Electronic Colloquium on Computational Complexity, Report No. 13 (2015)

On Hardness of Approximating the Parameterized Clique Problem

Subhash Khot * Igor Shinkar †

January 28, 2015

Abstract

In the Gap-clique$(k, \frac{k}{2})$ problem, the input is an n-vertex graph G, and the goal is to decide whether G contains a clique of size k or contains no clique of size $\frac{k}{2}$. It is an open question in the study of fixed parameterized tractability whether the Gap-clique$(k, \frac{k}{2})$ problem is fixed parameter tractable, i.e., whether it has an algorithm that runs in time $f(k) \cdot n^{\alpha}$, where $f(k)$ is an arbitrary function of the parameter k and the exponent α is a constant independent of k.

In this paper, we give some evidence that the Gap-clique$(k, \frac{k}{2})$ problem is not fixed parameter tractable. Specifically, we define a constraint satisfaction problem, which we call Deg-2-sat, where the input is a system of k' quadratic equations in k' variables over a finite field $\mathbb{F}$ of size n', and the goal is to decide whether there is a solution in $\mathbb{F}$ that satisfies all the equations simultaneously. The main result in this paper is an "FPT-reduction" from Deg-2-sat to the Gap-clique$(k, \frac{k}{2})$ problem. If one were to hypothesize that the Deg-2-sat problem is not fixed parameter tractable, then our reduction would imply that the Gap-clique$(k, \frac{k}{2})$ problem is not fixed parameter tractable either. The reduction relies on the algebraic techniques used in proof of the PCP theorem.

*Courant Institute of Mathematical Sciences, New York University. Research supported by NSF grants CCF 1422159, 1061938, 0832795 and Simons Collaboration on Algorithms and Geometry grant.

†Courant Institute of Mathematical Sciences, New York University. Same funding as Subhash Khot.

ISSN 1433-8092

1 Introduction

Parameterized complexity is a promising approach to cope with $\mathcal{NP}$-hard problems [DF99, FG06]. For many $\mathcal{NP}$-hard problems, the input consists of a pair (x, k) where k is an integer parameter and x is the "actual" input with size $|x| = n$. For instance, the input for the VERTEX-COVER problem is a pair (G, k) where G is an n-vertex graph, and the goal is to decide whether G has a vertex cover of size at most k. This is a well-known $\mathcal{NP}$-hard problem and a brute-force algorithm that tries out all vertex subsets of size k runs in time $O(n^k)$. It is not difficult to see that there is another algorithm that runs in time $O(2^k \cdot n^2)$: pick an edge of the graph, choose one of its endpoints to include in the vertex cover, remove all edges incident on the chosen endpoint, and repeat this step until at most k vertices are chosen. The algorithm accepts if no edges are left in the graph. The factor 2^k in the running time corresponds to trying out each of the two choices in the (at most) k steps. Thus the VERTEX-COVER problem is tractable for "fixed" values of the parameter k.

More generally, a problem parameterized by k is said to be *fixed-parameter tractable* (FPT) if it can be solved in time $f(k) \cdot n^\alpha$, where f is an arbitrary function depending only on k and α is a constant independent of k. For some $\mathcal{NP}$-hard problems, e.g. VERTEX-COVER as mentioned above and LONGEST PATH as another example, such an algorithm exists, while for some problems, e.g. CLIQUE, such algorithm is not known. Downey and Fellows [DF95a, DF95b] define a hierarchy of classes of parameterized problems

$$\text{FPT} \subseteq W[1] \subseteq W[2] \subseteq \ldots \subseteq W[SAT] \subseteq W[P],$$

and identify complete problems for these classes. Each class inclusion above is believed to be strict. In particular, classes FPT and $W[1]$ are thought of as analogues of the classes $\mathcal{P}$ and $\mathcal{NP}$ respectively, and are believed to be distinct. It has been shown in [DF95b] that the CLIQUE problem is $W[1]$-complete under "FPT-reduction" defined below.

Definition 1.1. *Given two parameterized problems A and B, an FTP-reduction from A to B is an algorithm that gets as input an instance (x, k) of A and outputs an instance (x', k') of B such that:*

1. *$(x, k) \in A$ if and only if $(x', k') \in B$.*
2. *k' depends only on k, in an arbitrary manner, but not on x.*
3. *The running time of the reduction is $f(k) \cdot |x|^\beta$ where f is an arbitrary function depending only on k and β is a constant independent of k.*

If such a reduction exists, then we write $A \leqslant_{\text{FPT}} B$.

It is easily seen that the class FPT of fixed parameterized tractable problems is closed under FPT-reductions, that is, if $B \in \text{FPT}$ and $A \leqslant_{\text{FPT}} B$, then $A \in \text{FPT}$. Since CLIQUE is $W[1]$-complete, it is considered unlikely that CLIQUE has a FPT-algorithm. It is, therefore, natural to ask whether CLIQUE has a good "FPT approximation algorithm", i.e.

given a graph G that is guaranteed to contain a clique of size k, the goal would be to find a clique of size $\rho(k)$ for some monotone function $\rho(k)$, e.g. $\frac{k}{2}$, $\sqrt{k}$ or even $\log k$. However, the fixed parameter complexity of the approximation problem, for CLIQUE as well as most other natural problems, is poorly understood. In particular, no FPT approximation algorithm is known for CLIQUE for any unbounded function $\rho(k)$ and on the other hand, there is no evidence that the approximation problem is hard either [Mar08, CGG06].

1.1 Our result

In this paper, we give some evidence that the CLIQUE problem is hard to approximate in the parameterized setting. Specifically, we show that there is an FPT-reduction from a problem that we call DEG-2-SAT to the "gap version" of the CLIQUE problem. We first define both these problems and then remark on the plausible hardness of the DEG-2-SAT problem.

Definition 1.2. *For a constant $0 < \varepsilon < 1$, GAP-CLIQUE$(k, \varepsilon k)$ is the following promise problem: given a k-partite graph G with n-vertices in each part, the goal is to decide whether G has a clique of size k or has no clique of size εk.*

Clearly, GAP-CLIQUE$(k, \varepsilon k)$ can be solved in time $O(n^k)$ (or even $O(n^{\varepsilon k})$). Next, we define the DEG-2-SAT problem that is central to this paper.

Definition 1.3. DEG-2-SAT$(\mathbb{F}, k)$ *is the following problem: given a finite field $\mathbb{F}$ of size n and a system of k quadratic equations*

$$p_1(x_1, \ldots, x_k) = 0, \quad \ldots \quad p_k(x_1, \ldots, x_k) = 0,$$

in k variables $x_1, \ldots, x_k$, the goal is to decide whether there is a solution $x = (x_1, \ldots, x_k) \in \mathbb{F}^k$ that satisfies all the equations simultaneously.

