3-Query RLDCs are Strictly Stronger than 3-Query LDCs

Tom Gur[*]   Dor Minzer [†]   Guy Weissenberg [‡]   Kai Zhe Zheng[§]

Abstract

We construct 3-query relaxed locally decodable codes (RLDCs) with constant alphabet size and length $\tilde{O}(k^2)$ for k-bit messages. Combined with the lower bound of $\tilde{\Omega}(k^3)$ of [Alrabiah, Guruswami, Kothari, Manohar, STOC 2023] on the length of locally decodable codes (LDCs) with the same parameters, we obtain a separation between RLDCs and LDCs, resolving an open problem of [Ben-Sasson, Goldreich, Harsha, Sudan and Vadhan, SICOMP 2006].

Our RLDC construction relies on two components. First, we give a new construction of probabilistically checkable proofs of proximity (PCPPs) with 3 queries, quasi-linear size, constant alphabet size, perfect completeness, and small soundness error. This improves upon all previous PCPP constructions, which either had a much higher query complexity or soundness close to 1. Second, we give a query-preserving transformation from PCPPs to RLDCs.

At the heart of our PCPP construction is a 2-query decodable PCP (dPCP) with matching parameters, and our construction builds on the HDX-based PCP of [Bafna, Minzer, Vyas, Yun, STOC 2025] and on the efficient composition framework of [Moshkovitz, Raz, JACM 2010] and [Dinur, Harsha, SICOMP 2013]. More specifically, we first show how to use the HDX-based construction to get a dPCP with matching parameters but a large alphabet size, and then prove an appropriate composition theorem (and related transformations) to reduce the alphabet size in dPCPs.

1   Introduction

Probabilistically checkable proofs (PCPs, for short) are a central object of study in theoretical computer science, with connections to hardness of approximation, cryptography, error-correcting codes, expansion in graphs and hypergraphs, property testing, analysis of Boolean functions and more. The connection to error-correcting codes, which is the focus of this paper, runs in both directions. In one direction, arguably all PCP constructions rely on error-correcting codes (such as the Reed-Solomon, Reed-Muller and Hadamard codes) as building blocks. In the other direction, as we shall see later on, strong PCPs are often building blocks in error-correcting codes with additional desirable properties. To discuss PCPs and their parameters, we shift to the perspective of constraint satisfaction problems, defined as follows.

Definition 1.1. *An instance Ψ of a constraint satisfaction problem (CSP) consists of a weighted uniform hypergraph $G = (V, E, \mathcal{P})$, where $\mathcal{P} \colon E \to (0, \infty)$ is a distribution over the edges, as well as an alphabet Σ and a collection of constraints $\{\Phi_e \colon \Sigma^q \to \{0, 1\}\}_{e \in E}$, one for each edge.*

An assignment $T \colon V \to \Sigma$ is said to satisfy the constraint on $e = (v_1, \ldots, v_q) \in E$ if $\Phi_e(T(v_1), \ldots, T(v_q)) = 1$, and the value of T is defined as the total weight of edges $e \in E$ that are satisfied. The value of the instance Ψ is the maximum value obtained by any assignment T.

The PCP theorem [FGL+96, AS98, ALM+98] asserts that there exists $\varepsilon > 0$ and a polynomial time reduction f from 3-SAT to CSP instances on $O(1)$-uniform hypergraphs with constant alphabet size, such that:

1. **Completeness:** if φ is satisfiable, then $f(\varphi)$ has value 1.

2. **Soundness:** if φ is unsatisfiable, then $f(\varphi)$ has value at most $1 - \varepsilon$.

Since the parameters of PCPs turn out to be of great importance in applications, subsequent research has put significant effort into optimizing each one of them. These parameters include the blow-up of the reduction f (sometimes called the length/size of the PCP), the number of queries q (the uniformity of the hypergraph), the structure of the constraints Φ_e, the soundness error, and the alphabet size. Indeed, in applications, one ideally wants a nearly-linear size PCP (as opposed to a polynomial blow-up as above) with as few queries as possible (optimally $q = 2$), and with soundness close to 0 (as opposed to just bounded away from 1 as above).

1.1 PCPs of Proximity, Decodable PCPs and Their Combinatorial Analogs

It turns out that to optimize parameters of PCPs, one needs, as building blocks, PCPs that have more specialized properties. In this section, we discuss two key such variants.

1.1.1 PCPs of Proximity

PCPs of proximity (PCPPs), also known as assignment testers, were initially introduced by the works [BGH+06, DR06] to get PCPs of small size (say, nearly linear). They are variants of PCPs with a more subtle notion of soundness. Whereas a PCP encoding of an assignment to 3-SAT can be used to check whether it is satisfying or not, a PCPP enables checking if the encoded assignment is *close* to a satisfying assignment. Here and throughout, the distance between a string and a language is the normalized Hamming distance, measuring the fraction of coordinates that need to be changed to get an input in the language. Below is the formal definition of PCPPs.

Definition 1.2. *Let $q \in \mathbb{N}$ and $\varepsilon, \eta > 0$. A q-query probabilistically checkable proof of proximity (PCPP) with proximity parameter η and soundness $1 - \varepsilon$ for a language $\mathcal{L} \subseteq \Sigma_0^n$ consists of a CSP instance $\Psi = (X \cup V, E, \mathcal{P}, \Sigma, \{\Phi_e\}_{e \in E})$ over a q-uniform hypergraph, where X has size n and is identified with the coordinates of Σ_0^n and $\Sigma_0 \subseteq \Sigma$. On an input $w : X \to \Sigma_0$ it has the following properties:*

- ***Completeness:*** *if $w \in \mathcal{L}$, then there exists an assignment $A : X \cup V \to \Sigma$ extending w such that*

$$\Pr_{e \sim \mathcal{P}}[\Phi_e(A(e)) = 1] = 1.$$

- *(η, ε)-**Soundness:** if w is η-far from all members of $\mathcal{L}$, then for any $A : X \cup V \to \Sigma$ extending w we have*

$$\Pr_{e \sim \mathcal{P}}[\Phi_e(A(e)) = 1] \leqslant 1 - \varepsilon.$$

We note that in contrast to a standard PCP, where one has full access to the input w, in a PCPP one only has *oracle access* to w. In particular, the number of locations read in w is also counted towards the query complexity. Thus, if the input string is very close to the language, one cannot expect constant soundness to hold in a PCPP with constantly many queries, and hence PCPPs must give soundness guarantees which are parameterized by the distance of the input from the language.

The definition of PCPPs was originally motivated by considerations arising from composition of PCPs. In a sense, a PCPP can be used to ensure that if an assignment A passes almost all of the constraints of a CSP, then it is associated with some (unique) actual word from $\mathcal{L}$. This assertion is helpful when composing PCPs with soundness close to 1. In fact, using PCPPs, one can state generic composition theorems in the close-to-1 soundness regime; see Section 1.3 for a more thorough discussion. This feature plays a crucial role in Dinur's combinatorial proof of the PCP theorem [BS08, DR06, Din07], which, among other things, is the first PCP construction of quasi-linear length and soundness bounded away from 1.

Constructing PCPPs. The works [BGH+06, DR06] show composition theorems for PCPPs, which, alongside Dinur's gap amplification technique [Din07], yield a 2-query PCPP with small (but constant) $\eta, \varepsilon > 0$, see [Mie09]. As we will see, however, some applications require PCPPs with small soundness. That is, the $1 - \varepsilon$ above is replaced by a small positive ε, and intuitively this may be achievable if η is close to 1. The composition techniques of [BGH+06, DR06] cannot achieve soundness better than $1/2$, and we are not aware of any other previous attempts

at establishing a small soundness PCPP with concretely small values of q, such as $q = 2, 3$ (one can always improve the soundness by increasing the query complexity, of course). While PCPPs seem only slightly stronger than standard PCPs, we are not aware of any black-box transformation from PCPs to PCPPs. For example, we are not aware of a PCPP analogous to the construction of [MR10].

1.1.2 Decodable PCPs

The parallel repetition theorem of Raz [Raz98] is a generic method for improving the soundness of a 2-query PCP. It is, however, incapable of giving soundness which is vanishing with the instance size. Additionally, it cannot give small (but constant) soundness PCPs with nearly-linear length. Thus, to achieve such a size-efficient PCPs, one must resort to other techniques, which all involve the composition of PCPs in one way or another. Unfortunately, composition is considerably more difficult in the low-soundness regime. The works [RS97, AS03, DFK+11, DHK15] show how to get low-soundness PCPs from a suitable composition theorem, but the number of queries their PCPs make grows as the soundness decreases. In particular, their approach falls short of giving 2-query PCPs with small soundness.

The work of [MR10], later abstracted in [DH13], gives a novel composition technique in the low-soundness regime that maintains the number of queries being 2. A key concept of that approach is decodable PCPs (dPCPs, for short).[1] Below is a somewhat simplified definition sufficient for this introductory section, and we refer the reader to Definition 3.1 for the more precise definition that we actually use.

Definition 1.3. *Let $q, L \in \mathbb{N}$ and $\varepsilon > 0$. A q-query decodable PCP (dPCP) for a language $\mathcal{L} \subseteq \Sigma_0^n$ consists of a CSP instance $\Psi = (V, E, \mathcal{P}, \Sigma, \{\Phi_e\}_{e \in E})$, a collection of decoding maps $\{D_t \colon \Sigma^q \to \Sigma_0 \cup \{\bot\}\}_{t \in [n]}$, and a collection of decoding distributions $\{\mathcal{P}_t\}_{t \in [n]}$ over E, one for each coordinate of $\mathcal{L}$. It has the following properties:*

- *__Completeness:__ for any $w \in \mathcal{L}$ there exists an assignment $A \colon V \to \Sigma$ satisfying that*

$$\Pr_{e \sim \mathcal{P}} [\Phi_e(A(e)) = 1] = 1, \qquad \Pr_{t \in [n], e \sim \mathcal{P}_t} [D_t(A(e)) = w_t] = 1.$$

  In words, there is an assignment to Ψ with value 1 that decodes to w.

- *(L, ε)-__List-decoding soundness:__ for any $A : V \to \Sigma$ we may find a list of words of size at most L, say $\{y_1, \ldots, y_L\} \subseteq \mathcal{L}$, such that*

$$\Pr_{t \in [n], e \sim \mathcal{P}_t} [\Phi_e(A(e)) = 1 \land D_t(A(e)) \notin \{(y_1)_t, \ldots, (y_L)_t\}] \leqslant \varepsilon.$$

  In words, the probability that a randomly chosen edge e is satisfied by A but decodes to something inconsistent with the list, is at most ε.

The size of Ψ is defined to be $|V| + |E|$.

Intuitively, in the low-soundness regime, there is no hope of associating a single word from $\mathcal{L}$ with an assignment A that satisfies a non-negligible fraction of the constraints, so one must resort to a list-decoding notion of soundness. It turns out that requiring the local decoding feature is the crucial additional feature that enables generic composition theorems, and we elaborate on this in Section 1.3.

Constructing dPCPs. Most known dPCP constructions are based on some variant of the manifold-versus-point test (arising from Reed-Muller codes and low-degree testing). The only other known dPCP construction is by Dinur and Meir [DM11, Theorem 1.6], which gives a 2-query dPCP construction with small soundness. Their construction, though, only achieves polynomial size (as opposed to nearly-linear size), and has super-constant alphabet size. While the fact that their dPCP is not of nearly-linear size seems inherent to their approach, their alphabet size being super-constant is due to the lack of composition results for dPCPs.

---

[1] The work [MR10] uses concrete instantiations of this object and refers to them as locally decode/reject codes. The terminology, which by now is more standard in the area, is from [DH13].

1.1.3 Relaxed Locally Decodable Codes

The above variants of PCPs have close relations to locally testable codes, locally decodable codes, and relaxed locally decodable codes, which can all be thought of as combinatorial variants of PCPs. Indeed, composition techniques in the high-soundness regime often require constructions coming from locally testable codes, and in the low-soundness regime, they often require constructions coming from locally decodable codes.

In the other direction, Dinur [Din07] shows how to obtain locally testable codes matching the parameters of her PCP. The work [BGH+06] shows (and actually, defines) a relaxed variant of locally decodable codes, called *relaxed locally decodable codes* (or RLDC in short), which is built upon their PCPP construction. RLDCs will be a central object in this paper, and roughly speaking, they are codes in which individual message bits can be locally recovered from a corrupted codeword, with the relaxation that the decoder may output a rejection symbol $\perp$ upon detecting corruption. Below is the formal definition.

Definition 1.4. *A code $C : \{0,1\}^k \to \Sigma_0^n$ is called q-query RLDC with decoding radius δ if there is a randomized decoding algorithm* Dec*, which satisfies the following:*

- **Completeness:** *for every $m \in \{0,1\}^k$ and $i \in [k]$, $\Pr[\mathsf{Dec}(\mathcal{C}(m), i) = m_i] = 1$.*

- **Soundness:** *if $w \in \Sigma_0^n$ is δ-close to $\mathcal{C}(m)$ for some $m \in \{0,1\}^k$, then for every $i \in [k]$ we have that $\Pr[\mathsf{Dec}(w, i) \in \{m_i, \perp\}] \geqslant \frac{2}{3}$.*

- **Success-rate:** *if $w \in \Sigma_0^n$ is δ-close to $\mathcal{C}(m)$ for some $m \in \{0,1\}^k$, then for at least $(1 - 16\delta - o(1))$-fraction of $i \in [k]$ we have that $\Pr[\mathsf{Dec}(w, i) = m_i] \geqslant \frac{2}{3}$.*

The above relaxation of LDCs allows for constructions with dramatically better parameters compared to their non-relaxed counterparts, where the decoder is not allowed to output $\perp$. In particular, the construction in [BGH+06] is an $O(1)$-query RLDC with nearly-linear length, while the best known construction of $O(1)$-query (non-relaxed) LDCs has superpolynomial length [Yek08, Efr12]. These improved parameters, in turn, have led to many applications, including in property testing lifting lemmas [GR18, CG18], data structures for code membership and polynomial evaluation [CGdW09], distributions that defy the law of large numbers [BDG16], constructions of universal locally testable codes [GG18], interactive proofs of proximity [GG21], quantum complexity separations [DGMT22], and beyond.

However, despite the great attention that RLDCs have received (cf. [GGK19, GRR20, GL21, AS21, BBC+22, CGS22, CY22, Gol24, KM24, CY24]), prior to this work, no separation result was known between RLDCs and LDCs. This is since, other than the recent lower bound of $\tilde{\Omega}(k^3)$ of [AGKM23] for 3-query LDCs, the known lower bounds for RLDCs [GL21, DGL23, GGS25] nearly match the state-of-the-art lower bounds for non-relaxed LDCs [KT00, WDW05].

Constructing RLDCs. There are two main paradigms for constructing RLDCs. In the low-query regime, constructions start with a base code and augment each bit of the message with a PCPP that certifies consistency of the (relevant local view of the) codeword with that bit. In the high-rate regime, constructions are closer to those for high-rate locally testable codes. Namely, they are obtained by iteratively alternating between transformations that reduce query complexity at the expense of distance, and transformations that increase distance while only mildly worsening locality (in the spirit of Dinur's gap-amplification proof of the PCP theorem).

1.2 Main Results

In this section, we state our main results, which will be convenient to express in terms of the Circuit-SAT problem.

Definition 1.5. *For a circuit $\varphi : \Sigma_0^n \to \{0,1\}$ we define the language $\mathsf{SAT}(\varphi) := x \in \Sigma_0^n \mid \varphi(x) = 1$.*

We will also discuss the size of a circuit $\varphi : \Sigma_0^n \to \{0,1\}$, which is defined in the standard way. A circuit is a directed acyclic graph with a single vertex of out-degree 0 called the output. Each node has fan-in 0 or 2, and is labeled either by an input variable if its fan-in is 0, or else by an arbitrary function from Σ_0^2 to Σ_0. The size of a circuit is the sum of the number of nodes and the number of edges in the graph, and we say it computes φ if for any input $x \in \Sigma_0^n$, the value of the output node is $\varphi(x)$.

1.2.1 Quasi-linear, Low-soundness 2-Query dPCPs and 3-Query PCPPs

Our first result is a new dPCP construction for $\mathsf{SAT}(\varphi)$ for any circuit φ, improving upon all previous constructions.

Theorem 1.6. *For all $\varepsilon > 0$ there exists $L \in \mathbb{N}$, such that for all Σ_0 there is $C \in \mathbb{N}$ such that the following holds. If $\varphi \colon \Sigma_0^n \to \{0, 1\}$ is a circuit of size N, then the language $\mathsf{SAT}(\varphi)$ has a 2-query dPCP with size $N \log^C N$ that has (L, ε)-list-decoding soundness and $O_{\varepsilon, |\Sigma_0|}(1)$ alphabet size.*

It is clear that any dPCP must make at least 2 queries, so Theorem 1.6 achieves the best parameters possible, up to the logarithmic factor in the size and the implicit dependency between L and ε (which for us is $L = (1/\varepsilon)^{O(1)}$).

Using a straightforward transformation from dPCPs to PCPPs, we use Theorem 1.6 to conclude a 3-query PCPP construction with small soundness.

Theorem 1.7. *For all $\varepsilon > 0$ there is $\eta > 0$, such that for all Σ_0 there is $C > 0$, such that the following holds. If $\varphi \colon \Sigma_0^n \to \{0, 1\}$ is a circuit of size N, then the language $\mathsf{SAT}(\varphi)$ has a 3-query PCPP with size $N \log^C N$ that has $(1 - \eta, 1 - \varepsilon)$-soundness and $O_{\varepsilon, |\Sigma_0|}(1)$ alphabet size.*

Intuitively, the construction in Theorem 1.7 is the same as the construction in Theorem 1.6, except that we query a random index i of the supposed word w from $\mathcal{L}$, decode it using the dPCP, and compare to w_i. In other words, we "spend" one query to check if the assignment to the CSP decodes to w. As long as one sticks to such a recipe, it seems impossible to achieve query complexity smaller than 3. As far as we know, though, it may be possible to achieve a small soundness, 2-query PCPP for certain languages of interest.

At the cost of paying one additional query, we can get a tight relationship between ε and η, which is important in some applications. The statement below is meaningful when $\varepsilon > 0$ is thought of as small, and η can be taken to be any value between 0 and 1.

Theorem 1.8. *For all $\varepsilon > 0$, $0 < \eta < 1$ and Σ_0, there is $C > 0$ such that the following holds. If $\varphi \colon \Sigma_0^n \to \{0, 1\}$ is a circuit of size N, then the language $\mathsf{SAT}(\varphi)$ has a 4-query PCPP with size $N \log^C N$ that has $(\eta, \eta + \varepsilon)$-soundness and $O_{\varepsilon, \eta, |\Sigma_0|}(1)$ alphabet size.*

1.2.2 3-Query RLDCs are Strictly Stronger than 3-Query LDCs

We provide a new black-box transformation of PCPPs into RLDCs, which preserves query complexity. All prior similar transformations led to an increase in the query complexity, and in particular, they cannot be used to get a 3-query RLDC. Combining our transformation with Theorem 1.7, we construct RLDCs with 3 queries. More precisely, we show the following.

Theorem 1.9. *There are $\varepsilon, \delta > 0$, constant C, and alphabet Σ_0 such that for sufficiently large k, there is a 3-query relaxed locally decodable code $\mathcal{C} \colon \{0, 1\}^k \to \Sigma_0^n$ with decoding radius δ, distance ε and $n \leqslant k^2 \log^C k$.*

In words, Theorem 1.9 establishes the existence of a 3-query RLDC with constant decoding radius and distance, and roughly quadratic blow-up in the message length. All previous $O(1)$-query RLDC constructions (such as in [BGH+06, GGK19, GRR20, AS21, CGS22]) require more than 3 queries. The significance of Theorem 1.9 achieving 3 queries is more apparent once it is compared to the lower bounds for LDCs of [AGKM23]:

Theorem 1.10 (Theorem A.2 in [AGKM23])**.** *Suppose $C \colon \{0, 1\}^k \to \Sigma_0^n$ is a 3-query locally decodable code with decoding radius $\Omega(1)$, distance $\Omega(1)$ and alphabet $|\Sigma_0| = O(1)$. Then $n \geqslant \Omega(k^3 / \log^6 k)$.*

In particular, combining Theorems 1.9 and 1.10 gives the following corollary, which asserts that 3-query RLDCs achieve strictly stronger parameters than 3-query LDCs. This provides a separation between RLDCs and LDCs, resolving an open problem due to Ben-Sasson, Goldreich, Harsha, Sudan, and Vadhan [BGH+06] (see also Goldreich's textbook [Gol10, Chapter 13.4.4] and recent surveys [Gol24, Gas24]).

Corollary 1.11. *There is a family of 3-query RLDCs with constant decoding radius, distance, and alphabet size that achieves polynomially better length than any family of 3-query LDCs with constant decoding radius, distance, and alphabet size.*

Remark 1.12. *The recent work [GKMM25] showed that linear 3-query RLDCs with a* binary *alphabet imply non-relaxed LDCs with similar parameters. While our construction is not linear, this barrier suggests that a non-binary alphabet is likely a prerequisite for achieving such polynomial separation.*

Other Implications. Our results have a few direct implications using existing results in the literature, and we mention one of them.

Using the connection between gaps of PSPACE-hardness for the q-CSP reconfiguration problem and parameters of PCPPs established by [GRW25], we get the following two corollaries:

Corollary 1.13. *For any constant $\eta > 0$ there is an alphabet Σ of size $O_\eta(1)$, such that $\mathrm{Gap}_{1,\frac{1}{2}+\eta}$ 5-CSP$_\Sigma$ Reconfiguration is* PSPACE-*hard.*

Corollary 1.14. *For any constant $\eta > 0$ there is an alphabet Σ of size $O_\eta(1)$, such that $\mathrm{Gap}_{1,0.9+\eta}$ 2-CSP$_\Sigma$ Reconfiguration is* PSPACE-*hard.*

These results improve upon [HO24, GRW25], who proved that $\mathrm{Gap}_{1,0.9942+\eta}$ 2-CSP$_\Sigma$ is PSPACE-hard for some Σ of size $O_\eta(1)$. We remark that for both of them, it is important that we use Theorem 1.8, which has a near-tight relationship between the distance and rejection parameters.

1.3   Techniques in our PCP Constructions

In this section, we discuss the techniques in the proofs of Theorems 1.6 and 1.7. We begin with a general discussion about how PCPs are constructed.

1.3.1   The General PCP Recipe

Essentially all known PCP constructions can be viewed as using the following high-level recipe. Starting with a language $\mathcal{L} \subseteq \Sigma_0^n$, they consist of:

1. **Outer PCP:** is a fairly good PCP construction for the language $\mathcal{L}$, except that its alphabet Σ is too large (being either polynomial in n or sometimes even quasi-polynomial in n, provided it has small "decision complexity", a notion we will define shortly).

2. **Inner PCP:** is a PCP construction for a language $\mathcal{L}' \subseteq (\Sigma_0')^{n'}$ that arises from the constraint structure or the alphabet structure of the outer PCP. The key additional features of the inner PCP construction are that its alphabet size is much smaller than that of the outer PCP, and that it satisfies a stronger notion of soundness as discussed above. The inner PCP could be fairly "lossy" with respect to other parameters, such as its length relative to n'. This is because the n' we take is typically much smaller than the above n, as the intention here is to encode an alphabet symbol of the outer PCP.

3. **A composition procedure:** this is a technique that combines the outer PCP and the inner PCP constructions to yield a composed PCP. Roughly speaking, the soundness and completeness of the composed PCP are governed by the soundness and completeness of the two components, the alphabet size of the composed PCP is inherited from the inner PCP construction, and the size is the product of the sizes.

To illustrate this recipe, we define a special class of CSPs, which goes by the name Label-Cover.

Definition 1.15. *An instance Ψ of Label-Cover consists of a weighted bipartite graph $G = (A \cup B, E, \mathcal{P})$, where $\mathcal{P}: E \to (0, \infty)$ is a distribution over the edges, alphabets Σ_A, Σ_B and constraints $\{\Phi_e: \Sigma_A \times \Sigma_B \to \{0,1\}\}_{e \in E}$.*

1. *The value of a pair of assignments $T_A: A \to \Sigma_A$, $T_B: B \to \Sigma_B$ is the sum of weights of constraints they satisfy, and the value of Ψ is the maximum weight of constraints satisfied by any T_A, T_B.*

2. *We say Ψ is a projection Label-Cover instance if for each edge $e \in E$, the constraint Φ_e is projection from Σ_A to Σ_B. Namely, there exists a map $\phi_e: \Sigma_A \to \Sigma_B$ such that $\Phi_e(\sigma, \tau) = 1$ if and only if $\tau = \phi_e(\sigma)$.*[2]

---

[2]The notion of projection Label-Cover is essentially equivalent to the notion of robust PCPs, which is sometimes used in the literature.

Suppose we have an outer PCP construction $\Psi = (G = (V, E, \mathcal{P}), \Sigma, \{\Phi_e\}_{e \in E})$ as in Definition 1.1 which is good (in the sense that $\varepsilon > 0$ and q are absolute constants), but its alphabet Σ is too large. To discuss composition, it is first convenient to convert Ψ into a Label-Cover instance Ψ'.

The instance Ψ' is defined over the graph $H = (A \cup B, E', \mathcal{P}')$ whose sides A and B are the edge set E and the vertex set V of Ψ respectively, its weight function $\mathcal{P}'$ is naturally derived from $\mathcal{P}$ by: sample $e \sim \mathcal{P}$, sample a vertex v from e uniformly and output (e, v). For $e = (v_1, \ldots, v_q) \in A$, we interpret the alphabet Σ_A as an assignment of Σ symbols to the variables $v_1, \ldots, v_q$ that satisfies the constraint Φ_e; for $v \in B$, we interpret the alphabet Σ_B simply as Σ. Finally, the constraint on $(e, v) \in E'$ is that the symbol assignment to e and the symbol assigned to v are consistent with respect to the value given to v. Using error-correcting codes, the instance Ψ' can be transformed into an instance Ψ'' over a similar graph $(A \cup C, E'', \mathcal{P}'')$ with similar constraints and alphabets, except that the alphabet Σ_C now has constant size, and for the sake of this introduction, we just take $\Sigma_C = \{0, 1\}$.

Suppose for the sake of simplicity that the constraint Φ_e is the same for all edges $e \in E$. In this case, to reduce the alphabet size of Ψ', we would like to be able to encode assignments to H, so that we can check: (1) the hardcoded condition in Σ_A, which is morally how the constraint Φ_e is manifested, (2) the consistency between the assignment to A and B, which ultimately ensures that the local assignments to the A side come from a legitimate assignment to V. Towards this end, we may think of symbols in Σ_A as strings of length n', say $\Sigma_A \subseteq \{0, 1\}^{n'}$, and take an inner PCP construction suitable for the language

$$\mathcal{L}' = \{x \in \{0, 1\}^{n'} \mid x \in \Sigma_A\}.$$

At this point, it becomes clear that we care about the complexity of the language $\mathcal{L}'$, otherwise there is no reason we would have a decent PCP construction for it. This is the motivation for the notion of *decision complexity*, which captures the circuit complexity of the constraints (in the context of CSP instances) or alphabet membership checks (in the context of Label-Cover), and we define it formally later. In our applications, n' and the circuit complexity of $\mathcal{L}'$ will both be poly-logarithmic in n, and in particular much smaller than n.

In the high-soundness regime, the composition can now be performed by replacing a vertex in the A-side of Ψ'' with a copy of the inner PCP of $\mathcal{L}'$. Morally, this automatically takes care of issue (1) above, namely the hardcoded condition in the alphabet Σ_A. To address issue (2), namely the consistency between A and C, we require the inner PCP to be a PCP of proximity for $\mathcal{L}'$. In that case, we could actually reliably read off coordinates of the PCPP corresponding to the coordinates in $\mathcal{L}'$ and compare them to what they are supposed to be from the assignment to the C side. This composed PCP can be either thought of as having more than 2 queries, or equivalently, as having 2 queries but soundness close to 1 (by simply running each one of the above-mentioned tests with probability $1/2$).

Composing in the low-soundness regime is more challenging, since the two tests above need to be combined into a single test in a way that preserves soundness. This is precisely where the notion of decodable PCPs comes in handy. By definition, by reading off the locations corresponding to a constraint in a dPCP, we can – at the same time – check a constraint and decode a coordinate from the supposed word of the language. With this description in mind, the main challenges in proving Theorem 1.6 are as follows.

1. **Starting point:** no prior dPCP constructions achieve the parameters of Theorem 1.6, even if we relax the requirement on the alphabet size and allow it to be somewhat large (so as to serve as the outer PCP in the above recipe). The closest result in the literature is due to Dinur and Meir [DM11], but its length is significantly bigger than quasi-linear.

2. **Composition theorem:** we are not aware of any prior composition results that yield a composed dPCP. As a matter of fact, in the low-soundness regime, we are only aware of a single query-efficient composition result for any variant of PCPs, namely the one in [MR10, DH13].

In the rest of this section we discuss the way we overcome both of these challenges.

1.3.2 Our Starting Point

Our starting point is the recent PCP construction of [BMVY25], which is based on high-dimensional expanders. Roughly speaking, the bulk of their work goes into showing the existence of a quasi-linear, low-soundness PCP for

NP, with a quasi-polynomial alphabet and poly-logarithmic decision complexity. They then reduce its alphabet using the techniques from [MR10, DH13].

We show that, after appropriate modifications, applying their construction on the PCPP of Mie [Mie09] gives a decodable 2-query PCP (as opposed to just a normal PCP) with similar parameters. Since we are not aware of any black-box transformation, our argument has to carefully go through the steps of their construction and show that they all preserve the stronger notion of soundness. These steps include:

1. **Embedding a given PCPP into an HDX**: we use the same link-to-link routing protocol to transform a given 2-query dPCP to a 2-query dPCP over the base graph of a high-dimensional expander X, provided the latter satisfies sufficiently good expansion properties.

2. **Amplification via direct product testing:** we show that the sparse direct product testing results in the low-soundness regime of [BM24, BLM24, DD24a, DD24c, DD24b] as used in [BMVY25], can also be used in the context of decodable PCPs to decrease the soundness of our HDX-based 2-query dPCPs.

Combining these two components gives a version of the 2-query dPCP of Dinur and Meir [DM11] with only quasi-linear size. The next step in our argument is to show a way of reducing the alphabet size.

1.3.3 Our Composition Theorem

To reduce the alphabet size of our HDX-based dPCP, we also use a composition theorem, except that we have to prove versions that yield dPCPs. To discuss our composition theorems we have to define a few additional parameters of interest of dPCPs, and to do so we first give the appropriate definition of label-cover dPCPs.

Definition 1.16. *Let $\mathcal{L} \subseteq \Sigma_0^n$ be a language. A label-cover, decodable PCP for $\mathcal{L}$ is a tuple*

$$\mathcal{D} = ((A \cup B, E), \{\mathcal{P}_t\}_{t \in [n]}, \Sigma_A, \Sigma_B, \{\Phi_e\}_{e \in E}, \{D_t\}_{t \in [n]}),$$

where $(A \cup B, E)$ is graph, $\{\Phi_e \colon \Sigma_A \times \Sigma_B \to \{0,1\}\}_{e \in E}$ is a collection of constraints, $\mathcal{P}_t$ is a distribution over E for each $t \in [n]$ and $D_t \colon A \times \Sigma_A \to \Sigma_0$ is a decoding map for each $t \in [n]$. We say that $\mathcal{D}$ is a projection dPCP if for each $e \in E$, Φ_e is a projection constraint.

There are several differences between Definition 1.16 and Definition 1.3. First, the former uses bipartite graphs, whereas the latter uses hypergraphs. Second, in the former, the decoding maps only read the left vertex and an assignment to it, whereas in the latter, they read an assignment to an edge. These two differences are relatively minor and are very natural in light of the transformation discussed in Section 1.3.1.

Besides the obvious PCP-related parameters (such as completeness and soundness, which are defined analogously), we will also care about graph-theoretic parameters, such as left and right degrees of vertices. A somewhat less standard parameter that will be important for us is the decoding degree:

Definition 1.17. *Let $\mathcal{D}$ be a label-cover decodable PCP as in Definition 1.16. The decoding-degree of a vertex $a \in A$, denoted by $\mathrm{ddeg}(a)$, is the number of indices it is in charge of decoding, namely $|\{t \in [n] \mid \exists b \in B, (a,b) \in \mathrm{supp}(\mathcal{P}_t)\}|$. The decoding-degree of $\mathcal{D}$ is defined as $\max_{a \in A} \mathrm{ddeg}(a)$.*

With this notion in place, we now discuss our composition theorem.

Composition when the decoding-degree is constant: our composition theorem applies in the case we have a projection label-cover dPCP $\mathcal{D}$ with constant decoding-degree. In this case, we show how to combine two copies of it, one of the scale of n as an outer PCP and one at the scale of a much smaller n' as an inner PCP, to get a projection label-cover dPCP with similar parameters, except that its alphabet size is much smaller.

This construction is a natural extension of the abstract composition theorem of Dinur and Harsha [DH13], and morally proceeds as follows. First, we use the standard right-alphabet reduction technique from [MR10, DH13] to ensure that Σ_B is constant. Next, in the context of standard PCPs, the alphabet Σ_A is replaced by $\Sigma_A' \subseteq \Sigma_B^{\mathrm{deg}(a)}$ reflecting the supposed assignments of all neighbors $b \in B$ of A; only strings that are fully consistent with some symbol in Σ_A are allowed. This change in the alphabet of A is not quite valid in our setting of dPCPs since it is not

clear how to use the decoders of $\mathcal{D}$ to construct decoders for the new construction. To remedy that, we actually replace the alphabet Σ_A with $\Sigma'_A \subseteq \Sigma_B^{\deg(a)} \times \Sigma_0^{\mathrm{ddeg}(a)}$, which is thought of as a string reflecting the assignments of all of the neighbors $b \in B$ of a, as well as of all of the $t \in [n]$ that a is in charge of. Once again, in Σ'_A we only allow strings that are fully consistent with some label σ to a from the original alphabet Σ_A, which additionally decodes correctly (under the decoding map $D_t(a, \cdot)$) on all t that a is in charge of.

In the simplified setting, all constraints Φ_e are the same, and a similar analysis as in [DH13] can be carried out. The case that Φ_e may be different introduces several technical difficulties, stemming from the following technical issue. In the composed PCP, a left vertex corresponds to a right vertex b in the outer PCP, and to the invocation of all of the inner PCPs of neighbors $\mathbf{a}$ of b with a certain seed of randomness. Thus, we need the inner PCPs neighbors $\mathbf{a}$ of b to be "aligned"; for example, it would be nice if they use the same seed of randomness. We show how to accomplish this, and this requires a few fairly technical transformations of dPCP that manipulate its size while retaining its parameters.

Unfortunately, this composition result alone is insufficient for our purposes. As is clear from the above description, the alphabet size of the composed PCP is at least exponential in the decoding degree of the outer PCP. With that in mind, our HDX-based 2-query dPCP construction has poly-logarithmic decoding degree, so if we plainly apply the aforementioned composition theorem, the alphabet size of the composed PCP will be (at the very least) $2^{\mathsf{poly}(\log n)}$. In other words, nothing was gained from the composition. Having a poly-logarithmic decoding degree seems inherent in the HDX construction (due to the use of routing protocols). In particular, it is not clear how to use the composition theorem we have just described to get a better dPCP.

Decoding-degree reduction: in light of the above, it makes sense to ask if there is a transformation that reduces the decoding degree of a given dPCP without hurting the other parameters too much. Such transformations do exist for more combinatorial notions, such as right degree and right alphabet size. While we do not have a generic result of this form, we give a transformation that works provided that the dPCP satisfies an additional property called *agnosticism*, which our HDX-based construction luckily possesses. To present it, we first introduce the complete decoding distribution.

Definition 1.18 (Complete Decoding Distribution). *Let $\mathcal{D}$ be a decodable PCP as in Definition 1.16. The complete decoding distribution of $\mathcal{D}$ is the distribution $\mathcal{Q}$ over $[n] \times A \times B$, in which a sample is drawn by taking $t \in [n]$ uniformly, $(a, b) \sim \mathcal{P}_t$, and outputting (t, a, b).*

Letting $\mathcal{Q}$ be the complete decoding distribution of a dPCP $\mathcal{D}$, we denote by $\mathcal{Q}(t, \cdot, b)$ the marginal distribution of a conditioned on t and b, and by $\mathcal{Q}(\cdot, \cdot, b)$ the marginal distribution of (t, a) conditioned on b.

Definition 1.19. *We say a dPCP $\mathcal{D}$ has agnostic decoding distributions if for all t and b, the marginal distribution of $\mathcal{Q}(\cdot, \cdot, b)$ on a is identical to $\mathcal{Q}(t, \cdot, b)$.*

With this notion in mind, we can now describe our decoding-degree reduction transformation. Given a dPCP $\mathcal{D}$ with an agnostic complete decoding distribution, we construct a new dPCP $\mathcal{D}_2$ as follows. We first replace a right-vertex $b \in B$ with copies (b, t) of it for $t \in [n]$ such that there is a neighbour a of b such that $(a, b) \in \mathrm{supp}(\mathcal{P}_t)$, and modify the alphabets and constraints appropriately. Namely, denoting the right degree of $\mathcal{D}$ by d, the alphabet of the right is now a subset of $\Sigma_A^d \times \Sigma_0$. Thinking of a right vertex $b \in B$ and its neighbors $a_1, \ldots, a_d$, a symbol $(\tau_1, \ldots, \tau_d, s)$ for (b, t) is allowed if all the projections $\phi_{(a_i, b)}(\sigma_i)$ are the same, and for any i such that t is in the decoding neighbourhood of a_i we have that $D_t(a_i, \tau_i) = s$. The constraint on the edge $(a_i, (b, t))$ is that the assignments σ to a and $(\tau_1, \ldots, \tau_d, s)$ to (b, t) satisfy that $\sigma = \tau_i$. The complete decoding distribution of $\mathcal{D}_2$ is defined by choosing $t \in [n]$ uniformly, then $(a, b) \sim \mathcal{P}_t$ and output $(t, (b, t), a)$. In particular, the side A of $\mathcal{D}$ is the right side of $\mathcal{D}_2$, and the pairs (b, r) are the left side of $\mathcal{D}_2$. As per the new decoder, when D_t is run on $((b, t), (\tau_1, \ldots, \tau_d, s))$, the output is simply s.

In a sense, in this transformation we have created copies of b according to which t we want it to be able to decode, and also we "flipped" the sides of the projections (by making the alphabet of (b, t) include an opinion for the label of each neighbour of b). The latter step is similar to the flipping-of-sides transformation in [MR10, DH13].

Most of the features of the new dPCP construction, such as decoding degree (which is 1 in $\mathcal{D}_2$), alphabet size, and so on, are rather immediate to analyze, and the key challenge is the soundness. If $\mathcal{D}$ were an arbitrary PCP, because

we have split a vertex b into several copies depending on t, the extra knowledge of both b and t immediately reveals from which decoding distribution $\mathcal{P}_t$ the constraint was sampled. When designing an assignment for (b, t), this gives extra information about which vertex a is chosen to test the constraint, which breaks the soundness guarantee of $\mathcal{D}$. This is precisely where the fact that $\mathcal{D}$ has agnostic decoding distribution comes in handy, since in that case no extra information about a is gained, and the soundness guarantee of $\mathcal{D}$ persists.

Remark 1.20. *Our actual decoding-degree procedure in Section 8 has one minor difference, which is that we may create several copies of each (b, t). This is related to the fact that we want to "align" the size of the inner PCPs as per our earlier discussion.*

1.4 Techniques in our RLDC Construction

We obtain our 3-query RLDC via a new query-preserving black box transformation from PCPPs to RLDCs.

While many RLDC constructions invoke PCPPs as a subroutine, prior approaches either employ multiple invocations of PCPP verifiers or introduce additional queries beyond those of the underlying PCPP.

Our starting point is the quadratic-length, $O(1)$-query construction in [BGH+06], which uses a PCP of proximity to construct an RLDC as follows. Let $\mathcal{C}_0 \colon \{0, 1\}^k \to \{0, 1\}^n$ be a linear-time encodable code with constant distance rate and distance, and let $\pi_i(\cdot)$ be a PCP of proximity encoding for the language $\mathcal{L}_i = \{(z, y) \mid \exists x \in \{0, 1\}^k, \ y = \mathcal{C}_0(x), z = (x_i)^n\}$. The code $\mathcal{C}$ constructed by [BGH+06] is given as:

$$\mathcal{C}(x) = (\mathcal{C}_0(x))^t \circ (x)^{t'} \circ \pi_1(x_i^n, \mathcal{C}(x)) \circ \cdots \circ \pi_k(x_i^n, \mathcal{C}(x)), \tag{1}$$

namely, the first part is t repetitions of the encoding of x under $\mathcal{C}_0$, the second part is t' repetitions of x itself, and the last part consists of a PCP of proximity for each bit in x. Here, the purpose of t, t' is to balance the length of the different parts, so that the code $\mathcal{C}$ has a constant distance. On input word w and index $i \in [k]$, the basic decoder of [BGH+06] proceeds by running the PCP of proximity on the oracle $((z, y), \pi)$, where (1) z is taken to be $z = \sigma^n$ where σ is the symbol appearing in a random place in the received word w that is supposed to be x_i, (2) y is taken to be a supposed random copy of $\mathcal{C}_0(x)$, (3) π is taken to be the supposed PCPP encoding of $\mathcal{L}_i$. The query complexity of this basic decoder is the same as that of the PCPP, and [BGH+06] show that it works provided that the PCPP for $\mathcal{L}_i$ has sufficiently small decoding radius η. This already implies that the PCPP must make $\Omega(1/\eta)$-queries, which means the RLDC construction above must make some unspecified, likely not-too-small, number of queries.

Our construction is syntactically similar, but with a few important distinctions. Since we wish to use Theorem 1.7, whenever we apply a PCPP decode, we want it to be in the case that the word is very far from the relevant language. In particular, we must take the code $\mathcal{C}_0 \colon \{0, 1\}^k \to \Sigma_0^n$ to have distance close to 1, and (more importantly) we cannot afford to have the "x-part" as in the encoding in (1). We thus use the following variant:

$$\mathcal{C}(x) = (\mathcal{C}_0(x))^t \circ \pi'_{1, x_1}(\mathcal{C}(x)) \circ \cdots \circ \pi'_{k, x_k}(\mathcal{C}(x)), \tag{2}$$

where again t is a suitable repetition parameter, and the PCPP part of the encoding is a bit different. To define $\pi'_{i, b}$ we first take $\pi_{i, b}$ to be the PCPP encoding for the language

$$\mathcal{L}_{i, b} = \{w \in \Sigma_0^n \mid \exists x \in \{0, 1\}^k, \mathcal{C}_0(x) = w, x_i = b\}.$$

Denoting the alphabet of $\pi_{i, b}$ by Σ, the alphabet of $\pi'_{i, b}$ is now $\Sigma \times \{0, 1\}$, and the jth symbol in $\pi'_{i, b}(w)$ is the jth symbol in $\pi_{i, b}(w)$ with b appended to it.

The reason we need to use $\pi'_{i, b}$ in place of $\pi_{i, b}$ is that, for a fixed i, upon reading a single symbol from $\pi_{i, b}(w)$, we may not know whether we need to run the verifier of $\mathcal{L}_{i, 0}$ or of $\mathcal{L}_{i, 1}$. Appending b to each symbol resolves this issue.

With this construction in mind, when run on a word w and an index $i \in [k]$, the RLDC decoder looks at ith part in the PCPP part of (2) and invokes it on a randomly chosen copy of $\mathcal{C}_0(x)$ from the first part of (2). Indeed, we show that this decoder works (provided that the PCPP decoder has a few additional technical features, which hold in our case).

1.5 Discussion and Open Problems

While Theorem 1.6 gives dPCPs with essentially the best possible query complexity, size, and soundness, the same cannot be said about Theorem 1.7. Existing lower bounds for PCPPs [BHLM09] do not rule out quasi-linear 2-query PCPPs with constant (but not Boolean) alphabet size that have small soundness, and we leave the problems of giving better 2-query PCPP constructions or lower bounds to future research. We remark that in some applications, there may be more context-dependent reductions that use our dPCP (instead of our PCPP) directly and thus do not suffer the loss incurred from the additional query.

Another interesting question is whether one can establish a PCPP with a tight relationship between the proximity parameter and the soundness, as in Theorem 1.8, but with a smaller number of queries. This will yield direct improvements in some applications, such as better hardness gap in Corollary 1.14. Lastly, it would be interesting to come up with a variant of our RLDC construction that is additionally locally testable (we suspect our code is locally testable, but investigating it seems to require uniqueness properties from our PCP constructions, which go beyond the scope of this work).

Lastly, our codes from Theorem 1.9 have 3 queries and nearly quadratic length, and in a companion paper we show how to generalize these codes for any odd number of queries $q \geqslant 3$. We do not know if better 3-query RLDCs exist, and it would be interesting to either prove lower bounds that match the performance of our codes, or come up with sub-quadratic 3-query RLDCs.

2 Preliminaries

In this section we introduce several basic notions and tools that will be used in our arguments.

Notations. Throughout the paper, for $n \in \mathbb{N}$ we denote $[n] = \{1, \ldots, n\}$ and use $t \in [n]$ to denote a uniformly chosen element in $[n]$. Given a distribution $\mathcal{Q}$ over a finite set S, we write $s \sim \mathcal{Q}$ to denote a random sample distributed according to $\mathcal{Q}$. For a subset $U \subseteq S$, $\mathcal{Q}(U) = \sum_{s \in U} \mathcal{Q}(s)$ is the measure of U under $\mathcal{Q}$. The total variation distance between two distributions, $\mathcal{Q}, \mathcal{Q}'$ is defined as

$$\mathsf{TV}(\mathcal{Q}, \mathcal{Q}') = \frac{1}{2} \sum_{s \in S} |\mathcal{Q}(s) - \mathcal{Q}'(s)| = \max_{U \subseteq S} |\mathcal{Q}(U) - \mathcal{Q}'(U)|.$$

We say that $\mathcal{Q}$ is M-discrete if for every $s \in S$, there is $w(s) \in \mathbb{N}$ such that $\mathcal{Q}(s) = \frac{w(s)}{M}$.

When $\mathcal{Q}$ is a multivariate distribution, say m-variate, we denote by $\mathcal{Q}(a_1, \ldots, a_i, \cdot, \ldots \cdot)$ the distribution of $a_{i+1}, \ldots, a_m$ conditioned on $a_1, \ldots, a_i$. We use a similar notation when the coordinates we condition on are not consecutive. We use the notation $\mathcal{Q}(a_1, \ldots, a_i, \circ, \ldots, \circ)$ to denote $\sum_{a_{i+1}, \ldots, a_m} \mathcal{Q}(a_1, \ldots, a_i, a_{i+1}, \ldots, a_m)$, i.e., the marginal distribution of $\mathcal{Q}$ on the first i coordinates.

Given a length n vector v, we write v_i to denote the ith coordinate of v. Given a function F over a domain X and a subset $Y \subseteq X$, we denote by $F|_Y$ the restriction of F to Y.

We use standard big-O notation: for functions $f, g \colon \mathbb{N} \to [0, \infty)$, the notation $f = O(g)$ means that there is an absolute constant C such that $f(n) \leqslant Cg(n)$ for all $n \in \mathbb{N}$, and the notation $f = \Omega(g)$ means that there is an absolute constant $C > 0$ such that $f(n) \geqslant Cg(n)$ for all $n \in \mathbb{N}$. Often we will have dependency on auxiliary parameters, and we write $f = O_\varepsilon(g)$ or $f = \Omega_\varepsilon(g)$ if the constant C depends on ε. When we write $\mathrm{poly}_\varepsilon(N)$ we mean a polynomial in N whose degree and coefficients may be increasing in $1/\varepsilon$.

2.1 Error-Correcting Codes

An error-correcting code over an alphabet Σ is a map $\mathcal{C} \colon \{0, 1\}^k \to \Sigma^n$. We refer to k as the message length and to n as the blocklength of the code. Abusing notation, we will often use $\mathcal{C}$ to refer to both the map and its image, and it will always be clear from context which we are referring to. We refer to the strings in $\mathcal{C}$ as codewords. The distance between $w, w' \in \Sigma^n$ is the relative Hamming distance between them, namely

$$\mathsf{dist}(w, w') = \Pr_{i \in [n]}[w_i \neq w'_i].$$

The relative distance of a set $\mathcal{C}$ is defined as

$$\mathsf{dist}(\mathcal{C}) = \min_{w,w' \in \mathcal{C}, w \neq w'} \mathsf{dist}(w, w').$$

Informally, when discussing error-correcting codes we will deal with sets $\mathcal{C}$ that have $\Omega(1)$ relative distance.

We will require an error-correcting code with message length k, quasi-linear time encoding, quasi-linear time decoding, constant alphabet size, relative distance arbitrarily close to 1 and blocklength $n = O(k)$. By quasi-linear time encoding, we mean that there is a circuit of size $\tilde{O}(k)$ that on input $x \in \{0,1\}^k$ computes its encoding, and similarly for decoding. There are multiple examples of such codes throughout the literature [Spi96, GI05, RS06], and we use the following formulation from [GI05, Theorem 3].

Lemma 2.1. *For every $\varepsilon > 0$, $k \in \mathbb{N}$, sufficiently large alphabet Σ relative to $1/\varepsilon$, and blocklength $n = O_\varepsilon(k)$ there exists a quasi-linear time encodable/decodable code $\mathcal{C} : \{0,1\}^k \to \Sigma^n$ with distance $1 - \varepsilon$. Furthermore, given k one can construct in time $k^{O_\varepsilon(1)}$ a circuit of size $\tilde{O}_\varepsilon(k)$ computing $\mathcal{C}$.*

We will also require the well-known Johnson Bound, which bounds the number of codewords in $\mathcal{C} \subseteq \Sigma^n$ that lie within proximity of a given string $w \in \Sigma^n$. We refer to [DH13, Fact 5.3] for a proof.

Fact 2.2. *Let $\mathcal{C} \subseteq \Sigma^n$ be a code with minimum distance $1 - \delta$. Then, for any $w \in \Sigma^n$ and $\eta > 2\sqrt{\delta}$, the number of $w' \in \mathcal{C}$ such that $\mathsf{dist}(w, w') \leqslant 1 - \eta$ is at most $\frac{2}{\eta}$.*

2.2 Expanders and Bipartite Expanders

Our arguments will use bipartite expander graphs and the bipartite expander mixing lemma, and we introduce these notions in this section. Let G be a weighted bipartite graph with sides U, V, and let $w : E \to \mathbb{R}_{\geqslant 0}$ be the weight function. The weighted adjacency matrix of G is $M \in \mathbb{R}^{(U \cup V) \times (U \cup V)}$ defined as $M = D^{-1/2} W D^{-1/2}$, where $W(u, v) = w(u, v)$ and the degree matrix D is a diagonal matrix with the entries $D(u, u) = \sum_v w(u, v)$. The transition matrix of G is defined as $P = D^{-1} M_{U,V}$, where $M_{U,V} \in \mathbb{R}^{U \times V}$ is the matrix with $M_{U,V}(u, v) = w(u, v)$. It is well known that the largest singular of P is 1, and is given by the vectors $x \in \mathbb{R}^U$ where $x(u) = D(u, u)$. Therefore, the stationary distribution of G on each side is proportional to the weighted-degrees given in D. We denote samples according to the stationary on U and V by $u \sim U$ and $v \sim V$ respectively. It is well known that the second largest eigenvalue of M is the same as the second largest singular value of P, and we often refer to it as the second singular value of G.

Lemma 2.3. *Let $G = (U, V, E)$ be a weighted bipartite graph whose second singular value is at most λ. Then for any functions $F : U \to [0, 1], G : V \to [0, 1]$ we have*

$$\left| \mathbb{E}_{(u,v) \sim E}[F(u) \cdot G(v)] - \mu_U(F)\mu_V(G) \right| \leqslant \lambda \sqrt{\mu_U(F)\mu_V(G)},$$

where $\mu_U := \mathbb{E}_{u \sim U}[F(u)], \mu_V(G) := \mathbb{E}_{v \sim V}[G(v)]$.
In the special case where F and G are indicator functions of sets, say of $A \subseteq U$ and $B \subseteq V$, we get

$$\left| \Pr_{(u,v) \sim E}[u \in A, v \in B] - \mu_U(A)\mu_V(B) \right| \leqslant \lambda \sqrt{\mu_U(A)\mu_V(B)}.$$

2.2.1 Constructions of Expanders and Bipartite Expanders

We will require the fact that expanders of arbitrary degree and size can be constructed efficiently.

Lemma 2.4. *For integers $n \geqslant d$ there is a polynomial time algorithm which constructs an (unweighted) d-regular graph with n vertices and second singular value $O(d^{-1/2})$.*

Proof. The algorithm is by [Alo21] if n is sufficiently large and by brute force otherwise. $\square$

We also require a version of the above lemma for bipartite, bi-regular expanders.

Lemma 2.5. *For integers n_1, n_2, d_1, d_2 such that $n_1 \cdot d_1 = n_2 \cdot d_2$ and either n_1 divides n_2 or n_2 divides n_1, there is a $\mathsf{poly}_{d_1, d_2}(n_1, n_2)$-time algorithm which constructs an (unweighted) (d_1, d_2)-bi-regular bipartite graph with n_1 vertices on the left and n_2 vertices on the right and second singular value $O\left((\min\{d_1, d_2\})^{-1/2}\right)$.*

Proof. The algorithm is by [GM21] if n_1, n_2 are sufficiently large and by brute force otherwise. $\qquad\square$

2.3 High-Dimensional Expanders

In this section, we give some necessary background on high-dimensional expanders (HDXs).

2.3.1 Simplicial Complexes

A d-dimensional high-dimensional expander (HDX) on a ground set $\mathcal{U}$ is a downward closed collection of subsets of $\mathcal{U}$, typically denoted by $X = (X(1), \ldots, X(d))$, where $X(1) = \mathcal{U}$ and $X(i)$ consists of size i subsets of $\mathcal{U}$. By downwards closed we mean that for $1 \leqslant i \leqslant j \leqslant d$, if $I \subseteq J$ have sizes i and j respectively and $J \in X(j)$, then $I \in X(i)$. We refer to $X(1)$ as the vertices of the complex, and to $X(k)$ as the k-faces of X.

For any $I \subseteq \mathcal{U}$, one may consider the sets in X containing I. These form a smaller, $(d - |I|)$-dimensional complex, which goes by the name the link of I.

Definition 2.6. *Given a set $I \in X(i)$ for $0 \leqslant i \leqslant d - 2$, the link of I is the $(d - i)$-dimensional complex X_I whose k-faces are*

$$X_I(k) = \{J \setminus I \mid J \in X(k + i), J \supseteq I\}.$$

2.3.2 Measures and Walks on Complexes

A complex $X = (X(1), \ldots, X(d))$ is equipped with a natural measure arising as the "push down" measure of the uniform measure over $X(d)$. More precisely, for each $i \in [d]$, let μ_i denote the measure over $X(i)$ obtained by choosing $U \in X(d)$ uniformly and then outputting a uniformly chosen $A \subseteq U$ of size i. At times, when i is clear from context, we drop i from the subscript and simply write μ to be the appropriate measure over $X(i)$. We will write $U \sim X(i)$ to denote a random U chosen according to the measure μ_i.

We also consider the above measure when restricted to a link of X. Specifically, for a set $I \in X(i)$ where $0 \leqslant i \leqslant d - 2$, we define the measure $\mu_{I, j-i}$ over $X_I(j - i)$ to be the measure obtained by choosing $J \sim X(j)$ conditioned on $J \supseteq I$, and then outputting $J \setminus I$. At times, if the parameter $j - i$ is clear from context, we drop it from the subscript and simply write μ_I.

These measures also induce weighted graphs between layers of the complex. Throughout the paper we will consider two graphs derived from a complex $X = (X(1), \ldots, X(d))$.

The Base Graph. This is the graph with vertex set $X(1)$ and edge set $X(2)$. The edges are weighted by the measure μ_2 over $X(2)$ and the stationary distribution of this graph is given the measure μ_1 over $X(1)$.

Inclusion Graphs. For two levels $j < k$, the inclusion graph between $X(j)$ and $X(k)$ is a bipartite graph with parts $X(j)$ and $X(k)$. The edges of this graph are all pairs $(V, U) \in X(j) \times X(k)$ such that $V \subset U$ and are weighted according to the following natural distribution derived from X: choose $U \sim X(k)$, $V \subset U$ uniformly of size j, and output (V, U).

2.3.3 Direct Product Testing on HDX

We will need direct product testers over HDXs. Fix a complex $X = (X(1), \ldots, X(d))$ and suppose that for $k \in [d]$ (thought of as a large constant but much smaller than d) there is a table $T : X(k) \to \Sigma^k$ assigning local functions on each $U \in X(k)$. We wish to determine if these individual functions assigned to each set of the complex actually correspond to some global function, $F : X(1) \to \Sigma$, meaning $T[U] = F|_U$. We will consider a natural 2-query tester, given by the following definition.

