# Improved 3LIN Hardness via Linear Label Cover

**Prahladh Harsha**
School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai, India
http://www.tcs.tifr.res.in/~prahladh/
prahladh@tifr.res.in

**Subhash Khot**
Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, USA

**Euiwoong Lee**
Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, USA

**Devanathan Thiruvenkatachari**
Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, USA

## Abstract

We prove that for every constant $c$ and $\varepsilon = (\log n)^{-c}$, there is no polynomial time algorithm that when given an instance of 3-LIN with $n$ variables where an $(1-\varepsilon)$-fraction of the clauses are satisfiable, finds an assignment that satisfies atleast $(\frac{1}{2}+\varepsilon)$-fraction of clauses unless $\mathbf{NP} \subseteq \mathbf{BPP}$. The previous best hardness using a *polynomial time* reduction achieves $\varepsilon = (\log \log n)^{-c}$, which is obtained by the Label Cover hardness of Moshkovitz and Raz [*J. ACM*, 57(5), 2010] followed by the reduction from Label Cover to 3-LIN of Håstad [*J. ACM*, 48(4):798–859, 2001].

Our main idea is to prove a hardness result for Label Cover similar to Moshkovitz and Raz where each projection has a *linear* structure. This linear structure of Label Cover allows us to use Hadamard codes instead of long codes, making the reduction more efficient. For the hardness of Linear Label Cover, we follow the work of Dinur and Harsha [*SIAM J. Comput.*, 42(6):2452–2486, 2013] that simplified the construction of Moshkovitz and Raz, and observe that running their reduction from a hardness of the problem LIN (of unbounded arity) instead of the more standard problem of solving quadratic equations ensures the linearity of the resultant Label Cover.

**2012 ACM Subject Classification** Theory of computation → Design and analysis of algorithms

**Keywords and phrases** probabilistically checkable proofs, PCP, composition, 3LIN, low soundness error

**Digital Object Identifier** 10.4230/LIPIcs.APPROX-RANDOM.2019.9

**Category** APPROX

**Funding** *Prahladh Harsha*: Supported in part by the DIMACS/Simons Collaboration in Cryptography through NSF grant #CNS-1523467 (while the author was visiting Rutgers University and DIMACS) and the Swarnajayanti Fellowship.
*Subhash Khot*: Supported by the NSF Award CCF-1422159, the Simons Collaboration on Algorithms and Geometry and the Simons Investigator Award.
*Euiwoong Lee*: Supported in part by the Simons Collaboration on Algorithms and Geometry.
*Devanathan Thiruvenkatachari*: Supported by same sources as Subhash Khot.

## 1 Introduction

In this paper, we study the 3-LIN problem. An instance of 3-LIN consists of a set of $n$ variables over $\mathbb{F}_2$ and a set of $m$ equations that contain at most three variables each, and the goal is to find an assignment to the $n$ variables that satisfies the most number of equations.[1] If the given set of linear equations admits an assignment that satisfies every equation, then one such assignment can be found in polynomial time by Gaussian elimination. However, the general problem of finding the most of number of equations is **NP**-hard when the instance does not admit a satisfying assignment, and a large amount of research has been done on the limit of polynomial time approximation algorithms.

Assigning random values to variables satisfies exactly half the equations in expectation, giving a $1/2$-approximation algorithm. Håstad and Venkatesh [7] achieved an approximation factor of $1/2 + 1/O(\sqrt{m})$, which was improved by Khot and Naor [10] to $1/2 + O(\sqrt{\log n/n})$.

From the hardness side, there are strong hardness results even when the instance is *almost-satisfiable*. For $1 \geq c > s > 0$, let Gap 3-LIN$(c, s)$ denote the problem of distinguishing whether the given instance of 3-LIN is at least $c$-satisfiable or at most $s$-satisfiable. Håstad's classic hardness results [6] show the following.

▶ **Theorem 1.1** ([6]). *The following hardness results for* Gap 3-LIN *hold.*

1. *For any constant $\varepsilon > 0$,* Gap 3-LIN$(1 - \varepsilon, 1/2 + \varepsilon)$ *is* **NP***-hard.*
2. *There exists a constant $c > 0$ such that for $\varepsilon = 1/(\log n)^c$, there is no polynomial time algorithm that solves* Gap 3-LIN$(1 - \varepsilon, 1/2 + \varepsilon)$ *unless* $\mathbf{NP} \subseteq \mathbf{DTIME}[n^{O(\log \log n)}]$.

Håstad's results are proved by giving the reduction from Label Cover to 3-LIN. Label Cover is a common starting point for hardness results, and we define the optimization problem below.

▶ **Definition 1.2** (Label Cover). *An instance of* Label Cover *contains a regular bipartite multi- graph $G = (A, B, E)$ and two finite sets $\Sigma_A$ and $\Sigma_B$, where $|\Sigma_A| \geq |\Sigma_B|$. Every vertex in $A$ is supposed to get a label in $\Sigma_A$, and every vertex in $B$ is supposed to get a label in $\Sigma_B$. For each edge $e \in E$ there is a projection $\pi_e : \Sigma_A \to \Sigma_B$. Given a labeling to the vertices of the graph, i.e., functions $\phi_A : A \to \Sigma_A$ and $\phi_B : B \to \Sigma_B$, an edge $e = (a, b) \in E$ is said to be "satisfied" if $\pi_e(\phi_A(a)) = \phi_B(b)$. For $1 \geq c > s > 1$,* Gap Label Cover$(c, s)$ *is the problem if distinguishing whether the given instance of* Label Cover *is at least $c$-satisfiable or at most $s$-satisfiable.*

Håstad's theorem can be stated in terms of reduction from Gap Label Cover$(1, \delta)$ as follows.

▶ **Theorem 1.3** ([6]). *For every $\varepsilon \in (0, 1)$ and positive integer $\ell$, there exists a $\delta = \text{poly}(\varepsilon)$ and a $\text{poly}(n, 2^\ell, 2^{1/\varepsilon})$-time reduction from $n$-sized instances of* Gap Label Cover$(1, \delta)$ *with label size $\ell$ to* Gap 3-LIN$(1 - \varepsilon, 1/2 + \varepsilon)$.

When [6] was published, the hardness of Label Cover was achieved by the PCP theorem [2, 1] and parallel repetition [13]. More precisely, Gap Label Cover$(1, \varepsilon)$ with label size $\text{poly}(1/\delta)$ was **NP**-hard under $\text{poly}(n^{\log 1/\delta})$-time reductions. The two results of Håstad stated in Theorem 1.1 follow from this hardness of Gap Label Cover and Theorem 1.3 by setting $\delta$ to be an arbitrarily small constant and $1/\log n$ respectively. Since

[1] This maximization version is also known as Max 3-LIN in the literature.

achieving a subconstant soundness for Label Cover by parallel repetition requires a superpolynomial blowup in the instance size, $\varepsilon > 0$ could not be taken to subconstant under *polynomial time* reductions. Later in a celebrated paper, Moshkovitz and Raz [12] gave an improved hardness of Label Cover that achieves sub-constant error under polynomial time reductions. Their main result can be stated as follows.

