Zero-Knowledge PCPs of Quasilinear Size via Locally Simulatable Sheaf Codes

Hadas Zeilberger
Yale University

Jack O'Connor
Riverlane

Tom Gur*
University of Cambridge

Nicholas Spooner
Cornell University

Abstract

We show that for every polynomial $T, b\colon \mathbb{N} \to \mathbb{N}$, there exist an $O(1)$-query probabilistically checkable proof (PCP) for $\mathsf{NTIME}(T)$ of length $\widetilde{O}(T(n) + b(n)^2)$, which is $b(n)$-query perfect zero knowledge, where n is the input size. This improves on the polynomial-length zero-knowledge PCPs of Gur, O'Connor, and Spooner (STOC 2024; STOC 2025), giving quasilinear-length, $O(1)$-query ZK-PCPs for NP.

Our construction builds on the PCPs of proximity of Ben-Sasson and Sudan (SICOMP 2008) and Dinur (JACM 2007). We prove the zero-knowledge property of our PCPs via the new machinery of locally simulatable sheaf codes, showing that sheaf codes satisfying testability and extendability conditions yield perfect zero-knowledge PCPs of proximity.

*Supported by ERC Starting Grant 101163189 and UKRI Future Leaders Fellowship MR/X023583/1.

Contents

1 Introduction **3**
- 1.1 Locally Simulatable Sheaf Codes . . . 3

2 Technical Overview **5**
- 2.1 A Perfect Zero-Knowledge PCPP for Reed-Solomon Codes . . . 5
- 2.2 Short PCPPs via code gluing . . . 9

3 Preliminaries **9**
- 3.1 Coding Theory . . . 11
- 3.2 Probabilistically Checkable Proofs . . . 13

4 Sheaf Codes **14**
- 4.1 Basic Definitions . . . 14
- 4.2 Gluing Posets . . . 16
- 4.3 Glueing Sheaf Codes . . . 18
- 4.4 The Dimensions Of Local Codes . . . 20

5 Testability of Sheaf Codes **22**
- 5.1 Definitions . . . 22
- 5.2 Recursive Testing of Sheaf Codes . . . 23
- 5.3 Claims Related To Testability Of Sheaf Codes . . . 24

6 Locally Simulatable Sheaf Codes **25**
- 6.1 Basic Definitions . . . 25
- 6.2 Extendability Implies Local Simulatability . . . 26
- 6.3 Proving κ-Extendability in Sheaf Codes . . . 28

7 Perfect Zero-Knowledge PCPPs From Sheaf Codes **31**
- 7.1 Perfect Zero-Knowledge PCPPs From d-Testable Sheaf Codes with κ-extendable covers . . . 31
- 7.2 Constructing d-testable and κ-extendable covers through glueing . . . 32
- 7.3 d-Testable Sheaf Codes with κ-Extendable Covers From Recursively Testable Codes . . . 35

8 Short, Perfect Zero-Knowledge PCPP For Reed-Solomon Codes **36**
- 8.1 A Useful Generalization Of The BS08 Bivariate Reed-Muller Code . . . 36
- 8.2 κ-Extendable Covers In Bivariate Reed-Muller Sheaf Codes . . . 37
- 8.3 Testing Reed-Solomon Codes With Bivariate Reed-Muller Sheaf Codes . . . 40
- 8.4 Our Construction . . . 41

9 Perfect Zero-Knowledge PCP For NTIME **42**
- 9.1 Linear Algebraic Constraint Satisfaction Problems . . . 43
- 9.2 From zkPCPP for Reed-Solomon Codes to zkPCP For NTIME . . . 43

References **44**

A Zero-Knowledge Gap Amplification **46**

B Deferred Proofs Of Facts From Proof of Theorem 8.3. **47**

C Proof Of Lemma 8.16 **48**

1 Introduction

A *probabilistically-checkable proof* (PCP) is a proof that can be efficiently verified by querying only a few locations chosen at random. PCPs are a central object in complexity theory, with deep connections to cryptography, coding theory and hardness of approximation. In this work we continue the study of *zero-knowledge PCPs* [KPT97] (ZKPCPs), which combine efficient verification with a strong privacy property, namely: *any* view of the PCP of bounded size should not reveal any information beyond the truth of the claim.

In recent work, Gur, O'Connor and Spooner [GOS24; GOS25] showed the existence of zero knowledge PCPs with parameters matching the original PCP theorem [AS98; ALMSS98]. That is, they showed that for any polynomial query bound $b\colon \mathbb{N} \to \mathbb{N}$, there exist $O(1)$-query PCPs NP with proof length $\mathrm{poly}(n, b(n))$, where n is the input size, such that the view of any verifier making at most $b(n)$ queries can be perfectly simulated efficiently; we refer to this property as b-query perfect zero knowledge.

Over the decades that followed the original proof of the PCP theorem, much work has been done to reduce the length of the PCP proof, while maintaining constant query complexity. The current state-of-the-art of (non-ZK) $O(1)$-query PCPs [BS08; Din07] achieves quasi-linear $\widetilde{O}(T(n))$ length for languages in $\mathsf{NTIME}(T)$. This leads to the natural question, raised in [GOS25, Open Problem 1], of whether it is possible to construct $O(1)$-query PCPs with quasilinear proof length that satisfy zero knowledge.

In this work we give a positive answer to this question by showing there exist ZKPCPs with comparable proof length to the best known non-ZK PCPs. Our main result is the following.

Theorem 1.1 (informally stated; see Theorem 9.2)**.** *For any polynomials $T, b\colon \mathbb{N} \to \mathbb{N}$, there exist $O(1)$-query ZKPCPs for $\mathrm{NTIME}(T)$ of length $\widetilde{O}\big(T(n) + b(n)^2\big)$ over a binary alphabet, which are $b(n)$-query perfect zero knowledge.*

In particular, for every query bound $b(n) \leq \sqrt{T(n)}$, our ZK-PCPs have length $\widetilde{O}(T(n))$, matching the best known non-zero-knowledge bound up to polylogarithmic factors.

Our construction is based on the short PCP of [BS08]. Underlying that construction is a PCP of proximity (PCPP) for membership in Reed–Solomon codes, a proof system which enables a prover to demonstrate that an input is close in Hamming distance to the evaluations of a low-degree univariate polynomial. For PCPPs, zero knowledge is captured by *t-local simulation*, where the view of a verifier making at most b queries to both the input and proof can be perfectly simulated using at most $t \cdot b$ queries to the input alone. Our key technical contribution is a modification of the BS08 PCPP that admits such local simulation.

Theorem 1.2 (informally stated; see Theorem 8.18)**.** *There exists a $\mathrm{polylog}(N)$-query PCPP for Reed–Solomon codes of blocklength N, with proof length $\widetilde{O}(N)$, which is $O(\sqrt{N})$-query perfect zero knowledge with a black-box $O(\sqrt{N})$-local simulator.*

To obtain Theorem 1.1, we combine the Reed–Solomon PCPP of Theorem 1.2 with algebraic reduction of [BCFGRS17], which we adapt to account for both the bounded-query guarantee and the locality of our simulator. To achieve constant query complexity, we apply Dinur's query reduction and gap amplification [Din07], showing that these transformations preserve perfect zero knowledge with only polylogarithmic losses.

1.1 Locally Simulatable Sheaf Codes

En route to obtaining our main technical contribution, we introduce and develop a general framework for constructing perfect zero-knowledge PCPs of proximity from *locally simulatable sheaf codes*, which may be of independent interest. Loosely speaking, we show that sheaf codes with *testability* and *extendability* properties admit ZK-PCPPs. Our Reed–Solomon ZK-PCPPs (Theorem 1.2) can be viewed as a special case of this general construction.

To make this more precise, recall that a Tanner code specifies a global code by imposing local code constraints on overlapping sets of coordinates. A *sheaf code*, recently defined by Panteleev and

Kalachev [PK24], is a high-dimensional generalization of Tanner codes, which consists of local linear codes $\mathcal{F}_A \subseteq \mathbb{F}^A$ on a hierarchy of overlapping domains $A \subseteq V$ such that restricting a codeword on a larger domain to a smaller one yields a codeword of the corresponding local code; that is, for any $A \subseteq B$, we have $\mathcal{F}_B|_A \subseteq \mathcal{F}_A$. We write $\mathcal{F} \subseteq \mathbb{F}^V$ for the global code.

We can construct a PCPP from a sheaf code by selecting a local domain $D \subseteq V$ whose code $\mathcal{F}_D$ is the code to be tested, such that every codeword on D extends to a global codeword, i.e., $\mathcal{F}_D = \mathcal{F}|_D$. For an input $c \in \mathcal{F}_D$, the prover samples a uniformly random global codeword that extends it; that is, it samples w uniformly from $\mathcal{F}_{D=c} := \{z \in \mathcal{F} : z|_D = c\}$. Then, it supplies the remaining coordinates, $w|_{V \setminus D}$, as the proof. The additional domains then provide the local constraints through which the verifier checks this larger word.

To obtain soundness, we require the local constraints to form a *testable cover*. Roughly speaking, this means that if the input is far from $\mathcal{F}_D$, then the local views must be far from their codes on average, regardless of the proof. More precisely, a set of local domains covering D, together with a distribution μ forms a testable cover if every purported $\widetilde{w} \in \mathbb{F}^V$ satisfies

$$\mathop{\mathbb{E}}_{A \leftarrow \mu}[\Delta(\widetilde{w}|_A, \mathcal{F}_A) = \Omega(\Delta(\widetilde{w}|_D, \mathcal{F}_D)) \ ,$$

where Δ denotes fractional distance. This local-to-global condition ensures a proximity-oblivious verifier: if the input is ϵ-far from $\mathcal{F}_D$, then sampling A and checking whether $\widetilde{w}|_A \in \mathcal{F}_A$ will lead to rejection with probability $\Omega(\epsilon)$. To reduce the query complexity, we leverage the high-dimensional structure of sheaf codes to apply the same idea recursively, "gluing" on smaller sheaf codes to test the corresponding local codes. This notion of testability allows the soundness guarantees to propagate across the hierarchy.

To obtain zero knowledge, we need to simulate a bounded view of the proof using limited access to the input. We require *extendability* for small collections of local domains even after fixing the input, i.e., any locally valid assignment agreeing with the input on the overlap must extend to a global codeword agreeing with the entire input. More precisely, if U is the union of such a collection, then for any assignment $a \in \mathbb{F}^U$ satisfying the local code constraints on the chosen domains and agreeing with c on $U \cap D$ admits a global extension $w \in \mathcal{F}$ such that $w|_U = a$ and $w|_D = c$.

Our central observation is that this property yields *local simulatability*. The simulator can answer queries by uniformly sampling local codewords consistent with its previous answers and with c on overlapping coordinates. Extendability guarantees that the sampled answers remain compatible with a global encoding in $\mathcal{F}_{D=c}$. By linearity, we can ensure that compatible local assignments have equally many global extensions, hence uniform local sampling reproduces exactly the distribution induced by a uniformly random global encoding. Thus, the analysis of zero knowledge reduces to establishing an extendability property of the sheaf code.

In Sections 5–7, we make the intuition above precise, showing that *any sheaf code with sufficient testability and extendability implies a perfect zero knowledge PCP of proximity*, where testability gives soundness, and extendability gives perfect zero knowledge. We also show how to analyse recursive composition of our PCPP via gluing of sheaf codes, and analyse how testability and extendability behave under this operation.

Concurrent work

Independently and concurrently to this work, Bafna and Vyas [BV26] obtained constructions of similar but incomparable objects, which they refer to as "private PCPs". Unlike ZKPCPs, private PCPs do not require efficient simulation; on the other hand, the privacy guarantee for private PCPs should also hold in some sense in the "no" case, whereas ZKPCPs provide a privacy guarantee only in the "yes" case.

AI disclosure

The authors used AI solely for proofreading, assistance in verification, literature review, minor stylistic editing, converting diagrams to TikZ, and converting PDF to TeX for some of the preliminary definitions.

2 Technical Overview

2.1 A Perfect Zero-Knowledge PCPP for Reed-Solomon Codes

We begin with the proof overview of Theorem 1.2, which is based on the PCPP of Ben-Sasson and Sudan [BS08], to which we shall we refer as the BS08 PCPP. We will first recall the BS08 PCPP, then present our construction, and the main ideas of its analysis.

2.1.1 The Reed–Solomon PCPP of Ben-Sasson and Sudan

In a Reed–Solomon PCPP, the prover aims to convince a verifier that a function $f : L \to \mathbb{F}$, $|L| = N$, is close in Hamming distance to a degree-d Reed–Solomon codeword. Ben-Sasson and Sudan observed that while Reed–Solomon codes do not themselves admit local tests, one can embed a Reed–Solomon codeword into a *bivariate* Reed–Muller codeword[1], which does admit a nontrivial test. Moreover, as shown in [BS08], for certain choices of embedding, a proximity guarantee on the bivariate codeword can be lifted to a guarantee about the embedded Reed–Solomon codeword.

This suggests a natural proof of proximity: the evaluation of the bivariate polynomial on an appropriate domain. The BS08 embedding is done via bivariate polynomial division in the ring $\mathbb{F}[X, Y]$: for a fixed univariate polynomial g, dividing P by $Y - g(X)$ yields the polynomial equation

$$P(X) \equiv Q'(X, Y) \cdot (Y - g(X)) + Q(X, Y).$$

In particular, substituting $Y = g(X)$ yields

$$P(X) \equiv Q(X, g(X)).$$

In other words, the codeword P can be found by evaluating the remainder polynomial Q over the curve $T(x) = (x, g(x))$.

Rules of division in the ring $\mathbb{F}[X, Y]$ imply that $d_Y(Q) = \lfloor \deg(P)/\deg(g) \rfloor$ and $d_X(Q) < \deg(g)$.

The BS08 PCPP is obtained by applying the idea above recursively, since the restriction of the bivariate evaluation to each axis-parallel line is itself a Reed–Solomon codeword. For simplicity, we begin by restricting our attention to the "top level" bivariate, and we consider a product domain $D = L \times L$, which simplifies exposition but leads to a poor soundness guarantee. Let $g(X)$ be a univariate polynomial, and define $T := \{(x, g(x)) : x \in L\}$. The PCPP operates as follows.

Prover:

- Compute Q satisfying $Q(X, g(X)) = P(X)$, where P is the polynomial that interpolates f.
- Output $\pi = Q|_{D \setminus T}$.

Verifier: The verifier has oracle access to the input f and purported proof $\tilde{\pi}$.

- Define $f' : D \to \mathbb{F}$ where $f'(x, y) = \tilde{\pi}(x, y)$ if $(x, y) \in D \setminus T$ and $f'(x, y) = f(x)$ if $(x, y) \in T$.
- The verifier chooses a random axis-parallel line R.
- It reads $f'(R)$ and checks that $f'(R)$ is close to a degree d_X-univariate polynomial (resp. a degree d_Y-univariate polynomial).

Why BS08 is not zero knowledge. Recall that our goal is to design a PCPP for Reed–Solomon codes with a simulator that is $O(\sqrt{N})$-local: that is, the simulator can simulate the view of any q-query verifier by making only $O(q\sqrt{N})$ queries to the input. Note that *deterministic* PCPPs with small query complexity cannot have such a guarantee, as combining the simulator with the verifier would yield an $\tilde{O}(\sqrt{N})$-query proximity test for Reed–Solomon codes.

[1] In this work, we use "Reed–Muller" to describe multivariate polynomial codes of bounded *individual* degree, i.e., the tensor product of Reed–Solomon codes.

We therefore need to design a *randomized* variant of the PCPP above. We must take care to achieve two competing goals. We must introduce sufficient randomness to Q to allow for local simulation, which must increase the total degree of Q compared to the non-ZK case. At the same time, we must maintain the original degree bound on the embedded univariate.

2.1.2 Our construction

We first take a different perspective on BS08, based on interpolation rather than polynomial division. Recall that BS08 embeds the input on the curve $T(x) = (x, g(x))$ for a univariate polynomial g. The specific polynomial g they choose is of degree $\sqrt{N}$ and has a special structure: is $\sqrt{N}$-to-1 on L, and therefore each column of D intersects T has at exactly $\sqrt{N}$ points. Therefore, since $d_X = \sqrt{N} - 1$, the univariate polynomial embedded in each column restriction is uniquely determined by the $\sqrt{N}$ points of the input that intersects with that column. We can therefore use the following alternative procedure to construct the BS08 embedding polynomial Q:

- Choose any set $S \subseteq D$ consisting of $d_Y + 1$ columns.
- For each column $K \subseteq S$ interpolate the unique degree-d_X univariate polynomial on K from the evaluations of f that are embedded in the intersection $T \cap K$.
- Interpolate the bivariate polynomial Q from S.

The steps in the above algorithm are based on well-known properties of univariate and bivariate polynomials. The second one uses the fact that a degree-d univariate polynomial can be interpolated from any $d + 1$ field elements and the third uses the well-known fact (and proven in Theorem H.1 in [BCFGRS17]) that a bivariate polynomial can be interpolated from any $d_Y + 1$ column polynomials. Thus the output of that algorithm, Q, is well-defined.

By a simple degree counting argument, $\deg(Q \circ T) = d_X + \sqrt{N} \cdot d_Y$. Moreover, $Q \circ T$ agrees with input P on $\sqrt{N}(d_Y + 1) = \deg(Q \circ T) + 1$ points. Thus if $\deg(P) = d_X + \sqrt{N} \cdot d_Y$, $Q \circ T \equiv P$. Choosing $d_X = \sqrt{N} - 1$ and $d_Y = \rho\sqrt{N} - 1$ yields $\deg(Q \circ T) = \rho N - 1$, giving a test for polynomials of degree $\rho N - 1$.

Now suppose that we modify the above algorithm to allow for a larger value of d_X. The degree-d_X extension of $T \cap K$ is then no longer unique, so we choose one at random. Note that in order to maintain that $\deg(Q \circ T) = \rho N - 1$, we have to compensate a larger d_X by taking a smaller d_Y. For our purposes it suffices to choose $d_X = 2\sqrt{N} - 1$ and $d_Y = \rho\sqrt{N} - 2$. Unfortunately, this reparametrization leads to a new issue: the number of agreement points between $Q \circ T$ and P drops to $\rho N - \sqrt{N}$, which is less than $\deg(Q \circ T)$. We can therefore no longer conclude, via our prior reasoning, that $Q \circ T \equiv P$. In the next section we show how to resolve this issue, and establish the necessary zero knowledge property, using the framework of *extendability*.

2.1.3 Completeness and zero knowledge

Before diving into our proof, we connect BS08 and our new construction to sheaf codes and extendability, which simplifies the analysis of both completeness and zero-knowledge. A bivariate Reed–Muller code $\mathcal{F}$ of degree (d_X, d_Y) is a tensor code, and is therefore fully characterized by local codes $\mathcal{F}_\sigma$ on each row (resp. column) σ, where $\mathcal{F}_\sigma$ is a polynomial of degree d_Y (resp. d_X). Yet this is not the only local code structure available: the algebraic structure of bivariate polynomials is richer than that of general tensor codes. We include one additional local code: for the set T defined above, we set $\mathcal{F}_T$ to be a degree-$(\rho N - 1)$ Reed–Solomon code. By the above degree calculation, this does not affect the code $\mathcal{F}$ itself, but the inclusion of $\mathcal{F}_T$ will be crucial in our proofs of completeness and zero knowledge.

Showing zero knowledge of our construction amounts to proving the following extendability property:

Lemma 2.1 (informal)**.** *Let $S \subseteq D$ be a set that is the union of at most $\rho\sqrt{N} - 2$ rows and columns.*[2]

[2]Looking ahead, showing extendability for collections of rows and columns rather than for sets of individual points will be crucial for showing zero knowledge of the full recursive construction.

For any function $g\colon S \cup T \to \mathbb{F}$ whose restriction to each full row or column in S and to the set T belongs to the corresponding local code, there is a codeword $h \in \mathcal{F}$ such that $h|_{S\cup T} = f$.

Note that completeness is equivalent to the special case $S = \emptyset$. Before we show how we prove this property, we briefly explain how to use it to show local simulation of the top-level proof. Given a desired query set $V \subseteq D$ of size at most $\rho\sqrt{N}$, the simulator will first choose an arbitrary set S of rows and columns such that $V \subseteq S$. It queries the input f on the set $A = \{x : T(x) \in S\}$; by the structure of T, $|A| \leq \sqrt{N} \cdot |V|$. It then samples a function $g'\colon S \to \mathbb{F}$ whose restriction to each full row or column belongs to the corresponding local code and which is consistent with f, and outputs $g'|_V$. By the lemma, the function $g\colon S \cup T \to \mathbb{F}$, given by extending g' so that $g(x, y) = f(x)$ for $(x, y) \in T$, is extendable to a global codeword $h \in \mathcal{F}$, which implies that $g'|_V$ is correctly distributed.

Suppose that S consists of m rows and n columns, $m + n \leq \rho\sqrt{N} - 2$. We denote the domain $S \cup T$ by D_1; our goal is to prove that D_1 is extendable. We will achieve this by constructing domains D_2, D_3, D_4 such that D_4 is a set of rows of size $d_X + 1$, which is extendable by standard properties of polynomial interpolation, and such that extendability of D_{i+1} implies extendability of D_i for all $i \in [3]$. These domains are depicted in Figure 1.

We start with the reduction from D_2 to D_1. D_2 consists of $d_X + 1$ rows, d_Y columns, and the set T, chosen such that $D_1 \subseteq D_2$. It suffices to show that any codeword satisfying the local Reed–Solomon codes in D_1 can be extended to a local codeword on D_2; this implies that D_1 is extendable by a simple transitivity argument. Extendability of D_1 to D_2 follows because each column in D_1 is either full or contains at most $(1 + \rho)\sqrt{N}$ points, which is smaller than $d_X + 1 = 2\sqrt{N}$, and so it is possible to interpolate a polynomial of degree d_X agreeing with any column. This is where the reparametrization is crucial: in the original formulation of BS08, $d_X = \sqrt{N} - 1$, and so extendability of D_1 to D_2 fails. A similar inequality holds for the rows, noting that every row intersects with T in at most one point.

The remaining steps require more care, as $D_2 \supset D_3 \supset D_4$, which means we cannot appeal to transitivity. Instead, we show that many points in D_2 (resp. D_3) are redundant: every codeword that is consistent with the local codes in D_3 (resp. D_4) has a *unique* extension to D_2 (resp. D_3). Removing redundant points does not affect extendability; we refer to this property later as the "triangle of extendability", as it arises from a triangular commutative diagram.

In D_2, the entries of T that lie outside of the specified rows and columns are redundant: each row intersects T at exactly one point, and each column intersects T at $\sqrt{N}$ points, thus the size of the intersection is $d_X + \sqrt{N}d_Y + 1 = \deg(Q \circ T) + 1$, provided no row and column intersect on T (we construct a domain for which this is possible in the following paragraph). In D_3, elements on the columns that lie outside of the rows are uniquely determined by the rows, since there are $d_X + 1$ many.

2.1.4 Soundness

So far we have only discussed completeness and zero-knowledge. We now turn our attention to soundness. As mentioned earlier, the simplified version of BS08 from Section 2.1.1 does not satisfy soundness. The reason for this is that we set $D = L \times L$, and as a result $|D|/|T| = \Theta(N)$ is not a constant. This means that even if a function $f : D \to \mathbb{F}$ has small relative Hamming distance from a Reed-Muller codeword, the relative distance of $f|_T$ can be much larger. To fix this, BS08 constructs a careful puncturing of the Reed-Muller code. They start with a product set, as required for the Polishchuck-Spielman bivariate local test [PS94]. This product set only contains $16\rho\sqrt{N}$ elements of T which is far lower than the degree of $Q \circ T$. To fix this, for each column $K \subseteq L \times L$, they adjoin to the domain the $\sqrt{N}$ points in $K \cap T$. In the end, the domain is supported on $\sqrt{N}$ columns, with each column containing $O(\sqrt{N})$ points; thus T makes up a constant fraction of this new domain.

The domain we choose for our construction departs from that of BS08 in two key ways. First, we make adjustments to accommodate the new degree bounds of our setting. Second, as previously mentioned, our extendability proof does not go through for a domain containing a row and column that intersect at a point in T. We observe that, since the part of the product set that intersects T is tiny, we can simply remove those columns from the proof while maintaining soundness of the test.

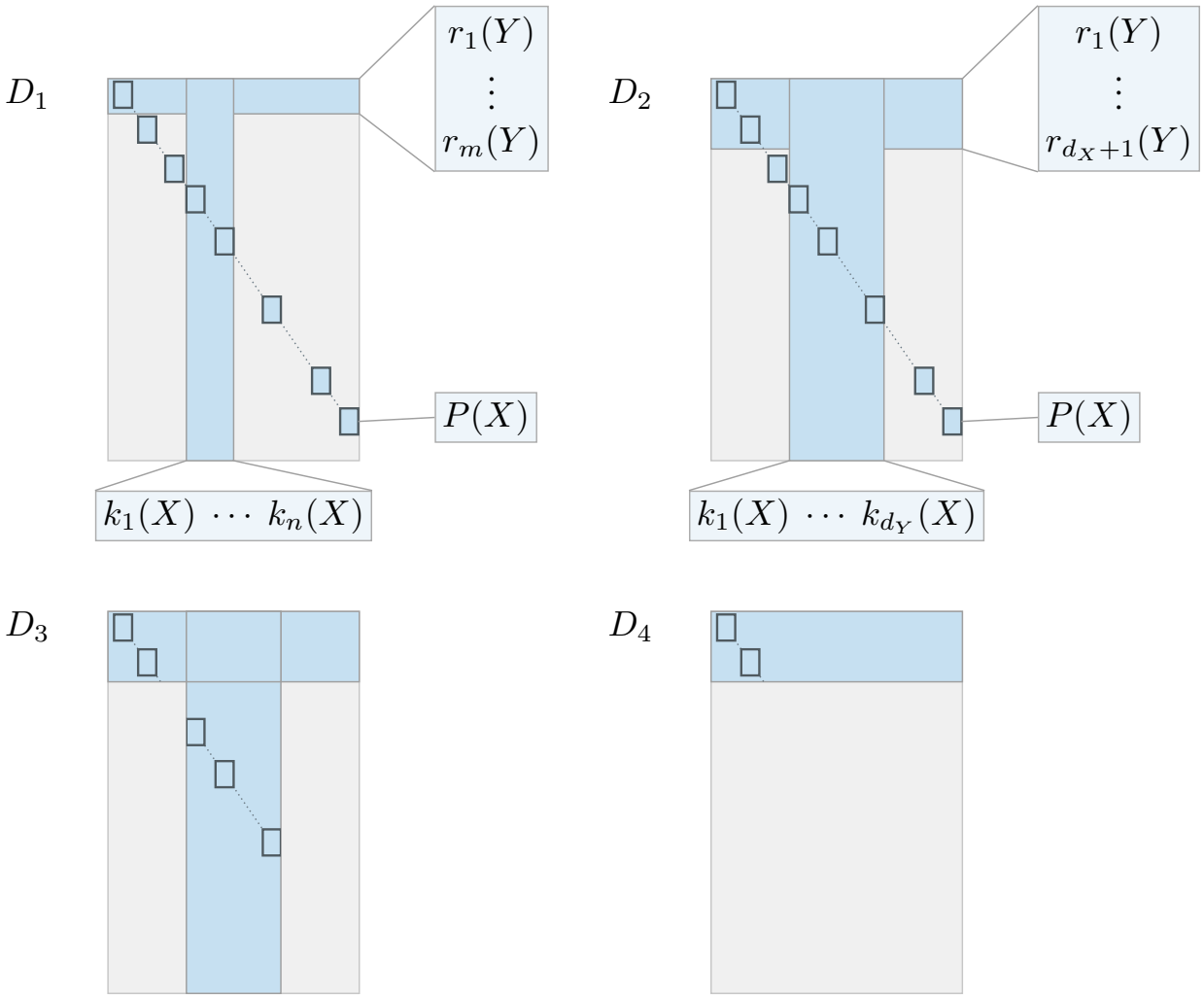


Figure 1: Our goal is to prove extendability of the domain D_1. We do this via a series of reductions, showing that for each $i \in [3]$, D_i inherits the extendability of D_{i+1}. D_4 is extendable by standard bivariate interpolation. See Section 2.1.3 for details.

