{L}$ satisfies the conditions to being the Hadamard encoding of some u, finishing the first part of the lemma.

For the second part, we show that the restriction of H to degree 1 is also close to $\mathrm{Had}^{(1)}(u)$. Since $\delta(H, \mathrm{Had}^{(r)}(u)) \leqslant \delta$, we have for any $L \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}$

$$\Pr_R\left[\mathrm{Had}^{(r)}(u)(L+R) \neq H(L+R) \;\vee\; \mathrm{Had}^{(r)}(u)(R) \neq H(R)\right] \;\leqslant\; 2\delta.$$

We then have

$$\Pr_{L}\left[H[L] \neq \mathrm{Had}^{(1)}(u)(L)\right] \leqslant 2\delta + \Pr_{L,R}[\underbrace{H[L] \neq H[L+R] + H[R]}_{\mathcal{E}_{\deg 1}}] \leqslant 3\delta.$$

This finishes the proof of Lemma 3.10. ■

C Proof of Local Correction

Proof of Theorem 5.1. For completeness, the argument is almost verbatim to the completeness argument of Algorithm 5 (see the proof of Theorem 4.1). Lines 2, 3, and 4 always return Accept and Line 6 returns

$$\sum_{i=1}^{c+1} f'[(\mathbf{a},\mathbf{b}), \Phi(0)+\zeta^i\mathbf{v}, \Lambda_{\eta_i,L}] = \sum_{i=1}^{c+1} \mathrm{Had}^{(1)}\left(\Psi^*(P_{\mathrm{lines}})\right)[(\mathbf{a},\mathbf{b}), \Phi(0)+\zeta^i\mathbf{v}, \Lambda_{\eta_i,L}],$$

which we show is equal to $\mathrm{Had}^{(1)}(P)(\mathbf{a},L)$. We note that since $\Psi(P_{\mathrm{lines}}(\mathbf{a},\mathbf{b}))$ is a degree-c polynomial, using Fact 2.8 we have that

$$\sum_{i=1}^{c+1} \eta_i \Psi(P_{\mathrm{lines}}(\mathbf{a},\mathbf{b}))(\Phi(0)+\zeta^i\mathbf{v}) = \Psi(P_{\mathrm{lines}}(\mathbf{a},\mathbf{b}))(\Phi(0)) = P_{\mathrm{lines}}(\mathbf{a},\mathbf{b})(0) = P(\mathbf{a}).$$

Hence, using that $\Lambda_{\eta_i,L}(\rho(x)) = L(\rho(\eta_i x))$ (Claim 3.7) and linearity of ρ and L, we conclude that NewLC indeed returns $L(\rho(P(\mathbf{a}))) = \mathrm{Had}^{(1)}(P)(\mathbf{a},L)$.

Soundness: Assume there exists a polynomial $P \in \mathcal{P}_{d'}(m,\mathbb{F}_q)$ such that f is δ-close to $\mathrm{Had}^{(1)}(P)$. Let η be the probability that $\mathsf{NewLC}^{f,f'}(\mathbf{a},L)$ rejects. We first introduce some notation similar to Theorem 4.1, which will be useful for the proof.

Notation: For every pair $(\mathbf{a},\mathbf{b}) \in \mathbb{F}_q^{2m}$, for every $\mathbf{u} \in \mathbb{F}_q^{cm_1}$, and for every $L \in \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}$, we will use $\mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,L]}(\mathbf{u};\mathbf{v})$ to denote the following sum:

$$\mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,L]}(\mathbf{u};\mathbf{v}) := \sum_{i=1}^{c+1} f'[(\mathbf{a},\mathbf{b}),\ \mathbf{u}+\zeta^i\mathbf{v},\ \Lambda_{\eta_i,L}].$$

Note that $\mathsf{NewLC}_{d,c}^{f,f'}(\mathbf{a},L;\mathbf{b},\mathbf{u},\mathbf{v},\lambda)$ returns $\mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,L]}(\Phi(0);\mathbf{v})$, if it does not reject. We define $Q_{\mathbf{a},\mathbf{b}}$ as the degree-c polynomial, whose degree-1 Hadamard encoding is closest to $f'[(\mathbf{a},\mathbf{b}),\cdot,\cdot]$, and $F(\mathbf{a},\mathbf{b}) = Q_{\mathbf{a},\mathbf{b}} \circ \Phi$. Note that $\deg(F(\mathbf{a},\mathbf{b})) \leqslant c(m_1-1)m_1^{c-1} \leqslant cm_1^c \leqslant d'$.