▶ **Theorem 1.4** ([12, Theorem 11]). *For every $n$, and every $\delta > 0$ (that can be any function of $n$),* 3-SAT *on inputs of size $n$ can be reduced to* Gap Label Cover$(1, \delta)$ *when* Label Cover *instance has $n^{1+o(1)} \cdot \mathrm{poly}(1/\delta)$ vertices and $|\Sigma_A| \leq \exp(\mathrm{poly}(1/\delta)), |\Sigma_B| \leq \mathrm{poly}(\log 1/\delta)$.*

A corollary of the above result, obtained by combining it with Håstad's reduction from Theorem 1.3, is that given a system of linear equations, it is **NP**-hard to distinguish between cases where $1 - o(1)$ fraction of equations are satisfied vs at most $1/2 + o(1)$ fraction are satisfied, where the $o(1)$ term is $1/(\log \log n)^{-\Omega(1)}$.

▶ **Theorem 1.5** ([12]). *There exists some constant $c > 0$ such that for $\varepsilon = 1/(\log \log n)^c$,* Gap 3-LIN$(1 - \varepsilon, 1/2 + \varepsilon)$ *is* **NP***-hard.*

Later, an improved parallel repetition by Dinur and Steurer [4] allowed $c$ to be an arbitrary constant.

The above route prove hardness of 3-LIN is restricted by the large size of the alphabet in the resulting Label Cover instance in Theorem 1.4. Quantitatively, the alphabet size is exponential in $\mathrm{poly}(1/\varepsilon)$. The fact that the long code in Håstad's reduction has size exponential in the alphabet size restricts $\varepsilon = 1/(\log \log n)^{O(1)}$.

Our main contribution for 3-LIN is to bring $\varepsilon$ in the above result down to $1/(\log n)^c$ for any constant $c$, while keeping the size of the reduced instance polynomial (albeit the reduction becomes randomized).

▶ **Theorem 1.6** (Main). *For any constant $c > 0$ and $\varepsilon = 1/(\log n)^c$, there is no polynomial time algorithm for* Gap 3-LIN$(1 - \varepsilon, 1/2 + \varepsilon)$ *unless* $\mathbf{NP} \subseteq \mathbf{BPP}$.

We get around the above alphabet barrier by starting with a reduction that would make the resulting Label Cover *linear*, and use Hadamard codes instead of long codes. Since the Hadamard code keeps the reduction size polynomial in the alphabet size, we can take $\varepsilon = 1/(\log n)^{\Omega(1)}$. A similar idea was previously used by Khot [8]. We define Linear Label Cover as follows.

▶ **Definition 1.7** (Linear Label Cover). *A* Linear Label Cover *is a special case of* Label Cover *where the alphabets are of the form $\Sigma_A = \mathbb{F}_2^a, \Sigma_B = \mathbb{F}_2^b$ where $a, b$ are natural numbers. Each projection $\pi : \mathbb{F}_2^a \to \mathbb{F}_2^b$ is* affine *in the sense that $\pi(x) = \alpha x + \beta$ for some $\alpha \in \mathbb{F}_2^{b \times a}, \beta \in \mathbb{F}_2^b$. For $1 \geq c > s > 0$, the* Gap Linear Label Cover$(c, s)$ *is defined similarly to* Gap Label Cover$(c, s)$.

We prove the following hardness result for Linear Label Cover, which may be of independent interest.

▶ **Theorem 1.8** (Hardness of Linear Label Cover). *For any constant $c > 0$, for $\delta = 1/(\log n)^c$, there is no polynomial time algorithm for* Gap Linear Label Cover$(1 - \delta, \delta)$ *unless* $\mathbf{NP} \subseteq \mathbf{BPP}$, *when* Label Cover *instance has* $\mathrm{poly}(n)$ *vertices and $|\Sigma_A| = \mathrm{poly}(n), |\Sigma_B| = \mathrm{polylog}(n)$.*

We remark that if the above theorem can be further strengthened to obtain $\delta = 1/n^c$ (i.e., a linear version of the Sliding Scale conjecture), then this leads to near-optimal hardness of 3-LIN (i.e, Gap 3-LIN$(1 - \varepsilon, 1/2 + \varepsilon)$ is hard for $\varepsilon = 1/\mathrm{poly}(n)$) [11].

## 1.1 Proof Ideas

Our main technical contribution is Theorem 1.8 for Linear Label Cover, essentially proving a linear analogue of the Moshkovitz-Raz PCP [12] followed by the Dinur-Steurer parallel repetition [4]. The proof is given through a long sequence of reductions. We split them in 3 major steps.

1. Interestingly, the starting point of our reduction is again the hardness of (not necessarily linear) Label Cover proved by Moshkovitz and Raz [12] augmented by Dinur and Steurer [4], proving **NP**-hardness of Gap Label Cover$(1, 1/\log^c n)$ for any $c > 0$, while keeping the reduction size and the alphabet size polynomial. In Section 2, we give a *randomized* reduction from this Label Cover to Gap LIN$(1 - 1/\log^c n, 0.9)$. This style of reduction appeared previous from Label Cover to Closest Vector Problem [9]. Note that the standard proof of the PCP theorem encodes 3-SAT (or Circuit SAT) by solving quadratic equations over $\mathbb{F}_2$, and this is essentially the only place that needs where nonlinearity occurs. Our hardness result for solving linear equations with completeness very close to (but not exactly) 1 allows us to follow previous PCP constructions that will ensure linearity of the Label Cover instance in the subsequent steps.
2. To prove the hardness of Linear Label Cover given the above hardness of LIN, we closely follow the steps of Dinur and Harsha [3], who gave a simpler and modular proof of [12]. The two basic building blocks in their proof are robust PCPs and decodable PCPs. Robust PCPs are PCPs where in the soundness case, for any proof and most random choices of the verifier, not only are the local views non-accepting, but they are also very far from any accepting string. It is indeed equivalent to Label Cover. Using our previous hardness for LIN as the starting point and following the standard robust PCP construction (e.g., low-degree extension and sum-check protocol), we can prove a polynomial time reduction to Linear Label Cover$(1 - 1/\log^c n, 1/\log^c n)$ for any $c > 1$, but the alphabet size will be always $\exp(\log^{c_0} n)$ for some $c_0 > 1$, which is superpolynomial.
3. The second building block, decodable PCP, is similar to robust PCP with the additional requirement that the prover is given a position $i$ in the original string and supposed to output the value of the $i$th position if the given proof is a honest encoding of a valid original string. The main idea of Dinur and Harsha [3] is to iteratively compose a robust PCP with a suitable decodable PCP, where the composed PCP is another robust PCP that consists of a decodable PCP for each constraint of the original robust PCP. This iteratively reduces the query complexity and the alphabet size of the robust PCP, which is related to the alphabet size of the equivalent Label Cover instance. This iterative composition is interleaved and preprocessed by technical operations that reduce the alphabet size of the robust PCP and make it regular.

   Once these two building blocks are linear, the operations of [3] can be used verbatim in our construction. Our main observation is that every step of this construction preserves (1) the robust completeness $1-\delta$ for some $\delta = 1/\mathrm{polylog}(n)$, and (2) the linearity, which were not issues in [3]. In Section 3, we introduce the basic building blocks and these operations, and show how they preserve robust completeness and linearity. These iterative operations will eventually reduce the alphabet size of the Linear Label Cover polynomial, proving Theorem 1.8.

After the hardness of Linear Label Cover is proved, we give a reduction from Linear Label Cover with the above parameters to 3-LIN with the required parameters. We do this by composing with the Hadamard Code to get a $(1 - \varepsilon)$ vs $(1/2 + \varepsilon)$ **NP**-hardness result for 3LIN. Similar PCP constructions based on Hadamard codes were presented in [8]. Details of this step can be found in Section 4.

