Electronic Colloquium on Computational Complexity, Report No. 353 (2026)

ECCC

On The Proof of The Unique Games Conjecture

Dana Moshkovitz*

November 28, 2026

Abstract

The Unique Games Conjecture of Khot postulates that for any $\delta, \varepsilon > 0$ there is a sufficiently large finite field $\mathbb{F}$, such that the following problem is NP-hard: on input a system of linear equations over variables $x_1, \ldots, x_n$ ranging over $\mathbb{F}$, where each equation is of the form $x_i - x_j = c_{ij}$ for some $c_{ij} \in \mathbb{F}$, the goal is to distinguish between the case where there is an assignment to the variables that satisfies at least $1 - \delta$ fraction of the equations, and the case where every assignment satisfies at most ε fraction of the equations.

The conjecture was proved by OpenAI. The proof follows the existing "Outer PCP – Inner code" framework in hardness of approximation. Its innovation is in the code, which can be thought of as *rounding* of the Hadamard code. The rounding gives rise to a unique *noise test* for the code. In this note we give an exposition of this construction.

Acknowledgements: This writeup is based on talks I gave at the Simons Institute in Berkeley on October 28-29, 2026, and at the University of Texas at Austin on November 4, 2026. I am especially grateful to Subhash Khot for discussions that led to the talks, and to Johan Håstad who shared with me his own writeup. I am also thankful to Scott Aaronson, Dor Minzer, Salil Vadhan and Will Sawin for their comments.

AI Use: I repeatedly questioned Astra about the OpenAI proof and my understanding of it. Astra also generated a figure for this writeup that describes the construction.

*Department of Computer Science, University of Texas at Austin. danama@cs.utexas.edu. Supported by NSF Grant CCF-1705028.

ISSN 1433-8092

1 Introduction

The Unique Games Conjecture of Khot [16] was a central open problem in hardness of approximation. One of its formulations, due to [22], is in terms of approximately solving linear equations:

Definition 1 (Max-2Lin)**.** *In the* Max-2Lin *problem over a finite field* $\mathbb{F}$ *the input is a system of linear equations over variables* $x_1, \ldots, x_n$*. Each equation is of the form* $x_i - x_j = c_{ij}$ *for some* $c_{ij} \in \mathbb{F}$*. The goal is to find an assignment to the variables over* $\mathbb{F}$ *that satisfies as many equations as possible.*

The *unique* in the name is because when $x_i - x_j = c_{ij}$, any value for x_i uniquely determines a value for x_j, and vice versa. The term *games* refers to *two prover games*. Here one imagines a verifier that picks a random equation $x_i - x_j = c_{ij}$, asks one prover what is the value of x_i, asks another prover what is the value of x_j, then checks if $x_i - x_j = c_{ij}$.

For $\delta, \varepsilon > 0$, the $1-\delta$ vs. ε Max-2Lin problem is to distinguish the case where at least $1-\delta$ fraction of the equations can be satisfied, and the case where at most ε fraction of the equations can be satisfied. The Unique Games Conjecture postulates the NP-hardness of this problem:

Theorem 2 (Unique Games Theorem [31])**.** *For any* $\delta, \varepsilon > 0$*, for sufficiently large* $\mathbb{F}$*, the* $1-\delta$ *vs.* ε Max-2Lin *problem over* $\mathbb{F}$ *is NP-hard.*

To prove Theorem 2 it suffices to prove it for any $\delta > 0$ and $\varepsilon = 0.99$ [34].

The Unique Games Theorem is a convenient starting point for hardness of approximation reductions, and Theorem 2 implies *optimal* hardness of approximation for problems like Vertex-Cover [26], Max-Cut [22] and Max-2Sat [4]. The theorem implies that a simple semidefinite programming algorithm is optimal for general constraint satisfaction problems [33]. The reader is referred to surveys by Khot for further information about the conjecture and its importance [19, 18, 20, 21].

In terms of approximation algorithms, there is an efficient $1 - \Theta(\sqrt{\delta \log |\mathbb{F}|})$ approximation algorithm for unique games (good for $|\mathbb{F}| \ll 2^{1/\delta}$), as well as an efficient $\tilde{\Omega}(n^{-\delta/(2-\delta)})$ approximation algorithm (good for $\delta \approx 1/\log n$) [10]. There is also an algorithm in time $2^{n^{\text{poly}\,\delta}}$ [1]. The 2-to-2 Theorem [23, 12, 11, 7, 24] implies that $\frac{1}{2}$ vs. ε unique games are NP-hard.

On October 6, 2026, OpenAI released a proof of the Unique Games Conjecture [31]. This writeup is an exposition of the proof. Along with the proof, OpenAI released proofs of optimal inapproximability for Max-Cut [29] and Vertex-Cover [30] that do not rely on the Unique Games Theorem.

2 The Reduction, Background on Hardness of Approximation

The reduction to unique games follows the *Outer PCP–Inner Code* paradigm in hardness of approximation initiated in [8, 15]. We start from the NP-hard 3Sat, and show a sequence of reductions that eventually reduce to Max-2Lin.

$$\text{3SAT} \xrightarrow{\text{PCP Theorem}} \text{Max-3LIN} \xrightarrow{\text{parallel repetition}} \text{Label-Cover} \xrightarrow{\text{code}} \text{Max-2LIN}$$

The first reduction is from 3Sat to approximating Max-3Lin:

Definition 3 (Max-3Lin)**.** *In the* Max-3Lin *problem the input consists of n binary variables $x_1, \ldots, x_n$ and equations of the form $x_i + x_j + x_k = c_{ijk}$ (mod 2) over them. The goal is to satisfy as many equations as possible.*

Håstad's work [15] based on the PCP Theorem [6, 5, 3, 2] shows that the $1-\delta$ vs. $\frac{1}{2}+\varepsilon$ Max-3Lin problem is NP-hard for any $\varepsilon, \delta > 0$ (a weaker result on NP-hardness of $1-\delta$ vs. 0.9 would suffice for us). The next reduction uses parallel repetition to reduce Max-3Lin to a structured instance of problem called Label-Cover; this is the "Outer PCP". The final reduction, which is the innovation of the proof, reduces to Max-2Lin. The reduction is based on a code. Traditionally, either the *long code* or the *Hadamard code* are used. The proof introduces a new twist on Hadamard code, a *Hadamard with a noise test*, which gets the best of both codes. In the remainder of this section we first discuss parallel repetition and the outer PCP. Then we describe how the outer PCP is composed with a code. We specify the long code and the Hadamard code, and discuss their properties that are useful for understanding the current construction. We highlight why neither code was sufficient for a reduction to Max-2Lin.

2.1 The Outer PCP

In Label-Cover the goal is to label the vertices of the graph as to satisfy as many edge constraints as possible:

Definition 4 (Label-Cover)**.** *The input is a bipartite graph $G = (U \cup V, E)$ together with a distribution over E; alphabets Σ_U, Σ_V; and constraints $\Sigma_u \subseteq \Sigma_U$ for $u \in U$ and $\pi_e : \Sigma_U \to \Sigma_V$ on $e \in E$. The goal is to label each vertex $u \in U$ with a label $\sigma_u \in \Sigma_u$ and every vertex $v \in V$ with a label $\sigma_v \in \Sigma_V$ as to satisfy as many edge constraints $\pi_e(\sigma_u) = \sigma_v$ as possible.*

Assume a Max-3Lin instance, and let us construct from it a Label-Cover instance. In the bipartite graph $G = (U \cup V, E)$ there is a U vertex for every tuple of k equations. There is a V vertex for every tuple that consists of k' equations and $k - k'$ variables for $k' \leq k$. A label of a vertex consists of binary values for each of the variables associated with the vertex. This includes variables in equations and other variables. We therefore set $\Sigma_U, \Sigma_V = \{0,1\}^{3k}$ with the understanding that labels to vertices in V that correspond to less than $3k$ variables would have 0's for the non-variables. Σ_u includes only the assignments to u's variables that satisfy all the Max-3Lin equations associated with u. We define the edges in E along with a probability distribution over them: Let $\eta > 0$ be a parameter. Pick a vertex $u \in U$ uniformly at random. Pick its neighboring vertex $v \in V$ by picking independently for each equation in u: with probability $1-\eta$, the equation, and with probability η, one of the three variables in the equation, uniformly at random. The constraint π_e on an edge $e = (u, v)$ maps a label to u, which consists of binary values to all of u's variables, to a label to the subset of u's variables associated with v.

Importantly, for every edge $(u, v) \in E$ the labels to u and to v are supposed to be the same on about $1-\eta$ fraction of their bits.

