

Constraint Satisfaction Problems with Advice

Suprovat Ghoshal* Konstantin Makarychev† Yury Makarychev‡
 Northwestern and TTIC Northwestern TTIC

Abstract

We initiate the study of algorithms for constraint satisfaction problems with ML oracle advice. We introduce two models of advice and then design approximation algorithms for Max Cut, Max 2-Lin, and Max 3-Lin in these models. In particular, we show the following.

- For Max-Cut and Max 2-Lin, we design an algorithm that yields near-optimal solutions when the average degree is larger than a threshold degree, which only depends on the amount of advice and is independent of the instance size. We also give an algorithm for nearly satisfiable Max 3-Lin instances with quantitatively similar guarantees.
- Further, we provide impossibility results for algorithms in these models. In particular, under standard complexity assumptions, we show that Max 3-Lin is still $1/2 + \eta$ hard to approximate given access to advice, when there are no assumptions on the instance degree distribution. Additionally, we also show that Max 4-Lin is $1/2 + \eta$ hard to approximate even when the average degree of the instance is linear in the number of variables.

1 Introduction

In recent years, numerous breakthroughs have occurred in machine learning (ML), and today, ML tools can solve tasks that were completely out of reach even a decade ago. This has sparked an interest in designing and using algorithms and data structures that rely on ML oracle advice (see e.g. [BDSV18; Mit18; PSK18; HIKV19; GP19; LV21]). Algorithms with access to ML advice can be used in the setting when we can learn some unknown information about the problem at hand using machine learning tools. This can be some information about the result or optimal solution (e.g., the position of an element in the sorted array [LRSZ21; CL23; BC24] or membership in a set [KBCDP18; Mit18]), information about future events or yet unread data in the input stream (see e.g., [MNS07; DH09; VVS10; MV17; GP19; LLMV20; Mit20]), as well as some information about the optimal set of parameters for the algorithm [BDSV18; DILMV21].

In this paper, we introduce two models for solving constraint satisfaction problems (CSPs) with oracle advice. Suppose that we are given a constraint satisfaction problem with predicates $\Psi = \{\psi_1, \dots, \psi_m\}$ and Boolean variables x_1, \dots, x_n .¹ It will be convenient for us to assume that value 1 represents false and -1 represents true. Let $x^* = (x_1^*, \dots, x_n^*)$ be a fixed optimal solution, which we will refer to as the ground-truth solution. Now we assume that we are given noisy advice about x_i^* . Specifically, we consider two models.

- **Label Advice.** In the first model, the algorithm receives advice $\tilde{x} = (\tilde{x}_1, \dots, \tilde{x}_n)$, where \tilde{x}_i is a noisy prediction of the ground-truth value x_i^* . Each \tilde{x}_i is a random variable taking values -1 and 1 and is slightly biased toward x_i^* . Namely, $\tilde{x}_i = x_i^*$ with probability $\frac{1+\varepsilon}{2}$ and $\tilde{x}_i = -x_i^*$ with probability $\frac{1-\varepsilon}{2}$. All variables \tilde{x}_i are independent.

*Supported by NSF Awards CCF-1955351.

†Supported by NSF Awards CCF-1955351 and EECS-2216970.

‡Supported by NSF Awards CCF-1955173, CCF-1934843, and EECS-2216899.

¹More generally, we can also consider the case where variables take values in some domain $[d] = \{1, \dots, d\}$.

- **Variable Subset Advice.** In the second model, the algorithm receives a random subset of variables/indices $S \subseteq \{1, \dots, n\}$ and their values $(x_i^*)_{i \in S}$. Subset S includes every i with probability ε ; all events $i \in S$ are independent.

These models capture the setting where we have an ML algorithm (or another oracle) that provides unreliable predictions for values x_i^* . Since the predictions are very noisy, they do not provide a good solution. Consider, for example, the Label Advice model for Max 2-Lin, a constraint satisfaction problem with constraints $x_i \cdot x_j = -1$ and $x_i \cdot x_j = 1$. Even if the optimal solution satisfies all the constraints, solution \tilde{x} satisfies only a $\frac{1+\varepsilon^2}{2}$ fraction of the constraints in expectation; this is just a tiny bit better than the $1/2$ fraction that a random solution satisfies. The aim of this paper is to show that, nevertheless, advice \tilde{x} may be very valuable. In this paper, we focus on the Max k -Lin problem. Our choice of the Max k -Lin problem is not arbitrary. Håstad [Hås01] proved that Max 3-Lin is approximation resistant (as well as every Max k -Lin problem with $k \geq 3$). In other words, it is not possible to obtain a solution satisfying $1/2 + \eta$ fraction of all constraints in polynomial time even if the optimal solution satisfies $1 - \delta$ fraction of all constraints (for every positive δ and η ; assuming $P \neq NP$). For Max 2-Lin and Max Cut problems, the best approximation can be obtained using the Goemans-Williamson algorithm [GW95; KKM07]. We show how to get nearly optimal solutions using oracle advice for Max Cut, Max 2-Lin and almost satisfiable instances of Max 3-Lin when the number of constraints is sufficiently large (for unweighted instances, the number of constraints should be at least $C_\varepsilon n$ for Max Cut and $C_{\varepsilon, \delta} n$ for Max 3-Lin, where $C_\varepsilon n$ and $C_{\varepsilon, \delta} n$ depend only on ε and ε, δ , respectively). We complement our algorithmic results with hardness of approximation bounds. Specifically, we show that there are no polynomial-time algorithms for Max k -Lin instances with $k \geq 3$ in the Label Advice and Variable Subset Advice models that given a $(1 - \delta)$ -satisfiable instance and advice find a solution satisfying at least a $(1/2 + \delta)$ -fraction of the constraints with constant probability (for every $\delta > 0$). Furthermore, for $k \geq 4$, we show that there are no algorithms for $(1 - \delta)$ -satisfiable Max k -Lin instances with n variables and $\Theta(n^2)$ constraints (“high-degree instances”) that satisfy at least a 0.99 fraction of the constraints with high probability.

Comparing the Models. We note that the Variable Subset Advice model provides more information than the Label Advice one. Indeed, given set S and values $(x_i^*)_{i \in S}$, we can generate advice \tilde{x} as follows: if $i \in S$, let $\tilde{x}_i = x_i^*$; otherwise, sample \tilde{x}_i uniformly at random. It is immediate that

$$\Pr(\tilde{x}_i = x_i^*) = \Pr(x_i = x_i^* \mid i \in S) \Pr(i \in S) + \Pr(x_i = x_i^* \mid i \notin S) \Pr(i \notin S) = 1 \cdot \varepsilon + 1/2 \cdot (1 - \varepsilon) = \frac{1 + \varepsilon}{2}$$

as required, and all \tilde{x}_i are independent. Thus, every algorithm for the Label Advice model also works in the Variable Subset Advice model. For this reason, we will consider the Label Advice model in this paper.

Max Cut and Max k -Lin Problems. We recall the definitions of Max Cut and Max 2-Lin problems.

Definition 1.1 (Max Cut) In Max Cut, we are given an undirected graph $G = (V, E)$ with edge weights $w_e > 0$. The goal is to find a cut (S, T) that maximizes the total weight of cut edges.

Alternatively, Max Cut can be stated as a constraint satisfaction problem. We are given a set of Boolean variables x_1, \dots, x_n and a set of constraints of the form $x_i \cdot x_j = -1$ (or, equivalently, $x_i \neq x_j$); each constraint has a non-negative weight. The goal is to find an assignment that maximizes the total weight of satisfied constraints.

The connection between the graph and CSP formulations of Max Cut is straightforward: vertex v_i corresponds to variable x_i and edge (v_i, v_j) corresponds to constraint $x_i \cdot x_j = -1$. If $v_i \in S$ then $x_i = -1$; if $v_i \in T$ then $x_i = 1$. We now define Max 2-Lin, which is a generalization of Max Cut.

Definition 1.2 (Max 2-Lin) In Max 2-Lin, we are given a set of Boolean variables x_1, \dots, x_n and a set of constraints of the form $x_i \cdot x_j = c_{ij}$ where $c_{ij} \in \{-1, 1\}$; each constraint has a non-negative weight w_{ij} . The goal is to find an assignment that maximizes the total weight of satisfied constraints.

Both problems Max Cut and Max 2-Lin are NP-hard [Kar10]. The Goemans–Williamson algorithm provides an $\alpha_{GW} = 0.878\dots$ approximation [GW95] for them and, as Khot, Kindler, Mossel, and O’Donnell showed, this is optimal assuming the Unique Games Conjecture [KKMO07]. If a Max Cut or Max 2-Lin instance is almost satisfiable – that is, the value of the optimal solution is $(1 - \delta)W$, where W is the total weight of all the constraints/edges – then the Goemans–Williamson algorithm finds a solution of value $1 - O(\sqrt{\delta})$. This result is again optimal assuming the Unique Games Conjecture [KKMO07]. See [MM17] for a detailed discussion of these and other approximation and hardness results for constraint satisfaction problems.

Definition 1.3 (Max k -Lin) *In Max k -Lin, we are given a set of Boolean variables x_1, \dots, x_n and a set of constraints of the form $x_{i_1} \cdots x_{i_k} = c_{i_1 \dots i_k}$ where $c_{i_1 \dots i_k} \in \{-1, 1\}$; each constraint has a non-negative weight w_{ij} . The goal is to find an assignment that maximizes the total weight of satisfied constraints.*

Unlike the case of Max 2-Lin, one cannot find near-optimal solutions to nearly satisfiable instances of Max 3-Lin. In particular, Håstad [Hås01] showed that for Max 4-Lin is approximation resistant i.e., for instances where the optimal solution satisfies at least $1 - \delta$ fraction of constraints, it is NP-hard to find an assignment that satisfies more than $1/2 + \delta$ fraction of constraints, for every small constant $\delta > 0$.

1.1 Our results

We design polynomial-time approximation algorithms for Max Cut, Max 2-Lin, and Max 3-Lin. In unweighted graphs, the algorithm for Max Cut finds a nearly optimal solution if the average degree $\Delta = 2m/n$ is a sufficiently large constant (for a fixed parameter ε), specifically $\Delta \geq C/\varepsilon^2$ (here m is the number of constraints/edges, n is the number of variables/vertices; C is a constant).

Theorem 1.4 *There exists a polynomial-time algorithm for the Label Advice model that given an unweighted instance of Max Cut or Max 2-Lin and advice \tilde{x} finds a solution of value at least $(1 - O(1/\varepsilon\sqrt{\Delta})) OPT$ in expectation (over the random advice), where OPT is the value of the optimal solution, ε is the parameter of the model, and Δ is the average degree (see above).*

In weighted graphs, our results require that $n \sum_{ij} \left(\frac{w_{ij}}{W}\right)^2 \ll \varepsilon^2$, where W is the total weight of all constraints.

Theorem 1.5 *There exists a polynomial-time algorithm for the Label Advice model that given an instance of Max Cut or Max 2-Lin and advice \tilde{x} finds a solution of value at least $OPT - \sqrt{n \sum_{ij} w_{ij}^2}/\varepsilon$ in expectation (over random advice), where OPT is the value of the optimal solution, ε is the parameter of the model, and w_{ij} is the weight of the constraint for x_i and x_j .*

Next, we state our result for the Max 3-Lin problem. As stated in the theorem below, we show that for nearly satisfiable Max 3-Lin instances, there exists an efficient algorithm that finds a nearly satisfying assignment, if the average degree of the instance is large enough.