Definition 2.7. *Let X be a d-dimensional complex and let $k < d$. Given a table $T : X(k) \to \Sigma^k$, the $(k, \sqrt{k})$-direct product test can be described as follows.*

- *Sample $D \in X(d)$ uniformly.*

- *Sample $B \subseteq D$ of size $\sqrt{k}$ uniformly.*

- *Sample A, A' independently and uniformly at random conditioned on $B \subseteq A, A' \subseteq D$.*

- *The test passes if and only if $T[A]|_B = T[A']|_B$.*

Definition 2.8. *We say that the $(k, \sqrt{k})$-direct product test on X has soundness δ if the following holds. Let T be any table that passes the $(k, \sqrt{k})$-direct product test with probability at least δ. Then there exists a function $F : X(1) \to \Sigma$ such that*

$$\Pr_{A \sim X(k)}[\mathsf{dist}(T[A], F|_A) \leqslant \delta] \geqslant \mathsf{poly}(\delta).$$

In words, if a table passes the $(k, \sqrt{k})$-direct product test on X with probability at least δ, then it almost agrees with a global direct product function on a sizable weight of the k-faces.

2.3.4 An Algebraic HDX Construction

We will require an HDX construction X as in [BMVY25], which is a variant of the constructions in [DD24b, BLM24]. Some key features of this construction that are important for us include the fact they have poly-logarithmic degree, good expansion properties, and that the $(k, \sqrt{k})$-direct product test on X has arbitrarily small soundness.

Theorem 2.9. *[BMVY25, Theorem 2.13] For all $\delta \in (0, 1)$, $C \in \mathbb{N}$, for sufficiently large $k \in \mathbb{N}$ and sufficiently large $d \in \mathbb{N}$ the following holds. For any sufficiently large $n \in \mathbb{N}$ and $q = \Theta(\log^C(n))$, there is an algorithm which takes as input δ, C, d, k, n, q, and outputs in $\mathsf{poly}(n)$-time, a d-dimensional complex $X = (X(1), \ldots, X(d))$ which satisfies the properties in [BMVY25, Theorem 2.13], as well as the following properties.*

1. *$n \leqslant |X(1)| \leqslant O_d(n)$.*

2. *Every vertex is contained in at most $q^{O(d^2)}$ d-faces.*

3. *For any $1 \leqslant i \leqslant d$, the second singular value of the weighted bipartite inclusion graph between $(X(1), X(i))$ is $\frac{1}{i} + \frac{i^{O(1)}}{d}$.*

4. *For any vertex $v \in X(1)$, the second eigenvalue of the graph $(X_v(1), X_v(2))$ is at most $O(1/\sqrt{q})$.*

5. *The $(k, \sqrt{k})$-direct product test on X has soundness δ.*

We think of δ, C, d, k, n, q as inputs to this algorithm and refer to δ as the direct-product soundness parameter, C as the exponent, d as the dimension parameter, k as the direct-product dimension parameter, n as the target number of vertices, and q as the prime.

Proof. All items, except for the third and fourth, follow from the statement of [BMVY25, Theorem 2.13]. The fourth item follows from the proof in [BMVY25, Theorem 2.13], and more precisely from the paragraph "Local spectral expansion of X" in the proof. The third item follows from [BMVY25, Lemma 2.8]. $\qquad\square$

Since our arguments use several results from [BMVY25] in a black-box way, we make an implicit use of the properties of X outlined in [BMVY25, Theorem 2.13]. We thus omit a detailed description of them (as well as the set-up they require). The properties we mention in Theorem 2.9 will be used explicitly in our arguments.

We need the following (rather standard) sampling bound.

Lemma 2.10. *[BMVY25, Claim 4.7] Let X be a d-dimensional complex satisfying Theorem 2.9 with prime q, and let $\mathcal{E} \subseteq X(2)$ be a set of edges with measure $\mu_2(\mathcal{E}) \geqslant \varepsilon$ for some $\varepsilon > 0$. Then,*

$$\Pr_{u \in X_1(1)}[\mu_{u,2}(\mathcal{E}) \geqslant \sqrt{\varepsilon}] \leqslant \frac{\mathsf{poly}(d)}{q \cdot \varepsilon}.$$

3 Decodable PCPs and Uniquely-Decodable PCPs

In this section we formally define dPCPs, their unique-decoding analog called uniquely-decodable PCPs, and give a more precise version of Theorem 1.6, which is Theorem 3.12 below.

3.1 Decodable PCPs

The definition of dPCPs below expands Definition 1.3 as it includes the definitions of additional parameters of interest.

Definition 3.1 (Decodable PCP). *Let Σ_0 be an alphabet and let $\mathcal{L} \subseteq \Sigma_0^n$ be a language. A decodable PCP (dPCP) for $\mathcal{L}$ is a tuple*

$$\mathcal{D} = \left(A \cup B, E, \Sigma_A, \Sigma_B, \{\Phi_e\}_{e \in E}, \{\mathcal{P}_t\}_{t \in [n]}, \{D_t\}_{t \in [n]} \right)$$

where:

- **Constraint Graph.** $(A \cup B, E)$ *is a bipartite graph with parts A, B and edge set E.*

- **Left Alphabet.** Σ_A *is the alphabet of the vertices in A. We allow each individual vertex $a \in A$ to have a constrained alphabet, which is a subset of Σ_A.*

- **Right Alphabet.** Σ_B *is the alphabet of the vertices in B. We allow each individual vertex $b \in B$ to have a constrained alphabet, which is a subset of Σ_B.*

- **Constraints.** *For each $e = (a, b) \in E$, $\Phi_e : \Sigma_A \times \Sigma_B \to \{0, 1\}$ is a constraint on the edge e. We say that an input satisfies it if it evaluates to 1 and unsatisfied if it evaluates to 0. If either input to the constraint is an invalid alphabet symbol for a or b, then the constraint outputs 0.*

- **Projection Constraints.** *We say a constraint Φ_e is a projection constraint if for every $\sigma \in \Sigma_A$, there is a unique $\sigma' \in \Sigma_B$ such that $\Phi_e(\sigma, \sigma') = 1$. We say $\mathcal{D}$ is a projection dPCP if all of its constraints are projections.*

- **Decoding Distribution.** *For each $t \in [n]$, $\mathcal{P}_t$ is a distribution over E.*

- **Decoder.** *For each $t \in [n]$, the decoder $D_t : A \times \Sigma_A \to \Sigma_0$ receives the left vertex of a constraint along with a symbol for it, and outputs a decoding for t.*

We also define the following parameters related to dPCPs:

- **Length.** *The number of vertices in the constraint graph.*

- **Alphabet Size.** $\max(|\Sigma_A|, |\Sigma_B|)$, *and typically this will be $|\Sigma_A|$.*

- **Decision Complexity.** *Viewing $\Sigma_A = \{0, 1\}^{\log |\Sigma_A|}, \Sigma_B = \{0, 1\}^{\log |\Sigma_B|}$, the decision complexity is the maximum over all constraints (a, b) of the size of the circuit with inputs $\{0, 1\}^{\log |\Sigma_A| + \log |\Sigma_B|}$ that outputs 1 if the constraint on (a, b) is satisfied and both alphabet symbols are valid and 0 otherwise.*

- **Projection Decision Complexity.** *In the case that the dPCP is a projection dPCP, its projection decision complexity is the maximum over all constraints (a, b) of the size of the circuit with input $\sigma \in \Sigma_A$ thought of as an element in $\{0, 1\}^{\log |\Sigma_A|}$, outputs the unique Σ_B symbol which satisfies the constraint on (a, b), and outputs $\perp$ if σ is not a valid a alphabet symbol.*

  We note that the (standard) decision complexity is at most $O(\log |\Sigma_B|)$ more than the projection decision complexity.

- **Decoding Complexity.** *Viewing $\Sigma_A = \{0, 1\}^{\log |\Sigma_A|}$, the decoding complexity is maximum over all $a \in A$ of the size of the circuit which computes the decoding function $D_t(a, \cdot) : \Sigma_A \to \Sigma_0$.*

- **Neighborhoods and Degrees.** *The neighborhood of $u \in A \cup B$ is $\Gamma(u) = \{v \in A \cup B \mid (u, v) \in E\}$, and the degree of u is $\deg(u) = |\Gamma(u)|$. The left degree of $\mathcal{D}$ is $\max_{a \in A} \deg(a)$ and the right degree of $\mathcal{D}$ is $\max_{b \in B} \deg(b)$.*

- ***Decoding-Neighborhoods and Decoding-Degrees.*** *The decoding neighborhood of $a \in A$ is $\Gamma_{\mathsf{dec}}(a) = \{t \in [n] \mid a \in \mathrm{supp}\,(\mathcal{P}_t)\}$ and the decoding-degree of a is $\mathsf{ddeg}(a) = |\Gamma_{\mathsf{dec}}(a)|$. The decoding degree of $\mathcal{D}$ is $\max_{a \in A} \mathsf{ddeg}(a)$.*

We next move on to define the completeness and soundness of a dPCP. We will always use the notion of perfect completeness (all of the constructions in this paper have it).

Definition 3.2 (Perfect Completeness). *Let $\mathcal{D}$ be a decodable PCP for $\mathcal{L}$. We say that $\mathcal{D}$ has perfect completeness if for all $w \in \mathcal{L}$ there are assignments $T_A : A \to \Sigma_A$ and $T_B : B \to \Sigma_B$ such that*

$$\Pr_{\substack{t \in [n], \\ (a,b) \sim \mathcal{P}_t}} [\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge D_t(a, T_A[a]) = w_t] = 1.$$

As per the soundness notion, we use a list-decoding soundness in the style of [DH13].[3]

Definition 3.3 (List-Decoding Soundness). *Let $\mathcal{D}$ be a decodable PCP for $\mathcal{L}$. Given $\varepsilon \in [0,1]$, $L \in \mathbb{N}$ we say that $\mathcal{D}$ satisfies (L, ε)-list-decoding soundness if the following holds. For any left assignment $T_A : A \to \Sigma_A$ there exists a list $\{w_1, \ldots, w_L\} \subseteq \mathcal{L}$ such that for any $T_B : B \to \Sigma_B$,*

$$\Pr_{\substack{t \in [n], \\ (a,b) \sim \mathcal{P}_t}} [\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge D_t(a, T_A[a]) \notin \{(w_i)_t\}_{i \in [L]}] \leqslant \varepsilon.$$

Agnostic Decoding Distributions. We recall the notions of complete decoding distribution and agnosticism.

Definition 3.4 (Complete Decoding Distribution). *Let $\mathcal{D}$ be a decodable PCP for $\mathcal{L}$. The complete decoding distribution of $\mathcal{D}$, denoted by $\mathcal{Q}$, is the distribution over $[n] \times A \times B$ generated by choosing $t \in [n]$ uniformly, sampling $(a,b) \sim \mathcal{P}_t$ and outputting (t, a, b).*

Definition 3.5 (Agnostic Decoding Distribution). *For a decodable PCP, $\mathcal{D}$, we say that its complete decoding distribution $\mathcal{Q}$ is agnostic if for all $b \in B$, $t \in [n]$, the distributions $\mathcal{Q}(t, \cdot, b)$ and $\mathcal{Q}(\circ, \cdot, b)$ are the same.*

3.2 Uniquely-Decodable PCPs

We will also consider a variant of decodable PCPs called uniquely-decodable PCPs. Uniquely-decodable PCPs can be thought of as high-soundness versions of decodable-PCPs and were first introduced by Dinur and Meir [DM11]. Uniquely-decodable PCPs differ from decodable PCPs in three ways:

1. The constraint graph is not necessarily bipartite. We remark that in some cases it will be bipartite, and this will be important for us.

2. The decoding distributions are over vertices, as opposed to being over edges.

3. The constraint graph is a weighted graph and in the soundness cases edges are sampled according to this weighting. This should be thought of as the analog of the marginal distribution over edges of the complete decoding distribution $\mathcal{Q}$ of a dPCP as in Definition 3.1.

Definition 3.6 (Uniquely-Decodable PCP). *Let Σ_0 be an alphabet and let $\mathcal{L} \subseteq \Sigma_0^n$ be a language. A uniquely-decodable PCP (udPCP) for $\mathcal{L}$ is a tuple*

$$\mathcal{D} = \big(V, E, \Sigma, \{\Phi_e\}_{e \in E}, \{\mathcal{P}_t\}_{t \in [n]}, \{D_t\}_{t \in [n]}\big)$$

where:

---

[3] The more common definition in the literature asserts that the list $w_1, \ldots, w_L$ can be constructed from an assignment T_B to the right side of the given dPCP $\mathcal{D}$. We chose the notion below as it is more convenient for us to work with, and in Appendix A.1 we show a relation with the more usual one.

- **Constraint Graph:** *The constraint graph (V, E), which may have multi-edges, and we think of it as equipped with the uniform measure.*

- **Alphabet.** *Σ is the alphabet and we allow each individual vertex $v \in V$ to have a constrained alphabet $\Sigma_v \subseteq \Sigma$.*

- **Constraints.** *For each $e \in E$, $\Phi_e : \Sigma \times \Sigma \to \{0, 1\}$ is a constraint on the edge e.*

- **Decoding Distribution.** *For each $t \in [n]$, $\mathcal{P}_t$ is a distribution over V.*

- **Decoder.** *For each $t \in [n]$, the decoder $D_t : V \times \Sigma \to \Sigma_0$ receives a vertex and a symbol for it, and outputs a decoding for t.*

We define the parameters length, alphabet size, decision complexity, decoding complexity, neighborhoods, decoding-neighborhoods, degrees, and decoding degrees in the same way as in Definition 3.1.

For technical purposes, we will also want to distinguish when a uniquely-decodable PCP has bipartite constraint graph and projection constraints (as defined in Definition 3.1). In the case that it does, we will also use a notion called left-canonical, defined below.

Definition 3.7 (Bipartite and Projection Uniquely-Decodable PCP). *We say that a uniquely-decodable PCP is* bipartite *if its constraint graph is bipartite, and in this case we will denote its left and right alphabets separately. We say that a bipartite uniquely-decodable PCP has projection constraints if all of its constraints are projections from the left side to the right side.*

Definition 3.8 (Left-Canonical Uniquely-Decodable PCP). *We say that a projection bipartite uniquely-decodable PCP is left-canonical if for every left vertex, and any labeling of its right neighborhood, there is at most one assignment to the left vertex that satisfies all of the constraints involving that vertex.*

In order for the decoding distributions to be meaningful they should be aligned with the uniform distribution over E, in the following sense. Choosing $t \in [n]$, $u \sim \mathcal{P}_t$, and v a random neighbor of u (weighted by multiplicities), (u, v) is uniformly random in E. Indeed, this will always be the case for our uniquely-decodable PCP constructions. [4]

Perfect completeness and unique-decoding soundness are defined similarly to Definition 3.2 and Definition 3.3.

Definition 3.9 (Perfect Completeness). *Let $\mathcal{D}$ be a uniquely-decodable PCP for $\mathcal{L}$. We say that $\mathcal{D}$ has perfect completeness if for all $w \in \mathcal{L}$ there is an assignment $T : V \to \Sigma$ such that*

$$\Pr_{(a,b)\sim E}[\Phi_{(a,b)}(T_A[a], T_B[b])] = 1, \qquad \Pr_{t \in [n], a \sim \mathcal{P}_t}[D_t(a, T[a]) = w_t] = 1.$$

In words, the first condition says that the assignment is fully satisfying and the second condition says that the assignment always decodes according to w.

Definition 3.10 (Unique-Decoding Soundness). *Let $\mathcal{D}$ be a uniquely-decodable PCP for $\mathcal{L}$. Given $\varepsilon, \eta \in [0, 1]$, we say that $\mathcal{D}$ has (η, ε)-unique decoding soundness if the following holds. For any assignment $T : V \to \Sigma$ satisfying*

$$\Pr_{(a,b)\sim E}[\Phi_{(a,b)}(T[a], T[b]) = 1] \geqslant 1 - \varepsilon,$$

there exists $w \in \mathcal{L}$ such that

$$\Pr_{\substack{t \in [n], \\ a \sim \mathcal{P}_t}}[D_t(a, T_A[a]) \neq w_t] \leqslant \eta.$$

Finally, we define the complete decoding distribution for uniquely-decodable PCPs.

Definition 3.11 (Complete Decoding Distribution). *Let $\mathcal{D}$ be a uniquely-decodable PCP for $\mathcal{L}$. The complete decoding distribution of $\mathcal{D}$, denoted by $\mathcal{Q}$, is the distribution over $[n] \times V$ generated by choosing $t \in [n]$ uniformly, $a \sim \mathcal{P}_t$, and outputting (t, a).*

---

[4]Morally speaking, a uniquely-decodable PCP must satisfy this property, at least approximately, to have any useful notion of soundness.

3.3 Main Decodable PCP Theorem

With the definitions above in mind, below we state a more detailed version of our main dPCP construction from Theorem 1.6, specifying additional features of our new dPCP construction.

Theorem 3.12. *For all $\varepsilon > 0$ there exist $L, C \in \mathbb{N}$ such that $L = \mathsf{poly}(1/\varepsilon)$ and the following holds for all alphabets Σ_0. If $\varphi \colon \Sigma_0^n \to \{0, 1\}$ is a circuit of size N, then the language $\mathsf{SAT}(\varphi)$ has a 2-query, poly-time constructible dPCP with the following properties:*

- ***Length.*** *$O_{|\Sigma_0|}\left(N \log^C N\right)$,*

- ***Alphabet Size.*** *$O_{\varepsilon, |\Sigma_0|}(1)$,*

- ***Complete Decoding Distribution.*** *The marginal of the complete decoding distribution over the right vertices is uniform, and the marginal of the complete decoding distribution over the left vertices is also uniform.*

- ***Completeness.*** *The dPCP has perfect completeness,*

- ***Soundness.*** *(L, ε)-list-decoding soundness.*

Proof Outline. The proof of Theorem 3.12 spans Section 4 to Section 10, and proceeds as follows:

1. In Section 4 we give a uniquely-decodable PCP construction with constant alphabet size and quasi-linear length.

2. In Section 5 we show how to transform the previous result to a uniquely-decodable PCP construction over the base graph of the high-dimensional expander X from Theorem 2.9. This increases the alphabet size to be super-constant.

3. In Section 6 we perform the soundness amplification, transforming the HDX-based uniquely-decodable PCP to a 2-query dPCP with small soundness, but large alphabet size. The goal in the rest of the argument is to reduce the alphabet size of this construction.

4. In Section 7 we give dPCP variants of a few standard transformations on PCPs.

5. In Section 8 we show our decoding degree reduction transformation.

6. In Section 9 we prove our composition theorem.

7. In Section 10 we show how to combine the above tools to prove Theorem 3.12.

4 A Uniquely Decodable PCP Construction

Our starting point is a quasi-linear size uniquely-decodable PCP construction that is implicit in prior works [BS08, Din07, Mie09]. Since we are unaware of an explicit reference we give it in full detail here. We start with [Mie09, Theorem 1], stated below.

Theorem 4.1. *There exists $\rho > 0$ such that for all $\eta > 0$ the following holds. For every circuit $\varphi \colon \Sigma_0^n \to \{0, 1\}$ of size N there is a poly-time constructible PCPP $\Psi_0 = (V_0, E_0, \mathcal{P}_0, \Sigma, \{\Phi_{0,e}\}_{e \in E_0})$ for $\mathsf{SAT}(\varphi)$, such that:*

- ***Length.*** *The constraint hypergraph has $O_{|\Sigma_0|}(N \cdot \mathsf{poly}(\log(N)))$ vertices and $O_{|\Sigma_0|}(N \cdot \mathsf{poly}(\log(N)))$ hyperedges, each of size $k = O(1)$. The distribution $\mathcal{P}_0$ is uniform over the hyperedges.*

- ***Alphabet size.*** *$|\Sigma| = O_{|\Sigma_0|}(1)$.*

- ***Completeness.*** *The PCPP has perfect completeness.*

- ***Soundness.*** *If $w \in \Sigma_0^n$ is η-far from the language $\mathsf{SAT}(\varphi)$, then any assignment A extending w satisfies at most $(1 - \rho \cdot \eta)$-fraction of the constraints.*

The rest of this section is devoted to deriving from Theorem 4.1 our initial uniquely-decodable PCP construction, and the argument is organized as follows:

1. In Section 4.1 we show how to transform a PCPP into a uniquely-decodable PCP.

2. In Section 4.2 we show how to change the number of vertices in a uniquely-decodable PCP without affecting its parameters too much. This transformation will be needed in Section 5 for technical reasons.

3. In Section 4.3 we show how to make a uniquely-decodable PCP right-regular.

4. In Section 4.4 we combine the transformations in prior sections and prove Lemma 4.5, which is the uniquely-decodable PCP construction we use in subsequent sections.

4.1 Converting a PCPP into a udPCP

We now describe how to obtain a udPCP from the PCPP in Theorem 4.1. Our transformation is similar to the one in [DM11, Proposition 6.12], but it has quasi-linear length rather than quadratic length at the expense of achieving a weaker soundness guarantee.

Lemma 4.2. *There exists $k = O(1)$ such that for all $\eta > 0$ and any circuit $\varphi : \Sigma_0^n \to \{0,1\}$ of size N, there is a poly-time constructible $(k+1)$-left regular, left-canonical, projection bipartite udPCP*

$$\mathcal{D} = (A \cup B, E, \Sigma_A, \Sigma_B, \{\Phi_e\}_{e \in E}, \{D_t\}_{t \in [n]}, \{\mathcal{P}_t\}_{t \in [n]}),$$

satisfying:

- **Length.** $O_{|\Sigma_0|}(N \cdot \mathsf{poly}(\log(N)))$.

- **Alphabet Size.** *Both left and right alphabets are of size $O_{|\Sigma_0|}(1)$.*

- **Complete Decoding Distribution.** *Letting $\mathcal{Q}$ be the complete decoding distribution, the distribution $\mathcal{Q}(\circ, \cdot)$ is uniform over A.*

- **Completeness.** *The udPCP has perfect completeness.*

- **Soundness.** $(\eta, \frac{\rho \cdot \eta}{2(k+1)})$-*unique decoding soundness, where ρ is the constant from Theorem 4.1.*

Proof. Take $\Psi = (V_0, E_0, \Sigma', \{\Phi_{0,e}\}_{e \in E_0})$ to be the PCPP from Theorem 4.1. We assume without loss of generality that $|E_0|/n$ is an integer, otherwise we may add trivial edges to Ψ_0 at the cost of at most halving the constant ρ in the soundness guarantee. We construct the udPCP $\mathcal{D}$ as follows.

Vertices. The vertex set, which we denote by V, consists of two parts A, B, where $B := V_0$; we recall that $[n] \subseteq V_0$ by Definition 1.2. To construct A, we first partition the edges E_0 evenly into n parts of equal size, which we denote by $\{E_{0,t}\}_{t \in [n]}$. For each $t \in [n]$ and $e \in E_{0,t}$ we add the vertex (e, t) to A, so in total $|A| = |E_0|$. Writing an edge $e \in E_0$ as a k-tuple $e = (v_1, \dots, v_k) \in B^k$, we also think of (e, t) as $(v_1, \dots, v_k, t)$. Throughout the proof, we will refer to each v_i as a vertex in e.

Edges. For each $(e, t) \in A$ where $e = (v_1, \dots, v_k)$, we add an edge from (e, t) to t and to each one of $\{v_1, \dots, v_k\}$. Thus in total, the number of neighbors of (e, t) is $k + 1$.

Alphabets. We describe a constrained alphabet for each individual vertex. For each $v \in B$ its alphabet is Σ'. For each $(e, t) = (v_1, \dots, v_k, t) \in A$, the alphabet $\Sigma_{e,t}$ consists of all maps $F_{e,t} : \{v_1, \dots, v_k\} \cup \{t\} \to \Sigma'$ such that the assignment to e satisfies the constraint $\Phi_{0,e}$ from Ψ_0. If $v = t$ or v is a vertex in e, we will write $F_{e,t}(v)$ to denote the value of the symbol corresponding to v.

19

Constraints. For each $(e,t) \in A$ and $v \in B$ such that v is a vertex in e or $v = t$, the constraint Φ_e checks for agreement. That is, for $F_{e,t} \in \Sigma_{e,t}$ and $\sigma \in \Sigma_v$, the constraint is given by

$$\Phi_{((e,t),v)}(F_{e,t}, \sigma) = 1 \iff F_{e,t}(v) = \sigma.$$

Decoding Distributions. For each $t \in [n]$, the decoding distribution $\mathcal{P}_t$ chooses a uniform $(e,t) \in \{(e,t) \mid e \in E_{0,t}\}$. One can check that choosing $t \in [n]$ uniformly, and (e,t) according to the marginal of $\mathcal{P}_t$, we have (e,t) is uniform in A. This is because we split $|E_0|$ into n equal parts before appending the edges of each part with a unique $t \in [n]$. Hence the condition on the complete decoding distribution is satisfied.

Decoder. The decoder is defined as $D_t((e,t), F_{e,t}) = F_{e,t}(t)$.

We now move on to the analysis of of $\mathcal{D}$. The length of our construction is $|A| + |B| = |V_0| + |E_0|$ and this is $N \cdot \text{poly}(\log(N))$. It is also clear that the alphabet size is $O(1)$, that the constraint graph is $(k+1)$-left regular and bipartite, that the constraints are projections, and that the udPCP is left-canonical.

Perfect Completeness. Fix $w \in \mathcal{L}$; then by the completeness of Ψ, there is an assignment $A_0 : V_0 \to \Sigma'$ which satisfies all of the constraints of Ψ_0 and is equal to w on the vertices of V_0 associated with $[n]$. Define the assignment $F : A \to B$ by setting $F(v) = A_0(v)$ for each $v \in B$, and setting $F((e,t))$ to be the restriction of A_0 to the vertices in (e,t) for each $(e,t) \in A$. It is easy to see that F assigns a valid alphabet symbol to each vertex, that it satisfies all of the constraints in $\mathcal{D}$, and that for any t and $(u,v) \in \text{supp}(\mathcal{P}_t)$, we have $D_t(u, F(u)) = w_t$.

Unique Decoding Soundness. Fix $\eta > 0$, set $\varepsilon = \frac{\rho \cdot \eta}{2k}$ and suppose $A : V \to \Sigma$ is an assignment to $\mathcal{D}$ that satisfies at least $(1 - \varepsilon)$-fraction of the constraints. We will show that there exists $w \in \mathcal{L}$ such that the decoding error with respect to this w is at most η.

For each $(e,t) \in A$ define $p(e,t) = \Pr_{v \in (e,t)}[A(v) \neq A(e,t)(v)]$, and note that the assumption that A satisfies at least $(1 - \varepsilon)$-fraction can be written as $\mathbb{E}_{(e,t) \in A}[p(e,t)] \leqslant \varepsilon$. Note that if $p(e,t) > 0$ for some (e,t), then it must be the case that $p(e,t) \geqslant \frac{1}{k+1}$. Applying Markov's inequality we conclude that

$$\Pr_{(e,t) \in A}[p(e,t) = 0] \geqslant 1 - (k+1)\varepsilon. \tag{3}$$

In words, for at least $(1 - (k+1)\varepsilon)$-fraction of $(e,t) \in A$, under the assignment A, there is perfect agreement with all v appearing in e and with $v = t$. Consider the restriction $A_0 = A|_B$ and view it as an assignment to the PCPP Ψ_0. Then

$$\Pr_{e \in E_0}[\Phi_{0,e}(A_0(e)) = 1] \geqslant \Pr_{(e,t) \in A}[p(e,t) = 0] \geqslant 1 - (k+1)\varepsilon \geqslant 1 - \frac{\rho \cdot \eta}{2}, \tag{4}$$

so A_0 satisfies at least $(1 - \rho\eta/2)$-fraction of the constraints of Ψ_0. Here, in the first transition we used the fact that for a uniformly random $(e,t) \in E$, the distribution of e is uniform over E_0, and also that if $p(e,t) = 0$, then A_0 satisfies the constraint $\Phi_{0,e}$. Combining (4) with the soundness guarantee of Theorem 4.1 we get that the restriction of $A_0|_X$ is $\frac{\eta}{2}$-close to $\mathcal{L}$. Namely, there exists $w \in \mathcal{L}$ such that

$$\Pr_{t \in [n]}[A_0(t) \neq w_t] \leqslant \frac{\eta}{2}. \tag{5}$$

The decoding error of Ψ_0 can now be bounded by:

$$\begin{aligned}
\Pr_{t \in [n], (e,t) \sim \mathcal{P}_t}[D_t((e,t), A(e,t)) \neq w_t] &= \Pr_{(e,t) \in E}[A(e,t)(t) \neq w_t] \\
&\leqslant \Pr_{(e,t) \in E}[A(e,t)(t) \neq w_t \wedge A(e,t) = A_0|_{(e,t)}] \\
&\quad + \Pr_{(e,t) \in E}[A(e,t) \neq A_0|_{(e,t)}] \\
&\leqslant \frac{\eta}{2} + (k+1)\varepsilon \\
&\leqslant \eta.
\end{aligned}$$

20

By $A_0|_{(e,t)}$, we mean the restriction of A_0 to the vertices in e and t. In the last transition we bound the first term using (5) and use the fact that when choosing $(e, t) \in E$ the element t from the tuple is uniformly random in $[n]$. We bound the second term by the probability that $p(e, t) > 0$, which is at most $(k+1)\varepsilon$ by (3). $\qquad \square$

4.2 Enlarging the Constraint Graph

Here, we describe how to enlarge the constraint graph of a bipartite, projection, left-canonical udPCP. When working with standard PCPs, one can simply create multiple disjoint copies of the constraint graph and be done. It is not hard to see that essentially all properties, including the soundness, of the PCP are preserved when copying the constraint graph. This transformation does not work for udPCPs, since it does not necessarily preserve unique decoding soundness. Indeed, there could be two completely different, fully satisfying assignments on each copy that decode to different members of the language. We circumvent this issue by using the left-canonical property.

Lemma 4.3. *Suppose that*

$$\mathcal{D} = (A \cup B, E, \Sigma_A, \Sigma_B, \{\Phi_e\}_{e \in E}, \{D_t\}_{t \in [n]}, \{\mathcal{P}_t\}_{t \in [n]})$$

is a bipartite, k-left-regular, projection, left-canonical udPCP for the language $\mathcal{L} \subseteq \Sigma_0^n$ with (η_1, η_2)-unique decoding soundness. Suppose additionally that the marginal of the complete decoding distribution over the vertices is uniform in A. Then for all $C \in \mathbb{N}$, there is a $\mathsf{poly}(C, |A| + |B|)$-time algorithm which takes as input $\mathcal{D}$ and outputs a udPCP for $\mathcal{L}$,

$$\mathcal{D}' = (A \times [C] \cup B, E', \Sigma_A, \Sigma_B, \{\Phi'_e\}_{e \in E'}, \{\mathcal{P}'_t\}, \{D_t\}_{t \in [n]})$$

with the following properties:

- **Constraint graph.** *The constraint graph has $C|A| + |B|$ vertices, and each $(a, i) \in A \times [C]$ is adjacent to the same vertices that a is adjacent to in the original constraint graph.*

- **Degrees.** *The left degree and the left decoding degree of each vertex is preserved, while the right degree of each vertex is multiplied by C.*

- **Soundness.** *$\mathcal{D}'$ has $(\eta_1 + \eta_2, \frac{\eta_2}{2k})$-unique decoding soundness.*

Furthermore, the completeness, decision complexity, decoding complexity, and alphabet sizes of $\mathcal{D}$ and $\mathcal{D}'$ are the same, and the marginal of the complete decoding distribution over the vertices is uniform over $A \times [C]$. If $\mathcal{D}$ has projection constraints then $\mathcal{D}'$ does as well and if $\mathcal{D}$ is left-canonical then $\mathcal{D}'$ is as well.

Proof. Construct the new udPCP

$$\mathcal{D}' = (A \times [C] \cup B, E', \{\mathcal{P}'_t\}_{t \in [n]}, \Sigma_A, \Sigma_B, \{\Phi'_e\}_{e \in E'}, \{D'_t\}_{t \in [n]})$$

as follows. The edges are $E' = \{((a, i), b) \in (A \times [C]) \times B \mid (a, b) \in E\}$. The constraint on $((a, i), b)$ is the same as that on (a, b) in the original udPCP, $\mathcal{D}$. The new decoding distribution $\mathcal{P}'_t$ is generated by choosing $a \sim \mathcal{P}_t$, choosing $i \in [C]$ uniformly at random, and outputting (a, i). The new decoder D'_t on vertex (a, i) is the same as the original decoder D_t on vertex a.

It is clear that completeness is preserved, the alphabets are unchanged, the distribution of $(a, i) \in A \times [C]$ is uniform in the complete decoding distribution of $\mathcal{D}'$. It is also straightforward to see that $\mathcal{D}'$ has projection constraints if $\mathcal{D}$ has projection constraints and $\mathcal{D}'$ is left-canonical if $\mathcal{D}$ is left-canonical. Furthermore, the conditions regarding the constraint graph, and degrees are straightforward to check.

We complete the proof by showing $\mathcal{D}$ has $(\eta_1 + \eta_2, \frac{\eta_2}{2k})$-unique decoding soundness. Suppose $T_1 : A \times [C] \to \Sigma_A$ and $T_2 : B \to \Sigma_B$ are a $(1 - \eta')$-satisfying assignments to $\mathcal{D}'$ for $\eta' := \frac{\eta_2}{2k}$. For each $(a, i) \in A \times [C]$, let $p(a, i)$ denote the fraction of constraints involving (a, i) that are not satisfied. Since each (a, i) has degree at most k, we have that if $p(a, i) \neq 0$ then $p(a, i) \geqslant \frac{1}{k}$, and $\mathbb{E}_{(a,i) \in A \times [C]}[p(a, i)] \leqslant \eta'$. By Markov's inequality, we have that

$$\Pr_{(a,i) \in A \times [C]}[p(a, i) = 0] = \mathbb{E}_{a \in A}\left[\Pr_{i \in [C]}[p(a, i) = 0]\right] \geqslant 1 - k\eta'. \tag{6}$$

21

By Markov's inequality again, we have that for at least $(1 - k\eta')$-fraction of $a \in A$, $\Pr_{i \in [C]}[p(a, i) = 0] > 0$, and hence for $(1 - k\eta')$-fraction of a, there exists an assignment in $\mathcal{D}$ to a which satisfies all of the constraints involving a conditioned on the right side being assigned T_2. We call such a good. Let $T^* : A \to \Sigma_A$ be the table such that $T^*[a] = T_1[(a, i)]$ if there exists $i \in [C]$ such that $p(a, i) = 0$, and $T^*[a]$ is arbitrary otherwise. Note that since $\mathcal{D}$ is left-canonical, if a is good then $T_1[(a, i)]$ is the same for all $i \in [C]$ such that $p(a, i) = 0$, so the assignment T^* is well defined.

It follows that for the T^*, T_2 satisfy $(1 - k\eta')$-fraction of the constraints in $\mathcal{D}$. Since $k\eta' \leqslant \eta_2$ and $\mathcal{D}$ satisfies (η_1, η_2)-unique decoding, we conclude that there exists $w \in \mathcal{L}$ such that

$$\Pr_{t \in [n], a \sim \mathcal{P}_t}[D_t(a, T^*[a]) \neq w_t] \leqslant \eta_1. \tag{7}$$

We now bound the decoding error of T_1, T_2 in $\mathcal{D}'$ as

$$\Pr_{t \in [n], a \sim \mathcal{P}_t, i \in [C]}[D_t((a, i), T_2[(a, i)]) \neq w_t]$$
$$\leqslant \Pr_{t \in [n], a \sim \mathcal{P}_t, i \in [C]}[T_2[(a, i)] \neq T^*[a]] + \Pr_{t \in [n], a \sim \mathcal{P}_t, i \in [C]}[D_t(a, T^*[a]) \neq w_t]$$
$$\leqslant \Pr_{a \in A, i \in [C]}[a \text{ is not good} \vee p(a, i) \neq 0] + \eta_1$$
$$\leqslant 2k\eta' + \eta_1$$
$$= \eta_1 + \eta_2$$

In the second transition we are using the assumption that $a \in A$ is uniform in the complete decoding distribution of $\mathcal{D}$ and that if a is good and $p(a, i) = 0$, then we must have $T^*[a] = T_2[(a, i)]$ by definition of T^*. In the third transition we are using (6), (7) and the fact that at least $(1 - k\eta')$-fraction of $a \in A$ are good. $\qquad\square$

4.3   Making the udPCP Right-Regular

The next step is to convert the construction in Lemma 4.2 into a bi-regular one. Towards this end we use the following transformation (similar transformations have been used in the context of standard PCPs in [MR10, DH13]).

Lemma 4.4. *Suppose $\mathcal{L}$ has a k-left-regular projection udPCP $\mathcal{D} = \left(A \cup B, E, \Sigma_A, \Sigma_B, \{D_t\}_{t \in [n]}, \{\mathcal{P}_t\}_{t \in [n]}\right)$ with (η, ε)-unique decoding soundness and minimum right degree d_0. Then for every integer $d' \leqslant d_0$, there is a polynomial time transformation which takes $\mathcal{D}$ and outputs a udPCP, $\mathcal{D}' = \left(A \cup B', E', \Sigma_A, \Sigma_B, \{D_t\}_{t \in [n]}, \{\mathcal{P}'_t\}_{t \in [n]}\right)$, for $\mathcal{L}$ that is $(d'k, d')$-regular and has $(\eta, \varepsilon - O(d'^{-1/2}))$-unique decoding soundness. Furthermore $|B'| = |E| = k \cdot |A|$ and if $\mathcal{D}$ is left-canonical, then $\mathcal{D}'$ is as well.*

Proof. Fix d' as in the statement. By Lemma 2.5 we get that for every $N' \geqslant d'$ there is a poly(N')-time algorithm that constructs a d'-regular bipartite expander graph with N' vertices on each side and second singular value $O(d'^{-1/2})$, and we will use these graphs in our construction.

The udPCP $\mathcal{D}'$ is constructed as follows. Recall that $(A \cup B, E)$ is the constraint graph of $\mathcal{D}$ and that for each vertex $b \in B$, $\Gamma(b)$ is the neighborhood of b in this graph. For each $b \in B$ we construct a d'-regular bipartite expander graph $H_b = (A_b, B_b, E_b)$ with second singular value $O(d'^{-1/2})$, and we identify A_b with $\Gamma(b)$. With these expander graphs in mind, the left side of $\mathcal{D}'$ is A, while the right side of $\mathcal{D}'$ is $\bigcup_{b \in B} B_b$. The edges, E', are obtained as follows. For each $b \in B$, add the edges E_b from the graph H_b, where the left side, A_b, is identified with $\Gamma(b)$ and the right side is B_b. Overall, the edges are $E' = \cup_{b \in B} E_b$. Note that choosing $(a, b) \in E$ uniformly at random and $b' \in B_b$ adjacent to a uniformly at random, the edge $(a, b') \in E'$ is uniformly random.

The left and right alphabets of $\mathcal{D}'$ are still Σ_A, Σ_B and the decoders of $\mathcal{D}'$ are still $\{D_t\}_{t \in [n]}$. For each $(a, b') \in E'$, if $b' \in B_b$, the constraint on (a, b') is the same as the constraint on (a, b) in $\mathcal{D}$. Finally the decoding distributions for $\mathcal{D}'$ are the same as in $\mathcal{D}$.

This completes the description of $\mathcal{D}'$, and we now move on to the analysis. It is clear from the construction that $|B'| = \sum_{b \in B} |\Gamma(b)| = |E| = k|A|$. Also, $\mathcal{D}'$ is $(d'k, d')$-regular as each neighbor b of $a \in A$ in $\mathcal{D}$ contributes d' neighbors for a in $\mathcal{D}'$. We move on to showing that the left-canonical property is preserved (if $\mathcal{D}$ satisfies it) as well as verifying completeness and soundness. Throughout, we write $(a', b') \sim H_b$ to denote a uniformly random edge from H_b.

The left-canonical property is preserved: suppose $\mathcal{D}$ is left-canonical, and consider any vertex $a \in A$, let $b_1, \ldots, b_k$ be neighbors of a in $\mathcal{D}$, and let $b_1', \ldots, b_k'$ be neighbors of a in $\mathcal{D}'$, where $b_i' \in B_{b_i}$ for each $i \in [k]$. Fix any assignment to $b_1', \ldots, b_k'$, and consider the same assignment to $b_1, \ldots, b_k$. Since $\mathcal{D}$ is left-canonical, there is at most one assignment to a satisfying all of the constraints in $\mathcal{D}$, and since the constraints between (a, b_i) in $\mathcal{D}$ and (a, b_i') in $\mathcal{D}'$ are the same, there is also at most one assignment to a satisfying all of the constraints in $\mathcal{D}'$. Hence, $\mathcal{D}'$ is left-canonical as well.

Perfect Completeness: fix $w \in \mathcal{L}$; by the perfect completeness of $\mathcal{D}$, there are assignments $T_1 : A \to \Sigma_A$ and $T_2 : B \to \Sigma_B$ that satisfy all of the constraints in $\mathcal{D}$, and additionally for each $t \in [n]$, the decoder D_t outputs w_t with probability 1. We take the assignments T_1 and T_2' to $\mathcal{D}'$, where $T_2' : B' \to \Sigma_B$ is defined by $T_2'[b'] = T_2[b]$ if $b' \in B_b$. It is clear that T_1 and T_2' satisfy all of the constraints in $\mathcal{D}'$ and that for all $t \in [n]$, the decoder D_t' outputs w_t with probability 1.

Unique Decoding Soundness: fix assignments $T_1 : A \to \Sigma_A$ and $T_2' : B' \to \Sigma_B$ to $\mathcal{D}'$, and suppose that T_1, T_2' satisfy at least $(1 - \varepsilon')$-fraction of the constraints, where $\varepsilon' := \varepsilon - O(d^{-1/2})$. For each $b \in B$ and $\sigma \in \Sigma_B$ define

$$X_{b,\sigma} = \{b' \in B_b \mid T_2'[b'] = \sigma\} \quad \text{and} \quad Y_{b,\sigma} = \{a' \in A_b \mid \Phi_{(a',b)}(T_1[a'], \sigma) = 1\}.$$

In words, $X_{b,\sigma}$ is the set of vertices in B_b that are assigned σ under T_2', and $Y_{b,\sigma}$ is the set of vertices in $a' \in \Gamma(b)$ such that the assignments $a \to T_1[a']$ and $b \to \sigma$ satisfy the constraint on (a', b) in $\mathcal{D}$. It is clear that for each $b \in b$, the sets $\{X_{b,\sigma}\}_{\sigma \in \Sigma_B}$ form a partition of B_b. Additionally, since the constraints in $\mathcal{D}$ are all projections, for all $b \in B$ the sets $\{Y_{b,\sigma}\}_{\sigma \in \Sigma_B}$ form a partition of A_b. Define $F_{b,\sigma} : A_b \to [0, 1]$, $G_{b,\sigma} : B_b \to \{0, 1\}$ by

$$G_{b,\sigma}(b') := \mathbb{1}_{b' \in X_{b,\sigma}} \quad \text{and} \quad F_{b,\sigma}(a') := \mathbb{1}_{a' \in Y_{b,\sigma}},$$

and denote $\mu(F_{b,\sigma}) = \mathbb{E}_{a \in A_b}[F_{b,\sigma}(a')]$ and $\mu(G_{b,\sigma}) = \mathbb{E}_{b' \in B_b}[G_{b,\sigma}(b')]$. Consider the randomized assignment $T_2^* : B \to \Sigma_B$ obtained by setting $T_2^*(b) = \sigma$ with probability $\mu(G_{b,\sigma})$. The expected fraction of constraints in $\mathcal{D}$ satisfied by T_1 and T_2^* is

$$\mathbb{E}_{b \sim B}\left[\sum_{\sigma \in \Sigma_B} \mu(G_{b,\sigma}) \cdot \mu(F_{b,\sigma})\right], \tag{8}$$

where $b \sim B$ is sampled according to the stationary distribution over $b \in B$ in the constraint graph of $\mathcal{D}$. Next, the fraction of constraints satisfied by T_1 and T_2 in $\mathcal{D}'$ is

$$\mathbb{E}_{b \sim B}\left[\sum_{\sigma \in \Sigma_B} \mathbb{E}_{(a',b') \sim H_b}[F_{b,\sigma}(a') \cdot G_{b,\sigma}(b')]\right] \geqslant 1 - \varepsilon', \tag{9}$$

where the inequality is because we assumed that T_1, T_2 satisfy at least $(1 - \varepsilon')$-fraction of constraints in $\mathcal{D}'$.

Now fix a $b \in B$ and a $\sigma \in \Sigma_B$. We will attempt to relate the inner expectation from (9) to a product in the summation of (8). By Lemma 2.3 we get

$$\mathbb{E}_{(a',b') \sim H_b}[F_{b,\sigma}(a') \cdot G_{b,\sigma}(b')] \leqslant \mu(F_{b,\sigma}) \cdot \mu(G_{b,\sigma}) + O(d'^{-1/2}) \cdot \sqrt{\mu(F_{b,\sigma}) \cdot \mu(G_{b,\sigma})}.$$

Keeping $b \in B$ fixed and summing this inequality over all $\sigma \in \Sigma_B$, we get

$$\sum_{\sigma \in \Sigma_B} \mathbb{E}_{(a',b') \sim H_b}[F_{b,\sigma}(a') \cdot G_{b,\sigma}(b')] \leqslant \sum_{\sigma \in \Sigma_B} \mu(F_{b,\sigma}) \cdot \mu(G_{b,\sigma}) + O(d'^{-1/2}) \cdot \sqrt{\mu(F_{b,\sigma}) \cdot \mu(G_{b,\sigma})}$$

$$\leqslant \sum_{\sigma \in \Sigma_B} \mu(F_{b,\sigma}) \cdot \mu(G_{b,\sigma}) + O(d'^{-1/2}),$$

where the second inequality uses Cauchy-Schwarz and the fact that $\sum_{\sigma \in \Sigma_B} \mu(G_{b,\sigma}) \leqslant 1$ and $\sum_{\sigma \in \Sigma_B} \mu(F_{b,\sigma}) = 1$. Thus, we have

$$\mathbb{E}_{b \sim B}\left[\sum_{\sigma \in \Sigma_B} \mu(F_{b,\sigma}) \cdot \mu(G_{b,\sigma})]\right] \geqslant \mathbb{E}_{b \sim B}\left[\sum_{\sigma \in \Sigma_B} \mathbb{E}_{(a',b') \sim H_b}[F_{b,\sigma}(a') \cdot G_{b,\sigma}(b')]\right] - O(d'^{-1/2}) \geqslant 1 - \varepsilon' - O(d'^{-1/2}),$$

which is at least $1 - \varepsilon$ by the choice of ε'. It follows that there exists a choice for $T_2^* : B \to \Sigma_B$ such that T_1 and T_2^* satisfy at least $(1 - \varepsilon)$-fraction of constraints in $\mathcal{D}$. By the (η, ε)-unique decoding of $\mathcal{D}$, there exists $w \in \mathcal{L}$ such that

$$\Pr_{t \in [n], a \sim \mathcal{P}_t}[D_t(a, T_1[a]) \neq w_t] \leqslant \eta.$$

Notice that this is exactly the same as the decoding error of $\mathcal{D}'$ with assignments T_1, T_2. This is because for all t, D_t and D_t' are the same function and the marginal distribution over A in $\mathcal{P}_t$ and $\mathcal{P}_t'$ is the same. Thus, we have

$$\Pr_{t \in [n], a' \sim \mathcal{P}_t'}[D_t'(a', T_1[a']) \neq w_t] = \Pr_{t \in [n], a \sim \mathcal{P}_t}[D_t(a, T_1[a]) \neq w_t] \leqslant \eta.$$

It follows that $\mathcal{D}'$ satisfies $(\eta, \varepsilon - O(d^{-1/2}))$-unique decoding soundness. $\qquad\square$

4.4 A Bi-regular Uniquely-decodable PCP of Arbitrary Size

In this section we put all of the previous pieces together to obtain a final udPCP that will go into the routing transformation of the next round. We highlight the following important properties:

- The constraint graph is d'-regular and bipartite.

- The number of vertices in the constraint graph can be multiplied by any arbitrary $C \in \mathbb{N}$.

- The marginal of the complete decoding distribution over the vertices is uniform.

We remark that the last item above is in contrast to our original udPCP, from Lemma 4.2, whose complete decoding distribution has a marginal which is uniform over only a subset of the vertices.

Lemma 4.5. *There is $\rho' > 0$ such that for every $\eta > 0$ there is a sufficiently large $d' \in \mathbb{N}$ such that the following holds. For any alphabet Σ_0, there exists a function $s(N) = O_{|\Sigma_0|, \eta}(N \cdot \mathrm{poly}(\log(N)))$ such that for any $C \in \mathbb{N}$ and any size-N circuit $\varphi : \Sigma_0^n \to \{0, 1\}$, the language $\mathsf{SAT}(\varphi)$ has a udPCP which satisfies the following properties:*

- ***Constraint Graph.*** *The constraint graph is a d'-regular bipartite graph.*

- ***Length.*** *$C \cdot s(N)$.*

- ***Alphabet Size.*** *$O_{|\Sigma_0|}(1)$.*

- ***Degrees.*** *The left and right degree are both d' and the decoding degree is 1.*

- ***Complete Decoding Distribution.*** *The marginal of the complete decoding distribution over the vertices is uniform.*

- ***Completeness.*** *The udPCP has perfect completeness.*

- ***Soundness.*** *The udPCP has $(\eta, \rho' \cdot \eta^5)$-unique decoding soundness.*

Proof. Fix $\eta > 0$ and take the bipartite, $(k+1)$-left regular, projection, left-canonical udPCP $\mathcal{D}$, from Lemma 4.2 with $\left(\eta, \frac{\rho \cdot \eta}{2(k+1)}\right)$-unique decoding soundness. Let $(A \cup B, E)$ be the constraint graph of this udPCP, set $d' = \Theta(\eta^{-2})$, and fix an arbitrary $C \in \mathbb{N}$. Also set $s(N) = 2d'(k+1)|A| = O_{|\Sigma_0|, \eta}(N \cdot \mathrm{poly}(\log(N)))$. Let $\{D_t\}_{t \in [n]}$ and $\{\mathcal{P}_t\}_{t \in [n]}$ be the decoders and decoding distributions respectively of $\mathcal{D}$.

24

We start by repeating every edge in $\mathcal{D}$ d'-times so that the minimum degree on the right hand side is d'. Now apply Lemma 4.3 with repetition parameter C. At this point, we get a $d'(k+1)$-left regular udPCP which has $C|A|$ vertices on the left, $|B|$ vertices on the right, and $(\eta + \frac{\rho\eta}{2(k+1)}, \frac{\rho\eta}{4(k+1)^2})$-unique decoding soundness.

Next, we apply Lemma 4.4 with degree parameter $d' = O(\eta^{-2})$ chosen sufficiently large so that $O(d'^{-1/2})$ is sufficiently small relative to $\frac{\rho\eta}{4(k+1)^2}$. Let $R = d'^2(k+1)$. The resulting udPCP's constraint graph has $C|A|$ vertices on the left, $C|A|d'(k+1)$ vertices on the right, is (R, d')-bi-regular, and has $(\eta + \frac{\rho\eta}{2(k+1)}, \frac{\rho\eta}{5(k+1)^2})$-unique decoding soundness.

We then apply Lemma 4.3 one more time with repetition parameter $d'(k+1)$ to obtain an R-regular udPCP. Since the constraint graph is fully regular, the number of vertices is just twice the number on either side, so the length of the resulting udPCP is $2Cd'(k+1)|A| = C \cdot s(N)$. Let $\mathcal{D}'$ be the resulting udPCP and $G' = (A' \cup B', E')$ be its constraint graph. Note that G' is R-regular and $|A'| = |B'|$. Since C was arbitrary, we can indeed obtain length $C \cdot s(N)$ for any $C \in \mathbb{N}$ for $\mathcal{D}'$. Since ρ and k are constants and d' is large enough, one can check through Lemma 4.3 that $\mathcal{D}'$ has $(1.1\eta, \frac{\rho\eta}{5(k+1)^2 R})$-unique decoding soundness. Here we use the fact that

$$\eta + \frac{\rho\eta}{2(k+1)} + \frac{\rho\eta}{5(k+1)^2} \leqslant 1.1\eta.$$

It is also straightforward to verify that perfect completeness is preserved from $\mathcal{D}$, the alphabets are the same as $\mathcal{D}$, that the left decoding degree is still 1 as in $\mathcal{D}$, and that the marginal of the complete decoding distribution over the vertices is uniform over A, the left side of the constraint graph.

Now we will further modify $\mathcal{D}'$, to obtain a new udPCP, $\mathcal{D}''$ whose parts are as follows. In the below, recall that each $a \in A$ has decoding degree 1, so we denote by $t(a) \in [n]$ its only decoding neighbor.

- **Constraint Graph.** Starting with G', the constraint graph of $\mathcal{D}'$, choose an arbitrary matching $\pi : A' \to B'$, meaning the edges of the matching are $\{(a, \pi(a))\}_{a \in A'}$. Add R copies of this matching to the graph (meaning between each a and $\pi(a)$, we add R parallel edges). The resulting $2R$-regular graph, $G'' = (A' \cup B', E'')$, is the constraint graph of $\mathcal{D}''$.

- **Alphabet.** The alphabet for each vertex in A' is the same as in $\mathcal{D}'$. For each right vertex $b \in B'$, we require its alphabet to additionally include one Σ_0 symbol. It is intended that this symbol holds the $t(\pi^{-1}(b))$-th index of some satisfying assignment.

- **Decoding Distribution.** For each $t \in [n]$, the decoding distribution $\mathcal{P}''_t$ now samples $a \sim \mathcal{P}_t$, and then outputs one of a or $\pi(a)$ uniformly at random.

- **Decoder.** For each $a \in A'$ and symbol σ for a, the decoder $D''_t(a, \sigma)$ is unchanged and outputs $D_t(a, \sigma)$. For $b = \pi(a) \in B'$, the decoder outputs the Σ_0 symbol held in b's symbol (we note that it supposedly corresponds to what the decoder would have output if given the vertex $t(\pi^{-1}(b)) = t(a)$ and its label).

- **Constraints.** For each $(a, b) \in E''$, if $b \neq \pi(a)$, then the constraint on (a, b) checks the constraint from $\mathcal{D}'$. Specifically, given labels σ_a for a and σ_b for b is as follows. Recall that σ_b contains a label for b in the udPCP $\mathcal{D}'$. Let σ'_b be this symbol. The constraint checks if σ_a, σ'_b satisfy the constraint on (a, b) in $\mathcal{D}'$. If $b = \pi(a)$, then the constraint checks the previous condition and in addition checks that the symbols held by a and $\pi(a)$ lead to the same decoding under D'_t. That is, given symbols σ, σ', the constraint additionally checks if $D'_t(a, \sigma) = D'_t(b, \sigma')$.

We now verify that the guarantees of the lemma are still satisfied after the above modification. It is clear that the length is unchanged, the alphabet size only increases by a constant factor, the constraint graph is $2R$-regular, every vertex has decoding degree 1, and perfect completeness is preserved. Regarding the new decoding distribution, it is clear that choosing $t \in [n]$ and sampling $v \sim \mathcal{P}''_t$, we have that v is uniformly random in $A'' \cup B''$. This is because, choosing $t \in [n]$ and $a \sim \mathcal{P}'_t$ we have that a is uniformly random in the left side. Since π is a matching, if we then output a or $\pi(a)$ uniformly at random, the resulting vertex is uniform in the whole graph.

Finally, we show that $\mathcal{D}''$ satisfies $(1.11\eta, \frac{\rho\eta}{10(k+1)^2 R})$ unique decoding soundness. Suppose T is an assignment to $A' \cup B'$ satisfying at least $(1 - \frac{\rho\eta}{10(k+1)^2 R})$-fraction of the constraints in $\mathcal{D}'$. Abusing notation, we will also think of T as an assignment to $\mathcal{D}$ by removing the Σ_0 symbol attached to each $T(b)$.

