Almost Optimal Time Lower Bound for Approximating Parameterized Clique, CSP, and More, under ETH

Venkatesan Guruswami[*]   Bingkai Lin[†]   Xuandi Ren[‡]   Yican Sun[§]

Kewen Wu[¶]

Abstract

The Parameterized Inapproximability Hypothesis (PIH), which is an analog of the PCP theorem in parameterized complexity, asserts that, there is a constant $\varepsilon > 0$ such that for any computable function $f : \mathbb{N} \to \mathbb{N}$, no $f(k) \cdot n^{O(1)}$-time algorithm can, on input a k-variable CSP instance with domain size n, find an assignment satisfying $1 - \varepsilon$ fraction of the constraints. A recent work by Guruswami, Lin, Ren, Sun, and Wu (STOC'24) established PIH under the Exponential Time Hypothesis (ETH).

In this work, we improve the quantitative aspects of PIH and prove (under ETH) that approximating sparse parameterized CSPs within a constant factor requires $n^{k^{1-o(1)}}$ time. This immediately implies that, assuming ETH, finding a $(k/2)$-clique in an n-vertex graph with a k-clique requires $n^{k^{1-o(1)}}$ time. We also prove almost optimal time lower bounds for approximating k-ExactCover and Max k-Coverage.

Our proof follows the blueprint of the previous work to identify a "vector-structured" ETH-hard CSP whose satisfiability can be checked via an appropriate form of "parallel" PCP. Using further ideas in the reduction, we guarantee additional structures for constraints in the CSP. We then leverage this to design a parallel PCP of almost linear size based on Reed-Muller codes and derandomized low degree testing.

Contents

1 Introduction

One of the goals of complexity theory is to pinpoint the asymptotically optimal time (or other resource) needed to solve basic computational problems or classes of problems. The theory of NP-completeness attacks this at a coarse level, but modern complexity theory also has tools to give more fine-grained information on computational complexity.

A common setting for fine-grained understanding of computational hardness is considering *parameterized problems*. Under this setting, each instance is attached with an additional parameter k indicating some specific quantities (e.g., the optimum or the treewidth). We treat k as some super constant that is much smaller than the instance size n and consider the existence or absence of algorithms with running time depends both on n and k (e.g., with running time $2^{2^k}n^{O(1)}$, or $n^{\sqrt{k}}$). The hardness of parameterized problems is studied under the realm of *parameterized complexity theory* [FG06]. It is a central challenge to figure out the minimal time (depending on both n and k) to solve prototypical parameterized problems.

A representative example is the k-CLIQUE problem parameterized by the optimum k, which is one of the most fundamental problems in parameterized complexity theory: given an n-vertex graph as input, determine if it has a clique of size k. The naive brute force algorithm takes roughly n^k time. Using fast matrix multiplication, there are better algorithms that take $n^{\omega k/3}$ time, where ω is the matrix multiplication exponent. On the hardness side, it is known that no algorithm can decide k-CLIQUE within running time $f(k)n^{o(k)}$, for any computable function $f(k)$, under the widely-considered Exponential Time Hypothesis (ETH) [CHKX06], which states that algorithms cannot solve 3SAT formulas on n variables within running time $2^{o(n)}$. The optimal running time for k-clique is therefore pinpointed to be $n^{\Omega(k)}$, assuming ETH.

Can one design a faster algorithm if one settles for *approximating k-CLIQUE*? For example, what if we only want to find a clique of size $k/2$ in a graph that is promised to have a k-clique.[1]

It was shown that such an approximation to k-CLIQUE still requires the tightest $f(k)n^{\Omega(k)}$ time [CCK+17], under the very strong assumption Gap-ETH. The Gap-ETH postulates an exponential time lower bound of approximating MAX 3-SAT within a constant ratio. The constant gap baked into the assumption is then transformed into a constant gap for approximating k-CLIQUE. Though Gap-ETH has been proved under particular strengthening of ETH (smooth version of ETH) [App17], a theoretically more satisfactory result is to obtain the hardness of approximating k-CLIQUE under the original ETH. Under ETH, weaker time lower bounds were known for constant-factor approximations of k-CLIQUE: $f(k)n^{\Omega\left(\sqrt[6]{\log k}\right)}$ in [Lin21], which was later improved to $f(k)n^{\Omega(\log k)}$ in [LRSW22, CFLL23] and $f(k)n^{k^{\Omega(1/\log\log k)}}$ in [LRSW23].[2] However, all these approaches cannot obtain lower bounds better than $f(k)n^{\Omega(\sqrt{k})}$ due to the coding theoretic barriers [KT00].

However, this paper significantly improves this lower bound, assuming only the original ETH.

Theorem 1.1. *Assume ETH. For any constant $\varepsilon > 0$ and any computable function $f(k)$, any algorithm that approximates k-CLIQUE within an ε ratio must take runtime $f(k)n^{k^{1-o(1)}}$.*

To prove the theorem above, we follow a similar idea of proving the NP-hardness of approximating cliques, which relies on the NP-hardness of constant approximating CSP (a.k.a., the PCP

---

[1]This can also be formulated as a "gap" decision problem of distinguishing graphs with a k-clique from those which do not even have a $(k/2)$-clique.

[2]There is another line of work on improving the inapproximability factor of k-CLIQUE under the minimal hypothesis W[1]$\neq$FPT: constant factor in [Lin21], and the improved $k^{o(1)}$ in [KK22, CFLL23].

theorem) and a subsequent FGLSS reduction [FGL+96]. In the parameterized setting, we can apply an analogous reduction.

- We first establish a near-optimal lower bound for approximating (sparse) "parameterized CSPs" with k variables, $O(k)$ constraints, an alphabet of size n, and some constant inapproximability factor.
- Then Theorem 1.1 follows immediately by the FGLSS reduction [FGL+96] and an expander-based gap-amplification procedure [AFWZ95, LRSW22].

The first step is equivalent to establishing a quantitative version of the Parameterized Inapproximability Hypothesis (PIH) [LRSZ20], which plays the role of the PCP theorem in the parameterized complexity theory. The first quantitative version of PIH was established under Gap-ETH [CCK+17], with the lower bound $f(k)n^{\Omega(\log k)}$. A recent improvement proved PIH under ETH [GLR+23], with a weaker lower bound $f(k)n^{\Omega(\sqrt{\log \log k})}$. However, both results are too weak to establish Theorem 1.1. In this paper, we make a significant quantitative improvement to the reduction in [GLR+23], obtaining our main theorem stated below.

Theorem 1.2 (Informal version of Theorem 4.1). *Assume ETH. For some constant $\varepsilon \in (0,1)$, for any computable function $f(k)$, every algorithm that takes as input a satisfiable parameterized 2CSP instance with k variables, $O(k)$ constraints and size-n alphabet finds an assignment satisfying $1 - \varepsilon$ fraction of the constraints, must take $f(k)n^{k^{1-o(1)}}$ time.*

Combining with the parallel repetition in projection games [Rao11], we can immediately boost the soundness to any constant $\varepsilon > 0$ with lower bound $f(k)n^{k^{\Omega(1/\log(1/\varepsilon))}}$.

Corollary 1.3. *Assume ETH. For any computable function $f(k)$ and any constant $\varepsilon \in (0,1)$, every algorithm that takes as input a satisfiable parameterized 2CSP instance with k variables, $O(k)$ constraints and size-n alphabet finds an assignment satisfying $1 - \varepsilon$ fraction of the constraints, must take $f(k)n^{k^{\Omega(1/\log(1/\varepsilon))}}$ time.*

By the discussion above, Theorem 1.1 follows immediately by combining Theorem 1.2, the FGLSS reduction, and the expander-based gap-amplification procedure [AFWZ95, LRSW22]. Moreover, using Theorem 1.2 as the foundation, we can obtain, via reductions, strong inapproximability results for other fundamental parameterized problems. Below, we list two application highlights and refer interested readers to [GLR+23] for more detailed discussions.

Application Highlight: k-EXACTCOVER. k-EXACTCOVER (a.k.a., k-UNIQUESETCOVER) is one of the canonical problems in the parameterized world. It is a weaker version of the k-SETCOVER problem. For the ρ-approximation version of k-EXACTCOVER with $\rho \geq 1$, denoted by $(k, \rho \cdot k)$-EXACTCOVER, the instance consists of a universe U and a collection $\mathcal{S}$ of subsets of U, with a goal to distinguish the following two cases.

- There exists k *disjoint* subsets from $\mathcal{S}$ whose union equals the whole universe U.
- The union of any $\rho \cdot k$ subsets of $\mathcal{S}$ is a proper subset of U.

Here, the parameter is the optimum k. We remark that the additional disjointness requirement in the completeness part makes $(k, \rho \cdot k)$-EXACTCOVER an excellent intermediate problem for proving the hardness of other problems [ABSS97, Man20].

On the algorithmic side, the $(k, \rho \cdot k)$-EXACTCOVER has a brute-force $|\mathcal{S}|^{\Omega(k)}$-time algorithm. However, no $|\mathcal{S}|^{o(k)}$-time algorithms are known. Thus, it is natural to consider whether we can establish a matching $|\mathcal{S}|^{\Omega(k)}$ lower bound. Our work almost establishes a lower bound for $(k, \rho \cdot k)$-EXACTCOVER, under ETH, for some constant ρ. Previously, this was only known under the Gap-ETH assumption [Man20].

Theorem 1.4. *Assume ETH. There exists some constant $\rho \geq 1$, such that for any computable function $f(k)$, any algorithm deciding $(k, \rho \cdot k)$-EXACTCOVER must take runtime $f(k)|\mathcal{S}|^{k^{1-o(1)}}$.*

To prove the theorem above, We note that the previous work [GRS23] achieves a parameter-preserving reduction from PIH to $(k, \rho \cdot k)$-EXACTCOVER for any constant ρ, by imitating the beautiful reduction of Feige [Fei98].[3]

Therefore, Theorem 1.4 follows by combining the reduction of [GRS23] with our Theorem 1.2. Applying Corollary 1.3, we can boost the ratio ρ to any constant with lower bound $f(k)|\mathcal{S}|^{k^{\Omega(1/\log(\rho))}}$.

Proposition 1.5. *Assume ETH. For any constant $\rho \geq 1$ and computable function $f(k)$, any algorithm deciding $(k, \rho \cdot k)$-EXACTCOVER must take runtime $f(k)|\mathcal{S}|^{k^{\Omega(1/\log(\rho))}}$.*

Application Highlight: MAX k-COVERAGE. The MAX k-COVERAGE is the maximization variant of the k-SETCOVER problem. For the ρ-approximation version of MAX k-COVERAGE with $\rho < 1$, denoted by MAX (ρ, k)-COVERAGE, the instances are the same as k-EXACTCOVER above, but the goal changes to distinguish the following two case:

- There exists k subsets from $\mathcal{S}$ whose union equals $\mathcal{U}$.
- Any k subsets from $\mathcal{S}$ has the union size at most $\rho \cdot |\mathcal{U}|$.

MAX (ρ, k)-COVERAGE has been widely studied in previous literature. There exists a simple greedy algorithm solving MAX $(1 - \frac{1}{e}, k)$-COVERAGE within polynomial runtime [Hoc97].

On the hardness side, a celebrated result of Feige [Fei98] showed the NP-hardness of MAX $(1 - \frac{1}{e} + \varepsilon, k)$-COVERAGE for any $\varepsilon > 0$, thus proving a tight inapproximability result.

In the parameterized world, one can solve MAX k-COVERAGE in $|\mathcal{S}|^k$ time by brute force enumeration. On the other hand, Cohen-Addad, Gupta, Kumar, Lee, and Li [CAGK+19] showed that assuming Gap-ETH, MAX $(1 - \frac{1}{e} + \varepsilon, k)$-COVERAGE requires $f(k)|\mathcal{S}|^{k^{\text{poly}(1/\varepsilon)}}$ runtime. Manurangsi [Man20] further improved this lower bound to the tightest $f(k)|\mathcal{S}|^{\Omega(k)}$ under Gap-ETH.

Our work implies an almost-optimal time lower bound for MAX (ρ, k)-COVERAGE under ETH for some constant ρ.

Theorem 1.6. *Assume ETH. There exists some constant $\rho \in (0, 1)$, such that for any computable function $f(k)$, any algorithm deciding MAX (ρ, k)-COVERAGE must take runtime $f(k)|\mathcal{S}|^{k^{1-o(1)}}$.*

Theorem 1.6 follows from our Theorem 1.2 and the analysis in [Man20, Sections 9.1 and 9.2] that accomplishes a gap-preserving reduction from k-CSP to MAX k-COVERAGE. Applying Corollary 1.3, we can boost the ratio to the tightest $1 - \frac{1}{e} + \varepsilon$ with lower bound $f(k)|\mathcal{S}|^{k^{\varepsilon'}}$.

Proposition 1.7. *Assume ETH. for any constant $\varepsilon \in (0, 1)$ and computable function $f(k)$, there exists a constant $\varepsilon' = \varepsilon'(\varepsilon)$, any algorithm deciding MAX $(1 - \frac{1}{e} + \varepsilon, k)$-COVERAGE must take runtime $f(k)|\mathcal{S}|^{k^{\varepsilon'}}$.*

---

[3]In fact, a weaker variant of PIH, named Average Baby PIH over rectangular constraints, suffices.

New PCP Characterizations. An interesting byproduct of Theorem 1.2 is a new PCP theorem for 3SAT as follows.

Theorem 1.8 (Informal Version of Theorem 4.2). *For any parameter $k \ll n$, 3SAT has a constant-query PCP verifier with alphabet size $|\Sigma| = 2^{n/k^{1-o(1)}}$, runtime $\mathrm{poly}(|\Sigma|, n)$, and $\log k + O\left(\sqrt{\log k} \log \log k\right)$ random coins, which has perfect completeness and soundness $\frac{1}{2}$.*

Theorem 1.8 generalizes the classic PCP theorem and gives a smooth trade-off between the proof length and the alphabet size, connecting parameterized complexity and the classical complexity theory.

Paper Organization. In Section 2, we provide an overview for our proof of Theorem 1.2 and discuss future works. In Section 3, we formalize necessary notation and concepts. The formal structure for proving Theorem 1.2 is presented in Section 4, with most technical statements deferred to other sections: the reduction from ETH-hard problems to special vector-valued CSPs is provided in Section 5, and the PCPP verifier for a helper language is constructed in Section 6. In Appendix A, we give details of derandomized parallel low degree tests which is a key component for the PCPP construction.

2 Technical Overview

In this part, we provide general ideas of proving Theorem 1.2. Due to the equivalence between the existence of PCP systems and the inapproximability of constraint satisfaction problems [Din07, AB09], Theorem 1.8 directly follows from Theorem 1.2.

We follow the spirit of [GLR+23] to prove Theorem 1.2. The proof framework is as follows.

- First, we reduce an ETH-hard problem to a CSP problem with specific structures.
- Then, leveraging the special structures, we construct a probabilistic checkable proof of proximity (PCPP) verifier to check whether the encoding of some given solution satisfies all constraints. Theorem 1.2 follows by converting the PCP verifier into CSP instances [AB09, Din07].

For the first step, [GLR+23] reduces 3SAT to vector-valued CSPs (VecCSPs for short), whose variables take values in a vector space $\mathbb{F}_4^t$. In addition, each constraint is either a coordinate-wise parallel constraint or a linear constraint.

- A parallel constraint (over variables x and y) is defined by a sub-constraint $\Pi^{sub} : \mathbb{F}_4 \times \mathbb{F}_4 \to \{0, 1\}$ and a subset of coordinate $Q \subseteq [t]$. It checks whether $\Pi^{sub}(x_i, y_i) = 1$ for every coordinate $i \in Q$.
- A linear constraint enforces that two vector-valued variables satisfy a linear equation specified by a matrix $M \in \mathbb{F}_4^{t \times t}$, i.e., $y = Mx$.

Then, in the second step, [GLR+23] encodes the solution by parallel Walsh-Hadamard code and constructs a PCPP verifier with double-exponential proof length, resulting in an $f(k)n^{\Omega(\sqrt{\log \log k})}$ lower bound for ε-Gap k-Variable CSP.

6

2.1 More Refined Vector Structure

Unfortunately, the vector structure used in [GLR$^+$23] is far from being enough for obtaining an almost-optimal lower bound due to the following reasons.

- For parallel constraints, VecCSP sets up the sub-constraints over *a subset of the coordinates*. There might be $2^{|E_\mathsf{p}|}$ (where $|E_\mathsf{p}|$ is the number of parallel constraints) different sub-CSP instances over all coordinates, each of which requires an individual PCPP verifier. To simultaneously check the satisfiability of all these sub-instances, they tuple these verifiers into a giant verifier, resulting in an exponential blowup of the proof length. Hence, an almost linear proof size is impossible.

- For linear constraints, VecCSP defined in [GLR$^+$23] allow them to be over arbitrary pairs of variables. They introduce auxiliary variables for all pairs of variables and their corresponding linear constraints to check such unstructured constraints. The number of auxiliary variables is $|V| \cdot |E_\mathsf{l}|$, where $|V|, |E_\mathsf{l}|$ are the numbers of variables and linear constraints, respectively. This means that the proof length is at least quadratic, making an almost linear proof size impossible.

- Furthermore, the VecCSP instance in [GLR$^+$23] has parameters $|V| = O(k^2), |E| = O(k^2)$, which is a starting point with already a quantitative loss for any subsequent constructions.

The analysis above urges us to mine more vector structures and devise new reductions with smaller parameter blowups. In this work, we further engineer VecCSP and obtain special vector-valued CSPs (SVecCSP for short) with three more features.

- First, SVecCSP partitions the variables into two disjoint parts $\{x_1, \ldots x_k\} \dot\cup \{y_1, \ldots y_k\}$.

- Second, for parallel constraints, SVecCSP sets up the sub-constraint on *all* coordinates, which implies a unified sub-CSP instance among all coordinates. As a result, we avoid the tuple procedure, enabling a highly succinct proof.

- Third, for linear constraints, SVecCSP only sets up linear constraints over x_i and y_i, with the same index i. After encoding x and y by the parallel Reed-Muller code, we can leverage this alignment and introduce auxiliary proofs, which is also a codeword of the parallel Reed-Mulle code, to check the validity of linear constraints efficiently, with an almost-linear blowup.

In addition, to decrease the parameter blowup, we apply the reduction in [Mar10, KMPS23] to obtain ETH-hard *sparse* parameterized CSP instances. Then, we imitate the reduction in [GLR$^+$23] to obtain a sparse VecCSP instance with $|V| = O(k), |E| = O(k)$, which ultimately reduces to SVecCSP, with an almost optimal lower bound, by properly duplicating variables and relocating constraints.

2.2 Applying The Parallel Reed-Muller Code

After establishing the almost optimal runtime lower bound for SVecCSP, we need to design a PCPP verifier certifying the encoding of a given assignment satisfies all constraints. However, the previous work [GLR$^+$23] designs a PCPP verifier that requires a proof with a double-exponential length, which is far from obtaining Theorem 1.2.

This paper applies the parallel Reed-Muller (RM) encoding with an almost-linear codeword length. Recall that the variables in the SVecCSP instance are divided into two disjoint parts

7

$\{x_1, \ldots x_k\} \dot\cup \{y_1, \ldots y_k\}$, the input proof of the PCPP verifier consists of $\widehat{x}$ and $\widehat{y}$, which are supposed to be the parallel RM encoding of the assignment over x and y respectively. The verification procedure is demonstrated as follows.

First, to ensure that $\widehat{x}$ and $\widehat{y}$ are indeed codewords of the parallel RM code, we apply parallel low degree testing. The standard low degree testing causes a quadratic blowup in the proof length. To ensure an almost-linear proof length, we use the parallel version of the *derandomized low degree testing* [BSSVW03] (Appendix A).

Second, we check whether $\widehat{x} \circ \widehat{y}$ satisfies all parallel constraints. Since a unified sub-CSP instance exists among all coordinates, as long as we have constructed a PCPP verifier for this sub-CSP, we can simulate it in parallel to check all coordinates at the same time, with no blowup in the proof length (as opposed to an exponential blowup in [GLR$^+$23]). Our sub-CSP has an alphabet of constant size. Thus, we can apply the existing approach (see, e.g., [BGH$^+$06, Theorem 3.3]) in a black box way to construct an almost-linear PCPP verifier for all parallel constraints in our sub-CSP.

Finally, we check whether $\widehat{x} \circ \widehat{y}$ satisfies all linear constraints. Recall that SVecCSP only sets up linear constraints over x_i and y_i, with the same index i. Denote M_i as the matrix for the index i, i.e., the i-th linear constraint is $y_i = M_i x_i$. We introduce an auxiliary proof $\widehat{z}$, satisfying

$$\widehat{z} = \widehat{y} - \widehat{M}\widehat{x}$$

where $\widehat{M}$ is the parallel RM encoding for matrices $\{M_1, M_2, \ldots, M_k\}$. Then, all systematic parts of $\widehat{z}$, i.e., the codeword entries corresponding to $z_i := y_i - M_i x_i$ for $i \in [k]$, should be $\vec{0}$. The key observation is that if $\widehat{x}$ and $\widehat{y}$ are codewords of parallel RM code, then so does $\widehat{z}$. Hence, we can apply parallel derandomized low degree testing for $\widehat{z}$ and apply another PCPP, as in the parallel part, to check whether all systematic parts of $\widehat{z}$ are $\vec{0}$. Finally, we check whether $\widehat{z}$ satisfies the equation above by simply querying a random index, for which the soundness and completeness are guaranteed by Schwartz-Zippel lemma [Sch80]. In this way, we have a highly efficient PCPP verifier that checks all linear constraints with an almost linear proof length.

The overall PCPP verifier is the combination of the verifiers for parallel constraints and for linear constraints. A more detailed framework of our proof is in Section 4.

2.3 Future Works

Our work gives almost optimal time lower bounds for the approximation version of many canonical parameterized problems under ETH, including k-CLIQUE, k-EXACTCOVER, k-VARIABLE CSPs, and MAX k-COVERAGE.

Technically, we prove the almost optimal time lower bound for constant-gap k-Variable CSPs. Using this result as the cornerstone, we obtain the inapproximability for other problems by existing reductions. One open question is whether almost optimal time lower bounds for other problems also follow from this result, e.g., k-BALANCED BICLIQUE [CCK$^+$17, FSLM20].

The second question is to obtain the (actual) optimal $f(k)n^{\Omega(k)}$ time lower bounds for constant-gap k-Variable CSPs. This problem can be seen as the parameterized extension of the long-standing linear-size PCP conjecture [BEKP13]. In the non-parameterized world, the state-of-the-art PCP theorem with the shortest proof length is due to [Din07] with quasilinear proof length. It is also interesting if this optimal bound (i.e., the parameterized extension of the linear-size PCP conjecture) can be established assuming the existence of linear-size PCP.

For more interesting directions, please refer to [GLR$^+$23].

3 Preliminaries

For a positive integer n, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We use log to denote the logarithm with base 2. For a prime power $q = p^c$ where p is a prime and $c \geq 1$ is an integer, we use $\mathbb{F}_q$ to denote the finite field of order p^c and characteristic $\text{char}(\mathbb{F}) = p$.

For an event $\mathcal{E}$, $\mathbb{1}_{\mathcal{E}}$ is defined the indicator function, which equals 1 if $\mathcal{E}$ happens and 0 otherwise. For a finite set $S \neq \emptyset$, we use $x \sim S$ to denote a uniformly random element from S.

For disjoint sets S and T, we use $S \dot{\cup} T$ to denote their union while emphasizing $S \cap T = \emptyset$.

Asymptotics. Throughout the paper, we use $O(\cdot), \Theta(\cdot), \Omega(\cdot)$ to hide absolute constants that do not depend on any other parameter. We also use $\text{poly}(\cdot)$ to denote some implicit polynomial in terms of the parameters within, e.g., $\text{poly}(f, g)$ is upper bounded by $(f^2 + g^2 + C)^C$ for some absolute constant $C \geq 0$.

3.1 Constraint Satisfaction Problem

In this paper, we only focus on constraint satisfaction problems (CSPs) of arity two. Formally, a CSP instance G is a quadruple $(V, E, \Sigma, \{\Pi_e\}_{e \in E})$, where:

- V is for the set of variables.

- E is for the set of constraints. Each constraint $e = \{u_e, v_e\} \in E$ connects two distinct variables $u_e, v_e \in V$.

  The *constraint graph* is the undirected graph on vertices V with edges E. Note that we allow multiple constraints between the same pair of variables; thus, the constraint graph may have parallel edges.