We have the following completeness and soundness properties:

- *Completeness:* If there is an assignment to Max-3Lin that satisfies $1-\delta$ fraction of the equations, then there is a labeling to the vertices of G that satisfies $1-\delta k$ fraction of the edges. We will pick $\delta \ll 1/k$.
- *Soundness:* If every assignment to Max-3Lin satisfies $\frac{1}{2}+\varepsilon$ fraction of the equations, then every labeling to the vertices of G satisfies $2^{-\Omega(\eta k)}$ fraction of the edges.

The soundness follows from the Parallel Repetition Theorem [35, 34]. It holds not just for G, but also for certain sub-graphs of G, e.g., the subgraph obtained by only considering vertices that contain a small fixed set of Max-3Lin variables.

2.2 Outer PCP – Inner Code Composition

To reduce from Label-Cover to Max-2Lin, we compose the Label-Cover instance with a suitable code. We replace each vertex in G with Max-2Lin variables that correspond to an encoding of the label to the vertex, introducing a variable per symbol in the encoding. We then introduce Max-2Lin equations that make three tests:

1. *Constraint test:* For a vertex $w \in U \cup V$, check that the Max-3Lin equations contained in w are satisfied.
2. *Consistency test:* For an edge $(u, v) \in E$, check that the labels to u and v are consistent on the joint variables.
3. *Codeword test:* Locally test the encoding of the label to a vertex $w \in U \cup V$.

Next we will define the long code and the Hadamard code, the two codes that are usually used as inner codes. We will see that for each of them, Max-2Lin equations could achieve only two out of the three tests.

2.3 The Long Code

The *long code* was introduced in [8] and is widely used in hardness of approximation [15, 17]. It encodes bits $\sigma \in \{0,1\}^k$ by the evaluations $\{f(\sigma)\}_f$ of all 2^{2^k} Boolean functions $f : \{0,1\}^k \to \{0,1\}$ on σ. If we were to use the long code in a hypothetical reduction from Label-Cover to Max-2Lin we would have had a variable for every vertex w and $f : \{0,1\}^{3k} \to \{0,1\}$. The variable is supposed to be assigned $f(\sigma_w)$ for σ_w the values of the (up to $3k$) Max-3Lin variables that correspond to w. This introduces $2^{2^{3k}}$ variables per vertex. While this is a large number, when k is a constant, $2^{2^{3k}}$ is a constant too.

Towards a constraint test: Denote by F_u the function that assigns each f the value of the variable associated with u and f. Consider a Max-3Lin equation on some of the bits in σ_u. Write the equation as $l(\sigma_u) = b$ for a linear $l : \{0,1\}^{3k} \to \{0,1\}$ and $b \in \{0,1\}$. Then, $(f + l)(\sigma_u) = f(\sigma_u) + l(\sigma_u) = f(\sigma_u) + b$ for any $f : \{0,1\}^{3k} \to \{0,1\}$. Therefore, for a long code codeword F_u we have $F_u(f + l) = F_u(f) + b$ for all f. We can impose this equality on any F_u by having just one variable for $F_u(f)$ and $F_u(f + l)$ for each f and determining the other value accordingly. This technique is called *folding* [15], and it is crucial for the construction we describe in this writeup. Another folding that is important for the analysis of the long code is that $F_u(f + 1) = F_u(f) + 1$ for all f. This property ensures that F_u is balanced.

Towards a consistency test: If σ_v is part of σ_u, then the long code of σ_v is part of the long code of σ_u. This is because every Boolean function of σ_v is also a Boolean function of σ_u. This yields a natural consistency test by comparing the encoding of σ_v and the corresponding sub-code of σ_u's encoding on a uniformly random f in σ_v's encoding. Unfortunately, this does not give a sound Max-2Lin consistency test, since the encoding of σ_v is a tiny portion of σ_u's encoding. We can change this portion of σ_u's encoding so it will look like it is consistent with

any σ_v of our choice! There is a sound MAX-3LIN consistency test that is used instead [15]. It follows from a technique called local decoding. However, a MAX-3LIN test is not good enough for a reduction to MAX-2LIN.

Towards a codeword test: There is a natural MAX-2LIN test for the long code. Let F be a purported long code codeword.

Noise Test:

1. Pick uniformly at random $f : \{0,1\}^{3k} \to \{0,1\}$.
2. Pick a noisy version $\tilde{f}$ of f by setting for every $x \in \{0,1\}^{3k}$ independently: $\tilde{f}(x) = f(x)$ with probability $1-\delta$ and $\tilde{f}(x) = r$ for a random $r \in \{0,1\}$ with probability δ.
3. Test if $F(f) = F(\tilde{f})$.

If F is the long code of σ, then the test passes with probability at least $1-\delta$, since $\tilde{f}(\sigma) = f(\sigma)$ with probability at least $1-\delta$.

Following [15], the noise test on general functions F is analyzed via Fourier analysis. We switch F to a $\{+1,-1\}$-valued function by considering $(-1)^F$ when we write F. We then decompose F in the orthonormal basis of *characters* $\chi_\alpha(f) = (-1)^{\langle \alpha, f\rangle}$, where the exponent is a linear function $\langle \alpha, f\rangle = \sum \alpha(x)f(x)$ for $\alpha : \{0,1\}^{3k} \to \{0,1\}$. The decomposition means that we write $F = \sum_\alpha \hat{F}(\alpha)\chi_\alpha$ for coefficients $\hat{F}(\alpha)$ with $\sum_\alpha \hat{F}(\alpha)^2 = 1$. We think of α as specifying a set of assignments in $\{0,1\}^{3k}$. We think of $\hat{F}(\alpha)^2$ as the weight on χ_α. A standard calculation shows that any F that passes the noise test with probability at least $1-\varepsilon$, has most of its weight on small sets: there exists $R = R(\delta)$ for which $\sum_{|\alpha| \le R} \hat{F}(\alpha)^2 \ge 1 - O(\varepsilon)$.

2.4 The Hadamard Code

The *Hadamard code* is sometimes used instead of the long code. It encodes bits $\sigma \in \{0,1\}^k$ by the evaluations $\{\langle a, \sigma\rangle\}_a$ of all 2^k Boolean $\mathbb{F}_2$-*linear* on z, where $a \in \{0,1\}^k$.

This means that in a supposed MAX-2LIN instance we will have a variable for every vertex w and $a \in \{0,1\}^{3k}$. The variable is supposed to be assigned $\langle a, \sigma_w\rangle$ for σ_w the values of the MAX-3LIN variables that correspond to w.

Towards a constraint test: Let F be a purported Hadamard codeword. Note that we can extend F to include all affine functions $\langle a, \cdot\rangle + b$ for $a \in \{0,1\}^k$ and $b \in \{0,1\}$, by defining $F(a,b) = F(a) + b$. Since the MAX-3LIN constraints are affine, folding works for Hadamard just as it works for the long code, and this allows a constraint test.

Towards a consistency test: If σ_v is part of σ_u, then the Hadamard encoding of σ_v is part of the Hadamard encoding of σ_u, since every linear function of σ_v is also a linear function of σ_u. However, the sub-code of σ_v's encoding is a much larger portion of the encoding of σ_u, namely roughly a $2^{-\eta k}$ fraction. While this is still a small fraction, it makes the natural MAX-2LIN consistency test work! This is thanks to an observation about sub-code covering [27]: for every vertex u, the *union* of the Hadamard encodings of all its neighboring v covers approximately uniformly the Hadamard encoding of σ_u. The consistency test now compares two copies of the Hadamard encoding of σ_u on a random location f.

Towards a codeword test: There is no MAX-2LIN test for the Hadamard code. There is only a MAX-3LIN linearity test [9].

2.5 Generalized Hadamard Code

While the Hadamard code does not have a MAX-2LIN codeword test, a generalization of it with larger alphabet does have a 2-to-2 codeword test with low error. We say that a test is 2-to-2, if it makes two queries, and given the value of one query there are exactly *two* acceptable values to the other query. The *generalized Hadamard code* encodes a message $\sigma \in \{0,1\}^k$ by the evaluations of all linear transformations on σ. Let $\mathcal{M}$ consist of all $l \times k$ binary matrices. Then the generalized Hadamard code maps σ to $\{M\sigma\}_{M\in\mathcal{M}}$.

As before, use $F : \mathcal{M} \to \{0,1\}^l$ to denote a purported codeword. The 2-to-2 generalized Hadamard tester is:

1. Pick uniformly at random $M \in \mathcal{M}$.
2. Pick uniformly at random $h \in \{0,1\}^l$ and $w \in \{0,1\}^k$.
3. Test whether $F(M + hw^T)$ is either $F(M)$ or $F(M) + h$.