Note that DEG-2-SAT, and also DEG-d-SAT where the equations have degree d, has a trivial algorithm with running time $O(n^k)$. The algorithm simply tries every possible assignments to $x_1, \ldots, x_k \in \mathbb{F}$, and checks whether it satisfies all the equations. Solving systems of polynomial equations is a classical and well studied problem. For a comprehensive study of the topic, we refer to the book of von zur Gathen and Gerhard [vzGG03], and quote a few of the known results here. Given a system of degree d equations over m variables, an algorithm of Buchberger uses Gröbner basis to find a solution to the DEG-d-SAT problem in an *extension field of* $\mathbb{F}$, if a solution exists, in time $d^{\exp(m)} \cdot \operatorname{poly}\log(|\mathbb{F}|)$. However, note that the algorithm does not necessarily find a solution *in the field* $\mathbb{F}$. We also note that if the number of solutions is finite in the closure of $\mathbb{F}$ (known as the "zero-dimensional" case), then there are algorithms that find all the solutions in time $f(d, m) \cdot \operatorname{poly}\log(|\mathbb{F}|)$, see e.g., [Laz79]. Still, we are not aware of an FPT-algorithm for DEG-2-SAT that finds a solution in the field $\mathbb{F}$, and it might be the case that no FPT-algorithm exists.

Note also that for a field $\mathbb{F}$ of size $|\mathbb{F}| = n$ and the parameter k, there are only $n^{O(k^3)}$ instances of the DEG-2-SAT problem. This is because each of the k equations contains $O(k^2)$ monomials and the instance is completely specified by $O(k^3)$ coefficients of all these

monomials. In this respect, the problem differs from the standard problems in $W[1]$, e.g. CLIQUE, where there are exponentially many instances of size n. Nonetheless, we do not know whether the fact that there are only $n^{O(k^3)}$ instances necessarily rules out the possibility that DEG-2-SAT is hard, or even $W[1]$-hard. Indeed, a complexity class known as MINI[1] defined in [DECF+03] has the property that the languages in MINI[1] contain only n^k instances of size n. It has been shown in [DECF+03] that FPT $\subseteq$ MINI[1] $\subseteq W[1]$, and to the best of our knowledge, it is plausible that the containments above are strict. The main result in this paper is an FPT-reduction from DEG-2-SAT to GAP-CLIQUE.

Theorem 1.4 (Main Theorem)**.** *Let k be a parameter and let $\mathbb{F}$ be a finite field. There is an* FPT*-reduction*

$$\text{DEG-2-SAT}(\mathbb{F}, k) \leqslant_{\text{FPT}} \text{GAP-CLIQUE}\left(k', \frac{k'}{2}\right).$$

We note that, by definition of an FPT*-reduction, k' depends only on k but not on $\mathbb{F}$.*

Thus, if[1] there is no FPT-algorithm for DEG-2-SAT$(\mathbb{F}, k)$, then one may conclude that there is no FPT-algorithm for GAP-CLIQUE $\left(k', \frac{k'}{2}\right)$ either. For the GAP-CLIQUE problem, the "gap" can be amplified by a standard graph product operation, so for any constant C, one may conclude that there is no FPT-approximation for CLIQUE with approximation factor C. It is likely that hardness of approximating CLIQUE implies hardness of approximating other problems in parameterized setting, but we leave out this aspect from the current paper.

2 Proof of Theorem 1.4

Towards proving Theorem 1.4, we work with a more general version of the DEG-2-SAT problem than the version specified in Definition 1.3. The general version allows the number of variables, the number of equations and the arity of equations to be separate parameters. Also, the instance is supposed to be a "gap instance", i.e. it is promised to be either fully satisfiable or far from satisfiable, and the "gap" itself is an additional parameter.

Definition 2.1. *An instance Φ of* DEG-2-SAT$(\mathbb{F}, k, e, q, \varepsilon)$ *consists of a system of quadratic equations in k variables over the field $\mathbb{F}$. The number of equations is e and each equation depends on only q out of the k variables.*

Let val(Φ) *denote the maximum fraction of equations that can be satisfied by any assignment (over the field $\mathbb{F}$) to the variables. The instance is a promise instance where either* val$(\Phi) = 1$ *(the YES instance) or else* val$(\Phi) \leqslant \varepsilon$ *(the NO instance).*

With this definition, we note that DEG-2-SAT$(\mathbb{F}, k)$ as specified in Definition 1.3, is now denoted as

$$\text{DEG-2-SAT}\left(\mathbb{F}, k, e = k, q = k, \varepsilon = 1 - \frac{1}{k}\right),$$

[1] We stress that we are not proposing this as conjecture.

i.e. the number of variables and equations in the system is both k, each equation may depend on all $q = k$ variables, and the instance is either satisfiable or not satisfiable, with $\mathrm{val}(\Phi) \leqslant 1 - \frac{1}{k}$ in the latter case. Note that there is really no "gap" here.

2.1 Overview of the overall reduction

Our reduction starts with an instance Φ of the DEG-2-SAT $\left(\mathbb{F}, k, e = k, q = k, \varepsilon = 1 - \frac{1}{k}\right)$ problem and then transforms it, through a sequence of steps, to an instance Φ' that has a constant gap and each equation has a constant arity. From the instance Φ', it is easy to construct an instance of GAP-CLIQUE by the well-known "FGLSS reduction". We give a quick overview of the steps involved before presenting the actual reductions.

Creating gap: In the first step, we give a FPT-reduction that "creates" a constant gap:

$$\text{DEG-2-SAT}(\mathbb{F}, k, e = k, q = k, \varepsilon = 1 - \frac{1}{k}) \leqslant_{\text{FPT}} \text{DEG-2-SAT}(\mathbb{F}, k, e' = 2k, q' = k, \varepsilon' = 0.5).$$

The number of variables stays the same, the number of equations doubles, each equation may still depend on all the variables, but now, in the NO case, the instance is only 0.5-satisfiable. The field $\mathbb{F}$ stays the same in this step as well as all the subsequent steps.

Simplifying equations: In the second step, we construct an instance where the equations have a certain simplified form (the corresponding problem is referred to as SIMPLE-DEG-2-SAT):

$$\text{DEG-2-SAT}(\mathbb{F}, k, e, q = k, \varepsilon = 0.5) \leqslant_{\text{FPT}} \text{SIMPLE-DEG-2-SAT}(\mathbb{F}, k', e', q' = k', \varepsilon' = 0.95).$$

The number of variables k' and the number of equations e' depend only on their initial number k and e respectively, each equation may still depend on all the variables, and the gap suffers (which is not much of an issue; it is still bounded away from 1). The main feature of this reduction is that in the new instance, each equation is of the form

$$\ell_1(x) = a \cdot \ell_2(x) \cdot \ell_3(x) + b \cdot \ell_4(x) + c,$$

where $a, b, c \in \mathbb{F}$ and $\ell_1, \ell_2, \ell_3, \ell_4$ are linear forms over the set of variables. Moreover, the coefficients of these linear forms are from a subset $L \subseteq \mathbb{F}$ such that $|L|$ is "small", depending only on k.

Reducing arity to constant: In the third step, starting with the "simple instance" as above, we construct an instance where each equation depends only on a constant number of variables:

$$\text{SIMPLE-DEG-2-SAT}(\mathbb{F}, k, e, q = k, \varepsilon = 0.95) \leqslant_{\text{FPT}} \text{DEG-2-SAT}(\mathbb{F}, k', e', q' = O(1), \varepsilon' = 0.999).$$

The soundness suffers, but is still a constant bounded away from 1. The number of variables k' and the number of equations e' depend only on their initial number k and e respectively and on $|L|$ where $L \subseteq \mathbb{F}$ is the set of coefficients of the linear forms in the simple instance.

