Electronic Colloquium on Computational Complexity, Report No. 85 (2015)

Polynomially Low Error PCPs with polyloglog n Queries via Modular Composition*

Irit Dinur[†] Prahladh Harsha[‡] Guy Kindler[§]

May 23, 2015

Abstract

We show that every language in NP has a PCP verifier that tosses $O(\log n)$ random coins, has perfect completeness, and a soundness error of at most $1/\text{poly}(n)$, while making at most $O(\text{poly} \log \log n)$ queries into a proof over an alphabet of size at most $n^{1/\text{poly} \log \log n}$. Previous constructions that obtain $1/\text{poly}(n)$ soundness error used either poly $\log n$ queries or an exponential sized alphabet, i.e. of size 2^{n^c} for some $c > 0$. Our result is an exponential improvement in both parameters simultaneously.

Our result can be phrased as a polynomial-gap hardness for approximate CSPs with arity poly $\log \log n$ and alphabet size $n^{1/\text{poly} \log n}$. The ultimate goal, in this direction, would be to prove polynomial hardness for CSPs with constant arity and polynomial alphabet size (aka the sliding scale conjecture for inverse polynomial soundness error).

Our construction is based on a modular generalization of previous PCP constructions in this parameter regime,which involves a composition theorem that uses an extra 'consistency' query but maintains the inverse polynomial relation between the soundness error and the alphabet size.

Our main technical/conceptual contribution is a new notion of soundness, which we refer to as *distributional soundness*, that replaces the previous notion of "list decoding soundness", and that allows us to prove a modular composition theorem with tighter parameters. This new notion of soundness allows us to invoke composition a super-constant number of times without incurring a blow-up in the soundness error.

[*] A preliminary version of this paper appeared in the *Proc. 47th ACM Symp. on Theory of Computing (STOC)*, 2015 [DHK15].

[†] Weizmann Institute of Science, ISRAEL. email: `irit.dinur@weizmann.ac.il`. Research supported in part by a ISF-UGC grant 1399/4 and by an ERC grant 239985.

[‡] Tata Institute of Fundamental Research (TIFR), Mumbai, INDIA. email: `prahladh@tifr.res.in`. Research supported in part by ISF-UGC grant 1399/4. Part of this work was done while visiting the Simons Institute for the Theory of Computing, UC Berkeley.

[§] The Hebrew University of Jerusalem, ISRAEL. email: `gkindler@cs.huji.ac.il`. Research supported in part by an Israeli Science Foundation grant no. 1692/13 and by US-Israel Binational Science Foundation grant no. 2012220. Part of this work was done while visiting the Simons Institute for the Theory of Computing, UC Berkeley.

ISSN 1433-8092

1 Introduction

Probabilistically checkable proofs (PCPs) provide a proof format that enables verification with only a small number of queries into the proof such that the verification, though probabilistic, has only a small probability of error. This is formally captured by the following notion of a probabilistic verifier.

Definition 1.1 (PCP Verifier). *A PCP verifier V for a language L is a polynomial time probabilistic algorithm that behaves as follows: On input x, and oracle access to a (proof) string Π (over an alphabet Σ), the verifier reads the input x, tosses some random coins R, and based on x and R computes a (local) window $I = (i_1, \ldots, i_q)$ of q indices to read from Π, and a (local) predicate $\varphi : \Sigma^q \to \{0,1\}$. The verifier then accepts iff $\varphi(\Pi|_I) = 1$.*

- *The verifier has* perfect completeness*: if for every $x \in L$, there is a proof Π that is accepted with probability 1. I.e., $\exists \Pi,\ \Pr_R[\varphi(\Pi|_I) = 1] = 1$.*
- *The verifier has* soundness error $\delta < 1$*: if for any $x \notin L$, every proof Π is accepted with probability at most δ. I.e., $\forall \Pi,\ \Pr_R[\varphi(\Pi|_I) = 1] \le \delta$.*

The celebrated PCP Theorem [AS98, ALM+98] states that every language in NP has a verifier that has perfect completeness and soundness error bounded by a constant $\delta < 1$, while using only a logarithmic number of random coins, and reading only $q = O(1)$ proof bits. Naturally (and motivated by the fruitful connection to inapproximability due to Feige *et al.* [FGL+96]), much attention has been given to obtaining PCPs with desirable parameters, such as a small number of queries q, smallest possible soundness error δ, and smallest possible alphabet size $|\Sigma|$.

How small can we expect the soundness error δ to be? There are a couple of obvious limitations. First observe that the soundness error δ cannot be smaller than $1/\text{poly}(n)$ just because there are only $\text{poly}(n)$ different random choices for the verifier and at least one of the corresponding local predicates must be satisfiable[1]. Next, note that if the verifier reads a total of k bits from the proof (namely, $q \log |\Sigma| \le k$), the soundness error cannot be smaller than 2^{-k}, just because a random proof is expected to cause the verifier to accept with at least this probability.

The best case scenario is thus if one can have the verifier read $k = O(\log n)$ bits from the proof and achieve a soundness error of $1/2^k = 1/\text{poly}(n)$. Indeed, the following is well known (obtained by applying a randomness efficient sequential repetition to the basic PCP Theorem):

Theorem 1.2 (PCP theorem + randomness efficient sequential repetition). *For every integer k, every language in NP has a PCP verifier that tosses at most $O(k + \log n)$ random coins, makes $q = O(k)$ queries into a proof over the Boolean alphabet $\{0,1\}$, has perfect completeness, and soundness error $\delta = 2^{-k}$.*

In particular, setting $k = \log n$ we get $q = O(\log n)$ and $\delta = 1/poly(n)$.

This theorem gives a ballpark optimal tradeoff (up to constants) between soundness error and the number of bits read from the proof. However it does not achieve a *small number of queries*, a fundamental requirement that is important, among other things, for hardness of approximation. The goal of constructing a PCP with both a small error and a small number of queries turns out to be much more challenging and has attracted considerable attention. This was first formulated by Bellare *et al.* [BGLR93] as the "sliding scale" conjecture.

[1] One may assume that every local predicate φ is satisfiable. Otherwise the question of "$x \overset{?}{\in} L$" reduces to the question of whether φ is satisfiable for *any* of the predicates computed by the verifier. This cannot occur without a collapse of NP into $\text{NTIME}(q \log |\Sigma|)$.

Conjecture 1.3 (Sliding Scale Conjecture [BGLR93])**.** *For any* $\frac{1}{poly(n)} \leq \delta < 1$*, every language in NP has a PCP verifier that tosses* $O(\log n)$ *random coins, makes* $q = O(1)$ *queries*[2] *into a proof over an alphabet* Σ *of size* $poly(1/\delta)$*, has perfect completeness, and soundness error* δ*.*

As we describe shortly below, this conjecture is known to hold for $1 > \delta \geq 2^{-(\log n)^{1-\varepsilon}}$, namely where δ can be made 'almost' polynomially small. The interesting regime, that has remained open for two decades, is that of (inverse) polynomially small δ. This is the focus of our work. Our main goal is to find the smallest q and $|\Sigma|$ parameters for which we can get δ to be polynomially small. Our main result is the following.

Main Theorem 1.4. *Every language in NP has a PCP verifier that tosses* $O(\log n)$ *random bits, makes* $q = (\log\log n)^{O(1)}$ *queries into a proof over an alphabet* Σ *of size* $|\Sigma| = n^{1/(\log\log n)^{O(1)}}$*, has perfect completeness, and soundness error* $\delta = 1/poly(n)$*.*

Previous PCP constructions require at least $(\log n)^{\Omega(1)}$ queries in order to achieve polynomially small error (and this remains true even for constructions that are allowed quasi-polynomial size, see further discussion at the end of this introduction).

The first works making progress towards this conjecture are due to Raz and Safra [RS97], and Arora and Sudan [AS03], and rely on the classical (algebraic) constructions of PCPs. They prove the conjecture for all δ such that $\delta \geq 2^{-(\log n)^{\beta}}$ for some constant $\beta > 0$. These ideas were then extended by Dinur *et al.* [DFK+11] with an elaborate composition-recursion structure, proving the conjecture for all $\delta \geq 2^{-(\log n)^{1-\varepsilon}}$ for *any* $\varepsilon > 0$. The small catch here is that the number of queries grows as ε approaches 0. The exact dependence of q on ε was not explicitly analyzed in [DFK+11], but we show that it can be made $O(1/\varepsilon)$ while re-deriving their result.

Theorem 1.5 ([DFK+11])**.** *For every* $\varepsilon > 0$ *and* $\delta = 2^{-(\log n)^{1-\varepsilon}}$*, every language in NP has a PCP verifier that tosses* $O(\log n)$ *random coins, makes* $q = O(1/\varepsilon)$ *queries into a proof over an alphabet* Σ *of size* $|\Sigma| = 1/poly(\delta)$*, has perfect completeness, and has soundness error* δ*.*

The focus of [DFK+11] was on a constant number of queries but their result can also be applied towards getting polynomially small error with a non-trivially small number of queries. This is done by combining it with sequential repetition. We get,

Corollary 1.6 ([DFK+11] + randomness efficient sequential repetition)**.** *For every* $\varepsilon > 0$*, every language in NP has a PCP verifier that tosses* $O(\log n)$ *random coins, makes* $q = O((\log n)^{\varepsilon}/\varepsilon)$ *queries into a proof over an alphabet* Σ *of size* $|\Sigma| = 2^{(\log n)^{1-\varepsilon}}$*, has perfect completeness, and has soundness error* $\delta = 1/poly(n)$*.*

Corollary 1.6 describes the previously known best result in terms of minimizing the number of queries subject to achieving a polynomially small error and using at most a logarithmic amount of randomness. Whereas in Corollary 1.6 the number of queries is $q = (\log n)^{\varepsilon}$, our Main Theorem 1.7 requires only $q = \text{poly}\log\log n$ queries.

PCP Composition and dPCPs

Like in recent improved constructions of PCPs [BGH+06, DR06, BS08, Din07, MR10, DH13], our main theorem is obtained via a better understanding of composition. All known constructions

[2] It is even conjectured that this constant can be made as low as 2.

of PCPs rely on proof composition. This paradigm, introduced by Arora and Safra [AS98], is a recursive procedure applied to PCP constructions to reduce the alphabet size. The idea is to start with an easier task of constructing a PCP over a very large alphabet Σ. Then, proof composition is applied (possibly several times over) to PCPs over the large alphabet to obtain PCPs over a smaller (even binary) alphabet, while keeping the soundness error small.

In the regime of high soundness error (greater than $1/2$), composition is by now well understood using the notion of *PCPs of proximity* [BGH+06] (called *assignment testers* in [DR06]) (see also [Sze99]). The idea is to bind the PCP proof of a statement to an NP witness for it, so that the verifier not only checks that the statement is correct but also that the given witness is (close to) a valid one. This extension allows one to prove a modular composition theorem, which is oblivious to the inner makings of the PCPs being composed. This modular approach has facilitated alternate proofs of the PCP theorem and constructions of shorter PCPs [BGH+06, BS08, Din07]. However, the notion of a PCP of proximity, or assignment tester, is not useful for PCPs with low-soundness error. The reason is that for small δ we are in a "list decoding regime", in that the PCP proof can be simultaneously correlated with more than one valid NP witness.

The works mentioned earlier [RS97, AS03, DFK+11] addressed this issue by using a notion of local list-decoding. This was called a local-reader in [DFK+11] and formalized nicely as a locally-decode-or-reject-code (LDRC) by Moshkovitz and Raz [MR10]. Such a code allows "local decoding" in that for any given string w there is a list of valid codewords $\{c_1, \ldots, c_L\}$ such that when the verifier is given a tuple of indices $j_1, \ldots, j_k$, then but for an error probability of δ, the verifier either rejects or outputs $(c_i)|_{j_1,\ldots,j_k}$ for some $i \in [L]$.

Decodable PCPs (dPCPs) Dinur and Harsha [DH13] introduced the notion of a PCP decoder (dPCP), which extends the earlier definitions of LDRCs and local-readers from codes to PCP verifiers. A PCP decoder is like a PCP verifier except that it also gets as input an index j (or a tuple of indices). The PCP decoder is supposed to check that the proof is correct, and to also return the j-th symbol of the NP witness encoded by the PCP proof. As in previous work, the soundness guarantee for dPCPs is that for any given proof Π there is a *short* list of valid NP witnesses $\{x_1, \ldots, x_L\}$ such that except with probability δ the verifier either rejects or outputs $(x_i)|_j$ for some x_i in the list.

The main advantage of dPCPs is that they allow a modular composition theorem in the regime of small soundness error. The composition theorem proved by Dinur and Harsha [DH13] was a two-query composition theorem, generalizing from the ingenious construction of Moshkovitz and Raz [MR10]. The two-query requirement is a stringent one, and in this construction it inherently causes an exponential increase in the alphabet, so that instead of $|\Sigma| = \text{poly}(1/\delta)$ one gets a PCP with $|\Sigma| = \exp(\text{poly}(1/\delta))$.

In this work we give a different (modular) dPCP composition theorem. Essentially, our theorem is a modular generalization of the composition method, as done implicitly in previous works [RS97, AS03, DFK+11], which uses an extra 'consistency' query but maintains the inverse polynomial relation between δ and $|\Sigma|$.

We remark that unlike recent PCP constructions [BGH+06, DR06, MR10, DH13] which recurse on the large (projection) query of the outer PCP, our composition recurses on the entire test as was done originally by Arora and Safra [AS98]. This aspect of the composition is explained and abstracted nicely in [Mos14].

Distributional Soundness Our main technical/conceptual contribution is a new notion of soundness, which we refer to as *distributional soundness*, which replaces the previous notion of *list decoding soundness* described above, and allows us to apply a non-constant number of compositions without a blowup in the error.

We say that a verifier has *distributional soundness* δ if its output is "δ-indistinguishable" from the output of an idealized verifier. The idealized verifier has access to a distribution $\widetilde{\Pi}$ over *valid proofs* or $\perp$. When it is run with random coins R it samples a proof $\widetilde{\Pi}(R)$ from this distribution and either rejects if $\widetilde{\Pi}(R) = \perp$ or outputs what the actual verifier would output when given access to $\widetilde{\Pi}(R)$. By δ-indistinguishable, we mean that there is a coupling between the actual verifier and the idealized verifier, such that the probability that the verifier does not reject *and* its output differs from the output of the idealized verifier, is at most δ.

The advantage of moving from list decoding soundness to distributional soundness, is that it removes the extra factor of L_{in} (the list size) incurred in previous composition analyses. Recall that, e.g. in the composition theorem of Dinur-Harsha [DH13], one takes an outer PCP with soundness δ_{out} and an inner PCP decoder with soundness δ_{in} and out comes a PCP with soundness $\delta_{\text{in}} + L_{\text{in}} \cdot \delta_{\text{out}}$. This is true in all (including implicit) prior composition analyses. When making only a constant number of composition steps, this is not an issue, but when the number t of composition steps grows, the soundness is at least $(L_{\text{in}})^t \cdot \delta_{\text{out}}$ and this is too expensive for the parameters we seek. Using distributional soundness, we prove that the composition of a PCP with soundness error δ_{out} and a dPCP with soundness error δ_{in} yields a PCP with soundness error $\delta_{\text{out}} + \delta_{\text{in}} + \eta$ where η is an error term that is related to the distance of an underlying error correcting code, and can be controlled easily. Thus, after t composition steps the soundness error will only be $O(t(\delta + \eta))$.

We remark that this notion of soundness, though new, is satisfied by most of the earlier PCP constructions (at least the ones based on the low-degree test). In particular, it can be shown that list-decoding soundness and very good error-correcting properties of the PCP imply distributional soundness.

Proof Overview

At a high level, our main theorem is derived by adopting the recursive structure of the construction in [DFK+11]. The two main differences are the use of our modular composition theorem, and the soundness analysis that relies on the notion of distributional soundness[3].

We fix a field $\mathbb{F}$ at the outset and use the same field throughout the construction. This is important for the interface between the outer and the inner dPCPs, as it provides a convenient representation of the output of the outer dPCP as an arithmetic circuit over $\mathbb{F}$, which is then the input for the inner dPCP.

As in the construction of [DFK+11], we take $|\mathbb{F}| \approx 2^{(\log n)^{1-\varepsilon}}$ and begin by constructing PCPs over a fairly large alphabet size which we gradually reduce via composition. The initial alphabet size is $2^{2^{(\log n)^{1-\varepsilon}}}$, and then it drops to $2^{2^{(\log n)^{1-\mathbf{2}\varepsilon}}}$ and then to $2^{2^{(\log n)^{1-\mathbf{3}\varepsilon}}}$ and so on. After $1/\varepsilon$

[3] Looking at the construction as it is presented here, one may ask, why wasn't this done back in 1999, when the conference version of [DFK+11] was published. This construction is notoriously far from modular. Thus, tweaking parameters, following them throughout the construction, and making the necessary changes, would have been a daunting task. Without the modular approach it was not at all clear what the bottlenecks were, let alone address them.

steps we make a couple of final composition steps and end up with the desired alphabet size of $2^{(\log n)^{1-\varepsilon}}$, logarithmic in the initial alphabet size.

Unlike the construction in [DFK+11], we can afford to plug in a sub-constant value for ε, and we take $\varepsilon = c \log\log\log n / \log\log n$ for some constant c so that $2^{(\log n)^{1-\varepsilon}} = 2^{\log n/(\log\log n)^c} = n^{1/(\log\log n)^c}$.

The number of composition steps is $O(1/\varepsilon)$, resulting in a PCP with $O(\log\log n)$ queries and soundness error $n^{1/(\log\log n)^c}$ for some constant $c < 3$ (see Theorem 5.1). Finally, $(\log\log n)^c$ steps of (randomness-efficient) sequential repetition yield a PCP with polynomially small error and poly $\log\log n$ queries as stated in Main Theorem 1.7. It can be shown that the parameters obtained in Theorem 5.1 (namely, soundness error $n^{1/(\log\log n)^{O(1)}}$) is tight given the basic Reed-Muller and Hadamard based building blocks (see § 5.3 for details).

Further Background and Motivation

Every PCP theorem can be viewed as a statement about the local-vs.-global behaviour of proofs, in that the correctness of a proof, which is clearly a *global* property, can be checked by *local* checks, on average. The parameters of the PCP (number of queries, soundness error, alphabet size) give a quantitative measure to this local to global behavior. The sliding scale conjecture essentially says that even with a *constant* number of queries, this local to global phenomenon continues to hold, for all ranges of the soundness error.

Another motivation for minimizing the number of queries becomes apparent when considering interaction with provers instead of direct access to proofs (i.e. MIP instead of PCP). A PCP protocol can not in general be simulated by a protocol between a verifier and a prover because the prover might cheat by adaptively changing her answers. This can be sidestepped by sending each query to a different prover, such that the provers are not allowed to communicate with each other. This is the MIP model of Ben-Or *et al.* [BGKW88]. It is only natural to seek protocols using the smallest number of (non-communicating) provers.

The importance of the sliding scale conjecture stems, in addition to the fundamental nature of the question, from its applications to hardness of approximation. First, it is known that every PCP theorem can be phrased as a hardness-of-approximation for MAX-CSP: the problem of finding an assignment that satisfies a maximal number of constraints in a given constraint system. The soundness error translates to the approximation factor, the alphabet of the proof is the alphabet of the variables, and the number of queries becomes the arity of the constraints in the CSP.

