Hardness of Minimizing and Learning DNF Expressions

Subhash Khot
NYU
khot@cs.nyu.edu

Rishi Saket
Georgia Tech
saket@cc.gatech.edu

October 24, 2008

Abstract

We study the problem of finding the minimum size DNF formula for a function $f : \{0,1\}^d \mapsto \{0,1\}$ given its truth table. We show that unless NP $\subseteq$ DTIME($n^{\mathrm{poly}(\log n)}$), there is no polynomial time algorithm that approximates this problem to within factor $d^{1-\varepsilon}$ where $\varepsilon > 0$ is an arbitrarily small constant. Our result essentially matches the known $O(d)$ approximation for the problem.

We also study weak learnability of small size DNF formulas. We show that assuming NP $\not\subseteq$ RP, for arbitrarily small constant $\varepsilon > 0$ and any fixed positive integer t, a two term DNF cannot be PAC-learnt in polynomial time by a t term DNF to within $\frac{1}{2} + \varepsilon$ accuracy. Under the same complexity assumption, we show that for arbitrarily small constants $\mu, \varepsilon > 0$ and any fixed positive integer t, an AND function (i.e. a single term DNF) cannot be PAC-learnt in polynomial time under adversarial μ-noise by a t-CNF to within $\frac{1}{2} + \varepsilon$ accuracy.

1 Introduction

Any given function $f : \{0,1\}^d \mapsto \{0,1\}$ can be written in an equivalent disjunctive normal form (DNF), i.e. an OR of some *terms*, where a *term* is an AND of literals. The *size* of the DNF formula is the number of terms it contains. Given a truth table of a function $f : \{0,1\}^d \mapsto \{0,1\}$, the problem of finding an equivalent DNF formula of minimum size is a well studied problem in computer science. We denote the problem by TT-MINDNF. It was first studied by Quine [Qui52, Qui56] in the context of mathematical logic and later by McCluskey[McC56] in relation to circuit design and both discovered a heuristic to solve the problem. Since then, a large number of heuristics and software tools have been developed; we refer the interested reader to [CS01] for a survey.

TT-MINDNF is a special case of the SET-COVER problem. The greedy set cover algorithm gives an $O(\log N) = O(d)$ approximation and runs in time polynomial in N where $N = 2^d$ is the size of the truth table. One the hardness side, the problem was proved to be NP-complete by Masek [Mas79]. Czort [Czo99] showed that unless P = NP, TT-MINDNF cannot be approximated efficiently to within any additive constant. Recently, Feldman [Fel06a] showed that TT-MINDNF cannot be approximated to within factor d^γ in polynomial time for some constant $\gamma > 0$ unless P = NP. Allender, Hellerstein, McCabe, Pitassi and Saks [AHM$^+$06] independently obtained the same inapproximability result under a stronger assumption that NP $\not\subseteq$ DTIME($n^{\mathrm{poly}(\log n)}$). The constant γ in both results is unspecified; it depends on the parameters of Raz's parallel repetition theorem [Raz98] and is presumably very small. In this paper, we show an essentially optimal hardness result that there is no polynomial time algorithm to approximate TT-MINDNF to within factor $d^{1-\varepsilon}$ assuming NP $\not\subseteq$ DTIME($n^{\mathrm{poly}(\log n)}$), where $\varepsilon > 0$ is an arbitrarily small constant.

Learning DNFs is a central problem in learning theory. Valiant [Val84] defined a widely studied model of learning, namely the Probably Approximately Correct (PAC) model. He showed that for every constant $k \geq 1$, k term DNF can be PAC learnt in polynomial time by a k-CNF, i.e. a CNF with at most k literals in each clause. For unrestricted DNFs (that is when the number of terms could be polynomially large in the number of variables n), the best learning algorithm runs in time $2^{O(n^{1/3}\log n)}$ due to Klivans and Servedio [KS04]. For learning under uniform distribution, Jackson [Jac97] showed that unrestricted DNFs can be learnt with membership queries, i.e. the algorithm can query for the value of the function at a point. Alekhnovich, Braverman, Feldman, Klivans and Pitassi [ABF$^+$08] gave an $n^{O(\sqrt{n\log n})}$ time algorithm to properly learn unrestricted DNFs, i.e. when the hypothesis is also a DNF.

On the hardness side, Pitt and Valiant [PV88] showed that unless NP $=$ RP, there is no efficient algorithm to PAC learn s-term DNF by an s-term DNF where s is unrestricted, i.e. $2 \le s \le n^c$, for any constant $c > 0$. In particular, Alekhnovich *et al.* [ABF$^+$08] showed that unless NP $=$ RP, there is no efficient algorithm to learn a 2 term DNF by a k term DNF for any constant k. Nock, Jappy and Sallantin [NJS98] showed that unless NP $\subseteq$ ZPP, given constants $0 \le \alpha \le 1 + \frac{1}{145}$ and $\beta \ge 0$, there is no efficient algorithm to PAC-learn n^c term DNF with $n^{\alpha c+\beta}$ term DNF. This was improved by Alekhnovich *et al.* [ABF$^+$08] who showed that unless NP $=$ RP, for any given constant $\alpha \ge 0$, n^c term DNF cannot be efficiently learnt by a $n^{\alpha c}$ term DNF. Their result rules out polynomial time proper PAC learning of DNFs, unless NP $=$ RP. This was further strengthened by Feldman [Fel06a] to the case when the algorithm even has access to membership queries. We note that all these intractability results rule out (under appropriate complexity assumptions) a learning algorithm that learns within error $\frac{1}{\mathrm{poly}(n)}$, but do not rule out a learning algorithm that learns within constant error (say within 1%). In other words, for the underlying optimization problem of finding a DNF formula consistent with the maximum number of given set of labeled examples, these are NP-hardness results and do not give APX-hardness. Another reason to study stronger inapproximability is that given an algorithm to PAC-learn find a $(\frac{1}{2} + \varepsilon)$-consistent hypothesis, using boosting techniques [Sch90] it can used to efficiently find a $(1 - \varepsilon)$-consistent hypothesis[1]. A hardness result for weak learning provides evidence against such boosting based approaches.

In this paper, we show that unless NP $=$ RP, for any constant $\varepsilon > 0$ and any fixed integer t, a 2-term DNF formula cannot be weakly learnt in polynomial time by a t-term DNF formula, i.e. within accuracy $\frac{1}{2} + \varepsilon$. We note that this hardness result is very much hypothesis dependent since for every $k \ge 2$, k-term DNF *can* be efficiently PAC-learnt by k-CNF as mentioned earlier [Val84]. We then investigate whether this algorithmic result holds under noise and resolve it negatively. We show that unless NP $=$ RP, for any constants $\varepsilon, \mu > 0$ and constant t, an AND (i.e. a single term DNF) cannot be learnt to accuracy of $\frac{1}{2} + \varepsilon$ by even a t-CNF formula, under adversarial μ-noise unless NP $=$ RP. This result generalizes the results of [Fel06b, FGKP06] which showed hardness of weak-learning noisy AND function by an AND function. We note that both the results are inapproximability results for the underlying optimization problem of finding a formula (t term DNF or t-CNF) maximizing the number of agreements on a given set of labeled examples. As inapproximability results, they are essentially optimal since a trivial formula that is either constant 1 or constant 0 agrees with half of the samples.[2]

2 Overview of Our Reductions

In this section we formally state our main results and give an overview of the proof techniques involved. The result for minimizing DNF formulas is a simple reduction from a new PCP that is constructed by a straightforward composition of known PCPs. The other two results are direct reductions from label cover.

[1] After applying the boosting algorithm, the hypothesis class is now a majority over a set of hypotheses used in the weak learning algorithm.

[2] Our inapproximability results hold under the assumption P $\neq$ NP, which translates to hardness of weak PAC-learning under the assumption NP $\neq$ RP.

2.1 Minimizing DNF formulas

Let the size of a DNF formula be the number of terms in it. We prove the following theorem.

Theorem 1 *For any $\varepsilon > 0$, there is no polynomial time algorithm that, given the truth table of a boolean function $f : \{0,1\}^d \mapsto \{0,1\}$, over d variables, computes an equivalent DNF formula for f of size within $d^{1-\varepsilon}$ of the minimum size equivalent DNF formula for f, unless NP $\subseteq$ DTIME($n^{\mathrm{poly}(\log n)}$).*

Since there is a $O(d)$ approximation algorithm for this problem, our hardness of approximation factor is essentially optimal. Our reduction actually proves hardness of approximation factor of $d^{1-\varepsilon}$ for a related problem, PHC-COVER of covering a subset $\mathcal{S}$ of the hypercube $\{0,1\}^d$ using minimum number of terms from a given set $\mathcal{T}$ of terms. Feldman [Fel06a] showed that this implies the same hardness of approximation factor for the problem of minimizing the size of DNF formulas.

Overview of Reduction: The reduction proceeds by first constructing a specialized version of a constraint satisfaction problem (or PCP) and then reducing it to PHC-COVER. However, for simplicity let us assume that we begin with a bipartite label cover problem over the label set $[k]$, with n vertices in each bipartition. Consider the vertices of the U layer. It is easy to see that we require at most $\log n$ variables so that every vertex in U is mapped to a unique setting of these variables. Call these variables *vertex variables* for the U layer. In the set of terms $\mathcal{T}$ of the PHC-COVER instance, we would like to have k unique terms for every vertex u in U, corresponding to the k labels for u. For this purpose we create k *label variables*, one for each label. For each vertex u and label i, there is a term which is 1 exactly on the unique setting, corresponding to u, of the vertex variables and when the label variable for label i is set to 0. Therefore for the U layer there are $\log n + k$ variables and nk terms, k for each vertex, where each term is over $\log n$ vertex variables and one label variable. We similarly construct distinct variables and terms for the V layer. In total we have $2(\log n + k)$ variables and $2nk$ terms.

Now, we construct the subset of points of the hypercube to be covered as follows. Pick an edge $e = (u, v)$ and two sets $S_1, S_2 \subseteq [k]$ such that $S_1 \times S_2$ does not contain any satisfying assignment to e. Set the coordinates such that only the terms corresponding to u and v are active. Set the coordinates corresponding to the labels in S_1 (in the U layer) and those corresponding to labels in S_2 (in the V layer) to be 1. Do this for all edges e of the label cover, and all such subsets S_1 and S_2 corresponding to C_e. It is easy to see that for a given edge $e = (u, v)$, if all points corresponding to such sets S_1, S_2 are covered then the set of terms corresponding to u and to v, must 'contain' a labeling to u and v, respectively, satisfying the edge e. Moreover, unless the number of terms chosen to cover the points is large enough, our analysis gives a way to pick a 'good' labeling to the vertices of the label cover. Therefore, in the YES case, the number of terms required to cover all points is small, in the NO case it is necessarily large.

While this reduction works even with the standard bipartite label cover, it does not give the desired hardness of approximation factor. In order to achieve that, we combine it with a multi layered constraint system based on a variant of the query efficient PCP of Samorodnitsky and Trevisan[ST00]. The PCP we construct is similar to the one constructed by Khot[Kho01] as it uses Hadamard encodings instead of Long Codes. We need this crucially as using Long Codes would blow up the size of the PCP in relation to the size of the label set. In order to use Hadamard encodings, we need to start with an instance with linear constraints. As a result, we lose perfect completeness. However, our reduction tolerates the loss of perfect completeness as long as the completeness parameter is suitably close to 1. In order to achieve this we start the construction of the PCP using the Max-3LIN instance constructed by Khot and Ponnuswami[KP06], which has completeness very close to 1 which we desire. We also need to ensure a large sized label set. For this purpose, the Hadamard encodings are over an appropriately large field extension of $\mathbb{F}[2]$. The PCP thus constructed is transformed into a multi layered constraint system via standard reductions.

We note that the previous hardness reductions of Feldman[Fel06a] and Allender *et al.* [AHM$^+$06] used a construction of certain *union free families of sets*, similar to the *partition systems* used in the reductions for the SET-COVER problem [LY94, Fei98]. Our result does not need such constructions (which we find interesting, since we in particular obtain $\log^{1-\varepsilon} N$ hardness for SET-COVER without using partition systems). In [Fel06a, AHM$^+$06], the parameters involved in constructing union free families limits the hardness factor achievable to $\sqrt{d}$ in addition to the limitation on γ (in the d^γ hardness) imposed by the parameters in Raz's parallel repetition theorem. Our reduction bypasses both these limitations.

2.2 Learning 2-term DNF by t-term DNF

We prove the following theorem.

Theorem 2 *For any $\varepsilon > 0$ and any given positive integer t, given a distribution $\mathcal{D}$ over point-value pairs (examples) (x, y), where $x \in \{0,1\}^n$ and $y \in \{0,1\}$, with the guarantee that there is a 2 term DNF formula that is consistent with all the examples of $\mathcal{D}$, unless NP $=$ RP there is no polynomial time algorithm to compute a DNF formula of up to t terms that is consistent with the examples with probability $\frac{1}{2} + \varepsilon$ under the distribution $\mathcal{D}$.*

The result is essentially optimal since a trivial formula that is either the constant 1 or the constant 0 satisfies the examples with probability $\frac{1}{2}$. The distribution $\mathcal{D}$ in our instance is supported over polynomially (in n) many points of the hypercube, and therefore it can be given explicitly.

Overview of Reduction: Our reduction proves an equivalent result for learning 2-clause CNF by t-clause CNF. We give a direct reduction from the bipartite label cover problem with vertex sets U and V, and label sets $[m]$ and $[k]$ respectively. The examples of the distribution $\mathcal{D}$ simulate the junta and consistency tests. We create one coordinate for every vertex and its potential label. So we have $m|U| + k|V|$ coordinates. The 1 examples have the property that there is an edge (u, v) such that all the m coordinates corresponding to u and k coordinates corresponding to v are set to 1 and all other coordinates are set to 0. The 0 examples are constructed by choosing a vertex $u \in U$ and a set $\alpha \subseteq [m]$ and setting all the coordinates of u corresponding to $[m] \setminus \alpha$ to be 1. Moreover for every neighbor v of u, all coordinates corresponding to $\pi_{uv}^{-1}(\alpha)$ are set to 1, where π_{uv} is the projection map for the edge (u, v). All the other coordinates are set to 0.

Suppose there is a labeling σ to the vertices that satisfies all edges. Now consider the clause C_U consisting of the variables corresponding to vertex u and its label $\sigma(u)$ for all $u \in U$. Let clause C_V be similarly defined for V. It is easy to see that the formula $C_U \wedge C_V$ satisfies all the examples. In the NO case we show that if there a t clause CNF that is consistent with the examples with probability at least $\frac{1}{2} + \varepsilon$, then one can construct a labeling to the vertices of label cover which satisfies a significant fraction of edges. This leads to a contradiction if we choose the soundness parameter of the label cover to be small enough.

