New NP-hardness results for 3-Coloring and 2-to-1 Label Cover[*]

Per Austrin[†]   Ryan O'Donnell[‡]   Li-Yang Tan[§]   John Wright[¶]

October 16, 2018

Abstract

We show that given a 3-colorable graph, it is NP-hard to find a 3-coloring with $(\frac{16}{17} + \epsilon)$ of the edges bichromatic. In a related result, we show that given a satisfiable instance of the 2-to-1 Label Cover problem, it is NP-hard to find a $(\frac{23}{24} + \epsilon)$-satisfying assignment.

1 Introduction

Graph coloring problems differ from many other Constraint Satisfaction Problems (CSPs) in that we typically care about the case of *perfect completeness*, e.g. when the graph under consideration is 3-colorable rather than almost 3-colorable. Unfortunately, this means that many of the powerful tools which have been developed for proving inapproximability results are no longer applicable. Most prominently, Raghavendra's [Rag08] optimal inapproximability results for all CSPs, which are conditioned on the unproven *Unique Games Conjecture* (UGC), only apply to the case of imperfect completeness. The UGC states that it is NP-hard to distinguish between nearly satisfiable and almost completely unsatisfiable instances of *Unique, or 1-to-1*, Label Cover. As a result, by starting a reduction with the UGC, one has already lost perfect completeness. Thus, any inapproximability result for a graph coloring problem must begin with a different unproven assumption, such as P ≠ NP or Khot's [Kho02] 2-to-1 Conjecture.

The motivation of this paper is to study both of these assumptions as they relate to the graph k-Coloring problem, specifically in the $k = 3$ case. In the k-Coloring problem, the input is a k-colorable graph G, and the task is to find a k-coloring of the vertices of G which maximizes the number of bichromatic edges. This problem has previously gone under the names "Max-k-Colorability" [Pet94] and "Maximum k-Colorable Subgraph" [GS09]. Graph k-Coloring, along with its many studied variants, is a central problem in Computer Science, and pinning down its exact approximability is an important open problem. The main result of our paper is an improved inapproximability result for 3-Coloring, predicated only on P ≠ NP:

Theorem 1.1. *For all $\epsilon > 0$, $(1, \frac{16}{17} + \epsilon)$-deciding the 3-Coloring problem is NP-hard.*

Here by (c, s)-deciding a CSP we mean the task of determining whether an instance is at least c-satisfiable or less than s-satisfiable. In fact, this is the best known hardness result for the 3-Coloring problem, even assuming conjectures such as the 2-to-1 Conjecture. The previous best NP-hardness for 3-Coloring was due to Guruswami and Sinop [GS09], who showed a factor $\frac{32}{33}$-hardness via a somewhat involved gadget reduction from the 3-query adaptive PCP result of [GLST98]. In contrast, the best current algorithm achieves an approximation ratio of 0.836 (and does not need the instance to be satisfiable) [GW04]. As for larger values of k, [GS09] construct a reduction which directly translates hardness results for 3-Coloring into hardness results for k-Coloring, for $k \geq 3$. Applying this to our Theorem 1.1 yields

Theorem 1.2. *For all $k \geq 3$ and $\epsilon > 0$, it is NP-hard to $(1, 1 - \frac{1}{17(k+c_k)+c_k} + \epsilon)$-decide the k-Coloring problem. Here $c_k = k \pmod 3$.*

This is the best known NP-hardness for k-Coloring. For sufficiently large k, stronger inapproximability results are known to follow from the 2-to-1 Conjecture:

2-to-1 Conjecture ([Kho02]). *For every integer $\epsilon > 0$, there is a label set size q such that it is NP-hard to $(1, \epsilon)$-decide the 2-to-1 Label Cover problem.*

Guruswami and Sinop [GS09] have shown that the 2-to-1 Conjecture implies it is NP-hard to $(1, 1 - \frac{1}{k} + O(\frac{\ln k}{k^2}))$-decide the k-Coloring problem. This result would be tight up to the $O(\cdot)$ by an algorithm of Frieze and Jerrum [FJ97]. In a prior result, Dinur, Mossel, and Regev [DMR09] showed that the 2-to-1 Conjecture implies that it is NP-hard to C-color a 4-colorable graph for any constant C. (They also showed hardness for 3-colorable graphs via another Unique Games variant.) It is therefore clear that settling the 2-to-1 Conjecture is important to the study of the inapproximability of graph coloring problems.

Interestingly, to a certain extent the reverse is also true: it is "folklore" that hardness results for graph 3-Coloring immediately imply hardness results for the 2-to-1 Label Cover problem with label sizes 3 & 6 by a simple "constraint-variable" reduction. Indeed, Theorem 1.1 by itself would give the best-known NP-hardness for 2-to-1 Label Cover. However, we are able to get an even better hardness result than this by studying a CSP closely related to 3-Coloring. Our hardness result is:

Theorem 1.3. *For all $\epsilon > 0$, $(1, \frac{23}{24} + \epsilon)$-deciding the 2-to-1 Label Cover problem with label set sizes 3 & 6 is* NP*-hard.*

By duplicating labels, this result also holds for label set sizes $3k$ & $6k$ for any $k \in \mathbb{N}^+$. To the best of our knowledge, no explicit NP-hardness for this problem has previously been stated in the literature. Combining the constraint-variable reduction with the above-mentioned 3-Coloring hardness of [GS09] gives an NP-hardness of $(1, \frac{65}{66} + \epsilon)$ for the problem with label sizes 3 & 6, which we believe to be the best previously known. It is not known how to take advantage of larger label set sizes. On the other hand, for label set sizes 2 & 4 it is known that satisfying 2-to-1 Label Cover instances can be found in polynomial time.

Regarding the hardness of the 2-to-1 Label Cover problem, the only evidence we have is a family of integrality gaps for the canonical SDP relaxation of the problem, in [GKO+10]. Regarding algorithms for the problem, an important recent line of work beginning in [ABS10] (see also [BRS11, GS11, Ste10]) has sought subexponential-time algorithms for Unique Label Cover and related problems. In particular, Steurer [Ste10] has shown that for any constant $\beta > 0$ and label set size, there is an $\exp(O(n^\beta))$-time algorithm which, given a satisfiable 2-to-1 Label Cover instance, finds an assignment satisfying an $\exp(-O(1/\beta^2))$-fraction of the constraints. E.g., there is a $2^{O(n^{.001})}$-time algorithm which $(1, s_0)$-approximates 2-to-1 Label Cover, where $s_0 > 0$ is a certain universal constant.

In light of this, it is interesting not only to seek NP-hardness results for certain approximation thresholds, but to additionally seek evidence that *nearly full exponential time* is required for these thresholds. This can done by assuming the Exponential Time Hypothesis (ETH) [IP01] and by reducing from the Moshkovitz–Raz Theorem [MR10], which shows a near-linear size reduction from 3Sat to the standard Label Cover problem with subconstant soundness. In this work, we show reductions from 3Sat to the problem of $(1, s + \epsilon)$-approximating several CSPs, for certain values of s and for all $\epsilon > 0$. In fact, though we omit it in our theorem statements, it can be checked that all of the reductions in this paper are quasilinear in size for $\epsilon = \epsilon(n) = \Theta\left(\frac{1}{(\log \log n)^\beta}\right)$, for some $\beta > 0$.

1.1 Our techniques

Let us describe the high-level idea behind our result. The folklore constraint-variable reduction from 3-Coloring to 2-to-1 Label Cover would work just as well if we started from "3-Coloring with literals" instead. By this we mean the CSP with domain $\mathbb{Z}_3$ and constraints of the form "$v_i - v_j \neq c$ (mod 3)". Starting from this CSP — which we call 2NLin($\mathbb{Z}_3$) — has two benefits: first, it is at least as hard as 3-Coloring and hence could yield a stronger hardness result; second, it is a bit more "symmetrical" for the purposes of designing reductions. Finally, having proven a hardness result for 2NLin, it seems reasonable that it can be modified into a hardness result for 3-Coloring. We obtain the following hardness result for 2NLin($\mathbb{Z}_3$).

Theorem 1.4. *For all $\epsilon > 0$, it is* NP*-hard to $(1, \frac{11}{12} + \epsilon)$-decide the 2NLin problem.*

As 3-Coloring is a special case of 2NLin($\mathbb{Z}_3$), [GS09] also shows that $(1, \frac{32}{33} + \epsilon)$-deciding 2NLin is NP-hard for all $\epsilon > 0$, and to our knowledge this was previously the only hardness known for

$\mathsf{2NLin}(\mathbb{Z}_3)$. Further, the 0.836-approximation algorithm for $\mathsf{3\text{-}Coloring}$ from above achieves the same approximation ratio for $\mathsf{2NLin}(\mathbb{Z}_3)$, and this is the best known [GW04]. To prove Theorem 1.4, we proceed by designing an appropriate "function-in-the-middle" dictator test, as in the recent framework of [OW12]. Although the [OW12] framework gives a direct translation of certain types of function-in-the-middle tests into hardness results, we cannot employ it in a black-box fashion. Among other reasons, [OW12] assumes that the test has "built-in noise", but we cannot afford this as we need our test to have perfect completeness.

Thus, we need a different proof to derive a hardness result from this function-in-the-middle test. We first were able to accomplish this by an analysis similar to the Fourier-based proof of $\mathsf{2Lin}(\mathbb{Z}_2)$ hardness given in Appendix F of [OW12]. Just as that proof "reveals" that the function-in-the-middle $\mathsf{2Lin}(\mathbb{Z}_2)$ test can be equivalently thought of as Håstad's $\mathsf{3Lin}(\mathbb{Z}_2)$ test composed with the $\mathsf{3Lin}(\mathbb{Z}_2)$-to-$\mathsf{2Lin}(\mathbb{Z}_2)$ gadget of [TSSW00], our proof for the $\mathsf{2NLin}(\mathbb{Z}_3)$ function-in-the-middle test revealed it to be the composition of a function test for a certain four-variable CSP with a gadget. We have called the particular four-variable CSP $\mathsf{4\text{-}Not\text{-}All\text{-}There}$, or $\mathsf{4NAT}$ for short. Because it is a 4-CSP, we are able to prove the following NP-hardness of approximation result for it using a classic, Håstad-style Fourier-analytic proof.

Theorem 1.5. *For all $\epsilon > 0$, it is NP-hard to $(1, \frac{2}{3} + \epsilon)$-decide the $\mathsf{4NAT}$ problem.*

Thus, the final form in which we present our Theorem 1.3 is as a reduction from Label-Cover to $\mathsf{4NAT}$ using a function test (yielding Theorem 1.5), followed by a $\mathsf{4NAT}$-to-$\mathsf{2NLin}(\mathbb{Z}_3)$ gadget (yielding Theorem 1.4), followed by the constraint-variable reduction to 2-to-1 Label Cover. Indeed, all of the technology needed to carry out this proof was in place for over a decade, but without the function-in-the-middle framework of [OW12] it seems that pinpointing the $\mathsf{4NAT}$ predicate as a good starting point would have been unlikely.

Our proof of Theorem 1.1 is similar: we design a function-in-the-middle test for $\mathsf{3\text{-}Coloring}$ which uses the $\mathsf{2NLin}(\mathbb{Z}_3)$ function test as a subroutine. And though we do not find a gadget reduction from $\mathsf{3\text{-}Coloring}$ to $\mathsf{4NAT}$, we are able to express the success probability of the test in terms of the $\mathsf{4NAT}$ test. Thus, there is significant overlap in the proofs of our two main theorems, and we are able to carry out the proofs simultaneously.

1.2 Organization

We leave to Section 2 most of the definitions, including those of the CSPs we use. The heart of the paper is in Section 3, where we give the $\mathsf{2NLin}(\mathbb{Z}_3)$, $\mathsf{3\text{-}Coloring}$, and $\mathsf{4NAT}$ function tests and explain how they are related. Section 4 contains the Fourier analysis of the tests. The actual hardness proof for $\mathsf{4NAT}$ is presented in Section 5, and it follows mostly the techniques put in place by Håstad in [Hås01]. Because the hardness proof for $\mathsf{3\text{-}Coloring}$ is almost identical, we omit it. Appendix A contains a technical lemma.

2 Preliminaries

We primarily work with strings $x \in \mathbb{Z}_3^K$ for some integer K. We write x_i to denote the ith coordinate of x.