FGLSS reduction: Given a gap instance with equations of constant arity, it is straightforward to construct a gap instance of the CLIQUE problem, with the same gap.

$$\text{DEG-2-SAT}(\mathbb{F}, k, e, q = O(1), \varepsilon = 0.999) \leqslant_{\text{FPT}} \text{GAP-CLIQUE}(k', 0.999k').$$

The graph is k'-partite and either has a clique of size k' or has no clique of size $0.999k'$. Here $k' = e$ and the number of vertices in each of the k' groups of the k'-partite graph is at most $n^{O(1)}$ where $n = |\mathbb{F}|$.

Combining the sequence of four reductions above, we get the desired reduction

$$\text{DEG-2-SAT}(\mathbb{F}, k) \leqslant_{\text{FPT}} \text{GAP-CLIQUE}(k', 0.999k') \leqslant_{\text{FPT}} \text{GAP-CLIQUE}(k'', 0.5k''),$$

where, at the end, the gap in the GAP-CLIQUE problem is boosted from 0.999 to 0.5 by the standard operation of graph products.

The reductions are based on standard techniques used in the algebraic proof of the PCP Theorem [FGL+96, ALM+98, AS98], though there are some new variations and ingredients. Of the four reductions, the first and the fourth are straightforward, so we present them first.

2.1.1 Creating gap

We present a FPT-reduction that creates a constant gap to begin with:

$$\text{DEG-2-SAT}\left(\mathbb{F}, k, e = k, q = k, \varepsilon = 1 - \frac{1}{k}\right) \leqslant_{\text{FPT}} \text{DEG-2-SAT}(\mathbb{F}, k, e' = 2k, q' = k, \varepsilon' = 0.5).$$

Let Φ be the instance of DEG-2-SAT $\left(\mathbb{F}, k, e = k, q = k, \varepsilon = 1 - \frac{1}{k}\right)$ with equations $p_1 = 0, \ldots, p_k = 0$ in k variables over the field $\mathbb{F}$. We may assume that $|\mathbb{F}| = n \gg 2k$. We take a $2k \times k$ matrix M over the field $\mathbb{F}$ such that for every $v \in \mathbb{F}^k, v \neq 0$, it holds that at least half of the co-ordinates of Mv are non-zero. Such matrix can be constructed, e.g., by taking the generator matrix of the degree-k Reed-Solomon code over the field $\mathbb{F}$ restricted to $2k$ elements in the field. More specifically, we can define M by taking $2k$ distinct elements $a_1, \ldots a_{2k} \in \mathbb{F}$, and letting $M_{i,j} = a_i^{j-1}$.

Now construct an instance Φ' of DEG-2-SAT with the same set of variables as Φ, but whose equations are linear combinations of equations of Φ using the rows of the matrix M as coefficients. Specifically, for every $i \in \{1, \ldots, 2k\}$, the instance Φ' contains an equation $p'_i = 0$ where

$$p'_i = \sum_{j=1}^{k} M_{ij} p_j.$$

Clearly, if Φ has a satisfying assignment, the same assignment also satisfies all the equations of Φ'. On the other hand, if Φ has no satisfying assignment, any assignment $x \in \mathbb{F}^k$ satisfies at most half of the equations in Φ'. This is because,

$$(p'_1(x), \ldots, p'_k(x))^{\mathsf{T}} = M \cdot (p_1(x), \ldots, p_k(x))^{\mathsf{T}},$$

and since the vector $v = (p_1(x), \ldots, p_k(x))^{\mathsf{T}}$ is non-zero, at least half of the co-ordinates of Mv are non-zero, meaning at least half of the equations $p'_1(x) = 0, \ldots, p'_k(x) = 0$ fail.

2.1.2 FGLSS reduction

We describe a FPT-reduction (known as the FGLSS reduction [FGL+96]) from DEG-2-SAT with constant gap and constant arity to the GAP-CLIQUE problem:

$$\text{DEG-2-SAT}(\mathbb{F}, k, e, q = O(1), \varepsilon = 0.999) \leqslant_{\text{FPT}} \text{GAP-CLIQUE}(k', 0.999k'),$$

where $k' = e$ and the GAP-CLIQUE instance is a k'-partite graph with at most $|\mathbb{F}|^q = n^q$ vertices in each group. Since $q = O(1)$, the exponent of n is independent of the parameters k and e.

Given an instance Φ of DEG-2-SAT$(\mathbb{F}, k, e, q = O(1), \varepsilon = 0.999)$, construct an e-partite graph $G = (V = (V_1, \ldots, V_e), E)$ as follows. For each equation $p_i = 0$ of Φ, $i \in \{1, \ldots, e\}$, the group of vertices V_i contains at most $|\mathbb{F}|^q$ vertices, where each vertex corresponds to a satisfying assignment to the variables of the equation $p_i = 0$. We note here that p_i depends only on q variables. There is an edge between two vertices in the graph G if the corresponding assignments to the variables are consistent, i.e., if the assignments agree on the shared variables. It is easily seen that there is a one-to-one correspondence between assignments that satisfy ℓ equations of Φ and cliques of size ℓ in G.

2.2 Simplifying equations

In this section, we describe the reduction that leads to quadratic equations with a very simple structure:

Lemma 2.2. *There is an* FPT*-reduction*

$$\text{DEG-2-SAT}(\mathbb{F}, k, e, q = k, \varepsilon = 0.5) \leqslant_{\text{FPT}} \text{SIMPLE-DEG-2-SAT}(\mathbb{F}, k', e', q' = k', \varepsilon' = 0.95),$$

mapping an instance Φ *of* DEG-2-SAT *to an instance* Φ' *of* SIMPLE-DEG-2-SAT *such that:*

- k', e' *depend only on* k, e*.*
- *Each equation may still depend on all the variables.*
- *Each equation is of the form:*

$$\ell_1 = a \cdot \ell_2 \cdot \ell_3 + b \cdot \ell_4 + c,$$

  where $a, b, c \in \mathbb{F}$ *and* $\ell_1, \ell_2, \ell_3, \ell_4$ *are linear forms over the set of variables. Moreover, the coefficients of these linear forms are from a subset* $L \subseteq \mathbb{F}$ *such that* $|L|$ *depends only on* k*.*

Proof. The reduction uses algebraic ingredients used to prove the PCP Theorem, in particular the *polynomial encoding method* and the *sum check protocol* [LFKN92, Sha92]. However, we use these ingredients in a somewhat different and restricted manner. For convenience of the reader, the reduction below is presented directly, without using the language of probabilistic verifiers.