2.2 Short PCPPs via code gluing

An important feature of BS08 is that the verifier can check the proof with only a polylogarithmic number of queries (which is later made constant via gap amplification). This was not true for the simplified version of BS08 we described in Section 2.1.1 because the verifier queried entire rows and columns, each of length $\sqrt{N}$. In the full PCPP, the prover sends a separate BS08 "subproof" for each row and column polynomial, and the PCPP verifier invokes the subproof verifier instead of reading the entire row or column. The proof is constructed recursively in this way until the local codewords are of constant size.

Sheaf code gluing. We express this construction in the language of sheaf codes as follows. The "top level" of the proof is a sheaf code containing a local code to be tested (a degree $\rho N - 1$ Reed-Solomon code in the case of BS08). The test is performed by checking a random local code from a prescribed set of local codes that cover the domain; we refer to such a set as a *testable cover*.

To derive the recursive construction, for each local code $\mathcal{F}_\sigma$ in the cover we specify another sheaf code that contains $\mathcal{F}_\sigma$ as a local code, and has a testable cover. We then "glue" the two codes together at σ via a quotient construction inspired by the adjunction of topological spaces; see Sections 4.2 and 4.3 for details. Applying this operation recursively yields a PCPP with small query complexity.

Zero knowledge of the full recursive construction. So far, we have only discussed zero knowledge for the top-level bivariate polynomial (or sheaf code in the general setting). Observe, however, that for each element of the testable cover in the top level, the full tree of subproofs attached to that element can be efficiently generated in full from the local restriction alone. We can thus simulate as follows: for a query to a location v, determine the top-level local code σ of which v is a descendant, query the top-level simulator on σ, and compute the full tree of subproofs rooted at σ.

Recursively testable code families. Finally, we isolate the property of Reed-Solomon codes that enables a recursive PCPP like the one described above, i.e., where it is possible to find a testable sheaf code for each local code. We refer to this property as "recursive testability". We say that a code is recursively testable if there exists some sheaf code that has a testable cover for it and every code in that cover is also in the family. From that perspective, we can view [BS08] as a proof that Reed–Solomon codes are a recursively testable family. We consider the separation of algebraic and sheaf-theoretic components of the BS08 PCPP as a significant contribution of this work, and we hope that this analysis paves the way to PCPP constructions for other codes.

Organization of the Paper

The rest of the paper is organized as follows. In Section 4, we lay the ground work by defining concepts related to sheaf codes, and introduce our gluing operation. Next, in Section 5, we describe the properties needed for a sheaf code to be testable, and construct a simple codeword test. In Section 6, we show that sheaf codes satisfying certain extendability properties are locally simulatable and we present a toolbox for proving extendability. In Section 7, we combine the results from Sections 5 and 6 to build perfect zero-knowledge PCPPs from sheaf codes satisfying both testability and extendability. We also show how to build testable and extendable sheaf codes from "recursively testable code families", which we define. Next, we show that Reed-Solomon codes are a recursively testable code family and use them to instantiate the generic PZK-PCPP from sheaf codes. We conclude the paper with Section 9, where we build our PZK-PCP using techniques from [BCFGRS17].

3 Preliminaries

Notation. $\mathbb{F}$ is a finite field. For $d \in \mathbb{N}$, denote by $\mathbb{F}^{\leq d}[X]$ the set of univariate polynomials with coefficients in $\mathbb{F}$ and degree at most d. For $d_X, d_Y \in \mathbb{N}$, denote by $\mathbb{F}^{\leq [d_X, d_Y]}[X, Y]$ the set of bivariate polynomials with coefficients in $\mathbb{F}$, X-degree at most d_X and Y-degree at most d_Y. For $p : \mathbb{F} \to \mathbb{F}$ and $D \subset \mathbb{F}$, denote by $p(D) = \{p(x) : x \in D\}$ and for $Q : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ and $D \subseteq \mathbb{F} \times \mathbb{F}$, denote by $Q(D) = \{Q(x, y) : (x, y) \in D\}$. For all $X, Y \subset \mathbb{F}$, $D \subset X \times Y$, denote by D_X the set

$\{x \in \mathbb{F} : \exists y \in \mathbb{F} \mid (x,y) \in D\}$ and by D_Y the set $\{y \in \mathbb{F} : \exists x \in \mathbb{F} \mid (x,y) \in D\}$. For $n \in \mathbb{N}$, denote by $[n]$ the interval between 1 and n inclusive.

We will need the following basic set theoretic and linear algebraic claims.

Claim 3.1. *Let $n, m \in \mathbb{N}$ and let $V_1, .., V_n, W_1, .., W_m$ be sets such that for all $i \in [n]$, $j \in [m]$, $V_i \cap W_j = \varnothing$. Let $V = \bigcup_{i\in[n]} V_i$ and $W = \bigcup_{i\in[m]} W_i$. Then,*

$$V \cap W = \varnothing.$$

Proof.

$$V \cap W = (\bigcup_{i\in[n]} V_i) \cap (\bigcup_{j\in[m]} W_j) = \bigcup_{i\in[n]} V_i \cap (\bigcup_{j\in[m]} W_j) = \bigcup_{i\in[n]} \bigcup_{j\in[m]} V_i \cap W_j = \bigcup_{i\in[n]} \bigcup_{j\in[m]} \varnothing.$$

□

Claim 3.2. *Let D_1, D_2 be two domains, and let $g : D_1 \to D_2$ be a bijection. Let $X, Y \subseteq D_1$ such that $X \subseteq Y$. Then $g(X) \subseteq g(Y)$.*

Proof. Let $v \in g(X)$. Then there exists $w \in X$ such that $v = g(w)$. Since $X \subseteq Y$, $w \in Y$. Therefore $v = g(w) \in g(Y)$. □

For set D, we view a vector as a function $v : D \to \mathbb{F}$. For sets D_1, D_2, we view a matrix, M, as a function $M : D_1 \times D_2 \to \mathbb{F}$. For $n \in \mathbb{N}$, we denote by Id_n the $n \times n$ identity matrix.

Definition 3.3 (Function Restriction)**.** *Let D be a set, let U be a subset of D, and let $w : D \to \mathbb{F}$ be a function from D to $\mathbb{F}$. We denote by $w|_U$ the restriction of w to U, that is the function $w|_U : U \to \mathbb{F}$ such that $w_U(x) = w(x)$ for all $x \in U$.*

Definition 3.4 (Vector Space Restriction)**.** *Let D be a set and let $V \subseteq \mathbb{F}^D$ be a vector space. Let $D' \subseteq D$. Then the restriction of V to D', denoted by $V|'_D$ is the following vector space.*

$$V|_{D'} = \{v|_{D'} : v \in V\}.$$

Definition 3.5 (Support Of A Vector)**.** *Let D be a finite set and let $V \subseteq \mathbb{F}^D$ be a vector space. Then the support of a vector $v \in V$, denote by* $\mathsf{supp}(v)$*, is the set $\{x \in D : v[x] \neq 0\}$.*

Claim 3.6. *Let V_1, V_2 be two vector spaces and let $V = V_1 + V_2$. Then*

$$\dim(V) = \dim(V_1) + \dim(V_2) - \dim(V_1 \cap V_2).$$

Furthermore if $V_1 \cap V_2 = \{\vec{0}\}$, then $\dim(V) = \dim(V_1) + \dim(V_2)$.

Permutation of a vector space.

Definition 3.7 (Permutation Mapping)**.** *Let D, E be two domains, and let $g : D \to E$ be a bijection between them. Then g induces a natural mapping of vector spaces: for any subspace V of $\mathbb{F}^D$,*

$$g(V) = \{w \in \mathbb{F}^E \mid \exists v \in V : w(g(x)) = v(x) \quad \forall x \in D\}$$

is a subspace of $\mathbb{F}^E$. For each $v \in V$, we denote by $g(v)$ the vector in $g(V)$ satisfying $g(v)(g(x)) = v(x)$ for all $x \in D$. Similarly for each $w \in g(V)$, we denote by $g^{-1}(w)$ the vector in V such that $g^{-1}(w)(x) = w(g(x))$ for all $x \in D$.

Proposition 3.8. *Let D, E be domains. Let $V \subseteq \mathbb{F}^D$ be a vector space and let $g : D \to E$ be a bijection. Then for all $v \in V$ and $S \subseteq D$,*

$$g(v)|_{g(S)} = g|_S(v|_S) \tag{1}$$

$$g(V)|_{g(S)} = g|_S(V|_S) \tag{2}$$

Going forward, we denote $g|_S(V|_S)$ by $g(V|_S)$ and $g|_S(v|_S)$ by $g(v|_S)$.

Proof. To prove (1), for all $x \in S$,

$$g(v)|_{g(S)}(g(x)) = g(v)(g(x)) = v(x) \text{ and} \qquad g|_S(v|_S)(g(x)) = v|_S(x) = v(x)$$

To prove (2), we have the following.

$$g(V)|_{g(S)} = \{g(v)|_{g(S)} : v \in V\} = \{g|_S(v|_S) : v \in V\} = \{g|_S(w) : w \in V|_S\} = g|_S(V|_S) \quad \square$$

Padded vector spaces. Let D be a finite set, let $D' \subseteq D$ and let $V \subseteq \mathbb{F}^{D'}$.

Definition 3.9 (Zero-Padding of a Vector Space)**.** *A zero-padding of $v \in \mathbb{F}^{D'}$ with respect to D, denoted by $\overline{v}_D$, is the vector in $\mathbb{F}^D$ such that*

$$\overline{v}_D|_{D'} = v \text{ and } \overline{v}_D|_{D \setminus D'} = \vec{0}.$$

A zero padded vector space, denoted by $\overline{V}_D \subseteq \mathbb{F}^D$, is the following subspace.

$$\overline{V}_D = \{\overline{v}_D : v \in V\} = V \times \{\vec{0}_{D \setminus D'}\}.$$

Claim 3.10. *The function $f : V \to \overline{V}_D$ defined by $f(v) = \overline{v}_D$ is a bijection and its inverse is the restriction map $g : \overline{V}_D \to V$ defined by $g(w) = w|_{D'}$.*

Proof. By definition of f and g, for all $v \in V$, $g(f(v)) = \overline{v}_D|_{D'} = v$ and for all $w \in \overline{V}_D$, $f(g(w)) = f(w|_{D'}) = \overline{(w|_{D'})}_D = w$. Therefore f and g are inverses of each other and so f is a bijection. $\square$

Corollary 3.11.

$$\dim(\overline{V}_D) = \dim(V).$$

Proof. By Claim 3.10, $|\overline{V}_D| = |V|$, and therefore $\dim(\overline{V}_D) = \log_{|\mathbb{F}|}(|\overline{V}_D|) = \log_{|\mathbb{F}|}(|V|) = \dim(V)$. $\square$

3.1 Coding Theory

Let Σ an alphabet and let $\ell \in \mathbb{N}$. A code C is a subset $C \subseteq \Sigma^\ell$. Given two strings $x, y \in \Sigma^n$, we denote the Hamming distance between x and y by $\Delta(x, y) := |\{i \in [n] : x_i \neq y_i\}|/n$. The minimum distance of a code, which we denote by Δ_C, $\min_{x,y \in C, x \neq y}(\Delta(x, y))/n$. When clear from context, we will sometimes abuse notation and refer instead to the absolute distance and absolute minimum distance as Δ and Δ_C respectively. If $\Sigma = \mathbb{F}$ and C is a linear subspace of $\mathbb{F}^l$ then C is a linear code.

Definition 3.12 (Code family)**.** *Let $\{n_i\}_{i \geq 1}$ be an increasing sequence of block lengths, and suppose there exist sequences $\{k_i\}_{i \geq 1}$ and $\{d_i\}_{i \geq 1}$ such that, for every $i \geq 1$, there exists a code $C_i \subseteq \Sigma^{n_i}$ with distance d_i such that $|C| = k_i$. Then the sequence*

$$\mathcal{C} = \{C_i\}_{i \geq 1}$$

is a family of codes.

Definition 3.13 (Dual Code)**.** *Let D be a finite set and let $C \subseteq \mathbb{F}^D$ be a linear code. Then the dual code, denoted by $C^\perp$ is equal to the following.*

$$C^\perp = \{v \in \mathbb{F}^D : \langle v, c \rangle = 0 \quad \forall c \in C\}.$$

Claim 3.14. *Let C_1, C_2 be two linear codes.*

$$(C_1 + C_2)^\perp = C_1^\perp \cap C_2^\perp.$$

Claim 3.15. *Let D be a finite set and let C be a linear code in $\mathbb{F}^D$. Then,*

$$\dim(C) = |D| - \dim(C^\perp).$$

Claim 3.16. *Let D be a finite set, let $D' \subseteq D$ and let $C \subseteq \mathbb{F}^{D'}$ be a linear code. Denote by $\widehat{C}_D$ the space $(\overline{C^\perp}_D)^\perp$, i.e the dual code of the padded vector of the dual code of C. Then,*

$$\widehat{C}_D = \{c \in \mathbb{F}^D \mid c|_{D'} \in C\} = C \times \mathbb{F}^{D \setminus D'}$$

Proof. First we show that $\widehat{C}_D \subseteq \{c \in \mathbb{F}^D \mid c|_{D'} \in C\}$. If $v \in \widehat{C}_D$, then by Definition 3.13, $\langle v, c\rangle = 0$ for all $c \in \overline{C^\perp}_D$. Since $\mathsf{supp}(\overline{C^\perp}_D) \subseteq D'$, this implies that $\langle v|_{D'}, c|_{D'}\rangle = 0$ for all $c \in \overline{C^\perp}_D$, which implies that $\langle v|_{D'}, c'\rangle = 0$ for all $c' \in \overline{C^\perp}_D|_{D'} = C^\perp$, where the equality follows by Claim 3.10. Therefore $v|_{D'} \in (C^\perp)^\perp$. Next, we show that $\{c \in \mathbb{F}^D \mid c|_{D'} \in C\} \subseteq \widehat{C}_D$. Suppose that $v \in \{c \in \mathbb{F}^D \mid c|_{D'} \in C\}$. Then for all $w \in \overline{C^\perp}_D$, $\langle v, w\rangle = \langle v|_{D'}, w|_{D'}\rangle = 0$, where the first equality follows from the fact that $\mathsf{supp}(\overline{C^\perp}_D) \subseteq D'$, and the second equality follows because, by assumption, $v|_{D'} \in C$ and by definition of padded vector space (Definition 3.9), $w|_{D'} \in C^\perp$. Therefore, by definition of dual code (Definition 3.13), $v \in \widehat{C}_D$. □

3.1.1 Maximum Distance Separable Codes

Definition 3.17 (Maximum Distance Separable Code)**.** *Let D be a finite set and let $C \subseteq \mathbb{F}^D$ be a linear code. Then C is maximum distance separable (MDS) if*

$$\Delta_C = |D| - \dim(C) + 1.$$

Lemma 3.18 (Small Restrictions of MDS Codes Have Full Rank)**.** *Let D be a finite set and let $C \subseteq \mathbb{F}^D$ be a linear, MDS, code (Definition 3.17). Then, for all $D' \subseteq D$ such that $|D'| \leq \dim(C)$, then*

$$\dim(C|_{D'}) = |D'|$$

Proof. First, suppose that $|D'| = \dim(C)$. Consider the map $f : C \to C|_{D'}$. By definition of vector space restriction (Definition 3.4), f is surjective. Next we prove that it's injective. Let $c_1, c_2 \in C$ such that $c_1 \neq c_2$. Suppose by contradiction that $f(c_1) = f(c_2)$. Then $f(c_1) - f(c_2) = 0$. By definition of vector space restriction (Definition 3.4), $f(c_1) - f(c_2) = c_1|_{D'} - c_2|_{D'} = (c_1 - c_2)|_{D'} = 0$. But since C is MDS, any non-zero codeword in C has at most $\dim(C) - 1$ zeroes. By assumption $c_1 - c_2 \neq 0$, but $|D'| = \dim(C)$, which implies that $c_1 - c_2$ has $\dim(C)$ zeroes, which is a contradiction. Therefore f is injective, which implies it is bijective and so $\dim(C|_{D'}) = \dim(C) = |D'|$. Next, let $E \subseteq D$ such that $|E| < \dim(C)$. Then there exists $D' \subseteq D$ such that $E \subseteq D'$ and $|D'| = \dim(C)$. Since $C|_{D'} \subseteq \mathbb{F}^{D'}$ and their dimensions are equal, it follows that $C|_{D'} = \mathbb{F}^{D'}$. By definition of restriction map, $C|_E = C|_{D'}|_E = \mathbb{F}^{D'}|_E = \mathbb{F}^E$, which completes the proof. □

Lemma 3.19 (Support of Dual Vectors of MDS Code Is Larger Than Dimension)**.** *Let D be a finite set and let $C \subseteq \mathbb{F}^D$ be an MDS code. Then for all non-zero $v \in C^\perp$, $|\mathsf{supp}(v)| > \dim(C)$.*

Proof. Suppose that $|\mathsf{supp}(v)| \leq \dim(C)$. Then there exists $D' \subseteq D$, $|D'| \leq \dim(C)$ such that $\mathsf{supp}(v) = D'$. But this implies that for all $c \in C$, $\langle v, c\rangle = \langle v|_{D'}, c|_{D'}\rangle = 0$. In particular, this implies that $C|_{D'}^\perp$ contains $v|_{D'} \neq \vec{0}$ and is therefore nonzero. But this is a contradiction because by Lemma 3.18, since $|D'| \leq \dim(C)$, it follows that $\dim(C|_{D'}) = |D'|$ which implies that $\dim((C|_{D'})^\perp) = 0$. □

3.1.2 Polynomial Codes

Definition 3.20 (Reed-Muller Code)**.** *Given a field $\mathbb{F}$, a positive integer m, a degree vector $\vec{d} = (d_1, \ldots, d_m) \in \mathbb{N}^m$, and $D \subset \mathbb{F}^m$, we denote by $RM[\mathbb{F}, D, \vec{d}]$ the following vector space.*

$$RM[\mathbb{F}, D, \vec{d}] = \{Q|_D : Q \in \mathbb{F}^{\leq \vec{d}}[X_1, .., X_m]\}.$$

In this section, we focus on bivariate Reed-Muller codes, i.e. the code $RM[\mathbb{F}, D, (d_X, d_Y)]$ for $d_X, d_Y \in \mathbb{N}$. The local codes of bivariate Reed-Muller codes are Reed-Solomon codes, which we define next.

Definition 3.21 (Reed-Solomon Code)**.** *Given a field $\mathbb{F}$, a domain $D \subseteq \mathbb{F}$ a positive integer d, we denote by $RS[\mathbb{F}, D, d]$ the following vector space.*

$$RS[\mathbb{F}, D, d] = \{p|_D : p \in \mathbb{F}^{\leq d}[X]\}$$

i.e. the code of evaluations of univariate polynomials over D with degree less than or equal to d.

It is well-known that Reed-Solomon codes are maximum-distance separable.

Claim 3.22. *Let $D \subseteq \mathbb{F}$, let d be a positive integer smaller than $|D|$. Then $C = RS[\mathbb{F}, D, d]$ is maximum distance separable.*

Proof. By the Schwartz-Zippel Lemma, a univariate polynomial of degree at most d has at most d zeroes. Therefore, it's minimum Hamming weight is $|D| - d$, and since C is a linear code, this implies $\Delta_C = |D| - d$. Furthermore, $\dim(C) = d + 1$. Therefore $\Delta_C \geq |D| - d = |D| - (\dim(C) - 1) = |D| - \dim(C) + 1$, which completes the proof. □

3.1.3 Locally Testable Codes

Definition 3.23 (Codeword tests)**.** *Let $C \colon \{0,1\}^k \to \{0,1\}^n$ be a code (of distance d), and let $q \in \mathbb{N}$ and $\delta \in (0,1)$. A q-local δ-tester for C is a probabilistic oracle machine M that makes at most q (non-adaptive) queries and satisfies the following two conditions:*

- *Accepts codewords (completeness): For any $x \in \{0,1\}^k$, given oracle access to $C(x)$, machine T accepts with probability 1. That is, $\Pr[T^{C(x)}(1^k) = 1] = 1$, for any $x \in \{0,1\}^k$.*
- *Rejects words far from C: For any $w \in \{0,1\}^n$ that is δ-far from C, given oracle access to w, machine T rejects with probability at least $1/2$. That is, $\Pr[M^w(1^k) = 1] \leq 1/2$, for any $w \in \{0,1\}^n$ that is δ-far from C.*

We call q the query complexity of M, and δ the proximity parameter.

3.2 Probabilistically Checkable Proofs

Definition 3.24 (PCP)**.** *A probabilistically checkable proof (PCP) for a relation R consists of a prover P and a PPT verifier V such that the following holds.*

1. Completeness. For every $(x, w) \in R$,

$$\Pr_{\pi \leftarrow P(x,w)}[V^\pi(x) = 1] = 1.$$

2. Soundness. For every $x \notin L(R)$ and every oracle π^*,

$$\Pr[V^{\pi^*}(x) = 1] \leq \frac{1}{2}.$$

We say that a PCP is efficient if $P(x, w)$ can be computed efficiently for any $(x, w) \in R$ and we say that it has query complexity q if the number of queries the verifier makes is bounded by q.

Definition 3.25 (PCPP)**.** *For $\delta : \mathbb{N} \to [0, 1]$, a probabilistically checkable proof of proximity (PCPP) for a pair language L with proximity parameter δ consists of a prover P and a verifier V such that the following holds for every pair of strings (x, y).*

1. Completeness. If $(x, y) \in L$,

$$\Pr_{\pi \leftarrow P(x,y)}[V^{(y,\pi)}(x) = 1] = 1.$$

2. Soundness. If y is δ-far from the set $L[x] := \{y : (x, y) \in L\}$, then for every oracle π^*,

$$\Pr[V^{(y,\pi^*)}(x) = 1] \leq \frac{1}{2}.$$

We say that a PCPP has query complexity q if the number of queries the verifier makes is bounded by q.

Zero-Knowledge PCPs.

Definition 3.26 (View of PCP of proximity)**.** *For a PCP of proximity (P, V) and a (possibly malicious) verifier V^*, $\mathsf{View}_{V^*,P}(x, y)$ denotes the view of V^* with explicit input x and oracle access to both implicit input y and $\pi \leftarrow P(x)$. That is, $\mathsf{View}_{V^*,P}(x, y)$ comprises V^*'s random coins and all answers to V^*'s queries to (y, π).*

Definition 3.27 (Black-box l-local perfect zero-knowledge PCP of proximity)**.** *Let $l : \mathbb{N} \to \mathbb{N}$. We say that a PCP of proximity system (P, V) for a pair language L is black-box l-local, perfect zero knowledge with query bound q^* if there exists an algorithm Sim, such that for every (possibly malicious) adaptive q^*-query-bounded verifier V^*, and for every $(x, y) \in L$, $\mathsf{Sim}^{(V^*,y)}(x)$ is distributed identically to $\mathsf{View}_{V^*,P}(x, y)$, and moreover, Sim runs in time $\mathrm{poly}(|x|, q^*)$ and makes at most $q^* \cdot l(n)$ queries to y.*

Definition 3.28 (Black-box perfect zero-knowledge PCP)**.** *We say that a PCP system (P, V) for a relation R is* black-box perfect zero knowledge *with query bound q^* if there exists a PPT algorithm Sim, called the* simulator*, such that for every possibly malicious, adaptive, q^*-query-bounded verifier V^*, and for every $(x, w) \in R$,*

$$\mathsf{Sim}^{V^*}(x) \equiv \mathrm{View}_{V^*,P}(x, w),$$

where $\equiv$ denotes that the two random variables are distributed identically.

4 Sheaf Codes

In this section, we present results and definitions related to sheaf codes [PK24], which sets the foundation for the rest of the paper.

4.1 Basic Definitions

We start with some basic definitions.

Definition 4.1 (Poset)**.** *A poset is a set P equipped with a partial order $\leqslant$.*

Definition 4.2 (Upper Sets of a Poset)**.** *Let P be a poset. For $\sigma \in P$, denote by $P_{\geqslant\sigma}$ the set $\{x \in P \mid x \geqslant \sigma\}$. $U \subseteq P$ is an upper set if for all $\sigma \in U$, $P_{\geqslant\sigma} \subseteq U$; i.e., for all $\sigma \in U$ and $x \in P$ such that $x \geq \sigma$, $x \in U$. Going forward, we refer to $P_{\geqslant\sigma}$ as a primitive upper set.*

Fact 4.3. Let P be a poset and let $U, V \subseteq P$ be upper sets. Let $x \in U \setminus V$ and $y \in V \setminus U$. Then x and y are unordered.

Proof. Suppose otherwise. Then either $x \leqslant y$ or $y \leqslant x$. If the former, it follows by definition of upper set that $y \in U$, which is a contradiction. If the latter, it follows by definition of upper set that $x \in V$, which is also a contradiction and completes the proof. □

Definition 4.4 (Maximal Elements)**.** *The set of maximal elements, denoted by P_*, is the set of elements in P that are either greater than or unordered with every other element in P. For $\sigma \in P$, denote by P_σ the set $\{x \in P_* \mid x \geqslant \sigma\}$, i.e., the maximal elements in $P_{\geqslant\sigma}$. For upper set $U \subseteq P$, $P_U = \bigcup_{\sigma \in U} P_\sigma$. Equivalently, $P_U = P_* \cap U$.*

Claim 4.5. *Let P be a poset and U_1, U_2 be upper sets in P. Then, $P_{U_1} \cup P_{U_2} = P_{U_1 \cup U_2}$ and $P_{U_1} \cap P_{U_2} = P_{U_1 \cap U_2}$.*

Proof. First we prove that $P_{U_1} \cup P_{U_2} = P_U$, where $U = U_1 \cup U_2$. By definition, $P_U = U \cap P_*$. Therefore,

$$P_U = U \cap P_* = (U_1 \cup U_2) \cap P_* = (U_1 \cap P_*) \cup (U_2 \cap P_*) = P_{U_1} \cup P_{U_2}$$

Next, we prove that $P_{U_1 \cap U_2} = P_{U_1} \cap P_{U_2}$. We have the following.

$$P_{U_1 \cap U_2} = P_* \cap (U_1 \cap U_2) = (P_* \cap U_1) \cap (P_* \cap U_2) = P_{U_1} \cap P_{U_2}. \qquad \square$$

Definition 4.6 (Coded Space)**.** *Let D be a domain and let P be a poset. Then P is a coded space over D if $P_* = D$ and for all $\rho, \tau \in P$, $\rho \leqslant \tau$ if and only if $P_\rho \supseteq P_\tau$.*

Claim 4.7. *Let P be a coded space and let $U \subseteq P$ be a primitive upper set. Then for all $\rho \in P$, if $P_\rho \subseteq P_U$ then $\rho \in U$.*

Proof. By definition 4.4, there exists $\tau \in P$ such that $U = P_{\geqslant \tau}$. Therefore $P_\tau = P_U$ and so $P_\rho \subseteq P_\tau$. By definition of coded space, this implies that $\rho \geqslant \tau$ and therefore $\rho \in U$, which completes the proof. $\square$

We start by showing how to apply functions on the maximal elements upper sets.