2.1 Definitions of problems

An instance $\mathcal{I}$ of a *constraint satisfaction problem* (CSP) is a set of variables V, a set of labels D, and a weighted list of constraints on these variables. We assume that the weights of the constraints are

nonegative and sum to 1. The weights therefore induce a probability distribution on the constraints. Given an assignment to the variables $f : V \to D$, the *value* of f is the probability that f satisfies a constraint drawn from this probability distribution. The *optimum* of $\mathcal{I}$ is the highest value of any assignment. We say that an $\mathcal{I}$ is *s-satisfiable* if its optimum is at least s. If it is 1-satisfiable we simply call it satisfiable.

We define a CSP $\mathcal{P}$ to be a set of CSP instances. Typically, these instances will have similar constraints. We will study the problem of (c, s)-*deciding* $\mathcal{P}$. This is the problem of determining whether an instance of $\mathcal{P}$ is at least c-satisfiable or less than s-satisfiable. Related is the problem of (c, s)-*approximating* $\mathcal{P}$, in which one is given a c-satisfiable instance of $\mathcal{P}$ and asked to find an assignment of value at least s. It is easy to see that (c, s)-deciding $\mathcal{P}$ is at least as easy as (c, s)-approximating $\mathcal{P}$. Thus, as all our hardness results are for (c, s)-deciding CSPs, we also prove hardness for (c, s)-approximating these CSPs.

We now state the four CSPs that are the focus of our paper.

3-coloring: In this CSP the label set is $\mathbb{Z}_3$ and the constraints are of the form $v_i \neq v_j$.

2-NLin($\mathbb{Z}_3$): In this CSP the label set is $\mathbb{Z}_3$ and the constraints are of the form

$$v_i - v_j \neq a \pmod 3, \quad a \in \mathbb{Z}_3.$$

The special case when each RHS is 0 is the 3-Coloring problem. We often drop the ($\mathbb{Z}_3$) from this notation and simply write 2NLin. The reader may think of the 'N' in 2NLin($\mathbb{Z}_3$) as standing for 'N'on-linear, although we prefer to think of it as standing for 'N'early-linear. The reason is that when generalizing to moduli $q > 3$, the techniques in this paper generalize to constraints of the form "$v_i - v_j \pmod q \in \{a, a + 1\}$" rather than "$v_i - v_j \neq a \pmod q$". For the ternary version of this constraint, "$v_i - v_j + v_k \pmod q \in \{a, a + 1\}$", it is folklore[1] that a simple modification of Håstad's work [Hås01] yields NP-hardness of $(1, \frac{2}{q})$-approximation.

4-Not-All-There: For the 4-Not-All-There problem, denoted 4NAT, we define $\mathsf{4NAT} : \mathbb{Z}_3^4 \to \{0, 1\}$ to have output 1 if and only if at least one of the elements of $\mathbb{Z}_3$ is not present among the four inputs. The 4NAT CSP has label set $D = \mathbb{Z}_3$ and constraints of the form $\mathsf{4NAT}(v_1 + k_1, v_2 + k_2, v_3 + k_3, v_4 + k_4) = 1$, where the k_i's are constants in $\mathbb{Z}_3$.

We additionally define the "Two Pairs" predicate $\mathsf{TwoPair} : \mathbb{Z}_3^4 \to \{0, 1\}$, which has output 1 if and only if its input contains two distinct elements of $\mathbb{Z}_3$, each appearing twice. Note that an input which satisfies TwoPair also satisfies 4NAT.

d-to-1 Label Cover: An instance of the d-to-1 Label Cover problem is a bipartite graph $G = (U \cup V, E)$, a label set size K, and a d-to-1 map $\pi_e : [dK] \to [K]$ for each edge $e \in E$. The elements of U are labeled from the set $[K]$, and the elements of V are labeled from the set $[dK]$. A labeling $f : U \cup V \to [dK]$ satisfies an edge $e = (u, v)$ if $\pi_e(f(v)) = f(u)$. Of particular interest is the $d = 2$ case, i.e., 2-to-1 Label Cover.

Label Cover serves as the starting point for most NP-hardness of approximation results. We use the following theorem of Moshkovitz and Raz:

Theorem 2.1 ([MR10])**.** *For any $\epsilon = \epsilon(n) \geq n^{-o(1)}$ there exists $K, d \leq 2^{\mathrm{poly}(1/\epsilon)}$ such that the problem of deciding a **3Sat** instance of size n can be Karp-reduced in $\mathrm{poly}(n)$ time to the problem of $(1, \epsilon)$-deciding d-to-1 Label Cover instance of size $n^{1+o(1)}$ with label set size K.*

---

[1]Venkatesan Guruswami, Subhash Khot personal communications.

2.2 Gadgets

A typical way of relating two separate CSPs is by constructing a *gadget reduction* which translates from one to the other. A gadget reduction from CSP_1 to CSP_2 is one which maps any CSP_1 constraint into a weighted set of CSP_2 constraints. The CSP_2 constraints are over the same set of variables as the CSP_1 constraint, plus some new, auxiliary variables (these auxiliary variables are not shared between constraints of CSP_1). We require that for every assignment which satisfies the CSP_1 constraint, there is a way to label the auxiliary variables to fully satisfy the CSP_2 constraints. Furthermore, there is some parameter $0 < \gamma < 1$ such that for every assignment which does not satisfy the CSP_1 constraint, the optimum labeling to the auxiliary variables will satisfy exactly γ fraction of the CSP_2 constraints. Such a gadget reduction we call a γ-*gadget-reduction* from CSP_1 to CSP_2. The following proposition is well-known:

Proposition 2.2. *Suppose it is* NP*-hard to* (c, s)*-decide* CSP_1*. If there exists a* γ*-gadget-reduction from* CSP_1 *to* CSP_2*, then it is* NP*-hard to* $(c + (1 - c)\gamma, s + (1 - s)\gamma)$*-decide* CSP_2*.*

We note that the notation γ-gadget-reduction is similar to a piece of notation employed by [TSSW00], but the two have different (though related) definitions.

2.3 Fourier analysis on $\mathbb{Z}_3$

Let $\omega = e^{2\pi i/3}$ and set $U_3 = \{\omega^0, \omega^1, \omega^2\}$. For $\alpha \in \mathbb{Z}_3^n$, consider the Fourier character $\chi_\alpha : \mathbb{Z}_3^n \to U_3$ defined as $\chi_\alpha(x) = \omega^{\alpha \cdot x}$. Then it is easy to see that $\mathbf{E}[\chi_\alpha(\boldsymbol{x})\overline{\chi_\beta(\boldsymbol{x})}] = \mathbf{1}[\alpha = \beta]$, where here and throughout $\boldsymbol{x}$ has the uniform probability distribution on $\mathbb{Z}_3^n$ unless otherwise specified.. As a result, the Fourier characters form an orthonormal basis for the set of functions $f : \mathbb{Z}_3^n \to U_3$ under the inner product $\langle f, g \rangle = \mathbf{E}[f(\boldsymbol{x})g(\boldsymbol{x})]$; i.e.,

$$f = \sum_{\alpha \in \mathbb{Z}_3^n} \hat{f}(\alpha)\chi_\alpha,$$

where the $\hat{f}(\alpha)$'s are complex numbers defined as $\hat{f}(\alpha) = \mathbf{E}[f(\boldsymbol{x})\overline{\chi_\alpha(\boldsymbol{x})}]$. For $\alpha \in \mathbb{Z}_3^n$, we use the notation $|\alpha|$ to denote $\sum \alpha_i$ and $\#\alpha$ to denote the number of nonzero coordinates in α. When d is clear from context and $\alpha \in \mathbb{Z}_3^{dK}$, define $\pi_3(\alpha) \in \mathbb{Z}_3^K$ so that $(\pi_3(\alpha))_i \equiv |\alpha[i]| \pmod 3$ (recall the notation $\alpha[i]$ from the beginning of this section). We have Parseval's identity: for every $f : \mathbb{Z}_3^n \to U_3$ it holds that $\sum_{\alpha \in \mathbb{Z}_3^n} |\hat{f}(\alpha)|^2 = 1$. Note that this implies that $|\hat{f}(\alpha)| \leq 1$ for all α, as otherwise $|\hat{f}(\alpha)|^2$ would be greater than 1.

A function $f : \mathbb{Z}_3^n \to \mathbb{Z}_3$ is said to be *folded* if for every $x \in \mathbb{Z}_3^n$ and $c \in \mathbb{Z}_3$, it holds that $f(x + c) = f(x) + c$, where $(x + c)_i = x_i + c$.

Proposition 2.3. *Let* $f : \mathbb{Z}_3^n \to U_3$ *be folded. Then* $\hat{f}(\alpha) \neq 0 \Rightarrow |\alpha| \equiv 1 \pmod 3$.

Proof.

$$\hat{f}(\alpha) = \mathbf{E}[f(\boldsymbol{x} + 1)\overline{\chi_\alpha(\boldsymbol{x} + 1)}] = \mathbf{E}[\omega f(\boldsymbol{x})\overline{\chi_\alpha(\boldsymbol{x})\chi_\alpha(1, 1, \dots, 1)}] = \omega\overline{\chi_\alpha(1, 1, \dots, 1)}\hat{f}(\alpha).$$

This means that $\omega\overline{\chi_\alpha(1, 1, \dots, 1)}$ must be 1. Expanding this quantity,

$$\omega\overline{\chi_\alpha(1, 1, \dots, 1)} = \omega^{1 - \alpha \cdot (1, 1, \dots, 1)} = \omega^{1 - |\alpha|}.$$

So, $|\alpha| \equiv 1 \pmod 3$, as promised. $\square$

2.4 Dictatorship tests

In this paper, we make use dictatorship tests, which are a standard tool for proving NP-hardness of approximation results. Generally speaking, the input of a dictatorship test is two functions $f : \mathbb{Z}_3^K \to \mathbb{Z}_3$ and $g : \mathbb{Z}_3^{dK} \to \mathbb{Z}_3$ and a d-to-1 map $\pi : [dK] \to [K]$. The map π naturally groups strings $y \in \mathbb{Z}_3^{dK}$ into K separate "blocks" of coordinates, the first block being the coordinates in $\pi^{-1}(1)$, the second block being the coordinates in $\pi^{-1}(2)$, etc. Without loss of generality we will assume that π is the map where $\pi(k) = 1$ for $1 \le k \le d$, $\pi(k) = 2$ for $d+1 \le k \le 2d$, and so on. In this case, we write $y[i] \in \mathbb{Z}_3^d$ for the ith block of y, and $(y[i])_j \in \mathbb{Z}_3$ for the jth coordinate of this block.

The goal of a dictatorship test is to distinguish the case when f and g are "matching dictators" from the case when f and g are "far from matching dictators". A function f is a *dictator* if $f(x) = x_i$, for some i. Furthermore, f and g are *matching dictators* if $f(x) = x_i$, $g(y) = y_j$, and $\pi(j) = i$. In other words, they are dictators whose dictator coordinates match up according to the map π. A property of matching dictators is that both f and g "depend on" certain coordinates, meaning that these coordinates are important to the output of f and g, and these coordinates match each other. Thus, f and g are *far from matching dictators* if there are no coordinates i and j which f and g depend on, respectively, for which $\pi(j) = i$. An example of this is "nonmatching" dictators, when, say, $f(x) = x_1$ and $g(y) = y_{d+1}$.

To prove hardness for 3-Coloring (i.e., the $\neq$ constraint), one should construct a dictatorship test with the following outline: first, the test picks $\boldsymbol{x} \in \mathbb{Z}_3^K$ and $\boldsymbol{y} \in \mathbb{Z}_3^{dK}$ from some random distribution, and checks whether $f(\boldsymbol{x}) \neq g(\boldsymbol{y})$. If indeed this is the case, then the test passes, and otherwise it fails. Generally, if one is interested in showing that (c, s)-deciding a given problem is NP-hard, it suffices to construct a test for which matching dictators pass with probability at least c and functions far from matching dictators pass with probability less than s.