There are three events which can make the test reject:

- Event $\mathcal{E}_1$: $\mathsf{HadTest}_1^{f'[(\mathbf{a},\mathbf{b}),\mathbf{u},\cdot]}$ returns Reject.
- Event $\mathcal{E}_2$: $f'[(\mathbf{a},\mathbf{b}),\mathbf{u},L] + \sum_{i=1}^{c+1} f'[(\mathbf{a},\mathbf{b}),\mathbf{u}+\zeta^i\mathbf{v},\Lambda_{\eta_i,L}]$ is not equal to 0.
- Event $\mathcal{E}_3$: $\sum_{i=1}^{c+1} f'[(\mathbf{a},\mathbf{b}),\Phi(\lambda)+\zeta^i\mathbf{v},\Lambda_{\eta_i,L}]$ is not equal to $f[\mathbf{a}+\lambda\mathbf{b},L]$.

If the first two events happen with low probability, then by Lemma 4.3, with high probability $f'[(\mathbf{a},\mathbf{b}),\cdot,\cdot]$ must be close to the Hadamard encoding of a degree-c multivariate polynomial, which by definition is $Q_{\mathbf{a},\mathbf{b}}$. This then implies that we can do standard local correction on $f'[(\mathbf{a},\mathbf{b}),\Phi(\lambda),\cdot]$ to get the value of $\mathrm{Had}^{(1)}\left(Q_{\mathbf{a},\mathbf{b}}\right)[\Phi(\lambda),\cdot]$ for any $\lambda \in \mathbb{F}_q$. We state it more formally in the next claim.

Claim C.1. *Suppose that* $\mathsf{NewLC}_c^{f,f'}(\mathbf{a},L)$ *rejects with probability less than* η *for a random* L*. Then for any* $\lambda \in \mathbb{F}_q$ *we have*

$$\Pr_{\mathbf{b},\mathbf{v},L}\left[\mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,L]}(\Phi(\lambda);\mathbf{v}) \neq \mathrm{Had}^{(1)}\left(F(\mathbf{a},\mathbf{b})\right)(\lambda,L)\right] \leqslant \mathcal{O}_c(\eta),$$

where the probability is over $\mathbf{b} \sim \mathbb{F}_q^m$, $\mathbf{v} \sim \mathbb{F}_q^{cm_1}$, *and* $L \sim \mathbb{F}_2[\mathbf{Z}]^{\leqslant 1}_{\mathrm{hom}}$.

We assume the claim for now, and prove it at the end of the section.

If the event $\mathcal{E}_3$ also happens with low probability, then the local correction of f' is close to f, which also is close to $\mathrm{Had}^{(1)}(P)$ from our assumption. We then get the following bound by the union bound for a random $\lambda \in \mathbb{F}_q^\times$

$$\begin{aligned}
&\Pr_{\mathbf{b},\lambda,L}\left[\mathrm{Had}^{(1)}\left(F(\mathbf{a},\mathbf{b})\right)(\lambda,L) \neq \mathrm{Had}^{(1)}(P)\left(\mathbf{a}+\mathbf{b}\lambda,L\right)\right]\\
\leqslant\ &\Pr_{\mathbf{b},\mathbf{v},\lambda,L}\left[\mathrm{Had}^{(1)}\left(F(\mathbf{a},\mathbf{b})\right)(\lambda,L) \neq \mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,L]}(\Phi(\lambda);\mathbf{v})\right]\\
+\ &\Pr_{\mathbf{b},\mathbf{v},\lambda,L}\left[\mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,L]}(\Phi(\lambda);\mathbf{v}) \neq f[\mathbf{a}+\lambda\mathbf{b},L]\right]\\
+\ &\Pr_{\mathbf{b},\lambda,L}\left[f[\mathbf{a}+\lambda\mathbf{b},L] \neq \mathrm{Had}^{(1)}(P)\left(\mathbf{a}+\lambda\mathbf{b},L\right)\right]\\
\leqslant\ &\mathcal{O}_c(\eta)+\eta+2\delta.
\end{aligned}$$

The first summand is bounded by Claim C.1. The second summand is one of the rejection criteria for the test so it is bounded by η. The last summand is bounded by 2δ, by the assumption that $\mathrm{Had}^{(1)}(P)$ is δ-close to f, and half of linear polynomials are homogeneous. Applying Observation 3.8, we then get

$$\Pr_{\mathbf{b},\lambda}\left[F(\mathbf{a},\mathbf{b})(\lambda) \neq P(\mathbf{a}+\mathbf{b}\lambda)\right] \leqslant \mathcal{O}_c(\eta+\delta).$$

Apply Theorem 2.14[16] on P and F to get

$$\Pr_{\mathbf{b},\lambda}\left[F(\mathbf{a},\mathbf{b})(0) = P(\mathbf{a}) \quad \text{OR} \quad F(\mathbf{a},\mathbf{b})(\lambda) \neq P(\mathbf{a}+\mathbf{b}\lambda)\right] \geqslant 1-\frac{d'}{q-1}.$$

Combining the two inequalities together with Observation 3.8, we get

$$\Pr_{\mathbf{b},L}\left[\mathrm{Had}^{(1)}\left(F(\mathbf{a},\mathbf{b})\right)(0,L) \neq \mathrm{Had}^{(1)}(P)\left(\mathbf{a},L\right)\right] \leqslant \mathcal{O}_c(d'/q+\eta+\delta).$$

[16] One can also simply apply Theorem 2.5 here instead.