The main goal of this paper can be phrased as proving polynomial hardness of approximation factors for CSPs with smallest possible arity (and over an appropriately small alphabet). Our main theorem translates to the following result

Theorem 1.7. *It is NP-hard to decide if a given CSP with n variables and poly(n) constraints, is perfectly satisfiable or whether every assignment satisfies at most $1/$poly(n) fraction of the constraints. The CSP is such that each constraint has arity at most poly $\log\log n$ and the variables take values over an alphabet of size at most $n^{1/\log\log n^{O(1)}}$.*

In addition to the syntactic connection to MAX-CSP, it is also known that a proof of the sliding scale conjecture would immediately imply polynomial factors inapproximability for DIRECTED-SPARSEST-CUT and DIRECTED-MULTI-CUT [CK09].

The results of [DFK+11] were used in [DS04] for proving hardness of approximation for a certain ℓ_p variant of label cover. However, that work mis-quoted the main result from [DFK+11] as holding true even for a super-constant number of queries, up to $\sqrt{\log\log n}$. In this work, we fill the gap proving the required PCP statement. We thank Michael Elkin for pointing this out.

Further Discussion

A possible alternate route to small soundness PCPs is via the combination of the basic PCP theorem [AS98, ALM+98] with the parallel repetition theorem [Raz98]. Applying k-fold parallel repetition yields a two-query PCP verifier over alphabet of size $|\Sigma| = 2^{O(k)}$, that uses $O(k \log n)$ random bits, and has soundness error $\delta = 2^{-\Omega(k)}$.

If we restrict to polynomial-size constructions, then parallel repetition is of no help compared to Theorem 1.2. If we allow k to be super constant, then more can be obtained. First, it is important to realize that the soundness error should be measured in terms of the output size, namely $N = 2^{k \log n}$. For $k = (\log n)^c$ a simple calculation shows $\log n = (\log N)^{1/(c+1)}$, and hence the soundness error is $\delta(N) = 2^{-(\log n)^c} = 2^{-(\log N)^{1-1/(c+1)}}$. This is no better than the result of [DFK+11] in terms of the soundness error, and in fact, worse in terms of the instance size blow up ($N = 2^{(\log n)^{c+1}}$ as opposed to $N = n^{O(1)}$). Even parameters similar to our main theorem can be obtained, albeit with an almost exponential blowup. Consider $k = \sqrt{n}$ for example. In this case $\log n = 2 \log\log N - \Theta(\log\log\log N)$, and so $\delta(N) = 2^{-\sqrt{n}} = N^{-1/\Theta(\log\log N)}$. From here, to get a polynomially small error one can take $O(\log\log N)$ rounds of (randomness efficient) sequential repetition, coming up with a result that is similar to our Main Theorem 1.7 but with a huge blow up $(N = n^{\sqrt{n}}$ as opposed to $N = n^{O(1)})$.

We remark that a natural approach towards the sliding scale conjecture is to try and find a *randomness-efficient* version of parallel repetition to match the parameters of Theorem 1.2 but with $q = O(1)$. Unfortunately, this approach has serious limitations [FK95] and has so-far been less successful than the algebra-and-composition route, see also [DM11, Mos14].

Organization

We begin with some preliminaries in § 2. We introduce and define dPCPs and distributional soundness in § 3. Our dPCPs have k provers which are analogous to (and stronger than) PCPs that make k queries. In § 4, we state and prove a modular composition theorem for two (algebraic) dPCPs. In § 5, we prove the main theorem, relying on specific "classical" constructions of PCPs that are given in § 6, one based on the Reed-Muller code and low degree test, and one based on the quadratic version of the Hadamard code. These PCPs are the same as in earlier constructs [RS97, AS03, DFK+11, MR10, DH13] except that here we prove that they have the stronger notion of small distributional soundness.

2 Preliminaries

2.1 Notation

All *circuits* in this paper have fan-in 2 and fan-out 2, and we allow only unary NOT and binary AND Boolean operations as internal gates. The *size* of a Boolean circuit/predicate Φ is the number of gates in Φ. Given a circuit/predicate $\Phi : \{0,1\}^n \to \{0,1\}$, we denote by $\mathrm{SAT}(\Phi)$ the set of satisfying assignments for Φ, i.e.,

$$\mathrm{SAT}(\Phi) = \{x \in \{0,1\}^n \mid \ \Phi(x) = 1\}\,.$$

We will refer to the following NP-complete language associated with circuits:

$$\textsc{CktSAT} = \{\Phi \mid \ \Phi \text{ is specified as a Boolean circuit and } \mathrm{SAT}(\Phi) \neq \emptyset\}\,.$$

We will follow the following convention regarding input lengths: n will refer to the length of the input to the circuit Φ (i.e., $\Phi : \{0,1\}^n \to \{0,1\}$) while N will refer to the size of the circuit/predicate (i.e., size(Φ) $= N$). Thus, N is the input size to the problem $\textsc{CktSAT}$.

We will also refer to a similar language associated with arithmetic circuits. First, for some notation. Given a finite field $\mathbb{F}$, we consider arithmetic circuits over $\mathbb{F}$ with addition ($+$) and multiplication ($\times$) gates and constants from the field $\mathbb{F}$. For a function $\Phi : \mathbb{F}^n \to \mathbb{F}$, the size of Φ is the number of gates in the arithmetic circuit specifying Φ. We denote by $\mathrm{SAT}(\Phi)$ the set of all x such that $\Phi(x) = 0$.

Definition 2.1 (Algebraic Circuit SAT). *Given a field $\mathbb{F}$, the* Algebraic-Circuit-Satisfiability *problem, denoted by $\textsc{Alg-CktSAT}_{\mathbb{F}}$, is defined as follows:*

$$\textsc{Alg-CktSAT}_{\mathbb{F}} = \{\Phi \mid \ \Phi \textit{ is specified by an arithmetic circuit over } \mathbb{F} \textit{ and } \mathrm{SAT}(\Phi) \neq \emptyset\}\,.$$

As in the case of $\textsc{CktSAT}$, n refers to the length of the input to the function Φ (i.e., $\Phi : \mathbb{F}^n \to \mathbb{F}$), while N refers to the size of the arithmetic circuit Φ.

2.2 Error Correcting Codes

Let $E : \mathbb{F}^n \to \mathbb{F}^N$ be an error correcting code with relative distance $1 - \mu$, i.e., for every $x \neq x'$, $\Pr_{j \in [N]}[E(x)_j = E(x')_j] \leq \mu$. For a word $w \in \mathbb{F}^N$ that is not necessarily a correct codeword, we can consider the list of all "admissible" codewords, i.e. codewords that have a non-negligible correlation with w. We are interested in more than just a list: we want to associate with each index $j \in [N]$ an element in that list in a unique way. This will allow us to treat w as a random variable W: for a random index j, the random variable $W(j)$ will output the list-element associated with the jth index.

Definition 2.2. *Let $\tau > 0$ be a parameter an let $w : [N] \to \mathbb{F}$. We define the* τ-local decoding function *of w with respect to the code E, $W : [N] \to \mathbb{F}^n \cup \{\perp\}$, as follows:*

- *The τ-admissible words for w are*

$$agr_\tau(w) = \left\{ x \in \mathbb{F}^n \,\middle|\, \Pr_{j \in [N]}[E(x)_j = w_j] \geq \tau \right\}.$$

- *For any $j \in [N]$, if there is a unique word $x \in \mathrm{agr}_\tau(w)$ such that $E(x)_j = w_j$ we set $W(j) = x$. Otherwise, we set $W(j) = \perp$.*

Claim 2.3. *Let $E : \mathbb{F}^n \to \mathbb{F}^N$ be an error correcting code with relative distance $1 - \mu$, let $w : [N] \to F$, and let $W : [N] \to \mathbb{F}^n$ be its τ-local decoding function with respect to E. Also, suppose that $v : [N] \to \mathbb{F}$ is a legal codeword, i.e., $v = E(y)$ for some $y \in \mathbb{F}^n$. Then*

$$\Pr_{j\in[N]} [v_j = w_j \textit{ and } W(j) \neq y] \leq \tau + 4\mu/\tau^2.$$

Proof. Without loss of generality, we may assume that $\tau \geq 2\sqrt{\mu}$. Let us write $\mathrm{agr}_\tau(w) = \{x_1, x_2, \ldots\}$ and let $S_i = \{j \in [N] \mid w_j = (E(x_i))_j\}$. We say that $j \in [N]$ is an ambiguous point for w if $j \in S_i \cap S_{i'}$ for some distinct i, i'. We first bound the fraction of ambiguous points for w.

By inclusion-exclusion, for any $\ell \leq |\mathrm{agr}_\tau(w)|$

$$N \geq \left|\cup_{i=1}^{\ell} S_i\right| \geq \sum_{i=1}^{\ell} |S_i| - \sum_{i\neq i'\leq\ell} |S_i \cap S_{i'}|) \geq \ell\tau N - \binom{\ell}{2}\mu N,$$

where we have used that by definition $|S_i| \geq \tau N$ and by the distance of the code $|S_i \cap S_{i'}| \leq \mu N$. This implies that for every $\ell \leq |\mathrm{agr}_\tau(w)|$,

$$1 \geq \ell\tau - \binom{\ell}{2}\mu \geq \ell\tau - \ell^2\tau^2/8$$

which clearly fails if $\ell = 2/\tau$, so $|\mathrm{agr}_\tau(w)| \leq 2/\tau$ and so

$$\Pr_{j\in[N]}[j \text{ is ambiguous}] \leq \frac{1}{N}\sum_{i\neq i'} |S_i \cap S_{i'}| \leq 4\mu/\tau^2 \tag{2.1}$$

Now the event $[v_j = w_j \text{ and } W(j) \neq y]$ can occur either if j is an ambiguous point for w, or if v is not τ-admissible with respect to w. But the former happens with probability at most $4\mu/\tau^2$ by (2.1), and the latter happens with probability at most τ, as otherwise v would have been τ-admissible. □

Remark 2.4. We minimize the quantity $\tau + 4\mu/\tau^2$, by setting $\tau = (4\mu)^{1/3}$. We refer to this minimum as the agreement parameter η of the code E. Thus, $\eta = 2\tau = 2(4\mu)^{1/3}$.

3 PCPs with distributional soundness

3.1 Standard PCPs

We begin by recalling the definition of a standard k-prover projection PCP verifier.

Definition 3.1 (PCP verifier)**.**

- *A* k-prover projection PCP verifier *over alphabet $\mathbb{F}$ is a probabilistic-time algorithm V that on input Φ, a circuit of size N and a random input R of $r(N)$ random bits generates a tuple (q, φ, g) where $q = (u, v_1 \ldots, v_{k-1})$ is a vector of k queries, $\varphi : \mathbb{F}^m \to \{0, 1\}$ is a predicate, and $g = (g_1, \ldots, g_{k-1})$ is a list of $k - 1$ functions $g_i : \mathbb{F}^m \to \mathbb{F}$ such that the size of the tuple (φ, g) is at most $s(N)$.*

- *We write $(q, \varphi, g) = V(\Phi; R)$ to denote the query-predicate-function tuple output by the verifier V on input Φ and random input R.*

- *It is good to keep in mind the $k = 2$ case as it captures all of the difficulty. In this case the output of V is a* label cover *instance, when enumerating over all of V's random inputs. (The query pairs specify edges (u, v) and (φ, g) specify which pairs of labels are acceptable).*

- *We think of V as a probabilistic oracle machine that on input $(\Phi; R)$ queries k provers $\Pi = (A, B_1, \ldots, B_{k-1})$ at positions $q = (u, v_1, \ldots, v_{k-1})$ respectively to receive the answers $\Pi|_q := (A(u), B_1(v_1), \ldots, B_{k-1}(v_{k-1})) \in \mathbb{F}^m \times \mathbb{F}^{k-1}$, and accepts iff the following checks pass: $\varphi(A(u)) = 1$ and $g_i(A(u)) = B_i(v_i)$ for all $i \in \{1, \ldots, k-1\}$.*

- *Given k provers $\Pi = (A, B_1, \ldots, B_{k-1})$, we will sometimes collectively refer to them as the "proof Π" . Furthermore, we refer to A as the large prover and the B_i's as the projection provers. We call $\Pi|_q$ the local view of the proof Π on queries q and denote by $V^{\Pi}(\Phi; R)$ the output of the verifier V on input $(\Phi; R)$ when interacting with the k provers Π. Thus, $V^{\Pi}(\Phi; R) =$ ACC if the checks pass and is REJ otherwise.*

- *We call N the* input size*, k the* number of provers*, $r(N)$ the* randomness complexity*, and $s(N)$ the* answer size *of the verifier V.*

Definition 3.2 (standard PCPs)**.** *For a function $\delta : \mathbb{Z}^+ \to [0, 1]$, a k-prover projection PCP verifier V is a* k-prover probabilistically checkable proof system *for* CKTSAT *with soundness error δ if the following completeness and soundness properties hold for every circuit Φ:*

Completeness: *If $x \in \mathrm{SAT}(\Phi)$, then there exist k provers $\Pi = (A, B_1, \ldots, B_{k-1})$ that cause the verifier V to accept with probability 1. Formally,*

$$\exists \Pi = (A, B_1, \ldots, B_{k-1}), \qquad \Pr_R\left[V^{\Pi}(\Phi; R) = \text{ACC}\right] = 1.$$

In this case, we say that Π is a valid proof *for the statement $x \in \mathrm{SAT}(\Phi)$.*

Soundness: *If $\Phi \notin$ CKTSAT (i.e, $\mathrm{SAT}(\Phi) = \emptyset$), then for every k provers $\Pi = (A, B_1, \ldots, B_{k-1})$, the verifier V accepts Φ with probability at most $\delta(N)$. Formally,*

$$\forall \Pi = (A, B_1, \ldots, B_{k-1}), \qquad \Pr_R\left[V^{\Pi}(\Phi; R) = \text{ACC}\right] \leq \delta(N).$$

We then say that CKTSAT *has a k-prover projective PCP with soundness error δ.*

3.2 Distributional Soundness

We now present *distributional soundness*, a strengthening of the standard PCP soundness condition that we find to be very natural. Informally, distributional soundness means that the event of the verifier accepting is roughly the same as the event of the local view of the verifier being consistent with a globally consistent proof, up to "the soundness error". I.e.,

$$\Pr[\text{ accept }] = \Pr[\text{ accept } \quad \text{and} \quad \text{the local view agrees with a correct proof }] \pm \delta.$$

Thus, the local acceptance of the verifier is "fully explained" in terms of global consistency.

A little more formally, every purported proof Π (valid or not) can be coupled with an "idealized" distribution $\widetilde{\Pi}(R)$ over valid proofs and $\perp$ such that the behavior of the verifier on random string R when interacting with the proof Π is identical to the corresponding behavior when interacting with the "idealized" proof $\widetilde{\Pi}(R)$ upto an error of δ, which we call the distributional soundness error. Formally,

Definition 3.3 (Distributional Soundness for k-prover PCPs)**.** *For a function $\delta : \mathbb{Z}^+ \to [0,1]$, a k-prover projection PCP verifier V for* CKTSAT *is said to have* distributional soundness error *δ if for every circuit Φ and any set of provers $\Pi = (A, B_1, \ldots, B_{k-1})$ there is an 'idealized pair' of functions $\widetilde{x}(R)$ and $\widetilde{\Pi}(R)$ defined for every random string R such that the following holds.*

- *For every random string R, $\widetilde{\Pi}(R)$ is either a valid proof for the statement $\widetilde{x}(R) \in SAT(\Phi)$, or $\widetilde{\Pi}(R) = \perp$.*
- *With probability at least $1-\delta$ over the choice of the random string R, the local view of the provers Π completely agrees with the local view of the provers $\widetilde{\Pi}(R)$ or is a rejecting local view. In other words,*

  $$\Pr_R\left[V^{\Pi}(\Phi; R) = \perp \text{ or } \Pi|_q = \widetilde{\Pi}(R)|_q\right] \geq 1-\delta,$$

  where q is the query vector generated by the PCP verifier V on input $(\Phi; R)$.

  When the local views agree, i.e. $\Pi|_q = \widetilde{\Pi}(R)|_q$, we say that $\widetilde{\Pi}$ is successful *(in explaining the success of Π).*

The advantage of distributional soundness is that it explains the acceptance probability of every proof Π, valid or otherwise, in the following sense. Suppose a proof Π is accepted with probability p. I.e., p fraction of the local views $\Pi|_q$ are "accepting". Then, it must be the case that but for an error probability of δ, each of these accepting views are projections of (possibly different) valid proofs. It is an easy consequence of this, that distributional soundness implies (standard) soundness.

Proposition 3.4. *If CSAT has a k-prover PCP with distributional soundness error δ, then* CKTSAT *has a k-prover PCP with (standard) soundness error δ. Furthermore, all other parameters (randomness, answer size, alphabet, perfect completeness) are identical.*

Proof. Suppose there exists a circuit Φ and a proof Π such that $\Pr_R\left[V^{\Pi}(\Phi; R) = \text{ACC}\right] > \delta$. Then, by the distributional soundness property it follows that there exists at least one local accepting view which is a projection of a valid proof. In particular, there exists a valid proof which implies $\Phi \in$ CKTSAT. □

3.3 PCP decoders

We now present a variant of PCP verifiers, called PCP decoders, introduced by Dinur and Harsha [DH13]. PCP decoders, as the name suggests, have the additional property that they not only *locally check* the PCP proof Π, but can also *locally decode* symbols of an encoding of the original NP witness from the PCP proof Π. PCP decoders are implicit in many previous constructions of PCPs with small soundness error and were first explicitly defined under the name of local-readers by Dinur *et al.* [DFK+11], as locally-decode-or-reject-codes (LDRC) by Moshkovitz and Raz [MR10] and as decodable PCPs by Dinur and Harsha [DH13]. As in the case of PCP verifiers, our PCP decoders will be projection PCP decoders.

Definition 3.5 (PCP decoder)**.**

- *A k-*prover l-answer projection PCP decoder *over alphabet $\mathbb{F}$ and encoding length t is a probabilistic-time algorithm $\mathcal{D}$ that on input (Φ, F) of size N, a random input string R of $r(N)$ random bits and an additional input index $j \in [t]$, where $\Phi : \mathbb{F}^n \to \{0,1\}$ is a predicate, and $F = (F_1, \ldots, F_{l-1})$ a list of $l-1$ functions $F_i : \mathbb{F}^n \to \mathbb{F}$, generates a tuple (q, φ, g, f) where $q = (u, v_1 \ldots, v_{k-1})$ is a vector of k queries, $\varphi : \mathbb{F}^m \to \{0,1\}$ is a predicate, $g = (g_1, \ldots, g_{k-1})$ is a list of $k-1$ functions $g_i : \mathbb{F}^m \to \mathbb{F}$ and $f = (f_0, \ldots, f_{l-1})$ is a list of l functions $f_i : \mathbb{F}^m \to \mathbb{F}$ such that the size of the tuple (φ, g, f) is at most $s(N)$.*

- *We write $(q, \varphi, g, f) = \mathcal{D}(\Phi, F; R, j)$ to denote the query-predicate-functions tuple output by the decoder $\mathcal{D}$ on input pair (Φ, F), random input R and input index j.*

- *We think of $\mathcal{D}$ as a probabilistic oracle machine that on input $(\Phi, F; R, j)$ queries k provers $\Pi = (A, B_1, \ldots, B_{k-1})$ at positions $q = (u, v_1, \ldots, v_{k-1})$ respectively to receive the answers $\Pi|_q := (A(u), B_1(v_1), \ldots, B_{k-1}(v_{k-1})) \in \mathbb{F}^m \times \mathbb{F}^{k-1}$, then checks if $\varphi(A(u)) = 1$ and $g_i(A(u)) = B_i(v_i)$ for all $i \in \{1, \ldots, k-1\}$ and if these tests pass outputs the l-tuple $(f_0(A(u)), f_1(A(u)), \ldots, f_{l-1}(A(u))) \in \mathbb{F}^{l+1}$ and otherwise outputs $\perp$.*

- *Given k provers $\Pi = (A, B_1, \ldots, B_{k-1})$, we will sometimes collectively refer to them as the "proof Π". Furthermore, we refer to A as the large prover and the B_i's as the projection provers. We call $\Pi|_q$ the local view of the provers Π on queries q and denote by $\mathcal{D}^{\Pi}(\Phi, F; R, j)$ the output of the decoder $\mathcal{D}$ on input $(\Phi, F; R, j)$ when interacting with the k provers Π. Note that the output is an element of $\mathbb{F}^{l+1} \cup \{\perp\}$.*

- *We call N the* input size, *k the* number of provers, *l the* number of answers, *$r(N)$ the* randomness complexity, *and $s(N)$ the* answer size *of the decoder $\mathcal{D}$.*

We now equip the above defined PCP decoders with the new notion of soundness, distributional soundness. We find it convenient (and sufficient) to define decodable PCPs only for predicates and function tuples which have an algebraic structure over the underlying alphabet, which is the field $\mathbb{F}$. In other words, both the input tuple (Φ, F) and output tuple (φ, g, f) have the property that the predicates Φ, φ and the functions F, f, g are specified as arithmetic circuits over $\mathbb{F}$. For the above reasons, we define dPCPs for ALG-CKTSAT (see Definition 2.1).