We use a variant of this outline proposed in [OW12], in which the test involves a third auxiliary function $h : S \to \mathbb{Z}_3$, where S is some set. We still want to distinguish the cases of f and g being matching dictators and functions far from matching dictators, but now the outline is a little different: in addition to selecting $\boldsymbol{x}$ and $\boldsymbol{y}$, we also select from some distribution a string $\boldsymbol{z} \in S$. Then with some probability we test $h(\boldsymbol{z}) \neq f(\boldsymbol{x})$ and with some probability $h(\boldsymbol{z}) \neq g(\boldsymbol{y})$. A test with this outline we refer to as a "function-in-the-middle" test, as h acts as an intermediary between the functions f and g.

3 3-Coloring and 2NLin tests

In this section, we give our hardness results for 3-Coloring and 2-to-1 Label Cover, following the proof outlines described at the end of Section 1.1. First, we state a pair of simple gadget reductions:

Lemma 3.1. *There is a $3/4$-gadget-reduction from 4NAT to 2NLin.*

Lemma 3.2. *There is a $1/2$-gadget-reduction from 2NLin to 2-to-1 Label Cover.*

Together with Proposition 2.2, these imply the following corollary:

Corollary 3.3. *There is a $7/8$-gadget-reduction from 4NAT to 2-to-1 Label Cover. Thus, if it is NP-hard to (c, s)-decide the 4NAT problem, then it is NP-hard to $((7 + c)/8, (7 + s)/8)$-decide the 2-to-1 Label Cover problem.*

The gadget reduction from 4NAT to 2NLin relies on the simple fact that if $a, b, c, d \in \mathbb{Z}_3$ satisfy the 4NAT predicate, then there is some element of $\mathbb{Z}_3$ that none of them equal.

6

Proof of Lemma 3.1. A 4NAT constraint C on the variables $S = (v_1, v_2, v_3, v_4)$ is of the form

$$4\mathsf{NAT}(v_1 + k_1, v_2 + k_2, v_3 + k_3, v_4 + k_4),$$

where the k_i's are all constants in $\mathbb{Z}_3$. To create the 2NLin instance, introduce the auxiliary variable y_C and add the four 2NLin equations

$$v_i + k_i \neq y_C \pmod 3, \quad i \in [4]. \tag{1}$$

If $f : S \to \mathbb{Z}_3$ is an assignment which satisfies the 4NAT constraint, then there is some $a \in \mathbb{Z}_3$ such that $f(v_i) + k_i \neq a \pmod 3$ for all $i \in [4]$. Assigning a to y_C satisfies all four equations (1). On the other hand, if f doesn't satisfy the 4NAT constraint, then $\{f(v_i) + k_i\}_{i \in [4]} = \mathbb{Z}_3$, so no assignment to y_C satisfies all four equations. However, it is easy to see that there is an assignment which satisfies three of the equations. This gives a $\frac{3}{4}$-gadget-reduction from 4NAT to 2NLin, which proves the lemma. $\square$

The reduction from 2NLin to 2-to-1 Label Cover is the well-known constraint-variable reduction, and uses the fact that in the equation $v_i - v_j \neq a \pmod 3$, for any assignment to v_j there are two valid assignments to v_i, and vice versa.

Proof of Lemma 3.2. A 2NLin constraint C on the variables $S = (v_1, v_2)$ is of the form

$$v_1 - v_2 \neq a \pmod 3,$$

for some $a \in \mathbb{Z}_3$. To create the 2-to-1 Label Cover instance, introduce the variable y_C which will be labeled by one of the six possible functions $g : S \to \mathbb{Z}_3$ which satisfies C. Finally, introduce the 2-to-1 constraints $y_C(v_1) = v_1$ and $y_C(v_2) = v_2$. Here v_1 and v_2 are treated on the left as inputs to the function labeling y_C and on the right as variables to be labeled with values in $\mathbb{Z}_3$.

If $f : S \to \mathbb{Z}_3$ is an assignment which satisfies the 2NLin constraint, then we label y_C with f. In this case,

$$y_C(v_i) = f(v_i), \quad i = 1, 2.$$

Thus, both equations are satisfied. On the other hand, if f does not satisfy the 2NLin constraint, then any g which y_C is labeled with disagrees with f on at least one of v_1 or v_2. It is easy to see, though, that a g can be selected to satisfy one of the two equations. This gives a $\frac{1}{2}$-gadget-reduction from 2NLin to 2-to-1, which proves the lemma. $\square$

3.1 Three tests

Now that we have shown that 2NLin hardness results translate into 2-to-1 Label Cover hardness results, we present our 2NLin function test. From here, the 3-Coloring function test follows immediately. Finally, we will show how in the course of analyzing the 2NLin test one is lead naturally to our 4NAT test. This correspondence between the 2NLin test and the 4NAT test parallels the gadget reduction from Lemma 3.1. The test is:

$$\boxed{\text{2NLin Test}}$$

Given folded functions $f : \mathbb{Z}_3^K \to \mathbb{Z}_3$, $g, h : \mathbb{Z}_3^{dK} \to \mathbb{Z}_3$:

- Let $\boldsymbol{x} \in \mathbb{Z}_3^K$ and $\boldsymbol{y} \in \mathbb{Z}_3^{dK}$ be independent and uniformly random.

- For each $i \in [K], j \in [d]$, select $(\boldsymbol{z}[i])_j$ independently and uniformly from the elements of $\mathbb{Z}_3 \setminus \{\boldsymbol{x}_i, (\boldsymbol{y}[i])_j\}$.

- With probability $\frac{1}{4}$, test $f(\boldsymbol{x}) \neq h(\boldsymbol{z})$; with probability $\frac{3}{4}$, test $g(\boldsymbol{y}) \neq h(\boldsymbol{z})$.

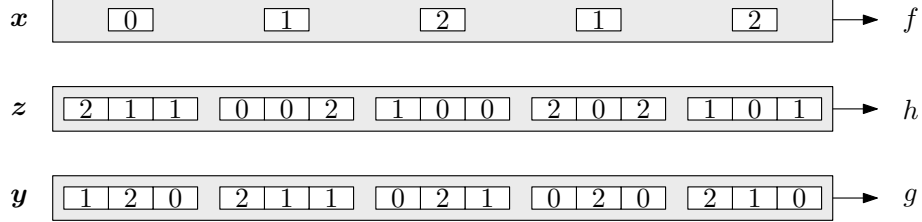

Figure 1: An illustration of the 2NLin test distribution; $d = 3$, $K = 5$

Above is an illustration of the test. We remark that for any given block i, $z[i]$ determines x_i (with very high probability), because as soon as $z[i]$ contains two distinct elements of $\mathbb{Z}_3$, x_i must be the third element of $\mathbb{Z}_3$. Notice also that in every column of indices, the input to h always differs from the inputs to both f and g. Thus, "matching dictator" assignments pass the test with probability 1. (This is the case in which $f(x) = x_i$ and $g(y) = (y[i])_j$ for some $i \in [K]$, $j \in [d]$.) On the other hand, if f and g are "nonmatching dictators", then they succeed with only $\frac{11}{12}$ probability. This turns out to be essentially optimal among functions f and g without "matching influential coordinates/blocks". We will obtain the following theorem:

Theorem 1.4 restated. *For all $\epsilon > 0$, it is NP-hard to $(1, \frac{11}{12} + \epsilon)$-decide the 2NLin problem.*

We would like to use a similar test for our 3-Coloring hardness result, but we can no longer assume that the functions f, g, and h are folded. This is problematic, as without this guarantee f and g could both be identically 0 and h could be identically 1, in which case the three functions would pass the test with probability 1. Since constant functions cannot be decoded to Label Cover solutions, we would like to prevent this from happening. Thus, we will add "folding tests" to force f and g to look folded. Having ensured this, we are free to run the 2NLin test without worry. The test is:

3-Coloring Test

Given functions $f : \mathbb{Z}_3^K \to \mathbb{Z}_3$, $g, h : \mathbb{Z}_3^{dK} \to \mathbb{Z}_3$:

- Let $\boldsymbol{x} \in \mathbb{Z}_3^K$ and $\boldsymbol{y} \in \mathbb{Z}_3^{dK}$ be independent and uniformly random.

- With probability $\frac{1}{17}$, test $f(\boldsymbol{x}) \neq f(\boldsymbol{x} + 1)$; with probability $\frac{4}{17}$, test $g(\boldsymbol{y}) \neq g(\boldsymbol{y} + 1)$.

- With the remaining $\frac{12}{17}$ probability, run the "non-folded" version of the 2NLin test on f, g, and h.

Here, by the "non-folded" version of the 2NLin test, we mean the test which is identical to the 2NLin test, only it does not assume f, g, and h are folded. If f and g are matching dictators, then they always pass the folding tests, so as before they succeed with probability 1. If on the other hand f and g are nonmatching dictators, then they also always pass the folding tests, so they succeed with probability $\frac{5}{17} + \frac{12}{17} \cdot \frac{11}{12} = \frac{16}{17}$. Just as before, this turns out to be basically optimal among functions without matching influential coordinates:

Theorem 1.1 restated. *For all $\epsilon > 0$, it is NP-hard to $(1, \frac{16}{17} + \epsilon)$-decide the 3-Coloring problem.*

Let us further discuss the 2NLin test. Given $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ from the 2NLin test, consider the following method of generating two additional strings $\boldsymbol{y}', \boldsymbol{y}'' \in \mathbb{Z}_3^{dK}$ which represent h's "uncertainty" about $\boldsymbol{y}$. For $j \in [d]$, if $\boldsymbol{x}_i = (\boldsymbol{y}[i])_j$, then set both $(\boldsymbol{y}'[i])_j$ and $(\boldsymbol{y}''[i])_j$ to the lone element of $\mathbb{Z}_3 \setminus \{\boldsymbol{x}_i, (\boldsymbol{z}[i])_j\}$. Otherwise, set one of $(\boldsymbol{y}'[i])_j$ or $(\boldsymbol{y}''[i])_j$ to $\boldsymbol{x}_i$, and the other one to $(\boldsymbol{y}[i])_j$. It can be checked that $\mathsf{TwoPair}(\boldsymbol{x}_i, (\boldsymbol{y}[i])_j, (\boldsymbol{y}'[i])_j, (\boldsymbol{y}''[i])_j) = 1$, a more stringent requirement than satisfying 4NAT. In fact, the marginal distribution on these four variables is a uniformly random assignment that satisfies the TwoPair predicate. Conditioned on $\boldsymbol{x}$ and $\boldsymbol{z}$, the distribution on $\boldsymbol{y}'$ and $\boldsymbol{y}''$ is identical to the distribution on $\boldsymbol{y}$. To see this, first note that by construction, neither $(\boldsymbol{y}'[i])_j$ nor $(\boldsymbol{y}''[i])_j$ ever equals $(\boldsymbol{z}[i])_j$. Further, because these indices are distributed as uniformly random satisfying assignments to TwoPair, $\mathbf{Pr}[(\boldsymbol{y}'[i])_j = x_i] = \mathbf{Pr}[(\boldsymbol{y}''[i])_j = x_i] = \frac{1}{3}$, which matches the corresponding probability for $\boldsymbol{y}$. Thus, as $\boldsymbol{y}$, $\boldsymbol{y}'$, and $\boldsymbol{y}''$ are distributed identically, we may rewrite the test's success probability as:

$$\mathbf{Pr}[f, g, \text{ and } h \text{ pass the 2NLin test}] = \tfrac{1}{4} \mathbf{Pr}[f(\boldsymbol{x}) \neq h(\boldsymbol{z})] + \tfrac{3}{4} \mathbf{Pr}[g(\boldsymbol{y}) \neq h(\boldsymbol{z})]$$

$$= \operatorname{avg} \left\{ \begin{array}{c} \mathbf{Pr}[f(\boldsymbol{x}) \neq h(\boldsymbol{z})], \\ \mathbf{Pr}[g(\boldsymbol{y}) \neq h(\boldsymbol{z})], \\ \mathbf{Pr}[g(\boldsymbol{y}') \neq h(\boldsymbol{z})], \\ \mathbf{Pr}[g(\boldsymbol{y}'') \neq h(\boldsymbol{z})] \end{array} \right\}$$

$$\leq \frac{3}{4} + \frac{1}{4} \mathbf{E}[4\mathsf{NAT}(f(\boldsymbol{x}), g(\boldsymbol{y}), g(\boldsymbol{y}'), g(\boldsymbol{y}''))]. \qquad (2)$$

This is because if 4NAT fails to hold on the tuple $(f(\boldsymbol{x}), g(\boldsymbol{y}), g(\boldsymbol{y}'), g(\boldsymbol{y}''))$, then $h(\boldsymbol{z})$ can disagree with at most 3 of them.