## 2 Reduction to System of Linear Equations

In this section, we first prove the hardness of approximate solving linear equations over large fields, where each equation can involve as many variables as possible. It will serve as the starting point towards proving hardness of Linear Label Cover.

▶ **Theorem 2.1.** *For any constant $c > 0$, $\varepsilon = 1/(\log n)^c$, Gap LIN$(1 - 1/(\log n)^c, 0.9)$ is* **NP***-hard under polynomial time randomized reductions.*

**Proof.** The proof starts from the following hardness of Label Cover, which is obtained by combining the main result of Moshkovitz and Raz [12] with the parallel repetition of Dinur and Steurer [4].

▶ **Theorem 2.2** ([12, 4])**.** *For any constant $c > 0$, for $\delta = 1/(\log n)^c$, Gap Label Cover$(1, \delta)$ is* **NP***-hard when the* Label Cover *instance satisfies $|\Sigma_A|, |\Sigma_B| \leq |A| + |B|$.*

Let $G = (A, B, E)$, $\Sigma_A$, $\Sigma_B$, and $\{\pi_e\}_{e \in E}$ be an instance of Label Cover. We show a reduction to LIN over $\mathbb{F}_2$ where

- If all Label Cover edges are satisfiable, at least $(1 - \frac{1}{|\Sigma_A|})$ fraction of equations are satisfiable.
- If at most $\delta$ fraction of Label Cover edges are satisfiable, at most $(1 - \frac{1}{(\delta|\Sigma_A|)})$ fraction of equations are satisfiable.

For each vertex $v \in \Sigma_A \cup \Sigma_B$ and possible label $\ell$ on the Label Cover instance, we have a variable $x_{v,\ell}$ in the LIN instance. Let $n = |A||\Sigma_A| + |B||\Sigma_B| = \text{poly}(|A| + |B|)$ be the number of variables. Consider the following four kinds of equations. Recall that every arithmetic is performed over $\mathbb{F}_2$.

$$(1) \quad \sum_{\ell \in \Sigma_A} x_{v,\ell} = 1 \qquad \forall v \in A$$

$$(2) \quad \sum_{\ell \in \Sigma_B} x_{v,\ell} = 1 \qquad \forall v \in B$$

$$(3) \quad \sum_{r : \pi_{uv}(r) = \ell} x_{v,r} = x_{u,\ell} \qquad \forall (u,v) \in E, \forall \ell \in \Sigma_B$$

$$(4) \quad x_{v,\ell} = 0 \qquad \forall (v,\ell) \in A \times \Sigma_A$$

In our final LIN instance, we treat (1), (2), and (3) as *hard constraints* that need to be always satisfied, and find $x$ that always satisfies all hard constraints and as many constraints in (4) as possible. Also note that in (4), we only consider vertices in $A$.

This is equivalent to the usual LIN problem with hard constraints by *folding*. Formally, let $V$ be the set of assignments that satisfy (1), (2), and (3). If $V$ is empty, we can conclude that the Label Cover instance is unsatisfiable. Otherwise, there exist $c \in \mathbb{N}$ and linearly independent vectors $y_0, \ldots, y_c \in \mathbb{F}_2^{(A \times \Sigma_A) \cup (B \times \Sigma_B)}$ such that $V = \{y_0 + \sum_{i=1}^{c} y_i z_i : z_1, \ldots, z_c \in \mathbb{F}_2\}$. This gives an one-to-one correspondence between $\mathbb{F}_2^c$ and $V$, so we can treat $z_1, \ldots, z_c$ as the variables of LIN and write the fourth constraints $x_{v,\ell} = 0$ in terms of $z$, which gives an instance of LIN without hard constraints.

### Completeness

If the Label Cover instance is satisfiable, $x_{v,\ell} = 1$ if and only if $v$ is assigned with $\ell$ gives an assignment that satisfies (1), (2), and (3), and violates one equation in (4) for each $v \in A$.

### Soundness

Let $x$ be an assignment that satisfies (1), (2), and (3). For $v \in A \cup B$, let $L_v := \{\ell : x_{v,\ell} = 1\}$. Since (1) and (2) require $\sum_\ell x_{v,\ell} = 1$ for every $v \in A \cup B$, $L_v$ is not empty for every $v$.

Consider the randomized strategy for Label Cover where each $v \in A \cup B$ is assigned with a uniform random label from $L_v$ independently. For $(u, v) \in E$ with $u \in A, v \in B$, by (3), $x_{v,\ell} = 1$ for some $\ell \in \Sigma_B$ implies that there exists $r \in \Sigma_A$ with $\pi_{uv}(r) = \ell$ such that $x_{u,r} = 1$. This implies $(u, v)$ is satisfied with probability at least $\frac{1}{|L_u|}$ by the randomized strategy. Then the expected fraction of the Label Cover constraints satisfied by the strategy is at least

$$\mathbf{E}_{u\in A}\left[\frac{1}{|L_u|}\right] \geqslant \frac{1}{\mathbf{E}_{u\in A}[|L_u|]}.$$

Therefore, if at most $\delta$ fraction of Label Cover constraints are simultaneously satisfiable, we can conclude that

$$\delta \geqslant \frac{1}{\mathbf{E}_{u\in A}[|L_u|]} \quad \Leftrightarrow \quad \mathbf{E}_{u\in A}[|L_u|] \geqslant \frac{1}{\delta}.$$

So in total, at least $\frac{1}{(\delta|\Sigma_A|)}$ fraction of equations are violated.

### Gap Amplification

We have a hardness of LIN over $\mathbb{F}_2$ where the completeness value is at least $1 - \frac{1}{|\Sigma_A|}$ and the soundness value is at most $1 - \frac{1}{(\delta|\Sigma_A|)}$. Consider a new system of linear equations where we sample $m$ linear equations independently, where each new equation randomly chooses $\delta \cdot |\Sigma_A|$ old equations and takes a random linear combination of them. In the completeness case, at least an $(1 - O(\delta))$ fraction of new equations can be satisfied by a good assignment to old equations.

In the soundness case, fix an assignment to $n$ possible variables. (There are $2^n$ of them.) It satisfies at most an $1 - \frac{1}{(\delta|\Sigma_A|)}$ fraction of old equations. Note that if a new equation chooses an old equation not satisfied by the assignment, it is satisfied with probability exactly $1/2$. Therefore, the expected number of new equations satisfied by this fixed assignment is at most

$$m \cdot \left(\left(1 - \frac{1}{(\delta|\Sigma_A|)}\right)^{\delta\cdot|\Sigma_A|} + \frac{1}{2}\right) \leq m \cdot \left(\frac{1}{e} + \frac{1}{2}\right) \leq 0.87m.$$

For a given $c \in \mathbb{N}$, let $\delta = 1/\log^c n$. By taking sufficiently large $m = O(n)$, we can apply the Chernoff and union bound to conclude that no assignment satisfies more than a 0.9 fraction of new equations. So we reduce from Label Cover to Gap LIN$(1 - O(\delta), 0.9)$, which finishes the proof. ◀

We remark that the sampling performed above is the only step in our reduction involving randomization.

## 3 Reduction to Linear Label Cover

In this section, we show for any $c > 0$, unless **NP** $\subseteq$ **BPP**, there is no polynomial time algorithm for Gap Linear Label Cover$(1 - \varepsilon, \varepsilon)$ with $\varepsilon = 1/(\log n)^c$, proving Theorem 1.8.