Let $S = \{s_1, \ldots, s_k\} \subseteq \mathbb{F}$ be a subset of k field elements and H be a subset of $10k^2$ field elements such that $S \subseteq H \subseteq \mathbb{F}$. Denote a typical (quadratic) equation of Φ as $p = 0$ over the variables $x_1, \ldots, x_k$. We first define the variables of Φ'. There are two kinds of variables:

1. Let $\sigma : \{x_1, \ldots, x_k\} \to \mathbb{F}$ be a supposed satisfying assignment to Φ. Thus there exists a (unique) univariate polynomial $Q(z)$ of degree at most $k-1$ such that $Q(s_i) = \sigma(x_i)$ for all $i = 1, \ldots, k$. The instance Φ' has variables $q_0, \ldots, q_{k-1}$ representing the coefficients of the polynomial $Q(z)$, i.e. $Q(z) = \sum_{i=0}^{k-1} q_i z^i$. To state differently, the instance Φ' has variables $q_0, \ldots, q_{k-1}$ and the intention is that defining a polynomial $Q(z) = \sum_{i=0}^{k-1} q_i z^i$, the values $Q(s_1), \ldots, Q(s_k)$ serve as a supposed satisfying assignment to Φ.

2. Suppose a typical equation in Φ is $p = 0$ where

$$p(x_1, \ldots, x_k) = \sum_{1 \leqslant i,j \leqslant k} c_{i,j} x_i x_j + \sum_{1 \leqslant i \leqslant k} c_i x_i + c_0.$$

   Let $C_2(u, v)$ be a bi-variate polynomial of degree $k-1$ in each variable such that $C_2(s_i, s_j) = c_{i,j}$ for all $i, j = 1, \ldots, k$. Similarly, let $C_1(u)$ be a univariate polynomial of degree $k-1$ such that $C_1(s_i) = c_i$ for all $i = 1, \ldots, k$. Note that the polynomials C_1 and C_2 depend only on the coefficients of p, and hence can be computed explicitly.

   The instance Φ' will have variables that represent the coefficients of a bi-variate polynomial Ψ_2^p and the intention is that

$$\Psi_2^p(u, v) = C_2(u, v) Q(u) Q(v).$$

   Note that Ψ_2^p is intended to have degree at most $2k-2$ in each variable. Denote its coefficients by $\{\psi_{i,j}^p : i, j = 0, \ldots, 2k-2\}$ so that these are variables of the instance Φ' and $\Psi_2^p(u, v) = \sum_{i,j=0}^{2k-2} \psi_{i,j}^p u^i v^j$.

   Similarly the instance Φ' will have variables that represent the coefficients of a univariate polynomial Ψ_1^p and the intention is that

$$\Psi_1^p(u) = C_1(u) Q(u).$$

   Note that Ψ_1^p is intended to have degree at most $2k-2$. Denote its coefficients by $\{\psi_i^p : i = 0, \ldots, 2k-2\}$ so that these are variables of the instance Φ' and $\Psi_1^p(u) = \sum_{i=0}^{2k-2} \psi_i^p u^i$.

We describe the equations of Φ' by describing how to pick one equation at random from the set of its equations. To pick an equation of Φ' at random, first pick an equation $p = 0$ of Φ at random and then, with probability $\frac{1}{3}$ each, write one the three equations below:

- Write the equation

$$\sum_{u,v \in S} \Psi_2^p(u, v) + \sum_{u \in S} \Psi_1^p(u) = -c_0.$$

More concretely, the equation, in terms of variables $\psi^p_{i,j}$ and ψ^p_i is

$$\sum_{i,j=0}^{2k-2} \psi^p_{i,j} \left(\sum_{u,v \in S} u^i v^j \right) + \sum_{i=0}^{2k-2} \psi^p_i \left(\sum_{u \in S} u^i \right) = -c_0.$$

.

- Pick $u, v \in H$ at random and write the equation

$$\Psi^p_2(u,v) = C_2(u,v) Q(u) Q(v).$$

More concretely, the equation, in terms of variables $\psi^p_{i,j}$ and $q_0, \ldots, q_{k-1}$ is

$$\sum_{i,j=0}^{2k-2} \psi^p_{i,j} \cdot u^i v^j = C_2(u,v) \cdot \left(\sum_{i=0}^{k-1} q_i \cdot u^i \right) \cdot \left(\sum_{j=0}^{k-1} q_j \cdot v^j \right),$$

where $C_2(u,v) \in \mathbb{F}$ is explicitly computed.

- Pick $u \in H$ at random and write the equation

$$\Psi^p_1(u) = C_1(u) Q(u).$$

More concretely, the equation, in terms of variables ψ^p_i and $q_0, \ldots, q_{k-1}$ is

$$\sum_{i=0}^{2k-2} \psi^p_i \cdot u^i = C_1(u) \cdot \left(\sum_{i=0}^{k-1} q_i \cdot u^i \right),$$

where $C_1(u) \in \mathbb{F}$ is explicitly computed.

This completes the description of the instance Φ' and now we proceed to show the stated properties of the instance Φ' and correctness of the reduction. Clearly, the number of variables and equations in Φ' depends only on their numbers in the instance Φ (strictly speaking, the equations in Φ' have weights, but making copies of equations proportional to their weights, Φ' can easily be turned into an un-weighted instance). Also, each equation is of the form

$$\ell_1 = a \cdot \ell_2 \cdot \ell_3 + b \cdot \ell_4 + c,$$

with $a, b, c \in \mathbb{F}$ and $\ell_1, \ell_2, \ell_3, \ell_4$ are linear forms (possibly zero) in the variables of Φ'. Finally, the coefficients of these linear forms are of the type

$$\sum_{u,v \in S} u^i v^j, \quad \sum_{u \in S} u^i, \quad u^i v^j, \quad u^i, \quad 0,$$

with $u, v \in H$ and $0 \leqslant i, j \leqslant 2k-2$. There are at most $O(k^6)$ possibilities for these coefficients. Now we prove the correctness of the reduction.

2.2.1 YES Case

We show that if $\mathrm{val}(\Phi) = 1$, then $\mathrm{val}(\Phi') = 1$. This is simply by design, but we present the details for the convenience of the reader. Let $\sigma : \{x_1, \ldots, x_k\} \to \mathbb{F}$ be an assignment that satisfies every equation $p = 0$ in Φ. Define the assignment to variables of Φ', i.e. to the variables $q_0, \ldots, q_{k-1}, \psi^p_{i,j}, \psi^p_i$ so that (the polynomials C_1, C_2 depend on the equation p though our notation suppresses this):

$$Q(z) = \sum_{i=0}^{k-1} q_i z^i,\ Q(s_i) = \sigma(x_i),\ \Psi^p_2(u,v) = C_2(u,v)Q(u)Q(v),\ \Psi^p_1(u) = C_1(u)Q(u).$$

Now, we verify that this assignment satisfies each of the three kinds of equations in Φ'. The second and the third kind of equations are satisfied by definition of $\Psi^p_2(u,v)$ and $\Psi^p_1(u)$ as above. For the equations of the first kind, we have

$$\begin{aligned}
\sum_{u,v\in S} \Psi^p_2(u,v) + \sum_{u\in S} \Psi^p_1(u) &= \sum_{u,v\in S} C_2(u,v)Q(u)Q(v) + \sum_{u\in S} C_1(u)Q(u) \\
&= \sum_{i,j=1}^{k} C_2(s_i,s_j)Q(s_i)Q(s_j) + \sum_{i=1}^{k} C_1(s_i)Q(s_i) \\
&= \sum_{i,j=1}^{k} c_{i,j}\cdot\sigma(x_i)\sigma(x_j) + \sum_{i=1}^{k} c_i\cdot\sigma(x_i) \\
&= -c_0,
\end{aligned}$$

where in the last step, we used the fact that σ satisfies the equation $p = 0$.