Here hw^T is a rank-1 matrix whose every column i is either h or $\vec{0}$, depending on the random w_i. If F is the generalized Hadamard encoding of some σ, then $F(M + hw^T) = F(M) + F(hw^T)$, where $F(hw^T) = hw^T\sigma$, so the difference $F(M) - F(M + hw^T)$ is either h or $\vec{0}$, depending on $w^T\sigma$.

If F passes this test even with a small probability, then F must agree with a generalized Hadamard codeword on a *slice* of $\mathcal{M}$:

Theorem 5 (Generalized Hadamard 2-to-2 tester [7, 24])**.** *For every $\alpha > 0$, there are $\gamma > 0$ and $r \geq 1$, such that if F passes the 2-to-2 test with probability at least α, then there are "zoom in" and "zoom out" binary matrices I_0 ($r_1 \times l$), I_1 ($r_1 \times k$), O_0 ($k \times r_2$), O_1 ($l \times r_2$), $r_1, r_2 \leq r$, as well as $\sigma \in \{0,1\}^k$, $b \in \{0,1\}^l$, such that for at least γ fraction of the matrices $M \in \mathcal{M}$ satisfying $I_0M = I_1$ and $MO_0 = O_1$,*

$$F_u(M) = M\sigma + b.$$

The proof of the 2-to-2 Theorem, and specifically the paper [12], show that the structure guaranteed by Theorem 5 suffices for the reduction from MAX-3SAT we described.

3 Hadamard with a Noise Test

The innovation in the proof of the Unique Games Conjecture is a rounding of the generalized Hadamard code that enables a noise test. Generalized Hadamard is as before, except that we work with a large finite field $\mathbb{F} = \mathbb{F}_{2^d}$ of characteristic 2 instead of $\mathbb{F}_2$.

For any $\delta > 0$, we will define a non-linear rounding function[1] $\mathsf{Round} : \mathbb{F}^l \to \mathbb{F}$ that maps a long vector to a short output. Round satisfies a folding condition: There exist $H \subseteq \mathbb{F}^l$ and linear and onto $\lambda : H \to \mathbb{F}$, such that

$$\mathsf{Round}(x + h) = \mathsf{Round}(x) + \lambda(h),$$

[1] In the actual construction we have $\mathsf{Round} : (\mathbb{F}^3)^l \to \mathbb{F}^3$. We clarify why when we discuss the construction in Section 5.

for every $x \in \mathbb{F}^l$ and $h \in H$. The rounded generalized Hadamard encodes[2] $\sigma \in \{0,1\}^k$ by mapping every $M \in \mathcal{M}$ to $\mathsf{Round}(M\sigma)$. Because of the folding of Round, the encoding satisfies the following folding condition:

Folding: Let $K \subseteq H$ contain those $h \in H$ for which $\lambda(h) = 0$. We say that $F : \mathcal{M} \to \mathbb{F}$ is folded if for every $M \in \mathcal{M}$ and $h \in K$ and every $i \in [k]$ it holds $F(M + he_i^T) = F(M)$. For every $h \in H$ it holds $F(M + he_0^T) = F(M) + \lambda(h)$.

For a purported codeword $F : \mathcal{M} \to \mathbb{F}$, there is a Max-2Lin test:

Noise test:

1. Pick a uniformly random $M \in \mathcal{M}$.
2. Pick uniformly random $w \in \mathbb{F}^k$ and $i \in [l]$. Let $N \in \mathcal{M}$ be a matrix that is 0 everywhere except for the i'th row where it is w.
3. Test whether $F(M) = F(M + N)$.

We associate with F the collection of binary functions $F_z(M) = (-1)^{\phi_z(F(M))}$ for linear $\phi_z : \mathbb{F} \to \{0,1\}$. We can then decompose each F_z in the Fourier basis (see the preliminaries for information about the Fourier transform over $\mathcal{M}$). There is a character for every S corresponding to k linear functions $\psi_i : \mathbb{F}^l \to \{0,1\}$. We define the folded rank $\mathrm{rank}^{\mathrm{fold}}(S)$ as the dimension of the subspace spanned by the linear functions when they are evaluated on $H \subseteq \mathbb{F}^l$. The folded rank of a generalized Hadamard codeword is 1.

The main theorem concerning generalized Hadamard and the noise test argues that the noise test accepts rounded generalized Hadamard codewords with high probability, while rejecting folded F's of high folded rank:

Theorem 6 (Hadamard with a Noise Test)**.** *For any $\delta > 0$, for sufficiently large $\mathbb{F}$, we have:*

- *Completeness: Any rounded generalized Hadamard encoding passes the noise test with probability at least $1 - \delta$.*
- *Soundness: For any $\varepsilon > 0$, for any folded $F : \mathcal{M} \to \mathbb{F}$, if the noise test passes with probability at least $1-\varepsilon$, then there exists $R = R(\delta)$ such that $\mathbf{E}_z \left[\sum_{S:\mathrm{rank}^{\mathrm{fold}}(S)>R} \hat{F}_z(S)^2\right] = O(\varepsilon)$.*

The key observation is that low folded rank implies that a modified 2-to-2 test passes with significant probability. The modified test shifts the matrix by a matrix whose columns are randomly either $\vec{0}$ or h where $h \in H$:

2-to-2 on H Linearity Test:

1. Pick uniformly at random $M \in \mathcal{M}$.
2. Pick uniformly at random $h \in H$ and $w \in \{0,1\}^k$.
3. Test if $F(M + hw^T)$ is either $F(M)$ or $F(M) + \lambda(h)$

The acceptance probability of the 2-to-2 on H test we will argue is roughly 2^{-R} for R the bound on the folded rank. This acceptance probability is significant since R is a constant depending only on δ:

[2]In fact, since we want to include all affine transformations $\mathbb{F}^k \to \mathbb{F}^l$, and not just all linear transformations, and the non-linearity of Round would not let us get those from the linear transformations "for free", in the actual construction we append 1 as the 0'th coordinate of σ, and consider $l \times (k+1)$ matrices M. For notational convenience, we will continue to refer to $l \times k$ matrices.

Lemma 3.1. *For F with $\mathbf{E}_z\left[\sum_{S:\mathrm{rank}^{\mathrm{fold}}(S)>R}\hat{F}_z(S)^2\right]=O(\varepsilon)$, the 2-to-2 on H test passes with probability at least $(1-O(\varepsilon))\cdot 2^{-R}$.*

Proof. Consider a character χ_S where S corresponds to linear functions $\psi_i:\mathbb{F}^l\to\{0,1\}$.

$$\chi_S(hw^T)=(-1)^{\sum_i w_i\psi_i(h)}$$

Averaging over uniform $w\in\{0,1\}^k$, the character is 0 unless $\psi_i(h)=0$ for all i, i.e.,

$$\mathop{\mathbf{E}}_{h,w}\left[\chi_S(hw^T)\right]=\Pr_h\left[\psi_i(h)=0\forall i\right].$$

If $\mathrm{rank}^{\mathrm{fold}}(S)\le R$, then for a uniform $h\in H$, we have $\Pr_h\left[\psi_i(h)=0\forall i\right]\ge 2^{-R}$.

We write $F_z=\sum\hat{F}_z(S)\chi_S$. We lower bound the probability that the 2-to-2 test passes by the probability a unique test $F(M+hw^T)=F(M)$ passes. We can write this probability as

$$\mathop{\mathbf{E}}_{z,M,h,w}\left[(-1)^{\phi_z(F(M)+F(M+hw^T))}\right],$$

which equals:

$$\begin{aligned}
\mathop{\mathbf{E}}_{z,M,h,w}\left[F_z(M)F_z(M+hw^T)\right] &= \mathop{\mathbf{E}}_{z,M,h,w}\left[\sum_{S,S'}\hat{F}_z(S)\chi_S(M)\hat{F}_z(S')\chi_{S'}(M+hw^T)\right]\\
&= \mathop{\mathbf{E}}_{z,M,h,w}\left[\sum_{S,S'}\hat{F}_z(S)\hat{F}_z(S')\chi_S(M)\chi_{S'}(M)\chi_{S'}(hw^T)\right]\\
&= \mathop{\mathbf{E}}_{z}\left[\sum_{S}\hat{F}_z(S)^2\mathop{\mathbf{E}}_{h,w}\left[\chi_S(hw^T)\right]\right]\\
&\ge (1-O(\varepsilon))\cdot 2^{-R}
\end{aligned}$$

■

By the soundness of the 2-to-2 test, we can deduce a linear structure on parts of the table F: We can view each M as decomposed to a part that maps to H and to a linear complement part, and fix the linear complement part uniformly at random. The proof proceeds by showing that structure on such M's suffices for the reduction.