Then applying Claim C.1 with $\lambda = 0$ we have

$$\Pr_{\mathbf{b},\mathbf{v},L}\left[\mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,\cdot,L]}(\Phi(0);\mathbf{v}) \neq \mathrm{Had}^{(1)}\left(P\right)(\mathbf{a},L)\right] \leqslant \mathcal{O}_c(d'/q+\eta+\delta)+\mathcal{O}_c(\eta) = \mathcal{O}_c(d'/q+\eta+\delta).$$

Since NewLC returns $\mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,\cdot,L]}(\Phi(0);\mathbf{v})$ whenever it does not reject, and it rejects with probability at most η, a union bound gives

$$\Pr\left[\mathsf{NewLC}_{d,c}^{f,f'}(\mathbf{a},L) \text{ does not return } \mathrm{Had}^{(1)}\left(P\right)(\mathbf{a},L)\right] \leqslant \eta + \mathcal{O}_c(d'/q+\eta+\delta).$$

This finishes the proof of Theorem 5.1, assuming the proof of Claim C.1. ■

We end by proving Claim C.1.

Proof of Claim C.1. For $\mathbf{b} \in \mathbb{F}_q^m$, set

$$\mathrm{Rej}_{\mathbf{b}} := \Pr_{L,\mathbf{u},\mathbf{v},\lambda}\left[\mathsf{NewLC}_c^{f,f'}(\mathbf{a},L;\mathbf{b},\mathbf{u},\mathbf{v},\lambda) \text{ returns } \textsc{Reject}\right].$$

Now fix $\mathbf{b} \in \mathbb{F}_q^m$. We have $\Pr[\mathcal{E}_1|\mathbf{b}], \Pr[\mathcal{E}_2|\mathbf{b}] \leqslant \mathrm{Rej}_{\mathbf{b}}$, so applying Lemma 4.3 to $f'[(\mathbf{a},\mathbf{b}),\cdot,\cdot]$, we get that for every $\mathbf{u} \in \mathbb{F}_q^{cm_1}$,

$$\Pr_{\mathbf{v},L}\left[\mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,\cdot,L]}(\mathbf{u};\mathbf{v}) \neq \mathrm{Had}^{(1)}\left(Q_{\mathbf{a},\mathbf{b}}\right)(\mathbf{u},L)\right] \leqslant \mathcal{O}_c(\mathrm{Rej}_{\mathbf{b}}).$$

We will use this inequality with $\mathbf{u} = \Phi(\lambda)$. In this case by definition we have $F(\mathbf{a},\mathbf{b})(\lambda) = Q_{\mathbf{a},\mathbf{b}}(\Phi(\lambda))$. Averaging over $\mathbf{b}$ and using that $\mathbb{E}_{\mathbf{b}}[\mathrm{Rej}_{\mathbf{b}}] \leqslant \eta$ we then get

$$\Pr_{\mathbf{b},\mathbf{v},L}\left[\mathsf{StdLC}_c^{f'[(\mathbf{a},\mathbf{b}),\cdot,\cdot,L]}(\Phi(\lambda);\mathbf{v}) \neq \mathrm{Had}^{(1)}\left(F(\mathbf{a},\mathbf{b})(\lambda)\right)(L)\right] \leqslant \mathcal{O}_c(\eta),$$

giving the wanted claim. ■

D Proof of Zero-on-Grid Test

Proof of Theorem 6.1. We start by discussing completeness.

Completeness Line 2 and Line 3 always pass because of Corollary 2.10 and the completeness property of NewLC.

Soundness We will consider the following events (we will hide the internal randomness of NewLC for the sake of clarity in writing):

1. Event $\mathcal{E}_1$: $\mathsf{NewLC}_d^{\Pi,\Pi'}((\mathbf{a},\mathbf{0}),L)$ returns Reject or is not equal to 0, and
2. Event $\mathcal{E}_2$: $\mathsf{NewLC}_d^{\Pi,\Pi'}((\mathbf{a},Z_H(\mathbf{a})),L)$ returns Reject or is not equal to $f[\mathbf{a},L]$.