2.2.2 NO Case

Now we show that if $\mathrm{val}(\Phi) \leqslant 0.5$, then $\mathrm{val}(\Phi') \leqslant 0.95$. Suppose on the contrary that $\mathrm{val}(\Phi') \geqslant 0.95$ and fix a corresponding "highly satisfying" assignment to Φ', i.e. it is an assignment to the variables $q_0, \ldots, q_{k-1}, \psi^p_{i,j}, \psi^p_i$. As before, $p = 0$ denotes a typical equation in Φ. We may define formal polynomials

$$Q(z) = \sum_{i=0}^{k-1} q_i z^i, \quad \Psi^p_2(u,v) = \sum_{i,j=0}^{2k-2} \psi^p_{i,j} u^i v^j, \quad \Psi^p_1(u) = \sum_{i=0}^{2k-2} \psi^p_i u^i.$$

Since the assignment to Φ' satisfies at least 0.95 fraction of its equations, by an averaging argument, it must be the case that for at least 0.55 fraction of the equations $p = 0$ in Φ, after picking the equation $p = 0$, for each of the three kinds of equations in Φ', at least 0.5 fraction of the equations of that kind are satisfied. Fix any such "good" equation $p = 0$ in Φ. Note that there is only one equation of the first kind, so that equation is satisfied. Since at least 0.5 fraction of the equations of the second and the third kind are satisfied, we conclude that

$$\Pr_{u,v\in H}\left[\Psi^p_2(u,v) = C_2(u,v)Q(u)Q(v)\right] \geqslant 0.5, \tag{1}$$

and

$$\Pr_{u \in H}\left[\Psi_1^p(u) = C_1(u)Q(u)\right] \geqslant 0.5. \tag{2}$$

Since the polynomials $\Psi_2^p(u,v), \Psi_1^p(u), C_2(u,v), C_1(u), Q(u)$ all have degree at most $2k-2$ in each variable, and $|H| = 10k^2$, by the Schwartz-Zippel lemma, we must have a formal identity

$$\Psi_2^p(u,v) = C_2(u,v)Q(u)Q(v), \quad \Psi_1^p(u) = C_1(u)Q(u).$$

Now, since the equation of the first kind is satisfied, we conclude

$$\begin{aligned}
-c_0 &= \sum_{u,v \in S} \Psi_2^p(u,v) + \sum_{u \in S} \Psi_1^p(u) \\
&= \sum_{u,v \in S} C_2(u,v)Q(u)Q(v) + \sum_{u \in S} C_1(u)Q(u) \\
&= \sum_{i,j=1}^{k} C_2(s_i,s_j)Q(s_i)Q(s_j) + \sum_{i=1}^{k} C_1(s_i)Q(s_i) \\
&= \sum_{i,j=1}^{k} c_{i,j} \cdot Q(s_i)Q(s_j) + \sum_{i=1}^{k} c_i \cdot Q(s_i).
\end{aligned}$$

Thus, the assignment $\sigma : \{x_1, \ldots, x_k\} \to \mathbb{F}$ defined as $\sigma(x_i) = Q(s_i)$ satisfies the equation $p = 0$ in Φ. Since at least 0.55 fraction of the equations $p = 0$ in Φ are "good", it follows that $\mathrm{val}(\Phi) \geqslant 0.55$, a contradiction. □

2.2.3 The choice of the set L in Lemma 2.2

Note that we have some degree of freedom in the choice of the set L, which we discuss below. The choices of the sets $S \subseteq H$ were completely arbitrary as long as $|S| = k$ and $|H| = 10k^2$. The set L contains the elements

$$\sum_{u,v \in S} u^i v^j, \quad \sum_{u \in S} u^i, \quad u^i v^j, \quad u^i, \quad 0, \tag{3}$$

with $u, v \in H$ and $0 \leqslant i, j \leqslant 2k-2$. Depending on whether the characteristic of the field $\mathbb{F}$ is large or small, we choose the set L as below. Let C be a large enough constant chosen as below.

- **Large characteristic:** If $p = \mathrm{char}(\mathbb{F}) \geqslant k^{Ck}$, then we choose $S = \{0, \ldots, k-1\}$, $H = \{0, \ldots, 10k^2 - 1\}$, and let $L = \{0, \ldots, D\} \subseteq \mathbb{F}_p$, where $D = k^{O(k)}$ is large enough so that all the coefficients in (3) are contained in L. Our choice of the constant C will be such that $3k^2 D \leqslant p$.

- **Small characteristic:** If $p = \mathrm{char}(\mathbb{F}) \leqslant k^{Ck}$, then we choose S and $S \subseteq H$ to be arbitrary subsets of $\mathbb{F}$ of size k and $10k^2$ respectively. Then, we choose L to be the linear span, over $\mathbb{F}_p$, of all the, at most $O(k^6)$, elements in (3). Note that in this case, L is closed under addition and $|L| \leqslant k^{O(k^7)}$.

2.3 Reducing arity to constant

In this section, we describe the reduction that starts with an instance of SIMPLE-DEG-2-SAT as in Lemma 2.2 and constructs an instance where the (quadratic) equations have constant arity and the gap is bounded away from 1.

Lemma 2.3. *There is an* FPT*-reduction*

$$\text{SIMPLE-DEG-2-SAT}(\mathbb{F}, k, e, q = k, \varepsilon = 0.95) \leqslant_{\text{FPT}} \text{DEG-2-SAT}(\mathbb{F}, k', e', q' = O(1), \varepsilon' = 0.999),$$

mapping an instance Φ *of* SIMPLE-DEG-2-SAT *to an instance* Φ' *of* DEG-2-SAT *such that:*

- k', e' *depend only on* k, e*.*
- *Each equation depends only on a constant number of variables.*

We sketch the main idea first. Consider the instance Φ of SIMPLE-DEG-2-SAT such that each equation is of the form

$$\ell_1(x) = a \cdot \ell_2(x) \cdot \ell_3(x) + b \cdot \ell_4(x) + c, \tag{4}$$

where $a, b, c \in \mathbb{F}$ and the coefficients of the linear forms $\ell_i(x)$ are in $L \subseteq \mathbb{F}$ as in Lemma 2.2. Note that the linear forms are of the type $\ell(x) = \sum_{i=1}^{k} u_i x_i$ where $u_i \in L$ and $x_1, \ldots, x_k$ are the variables of the instance Φ. Our reduction constructs a new instance Φ' whose variables are intended to be the values of *all* linear forms $\ell(x) = \sum_{i=1}^{k} u_i x_i$ over *all* choices of $u_1, \ldots, u_k \in L$. Alternately, we may think of the variables of Φ' as the entries in the table of values of a function $f : L^k \to \mathbb{F}$, where the intention is that f is a linear function defined as

$$f(u_1, \ldots, u_k) = \sum_{i=1}^{k} u_i \cdot \sigma(x_i),$$

and $\sigma : \{x_1, \ldots, x_k\} \to \mathbb{F}$ is a supposed satisfying assignment to Φ. Now consider a typical equation (4) in Φ. Since the values of the linear forms $\ell_i(x)$ are supposed to appear as variables $f(u^{(i)})$ in Φ', the equation (4) is now a quadratic equation that depends only on 4 variables of Φ', i.e. the new equations have constant arity! However, to ensure the correctness of this reduction, one needs to ensure that the assignment to Φ' (given by an adversary) is indeed a linear function $f : L^k \to \mathbb{F}$, or "close" to being a linear function as we see next.