Let us split the constraints into E_0, consisting of the constraints in $\mathcal{D}$, and E_1 consisting of the new constraints added from the matching. Since each of E_0, E_1 consists of half of the overall constraints, we get that the fraction of satisfied constraints in both E_0, E_1 is at least $\left(1 - \frac{\rho\eta}{5(k+1)^2 R}\right)$-fraction of the constraints there, and hence applying the unique-decoding soundness of $\mathcal{D}'$, we get that there exists a $w \in \mathsf{SAT}(\varphi)$ such that

$$\Pr_{t \in [n], a \sim \mathcal{P}'_t}[D'_t(a, T[a]) \neq w_t] \leqslant 1.1\eta. \tag{10}$$

On the other hand the decoding error of T in $\mathcal{D}''$ relative to w can be bounded by

$$\begin{aligned}
&\Pr_{t \in [n], a \sim \mathcal{P}'_t}[D''_t(a, T[a]) \neq w_t \vee D''_t(\pi(a), T[\pi(a)]) \neq w_t] \\
&\leqslant \Pr_{t \in [n], a \sim \mathcal{P}'_t}[D'_t(a, T[a]) \neq w_t] + \Pr_{t \in [n], a \sim \mathcal{P}'_t}[D'_t(\pi(a), T[\pi(a)]) \neq D'_t(a, T[a])] \\
&\leqslant 1.1\eta + \frac{\rho\eta}{5(k+1)^2 R} \\
&\leqslant 1.11\eta,
\end{aligned} \tag{11}$$

where in the third transition we bound the first term using (10) and the second term using the fact that it is at most the fraction of unsatisfied constraints in E_1. This shows that $\mathcal{D}''$ has $(1.11\eta, \frac{\rho\eta}{10(k+1)^2 R})$-unique decoding soundness which is as desired because $R = O(\eta^{-2})$. $\qquad\qquad\square$

5 Routing Onto an HDX

In this section, we show how to transform the udPCP construction from Lemma 4.5 into a udPCP whose constraint graph is the base layer of an HDX from Theorem 2.9, and for that we use the fault-tolerant routing protocol of [BMVY25]. The main difference is that we analyze unique-decoding soundness, whereas [BMVY25] analyze the standard notion of soundness.

Theorem 5.1. *For every sufficiently small $\eta > 0$, any $\delta \in (0, 1)$, sufficiently large C', sufficiently large $k \in \mathbb{N}$ and sufficiently large $d \in \mathbb{N}$ the following holds. Any circuit $\varphi : \Sigma_0^n \to \{0, 1\}$ of size N has a udPCP $\mathcal{D}$ which satisfies the following properties:*

- ***Constraint graph.** Let X be the complex constructed from the algorithm in Theorem 2.9 where the direct product soundness, exponent, direct-product dimension, target number of vertices, and prime parameters are set to δ, C', k, d, $N \cdot \mathsf{poly}(\log N)$, $\Theta(\log^{C'}(N))$ respectively. The constraint graph of $\mathcal{D}$ is $(X(1), X(2))$, where the multiplicity of each edge (u, v) is proportional to the weight of (u, v) in stationary distribution over $(X(1), X(2))$. The udPCP has length $|X(1)| = O_{|\Sigma_0|, \eta, \delta}(N\mathsf{poly}(\log(N))$.*

- ***Degrees.** Each vertex in the constraint graph has degree $\mathsf{poly}_\delta(\log N)$ and decoding degree $\mathsf{poly}_\delta(\log N)$.*

- ***Alphabet Size.** $O_{|\Sigma_0|}\left(2^{\mathsf{poly}_{\eta,\delta}(\log(N))}\right)$.*

- ***Decision Complexity.** $O_{|\Sigma_0|}(\mathsf{poly}_{\eta,\delta}(\log(N)))$.*

- ***Decoding Distribution.** The marginal of the complete decoding distribution over the vertices is the stationary distribution of $(X(1), X(2))$ over $X(1)$.*

- ***Decoding Complexity.** $O_{|\Sigma_0|}(\mathsf{poly}_{\eta,\delta}(\log N))$.*

- ***Unique-decoding Soundness.** The udPCP has $(\eta, \eta^2/6)$-unique decoding soundness.*

5.1 The Zig-Zag Product

In order to convert from a udPCP over an arbitrary constraint graph $G = (V, E)$ into a udPCP over the base layer $(X(1), X(2))$ of an HDX, we need a way to associate the vertices of G with the vertices $X(1)$. As in [BMVY25], to do so we rely on the zig-zag product of [RVW02], which we present next.

The zig-zag product takes a graph $G = (V, E)$, with minimum degree m_0, and a family of k-regular expander graphs $\mathcal{G} = \{\mathcal{G}_m\}_{m \geqslant m_0}$ where $\mathcal{G}_m$ has m vertices, and outputs a k^2-regular graph on $2|E|$ vertices. We denote the output of the zig-zag product as $\mathrm{ZZ}(G, \mathcal{G})$. Typically, the family of graphs $\mathcal{G}$ is chosen to have second singular value at most σ, for some chosen parameter σ. Before describing the zig-zag product, it is convenient to describe the replacement product of G' and $\mathcal{G}$. We denote this graph by $\mathrm{R}(G, \mathcal{G})$, defined in the following way.

- For each vertex v, arbitrarily order its neighbors from 1 to $\deg(v)$. We will refer to the cth neighbor of v as the vertex labeled c under this ordering.

- Replace each vertex $v \in V'$ with a copy of $\mathcal{G}_{\deg(v)}$, where $\deg(v)$ is the degree of v in G'. Hence v is replaced by a cloud of $\deg(v)$ vertices, and we also label these vertices by $\{v\} \times [\deg(v)]$.

- Let $(u, v) \in E'$ be an edge where v is the c_1th neighbor of u and u is the c_2th neighbor of v, then we add the edge $((u, c_1), (v, c_2))$ to $\mathrm{R}(G, \mathcal{G})$. In addition, we add the edge $((v, c_1), (v, c_2))$ to $\mathrm{R}(G, \mathcal{G})$ for all v and (c_1, c_2) such that $((v, c_1), (v, c_2))$ is an edge in $\mathcal{G}_{\deg(v)}$.

Note that $\mathrm{R}(G, \mathcal{G})$ has $2|E|$ vertices and is $(k + 1)$-regular. We define the zig-zag product, $\mathrm{ZZ}(G', \mathcal{G})$ as follows.

- The vertex set is $V(\mathrm{ZZ}(G, \mathcal{G})) = V(\mathrm{R}(G, \mathcal{G}))$.

- We add an edge $((u, c_1), (v, c_4))$ to $E(\mathrm{ZZ}(G, \mathcal{G}))$ if there exist c_2 and c_3 such that all of $((u, c_1), (u, c_2))$, $((u, c_2), (v, c_3))$, $((v, c_3), (v, c_4))$ are edges in $E(\mathrm{R}(G, \mathcal{G}))$. In words, the vertex (u, c_1) is adjacent to (v, c_4) if the latter vertex can be reached, in $\mathrm{R}(G, \mathcal{G})$, by first taking a step in the cloud of u, then taking a step between the clouds u and v, and finally taking a step in the cloud of v.

It is straightforward to verify that $\mathrm{ZZ}(G, \mathcal{G})$ has $2|E|$ vertices and is k^2 regular.

5.2 Routing Protocols

We now discuss routing protocols and associated useful notations, and we start by explaining what a T-round routing protocol on a graph $G = (V, E)$ is. It has three stages: input, communication, and output, described as follows.

Input. Each vertex $v \in V$ holds an initial message, which we think of as a symbol from some global alphabet Σ.

Communication. During each round $r \in [T]$, each vertex $v \in V$ performs some computation and sends a message, which is a symbol from Σ, to each of its neighbors. The message to each neighbor may be different. Moreover, the computation v makes depends on its initial message and any messages received from its neighbors in prior rounds. The history of messages v sent/received before step r is called the *transcript* of v in round r, so that v's outgoing messages during round r are functions of this transcript. For $r = 1$, v's message depends only on its initial message.

Output. After round T, each vertex calculates an output (as some function of its transcript in round T).

Definition 5.2. *Given an arbitrary graph $G = (V, E)$, a T-round routing protocol on G, which we denote by $\mathcal{R}$, consists of the following items.*

- ***Initial Message.*** *An initial message $A : V \to \Sigma$ where Σ is some alphabet.*

- ***Out Maps.*** *For each round $r \in [T]$, a map $\mathrm{OUT}_r : 2E \times \Sigma^* \to \Sigma$, where $\mathrm{OUT}_r(u, v, \sigma)$ denotes the message that u sends to v in round r given that u's transcript at round r is σ. Here $2E$ consists of all ordered pairs of adjacent vertices, meaning each edge $(u, v) \in E$ appears as (u, v) and (v, u) in $2E$, and the map is defined only for σ which represent a valid transcript of the protocol up to step r.*

- ***In Maps.*** *For each round $r \in [T]$, a map $\mathrm{IN}_r : 2E \to \Sigma$ where for each $r \in [T]$, $\mathrm{IN}_r(u, v)$ should be thought of as the message received by v from vertex u in round r of the protocol.*

- ***Output***. *A function* $A_T : V \times \Sigma^* \to \Sigma$, *where* $A_T(v, \sigma)$ *is the symbol output by* v *at the end of the routing protocol given that its transcript at the end of the routing protocol is* σ. *We refer to* A_T *as the function computed by the protocol.*

We will also have to consider a quantity known as the *work complexity* of a routing protocol.

Definition 5.3. *The work complexity of a* T*-round routing protocol over* $G = (V, E)$ *is the maximum over* $r \in [T]$ *and* $(u, v) \in 2E$, *of the circuit complexity of the function* $\mathrm{OUT}_r(u, v, \cdot) : \Sigma^* \to \Sigma$.

Although we are able to use the routing protocol of [BMVY25] in a black-box way, it will still be helpful to give a brief description of why routing protocols are useful for transforming the constraint graph of udPCPs. We refer the reader to [BMVY25] for a more detailed discussion.

In the setting of udPCPs, routing protocols are used as follows. Let $\mathcal{D}$ be an initial udPCP with bipartite, k-regular constraint graph $G = (A \cup B, E)$, and suppose that $G' = (V, E')$ is the target graph which we want to use as the constraint graph of a new udPCP, $\mathcal{D}'$. Suppose for this exposition that G is a matching, that $|V| = |A| = |B|$, and let $\pi : A \to B$ denote this matching. Identifying both A and B with V, we can view π as a permutation on V, i.e. as a mapping $\pi : V \to V$. Now the initial message $A : V \to \Sigma$ to the routing protocol should be thought of as the assignment of A and the goal of the communication of the routing protocol is to pass the value $A(v)$, which is initially held at vertex v, to the vertex $\pi(v)$. If this is done correctly, then each vertex $\pi(v)$ can check if the assignment A actually satisfied the constraint on $(v, \pi(v))$ in the original udPCP. Hence, the desired output function is $A_T(\pi(v), \sigma) = A(v)$, where here σ is the transcript of the honest routing protocol received by vertex $\pi(v)$. Assuming $\pi(v)$ correctly computes $A_T(\pi(v), \sigma) = A(v)$, it can check itself if $A(v)$ and its own initial value $A(\pi(v))$ should satisfy the constraint on $(v, \pi(v))$.

The above gives a high level overview of the connection between routing protocols and transforming constraint graphs of udPCPs, and roughly speaking it describes what happens in the perfect completeness case. In order to show soundness, one has to consider what happens to the routing protocol under adversarial edge corruptions.

Definition 5.4. *We say that an edge* (u, v) *is adversarially corrupted if, for every round, the outgoing message from* u *may be an adversarially selected symbol in* Σ.

In [BMVY25], routing protocols are considered in a slightly different setting. There, the target graph $G = (X(1), X(2))$ is the base layer of an arbitrary HDX, X, and one considers the zig-zag product $Z = \mathrm{ZZ}(G, \mathcal{G})$, where $\mathcal{G}$ is an appropriately chosen family of expanders. Instead of considering a function directly over the vertices of G, one considers a function over the zig-zag product, Z, which is then associated with G. Strictly speaking, the zig-zag product is not necessary to describe the [BMVY25] routing protocol, but it is helpful to have it in mind as it virtually appears in the definition of the initial messages and the function computed. We will similarly use Z and below we describe how routing protocols over Z translate to protocols over G.

Let us label the vertices of Z as $V(Z) = \{(u, v) \mid u \in X(1), (u, v) \in X(2)\}$ and let the initial messages over $V(Z)$ be given by $A'_0 : V(Z) \to \Sigma$. Fix a permutation $\pi : V(Z) \to V(Z)$. The goal of the routing protocol is to transmit $A'_0(v)$ to $A'_0(\pi(v))$ for all $v \in V(Z)$. The actual routing protocol should take place over G though, so we would like a way to simulate the initial messages and final function (which are over $V(Z)$), as functions over $X(1)$. To this end, define $\mathcal{H} = \{(j, u) \mid j \in V(Z), u \in X_{j_1}(1)\}$, and define the function $A_0 : \mathcal{H} \to \Sigma$, where $A_0(j, u) = A'_0(j)$. Now, we think of the initial message of a vertex $u \in X(1)$ as holding all of the symbols $(A_0(j, u))_{j \in V(Z) \text{ such that } u \in X_{j_1}(1)}$. One can think of the function A_0 as a function over the links of X: for each $j \in V(Z)$, each vertex $u \in X_{j_1}(1)$ contains the symbol of j (among other things). Then, the routing protocol over Z naturally translates over to a link-to-link transfer protocol over G.

Below we describe the formal guarantee of the routing protocol from [BMVY25, Lemma 4.6]. In the soundness case, we have the following lemma.

Lemma 5.5. *There exist* $\varepsilon_0, \alpha > 0$ *and* $d_0, C' \in \mathbb{N}$ *such that the following holds for sufficiently large* $n \in \mathbb{N}$. *Let* $\mathcal{G}$ *be a family of expanders, each with second singular value at most* $\frac{\alpha}{2}$, *and let* X *be any complex constructed as in Theorem 2.9 with dimension parameter, exponent, target number of vertices, and prime set to* $d \geqslant d_0$, C', n, *and* $q = \Theta\left(\log^{C'}(n)\right)$ *respectively. Let* $Z = \mathrm{ZZ}(G, \mathcal{G})$, *and let* $\pi : V(Z) \to V(Z)$ *be any permutation over* $V(Z)$.

Then there is a routing protocol on $G = (X(1), X(2))$ with round complexity $T = O(\log(n))$ and work complexity $q^{O(d^2)} \log |\Sigma|$ such that for any initial function $A_0 : \mathcal{G} \to \Sigma$ which satisfies

$$\Pr_{j \in V(Z)}[\mathsf{Maj}_{0.99}(A_0(j, u)) \mid u \sim X_{j_1}(1)) \neq \perp] \geqslant 1 - \xi,$$

and for all possible adversarial corruptions of at most ε-fraction of edges with $\varepsilon \leqslant \varepsilon_0$, the protocol computes a function $A_T : \mathcal{G} \to \Sigma$ which satisfies

$$\Pr_{j \in V(Z)}[\mathsf{Maj}_{0.99}(A_T(\pi(j), w) \mid w \sim X_{\pi(j)_1}(1)) = \mathsf{Maj}_{0.99}(A_T(j, u) \mid u \sim X_{\pi(j)_1}(1))] \geqslant 1 - \xi - \frac{1}{\log n}.$$

In words, Lemma 5.5 gives, for any permutation π over $V(Z)$, a routing protocol that is tolerant against up to ε errors. If an overwhelming majority value exists in at least $1 - \xi$ of the links in the input message, then the same overwhelming majority value exists in at least $1 - \xi - \frac{1}{\log n}$ of the links in the output message.

The completeness of the protocol from Lemma 5.5 is stated in the following lemma.

Lemma 5.6. *Let $G = (X(1), X(2))$ where $X = (X(1), \ldots, X(d))$ is a d-dimensional complex, let $\mathcal{G}$ be an appropriate family of expanders as above and let $A_0 : \mathcal{G} \to \Sigma'$ be a function such that $A_0(j, v)$ is the same for all $v \in X_{j_1}(1)$. Then for each permutation $\pi : V(Z) \to V(Z)$, an honest run of the protocol from Lemma 5.5 satisfies*

$$A_0(j, v) = A_T(\pi(j), u)$$

for every $(j, v), (\pi(j), u) \in \mathcal{G}$.

We will require the following fact from [BMVY25] which translates the uniform distributions over vertices and edges of Z to the stationary distributions over vertices and edges of X.

Fact 5.7. *Choosing $j \sim V(Z)$ uniformly at random and outputting j_1, we have that $j_1 \in X(1)$ is distributed according to the underlying distribution of $X(1)$. Additionally, choosing an edge (j, j') of Z uniformly at random and outputting $(j_1, j_1') \in X(2)$, we have that (j_1, j_1') is distributed according to the underlying distribution of $X(2)$.*

5.3 Proof of Theorem 5.1

In this section we prove Theorem 5.1.

5.3.1 The Construction

Fix $\eta > 0$, $\delta \in (0, 1)$, a size N circuit $\varphi : \Sigma_0^n \to \{0, 1\}$ and set $\mathcal{L} = \mathsf{SAT}(\varphi) \subseteq \Sigma_0^n$. Let

$$\mathcal{D}' = \left(A' \cup B', E', \Sigma', \{\Phi_e\}_{e \in E'}, \{\mathcal{P}_t'\}_{t \in [n]}, \{D_t'\}_{t \in [n]} \right)$$

to be the udPCP for $\mathcal{L}$ guaranteed by Lemma 4.5 with $\left(\frac{\eta}{100}, \rho' \cdot \left(\frac{\eta}{100} \right)^5 \right)$-unique decoding soundness. Note that by Lemma 4.5, we may take the length of $\mathcal{D}'$ to be $C \cdot s(N)$ for any $C \in \mathbb{N}$, where $s(N) = O_{|\Sigma_0|, \eta}(N \cdot \log^{c_0}(N))$ for some constant c_0. Here we are slightly abusing notation as technically, there is a separate udPCP for each C, but let us just refer to the udPCP as $\mathcal{D}'$ for now and fix C later. The udPCP $\mathcal{D}'$ is bipartite and d'-regular. We denote its constraint graph by $G' = (A' \cup B', E')$, write $V' = A' \cup B'$, and set $N' = |V'|$.

Take $\varepsilon_0, \alpha, d_0, C'$ to be the constants from Lemma 5.5 and let $\mathcal{G}$ be a family of $O(\alpha^{-2})$-regular expander graphs such that each graph has second singular value at most $\frac{\alpha}{2}$. In particular, we use Lemma 2.4 where $d = O(\alpha^{-2})$, and obtain a family $\mathcal{G}$ with one graph of each size m for all $m \geqslant \Omega(\alpha^{-2})$. Note that by Lemma 2.4, each graph in the family $\mathcal{G}$ can be constructed in time polynomial in the number of vertices. We take the complex X from Theorem 2.9 so that the following items hold:

- the direct-product soundness parameter is set to the δ fixed above,

- the dimension parameter $d \geqslant d_0$,

- the number of vertices satisfies $N \cdot \log^{c_0+10}(N) \leqslant |X(1)| \leqslant N \cdot \log^{10c_0+10}(N)$, and is sufficiently large in accordance with Lemma 5.5,

- the prime q is set to $q = \log^{c'}(N)$ for some constant c',

- $N \cdot \log^{c_0+10}(N) \leqslant |X(2)| \leqslant N \cdot \log^{c_1}(N)$, where $c_1 \geqslant c_0 + 10$ is some constant. This item is implied by the fact that d is constant along with the prior two items and item 3 of Theorem 2.9.

Let us briefly argue why such a complex X can be constructed from Theorem 2.9. There, one can fix d arbitrarily and then choose a desired number of vertices, say n', arbitrarily. In our case, we set n' to be in the range $[N \cdot \log^{c_0+10}(N), N \cdot \log^{10c_0+10}(N)]$, and note that we can assume n' is sufficiently large in accordance with Lemma 5.5 by assuming that the original circuit is sufficiently large. After n' is fixed, we set $q = \Theta(\log^{C'}(n'))$ and we run the algorithm of Theorem 2.9 with direct-product soundness, exponent, dimension, target number of vertices, and prime parameters set to δ, C', d, n', q and direct-product dimension set arbitrarily. Theorem 2.9 guarantees that this X is constructed in $\text{poly}(n') = \text{poly}(N)$ time.

Viewing $(X(1), X(2))$ **as a multigraph.** We will want to consider $(X(1), X(2))$ as a weighted graph according to its stationary distribution and this is accomplished by having each edge $(u, v) \in X(2)$ have multiplicity proportional to its weight under the stationary distribution. Specifically, recall that the stationary distribution over (u, v) samples a d-face $U \sim X(d)$ and then $\{u, v\} \in \binom{U}{2}$ uniformly, so the weight of (u, v) under this distribution is, $\frac{m(u,v)}{\binom{d}{2}}$, where $m(u, v)$ is the number of d-faces containing u and v, and is hence $\text{poly}(\log(N))$. For each $(u, v) \in X(2)$, we think of it as having multiplicity $m(u, v)$ in the multigraph $(X(1), X(2))$.

Identifying V' **with** $V(Z)$ **by adding a negligible number of vertices.** The first step will be to modify G' so that it has size equal to $|V(Z)|$. Recall $|V(Z)| = 2|X(2)|$, so

$$N \cdot \log^{c_0+10}(N) \leqslant |X(2)| \leqslant N \cdot \log^{c_1}(N), \tag{12}$$

for some $c_1 \geqslant c_0$. We can thus write $|V(Z)| = s(N) \cdot Q + R$ for integers Q and R satisfying $\log^{10}(N) \leqslant Q \leqslant \log^{c_1-c_0}(N)$ and $0 \leqslant R < s(N)$. We set the repetition parameter $C = Q$, so that we have $|V'| = Q \cdot s(N)$ and add R vertices to the constraint graph of $\mathcal{D}'$. On these R additional vertices we add an arbitrary bipartite d'-regular graph with equality constraints. Note that these $d'R$ extra equality constraints are at most $\frac{d'R}{d' \cdot s(N) \cdot Q} \leqslant \frac{1}{\log^{10}(N)}$ fraction of all constraints, so they can essentially be ignored. The decoders for $\mathcal{D}'$ are unchanged after adding R additional vertices (we can define the decoders arbitrarily when one of these R additional vertices is given as input). We define the new decoding distributions as follows. Let $\xi := R/|V'|$-so that the R-additional vertices account for ξ-fraction of the vertices overall in G' and divide the R additional vertices into equal parts of size n and label them 1 through n. For each $t \in [n]$, the decoding distribution $\mathcal{P}'_t$ is modified as follows

- With probability $1 - \xi$ choose u according to the old decoding distribution for t without the R extra vertices.

- With probability ξ choose one of the extra R vertices in the part labeled by t.

One can check that choosing $t \in [n]$ and then $u \sim \mathcal{P}'_t$ as above, we still get that u is uniform in V' (with the R additional vertices). Since $\xi \leqslant \frac{1}{\log^{10}(N)}$, it is easy to see that after adding the R-additional vertices, the udPCP still has $\left(\frac{\eta}{100}, \frac{\rho'}{2}\left(\frac{\eta}{100}\right)^5\right)$-unique decoding soundness. It is also straightforward to check that the resulting udPCP has length $|V(Z)|$ and perfect completeness. Furthermore, alphabet size, d'-regularity, and left-decoding degree are maintained. Henceforth, we slightly abuse notation and still refer to the modified constraint graph as $G' = (A' \cup B', E')$ and assume that it has the same number of vertices as Z for the remainder of the section.

Breaking G' **into matchings and setting the routing protocols.** Since G' is a d'-regular bipartite graph we can decompose E' into d' perfect matchings $\pi_1, \ldots, \pi_{d'}$. Since $|V(Z)| = |V'|$ we can associate V' with $V(Z)$ in an arbitrary manner, and we fix such an identification henceforth. Abusing notation, we view each π_i as a function from $V(Z) \to V(Z)$ satisfying that π_i^2 is the identity. In particular, for $j \in V(Z)$ corresponding to some $a \in A'$, $\pi_i(a)$ is the vertex in $V(Z)$ corresponding to the vertex in B' matched with a under π_i, and vice-versa if $j \in V(Z)$ corresponds to some $b \in B'$.

In order to describe the construction, it will be helpful to define $\mathcal{H} = \{(j, u) \mid j \in V(Z), u \in X_{j_1}(1)\}$ and consider an initial function $A_0 : \mathcal{H} \to \Sigma'$ where recall that Σ' is the alphabet of $\mathcal{D}'$. One should think of A_0 as coming from a satisfying assignment of $\mathcal{D}'$, so that $A_0(j, u)$ is the Σ' symbol assigned to the vertex in V' corresponding to j under this satisfying assignment. Moreover, one should think of every $A_0(j, u)$ as being "held" at the vertex $u \in X(1)$, so that $u \in X(1)$ holds many Σ' symbols (and in particular, one for each link it is in). That is, an assignment to the new udPCP is supposed to provide such values for each $u \in X(1)$.

For each $i \in [d']$ let $\mathcal{R}_i$ be the routing protocol from Lemma 5.5 for the permutation π_i. The protocols $\mathcal{R}_i$ all have round complexity T for some $T = O(\log N)$. Here, one should think of the routing protocols as being run with the initial maps A_0 above. Then in words, for every $j \in V(Z)$, the routing protocol $\mathcal{R}_i$ is meant to route an assignment of the V' vertex corresponding to j from every vertex in the link X_{j_1} to every vertex in the link $X_{\pi_i(j_1)}(1)$. For each $i \in [d'], r \in [T]$, let $\mathrm{IN}_{i,r}$ and $\mathrm{OUT}_{i,r}$ denote the in map and out map respectively during round r of protocol $\mathcal{R}_i$. Also, let $\mathcal{A}_{i,T}$ denote the output function of protocol $\mathcal{R}_i$.

The description of $\mathcal{D}$. We denote the new dPCP that we construct as

$$\mathcal{D} = \big(X(1), X(2), \Sigma, \{\Phi_e\}_{e \in X(2)}, \{D_t\}_{t \in [n]}, \{\mathcal{P}_t\}_{t \in [n]}\big),$$

and define its parts as follows.

- **Constraint Graph.** The constraint graph is the base layer of X, i.e. $(X(1), X(2))$ (with the multiplicities proportional to the weights of the stationary distribution of $X(2)$ over $X(1)$).

- **Alphabets:** For each $u \in X(1)$, an alphabet symbol for u consists of the following.

    - **Initial Messages:** one Σ' symbol for each $j \in V(Z)$ such that $u \in X_{j_1}(1)$. This symbol is intended to be the value $A_0(j, u)$ for each $j \in V(Z)$ which is supposed to come from a satisfying assignment to $\mathcal{D}'$ as described above.

    - **Routing Transcript:** the maps $\mathrm{IN}_{i,r} : 2E(G) \to \Sigma$ for each $i \in [d']$ and $r \in [T]$. For each $i \in [d']$ and $r \in [T]$, this is the in map of round r for the protocol $\mathcal{R}_i$. Recall that $\mathrm{IN}_{i,r}(u, v)$ should be thought of as the message received by v from u during round r in the routing protocol for the matching π_i.

In addition, we constrain the alphabet of u in the following ways.

    - **Hardcoding $\mathcal{D}'$ Constraints:** for each $i \in [d']$ and $u \in V'$, we only allow symbols σ for u such that the values in σ corresponding to $A_0(j, u)$ and $A_{i,T}(\pi_i(j), u)$ satisfy the constraint $\Phi'_{(j, \pi_i(j))}$ in $\mathcal{D}'$, for every j such that $u \in X_{j_1}(1)$.

    - **Hardcoding Decoding Consistency:** for all $t \in [n]$ and $u \in V'$, we only allow symbols σ for u which satisfy the following. Denoting by σ_j the symbol corresponding to $A_0(j, u)$ in σ, we require that $\mathcal{D}'_t(j, \sigma_j)$ is the same for all j such that $u \in X_{j_1}(1)$ and $j \in \mathrm{supp}(\mathcal{P}'_t)$.

We denote the constrained alphabet for u as Σ_u.

- **Constraints:** For each $(u, v) \in X(2)$, the constraint between u and v checks the following.

    - For each $j \in V(Z)$ such that $u, v \in X_{j_1}(1)$, check that

    $$A_0(j, u) = A_0(j, v) \quad \text{and} \quad A_{i,T}(j, u) = A_{i,T}(j, v), \ \forall i \in [d'].$$

    - For each $i \in [d']$ and $r \in [T-1]$, check that

    $$\mathrm{IN}_{i,r+1}(u, v) = \mathrm{OUT}_{i,r}(v, u) \quad \text{and} \quad \mathrm{IN}_{i,r+1}(v, u) = \mathrm{OUT}_{i,r}(v, u).$$

In words, the first item checks that the initial message parts of the symbols assigned to u and v are consistent. The second item checks that the routing transcripts of u and v are consistent, that is, for each protocol $\mathcal{R}_i$ and every round $r \in [T-1]$, the constraint checks that the message that v receives from u (according to v's alphabet symbol), is equal to the message that u sends to v (calculated based on the prior round in maps held at u's alphabet symbol), and vice-versa.

- **Decoding Distributions:** For each $t \in [n]$, the distribution $\mathcal{P}_t$ is as follows

  - Choose $j \sim \mathcal{P}'_t$ and think of $j \in V(Z)$ as a vertex in identifying V' (by the identification we have between V' and $V(Z)$).
  - Output $u \sim X_{j_1}(1)$.

- **Decoders:** For each $t \in [n]$, $u \in \mathrm{supp}(\mathcal{P}_t)$ and label $\sigma \in \Sigma_u$ for u, the decoder outputs

$$\mathcal{D}_t(u, \sigma) = \mathcal{D}'_t(j, \sigma_j),$$

where j is an arbitrary vertex in $V(Z)$ such that $u \in X_{j_1}(1)$, and σ_j refers to the part of σ which contains a label for $A_0(j, u)$. Note that the specific identity of j above does not matter due to the way the alphabet of Σ_u is constrained (specifically, the second item) and hence the decoder is well defined.

5.3.2 Proof of Everything Except Unique-Decoding Soundness

We start by proving all of the properties except for unique decoding soundness, which is deferred to the next section.

Constraint Graph and Length. It is clear that the constraint graph is $(X(1), X(2))$ and hence the length is $|X(1)| = O_{|\Sigma_0|, \eta}(N \cdot \mathsf{poly}(\log(N)))$ by construction of the complex X.

Degrees. Let $\Gamma', \Gamma'_{\mathsf{dec}}$ denote neighborhoods and decoding neighborhoods in $\mathcal{D}'$ and let $\Gamma, \Gamma_{\mathsf{dec}}$ be defined similarly for $\mathcal{D}$. The degree of each vertex in the constraint graph is $\mathsf{poly}_\delta(\log n)$ by the properties of the HDX. In particular, it follows from the fact that each vertex is contained in $q^{O(d^2)}$ many d-faces by item 3 of Theorem 2.9. As for the decoding degree, we have that $t \in \Gamma_{\mathsf{dec}}(v)$ if and only if there exists $j \in V(Z)$ such that $v \in X_{j_1}(1)$ and $t \in \Gamma'_{\mathsf{dec}}(j)$. Since $|\Gamma'_{\mathsf{dec}}(j)| = O(1)$ for all $j \in V(Z)$, and the number of $j \in V(Z)$ such that $v \in X_{j_1}(1)$ is $\mathsf{poly}_\delta(\log n)$, the size of v's decoding neighborhood is $\mathsf{poly}_\delta(\log n)$.

Alphabet Size. An alphabet for each vertex consists of the initial messages which the vertex holds, and the messages it received throughout all of the rounds of the routing protocol. For the initial messages, each vertex v contains one symbol from a $O_{|\Sigma_0|}(1)$-sized alphabet for each $j \in V(Z)$ such that $v \in X_{j_1}(1)$. The number of such j is $\mathsf{poly}_\delta(\log(N))$ since v has $\mathsf{poly}_\delta(\log(N))$ neighbors in $(X(1), X(2))$ and for each u, there are a constant number of $j \in V(Z)$ such that $j_1 = u$. Thus, the initial message part of the alphabet consists of $\mathsf{poly}_\delta(\log(N))$ symbols from a $O_{|\Sigma_0|}(1)$-sized alphabet. For the routing transcript part, the alphabet of a vertex v consists of one message per round of the protocol. By the work complexity of the protocol, the message contains at most $\mathsf{poly}_\delta(\log(N))$ bits, and there are $T = O(\log N)$ rounds. Altogether, this shows that each alphabet symbol for v can be written using $\mathsf{poly}_\delta(\log(N))$ symbols from a $O_{|\Sigma_0|}(1)$-sized alphabet and hence the alphabet size is $2^{O_{|\Sigma_0|}(\mathsf{poly}_\delta(\log(N)))}$ as desired.

Decision Complexity. Fix an edge $(u, v) \in X(2)$. The decision complexity of (u, v) is the sum of the following 2 parts: first, circuit sizes for checking that the alphabet symbols for u and v are valid symbols (i.e. that they satisfy the hardcoded constraints), and second, checking that the constraint is satisfied. For the first part, let us describe the circuit complexity for checking validity of u's symbol, as the circuit complexity for checking v's symbol is the same. The circuit needs to check that for each of $O_\eta(1)$ matchings π, the parts of the alphabet corresponding to $A_0(j, \pi(u))$ and $A_{i,T}(j, u)$ satisfy some constraint in $\mathcal{D}'$ for every j such that $u \in X_{j_1}(1)$. Since the constraint in $\mathcal{D}'$ is over $O_{|\Sigma_0|}(1)$ sized alphabet symbols, its circuit complexity is $O_{|\Sigma_0|}(1)$. Moreover, there are at most $\mathsf{poly}(\log(N))$ many such j's. Altogether, this results in circuit complexity $O_{|\Sigma_0|}(\mathsf{poly}_\eta(\log(N)))$. For the second part, the circuit needs to calculate the outgoing messages of u and v in each of the routing protocols $\mathcal{R}_1, \ldots, \mathcal{R}_{d'}$, and then check an equality on the computed message. As the work complexity is $O_{|\Sigma_0|}(\mathsf{poly}(\log(N)))$ and there are $O_\eta(1)$ routing protocols, this results in $O_{|\Sigma_0|}(\mathsf{poly}_\eta(\log(N)))$ circuit complexity as well. Overall, the decision complexity is $O_{|\Sigma_0|}(\mathsf{poly}_\eta(\log(N)))$.

Decoding Distribution. Choosing $t \in [n]$ uniformly and $j \sim \mathcal{P}'_t$ uniformly, we have that $j \in V(Z)$ is distributed uniformly at random by the guarantee of starting udPCP $\mathcal{D}'$ (specifically the fourth item of Lemma 4.5). Then the decoding distribution of $\mathcal{D}$ chooses $u \sim X_{j_1}(1)$, and we have $u \sim X(1)$ by Fact 5.7.

Decoding Complexity. It is clear from the description of the decoder, that given a vertex u and a symbol σ for it, the decoder can restrict σ to a symbol from the alphabet of $\mathcal{D}'$ and run the decoder of $\mathcal{D}'$ (which is another restriction of the symbol to a symbol from Σ_0). The circuit complexity for both of these tasks is on the order of logarithmic in the left alphabet size (which is also the length of the circuit's input), and hence it is $O_{|\Sigma_0|}(\mathsf{poly}_{\eta,\delta}(\log(N)))$.

Completeness. Fix $w \in \mathcal{L}$ and let $A' : V' \to \Sigma'$ be the assignment to $\mathcal{D}'$ which satisfies all of the constraints and always decodes according to w. Now, define the initial messages according to A' and run the routing protocols $\mathcal{R}_1, \ldots, \mathcal{R}_{d'}$ with these initial messages. Let A be the resulting assignment for $\mathcal{D}$. We will show that A is a valid assignment satisfying all of the constraints to $\mathcal{D}$ and that it always decodes according to w.

For the first part, it is clear that all of the constraints of $\mathcal{D}$ are satisfied, and it remains to check that A is a valid assignment, i.e. it satisfies the harded constraints of the alphabets. Indeed, for any v, any of the matchings π_i, and any j such that $v \in X_{j_1}(1) \cap X_{\pi_i(j)_1}(1)$, the alphabet symbols corresponding to $A_0(j, v), A_{i,T}(\pi(j), v)$ are precisely $A'(j)$ and $A'(\pi(j))$ by Lemma 5.6, so they satisfy the constraint in $\mathcal{D}'$. For the decoding hardcoded constraints fix any vertex $v \in X(1)$ and $j \in V(Z), t \in [n]$ such that $v \in X_{j_1}(1)$ and t is in the decoding neighborhood of j in $\mathcal{D}'$. Then by the completeness of $\mathcal{D}'$, we have that $D'_t(j, A'(j)) = w_t$. Since the symbol corresponding to the initial value of j under assignment $A(v)$ is precisely $A'(j)$, this shows that the folding for decoding consistency is satisfied.

5.3.3 Proof of Unique-Decoding Soundness

We will show that $\mathcal{D}$ satisfies $(\eta, \eta^2/6)$-unique decoding soundness for a fixed η sufficiently small relative to the constant ε_0 from Lemma 5.5. Suppose there is an assignment to $\mathcal{D}$ that satisfies at least $(1-\varepsilon)$-weight of the constraints, for $\varepsilon := \eta^2/6$. Let $A : X(1) \to \Sigma$ be the assignment to $\mathcal{D}$ achieving this value. From A, we can derive the function $A_0 : \mathcal{H} \to \Sigma'$. Specifically, for each $(j, v) \in \mathcal{H}$, we set $A_0(j, v)$ to be the part of the symbol $A(v)$ corresponding to an assignment for j from the alphabet of $\mathcal{D}'$, Σ'. Now, define the following assignment, $B : V' \to \Sigma'$:

$$B(j) = \mathsf{plurality}_{u \sim X_{j_1}(1)}\{A_0(j, u)\},$$

where the function plurality refers to the most common value $A_0(j, u)$ when choosing $u \sim X_{j_1}(1)$. It is shown in [BMVY25] that this assignment B is a nearly satisfying assignment for $\mathcal{D}'$.

Lemma 5.8. *[BMVY25, Lemma 5.3] Suppose that A satisfies $(1 - \varepsilon)$-weight of the constraints in $\mathcal{D}$. Then B defined as above satisfies at least $\left(1 - O\left(\frac{1}{\log N}\right)\right) \geqslant \left(1 - \frac{\rho'}{2}\left(\frac{\eta}{100}\right)^5\right)$-fraction of the constraints of $\mathcal{D}'$.*

Proof. The lemma is implicitly shown in the proof of the soundness claim in Lemma 5.3 of [BMVY25]. Specifically, they argue that if A is a $(1 - \varepsilon)$-satisfying assignment to $\mathcal{D}$, then for each of the routing protocols $\mathcal{R}_i$ the routing transcript of $\mathcal{R}_i$ corresponding to the assignments of A corresponds to a routing on $(X(1), X(2))$ with at most ε-fraction of adversarial edge corruptions.

Additionally, since A is $(1 - \varepsilon)$-satisfying, one can show that for $\left(1 - \frac{1}{\log N}\right)$-fraction of $j \in V(Z)$, we have that the probability a random vertex in the link $X_{j_1}(1)$ holds a value for j that is consistent with B, i.e. the probability $\mathrm{Pr}_{u \in X_{j_1}(1)}[A_0(j, u) = B(j)]$, is very close to 1 (in fact this statement is required for our proof of unique-decoding soundness too, so the argument is reproduced, essentially in Lemma 5.9, below). As a result, the initial messages A_0 satisfy the condition in Lemma 5.5 and applying Lemma 5.5, we get that for $\left(1 - \frac{1}{\log N}\right)$-fraction of $j \in V(Z)$, the values $B(j)$ and $B(\pi_i(j))$ are consistent with those held at some vertex $u \in X_{j_1}(1)$. By the hardcoding of the alphabet of u, this shows that B satisfies the constraint on $(j, \pi_i(j))$, and applying this over all routing protocols shows that B satisfies almost all of the constraints in the original udPCP. $\square$

Because $\mathcal{D}'$ satisfies $\left(\frac{\eta}{100}, \frac{\rho'}{2} \cdot \left(\frac{\eta}{100}\right)^5\right)$-unique decoding soundness with respect to $\mathcal{L}$ for some constant ρ, there exists $w \in \mathcal{L}$ such that

$$\Pr_{t \in [n], j \sim \mathcal{P}'_t}[D'_t(j, B(j)) \neq w_t] \leqslant \frac{\eta}{100}. \tag{13}$$

33

We can bound the decoding error of the assignment A for the udPCP $\mathcal{D}$ as follows

$$
\begin{aligned}
\Pr_{t\in[n],u\sim\mathcal{P}_t}[D_t(u,A(u))\neq w_t] &= \Pr_{t\in[n],j\sim\mathcal{P}'_t,u\sim X_{j_1}(1)}[D'_t(j,A_0(j,u))\neq w_t] \\
&\leqslant \Pr_{t\in[n],j\sim\mathcal{P}'_t}[D'_t(j,B(j))\neq w_t] \\
&\quad + \Pr_{t\in[n],j\sim\mathcal{P}'_t,u\sim X_{j_1}(1)}[A_0(j,u)\neq B(j)] \\
&\leqslant \frac{\eta}{100} + \Pr_{t\in[n],j\sim\mathcal{P}'_t,u\sim X_{j_1}(1)}[A_0(j,u)\neq B(j)].
\end{aligned}
\tag{14}
$$

In the first transition we used the fact that $D_t(u,A(u)) = D'_t(j,A_0(j,u))$, in the second transition we used the union bound, and in the third transition we used (13). Note that for any fixed j, we have

$$
\Pr_{u\sim X_{j_1}(1)}[A_0(j,u)\neq B(j)] \leqslant \Pr_{u,v\sim X_{j_1}(1)}[A_0(j,u)\neq A_0(j,v)].
$$

This is because $B(j)$ is defined to be the most popular value of $A_0(j,u)$ when sampling $u \sim X_{j_1}(1)$. Plugging this into (14) gives

$$
\Pr_{t\in[n],u\in\mathcal{P}_t}[D_t(u,A(u))\neq w(t)] \leqslant \frac{\eta}{100} + \mathbb{E}_{t\in[n],j\sim\mathcal{P}'_t}\left[\Pr_{u,v\in X_{j_1}(1)}[A_0(j,u)\neq A_0(j,v)]\right].
\tag{15}
$$

We say $j\in V(Z)$ is bad if

$$
\Pr_{u,v\in X_{j_1}(1)}[A_0(j,u)\neq A_0(j,v)] \geqslant 2\sqrt{\varepsilon}.
\tag{16}
$$

The expectation from (15) can be bounded by

$$
\mathbb{E}_{t\in[n],j\sim\mathcal{P}'_t}\left[\Pr_{u,v\in X_{j_1}(1)}[A_0(j,u)\neq A_0(j,v)]\right] \leqslant 2\sqrt{\varepsilon} + \Pr_{t\in[n],j\sim\mathcal{P}'_t}[j\text{ is bad}].
\tag{17}
$$

To bound the probability that j is bad, we will show that if j is bad, then the link $X_{j_1}(1)$ must contain many violated constraints. To this end, let $\mathcal{E}\subseteq X(2)$ denote the set of constraints that are not satisfied by the assignment A. By assumption, A satisfies at least $(1-\varepsilon)$-fraction of constraints, so $\mu_2(\mathcal{E})\leqslant\varepsilon$. On the other hand, if $A_0(j,u)\neq A_0(j,v)$ for any $j\in V(Z)$, and $(u,v)\in X_{j_1}(2)$, then the constraint on (u,v) is violated and $(u,v)\in\mathcal{E}$. We can thus show the following.

Lemma 5.9. *If j is bad then $\mu_{j_1,2}(\mathcal{E})\geqslant\sqrt{\varepsilon}$, and consequently $\Pr_{t\in[n],j\sim\mathcal{P}'_t}[j\text{ is bad}]\leqslant\frac{\mathsf{poly}(d)}{q}$.*

Proof. Fix j that is bad and for each symbol for j in the original udPCP $\mathcal{D}'$, $\sigma\in\Sigma'_j$, let

$$
U_\sigma = \{u\in X_{j_1}(1)\mid A_0(j,u)=\sigma\}.
$$

Then

$$
\mu_{j_1,2}(\mathcal{E}) \geqslant \Pr_{(u,v)\sim X_{j_1}(2)}[A_0(j,u)\neq A_0(j,v)] = 1 - \sum_{\sigma\in\Sigma'_j}\Pr_{(u,v)\sim X_{j_1}(2)}[u\in U_\sigma \wedge v\in U_\sigma].
\tag{18}
$$

Now, we use the fact that the graph $(X_{j_1}(1),X_{j_1}(2))$ has second eigenvalue at most $O\left(\frac{1}{\sqrt{q}}\right)$ by item 4 of Theorem 2.9, so we can bound

$$
\begin{aligned}
\sum_{\sigma\in\Sigma'_j}\Pr_{(u,v)\sim X_{j_1}(2)}[u\in U_\sigma \wedge v\in U_\sigma] &\leqslant \sum_{\sigma\in\Sigma'_j}\mu_{j_1}(U_\sigma)^2 + O(q^{-1/2})\mu_{j_1}(U_\sigma) \\
&\leqslant O(q^{-1/2}) + \sum_{\sigma\in\Sigma'_j}\mu_{j_1}(U_\sigma)^2.
\end{aligned}
\tag{19}
$$

Combining (18), (19), and the observation that $1 - \sum_{\sigma \in \Sigma'_j} \mu_{j_1}(U_\sigma)^2 = \Pr_{u,v \sim X_{j_1}(1)}[A_0(j,u) \neq A_0(j,v)]$ we get that

$$\mu_{j_1,2}(\mathcal{E}) \geqslant 1 - \sum_{\sigma \in \Sigma'_j} \mu_{j_1}(U_\sigma)^2 - O(q^{-1/2}) = \Pr_{u,v \sim X_{j_1}(1)}[A_0(j,u) \neq A_0(j,v)] - O\left(\frac{1}{\sqrt{q}}\right) \geqslant 2\sqrt{\varepsilon} - O(q^{-1/2}) \geqslant \sqrt{\varepsilon},$$

where the penultimate transition uses the assumption that j is bad from (16). This shows the first part of the lemma.

To obtain the "consequently" part of the lemma, recall that by Fact 5.7, when choosing $j \in V' = A' \cup B'$ uniformly at random and then outputting j_1, the resulting $j_1 \in X(1)$ is distributed according to the stationary distribution over $X(1)$. On the other hand, choosing $t \in [n]$ and $j \sim \mathcal{P}'_t$, we have j uniformly distributed in V'. Thus,

$$\Pr_{t \in [n], j \sim \mathcal{P}'_t}[j \text{ is bad}] = \Pr_{t \in [n], j \sim \mathcal{P}'_t}[\mu_{j_1}(\mathcal{E}) \geqslant \sqrt{\varepsilon}] = \Pr_{j \in V'}[\mu_{j_1}(\mathcal{E}) \geqslant \sqrt{\varepsilon}] \leqslant \frac{\mathsf{poly}(d)}{q}.$$

In the second transition we used the above observation on the distribution of j_1. In the third transition we used Lemma 2.10, $\mu_2(\mathcal{E}) \leqslant \varepsilon$, and the fact that a uniformly random $j \in V'$ is on the side A' with probability $1/2$. $\square$

We can now conclude the proof of unique decoding soundness.

Proof of Unique Decoding Soundness. Fix $\eta > 0$ sufficiently small, and suppose there is an assignment, A, to $\mathcal{D}$ that satisfies at least $(1 - \varepsilon)$-fraction of the constraints, for $\varepsilon = \eta^2/6$. Then, combining (15), (17), and Lemma 5.9, we get that there is $w \in \mathcal{L}$ such that the decoding error of A with respect to w is at most

$$\Pr_{t \in [n], u \sim \mathcal{P}_t}[D_t(u, A(u)) \neq w_t] \leqslant \frac{\eta}{100} + 2\sqrt{\varepsilon} + \frac{\mathsf{poly}(d)}{q} \leqslant \eta.$$ $\square$

6 Gap Amplification

Starting from the udPCP construction from Theorem 5.1, we now obtain a dPCP with small list-decoding soundness error by using a direct product based gap amplification.

Theorem 6.1. *For all $\varepsilon > 0$ and a circuit $\varphi : \Sigma_0^n \to \{0,1\}$ of size N, the language $\mathsf{SAT}(\varphi)$ has a dPCP satisfying the following:*

- **Length.** $O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon(\log N))$.

- **Left Alphabet.** *For each left vertex a, the left alphabet of a is contained in Σ^k for some $\Sigma \supseteq \Sigma_0$ of size $|\Sigma_2| = 2^{O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))}$ and $k = O_\varepsilon(1)$. Hence the left alphabet size is $2^{O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))}$.*

- **Right Alphabet.** *The right alphabet is $\Sigma^{\sqrt{k}}$ where Σ and k are the same as in the left alphabet condition.*

- **Degrees.** *The left degree is $O_\varepsilon(1)$, the decoding degree is $\mathsf{poly}_\varepsilon(\log N)$, and the right degree is $\mathsf{poly}_\varepsilon(\log N)$.*

- **Decision Complexity.** $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$.

- **Decoding Complexity.** $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$.

- **Decoding Distribution.** *The complete decoding distribution is agnostic. Furthermore, denoting the complete decoding distribution by $\mathcal{Q}$, for every $\boldsymbol{b} \in X(\sqrt{k})$ we have that*

$$\mathcal{Q}(\circ, \circ, \boldsymbol{b}) \leqslant \frac{\mathsf{poly}_\varepsilon(\log N)}{N}.$$

- **Completeness.** *The dPCP has perfect completeness.*

- **Soundness.** *The dPCP has (L, ε)-list-decoding soundness for $L := \varepsilon^{-O(1)}$.*

We now dscribe the construction for Theorem 6.1. Fix $\varepsilon > 0$ and a size N circuit $\varphi : \Sigma_0^n \to \{0, 1\}$ and set $\mathcal{L} = \mathsf{SAT}(\varphi)$. Set $\delta := \delta(\varepsilon) > 0$ sufficiently small relative to ε and take $\mathcal{D}_{\mathsf{hdx}}$ to be the udPCP for $\mathsf{SAT}(\varphi)$ obtained through Theorem 5.1 with the direct product soundness parameter there set to be $\delta/100$ and $\eta > 0$ there to be a small absolute constant so that the result holds. After setting δ, the other parameters are all chosen sufficiently large as described in Theorem 5.1, and in particular, we have that the parameters of X are as follows. The direct product parameter $k = O_\delta(1)$ is sufficiently large, the dimension parameter is $d = O_\delta(1)$ is sufficiently large, the number of vertices is $O_{|\Sigma_0|, \delta}(N \cdot \mathsf{poly}(\log(N)))$, the prime is $q = \mathsf{poly}_\delta(\log N)$. Let us denote the parts of $\mathcal{D}_{\mathsf{hdx}}$ by

$$\mathcal{D}_{\mathsf{hdx}} = \left(X(1), X(2), \Sigma, \{\Phi_e^{\mathsf{hdx}}\}_{e \in X(2)}, \{\mathcal{P}_t^{\mathsf{hdx}}\}_{t \in [n]}, \{D_t^{\mathsf{hdx}}\}_{t \in [n]} \right).$$

The amplified dPCP takes the form

$$\mathcal{D}_{\mathsf{dp}} = \left(X(k) \cup X(\sqrt{k}), E, \Sigma^k, \Sigma^{\sqrt{k}}, \{\Phi_e^{\mathsf{dp}}\}_{e \in E}, \{\mathcal{P}_t^{\mathsf{dp}}\}_{t \in [n]}, \{D_t^{\mathsf{dp}}\}_{t \in [n]} \right).$$

Throughout the proof we will use boldface letters to denote vertices of $\mathcal{D}_{\mathsf{dp}}$. The components of $\mathcal{D}_{\mathsf{dp}}$ are as follows:

- **Constraint Graph.** The bipartite inclusion graph with parts $X(k)$ and $X(\sqrt{k})$. Throughout the section we will use $\mathbf{a}$ to denote a generic vertex in $X(k)$ and $\mathbf{b}$ to denote a generic vertex in $X(\sqrt{k})$. We remark that $\mathbf{a}$ and $\mathbf{b}$ should also be thought of, respectively, as size k and $\sqrt{k}$ subsets of $X(1)$. For $\mathbf{a} \in X(k), \mathbf{b} \in X(\sqrt{k})$ we include an edge between them in E if $\mathbf{a} \supseteq \mathbf{b}$.

- **Left Alphabet:** For each $\mathbf{a} \in X(k)$, its alphabet is a subset Σ^k and each symbol σ should be thought of as a map from assigning a Σ symbol to each $v \in \mathbf{a}$. We constrain the left alphabet of each $\mathbf{a} \in X(k)$ to only contain maps $\sigma : \mathbf{a} \to \Sigma$ that satisfy the following properties:

  - **Hardcoding $\mathcal{D}_{\mathsf{hdx}}$ constraints.** For any $(u, v) \in X(2)$ such that both vertices are contained in $\mathbf{a}$, $\sigma(u)$ and $\sigma(v)$ satisfy the constraint on (u, v) from $\mathcal{D}_{\mathsf{hdx}}$.
  - **Hardcoding for decoding consistency.** For any $t \in [n]$, and any two $u, v \in \mathbf{a}$ such that t is in the decoding neighborhood of both u and v in $\mathcal{D}_{\mathsf{hdx}}$, we have

    $$D_t^{\mathsf{hdx}}(u, \sigma(u)) = D_t^{\mathsf{hdx}}(v, \sigma(v)).$$

- **Right Alphabet.** For each $\mathbf{b} \in X(\sqrt{k})$, its alphabet is $\Sigma^{\sqrt{k}}$, and we similarly think of each symbol as a map from $R \to \Sigma$ assigning a Σ symbol to each $v \in \mathbf{b}$.

- **Constraints.** For each edge of the constraint graph, $e = (\mathbf{a}, \mathbf{b})$, the constraint Φ_e checks that the assignments to $\mathbf{a}$ and $\mathbf{b}$ agree. That is, if the left and right vertices are assigned symbols $\sigma \in \Sigma^k, \sigma' \in \Sigma^{\sqrt{k}}$ respectively,

  $$\Phi_e(\sigma, \sigma') = 1 \quad \text{if and only if} \quad \sigma|_{\mathbf{b}} = \sigma',$$

  where $\sigma|_{\mathbf{b}} \in \Sigma^{\sqrt{k}}$ is the restriction of σ to the vertices in $\mathbf{b}$.

- **Decoding Distributions:** Let $\mathcal{Q}_{\mathsf{hdx}}$ be the complete decoding distribution (over $[n] \times X(1)$) of $\mathcal{D}_{\mathsf{hdx}}$ wherein one chooses $t \in [n]$ uniformly, $u \sim \mathcal{P}_{\mathsf{hdx}}(t)$, and outputs (t, u). The total decoding distribution of $\mathcal{D}_{\mathsf{dp}}$, denoted by $\mathcal{Q}_{\mathsf{dp}}$, is a distribution over triples $(t, \mathbf{a}, \mathbf{b}) \in [n] \times X(k) \times X(\sqrt{k})$ is as follows:

  - Choose $\mathbf{a} \sim X(k)$, and then $\mathbf{b} \subseteq \mathbf{a}$ uniformly.
  - Choose $u \in \mathbf{b}$ uniformly and choose $t \sim \mathcal{Q}_{\mathsf{hdx}}(\cdot, u)$.

  Then the decoding distribution $\mathcal{P}_t^{\mathsf{dp}}$ is defined as $\mathcal{Q}_{\mathsf{dp}}(t, \cdot, \cdot)$.

- **Decoders:** For $(t, \mathbf{a}, \mathbf{b}) \in \mathrm{supp}(\mathcal{Q}_{\mathsf{dp}})$ and left alphabet symbol $\sigma \in \Sigma^k$ for $\mathbf{a}$, the decoder outputs

  $$D_t^{\mathsf{dp}}(\mathbf{a}, \sigma_{\mathbf{a}}) = D_t^{\mathsf{hdx}}(u, \sigma_{\mathbf{a}}(u)).$$

  where $u \in X(1)$ is an arbitrary vertex contained in both $\mathbf{a}$ and $\mathrm{supp}(\mathcal{P}_t^{\mathsf{hdx}})$. Due to the hardcoding for decoding consistency, the above value is the same for any u and hence the decoder is well defined.

The above completes the description of the new dPCP, and the remainder of the section is dedicated to its analysis.

6.1 Proof of Theorem 6.1: Everything Except the Soundness

In this section we analyze the length, alphabet size, decoding degree, decision complexity and the completeness of the construction.

Length. The length is $|X(k)| + |X(\sqrt{k})| = O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon(\log N))$ by the properties of X from Theorem 2.9.

Alphabet Size. The left and right alphabet sizes are at most $|\Sigma|^k$ and $|\Sigma|^{\sqrt{k}}$ respectively. Since $k = O_\varepsilon(1)$ and $|\Sigma| = 2^{O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))}$, both the left and right alphabet sizes are $2^{O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))}$.

Degrees. For $\mathbf{a} \in X(k)$, its degree is $\binom{k}{\sqrt{k}} = O_\varepsilon(1)$ and its decoding degree is at most the sum of the decoding degrees, in $\mathcal{D}_{\mathsf{hdx}}$, of the vertices $u \in \mathbf{a}$. Hence, the decoding degree is at most $k\mathsf{poly}_\varepsilon(\log N) \leqslant \mathsf{poly}_\varepsilon(\log N)$. For $\mathbf{b} \in X(\sqrt{k})$, its degree is at most $d(\mathbf{b}) \cdot d^k$, where $d(\mathbf{b})$ is the number of d-faces of the complex X containing $\mathbf{b}$. Since $d(\mathbf{b}) \leqslant \mathsf{poly}_\varepsilon(\log N)$ by Theorem 2.9, the right degree is $\mathsf{poly}_\varepsilon(\log N)$.