We remark that human efforts towards a proof of the Unique Games Conjecture [25, 13] also attempted to "round" a version of the Hadamard code to introduce a noise test. The concrete suggestion was to use *half-spaces* in Gaussian space, namely, $sign(\langle a,x\rangle)$, so the linear $\langle a,x\rangle$ is analogous to the Hadamard code, and the rounding is given by the sign function. There is a natural noise test in Gaussian space, and it guarantees that functions that pass with probability $1-\varepsilon$ have $1-O(\varepsilon)$ of their Hermite mass on degree $O(1/\delta)$ (The Hermite transform is analogous to the Fourier transform). The OpenAI proof achieves a similar property in discrete space, which implies a codeword test as we saw.

4 Preliminaries

4.1 Fourier Transform

Define inner product between functions $f,g:\mathbb{F}_2^n\to\mathbb{R}$ as $\langle f,g\rangle=\frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n}f(x)g(x)$. A character $\chi:\mathbb{F}_2^n\to\mathbb{R}$ is of the form $\chi_\alpha(x)=(-1)^{\sum_i\alpha_i x_i}$ for $\alpha\in\mathbb{F}_2^n$. The characters form an

orthonormal basis for the space of all functions $f : \mathbb{F}_2^n \to \mathbb{R}$. The function can be decomposed as $f(x) = \sum_\alpha \hat{f}(\alpha)\chi_\alpha(x)$.

In this writeup we consider functions with several different domains, each can be identified with $\mathbb{F}_2^n$ for some n. Depending on the domain, we use different conventions to describe the linear functions associated with the character. When the domain is $\mathcal{M}$, which corresponds to $l \times k$ matrices over $\mathbb{F}_{2^d}$, the character is indexed by $S = (\psi_1, \ldots, \psi_k)$ for linear functions $\psi_i : \mathbb{F}^l \to \{0,1\}$, $i = 1, \ldots, k$. The character is $\chi_S(M) = (-1)^{\sum \psi_i(M_i)}$ where the M_i are the columns of M. When the domain is $\mathbb{F}^3$ for $\mathbb{F} = \mathbb{F}_2^d$ and we wish to use finite field arithmetics, we associate characters with $z \in \mathbb{F}^3$ and think of the characters as $\chi_z(x) = (-1)^{\mathrm{Tr}(\langle z,x \rangle)}$, where Tr is the binary trace function $\mathrm{Tr}(t) = t + t^2 + \cdots + t^{2^{d-1}}$.

5 The Rounding

5.1 The Near-Linear Quadratic

The rounding is based on the quadratic $Q : \mathbb{F}^3 \to \mathbb{F}^3$,

$$Q(x, y, z) := (yz, xz, xy).$$

For a binary linear function $L : \mathbb{F}^3 \to \mathbb{F}^3$ it holds that $L(a) + L(b) = L(a + b)$ for every $a, b \in \mathbb{F}^3$. The quadratic Q is near-linear in the following sense:

Lemma 5.1. *For every $a \neq b \in \mathbb{F}^3$, it holds that $Q(a) + Q(b) = Q(a + b) + w$, where $w \in \mathbb{F}^3$ is orthogonal to a and b. If $a = t \cdot b$ for some $t \in \mathbb{F}$ then $w = \vec{0}$.*

Proof. Recall that $\mathbb{F}$ is of characteristic 2. Denote $a = (a_1, a_2, a_3)$ and $b = (b_1, b_2, b_3)$. Then

$$Q(a + b)_1 + Q(a)_1 + Q(b)_1 = (a_2 + b_2)(a_3 + b_3) + a_2a_3 + b_2b_3 = a_2b_3 + b_2a_3.$$

Similarly, $Q(a+b)_2 + Q(a)_2 + Q(b)_2 = a_1b_3 + b_1a_3$ and $Q(a+b)_3 + Q(a)_3 + Q(b)_3 = a_1b_2 + b_1a_2$.

If $a = t \cdot b$ for $t \in \mathbb{F}$, then each of the three coordinates is 0. For instance, the first coordinate becomes $t \cdot b_2b_3 + t \cdot b_2b_3 = 0$.

For general a, the inner product $\langle Q(a + b) + Q(a) + Q(b), a \rangle$ equals

$$a_1(a_2b_3 + b_2a_3) + a_2(a_1b_3 + b_1a_3) + a_3(a_1b_2 + b_1a_2).$$

Every term appears twice in the sum, and therefore gets cancelled. ∎

Note that the second part of the lemma says that the restriction of Q to lines through the origin $\{t \cdot b \mid t \in \mathbb{F}\}$ is binary linear.

An observation due to Scott Aaronson is that Lemma 5.1 relies on $Q(a+b)+Q(a)+Q(b)$ being the *cross product* of a and b in $\mathbb{F}^3$. Indeed, the lemma proves that, as well as known properties of the cross product (like orthogonality). Cross product only exists in 3 and 7 dimensions, so this was the reason to switch to $\mathbb{F}^3$. The mapping Q is apparently classical in algebraic geometry, and specifically in the theory of cubic Jordan algebras. Astra gives the references [32, 28, 14].

5.2 The Tree

In this section we define Round. In the actual construction, Round is a function $(\mathbb{F}^3)^l \to \mathbb{F}^3$ rather than a function $\mathbb{F}^l \to \mathbb{F}$. The function is defined using a tree. There are l leaves, and we think of the input to Round as assigning labels from $\mathbb{F}^3$ to the leaves. We will describe how to

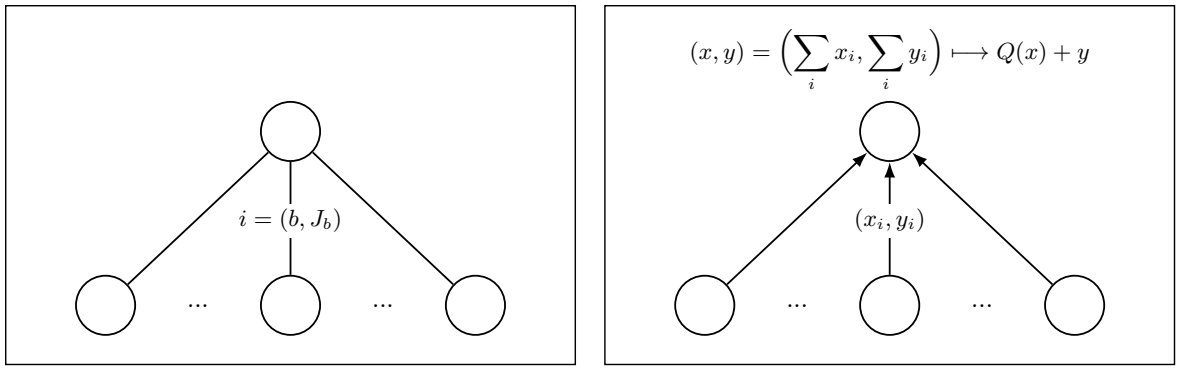

Figure 1: Each child is associated with $i = (b, J_b)$. The mapping J_b maps the child's label in $\mathbb{F}^3$ to $(x_i, y_i) \in \mathbb{F}^3 \times \mathbb{F}^3$. At the node we compute $(x, y) = (\sum x_i, \sum y_i)$. The node's label is $Q(x) + y$.

assign a label to a node based on the labels of its many children. Round outputs the label of the root.

The noise changes the label of one random leaf in the tree. We will arrange that at every node, independently, random noise in one random child gets canceled at the node with probability approximately $1/|\mathbb{F}|^2$. If the noise is canceled, then the labels of all the nodes up to the root are unchanged by the noise as well. The tree will be of depth $\Theta(|\mathbb{F}|^2 \log(1/\delta))$, so at the root there is probability at least $1 - \delta$ that the label is unchanged despite the leaf change.

For every $b \neq \vec{0} \in \mathbb{F}^3$ which represents a random line through the origin, define

$$U_b = \{(a, Q(a) + w) \mid a = t \cdot b,\ t \in \mathbb{F}, w \perp b\}.$$

Note that U_b is a binary linear subspace of $\mathbb{F}^3 \times \mathbb{F}^3$ because of the linear property of the quadratic (Lemma 5.1). Moreover, U_b is of the same size as $\mathbb{F}^3$. We will consider linear bijections $\mathbb{F}^3 \to U_b$.