Definition 3.6 (decodable PCPs with distributional soundness)**.** *For $\delta \in (0,1)$ and a code $E : \mathbb{F}^n \to \mathbb{F}^t$, a k-prover l-answer projection PCP decoder $\mathcal{D}$ is a k-*prover l-answer decodable probabilistically checkable proof system *for ALG-CKTSAT$_{\mathbb{F}}$ with respect to encoding E with distributional soundness error δ if the following properties hold for every input pair (Φ, F):*

Perfect Completeness: *For every $x \in \mathrm{SAT}(\Phi)$, there exist k provers $\Pi = (A, B_1, \ldots, B_{k-1})$ such that the PCP decoder $\mathcal{D}$ when interacting with provers Π outputs $(E(x)_j, F_1(x), \ldots, F_{l-1}(x))$ for every random input R and index j. I.e.,*

$$\Pr_{R,j}\left[\mathcal{D}^{\Pi}(\Phi, F; R, j) = (E(x)_j, F_1(x), \ldots, F_{l-1}(x))\right] = 1.$$

In other words, the decoder $\mathcal{D}$ on input $(\Phi, F; R, j)$ outputs the j-th symbol of the encoding $E(x)$ and the tuple F evaluated at x. In this case, we say that Π is a valid proof *for the statement $x \in \mathrm{SAT}(\Phi)$.*

Distributional Soundness: *For any set of provers $\Pi = (A, B_1, \dots, B_{k-1})$ there exists an idealized pair of functions $\widetilde{x}(R)$ and $\widetilde{\Pi}(R)$ defined for every random string R such that the following holds.*

- *For every random string R, $\widetilde{\Pi}(R)$ is either a valid proof for the statement $\widetilde{x}(R) \in SAT(\Phi)$, or $\widetilde{\Pi}(R) = \perp$.*
- *For every $j \in \{1, \dots, t\}$, with probability at least $1 - \delta$ over the choice of the random string R, the local view of the provers Π completely agrees with the local view of the provers $\widetilde{\Pi}(R)$ or is a rejecting local view. In other words,*

  $$\forall j \in [t], \qquad \Pr_R \left[\mathcal{D}^{\Pi}(\Phi, F; R, j) = \perp \text{ or } \Pi|_q = \widetilde{\Pi}(R)|_q \right] \geq 1 - \delta,$$

  where q is the query vector generated by the PCP decoder $\mathcal{D}$ on input $(\Phi, F; R, j)$.

  When the local views agree, i.e. $\Pi|_q = \widetilde{\Pi}(R)|_q$, we say that $\widetilde{\Pi}$ is successful *(in explaining the success of Π). In this case we have that $\mathcal{D}^{\Pi}(\Phi, F; R, j) = (E(\widetilde{x}(R))_j, F(\widetilde{x}(R)))$.*

We then say that $\mathcal{D}$ is a k-prover l-answer PCP decoder for ALG-CKTSAT *with respect to encoding E with perfect completeness and distributional soundness error δ.*

Remark 3.7. The above definition is a *non-uniform* one in the sense that it is defined for a particular choice of input lengths n, N, size of field $\mathbb{F}$ and encoding $E : \mathbb{F}^n \to \mathbb{F}^t$. A *uniform* version of the above definition can be obtained as follows: there exists a polynomial time uniform procedure that on input n, N (both in unary), the field $\mathbb{F}$ (specified by a prime number and an irreducible polynomial) and the encoding E (specified by the generator matrix) outputs the PCP decoder algorithm. We note that our construction satisfies this stronger uniform property.

As in previous works [MR10, DH13], dPCPs imply PCPs with similar parameters

Proposition 3.8. *If* ALG-CKTSAT *has a k-prover dPCP with distributional soundness error δ, then* ALG-CKTSAT *has a k-prover PCP with distributional soundness error δ. Furthermore, all other parameters (randomness, answer size, alphabet, perfect completeness) are identical.*

We conclude this section highlighting the differences/similarities between the above notion of PCP decoders/dPCPs with that of Dinur and Harsha [DH13] besides the obvious difference in the soundness criterion.

Remark 3.9.

- The above definition of PCP decoders is a generalization of the corresponding definition of Dinur and Harsha [DH13] to the multi-prover ($k > 2$) setting. Since our PCP verifiers are multi-prover verifiers and not just 2-prover verifiers, so are our PCP decoders. Thus, in our notation, the PCP decoders of [DH13] are 2-prover 1-answer projection PCP decoders.

- The above defined PCP decoders locally decode symbols of some pre-specified encoding E of the NP-witness. The PCP decoders of Dinur and Harsha [DH13] is a special case of this when the encoding E is the identity encoding. However as we will see in the next section, it will be convenient to work with encodings which have good distance. In particular, the dPCP composition (considered in this paper) requires the encoding of the "inner" PCP decoder to have good distance.

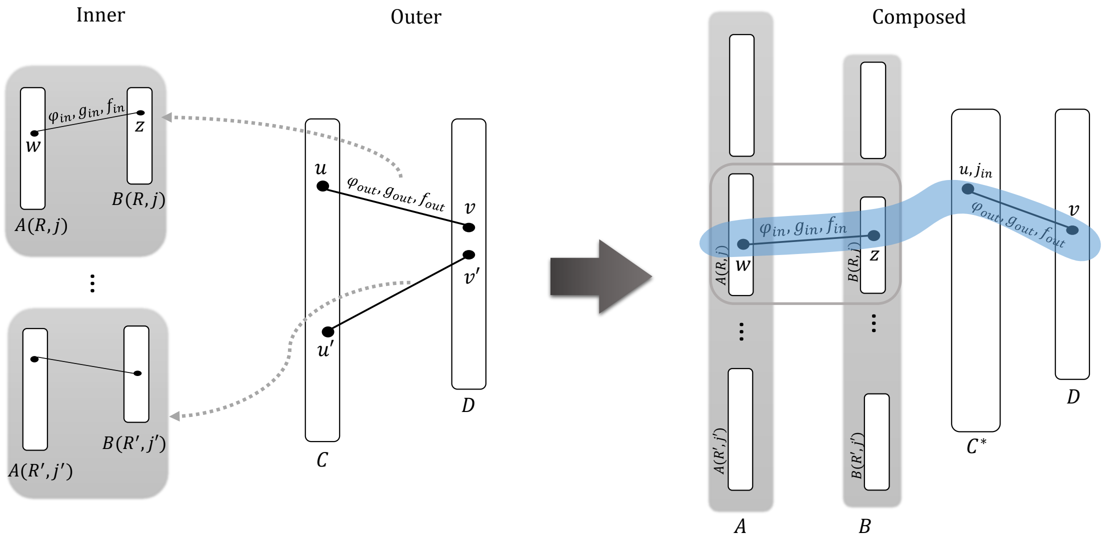

Figure 1: Composition of two 2-prover dPCPs $\mathcal{D}_{\text{out}}$ and $\mathcal{D}_{\text{in}}$ to yield composed dPCP $\mathcal{D}_{\text{comp}}$. Note that $\mathcal{D}_{\text{in}}, \mathcal{D}_{\text{out}}$ make two queries each, and $\mathcal{D}_{\text{comp}}$ makes four queries: $w, z, (u, j_{\text{in}}), v$ to A, B, C^*, D respectively.

4 Composition

In this section, we describe how to compose two PCP decoders. Informally speaking, an "outer" PCP decoder $\mathcal{D}_{\text{out}}$ can be composed with an "inner" PCP decoder $\mathcal{D}_{\text{in}}$ if the answer size of the outer PCP decoder matches the input size of the inner PCP decoder and the number of answers of the inner PCP decoder is the sum of the number of answers of the outer PCP decoder and the number of provers of the outer PCP decoder.

We begin with an informal description of the composition procedure. It might be useful to read this description while looking at Figure 4, in which there are three dPCPs: the inner, the outer, and the composed. We depict each dPCP as a bi-partite (or 4-partite) label-cover-like graph whose vertices correspond to proof locations, and whose (hyper-)edges correspond to local views of the PCP decoder. The main goal of composition is to reduce the answer size of the outer PCP decoder. By this we are referring to the answer size of the large prover; as it is always possible to reduce the answer size of projection provers at negligible cost. For simplicity, let us assume that each of the inner and the outer PCP decoders use only two provers. The inner PCP decoder $\mathcal{D}_{\text{in}}$ interacts with provers A and B, and the outer PCP decoder $\mathcal{D}_{\text{out}}$ interacts with provers C and D. The composed PCP decoder $\mathcal{D}_{\text{comp}}$ works as follows: On input (Φ, F), $\mathcal{D}_{\text{comp}}$ simulates $\mathcal{D}_{\text{out}}$ to obtain the tuple $(q_{\text{out}}, \varphi_{\text{out}}, g_{\text{out}}, f_{\text{out}})$. Letting $q_{\text{out}} = (u, v)$ we picture this as an edge in the bipartite graph of the outer dPCP, and we label this edge with $(\varphi_{\text{out}}, g_{\text{out}}, f_{\text{out}})$. In its normal running $\mathcal{D}_{\text{out}}$ generates queries $q_{\text{out}} = (u, v)$ and queries C on u and D on v. It then checks that $\varphi_{\text{out}}(C(u)) = 0$ and $g_{\text{out}}(C(u)) = D(v)$ and if so it outputs $f_{\text{out}}(C(u))$.

However, the answer $C(u)$ is too large for $\mathcal{D}_{\text{comp}}$, and we would like to use the inner PCP decoder $\mathcal{D}_{\text{in}}$ to replace querying C directly, reducing the answer size at the cost of a few extra queries. For this purpose, the composed PCP decoder $\mathcal{D}_{\text{comp}}$ now simulates the inner PCP decoder $\mathcal{D}_{\text{in}}$ on input $(\varphi_{\text{out}}, (g_{\text{out}}, f_{\text{out}}))$ to generate the tuple $(q_{\text{in}}, \varphi_{\text{in}}, g_{\text{in}}, f_{\text{in}})$. The composed PCP decoder $\mathcal{D}_{\text{comp}}$ then queries the inner provers A, B on queries $q_{\text{in}} = (w, z)$ to obtain the answers $\alpha = A(w)$ and $\beta = B(z)$. It then performs the projection tests g_{in} of the inner PCP decoder $\mathcal{D}_{\text{in}}$ and produces its output $f_{\text{in}}(\alpha)$. These answers are then used to both perform the projection test of the outer PCP decoder as well as produce the required output of the outer PCP decoder.

As usual in composition, we need to enforce consistency between the different invocations of $\mathcal{D}_{\text{in}}$. The input for $\mathcal{D}_{\text{in}}$, namely $(q_{\text{in}}, \varphi_{\text{in}}, g_{\text{in}}, f_{\text{in}})$, is generated using $\mathcal{D}_{\text{out}}$'s randomness, namely R and j. The provers A and B must be told this input because they need to know what they are supposed to prove. Thus A and B are actually aggregates of prover-pairs $A(R, j), B(R, j)$ ranging over all possible R, j. There is a possibility that they could "cheat" by outputting a different answer for the same outer question, depending on R and j. In particular, think of two outer query pairs (u, v_1) and (u, v_2) generated by two different random strings R_1, j_1 and R_2, j_2. We need to ensure that both invocations of $\mathcal{D}_{\text{in}}$ are consistent with the same answer $C(u)$.

We address this issue using the decoding feature of the inner PCP decoder $\mathcal{D}_{\text{in}}$. We replace the outer prover C by a prover C^*, which we call the consistency prover. This prover is supposed to hold an encoding, via E_{in}, of the outer prover C. The composed PCP decoder $\mathcal{D}_{\text{comp}}$ expects the inner PCP decoder $\mathcal{D}_{\text{in}}$ to decode a random symbol in this encoding (i.e., in $E_{\text{in}}(C(u))$). This decoded value is then checked against the consistency prover C^*, which unlike the inner provers is not informed of the outer randomness R. In all, the queries of $\mathcal{D}_{\text{comp}}$ are $w, z, (u, j_{\text{in}}), v$ to A, B, C^*, D respectively.

This additional consistency query helps us get around the above mentioned issue at a small additional cost of η_{in} in the soundness error, where η_{in} is the agreement parameter of the encoding E_{in} (see Remark 2.4).

It can be shown that this consistency query ensures that the distributional soundness error of the composed decoder is at most the sum of the distributional soundness errors of the outer and inner PCP decoders and the agreement parameter of the encoding E_{in}. Previous soundness analyses using list-decoding soundness typically involved a L_{in}-fold multiplicative blowup in the soundness error δ_{out} of the outer PCP decoder (i.e., $\delta_{\text{comp}} \geq L_{\text{in}} \cdot \delta_{\text{out}}$) where L_{in} is the list-size of the inner PCP decoder. Distributional soundness has the advantage of getting rid of this L_{in}-fold blowup at the cost an additional η_{in} additive error.

The above description easily generalizes to $k > 2$ by replacing B by $B_1, \ldots, B_{k_{\text{in}}-1}$ and D by $D_1, \ldots, D_{k_{\text{out}}-1}$.

As in the case of the definition of decodable PCPs, we find it sufficient to describe composition of algebraic dPCPs and not general dPCPs.

Theorem 4.1 (Composition Theorem)**.** *Let $\mathbb{F}$ be a finite field. Suppose that $N_{\text{out}}, N_{\text{in}}, r_{\text{out}}, r_{\text{in}}, s_{\text{out}}, s_{\text{in}}, n_{\text{out}}, n_{\text{in}}, t_{\text{out}}, t_{\text{in}}, k_{\text{out}}, k_{\text{in}}, l_{\text{out}}, l_{\text{in}} \in \mathbb{Z}^+$, and $\delta_{\text{out}}, \delta_{\text{in}}, \eta_{\text{in}} \in [0, 1]$ are such that*

- ALG-CKTSAT *has a k_{out}-prover l_{out}-answer decodable PCP $\mathcal{D}_{\text{out}}$ with respect to encoding $E_{\text{out}} : \mathbb{F}^{n_{\text{out}}} \to \mathbb{F}^{t_{\text{out}}}$ with randomness complexity r_{out}, answer size s_{out}, and distributional soundness error δ_{out} on inputs Φ of size N_{out},*

- ALG-CKTSAT$_{\mathbb{F}}$ *has a k_{in}-prover l_{in}-answer decodable PCP $\mathcal{D}_{\text{in}}$ with respect to encoding E_{in} :*

$\mathbb{F}^{n_{\rm in}} \to \mathbb{F}^{t_{\rm in}}$ with randomness complexity $r_{\rm in}$, answer size $s_{\rm in}$, and distributional soundness soundness error $\delta_{\rm in}$ on inputs φ of size $N_{\rm in}$

- $s_{\rm out} \leq n_{\rm in} \leq N_{\rm in}$,
- $l_{\rm in} = k_{\rm out} + l_{\rm out}$,
- *the inner encoding $E_{\rm in}$ has agreement parameter $\eta_{\rm in}$*

Then, ALG-CKTSAT *has a $k_{\rm out} + k_{\rm in}$-prover $l_{\rm out}$-answer dPCP, denoted $\mathcal{D}_{\rm comp} = \mathcal{D}_{\rm out} \circledast \mathcal{D}_{\rm in}$, with respect to encoding $E_{\rm out}$ on inputs Φ of size N with*

- *randomness complexity $r_{\rm out} + r_{\rm in} + \log_2(t_{\rm in})$,*
- *answer size $s_{\rm in}$, and*
- *distributional soundness error $\delta_{\rm out} + \delta_{\rm in} + \eta_{\rm in}$.*

Furthermore, there exists a universal algorithm with black-box access to $\mathcal{D}_{\rm out}$ and $\mathcal{D}_{\rm in}$ that can perform the actions of $\mathcal{D}_{\rm comp}$ (i.e. evaluating $\mathcal{D}_{\rm comp}(\Phi, F; R, j)$). On inputs of size N, this algorithm runs in time N^c for a universal constant c, with one call to $\mathcal{D}_{\rm out}$ on an input of size N and one call to $\mathcal{D}_{\rm in}$ on an input of size $s_{\rm out}$.

Proof. We will follow the following notation to describe the composed decoder.

Provers of $\mathcal{D}_{\rm comp}$ Suppose the inner PCP decoder $\mathcal{D}_{\rm in}$ interacts with provers $A, B_1, \ldots, B_{k_{\rm in}-1}$ (here A is the large prover and B_i's are the projection provers), and the outer PCP decoder interacts with provers $C, D_1, \ldots, D_{k_{\rm out}-1}$ (here C is the large prover and D_i's are the projection provers).

As mentioned in the informal description, the composed PCP decoder $\mathcal{D}_{\rm comp}$ simulates $\mathcal{D}_{\rm out}$ except that instead of querying C, uses the inner PCP decoder $\mathcal{D}_{\rm in}$ and an additional consistency prover C^*. Thus, the provers for the composed PCP decoder $\mathcal{D}_{\rm comp}$ will be the following: $A, B_1, \ldots, B_{k_{\rm in}-1}, C^*, D_1, \ldots, D_{k_{\rm out}-1}$; the main prover being A and the projection provers being the rest. As mentioned in the outline, for each choice of the outer randomness $R_{\rm out}$ and index $j_{\rm out}$ the inner PCP decoder $\mathcal{D}_{\rm in}$ is simulated on a different input. Hence the corresponding inner provers for the composed dPCP $\mathcal{D}_{\rm comp}$ (i.e., $A, B_1, \ldots, B_{k_{\rm in}-1}$) are explicitly given the specification of the outer randomness $R_{\rm out}$ and index $j_{\rm out}$ as part of their queries. (Alternatively, one can think of A and B_i as an aggregate of separate provers $A(R_{\rm out}, j_{\rm out})$ and $B_i(R_{\rm out}, j_{\rm out})$ per $R_{\rm out}, j_{\rm out}$).

Randomness of $\mathcal{D}_{\rm comp}$ The randomness of $\mathcal{D}_{\rm comp}$ comes in three parts: the randomness $R_{\rm out}$ of $\mathcal{D}_{\rm out}$, the randomness $R_{\rm in}$ of $\mathcal{D}_{\rm in}$ and a random index $j_{\rm in}$ to perform the consistency test. Thus, $R_{\rm comp} = (R_{\rm out}, R_{\rm in}, j_{\rm in})$.