The construction we employ is almost identical to that of Dinur and Harsha [3], except that the basic building blocks (robust PCP and decodable PCP) try to prove (almost) satisfiability of linear equations instead of standard quadratic equations. They are introduced in Sections 3.1 and 3.2.

After constructing the building blocks, the result of [3] is proved by iterative composition of them followed by technical steps including alphabet and degree reduction. Our main observation in this part is that each of the steps in the construction preserves *linearity* so that the final Label Cover instance produced also has a liear structure. We present them in Section 3.3 and Section 3.4. Finally, Section 3.5 shows how to combine all these steps to prove Theorem 1.8.

## 3.1 Robust PCPs

In this subsection, we define robust PCPs. For two strings $x, y$ of the same length $n$, let $\mathsf{agr}(x, y)$ denote the relative agreement of the strings $x, y$, defined as

$$\mathsf{agr}(x, y) := \Pr_{i \in [n]}[x_i = y_i]$$

If $S$ is a set of strings, $\mathsf{agr}(x, S)$ is defined as $\max_{y \in S}\{\mathsf{agr}(x, y)\}$.

▶ **Definition 3.1** (Robust PCPs). *For functions $\mathsf{r}, \mathsf{q}, \mathsf{m}, \mathsf{a}, \mathsf{s} : \mathbb{N} \to \mathbb{N}$ and $c, \delta : \mathbb{N} \to [0, 1]$, a verifier $V$ is a* robust probabilistically checkable proof (robust PCP) system *for a promise problem $L = (L_{\text{YES}}, L_{\text{NO}})$ with randomness complexity $\mathsf{r}$, query complexity $\mathsf{q}$, proof length $\mathsf{m}$, alphabet size $\mathsf{a}$, robust completeness $c$, and robust soundness error $\delta$ if $V$ is a probabilistic polynomial-time algorithm that behaves as follows: On input $x$ of length $n$ and oracle access to a proof string $\pi \in \Sigma^{\mathsf{m}(n)}$ over the (proof) alphabet $\Sigma$ where $|\Sigma| = \mathsf{a}(n)$, $V$ reads the input $x$, tosses at most $\mathsf{r} = \mathsf{r}(n)$ random coins, and generates a sequence of locations $I = (i_1, \ldots, i_q) \in [\mathsf{m}]^{\mathsf{q}(n)}$ and a predicate $f : \Sigma^{\mathsf{q}} \to \{0, 1\}$, which satisfy the following properties.*

***Robust Completeness.*** *If $x \in L_{\text{YES}}$ then there exists $\pi$ such that*

$$\mathop{\boldsymbol{E}}_{(I,f)}[\mathsf{agr}(\pi_I, f^{-1}(1))] \geqslant c. \tag{1}$$

***Robust Soundness.*** *If $x \in L_{\text{NO}}$ then for every $\pi$,*

$$\mathop{\boldsymbol{E}}_{(I,f)}[\mathsf{agr}(\pi_I, f^{-1}(1))] \leqslant \delta, \tag{2}$$

*where the distribution over $(I, f)$ is determined by $x$ and the random coins of $V$.*

*We say that $V$ is* linear *if $\Sigma = \mathbb{F}_2^b$ for some $b$ and for every $f$, the accepting sets of the predicate $f$, i.e., $f^{-1}(1)$, forms an affine subspace of $\Sigma^{\mathsf{q}} = \mathbb{F}_2^{b\mathsf{q}}$ over the field $\mathbb{F}_2$.*

Robust completeness and soundness must be contrasted with (regular) completeness and soundness of standard PCP verifiers in which the expression for completeness and soundness given in (1) and (2) respectively are replaced as follows:

$$\text{Completeness: } \Pr_{I,f}[f(\pi_I) = 1] \geqslant c,$$

$$\text{Soundness: } \Pr_{I,f}[f(\pi_I) = 1] \leqslant \delta.$$

In fact, this is the only difference between the above definition and the standard definition of a PCP system. The robust soundness states that not only does the local view violate the local predicate $f$, but in fact has very little agreement with any of the satisfying assignments of $f$ (and thus is a strengthening of standard robustness). Robust completeness on the other hand is a weakening of standard completeness.

Another crucial aspect of robust PCP is its equivalence to Label Cover. Namely, existence of robust PCP for $L$ with parameters $\mathsf{r}, \mathsf{q}, \mathsf{m}, \mathsf{a}, \mathsf{s}, c, \delta$ is equivalent to existence of a reduction from $L$ to Gap Label Cover$(c, \delta)$ where $|A| = 2^{\mathsf{r}}, |B| = \mathsf{m}, |\Sigma_A| \leq \mathsf{a}^{\mathsf{q}}, |\Sigma_B| = \mathsf{a}$ and each $v \in A$ has degree $\mathsf{q}$. See Lemma 2.5 of [3]. Also note that the definition of linearity is equivalent in robust PCP and Label Cover.

▶ **Theorem 3.2** (Robust PCP, Analog of [3, Theorem 6.4]). *There exist constants $b_1, b_2 > 0$, $c_0 > 1$ such that for any $c > c_0$ and $\varepsilon = 1/\log^c n$, Gap LIN$(1-\varepsilon, 0.9)$ with $n$ variables has a linear robust verifier with robust completeness $1-\varepsilon$, robust soundness error $\varepsilon$, query complexity $1/\varepsilon^{b_1}$, proof length* poly$(n)$*, randomness complexity $O(\log n)$, and proof alphabet size at most $1/\varepsilon^{b_2}$.*

*Equivalently, there is a (deterministic) polynomial time reduction from* Gap LIN$(1-\varepsilon, 0.9)$ *to* Gap Linear Label Cover$(1-\varepsilon, \varepsilon)$*, where the* Label Cover *instance has* poly$(n)$ *veritces, $|\Sigma_A| \leq \exp(1/\varepsilon^{b_1} \log(1/\varepsilon^{b_2}))$, $|\Sigma_B| \leq 1/\varepsilon^{b_2}$, and each $v \in A$ has degree $1/\varepsilon^{b_1}$.*

The proof of this theorem is identical to that of [3, Theorem 6.4] and omitted here. The only difference is Gap LIN$(1-\varepsilon, 0.9)$ with $1/\varepsilon = \log^{O(c)} n$ instead of standard quadratic equations when performing the low degree-extension and the sum-check protocol. The theorem follows by observing that all the operations are linear and hence the final predicate is also linear. The completeness of the robust PCP is dictated by the completeness value in Theorem 2.1.

Combining this reduction with the randomized reduction from Theorem 2.1, we obtain the following theorem (which is a more formal version of Theorem 1.8).

▶ **Theorem 3.3** (Hardness of Linear Label Cover). *There exist constants $b_1, b_2 > 0$, $c_0 > 1$ such that for any $c > c_0$ and $\varepsilon = 1/\log^c n$, unless* **NP** $\subseteq$ **BPP**, *there is no polynomial time algorithm for* Gap Linear Label Cover$(1-\varepsilon, \varepsilon)$ *where the* Label Cover *instance has* poly$(n)$ *veritces, $|\Sigma_A| \leq \exp(1/\varepsilon^{b_1} \log(1/\varepsilon^{b_2}))$, $|\Sigma_B| \leq 1/\varepsilon^{b_2}$, and each $v \in A$ has degree $1/\varepsilon^{b_1}$.*

## 3.2 Decodable PCPs

We now discuss the decodable PCP (dPCP), which differs from a PCP in that it has a decoder as opposed to a verifier. A *decoder* is similar to a verifier in that it checks whether a string is in the given language or not by probabilistically checking a small number of positions in the proof, but it is additionally supposed to return the $i$th position of the original string for given $i$.