Definition 4.8 (Sheaf Of Linear Codes)**.** *Let P be a coded space. A sheaf of linear codes $\mathcal{F}$ on P is a sequence of linear codes $(\mathcal{F}_\sigma)_{\sigma \in P}$ where $\mathcal{F}_\sigma \subseteq \mathbb{F}^\sigma$, such that for every $\sigma, \tau \in P$ with $\sigma \leqslant \tau$, if $c \in \mathcal{F}_\sigma$ then $c|_\tau \in \mathcal{F}_\tau$. For each upper set $U \subseteq P$, we define the local section*

$$\mathcal{F}(U) = \{c \in \mathbb{F}^{P_U} \mid \forall \sigma \in U : c|_{P_\sigma} \in \mathcal{F}_\sigma\}.$$

When clear from context, we will refer to $\mathcal{F}(P)$ as $\mathcal{F}$. The domain *of a sheaf code, denoted Dom$(\mathcal{F})$, is the domain P_* of its coded space.*

Proposition 4.9. *Let $\mathcal{F}$ be a sheaf code over coded space P and let $U \subseteq P$ be an upper set. Then, for all $\tau \in U$,*

$$\mathcal{F}(U)|_\tau \subseteq \mathcal{F}_\tau.$$

Proof. Let $\sigma \in U$ such that $\sigma \leqslant \tau$. Then $c|_\tau = (c|_\sigma)|_\tau$. Since $c \in \mathcal{F}(U)$ and $\sigma \in U$, it follows $c|_\sigma \in \mathcal{F}_\sigma$.

$$c|_\tau = (c|_\sigma)|_\tau \in \mathcal{F}_\tau,$$

where the final inclusion follows from definition of sheaf code. $\square$

Definition 4.10 (Extendable Upper Set)**.** *Let $\mathcal{F}$ be a sheaf code on coded space P. Let U, V be upper sets in P with $U \subseteq V$. Then U is extendable to V with respect to $\mathcal{F}$ if there exists a map $e \colon \mathcal{F}(U) \to \mathcal{F}(V)$ such that for all $f \in \mathcal{F}(U)$, $e(f)|_U = f$. We refer to e as an extension map and $e(f)$ as an extension of f in $\mathcal{F}(V)$. If e is unique, we say that U is* uniquely extendable *to V. We say U is* globally extendable *if it is extendable to P. Equivalently, U is globally extendable if $\mathcal{F}(P)|_U = \mathcal{F}(U)$.*

When $\mathcal{F}, P$ are clear from context, we drop the "with respect to $\mathcal{F}$".

Proposition 4.11. *If U is uniquely extendable to V then the unique extension map e is a bijection.*

Proof. Clearly e is an injection; it remains to show surjectivity. Suppose otherwise; then there exists $c \in \mathcal{F}(V)$ that is not in the image of e. But then $e(c|_U) = c' \neq c$, and both c and c' are extensions of $c|_U$, which contradicts the uniqueness of e. $\square$

Definition 4.12 (Code Cover)**.** *Let P be a coded space. $S \subseteq P$ is a cover of P if for all $x \in P_*$, there exists $\rho \in S$ such that $x \geq \rho$.*

Definition 4.13 (Upper Set Closure)**.** *Let P be a poset. Let $S \subseteq P$ be a set of elements in P. The upper set closure of S, denoted by U_S, is equal to $\bigcup_{\sigma \in S} P_{\geqslant \sigma}$.*

4.2 Gluing Posets

4.2.1 Upper set isomorphisms

Definition 4.14 (Upper set homomorphisms)**.** *Let P, P' be two posets and let $U \subseteq P$, $V \subseteq P'$ be upper sets. A function $g : U \to V$ is an upper set homomorphism if it preserves ordering, i.e. for all $\sigma, \tau \in U$, $\sigma \leqslant_P \tau$ if and only if $g(\sigma) \leqslant_{P'} g(\tau)$. We call g an upper set isomorphism if it is additionally bijective.*

Proposition 4.15. *Let P, P' be two posets, let $U \subseteq P, V \subseteq P'$ be upper sets and let $g : U \to V$ be an upper set isomorphism. Then $g|_{P_U}$ is a bijection between P_U and $P'_{g(U)}$.*

Proof. First we prove that $g|_{P_U}$ is injective. Let $x, y \in P_U$ such that $x \neq y$. Suppose by contradiction that $g|_{P_U}(x) = g|_{P_U}(y)$. Then

$$x = g^{-1}(g(x)) = g^{-1}(g|_{P_U}(x)) = g^{-1}(g|_{P_U}(y)) = g^{-1}(g(y)) = y,$$

which is a contradiction. Next we prove surjectivity. For all $y \in P'_V$, set $x = g^{-1}(y)$. To prove that $x \in P_U$, suppose by contradiction there exists $\alpha \in P_*$ such that $\alpha \succ x$. Then since U is an upper set, it follows that $\alpha \in U$ and so $g(\alpha) \geqslant g(x) = y$, which is a contradiction since $y \in P'_*$. Therefore, $g|_{P_U}$ satisfies injectivity and surjectivity, which completes the proof. □

Proposition 4.16 (Composition of upper set isomorphisms.)**.** *For each $i \in \{1, 2, 3\}$, let P_i be a poset and let $U_i \subseteq P_i$ be an upper set. Let $h_{1,2} : U_1 \to U_2$ and $h_{2,3} : U_2 \to U_3$ be upper set isomorphisms. Then $h_{2,3} \circ h_{1,2} : U_1 \to U_3$ is an upper set isomorphism.*

Proof. Let $\sigma, \tau \in U_1$ such that $\sigma \leqslant_{P_1} \tau$. Then since $h_{1,2}$ is an upper set isomorphism, it follows that $h_{1,2}(\sigma) \leqslant_{P_2} h_{1,2}(\tau)$. Therefore, since $h_{2,3}$ is an upper set isomorphism, it follows that $h_{2,3}(h_{1,2}(\sigma)) \leqslant_{P_3} h_{2,3}(h_{1,2}(\tau))$. This completes the proof. □

Proposition 4.17. *Let P, P' be two posets. Let $S \subseteq P_*, S' \subseteq P'_*$ be two subsets of maximal elements. Let $g : S \to S'$ be a bijection. Then g is an upper set isomorphism.*

Proof. For any poset, P, every subset of P_* is an upper set because for all $x \in P_*, y \in P$, y is not greater than x. Similarly, every two elements in S are unordered, and every two elements in S' are unordered so ordering is preserved by g. □

Proposition 4.18. *Let P, P' be two posets, let $U \subseteq P$ be a primitive upper set and let $V \subseteq P'$ be an upper set. Let $g : U \to V$ be an upper set isomorphism. Then $g(U)$ is a primitive upper set in P'.*

Proof. There exists $\rho \in P$ such that $U = P_{\geqslant \rho} = \{\tau \in P : \tau \geqslant \rho\}$. We show that $g(U) = P'_{\geqslant g(\rho)}$. Let $\tau \in g(U)$. Then $g^{-1}(\tau) \in U$ so $g^{-1}(\tau) \geqslant \rho$, therefore $\tau \geqslant g(\rho)$ and so $\tau \in P'_{\geqslant g(\rho)}$. On the other hand, suppose that $\tau \in P'_{\geqslant g(\rho)}$. Then $\tau \geqslant g(\rho)$. Since $g(\rho) \in g(U)$, by definition of upper set, $\tau \in g(U)$, which completes the proof. □

Proposition 4.19. *Let P, P' be two posets, let $U \subseteq P, V \subseteq P'$ be upper sets and let $g : U \to V$ be an upper set isomorphism. Then for all $\sigma \in U$,*

$$g(P_\sigma) = P'_{g(\sigma)}.$$

Proof. Let $y \in g(P_\sigma)$. Then $y = g(x)$ for some $x \in P_\sigma$. We know y is a maximal element in $P'_{g(U)}$. We need to prove now that it is larger than $g(\sigma)$. Well since $x \geqslant \sigma$, it follows that $g(x) \geqslant g(\sigma)$, so that holds. Next, let $\rho \in P'_{g(\sigma)}$. Then ρ is a maximal element larger than $g(\sigma)$. Therefore $g^{-1}(\rho)$ is a maximal element in P larger than σ. Therefore $g^{-1}(\rho) \in P_\sigma$. Therefore $\rho \in g(P_\sigma)$. □

4.2.2 Gluing of posets and coded spaces

Definition 4.20 (Disjoint union of posets)**.** *For a sequence of posets $(P_1, \leqslant_1), \ldots, (P_n, \leqslant_n)$, we denote their disjoint union by $\biguplus_{i=1}^n P_i$. That is, $\biguplus_{i=1}^n P_i = (\biguplus_{i=1}^n P_i, \leqslant)$, where $x \leqslant y$ if and only if there exists $i \in [n]$ such that $x, y \in P_i$ and $x \leqslant_i y$.*

Definition 4.21 (Quotient poset)**.** *For a poset $(P, \leqslant)$ and equivalence relation $\sim \subseteq P \times P$, denote by $P/\!\sim$ the set whose elements are equivalence classes of P, and let $\leqslant_\sim$ be the relation: $[x] \leqslant_\sim [y]$ if and only if there exists $x_0 \in [x], y_0 \in [y]$ such that $x_0 \leqslant y_0$. This relation is reflexive but is not in general a partial ordering; when it is, we overload notation and denote by $P/\!\sim$ the poset $(P/\!\sim, \leqslant_\sim)$.*

Definition 4.22 (Gluing of posets)**.** *Let $\mathcal{P} = (P_1, \ldots, P_n)$ be a sequence of posets, $U_{i,j} \subseteq P_i$ be upper sets for each $i, j \in [n]$, and $\mathcal{H} = (h_{i,j} \colon U_{i,j} \to U_{j,i})_{i,j \in [n]}$ a sequence of isomorphisms of upper sets. (Going forward we will take $U_{i,i} = P_i$ and $h_{i,i}$ to be the identity.) Denote by $\sim$ the equivalence relation generated by $x \sim h_{i,j}(x)$ for all $i, j \in [n]$, $x \in U_{i,j}$. Then $\cup_{\mathcal{H}} \mathcal{P} \coloneqq (\biguplus_{i=1}^n P_i)/\!\sim$.*

Definition 4.23 (Compatibility condition)**.** *We say that $\mathcal{H}$ is* compatible *if, for every $i, j, k \in [n]$, for all $x \in U_{k,i} \cap U_{k,j}$, $h_{k,i}(x) \in U_{i,j}$ and $h_{i,j}(h_{k,i}(x)) = h_{k,j}(x)$.*

Lemma 4.24. *If $\mathcal{H}$ is compatible then $\cup_{\mathcal{H}} \mathcal{P}$ is a poset. Moreover, the natural projection maps $h_i \colon P_i \to \cup_{\mathcal{H}} \mathcal{P}$, given by $h_i(x) = [x]$ for all $x \in P_i$, are injective and order-preserving.*

Proof. We show that $\leqslant_\sim$ is a partial order. First, it is easy to see that the compatibility condition implies $h_{i,j} = h_{j,i}^{-1}$ for any $i, j \in [n]$. We next show the following key structural property.

Claim 4.25. *For any $x \in P_i$ and $y \in P_j$, $x \sim y$ if and only if $x \in U_{i,j}$ and $y = h_{i,j}(x)$.*

Proof. It suffices to show that any three-term sequence

$$x \xrightarrow{h_{i,k}} z \xrightarrow{h_{k,j}} y$$

can be contracted; this operation can then be repeated to contract a sequence of any length. Note that $z \in U_{k,i} \cap U_{k,j}$, and so $h_{i,j}(h_{k,i}(z)) = h_{k,j}(z)$ by the compatibility condition, whence $h_{i,j}(x) = y$. □

Transitivity. Let $[x], [y], [z] \in \cup_{\mathcal{H}} \mathcal{P}$ such that $[x] \leqslant_\sim [y], [y] \leqslant_\sim [z]$. Then there exist i, j, $x_i \in [x] \cap P_i, y_i \in [y] \cap P_i, y_j \in [y] \cap P_j, z_j \in [z] \cap P_j$ such that $x_i \leqslant_i y_i, y_j \leqslant_j z_j$. Since $h_{j,i}$ is an isomorphism of upper sets, $h_{j,i}(y_j) \leqslant_i h_{j,i}(z_j)$. By Claim 4.25, $h_{j,i}(y_j) = y_i$. Therefore $x_i \leqslant_i h_{j,i}(z_j)$, whence $[x] \leqslant_\sim [z]$.

Antisymmetry. Let $[x], [y] \in \cup_{\mathcal{H}} \mathcal{P}$ such that $[x] \leqslant_\sim [y]$ and $[y] \leqslant_\sim [x]$. Then there exist i, j, $x_i \in [x] \cap P_i, y_i \in [y] \cap P_i, y_j \in [y] \cap P_j, x_j \in [x] \cap P_j$ such that $x_i \leqslant_i y_i, y_j \leqslant_j x_j$. Since $h_{j,i}$ is an isomorphism of upper sets, $h_{j,i}(y_j) \leqslant_i h_{j,i}(x_j)$. By Claim 4.25, $h_{j,i}(y_j) = y_i$ and $h_{j,i}(x_j) = x_i$, whence $y_i \leqslant_i x_i$, and so $y_i = x_i$ and $[x] = [y]$ as required.

Finally, we show that the projection maps h_i are injective and order-preserving. Injectivity follows directly from Claim 4.25. By definition of $\leq_\sim$, if $x_i \leq_i y_i$ then $[x_i] \leq_\sim [y_i]$. Now suppose that $[x_i] \leq_\sim [y_i]$ for $x_i, y_i \in P_i$. Then there exists $x_j \sim x_i$, $y_j \sim y_i$ such that $x_j \leq y_j$. By Claim 4.25, $x_i = h_{j,i}(x_j)$ and $y_i = h_{j,i}(y_j)$. Since $h_{j,i}$ is order-preserving, $x_i \leq y_i$. □

Claim 4.26. *For all $\alpha \in U_{j,i}$,*

$$h_i^{-1}(h_j(\alpha)) = h_{j,i}(\alpha).$$

Proof. α and $h_{j,i}(\alpha)$ are in the same equivalence class. $h_i^{-1}(h_j(\alpha))$ is equal to the element in $[\alpha]$ that is in P_i. Since $h_{j,i}(\alpha) \in [\alpha]$ and since $h_{j,i}(\alpha) \in P_i$, it follows that $h^{-1}(h_j(\alpha)) = h_{j,i}(\alpha)$, which completes the proof. □

Definition 4.27. *Let P be a coded space over domain D, and let $S \subseteq D$. We say that an upper set $U \subseteq P$ is* induced *by S if $\rho \in U$ if and only if $P_\rho \subseteq S$. We simply say that U is* induced *if there exists S such that U is induced by S.*

Lemma 4.28. *If $\mathcal{H}$ is compatible, each P_i is a coded space, and for every i, j the upper set $U_{i,j}$ is induced, then $\mathcal{P} \coloneqq \cup_{\mathcal{H}} \mathcal{P}$ is a coded space.*

Proof. Let $[\rho], [\tau] \in \cup_{\mathcal{H}} \mathcal{P}$. Suppose first that $[\rho] \leq_\sim [\tau]$ (this direction does not require a condition on the $U_{i,j}$). Then there exists $i \in [n]$, $\rho_i \in [\rho] \cap P_i$, $\tau_i \in [\tau] \cap P_i$, such that $\rho_i \leq_i \tau_i$. Since P_i is a coded space, we have $P_{i,\rho_i} \supseteq P_{i,\tau_i}$. Note that by Proposition 4.19, $h_i(P_{i,\rho_i}) = \mathcal{P}_{[\rho]}$; likewise for τ. Hence $\mathcal{P}_{[\rho]} \supseteq \mathcal{P}_{[\tau]}$.

Now suppose that $\mathcal{P}_{[\rho]} \supseteq \mathcal{P}_{[\tau]}$, $\rho \in P_i, \tau \in P_j$. It follows that $P_{i,\rho} \supseteq h_{j,i}(P_{j,\tau})$; in particular, $P_{j,\tau} \subseteq U_{j,i}$. Since $U_{j,i}$ is induced, $\tau \in U_{j,i}$, and so $P_{i,\rho} \supseteq P_{i,h_{j,i}(\tau)}$ by Proposition 4.19. Then $\rho \leq_i h_{j,i}(\tau)$ since P_i is a coded space, so $[\rho] \leq_\sim [\tau]$. □

The following proposition will be useful when proving that upper sets are induced.

Proposition 4.29. *Let $\sigma \in P$ and let $U = P_{\geqslant \sigma}$ be a primitive upper set. Then, U is induced by P_σ.*

Proof. By definition of primitive upper set, $\tau \in U$ if and only if $\tau \geqslant \sigma$ if and only if $P_\tau \subseteq P_\sigma$. This completes the proof. □

4.3 Glueing Sheaf Codes

4.3.1 Isomorphism of Sheaf Codes

In this section, we define a sheaf code isomorphism, which preserves local restrictions. In the next subsection, we will use this to define glued sheaf codes.

Definition 4.30 (Isomorphism of sheaf codes)**.** *Let $\mathcal{F}, \mathcal{F}'$ be sheaf codes on P, P' respectively, and let $U \subseteq P, V \subseteq P'$ be two upper sets. Let $g : U \to V$ be an upper set isomorphism. Then g is a sheaf code isomorphism between $\mathcal{F}$ and $\mathcal{F}'$ at U if*

$$g|_{P_U}(\mathcal{F}_\sigma) = \mathcal{F}'_{g(\sigma)} \quad \forall \sigma \in U,$$

where we view $g|_{P_U} : P_U \to P'_V$ as a permutation mapping (Definition 3.7).

For the remainder of the section, $\mathcal{F}, \mathcal{F}'$ are two sheaf codes on coded spaces P, P', $U \subseteq P, V \subseteq P'$ are two upper sets, and $g : U \to V$ is an upper set isomorphism and a sheaf code isomorphism between $\mathcal{F}$ and $\mathcal{F}'$ at U.

Proposition 4.31. *$g^{-1} : V \to U$ is a sheaf code isomorphism between $\mathcal{F}'$ and $\mathcal{F}$ at V.*

Proof. For all $\sigma \in V$,

$$g^{-1}(\mathcal{F}'_\sigma) = g^{-1}(g(\mathcal{F}_{g^{-1}(\sigma)})) = \mathcal{F}_{g^{-1}(\sigma)}.$$

The first equality follows because g is a sheaf code isomorphism at U and $g^{-1}(\sigma) \in U$. The second equality follows because $g^{-1} \circ g$ is the identity. This completes the proof. □

Proposition 4.32. *Let $W \subseteq U$ be an upper set on P. Then $g|_W$ is a sheaf code isomorphism between $\mathcal{F}$ and $\mathcal{F}'$ at W.*

Proof. For all $\sigma \in W$, $g|_{P_W}(\mathcal{F}_\sigma) = \mathcal{F}'_{g(\sigma)}$ because g is a sheaf code isomorphism at U and $\sigma \in W \subseteq U$. □

Proposition 4.33 (Composition)**.** *Let $\hat{\mathcal{F}}$ be a sheaf code with coded space $\hat{P}$ and let $W \subseteq \hat{P}$ be an upper set. Let $h : W \to U$ be an upper set isomorphism. If h is a sheaf code isomorphism between $\hat{\mathcal{F}}$ and $\mathcal{F}$ at W, then $g \circ h : W \to V$ is a sheaf code isomorphism between $\hat{\mathcal{F}}$ and $\mathcal{F}'$ at W.*

Proof. Let $\sigma \in W$.

$$g(h(\hat{\mathcal{F}}_\sigma)) = g(\mathcal{F}_{h(\sigma)}) = \mathcal{F}'_{g(h(\sigma))}.$$

The first equality follows because h is a sheaf code isomorphism at W, the second equality follows because g is a sheaf code isomorphism at U and by definition, $h(\sigma) \in U$. This completes the proof. □

Proposition 4.34 (Isomorphisms preserve local sections)**.**

$$g|_{P_U}(\mathcal{F}(U)) = \mathcal{F}'(g(U)).$$

Proof. Let $v \in g|_{P_U}(\mathcal{F}(U))$. We will show that $v \in \mathcal{F}'(g(U))$, which by definition of sheaf code requires proving that for all $\sigma \in g(U)$, $v|_\sigma \in \mathcal{F}'_\sigma$. For all $\sigma \in g(U)$, we have the following.

$$v|_\sigma \in g|_{P_U}(\mathcal{F}(U))|_\sigma = g|_{P_U}(\mathcal{F}(U)|_{g^{-1}(\sigma)}) \subseteq g|_{P_U}(\mathcal{F}_{g^{-1}(\sigma)}) = \mathcal{F}'_\sigma.$$

The first inclusion follows by definition of vector space restriction, the first equality follows by Proposition 3.8 because $g|_{P_U}$ is a permutation mapping (Definition 3.7), the second inclusion follows since $g^{-1}(\sigma) \in U$ and by Proposition 4.9. The final equality follows by definition of sheaf isomorphism.

For the other direction, let $w \in \mathcal{F}'(g(U))$. We show that $g^{-1}(w) \in \mathcal{F}(U)$, which is sufficient to prove the claim. For all $\sigma \in U$,

$$g|^{-1}_{P_U}(w)|_\sigma = g|^{-1}_{P_U}(w|_{g(\sigma)}) \in g|^{-1}_{P_U}(\mathcal{F}'_{g(\sigma)}) = g|^{-1}_{P_U}(g|_{P_U}(\mathcal{F}_\sigma)) = \mathcal{F}_\sigma.$$

The first equality follows from Proposition 3.8, the inclusion follows because $g(\sigma) \in g(U)$, $w \in \mathcal{F}'(g(U))$ and by definition of upper set, the second equality follows by definition of sheaf code isomorphism, and the final equality follows from definition of permutation map. □

Lemma 4.35 (Isomorphisms preserve extendability)**.** *Let $X, W \subseteq U$ be two upper sets. If X is extendable to W with respect to $\mathcal{F}$, then $g(X)$ is extendable to $g(W)$ with respect to $\mathcal{F}'$.*

Proof. Let $f \in \mathcal{F}'(g(X))$. By Proposition 4.34, $\mathcal{F}'(g(X)) = g|_{P_U}(\mathcal{F}(X))$. Therefore,

$$g|^{-1}_{P_U}(f) \in g|^{-1}_{P_U}(\mathcal{F}'(g(X)) = g|^{-1}_{P_U}(g|_{P_U}(\mathcal{F}(X))) = \mathcal{F}(X).$$

Let $h \in \mathcal{F}(W)$ be the extension of $g|^{-1}_{P_U}(f)$ in W, which must exist since X is extendable to W. We show next that $g|_{P_U}(h)$ is an extension of f in $g(W)$.

$$g|_{P_U}(h)|_{g(X)} = g|_{P_U}(h|_X) = g|_{P_U}(g|^{-1}_{P_U}(f)) = f.$$

The first equality follows by Proposition 3.8, the second follows by definition of h and the third follows because g is a bijection. □

4.3.2 Glueing Sheaf Codes

Let $n \in \mathbb{N}$ and let $P_1, .., P_n$ be a set of pairwise disjoint coded spaces, for each $i < j \in [n]$, let $h_{i,j} : V_{i,j} \to P_j$ be an upper set isomorphism for some upper set $V_{i,j} \subseteq P_i$. For $j > i \in [n]$, denote by $h_{j,i} := h^{-1}_{i,j}$. Furthermore, for all $i, j, k \in [n]$, P_i, P_j, P_k satisfy three-way compatibility. Finally, for each $i \in [n]$ and non-empty subset $S \subseteq [n] \setminus \{i\}$, $\bigcup_{j \in S} \text{Dom}(h_{i,j})$ is a primitive upper set. Let $P_M, h_1, .., h_n$ be the output of merging $P_1, .., P_n$ with $\{h_{i,j} : i < j \in [n]\}$.

Definition 4.36 (Glueing Sheaf Codes)**.** *Let $n \in \mathbb{N}$ and let $\mathcal{F}_1, .., \mathcal{F}_n$ be a set of sheaf codes where for each $i \in [n]$, $\mathcal{F}_i$ is defined over coded space P_i, and for each $j < i$, $h_{i,j}$ is a sheaf code isomorphism between $\mathcal{F}_i$ and $\mathcal{F}_j$ at upper set Dom$(h_{i,j})$. We define the glued sheaf $\mathcal{F}$ on coded space P_M as follows.*

$$\mathcal{F}_{h_i(\alpha)} \coloneqq h_i|_{P_{i*}}(\mathcal{F}_{i\alpha}) \quad \forall i \in [n], \alpha \in P_i \qquad \mathcal{F}_{h_i(P_{i*})} \coloneqq h_i|_{P_{i*}}(\mathcal{F}_i) \quad \forall i \in [n] \tag{3}$$

For the following claims and lemma, let G be the output of Definition 4.36 when given sheafs $\mathcal{F}_1, .., \mathcal{F}_n$ as input. We first show that G is indeed a sheaf code.

Proposition 4.37. *$\mathcal{F} = G$ is a sheaf code.*

Proof. Let P be the coded space for $\mathcal{F}$. We start by showing that $\mathcal{F}$ is well-defined. Let $i, j \in [n], \alpha \in P_i, \beta \in P_j$ such that $h_i(\alpha) = h_j(\beta)$. Therefore $\alpha \sim \beta$ and so by definition of $\sim$, $\alpha \in U_{i,j}$ and $\beta = h_{i,j}(\alpha)$. Therefore,

$$h_j(\mathcal{F}_{j\beta}) = h_j(\mathcal{F}_{jh_{i,j}(\alpha)}) = h_j(h_{i,j}(\mathcal{F}_{i\alpha})) = h_i(\mathcal{F}_{i\alpha}),$$

where the last equality follows from Claim 4.26. Next, we show that $\mathcal{F}$ satisfies the definition of a sheaf code. It suffices to prove that for every $\sigma, \tau \in P$ with $\sigma \leqslant \tau$, $\mathcal{F}_\sigma|_\tau \subseteq \mathcal{F}_\tau$. Let $\sigma, \tau \in P$ with $\sigma \leqslant \tau$. By construction of G there exist $i, j \in [n]$ such that $\sigma = h_i(\alpha), \tau = h_j(\beta)$ for some $\alpha \in P_i, \beta \in P_j$. Since $Im(h_i)$ is an upper set and since $h_i(\alpha) \leqslant h_j(\beta)$, it follows that $h_j(\beta) \in Im(h_i)$ and therefore there exists $x \in P_i$ such that $h_j(\beta) = h_i(x)$. We prove the following fact.

Fact 4.38.

$$\alpha \leqslant h_i^{-1}(h_j(\beta))$$

Proof. Since $Im(h_i)$ is an upper set and since $h_i(\alpha) \leqslant h_j(\beta)$, it follows that $h_j(\beta) \in Im(h_i)$. Therefore, $h_j(\beta)$ is in the domain of h_i^{-1}. Since h_i^{-1} is an upper set isomorphism, $h_i(\alpha) \leqslant h_j(\beta)$ implies $\alpha \leqslant h^{-1}(h_j(\beta))$, which completes the proof. □

$$\begin{aligned}
&\mathcal{F}_\sigma|_\tau = \mathcal{F}_{h_i(\alpha)}|_{h_j(\beta)} \\
&= h_i(\mathcal{F}_{i\alpha})|_{h_j(\beta)} && \text{(by definition of merged sheaf)} \\
&= h_i(\mathcal{F}_{i\alpha})|_{h_i(h_i^{-1}(h_j(\beta)))} && \text{(Since } h_j(\beta) \in Im(h_i)\text{)} \\
&= h_i(\mathcal{F}_{i\alpha}|_{h_i^{-1}(h_j(\beta))}) && \text{(by Proposition 3.8)} \\
&\subseteq h_i(\mathcal{F}_{i\, h_i^{-1}(h_j(\beta))}) && \text{(By Fact 4.38 and since } \mathcal{F} \text{ is a sheaf code)} \\
&= \mathcal{F}_{h_i(h_i^{-1}(h_j(\beta)))} && \text{(by definition of merged sheaf} \\
&= \mathcal{F}_{h_j(\beta)} \\
&= \mathcal{F}_\tau. && \square
\end{aligned}$$

Proposition 4.39. *For each $i \in [n]$, h_i is a sheaf code isomorphism (Definition 4.30) between $\mathcal{F}_i$ and $\mathcal{F} = G$ at upper set P_i.*

Proof. Let $\sigma \in P_i$. Then $h_i|_{P_{i*}}(\mathcal{F}_{i\sigma}) = \mathcal{F}_{h_i(\sigma)}$ by definition of merged sheaf code (Definition 4.36). This completes the proof. □

4.4 The Dimensions Of Local Codes

In this subsection, we show that the padded dual of $\mathcal{F}(U)$ decomposes into the sum of the padded duals of the local codes in U. We then use this to derive the dimension of $\mathcal{F}(U)$ from the dimensions of its local codes. This will be useful later when proving extendability.