Suppose the probability of $\mathcal{E}_1 \vee \mathcal{E}_2$ is at most η. We will then show that $P|_{H^m} \equiv 0$. To do this, we show that the polynomial $M \in \mathcal{P}_{d'}(2m, \mathbb{F}_q)$ from the soundness hypothesis of Theorem 6.1 satisfies the properties from Corollary 2.10, i.e.,

1. $M(\mathbf{X}, \mathbf{0}) = 0$, as polynomials.
2. $M(\mathbf{X}, Z_H(\mathbf{X})) = P(\mathbf{X})$, as polynomials (here $P \in \mathcal{P}_{d'}(m, \mathbb{F}_q)$ from the soundness hypothesis of Theorem 6.1).

These two conditions imply that $P|_{H^m} \equiv 0$. To see this, note that the first condition implies that M can be written in the form

$$M(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} M_i(\mathbf{x}, \mathbf{y}) \cdot y_i$$

for some polynomials $M_1, \ldots, M_m$. From the second condition we then have

$$P(\mathbf{x}) = M(\mathbf{x}, Z_H(\mathbf{x})) = \sum_{i=1}^{m} M_i(\mathbf{x}, Z_H(\mathbf{x})) \cdot Z_H^{(i)}(\mathbf{x}),$$

so since $Z_H(\mathbf{x})$ vanishes on H^m, we have P also vanishes on H^m. We will now show that both conditions hold.

For the first condition we have the following union bound:

$$\begin{aligned}
\Pr_{\mathbf{a},L}\left[\mathrm{Had}^{(1)}(M)((\mathbf{a},\mathbf{0}),L) \neq 0\right] &\leqslant \Pr_{\mathbf{a},L}\left[\mathrm{Had}^{(1)}(M)((\mathbf{a},\mathbf{0}),L) \neq \mathsf{NewLC}_d^{\Pi,\Pi'}((\mathbf{a},\mathbf{0}),L)\right] \\
&\quad + \Pr_{\mathbf{a},L}\left[\mathsf{NewLC}_d^{\Pi,\Pi'}((\mathbf{a},\mathbf{0}),L) \neq 0\right] \\
&\leqslant \alpha(c) \cdot \left(d'/q + \eta + \delta\right)
\end{aligned}$$

where the first summand is bounded by $\alpha(c) \cdot (d'/q + \eta + \delta)$ from Theorem 5.1, where $\alpha(c)$ is a large constant depending on c, and the second summand is bounded by η by assumption that event $\mathcal{E}_1$ happens with probability at most η. From Observation 3.8, we then get

$$\Pr_{\mathbf{a}}\left[M(\mathbf{a},\mathbf{0}) \neq 0\right] \leqslant 2\alpha(c) \cdot \left(d'/q + \eta + \delta\right) < 1 - \frac{d'}{q},$$

where the last inequality holds under the assumption that $\eta < 1/(2\alpha(c)) - 2d'/q - \delta$. Since M is a polynomial of degree d', from Theorem 2.5, it follows that $M(\mathbf{X}, \mathbf{0}) = 0$. This shows that M satisfies the first condition.

The proof that the second condition also holds for M is almost identical. Recall that every polynomial in $Z_H(\mathbf{X})$ is a degree-$|H|$ polynomial, so $M(\mathbf{X}, Z_H(\mathbf{X}))$ is a degree-$(d'|H|)$ polynomial. By a union bound we have

$$\Pr_{\mathbf{a},L}\left[\mathrm{Had}^{(1)}(M)((\mathbf{a}, Z_H(\mathbf{a})),L) \neq \mathrm{Had}^{(1)}(P)(\mathbf{a},L)\right]$$

$$
\begin{aligned}
&\leqslant \Pr_{\mathbf{a},L}\left[\mathrm{Had}^{(1)}(M)((\mathbf{a}, Z_H(\mathbf{a})), L) \neq \mathsf{NewLC}_d^{\Pi,\Pi'}((\mathbf{a}, Z_H(\mathbf{a})), L)\right] \\
&+ \Pr_{\mathbf{a},L}\left[\mathsf{NewLC}_d^{\Pi,\Pi'}((\mathbf{a}, Z_H(\mathbf{a})), L) \neq f[\mathbf{a}, L]\right] \\
&+ \Pr_{\mathbf{a},L}\left[f[\mathbf{a}, L] \neq \mathrm{Had}^{(1)}(P)(\mathbf{a}, L)\right] \\
&\leqslant \alpha(c) \cdot \left(d'/q + \eta + \delta\right)
\end{aligned}
$$

where the first two summands are bounded as above, with the addition that the last summand is bounded by 2δ by assumption. Using Observation 3.8, we get,