Theorem 1.6 *There exists a polynomial-time algorithm that given a $(1 - \delta)$ -satisfiable (unweighted) instance Φ of Max 3-Lin having n variables and at least $\delta^{-1} \ln \frac{1}{\delta} \cdot \varepsilon^{-6} \cdot n$ constraints along with a label advice with parameter $\varepsilon > 0$, returns a solution that satisfies a $1 - O(\sqrt{\delta})$ fraction of all the constraints.*

Hardness Results. We complement the above positive results with a couple of hardness results. Our first result is the following theorem which shows that under standard assumptions, Max 3-Lin retains its $(1/2 + o(1))$ -inapproximability if there are no guarantees on its average degree, even in the setting where the algorithm has access to the oracle advice.

Theorem 1.7 *Assume that the Exponential Time Hypothesis (ETH) and Linear Size PCP Conjecture hold. For every $\delta > 0$, there exists $\varepsilon_0 = \varepsilon_0(\delta)$ such that for every $\varepsilon \in [0, \varepsilon_0]$, there is no polynomial time algorithm for Max 3-Lin in the variable subset advice model with parameter ε that given a $(1 - \delta)$ -satisfiable instance returns a solution satisfying at least a $(1/2 + \delta)$ -fraction of the constraints with probability at least 0.9 over the random advice.*

Next, we show that for the Max 4-Lin problem, there are no efficient algorithms in the oracle advice setting that can find nearly optimal solutions, even when the average degree of the instance is large.

Theorem 1.8 *Assume that the Exponential Time Hypothesis (ETH) and Linear Size PCP Conjecture hold. Then there exists a constant $\delta_0 \in (0, 1)$ such that for every $\delta \in (0, \delta_0)$, there exists $d(\delta) \in \mathbb{N}$ and $\varepsilon(\delta) \in (0, 1)$ for which the following statement holds. For every $\varepsilon \in (0, \varepsilon(\delta))$, $n \geq n(\delta)$, and $d \in [d(\delta), n]$, there is no polynomial time algorithm for Max 4-Lin in the variable subset advice model with parameter ε that, given a $(1 - \delta)$ -satisfiable Max 4-Lin instance with n variables and at least $\Omega(dn)$ constraints, returns a solution satisfying a least a $(1/2 + \delta)$ -fraction of the constraints with probability at least 0.9 over the random advice.*

Remark 1.9 *Theorem 1.7 holds not only for Max 3-Lin but also for Max k -Lin with $k \geq 3$ and Theorem 1.8 holds for Max k -Lin with $k \geq 4$. To see the former, note that any instance of 3-Lin can be converted to an equivalent instance of Max k -Lin as follows. Let us introduce $k - 3$ new variables y_1, \dots, y_{k-3} and then replace each constraint $x_a x_b x_c = c_{abc}$ with $x_a x_b x_c y_1 \cdots y_{k-3} = c_{abc}$. Every solution of the original Max 3-Lin instance can be extended to a solution of the obtained Max k -Lin by letting $y_1 = \cdots = y_{k-3} = 1$ and keeping all x_i as is. Also, every solution of Max k -Lin can be transformed to a solution of Max 3-Lin: if $y_1 \cdots y_{k-3} = 1$, keep all x_i as is; otherwise, negate all x_i . The same transformation shows that Theorem 1.8 applies when $k \geq 4$. In this case, however, the analysis of the transformation is non-black-box but still pretty simple.*

Note that while Theorems 1.7 and 1.8 have been stated for the variable subset advice model, by the discussion above, it follows that they also apply to the label advice model as well (with parameter $\varepsilon/2$ instead of ε).

Discussion. Our results for Max-2-Lin and Max-Cut (Theorem 1.4) show that using oracle advice we can find a near-optimal solution as long as the average degree of the instance is large enough, irrespective of the optimal value of the instance (importantly, the degree requirement depends on the amount of oracle advice ε but not on the size of the instance). Theorem 1.5 gives an analogous result for the weighted variants of these problems. For Max 3-Lin, we obtain a similar guarantee for *nearly satisfiable* instances (with a different requirement on the average degree), and it is an interesting open question if one can find near-optimal solutions for Max 3-Lin instances that are not nearly satisfiable, given access to oracle advice. To complement our positive results, Theorem 1.7 shows that if there are no assumptions on the average degree of the instance, Max 3-Lin is still $1/2 + o(1)$ hard to approximate even with oracle advice (under certain complexity assumptions). That is, our assumption on the average degree (or another related assumption) is necessary in our algorithmic result for Max 3-Lin. Finally, Theorem 1.8 shows that Max 4-Lin is much more difficult than Max 2-Lin and Max 3-Lin: even if the input instance is nearly satisfiable and has a high average degree (the number of constraints is quadratic in the number of variables), it is hard to find a solution that satisfies strictly better than $1/2$ -fraction of the constraints (in our oracle-advice models).

1.2 Comparison with Independent Works

Independently and concurrently with our work, Cohen-Addad, d’Orsi, Gupta, Lee, Panigrahi [CdGLP24] proposed the same model of ML oracle advice for constraint satisfaction problems. In their work, they provide approximation algorithms for Max Cut that have approximation factors $\alpha_{GW} + \tilde{\Omega}(\varepsilon^4)$ and $\alpha_{GW} + \Omega(\varepsilon)$ in the Label Advice and Variable Subset Advice models, respectively (they call these models *Noisy Predictions* and *Partial Predictions*). They also present an algorithm that finds almost optimal solutions for Max Cut

on Δ -wide graphs, a notion introduced in their paper. Loosely speaking, a graph is Δ -wide if not all weight is concentrated on few edges. We refer the reader to [CdGLP24] for details. Given an instance of Max Cut satisfying OPT constraints, the algorithm by Cohen-Addad et al. [CdGLP24] finds a solution of value $(1 - O(\eta + 1/\varepsilon\sqrt{\Delta})) OPT$. This result is somewhat similar but not directly comparable to our Theorems 1.4 and 1.5. For example, in regular unweighted and weighted Δ -wide graphs, our algorithm finds a solution of value at least $(1 - O(\sqrt{\eta}/\varepsilon\sqrt{\Delta})) OPT$; this guarantee is stronger than the one given by Cohen-Addad et al. However, our algorithm cannot be used on some Δ -wide graphs, whose vertex degrees are very irregular. The authors complement their results with an algorithm for Δ -narrow graphs i.e., graphs that are not Δ -wide. This algorithm does not use any oracle advice. It is based on the algorithms by Feige, Karpinski, and Langberg [FKL02] and Hsieh and Kothari [HK22] for low-degree instances of Max Cut. In their paper, Cohen-Addad et al. [CdGLP24] do not study Max k -Lin for $k > 2$.

In an independent work [BEX24], Bampis, Escoffier, and Xeferis considered a similar (but not identical) notion of advice and provided PTAS for δ -dense instances of Max Cut and other problems (a δ -dense instance of Max Cut contains $\delta n(n-1)/2$ edges).

2 Algorithm for Max Cut and Max 2-Lin with Advice

Our algorithm finds a solution for Max 2-Lin instance by (approximately) maximizing the quadratic form

$$\sum_{i,j} \frac{|a_{ij}| + a_{ij}x_i x_j}{4}, \quad (1)$$

where the matrix $A = (a_{ij})$ is defined as follows. For Max Cut, A is minus adjacency matrix i.e., $a_{ij} = -w_{ij}$, where w_{ij} is the weight of edge (i, j) . For Max 2-Lin, $a_{ij} = w_{ij}$ if we have constraint $x_i x_j = 1$ and $a_{ij} = -w_{ij}$ if we have $x_i x_j = -1$, w_{ij} is the weight of the constraint for x_i and x_j . Note that matrix A is an $n \times n$ symmetric matrix with zero diagonal. We remark that Quadratic Program (1) was used by Goemans and Williamson [GW95] in their seminal paper on semi-definite programming algorithm for Max Cut.

In this section, we give an algorithm for Max Cut that finds a solution of value $(1 - O(1/\sqrt{\Delta} \cdot \varepsilon^{-1})) OPT$, where OPT is the value of the optimal solution, ε is the parameter of our model, and $\Delta = 2m/n$ is the average vertex degree in the graph. Our algorithms give a nontrivial approximation when $\Delta \geq C/\varepsilon^2$. Note that matrix $A = (a_{ij})$ is symmetric with a zero diagonal.

The value of quadratic form (1) exactly equals the number of satisfied constraints in the corresponding instance of Max 2-Lin. Indeed, if the constraint for x_i and x_j is satisfied, then the term $|a_{ij}| + a_{ij}x_i x_j = 2|a_{ij}|$; if it is not satisfied, then $|a_{ij}| + a_{ij}x_i x_j = 0$. Consequently,

$$\frac{|a_{ij}| + a_{ij}x_i x_j + |a_{ji}| + a_{ji}x_i x_j}{4} = w_{ij},$$

if the constraint for x_i and x_j is satisfied; and

$$\frac{|a_{ij}| + a_{ij}x_i x_j + |a_{ji}| + a_{ji}x_i x_j}{4} = 0,$$

if the constraint is not satisfied. Since the sum of all $|a_{ij}|$ equals $2W$, where W is the total weight of all constraints, quadratic form (1) equals

$$\frac{W}{2} + \sum_{i,j} \frac{a_{ij}x_i x_j}{4}. \quad (2)$$

In the next section, we show how to obtain a solution of value $OPT - \varepsilon^{-1}\sqrt{n}\|A\|_F$ for Max QP (see Theorem 3.1). Here, $\|A\|_F$ is the Frobenius norm of A . This result implies Theorem 1.5 and also Theorem 1.4, because $OPT \geq W/2$.

3 Quadratic Forms with Advice

In this section, we consider the quadratic form maximization problem with advice. Let $A = (a_{ij})$ be a symmetric $n \times n$ matrix. Our goal is to maximize the quadratic form $\sum_{ij} a_{ij} x_i x_j$ for $x \in \{-1, 1\}^n$. This quadratic form can also be written as $\langle x, Ax \rangle$. We assume that the ground truth solution is x^* . The algorithm receives advice \tilde{x} . Each \tilde{x}_i is a random variable, $\tilde{x}_i = x_i^*$ with probability $(1 + \varepsilon)/2$ and $\tilde{x}_i = -x_i^*$ with probability $(1 - \varepsilon)/2$. All variables \tilde{x}_i are independent. The main result of this section is the following theorem.

Theorem 3.1 *Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix with zero diagonal entries. Then, there exists a deterministic polynomial time algorithm that finds a labeling x' such that randomizing over the choice of advice \tilde{x} , we have that*

$$\mathbb{E}_{\tilde{x}} [\langle x', Ax' \rangle] \geq \langle x^*, Ax^* \rangle - \varepsilon^{-1} \sqrt{n} \|A\|_F. \quad (3)$$

The algorithm for the theorem is described below.

Algorithm

Input: Coefficient matrix A , oracle advice $\tilde{x}_1, \dots, \tilde{x}_n \in \{\pm 1\}$.

Output: Solution x'_1, \dots, x'_n .

1. Define $F(x, y) := \langle x, Ay \rangle - \|A(\varepsilon x - y)\|_1$.
2. Solve the following mathematical program with a concave objective:

$$\text{Maximize} \quad F(x, \tilde{x}) \quad (4)$$

$$\text{Subject to} \quad x_i \in [-1, 1] \quad \forall i \in [n] \quad (5)$$

3. Round the fractional solution x coordinate-by-coordinate to a solution $x' \in \{\pm 1\}^n$ such that $\langle x', Ax' \rangle \geq \langle x, Ax \rangle$.
4. Output labeling x' .