Projection Decision Complexity. Given a left alphabet symbol, it is clear that the unique right alphabet symbol which could satisfy the constraint is simply a restriction, so it can be computed with a circuit of size $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$. Additionally, the projection circuit needs to checking that the left alphabet symbol $\sigma \in \Sigma^k$ satisfies (1) the hardcoding of $\mathcal{D}_{\mathsf{hdx}}$ constraints, and (2) checking that the left alphabet symbol $\sigma \in \Sigma^k$ satisfies hardcoding for decoding consistency.

The circuit complexity of (1) is $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$ as there are $O_\varepsilon(1)$ edges $e \in X(2)$ contained in $\mathbf{a}$ and each $\mathcal{D}_{\mathsf{hdx}}$ constraint has circuit complexity $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$. The circuit complexity of (2) is also $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$. This is because, by the decoding degree of $\mathcal{D}$, there are $\mathsf{poly}_\varepsilon(\log N)$ pairs of $u \in \mathbf{a}$ and $t \in [n]$ such that t is in the decoding neighborhood of u in $\mathcal{D}_{\mathsf{hdx}}$ and the decoders $\mathcal{D}_{\mathsf{hdx},t}$ all have circuit complexity $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$ by the decoding complexity of Theorem 5.1. Finally, computing the restriction and checking that is is equal to a given string amounts to equality between two strings of $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$ bits, which can be done by a circuit of size $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$. Altogether, this shows that the projection decision complexity is $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$.

Decoding Distribution. To see that $\mathcal{Q}_{\mathsf{dp}}$ is agnostic, note that for any $(t, \mathbf{b}) \in \mathsf{supp}(\mathcal{Q}_{\mathsf{dp}}(\cdot, \circ, \cdot))$, both of the distributions $\mathcal{Q}_{\mathsf{dp}}(\circ, \cdot, \mathbf{b})$ and $\mathcal{Q}_{\mathsf{dp}}(t, \cdot, \mathbf{b})$ are of $\mathbf{a} \sim X(k)$ conditioned on $\mathbf{a} \supseteq \mathbf{b}$, so they are the same.

For the second property, we describe the following way of choosing $(\mathbf{a}, \mathbf{b}) \sim \mathcal{Q}(\circ, \cdot, \cdot)$:

- Choose $U \in X(d)$ uniformly at random,

- Choose $\mathbf{a} \subseteq U$ of size k uniformly and $\mathbf{b} \subseteq \mathbf{a}$ of size $\sqrt{k}$ uniformly,

- Output $(\mathbf{a}, \mathbf{b})$.

It is clear by the above that for every $\mathbf{b} \in X(\sqrt{k})$, we have

$$\mathcal{Q}(\circ, \circ, \mathbf{b}) = \sum_{\mathbf{a} \supseteq \mathbf{b}} \mathcal{Q}(\circ, \mathbf{a}, \mathbf{b}) = \sum_{\substack{\mathbf{a} \supseteq \mathbf{b} \\ U \in X(d), U \supseteq \mathbf{a}}} \frac{1}{|X(d)|} \cdot \frac{1}{\binom{d}{k}\binom{k}{\sqrt{k}}} \leqslant \frac{\mathsf{poly}_\varepsilon(\log N)}{N}. \tag{20}$$

In the last transition, we use the fact that for a fixed $\mathbf{b}$, the number of d-faces U and k-faces $\mathbf{a}$ that contain $\mathbf{b}$ is at most $\mathsf{poly}_\varepsilon(\log N)^d \leqslant \mathsf{poly}_\varepsilon(\log N)$ (as the degree of X is $\mathsf{poly}_\varepsilon(\log N)$), and also that $|X(d)| \geqslant N$.

Decoding Complexity. It is clear that given a left vertex $\mathbf{a}$ and a symbol for it, σ, the decoder runs the decoder of $\mathcal{D}_{\mathsf{hdx}}$ on a restriction of σ, and hence the decoding complexity is $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$ by Theorem 5.1.

Perfect Completeness. Fix $w \in \mathcal{L}$. By the perfect completeness of $\mathcal{D}_{\mathsf{hdx}}$, there is an assignment $A' : X(1) \to \Sigma$ which satisfies all of the constraints in $\mathcal{D}_{\mathsf{hdx}}$ and decodes to w_t with probability 1 for all $t \in [n]$. Now consider the assignments $T_1 : X(k) \to \Sigma^k$ and $T_2 : X(\sqrt{k}) \to \Sigma^{\sqrt{k}}$ such that $T_1[\mathbf{a}] = A'|_{\mathbf{a}}$ for each $\mathbf{a} \in X(k)$ and $T_2[\mathbf{b}] = A'|_{\mathbf{b}}$ for each $\mathbf{b} \in X(\sqrt{k})$. It is clear that the alphabet hardcoding is satisfied by $T_1[\mathbf{a}]$ for every $\mathbf{a} \in X(k)$. Indeed, for any $e \in X(2)$ contained in $\mathbf{a}$, the restriction of $T_1[\mathbf{a}]$ to the vertices in e is consistent with A' and hence the constraint e is satisfied by the corresponding values of $T_1[\mathbf{a}]$. As for the hardcoding for decoding consistency, for any $u \in \mathbf{a}$ and $t \in [n]$ which is a decoding neighbor of u in $\mathcal{D}_{\mathsf{hdx}}$, we have that $D'_t(u, T_1[\mathbf{a}](u)) = w_t$. This latter fact also shows that T_1, T_2 decode to w_t with probability 1 for all $t \in [n]$. Finally, it is straightforward to see that all of the constraints in $\mathcal{D}_{\mathsf{dp}}$ are satisfied.

6.2 Proof of Theorem 6.1: Tools for List-Decoding Soundness

The proof of list-decoding soundness involves two steps. First, we show a list-decoding variant of the soundness guarantee of the direct product test, informally saying that there is always a small list of functions $f_i : X(1) \to \Sigma$ which explains almost all of the passing probability of the direct product test in $\mathcal{D}_{\mathsf{dp}}$. Additionally, each f_i, when viewed as an assignment to $\mathcal{D}_{\mathsf{hdx}}$, is nearly a satisfying assignment of $\mathcal{D}_{\mathsf{hdx}}$. Second, we translate the list-decoding guarantee above into a list-decoding guarantee relative to valid members of the language $\mathcal{L}$. To accomplish this, we combine the unique decoding soundness of $\mathcal{D}_{\mathsf{hdx}}$ with fact that the f_i's are nearly satisfying assignments of $\mathcal{D}_{\mathsf{hdx}}$.

The first step outlined above is achieved via the following lemma.

Lemma 6.2. *Let $T_1 : X(k) \to \Sigma^k$ be a left assignment to $\mathcal{D}_{\mathsf{dp}}$ and let δ be defined as in the construction of $\mathcal{D}_{\mathsf{dp}}$. Then, there exists an absolute constant C such that, there is a list of at most $L \leqslant \delta^{-C}$ functions $f_1, \ldots, f_L : X(1) \to \Sigma$, along with disjoint sets $\mathcal{U}_1, \ldots, \mathcal{U}_L \subseteq X(k)$ which together satisfy the following:*

- *for each $i \in [L]$, we have $\mu_k(\mathcal{U}_i) \geqslant \delta^C$ and $\mathsf{dist}(f_i|_{\boldsymbol{a}}, T_1[\boldsymbol{a}]) \leqslant \delta$ for all $\boldsymbol{a} \in \mathcal{U}_i$;*

- *for any right assignment $T_2 : X(\sqrt{k}) \to \Sigma^{\sqrt{k}}$, we have*

$$\Pr_{\boldsymbol{a} \sim X(k), \boldsymbol{b} \subseteq \boldsymbol{a}}[T_1[\boldsymbol{a}]|_{\boldsymbol{b}} = T_2[\boldsymbol{b}] \wedge \boldsymbol{a} \notin \cup_{i=1}^{L} \mathcal{U}_i] \leqslant \frac{\sqrt{\delta}}{10}.$$

Towards the proof of Lemma 6.2, we first show an auxiliary lemma which says that there exists an assignment $X(k) \to \Sigma^k$ which has only trivial agreement with all possible functions $F : X(1) \to \Sigma$.

Lemma 6.3. *There exists a table $T^* : X(k) \to \Sigma^k$ such that for any function $F : X(1) \to \Sigma$,*

$$\Pr_{\boldsymbol{a} \sim X(k)}\left[\mathsf{dist}(F|_{\boldsymbol{a}}, T^*[\boldsymbol{a}]) \leqslant \frac{1}{3}\right] \leqslant \frac{10}{\sqrt{d}},$$

where recall d is the dimension of the complex X.

Proof. We use the fact that the direct-product dimension parameter, k, in constructing $\mathcal{D}_{\mathsf{dp}}$ is large and in particular $k \geqslant 8$. First let us show that there exists a table T^* satisfying such that every pair for every pair $\mathbf{a}, \mathbf{a}' \in X(k)$ with nonempty intersection, we have that $T^*[\mathbf{a}]$ and $T^*[\mathbf{a}']$ disagree at every $x \in \mathbf{a} \cap \mathbf{a}'$. Indeed, $|\Sigma|$ is super-polynomial in n whereas $k \cdot |X(k)| \leqslant n^{O_\varepsilon(1)}$, and hence we can arrange that the $T^*[\mathbf{a}] : \mathbf{a} \to \Sigma$ have disjoint image over all $\mathbf{a} \in X(k)$.

Suppose for the sake of contradiction that this is not the case, and that there exists $F : X(1) \to \Sigma$, such that

$$\Pr_{\mathbf{a} \sim X(k)}[\mathsf{dist}(F|_{\mathbf{a}}, T^*[\mathbf{a}]) \leqslant \frac{1}{3}] > \frac{10}{\sqrt{d}} \tag{21}$$

Now consider the probability that, when choosing $U \sim X(d)$, $\mathbf{b} \subseteq U$ of size $k - 1$, and finally $\mathbf{a}, \mathbf{a}'$ of size k such that $\mathbf{b} \subseteq \mathbf{a}, \mathbf{a}' \subseteq U$, we have both $\mathsf{dist}(F|_{\mathbf{a}}, T^*[\mathbf{a}]) \leqslant \frac{1}{3}$ and $\mathsf{dist}(F|_{\mathbf{a}'}, T^*[\mathbf{a}']) \leqslant \frac{1}{3}$. For simplicity, let us refer to these two events as $\mathsf{E}(\mathbf{a})$ and $\mathsf{E}(\mathbf{a}')$ respectively. Then,

$$\mathbb{E}_{U \sim X(d), \mathbf{b} \subseteq U}\left[\Pr_{\mathbf{b} \subseteq \mathbf{a}, \mathbf{a}' \subseteq U}[\mathsf{E}(\mathbf{a}) \wedge \mathsf{E}(\mathbf{a}')]\right] = \mathbb{E}_{U \sim X(d), \mathbf{b} \subseteq U}\left[\Pr_{\mathbf{a}}[\mathsf{E}(\mathbf{a}) \mid U, \mathbf{b}]^2\right] \geqslant \mathbb{E}_{U \sim X(d), \mathbf{b} \subseteq U}\left[\Pr_{\mathbf{a}}[\mathsf{E}(\mathbf{a}) \mid U, \mathbf{b}]\right]^2 > \frac{100}{d}.$$

In the first transition we used the fact that $\mathbf{a}, \mathbf{a}'$ are independent, in the second transition we used the Cauchy-Schwarz inequality, and in the third transition we used (21). Now using the fact that the probability of choosing $\mathbf{a} = \mathbf{a}'$ is at most $\frac{1}{d-k+1} \leqslant \frac{100}{d}$, we get there exist $\mathbf{a}$ and $\mathbf{a}'$ distinct such that both $\mathsf{E}(\mathbf{a})$ and $\mathsf{E}(\mathbf{a}')$ occur. Then $T^*[\mathbf{a}]$ and $T^*[\mathbf{a}']$ disagree on at most $\frac{2}{3}k + 1 < k$ coordinates, which is a contradiction. $\square$

Proof of Lemma 6.2. Initialize $\mathcal{U}_1 = \mathcal{U}_1^\star = \emptyset$ and $i := 1$. Also set $T^{(1)} := T_1$. While there exists $F : X(1) \to \Sigma$ such that

$$\Pr_{\mathbf{a} \sim X(k)}[\mathsf{dist}(F|_\mathbf{a}, T^{(i)}[\mathbf{a}]) \leqslant \delta \wedge \mathbf{a} \notin \mathcal{U}_i^\star] \geqslant \delta^C$$

pick such F, set $f_i = F$, set $\mathcal{U}_i = \{\mathbf{a} \in X(k) \mid \mathsf{dist}(F|_\mathbf{a}, T^{(i)}[\mathbf{a}]) \leqslant \delta \wedge \mathbf{a} \notin \mathcal{U}_i^\star\}$, and update $\mathcal{U}_{i+1}^\star = \mathcal{U}_i^\star \cup \mathcal{U}_i$. Then increment i by 1, and repeat. In the above we fix C to be some absolute constant which we can set arbitrarily large.

Note that this process must terminate in $J \leqslant \delta^{-C}$ steps. Indeed, after each increment of i, the measure $\mu_k(\mathcal{U}_i^\star)$ increases by at least δ^C. At the end of the process, let $f_1, \ldots, f_J$ and $\mathcal{U}_1, \ldots, \mathcal{U}_J$ be the functions and sets obtained respectively, and let $\mathcal{U}^\star := \bigcup_{i \in [J]} \mathcal{U}_i$. The following conditions hold:

- For each $i \in [J]$ and $\mathbf{a} \in \mathcal{U}_i$ we have $\mathsf{dist}(f_i|_\mathbf{a}, T_1[\mathbf{a}]) \leqslant \delta$.

- $\mu_k(\mathcal{U}_i) \geqslant \delta^C$,

- The sets $\mathcal{U}_i$ are disjoint.

- For any $\mathbf{a} \notin \mathcal{U}^\star$, and any $i \in [J]$, we have $\mathsf{dist}(f_i|_\mathbf{a}, T[\mathbf{a}]) > \delta$.

- For any $F : X(1) \to \Sigma$, we have

$$\Pr_{\mathbf{a} \sim X(k)}[\mathsf{dist}(T[\mathbf{a}], F|_\mathbf{a}) \leqslant \delta \wedge \mathbf{a} \notin \mathcal{U}^\star] \leqslant \delta^C. \tag{22}$$

We will now show that $f_1, \ldots, f_J$ and $\mathcal{U}_1, \ldots, \mathcal{U}_J$ satisfy Lemma 6.2. First, note that the number of functions is indeed $J \leqslant \delta^{-C}$ for some absolute constant C and the sets $\mathcal{U}_i$ are indeed disjoint as required by Lemma 6.2. It is also clear that the first item of Lemma 6.2 holds by construction of the $\mathcal{U}_i$'s.

It remains to verify the second item. Let T^* be the table from Lemma 6.3 — which, recall, does not have significant agreement with any function over $X(1)$ — and define the table T' as follows. For $\mathbf{a} \in \mathcal{U}^\star$, set $T'[\mathbf{a}] = T^*[\mathbf{a}]$ and for $\mathbf{a} \notin \mathcal{U}^\star$, set $T'[\mathbf{a}] = T[\mathbf{a}]$. Suppose for the sake of contradiction that the second item fails, so that

$$\Pr_{\mathbf{a} \sim X(k), \mathbf{b} \subseteq \mathbf{a}}[T'[\mathbf{a}]|_\mathbf{b} = T_2[\mathbf{b}]] > \frac{\sqrt{\delta}}{10}. \tag{23}$$

Choosing $U \in X(d)$, $\mathbf{b} \subseteq U$ of size $\sqrt{k}$, and $\mathbf{a}, \mathbf{a}'$ of size k such that $\mathbf{b} \subseteq \mathbf{a}, \mathbf{a}' \subseteq U$ according to the direct product testing distribution, we have

$$
\begin{aligned}
\Pr_{U,\mathbf{a},\mathbf{a}',\mathbf{b}}[T'[\mathbf{a}]|_\mathbf{b} = T'[\mathbf{a}']|_\mathbf{b}] &\geqslant \mathbb{E}_U\left[\Pr_{\mathbf{a},\mathbf{a}',\mathbf{b}}[T'[\mathbf{a}]|_\mathbf{b} = T_2[\mathbf{b}] \wedge T'[\mathbf{a}']|_\mathbf{b} = T_2[\mathbf{b}]]\right] \\
&= \mathbb{E}_U\left[\Pr_{\mathbf{a},\mathbf{b}}[T'[\mathbf{a}]|_\mathbf{b} = T_2[\mathbf{b}]]^2\right] \\
&\geqslant \left(\mathbb{E}_U\left[\Pr_{\mathbf{a},\mathbf{b}}[T'[\mathbf{a}]|_\mathbf{b} = T_2[\mathbf{b}]]\right]\right)^2 \\
&\geqslant \frac{\delta}{100}.
\end{aligned}
\tag{24}
$$

In the third transition we used by Cauchy-Schwarz, and in the last transition we used (23). Since the complex X supports a direct product test with soundness $\delta/100$, we get that there exists $F : X(1) \to \Sigma$ such that

$$\Pr_{\mathbf{a} \sim X(k)}\left[\mathsf{dist}(T'[\mathbf{a}], F|_\mathbf{a}) \leqslant \frac{\delta}{100}\right] \geqslant \delta^{C_1},$$

for some absolute constant C_1 coming from the soundness of the direct product test as described in Theorem 2.9. However, this is a contradiction because

$$
\begin{aligned}
\Pr_{\mathbf{a}\sim X(k)}\left[\mathsf{dist}(T'[\mathbf{a}], F|_{\mathbf{a}}) \leqslant \frac{\delta}{100}\right] &= \Pr_{\mathbf{a}\sim X(k)}\left[\mathsf{dist}(T'[\mathbf{a}], F|_{\mathbf{a}}) \leqslant \frac{\delta}{100} \wedge \mathbf{a} \in \mathcal{U}^\star\right] \\
&\quad + \Pr_{\mathbf{a}\sim X(k)}\left[\mathsf{dist}(T'[\mathbf{a}], F|_{\mathbf{a}}) \leqslant \frac{\delta}{100} \wedge \mathbf{a} \notin \mathcal{U}^\star\right] \\
&\leqslant \Pr_{\mathbf{a}\sim X(k)}\left[\mathsf{dist}(T^*[\mathbf{a}], F|_{\mathbf{a}}) \leqslant \frac{\delta}{100}\right] + \delta^C \\
&\leqslant \frac{10}{\sqrt{d}} + \delta^C \\
&< \delta^{C_1}.
\end{aligned}
$$

where in the second transition we used (22) along with the construction of T' and in the third transition we used $\delta/100 \leqslant 1/3$ and Lemma 6.3. To obtain the contradiction in the last transition, we use the fact that the dimension parameter of the HDX X is sufficiently large relative to $1/\delta$ and the fact that the absolute constant C fixed at the start of the proof can be chosen sufficiently large relative to the constant C_1 coming from the direct product test soundness. $\qquad\square$

6.3 Proof of Theorem 6.1: Proof of List-Decoding Soundness

We now accomplish the second step in the proof of the list-decoding soundness. Fix a left assignment, $T_1 : X(k) \to \Sigma^k$, to $\mathcal{D}_{\mathsf{dp}}$. Let $f_1, \ldots, f_L$ and $\mathcal{U}_1, \ldots, \mathcal{U}_L$ be the functions and subsets of $X(k)$ obtained by applying Lemma 6.2 with respect to T_1, and let $T_2 : X(\sqrt{k}) \to \Sigma^{\sqrt{k}}$ be any right assignment to $\mathcal{D}_{\mathsf{dp}}$. We start by observing that each $f_i : X(1) \to \Sigma$ is a nearly-satisfying assignment to $\mathcal{D}_{\mathsf{hdx}}$.

Lemma 6.4. *For each $i \in [L]$, we have*

$$
\Pr_{(u,v)\sim X(2)}[\Phi^{\mathsf{hdx}}_{(u,v)}(f_i(u), f_i(v)) = 1] \geqslant 1 - 3\delta.
$$

Proof. Fix i and let $\mathcal{B} \subseteq X(2)$ be the set of constraints not satisfied by f_i, so that $\mu_2(\mathcal{B})$ is precisely the probability that a randomly chosen constraint, according to the measure of X over $X(2)$, is not satisfied by f_i. To prove the lemma, we show $\mu_2(\mathcal{B}) \leqslant 3\delta$.

To start, notice that if there exists $\mathbf{a} \in X(k)$ such that $T_1[\mathbf{a}](u) = f_i(u)$ and $T_1[\mathbf{a}](v) = f_i(v)$, then we automatically have $\Phi^{\mathsf{hdx}}_{(u,v)}(f_i(u), f_i(v)) = 1$ due to how the alphabet of $\mathbf{a}$ is hardcoded. This shows that a constraint is highly likely to be satisfied if it is chosen by first choosing $\mathbf{a} \sim \mathcal{U}_i$, and then choosing $u, v \in \mathbf{a}$. In the sequence below, we write $\mathbf{a} \sim \mathcal{U}_i$ to denote $\mathbf{a}$ chosen according to $\sim X(k)$, conditioned on being in $\mathcal{U}_i$. After $\mathbf{a}$ is chosen, u and v are chosen as uniformly random vertices inside $\mathbf{a}$.

$$
\begin{aligned}
\Pr_{\mathbf{a}\sim\mathcal{U}_i, u,v\in\mathbf{a}, u\neq v}[(u,v) \notin \mathcal{B}] &\geqslant \mathbb{E}_{\mathbf{a}\sim\mathcal{U}_i}\left[\Pr_{u,v\in\mathbf{a}, u\neq v}[T_1[\mathbf{a}](u) = f_i(u) \wedge T_1[\mathbf{a}](v) = f_i(v)]\right] \\
&\geqslant \mathbb{E}_{\mathbf{a}\sim\mathcal{U}_i}\left[\Pr_{u\in\mathbf{a}}[T_1[\mathbf{a}](u) = f_i(u)]^2\right] - \frac{1}{k} \\
&\geqslant (1-\delta)^2 - \frac{1}{k} \\
&\geqslant 1 - 2\delta - \frac{1}{k}.
\end{aligned}
\tag{25}
$$

In the second transition we are going from a probability over distinct $u, v \in \mathbf{a}$ to a probability over $u, v \in \mathbf{a}$ independent, and the $-1/k$ term accounts for the probability that $u = v$. In the third transition we use the Cauchy-Schwarz inequality and the fact that f_i and $T_1[\mathbf{a}]$ agree on at least $(1 - \delta)$-fraction of vertices in $\mathbf{a}$ for every $\mathbf{a} \in \mathcal{U}_i$.

Next, by item 3 of Theorem 2.9 and the fact that d is sufficiently large relative to k, we get that the second singular value of the bipartite inclusion graph $(X(k), X(2))$ is $O(1/k)$. Applying Lemma 2.3 with the left and right subsets are $\mathcal{U}_i$ and $\mathcal{B}$ respectively gives:

$$\left| \Pr_{\mathbf{a} \sim X(k), \{u,v\} \in \binom{\mathbf{a}}{2}} [\mathbf{a} \in \mathcal{U}_i, \{u,v\} \in \mathcal{B}] - \mu_k(\mathcal{U}_i)\mu_2(\mathcal{B}) \right| \leqslant O\left(\frac{1}{k}\right) \cdot \sqrt{\mu_k(\mathcal{U}_i)\mu_2(\mathcal{B})}.$$

Dividing both sides by $\mu_k(\mathcal{U}_i)$ and rearranging gives

$$\mu_2(\mathcal{B}) \leqslant \Pr_{\mathbf{a} \sim \mathcal{U}_i, \{u,v\} \in \binom{\mathbf{a}}{2}} [(u,v) \in \mathcal{B}] + O\left(\frac{1}{k}\right) \cdot \frac{\sqrt{\mu_k(\mathcal{U}_i)\mu_2(\mathcal{B})}}{\mu_k(\mathcal{U}_i)}$$

$$\leqslant 2\delta + \frac{1}{k} + O\left(\frac{1}{k\sqrt{\mu_k(\mathcal{U}_i)}}\right)$$

$$\leqslant 3\delta,$$

where in the transition we used (25), and in the last transition we used $\mu_k(\mathcal{U}_i) \geqslant \delta^C$ and the fact that k is sufficiently large relative to $1/\delta$. $\qquad\square$

Now combining Lemma 6.4 with the unique decoding soundness assumption of $\mathcal{D}_{\mathsf{hdx}}$ from Theorem 5.1, we get that for each $i \in [L]$, there is $w_i \in \mathcal{L}$ such that

$$\Pr_{(t,u) \sim \mathcal{Q}_{\mathsf{hdx}}} [D_t^{\mathsf{hdx}}(u, f_i(u)) \neq (w_i)_t] \leqslant 5\sqrt{\delta}, \tag{26}$$

and we recall that the marginal distribution of u in the probability above is the stationary over $X(1)$. We are now ready to prove list-decoding soundness of $\mathcal{D}_{\mathsf{dp}}$ by showing that the w_i obtained by (26) give the desired list.

Proof of List-Decoding Soundness for Theorem 6.1. Take $L = \delta^{-C}$ and $f_i, \mathcal{U}_i, w_i$ as above for each $i \in [L]$. Set $\mathcal{U}^\star := \bigcup_{i \in [L]} \mathcal{U}_i$. Then

$$\Pr_{(t,\mathbf{a},\mathbf{b}) \sim \mathcal{Q}_{\mathsf{dp}}} \left[T_1[\mathbf{a}]|_{\mathbf{b}} = T_2[\mathbf{b}] \wedge D_t(\mathbf{a}, T_1[\mathbf{a}]) \notin \{(w_i)_t\}_{i \in [L]} \right]$$

$$= \Pr_{(t,\mathbf{a},\mathbf{b}) \sim \mathcal{Q}_{\mathsf{dp}}} \left[T_1[\mathbf{a}]|_{\mathbf{b}} = T_2[\mathbf{b}] \wedge D_t(\mathbf{a}, T_1[\mathbf{a}]) \notin \{(w_i)_t\}_{i \in [L]} \wedge \mathbf{a} \in \mathcal{U}^\star \right]$$

$$+ \Pr_{(t,\mathbf{a},\mathbf{b}) \sim \mathcal{Q}_{\mathsf{dp}}} \left[T_1[\mathbf{a}]|_{\mathbf{b}} = T_2[\mathbf{b}] \wedge D_t(\mathbf{a}, T_1[\mathbf{a}]) \notin \{(w_i)_t\}_{i \in [L]} \wedge \mathbf{a} \notin \mathcal{U}^\star \right]$$

$$\leqslant \Pr_{(t,\mathbf{a},\mathbf{b}) \sim \mathcal{Q}_{\mathsf{dp}}} \left[D_t(\mathbf{a}, T_1[\mathbf{a}]) \notin \{(w_i)_t\}_{i \in [L]} \wedge \mathbf{a} \in \mathcal{U}^\star \right]$$

$$+ \Pr_{(t,\mathbf{a},\mathbf{b}) \sim \mathcal{Q}_{\mathsf{dp}}} \left[T_1[\mathbf{a}]|_{\mathbf{b}} = T_2[\mathbf{b}] \wedge \mathbf{a} \notin \mathcal{U}^\star \right]$$

$$\leqslant \Pr_{(t,\mathbf{a},\mathbf{b}) \sim \mathcal{Q}_{\mathsf{dp}}} \left[D_t(\mathbf{a}, T_1[\mathbf{a}]) \notin \{(w_i)_t\}_{i \in [L]} \wedge \mathbf{a} \in \mathcal{U}^\star \right] + O\left(\sqrt{\delta}\right),$$

where in the last transition, we are using the second item of Lemma 6.2 to bound

$$\Pr_{(t,\mathbf{a},\mathbf{b}) \sim \mathcal{Q}_{\mathsf{dp}}} [T_1[\mathbf{a}]|_{\mathbf{b}} = T_2[\mathbf{b}] \wedge \mathbf{a} \notin \mathcal{U}^\star] = \Pr_{\mathbf{a} \sim X(k), \mathbf{b} \subseteq \mathbf{a}} [T_1[\mathbf{a}]|_{\mathbf{b}} = T_2[\mathbf{b}] \wedge \mathbf{a} \notin \mathcal{U}^\star] \leqslant \frac{\sqrt{\delta}}{10}.$$

Next, we perform a union bound

$$\Pr_{(t,\mathbf{a},\mathbf{b}) \sim \mathcal{Q}_{\mathsf{dp}}} \left[D_t(\mathbf{a}, T_1[\mathbf{a}]) \notin \{(w_i)_t\}_{i \in [L]} \wedge \mathbf{a} \in \mathcal{U}^\star \right] \leqslant \sum_{i=1}^{L} \Pr_{(t,\mathbf{a},\mathbf{b}) \sim \mathcal{Q}_{\mathsf{dp}}} \left[D_t(\mathbf{a}, T_1[\mathbf{a}]) \neq (w_i)_t \wedge \mathbf{a} \in \mathcal{U}_i \right]$$

$$\leqslant \mathbb{E}_{i \sim [L]} \left[\Pr_{(t,\mathbf{a},\mathbf{b}) \sim \mathcal{Q}_{\mathsf{dp}}} [D_t(\mathbf{a}, T_1[\mathbf{a}]) \neq (w_i)_t \mid \mathbf{a} \in \mathcal{U}_i] \right].$$

In the last line we let $i \sim [L]$ be the distribution where i is chosen with weight proportional to $\mu_k(\mathcal{U}_i)$. Then, the last line follows from the fact that the $\mathcal{U}_i$'s are disjoint, so the events $\mathbf{a} \in \mathcal{U}_i$ are mutually exclusive. Fix i and examine a probability from the last line more closely. Here, we unpack the joint distribution of $(t, \mathbf{a})$ under $\mathcal{Q}_{\mathsf{dp}}$. By definition, this distribution is obtained by sampling $\mathbf{a} \sim X(k)$, then a random $u \in \mathbf{a}$ and finally $t \sim \mathcal{Q}_{\mathsf{hdx}}(\cdot, u)$. We have

$$\Pr_{(t,\mathbf{a},\mathbf{b})\sim\mathcal{Q}_{\mathsf{dp}}}[D_t(\mathbf{a}, T_1[\mathbf{a}]) \neq (w_i)_t \mid \mathbf{a} \in \mathcal{U}_i] \leqslant \Pr_{\mathbf{a}\sim\mathcal{U}_i, u\in\mathbf{a}, t\sim\mathcal{Q}_{\mathsf{hdx}}(\cdot,u)}[D_t^{\mathsf{hdx}}(u, f_i(u)) \neq (w_i)_t] \tag{27}$$
$$+ \Pr_{\mathbf{a}\sim\mathcal{U}_i, u\in\mathbf{a}, t\sim\mathcal{Q}_{\mathsf{hdx}}(\cdot,u)}[T_1[\mathbf{a}](u) \neq f_i(u)].$$

Now we bound each term individually. Towards the first term, define the function $H_i : X(1) \to [0, 1]$ as

$$H_i(u) = \Pr_{t\sim\mathcal{Q}_{\mathsf{hdx}}(\cdot,u)}[D_t^{\mathsf{hdx}}(u, f_i(u)) \neq (w_i)_t].$$

In particular, from (26) we have

$$\mu_1(H_i) := \mathbb{E}_{u\sim X(1)}[H_i(u)] = \Pr_{(t,u)\sim\mathcal{Q}_{\mathsf{hdx}}}[D_t^{\mathsf{hdx}}(u, f_i(u)) \neq (w_i)_t] \leqslant 5\sqrt{\delta}. \tag{28}$$

By Lemma 2.3 and the fact that the bipartite inclusion graph on $(X(k), X(1))$ has second singular value at most $O(1/k)$ by item 3 of Theorem 2.9 (and the fact that d is sufficiently large relative to k), it follows that

$$\left| \mathbb{E}_{\mathbf{a}\sim X(k), u\in\mathbf{a}}[\mathbb{1}_{\mathbf{a}\in\mathcal{U}_i} \cdot H_i(u)] - \mu_k(\mathcal{U}_i)\mu_1(H_i) \right| \leqslant O\left(\frac{1}{k}\right) \cdot \sqrt{\mu_k(\mathcal{U}_i)}. \tag{29}$$

Now note that the first term on the right hand side of (27) is

$$\Pr_{\mathbf{a}\sim\mathcal{U}_i, u\in\mathbf{a}, t\sim\mathcal{Q}_{\mathsf{hdx}}(\cdot,u)}[D_t^{\mathsf{hdx}}(u, f_i(u)) \neq (w_i)_t] = \mathbb{E}_{\mathbf{a}\sim X(k), u\in\mathbf{a}}[H_i(u) \mid \mathbf{a} \in \mathcal{U}_i] = \frac{\mathbb{E}_{\mathbf{a}\sim X(k), u\in\mathbf{a}}[\mathbb{1}_{\mathbf{a}\in\mathcal{U}_i} \cdot H_i(u)]}{\mu_k(\mathcal{U}_i)}.$$

Thus, by (29),

$$\Pr_{\mathbf{a}\sim\mathcal{U}_i, u\in\mathbf{a}, t\sim\mathcal{Q}_{\mathsf{hdx}}(\cdot,u)}[D_t^{\mathsf{hdx}}(u, f_i(u)) \neq (w_i)_t] \leqslant 5\sqrt{\delta} + O\left(\frac{1}{k \cdot \sqrt{\mu_k(\mathcal{U}_i)}}\right) \leqslant 6\sqrt{\delta}.$$

The first inequality follows by (28) and the second inequality follows from the fact that $\mu_k(\mathcal{U}_i) \geqslant \delta^C$ and k is sufficiently large relative to $1/\delta$.

For the second term on the right hand side of (27), we have that

$$\Pr_{\mathbf{a}\sim\mathcal{U}_i, u\in\mathbf{a}}[T_1[\mathbf{a}](u) \neq f_i(u)] = \mathbb{E}_{\mathbf{a}\sim\mathcal{U}_i}[\mathsf{dist}(f_i|_{\mathbf{a}}, T_1[\mathbf{a}])] \leqslant \delta$$

by definition of f_i and $\mathcal{U}_i$. Altogether, we get that (27) is at most $6\sqrt{\delta} + \delta \leqslant 7\sqrt{\delta}$, and plugging this up gives that the list-decoding error is at most $O(\sqrt{\delta})$. $\qquad\qquad\square$

7 Transformations on dPCPs

Theorem 6.1 gives us constructions of quasi-linear length dPCPs with arbitrarily small error in the list-decoding soundness, as required by Theorem 3.12, but with alphabet size that is super-polynomial, rather than constant. Our goal now is to reduce the alphabet size in Theorem 6.1 down to a constant while maintaining all of the other desired properties; namely, length and small soundness. As discussed in the introduction, we achieve this via a series of transformations, and in this section we present a few of the basic ones here. The transformations in this section are both variants of well known transformations for PCPs.

7.1 Right Alphabet Reduction

We start with right alphabet reduction. This procedure is a simple application of error correcting codes along the lines of [DH13, Section 5.2] except that for our purposes, we also need to keep track of the degree related parameters and analyze the list-decoding soundness (rather than the standard soundness).

Lemma 7.1. *Suppose a language $\mathcal{L}$ has a projection dPCP*

$$\mathcal{D} = \big(A \cup B, E, \Sigma_A, \Sigma_B, \{\mathcal{P}_t\}_{t \in [n]}, \{D_t\}_{t \in [n]}\big)$$

with perfect completeness, (L, ε)-list-decoding soundness, and projection decision complexity $\mathsf{ProjComp}$. Then for every $0 < \eta < 1$, there is a polynomial-time algorithm which takes $\mathcal{D}$ and outputs a new projection dPCP for $\mathcal{L}$,

$$\mathcal{D}' = \big(A \cup (B \times [m]), E', \Sigma_A, \Omega, \{\mathcal{P}'_t\}_{t \in [n]}, \{D_t\}_{t \in [n]}\big)$$

where $m = O_\eta(\log |\Sigma_B|)$ and Ω is an alphabet of size $O_\eta(1)$. Additionally, $\mathcal{D}'$ satisfies the following:

- ***Degrees.*** *The decoding degree of each vertex and right degree of each vertex are the same as in $\mathcal{D}$. The left degree of each vertex is m times its left degree in $\mathcal{D}$.*

- ***Projection Decision Complexity.*** *The decision complexity is $\mathsf{ProjComp} + O(\log |\Sigma_B|)$.*

- ***Decoding Complexity.*** *The decoding complexity is preserved.*

- ***Preserves Agnosticity.*** *If the complete decoding distribution of $\mathcal{D}$ is agnostic, then so is that of $\mathcal{D}'$.*

- ***Completeness.*** *$\mathcal{D}'$ has perfect completeness.*

- ***Soundness.*** *$(L, \varepsilon + 3\eta)$-list-decoding soundness.*

Proof. Fix $\eta \in (0, 1)$ and let $\mathcal{D}$ denote the original dPCP. Let us arbitrarily identify Σ_B with $\{0, 1\}^k$ for $k = \log(|\Sigma_B|)$ and take $\mathcal{C} : \{0, 1\}^k \to \Omega^m$ to be the code from Lemma 2.1 instantiated with relative distance $1 - \eta^3$, alphabet size $|\Omega| = O_\eta(1)$, and blocklength $m = O_\eta(k)$.

The new dPCP $\mathcal{D}'$ is constructed as follows. The constraint graph is bipartite and has vertices $A \cup B \times [m]$. The edges of the constraint graph are all $(a, (b, i)) \subseteq A \times (B \times [m])$ such that $(a, b) \in E$. The constraint on $(a, (b, i))$ is as follows. Let σ_a be label for a and ω be a label for (b, i). Since $\mathcal{D}$ has projection constraints, there is a unique $\sigma_b \in \Sigma_B$ satisfying $\Phi_{(a,b)}(\sigma_a, \sigma_b) = 1$. The constraint on $\mathcal{D}'$ checks if $\mathcal{C}(\sigma_b)_i = \omega$. The new decoding distributions $\mathcal{P}'_t$ are generated by choosing $(a, b) \sim \mathcal{P}_t$, choosing $i \in [m]$ uniformly, and outputting $(a, (b, i))$. We proceed to verifying the desired properties.

Degrees. It is clear that decoding degree is preserved, as each $a \in A$ has the same decoding neighborhood in both $\mathcal{D}$ and $\mathcal{D}'$. The right degree is preserved because each $(b, i) \in B \times [m]$ has the same neighbors in the new constraint graph as neighbors as b in the original constraint graph. For the left degree, note that each $a \in A$ is adjacent to (b, i) for all b that are adjacent to a in the original constraint graph and all $i \in [m]$.

Projection Decision Complexity. Fix a constraint $(a, (b, i))$ in $\mathcal{D}'$ and suppose the assignment to a is σ. The projection circuit first computes the $\sigma' \in \Sigma_B$ which is the unique b alphabet symbol that would satisfy the constraint in $\mathcal{D}$. This part has circuit complexity $\mathsf{ProjComp}$. Then, the projection circuit calculates the encoding of σ', under the linear-time encodable code, and outputs its ith symbol. This part has circuit complexity $O(\log |\Sigma_B|)$ since $\mathcal{C}$ is linear-time encodable. Overall, this shows that the projection decision complexity is $\mathsf{ProjComp} + O(\log |\Sigma_B|)$.

Decoding Complexity. The decoder is unchanged and hence the decoding complexity is preserved.

Preserves Agnosticity Let $\mathcal{Q}$ and $\mathcal{Q}'$ denote the complete decoding distributions of $\mathcal{D}$ and $\mathcal{D}'$ respectively. Note that for any $(b, i) \in B \times [m]$ and $t \in [n]$, we have $\mathcal{Q}(\cdot, \cdot, b) = \mathcal{Q}'(\cdot, \cdot, (b, i))$ and $\mathcal{Q}(t, \cdot, b) = \mathcal{Q}'(t, \cdot, (b, i))$. It follows that agnosticity is preserved.

Perfect Completeness. Fix $w \in \mathcal{L}$ and let $T_1 : A \to \Sigma_A$ and $T_2 : B \to \Sigma_B$ be the assignment satisfying all of the constraints to $\mathcal{D}$ and decoding to w with probability 1 for every $t \in [n]$. Then consider the assignment to $\mathcal{D}'$ where the left assignment is still T_1 and the new right assignment, which we call $T_2' : B \times [m] \to \Omega$ is defined as

$$T_2'[(b, i)] = \mathcal{C}(T_2[b])_i.$$

It is straightforward to check that this assignment satisfies all of the constraints of $\mathcal{D}'$ (and T_1 still decodes to w with probability 1 for every $t \in [n]$ because it is unchanged).

List-Decoding Soundness. Fix an assignment to $T_1 : A \to \Sigma$ and let $w_1, \ldots, w_L \in \mathcal{L}$ be the list guaranteed by the list-decoding assumption of $\mathcal{D}$. For each $a \in A$ and $b \in B$ such that $(a, b) \in E$, we let $T_1[a]_{\to b} \in \Sigma_B$ be the unique symbol σ such that $\Phi_{(a,b)}(T_1[a], \sigma) = 1$. Let $\mathcal{Q}$ be the complete decoding distribution of $\mathcal{D}$ generated by choosing $t \in [n]$, $(a, b) \sim \mathcal{P}_t$, and outputting (t, a, b). For each $b \in B$ and $\sigma \in \Sigma_B$, define

$$\delta_b(\sigma) = \Pr_{(t,a) \sim \mathcal{Q}(\cdot, \cdot, b)}[T_1[a]_{\to b} = \sigma \wedge D_t(a, T_1[a]) \notin \{(w_i)_t\}_{i \in [L]}].$$

For any $T_2 : B \to \Sigma_B$, the expectation $\mathbb{E}_{b \sim \mathcal{Q}}[\delta_b(T_2[b])]$ is precisely the list-decoding error in $\mathcal{D}$ of T_1, T_2 relative to the list $\{w_1, \ldots, w_L\}$. Here and henceforth, $b \sim \mathcal{Q}$, refers to b sampled according to the marginal of $\mathcal{Q}$ over B. By the list-decoding soundness assumption of $\mathcal{D}$, we have

$$\mathbb{E}_{b \sim \mathcal{Q}}[\delta_b(T_2[b])] \leq \varepsilon. \tag{30}$$

for any $T_2 : B \to \Sigma_B$.

Now, fix an arbitrary right assignment $T_2' : B \times [m] \to \Omega$ for $\mathcal{D}'$. We will show that the list-decoding error of T_1 and T_2' relative to $\{w_1, \ldots, w_L\}$ is at most $\varepsilon + 3\eta$. We can express this list-decoding error as follows:

$$\mathbb{E}_{b \sim \mathcal{Q}}\left[\sum_{\sigma \in \Sigma_B} \delta_b(\sigma) \cdot \Pr_{i \in [m]}[T_2'[(b, i)] = \mathcal{C}(\sigma)_i]\right], \tag{31}$$

so our goal is to now bound the expectation above. Fix a $b \in B$ and examine the summation in the expectation. Define list_b to consist of all $\sigma \in \Sigma_B$ such that $\mathcal{C}(\sigma)$ agrees with at least η-fraction of the entries of the string $(T[(b, 1)], \ldots, T[(b, m)]) \in \Omega^m$. By the list-decoding bound in Fact 2.2 and the fact that $\eta > 2\eta^{3/2}$, we have that $|\mathsf{list}_b| \leq 2/\eta$. We define a right assignment, $T_2^* : B \to \Sigma_B$, for $\mathcal{D}$ as follows. For each $b \in B$, set $T_2^*[b]$ to be the alphabet symbol $\sigma \in \Sigma_B$ which maximizes $\delta_b(\sigma)$. Then, the summation from (31) can be bounded by

$$\sum_{\sigma \in \Sigma_B} \delta_b(\sigma) \cdot \Pr_{i \in [m]}[T_2'[(b, i)] = \mathcal{C}(\sigma)_i]$$

$$= \sum_{\sigma \in \mathsf{list}_b} \delta_b(\sigma) \cdot \Pr_{i \in [m]}[T_2'[(b, i)] = \mathcal{C}(\sigma)_i] + \sum_{\sigma \notin \mathsf{list}_b} \delta_b(\sigma) \cdot \Pr_{i \in [m]}[T_2'[(b, i)] = \mathcal{C}(\sigma)_i]$$

$$\leq \sum_{\sigma \in \mathsf{list}_b} \delta_b(\sigma) \cdot \Pr_{i \in [m]}[T_2'[(b, i)] = \mathcal{C}(\sigma)_i] + \eta$$

$$\leq \delta_b(T_2^*[b]) \cdot \left(\sum_{\sigma \in \mathsf{list}_b} \Pr_{i \in [m]}[T_2'[(b, i)] = \mathcal{C}(\sigma)_i]\right) + \eta$$

$$\leq \delta_b(T_2^*[b]) \cdot \left(\Pr_{i \in [m]}[\exists \sigma \in \mathsf{list}_b, \, T_2'[(b, i)] = \mathcal{C}(\sigma)_i] + \sum_{\sigma \neq \sigma' \in \mathsf{list}_b} \Pr_{i \in [m]}[T_2(b, i) = \mathcal{C}(\sigma)_i = \mathcal{C}(\sigma')_i]\right) + \eta$$

$$\leq \delta_b(T_2^*[b]) \cdot (1 + 2\eta) + \eta$$

$$\leq \delta_b(T_2^*[b]) + 3\eta.$$

In the second transition we used $\sum_{\sigma \in \Sigma_B} \delta_b(\sigma) \leq 1$ and that $\Pr_{i \in [m]}[T_2'[(b, i)] = \mathcal{C}(\sigma)_i] \leq \eta$ for each $\sigma \notin \mathsf{list}_b$. In the third transition we used the definition of $T_2^*[b]$. In the fourth transition we used the inclusion-exclusion principle.

In the fifth transition we used the fact that $|\text{list}_b| \leqslant 2/\eta$ and that for $\sigma \neq \sigma'$ we have $\Pr_{i \in [m]}[\mathcal{C}(\sigma)_i = \mathcal{C}(\sigma')_i] \leqslant \eta^3$. Plugging this bound back into the expectation from (31), we get that the list-decoding error of $\mathcal{D}'$ is at most

$$\mathbb{E}_{b \sim \mathcal{Q}} \left[\sum_{\sigma \in \Sigma_B} \delta_b(\sigma) \cdot \Pr_{i \in [m]}[T_2'[(b,i)] = \mathcal{C}(\sigma)_i] \right] \leqslant \mathbb{E}_{b \sim \mathcal{Q}} [\delta_b (T_2^*[b])] + 3\eta \leqslant \varepsilon + 3\eta,$$

where in the last transition we used (30) applied to the tables T_1, T_2^*. $\qquad\square$

7.2 Right Degree Reduction

The goal of this subsection is to describe a procedure which transforms an arbitrary dPCP, $\mathcal{D}$, into a right-regular dPCP $\mathcal{D}'$ with constant right degree. Additionally, in the complete decoding distribution of $\mathcal{D}'$, the marginal distribution over the right vertices is uniform. This transformation is similar to the one in Section 4.3, but some extra care is needed since we deal with list-decoding soundness rather than unique-decoding soundness.

Lemma 7.2. *Suppose a language $\mathcal{L} \subseteq \Sigma_0^n$ has a projection dPCP*

$$\mathcal{D} = \left(A \cup B, E, \{\mathcal{P}_t\}_{t \in [n]}, \Sigma_A, \Sigma_B, \{D_t\}_{t \in [n]} \right)$$

with (L, ε)-list-decoding soundness and suppose there exists $M \in \mathbb{N}$ such that for every $(a, b) \in A \times B$, one can write $\mathcal{Q}(\circ, a, b) = \frac{w(a,b)}{M}$ for some $w(a,b) \in \mathbb{N}$. Then for any $d' \geqslant \min_{a,b} w(a,b)$, there is a polynomial-time algorithm which takes $\mathcal{D}$ and outputs a projection dPCP

$$\mathcal{D}' = \left(A \cup B', E', \{\mathcal{P}_t'\}_{t \in [n]}, \Sigma_A, \Sigma_B, \{D_t\}_{t \in [n]} \right),$$

for $\mathcal{L}$ such that the following hold:

- **Length.** $|B'| = M$ *and hence the length is $|A| + M$.*

- **Degrees.** *The left degree of every vertex is multiplied by d', the decoding degree of every vertex is preserved, and the right degree of every vertex is d'.*

- **Projection decision complexity.** *The projection decision complexity is preserved.*

- **Decoding complexity.** *The decoding complexity is preserved.*

- **Complete decoding distribution.** *Call the complete decoding distribution $\mathcal{Q}'$. Then,*

  - *$\mathcal{Q}'$ is $d'M$-discrete.*
  - *For every $b \in B'$, we have $\mathcal{Q}'(\circ, \circ, b) = \frac{1}{M}$ and hence the marginal of $\mathcal{Q}'$ over B' is uniform.*
  - *For every $a \in A$, we have $\mathcal{Q}'(\circ, a, \circ) = \mathcal{Q}(\circ, a, \circ)$*
  - *If $\mathcal{Q}$ is agnostic, then $\mathcal{Q}'$ is agnostic as well.*

- **Completeness.** *The dPCP $\mathcal{D}'$ has perfect completeness.*

- **List-decoding soundness.** *$(L, \varepsilon + O(d'^{-1/2}))$-list-decoding soundness.*

Proof. Let G_{disc} be the bipartite graph on $A \cup B$, where there are $w(a, b)$ multi-edges between a and b for each $(a, b) \in A \times B$. The dPCP $\mathcal{D}'$ is constructed as follows. For each vertex $b \in B$, let $\Gamma_{\text{disc}}(b)$ be the neighborhood of b in G_{disc}, where we view $\Gamma_{\text{disc}}(b)$ as a multiset. Specifically, each neighbor a is listed with multiplicity equal to the number of edges going from b to a, so that $|\Gamma_{\text{disc}}(b)| = d(b)$. Then, let $H_b = (A_b, B_b, E_b)$ be a d'-regular bipartite graph with $d(b)$ vertices on each side and second singular value $O(d'^{-1/2})$, and identify A_b with $\Gamma_{\text{disc}}(b) \subseteq A$. Recall that such a graph can be constructed in polynomial-time by Lemma 2.5. We note that after this identification, there may be multiple vertices in A_b that have been identified with the same vertex $a \in \Gamma_{\text{disc}}(b)$, so there may be multiple edges between two vertices.

45

The constraint graph of $\mathcal{D}'$ has left side A and right side $\cup_{b\in B} B_b$. We include an edge (a, b') if for the unique b such that $b' \in B_b$, it holds that $a \in \Gamma(b)$ and $(a, b') \in E_b$ (we recall that each vertex in A_b is identified with a vertex in A). The constraint on (a, b') in $\mathcal{D}'$ is the same as the constraint on (a, b) in $\mathcal{D}$, where $b' \in B_b$. The left and right alphabets of $\mathcal{D}'$ remain Σ_A, Σ_B. The decoders for $\mathcal{D}'$ also remain the same and are still $\{D_t\}_{t\in[n]}$. By this we mean, for a left vertex a and a label σ in its alphabet, the decoder for $\mathcal{D}'$ still outputs $D_t(a, \sigma)$, and hence we still denote it by D_t. Finally, the complete decoding distribution of $\mathcal{D}'$ is $\mathcal{Q}'$, which is generated by choosing $(t, a, b) \sim \mathcal{Q}$, sampling a uniform $b' \in B_b$ that is adjacent to a in H_b and outputting (t, a, b'). The decoding distributions are thus given by $\mathcal{P}'_t = \mathcal{Q}'(t, \cdot, \cdot)$.

It is clear that the algorithm preserves the left alphabet, right alphabet, projection decision complexity, decoding complexity, and decoding degree. Furthermore, one can check that the left degree of each vertex in the constraint graph gets multiplied by a factor of d'.

Complete Decoding Distribution. Fix $a \in A$ and $b' \in B'$ and suppose b' appears in B_b for $b \in B$. Let us first check that the complete decoding distribution is $d'M$-discrete. We have

$$\mathcal{Q}'(\circ, a, b') = \mathcal{Q}(\circ, a, b) \cdot \frac{c_b(a, b')}{d'} = \frac{w(a, b) \cdot c_b(a, b')}{d'M} \tag{32}$$

where $c_b(a, b')$ is the number of multiedges between a and b' in the graph H_b, and is an integer in the range $[0, d']$. To see the first equality, note that in order for (a, b') to be sampled from $\mathcal{Q}'$, we first need (a, b) to be sampled from $\mathcal{Q}$, where b is such that $b' \in H_b$. Then $\mathcal{Q}'$ chooses a neighbor of a in H_b, so there is a $c_b(a, b')/d'$ probability of choosing b'. The second transition is due to the assumption that $\mathcal{Q}(\circ, a, b) = \frac{w(a,b)}{M}$ given by the lemma statement. It follows that the complete decoding distribution is $d'M$-discrete

To see the second item, note that the probability that b' is output by the complete decoding distribution $\mathcal{Q}'$ is

$$\mathcal{Q}'(\circ, \circ, b') = \mathcal{Q}(\circ, \circ, b) \cdot \frac{1}{d(b)} = \left(\sum_{a\in A} \frac{w(a, b)}{M}\right) \frac{1}{d(b)} = \frac{1}{M}.$$

In the first transition we used the fact that conditioned on b, the right-vertex we output is uniform in B_b and $|B_b| = d(b)$. In the second transition we used $\mathcal{Q}(\cdot, a, b) = w(a, b)/M$. In the last transition we used the fact that $\sum_{a\in A} w(a, b) = d(b)$.

For the third item, fix $a \in A$, then

$$\mathcal{Q}'(\circ, a, \circ) = \sum_{b\in B}\sum_{b'\in B_b} \mathcal{Q}'(\circ, a, b') = \sum_{b\in B} \mathcal{Q}(\circ, a, b) \sum_{b'\in B_b} \frac{c_b(a, b')}{d'} = \sum_{b\in B} \mathcal{Q}(\circ, a, b) = \mathcal{Q}(\circ, a, \circ).$$

In the second transition we used (32) and in the third transition we used the fact that the degree of a in H_b is d'.

For the fourth item, suppose that $\mathcal{Q}$ is agnostic and take $b' \in B_b$. Note that $\mathcal{Q}(\circ, \cdot, b) = \mathcal{Q}'(\circ, \cdot, b')$, and for any $t \in [n]$ we have $\mathcal{Q}(t, \cdot, b) = \mathcal{Q}'(t, \cdot, b')$. It follows $\mathcal{Q}'$ is agnostic.

Perfect Completeness. Suppose $w \in \mathcal{L}$ and let $T_1 : A \to \Sigma_A$, $T_2 : B \to \Sigma_B$ be the assignments satisfying all of the constraints in $\mathcal{D}$ that are decoded to w with probability 1 over $\mathcal{Q}$. Define T'_2 by setting $T'_2[b'] = T_2[b]$ for each $b' \in B_b$. It is straightforward to check that T_1, T'_2 satisfy all of the constraints in $\mathcal{D}'$, and that T_1 decodes to w with probability 1 over $\mathcal{Q}'$.

List-Decoding Soundness. Fix an assignment $T_1 : A \to \Sigma_A$. Then by the (L, ε)-list-decoding soundness of $\mathcal{D}$ we may find $\{w_1, \ldots, w_L\} \subseteq \mathcal{L}$ such that for any $T_2 : B \to \Sigma_B$, we have

$$\Pr_{(t,a,b)\sim\mathcal{Q}}[\Phi_{(a,b)}(T_1[a], T_2[b]) = 1 \wedge D_t(a, T_1[a]) \notin \{(w_i)_t\}_{i\in[L]}] \leqslant \varepsilon.$$

To establish the desired list-decoding soundness for $\mathcal{D}'$, we will show that for any right-assignment, the list decoding error relative to the same list $\{w_1, \ldots, w_L\}$ is small.