- Σ is for the alphabet of each variable in V. For convenience, we sometimes have different alphabets for different variables, and we will view them as a subset of a grand alphabet with some natural embedding.

- $\{\Pi_e\}_{e \in E}$ is the set of constraint validity functions. Given a constraint $e \in E$, the function $\Pi_e \colon \Sigma \times \Sigma \to \{0, 1\}$ checks whether the constraint e between u_e and v_e is satisfied.

We use $|G| = (|V| + |E|) \cdot |\Sigma|$ to denote the *size* of a CSP instance G.

Assignment and Satisfiability Value. An *assignment* is a function $\sigma \colon V \to \Sigma$ that assigns each variable a value in the alphabet. We use

$$\text{val}(G, \sigma) = \frac{1}{|E|} \sum_{e \in E} \Pi_e(\sigma(u_e), \sigma(v_e))$$

to denote the *satisfiability value* for an assignment σ. The satisfiability value for G is $\text{val}(G) = \max_{\sigma \colon V \to \Sigma} \text{val}(G, \sigma)$. We say that an assignment σ is a *solution* if $\text{val}(G, \sigma) = 1$, and G is *satisfiable* iff G has a solution. When the context is clear, we omit σ in the description of a constraint, i.e., $\Pi_e(u_e, v_e)$ stands for $\Pi(\sigma(u_e), \sigma(v_e))$.

Boolean Circuits. Throughout the paper, we consider standard Boolean circuits with AND/OR gate of fan-in two and fan-out two, and NOT gate with fan-out two.

Exponential Time Hypothesis (ETH). *Exponential Time Hypothesis (ETH)*, first proposed by Impagliazzo and Paturi [IP01], is a famous strengthening of the P $\neq$ NP hypothesis and that has

since found numerous applications in modern complexity theory, especially in fine-grained complexity.

The ETH postulates that the general 3SAT problem has no sub-exponential algorithm. In this paper, we use the ETH-based hardness of the 4-REGULAR 3-COLORING problem.

Definition 3.1 (4-REGULAR 3-COLORING). A 2CSP instance $G = (V, E, \Sigma, \{\Pi_e\}_{e \in E})$ is an instance of 4-REGULAR 3-COLORING if (1) $\Sigma = \mathbb{F}_4$, (2) the constraint graph is 4-regular, and (3) each Π_e checks whether the two endpoints of e are assigned with different colors from Λ, where $\Lambda \subset \mathbb{F}_4$ has size three and is fixed in advance.

We remark that usually 3-COLORING is defined directly with a ternary alphabet Λ. Here for simplicity of later reductions, we assume the alphabet is $\mathbb{F}_4$. This is without loss of generality since the coloring constraint is also upgraded to check additionally whether the colors are from Λ.

Theorem 3.2 (ETH Lower Bound for 4-REGULAR 3-COLORING [CFG$^+$16]). *Assuming ETH, no algorithm can decide* 4-REGULAR 3-COLORING *in* $2^{o(|V|)}$ *time.*

3.2 Parameterized Complexity Theory

In parameterized complexity theory, we consider a promise language $L_{\text{yes}} \dot\cup L_{\text{no}}$ equipped with a computable function κ, which returns a parameter $\kappa(x) \in \mathbb{N}$ for every input instance x. We use $(L_{\text{yes}} \dot\cup L_{\text{no}}, \kappa)$ to denote a parameterized language. We think of $\kappa(x)$ as a growing parameter that is much smaller than the instance size $|x|$.

A parameterized promise language $(L_{\text{yes}} \dot\cup L_{\text{no}}, \kappa)$ is *fixed parameter tractable* (FPT) if there is an algorithm such that for every input $(x, \kappa(x)) \in (L_{\text{yes}} \dot\cup L_{\text{no}}, \kappa)$, it decides whether $x \in L_{\text{yes}}$ in $f(\kappa(x)) \cdot |x|^{O(1)}$ time for some computable function f.

An *FPT reduction* from $(L_{\text{yes}} \cup L_{\text{no}}, \kappa)$ to $(L'_{\text{yes}} \cup L'_{\text{no}}, \kappa')$ is an algorithm $\mathcal{A}$ which, on every input $(x, \kappa(x))$ outputs another instance $(x', \kappa'(x'))$ such that:

- COMPLETENESS. If $x \in L_{\text{yes}}$, then $x' \in L'_{\text{yes}}$.

- SOUNDNESS. If $x \in L_{\text{no}}$, then $x' \in L'_{\text{no}}$.

- FPT. There exist universal computable functions f and g such that $|\kappa'(x')| \leq g(\kappa(x))$ and the runtime of $\mathcal{A}$ is bounded by $f(\kappa(x)) \cdot |x|^{O(1)}$.

We refer to [FG06] for the backgrounds on fixed parameter tractability and FPT reductions.

ε**-Gap k-Variable CSP.** We mainly focus on the gap version of the parameterized CSP problem. Formally, an ε-Gap k-Variable CSP problem is the following parameterized promise language $(L_{\text{yes}} \dot\cup L_{\text{no}}, \kappa)$.

- L_{yes} consists of all CSPs G with $\text{val}(G) = 1$.

- L_{no} consists of all CSPs G with $\text{val}(G) < 1 - \varepsilon$.

- $\kappa(G)$ equals the number of variables in G.

In other words, we need to decide whether a given CSP instance $(G, |V|)$ with k variables satisfies $\text{val}(G) = 1$ or $\text{val}(G) < 1 - \varepsilon$.

10

Parameterized Inapproximability Hypothesis (PIH). *Parameterized Inapproximability Hypothesis (PIH) is a folklore conjecture generalizing the celebrated PCP theorem to the parameterized complexity. It was first rigorously formulated in [LRSZ20]. Below, we present a slight reformulation, asserting fixed parameter intractability (rather than $W[1]$-hardness specifically) of gap CSP.*

Hypothesis 3.3 (PIH). *For an absolute constant $0 < \varepsilon < 1$, no FPT algorithm can decide ε-Gap k-Variable CSP.*

3.3 Parallel Reed-Muller Code

Word. We say x is a word (a.k.a., vector) with alphabet Σ if x is a string of finite length and each entry is an element from Σ; and Σ^* contains all words with alphabet Σ. Assume x has length m. For each $I \subseteq [m]$, we use x_I to denote the sub-string of x on entries in I. When $I = \{i\}$ is a singleton set, we simply use x_i to denote $x_{\{i\}}$. For a word x over a vector alphabet Σ^t, for each entry x_i and $j \in [t]$, we define $x_i[j]$ as the j-th coordinate of x_i. We define $x[j]$ as $x_1[j] \circ x_2[j] \circ \cdots \circ x_m[j]$, which is a word over Σ.

Let y be another word. We use $x \circ y$ to denote the concatenation of x and y. If y is also of length m, we define $\Delta(x, y) := \mathbf{Pr}_{i \in [m]}[x_i \neq y_i]$ as their relative Hamming distance. For a set S of words, if $\Delta(x, z) \geq \delta$ holds for every $z \in S$, we say x is δ-far from S; otherwise we say x is δ-close to S. In particular, if $S = \emptyset$, then x is 1-far from S. Below, we recall the notion of error correcting codes.

Definition 3.4 (Error Correcting Code (ECC)). An error correcting code is the image of the encoding map $C \colon \Sigma_1^k \to \Sigma_2^K$ with message length k and codeword length K. We say that the ECC has a relative distance δ if $\Delta(C(x), C(y)) \geq \delta$ holds for any distinct $x, y \in \Sigma_1^k$. We use $\delta(C)$ to denote the relative distance of the image map of C and use $\text{Im}(C)$ to denote the codewords of C.

Reed-Muller Code. We use the parallel Reed-Muller (RM) code to construct PCPPs. This parallel operation is called interleaving in coding theory. Below, we present the formal definition.

For an m-variate parallel-output function $f \colon \mathbb{F}^m \to \mathbb{F}^t$, we denote $f[1], \ldots, f[t] \colon \mathbb{F}^m \to \mathbb{F}$ as its single-output components, i.e., $f(x) = (f[1](x), \ldots, f[t](x))$ We say f is of *parallel degree-d* if $f[1], \ldots, f[t]$ are degree-d polynomials, where a polynomial is degree-d if all monomials with (total) degree larger than d have zero coefficients.

If $|\mathbb{F}| > d$ and $|\mathbb{F}|^m \geq \binom{m+d}{d}$, by a dimension argument, there exist $\binom{m+d}{d}$ distinct points (a.k.a., interpolation set) $\{\xi_1, \ldots, \xi_{\binom{m+d}{d}}\} \subseteq \mathbb{F}^m$ whose values $a = (a_1, \ldots, a_{\binom{m+d}{d}}) \in (\mathbb{F}^t)^{\binom{m+d}{d}}$ can uniquely determine the polynomial f_a of parallel degree d.

Definition 3.5 (Parallel RM Code). Assume $|\mathbb{F}| > d$ and $|\mathbb{F}|^m \geq \binom{m+d}{d}$. Let $\{\xi_1, \ldots, \xi_{\binom{m+d}{d}}\}$ be the set above. The $(\mathbb{F}, m, d, t)$-parallel RM code is the image of the following encoding map:

$$\text{RM}^{\mathbb{F},m,d,t} \colon \left(\mathbb{F}^t\right)^{\binom{m+d}{d}} \to \left(\mathbb{F}^t\right)^{|\mathbb{F}|^m},$$

where for each $a = (a_1, \ldots, a_{\binom{m+d}{d}}) \in (\mathbb{F}^t)^{\binom{m+d}{d}}$, the encoding $\text{RM}^{\mathbb{F},m,d,t}(a)$ is the truth table of f_a over the whole space $\mathbb{F}^m$.

In addition, a is the systematic part of the parallel RM encoding, which can be read off directly: for each $j \in \binom{m+d}{d}$, a_j equals the entry indexed by ξ_j in the codeword.

11

By Schwartz-Zippel lemma[4], the relative distance of $(\mathbb{F}, m, d, t)$-parallel RM code is

$$\delta(\text{RM}^{\mathbb{F},m,d,t}) = 1 - \frac{d}{|\mathbb{F}|}.$$

Furthermore, there is an efficient codeword testing procedure (Theorem 3.6) for $\text{RM}^{\mathbb{F},m,d,t}$, the proof of which is in Appendix A.

Theorem 3.6 (Codeword Testing). *Assume* $\text{char}(\mathbb{F}) = 2$ *and* $|\mathbb{F}| \geq \max\left\{6md, 2^{100}m\log|\mathbb{F}|\right\}$. *Let* $\Sigma = \mathbb{F}^{d+1}$ *be the set of univariate degree-d polynomials over* $\mathbb{F}$. *There exists an efficient verifier* $\mathcal{P}_{\text{ldt}}$ *with the following properties.*

- *The input of* $\mathcal{P}_{\text{ldt}}$ *is* $T \circ \pi$, *where* $T \in (\mathbb{F}^t)^{|\mathbb{F}|^m}$ *is supposed to be a codeword of* $\text{RM}^{\mathbb{F},m,d,t}$ *and* $\pi \in (\Sigma^t)^{|\mathbb{F}|^m \cdot (m\log|\mathbb{F}|)^{O(1)}}$ *is the auxiliary proof.*
- $\mathcal{P}_{\text{ldt}}$ *tosses* $m\log|\mathbb{F}| + O(\log\log|\mathbb{F}| + \log m)$ *unbiased coins and makes* 2 *queries on* $T \circ \pi$.
- *If* $T \in \text{Im}(\text{RM}^{\mathbb{F},m,d,t})$, *then there exists some* π *such that* $\mathcal{P}_{\text{ldt}}(T \circ \pi)$ *always accepts.*
- *If* T *is* δ-far from $\text{Im}(\text{RM}^{\mathbb{F},m,d,t})$, *then* $\mathbf{Pr}[\mathcal{P}_{\text{ldt}}(T \circ \pi) \text{ rejects}] \geq 2^{-40}\delta$ *for any* π.

Given a Boolean circuit $C : \{0,1\}^k \to \{0,1\}$, we say a word $x \in (\{0,1\}^t)^k$ *parallel satisfies* C iff $C(x[j]) = 1$ holds for every coordinate $j \in [t]$.

Extracting from the verifier in Theorem 3.6, we obtain a circuit of small size that describes the codeword testing procedure, the proof of which is also in Appendix A.

Theorem 3.7. *Assume* $\text{char}(\mathbb{F}) = 2$ *and* $|\mathbb{F}| \geq \max\left\{6md, 2^{100}m\log|\mathbb{F}|\right\}$. *There exists a Boolean circuit* C_{ldt} *of size* $|\mathbb{F}|^m\text{poly}|\mathbb{F}|$ *for* $T \in (\mathbb{F}^t)^{|\mathbb{F}|^m}$, *where we encode* $\mathbb{F}$ *as* $\{0,1\}^{\log|\mathbb{F}|}$, *such that* T *is codeword of* $\text{RM}^{\mathbb{F},m,d,t}$ *iff* T *parallel satisfies* C_{ldt}.

3.4 Pair Language and Probabilistic Checkable Proof of Proximity

We will perform various satisfiability testing for assignments of Boolean circuits where both are given as input. This motivates the notion of pair language, where the input is naturally divided into two parts corresponding to circuits and assignments respectively.

Pair Language. Formally, L is a pair language over the alphabet Σ_x and Σ_y if all words of L are in the form (x, y) where $x \in \Sigma_x^*, y \in \Sigma_y^*$. For each $x \in \Sigma_x^*$, we define $L(x) = \{y \in \Sigma_y^* | (x, y) \in L\}$ as the restriction of L on x.

Probabilistic checkable proof of proximity (PCPP). PCPP provides robust testing of satisfiability for pair languages.

Definition 3.8 (PCPP). Given a pair language L with alphabet Σ, a $(q, r, \delta, \varepsilon, \Sigma_2)$-PCPP verifier $\mathcal{P}$ takes as input a pair of words (x, y) and an auxiliary proof π with alphabet Σ_2, such that:

(T1) The verifier $\mathcal{P}$ reads all bits of x, tosses at most $r(|x|)$ many unbiased coins, makes at most $q(|x|)$ queries on y and π, and then decides to accept or reject within runtime $(|x| + |\Sigma| + |\Sigma_2|)^{O(1)}$.

   We use I_r to denote the query positions on $y \circ \pi$ under randomness r, and use $\mathcal{P}(x, y, r, a)$ to indicate the behavior (i.e., accept or reject) of $\mathcal{P}$ under randomness r and query answer a.

---

[4]Schwartz-Zippel lemma is usually stated only for single-output polynomials, but it naturally generalizes to the parallel case.

(T2) If $(x,y) \in L$, then there exists some π such that $\mathcal{P}$ always accepts.

(T3) If y is δ-far from $L(x)$, then $\mathcal{P}$ rejects with probability at least ε for every π.

We say $\mathcal{P}$ is a $(q,r,\varepsilon,\Sigma_2)$-PCP verifier if it is a $(q,r,1,\varepsilon,\Sigma_2)$-PCPP verifier.

Remark 3.9 (Proof Length)**.** We can always upper bound $|\pi|$ by $2^{r(|x|)}q(|x|)$, for which reason we do not put an additional parameter in Definition 3.8.

From PCPP to CSP. PCPPs are tightly connected with CSPs. The following standard reduction establishes the connection.

Definition 3.10 (From PCPP to CSP)**.** Given a $(q,r,\delta,\varepsilon,\Sigma_2)$-PCPP verifier $\mathcal{P}$ for the pair language $L = \{(x,y)|x,y \in \Sigma^*\}$. We define a CSP instance $G' = (V',E',\Sigma',\{\Pi'_e\}_{e\in E'})$, where $V' = V_y \dot\cup V_\pi \dot\cup V_{\mathrm{pcpp}}$ and $\Sigma' = \Sigma \cup \Sigma_2$, by the following steps:

- First, we treat each position of y (resp., π) as a single variable in V_y and (resp., V_π) with alphabet Σ (resp., Σ_2). Note that $|V_y| = |y|$ and $|V_\pi| \leq 2^{r(|x|)}q(|x|)$ by Remark 3.9.
- Then, for each choice of random coins $r \in \{0,1\}^{r(|x|)}$, let S_r be the set of query positions over $y \circ \pi$ under randomness r; and we add a variable z_r to V_{pcpp} whose alphabet is $(\Sigma \cup \Sigma_2)^{|S_r|}$, i.e., all possible configurations of the query result. Note that $|V_{\mathrm{pcpp}}| \leq 2^{r(|x|)}$.
- Finally, we add constraints between z_r and every query position $i \in S_r$. The constraint checks whether z_r is an accepting configuration, and the assignment of the position i is consistent with the assignment of z_r.

By construction, the completeness and soundness are preserved up to a factor of q under this reduction, where the loss comes from splitting q queries into q consistency checks. In addition, the reduction from $\mathcal{P}$ to G' is an FPT reduction.

Fact 3.11. *The reduction described in Definition 3.10 is an FPT reduction. Recall that $\mathcal{P}$ is a $(q,r,\delta,\varepsilon,\Sigma_2)$-PCPP verifier for the pair language L over the alphabet Σ. We have the following properties for G':*

- ALPHABET. *The alphabet of G' is $(\Sigma \cup \Sigma_2)^q$.*
- PARAMETER BLOWUP. *The number of varialbes $|V'| \leq |y| + 2^{r(|x|)}q(|x|) + 2^{r(|x|)}$. The number of constraints $|E'| = O(V' \cdot q(|x|))$.*
- COMPLETENESS. *If $(x,y) \in L$, then there exists a solution σ' of G' assigning y to V_y.*
- SOUNDNESS. *If y is δ-far from $L(x)$, then any assignment σ' assigning y to V_y satisfies at most $1 - \frac{\varepsilon}{q}$ fraction of the constraints in G'. Note that if $L(x) = \varnothing$, then $\mathrm{val}(G') \leq 1 - \frac{\varepsilon}{q}$.*

Circuit Value Problem (CKTVAL). CKTVAL is a standard pair language widely used in the classical PCP theorems (see e.g., [BGH$^+$06]). It directly checks if a binary string is a solution to a Boolean circuit.

Definition 3.12 (CKTVAL)**.** CKTVAL is a pair language over $\{0,1\}$ consisting of all words in the form of $w = (C,z)$, where C is a Boolean predicate with $|z|$ input bits and z is a binary string that satisfies C. We define the input length $|w|$ to be the size of C plus $|z|$.

We quote the following almost-linear PCPP result for CKTVAL, which follows from [BGH$^+$06, Theorem 3.3] by setting t to be a constant sufficiently large.

Theorem 3.13 (PCPP for CKTVAL, [BGH⁺06]). *There exists an absolute constant $0 < \delta_\star \le 2^{-100}$ such that* CKTVAL *has*

$$an \left(O(1), \log |w| + O(\log^{0.1} |w|), \delta_\star, \frac{1}{2}, \{0,1\} \right)\text{-PCPP verifier } \mathcal{P}_{\mathsf{ckt}},$$

where $|w|$ is the input length of CKTVAL.

Remark 3.14. In the parallel setting, we want to check if a circuit C is satisfied on t different inputs $z_1, z_2, \ldots, z_t$. Note that according to the Definition 3.8, given (C, z_i) and an auxiliary proof π_i for every $i \in [t]$, the PCPP reads the whole C and then queries z_i and π_i. In other words, the query locations depend only on C. So, when applied in parallel, the PCPP queries the same locations for different $i \in [t]$.

4 Proof of the Main Theorems

This section is devoted to giving a landscape of the proof of Theorem 1.2 and Theorem 1.8. To depict a clear picture, we relegate the proof of technical lemmas in subsequent sections. Below, we first present the formal statement of Theorem 1.2 and Theorem 1.8.

Theorem 4.1 (Formal Version of Theorem 1.2). *Assume ETH. Then for some constant $\varepsilon \in (0,1)$ and any computable function $f(K)$, no algorithm can take as input a 2CSP instance Λ with K variables, $O(K)$ constraints and alphabet $[N]$, distinguish between the following two cases in $f(K)N^{K/2^{\omega(\sqrt{\log K} \log \log K)}}$ time:*

- *Λ is satisfiable;*
- *any assignment satisfies at most $1 - \varepsilon$ fraction of the constraints in Λ.*

Theorem 4.2 (Formal Version of Theorem 1.8). *For any integer $k \ll n$ sufficiently large,* 3SAT *has a constant-query PCP verifier of alphabet size $|\Sigma| = \exp\left(\frac{n}{k} \cdot 2^{O(\sqrt{\log k})} \right)$, runtime $\mathrm{poly}(|\Sigma|, n)$, and $\log k + O\left(\sqrt{\log k} \log \log k \right)$ random coins, which has perfect completeness and soundness $\frac{1}{2}$.*

The proof of Theorem 4.1 follows a similar idea in the previous work [GLR⁺23].

- First, we reduce from 4-REGULAR 3-COLORING, which has the lower bound $2^{\Omega(|V|)}$ under ETH, to a CSP with specific vector structures, termed as the special vector-valued CSP (SVecCSP for short). This step is somewhat similar to the previous approach [GLR⁺23], in the sense that both approaches engineer non-parameterized CSP problems into parameterized CSPs with vector structure. However, the vector structure in [GLR⁺23] is not enough for obtaining the almost-optimal lower bound for k-Variable CSP. This paper obtains a much more refined structure.

- Second, we design a PCPP verifier for SVecCSPs. To obtain an almost-optimal lower bound, we encode the solution of SVecCSPs via Reed-Muller code with an almost-linear blowup. Below, we present the details of our proof.

4.1 Reduction from 4-REGULAR 3-COLORING to SVecCSP

In this section, we define *special vector-valued CSP* (SVecCSP for short). The notion of vector-valued CSPs was first defined in [GLR⁺23], and here we consider a better-structured variant.

14

Definition 4.3 (SVecCSP). A CSP instance $G = (V, E = E_p \dot\cup E_l, \Sigma, \{\Pi_e\}_{e \in E})$ is an SVecCSP if the following properties hold.

(S1) $V = \{x_1, \ldots, x_k\} \dot\cup \{y_1, \ldots, y_k\}$.

(S2) $\Sigma = \mathbb{F}^t$ is a t-dimensional vector space over a finite field $\mathbb{F}$ with $\text{char}(\mathbb{F}) = 2$.

(S3) For each constraint $e = \{u, v\} \in E_p$ where $u = (u_1, \ldots, u_t)$ and $v = (v_1, \ldots, v_t)$ are two variables in V, there is a sub-constraint $\Pi_e^{sub} : \mathbb{F} \times \mathbb{F} \to \{0, 1\}$ such that $\Pi_e(u, v)$ checks Π_e^{sub} on all coordinates, i.e.,

$$\Pi_e(u, v) = \bigwedge_{i \in [t]} \Pi_e^{sub}(u_i, v_i).$$

(S4) $E_l = \{\{x_i, y_i\}_{i \in [k]}\}$. For each $i \in [k]$, there exists a matrix $M_i \in \mathbb{F}^{t \times t}$ and a constraint $\Pi_i(x_i, y_i)$ check if $y_i = M_i x_i$, i.e.,

$$\Pi_i(x_i, y_i) = \mathbb{1}_{y_i = M_i x_i}.$$

The following theorem establishes the hardness of SVecCSP.

Theorem 4.4. *There is a reduction algorithm that holds the following. Given as input an integer $6 \leq k \leq n$ and an n-variable 4-REGULAR 3-COLORING instance Γ, it produces an SVecCSP instance $G = (V, E, \Sigma, \{\Pi_e\}_{e \in E})$ where:*

(R1) VARIABLES AND CONSTRAINTS. $|V| = O(k)$ *and* $|E| = O(k)$.

(R2) RUNTIME. *The reduction runs in time* $\text{poly}(n, 2^{n \log k / k})$.

(R3) ALPHABET. $\Sigma = \mathbb{F}_4^t$ *where* $t = O\left(\frac{n \log k}{k}\right)$.

(R4) SATISFIABILITY. *G is satisfiable iff Γ is satisfiable.*

We defer the proof of Theorem 4.4 to Section 5. Compared with vector-valued CSPs (VecCSP) used in the previous work [GLR+23, Definition 3.3], SVecCSP offers more structural properties.

- For parallel constraints, the previous work [GLR+23] sets up the sub-constraint over *a subset of the coordinates*. By construction, there might be $2^{|E_p|}$ different sub-CSP instances over all coordinates. Each of the $2^{|E_p|}$ sub-CSPs requires an individual PCPP verifier. To simultaneously check the satisfiability of all these sub-instances, they tuple these verifiers into a giant verifier, resulting in an exponential blowup of the proof length. Hence, an almost-linear size of proof is impossible.