2.3 Learning AND by t-CNF under adversarial noise

We prove the following theorem.

Theorem 3 *For any constants $\varepsilon, \mu > 0$ and any positive integer t, given a distribution $\mathcal{D}$ over point-value pairs (examples) (x, y), where $x \in \{0,1\}^n$ and $y \in \{0,1\}$, with the guarantee that there is an AND formula that is consistent with the examples with probability (under $\mathcal{D}$) at least $1 - \mu$, unless NP $=$ RP there is no polynomial time algorithm to compute a t-CNF formula, i.e. a CNF formula with at most t literals in each clause, that is consistent with examples with probability(under $\mathcal{D}$) at least $\frac{1}{2} + \varepsilon$.*

Again the result is essentially optimal since it is trivial to output a formula that is consistent with half the examples. Moreover, without any noise an AND formula can be properly learnt in polynomial time. Our reduction proves an equivalent result for learning OR by t-DNF, i.e. DNF formula with at most t literals in each term. The reduction is similar to the one described in section 2.2, and the distribution $\mathcal{D}$ has a polynomial (in n) support and can be given explicitly. It starts with an instance of bipartite label cover. In a similar manner the examples simulate the junta and consistency tests, with the property that in the YES instance, the labeling gives a OR formula that is consistent with the examples with probability close to 1. In the NO case, any DNF formula with at most t literals in each term that is consistent with the examples with probability at least $\frac{1}{2} + \varepsilon$ yields a labeling to the vertices of the label cover that satisfies a significant fraction of edges, and choosing the soundness parameter of the label cover to be small enough, this leads to a contradiction.

Organization of the paper. In the next section we formally define the problems considered, and the tools we require for our reductions. We present the hardness result for minimizing DNF formulas in section 4. It is a reduction from a multi-layered CSP to PHC-COVER. Due to space constraints, the results for learning 2 term DNF, learning AND under adversarial noise and the construction of the multi-layered CSP are presented in appendices A, B and C respectively.

3 Preliminaries

Let $f : \{0,1\}^d \mapsto \{0,1\}$ be a boolean function. We say that a boolean function g is *equivalent* to f if it agrees with f at every point of the hypercube. A DNF formula is a OR of *terms* where a *term* is an AND of literals. Similarly, a CNF formula is a AND of clauses, where each clause is an OR of literals. We define the problem TT-MINDNF as follows.

Definition 1 *The problem* TT-MINDNF *is the following: given the truth table of a boolean function f on d variables, to find an equivalent DNF formula ϕ with the minimum number of terms.*

In our reduction we prove a hardness of approximation factor of $d^{1-\varepsilon}$ for any $\varepsilon > 0$, for the partial hypercube cover (PHC-COVER) problem which is defined as follows.

Definition 2 *The problem* PHC-COVER *is the following: given a subset $\mathcal{S} \subseteq \{0,1\}^d$, and a set of terms $\mathcal{T}$, to find a minimum subset of terms $\mathcal{T}^* \subseteq \mathcal{T}$ that covers all the points in $\mathcal{S}$.*

Feldman [Fel06a] showed that a hardness of approximation factor of d^γ for PHC-COVER implies same hardness factor for TT-MINDNF, for any constant $\gamma > 0$. Therefore, our result implies hardness of approximation factor of $d^{1-\varepsilon}$ for TT-MINDNF.

We also define the following problems related to learning boolean functions.

Definition 3 *For any positive integer t, the problem of* LEARN-t-TERM-DNF *is the following: given a distribution $\mathcal{D}$ on point-value pairs (examples) (x, y), where $x \in \{0,1\}^n$ and $y \in \{0,1\}$, the goal is to find a DNF formula with up to t terms that is consistent with the examples with maximum probability under the distribution $\mathcal{D}$.*

Definition 4 *For any positive integer t, the problem of* LEARN-t-CNF *is the following: given a distribution $\mathcal{D}$ on point-value pairs (examples) (x, y), where $x \in \{0,1\}^n$ and $y \in \{0,1\}$, the goal is to find a CNF formula with up to t literals in each clause that is consistent with the examples with maximum probability under the distribution $\mathcal{D}$.*

The starting point for our inapproximability results for LEARN-t-TERM-DNF and LEARN-t-CNF is the label cover problem, which is defined below.

Definition 5 *An instance $\mathcal{L}$ of* $\textsc{LabelCover}(m,k)$ *consists of a bipartite graph $G(U,V,E)$ and a set of projections $\{\pi_{uv}\}_{(u,v)\in E}$, where $\pi_{uv} : [k] \mapsto [m]$ for every edge $(u,v) \in E$, where $u \in U$ and $v \in V$. A labeling $\sigma_U : U \mapsto [m]$ and $\sigma_V : V \mapsto [k]$ satisfies the edge (u,v), iff $\pi_{uv}(\sigma_V(v)) = \sigma_U(u)$. The goal is to find a labeling that satisfies maximum number of edges of $\mathcal{L}$.*

The following theorem is a consequence of the PCP Theorem [AS98, ALM$^+$98] and Raz's Parallel Repetition Theorem [Raz98].

Theorem 4 *For any constant $\delta > 0$, there exist m and k such that, given an instance $\mathcal{L}$ of* $\textsc{LabelCover}(m,k)$*, it is NP-hard to distinguish between the following two cases,*
YES Case. There is a labeling to the vertices of $\mathcal{L}$ that satisfies all the edges.
NO case. Any labeling to the vertices of $\mathcal{L}$ satisfies at most δ fraction of the edges.

The following theorem is proved in Appendix A and implies Theorem 2.

Theorem 5 *For any $\varepsilon > 0$ and any positive integer $t > 0$, given an instance of* LEARN-t-TERM-DNF *consisting of a distribution $\mathcal{D}$ on the set of examples (x,y), where $x \in \{0,1\}^n$ and $y \in \{0,1\}$, it is NP-hard to distinguish between the following cases,*
YES Case. There is a two term DNF ϕ that is consistent with all the examples of the distribution $\mathcal{D}$.
NO Case. There is no DNF formula ϕ' of up to t terms that is consistent with the examples of $\mathcal{D}$ with probability $\frac{1}{2} + \varepsilon$.

The following theorem is proved in Appendix B and implies Theorem 3.

Theorem 6 *For any $\mu, \varepsilon > 0$ and any positive integer $t > 0$, given an instance of* LEARN-t-CNF *consisting of a distribution $\mathcal{D}$ on the set of examples (x,y), where $x \in \{0,1\}^n$ and $y \in \{0,1\}$, it is NP-hard to distinguish between the following cases,*
YES Case. There is a AND formula that is consistent with all the examples of the distribution $\mathcal{D}$ with probability at least $1-\mu$.
NO Case. There is no CNF formula ϕ' with up to t literals in each clause that is consistent with the examples of $\mathcal{D}$ with probability $\frac{1}{2} + \varepsilon$.

For the reduction to TT-MINDNF we require a more specialized constraint satisfaction problem which we define below. Let t be a parameter. We define the problem t-LAYERED-CSP as follows.

Definition 6 *An instance of t-*LAYERED-CSP *consists of the following,*

1. A t-uniform hypergraph $G(V,E)$ which has the following properties,

a. Let V be the vertex set of the hypergraph. Then V can be partitioned into sets $V_1, \ldots, V_t$ such that each edge of the hypergraph has exactly one vertex from each V_i for $i = 1, \ldots, t$. Moreover $|V_1| = |V_2| = \cdots = |V_t|$.

b. Every vertex in V has the same degree.

2. A set of labels $[k]$, and constraints for each hyperedge of the graph defined as follows,

a. Let $e = (v_1, v_2, \ldots, v_t)$ be a hyperedge such that $v_i \in V_i$ for all $i = 1, \ldots, t$. Then the constraint C_e is a non empty subset of $[k]^t$.

b. Let $\sigma : V \mapsto [k]$ be a labeling of the vertices in V. Then the hyperedge $e = (v_1, v_2, \ldots, v_t)$, where $v_i \in V_i$ for all $i = 1, \ldots, t$, is satisfied iff $(\sigma(v_1), \ldots, \sigma(v_t)) \in C_e$.

The goal is to find a labeling $\sigma : V \mapsto [k]$ to the vertices of V that satisfies the maximum number of hyperedges in E.

The following theorem is proved in Appendix C.

Theorem 7 *There is an absolute constant $\xi > 0$ such that, for a given arbitrarily large integer $t > 0$, there is a DTIME($n^{\mathrm{poly}(\log n)}$) time reduction from* 3SAT *to an instance of t-*LAYERED-CSP *with $|V| = n$ and $k = \theta(\log^2 n)$ such that,*

YES CASE: If the 3SAT *formula is satisfiable then there is a set $V' \subset V$ of vertices of size at most $n/(2^{(\log n)^{\xi}})$ and a labeling $\sigma^* : V \setminus V' \mapsto [k]$ such that,*

1. *(Strong Completeness) σ^* satisfies all hyperedges induced by $V \setminus V'$.*
2. *(Extendability) For any hyperedge $e \in E$ (possibly containing vertices from V'), there is an labeling σ'_e to vertices in $e \cap V'$ such that σ^* extended by σ'_e satisfies hyperedge e.*

NO CASE: If the 3SAT *formula is not satisfiable then any labeling σ to the vertices of V satisfies at most $k^{-t+O(\sqrt{t})}$ fraction of the hyperedges.*

The following theorem is proved in Section 4 via a reduction from t-LAYERED-CSP, and it implies Theorem 1.

Theorem 8 *For any $\varepsilon > 0$, there exists a function $h : \mathbb{Z}^+ \mapsto \mathbb{Z}^+$ such that given an instance of* PHC-COVER *consisting of a subset $\mathcal{S}$ of $\{0,1\}^d$ and a set of terms $\mathcal{T}$,unless NP $\subseteq$ DTIME($n^{\mathrm{poly}(\log n)}$), there is no polynomial time algorithm to distinguish between the following two cases,*

YES Case. There is a subset $\mathcal{T}^* \subseteq \mathcal{T}$ of size at most $h(d)$ that covers all the points in $\mathcal{S}$.

NO Case. There is no subset $\mathcal{T}' \subseteq \mathcal{T}$ of size at most $d^{1-\varepsilon}h(d)$ that covers all the points in $\mathcal{S}$.

4 Reduction from t-LAYERED-CSP to PHC-COVER

In this section we show a reduction from the problem t-LAYERED-CSP to PHC-COVER. With the t-LAYERED-CSP problem as defined in Def 6, we first construct the set of variables.
Vertex Variables: For every layer V_i $(1 \leq i \leq t)$, we have a set $P^i = \{x_{ij}\}_{1 \leq j \leq D}$ of D variables where $D = \lceil \log |V_i| \rceil$. We refer to them as *vertex variables* for layer i. Clearly we have a one to one mapping from every vertex $u \in V_i$ to a setting of the variables in P^i for every layer $1 \leq i \leq t$. Call this setting $s^i(u)$. Thus, we have a set of variables for every layer whose settings encode all the vertices of that layer.
Label variables: For every layer V_i $(1 \leq i \leq t)$, we have a set $Q^i = \{y_{ij}\}_{1 \leq j \leq k}$ of k variables each corresponding to a label. We refer to this set as *label variables* for the layer i.

Let $\mathcal{M} = \bigcup_{i=1}^{t}(P^i \cup Q^i)$ be the set of all the variables, and let $d := |\mathcal{M}| = t(D + k)$. We now describe the set of terms $\mathcal{T}$.

Terms: Let V_i $(1 \leq i \leq t)$ be a layer of vertices and let $u \in V_i$. Then there is a set $T^i(u)$ of k terms corresponding to u as follows. Let $t^i(u)$ be the unique AND of the literals corresponding to the variables in $P^i = \{x_{ij}\}_{1 \leq j \leq D}$ such that $t^i(u)$ is 1 only on the setting $s^i(u)$ of the variables in P^i corresponding to u. Let,

$$T^i(u) := \{t^i(u) \wedge \overline{y_{ij}} \;\mid\; 1 \leq j \leq k\}.$$

Therefore, for every layer i ($1 \leq i \leq t$) and every $u \in V_i$, there is a set of k terms $T^i(u)$. We define,

$$\mathcal{T} = \bigcup_{i=1}^{t} \bigcup_{u \in V_i} T^i(u).$$

In all there are nk terms. Next we define the set of points $\mathcal{S} \subseteq \{0,1\}^{\mathcal{M}}$ for our instance of PHC-COVER.

Points: Let $e = (v_1, v_2, \ldots, v_t)$ be a hyperedge in the graph G, where $v_i \in V_i$ for $1 \leq i \leq t$, and let $C_e \subseteq [k]^t$ be its constraint. Let $I = (I_1, I_2, \ldots, I_t)$, be a t tuple where $I_i \subseteq [k]$ and let $\omega(I) := I_1 \times I_2 \times \cdots \times I_t$. We consider those $I \in (2^{[k]})^t$ such that $\omega(I) \cap C_e = \emptyset$. Note that this is trivially true if any of I_i is empty. In other words, the set $\omega(I)$ does not 'contain' any satisfying assignment to the hyperedge e. Let $\mathcal{I}_e$ be the set of all such t-tuples I corresponding to hyperedge e. Formally,

$$\mathcal{I}_e = \{I \in (2^{[k]})^t \ \mid \ \omega(I) \cap C_e = \emptyset\}$$

For every such $I \in \mathcal{I}_e$, we create the following point $\gamma_e(I) \in \{0,1\}^{\mathcal{M}}$ as follows. The coordinates corresponding to P^i are set to $s^i(v_i)$ for all $1 \leq i \leq t$. For $1 \leq i \leq t$, the coordinate corresponding to $y_{ij} \in Q^i$ is set to 1 if $j \in I_i$ and 0 otherwise, for every $1 \leq j \leq k$. We define,

$$\mathcal{S} := \bigcup_{e \in E} \bigcup_{I \in \mathcal{I}_e} \{\gamma_e(I)\}$$

Now consider any subset $\mathcal{T}^* \subseteq \mathcal{T}$. Let i ($1 \leq i \leq t$) be a layer, and $u \in V_i$ be a vertex. Define,

$$L^i_{\mathcal{T}^*}(u) := \{j \ \mid \ t^i(u) \wedge \overline{y_{ij}} \in \mathcal{T}^*\},$$

for all $1 \leq i \leq t$ and $u \in V_i$. Thus, $L^i_{\mathcal{T}^*}(u)$ is precisely the set of labels of u such that the corresponding terms are present in $\mathcal{T}^*$, where $u \in V_i$. Additionally, for every hyperedge $e = (v_1, v_2, \ldots, v_t) \in E$, let,

$$L_{\mathcal{T}^*}(e) = L^1_{\mathcal{T}^*}(v_1) \times \cdots \times L^t_{\mathcal{T}^*}(v_t).$$

The following is a simple lemma.