Lemma 4.40 (Decomposition of Local (Padded) Dual Codes)**.** *Let $\mathcal{F}$ be a sheaf code on coded space P. Let U be an upper set in P. Then,*

$$\overline{\mathcal{F}(U)^\perp} = \sum_{\sigma \in U} \overline{\mathcal{F}_\sigma^\perp}.$$

Proof. To show $\overline{\mathcal{F}(U)^\perp} \subseteq \sum_{\sigma\in U} \overline{\mathcal{F}_\sigma^\perp}$, it suffices to show that $(\sum_{\sigma\in U} \overline{\mathcal{F}_\sigma^\perp})^\perp \subseteq \widehat{\mathcal{F}(U)}$. By Claim 3.14, $(\sum_{\sigma\in U} \overline{\mathcal{F}_\sigma^\perp})^\perp = \bigcap_{\sigma\in U} \overline{\mathcal{F}_\sigma^\perp}^\perp = \bigcap_{\sigma\in U} \widehat{\mathcal{F}_\sigma}$. Next, we prove the following claim.

Claim 4.41.

$$\bigcap_{\sigma\in U} \widehat{\mathcal{F}_\sigma} \subseteq \widehat{\mathcal{F}(U)}_D.$$

Proof. If $v \in \bigcap_{\sigma\in U} \widehat{\mathcal{F}_\sigma}$ then $v \in \widehat{\mathcal{F}_\sigma}$ for all $\sigma \in U$, which by Claim 3.16, implies that $v|_{P_\sigma} \in \mathcal{F}_\sigma$ for all $\sigma \in U$. Therefore, by definition of sheaf of linear codes (Definition 4.8), $v|_{P_U} \in \mathcal{F}(U)$, and so by Claim 3.16, $v \in \widehat{\mathcal{F}(U)}$. □

Next, we show that $\sum_{\sigma\in U} \overline{\mathcal{F}_\sigma^\perp} \subseteq \overline{\mathcal{F}(U)^\perp}$. Let $v \in \sum_{\sigma\in U} \overline{\mathcal{F}_\sigma^\perp}$. Then, there exists a set $\{v_\sigma \in \overline{\mathcal{F}_\sigma^\perp} : \sigma \in U\}$ such that $v = \sum_{\sigma\in U} v_\sigma$. We will use the following fact, which shows that v is in the dual of $\widehat{\mathcal{F}(U)}$.

Fact 4.42. For all $c \in \widehat{\mathcal{F}(U)}$,

$$\langle c, v\rangle = 0.$$

Proof.

$$\langle c, v\rangle = \langle c, \sum_{\sigma\in U} v_\sigma\rangle = \sum_{\sigma\in U} \langle c, v_\sigma\rangle = \sum_{\sigma\in U} \langle c|_{P_\sigma}, v_\sigma|_{P_\sigma}\rangle = 0.$$

The second to last equality follows because for each $\sigma \in U$, $\mathsf{supp}(\overline{\mathcal{F}_\sigma^\perp}) \subseteq P_\sigma$. The last equality follows because by Claim 3.16 and definition of sheaf of linear codes (Definition 4.8), for all $\sigma \in U$, $c|_{P_\sigma} \in \mathcal{F}_\sigma$ and by definition of padded vector space (Definition 3.9), $v_\sigma|_{P_\sigma} \in \mathcal{F}_\sigma^\perp$. □

Therefore, by definition of dual (Definition 3.13), $v \in (\widehat{\mathcal{F}(U)}_D)^\perp = \overline{\mathcal{F}(U)^\perp}$, which completes the proof. □

Corollary 4.43. *Let $\mathcal{F}$ be a sheaf code on coded space P. Let U_1, U_2 be upper sets in P. Then,*

$$\overline{\mathcal{F}(U_1 \cup U_2)^\perp} = \overline{\mathcal{F}(U_1)^\perp} + \overline{\mathcal{F}(U_2)^\perp},$$

Proof.

$$\overline{\mathcal{F}(U_1 \cup U_2)^\perp} = \sum_{\sigma\in U_1\cup U_2} \overline{\mathcal{F}_\sigma^\perp} = \sum_{\sigma\in U_1} \overline{\mathcal{F}_\sigma^\perp} + \sum_{\sigma\in U_2} \overline{\mathcal{F}_\sigma^\perp} = \overline{\mathcal{F}(U_1)^\perp} + \overline{\mathcal{F}(U_2)^\perp}$$

where the first equality follows by Lemma 4.40, the second equality follows by definition of summation, and the third equality once again follows by Lemma 4.40. □

Lemma 4.44 (Dimension of $\mathcal{F}(U)$)**.** *Let $\mathcal{F}$ be a sheaf code on coded space P. Let U_1, U_2 be upper sets in P such that $U = U_1 \cup U_2$. Then,*

$$\dim(\mathcal{F}(U)) = \dim(\mathcal{F}(U_1)) + \dim(\mathcal{F}(U_2)) - (|P_{U_1} \cap P_{U_2}| - \dim(\overline{\mathcal{F}(U_1)^\perp} \cap \overline{\mathcal{F}(U_2)^\perp})).$$

Proof.

$$\begin{aligned}
&\dim(\mathcal{F}(U)) = |P_U| - \dim(\mathcal{F}(U)^\perp) && \text{(By Claim 3.15)}\\
&= |P_U| - \dim(\overline{\mathcal{F}(U)^\perp}) && \text{(By Corollary 3.11)}\\
&= |P_U| - \dim(\overline{\mathcal{F}(U_1)^\perp} + \overline{\mathcal{F}(U_2)^\perp}) && \text{(By Corollary 4.43)}\\
&= |P_{U_1}| + |P_{U_2}| - |P_{U_1 \cap U_2}| && \text{(By Claim 4.5)}\\
&\quad - \left(\dim(\overline{\mathcal{F}(U_1)^\perp}) + \dim(\overline{\mathcal{F}(U_2)^\perp}) - \dim(\overline{\mathcal{F}(U_1)^\perp} \cap \overline{\mathcal{F}(U_2)^\perp})\right) && \text{(By Claim 3.6)}\\
&= (|P_{U_1}| - \dim(\overline{\mathcal{F}(U_1)^\perp})) + (|P_{U_2}| - \dim(\overline{\mathcal{F}(U_2)^\perp})) - (|P_{U_1 \cap U_2}| - \dim(\overline{\mathcal{F}(U_1)^\perp} \cap \overline{\mathcal{F}(U_2)^\perp}))\\
&= (|P_{U_1}| - \dim(\mathcal{F}(U_1)^\perp)) + (|P_{U_2}| - \dim(\mathcal{F}(U_2)^\perp)) - (|P_{U_1 \cap U_2}| - \dim(\overline{\mathcal{F}(U_1)^\perp} \cap \overline{\mathcal{F}(U_2^\perp)})) && \text{(By Claim 3.11)}\\
&= \dim(\mathcal{F}(U_1)) + \dim(\mathcal{F}(U_2)) - (|(P_{U_1} \cap P_{U_2})| - \dim(\overline{\mathcal{F}(U_1)^\perp} \cap \overline{\mathcal{F}(U_2^\perp)}))) && \text{(By Claim 3.15)}
\end{aligned}$$

□

Corollary 4.45. *Let $\mathcal{F}$ be a sheaf code on coded space P. Let U_1, U_2 be upper sets in P such that $U = U_1 \cup U_2$ and $\overline{\mathcal{F}(U_1)^\perp} \cap \overline{\mathcal{F}(U_2)^\perp} = \{\vec{0}\}$. Then,*

$$\dim(\mathcal{F}(U)) = \dim(\mathcal{F}(U_1)) + \dim(\mathcal{F}(U_2)) - |P_{U_1} \cap P_{U_2}|.$$

Proof. Follows directly from Lemma 4.44 since $\dim(\{\vec{0}\}) = 0$. □

5 Testability of Sheaf Codes

In this section, we show how to use sheaf codes to test proximity of a function to a codeword.

5.1 Definitions

Definition 5.1 (Testable cover)**.** *Let $\mathcal{F}$ be a sheaf code with coded space P, and let $\rho \in P$. Let S be a cover for P_ρ. For $c \in \mathbb{N}$, S is a c-testable cover for $\mathcal{F}_\rho$ with distribution $\mathcal{D}(S)$ over S if for all functions $f : P_* \to \mathbb{F}$ and all $\delta \in (0, 1]$, if*

$$\mathop{\mathbb{E}}_{\sigma \leftarrow \mathcal{D}(S)} [\Delta(f|_\sigma, \mathcal{F}_\sigma)] \leq \delta,$$

then,

$$\Delta(f|_\rho, \mathcal{F}_\rho) \leq c \cdot \delta.$$

Definition 5.2 (d-testable cover)**.** *Let $\mathcal{F}$ be a sheaf of linear codes, P its coded space, $\sigma \in P$ and S a c-testable cover for $\mathcal{F}_\sigma$. Then S is a $(c, 1)$-testable cover. Moreover, for $d > 1$, S is a (c, d)-testable cover if for all $\tau \in S$, $\mathcal{F}_\tau$ has a $(c, d-1)$-testable cover in $\mathcal{F}$. For $i \leq d$, denote by $\mathcal{S}_i \subseteq P$ the set of codes that are in an i-testable cover and are at most $(c', i-1)$ testable for some constant c'.*

Definition 5.3 (d-testable sheaf code)**.** *Let $\mathcal{F}$ be a sheaf of linear codes, P its coded space, and let $\sigma \in P$. $\mathcal{F}$ is a (c, d)-testable sheaf code for $\mathcal{F}_\sigma$ if it contains a (c, d)-testable cover for $\mathcal{F}_\sigma$.*

Lemma 5.4. *Let $\mathcal{F}, \mathcal{F}'$ be two sheaf codes, let $U \subseteq P$ be an upper set, let $h : P \to P'$ be a sheaf code isomorphism on upper set U, let $S \subseteq U$ be a c-testable cover for $\sigma \in U$. Then $h(S)$ is a c-testable cover for $h(\sigma) \in h(U)$.*

Proof. Define $\mathcal{D}(h(S))$ to be the distribution where the probability of sampling $\rho \leftarrow \mathcal{D}(h(S))$ is equal to the probability of sampling $h^{-1}(\rho) \leftarrow \mathcal{D}(S)$. Then, we have the following.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\rho \leftarrow \mathcal{D}(h(S))}[\Delta(f|_\rho, h(\mathcal{F}_\rho)] \\
&= \mathop{\mathbb{E}}_{\rho \leftarrow \mathcal{D}(h(S))}[\Delta(f|_\rho, h(\mathcal{F}_\rho)] \\
&= \mathop{\mathbb{E}}_{\rho \leftarrow \mathcal{D}(h(S))}[\Delta(h^{-1}(f|_\rho), h^{-1}(h(\mathcal{F}_\rho))] \\
&= \mathop{\mathbb{E}}_{\rho \leftarrow \mathcal{D}(h(S))}[\Delta(h^{-1}(f)|_{h^{-1}(\rho)}, \mathcal{F}_\rho)] \\
&= \mathop{\mathbb{E}}_{h^{-1}(\rho) \leftarrow \mathcal{D}(S)}[\Delta(h^{-1}(f)|_{h^{-1}(\rho)}, \mathcal{F}_\rho)] \\
&= \mathop{\mathbb{E}}_{\alpha \leftarrow \mathcal{D}(S)}[\Delta(h^{-1}(f)|_\alpha, \mathcal{F}_\rho)] \\
&\leq \delta
\end{aligned}$$

Therefore, since S is a c-testable cover, $\Delta(h^{-1}(f), \mathcal{F}_\sigma) \leq c \cdot \delta$. Therefore $\Delta(f, h(\mathcal{F}_\sigma)) \leq c \cdot \delta$, which completes the proof. □

Corollary 5.5. *Let $\mathcal{F}, \mathcal{F}'$ be two sheaf codes, let $U \subseteq P$ be an upper set, let h be a sheaf code isomorphism at upper set U, let $S \subseteq U$ be a (c, d)-testable cover for $\sigma \in U$. Then $h(S)$ is a (c, d)-testable cover for $h(\sigma)$ in $h(U)$.*

Proof. We proceed with proof by induction on d. When $d = 1$, we apply Lemma 5.4. Next, let $d \in \mathbb{N}$. We show that for each $\rho \in h(S)$, $h(\mathcal{F})$ contains a $d-1$-testable cover for ρ. Since S is a d-testable cover, it follows that $h^{-1}(\rho)$ has a $d-1$ testable cover in $\mathcal{F}$. By inductive hypothesis, this implies that $h(\mathcal{F})$ contains a $d-1$ testable cover for ρ, which completes the proof. □

5.2 Recursive Testing of Sheaf Codes

In this subsection, we construct a codeword test for (c, d)-testable sheaf codes. This will be used as a building block for the PCPP presented in the next section.

Construction 5.6. Let $c, d \in \mathbb{N}$, let $\mathcal{F}$ be a sheaf of linear codes, let P be a coded space, let $\sigma \in P$, and let S be a (c, d)-testable cover for $\mathcal{F}_\sigma$ according to distribution $\mathcal{D}(S)$. This construction takes a function $f : \mathrm{Dom}(\mathcal{F}) \to \mathbb{F}$, σ, and S as input.

- If $d = 0$, then check that $f|_\sigma \in \mathcal{F}_\sigma$. If it is, output yes, otherwise output no.
- Sample $\rho \leftarrow \mathcal{D}(S)$ and let S_ρ be a $(d-1)$-testable cover of $\mathcal{F}_\rho$. Run this test recursively on inputs f, ρ, and S_ρ.

Theorem 5.7. *For any $\epsilon > 0$, Construction 5.6 is a $(\max_{\alpha \in \mathcal{S}_1}(|\alpha|))$-local codeword $\epsilon \cdot c^d$-tester for $\mathcal{F}_\sigma$.*

Proof. **Completeness.** If $d = 0$, then the construction outputs yes with probability 1 because by definition of sheaf code, $f \in \mathcal{F}(P)$ implies that $f|_{P_\sigma} \in \mathcal{F}_\sigma$. For $d > 1$, by definition of sheaf code, $f|_{P_\rho} \in \mathcal{F}_\rho$ and by assumption $\mathcal{F}_\rho$ has a $(d-1)$-testable cover, S_ρ. Therefore, by the inductive hypothesis, invoking $T(f, \rho, S_\rho)$ outputs yes with probability 1.

Soundness. We show that if $\Pr[T(f, \sigma, S) = \mathsf{no}] \leq \epsilon$ then $\Delta(f|_{P_\sigma}, \mathcal{F}_\sigma) \leq \epsilon \cdot c^d$. We proceed via induction. For the base case, let $d = 0$. Since the verifier accepts with probability strictly greater than zero, it follows that $f|_{P_\sigma} \in \mathcal{F}_\sigma$, since this condition is checked directly. Next, for the inductive step, for each $\tau \in S$, let ϵ_τ be the failure probability of $T(f, \tau, S_\tau)$. Then,

$$\mathop{\mathbb{E}}_{\tau \leftarrow \mathcal{D}(S)}[\Delta(f|_{P_\tau}, \mathcal{F}_\tau)] \leq \mathop{\mathbb{E}}_{\tau \leftarrow \mathcal{D}(S)}[\epsilon_\tau \cdot c^{d-1}] = \mathop{\mathbb{E}}_{\tau \leftarrow \mathcal{D}(S)}[\epsilon_\tau] \cdot c^{d-1}, \tag{4}$$

where the inequality follows by the inductive hypothesis, and the equality follows by linearity of expectation. Next, observe that

$$\mathop{\mathbb{E}}_{\tau \leftarrow \mathcal{D}(S)}[\epsilon_\tau] = \Pr[T(f, \sigma, S) = \mathsf{no}] \leq \epsilon$$

where the inequality is by assumption. Hence

$$\mathop{\mathbb{E}}_{\tau \leftarrow \mathcal{D}(S)}[\Delta(f|_{P_\tau}, \mathcal{F}_\tau)] \leq \epsilon \cdot c^{d-1},$$

where the last inequality follows by assumption of the theorem statement. Since S is a c-testable cover for $\mathcal{F}_\sigma$ according to $\mathcal{D}(S)$, it follows by Definition 5.1 that $\Delta(f|_{P_\sigma}, \mathcal{F}_\sigma) \leq c \cdot \epsilon \cdot c^{d-1} = \epsilon \cdot c^d$. □

5.3 Claims Related To Testability Of Sheaf Codes

The following lemma will be useful in Section 8.3, when we prove that a testable cover for Reed-Muller codes is a testable cover for its local Reed-Solomon codes with sufficiently large domains.

Lemma 5.8 (Distance Of Local Functions From Local Codes)**.** *Let $\mathcal{F}$ be a sheaf code with coded space P, and let $\sigma \in P$. Let $f : Dom(\mathcal{F}) \to \mathbb{F}$ be a function. Then*

$$\Delta(f|_\sigma, \mathcal{F}_\sigma) \leq \Delta(f, \mathcal{F}) \cdot \frac{|Dom(\mathcal{F})|}{|\sigma|}. \tag{5}$$

Proof. Let c be a codeword in $\mathcal{F}$ that is closest to f in Hamming distance, breaking ties arbitrarily. Let $E \subseteq \mathrm{Dom}(\mathcal{F})$ be a maximal subset such that $f|_E \neq c|_E$. By definition, $|E| = \Delta(f, \mathcal{F}) \cdot |\mathrm{Dom}(\mathcal{F})|$. Let $E' = E \cap \sigma$. Then $|E'| \leq |E|$ and so we have the following.

$$\Delta(f|_\sigma, c|_\sigma) = \frac{|E'|}{|\sigma|} \leq \frac{|E|}{|\sigma|} = \frac{\Delta(f, \mathcal{F}) \cdot |\mathrm{Dom}(\mathcal{F})|}{|\sigma|}.$$

By definition of sheaf code, $c|_\sigma \in \mathcal{F}_\sigma$, which completes the proof. □

Corollary 5.9 (Testable Covers Of Large Subcodes)**.** *Let $c \in \mathbb{N}$, let $\mathcal{F}$ be a sheaf code with coded space P, and let $\rho \in P$. Let S be a c-testable cover according to distribution $\mathcal{D}(S)$ for $\mathcal{F}_\rho$ and let $\sigma \in P$ such that $\sigma \geqslant \rho$. If there exists a constant $k \in \mathbb{N}$ such that*

$$\frac{|\rho|}{|\sigma|} = k,$$

then S is a $(c \cdot k)$-testable cover for $\mathcal{F}_\sigma$ according to distribution $\mathcal{D}(S)$.

Proof. First, if $x \in \sigma$ then it is also in ρ. Since S is a cover of ρ, there must exist $\tau \in S$ such that $x \in \tau$. Therefore, S is a cover for $\mathcal{F}_\sigma$. Next, let $f : \rho \to \mathbb{F}$ be a function. If $\mathbb{E}_{\tau \leftarrow \mathcal{D}(S)}[\Delta(f|_\tau, \mathcal{F}_\tau] \leq \delta$ then by definition of c-testable cover, there exists $c \in \mathbb{N}$ such that $\Delta(f, \rho) \leq c \cdot \delta$. Therefore, by Lemma 5.8,

$$\Delta(f|_\sigma, \mathcal{F}_\sigma) \leq \Delta(f, \mathcal{F}_\rho) \cdot \frac{|\rho|}{|\sigma|} \leq c \cdot k \cdot \delta,$$

which completes the proof. □

The following claim considers a code cover where the domains of the local codes are pairwise disjoint and the local codes all belong to the same space. This will be useful in Lemma 8.16, when proving that our final Reed-Muller sheaf code contains a testable cover.

Claim 5.10 (Probability a Local Restriction Is Far From Its Code)**.** *Let $\mathcal{F}$ be a sheaf code over coded space P. Let S be a cover for P such that for each $\tau \neq \rho \in S$, $\tau \cap \rho = \emptyset$ and there exists a code C such that for each $\tau \in S$, $\mathcal{F}_\tau = C$. Let $c \in \mathcal{F}(P)$, $f : Dom(\mathcal{F}) \to \mathbb{F}$ and let $\epsilon := \Delta(c, f)$. Then for all $\delta \in (0, 1]$,*

$$\Pr_{\tau \in S}[\Delta(c|_\tau, f|_\tau) > \delta] \leq \frac{\epsilon}{\delta}.$$

Proof. Let $S_\delta := \{\tau \in S : \Delta(f|_\tau, c|_\tau) > \delta\}$. Let D_S be the domain of C, which by definition is equal to τ for each $\tau \in S$.

$$\begin{aligned}
\Delta(f, c) \cdot |\mathrm{Dom}(\mathcal{F})| &= \sum_{\tau \in S} \Delta(f|_\tau, c|_\tau) \cdot |\tau| && \text{(Since domains of codes in } S \text{ are pairwise disjoint)} \\
&\geq \sum_{\rho \in S_\delta} \delta \cdot |\rho| + \sum_{\tau \in S \setminus S_\delta} \Delta(f|_\tau, c|_\tau) \cdot |\tau| && \text{(By definition of } S_\delta) \\
&\geq \sum_{\rho \in S_\delta} \delta \cdot |\rho| + 0 && \text{(Since all terms are positive)} \\
&= |S_\delta| \cdot \delta \cdot |D_S| && \text{(By definition of } D_S)
\end{aligned}$$

Therefore, $|S_\delta| \leq \frac{\Delta(f,c) \cdot |\mathrm{Dom}(\mathcal{F})|}{\delta \cdot |D_S|} \leq \frac{\epsilon \cdot |\mathrm{Dom}(\mathcal{F})|}{\delta \cdot |D_S|}$. By definition of S, $|\mathrm{Dom}(\mathcal{F})| = |S| \cdot |D_S|$. Therefore $|S_\delta| \leq \frac{\epsilon |S|}{\delta}$. Finally, we have the following.

$$\Pr_{\beta \in S}[\Delta(f|_\tau, c|_\tau) > \delta] = \frac{|S_\delta|}{|S|} \leq \frac{\epsilon |S|}{\delta |S|} = \frac{\epsilon}{\delta}.$$

□

6 Locally Simulatable Sheaf Codes

6.1 Basic Definitions

In this subsection, we provide everything we need to prove local simulatability of bivariate Reed-Muller codes. We start by defining locally simulatable encodings in the language of sheaf codes. The following definitions are modified from [GOS24], but put into the language of sheaf codes. Instead of point queries, the simulator accepts local codes in a cover of the sheaf code. In this subsection, $\mathcal{F}, P$ is a sheaf of linear codes and it's associated coded space.

Definition 6.1 (Linear randomised encoding function)**.** *We say that ENC : $M \to \mathcal{F}$ is a linear randomised encoding function if ENC(m) is distributed as ENC$(m; \mu)$ where $\mu \sim U(F^r)$ and ENC : $M \times F^r \to \mathcal{F}(P)$ is a (deterministic) linear function (of both message and randomness).*

Definition 6.2 (Locally simulatable encoding)**.** *Let $\mathsf{ENC} : \mathcal{M} \to \mathcal{F}(P)$ be a randomized encoding function, let S be a cover of $\mathcal{F}(P)$, let $t : \mathbb{N} \to \mathbb{N}$, let $\ell : \mathbb{N} \to \mathbb{N}$, and let $q \in \mathbb{N}$. We say that (ENC, S) is* (t, ℓ, q)-locally simulatable *if there exists a probabilistic algorithm Sim, called a* local simulator*, which receives oracle access to a message $m \in \mathcal{M}$, a set $T \subseteq S$ of local codes such that $|T| = n$, a function $f : P_{U_T} \to \mathbb{F}$, and a new query $\alpha \in S$, runs in time $t(n)$, and makes at most $\ell(n)$ queries to m, satisfying:*

$$\Pr\big[\mathcal{S}^m(T, \alpha) = \beta\big] \;=\; \Pr\big[\mathsf{ENC}(m)|_\alpha = \beta \;\big|\; \mathsf{ENC}(m)|_\rho = f(\rho) \;\; \forall \rho \in T\big],$$

for all $m \in \mathcal{M}$, $\alpha \in \mathcal{D}$, $\beta \in \Sigma$, and for all T such that $|T| < q$ and there exists $c \in \mathrm{supp}(\mathsf{ENC}(m))$ with $c_\rho = y$ for all $\rho \in T$.

The following is the definition of κ-local cover (Definition 4.15) from [BCFGRS17], rewritten to fit this context.

Definition 6.3 (κ-extendable cover)**.** *Let $\mathcal{F}$ be a sheaf of linear codes on P. Let $S \subseteq P$ be a cover of P (Definition 4.12). Then S is a κ-extendable cover with respect to $\mathcal{F}$ if, for every $S' \subseteq S$ such that $|S'| \leq \kappa$, $U_{S'}$ is globally extendable (Definition 4.10) with respect to $\mathcal{F}$.*

6.2 Extendability Implies Local Simulatability

This is a short section describing how to go from sheaf codes to locally simulatable codes. We show that it suffices to find sheafs with good extendability properties.

Definition 6.4 (Extension Sheaf Code)**.** *Let $U \subseteq P$ be an upper set and let $d \in \mathcal{F}(U)$. An extension code for d, which we denote $\mathcal{F}_{U=d}$, is the function mapping each upper set, $V \subseteq P$, to a subset of $\mathcal{F}(V)$ as follows.*

$$\mathcal{F}_{U=d}(V) = \{c \in \mathcal{F}(V) : c|_{P_V \cap P_U} = d|_{P_V \cap P_U}\}.$$

We refer to $\mathcal{F}_{U=d}(P)$ as an extension sheaf code. When $U = P_{\geqslant \sigma}$ for some $\sigma \in P$, we use the notation $\mathcal{F}_{\sigma=d}$.

Lemma 6.5 (Efficiently Computable Extensions)**.** *Let U, V be two upper sets in P. Let $f \in \mathcal{F}(U)$. If U is extendable to $U \cup V$, then there exists an algorithm that takes as input $f|_{P_{U \cap V}}$ and outputs a uniformly random element of $\mathcal{F}_{U=f}(V)$ in time* $\mathrm{poly}(|P_V|)$.

Proof. Let B be a basis for $\mathcal{F}(V)$, let $f \in \mathcal{F}(U)$, and let $\{X_b : b \in B\}$ and $\{Y_x : x \in P_V \setminus P_U\}$ be formal variables. Consider the following system of equations.

$$S := \{\sum_{b \in B} X_b \cdot b[x] = Y_x : x \in P_V \setminus P_U\} \cup \{\sum_{b \in B} X_b \cdot b[x] = f(x) : x \in P_U \cap P_V\},$$

Choose uniformly a solution, $\{x_b : b \in B\} \cup \{y_x : x \in P_V \setminus P_U\}$, to S, which must exist by definition of extendability (Definition 4.10) and because U is extendable to $U \cup V$. Define $\tilde{f} \in \mathcal{F}(V)$ to be the function such that $\tilde{f}(x) = y_x$ for all $x \in P_V \setminus P_U$ and $\tilde{f}(x) = f(x)$ for all $x \in P_V \cap P_U$. Since B is a basis for $\mathcal{F}(V)$, it follows by definition of S that $\tilde{f} \in \mathcal{F}(V)$, and so by definition of extension sheaf code (Definition 6.4), $\tilde{f} \in \mathcal{F}_{U=f}(V)$. □

Definition 6.6 (Randomized Encoding For Extension Sheaf Codes)**.** *Let $\sigma \in P$. Denote by ENC_σ the distribution that takes a codeword $c \in \mathcal{F}_\sigma$ as input and outputs a uniformly sampled codeword in $\mathcal{F}_{\sigma=c}(P)$.*

Theorem 6.7. *Let $\sigma \in P$, let $q \in \mathbb{N}$ and $S \subseteq P$ such that for all $c \in \mathcal{F}_\sigma$, S is a q-extendable cover for $\mathcal{F}_{\sigma=c}$. Let $l_S, t_S : \mathbb{N} \to \mathbb{N}$ be functions satisfying the following.*

$$l_S(k) \leq k \cdot \max_{\tau \in S} |P_\tau \cap P_\sigma| \qquad\qquad t_S(k) \in \mathrm{poly}(k \cdot \max_{\tau \in S} |P_\tau|)$$

Then (ENC_σ, S) is (t_S, l_S, q)-locally simulatable.