  In contrast, SVecCSP sets up the sub-constraint on *all* coordinates, which implies a unified sub-CSP instance among all coordinates. As a result, we avoid the tuple procedure, enabling a highly succinct proof.

- For linear constraints, VecCSP defined in [GLR+23] allow them to be over arbitrary pairs of variables. They introduce auxiliary variables for all pairs of variables and their corresponding linear constraints to check such unstructured constraints. The number of auxiliary variables is $|V| \cdot |E_l|$, which means that the proof length is at least quadratic, making an almost linear size of proof impossible.

  However, SVecCSP only sets up linear constraints over x_i and y_i, with the same index i. After encoding x and y by the parallel Reed-Muller code, we can leverage this alignment and introduce auxiliary proofs, which is also a codeword of the parallel Reed-Mulle code, to check the validity of linear constraints efficiently, with an almost-linear blowup.

15

Though SVecCSP has a more refined structure than VecCSP, we can obtain it from VecCSP by properly duplicating variables and relocating constraints. However, the VecCSP instance in [GLR$^+$23] has parameters $|V| = O(k^2), |E| = O(k^2)$, which are too large to obtain Theorem 4.4. To get around this, we combine results from [Mar10, KMPS23] and the reduction in [GLR$^+$23] to obtain a sparse VecCSP instance with $|V| = O(k), |E| = O(k)$, which ultimately leads to Theorem 4.4.

4.2 Reduction from SVecCSPs to ε-GAP k-VARIABLE CSPs

With the ETH-hardness of SVecCSP by Theorem 4.4, the remaining work is to reduce SVecCSP to ε-Gap k-Variable CSP. To this end, we follow the same idea of previous works [ALM$^+$01, GLR$^+$23].

(L1) First, we encode the solution using an error correcting code.

(L2) Then, we design a constant-query PCPP verifier to check whether the given proof is the codeword of some solution with the aid of auxiliary proofs.

(L3) Finally, we obtain Theorem 4.1 by converting the PCPP verifier into an instance of ε-Gap k-Variable CSP by Fact 3.11.

Remark 4.5. In our actual construction of Item (L2), parallel constraints and linear constraints will be processed separately, building on a (supposedly) same solution. Hence we need to encode the solution using error correcting codes in Item (L1) and design PCPP verifier (instead of PCP verifier) for Item (L2) on top of the shared encoding of the solution.

For the first step (Item (L1)), we need error correcting codes with almost-linear length blowup. In detail, we choose the parallel Reed-Muller code (see Definition 3.5) with suitable choice of parameters. This motivates us to define the following pair language SVecCSP Satisfiability (SVSAT).

Definition 4.6 (SVSAT). $(\mathbb{F}, m, d, t)$-SVSAT is a pair language consisting of $w = (G, (\widehat{x}, \widehat{y}))$ where:

- $G = (V = \{x_1, \ldots, x_k\} \dot\cup \{y_1, \ldots, y_k\}, E = E_\mathsf{p} \dot\cup E_\mathsf{l}, \Sigma = \mathbb{H}^t, \{\Pi_e\}_{e \in E})$ is an SVecCSP instance. We require that $k \leq \binom{m+d}{d}$ and $\mathbb{H}$ is a subfield of $\mathbb{F}$.
- $\widehat{x}, \widehat{y}$ are codewords of $\mathsf{RM}^{\mathbb{F}, m, d, t}$. Suppose $\widehat{x} = \mathsf{RM}^{\mathbb{F}, m, d, t}(\sigma_x)$ and $\widehat{y} = \mathsf{RM}^{\mathbb{F}, m, d, t}(\sigma_y)$ for some $\sigma_x, \sigma_y \in (\mathbb{F}^t)^{\binom{m+d}{d}}$. Define assignment $\sigma : V \to \mathbb{F}^t$ by

$$\sigma(v) := \begin{cases} \sigma_x(i) & v = x_i, \\ \sigma_y(i) & v = y_i. \end{cases}$$

We further require that σ is a solution of G, which implicitly demands $\sigma(v) \in \mathbb{H}^t$ for $v \in V$.

Remark 4.7. Since $\binom{m+d}{d} \geq k$, the index i in defining σ is at most the length of σ_x, σ_y and thus σ is well defined in Definition 4.6. The parameters $\mathbb{F}, m, d, t$ come from the parameters of the parallel Reed-Muller code.

In this pair language, G is the starting SVecCSP instance in our reduction, and $(\widehat{x}, \widehat{y})$ serves as the encoding of a solution. We have the following connection between $(\mathbb{F}, m, d, t)$-SVSAT and the satisfiability of SVecCSP.

Fact 4.8. Let $G = (V = \{x_1, \ldots, x_k\} \dot\cup \{y_1, \ldots, y_k\}, E = E_\mathsf{p} \dot\cup E_\mathsf{l}, \Sigma = \mathbb{H}^t, \{\Pi_e\}_{e \in E})$ be an SVecCSP instance. Assume $k \leq \binom{m+d}{d}$ and $\mathbb{H}$ is a subfield of $\mathbb{F}$. Then, G is satisfiable iff there exists $(\widehat{x}, \widehat{y})$ such that $(G, (\widehat{x}, \widehat{y})) \in (\mathbb{F}, m, d, t)$-SVSAT.

16

Remark 4.9 (Encoding Choice)**.** One may wonder the necessity for encoding $\widehat{x}$ and $\widehat{y}$ separately, as opposed to encoding them jointly as $\widehat{x \circ y}$. This is due to the bipartite structure of the linear constraints E_l (see Item **(S4)**) that we will need to ensure proximity for the encoding on both x and y. This will be clear in Item **(T3)** and Lemma 4.23.

For the second step (Item **(L2)**), we need to construct a PCPP verifier for $(\mathbb{F}, m, d, t)$-SVSAT. Formally, we construct a PCPP verifier with the following parameters.

Theorem 4.10. *Assume* $\mathsf{char}(\mathbb{F}) = 2$ *and* $|\mathbb{F}| \geq \max\{12md, 2^{101}m \log|\mathbb{F}|\}$. *Then for any* $\delta \in [0, 1]$, $(\mathbb{F}, m, d, t)$-SVSAT *has*

$$\text{and } \left(O(1), \log(|E| + |\mathbb{F}|^m) + O\left(\log^{0.1}(|E| + |\mathbb{F}|^m) + \log|\mathbb{F}|\right), \delta, \Omega(\delta), \Sigma^t\right)\text{-PCPP verifier } \mathcal{P},$$

where $|E|$ *is the number of constraints in* $(\mathbb{F}, m, d, t)$-SVSAT *and* $\Sigma = \mathbb{F}^{d+1}$.

Fix some $(G, (\widehat{x}, \widehat{y}))$ supposed to be in $(\mathbb{F}, m, d, t)$-SVSAT. We use σ to denote the assignment recovered (if succeeded) from $\widehat{x}$ and $\widehat{y}$. Our goal is to verify whether σ is the solution of G. To this end, we will handle parallel and linear constraints in Subsection 4.2.1 and Subsection 4.2.3 separately, then combine them together in Subsection 4.2.3 to prove Theorem 4.10.

4.2.1 Verification of Parallel Constraints

We first consider verifying whether σ satisfies all parallel constraints by probing the indices of $(\widehat{x}, \widehat{y})$. Since $\mathsf{RM}^{\mathbb{F}, m, d, t}$ is a systematic code (Definition 3.5), we can recover the value of an index in the assignment σ directly from probing an index of $\widehat{x} \circ \widehat{y}$. In addition, recall that parallel constraints set up the same sub-constraints over all coordinates (Item **(S3)**), we can build up a unified Boolean circuit $C_\mathsf{p} : \{0, 1\}^{2|\mathbb{F}|^m \log|\mathbb{F}|} \rightarrow \{0, 1\}$ that checks whether the systematic part of $\widehat{x}$ and $\widehat{y}$ satisfies the sub-constraints in a single coordinate. In detail, the circuit C_p is defined as follows.

Definition 4.11 (The Circuit C_p)**.** Let $\widetilde{x}, \widetilde{y}$ be words of length $|\mathbb{F}|^m$ over alphabet[5] $\mathbb{F}$. Assume $\widetilde{x}, \widetilde{y}$ are codewords of $\mathsf{RM}^{\mathbb{F}, m, d, t}$. The circuit C_p executes the following.

- C_p recovers the messages $\sigma_{\widetilde{x}}, \sigma_{\widetilde{y}}$ from the systematic part of $\widetilde{x}, \widetilde{y}$ respectively.
- After that, C_p checks whether the assignment $\widetilde{\sigma}$ specified by $\sigma_{\widetilde{x}}, \sigma_{\widetilde{y}}$ has the correct subfield entries and satisfies all constraints in the E_p part of G, at the single-coordinate level. Specifically, $\widetilde{\sigma}$ is the assignment of V defined by

$$\sigma(v) = \begin{cases} \sigma_{\widetilde{x}}(i) & v = x_i, \\ \sigma_{\widetilde{y}}(i) & v = y_i. \end{cases}$$

For every $v \in V$, C_p checks whether $\sigma(v) \in \mathbb{H}$; and for every constraint $e = \{u, v\} \in E_\mathsf{p}$, C_p checks whether $\Pi_e^{sub}(\sigma(u), \sigma(v)) = 1$.

The size of circuit C_p is bounded by $(|\mathbb{F}|^m + k + |E_\mathsf{p}|) \cdot \mathsf{poly}|\mathbb{F}| = (|\mathbb{F}|^m + |E|) \cdot \mathsf{poly}|\mathbb{F}|$.

Double Test Problem (DOUBLETEST**).** In light of Definition 4.11, $(\widehat{x}, \widehat{y})$ satisfies all parallel constraints iff $\widehat{x}, \widehat{y}$ are correct codewords and $C_\mathsf{p}(\widehat{x}[i] \circ \widehat{y}[i]) = 1$ for each $i \in [t]$. This motivates us to consider the following pair language DOUBLETEST related to CKTVAL.

---

[5]For ease of presentation, the circuit's input is described as alphabet $\mathbb{F}$. We take the trivial conversion from an element of $\mathbb{F}$ into a binary string of length $\log|\mathbb{F}|$.

Definition 4.12 (DOUBLETEST). Assume $\mathrm{char}(\mathbb{F}) = 2$. $(\mathbb{F}, m, d, t)$-DOUBLETEST is a pair language over $\Sigma_x = \{0, 1\}, \Sigma_y = \mathbb{F}^t$ consisting of $w = (C, T_1 \circ T_2)$ where

- C is a Boolean circuit with $2|\mathbb{F}|^m \log |\mathbb{F}|$ input bits and $T_1, T_2 \in (\mathbb{F}^t)^{|\mathbb{F}|^m}$ are codewords of $\mathrm{RM}^{\mathbb{F},m,d,t}$;
- if we view $\mathbb{F}$ as $\{0, 1\}^{\log |\mathbb{F}|}$, $T_1 \circ T_2$ parallel satisfies C.

We define the input length $|w|$ to be the size of C plus $2|\mathbb{F}|^m \log |\mathbb{F}|$. Note that the dimension t is reflected on the alphabet, not on the length.

In short, DOUBLETEST extends CKTVAL by allowing the assignment to have more dimensions (i.e., t), allowing the assignment to be partitioned into two parts (i.e., T_1, T_2), and assuming each part is encoded by parallel RM code.

Given Definition 4.12 and Definition 4.11, we have the following statement. Note that the assumption that $\widetilde{x}, \widetilde{y}$ are codewords of $\mathrm{RM}^{\mathbb{F},m,d,t}$ in Definition 4.11 is guaranteed in Definition 4.12.

Fact 4.13. *Parallel constraints E_{p} in G are satisfied iff $(C_{\mathsf{p}}, \widehat{x} \circ \widehat{y}) \in (\mathbb{F}, m, d, t)$-DOUBLETEST.*

Thus, to verify that σ satisfies all parallel constraints, if suffices to construct a PCPP verifier for DOUBLETEST. The verifier is formally stated as follows and will be proved in Section 6.

Theorem 4.14 (PCPP for DOUBLETEST). *Assume $\mathrm{char}(\mathbb{F}) = 2$ and $|\mathbb{F}| \geq \max\{6md, 2^{100}m \log |\mathbb{F}|\}$. For any $\delta \in [0, 1]$, $(\mathbb{F}, d, m, t)$-DOUBLETEST has*

$$an \left(O(1), \log |w| + O\left(\log^{0.1} |w| + \log |\mathbb{F}|\right), \delta, \Omega(\delta), \Sigma^t\right)\text{-PCPP verifer } \mathcal{P}_{\mathsf{dt}},$$

where $|w|$ is the input length of $(\mathbb{F}, d, m, t)$-DOUBLETEST and $\Sigma = \mathbb{F}^{d+1}$.

4.2.2 Verification of Linear Constraints

We then turn to verifying whether σ satisfies all linear constraints E_{l}. Recall from Item (S4) that all linear constraints are set up between x_i and y_i, with the constraint that $y_i = M_i x_i$. By defining auxiliary variables $z_i := y_i - M_i x_i$, it suffices to check whether $z_i \equiv \vec{0}_t$ for every $i \in [k]$.

Thus, we add the auxiliary proof $\widehat{z} \in (\mathbb{F}^t)^{|\mathbb{F}|^m}$, which is supposed to fulfill the following condition

$$\widehat{z}(p) = \widehat{y}(p) - \widehat{M}(p)\widehat{x}(p) \quad \text{holds for all } p \in \mathbb{F}^m, \tag{1}$$

where $\widehat{M} \in (\mathbb{F}^{t \times t})^{|\mathbb{F}|^m}$ is the parallel RM encoding of $(M_1, \ldots, M_k, 0^{t \times t}, \ldots, 0^{t \times t}) \in (\mathbb{F}^{t \times t})^{\binom{m+d}{d}}$. Here we extend Definition 3.5 for matrix values: the value of ξ_i is M_i for $i \leq k$ and is $0^{t \times t}$ for $i > k$. Equivalently, for each matrix coordinate $(i, j) \in [t] \times [t]$, $\widehat{M}[i, j] \in \mathbb{F}^{|\mathbb{F}|^m}$ is the RM encoding of $(M_1[i, j], \ldots, M_k[i, j], 0, \ldots, 0)$. We remark that entries of $\widehat{M}$ can be efficiently computed on the fly by demand and are not included as a proof for the PCPP verifier.

Based on the discussion above, we obtain the following fact.

Fact 4.15. *Linear constraints E_{l} in G are satisfied iff $\widehat{x}, \widehat{y}$ are codewords of $\mathrm{RM}^{\mathbb{F},m,d,t}$ and the systematic part of $\widehat{z}$, defined by (1), are all 0^t.*

Recall that $\widehat{x}, \widehat{y}$ being correct codewords are guaranteed by the analysis of parallel constraints above. Hence we focus on testing the systematic part of $\widehat{z}$, which amount to $\widehat{z}$ parallel satisfying the following circuit.

18

Definition 4.16 (The Circuit C_l). The circuit C_l receives as input a word $\widetilde{z}$ of length $|\mathbb{F}|^m$ over alphabet $\mathbb{F}$. It checks if $\widetilde{z}(\xi_i) = 0$ holds for all $i \in [k]$, where we recall that ξ_i from Definition 3.5.

The size of the circuit C_l is bounded by $(|\mathbb{F}|^m + k) \cdot \mathsf{poly}|\mathbb{F}|$.

We will use a variant of DOUBLETEST, denoted SINGLETEST, on C_l and $\widehat{z}$. But before that, we give a degree bound for $\widehat{z}$.

Claim 4.17. If $\widehat{x}, \widehat{y}$ are codewords of $\mathrm{RM}^{\mathbb{F},m,d,t}$, then the $\widehat{z}$ defined by (1) is a codeword of $\mathrm{RM}^{\mathbb{F},m,2d,t}$.

Proof. Expanding the matrix multiplication, for each coordinate $i \in [t]$, we have

$$\widehat{z}[i](p) = \widehat{y}[i](p) - \sum_{j \in [t]} \widehat{M}[i,j](p) \cdot \widehat{x}[j](p) \quad \text{for all } p \in \mathbb{F}^m,$$

where $\widehat{M}[i,j](p)$ is the (i,j)-th entry of $\widehat{M}(p)$. Since $\widehat{y}[i], \widehat{M}[i,j], \widehat{x}[j]$ are truth tables of degree-d polynomials, $\widehat{z}[j]$ is the truth table of a degree-$2d$ polynomial, which means $\widehat{z} \in \mathrm{Im}(\mathrm{RM}^{\mathbb{F},m,2d,t})$. $\square$

Single Test Problem (SINGLETEST**).** At this point, checking Fact 4.15 can be safely handled by the following SINGLETEST with proper degree conditions.

Definition 4.18 (SINGLETEST). Assume $\mathrm{char}(\mathbb{F}) = 2$. $(\mathbb{F}, m, d, t)$-SINGLETEST is a pair language over $\Sigma_{\mathsf{x}} = \{0, 1\}, \Sigma_{\mathsf{y}} = \mathbb{F}^t$ consisting of $w = (C, T_1)$ where

- C is a Boolean circuit with $2|\mathbb{F}|^m \log |\mathbb{F}|$ input bits and $T_1 \in (\mathbb{F}^t)^{|\mathbb{F}|^m}$ is a codeword of $\mathrm{RM}^{\mathbb{F},m,d,t}$;
- if we view $\mathbb{F}$ as $\{0,1\}^{\log |\mathbb{F}|}$, T_1 parallel satisfies C.

We define the input length $|w|$ to be the size of C plus $|\mathbb{F}|^m \log |\mathbb{F}|$.

In comparison, SINGLETEST simply removes the second table T_2 from DOUBLETEST. Combining Claim 4.17 and Fact 4.15, we have the following result.

Fact 4.19. *Linear constraints E_l in G are satisfied iff $\widehat{x}, \widehat{y}$ are codewords of $\mathrm{RM}^{\mathbb{F},m,d,t}$, $\widehat{z}$ satisfies (1), and $(C_l, \widehat{z}) \in (\mathbb{F}, m, 2d, t)$-SINGLETEST.*

Analogous to Theorem 4.14, we also have an efficient PCPP verifier for SINGLETEST as follows, which will also be proved in Section 6.

Theorem 4.20 (PCPP for SINGLETEST). *Assume $\mathrm{char}(\mathbb{F}) = 2$ and $|\mathbb{F}| \geq \max\{6md, 2^{100}m\log|\mathbb{F}|\}$. For any $\delta \in [0,1]$, $(\mathbb{F}, d, m, t)$-SINGLETEST has*

$$an \left(O(1), \log|w| + O\left(\log^{0.1}|w| + \log|\mathbb{F}|\right), \delta, \Omega(\delta), \Sigma^t \right)\text{-PCPP verifer } \mathcal{P}_{\mathsf{st}},$$

where $|w|$ is the input length of $(\mathbb{F}, d, m, t)$-SINGLETEST and $\Sigma = \mathbb{F}^{d+1}$.

Finally, we briefly sketch how to check if $\widehat{z}$ satisfies (1), as needed in Fact 4.19. This will be done by randomly picking an index p and checking whether (1) holds on that p. The soundness will be analyzed by Schwartz-Zippel Lemma. This will be formalized in Subsection 4.2.3.

19

4.2.3 The Whole Construction and Analysis

Based on the above discussion, we are now ready to construct the PCPP verifier $\mathcal{P}$ for SVSAT. We invoke Theorem 4.14 to obtain PCPP verifiers $\mathcal{P}_{\mathsf{dt}}$ and $\mathcal{P}_{\mathsf{st}}$ for $(\mathbb{F}, m, d, t)$-DOUBLETEST and $(\mathbb{F}, m, 2d, t)$-SINGLETEST respectively.

Recall that the input of $(\mathbb{F}, m, d, t)$-SVSAT is $(G, (\widehat{x}, \widehat{y}))$. The auxiliary proof consists of $\widehat{z}$, π_1, and π_2, where

- $\widehat{z}$ is supposed to be a codeword in $\mathsf{RM}^{\mathbb{F}, m, 2d, t}$ and $\widehat{z}(p) = \widehat{y}(p) - \widehat{M}(p)\widehat{x}(p)$ for all $p \in \mathbb{F}^m$;

- π_1 is supposed to be the auxiliary proof to convince $\mathcal{P}_{\mathsf{dt}}$ that $(C_{\mathsf{p}}, \widehat{x} \circ \widehat{y})$ belongs to the pair language $(\mathbb{F}, m, d, t)$-DOUBLETEST.

- π_2 is supposed to be the auxiliary proof to convince $\mathcal{P}_{\mathsf{st}}$ that $(C_{\mathsf{l}}, \widehat{z})$ belongs to the pair language $(\mathbb{F}, m, 2d, t)$-SINGLETEST.

The verifier $\mathcal{P}$ performs one of the following three tests with equal probability.

(T1) Feed the pair of words $(C_{\mathsf{p}}, \widehat{x} \circ \widehat{y})$ and the auxiliary proof π_1 into $\mathcal{P}_{\mathsf{dt}}$. Reject if $\mathcal{P}_{\mathsf{dt}}$ rejects.

(T2) Feed the pair of words $(C_{\mathsf{l}}, \widehat{z})$ and the auxiliary proof π_2 into $\mathcal{P}_{\mathsf{st}}$. Reject if $\mathcal{P}_{\mathsf{st}}$ rejects.

(T3) Generate a random point $p \in \mathbb{F}^m$, reject if $\widehat{z}(p) \neq \widehat{y}(p) - \widehat{M}(p)\widehat{x}(p)$.

At this point, we are ready to analyze the PCPP verifier $\mathcal{P}$. In particular, Theorem 4.10 follows from the combination of the following Lemma 4.21, Lemma 4.22, and Lemma 4.23.

Lemma 4.21 (Parameters). *Assume* $\mathsf{char}(\mathbb{F}) = 2$ *and* $|\mathbb{F}| \geq \max\left\{12md, 2^{101}m\log|\mathbb{F}|\right\}$. *Then* $\mathcal{P}$ *tosses*

$$\log(|E| + |\mathbb{F}|^m) + O\left(\log^{0.1}(|E| + |\mathbb{F}|^m) + \log|\mathbb{F}|\right).$$

unbiased coins and makes $O(1)$ *queries. The alphabet of the auxiliary proof is* Σ^t *where* $\Sigma = \mathbb{F}^{d+1}$.

Proof. By Theorem 4.14 and Theorem 4.20, both $\mathcal{P}_{\mathsf{dt}}$ and $\mathcal{P}_{\mathsf{st}}$ make constant queries. Also in Item **(T3)**, $\mathcal{P}$ makes 3 queries on $\widehat{x}, \widehat{y}, \widehat{z}$. Thus $\mathcal{P}$ makes $O(1)$ total queries.

By Definitions 4.11 and 4.16, the input lengths of $\mathcal{P}_{\mathsf{dt}}$ and $\mathcal{P}_{\mathsf{st}}$ in Item **(T1)** and Item **(T2)** are

$$\begin{cases} |w_1| = (|\mathbb{F}|^m + |E|) \cdot \mathsf{poly}|\mathbb{F}| + 2|\mathbb{F}|^m\log|\mathbb{F}| & \leq (|E| + |\mathbb{F}|^m)\mathsf{poly}|\mathbb{F}|, \\ |w_2| = (|\mathbb{F}|^m + k) \cdot \mathsf{poly}|\mathbb{F}| + |\mathbb{F}|^m\log|\mathbb{F}| & \leq (|E| + |\mathbb{F}|^m)\mathsf{poly}|\mathbb{F}|, \end{cases}$$

respectively. Putting this into Theorem 4.14 and Theorem 4.20, the number of unbiased coins used in $\mathcal{P}_{\mathsf{dt}}$ and $\mathcal{P}_{\mathsf{st}}$ is

$$\log(|E| + |\mathbb{F}|^m) + O\left(\log^{0.1}(|E| + |\mathbb{F}|^m) + \log|\mathbb{F}|\right).$$

In Item **(T3)**, $\mathcal{P}$ tosses $m\log|\mathbb{F}|$ coins. Since $\mathcal{P}$ only executes one of the three tests, the randomness is bounded by their maximum.