To ensure that $f : L^k \to \mathbb{F}$ is close to a linear function, we perform a "linearity test" that makes a constant number of queries to the table of f (three queries are enough). The test itself is linear in the queries made by the tester. The tests are then thought of as equations in the values of table f, i.e. the variables of Φ'. Such linearity tests are well-studied. In particular, a three query test is known so that if the test passes with probability close to 1, then the function f agrees with a (unique) linear function g, say on 0.99 fraction of the inputs in L^k. Having ensured that f is close to a linear function g, we are then faced with another issue. Equation (4) involves values of f at *specific* inputs $u \in L^k$ and even though f is close to a linear function g, it might be the case that $f(u) \neq g(u)$ at these specific

inputs that we are interested in. It turns out that there is a "self-correction" procedure, that given a query access to a function f that is close to a linear function g, makes a constant number of queries to f (two queries suffice) and outputs the "correct value" $g(u)$ with high probability. The linearity testing and self-correction procedures were first considered in the paper of Blum, Luby, and Rubinfeld [BLR93].

This describes the main idea behind our reduction. We recall, from Section 2.2.3, that if the field $\mathbb{F}$ has small characteristic, then $L \subseteq \mathbb{F}$ can be taken as an additive subgroup of $\mathbb{F}$. In this case, the linearity testing and the self-correction procedures are already known, e.g. [BLR93, BOCLR08], and can be used directly. However, if the field $\mathbb{F}$ has large characteristic $p \geqslant 3k^2D$, then $L = \{0, 1, \ldots, D\} \subseteq \mathbb{F}_p$ is not closed under addition. In this case, we design new procedures for linearity testing and self-correction that might be of independent interest. These procedures closely mimic the corresponding procedures when L does have an additive group structure, but one main difference is that in addition to the table of $f : L^k \to \mathbb{F}$, the "tester" needs access to additional, auxiliary table $\pi : \Gamma^k \to \mathbb{F}$, where $L \subseteq \Gamma = \{0, 1, \ldots, 3k^2D\} \subseteq \mathbb{F}_p$. The table π is supposed to be the *same* linear function as f, but evaluated over a larger domain Γ^k. We summarize the linearity testing and the self-correction procedures below in Lemma 2.4, prove Lemma 2.3, and then present a proof of Lemma 2.4.

Lemma 2.4. *Let $\mathbb{F}$ be a finite field. Let $L \subseteq \mathbb{F}$ and $L \subseteq \Gamma \subseteq \mathbb{F}$ be such that*

- *Either, L is an additive subgroup of $\mathbb{F}$ and $\Gamma = L$,*
- *Or else, $p = \mathrm{char}(\mathbb{F}) \geqslant 3k^2D$, $L = \{0, 1, \ldots, D\}$, $\Gamma = \{0, 1, \ldots, 3k^2D\}$.*

There is a randomized 3-query test T that gets as input a query access to a function $f : L^k \to \mathbb{F}$ as well as an additional function $\pi : \Gamma^k \to \mathbb{F}$ such that $\pi|_{L^k} = f$, makes 3 queries to (f, π) and has the following guarantee:

- *The test is linear in the 3 queries.*
- *If f is linear, then there exists π such that T accepts with probability 1.*
- *For any $\varepsilon > 0$, if the test T accepts (f, π) with probability at least $1 - \varepsilon$, then f is $(1 - 4\varepsilon)$-close to some linear function $g : L^k \to \mathbb{F}$. Furthermore, there is a self-correcting procedure C that for any input $u \in L^k$ makes 2 queries to (f, π) and outputs $C(u)$ such that*

$$\Pr[C(u) = g(u)] \geqslant 1 - (4\varepsilon + 2/k),$$

where the probability is over the randomness of C. The output $C(u)$ is linear in the 2 queries.

If L has a group structure, the additional function π is not really needed, i.e. all queries are made to f and then the role of π is redundant. Lemma 2.4 is stated so that it conveniently applies to both the cases, when L has a group structure as well as when it doesn't. We now show how Lemma 2.4 implies Lemma 2.3.

Proof of Lemma 2.3. Given an instance Φ of SIMPLE-DEG-2-SAT with variables $x_1, \ldots, x_k$ and equations of the form

$$\ell_1(x) = a \cdot \ell_2(x) \cdot \ell_3(x) + b \cdot \ell_4(x) + c,$$

we construct an instance Φ' as follows. The variables of Φ' will be the table of values of $f : L^k \to \mathbb{F}$ and the table of values of $\pi : \Gamma^k \to \mathbb{F}$ as in Lemma 2.4. In order to describe the equations of Φ', we describe a tester that uses the linearity testing, self-correction primitives as well as the equations of Φ. The equations of Φ' then correspond to the tests on the queries made by the tester. The tester works as follows:

1. With probability 0.5, perform linearity test T on (f, π) as in Lemma 2.4.
2. With probability 0.5, perform the following steps:
   (a) Pick a random equation of Φ of the form
   $$\ell_1(x) = a \cdot \ell_2(x) \cdot \ell_3(x) + b \cdot \ell_4(x) + c.$$
   (b) For each $\ell_j(x)$ let $u^{(j)} \in L^k$ be such that $\ell_j(x) = \sum_{i=1}^k u_i^{(j)} x_i$. Apply the self correcting procedure C in Lemma 2.4 to obtain the value $C(u^{(j)})$.
   (c) Accept if and only if
   $$C(u^{(1)}) = a \cdot C(u^{(2)}) \cdot C(u^{(3)}) + b \cdot C(u^{(4)}) + c.$$

That is, the equations of Φ' of the first type are independent of Φ. The second type of equations do depend on Φ. Specifically, each equation of Φ chosen in step (a) induces a collection of equations of Φ' that come from the self-correcting procedure for each $u^{(j)}$. The equation in step (c) depends on 8 variables of Φ', since each $C(u^{(j)})$ depends linearly on two values of f and π. We now prove the correctness of the reduction.

Yes Case: If $\mathrm{val}(\Phi) = 1$, it is clear that $\mathrm{val}(\Phi') = 1$. Indeed, if $\mathrm{val}(\Phi) = 1$, then there exists an assignment $\sigma(x_1), \ldots, \sigma(x_k) \in \mathbb{F}$ to the variables of Φ that satisfies all the equations, and the corresponding assignment $\sum_{i=1}^k u_i \cdot \sigma(x_i)$ for both $f(u), u \in L^k$ and $\pi(u), u \in \Gamma^k$, will satisfy all equations of Φ'.