For $\Sigma = \mathbb{F}_2^a$ for some $a \in \mathbb{N}$, let $\mathrm{LIN}_\Sigma$ denote the problem of solving linear equations where an instance consists of $k$ variables that can have a value from $\Sigma$, and a system of linear equations $C$ on $k \cdot a$ variables over $\mathbb{F}_2$ canonically represented by the $k$ variables over $\Sigma$. It is equivalent to LIN over $\mathbb{F}_2$ on $k \cdot a$ variables, except that we consider each block of $a$ variables as one variable that can take a value from $\Sigma$. We define a decoder for $\mathrm{LIN}_\Sigma$ below.

▶ **Definition 3.4** (Decoder for $\mathrm{LIN}_\Sigma$). *Let $\Sigma = \mathbb{F}_2^a$ and $\sigma = \mathbb{F}_2^b$ for some $a$ and $b$. A decoder for $\mathrm{LIN}_\Sigma$ over a proof alphabet $\sigma$ with parameters $\mathsf{m}, \mathsf{q}, \mathsf{r} : \mathbb{N} \to \mathbb{N}$ is a probabilistic polynomial-time algorithm $\mathcal{D}$. It is given a system of linear equations $C$ on $n$ variables over $\Sigma$, and an index $j \in [n]$ as input, and oracle access to a proof $\pi$ of length $\mathsf{m}(n)$ over proof alphabet $\sigma$. It tosses $\mathsf{r} = \mathsf{r}(n)$ random coins and generates (1) a sequence of $\mathsf{q} = \mathsf{q}(n)$ locations $I = (i_1, \ldots, i_{\mathsf{q}})$ and (2) a (local decoding) function $f : \sigma^{\mathsf{q}} \to \Sigma \cup \{\perp\}$. $\mathcal{D}$ is called linear if for every $f$, $P := f^{-1}(\Sigma)$ is an affine space of $\sigma^{\mathsf{q}} = (\mathbb{F}_2^{\mathsf{q}b})$ and $f : P \to \Sigma$ is an affine function over the base field $\mathbb{F}_2$.*

Now we define a dPCP for $\mathrm{LIN}_\Sigma$. The dPCP in [3] is defined for Circuit SAT, whereas ours is for $\mathrm{LIN}_\Sigma$. Note that unlike in [3], the dPCP we will construct does not imply any computational hardness, because it only proves whether the given system of linear equations is perfectly satisfiable or not, which is a computationally easy problem. The key point is it proves the system is satisfiable using a proof which is in some sense "locally decodable". The dPCP will then be composed with the previous linear robust PCP, which is a system of linear equations with *imperfect completeness*, to reduce the query complexity.

▶ **Definition 3.5** (Decodable PCPs for $\mathrm{LIN}_\Sigma$)**.** *For functions $\delta : \mathbb{N} \to [0,1]$ and $\mathsf{L} : \mathbb{N} \to \mathbb{N}$, we say that a PCP decoder $\mathcal{D}$ is a* decodable probabilistically checkable proof (dPCP) system *for $\mathrm{LIN}_\Sigma$ with perfect completeness, soundness $\delta$ and list size $\mathsf{L}$ if the following completeness and soundness properties hold for every system of linear equations $C$ on $n$ variables over $\Sigma$.*

***Completeness.*** *For any $y \in \Sigma^n$ that satisfies every equation in $C$, there exists a proof $\pi \in \sigma^{\mathsf{m}}$, also called a decodable PCP, such that*

$$\Pr_{j,I,f}[f(\pi_I) = y_j] = 1,$$

*where $j \in [n]$ is chosen uniformly at random and $I, f$ are distributed according to $C, j$, and the verifier's random coins.*

***Soundness.*** *For any $\pi \in \sigma^{\mathsf{m}}$, there is a list of $0 \leqslant \ell \leqslant \mathsf{L}$ strings $y^1, \ldots, y^\ell$, where each $y^i$ satisfies all equations in $C$, such that*

$$\Pr_{j,I,f}[f(\pi_I) \notin \{\perp, y_j^1, \ldots, y_j^\ell\}] \leqslant \delta.$$

***Robust soundness.*** *We say that $\mathcal{D}$ is a* robust dPCP system *for $\mathrm{LIN}_\Sigma$ with robust soundness error $\delta$, if the soundness criterion above can be strengthened to the following robust soundness criterion,*

$$\boldsymbol{E}_{j,I,f}[\mathsf{agr}(\pi_I, \mathrm{BAD}(f))] \leqslant \delta,$$

*where*

$$\mathrm{BAD}(f) := \{w \in \sigma^{\mathsf{q}} : f(w) \notin \{\perp, y_j^1, \ldots, y_j^\ell\}\}.$$

The dPCP result we use is the following.

▶ **Theorem 3.6** (dPCP, Analog of [3, Theorem 6.5])**.** *There exist constants $\alpha, \gamma > 0$ such that for every $\delta \geqslant n^{-\alpha}$ and input alphabet size $\Sigma$ of size at most $n^\gamma$, $\mathrm{LIN}_\Sigma$ has a linear robust decodable PCP system with perfect completeness, robust soundness error $\delta > 0$ and list size $\mathsf{L} \leqslant 2/\delta$, query complexity $n^{1/8}$, proof alphabet $\sigma$ of size $n^\gamma$, proof length $\mathrm{poly}(n)$, and randomness complexity $O(\log n)$.*

The proof of this theorem is identical to that of [3, Theorem 6.5], except that the initial starting point is $\mathrm{LIN}_\Sigma$ instead of Circuit $\mathrm{SAT}_\Sigma$. Since the starting point is linear and all transformations are linear, the final object is also linear. The perfect completeness is also maintained. As mentioned before, the dPCP constructed here does not imply any computational hardness unlike in [3].

## 3.3 Composition

After having building blocks, Dinur and Harsha [3] show how to compose those blocks iteratively to reduce the query complexity and the alphabet size. Each composition involves several other operations including alphabet and degree reductions. While the soundness analyses for them are already proved in [3], we show that all of their operations preserve linearity and robust completeness.

### Efficient Composition ([3, Theorem 4.2])

In the composition, given a regular robust linear PCP verifier $V$ and a robust linear PCP decoder $\mathcal{D}$, the composed verifier $V'$ expects a decodable PCP for each constraint of $V$. Recall that the linearity of $V$ is equivalent to the fact that each constraint of $V$ is a system of linear equations over $\mathbb{F}_2$, which is exactly what $\mathcal{D}$ expects. An informal description of the composed verifier is as follows:

1. Randomly choose a location $i$ of the proof for $V$. Let $C_1, \ldots, C_D$ be the constraints of $V$ containing the location.
2. Using a $(\varepsilon, \varepsilon^2)$-sampler $([D], [D], E)$ and a random $s \in [D]$, choose a subset $S \subseteq \{1, \ldots, D\}$ and run the inner PCP decoder $\mathcal{D}$ for each $C_j$ with $j \in S$ to decode the $i$th symbol in the original proof.
3. Accept if all the values returned by the PCP decoders are the same.