Fix any assignment $T'_2 : B' \to \Sigma_B$ and for each $b \in B$ and $\sigma \in \Sigma_B$, define the sets

$$X_{b,\sigma} = \{b' \in B_b \mid T'_2[b'] = \sigma\} \quad \text{and} \quad Y_{b,\sigma} = \{a' \in A_b \mid \Phi_{(a',b)}(T_1[a'], \sigma) = 1\}.$$

In words, $X_{b,\sigma}$ is the set of vertices in B_b that are assigned σ under T_2', and $Y_{b,\sigma}$ is the set of vertices in $a' \in A_b$ such that the assignments $a \to T_1[a']$ and $b \to \sigma$ satisfy the constraint on (a', b) in the original dPCP, $\mathcal{D}$. It is clear that the sets $\{X_{b,\sigma}\}_{\sigma \in \Sigma_b}$ form a partition of B_b. Additionally, since the constraints in $\mathcal{D}$ are all projections, the sets $\{Y_{b,\sigma}\}_{\sigma \in \Sigma_B}$ form a partition of A_b. Define the functions $F_{b,\sigma} : A_b \to [0,1]$, $G_{b,\sigma} : B_b \to \{0,1\}$ where

$$G_{b,\sigma}(b') := \mathbb{1}_{b' \in X_{b,\sigma}} \quad \text{and} \quad F_{b,\sigma}(a) := \mathbb{1}_{a \in Y_{b,\sigma}} \cdot \Pr_{t \sim \mathcal{Q}'(\cdot, a, b)} \left[D_t(a, T_1[a]) \notin \{(w_i)_t\}_{i \in [L]} \right].$$

Abusing notation, we denote $\mu(F_{b,\sigma}) = \mathbb{E}_{a \in A_b}[F_{b,\sigma}(a')]$ and $\mu(G_{b,\sigma}) = \mathbb{E}_{b' \in B_b}[G_{b,\sigma}(b')]$. Consider the randomized assignment $T_2^* : B \to \Sigma_B$ obtained by setting $T_2^*(b) = \sigma$ with probability $\mu(G_{b,\sigma})$. By the list-decoding soundness applied with each fixing of T_2^*, we get

$$
\begin{aligned}
\varepsilon &\geqslant \mathbb{E}_{T_2^*} \left[\Pr_{(t,a,b) \sim \mathcal{Q}} \left[\Phi_{(a,b)}(T_1[a], T_2^*[b]) = 1 \wedge D_t(a, T_1[a]) \notin \{(w_i)_t\}_{i \in [L]} \right] \right] \\
&= \mathbb{E}_{(t,a,b) \sim \mathcal{Q}} \left[\Pr_{b' \in B_b} \left[\Phi_{a,b}(T_1[a], T_2[b']) = 1 \wedge D_t(a, T_1[a]) \notin \{(w_i)_t\}_{i \in [L]} \right] \right] \\
&= \mathbb{E}_{b \sim \mathcal{Q}(\circ, \circ, \cdot)} \left[\sum_{\sigma \in \Sigma_B} \mu(G_{b,\sigma}) \cdot \mu(F_{b,\sigma}) \right],
\end{aligned}
\tag{33}
$$

In the third transition, we are using the fact that conditioned on b, the distribution of a under $\mathcal{Q}$ is uniformly random in A_b. On the other hand, the list-decoding error of the assignment T_1, T_2' in $\mathcal{D}'$ relative to the list $\{w_1, \ldots, w_L\}$, can be written as:

$$
\mathbb{E}_{b \sim \mathcal{Q}(\circ, \circ, \cdot)} \left[\sum_{\sigma \in \Sigma_B} \mathbb{E}_{(a,b') \in E_b} [F_{b,\sigma}(a) \cdot G_{b,\sigma}(b')] \right].
\tag{34}
$$

Hence, our goal is to now bound the expression above, and we will do so by relating it to (33) via the expander mixing lemma. Fix a vertex $b \in B$ and an alphabet symbol $\sigma \in \Sigma_B$. Using Lemma 2.3 we get

$$
\mathbb{E}_{(a,b') \in E_b} [F_{b,\sigma}(a) \cdot G_{b,\sigma}(b')] \leqslant \mu(F_{b,\sigma}) \cdot \mu(G_{b,\sigma}) + O(d'^{-1/2}) \cdot \sqrt{\mu(F_{b,\sigma}) \cdot \mu(G_{b,\sigma})}.
$$

Keeping $b \in B$ fixed and summing this inequality over all $\sigma \in \Sigma_B$, we get

$$
\begin{aligned}
\sum_{\sigma \in \Sigma_B} \mathbb{E}_{(a,b') \in E_b} [F_{b,\sigma}(a) \cdot G_{b,\sigma}(b')] &\leqslant \sum_{\sigma \in \Sigma_B} \mu(F_{b,\sigma}) \cdot \mu(G_{b,\sigma}) + O(d'^{-1/2}) \cdot \sqrt{\mu(F_{b,\sigma}) \cdot \mu(G_{b,\sigma})} \\
&\leqslant \sum_{\sigma \in \Sigma_B} \mu(F_{b,\sigma}) \cdot \mu(G_{b,\sigma}) + O(d'^{-1/2}),
\end{aligned}
$$

where the second inequality uses Cauchy-Schwarz and the fact that $\sum_{\sigma \in \Sigma_B} \mu(F_{b,\sigma}) \leqslant 1$ and $\sum_{\sigma \in \Sigma_B} \mu(G_{b,\sigma}) = 1$. Thus, we have

$$
\mathbb{E}_{b \sim \mathcal{Q}(\circ, \circ, \cdot)} \left[\sum_{\sigma \in \Sigma_B} \mathbb{E}_{(a,b') \in E_b} [F_{b,\sigma}(a) \cdot G_{b,\sigma}(b')] \right] \leqslant \mathbb{E}_{b \sim \mathcal{Q}(\circ, \circ, \cdot)} \left[\sum_{\sigma \in \Sigma_B} \mu(G_{b,\sigma}) \cdot \mu(F_{b,\sigma}) \right] + O(d'^{-1/2}) \leqslant \varepsilon + O(d'^{-1/2}),
$$

where in the last transition we used (33). $\qquad \square$

Lemma 7.2 gives a right degree reduction when the complete decoding distribution is M-discrete and results in a dPCP of size at least M. For dPCPs with arbitrary complete decoding distribution, which may not be M-discrete for any small M, we can still obtain a right degree reduction by slightly altering the complete decoding distribution first. Below we state a simple fact about distributions that we will use for this task, which will also be used in subsequent sections.

Lemma 7.3. *Let Δ be a distribution over a support of size n, say $[n]$. Then for every $\varepsilon > 0$ and every integer $M \geqslant \frac{n}{2\varepsilon}$, there exist $w(1), \ldots, w(n) \in \mathbb{N}$ such that $\sum_{i \in [n]} w(i) = M$ and the distribution Δ_{disc}, given by $\Delta_{\mathsf{disc}}(i) = \frac{w(i)}{M}$, satisfies*

- $\mathsf{TV}(\Delta, \Delta_{\mathsf{disc}}) \leqslant \varepsilon$,

- *for every $i \in [n]$, $|\Delta_{\mathsf{disc}}(i) - \Delta(i)| \leqslant \frac{1}{M}$,*

- $\mathsf{supp}(\Delta_{\mathsf{disc}}) \subseteq \mathsf{supp}(\Delta)$.

As a consequence, every distribution with support size n is ε-close in total-variation distance to an $O_\varepsilon(n)$-discrete distribution.

Proof. Suppose $\mathsf{supp}(\Delta)$ has size $n' \leqslant n$, and without loss of generality suppose $\mathsf{supp}(\Delta) = [n']$. For each $i \in [n']$, let $z_i = \lfloor M \cdot \Delta(i) \rfloor$ and let $r = M - \sum_{i=1}^{n'} z_i$. Note that $r \leqslant n'$. Now increment the each of $z_1, \ldots, z_r$ by 1 and keep the remainder of the z_i's the same. Let the resulting values be z_i' after incrementing.

We have

$$|z_i' - M \cdot \Delta(i)| \leqslant 1, \tag{35}$$

for all $i \in [n']$. Now define the distribution $\Delta_{\mathsf{disc}}(i)$ by

$$\Delta_{\mathsf{disc}}(i) = \begin{cases} \frac{z_i'}{M} & \text{if } i \in [n'], \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\Delta_{\mathsf{disc}}(i)$ is indeed a distribution over $[n]$ because $\sum_{i \in [n]} z_i' = M$ and by construction $\mathsf{supp}(\Delta_{\mathsf{disc}}) \subseteq \mathsf{supp}(\Delta)$. Moreover, (35) implies that for every $i \in [n]$, $|\Delta_{\mathsf{disc}}(i) - \Delta(i)| \leqslant \frac{1}{M}$, and as a consequence,

$$\mathsf{TV}(\Delta, \Delta_{\mathsf{disc}}) = \frac{1}{2} \sum_{i \in [n]} |\Delta(i) - \Delta_{\mathsf{disc}}(i)| \leqslant \sum_{i \in [n']} \frac{1}{2M} \leqslant \frac{n}{2M} \leqslant \varepsilon. \qquad \square$$

The following result is a version of Lemma 7.2 in which the given dPCP $\mathcal{D}$ does not necessarily have a complete decoding distribution whose marginal over the edges is M-discrete . Instead, we require $\mathcal{D}$ to have bounded decoding degree, and we show how to modify the complete decoding distribution into a discrete one without affecting the rest of the parameters by much. For sake of simplicity and because it suffices for our application, we focus on the case where decoding degree is 1.

Lemma 7.4. *Suppose a language $\mathcal{L} \subseteq \Sigma_0^n$ has a projection dPCP*

$$\mathcal{D} = \left(A \cup B, E, \Sigma_A, \Sigma_B, \{D_t\}_{t \in [n]}, \{\mathcal{P}_t\}_{t \in [n]} \right)$$

with (L, ε)-list-decoding soundness, left degree ld, and decoding degree 1. Then for every $M \geqslant \frac{200 \cdot \mathsf{ld} \cdot |A|}{\varepsilon^2 \cdot n}$ and for every $d' \in \mathbb{N}$, there is a polynomial-time algorithm which takes $\mathcal{D}$ and outputs a dPCP

$$\mathcal{D}'' = \left(A'' \cup B'', E'', \Sigma_A, \Sigma_B, \{D_t\}_{t \in [n]}, \{\mathcal{P}_t''\}_{t \in [n]} \right)$$

for $\mathcal{L}$ such that the following hold:

- **Length.** *$|B''| = d' \cdot n \cdot M$ and $A'' \subseteq A$, so the length is at most $|A| + d' \cdot n \cdot M$.*

- **Degrees.** *The decoding degree of every vertex is 1.*

- **Projection decision complexity.** *The projection decision complexity is preserved.*

- **Decoding complexity.** *The decoding complexity is preserved.*

- **Complete decoding distribution.** *The complete decoding distribution, which we call* $\mathcal{Q}''$, *is* $d'^2 \cdot n \cdot M$*-discrete and its marginal over* B' *is uniform. For any* $b \in B$, *we have*

$$\mathcal{Q}''(\circ, \circ, b) = \frac{d'}{d'^2 \cdot n \cdot M}$$

  and for any $a \in A$ *we have*

$$\mathcal{Q}''(\circ, a, \circ) \leqslant \max\left\{\frac{1}{nM} + \frac{\varepsilon}{10\mathsf{ld}|A|}, \mathcal{Q}(\circ, a, \circ) + \frac{\mathsf{ld}}{nM}\right\}.$$

- **Completeness.** *The dPCP* $\mathcal{D}'$ *has perfect completeness.*

- **List-decoding soundness.** $\left(L, \frac{9}{8} \cdot \varepsilon + O(d'^{-1/2})\right)$*-list-decoding soundness.*

Proof. Let $\mathcal{Q}$ be the complete decoding distribution of $\mathcal{D}$ and let ld, ε and M be as in the lemma statement. For each $a \in A$, we have that the decoding degree of a is 1, and we denote by t_a its unique decoding neighbor. Since the decoding degree is 1, we have

$$\sum_{t \in [n]} |\operatorname{supp}(\mathcal{Q}(t, \cdot, \cdot))| \leqslant \sum_{a \in A} |\operatorname{supp}(\mathcal{Q}(t_a, a, \cdot))| \leqslant \mathsf{ld} \cdot |A|.$$

Hence, taking $S \subseteq [n]$ to be the set of $t \in [n]$ such that $|\operatorname{supp}(\mathcal{Q}(t, \cdot, \cdot))| \leqslant \frac{10 \cdot \mathsf{ld} \cdot |A|}{\varepsilon \cdot n}$, we have by Markov's inequality that

$$\Pr_{t \in [n]}[t \in S] \geqslant 1 - \frac{\varepsilon}{10}. \tag{36}$$

Now, for each $t \in S$, let Δ_t be the discretized distribution over $A \times B$ obtained by applying Lemma 7.3 to $\mathcal{Q}(t, \cdot, \cdot)$ with denominator M. For $t \notin S$, let Δ_t be an arbitrary distribution supported on $\operatorname{supp}(\mathcal{Q}(t, \cdot, \cdot))$ such that for each $(a, b) \in \operatorname{supp}(\mathcal{Q}(t, \cdot, \cdot))$ we have either

$$\Delta_t(a, b) = \frac{1}{M} \cdot \left\lceil \frac{M}{|\operatorname{supp}(\mathcal{Q}(t, \cdot, \cdot))|} \right\rceil \quad \text{or} \quad \Delta_t(a, b) = \frac{1}{M} \cdot \left\lfloor \frac{M}{|\operatorname{supp}(\mathcal{Q}(t, \cdot, \cdot))|} \right\rfloor.$$

It is straightforward to see that such a distribution indeed exists.

Now, let the distribution Q' be generated by choosing $t \in [n]$ uniformly at random, choosing $(a, b) \sim \Delta_t$, and outputting (t, a, b). We note the following items:

- The marginal over $[n]$ of $\mathcal{Q}'$ is uniform; this is clear by description of $\mathcal{Q}'$.

- $\operatorname{supp}(\mathcal{Q}') \subseteq \operatorname{supp}(\mathcal{Q})$; this is clear by construction of Δ_t.

- For each $(a, b) \in A \times B$, if $t_a \in S$ then $|\mathcal{Q}(\circ, a, b) - \mathcal{Q}'(\circ, a, b)| \leqslant \frac{1}{nM}$, otherwise $\mathcal{Q}'(\circ, a, b) \leqslant \frac{1}{nM} + \frac{\varepsilon}{10\mathsf{ld}|A|}$; we explain this below.

- $\mathsf{TV}(\mathcal{Q}, \mathcal{Q}') \leqslant \frac{\varepsilon}{8}$; we explain this below.

To see the third item first fix an $(a, b) \in A \times B$. If $t_a \in S$, then $\mathcal{Q}(\circ, a, b) = \mathcal{Q}(t_a, a, b)$, $\mathcal{Q}'(\circ, a, b) = \Delta_{t_a}(a, b)$ and the item follows from the guarantee of Lemma 7.3. If $t_a \notin S$ we have,

$$\mathcal{Q}'(\circ, a, b) \leqslant \frac{1}{n} \cdot \Delta_{t_a}(a, b) \leqslant \frac{1}{n} \cdot \frac{1 + M/|\operatorname{supp}(\mathcal{Q}(t, \cdot, \cdot))|}{M} \leqslant \frac{1}{nM} + \frac{\varepsilon}{10\mathsf{ld}|A|}.$$

For the fourth item, we use the fact that $\mathcal{Q}$ and $\mathcal{Q}'$ have the same marginal distribution over $[n]$, hence

$$\begin{aligned}
\mathsf{TV}(\mathcal{Q}, \mathcal{Q}') &= \sum_{t \in [n]} \mathcal{Q}(t, \circ, \circ) \cdot \mathsf{TV}(\mathcal{Q}(t, \cdot, \cdot), \mathcal{Q}'(t, \cdot, \cdot)) \\
&\leqslant \frac{\varepsilon}{10} + \sum_{t \in S} \mathcal{Q}(t, \circ, \circ) \cdot \mathsf{TV}(\mathcal{Q}(t, \cdot, \cdot), \mathcal{Q}'(t, \cdot, \cdot)) \\
&\leqslant \frac{\varepsilon}{8}.
\end{aligned}$$

The first transition uses (36) and the second transition uses the fact that

$$\mathsf{TV}(\mathcal{Q}(t,\cdot,\cdot),\mathcal{Q}'(t,\cdot,\cdot)) = \mathsf{TV}(\mathcal{Q}(t,\cdot,\cdot),\Delta_t) \leqslant \frac{\varepsilon}{40}$$

for all $t \in S$ by Lemma 7.3.

Let $\mathcal{D}'$ be the dPCP obtained by taking $\mathcal{D}$ and changing its complete decoding distribution to $\mathcal{D}'$. Note that the length, projection decision complexity, and decoding complexity are unchanged. For the decoding degrees, note that each a has decoding degree either 0 or 1 now. Completeness is preserved by the fact that $\mathrm{supp}(\mathcal{Q}') \subseteq \mathrm{supp}(\mathcal{Q})$, stated above, and $\mathcal{D}'$ still has $(L, 9\varepsilon/8)$-list decoding soundness by the fact $\mathsf{TV}(\mathcal{Q}, \mathcal{Q}') \leqslant \varepsilon/8$, stated above.

Let us now write $\mathcal{Q}'(\circ, a, b) = \frac{w(a,b)}{n \cdot M} = \frac{d' w(a,b)}{d' \cdot n \cdot M}$ for $w(a, b) \in \mathbb{N}$. In expressing $\mathcal{Q}'$ this way, we are using the fact that a has decoding degree 1. Here, we multiplied both the numerator and denominator by d' to ensure that all of the nonzero numerators are at least d', in accordance with Lemma 7.2.

We now apply Lemma 7.2 on $\mathcal{D}'$ where its complete decoding distribution is written with denominator $d'nM$, and the weight of each edge (a, b) is $d'w(a, b)$. Call the resulting dPCP $\mathcal{D}''$ and let $\mathcal{Q}''$ be its complete decoding distribution. The lemma follows by checking the guarantees from Lemma 7.2. In particular, the length, projection decision complexity, decoding complexity, perfect completeness, and list-decoding soundness are immediate.

For the guarantee on the complete decoding distribution, we first note that by Lemma 7.2, $\mathcal{Q}''$ has uniform marginal over the right side and is d'^2nM discrete. This also implies that

$$\mathcal{Q}''(\circ, \circ, b) = \frac{d'}{d'^2 \cdot n \cdot M}.$$

For the remaining guarantee on the complete decoding distribution, fix a left vertex a. We start by using the third part of the decoding distribution guarantee from Lemma 7.2 to get:

$$\mathcal{Q}''(\circ, a, \circ) = \mathcal{Q}'(\circ, a, \circ) = \sum_{b \in B} \mathcal{Q}'(\circ, a, b).$$

If $t_a \notin S$, then $\mathcal{Q}''(\circ, a, \circ) = \mathcal{Q}'(\circ, a, \circ) \leqslant \frac{1}{nM} + \frac{\varepsilon}{10\mathsf{ld}|A|}$ by the third item above and we are done. Otherwise,

$$|\mathcal{Q}(\circ, a, \circ) - \mathcal{Q}''(\circ, a, \circ)| = \left| \sum_{b \in B} (\mathcal{Q}(\circ, a, b) - \mathcal{Q}'(\circ, a, b)) \right| \leqslant \frac{\mathsf{ld}}{nM},$$

where in the last transition we use the fact that the summation has ld many nonzero terms and each is at most $\frac{1}{nM}$ by the third item listed above. The above implies,

$$\mathcal{Q}''(\circ, a, \circ) \leqslant \mathcal{Q}(\circ, a, \circ) + \frac{\mathsf{ld}}{nM}.$$

Finally, to get decoding degree 1, note that each vertex $a \in A$ has decoding degree either 1 or 0, and we can delete every vertex with decoding degree 0 without affecting any of the previous guarantees. $\square$

8 Decoding Degree Reduction

In this section we show how to reduce the decoding degree of a dPCP assuming that its complete decoding distribution is agnostic. After performing this operation once, we will preserve constant decoding degree throughout the remainder of our construction. At a high level, after Theorem 6.1 the decoding degree is $\mathrm{poly}(\log(N))$, and hence holding this many Σ_0-symbols itself already causes superpolynomial alphabet size. Therefore any method of reducing alphabet size to a constant must first reduce decoding degree to a constant. This is a non-issue in standard PCPs (as there is no decoding going on), and partly the reason earlier composition techniques did not apply to dPCPs.

The statement of our decoding degree transformation is as follows.

Lemma 8.1. *Suppose Σ_0 is an alphabet and that the language $\mathcal{L} \subseteq \Sigma_0^n$ has a projection dPCP*

$$\mathcal{D} = (A \cup B, E, \{\mathcal{P}_t\}_{t \in [n]}, \Sigma_A, \Sigma_B, \{\Phi_e\}_{e \in E}, \{D_t\}_{t \in [n]})$$

with perfect completeness, (L, ε)-list-decoding soundness for $L := \mathsf{poly}(1/\varepsilon)$, agnostic complete decoding distribution, projection decision complexity $\mathsf{ProjComp}$, decoding complexity $\mathsf{DecodeComp}$, and left degree, decoding degree, right degree that are at most $\mathsf{ld}, \mathsf{ldd}, d$ respectively. Let $\mathcal{I} = \{(t, b) \mid \exists a \in A, (a, b) \in \mathrm{supp}(\mathcal{P}_t)\}$, and let $\{C_{t,b} \in \mathbb{N} \mid (t, b) \in \mathcal{I}\}$ be a collection of integers. Then $\mathcal{L}$ has a projection dPCP

$$\mathcal{D}' = (B' \cup A, \Sigma_A^d \times \Sigma_0, \Sigma_A, E', \{\mathcal{P}'_t\}_{t \in [n]}, \{\Phi'_e\}_{e \in E'}, \{D'_t\}_{t \in [n]}),$$

satisfying the following properties:

- **Length.** *$B' = \bigcup_{(t,b) \in \mathcal{I}} \{(t, b, i) \mid i \in [C_{t,b}]\}$, and in particular the length is $|A| + \sum_{(t,b) \in \mathcal{I}} C_{t,b}$.*

- **Degrees.** *The left degree of every vertex is at most d, the right degree of every vertex is at most $\mathsf{ld} \cdot \mathsf{ldd} \cdot \max_{(t,b) \in \mathcal{I}} C_{t,b}$, and the decoding degree of every vertex is 1. If $\mathcal{D}$ is d-right regular, then $\mathcal{D}'$ is d-left regular.*

- **Projection Decision Complexity.** *$O(d \cdot (\mathsf{ProjComp} + \mathsf{DecodeComp}))$.*

- **Decoding Complexity.** *$O(d \log(|\Sigma_A|) + \log(|\Sigma_0|))$.*

- **Decoding Distributions.** *For each $(t, b, i) \in B'$ and $a \in A$, t is the unique index in the decoding neighborhood of (t, b, i) and we have*

$$\mathcal{P}'_t((t, b, i), \circ) = \frac{\mathcal{P}_t(\circ, b)}{C_{t,b}} \quad and \quad \mathcal{Q}'(t, (t, b, i), a) = \frac{\mathcal{Q}(t, b, a)}{C_{t,b}},$$

  where $\mathcal{Q}$ and $\mathcal{Q}'$ are the complete decoding distributions of $\mathcal{D}$ and $\mathcal{D}'$ respectively.

- **Perfect Completeness.** *$\mathcal{D}'$ has perfect completeness.*

- **List-Decoding Soundness.** *$\mathcal{D}'$ has $(L \cdot \mathsf{poly}(1/\varepsilon), 2\varepsilon^{1/5})$-list-decoding soundness.*

Proof. Let

$$\mathcal{D} = \left(A \cup B, \Sigma_A, \Sigma_B, E, \{\Phi_e\}_{e \in E}, \{D_t\}_{t \in [n]}, \{\mathcal{P}_t\}_{t \in [n]}\right)$$

be a dPCP for the language $\mathcal{L}$ as in the premise of the lemma. Denote its complete decoding distribution by $\mathcal{Q}$, namely a sample is obtained by choosing $t \in [n]$ uniformly, $(a, b) \sim \mathcal{P}_t$, and outputting (t, a, b).

We denote the new dPCP for $\mathcal{L}$ by

$$\mathcal{D}' = \left(B' \cup A, \Sigma_A^d \times \Sigma_0, \Sigma_A, E', \{\Phi'_e\}_{e \in E'}, \{D'_t\}_{t \in [n]}, \{\mathcal{P}'_t\}_{t \in [n]}\right),$$

and define its components in the following way. Below, we use Γ and Γ_{dec} to denote the neighborhood and decoding neighborhood functions in $\mathcal{D}$ respectively, and use Γ' and Γ'_{dec} to denote the neighborhood and decoding neighborhood functions in $\mathcal{D}'$ respectively.

- **Constraint Graph.** The constraint graph is as follows:
    - The left side $B' = \bigcup_{(t,b) \in \mathcal{I}} \{(t, b, i) \mid i \in [C_{t,b}]\}$.
    - The right side is A.
    - The edge set E' consist of all $((t, b, i), a)$ such that $(a, b) \in E$.

- **Alphabets.** The alphabets are as follows.

- The left alphabet is $\Sigma_{B'} \subseteq \Sigma_A^d \times \Sigma_0$. For a fixed $(t, b, i) \in B'$, we think of a label to it as assigning a Σ_A label for each $a \in \Gamma(b)$ and a Σ_0 label for t. Furthermore, we hardcode constraints on the alphabet of $(t, b, i) \in B'$ as follows. Writing $\Gamma(b) = \{a_1, \ldots, a_d\}$, for a label $\sigma \in \Sigma_A^d \times \Sigma_0$ to (t, b, i) we denote by σ_{a_j} be the label assigned to a_j in σ, and by σ_t the label for t in σ. We denote by $\sigma_{a_j}^{(b)}$ the unique value in Σ_B satisfying $\Phi_{(a_j, b)}(\sigma_{a_j}, \sigma_{a_j}^{(b)}) = 1$ (recall that there is a unique such element in Σ_B since the constraints in $\mathcal{D}$ are projections). Then, we require that $D_t(a_j, \sigma_{a_j}) = \sigma_t$ for all $i \in [d]$, and that $\sigma_{a_j}^{(b)}$ is the same for all $j \in [d]$.

  - The right alphabet is Σ_A.

- **Constraints.** For each edge $((t, b, i), a) \in E'$, the constraint $\Phi'_{((t,b,i),a)} : \Sigma_{B'} \times \Sigma_A \to \{0, 1\}$ is defined as follows. Given labels $\sigma \in \Sigma_B$ and $\sigma' \in \Sigma_A$, let σ_a be the label for a in σ. The constraint $\Phi'_{((t,b,i),a)}(\sigma, \sigma')$ is satisfied if and only if $\sigma_a = \sigma'$.

- **Decoders.** For each $t \in [n]$, the decoder $D'_t : B' \times \Sigma_B \to \Sigma_0$ is defined as follows. For $(t, b, i) \in B'$ and $\sigma \in \Sigma_{B'}$ label for it, the decoder $D'_t((t, b, i), \sigma)$ outputs σ_t, the label for t in σ.

  We note that by the constraints on the left alphabet symbol in $\mathcal{D}$, for any $a \in \Gamma(b)$, letting σ_a be the label for a in σ, we have that $D_t(a, \sigma_a) = \sigma_t = D'_t((t, b, i), \sigma)$.

- **Decoding Distributions.** For each $t \in [n]$, the decoding distribution $\mathcal{P}'_t$ is defined as follows. To sample from $\mathcal{P}'_t$, sample $(a, b) \sim \mathcal{P}_t$, $i \in [C_{t,b}]$ uniformly, and then output $((t, b, i), a)$. Since we think of the side-decoding distribution as uniform over $[n]$, the complete decoding distribution, denoted $\mathcal{Q}'$ is obtained by sampling $t \in [n]$ uniformly and then $((t, b, i), a) \sim \mathcal{P}'_t$. We remark that the complete decoding distribution $\mathcal{Q}'$ can also be obtained by choosing $(t, a, b) \sim \mathcal{Q}$ and then choosing $i \in [C_{t,b}]$ uniformly and outputting $(t, (t, b, i), a)$.

Let us now analyze the parameters of the new dPCP $\mathcal{D}'$. As usual, the main difficulty is in analyzing the list-decoding soundness.

Length. It is clear that $|B'| = \sum_{(t,b) \in \mathcal{I}} C_{t,b}$ and hence the length is as described.

Degrees. It is clear that the degree of every vertex in B' is at most d, as for each $(t, b, i) \in B'$, $\Gamma'((t, b, i)) = \Gamma(b)$, and in the case that $\mathcal{D}$ is right-regular, it is clear that $\mathcal{D}'$ is left-regular. As for the degrees of each $a \in A$, it is clear that its degree is at most $\mathsf{ld} \cdot \mathsf{ldd} \cdot \max_{(t,b) \in \mathcal{I}} C_{t,b}$ as desired. Finally, the decoding degree of every vertex in B' is 1, as one can see that each $(t, b, i) \in B'$ is only responsible for decoding t.

Projection Decision Complexity. Fix a constraint $((t, b, i), a)$ and let σ be the symbols assigned to the (t, b, i). The constraint must check that σ is a valid alphabet symbol. After that, the unique symbol to a that would satisfy the constraint can be found by taking a restriction in σ, and this has circuit complexity at most $\mathsf{ProjComp}$. To check that σ is valid, the projection circuit must compute one projection from $\mathcal{D}$ and one decoding for each neighbor of b's d neighbors and perform d equality checks. Overall, this shows the projection decision complexity is $O(d \cdot (\mathsf{ProjComp} + \mathsf{DecodeComp}))$.

Decoding Complexity. The decoder outputs the restriction to a single symbol from an alphabet of size $|\Sigma_A|^d \cdot |\Sigma_0|$ and hence the decoding complexity is $O(d \log |\Sigma_A| + \log |\Sigma_0|)$.

Perfect Completeness. Fix any $w \in \mathcal{L}$. As $\mathcal{D}$ has perfect completeness, there are assignments $T_1 : A \to \Sigma_A$ and $T_2 : B \to \Sigma_B$ which satisfy all of the constraints in $\mathcal{D}$ and decode to w_t with probability 1 under $\mathcal{P}_t$ for every $t \in [n]$. Then we construct assignments $T'_1 : B' \to \Sigma_A^d \times \Sigma_0$ and $T'_2 : A \to \Sigma_A$ in the natural way as follows. Set $T'_2 = T_1$. For each $(t, b, i) \in B'$ write $\Gamma(b) = \{a_1, \ldots, a_d\}$, and take $T'_1[(t, b, i)] = (T_1[a_1], \ldots, T_1[a_d], D_t(a_1, T_1[a_1]))$. In words, this is the symbol in $\Sigma_A^d \times \Sigma_0$ which contains $T_1[a_i]$ as the label for a_i for all $i \in [d]$, and $D_t(a_i, T_1[a_i])$ as the label for t (we note that by the hardcoded constraints it does not matter which a_i we pick). Note that by the fact that T_1 and T_2 satisfy all of the constraints in $\mathcal{D}$ and always decode to w_t under $\mathcal{P}_t$, this assignment indeed satisfies the constraints on the left alphabet and is a valid assignment. It is straightforward to verify that these assignments satisfy all of the constraints in $\mathcal{D}'$ and decode to w_t with probability 1 under $\mathcal{P}'_t$ for every $t \in [n]$, so $\mathcal{D}'$ has perfect completeness.

List-Decoding Soundness. Fix any left assignment $T'_1 : B' \to \Sigma_A^d \times \Sigma_0$. Based on T'_1, we construct a candidate list of words from $\mathcal{L}$ and show that it satisfies the list-decoding soundness. Recall that for each $(t, b, i) \in B'$ and $a \in A$ adjacent to it in $\mathcal{D}'$, the assignment $T'_1[(t, b, i)]$ contains a symbol for a, which we denote by $T'_1[(t, b, i)]_a$. For each $a \in A$, let $\mathsf{cand}(a)$ consist of all symbols $\sigma \in \Sigma_A$ such that

$$\Pr_{(t,(t,b,i))\sim\mathcal{Q}'(\cdot,a,\cdot)}[T'_1[(t,b,i)]_a = \sigma] = \Pr_{(t,b)\sim\mathcal{Q}(\cdot,a,\cdot),i\in[C_{t,b}]}[T'_1[(t,b,i)]_a = \sigma] \geqslant \varepsilon^{1/5}.$$

where the first equality is by the observation regarding how to generate $\mathcal{Q}'$ from $\mathcal{Q}$ made in the definition of $\mathcal{D}'$ above.

Clearly, $|\mathsf{cand}(a)| \leqslant \varepsilon^{-1/5} := J$ for every $a \in A$. Therefore, one can find assignments $T_{1,1}, \ldots, T_{1,J} : A \to \Sigma_A$ such that for every $a \in A$ and every $\sigma \in \mathsf{cand}(a)$ there is some $j \in [J]$ such that $T_{1,j}[a] = \sigma$. Since each $T_{1,j}$ is a left assignment to $\mathcal{D}$, we can apply the list-decoding soundness of $\mathcal{D}$ to it. Specifically, for each $j \in [J]$, let $\mathsf{List}_j = \{w_{j,1}, \ldots, w_{j,L}\} \subseteq \mathcal{L}$ be the list guaranteed by the (L, ε)-list-decoding soundness of $\mathcal{D}$ for the left assignment $T_{1,j}$. Then by the list-decoding soundness of $\mathcal{D}$ we conclude that for any right assignment $T_2 : B \to \Sigma_B$ and any $j \in [J]$, we have

$$\Pr_{(t,a,b)\sim\mathcal{Q}}[\Phi_{(a,b)}(T_{1,j}[a], T_2[b]) = 1 \wedge D_t(a, T_{1,j}[a]) \notin \{w_{j,1}[t], \ldots, w_{j,L}[t]\}] \leqslant \varepsilon. \tag{37}$$

We show that $\mathsf{List} = \bigcup_{j\in[J]} \mathsf{List}_j$ satisfies the list-decoding soundness condition of $\mathcal{D}'$ with size $J \cdot L = L \cdot \mathsf{poly}(1/\varepsilon)$ and error $2\varepsilon^{1/5}$. The size is clear, and next we show that for any right assignment $T'_2 : A \to \Sigma_A$, the list-decoding error of T'_1, T'_2 relative to List is small. We define the following events:

- Let $\mathcal{E}_{(t,b,i),a}$ be the event that the constraint on $((t, b, i), a)$ is satisfied by assignments $T'_1[(t, b, i)]$ and $T'_2[a]$.

- Let $\mathcal{F}_{j,(t,b,i),a}$ be the event that $D'_t((t, b, i), T'_1[(t, b, i)]) \notin \{(w_{j,1})_t, \ldots, (w_{j,L})_t\}$.

It is clear that the list-decoding error can be expressed and bounded as follows

$$
\begin{aligned}
&\Pr_{(t,(t,b,i),a)\sim\mathcal{Q}'} \left[\mathcal{E}_{(t,b,i),a} \bigwedge \left(\bigwedge_{j\in[J]} \mathcal{F}_{j,(t,b,i),a}\right)\right] \\
&= \Pr_{(t,a,b)\sim\mathcal{Q},i\in[C_{t,b}]} \left[\mathcal{E}_{(t,b,i),a} \bigwedge \left(\bigwedge_{j\in[J]} \mathcal{F}_{j,(t,b,i),a}\right)\right] \\
&= \Pr_{(t,a,b)\sim\mathcal{Q},i\in[C_{t,b}]} \left[\mathcal{E}_{(t,b,i),a} \bigwedge \left(\bigwedge_{j\in[J]} \mathcal{F}_{j,(t,b,i),a}\right) \wedge (T'_1[(t,b,i)]_a \in \mathsf{cand}(a))\right] \\
&\quad + \Pr_{(t,a,b)\sim\mathcal{Q},i\in[C_{t,b}]} \left[\mathcal{E}_{(t,b,i),a} \bigwedge \left(\bigwedge_{j\in[J]} \mathcal{F}_{j,(t,b,i),a}\right) \wedge (T'_1[(t,b,i)]_a \notin \mathsf{cand}(a))\right] \\
&\leqslant \Pr_{(t,a,b)\sim\mathcal{Q},i\in[C_{t,b}]} \left[\mathcal{E}_{(t,b,i),a} \bigwedge \left(\bigwedge_{j\in[J]} \mathcal{F}_{j,(t,b,i),a}\right) \wedge (T'_1[(t,b,i)]_a \in \mathsf{cand}(a))\right] \\
&\quad + \Pr_{(t,a,b)\sim\mathcal{Q},i\in[C_{t,b}]} \left[\mathcal{E}_{(t,b,i),a} \wedge (T'_1[(t,b,i)]_a \notin \mathsf{cand}(a))\right] \\
&\leqslant \sum_{j\in[J]} \Pr_{(t,a,b)\sim\mathcal{Q},i\in[C_{t,b}]} \left[\mathcal{E}_{(t,b,i),a} \wedge \mathcal{F}_{j,(t,b,i),a} \wedge (T'_1[(t,b,i)]_a = T_{1,j}[a])\right] \\
&\quad + \Pr_{(t,a,b)\sim\mathcal{Q},i\in[C_{t,b}]} \left[\mathcal{E}_{(t,b,i),a} \wedge (T'_1[(t,b,i)]_a \notin \mathsf{cand}(a))\right]
\end{aligned} \tag{38}
$$

In the last transition we used the union bound, and we next bound the terms on the right hand side. We upper bound

the second term on the right hand side of (38) as

$$\Pr_{(t,a,b)\sim\mathcal{Q},i\in[C_{t,b}]}\left[\mathcal{E}_{(t,b,i),a}\wedge(T_1'[(t,b,i)]_a\notin\mathsf{cand}(a))\right]$$

$$\leqslant\mathbb{E}_a\left[1_{T_2'[a]\notin\mathsf{cand}(a)}\Pr_{(t,b)\sim\mathcal{Q}(\cdot,a,\cdot),i\in[C_{t,b}]}[T_1'[(t,b,i)]_a=T_2'[a]]\right] \quad (39)$$

$$\leqslant\varepsilon^{1/5},$$

where in the last inequality we used the definition of $\mathsf{cand}(a)$.

We now go on to bound the first term on the right hand side of (38). We bound the summand corresponding to each $j\in[J]$ separately, and we fix a j henceforth. Define

- $\mathcal{E}'_{j,(t,b,i),a}$ be the event $\mathcal{E}_{(t,b,i),a}\wedge\mathcal{F}_{j,(t,b,i),a}\wedge(T_{1,j}[a]=T_1'[(t,b,i)]_a)$,

so our task is now to upper bound $\Pr_{(t,a,b)\sim\mathcal{Q},i\in[C_{t,b}]}[\mathcal{E}'_{j,(t,b,i),a}]$. For any constraint of $\mathcal{D}$, say $(a,b)\in E$, and label σ for a, we denote by $\sigma_{\to b}$ the unique label for b such that $\Phi_{(a,b)}(\sigma,\sigma_{\to b})=1$. Here, the uniqueness follows since $\mathcal{D}$ is a projection dPCP. For each $(t,b,i)\in B'$, let $T_1'[(t,b,i)]_{\to b}$ be the label deduced for b in $T_1'[(t,b,i)]$. In particular, by the hardcoded alphabet constraints we have that $T_1'[(t,b,i)]_{\to b}=(T_1'[(t,b,i)]_a)_{\to b}$ for any $a\in\Gamma(b)$.

We say a symbol $\sigma\in\Sigma_B$ is b-heavy if

$$\Pr_{(t,a)\sim\mathcal{Q}(\cdot,\cdot,b)}[T_{1,j}[a]_{\to b}=\sigma]\geqslant\sqrt{\varepsilon}. \quad (40)$$

It is clear that for every $b\in B$, there are at most $J'=1/\sqrt{\varepsilon}$ many b-heavy symbols, hence we can find tables $H_1,\ldots,H_{J'}:B\to\Sigma_B$ such that for every b-heavy symbol σ, there is some $j'\in[J']$ such that $H_{j'}[b]=\sigma$. Thus, we may write

$$\Pr_{(t,a,b)\sim\mathcal{Q}}[\mathcal{E}'_{j,(t,b,i),a}]=\Pr_{(t,a,b)\sim\mathcal{Q}}[\mathcal{E}'_{j,(t,b,i),a}\wedge(T_1'[(t,b,i)]_{\to b}\text{ is }b\text{-heavy})]$$

$$+\Pr_{(t,a,b)\sim\mathcal{Q}}[\mathcal{E}'_{j,(t,b,i),a}\wedge(T_1'[(t,b,i)]_{\to b}\text{ is not }b\text{-heavy})]. \quad (41)$$

To bound the first probability, notice that if $\mathcal{E}'_{j,(t,b,i),a}\wedge(T_1'[(t,b,i)]_{\to b}\text{ is }b\text{-heavy})$ occurs, then for some $j'\in[J']$, both events below occur:

- the assignments to a and b given by $T_{1,j}[a]$ and $H_{j'}[b]$ respectively satisfy the constraint $\Phi_{(a,b)}$,

- $D_t(a,T_{1,j}[a])\notin\{(w_{j,1})_t,\ldots,(w_{j,L})_t\}$.

The first item requires unpacking the event $\mathcal{E}'_{j,(t,b,i),a}$. Indeed, if this event holds then $T_1'[(t,b,i)]_a=T_{1,j}[a]$, which implies $T_1'[(t,b,i)]_{\to b}=T_{1,j}[a]|_{\to b}$. Hence, $T_{1,j}[a]|_{\to b}\in\{H_1[b],\ldots,H_{J'}[b]\}$ because $T_1'[(t,b,i)]_{\to b}$ is heavy. Now, to bound the first probability in (41), notice that for a fixed $j'\in[J']$, the above two items occur simultaneously with probability at most ε. Indeed, this follows by (37) applied to $T_{1,j}$ and $H_{j'}$. Union bounding over all $j'\in[J']$ gives that

$$\Pr_{(t,a,b)\sim\mathcal{Q}}[\mathcal{E}'_{j,(t,b,i),a}\wedge(T_1'[(t,b,i)]_{\to b}\text{ is }b\text{-heavy})]\leqslant J'\cdot\varepsilon.$$

We move on to bounding the second probability in (41), and here we will use the agnostic property of $\mathcal{Q}$. We have

$$\Pr_{(t,a,b)\sim\mathcal{Q},i\in[C_{t,b}]}[\mathcal{E}'_{j,(t,b,i),a}\wedge(T_1'[(t,b,i)]_{\to b}\text{ is not }b\text{-heavy})]$$

$$\leqslant\Pr_{(t,a,b)\sim\mathcal{Q},i\in[C_{t,b}]}[T_{1,j}[a]_{\to b}=T_1'[(t,b,i)]_{\to b}\wedge T_1'[(t,b,i)]_{\to b}\text{ is not }b\text{-heavy}]$$

$$=\mathbb{E}_{(t,b)\sim\mathcal{Q},i\in[C_{t,b}]}\left[1_{T_1'[(t,b,i)]_{\to b}\text{ is not }b\text{-heavy}}\Pr_{a\sim\mathcal{Q}(t,\cdot,b)}[T_{1,j}[a]_{\to b}=T_1'[(t,b,i)]_{\to b}]\right] \quad (42)$$

$$=\mathbb{E}_{(t,b)\sim\mathcal{Q},i\in[C_{t,b}]}\left[1_{T_1'[(t,b,i)]_{\to b}\text{ is not }b\text{-heavy}}\Pr_{a\sim\mathcal{Q}(\cdot,\cdot,b)}[T_{1,j}[a]_{\to b}=T_1'[(t,b,i)]_{\to b}]\right]$$

$$\leqslant\sqrt{\varepsilon}.$$

The first transition is by definition of $\mathcal{E}'_{j,(t,b,i),a}$, the second transition is clear, the third transition uses the agnostic property of $\mathcal{Q}$, and the last transition uses the fact that the distribution of a in the probability of the fourth line is the same as the distribution of a in (40) and hence the probability in the expectation of the fourth line is always at most $\sqrt{\varepsilon}$ by definition of not being b-heavy.

Altogether, this yields

$$\Pr_{(t,a,b)\sim\mathcal{Q}}[\mathcal{E}'_{j,(t,b,i),a}] \leqslant J' \cdot \varepsilon + \sqrt{\varepsilon}.$$

Hence, going back to the list-decoding error expression in (38), we get that each term in the summation of the last expression is at most $J' \cdot \varepsilon + \sqrt{\varepsilon}$, and combined with (39) we get that the list-decoding error is at most

$$J \cdot (J' \cdot \varepsilon + \sqrt{\varepsilon}) + \varepsilon^{1/5} \leqslant 2\varepsilon^{1/5}.$$

Hence $\mathcal{D}'$ has $(J \cdot L, 2\varepsilon^{1/5})$-list-decoding soundness, where $J \cdot L = L \cdot \mathsf{poly}(1/\varepsilon)$. $\qquad\square$

8.1 An Inner dPCP

Now we describe how to find a version of our dPCP that will be used as the inner dPCP for composition. Below, we highlight a few technical properties of this inner PCP construction which will be used later on in the composition step.

- The decoding degree is 1.

- There is some integer J such that for every pair of vertices (a,b), the weight given by the complete decoding distribution can be written as $\mathcal{Q}(\circ, a, b) = \frac{w(a,b)}{J}$ for some integer $w(a,b)$. Furthermore, for each a, we have that the sum $\sum_{b\in B} w(a,b)$ is the same. This latter condition also implies that the marginal of the complete decoding distribution over the left vertices is uniform.

We will obtain these properties by using Lemma 8.1, which gives a dPCP with decoding degree 1, thereby establishing the first item above. To achieve the second item, we carefully choose the parameters $C_{t,b}$ so as to adjust the complete decoding distribution, following an approach similar to that of Lemma 7.3.

Lemma 8.2. *Suppose the language $\mathcal{L} \subseteq \Sigma_0^n$ has a projection dPCP*

$$\mathcal{D} = (A \cup B, E, \Sigma_A, \Sigma_B, \{\Phi_e\}_{e\in E}, \{D_t\}_{t\in[n]}, \{\mathcal{P}_t\}_{t\in[n]})$$

with perfect completeness, (L,ε)-list-decoding soundness, agnostic complete decoding distribution, projection decision complexity $\mathsf{ProjComp}$*, decoding complexity* $\mathsf{DecodeComp}$*, and left degree, decoding degree, right degree that are at most* $\mathsf{ld}, \mathsf{ldd}, d$ *respectively. Then for every integer $M \geqslant |B|$, $\mathcal{L}$ has a projection dPCP*

$$\mathcal{D}' = (Z \cup A, E', \Sigma_A^d \times \Sigma_0, \Sigma_A, \{\Phi'_e\}_{e\in E'}, \{D'_t\}_{t\in[n]}, \{\mathcal{P}'_t\}_{t\in[n]})$$

which satisfies the following properties:

- **Length.** *The left side has the form $Z = [n] \times [M]$, so the length is $n \cdot M + |A|$.*

- **Degrees.** *The decoding degree is 1 and the left degrees are at most d. Furthermore if $\mathcal{D}$ is right-regular, then $\mathcal{D}'$ is left regular.*

- **Projection Decision Complexity.** $O(d \cdot (\mathsf{ProjComp} + \mathsf{DecodeComp}))$.

- **Decoding Complexity.** $O(d \log |\Sigma_A| + \log |\Sigma_0|)$.

- **Decoding Distribution.** *For each $t \in [n]$, the marginal of $\mathcal{P}'_t$ over the left vertices, Z, is uniform in $\{t\} \times [M]$.*

- **Completeness.** *The dPCP has perfect completeness.*

- **List-Decoding Soundness.** $\left(L \cdot \mathsf{poly}(1/\varepsilon), 2\varepsilon^{1/5} + \frac{|B|}{M}\right)$*-list-decoding soundness.*

Proof. Fix $M \geqslant |B|$ and take $\mathcal{D}$ to be the dPCP from the lemma statement. We start by using Lemma 7.3 to obtain a discretized version of the marginal of $\mathcal{P}_t$ over B for each $t \in [n]$. We apply Lemma 7.3 with denominator there set to M for each $t \in [n]$, which yields the following. For each $t \in [n]$, there are nonnegative integers $z'_{t,b}$ for each $b \in B$ such that $M := \sum_{b \in B} z'_{t,b}$, and

$$\left| z'_{t,b} - M \cdot \mathcal{P}_t(\circ, b) \right| \leqslant 1, \tag{43}$$

for every $b \in B$. We let $\mathcal{R}_t$ be the distribution over B given by

$$\mathcal{R}_t(b) = \frac{z'_{t,b}}{M}.$$

Note that $\mathcal{R}_t$ is the discretized version of the marginal of $\mathcal{P}_t$ over B given by Lemma 7.3.

Apply Lemma 8.1 to $\mathcal{D}$ with the nonnegative integers $C_{t,b}$ therein set to $C_{t,b} = z'_{t,b}$ to get a dPCP of the form

$$\mathcal{D}_2 = \left(\left(\cup_{t \in [n]} S_t \right) \cup A, E_2, \Sigma_A^d \times \Sigma_0, \Sigma_A, \{\Phi_e\}_{e \in E_2}, \{D_{2,t}\}_{t \in [n]}, \{\mathcal{P}_{2,t}\}_{t \in [n]} \right),$$

where $S_t = \bigcup_{b \in B} \{(t,b)\} \times [z'_{t,b}]$ for each $t \in [n]$. Note that the sets S_t are disjoint and that when defining S_t, if $z'_{t,b} = 0$ we think of $\{(t,b)\} \times [z'_{t,b}]$ as the empty set. Furthermore, for each t,

$$|S_t| = \sum_{b \in B} z'_{t,b} = M. \tag{44}$$

By Lemma 8.1, $\mathcal{D}_2$ has perfect completeness, $(L \cdot \mathsf{poly}(1/\varepsilon), 2\varepsilon^{1/5})$-list-decoding soundness, and decoding degree 1. Likewise, we get that if $\mathcal{D}$ is right-regular, then $\mathcal{D}_2$ is left-regular.

Next, we will modify the decoding distributions of $\mathcal{D}_2$ to obtain our final dPCP, $\mathcal{D}'_2$. Our goal is to obtain a dPCP whose complete decoding distribution has uniform marginal over the left vertices (and we will later show that this implies the decoding distribution guarantee of the theorem statement). Let $\mathcal{Q}_2$ be the complete decoding distribution of $\mathcal{D}_2$. By the decoding distribution guarantee of Lemma 8.1, we have that

$$\mathcal{Q}_2(t, (t, b, i), \circ) = \frac{1}{n} \cdot \frac{\mathcal{P}_t(\circ, b)}{z'_{t,b}} = \frac{1}{n} \cdot \frac{p_{t,b}}{z'_{t,b}}$$

The above shows that the marginal of $\mathcal{Q}_2$ over the left vertices is almost uniform, and hence we will only need to make a small modification.

We note that for each t, b and i, the distribution of $a \sim \mathcal{Q}_2(t, (t, b, i), \cdot)$ is equal to $\mathcal{Q}(t, \cdot, b)$. Indeed, for any $a \in A$, using the decoding distribution guarantee of Lemma 8.1 we have

$$\frac{\mathcal{Q}_2(t, (t, b, i), a)}{\mathcal{Q}_2(t, (t, b, i), \circ)} = \frac{\mathcal{Q}(t, a, b)/C_{t,b}}{\mathcal{Q}(t, \circ, b)/C_{t,b}} = \frac{\mathcal{Q}(t, a, b)}{\mathcal{Q}(t, \circ, b)},$$

where in the second transition we used the fact that (using the decoding distribution guarantee of Lemma 8.1 again)

$$\mathcal{Q}_2(t, (t, b, i), \circ) = \sum_{a \in A} \mathcal{Q}_2(t, (t, b, i), a) = \sum_{a \in A} \frac{\mathcal{Q}(t, a, b)}{C_{t,b}} = \frac{\mathcal{Q}(t, \circ, b)}{C_{t,b}}.$$

On the other hand, consider the distribution $\mathcal{Q}'_2$, generated by choosing $t \in [n]$ uniformly, $b \sim \mathcal{R}_t$, $a \sim \mathcal{Q}(t, \cdot, b)$, $i \in [z'_{t,b}]$ uniformly and outputting $(t, (t, b, i), a)$. Then for any t and (t, b, i), we have

$$\mathcal{Q}'_2(t, (t, b, i), \circ) = \frac{1}{n} \cdot \frac{z'_{t,b}}{M} \cdot \frac{1}{z'_{t,b}} = \frac{1}{n} \cdot \frac{1}{M}.$$

Now we show that $\mathcal{Q}'_2$ is close to $\mathcal{Q}_2$ in TV-distance. First note that for every t, b and i, by definition the distributions $\mathcal{Q}_2(t, \cdot, b) = \mathcal{Q}'_2(t, (t, b, i), \cdot)$, which in turn implies that, $\mathcal{Q}_2(t, (t, b, i), \cdot)$ and $\mathcal{Q}'_2(t, (t, b, i), \cdot)$ are identical. Therefore, the TV-distance between $\mathcal{Q}_2$ and $\mathcal{Q}'_2$ is equal to the TV-distance between their marginal on (t, b, i). It follows that

$$\mathsf{TV}(\mathcal{Q}_2, \mathcal{Q}'_2) \leqslant \sum_{t \in [n], b \in B, i \in [z'_{t,b}]} |\mathcal{Q}_2(t, (t, b, i), \circ) - \mathcal{Q}'_2(t, (t, b, i), \circ)| \leqslant \sum_{t \in [n], b \in B, i \in [z'_{t,b}]} \left| \frac{1}{n} \left(\frac{p_{t,b}}{z'_{t,b}} - \frac{1}{M} \right) \right|.$$

Since the summation is only taken over t, b such that $z'_{t,b} \geqslant 1$, so we can bound each summand there as

$$\frac{1}{n} \left| \frac{p_{t,b}}{z'_{t,b}} - \frac{1}{M} \right| = \frac{1}{n} \left| \frac{M p_{t,b} - z'_{t,b}}{M z'_{t,b}} \right| \leqslant \frac{1}{n M z'_{t,b}},$$

where we used (43). Hence, putting everything together, we get

$$\mathsf{TV}(\mathcal{Q}_2, \mathcal{Q}'_2) \leqslant \frac{|B|}{M}. \tag{45}$$

Our final dPCP, which we call $\mathcal{D}'_2$, is obtained by taking $\mathcal{D}_2$ and changing its complete decoding distribution to $\mathcal{Q}'_2$. In particular all of the parts of $\mathcal{D}'_2$ are the same as in $\mathcal{D}_2$ except for the decoding distributions. We have,

$$\mathcal{D}'_2 = \left(\left(\cup_{t \in [n]} S_t \right) \cup A, E_2, \Sigma_A^d \times \Sigma_0, \Sigma_A, \{\Phi'_e\}_{e \in E_2}, \{D_{2,t}\}_{t \in [n]}, \{\mathcal{P}'_{2,t}\}_{t \in [n]} \right),$$

where for each $t \in [n]$, the new decoding distribution $\mathcal{P}'_{2,t}$ is the distribution over E_2 given by $\mathcal{Q}'_2(t, \cdot, \cdot)$.

It is clear that $\mathcal{D}'_2$ still has perfect completeness and the degree related parameters, projection decision complexity, and decoding complexity are the same as in $\mathcal{D}_2$. By (45), $\mathcal{D}'_2$ has $(L \cdot \mathsf{poly}(1/\varepsilon), 2\varepsilon^{1/5} + |B|/M)$-list decoding soundness. Finally, for each $t \in [n]$, it can be checked that the decoding distribution, $\mathcal{P}'_{2,t}$, chooses a vertex in S_t uniformly at random, and furthermore $|S_t| = M$. Hence, we can view the left vertices as $[n] \times [M]$ and the decoding distribution $\mathcal{P}'_{2,t}$ is indeed as described in the lemma. $\qquad \square$

We now describe how to obtain the inner dPCP that we will use for composition.

Theorem 8.3 (Inner dPCP). *For every $\varepsilon > 0$ there exist $L = \mathsf{poly}(1/\varepsilon)$ and functions $M, K : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ satisfying*

$$M(T, N) = O_T(N \mathsf{poly}_\varepsilon(\log N)) \quad and \quad K(T, N) = O_T(\mathsf{poly}_\varepsilon(\log N))$$

such that the following holds. For every alphabet Σ_0 of size at most T and circuit $\varphi : \Sigma_0^n \to \{0,1\}$ of size at most N, the language $\mathsf{SAT}(\varphi)$ has a projection dPCP of the form

$$\mathcal{D} = \left(([n] \times [M]) \cup B, E, \Sigma_A, \Sigma_B, \{\Phi_e\}_{e \in E}, \{D_t\}_{t \in [n]}, \{\mathcal{P}_t\}_{t \in [n]} \right)$$

which satisfies the following with $M := M(T, N)$ and $K := K(T, N)$

- **Length.** $O_T(nM + N \mathsf{poly}_\varepsilon(\log N))$.

- **Alphabet Sizes.** *The left and right alphabet sizes are $2^{(O_T(\mathsf{poly}_\varepsilon(\log N)))}$ and $O_\varepsilon(1)$ respectively.*

- **Degrees.** $\mathcal{D}$ *has decoding degree* 1.

- **Projection Decision Complexity.** $O_{|T|}(\mathsf{poly}_\varepsilon(\log N))$.

- **Decoding Complexity.** $O_{|T|}(\mathsf{poly}_\varepsilon(\log N))$.