NO Case: Now suppose that $\mathrm{val}(\Phi') \geqslant 1 - \varepsilon$ for $\varepsilon = 0.001$. Let $f : L^k \to \mathbb{F}$ and $\pi : \Gamma^k \to \mathbb{F}$ be the assignment to the variables that satisfies $1 - \varepsilon$ of the equations of Φ'. Then, the linearity test accepts (f, π) with probability at least $1 - 2\varepsilon$, and so by Lemma 2.4, there exists a linear function $g : L^k \to \mathbb{F}$ that agrees with f on at least $1 - 8\varepsilon$ fraction of the inputs.

Similarly, (f, π) satisfies at least $1 - 2\varepsilon$ fraction of the equations of the second type. Consider now an equation of Φ and a collection of tests of Φ' of the second type defined by this equation. By an averaging argument, it follows that for $1 - 20\varepsilon$ fraction of the equations of Φ chosen in step (a), the induced tests in step (c) accept with probability at least 0.9. Call such an equation of Φ good. We show that values of $g : L^k \to \mathbb{F}$ (at specific, relevant

inputs), when viewed as assignment to Φ, satisfy every good equation of Φ. Since $1 - 20\varepsilon$ fraction of the equations of Φ are good, but $\mathrm{val}(\Phi) \leqslant 0.95$, this would be a contradiction.

Indeed, fix a good equation of Φ so that the induced test in step (c) accepts with probability at least 0.9. Let $\mathcal{E}$ denote the event that the test accepts. By the "furthermore" part of Lemma 2.4 and using a union bound for $u^{(1)}, \ldots, u^{(4)}$ appearing in the equation in step (c), we get that

$$\Pr\left[C(u^{(i)}) = g(u^{(i)}) \text{ for all } i = 1, \ldots, 4\right] \geqslant 1 - (32\varepsilon + 8/k) \geqslant 0.5.$$

Let $\mathcal{E}'$ denote the event that $C(u^{(i)}) = g(u^{(i)})$ for all $i = 1, \ldots, 4$ so that $\Pr[\mathcal{E}'] \geqslant 0.5$. Thus, with probability at least 0.4, both events $\mathcal{E}$ and $\mathcal{E}'$ occur, which is same as saying that the values $g(u^{(i)})$ satisfy the (good) equation. This completes the proof of Lemma 2.3.

□

2.4 Proof of Lemma 2.4 - Linearity-Testing and Self-Correcting

In this section, we prove Lemma 2.4. As we mentioned, when L has a group structure, the lemma is well-known, e.g. in [BLR93, BOCLR08]. Therefore, we prove the lemma only for the case when $p = \mathrm{char}(\mathbb{F})$ is large and the set L is of the form $\{0, 1 \ldots, D\}$ for some $D \ll p$. The proof follows the outline from [BOCLR08] with appropriate modifications to our setting. After presenting the proof, we also point out, for the benefit of non-expert readers, how the proof works when L does have a group structure.

We recall that $p = \mathrm{char}(\mathbb{F}) \geqslant 3k^2D$, $L = \{0, 1, \ldots, D\}$, and $\Gamma = \{0, 1, \ldots, 3k^2D\}$. The tester is given query access to function $f : L^k \to \mathbb{F}$ and to $\pi : \Gamma^k \to \mathbb{F}$ such that $\pi|_{L^k} = f$. Since the restriction of π to L^k coincides with f, in the following, we denote both f and π by the same function f, keeping in mind that the "actual" function f is the restriction to L^k. The tester T works as follows:

1. With probability 0.5, perform the following test T_1.
   (a) Pick $x \in \{0, 1, \ldots, D\}^k$, $y \in \{0, 1, \ldots, k^2D\}^k$ independently, uniformly at random.
   (b) Accept if and only if $f(x) + f(y) = f(x + y)$.
2. With probability 0.5, perform the following test T_2.
   (a) Pick $x, y \in \{0, 1, \ldots, k^2D\}^k$ independently, uniformly at random.
   (b) Accept if and only if $f(x) + f(y) = f(x + y)$.

That is, we perform the standard linearity testing as in [BLR93]. However, since the domain of f does not have a group structure, we need to choose the distribution from which we choose x and y carefully.

Clearly, if f is linear, i.e. $f(u_1, \ldots, u_k) = \sum_{i=1}^{k} \sigma_i u_i$, $\sigma_i \in \mathbb{F}$, then T always accepts. Towards proving the soundness property, suppose now that T accepts f with probability

$1-\varepsilon$. Note that this implies that both T_1 and T_2 accept with probability at least $1-2\varepsilon$ each. Our goal is to prove that the restriction of f to $\{0,1,\ldots,D\}^k$ is close to a linear function. Towards this goal, we define the following function $g:\{0,1,\ldots,D\}^k \to \mathbb{F}$,

$$g(x) = \mathsf{Plurality}_y(f(x+y)-f(y)),$$

where the plurality is taken over a uniformly random $y \in \{0,1,\ldots,k^2D\}^k$. To clarify, the plurality refers to the element in $\mathbb{F}$ that occurs most frequently as the value $f(x+y)-f(y)$. A tie is broken arbitrarily, but we show next, that the plurality is in fact always an overwhelming majority.

Claim 2.5. *For each $x \in \{0,1,\ldots,D\}^k$, let*

$$P_x = \Pr_y[g(x) = f(x+y)-f(y)],$$

where y is chosen uniformly at random from $\{0,1,\ldots,k^2D\}^k$. Then, $P_x \geqslant 1-(4\varepsilon+2/k)$ for every $x \in \{0,1,\ldots,D\}^k$.

Proof. Let $A_x = \Pr_{y,z}[f(x+y)-f(y) = f(x+z)-f(z)]$, where y,z are chosen independently and uniformly from $\{0,1,\ldots,k^2D\}^k$. Note first that

$$A_x \leqslant P_x. \tag{5}$$

Indeed,

$$\begin{aligned}
A_x &= \sum_{u\in\mathbb{F}} \Pr_{y,z}[f(x+y)-f(y)=u=f(x+z)-f(z)] \\
&= \sum_{u\in\mathbb{F}} \Pr_{y}[f(x+y)-f(y)=u]^2 \\
&\leqslant \max_{u\in\mathbb{F}}\left(\Pr_y[f(x+y)-f(y)=u]\right)\cdot\left(\sum_{u\in\mathbb{F}}\Pr_y[f(x+y)-f(y)=u]\right) \\
&= P_x,
\end{aligned}$$

which proves (5). On the other hand, we have

$$\begin{aligned}
1-A_x &= \Pr_{y,z}[f(x+y)+f(z)\neq f(x+z)+f(y)] \\
&\leqslant \Pr_{y,z}[f(x+y)+f(z)\neq f(x+y+z)] + \Pr_{y,z}[f(x+z)+f(y)\neq f(x+y+z)] \\
&= 2\cdot\Pr_{y,z}[f(x+y)+f(z)\neq f(x+y+z)].
\end{aligned}$$