Lemma 9 *Let $\mathcal{T}^* \subseteq \mathcal{T}$. Then $\mathcal{T}^*$ covers all the points in $\mathcal{S}$ if and only if for every hyperedge $e = (v_1, v_2, \ldots, v_t) \in E$, where $v_i \in V_i$ for $1 \leq i \leq t$, $L_{\mathcal{T}^*}(e) \cap C_e \neq \emptyset$.*

Proof: Let us fix a hyperedge $e = (v_1, \ldots, v_t)$ where $v_i \in V_i$ for all $1 \leq i \leq t$. Consider any point $\gamma_e(I)$ for $I \in \mathcal{I}_e$. First we show that $\gamma_e(I)$ can be covered by terms only from the sets $T^i(v_i)$ for $1 \leq i \leq t$. Let u be any vertex such that $u \neq v_i$ for $1 \leq i \leq t$. Assume that $u \in V_{i'}$ for some $1 \leq i' \leq t$. By the construction of $\gamma_e(I)$, the coordinates corresponding to $P^{i'}$ are set to $s^{i'}(v_{i'})$, and the AND formula $t^{i'}(u)$ is 0 on this setting since $v_{i'} \neq u$.

Since, for any point $\gamma_e(I)$, the variables P^i are set to $s^i(v_i)$ for all $1 \leq i \leq t$, all the AND formulas, $t^i(v_i)$ are set to 1. Therefore, $\gamma_e(I)$ is not covered by $\mathcal{T}^*$ if and only if, for all layers i ($1 \leq i \leq t$), the coordinates corresponding to $\{y_{ij} \ \mid \ j \in L^i_{\mathcal{T}^*}(v_i)\}$ are set to 1. Equivalently, $\omega(I) \supseteq L^1_{\mathcal{T}^*}(v_1) \times \cdots \times L^t_{\mathcal{T}^*}(v_t) = L_{\mathcal{T}^*}(e)$. By the definition of $\mathcal{I}_e$, $\omega(I) \cap C_e = \emptyset$. Therefore, if there is an $I \in \mathcal{I}_e$ such that $\gamma_e(I)$ is not covered by $\mathcal{T}^*$, then $L_{\mathcal{T}^*}(e) \cap C_e = \emptyset$. For the reverse direction, we note that if $L_{\mathcal{T}^*}(e) \cap C_e = \emptyset$, then we can set $I = (L^1_{\mathcal{T}^*}(v_1), \ldots, L^t_{\mathcal{T}^*}(v_t))$, and $\gamma_e(I)$ is not covered by $\mathcal{T}^*$. This completes the proof. ■

4.1 Analysis

To prove our hardness of approximation result for PHC-COVER we reduce from the t-LAYERED-CSP instance obtained from the Theorem 7 to an instance of PHC-COVER via the reduction described above. Next we present the analysis.

4.1.1 YES Case In the YES case we have a set of vertices $V' \subseteq V$ of size at most $n/2^{(\log n)^{\xi}}$, and a labeling σ^* to the vertices $V \setminus V'$ satisfying the properties in Theorem 7. Now we construct a set of terms $\mathcal{T}^* \subseteq \mathcal{T}$ as follows. For every layer i, ($1 \leq i \leq t$), do the following. For every vertex $u \in V_i$, if $u \in V'$ then $\mathcal{T}^*$ contains the k terms in the set $T^i(u)$ corresponding to u. Otherwise, if $u \notin V'$, then $\mathcal{T}^*$ contains only the term $t^i(u) \wedge \overline{y_{i\sigma^*(u)}}$, i.e. the term in $T^i(u)$ corresponding to the label of u given by σ^*.

We show that $\mathcal{T}^*$ covers all the points in $\mathcal{S}$. Let $e = (v_1, \ldots, v_t)$ be a hyperedge, where $v_i \in V_i$ for $1 \leq i \leq t$. We have two cases.

Case 1. e is induced by $V \setminus V'$. The labeling σ satisfies e. Then $L^i_{\mathcal{T}^*}(v_i) = \{\sigma^*(v_i)\}$, for $1 \leq i \leq t$ and $L_{\mathcal{T}^*}(e) = \{(\sigma^*(v_1), \ldots, \sigma^*(v_t))\}$. And therefore $L_{\mathcal{T}^*}(e) \cap C_e \neq \emptyset$.

Case 2. e contains vertices from V'. Then, $L^i_{\mathcal{T}^*}(v_i) = \{\sigma^*(v_i)\}$ if $v_i \in V \setminus V'$ and $L^i_{\mathcal{T}^*}(v_i) = \{1, 2, \ldots, k\}$ otherwise. Now, by the Extendability property in Theorem 7, there is a labeling σ'_e to vertices in $e \cap V'$ such that σ^* extended by σ'_e satisfies e. Clearly, this implies there is a labeling to the vertices v_i in e from the sets $L^i_{\mathcal{T}^*}(v_i)$ for $1 \leq i \leq t$ that satisfies the hyperedge e. Therefore, $L_{\mathcal{T}^*}(e) \cap C_e \neq \emptyset$.

Therefore, for every edge e, $L_{\mathcal{T}^*}(e) \cap C_e \neq \emptyset$. And by Lemma 9 the set of terms $\mathcal{T}^*$ covers all the points in $\mathcal{S}$. The number of terms in $\mathcal{T}^*$ is,

$$\begin{aligned} &|V \setminus V'| + k|V'| \\ &\leq n\left(1 - \frac{1}{2^{(\log n)^{\xi}}}\right) + k\left(\frac{n}{2^{(\log n)^{\xi}}}\right). \end{aligned}$$

Since, we have $k = \theta(\log^2 n)$, the above expression is at most $2n$ for large enough n. Therefore, the number of terms in $\mathcal{T}^*$ is at most $2n$.

4.1.2 NO Case Suppose that there is a set of terms $\mathcal{T}' \subseteq \mathcal{T}$ that covers all the points in $\mathcal{S}$. By Lemma 9, for every hyperedge e, $L_{\mathcal{T}'}(e) \cap C_e \neq \emptyset$. Now, consider the labeling σ' constructed in a randomized manner as follows. Let u be a vertex in, say, V_i for some $1 \leq i \leq t$. Select $\sigma'(u)$ to be a random label from $L^i_{\mathcal{T}'}(u)$. Suppose $e = (v_1, \ldots, v_t)$ is a hyperedge where $v_i \in V_i$ for $1 \leq i \leq t$. Since $L_{\mathcal{T}'}(e)$ contains a satisfying assignment from C_e we have the following,

$$\Pr\left[e \text{ is satisfied by } \sigma'\right] \geq \frac{1}{\prod_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)|}.$$

Therefore, the expected fraction of edges satisfied is at least,

$$\mathrm{E}_{\sigma'}\left[\text{Fraction of edges satisfied by } \sigma'\right] \geq \mathrm{E}_{e=(v_1,\ldots,v_t)}\left[\frac{1}{\prod_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)|}\right] \quad (1)$$

The left hand side of the above expression is less than the soundness $\delta = k^{-t+O(\sqrt{t})}$ of the NO case. Therefore,

$$\mathrm{E}_{e=(v_1,\ldots,v_t)}\left[\frac{1}{\prod_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)|}\right] \leq \delta.$$

Therefore, for at least $\frac{1}{2}$ fraction of the hyperedges $e = (v_1, \ldots, v_t)$ we have,

$$\begin{aligned} & \frac{1}{\prod_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)|} \leq 2\delta \\ \Rightarrow \quad & \prod_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)| \geq \frac{1}{2\delta} \\ \Rightarrow \quad & \frac{\sum_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)|}{t} \geq \frac{1}{(2\delta)^{\frac{1}{t}}}. \end{aligned} \tag{2}$$

Now, since each vertex in V has the same degree,

$$\begin{aligned} |\mathcal{T}'| &= \sum_{i=1}^{t} \sum_{u \in V_i} L^i_{\mathcal{T}'}(u) \\ &= n\mathrm{E}_{e=(v_1,\ldots,v_t)} \left[\frac{\sum_{i=1}^{t} |L^i_{\mathcal{T}'}(v_i)|}{t} \right] \end{aligned} \tag{3}$$

And combining equations (2) and (3), we have,

$$|\mathcal{T}'| \geq n \left(\frac{1}{2}\right) \left(\frac{1}{(2\delta)^{\frac{1}{t}}}\right)$$

Substituting the value of δ, we obtain that,

$$|\mathcal{T}'| \geq \frac{nk^{1-O\left(\frac{1}{\sqrt{t}}\right)}}{2^{1+\frac{1}{t}}}$$

Since t can be made to be an arbitrarily large constant, combining the above with the analysis of the YES case, we get a gap of $k^{1-\varepsilon}$ for the optimum of the instance of PHC-COVER, for any constant $\varepsilon > 0$. Also, the number of variables d is at most $t(\log n + k) = O(k)$, since $k = \theta(\log^2 n)$. In terms of d, we obtain a gap of $d^{1-\varepsilon}$. Clearly the reduction runs in time $2^{O(d)}$, which is $2^{O(k)} = O(2^{\log^3 n})$. Therefore, along with the inapproximability of t-LAYERED-CSP given in Theorem 7, this proves Theorem 8.

5 Conclusion

An open problem is to improve upon the $d^{1-\varepsilon}$ hardness of approximation factor for any constant $\varepsilon > 0$ to a $\Omega(d)$ factor hardness for TT-MINDNF.

Another open question is to obtain results on hardness of weak learning DNFs even when membership queries are available. It would be interesting to extend hardness of weak learning results for polynomial size DNFs.

6 Acknowledgment

We would like to thank Vitaly Feldman for introducing the problem of logic minimization to us, and for helpful discussions in the early stages of this work.

References

[ABF$^+$08] M. Alekhnovich, M. Braverman, V. Feldman, A. Klivans, and T. Pitassi. The complexity of properly learning simple concept classes. *J. Comput. Syst. Sci.*, 74(1):16–34, 2008.

[AHM$^+$06] E. Allender, L. Hellerstein, P. McCabe, T. Pitassi, and M. Saks. Minimizing DNF formulas and AC^0_d circuits given a truth table. In *IEEE Conference on Computational Complexity*, pages 237–251, 2006.

[ALM$^+$98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[AS98] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[CS01] O. Coudert and T. Sasao. *Two level logic minimization*. Kluwer Academic Publishers, 2001.

[Czo99] S. Czort. The complexity of minimizing disjunctive normal form formulas. Master's Thesis, University of Aarhus. 1999.

[Eng00] L. Engebretsen. Lower bounds for non-boolean constraint satisfaction. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(42), 2000.

[Fei98] U. Feige. A threshold of ln *n* for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[Fel06a] V. Feldman. Hardness of approximate two-level logic minimization and PAC learning with membership queries. In *Proc.* 38^{th} *ACM STOC*, pages 363–372, 2006.

[Fel06b] V. Feldman. Optimal hardness results for maximizing agreements with monomials. In *IEEE Conference on Computational Complexity*, pages 226–236, 2006.

[FGKP06] V. Feldman, P. Gopalan, S. Khot, and A. Ponnuswami. New results for learning noisy parities and halfspaces. In *Proc.* 47^{th} *IEEE FOCS*, pages 563–574, 2006.

[Jac97] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *J. Comput. Syst. Sci.*, 55(3):414–440, 1997.

[Kho01] S. Khot. Improved inaproximability results for maxclique, chromatic number and approximate graph coloring. In *Proc.* 42^{nd} *IEEE FOCS*, pages 600–609, 2001.

[KP06] S. Khot and A. Ponnuswami. Better inapproximability results for maxclique, chromatic number and min-3Lin-deletion. In *ICALP*, pages 226–237, 2006.

[KS04] A. Klivans and R. Servedio. Learning DNF in time $2^{\tilde{\mathrm{O}}(\mathrm{n}^{1/3})}$. *J. Comput. Syst. Sci.*, 68(2):303–318, 2004.

[LY94] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994.

[Mas79] W. Masek. Some NP-complete set covering problems. Unpublished. 1979.

[McC56] E. McCluskey. Minimization of boolean functions. *Bell Sys. Tech. Jour.*, 35:1417–1444, 1956.

[NJS98] R. Nock, P. Jappy, and J. Sallantin. Generalized graph colorability and compressibility of boolean formulae. In *Proc. ISAAC*, pages 237–246, 1998.

[PV88] L. Pitt and L. Valiant. Computational limitations of learning from examples. *J. ACM*, 35(4), 1988.

[Qui52] W. Quine. The problem of simplifying truth functions. *American Mathematical Monthly*, 59:521–531, 1952.

[Qui56] W. Quine. A way to simplify truth functions. *American Mathematical Monthly*, 62:627–631, 1956.

[Rao08] A. Rao. Parallel repetition in projection games and a concentration bound. In *Proc.* 40^{th} *ACM STOC*, 2008.

[Raz98] R. Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998.

[Sch90] R. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

[ST98] M. Sudan and L. Trevisan. Probabilistically checkable proofs with low amortized query complexity. In *Proc.* 39^{th} *IEEE FOCS*, pages 18–27, 1998.

[ST00] A. Samorodnitsky and L. Trevisan. A PCP characterization of NP with optimal amortized query complexity. In *Proc.* 32^{nd} *ACM STOC*, pages 191–199, 2000.

[TS97] A. Ta-Shma. A note on PCP vs MIP. *Information Processing Letters*, 58(3):475–484, 1997.

[Val84] L. Valiant. A theory of the learnable. In *Proc.* 16^{th} *ACM STOC*, pages 436–445, 1984.

A Hardness of Learning 2-clause CNF by t-clause CNF

In this section we prove Theorem 5. For convenience we prove an equivalent result for learning 2-clause CNF by t-clause CNF.

We start with an instance $\mathcal{L}$ of $\textsc{LabelCover}(m, k)$ consisting of a bipartite graph $G(U, V, E)$, set of labels $[m]$ (for vertices in U), $[k]$ (for vertices in V), the projections $\pi_{uv} : [k] \mapsto [m]$ for every edge $e = (u, v) \in E$, where vertices in U have degree d_U, and those in V have degree d_V. All the parameters are constants independent of the sizes $N_U = |U|$ and $N_V = V$. Let $N = N_U + N_V$.

A.1 Construction

Variables. First we define the set of variables. Let v be any vertex in V. We have the set of variables $S_v = \{x_i^v\}_{i=1}^k$. Similarly, let u be any vertex in U, and let $S_u = \{y_i^u\}_{i=1}^m$. Thus, we have one variable for every vertex and every potential label for that vertex. Let,

$$\mathcal{S} = (\cup_{u \in U} S_u) \bigcup (\cup_{v \in V} S_v)$$

be the set of all variables. Let the corresponding boolean hypercube be $\{0, 1\}^{\mathcal{S}}$ where the coordinates are indexed by the variables in $\mathcal{S}$.

