Hardness of Euclidean Closest Vector within $n^{1/2-\epsilon}$ and Binary Nearest Codeword within $n^{1-\epsilon}$

Zhao Song*

August 7, 2026

Abstract

We prove two deterministic inapproximability results.

First, for every fixed $0 < \epsilon < 1/2$, Euclidean GapCVP$^{(2)}$ is NP-hard with gap factor $n^{1/2-\epsilon}$ under deterministic polynomial-time many-one reductions, where n denotes the lattice rank. Consequently, the Euclidean closest vector problem is NP-hard to approximate within the same factor. This improves the previous $n^{1/400}$ hardness factor in Chapter 7 of the OpenAI report [Ope26]. Aharonov and Regev gave short certificates for both the YES and the NO case of GapCVP$^{(2)}$ at gap factor $C\sqrt{n}$, placing that problem in NP $\cap$ coNP for an absolute constant $C > 0$ [AR05]. An NP-hard problem lying in coNP would give NP = coNP, so the factor $n^{1/2-\epsilon}$ above cannot be improved to $C\sqrt{n}$ unless the two classes coincide.

Second, for every fixed $0 < \epsilon < 1$, the gap versions of binary nearest codeword and binary syndrome decoding are NP-hard with factor $n^{1-\epsilon}$ under deterministic polynomial-time many-one reductions, where n denotes the binary block length. Consequently, both optimization problems are NP-hard to approximate within the same factor. This improves the previous $n^{1/200}$ hardness factor in Chapter 7 of the OpenAI report [Ope26].

*magic.linuxkde@gmail.com. The author would like to thank Josh Alman and Omri Weinstein for helpful discussions. The author would like to thank Johan Håstad for providing several useful writing suggestions.

1 Introduction

The gap version of the closest vector problem provides the standard decision formulation for studying approximation hardness. Let $B \in \mathbb{Z}^{n\times n}$ be nonsingular. Its columns generate the full-rank lattice $\mathcal{L}(B) := B\mathbb{Z}^n$, whose rank is n. Given a target $t \in \mathbb{Q}^n$ and a norm $\|\cdot\|$, the optimization version of the closest vector problem asks for $z \in \mathbb{Z}^n$ minimizing $\|t - Bz\|$; see Micciancio and Goldwasser [MG02, Chapter 3] for standard background.

For an approximation factor $\gamma : \mathbb{N} \to [1, \infty)$ and a positive rational radius r, the promise problem $\mathrm{GapCVP}_\gamma^{\|\cdot\|}$ asks one to distinguish

$$\mathrm{dist}_{\|\cdot\|}(t, \mathcal{L}(B)) \le r \quad \text{from} \quad \mathrm{dist}_{\|\cdot\|}(t, \mathcal{L}(B)) > \gamma(n)r,$$

where $\mathrm{dist}_{\|\cdot\|}(t, \mathcal{L}(B)) := \min_{z\in\mathbb{Z}^n} \|t - Bz\|$. Instances whose optimum lies between the two thresholds are outside the promise. A $\gamma(n)$-approximation algorithm for the optimization problem solves this promise problem by comparing the distance of its output with $\gamma(n)r$. Thus NP-hardness of $\mathrm{GapCVP}_\gamma^{\|\cdot\|}$ rules out such an approximation algorithm unless $\mathrm{P} = \mathrm{NP}$.

The Euclidean closest vector problem is the specialization to the ℓ_2 norm. We write