Figure 1: Quadratic Program Maximization Algorithm

Proof. First, observe that function $F(x, y)$ is a concave function of x for a fixed y , since $\langle x, Ay \rangle$ is a linear function of x , and $\|A(\varepsilon x - y)\|_1$ is a convex function (because all vector norms are convex). Thus, we can find the maximum of $F(x, \tilde{x})$ subject to the constraint $x \in [-1, 1]^n$ in polynomial time. Also, note that $\langle x, Ax \rangle$ is a linear function of each x_i when all other coordinates x_j ($j \neq i$) are fixed. Thus, the algorithm can round each x to a $x' \in \{\pm 1\}^n$ by rounding coordinates one-by-one. At every step, the algorithm replaces one coordinate $x_i \in [-1, 1]$ with -1 or $+1$ making sure that the quadratic form $\langle x, Ax \rangle$ does not decrease.

We now show that inequality (3) holds. We begin with the following claim.

Claim 3.2 *For every $x, y \in [-1, 1]^n$, we have*

$$\langle Ax, x \rangle \geq \frac{F(x, y)}{\varepsilon}.$$

Proof. Write:

$$\langle x, Ax \rangle = \frac{\langle x, Ay \rangle + \langle x, A(\varepsilon x - y) \rangle}{\varepsilon} \geq \frac{\langle x, Ay \rangle - \|A(\varepsilon x - y)\|_1}{\varepsilon} = \frac{F(x, y)}{\varepsilon},$$

where the inequality follows from Hölder's inequality:

$$|\langle x, A(\varepsilon x - y) \rangle| \leq \|A(\varepsilon x - y)\|_1.$$

□

Next, we bound the expected value of the optimization program (4)

Lemma 3.3 *Let $x \in [-1, 1]^n$ denote the optimal solution of concave program (4). Then,*

$$\mathbb{E}_{\tilde{x}} [F(x, \tilde{x})] \geq \varepsilon \langle x^*, Ax^* \rangle - \sqrt{n} \|A\|_F.$$

Proof. The ground truth solution x^* is always a feasible solution to program (4). Hence, we have

$$\mathbb{E}_{\tilde{x}} [F(x, \tilde{x})] \geq \mathbb{E}_{\tilde{x}} [F(x^*, \tilde{x})] = \mathbb{E}_{\tilde{x}} [\langle x^*, A\tilde{x} \rangle - \|A(\varepsilon x^* - \tilde{x})\|_1] = \varepsilon \langle x^*, Ax^* \rangle - \mathbb{E}_{\tilde{x}} [\|A(\varepsilon x^* - \tilde{x})\|_1].$$

Here, we used that $\tilde{x} = \varepsilon x^*$. To bound the last term, we let $z = \varepsilon x^* - \tilde{x}$ and observe that $\|Az\|_1 \leq \sqrt{n} \|Az\|_2$ by the Cauchy-Schwarz inequality (since Az is an n -dimensional vector). Then,

$$\mathbb{E} \|A(\varepsilon x^* - \tilde{x})\|_1 = \mathbb{E} \|Az\|_1 \leq \sqrt{n} \mathbb{E} \|Az\|_2.$$

By Jensen's inequality,

$$\sqrt{n} \mathbb{E} \|Az\|_2 \leq \sqrt{n} \mathbb{E} [\|Az\|_2^2]^{1/2} = \sqrt{n} \mathbb{E} [\langle Az, Az \rangle]^{1/2} = \sqrt{n} \mathbb{E} [\langle z, A^* A z \rangle]^{1/2} = \sqrt{n} \mathbb{E} \left[\sum_{ij} (A^* A)_{ij} z_i z_j \right]^{1/2}.$$

Random variables z_i are mutually independent and $\mathbb{E}[z_i] = 0$ for all i . Thus, $\mathbb{E}[z_i z_j] = 0$ if $i \neq j$. Also, $\mathbb{E}[z_i^2] = \text{Var}[\tilde{x}_i] = 1 - \varepsilon^2$. Therefore,

$$\mathbb{E} \left[\sum_{ij} (A^* A)_{ij} z_i z_j \right] = (1 - \varepsilon^2) \sum_i (A^* A)_{ii} = (1 - \varepsilon^2) \text{tr}(A^* A) = (1 - \varepsilon^2) \|A\|_F^2.$$

Putting the bounds together completes the proof. \square

We are now ready to finish the proof of Theorem 3.1. Recall that $x' \in \{-1, 1\}^n$ is the integral solution obtained by greedy coordinate-wise rounding of x . Thus,

$$\mathbb{E}_{\tilde{x}} [\langle x', Ax' \rangle] \geq \mathbb{E}_{\tilde{x}} [\langle x, Ax \rangle] \geq \mathbb{E}_{\tilde{x}} \left[\frac{F(x, \tilde{x})}{\varepsilon} \right] \geq \frac{\varepsilon \langle x^*, Ax^* \rangle - \sqrt{n} \|A\|_F}{\varepsilon} = \langle x^*, Ax^* \rangle - \sqrt{n} \varepsilon^{-1} \|A\|_F.$$

\square

4 Algorithm for Max 3-Lin with Advice

Theorem 1.6 *There exists a polynomial-time algorithm that given a $(1 - \delta)$ -satisfiable (unweighted) instance Φ of Max 3-Lin having n variables and at least $\delta^{-1} \ln \frac{1}{\delta} \cdot \varepsilon^{-6} \cdot n$ constraints along with a label advice with parameter $\varepsilon > 0$, returns a solution that satisfies a $1 - O(\sqrt{\delta})$ fraction of all the constraints.*

Remark: Instance Φ must have at least $\ln \frac{1}{\delta} \cdot \delta^{-1} \varepsilon^{-6} \cdot n$ constraints. This means that every variable should participate in $3\delta^{-1} \ln \frac{1}{\delta} \cdot \varepsilon^{-6}$ constraints on average.

Proof. Our algorithm works as follows. First, it creates a new *weighted* instance Ψ of Max 2-Lin. This instance has the same set of variables, x_1, \dots, x_n , as Max 3-Lin instance Φ but a different set of constraints, which are created by functions CREATE-H-CONSTRAINTS and CREATE-L-CONSTRAINTS (see below). Then, the algorithm solves the Max 2-Lin instance using the Goemans and Williamson algorithm for MAX CUT [GW95] and obtains a solution $\hat{x}_1, \dots, \hat{x}_n$. Finally, it outputs $\hat{x}_1, \dots, \hat{x}_n$ as a solution to the original Max 3-Lin instance. We now describe functions CREATE-H-CONSTRAINTS and CREATE-L-CONSTRAINTS.

Fix a threshold $t = 8\varepsilon^{-2} \ln 1/\delta$. Denote by $E(\Phi)$ the set of indices of all constraints in Φ i.e., for every constraint $x_i x_j x_k = c_{ijk}$, set $E(\Phi)$ contains an *unordered* tuple (i, j, k) . Let $E_{ij} = \{(i, j, k) \in E(\Phi)\}$ be

the set of indices of all constraints in Φ that contain variables x_i and x_j . We say that set E_{ij} is heavy if it contains at least t elements. If a constraint belongs to at least one heavy set, we call it heavy. In other words, constraint $x_i x_j x_k = c_{ijk}$ is heavy if E_{ij} , E_{jk} , or E_{ik} is heavy. If a constraint is not heavy, we call it light.

For every *heavy* set E_{ij} , function CREATE-H-CONSTRAINTS creates $2|E_{ij}|$ constraints in Ψ . It first computes

$$\sigma_{ij} = \text{sgn} \left(\sum_{k: (i,j,k) \in E_{ij}} c_{ijk} \tilde{x}_k \right)$$

and then creates constraints $x_i x_j = \sigma_{ij}$ and $x_k = \sigma_{ij} c_{ijk}$ in Ψ for each constraint $x_i x_j x_k = c_{ijk}$ in E_{ij} . We say that these constraints are representatives for $x_i x_j x_k = c_{ijk}$ in the new instance Ψ . Note that the first constraint ($x_i x_j = \sigma_{ij}$) is identical for all constraints in E_{ij} and does not depend on k . However, the second constraint ($x_k = \sigma_{ij} c_{ijk}$) does depend on k . Strictly speaking, if $\sigma_{ij} = 0$, constraint $x_i x_j = \sigma_{ij}$ is not a valid constraint because the right hand side must be either 1 or -1 . To make this constraint valid, we replace σ_{ij} with 1 but nevertheless, we conservatively assume that this constraint is always violated.

For every variable x_i , function CREATE-L-CONSTRAINTS finds all light constraints that contain x_i . Denote this set by L_i . Then, for each constraint $x_i x_j x_k = c_{ijk}$ in L_i , CREATE-L-CONSTRAINTS creates a constraint $x_i = \sigma_i$, where

$$\sigma_i = \text{sgn} \left(\sum_{j,k: (i,j,k) \in L_i} c_{ijk} \tilde{x}_j \tilde{x}_k \right).$$

We say that this constraint is a representative for $x_i x_j x_k = c_{ijk}$ in the new instance Ψ . Note that CREATE-L-CONSTRAINTS creates identical constraints for all constraints in L_i . If $\sigma_i = 0$, then, as before, we replace σ_i with 1 but will assume that $x_i = \sigma_i$ is violated for every solution x .

Analysis. We have completely described the algorithm and now proceed to the analysis. It is clear that our algorithm runs in polynomial time. We prove that the expected fraction of satisfied constraints is $1 - O(\sqrt{\delta})$. We first state the main technical lemma.

Lemma 4.1 *The ground truth solution x_1^*, \dots, x_n^* to Φ satisfies a $(1 - O(\delta))$ fraction of all the constraints in Ψ , in expectation.*

We prove Lemma 4.1 in Section 4.1. Now, we show that Lemma 4.1 implies Theorem 1.6. First, observe that for every constraint in the original Max 3-Lin instance Φ our algorithm creates 2, 3, 4, or 6 constraints in Max 2-Lin instance Ψ : For each light constraint, it creates exactly 3 constraints; for each heavy constraint, it creates from 2 to 6 new constraints. Also, note that every constraint in Ψ is a representative for exactly one constraint in Φ . By Lemma 4.1, the optimal solution satisfies a $(1 - O(\delta))$ fraction of the constraints in Ψ , in expectation. Thus, the Goemans–Williamson algorithm [GW95] finds a solution satisfying a $(1 - O(\sqrt{\delta}))$ fraction of the constraints in Ψ , in expectation (see also Theorem 2 in survey [MM17])². This means that at most $O(\sqrt{\delta}m)$ constraints in Max 2-Lin instance Ψ are not satisfied by $\hat{x}_1, \dots, \hat{x}_n$, in expectation.

We now bound the number of constraints in Φ not satisfied by $\hat{x}_1, \dots, \hat{x}_n$. We separately consider heavy and light constraints. If $x_i x_j x_k = c_{ijk}$ is a heavy constraint, then at least one of the sets E_{ij} , E_{jk} , E_{ik} must be heavy. Assume without loss of generality that E_{ij} is heavy. Then, $x_i x_j x_k = c_{ijk}$ must have two representatives in Ψ : $x_i x_j = \sigma_{ij}$ and $x_k = \sigma_{ij} c_{ijk}$. If both of these constraints are satisfied in Ψ by $\hat{x}_1, \dots, \hat{x}_n$, then $x_i x_j x_k = c_{ijk}$ is also satisfied, because

$$\hat{x}_i \hat{x}_j \hat{x}_k = (\hat{x}_i \hat{x}_j) \hat{x}_k = \sigma_{ij} (\sigma_{ij} c_{ijk}) = (\sigma_{ij})^2 c_{ijk} = c_{ijk}.$$

Therefore, if a heavy constraint in Φ is not satisfied by $\hat{x}_1, \dots, \hat{x}_n$, then one of its representatives is not satisfied by $\hat{x}_1, \dots, \hat{x}_n$. Since every constraint in Ψ is a representative for exactly one constraint in Φ , the

²The intermediate Max 2-Lin instance Ψ may contain duplicate constraints, and hence the analysis counts the constraints with their multiplicities. However, this is fine since the result of [GW95] holds for weighted instances as well.

number of unsatisfied heavy constraints is upper bounded by the number of unsatisfied constraints in Ψ , which is at most $O(\sqrt{\delta m})$.