Distribution. We now describe how the oracle generates a sample point. This describes the distribution $\mathcal{D}$ on the samples. Let $\mu \in (0, 1)$ be a ‘perturbation’ parameter, which we will fix later. On being queried for a sample, the oracle does the following,

1. Chooses a vertex $u \in U$ at random from the vertices in U. Let $N(u) \subseteq V$ be the neighborhood of u.
2. With probability $\frac{1}{2}$ does the following,

   2a. Picks $v \in N(u)$ at random.

   2b. Creates the following point $Z_1^{uv} \in \{0,1\}^{\mathcal{S}}$ as follows,

$$\forall j \in [k], \qquad Z_1^{uv}(x_j^{v'}) = \begin{cases} 1 & \text{if } v' = v \\ 0 & \text{otherwise} \end{cases}$$

   and,

$$\forall j \in [m], \qquad Z_1^{uv}(y_j^{u'}) = \begin{cases} 1 & \text{if } u' = u \\ 0 & \text{otherwise} \end{cases}$$

   2.c Output the sample $(Z_1^{uv}, 1)$.

3. With probability $\frac{1}{2}$ does the following,

   3.a Chooses a set $\alpha \subseteq [m]$ by picking every $i \in [m]$ independently with probability μ.

   3.b Creates the following point $Z_0^{u\alpha} \in \{0,1\}^{\mathcal{S}}$ as follows,

$$\forall j \in [k], \qquad Z_0^{u\alpha}(x_j^{v'}) = \begin{cases} 0 & \text{if } v' \notin N(u) \\ 1 & \text{if } v' \in N(u) \text{ and } \pi_{uv'}(j) \in \alpha \\ 0 & \text{if } v' \in N(u) \text{ and } \pi_{uv'}(j) \notin \alpha \end{cases}$$

   and,

$$\forall j \in [m], \qquad Z_0^{u\alpha}(y_j^{u'}) = \begin{cases} 0 & \text{if } u' \neq u \\ 0 & \text{if } u' = u \text{ and } j \in \alpha \\ 1 & \text{if } u' = u \text{ and } j \notin \alpha \end{cases}$$

   3.c Output the sample $(Z_0^{u\alpha}, 0)$.

We note that the distribution has a polynomial (in $|\mathcal{S}|$) support, and therefore can be given explicitly.

Let $t > 0$ be a given integer and $\varepsilon > 0$ be a given parameter. We will show that if $\mathcal{L}$ is a YES instance of LABELCOVER(m, k), i.e if there is a labeling that satisfies all edges then there is a 2-clause CNF which is consistent with all the samples. On the other hand, if $\mathcal{L}$ is a NO instance then there is no t clause CNF that is consistent with the samples with probability $\frac{1}{2} + \varepsilon$ under the distribution $\mathcal{D}$ provided the soundness η of the label cover instance is chosen to be suitably small.

A.2 YES Case

Let $\mathcal{L}$ be a YES instance of LABELCOVER(m, k). Then there is a labeling σ to the vertices of G that satisfies all the edges. Consider the following two clauses,

$$C_V = \bigvee_{v \in V} x^v_{\sigma(v)} \ ,$$

and,

$$C_U = \bigvee_{u \in U} y^u_{\sigma(u)}.$$

Let $\phi = C_V \wedge C_U$. We will show that ϕ is consistent with all the data points.

Consider any data point of the form $(Z_1^{uv}, 1)$ where $Z_1^{uv} \in \{0,1\}^{\mathcal{S}}$. Recall that Z_1^{uv} was generated by picking a vertex $u \in U$ then a vertex $v \in N(u)$. By the construction of Z_1^{uv}, clearly $Z_1^{uv}(x^v_{\sigma(v)}) = 1$ and $Z_1^{uv}(y^u_{\sigma(u)}) = 1$. Therefore, the clauses C_V and C_U, both are 1 on the point Z_1^{uv}, and therefore ϕ is also 1 at the point Z_1^{uv}. So the formula ϕ is consistent with all the data points of the form $(Z_1^{uv}, 1)$.

Now consider any data point $(Z_0^{u\alpha}, 0)$. Recall that $Z_0^{u\alpha}$ was constructed by first picking a vertex $u \in U$ and then a set $\alpha \subseteq [m]$. We consider two cases.

Case 1. Let $\sigma(u) \in \alpha$. We observe that in this case $Z_0^{u\alpha}(y^u_{\sigma(u)}) = 0$, and further, for all $u' \in U$, $Z_0^{u\alpha}(y^{u'}_{\sigma(u')}) = 0$. And so C_U evaluates to 0 on $Z_0^{u\alpha}$, and therefore ϕ evaluates to 0 on $Z_0^{u\alpha}$.

Case 2. Let $\sigma(u) \notin \alpha$. Then $\forall v \in N(u)$, $\pi_{uv}(\sigma(v)) = \sigma(u) \notin \alpha$ by construction of the point $Z_0^{u\alpha}$. Therefore, for all $v \in N(u)$, $Z_0^{u\alpha}(x^v_{\sigma(v)}) = 0$, and moreover for all $v' \in V \setminus N(u)$, $Z_0^{u\alpha}(x^{v'}_{\sigma(v')}) = 0$. Therefore, C_V evaluates to 0 on $Z_0^{u\alpha}$ and therefore ϕ evaluates to 0 on $Z_0^{u\alpha}$.

Therefore, ϕ is consistent with all the data points of the form $(Z_0^{u\alpha}, 0)$.

From the above analysis we conclude that the 2-clause CNF formula ϕ is consistent with all the data points of $\mathcal{D}$.

A.3 NO Case

For the sake of contradiction we assume that there is a t clause CNF formula ϕ^* which is consistent with the data points with probability at least $\frac{1}{2} + \varepsilon$ for some given constants $\varepsilon, t > 0$. We will set the perturbation parameter $\mu = \frac{\varepsilon^2}{16t^3}$.

Let the given t clause CNF formula be $\phi^* = C_1 \wedge \cdots \wedge C_t$. We will first show that not all the clauses $C_1, \ldots, C_t$ can contain a negative literal.

From the construction of the data points it is easy to see that any given coordinate of $\{0,1\}^{\mathcal{S}}$ is set to 1 with probability at most $\frac{d_U + d_V}{\min\{|U|,|V|\}} = \xi(N) = o(1)$. Therefore, if all the clauses in ϕ^* had a negative literal, then ϕ^* would evaluate to 1 with probability at least $1 - t\xi(N) = 1 - o(1)$ over the distribution $\mathcal{D}$, which is a contradiction to the assumption that ϕ^* is consistent with the data points with probability at least $\frac{1}{2} + \varepsilon$ for constant $\varepsilon > 0$, since the 0 and 1 data points are equally likely in $\mathcal{D}$. This implies that there is a non empty subset Q of clauses of ϕ^*, such that none of the clauses in Q contains a negative literal. W.l.o.g. we may assume that $Q = \{C_1, \ldots, C_\ell\}$, where $\ell \leq t$. Moreover, the formula $\phi = C_1 \wedge \cdots \wedge C_\ell$ must be consistent with the data points of the oracle with probability at least $\frac{1}{2} + \varepsilon - t\xi(N) \geq \frac{1}{2} + \varepsilon/2$, for large enough size of instance. For the remainder of the argument we shall only consider the CNF ϕ and use it to construct a 'good' labeling to the vertices of the label cover.

Before proceeding we first define ℓ distinguished labels from $[k] \cup \{0\} : \{q_i^v\}_{i=1}^{\ell}$ for each $v \in V$. Let q_i^v be any arbitrary label $j \in [k]$ such that the positive literal x_j^v is present in clause C_i of ϕ, and 0 if there is no such variable in C_i. We call this setting of distinguished labels Γ.

Since ϕ is consistent with the data points of the oracle with probability at least $\frac{1}{2} + \frac{\varepsilon}{2}$, by an averaging argument we have that there is a set $U' \subseteq U$ such that $|U'| \geq \frac{\varepsilon}{4}|U|$, such that for every vertex $u \in U'$, ϕ is consistent with probability at least $\frac{1}{2} + \frac{\varepsilon}{4}$ with the data points generated by the oracle on picking u in step 1. Call such vertices $u \in U'$ as 'good'.

A.3.1 Analysis for a fixed 'good' vertex $u \in U'$. We now fix one such 'good' vertex u. The rest of the analysis is with respect to this 'good' vertex. Let $N(u)$ be its neighborhood. After picking u in step 1, the oracle outputs a 0 example and a 1 example with equal probability. Therefore, again by averaging, it must be the case that ϕ is consistent with the 1 examples (of u) with probability (over choice of $v \in N(u)$ in step 2a) at least $\frac{\varepsilon}{4}$; and consistent with the 0 examples (of u) with probability (over the choice of the set α in step 3a) at least $\frac{\varepsilon}{4}$.

Suppose C_i is a clause in ϕ for some $1 \leq i \leq \ell$, such that C_i contains a positive literal y^u_j for some $j \in [m]$. Then, C_i will be 0 with probability at most μ on the 0 examples of u. Therefore, by union bound, the probability that any of the clauses of ϕ containing a positive literal evaluates to 0 on the 0 examples is at most $t\mu$, which is at most $\frac{\varepsilon}{8}$ for our setting of the parameter μ. Therefore, there is a subformula ϕ_u of ϕ containing the clauses $\{C_i\}_{i \in L_u}$, where $L_u \subseteq [\ell]$, such that none of the clauses of ϕ_u contains a variable of the form y^u_j for $j \in [m]$, and moreover ϕ_u is consistent with the 0 examples with probability at least $\varepsilon' = \frac{\varepsilon}{4} - t\mu \geq \frac{\varepsilon}{8}$, and with the 1 examples also with probability at least ε'. The rest of the analysis will show that there is an appropriate clause in ϕ_u which gives a good labeling for a significant fraction of the vertices in $N(u)$.

Since ϕ_u is consistent with the 1 examples of u, with probability ε', there must be a set $M(u) \subseteq N(u)$ such that $|M(u)| \geq \varepsilon'|N(u)|$ and for every $v \in M(u)$, ϕ_u is 1 on the point Z^{uv}_1 constructed on choosing v in step 2a. Call such vertices 'good neighbors' of u. Since we have shown that ϕ_u does not contain any negative literal or any positive literal y^u_j for any $j \in [m]$ in any of its clauses, from the construction of the point Z^{uv}_1, this implies that every clause C_i ($i \in L_u$) contains a positive literal from the set $\{x^v_j\}^k_{j=1}$, for all 'good neighbors' v of u. So the setting q^v_i given by Γ, of distinguished labels for the 'good neighbors' v corresponding to the clauses C_i of ϕ_u is not 0.

We also have that with probability ε' over the sets α chosen in step 3a, ϕ_u is 0 on the points $Z^{u\alpha}_0$. This implies that there is a clause C_{i_u} of ϕ_u, for some $i_u \in L_u$, such that C_{i_u} is 0 on the points $Z^{u\alpha}_0$ with probability at least $\frac{\varepsilon'}{\ell}$. We have,

$$\Pr_{\alpha}[C_{i_u} \text{ is } 0 \text{ on } Z^{u\alpha}_0] \geq \frac{\varepsilon'}{\ell} \tag{4}$$

Now, since C_{i_u} is a clause of ϕ_u, it contains positive literals corresponding to all the 'good neighbors' $v \in M(u)$, and therefore $q^v_{i_u} \in [k]$ for all $v \in M(u)$. Define the set $T_u \subseteq [m]$ as,

$$T_u = \{\pi_{uv}(q^v_{i_u}) \mid v \in M(u)\}.$$

In other words, T_u is the subset of $[m]$ onto which the distinguished labels of the vertices $v \in M(u)$ corresponding to the clause C_{i_u} project. From the construction of the points $Z^{u\alpha}_0$, we have the following observation.

Observation 10 *If $\alpha \cap T_u \neq \emptyset$ then C_{i_u} is 1 on the point $Z^{u\alpha}_0$.*

We will show that the above observation implies that the set T_u cannot be too large. We have,

$$\begin{aligned}\Pr_{\alpha}[\alpha \cap T_u = \emptyset] &= (1-\mu)^{|T_u|} \\ &\geq \Pr_{\alpha}[C_{i_u} \text{ is } 0 \text{ on } Z^{u\alpha}_0]\end{aligned}$$

and combining the above with equation (4), we have,

$$\begin{aligned}(1-\mu)^{|T_u|} &\geq \frac{\varepsilon'}{\ell} \\ &\geq \frac{\varepsilon'}{t}\end{aligned}$$

Therefore,

$$|T_u| \leq \frac{1}{\mu} \ln \left(\frac{t}{\varepsilon'} \right).$$

For convenience let $\nu = \left(\frac{1}{\mu} \ln \left(\frac{t}{\varepsilon'} \right) \right)^{-1}$. Define,

$$\Lambda_j^u = \{ v \in M(u) \ \mid \ \pi_{uv}(q_{i_u}^v) = j \}$$

for all $j \in [m]$. Essentially, Λ_j^u is the subset of the 'good' neighbors v of u whose distinguished label corresponding to the clause C_{i_u} projects onto j. We have the following simple lemma.

Lemma 11 $\exists j_u \in T_u$ *such that* $|\Lambda_{j_u}^u| \geq \nu |M(u)|$.

Proof: Note that $M(u) = \bigcup_{j \in T_u} \Lambda_j^u$. And since $|T_u| \leq \frac{1}{\nu}$, the lemma follows. ■

A.3.2 Labeling.

We now define the labeling. The partial labeling $\sigma_V : V \mapsto [k]$ is constructed in a randomized manner as follows. For every vertex $v \in V$, choose i_v randomly from $\{1, \ldots, \ell\}$. If $q_{i_v}^v \in [k]$ then set $\sigma(v) = q_{i_v}^v$. Essentially, for every vertex v, we label it by its distinguished label (given by the setting Γ) corresponding to a random clause of ϕ (if the label is not 0).

We construct the partial labeling $\sigma_U : U \mapsto [m]$ as follows. For every 'good' vertex $u \in U'$, let $\sigma(u) = j_u$ as in lemma 11.

Now we analyze how many edges are satisfied by the partial assignment σ_V, σ_U. Let (u, v) be a random edge chosen by picking u randomly from U and then choosing v randomly from $N(u)$. With probability $\frac{\varepsilon}{4}$, u is a good vertex. With probability at least $\varepsilon' \nu$, the vertex v is selected from $\Lambda_{j_u}^u$, and with a further probability at least $\frac{1}{\ell} \geq \frac{1}{t}$, the vertex v is labeled with the label $q_{i_u}^v$ which projects onto j_u via the map π_{uv}. Therefore, the edge is satisfied with probability at least

$$\begin{aligned} p^* &= \left(\frac{\varepsilon}{4} \right) \varepsilon' \nu \left(\frac{1}{t} \right) \\ &\geq \left(\frac{\varepsilon}{4} \right) \left(\frac{\varepsilon}{8} \right) \nu \left(\frac{1}{t} \right) \end{aligned}$$

which, by the definition of ν and our choice of μ, is a constant depending only on ε and t. Since a random edge is satisfied with probability p^*, the expected fraction of edges satisfied is p^*. This implies that there must be a labeling that satisfies at least p^* fraction of the edges. Now, the soundness η of the Label Cover instance can be chosen arbitrarily small to obtain a contradiction.