$$\mathrm{dist}_2(t, \mathcal{L}(B)) := \min_{z\in\mathbb{Z}^n} \|t - Bz\|_2$$

and denote its gap version by $\mathrm{GapCVP}_\gamma^{(2)}$. Van Emde Boas [vEB81] proved NP-hardness of exact Euclidean CVP. Arora, Babai, Stern, and Sweedyk [ABSS97] established hardness within every fixed constant. For every fixed $\epsilon > 0$, they also proved hardness within $2^{(\log n)^{1-\epsilon}}$ under the assumption $\mathrm{NP} \not\subseteq \mathrm{DTIME}(2^{(\log n)^{O(1)}})$ [ABSS97]. Dinur, Kindler, and Safra [DKS98], in work subsequently refined by Dinur, Kindler, Raz, and Safra [DKRS03], strengthened this to $n^{a/\log\log n}$ for some absolute constant $a > 0$. Although this factor dominates every fixed power of $\log n$, its exponent tends to zero and hence does not give n^c hardness for any fixed $c > 0$. The OpenAI report [Ope26, Chapter 7] subsequently obtained the fixed polynomial factor $n^{1/400}$ by a deterministic reduction from 3SAT. Fixed-polynomial lattice inapproximability had previously been obtained conditionally on the Projection Games Conjecture by Moshkovitz [Mos15, Section 5.1] and Mukhopadhyay [Muk22]. Those conditional results are qualitatively different from the unconditional deterministic reduction from ordinary 3SAT proved here. The Exponential Time Hypothesis (ETH), introduced by Impagliazzo and Paturi [IP01], has also served as the basis for recent fine-grained lower bounds for constant-factor approximations of CVP and related problems [AGMZ26, HKW26]. These results concern the running time of constant-factor approximation rather than unconditional NP-hardness for polynomial approximation factors.

The square-root scale forms a complexity-theoretic barrier. Goldreich and Goldwasser [GG00] established interactive-proof upper bounds that constrain the nonapproximability of lattice problems. Aharonov and Regev [AR05] subsequently proved that, for some absolute constant $C > 0$,

$$\mathrm{GapCVP}_{C\sqrt{n}}^{(2)} \in \mathrm{NP} \cap \mathrm{coNP}.$$

Consequently, NP-hardness at that scale under deterministic polynomial-time many-one reductions would imply $\mathrm{NP} = \mathrm{coNP}$. The reduction given here achieves every fixed exponent below $1/2$.

Binary nearest codeword is an optimization problem over $\mathbb{F}_2$. Given a k-dimensional binary linear code $C \subseteq \mathbb{F}_2^n$ and a target word $y \in \mathbb{F}_2^n$, the task is to find a codeword in C minimizing its Hamming distance from y [BMT78]. Binary syndrome decoding takes as input a binary parity-check matrix H and a syndrome $s \in \mathrm{im}\, H$, and asks for a minimum-weight binary vector e satisfying

$He = s$ [BMT78]. The two formulations are equivalent. Indeed, if $C = \ker H$ and $s = Hy$, then the map $c \mapsto y - c$ identifies codewords $c \in C$ with vectors e satisfying $He = s$. Consequently, the two instances have the same optimum value, and the correspondence preserves approximation ratios. Conversely, given a consistent syndrome instance (H, s), Gaussian elimination finds y with $Hy = s$, and the same correspondence reduces it to nearest codeword for $\ker H$.

Berlekamp, McEliece, and van Tilborg [BMT78] proved that binary syndrome decoding is NP-complete. For binary nearest codeword, Arora, Babai, Stern, and Sweedyk [ABSS97] proved unconditional NP-hardness of approximation within every constant factor. For every fixed $\epsilon > 0$, they also proved hardness within $2^{(\log n)^{1-\epsilon}}$ under the assumption that NP has no quasipolynomial-time algorithms, namely NP $\not\subseteq$ DTIME$(2^{(\log n)^{O(1)}})$. Arora's thesis [Aro94, Section 6.4] gives a contemporaneous account and explicitly records the stronger $1 - \epsilon$ exponent for nearest codeword. For nearest codeword with preprocessing, Alekhnovich, Khot, Kindler, and Vishnoi [AKKV05] proved hardness within $(\log n)^{1-\epsilon}$ under the same quasipolynomial-time assumption. Khot, Popat, and Vishnoi [KPV14] strengthened the factor to $2^{(\log n)^{1-\epsilon}}$ under NP $\not\subseteq$ DTIME$(2^{(\log n)^{O(1/\epsilon)}})$. For ordinary nearest codeword, Bhattiprolu, Guruswami, and Ren [BGR25] subsequently gave a PCP-free proof of the same two bounds: unconditional NP-hardness within every constant factor and $2^{(\log n)^{1-\epsilon}} = n^{o(1)}$ hardness under the assumption that NP has no quasipolynomial-time algorithms. Their contribution is a new proof rather than a stronger approximation factor. Bhattiprolu, Guruswami, Lee, and Ren [BGLR25] subsequently studied the inapproximability of finding sparse vectors in codes, subspaces, and lattices via randomized reductions; their lattice result does not yield deterministic fixed-polynomial hardness for Euclidean CVP. Methodologically, Bennett and Peikert [BP23] also combine Reed–Solomon codes [RS60] with an integer-lattice lift to obtain randomized hardness for approximate SVP and identify derandomization as a central obstacle.