$$
\Pr_{\mathbf{a}}\left[M(\mathbf{a}, Z_H(\mathbf{a})) \neq P(\mathbf{a})\right] \leqslant 2\alpha(c) \cdot \left(\frac{d'|H|}{q} + \eta + \delta\right) < 1 - \frac{d'|H|}{q},
$$

where the last inequality holds under the assumption that $\eta < 1/(2\alpha(c)) - 2d'|H|/q - \delta$. From Theorem 2.5, it then follows that $M(\mathbf{X}, Z_H(\mathbf{X})) = P(\mathbf{X})$ as polynomials. The two criteria of Corollary 2.10 are then satisfied, so we must have $P|_{H^m} \equiv 0$ finishing the proof. ■

E Honest Prover and Full PCP Verifier

We give a complete description of the PCP verifier for the language 3-COLOR in Algorithm 9. After that, in Algorithm 10, we give the description of an honest prover which takes as input a graph in 3-COLOR and outputs proof oracles that the verifier of Algorithm 9 always accepts. Given a proper 3-coloring, an honest prover can construct these proof oracles in polynomial time by running Algorithm 10.

Comparison with Algorithm 8 We note that Algorithm 9 is essentially equivalent to Algorithm 8, except here we unravel all the subroutines of Algorithm 8 into a single algorithm. However, we make some minor changes to make the overall description simpler, so some ingredients in this version are redundant. In particular, we use degree-3 Hadamard encodings (and either verify these using some additional checks or simply ignore the additional information) for all the proof oracles, as opposed to using degree-1 encodings for some oracles like in Algorithm 8. For example, we incorporate the checks of HadTest into two lines (Lines 13 and 14) compactly. Comparing at a more granular level, the SC subroutine serves the same purpose as it does in Algorithm 8 and the LC subroutine simulates StdLC (which is used in NewLDT and ZERO). Lines 10 to 15 correspond to the initial low-degree tests in Line 6 of Algorithm 8, Line 21 corresponds to Line 8 of Algorithm 8, and the remaining Lines 17 to 23 correspond to the zero-on-grid tests in Lines 7 and 9 of Algorithm 8. Similarly, Algorithm 10 generates proof oracles that are (almost) identical to the ones constructed in the completeness proof of Algorithm 8. In the following two algorithms, all randomness (highlighted in color red) is sampled independently from an underlying uniform distribution. Recall that $\{A \to B\}$ refers to the set of functions from A to B, $\mathcal{P}_d(m, \mathbb{F}_q)$ denotes degree-d functions over $\mathbb{F}_q^m$, and for a function $f : H^m \to \mathbb{F}_q$ with $H \subseteq \mathbb{F}_q$, $\mathrm{LDE}(f) \in \mathcal{P}_{(|H|-1)m}(m, \mathbb{F}_q)$ denotes the low-degree extension of f (see Fact 2.4). We have also implemented the following algorithms in Python, and the code (with a demo) can be found at [ABSSW26b].

Algorithm 9: The PCP Verifier

Constants: Constants $c, c_1, c_2 \in \mathbb{N}$.

Parameters: Given $n \in \mathbb{N}$, let $h := \lceil \log n \rceil$, $m := \lceil \log n / \log\log n \rceil$, $t := 2\lceil c_1 \log(hm) \rceil$, $D := c_2 hm$, and $m_1 := \lfloor D^{1/c} \rfloor + 1$. Let $q := 2^t$, $H \subseteq \mathbb{F}_q$ be of size h, $\rho : \mathbb{F}_q \to \mathbb{F}_2^t$ be an $\mathbb{F}_2$-linear bijection, $\omega, \zeta \in \mathbb{F}_q^\times$ be of order 3 and a generator respectively, and η_i's as in Fact 2.7. Let $\mathcal{F}(m')$ be $\{\mathbb{F}_q^{m'} \times \mathcal{P}_3(t, \mathbb{F}_2) \to \mathbb{F}_2\}$ and $\mathcal{F}_1(m')$ be $\{\mathbb{F}_q^{m'} \times \mathbb{F}_q^{cm_1} \times \mathcal{P}_3(t, \mathbb{F}_2) \to \mathbb{F}_2\}$.

Input: Graph $G = (V, E)$ over n vertices, where $V = H^m$.

Oracles: $\chi \in \mathcal{F}(m)$, $\chi_1 \in \mathcal{F}_1(2m)$, $\chi' \in \mathcal{F}(2m)$, $\chi'_1 \in \mathcal{F}_1(4m)$, $A_0 \in \mathcal{F}(2m)$, $A_{01} \in \mathcal{F}_1(4m)$, $B_0 \in \mathcal{F}(4m)$, $B_{01} \in \mathcal{F}_1(8m)$.