Every non-leaf node v has a child for every pair: (i) line through the origin in $\mathbb{F}^3$, given by a direction $b \neq \vec{0} \in \mathbb{F}^3$; (ii) linear bijection $J_b : \mathbb{F}^3 \to U_b$. We denote the i'th child of v for $i = (b, J_b)$ by v_i. We interpret the label $\sigma_i \in \mathbb{F}^3$ to v_i as a pair $(x_i, y_i) \in U_b$ where $(x_i, y_i) = J_b(\sigma_i)$. At the node v, we compute $(x, y) = (\sum_i x_i, \sum_i y_i)$. The label to the node v is $Q(x) + y$. We use the notation $label_v(leaves)$ to denote the label of v given leaf labels $leaves$.

Folding. When $x = 0$ we have that $Q(x) = 0$ and the label to the node v is y. Similarly, a translation of the labels of the children for which $x = 0$ leads to a translation of the label to the node by y. We will use this to define folding recursively. For every node v we maintain a linear subspace H_v of the labels to the leaves of v's sub-tree, and a linear translation function $\lambda_v : H_v \to \mathbb{F}^3$, so $label_v(leaves + h) = label_v(leaves) + \lambda_v(h)$ for all $h \in H_v$ and $leaves$ a label to the leaves of v's sub-tree. In particular, $label_v(h) = \lambda_v(h)$ for all $h \in H_v$. Moreover, for a uniform $h \in H_v$ we have that $\lambda_v(h)$ is uniform over $\mathbb{F}^3$:

For the leaves $H_v = \mathbb{F}^3$ and λ_v is the identity function. For a non-leaf node we have H_v correspond to the elements $(h_i)_i \in \prod_i H_{v_i}$ for which $x = 0$ where x is computed based on children labels $(\lambda_{v_i}(h_i))_i$. The linear translation function λ_v maps $(h_i)_i$ to the corresponding y based on $(\lambda_{v_i}(h_i))_i$. Note that indeed for a uniform $h \in H_v$ we have that $\lambda_v(h)$ is uniform over $\mathbb{F}^3$. For the root v we set $H = H_v$ and $\lambda = \lambda_v$.

6 Completeness

The noise test corresponds to a linear change at one random leaf of the rounding tree. We first show that thanks to the near-linearity (Lemma 5.1), noise at one random child of a node is somewhat likely to cancel at the node:

Lemma 6.1. *For any node v, and any labeling to its children, picking a uniformly random child i and adding to its label a uniformly random element in $\mathbb{F}^3$ leads to the same label to v as the original labeling of the children with probability $1/|\mathbb{F}|^2$.*

Proof. Let $i = (b, J_b)$ be the child we pick. Let $(x, y) \in \mathbb{F}^3 \times \mathbb{F}^3$ denote the aggregate at v computed based on the original labeling of the children. The original value at v is $Q(x) + y$. The random addition to the i'th child adds a uniform $(a, Q(a)+w)$ to (x, y), where a is a random element on the line in direction b, and w is a uniform vector that is orthogonal to the line. The new label becomes $Q(x + a) + (y + Q(a) + w)$. By Lemma 5.1, $Q(x + a) + Q(a) = Q(x) + w'$ for w' orthogonal to a. The label at the node does not change iff $w = w'$, in which case both labels are $Q(x) + y$. This happens with probability $1/|\mathbb{F}|^2$. ∎

Pick the noisy leaf by picking uniformly and independently at random $i_1, \ldots, i_t$, where i_1 denotes which of its father's children the noisy leaf is, i_2 denotes the same for the leaf's father, etc. The next lemma gives the probability of the noise canceling at the root:

Lemma 6.2. *For every $t \geq 0$, for any labeling of the leaves in the depth-t sub-tree, when adding a random element from $\mathbb{F}^3$ to the noisy leaf, the label at the depth-t root doesn't change with probability at most*

$$\left(1 - \frac{1}{|\mathbb{F}|^3}\right)\left(1 - \frac{1}{|\mathbb{F}|^2} + \frac{1}{|\mathbb{F}|^3}\right)^t.$$

Proof. At depth 0, only zero noise, which happens with probability $\frac{1}{|\mathbb{F}|^3}$ does not change the label. At depth t, denote $i_t = (b, J_b)$. Let $\Delta \in \mathbb{F}^3$ denote the change in the label of the i_t'th child. Note that i_t is independent of Δ. If $\Delta \neq 0$, then $J_b(\Delta)$ is uniformly random in $U_b - \{0\}$. We can then invoke Lemma 6.1 and deduce that the noise gets canceled with probability at least $\frac{1}{|\mathbb{F}|^2} - \frac{1}{|\mathbb{F}|^3}$. ∎

For appropriate $t = O(|\mathbb{F}|^2 \log(1/\delta))$ the probability that the noise changes the root label is at most δ, and the completeness of the noise test follows.

7 Soundness

We wish to prove the soundness of the noise test for folded functions $F : \mathcal{M} \to \mathbb{F}^3$ using Fourier analysis. To do that, we first consider one folded Fourier character χ_S. Recall that χ_S is associated with linear functions $\psi_i : (\mathbb{F}^3)^l \to \{0, 1\}$ for $i = 1, \ldots, k$. In this section we upper bound the probability that all the ψ_i vanish when picking uniformly at random a coordinate in $[l]$ and substituting 0's elsewhere. The upper bound will be an arbitrarily small p_0, and we use it to deduce an upper bound on the noisy character:

Lemma 7.1. *Let χ_S be a folded character of folded rank at least R. Then, for N the noise matrix in the noise test, $\mathbf{E}[\chi_S(N)] \leq p_0$.*

Proof. The noise matrix N is 0 everywhere, except for a single row j corresponding to the noisy leaf, where it is uniformly random $w_1, \ldots, w_k \in \mathbb{F}^3$. The character S corresponds to linear functions $\psi_i : (\mathbb{F}^3)^l \to \{0,1\}$ for $i = 1, \ldots, k$.

$$\chi_S(N) = (-1)^{\sum_i \psi_i(0 \cdots 0 w_i 0 \cdots 0)}.$$

Thus,

$$\mathop{\mathbf{E}}_{j,w_1,\ldots,w_k}[\chi_S(N)] = \mathop{\mathbf{E}}_{j}\left[\prod_i \mathop{\mathbf{E}}_{w_i}\left[(-1)^{\psi_i(0\cdots 0 w_i 0\cdots 0)}\right]\right].$$

This expectation is non-zero precisely when all the ψ_i vanish after the substitution of zeros outside the j'th coordinate. ∎

To obtain the upper bound p_0, we imagine sampling the noise by going down the tree until we pick the noisy leaf: Pick independently uniform $i_1, i_2, \ldots$, where i_1 identifies a child v_1 of the root v_0, i_2 identifies its child v_2, etc. When we pick i_j we set the labels of all the leaves outside its sub-tree to 0. The substitution leads to a function on the leaves of the sub-tree. We denote the number of leaves in v_j's tree by l_j.

Since the folded rank of S is at least $R \geq 1$, there must initially be R linear functions $\psi_i : (\mathbb{F}^3)^l \to \{0,1\}$ that are linearly independent when evaluated on H. We will denote by $\mathcal{S}_0$ the binary linear space spanned by those ψ_i. At the j'th step we can think of the ψ_i as $\psi_{j,i} : (\mathbb{F}^3)^{l_j} \to \{0,1\}$.

By folding, for all $h \in H$ with $\lambda(h) = 0$ we have $\psi_i(h) = 0$. Moreover, folding at the root implies folding at the node v_j for all $j \geq 0$: For every $j \geq 0$ let K_{v_j} consist of all the elements in H_{v_j} that λ_{v_j} maps to 0. We argue that for all $j \geq 0$: $\psi_{j,i}(h) = 0$ for all $h \in K_{v_j}$. Indeed, this is true at the root. Moreover, if it is true at v_{j-1}, then it is true at v_j by construction. Now consider two elements $h, h' \in H_{v_j}$ that λ_{v_j} maps to the same element in $\mathbb{F}^3$. Then, $\lambda_{v_j}(h - h') = \lambda_{v_j}(h) - \lambda_{v_j}(h') = 0$, and hence $\psi_{j,i}(h - h') = 0$. Thus, $\psi_{j,i}(h) = \psi_{j,i}(h')$.