At this point, we have removed h from the test analysis and have uncovered what appears to be a hidden 4NAT test inside the 2NLin test: simply generate four strings $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{y}'$, and $\boldsymbol{y}''$ as described earlier, and test $4\mathsf{NAT}(f(\boldsymbol{x}), g(\boldsymbol{y}), g(\boldsymbol{y}'), g(\boldsymbol{y}''))$. With some renaming of variables, this is exactly what our 4NAT test does:

$$\boxed{\text{4NAT Test}}$$

Given folded functions $f : \mathbb{Z}_3^K \to \mathbb{Z}_3$, $g : \mathbb{Z}_3^{dK} \to \mathbb{Z}_3$:

- Let $\boldsymbol{x} \in \mathbb{Z}_3^K$ be uniformly random.

- Select $\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{w}$ as follows: for each $i \in [K], j \in [d]$, select $((\boldsymbol{y}[i])_j, (\boldsymbol{z}[i])_j, (\boldsymbol{w}[i])_j)$ uniformly at random from the elements of $\mathbb{Z}_3$ satisfying $\mathsf{TwoPair}(\boldsymbol{x}_i, (\boldsymbol{y}[i])_j, (\boldsymbol{z}[i])_j, (\boldsymbol{w}[i])_j)$.

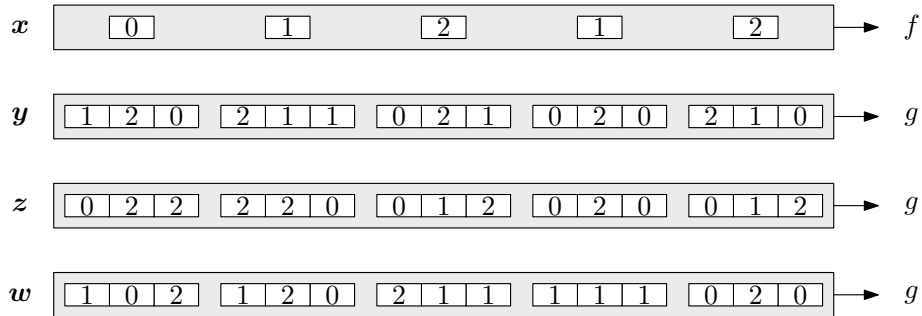- Test $4\mathsf{NAT}(f(\boldsymbol{x}), g(\boldsymbol{y}), g(\boldsymbol{z}), g(\boldsymbol{w}))$.

Figure 2: An illustration of the 4NAT test distribution; $d = 3$, $K = 5$

Above is an illustration of this test. In this illustration, the strings $\boldsymbol{z}$ and $\boldsymbol{w}$ were derived from the strings in Figure 1 using the process detailed above for generating $\boldsymbol{y}'$ and $\boldsymbol{y}''$. Note that each column is missing one of the elements of $\mathbb{Z}_3$, and that each column satisfies the TwoPair predicate. Because satisfying TwoPair implies satisfying 4NAT, matching dictators pass this test with probability 1. On the other hand, it can be seen that nonmatching dictators pass the test with probability $\frac{2}{3}$. This is basically optimal among functions with no matching influential coordinates:

Theorem 1.5 restated. *For all $\epsilon > 0$, it is NP-hard to $(1, \frac{2}{3} + \epsilon)$-decide the 4NAT problem.*

Unfortunately, it is not clear if there is a similar gadget reducing 3-Coloring to 4NAT, or to any other simple 4CSP for that matter. However, by using Equation (2), we can still reduce the analysis of the 3-Coloring test to analyzing the 4NAT test:

$$\mathbf{Pr}[f, g, \text{ and } h \text{ pass the 3-Coloring test}] \leq \frac{1}{17} \cdot p_f + \frac{4}{17} \cdot p_g$$
$$+ \frac{12}{17} \cdot \left(\frac{3}{4} + \frac{1}{4} \mathbf{E}[4\mathsf{NAT}(f(\boldsymbol{x}), g(\boldsymbol{y}), g(\boldsymbol{z}), g(\boldsymbol{w}))] \right). \quad (3)$$

Here p_f and p_g are the probabilities that f and g pass the folding test, respectively, and $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, and $\boldsymbol{w}$ are distributed as in the 4NAT test. This equation will be the focus of our 3-Coloring soundness proof.

(As one additional remark, our 2NLin test is basically the composition of the 4NAT test with the gadget from Lemma 3.1. In this test, if we instead performed the $f(\boldsymbol{x}) \neq h(\boldsymbol{z})$ test with probability $\frac{1}{3}$ and the $g(\boldsymbol{y}) \neq h(\boldsymbol{z})$ test with probability $\frac{2}{3}$, then the resulting test would basically be the composition of a 3NLin test with a suitable 3NLin-to-2NLin gadget.)

4 Fourier analysis

Let $\omega = e^{2\pi i/3}$, and set $U_3 = \{\omega^0, \omega^1, \omega^2\}$. In what follows, we identify f and g with the functions ω^f and ω^g, respectively, whose range is U_3 rather than $\mathbb{Z}_3$. Set $L = dK$. Define

$$\mathrm{Dec}(f, g) := \sum_{\alpha : \pi_3(\alpha) \neq 0} |\hat{f}(\pi_3(\alpha))| \cdot |\hat{g}(\alpha)|^2 \cdot (1/2)^{\#\alpha}.$$

This quantity corresponds to the "decodable" part of f and g. This section is devoted to proving the following two lemmas:

Lemma 4.1. *Let $f : \mathbb{Z}_3^K \to U_3$ and $g : \mathbb{Z}_3^L \to U_3$ be folded. Then the probability f and g pass the 4NAT test is at most $\frac{2}{3} + \frac{2}{3}\mathrm{Dec}(f, g)$.*

Lemma 4.2. *Let $f : \mathbb{Z}_3^K \to U_3$ and $g : \mathbb{Z}_3^L \to U_3$. Then the probability f and g pass the 3-Coloring test is at most $\frac{16}{17} + \frac{2}{17}\mathrm{Dec}(f, g)$.*

Having proven these, our hardness results follow immediately from a standard application of Håstad's method. See Section 5 for details.

The first step is to "arithmetize" the 4NAT predicate. It is not hard to verify that

$$4\mathsf{NAT}(a_1, a_2, a_3, a_4) = \frac{5}{9} + \frac{1}{9} \sum_{i \neq j} \omega^{a_i} \overline{\omega}^{a_j} - \frac{1}{9} \sum_{i < j < k} \omega^{a_i} \omega^{a_j} \omega^{a_k} - \frac{1}{9} \sum_{i < j < k} \overline{\omega}^{a_i} \overline{\omega}^{a_j} \overline{\omega}^{a_k}$$

$$= \frac{5}{9} + \frac{2}{9} \sum_{i < j} \Re[\omega^{a_i} \overline{\omega}^{a_j}] - \frac{2}{9} \sum_{i < j < k} \Re[\omega^{a_i} \omega^{a_j} \omega^{a_k}].$$

Using the symmetry between $\boldsymbol{y}$, $\boldsymbol{z}$, and $\boldsymbol{w}$, we deduce

$$
\mathbf{E}[4\mathsf{NAT}(f(\boldsymbol{x}), g(\boldsymbol{y}), g(\boldsymbol{z}), g(\boldsymbol{w}))]
$$
$$
= \tfrac{5}{9} + \tfrac{2}{3}\Re\,\mathbf{E}[f(\boldsymbol{x})\overline{g(\boldsymbol{y})}] + \tfrac{2}{3}\Re\,\mathbf{E}[g(\boldsymbol{y})\overline{g(\boldsymbol{z})}] - \tfrac{2}{3}\Re\,\mathbf{E}[f(\boldsymbol{x})g(\boldsymbol{y})g(\boldsymbol{z})] - \tfrac{2}{9}\Re\,\mathbf{E}[g(\boldsymbol{y})g(\boldsymbol{z})g(\boldsymbol{w})]. \quad (4)
$$

To analyze this expression, we will need the following lemma:

Lemma 4.3. *Let $a \in \mathbb{Z}_3$, $\beta, \gamma \in \mathbb{Z}_3^{dK}$, and i and j be such that $\pi(j) = i$. Then*

$$
\mathop{\mathbf{E}}_{\boldsymbol{y}, \boldsymbol{z}}\left[\omega^{\beta_j \boldsymbol{y}_j + \gamma_j \boldsymbol{z}_j} \mid \boldsymbol{x}_i = a\right] = \begin{cases} \left(-\tfrac{1}{2}\right)^{\#\beta_j} \omega^{2a\beta_j} & \text{if } \beta_j = \gamma_j, \\ 0 & \text{otherwise.} \end{cases}
$$

Proof. Conditioned on $\boldsymbol{x}_i = a$, the distribution on the values for $(\boldsymbol{y}_j, \boldsymbol{z}_j)$ is uniform on the six possibilities $(a, a+1)$, $(a, a+2)$, $(a+1, a)$, $(a+1, a+1)$, $(a+2, a)$, and $(a+2, a+2)$. If $\beta_j = \gamma_j$, then the expectation equals $\mathbf{E}[\omega^{\beta_j(\boldsymbol{y}_j + \boldsymbol{z}_j)} \mid \boldsymbol{x}_i = a]$. As either $\boldsymbol{y}_j + \boldsymbol{z}_j \equiv 2a + 1 \pmod{3}$ or $\boldsymbol{y}_j + \boldsymbol{z}_j \equiv 2a + 2 \pmod{3}$, each with probability half, this is equal to

$$
\frac{1}{2}\left(\omega^{\beta_j(2a+1)} + \omega^{\beta_j(2a+2)}\right) = \frac{(\omega^{\beta_j} + \omega^{2\beta_j})}{2}\omega^{2a\beta_j} = \left(-\frac{1}{2}\right)^{\#\beta_j}\omega^{2a\beta_j}.
$$

On the other hand, If $\beta_j \neq \gamma_j$, then either only one of β_j or γ_j is zero, or neither is zero, and $\gamma_j \equiv -\beta_j \pmod{3}$. In the first case, the expectation is either $\mathbf{E}[\omega^{\beta_j \boldsymbol{y}_j} \mid \boldsymbol{x}_i = a]$ or $\mathbf{E}[\omega^{\gamma_j \boldsymbol{z}_j} \mid \boldsymbol{x}_i = a]$ for a nonzero β_j or a nonzero γ_j, respectively. Both of these expectations are zero, as both $\boldsymbol{y}_j$ and $\boldsymbol{z}_j$ are uniform on $\mathbb{Z}_3$. In the second case,

$$
\mathbf{E}[\omega^{\beta_j \boldsymbol{y}_j + \gamma_j \boldsymbol{z}_j} \mid \boldsymbol{x}_i = a] = \mathbf{E}[\omega^{\beta_j \boldsymbol{y}_j - \beta_j \boldsymbol{z}_j} \mid \boldsymbol{x}_i = a]
$$
$$
= \mathbf{E}[\omega^{\beta_j(\boldsymbol{y}_j - \boldsymbol{z}_j)} \mid \boldsymbol{x}_i = a],
$$

which is zero, because β_j is nonzero and $\boldsymbol{y}_j - \boldsymbol{z}_j$ is uniformly distributed on $\mathbb{Z}_3$. $\square$

Now we use this to find an expression for a general form of the $\mathbf{E}[f(\boldsymbol{x})g(\boldsymbol{y})g(\boldsymbol{z})]$ term:

Lemma 4.4. *Let $f_1 : \mathbb{Z}_3^K \to \mathbb{R}$ and $g_1, g_2 : \mathbb{Z}_3^L \to \mathbb{R}$. Then*

$$
\mathbf{E}[f_1(\boldsymbol{x})g_1(\boldsymbol{y})g_2(\boldsymbol{z})] = \sum_{\alpha \in \mathbb{Z}_3^L} \hat{f}_1(\pi_3(\alpha))\hat{g}_1(\alpha)\hat{g}_2(\alpha)\left(-\frac{1}{2}\right)^{\#\alpha}.
$$

From this, we can derive the following two corollaries:

Corollary 4.5. *Let $g : \mathbb{Z}_3^{dK} \to \mathbb{R}$. Then*

$$
\mathbf{E}[g(\boldsymbol{y})\overline{g(\boldsymbol{z})}] = \sum_{\alpha : |\alpha[i]| \equiv 0 \ \forall i} \hat{g}(\alpha)\overline{\hat{g}(-\alpha)}\left(-\frac{1}{2}\right)^{\#\alpha}.
$$

Proof (assuming Lemma 4.4). Set $f_1 \equiv 1$, $g_1 = g$, and $g_2 = \overline{g}$. The only nonzero Fourier coefficient of f_1 is $\hat{f}_1(0) = 1$, and the only elements $\alpha \in \mathbb{Z}_3^L$ for which $\pi_3(\alpha) = 0$ are those where $|\alpha[i]| \equiv 0$ for all i. Apply Lemma 4.4 to these three functions:

$$
\mathbf{E}[g(\boldsymbol{y})\overline{g(\boldsymbol{z})}] = \sum_{\alpha : |\alpha[i]| \equiv 0 \ \forall i} \hat{g}_1(\alpha)\hat{g}_2(\alpha)\left(-\frac{1}{2}\right)^{\#\alpha}.
$$

11

Since $\hat{g}_1(\alpha) = \hat{g}(\alpha)$, it remains to show that $\hat{g}_2(\alpha) = \overline{\hat{g}(-\alpha)}$, and this is true because

$$\hat{g}_2(\alpha) = \mathbf{E}[g_2(\boldsymbol{y})\overline{\chi_\alpha(\boldsymbol{y})}] = \mathbf{E}\overline{[g(\boldsymbol{y})\chi_\alpha(\boldsymbol{y})]} = \mathbf{E}\overline{[g(\boldsymbol{y})}\chi_{-\alpha}(\boldsymbol{y})] = \overline{\mathbf{E}[g(\boldsymbol{y})\overline{\chi_{-\alpha}(\boldsymbol{y})}]} = \overline{\hat{g}(-\alpha)},$$

where the third equality follows from $\overline{\chi_\beta(\boldsymbol{z})} = \overline{\omega^{\beta \cdot \boldsymbol{z}}} = \omega^{-\beta \cdot \boldsymbol{z}} = \chi_{-\beta}(\boldsymbol{z})$. $\qquad\square$

Corollary 4.6. $-\Re\,\mathbf{E}[f(\boldsymbol{x})g(\boldsymbol{y})g(\boldsymbol{z})] \le \mathrm{Dec}(f,g) + |\hat{f}(0)|\sum_{\alpha:\pi_3(\alpha)=0}|\hat{g}(\alpha)|^2 \cdot (1/2)^{\#\alpha}.$

Proof (assuming Lemma 4.4). Applying Lemma 4.4 to f, g, and g:

$$
\begin{aligned}
-\Re\,\mathbf{E}[f(\boldsymbol{x})g(\boldsymbol{y})g(\boldsymbol{z})] &= -\Re \sum_{\alpha \in \mathbb{Z}_3^L} \hat{f}(\pi_3(\alpha))\hat{g}(\alpha)^2 \left(-\frac{1}{2}\right)^{\#\alpha} \\
&\le \sum_{\alpha \in \mathbb{Z}_3^L} |\hat{f}(\pi_3(\alpha))| \cdot |\hat{g}(\alpha)|^2 \cdot (1/2)^{\#\alpha} \\
&= \mathrm{Dec}(f,g) + \sum_{\alpha:\pi_3(\alpha)=0} |\hat{f}(0)| \cdot |\hat{g}(\alpha)|^2 \cdot (1/2)^{\#\alpha}. \qquad\square
\end{aligned}
$$

We now prove Lemma 4.4.

Proof of Lemma 4.4. Begin by expanding out $\mathbf{E}[f_1(\boldsymbol{x})g_1(\boldsymbol{y})g_2(\boldsymbol{z})]$:

$$\mathbf{E}[f_1(\boldsymbol{x})g_1(\boldsymbol{y})g_2(\boldsymbol{z})] = \sum_{\alpha \in \mathbb{Z}_3^K, \beta, \gamma \in \mathbb{Z}_3^L} \hat{f}_1(\alpha)\hat{g}_1(\beta)\hat{g}_2(\gamma)\,\mathbf{E}[\chi_\alpha(\boldsymbol{x})\chi_\beta(\boldsymbol{y})\chi_\gamma(\boldsymbol{z})]. \tag{5}$$

We focus on the products of the Fourier characters:

$$\mathbf{E}[\chi_\alpha(\boldsymbol{x})\chi_\beta(\boldsymbol{y})\chi_\gamma(\boldsymbol{z})] = \prod_{i \in [K]} \mathbf{E}[\chi_{\alpha_i}(\boldsymbol{x}_i)\chi_{\beta[i]}(\boldsymbol{y}[i])\chi_{\gamma[i]}(\boldsymbol{z}[i])] \tag{6}$$

We can attend to each block separately:

$$
\begin{aligned}
\mathbf{E}[\chi_{\alpha_i}(\boldsymbol{x}_i)\chi_{\beta[i]}(\boldsymbol{y}[i])\chi_{\gamma[i]}(\boldsymbol{z}[i])] &= \mathbf{E}\left[\omega^{\alpha_i \cdot \boldsymbol{x}_i + \beta[i] \cdot \boldsymbol{y}[i] + \gamma[i] \cdot \boldsymbol{z}[i]}\right] \\
&= \mathop{\mathbf{E}}_{\boldsymbol{x}}\left[\omega^{\alpha_i \cdot a} \prod_{j:\pi(j)=i} \underbrace{\mathop{\mathbf{E}}_{\boldsymbol{y},\boldsymbol{z}}\left[\omega^{\beta_j \boldsymbol{y}_j + \gamma_j \boldsymbol{z}_j} \mid \boldsymbol{x}_i = a\right]}_{(*)}\right]. \tag{7}
\end{aligned}
$$

Lemma 4.3 tells us that the expectation $(*)$ is zero if $\beta_j \ne \gamma_j$. Thus, if Equation (6) is to be nonzero, it must be the case that $\beta = \gamma$. If this is the case, then we can rewrite Equation (7) as

$$(7) = \mathop{\mathbf{E}}_{\boldsymbol{x}}\left[\omega^{\alpha_i \cdot a} \prod_{j:\pi(j)=i} \left(-\frac{1}{2}\right)^{\#\beta_j} \omega^{2a\beta_j}\right] = \mathop{\mathbf{E}}_{\boldsymbol{x}}\left[\left(-\frac{1}{2}\right)^{\#\beta[i]} \omega^{(\alpha_i + 2|\beta[i]|)a}\right].$$

Note that the exponent of ω, $(\alpha_i + 2|\beta[i]|)a$, is zero if $\alpha_i \equiv |\beta[i]| \pmod 3$, in which case the expectation is just the constant $(-1/2)^{\#\beta[i]}$. This occurs for all $i \in [K]$ exactly when $\alpha = \pi_3(\beta)$. If, on the other hand, $\alpha_i + 2|\beta[i]|$ is nonzero, then the entire expectation is zero because a, the

value of $\boldsymbol{x}_i$, is uniformly random from $\mathbb{Z}_3$. Thus, Equation (6) is nonzero only when $\alpha = \pi_3(\beta)$ and $\beta = \gamma$, in which case it equals

$$(6) = \left(-\frac{1}{2}\right)^{\#\beta}.$$

We may therefore conclude with

$$(5) = \sum_{\alpha \in \mathbb{Z}_3^L} \hat{f}_1(\pi_3(\alpha))\hat{g}_1(\alpha)\hat{g}_2(\alpha) \left(-\frac{1}{2}\right)^{\#\alpha}. \qquad \square$$

4.1 4NAT Analysis

In the 4NAT test, we may assume that f and g are folded, which immediately implies that $\mathbf{E}[f(\boldsymbol{x})\overline{g(\boldsymbol{y})}] = 0$. This is because $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent, and hence

$$\mathbf{E}[f(\boldsymbol{x})\overline{g(\boldsymbol{y})}] = \mathbf{E}[f(\boldsymbol{x})]\,\mathbf{E}[\overline{g(\boldsymbol{y})}] = 0 \cdot 0$$

since f and g are folded. Next, folding also implies that $\mathbf{E}[g(\boldsymbol{y})\overline{g(\boldsymbol{z})}] = 0$. To see this, first note that for any α for which $|\alpha[i]| \equiv 0$ for all i, we have that $|\alpha| \equiv 0$. Thus, any such α must satisfy $\hat{g}(\alpha) = 0$, as Proposition 2.3 implies that $\hat{g}(\alpha') \neq 0$ only when $|\alpha'| \equiv 1$. This means the sum in Corollary 4.5 must be zero, which implies that $\mathbf{E}[g(\boldsymbol{y})\overline{g(\boldsymbol{z})}] = 0$ as well.

Equation (4) has now been reduced to

$$(4) = \tfrac{5}{9} - \tfrac{2}{3}\Re\,\mathbf{E}[f(\boldsymbol{x})g(\boldsymbol{y})g(\boldsymbol{z})] - \tfrac{2}{9}\Re\,\mathbf{E}[g(\boldsymbol{y})g(\boldsymbol{z})g(\boldsymbol{w})]. \qquad (8)$$

As $g(\boldsymbol{y})g(\boldsymbol{z})g(\boldsymbol{w})$ is always in U_3, $\Re\,\mathbf{E}[g(\boldsymbol{y})g(\boldsymbol{z})g(\boldsymbol{w})]$ is always at least $-\frac{1}{2}$. Therefore,

$$(8) \le \tfrac{2}{3} - \tfrac{2}{3}\Re\,\mathbf{E}[f(\boldsymbol{x})g(\boldsymbol{y})g(\boldsymbol{z})] = \frac{2}{3} + \frac{2}{3}\mathrm{Dec}(f,g), \qquad (9)$$

using Corollary 4.6 and the fact that $\hat{f}(0) = 0$ by folding. This proves Lemma 4.1.

4.2 3-Coloring Analysis

The analysis of the 3-Coloring test is more involved, partially because we can no longer assume either of the functions are folded, and partially because we need a more careful analysis of the $\mathbf{E}[g(\boldsymbol{y})g(\boldsymbol{z})g(\boldsymbol{w})]$ term. Instead, we upper-bound these terms with expressions involving the empty coefficients $\hat{f}(0)$ and $\hat{g}(0)$, which, when large, cause the folding tests to fail with high probability. In addition, the analysis of the 3-Coloring test also involves analyzing the folding tests on f and g, and it is with these that we start.

For a function $f_1 : \mathbb{Z}_3^n \to U_3$, define $\mathrm{EVEN}(f_1) = \sum_{\alpha : |\alpha| \equiv 0} |\hat{f}_1(\alpha)|^2$.

Lemma 4.7. $\mathbf{Pr}[f(\boldsymbol{x}) \neq f(\boldsymbol{x}+1)] = 1 - \mathrm{EVEN}(f)$.

Proof. It is easy to see that $\mathbf{Pr}[f(\boldsymbol{x}) \neq f(\boldsymbol{x}+1)] = \tfrac{2}{3}\left(1 - \Re\,\mathbf{E}[f(\boldsymbol{x})\overline{f(\boldsymbol{x}+1)}]\right)$. Expanding the expectation,

$$\mathbf{E}[f(\boldsymbol{x})\overline{f(\boldsymbol{x}+1)}] = \sum_{\alpha,\beta} \hat{f}(\alpha)\overline{\hat{f}(\beta)}\,\mathbf{E}[\chi_\alpha(\boldsymbol{x})\overline{\chi_\beta(\boldsymbol{x}+1)}]$$

$$= \sum_{\alpha,\beta} \hat{f}(\alpha)\overline{\hat{f}(\beta)}\,\mathbf{E}\left[\chi_\alpha(\boldsymbol{x})\overline{\chi_\beta(\boldsymbol{x})}\omega^{-|\beta|}\right].$$

13

Since $\mathbf{E}[\chi_\alpha(\boldsymbol{x})\overline{\chi_\beta(\boldsymbol{x})}] = \mathbf{1}[\alpha = \beta]$, this equals $\sum_\alpha |\hat{f}(\alpha)|^2 \omega^{-|\alpha|}$. Taking the real part,

$$\Re \sum_\alpha |\hat{f}|^2 \omega^{-|\alpha|} = \mathrm{EVEN}(f) - \frac{1}{2}\sum_{|\alpha|\not\equiv 0}|\hat{f}|^2$$

$$= \mathrm{EVEN}(f) - \frac{1}{2}(1 - \mathrm{EVEN}(f)) = \frac{3}{2}\mathrm{EVEN}(f) - \frac{1}{2}.$$

Thus, the probability of passing the folding test is $\frac{2}{3}\left(1 - \frac{3}{2}\mathrm{EVEN}(f) + \frac{1}{2}\right) = 1 - \mathrm{EVEN}(f)$.   □

Now we focus on the $\mathbf{E}[4\mathsf{NAT}(\cdots)]$ term. Let us upper-bound the terms in Equation (4) from left to right. First,

Proposition 4.8. $\Re\,\mathbf{E}[f(\boldsymbol{x})\overline{g(\boldsymbol{y})}] \le \frac{1}{2}(|\hat{f}(0)|^2 + |\hat{g}(0)|^2)$.