- **Decoding Distribution.** *Let $\mathcal{Q}$ be the complete decoding distribution. Then, $\mathcal{Q}$ satisfies the following:*

  - *For each $(t, r) \in [n] \times [M]$ and $b \in B$ we can write $\mathcal{Q}(\circ, (t, r), b) = \frac{w((t,r),b)}{nMK}$ for some $w((t,r), b) \in \mathbb{N}$.*
  - *For each $t \in [n]$, the distribution $\mathcal{Q}(t, \cdot, \circ)$ is uniform over $\{t\} \times [M]$ and hence for every $(t, r) \in [n] \times [M]$,*

$$\sum_{b \in B} w((t,r), b) = K$$

- **Completeness.** *The dPCP has perfect completeness.*

- **List-Decoding Soundness.** *The dPCP has (L, ε)-list-decoding soundness.*

Proof. Fix $\varepsilon > 0$, fix sizes T and N and let $\varphi : \Sigma_0^n \to \{0,1\}$ be a circuit of size at most N with $|\Sigma_0| \leqslant T$. We take $\mathcal{D}_{\mathsf{dp}}$ to be the dPCP for $\mathsf{SAT}(\varphi)$ in Theorem 6.1 with soundness parameter set to ε:

$$\mathcal{D}_{\mathsf{dp}} = (X(k), X(\sqrt{k}), E_{\mathsf{dp}}, \Sigma^k, \Sigma^{\sqrt{k}}, \{\Phi_e\}_{e \in E_{\mathsf{dp}}}, \{D_{\mathsf{dp},t}\}_{t \in [n]}, \{\mathcal{P}_{\mathsf{dp},t}\}_{t \in [n]}).$$

Now set $M = O_T(N\mathsf{poly}_\varepsilon(\log N))$ sufficiently large relative to $|X\left(\sqrt{k}\right)| = O_T(N\mathsf{poly}_\varepsilon(\log N))$. We also have that there is some $d_r = \mathsf{poly}_\varepsilon(\log N)$ such that regardless of the initial circuit φ, the right degree of every vertex is at most d_r. We will transform $\mathcal{D}_{\mathsf{dp}}$ into the desired inner dPCP.

Decoding Degree Reduction + Right Alphabet reduction. We first apply Lemma 8.2 with M set as above, and then apply right alphabet reduction from Lemma 7.1 with sufficiently small $\eta = \mathsf{poly}(\varepsilon)$ on the resulting dPCP. Call the dPCP after both transformations $\mathcal{D}_2$ and let its constraint graph be $([n] \times [M] \cup B, E)$. Note that we can assume such structure on the left vertices by Lemma 8.2. Call the resulting complete decoding distribution $\mathcal{Q}_2$. Let us list a few properties we have that are straightforward to verify from Lemma 8.2 and Lemma 7.1.

- **Length.** $|B| = |X(k)| \cdot O_\eta(\sqrt{k} \log |\Sigma|) = O_T(N\mathsf{poly}_\varepsilon(\log N))$. This follows from the length guarantees of Lemma 8.2 and the structure of the resulting constraint graph from Lemma 7.1. The length is therefore $nM + |B| = O_T(nM + N\mathsf{poly}_\varepsilon(\log N))$.

- **Alphabets.** The left alphabet has size at most $|\Sigma|^{k \cdot d_r} \cdot T = 2^{O_T(\mathsf{poly}_\varepsilon(\log N))}$ after applying Lemma 8.2, and this is unaffected by Lemma 7.1. The right alphabet has size at most $O_\varepsilon(1)$ by Lemma 7.1.

- **Degrees.** The decoding degree is 1 after applying Lemma 8.2 and this is unaffected by Lemma 7.1. The degree of each left vertex is at most $O_T(\mathsf{poly}_\varepsilon(\log N))$. This follows from the left degree guarantee of Lemma 8.2 (which turns right degree of $\mathcal{D}_{\mathsf{dp}}$ into left degree) and the fact that Lemma 7.1 multiplies the left degree by $O_\eta(\sqrt{k} \log |\Sigma|) = O_T(\mathsf{poly}_\varepsilon(\log N))$. We refer to Theorem 6.1 to bound $\sqrt{k} \log |\Sigma| \leqslant O_T(\mathsf{poly}_\varepsilon(\log N))$.

- **Projection Decision Complexity.** $O_T(\mathsf{poly}_\varepsilon(\log N))$. This is straightforward to check by using Theorem 6.1, then Lemma 8.2, and finally Lemma 7.1.

- **Decoding Complexity.** $O_T(\mathsf{poly}_\varepsilon(\log N))$. This is straightforward to check by using Theorem 6.1, then Lemma 8.2, and finally Lemma 7.1.

- **Decoding Distribution.** The marginal of $\mathcal{Q}_2$ over $[n] \times [M]$ is uniform and each left vertex $(t, r) \in [n] \times [M]$ has exactly one decoding neighbor, t. This follows from the decoding distribution and decoding degree guarantees of Lemma 8.2,

- **Completeness.** $\mathcal{D}_2$ has perfect completeness because $\mathcal{D}_{\mathsf{dp}}$ does and both Lemma 8.2 and Lemma 7.1 preserve it.

- **Soundness.** $\mathcal{D}_2$ has $(L \cdot \mathsf{poly}(1/\varepsilon), O(\varepsilon^{1/5}))$ soundness by Theorem 6.1, Lemma 8.2, and Lemma 7.1. Here we use the fact that M is sufficiently large relative to $|X(\sqrt{k})|$.

At this point, we have all of the desired properties, except for the discreteness of the marginal of the complete decoding distribution over the edges (i.e., the first guarantee of the decoding distribution above). We resolve this next, and some care is needed to accomplish this while also preserving a uniform marginal over the left vertices.

Discretizing the Complete Decoding Distribution in a Left Uniform Manner. We construct a discretized version of the distribution $\mathcal{Q}_2(t, (t, r), \cdot)$ for each $(t, r) \in [n] \times [M]$. Specifically, note that $|\mathsf{supp}(\mathcal{Q}_2(t, (t, r), \cdot))|$ is upper bounded by the degree of (t, r), which is $O_T(\mathsf{poly}_\varepsilon(\log N))$. We apply Lemma 7.3 with denominator set to a sufficiently large $M' = O_T(\mathsf{poly}_\varepsilon(\log N))$ and let $\Delta_{t,r,\mathsf{disc}}$ be the distribution obtained. Here we again note that we can choose the same M' regardless of the specific circuit φ fixed at the start. By Lemma 7.3 and the fact that M' is sufficiently large, we have

$$\mathsf{TV}(\Delta_{t,r,\mathsf{disc}}, \mathcal{Q}_2(t, (t, r), \cdot)) \leqslant \varepsilon \tag{46}$$

and $\mathsf{supp}(\Delta_{t,r,\mathsf{disc}}) \subseteq \mathsf{supp}(\mathcal{Q}_2(t, (t, r), \cdot))$. Now consider the following complete decoding distribution $\mathcal{Q}_2'$:

- Choose $(t, r) \in [n] \times [M]$ uniformly.

- Choose $b \sim \Delta_{t,r,\mathsf{disc}}$.

- Output $(t, (t, r), b)$,

and let $\mathcal{D}_2'$ be the dPCP obtained by changing the complete decoding distribution of $\mathcal{D}_2$ from $\mathcal{Q}_2$ to $\mathcal{Q}_2'$.

Since $\mathcal{Q}_2'$ and $\mathcal{Q}_2$ have the same marginal distribution over $[n] \times ([n] \times [M])$, we have

$$
\begin{aligned}
\mathsf{TV}(\mathcal{Q}_2', \mathcal{Q}_2) &= \sum_{(t,r) \in [n] \times [M]} \mathcal{Q}_2(t, (t, r), \circ) \cdot \mathsf{TV}(\mathcal{Q}_2'(t, (t, r), \cdot), \mathcal{Q}(t, (t, r), \cdot)) \\
&= \sum_{(t,r) \in [n] \times [M]} \mathcal{Q}_2(t, (t, r), \circ) \cdot \mathsf{TV}(\Delta_{t,r,\mathsf{disc}}, \mathcal{Q}(t, (t, r), \cdot)) \qquad (47) \\
&\leqslant \varepsilon,
\end{aligned}
$$

where the last transition uses (46). We now verify that $\mathcal{D}_2'$ satisfies all of the desired properties. The length, degree, projection decision complexity, and decoding complexity are all unchanged from $\mathcal{D}_2$ and hence follow by the items above. Perfect completeness is preserved in $\mathcal{D}_2'$ because $\mathrm{supp}(\Delta_{t,r,\mathsf{disc}}) \subseteq \mathrm{supp}(\mathcal{Q}_2(t, (t, r), \cdot))$ as noted above, and $\mathcal{D}_2'$ still has $(L \cdot \mathsf{poly}(1/\varepsilon), O(\varepsilon^{1/5}))$-soundness because of (47). This shows the desired soundness condition because ε can be taken arbitrarily small and we can reparameterize the list size. Finally, we check the two properties of the decoding distribution. First, it is clear from the above that $\mathcal{Q}_2'$ satisfies that for each $t \in [n]$, the distribution $\mathcal{Q}_2'(t, \cdot, \circ)$ over $\{t\} \times [M]$ is uniform, so this takes care of the second condition there. For the first condition, we note that for any $(t, r) \in [n] \times [M], b \in B$ we can write

$$
\mathcal{Q}_2'(t, (t, r), b) = \frac{1}{n} \cdot \frac{1}{M} \cdot \Delta_{t,r,\mathsf{disc}}(b) = \frac{w((t, r), b)}{nMM'},
$$

for some $w((t, r), b) \in \mathbb{N}$. Hence, the first condition follows for $K(T, N) = M' = O_T(\mathsf{poly}_\varepsilon(\log N))$. $\qquad \square$

8.2 An Outer dPCP

The following result will be used as our outer dPCP. Compared to Lemma 8.2, the construction below has quasi-linear length and a complete decoding distribution whose marginal over the right vertices is uniform.

Theorem 8.4. *For every $\varepsilon > 0$ there exist $d', L = \mathsf{poly}(1/\varepsilon)$ such that the following holds. For any alphabet Σ_0 and circuit $\varphi : \Sigma_0^n \to \{0, 1\}$ of size at most N, the language $\mathsf{SAT}(\varphi)$ has a projection dPCP of the form*

$$
\mathcal{D} = (A \cup B, E, \Sigma_B^{\mathsf{poly}_\varepsilon(\log N)} \times \Sigma_0, \Sigma_B, \{\Phi_e\}_{e \in E}, \{D_t\}_{t \in [n]}, \{\mathcal{P}_t\}_{t \in [n]})
$$

which satisfies the following

- ***Length.*** $O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon(\log N))$.

- ***Alphabet Sizes.*** *The left and right alphabet sizes are $2^{O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))}$ and $O_\varepsilon(1)$ respectively.*

- ***Degrees.*** *The dPCP has decoding degree 1.*

- ***Projection Decision Complexity.*** $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$.

- ***Decoding Complexity.*** $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$.

- ***Decoding Distribution.*** *The complete decoding distribution, $\mathcal{Q}$, satisfies*

  - *The marginal of the complete decoding distribution over $A \times B$ is K-discrete for some*

    $$
    K = O_{|\Sigma_0|}(N\mathsf{poly}_\varepsilon(\log N)).
    $$

59

- *Writing $\mathcal{Q}(\circ, a, b) = \frac{w(a,b)}{K}$ for integers $w(a, b)$, we have $\sum_{a \in A} w(a, b) = d'$ for all $b \in B$ and hence the marginal over the right vertices is uniform.*

- *For each left vertex a, $\sum_{b \in B} w(a, b) \leqslant O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$.*

- **Completeness.** *The dPCP has perfect completeness.*

- **List-Decoding Soundness.** *The dPCP has (L, ε)-list-decoding soundness.*

Proof. Fix $\varepsilon > 0$, and let $\mathcal{D}_1 = (X(k), X(\sqrt{k}), E_1, \Sigma^k, \Sigma^{\sqrt{k}}, \{\Phi_e\}_{e \in E_1}, \{D_{1,t}\}_{t \in [n]}, \{\mathcal{P}_{1,t}\}_{t \in [n]})$ be the dPCP from Theorem 6.1 with (L, ε)-list-decoding soundness where $L = \mathsf{poly}(1/\varepsilon)$. This dPCP has decoding degree, left degree, and right degree that are $\mathsf{poly}_\varepsilon(\log N), O_\varepsilon(1), \mathsf{poly}_\varepsilon(\log N)$ respectively, and its complete decoding distribution, which we call $\mathcal{Q}_1$, is agnostic and has marginal over $X(k) \times X(\sqrt{k})$ which is K-discrete for some $K = N\mathsf{poly}_\varepsilon(\log N)$. By the decoding distribution guarantee of Theorem 6.1, for each $\mathbf{b} \in X(\sqrt{k})$, we have

$$\mathcal{Q}_1(\circ, \circ, \mathbf{b}) \leqslant \frac{\mathsf{poly}_\varepsilon(\log N)}{N}. \tag{48}$$

Starting from $\mathcal{D}_1$, we will first apply decoding degree reduction (Lemma 8.1) where all of the $C_{t,b}$ therein are set to 1, and then apply right alphabet reduction (Lemma 7.1). We call the resulting dPCP $\mathcal{D}_2$, and obtain our final dPCP, which we call $\mathcal{D}_3$, by applying right degree reduction on $\mathcal{D}_2$.

Decoding Degree Reduction + Right Alphabet Reduction. First, we note that $\mathcal{D}_1$ has agnostic complete decoding distribution by Theorem 6.1, so we may apply Lemma 8.1 with the $C_{t,b}$ therein all set to 1. We then apply Lemma 7.1 with $\eta = O(\varepsilon)$. The resulting dPCP,

$$\mathcal{D}_2 = \left(A_2 \cup B_2, E_2, \Sigma_A, \Sigma_B, \{\Phi_e\}_{e \in E_2}, \{D_{2,t}\}_{t \in [n]}, \{\mathcal{P}_{2,t}\}_{t \in [n]} \right),$$

has the following properties, which are straightforward to verify.

- **Completeness.** Perfect completeness.

- **Length.** We have $|A_2| \leqslant |X(\sqrt{k})| \cdot \mathsf{poly}_\varepsilon(\log N)$ and $|B_2| \leqslant O_\varepsilon(|X(k)| \cdot \log(|\Sigma|^k))$, so the overall length is $O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon(\log N))$.

- **Degrees.** The left degree is $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$, the decoding degree is 1, and the right degree is $\mathsf{poly}_\varepsilon(\log N)$.

- **Projection Decision Complexity.** $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$.

- **Decoding Complexity.** $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$.

- **List-Decoding Soundness.** $(\mathsf{poly}(1/\varepsilon), O(\varepsilon^{1/5}))$-list-decoding soundness.

In order to apply right degree reduction next, we need to analyze the complete decoding distribution of $\mathcal{D}_2$, which we call $\mathcal{Q}_2$. Fix an arbitrary $a_2 \in A_2$ and let us bound $\mathcal{Q}_2(\circ, a_2, \circ)$. First, since the decoding degree is 1, there is some t such that

$$\mathcal{Q}_2(\circ, a_2, \circ) = \mathcal{Q}_2(t, a_2, \circ) = \mathcal{P}_{2,t}(a_2, \circ)$$

Then, by the decoding distribution guarantee of Lemma 8.1, there is some $\mathbf{b} \in X(\sqrt{k})$, such that

$$\mathcal{Q}_2(\circ, a_2, \circ) = \mathcal{P}_{2,t}(a_2, \circ) = \mathcal{P}_{1,t}(\circ, \mathbf{b}) = \mathcal{Q}_1(t, \circ, \mathbf{b}).$$

Now applying (48), we have

$$\mathcal{Q}_2(\circ, a_2, \circ) = Q_1(t, \circ, \mathbf{b}) \leqslant Q_1(\circ, \circ, \mathbf{b}) \leqslant \frac{\mathsf{poly}_\varepsilon(\log N)}{N}. \tag{49}$$

Right Degree reduction. Having analyzed the complete decoding distribution of $\mathcal{D}_2$, we are now ready to apply Lemma 7.4 on it. Note that we need to apply the version of right degree reduction in Lemma 7.4, which changes the complete decoding distribution, because we did not analyze the discreteness of the marginal of $\mathcal{Q}_2$ over the edges.

To this end, set

$$M = \Theta_{|\Sigma_0|}\left(\frac{N\mathsf{poly}_\varepsilon(\log N)}{n}\right)$$

sufficiently large and degree parameter d' sufficiently large relative to ε and apply Lemma 7.4 with these parameters. To see that we have set M large enough to apply Lemma 7.4, we use the fact that $\mathcal{D}_2$ has left degree at most $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$, and $|A_2| \leqslant N\mathsf{poly}_\varepsilon(\log N)$. Call the resulting dPCP

$$\mathcal{D}_3 = \left(A_2 \cup B_3, E_3, \Sigma_A, \Sigma_B, \{\Phi_e\}_{e \in E_3}, \{D_{2,t}\}_{t \in [n]}, \{\mathcal{P}_{3,t}\}_{t \in [n]}\right),$$

and let $\mathcal{Q}_3$ be the complete decoding distribution of $\mathcal{D}_3$. By Lemma 7.4 there are positive integers $w(a, b)$ for each $(a, b) \in A_2 \times B_3$ such that we can write $\mathcal{Q}_3(\circ, a, b) = \frac{w(a,b)}{d'^2 nM}$. Fix a left vertex a of $\mathcal{D}_3$. By the decoding distribution guarantee of Lemma 7.4, we have that either,

$$\mathcal{Q}_3(\circ, a, \circ) \leqslant \mathcal{Q}_2(\circ, a, \circ) + \frac{|\Gamma_2(a)|}{nM} \leqslant \frac{O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))}{N} \tag{50}$$

or

$$\mathcal{Q}_3(\circ, a, \circ) \leqslant \frac{1}{nM} + \frac{\varepsilon}{10|A_2|} \leqslant \frac{1}{N}. \tag{51}$$

In the above inequalities, $\Gamma_2(a)$, denotes a's neighborhood in $\mathcal{D}_2$, and the first inequality uses Lemma 7.4 while the second inequality uses the fact that $\mathcal{D}_2$ has left degree $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$ and $|A_2| \geqslant N$. We can now verify the listed properties for $\mathcal{D}_3$

Length. The right side has size $d'nM = O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(N \log N))$ and the left side has size $|A_2|$, so the final length is $O_{|\Sigma_0|}(N\mathsf{poly}_\varepsilon(\log N))$.

Alphabet Sizes. The left alphabet size is $|\Sigma_A| = 2^{O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))}$ and the right alphabet size is $O_\varepsilon(1)$ because we applied Lemma 7.1.

Degrees. The decoding degree is 1 in $\mathcal{D}_2$ and this is preserved by Lemma 7.4 when going to $\mathcal{D}_3$.

Projection Decision Complexity. The decision complexity is $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$ in $\mathcal{D}_2$ and this is preserved by Lemma 7.4.

Decoding Complexity. The decoding complexity is $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$ in $\mathcal{D}_2$ and this is preserved by Lemma 7.4.

Decoding Distribution. The complete decoding distribution is $\mathcal{Q}_3$ which is K-discrete for

$$K := d'^2 nM = O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon(\log N)).$$

The second item in the decoding distribution condition, follows because the complete decoding distribution guarantee of Lemma 7.4 results in $\mathcal{Q}_3$ having uniform marginal over the right vertices. The third item follows from (50) and (51) as for any a we have

$$\sum_{b \in B} w(a, b) = K \cdot \mathcal{Q}_3(\circ, a, \circ) \leqslant O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N)).$$

Completeness. We have perfect completeness in $\mathcal{D}_2$ and this is preserved by Lemma 7.4.

Soundness. It is straightforward to check that the $\mathcal{D}_3$ has $(\mathsf{poly}(1/\varepsilon), O(\varepsilon^{1/5}))$-list-decoding soundness by the list-decoding soundness of $\mathcal{D}_2$ and Lemma 7.4. □

9 A Composition Theorem for Decodable PCPs

This section proves a composition theorem for dPCPs, which facilitates left alphabet reduction. In Section 10 we show how to apply it on the constructions from Section 8 (with a final composition step with a standard Hadamard-code based dPCP) to reduce the alphabet size from $2^{\mathsf{poly}(\log(N))}$ all the way down to $O(1)$.

Our composition result is as follows. Throughout the proof, we will use bold letters (e.g., $\mathbf{a}, \mathbf{b}$) for outer dPCP vertices, and non-boldface symbols for inner dPCP vertices (e.g. a, b).

Theorem 9.1. *Suppose we have the following two dPCPs, which we call outer dPCP and inner dPCP.*

- ***Outer dPCP.*** *There exists $C_1 \in \mathbb{N}$ such that for every $\varepsilon > 0$ there exist $d_1 \in \mathbb{N}$ and functions $F_1, \mathsf{DecisionComp}_1,$ $\mathsf{DecodeComp}_1, K_1, k_1 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that every circuit $\varphi : \Sigma_0^n \to \Sigma$ of size N over some alphabet Σ_0 has a dPCP as follows*

  - ***Parameters.*** *Length $S_1(N, |\Sigma_0|)$, left and right alphabets of sizes $F_1(N, |\Sigma_0|)$ and $O_\varepsilon(1)$ respectively, decoding degree $O_\varepsilon(1)$, projection decision complexity at most $\mathsf{DecisionComp}_1(N, |\Sigma_0|)$, and decoding complexity at most $\mathsf{DecodeComp}_1(N, |\Sigma_0|)$.*

  - ***Complete Decoding Distribution.*** *Call the complete decoding distribution $\mathcal{Q}_{\mathsf{out}}$ and the sides of the constraint graph $A_{\mathsf{out}}, B_{\mathsf{out}}$. Then, the following holds*
    - *(Discreteness): for every $\mathbf{a} \in A_{\mathsf{out}}, \mathbf{b} \in B_{\mathsf{out}}$, $\mathcal{Q}_{\mathsf{out}}(\circ, \mathbf{a}, \mathbf{b}) = \frac{\mathsf{wt}(\mathbf{a}, \mathbf{b})}{K_1(N, |\Sigma_0|)}$ for $\mathsf{wt}(\mathbf{a}, \mathbf{b}) \in \mathbb{N}$.*
    - *(Right Uniform): for every right vertex $\mathbf{b}$, $\sum_{\mathbf{a} \in A_{\mathsf{out}}} \mathsf{wt}(\mathbf{a}, \mathbf{b}) = d_1$, and hence the marginal of $\mathcal{Q}_{\mathsf{out}}$ is uniform over B_{out}.*
    - *(Left Bounded): for every left vertex $\mathbf{a}$, $\mathsf{wt}(\mathbf{a}) := \sum_{\mathbf{b} \in B_{\mathsf{out}}} w(\mathbf{a}, \mathbf{b}) \leqslant k_1(N, |\Sigma_0|)$.*

  - ***Completeness.*** *Perfect completeness.*

  - ***Soundness.*** *$(\mathsf{L}_{\mathsf{out}}(\varepsilon), \varepsilon)$-list-decoding soundness where $\mathsf{L}_{\mathsf{out}} = \varepsilon^{-C_1}$.*

- ***Inner dPCP.*** *There exists $C_2 \in \mathbb{N}$ such that for every $\varepsilon > 0$ there are bivariate functions $F_2, \mathsf{DecisionComp}_2,$ $\mathsf{DecodeComp}_2, K_2 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $M : \mathbb{N} \to \mathbb{N}$ such that every circuit $\varphi : (\Sigma_0')^{n'} \to \Sigma$ of size at most N over some alphabet Σ_0' of size at most T' has a dPCP as follows.*

  - ***Parameters.*** *Length $S_2(N', T')$, left and right alphabets of sizes $F_2(N', T')$ and $O_\varepsilon(1)$ respectively, decoding degree 1, projection decision complexity at most $\mathsf{DecisionComp}_2(N', T')$, and decoding complexity at most $\mathsf{DecodeComp}_2(N', T')$.*

  - ***Complete Decoding Distribution.*** *Call the complete decoding distribution $\mathcal{Q}_{\mathsf{in}}$ and the sides of the constraint graph $A_{\mathsf{in}}, B_{\mathsf{in}}$. Then the following holds:*
    - *(Discreteness): for every $a \in A_{\mathsf{in}}, b \in B_{\mathsf{in}}$, we have $\mathcal{Q}_{\mathsf{in}}(\circ, a, b) = \frac{\mathsf{wt}'(a, b)}{n' \cdot M(N', T') \cdot K_2(N', T')}$ for $\mathsf{wt}'(a, b) \in \mathbb{N}$.*
    - *(Left Uniform): the left vertices can be labeled as $[n'] \times [M(N', T')]$, so that for every $t \in [n']$, the distribution $\mathcal{Q}_{\mathsf{in}}(t, \cdot, \circ)$ is uniform over $\{t\} \times [M(N', T')]$. It follows that the marginal distribution of $\mathcal{Q}_{\mathsf{in}}$ over the left vertices is uniform and for every left vertex a, $\mathsf{wt}'(a) := \sum_{b \in B_{\mathsf{in}, a}} \mathsf{wt}'(a, b) = K_2(N', T')$.*

  - ***Completeness.*** *perfect completeness.*

  - ***Soundness.*** *$(\mathsf{L}_{\mathsf{in}}(\varepsilon), \varepsilon)$-list-decoding soundness where $\mathsf{L}_{\mathsf{in}} = \varepsilon^{-C_2}$.*

Then for every $\varepsilon > 0$, and every circuit $\varphi : \Sigma_0^n \to \{0, 1\}$ of size N, the language $\mathsf{SAT}(\varphi)$ has a projection dPCP with the following properties, where we set

$$N' = O(k_1(N, |\Sigma_0|) + \mathsf{DecisionComp}_1(N, |\Sigma_0|)) \qquad and \qquad T' = O_{\varepsilon, |\Sigma_0|}(1).$$

- **Length.** $S_1(N, |\Sigma_0|) \cdot (M(N', T') + S_2(N', T'))$.

- **Left alphabet size.** $F_2(N', T')^{d_1}$.

- **Right alphabet size.** $O_\varepsilon(1)$.

- **Decoding Degree.** $O_\varepsilon(d_1)$.

- **Projection Decision Complexity.**

$$O(\log(F_1(N, |\Sigma_0|))) + O_{\varepsilon, |\Sigma_0|}(\mathsf{DecisionComp}_2(N', T') + \mathsf{DecodeComp}_2(N', T')).$$

- **Decoding Complexity.** $\log(F_1(N, |\Sigma_0|)) + \mathsf{DecodeComp}_2(N', T')$.

- **Complete Decoding Distribution.** *Call the complete decoding distribution $\mathcal{Q}_{\mathsf{comp}}$ and the sides of the constraint graph $A_{\mathsf{comp}}, B_{\mathsf{comp}}$. Then the following holds:*

  - *(Discreteness):* $\mathcal{Q}_{\mathsf{comp}}(\circ, a, b) = \frac{\mathsf{wt}''(a,b)}{J}$ *for* $\mathsf{wt}''(a, b) \in \mathbb{N}$ *and* $J = K_1(N, |\Sigma_0|) \cdot M(N', T') \cdot K_2(N', T')$.
  - *(Left Uniform): The marginal of $\mathcal{Q}_{\mathsf{comp}}$ over A_{comp} is uniform, and for every $a \in A_{\mathsf{comp}}$ we have $\sum_{b \in B} \mathsf{wt}''(a, b) = d_1 K_2(N', T')$.*

- **Completeness.** *The dPCP has perfect completeness.*

- **Soundness.** *The dPCP has $(\mathsf{poly}(1/\varepsilon), \varepsilon)$-list-decoding soundness.*

Proof overview. Our proof strategy follows the lines of the proof of [MR10] and [DH13]. Specifically, we construct a version of the inner dPCP for each left vertex **a** in the outer dPCP. The size of the circuit relevant for a given left vertex is determined by (1) the projection decision complexity, (2) the parameter k_1 which should be thought of as a weighted version of left degree (with weights coming from the complete decoding distribution), and (3) the decoding degree. In our application all of these are $O_{|\Sigma_0|}(\mathsf{poly}(\log(N)))$ or $O_{\varepsilon, |\Sigma_0|}(1)$, so the inner dPCP is run on a much smaller scale.

In order to compose all of these smaller inner dPCPs into one overall dPCP, we attempt to check several constraints "in parallel". Namely, as in [MR10, DH13] we think of choosing a right vertex **b** of the outer PCP and checking all of the inner PCPs of **a** that are neighbours of **b**. This requires some alignment between the inner PCPs of various **a**, which is handled by the extra flexibility we have arranged in Section 8 in the size of the PCPs. Additional care is needed to make sure that the decoding is still maintained throughout, and this aspect is where we differ from the prior compositions mentioned. Indeed, this is why we went through extensive efforts in Lemma 8.1 to reduce the decoding degree of the dPCP from Theorem 6.1 to 1. Roughly speaking, having small decoding degree allows us to pass decoding to the inner dPCPs while not increasing their alphabet size by too much.

9.1 The Construction

Fix $\varepsilon > 0$ and a size N circuit $\varphi : \Sigma_0^n \to \{0, 1\}$. Our goal is to construct a dPCP for the language $\mathcal{L} := \mathsf{SAT}(\varphi) \subseteq \Sigma_0^n$, and we have the outer dPCP $\mathcal{D}_{\mathsf{out}}$ and the inner dPCP $\mathcal{D}_{\mathsf{in}}$ at our disposal. We will apply the outer dPCP to φ and the inner dPCP for circuits $\varphi_{\mathbf{a}}$ of size roughly $O(k_1(N, |\Sigma_0|)) + O(\mathsf{DecisionComp}_1(N, |\Sigma_0|))$, which should be thought of as much smaller than N.

The Outer dPCP. Take $\mathcal{D}_{\mathsf{out}}$ to be the outer dPCP for $\mathcal{L}$ as in the premise with the soundness error set to $\delta_{\mathsf{out}} := \varepsilon^{C_2+10}$. We denote its parts by

$$\mathcal{D}_{\mathsf{out}} = \left(A_{\mathsf{out}} \cup B_{\mathsf{out}}, E_{\mathsf{out}}, \Sigma_{A_{\mathsf{out}}}, \Sigma_{B_{\mathsf{out}}}, \{\Phi_{\mathsf{out}, e}\}_{e \in E_{\mathsf{out}}}, \{D_t^{\mathsf{out}}\}_{t \in [n]}, \{\mathcal{P}_t^{\mathsf{out}}\}_{t \in [n]}\right).$$

Let q_1 denote the decoding degree of $\mathcal{D}_{\mathsf{out}}$ so that $q_1 = O_\varepsilon(1)$ by assumption.

Henceforth, it will be convenient to let $\Gamma_{\mathsf{disc,out}}(\mathbf{a})$ denote the set of $\mathbf{a}$'s neighbors listed with multiplicity $\mathsf{wt}(\mathbf{a},\mathbf{b})$. In order to avoid confusion when referring to multiple copies of $\mathbf{b}$ (which appears in $\Gamma_{\mathsf{disc,out}}(\mathbf{a})$ more than once if $\mathsf{wt}(\mathbf{a},\mathbf{b}) > 1$), we refer to the members of $\Gamma_{\mathsf{disc,out}}(\mathbf{a})$ as $(\mathbf{b},i)$ where for each $\mathbf{b} \in \Gamma_{\mathsf{out}}(\mathbf{a})$, $i \in [\mathsf{wt}(\mathbf{a},\mathbf{b})]$. Hence,

$$\Gamma_{\mathsf{disc,out}}(\mathbf{a}) = \bigcup_{b \in \Gamma_{\mathsf{out}}(\mathbf{a})} \{\mathbf{b}\} \times [\mathsf{wt}(\mathbf{a},\mathbf{b})].$$

Notice that choosing $(\mathbf{b},i) \in \Gamma_{\mathsf{disc,out}}(\mathbf{a})$ uniformly, the marginal distribution of $\mathbf{b}$ is $\mathcal{Q}_{\mathsf{out}}(\circ,\mathbf{a},\cdot)$.

The following notation will be useful.

Definition 9.2. *For a symbol σ for $\boldsymbol{a} \in A_{\mathsf{out}}$ and $\boldsymbol{b} \in \Gamma_{\mathsf{out}}(\boldsymbol{a})$, let $\sigma_{\to\boldsymbol{b}}$ denote the unique Σ_B symbol that satisfies the projection constraint on $(\boldsymbol{a},\boldsymbol{b})$. For each $t \in \Gamma_{\mathsf{dec,out}}(\boldsymbol{a})$, we write $\sigma_{\rightsquigarrow t} = D_t^{\mathsf{out}}(\boldsymbol{a},\sigma)$. Here, the vertex $\boldsymbol{a}$ will always be clear from context.*

The Inner dPCPs. We will also construct an inner dPCP corresponding to each outer vertex $\mathbf{a}$. For each $\mathbf{a} \in A_{\mathsf{out}}$ we start by defining a language

$$\mathcal{L}_{\mathbf{a}} \subseteq \left(\Sigma_{B_{\mathsf{out}}} \times \Sigma_0^{q_1}\right)^{\mathsf{wt}(\mathbf{a})},$$

which is supposed to correspond to the set of valid alphabet symbols for $\mathbf{a}$. Set,

$$T' = |\Sigma_{B_{\mathsf{out}}}| \times |\Sigma_0|^{q_1} = O_{\varepsilon,|\Sigma_0|}(1), \tag{52}$$

to be the size of the alphabet which each of the languages $\mathcal{L}_{\mathbf{a}}$ is over. In the definition of $\mathcal{L}_{\mathbf{a}}$, we identify $[\mathsf{wt}(\mathbf{a})]$ with the multiset $\Gamma_{\mathsf{disc,out}}(\mathbf{a})$. Let us now fix an $\mathbf{a}$ and define the language $\mathcal{L}_{\mathbf{a}}$.

For each $w \in \mathcal{L}_{\mathbf{a}}$, think of its $\mathsf{wt}(\mathbf{a})$ indices as being indexed by $\Gamma_{\mathsf{disc,out}}(\mathbf{a})$ and for each $(\mathbf{b},i) \in \Gamma_{\mathsf{disc,out}}(\mathbf{a})$ we write $w_{(\mathbf{b},i)}$ as the symbol in w at index $(\mathbf{b},i)$. The symbol $w_{(\mathbf{b},i)}$ is thought of as consisting of one $\Sigma_{B_{\mathsf{out}}}$ symbol corresponding to an implied label for $\mathbf{b}$, and q_1 symbols from Σ_0, corresponding to one implied label for each $t \in \Gamma_{\mathsf{out,dec}}(\mathbf{a})$. With this setup in mind, the members of $\mathcal{L}_{\mathbf{a}}$ are exactly the $w \in (\Sigma_{B_{\mathsf{out}}} \times \Sigma_0^{q_1})^{\mathsf{wt}(\mathbf{a})}$ such that there exists $\sigma \in \Sigma_{\mathbf{a}}$ satisfying the following two conditions:

- for every $(\mathbf{b},i) \in \Gamma_{\mathsf{disc,out}}(\mathbf{a})$, the $\Sigma_{B_{\mathsf{out}}}$ symbol in $w_{(\mathbf{b},i)}$, which we denote by $\left(w_{(\mathbf{b},i)}\right)_{\mathbf{b}}$, is equal to $\sigma_{\to\mathbf{b}}$,

- for every $t \in \Gamma_{\mathsf{out,dec}}(\mathbf{a})$ and $(\mathbf{b},i) \in \Gamma_{\mathsf{disc,out}}(\mathbf{a})$, the symbol corresponding to t in $w_{(\mathbf{b},i)}$, which we denote by $\left(w_{(\mathbf{b},i)}\right)_t$, is equal to $\sigma_{\rightsquigarrow t}$.

In words, the above two conditions mean that all of the symbols in w are consistent with a single $\sigma \in \Sigma_{A_{\mathsf{out}}}$.

By the projection decision complexity of $\mathcal{D}_{\mathsf{out}}$ and the bound on $\mathsf{wt}(\mathbf{a})$, the language $\mathcal{L}_{\mathbf{a}}$ can be decided by a circuit of size $O(\mathsf{wt}(\mathbf{a}) + \mathsf{DecisionComp}_1(N)) = O(k_1(N,|\Sigma_0|) + \mathsf{DecisionComp}_1(N,|\Sigma_0|))$, and hence there exists a uniform bound

$$N' = O(k_1(N,|\Sigma_0|) + \mathsf{DecisionComp}_1(N,|\Sigma_0|)) \tag{53}$$

such that for *every* $\mathbf{a} \in A_{\mathsf{out}}$, membership in the language $\mathcal{L}_{\mathbf{a}}$ can be decided by a circuit of size *at most* N'. Henceforth, we fix

$$M = M(N',T') \quad \text{and} \quad K_2 = K_2(N',T'), \tag{54}$$

where $M(\cdot,\cdot)$ is the function related to the left vertices of the inner dPCP given by the theorem statement, and $K_2(\cdot,\cdot)$ is the function related to the complete decoding distribution of the inner dPCP given by the theorem statement.

Now, for each $\mathbf{a} \in A_{\mathsf{out}}$ we take $\mathcal{D}_{\mathsf{in,a}}$ to be the inner dPCP guaranteed by the theorem statement for the language $\mathcal{L}_{\mathbf{a}}$ with soundness error set to

$$\delta_{\mathsf{in}} := \varepsilon.$$

Note that we can use the same value M and K_2 here for every inner dPCP, $\mathcal{D}_{\mathsf{in,a}}$. This is because the inner dPCP assumption in the theorem statement, along with the fact that for all $\mathbf{a} \in A_{\mathsf{out}}$, the language $\mathcal{L}_{\mathbf{a}}$ is over strings of length at most $k_1(N,|\Sigma_0|)$ with an alphabet of size at most T', and membership in $\mathcal{L}_{\mathbf{a}}$ can be decided by a circuit of size at most N'.

For each $\mathbf{a} \in A_{\text{out}}$, we denote the parts of $\mathcal{D}_{\text{in},\mathbf{a}}$ as

$$\mathcal{D}_{\text{in},\mathbf{a}} = (A_{\text{in},\mathbf{a}} \cup B_{\text{in},\mathbf{a}}, E_{\text{in},\mathbf{a}}, \Sigma_{A_{\text{in},\mathbf{a}}}, \Sigma_{B_{\text{in},\mathbf{a}}}, \{\Phi_{\text{in},e}\}_{e \in E_{\text{in},\mathbf{a}}}, \{D^{\text{in},\mathbf{a}}_{(\boldsymbol{b},i)}\}_{(\boldsymbol{b},i) \in \Gamma_{\text{out}}(\mathbf{a})}, \{\mathcal{P}^{\text{in},\mathbf{a}}_{(\boldsymbol{b},i)}\}_{(\boldsymbol{b},i) \in \Gamma_{\text{out}}(\mathbf{a})}).$$

We have that the alphabet sizes are

$$|\Sigma_{A_{\text{in},\mathbf{a}}}| \leqslant F_2(N', T') \quad \text{and} \quad |\Sigma_{B_{\text{in},\mathbf{a}}}| = O_\varepsilon(1). \tag{55}$$

For, convenience, we will express $A_{\text{in},\mathbf{a}}$ as $\Gamma_{\text{disc,out}}(\mathbf{a}) \times [M]$, rather than $[\text{wt}(\mathbf{a})] \times [M]$, where recall that $|\Gamma_{\text{disc,out}}(\mathbf{a})| = \text{wt}(\mathbf{a})$. Also, for simplicity, we refer to members of $\Gamma_{\text{disc,out}}(\mathbf{a}) \times [M]$ as $(\boldsymbol{b}, i, r)$ (rather than $((\boldsymbol{b}, i), r)$).

We will also use the following simplifying notation, which is similar to that in Definition 9.2, but now specified for the inner dPCP.

Definition 9.3. *Given a symbol $\sigma \in \Sigma_{A_{\text{in},a}}$ for $(\boldsymbol{b}, i, r) \in A_{\text{in},a}$, recall that $D^{\text{in},a}_{(\boldsymbol{b},i)}((\boldsymbol{b}, i, r), \sigma)$ outputs a $\Sigma_{B_{\text{out}}}$ symbol for $\boldsymbol{b}$ and one Σ_0 symbol for each $t \in \Gamma_{\text{out,dec}}(\boldsymbol{a})$. We let $\left(\sigma_{\leadsto(\boldsymbol{b},i)}\right)_{\boldsymbol{b}}$ denote the $\Sigma_{B_{\text{out}}}$ symbol part of the output from $D^{\text{in},a}_{(\boldsymbol{b},i)}((\boldsymbol{b}, i, r), \sigma)$. Likewise, for each $t \in \Gamma_{\text{dec,out}}(\boldsymbol{a})$, we write $\left(\sigma_{\leadsto(\boldsymbol{b},i)}\right)_t$ to denote the Σ_0 symbol for t in $D^{\text{in},a}_{(\boldsymbol{b},i)}((\boldsymbol{b}, i, r), \sigma)$. When we use this notation, the identity of $r \in [M]$ will be clear from context.*

By design of the language, the output of the decoder, $D^{\text{in},\mathbf{a}}_{(\boldsymbol{b},i)}$ consists of one $\Sigma_{B_{\text{out}}}$ symbol for $\mathbf{b}$ and one Σ_0 symbol for each $t \in \Gamma_{\text{dec}}(\mathbf{a})$. The list-decoding soundness of $D^{\text{in},\mathbf{a}}_{(\boldsymbol{b},i)}$ translates to the following claim, which morally states that one can view $\mathcal{D}_{\text{in},\mathbf{a}}$ as a decodable PCP for the language of valid $\mathbf{a}$-alphabet symbols, with $(\mathsf{L}_{\text{in}}, \delta_{\text{in}})$-list decoding soundness.

Claim 9.4. *For every $\boldsymbol{a} \in A_{\text{out}}$ the following holds. For any left assignment $T_{A_{\text{in},a}} : \Gamma_{\text{disc,out}}(\boldsymbol{a}) \times [M] \to \Sigma_{A_{\text{in},a}}$ to $\mathcal{D}_{\text{in},\mathbf{a}}$, there is a list $\{\sigma^a_1, \ldots, \sigma^a_{\mathsf{L}_{\text{in}}}\} \subseteq \Sigma_{\boldsymbol{a}}$ such that for any right assignment $T_{B_{\text{in},a}} : B_{\text{in},a} \to \Sigma_{B_{\text{in},a}}$*

$$\Pr_{\boldsymbol{b} \sim \mathcal{Q}_{\text{out}}(\circ, \boldsymbol{a}, \cdot), i \in [\text{wt}(\boldsymbol{a}, \boldsymbol{b})], r \in [M], b \sim \mathcal{P}^{\text{in},a}_{(\boldsymbol{b},i)}((\boldsymbol{b},i,r), \cdot)} [\mathcal{E}_1 \wedge \mathcal{E}_2] \leqslant \delta_{\text{in}}.$$

Here $\mathcal{E}_1$ and $\mathcal{E}_2$ are the following events over the sampled $(\boldsymbol{b}, i) \in \Gamma_{\text{out,dec}}(\boldsymbol{a}), r \in [M], b \in B_{\text{in},\mathbf{a}}$.

- *$\mathcal{E}_1$. the constraint on $(\boldsymbol{b}, i, r)$ and b is satisfied by $T_{A_{\text{in},a}}[(\boldsymbol{b}, i, r)]$ and $T_{B_{\text{in},a}}[b]$.*

- *$\mathcal{E}_2$. the symbols for $(\boldsymbol{b}, i)$ and $t \in \Gamma_{\text{out,dec}}(\boldsymbol{a})$ output by the decoder are not (simultaneously) equal to $\left(\sigma^a_j\right)_{\to \boldsymbol{b}}$ and $\left(\sigma^a_j\right)_{\leadsto t}$ for any $j \in [\mathsf{L}_{\text{in}}]$. That is, for each $j \in [\mathsf{L}_{\text{in}}]$, either*

$$\left(D^{\text{in},a}_{(\boldsymbol{b},i)}((\boldsymbol{b}, i, r), T_{A_{\text{in},a}}[(\boldsymbol{b}, i, r)])\right)_{\boldsymbol{b}} \neq \left(\sigma^a_j\right)_{\to \boldsymbol{b}}$$

or there is $t \in \Gamma_{\text{out,dec}}(\boldsymbol{a})$ such that

$$\left(D^{\text{in},a}_{(\boldsymbol{b},i)}((\boldsymbol{b}, i, r), T_{A_{\text{in},a}}[(\boldsymbol{b}, i, r)])\right)_t \neq \left(\sigma^a_j\right)_{\leadsto t}.$$

Proof. Fix $T_{A_{\text{in},\mathbf{a}}}$. Applying the list-decoding soundness of $\mathcal{D}_{\text{in},\mathbf{a}}$, we get a list of words $\{w_1, \ldots, w_{\mathsf{L}_{\text{in}}}\} \subseteq \mathcal{L}_{\mathbf{a}}$ such that for any right assignment $T_{B_{\text{in},\mathbf{a}}}$, the list-decoding error of $T_{A_{\text{in},\mathbf{a}}}$ and $T_{B_{\text{in},\mathbf{a}}}$ relative to this list is at most δ_{in}. We can translate this list, $\{w_1, \ldots, w_{\mathsf{L}_{\text{in}}}\}$, into a corresponding list of L_{in} symbols $\sigma^{\mathbf{a}}_1, \ldots, \sigma^{\mathbf{a}}_{\mathsf{L}_{\text{in}}} \in \Sigma_{\mathbf{a}}$. Specifically, for each $w_\ell \in \{w_1, \ldots, w_{\mathsf{L}_{\text{in}}}\}$, by definition of being in $\mathcal{L}_{\mathbf{a}}$, we have that there exists $\sigma \in \Sigma_{\mathbf{a}}$ such that for any $(\mathbf{b}, i) \in \Gamma_{\text{disc,out}}(\mathbf{a})$ and $t \in \Gamma_{\text{out,dec}}(\mathbf{a})$

$$(w_{\mathbf{b},i})_{\mathbf{b}} = \sigma_{\to \mathbf{b}} \quad \text{and} \quad (w_{\mathbf{b},i})_t = \sigma_{\leadsto t}.$$

We set $\sigma^{\mathbf{a}}_\ell$ to be this symbol and if for some $w \in \{w_1, \ldots, w_{\mathsf{L}_{\text{in}}}\}$ there exist multiple such $\sigma \in \Sigma_{\mathbf{a}}$, we choose one of them arbitrarily to be $\sigma^{\mathbf{a}}_\ell$. Now it is straightforward to see that the list-decoding soundness of $\mathcal{D}_{\text{in},\mathbf{a}}$ for the language $\mathcal{L}_{\mathbf{a}}$ relative to the list $\{w_1, \ldots, w_{\mathsf{L}_{\text{in}}}\}$ translates exactly into the statement of the claim with the list $\{\sigma^{\mathbf{a}}_1, \ldots, \sigma^{\mathbf{a}}_{\mathsf{L}_{\text{in}}}\} \subseteq \Sigma_{\mathbf{a}}$. Here, we use the fact that sampling $(\mathbf{b}, i)$ in the multiset, $\Gamma_{\text{disc,out}}(\mathbf{a})$, uniformly, we have $\mathbf{b} \sim \mathcal{Q}_{\text{out}}(\circ, \mathbf{a}, \cdot)$ and i uniform in $[\text{wt}(\mathbf{a}, \mathbf{b})]$. $\square$

Finally, let us write the complete decoding distribution of each inner dPCP, $\mathcal{D}_{\text{in},\mathbf{a}}$, as $\mathcal{Q}_{\text{in},\mathbf{a}}$. Using the discreteness property of the inner dPCP's complete decoding distribution, we have that for every $((\mathbf{b}, i, r), b) \in A_{\text{in},\mathbf{a}} \times B_{\text{in},\mathbf{a}}$, there is $\text{wt}'_{\mathbf{a}}((\mathbf{b}, i, r), b)$ such that

$$\mathcal{Q}_{\text{in},\mathbf{a}}(\circ, (\mathbf{b}, i, r), b) = \mathcal{Q}_{\text{in},\mathbf{a}}((\mathbf{b}, i), (\mathbf{b}, i, r), b) = \frac{\text{wt}'_{\mathbf{a}}((\mathbf{b}, i, r), b)}{\text{wt}(\mathbf{a}) \cdot M \cdot K_2}. \tag{56}$$

By the third property of the complete decoding distribution of the inner dPCP, for all $\mathbf{a} \in A_{\text{out}}$ and $(\mathbf{b}, i, r) \in A_{\text{in},\mathbf{a}}$ we have

$$\sum_{b \in \Gamma_{\mathbf{a},\text{in}}((\mathbf{b},i,r))} \text{wt}'_{\mathbf{a}}((\mathbf{b}, i, r), b) = K_2, \tag{57}$$

where in the above $\Gamma_{\mathbf{a},\text{in}}$ denotes the neighborhood of a vertex in $\mathcal{D}_{\text{in},\mathbf{a}}$. We emphasize that, by assumption, K_2 above is the same for all of the inner dPCPs, $\mathcal{D}_{\text{in},\mathbf{a}}$. We are ready to describe our composed dPCP.

The composed dPCP $\mathcal{D}_{\text{out}} \circ \mathcal{D}_{\text{in}}$.

- **Constraint Graph:** The constraint graph is bipartite with the following left and right sides respectively:

$$A_{\text{comp}} := \{(\mathbf{b}, r) \ : \ r \in [M] \text{ and } \mathbf{b} \in B_{\text{out}}\},$$
$$B_{\text{comp}} := \{(\mathbf{a}; b) \ : \ \mathbf{a} \in A_{\text{out}}, b \in B_{\text{in},\mathbf{a}}\} = \bigcup_{\mathbf{a} \in A_{\text{out}}} \{\mathbf{a}\} \times B_{\text{in},\mathbf{a}}.$$

Let us explain how the vertices in the composed PCP should be thought of. First, take the inner constraint graphs, $(A_{\text{in},\mathbf{a}} \cup B_{\text{in},\mathbf{a}}, E_{\text{in},\mathbf{a}})$, for each $\mathbf{a} \in A_{\text{out}}$. Viewing each left side as $\Gamma_{\text{disc,out}}(\mathbf{a}) \times [M]$, we see that for each $\mathbf{b} \in B_{\text{out}}$ and $r \in [M]$, there are d_1-many vertices of the form $(\mathbf{b}, i, r)$ which appear in A_{comp}. This is because of the assumption that

$$\sum_{\mathbf{a} \in A_{\text{out}}} \text{wt}(\mathbf{a}, \mathbf{b}) = d_1, \tag{58}$$

from the outer dPCP's complete decoding distribution. Henceforth, when we are in the context of the composed dPCP, we will denote the copy of $(\mathbf{b}, i, r)$ in $A_{\text{in},\mathbf{a}}$ by $(\mathbf{a}; \mathbf{b}, i, r)$.

The left side of the composed constraint graph is obtained by identifying, for each $\mathbf{b} \in B_{\text{out}}$ and $r \in [M]$, the vertices of $(\mathbf{a}; \mathbf{b}, i, r)$ over all $\mathbf{a}$ adjacent to $\mathbf{b}$ and $i \in [\text{wt}(\mathbf{a}, \mathbf{b})]$. One can think of the vertex $(\mathbf{b}, r) \in A_{\text{comp}}$ as a cloud consisting of all $(\mathbf{a}; \mathbf{b}, i, r)$ appearing over all copies of the inner dPCPs $\mathcal{D}_{\text{in},\mathbf{a}}$.

The right side of the composed constraint graph is the union of the right sides of the inner constraint graph, and the edges are the same after identification.

- **Left Alphabet:** Fix a vertex $(\mathbf{b}, r) \in A_{\text{comp}}$. For each $\mathbf{a} \in \Gamma_{\text{out}}(\mathbf{b})$ and $i \in [\text{wt}(\mathbf{a}, \mathbf{b})]$, there is a vertex $(\mathbf{a}; \mathbf{b}, i, r)$ coming from the inner dPCP of $\mathcal{D}_{\text{in},\mathbf{a}}$. Then, the composed vertex $(\mathbf{b}, r)$ holds a symbol from $\Sigma_{A_{\text{in},\mathbf{a}},\mathbf{a}}$ for each $(\mathbf{a}; \mathbf{b}, i, r)$. Overall, using (58) see that these are d_1 inner left alphabet symbols.

If σ is an alphabet symbol for $(\mathbf{b}, r)$, we use $\sigma\{\mathbf{a}, i\}$ to denote the $\Sigma_{A_{\text{in},\mathbf{a}}}$ symbol corresponding to a label for $(\mathbf{a}; \mathbf{b}, i, r)$. We also constrain the alphabet of $(\mathbf{b}, r)$ to only contain σ as follows.

   - **Hardcoding Outer Constraints.** We require that $(\sigma\{\mathbf{a}, i\})_{\rightsquigarrow \mathbf{b}}$ is the same for all $\mathbf{a} \in \Gamma_{\text{disc}}(\mathbf{b})$ and $i \in [\text{wt}(\mathbf{a}, \mathbf{b})]$. Henceforth, we denote this value as $\sigma_{\rightsquigarrow \mathbf{b}}$ so that

$$\sigma_{\rightsquigarrow \mathbf{b}} = \left(D^{\text{in},\mathbf{a}}_{(\mathbf{b},i)}((\mathbf{a}; \mathbf{b}, i, r), \sigma\{\mathbf{a}, i\}) \right)_{\mathbf{b}}. \tag{59}$$

On the right side, we are first running the inner decoder from $\mathcal{D}_{\text{in},\mathbf{a}}$, where the index to decode is $(\mathbf{b}, i)$, the left vertex is $(\mathbf{a}; \mathbf{b}, i, r) \in A_{\text{in},\mathbf{a}}$, and the symbol is $\sigma\{\mathbf{a}, i\}$. We then take the symbol for $\mathbf{b}$ given by the output symbol. We note that whenever we use the notation of (59), the symbol σ will satisfy the hardcoding outer constraints, so all the possible $\mathbf{a}$ lead to the same value.

66

– **Hardcoding Decoding Consistency.** For any $t \in [n]$ and all $\mathbf{a}, i$ such that $(\mathbf{a}, \mathbf{b}) \in \mathrm{supp}(\mathcal{P}_t)$ and $i \in [\mathrm{wt}(\mathbf{a}, \mathbf{b})]$, we require that the decoded values $\left(D^{\mathrm{in},\mathbf{a}}_{\mathbf{b}}((\mathbf{a}; \mathbf{b}, i, r), \sigma\{\mathbf{a}, i\}) \right)_t$ are the same, and we denote this value by $\sigma_{\leadsto t}$.

We denote the left alphabet by $\Sigma_{A_{\mathrm{comp}}}$.

- **Right Alphabet:** The right alphabet is $\Sigma_{B_{\mathrm{in},\mathbf{a}}}$.

- **Constraints:** By our construction, $((\mathbf{b}, r), (\mathbf{a}; b)) \in E_{\mathrm{comp}}$ if and only if there is an i such that $(\mathbf{b}, i) \in \Gamma_{\mathrm{disc,out}}(\mathbf{a})$ and, in $\mathcal{D}_{\mathrm{in},\mathbf{a}}$, the vertex $(\mathbf{a}; \mathbf{b}, i, r)$ is adjacent to $(\mathbf{a}; b)$. Let us describe the constraint on this edge now.

  Given valid labels $\sigma \in \Sigma_{A_{\mathrm{comp}}}$ and $\sigma' \in \Sigma_{B_{\mathrm{in},\mathbf{a}}}$ for $(\mathbf{b}, r)$ and $(\mathbf{a}; b)$ respectively in the composed dPCP, the constraint on $((\mathbf{b}, r), (\mathbf{a}; b))$ checks that, for all such $\mathbf{a}, i$ as above, where $(\mathbf{b}, i, r)$ is adjacent to b in $\mathcal{D}_{\mathrm{in},\mathbf{a}}$, the inner constraint of $\mathcal{D}_{\mathrm{in},\mathbf{a}}$, between $(\mathbf{a}; \mathbf{b}, i, r)$ and $(\mathbf{a}; b)$ is satisfied by the labels $\sigma\{\mathbf{a}, i\} \in \Sigma_{A_{\mathrm{in},\mathbf{a}}}$ and $\sigma' \in \Sigma_{B_{\mathrm{in},\mathbf{a}}}$. That is,
  $$\Phi_{((\mathbf{b},r),(\mathbf{a};b))}(\sigma, \sigma') = 1$$
  if and only if
  $$\Phi_{\mathrm{in},((\mathbf{a};\mathbf{b},i,r),(\mathbf{a};b))}(\sigma, \sigma') = 1$$
  over all $\mathbf{a}, i$ such that $(\mathbf{a}; \mathbf{b}, i, r)$ is adjacent to $(\mathbf{a}; b)$ in $\mathcal{D}_{\mathrm{in},\mathbf{a}}$. In the second equality, the constraint is coming from $\mathcal{D}_{\mathrm{in},\mathbf{a}}$.

- **Decoding Distributions:** Let $\mathcal{Q}_{\mathrm{out}}$ be the complete decoding distribution of the outer dPCP. Then the complete decoding distribution of the composed dPCP, $\mathcal{Q}_{\mathrm{comp}}$, is generated as follows.

  – Choose $(t, \mathbf{a}, \mathbf{b}) \sim \mathcal{Q}_{\mathrm{out}}$, and $i \in [\mathrm{wt}(\mathbf{a}, \mathbf{b})]$ uniformly at random.
  – Sample $r \in [M]$ uniformly.
  – Choose $(\mathbf{a}; b) \sim \mathcal{Q}_{\mathrm{in},\mathbf{a}}((\mathbf{b}, i), (\mathbf{a}; \mathbf{b}, i, r), \cdot)$,
  – Output $(t, (\mathbf{b}, r), (\mathbf{a}; b))$.

  It is clear that the marginal of $\mathcal{Q}_{\mathrm{comp}}$ over $[n]$ is uniform, and for each $t \in [n]$ we take the decoding distribution to be $\mathcal{P}^{\mathrm{comp}}_t = \mathcal{Q}_{\mathrm{comp}}(t, \cdot, \cdot)$.