Decoded Index of $\mathcal{D}_{\rm comp}$ The index $j_{\rm comp}$ being decoded by $\mathcal{D}_{\rm comp}$ is passed as the index $j_{\rm out}$ being decoded by $\mathcal{D}_{\rm out}$.

Indexing the answers of $\mathcal{D}_{\text{comp}}$ Note that the number of answers l_{in} of the inner PCP decoder $\mathcal{D}_{\text{in}}$ is the sum of the number of answers l_{out} of the outer $\mathcal{D}_{\text{out}}$ and the number of provers k_{out} of outer $\mathcal{D}_{\text{out}}$. Thus, f_{in} is a list of $l_{\text{out}} + k_{\text{out}}$ functions. We will find it convenient to index the functions in f_{in} with $\{0\} \cup (\{\text{out}\} \times \{0, 1, \ldots, l_{\text{out}} - 1\}) \cup (\{\text{proj}\} \times \{1, \ldots, k_{\text{out}} - 1\})$, such that $f_{\text{in},(\text{proj},i)}$, $i = 1, \ldots, k_{\text{out}} - 1$, are the answers to be compared with the outer projection provers, $f_{\text{in},0}$ is intended for the consistency test, and $f_{\text{in},(\text{out},i)}$, $i = 0, \ldots, \ell_{\text{out}} - 1$, give the answers for the outer decoder.

With these conventions in place, here is the description of the composed PCP decoder, $\mathcal{D}_{\text{comp}}$:

$\mathcal{D}_{\text{comp}}(\Phi, F; R_{\text{out}}, R_{\text{in}}, j_{\text{in}})$:

- Input: (Φ, F)
- Random input string: $(R_{\text{out}}, R_{\text{in}}, j_{\text{in}})$
- Index to be decoded: j_{out}
- Provers: $\Pi = (A, B_1, \ldots, B_{k_{\text{in}}-1}, C^*, D_1, \ldots, D_{k_{\text{out}}-1})$

1. Initial Computation:
   (a) [Simulating $\mathcal{D}_{\text{out}}$] Run $\mathcal{D}_{\text{out}}\,(\Phi, F; R_{\text{out}}, j_{\text{out}})$ to obtain $(q_{\text{out}}, \varphi_{\text{out}}, g_{\text{out}}, f_{\text{out}})$.
   (b) [Simulating $\mathcal{D}_{\text{in}}$] Run $\mathcal{D}_{\text{in}}\,(\varphi_{\text{out}}, (g_{\text{out}}, f_{\text{out}})\,; R_{\text{in}}, j_{\text{in}})$ to obtain $(q_{\text{in}}, \varphi_{\text{in}}, g_{\text{in}}, f_{\text{in}})$.
2. Queries: Let $q_{\text{out}} = (u, v_1, \ldots, v_{k_{\text{out}}-1})$ and let $q_{\text{in}} = (w, z_1, \ldots, z_{k_{\text{in}}-1})$.
   (a) Send query $(R_{\text{out}}, j_{\text{out}}, w)$ to prover A to obtain answer $\alpha = A(R_{\text{out}}, j_{\text{out}}, w)$.
   (b) For $i = 1, \ldots, k_{\text{in}-1}$, send query $(R_{\text{out}}, j_{\text{out}}, z_i)$ to prover B_i to obtain answer $\beta_i = B_i(R_{\text{out}}, j_{\text{out}}, z_i)$.
   (c) Send query (u, j_{in}) to prover C^* to obtain answer $\gamma = C^*(u, j_{\text{in}})$.
   (d) For $i = 1 \ldots, k_{\text{out}} - 1$, send query v_i to prover D_i to obtain answer $\zeta_i = D_i(v_i)$.
3. Checks:
   (a) [Inner local predicate] Check that $\varphi_{\text{in}}\,(\alpha) = 1$.
   (b) [Inner projection tests] For $i = 1, \ldots, k_{\text{in}-1}$, check that $g_{\text{in},i}\,(\alpha) = \beta_i$.
   (c) [Consistency test] Check that $f_{\text{in},0}\,(\alpha) = \gamma$.
   (d) [Outer projection tests] For $i = 1, \ldots, k_{\text{out}-1}$, check that $f_{\text{in},(\text{proj},i)}\,(\alpha) = \zeta_i$.
4. Output: If all the checks in the above step pass, then return $f_{\text{in},(\text{out},\cdot)}\,(\alpha)$ else return $\perp$.

The claims about $\mathcal{D}_{\text{comp}}$'s parameters (randomness complexity, answer size, number of provers, number of answers) except completeness and soundness error can be verified by inspection. Thus, we only need to check completeness and soundness.

Completeness Let $x \in SAT\,(\Phi)$. By the completeness of outer $\mathcal{D}_{\text{out}}$, there exist provers $\Pi^{\text{out}} = (C, D_1, \ldots, D_{k_{\text{out}}-1})$, such that for all $(R_{\text{out}}, j_{\text{out}})$ we have

$$\mathcal{D}_{\text{out}}^{\Pi^{\text{out}}}\,(\Phi, F; R_{\text{out}}, j_{\text{out}}) = \Big(E_{\text{out}}\,(x)_{j_{\text{out}}}\,, F_1\,(x)\,, \ldots, F_{l_{\text{out}}}\,(x)\Big)\,.$$

Fix any particular outer random string R_{out} and index j_{out}. Let $\mathcal{D}_{\text{out}}\,(\Phi, F; R_{\text{out}}, j_{\text{out}}) = (q_{\text{out}}, \varphi_{\text{out}}, g_{\text{out}}, f_{\text{out}})$. Since the outer decoder $\mathcal{D}_{\text{out}}$ does not reject, we must have that $y_{(R_{\text{out}}, j_{\text{out}})} := C(u)$ satisfies φ_{out}. In other words, $y_{(R_{\text{out}},j_{\text{out}})} \in \text{SAT}(\varphi_{\text{out}})$. Now, by the completeness of the inner $\mathcal{D}_{\text{in}}$, we have that for these $(R_{\text{out}}, j_{\text{out}})$ there exist provers $\Pi^{\text{in}}_{(R_{\text{out}},j_{\text{out}})} = \left(A_{(R_{\text{out}},j_{\text{out}})}, B_{(R_{\text{out}},j_{\text{out}}),1}, \dots, B_{(R_{\text{out}},j_{\text{out}}),k_{\text{in}}-1}\right)$ such that for all $(R_{\text{in}}, j_{\text{in}})$ we have

$$\mathcal{D}_{\text{in}}^{\Pi^{\text{in}}_{(R_{\text{out}},j_{\text{out}})}}\left(\varphi_{\text{out}}, (g_{\text{out}}, f_{\text{out}}); R_{\text{in}}, j_{\text{in}}\right) = \left(E_{\text{in}}\left(y_{(R_{\text{out}},j_{\text{out}})}\right)_{j_{\text{in}}}, g_{\text{out}}\left(y_{(R_{\text{out}},j_{\text{out}})}\right), f_{\text{out}}\left(y_{(R_{\text{out}},j_{\text{out}})}\right)\right).$$

We are now ready to define the provers

$$\Pi = \left(A, B_1, \dots, B_{k_{\text{in}}-1}, C^*, D_1, \dots, D_{k_{\text{out}}-1}\right)$$

for the composed decoder $\mathcal{D}_{\text{comp}}$. As the name suggests, the projection provers $D_i, i = 1, \dots, k_{\text{out}} - 1$ are exactly the same as the outer projection provers in Π^{out}. The consistency prover C^* is defined by encoding $C(u)$ separately for each u as

$$C^*(u, j_{\text{in}}) := E_{\text{in}}(C(u))_{j_{\text{in}}}.$$

The projection provers $B_i, i = 1, \dots, k_{\text{in}} - 1$ are defined as $B_i(R_{\text{out}}, j_{\text{out}}, z_i) := B_{(R_{\text{out}},j_{\text{out}}),i}(z_i)$. Finally, the large prover A is defined as $A(R_{\text{out}}, j_{\text{out}}, w) := A_{(R_{\text{out}},j_{\text{out}})}(w)$. It is easy to check that according to this definition of Π, for each $((R_{\text{out}}, R_{\text{in}}, j_{\text{in}}), j_{\text{out}})$ it holds that

$$\mathcal{D}_{\text{comp}}^{\Pi}\left(\Phi, F; (R_{\text{out}}, R_{\text{in}}, j_{\text{in}}), j_{\text{out}}\right) = \left(E_{\text{out}}(x)_{j_{\text{out}}}, F_1(x), \dots, F_{l_{\text{out}}}(x)\right).$$

This proves the completeness of $\mathcal{D}_{\text{comp}}$. □

Distributional Soundness of $\mathcal{D}_{\text{comp}}$ We prove the following statement about the distributional soundness of $\mathcal{D}_{\text{comp}}$.

Lemma 4.2. *Suppose the outer PCP decoder $\mathcal{D}_{\text{out}}$ has distributional soundness error δ_{out} with respect to encoding E_{out}, and the inner PCP decoder $\mathcal{D}_{\text{in}}$ has distributional soundness error δ_{in} with respect to encoding E_{in}, and suppose E_{in} has agreement parameter η_{in} (see Remark 2.4). Then, the composed PCP decoder $\mathcal{D}_{\text{comp}} = \mathcal{D}_{\text{out}} \circledast \mathcal{D}_{\text{in}}$ has distributional soundness error $\delta_{\text{comp}} \leq \delta_{\text{out}} + \delta_{\text{in}} + \eta_{\text{in}}$ with respect to encoding E_{out}.*

Proof. Suppose $\mathcal{D}_{\text{comp}}$ on input (Φ, F) interacts with provers

$$\Pi = \left(A, B_1, \dots, B_{k_{\text{in}}-1}, C^*, D_1, \dots, D_{k_{\text{out}}-1}\right).$$

To prove soundness of $\mathcal{D}_{\text{comp}}$, we need to construct for each composed random string $R_{\text{comp}} := (R_{\text{out}}, R_{\text{in}}, j_{\text{in}})$, functions $\widetilde{x}(R_{\text{comp}})$ and $\widetilde{\Pi}(R_{\text{comp}})$ such that $\widetilde{\Pi}(R_{\text{comp}})$ is either $\perp$ or a valid proof for the statement $\widetilde{x}(R_{\text{comp}}) \in \text{SAT}(\Phi)$, and for every j_{out} we have

$$\Pr_{R_{\text{comp}}}\left[\mathcal{D}_{\text{comp}}^{\Pi}(\Phi, F; R_{\text{comp}}, j_{\text{out}}) = \perp \text{ or } \Pi|_{q_{\text{comp}}} = \widetilde{\Pi}(R_{\text{comp}})|_{q_{\text{comp}}}\right] \geq 1 - (\delta_{\text{out}} + \delta_{\text{in}} + \eta_{\text{in}}),$$

where q_{comp} is the query vector generated by $\mathcal{D}_{\text{comp}}$.

The construction of $\widetilde{x}(R_{\text{comp}})$ and $\widetilde{\Pi}(R_{\text{comp}})$ relies on the soundness properties of $\mathcal{D}_{\text{out}}$ and $\mathcal{D}_{\text{in}}$. We first locally-decode C^* to obtain a distribution over outer provers C. We then use the distributional soundness of $\mathcal{D}_{\text{out}}$ to obtain an idealized outer proof $(\widetilde{C}, \widetilde{D}_i)$. We then use the soundness of $\mathcal{D}_{\text{in}}$ to obtain idealized inner proofs $(\widetilde{A}, \widetilde{B}_i)$.

Outer main prover $C_{j_{\text{in}}}$: For each j_{in}, we define an outer main prover $C_{j_{\text{in}}}$ using the consistency prover C^* as follows. Let $\tau = \eta_{\text{in}}/2$ be the agreement parameter of E_{in}, as in Remark 2.4. For each query $u := q_{\text{out},0}$ to the outer main prover, $C_{j_{\text{in}}}(u)$ is defined to be the j_{in}-th entry of the τ-local decoding of $C^*(u, \cdot)$ as in Definition 2.2 if well-defined and $\perp$ otherwise.

Idealized outer pairs $\widetilde{x}^{\text{out}}_{j_{\text{in}}}(R_{\text{out}})$ and $\widetilde{\Pi}^{\text{out}}_{j_{\text{in}}}(R_{\text{out}})$: For each j_{in} we have an outer proof

$$\Pi^{\text{out}}_{j_{\text{in}}} = \left(C_{j_{\text{in}}}, D_1, \ldots, D_{k_{\text{out}}-1}\right).$$

Note that only the C provers are different in the various outer provers $\Pi^{\text{out}}_{j_{\text{in}}}$ as we range over j_{in}. From the soundness of $\mathcal{D}_{\text{out}}$ for every $\Pi^{\text{out}}_{j_{\text{in}}}$ there is an idealized pair $\widetilde{x}^{\text{out}}_{j_{\text{in}}}(R_{\text{out}})$ and $\widetilde{\Pi}^{\text{out}}_{j_{\text{in}}}(R_{\text{out}}) = \left(\widetilde{C}, \widetilde{D}_1, \ldots, \widetilde{D}_{k_{\text{out}}-1}\right)$[4] that "explain" its success.

Idealized inner pairs $\widetilde{y}^{\text{in}}_{(R,j)}(R_{\text{in}})$ and $\widetilde{\Pi}^{\text{in}}_{(R,j)}(R_{\text{in}})$: For every outer randomness R and index j let

$$\Pi^{\text{in}}_{(R,j)} := \left(A(R,j,\cdot), B_1(R,j,\cdot), \ldots, B_{k_{\text{in}}-1}(R,j,\cdot)\right)$$

be the relevant part of the proof for $\mathcal{D}_{\text{in}}$.

Let $(q, \varphi, g, f) = \mathcal{D}_{\text{out}}(\Phi, F; R, j)$. When $R_{\text{out}} = R$ and $j_{\text{out}} = j$, the composed decoder $\mathcal{D}_{\text{comp}}$ simulates running $\mathcal{D}_{\text{in}}$ with input $(\varphi, (g, f); R_{\text{in}}, j_{\text{in}})$ and with the proof $\Pi^{\text{in}}_{(R,j)}$.

For each $\Pi^{\text{in}}_{(R,j)}$, the soundness of $\mathcal{D}_{\text{in}}$ guarantees idealized prover pairs $\widetilde{y}^{\text{in}}_{(R,j)}$ and $\widetilde{\Pi}^{\text{in}}_{(R,j)}$ (functions of R_{in}) that "explain" its success.

We are ready to define the idealized $(\widetilde{x}, \widetilde{\Pi})$ pairs for the composed decoder $\mathcal{D}_{\text{comp}}$.

Idealized composed pairs $\widetilde{x}(R_{\text{comp}})$ and $\widetilde{\Pi}(R_{\text{comp}})$: Recall that R_{comp} is short for $(R_{\text{out}}, R_{\text{in}}, j_{\text{in}})$. Define $\widetilde{x}(R_{\text{comp}}) := \widetilde{x}^{\text{out}}_{j_{\text{in}}}(R_{\text{out}})$. We then set $\widetilde{\Pi}(R_{\text{comp}})$ as follows: If $\widetilde{\Pi}^{\text{out}}_{j_{\text{in}}}(R_{\text{out}})$ is successful, in particular $C_{j_{\text{in}}}(u) = \widetilde{C}(u)$, set $\widetilde{\Pi}(R_{\text{comp}})$ to be the set of provers $\left(\widetilde{A}, \widetilde{B}_1, \ldots, \widetilde{B}_{k_{\text{in}}-1}, \widetilde{C}^*, \widetilde{D}_1, \ldots, \widetilde{D}_{k_{\text{out}}-1}\right)$ defined next.

- The outer projection provers $\widetilde{D}_i$ are defined to be the same as in $\widetilde{\Pi}^{\text{out}}_{j_{\text{in}}}(R_{\text{out}})$.
- Let $\widetilde{C}$ be the main prover in $\widetilde{\Pi}^{\text{out}}_{j_{\text{in}}}(R_{\text{out}})$. We define $\widetilde{C}^*$ as follows:

$$\widetilde{C}^*(u, j) := E_{\text{in}}\left(\widetilde{C}(u)\right)_j.$$

- For any pair (R, j) where $R \in \{0,1\}^{r_{\text{out}}}$ and $j \in [t_{\text{out}}]$, we define $\widetilde{A}(R,j,\cdot)$ and $\widetilde{B}_i(R,j,\cdot)$ (for $i = 1, \ldots, k_{\text{in}} - 1$) as follows. Denote $(q, \varphi, g, f) = \mathcal{D}_{\text{out}}(\Phi, F; R, j)$ and suppose $q = (u, v_1, \ldots, v_{k_{\text{out}}-1})$. If $\widetilde{y}^{\text{in}}_{(R,j)}(R_{\text{in}}) = \widetilde{C}(u)$ and also $\widetilde{\Pi}^{\text{in}}_{(R,j)}(R_{\text{in}})$ is successful, we set $\widetilde{A}(R,j,\cdot)$ and the provers $\widetilde{B}_i(R,j,\cdot)$ to be the main prover and projection provers in $\widetilde{\Pi}^{\text{in}}_{(R,j)}(R_{\text{in}})$ respectively.

[4] The proofs $\left(\widetilde{C}, \widetilde{D}_1, \ldots, \widetilde{D}_{k_{\text{out}}-1}\right)$ depend on j_{in} and R_{out} so more formally could be written as $\left(\widetilde{C}_{j_{\text{in}}}(R_{\text{out}}), \widetilde{D}_{j_{\text{in}},1}(R_{\text{out}}), \ldots, \widetilde{D}_{j_{\text{in}},k_{\text{out}}-1}(R_{\text{out}})\right)$, but we will drop the indices for ease of readability.

Otherwise, if $\widetilde{y}^{\text{in}}_{(R,j)}(R_{\text{in}}) \neq \widetilde{C}(u)$ or $\widetilde{\Pi}^{\text{in}}_{(R,j)}(R_{\text{in}})$ is not successful, we define $\widetilde{A}(R,j,\cdot)$ and $\widetilde{B}_i(R,j,\cdot)$ by letting them be some valid proofs for the statement $\widetilde{C}(u) \in SAT(\varphi)$ (Note that $\widetilde{C}(u)$ satisfies φ since $\widetilde{\Pi}^{\text{out}}_{j_{\text{in}}}(R_{\text{out}})$ is successful).

If either $\widetilde{\Pi}^{\text{out}}_{j_{\text{in}}}(R_{\text{out}})$ is not successful or any of the intermediate objects in the above definition are $\perp$, then we set $\widetilde{\Pi}(R_{\text{comp}})$ to $\perp$.

It remains to show that the pair $\widetilde{x}(R_{\text{comp}})$ and $\widetilde{\Pi}(R_{\text{comp}})$ has the desired properties. It follows by inspection of the definition of $\widetilde{\Pi}(R_{\text{comp}})$ that whenever it is not $\perp$, it is a valid proof of the statement $\widetilde{x}(R_{\text{comp}}) \in SAT(\Phi)$ and agrees with the local view of Π on input $(\Phi, F; R_{\text{comp}}, j_{\text{out}})$.

So it remains to show that for every j_{out},

$$\Pr_{R_{\text{comp}}}\left[\widetilde{\Pi}(R_{\text{comp}}) = \perp \text{ and } \mathcal{D}^{\Pi}_{\text{comp}}(\Phi, F; R_{\text{comp}}, j_{\text{out}}) \neq \perp\right] \leq \delta_{\text{out}} + \delta_{\text{in}} + \eta_{\text{in}}.$$

We partition the above event intro three parts according to the highest indexed condition among the following three conditions that does not hold — one of them must not hold for $\widetilde{\Pi}(R_{\text{comp}})$ to be equal to $\perp$.