B Hardness of Learning OR by t-DNF under adversarial noise

In this section we prove Theorem 3. For convenience, we prove an equivalent result for learning OR by a t-DNF under adversarial noise.

We start with an instance $\mathcal{L}$ of $\textsc{LabelCover}(m, k)$ consisting of the a bipartite graph $G(U, V, E)$, set of labels $[m]$ (for vertices in U), $[k]$ (for vertices in V), the projections $\pi_{uv} : [k] \mapsto [m]$ for every edge $e = (u, v) \in E$, where vertices in U have degree d_U, and those in V have degree d_V. All the parameters are constants independent of the sizes $N_U = |U|$ and $N_V = |V|$. Let $N = N_U + N_V$. Let the soundness parameter be η.

B.1 Construction

Variables. First we define the set of variables. Let v be any vertex in V. We have a set k variables, $S_v = \{x_i^v\}_{i=1}^k$ for every vertex $v \in V$, with one variable for every (potential) label for that vertex. Let,

$$\mathcal{S} = \bigcup_{v \in V} S_v$$

be the set of all variables. Let the corresponding boolean hypercube be $\{0, 1\}^{\mathcal{S}}$ where the coordinates are indexed by the variables in $\mathcal{S}$.

Distribution. We now describe how the oracle generates a sample point. This describes the distribution $\mathcal{D}$ on the samples. Let $\mu \in (0, 1)$ be a given parameter. Let $\ell > 0$ be a positive integer to be fixed later. On being queried for a sample, the oracle does the following,

1. Chooses a vertex $u \in U$ at random from the vertices in U. Let $N(u) \subseteq V$ be the neighborhood of u.
2. With probability $\frac{1}{2}$ does the following,

   2a. Picks $v \in N(u)$ at random.

   2b. Creates the following point $Z_1^u[v] \in \{0, 1\}^{\mathcal{S}}$ as follows,

   $$\forall j \in [k], \qquad Z_1^u[v](x_j^{v'}) = \begin{cases} 1 & \text{if } v' = v \\ 0 & \text{otherwise} \end{cases}$$

   2c. Output $(Z_1^u[v], 1)$ as a data point.
3. With probability $\frac{1}{2}$ does the following,

   3a. Picks a ℓ tuple $(v_1, \ldots, v_\ell)$ such that each v_i is chosen uniformly at random from $N(u)$ for $1 \leq i \leq \ell$.

   3b. Picks a set $\alpha \subseteq [m]$ by picking every element of $[m]$ independently at random with probability μ.

   3c. Creates the following point $Z_0^u[\alpha, (v_1, \ldots, v_\ell)]$ as follows,

   $$\forall j \in [k], \qquad Z_0^u[\alpha, (v_1, \ldots, v_\ell)](x_j^{v'}) = \begin{cases} 1 & \text{if for any } i \in [\ell], v' = v_i \text{ and } \pi_{uv'}(j) \in \alpha \\ 0 & \text{otherwise} \end{cases}$$

   3d. Outputs $(Z_0^u[\alpha, (v_1, \ldots, v_\ell)], 0)$ as a data point.

Note that the support of $\mathcal{D}$ is polynomial in the size of the label cover instance and hence $\mathcal{D}$ can be given explicitly. Let $t > 0$ be a given positive integer and $\varepsilon, \mu > 0$ be given parameters that may be arbitrarily small constants. We will show that if the instance of label cover $\mathcal{L}$ is a YES instance, i.e there is a labeling that satisfies all edges then there is a OR formula which is consistent with the samples with probability $1-\mu$. On the other hand, if the instance is a NO instance then there is no t-DNF formula that is consistent with the samples with probability $\frac{1}{2} + \varepsilon$ under the distribution $\mathcal{D}$ with the soundness η and the degrees d_U and d_V suitably chosen.

B.2 YES Case

Suppose the instance of label cover is a YES instance. In this case, there is a labeling σ to the vertices of the label cover that satisfies all the edges. Consider the following OR formula,

$$\phi = \bigvee_{v \in V} x^v_{\sigma(v)} \tag{5}$$

The formula ϕ contains one positive literal for every vertex $v \in V$, corresponding to the label assigned to v by σ. Clearly, ϕ is consistent with any 1 example $(Z^u_1[v], 1)$ generated by the oracle. This is because in $Z^u_1[v]$, the coordinates corresponding to all the labels of v are set to 1, and therefore the literal $x^v_{\sigma(v)}$ is 1 on $Z^u_1[v]$.

Now suppose the oracle selects a vertex $u \in U$ and then generates a 0 example $(Z^u_0[\alpha, (v_1, \ldots, v_\ell)], 0)$. In the point $Z^u_0[\alpha, (v_1, \ldots, v_\ell)]$ all the variables $x^v_{\sigma(v)}$ are set to 0 where $v \neq v_i$ for all $1 \leq i \leq \ell$. Now suppose $\sigma(u) \notin \alpha$. Then $x^{v_i}_{\sigma(v_i)}$ is set to 0 for all $1 \leq i \leq \ell$. Therefore, ϕ evaluates to 0 in this case. Now the probability that $\sigma(u) \notin \alpha$ is exactly $1 - \mu$, by the construction of the set α. Therefore, with probability at least $1 - \mu$, ϕ is consistent with the 0 examples of u.

The above analysis holds for any vertex u as the choice in step 1. Therefore, overall, ϕ is consistent with the data points of the verifier with probability at least $1 - \mu$.

B.3 NO Case

Suppose that the label cover instance is a NO instance, i.e. no labeling to the vertices of the label cover satisfies η fraction of the edges, where η is the soundness parameter which will be chosen to be small enough later. We assume that there is a t-DNF formula ϕ^* that is consistent with the examples of the oracle with probability at least $\frac{1}{2} + \varepsilon$, under the distribution $\mathcal{D}$. We have that,

$$\phi^* = \bigvee_{j=1}^{M} T_j \tag{6}$$

for some M, and each term T_j is the AND of at most t literals. Suppose there is a term T' of ϕ^* such that it is an AND of only negative literals. Now such a term will be 1 with probability at least $1 - \frac{td_U}{|V|}$, which would imply that ϕ^* is 1 with probability at least $1 - \frac{td_U}{|V|}$. Since the oracle outputs 0 and 1 examples equally often, this is a contradiction to the assumption that ϕ^* is consistent with the examples of the oracle with probability at least $\frac{1}{2} + \varepsilon$ for large enough $|V|$. Therefore, we may assume that every term of ϕ^* has at least one positive literal. We now make this simple observation.

Observation 12 *If a given term T_j is never 1 on any of the 1 examples of the oracle, then $\phi \setminus \{T_j\}$, is also consistent with the examples of the oracle with probability $\frac{1}{2} + \varepsilon$.*

This is because removing a term can hurt us only in the case of 1 examples, so we can remove all the terms that are never 1 on the 1 examples. This leads to the following simple lemma.

Lemma 13 *Let ϕ be the OR of all the terms T_j of ϕ^* such that for each T_j there is a vertex $v \in V$, such that all the positive literals in the term T_j are of the form x_i^v for some $1 \leq i \leq k$, and T_j does not contain any negative literal of the form $\bar{x}_{i'}^v$ for any $1 \leq i' \leq k$. Then ϕ is also consistent with the examples of the oracle with probability $\frac{1}{2} + \varepsilon$.*

Proof: Suppose there is a term T_j of ϕ such that it contains positive literals of the form $x_{i_1}^{v_1}$ and $x_{i_2}^{v_2}$, where $v_1 \neq v_2$ and $1 \leq i_1, i_2, \leq k$. Since all the 1 data points of the oracle have the property that all the coordinates that are set to 1 correspond to the variables x_i^v $1 \leq i \leq k$, for exactly one such vertex $v \in V$, the term T_j will be 0 on all such points as it contains positive literals corresponding to two different vertices.

Moreover, if T_j contains a positive literal of the form $x_{i_1}^v$ and a negative literal of the form $\bar{x}_{i_2}^v$, then again T_j will always be 0 on the 1 examples since in the 1 data points, for any vertex $v' \in V$, either all the coordinates corresponding to $\{x_i^{v'}\}_{i=1}^k$ are set to 1 or all of them are set to 0. Therefore, removing T_j does not hurt us in the 1 examples and clearly $\phi^* \setminus \{T_j\}$ is as good as ϕ^* on the 0 examples. Therefore, we can remove all such terms and obtain the t-DNF formula ϕ which is also consistent with the examples of the oracle with probability at least $\frac{1}{2} + \varepsilon$. ■

In the rest of the analysis we will use the formula ϕ to construct a good labeling for the vertices of the label cover.

Before proceeding, we will construct the following assignment of terms to vertices. For every vertex $v \in V$, let $T^v = T_{j'}$ be any arbitrary term of ϕ containing at least one positive literal of the form x_i^v. If no such term exists for v in ϕ let $T^v = 0$. Call this assignment Γ. Clearly Γ is well defined since, every term has at least one positive literal of the form x_i^v for exactly one $v \in V$. For every vertex $v \in V$, let us also define the set $W(v) := \{i \in [k] \mid x_i^v \text{ is a positive literal of } T^v\}$. As mentioned, all the positive literals of T^v are necessarily of the form x_i^v for $1 \leq i \leq k$. Therefore, unless $T^v = 0$ the set $W(v)$ is non empty. Rest of the analysis will be with respect to this assignment Γ.

Since ϕ is consistent with the data points of the oracle with probability at least $\frac{1}{2} + \varepsilon$, by an averaging argument we have that there is a set $U' \subseteq U$ such that $|U'| \geq \frac{\varepsilon}{2}|U|$, such that for every vertex $u \in U'$, ϕ is consistent with probability at least $\frac{1}{2} + \frac{\varepsilon}{2}$ with the data points generated by the oracle on picking u in step 1. Call such vertices $u \in U'$ as 'good'. We fix one such 'good' vertex u and do the analysis for the 0 and 1 examples output by the oracle after choosing u in the initial step.

B.3.1 Analysis for a fixed 'good' vertex $u \in U'$.

Let $N(u)$ be the neighborhood of u. After picking u in step 1, the oracle outputs a 0 example and a 1 example with equal probability. Therefore, again by averaging, it must be the case that ϕ is consistent with the 1 examples (of u) with probability (over choice of $v \in N(u)$ in step 2a) at least $\frac{\varepsilon}{2}$; and consistent with the 0 examples (of u) with probability (over the choice of the set α, and the ℓ tuple $(v_1, \ldots, v_\ell)$ in step 3a and 3b) at least $\frac{\varepsilon}{2}$. For convenience, let $\varepsilon' = \frac{\varepsilon}{2}$.

Since ϕ is consistent with the 1 examples with probability at least ε', this implies that ϕ is 1 on the points $Z_1^u[v]$ for at least ε' fraction of the neighbors $v \in N(u)$. Let the set of such vertices v be $M(u)$, where $|M(u)| \geq \varepsilon'|N(u)|$, and call such v as 'good neighbors' of u. We have shown that ϕ does not have any term with all negative literals, and every term of ϕ must contain positive literals, all of them from exactly one vertex of V. Since the only coordinates of $Z_1^u[v]$ that are set to 1 correspond to x_i^v for $1 \leq i \leq k$, it must be that for every 'good neighbor' v, there is a term of ϕ containing positive literals only of the form x_i^v for some $1 \leq i \leq k$. This implies that for such vertices v, $T^v \neq 0$ and $W(v) \neq \emptyset$ in our setting Γ.

Consider an ℓ-tuple $\bar{v} = (v_1, \ldots, v_\ell)$ chosen randomly by choosing every v_i uniformly at random from $N(u)$. Let $D_{\bar{v}} := \{r \in [\ell] \mid v_r \in M(u)\}$. Essentially, $D_{\bar{v}}$ is the set of indices r such that v_r is a 'good' vertex. We call $\bar{v}$ as 'dense' if $|D_{\bar{v}}| \geq \frac{\varepsilon'}{2}\ell$. Since each coordinate of $\bar{v}$ is chosen uniformly at random from $N(u)$ and $|M(u)| \geq \varepsilon'|N(u)|$, we expect $\bar{v}$ to be 'dense' with high probability. Indeed, using Chernoff bound, we have,

$$\Pr_{\bar{v}}[\bar{v} \text{ is not dense }] \leq \exp(-\varepsilon'\ell/8)$$

Consider the ordered pair (r_1, r_2) such that $1 \leq r_1 \neq r_2 \leq \ell$. Call such a pair intersecting for an ℓ-tuple $\bar{v}$ if $T^{v_{r_1}}$ contains a literal of the form $\bar{x}_i^{v_{r_2}}$ for some $1 \leq i \leq k$. Now, the number of literals in $T^{v_{r_1}}$ is at most t. And since v_{r_2} is chosen independently at random from $N(u)$, we have,

$$\Pr_{\bar{v}}[(r_1, r_2) \text{ is intersecting }] \leq \frac{t}{d_U}$$

for every $1 \leq r_1 \neq r_2 \leq \ell$. Call $\bar{v}$ intersection-free, if it contains no intersecting pair of coordinates. Since, there are ℓ^2 such pairs,

$$\Pr_{\bar{v}}[\bar{v} \text{ is not intersection-free }] \leq \frac{t\ell^2}{d_U}.$$

Now ϕ is consistent with the 0 examples with probability at least ε'. Again, by averaging, we have that for $\frac{\varepsilon'}{2}$ of the ℓ-tuples $\bar{v}$, ϕ is 0 on the points $Z_0^u[\alpha, \bar{v}]$ generated after choosing $\bar{v}$ in step 3a, with probability at least $\frac{\varepsilon'}{2}$. We call such ℓ-tuples $\bar{v}$ as 'good'. More formally we have,

$$\Pr_{\bar{v}}[\bar{v} \text{ is 'good' }] \geq \frac{\varepsilon'}{2},$$

where, for a given 'good' ℓ-tuple $\bar{v}$,

$$\Pr_{\alpha}[\phi \text{ is 0 on } Z_0^u[\alpha, \bar{v}] \geq \frac{\varepsilon'}{2}.$$

Using union bound, we have,

$$\Pr_{\bar{v}}[\bar{v} \text{ is good, dense and intersection free}] \geq \nu \tag{7}$$

where,

$$\nu = \frac{\varepsilon'}{2} - \exp(-\varepsilon'\ell/8) - \frac{t\ell^2}{d_U}. \tag{8}$$

We now fix a good, dense and intersection-free ℓ-tuple $\bar{v} = (v_1, \ldots, v_t)$. Consider any $r \in D_{\bar{v}}$, $v_r \in M(u)$ and so $T^{v_r} \neq 0$. Moreover, since $\bar{v}$ is intersection free, the negative literals in T^{v_r} correspond to vertices that are not contained in any coordinate of $\bar{v}$. Therefore, the negative literals in T^{v_r} are always set to 1 on the points $Z_0^u[\alpha, \bar{v}]$ for any $\alpha \subseteq [m]$. Therefore, the term T^{v_r} will be 1 if all the variables (positive literals) in T^{v_r} are set to 1, which happens if $\pi_{uv_r}(W(v_r)) \subseteq \alpha$. This leads to the following key lemma.