Suppose $x_i x_j x_k = c_{ijk}$ is a light constraint. We claim that this constraint is satisfied by $\hat{x}_1, \dots, \hat{x}_n$ if (1) it is satisfied by the ground truth solution x_1^*, \dots, x_n^* and (2) its representatives – constraints $x_i = \sigma_i$, $x_j = \sigma_j$, $x_k = \sigma_k$ – are satisfied by both x_1^*, \dots, x_n^* and $\hat{x}_1, \dots, \hat{x}_n$. Indeed, in this case, we have $\hat{x}_i = \sigma_i = x_i^*$, $\hat{x}_j = \sigma_j = x_j^*$, $\hat{x}_k = \sigma_k = x_k^*$, and $\hat{x}_i \hat{x}_j \hat{x}_k = x_i^* x_j^* x_k^* = c_{ijk}$. Therefore, if $x_i x_j x_k = c_{ijk}$ is not satisfied by $\hat{x}_1, \dots, \hat{x}_n$, then (1) it is also not satisfied in the ground truth solution; or (2) one of its representatives is not satisfied by x_1^*, \dots, x_n^* or $\hat{x}_1, \dots, \hat{x}_n$. Consequently, the number of light constraints unsatisfied by $\hat{x}_1, \dots, \hat{x}_n$ is upper bounded by the number of light constraints unsatisfied by x_1^*, \dots, x_n^* plus the number of constraints in Ψ not satisfied by either x_1^*, \dots, x_n^* or $\hat{x}_1, \dots, \hat{x}_n$. The total number of such constraints is $O(\delta m) + O(\delta m) + O(\sqrt{\delta m}) = O(\sqrt{\delta m})$. This concludes the proof of Theorem 1.6. \square

4.1 Proof of Lemma 4.1

In this section, we prove Lemma 4.1.

Proof of Lemma 4.1. We first upper bound the expected number of representatives for heavy constraints violated by the ground truth solution x^* . Let R_{ij} be the set of representative constraints for constraints in E_{ij} . In other words, R_{ij} is the set of constraints created by function `CREATE-H-CONSTRAINTS` for set E_{ij} . Let $\text{cost}(E_{ij}, x)$ and $\text{cost}(R_{ij}, x)$ be the numbers of constraints violated by solution x in E_{ij} and R_{ij} , respectively. We claim that the following bound holds.

Lemma 4.2 *For every heavy set E_{ij} ,*

$$\mathbb{E}[\text{cost}(R_{ij}, x^*)] \leq 8 \text{cost}(E_{ij}, x^*) + 2e^{-\varepsilon^2 t/8} |E_{ij}|. \quad (6)$$

Proof. If $\text{cost}(E_{ij}, x^*) \geq |E_{ij}|/4$, then $8 \text{cost}(E_{ij}, x^*) \geq 2|E_{ij}|$. Since $\text{cost}(R_{ij}, x^*) \leq |R_{ij}| = 2|E_{ij}|$, bound (6) holds. For the rest of the proof, we shall assume that $\text{cost}(E_{ij}, x^*) < |E_{ij}|/4$. We upper bound the probability that $\sigma_{ij} \neq x_i^* x_j^*$. To simplify the exposition, let us assume that $x_i^* x_j^* = 1$ (the proof for the case $x_i^* x_j^* = -1$ is analogous). Write

$$\Pr(\sigma_{ij} \neq x_i^* x_j^*) = \Pr(\sigma_{ij} \neq 1) = \Pr\left(\sum_{k:(i,j,k) \in E_{ij}} c_{ijk} \tilde{x}_k \leq 0\right).$$

The expectation of \tilde{x}_k is εx_k^* . Thus, $\mathbb{E}[c_{ijk} \tilde{x}_k] = \varepsilon c_{ijk} x_k^*$. For each constraint $x_i x_j x_k = c_{ijk}$ satisfied by x^* , we have $c_{ijk} = x_i^* x_j^* x_k^* = x_k^*$ and $c_{ijk} x_k^* = (x_k^*)^2 = 1$. Since the number of unsatisfied constraints is upper bounded by $|E_{ij}|/4$, we have

$$\mathbb{E}\left[\sum_{k:(i,j,k) \in E_{ij}} c_{ijk} \tilde{x}_k\right] = \sum_{k:(i,j,k) \in E_{ij}} \varepsilon c_{ijk} x_k^* \geq \varepsilon |E_{ij}|/2.$$

By Hoeffding's inequality,

$$\Pr(\sigma_{ij} \neq 1) = \Pr\left(\sum_{k:(i,j,k) \in E_{ij}} c_{ijk} \tilde{x}_k \leq 0\right) \leq e^{-\frac{2(\varepsilon |E_{ij}|/2)^2}{4|E_{ij}|}} = e^{-\frac{\varepsilon^2 |E_{ij}|}{8}}.$$

Recall that E_{ij} is a heavy set and thus contains at least t elements. Hence, $\Pr(\sigma_{ij} \neq x_i^* x_j^*) \leq e^{-\varepsilon^2 t/8}$.

If $\sigma_{ij} \neq x_i^* x_j^*$, then we bound the number of unsatisfied constraints by $|R_{ij}| = 2|E_{ij}|$. This gives us the second term in bound (6). If $\sigma_{ij} = x_i^* x_j^*$, then all constraints $x_i x_j = \sigma_{ij}$ in Ψ are satisfied by x^* . A constraint $x_k = \sigma_{ij} c_{ijk}$ in Ψ is not satisfied by x^* if and only if $x_i^* x_j^* x_k^* \neq c_{ijk}$. In other words, $x_k = \sigma_{ij} c_{ijk}$ is not satisfied only if constraint $x_i x_j x_k = c_{ijk}$ is not satisfied by x^* . The number of such constraints is $\text{cost}(|E_{ij}|, x^*)$. It is upper bounded by the first term in (6) \square

We now upper bound the expected number of representatives for light constraints violated by the ground truth solution x^* . Let R_i be the set of representative constraints for constraints in L_i . In other words, R_i is set of constraints created by function `CREATE-L-CONSTRAINTS` for set L_i . Let $\text{cost}(L_i, x)$ and $\text{cost}(R_i, x)$ be the number of constraints violated by solution x in L_i and R_i , respectively. We prove the following lemma.

Lemma 4.3 *For every set L_i ,*

$$\mathbb{E}[\text{cost}(R_i, x^*)] \leq 4 \text{cost}(L_i, x^*) + \frac{6t}{\varepsilon^4}. \quad (7)$$

Proof. If $\text{cost}(L_i, x^*) \geq |L_i|/4$, then the desired bound holds since, in this case, $\text{cost}(R_i, x^*) \leq |R_i| = |L_i| \leq 4 \text{cost}(L_i, x^*)$. So, we shall assume that $\text{cost}(L_i, x^*) < |L_i|/4$.

Recall that R_i contains $|L_i|$ identical constraints $x_i = \sigma_i$. These constraints are not satisfied by x^* if $x_i^* \neq \sigma_i$. Similarly to the proof of Lemma 6, we shall assume that $x_i^* = 1$ (the proof for the case $x_i^* = -1$ is analogous). We have

$$\Pr(\sigma_i \neq x_i^*) = \Pr(\sigma_i \neq 1) = \Pr\left(\sum_{j,k:(i,j,k) \in L_i} c_{ijk} \tilde{x}_j \tilde{x}_k \leq 0\right).$$

The expected value of each term $c_{ijk} \tilde{x}_j \tilde{x}_k$ is $\varepsilon^2 c_{ijk} x_j^* x_k^*$ since \tilde{x}_j and \tilde{x}_k are independent random variables with means εx_j^* and εx_k^* , respectively. Thus,

$$\mathbb{E}\left[\sum_{j,k:(i,j,k) \in L_i} c_{ijk} \tilde{x}_j \tilde{x}_k\right] = \sum_{j,k:(i,j,k) \in E_{ij}} \varepsilon^2 c_{ijk} x_j^* x_k^* \geq \varepsilon^2 |L_i|/2.$$

Here, we used that for every satisfied constraint $x_i x_j x_k = c_{ijk}$, we have $c_{ijk} x_j^* x_k^* = 1$ and the assumption that $\text{cost}(L_i, x^*) \leq |L_i|/4$. We now use McDiarmid's bounded difference inequality [McD89]. Let ρ_s be the number of occurrences of variable \tilde{x}_s in the sum $\Sigma_i = \sum_{j,k:(i,j,k) \in L_i} c_{ijk} \tilde{x}_j \tilde{x}_k$. If we change the value of variable \tilde{x}_s from -1 to 1 or from 1 to -1 , then sum Σ_i can change by at most $2\rho_s$, since all coefficients $c_{ijk} \in \{\pm 1\}$. Consequently, σ_i satisfies the bounded differences property with constants ρ_s . By McDiarmid's inequality,

$$\Pr(\sigma_i \neq 1) = \Pr\left(\sum_{j,k:(i,j,k) \in L_i} c_{ijk} \tilde{x}_j \tilde{x}_k \leq 0\right) \leq e^{-\frac{2(\varepsilon^2 |L_i|/2)^2}{\sum_s (2\rho_s)^2}} = e^{-\frac{\varepsilon^4 |L_i|^2}{8 \sum_s \rho_s^2}}.$$

The total number of random variables participating in the sum Σ_i counted with repetitions is $2|L_i|$. The number of times x_j appears in the sum is at most $|E_{ij}|$, which is at most t if E_{ij} is light. On the other hand, if E_{ij} is heavy, then so is every $(i, j, k) \in E(\Phi)$, and thus x_j does not appear in the sum at all. In either case, x_j appears in the sum at most t times. Thus, by Hölder's inequality, $\sum_s \rho_s^2 \leq \|\rho\|_1 \|\rho\|_\infty \leq 2|L_i| \cdot t$. Therefore, $\Pr(\sigma_i \neq 1) \leq e^{-\frac{\varepsilon^4 |L_i|}{16t}}$ and

$$\mathbb{E}[\text{cost}(R_i, x^*)] = \Pr(\sigma_i \neq 1) |L_i| \leq e^{-\frac{\varepsilon^4 |L_i|}{16t}} |L_i| \leq \frac{6t}{\varepsilon^4}.$$

Here, we used inequality $e^{-z} z \leq 1/e$ for $z = \frac{\varepsilon^4 |L_i|}{16t}$. □

To finish the proof of Lemma 4.1, we sum the bounds obtained in Lemmas 4.2 and 4.3 over all heavy and light sets. The total number of violated constraints in Ψ is upper bounded in expectation by

$$\underbrace{\sum_{i,j} 8 \text{cost}(E_{ij}, x^*) + \sum_i 4 \text{cost}(L_i, x^*)}_{\leq 24\delta |E(\Phi)|} + \underbrace{\sum_{i,j} 2e^{-\varepsilon^2 t/8} |E_{ij}|}_{6\delta |E(\Phi)|} + \underbrace{\sum_i \frac{6t}{\varepsilon^4}}_{6t\varepsilon^{-4}n}.$$

The first two terms together are upper bounded by the total number of violated constraints in Φ times 24, because each violated constraint can belong to at most three sets E_{ij} or three sets L_i . The third term is upper bounded by $6\delta |E(\Phi)|$ because $t = 8\varepsilon^{-2} \ln 1/\delta$ and $\sum_{i,j} |E_{ij}| \leq 3|E(\Phi)|$. Finally, the last term is upper bounded by $6\varepsilon^{-4}tn = 6(8\varepsilon^{-2} \ln 1/\delta) \cdot \varepsilon^{-4} \cdot n < 48\delta |E(\Phi)|$, because $|E(\Phi)| \geq \ln 1/\delta \cdot \delta^{-1} \varepsilon^{-6} \cdot n$. □

5 Hardness of Max k -Lin with Oracle Advice

In this section, we prove our hardness results for Max k -Lin problems in the advice model, i.e., Theorem 1.7 and Theorem 1.8. We build towards these results in several steps; we briefly outline them below:

- Firstly, in Section 5.1, we introduce our first ingredient, Lemma 5.1, where we show that $(1/2 + \varepsilon)$ -approximation for Max 3-Lin on n -variables requires $2^{c_\varepsilon n}$ -running time, assuming ETH. We show this by combining the $2^{c'_\varepsilon n}$ -running time lower bound for $(1 - \varepsilon, \varepsilon)$ -Gap Label Cover (Lemma 5.8), with Håstad's reduction from Label Cover to Max 3-Lin (Theorem 5.9).
- Next, in Section 5.2, we show that we can transfer the above hardness for Max 3-Lin to Max 4-Lin instances with up to linear average degree, using a simple combinatorial reduction (Corollary 5.13).
- Then in Section 5.3, we give a generic tool (Lemma 5.14), which shows that $2^{c_\varepsilon n}$ -time lower bound for a problem can be transferred to give polynomial time hardness for the same problem in the variable-subset advice model with parameter c''_ε .
- Finally, in Section 5.4, we combine the above ingredients to prove Theorems 1.7 and 1.8.