We will show that with high probability, for all j, some $\psi_{j,i}$'s still remain linearly independent when evaluated on H_{v_j}. We will denote by $\mathcal{S}_j$ the binary linear subspace spanned by the $\psi_{j,i}$ when evaluated on H_{v_j}. We define the folded rank $\mathrm{rank}^{\mathrm{fold}}(v_j)$ as the dimension of $\mathcal{S}_j$. We keep only $\mathrm{rank}^{\mathrm{fold}}(v_j)$ of the $\psi_{j,i}$ whose restrictions to H_{v_j} span $\mathcal{S}_j$, and discard the other $\psi_{j,i}$. Recall that at a leaf, i.e., when $l_j = 1$, we have $H_{v_j} = \mathbb{F}^3$.

The main lemma (proved in Section 8) is that as we go down the tree: (1) folded rank decreases are rare. (2) if $\mathcal{S}_j$ is randomized, then the folded rank decreases by at most 1. (3) $\mathcal{S}_{j+1}$ is randomized. In Section 8 we define a specific property of a randomized $\mathcal{S}_j$ that is relevant for folded rank; we call $\mathcal{S}_j$ that satisfy it *generic*. The lemma is as follows:

Lemma 7.2 (Rank Decrease)**.** *For every j, if $\mathrm{rank}^{\mathrm{fold}}(v_j) = r \geq 1$, then*

1. *The probability over i_{j+1} that $\mathrm{rank}^{\mathrm{fold}}(v_{j+1}) < r$ is at most $2^{O(r)}/|\mathbb{F}|^2$.*
2. *If $\mathcal{S}_j$ is generic then $\mathrm{rank}^{\mathrm{fold}}(v_{j+1}) \geq r-1$, and the probability over i_{j+1} that $\mathrm{rank}^{\mathrm{fold}}(v_{j+1}) = r - 1$ is $O(r/|\mathbb{F}|^2)$.*
3. *$\mathcal{S}_{j+1}$ is generic except with probability $2^{O(r^2)}/|\mathbb{F}|$ over i_{j+1}.*

If the folded rank at the root is R, what is the probability that the folded rank at the leaf is 0? The probability that there is a folded rank decrease at the root is $2^{O(r)}/|\mathbb{F}|^2$. For any internal node, the folded rank can drop by more than 1 only when we have both a non-generic $\mathcal{S}_j$ (because i_j was unlucky) and a folded rank drop (because i_{j+1} is unlucky). This happens

with probability $2^{O(r^2)}/|\mathbb{F}|^3$. The height of the tree is $\Theta(\log(1/\delta)\,|\mathbb{F}|^2)$. Therefore, We can upper bound the error term $2^{O(r^2)}/|\mathbb{F}|^3$ for $r \leq R$ by a union bound. This means that except with probability $O(2^{O(R^2)}\log(1/\delta)/|\mathbb{F}|)$, the rank never decrease by more than 1 each time we go down the tree.

However, upper bounding the rest of the rank decrease by summing $O(R/\,|\mathbb{F}|^2)$ along the path does not give a meaningful upper bound. Luckily, Lemma 7.2 implies that the error probability gets lower as the rank gets lower, and we'll be able to employ this fact to bound the rank decrease as follows.

Let $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$ denote the n'th harmonic number. We define $H_0 = 0$. For $n \geq 1$ it holds that $H_n = \ln n + O(1)$.

Lemma 7.3 (The Harmonic trick)**.** *If $R_0, R_1, R_2, \ldots, R_s$ are natural random variables that are bounded by $R_0 = R$, and $\mathbf{E}\left[H_{R_i} - H_{R_{i+1}}\right] \leq \varepsilon$, then*

$$\Pr\left[R_s = 0\right] \leq \frac{\varepsilon s}{H_R}.$$

Proof. The premise implies by telescoping that $\mathbf{E}\left[H_R - H_{R_s}\right] \leq \varepsilon s$. By the definition of expectation, and since $H_R - H_{R_s}$ is non-negative, $\mathbf{E}\left[H_R - H_{R_s}\right] \geq \Pr\left[R_s = 0\right] \cdot H_R$ the inequality follows. ∎

When the folded rank is r and decreases by 1, the r'th harmonic number decreases by $1/r$. By Lemma 7.2, the expected decrease in H_r is $O(1/\,|\mathbb{F}|^2)$. By Lemma 7.3, the probability that the folded rank vanishes at depth $\Theta(\log(1/\delta)\,|\mathbb{F}|^2)$ is $O(\log(1/\delta)/\log R)$. For sufficiently large R, this probability is arbitrarily small. This is the upper bound p_0 we wanted to prove.

Soundness now follows via a standard Fourier analysis argument. Take any folded function $F : \mathcal{M} \to \mathbb{F}^3$ that passes the noise test with probability at least $1 - \varepsilon$. Recall that we consider $F_z(M) = (-1)^{\phi_z(F(M))}$ for linear $\phi_z : \mathbb{F}^3 \to \{0,1\}$. Write $F_z = \sum_S \hat{F}_z(S)\chi_S$ where the sum is taken only over folded characters. The acceptance probability of the noise test can be written as

$$\mathop{\mathbf{E}}_{z,M,N}\left[(-1)^{\phi_z(F(M)+F(M+N))}\right] \geq 1 - \varepsilon.$$

Hence we bound

$$\begin{aligned}
\mathop{\mathbf{E}}_{z,M,N}\left[F_z(M)F_z(M+N)\right] &\leq \mathop{\mathbf{E}}_{z,M,N}\left[\sum_{S,S'} \hat{F}_z(S)\chi_S(M)\hat{F}_z(S')\chi_{S'}(M+N)\right] \\
&= \mathop{\mathbf{E}}_{z,M,N}\left[\sum_{S,S'} \hat{F}_z(S)\hat{F}_z(S')\chi_S(M)\chi_{S'}(M)\chi_{S'}(N)\right] \\
&= \mathop{\mathbf{E}}_{z}\left[\sum_{S} \hat{F}_z(S)^2 \mathop{\mathbf{E}}_{N}\left[\chi_S(N)\right]\right].
\end{aligned}$$

Assume that R is sufficiently large as a function of δ, and $\mathbb{F}$ is sufficiently large as function of R. For every z, we have $\sum_S \hat{F}_z(S)^2 = 1$. By Lemma 7.1, for every z, the sum $\sum_S \hat{F}_z(S)^2 \mathbf{E}_N\left[\chi_S(N)\right]$ taken only over folded characters χ_S of folded-rank larger than R is at most p_0. For sufficiently small p_0, we have that $\mathbf{E}_z\left[\sum_S \hat{F}_z(S)^2\right]$ over those characters is $O(\varepsilon)$. This proves soundness.

8 Analyzing The Rank Decrease

In this section we prove Lemma 7.2. Recall that we consider a sub-tree corresponding to a node v_j, and we analyze binary linear functions $\psi_{j,i} : (\mathbb{F}^3)^{l_j} \to \{0,1\}$ for $i = 1, 2, \ldots, r$ on the labels to the leaves. The folded rank r is the dimension of the linear subspace spanned by the $\psi_{j,i}$ when restricted to H_{v_j}. Evaluated on H_{v_j} each of the $\psi_{j,i}$ is folded according to $\lambda_{v_j} : H_{v_j} \to \mathbb{F}^3$. That is, for $h, h' \in H_{v_j}$, if $\lambda_{v_j}(h) = 0$ then $\psi_{j,i}(h) = 0$, and if $\lambda_{v_j}(h) = \lambda_{v_j}(h')$ then $\psi_{j,j}(h) = \psi_{j,j}(h')$. Therefore, we can think of each $\psi_{j,i}$ as corresponding to a linear function $\mathbb{F}^3 \to \{0,1\}$. Such a linear function is of the form $x \mapsto \mathrm{Tr}(\langle z, x\rangle)$ for some $z \in \mathbb{F}^3$. Denote by $\mathcal{Z} \subseteq \mathbb{F}^3$ the binary space of z's corresponding to our $\psi_{j,i}$.