Proof. We build a simulator for ENC_σ. The simulator has oracle access to $c \in \mathcal{F}_\sigma$, and is additionally given S, $\rho \in S$, subset $Q \subseteq S$ such that $|Q| < q$, and function $f : P_{U_Q} \to \mathbb{F}$.

Construction 6.8 ($Sim^c(f, Q, \rho, S)$)**.**

- If $\rho \in Q$, output $f|_\rho$.
- Fix $A := U_Q$ and $A^* := A \cup P_{\geqslant \rho}$.
- Uniformly sample $\tilde{f} \leftarrow \mathcal{F}_{\sigma=c}(A^*)$ such that $\tilde{f}|_{P_A} = f|_{P_A}$.
- Set $f := \tilde{f}$. Output $\tilde{f}|_{P_\rho}$.

We prove that the output of the simulator is identically distributed to the output of the randomized encoding.

Lemma 6.9. *Let ρ, f, Q, S be the input to the simulator. Fix $A := U_Q$ and $A^* = A \cup P_{\geqslant\rho}$. Let $c \in \mathcal{F}_\sigma$ be the input to the randomized encoding. Then,*

$$\Pr_{\pi \leftarrow Sim^c(f,Q,\rho)}[\pi = \pi^*] = \Pr_{\tilde{f} \leftarrow \mathsf{ENC}_\sigma(c)}[\tilde{f}|_{P_\rho} = \pi^* \mid \tilde{f}|_{P_A} = f|_{P_A}], \tag{6}$$

where $\pi^* \in \mathbb{F}^{P_\rho}$.

Proof. We start with the following claim.

Claim 6.10. *A^* is globally extendable with respect to $\mathcal{F}_{\sigma=c}$.*

Proof. Let $Q' := Q \cup \{\rho\}$. By definition, $A^* = U_{Q'}$. Since $Q' \subseteq S$ and $|Q'| \leq q$, and since by assumption, S is a q-extendable cover for $\mathcal{F}_{\sigma=c}$, it follows by Definition 6.3 that A^* is globally extendable with respect to $\mathcal{F}_{\sigma=c}$.

□

Using this fact, we have the following.

$$\begin{aligned}
&\Pr_{\pi \leftarrow Sim^p(f,Q,\rho)}[\pi = \pi^*] \\
&= \Pr_{\tilde{f} \leftarrow \mathcal{F}_{\sigma=c}(A^*)}[\tilde{f}|_{P_\rho} = \pi^* \mid \tilde{f}|_{P_A} = f|_{P_A}] && \text{(By definition of simulator)} \\
&= \Pr_{\tilde{f} \leftarrow \mathcal{F}_{\sigma=c}(P)|_{P_{A^*}}}[\tilde{f}|_{P_\rho} = \pi^* \mid \tilde{f}|_{P_A} = f|_{P_A}] && \text{(By Claim 6.10 and definition of extendability)} \\
&= \Pr_{\tilde{f} \leftarrow \mathsf{ENC}_\sigma(c)}[\tilde{f}|_{P_\rho} = \pi^* \mid \tilde{f}|_{P_A} = f|_{P_A}] && \text{(By definition of } \mathsf{ENC}_\sigma)
\end{aligned}$$

□

Next, we measure the simulator runtime and number of queries the simulator makes to the input for each query.

Lemma 6.11. *Let ρ, f, Q be input to the simulator. The simulator makes*

$$|P_\rho \cap P_\sigma|$$

queries to input c and runs in time $\mathrm{poly}(|P_\rho|)$.

Proof. Let $U := P_{\geqslant\sigma} \cup A$ and define $g \in \mathcal{F}(U)$ to be the codeword satisfying $g|_{P_A} = f$ and $g|_{P_\sigma} = c$. We prove the following fact.

Fact 6.12.

$$\tilde{f} \in \mathcal{F}_{U=g}(A^*).$$

Proof. By definition of $\tilde{f}$, $\tilde{f}$ is an extension of f with respect to $\mathcal{F}_{\sigma=c}$. Therefore it is in the set $(\mathcal{F}_{\sigma=c})_{A=f}(A^*)$. By definition of extension sheaf code, we have the following.

$$\begin{aligned}
&(\mathcal{F}_{\sigma=c})_{A=f}(A^*) \\
&= \{d \in \mathcal{F}_{\sigma=c}(A^*) : d|_{P_A \cap P_{A^*}} = f|_{P_A \cap P_{A^*}}\} \\
&= \{d \in \mathcal{F}(A^*) : d|_{P_\sigma \cap P_{A^*}} = c|_{P_\sigma \cap P_{A^*}} \wedge d|_{P_A \cap P_{A^*}} = f|_{P_A \cap P_{A^*}}\} \\
&= \{d \in \mathcal{F}(A^*) : d|_{P_U \cap P_{A^*}} = g|_{P_U \cap P_{A^*}}\} \\
&= \mathcal{F}_{U=g}(A^*)
\end{aligned}$$

□

By Lemma 6.5 and Fact 6.12, $\tilde{f}$ can be computed in time $\mathrm{poly}(|P_{A^*}|)$ with input $g|_{U \cap P_{A^*}}$. $U \cap P_{A^*} = P_{\geqslant\sigma} \cap P_{A^*} \cup A \cap A^*$. $g|_{A \cap A^*}$ was already computed. So the simulator needs to find $g|_{P_{\geqslant\sigma} \cap (P_{A^*} \setminus P_A)} = g|_{P_{\geqslant\sigma} \cap P_{\geqslant\rho}} = c|_{P_{\geqslant\sigma} \cap P_{\geqslant\rho}}$. This completes the proof. □

□

6.3 Proving κ-Extendability in Sheaf Codes

In the previous subsection, we observed that κ-extendability in extension sheaf codes implies a locally simulatable encoding. In this section, we provide the tools needed to later prove that covers of explicit extension sheaf codes are κ-extendable. We start by relating κ-extendability of extension sheaf codes to κ-extendability of regular (non extension) sheaf codes. Then, we provide various lemmas that show how to reduce one extendability problem to a simpler extendability problem. These facts will later be used to prove κ-extendability of different sheaf codes.

6.3.1 Extendability in Extension Sheaf Codes

In the following two lemmas, we related this property to κ-extendability in (non-extension) sheaf codes so that going forward we can focus on the more natural setting of general sheaf codes.

Lemma 6.13. *Let $V \subseteq P$, $d \in \mathcal{F}(V)$. Let U be an upper set in P. If $U \cup V$ is globally extendable with respect to $\mathcal{F}$ then it is globally extendable with respect to $\mathcal{F}_{V=d}$.*

Proof. Let $f \in \mathcal{F}_{V=d}(U \cup V)$. Then $f \in \mathcal{F}(U \cup V)$ and $f|_{P_V} = d$. Since $U \cup V$ is globally extendable with respect to $\mathcal{F}$, there exists a codeword $f^* \in \mathcal{F}(P)$ such that $f^*|_{P_{U\cup V}} = f$. Therefore $f^*|_{P_V} = f|_{P_V} = d$. Therefore $f^* \in \mathcal{F}_{V=d}(P)$, which completes the proof. □

6.3.2 Basic Facts About Extendability

We now turn our attention to proving extendability in (non-extension) sheaf codes. The following lemma relates unique extendability between U_1 and U_2 to the dimensions of $\mathcal{F}(U_1), \mathcal{F}(U_2)$.

Lemma 6.14. *Let U_1, U_2 be two upper sets in P such that U_1 is extendable to U_2. If $\dim(\mathcal{F}(U_1)) = \dim(\mathcal{F}(U_2))$, then U_1 is uniquely extendable to U_2.*

Proof. Let $r : \mathcal{F}(U_2) \to \mathcal{F}(U_1)$ be the restriction map defined as $r(f) = f|_{P_{U_1}}$. Since U_1 is extendable in U_2 it follows by definition of extendability (Definition 4.10) that r is surjective. Since $|\mathcal{F}(U_1)| = |\mathcal{F}(U_2)|$, it follows that r is also injective. By definition of injectivity, for all $f_1, f_2 \in \mathcal{F}(U_2)$, $r(f_1) = r(f_2)$ implies $f_1 = f_2$, which implies that U_1 is uniquely extendable to U_2. □

In the following lemma, we show that if $U \cap V$ is extendable to V then U is extendable to $U \cup V$. We also show the conditions under which U is uniquely extendable.

Lemma 6.15. *Let U, V be upper sets. If $U \cap V$ is extendable to V then U is extendable to $U \cup V$. Furthermore, if $\overline{\mathcal{F}(V)^\perp} \cap \overline{\mathcal{F}(U)^\perp} = \vec{0}$, and $\dim(\mathcal{F}(V)) = |P_U \cap P_V|$, then U is uniquely extendable to $U \cup V$.*

Proof. Let $f \in \mathcal{F}(U)$. Let $\tilde{f}$ be the extension of $f|_{P_{U\cap V}}$ in $\mathcal{F}(V)$. Then we construct function $f^* \in \mathcal{F}(U \cup V)$ as follows. $f^*|_{P_U} = f$ and $f^*|_{P_V} = \tilde{f}$. First, f^* is well-defined because $f|_{P_{U\cap V}} = \tilde{f}|_{P_{U\cap V}}$ by definition of extension. Next, $f^* \in \mathcal{F}(U \cup V)$ because $f^*|_{P_U} = f \in \mathcal{F}(U)$ and $f^*|_{P_V} = \tilde{f} \in \mathcal{F}(V)$. To prove unique extendability, we have the following.

$$\dim(\mathcal{F}(V \cup U)) = \dim(\mathcal{F}(V)) + \dim(\mathcal{F}(U)) - |P_{V\cap U}| = \dim(\mathcal{F}(V)) + \dim(\mathcal{F}(U)) - \dim(\mathcal{F}(V))$$
$$= \dim(\mathcal{F}(U)).$$

The first equality follows by Corollary 4.45 because $\overline{\mathcal{F}(V)^\perp} \cap \overline{\mathcal{F}(U)^\perp} = \vec{0}$, and the second equality follows because $|P_{V\cap U}| = \dim(\mathcal{F}(V))$. By Claim 6.14, U is uniquely extendable to $V \cup U$, which completes the proof. □

Finally, in the next lemma, we show that if the dimension of the restriction of $\mathcal{F}(V)$ is equal to the size of the domain of $\mathcal{F}(U)$, then U must be extendable to V.

Lemma 6.16 (Full Rank Restrictions Are Extendable)**.** *Let $V \subseteq P$ be an upper set and let $U \subseteq V$. If $\dim(\mathcal{F}(V)|_{P_U}) = |P_U|$, then U is extendable to V.*

Proof. Since $\mathcal{F}(V)|_{P_U} \subseteq \mathbb{F}^{P_U}$, it follows by assumption on the dimensions of the two spaces that $\mathcal{F}(V)|_{P_U} = \mathbb{F}^{P_U}$. Let $f \in \mathcal{F}(U)$. By definition sheaf code, $\mathcal{F}(U) \subseteq \mathbb{F}^{P_U}$ and so $f \in \mathbb{F}^{P_U}$. Therefore $f \in \mathcal{F}(V)|_{P_U}$ and so there exists $g \in \mathcal{F}(V)$ such that $g|_{P_U} = f$, which completes the proof. □

6.3.3 Extendability Reductions

In this section, we show how to reduce the extendability between two upper sets to a set of simpler extendability problems. This will be very useful when proving extendability in Sections 7 and 8.

Lemma 6.17 (Extendability To Disjoint Upper Sets)**.** *Let P be a coded space and $\mathcal{F}$ a sheaf of codes on P. Let U_1, U_2, V be upper sets in P such that $P_{U_1} \cap P_{U_2} = \emptyset$. If V is (uniquely) extendable to both $V \cup U_1$ and $V \cup U_2$, then V is (uniquely) extendable to $V \cup U_1 \cup U_2$.*

Proof. Let $f \in \mathcal{F}(V)$. Let f_1 be an (unique) extension of f in $U_1 \cup V$ and let f_2 be an (unique) extension of f in $U_2 \cup V$. Let $f^* \in \mathbb{F}^{P_{U_1} \cup P_{U_2} \cup P_V}$ be the following vector.

$$f^*_{P_V} = f \qquad f^*_{P_{U_1 \setminus V}} = f_1|_{P_{U_1 \setminus V}} \qquad f^*|_{P_{U_2 \setminus V}} = f_2|_{P_{U_2 \setminus V}}$$

f^* is well defined in terms of f_1, f_2 since $V, U_1 \setminus V$ and $U_2 \setminus V$ are pairwise disjoint. We show that $f^* \in \mathcal{F}(V \cup U_1 \cup U_2)$. By definition of f^*, $f^*|_{P_V} \in \mathcal{F}(V)$. Next, we prove the following.

Fact 6.18.

$$f^*|_{P_{U_1}} = f_1|_{P_{U_1}}.$$

Proof. $f^*|_{P_{V \cap U_1}} = f|_{P_{V \cap U_1}} = f_1|_{P_{V \cap U_1}}$, where the first equality follows by definition of f^* and the second equality follows because $V \cap U_1 \subseteq V \cup U_1$ and f_1 is an extension of f in $U_1 \cup V$. Also by definition of f^*, $f^*|_{P_{U_1 \setminus V}} = f_1|_{P_{U_1 \setminus V}}$. Therefore f^* agrees with f_1 on the upper set $U_1 \cap V$ as well as $U_1 \setminus V$ and so $f^*|_{P_{U_1}} = f_1|_{P_{U_1}}$. □

Next, we prove the analogous statement for $f^*|_{P_{U_2}}$.

Fact 6.19.

$$f^*|_{P_{U_2}} = f_2|_{P_{U_2}}.$$

Proof. Similar to the argument in the previous fact, $f^*|_{P_{V \cap U_2}} = f|_{P_{V \cap U_2}} = f_2|_{P_{V \cap U_2}}$ and $f^*|_{P_{U_2 \setminus V}} = f_2|_{P_{U_2 \setminus V}}$. Therefore, $f^*|_{P_{U_2}} = f_2|_{U_2}$. □

Since $f^*|_{P_{U_1}} = f_1|_{P_{U_1}} \in \mathcal{F}(U_1)$ and $f^*_{P_{U_2}} = f_2|_{P_{U_2}} \in \mathcal{F}(U_2)$, and since by definition of f^*, $f^*|_{P_V} \in \mathcal{F}(V)$, it follows that $f^* \in \mathcal{F}(V \cup U_1 \cup U_2)$. Finally, if V is uniquely extendable to both $V \cup U_1$ and $V \cup U_2$, then there is only one choice for both f_1 and f_2 and therefore f^* is the only extension of f in $U_1 \cup U_2 \cup V$, and so V is uniquely extendable to $U_1 \cup U_2 \cup V$. □

In the following two corollaries, $n \in \mathbb{N}$, $V, U_0, .., U_{n-1}$ are distinct upper sets in P, and V is an upper set in P such that V is (uniquely) extendable to $V \cup U_i$ for all $i \in [n]$.

Corollary 6.20. *Suppose that for each $i, j \in [n]$, $P_{U_i} \cap P_{U_j} = \emptyset$. Then V is (uniquely) extendable to $V \cup \bigcup_{i \in [n]} U_i$.*

Proof. We proceed with proof by induction on n. When $n = 1$, the statement follows trivially. When $n = 2$, the statement follows by Lemma 6.17. Next, let $n \in \mathbb{N}$ and let $U^* = \bigcup_{i \in [n-1]} U_i$. Then $\bigcup_{i \in [n]} U_i = U^* \cup U_{n-1}$. Furthermore, since $U_{n-1} \cap U_i = \emptyset$ for all $i \in [n-1]$, it follows by Claim 3.1 that $U_{n-1} \cap U^* = \emptyset$. By inductive hypothesis, V is (uniquely) extendable to $V \cup U^*$ and by assumption, V is (uniquely) extendable to $V \cup U_{n-1}$. Therefore, by Lemma 6.17, V is (uniquely) extendable to $V \cup U^* \cup U_{n-1} = V \cup \bigcup_{i \in [n]} U_i$, which completes the proof. □

Corollary 6.21. *Suppose that for each $i, j \in [n]$, $P_{U_i \setminus V} \cap P_{U_j \setminus V} = \emptyset$. Then V is (uniquely) extendable to $V \cup \bigcup_{i \in [n]} U_i$.*

Proof. By assumption, V is (uniquely) extendable to $V \cup U_i$ for all $i \in [n]$. Since $V \cup U_i = V \cup (U_i \setminus V)$, it follows that V is (uniquely) extendable to codewords satisfying constraints contained in $V \cup (U_i \setminus V)$ for all $i \in [n]$. Therefore, by Corollary 6.20, V is (uniquely) extendable to codewords satisfying constraints contained in $V \cup \bigcup_{i \in [n]} (U_i \setminus V) = \bigcup_{i \in [n]} (U_i \setminus V) \cup V = \bigcup_{i \in [n]} U_i \cup V = V \cup \bigcup_{i \in [n]} U_i$, which completes the proof. □

The following lemma states that if we have three upper sets U, V, W, and we know that U is *uniquely* extendable to V and extendable to W, then V is also extendable to W.

Lemma 6.22 (triangle of extendability)**.** *Let U, V, G be upper sets in P such that $U \subseteq V \subseteq G$. If U is uniquely extendable to V and extendable to G, then V is extendable to G.*

Proof. Let $f : \mathcal{F}(G) \to \mathcal{F}(V)$ be the restriction map of G to V and let $h : \mathcal{F}(U) \to \mathcal{F}(G)$ be any extension map of U to G. Then $f \circ h$ is an extension map of U to V. Since U is uniquely extendable to V, $f \circ h$ is a bijection by Proposition 4.11. Therefore, for each $v \in V$ there exists $u \in \mathcal{F}(U)$ such $f(h(u)) = v$. Since $h(u) \in \mathcal{F}(G)$, this completes the proof. □

Lemma 6.23 (extendability is transitive)**.** *Let $U_1, U_2, U_3 \subseteq P$ with $U_1 \subseteq U_2 \subseteq U_3$. If U_1 is (uniquely) extendable to U_2 and U_2 is (uniquely) extendable to U_3, then U_1 is (uniquely) extendable to U_3.*

Proof. Let $f_1 \in \mathcal{F}(U_1)$ and let $f_2 \in \mathcal{F}(U_2)$ be an extension of f_1 in $\mathcal{F}(U_2)$. Let $f_3 \in \mathcal{F}(U_3)$ be an extension of f_2 in $\mathcal{F}(U_3)$. Then $f_3|_{P_{U_1}} = (f_3|_{P_{U_2}})|_{P_{U_1}} = f_2|_{P_{U_1}} = f_1$ and so f_3 is an extension of f_1 in U_3. □

6.3.4 Extendability For Maximum Distance Separable Codes

In this subsection, we apply the results from the previous subsections to sheaf codes satisfying the maximum distance separable property 3.17. Since our the local codes of our bivariate Reed-Muller sheaf codes are Reed-Solomon codes, and therefore MDS, we will use this lemma when proving κ-extendability in Section 8.

Lemma 6.24. *Suppose that P contains a local code $\sigma \in P$ such that $\mathcal{F}_\sigma$ is a maximum distance separable code. Let $U \subseteq P$ be an upper set. If $|P_\sigma \cap P_U| \leq \dim(\mathcal{F}_\sigma)$ (resp. $= \dim(\mathcal{F}_\sigma)$), then U is extendable (resp. uniquely extendable) to $U \cup P_{\geqslant \sigma}$.*

Proof. Because $|P_U \cap P_\sigma| \leq \dim(\mathcal{F}_\sigma)$ it follows by Lemma 3.18 and the fact that $\mathcal{F}_\sigma$ is MDS that $\dim(\mathcal{F}_\sigma|_{P_U \cap P_\sigma}) = P_U \cap P_\sigma$ and so by Lemma 6.16, $P_{\geqslant \sigma} \cap U$ is extendable to $P_{\geqslant \sigma}$. Therefore by Lemma 6.15, U is extendable to $U \cup P_{\geqslant \sigma}$. Next, we prove that when $|P_U \cap P_\sigma| = \dim(\mathcal{F}_\sigma)$ then U is *uniquely* extendable to $U \cup P_{\geqslant \sigma}$. To prove this, we apply the second clause of Lemma 6.15, which requires us to prove the following claim.

Claim 6.25.

$$\overline{\mathcal{F}(U)^\perp} \cap \overline{\mathcal{F}_\sigma^\perp} = \vec{0}.$$

Proof. For each $v \in \overline{\mathcal{F}(U)^\perp} \cap \overline{\mathcal{F}_\sigma^\perp}$, the following must be satisfied. $v \in \overline{\mathcal{F}_\sigma^\perp}$ and $\mathsf{supp}(v) \subseteq P_U \cap P_\sigma$. By definition of padded vector space (Definition 3.9), for all $v \in \overline{\mathcal{F}_\sigma^\perp}$, $|\mathsf{supp}(v)| = |\mathsf{supp}(v|_{P_\sigma})|$ and $v|_{P_\sigma} \in \mathcal{F}_\sigma^\perp$. By Lemma 3.19, $|\mathsf{supp}(v|_{P_\sigma})| > \dim(\mathcal{F}_\sigma)$. This is a contradiction and completes the proof. □

□

7 Perfect Zero-Knowledge PCPPs From Sheaf Codes

In the previous section, we have shown that any sheaf code with a κ-extendable cover is a locally simulatable code. In this section, we construct a PCPP where the space of honest prover outputs is a sheaf code with a κ-extendable cover. This implies that the prover is a locally simulatable encoding, and therefore that the PCPP is perfect zero-knowledge.

We build up to our target sheaf code in stages. Recall in Section 5, we defined a d-testable sheaf code, which has a recursive structure that enables efficient local tests. In Subsections 7.1, we build a PCPP for d-testable sheaf codes. Finally, in Subsection 7.3, we construct a d-testable sheaf code from a class of code families. We show that for certain code families, the sheaf code also has a κ-extendable cover.

Looking ahead to Section 8, the PCPP from [BS08] can be recovered by instantiating Construction 7.19 with the class of Reed-Solomon codes in [BS08]. We identify a different class of Reed-Solomon codes, which force the output sheaf code of Construction 7.19 to have a κ-extendable cover. Combining that finding with the results of this section yields our perfect zero-knowledge PCPP for Reed-Solomon codes.

7.1 Perfect Zero-Knowledge PCPPs From d-Testable Sheaf Codes with κ-extendable covers

In this subsection, we construct PCPPs for d-testable sheaf codes. We show that if the sheaf code also has a κ-extendable cover, then the PCPP satisfies perfect zero-knowledge. The results of this section reduce the problem of PCPP construction to the task of finding sheaf codes with certain properties.

We begin by relating the definition of locally simulatable encoding to that of a perfect zero-knowledge PCPP. Recall that the local simulator in Definition 6.2 accepts elements of a coded space as queries, and simulates entire *local* codewords. In the following lemma, we build a simulator that simulates points instead of codewords by using the simulator from Definition 6.2.

Lemma 7.1 (Local simulatability implies black-box zero-knowledge PCPP)**.** *Let $q \in \mathbb{N}$ and let $t(n)$ be a function in $\mathrm{poly}(n, q)$. Let Π be a PCPP such that the prover of Π is a (t, l, q)-locally simulatable encoding. Then Π is black-box, l-local, perfect zero-knowledge PCPP with query bound q.*

Proof. Let $\mathcal{S}$ be the local simulator for the prover of the PCPP, which is guaranteed by Definition 6.2. We define a simulator $\mathcal{S}^*$ for the PCPP as follows. The simulator has oracle access to a message $m \in \mathcal{M}$ and is additionally given $x \in P_*$, $f : P_{U_T} \to \mathbb{F}$ for some $T \subseteq S$ which records the previous queries of the simulator.

Construction 7.2 (Simulator)**.**

- If $x \in P_{U_T}$, then output $f(x)$.
- Otherwise, find $\rho \in S$ such that $x \in \rho$, query $\beta \leftarrow \mathcal{S}^m(\rho)$
- Let $\tilde{f} : P_{U_{T \cup \{\rho\}}} \to \mathbb{F}$ such that $\tilde{f}|_{P_{U_T}} = f$ and $\tilde{f}|_\rho = \beta$.
- Set $f := \tilde{f}$.
- Output $\beta(x)$

We show that the above simulator is identically distributed to the view of the PCPP. Denote by $\Pi.P$ the prover of the PCPP.

$$\Pr_{y \leftarrow S^*(f,T,x)}[y = y^*] = \Pr_{\beta \leftarrow S(f,T,\alpha)}[\beta|_x = y^*]$$
$$= \Pr_{\tilde{f} \leftarrow \Pi.P(m)}[\tilde{f}|_x = y^* \mid \tilde{f}|_{P_{U_T}} = f|_{P_{U_T}}]$$

The second to last inequality follows because S is a locally simulatable encoding and the final term is the distribution of the PCPP view according to Definition 3.26. Furthermore, the simulator uses only

black-box access to the verifier, tolerates up to q verifier queries, and has the same running time as $\mathcal{S}$. Each query requires at most $l(n)$ input points, and runs in time $t(n)$. This completes the proof. □

In the following construction, we build a PCPP using the local codeword test in Construction 5.6. The key difference in this construction is that the verifier only has oracle access to the input, which is the local code $\mathcal{F}_\sigma$. As such, the verifier is testing a virtual function, which is partly made up of the proof string and partly made up of the input elements.

Construction 7.3 (PCPP)**.** In this construction, d is in $\mathbb{N}$, $\mathcal{F}$ is a d-testable sheaf of linear codes for some code C, P is the coded space for $\mathcal{F}$, $\sigma \in P$ such that $\mathcal{F}_\sigma = C$, and $S \subseteq P$ is a d-testable cover for $\mathcal{F}_\sigma$. Furthermore, we denote by T the test from Construction 5.6.

- **Prover**:
  - *Input:* $\mathcal{F}$, $c \in \mathcal{F}_\sigma$
  - *Output:* A uniformly distributed sample from $\mathcal{F}_{\sigma=c}(P)$.
- **Verifier**:
  - *Input*: $f : \mathbb{F}^{P_\sigma} \to \mathbb{F}$, and explicit input P, $g : \mathbb{F}^{P_*} \to \mathbb{F}$, σ and S.
  - *Output*: Let $\widehat{g} : \mathbb{F}^{P_*} \to \mathbb{F}$ be the function such that $\widehat{g}|_{P_* \setminus P_\sigma} = g|_{P_* \setminus P_\sigma}$ and $\widehat{g}|_{P_\sigma} = f$. The verifier outputs $T(\widehat{g}, \sigma, S)$.