1. $\widetilde{\Pi}^{\text{out}}_{j_{\text{in}}}(R_{\text{out}})$ is successful, in particular $C_{j_{\text{in}}}(u) = \widetilde{C}(u)$.

2. $\widetilde{y}^{\text{in}}_{(R_{\text{out}},j_{\text{out}})}(R_{\text{in}}) = \widetilde{C}_{j_{\text{in}}}(u)$.

3. $\widetilde{\Pi}^{\text{in}}_{(R_{\text{out}},j_{\text{out}})}(R_{\text{in}})$ is successful.

We separately bound the probability of each event in this partition.

- We bound the probability that Condition 3 does not hold, namely that $\widetilde{\Pi}^{\text{in}}_{(R_{\text{out}},j_{\text{out}})}(R_{\text{in}})$ is not successful, and yet $\mathcal{D}_{\text{comp}}$ does not reject. If $\mathcal{D}_{\text{comp}}$ doesn't reject then in particular checks Check 3a and Check 3b pass, which means that $\mathcal{D}_{\text{in}}^{\Pi^{\text{in}}_{(R_{\text{out}},j_{\text{out}})}}(\varphi_{\text{out}}, (g_{\text{out}}, f_{\text{out}}); R_{\text{in}}, j_{\text{in}}) \neq \perp$. But the soundness of $\mathcal{D}_{\text{in}}$ implies that the probability over the choice of R_{in} that this occurs and yet $\widetilde{\Pi}^{\text{in}}_{(R_{\text{out}},j_{\text{out}})}(R_{\text{in}})$ is not successful is bounded by δ_{in}.

- Now we bound the probability that Condition 3 holds, Condition 2 does *not* hold, and yet $\mathcal{D}_{\text{comp}}$ does not reject. When Condition 3 holds, the output of the $\mathcal{D}_{\text{in}}$ simulation for the encoding is $E_{\text{in}}\left(y^{\text{in}}_{(R_{\text{out}},j_{\text{out}})}(R_{\text{in}})\right)_{j_{\text{in}}}$. It is thus enough to bound the probability that $\widetilde{y}^{\text{in}}_{(R_{\text{out}},j_{\text{out}})}(R_{\text{in}}) \neq \widetilde{C}(u)$ and yet $E_{\text{in}}\left(\widetilde{y}^{\text{in}}_{(R_{\text{out}},j_{\text{out}})}(R_{\text{in}})\right)_{j_{\text{in}}} = C^*(u, j_{\text{in}})$, i.e. Check 3c passes. Since by definition $C_{j_{\text{in}}}(u)$ is the τ-local decoding of $C^*(u,\cdot)$ at position j_{in}, Claim 2.3 and Remark 2.4 imply that the probability of this event over the choice of j_{in} is bounded by the agreement parameter η_{in}.

- It remains to bound the probability that Conditions 3 and 2 hold but Condition 1 does not, and yet $\mathcal{D}_{\text{comp}}$ does not reject. When Condition 2 and 3 hold it means that $\widetilde{C}(u) = \widetilde{y}^{\text{in}}_{(R_{\text{out}},j_{\text{out}})}(R_{\text{in}}) \in SAT(\varphi_{\text{out}})$ and the output of the simulated $\mathcal{D}_{\text{in}}$ computed by $\mathcal{D}_{\text{comp}}$ is

$$f_{\text{in}}(A(R_{\text{out}}, j_{\text{out}}, w)) = \left(\left(E_{\text{in}}\left(\widetilde{C}(u)\right)_{j_{\text{in}}}, g_{\text{out}}\left(\widetilde{C}(u)\right), f_{\text{out}}\left(\widetilde{C}(u)\right)\right)\right).$$

If $\mathcal{D}_{\text{comp}}$ does not reject it means that the values of $g_{\text{out}}\left(\widetilde{C}(u)\right)$ match the ones obtained from the outer projection provers, to which it is compared in Check 3d. But these are also the values used by $\mathcal{D}_{\text{out}}$ when it is run on input $(\Phi, F; R_{\text{out}}, j_{\text{out}})$ with the proof $\widetilde{\Pi}^{\text{out}}_{j_{\text{in}}}$, which means that $\mathcal{D}_{\text{out}}^{\widetilde{\Pi}^{\text{out}}_{j_{\text{in}}}}(\Phi, F; R_{\text{out}}, j_{\text{out}}) \neq \perp$. But the probability over the choice of R_{out} that this happens while Condition 1 fails is bounded by δ_{out}, the distributional soundness error of $\mathcal{D}_{\text{out}}$.

This proves the distributional soundness of $\mathcal{D}_{\text{comp}}$.

□

This completes the proof of the Composition Theorem 4.1. □

5 Proof of Main Theorem

Theorem 5.1 (Main Construct). *Every language L in* NP *has a $O(\lg\lg N/\lg\lg\lg N)$-prover projective PCP with the following parameters. On input a Boolean predicate/circuit Φ of size N, the PCP has*

- *randomness complexity $O(\lg N)$,*
- *query complexity $O(\lg\lg N/\lg\lg\lg N)$,*
- *answer size $O(\lg N/poly \lg\lg N)$,*
- *perfect completeness, and*
- *soundness error $N^{1/(\lg\lg N)^{\Omega(1)}}$.*

The PCP with inverse polynomial soundness error stated in Main Theorem 1.7 is obtained by sequentially repeating the above PCP $\operatorname{poly}(\lg\lg N)$ times in a randomness efficient manner.

5.1 Building Blocks

The two building blocks, we need for our construction, are two decodable PCP based on the Reed-Muller code and the Hadamard code respectively. The constructions of both these objects is standard given the requirements of the dPCP. These PCPs are based on two encodings the low-degree encoding LDE and the quadratic Hadamard encoding QH respectively. The definition of these codes is given in the next section (§ 6). For the purpose of this section, it suffices that these are error correcting codes with very good distance.

Theorem 5.2 (Reed-Muller based dPCP). *For any finite field $\mathbb{F}$, and parameter h such that $1 < h < |\mathbb{F}|^{0.1}$ and any $\ell > 0$, there is a 2-prover $\ell+1$-answer decodable PCP $\mathcal{D}$ with respect to the encoding $\mathrm{LDE}_{\mathbb{F},h}$ for the language $\textsc{Alg-CktSAT}_{\mathbb{F}}$ with the following parameters: On inputs (i) a predicate $\Phi : \mathbb{F}^n \to \{0,1\}$ and (ii) functions $F_1, \ldots, F_\ell : \mathbb{F}^n \to \mathbb{F}$ given by arithmetic circuits over $\mathbb{F}$ whose total size is N, the dPCP $\mathcal{D}$ has (let $m = \log N/\log h$),*

- *randomness complexity $O(\log N + m\log|\mathbb{F}|) = O(m\log|\mathbb{F}|)$,*
- *answer size $s, s' = O(m(m+\ell))$,*

- *and distributional soundness error $1/|\mathbb{F}|^{0.1}$.*

Theorem 5.3 (Hadamard based dPCP)**.** *For any finite field $\mathbb{F}$, and any $\ell > 0$, there is a 2-prover $\ell + 1$-answer decodable PCP $\mathcal{D}_{\mathrm{QH},\mathbb{F}}$ with respect to the encoding $\mathrm{QH}_{\mathbb{F}}$ for the language* ALG-CKTSAT$_{\mathbb{F}}$ *with the following parameters: On inputs (i) a predicate $\Phi : \mathbb{F}^n \to \{0, 1\}$ and (ii) functions $F_1, \ldots, F_\ell : \mathbb{F}^n \to \mathbb{F}$ given by arithmetic circuits over $\mathbb{F}$ whose total size is N, the dPCP $\mathcal{D}_{\mathrm{QH},\mathbb{F}}$ has*

- *randomness complexity $O(N^2 \log |\mathbb{F}|)$,*
- *answer size $s, s' = O(\ell)$,*
- *perfect completeness, and*
- *distributional soundness error $\le 1/|\mathbb{F}|^{0.1}$.*

These theorems are proved in Section 6.

5.2 Putting it together (Proof of Theorem 5.1)

By NP-completeness of CKTSAT it suffices to prove Theorem 5.1 for CKTSAT. Let Ψ be an instance of CKTSAT and let N denote its size. Let $\varepsilon = 20 \lg \lg \lg N/9 \lg \lg N$ be a parameter[5]. Note that $(\lg N)^\varepsilon = (\lg \lg N)^{20/9}$. Let $M = 2^{(\lg N)^{1-\varepsilon}} = N^{1/(\lg \lg N)^{20/9}}$. Choose a prime number $p \in (M, 2M)$[6] and let $\mathbb{F} = GF(p)$ be the finite field of size p, which we fix for the rest of the proof. We may assume wlog. that the predicate Ψ has only AND and NOT gates. Given this, we can arithmetize Φ to obtain an arithmetic circuit Φ over $\mathbb{F}$ by replacing AND gates by multiplication gates and NOT gates by $1 - x$ gates. Thus, we can view the N-sized predicate Φ as an N-sized arithmetic circuit over the field $\mathbb{F}$.

We construct a PCP for Ψ with the required parameters by constructing a dPCP for Φ with respect to the encoding $\mathrm{LDE}_{\mathbb{F},h_0}$ for some suitable choice of h_0. This dPCP is in turn constructed by composing a sequence of dPCPs each with smaller and smaller answer size. Each dPCP in the sequence will be obtained by composing the prior dPCP (used as an outer dPCP) with an adequate inner dPCP. The outermost dPCP as well as the inner dPCP in all but the last step of composition will be obtained from Theorem 5.2 by various instantiations of the parameter h. The innermost dPCP used in the final stage of the composition will be the dPCP obtained from Theorem 5.3.

Stage I: Let $n_0 = N$ and $h_0 = |\mathbb{F}|^{0.1} = 2^{0.1(\lg N)^{1-\varepsilon}} = N^{0.1/(\lg \lg N)^{20/9}}$. For this choice of n_0, h_0 and $\mathbb{F}$ and $l_0 = 0$, let $\mathcal{D}^{(0)} := \mathcal{D}_0$ be the dPCP obtained from Theorem 5.2. This will serve as our outermost dPCP. Let us recall the parameters of this dPCP. Observe that for this setting $m_0 = \log_{h_0} n_0 = \lg N/ \lg h_0 = 10(\lg \lg N)^{20/9}$. $\mathcal{D}^{(0)}$ is a 2-prover decodable PCP with respect to the encoding $\mathrm{LDE}_{\mathbb{F},h_0}$ for the language ALG-CKTSAT$_{\mathbb{F}}$ with the following parameters: On inputs Φ of size N over $\mathbb{F}$, $\mathcal{D}^{(0)}$ has randomness complexity $R_0 = c \cdot m_0 \lg |\mathbb{F}| = 10c \lg N$, answer size $s_0 = 2(m_0 h_0)^2 < 2^{0.3(\lg N)^{1-\varepsilon}} = N^{0.3/(\lg \lg N)^{20/9}}$ and distributional soundness error $1/|\mathbb{F}|^{0.1}$.

[5] In the construction, setting $\varepsilon = 20 \lg \lg \lg N/9 \lg \lg N$ will prove the *poly* $\lg \lg N$-query PCP with inverse polynomially soundness error (as stated in the main theorem). It is to be noted that setting ε to a constant in $(0, 1)$ will recover the DFKRS PCP.

[6] Since the procedure is allowed to run in polynomial time in N, it has enough time to examine every number in the range $(M, 2M)$ and check if it is prime or not.

Let $\varepsilon' = \varepsilon/10 = 2 \lg\lg\lg N / 9 \lg\lg N$. Let i^* be the smallest integer such that $1 - \varepsilon - i\varepsilon' < 9\varepsilon/80$. Note that $i^* = O(1/\varepsilon) = O(\lg\lg N / \lg\lg\lg N)$. For $i = 1, \ldots, i^*$, let $\mathcal{D}_i$ be the dPCP obtained by instantiating the dPCP in Theorem 5.2 with parameters $h_i = 2^{(\lg N)^{1-\varepsilon-i\varepsilon'}} = N^{1/(\lg\lg N)^{20/9\cdot(1+i/10)}}$ and $l_i = 2i$. We will run dPCP $\mathcal{D}_i$ on inputs of instance size $n_i = 2^{3(\lg N)^{1-\varepsilon-(i-1)\varepsilon'}} = N^{3/(\lg\lg N)^{20/9\cdot(1+(i-1)/10)}}$. Thus, $m_i = \lg n_i / \lg h_i = 3(\lg N)^{\varepsilon'} = 3(\lg\lg N)^{2/9}$. Hence, $\mathcal{D}_i$ is a $(2i+1)$-answer 2-prover dPCP that on inputs of instance size n_i has randomness complexity $R_i = cm_i \lg|\mathbb{F}| = 3c(\lg N)^{1-\varepsilon+\varepsilon'} = 3c \lg N/(\lg\lg N)^2$, answer size $s_i = 2(m_i h_i)^2 < 2^{3(\lg N)^{1-\varepsilon-i\varepsilon'}} = N^{3/(\lg\lg N)^{20/9\cdot(1+i/10)}}$ and distributional soundness error $\delta_i = 1/|\mathbb{F}|^{0.1}$.

Observe that our setting of parameters satisfy $s_{i-1} \leq n_i$ and $l_{i+1}+1 = 2(i+1)+1 = (l_i+1)+2$. So the answer size of the predicates produced by dPCP $\mathcal{D}_i$ are valid input instances for dPCP $\mathcal{D}_{i+1}$, for $i = 0, \ldots, i^*-1$. Hence, we can compose them with each other. Consider the dPCPs $\mathcal{D}^{(i)}$ defined as follows:

$$\mathcal{D}^{(i)} := \mathcal{D}^{(i-1)} \circledast \mathcal{D}_i, \qquad i = 1, \ldots, i^*.$$

Also note that the code $\mathrm{LDE}_{\mathbb{F},h_i}$ has block length $|\mathbb{F}|^{m_i}$ and distance $(1 - O(m_i h_i)/|\mathbb{F}|) \geq 1 - 1/\sqrt{|\mathbb{F}|}$. Thus, the agreement parameter η_i is at least $1/|\mathbb{F}|^{1/6}$.

Let $\mathcal{D}^{(I)} := \mathcal{D}^{(i^*)}$ be the final dPCP obtained as above. Observe that it is a $2i^* = O(1/\varepsilon) = O(\lg\lg N/\lg\lg\lg N)$-prover dPCP with respect to the encoding $\mathrm{LDE}_{\mathbb{F},h_0}$ for the language $\textsc{Alg-CktSAT}_{\mathbb{F}}$ with the following parameters: On inputs Φ of size N over $\mathbb{F}$, the dPCP $\mathcal{D}^{(I)}$ has randomness complexity $R^{(I)}$, distributional soundness error $\delta^{(I)}$ and answer size $s^{(I)}$ (which are calculated below).

$$\begin{aligned}
R^{(I)} &= R_0 + \sum_{i=1}^{i^*} (R_i + \log(\mathrm{blocklength}(\mathrm{LDE}_{\mathbb{F},h_i}))) \\
&= 10c \lg N + \sum_{i=1}^{i^*} (cm_i \lg|\mathbb{F}| + m_i \lg|\mathbb{F}|) \\
&= 10c \lg N + \sum_{i=1}^{i^*} 3(c+1)\frac{\lg N}{(\lg\lg N)^2} \\
&= 11c \lg N \qquad [\text{ since } i^* \leq \lg\lg N]. \\
s^{(I)} &= s_{i^*} \\
&= 2^{3(\lg N)^{1-\varepsilon-i^*\varepsilon'}} \\
&\leq 2^{3(\lg N)^{9\varepsilon/80}} \\
&= 2^{3(\lg\lg N)^{1/4}}. \\
\delta^{(I)} &= \delta_0 + \sum_{i=1}^{i^*} (\delta_i + \eta_i) \\
&= (i^*+1) \cdot \left(\frac{1}{|\mathbb{F}|^{0.1}} + \frac{1}{|\mathbb{F}|^{1/6}}\right) \\
&\leq \frac{1}{|\mathbb{F}|^{0.05}}.
\end{aligned}$$

Stage II: We now compose the dPCP $\mathcal{D}^{(I)}$ constructed in Stage I with another dPCP obtained from Theorem 5.2 as follows. Let $\mathcal{D}_{II}$ be the dPCP obtained from Theorem 5.2 by setting $h = 2$ and $l = 2i^*$. This dPCP will run on inputs of instance size $n_{II} \geq s^{(I)} = 2^{3(\lg N)^{9\varepsilon/80}} = 2^{3(\lg\lg N)^{1/4}}$. Thus, $m_{II} = \lg n_{II}/\lg h = 3(\lg N)^{9\varepsilon/80} = 3(\lg\lg N)^{1/4}$. Thus, $\mathcal{D}_{II}$ is a 2-query $(2i^*+1)$-answer dPCP on inputs of instance size n_{II}, randomness $R_{II} = c \cdot m_{II} \log|\mathbb{F}| = 3c(\lg N)^{1-71\varepsilon/80} = 3c\lg N/(\lg\lg N)^{71/36}$, answer size $s_{II} = 2(m_{II}h_{II})^2 < O((\lg N)^{9\varepsilon/40}) = O((\lg\lg N)^{1/2})$ and distributional soundness error $\delta_{II} = 1/|\mathbb{F}|^{0.1}$. Let $\mathcal{D}^{(II)}$ be the dPCP obtained by composing dPCP $\mathcal{D}^{(I)}$ obtained in the previous stage with dPCP $\mathcal{D}_{II}$, i.e., $\mathcal{D}^{(II)} = \mathcal{D}^{(I)} \circledast \mathcal{D}_{II}$. The encoding $\mathrm{LDE}_{\mathbb{F},h}$ has blocklength $|\mathbb{F}|^{m_{II}}$ and distance $1 - O(m_{II}h)/|\mathbb{F}| \geq 1 - 2/|\mathbb{F}|$. Hence, its agreement parameter is at least $1/|\mathbb{F}|^{1/6}$. Thus, dPCP $\mathcal{D}^{(II)}$ is a $2(i^*+1)$-prover dPCP with respect to the encoding $\mathrm{LDE}_{\mathbb{F},h_0}$ for the language ALG-CKTSAT$_{\mathbb{F}}$ with the following parameters: On inputs Φ of size N over $\mathbb{F}$, the dPCP $\mathcal{D}^{(II)}$ has randomness complexity $R^{(II)} = R^{(I)} + R_{II} + m_{II}\lg|\mathbb{F}| = O(\lg N)$, distributional soundness error $\delta^{(II)} = \delta^{(I)} + \delta_{II} + \eta_{II} \leq \frac{1}{|\mathbb{F}|^{0.05}}$ and answer size $s^{(II)} = s_{II} = O(\sqrt{\lg\lg N})$.