- **Decoder.** For $(\mathbf{b}, r) \in A_{\mathrm{comp}}$ and label σ, the decoder outputs $D^{\mathrm{comp}}_t((\mathbf{b}, r), \sigma) = \sigma_{\leadsto t}$.

This completes the description of the composed dPCP, $\mathcal{D}_{\mathrm{out}} \circ \mathcal{D}_{\mathrm{in}}$, and we will now show that it satisfies the requirements of Theorem 9.1.

9.2 Basic Properties of the Composed PCP

In this section we establish all of the properties of $\mathcal{D}_{\mathrm{out}} \circ \mathcal{D}_{\mathrm{in}}$ as promised in the statement of Theorem 9.1, except for the list-decoding soundness, which is deferred to the next section. In the below, recall that T' is given by (52), N' is given by (53), K_2 and M are given by (54).

Length. By construction, left side has size at most $|B_{\mathrm{out}}| \cdot M$ vertices and the right side has size $\sum_{\mathbf{a} \in A_{\mathrm{out}}} |B_{\mathrm{in},\mathbf{a}}|$, so overall the length is
$$S_1(N, |\Sigma_0|) \cdot M(N', T') + S_1(N, |\Sigma_0|) \cdot S_2(N', T').$$

Alphabet Sizes. The left alphabet size is at most $|\Sigma_{A_{\mathrm{in},\mathbf{a}}}|^{d_1}$. Indeed, fix a left vertex $(\mathbf{b}, r)$. Then for every $\mathbf{a} \in A_{\mathrm{out}}$ and $i \in [\mathrm{wt}(\mathbf{a}, \mathbf{b})]$, $(\mathbf{b}, r)$ holds one symbol from $\Sigma_{A_{\mathrm{in},\mathbf{a}}}$, corresponding to a label to $(\mathbf{a}; \mathbf{b}, i, r)$ in $\mathcal{D}_{\mathrm{in},\mathbf{a}}$, which by (58) is a total of d_1 symbols from $\Sigma_{A_{\mathrm{in},\mathbf{a}}}$. Hence the alphabet size is $F_2(N', T')^{d_1}$, where we use (55) to bound the size of $|\Sigma_{A_{\mathrm{in},\mathbf{a}}}|$. The right alphabet is $\Sigma_{B_{\mathrm{in},\mathbf{a}}}$ and its size is $O_\varepsilon(1)$ by (55).

Decoding Degree. The decoding degree is $O_\varepsilon(d_1)$. Each $(\mathbf{b}, r)$ is responsible for decoding t if there is $\mathbf{a} \in \Gamma_{\mathsf{out}}(\mathbf{b})$ such that $t \in \Gamma_{\mathsf{out,dec}}(\mathbf{a})$. The decoding degree bound follows from the fact that $|\Gamma_{\mathsf{out,dec}}(\mathbf{a})| \leqslant O_\varepsilon(1)$ and that $|\Gamma_{\mathsf{out}}(\mathbf{b})| \leqslant d_1$ by (58).

Projection Constraints. It is straightforward to check that the composed dPCP has projection constraints because the inner dPCPs are all projection dPCPs.

Projection Decision Complexity. Fix a constraint $((\mathbf{b}, r), (\mathbf{a}; b))$ and let σ be the left alphabet symbol. Recall that σ contains, as its restrictions, one $\Sigma_{A_{\mathsf{in},\mathbf{a}}}$ symbol for each of d_1 vertices corresponding to all $\mathbf{a}, i$ such that $\mathbf{a} \in \Gamma_{\mathsf{disc}}(\mathbf{b}), i \in [\mathsf{wt}(\mathbf{a}, \mathbf{b})]$.

The projection circuit needs to do the following: (1) check that the left alphabet symbol satisfies hardcoding of outer constraints, (2) check that the left alphabet symbol satisfies hardcoding decoding consistency, and (3) compute the unique right alphabet symbol that satisfies the constraint.

Each of (1) and (2) can be done by computing a restriction of the given left alphabet symbol and then running the inner decoder d_1 times and then checking an equality between d_1 symbols of either $\Sigma_{B_{\mathsf{out}}}$, in the case of (1), or Σ_0 in the case of (2). Altogether, the circuit complexity of (1) and (2) is a constant multiple of

$$\log(F_1(N, |\Sigma_0|)) + d_1 \cdot (\mathsf{DecodeComp}_2(N', T') + \log|\Sigma_{B_{\mathsf{out}}}| + \log|\Sigma_0|)$$
$$= \log(F_1(N, |\Sigma_0|)) + O_{\varepsilon, |\Sigma_0|}(\mathsf{DecodeComp}_2(N', T')).$$

To accomplish (3), the projection circuit runs the projection circuit of the inner dPCP, $\mathcal{D}_{\mathsf{in},\mathbf{a}}$, on a restriction of the given left alphabet symbol which requires circuit complexity

$$\log(F_1(N, |\Sigma_0|)) + \mathsf{DecisionComp}_2(N', T').$$

Overall this gives projection decision complexity that is,

$$O(\log(F_1(N, |\Sigma_0|))) + O_{\varepsilon, |\Sigma_0|}(\mathsf{DecisionComp}_2(N', T') + \mathsf{DecodeComp}_2(N', T')).$$

Decoding Complexity. The decoder restricts its input to obtain a symbol for the inner dPCP and runs the decoder of the inner dPCP. Hence, the decoding complexity is $\log(F_1(N, |\Sigma_0|)) + \mathsf{DecodeComp}_2(N', T')$.

Complete Decoding Distribution. Fix $(\mathbf{b}, r) \in A_{\mathsf{comp}}$ and $(\mathbf{a}; b) \in B_{\mathsf{comp}}$. Then

$$
\begin{aligned}
\mathcal{Q}_{\mathsf{comp}}(\circ, (\mathbf{b}, r), (\mathbf{a}; b)) &= \mathcal{Q}_{\mathsf{out}}(\circ, \mathbf{a}, \mathbf{b}) \cdot \frac{1}{M} \cdot \underset{i \in [\mathsf{wt}(\mathbf{a},\mathbf{b})]}{\mathbb{E}} \left[\frac{\mathcal{Q}_{\mathsf{in},\mathbf{a}}((\mathbf{b}, i), (\mathbf{a}; \mathbf{b}, i, r), (\mathbf{a}; b))}{\mathcal{Q}_{\mathsf{in},\mathbf{a}}((\mathbf{b}, i), (\mathbf{a}; \mathbf{b}, i, r), \circ)} \right] \\
&= \mathcal{Q}_{\mathsf{out}}(\circ, \mathbf{a}, \mathbf{b}) \cdot \frac{1}{M} \cdot \underset{i \in [\mathsf{wt}(\mathbf{a},\mathbf{b})]}{\mathbb{E}} \left[\frac{\mathsf{wt}'_{\mathbf{a}}((\mathbf{a}; \mathbf{b}, i, r), (\mathbf{a}; b))}{K_2} \right] \\
&= \frac{\mathsf{wt}(\mathbf{a}, \mathbf{b})}{K_1 M K_2} \underset{i \in [\mathsf{wt}(\mathbf{a},\mathbf{b})]}{\mathbb{E}} [\mathsf{wt}'_{\mathbf{a}}((\mathbf{a}; \mathbf{b}, i, r), (\mathbf{a}; b))] \\
&= \frac{\sum_{i \in [\mathsf{wt}(\mathbf{a},\mathbf{b})]} \mathsf{wt}'_{\mathbf{a}}((\mathbf{a}; \mathbf{b}, i, r), (\mathbf{a}; b))}{K_1 M K_2}
\end{aligned}
\tag{60}
$$

In the first transition, we are using the definition of $\mathcal{Q}_{\mathsf{comp}}$. In the second transition, we are using (56) to plug in the term inside the expectation, and then (57). In the third transition, we are writing out $\mathcal{Q}_{\mathsf{out}}(\circ, \mathbf{a}, \mathbf{b})$. In the final transition, we are expanding out the expectation from the second line.

By the above, we get that the first property of the complete decoding distribution holds with discreteness parameter

$$J = K_1 \cdot M \cdot K_2 \quad \text{and} \quad \mathsf{wt}''((\mathbf{b}, r), (\mathbf{a}; b)) = \sum_{i \in [\mathsf{wt}(\mathbf{a},\mathbf{b})]} \mathsf{wt}'_{\mathbf{a}}((\mathbf{a}; \mathbf{b}, i, r), (\mathbf{a}; b)). \tag{61}$$

We move onto the second property which requires showing that the marginal over A_{comp} is uniform. From the description, it is clear that

$$\mathcal{Q}_{\mathsf{comp}}(\circ, (\mathbf{b}, r), \circ) = \mathcal{Q}_{\mathsf{out}}(\circ, \circ, \mathbf{b}) \cdot \frac{1}{M} = \frac{d_1}{K_1(N, |\Sigma_0|)M}.$$

68

In the second equality, we use the fact that $\mathcal{Q}_{\text{out}}(\circ, \circ, \mathbf{b}) = \frac{d_1}{K_1(N,|\Sigma_0|)}$, by the right uniformity property of the outer dPCP's complete decoding distribution. It follows that for any $(\mathbf{b}, r) \in A_{\text{comp}}$

$$\sum_{(\mathbf{a};b)\in B_{\text{comp}}} \text{wt}''((\mathbf{b}, r), (\mathbf{a}; b)) = \frac{Jd_1}{K_1(N,|\Sigma_0|)M} = d_1 K_2,$$

as desired.

Completeness. Since the outer dPCP $\mathcal{D}_{\text{out}}$ has perfect completeness, for any $w \in \text{SAT}(\varphi)$ there are assignments $T_{A_{\text{out}}} : A_{\text{out}} \to \Sigma_{A_{\text{out}}}$ and $T_{B_{\text{out}}} : B_{\text{out}} \to B_{\text{out}}$ such that all of the constraints in $\mathcal{D}_{\text{out}}$ are satisfied and the decoder outputs w_t with probability 1 for all $t \in [n]$. Now, for each $\mathbf{a} \in A_{\text{out}}$, using the value $T_{A_{\text{out}}}[\mathbf{a}]$, one can construct $w_{\text{in},\mathbf{a}} \in (\Sigma_{B_{\text{out}}} \times \Sigma_0^{q_1})^{\text{wt}(\mathbf{a})}$ such that for every $(\mathbf{b}, i) \in \Gamma_{\text{disc,out}}(\mathbf{a})$ and $t \in \Gamma_{\text{out,dec}}(\mathbf{a})$, we have that the $\Sigma_{B_{\text{out}}}$ part of $(w_{\text{in},\mathbf{a}})_{(\mathbf{b},i)}$ equals $(T_{A_{\text{out}}}[\mathbf{a}])_{\to\mathbf{b}}$, and the symbols corresponding to $t \in \Gamma_{\text{out,dec}}(\mathbf{a})$ in $(w_{\text{in},\mathbf{a}})_\mathbf{b}$ are consistent $(T_{A_{\text{out}}}[\mathbf{a}])_{\leadsto t}$. For each such $w_{\text{in},\mathbf{a}}$, which is in $\mathcal{L}_\mathbf{a}$, one can then use the perfect completeness of the inner dPCP's to construct $T_{A_{\text{in},\mathbf{a}}} : A_{\text{in},\mathbf{a}} \to \Sigma_{A_{\text{in},\mathbf{a}}}$ and $T_{B_{\text{in},\mathbf{a}}} : B_{\text{in},\mathbf{a}} \to B_{\text{in},\mathbf{a}}$ which satisfy all of the constraints on the inner dPCP $\mathcal{D}_{\text{in},\mathbf{a}}$ and always output a decoding that is consistent with $(w_{\text{in},\mathbf{a}})_{\to\mathbf{b}}$ for all $(\mathbf{b}, i) \in \Gamma_{\text{disc,out}}(\mathbf{a})$ and $(w_{\text{in},\mathbf{a}})_{\leadsto t}$ for all $t \in \Gamma_{\text{out,dec}}(\mathbf{a})$.

From these assignments, we construct the composed assignments $T_{A_{\text{comp}}} : A_{\text{comp}} \to \Sigma_{A_{\text{comp}}}$ and $T_{B_{\text{comp}}} : B_{\text{comp}} \to \Sigma_{B_{\text{comp}}}$ as follows:

- For every $(\mathbf{a}; b) \in A_{\text{comp}}$, note that b is a right vertex in $\mathcal{D}_{\text{in},\mathbf{a}}$, and hence we set $T_{B_{\text{comp}}}[(\mathbf{a}; b)] = T_{B_{\text{in},\mathbf{a}}}[b]$.

- For each $(\mathbf{b}, r) \in A_{\text{comp}}$, $T_{A_{\text{comp}}}[(\mathbf{b}, r)]$ consists of the label $T_{A_{\text{in},\mathbf{a}}}[(\mathbf{b}, i, r)]$ for each $\mathbf{a} \in \Gamma_{\text{out}}(\mathbf{b})$ and $i \in [\text{wt}(\mathbf{a}, \mathbf{b})]$.

One can check that this assignment indeed satisfies all of the constraints of the composed dPCP and the decoder outputs w_t with probability 1 for all $t \in [n]$.

9.3 Proof of List-Decoding Soundness

This section is devoted to the proof of the following lemma, establishing the list-decoding soundness of Theorem 9.1 and thereby completing its proof.

Lemma 9.5. *The composed dPCP constructed above has decoding error* $\delta_{\text{in}} + \mathsf{L}_{\text{in}} \cdot \delta_{\text{out}} = \varepsilon + \varepsilon^{10}$ *relative to a list of size* $\mathsf{L}_{\text{in}} \cdot \mathsf{L}_{\text{out}} = \varepsilon^{C_1 + C_2 + 10}$, *and hence satisfies* $(\text{poly}(1/\varepsilon), O(\varepsilon))$-*list-decoding soundness.*

Proof. Fix an assignment. $T_{A_{\text{comp}}} : A_{\text{comp}} \to \Sigma_{A_{\text{comp}}}$ We will first define a list of assignments for $\mathcal{D}_{\text{out}}$. Specifically, we will have a left outer assignment, $T_{A_{\text{out}}}^{(j)} : A_{\text{out}} \to \Sigma_{A_{\text{out}}}$, for each $j \in [\mathsf{L}_{\text{in}}]$.

Obtaining assignments to the left side of the Outer PCP. For each $\mathbf{a} \in A_{\text{out}}$, we will attempt to obtain candidate assignments to $\mathbf{a}$ in the outer dPCP by using the list-decoded values of the inner dPCP, $\mathcal{D}_{\text{in},\mathbf{a}}$. To do so, we first use $T_{A_{\text{comp}}}$ to obtain an assignment $T_{A_{\text{in},\mathbf{a}}}$ for $\mathcal{D}_{\text{in},\mathbf{a}}$, and subsequently apply the list-decoding condition of $\mathcal{D}_{\text{in},\mathbf{a}}$ to these assignments. Set

$$T_{A_{\text{in},\mathbf{a}}}[(\mathbf{a}; \mathbf{b}, i, r)] := T_{A_{\text{comp}}}[(\mathbf{b}, r)]\{\mathbf{a}, i\}, \quad \forall (\mathbf{a}; \mathbf{b}, i, r) \in A_{\text{in},\mathbf{a}}.$$

Now that $T_{A_{\text{in},\mathbf{a}}}$ is an assignment to $\mathcal{D}_{\text{in},\mathbf{a}}$, we can apply the list-decoding guarantee from Claim 9.4. We get that there is a list $\text{List}_{\text{in},\mathbf{a}} = \{y_1^\mathbf{a}, \ldots, y_{\mathsf{L}_{\text{in}}}^\mathbf{a}\} \subseteq \Sigma_\mathbf{a}$ such that the following holds for any right assignment $T' : B_{\text{in},\mathbf{a}} \to \Sigma_{B_{\text{in},\mathbf{a}}}$,

$$\Pr_{(\mathbf{b},i)\in\Gamma_{\text{disc,out}}(\mathbf{a}), r\in[M], (\mathbf{a};b)\sim\mathcal{P}_{(\mathbf{b},i)}^{\text{in},\mathbf{a}}((\mathbf{a};\mathbf{b},i,r),\cdot)} [\mathcal{E}_{\text{in},1} \wedge \mathcal{E}_{\text{in},2}] \leqslant \delta_{\text{in}}. \tag{62}$$

Here, $\mathcal{E}_{\text{in},1}$ and $\mathcal{E}_{\text{in},2}$ are the following events:

- $\mathcal{E}_{\text{in},1}$ (the inner constraint is satisfied): this is the event that $T_{A_{\text{in},\mathbf{a}}}[(\mathbf{a}; \mathbf{b}, i, r)]$ and $T'[(\mathbf{a}; b)]$ satisfy the constraint on $((\mathbf{a}; \mathbf{b}, i, r), (\mathbf{a}, b))$ from $\mathcal{D}_{\text{in},\mathbf{a}}$.

- $\mathcal{E}_{\text{in},2}$ (the inner decoding fails): this is the event that the inner decoder does not output something consistent with the list. That is, the symbols for $\mathbf{b}$ and $t \in \Gamma_{\text{out,dec}}(\mathbf{a})$ output by the decoder are not (simultaneously) equal to $\left(y_j^{\mathbf{a}}\right)_{\to \mathbf{b}}$ and $\left(y_j^{\mathbf{a}}\right)_{\rightsquigarrow t}$ for any $j \in [\mathsf{L}_{\text{in}}]$.

Using $\mathsf{List}_{\text{in},\mathbf{a}}$ we can now define L_{in} assignments to A_{out}. More specifically, for each $j \in [\mathsf{L}_{\text{in}}]$, define the assignment $T_{A_{\text{out}}}^{(j)} : A_{\text{out}} \to \Sigma_{A_{\text{out}}}$ by

$$T_{A_{\text{out}}}^{(j)}[\mathbf{a}] = y_j^{\mathbf{a}}, \quad \forall \mathbf{a} \in A_{\text{out}}.$$

Outer List decodings for each $T_{A_{\text{out}}}^{(j)}$. Fix a $j \in [\mathsf{L}_{\text{in}}]$ and consider one of the left outer assignments $T_{A_{\text{out}}}^{(j)}$. By the list-decoding condition of $\mathcal{D}_{\text{out}}$, there exists a list $\mathsf{List}_{\text{out}}^{(j)} = \{w_1^{(j)}, \ldots, w_{\mathsf{L}_{\text{out}}}^{(j)}\} \subseteq \mathcal{L}$ such that for any assignment to the right side of the outer dPCP, $T_{B_{\text{out}}} : B_{\text{out}} \to \Sigma_{B_{\text{out}}}$, the list-decoding error relative to $\mathsf{List}_{\text{out}}^{(j)}$ is bounded by δ_{out}. Specifically, we have that for each $j \in [\mathsf{L}_{\text{in}}]$ and any right assignment $T_{B_{\text{out}}}$,

$$\Pr_{\substack{t \in [n] \\ (\mathbf{a},\mathbf{b}) \sim \mathcal{P}_t^{\text{out}}}} [\mathcal{E}_{\text{out},1} \wedge \mathcal{E}_{\text{out},2}] \leqslant \delta_{\text{out}}, \tag{63}$$

where the events above are as follows:

- $\mathcal{E}_{\text{out},1}$ (satisfied outer constraint): this is the event that $\Phi_{\text{out},(\mathbf{a},\mathbf{b})}\left(T_{A_{\text{out}}}^{(j)}[\mathbf{a}], T_{B_{\text{out}}}[\mathbf{b}]\right) = 1$, meaning that the outer constraint is satisfied on $(\mathbf{a}, \mathbf{b})$.

- $\mathcal{E}_{\text{out},2}$ (the outer decoding fails): this is the event that $D_t^{\text{out}}(\mathbf{a}, T_{A_{\text{out}}}^{(j)}[\mathbf{a}]) \notin \left\{\left(w_\ell^{(j)}\right)_t\right\}_{\ell \in [\mathsf{L}_{\text{out}}]}$, meaning that the decoding does not agree with any member of $\mathsf{List}_{\text{out}}^{(j)}$.

Showing List-Decoding Soundness. We are now ready to show that the composed dPCP for $\mathcal{L}$ satisfies $(\mathsf{L}_{\text{out}} \cdot \mathsf{L}_{\text{in}}, \delta_{\text{in}} + \mathsf{L}_{\text{in}} \cdot \delta_{\text{out}})$-list-decoding soundness. The list-decoding soundness will be with respect to the following list from $\mathcal{L}$:

$$\mathsf{List}_{\text{comp}} := \bigcup_{j \in [\mathsf{L}_{\text{in}}]} \mathsf{List}_{\text{out}}^{(j)} = \{w_\ell^{(j)}\}_{j \in [\mathsf{L}_{\text{in}}], \ell \in [\mathsf{L}_{\text{out}}]}.$$

Fix any right assignment $T_{B_{\text{comp}}} : B_{\text{comp}} \to \Sigma_{B_{\text{in},\mathbf{a}}}$. For each $\mathbf{a} \in A_{\text{out}}$ define the following inner right assignment:

$$T_{B_{\text{in},\mathbf{a}}}[(\mathbf{a}; b)] := T_{B_{\text{comp}}}[(\mathbf{a}; b)], \quad \forall (\mathbf{a}; b) \in B_{\text{in},\mathbf{a}}.$$

Let us also define a candidate right assignments for the outer dPCP. For each $(\mathbf{b}, r) \in A_{\text{comp}}$, recall that $T_{A_{\text{comp}}}[(\mathbf{b}, r)]_{\mathbf{b}}$ gives an assignment to $\mathbf{b} \in B_{\text{out}}$. For each $r \in [M]$, recalling the notation from (59), we define a table $T_{B_{\text{out}}}^{(r)}$ as

$$T_{B_{\text{out}}}^{(r)}[\mathbf{b}] = T_{A_{\text{comp}}}[(\mathbf{b}, r)]_{\rightsquigarrow \mathbf{b}}.$$

Now, we define the following events:

- $\mathcal{E}_{\text{comp},1}$: this is the event that the constraint on $(\mathbf{b}, r)$ and $(\mathbf{a}; b)$ is satisfied in the composed dPCP:

$$\Phi_{\text{comp},((\mathbf{b},r),(\mathbf{a};b))}\left(T_{A_{\text{comp}}}[(\mathbf{b}, r)], T_{B_{\text{comp}}}[(\mathbf{a}; b)]\right) = 1.$$

Unpacking the definition of the constraint, this is also the event that the constraint on the edge $((\mathbf{b}, r), (\mathbf{a}; b))$ of the *inner* dPCP is satisfied by the induced assignments, $T_{A_{\text{in},\mathbf{a}}}[(\mathbf{a}; \mathbf{b}, i, r)]$ and $T_{B_{\text{in},\mathbf{a}}}[(\mathbf{a}; b)]$.

- $\mathcal{E}_{\text{comp},2}$ (the composed decoding is outside the list): is the event that

$$D_t^{\text{comp}}((\mathbf{b}, r), T_{A_{\text{comp}}}[(\mathbf{b}, r)]) \notin \left\{\left(w_\ell^{(j')}\right)_t\right\}_{j' \in [\mathsf{L}_{\text{in}}], \ell \in [\mathsf{L}_{\text{out}}]}$$

meaning the output of the decoder for the decoding set t is *not* consistent with w_ℓ^j for any $j \in [\mathsf{L}_{\text{in}}]$ or $\ell \in [\mathsf{L}_{\text{out}}]$.

In the notation of the events above, showing list-decoding soundness of the composed dPCP amounts to showing the following inequality, where the term on the left hand side is precisely the list-decoding error.

$$\Pr_{(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\mathsf{comp}}}[\mathcal{E}_{\mathsf{comp},1}\wedge\mathcal{E}_{\mathsf{comp},2}]\leqslant\delta_{\mathsf{in}}+\mathsf{L}_{\mathsf{in}}\cdot\delta_{\mathsf{out}}. \tag{64}$$

It will be convenient now to define a few events. Call the event on the left hand side of (64) $\mathcal{E}=\mathcal{E}_{\mathsf{comp},1}\wedge\mathcal{E}_{\mathsf{comp},2}$ and define the following events over $t,(\mathbf{b},r),(\mathbf{a};b)$ output by the distribution from (64):

- $\mathcal{E}_1$ is the event that there does not exist $j\in[\mathsf{L}_{\mathsf{in}}]$ such that $T_{A_{\mathsf{comp}}}[(\mathbf{b},r)]_{\leadsto\mathbf{b}}=\left(y_j^{\mathbf{a}}\right)_{\to\mathbf{b}}$ and $T_{A_{\mathsf{comp}}}[(\mathbf{b},r)]_{\leadsto t}=\left(y_j^{\mathbf{a}}\right)_{\leadsto t}$ for all t such that $(\mathbf{a},\mathbf{b})\in\mathrm{supp}(\mathcal{P}_t^{\mathsf{out}})$.

- $\mathcal{E}_2$ is the complementary event of the above. That is, that there does exist $j\in[\mathsf{L}_{\mathsf{in}}]$ such that $T_{A_{\mathsf{comp}}}[(\mathbf{b},r)]_{\leadsto\mathbf{b}}=\left(y_j^{\mathbf{a}}\right)_{\to\mathbf{b}}$ and $T_{A_{\mathsf{comp}}}[(\mathbf{b},r)]_{\leadsto t}=\left(y_j^{\mathbf{a}}\right)_{\leadsto t}$ for all t such that $(\mathbf{a},\mathbf{b})\in\mathrm{supp}(\mathcal{P}_t^{\mathsf{out}})$.

It is clear that $\mathcal{E}=(\mathcal{E}\wedge\mathcal{E}_1)\vee(\mathcal{E}\wedge\mathcal{E}_2)$ so the probability of interest is

$$\begin{aligned}\Pr_{(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\mathsf{comp}}}[\mathcal{E}]&=\Pr_{(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\mathsf{comp}}}[\mathcal{E}\wedge\mathcal{E}_1]+\Pr_{(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\mathsf{comp}}}[\mathcal{E}\wedge\mathcal{E}_2]\\&\leqslant\Pr_{(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\mathsf{comp}}}[\mathcal{E}\wedge\mathcal{E}_1]+\Pr_{(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\mathsf{comp}}}[\mathcal{E}_{\mathsf{comp},2}\wedge\mathcal{E}_2].\end{aligned} \tag{65}$$

We start by bounding $\Pr[\mathcal{E}\wedge\mathcal{E}_1]$. We think of $(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\mathsf{comp}}$ as being sampled in the following manner. First, $\mathbf{a}\sim\mathcal{Q}_{\mathsf{out}}(\circ,\cdot,\circ)$ is sampled, then $(\mathbf{b},i)\in\Gamma_{\mathsf{disc},\mathsf{out}}(\mathbf{a}),r\in[M]$ are chosen uniformly at random. Finally, t is chosen according to the marginal $\mathcal{Q}_{\mathsf{out}}(\cdot,\mathbf{a},\mathbf{b})$, and $(t,(\mathbf{b},r),(\mathbf{a};b))$ is output. Rewriting the distribution in this manner, one can check that we have

$$\Pr_{(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\mathsf{comp}}}[\mathcal{E}\wedge\mathcal{E}_1]\leqslant\mathbb{E}_{\mathbf{a}\sim\mathcal{Q}_{\mathsf{out}}(\circ,\cdot,\circ)}\left[\Pr_{(\mathbf{b},i)\in\Gamma_{\mathsf{disc},\mathsf{out}}(\mathbf{a}),r\in[M]}[\mathcal{F}_{\mathbf{a},\mathbf{b},i,r}\wedge\mathcal{E}_1]\right], \tag{66}$$

where $\mathcal{F}_{\mathbf{a},\mathbf{b},i,r}$ is the event that the assignments, $T_{A_{\mathsf{in},\mathbf{a}}}[(\mathbf{a};\mathbf{b},i,r)]:=T_{A_{\mathsf{comp}}}[(\mathbf{b},r)]\{\mathbf{a},i\}$ and $T_{B_{\mathsf{in},\mathbf{a}}}[(\mathbf{a};b)]:=T_{B_{\mathsf{comp}}}[(\mathbf{a};b)]$, satisfy the constraint on $((\mathbf{a};\mathbf{b},i,r),(\mathbf{a};b))$ in $\mathcal{D}_{\mathsf{in},\mathbf{a}}$. We note that the event $\mathcal{E}_1$ contains the event that the inner decoding on $\mathcal{D}_{\mathsf{in},\mathbf{a}}$ for $(\mathbf{b},i)$ is not consistent with any member of $\mathsf{List}_{\mathsf{in},\mathbf{a}}$. That is, that for every $j\in[\mathsf{L}_{\mathsf{in}}]$, either,

$$\left(D_{(\mathbf{b},i)}^{\mathsf{in},\mathbf{a}}((\mathbf{a};\mathbf{b},i,r),T_{A_{\mathsf{in},\mathbf{a}}}[(\mathbf{a};\mathbf{b},i,r)])\right)_{\mathbf{b}}=T_{A_{\mathsf{comp}}}[(\mathbf{b},r)]_{\leadsto\mathbf{b}}\neq\left(y_j^{\mathbf{a}}\right)_{\to\mathbf{b}}$$

or for some $t\in\Gamma_{\mathsf{dec},\mathsf{out}}(\mathbf{a})$,

$$\left(D_{(\mathbf{b},i)}^{\mathsf{in},\mathbf{a}}((\mathbf{a};\mathbf{b},i,r),T_{A_{\mathsf{in},\mathbf{a}}}[(\mathbf{a};\mathbf{b},i,r)])\right)_{t}=T_{A_{\mathsf{comp}}}[(\mathbf{b},r)]_{\leadsto t}\neq\left(y_j^{\mathbf{a}}\right)_{\leadsto t}.$$

In the above two equations we are using the hardcoding of the alphabet of $(\mathbf{b},r)$. Specifically, the equality in the first equation is by the "hardcoding outer constraints" and the equality in the second equation is by the "hardcoding decoding consistency".

Hence for each $\mathbf{a}$, the inner probability on the right hand side of (66) is at most the list-decoding error of $\mathcal{D}_{\mathsf{in},\mathbf{a}}$ applied to left and right assignments $T_{B_{\mathsf{in},\mathbf{a}}},T_{A_{\mathsf{in},\mathbf{a}}}$. This probability is at most δ_{in} by Claim 9.4, so

$$\Pr_{(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\mathsf{comp}}}[\mathcal{E}\wedge\mathcal{E}_1]\leqslant\delta_{\mathsf{in}},$$

and going back to (65), we get

$$\Pr_{(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\mathsf{comp}}}[\mathcal{E}]\leqslant\delta_{\mathsf{in}}+\Pr_{(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\mathsf{comp}}}[\mathcal{E}_2\wedge\mathcal{E}_{\mathsf{comp},2}]. \tag{67}$$

It remains to bound the probability on the right hand side of (67). To this end, for each $j\in[\mathsf{L}_{\mathsf{in}}],r\in[M]$ let $\mathcal{E}_{j,r}$ be the event that $T_{A_{\mathsf{out}}}^{(j)}[\mathbf{a}]$ and $T_{B_{\mathsf{out}}}^{(r)}[\mathbf{b}]$ satisfy the outer constraint, $\Phi_{\mathsf{out},(\mathbf{a},\mathbf{b})}$, or equivalently, the event that

$(T^{(j)}_{A_{\text{out}}}[\mathbf{a}])_{\to\mathbf{b}} = (y^{\mathbf{a}}_j)_{\to\mathbf{b}}$. Also define the event $\mathcal{F}_j$ to be the event that $D^{\text{out}}_t(\mathbf{a}, T^{(j)}_{A_{\text{out}}}[\mathbf{a}]) \notin \{(y^{\mathbf{a}}_1)_{\leadsto t}, \ldots, (y^{\mathbf{a}}_{\mathsf{L}_{\text{in}}})_{\leadsto t}\}$. The probability on the right hand side of (67) can be bounded as

$$
\Pr_{(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\text{comp}}}[\mathcal{E}_2 \wedge \mathcal{E}_{\text{comp},2}] \leqslant \mathbb{E}_{r\in[M]}\left[\Pr_{(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\text{comp}}}\left[\bigvee_{j\in[\mathsf{L}_{\text{in}}]} \mathcal{E}_{j,r} \wedge \mathcal{F}_j \;\middle|\; r\right]\right]
$$
$$
\leqslant \sum_{j\in[\mathsf{L}_{\text{in}}]} \mathbb{E}_{r\in[M]}\left[\Pr_{\substack{t\in[n]\\(\mathbf{a},\mathbf{b})\sim\mathcal{P}^{\text{out}}_t}}[\mathcal{E}_{j,r} \wedge \mathcal{F}_j]\right].
$$
(68)

In the first transition, we use the observation that $r \in [M]$ is chosen uniformly and independently of $t, \mathbf{b}, \mathbf{a}, b$ in the distribution of the probability on the left. In addition, we use the fact that if $\mathcal{E}_2$ occurs, it must be the case that for some $j \in [\mathsf{L}_{\text{in}}]$, $T^{(j)}_{A_{\text{out}}}[\mathbf{a}]$ is consistent with $T_{A_{\text{comp}}}[(\mathbf{b},r)]$ on $\mathbf{b}$ and all t such that $(\mathbf{a},\mathbf{b}) \in \text{supp}(\mathcal{P}^{\text{out}}_t)$. Hence, if $\mathcal{E}_2 \wedge \mathcal{E}_{\text{comp},2}$ occurs, then there must exist $j \in [\mathsf{L}_{\text{in}}]$ such that

$$
(T^{(j)}_{A_{\text{out}}}[\mathbf{a}])_{\to\mathbf{b}} = (y^{\mathbf{a}}_j)_{\to\mathbf{b}} \quad\text{and}\quad D^{\text{comp}}_t((\mathbf{b},r), T_{A_{\text{comp}}}[(\mathbf{b},r)]) = D^{\text{out}}_t(\mathbf{a}, T^{(j)}_{\text{out}}[\mathbf{a}]) \notin \left\{\left(w^{(j')}_{\ell}\right)_t\right\}_{j'\in[\mathsf{L}_{\text{in}}],\ell\in[\mathsf{L}_{\text{out}}]}.
$$

This takes care of the first transition. The second transition is by a union bound. Now fix a $j \in [\mathsf{L}_{\text{in}}]$ and notice that the probability, in the summand of the last line is precisely the list-decoding error of $T^{(j)}_{A_{\text{out}}}, T^{(r)}_{B_{\text{out}}}$, and hence for any $j \in [\mathsf{L}_{\text{in}}]$ and $r \in [M]$,

$$
\Pr_{\substack{t\in[n]\\(\mathbf{a},\mathbf{b})\sim\mathcal{P}^{\text{out}}_t}}[\mathcal{E}_{j,r} \wedge \mathcal{F}_j] \leqslant \delta_{\text{out}}.
$$

Plugging this into (68) and going back to (67) we get that the list-decoding error is

$$
\Pr_{(t,(\mathbf{b},r),(\mathbf{a};b))\sim\mathcal{Q}_{\text{comp}}}[\mathcal{E}] \leqslant \delta_{\text{in}} + \mathsf{L}_{\text{in}} \cdot \delta_{\text{out}}. \qquad\square
$$

10 The Final dPCP

We are now ready to construct our final dPCP, using the constructions and tools developed up to this point. For the remainder of the section, fix a size N circuit $\varphi \colon \Sigma^n_0 \to \{0,1\}$ and let $\mathcal{L} := \mathsf{SAT}(\varphi)$.

10.1 An Intermediate dPCP with Nearly Constant Alphabet Size

Our starting point is the dPCP for $\mathcal{L}$ given by Theorem 8.4 which we denote by $\mathcal{D}_1$. By composing it with the inner dPCP from Theorem 8.3, we get the following result.

Theorem 10.1 (Intermediate dPCP). *For every $\varepsilon > 0$ there exist $L = \text{poly}(1/\varepsilon)$ and $M = O_{|\Sigma_0|}(\text{poly}_\varepsilon(\log N))$, $d = O_\varepsilon(1)$, $k = O_{|\Sigma_0|}(\text{poly}_\varepsilon(\log\log\log N))$ such that the following holds. The language $\mathcal{L}$ has a projection dPCP of the form*

$$
\mathcal{D} = \left(A \cup B, E, \Sigma_A, \Sigma_B, \{\Phi_e\}_{e\in E}, \{D_t\}_{t\in[n]}, \{\mathcal{P}_t\}_{t\in[n]}\right)
$$

which satisfies the following

- ***Length.*** $O_{|\Sigma_0|}(N \cdot \text{poly}_\varepsilon(\log N))$.

- ***Alphabet Sizes.*** *The left and right alphabet sizes are* $2^{O_{|\Sigma_0|}(\text{poly}_\varepsilon(\log\log\log N))}$ *and* $O_\varepsilon(1)$ *respectively.*

- ***Decoding Degree.*** $O_\varepsilon(1)$.

- ***Projection Decision Complexity.*** $O_{|\Sigma_0|}(\text{poly}_\varepsilon(\log\log\log N))$.

- ***Decoding Complexity.*** $O_{\varepsilon,|\Sigma_0|}(1)$.

- **Complete Decoding Distribution.** *Call the complete decoding distribution $\mathcal{Q}$. Then the following holds:*

  - *For all $a \in A, b \in B$, we can write $\mathcal{Q}(\circ, a, b) = \frac{\mathsf{wt}(a,b)}{M}$ for some integer $\mathsf{wt}(a,b)$.*
  - *For every $b \in B$, $\sum_{a \in A} \mathsf{wt}(a,b) = d$.*
  - *For every $a \in A$, $\sum_{b \in B} \mathsf{wt}(a,b) = k$.*

- **Completeness.** *The dPCP has perfect completeness.*

- **Soundness.** *The dPCP has $(\mathsf{poly}(1/\varepsilon), \varepsilon)$-list-decoding soundness.*

Proof. Let us start by recalling the parameters of our Outer dPCP and inner dPCP in Table 1 and Table 2. Ultimately we will compose two times, where both times the inner dPCP is taken from Table 2 (although instantiated on different languages and hence different language-related parameters N', n', T'). Throughout all tables in this section, we abbreviate complete decoding distribution as CDD. Also, once we specify that a complete decoding distribution is M-discrete and can be written as $\mathcal{Q}(\circ, a, b) = \mathsf{wt}(a,b)/M$, we refer to the left weighted degrees as the sum $\sum_{b \in B} \mathsf{wt}(a,b)$ for each left vertex a, and the right weighted degree as the sum $\sum_{a \in A} \mathsf{wt}(a,b)$ for each right vertex b. Viewing the bipartite constraint graph as having $\mathsf{wt}(a,b)$ edges between a and b for each pair of vertices a and b, these quantities indeed correspond to degrees in this graph. Finally, all of the dPCPs in this section have perfect completeness, and all of the transformations we use preserve perfect completeness, so we do not include perfect completeness in any of the tables.

Henceforth, fix the circuit $\varphi : \Sigma_0^n \to \{0,1\}$ from the theorem statement. We will construct a dPCP for $\mathsf{SAT}(\varphi)$.

Parameter	Outer dPCP: $\mathcal{D}_1$
Length	$O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon(\log N))$
Left alphabet size	$2^{O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))}$
Right alphabet size	$O_\varepsilon(1)$
Decoding Degree	$O_\varepsilon(1)$
Projection Decision Complexity	$O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$
Decoding Complexity	$O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$
List-Decoding Soundness	$(\mathsf{poly}(1/\varepsilon), \varepsilon)$
CDD Marginal over Edges	$O_{|\Sigma_0|}(N\mathsf{poly}_\varepsilon(\log N))$-discrete
Left Weighted Degree	all at most $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$
Right Weighted Degree	all equal and $O_\varepsilon(1)$

Table 1: The parameters of the dPCP from Theorem 8.4 applied to the language $\mathsf{SAT}(\varphi)$ with list-decoding soundness error ε.

Parameter	Inner dPCP: $\mathcal{D}_2$
Length	$O_{T'}(N'^2 \cdot \mathsf{poly}_\varepsilon(\log N'))$
Left Vertices	$[n'] \cdot [M]$ for $M = O_{T'}(N'\mathsf{poly}_\varepsilon(\log N'))$
Left alphabet size	$2^{O_{T'}(\mathsf{poly}_\varepsilon(\log N'))}$
Right alphabet size	$O_\varepsilon(1)$
Decoding Degree	$O_\varepsilon(1)$
Projection Decision Complexity	$O_{T'}(\mathsf{poly}_\varepsilon(\log N'))$
Decoding Complexity	$O_{T'}(\mathsf{poly}_\varepsilon(\log N'))$
List-Decoding Soundness	$(\mathsf{poly}(1/\varepsilon), \varepsilon)$
CDD Marginal over Edges	$n' \cdot M \cdot O_{T'}(\mathsf{poly}_\varepsilon(\log N'))$-discrete
	(so that $K(T', N')$ from Theorem 8.3 is $O_{T'}(\mathsf{poly}_\varepsilon(\log N'))$)
Left Weighted Degree	all equal and $O_{T'}(\mathsf{poly}_\varepsilon(\log N'))$

Table 2: The parameters of the dPCP from Theorem 8.3 applied to the language $\mathsf{SAT}(\varphi')$ for some circuit φ : $(\Sigma'_0)^{n'} \to \{0, 1\}$ of size at most N' over an alphabet of size $|\Sigma'_0| \leqslant T'$ with list-decoding soundness error ε.

First Composition. We apply the composition of Theorem 9.1 with outer dPCP $\mathcal{D}_1$ and with inner dPCPs given Theorem 8.3. Referring to the parameters from Table 1 and Table 2, one can check that we get a dPCP, called $\mathcal{D}_3$, with parameters as described in Table 3. In particular, the parameters N', T' arising in the composed dPCP guarantee of Theorem 9.1 are $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$ and $O_{\varepsilon, |\Sigma_0|}(1)$ respectively.

Parameter	Composed dPCP: $\mathcal{D}_3$
Length	$O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon(\log N))$
Left alphabet size	$2^{O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log \log N))}$
Right alphabet size	$O_\varepsilon(1)$
Decoding Degree	$O_\varepsilon(1)$
Projection Decision Complexity	$O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log \log N))$
Decoding Complexity	$O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log \log N))$
List-Decoding Soundness	$(\mathsf{poly}(1/\varepsilon), \varepsilon)$
CDD Marginal over Edges	$O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon(\log N))$-discrete
Left Weighted Degree	all equal and $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log \log N))$

Table 3: Composing $\mathcal{D}_1$ and $\mathcal{D}_2$. In $\mathcal{D}_2$, the parameters N' and T' are $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log N))$ and $O_{\varepsilon, |\Sigma_0|}(1)$ respectively.

Let $A_3 \cup B_3$ be the sides of the constraint graph, and call $\mathcal{Q}_3$ the complete decoding distribution of $\mathcal{D}_3$. By the discreteness of the marginal of $\mathcal{Q}_3$ over the edges, we can thus write, for every $(a, b) \in A_3 \times B_3$, $\mathcal{Q}_3(\circ, a, b) = \frac{\mathsf{wt}(a,b)}{J}$ for some $\mathsf{wt}(a, b) \in \mathbb{N}$, where $J = O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon(\log N))$.

The alphabet size is now significantly smaller but is still super-constant. We would like to compose again to further reduce the alphabet size, but we note that now $\mathcal{D}_3$ does not satisfy all of the requirements of the outer dPCP in Theorem 9.1. In particular, Theorem 9.1 requires the outer dPCP's complete decoding distribution to have uniform marginal over the right vertices — a property that we do not have in Table 3.

To fix this, we set $d' = O(1/\varepsilon^{100})$, multiply all of the weights by d', and then apply Lemma 7.2 with degree d'. We let the resulting dPCP be $\mathcal{D}_3'$, and let its constraint graph have sides $A_3 \cup B_3'$ with complete decoding distribution $\mathcal{Q}_3'$. It is straightforward to check that $\mathcal{D}_3'$ has all of the same parameters as $\mathcal{D}_3$ in Table 3, except for the following changes

- The length becomes $d'J$, where $J = O_{|\Sigma_0|}(N\mathsf{poly}_\varepsilon(\log N))$ is the discreteness parameter of the complete decoding distribution's marginal over the edges.

- The discreteness parameter of the CDD Marginal over edges is multiplied by d' and is still $O_{|\Sigma_0|}(N\cdot\mathsf{poly}_\varepsilon(\log N))$.

- The right weighted degrees are all $d' = O_\varepsilon(1)$.

- The left weighted degrees are all multiplied by d' and are still all equal to some $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log\log N))$.

Now we can compose $\mathcal{D}_3'$ with the inner dPCP from Theorem 8.3 one more time using Theorem 9.1 and again apply Lemma 7.2 with degree d' on the resulting dPCP. This time, the inner dPCPs are applied with parameters N' and T' that are $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log\log N))$ and $O_{\varepsilon,|\Sigma_0|}(1)$. Altogether, we obtain the dPCP, $\mathcal{D}_4$, whose parameters are listed in Table 4. We note that, similarly to in $\mathcal{D}_3'$, all of the parameters except for weighted right degrees come from Theorem 9.1. The weighted right degrees from Table 4 are obtained by applying Lemma 7.2 which changes the length to $O_\varepsilon(1)$ times the discreteness parameter of the complete decoding distribution's marginal over the edges, and increases all of the other parameters by an $O_\varepsilon(1)$ factor which is absorbed. $\qquad\square$

Parameter	Composed dPCP: $\mathcal{D}_4$
Length	$O_{|\Sigma_0|}\left(N\cdot\mathsf{poly}_\varepsilon(\log N)\right)$
Left alphabet size	$2^{O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log\log\log N))}$
Right alphabet size	$O_\varepsilon(1)$
Decoding Degree	$O_\varepsilon(1)$
Projection Decision complexity	$O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log\log\log N))$
Decoding Complexity	$O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log\log\log N))$
List-Decoding Soundness	$(\mathsf{poly}(1/\varepsilon),\varepsilon)$
CDD Marginal over Edges	$O_{|\Sigma_0|}(N\cdot\mathsf{poly}_\varepsilon(\log N))$-discrete
Left Weighted Degree	all equal and $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log\log\log N))$
Right Weighted Degree	all equal and $O_\varepsilon(1)$

Table 4: Composing $\mathcal{D}_3'$ and $\mathcal{D}_2$. In $\mathcal{D}_2$, the parameters N' and T' are $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log\log N))$ and $O_{\varepsilon,|\Sigma_0|}(1)$ respectively.

10.2 The Final dPCP with Constant Alphabet Size

Now we are finally ready to prove our main dPCP theorem, Theorem 3.12, which we restate below for convenience.

Theorem 10.2. *For all $\varepsilon > 0$ there exists $C \in \mathbb{N}$ and $L = \mathsf{poly}(1/\varepsilon)$ such that the following holds for all size N circuits $\varphi\colon \Sigma_0^n \to \{0,1\}$ over some alphabet Σ_0. The language $\mathsf{SAT}(\varphi)$ has a 2-query, poly-time constructible dPCP with the following properties*

- ***Length.*** $O_{|\Sigma_0|}\left(N\log^C N\right)$,

- ***Alphabet Size.*** $O_{\varepsilon,|\Sigma_0|}(1)$,

- **Complete Decoding Distribution.** *The marginal of the complete decoding distribution over the right vertices is uniform and the marginal of the complete decoding distribution over the left vertices is uniform.*

- **Completeness.** *The dPCP has perfect completeness,*

- **Soundness.** $(L, 2\varepsilon^{1/5})$-*list-decoding soundness.*

Theorem 10.2 is obtained from Theorem 10.1 by performing yet another composition step. This time, though, we will compose with a Hadamard-code based dPCP construction, which has a constant size alphabet (but an exponential length). Starting from $\mathcal{D}_4$, however, this exponential size dPCP is applied to a circuit of size $\mathsf{poly}(\log \log \log N)$, hence the overall length increase is only a factor of $\mathsf{poly}(\log(N))$, which is affordable to us. Below we state the properties of the Hadamard dPCPs, which can be found in slightly different forms in [DH13, BMVY25].

Theorem 10.3 (Hadamard dPCP)**.** *For every* $\varepsilon > 0$ *there exist* $L = \mathsf{poly}(1/\varepsilon)$ *and functions* $M, K : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *satisfying*

$$M(T, N) = 2^{O_{\varepsilon,T}(N^2)} \quad and \quad K(T, N) = O_\varepsilon(1)$$

such that the following holds. For every alphabet Σ_0 *of size at most* T *and circuit* $\varphi : \Sigma_0^n \to \{0, 1\}$ *of size at most* N, *the language* $\mathsf{SAT}(\varphi)$ *has a projection dPCP of the form*

$$\mathcal{D} = \big(([n] \times [M]) \cup B, E, \Sigma_A, \Sigma_B, \{\Phi_e\}_{e \in E}, \{D_t\}_{t \in [n]}, \{\mathcal{P}_t\}_{t \in [n]}\big)$$

which satisfies the following with $M := M(T, N)$ *and* $K := K(T, N)$.

- **Length.** $O(nM)$.

- **Alphabet Sizes.** *The left alphabet size is* $O_{\varepsilon,T}(1)$ *and the right alphabet size is* $O_\varepsilon(1)$.

- **Degrees.** $\mathcal{D}$ *has decoding degree* 1.

- **Projection Decision Complexity.** $O_{\varepsilon,T}(1)$.

- **Decoding Complexity.** $O_{\varepsilon,T}(1)$.

- **Decoding Distribution.** *Let* $\mathcal{Q}$ *be the complete decoding distribution. Then,* $\mathcal{Q}$ *satisfies the following:*

  - *For each* $(t, r) \in [n] \times [M]$ *and* $b \in B$ *we can write* $\mathcal{Q}(\circ, (t, r), b) = \frac{\mathsf{wt}((t,r),b)}{nMK}$ *for some* $\mathsf{wt}((t, r), b) \in \mathbb{N}$.
  - *For each* $t \in [n]$, *the distribution* $\mathcal{Q}(t, \cdot, \circ)$ *is uniform over* $\{t\} \times [M]$ *and hence for every* $(t, r) \in [n] \times [M]$,

$$\sum_{b \in B} w((t, r), b) = K.$$

- **Completeness.** *The dPCP has perfect completeness.*

- **Soundness.** *The dPCP has* $(L, 2\varepsilon^{1/5})$-*list-decoding soundness.*

The proof appears in Appendix A. On a high-level, the construction is identical to the one in [BMVY25, Theorem 7.7], with minor additional considerations in the analysis.

We can now complete the proof of Theorem 10.2 by composing $\mathcal{D}_4$ with the Hadamard dPCP from Theorem 10.3 which we denote by $\mathcal{D}_{\mathsf{Had}}$.

Proof of Theorem 10.2. We use Theorem 9.1 with $\mathcal{D}_4$ as the outer dPCP, and the Hadamard based dPCP $\mathcal{D}_{\mathsf{Had}}$ from Theorem 10.3 as the inner dPCP. When applying $\mathcal{D}_{\mathsf{Had}}$, the parameters N' and T' are $\mathsf{poly}_\varepsilon(\log \log \log N)$ and $O_{\varepsilon,|\Sigma_0|}(1)$ respectively. We also have to apply Lemma 7.2 again on the resulting dPCP to obtain regularity for the right weighted degrees (and hence uniformity of the marginal of the complete decoding distribution over the right vertices). We call the resulting dPCP $\mathcal{D}_5$ and list its parameters in Table 5. Here, we remark that it is important that the dPCP given by Theorem 9.1 has complete decoding distribution whose marginal over the edges is $O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon(\log N))$-discrete,

76

so the length blowup from applying Lemma 7.2 is controlled. Similar to in the previous two compositions, all of the parameters in Table 5 are given by Theorem 9.1, except for the weighted right degrees. These are obtained by applying Lemma 7.2 which changes the length to $O_\varepsilon(1)$ times the discreteness parameter of the complete decoding distribution's marginal over the edges, and increases all of the other parameters by an $O_\varepsilon(1)$ factor which is absorbed. □

Parameter	Composed dPCP: $\mathcal{D}_5$
Length	$O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon(\log N))$
Left alphabet size	$O_{\varepsilon,|\Sigma_0|}(1)$
Right alphabet size	$O_\varepsilon(1)$
List-decoding soundness	$(\mathsf{poly}(1/\varepsilon), \varepsilon)$
CDD Marginal over Edges	$O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon(\log N))$-discrete
Left Weighted Degree	all equal
Right Weighted Degree	all equal

Table 5: Composing $\mathcal{D}_4$ and $\mathcal{D}_{\mathsf{Had}}$. In $\mathcal{D}_{\mathsf{Had}}$, the parameters N' and T' are $O_{|\Sigma_0|}(\mathsf{poly}_\varepsilon(\log \log \log N))$ and $O_{\varepsilon,|\Sigma_0|}(1)$ respectively.

11 Applications

In this section we describe applications of our main dPCP result (Theorem 3.12) to PCPPs, RLDCs, and CSP-reconfiguration.

11.1 Proof of Theorem 1.7

Our 3-query PCPP construction is based on the dPCP from Theorem 3.12, along with an additional consistency check. Due to the latter, we have to take into account the list size of the dPCP, which makes our PCPP effective only for words which are very far from the language.

Theorem 11.1. *For any $\varepsilon > 0$ the following holds for any alphabet Σ_0. For all circuits $\varphi : \Sigma_0^n \to \{0,1\}$ of size N, there exists a 3-query PCPP $\Psi = (X \cup A \cup B, E, \mathcal{P}, \Sigma_0 \cup \Sigma_A \cup \Sigma_B, \{\Phi_e\}_{e \in E})$ for the language $\mathsf{SAT}(\varphi)$ with the following parameters:*

- ***Length.** $|A|, |B| = O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon \log N)$, $|X| = n$ and the edge set $E \subseteq X \times A \times B$ consists of $O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon \log N)$ hyperedges, each has size 3 and includes one vertex from each one of A, B and X.*

- ***Distribution.** The marginal distribution of $\mathcal{P}$ on B is uniform.*

- ***Alphabet Size.** Vertices in A, B have alphabets Σ_A, Σ_B of size $O_{|\Sigma_0|,\varepsilon}(1)$ each, the alphabet of X is Σ_0.*

- ***Soundness.** There exists a universal constant $C > 0$ such that for any $\eta \geqslant 1 - \varepsilon^{C+1}$, if $w : X \to \Sigma_0$ is η-far from $\mathsf{SAT}(\varphi)$ then for any $T_A : A \to \Sigma_A, T_B : B \to \Sigma_B$,*

$$\Pr_{(x,a,b)\sim\mathcal{P}}[\Phi_{(x,a,b)}(w_t, T_A[a], T_B[b]) = 1] \leqslant 2\varepsilon.$$

Proof. Fix $\varepsilon > 0$. We use Theorem 3.12 to construct a dPCP for $\mathsf{SAT}(\varphi)$:

$$\mathcal{D} = \left(A \cup B, E, \Sigma_A, \Sigma_B, \{\Phi_e\}_{e \in E}, \{\mathcal{P}_t\}_{t \in [n]}, \{D_t\}_{t \in [n]}\right)$$

with $(\mathsf{L}, \varepsilon)$ list-decoding. Then, we construct a PCPP $\Psi = (V', E', \mathcal{P}', \Sigma', \{\Phi'_e\}_{e \in E'})$ with $X \subseteq V'$ as follows: $V' = A \cup B \cup X$ where $|X| = n$, $E' = \{(t, a, b) \ : \ t \in X, (a, b) \in \operatorname{supp}(\mathcal{P}_t)\}$, $\Sigma' = \Sigma_A \cup \Sigma_B \cup \Sigma_0$. For an edge $(t, a, b) \in E'$, the satisfying assignments of the constraint $\Phi_{(t,a,b)}$ are tuples (σ, α, β) such that (α, β) satisfies the constraint $\Phi_{(a,b)}$ of $\mathcal{D}$, and $D_t(a, \alpha)) = \sigma$. Finally, the distribution $\mathcal{P}'$ is the complete distribution of $\mathcal{D}$, which we denote by $\mathcal{Q}$.

The completeness, query complexity, number of vertices, edges and alphabet size of Ψ follow directly from the construction. The marginal distribution over B is uniform as $\mathcal{Q}(\circ, \circ, \cdot) = \mathcal{P}'(\circ, \circ, \cdot)$ is uniform by Theorem 3.12.