Theorem 7.4. *If there exists $\kappa \in \mathbb{N}$ such that $\mathcal{F}_{\sigma=c}$ has a κ-extendable cover, then for all $\epsilon \in [0,1]$, Construction 7.3 is a black-box l_S-local (where l_S is a function defined in Theorem 6.7), perfect zero-knowledge PCPP for $\mathcal{F}_\sigma$ with query complexity $\max_{\alpha \in \mathcal{S}_1} |\alpha|$ (where $\mathcal{S}_1$ is from Definition 5.3), query bound κ, soundness error $(1-\epsilon)$, and proximity parameter $c^d \cdot \epsilon$.*

Proof. Completeness and soundness follow immediately from Theorem 5.7, and perfect zero-knowledge follows because by Theorem 6.7, and since $\mathcal{F}_{\sigma=c}$ has a κ-extendable cover, it follows that the prover is a locally simulatable encoding. Applying Lemma 7.1 completes the proof. □

7.2 Constructing d-testable and κ-extendable covers through glueing

In this subsection, we construct a d-testable sheaf code, which generalizes the recursive structure of the PCPP from [BS08]. We utilize the sheaf code merging (Definition 4.36) to "glue" together a set of $(d-1)$-testable sheaf codes in order to test a 1-testable sheaf code with testable cover S. Each of the $(d-1)$-testable sheaf codes gives a test for one of the codes in S. The following construction takes as input a main sheaf which contains the code, σ, that we would like to test, and which we denote $\mathcal{F}[t]$; a testable cover, S for local code $\sigma \in P[t]$; and for each $\rho \in S$, a $(d-1)$-testable sheaf code which we denote by $\mathcal{F}[\rho]$. Furthermore, to invoke the sheaf merging construction (Construction 4.36), we additionally require an upper set isomorphism $h_{t,\rho} : P[t]_{\geqslant \rho} \to P[\rho]$.

Construction 7.5 (d-Testable Sheaf Codes From $d-1$ Testable Sheaf Codes)**.** The construction takes the following inputs: a sheaf, $\mathcal{F}[t]$, defined over $P[t]$, a local code domain $\sigma \in P[t]$, testable cover $S \subseteq P[t]$ for code $\mathcal{F}[t]_\sigma$, and a set $\{\mathcal{F}[\rho] : \rho \in S\}$ where for each $\rho \in S$, $\mathcal{F}[\rho]$ is defined over coded space $P[\rho]$ such that there exists $\sigma_\rho \in P[\rho]$ such that $\mathcal{F}[\rho]$ is a $(d-1)$-testable sheaf code for $\mathcal{F}[\rho]_{\sigma_\rho}$ and finally, there exists an upper set isomorphism (Definition 4.14) $h_{t,\rho} : P[t]_{\geqslant \rho} \to P[\rho]$ such that $Im(h_{t,\rho}) = P[\rho]_{\geqslant \sigma_\rho}$.

- For each $\rho, \tau \in S$, let $\sigma := \rho \cap \tau$. Define $h_{\rho,\tau} : h_{t,\rho}(\sigma) \to h_{t,\tau}(\sigma)$ as $h_{\rho,\tau}(h_{t,\rho}(x)) = h_\tau(x)$ for all $x \in \sigma$. For each $\rho \in S$, set $h_{\rho,t} := h_{t,\rho}^{-1}$.
- Output the glueing (Definition 4.36) of sheafs $\{\mathcal{F}'\} \cup \{\mathcal{F}[\rho] : \rho \in S\}$ and upper set isomorphisms $\{h_{t,\rho}(\rho) : \rho \in S\} \cup \{h_{\rho,t} : \rho \in S\} \cup \{h_{\rho,\tau} : \rho, \tau \in S\}$.

For the remainder of the section, $\mathcal{F}[t]$ is a sheaf code with coded space $P[t]$, $\sigma \in P[t]$, $S \subseteq P[t]$ is a testable cover for $\mathcal{F}[t]_\sigma$, and $\{\mathcal{F}[\rho] : \rho \in S\}$ are as specified in Construction 7.5. We refer to $\mathcal{F}$ as the corresponding output of Construction 7.5, and to $h_t, \{h_\rho : \rho \in S\}$ as the corresponding upper set isomorphisms.

7.2.1 Correctness of Construction 7.5

In this paragraph, we prove that Construction 7.5 outputs a glued sheaf code. To prove this, we show that the inputs given to the glued sheaf construction satisfy the conditions required by Definition 4.36.

Lemma 7.6. *For any $\mathcal{F}[t], \sigma, S, \{\mathcal{F}[\rho] : \rho \in S\}$, Construction 7.5 outputs a glued sheaf, i.e. a sheaf satisfying Definition 4.36.*

Proof. We need to prove that $\{h_{t,\rho}(\rho) : \rho \in S\} \cup \{h_{\rho,t} : \rho \in S\} \cup \{h_{\rho,\tau} : \rho, \tau \in S\}$ are all upper set isomorphisms satisfying the compatibility condition and satisfying $h_{\rho,\tau} = h_{\tau,\rho}^{-1}$ for all $\tau, \rho \in S$. Furthermore, we show that for all $\rho \in S$, $P[\rho]_{\geqslant \sigma_\rho}$ and $P[t]_{\geqslant \rho}$ are induced upper sets.

Fact 7.7. For all $\rho, \tau \in S$, $h_{\rho,\tau}$ is an upper set isomorphism.

Proof. This follows directly from Proposition 4.17 because the domain of $h_{\rho,\tau}$ contains only maximal elements. □

Fact 7.8. For all $\rho, \tau \in S$, $h_{\tau,\rho} = h_{\rho,\tau}^{-1}$.

Proof. Let $x \in h_{t,\rho}(\rho \cap \tau)$. Our goal is to prove that $h_{\tau,\rho}(h_{\rho,\tau}(x)) = x$ and that for all $y \in h_{t,\tau}(\rho \cap \tau)$, $h_{\rho,\tau}(h_{\tau,\rho}(y)) = y$. We have the following.

$$h_{\tau,\rho}(h_{\rho,\tau}(x)) = h_{\tau,\rho}(h_{\rho,\tau}(h_{t,\rho}(h_{t,\rho}^{-1}(x)))) = h_{\tau,\rho}(h_{t,\tau}(h_{t,\rho}^{-1}(x))) = h_{t,\rho}(h_{t,\rho}^{-1}(x)) = x,$$

where the first equality follows because a function composed with its inverse is the identity, the second equality follows by definition of $h_{\rho,\tau}$, the third equality follows by definition of $h_{\tau,\rho}$, the fourth equality follows because a function composed with its inverse is the identity. For the reverse direction, we analogously have the following.

$$h_{\rho,\tau}(h_{\tau,\rho}(y)) = h_{\rho,\tau}(h_{\tau,\rho}(h_{t,\tau}(h_{t,\tau}^{-1}(y)))) = h_{\rho,\tau}(h_{t,\rho}(h_{t,\tau}^{-1}(y))) = h_{t,\tau}(h_{t,\tau}^{-1}(y)) = y,$$

□

Fact 7.9. For all $\rho, \tau, \sigma \in S$, $P[\rho], P[\tau], P[\sigma]$ satisfy three-way compatibility.

Proof. For $\rho, \tau \in S$, set $V_{\rho,\tau} := h_{t,\rho}(\rho \cap \tau)$. Without loss of generality, suppose that $x \in V_{\rho,\tau} \cap V_{\rho,\sigma}$. Then $x \in h_{t,\rho}(\rho \cap \tau) \cap h_{t,\rho}(\rho \cap \sigma)$, which implies $h_{t,\rho}^{-1}(x) \in \rho \cap \tau \cap \sigma$ and therefore $x \in h_{t,\rho}(\rho \cap \tau \cap \sigma)$. Therefore, for some $y \in \rho \cap \sigma \cap \tau$, $x = h_{t,\rho}(y)$. Then,

$$h_{\tau,\sigma}(h_{\rho,\tau}(x)) = h_{\tau,\sigma}(h_{\rho,\tau}(h_{t,\rho}(y))) = h_{\tau,\sigma}(h_{t,\tau}(y)) = h_{t,\sigma}(y) = h_{\rho,\sigma}(h_{t,\rho}(y)) = h_{\rho,\sigma}(x).$$

The first equality follows by definition of y, the second follows by definition of $h_{\rho,\tau}$, the third follows by definition of $h_{\tau,\sigma}$, the fourth follows by definition of $h_{\rho,\sigma}$, and the fifth follows by definition of y. □

Fact 7.10. For all $\rho, \tau \in S$, $P[\rho], P[\tau], P[t]$ satisfy three-way compatbility.

Proof. There are two cases to consider. In the first, $x \in P[\rho]_{\geqslant \sigma_\rho} \cap h_{t,\rho}(\tau \cap \rho)$. For each $\tau \in S$, set $h_{\tau,t} := h_{t,\tau}^{-1}$. We want to show that $h_{\tau,t}(h_{\rho,\tau}(x)) = h_{\rho,t}(x)$. We have the following.

$$h_{\rho,t}(x) = h_{t,\rho}^{-1}(x) = h_{t,\tau}^{-1}(h_{\tau,\rho}^{-1}(x)) = h_{t,\tau}^{-1}(h_{\rho,\tau}(x)) = h_{\tau,t}(h_{\rho,\tau}(x)).$$

The first equality follows by definition of $h_{\rho,t}$, the second follows by definition of $h_{t,\rho}$ and because the inverse of composition is the composition of the inverses, the third follows by Fact 7.8 and the final equality follows by definition of $h_{\tau,t}$. This completes the proof of the first case. Next, suppose $x \in \rho \cap \tau$. Then we need to show that $h_{\tau,\rho}(h_{t,\tau}(x)) = h_{t,\rho}(x)$. This follows by definition. □

Fact 7.11. For all $\rho \in S$, $P[\rho]_{\geqslant \sigma_\rho}$ and $P[t]_{\geqslant \rho}$ are induced upper sets.

Proof. This follows directly from Proposition 4.29, since all of the upper sets mentioned in the statement of the fact are primitive upper sets. □

□

7.2.2 Testability of Construction 7.5

Lemma 7.12. *$\mathcal{F}$ is a d-testable sheaf code for $\mathcal{F}_{h_t(\sigma)}$.*

Proof. By Construction 7.5, $S \subseteq P[t]$ is a 1-testable cover for $\mathcal{F}[t]_\sigma$. By Proposition 4.39, h_t is a sheaf code isomorphism between $\mathcal{F}[t]$ and $\mathcal{F}$ at upper set $P[t]$. Therefore, by Lemma 5.4, $h_t(S)$ is a 1-testable cover for $h_t(\mathcal{F}[t]_\sigma) = \mathcal{F}_{h_t(\sigma)}$. Next, let $\rho \in h_t(S)$. We show that $\mathcal{F}$ contains a $d-1$ testable cover for $\mathcal{F}_\rho$. Set $\alpha := {h_t}^{-1}(\rho) \in S$. Then, $\mathcal{F}[\alpha]$ contains a $d-1$ testable cover for $\mathcal{F}[\alpha]_{h_{t,\alpha}}$. Therefore, by Proposition 4.39 and Corollary 5.5, $h_\alpha(\mathcal{F}[\alpha])$ contains a $d-1$ testable cover for $h_\alpha(\mathcal{F}[\alpha]_{h_{t,\alpha}})$.

$$h_\alpha(\mathcal{F}[\alpha]_{h_{t,\alpha}}) = \mathcal{F}_{h_\alpha(h_{t,\alpha})} = \mathcal{F}_{h_t(\alpha)} = \mathcal{F}_\rho.$$

The first equality follows by Lemma 7.6 and by definition of merged sheaf (Definition 4.36), the second equality follows by definition of α and h_t. By Proposition 4.39, h_α is a sheaf code isomorphism between $\mathcal{F}[\alpha]$ and $\mathcal{F}$, and therefore $h_\alpha(\mathcal{F}[\alpha])$ is a subspace of $\mathcal{F}$, which completes the proof. □

7.2.3 Extendability of Construction 7.5

We prove the following proposition, which will be useful when arguing that Construction 7.5 is κ-extendable.

Proposition 7.13. *Suppose that $P[\rho]_{\geqslant h_{t,\rho}}$ is globally extendable with respect to $\mathcal{F}[\rho]$ for all $\rho \in S$. Let $U \subseteq P[t]$ be a globally extendable upper set with respect to $\mathcal{F}[t]$ and let $K \subseteq S$ be a subset. Then, $h_t(U)$ is extendable to*

$$h_t(P[t]) \cup \bigcup_{\rho \in K} h_\rho(P[\rho])$$

Proof. Since U is extendable to $P[t]$ with respect to $\mathcal{F}[t]$, it follows by Lemma 4.35 that $h_t(U)$ is extendable to $h_t(P[t])$ with respect to $\mathcal{F}$. Next, we show the following.

Claim 7.14. *For each $\rho \in S$, $h_t(P[t])$ is extendable to $h_t(P[t]) \cup h_\rho(P[\rho])$.*

Proof. By Construction 7.5, $h_t(P[t]) \cap h_\rho(P[\rho]) = h_t(P[t]_{\geqslant \rho}) = h_\rho(P[\rho]_{\geqslant h_{t,\rho}})$, where both equalities follow by definition of h_t, h_ρ. By assumption, $P[\rho]_{\geqslant h_{t,\rho}}$ is globally extendable with respect to $\mathcal{F}[\rho]$. Therefore by Lemma 4.35, $h_\rho(P[\rho]_{\geqslant h_{t,\rho}})$ is extendable to $h_\rho(P[\rho])$ with respect to $h_\rho(\mathcal{F}[\rho])$. Therefore by Lemma 6.15, since $h_t(P[t]) \cap h_\rho(P[\rho])$ is extendable to $h_\rho(P[\rho])$, it follows that $h_t(P[t])$ is extendable to $h_t(P[t]) \cup h_\rho(P[\rho])$. □

Furthermore, by definition of h_ρ and h_t, for each $\rho_1, \rho_2 \in S$ with $\rho_1 \neq \rho_2$,

$$(h_{\rho_1}(P[\rho_1]) \setminus h_t(P[t])) \cap (h_{\rho_2}(P[\rho_2]) \setminus h_t(P[t])) = \emptyset.$$

Therefore, applying Lemma 6.21, with Claim 7.14, it follows that $h_t(P[t])$ is extendable to $h_t(P[t]) \cup \bigcup_{\rho \in K} h_\rho(P[\rho]) = h_t(P[t]) \cup \bigcup_{\rho \in K} h_\rho(P[\rho])$. Applying Lemma 6.23 (transitivity of extendability) completes the proof. □

Corollary 7.15. *Suppose that $P[\rho]_{\geqslant h_{t,\rho}}$ is globally extendable with respect to $\mathcal{F}[\rho]$ for all $\rho \in S$. Let $U \subseteq P[t]$ be a globally extendable upper set with respect to $\mathcal{F}[t]$. Then $h_t(U)$ is globally extendable with respect to $\mathcal{F}$.*

Proof. By definition of merged sheaf and Construction 7.5, $P = \bigcup_{\rho \in S} h_\rho(P[\rho]) \cup h_t(P[t])$. Therefore, applying Proposition 7.13 completes the proof. □

The following shows that if S is a κ-extendable cover for input $\mathcal{F}[t]$, and if $P[\rho]_{\geqslant h_{t,\rho}}$ is extendable in $\mathcal{F}[\rho]$ for each $\rho \in S$, then the output sheaf code, $\mathcal{F}$, has a κ-extendable cover, where each code in the cover consists of the codes in S, along with their $d-1$ testable sheaf codes.

Theorem 7.16 (κ-extendable covers in $\mathcal{F}$)**.** *Suppose that for all $c \in \mathcal{F}[t]_\sigma$, S is a κ-extendable cover for $\mathcal{F}[t]_{\sigma=c}$. Furthermore, suppose that for all $\rho \in S$, $P[\rho]_{\geqslant h_{t,\rho}}$ is globally extendable with respect to $\mathcal{F}[\rho]$. Then for all $c \in C$, $\{g_\rho(P[\rho]_*) : \rho \in S\}$ is a κ-extendable cover for $\mathcal{F}_{h_t(\sigma)=h_t(c)}$.*

Proof. Let $K \subseteq S$ such that $|K| \leq \kappa$. Define U_K as follows.

$$U_K := \bigcup_{\rho \in K} h_\rho(P[\rho]).$$

To complete the proof, we will apply Lemma 6.13, which requires us to show that $U_K \cup P_{\geqslant h_t(\sigma)}$ is globally extendable with respect to $\mathcal{F}$. First, by Construction 7.5, $P = U_K \cup h_t(P[t]) \cup U_{K^c}$. Let $U_K^* := U_K \cup P_{\geqslant h_t(\sigma)}$. We prove the following.

Fact 7.17.

$$U_K^* \cap (h_t(P[t]) \cup U_{K^c}) = \bigcup_{\rho \in K} h_t(P[t]_\rho) \cup P_{h_t(\sigma)}$$

Let $V := \bigcup_{\rho \in K} P[t]_\rho \cup P[t]_\sigma$. By Definition 3.7,

$$\bigcup_{\rho \in K} h_t(P[t]_\rho) \cup P_{h_t(\sigma)} = \bigcup_{\rho \in K} h_t(P[t]_\rho) \cup h_t(P[t]_\sigma) = h_t(\bigcup_{\rho \in K} P[t]_\rho \cup P[t]_\sigma) = h_t(V)$$

By assumption of the lemma, S is a κ-extendable cover for $\mathcal{F}[t]_{\sigma=c}$. Therefore, since $|K| \leq \kappa$, it follows that V is globally extendable with respect to $\mathcal{F}[t]$. Therefore, by Proposition 7.13, $h_t(V)$ is extendable to $h_t(P[t]) \cup U_{K^c}$. Therefore, by Lemma 6.15, U_K^* is extendable to $U_K^* \cup h_t(P[t]) \cup U_{K^c}$. Since $U_K \cup U_{K^c}$ contains all codes greater than or equal to $\{h_\rho(P[\rho])$ for each $\rho \in S$ it follows that $U_K^* \cup h_t(P[t]) \cup U_{K^c} = P$. This completes the proof. □

7.3 d-Testable Sheaf Codes with κ-Extendable Covers From Recursively Testable Codes

In the following construction, we build a d-testable sheaf code via recursive calls to Construction 7.5, where the inputs to each call is dictated by the next code in a "recursively testable code family" (Definition 7.18). This sets us up for our ultimate goal: an explicit perfect zero-knowledge PCPP, which we will obtain by instantiating Construction 7.19 with a well-chosen family of Reed-Solomon codes.

Definition 7.18 (Recursively testable code family)**.** *Let $n \in \mathbb{N}$. A code family (Definition 3.12), $\{C_i\}_{i \geq 1}$, is recursively testable if for all $i > 1$, there exists a testable coded space and sheaf, $P'[i], \mathcal{F}'[i]$ for C_i with testable cover S_i such that for all $i \in [n]$ and $\rho \in S_i$, there exists a sheaf code isomorphism between $\mathcal{F}'[i]$ and $C_{j(\rho)}$ at upper set $P'[i]_{\geqslant \rho}$ for some $j(\rho) < i$.*

Construction 7.19. Let $\{C_i\}_{i \geq 1}$ be a recursive testable code family, with associated sheafs $\{\mathcal{F}'[i] : i \geq 1\}$. Let $n, d \geq 1$. We construct a d-testable sheaf code for C_n, which we denote by $\mathcal{F}_d[n]$, as follows.

- $\mathcal{F}_1[n] = \mathcal{F}'[n]$, $P_1[n] = P'[n]$.
- For $d > 1$, for each $\rho \in S$, let $h_\rho : P'[n]_{\geqslant \rho} \to P'[j(\rho)]_{\geqslant \sigma_\rho}$ be a sheaf code isomorphism between $\mathcal{F}'[n]$ and $\mathcal{F}'[j(\rho)]$ at upper set $P'[n]_{\geqslant \rho}$ for some $\sigma_\rho \in P'[j(\rho)]$ where $\mathcal{F}'[j(\rho)]_{\sigma_\rho} = C_{j(\rho)}$. Let $\sigma \in P'[n]$ such that $\mathcal{F}'[n]_\sigma = C_n$. Then, $\mathcal{F}_d[n]$ is the output of Construction 7.5 on inputs $\sigma, \mathcal{F}'[n], S_n$, and
$$\{(\mathcal{F}_{d-1}[j(\rho)], h_\rho) : \rho \in S_n\}$$

Lemma 7.20 (Construction 7.19 is Correct)**.** *The output of Construction 7.19 is well defined.*

Proof. Proof by induction. When $d := 1$, the output is just $\mathcal{F}'[n]$. Next, let $d > 1$. By construction, $h_{t,\rho}(\rho)$ is a sheaf code isomorphism at $P'[n]_{\geqslant \rho}$ and by inductive hypothesis and Construction 7.5, there is a sheaf code isomorphism $g : P'[j(\rho)]_{\geqslant \sigma_\rho} \to P_{d-1}[j(\rho)]_{\geqslant h_t[d-1,j(\rho)](\sigma_\rho)}$. Therefore, it follows by Proposition 4.33 (composition of sheaf isomorphisms) that there is a sheaf code isomorphism between $\mathcal{F}'[n]$ and $\mathcal{F}_{d-1}[j(\rho)]$ on $P'[n]_{\geqslant \rho}$. □

For the remainder of the section, let $h_t[d,n]$ be the sheaf code isomorpshim between $\mathcal{F}'[n]$ and $\mathcal{F}_d[n]$ on upper set $P'[n]$ as defined in Construction 7.5.

Proposition 7.21. *Let $n, d \geq 1$. If for all $j \leq n$, $P'[j]_{\geqslant \alpha_j}$ is globally extendable with respect to $\mathcal{F}'[j]$, then $P_d[n]_{\geqslant h_t[d,n](\alpha_n)}$ is globally extendable with respect $\mathcal{F}_d[n]$.*

Proof. Proof by induction. When $d = 1$, $\mathcal{F}'[n] = \mathcal{F}_d[n]$ and the proof follows trivially. Next, when $d > 1$, since $j_\rho < n$, we can apply the inductive hypothesis and conclude that for all $\rho \in S_n$, $h_t[d-1, j_\rho](P'[j_\rho]_{\geqslant \alpha_{j_\rho}})$ is globally extendable with respect $\mathcal{F}_{d-1}[j_\rho]$. Therefore, by Proposition 7.13, since $P'[n]_{\geqslant \alpha_n}$ is globally extendable with respect to $\mathcal{F}'[n]$, it follows that $h_t[d,n](P'[n]_{\geqslant \alpha_n}) = P_d[n]_{\geqslant h_t[d,n](\alpha_n)}$ is globally extendable with respect to $\mathcal{F}_d[n]$, which completes the proof. □

Lemma 7.22 ($\mathcal{F}_{d,n}$ Contains A d-Testable Cover For C_n)**.** *Let $d, n \in \mathbb{N}$ and let $\{C_i\}_{i\geq 1}$ be a recursively testable code family. Then for every integer $n \geq 1$, $\mathcal{F}_d[n]$ contains a d-testable cover for C_n.*

Proof. By Lemma 7.20, $\mathcal{F}_d[n]$ is an output of Construction 7.5. Therefore the proof follows directly from Lemma 7.12 and by Construction 7.19. □

Lemma 7.23 (κ-extendable covers for $\mathcal{F}_{d,i}$)**.** *Let $\kappa, d, n \in \mathbb{N}$ and let $\{C_i\}_{i\geq 1}$ be a recursively testable code family. For all $n \geq 1$, if for all $c \in C_n$, S_n is a κ extendable cover for $(\mathcal{F}'[n])_{\alpha_n = c}$, then $(\mathcal{F}_d[n])_{h_t(\alpha_n)=h_t(c)}$ has a κ-extendable cover.*

Proof. Fix $g := h_t[d,n]$. When $d = 1$, the statement follows trivially. Otherwise, we apply Theorem 7.16, which requires us to prove that for all $c \in C_n$, S is a κ extendable cover in $(\mathcal{F}'[n])_{\alpha_n = c}$ (which we know by assumption of the lemma), and that for each $\rho \in S$, $P_{d-1}[j_\rho]_{\geqslant g(\alpha_{j_\rho})}$ is globally extendable with respect to $\mathcal{F}_{d-1}[j_\rho]$. This follows from Proposition 7.21. □

8 Short, Perfect Zero-Knowledge PCPP For Reed-Solomon Codes

8.1 A Useful Generalization Of The BS08 Bivariate Reed-Muller Code

In this section, $\mathbb{F} = \mathbb{F}_{2^l}$ is a binary field, which we can view as a vector space over $\mathbb{F}_2$ of dimension l. For $L \subseteq \mathbb{F}$, denote by q_L the polynomial of minimal degree that vanishes on L, i.e.,

$$q_L(z) := \prod_{\alpha \in L} (z - \alpha).$$

Definition 8.1. *Let $k, \hat{k} \in \mathbb{N}$, such that k is even, $k < l$, and $k/2 + \hat{k} + 2 < k$. For a subspace $L \subseteq \mathbb{F}$ of dimension k, fix a canonical basis $b_1, \ldots, b_k$ of L, and define*

$$D_{\hat{k}}(L) := \bigcup_{\beta \in L_1} (L_\beta \times \{q_{L_0}(\beta)\}) \cup (L'_0 \times q_{L_0}(L)) \subseteq L \times q_{L_0}(L)$$

where L_β, L_1, L_0 are defined as follows:

$$L_0 := \mathrm{span}(b_1, .., b_{k/2}) \qquad L_1 := \mathrm{span}(b_{k/2+1}, .., b_k) \setminus L_R$$
$$L_R := \mathrm{span}(b_{k/2+1}, .., b_{k/2+\hat{k}+2}) \qquad L_\beta = \mathrm{span}(L'_0, \beta) \quad \forall \beta \in L_1$$
$$L'_0 := L_0 + L_R$$

Define the coded space $P_{\hat{k}}(L) := D_{\hat{k}}(L) \cup \mathcal{R} \cup \mathcal{K} \cup \{T\}$, consisting of

- *rows $\mathcal{R} := \{K_x : x \in L'_0\}$, where $K_x = \{x\} \times \{y : (x,y) \in D_{\hat{k}}(L)\}$*
- *columns $\mathcal{K} := \{K_y : y \in q_{L_0}(L_1)\}$, where $K_y = \{x : (x,y) \in D_{\hat{k}}(L)\} \times \{y\}$;*

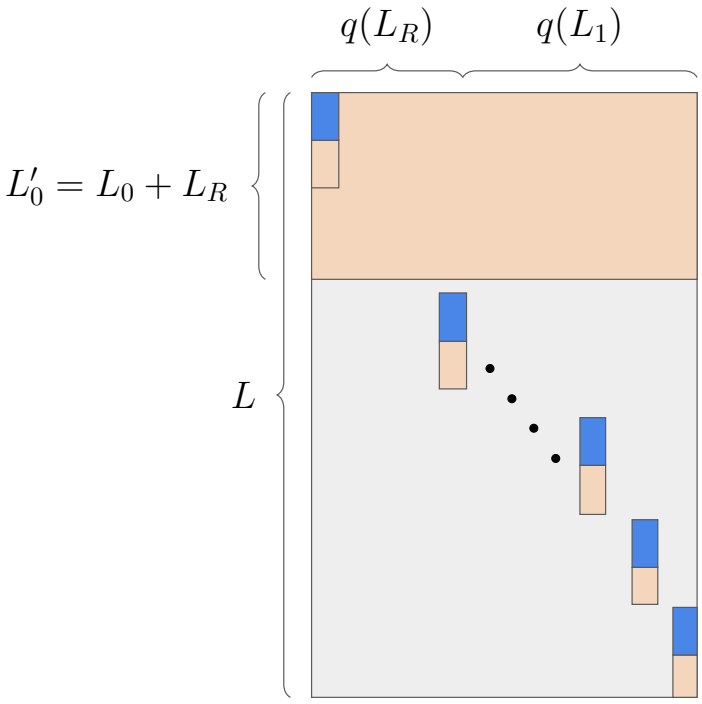


Figure 2: The domain $D_{\hat{k}}(L)$, with the subspace curve highlighted in blue. Only the rows in L'_0 and the columns in $q(L_1)$ are included.

- *a curve* $T \coloneqq \{(x, q_{L_0}(x)) : x \in L\}$.