Stage III: We now compose dPCP $\mathcal{D}^{(II)}$ with the Hadamard based dPCP constructed in Theorem 5.3 to obtain our final dPCP. Let $\mathcal{D}_{III}$ be the Hadamard based dPCP constructed in Theorem 5.3 with $l = 2(i^*+1)$, i.e., $\mathcal{D}_{III} = \mathcal{D}_{\mathrm{QH},\mathbb{F},2(i^*+1)}$. $\mathcal{D}_{III}$ will be run on instances of size $n_{III} = O(\sqrt{\lg\lg N})$. Thus, $\mathcal{D}_{III}$ is a 2-prover $(2i^*+3)$-answer dPCP with respect to the encoding $\mathrm{QH}_{\mathbb{F}}$ for the language ALG-CKTSAT with the following parameters: on inputs of instance size n_{III}, it has randomness complexity $R_{III} = O(n_{III}^2 \lg|\mathbb{F}|) = O(\lg N)$, answer size $s_{III} = O(i^*)$ and distributional soundness error $\delta_{III} = 1/|\mathbb{F}|^{0.1}$. Furthermore, the blocklength of the encoding is $|\mathbb{F}|^{O(n_{III}^2)}$ and has agreement parameter $1/\sqrt{|\mathbb{F}|}$. The final dPCP $\mathcal{D}^{(III)}$ is given by composing $\mathcal{D}^{(II)}$ with $\mathcal{D}_{III}$, i.e., $\mathcal{D}^{(III)} = \mathcal{D}^{(II)} \circledast \mathcal{D}_{III}$. Note that $s^{(II)} \leq n_{III}$. Thus, the final dPCP $\mathcal{D}^{(III)}$ is a $2(i^*+2)$-prover dPCP with respect to the encoding $\mathrm{LDE}_{\mathbb{F},h_0}$ for the language ALG-CKTSAT$_{\mathbb{F}}$ with the following parameters: On inputs Φ of size N over $\mathbb{F}$, the dPCP $\mathcal{D}^{(III)}$ has randomness complexity $R^{(III)} = R^{(II)} + R_{III} + O(n_{III}^2\lg|\mathbb{F}|) = O(\lg N)$, distributional soundness error $\delta^{(III)} = \delta^{(II)} + \delta_{III} + \eta_{III} \leq \frac{1}{|\mathbb{F}|^{0.05}}$ and answer size $s^{(III)} = s_{III} = O(i^*) = O(1/\varepsilon)$.

Summarizing, we have constructed a $O(\lg\lg n/\lg\lg\lg n)$-prover dPCP $\mathcal{D}^{(III)} for$ ALG-CKTSAT$_{\mathbb{F}}$ with respect to the encoding $\mathrm{LDE}_{\mathbb{F},h_0}$ with the following parameters: on inputs Φ of size N, $\mathcal{D}^{(III)}$ has randomness complexity $O(\lg N)$, answer size $O(\lg\lg N/\lg\lg\lg N)$ and and distributional soundness error $N^{1/\mathrm{poly}\lg\lg N}$. This dPCP implies a PCP for CKTSAT with parameters as stated in Theorem 5.1. Note that the answer size is larger by a factor of $\log|\mathbb{F}| = \lg N/\mathrm{poly}\lg\lg N$ since the size of the output predicate is measured in terms of its Boolean circuit complexity as opposed to arithmetic complexity. □

5.3 Optimality of our parameter choices

In this section, we show the optimality of the parameters (upto constants) obtained in our Theorem 5.1 using the Reed-Muller based dPCP (Theorem 5.2) and the Hadamard based (Theorem 5.3) dPCP as building blocks in our composition paradigm. Of course, if one had an improved building block, then one can potentially improve on the construction.

Let N be the size of the instance and let δ be the soundness error of the construction. Define parameter ε as follows: $\log(1/\delta) = (\log N)^{1-\varepsilon}$. Consider any sequence of compositions of the Reed-Muller based dPCP and Hadamard based dPCP. Observe that the size of the Hadamard based dPCP is exponential in its input instance size. Hence, the first sequence of composition steps must involve only the Reed-Muller based dPCP wherein the size of the instance is sufficiently reduced to allow for composition with the Hadamard based dPCP.

We first argue that one needs to perform at least $\Omega(1/\varepsilon)$ steps of composition of the Reed-Muller based dPCP so that the instance size is sufficiently small to apply the Hadamard based dPCP. Suppose we perform t steps of composition of the Reed-Muller based dPCP wherein at the i-th step the instance size drops from N_{i-1} to N_i (here, $N_0 = N$). Since the error at each step is at most δ, the field size used in each stage of the Reed-Muller based dPCP must be at least $1/\delta = 2^{\log N^{1-\varepsilon}}$. To maintain polynomial size of the overall construction, each of the Reed-Muller based dPCPs used in the t steps of composition must satisfy $|\mathbb{F}_i|^{m_i} = N^{O(1)}$ where F_i and m_i are the field and dimension used in the construction of the Reed-Muller based dPCP used in the i-th stage of the composition. Hence, $m_i \leq O((\log N)^{\varepsilon})$. Thus, the reduction in size in the i-th step is at most $N_i \geq N_{i-1}^{1/m_i} = N_{i-1}^{1/(\log N)^{\varepsilon}}$, which implies inductively that the instance size after t steps of composition of the Reed-Muller based dPCP is at least $2^{\log N^{1-t\varepsilon}}$. Hence, to obtain a size that allows for composition with the Hadamard-based dPCP we must have at least $t = \Omega(1/\varepsilon)$ steps of composition.

We now account for the total randomness used in these $t = \Omega(1/\varepsilon)$ steps of composition. Since the error in each step is at most δ, the randomness uses in each step must be at least $\log(1/\delta) = (\log N)^{1-\varepsilon}$. Hence, the total randomness used in these t steps is at least $t \cdot \log(1/\delta) = \Omega(1/\varepsilon) \cdot (\log N)^{1-\varepsilon}$. Since the size of the entire construction is at most polynomial we must have that $1/\varepsilon \cdot (\log N)^{1-\varepsilon} = O(\log N)$. Solving for ε[7], we obtain that $\varepsilon \geq \log\log\log N/\log\log N$. Hence, the best soundness error obtained by a sequence of composition involving the Reed-Muller and Hadamard based dPCPs is at least $\delta = 2^{-(\log N)^{1-\varepsilon}} = N^{1/\mathrm{poly}\log\log N}$ proving optimality of the Theorem 5.1 construction.

6 Construction of specific dPCPs

In this section, we construct our two building blocks; the Hadamard-based dPCP (Theorem 5.3) and the Reed-Muller-based dPCP (Theorem 5.2). Our construction proceeds by adapting previous constructions of these objects which guaranteed only list-decoding soundness. We obtain distributional soundness by observing that if the dPCP satisfies list-decoding soundness and the encoding has very good distance (nearly 1), then the dPCP satisifies distributional soundness.

6.1 Preliminaries

Let $\mathbb{F}$ be a finite field.

Definition 6.1 (Hadamard). *The Hadamard encoding of a string $a \in \mathbb{F}^m$ is a function $h : \mathbb{F}^m \to \mathbb{F}$ defined by*

$$\forall \alpha \in \mathbb{F}^m, \quad h(\alpha) = \sum_i \alpha_i a_i.$$

Definition 6.2 (Quadratic Hadamard). *The Quadratic Hadamard encoding (QH encoding for short) of a string $a \in \mathbb{F}^m$, denoted QH_a, is defined to be the Hadamard encoding of the string $w = a \circ b \in \mathbb{F}^{m+m^2}$ where $b \in \mathbb{F}^{m^2}$ is defined by $b_{im+j} = a_i a_j$ for all $1 \leq i, j, \leq m$ (i.e. $b = a \otimes a$).*

[7] $1/\varepsilon \cdot (\log N)^{1-\varepsilon} = O(\log N)$ implies that $1/\varepsilon \leq O((\log N)^{\varepsilon}$ or equivalently $1/\varepsilon \cdot \log(1/\varepsilon) \leq O(\log\log N)$. This implies that $\varepsilon \geq \log\log\log N/\log\log N$.

Let $\varepsilon_i \in \mathbb{F}^{m+m^2}$ be the unit vector with 1 on the ith coordinate and zeros elsewhere. Observe that if $h = QH_a$ is the quadratic functions encoding of a, then for each $1 \le i,j \le m$,

$$h(\varepsilon_i) = a_i \qquad \text{and} \qquad h(\varepsilon_{i\cdot m+j}) = a_i a_j.$$

Let $H \subset \mathbb{F}$ and denote $h = |H|$. Fix an arbitrary 1-1 mapping $H \leftrightarrow [h] := \{0, 1, \dots, h-1\}$. We refer to elements in H as integers in $[h]$ relying on this mapping. For any $m > 0$ we map $x = (x_1, \dots, x_m) \in H^m$ to $\tilde{x} = x_1 + x_2 h + \ldots + x_m h^{m-1} + 1 \in [h^m]$.

Definition 6.3 (Low Degree Extension). *Given a string $a \in \mathbb{F}^n$, we define its* Low Degree Extension *with respect to $H \subseteq \mathbb{F}$, denoted LDE_a, as follows. Let m be the smallest integer such that $h^m \ge n$. Let $f : \mathbb{F}^m \to \mathbb{F}$ be the unique function whose degree in each variable is at most h, defined on H^m by*

$$\forall x \in H^m, \quad f(x) = \begin{cases} a_{\tilde{x}} & \tilde{x} \in [n]; \\ 0 & n < \tilde{x} \le h^m. \end{cases}$$

and extend f to $\mathbb{F}^m$ by interpolation, and set $LDE_a = f$.

Claim 6.4. *Let $a \in \mathbb{F}^{h^{m_1}}$, and let $b \in \mathbb{F}^{h^{m_2}-h^{m_1}}$, so that $a \circ b \in \mathbb{F}^{h^{m_2}}$. If $g_1 = LDE_a$ and $g_2 = LDE_{a\circ b}$ then*

$$\forall x_1, \dots, x_m \in \mathbb{F}^m, \quad g_1(x_1, \dots, x_{m_1}) = g_2(x_1, \dots, x_{m_1}, \bar{0}).$$

Proof. For each $(x_1, \dots, x_m) \in H^m$ we have

$$g_1(x_1, \dots, x_m) = a_{\tilde{x}} = (a \circ b)_{\tilde{x}} = g_2(x_1, \dots, x_m, \bar{0}).$$

Thus, g_1 and g_2 coincide for all points in H^m. As a function of $x_1, \dots, x_m$, g_1 and $g_2(x_1, \dots, x_m, \bar{0})$ have degree at most h in each variable, so they must coincide for all points in $\mathbb{F}^m$ too. □

Definition 6.5 (Curve). *Given $k < |\mathbb{F}|$ and a sequence of $k+1$ points $\tau = (z_0, \dots, z_k)$ in $\mathbb{F}^m$, define*

$$curve_\tau : \mathbb{F} \to \mathbb{F}^m$$

to be the polynomial function of degree at most k which satisfies $curve_\tau(i) = z_i$ for $i = 0, \dots, k$.

Definition 6.6 (Manifold). *Given $\tau = (z_1, \dots, z_k) \in \mathbb{F}^m$, and three points $x_1, x_2, x_3 \in \mathbb{F}^m$ define $\gamma_{z_1,\dots,z_k;x_1,x_2,x_3} : \mathbb{F}^4 \to \mathbb{F}^m$ to be the following degree $k+1$ function*

$$\gamma_{z_1,\dots,z_k;x_1,x_2,x_3}(t_0, t_1, t_2, t_3) = t_0 \cdot curve_{x_1,z_1,\dots,z_k}(t_1) + t_2 x_2 + t_3 x_3.$$

Observe that $\gamma_{z_1,\dots,z_k;x_1,x_2,x_3}$ contains the points $z_1, \dots, z_k$ and x_1, x_2, x_3.

We now state a low degree test, which has appeared in several places in the literature [RS97, DFK$^+$11, MR10]. First, a little notation. Supposed that $Q : \mathbb{F}^m \to \mathbb{F}$ is a function of degree $\le d$, and $\gamma_{z_1,\dots,z_k,x_1,x_2,x_3}(t_0, t_1, t_2, t_3) = t_0 \cdot curve_{x_1,z_1,\dots,z_k}(t_1) + t_2 x_2 + t_3 x_3$ is a manifold in $\mathbb{F}^m$ of degree at most $k+1$. Then the function $Q \circ \gamma : F^4 \to \mathbb{F}$ has degree at most $d(k+1)$ and can be specified by $\binom{d(k+1)}{4}$ coefficients. Given a manifold γ and a function $M(\gamma) : \mathbb{F}^4 \to \mathbb{F}$, we denote for each $x \in Im(\gamma)$

$$M(\gamma)[x] := M(\gamma)(t_1, \dots, t_4) \quad \text{for } t_1, \dots, t_4 \text{ such that } \gamma(t_1, \dots, t_4) = x.$$

The following lemma appears in [MR10, Lemma 4.4, Section 10.2 (in appendix)] and a similar lemma can be found in [Har10, Lecture 9].

Lemma 6.7 (Low Degree Test - Manifold vs. Point)**.** *Let $m, k, d > 0$, let $\delta = (mkd/|\mathbb{F}|)^{1/8}$, and let $z_1, \ldots, z_k \in \mathbb{F}^m$ be fixed. Let $Q : \mathbb{F}^m \to \mathbb{F}$ be an arbitrary function, supposedly of degree $\leq d$. There exists a list of $L \leq 2/\delta$ degree d functions $Q_1, \ldots, Q_L : \mathbb{F}^m \to \mathbb{F}$ such that the following holds. Let Γ be a collection of manifolds,*

$$\Gamma = \{\gamma_{z_1,\ldots,z_k,x_1,x_2,x_3}(t_0, t_1, t_2, t_3) = t_0 \cdot curve_{x_1,z_1,\ldots,z_k}(t_1) + t_2 x_2 + t_3 x_3\}_{x_1,x_2,x_3}$$

one per choice of $x_1, x_2, x_3 \in \mathbb{F}^m$. Let $M : \Gamma \to \mathbb{F}^{\binom{d(k+1)}{4}}$ specify for each γ the coefficients of a degree-$d(k+1)$ function supposedly equal to $Q \circ \gamma : \mathbb{F}^4 \to \mathbb{F}$. Then,

$$\Pr_{z,\gamma \ni z}[Q(z) = M(\gamma)[z] \text{ and } Q(z) \notin \{Q_1(z), \ldots, Q_L(z)\}] \leq \delta.$$

Finally, we state the following lemma, which gives a probabilistic verifier that inputs a predicate $\Phi : \mathbb{F}^n \to \{0, 1\}$ and a list of functions $F_1, \ldots, F_\ell : \mathbb{F}^n \to \mathbb{F}$, and checks that (a, b) are such that $\Phi(a) = 1$ and $b_i = F_i(a)$ for each $i = 1, \ldots, \ell$ ($b = F(a)$ for short).

Lemma 6.8 (Initial Verifier)**.** *Given a predicate $\Phi : \mathbb{F}^n \to \{0, 1\}$ and functions $F_1, \ldots, F_\ell : \mathbb{F}^n \to \mathbb{F}$ whose total circuit complexity is N, there is a randomized verifier V_0 that uses $O(\log |\mathbb{F}| + \log N)$ random bits and generates a quadratic polynomial $p : \mathbb{F}^m \to \mathbb{F}$ on $m = O(N)$ variables such that, given access to a proof $\pi = a \circ b \circ s \in \mathbb{F}^m$,*

- *If $\Phi(a) = 1$ and $F_i(a) = b_i$ for each $i = 1, \ldots, \ell$, then there is a* unique *string $s = s(a, b)$ such that*

$$\Pr_{p \sim V_0}[p(a, b, s) = 0] = 1.$$

- *If either $\Phi(a) = 0$ or $b_i \neq F_i(a)$ for some $1 \leq i \leq \ell$, or $s \neq s(a, b)$, then*

$$\Pr_{p \sim V_0}[p(a, b, s) = 0] \leq \frac{2}{|\mathbb{F}|}.$$

This verifier would be ideal except for one drawback: in order to evaluate $p(\pi)$ it makes an *unbounded* number of queries to the proof π.

Proof. (sketch) The proof of this lemma is standard: s will specify the values of all of the intermediate gates of the circuit computing Φ as well as the circuits computing $F_1, \ldots, F_\ell$. The validity of each intermediate computation step can be checked by a quadratic or linear equation over the entries in s. The verifier V_0 will use its randomness to generate a (pseudo)random sum of these equations (using an error correcting code, details are omitted). This can be expressed as a quadratic polynomial over the set of new variables. □

6.2 Hadamard based dPCP

In this section we construct a dPCP based on the Hadamard encoding, given formally in the following lemma.

Theorem 5.3 (Restated) (Hadamard based dPCP) *For any finite field $\mathbb{F}$, and any $\ell > 0$, there is a 2-prover ℓ+1-answer decodable PCP $\mathcal{D}_{\mathrm{QH},\mathbb{F}}$ with respect to the encoding $\mathrm{QH}_{\mathbb{F}}$ for the language* ALG-CKTSAT$_{\mathbb{F}}$ *with the following parameters: On inputs (i) a predicate $\Phi : \mathbb{F}^n \to \{0, 1\}$ and (ii) functions $F_1, \ldots, F_\ell : \mathbb{F}^n \to \mathbb{F}$ given by arithmetic circuits over $\mathbb{F}$ whose total size is N, the dPCP $\mathcal{D}_{\mathrm{QH},\mathbb{F}}$ has*

- *randomness complexity* $O(N^2 \log |\mathbb{F}|)$*,*
- *answer size* $s, s' = O(\ell)$*,*
- *perfect completeness, and*
- *distributional soundness error* $\leq 1/|\mathbb{F}|^{0.1}$*.*

We define the verifier for Theorem 5.3.

Decoder Protocol On input $\Phi, F_1, \ldots, F_\ell; j, r$, let V_0 be the verifier from Lemma 6.8, and let $\pi \in \mathbb{F}^m$ be the proof that V_0 expects. Our decoder V expects the B prover to hold the QH encoding of π and the A prover is expected to give restrictions of B to specified subspaces. It is known that with $O(1)$ queries into B the decoder could check that B is indeed a QH encoding of a valid proof π, as well as decode any quadratic function of π. The A prover is used to simulate this while making only one query to A and one to B. This is done by computing several query points for the former test, and then taking a random subspace S containing these points as well as a couple of uniformly random ones.

The low degree test (see Lemma 6.11 below) guarantees that if A's answer on the subspace S is consistent with B's answer on a random point in S, then B is linear (in other words, it is a Hadamard encoding of some string). The decoder will also perform some other tests on values in S which ensure that moreover B is a valid QH encoding of a valid π.

1. Computing the query points.
   (a) Choose $\beta, \gamma \in \mathbb{F}^m$ uniformly at random, and define $u_1, u_2, u_3 \in \mathbb{F}^{m+m^2}$ as follows:
   $$u_1 = \sum_{i=1}^{m} \beta_i \varepsilon_i, \quad u_2 = \sum_{i=1}^{m} \gamma_i \varepsilon_i, \quad u_3 = \sum_{i=1}^{m} \sum_{i'=1}^{m} \beta_i \gamma_{i'} \varepsilon_{im+i'}.$$
   (These are points for the multiplication test: if we already know that B is a Hadamard encoding of some string, then this test will ensure it is moreover a QH encoding.)
   (b) Draw a random quadratic polynomial $p(t_1, \ldots, t_m) = \alpha_0 + \sum_i \alpha_i t_i + \sum_{i,i'} \alpha_{ii'} t_i t_{i'}$ from the distribution of V_0 (from Lemma 6.8). To check that $p(\pi) = 0$ we define $z \in \mathbb{F}^{m_2}$ (for $m_2 = m + m^2$) by
   $$z = \sum_{i=1}^{m} \alpha_i \varepsilon_i + \sum_{i=1}^{m} \sum_{i'=1}^{m} \alpha_{ii'} \varepsilon_{im+i'}$$
   (If B were equal to QH_π for some string π then $B(z) + \alpha_0 = p(\pi)$ so the value of $B(z)$ could be used to check that $p(\pi) = 0$).
   (c) For each $i = 1, \ldots, \ell$ let $o_i = \varepsilon_{i+n}$. Let $o_{\ell+1}$ be the point in $\mathbb{F}^{m_2}$ corresponding to $j = (\delta_1, \ldots, \delta_{n+n^2}) \in \mathbb{F}^{n+n^2}$. (We are using here the fact that $\pi = a \circ b \circ s$ so the QF encoding of π contains in it the QF encoding of a. Explicitly, set
   $$o_{\ell+1} = \sum_{i=1}^{n} \delta_i \varepsilon_i + \sum_{i=1}^{n} \sum_{i'=1}^{n} \delta_{in+i'} \varepsilon_{im+i'}.$$

(These are the points to be output by the decoder.)