5.1 ETH and Hardness of Max 3-Lin

We first review the Exponential Time Hypothesis (ETH) and Linear PCP Conjecture and prove the following hardness result for Max 3-Lin.

Lemma 5.1 *Assume the ETH and Linear Size PCP Conjecture (see Conjecture 5.4). For some absolute constants $C', C'', C''' > 0$, $\varepsilon_0 \in (0, 1/2)$, and $\eta(\varepsilon) = C'/\sqrt{\log(1/\varepsilon)}$, the following holds. For every $\varepsilon \in (0, \varepsilon_0)$, there is no algorithm that given a Max 3-Lin instance \mathcal{I} on n variables and $2^{O(1/\varepsilon)C'''} n$ constraints, distinguishes between the following cases:*

$$\text{Yes Case : } \text{Val}(\mathcal{I}) \geq 1 - \eta(\varepsilon) \quad \text{and} \quad \text{No Case : } \text{Val}(\mathcal{I}) \leq 1/2 + \eta(\varepsilon).$$

in time $2^{2^{-(1/\varepsilon)C''}} n \cdot \text{poly}(n)$.

We point out that the above hardness result is folklore³, and it is well-known that it can be derived by combining ETH, Linear Size PCP conjecture, and the techniques from [Hås01]. For the sake of completeness, we provide a proof of this lemma in this section.

We remind the reader the Exponential Time Hypothesis (ETH) and definition of the Label Cover problem.

Conjecture 5.2 (ETH [IPZ01]) *There exists a constant $c \in (0, 1)$ such that for all large enough integers n , the 3-SAT problem on n variables cannot be solved in time $2^{cn} \text{poly}(n)$.*

Definition 5.3 (Label Cover) *An instance $\mathcal{L}(U, V, E, \Sigma_U, \Sigma_V, \{\pi_e\}_{e \in E})$ of Label Cover is a 2-CSP on a bipartite graph (U, V, E) . The left and right label sets of instance \mathcal{L} are Σ_U and Σ_V , respectively. Every edge $(u, v) \in E$ is identified with a **projection** constraint $\pi_{uv} : \Sigma_V \rightarrow \Sigma_U$. A labeling $\sigma : U \cup V \rightarrow \Sigma_U \cup \Sigma_V$ satisfies an edge (u, v) if $\pi_{uv}(\sigma(v)) = \sigma(u)$. We denote the maximum fraction of constraints that can be satisfied by a labeling by $\text{Val}(\mathcal{L})$. For every $0 < s < c \leq 1$, the objective of the (c, s) -Gap Label Cover problem on \mathcal{L} is to decide whether $\text{Val}(\mathcal{L}) \geq c$ or $\text{Val}(\mathcal{L}) \leq s$.*

We write $N = |U| + |V|$ and $K = |\Sigma_U| + |\Sigma_V|$ to denote the number of variables and labels in \mathcal{L} .

Linear Size PCPs. We will use the following conjecture on linear size PCPs.

³See e.g., the TCS Stack Exchange post [Exc] for a discussion on quantitatively similar bounds for Max 3-SAT.

Conjecture 5.4 (Linear Size PCP Conjecture, Conjecture 4.2 in [Din16]) *For some $C_1, C_2 > 0$ and all sufficiently small $\varepsilon > 0$, there exists a polynomial-time reduction from 3-SAT to Label Cover that satisfies the following properties. Assume that the reduction maps a 3-SAT instance Φ of size m to a Label Cover instance $\mathcal{L} = (U, V, E, \Sigma_U, \Sigma_V, \{\pi_e\}_{e \in E})$. Then,*

- $|U|, |V| \leq (1/\varepsilon)^{C_1} \cdot m$.
- $|\Sigma_U|, |\Sigma_V| \leq (1/\varepsilon)^{C_2}$.
- If $\text{Val}(\Phi) = 1$, then $\text{Val}(\mathcal{L}) = 1$.
- If $\text{Val}(\Phi) < 1$, then $\text{Val}(\mathcal{L}) \leq \varepsilon$.

The above conjecture posits that for any ε , there exists a reduction from 3-SAT on m -clauses to the $(1, \varepsilon)$ -Gap-Label-Cover problem, such that the number of variables is at most $\text{poly}(1/\varepsilon)$ times the number of clauses, and the label set size is also bounded by $\text{poly}(1/\varepsilon)$. It is instructive to compare this with the result of Moshkovitz–Raz [MR08], which gives a reduction from 3-SAT to $(1, \varepsilon)$ -label cover with a weaker guarantee of $O_\varepsilon(\log m)$ -blow up in the instance size. The Linear Size PCP Conjecture states that one can get (qualitatively) the same parameters in the reduction from 3-SAT to $(1, \varepsilon)$ -Gap-Label Cover without the additional $O(\log m)$ blow up in the number of variables.

Degree Increment Lemma. We shall also need the following lemma which says that one can increase the average degree of the Label Cover instance while preserving its optimal value.

Lemma 5.5 *The following holds for every integer $\ell \geq 1$. Let $\mathcal{L} = (U, V, E, \Sigma_U, \Sigma_V, \{\pi_e\}_{e \in E})$ be a Label Cover instance. Then there exists a polynomial time procedure that constructs a Label Cover instance $\mathcal{L}' = (U', V', E', \Sigma_U, \Sigma_V, \{\pi'_e\}_{e \in E'})$ which satisfies the following properties:*

- $\text{Val}(\mathcal{L}') = \text{Val}(\mathcal{L})$.
- $|U'| = \ell|U|$ and $|V'| = \ell|V|$.
- $|E'| = \ell^2|E|$.

The above lemma is folklore; we provide a proof of it in Appendix A for the sake of completeness. The following corollary follows immediately by combining Conjecture 5.4 and Lemma 5.5.

Corollary 5.6 *Assume Conjecture 5.4. For some $C_1, C_2 > 0$ and all sufficiently small $\varepsilon > 0$, the following holds for every integer $\ell \geq 1$. There exists a polynomial-time reduction from 3-SAT to Label Cover that satisfies the following properties. Assume that the reduction maps a 3-SAT instance Φ of size m to a Label Cover instance $\mathcal{L} = (U, V, E, \Sigma_U, \Sigma_V, \{\pi_e\}_{e \in E})$. Then,*

- $|U|, |V| \leq (1/\varepsilon)^{C_1} \cdot \ell \cdot m$.
- $|\Sigma_U|, |\Sigma_V| \leq (1/\varepsilon)^{C_2}$.
- If $\text{Val}(\Phi) = 1$, then $\text{Val}(\mathcal{L}) = 1$.
- If $\text{Val}(\Phi) < 1$, then $\text{Val}(\mathcal{L}) \leq \varepsilon$.
- The average degree of \mathcal{L} is at least ℓ .

Sparsification Lemma. Additionally, we will use the following sparsification lemma for 3-SAT by Calabro, Impagliazzo, and Paturi [CIP06].

Lemma 5.7 (Sparsification Lemma [CIP06]) *For every $\gamma > 0$, there exists a deterministic algorithm that given a 3-SAT formula F on n -variables outputs a sequence of 3-SAT formulas F_1, \dots, F_s such that*

1. $s \leq 2^{\gamma n}$.
2. F is satisfiable if and only if at least one of F_1, \dots, F_s is satisfiable.
3. Each formula F_i is on n variables. The number of clauses in each F_i is at most $(1/\gamma)^9 \cdot n$.
4. The algorithm runs in time $2^{\gamma n} \cdot \text{poly}(n)$, where the degree of the polynomial may depend on γ .

5.1.1 Reduction to Gap Label Cover

In this section, we reduce 3-SAT to Gap Label Cover.

Lemma 5.8 *Assume ETH and the Linear Size PCP conjecture. For some constants $C > 2C_2 > 0$ and every choice of constants $0 < \varepsilon < c/2$, there exists no algorithm for $(1, \varepsilon)$ -Gap Label Cover that given an instance with at most N vertices, $K = (1/\varepsilon)^{C_2}$ labels, and at least $2^{5(1/\varepsilon)^{C_2}} N$ constraints, decides whether the instance is completely satisfiable or at most ε satisfiable in time $2^{2^{-5(1/\varepsilon)^C} \cdot NK} \text{poly}(N)$. Here c is the constant from Conjecture 5.2.*

Proof. Let C_1, C_2 be the constants from Conjecture 5.4 and let

$$C := C_1 + 2C_2 + 10.$$

We describe a Turing reduction from the 3-SAT problem to the Gap Label Cover problem:

Reduction. Given a 3-SAT instance Φ on n variables, we do the following:

- Run the sparsification algorithm from Lemma 5.7 on Φ with $\gamma = \varepsilon$ and get 3-SAT instances Φ_1, \dots, Φ_s .
- For every $i \in [s]$, run the reduction from Corollary 5.6 on Φ_i with $\ell = 2^{5(1/\varepsilon)^{C_2}}$ and get a $(1, \varepsilon)$ -Gap Label Cover instance \mathcal{L}_i .
- Solve each of the $(1, \varepsilon)$ -Gap Label Cover instances \mathcal{L}_i and output YES if at least one of the \mathcal{L}_i is a YES instance.

We make a couple of immediate observations. Note that for every $i \in [s]$, the formula Φ_i has n variables and at most $m := (1/\gamma)^9 \cdot n$ clauses. Furthermore, the corresponding label cover instance \mathcal{L}_i has at most $N = (1/\varepsilon)^{C_1} \cdot \ell \cdot m = 2^{5(1/\varepsilon)^{C_2}} (1/\varepsilon)^{C_1+9} n$ variables, and $K \leq (1/\varepsilon)^{C_2}$ labels. Furthermore, it has at least $2^{5K} N$ constraints.

Completeness. Suppose Φ is satisfiable. Then, for some $i \in [s]$, the corresponding 3-SAT instance Φ_i is satisfiable, and thus the corresponding Label Cover instance \mathcal{L}_i is also satisfiable.

Soundness. Suppose Φ is not satisfiable, then for every $i \in [s]$, Φ_i is not satisfiable, and hence $\text{Val}(\mathcal{L}_i) < \varepsilon$.