For the second step above, we use $(\varepsilon, \varepsilon^2)$-samplers given in [5]. Theorem 4.2 of [3] shows the soundness of the composed verifier $V'$, yielding Table 1 below (Table 4.2 in [3]).

**Table 1** Parameters for Composition.

| | $V$ | $\mathcal{D}$ | $V'$ |
|---|---|---|---|
| proof alphabet | $\Sigma$ | $\sigma$ | $\sigma$ |
| randomness complexity | $\mathsf{R}$ | $\mathsf{r}$ | $\log M + \mathsf{r} + \log D$ |
| query complexity | $Q$ | $q$ | $4/\varepsilon^4 \cdot q$ |
| proof degree | $D$ | $d$ | $d$ |
| proof length | $M$ | $\mathsf{m}$ | $2^{\mathsf{R}} \cdot \mathsf{m}$ |
| robust soundness error | $\Delta$ | $\delta$ | $\Delta\mathsf{L} + 4\mathsf{L}\varepsilon + \delta$ |
| list size | - | $\mathsf{L}$ | - |

We check this composition preserves robust completeness and linearity.

- Linearity: Linearity (over $\mathbb{F}_2$) is preserved if both $V$ and $\mathcal{D}$ are linear, since the only additional check we perform is to check whether the returned values are equal.
- Robust completeness: Suppose that there exists a proof $\Pi$ for $V$ that achieves the robust completeness of at least $1 - \xi$. Recall that the composed verifier expects, for each constraint of the outer PCP, a satisfying assignment encoded by the inner dPCP. The proof for the composed verifier is the concatenation of all these encodings. Consider the proof to the composed verifier constructed by the honest encoding of the assignment that achieves the robust completeness for the outer PCP verifier. We will show that this proof achieves robust completeness $1 - \xi$.
  Let $i$ be a proof location in the outer PCP and $C_1, \ldots, C_D$ be the constraints involving $i$. Furthermore, let $\xi_i$ be the fraction of these constraints violated by the proof. Since $\Pi$ is at least $(1 - \xi)$-robustly complete, we have $\mathbf{E}_i[\xi] \leq \xi$. For each sample $s$ chosen in the sampler, let $\xi_{i,s}$ be the fraction of constraints in $S$ (chosen by sampler) that are violated. By regularity of sampler, we have $\mathbf{E}_s[\xi_{i,s}] \leq \xi_i$.
  A local view of the composed verifier (corresponding to $i, s$ and the inner dPCP randomness) comprises of the concatenation of the local views of the dPCP encodings corresponding to the constraints in $S$. Since the the inner dPCP has perfect completeness we have the following. Whenever the constraint is satisfied, the corresponding inner dPCP's encodings satisfies all constraints while we have no guarantee when the constraint is not satisfied. Since for each $(i, s)$, the fraction of violated constraints is $\xi_{i,s}$, we have

that at least $(1-\xi_{i,s})$-fraction of the local inner views corresponding to $(i,s)$ are satisfying and furthermore they all decode to the same $\Pi(i)$. Hence, the local view of the composed verifier corresponding to $(i,s)$ is at least $(1-\xi_{i,s})$-close to a satisfying view. Hence, the robust completeness of this honest proof is at least $\mathbf{E}_{i,s}[1-\xi_{i,s}] \geq 1-\xi$.

## 3.4 Label Cover Operations

After the composition, the alphabet reduction step is applied to ensure that the alphabet size is polynomial in the query complexity and the inverse of the soundness. Also, since the basic robust PCP given in Theorem 3.2 is not necessarily regular, we also need to show how to make the initial robust PCP regular. This subsection introduces various such operations and explains why they preserve robust completeness and linearity.

### Degree Reduction ([3, Theorem 5.1])

Given an instance of Label Cover $G=(A,B,E)$, the degree reduction makes the instance right-regular by appropriately duplicating right vertices and each edge exactly the same number of times. Theorem 5.1 of [3] ensures that by increasing robust soundness by $4\mu$ additively, we can ensure that the right degree is $4/\mu^4$ for all right vertices. We check that this operation preserves linearity and robust completeness.

- Linearity: Linearity is obviously preserved, because there is no change in the constraint.
- Robust completeness: Since each edge is duplicated the same number of times, robust completeness does not decrease.

### Alphabet Reduction ([3, Theorem 5.5])

Given an instance of Label Cover $G=(A,B,E)$ where $\Sigma_A$ and $\Sigma_B$ are the alphabet set of the left (bigger) side and the right (smaller) side respectively, the alphabet reduction replaces $\Sigma_B$ by a smaller set $\sigma$ by finding a suitable linear code $C:\Sigma_B \to \sigma^k$ and replacing each vertex $b \in B$ by $k$ vertices $b_1,\ldots,b_k$. Then assigning $x \in \Sigma_B$ to $b$ corresponds to assigning $(C(x))_i$ to $b_1,\ldots,b_i$. Theorem 5.5 of [3] ensures that if $C$ has a relative distance $1-\eta^3$, this operation increases robust soundness by at most $3\eta$ additively. We check that this operation preserves linearity and robust completeness.

- Linearity: Linearity over $\mathbb{F}_2$ is preserved if the code $C:\Sigma_B \to \sigma^k$ is linear with $\sigma = \mathbb{F}_{2^a}$ as the base field for some $a \in \mathbb{N}$. The code used in Remark 5.4 of [3] is already linear.
- Robust completeness: If an edge $(a,b)$ of the original Label Cover instance is preserved and the new instance follows the honest encoding, all $k$ edges of the new instance corresponding to $(a,b)$ will be satisfied. Therefore, robust completeness cannot decrease.

### Flip Sides ([3, Section 5.3])

Given an instance of Label Cover $G=(A,B,E)$ where each right vertex $b \in B$ has degree $d$, the flip side is achieved by flipping $A$ and $B$, and assigning each $v \in B$ a label from $\Sigma_A^d$, which is supposed to denote the assignments to its neighbors in the original instance. If $v \in B$ has $u_1,\ldots,u_d \in A$ as neighbors, $(v,u_i)$ in the new instance is satisfied (i) if the label $(a_1,\ldots,a_d) \in \Sigma_A^d$ for $v$ has $b \in \Sigma_B$ such that the label pair $(a_i,b)$ satisfies the edge $(u_i,v)$ in the old instance, and (ii) if $a_i$ is equal to the label assigned to $u_i$. This obviously does not change the robust soundness. We check that it also preserves linearity and robust completeness.