Proof. By the independence of $\boldsymbol{x}$ and $\boldsymbol{y}$, $\mathbf{E}[f(\boldsymbol{x})\overline{g(\boldsymbol{y})}] = \mathbf{E}[f(\boldsymbol{x})] \cdot \mathbf{E}[\overline{g(\boldsymbol{y})}] = \hat{f}(0)\overline{\hat{g}(0)}$. Then,

$$\Re\hat{f}(0)\overline{\hat{g}(0)} \le |\hat{f}(0)| \cdot |\hat{g}(0)| \le \frac{1}{2}(|\hat{f}(0)|^2 + |\hat{g}(0)|^2),$$

using the fact that $a^2 + b^2 \ge 2ab$ for all real numbers a and b.   □

Next,

Lemma 4.9. $\Re\,\mathbf{E}[g(\boldsymbol{y})\overline{g(\boldsymbol{z})}] \le \mathrm{EVEN}(g)$.

Proof. From Corollary 4.5,

$$\Re\,\mathbf{E}[g(\boldsymbol{y})\overline{g(\boldsymbol{z})}] = \Re \sum_{\alpha:|\alpha[i]|\equiv 0\ \forall i} \hat{g}(\alpha)\overline{\hat{g}(-\alpha)}\left(-\frac{1}{2}\right)^{\#\alpha} \le \sum_{\alpha:|\alpha[i]|\equiv 0\ \forall i} |\hat{g}(\alpha)||\hat{g}(-\alpha)|\left(\frac{1}{2}\right)^{\#\alpha}.$$

By Cauchy-Schwarz, this is at most

$$\sqrt{\sum_{\alpha:|\alpha[i]|\equiv 0\ \forall i} |\hat{g}(\alpha)|^2 \left(\frac{1}{2}\right)^{\#\alpha}} \cdot \sqrt{\sum_{\alpha:|\alpha[i]|\equiv 0\ \forall i} |\hat{g}(-\alpha)|^2 \left(\frac{1}{2}\right)^{\#\alpha}} = \sum_{\alpha:|\alpha[i]|\equiv 0\ \forall i} |\hat{g}(\alpha)|^2 \left(\frac{1}{2}\right)^{\#\alpha},$$

which is clearly at most $\sum_{|\alpha|\equiv 0}|\hat{g}(\alpha)|^2 = \mathrm{EVEN}(g)$.   □

Next,

Lemma 4.10. $-\Re\,\mathbf{E}[f(\boldsymbol{x})g(\boldsymbol{y})g(\boldsymbol{z})] \le |\hat{f}(0)| \cdot \left(\frac{3}{4}|\hat{g}(0)|^2 + \frac{1}{4}\mathrm{EVEN}(g)\right) + \mathrm{Dec}(f,g)$.

Proof. By Corollary 4.6,

$$-\Re\,\mathbf{E}[f(\boldsymbol{x})g(\boldsymbol{y})g(\boldsymbol{z})] \le \mathrm{Dec}(f,g) + |\hat{f}(0)|\sum_{\alpha:\pi_3(\alpha)\equiv 0}|\hat{g}(\alpha)|^2 \cdot (1/2)^{\#\alpha}.$$

Consider the sum $\sum_{\alpha:\pi_3(\alpha)\equiv 0}|\hat{g}(\alpha)|^2 \cdot (1/2)^{\#\alpha}$. The only time that $\#\alpha = 0$ is when $\alpha = 0$. In addition, no α with $\#\alpha = 1$ contributes to the sum, because such an α cannot satisfy $\pi_3(\alpha) \equiv 0$ (one of its coordinates must be 1 or 2). Thus, the sum is upper-bounded by

$$|\hat{g}(0)|^2 + \frac{1}{4}\sum_{\alpha:\pi_3(\alpha)\equiv 0}|\hat{g}(\alpha)|^2 \le |\hat{g}(0)|^2 + \frac{1}{4}\sum_{|\alpha|=0}|\hat{g}(\alpha)|^2$$

$$= |\hat{g}(0)|^2 + \frac{1}{4}(\mathrm{EVEN}(g) - |\hat{g}(0)|^2) = \frac{3}{4}|\hat{g}(0)|^2 + \frac{1}{4}\mathrm{EVEN}(g).$$

This concludes the lemma.   □

The last term, $\mathbf{E}[g(\boldsymbol{y})g(\boldsymbol{z})g(\boldsymbol{w})]$, is more difficult to bound. The bound we use is:

Lemma 4.11. $-\Re\mathbf{E}[g(\boldsymbol{y})g(\boldsymbol{z})g(\boldsymbol{w})] \leq \frac{1}{2} - \frac{3}{2}|\hat{g}(0)|^2.$

The proof of Lemma 4.11 is presented in Appendix A.

Substituting Proposition 4.8 and Lemmas 4.9, 4.10, and 4.11 into Equation (4) and performing some arithmetic yields

$$\mathbf{E}[4\mathsf{NAT}(\cdots)] \leq \frac{2}{3} + \frac{2}{3}\mathrm{Dec}(f,g) + \frac{1}{3}|\hat{f}(0)|^2 + \frac{1}{2}|\hat{f}(0)||\hat{g}(0)|^2 + \mathrm{EVEN}(g)\left(\frac{2}{3} + \frac{|\hat{f}(0)|}{6}\right).$$

By plugging this bound into Equation 3, applying Lemma 4.7, and performing more arithmetic, we can upper bound the probability that f and g pass the 3-Coloring test by

$$-\frac{1}{17}\mathrm{EVEN}(f) - \mathrm{EVEN}(g)\left(\frac{2}{17} - \frac{|\hat{f}(0)|}{34}\right) + \frac{1}{17}|\hat{f}(0)|^2 + \frac{3}{34}|\hat{f}(0)||\hat{g}(0)|^2 + \frac{2}{17}\mathrm{Dec}(f,g) + \frac{16}{17}.$$

Note that because $0 \leq |\hat{f}(0)| \leq 1$, the coefficient of $\mathrm{EVEN}(g)$ is always negative. Thus, we may bound $-\mathrm{EVEN}(f)$ and $-\mathrm{EVEN}(g)$ by $-|\hat{f}(0)|^2$ and $-|\hat{g}(0)|^2$, respectively, resulting in a total upper bound of

$$\frac{2}{17}\left(|\hat{f}(0)||\hat{g}(0)|^2 - |\hat{g}(0)|^2\right) + \frac{2}{17}\mathrm{Dec}(f,g) + \frac{16}{17}.$$

The leftmost term is always at most zero, so this is at most $\frac{2}{17}\mathrm{Dec}(f,g) + \frac{16}{17}$, the expression claimed in Lemma 4.2.

5 Hardness of 4NAT

In this section, we show the following theorem:

Theorem 1.5 (detailed). *For all $\epsilon > 0$, it is NP-hard to $(1, \frac{2}{3} + \epsilon)$-decide the 4NAT problem. In fact, in the "yes case", all 4NAT constraints can be satisfied by TwoPair assignments.*

Combining this with Lemma 3.1 yields Theorem 1.4, and combining this with Corollary 3.3 yields Theorem 1.3. It is not clear whether this gives optimal hardness assuming perfect completeness. The 4NAT predicate is satisfied by a uniformly random input with probability $\frac{5}{9}$, and by the method of conditional expectation this gives a deterministic algorithm which $(1, \frac{5}{9})$-approximates the 4NAT CSP. This leaves a gap of $\frac{1}{9}$ in the soundness, and to our knowledge there are no better known algorithms.

On the hardness side, consider a uniformly random satisfying assignment to the TwoPair predicate. It is easy to see that each of the four variables is assigned a uniformly random value from $\mathbb{Z}_3$, and also that the variables are pairwise independent. As any satisfying assignment to the TwoPair predicate also satisfies the 4NAT predicate, the work of Austrin and Mossel [AM09] immediately implies that $(1 - \epsilon, \frac{5}{9} + \epsilon)$-approximating the 4NAT problem is NP-hard under the Unique Games conjecture. Thus, if we are willing to sacrifice a small amount in the completeness, we can improve the soundness parameter in Theorem 1.5. Whether we can improve upon the soundness without sacrificing perfect completeness is open.

We now arrive at the proof of Theorem 1.5. The proof is entirely standard, and proceeds by reduction from d-to-1 Label Cover. A nearly identical proof gives Theorem 1.1, which we omit. The proof makes use of our analysis of the 4NAT test, which is presented in Section 4. One preparatory

note: most of the proof concerns functions $f : \mathbb{Z}_3^K \to \mathbb{Z}_3$ and $g : \mathbb{Z}_3^{dK} \to \mathbb{Z}_3$. However, we also be making use of Fourier analytic notions defined in Section 2.3, and this requires dealing with functions whose range is U_3 rather than $\mathbb{Z}_3$. Thus, we associate f and g with the functions ω^f and ω^g, and whenever Fourier analysis is used it will actually be with respect to the latter two functions.

Proof. Let $G = (U \cup V, E)$ be a d-to-1 Label Cover instance with alphabet size K and d-to-1 maps $\pi_e : [dK] \to [K]$ for each edge $e \in E$. We construct a 4NAT instance by replacing each vertex in G with its Long Code and placing constraints on adjacent Long Codes corresponding to the tests made in the 4NAT test. Thus, each $u \in U$ is replaced by a copy of the hypercube $\mathbb{Z}_3^K$ and labeled by the function $f_u : \mathbb{Z}_3^K \to \mathbb{Z}_3$. Similarly, each $v \in V$ is replaced by a copy of the Boolean hypercube $\mathbb{Z}_3^{dK}$ and labeled by the function $g_v : \mathbb{Z}_3^{dK} \to \mathbb{Z}_3$. Finally, for each edge $\{u, v\} \in E$, a set of 4NAT constraints is placed between f_u and g_v corresponding to the constraints made in the 4NAT test, and given a weight equal to the probability the constraint is tested in the 4NAT test multiplied by the weight of $\{u, v\}$ in G. This produces a 4NAT instance whose weights sum to 1 which is equivalent to the following test:

- Pick an edge $e = (u, v) \in E$ uniformly at random.

- Reorder the indices of g_v so that the kth group of d indices corresponds to $\pi_e^{-1}(k)$.

- Run the 4NAT test on f_u and g_v. Accept iff it does.

Completeness   If the original Label Cover instance is fully satisfiable, then there is a function $F : U \cup V \to [dK]$ for which $\mathsf{val}(F) = 1$. Set each f_u to the dictator assignment $f_u(x) = x_{F(u)}$ and each g_v to the dictator assignment $g_v(y) = y_{F(v)}$. Let $e = \{u, v\} \in E$. Because F satisfies the constraint π_e, $F(u) = \pi_e(F(v))$. Thus, f_u and g_v correspond to "matching dictator" assignments, and above we saw that matching dictators pass the 4NAT test with probability 1. As this applies to every edge in E, the 4NAT instance is fully satisfiable.