The auxiliary proofs π_1, π_2 have alphabet Σ by Theorem 4.14 and Theorem 4.20. The alphabet of $\widehat{z}$ is $\mathbb{F}^t$, which can also be embedded into the larger Σ^t. $\qquad \square$

Lemma 4.22 (Completeness). *Assume* $\mathsf{char}(\mathbb{F}) = 2$ *and* $|\mathbb{F}| \geq \max\left\{12md, 2^{101}m\log|\mathbb{F}|\right\}$. *Suppose* $\widehat{x} = \mathsf{RM}^{\mathbb{F}, m, d, t}(\sigma_x), \widehat{y} = \mathsf{RM}^{\mathbb{F}, m, d, t}(\sigma_y)$, *and the assignment* σ *given by* σ_x *and* σ_y *(recall Definition 4.6) is a solution to* G. *Then there exist* $\widehat{z}, \pi_1, \pi_2$ *which* $\mathcal{P}$ *accepts with probability* 1.

Proof. By Definition 4.11, $(\widehat{x}, \widehat{y})$ parallel satisfies the circuit C_p. Thus $(C_p, \widehat{x} \circ \widehat{y}) \in (\mathbb{F}, m, d, t)$-DOUBLETEST by Fact 4.13. Hence there exists an auxiliary proof π_1 which makes $\mathcal{P}_{dt}$ accepts with probability 1. Item **(T1)** therefore always passes.

For each $p \in \mathbb{F}^m$, define $\widehat{z}(p) = \widehat{y}(p) - \widehat{M}(p)\widehat{x}(p)$. By Claim 4.17, $\widehat{z} \in \text{Im}(\text{RM}^{\mathbb{F}, m, 2d, t})$. Let $\sigma_x, \sigma_y, \sigma_z$ be the messages $\widehat{x}, \widehat{y}, \widehat{z}$ encodes respectively. Note that (σ_x, σ_y) satisfies the linear constraints E_l in G, i.e., $\sigma_y(i) - M_i\sigma_x(i) = 0^t$ for all $i \in [k]$. Hence all $i \in [k]$, we have

$$\sigma_z(i) = \widehat{z}(\xi_i) = \widehat{y}(\xi_i) - \widehat{M}(\xi_i)\widehat{x}(\xi_i) = \sigma_y(i) - M_i\sigma_x(i) = 0^t,$$

where $\{\xi_1, \ldots, \xi_{\binom{m+d}{d}}\}$ are the distinct points defining the encoding of parallel RM code (see Definition 3.5). Therefore, $\widehat{z}$ parallel satisfies C_l. By Fact 4.19, there exists an auxiliary proof π_2, which makes $\mathcal{P}_{st}$ accepts with probability 1, and Item **(T2)** always passes.

Finally Item **(T3)** always passes due to the definition of $\widehat{z}$. This completes the proof. $\square$

Lemma 4.23 (Soundness). *Assume* $\text{char}(\mathbb{F}) = 2$ *and* $|\mathbb{F}| \geq \max\{12md, 2^{101}m \log |\mathbb{F}|\}$. *Let* $\delta \in [0, 1]$ *be arbitrary. If* $(\widehat{x}, \widehat{y})$ *is δ-far from satisfying, i.e., δ-far from the restriction of $(\mathbb{F}, m, d, t)$-SVSAT on G, then $\mathcal{P}$ rejects with probability $\Omega(\delta)$.*

Proof. Let $\kappa \geq 1$ be a large constant, the specific value of which depends on the hidden constants in Theorem 4.14 and Theorem 4.20. By modifying the hidden constant in $\Omega(\cdot)$ here and noticing that δ-far implies δ'-far for any $\delta' \leq \delta$, we safely assume $\delta \leq 1/\kappa^2$.

Fix arbitrary $(\widehat{x}, \widehat{y})$ that is δ-far from satisfying. Assume that $\mathcal{P}$ rejects with probability at most $\kappa \cdot \delta$, since otherwise the statement already holds. Then each of the tests Item **(T1)**, Item **(T2)**, and Item **(T3)** reject with probability at most $3\kappa \cdot \delta$. By choosing κ sufficiently large and according to soundness guarantee of $\mathcal{P}_{dt}$ and $\mathcal{P}_{st}$ in Theorem 4.14 and Theorem 4.20, we know

1. $(\widehat{x}, \widehat{y})$ is δ-close to $(\overline{x}, \overline{y})$, which is a pair of codewords of $\text{RM}^{\mathbb{F}, m, d, t}$ that parallel satisfies C_p.

   Since $\widehat{x}, \widehat{y}$ have the same length, this alse implies that $\widehat{x}$ is 2δ-close to $\overline{x}$ and $\widehat{y}$ is 2δ-close to $\overline{y}$.

2. $\widehat{z}$ is δ-close to $\overline{z}$, which is a codeword of $\text{RM}^{\mathbb{F}, m, 2d, t}$ that parallel satisfies C_l.

We aim to show that $(G, (\overline{x}, \overline{y})) \in (\mathbb{F}, m, d, t)$-SVSAT, which contradicts to the assumption that $(\widehat{x}, \widehat{y})$ is δ-far from satisfying and completes the proof.

Let $\sigma_{\overline{x}}, \sigma_{\overline{y}}, \sigma_{\overline{z}}$ be the messages of $\overline{x}, \overline{y}, \overline{z}$ respectively. It now suffices to prove the assignment σ given by

$$\sigma(v) = \begin{cases} \sigma_{\overline{x}}(i) & v = x_i \\ \sigma_{\overline{y}}(i) & v = y_i \end{cases}$$

satisfies all constraints in $E_p \dot\cup E_l$.

By Item 1 and Fact 4.13, σ satisfies all constraints in E_p. To analyze constraints in E_l, we first prove that $\overline{z} = \overline{y} - \widehat{M}\overline{x}$ in accordance with (1). Assume this is false for some entry $p \in \mathbb{F}^m$, i.e.,

$$\overline{z}(p) \neq \overline{y}(p) - \widehat{M}(p)\overline{x}(p). \tag{2}$$

Note that $\overline{x}, \overline{y}, \widehat{M}$ are all of parallel degree-d and $\overline{z}$ is of parallel degree-$2d$. Then by Schwartz-Zippel lemma, (2) actually happens for at least $1 - \frac{2d}{|\mathbb{F}|}$ fraction of points $p \in \mathbb{F}^m$. Now recall the test in Item **(T3)**, which checks precisely the above for a random $p \sim \mathbb{F}^m$ with $\overline{x}, \overline{y}, \overline{z}$ replaced by $\widehat{x}, \widehat{y}, \widehat{z}$. By Items 1 and 2 and a union bound, with probability at least $1 - 5\delta - \frac{2d}{|\mathbb{F}|}$, on this random p

21

we have $\widehat{x}(p) = \overline{x}(p), \widehat{y}(p) = \overline{y}(p), \widehat{z}(p) = \overline{z}(p)$ and (2) happens, which makes Item **(T3)** reject. By our assumption on $|\mathbb{F}|$ and $\delta \leq 1/\kappa^2$ with κ sufficiently large, this rejection probability is at least $0.9 > 3\kappa \cdot \delta$ and contradicts to our assumption on the rejection probability of Item **(T3)**. In short, (2) can never happen.

Finally we are ready to show that constraints in E_l are satisfied by σ. By Item 2 and Fact 4.19, $\sigma_{\overline{z}}(i) = 0^t$ holds for all $i \in [k]$, and thus

$$\sigma_{\overline{y}}(i) - M_i \sigma_{\overline{x}}(i) = \overline{y}(\xi_i) - \widehat{M}(\xi_i)\overline{x}(\xi_i) = \overline{z}(\xi_i) = \sigma_{\overline{z}}(i) = 0.$$

Therefore, all constraints in E_l are also satisfied. This completes the whole soundness proof. □

4.3 Putting Everything Together

Now, we are ready to prove the main theorems.

Proof of Theorem 4.1. We start with an arbitrary n-variable 4-REGULAR 3-COLORING instance Γ, which prohibits algorithms of runtime $2^{o(n)}$ in the worst case by Theorem 3.2 assuming ETH. By Theorem 4.4, we obtain an SVecCSP instance $G = (V, E, \mathbb{F}_4^t, \{\Pi_e\}_{e \in E})$ in time $\mathrm{poly}(n, 2^{n \log k / k})$ which preserves the satisfiability of Γ. In addition, $t = O\left(\frac{n \log k}{k}\right)$ and $|V| = O(k), |E| = O(k)$.

Let m and d be integers to be chosen later satisfying

$$k \leq \binom{m+d}{d}. \tag{3}$$

Let $\mathbb{F}$ be a field of characteristic two which contains $\mathbb{F}_4$ as a subfield and satisfies

$$|\mathbb{F}| \geq \max\left\{12md, 2^{101} m \log |\mathbb{F}|\right\}. \tag{4}$$

By Fact 4.8, G is satisfiable iff there exists $\widehat{x}, \widehat{y} \in (\mathbb{F}^t)^{|\mathbb{F}|^m}$ such that $(G, (\widehat{x}, \widehat{y})) \in (\mathbb{F}, m, d, t)$-SVSAT. Then we construct the PCPP verifier $\mathcal{P}$ for $(\mathbb{F}, m, d, t)$-SVSAT from Theorem 4.10 with $\delta = 1$ to obtain a PCP verifier (recall Definition 3.8) $\mathcal{P}'$ for the satisfiability of G.

The query complexity, completeness, alphabet, and randomness of $\mathcal{P}'$ follow from those of $\mathcal{P}$ in Theorem 4.10; and the soundness of $\mathcal{P}'$ is the (unspecified) constant soundness parameter by setting $\delta = 1$ in Theorem 4.10. In particular,

$$\text{the alphabet size is } |\Sigma|^t = |\mathbb{F}|^{(d+1) \cdot t} = |\mathbb{F}|^{O(dn \log k / k)}$$

and

$$\text{the randomness is } \log(k + |\mathbb{F}|^m) + O\left(\log^{0.1}(k + |\mathbb{F}|^m) + \log |\mathbb{F}|\right) \text{ coins.}$$

Then we apply Fact 3.11 and obtain a 2CSP instance Λ preserving the satisfiability of G (and thus Γ) where

- the size of the alphabet of Λ is
$$N = |\mathbb{F}|^{O(dn \log k / k)},$$

- the number of variables in Λ is at most
$$K = 2|\mathbb{F}|^m + 2^{\log(k + |\mathbb{F}|^m) + O\left(\log^{0.1}(k + |\mathbb{F}|^m) + \log |\mathbb{F}|\right)} \cdot O(1) = (k + |\mathbb{F}|^m) \cdot \mathrm{poly}\left(|\mathbb{F}|, 2^{\log^{0.1}(k + |\mathbb{F}|^m)}\right),$$

- the number of constraints in Λ is a constant multiple of the number of variables in Λ.

Finally we optimize the choice of $m, d, |\mathbb{F}|$. Assume $\log k$ is a perfect square and is sufficiently large and set

$$|\mathbb{F}| = 2^{1000} \log k \cdot 2^{\sqrt{\log k}}, \quad m = \sqrt{\log k}, \quad d = 2^{30} \sqrt{\log k} \cdot 2^{\sqrt{\log k}}.$$

Then

$$\binom{m+d}{d} \geq \left(\frac{d}{m}\right)^m = \left(2^{30} \cdot 2^{\sqrt{\log k}}\right)^{\sqrt{\log k}} \geq k,$$

which is consistent with (3). We also have

$$12md = 12 \cdot 2^{30} \cdot \log k \cdot 2^{\sqrt{\log k}} \leq |\mathbb{F}| \quad \text{and}$$

$$2^{101} m \log |\mathbb{F}| \leq 2^{101} \sqrt{\log k} \left(\sqrt{\log k} + \log \log k + 1000\right) \leq |\mathbb{F}|,$$

which is consistent with (4). Moreover, Λ has alphabet size N and the number of variables K as follows.

$$N = \left(\log k \cdot 2^{\sqrt{\log k}}\right)^{O\left(\frac{n \cdot \log^{1.5} k \cdot 2^{\sqrt{\log k}}}{k}\right)} \quad \text{and} \quad K = k \cdot 2^{O(\sqrt{\log k} \log \log k)}$$

where N can be furthered simplified to

$$N = \left(2^{n/k}\right)^{2^{O(\sqrt{\log k})}},$$

Since Γ has no $2^{o(n)}$-time algorithm by assumption, Λ has the lower bound $f(K) \cdot N^{K/2^{\omega(\sqrt{\log K} \log \log K)}}$-time for any computable function f as claimed.

$\square$

The PCP statement Theorem 4.2 follows directly from the proof above.

Proof of Theorem 4.2. From any instance of 3SAT, there is a linear-size reduction to an instance of 4-REGULAR 3-COLORING [Mar10]. Therefore we only need to construct the desired PCP for 4-REGULAR 3-COLORING. This follows directly from the verifier $\mathcal{P}'$ in the proof of Theorem 4.1. In particular, we stick to the parameter choice there and obtain a PCP verifier with alphabet size

$$2^{n \cdot 2^{O(\sqrt{\log k})}/k}$$

and randomness

$$\log k + O\left(\sqrt{\log k} \log \log k\right).$$

The implicit constant soundness of $\mathcal{P}'$ can be boosted to $\frac{1}{2}$ by a constant number of randomness-efficient query repetitions (see e.g., [BGH+06, Lemma 2.11]).

$\square$

5 From 4-REGULAR 3-COLORING to SVecCSP

The goal of this section is to reduce 4-REGULAR 3-COLORING, which is known to be ETH-hard Theorem 3.2, to SVecCSP.

Theorem (Theorem 4.4 Restated). *There is a reduction algorithm such that the following holds. Given as input an integer $6 \leq k \leq n$ and an n-variable 4-REGULAR 3-COLORING instance Γ, it produces an SVecCSP instance $G = (V, E, \Sigma, \{\Pi_e\}_{e \in E})$ where:*

(R1) VARIABLES AND CONSTRAINTS. *$|V| = O(k)$ and $|E| = O(k)$.*

(R2) RUNTIME. *The reduction runs in time $\mathsf{poly}(n, 2^{n \log k / k})$.*

(R3) ALPHABET. *$\Sigma = \mathbb{F}_4^t$ where $t = O\left(\frac{n \log k}{k}\right)$.*

(R4) SATISFIABILITY. *G is satisfiable iff Γ is satisfiable.*

The reduction starts by grouping vertices into supernodes, which take vector values. Then the constraints between supernodes correspond to (possibly multiple) constraints in the original instance. To make sure the new instance has small size, we need a grouping method (Lemma 5.1) that produces as few supernodes and constraints as possible. Then we make duplicates of variables and rearrangements of their coordinates to make sure parallel constraints are scattered properly (Proposition 5.4). Finally we make more duplicates to ensure that linear constraints form a matching and parallel constraints are applied on all coordinates (Proposition 5.5).

5.1 From 4-REGULAR 3-COLORING to VecCSP

The following Lemma 5.1 serves the purpose of the grouping method and can be seen as a parameterized version of the *sparsification lemma* [IP01, IPZ01] in classical computation complexity.

Lemma 5.1 ([Mar10, KMPS23]). *There is an algorithm $\mathcal{A}$ such that the following holds. $\mathcal{A}$ takes as input a 2CSP instance $G = (V, E, \Sigma, \{\Pi_e\}_{e \in E})$ and an integer $6 \leq k \leq |V|$, outputs a 2CSP instance $G' = (V', E', \Sigma^t, \{\Pi_e'\}_{e \in E'})$ in time $\mathsf{poly}\left(|V|, |\Sigma|^t\right)$ where $|V'| = k$ and $t \leq O\left((|V| + |E|) \cdot \frac{\log k}{k}\right)$, such that G is satisfiable iff G' is satisfiable. In addition, the constraint graph of G' is a 3-regular graph.*

In the actual construction, each $x \in V'$ corresponds to a subset $S(x) \subseteq V$ of size t and takes values in $\Sigma^{S(x)}$ as assignments to variables in $S(x)$. For each $e = \{x, y\} \in E'$, the constraint Π_e is the conjunction of the following:

1. *equality constraints on common variables of $S(x)$ and $S(y)$;*
2. *constraints across[6] $S(x)$ and $S(y)$ in G.*

Remark 5.2. Note that since the constraint graph of G' is 3-regular, the total number of constraints in G' is linear in $|V'|$. In addition, t only incurs an extra $\log k = k^{o(1)}$ blowup over the information theoretic limit $(|V| + |E|)/k$. This is crucial for us to get almost tight hardness. Indeed, naive approaches (e.g., the reduction in [GLR⁺23]) will incur a polynomial loss. We also remark that it is an open problem whether the extra $\log k$ can be further removed, which, if true, implies a precise parameterized analog of the sparsification lemma and has many applications. See [Mar10] for detailed discussions.

---

[6]The construction in [KMPS23] ensures that each original constraint $(u, v) \in E$ is covered by some new constraint $(x, y) \in E'$ in the sense that $u \in S(x), v \in S(y)$. This means that we only need to check the cross constraints in G' and omit the ones purely inside $S(x)$ or inside $S(y)$.

Given Lemma 5.1, we first obtain a vector-valued CSP (VecCSP) instance (Proposition 5.4), which we will soon convert into SVecCSP (Proposition 5.5).

Definition 5.3 (VecCSP, [GLR⁺23]). A CSP instance $G = (V, E, \Sigma, \{\Pi_e\}_{e \in E})$ is a VecCSP if the following properties hold.

- $\Sigma = \mathbb{F}^t$ is a t-dimensional vector space over a finite field $\mathbb{F}$ with $\text{char}(\mathbb{F}) = 2$.
- For each constraint $e = \{u, v\} \in E$ where $u = (u_1, \ldots, u_t)$ and $v = (v_1, \ldots, v_t)$ are two variables in V, the constraint validity function Π_e is classified as one of the following cases:

    - LINEAR. There exists a matrix $M_e \in \mathbb{F}^{t \times t}$ such that

    $$\Pi_e(u, v) = \mathbb{1}_{u = M_e v}.$$

    - PARALLEL. There exists a *sub-constraint* $\Pi_e^{sub} : \mathbb{F} \times \mathbb{F} \to \{0, 1\}$ and a subset of coordinates $Q_e \subseteq [t]$ such that Π_e checks Π_e^{sub} for every coordinate in Q_e, i.e.,

    $$\Pi_e(u, v) = \bigwedge_{i \in Q_e} \Pi_e^{sub}(u_i, v_i).$$

- Each variable is related to at most one parallel constraint.

Note that SVecCSP, the special case of VecCSP we use, additionally enforces linear constraints to be a matching and enforces parallel constraints to operate on all coordinates (i.e., $Q_e = [t]$).

Proposition 5.4 (VecCSP Intermediate Instance). *There is a reduction algorithm such that the following holds. Given as input an integer $6 \leq k \leq n$ and an n-variable 4-REGULAR 3-COLORING instance Γ, it produces an VecCSP instance $G = (V, E, \Sigma, \{\Pi_e\}_{e \in E})$ where:*

- VARIABLES AND CONSTRAINTS. $|V| = O(k)$ *and* $|E| = O(k)$.
- RUNTIME. *The reduction runs in time* $\text{poly}(n, 2^{n \log k / k})$.
- ALPHABET. $\Sigma = \mathbb{F}_4^t$ *where* $t = O\left(\frac{n \log k}{k}\right)$.
- SATISFIABILITY. *G is satisfiable iff Γ is satisfiable.*

Proof. We first plug the 4-REGULAR 3-COLORING instance Γ from Theorem 3.2 into Lemma 5.1 and obtain a 2CSP instance $G' = (V', E', \mathbb{F}_4^t, \{\Pi_e'\}_{e \in E'})$. By the construction in Lemma 5.1, each $x \in V'$ corresponds to a set $S(x)$ of t variables in Γ. We fix an arbitrary order in $S(x)$ and use $x[i]$ to denote the i-th variable in $S(x)$. From the fact that Γ is 4-regular and the construction in Lemma 5.1, the produced 2CSP instance G' has the following properties:

- for each $\{x, y\} \in E'$ and $i \in [t]$, there are at most five sub-constraints between $x[i]$ and $\{y[j] : j \in [t]\}$: at most one equality check (Item 1 of Lemma 5.1) and at most four coloring checks (Item 2 of Lemma 5.1);
- the constraint graph of G' is 3-regular;
- $|V'| = k$, $|E'| = O(k)$, $t = O\left(\frac{n \log k}{k}\right)$, and the runtime is $\text{poly}(n, k, 4^t) = \text{poly}(n, 2^{n \log k / k})$;
- G' is satisfiable iff Γ is satisfiable.

25

part of a 4-Regular 3-Coloring instance Γ

a constraint in the constructed 2CSP instance G',
$S(x) = \{z_1, z_2, \ldots, z_5\}, S(y) = \{z_2, z_3, \ldots, z_6\}$

break the constraints into $O(1)$ matchings to get G'',
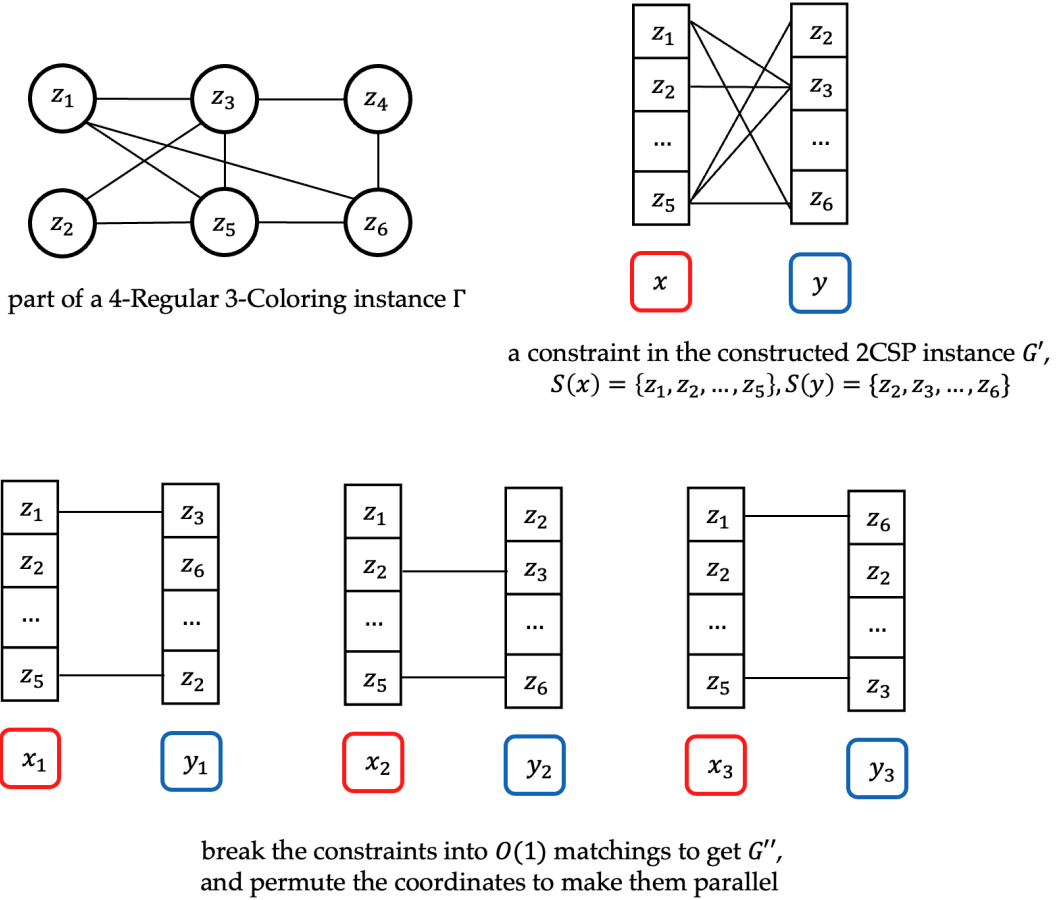and permute the coordinates to make them parallel

Figure 1: An example of G' and G'' and the permutation to parallelize sub-constraints.

Thus, we duplicate each $x \in V'$ into constant many copies, and distribute the sub-constraints in G' onto different copies. This produces another 2CSP instance $G'' = (V'', E'', \mathbb{F}_4^t, \{\Pi_e''\}_{e \in E''})$ where

- for each $e = \{u, v\} \in E''$, the constraint Π_e'' has exactly one type of sub-constraint (i.e., equality or coloring), which forms a partial (non-parallel) matching across the coordinates (i.e., $S(u), S(v)$) of u, v;

  (Note that the consistency checks among duplicates will be added later.)