Bitansky, Harsha, Ishai, Rothblum, and Wu [BHI+24] showed that ETH rules out $2^{o(n)}$-time approximation within a certain constant factor, using the equivalent formulation that minimizes the number of unsatisfied linear equations. Subsequently, Corollary 15 of the OpenAI report [Ope26] established, without an additional complexity assumption, NP-hardness of approximation within $n^{1/200}$ for binary nearest codeword and binary syndrome decoding under deterministic polynomial-time many-one reductions. The result proved here uses the same unconditional reduction model and improves the approximation factor to $n^{1-\epsilon}$ for every fixed $0 < \epsilon < 1$.

On the algorithmic side, Alon, Panigrahy, and Yekhanin gave several deterministic approximation algorithms and time–ratio tradeoffs [APY10]. Their general polynomial-time algorithm achieves an $O(n/\log n)$ approximation. Berman and Karpinski gave, for every fixed constant $c > 0$, a deterministic algorithm with approximation factor $O(k/c)$ and a randomized algorithm with factor $O(k/\log k)$ [BK02]. Alon, Panigrahy, and Yekhanin observed that the latter analysis yields approximation factor $O(k/\log n)$ [APY10].

1.1 Our Results

We first record the two Euclidean lattice problems used below.

Definition 1.1 (Euclidean closest vector)**.** *An instance consists of a nonsingular matrix $B \in \mathbb{Z}^{n\times n}$ and a target $t \in \mathbb{Q}^n$. The matrix B generates the full-rank lattice $\mathcal{L}(B) := B\mathbb{Z}^n$. The objective is to output a vector $z \in \mathbb{Z}^n$ minimizing $\|t - Bz\|_2$. Its optimum value is* $\mathrm{dist}_2(t, \mathcal{L}(B)) := \min_{z\in\mathbb{Z}^n} \|t - Bz\|_2$.

Definition 1.2 (Euclidean GapCVP)**.** *Let $\gamma : \mathbb{N} \to [1, \infty)$. An instance of* $\mathrm{GapCVP}^{(2)}_\gamma$ *consists of a nonsingular matrix $B \in \mathbb{Z}^{n\times n}$, a target $t \in \mathbb{Q}^n$, and a radius $r \in \mathbb{Q}_{>0}$. It is promised that either*

$\mathrm{dist}_2(t, \mathcal{L}(B)) \le r$, *which is the YES case, or* $\mathrm{dist}_2(t, \mathcal{L}(B)) > \gamma(n)r$, *which is the NO case. The task is to distinguish the two cases.*

We now state our main Euclidean result. It improves the $n^{1/400}$ hardness factor in Theorem 1 on page 184 of the OpenAI report [Ope26, Chapter 7].

Theorem 1.3 (Euclidean closest vector, informal version of Theorem 8.1)**.** *For every fixed* $0 < \epsilon < 1/2$*, there is a deterministic polynomial-time mapping that assigns to each 3SAT formula* φ *a nonsingular matrix* $B \in \mathbb{Z}^{n\times n}$*, a target* $t \in \mathbb{Z}^n$*, and a radius* $r \in \mathbb{Q}_{>0}$ *such that*

$$\varphi \text{ satisfiable} \implies \mathrm{dist}_2(t, \mathcal{L}(B)) \le r,$$
$$\varphi \text{ unsatisfiable} \implies \mathrm{dist}_2(t, \mathcal{L}(B)) > n^{1/2-\epsilon} r.$$