Finally, for degrees $d_X, d_Y \in \mathbb{N}$ *where* $d_X + 1$ *is a power of two, define the sheaf of linear codes* $\mathcal{F} = \mathcal{F}_{d_X,d_Y}(L)$ *on* $P_{\hat{k}}(L)$ *where* $d_X + 1 = 2^{\hat{k}+k/2}$*:*

- *for* $p \in D_{\hat{k}}(L)$, $\mathcal{F}_p = \mathbb{F}$*;*
- *for* $R_x \in \mathcal{R}$, $\mathcal{F}_{R_x} = \mathrm{RS}[\mathbb{F}, R_x, d_Y]$*;*
- *for* $K_y \in \mathcal{K}$, $\mathcal{F}_{K_y} = \mathrm{RS}[\mathbb{F}, K_y, d_X]$*;*
- *and* $\mathcal{F}_T = \{c \in \mathbb{F}^T : (c(x, q_{L_0}(x))_{x \in L} \in \mathrm{RS}[\mathbb{F}, L, d_X + d_Y \cdot |L_0|]\}$.

Remark 8.2. When $\hat{k} = 0$ we recover the bivariate Reed-Muller code from [BS08].

Figure 2 depicts the domain $D_{\hat{k}}(L)$ in orange, with the subspace curve highlighted in blue.

8.2 κ-Extendable Covers In Bivariate Reed-Muller Sheaf Codes

Theorem 8.3. *Let* $L \subseteq \mathbb{F}$ *be a subspace, let* $d_X, d_Y \in \mathbb{N}$ *such that* $d_X \geq |L_0| + d_Y$, $d_Y \geq 1$*, and let* $\mathcal{F} := \mathcal{F}_{d_X,d_Y}(L)$ *be a bivariate Reed-Muller sheaf code (Definition 8.1). For all* $c \in \mathcal{F}_T$, $\mathcal{R} \cup \mathcal{K}$ *is a* d_Y*-extendable cover for* $\mathcal{F}_{T=c}$*.*

We prove Theorem 8.3 using results about extendability from Section 6.3. To build intuition, we start with a high-level overview that uses only elementary facts about bivariate polynomials. Then, we describe the same argument through the more general lens of upper set extendability, and continue with the formal proof. Figures 3 depicts the former viewpoint and Figure 4 depicts the latter.

Overview of the proof of Theorem 8.3 in the language of bivariate polynomials. Let $S \subseteq \mathcal{R} \cup \mathcal{K}$ be a collection of rows and columns in P such that $|S| < d_Y$. Let $S' := S \cup \{T\}$. Let $D_{S'} = U^*_{S'} \subseteq \mathbb{F} \times \mathbb{F}$ be the maximal set of $U_{S'}$, i.e. the union of the domains of the local codes in S'. We need to prove

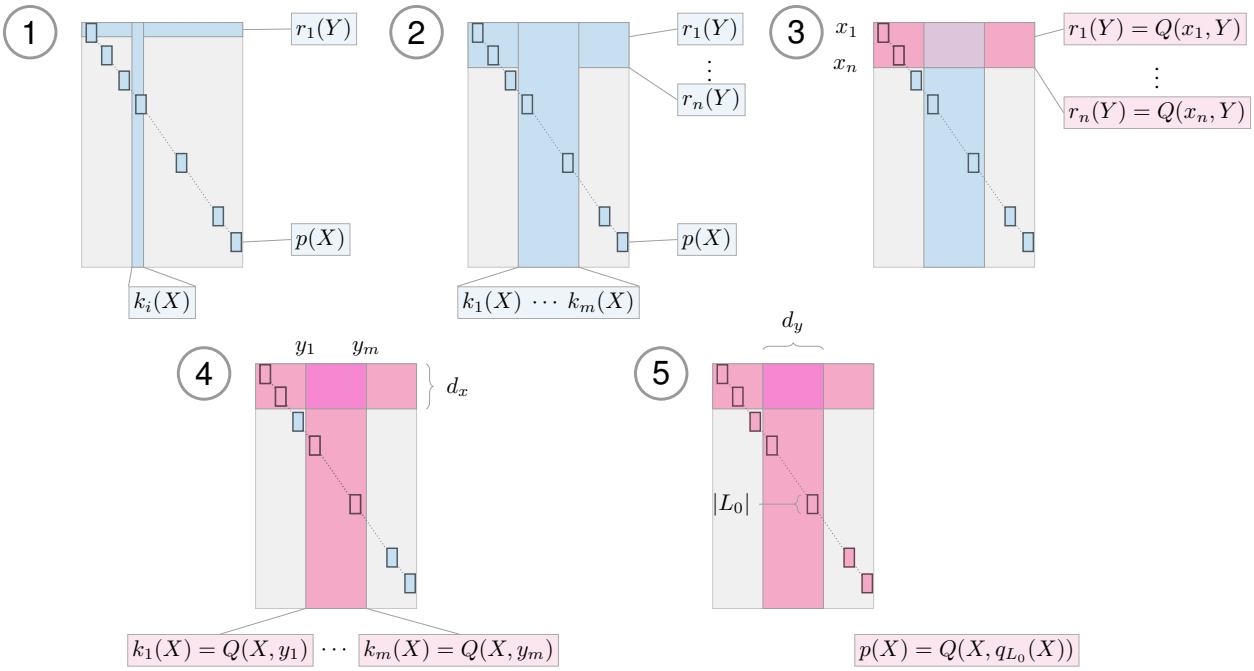


Figure 3: This image depicts the proof strategy described in Section 8.2, which uses polynomial interpolation. In Frame (1), we are given a function on a small domain consisting of one row, one column and the subspace curve. The function consists of a d_X univariate polynomial on the column, a d_Y univariate polynomial on the row, and a degree-Δ univariate polynomial on T. In Frame (2), we extend this domain, and assign to each new column a univariate polynomials of degree $\leq d_X$ and to each new row a univariate polynomials of degree $\leq d_Y$. In Frame (3), we find $Q \in \mathbb{F}^{\leq [d_X, d_Y]}[X, Y]$ that agrees with evaluations on the rows. Because each column is fully determined by the values in the rows, the columnns must also agree with Q, which is depicted in Frame (4). Then, because there are $\Delta + 1$ points in T which are determined by the rows and columns in Frame (4), the degree-Δ polynomial on T must also agree with Q, which is depicted in Frame (5).

that $U_{S'}$ is globally extendable with respect to $\mathcal{F}$. Since $\mathcal{F}(P)$ is a bivariate Reed-Muller sheaf code (Definition 8.1), this translates to proving that for any $f \in \mathcal{F}(U_{S'})$, there exists a bivariate polynomial $Q \in \mathbb{F}^{\leq [d_X, d_Y]}[X, Y]$ such that $Q(x, y) = f(x, y)$ for all $(x, y) \in D_{S'}$.

We start by considering some collection $W \subseteq P$ such that $S \subseteq W$, where W consists of exactly $d_X + 1$ rows, d_Y columns, and the curve T. Let $D_W = P_{U_W}$ be the maximal elements of U_W. Let R' be the rows and let K' be the columns in W. Let $D_{R'} = P_{U_{R'}}, D_{K'} = P_{U_{K'}}$ be the maximal elements of upper sets $U_{R'}$ and $U_{K'}$ respectively, and let $f \in \mathcal{F}(U_{R'})$. First, we find a bivariate polynomial $Q \in \mathbb{F}^{\leq [d_X, d_Y]}[X, Y]$ satisfying $Q|_{D_{R'}} = f|_{D_{R'}}$, which is proven to exist in Lemma H.1 in [BCFGRS17]. We will argue next that $Q|_{D_W} = f|_{D_W}$. We do this in two steps. First, $Q|_{D_{K'}}$ is equal to $f|_{D_{K'}}$ because for each column $\rho \in K$, $Q|_\rho$ and $f|_{P_\rho}$ are univariate polynomials of degree $\leq d_X$ and $Q|_{P_\rho}$ agrees with $f|_{P_\rho}$ on each of the $d_X + 1$ points in $P_\rho \cap D_{R'}$. By a similar argument, $Q|_T$ is equal to $f|_T$ because $f|_T$ and $Q|_T$ are both univariate polynomials of degree $\leq \Delta = d_X + d_Y \cdot |L_0|$ and $Q|_T$ agrees with $f|_T$ on each of the $\Delta + 1$ points in $T \cap (D_{K'} \cup D_{R'})$. Putting these two facts together, Q agrees with f on all of D_W.

Finally, we are ready to reason about the global extendability of $U_{S'}$. Let $f \in \mathcal{F}(U_{S'})$. Fix a codeword $g \in \mathcal{F}(W)$ such that $g|_{D_{S'}} = f|_{D_{S'}}$. We use the argument from the previous paragraph to find a bivariate polynomial $Q \in \mathbb{F}^{\leq [d_X, d_Y]}[X, Y]$ such that $Q|_{D_W} = g|_{D_W}$. Since $D_{S'} \subseteq D_W$, it follows that $Q|_{D_{S'}} = g|_{D_{S'}}$, which by definition is equal to $f|_{D_{S'}}$. This completes the proof.

Proof of Theorem 8.3. We are now ready for the formal proof of Theorem 8.3. Using the notation

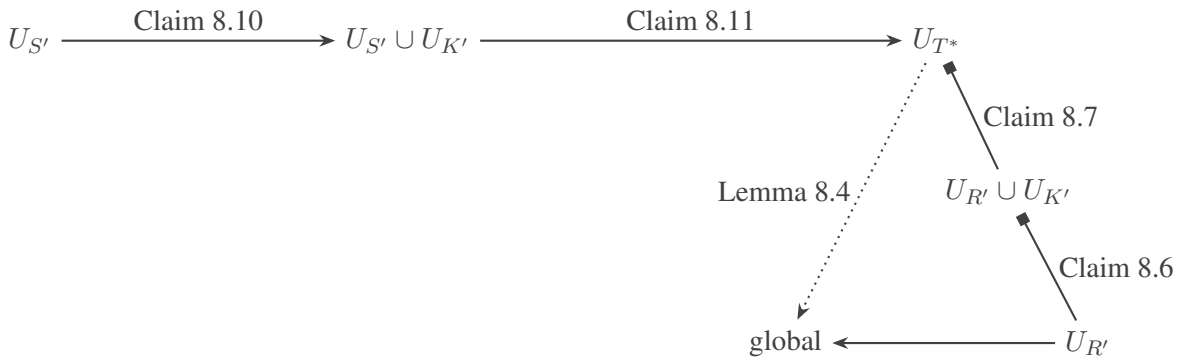


Figure 4: A diagram depicting the proof strategy of Theorem 8.3 using results from Section 6.3. Standard arrows denote extendability and square arrows denote unique extendability. $U_{S'}$ corresponds to Frame (1) in Figure 1. U_{T^*} refers to Frame (2) and the triangle describes the argument outlined by Frames (3), (4), and (5) in Figure 1.

of the previous subsection, we start by proving that U_W is globally extendable. We observe that the above argument from bivariate polynomial interpolation is an instance of more abstract extendability properties as established by Lemma 6.22 ("triangle of extendability") and Lemma 6.23 ("transitivity of extendability"). Lemma 6.22 tells us that *if* $U_{R'}$ is uniquely extendable to U_W and is itself globally extendable (which Lemma H.1 of [BCFGRS17] proves it is), then U_{T^*} must also be globally extendable. Similarly, Lemma 6.23 tells us that if U_W is globally extendable, and $U_{S'}$ is extendable to U_W, then $U_{S'}$ is globally extendable. Finally, we can prove the extendability of $U_{R'}$ and $U_{S'}$ to U_W by applying Lemmas 6.24 and 6.17 and then counting intersections of the different local domains.

Proof. Let $S \subseteq \mathcal{R} \cup \mathcal{K}$, $|S| \leq d_Y$, and let $S' := S \cup \{T\}$. To prove the theorem, it suffices by Lemma 6.13 to prove that $U_{S'}$ is globally extendable (Definition 4.10) with respect to $\mathcal{F}$. Let $R := S \cap \mathcal{R}$, $K := S \cap \mathcal{K}$ be the rows and columns in S, respectively. Choose any sets $R' \subseteq \mathcal{R}$, $K' \subseteq \mathcal{K}$ such that $R \subseteq R'$, $|R'| = d_X + 1$, $K \subseteq K'$, $|K'| = d_Y$. Let $W := K' \cup R' \cup T$. We prove the following two lemmas.

Lemma 8.4. *U_W is globally extendable.*

Lemma 8.5. *$U_{S'}$ is extendable to U_W.*

By transitivity of extendability (Lemma 6.23), Lemmas 8.4 and 8.5 together imply that $U_{S'}$ is globally extendable, which establishes the theorem. □

Next, we present the proofs of these two lemmas.

Proof of Lemma 8.4. This lemma is a consequence of the following two claims, proved below.

Claim 8.6. *$U_{R'}$ is uniquely extendable to $U_{R' \cup K'}$.*

Claim 8.7. *$U_{R' \cup K'}$ is uniquely extendable to U_W.*

By transitivity of unique extendability (Lemma 6.23), it follows from Claims 8.6 and 8.7 that $U_{R'}$ is uniquely extendable to U_W. Furthermore, by Lemma H.1 in [BCFGRS17], $U_{R'}$ is globally extendable. Therefore, by Lemma 6.22 (triangle of extendability), it follows that U_W is globally extendable. □

Next we prove Claims 8.6 and 8.7. Each claim is proven by decomposing one of the upper sets and analyzing the relevant intersections. We present the statements about intersections as facts whose proofs we defer to Appendix B.

Proof of Claim 8.6.

Fact 8.8. For all $y \in K'$,

$$|K_y \cap P_{U_{R'}}| = \dim(\mathcal{F}_{K_y}).$$

By Fact 8.8 and by Lemma 6.24, since $\mathcal{F}_{K_y}$ is MDS, $U_{R'}$ is uniquely extendable to $U_{R'} \cup P_{\geqslant K_y}$ for each $y \in K'$. Furthermore, for each $y_1 \neq y_2 \in K'$, $K_{y_1} \cap K_{y_2} = \emptyset$. Therefore, by Lemma 6.17, $U_{R'}$ is uniquely extendable to $U_{R'} \cup U_{K'} = U_{R' \cup K'}$. □

Proof of Claim 8.7. Recall that $U_W = U_{R' \cup K'} \cup P_{\geqslant T}$. We use the following fact.

Fact 8.9.

$$|T \cap P_{U_{R' \cup K'}}| = \dim(\mathcal{F}_T)$$

By Lemma 6.24, since $\mathcal{F}_T$ is MDS, $U_{R'} \cup U_{K'}$ is uniquely extendable to U_W. □

We now turn to the proof of Lemma 8.5.

Proof of Lemma 8.5. The lemma follows from the following two claims by transitivity (Lemma 6.23).

Claim 8.10. *$U_{S'}$ is extendable to $U_{S'} \cup U_{K'}$.*

Claim 8.11. *$U_{S'} \cup U_{K'}$ is extendable to U_W.* □

Next, we prove the claims.

Proof of Claim 8.10. We use the following fact.

Fact 8.12. For all $y \in K' \setminus K$,

$$|K_y \cap P_{U_{S'}}| \leq \dim(\mathcal{F}_{K_y}).$$

By Fact 8.12 and by Lemma 6.24, since $\mathcal{F}_{K_y}$ is MDS, $U_{S'}$ is extendable to $U_{S'} \cup P_{\geqslant K_y}$ for each $y \in K'$. Furthermore, for each $y_1, y_2 \in K' \setminus K$, $K_{y_1} \cap K_{y_2} = \emptyset$. Therefore, by Lemma 6.17, $U_{S'}$ is extendable to $U_{S'} \cup U_{K'}$. □

Proof of Claim 8.11. By definition, $U_W = U_{S'} \cup U_{K'} \cup U_{R' \setminus R}$. We use the following fact.

Fact 8.13. For all $x \in R' \setminus R$,

$$|R_x \cap P_{U_{S'} \cup U_{K'}}| = \dim(\mathcal{F}_{R_x}).$$

Therefore, by Fact 8.13 and by Lemma 6.24, $U_{S'} \cup U_{K'}$ is extendable to $U_{S'} \cup U_{K'} \cup P_{\geqslant R_x}$ for each $x \in R'$. Furthermore, for each $x_1, x_2 \in R'$, $R_{x_1} \cap R_{x_2} = \emptyset$. Therefore, by Lemma 6.17, $U_{S'} \cup U_{K'}$ is extendable to U_W. □

8.3 Testing Reed-Solomon Codes With Bivariate Reed-Muller Sheaf Codes

In this section, we construct a recursively testable code family (Definition 7.18) for Reed-Solomon codes, using the bivariate Reed-Muller codes described in Section 8.1. For the target code of the family, C_n, we choose the degree bounds so that $(\mathcal{F}_n)_{\sigma=c}$ has a κ-extendable cover for any $c \in C_n$ so that the final PCPP satisfies perfect zero-knowledge. In particular, we choose degree bounds less than or equal to $(2|L_0| - 1, |L_0|/8 - 2)$ that satisfy the conditions of Theorem 8.3, i.e. $(2|L_0| - 1) \geq (|L_0|/8 - 2) + |L_0|$. We start by recalling Definition 7.9 from [BCGRS17].

Definition 8.14. *We denote by RS^+ the linear code family indexed by tuples*

$$n = (\mathbb{F}, L, d),$$

where $\mathbb{F}$ is an extension field of $\mathbb{F}_2$ and L is an $\mathbb{F}_2$-linear subspace of $\mathbb{F}$ with

$$d \leq |L|/8,$$

and the n-th code consists of the codewords from the Reed–Solomon code

$$\mathrm{RS}[\mathbb{F}, L, d].$$

Theorem 8.15. RS^+ *is recursively testable.*

Proof. Let $\hat{k} \geq 0$, $i \geq 1$, and let $(\mathbb{F}, L, d)$ be the ith tuple in RS^+. Let $\mathcal{F}'[i]$ be the Reed-Muller sheaf code defined in Definition 8.1 over domain $D_{\hat{k}}(L_i)$. First, we show that $\mathcal{F}'[i]$ has a 1-testable cover for $C_i = RS[\mathbb{F}, L, d]$. Let $P'[i]$ be the coded space for $\mathcal{F}'[i]$. We show the following.

Lemma 8.16. *For each $i \in [n]$, $\mathcal{R} \cup \mathcal{K}$ is a 1-testable cover for C_i.*

The proof of Lemma 8.16 closely follows the soundness proof of the short PCPP from [BS08] so we defer its proof to Appendix C. Next, by definition of bivariate Reed-Muller sheaf code (Definition 8.1), for each $\rho \in \mathcal{R} \cup \mathcal{K}$, $\mathcal{F}'[i]_\rho$ is equal (up to permutation) to $RS[\mathbb{F}, L_j, d_j]$ for some $j < i$. This completes the proof. □

8.4 Our Construction

We build our construction using the framework outlined in Section 7.

Construction 8.17 (Perfect Zero-Knowledge PCPP for Reed–Solomon Codes)**.** Let $\mathcal{C} = \mathrm{RS}^+$ be the Reed-Solomon code family (Definition 8.14) and let $C_i = \mathrm{RS}[\mathbb{F}, L, |L|/8 - 1]$. Let $\mathcal{F}'[i]$ be the bivariate Reed-Muller sheaf code over coded space $P'[i]$ (Definition 8.1) over domain $D_1(L)$ such that $P'[i]_T = RS[\mathbb{F}, L, |L|/8 - 1]$. Let $\mathcal{F}, P$ be the output of Construction 7.19 when given code family $\mathcal{C}$, code index i, and recursive depth $\log(\log(|L|))$ as input.

- **Prover:**
  - *Input:* Codeword $c \in \mathrm{RS}[\mathbb{F}, L, |L|/8 - 1]$
  - *Output:* The prover output from Construction 7.3 on inputs $\mathcal{F}, P, c$.
- **Verifier:**
  - *Input:* Impicit input $f : \mathbb{F}^{P_\sigma} \to \mathbb{F}$ and explicit input $g : \mathbb{F}^{P_*} \to \mathbb{F}$
  - *Output:* The verifier output from Construction 7.3 on inputs $\mathcal{F}, P, f, g, T, \mathcal{R} \cup \mathcal{K}$

Next, we use the above construction to prove the existence of a perfect-zero knowledge PCPP for Reed-Solomon codes.

Theorem 8.18. *For all $\epsilon, \delta \in [0, 1]$, there exists a black-box, $\sqrt{n}$-local, perfect zero-knowledge PCPP for RS^+ with proof length $\widetilde{O}(N)$, verifier query complexity $\mathrm{polylog}(n)$, zero-knowledge query bound $\sqrt{n}/8 - 2$, soundness error δ, and proximity parameter ϵ.*

Proof. We start by showing that Construction 8.17 is black-box, $\sqrt{n}$-local, perfect zero-knowledge PCPP with verifier query complexity $O(1)$, zero-knowledge query bound $\sqrt{n}/8 - 2$, soundness error $\epsilon/\mathrm{polylog}(n)$ and proximity parameter ϵ. By Theorem 8.15, $\mathcal{C}$ is a recursive 1-testable code family. Therefore, since P is the output of Construction 7.19, it follows by Lemma 7.22 that P contains a d-testable cover for $RS[\mathbb{F}, L, |L|/8]$. Next, we apply Theorem 7.4, which just requires us to prove that $\mathcal{F}'[i]$ has a $(\sqrt{n}/8 - 2)$-extendable cover. By Lemma 7.23, this follows as long as S_i is a $(|L_0|/8 - 2)$-extendable cover for $\mathcal{F}_i$. This follows from Theorem 8.3 because by definition of our bivariate Reed-Muller sheaf code (Definition 8.1), $\log_2(d_X + 1) = \hat{k} + \log 2(|L|)/2 = \log_2(|L|)/2 + 1$ and so $d_X = 2^{\log_2(|L|)/2+1} = 2 \cdot |L_0| - 1$. Definition 8.1 also states that $\mathcal{F}'[i]|_T$ has degree bound $d_X + d_Y \cdot |L_0|$. In our case $P'[i]_T$ has degree bound $|L|/8 - 1$. This implies that

$$d_Y = (|L|/8 - 1 - d_X)/|L_0| = |L_0|/8 - 1/|L_0| - d_X/|L_0| = |L_0|/8 - 1/|L_0| - 2 + 1/|L_0| = |L_0|/8 - 2.$$

We show that d_X, d_Y satisfy the conditions of Theorem 8.3 with the following.

$$d_X = 2|L_0| - 1 \geq 9|L_0|/8 - 2 = |L_0| + d_Y$$

By Proposition 2.9 from [BS08], this implies that for any $\epsilon, \delta \in [0,1]$, we can obtain a PCPP with soundness error δ, proximity parameter ϵ and equivalent proof length to Construction 8.17 and query complexity

$$O\left(\frac{\log(1/\delta)}{(\epsilon/\mathrm{polylog}(n))}\right) = \mathrm{polylog}(n).$$

Furthermore, in the recursively testable code family of Reed-Solomon codes from Theorem 8.15, each sheaf that tests the Reed-Solomon code is the size of the BS08 domain for that Reed-Solomon code up to a constant. Furthermore, we use the same number of recursive rounds as BS08. Therefore, the proof length will also be the length of BS08 up to a constant, which completes the proof. □

9 Perfect Zero-Knowledge PCP For NTIME

In this section, we build a black-box perfect zero-knowledge PCP for NTIME using the zero-knowledge PCPP for Reed-Solomon codes from the previous section. Our argument follows that of section 8 in [BCFGRS17], which builds a perfect zero-knowledge k-round Interactive Oracle Proof for NTIME from a k-round perfect zero-knowledge Interactive Oracle Proof of Proximity for Reed–Solomon. Interactive Oracle Proofs (of Proximity) are generalizations of PCPs (resp. PCPPs), where the prover and verifier engage in $k \in \mathbb{N}$ rounds of interaction. Setting $k := 0$ yields the corresponding result for PCPs.

Another difference between our two settings is that they assume a perfect zero-knowledge IOPP whose simulator can tolerate unbounded verifier queries, and only queries one input symbol for every query to the proof. We modify their argument to handle l-local black-box perfect zero-knowledge PCPPs with an explicit query bound. The following is our main result.

Theorem 9.1 (Main Result)**.** *For every polynomial $b, T\colon \mathbb{N} \to \mathbb{N}$, there exists a perfect zero-knowledge PCP system (P, V) for $\mathsf{NTIME}(T)$ with the parameters:*

$$\begin{bmatrix} \textit{alphabet} & \mathbb{F}(n) = \mathbb{F}_2 \\ \textit{proof length} & \ell(n) = \widetilde{O}\big(T(n) + b(n)^2\big) \\ \textit{query complexity} & q(n) = \mathrm{polylog}(n) \\ \textit{soundness error} & \varepsilon(n) = 1/2 \\ \textit{query bound} & b(n) \end{bmatrix}.$$

Recall that Dinur [Din07] has shown that a PCP of length ℓ with query complexity q can be transformed to a PZK-PCP of length $\ell \cdot \mathrm{poly}(q)$ with query complexity $O(1)$. In Appendix A, we observe that Dinur's query reduction and gap amplification transformations are $\mathrm{poly}(q(n))$-locally computable, hence they preserve perfect zero-knowledge with query bound $b/\mathrm{poly}(q)$. Applying gap amplification (Lemma A.1) to Theorem 9.1, instantiated with query bound $B(n) \cdot \log(n)^c$, for a sufficiently large constant $c > 0$, we obtain the following immediate corollary (note that the additional poly-logarithmic factor is absorbed in the $\tilde{O}$ notation).

Theorem 9.2 (Quasilinear-length PCPs with constant query complexity)**.** *For every polynomial $b, T\colon \mathbb{N} \to \mathbb{N}$, there exists a perfect zero-knowledge PCP system (P, V) for $\mathsf{NTIME}(T)$ with the parameters:*

$$\begin{bmatrix} \textit{alphabet} & \mathbb{F}(n) = \mathbb{F}_2 \\ \textit{proof length} & \ell(n) = \widetilde{O}\big(T(n) + b(n)^2\big) \\ \textit{query complexity} & q(n) = O(1) \\ \textit{soundness error} & \varepsilon(n) = 1/2 \\ \textit{query bound} & b(n) \end{bmatrix}.$$

Proof overview. Our proof closely follows the proof of Theorem 8.1 in [BCFGRS17] closely, which itself follows [BCGV16]. Those works construct perfect zero-knowledge interactive oracle proofs (IOPs)

for (randomizable) linear-algebraic constraint satisfaction problems ((R)LACSPs), which are shown in [BCGV16] to be NP-complete. IOPs are a multi-round generalization of PCPs. We obtain our main result by replacing the two-message zero knowledge IOP of proximity for Reed–Solomon codes used in those works with our ZKPCPP from Theorem 8.18.

Our ZKPCPP does not precisely fit the requirements laid out in [BCFGRS17], and so it is not a drop-in replacement. In particular, the simulator for the ZKPCPP has locality $\sqrt{d}$, where d is the degree of the polynomial being tested; in the prior work it is assumed that the ZKIOPP has locality 1. In addition, the simulator for our ZKPCPP fails after around $\sqrt{n}$ queries, which must be accounted for in the overall query bound. These modifications are minor and do not affect the structure of the proof.

9.1 Linear Algebraic Constraint Satisfaction Problems

We begin by recalling the definitions of LACSP and RLACSP [BCGV16; BCFGRS17].

Definition 9.3. *Let $g\colon \Sigma^n \to \Sigma^m$ be a function. We say that g is ℓ-**local** if for every $j \in [m]$ there exists $I_j \subseteq [i]$ with $|I_j| \leq \ell$ such that $g(w)_j$ depends only on $w|_{I_j}$. Moreover, we say that g is s-**evasive** if for every $I \subseteq [n]$ the probability that, for a uniform $j \leftarrow [m]$, I_j intersects I is at most $s \cdot \frac{|I|}{n}$.*

Definition 9.4 (LACSP)**.** *Let $C_0(n)$ and $C_1(n)$ be descriptions of linear codes over $\mathbb{F}(n)$ with block length $\ell(n)$ and relative distance $\tau(n)$. The promise relation of linear algebraic CSPs (LACSPs)*

$$\mathrm{LACSP}\big[\mathbb{F}(n), C_0(n), C_1(n), \ell(n), \tau(n), q(n)\big] = \big(R^{\mathrm{YES}}_{\mathrm{LACSP}}, L^{\mathrm{NO}}_{\mathrm{LACSP}}\big)$$

considers instance–witness pairs (x, w) of the following form.