Now suppose there exists a $2^{2^{-5(1/\varepsilon)^C} \cdot NK} \text{poly}(N)$ time algorithm for $(1, \varepsilon)$ -Gap Label Cover. By running it on every instance \mathcal{L}_i , we solve 3-SAT in time:

$$2^{\varepsilon n} \cdot 2^{2^{-5(1/\varepsilon)^C} \cdot (2^{5(1/\varepsilon)^{C_2}} (1/\varepsilon)^{C_1+9} n \cdot (1/\varepsilon)^{C_2})} \text{poly}(N) \leq 2^{2\varepsilon n} \text{poly}(n),$$

which refutes ETH if $\varepsilon < c/2$. □

5.1.2 Reduction from Gap Label Cover to Max 3-Lin

Here, we recall Håstad's 3-bit PCP-based reduction from Label Cover to Max 3-Lin.

Theorem 5.9 ([Hås01]) *There exists an increasing continuous function $\eta : [0, 1] \rightarrow [0, 1]$ with $\eta(0) = 0$ for which the following holds. Given an instance $\mathcal{L} = (U, V, E, \Sigma_U, \Sigma_V, \{\pi_e\}_{e \in E})$ of $(1, \varepsilon)$ -Gap Label Cover with $n := |U| + |V|$ variables, and $k := |\Sigma_U| + |\Sigma_V|$ labels, and average degree at least 2^{5k} , there exists a randomized polynomial-time reduction that with probability at least $1 - 2^{-k/2}$, outputs an instance of Max 3-Lin \mathcal{I}'' such that*

- If $\text{Val}(\mathcal{L}) = 1$, then $\text{Val}(\mathcal{I}'') \geq 1 - \eta(\varepsilon)$.
- If $\text{Val}(\mathcal{L}) \leq \varepsilon$, then $\text{Val}(\mathcal{I}'') \leq 1/2 + \eta(\varepsilon)$.
- The number of variables in \mathcal{I}'' is $n' = 2^{|\Sigma_U|} |U| + 2^{|\Sigma_V|} |V|$, and the number of constraints is at most $8n'/\varepsilon^2$.
- All constraints in \mathcal{I}'' are non-edge-weighted and distinct.

Additionally, $\eta(\varepsilon) = C'/\sqrt{\log(1/\varepsilon)}$ for some constant $C' > 0$.

Proof. The reduction consists of two parts:

- First, we use Håstad's reduction [Hås01] to reduce $(1, \varepsilon)$ -Gap Label Cover to $(1 - \eta(\varepsilon), 1/2 + \eta(\varepsilon))$ -Gap Max 3-Lin⁴. Note that this will output a Max 3-Lin instance \mathcal{I}' where the constraints are edge-weighted.
- Then we use a sub-sampling step to sample a non-edge-weighted instance \mathcal{I}'' with no duplicate constraints.

To begin with, we recall the reduction from [Hås01] for the sake of completeness and observe the bounds on the number of variables and running time of the reduction. As is standard, the reduction is stated in the form of a dictatorship test, which we describe in Figure 2:

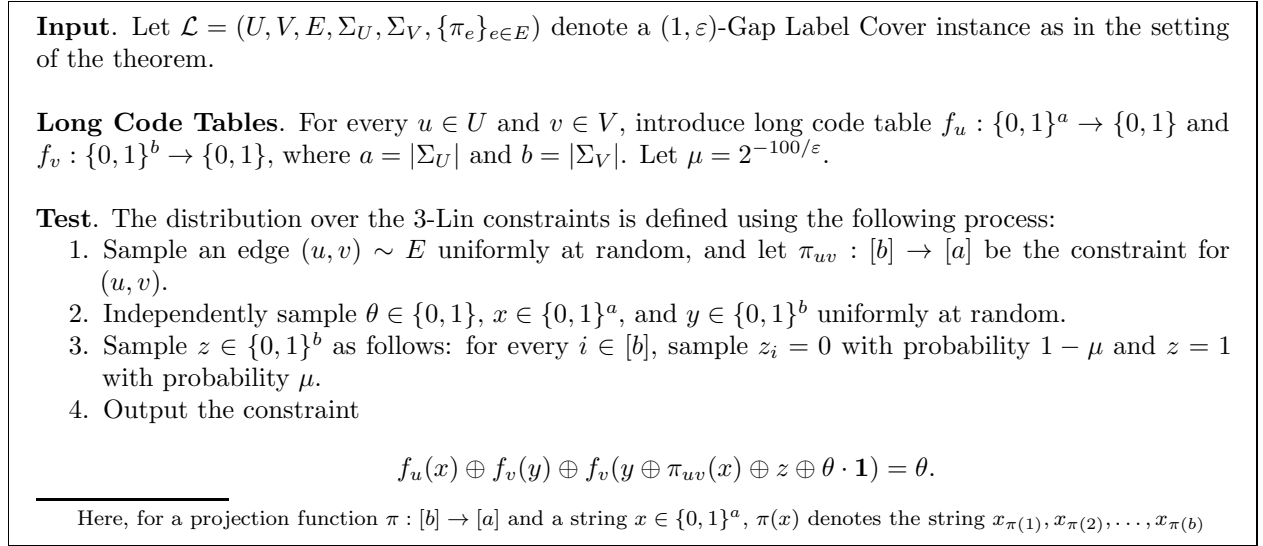


Figure 2: PCP Reduction to Max 3-Lin

It is easy to see that the resulting Max 3-Lin instance has $n' := 2^a|U| + 2^b|V| \leq 2^k \cdot n$ variables. The reduction runs in time $2^{O(k/\varepsilon)} \text{poly}(n)$. Furthermore, the analysis from Håstad's paper [Hås01] shows that if \mathcal{L} is satisfiable, then the resulting instance has value at least $1 - \mu \geq 1 - \eta(\varepsilon)$, and if \mathcal{L} has value at most ε , then \mathcal{I} has value at most $1/2 + \eta(\varepsilon)$.

Removing Edge Weights. Note that the instance (say, \mathcal{I}) output by the above reduction is constraint-weighted; in particular, the weights define a probability distribution (say, ν) over the set of constraints. Now consider the following new unweighted instance \mathcal{I}'' that is constructed as follows:

- First, we construct the following intermediate instance \mathcal{I}' . The variable set of \mathcal{I}' is the same as that of \mathcal{I} . Let $\eta = \eta(\varepsilon)$. For $m := 8n'/\eta^2$, sample constraints $e_1, e_2, \dots, e_m \sim \nu$ independently with replacement, and include the constraints e_1, \dots, e_m in \mathcal{I}' .
- For every constraint that has multiple copies in \mathcal{I}' , delete all but one copy of the constraint. Call the resulting duplicate free instance \mathcal{I}'' .

Clearly, by definition, the resulting instance still has $n' = 2^a|U| + 2^b|V|$ variables, and at most $m = 8n'/\eta^2$ constraints. It remains to be shown that the optimal value of \mathcal{I}'' is nearly identical to \mathcal{I} with high probability. We show this using a couple of claims. We first show that the intermediate instance \mathcal{I} has nearly the same fraction of satisfied edges as \mathcal{I}' w.r.t. every labeling.

⁴For any $1 \geq a > b \geq 1/2$, an instance of (a, b) -Gap Max 3-Lin is the decision problem, where the input is a Max 3-Lin instance, and the objective is to distinguish between the cases whether the optimal value of the instance is at least a or at most b .

Claim 5.10 *The following holds with probability at least $1 - e^{-O(n')}$ over the choice of \mathcal{I}' . Suppose $m \geq 8n'/\eta^2$. Then for every labeling σ of \mathcal{I} , we have that*

$$\left| \Pr_{e \sim \nu}(\sigma \text{ satisfies } e) - \Pr_{e \sim \mathcal{I}'}(\sigma \text{ satisfies } e) \right| \leq \eta,$$

where ν is the distribution over the constraints from the reduction in Figure 2.

Proof. Fix a labeling σ of \mathcal{I} , and let $\alpha \in [0, 1]$ denote the fraction of constraints satisfied by the labeling σ in \mathcal{I} . Now, for every $i \in [m]$, let $X_{\sigma,i}$ indicate whether the i^{th} sampled constraint is satisfied by σ . Note that $\mathbb{E}[X_{\sigma,i}] = \alpha$. Then using Hoeffding's bound, we have that

$$\Pr_{\mathcal{I}'} \left(\left| \sum_{i \in [m]} X_{\sigma,i} - \alpha m \right| \geq \eta m \right) \leq e^{-\frac{\eta^2 m}{4}} \leq e^{-2n'},$$

where the last inequality uses $m = 8n'/\eta^2$. The claim now follows by taking a union bound over all $2^{n'}$ possible labelings, and using our lower bound on m . \square

Next, we show that the number of duplicate edges in \mathcal{I}' is not too large.

Claim 5.11 *Suppose $|E| \geq 2^{5k}n$. With probability at least $1 - 2^{-2k}$, the number of duplicate constraints in \mathcal{I}' is at most ηm .*

Proof. Firstly, note that by definition of the distribution over the constraints ν (described in Figure 2), we have that for any constraint e , $\nu(e) := \Pr_{e' \sim \nu}(e' = e) \leq 2^{-(a+b)}/|E|$. This is due to the observation that every constraint that can be generated by the process described in Figure 2 is specified by the choice of (i) the label cover edge (u, v) , (ii) the choice of the strings x, y, z and the bit θ . Furthermore, since (u, v) is sampled uniformly from E and independently, x, y are sampled uniformly from $\{0, 1\}^a$ and $\{0, 1\}^b$ respectively, it follows that $\nu(e) \leq |E|^{-1} \cdot 2^{-(a+b)}$.

Therefore, for any $i, j \in [m]$, we have

$$\Pr_{\mathcal{I}'}(e_i = e_j) = \sum_e \nu(e)^2 \leq \frac{2^{-(a+b)}}{|E|} \sum_e \nu(e) = \frac{2^{-k}}{|E|}.$$

Therefore, for a fixed constraint e_i , \mathcal{I}' contains another copy of e_i with probability at most $2^{-k}m/|E|$. Hence, the expected number of constraints that have another copy in \mathcal{I}' is at most

$$\frac{2^{-k}}{|E|} \cdot m^2 \leq \frac{2^{-k}}{|E|} \left(\frac{2^{2k}n}{\eta^2} \right) \cdot m \leq 2^{-2k}\eta m$$

where the first inequality uses the bound

$$m = 8n'/\eta^2 \leq 8 \cdot 2^k n / \eta^2 \leq 2^{2k} n / \eta^2,$$

and the second inequality follows from the assumption that $|E| \geq 2^{5k}n$. The claim now follows using Markov's inequality. \square

Therefore, by combining the guarantees of Claims 5.10 and 5.11, we have that with probability at least $1 - 2^{-k}$, the following holds simultaneously for every labeling of \mathcal{I} : if a labeling σ satisfies α_σ fraction of constraints in \mathcal{I} , then it satisfies $\alpha_\sigma \pm 2\eta$ fraction of constraints in \mathcal{I}'' . This concludes the proof of the lemma. \square

5.1.3 Proof of Lemma 5.1

Let $C > 2C_2 > 0$ be the constants from Lemma 5.8. Let \mathcal{L} be a $(1, \varepsilon)$ -Label Cover instance with N variables and K Labels as in the setting of Lemma 5.8. Note that by the guarantee of Lemma 5.8, we have $K \leq (1/\varepsilon)^{C_2}$, and the average degree of the instance is at least 2^{5K} . Then we use the reduction from Theorem 5.9 to construct a $(1 - \eta(\varepsilon), 1/2 + \eta(\varepsilon))$ -Gap Max 3-Lin instance \mathcal{I} with $n \leq 2^{2K} \cdot N$ variables, and at most $8n/\varepsilon^2$ constraints. Furthermore, from the guarantee of Theorem 5.9, we know that \mathcal{I} is a YES instance if and only if \mathcal{L} is a YES instance.