Soundness   Assume that there are functions $\{f_u\}_{u \in U}$ and $\{g_v\}_{v \in V}$ which satisfy at least a $\frac{2}{3} + \epsilon$ fraction of the 4NAT constraints. Then there is at least an $\epsilon/2$ fraction of the edges $e = \{u, v\} \in E$ for which f_u and g_v pass the 4NAT test with probability at least $\frac{2}{3} + \epsilon/2$. This is because otherwise the fraction of 4NAT constraint satisfied would be at most

$$\left(1 - \frac{\epsilon}{2}\right)\left(\frac{2}{3} + \frac{\epsilon}{2}\right) + \frac{\epsilon}{2}(1) = \frac{2}{3} + \frac{2\epsilon}{3} - \frac{\epsilon^2}{4} < \frac{2}{3} + \epsilon.$$

Let E' be the set of such edges, and consider $\{u, v\} \in E'$. Set $L = dK$. By Lemma 4.1,

$$\frac{2}{3} + \frac{\epsilon}{2} \leq \mathbf{Pr}[f_u \text{ and } g_v \text{ pass the 4NAT test}] \leq \frac{2}{3} + \frac{2}{3}\left(\sum_{\alpha : \pi_3(\alpha) \neq 0} \left|\hat{f}_u(\pi_3(\alpha))\right| |\hat{g}_v(\alpha)|^2 \left(\frac{1}{2}\right)^{\#\alpha}\right),$$

meaning that

$$\frac{3\epsilon}{4} \leq \sum_{\alpha : \pi_3(\alpha) \neq 0} \left|\hat{f}_u(\pi_3(\alpha))\right| |\hat{g}_v(\alpha)|^2 \left(\frac{1}{2}\right)^{\#\alpha}. \tag{10}$$

Parseval's equation tells us that $\sum_{\alpha \in \mathbb{Z}_3^L} |\hat{g}_v(\alpha)|^2 = 1$. The function $\hat{g}_v$ therefore induces a probability distribution on the elements of $\mathbb{Z}_3^L$. As a result, we can rewrite Equation (10) as

$$\frac{3\epsilon}{4} \leq \mathop{\mathbf{E}}_{\alpha \sim \hat{g}_v} \left[\left| \hat{f}_u(\pi_3(\alpha)) \right| \left(\frac{1}{2} \right)^{\#\alpha} \mathbf{1}[\pi_3(\alpha) \neq 0] \right]. \tag{11}$$

As previously noted, $|\hat{f}_u(\pi_3(\alpha))|$ is less than 1 for all α, so the expression in this expectation as never greater than 1. We can thus conclude that

$$\frac{3\epsilon}{8} \leq \mathop{\mathbf{Pr}}_{\alpha \sim \hat{g}_v} \underbrace{\left[\left| \hat{f}_u(\pi_3(\alpha)) \right| \left(\frac{1}{2} \right)^{\#\alpha} \mathbf{1}[\pi_3(\alpha) \neq 0] \geq \frac{3\epsilon}{8} \right]}_{\mathsf{GOOD}_\alpha},$$

as otherwise the expectation in Equation (11) would be less than $3\epsilon/4$. Call the event in the probability GOOD_α. When GOOD_α occurs, the following happens:

- $|\hat{f}_u(\pi_3(\alpha))|^2 \geq 9\epsilon^2/64$.

- $\#\alpha \leq \log_2(8/3\epsilon)$.

- $\pi_3(\alpha) \neq 0$. As a result, $\#\alpha > 0$.

This suggests the following randomized decoding procedure for each $u \in U$: pick an element $\beta \in \mathbb{Z}_3^K$ with probability $|\hat{f}_u(\beta)|^2$ and choose one of its nonzero coordinates uniformly at random. Similarly, for each $v \in V$, pick an element $\alpha \in \mathbb{Z}_3^L$ with probability $|\hat{g}_v(\alpha)|^2$ and choose one of its nonzero coordinates uniformly at random. In both cases, nonzero coordinates are guaranteed to exist because all the f_u's and g_v's are folded.

Now we analyze how well this decoding scheme performs for the edges $e = \{u, v\} \in E'$ (we may assume the other edges are unsatisfied). Suppose that when the elements of $\mathbb{Z}_3^K$ and $\mathbb{Z}_3^L$ were randomly chosen, g_v's set α was in Good_α, and f_u's set β equals $\pi_3(\alpha)$. Then, as $\#\alpha \leq \log_2(8/3\epsilon)$, and each label in $\pi_3(\alpha)$ has at least one label in α which maps to it, the probability that matching labels are drawn is at least $1/\log_2(8/3\epsilon)$. Next, the probability that such an α and β are drawn is

$$\sum_{\alpha \in \mathsf{GOOD}} |\hat{f}_u(\pi_3(\alpha))|^2 |\hat{g}_v(\alpha)|^2 \geq \frac{9\epsilon^2}{64} \sum_{\alpha \in \mathsf{GOOD}} |\hat{g}_v(\alpha)|^2 \geq \frac{9\epsilon^2}{64} \frac{3\epsilon}{8} = \frac{27\epsilon^3}{512}.$$

Combining these, the probability that this edge is satisfied is at least $27\epsilon^3/512 \log_2(8/3\epsilon)$. Thus, the decoding scheme satisfies at least

$$\frac{27\epsilon^3}{512 \log_2(8/3\epsilon)} \cdot \frac{|E'|}{|E|} \geq \frac{27\epsilon^4}{1024 \log_2(8/3\epsilon)}$$

fraction of the Label Cover edges in expectation. By the probabilistic method, an assignment to the Label Cover instance must therefore exist which satisfies at least this fraction of the edges.

We now apply Theorem 2.1, setting the soundness value in that theorem equal to $O(\epsilon^5)$, which concludes the proof. $\square$

A   Proof of $\mathbf{E}[g(\boldsymbol{y})g(\boldsymbol{z})g(\boldsymbol{w})]$

Lemma A.1 (Lemma 4.11 restated). $-\Re\,\mathbf{E}[g(\boldsymbol{y})g(\boldsymbol{z})g(\boldsymbol{w})] \le \frac{1}{2} - \frac{3}{2}|\hat{g}(0)|^2$.

Given $\alpha, \beta, \gamma \in \mathbb{Z}_3^L$, define the predicate $\psi(\alpha, \beta, \gamma)$ to be true whenever $|\alpha[i]| + |\beta[i]| + |\gamma[i]| \equiv 0$ for all i. In addition, define the function $\Phi(\cdot, \cdot, \cdot)$ as

$$\Phi(\alpha, \beta, \gamma) = \prod_{i \in [K]} \prod_{\pi(j)=i} \left(1 - \frac{1}{2}(\#\rho_j + \#\sigma_j + \#\tau_j)\right)$$

where $\rho = \alpha + \beta$, $\sigma = \beta + \gamma$, and $\tau = \alpha + \gamma$. We will begin by deriving the following expansion for the expectation:

Lemma A.2. Let $g_1, g_2, g_3 : \mathbb{Z}_3^L \to \mathbb{Z}_3$. Then

$$\mathbf{E}[g_1(\boldsymbol{y})g_2(\boldsymbol{z})g_3(\boldsymbol{w})] = \sum_{\psi(\alpha,\beta,\gamma)} \hat{g}_1(\alpha)\hat{g}_2(\beta)\hat{g}_3(\gamma)\Phi(\alpha, \beta, \gamma).$$

Proof. Begin by expanding out $\mathbf{E}[g_1(\boldsymbol{y})g_2(\boldsymbol{z})g_3(\boldsymbol{w})]$:

$$\mathbf{E}[g_1(\boldsymbol{y})g_2(\boldsymbol{z})g_3(\boldsymbol{w})] = \sum_{\alpha,\beta,\gamma \in \mathbb{Z}_3^L} \hat{g}_1(\alpha)\hat{g}_2(\beta)\hat{g}_3(\gamma)\,\mathbf{E}[\chi_\alpha(\boldsymbol{y})\chi_\beta(\boldsymbol{z})\chi_\gamma(\boldsymbol{w})]. \tag{12}$$

We focus on the products of the Fourier characters:

$$\mathbf{E}[\chi_\alpha(\boldsymbol{y})\chi_\beta(\boldsymbol{z})\chi_\gamma(\boldsymbol{w})] = \prod_{i \in [K]} \mathbf{E}[\chi_{\alpha[i]}(\boldsymbol{y}[i])\chi_{\beta[i]}(\boldsymbol{z}[i])\chi_{\gamma[i]}(\boldsymbol{w}[i])] \tag{13}$$

We can attend to each block separately:

$$\mathbf{E}[\chi_{\alpha[i]}(\boldsymbol{y}[i])\chi_{\beta[i]}(\boldsymbol{z}[i])\chi_{\gamma[i]}(\boldsymbol{w}[i])] = \mathop{\mathbf{E}}_{\boldsymbol{x}} \left[\prod_{j:\pi(j)=i} \underbrace{\mathop{\mathbf{E}}_{\boldsymbol{y},\boldsymbol{z},\boldsymbol{w}} \left[\omega^{\alpha_j \boldsymbol{y}_j + \beta_j \boldsymbol{z}_j + \gamma_j \boldsymbol{w}_j} \mid \boldsymbol{x}_i = a\right]}_{(*)} \right]. \tag{14}$$

To analyze the expectation $(*)$, note that conditioned on $\boldsymbol{x}_i = a$, the distribution on the values for $(\boldsymbol{y}_j, \boldsymbol{z}_j)$ is uniform on the six possibilities $(a, a+1, a+1)$, $(a+1, a, a+1)$, $(a+1, a+1, a)$, $(a, a+2, a+2)$, $(a+2, a, a+2)$, $(a+2, a+2, a)$. Then the expectation $(*)$ is equal to

$$\frac{1}{6}\left(\omega^{a(\alpha_j+\beta_j+\gamma_j)}\left(\omega^{\alpha_j+\beta_j} + \omega^{2(\alpha_j+\beta_j)} + \omega^{\alpha_j+\gamma_j} + \omega^{2(\alpha_j+\gamma_j)} + \omega^{\beta_j+\gamma_j} + \omega^{2(\beta_j+\gamma_j)}\right)\right).$$

Note that $\omega^{\alpha_j+\beta_j} + \omega^{2(\alpha_j+\beta_j)} = \omega^{\rho_j} + \omega^{2\rho_j} = 2 - 3\#\rho_j$. Thus, the previous equation is equal to

$$\frac{1}{6}\left(\omega^{a(\alpha_j+\beta_j+\gamma_j)}(6 - 3\#\rho_j - 3\#\sigma_j - 3\#\tau_j)\right) = \omega^{a(\alpha_j+\beta_j+\gamma_j)}\left(1 - \frac{1}{2}(\#\rho_j + \#\sigma_j + \#\tau_j)\right)$$

Substituting this into Equation (14) yields

$$\mathbf{E}[\chi_{\alpha[i]}(\boldsymbol{y}[i])\chi_{\beta[i]}(\boldsymbol{z}[i])\chi_{\gamma[i]}(\boldsymbol{w}[i])] = \mathop{\mathbf{E}}_{\boldsymbol{x}}\left[\omega^{a(|\alpha[i]|+|\beta[i]|+|\gamma[i]|)}\prod_{j:\pi(j)=i}\left(1 - \frac{1}{2}(\#\rho_j + \#\sigma_j + \#\tau_j)\right)\right],$$

18

which is zero unless $|\alpha[i]| + |\beta[i]| + |\gamma[i]| \equiv 0$. Thus, the only Fourier coefficients α, β, and γ which contribute to Equation (12) are those which satisfy $\psi(\alpha, \beta, \gamma)$. Furthermore, for any such α, β, and γ, this equation is just

$$\prod_{j:\pi(j)=i} \left(1 - \frac{1}{2}(\#\rho_j + \#\sigma_j + \#\tau_j) \right),$$

and so Equation (13) is equal to $\Phi(\alpha, \beta, \gamma)$. This concludes the lemma. $\qquad\square$

Define $p = \mathbf{Pr}[g(\boldsymbol{y}) = 0]$, $q = \mathbf{Pr}[g(\boldsymbol{y}) = \omega]$, and $r = \mathbf{Pr}[g(\boldsymbol{y}) = \omega^2]$. For $a \in U_3$, $\mathbf{1}_a(y)$ is the indicator of the event $g(y) = a$. In the Fourier transform calculations that follow, we may write $\hat{f}^3$ for some function f. This is always shorthand for $\hat{f}(\alpha)\hat{f}(\beta)\hat{f}(\gamma)$. Similarly, given three functions f_1, f_2, and f_3, $\hat{f}_1\hat{f}_2\hat{f}_3$ is always shorthand for $\hat{f}_1(\alpha)\hat{f}_2(\beta)\hat{f}_3(\gamma)$.