2. Choose $x_1, x_2, x_3 \in \mathbb{F}^{m_2}$ uniformly at random, consider the $(l+8)$-dimensional linear subspace
$$S = \mathrm{span}(x_1, x_2, x_3,\ u_1, u_2, u_3,\ z,\ o_1, \ldots, o_{\ell+1}) \subset \mathbb{F}^{m_2}.$$
We assume there is a canonical mapping that maps each subspace S to a particular basis $\vec{v_S} = \{v_1, \ldots, v_{\ell+8}\} \subset \mathbb{F}^{m_2}$ for S and send it to the A prover and let $A(\vec{v_S}) \in \mathbb{F}^{\ell+8}$ be the prover's answer. The answer specifies a linear function $A_S : S \to \mathbb{F}$ defined by
$$\forall t_1, \ldots, t_{\ell+8} \in \mathbb{F}, \qquad A_S\big(\sum_i t_i v_i\big) := \sum_{i=1}^{\ell+8} t_i \cdot A(\vec{v_S})_i$$

3. Send x_1 to the B prover and let $B(x_1)$ be its answer.

4. Reject unless
   (a) $A_S(x_1) = B(x_1)$, and
   (b) $A_S(z) + \alpha_0 = 0$.
   (c) $A_S(u_1)A_S(u_2) = A_S(u_3)$, and

5. Output $A_S(o_1), \ldots, A_S(o_\ell)$.

The decoding PCP will follow the protocol above, using its randomness R for selecting $p, \beta, \gamma, x_1, x_2, x_3$, and generate an output (q, φ, f, g) as follows:

- The queries q are $q_0 = \vec{v_S}$ to the first prover and $q_1 = x_1$ to the second prover.
- The predicate φ - rejects iff at least one of the tests in Items 4b and 4c reject.
- The function g computes $A_S(x_1)$ (for the consistency test in Item 4a).
- The functions $f_1, \ldots, f_{\ell+1}$ - compute $A_S(o_i)$ for $i = 1, \ldots, \ell+1$.

Lemma 6.9 (Perfect Completeness)**.** *The verifier has perfect completeness. Namely, for every $a \in \Phi^{-1}(1)$, there is a proof Π such that for every $j \in \mathbb{F}^{n+(n)^2}$ and every random string $R \in \{0,1\}^{O(N^2 \log|\mathbb{F}|)}$, the verifier on input $(\Phi, F; j, R)$ accepts and outputs $F_1(a), \ldots, F_\ell(a), QH_a(j)$.*

Proof. Let $b = F(a)$ and let s be the string promised in Lemma 6.8. Let $B : \mathbb{F}^{m_2} \to \mathbb{F}$ be the quadratic functions encodings of $\pi = a \circ b \circ s$. For each $\vec{v_S} = (v_i)_i$, let $A(\vec{v_S}) = (B(v_1), B(v_2), \ldots, B(v_{\ell+8}))$. We claim that $\Pi = (A, B)$ is a valid proof for $a \in \Phi^{-1}(1)$:

By definition B is a linear function on $\mathbb{F}^{m_2}$, so $A_S(x) = B(x)$ for all $x \in S$ and in particular the test in Item 4a passes. Also, by definition B is the Hadamard encoding of the string $\sigma = \pi \circ (\pi \otimes \pi)$, so $B(\varepsilon_i) = \sigma_i$ for all $1 \le i \le m + m^2$. Thus
$$B(\varepsilon_{i_1 m + i_2}) = \sigma_{i_1 m + i_2} = \pi_{i_1} \cdot \pi_{i_2} = \sigma_{i_1} \cdot \sigma_{i_2} = B(\varepsilon_{i_1}) \cdot B(\varepsilon_{i_2})$$

which, by linearity, implies that the test in Item 4c passes. Next, for Item 4b, we know that for every p generated by V_0,

$$0 = p(\pi) = \alpha_0 + \sum_{i=1}^{n} \alpha_i \pi_i + \sum_{i,i'=1}^{n} \alpha_{ii'} \pi_i \pi_{i'} = \alpha_0 + \sum_i \alpha_i B(\varepsilon_i) + \sum_{ii'} \alpha_{ii'} B(\varepsilon_{im+i'}) = \alpha_0 + B(z),$$

so $A_S(z) + \alpha_0 = B(z) + \alpha_0 = 0$ as required. It is finally easy to check that

$$A_S(\varepsilon_{i+n}) = B(\varepsilon_{i+n}) = \pi_{i+n} = b_i, \qquad i = 1, \dots, \ell.$$

Finally, for the $\ell + 1$st output,

$$A_S(o_{\ell+1}) = B(o_{\ell+1}) = \sum_{i=1}^{n} \delta_i B(\varepsilon_i) + \sum_{i=1}^{n} \sum_{i'=1}^{n} \delta_{in+i'} B(\varepsilon_{im+i'}) = QH_a(j)$$

where the last equality is due to the fact that the QH encoding of $\pi = a \circ b \circ s$ contains the QH encoding of a. More precisely, $QH_a(\varepsilon_{in+i'}) = B(\varepsilon_{im+i'})$ for all $0 \le i \le n$ and $1 \le i' \le n$. □

Lemma 6.10 (Distributional Soundness)**.** *The verifier above has soundness error at most $\delta = |\mathbb{F}|^{-0.1}$. Namely, given (Φ, F) for every proof $\Pi = (A, B)$, there are functions $\widetilde{\Pi}(\cdot), \widetilde{x}(\cdot)$ such that*

- *For each R, either $\Phi(\widetilde{x}(R)) = 1$ and $\widetilde{\Pi}(R)$ is a valid proof for "$x \in SAT(\Phi)$" or $\widetilde{\Pi}(R) = \bot$.*
- *For every j, there is probability at least $1 - \varepsilon$ that when R is chosen randomly and V is run on $(\Phi, F; j, R)$ it either rejects, or $\widetilde{\Pi}(R)$ is a proof that completely agrees with the answers of the provers A, B on the queries of V (in which case V's output is consistent with $\widetilde{x}(R)$).*

Proof. Fix $\Pi = (A, B)$. Given B, let $g_1, \dots, g_L$ be as in the low degree test below, Lemma 6.11. Let

$$L^* = \{i \in [L] \mid \; g_i = QH_\pi \text{ for } \pi = a \circ b \circ s \text{ s.t. } \Pr[V_0^\pi \text{ accepts}] = 1\}\,.$$

Set $\widetilde{\Pi}(R) = \bot$ if events $E1$ or $E2$ occurred, where

E1: $B(x_1) \notin \{g_i(x_1) \mid i \in L^*\}$.

E2: there is more than one index $i \in L^*$ for which $B(x_1) = g_i(x_1)$.

Otherwise, there is a unique $i \in L^*$ such that $B(x_1) = g_i(x_1)$. By assumption g_i is the QH encoding of some $\pi = a \circ b \circ s$ for which $\Phi(a) = 1$ and $F(a) = b$ and $s = s(a, b)$. So we set $\widetilde{x}(R) = a$ and set $\widetilde{\Pi}(R) = (A_R, B_R)$ to be a valid proof for $a \in \Phi^{-1}(1)$ so that $B_R = g_i$.

Now fix an arbitrary $j \in \mathbb{F}^{n+n^2}$, and let R be chosen uniformly at random. We claim that the probability that the verifier accepts and yet the view of Π and of $\widetilde{\Pi}(R)$ differ is very small. We analyze two cases.

- **Accept and $\widetilde{\Pi}(R) = \bot$:** This event can be bounded by

  $$\Pr[\text{Accept and } E1] + \Pr[E2] \le \max\left(2/\left|\mathbb{F}\right|^{1/6}, \frac{4L}{|\mathbb{F}|}\right) + \binom{L}{2} / \left|\mathbb{F}\right|$$

  where the second item is bounded due to the large distance of the Hadamard code, and the first item is bounded as follows. If $B(x_1) \notin \{g_i(x_1) \mid i \in [L]\}$ then Lemma 6.11 implies that the probability of acceptance is small. If however $B(x_1) = g_i(x_1)$ for some $i \in [L] \setminus L^*$ then for each $i \in [L] \setminus L^*$ Lemma 6.12 shows that the acceptance probability is small, and we take a union bound over all such i.

- **Accept and $\widetilde{\Pi}(R)|_q \neq \Pi|_q$:** We defined $\widetilde{\Pi}(R)$ so that $B_R(x_1) = B(x_1) = g_i(x_1)$ for some $i \in L^*$. So this event occurs if $A_S \neq g_i|_S$. We observe that this event is contained in $\cup_{i \in L^*} E_i$ where E_i is the event that $A_S \neq g_i|_S$ yet $A_S(x_1) = g_i(x_1)$. For each i this event has probability at most $2/|\mathbb{F}|$, and we take a union bound over $i \in L^*$.

□

The proof of soundness is based on the following lemma, which has appeared in several places in the literature. The following lemma appears in [MR10, Proposition 11.0.3].

Lemma 6.11 (Subspace vs. Point - linearity testing list decoding soundness). *Let $\delta = 2/|\mathbb{F}|^{1/6}$. Given a pair of provers A, B, there is a list of $L \leq 2/\delta^3$ linear functions $g_1, \ldots, g_L : \mathbb{F}^{m_2} \to \mathbb{F}$ such that the probability that the decoder does not reject yet $B(x_1) \notin \{g_1(x_1), \ldots, g_L(x_1)\}$ is at most $O(\delta)$.*

The following claim shows that if B's answers are a *linear* function, then the verifier rejects unless B is a QH encoding of a valid proof.

Lemma 6.12. *Suppose that $B : \mathbb{F}^{m_2} \to \mathbb{F}$ is a* linear function. *Let $\pi = a \circ b \circ s$ be defined by*

$$a = B(\varepsilon_1) \ldots B(\varepsilon_n) \text{ and } b = B(\varepsilon_{n+1}), \ldots, B(\varepsilon_{n+\ell}) \text{ and } s = B(\varepsilon_{n+\ell+1}), \ldots, B(\varepsilon_m).$$

Assume that either $\Phi(a) = 0$ or $F(a) \neq b$ or $s \neq s(a,b)$ or $B \neq QH_\pi$. Then, for all provers A, for all j, $\Pr_R[V^{A,B}(\Phi, F; j; R) \text{ accepts}] \leq 4/|\mathbb{F}|$.

Proof. Assume first that $B = QH_\pi$, but either $\Phi(a) = 0$ or $F(a) \neq b$ or $s \neq s(a,b)$. By Lemma 6.8, when choosing a random quadratic p, there is at most probability $2/|\mathbb{F}|$ that $p(\pi) = 0$. So let p be such that $p(\pi) \neq 0$. Since B is the QH encoding of π, by the definition of z

$$B(z) + \alpha_0 = p(\pi) \neq 0.$$

However, V accepts so Item 4b passing implies that $A_S(z) + \alpha_0 = 0$. This means that as linear functions on S, $A_S \not\equiv B|_S$. It remains to observe that conditioned on S, x_1 is drawn almost uniformly from S, so the probability that Item 4a does not reject is at most $1/|\mathbb{F}| + neg \leq 2/|\mathbb{F}|$. Altogether the total probability of accepting in this case is at most $4/|\mathbb{F}|$.

We move to analyze the case where $\Phi(a) = 1$ and $F(a) = b$ and $s = s(a,b)$ but $B \neq QH_\pi$. There must be some indices i_1, i_2 such that $B(\varepsilon_{i_1})B(\varepsilon_{i_2}) \neq B(\varepsilon_{i_1 m + i_2})$. We need to upper bound the probability over random β, γ that the following expression is zero:

$$B(u_1)B(u_2) - B(u_3) = \sum_{i=1}^{m} \sum_{i'=1}^{m} \beta_i \gamma_{i'} (B(\varepsilon_i)B(\varepsilon_{i'}) - B(\varepsilon_{im+i'})).$$

Let us fix the value of β_i (for $i \neq i_1$) and $\gamma_{i'}$ (for $i' \neq i_2$) arbitrarily. The remaining expression becomes a non-zero quadratic polynomial in $\beta_{i_1}, \gamma_{i_2}$, so it can be zero with probability at most $2/|\mathbb{F}|$ over the choice of $\beta_{i_1}, \gamma_{i_2}$.

Consider the event where it is not zero. Since Item 4c accepts, $A_S(u_1)A_S(u_2) = A_S(u_3)$ so $A_S(u_i) \neq B(u_i)$ for some $i \in \{1,2,3\}$. So as linear functions $A_S \neq B|_S$ and the probability of Item 4a passing is at most $1/|\mathbb{F}| + neg \leq 2/|\mathbb{F}|$ (where neg is a small probability introduced because x_1 is only almost uniform in S conditioned on S). Altogether we get a bound of $4/|\mathbb{F}|$ in this case as well. □

6.3 Reed-Muller based dPCP

Theorem 5.2 (Restated) (Reed-Muller based dPCP) *For any finite field $\mathbb{F}$, and parameter h such that $1 < h < |\mathbb{F}|^{0.01}$ and any $\ell > 0$, there is a 2-prover $\ell + 1$-answer decodable PCP $\mathcal{D}$ with respect to the encoding $\mathrm{LDE}_{\mathbb{F},h}$ for the language $\textsc{Alg-CktSAT}_{\mathbb{F}}$ with the following parameters: On inputs (i) a predicate $\Phi : \mathbb{F}^n \to \{0,1\}$ and (ii) functions $F_1, \dots, F_\ell : \mathbb{F}^n \to \mathbb{F}$ given by arithmetic circuits over $\mathbb{F}$ whose total size is N, the dPCP $\mathcal{D}$ has (let $m = \log N / \log h$)*

- *randomness complexity $O(\log N + m \log |\mathbb{F}|) = O(\log N + \log |\mathbb{F}|)$,*
- *answer size $s, s' = O(m(m+\ell))$,*
- *and distributional soundness error $1/|\mathbb{F}|^{0.1}$.*

Fix $H \subseteq \mathbb{F}$ throughout this section and denote $h = |H|$.

We construct a PCP decoder that receives as proof a sequence of low degree polynomials that allow it to simulate the actions of the initial verifier V_0 (from Lemma 6.8) using fewer queries. We first construct this sequence of polynomials g_1, g_2, g_3, g_4, then describe a "verification protocol" checking that a given sequence of polynomials have the intended form, and finally describe the PCP decoder.

Constructing the low degree functions Let $\Phi, F_1, \dots, F_\ell$ be the input.

1. Suppose $a \in \Phi^{-1}(1)$ and let $b_i = F_i(a)$ for all $i = 1, \dots, \ell$, and let $s = s(a,b)$ so that the initial verifier V_0 accepts $\pi = a \circ b \circ s$ with probability 1. Wlog we assume that $n = |a|$ is a power of h, and also $n_1 = |\pi| + 1$ is a power of h. This can be arranged by padding a with zeros and then padding π with zeros and changing V_0 accordingly.

2. Let $m_1 = m = \log_h n$ and define $g_1 = LDE_a : \mathbb{F}^{m_1} \to \mathbb{F}$ (see Definition 6.3),

3. Let $m_2 = \log_h n_1$ and let $g_2 = LDE_{\pi \circ 1} : \mathbb{F}^{m_2} \to \mathbb{F}$ be the low degree extension of the string obtained by appending a 1 to π. By Claim 6.4 $g_2(x_1, \dots, x_{m_1}, \bar{0}) = g_1(x_1, \dots, x_{m_1})$. Let $z_0 \in H^{m_2}$ be the point associated with the last element in $\pi \circ 1$, i.e. such that $g_2(z_0) = 1$.

4. Let $m_3 = 2m_2$ and let $g_3 : \mathbb{F}^{m_3} \to \mathbb{F}$ be defined by $g_3(x,y) = g_2(x) \cdot g_2(y)$. Note that the degree of g_3 is at most $m_3 h$.

5. Let P_0 be the set of all quadratic polynomials generated by V_0 on input (Φ, F). Fix some $p \in P_0$, $p(t_1, \dots, t_{n_1}) = p_0 + \sum_i p_i t_i + \sum_{ii'} p_{ii'} t_i t_{i'}$. Define the function $\hat{p} : \mathbb{F}^{m_3} \to \mathbb{F}$ as follows. For $1 \le i \le h^{m_2}$ let $\vec{i}$ be the corresponding element in H^{m_2} (see discussion before Definition 6.3). For each $i < i' \in H^{m_2} \setminus \{z_0\}$, set

$$\hat{p}(z_0, z_0) = p_0, \quad \hat{p}(z_0, \vec{i}) = p_i, \quad \hat{p}(\vec{i}, \vec{i'}) = p_{ii'}, \quad \hat{p}(z) = 0 \text{ for all other } z \in H^{m_3}.$$

   Extend $\hat{p}$ from H^{m_3} to $\mathbb{F}^{m_3}$ by interpolation. The degree of $\hat{p}$ is at most $m_3 h$.

   This definition ensures that for $\sigma := (\pi \circ 1) \in \mathbb{F}^{n_1}$,

$$p(\sigma) = p_0 + \sum_i p_i \sigma_i + \sum_{ii'} p_{ii'} \sigma_i \sigma_{i'} = \sum_{x \in H^{m_3}} \hat{p}(x) \cdot g_3(x) = 0. \tag{6.1}$$

6. Define low degree functions $s^p_1, \dots, s^p_{m_3} : \mathbb{F}^{m_3} \to \mathbb{F}$ as nested partial sums of g_3 as follows.

$$s^p_{m_3}(x) = \hat{p}(x) \cdot g_3(x) \qquad \text{and} \qquad s^p_{i-1}(x) = \sum_{h \in H} s^p_i(x_1, \dots, x_{i-1}, h, 0, \dots, 0) \quad 1 < i \leq m_3$$

   The degree of each s^p_i is at most $2m_3h$.

7. Bundling: From the polynomials g_3 and s^p_i for each i and $p \in P_0$, we will now create one single polynomial g_4 that 'bundles' them together. Let us number them as $q_1, \dots, q_T$ for $T = m_3 |P_0| + 1$ and let $t = \lceil \log_h(T+1) \rceil = O(m_3)$.