Now, suppose there exists a $2^{2^{-10(1/\varepsilon)^C} n} \cdot \text{poly}(n)$ -time algorithm for solving $(1 - \eta(\varepsilon), 1/2 + \eta(\varepsilon))$ -Gap Max 3-Lin. Then there exists an algorithm for $(1, \varepsilon)$ -Gap Label Cover on N variables, K labels, and $2^{5K}N$ constraints, that runs in time

$$2^{2^{-10(1/\varepsilon)^C} n} \cdot \text{poly}(n) \leq 2^{2^{-10(1/\varepsilon)^C} 2^{2(1/\varepsilon)^{C_2}} N} \cdot 2^{O(K)} \text{poly}(N) \leq 2^{2^{-(1/\varepsilon)^C} NK} \cdot \text{poly}(N),$$

where the first inequality uses the bound $K \leq (1/\varepsilon)^{C_2}$, and the second inequality is due to the definition of $C \geq 2C_2$. Since this contradicts Lemma 5.8, it follows that there is no algorithm for $(1 - \eta(\varepsilon), 1/2 + \eta(\varepsilon))$ -Gap Max 3-Lin instances on n variables with average degree at most $2^{O((1/\varepsilon)^{C_2})}$.

5.2 Hardness of Max 4-Lin Instances with Large Average Degree

In this section, we reduce bounded degree Max 3-Lin instances to Max 4-Lin instances with large degrees. It will be more convenient to work with 3- and 4-Lin constraints over the 0-1 alphabet. Accordingly, the constraints will be of the form $x_i \oplus x_j \oplus x_k = b_{ijk}$ or $x_i \oplus x_j \oplus x_k \oplus x_l = b_{ijkl}$. We will say that the average degree of a 3-Lin or 4-Lin instance \mathcal{I} is the ratio between the number of constraints $|E|$ and variables $|V|$ in \mathcal{I} , divided by the arity r of the constraints, i.e., the average degree is defined as $|E|/(|V|r)$.

Lemma 5.12 *For $0 < \varepsilon, \delta \leq 1/2$, and a fixed integer parameter $t \in \mathbb{N}$ there exists a polynomial-time deterministic reduction from Max 3-Lin to Max 4-Lin that satisfies the following properties.*

- If the input instance \mathcal{I} is $(1 - \varepsilon)$ -satisfiable, then the output instance is also $(1 - \varepsilon)$ -satisfiable.
- If the output instance \mathcal{I}' is $(1/2 + \delta)$ -satisfiable, then the input instance \mathcal{I} is $(1/2 + \delta)$ -satisfiable.
- The number of variables in \mathcal{I}' is $n + t$, where n is the number of variables in \mathcal{I} .
- The average degree of \mathcal{I}' is $ndt/(n + t)$.

Proof. Let $\mathcal{I} = (V = [n], E)$ be an instance of Max 3-Lin. Given \mathcal{I} , we construct an instance of Max 4-Lin $\mathcal{I}' = (V', E')$ as follows.

Vertex Set. Let $V' = V \cup V_1$, where $V_1 := \{y_1, \dots, y_t\}$ is a set of t new variables.

Constraint Set. For every constraint $e := x_i \oplus x_j \oplus x_k = b_{ijk}$ and every variable $y_r \in V_1$, we add constraint

$$e^r := x_i \oplus x_j \oplus x_k \oplus y_r = b_{ijk}. \quad (8)$$

We will refer to the cloud of constraints corresponding to e as \mathcal{C}_e . Note that the degree of vertices in V and V_1 are dt and nd , respectively, which implies that the minimum degree of \mathcal{I}' is $\min\{dt, nd\}$.

Analysis. First, let us assume that $\text{Val}(\mathcal{I}) \geq 1 - \varepsilon$. Denote an optimal labeling by σ^* . We extend σ^* to a labeling σ' of V' by labeling all the variables in V_1 as 0. The resulting labeling satisfies at least a $1 - \varepsilon$ fraction of type-1 constraints and all type-2 constraints. Thus, it satisfies at least a $1 - \varepsilon/2$ fraction of all the constraints in \mathcal{I}' . Therefore, $\text{Val}(\mathcal{I}') \geq 1 - \varepsilon/2$.

Now let us assume that \mathcal{I}' has an assignment, say $\sigma' : V' \rightarrow \{0, 1\}$, that satisfies at least a $(1/2 + \delta)$ -fraction of the constraints in \mathcal{I}' . Consider the following randomized labeling scheme.

- Sample $r \sim [t]$ uniformly at random.
- Define labeling $\sigma : V \rightarrow \{0, 1\}$ as $\sigma(x) = \sigma'(x) \oplus y_r$ for every $x \in V$.

Note that under the above randomized scheme, the expected fraction of constraints satisfied by σ in \mathcal{I} is

$$\begin{aligned} & \mathbb{E}_\sigma \mathbb{E}_{e=(i,j,k) \sim E} \left[\mathbb{1}(\sigma(i) \oplus \sigma(j) \oplus \sigma(k) = b_{ijk}) \right] \\ &= \mathbb{E}_{e=(i,j,k) \sim E} \mathbb{E}_{r \sim [t]} \left[\mathbb{1}(\sigma'(i) \oplus \sigma'(j) \oplus \sigma'(k) + y_r = b_{ijk}) \right] \\ &= \frac{1}{2} + \delta, \end{aligned}$$

which implies that for at least one choice of y_r , the corresponding labeling σ of V satisfies at least $(1/2 + \delta)$ fraction of variables in \mathcal{I} , which concludes the soundness analysis. \square

By combining Lemmas 5.1 and 5.12, we get the following corollary.

Corollary 5.13 *Assume ETH and the Linear Size PCP conjecture. There exists $\delta_0 \in (0, 1)$ such that the following holds. Then for every $\delta \in (0, \delta_0]$ there exists $d(\delta) \in \mathbb{N}$ and $\varepsilon(\delta) \in (0, 1)$ such that given a n -variable Max 3-Lin instance \mathcal{I} of average degree within $[d_0(\delta), n]$, there is no $2^{\varepsilon(\delta)n}$ time algorithm that distinguish between the case when \mathcal{I} is at least $(1 - \delta)$ satisfiable and the case when \mathcal{I} is at most $1/2 + \delta$ -satisfiable.*

Proof. Let η and ε_0 be as in Lemma 5.1. Let $\delta_0 = \eta^{-1}(\varepsilon_0)$, and fix a $\delta \in (0, \delta_0)$. Let $\varepsilon = \eta^{-1}(\delta)$. Note that since η is increasing, we have that $\varepsilon \in (0, \varepsilon_0]$. Now, from Lemma 5.1. Then from Lemma 5.1 we know that there is no $2^{-c(\varepsilon)n}$ time algorithm which can distinguish between the following cases for a Max-3-Lin instance on n -variables and $d(\varepsilon)n$ constraints, where $d(\varepsilon) = 2^{O(1/\varepsilon)^{C'''}}$:

$$\text{YES Case : } \text{Val}(\mathcal{I}) \geq 1 - \delta \quad \text{and} \quad \text{NO Case : } \text{Val}(\mathcal{I}) \leq 1/2 + \delta,$$

since $\delta = \eta(\varepsilon)$. Now, we instantiate Lemma 5.12 with the above instance and a fixed choice of $t \in \{1, 2, \dots, n\}$, which will yield a 4-Lin instance \mathcal{I}' on $N := n + t$ variables and ndt constraints with the following guarantees:

- If \mathcal{I} is a YES instance (as above), then $\text{Val}(\mathcal{I}') \geq 1 - \delta$.
- If \mathcal{I} is a NO instance (as above), then $\text{Val}(\mathcal{I}') \leq 1/2 + \delta$

Since there is no $2^{c(\varepsilon(\delta))n}$ -time algorithm that can distinguish between the YES and NO cases of \mathcal{I} , since $N \in [n, 2n]$, it follows that there is no $2^{c(\varepsilon)N/2} = 2^{c(\eta^{-1}(\delta))N/2}$ -time algorithm that can distinguish between the YES and NO cases of \mathcal{I}' . Finally, also note that for any fixed choice of t , the average degree is $ndt/n + t = \Theta(d \cdot \min\{t, n\})$, from which the claim follows. \square

5.3 Hardness with Oracle Advice

In this section, we combine the hardness results from the previous section to prove ETH-based lower bounds for algorithms with oracle advice in the variable subset model. Our key tool here is the following lemma that shows that oracle advice can be simulated deterministically in near sub-exponential time. Let us say that an algorithm \mathcal{A} is a (c, s) -approximation algorithm if given a c -satisfiable instance, it finds a solution that satisfies at least an s -fraction of the constraints.

Lemma 5.14 *Suppose there exists a polynomial-time algorithm \mathcal{A} for Max r -Lin that given a c -satisfiable instance \mathcal{I} and advice with parameter ε in the variable-subset model, outputs a solution satisfying an s -fraction of the constraints with probability at least 0.9 over the choice of the advice string. Then there exists a deterministic (c, s) -approximation algorithm \mathcal{A}' for Max r -Lin that runs in time $2^{(\varepsilon \log(4/\varepsilon))n} \text{poly}(n)$.*

Proof. Consider the following algorithm:

Input: Max r -Lin instance \mathcal{I} .

1. For every $S \subseteq [n]$ of size at most $2\varepsilon n$ and $\sigma_S \in \{0, 1\}^S$, run algorithm \mathcal{A} with advice string (S, σ_S) and let $x_{S, \sigma_S} \in \{0, 1\}^n$ denote the corresponding labeling returned by algorithm.

2. Output the best assignment among all the assignments $\{x_{S,\sigma_S}\}_{S,\sigma_S}$ computed in the above step.

The running time of the algorithm is at most

$$\sum_{t=0}^{2\epsilon n} \binom{n}{t} \cdot 2^t \text{poly}(n) \leq \sum_{t=0}^{2\epsilon n} 2^{H(t/n)n} \cdot 2^t \text{poly}(n) = 2^{H(\epsilon)n + 2\epsilon n} \text{poly}(n) = 2^{(\epsilon \log_2 \frac{1}{\epsilon})n} \text{poly}(n).$$

where $H(x) = x \log_2 \frac{1}{x}$ is the Shannon entropy function.