We pick a random child $i_j = (b, J_b)$ of v_j and assign 0 to all remaining children. In terms of $U_b \subseteq \mathbb{F}^3 \times \mathbb{F}^3$, each z-function becomes $(x, y) \mapsto \mathrm{Tr}(\langle z, y\rangle + \langle f(z), x\rangle)$ where $f : \mathcal{Z} \to \mathbb{F}^3$ is linear. What does it take for a z-function to vanish after the substitution? The substitution means that we evaluate only on inputs of the form $(tb, Q(tb) + w)$ for w orthogonal to b. Therefore, vanishing means that for every $t \in \mathbb{F}$ and $w \perp b$, we have:

$$\mathrm{Tr}(\langle z, (Q(tb) + w)\rangle + \langle f(z), tb\rangle) = 0.$$

To vanish we must have $\langle z, w\rangle = 0$ for all $w \perp b$, which means that z is on the line in direction b. Write $b = s \cdot z$ for $s \neq 0 \in \mathbb{F}$. Using that $Q(tb) = t^2 Q(b)$, for the rest of the terms to vanish we need

$$\mathrm{Tr}(st \cdot \langle f(z), z\rangle + (st)^2 \cdot \langle z, Q(z)\rangle) = 0.$$

In characteristic 2 we have $\mathrm{Tr}(t\alpha + t^2\beta) = \mathrm{Tr}(t(\alpha + \sqrt{\beta}))$. Hence, vanishing for all t requires

$$\langle f(z), z\rangle^2 = \langle z, Q(z)\rangle = z_1 z_2 z_3.$$

This motivates the following definition of a generic subspace:

Definition 7 (Generic subspace)**.** *We say that a binary linear subspace $\mathcal{Z} \subseteq \mathbb{F}^3$ of dimension r is* generic *if* for all *binary linear $f : \mathcal{Z} \to \mathbb{F}^3$, there are at most $3r$ elements $z \neq 0 \in \mathcal{Z}$ satisfying $\langle f(z), z\rangle^2 = z_1 z_2 z_3$, and there are no two distinct non-zero $z, z' \in \mathcal{Z}$ on the same line through the origin.*

Recall that a subspace $\mathcal{Z} \subseteq \mathbb{F}^3$ corresponds to a binary linear subspace spanned by $\psi_{j,i} : (\mathbb{F}^3)^{l_j} \to \{0,1\}$ evaluated on H_{v_j}. We use the term *generic* to refer to these subspaces too.

In Lemma 8.1 below we show that a random $\mathcal{Z}$ is generic with high probability. With it, we can prove Lemma 7.2 as follows:

1. We showed that for a z-function to vanish, the random $b \neq 0$ we pick must be on the same line through the origin as z. Since there are 2^r different $z \in \mathcal{Z}$ and $\Omega(|\mathbb{F}|^2)$ choices for b, the probability of any vanishing is at most $O(2^r / |\mathbb{F}^2|)$.

2. If $\mathcal{Z}$ is generic, then there are at most $3r$ z-functions for $z \neq 0 \in \mathcal{Z}$ that could vanish, and, since no two non-zero $z, z' \in \mathcal{Z}$ are on the same line through the origin, b can only make one vanish. The probability b would cause any of the $3r$ to vanish is $O(r / |\mathbb{F}|^2)$.

3. If r' is the number of z-functions that survive, they then get randomized via the random linear transformation J_b. Lemma 8.1 guarantees that the binary subspace they span $\mathcal{Z}'$ is generic except with probability $2^{O(r^2)}/|\mathbb{F}|$.

Lemma 8.1. *A binary subspace $\mathcal{Z} \subseteq \mathbb{F}^3$ spanned by r random vectors in $\mathbb{F}^3$ is generic, except with probability $2^{O(r^2)}/|\mathbb{F}|$.*

Note that for sufficiently large field $\mathbb{F}$, there are no two non-zero $z, z' \in \mathcal{Z}$ on the same line through the origin with high probability. This is because the number of $z \in \mathcal{Z}$ is at most 2^r, and for sufficiently large $\mathbb{F}$, we can union bound over all $z' \in \mathcal{Z}$. Similarly, for any fixed f, for a random $\mathcal{Z}$, with high probability, there are no $z \in \mathcal{Z}$ that satisfy $\langle f(z), z\rangle^2 = z_1 z_2 z_3$. The challenge is to prove that the same $\mathcal{Z}$ is likely good no matter what f is (since f depends on $\mathcal{Z}$). Crucially, the number of f's is $|\mathbb{F}|^{3r}$ and is too large for a union bound.

Proof. (of Lemma 8.1) Suppose that $\mathcal{Z}$ is defined by r elements in $\mathbb{F}^3$, or, alternatively, by $3r$ elements in the field $\mathbb{F}$. We define formal variables X_{ij} to denote these elements, where $i = 1, 2, 3,\ j = 1, \ldots, r$. We view each $z \in \mathcal{Z}$ as a linear combination $z_i = \sum_{j=1}^{r} t_j X_{ij}$ for $i = 1, 2, 3$ and $t \in \{0, 1\}^r$. Also, f is defined by $3r$ field elements. We'll have variables Y_{ij} to denote these elements, so $f(z)_i = \sum_{j=1}^{r} t_j Y_{ij}$ for $i = 1, 2, 3$ and $t \in \{0, 1\}^r$. Let $E \subseteq \{0, 1\}^r - \{0\}$ contain $3r + 1$ vectors t. We'll inspect the event that for all $t \in E$ the corresponding z satisfies $\langle f(z), z\rangle^2 = \langle z, Q(z)\rangle = z_1 z_2 z_3$. Here is our plan:

1. **Linear system:** Set up a system of $3r + 1$ linear equations in the variables $Y_{i,j}^2$'s that represents this event. The system will be over the function field $\mathbb{F}_2(\{X_{ij}\})$. Let $Ay = c$ denote this system. We'll see that the matrix A only involves squares X_{ij}^2 and quadratics $t_i t_j$ for $t \in E$, whereas the vector c only involves cubic terms in the X_{ij} and the t_i's. We will use this structure to argue that there cannot be a solution to the system when X_{ij} are formal variables.

2. **Determinant:** Denote the rank of A by d. It follows from 1 that c is linearly independent of the columns of A. This means that the rank of A with c appended as a last column is $d + 1$. Find a $(d + 1) \times (d + 1)$ sub-matrix A' of rank $d + 1$. Consider the determinant $det(A')$. This is a polynomial in X_{ij} of degree $O(r)$ that is not identically 0.

3. **Random substitution:** Substitute uniformly random field elements in the $X_{i,j}$. Except with probability $O(r/|\mathbb{F}|)$, the determinant of A' is non-zero, and therefore the c column remains independent of the A columns, so there is no solution for the linear system.

By a union bound over all $\binom{2^r-1}{3r+1}$ different choices of E, we get that except with probability $2^{O(r^2)}/|\mathbb{F}|$ over the choice of $\mathcal{Z}$, *for all* f, the number of $z \neq 0 \in \mathcal{Z}$ satisfying $\langle f(z), z\rangle^2 = z_1 z_2 z_3$ is at most $3r$.

To finish the proof, it remains to implement the first item on the list. We will have an equation in the linear system for every $t \in E$:

$$\left(\sum_{i=1}^{3}\left(\sum_{j=1}^{r} t_j X_{i,j}\right)\left(\sum_{l=1}^{r} t_l Y_{il}\right)\right)^2 = \left(\sum_{j=1}^{r} t_j X_{1,j}\right)\left(\sum_{k=1}^{r} t_k X_{2,k}\right)\left(\sum_{l=1}^{r} t_l X_{3,l}\right).$$

In characteristic 2 it holds that $(\alpha + \beta + \gamma)^2 = \alpha^2 + \beta^2 + \gamma^2$, since every cross term like $\alpha\beta$ appears twice and cancels out. Hence, the equation simplifies to:

$$\sum_{i=1}^{3}\left(\sum_{j=1}^{r} t_j X_{i,j}\right)^2\left(\sum_{l=1}^{r} t_l Y_{il}\right)^2 = \sum_{j,k,l} t_j t_k t_l X_{1,j} X_{2,k} X_{3,l}.$$

For binary t_i it holds that $t_i^2 = t_i$. Using this and the previous equality, we can further simplify to:

$$\sum_{i=1}^{3}\left(\sum_{j=1}^{r} t_j X_{i,j}^2\right)\left(\sum_{l=1}^{r} t_l Y_{il}^2\right) = \sum_{j,k,l} t_j t_k t_l X_{1,j} X_{2,k} X_{3,l}.$$

Equivalently,

$$\sum_{i=1}^{3}\sum_{l=1}^{r}\left(\sum_{j=1}^{r} t_l t_j X_{i,j}^2\right) Y_{il}^2 = \sum_{j,k,l} t_j t_k t_l X_{1,j} X_{2,k} X_{3,l}.$$

The matrix A has $3r+1$ rows corresponding to the $t \in E$. It has columns corresponding to the variables Y_{il}^2. The entry of the matrix A in row t and column (i,l) is therefore $\sum_{j=1}^{r} t_l t_j X_{i,j}^2$. The vector c in position t is $\sum_{j,k,l} t_j t_k t_l X_{1,j} X_{2,k} X_{3,l}$.