**Table 2** Sequence of steps to regularize the Label Cover instance. * denotes irregular instances where the number denotes the average degree.

| Label Cover (Robust PCPs) | $I$ | Degree Red. ($\to d$) | Flip | Degree Red. ($\to d$) | Alphabet Red. ($\to \sigma$) |
|---|---|---|---|---|---|
| # left vertices (randomness) | $n$ | $n$ | $mD_B$ | $mD_B$ | $mD_B$ |
| # right vertices (proof length) | $m$ | $mD_B$ | $n$ | $nD_Ad$ | $nD_Adk$ |
| left degree (query complexity) | $D_A^*$ | $dD_A^*$ | $d$ | $d^2$ | $d^2k$ |
| right degree (proof degree) | $D_B^*$ | $d$ | $D_Ad^*$ | $d$ | $d$ |
| left alphabet (# accepting conf.) | $\Sigma_A$ | $\Sigma_A$ | $\Sigma_A^d$ | $\Sigma_A^d$ | $\Sigma_A^d$ |
| right alphabet (proof alphabet) | $\Sigma_B$ | $\Sigma_B$ | $\Sigma_A$ | $\Sigma_A$ | $\sigma$ |
| soundness error (rob. soundness error) | $\delta$ | $\delta + 4\mu$ | $\delta + 4\mu$ | $\delta + 8\mu$ | $\delta + 8\mu + 3\eta$ |
| rob. completeness (rob. completeness) | $1 - \xi$ | $1 - \xi$ | $1 - \xi$ | $1 - \xi$ | $1 - \xi$ |

- Linearity: Linearity is preserved, because for each $v \in B$, the set of $(a_1, ..., a_d)$ satisfying (i) above is an affine subspace of $(\Sigma_A)^d$, and the new constraint is merely a projection.
- Robust completeness: Cannot decrease since if $v \in B$ was assigned $b \in \Sigma_B$ in the original instance, it can be assigned $(a_1, \ldots, a_d) \in \Sigma_A$ such that (i) $\pi_{(u_i,v)}(a_i) = b$, and (ii) $a_i$ was assigned to $u_i$ if $(u_i, v)$ was satisfied in the original instance.

We use a combination of the above 3 operations to get a regular Label Cover instance, as shown below.

Given an $\varepsilon > 0$, by using $(O(\varepsilon), O(\varepsilon^2))$-samplers in the composition and doing the above operations with $\eta = O(\varepsilon)$, $d = O(1/\varepsilon^4)$, distance $1-O(\varepsilon^3)$, $|\sigma| = O(1/\varepsilon^6), k = O(1/\varepsilon^6)\cdot|\Sigma'| \leq O(1/\varepsilon^6) \cdot q|\Sigma|$, we can deduce the following lemma.

▶ **Lemma 3.7** ([3, Lemma 5.7]). *For all $\varepsilon : \mathbb{N} \to [0, 1]$, suppose $L$ has a robust linear PCP verifier $V$ with randomness complexity $r$, query complexity $q$, proof length $m$, average proof degree $D_B$, robust completeness $c$, robust soundness error $\delta$ over a proof alphabet $\Sigma$. Then $L$ has a regular reduced linear robust PCP verifier, which we shall denote by* $\mathsf{regular}_\varepsilon(V)$ *with*

- *randomness complexity* $\log m + \log D_B$,
- *query complexity* $O(q \log |\Sigma|/\varepsilon^{14})$,

- *proof length* $O(q^2 2^r \log|\Sigma|/\varepsilon^{10})$,
- *proof degree* $O(1/\varepsilon^4)$,
- *proof alphabet* $\sigma$ *of size at most* $O(1/\varepsilon^6)$,
- *robust completeness* $c$,
- *and robust soundness* $\delta + \varepsilon$.

## 3.5 Putting things together

Finally we prove Theorem 1.8 on the hardness of Linear Label Cover. Let $c > 0$ be an arbitrary constant. Let $\mathcal{D}$ be the PCP decoder from Theorem 3.6 and $\mathcal{V}$ be the robust PCP from Theorem 3.2 with robust completeness $1-\delta$ with $\delta = \log^c n$, robust soundness error $\varepsilon = 1/\log^{c_0} n$ for some $c_0 > 1$, query complexity $1/\varepsilon^{O(1)}$, randomness complexity $O(\log n)$ and proof length $\mathrm{poly}(n)$.

▶ **Lemma 3.8** ([3, Lemma 6.6]). *Let* $\mathcal{D}$, $\mathcal{V}$, $\varepsilon, \delta$ *be as defined above and set* $\varepsilon_i = (\varepsilon)^{1/3^i}$. *There exist constants* $c_0$, $c_1$, $c_3 > 0$ *such that for every* $i \geqslant 0$ *as long as* $\varepsilon_i < c_0$, *the following holds.* Gap $\mathrm{LIN}(1-\delta, 0.9)$ *has a regular linear robust PCP verifier* $V_i$ *with query complexity* $1/\varepsilon_i^{c_1}$, *robust completeness* $1-\delta$, *robust soundness error* $2\varepsilon_i$, *proof alphabet* $\Sigma_i$ *of size* ${c_3/\varepsilon_i}^6$, *randomness complexity* $O(\log n)$ *and proof length* $\mathrm{poly}(n)$.

**Proof.** The proof is similar to [3], and is a sequence of compositions. We start with the regularized robust verifier given by applying the sequence of steps given in Section 3.4 to the robust PCP verifier given in Theorem 3.2. In each subsequent step, we compose the robust verifier obtained in the previous step with a dPCP, and apply the alphabet reduction (Theorem 5.5 of [3]) to reduce the size of the alphabet to $c_3/\varepsilon_{i+1}^6$. All the parameters remain the same as in [3], and we only need to focus on the two additional properties we need, linearity and robust completeness.

Recall that a PCP with robust completeness $1-\delta$, when composed with a dPCP with perfect completeness, yields a composed PCP with robust completeness $1-\delta$. In each step the inner PCP decoder has perfect completeness, therefore the robust completeness of the composed PCP is preserved. Recall that the alphabet reduction step also doesn't affect the perfect completeness.

Linearity is also preserved because all basic components are linear and all steps (e.g., composition, alphabet reduction, and regularization) preserve linearity as previously discussed. ◀

The above lemma shows that we can iteratively reduce the query complexity until some absolute constant while maintaining the soundness and the alphabet size polynomial in the query complexity.(And the total size of the instance always remains polynomial in $n$.) Only a constant number of iterations is needed until $(\text{proof alphabet size})^{(\text{query complexity})}$, an upper bound on the size of alphabet in the equivalent Label Cover instance, becomes polynomial in $n$. This proves our main Theorem 1.8 for Linear Label Cover.

**Proof of Theorem 1.8.** Set $i$ from Lemma 3.8 so that

$$(\text{proof alphabet size})^{(\text{query complexity})} = (c_3/\varepsilon_i^6)^{1/\varepsilon_i^{c_1}} = \exp\left(\frac{1}{\varepsilon_i^{c_1}} \cdot \log\left(\frac{c_3}{\varepsilon_i^6}\right)\right) \leq \mathrm{poly}(n).$$

This ensures that $\varepsilon_i = 1/\log^{c_4} n$ for some $c_4 > 0$. Using the equivalence between Label Cover and robust PCP, we have a hardness of Label Cover where the number of vertices and the size of label are bounded by $\mathrm{poly}(n)$, and the completeness is at least $1 - 1/\log^c n$, the soundness is $1/\log^{c_4} n$. Applying the parallel repetition of [4] $O(c/c_4)$ times to reduce the soundness to $1/\log^c n$ finishes the proof. ◀

## 4 Reduction from Linear Label Cover to 3LIN

In this section, we prove our main Theorem 1.6 for 3-LIN. Recall that Theorem 3.3 shows a randomized polynomial reduction from 3-SAT to Gap Linear Label Cover$(1 - \log^c n, \log^c n)$ for any constant $c > 0$, where the number of vertices as well as the number of labels are bounded by a polynomial. Therefore, the following theorem finishes the proof of Theorem 1.6. The main idea is to use Hadamard codes instead of long codes using the fact that the Label Cover instance is linear. A similar argument was used in [8].