Sample $\mathbf{a}, \mathbf{b} \sim \mathbb{F}_q^m$, $\boldsymbol{\alpha} \sim \mathbb{F}_q^{2m}$, $\boldsymbol{\beta} \sim \mathbb{F}_q^{4m}$, $\mathbf{u}, \mathbf{v} \sim \mathbb{F}_q^{cm_1}$, $\lambda \sim \mathbb{F}_q^\times$, $P_i \sim \mathcal{P}_i(t, \mathbb{F}_2)$ for $0 \leqslant i \leqslant 3$, $R \sim \mathcal{P}_3(t, \mathbb{F}_2)$, $s \sim \mathbb{F}_2$, and $L \sim \mathcal{P}_1(t, \mathbb{F}_2)$ with $L(0) = 0$

For $\gamma \in \mathbb{F}_q$, interpolate $\Lambda^1_\gamma \in \mathcal{P}_1(t, \mathbb{F}_2)$ s.t. $\Lambda^1_\gamma(z) = L \circ \rho \circ (\gamma Y) \circ \rho^{-1}(z)\ \forall z \in \mathbb{F}_2^t$

Subroutines

- $\textsc{Sc}(g : \mathcal{P}_3(t, \mathbb{F}_2) \to \mathbb{F}_2,\ \ P \in \mathcal{P}_3(t, \mathbb{F}_2))$ **return** $g[P + R] - g[R]$
- $\textsc{Lc}(\theta : \mathbb{F}_q^{cm_1} \times \mathcal{P}_3(t, \mathbb{F}_2),\ \ \mathbf{u_0} \in \mathbb{F}_q^{cm_1})$ **return** $\sum_{i=1}^{c+1} \theta[\mathbf{u}_0 + \zeta^i \mathbf{v}, \Lambda^1_{\eta_i}]$

$Z_{H,\mathbf{a}} \leftarrow ((\prod_{\gamma \in H}(a_i - \gamma))_{i=1}^m)$ and $Z_{H,\mathbf{a},\mathbf{b}} \leftarrow ((\prod_{\gamma \in H}(a_i - \gamma))_{i=1}^m, (\prod_{\gamma \in H}(b_i - \gamma))_{i=1}^m)$

$\Phi_\lambda \leftarrow (1, \lambda^{m_1^i}, \ldots, \lambda^{(m_1-1)\cdot m_1^i})_{0 \leqslant i < c}$ and $\Phi_0 \leftarrow (1, 0, \ldots, 0)_{0 \leqslant i < c}$

For $\gamma \in \mathbb{F}_q$, interpolate $\Lambda_\gamma \in \mathcal{P}_3(t, \mathbb{F}_2)$ s.t. $\Lambda_\gamma(z) = L \circ \rho \circ (\gamma Y^3 - \gamma) \circ \rho^{-1}(z)\ \forall z \in \mathbb{F}_2^t$

$\theta_{A,1} \leftarrow A_{01}[(\mathbf{a}, \mathbf{0}, \boldsymbol{\alpha}), \cdot, \cdot]$ and $\theta_{A,2} \leftarrow A_{01}[(\mathbf{a}, Z_{H,\mathbf{a}}, \boldsymbol{\alpha}), \cdot, \cdot]$

$\theta_{B,1} \leftarrow B_{01}[(\mathbf{a}, \mathbf{b}, \mathbf{0}, \boldsymbol{\beta}), \cdot, \cdot]$ and $\theta_{B,2} \leftarrow B_{01}[(\mathbf{a}, \mathbf{b}, Z_{H,\mathbf{a},\mathbf{b}}, \boldsymbol{\beta}), \cdot, \cdot]$

for $(\Pi, \Pi_1) \in \{(\chi, \chi_1), (\chi', \chi'_1), (A_0, A_{01}), (B_0, B_{01})\}$ *where* $\Pi \in \mathcal{F}(m')$ *and* $\Pi_1 \in \mathcal{F}_1(2m')$ **do**