Let $\mathcal{T}_s$ be a subspace over $\mathbb{F}_2(\{X_{ij}\})$ that is spanned by monomials on between 1 and s variables evaluated on E, i.e., by all vectors of the form $(\prod_{i\in I} t_i)_{t\in E}$ where $I \subseteq \{1,\ldots,r\}$, $1 \le |I| \le s$. Note that all the columns of A only contain monomials on two variables (like $t_l t_j$ for $l \neq j$) or a single variable (when $l = j$, we have $t_l t_j = t_l$), and therefore the column space of A is contained in $\mathcal{T}_2$.

On the other hand, we'll show that, if there is a solution to $Ay = c$, the column space of A contains $\mathcal{T}_3$: Take a formal derivative of Ay in the formal variable $X_{1,j}$ to get

$$\frac{\partial(Ay)}{\partial X_{1,i}} = \frac{\partial A}{\partial X_{1,i}} y + A\frac{\partial y}{\partial X_{1,i}} = A\frac{\partial y}{\partial X_{1,i}},$$

where $\frac{\partial A}{\partial X_{1,i}} = 0$, since $X_{1,i}$ appears squared in A and the characteristic is 2. Further take formal derivatives in $X_{2,k}$ and $X_{3,l}$. The derivative is a vector in the column space of A, since

$$\frac{\partial^3(Ay)}{\partial X_{1,i}\partial X_{2,k}\partial X_{3,l}} = A\frac{\partial^3 y}{\partial X_{1,i}\partial X_{2,k}\partial X_{3,l}}.$$

On the other hand, since $Ay = c$, this vector equals

$$\frac{\partial^3 c}{\partial X_{1,i}\partial X_{2,k}\partial X_{3,l}} = t_j t_k t_l,$$

which proves that the column space of A contains $\mathcal{T}_3$ (note that some of j,k,l could be identical, leading to monomials with one or two variables). Since $\mathcal{T}_2 \subseteq \mathcal{T}_3$, the conclusion is that the column space of A must be equal to $\mathcal{T}_2 = \mathcal{T}_3$. The equality $\mathcal{T}_2 = \mathcal{T}_3$ also implies that $\mathcal{T}_2 = \mathcal{T}_s$ for all natural $s \ge 3$ (the proof is by induction on i; use $\mathcal{T}_2 = \mathcal{T}_3$ to replace a monomial with s variables with monomials with $s-1$ variables). This in turns implies that the column space of A spans $(h(t_1,\ldots,t_r))_{t\in E}$ for every function h on the r variables, since every such function is a linear combination of monomials on up to r variables: $h(t) = \sum_{a\in\{0,1\}^r} h(a)\cdot M_a(t)$ where $M_a(t)$ is a polynomial that is 1 iff $t = a$, i.e., $M_a(t) = \left(\prod_{i:a_i=1} t_i\right)\cdot\left(\prod_{i:a_i=0}(t_i+1)\right)$. Here we use that $0 \notin E$ so $M_a(t)$ for $t \in E$ does not involve constants (i.e., monomials on zero variables), which are not spanned. The dimension of the space over all h is $|E| = 3r+1$, however the column space of A has dimension at most $3r$ since it has $3r$ columns. This is a contradiction. ∎

9 Open Problems

Is there a simpler rounding for Hadamard that does not rely on the arbitrary quadratic mapping Q? Alternatively, can the rounding used in this proof be used elsewhere? Is there a proof of the Unique Games Theorem using the half-space code [25, 13] rather than the rounded Hadamard code?

References

[1] S. Arora, B. Barak, and D. Steurer. Subexponential algorithms for unique games and related problems. In *Proc. 51st IEEE Symp. on Foundations of Computer Science*, 2010.

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

[3] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[4] P. Austrin. Balanced max 2-SAT might not be the hardest. In *Proc. 39th ACM Symp. on Theory of Computing*, pages 189–197, New York, NY, USA, 2007. Association for Computing Machinery.

[5] L. Babai, L. Fortnow, L. A. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proc. 23rd ACM Symp. on Theory of Computing*, pages 21–32, 1991.

[6] L. Babai, L. Fortnow, and C. Lund. Nondeterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

[7] B. Barak, P. Kothari, and D. Steurer. Small-set expansion in shortcode graph and the 2-to-2 conjecture. Technical report, arXiv:1804.08662, 2018.

[8] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998.

[9] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993.

[10] M. Charikar, K. Makarychev, and Y. Makarychev. Near-optimal algorithms for unique games. In *Proc. 38th ACM Symp. on Theory of Computing*, pages 205–214, 2006.

[11] I. Dinur, S. Khot, G. Kindler, D. Minzer, and S. Safra. On non-optimally expanding sets in grassmann graphs. In *Proc. 50th ACM Symp. on Theory of Computing*, 2018.

[12] I. Dinur, S. Khot, G. Kindler, D. Minzer, and S. Safra. Towards a proof of the 2-to-1 games conjecture? In *Proc. 50th ACM Symp. on Theory of Computing*, pages 376–389, 2018.

[13] R. Eldan and D. Moshkovitz. Reduction from non-unique games to boolean unique games. In *ITCS*, 2022.

[14] S. Garibaldi, H. P. Petersson, , and Michel M. L. Racine. Albert algebras over $\mathbb{Z}$ and other rings. *Forum of Mathematics, Sigma*, 11:e18, 2023.

[15] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[16] S. Khot. On the power of unique 2-prover 1-round games. In *Proc. 34th ACM Symp. on Theory of Computing*, pages 767–775, 2002.

[17] S. Khot. Guest column: inapproximability results via long code based PCPs. *SIGACT News*, 36(2):25–42, 2005.

[18] S. Khot. Inapproximability of NP-complete problems, discrete Fourier analysis, and geometry. In *Proceedings of the International Congress of Mathematicians 2010*, volume 4, pages 2676–2697. Hindustan Book Agency, 2010.

[19] S. Khot. On the unique games conjecture (invited survey). In *IEEE Conference on Computational Complexity*, pages 99–121, 2010.

[20] S. Khot. Hardness of approximation. In *Proceedings of the International Congress of Mathematicians—Seoul 2014*, volume 1, pages 711–728. Kyung Moon Sa, 2014.

[21] S. Khot. On the proof of the 2-to-2 Games Conjecture. *Current Developments in Mathematics*, 2019(1):43–94, 2019.

[22] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other two-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007.

[23] S. Khot, D. Minzer, and S. Safra. On independent sets, 2-to-2 games, and grassmann graphs. In *Proc. 49th ACM Symp. on Theory of Computing*, pages 576–589, 2017.

[24] S. Khot, D. Minzer, and S. Safra. Pseudorandom sets in grassmann graph have near-perfect expansion. In *Proc. 59th IEEE Symp. on Foundations of Computer Science*, 2018.

[25] S. Khot and D. Moshkovitz. Hardness of approximately solving linear equations over reals. *SIAM Journal on Computing*, 42(3):752–791, 2013.

[26] S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2-epsilon. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.

[27] S. Khot and S. Safra. A two prover one round game with strong soundness. In *Proc. 52nd IEEE Symp. on Foundations of Computer Science*, pages 648–657, 2011.

[28] K. McCrimmon. The Freudenthal-Springer-Tits constructions of exceptional Jordan algebras. *Transactions of the American Mathematical Society*, 139:495–510, 1969.

[29] OpenAI. A direct proof of optimal max cut hardness. Technical report, https://github.com/openai/math/blob/main/preprints/A-Direct-Proof-of-Optimal-Max-Cut-Hardness-September-23-2026/paper.pdf, 2026.

[30] OpenAI. The factor-two hardness threshold for vertex cover. Technical report, https://github.com/openai/math/blob/main/preprints/The-Factor-Two-Hardness-Threshold-for-Vertex-Cover-September-23-2026/paper.pdf, 2026.

[31] OpenAI. The unique games theorem. Technical report, https://github.com/openai/math/blob/main/preprints/The-Unique-Games-Theorem-September-23-2026/paper.pdf, 2026.

[32] L. Pirio and F. Russo. Quadro-quadric Cremona transformations in low dimensions via the JC-correspondence. *Annales de l'Institut Fourier*, 64(1):71–111, 2014.

[33] P. Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proc. 40th ACM Symp. on Theory of Computing*, pages 245–254, 2008.

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

[35] R. Raz. A parallel repetition theorem. In *SIAM Journal on Computing*, volume 27, pages 763–803, 1998.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il