Lemma 14 *If* $\ell > \left(\frac{2}{\varepsilon'\mu^t}\right)\ln\left(\frac{2}{\varepsilon'}\right)$ *, then there must exist* $r_1, r_2 \in D_{\bar{v}}$, $r_1 \neq r_2$ *such that* $\pi_{uv_{r_1}}(W(v_{r_1})) \cap \pi_{uv_{r_2}}(W(v_{r_2})) \neq \emptyset$.

Proof: Assume that there is no such pair r_1 and r_2. Therefore, the events $(\pi_{uv_r}(W(v_r)) \subseteq \alpha)$ are independent events for $r \in D_{\bar{v}}$. From the discussion above, we have for any $r \in D_{\bar{v}}$,

$$\begin{aligned}\Pr_{\alpha}[T^{v_r} \text{ is 1 on } Z_0^u[\alpha, \bar{v}]] &= \Pr_{\alpha}[\pi_{uv_r}(W(v_r)) \subseteq \alpha] \\ &= \mu^{|\pi_{uv_r}(W(v_r))|} \\ &\geq \mu^t \end{aligned} \tag{9}$$

Therefore, we have,

$$\begin{aligned}\Pr_{\alpha}[\phi \text{ is 0 on } Z_0^u[\alpha, \bar{v}]] &\leq \Pr_{\alpha}\left[\bigwedge_{r=1}^{|D_{\bar{v}}|}(T^{v_r} \text{ is 0 on } Z_0^u[\alpha, \bar{v}])\right] \\ &= \Pr_{\alpha}\left[\bigwedge_{r=1}^{|D_{\bar{v}}|}(W(v_r) \not\subseteq \alpha)\right]\end{aligned}$$

and combining the independence of the events $(\pi_{uv_r}(W(v_r)) \subseteq \alpha)$ with equation (9), we obtain,

$$\Pr_{\alpha}[\phi \text{ is 0 on } Z_0^u[\alpha, \bar{v}]] \leq (1 - \mu^t)^{|D_{\bar{v}}|}$$

Now, since the left hand side is at least $\frac{\varepsilon'}{2}$, the above implies,

$$\begin{aligned} |D_{\bar{v}}| &\leq \left(\frac{1}{\mu^t}\right) \ln\left(\frac{2}{\varepsilon'}\right) \\ \Rightarrow \quad \ell &\leq \left(\frac{2}{\varepsilon'\mu^t}\right) \ln\left(\frac{2}{\varepsilon'}\right) \end{aligned} \tag{10}$$

since $\bar{v}$ is dense. This proves the lemma.

■

In our construction we choose ℓ large enough depending on μ, ε and t, and then (independently of ℓ), choose d_U large enough (by parallel repetition), to ensure the following.

$$\ell > \left(\frac{2}{\varepsilon'\mu^t}\right) \ln\left(\frac{2}{\varepsilon'}\right) \tag{11}$$

and

$$\nu \geq \frac{\varepsilon'}{4} \tag{12}$$

Note that the above analysis holds for any valid assignment Γ of terms to vertices. We are now ready to define a labeling to the vertices of the label cover.

B.3.2 Construction of labeling We will define a partial labeling σ_U, σ_V to the vertices in U and V respectively in the following randomized manner.

1. Let $u \in U$ be any given vertex. Choose a random vertex v' from $N(u)$. If $W(v') = \emptyset$ then do not assign any label to u. If not, select $i \in W(v')$ randomly, and let $\sigma_U(u) = \pi_{uv'}(i)$.
2. Let $v \in V$ be any vertex. If $W(v)$ is empty then do not assign any label to v, otherwise, let $\sigma_V(v) = i$ where i is randomly chosen from $W(v)$.

We will now analyze how many edges this labeling satisfies in expectation. Consider a random edge (u, v) of the label cover, selected by first choosing v randomly from U and then selecting v randomly from $N(u)$. Now, u is labeled by choosing a vertex v' at random from $N(u)$ and labeling u by $\pi_{uv'}(i)$ where i is chosen randomly from $W(v')$, unless $W(v') = \emptyset$. Therefore, the probability that a random edge (u, v) is satisfied is same as the probability that $\pi_{uv}(\sigma_V(v)) = \pi_{uv'}(\sigma_V(v'))$ where v and v' are vertices selected uniformly at random from $N(u)$.

With probability $\frac{\varepsilon}{2}$, u is a 'good' vertex. Also, choosing two neighbors of u uniformly at random is same as choosing a random ℓ-tuple $\bar{v}$ (for $\ell \geq 2$) and then selecting two distinct coordinates of $\bar{v}$. In this process, with probability ν, a good, dense and intersection-free ℓ-tuple $\bar{v}$ is picked. From our choice of ℓ depending on μ, ε and t, we have the bounds given by (11), and (12) and combining with Lemma 14, we have that with probability $\frac{1}{\ell^2}$, the vertices v and v' are such that $\pi_{uv}(W(v)) \cap \pi_{uv'}(W(v')) \neq \emptyset$. And with further $\frac{1}{t^2}$ probability the labels for v and v' are consistent, i.e. $\pi_{uv}(\sigma_V(v)) = \pi_{uv'}(\sigma_V(v'))$.

Combining everything we have that the probability that a random edge (u, v) is satisfied is,

$$\Delta = \left(\frac{\varepsilon}{2}\right) \nu \left(\frac{1}{\ell^2}\right) \left(\frac{1}{t^2}\right).$$

Now, since $\nu \geq \frac{\varepsilon'}{4} \geq \frac{\varepsilon}{8}$ and ℓ is chosen to depend only on μ, ε and t, the above probability depends only on μ, ε and t. Also, it implies that there is a labeling that satisfies Δ fraction of edges of the label cover, where Δ depends only on μ, ε and t. By choosing the soundness parameter of the NO instance η to be small enough, we obtain a contradiction.

C Proof of Theorem 7

In this section we will construct an instance of t-LAYERED-CSP we require for our reduction. First we will construct an appropriate PCP and then we will transform the PCP to a Multi Prover System with some desired properties. The Multi Prover System thus constructed can be thought of as an instance of t-LAYERED-CSP in a natural way.

We begin with the construction of the PCP. Our construction is very similar to the query efficient PCP constructed in [Kho01]. We start with an instance of MAX-3LIN and construct the Raz verifier using parallel repetition. The proofs are then encoded using Hadamard Codes. In order to obtain a PCP with a large alphabet, we take the encoding using Hadamard Codes over a large field. The analysis is similar to [Kho01] and relies heavily on the techniques developed in [ST98] and [ST00] and a similar construction over finite abelian groups in [Eng00].

We start with the instance of MAX-3LIN constructed in [KP06] with completeness $1 - \left(2^{-\Omega(\sqrt{\log n})}\right)$ and soundness $1 - \Omega\left(\log^{-3} n\right)$. They prove the following theorem.

Theorem 15 *Given a 7-regular instance $\mathcal{A}$ of* MAX-3LIN *over $\mathbb{F}[2]$ on n variables such that unless* NP $\subseteq$ DTIME$(2^{O(\log^2 N)})$ *there is no polynomial time algorithm to distinguish between the following two cases,*

YES CASE. There is an assignment to the variables of $\mathcal{A}$ that satisfies $1 - 2^{-\Omega(\sqrt{\log n})}$ fraction of the equations.

NO CASE. No assignment to the variables of $\mathcal{A}$ satisfies more than $1 - \Omega(\log^{-3} n)$ fraction of the equations.

Note that the equations of MAX-3LIN are over $\mathbb{F}[2]$. However, we may consider them to be over $\mathbb{F}[2^r]$ where r is some parameter and still the above theorem still holds. This is because the additive group $(\mathbb{F}[2^r], +)$ is isomorphic to $(\mathbb{F}[2]^r, +)$. Therefore, we can substitute the equation $x_1 + x_2 + x_3 = b$, where $x_1, x_2, x_3, b \in \mathbb{F}[2]$ with the equation over $\mathbb{F}[2^r]$, $x_1' + x_2' + x_3' = b_r$, where b_r is the element of $\mathbb{F}[2]^r$ with b in each or the r coordinates. Clearly, any assignment over $\mathbb{F}[2]$ can be extended to an assignment over $\mathbb{F}[2]^r$ by replicating it in every coordinate. Moreover, any assignment over $\mathbb{F}[2]^r$ that satisfies a particular equation must satisfy it in every coordinate, and so we can pick any coordinate and the corresponding assignment over $\mathbb{F}[2]$ will also satisfy all equations satisfied earlier. And since $(\mathbb{F}[2^r], +) \cong (\mathbb{F}[2]^r, +)$, we can write the entire system of equations over the field $\mathbb{F}[2^r]$.

C.1 Raz Verifier

We construct the Raz Verifier starting with an instance $\mathcal{A}$ of MAX-3LIN obtained from Theorem 15. For convenience we let the completeness of $\mathcal{A}$ be $1 - c(n)$ and soundness be $1 - s(n)$. The construction in [ST00] started with a GAP-3SAT instance, however we require constraints to be linear to be able to use Hadamard Codes instead of Long Codes, similar to the construction in [Kho01]. Note that our instance $\mathcal{A}$ of MAX-3LIN is over the field $\mathbb{F}[2^r]$ for some $r > 0$ to be fixed later, and has the same completeness and soundness as in Theorem 15.

Let $m > 0$ be a parameter to be fixed later. The Raz Verifier is given an instance $\mathcal{A}$ of MAX-3LIN. It expects two proofs, P and Q. The proof P is supposed to contain, for every set U of m variables, a length m vector $P(U)$ over $\mathbb{F}[2^r]$ giving the assignment to the variables in U. Similarly, for every set W of m equations, $Q(W)$ is supposed to be a length $3m$ vector giving the assignment to the $3m$ variables in the set of equations W.

The verifier works by picking a set of $U = (x_i)_{i=1}^m$ of m variables and then picking a set of m equations $W = (C_i)_{i=1}^m$ where each equation C_i is selected randomly from the constantly many equations containing the variable x_i. The verifier reads $P(U)$ and $Q(W)$ from the proof and accepts iff $Q(W)$ satisfies all the equations $(C_i)_{i=1}^m$ and the values of the variables $(x_i)_{i=1}^m$ in $P(U)$ and $Q(W)$ are the same (call this projection test).

Completeness. In the YES case $\mathcal{A}$ has an assignment that satisfies $1 - c(n)$ fraction of the equations. Let both proofs P and Q be consistent with that assignment. Since, the instance $\mathcal{A}$ is regular, with probability at least $(1 - c(n))^m$ all the equations $W = (C_i)_{i=1}^m$ chosen in the construction above will be satisfied by the proof Q. Therefore, the completeness is at least $(1 - c(n))^m \geq (1 - mc(n))$.

Soundness. In the NO case any assignment to the variables of $\mathcal{A}$ satisfies at most $1 - s(n)$ fraction of the equations. Using Raz's Parallel Repetition Theorem [Raz98], and the fact that each equation contains exactly 3 variables, we have the following upper bound.

Theorem 16 *There is a an absolute constant $\kappa > 0$ such that, the soundness of the Raz Verifier on the instance of* MAX-3LIN *(over $\mathbb{F}[2^r]$) with soundness $(1 - s(n))$ is at most $(1 - s(n)^\kappa)^{(m/(\kappa r))}$.* [3]

C.2 Fourier Analysis

We will be working over the field $\mathbb{F}[2^r]$ for $r > 0$, which is a field extension of $\mathbb{F}[2]$. Let φ be the isomorphism from the additive group $(\mathbb{F}[2^r], +)$ to $(\mathbb{F}[2]^r, +)$. Define the following homomorphism ϕ from $(\mathbb{F}[2^r], +)$ to the multiplicative group $(\{-1, 1\}, .)$.

$$\phi(a) = \begin{cases} 1 & \text{if } \varphi(a) \text{ contains even number of 1s} \\ -1 & \text{otherwise} \end{cases}$$

[3] Since in our case the constraints are projections, using Rao's [Rao08] proof of parallel repetition we can eliminate the dependence over r.

for any $a \in \mathbb{F}[2^r]$. We now define the 'characters' $\psi_a : \mathbb{F}[2^r] \mapsto \{-1, 1\}$ for $a \in \mathbb{F}[2^r]$ as follows.

$$\psi_a(b) := \phi(ab)$$

The characters ψ_a satisfy the following properties.

$$\begin{aligned} \psi_0(b) &= 1 && \forall b \in \mathbb{F}[2^r] \\ \psi_a(0) &= 1 && \forall a \in \mathbb{F}[2^r] \\ \psi_{a+b}(c) &= \psi_a(c)\psi_b(c) \end{aligned}$$

and,

$$\sum_{a \in \mathbb{F}[2^r]} \psi_a(b) = \begin{cases} |\mathbb{F}[2^r]| & \text{if } b = 0 \\ 0 & \text{otherwise} \end{cases}$$

We note that the 'character' functions form an orthonormal basis for the space $L^2(\mathbb{F}[2^r])$. We have that,

$$\langle \psi_a, \psi_b \rangle = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$$

where,

$$\langle \psi_a, \psi_b \rangle := \mathrm{E}_{c \in \mathbb{F}[2^r]} \left[\psi_a(c) \psi_b(c) \right].$$

We now consider the vector space $\mathbb{F}[2^r]^m$ for some positive integer m. We define the 'characters' $\chi_\alpha : \mathbb{F}[2^r]^m \mapsto \{-1, 1\}$ for every $\alpha \in \mathbb{F}[2^r]^m$ as,

$$\chi_\alpha(f) := \phi(\alpha \cdot f), \qquad f \in \mathbb{F}[2^r]^m$$

where '$\cdot$' is the inner product in the vector space $\mathbb{F}[2^r]^m$. From the way we defined the characters ψ_a, we have,

$$\chi_\alpha(f) = \prod_{i=1}^{m} \psi_{\alpha_i}(f_i),$$

where α_i and f_i are the i^{th} coordinates of α and f respectively. The characters χ_α satisfy the following properties,

$$\begin{aligned} \chi_0(f) &= 1 && \forall f \in \mathbb{F}[2^r]^m \\ \chi_\alpha(0) &= 1 && \forall \alpha \in \mathbb{F}[2^r]^m \\ \chi_{\alpha+\beta}(f) &= \chi_\alpha(f)\chi_\beta(f) \\ \chi_\alpha(f+g) &= \chi_\alpha(f)\chi_\alpha(g) \end{aligned}$$

and,

$$\mathrm{E}_{f \in \mathbb{F}[2^r]^m} \left[\chi_\alpha(f) \right] = \begin{cases} 1 & \text{if } \alpha = 0 \\ 0 & \text{otherwise} \end{cases}$$

The characters χ_α form an orthonormal basis for $L^2(\mathbb{F}[2^r]^m)$. We have,

$$\langle \chi_\alpha, \chi_\beta \rangle = \begin{cases} 1 & \text{if } \alpha = \beta \\ 0 & \text{otherwise} \end{cases}$$

where,

$$\langle \chi_\alpha, \chi_\beta \rangle := \mathrm{E}_{f \in \mathbb{F}[2^r]^m} \left[\chi_\alpha(f) \chi_\beta(f) \right].$$

Let $A : \mathbb{F}[2^r]^m \mapsto \mathbb{F}[2^r]$ be a function. We define $\widehat{A}_{\gamma,\alpha}$ to be the Fourier coefficient of the function $\psi_\gamma \circ A$ corresponding to the element χ_α of the basis, for $\alpha \in \mathbb{F}[2^r]^m$ and $\gamma \in \mathbb{F}[2^r]$. Formally,

$$\widehat{A}_{\gamma,\alpha} = \langle \psi_\gamma \circ A, \chi_\alpha \rangle = \mathrm{E}_{f \in \mathbb{F}[2^r]^m} \left[\psi_\gamma(A(f)) \chi_\alpha(f) \right]$$

and therefore,

$$\psi_\gamma \circ A = \sum_{\alpha \in \mathbb{F}[2^r]^m} \widehat{A}_{\gamma,\alpha} \chi_\alpha.$$

The following is a useful lemma.