▶ **Lemma 4.1.** *There is a polynomial time reduction from* Gap Linear Label Cover$(1 - \delta, s)$ *to* Gap 3-LIN$(1 - \delta, 1/2 + \sqrt{s}/2)$*, where the size of the* 3-LIN *instance is polynomial in the number of vertices and the size of label in the* Label Cover *instance.*

**Proof.** Let $G = (A, B, E)$, $\Sigma_A$, $\Sigma_B$, $\{\pi_e\}_{e \in E}$ be an instance of Gap Linear Label Cover $(1 - \delta, s)$. Moreover, since the label cover is linear, let the labels to left hand side vertices come from $\mathbb{F}_2^\ell$ and the right hand side vertices from $\mathbb{F}_2^r$, and the mapping on each edge is an affine mapping. Our reduction is described by the following test.

### Test

- Consider an edge $(u, v)$. The labels $x \in \mathbb{F}_2^\ell, y \in \mathbb{F}_2^r$ corresponding to the vertices have to satisfy $x = Ay + b$.
- From the proof, we randomly sample the Hadamard code of $x$ at location $\alpha$, and that of $y$ at locations $\beta$ and $\beta + \gamma$, where $\gamma = A^T \cdot \alpha$.
- Check if $\langle \alpha, x \rangle + \langle \beta, y \rangle + \langle \beta + \gamma, y \rangle = \langle \alpha, b \rangle$

### Completeness

In the completeness case, if the labels $x, y$ satisfy the edge in the Linear Label Cover, then we can see that the test will pass.

$$\begin{aligned}
&\langle \alpha, x \rangle + \langle \beta, y \rangle + \langle \beta + \gamma, y \rangle \\
=&\langle \alpha, Ay \rangle + \langle \alpha, b \rangle + \langle \beta, y \rangle + \langle \beta + \gamma, y \rangle \\
=&\langle \alpha, Ay \rangle + \langle \alpha, b \rangle + \langle A^T \alpha, y \rangle \\
=&\langle \alpha, b \rangle
\end{aligned}$$

Therefore, if $1 - \delta$ edges are satisfiable in the linear Label Cover, at least $1 - \delta$ fraction of 3LIN constraints are satisfied.

### Soundness

Consider the case where at most $s$ fraction of edges can be satisfied for any labeling in the Linear Label Cover. Let the Hadamard code encoding function for the left vertices be $L$ and right vertices be $R$. Consider their Fourier transforms,

$$L(\alpha) = \sum_x \hat{L}(x)\chi_x(\alpha)$$

$$R(\beta) = \sum_y \hat{R}(y)\chi_y(\beta)$$

Let's fix an edge, and analyze the probability that the test will accept. We switch to a -1,+1 notation for convenience.

$$\begin{aligned}\Pr[\text{Test accepts}] &= \Pr_{\alpha,\beta}[\langle\alpha,x\rangle + \langle\beta,y\rangle + \langle\beta + A^T\alpha, y\rangle + \langle\alpha,b\rangle = 0] \\ &= \Pr_{\alpha,\beta}[(-1)^{\langle\alpha,x\rangle+\langle\beta,y\rangle+\langle\beta+A^T\alpha,y\rangle+\langle\alpha,b\rangle} = 1] \\ &= \frac{1 + \mathbf{E}_{\alpha,\beta}\left[L(\alpha)R(\beta)R(\beta + A^T\alpha)(-1)^{\langle\alpha,b\rangle}\right]}{2}\end{aligned}$$

Consider the expectation on the right hand side of the above equation.

$$\begin{aligned}&\mathbf{E}_{\alpha,\beta}\left[L(\alpha)R(\beta)R(\beta + A^T\alpha)(-1)^{\langle\alpha,b\rangle}\right] \qquad (3)\\ &\leqslant \sum_{x,y} \hat{L}(x)\hat{R}(y)^2 \mathbf{E}_{\alpha,\beta}\left[\chi_x(\alpha)\chi_y(\beta)\chi_z(\beta + A^T\alpha)(-1)^{\langle\alpha,b\rangle}\right] \\ &\leqslant \sum_{x,y,x=Ay+b} \hat{L}(x)\hat{R}(y)^2 \\ &\leqslant \sqrt{\sum_{x,y,,x=Ay+b} \hat{R}(y)^2}\sqrt{\sum_{x,y,x=Ay+b} \hat{L}(x)^2\hat{R}(y)^2}\end{aligned}$$

In the above equation, the first term is bounded by 1, and therefore,

$$(3) \leqslant \sqrt{\sum_{x,y,,x=Ay+b} \hat{L}(x)^2\hat{R}(y)^2}$$

Consider a random assignment where a left vertex gets a label $x$ with probability $\hat{L}(x)^2$ and a right vertex gets a label $y$ with probability $\hat{R}(y)^2$. The probability that such a random assignment would satisfy the edge, and therefore the expected fraction of edges satisfied, is exactly

$$\sum_{x,y,x=Ay+b} \hat{L}(x)^2\hat{R}(y)^2$$

If at most $s$ fraction of edges can be satisfied by any assignment, then

$$s \geqslant \sum_{x,y,x=Ay+b} \hat{L}(x)^2\hat{R}(y)^2 \geqslant (2 \cdot \Pr[\text{Test accepts}] - 1)^2$$

or

$$\Pr[\text{Test accepts}] \leqslant \frac{1}{2} + \frac{\sqrt{s}}{2}$$

Therefore, the expected fraction of 3LIN constraints satisfied is at most $\frac{1}{2} + \frac{\sqrt{s}}{2}$. ◀

## References


1. 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). doi:10.1145/278298.278306.
2. 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.

3 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). doi:10.1137/100788161.

4 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proc. 46th ACM Symp. on Theory of Computing (STOC)*, pages 624–633, 2014. doi:10.1145/2591796.2591884.

5 Oded Goldreich. A Sample of Samplers: A Computational Perspective on Sampling. In Oded Goldreich, editor, *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, volume 6650 of *LNCS*, pages 302–332. Springer, 2011. doi:10.1007/978-3-642-22670-0_24.

6 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001. (Preliminary version in *29th STOC*, 1997). doi:10.1145/502090.502098.

7 Johan Håstad and Srinivasan Venkatesh. On the advantage over a random assignment. *Random Structures Algorithms*, 25(2):117–149, 2004. (Preliminary version in *34th STOC*, 2002). doi:10.1002/rsa.20031.

8 Subhash Khot. Improved Inaproximability Results for MaxClique, Chromatic Number and Approximate Graph Coloring. In *Proc. 42nd IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 600–609, 2001. doi:10.1109/SFCS.2001.959936.

9 Subhash Khot. Inapproximability Results for Computational Problems on Lattices. In Phong Q. Nguyen and Brigitte Vallée, editors, *The LLL Algorithm - Survey and Applications*, Information Security and Cryptography, pages 453–473. Springer, 2010. doi:10.1007/978-3-642-02295-1_14.

10 Subhash Khot and Assaf Naor. Linear Equations Modulo 2 and the $L_1$ Diameter of Convex Bodies. *SIAM J. Comput.*, 38(4):1448–1463, 2008. (Preliminary version in *48th FOCS*, 2007). doi:10.1137/070691140.

11 Dana Moshkovitz. The Projection Games Conjecture and the NP-Hardness of *lnn*-Approximating Set-Cover. *Theory Comput.*, 11:221–235, 2015. (Preliminary version in *15th APPROX*, 2012). doi:10.4086/toc.2015.v011a007.

12 Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *J. ACM*, 57(5), 2010. (Preliminary version in *49th FOCS*, 2008). doi:10.1145/1754399.1754402.

13 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.