   Let $m_4 = m_3 + t$ and define $g_4 : \mathbb{F}^{m_4} \to \mathbb{F}$ by

$$g_4(y_1, \dots, y_t, x) = \sum_{i=1}^{T} q_i(x) \cdot w_i(y_1, \dots, y_t) \tag{6.2}$$

   where $w_i : \mathbb{F}^t \to \mathbb{F}$ is the degree th polynomial for which $w_i(y_1, \dots, y_t) = 1$ iff $y_1, \dots, y_t$ is the H-ary representation of i, and zero for all other $y_1, \dots, y_t \in H^t$. The degree of g_4 is at most

$$d = ht + 2hm_3 = O(hm).$$

By construction the function g_4 contains inside of it (as restrictions) the functions g_1, g_2, g_3:

Claim 6.13. *There is a sequence of 1-1 (linear) mappings*

$$\mathbb{F}^{m_1} \overset{\sigma_1}{\to} \mathbb{F}^{m_2} \overset{\sigma_2}{\to} \mathbb{F}^{m_3} \overset{\sigma_3}{\to} \mathbb{F}^{m_4}$$

such that for each $x \in \mathbb{F}^{m_3}$ we have $g_4(\sigma_3(x)) = g_3(x)$; and for each $x \in \mathbb{F}^{m_2}$ we have $g_3(\sigma_2(x)) = g_2(x)$; and for each $x \in \mathbb{F}^{m_1}$ we have $g_2(\sigma_1(x)) = g_1(x)$. □

Proof. We map a point $x_1 \in \mathbb{F}^{m_1}$ to $(x_1, \bar{0}) \in \mathbb{F}^{m_2}$. We map a point $x_2 \in \mathbb{F}^{m_2}$ to $(z_0, x_2) \in \mathbb{F}^{m_3}$. We map a point $x_3 \in \mathbb{F}^{m_3}$ to $(y, x_3) \in \mathbb{F}^{m_4}$ in the domain of g_4, where $y \in \mathbb{F}^t$ is the index of g_3 in the bundling. □

Answers of the B prover will correspond to point-evaluations of g_4, and answers of the A prover will correspond to restrictions of g_4 to certain low-degree curves. A low degree curve, see Definition 6.5, is specified by a tuple of points in $\mathbb{F}^{m_4}$ through which it passes. This tuple contains

- Points for the verification check protocol, see below
- Output points: The are $\ell + 1$ points whose values will give $\ell + 1$ answers for the decoder. These answers are the values of $b_1, \dots, b_\ell$ and the j-th element in the encoding LDE_a.

  Let $v_1, \dots, v_\ell \in \mathbb{F}^{m_2}$ be the points in the domain of g_2 that correspond to $b_1, \dots, b_\ell$. For each i, let $o_i = \sigma_3(\sigma_2(v_i))$ be the corresponding point in the domain of g_4 (as in the claim above).

  Let $v_{\ell+1} \in \mathbb{F}^{m_1}$ be the jth point in the domain of $g_1 = LDE_a$ (we assume some canonical numbering of the indices of the LDE encoding). Let $o_{\ell+1} = \sigma_3(\sigma_2(\sigma_1(v_{\ell+1})))$ be the corresponding point in the domain of g_4.
- Random points

Verification Protocol The verification protocol accesses functions $\tilde{g}_3$ and $\{\tilde{s}_i^p\}_{i,p}$ and checks (locally) that they have the correct form, as intended in the construction, i.e. that there is some valid proof π such that they are equal to g_3 and s_i^p as in the construction above.

1. Check that $\tilde{g}_3(z_0, z_0) = 1$.
2. Choose two random points $x, y \in \mathbb{F}^{m_2}$, so that $(x, y) \in \mathbb{F}^{m_3}$. Check that $\tilde{g}_3(z_0, x) \cdot \tilde{g}_3(z_0, y) = \tilde{g}_3(x, y)$.
3. Choose a random quadratic p by simulating V_0, and compute $\hat{p} : \mathbb{F}^{m_3} \to \mathbb{F}$ as in Item 5 in the expected proof.
4. Do the sumcheck: Choose a random point $x = (x_1, \ldots, x_{m_3}) \in \mathbb{F}^{m_3}$, and do
   (a) Check that $\tilde{s}^p_{m_3}(x) = \hat{p}(x) \cdot \tilde{g}_3(x)$
   (b) For each $1 < i \leq m_3$ check that $\tilde{s}^p_{i-1}(x_1, \ldots, x_{i-1}, \bar{0}) = \sum_{h \in H} \tilde{s}^p_i(x_1, \ldots, x_{i-1}, h, \bar{0})$
   (c) Check that $\sum_{h \in H} \tilde{s}^p_1(h, \bar{0}) = 0$.

The verification protocol accesses $k = (h+1)m_3 + 5$ points in the domains of $\tilde{g}_3$ and $\tilde{s}_i^p$. Let $u_1, \ldots, u_k \in \mathbb{F}^{m_4}$ be the corresponding points in the domain of g_4 (as in Claim 6.13).

The PCP Decoder protocol

1. Compute the output points $o_1, \ldots, o_{\ell+1}$ as above.
2. Use the randomness R to compute $u_1, \ldots, u_M$ using the verification protocol.
3. Use the randomness R to choose $x_1, x_2, x_3 \in \mathbb{F}^{m_4}$ uniformly. Let $\gamma = \gamma_{o_1, \ldots, o_{\ell+1}, u_1, \ldots, u_M; x_1, x_2, x_3}$ be the manifold as in Definition 6.6, such that γ contains the points $o_1, \ldots, o_{\ell+1}, u_1, \ldots, u_M$ as well as x_1, x_2, x_3 and has degree at most $\ell + 1 + k + 1 = \ell + O(hm)$. Send γ to A and let A's answer $A(\gamma)$ be the coefficients of a function $\mathbb{F}^4 \to \mathbb{F}$ whose degree is at most $d' \leq d(M + \ell + 2)$. This function is supposed to equal $B \circ \gamma$.

   Let $A_\gamma : Im(\gamma) \to \mathbb{F}$ be defined for each $x \in Im(\gamma)$ as $A_\gamma(x) := A(\gamma)(t)$ where $x = \gamma(t)$. Clearly A_γ can be computed from $A(\gamma)$.
4. Send x_1 to the B prover and let $B(x_1)$ be its answer. Reject unless $A_\gamma(x_1) = B(x_1)$.
5. Simulate the checks of Item 4 using the values $A_\gamma(u_1), \ldots, A_\gamma(u_M)$. Reject unless all of the checks succeed.
6. Output $A_\gamma(o_1), \ldots, A_\gamma(o_{\ell+1})$.

To summarize, the PCP decoder computes (q, φ, f, g) as follows:

- The queries q: $q_0 = \gamma$ is the query to the A prover and $q_1 = x_1$ is the query to the B prover.
- The predicate φ rejects unless all of the checks in Item 4 pass.
- The function g computes $A_\gamma(x_1)$ (for the consistency test).

- The functions $f_1, \dots, f_{\ell+1}$ - compute $A_\gamma(o_i)$ for $i = 1, \dots, \ell+1$.

This completes the description of the PCP decoder, and we proceed to prove its correctness.

Lemma 6.14 (Perfect Completeness)**.** *The PCP decoder has perfect completeness. Namely, for every $a \in \Phi^{-1}(1)$, there is a proof Π such that for every $j \in \mathbb{F}^m$ and every random string R, the verifier on input $(\Phi, F; j, R)$ accepts and outputs $F_1(a), \dots, F_\ell(a), LDE_a(j)$.*

Proof. If $\Phi(a) = 1$ and $F(a) = b$ then there is a proof $\pi = abs \in \mathbb{F}^{n_1}$ such that for every quadratic p generated by the initial verifier V_0, $p(\pi) = 0$. Compute from π the function g_4 as described in the "expected proof" section above, and let B answer according to g_4. The checks in Item 4 will always succeed. It remains to take A to be the restrictions of B to the manifolds and then the consistency checks will pass and the verifier will always output as required. □

Lemma 6.15 (Distributional Soundness)**.** *The verifier above has soundness error at most $\delta = |\mathbb{F}|^{-0.1}$. Namely, given (Φ, F) for every proof $\Pi = (A, B)$, there are functions $\widetilde{\Pi}(\cdot), \widetilde{x}(\cdot)$ such that*

- *For each R, either $\Phi(\widetilde{x}(R)) = 1$ and $\widetilde{\Pi}(R)$ is a valid proof for "$x \in SAT(\Phi)$" or $\widetilde{\Pi}(R) = \perp$.*
- *For every j, there is probability at least $1 - \varepsilon$ that when R is chosen randomly and V is run on $(\Phi, F; j, R)$ it either rejects, or $\widetilde{\Pi}(R)$ is a proof that completely agrees with the answers of the provers A, B on the queries of V (in which case V's output is consistent with $\widetilde{x}(R)$).*

Proof. Fix $\Pi = (A, B)$. Given B, let $Q_1, \dots, Q_L$ be degree $\le d$ functions as in Lemma 6.7. We say that Q_i is a valid proof for $a \in \Phi^{-1}(1)$ when the B prover answers according to Q_i, there is an A prover causing the verifier to always output consistently with LDE_a. Let $I \subset [L]$ be the indices for which Q_i is a valid proof for some a, b.

For each R note that in the verifier protocol x_1 is chosen (based on R but) independently of j. Set $\widetilde{\Pi}(R) = \perp$ if events $E1$ or $E2$ occurred, where

E1: $B(x_1) \notin \{Q_i(x_1) \mid i \in I\}$.

E2: there is more than one index $i \in I$ for which $B(x_1) = Q_i(x_1)$.

Otherwise, there is a unique $i \in I$ such that $B(x_1) = Q_i(x_1)$. By assumption Q_i is a valid proof for some $a \in \Phi^{-1}(1)$ so we set $\widetilde{x}(R) = a$ and set $\widetilde{\Pi}(R) = (A_R, B_R)$ to be a valid proof for a.

Now fix an arbitrary $j \in \mathbb{F}^m$, and let R be chosen uniformly at random. We claim that the probability that the verifier accepts and yet the view of Π and of $\widetilde{\Pi}(R)$ differ is very small. We analyze two cases.

- **Accept and $\widetilde{\Pi}(R) = \perp$:** This event can be bounded by

$$\Pr[\text{Accept and } E1] + \Pr[E2] \le \max\left(O(|\mathbb{F}|)^{-0.1}), O(Lmd/|\mathbb{F}|)\right) + \binom{L}{2} \cdot d/|\mathbb{F}|$$

  where the second item is bounded due to the large distance between degree d functions, and the first item is bounded as follows. If $B(x_1) \notin \{Q_i(x_1) \mid i \in [L]\}$ then Lemma 6.7 with parameters $m = m_4, k' = k + \ell = O(hm), d$ implies that the probability of acceptance is at most

$$(mk'd/|\mathbb{F}|)^{1/8} = O(h^2m^3|\mathbb{F}|)^{1/8} \le |\mathbb{F}|^{-0.1}$$

(the last inequality is true since $h \leq |\mathbb{F}|^{0.01}$ and for large enough n since $m = \log n / \log h$ and $|\mathbb{F}| \gg \text{poly} \log n$.)

If however $B(x_1) = Q_i(x_1)$ for some $i \in [L] \setminus I$ then for each $i \in [L] \setminus I$ Lemma 6.16 below shows that the acceptance probability is at most $O(md/|\mathbb{F}|)$, and we take a union bound over all such i.

- **Accept and $\widetilde{\Pi}(R)|_q \neq \Pi|_q$:** We defined $\widetilde{\Pi}(R)$ so that $B_R(x_1) = B(x_1) = Q_i(x_1)$ for some $i \in I$. So this event occurs if $A_\gamma \neq Q_i|_\gamma$. We observe that this event is contained in $\cup_{i \in I} E_i$ where E_i is the event that $A_\gamma \neq Q_i|_\gamma$ yet $A_\gamma(x_1) = Q_i(x_1)$. For each i this event has probability at most $dd'/|\mathbb{F}|$, and we take a union bound over $i \in I$. The total probability of error in this event is at most $Ldd'/|\mathbb{F}|$.

□

Lemma 6.16 (Soundness against a low degree prover)**.** *Suppose that $B : \mathbb{F}^{m_4} \to \mathbb{F}$ is a function of degree at most $d = ht + 2hm_3$, and let $g, \{s_i^p\}$ be its unbundling. Suppose further that g is consistent with a, b such that either $\Phi(a) = 0$ or $b \neq F(a)$. Then, for all provers A, the probability that the verifier accepts is at most $O(md/|\mathbb{F}|)$.*

Proof. (of Lemma 6.16) Assume that $\Phi(a) = 0$ or $F(a) \neq b$ and denote $\sigma = abs$. The probability that a random quadratic p drawn according to V_0 (from Lemma 6.8) will satisfy $p(\sigma) = 0$ is at most $O(1/|\mathbb{F}|)$. Suppose $p(\sigma) \neq 0$. This means that

$$\sum_{x \in H^{m_3}} \hat{p}(x) g(x) \neq 0. \tag{6.3}$$

Observe that if the check in Item 4c passes then either

$$s^p_{m_3} \neq \hat{p} \cdot g, \tag{6.4}$$

or, for some i, as functions of $x_1, \ldots, x_{i-1}$,

$$s^p_{i-1}(x_1, \ldots, x_{i-1}, \bar{0}) \neq \sum_{h \in H} s^p_i(x_1, \ldots, x_{i-1}, h, \bar{0}). \tag{6.5}$$

Otherwise,

$$\sum_{x \in H^{m_3}} \hat{p}(x) g(x) = \sum_{x_1, \ldots, x_{m_3} \in H} s^p_{m_3}(x_1, \ldots, x_{m_3}) =$$

$$= \sum_{x_1, \ldots, x_{m_3 - 1} \in H} s^p_{m_3 - 1}(x_1, \ldots, x_{m_3 - 1}, 0) = \ldots = \sum_{x_1 \in H} s^p_{m_3 - 1}(x_1, \bar{0}) = 0$$

contradicting (6.3). The verifier checks each of these m_3 equalities in (6.4) and (6.5)) on a random point (in Items 4a and 4b), so the probability of acceptance is at most $m_3 \cdot \frac{d}{|\mathbb{F}|}$. □

References

[ALM$^+$98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. *Proof verification and the hardness of approximation problems*. J. ACM, 45(3):501–555, May 1998. (Preliminary version in *33rd FOCS*, 1992). eccc:TR98-008, doi:10.1145/278298.278306. 1, 6

[AS98] Sanjeev Arora and Shmuel Safra. *Probabilistic checking of proofs: A new characterization of NP*. J. ACM, 45(1):70–122, January 1998. (Preliminary version in *33rd FOCS*, 1992). doi:10.1145/273865.273901. 1, 3, 6

[AS03] Sanjeev Arora and Madhu Sudan. *Improved low-degree testing and its applications*. Combinatorica, 23(3):365–426, 2003. (Preliminary version in *29th STOC*, 1997). eccc:TR97-003, doi:10.1007/s00493-003-0025-0. 2, 3, 6

[BGH$^+$06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. *Robust PCPs of proximity, shorter PCPs and applications to coding*. SIAM J. Comput., 36(4):889–974, 2006. (Preliminary version in *36th STOC*, 2004). eccc:TR04-021, doi:10.1137/S0097539705446810. 2, 3

[BGKW88] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. *Multi-prover interactive proofs: How to remove intractability assumptions*. In *Proc.* 20*th ACM Symp. on Theory of Computing (STOC)*, pages 113–131. 1988. doi:10.1145/62212.62223. 5

[BGLR93] Mihir Bellare, Shafi Goldwasser, Carsten Lund, and Alexander Russell. *Efficient probabilistically checkable proofs and applications to approximation*. In *Proc.* 25*th ACM Symp. on Theory of Computing (STOC)*, pages 294–304. 1993. doi:10.1145/167088.167174. 1, 2

[BS08] Eli Ben-Sasson and Madhu Sudan. *Short PCPs with polylog query complexity*. SIAM J. Comput., 38(2):551–607, 2008. (Preliminary version in *37th STOC*, 2005). eccc:TR04-060, doi:10.1137/050646445. 2, 3

[CK09] Julia Chuzhoy and Sanjeev Khanna. *Polynomial flow-cut gaps and hardness of directed cut problems*. J. ACM, 56(2), 2009. (Preliminary version in *39th STOC*, 2007). doi:10.1145/1502793.1502795. 5

[DFK$^+$11] Irit Dinur, Eldar Fischer, Guy Kindler, Ran Raz, and Shmuel Safra. *PCP characterizations of NP: Toward a polynomially-small error-probability*. Comput. Complexity, 20(3):413–504, 2011. (Preliminary version in *31st STOC*, 1999). eccc:TR98-066, doi:10.1007/s00037-011-0014-4. 2, 3, 4, 5, 6, 10, 25

[DH13] Irit Dinur and Prahladh Harsha. *Composition of low-error 2-query PCPs using decodable PCPs*. SIAM J. Comput., 42(6):2452–2486, 2013. (Preliminary version in *51st FOCS*, 2009). eccc:TR09-042, doi:10.1137/100788161. 2, 3, 4, 6, 10, 12

[DHK15] Irit Dinur, Prahladh Harsha, and Guy Kindler. *Polynomially low error PCPs with polyloglog n queries via modular composition*. In *Proc.* 47*th ACM Symp. on Theory of Computing (STOC)*. 2015. (To appear). doi:10.1145/2746539.2746630. 0

[Din07] Irit Dinur. *The PCP theorem by gap amplification*. J. ACM, 54(3):12, 2007. (Preliminary version in *38th STOC*, 2006). eccc:TR05-046, doi:10.1145/1236457.1236459. 2, 3

[DM11] Irit Dinur and Or Meir. *Derandomized parallel repetition via structured PCPs*. Comput. Complexity, 20(2):207–327, 2011. (Preliminary version in *25th Conference on Computation Complexity*, 2010). arXiv:1002.1606, doi:10.1007/s00037-011-0013-5. 6

[DR06] Irit Dinur and Omer Reingold. *Assignment testers: Towards a combinatorial proof of the PCP Theorem*. SIAM J. Comput., 36:975–1024, 2006. (Preliminary version in *45th FOCS*, 2004). doi:10.1137/S0097539705446962. 2, 3

[DS04] Irit Dinur and Shmuel Safra. *On the hardness of approximating label-cover*. Inform. Process. Lett., 89(5):247–254, March 2004. eccc:TR99-015, doi:10.1016/j.ipl.2003.11.007. 6

[FGL^{+}96] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. *Interactive proofs and the hardness of approximating cliques*. J. ACM, 43(2):268–292, March 1996. (Preliminary version in *32nd FOCS*, 1991). doi:10.1145/226643.226652. 1

[FK95] Uriel Feige and Joe Kilian. *Impossibility results for recycling random bits in two-prover proof systems*. In *Proc. 27th ACM Symp. on Theory of Computing (STOC)*, pages 457–468. 1995. doi:10.1145/225058.225183. 6

[Har10] Prahladh Harsha. *Limits of approximation algorithsm: PCPs and unique games.*, 2010. A course on PCPs at TIFR and IMSc. 25

[Mos14] Dana Moshkovitz. *An approach to the Sliding Scale Conjecture via parallel repetition for low degree testing*. Technical Report TR14-030, Elect. Colloq. on Comput. Complexity (ECCC), 2014. eccc:TR14-030. 3, 6

[MR10] Dana Moshkovitz and Ran Raz. *Two-query PCP with subconstant error*. J. ACM, 57(5), 2010. (Preliminary version in *49th FOCS*, 2008). eccc:TR08-071, doi:10.1145/1754399.1754402. 2, 3, 6, 10, 12, 25, 30

[Raz98] Ran Raz. *A parallel repetition theorem*. SIAM J. Comput., 27(3):763–803, June 1998. (Preliminary version in *27th STOC*, 1995). doi:10.1137/S0097539795280895. 6

[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 *Proc. 29th ACM Symp. on Theory of Computing (STOC)*, pages 475–484. 1997. doi:10.1145/258533.258641. 2, 3, 6, 25

[Sze99] Mario Szegedy. *Many-valued logics and holographic proofs*. In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, eds., *Proc. 26th International Colloq. of Automata, Languages and Programming (ICALP)*, volume 1644 of *LNCS*, pages 676–686. Springer, 1999. doi:10.1007/3-540-48523-6_64. 3

ECCC ISSN 1433-8092

http://eccc.hpi-web.de