- *An instance x is a tuple $(1^n, g)$, where:*
  - *$g\colon \mathbb{F}(n)^{\ell(n)} \to \mathbb{F}(n)^{\ell(n)}$ is $q(n)$-local, $q(n)$-evasive, and can be computed in time $\mathrm{poly}(n)$;*
  - *$C_1(n) \cup g(C_0(n))$ has relative distance at least $\tau(n)$.*
- *A witness w is a tuple $w = (w_0, w_1)$, where $w_0, w_1 \in \mathbb{F}(n)^{\ell(n)}$.*

The yes-relation $R^{\mathrm{YES}}_{\mathrm{LACSP}}$ consists of all pairs (x, w) as above for which the instance x and witness w jointly satisfy $w_0 \in C_0(n), w_1 \in C_1(n)$, and $g(w_0) = w_1$. In particular, a witness $w = (w_0, g(w_0))$, with $w_0 \in C_0(n)$, satisfies x if and only if $g(w_0) \in C_1(n)$. The no-language $L^{\mathrm{NO}}_{\mathrm{LACSP}}$ consists of all instances x as above such that $x \notin L\left(R^{\mathrm{YES}}_{\mathrm{LACSP}}\right)$.

Definition 9.5 (randomizability)**.** *An LACSP instance $(1^n, g)$ is $t(n)$-**randomizable** (with respect to code families $C_0(n), C_1(n)$) if there exists a $t(n)$-wise independent subcode $C' \subseteq C_0(n)$ such that if $(c_0, g(c_0))$ satisfies x then, for every $c'_0 \in c_0 + C'$, the witness $(c'_0, g(c'_0))$ also satisfies x.*

Definition 9.6 (RLACSP)**.** *The promise relation of randomizable linear algebraic CSPs (RLACSPs) is*

$$\mathrm{RLACSP}[\mathbb{F}(n), C_0(n), C_1(n), \ell(n), \tau(n), q(n), c(n), t(n)] = \left(R^{\mathrm{YES}}_{\mathrm{RLACSP}}, L^{\mathrm{NO}}_{\mathrm{RLACSP}}\right),$$

where $R^{\mathrm{YES}}_{\mathrm{RLACSP}}$ is obtained by restricting $R^{\mathrm{YES}}_{\mathrm{LACSP}}$ to instances that are $t(n)$-randomizable, and $L^{\mathrm{NO}}_{\mathrm{RLACSP}} := L^{\mathrm{NO}}_{\mathrm{LACSP}}$.

Theorem 9.7 ([BCGV16])**.** *For any polynomials t, T and any constant $\tau \in (0, 1)$, $\mathrm{RLACSP}[\mathbb{F}, C_0, C_1, \ell, \tau, q, t]$ is $\mathsf{NTIME}(T)$-hard where $\mathbb{F}$ is a sufficiently large binary field, $H_0 \leq L$ are additive subgroups of $\mathbb{F}$, $C_0 = \mathrm{RS}[\mathbb{F}, L, d_0]$, C_1 is the subcode of $\mathrm{RS}[\mathbb{F}, L, d_1]$ vanishing on a coset H of H_0 with $H \cap L = \emptyset$, $d_0, d_1 \leq |L|/8 - 1$, $|L| = \ell = \tilde{O}(T + t)$; and $q = \mathrm{polylog}(T)$.*

9.2 From zkPCPP for Reed-Solomon Codes to zkPCP For NTIME

The following theorem is adapted from [BCFGRS17, Theorem 8.9].

Theorem 9.8. *Suppose that there exists a black-box κ-local perfect zero-knowledge PCPP for the relation $C_0 \times C_1$ with the parameters:*

$$\begin{bmatrix} \textit{alphabet} & \mathbb{F}(n) \\ \textit{proof length} & \widehat{l}(n) \\ \textit{query complexity} & \widehat{q}(n) \\ \textit{soundness error} & \widehat{\varepsilon}(n) \\ \textit{query bound} & \hat{b}(n) \end{bmatrix}.$$

Then there exists a PCP (P_R, V_R), for relation $(\mathcal{R}^{\mathrm{YES}}_{\mathrm{RLACSP}}, \mathcal{L}^{\mathrm{NO}}_{\mathrm{RLACSP}})$ $[\mathbb{F}, C_0, C_1, \ell, \tau, q, t]$ with parameters

$$\begin{bmatrix} \textit{alphabet} & \mathbb{F}(n) \\ \textit{proof length} & l(n) = \widehat{l}(\ell(n)) + \ell(n) \\ \textit{query complexity} & q(n) = \widehat{q}(\ell(n)) + q(n) + 1 \\ \textit{soundness error} & \varepsilon(n) = \max\Big\{\widehat{\varepsilon}(n), 1 - \tau(n) + 2\widehat{\delta}(n)\big(q(n)+1\big)\Big\} \\ \textit{query bound} & b(n) = \min(\dfrac{t(n)}{\kappa(\ell(n)) \cdot q(n)}, \hat{b}(\ell(n))) \end{bmatrix}.$$

Proof sketch. The proof follows [BCFGRS17, Theorem 8.9] very closely; here we outline the differences only, and refer the reader to the prior work for the remainder of the proof. The only modification is in the proof of perfect zero knowledge: for each query the verifier makes to the PCPP (up to $\hat{b}$, after which the zero knowledge guarantee may fail), its simulator will make at most κ queries to w_0, w_1, resulting in at most $\kappa \cdot q$ queries to w_0 (since w_1 is q-locally computable from w_0). □

Proof of Theorem 9.1. Let $L \in \mathsf{NTIME}(T)$. We construct a ZKPCP for L by first reducing to RLACSP via Theorem 9.7, for sufficiently small constant τ and $t = \max(b(n)^2, T(n))q(n)^2$. A ZKPCPP for $C_0 \times C_1$ can be obtained by combining our Theorem 8.18 with standard techniques for testing vanishing subcodes of Reed–Solomon (as in prior work, this crucially uses that $H \cap L = \emptyset$ so that the quotient can be simulated by pointwise division). That this achieves the specified parameters is easily verified. □

References

[ALMSS98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. "Proof verification and the hardness of approximation problems". In: *Journal of the ACM* 45.3 (1998). Preliminary version in FOCS '92., pp. 501–555.

[AS98] S. Arora and S. Safra. "Probabilistic checking of proofs: a new characterization of NP". In: *Journal of the ACM* 45.1 (1998). Preliminary version in FOCS '92., pp. 70–122.

[BCFGRS17] E. Ben-Sasson, A. Chiesa, M. A. Forbes, A. Gabizon, M. Riabzev, and N. Spooner. "Zero Knowledge Protocols from Succinct Constraint Detection". In: *Proceedings of the 15th Theory of Cryptography Conference*. TCC '17. 2017, pp. 172–206.

[BCGRS17] E. Ben-Sasson, A. Chiesa, A. Gabizon, M. Riabzev, and N. Spooner. "Interactive Oracle Proofs with Constant Rate and Query Complexity". In: *Proceedings of the 44th International Colloquium on Automata, Languages and Programming*. ICALP '17. 2017, 40:1–40:15.

[BCGV16] E. Ben-Sasson, A. Chiesa, A. Gabizon, and M. Virza. “Quasilinear-Size Zero Knowledge from Linear-Algebraic PCPs”. In: *Proceedings of the 13th Theory of Cryptography Conference*. TCC ’16-A. 2016, pp. 33–64.

[BS08] E. Ben-Sasson and M. Sudan. “Short PCPs with Polylog Query Complexity”. In: *SIAM Journal on Computing* 38.2 (2008). Preliminary version appeared in STOC ’05., pp. 551–607.

[BV26] M. Bafna and N. Vyas. “Private PCPs from Product Expansion”. In: *ECCC TR26-150* (2026).

[Din07] I. Dinur. “The PCP theorem by gap amplification”. In: *Journal of the ACM* 54.3 (2007), p. 12.

[GOS24] T. Gur, J. O’Connor, and N. Spooner. “Perfect Zero-Knowledge PCPs for #P”. In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. 2024, pp. 1724–1730.

[GOS25] T. Gur, J. O’Connor, and N. Spooner. “A zero-knowledge PCP theorem”. In: *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*. 2025, pp. 986–994.

[KPT97] J. Kilian, E. Petrank, and G. Tardos. “Probabilistically checkable proofs with zero knowledge”. In: *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*. STOC ’97. 1997, pp. 496–505.

[PK24] P. Panteleev and G. Kalachev. *Maximally Extendable Sheaf Codes*. 2024. arXiv: 2403.03651 [cs.IT]. URL: https://arxiv.org/abs/2403.03651.

[PS94] A. Polishchuk and D. A. Spielman. “Nearly-linear size holographic proofs”. In: *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*. STOC ’94. 1994, pp. 194–203.

A Zero-Knowledge Gap Amplification

In this appendix, we observe that Dinur's query reduction and gap amplification preserve perfect zero-knowledge, at the cost of reducing the query bound by a polynomial factor of the query complexity of the original PCP.

Dinur [Din07] showed that a PCP of length ℓ with query complexity q can be transformed to a PZK-PCP of length $\ell \cdot \mathrm{poly}(q)$ with query complexity $O(1)$. We note that if the original PCP was zero-knowledge with respect to query bound b, then the resulting $O(1)$-query PCP will have query bound $b/\mathrm{poly}(q)$.

By the local simulatability lemma [GOS25, Lemma 3.2], it suffices to show that the gap amplification procedure is $\mathrm{poly}(q(n))$-locally computable from the original PCP. We verify this below, assuming familiarity with Dinur [Din07].

Lemma A.1. *Let (P, V) be a non-adaptive PZK-PCP system with constnat-size alphabet, polynomial proof length $\ell(n)$, query complexity $q(n)$, query bound $b(n)$, polynomial decision complexity, perfect completeness, and constant soundness error.*

Then, there exist absolute constants $c_1, c_2 > 0$ and a PZK-PCP system (P', V') with binary alphabet, proof length $\ell(n) \cdot q(n)^{c_1}$, query complexity $O(1)$, query bound $b(n)/q(n)^{c_2}$, perfect completeness, and soundness error $1/2$.

Sketch of proof. The transformation and the resulting alphabet size, proof length, query complexity, completeness, and soundness are provided and analysed in [Din07, Theorem 1.5 and Section 8]. It remains to show the resulting PCP is perfect zero-knowledge with query bound $b(n)/q(n)^{c_2}$, for some absolute constant $c_2 > 0$.

Recall the that first step of the transformation are standard arity-reduction and alphabet-reduction steps by appending a large alphabet symbol to each $q(n)$-tuple of coordinates that may be queried by the non-adaptive verifier and providing a PCPP, with an efficient deterministic prover, for the decision predicates. Clearly, this step is $q(n)$-local, as each large-alphabet symbol can be obtained by querying $q(n)$ symbols of the original proof, and each PCPP oracle is a function of a single large-alphabet symbol.

Next, write down the PCP system as a constraint graph G of size $\ell(n) \cdot q(n)^{O(1)}$ over an constant-size alphabet Σ, with unsatisfiability gap $\gamma_0 = \Omega(1/q(n))$.

Following the notation in [Din07], recall that each iteration in Dinur's gap amplification transformation has the form

$$G \to \mathrm{prep}(\mathrm{G})^t \circ P_{\mathrm{in}} \tag{7}$$

where, $\mathrm{prep}(\mathrm{G})$ is the preparation transformation (degree reduction and expanderization), the powering degree t is a fixed constant, and P_{in} is the inner assignment tester (PCPP) for alphabet reduction. By [Din07, Theorem 1.5], each iteration has unsatisfiability gap

$$\gamma_{i+1} \geq \min\{2\gamma_i, \alpha\} \ ,$$

where $\alpha > 0$ is an absolute constant, and increases the proof length by a constant factor.

Observe that each iteration of the transformation in Eq. 7 yields an output proof oracle that is $O(1)$-locally computable from its input proof oracle. To see this, note that $\mathrm{prep}(\mathrm{G})$ is a $O(1)$-local transformation by the description of the degree-reduction and expanderization steps, then the constant-degree graph is raised to the t'th power, for a constant t, which is also $O(1)$ local, and the alphabet reduction is $O(1)$-local, as observed above.

Finally, note that after $O(\log(1/\gamma(n)))$ iterations of the transformation in Eq. 7, we obtain a constant gap γ, and in turn, the constant soundness error; Moreover, we can further reduce the soundness error to $1/2$ by applying a constant number of repetitions and accepting if all invocations accepted. Since each iteration of the gap amplification procedure increased the proof length by at most a constant and decreased the query bound by at most a constant, we obtain the claimed parameters, where the constant alphabet size can be reduced to binary via standard $O(1)$-locally computable alphabet reduction. □

B Deferred Proofs Of Facts From Proof of Theorem 8.3.

In this section, set $\mathcal{F} := \mathcal{F}'[i]$ and $P := P'[i]$.

Fact B.1. For all $y \in q_{L_0}(L_1)$,

$$|(T \cap K_y)| = \sqrt{n}.$$

Proof. For each $y \in q_{L_0}(L_1)$, $|T \cap K_y| = \{(x, y) : q_{L_0}(x) = y\}| = \sqrt{n}$, which completes the proof. □

Fact B.2. For all $x \in L$,

$$|(T \cap R_x)| = 1.$$

Proof. An X-curve intersects with each row exactly once. □

Proof of Fact 8.8.

$$|K_y \cap P_{U_{R'}}| = |K_y \cap \bigcup_{x \in R'} R_x| = |\bigcup_{x \in R'} K_y \cap R_x| = \sum_{x \in R'} |K_y \cap R_x| = |R'| \cdot 1$$
$$= d_X + 1 = \dim(\mathcal{F}_{K_y})$$

The first equality follows by definition of U_1, the second equality follows by distributivity of set intersection, the third equality follows because for each $x_1, x_2 \in R'$, $P_{R_{x_1}} \cap P_{R_{x_2}} = \emptyset$, the fourth equality follows because for each $(x, y) \in \mathbb{F}^{P_*}$, $|K_y \cap R_x| = 1$, the fifth equality follows by definition of R', and the final equality follows by definition of $\mathcal{F}$. □

Proof of Fact 8.9.

$$|T \cap P_{U_{R'} \cup U_{K'}}| = |T \cap (\bigcup_{x \in R'} R_x \cup \bigcup_{y \in K'} K_y|)|$$
$$= |T \cap \bigcup_{x \in R'} R_x) \cup (T \cap \bigcup_{y \in K'} K_y)| = |(\bigcup_{x \in R'} T \cap R_x)| + |(T \cap (\bigcup_{y \in K'} K_y))|$$
$$= \sum_{x \in R'} |T \cap R_x| + \sum_{y \in K'} |T \cap K_y| = |R'| \cdot 1 + |K'| \cdot |L_0| = (d_X + 1) + d_Y \cdot |L_0|$$
$$= \dim(\mathcal{F}_T)$$

The first equality follows definitions of $U_{R'}, U_{K'}$, the second equality follows by distributivity of set intersection, the third equality follows because for each $x \in R', y \in K'$, $R_x \cap K_y \cap T = \emptyset$, the fourth equality follows because for every $x_1, x_2 \in R'$ (resp. $y_1, y_2 \in K'$), $R_{x_1} \cap R_{x_2} = \emptyset$ (resp. $K_{y_1} \cap K_{y_2} = \emptyset$), the fifth equality follows by Facts B.2 and B.1, the sixth equality follows by definitions of R', K', and the final two equalities follow by the definition of T. □

Proof of Fact 8.12.

$$
\begin{aligned}
&|P_{K_{y'}} \cap P_{U_S \cup P_{\geqslant \sigma}}| \\
&= |K_y \cap (P_\sigma \cup \bigcup_{x \in R} R_x \cup \bigcup_{y' \in K} P_{K_{y'}})| \\
&= |(K_y \cap P_\sigma) \cup (K_y \cap \bigcup_{x \in R} R_x) \cup (K_y \cap \bigcup_{y' \in K} P_{K_{y'}}))| \\
&= |(K_y \cap P_\sigma) \cup (\bigcup_{x \in R} K_y \cap R_x)| \\
&= |(K_y \cap P_\sigma)| + |(\bigcup_{x \in R} K_y \cap R_x)| \\
&= |K_y \cap P_\sigma| + \sum_{x \in R} |K_y \cap R_x| \\
&= |L_0| + |R| \cdot 1 \leq |L_0| + d_Y \leq d_X < \dim(\mathcal{F}_{K_y})
\end{aligned}
$$

The first equality follows by definition of U_S, the second follows by distributivity of set intersection, the third follows by Claim 3.1 because for all $y_1, y_2 \in K$, $P_{K_{y_1}} \cap P_{K_{y_2}} = \emptyset$, the fourth follows because for each $x \in R', y \in K'$, $R_x \cap K_y \cap P_\sigma = \emptyset$, the fifth follows because for all $x_1, x_2 \in R$, $P_{R_{x_1}} \cap P_{R_{x_2}} = \emptyset$, the sixth follows by Fact B.1. On the last line, the first inequality follows by definition of R, the second inequality follows by assumption on d_X, and the last inequality follows by definition of $\mathcal{F}$. □

Proof of Fact 8.13.

$$
\begin{aligned}
&|R_x \cap P_{U_S \cup P_{\geqslant T} \cup U_{K'}}| \\
&= |R_x \cap (T \cup \bigcup_{x' \in R} P_{R_{x'}} \cup \bigcup_{y \in K'} K_y)| \\
&= |(R_x \cap T) \cup (R_x \cap \bigcup_{x' \in R} P_{R_{x'}}) \cup (R_x \cap \bigcup_{y \in K'} K_y)| \\
&= |(R_x \cap T) \cup (R_x \cap \bigcup_{y \in K'} K_y)| \\
&= |(R_x \cap T)| + |(R_x \cap \bigcup_{y \in K'} K_y)| \\
&= |(R_x \cap T)| + |\bigcup_{y \in K'} R_x \cap K_y| \\
&= |(R_x \cap T)| + \sum_{y \in K'} |R_x \cap K_y| \\
&= 1 + |K'| \cdot 1 = d_Y + 1 = \dim(\mathcal{F}_{R_x})
\end{aligned}
$$

The first equality follows by definition of U_S, the second one follows by distributivity of set intersection, the third follows by Claim 3.1, because for every $x_1, x_2 \in R'$, $P_{R_{x_1}} \cap P_{R_{x_2}} = \emptyset$, the fourth follows because for each $x \in R', y \in K'$, $R_x \cap K_y \cap P_\sigma = \emptyset$, the fifth follows by distributivity of set intersection, the sixth follows because for each $y_1 \neq y_2 \in K'$, $K_{y_1} \cap K_{y_2} = \emptyset$, the seventh follows by Fact B.2, and the final equalities follow by definition of K' and $\mathcal{F}$. □

C Proof Of Lemma 8.16

Proof. We follow the proof of Lemma 6.11 in [BS08] closely. Let $\mathcal{F} := \mathcal{F}_i$, $P := P_i$, and $S := S_i$. First, we note that is suffices to prove that $\mathsf{row}(P) \cup \mathsf{col}(P)$ is a 1-testable cover for a Reed-Solomon code over

the slightly smaller domain, $L \setminus (L'_0)_X$. Because $|(L'_0)_X|/|L|$ is approximately $1/\sqrt{n}$, this will simply cause a small constant loss in distance. Let $f \in \mathbb{F}^T$ and let

$$\epsilon = \mathop{\mathbb{E}}_{\rho \leftarrow \mathcal{D}(S_i)} [\Delta(f_{P_\rho}, \mathcal{F}_\rho)].$$

We prove that there exists constant $c^* \in \mathbb{N}$ such that $\Delta(f, \mathcal{F}_T) \leq c^* \cdot \epsilon$. We start with the following claim.

Claim C.1. *For all $f \in \mathbb{F}^P$, there exists a constant $c \in \mathbb{N}$ such that $\Delta(f, \mathcal{F}(P)) \leq c \cdot \epsilon$.*

Proof. We use the following constants, where $\hat{c}$ is a constant to be minimized and c_0 is the universal constant from Lemma 6.13 in [BS08].

$$c_1 := \frac{\hat{c}}{2} \qquad c_2 := \frac{c_1}{3c_0} \qquad c_3 := \frac{3c_1}{16(3c_0+2)}$$

We fix c to be the minimal $\hat{c}$ such that $c_3 \geq 2$ and $\frac{1}{c_3} + \frac{2}{c_1} < 1$. Let $\epsilon_{row} := \mathbb{E}_{\rho \leftarrow \mathsf{row}(P)}[\Delta(f_{P_\rho}, \mathcal{F}_\rho)]$ and let $\epsilon_{col} := \mathbb{E}_{\rho \leftarrow \mathsf{col}(P)}[\Delta(f_{P_\rho}, \mathcal{F}_\rho)]$. Then by definition of $\mathcal{D}(S)$, $(\frac{1}{2}) \cdot (\epsilon_{col} + \epsilon_{row}) = \epsilon$ and so $\epsilon_{col}, \epsilon_{row} \leq 2\epsilon$.

Next, we consider the restriction of f to a product set. Let $G = L'_0 \times q_{L_0}(L_1)$. Let $\mathcal{E}, Q$ be the bivariate Reed-Muller sheaf code for $D_{\hat{k}}(L)$. Let $S_G := \mathcal{R} \cup \mathcal{K}$. Let $\mathcal{D}(S_G) := \mathcal{D}(S_i)$. Let $\epsilon'_{col} := \mathbb{E}_{\rho \leftarrow \mathsf{col}(Q)}[\Delta(f|_{P_\rho}, \mathcal{E}_\rho]$ and let $\epsilon'_{row} := \mathbb{E}_{\rho \leftarrow \mathsf{row}(Q)}[\Delta(f|_{P_\rho}, \mathcal{E}_\rho]$. Since for each $\rho \in \mathsf{col}(P)$, $|Q_\rho|/|P_\rho| = 2$, we have the following.

$$\epsilon'_{col} \leq 2\epsilon_{col} \leq 4\epsilon = 2c\epsilon \cdot \frac{1}{c_1}.$$

Furthermore, since for each $\rho \in \mathsf{row}(P)$, $Q_\rho = P_\rho$, it follows that

$$\epsilon'_{row} = \epsilon_{row} \leq 2\epsilon = c\epsilon \frac{1}{c_1}.$$

By Lemma 6.13 in [BS08], we have the following. For all $f \in \mathbb{F}^Q$, there exists a constant c_0 such that

$$\Delta(f, \mathcal{E}(Q)) \leq c_0(\epsilon'_{row} + \epsilon'_{col}) = c\epsilon \frac{1}{c_2}.$$

Let $U_{col} := \bigcup_{y \in \mathsf{col}(P)} P_{\geqslant y}$. Let d be the closest codeword in $\mathcal{E}(Q)$ to $f|_G$ and let e be the closest codeword in $\mathcal{E}(U_{col})$ to $f|_G$. We say β is bad if $e|_{Q_\beta} \neq d|_{Q_\beta}$. Otherwise β is good.

$$\Pr_{(x,y) \in P_*}[f(x,y) \neq d(x,y)] = \Pr_{\beta \in \mathsf{col}(P)}[\beta \text{ is bad }] + \Pr_{(\alpha,\beta) \in P_*}[f(\alpha,\beta) \neq d(\alpha,\beta) \mid \beta \text{ is good }]. \quad (8)$$

First we analyze the left hand side of the sum. If β is bad, then $e|_{Q_\beta} \neq d|_{Q_\beta}$ and so $f|_{Q_\beta}$ is either $3/8$-far from $e|_{Q_\beta}$ or $(3/8)$-far from $d|_{Q_\beta}$, because otherwise, it follows from the triangle inequality that $\Delta(e|_{Q_\beta}, d|_{Q_\beta}) \leq 6/8 = 3/4$ which is a contradiction since the minimum distance of the code is $3/4$. $\mathsf{col}(Q)$ is a cover for both $\mathcal{E}(Q)$ and $\mathcal{E}(\mathsf{col}(Q))$. Each code in the cover is equal to $RS[\mathbb{F}, Q_*, 3/4]$ and every two codes in the cover have disjoint domains. Therefore, we apply Claim 5.10 and have the following.

$$\begin{aligned}
&\Pr_{\beta \leftarrow \mathsf{col}(P)}[\beta \text{ is bad }] \\
&\leq \Pr_{\beta \in \mathsf{col}(Q)}[\Delta(f|_{Q_\beta}, d|_{Q_\tau}) > 3/8] + \Pr_{\beta \in \mathsf{col}(Q)}[\Delta(f|_{Q_\beta}, e|_{Q_\beta}) > 3/8] \quad \text{(By the triangle inequality)} \\
&\leq \frac{8}{3} \cdot \Delta(f|_Q, d) + \frac{8}{3} \cdot \Delta(f|_Q, e) \quad \text{(By Claim 5.10)} \\
&= 2 \cdot \frac{8}{3} \cdot \max(\frac{1}{c_2}, \frac{2}{c_1}) \cdot \epsilon c \quad (9) \\
&\leq \frac{\epsilon c}{c_3} \quad (10)
\end{aligned}$$

Since we minimized $\hat{c}$ so that $c_3 \geq 2$ and since we can assume that $c\epsilon \leq 1$, we analyze the right hand side of the sum in Equation 8 as follows.

$$\Pr_{(\alpha,\beta)\in P_*}[f(\alpha,\beta) \neq d(\alpha,\beta) \mid \beta \text{ is good }] = \Pr_{(\alpha,\beta)\in P_*}[f(\alpha,\beta) \neq e(\alpha,\beta) \mid \beta \text{ is good }]$$
$$\leq \frac{\Pr[f(\alpha,\beta) \neq e(\alpha,\beta)]}{\Pr[\beta \text{ is good }]} = \frac{\Delta(f,e)}{1-\Pr[\beta \text{ is bad }]} \leq 2\Delta(f,e) \leq 2 \cdot \frac{c\epsilon}{c_1} = \frac{2c\epsilon}{c_1}$$

Summing up the above two equations we have that

$$\Pr_{(x,y)\in P_*}[f(x,y) \neq d(x,y)] = (\frac{1}{c_3} + \frac{2}{c_1})c\epsilon$$

Since we assume $\frac{1}{c_3} + \frac{2}{c_1} \leq 1$, this completes the proof. □

Finally we are ready to complete the proof of the lemma. By Definition 8.1, $|T \cap K_\beta| = |L_0|$ for all $\beta \in \mathsf{col}(P)$. The portion of the input that was tested is therefore equal to $|L_0| \cdot |L_1|$. Next,

$$\frac{|P_*|}{|L_0| \cdot |L_1|} = \frac{\sum_{\beta\in\mathsf{col}(P)} |L_\beta|}{|L_0| \cdot |L_1|} = \frac{|L_\beta| \cdot |L_1|}{|L_0| \cdot |L_1|} = \frac{|L_\beta|}{|L_0|}, \tag{11}$$

which is a constant. Therefore, since by Claim C.1, $\mathcal{R} \cup \mathcal{K}$ is a 1-testable cover for $\mathcal{F}(P)$, it follows by Corollary 5.9 and Equation 11 that $\mathcal{R} \cup \mathcal{K}$ is a 1-testable cover for

$$\mathcal{F}_T = \mathrm{RS}[\mathbb{F}, L \setminus (L'_0)_X, \deg^*(T, d_X, d_Y)] = C_i,$$

and therefore (with an additive $1/\sqrt{n}$ loss) for $\mathrm{RS}[\mathbb{F}, L, \deg^*(T, d_X, d_Y)]$. □