Lemma 17 *Let $A : \mathbb{F}[2^r]^m \mapsto \mathbb{F}[2^r]$ be a function such that $\exists h \in \mathbb{F}[2^r]^m$ and $\zeta \in \mathbb{F}[2^r]$ such that $A(f + \delta h) = A(f) + \delta\zeta$, for all $\delta \in \mathbb{F}[2^r]$. Then, if $\widehat{A}_{\gamma,\alpha} \neq 0$ for some $\alpha \in \mathbb{F}[2^r]^m$ and $\gamma \in \mathbb{F}[2^r]$, then $\alpha \cdot h = \gamma\zeta$.*

Proof: We have,

$$\begin{aligned} \widehat{A}_{\gamma,\alpha} &= \langle \psi_\gamma \circ A, \chi_\alpha \rangle \\ &= \mathrm{E}_{f \in \mathbb{F}[2^r]^m} \left[\psi_\gamma(A(f)) \chi_\alpha(f) \right] \\ &= \mathrm{E}_{f \in \mathbb{F}[2^r]^m} \left[\psi_\gamma(A(f + \delta h)) \chi_\alpha(f + \delta h) \right] \end{aligned}$$

for any $\delta \in \mathbb{F}[2^r]$. Therefore using the property of A we have,

$$\begin{aligned} \widehat{A}_{\gamma,\alpha} &= \mathrm{E}_{f \in \mathbb{F}[2^r]^m} \left[\psi_\gamma(A(f) + \delta\zeta) \chi_\alpha(f) \chi_\alpha(\delta h) \right] \\ &= \mathrm{E}_{f \in \mathbb{F}[2^r]^m} \left[\psi_\gamma(A(f)) \psi_\gamma(\delta\zeta) \chi_\alpha(f) \chi_\alpha(\delta h) \right] \\ &= \psi_\gamma(\delta\zeta) \chi_\alpha(\delta h) \mathrm{E}_{f \in \mathbb{F}[2^r]^m} \left[\psi_\gamma(A(f)) \chi_\alpha(f) \right] \\ &= \psi_\gamma(\delta\zeta) \chi_\alpha(\delta h) \widehat{A}_{\gamma,\alpha} \end{aligned}$$

and since $\widehat{A}_{\gamma,\alpha} \neq 0$, this implies,

$$\begin{aligned} & \psi_\gamma(\delta\zeta) = \chi_\alpha(\delta h) \\ \Rightarrow \quad & \phi(\delta\gamma\zeta) = \phi(\alpha \cdot (\delta h)) \\ \Rightarrow \quad & \phi(\delta(\gamma\zeta)) \phi(\delta(\alpha \cdot h)) = 1 \\ \Rightarrow \quad & \phi(\delta(\gamma\zeta + \alpha \cdot h)) = 1 \end{aligned}$$

for all $\delta \in \mathbb{F}[2^r]$. But since $\phi \not\equiv 1$, we must have that $\gamma\zeta + \alpha \cdot h = 0$, i.e. $\gamma\zeta = \alpha \cdot h$. This completes the proof. ■

Hadamard Codes. In the construction of the PCP, the prover expects the Hadamard encodings of the vectors $P(U)$ and $Q(W)$ for the sets U and W in the construction of the Raz Verifier.

Definition 7 *For any positive integer t, the Hadamard Code of $p \in \mathbb{F}[2^r]^t$ is given by a function $Had_p : \mathbb{F}[2^r]^t \mapsto \mathbb{F}[2^r]$ where,*

$$Had_p(a) = p \cdot a$$

for all $a \in \mathbb{F}[2^r]^t$.

Note that the string $x = Q(W)$, $x \in \mathbb{F}[2^r]^{3m}$ that the Raz Verifier reads is supposed to satisfy certain linear constraints over $\mathbb{F}[2^r]$, given by $h_i \cdot x = \zeta_i$, where $h_i \in \mathbb{F}[2^r]^{3m}$ and $\zeta_i \in \mathbb{F}[2^r]$ for $1 \leq i \leq m$.

Let $\pi : \mathbb{F}[2^r]^{3m} \mapsto \mathbb{F}[2^r]^m$ be a *projection* that maps vectors in $\mathbb{F}[2^r]^{3m}$ to some fixed m coordinates. Let $\pi^{-1}(a)$ denote the unique vector $b \in \mathbb{F}[2^r]^{3m}$ such that $\pi(b) = a$ and is 0 on all other coordinates other than those that are projected by π.

Folding. Let B_x be the Hadamard Code of a vector $x \in \mathbb{F}[2^r]^{3m}$ that satisfies the constraints $h_i \cdot x = \zeta_i$ for $1 \leq i \leq m$. Let H be the subspace of $\mathbb{F}[2^r]^{3m}$ spanned by $\{h_i\}_{i=1}^m$. Let $h \in H$ be such that $h = \sum_{i=1}^m \rho_i h_i$, where $\rho_i \in \mathbb{F}[2^r]$ for $1 \leq i \leq m$. Then, we have that for any $a \in \mathbb{F}[2^r]^{3m}$,

$$B_x(a + h) = B_x(a) + \sum_{i=1}^{m} \rho_i \zeta_i.$$

So, we can enforce the folding over linear constraints in the following manner. For any $a \in \mathbb{F}[2^r]^{3m}$, let,

$$a = v_a + \sum_{i=1}^{m} \rho_i h_i$$

where v_a is the lexicographically smallest vector in the coset $a + H$. The verifier expects a function $B' : \mathbb{F}[2^r]^{3m} \mapsto \mathbb{F}[2^r]$ defined only on the distinguished vectors v_a for the coset $a + H$, and then computes the value of $B(a)$ as follows,

$$B(a) = B'(v_a) + \sum_{i=1}^{m} \rho_i \zeta_i.$$

We say that B is 'folded' over the linear constraints. Therefore, we can enforce the folding of the supposed Hadamard encodings of the assignments $Q(W)$, over the linear constraints given by the equations in W. The following crucial lemma follows from directly from Lemma 17.

Lemma 18 *For any $\gamma \in \mathbb{F}[2^r]$, if $\widehat{B}_{\gamma,\beta} \neq 0$ for some $\beta \in \mathbb{F}[2^r]^{3m}$, then $\beta \cdot h_i = \gamma \zeta_i$ for all $1 \leq i \leq m$.*

Eventually our analysis will show that the supposed Hadamard Code B for $Q(W)$ can be decoded to obtain the vectors β with probability proportional to $\widehat{B}_{\gamma,\beta}^2$. Since we have ensured the folding, Lemma 18 would imply that $\gamma^{-1}\beta$ is a valid assignment to the variables in $Q(W)$ that satisfies all the linear constraints.

C.3 Construction of the PCP

We now construct the PCP verifier. The verifier V_{lin} is given an instance of MAX-3LIN over $\mathbb{F}[2^r]$ with the completeness and soundness parameters as before. The verifier expects proofs (P', Q') which are Hadamard encodings of the proofs (P, Q) given to the Raz Verifier. For sets U and W of the Raz Verifier, $P'(U)$ and $Q'(W)$ are supposed to be Hadamard codes of $P(U)$ and $Q(W)$ respectively. The verifier V_{lin} proceeds as follows,

1. Pick a set U of m variables and ℓ sets $(W_j)_{j=1}^{\ell}$ independently in a manner similar to the Raz Verifier. Let π_j be the projection function between W_j and U for $1 \leq j \leq \ell$.

2. Let A be the supposed Hadamard Code of $P(U)$ and B_j be the supposed Hadamard code of $Q(W_j)$. The codes B_j are assumed to be folded over the linear constraints.

3. Pick $a_1, \ldots, a_\ell \in \mathbb{F}[2^r]^m$ and $b_1, \ldots, b_\ell \in \mathbb{F}[2^r]^{3m}$ randomly.

4. Accept iff for $1 \leq i, j, \leq \ell$

$$A(a_i) + B_j(b_j) = B_j(\pi_j^{-1}(a_i) + b_j).$$

The following is the main theorem about the properties of this PCP.

Theorem 19 *Given an instance $\mathcal{A}$ of* MAX-3LIN *over n variables with completeness $1 - c(n)$ and soundness $1 - s(n)$,*

1. *V_{lin} uses $m \log n + O(\ell m r)$ random bits.*
2. *V_{lin} queries $\ell^2 + 2\ell$ positions from the proof.*
3. *If the instance $\mathcal{A}$ is a YES instance then there is a set $\bar{S}$ consisting of all the positions of the supposed encodings of $Q(W')$ for at most $mc(n)$ fraction of sets W', and an assignment τ^* to all the positions of the proof except those in $\bar{S}$ such that,*

   a. *(Strong Completeness) The verifier accepts on τ^* whenever none of the positions in $\bar{S}$ are queried.*

   b. *(Extendability) For any constraint q of the verifier which (possibly) queries positions from $\bar{S}$, there is an assignment τ_q to the positions in $\bar{S}$ queried in q, such that τ^* extended by τ_q satisfies the constraint q.*

4. *If the instance $\mathcal{A}$ is a NO instance then the probability that the verifier accepts is at most $|\mathbb{F}[2^r]|^{-\ell^2} + \delta$, for $\delta^2 = (1 - s(n)^\kappa)^{(m/(\kappa r))}(|\mathbb{F}[2^r]| - 1)^{\ell^2}$, for some universal constant κ.*

Proof: Properties 1 and 2 of verifier are clear. Assume that the MAX-3LIN $\mathcal{A}$ was a YES instance and had an assignment σ to the variables such that $1 - c(n)$ fraction of the equations were satisfied. Call the equations not satisfied as 'bad'. Therefore, at most $mc(n)$ fraction of the sets W' of the Raz Verifier are 'bad' i.e. they contain a 'bad' equation. Let the assignments $P(U)$ and $Q(W)$ be consistent with σ and $P'(U)$ and $Q'(W)$ be the respective Hadamard encodings given to verifier V_{lin}, for all sets of variables U and all sets W that are not 'bad'. We let the set $\bar{S}$ of positions in the proof correspond to the supposed Hadamard encodings of the assignment to the 'bad' sets W'.

Let U and $(W_j)_{j=1}^{\ell}$ be such that none of the W_j are 'bad', and A and $(B)_{j=1}^{\ell}$ be the Hadamard encodings of the assignments given by σ, which is a satisfying assignment for the sets U and $(W_j)_{j=1}^{\ell}$.

$$A(a_i) = a_i \cdot P(U) \qquad B_j(b_j) = b_j \cdot Q(W_j)$$

$$\begin{aligned} B_j(\pi_j^{-1}(a_i) + b_j) &= (\pi_j^{-1}(a_i) + b_j) \cdot Q(W_j) \\ &= (\pi_j^{-1}(a_i) \cdot Q(W_j) + b_j \cdot Q(W_j) \\ &= a_i \cdot \pi_j(Q(W_j)) + b_j \cdot Q(W_j) \\ &= A(a_i) + B_j(b_j) \end{aligned} \tag{13}$$

since $\pi_j(Q(W_j)) = P(U)$ as σ satisfies U and W_j. This proves the Strong Completeness property. Observe that every constraint of the verifier is a set of linear equalities of the form $A(a_i) = B(b_j) + B(\pi_j^{-1}(a_i) + b_j)$. Also, for $a_i \neq a_{i'}$, $\pi_j^{-1}(a_i) - \pi_j^{-1}(a_{i'}) \notin H^j$, where H^j is the subspace spanned by the linear constraints over which the supposed encoding B_j is folded. Therefore, if $a_i \neq a_{i'}$ then $B_j(\pi_j^{-1}(a_i) + b_j)$ and $B_j(\pi_j^{-1}(a_{i'}) + b_j)$ are distinct positions in the B_j. So, within any constraint every equation has a unique variable. Then, if B_j is an encoding corresponding to a 'bad' set W'_j, the proof Q' can be extended to satisfy equations involving positions in B_j, in a given constraint involving B_j. This implies that for any given constraint (possibly involving positions from $\bar{S}$), the encodings P', Q'given by σ, can be extended to the positions in $\bar{S}$ queried by the given constraint so that the constraint is satisfied. This proves the Extendability property, and completes the analysis for the YES case.

We now analyze the NO case. We assume that the verifier accepts with probability $|\mathbb{F}[2^r]|^{-\ell^2} + \delta$. It was shown in [Eng00] that the probability of acceptance of the verifier is,

$$\frac{1}{|\mathbb{F}[2^r]|^{\ell^2}} \sum_{S \subseteq [\ell] \times [\ell]} \mathrm{E}\left[T_S\right] \tag{14}$$

where,

$$T_S = \prod_{(i,j) \in S} \left(\sum_{\gamma \in \mathbb{F}[2^r] \setminus \{0\}} \psi_\gamma(A(a_i) + B_j(b_j) + B_j(\pi^{-1}(a_i) + b_j)) \right), \tag{15}$$

and the expectation is over the choice of U, $(W_j)_{j=1}^{\ell}$, $(a_i)_{i=1}^{\ell}$, $(b_j)_{j=1}^{\ell}$ and where $T_\emptyset = 1$.