- each variable in V'' is related to exactly one constraint;

- $|V''| = O(|V'|) = O(k)$ and $|E''| = |E'| = O(k)$.

The above procedure is efficient: we only need to perform matching decompositions for each $\{x, y\} \in E'$ separately for equality checks and coloring checks.

Before adding the consistency checks among duplicates, we first permute coordinates of each variable in V'' to parallelize the partial matchings. This is possible since each variable in V'' is related to exactly one constraint in G''. For a fixed $x \in V'$, let $x_1, \ldots, x_m \in V''$ be the duplicates of $x \in V'$. After the permutation, we add linear constraints between x_i and x_{i+1} for $1 \leq i < m$ to

26

check whether they are consistent (i.e., the correct permuted copies of each other). See Figure 1 for a streamlined presentation.

The construction of the VecCSP instance $G = (V, E, \Sigma = \mathbb{F}_4^t, \{\Pi_e\}_{e \in E})$ is completed after the permutation and adding the consistency checks among duplicated. The satisfiability is naturally preserved, $|V| = |V''| = O(k)$, and $|E| \leq |E''| + |V''| = O(k)$. In terms of Definition 5.3, the consistency checks are linear constraints and the constraints in G'' after permutation are parallel constraints. $\square$

5.2 From VecCSP to SVecCSP

Given Proposition 5.4, Theorem 4.4 follows directly by the following general reduction from VecCSP to SVecCSP.

Proposition 5.5 (VecCSP to SVecCSP). *There is a reduction algorithm such that the following holds. Given as input a VecCSP instance $G = (V, E, \mathbb{F}^t, \{\Pi_e\}_{e \in E})$, it produces an SVecCSP instance $G' = (V', E', \mathbb{F}^t, \{\Pi'_e\}_{e \in E'})$ where:*

- VARIABLES AND CONSTRAINTS. $|V'| = O(|V| + |E|)$ *and* $|E'| = O(|V| + |E|)$.
- RUNTIME. *The reduction runs in time* $\mathsf{poly}(|V|, |E|, |\mathbb{F}|^t)$.
- SATISFIABILITY. G' *is satisfiable iff G is satisfiable.*

Proof. Now that we get a VecCSP instance G, we show how to modify it to to obtain an SVecCSP instance G' that satisfies properties Item **(S3)** and Item **(S4)**. The construction consists of two steps and see Figure 2 for a streamlined presentation.

- First, from G, get another VecCSP instance $\widehat{G}$ which satisfies Item **(S3)**.
- Next, based on $\widehat{G}$, build a SVecCSP instance G' which in addition satisfies Item **(S4)**.

To satisfy Item **(S3)**, we split each variable x in V into a parallel variable x^{p} and a linear variable x^{l} in $\widehat{G}$ for parallel and linear constraints separately. Then we construct the constraints $\widehat{E}$ in $\widehat{G}$.

- For each linear constraint $e = \{x, y\} \in E$, we add the same linear constraint on $\{x^{\mathsf{l}}, y^{\mathsf{l}}\}$ in $\widehat{E}$.
- For each parallel constraint $e = \{x, y\} \in E$ with sub-constraint Π_e^{sub} and subset of coordinates Q_e, we add a parallel constraint on $\{x^{\mathsf{p}}, y^{\mathsf{p}}\}$ in $\widehat{E}$, which has Π_e^{sub} applied on *all* coordinates.
- We need to additionally check in $\widehat{G}$ the partial equality between x^{p} and x^{l} *only* on the subset of coordinates Q_e. Since each variable x is related to at most one parallel constraint as guaranteed in Definition 5.3, this additional check is well-defined.

  This check can be written as $M_{Q_e} x^{\mathsf{p}} = M_{Q_e} x^{\mathsf{l}}$, where M_{Q_e} is a matrix which projects on coordinates inside Q_e and zeroing out coordinates outside Q_e. To have the matrix only on one side, we introduce an additional variable x^{a} in $\widehat{G}$, and add two linear constraints $x^{\mathsf{a}} = M_{Q_e} x^{\mathsf{p}}$ and $x^{\mathsf{a}} = M_{Q_e} x^{\mathsf{l}}$ to $\widehat{E}$.

We first show that the above construction preserves the satisfiability as follows.

- Given a solution σ for the original G, assign $\sigma(x)$ to x^{l} and x^{a}, which satisfies all linear constraints in $\widehat{E}$. Then assign $\sigma(x)$ to x^{p} on the subset Q_e of coordinates, which satisfies all the

partial equality checks in $\widehat{E}$. Finally, assign arbitrary solution[7] of Π_e^{sub} to x^{p} on coordinates outside Q_e, which satisfies all the parallel constraints in $\widehat{E}$.

- Given a solution σ' of $\widehat{G}$, assign $\sigma'(x^{\mathsf{l}})$ to every x in G, which satisfies all linear constraints in E. Since $\sigma'(x^{\mathsf{p}})$ satisfies the parallel constraint in G' for all coordinates and the partial equality check guarantees consistency between $\sigma'(x^{\mathsf{l}})$ and $\sigma'(x^{\mathsf{p}})$ on the coordinates Q_e, the corresponding parallel constraints in E are satisfied as well.

Moreover, the variable set $\widehat{V} \subseteq \bigcup_{x \in V} \{x^{\mathsf{p}}, x^{\mathsf{l}}, x^{\mathsf{a}}\}$ has size $|\widehat{V}| = O(|V|)$ and the constraint set $\widehat{E}$ has size $|\widehat{E}| \le |E| + 2 \cdot |V| = O(|V| + |E|)$.

Now we construct G' from $\widehat{G}$ to satisfy Item **(S4)**. The final variable set of G' will be $V' = X \dot\cup Y$, which is constructed along with the constraint set $E' = E'_{\mathsf{l}} \dot\cup E'_{\mathsf{p}}$ as follows.

- Initialize X, Y as disjoint copies of $\widehat{V}$ and initialize $E'_{\mathsf{l}} = E'_{\mathsf{p}} = \varnothing$. For a variable $u \in \widehat{V}$, we denote as x_u, y_u its X-copy and Y-copy in V', respectively.

- For each $u \in \widehat{V}$, we add an equality, which is a linear constraint with the identity matrix, in E'_{l} between x_u and y_u.

  Note that this is consistent with Item **(S4)**.

- Then for each parallel constraint $e = \{x^{\mathsf{p}}, y^{\mathsf{p}}\} \in \widehat{E}$, we add the same constraint in E'_{p} on the X-copies of x^{p} and y^{p}.

- Finally for each linear constraint[8] $e = \{u, v\} \in \widehat{E}$ that checks $u = M_e v$, we add new variables x_e to X and y_e to Y. Then we impose a linear constraint $y_e = M_e x_e$ between them in E'_{l}, which is consistent with Item **(S4)**. We further add two equality constraints, which are identified as parallel constraints in E'_{p}, between x_e and x_u, as well as between y_e and x_v.

The construction of G' preserves the satisfiability of $\widehat{G}$ as all the duplicates in G' of variables in $\widehat{G}$ are connected by identity constraints. Moreover, $|X| = |Y| \le |\widehat{V}| + |\widehat{E}| = O(|V| + |E|)$ and $|E'| \le |\widehat{V}| + |\widehat{E}| + 3 \cdot |\widehat{E}| = O(|V| + |E|)$ as desired. $\qquad\square$

6 PCPP for Multi-Test Problem

In this section we prove Theorem 4.14 for DOUBLETEST and Theorem 4.20 for SINGLETEST. For convenience, we define the following multi-test problem (MULTITEST) which generalizes DOUBLETEST and SINGLETEST in a straightforward fashion.

Definition 6.1 (MULTITEST). Assume $\mathrm{char}(\mathbb{F}) = 2$. $(\mathbb{F}, m, d, t, u)$-MULTITEST is a pair language over $\Sigma_{\mathsf{x}} = \{0, 1\}, \Sigma_{\mathsf{y}} = \mathbb{F}^t$ consisting of all words in the form of $w = (C, T_1 \circ \cdots \circ T_u)$, where

- C is a Boolean circuit with $u \cdot |\mathbb{F}|^m \log |\mathbb{F}|$ input bits and $T_1, \ldots, T_u \in (\mathbb{F}^t)^{|\mathbb{F}|^m}$ are codewords of $\mathrm{RM}^{\mathbb{F}, m, d, t}$;

- if we view $\mathbb{F}$ as $\{0, 1\}^{\log |\mathbb{F}|}$, $T_1 \circ \cdots \circ T_u$ parallel satisfies C.

---

[7]Technically it is possible that Π_e^{sub} is not satisfiable. If $Q_e = \varnothing$, then we can simply replace Π_e^{sub} by any satisfiable sub-constraint. If otherwise $Q_e \ne \varnothing$, then the original G is not satisfiable and $\widehat{G}$ is also not satisfiable. Therefore the construction still works.

[8]In particular, $e = \{u, v\}$ can be $\{x^{\mathsf{l}}, y^{\mathsf{l}}\}$ or $\{x^{\mathsf{l}}, x^{\mathsf{a}}\}$ or $\{x^{\mathsf{p}}, x^{\mathsf{a}}\}$.
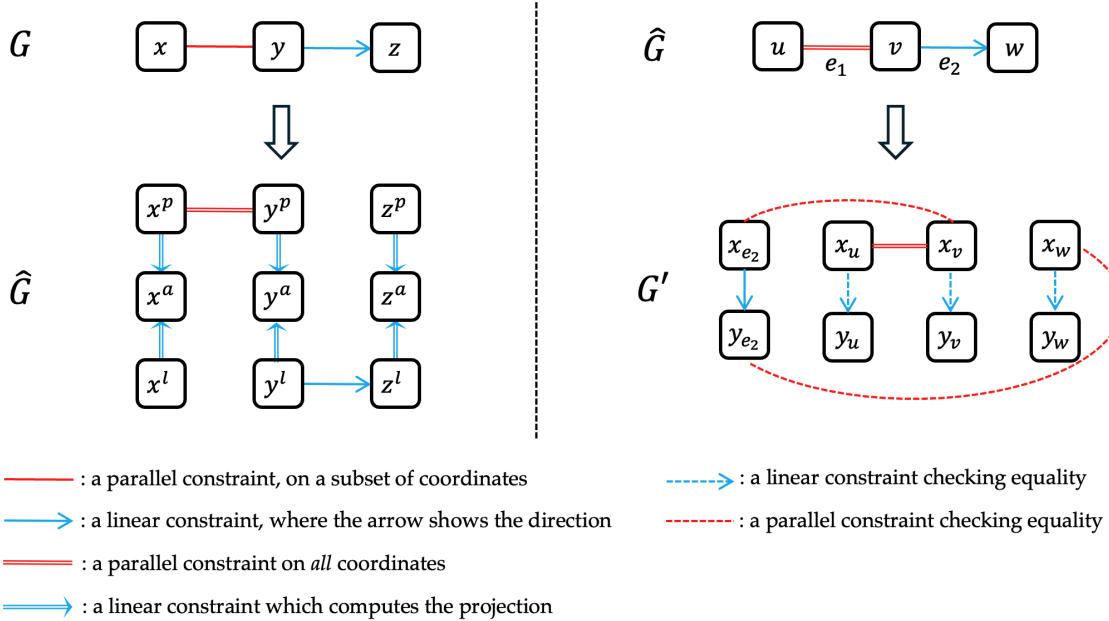
Figure 2: An illustration of the reduction from G to $\widehat{G}$, and from $\widehat{G}$ to G'.

We define the input length $|w|$ to be the size of C plus $u \cdot |\mathbb{F}|^m \log |\mathbb{F}|$.

By setting $u = 1$ or $u = 2$, we immediately obtain $(\mathbb{F}, m, d, t)$-SINGLETEST or $(\mathbb{F}, m, d, t)$-DOUBLETEST. Hence Theorem 4.20 and Theorem 4.14 follows directly from the following result for MULTITEST.

Theorem 6.2 (PCPP for MULTITEST). *Assume* $\mathsf{char}(\mathbb{F}) = 2$ *and* $|\mathbb{F}| \geq \max\{6md, 2^{100}m\log|\mathbb{F}|\}$. *Assume* $u \leq 2^{50}$ *is a positive integer. Then for any* $\delta \in [0,1]$, $(\mathbb{F}, d, m, t, u)$-MULTITEST *has*

$$an \left(O(1), \log|w| + O\left(\log^{0.1}|w| + \log|\mathbb{F}|\right), \delta, \Omega(\delta), \Sigma^t\right)\text{-PCPP verifer } \mathcal{P}_{\mathsf{mt}},$$

where $|w|$ *is the input length of* $(\mathbb{F}, d, m, t, u)$-MULTITEST *and* $\Sigma = \mathbb{F}^{d+1}$.

Proof sketch. Let $(C, T_1 \circ \cdots \circ T_u)$ be an input for $(\mathbb{F}, d, m, t, u)$-MULTITEST. Our goal is to check whether $T_1 \circ \cdots \circ T_u$ is δ-close to some $T_1^* \circ \cdots \circ T_u^* \in$ MULTITEST(C), i.e., the restriction of the pair language MULTITEST on C. In other words, the following two conditions hold:

(C1) for each $j \in [u]$, $T_j^* \in \mathbb{F}^{|\mathbb{F}|^m}$ is the truth table of a polynomial of parallel degree d;

(C2) $T_1^* \circ \cdots \circ T_u^*$, viewed as a word in $\{0,1\}^{u|\mathbb{F}|^m \log|\mathbb{F}|}$, parallel satisfies the Boolean circuit C.

To guarantee Item (C1), we use the PCPP verifier $\mathcal{P}_{\mathsf{ldt}}$ from the codeword testing of $\mathrm{RM}^{\mathbb{F},m,d,t}$ (see Theorem 3.6). Given $T_1^* \circ \cdots \circ T_u^*$ satisfying Item (C1), we aim to test that it also satisfies Item (C2). That is, for each fixed $i \in [t]$, $T_1^*[i] \circ \cdots \circ T_u^*[i]$ satisfies the Boolean circuit C.

To this end, we will use the PCPP verifier $\mathcal{P}_{\mathsf{ckt}}$ of CKTVAL (see Theorem 3.13). This alone, however, is not sufficient as $\mathcal{P}_{\mathsf{ckt}}$ cannot rule out the case that changing $o(1)$ fraction of entries in $T_1^*[i] \circ \cdots \circ T_u^*[i]$ satisfying the circuit C. To fix this issue, we have to exploit the fact that each $T_j^*[i]$

29

is supposed to be the truth table of a degree-d polynomial, which, by Schwartz-Zippel lemma and the fact that u is a constant, forbids such attacks. As a result, we need to incorporate the codeword testing circuit (see Theorem 3.7) to enforce the low degree condition.

Unfortunately, this still does not work due to a subtle alphabet mismatch: the codeword testing works over $\mathbb{F}$ but Item (C2) needs to flat $\mathbb{F}$ as $\{0,1\}^{\log |\mathbb{F}|}$. Therefore, the distance guaranteed by the low degree condition can be dilated by a worst-case factor of $\log |\mathbb{F}| = \omega(1)$ after converting $\mathbb{F}$ to $\{0,1\}^{\log |\mathbb{F}|}$, for which reason the mentioned attack can still be carried out. To address this issue, we employ a standard approach to lift the conversion of $\mathbb{F}$ via error correcting codes [BGH$^+$06, ALM$^+$01, Din07]. More formally, after flattening $\mathbb{F}$ as $\{0,1\}^{\log |\mathbb{F}|}$, we take it through an error correcting code with constant rate and distance, which produces a codeword in $\{0,1\}^{O(\log |\mathbb{F}|)}$ and, more importantly, has constant relative distance against other codewords.

In summary, to handle Item (C2), we need to use $\mathcal{P}_{\mathsf{ckt}}$ to check the validity of the combination of (1) the original circuit C, (2) the codeword testing procedure, and (3) the error correcting lifting. This is parallel for each coordinate and is presented in Subsection 6.1. Then in Subsection 6.2, we put together the argument for Item (C1) and prove Theorem 6.2.

Remark 6.3. One may wonder the necessity of using a separate codeword testing for Item (C1), as we anyway need to use it for Item (C2). The difference lies in the proximity: the former uses Theorem 3.6 which guarantees the parallel proximity (i.e., including all the dimensions $[t]$), whereas the latter uses Theorem 3.7 which only implies coordinate-wise proximity (i.e., individually for each coordinate $i \in [t]$).

Without the former, we can only get an $\Omega(t \cdot \delta)$ final proximity via a union bound. Upgrading the latter needs to generalize the construction of $\mathcal{P}_{\mathsf{ckt}}$ to the parallel setting, which arguably requires more work. Hence we stick to our current presentation for simplicity.

6.1 Single-Coordinate Checking Circuit

As sketched above, we construct another Boolean circuit C' for Item (C2), which augments the original circuit C with an alphabet lifting via error correcting code and a low-degree check. Later, this single-coordinate checking circuit will be applied in parallel for every coordinate.

Lifted Flattening of $\mathbb{F}$. We will use the following standard error correcting code to achieve such an alphabet lifting.

Proposition 6.4 (Binary ECC with Constant Rate and Distance [Jus72]). *For every $n \geq 1$, there exists an efficiently computable error correcting code with the encoding map $\mathrm{ECC}_n : \{0,1\}^n \to \{0,1\}^{7n}$ of relative distance $\delta(\mathrm{ECC_n}) \geq 0.01$.*

Let $\mathrm{Enc} : \mathbb{F} \to \{0,1\}^{7 \log |\mathbb{F}|}$ to be $\mathrm{ECC}_{\log |\mathbb{F}|}$ composed with the natural flattening of $\mathbb{F}$ into $\{0,1\}^{\log |\mathbb{F}|}$. Let Dec be the corresponding decoding function.

Later putting back into the construction of $\mathcal{P}_{\mathsf{mt}}$, we will apply Enc to every entry of $T_1[i] \circ \cdots \circ T_u[i] \in \mathbb{F}^{u|\mathbb{F}|^m}$ for every coordinate $i \in [t]$. For convenience, we define the parallel encoding map $\mathrm{Enc}^{\odot} : \mathbb{F}^{|\mathbb{F}|^m} \to \{0,1\}^{7|\mathbb{F}|^m \log |\mathbb{F}|}$ as

$$\mathrm{Enc}^{\odot}(z)(k, \cdot) = \mathrm{Enc}(z_k) \quad \text{for } z \in \mathbb{F}^{|\mathbb{F}|^m} \text{ and } k \in [|\mathbb{F}|^m], \tag{5}$$

where we view $\mathrm{Enc}^{\odot}(z) \in \{0,1\}^{7|\mathbb{F}| \log |\mathbb{F}|}$ as a function $[|\mathbb{F}|^m] \times [7 \log |\mathbb{F}|] \to \{0,1\}$ and the index k points to the lifted flattening of the k-th entry of z. Given the definition of $\mathrm{Enc}^{\odot}$, the lifted flattening of $T_1[i] \circ \cdots \circ T_u[i]$ can be simply represented as $\mathrm{Enc}^{\odot}(T_1[i]) \circ \cdots \circ \mathrm{Enc}^{\odot}(T_u[i])$.

The Construction of C'. Given the lifted flattening of alphabet, we now present the construction. The circuit C' takes as input $y_1 \circ \cdots \circ y_u$ where each $y_j \in \{0,1\}^{7|\mathbb{F}|^m \log |\mathbb{F}|}$ is supposed to be $\mathrm{Enc}^{\odot}(\widehat{y}_j)$ for some $\widehat{y}_j \in \mathbb{F}^{|\mathbb{F}|^m}$. Note that $\widehat{y}_j$ will be $T_j[i]$, rolling over all coordinates $i \in [t]$. The circuit C' check the following three things in order:

(S1) Check if each y_j is a codeword of $\mathrm{Enc}^{\odot}$ and, if so, compute each $\widehat{y}_j$ and view it as a binary string via the standard flattening of $\mathbb{F}$.

   This requires to decode a total number of $u \cdot |\mathbb{F}|^m$ words in $\{0,1\}^{7 \log |\mathbb{F}|}$, which can be has circuit size $u|\mathbb{F}|^m \cdot \mathrm{polylog}|\mathbb{F}| \leq |\mathbb{F}|^m \mathrm{poly}|\mathbb{F}|$.

(S2) Check if each $\widehat{y}_i$ satisfies C_{ldt} from Theorem 3.7, i.e., $\widehat{y}_i$ is the truth table of a degree-d polynomial.

   This has size $|\mathbb{F}|^m \mathrm{poly}|\mathbb{F}|$ by Theorem 3.7.

(S3) Check whether $\widehat{y}_1 \circ \cdots \circ \widehat{y}_u$ satisfies the original circuit C.

   This has size precisely the size of C.

   Now we list properties of C'.

Fact 6.5 (Satisfiability). *If $\mathrm{Enc}^{\odot}(\widehat{y}_1) \circ \cdots \circ \mathrm{Enc}^{\odot}(\widehat{y}_u)$ satisfies the circuit C', then $\widehat{y}_1 \circ \cdots \circ \widehat{y}_u$ satisfies the circuit C.*

Fact 6.6 (Size). *Recall from Definition 6.1 that $|w|$ is the input length of $(\mathbb{F}, m, d, t)$-MULTITEST, which equals the size of C plus $u \cdot |\mathbb{F}|^m \log |\mathbb{F}|$. The size of the circuit C' is at most $|w| \cdot \mathrm{poly}|\mathbb{F}|$ and the input of C' has length $O(|w|)$.*

Claim 6.7 (Distance). *If $y_1 \circ \cdots \circ y_u$ passes Items (S1) and (S2) but not Item (S3), then it is 2^{-60}-far from solutions of C'.*

Proof. Let $z_1 \circ \cdots \circ z_u$ be an arbitrary solution of C', which means it passes Items (S1) to (S3). Let $\widehat{z}_1, \ldots, \widehat{z}_u$ be the decoding outcome from Item (S1). Then there exists $j \in [u]$ such that $\widehat{y}_j \neq \widehat{z}_j$. Since they both pass Item (S2), $\widehat{y}_j$ and $\widehat{z}_j$, viewed as an element from $\mathbb{F}^{|\mathbb{F}|^m}$, correspond to truth tables of distinct degree-d polynomials. Hence $\Delta(\widehat{y}_j, \widehat{z}_j) \geq 1 - \frac{d}{|\mathbb{F}|} > \frac{1}{2}$ by Schwartz-Zippel lemma and our assumption on $|\mathbb{F}|$. Recall from (5) that $\mathrm{Enc}^{\odot}$ applies Enc entrywise. Since $y_j = \mathrm{Enc}^{\odot}(\widehat{y}_j)$ and $z_j = \mathrm{Enc}^{\odot}(\widehat{z}_j)$, we now have $\Delta(y_j, z_j) > 0.01 \cdot \frac{1}{2}$ by Proposition 6.4. Since $u \leq 2^{50}$, we have

$$\Delta(y_1 \circ \cdots \circ y_u, z_1 \circ \cdots \circ z_u) \geq \frac{\Delta(y_j, z_j)}{u} > 2^{-60}$$

as desired. $\qquad\square$

6.2 Combining Codeword Testing

Now we are ready to combine the codeword testing for Item (C1) and the single-coordinate checking circuit C' for Item (C2) to prove Theorem 6.2.

Proof of Theorem 6.2. The auxiliary proof for $\mathcal{P}_{\mathsf{mt}}$ consists of two parts.

- The first part is $\pi_{\mathsf{ldt},1}, \ldots, \pi_{\mathsf{ldt},u}$. Each $\pi_{\mathsf{ldt},i}$ has alphabet Σ^t and is constructed by Theorem 3.6, which is supposed to be the auxillary proof for codeword testing of $\mathrm{RM}^{\mathbb{F},m,d,t}$ on T_i.

- The second part is denoted by π_{ckt}, which has the alphabet $\{0,1\}^t$ naturally embedded into Σ^t. For each coordinate $i \in [t]$, $\pi_{\mathsf{ckt}}[i]$ is constructed by Theorem 3.13 for $\mathcal{P}_{\mathsf{ckt}}$ to check if $\mathrm{Enc}^{\odot}(T_1[i]) \circ \cdots \circ \mathrm{Enc}^{\odot}(T_u[i])$ satisfies the circuit C'.

Testing Procedure of $\mathcal{P}_{\mathsf{mt}}$. Now we describe the testing procedure. $\mathcal{P}_{\mathsf{mt}}$ executes one of the following two tests with equal probability.