Note that the quantity $\Pr_{y,z}[f(x+y)+f(z)\neq f(x+y+z)]$ is equal to $\Pr_{y',z}[f(y')+f(z)\neq f(y'+z)]$, where y' is chosen in the domain $\{0,1,\ldots,k^2D\}^k$ "shifted by x". For this distribution on y' we have $\Pr_{y'}\left[y'\in\{0,1,\ldots,k^2D\}^k\right] \geqslant (1-1/k^2)^k \geqslant 1-1/k$. That is, the

distribution of y' is close to the distribution of a query in T_2 and the distribution of z is the same as in T_2. Thus, since T_2 accepts f with probability at least $1-2\varepsilon$, it follows that

$$\Pr_{y,z}[f(x+y)+f(z)\neq f(x+y+z)] = \Pr_{y',z}[f(y')+f(z)\neq f(y'+z)] \leqslant 2\varepsilon + 1/k,$$

and hence

$$A_x \geqslant 1-(4\varepsilon+2/k). \tag{6}$$

Combining (5) with (6) we get that

$$P_x \geqslant 1-(4\varepsilon+2/k),$$

as required. □

Claim 2.6. *Suppose that $k \geqslant 20$ and $\varepsilon \leqslant 0.02$. Then $\Pr_{x\in\{0,1,\ldots,D\}^k}[f(x)\neq g(x)] \leqslant 4\varepsilon$.*

Proof. Note that if we choose $x \in \{0,1,\ldots,D\}^k$ and $y \in \{0,1,\ldots,k^2D\}^k$ according to the distribution of T_1 then

$$\begin{aligned}
2\varepsilon &\geqslant \Pr_{x,y}[T_1 \text{ rejects}] \\
&\geqslant \Pr_{x,y}[f(x)\neq f(x+y)-f(y) | f(x)\neq g(x)]\cdot \Pr[f(x)\neq g(x)] \\
&\geqslant \Pr_{x,y}[g(x)= f(x+y)-f(y) | f(x)\neq g(x)]\cdot \Pr[f(x)\neq g(x)] \\
&\geqslant \min_{x\in\{0,1,\ldots,D\}^k}(P_x)\cdot \Pr[f(x)\neq g(x)] \\
&\geqslant (1-(4\varepsilon+2/k))\cdot \Pr[f(x)\neq g(x)].
\end{aligned}$$

Therefore, if k is sufficiently large and ε is sufficiently small, then $\Pr_{x\in\{0,1,\ldots,D\}^k}[f(x)\neq g(x)] \leqslant 4\varepsilon$ and the claim follows. □

Claim 2.7. *Suppose that $k \geqslant 20$ and $\varepsilon \leqslant 0.02$. Then, the restriction of g to $\{0,1,\ldots,D\}^k$ is a linear function.*

Proof. In order to prove that g is linear, it is enough to show that for every $x \in \{0,\ldots,D\}^k$ and for every $i \in [k]$ it holds that $g(x)+g(e_i) = g(x+e_i)$, where $e_i \in \mathbb{F}^k$ is the vector with 1 in the i^{th} coordinate and 0 everywhere else. By Claim 2.5, we can write down the following three inequalities. In the first inequality, e_i+y is just a proxy for y and when $y \in \{0,1,\ldots,k^2D\}$ is uniformly chosen, the distribution of e_i+y is $\frac{1}{k^2}$-close to that of y. The extra $\frac{1}{k^2}$ on the R.H.S. of the first inequality accounts for this small difference.

$$\begin{aligned}
\Pr_y[g(x)= f(x+(e_i+y))-f(e_i+y)] &\geqslant 1-(4\varepsilon+2/k+1/k^2) \\
\Pr_y[g(e_i)= f(e_i+y)-f(y)] &\geqslant 1-(4\varepsilon+2/k) \\
\Pr_y[g(x+e_i)= f(x+e_i+y)-f(y)] &\geqslant 1-(4\varepsilon+2/k)
\end{aligned}$$

Therefore, for ε sufficiently small and k sufficiently large, by the union bound, all three events hold for the same $y \in \{0,1,\ldots,k^2D\}^k$, and thus

$$g(x)+g(e_i) = \Big(f(x+e_i+y)-f(e_i+y)\Big)+\Big(f(e_i+y)-f(y)\Big) = f(x+e_i+y)-f(y) = g(x+e_i).$$

Therefore, the restriction of g to $\{0,1,\ldots,D\}^k$ is a linear function, as required. □

By combining Claim 2.6 with Claim 2.7 we conclude that if T accepts f with probability $1-\varepsilon$, then the restriction of f to $\{0,1,\ldots,D\}^k$ is 4ε-close to a linear function $g(x) = \sum_{i=1}^{k} g(e_i)x_i$. The self-correcting procedure is straightforward: on input $x \in \{0,1,\ldots,D\}^k$,

- Pick $y \in \{0,1,\ldots,k^2D\}^k$ uniformly at random.
- Output $C(x) = f(x+y) - f(y)$.

Clearly, the procedure makes 2 queries to f and by Claim 2.5, it follows that $\Pr[C(x) = g(x)] \geqslant 1-(4\varepsilon+2/k)$. This completes the "furthermore" part of Lemma 2.4.

Finally, we comment on how the testing and self-correction works when L has a group structure. In this case, the tester is given query access to $f : L^k \to \mathbb{F}$ and there is no additional function π. The tester tests whether $f(x)+f(y) = f(x+y)$ for uniformly random x and y. A similar proof as above shows that if the tester accepts with probability $1-\varepsilon$, then f is $(1-O(\varepsilon))$-close to a linear function g. Moreover, for every fixed $x \in L^k$, $g(x) = f(x+y)-f(y)$ for $(1-O(\varepsilon))$ fraction of $y \in L^k$ and this serves as the self-correction procedure.

References

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

[AS98] S. Arora and S. Safra. Probabilistic Checking of Proofs: A New Characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[BLR93] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993.

[BOCLR08] M. Ben-Or, D. Coppersmith, M. Luby, and R. Rubinfeld. Non-abelian homomorphism testing, and distributions close to their self-convolutions. *Random Struct. Algorithms*, 32(1):49–70, 2008.

[CGG06] Y. Chen, M. Grohe, and M. Grber. On parameterized approximability. In *Parameterized and Exact Computation*, volume 4169 of *Lecture Notes in Computer Science*, pages 109–120. Springer Berlin Heidelberg, 2006.

[DECF+03] R. G. Downey, V. Estivill-Castro, M. Fellows, E. Prieto, and F.A. Rosamund. Cutting up is hard to do: The parameterised complexity of k-cut and related problems. *Electronic Notes in Theoretical Computer Science*, 78:209 – 222, 2003.

[DF95a] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness I: basic results. *SIAM J. Comput.*, 24(4):873–921, 1995.

[DF95b] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness II: on completeness for W[1]. *Theor. Comput. Sci.*, 141(1&2):109–131, 1995.

[DF99] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[FG06] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer-Verlag, 2006.

[FGL+96] U. Feige, S. Goldwasser, L. Lovasz, S. Safra, and M. Szegedy. Approximating Clique is almost NP-complete. *Journal of the ACM*, 43:268–292, 1996.

[Laz79] D. Lazard. Systems of algebraic equations. *Symbolic and Algebraic Computation*, 72:88–94, 1979.

[LFKN92] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, October 1992.

[Mar08] D. Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51:60–78, 2008.

[Sha92] A. Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

[vzGG03] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de