If the above probability is $|\mathbb{F}[2^r]|^{-\ell^2} + \delta$, there must be a nonempty set $S \subseteq [\ell] \times [\ell]$, such that $|\mathrm{E}[T_S]| \geq \delta$. This term was analyzed in [Eng00] and we use their analysis. In [Eng00], since Long Codes are analyzed, the notion of projections is slightly different from ours, but the proof is exactly the same even for our case. The analysis in [Eng00] also had a certain perturbation parameter, which is 0 in our case. We state the following theorem and refer the reader to the proof in section 4.5 of [Eng00].

Theorem 20 *Suppose that $|\mathrm{E}[T_S]| \geq \delta > 0$ for some nonempty set $S \subseteq [\ell] \times [\ell]$. Number the elements in S such that there is at least one element of the form $(1, j)$ and all the elements of that form are $(1, 1), \ldots, (1, d)$, where $d \leq \ell$. Then there exist $\gamma_1, \ldots, \gamma_d \in \mathbb{F}[2^r] \setminus \{0\}$ such that,*

$$\mathrm{E}_{U,W_1,\ldots,W_d}\left[\Delta\right] \geq \frac{\delta^2}{(|\mathbb{F}[2^r]| - 1)^{|S|}}$$

where,

$$\Delta = \sum_{\substack{\alpha,\beta_1,\ldots,\beta_d \\ \alpha = \pi_1(\beta_1) + \cdots + \pi_d(\beta_d)}} \widehat{A}^2_{\gamma,\alpha} \widehat{B}^2_{1,\gamma_1,\beta_1} \ldots \widehat{B}^2_{d,\gamma_d,\beta_d}$$

and,

$$\begin{aligned} \gamma &= \gamma_1 + \cdots + \gamma_d \\ \widehat{A}_{\gamma,\alpha} &= \langle \psi_\gamma \circ A, \chi_\alpha \rangle \\ \widehat{B}_{j,\gamma_j,\beta_j} &= \left\langle \psi_{\gamma_j} \circ B_j, \chi_{\beta_j} \right\rangle \end{aligned}$$

We now define proofs (P, Q) for the Raz Verifier as follows. For a set W, pick β with probability $\widehat{B}^2_{\gamma_1,\beta}$ and define $Q(W)$ to be $\gamma_1^{-1}\beta$, where B is the supposed encoding of $Q(W)$. Note that since $\widehat{B}_{\gamma_1,\beta} \neq 0$ for any set we pick, and $\gamma_1 \neq 0$ by lemma 18, $\gamma_1^{-1}\beta$ satisfies all the equations of W. For a set U, pick sets $(W_j)_{j=2}^{d}$ at random as in the Raz Verifier, and pick $(\beta_j)_{j=2}^{d}$ with probability $\prod_{j=2}^{d} \widehat{B}^2_{j,\gamma_j,\beta_j}$, and choose α

with probability $A^2_{\gamma,\alpha}$. Define $P(U)$ to be $\gamma_1^{-1}(\alpha + \sum_{j=2}^{d} \pi_j(\beta_j))$. Since $\gamma_1 \neq 0$, we have,

$$\begin{aligned} & \gamma_1^{-1}(\alpha + \sum_{j=2}^{d} \pi_j(\beta_j)) = \pi_1(\gamma_1^{-1}\beta_1) \\ \iff & \gamma_1^{-1}(\alpha + \sum_{j=2}^{d} \pi_j(\beta_j)) = \gamma_1^{-1}(\pi_1(\beta_1)) \\ \iff & \alpha + \sum_{j=2}^{d} \pi_j(\beta_j) = \pi_1(\beta_1) \\ \iff & \alpha = \sum_{j=1}^{d} \pi_j(\beta_j). \end{aligned}$$

Using the above observation it is easy to see that the acceptance probability of the the Raz Verifier is given by E[Δ] and therefore,

$$\begin{aligned} \Pr[\text{Raz Verifier accepts}] & \geq \frac{\delta^2}{(|\mathbb{F}[2^r]| - 1)^{|S|}} \\ & \geq \frac{\delta^2}{(|\mathbb{F}[2^r]| - 1)^{\ell^2}} \end{aligned}$$

since $|S| \leq \ell^2$. Using the bound given by Theorem 16, we obtain,

$$\delta^2 \leq (1 - s(n)^{\kappa})^{(m/(\kappa r))}(|\mathbb{F}[2^r]| - 1)^{\ell^2}$$

which completes the analysis of the NO case. ■

C.4 Construction of Multi Prover System

We will now give a reduction from the PCP system constructed to an appropriate Multi Prover System. For convenience we shall call the PCP system constructed in the previous subsection as PCP$_1$. Also, we let $t = \ell^2 + 2\ell$ and $k = |\mathbb{F}[2^r]|$. Clearly, PCP$_1$ is a t-query PCP where the answers are from $[k]$, with the properties specified in Theorem 19. We construct a Multi Prover System MIPS$_1$ as follows. Let $P_1, \ldots, P_t$ be t provers. The verifier V_{MIPS_1} computes the t queries of V_{lin}, say $q_1, \ldots, q_t$. It computes a random permutation $\nu : [t] \mapsto [t]$ and sends q_i to $P_{\nu(i)}$, for all $1 \leq i \leq t$ and expects answers from each prover from the set $[k]$. The acceptance predicate of V_{MIPS_1} is the same as V_{lin}. Let $\mathcal{Q}$ be the set of queries that V_{lin} makes, which is the set of positions in the proof expected by V_{lin}. Let $\mathcal{Q}_i$ be the set of queries sent to P_i by V_{MIPS_1}. Clearly, $\mathcal{Q}_i = \mathcal{Q}$ for all $1 \leq i \leq t$. It is easy to see that the completeness of V_{MIPS_1} is same as that of V_{lin}. It can be shown [TS97] that if the soundness of PCP$_1$ is ε then the soundness of V_{MIPS_1} is at most $t^t\varepsilon$. It is easy to check that the properties of Strong Completeness and Extendability hold. Analogous to the PCP construction, for every prover P_i, there is a set of 'bad' queries $\bar{S}_i = \bar{S}$ consisting of the positions of the encodings of $Q(W')$ for 'bad' sets W'. Let $\mu_i(\bar{S}_i)$ be the probability that the $i^t h$ query $q_i \in \bar{S}_i$. From the construction of PCP$_1$, it can be seen that $\mu_i(\bar{S}_i) \leq mc(n)$ for $1 \leq i \leq t$. We summarize the properties in the following theorem.

Theorem 21 *Given a 7-regular instance of* MAX-3LIN *over n variables with completeness $1 - c(n)$ and soundness $1 - s(n)$, for parameters m, k and t, (where $k = 2^r$ and $t = \ell^2 + 2\ell$), there is t prover system MIPS$_1$ with provers $P_1, \ldots, P_t$ and verifier V_{MIPS_1} such that,*

1. *The verifier uses $m \log n + O(\ell m r) + t \log t$ random bits to compute a query $\bar{q} = (q_1, \ldots, q_t)$ where q_i is sent to P_i and an answer from $[k] = [2^r]$ is expected, for all $1 \leq i \leq t$. Let $\mathcal{Q}_i$ be the set of queries given to prover P_i. Then $|\mathcal{Q}_1| = \ldots |\mathcal{Q}_t|$.*

2. *If the* MAX-3LIN *instance is a YES instance, then there is a set $\bar{S}_i \subseteq \mathcal{Q}_i$ such that $\mu_i(\bar{S}_i) \leq mc(n)$ where $\mu_i(\bar{S}_i)$ is the probability that $q_i \in \bar{S}_i$. Furthermore,*

   a. *(Strong Completeness) There is a strategy $\sigma_i^* : \mathcal{Q}_i \setminus \bar{S}_i \mapsto [k]$ ($1 \leq i \leq t$) of the provers such that the verifier accepts on all queries $\bar{q}$ such that $q_j \notin \bar{S}_j$ for all $1 \leq j \leq t$.*

   b. *(Extendability) For any given query $\bar{q}$ of the verifier (possibly containing query $q_i \in \bar{S}_i$ to individual provers P_i), the strategy given by σ_i^* can be extended to the queries from $\bar{S}_i$ contained in $\bar{q}$ so that the verifier accepts on the query $\bar{q}$.*

4. *If the instance of* MAX-3LIN *is a NO instance then the probability that the verifier accepts is at most $t^t(k^{-\ell^2} + \delta)$, where $\delta^2 = (1 - s(n)^\kappa)^{(m/(\kappa r))}(k-1)^{\ell^2}$, for some universal constant κ.*

We also need the condition that the queries are uniformly distributed over the set of all possible queries to prover P_i for all $1 \leq i \leq t$. For this construct another verifier V_{MIPS_2} for a Multi Prover System MIPS$_2$. Let R_{i,q_i} be the set of all random strings to V_{MIPS_1} that generate the query q_i to prover P_i. Then the verifier V_{MIPS_2} computes a query $\bar{q} = (q_1, \ldots, q_t)$ of V_{MIPS_1}, and sends the query $\bar{q}' = ((q_1, r_{1,q_1}), \ldots, (q_t, r_{t,q_t}))$ where r_{i,q_i} is a string uniformly chosen from R_{i,q_i}. The verifier expects answer to q_i from prover P_i, and the acceptance predicate remains the same. Clearly, sending uniformly chosen random strings r_{i,q_i} does not change the completeness, since provers can disregard them, and they do not provide any information to provers, so the soundness remains the same. In MIPS$_2$, let $\mathcal{Q}_i'$ be the set of all queries to P_i. It can be seen that $|\mathcal{Q}_1'| = \cdots = |\mathcal{Q}_t'|$ and the queries are uniformly distributed over the sets $|\mathcal{Q}_i'|$. Essentially, every query of MIPS$_1$ is replicated proportional to the probability it is queried. Let the corresponding 'bad' set $\bar{S}_i'$ be the set of all queries (q_i, r) such that $q_i \in \bar{S}_i$ for all $1 \leq i \leq t$. We have the following,

$$\begin{aligned}
\frac{|\bar{S}_i'|}{|\mathcal{Q}_i'|} &= \Pr_{V_{MIPS_2} \to (q_i, r)}[(q_i, r) \in \bar{S}_i'] \\
&= \Pr_{V_{MIPS_1} \to q_i}[q_i \in \bar{S}_i] \\
&= \mu_i(\bar{S}_i) \\
&\leq mc(n) \qquad (16)
\end{aligned}$$

for all $1 \leq i \leq t$. It is easy to see that the properties of Strong Completeness and Extendability are also satisfied. The number of random bits used by V_{MIPS_2} is at most t times that of V_{MIPS_1}. We summarize the properties of MIPS$_2$ in the following theorem.

Theorem 22 *Given a 7-regular instance of* MAX-3LIN *over n variables with completeness $1 - c(n)$ and soundness $1 - s(n)$, for parameters m, k and t, (where $k = 2^r$ and $t = \ell^2 + 2\ell$), there is t prover system MIPS$_2$ with provers $P_1, \ldots, P_t$ and verifier V_{MIPS_2} such that,*

1. *The verifier uses $t(m \log n + O(\ell m r) + t \log t)$ random bits to compute a query $\bar{q}' = (q_1', \ldots, q_t')$ where q_i' is sent to P_i and an answer from $[k] = [2^r]$ is expected, for all $1 \leq i \leq t$. Let $\mathcal{Q}_i'$ be the set of queries given to prover P_i. Then $|\mathcal{Q}_1'| = \cdots = |\mathcal{Q}_t'|$ and the queries are uniformly distributed over each $\mathcal{Q}_i'$.*

2. *If the* MAX-3LIN *instance is a YES instance, then there is a set $\bar{S}_i' \subseteq \mathcal{Q}_i'$ such that*

$$\frac{|\bar{S}_i'|}{|\mathcal{Q}_i'|} \leq mc(n).$$

Furthermore,

a. *(Strong Completeness) There is a strategy* $\sigma_i'^* : \mathcal{Q}_i' \setminus \bar{S}_i' \mapsto [k]$ *(*$1 \leq i \leq t$*) of the provers such that the verifier accepts on all queries* $\bar{q}'$ *such that* $q_j' \notin \bar{S}_j'$ *for all* $1 \leq j \leq t$.

b. *(Extendability) For any given query* $\bar{q}'$ *of the verifier (possibly containing query* $q_i' \in \bar{S}_i'$ *to individual provers* P_i*), the strategy given by* $\sigma_i'^*$ *can be extended to the queries from* $\bar{S}_i'$ *contained in* $\bar{q}'$ *so that the verifier accepts on the query* $\bar{q}'$.

4. *If the instance of* MAX-3LIN *is a NO instance then the probability that the verifier accepts is at most* $t^t(k^{-\ell^2} + \delta)$*, where* $\delta^2 = (1 - s(n)^\kappa)^{(m/(\kappa r))}(k-1)^{\ell^2}$*, for some universal constant* κ.

There is a canonical reduction from the above Multi Prover System, MIPS_2 to a t-LAYERED-CSP instance with the vertices of ith layer being $\mathcal{Q}_i'$ and the hyperedges being the constraints over t vertices, one from each layer, corresponding to the queries made by the verifier. The set of vertices in V' of t-LAYERED-CSP corresponds to $\cup_{i=1}^t \bar{S}_i'$. We now set the parameters used in our reduction, which along with reduction to the MAX-3LIN instance in [KP06] would prove theorem 7.

We start with the instance MAX-3LIN of [KP06] on n variables with $c(n) = 2^{-\Omega(\sqrt{\log n})}$ and $s(n) = \Omega(\log^{-3} n)$. We take $m = \theta(\log^{3\kappa+3} n)$ and $r = \theta(\log\log n)$ such that $k = \theta(\log^{6\kappa+8} n)$. Now let $N = |V|$ be the number of vertices in the t-LAYERED-CSP instance. From the properties of MIPS_2, we have $\log N = \theta(\log^{3\kappa+4} n)$. Moreover, the size of the vertex set V' of the t-LAYERED-CSP is $Nmc(n) \leq N/(2^{(\log N)^{(1/(10\kappa+20))}})$ for large enough N.
The size of the label set $k = \theta(\log^{6\kappa+8} n) = \theta(\log^2 N)$. Since $s(n) = \Omega(\log^{-3} n)$, we have $\delta^2 = (1 - s(n)^\kappa)^{(m/(\kappa r))}(k-1)^{\ell^2} = 2^{-\Omega(\log^2 n)}(k-1)^{\ell^2}$. Therefore, the soundness $t^t(k^{-\ell^2} + \delta) = k^{-t+O(\sqrt{t})}$.

The above analysis completes the construction of the t-LAYERED-CSP instance with the desired properties in Theorem 7.