Let x^* be the ground-truth solution. We assume that it satisfies at least a c -fraction of the constraints. Sample a random set S by including every $i \in [n]$ with probability ϵ (all decisions are independent). Then we are guaranteed that \mathcal{A} finds a solution that satisfies an s fraction of the constraints with probability at least 0.9 given advice $(S, X^*|_S)$ (here $x^*|_S$ is the restriction of x^* to S). The probability that $|S| \leq 2\epsilon n$ is $1 - e^{-\Omega(\epsilon n)}$. Thus, for all sufficiently large n , with positive probability, we have that (1) \mathcal{A} , with advice $(S, X^*|_S)$, finds a solution satisfying at least a c fraction of the constraints and (2) $|S| \leq 2\epsilon n$. Let S_0 be one of such sets S . When our algorithm goes over all S of size at most $2\epsilon n$, it also tries $S = S_0$. Then, for this $S = S_0$, it goes over assignments σ_S including $x^*|_{S_0}$. For this choice of $(S, \sigma_S) = (S_0, x^*|_{S_0})$, \mathcal{A} will find a solution satisfying at least a c -fraction of the constraints. As our algorithm returns the best of the solutions it finds at Step 1, it will output this or another solution satisfying at least a c -fraction of the constraints. \square

5.4 Proofs of Theorem 1.7 and Theorem 1.8

We now combine the ingredients from the previous section to prove the following hardness results:

Theorem 1.7 *Assume that the Exponential Time Hypothesis (ETH) and Linear Size PCP Conjecture hold. For every $\delta > 0$, there exists $\epsilon_0 = \epsilon_0(\delta)$ such that for every $\epsilon \in [0, \epsilon_0]$, there is no polynomial time algorithm for Max 3-Lin in the variable subset advice model with parameter ϵ that given a $(1 - \delta)$ -satisfiable instance returns a solution satisfying at least a $(1/2 + \delta)$ -fraction of the constraints with probability at least 0.9 over the random advice.*

Proof. We apply Lemma 5.1 with ϵ chosen so that $\delta = \eta(\epsilon)$. We get that there is no algorithm that decides whether a 3-Lin instance is at most $\frac{1}{2} + \delta$ or at least $1 - \delta$ satisfiable in time $2^{\left(2^{-(1/\epsilon)^{C''}}\right)_n} \text{poly}(n)$. Define ϵ_0 so that $\epsilon_0 \log_2 \frac{4}{\epsilon_0} \leq 2^{-(1/\epsilon)^{C''}}$. Now the theorem statement follows from Lemma 5.14. \square

Theorem 1.8 *Assume that the Exponential Time Hypothesis (ETH) and Linear Size PCP Conjecture hold. Then there exists a constant $\delta_0 \in (0, 1)$ such that for every $\delta \in (0, \delta_0)$, there exists $d(\delta) \in \mathbb{N}$ and $\epsilon(\delta) \in (0, 1)$ for which the following statement holds. For every $\epsilon \in (0, \epsilon(\delta))$, $n \geq n(\delta)$, and $d \in [d(\delta), n]$, there is no polynomial time algorithm for Max 4-Lin in the variable subset advice model with parameter ϵ that, given a $(1 - \delta)$ -satisfiable Max 4-Lin instance with n variables and at least $\Omega(dn)$ constraints, returns a solution satisfying at least a $(1/2 + \delta)$ -fraction of the constraints with probability at least 0.9 over the random advice.*

Proof. The proof of the theorem is almost identical to the proof of Theorem 1.7 – we simply use Corollary 5.13 in place of Lemma 5.1. Let δ_0 be as in Corollary 5.13, and for a fixing of $\delta \in (0, \delta_0)$, let $\epsilon(\delta)$ be as in Lemma 5.13. Define ϵ_0 so that $\epsilon_0 \log_2 \frac{4}{\epsilon_0} \leq \epsilon(\delta)$. The desired result then follows from Lemmas 5.1 and 5.14. \square

References

- [BC24] Xingjian Bai and Christian Coester. Sorting with predictions. *Advances in Neural Information Processing Systems*, 36, 2024.
- [BDSV18] Maria-Florina Balcan, Travis Dick, Tuomas Sandholm, and Ellen Vitercik. Learning to branch. In *International conference on machine learning*, pages 344–353. PMLR, 2018.

- [BEX24] Evripidis Bampis, Bruno Escoffier, and Michalis Xeferis. Parsimonious learning-augmented approximations for dense instances of np-hard problems. *arXiv preprint arXiv:2402.02062*, 2024.
- [CIP06] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. A duality between clause width and clause density for sat. In *Conference on Computational Complexity*, 7–pp, 2006.
- [CL23] Ivan Carvalho and Ramon Lawrence. Learnedsort as a learning-augmented samplesort: analysis and parallelization. In *Proceedings of the International Conference on Scientific and Statistical Database Management*, pages 1–9, 2023.
- [CdGLP24] Vincent Cohen-Addad, Tommaso d’Orsi, Anupam Gupta, Euiwoong Lee, and Debmalya Panigrahi. Max-cut with ε -accurate predictions. *arXiv preprint arXiv:2402.18263*, 2024.
- [DH09] Nikhil R Devanur and Thomas P Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the Conference on Electronic Commerce*, pages 71–78, 2009.
- [DILMV21] Michael Dinitz, Sungjin Im, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Faster matchings via learned duals. *Advances in neural information processing systems*, 34:10393–10406, 2021.
- [Din16] Irit Dinur. Mildly exponential reduction from gap-3SAT to polynomial-gap label-cover. In *Electronic colloquium on computational complexity ECCC; research reports, surveys and books in computational complexity*, page 128, 2016.
- [Exc] TCS Stack Exchange. Super-polynomial time approximation algorithms for MAX 3SAT. <https://cstheory.stackexchange.com/questions/9350/super-polynomial-time-approximation-algor>. Accessed: 2024-04-02.
- [FKL02] Uriel Feige, Marek Karpinski, and Michael Langberg. Improved approximation of Max-Cut on graphs of bounded degree. *Journal of Algorithms*, 43(2):201–219, 2002.
- [GW95] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.
- [GP19] Sreenivas Gollapudi and Debmalya Panigrahi. Online algorithms for rent-or-buy with expert advice. In *Proceedings of the International Conference on Machine Learning*, pages 2319–2327. PMLR, 2019.
- [Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001.
- [HK22] Jun-Ting Hsieh and Pravesh K Kothari. Approximating max-cut on bounded degree graphs: tighter analysis of the FKL algorithm. *arXiv preprint arXiv:2206.09204*, 2022.
- [HIKV19] Chen-Yu Hsu, Piotr Indyk, Dina Katabi, and Ali Vakilian. Learning-based frequency estimation algorithms. In *International Conference on Learning Representations*, 2019.
- [IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.
- [Kar10] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 2010.
- [KKMO07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O’Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007.
- [KBCDP18] Tim Kraska, Alex Beutel, Ed H Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures. In *Proceedings of the International Conference on Management of Data*, pages 489–504, 2018.
- [LLMV20] Silvio Lattanzi, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Online scheduling via learned weights. In *Proceedings of the Symposium on Discrete Algorithms*, pages 1859–1877, 2020.

- [LRSZ21] Pinyan Lu, Xuandi Ren, Enze Sun, and Yubo Zhang. Generalized sorting with predictions. In *Proceedings of the Symposium on Simplicity in Algorithms*, pages 111–117, 2021.
- [LV21] Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. *Journal of the ACM (JACM)*, 68(4):1–25, 2021.
- [MNS07] Mohammad Mahdian, Hamid Nazerzadeh, and Amin Saberi. Allocating online advertisement space with unreliable estimates. In *Proceedings of the Conference on Electronic Commerce*, pages 288–294, 2007.
- [MM17] Konstantin Makarychev and Yury Makarychev. Approximation algorithms for CSPs. In *Dagstuhl Follow-Ups*, volume 7. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
- [McD89] Colin McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.
- [Mit18] Michael Mitzenmacher. A model for learned bloom filters and optimizing by sandwiching. *Advances in Neural Information Processing Systems*, 31, 2018.
- [Mit20] Michael Mitzenmacher. Scheduling with predictions and the price of misprediction. In *Proceedings of the Innovations in Theoretical Computer Science Conference*, 2020.
- [MR08] Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *Journal of the ACM (JACM)*, 57(5):1–29, 2008.
- [MV17] Andres Munoz and Sergei Vassilvitskii. Revenue optimization with approximate bid predictions. *Advances in Neural Information Processing Systems*, 30, 2017.
- [PSK18] Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ML predictions. *Advances in Neural Information Processing Systems*, 31, 2018.
- [VVS10] Erik Vee, Sergei Vassilvitskii, and Jayavel Shanmugasundaram. Optimal online assignment with forecasts. In *Proceedings of the Conference on Electronic Commerce*, pages 109–118, 2010.

A Degree Increment Lemma

We prove Lemma 5.5 here.

Proof. Given \mathcal{L} , we construct the label cover instance $\mathcal{L}'(U', V', E', \Sigma_U, \Sigma_V, \{\pi'_e\}_{e \in E'})$ as follows.

Vertex Set. For every $u \in U$, we introduce ℓ copies of the vertex u , which we call $(u, 1), \dots, (u, \ell)$. Similarly, for every vertex $v \in V$, we introduce ℓ copies of the vertex v , which we call $(v, 1), \dots, (v, \ell)$. Let $U' = \{(u, i) : u \in U, i \in [\ell]\}$, and similarly, let $V' = \{(v, i) : v \in V, i \in [\ell]\}$.

Label Set. The left and right label sets are still Σ_U and Σ_V .

Constraint Set. For every constraint $e = (u, v) \in E$, and for every pair of indices $(i, j) \in [\ell] \times [\ell]$, we introduce an edge $e_{ij} = \{(u, i), (v, j)\}$ in E' , and define the corresponding projection function $\pi'_{e_{ij}} = \pi_e$.

Note that by construction $|U'| = \ell|U|$, $|V'| = \ell|V|$, and $|E'| = \ell^2|E|$. This concludes the description of the reduction. All that remains is to claim that $\text{Val}(\mathcal{L}) = \text{Val}(\mathcal{L}')$. To that end, let us first show that $\text{Val}(\mathcal{L}') \geq \text{Val}(\mathcal{L})$. Let σ be an optimal labeling of \mathcal{L} . Given σ , we construct a labeling σ' of \mathcal{L}' as follows. For every $(u, i) \in U'$, we let $\sigma'(u, i) = \sigma(u)$, and similarly, for every $(v, j) \in V'$, we let $\sigma'(v, j) = \sigma(v)$. Then the fraction of constraints in \mathcal{L}' satisfied by σ' is

$$\begin{aligned}
\Pr_{(u', v') \sim E'} \left[\pi'_{(u', v')}(\sigma'(v')) = \sigma'(u') \right] &= \mathbb{E}_{i, j \sim [\ell] \times [\ell]} \Pr_{(u, v) \sim E} \left[\pi'_{(u, i), (v, j)}(\sigma'(u, i)) = \sigma'(v, j) \right] \\
&= \mathbb{E}_{i, j \sim [\ell] \times [\ell]} \Pr_{(u, v) \sim E} \left[\pi'_{(u, v)}(\sigma(u)) = \sigma(v) \right] \\
&= \text{Val}(\mathcal{L}),
\end{aligned}$$

which shows that there exists a labeling of \mathcal{L}' which can satisfy at least $\text{Val}(\mathcal{L})$ fraction of constraints in \mathcal{L}' . This establishes that $\text{Val}(\mathcal{L}') \geq \text{Val}(\mathcal{L})$.

Conversely, let σ' be an optimal labeling of $\text{Val}(\mathcal{L})$. Consider the following randomized labeling procedure for \mathcal{L} .

- Sample $i \sim [\ell]$ and $j \sim [\ell]$ uniformly at random.
- Assign $\sigma(u) = \sigma'(u, i)$ and $\sigma(v) = \sigma'(v, j)$ for every $u \in U$ and $v \in V$ respectively.

Then, randomizing over the choice of σ , the expected fraction of constraints that are satisfied by the labeling σ is

$$\begin{aligned} \mathbb{E}_{\sigma} \Pr_{(u,v) \sim E} [\pi_{(u,v)}(\sigma(v)) = \sigma(u)] &= \mathbb{E}_{i,j \sim [\ell] \times [\ell]} \Pr_{(u,v) \sim E} [\pi_{(u,v)}(\sigma'(u, i)) = \sigma'(v, j)] \\ &= \mathbb{E}_{i,j \sim [\ell] \times [\ell]} \Pr_{(u,v) \sim E} [\pi'_{(u,i),(v,j)}(\sigma'(u, i)) = \sigma'(v, j)] \\ &= \text{Val}(\mathcal{L}'), \end{aligned}$$

which implies that there exists a labeling of σ of \mathcal{L} which satisfies at least $\text{Val}(\mathcal{L}')$ fraction of constraints, thus implying that $\text{Val}(\mathcal{L}) \geq \text{Val}(\mathcal{L}')$. \square