- For each $i \in [u]$, $\mathcal{P}_{\mathsf{mt}}$ invokes $\mathcal{P}_{\mathsf{ldt}}$ to run the codeword testing for $\mathsf{RM}^{\mathbb{F},m,d,t}$ on $T_i \circ \pi_{\mathsf{ldt},i}$. This checks whether $T_i \in \mathrm{Im}(\mathsf{RM}^{\mathbb{F},m,d,t})$.

- $\mathcal{P}_{\mathsf{mt}}$ parallel simulates $\mathcal{P}_{\mathsf{ckt}}$ to test if $\mathrm{Enc}^{\odot}(T_1[i]) \circ \cdots \circ \mathrm{Enc}^{\odot}(T_u[i]) \circ \pi_{\mathsf{ckt}}[i]$ satisfies C' for all coordinates $i \in [t]$.

  In detail, for each coordinate $i \in [t]$, $\mathcal{P}_{\mathsf{mt}}$ tosses random coins as $\mathcal{P}_{\mathsf{ckt}}$ does, and probes entries of $\pi_{\mathsf{ckt}}[i]$ if needed. Whenever $\mathcal{P}_{\mathsf{ckt}}$ needs to probe some bit of $\mathrm{Enc}^{\odot}(T_j[i])$, $\mathcal{P}_{\mathsf{mt}}$ queries the corresponding entry (i.e., the index k in (5)) of $T_j[i]$, performs the lifted flattening of $\mathbb{F}$ for that entry, and obtains the desired bit.

  We emphasize that the randomness used to simulate $\mathcal{P}_{\mathsf{ckt}}$ is the same for all coordinates. Therefore, the queries by $\mathcal{P}_{\mathsf{ckt}}$ are simulated in parallel for all coordinates $i \in [t]$, as the query locations are uniquely determined by the randomness.

Parameters of $\mathcal{P}_{\mathsf{mt}}$. Since u is a constant and both $\mathcal{P}_{\mathsf{ldt}}, \mathcal{P}_{\mathsf{ckt}}$ have constant queries (see Theorem 3.6 and Theorem 3.13), $\mathcal{P}_{\mathsf{mt}}$ makes constant queries.

Recall that the input length $|w|$ equals the size of C plus $u|\mathbb{F}|^m \log |\mathbb{F}|$. By Theorem 3.6, the first part tosses

$$m \log |\mathbb{F}| + O(\log\log |\mathbb{F}| + \log m) \leq \log |w| + O(\log m) \leq \log |w| + O(\log |\mathbb{F}|)$$

coins, where we used the assumption on $|\mathbb{F}|$. By Theorem 3.13 and Fact 6.6, the second part tosses

$$\log(|w| \cdot \mathsf{poly}|\mathbb{F}|) + O\left(\log^{0.1}(|w| \cdot \mathsf{poly}|\mathbb{F}|)\right) \leq \log |w| + O\left(\log^{0.1}|w| + \log |\mathbb{F}|\right)$$

coins. Since we only execute one of them, the number of random coins is the maximum of the above two as desired.

Completeness and Soundness. The completeness is straightforward by the completeness of $\mathcal{P}_{\mathsf{ldt}}$ (see Theorem 3.6) and $\mathcal{P}_{\mathsf{ckt}}$ (see Theorem 3.13) and the construction of C' (see Subsection 6.1). We focus on the soundness analysis: assuming that $T_1 \circ \cdots \circ T_u$ is δ-far from $\mathrm{MULTITEST}(C)$ (i.e., we do not have Items (C1) and (C2)), $\mathcal{P}_{\mathsf{mt}}$ rejects with probability $\Omega(\delta)$. By modifying the hidden constant in $\Omega(\cdot)$ and noticing that δ-far implies δ'-far for any $\delta' \leq \delta$, we additionally assume $\delta \leq 2^{-100}$.

Assume towards contradiction that the above soundness statement is false. We first show that each T_j is close to being parallel degree-d. This comes from the following Fact 6.8, which can be deduced directly from Theorem 3.6.

Fact 6.8. *If for some $j \in [u]$, T_j is δ-far from $\mathrm{Im}(\mathsf{RM}^{\mathbb{F},m,d,t})$, then $\mathcal{P}_{\mathsf{mt}}$ rejects with probability $\Omega(\delta)$.*

Now we assume each T_j is δ-close to some $T_j^* \in \mathrm{Im}(\mathsf{RM}^{\mathbb{F},m,d,t})$, which corresponds to Item (C1). Next, we show $T_1^* \circ \cdots \circ T_u^*$ parallel satisfies C, which corresponds to Item (C2). By Fact 6.5, it suffices to show for each coordinate $i \in [t]$ that $\mathrm{Enc}^{\odot}(T_1^*[i]) \circ \cdots \circ \mathrm{Enc}^{\odot}(T_u^*[i])$ satisfies C', which is precisely the following Claim 6.9.

Claim 6.9. For each coordinate $i \in [t]$, $\mathrm{Enc}^\odot(T_1^*[i]) \circ \cdots \circ \mathrm{Enc}^\odot(T_u^*[i])$ satisfies Items (S1) to (S3).

Given Claim 6.9, we arrive at a contradiction and complete the soundness analysis. $\qquad\square$

Finally we prove Claim 6.9.

Proof of Claim 6.9. Note that Item (S1) is already satisfied. Since each $T_j^* \in \mathrm{Im}(\mathsf{RM}^{\mathbb{F},m,d,t})$, each $T_j^*[i]$ is the truth table of a degree-d polynomial, which means Item (S2) is also satisfied. If Item (S3) is false for some $i \in [t]$, then by Claim 6.7, $\mathrm{Enc}^\odot(T_1^*[i]) \circ \cdots \circ \mathrm{Enc}^\odot(T_u^*[i])$ is 2^{-60}-far from any solution of C'. Observe that

$$\Delta(\mathrm{Enc}^\odot(T_1[i]) \circ \cdots \circ T_u[i], \mathrm{Enc}^\odot(T_1^*[i]) \circ \cdots \circ T_u^*[i]) \leq \Delta(\mathrm{Enc}^\odot(T_1) \circ \cdots \circ T_u, \mathrm{Enc}^\odot(T_1^*) \circ \cdots \circ T_u^*)$$
$$\leq \max_{j \in [u]} \Delta(\mathrm{Enc}^\odot(T_j), \mathrm{Enc}^\odot(T_j^*))$$
$$\text{(since } \Delta \text{ is relative distance)}$$
$$\leq \delta. \qquad \text{(by the choice of } T_j^*)$$

Since $\delta \leq 2^{-100}$, we know that $\mathrm{Enc}^\odot(T_1[i]) \circ \cdots \circ T_u[i]$ is 2^{-70}-far from solutions of C'. By Theorem 3.13, this means $\mathcal{P}_{\mathsf{mt}}$, which executes $\mathcal{P}_{\mathsf{ckt}}$ with half probability, rejects with probability $\frac{1}{2} \cdot \frac{1}{2} = \Omega(\delta)$. Recall that we assumed that $\mathcal{P}_{\mathsf{mt}}$ does not reject with probability $\Omega(\delta)$, which is a contradiction. Thus Item (S3) should also be satisfied and this completes the proof of Claim 6.9. $\qquad\square$

Acknowledgement

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, USA, 1st edition, 2009. 6

[ABSS97] Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997. 4

[AFWZ95] Noga Alon, Uriel Feige, Avi Wigderson, and David Zuckerman. Derandomized graph products. *Comput. Complex.*, 5(1):60–75, 1995. 4

[ALM⁺01] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45:501–555, 09 2001. 16, 30

[App17] Benny Applebaum. Exponentially-hard gap-csp and local prg via local hardcore functions. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 836–847, 2017. 3

[AR94] Noga Alon and Yuval Roichman. Random Cayley graphs and expanders. *Random Structures & Algorithms*, 5(2):271–284, 1994. 38

[BEKP13] Edouard Bonnet, Bruno Escoffier, Eun Jung Kim, and Vangelis Th. Paschos. On subexponential and fpt-time inapproximability. In Gregory Gutin and Stefan Szeider, editors, *Parameterized and Exact Computation*, pages 54–65, Cham, 2013. Springer International Publishing. 8

[BGH$^+$06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM Journal on Computing*, 36(4):889–974, 2006. 8, 13, 14, 23, 30

[BSSVW03] Eli Ben-Sasson, Madhu Sudan, Salil Vadhan, and Avi Wigderson. Randomness-efficient low degree tests and short pcps via epsilon-biased sets. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 612–621, New York, NY, USA, 2003. Association for Computing Machinery. 8, 33, 36, 38, 39, 41, 43, 45

[CAGK$^+$19] Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT Approximations for k-Median and k-Means. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 42:1–42:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 5

[CCK$^+$17] Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-ETH to FPT-inapproximability: Clique, dominating set, and more. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 743–754. IEEE Computer Society, 2017. 3, 4, 8

[CFG$^+$16] Marek Cygan, Fedor V. Fomin, Alexander Golovnev, Alexander S. Kulikov, Ivan Mihajlin, Jakub Pachocki, and Arkadiusz Socala. Tight bounds for graph homomorphism and subgraph isomorphism. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1643–1649. SIAM, 2016. 10

[CFLL23] Yijia Chen, Yi Feng, Bundit Laekhanukit, and Yanlin Liu. Simple combinatorial construction of the $k^{o(1)}$-lower bound for approximating the parameterized k-clique. *CoRR*, abs/2304.07516, 2023. 3

[CHKX06] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006. 3

[Din07] Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12–es, jun 2007. 6, 8, 30

[Fei98] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998. 5

[FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. 3, 10

[FGL+96] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, 1996. 4

[FS95]   Katalin Friedl and Madhu Sudan. Some improvements to total degree tests. In *Proceedings Third Israel Symposium on the Theory of Computing and Systems*, pages 190–198. IEEE, 1995. 36, 45

[FSLM20] Andreas Emil Feldmann, Karthik C. S., Euiwoong Lee, and Pasin Manurangsi. A survey on approximation in parameterized complexity: Hardness and algorithms. *Algorithms*, 13(6), 2020. 8

[GLR+23] Venkatesan Guruswami, Bingkai Lin, Xuandi Ren, Yican Sun, and Kewen Wu. Parameterized inapproximability hypothesis under ETH. *CoRR*, abs/2311.16587, 2023. 4, 6, 7, 8, 14, 15, 16, 24, 25

[GRS23]   Venkatesan Guruswami, Xuandi Ren, and Sai Sandeep. Baby PIH: Parameterized inapproximability of Min CSP. *arXiv preprint arXiv:2310.16344*, 2023. 5

[Hoc97]   Dorit S. Hochbaum. Approximation algorithms for NP-hard problems. *SIGACT News*, 28(2):40–52, jun 1997. 5

[IP01]   Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62:367–375, 2001. 9, 24

[IPZ01]   Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. 24

[JM21]   Akhil Jalan and Dana Moshkovitz. Near-optimal Cayley expanders for abelian groups. In *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, 2021. 38

[Jus72]   Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Trans. Inf. Theory*, 18:652–656, 1972. 30

[KK22]   Karthik C. S. and Subhash Khot. Almost polynomial factor inapproximability for parameterized k-clique. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 6:1–6:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. 3

[KMPS23] Karthik C. S., Dániel Marx, Marcin Pilipczuk, and Uéverton S. Souza. Conditional lower bounds for sparse parameterized 2-CSP: A streamlined proof. *CoRR*, abs/2311.05913, 2023. 7, 16, 24

[KT00]   Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, STOC '00, page 80–86, New York, NY, USA, 2000. Association for Computing Machinery. 3

[Lin21]   Bingkai Lin. Constant approximating *k*-clique is W[1]-hard. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1749–1756. ACM, 2021. 3

[LRSW22] Bingkai Lin, Xuandi Ren, Yican Sun, and Xiuhan Wang. On lower bounds of approximating parameterized k-clique. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 90:1–90:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. 3, 4

[LRSW23] Bingkai Lin, Xuandi Ren, Yican Sun, and Xiuhan Wang. Improved hardness of approximating k-clique under ETH. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 285–306. IEEE, 2023. 3, 36, 37, 38, 45

[LRSZ20] Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Parameterized complexity and approximability of directed odd cycle transversal. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2181–2200. SIAM, 2020. 4, 11

[Man20] Pasin Manurangsi. Tight running time lower bounds for strong inapproximability of maximum *k*-coverage, unique set cover and related problems (via *t*-wise agreement testing theorem). In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 62–81. SIAM, 2020. 4, 5

[Mar10] Dániel Marx. Can you beat treewidth? *Theory Comput.*, 6(1):85–112, 2010. 7, 16, 23, 24

[PS94] Alexander Polishchuk and Daniel A Spielman. Nearly-linear size holographic proofs. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 194–203, 1994. 45, 46

[Rao11] Anup Rao. Parallel repetition in projection games and a concentration bound. *SIAM Journal on Computing*, 40(6):1871–1891, 2011. 4

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996. 37, 41, 48, 49

[Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, oct 1980. 8

A Derandomized Parallel Low Degree Test

In this section, we design a derandomized parallel low degree test to prove Theorem 3.6 and Theorem 3.7. This is obtained by combining the derandomized low degree test [BSSVW03, Theorem 4.1] with the parallel low degree test [LRSW23, Lemma 7.7], where the latter builds on [FS95]. While the combination is standard, we decide to expand the proof sketch in [BSSVW03] to a full proof for completeness.

Theorem (Theorem 3.6 Restated)**.** *Assume* $\mathsf{char}(\mathbb{F}) = 2$ *and* $|\mathbb{F}| \geq \max\{6md, 2^{100}m\log|\mathbb{F}|\}$. *Let* $\Sigma = \mathbb{F}^{d+1}$ *be the set of univariate degree-d polynomials over* $\mathbb{F}$. *There exists an efficient verifier* $\mathcal{P}_{\mathsf{ldt}}$ *with the following properties.*

- *The input of $\mathcal{P}_{\mathsf{ldt}}$ is $T \circ \pi$, where $T \in (\mathbb{F}^t)^{|\mathbb{F}|^m}$ is supposed to be a codeword of $\mathsf{RM}^{\mathbb{F},m,d,t}$ and $\pi \in (\Sigma^t)^{|\mathbb{F}|^m \cdot (m \log |\mathbb{F}|)^{O(1)}}$ is the auxiliary proof.*

- *$\mathcal{P}_{\mathsf{ldt}}$ tosses $m \log |\mathbb{F}| + O(\log \log |\mathbb{F}| + \log m)$ unbiased coins and makes 2 queries on $T \circ \pi$.*

- *If $T \in \mathrm{Im}(\mathsf{RM}^{\mathbb{F},m,d,t})$, then there exists some π such that $\mathcal{P}_{\mathsf{ldt}}(T \circ \pi)$ always accepts.*

- *If T is δ-far from $\mathrm{Im}(\mathsf{RM}^{\mathbb{F},m,d,t})$, then $\mathbf{Pr}[\mathcal{P}_{\mathsf{ldt}}(T \circ \pi) \text{ rejects}] \geq 2^{-40}\delta$ for any π.*

Theorem (Theorem 3.7 Restated). *Assume $\mathsf{char}(\mathbb{F}) = 2$ and $|\mathbb{F}| \geq \max\{6md, 2^{100}m \log |\mathbb{F}|\}$. There exists a Boolean circuit C_{ldt} of size $|\mathbb{F}|^m \mathsf{poly}|\mathbb{F}|$ for $T \in (\mathbb{F}^t)^{|\mathbb{F}|^m}$, where we encode $\mathbb{F}$ as $\{0,1\}^{\log|\mathbb{F}|}$, such that T is codeword of $\mathsf{RM}^{\mathbb{F},m,d,t}$ iff T parallel satisfies C_{ldt}.*

A.1 Extra Notation

We first set up some necessary notation.

Parallel Low Degree Polynoimal. We first recall the notion of parallel polynomial from Section 3. Let $\mathbb{F}$ be a finite field. For a parallel-output function $f \colon \mathbb{F}^m \to \mathbb{F}^t$, we denote $f[1], \ldots, f[t] \colon \mathbb{F}^m \to \mathbb{F}$ as its single-output components, i.e., $f(x) = (f[1](x), \ldots, f[t](x))$. We aim to test if $f[1], \ldots, f[t]$ are consistent with degree-d polynomials on a large common set of inputs. Formally, we say f is δ-close to parallel degree-d iff f is δ-close to $\mathsf{RM}^{\mathbb{F},m,d,t}$; and we say f is parallel degree-d if $\delta = 0$.

A simple union bound shows that if each $f[i]$ is δ-close to degree-d (e.g., from the standard low degree test), then $f = (f[1], \ldots, f[t])$ is $t\delta$-close to parallel degree-d. However for our purposes, such a loss is not affordable since t is typically large. Therefore we need to open-box the specific low degree test and show that it implies consistency on $1 - \delta$ common fraction of inputs simultaneously for all $f[i]$.

Parallel Low Degree Test. For $x, y \in \mathbb{F}^m$, the line crossing x in direction y is the set

$$\ell_{x,y} := \{x + t \cdot y \colon t \in \mathbb{F}\}.$$

Note that if $y = 0^m$ then $\ell_{x,y} = \{x\}$, otherwise $\ell_{x,y}$ has $|\mathbb{F}|$ points. Let $\mathbb{L}$ be the set of lines in $\mathbb{F}^m$. For a parallel function $f \colon \mathbb{F}^m \to \mathbb{F}^t$ and any $\ell \in \mathbb{L}$, we use $f|_\ell \colon \ell \to \mathbb{F}^t$ to denote the restriction of f on the line ℓ.

Let $\mathbb{F}^{(d,t)}$ be the set of univariate parallel degree-d polynomials, i.e.,

$$\mathbb{F}^{(d,t)} := \left\{h \colon \mathbb{F} \to \mathbb{F}^t \colon h_i \text{ is a univariate degree-}d \text{ polynomial for each } i \in [t]\right\}.$$

The standard low degree test $\mathsf{LDTest}^{f,g}$ [RS96], translated to the parallel setting [LRSW23], is given oracle access to $f \colon \mathbb{F}^m \to \mathbb{F}^t$ and $g \colon \mathbb{L} \to \mathbb{F}^{(d,t)}$, which later tests their consistency[9]. For each line $\ell \in \mathbb{L}$, we denote $g(\ell) \in \mathbb{F}^{(d,t)}$ as its parallel line polynomial, which we interpret as a univariate parallel degree-d polynomial mapping elements in ℓ to $\mathbb{F}^t$. Thus for a line ℓ and a point $z \in \ell$, $g(\ell)(z) \in \mathbb{F}^t$ is well defined. We say f agrees with $g(\ell)$ on z if $f(z) = g(\ell)(z)$.

For each $f \colon \mathbb{F}^m \to \mathbb{F}^t$ and $d \in \mathbb{N}$, we use $f_{\mathbb{L}} \colon \mathbb{L} \to \mathbb{F}^{(d,t)}$ to denote the restriction of f on each line to its closest parallel degree-d polynomial. That is, for each line ℓ, we set $f_{\mathbb{L}}(\ell)$ to be the parallel degree-d polynomial closest to $f|_\ell$, where we break tie arbitrarily. Note that we will always assume the degree to be tested is d, and thus we omit d in defining $f_{\mathbb{L}}$ for simplicity.

---

[9]The term consistency has been called correlated agreement, e.g., in recent literature of proximity gaps for RS codes and related literature in FRI protocol/SNARKs.

In the completeness case (i.e., f is indeed parallel degree-d), we can pick $f_{\mathbb{L}}(\ell) = f|_\ell$ which is also parallel degree-d. The parallel low degree test $\mathsf{LDTest}^{f,g}$ explores the reverse direction: independently select $x, y \sim \mathbb{F}^m$ and accept iff $f(x) = g(\ell_{x,y})(x)$, i.e., f agrees with the parallel line polynomial $g(\ell_{x,y})$ on point x. The parallel low degree test [LRSW23] shows that if $\mathsf{LDTest}^{f,g}$ accepts with probability $1 - \delta$, then f is $O(\delta)$-close to parallel degree-d. Note that this bound does not depend on t.

Derandomized Parallel Low Degree Test. Now we introduce the derandomized version of the parallel low degree test. Following [BSSVW03], this simply replaces the uniformly random direction y by a pseudorandom $y \sim S$ for a much smaller set $S \subseteq \mathbb{F}^m$. Hence the number of randomness drops from $|\mathbb{F}|^{2m}$ to $|S| \cdot |\mathbb{F}|^m$.

Definition A.1 (Derandomized Parallel Low Degree Test). Let $S \subseteq \mathbb{F}^m$ and $f: \mathbb{F}^m \to \mathbb{F}^t, g: \mathbb{L} \to \mathbb{F}^{(d,t)}$.[10] The derandomized parallel low degree test $\mathsf{LDTest}_S^{f,g}$ is executed as follows: independently select $x \sim \mathbb{F}^m$ and $y \sim S$, then accept iff $f(x) = g(\ell_{x,y})(x)$.

Later the set S is chosen to be a λ-biased set as in [BSSVW03].

Definition A.2 (λ-Biased Set). $S \subseteq \mathbb{F}^m$ is a λ-biased set iff

- S is symmetric, i.e., if $y \in S$ then $-y \in S$;[11]
- $\left| \mathbb{E}_{y \sim S} [\chi(y)] \right| \leq \lambda$ holds for any non-trivial homomorphism[12] $\chi: \mathbb{F}^m \to \mu_p$, where μ is the multiplicative group of p-th unit root and p is the characteristic of $\mathbb{F}$.

In a graph theoretical reformulation, S is λ-biased iff the graph G_S is an (undirected) expander graph with expansion factor $1 - \lambda$, where the vertex set of G_S is $\mathbb{F}^m$ and $x, y \in G_S$ is connected iff $x - y \in S$ (see e.g., [JM21]). For our purposes, we quote the following results derived directly from the expanding property of biased sets.

Lemma A.3 ([BSSVW03, Lemma 4.3]). *Suppose $S \subseteq \mathbb{F}^m$ is λ-biased. Then for any $B \subseteq \mathbb{F}^m$ of density $\mu = \frac{|B|}{|\mathbb{F}|^m}$ and any $\varepsilon > 0$, we have*

$$\Pr_{x \sim \mathbb{F}^m, y \sim S} \left[\left| \frac{|\ell_{x,y} \cap B|}{|\ell_{x,y}|} - \mu \right| > \varepsilon \right] \leq \left(\frac{1}{|\mathbb{F}|} + \lambda \right) \cdot \frac{\mu}{\varepsilon^2}.$$

Finally we remark that λ-biased sets exist for $|S| = \Omega\left(\frac{m \log |\mathbb{F}|}{\lambda^2} \right)$ [AR94] and efficient explicit constructions of sizes $\mathrm{poly}(m, \log |\mathbb{F}|, \frac{1}{\lambda})$ are also obtained.

Fact A.4 (See e.g., [JM21]). *For any finite field $\mathbb{F}$, positive integer m, and parameter $\lambda \in (0, 1]$, a λ-biased set of size $\mathrm{poly}(m, \log |\mathbb{F}|, \frac{1}{\lambda})$ can be constructed efficiently in time $\mathrm{poly}(m, |\mathbb{F}|, \frac{1}{\lambda})$.*

Augmented Derandomized Parallel Low Degree Test. Unfortunately the derandomized parallel low degree test is not sufficient to guarantee the exact quantitative degree condition, as the directions S have fairly limited possibilities. More concretely, even if $\mathsf{LDTest}_S^{f,g}$ succeeds with probability 1, it is not guaranteed that f is parallel degree-d. To compensate the missing directions, we

---

[10]Technically, g only needs to be defined on lines whose directions are in S. We choose to assume g is defined over all lines $\mathbb{L}$ for simplicity.

[11]In some literature this symmetric assumption is not imposed. The parameter λ in that case is comparable with the one here by a multiplicative factor of 2.

[12]This homomorphism is usually referred as character. It is trivial if it maps everything to 1. An illustrative example is when $\mathbb{F} = \mathbb{F}_2$ and χ is a parity function.

need to augment Definition A.1 with an additional test that checks the consistency of f and g on a purely random direction from the origin [BSSVW03].

Definition A.5 (Augmented Derandomized Parallel Low Degree Test)**.** Let $S \subseteq \mathbb{F}^m$ and $f \colon \mathbb{F}^m \to \mathbb{F}^t$, $g \colon \mathbb{L} \to \mathbb{F}^{(d,t)}$. The augmented derandomized parallel low degree test $\mathsf{AugLDTest}_S^{f,g}$ is executed as follows: with equal probability we perform one of the following two tests:

- Independently select $x \sim \mathbb{F}^m$ and $y \sim S$, then accept iff $f(x) = g(\ell_{x,y})(x)$.
- Select $z \sim \mathbb{F}^m$, then accept iff $f(z) = g(\ell_{0^m,z})(z)$.