Low-Degree Tests

- Sample $(\mathbf{a}', \mathbf{b}') \sim \mathbb{F}_q^{2m'}$, let $f \leftarrow \Pi[\mathbf{a}', \cdot]$ and $f' \leftarrow \Pi_1[(\mathbf{a}', \mathbf{b}'), \cdot, \cdot]$
- Is $\textsc{Sc}(g, P_i + s) \stackrel{?}{=} g[P_i] + s$ for $0 \leqslant i \leqslant 3$ and $g \in \{f, f'[\mathbf{u}, \cdot]\}$
- Is $\textsc{Sc}(g, L \cdot P_i) \stackrel{?}{=} \textsc{Sc}(g, L) \cdot \textsc{Sc}(g, P_i)$ for $i \in \{1, 2\}$ and $g \in \{f, f'[\mathbf{u}, \cdot]\}$
- Is $f'[\mathbf{u}, \Lambda^1_{\eta_0}] + \sum_{i=1}^{c+1} f'[\mathbf{u} + \zeta^i \mathbf{v}, \Lambda^1_{\eta_i}] \stackrel{?}{=} 0$
- Is $\sum_{i=1}^{c+1} f'[\Phi_\lambda + \zeta^i \mathbf{v}, \Lambda^1_{\eta_i}] \stackrel{?}{=} \Pi[\mathbf{a}' + \lambda \mathbf{b}', L]$

3-Coloring Tests

for $\theta \in \{\theta_{A,1}, \theta_{A,2}, \theta_{B,1}, \theta_{B,2}\}$ **do**

- Is $\textsc{Sc}(\theta[\mathbf{u}, \cdot], P_i + s) \stackrel{?}{=} \theta[\mathbf{u}, P_i] + s$ for $0 \leqslant i \leqslant 1$ and $\textsc{Lc}(\theta, \mathbf{u}) \stackrel{?}{=} \theta[\mathbf{u}, L]$

Is $\textsc{Lc}(\theta_{A,1}, \Phi_\lambda) \stackrel{?}{=} A_0[(\mathbf{a}, \mathbf{0}) + \lambda\boldsymbol{\alpha}, L]$ and $\textsc{Lc}(\theta_{A,1}, \Phi_0) \stackrel{?}{=} 0$

Is $\textsc{Lc}(\theta_{A,2}, \Phi_\lambda) \stackrel{?}{=} A_0[(\mathbf{a}, Z_{H,\mathbf{a}}) + \lambda\boldsymbol{\alpha}, L]$ and $\textsc{Lc}(\theta_{A,2}, \Phi_0) \stackrel{?}{=} \textsc{Sc}(\chi[\mathbf{a}, \cdot], \Lambda_1)$

Is $\chi'[(\mathbf{a}, \mathbf{b}), L] \stackrel{?}{=} \chi[\mathbf{a}, L] - \chi[\mathbf{b}, L]$

Is $\textsc{Lc}(\theta_{B,1}, \Phi_\lambda) \stackrel{?}{=} B_0[(\mathbf{a}, \mathbf{b}, \mathbf{0}) + \lambda\boldsymbol{\beta}, L]$ and $\textsc{Lc}(\theta_{B,1}, \Phi_0) \stackrel{?}{=} 0$

Is $\textsc{Lc}(\theta_{B,2}, \Phi_\lambda) \stackrel{?}{=} B_0[(\mathbf{a}, \mathbf{b}, Z_{H,\mathbf{a},\mathbf{b}}) + \lambda\boldsymbol{\beta}, L]$ and $\textsc{Lc}(\theta_{B,2}, \Phi_0) \stackrel{?}{=} \textsc{Sc}(\chi'[(\mathbf{a}, \mathbf{b}), \cdot], \Lambda_{\mathrm{LDE}(E)(\mathbf{a},\mathbf{b})})$

if *any of the above tests return* REJECT **then return** REJECT

else return ACCEPT

Algorithm 10: Completeness Prover

Parameters: Given $n, h, m, c, c_1, c_2, t, D, m_1, q \in \mathbb{N}$, $\omega \in \mathbb{F}_q$, $\rho : \mathbb{F}_q \to \mathbb{F}_2^t$, $H \subseteq \mathbb{F}_q$, $V = H^m$, $\mathcal{F}(\cdot)$, $\mathcal{F}_1(\cdot)$ as in Algorithm 9.

Input: A 3-colorable graph $G = (V, E)$ and its proper 3-coloring $\mathsf{Color} : V \to \{1, \omega, \omega^2\}$.

Output: $\chi \in \mathcal{F}(m)$, $\chi_1 \in \mathcal{F}_1(2m)$, $\chi' \in \mathcal{F}(2m)$, $\chi'_1 \in \mathcal{F}_1(4m)$, $A_0 \in \mathcal{F}(2m)$, $A_{01} \in \mathcal{F}_1(4m)$, $B_0 \in \mathcal{F}(4m)$, $B_{01} \in \mathcal{F}_1(8m)$.