We'll start by computing the value of $|\hat{g}(\vec{0})|^2$.

Proposition A.3. $|\hat{g}(\vec{0})|^2 = p^3 + q^3 + r^3 - 3 \cdot pqr$.

Proof. Direct calculation shows that $|\hat{g}(\vec{0})|^2 = p^2 + q^2 + r^2 - (pq + pr + rq)$. Then because $p + q + r = 1$,

$$
\begin{aligned}
p^2 + q^2 + r^2 - (pq + pr + rq) &= p^2 + q^2 + r^2 - (pq + pr + rq) \cdot (p + q + r) \\
&= p^2 + q^2 + r^2 - p^2q - p^2r - q^2p - q^2r - r^2p - r^2q - 3 \cdot pqr \\
&= p^3 + q^3 + r^3 - 3 \cdot pqr,
\end{aligned}
$$

where the last step replaces $p^2 - p^2q - p^2r$ with p^3, using $1 - q - r = p$, and performs similar replacements for q and r. $\qquad\square$

Before proving Lemma 4.11, we'll need the following lemma.

Lemma A.4. $\sum_{a \in U_3} \mathbf{E}[\mathbf{1}_a(\boldsymbol{x})\mathbf{1}_a(\boldsymbol{z})\mathbf{1}_a(\boldsymbol{w})] = 3 \cdot \mathbf{E}[\mathbf{1}_1(\boldsymbol{y})\mathbf{1}_\omega(\boldsymbol{z})\mathbf{1}_{\omega^2}(\boldsymbol{w})] + |\hat{g}(\vec{0})|^2$.

Proof. The LHS is equal to

$$\Re \sum_{\psi(\alpha,\beta,\gamma)} (\hat{\mathbf{1}}_1^3 + \hat{\mathbf{1}}_\omega^3 + \hat{\mathbf{1}}_{\omega^2}^3) \cdot \Phi(\alpha, \beta, \gamma) \tag{15}$$

Because $\mathbf{1}_1 + \mathbf{1}_\omega + \mathbf{1}_{\omega^2} = 1$, the Fourier coefficients of $\mathbf{1}_{\omega^2}$ may be rewritten as follows:

$$\hat{\mathbf{1}}_{\omega^2} = \mathbf{1}[\alpha = 0] - \hat{\mathbf{1}}_1 - \hat{\mathbf{1}}_\omega.$$

When $\alpha \neq \vec{0}$, the corresponding term in Equation (15) equals

$$
\begin{aligned}
\hat{\mathbf{1}}_1^3 + \hat{\mathbf{1}}_\omega^3 + \hat{\mathbf{1}}_{\omega^2}^3 &= \hat{\mathbf{1}}_1^3 + \hat{\mathbf{1}}_\omega^3 + (-\hat{\mathbf{1}}_1 - \hat{\mathbf{1}}_\omega)^3 \\
&= \hat{\mathbf{1}}_1(\beta)\hat{\mathbf{1}}_\omega(\gamma) \cdot (-\hat{\mathbf{1}}_1(\alpha) - \hat{\mathbf{1}}_\omega(\alpha)) + \hat{\mathbf{1}}_1(\gamma)\hat{\mathbf{1}}_\omega(\alpha) \cdot (-\hat{\mathbf{1}}_1(\beta) - \hat{\mathbf{1}}_\omega(\beta)) \\
&\quad + \hat{\mathbf{1}}_1(\alpha)\hat{\mathbf{1}}_\omega(\beta) \cdot (-\hat{\mathbf{1}}_1(\gamma) - \hat{\mathbf{1}}_\omega(\gamma)) \\
&= \hat{\mathbf{1}}_{\omega^2}\hat{\mathbf{1}}_1\hat{\mathbf{1}}_\omega + \hat{\mathbf{1}}_\omega\hat{\mathbf{1}}_{\omega^2}\hat{\mathbf{1}}_1 + \hat{\mathbf{1}}_1\hat{\mathbf{1}}_\omega\hat{\mathbf{1}}_{\omega^2}.
\end{aligned}
$$

For the $\alpha = \vec{0}$ case,

$$p^3 + q^3 + r^3 = 3 \cdot pqr + p^3 + q^3 + r^3 - 3 \cdot pqr = 3 \cdot pqr + |\hat{g}(\vec{0})|^2,$$

where the last step uses Proposition A.3. Substituting these into Equation (15) yields

$$(15) = |\hat{g}(\vec{0})|^2 + \Re \sum_{\psi(\alpha,\beta,\gamma)} (\hat{\mathbf{1}}_1\hat{\mathbf{1}}_\omega\hat{\mathbf{1}}_{\omega^2} + \hat{\mathbf{1}}_{\omega^2}\hat{\mathbf{1}}_1\hat{\mathbf{1}}_\omega + \hat{\mathbf{1}}_\omega\hat{\mathbf{1}}_{\omega^2}\hat{\mathbf{1}}_1) \cdot \Phi(\alpha, \beta, \gamma),$$

which equal the RHS of the lemma. $\qquad\square$

We now proceed to the proof of Lemma 4.11

Proof of Lemma 4.11. Rewrite g in terms of its indicator functions, i.e. $g = \mathbf{1}_1 + \omega \cdot \mathbf{1}_\omega + \omega^2 \cdot \mathbf{1}_{\omega^2}$. Then

$$
\begin{aligned}
\Re\, \mathbf{E}[g(\boldsymbol{y})g(\boldsymbol{z})g(\boldsymbol{w})] &= \Re \sum_{\psi(\alpha,\beta,\gamma)} \hat{g}^3 \cdot \Phi(\alpha,\beta,\gamma) \\
&= \Re \sum_{\psi(\alpha,\beta,\gamma)} (\hat{\mathbf{1}}_1 + \hat{\mathbf{1}}_\omega + \hat{\mathbf{1}}_{\omega^2})^3 \cdot \Phi(\alpha,\beta,\gamma) \\
&= \Re \sum_{\psi(\alpha,\beta,\gamma)} \sum_{a,b,c \in U_3} abc \cdot \hat{\mathbf{1}}_a \hat{\mathbf{1}}_b \hat{\mathbf{1}}_c \cdot \Phi(\alpha,\beta,\gamma) \\
&= \sum_{\psi(\alpha,\beta,\gamma)} \left(\sum_{abc=1} \hat{\mathbf{1}}_a \hat{\mathbf{1}}_b \hat{\mathbf{1}}_c - \frac{1}{2} \sum_{abc \neq 1} \hat{\mathbf{1}}_a \hat{\mathbf{1}}_b \hat{\mathbf{1}}_c \right) \cdot \Phi(\alpha,\beta,\gamma) \\
&= \sum_{abc=1} \mathbf{E}[\mathbf{1}_a(\boldsymbol{x})\mathbf{1}_b(\boldsymbol{z})\mathbf{1}_c(\boldsymbol{w})] - \frac{1}{2} \sum_{abc \neq 1} \mathbf{E}[\mathbf{1}_a(\boldsymbol{x})\mathbf{1}_b(\boldsymbol{z})\mathbf{1}_c(\boldsymbol{w})] \\
&= \sum_{abc=1} \mathbf{E}[\mathbf{1}_a(\boldsymbol{x})\mathbf{1}_b(\boldsymbol{z})\mathbf{1}_c(\boldsymbol{w})] - \frac{1}{2} \left(1 - \sum_{abc=1} \mathbf{E}[\mathbf{1}_a(\boldsymbol{x})\mathbf{1}_b(\boldsymbol{z})\mathbf{1}_c(\boldsymbol{w})] \right) \\
&= \frac{3}{2} \cdot \sum_{abc=1} \mathbf{E}[\mathbf{1}_a(\boldsymbol{x})\mathbf{1}_b(\boldsymbol{z})\mathbf{1}_c(\boldsymbol{w})] - \frac{1}{2},
\end{aligned}
$$

where the second-to-last step uses the fact that $\sum_{a,b,c \in U_3} \mathbf{E}[\mathbf{1}_a(\boldsymbol{x})\mathbf{1}_b(\boldsymbol{y})\mathbf{1}_c(\boldsymbol{z})] = 1$. This concludes the proof, as

$$
\begin{aligned}
\sum_{abc=1} \mathbf{E}[\mathbf{1}_a(\boldsymbol{x})\mathbf{1}_b(\boldsymbol{z})\mathbf{1}_c(\boldsymbol{w})] &= \sum_{a \in U_3} \mathbf{E}[\mathbf{1}_a(\boldsymbol{x})\mathbf{1}_a(\boldsymbol{z})\mathbf{1}_a(\boldsymbol{w})] + 6 \cdot \mathbf{E}[\mathbf{1}_1(\boldsymbol{x})\mathbf{1}_\omega(\boldsymbol{z})\mathbf{1}_{\omega^2}(\boldsymbol{w})] \\
&= |\hat{g}(\vec{0})|^2 + 9 \cdot \mathbf{E}[\mathbf{1}_1(\boldsymbol{x})\mathbf{1}_\omega(\boldsymbol{z})\mathbf{1}_{\omega^2}(\boldsymbol{w})] \qquad \text{(By Lemma A.4.)} \\
&\geq |\hat{g}(\vec{0})|^2,
\end{aligned}
$$

using the fact that $\mathbf{E}[\mathbf{1}_1(\boldsymbol{x})\mathbf{1}_\omega(\boldsymbol{z})\mathbf{1}_{\omega^2}(\boldsymbol{w})] \geq 0$. $\qquad\square$

References

[ABS10] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for Unique Games and related problems. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 563–572, 2010. 1

[AM09]   Per Austrin and Elchanan Mossel. Approximation resistant predicates from pairwise independence. *Computational Complexity*, 18(2):249–271, 2009. 5

[AOW12] Per Austrin, Ryan O'Donnell, and John Wright. A new point of NP-hardness for 2-to-1 Label-Cover. In *Proceedings of the 15th Annual International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 1–12, 2012. ∗

[BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science*, 2011. 1

[DMR09]   Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring. *SIAM Journal on Computing*, 39(3):843–873, 2009. 1

[FJ97]   Alan Frieze and Mark Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica*, 18(1):67–81, 1997. 1

[GKO+10]   Venkatesan Guruswami, Subhash Khot, Ryan O'Donnell, Preyas Popat, Madhur Tulsiani, and Yi Wu. SDP gaps for 2-to-1 and other Label-Cover variants. In *Proceedings of the 37th Annual International Colloquium on Automata, Languages and Programming*, pages 617–628, 2010. 1

[GLST98]   Venkatesan Guruswami, Daniel Lewin, Madhu Sudan, and Luca Trevisan. A tight characterization of NP with 3 query PCPs. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 8–17, 1998. 1

[GS09]   Venkatesan Guruswami and Ali Kemal Sinop. Improved inapproximability results for Maximum k-Colorable Subgraph. In *Proceedings of the 12th Annual International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 163–176, 2009. 1, 1, 1, 1, 1.1

[GS11]   Venkatesan Guruswami and Ali Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for quadratic integer programming with PSD objectives. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science*, 2011. 1

[GW04]   Michel Goemans and David Williamson. Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming. *Journal of Computer & System Sciences*, 68(2):442–470, 2004. 1, 1.1

[Hås01]   Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. 1.2, 2.1

[IP01]   Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. 1

[Kho02]   Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 767–775, 2002. 1, 1

[MR10]   Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *Journal of the ACM*, 57(5):29, 2010. 1, 2.1

[OW12]   Ryan O'Donnell and John Wright. A new point of NP-hardness for Unique-Games. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 289–306, 2012. 1.1, 1.1, 2.4

[Pet94]   Erez Petrank. The hardness of approximation: gap location. *Computational Complexity*, 4:133–157, 1994. 1

[Rag08]   Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 245–254, 2008. 1

[Ste10]   David Steurer. Subexponential algorithms for d-to-1 two-prover games and for certifying almost perfect expansion. Available at the author's website, 2010. 1

[TSSW00]   Luca Trevisan, Gregory Sorkin, Madhu Sudan, and David Williamson. Gadgets, approximation, and linear programming. *SIAM Journal on Computing*, 29(6):2074–2097, 2000. 1.1, 2.2