The first test in $\mathsf{AugLDTest}_S^{f,g}$ is simply $\mathsf{LDTest}_S^{f,g}$, for which we will later show that it guarantees that f is close to parallel degree-md. Then the second test allows us to further bring the degree down to d.

A.2 Codeword Testing

Theorem 3.6 and Theorem 3.7 follow directly from the following result, combined with the explicit constructions for biased sets.

Theorem A.6. *Assume $|\mathbb{F}| \geq 6md$, $|\mathbb{F}| \geq 2^{100} \cdot m \log |\mathbb{F}|$, and $\lambda \leq \frac{1}{2^{100} \cdot m \log |\mathbb{F}|}$. Let $S \subseteq \mathbb{F}^m$ be a λ-biased set. If*

$$\mathbf{Pr}\left[\mathsf{AugLDTest}_S^{f,g} \text{ accepts} \right] \geq 1 - \delta,$$

then f is $2^{40}\delta$-close to parallel degree-d.

Proof of Theorem 3.6. We first note that $|\mathbb{F}|^m > \binom{m+d}{d}$ and $|\mathbb{F}| > d$ are satisfied assuming the conditions on $|\mathbb{F}|$ in Theorem 3.6, thus $\mathsf{RM}^{\mathbb{F},m,d,t}$ is well defined.

Then we instantiate Theorem A.6 with $\lambda = \frac{1}{2^{100} \cdot m \log |\mathbb{F}|}$ and the λ-biased set S by Fact A.4. The construction of S is efficient in time $\mathsf{poly}(|\mathbb{F}|, m)$ and S has size $(m \log |\mathbb{F}|)^{O(1)}$. Then we define $\mathcal{P}_{\mathsf{ldt}}$ as $\mathsf{AugLDTest}_S^{f,g}$ where $T = f$ and π is defined to be the entries of g that can be possibly queried. Recall Definition A.5. Then we have

$$|\pi| \leq |\mathbb{F}|^m \cdot |S| + |\mathbb{F}|^m = |\mathbb{F}|^m \cdot (m \log |\mathbb{F}|)^{O(1)}.$$

In addition, by merging the randomness of the two tests in $\mathsf{AugLDTest}_S^{f,g}$, $\mathcal{P}_{\mathsf{ldt}}$ tosses

$$1 + \log\left(|\mathbb{F}|^m |S|\right) = m \log |\mathbb{F}| + O\left(\log \log |\mathbb{F}| + \log m\right)$$

total coins. The completeness is obvious and the soundness follows from Theorem A.6. $\qquad \square$

Proof of Theorem 3.7. Now we turn to Theorem 3.7. By Theorem 3.6, it suffices to implement $\mathcal{P}_{\mathsf{ldt}}$ purely on f. In addition, since $\mathcal{P}_{\mathsf{ldt}} = \mathsf{AugLDTest}_S^{f,g}$ and by Definition A.5, it performs the same check in parallel for each coordinate $i \in [t]$. This means that we only need to instantiate $\mathcal{P}_{\mathsf{ldt}}$ for a single coordinate (or equivalently, think of $t = 1$) to design the circuit C_{ldt}.

To get rid of the extra proof g, we simply set $g = f_{\mathbb{L}}$. Then, whenever we need information about entries in g (i.e., a line polynomial), we can probe the entries along the line in f to compute it. We remark that this is inefficient in terms of the query complexity, but it is still efficient in terms of the circuit complexity.

Now we describe the construction of C_{ldt} for a fixed coordinate $i \in [t]$. Based on the coin toss of $\mathcal{P}_{\text{ldt}}$, it checks the consistency of an $\mathbb{F}$-valued point (i.e., $f[i](x)$ or $f[i](z)$) with an evaluation point (i.e., x or z) of a degree-d line polynomial over $\mathbb{F}$ (i.e., $f_{\ell_{x,y}}[i]$ or $f_{\ell_{0^m,z}}[i]$). To implement it as a circuit, we take the conjunction of all the sub-circuit outcome from coin toss possibilities, where each sub-circuit performs the following computation.

- It first interpolates the line polynomial using entries of $f[i]$ along the line, and checks if this line polynomial is degree-d.

  This can be efficiently done with $\text{poly}|\mathbb{F}|$ gates.

- Then it evaluates the value of the desired point of the line polynomial, and checks if it is the same as the one directly obtained from $f[i]$.

  This also requires $\text{poly}|\mathbb{F}|$ gates only.

The correctness of C_{ldt} follows directly from Theorem 3.6 and our choice of $g = f_{\mathbb{L}}$. Since the number of coin toss possibilities is $|\mathbb{F}|^m \text{poly}(m, \log |\mathbb{F}|)$, by our assumption on $|\mathbb{F}|$, the size of C_{ldt} is $|\mathbb{F}|^m \text{poly}|\mathbb{F}|$ as claimed. $\qquad\square$

To prove the statement about AugLDTest, we need to analyze LDTest first, which guarantees a weaker degree bound.

Theorem A.7. *Assume $|\mathbb{F}| \geq 3d$, $|\mathbb{F}| \geq 2^{100} \cdot m \log |\mathbb{F}|$, and $\lambda \leq \frac{1}{2^{100} \cdot m \log |\mathbb{F}|}$. Let $S \subseteq \mathbb{F}^m$ be a λ-biased set. If*

$$\mathbf{Pr}\left[\text{LDTest}_S^{f,g} \text{ accepts}\right] \geq 1 - \delta,$$

then f is $2^{30}\delta$-close to parallel degree-md.

Assuming Theorem A.7, we conclude the proof of Theorem A.6.

Proof of Theorem A.6. First we assume $\delta \leq 2^{-40}$ since otherwise $2^{40}\delta \geq 1$ and the statement trivially holds. Recall Definition A.5 that $\text{AugLDTest}_S^{f,g}$ executes $\text{LDTest}_S^{f,g}$ with probability $1/2$. Hence $\text{LDTest}_S^{f,g}$ must accept with probability at least $1 - 2\delta$. By Theorem A.7, this means that f is $2^{31}\delta$-close to a parallel degree-md polynomial f'. It suffices to show that f' is acutally parallel degree-d.

Assume towards contradiction that f' is not parallel degree-d. Now we consider the second half of $\text{AugLDTest}_S^{f,g}$, which checks if $f(z) = g(\ell_{0^m,z})(z)$ for $z \sim \mathbb{F}^m$. Then we have

$$\mathbf{Pr}_{z \sim \mathbb{F}^m}\left[f(z) = g(\ell_{0^m,z})(z)\right] \leq \mathbf{Pr}_{z \sim \mathbb{F}^m}\left[f(z) \neq f'(z)\right] + \mathbf{Pr}_{z \sim \mathbb{F}^m}\left[f'(z) = g(\ell_{0^m,z})(z)\right]$$
$$\leq 2^{31}\delta + \mathbf{Pr}_{z \sim \mathbb{F}^m}\left[f'(z) = g(\ell_{0^m,z})(z)\right]. \tag{6}$$

To analyze (6), we consider the following quantity:

$$\mathbf{Pr}_{w \sim \mathbb{F}^m, i \sim \mathbb{F}}\left[f'(i \cdot w) = g(\ell_{0^m,w})(i \cdot w)\right]. \tag{7}$$

On the one hand, conditioned on $i \neq 0$, we have $\ell_{0^m,w} = \ell_{0^m,i\cdot w}$ and $i \cdot w$ being uniform in $\mathbb{F}^m$. Therefore we can relate (6) with (7):

$$(7) \geq \left(1 - \frac{1}{|\mathbb{F}|}\right) \cdot \mathbf{Pr}_{z \sim \mathbb{F}^m}\left[f'(z) = g(\ell_{0^m,z})(z)\right]. \tag{8}$$

On the other hand, conditioned on w, $f'(i \cdot w)$ is a univariate parallel degree-md polynomial in i. Let $d < d' \leq md$ be the parallel degree of f'. Then the coefficient of $i^{d'}$ in $f'(i \cdot w)$ is a non-zero parallel degree-d' polynomial. By Schwartz–Zippel lemma, it vanishes on at most $\frac{d'}{|\mathbb{F}|}$ fraction of choices of w. For each w that the top coefficient of $i^{d'}$ does not vanish, by Schwartz–Zippel lemma, $f'(i \cdot w)$ agrees with $g(\ell_{0^m,w})(i \cdot w)$ on at most d' choices of i since $g(\ell_{0^m,w})$ is parallel degree-d and $d < d'$. This means

$$(7) \leq \Pr_{w \sim \mathbb{F}^m} \left[\text{coeff of } i^{d'} \text{ in } f'(i \cdot w) \text{ vanishes} \right] + \Pr_{w \sim \mathbb{F}^m, i \sim \mathbb{F}} \left[f'(i \cdot w) = g(\ell_{0^m,w})(i \cdot w) \mid \text{not vanish} \right]$$

$$\leq \frac{d'}{|\mathbb{F}|} + \frac{d'}{|\mathbb{F}|} \leq \frac{2md}{|\mathbb{F}|}. \qquad \text{(since } d' \leq md\text{)}$$

Combining this with (8) and (6), we have

$$\Pr_{z \sim \mathbb{F}^m} [f(z) = g(\ell_{0^m,z})(z)] \leq 2^{31}\delta + \frac{2md}{|\mathbb{F}| - 1} \leq \frac{1}{2},$$

where we used the fact that $\delta \leq 2^{-40}$ and $|\mathbb{F}| \geq 6md$. Since this is half of the actual AugLDTest$_S^{f,g}$, it means

$$\Pr \left[\text{AugLDTest}_S^{f,g} \text{ accepts} \right] \leq \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} < 1 - \delta,$$

which is a contradiction.

In conclusion, f' must be parallel degree-d. $\qquad \square$

Our Theorem A.7 is the parallel version of [BSSVW03, Theorem 4.1]. Its proof follows from iteratively applying the following lemma, which is the parallel version of [BSSVW03, Lemma 4.4].

Lemma A.8. *Assume $|\mathbb{F}| \geq 3d$. Let $S \subseteq \mathbb{F}^m$ be a λ-biased set and $T \subseteq S$ of size $|T| \geq |S|/2$. Let $f \colon \mathbb{F}^m \to \mathbb{F}^t$. If*

$$\Pr \left[\text{LDTest}_T^{f,f_\mathbb{L}} \text{ accepts} \right] \geq 1 - \delta,$$

then for any $2\delta \leq \gamma \leq 2^{-20}$, there exists $f' \colon \mathbb{F}^m \to \mathbb{F}^t$ and $T' \subseteq T$ with the following properties:

1. $|T'| \geq \left(1 - \frac{\delta}{\gamma}\right) |T| \geq \frac{|T|}{2}$.
2. $\Delta(f', f) \leq 4\delta$.
3. $\Pr \left[\text{LDTest}_{T'}^{f',f_\mathbb{L}} \text{ accepts} \right] \geq 1 - 2^{40} \cdot \gamma \cdot \left(\frac{1}{|\mathbb{F}|} + \lambda \right)$.

Assuming Lemma A.8, we first conclude Theorem A.7.

Proof of Theorem A.7. First we assume $\delta \leq 2^{-30}$ since otherwise $2^{30}\delta \geq 1$ and the statement trivially holds. Next, we can also assume without loss of generality that $g = f_\mathbb{L}$. This is because, for each possible[13] line $\ell \in \mathbb{L}$, LDTest$_S^{f,g}$ conditioned on this line checks a uniformly random point $x \in \ell$ whether $f(x) = g(\ell)(x)$. Since $g(\ell)$ is parallel degree-d, the success probability maximized when $g(\ell) = f_\mathbb{L}(\ell)$.

We will repeatedly apply Lemma A.8 to bring the soundness gap down to $\ll |\mathbb{F}|^{-2m}$, at which point it is actually zero by granularity. Then we use the following characterization of parallel low degree polynomials similar to [RS96] to arrive at an actual parallel degree-md polynomial.

---

[13] A line is possible in LDTest$_S^{f,g}$ if its direction lies in S.

Theorem A.9. *Let $S \subseteq \mathbb{F}^m$ be a λ-biased set and $T \subseteq S$ of size $|T| > \frac{1+\lambda}{2} \cdot |S|$. Then $f \colon \mathbb{F}^m \to \mathbb{F}^t$ is parallel degree-md if $f|_{\ell_{x,y}}$ is parallel degree-d for every $x \in \mathbb{F}^m, y \in T$.*

Lemma A.8 will be proved in Appendix A.4. Now we focus on reducing the soundness gap.

The $\delta \geq \frac{1}{2^{60} \cdot m \log |\mathbb{F}|}$ Case. In this case, we first perform a pre-processing round to bring down the soundness gap. By Lemma A.8 with $T = S$ and $\gamma = 2^{-20}$, we have $S_1 \subseteq S$ and $f^{(1)} \colon \mathbb{F}^m \to \mathbb{F}^t$ with the following property:

1. $|S_1| \geq \left(1 - 2^{-10}\right) \cdot |S|$, since $\delta \leq 2^{-30}$.

2. $\Delta(f^{(1)}, f) \leq 4\delta$.

3. $\mathbf{Pr}\left[\mathsf{LDTest}_{S_1}^{f^{(1)}, f_{\mathbb{L}}} \text{ accepts}\right] \geq 1 - \frac{1}{2^{60} \cdot m \log |\mathbb{F}|}$, since $|\mathbb{F}| \geq 2^{100} \cdot m \log |\mathbb{F}|$ and $\lambda \leq \frac{1}{2^{100} \cdot m \log |\mathbb{F}|}$.

Now define

$$\delta_i = \frac{2^{-i}}{2^{50} \cdot m \log |\mathbb{F}|} \quad \text{and} \quad \gamma_i = \delta_i \cdot 2^{10} \cdot m \log |\mathbb{F}| = \frac{2^{-i}}{2^{40}}.$$

For each $i = 1, \dots, 2\lceil m \log |\mathbb{F}| \rceil$, we apply Lemma A.8 on $\delta_i, S_i, f^{(i)}$ and obtain $\delta_{i+1}, S_{i+1}, f^{(i+1)}$. To show the correctness of this process, we verify by induction on i that the conditions in Lemma A.8 are satisfied, i.e.,[14]

$$|S_i| \geq 0.9|S| \quad \text{and} \quad \mathbf{Pr}\left[\mathsf{LDTest}_{S_i}^{f^{(i)}, f_{\mathbb{L}}}\right] \geq 1 - \delta_i, \tag{9}$$

where we omit $2\delta_i \leq \gamma_i \leq 2^{-20}$ since it holds by the definition of δ_i, γ_i.

The base case $i = 1$ is valid by Item 1 and Item 3. For the inductive cases $i \geq 2$, we first observe that the first condition in (9) follows from the following calculation:

$$|S_i| \geq \left(1 - \frac{\delta_{i-1}}{\gamma_{i-1}}\right)|S_{i-1}| = \left(1 - \frac{1}{2^{10} \cdot m \log |\mathbb{F}|}\right)|S_{i-1}| \qquad \text{(by Lemma A.8)}$$

$$\geq \cdots \geq \left(1 - \frac{1}{2^{10} \cdot m \log |\mathbb{F}|}\right)^{i-1}|S_1| \qquad \text{(by Lemma A.8 iteratively)}$$

$$\geq \left(1 - \frac{1}{2^{10} \cdot m \log |\mathbb{F}|}\right)^{i-1} \cdot \left(1 - 2^{-10}\right)|S| \qquad \text{(by Item 1)}$$

$$\geq 0.9|S|. \qquad \text{(since } i \leq 2\lceil m \log |\mathbb{F}| \rceil\text{)}$$

The second condition in (9) follows from last round of Lemma A.8, which establishes that

$$\mathbf{Pr}\left[\mathsf{LDTest}_{S_i}^{f^{(i)}, f_{\mathbb{L}}} \text{ accepts}\right] \geq 1 - 2^{40} \cdot \gamma_{i-1} \cdot \left(\frac{1}{|\mathbb{F}|} + \lambda\right) = 1 - 2^{-i+1} \cdot \left(\frac{1}{|\mathbb{F}|} + \lambda\right)$$

$$\geq 1 - 2^{-i+1} \cdot 2 \cdot \frac{1}{2^{100} \cdot m \log |\mathbb{F}|} \geq 1 - \delta_i.$$

Let $k = 2\lceil m \log |\mathbb{F}| \rceil$. Then the above analysis shows

$$|S_k| \geq 0.9|S| > \frac{1+\lambda}{2} \cdot |S| \quad \text{and} \quad \mathbf{Pr}\left[\mathsf{LDTest}_{S_k}^{f^{(k)}, f_{\mathbb{L}}} \text{ accepts}\right] \geq 1 - \frac{2^{-k}}{2^{50} \cdot m \log |\mathbb{F}|} > 1 - |\mathbb{F}|^{-2m}.$$

---

[14]In Lemma A.8 we only require $|S_i| \geq |S|/2$. Here we strengthen it for convenience of Theorem A.9.

Since $\text{LDTest}_{S_k}^{f^{(k)}, f_{\mathbb{L}}}$ samples $x \sim \mathbb{F}^m$ and $y \sim S_k \subseteq \mathbb{F}^m$ then perform a deterministic check whether $f^{(k)}(x) = f_{\mathbb{L}}(\ell_{x,y})(x)$, its accepting probability is an integer multiple of $\frac{1}{|\mathbb{F}|^m \cdot |S_k|} \geq |\mathbb{F}|^{-2m}$. Therefore, we actually have $\mathbf{Pr}\left[\text{LDTest}_{S_k}^{f^{(k)}, f_{\mathbb{L}}} \text{ accepts}\right] = 1$, which means that $f^{(k)}$ is parallel degree-d on $\ell_{x,y}$ for all $x \in \mathbb{F}^m, y \in S_k$. By Theorem A.9, this means that $f^{(k)}$ is parallel degree-md. In addition, its distance from the original f is

$$
\begin{aligned}
\Delta(f^{(k)}, f) &\leq \Delta(f^{(1)}, f) + \sum_{i=1}^{k-1} \Delta(f^{(i+1)}, f^{(i)}) \leq 4\delta + \sum_{i=1}^{k-1} \Delta(f^{(i+1)}, f^{(i)}) && \text{(by Item 2)} \\
&\leq 4\delta + \sum_{i=1}^{k-1} 4 \cdot \delta_i = 4\delta + \sum_{i=1}^{k-1} \frac{4 \cdot 2^{-i}}{2^{50} \cdot m \log |\mathbb{F}|} && \text{(by Lemma A.8)} \\
&\leq 2^{30}\delta. && \text{(since } \delta \geq \frac{1}{2^{60} \cdot m \log |\mathbb{F}|}\text{)}
\end{aligned}
$$

The $\delta < \frac{1}{2^{60} \cdot m \log |\mathbb{F}|}$ Case. In this case, the analysis is even simpler as we do not need preprocessing. Let $S_1 = S$ and $f^{(1)} = f$. Define

$$
\delta_i = 2^{-i+1} \cdot \delta \quad \text{and} \quad \gamma_i = \delta_i \cdot 2^{10} \cdot m \log |\mathbb{F}| = 2^{-i+11} \cdot \delta \cdot m \log |\mathbb{F}|.
$$

We also apply Lemma A.8 for each $i = 1, \ldots, 2\lceil m \log |\mathbb{F}| \rceil$ on $\delta_i, S_i, f^{(i)}$ to obtain $\delta_{i+1}, S_{i+1}, f^{(i+1)}$. The conditions in Lemma A.8 along this process can be verified in the similar fashion. Here we only highlight the difference: for the second condition, we have

$$
\begin{aligned}
\mathbf{Pr}\left[\text{LDTest}_{S_i}^{f^{(i)}, f_{\mathbb{L}}} \text{ accepts}\right] &\geq 1 - 2^{40} \cdot \gamma_{i-1} \cdot \left(\frac{1}{|\mathbb{F}|} + \lambda\right) = 1 - 2^{-i+51} \cdot \delta \cdot m \log |\mathbb{F}| \cdot \left(\frac{1}{|\mathbb{F}|} + \lambda\right) \\
&\geq 1 - 2^{-i-48} \cdot \delta \geq 1 - \delta_i,
\end{aligned}
$$

and for the third condition, we have

$$
\gamma_i = 2^{-i+11} \cdot \delta \cdot m \log |\mathbb{F}| \leq \frac{2^{-i+11}}{2^{60}} \leq 2^{-20}.
$$

Then similarly, we set $k = 2\lceil m \log |\mathbb{F}| \rceil$ and obtain $f^{(k)}$ as a parallel degree-md polynomial. Moreover, we have

$$
\begin{aligned}
\Delta(f^{(k)}, f) &\leq \sum_{i=1}^{k-1} \Delta(f^{(i+1)}, f^{(i)}) \leq \sum_{i=1}^{k-1} 4 \cdot \delta_i && \text{(since } f^{(1)} = f \text{ and by Lemma A.8)} \\
&= \sum_{i=1}^{k-1} 4 \cdot 2^{-i+1} \cdot \delta \leq 2^{30}\delta,
\end{aligned}
$$

which completes the proof. $\qquad\square$

A.3 One Round of Correction

This section is devoted to proving Lemma A.8, which follows the sketch outlined in [BSSVW03].

Lemma ([Lemma A.8](#) Restated). *Assume $|\mathbb{F}| \geq 3d$. Let $S \subseteq \mathbb{F}^m$ be a λ-biased set and $T \subseteq S$ of size $|T| \geq |S|/2$. Let $f\colon \mathbb{F}^m \to \mathbb{F}^t$. If*

$$\mathbf{Pr}\left[\mathsf{LDTest}_T^{f,f_\mathbb{L}} \text{ accepts}\right] \geq 1 - \delta,$$

then for any $2\delta \leq \gamma \leq 2^{-20}$, there exists $f'\colon \mathbb{F}^m \to \mathbb{F}^t$ and $T' \subseteq T$ with the following properties:

1. *$|T'| \geq \left(1 - \frac{\delta}{\gamma}\right)|T| \geq \frac{|T|}{2}$.*
2. *$\Delta(f', f) \leq 4\delta$.*
3. *$\mathbf{Pr}\left[\mathsf{LDTest}_{T'}^{f',f_\mathbb{L}} \text{ accepts}\right] \geq 1 - 2^{40} \cdot \gamma \cdot \left(\frac{1}{|\mathbb{F}|} + \lambda\right)$.*

Proof. Let $T' \subseteq T$ be the set of directions $y \in T$ such that for at least $1 - \gamma$ fraction of $x \in \mathbb{F}^m$, f agrees with the parallel line polynomial $\ell_{x,y}$ on x. That is,

$$T' = \left\{y \in T\colon \left|\left\{x \in \mathbb{F}^m\colon f(x) = f_\mathbb{L}(\ell_{x,y})(x)\right\}\right| \geq (1 - \gamma) \cdot |\mathbb{F}|^m\right\}. \tag{10}$$

Then

$$
\begin{aligned}
1 - \delta &\leq \mathbf{Pr}\left[\mathsf{LDTest}_T^{f,f_\mathbb{L}} \text{ accepts}\right] && \text{(by assumption)} \\
&= \Pr_{x \sim \mathbb{F}^m, y \sim T}\left[f(x) = f_\mathbb{L}(\ell_{x,y})(x)\right] && \text{(by Definition A.1)} \\
&= \frac{|T'|}{|T|} \Pr_{x \sim \mathbb{F}^m, y \sim T}\left[f(x) = f_\mathbb{L}(\ell_{x,y})(x) \mid y \in T'\right] + \frac{|T \setminus T'|}{|T|} \Pr_{x \sim \mathbb{F}^m, y \sim T}\left[f(x) = f_\mathbb{L}(\ell_{x,y})(x) \mid y \notin T'\right] \\
&\leq \frac{|T'|}{|T|} + \left(1 - \frac{|T'|}{|T|}\right) \cdot (1 - \gamma), && \text{(by (10))}
\end{aligned}
$$

which implies [Item 1](#) by rearranging.