Subroutines

- **Function** $\mathcal{L}(P \in \mathcal{P}_D(k, \mathbb{F}_q))$
  - **return** $\mathcal{L}(P) \in (\mathbb{F}_q^{D+1})^{q^{2k}}$ *defined by* $\mathcal{L}(P)[\mathbf{a}, \mathbf{b}] = P(\mathbf{a} + \mathbf{b}X)$ *for* $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^k$
- **Function** $\Psi((\alpha_0, \ldots, \alpha_D) \in \mathbb{F}_q^{D+1})$
  - **for** $0 \leqslant i \leqslant D$ **do**
    - $\mathrm{rep}(i) \leftarrow (i_0, \ldots, i_{c-1})$ such that $i = \sum_{j=0}^{c-1} i_j m_1^j$ and $0 \leqslant i_j < m_1$
  - **return** $\Psi(\alpha_0, \ldots, \alpha_D) \in (\mathbb{F}_q)^{q^{cm_1}}$ *defined by* $\Psi(\alpha_0, \ldots, \alpha_D)[\mathbf{u}] = \sum_{j=0}^{D} \alpha_j \cdot u_{0,\mathrm{rep}(j)[0]} \cdots u_{c-1,\mathrm{rep}(j)[c-1]}$ *for* $\mathbf{u} \in \mathbb{F}_q^{cm_1}$
- **Function** $\Psi^*(F \in (\mathbb{F}_q^{D+1})^{q^{2k}})$
  - **return** $\Psi^*(F) \in (\mathbb{F}_q)^{q^{2k} \times q^{cm_1}}$ *defined by* $\Psi^*(F)[\mathbf{a}, \mathbf{b}, \mathbf{u}] = \Psi(F[\mathbf{a}, \mathbf{b}])[\mathbf{u}]$ *for* $(\mathbf{a}, \mathbf{b}) \in \mathbb{F}_q^{2k}$ *and* $\mathbf{u} \in \mathbb{F}_q^{cm_1}$
- **Function** `PolyDivide`$(P = P(\mathbf{X}), Z = Z(X_i))$
  - **return** *polynomials* Q, R *such that* $P = Q \cdot Z + R$ *and* $\deg_{X_i}(R) < \deg(Z)$
- **Function** `MultiDivide`$(P, Z_1, \ldots, Z_k)$
  - List $\leftarrow \varnothing$, $R \leftarrow P$
  - **for** $1 \leqslant i \leqslant k$ **do**
    - $(Q_i, R) \leftarrow$ `PolyDivide`(R, Z_i)
    - List $\leftarrow$ (List, Q_i)
  - **return** List
- **Function** $\mathrm{Had}(f \in \{\mathcal{D} \to \mathbb{F}_q\})$
  - **return** $\mathrm{Had}(f) \in \mathbb{F}_2^{|\mathcal{D}| \times |\mathcal{P}_3(t, \mathbb{F}_2)|}$ *defined by* $\mathrm{Had}(f)[\mathbf{a}, P] = P(\rho(f(\mathbf{a})))$ *for* $\mathbf{a} \in \mathcal{D}$ *and* $P \in \mathcal{P}_3(t, \mathbb{F}_2)$

$Z_H(X) \leftarrow \prod_{\gamma \in H}(X - \gamma)$

$(A^{(1)}, \ldots, A^{(m)}) \leftarrow$ `MultiDivide`$(\mathrm{LDE}(\mathsf{Color})^3 - 1, Z_H(X_1), \ldots, Z_H(X_m))$

$\widetilde{\chi} \leftarrow \mathrm{LDE}(\mathsf{Color})(\mathbf{Y_1}) - \mathrm{LDE}(\mathsf{Color})(\mathbf{Y_2})$

$(B^{(1)}, \ldots, B^{(2m)}) \leftarrow$ `MultiDivide`$(\mathrm{LDE}(E) \cdot (\widetilde{\chi}^3 - 1), Z_H(X_1), \ldots, Z_H(X_{2m}))$

return $\mathrm{Had}(\mathrm{LDE}(\mathsf{Color}))$, $\mathrm{Had}(\Psi^*(\mathcal{L}(\mathrm{LDE}(\mathsf{Color}))))$, $\mathrm{Had}(\widetilde{\chi})$, $\mathrm{Had}(\Psi^*(\mathcal{L}(\widetilde{\chi})))$, $\mathrm{Had}(\sum_{i=1}^{m} A^{(i)}(\mathbf{X}) \cdot Y_i)$, $\mathrm{Had}(\Psi^*(\mathcal{L}(\sum_{i=1}^{m} A^{(i)}(\mathbf{X}) \cdot Y_i)))$, $\mathrm{Had}(\sum_{i=1}^{2m} B^{(i)}(\mathbf{X}) \cdot Y_i)$, $\mathrm{Had}(\Psi^*(\mathcal{L}(\sum_{i=1}^{2m} B^{(i)}(\mathbf{X}) \cdot Y_i)))$

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il