We prove the soundness condition. Suppose $w \in \Sigma_0^n$ is η-far from $\mathsf{SAT}(\varphi)$, and fix assignments $T_A : A \to \Sigma_A$ and $T_B : B \to \Sigma_B$ to the dPCP $\mathcal{D}$. By the list-decoding property of the dPCP, there exists a list $\{u_1, \ldots, u_\mathsf{L}\} \subseteq \mathsf{SAT}(\varphi)$ such that

$$\Pr_{(t,a,b) \sim \mathcal{Q}}[\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge D_t(a, T_A[a]) \notin \{(u_i)_t\}_{i \in [\mathsf{L}]}] \leqslant \varepsilon. \tag{69}$$

We conclude that

$$\Pr_{(t,a,b) \sim \mathcal{P}'}[\Phi_{(t,a,b)}(w_t, T_A[a], T_B[b]) = 1]$$
$$= \Pr_{(t,a,b) \sim \mathcal{Q}}[\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge (w_t = D_t(a, T_A[a]))]$$
$$= \Pr_{(t,a,b) \sim \mathcal{Q}}[\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge (w_t = D_t(a, T_A[a])) \wedge D_t(a, T_A[a]) \notin \{(u_i)_t\}_{i \in [\mathsf{L}]}]$$
$$+ \Pr_{(t,a,b) \sim \mathcal{Q}}[\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge (w_t = D_t(a, T_A[a])) \wedge D_t(a, T_A[a]) \in \{(u_i)_t\}_{i \in [\mathsf{L}]}]$$
$$\leqslant \varepsilon + \Pr_{(t,a,b) \sim \mathcal{Q}}[w_t \in \{(u_i)_t\}_{i \in [\mathsf{L}]}]$$
$$\leqslant \varepsilon + \sum_{i \in [\mathsf{L}]} \Pr_{(t,a,b) \sim \mathcal{Q}}[w_t = (u_i)_t]$$
$$\leqslant \varepsilon + \mathsf{L} \cdot (1 - \eta),$$

where in the third transition we us (69), in the fourth transition we use the union bound, and in the last transition we use the fact that $u_i \in \mathsf{SAT}(\varphi)$, that w is η-far from $\mathsf{SAT}(\varphi)$ and that the marginal distribution of $\mathcal{Q}$ over t is uniform. Since $L = 1/\varepsilon^C$ for some absolute constant $C > 0$, given that $\eta \geqslant 1 - \varepsilon^{C+1}$ we conclude the the soundness error is at most 2ε. $\qquad\square$

11.2 Proof of Theorem 1.8

Our 4-query PCPP with optimal proximity/soundness guarantees is constructed via a similar transformation as in Theorem 11.1 with an additional encoding idea.

Theorem 11.2. *For any $\varepsilon > 0$ the following holds for any alphabet Σ_0. For all circuits $\varphi : \Sigma_0^n \to \{0, 1\}$ of size N over Σ_0, there exists a 4-query PCPP $\Psi = (X \cup Y \cup A \cup B, E, \mathcal{P}, \Sigma_0 \cup \Sigma \cup \Sigma_A \cup \Sigma_B, \{\Phi_e\}_{e \in E})$ for the language $\mathsf{SAT}(\varphi)$ with the following parameters:*

- ***Length.*** *$|A|, |B| = O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon \log(N))$, $|X| = n, |Y| = O_\varepsilon(n)$ and the edge set $E \subseteq X \times Y \times A \times B$ consists of $O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon \log(N))$ hyperedges, each has size 4 and includes one vertex from each one of A, B, X, Y.*

- ***Alphabet Size.*** *Vertices in A, B have alphabets Σ_A, Σ_B of size $O_{|\Sigma_0|, \varepsilon}(1)$ each, the alphabet of X is Σ_0 and the alphabet of Y, denoted by Σ_Y, has size $O_\varepsilon(1)$.*

- ***Soundness.*** *For any $\eta > 0$, if $w : X \to \Sigma_0$ is η-far from $\mathsf{SAT}(\varphi)$ then for any $T_A : A \to \Sigma_A, T_B : B \to \Sigma_B, T_Y : Y \to \Sigma_Y$,*

$$\Pr_{(x,y,a,b) \sim \mathcal{P}}[\Phi_{(x,y,a,b)}(w_x, T_Y[y], T_A[a], T_B[b]) = 1] \leqslant 1 - \eta + \varepsilon.$$

Proof. Let $\mathcal{C} : \Sigma_0^n \to \Sigma^{n/\rho}$ be a linear time encodable code with distances $1 - \mu$ as in Lemma 2.1; we will choose the parameter μ later and then $\rho = \rho(\mu) > 0$ is picked. Define the circuit $\varphi' : \Sigma_0^n \times \Sigma^{n/\rho} \to \{0,1\}$ which on input $(m, m') \in \Sigma_0^n \times \Sigma^{n/\rho}$ outputs 1 only if $\mathcal{C}(m) = m'$ and $\varphi(m) = 1$. Note that since φ is of size N and the encoding circuit is linear in n, the size of φ' is $\widetilde{O}(N + n/\rho)$. Also, we have

$$\mathsf{SAT}(\varphi') = \{(m, m') \in \Sigma_0^n \times \Sigma^{n/\rho} \mid m \in \mathsf{SAT}(\varphi), m' = \mathcal{C}(m)\}.$$

Let $d \in \mathbb{N}$ be a parameter to be chosen later, and let $\mathcal{S} \subseteq [n] \times [n/\rho]$ be the edges of a bipartite expander with sides $[n] \times [n/\rho]$ and right degree d, left degree d/ρ and second singular value at most $O(1/\sqrt{d})$. We note that $\mathcal{S}$ can be constructed in polynomial time by Lemma 2.5.

Consider the circuit $\varphi'' : (\Sigma \times \Sigma_0)^{|\mathcal{S}|} \to \{0,1\}$ defined as follows. Given an input $w'' : \mathcal{S} \to \Sigma \times \Sigma_0$, define $w : [n] \to \Sigma_0$ by picking for each $i \in [n]$ an arbitrary j that $(i, j) \in \mathcal{S}$ and then defining $w(i) = w''(i, j)|_i$, and similarly define $u : [n/\rho] \to \Sigma$. The output $\varphi''(w'')$ is 1 if:

1. For all $(i, j) \in \mathcal{S}$, $w''(i, j) = (w(i), u(j))$.

2. $\phi'(w, u) = 1$.

We note that the size of φ'' is $\widetilde{O}(|\mathcal{S}|)$ plus the size of φ', which is $\widetilde{O}_{d,\rho}(N)$. Using Theorem 3.12 we get a dPCP for the language $\mathsf{SAT}(\varphi'')$ with $(\mathsf{L}, \varepsilon)$-list decoding, which we denote as

$$\mathcal{D} = \left(A \cup B, E, \Sigma_A, \Sigma_B, \{\Phi_{(a,b)}\}_{(a,b)\in E}, \{\mathcal{P}_{(x,y)}\}_{(x,y)\in\mathcal{S}}, \{D_{(x,y)}\}_{(x,y)\in\mathcal{S}}\right).$$

We denote the complete decoding distribution of $\mathcal{D}$ by $\mathcal{Q}$. We think of the decoder on $(x, y) \in \mathcal{S}$ as outputting two elements, which we call $D_{(x,y)}(\cdot, \cdot)_x$ and $D_{(x,y)}(\cdot, \cdot)_y$.

We construct the PCPP for $\mathsf{SAT}(\varphi)$ as $\Psi = (X \cup Y \cup A \cup B, E', \mathcal{P}', \Sigma_0 \cup \Sigma_A \cup \Sigma_B, \{\Phi_{e'}\}_{e'\in E'})$ where $|X| = n, |Y| = n/\rho$, the edge set is

$$E' = \{((x, y), a, b) \mid (x, y) \in \mathcal{S}, (a, b) \in \mathrm{supp}(\mathcal{P}_{(x,y)})\}$$

and the distribution $\mathcal{P}'$ is taken to be $\mathcal{Q}$. The alphabet of A is Σ_A, the alphabet of B is Σ_B, the alphabet of X is Σ_0 and the alphabet of Y is Σ. For $((x, y), a, b) \in E'$, a tuple $((\sigma, \tau), \alpha, \beta)$ satisfies the constraint $\Phi_{((x,y),a,b)}$ if (α, β) satisfies $\Phi_{(a,b)}$, $D_{(x,y)}(a, \alpha)_x = \sigma$, and $D_{(x,y)}(a, \alpha)_y = \tau$.

The size and the other parameters of the PCPP follow from the parameters of the dPCP from Theorem 3.12 where the circuit size is $|\varphi''|$, and in the rest of the argument we argue the soundness of Ψ. Let $w : X \to \Sigma_0$ be η-far from $\mathsf{SAT}(\varphi)$, and fix $T_A : A \to \Sigma_A, T_B : B \to \Sigma_B$ and $T_Y : Y \to \Sigma$. By the list-decoding soundness property of the $\mathcal{D}$ there exists a list $\{w_i''\}_{i\in\mathsf{L}} \subseteq \mathsf{SAT}(\varphi'')$ such that

$$\Pr_{((x,y),a,b)\sim\mathcal{Q}}[\Phi_{(a,b)}(T[a], T[b]) = 1 \ \wedge \ D_{(x,y)}(a, T[a]) \notin \{w_i''(x, y)\}_{i\in[\mathsf{L}]}] \leqslant \varepsilon. \tag{70}$$

As $w_i'' \in \mathsf{SAT}(\varphi'')$ we conclude that there exists $u_i \in \mathsf{SAT}(\varphi)$, such that we have $w''(x, y) = u_i(x), \mathcal{C}(u_i)_y$ for all $(x, y) \in \mathcal{S}$, and we fix such u_i henceforth. Let $\mathcal{E}$ be the event that $\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \ \wedge \ D_{(x,y)}(a, T_A[a])_x = w_x \ \wedge \ D_{(x,y)}(a, T_A[a])_y = T_Y[y]$. The soundness of the PCPP is thus $\Pr_{((x,y),a,b)\sim\mathcal{P}'}[\mathcal{E}]$, which can be written as

$$\Pr_{((x,y),a,b)\sim\mathcal{Q}}[\mathcal{E}] = \Pr_{((x,y),a,b)\sim\mathcal{Q}}[\mathcal{E} \ \wedge \ D_{(x,y)}(a, T_A[a]) \notin \{w_i''(x, y)\}_{i\in[\mathsf{L}]}]$$

$$+ \Pr_{((x,y),a,b)\sim\mathcal{Q}}[\mathcal{E} \wedge \ D_{(x,y)}(a, T_A[a]) \in \{((u_i)_x, (\mathcal{C}(u_i))_y)\}_{i\in[\mathsf{L}]}].$$

The first term is at most ε by (70), and we now focus on the second term. Let $\mathcal{E}_y$ be the event that there are $i \neq i''$ such that $\mathcal{C}(u_i)_y = \mathcal{C}(u_{i'})_y$. We get that

$$\Pr_{((x,y),a,b)\sim\mathcal{Q}}[\mathcal{E} \ \wedge \ D_{(x,y)}(a, T_A[a]) \in \{((u_i)_x, (\mathcal{C}(u_i))_y)\}_{i\in[\mathsf{L}]}]$$

$$\leqslant \Pr_{((x,y),a,b)\sim\mathcal{Q}}[\mathcal{E} \ \wedge \ D_{(x,y)}(a, T_A[a]) \in \{((u_i)_x, (\mathcal{C}(u_i))_y)\}_{i\in[\mathsf{L}]} \ \wedge \ \bar{\mathcal{E}}_y] + \Pr_{((x,y),a,b)\sim\mathcal{Q}}[\mathcal{E}_y].$$

Note that for any $i \neq i'$, the words $\mathcal{C}(u_i)$, $\mathcal{C}(u_{i'})$ agree on at most μ fraction of coordinates, so by the union bound $\Pr_{((x,y),a,b)\sim\mathcal{Q}}[\mathcal{E}_y] \leqslant \binom{\mathsf{L}}{2}\mu$. As for the other term on the right hand side, note that as $\mathcal{E}$ requires that $D_{(x,y)}(a, T_A[a]) = (w_x, T_Y[y])$ we get

$$
\Pr_{((x,y),a,b)\sim\mathcal{Q}}[\mathcal{E} \,\wedge\, D_{(x,y)}(a, T_A[a]) \in \{((u_i)_x, (\mathcal{C}(u_i))_y)\}_{i\in[\mathsf{L}]} \,\wedge\, \bar{\mathcal{E}}_y]
$$
$$
\leqslant \Pr_{(x,y)\in\mathcal{S}}[(w_x, T_Y[y]) \in \{((u_i)_x, (\mathcal{C}(u_i))_y)\}_{i\in[\mathsf{L}]} \,\wedge\, \bar{\mathcal{E}}_y].
$$

If the event above holds, then the values $\{(\mathcal{C}(u_i))_y\}$ are all different, so at most one of them is equal to $T_Y[y]$. Defining $A_I = \{y \,:\, \bar{\mathcal{E}}_y \,\wedge\, (\mathcal{C}(u_I))_y = T_Y[y]\}$ and $B_I = \{x \,:\, w_x = (u_I)_x\}$ for $I \in [\mathsf{L}]$, we get that

$$
\Pr_{(x,y)\in\mathcal{S}}[(w_x, T_Y[y]) \in \{((u_i)_x, (\mathcal{C}(u_i))_y)\}_{i\in[\mathsf{L}]} \,\wedge\, \bar{\mathcal{E}}_y] = \sum_{I\in[\mathsf{L}]} \mathbb{E}_{(x,y)\in\mathcal{S}}[1_{y\in A_I} \cdot 1_{x\in B_I}]
$$
$$
\leqslant \sum_{I\in[\mathsf{L}]} \mathbb{E}_{y\in[n/\rho]}[1_{y\in A_I}] \cdot \mathbb{E}_{x\in[n]}[1_{x\in B_I}] + \frac{1}{\sqrt{d}} \cdot \sqrt{\frac{|A_I|}{n/\rho} \cdot \frac{|B_I|}{n}}
$$
$$
\leqslant 1 - \eta + \frac{\mathsf{L}}{\sqrt{d}}.
$$

In the second transition we used Lemma 2.3, and in the last transition we used the fact that $\mathbb{E}_{x\in[n]}[1_{x\in B_I}] \leqslant 1 - \eta$ for all I (as w is η-far from $\mathsf{SAT}(\varphi)$) and that $\sum_{I\in[\mathsf{L}]} \mathbb{E}_{y\in[n/\rho]}[1_{y\in A_I}] = \Pr_{((x,y),a,b)\sim\mathcal{Q}}[\bar{\mathcal{E}}_y] \leqslant 1$.

Combining all of the above, we get that

$$
\Pr_{((x,y),a,b)\sim\mathcal{Q}}[\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \,\wedge\, D_{(x,y)}(a, T_A[a])_x = w_x] \leqslant 1 - \eta + \binom{\mathsf{L}}{2} \cdot \mu + \frac{\mathsf{L}}{\sqrt{d}}.
$$

As $\mathsf{L} = 1/\varepsilon^C$ for some absolute constant C by Theorem 3.12, choosing $\mu = \varepsilon^{2C}$ and $d = 4\varepsilon^{-2C}$, we get that the error is at most $1 - \eta + \varepsilon$. $\qquad\square$

11.3 A Query-Preserving PCPP to RLDC Transformation

To prove Theorem 1.9 we describe a generic transformation that converts a sufficiently good 3-query PCPP into a 3-query RLDC.

Theorem 11.3. *Suppose for some absolute constant $C > 0$ we have the following two ingredients:*

- *For any alphabet Σ_0, constant $\delta > 0$, and circuit $\varphi\colon \Sigma_0^n \to \{0,1\}$ of size N there is a 3-query PCPP $\Psi = (X \cup A \cup B, E, \mathcal{P}, \Sigma_0 \cup \Sigma, \{\Phi_e\}_{e\in E})$ satisfying:*

  - ***Length.*** $O_{|\Sigma_0|}(N \cdot \mathsf{poly}_\varepsilon(\log N))$.
  - ***Queries.*** *Each hyperedge $e = (e_1, e_2, e_3) \in E$ satisfies $e_1 \in X, e_2 \in A, e_3 \in B$, and choosing e uniformly, e_1 is uniformly random in X, e_2 is uniformly random in A, and e_3 is uniformly in B.*
  - ***Soundness.*** *For all $\eta \geqslant 1 - \delta^C$, if $w : X \to \Sigma_0$ is η-far from $\mathsf{SAT}(\varphi)$, then for any $T_A : A \to \Sigma_A, T_B : B \to \Sigma_B$*
  $$
  \Pr_{(t,a,b)\sim\mathcal{P}}[\Phi_{(t,a,b)}(w_t, T_A[a], T_B[b]) = 1] \leqslant 2\delta.
  $$

- *For any $\delta \in (0,1)$ there is $\rho > 0$, an alphabet Σ_0 of size $O_\delta(1)$, and a code $\mathcal{C}_0 : \{0,1\}^k \to \Sigma_0^{k/\rho}$ with relative distance δ, that is both quasi-linear time encodable and decodable.*

Then for all $\mu \in (0,1)$ sufficiently small, and sufficiently large k, there is a 3-query RLDC with message length k, blocklength $k^2 \cdot \mathsf{poly}_\mu(\log k)$, alphabet size $O_\mu(1)$, distance $\frac{1-\mu}{2}$, and decoding radius μ. Furthermore, for all $\delta > 0$, the RLDC has success-rate $1 - 16\delta - o(1)$ on inputs that are δ-close to a valid codeword.

Proof. Fix $\mu > 0$ sufficiently small and take $\rho > 0$ and a code $\mathcal{C}_0$ as per the second item with distance $1 - \mu$, rate ρ, and appropriate alphabet Σ_0. For an index $i \in [k]$ and a bit $b \in \{0, 1\}$, let $\varphi_{i,b} : \Sigma_0^{k/\rho_0} \to \{0, 1\}$ be a circuit that given a word w of length k/ρ, outputs 1 if w is a member of $\mathcal{C}_0$, and furthermore the i'th bit of the encoded message is equal to b. More precisely on an input $w \in \Sigma_0^{k/\rho}$, run the decoding map to get $m = \mathsf{Dec}_{\mathcal{C}_0}(w)$, then output 1 if and only if $\mathcal{C}_0(m) = w$ and $m_i = b$. Since the code $\mathcal{C}_0$ is quasi-linear time encodable and decodable, the circuit size of $\varphi_{i,b}$ is $\widetilde{O}_\mu(k)$. Define $\mathcal{L}_{i,b} := \mathsf{SAT}(\varphi_{i,b})$.

Unifying the proof length for PCPP invocations. For $i \in [k], b \in \{0, 1\}$, take

$$\Psi_{i,b} = ([k/\rho] \cup A_{i,b} \cup B_{i,b}, E_{i,b}, \Sigma, \{\Phi_e\}_{e \in E}, \mathcal{P}_{i,b})$$

to be the PCPP from the premise of the statement for the language $\mathcal{L}_{i,b}$. Without loss of generality, we assume that all the PCPPs have the same alphabet which has size $O_\varepsilon(1)$. The code we construct will contains multiple encodings of the message and relevant proofs, but we first address a small technical issue. The PCPPs are guaranteed to have length that is bounded by $k \cdot \log^C k$ (as the circuit size is $\widetilde{O}_\mu(k)$), but $A_{i,b}$ (as well as $B_{i,b}$) may differ in size across i and across b. Since each proof should be contained in a predefined part of the codeword we construct below, we must ensure that the right and left sizes of the PCPP have the same size across all invocations. Towards that, we set $C' = C + 10$ and let $K = k \cdot \log^{C'} k$. Given assignments to the PCPP $T_{A_{i,b}} : A_{i,b} \to \Sigma$ and $T_{B_{i,b}} : B_{i,b} \to \Sigma$ construct $\pi_{i,b} = (\pi_{i,b,A}, \pi_{i,b,B})$ as follows:

- $\pi_{i,b,A} = \left(T_{A_{i,b}}^{\lfloor K/|A_{i,b}| \rfloor}, \nu^{K - \lfloor K/|A_{i,b}| \rfloor \cdot |A_{i,b}|} \right)$, that is, repeat $T_{A_{i,b}}$ in full $\lfloor K/|A_{i,b}| \rfloor$-times, and put ν in the rest to make the overall size exactly K. Here, ν is an arbitrary symbol outside of Σ.

- $\pi_{i,b,B} = \left(T_{B_{i,b}}^{\lfloor K/|B_{i,b}| \rfloor}, \nu^{K - \lfloor K/|B_{i,b}| \rfloor \cdot |B_{i,b}|} \right)$, that is, repeat $T_{B_{i,b}}$ in full $\lfloor K/|B_{i,b}| \rfloor$-times, and put ν in the rest to make the overall size exactly K.

Note that the uniform distribution over $[K]$ and the distribution over $[K]$ which is uniform over the subset $[\lfloor K/|A_{i,b}| \rfloor \cdot |A_{i,b}|]$, are $1/\log^{10} k$-close in statistical distance; similarly for $B_{i,b}$. We think of the PCPP as the string $\pi_{i,b} = (\pi_{i,b,A}, \pi_{i,b,B})$ of length $2K$ over alphabet symbols from Σ. We denote by $\pi_{i,b}(w)$ an arbitrary assignment to the PCPP in the completeness case for w. By the perfect completeness of the PCPP, it is guaranteed that such an assignment exists which satisfies all of the constraints with w.

The construction. Let $\Sigma' = \Sigma \times \{0, 1\}$. We define $\mathcal{C} : \{0, 1\}^k \to \Sigma_0^{r \cdot k/\rho} \times (\Sigma')^{2K \cdot k}$ as follows:

$$\mathcal{C}(m) = \mathcal{C}_0(m)^r \pi'_{1,m_1}(\mathcal{C}_0(m)) \cdots \pi'_{k,m_k}(\mathcal{C}_0(m))$$

for $r = 2\rho K$, where the proof π'_{i,m_i} are constructed as follows. We first construct $\pi_{i,m_i}(\mathcal{C}_0(m))$, and to get $\pi'_{i,m_i}(\mathcal{C}_0(m))$ we append the bit m_i to each entry of $\pi_{i,m_i}(\mathcal{C}_0(m))$. This results in doubling the alphabet size of the proof. Overall, we get that a codeword consists of two equal length blocks:

- The first block consists of r repetitions of $\mathcal{C}_0(m)$.

- The second block consists of the concatenation of $\pi'_{i,m_i}(\mathcal{C}_0(m))$ for $i = 1, \ldots, k$, where each entry of the proof has the bit m_i "appended".

Moreover, the second block can be further partitioned into two equal sized blocks consisting of the "A"-vertices and "B"-vertices of each PCPP.

The decoder. Given an index $i \in [k]$ and $w \in \Sigma_0^{r \cdot k/\rho} \times \Sigma^{k \cdot r}$, the decoder $\mathsf{Dec}(w, i)$ proceeds in the following way:

1. Choose a random repetition of out of the r repetitions in the first block of w. Call this w_0.

2. Sample a uniform index b' in $[K]$, and read $\pi'_{i,m_i,B}$ at index b'. If that symbol is ν output $\perp$, and else proceed.

   Recall that $\pi'_{i,m_i,B}$ contains multiple copies of assignments to the right-side B_{i,m_i}. The index b' corresponds to a vertex in $b \in B_{i,m_i}$, and the distribution we chose is $(1/\log^{10} k)$-close to being uniform over B_{i,m_i}. We note that as the supposed value of m_i is not known to the decoder yet, it does not know the correct distribution over the locations that need to be read in the PCPP (as they may be different for each one of the languages $\mathcal{L}_{i,m_i}$). However, since the distribution over the query on the B-side is uniform for each of $\mathcal{L}_{i,m_i}$, the query we made in this step is sampled according to a distribution which is statistically close to the correct distribution.

3. Let x be the bit appended, so that supposedly $x = m_i$.

4. The decoder now knows which language $\mathcal{L}_{i,x}$ was supposedly used for the PCPP encoding, and in particular the decoder can calculate the length of $A_{i,x}$. Thus, the decoder also knows which locations in the A-part of $\pi_{i,x}(\mathcal{C}_0(m))$ are the non-ν ones. The decoder samples a uniformly random index $j \in [\lfloor K/|A_{i,x}|\rfloor]$ and looks at the jth copy of $\pi_{i,x}(\mathcal{C}_0(m))$ from the A-part. The verifier samples $(t, a, \mathbf{b}) \sim \mathcal{P}_{i,x}$ conditioned on $\mathbf{b} = b$, and then queries the locations corresponding to a in the jth copy of the A-part of $\pi_{i,x}(\mathcal{C}_0(m))$ and t in w_0.

5. If the read assignment to a does not have the bit x appended, output $\perp$. Else, remove the appended bit from each of the read symbols and proceed.

6. If the read assignment satisfies the constraint on (t, a, b) of $\Psi_{i,x}$, output x. Otherwise, output $\perp$.

Analysis of the construction. We now analyze the construction. It is easy to see that the code has distance at least $\frac{1-\mu}{2}$ due to the repetitions in the first block. The length and completeness for relaxed local decoding are clear from the corresponding properties of the PCPP.

For the success rate, suppose the codeword, w, is δ-close to some valid codeword w'. Let I_1, I_2, and I_3 be the $\mathcal{C}_0(m)^r$ part, the A-parts of the proofs, and the B-parts of the proofs respectively. The queries of the decoder are distributed as follows. Over a uniformly random index to decode, the marginal distribution of the queries of the decoder is: (1) a uniformly random query in I_1, (2) a query according to a distribution which is $o(1)$-close to uniform over I_2, and (3) a uniformly random query in I_3. Let δ_s be the fraction of locations in I_s on which w and w' differ, so that $\frac{1}{2}|I_1|\delta_1 + \frac{1}{4}|I_2|\delta_2 + \frac{1}{4}|I_3|\delta_3$ is the Hamming-distance between w and w' and hence at most $\delta \cdot 4Kk$. As $|I_1| = 2Kk$, $|I_2| = Kk$ and $|I_3| = Kk$ we get that $\delta_1 + \frac{1}{4}\delta_2 + \frac{1}{4}\delta_3 \leqslant 4\delta$. By the union bound, the probability that any of the queries of the decoder is a location in which w and w' differ is at most

$$\delta_1 + \delta_2 + \delta_3 + o(1) \leqslant 4(\delta_1 + \frac{1}{4}\delta_2 + \frac{1}{4}\delta_3) + o(1) \leqslant 16\delta + o(1).$$

In this case, the decoder acts on w as it would act on w', hence it outputs the correct bit by the completeness. We conclude that the success rate is at least $1 - 16\delta - o(1)$.

We now analyze the soundness of the decoder. Suppose that there is a message $m \in \{0,1\}^k$ such that

$$\mathsf{dist}(w, \mathcal{C}(m)) \leqslant \mu,$$

and fix an $i \in [k]$ to decode. Choosing the copy w_0 of the supposed encoding of m under $\mathcal{C}_0$ from the first block uniformly, we get that,

$$\mathbb{E}_{w_0}[\mathsf{dist}(w_0, \mathcal{C}_0(m))] \leqslant 2\mu.$$

Thus, by Markov's inequality with probability at least $9/10$ we have that the w_0 chosen satisfies

$$\mathsf{dist}(w_0, \mathcal{C}_0(m)) \leqslant 20\mu. \tag{71}$$

Now the decoder, as described above, looks at the proof π' from the second block which is supposedly the proof $\pi'_{i,m_i}(w_0)$. If one of the proof queries has the bit m_i appended, then the decoder will output either m_i or $\perp$, and we are done. Otherwise, they both have the bit $m_i \oplus 1$ appended. In that case, the verifier of the PCPP of $\mathcal{L}_{i,m_i\oplus 1}$ is run, and we lower bound its rejection probability. By (71) and the fact $\mathcal{C}_0$ has distance $1-\mu$, we get that $\mathsf{dist}(w_i, \mathcal{L}_{i,m_i\oplus 1}) \geqslant$

$(1 - \mu) - 20\mu = 1 - 21\mu$. In this case, the decoder will output $\perp$ with probability at least $1 - 2 \cdot (21\mu)^{1/C} - \frac{1}{\log^{10} k}$ for some fixed constant $C > 0$, by the soundness of the PCPP.[5] In particular, for sufficiently large k and sufficiently small μ, the decoder outputs either the correct bit or $\perp$ with probability at least

$$\frac{9}{10} \cdot \left(1 - 2 \cdot (21\mu)^{1/C} - \frac{1}{\log^{10} k}\right) \geqslant \frac{2}{3},$$

where the inequality follows by setting μ to be sufficiently small. $\qquad\square$

Proof of Theorem 1.9. Invoke Theorem 11.3 with the PCPP from Theorem 11.1 and the codes from Lemma 2.1. $\quad\square$

11.4 Improved PSPACE Inapproximability of CSP-Reconfiguration

The reconfiguration inapproximability problem [HO24, CM24, GRW25] studies the complexity of moving between two given solutions to a given CSP instance, by modifying the value of only a single variable at each step, and only going through assignments that satisfy a large fraction of the constraints. More precisely:

Definition 11.4 (Gap$_{1,\varepsilon}$ q-CSP$_\Sigma$ Reconfiguration). *Let $q \geqslant 1$ be an integer, $\varepsilon \in (0,1]$, and Σ be a finite alphabet. An instance of* Gap$_{1,\varepsilon}$ *q-CSP$_\Sigma$ Reconfiguration is specified by a q-CSP$_\Sigma$ instance Φ and two solutions $\sigma, \sigma' \in \Sigma^n$. The task is to distinguish the following two cases:*

- *Yes Case. There exist assignments $\sigma^{(0)}, \ldots, \sigma^{(t)} \in \Sigma^n$ such that $\sigma^{(0)} = \sigma$, $\sigma^{(t)} = \sigma'$, each $\sigma^{(i)}, \sigma^{(i-1)}$ differ on at most one coordinate, and each $\sigma^{(i)}$ satisfies all constraints in Φ.*

- *No Case. There exist no assignments $\sigma^{(0)}, \ldots, \sigma^{(t)} \in \Sigma^n$ such that $\sigma^{(0)} = \sigma$, $\sigma^{(t)} = \sigma'$, each $\sigma^{(i)}, \sigma^{(i-1)}$ differ on at most one coordinate, and each $\sigma^{(i)}$ satisfies at least ε fraction of the constraints in Φ.*

The recent work of [GRW25] establishes a formal connection between PCPPs and hardness of CSP reconfiguration problems, where the better the parameters of the PCPPs are, the better of a gap one obtains for the reconfiguration problem.

Theorem 11.5 (Theorem 2, [GRW25]). *Suppose for some $\eta \in (0,1)$, there is an alphabet Σ_0 such that we have the following two ingredients:*

- *Any error-correcting code $\mathcal{C} : \{0,1\}^n \to \Sigma_0^k$ [6] with relative distance $\eta \in (0,1)$ that is encodable in space $\mathsf{poly}(n)$.*

- *For any constant $\delta < \eta/2$ and every circuit $\varphi : \Sigma_0^n \to \{0,1\}$ of size N there is a q-query PCPP which is constructible in space polynomial in $|\Sigma_0|$, N, $(\delta, \varepsilon(\delta))$-soundness, randomness $O(\log N)$, alphabet size $O_{\delta, |\Sigma_0|}(1)$.*

Then for all $\delta < \eta/2$, there is an alphabet Σ of size $O_{\delta, |\Sigma_0|}(1)$, Gap$_{1,1-\varepsilon(\delta)}$ *$(q+1)$-CSP$_\Sigma$ Reconfiguration is* PSPACE-*hard.*

Since our PCPP from Theorem 11.2 achieves strong relation between the proximity and rejection probability we can use it to instantiate the theorems above with it.

We can now prove Corollaries 1.13 and 1.14.

Proof of Corollary 1.13. Take the code from Lemma 2.1 with distance greater than $1 - \delta$ for some arbitrarily small δ and take the 4-query PCPP of Theorem 11.2, which rejects $\frac{1-\delta}{2}$-far inputs with probability $\frac{1-\delta}{2} + \varepsilon$ for some $\varepsilon = \varepsilon(\delta)$ arbitrarily small relative to δ. Then, the result follows by plugging these two pieces into Theorem 11.5. $\quad\square$

Proof of Corollary 1.14. The result follows by a simple reduction from Gap$_{1,1-\varepsilon}$ $(q+1)$-CSP$_\Sigma$ to Gap$_{1,1-\frac{\varepsilon}{q+1}}$ 2-CSP$_\Sigma$, $\qquad\square$

---

[5]The $\frac{1}{\log^{10} k}$ term is because the query we made to the B-side is only $\frac{1}{\log^{10} k}$-close to uniform in total variation distance.

[6]In [GRW25], a binary code (rather than a code over a larger alphabet) is used. However, an inspection of their argument shows that they only use the fact the *messages* are binary. Thus, the alphabet size of the encoding could be taken as a large constant.

Acknowledgments

References

[AGKM23] Omar Alrabiah, Venkatesan Guruswami, Pravesh K. Kothari, and Peter Manohar. A near-cubic lower bound for 3-query locally decodable codes from semirandom CSP refutation. In *STOC 2023*, pages 1438–1448. Association for Computing Machinery, 2023.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[Alo21] Noga Alon. Explicit expanders of every degree and size. *Comb.*, 41(4):447–463, 2021.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[AS03] Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. *Comb.*, 23(3):365–426, 2003.

[AS21] Vahid R. Asadi and Igor Shinkar. Relaxed Locally Correctable Codes with Improved Parameters. In *ICALP 2021*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:12, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[BBC+22] Alex Block, Jeremiah Blocki, Kuan Cheng, Elena Grigorescu, Xin Li, Yu Zheng, and Minshen Zhu. On relaxed locally decodable codes for hamming and insertion-deletion errors. *arXiv preprint arXiv:2209.08688*, 2022.

[BDG16] Jop Briët, Zeev Dvir, and Sivakanth Gopi. Outlaw distributions and locally decodable codes. *arXiv preprint arXiv:1609.06355*, 2016.

[BGH+06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM Journal on Computing*, 36(4):889–974, 2006.

[BHLM09] Eli Ben-Sasson, Prahladh Harsha, Oded Lachish, and Arie Matsliah. Sound 3-query PCPPs are long. *ACM Trans. Comput. Theory*, 1(2):7:1–7:49, 2009.

[BLM24] Mitali Bafna, Noam Lifshitz, and Dor Minzer. Constant degree direct product testers with small soundness. In *FOCS 2024*, pages 862–869. IEEE, 2024.

[BM24] Mitali Bafna and Dor Minzer. Characterizing direct product testing via coboundary expansion. In *STOC 2024*, pages 1978–1989. ACM, 2024.

[BMVY25] Mitali Bafna, Dor Minzer, Nikhil Vyas, and Zhiwei Yun. Quasi-linear size PCPs with small soundness from HDX. In *STOC 2025*, pages 45–53. ACM, 2025.

[BS08] Eli Ben-Sasson and Madhu Sudan. Short PCPs with polylog query complexity. *SIAM Journal on Computing*, 38(2):551–607, 2008.

[CG18] Clément L. Canonne and Tom Gur. An adaptivity hierarchy theorem for property testing. *Computational Complexity*, 27(4):671–716, 2018.

[CGdW09] Victor Chen, Elena Grigorescu, and Ronald de Wolf. Efficient and error-correcting data structures for membership and polynomial evaluation. *arXiv preprint arXiv:0909.3696*, 2009.

[CGS22]   Alessandro Chiesa, Tom Gur, and Igor Shinkar. Relaxed locally correctable codes with nearly-linear block length and constant query complexity. *SIAM Journal on Computing*, 51(6):1839–1865, 2022.

[CM24]   Karthik C. S. and Pasin Manurangsi. On inapproximability of reconfiguration problems: PSPACE-hardness and some tight NP-hardness results. *Electronic Colloquium on Computational Complexity (ECCC)*, TR24-007, 2024. Available at https://eccc.weizmann.ac.il/report/2024/007/.

[CY22]   Gil Cohen and Tal Yankovitz. Relaxed locally decodable and correctable codes: Beyond tensoring. In *FOCS 2022*, pages 24–35. IEEE, 2022.

[CY24]   Gil Cohen and Tal Yankovitz. Asymptotically-Good RLCCs with $\log^{2+o(1)} n$ Queries. In *CCC 2024*, volume 300 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:16, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[DD24a]   Yotam Dikstein and Irit Dinur. Agreement theorems for high dimensional expanders in the low acceptance regime: The role of covers. In *STOC 2024*, pages 1967–1977. ACM, 2024.

[DD24b]   Yotam Dikstein and Irit Dinur. Agreement theorems for high dimensional expanders in the low acceptance regime: The role of covers. In *STOC 2024*, pages 1967–1977. ACM, 2024.

[DD24c]   Yotam Dikstein and Irit Dinur. Swap cosystolic expansion. In *STOC 2024*, pages 1956–1966. ACM, 2024.

[DFK+11]   Irit Dinur, Eldar Fischer, Guy Kindler, Ran Raz, and Shmuel Safra. PCP characterizations of NP: Toward a polynomially-small error-probability. *Computational Complexity*, 20(3):413–504, 2011.

[DGL23]   Marcel Dall'Agnol, Tom Gur, and Oded Lachish. A structural theorem for local algorithms with applications to coding, testing, and verification. *SIAM Journal on Computing*, 52(6):1413–1463, 2023.

[DGMT22]   Marcel Dall'Agnol, Tom Gur, Subhayan Roy Moulik, and Justin Thaler. Quantum proofs of proximity. *Quantum*, 6:834, 2022.

[DH13]   Irit Dinur and Prahladh Harsha. Composition of low-error 2-query PCPs using decodable PCPs. *SIAM Journal on Computing*, 42(6):2452–2486, 2013.

[DHK15]   Irit Dinur, Prahladh Harsha, and Guy Kindler. Polynomially low error PCPs with polyloglog n queries via modular composition. In *STOC 2015*, pages 267–276, 2015.

[Din07]   Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM*, 54(3), 2007.

[DM11]   Irit Dinur and Or Meir. Derandomized parallel repetition via structured PCPs. *Computational Complexity*, 20(2):207–327, 2011.

[DR06]   Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. *SIAM Journal on Computing*, 36(4):975–1024, 2006.

[Efr12]   Klim Efremenko. 3-query locally decodable codes of subexponential length. *SIAM Journal on Computing*, 41(6):1694–1703, 2012.

[FGL+96]   Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996.

[Gas24]   William Gasarch. Open problems column. *SIGACT News*, 55(4):32–48, December 2024.

[GG18]   Oded Goldreich and Tom Gur. Universal locally testable codes. *Chicago J. Theor. Comput. Sci.*, 2018.

[GG21]   Oded Goldreich and Tom Gur. Universal locally verifiable codes and 3-round interactive proofs of proximity for CSP. *Theoretical computer science*, 878:83–101, 2021.

[GGK19]   Oded Goldreich, Tom Gur, and Ilan Komargodski. Strong locally testable codes with relaxed local decoders. *ACM Transactions on Computation Theory (TOCT)*, 11(3):1–38, 2019.

[GGS25]   Guy Goldberg, Tom Gur, and Sidhant Saraogi. Nearly tight lower bounds for relaxed locally decodable codes via robust daisies. *ECCC preprint TR25-192*, 2025.

[GI05]   Venkatesan Guruswami and Piotr Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Trans. Inf. Theory*, 51(10):3393–3400, 2005.

[GKMM25] Elena Grigorescu, Vinayak M Kumar, Peter Manohar, and Geoffrey Mon. Relaxed vs. full local decodability with few queries: Equivalence and separations for linear codes. *arXiv preprint arXiv:2511.02633*, 2025.

[GL21]   Tom Gur and Oded Lachish. On the power of relaxed local decoding algorithms. *SIAM Journal on Computing*, 50(2):788–813, 2021.

[GM21]   Aurélien Gribinski and Adam W. Marcus. Existence and polynomial time construction of biregular, bipartite ramanujan graphs of all degrees. *arXiv preprint*, arXiv:2108.02534, 2021.

[Gol10]   Oded Goldreich. Property testing. *Lecture Notes in Comput. Sci*, 6390, 2010.

[Gol24]   Oded Goldreich. On the relaxed LDC of BGHSV: A survey that corrects the record. *Electron. Colloquium Comput. Complex.*, TR24-078, 2024.

[GR18]   Tom Gur and Ron D Rothblum. Non-interactive proofs of proximity. *Computational Complexity*, 27(1):99–207, 2018.

[GRR20]   Tom Gur, Govind Ramnarayan, and Ron Rothblum. Relaxed locally correctable codes. *Theory of Computing*, 16(1):1–68, 2020.

[GRW25]   Venkatesan Guruswami, Xuandi Ren, and Kewen Wu. PCPP-based reconfiguration inapproximability: Query complexity vs. soundness gap trade-offs, 2025.

[HO24]   Shuichi Hirahara and Naoto Ohsaka. Probabilistically checkable reconfiguration proofs and inapproximability of reconfiguration problems. In *STOC 2024*, pages 1435–1445, Vancouver, BC, Canada, June 2024. ACM.

[KM24]   Vinayak M. Kumar and Geoffrey Mon. Relaxed local correctability from local testing. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 1585–1593, New York, NY, USA, 2024. Association for Computing Machinery.

[KT00]   Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In F. Frances Yao and Eugene M. Luks, editors, *STOC 2000*, pages 80–86. ACM, 2000.

[Mie09]   Thilo Mie. Short PCPPs verifiable in polylogarithmic time with O(1) queries. *Annals of Mathematics and Artificial Intelligence*, 56(3-4):313–338, 2009.

[MR10]   Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *Journal of the ACM*, 57(5), June 2010.

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

[RS97]   Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *STOC 1997*, pages 475–484. ACM, 1997.

[RS06]   Ron M Roth and Vitaly Skachek. Improved nearly-mds expander codes. *IEEE transactions on information theory*, 52(8):3650–3661, 2006.

[RVW02] Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of Mathematics*, 155(1):157–187, 2002.

[Spi96] Daniel A Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 42(6):1723, 1996.

[WDW05] Stephanie Wehner and Ronald De Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In *ICALP 2005*, pages 1424–1436. Springer, 2005.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *J. ACM*, 55(1), February 2008.

A   The Final Inner dPCP (Hadamard dPCP)

In this subsection we prove Theorem 10.3. We do so in three steps:

- We present the notion of right-assignment decoding soundness, which most dPCP constructions in the literature satisfy. We show that it implies our notion of list-decoding soundness.

- We recall the definition of robust dPCPs from [DH13]. A construction of such dPCPs yields the desired structure of the left-vertices in the inner dPCPs for composition (see Theorem 8.3).

- Then we show that the folklore Hadamard dPCP (taken from [BMVY25, Lemma 7.7]) satisfies all of our requirement for the inner dPCP.

A.1   Right-Assignment Decoding to Left-Assignment Decoding

Below is the more commonly used list-decoding soundness for dPCPs. In it the list is constructed from a fixed assignment to the right side of the dPCP, as opposed to Definition 3.3 wherein it is constructed from a fixed assignment to the left side of the dPCP.

Definition A.1. *Let $\mathcal{D}$ be a decodable PCP for $\mathcal{L}$. Given $\varepsilon \in [0,1]$, $L \in \mathbb{N}$ we say that $\mathcal{D}$ satisfies* right-assignment (L, ε)-list-decoding soundness *if the following holds. For any right assignment $T_B : B \to \Sigma_B$ there exists a list $\{w_1, \ldots, w_L\} \subseteq \mathcal{L}$ such that for any $T_A : A \to \Sigma_A$,*

$$\Pr_{\substack{t \in [n] \\ (a,b) \sim \mathcal{P}_t}} [\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge D_t(a, T_A[a]) \notin \{(w_i)_t\}_{i \in [L]}] \leqslant \varepsilon.$$

In the definition above, the right-assignment T_B is fixed and induces a list of candidate words in the language, whereas in our definition of list-decoding soundness it is the left-assignment T_A that is fixed. In the following we show a simple transformation between the two notations:

Lemma A.2. *Suppose a language $\mathcal{L}$ has a projection dPCP $\mathcal{D} = \big(A \cup B, E, \Sigma_A, \Sigma_B, \{\mathcal{P}_t\}_{t \in [n]}, \{D_t\}_{t \in [n]}\big)$ with right-assignment-decoding (L, ε)-list-decoding soundness. Then $\mathcal{D}$ also satisfies $(L \cdot \varepsilon^{-1/5}, 2\varepsilon^{1/5})$-list-decoding soundness for the language $\mathcal{L}$.*

Proof. First, note that since $\mathcal{D}$ is a projection dPCP, for each $a \in A$ and an assignment to it σ, each neighbour $b \in \Gamma(a)$ has a unique assignment that would satisfy the constraint $\Phi_{(a,b)}$, and we denote it by $(\sigma)_b$ (the vertex a will always be clear from context). Let $\mathcal{Q}$ be the complete decoding distribution of $\mathcal{D}$.

Fix an assignment $T_A : A \to \Sigma_A$, and consider the following candidate list: for each $b \in B$, let $\mathsf{cand}(b)$ be the set of symbols $\sigma \in \mathbb{F}_q$ such that $\Pr_{a \sim \mathcal{Q}(\circ, \cdot, b)}[T_A[a]_b = \sigma] \geqslant \varepsilon^{1/5}$. One can find $J = 1/\varepsilon^{1/5}$ assignments $T'_1, \ldots T'_J : B \to \Sigma_B$, such that for every $b \in B$ and every $\sigma \in \mathsf{cand}(b)$ there is some $j \in [J]$ such that $T'_j[b] = \sigma$. We use the assignments $T'_1, ..., T'_J$ to invoke the right-assignment list-decoding property of the dPCP with respect to fixed

right assignments (and with T_A being the left assignment). Thus, for each j we get a list $L'_j = \{w_1^j, \ldots, w_L^j\} \subseteq \mathcal{L}$ such that

$$\Pr_{\substack{t\sim[n], \\ (a,b)\sim\mathcal{P}_t}} [\Phi_{(a,b)}(T_A[a], T'_j[b]) = 1 \wedge D_t(a, T_A[a]) \notin \{(w_i^j)_t\}_{i\in[L]}] \leqslant \varepsilon.$$

Define the list list $L' = \cup_{j\in[J]}L'_j$. To complete the proof, we will show that for any $T_B : B \to \Sigma_B$ we have

$$\Pr_{\substack{t\sim[n], \\ (a,b)\sim\mathcal{P}_t}} [\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge D_t(a, T_A[a]) \notin \{w_i^j[t]\}_{i\in[L],j\in[J]}] \leqslant J \cdot \varepsilon + \varepsilon^{1/5} \leqslant 2\varepsilon^{1/5}.$$

First, note that the probability on the left hand side can be written as

$$\Pr_{\substack{t\sim[n], \\ (a,b)\sim\mathcal{P}_t}} [\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge D_t(a, T_A[a]) \notin \{w_i^j[t]\}_{i\in[L],j\in[J]}]$$

$$= \Pr_{\substack{t\sim[n], \\ (a,b)\sim\mathcal{P}_t}} [\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge D_t(a, T_A[a]) \notin \{w_i^j[t]\}_{i\in[L],j\in[J]} \wedge T_A[a]_b \in \mathsf{cand}(b)]$$

$$+ \Pr_{\substack{t\sim[n], \\ (a,b)\sim\mathcal{P}_t}} [\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge D_t(a, T_A[a]) \notin \{w_i^j[t]\}_{i\in[L],j\in[J]} \wedge T_A[a] \notin \mathsf{cand}(b)].$$

The second term is bounded by $\varepsilon^{1/5}$ by the definition of the $\mathsf{cand}(b)$ list. For the first term, we use union bound:

$$\Pr_{\substack{t\sim[n], \\ (a,b)\sim\mathcal{P}_t}} [\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge D_t(a, T_A[a]) \notin \{w_i^j[t]\}_{i\in[L],j\in[J]} \wedge T_A[a]_b \in \mathsf{cand}(b)]$$

$$= \Pr_{\substack{t\sim[n], \\ (a,b)\sim\mathcal{P}_t}} [\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge D_t(a, T_A[a]) \notin \{w_i^j[t]\}_{i\in[L],j\in[J]} \wedge (\cup_{j\in[J]}T_A[a]_b = T'_j[b])]$$

$$\leqslant \sum_{j\in[J]} \Pr_{\substack{t\sim[n], \\ (a,b)\sim\mathcal{P}_t}} [\Phi_{(a,b)}(T_A[a], T_B[b]) = 1 \wedge D_t(a, T_A[a]) \notin \{w_i^j[t]\}_{i\in[L],j\in[J]} \wedge T_A[a]_b = T'_j[b])]$$

$$= \sum_{j\in[J]} \Pr_{\substack{t\sim[n], \\ (a,b)\sim\mathcal{P}_t}} [\Phi_{(a,b)}(T_A[a], T'_j[b]) = 1 \wedge D_t(a, T_A[a]) \notin \{w_i^j[t]\}_{i\in[L],j\in[J]})]$$

$$\leqslant J \cdot \varepsilon.$$

In this third transition we used the fact that if $\Phi_{(a,b)}(T_A[a], T_B[b]) = 1$ and $T_A[a]_b = T'_j[b]$, then we must have $T_B[b] = T'_j[b]$. In the last transition we used the right-assignment (L, ε)-list-decoding soundness of $\mathcal{D}$. $\qquad\square$

A.2   Going from Robust dPCPs to dPCPs

In order to present the Hadamard dPCP from prior work, we need to first present an alternative definition of dPCPs, called robust dPCPs, introduced by [DH13]. The main difference between robust dPCPs and our notion of dPCP is that robust dPCPs have a different notion of list-decoding soundness.

Definition A.3 (Robust Decodable PCPs). *Let $\varphi \colon \Sigma_0^n \to \{0,1\}$ be a circuit of size N. A PCP decoder for the language* $\mathsf{SAT}(\varphi)$ *over a proof alphabet Σ is a probabilistic polynomial-time algorithm $\mathcal{D}$ that on an input circuit $\varphi : \Sigma_0^n \to \{0,1\}$ of size N and an index $j \in [n]$, tosses $r = r(N)$ random coins to sample from a distribution $\mathcal{P}_j$ supported over tuples containing (I, f) which are:*

(1) A sequence of $q = q(N)$ locations $I = (i_1, \ldots, i_q)$ in a proof of length $m(N)$ over the alphabet Σ; the parameter q is referred to as the query complexity *of $\mathcal{D}$.*

(2) A (local decoding) function $f : \Sigma^q \to \Sigma_0 \cup \{\bot\}$ whose corresponding circuit has size at most $s(N)$, referred to henceforth as the decision complexity *of the decoder.*

For functions $\delta : \mathbb{Z}^+ \to [0, 1]$ and $L : \mathbb{Z}^+ \to \mathbb{Z}^+$, we say that a PCP decoder $\mathcal{D}$ is a dPCP for $\mathsf{SAT}(\varphi)$ with soundness error δ and list size L if the following completeness and soundness properties hold for every circuit $\varphi : \Sigma_0^n \to \{0, 1\}$ of size N:

- **Completeness.** For any $y \in \Sigma_0^n$ such that $\varphi(y) = 1$, there exists a proof $\pi \in \Sigma^m$, also called a decodable PCP, such that
$$\Pr_{j, I, f}[f(\pi_I) = y_j] = 1,$$
where $j \in [n]$ is chosen uniformly at random and I, f are distributed according to $\mathcal{P}_j$ and the verifier's random coins.

- **Soundness.** For any $\pi \in \Sigma^m$, there is a list of $0 \leqslant \ell \leqslant L$ strings $y^1, \ldots, y^\ell$ satisfying $\varphi(y^i) = 1$ for all i, and furthermore
$$\Pr_{j \in [n], (I, f) \sim \mathcal{P}_j} \left[f(\pi_I) \notin \{\perp, y_j^1, \ldots, y_j^\ell\} \right] \leqslant \delta.$$

- **Robust Soundness.** We say that $\mathcal{D}$ is a robust dPCP system for $\mathsf{SAT}(\varphi)$ with robust soundness error δ if the soundness criterion can be strengthened to the following robust condition:
$$\mathbb{E}_{j \in [n], (I, f) \sim \mathcal{P}_j} \left[\mathrm{agr}(\pi_I, \mathrm{BAD}(f)) \right] \leqslant \delta,$$
where $\mathrm{BAD}(f) := \{w \in \Sigma^q \mid f(w) \notin \{\perp, y_j^1, \ldots, y_j^\ell\}\}$.

We have the following correspondence between robust dPCPs as in Definition A.3 and our more usual notion of dPCPs.

Lemma A.4. *Suppose a language $\mathcal{L} \subseteq \Sigma_0^n$ has a robust dPCP with the following parameters*

- **Length.** m.

- **Query complexity.** d_1.

- **Proof alphabet.** Σ.

- **Randomness Complexity.** r.

- **Completeness.** *Perfect completeness.*

- **Soundness.** *Robust soundness error ε with list size L.*

Then $\mathcal{L}$ has a dPCP with the following parameters

- **Length.** *The left vertex set has the form $[n] \times [2^r]$ and the length is $m + n \cdot 2^r$.*

- **Alphabets.** *The left alphabet is a subset of Σ^{d_1} and the right alphabet is Σ.*

- **Degree.** *Decoding degree is 1.*

- **Complete Decoding distribution.** *Let $\mathcal{Q}$ be the complete decoding distribution and let the left and right sides be A and B respectively. Then the complete decoding distribution satisfies the following.*

  - *The marginal over A is uniform.*
  - *For each $a \in A$, the distribution $\mathcal{Q}(\circ, a, \cdot)$ is uniform over d_1 vertices in B.*

- **Completeness.** *Perfect completeness.*

- **Soundness.** *$(L \cdot \varepsilon^{-1/5}, 2\varepsilon^{1/5})$-list-decoding soundness.*

Proof. Fix a robust dPCP for $\mathcal{L}$ as in the premise. We construct a dPCP $\mathcal{D}$ for $\mathcal{L}$, whose parts are

$$\mathcal{D} = (A \cup B, E, \Sigma_A, \Sigma_B, \{\Phi_e\}_{e \in E}, \{D_t\}_{t \in [n]}, \{\mathcal{P}_t\}_{t \in [n]}),$$

as follows. The right vertices correspond to the proof locations of the robust dPCP. The left vertices are labeled as $[n] \times [2^r]$ and they correspond to ever possible pair of index $t \in [n]$ and randomness $\ell \in [2^r]$ when choosing the queries given index t. By the randomness complexity of the robust dPCP, each query set corresponds to one of 2^r random strings. The edges e correspond to all pair $(t, \ell) \in A$ and $b \in B$ such that the decoder of the robust dPCP given index t and randomness ℓ queries the location corresponding to b in the proof. The left alphabet is Σ^{d_1}. For the vertex $(t, \ell) \in A$, we think of an alphabet symbol as corresponding to answer to the d_1 queries made by the robust dPCP given index t and randomness ℓ. It also corresponds to a labeling of its d_1 neighbors on the right side. We constrain the alphabet to only consist of values which would have made the robust dPCP verifier accept. The right alphabet is Σ, and each constraint between $(t, \ell) \in A$ and $b \in B$ checks that both assignments agree on the value assigned to b. The decoder is the same as in the robust dPCP. That is, it thinks of the symbol assigned to $(t, \ell) \in A$ as the d_1 answers to its queries in the robust dPCP and decodes accordingly. Finally, for each $t \in [n]$, the decoding distribution $\mathcal{P}_t$ chooses $\ell \in [n]$ randomly, chooses one of d_1 many $b \in B$ adjacent to (t, ℓ) uniformly at random, and outputs $(t, (t, \ell), b)$.

It is straightforward to check that $\mathcal{D}$ has the desired length, alphabets, decoding degree, and properties for the complete decoding distribution. For perfect completeness, for any $w \in \mathcal{L}$, using the perfect completeness of the robust dPCP we get that there is an assignment to B such that the robust dPCP always accepts and decodes to w_t. Then, we assign the left side according to the assignment to B. That is, for each $(t, \ell) \in A$, the assignment to (t, ℓ) consists of the d_1 symbols from Σ assigned to its d_1 neighbors. It is straightforward to check that all of the constraints of $\mathcal{D}$ are satisfied and the decoder for $\mathcal{D}$ decodes w_t with probability 1 for all $t \in [n]$. Finally, it is straightforward to see that the definition of robust soundness corresponds directly to right-assignment (L, ε)-list-decoding soundness. Then, the desired list decoding soundness follows by applying Lemma A.2. $\qquad\square$

A.3 The Hadamard dPCP

We are now ready to present the Hadamard dPCP and prove Theorem 10.3. The result follows by plugging the robust (Hadamard) dPCP presented in [BMVY25] into the transformation of Lemma A.4.

Lemma A.5 (Lemma 7.7 from [BMVY25]). *For every $T, N \in \mathbb{N}$ and $\varepsilon > 0$ the following holds. Let $\varphi \colon \Sigma_0^n \to \{0, 1\}$ be a circuit of size at most N over an alphabet of size at most T. There exists $q = 1/\varepsilon^{O(1)}$ such that the language $\mathsf{SAT}(\varphi)$ has a robust dPCP (according to Definition A.3) with the following parameters:*

1. **Length.** $m(N) = q^{O(N^2)}$.

2. **Alphabet Size.** q.

3. **Randomness complexity.** $r(N) = O(N^2 \log(q))$.

4. **Query Complexity.** $q^{O(\log |\Sigma_0|)}$.

5. **Soundness.** *Robust soundness ε with list size $1/\varepsilon^{O(1)}$.*

Proof of Theorem 10.3. The result follows by plugging the robust dPCP of Lemma A.5 into the transformation of Lemma A.4. The projection decision complexity and decoding complexity follow from the fact that each can be implemented as a circuit which outputs 0 or 1 and has an input consisting of at most two symbols from alphabets of size at most $O_{\varepsilon, T}(1)$. Finally, the complete decoding distribution condition follows from the complete decoding distribution guarantee of Lemma A.4. $\qquad\square$