To construct $f'\colon \mathbb{F}^m \to \mathbb{F}^t$, for each $x \in \mathbb{F}^m$ we define $f'(x) \in \mathbb{F}^t$ to be the most common value of $f_\mathbb{L}(\ell_{x,y})(x)$ over $y \in T'$ where we break tie arbitrarily. Now we verify [Item 2](#). Let

$$B = \left\{x \in \mathbb{F}^m\colon \Pr_{y \sim T'}\left[f(x) \neq f_\mathbb{L}(\ell_{x,y})(x)\right] \geq \frac{1}{2}\right\}.$$

By the definition of f', we know that $f'(x) = f(x)$ holds for any $x \notin B$. Hence [Item 2](#) reduces to showing $\mathbf{Pr}_{x \sim \mathbb{F}^m}[x \in B] \leq 4\delta$ as follows:

$$
\begin{aligned}
\Pr_{x \sim \mathbb{F}^m}[x \in B] &= \Pr_{x \sim \mathbb{F}^m}\left[\Pr_{y \sim T'}\left[f(x) \neq f_\mathbb{L}(\ell_{x,y})(x)\right] \geq \frac{1}{2}\right] \\
&\leq 2 \cdot \Pr_{x \sim \mathbb{F}^m, y \sim T'}\left[f(x) \neq f_\mathbb{L}(\ell_{x,y})(x)\right] && \text{(by Markov's inequality)} \\
&\leq 2 \cdot \frac{|T|}{|T'|} \cdot \Pr_{x \sim \mathbb{F}^m, y \sim T}\left[f(x) \neq f_\mathbb{L}(\ell_{x,y})(x)\right] && \text{(since } T' \subseteq T\text{)} \\
&= 2 \cdot \frac{|T|}{|T'|} \cdot \mathbf{Pr}\left[\mathsf{LDTest}_T^{f,f_\mathbb{L}} \text{ rejects}\right] \leq 2 \cdot \frac{|T|}{|T'|} \cdot \delta && \text{(by assumption)} \\
&\leq 4\delta. && \text{(by Item 1)}
\end{aligned}
$$

To prove Item 3, we first get rid of f'. For a fixed $x \in \mathbb{F}^m$ and each $b \in \mathbb{F}^t$, let $U_b \subseteq T'$ be the set of directions y such that $f_{\mathbb{L}}(\ell_{x,y})(x) = b$. Denote $b^* = f'(x)$, which is defined to be the most common value of $f_{\mathbb{L}}(\ell_{x,y})(x)$ over $y \in T'$. Thus $|U_{b^*}| \geq |U_b|$ for any $b \in \mathbb{F}^t$. As a result, we have

$$\Pr_{y \sim T'} \left[f'(x) = f_{\mathbb{L}}(\ell_{x,y})(x) \right] = \frac{|U_{b^*}|}{|T'|} = \frac{|U_{b^*}|}{|T'|} \cdot \sum_b \frac{|U_b|}{|T'|} \qquad \text{(since } U_b\text{'s form a partition of } T')$$

$$\geq \sum_b \left(\frac{|U_b|}{|T'|} \right)^2 \qquad \text{(since } |U_{b^*}| \geq |U_b| \text{ for all } b)$$

$$= \Pr_{y_1, y_2 \sim T'} \left[f_{\mathbb{L}}(\ell_{x,y_1})(x) = f_{\mathbb{L}}(\ell_{x,y_2})(x) \right].$$

Now taking the negation and the expectation over random x, we have

$$\Pr \left[\mathsf{LDTest}_{T'}^{f', f_{\mathbb{L}}} \text{ rejects} \right] = \Pr_{x \sim \mathbb{F}^m, y \sim T'} \left[f'(x) \neq f_{\mathbb{L}}(\ell_{x,y})(x) \right]$$

$$\leq \Pr_{x \sim \mathbb{F}^m, y_1, y_2 \sim T'} \left[f_{\mathbb{L}}(\ell_{x,y_1})(x) \neq f_{\mathbb{L}}(\ell_{x,y_2})(x) \right]. \tag{11}$$

To upper bound (11), we will use bivariate testing theorems [PS94] similar to [LRSW23, FS95]. The idea is, as sketched in [BSSVW03], to use Lemma A.3 to show the following claim.

Claim A.10. With probability at least

$$1 - 2^{40} \cdot \gamma \cdot \left(\frac{1}{|\mathbb{F}|} + \lambda \right)$$

over x, y_1, y_2, we have

(a) $\Pr_{i \sim \mathbb{F}} \left[f_{\mathbb{L}}(\ell_{x+i \cdot y_1, y_2})(x + i \cdot y_1) \neq f(x + i \cdot y_1) \right] \leq \frac{1}{200}$.

(b) $\Pr_{j \sim \mathbb{F}} \left[f_{\mathbb{L}}(\ell_{x+j \cdot y_2, y_1})(x + j \cdot y_2) \neq f(x + j \cdot y_2) \right] \leq \frac{1}{200}$.

(c) $\Pr_{i,j \sim \mathbb{F}} \left[f_{\mathbb{L}}(\ell_{x+i \cdot y_1, y_2})(x + i \cdot y_1 + j \cdot y_2) \neq f(x + i \cdot y_1 + j \cdot y_2) \right] \leq \frac{1}{200}$.

(d) $\Pr_{i,j \sim \mathbb{F}} \left[f_{\mathbb{L}}(\ell_{x+j \cdot y_2, y_1})(x + i \cdot y_1 + j \cdot y_2) \neq f(x + i \cdot y_1 + j \cdot y_2) \right] \leq \frac{1}{200}$.

Claim A.10 intuitively says that f is almost parallel degree-d along the line ℓ_{x,y_1} (Item (a)), the line ℓ_{x,y_2} (Item (b)), and the plane spanned by $\ell_{x,y_1}, \ell_{x,y_2}$ (Items (c) and (d)). For simplicity, define

$$R(i,j) = f_{\mathbb{L}}(\ell_{x+i \cdot y_1, y_2})(x + i \cdot y_1 + j \cdot y_2) \quad \text{and} \quad C(i,j) = f_{\mathbb{L}}(\ell_{x+j \cdot y_2, y_1})(x + i \cdot y_1 + j \cdot y_2). \tag{12}$$

Given Claim A.10 and (11), it suffices to show

$$R(0,0) = C(0,0) \quad \text{assuming Items (a) to (d).} \tag{13}$$

Observe that $R \colon \mathbb{F} \times \mathbb{F} \to \mathbb{F}^t$ is a parallel bivariate polynomial. Recall that $f_{\mathbb{L}}$ is the restriction of f on each line of its closest parallel degree-d polynomial. Thus for each fixed $i \in \mathbb{F}$, $R(i,j)$ is parallel degree-d in the variable j. Let $R_1, \ldots, R_t \colon \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ be the single-output components of R. Then for each $r \in [t]$ and in $R_r(i,j)$, the variable i has degree at most $|\mathbb{F}| - 1$ and the variable j has degree at most d. We say R is of parallel degree $(|\mathbb{F}| - 1, d)$ for shorthand. Similarly, $C(i,j)$ is of parallel degree $(d, |\mathbb{F}| - 1)$.

Combining Items (c) and (d), we have

$$\Pr_{i,j \sim \mathbb{F}} [R(i,j) \neq C(i,j)] \leq \frac{1}{100}. \tag{14}$$

We will use the following lemma to zero out the inconsistent entries of $R(i,j)$ and $C(i,j)$.

45

Lemma A.11 ([PS94, Lemma 3]). *Let $Z \subseteq \mathbb{F} \times \mathbb{F}$ be arbitrary. There exists a non-zero bivariate polynomial $E \colon \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ of degree $\left(\left\lfloor \sqrt{|Z|} \right\rfloor, \left\lfloor \sqrt{|Z|} \right\rfloor \right)$ such that $E(i,j) = 0$ for all $(i,j) \in Z$.*

By setting Z to be the set of (i,j) with $R(i,j) \neq C(i,j)$, we have $|Z| \leq |\mathbb{F}|^2/100$ by (14). Thus by Lemma A.11, there is a non-zero bivariate polynomial E of degree at most $\left(\frac{|\mathbb{F}|}{10}, \frac{|\mathbb{F}|}{10} \right)$ such that $E(i,j)R(i,j) = E(i,j)C(i,j)$ holds for all $i,j \in \mathbb{F}$. Note that R and C are polynomials with t outputs and E has single output. The product $E(i,j)R(i,j)$ produces a vector of length t with entries of $R(i,j)$ scaled by $E(i,j)$; same for $E(i,j)C(i,j)$.

Then we use the following lemma to show that R and C are close to a parallel bivariate polynomial of parallel degree (d,d).

Lemma A.12 ([PS94, Lemma 8]). *Let $E, P \colon \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ be bivariate polynomials of degree (b,a) and $(b+d, a+d)$ respectively. Assume $n > \min\{2b+2d, 2a+2d\}$. Assume further there exist distinct $i_1, \ldots, i_n$ such that $E(i_k, \cdot)$ divides $P(i_k, \cdot)$ for all $k \in [n]$, and distinct $j_1, \ldots, j_n$ such that $E(\cdot, j_k)$ divides $P(\cdot, j_k)$ for all $k \in [n]$. Then $E(\cdot, \cdot)$ divides $P(\cdot, \cdot)$, i.e., there exists a bivariate polynomial $Q \colon \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ of degree (d,d) such that $E(i,j)Q(i,j) = P(i,j)$ holds for all $i,j \in \mathbb{F}$.*

Fix an arbitrary $r \in [t]$. For each $i, j \in \mathbb{F}$, define $P_r(i,j) = E(i,j)R_r(i,j)$, which also equals $E(i,j)C_r(i,j)$ since $E(i,j)R(i,j) = E(i,j)C(i,j)$. Since E is non-zero and has degree $\left(\frac{|\mathbb{F}|}{10}, \frac{|\mathbb{F}|}{10} \right)$, there are at least $\frac{9|\mathbb{F}|}{10}$ many distinct $i \in \mathbb{F}$ such that $E(i, \cdot)$ is an univariate polynomial not identically zero, for which $E(i, \cdot)$ divides $P_r(i, \cdot)$; same for $E(\cdot, j)$ and $P_r(\cdot, j)$. By Lemma A.12 with $a = b = \frac{|\mathbb{F}|}{10}$ and $n = \frac{9|\mathbb{F}|}{10}$, there exists a bivariate polynomial $Q_r \colon \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ with parallel degree (d,d) such that $E(i,j)Q_r(i,j) = P_r(i,j) = E(i,j)R_r(i,j) = E(i,j)C_r(i,j)$ holds for all $i, j \in \mathbb{F}$, where we used the fact that $|\mathbb{F}| \geq 3d$ implies $|\mathbb{F}| > \frac{10d}{7}$.[15]

Let $Q \colon \mathbb{F} \times \mathbb{F} \to \mathbb{F}^t$ be the parallel-output bivariate polynomial with single-output components $Q_1, \ldots, Q_r$ obtained above. Then $Q(i, \cdot) = R(i, \cdot)$ holds as long as $E(i, \cdot)$ is not identically zero, which has at least $\frac{9|\mathbb{F}|}{10}$ possibilities out of $|\mathbb{F}|$ total choices off i. Recall Item (a). For at least $\frac{9}{10} - \frac{1}{200} > \frac{4}{5}$ fraction of i's, we have $f(x + i \cdot y_1) = R(i,0) = Q(i,0)$. Note that the distance between any two distinct (parallel) degree-d polynomial is at least $1 - \frac{d}{|\mathbb{F}|}$, which is at least $\frac{2}{3} < 2 \cdot \frac{1}{5}$ since $|\mathbb{F}| \geq 3d$. In addition, $Q(\cdot, 0)$ is parallel degree-d. Thus $Q(\cdot, 0)$ is the closest parallel degree-d polynomial of $f|_{\ell_{x,y_1}}$, i.e., $Q(i,0) = f_{\mathbb{L}}(\ell_{x,y_1})(x + i \cdot y_1) = C(i,0)$ holds for all $i \in \mathbb{F}$ where we recall (12). In particular, this means $Q(0,0) = C(0,0)$.

Similarly, using Item (b), we can show $Q(0,0) = R(0,0)$, which verifies (13) and thus Item 3, and completes the proof of Lemma A.8. $\qquad\square$

Finally it remains to prove Claim A.10.

Proof of Claim A.10. We show that each item holds with probability at least $1 - 2^{30} \cdot \gamma \cdot \left(\frac{1}{|\mathbb{F}|} + \lambda \right)$, and then Claim A.10 follows from a union bound.

We first consider Item (a) and the analysis for Item (b) is almost identical. For each $y \in S$, define
$$B_y = \left\{ x \in \mathbb{F}^m \colon f_{\mathbb{L}}(\ell_{x,y})(x) \neq f(x) \right\}.$$

---

[15] If $d \geq 1$, then $3d > \frac{10d}{7}$; otherwise $\frac{10d}{7} = 0 < 1 \leq |\mathbb{F}|$.

Then for any fixed $y_2 \in T'$, we have $|B_{y_2}| \leq \gamma \cdot |\mathbb{F}|^m$ by (10), and, moreover,

$$\Pr_{x \sim \mathbb{F}^m, y_1 \sim T'} [\text{Item (a) does not hold}]$$

$$= \Pr_{x \sim \mathbb{F}^m, y_1 \sim T'} \left[\Pr_{i \sim \mathbb{F}} \left[f_{\mathbb{L}}(\ell_{x+i \cdot y_1, y_2})(x + i \cdot y_1) \neq f(x + i \cdot y_1) \right] > \frac{1}{200} \right]$$

$$= \Pr_{x \sim \mathbb{F}^m, y_1 \sim T'} \left[\Pr_{i \sim \mathbb{F}} \left[x + i \cdot y_1 \in B_{y_2} \right] > \frac{1}{200} \right] = \Pr_{x \sim \mathbb{F}^m, y_1 \sim T'} \left[\frac{|\ell_{x,y_1} \cap B_{y_2}|}{|\ell_{x,y_1}|} > \frac{1}{200} \right]$$

$$\leq \Pr_{x \sim \mathbb{F}^m, y_1 \sim T'} \left[\frac{|\ell_{x,y_1} \cap B_{y_2}|}{|\ell_{x,y_1}|} > \frac{|B_{y_2}|}{|\mathbb{F}|^m} + \frac{1}{400} \right] \qquad \left(\text{since } \frac{|B_{y_2}|}{|\mathbb{F}|^m} \leq \gamma \text{ and } \gamma \leq 2^{-20} \text{ by assumption}\right)$$

$$\leq \frac{|S|}{|T'|} \cdot \Pr_{x \sim \mathbb{F}^m, y_1 \sim S} \left[\frac{|\ell_{x,y_1} \cap B_{y_2}|}{|\ell_{x,y_1}|} > \frac{|B_{y_2}|}{|\mathbb{F}|^m} + \frac{1}{400} \right] \qquad (\text{since } T' \subseteq T \subseteq S)$$

$$\leq 4 \Pr_{x \sim \mathbb{F}^m, y_1 \sim S} \left[\frac{|\ell_{x,y_1} \cap B_{y_2}|}{|\ell_{x,y_1}|} > \frac{|B_{y_2}|}{|\mathbb{F}|^m} + \frac{1}{400} \right] \qquad (\text{by assumption and Item 1})$$

$$\leq 2^{20} \cdot \left(\frac{1}{|\mathbb{F}|} + \lambda \right) \cdot \frac{|B_{y_2}|}{|\mathbb{F}|^m} \qquad (\text{by Lemma A.3})$$

$$\leq 2^{20} \cdot \gamma \cdot \left(\frac{1}{|\mathbb{F}|} + \lambda \right). \qquad \left(\text{since } \frac{|B_{y_2}|}{|\mathbb{F}|^m} \leq \gamma\right)$$

By taking the expectation over y_2, Item (a) holds with the desired probability.

Now we consider Item (c) and the analysis for Item (d) is almost identical. For each $y \in S$, define

$$\overline{B}_y = \left\{ x \in \mathbb{F}^m : \Pr_{j \sim \mathbb{F}} \left[f_{\mathbb{L}}(\ell_{x,y})(x + j \cdot y) \neq f(x + j \cdot y) \right] > 2^{-10} \right\}. \tag{15}$$

For each $y_2 \in S$, we have

$$\frac{|\overline{B}_{y_2}|}{|\mathbb{F}|^m} = \Pr_{x \sim \mathbb{F}^m} \left[\Pr_{j \sim \mathbb{F}} \left[f_{\mathbb{L}}(\ell_{x,y_2})(x + j \cdot y_2) \neq f(x + j \cdot y_2) \right] > 2^{-10} \right]$$

$$\leq 2^{10} \cdot \Pr_{x \sim \mathbb{F}^m, j \sim \mathbb{F}} \left[f_{\mathbb{L}}(\ell_{x,y_2})(x + j \cdot y_2) \neq f(x + j \cdot y_2) \right] \qquad (\text{by Markov's inequality})$$

$$= 2^{10} \cdot \Pr_{x \sim \mathbb{F}^m, j \sim \mathbb{F}} \left[f_{\mathbb{L}}(\ell_{x+j \cdot y_2, y_2})(x + j \cdot y_2) \neq f(x + j \cdot y_2) \right] \qquad (\text{since } \ell_{x,y_2} = \ell_{x+j \cdot y_2, y_2})$$

$$= 2^{10} \cdot \Pr_{z \sim \mathbb{F}^m} \left[f_{\mathbb{L}}(\ell_{z,y_2})(z) \neq f(z) \right] = 2^{10} \cdot \frac{|B_{y_2}|}{|\mathbb{F}|^m}$$

$$\leq 2^{10} \cdot \gamma, \tag{16}$$

and

$$\Pr_{x \sim \mathbb{F}^m, y_1 \sim T'} [\text{Item (c) does not hold}]$$

$$= \Pr_{x \sim \mathbb{F}^m, y_1 \sim T'} \left[\Pr_{i,j \sim \mathbb{F}} \left[f_{\mathbb{L}}(\ell_{x+i \cdot y_1, y_2})(x + i \cdot y_1 + j \cdot y_2) \neq f(x + i \cdot y_1 + j \cdot y_2) \right] > \frac{1}{200} \right]$$

$$\leq \Pr_{x \sim \mathbb{F}^m, y_1 \sim T'} \left[\Pr_{i \sim \mathbb{F}} \left[x + i \cdot y_1 \in \overline{B}_{y_2} \right] > \frac{1}{300} \right], \tag{17}$$

where we use the following reasoning for the last inequality: if the event inside bracket does not happen, then

$$\Pr_{i,j \sim \mathbb{F}} \left[f_{\mathbb{L}}(\ell_{x+i \cdot y_1, y_2})(x + i \cdot y_1 + j \cdot y_2) \neq f(x + i \cdot y_1 + j \cdot y_2) \right]$$

47

$$\leq \frac{1}{300} + \Pr_{i,j\sim\mathbb{F}}\left[f_{\mathbb{L}}(\ell_{x+i\cdot y_1,y_2})(x+i\cdot y_1+j\cdot y_2) \neq f(x+i\cdot y_1+j\cdot y_2) \mid x+i\cdot y_1 \notin \overline{B}_{y_2}\right]$$

$$\leq \frac{1}{300} + 2^{-10} < \frac{1}{200}. \tag{by (15)}$$

Then similar to the analysis for Item (a), we continue upper bounding (17) as follows:

$$\text{RHS of (17)} = \Pr_{x\sim\mathbb{F}^m, y_1\sim T'}\left[\frac{|\ell_{x,y_1} \cap \overline{B}_{y_2}|}{|\ell_{x,y_1}|} > \frac{1}{300}\right]$$

$$\leq \Pr_{x\sim\mathbb{F}^m, y_1\sim T'}\left[\frac{|\ell_{x,y_1} \cap \overline{B}_{y_2}|}{|\ell_{x,y_1}|} > \frac{|\overline{B}_{y_2}|}{|\mathbb{F}|^m} + \frac{1}{400}\right] \qquad \text{(by (16) and } \gamma \leq 2^{-20}\text{)}$$

$$\leq 4 \Pr_{x\sim\mathbb{F}^m, y_1\sim S}\left[\frac{|\ell_{x,y_1} \cap \overline{B}_{y_2}|}{|\ell_{x,y_1}|} > \frac{|\overline{B}_{y_2}|}{|\mathbb{F}|^m} + \frac{1}{400}\right] \qquad \text{(since } T' \subseteq S \text{ and } |T'| \geq |S|/4\text{)}$$

$$\leq 2^{20} \cdot \left(\frac{1}{|\mathbb{F}|} + \lambda\right) \cdot \frac{|\overline{B}_{y_2}|}{|\mathbb{F}|^m} \leq 2^{30} \cdot \gamma \cdot \left(\frac{1}{|\mathbb{F}|} + \lambda\right). \qquad \text{(by Lemma A.3 and (16))}$$

Therefore Item (c) holds with the desired probabilit by taking the expectation over y_1. $\qquad\square$

A.4 Derandomized Characterizations of Parallel Low Degree Polynomials

In this part, we extend the characterization of low degree polynomials in [RS96] to the derandomized setting, where we only consider lines with directions generated by a (large subset of) small-biased set.

We use superscript $\top$ to denote vector and matrix transpose. For two vectors $u, v \in \mathbb{F}^d$, we use $\langle u, v\rangle$ to denote their inner product which equals $u^\top v$ (or $v^\top u$).

Theorem (Theorem A.9 Restated). *Let $S \subseteq \mathbb{F}^m$ be a λ-biased set and $T \subseteq S$ of size $|T| > \frac{1+\lambda}{2}\cdot |S|$. Then $f\colon \mathbb{F}^m \to \mathbb{F}^t$ is parallel degree-md if $f|_{\ell_{x,y}}$ is parallel degree-d for every $x \in \mathbb{F}^m, y \in T$.*

Proof. Assume without loss of generality that $t = 1$, since we can apply the analysis individually for each single-output component $f[1], \ldots, f[t]$.

We first prove that T has the full rank m. Assume towards contradiction that T has rank at most $m-1$. Then there exists a non-zero vector $z \in \mathbb{F}^m$ such that $z^\top y = 0$ holds for all $y \in T$. In light of Definition A.2, we construct a non-trivial homomorphism $\chi\colon \mathbb{F}^m \to \mu_p$ to derive a contradiction, where p is the characteristic of $\mathbb{F}$. Let $\xi\colon \mathbb{F} \to \mu_p$ be a non-trivial (group) homomorphism, where we view $\mathbb{F}$ as an additive group. We define for each $x \in \mathbb{F}^m$ that

$$\chi(x) = \xi(x^\top z),$$

which is a non-trivial homomorphism since ξ is a non-trivial homomorphism and $z \neq 0^m$. Note that for all $y \in T$, we have

$$\chi(y) = \xi(y^\top z) = \xi(0) = 1.$$

Thus

$$\left| \mathbb{E}_{y\sim S}[\chi(y)]\right| = \left| \mathbb{E}_{y\sim S}\left[\chi(y) \cdot \left(\mathbb{1}_{y\in T} + \mathbb{1}_{y\notin T}\right)\right]\right| = \left| \mathbb{E}_{y\sim S}\left[\mathbb{1}_{y\in T} + \chi(y) \cdot \mathbb{1}_{y\notin T}\right]\right|$$

48

$$\geq \mathop{\mathbb{E}}_{y \sim S} \left[\mathbb{1}_{y \in T} \right] - \mathop{\mathbb{E}}_{y \sim S} \left[\mathbb{1}_{y \notin T} \right] = \frac{|T|}{|S|} - \frac{|S| - |T|}{|S|}$$

$$> \lambda, \qquad\qquad\qquad\qquad \text{(since } |T| > \tfrac{1+\lambda}{2} \cdot |S|)$$

which contradicts Definition A.2.

Now we fix a set of m linearly independent directions $y_1, \ldots, y_m \in T$. Then we can interpolate f using degree-d polynomials $f|_{\ell_{x,y_1}}, \ldots, f|_{\ell_{x,y_m}}$ for $x \in \mathbb{F}^m$. For concreteness, we apply the invertible linear transform on $\mathbb{F}^m$ such that $y_1, \ldots, y_m$ map to axis parallel directions $e_1, \ldots, e_m$, where $e_i = (\underbrace{0, \ldots, 0}_{i-1}, 1, \underbrace{0, \ldots, 0}_{m-i})$. Let f' be the polynomial after this transform, which shares the same degree with f since the transform is invertible and linear. Since $f|_{\ell_{x,y_1}}, \ldots, f|_{\ell_{x,y_m}}$ are all degree-d, we know that f' is degree-d along the axis parallel lines. Then by polynomial interpolation (see e.g., [RS96, Lemma 28]), f' has degree at most md, which in turn means that f has degree at most md as claimed. $\qquad\square$