Consequently, $\mathrm{GapCVP}^{(2)}_{n^{1/2-\epsilon}}$ *is NP-hard under deterministic polynomial-time many-one reductions.*

Remark 1.4. Our Theorem improves the $n^{1/400}$ Euclidean closest-vector hardness factor established in Theorem 1 of the OpenAI report [Ope26] to $n^{1/2-\epsilon}$. Aharonov and Regev proved that $\mathrm{GapCVP}^{(2)}_{C\sqrt{n}}$ lies in $\mathrm{NP} \cap \mathrm{coNP}$ for an absolute constant $C > 0$ [AR05]. No NP-hardness result under deterministic polynomial-time many-one reductions can reach that scale unless $\mathrm{NP} = \mathrm{coNP}$.

We next formally define the two binary problems.

Definition 1.5 (Binary nearest codeword)**.** *An instance consists of a binary linear code* $C \subseteq \mathbb{F}_2^n$*, given by a generator matrix, and a target word* $y \in \mathbb{F}_2^n$*. The objective is to output a codeword* $c \in C$ *minimizing* $\mathrm{wt}(y - c)$*. We denote the optimum value by* $\mathrm{opt}_{\mathrm{nc}}(C, y) := \min_{c\in C} \mathrm{wt}(y - c)$.

Definition 1.6 (Binary syndrome decoding)**.** *An instance consists of a binary matrix* $H \in \mathbb{F}_2^{r\times n}$ *and a syndrome* $s \in \mathrm{im}\, H$*. The objective is to output a vector* $e \in \mathbb{F}_2^n$ *of minimum Hamming weight subject to* $He = s$*. We denote the optimum value by* $\mathrm{opt}_{\mathrm{sd}}(H, s) := \min\{\mathrm{wt}(e) : e \in \mathbb{F}_2^n,\ He = s\}$.

To state approximation hardness, we use the gap versions of these two optimization problems. Both are promise problems parameterized by a radius and an approximation factor.

Definition 1.7 (Binary gap nearest codeword and gap syndrome decoding)**.** *Let* $\gamma : \mathbb{N} \to [1, \infty)$*. An instance of* GapNCP_γ *is a binary nearest-codeword instance* (C, y) *together with a positive integer radius* R*; it is promised that either* $\mathrm{opt}_{\mathrm{nc}}(C, y) \le R$ *or* $\mathrm{opt}_{\mathrm{nc}}(C, y) > \gamma(n)R$*. An instance of* GapSD_γ *is a binary syndrome-decoding instance* (H, s) *together with a positive integer radius* R *and the analogous promise* $\mathrm{opt}_{\mathrm{sd}}(H, s) \le R$ *or* $\mathrm{opt}_{\mathrm{sd}}(H, s) > \gamma(n)R$*. In both cases* n *is the binary block length, and the task is to distinguish the two promised cases.*

The following theorem improves the $n^{1/200}$ hardness factor established by Corollary 15 on page 213 of the OpenAI report [Ope26, Chapter 7].

Theorem 1.8 (Main result, informal version of Theorem 7.1)**.** *For every fixed* $0 < \epsilon < 1$*,* $\mathrm{GapNCP}_{n^{1-\epsilon}}$ *and* $\mathrm{GapSD}_{n^{1-\epsilon}}$ *are NP-hard under deterministic polynomial-time many-one reductions, where* n *is the binary block length. Consequently, binary nearest codeword (Definition 1.5) and binary syndrome decoding (Definition 1.6) are NP-hard to approximate within* $n^{1-\epsilon}$.

Remark 1.9. The general polynomial-time algorithm of Alon, Panigrahy, and Yekhanin [APY10] approximates binary nearest codeword within $O(n/\log n)$, so the factor $n^{1-\epsilon}$ is optimal up to n^{ϵ}. This improves the $n^{1/200}$ hardness factor established in Corollary 15 of Chapter 7 of the OpenAI report [Ope26].

Table 1: Historical hardness lower bounds and algorithmic upper bounds for Euclidean closest vector, measured as a function of the lattice rank n. In the second panel, D and R denote deterministic and randomized algorithms, respectively, and poly abbreviates polynomial time. In the 1997 row, the constant-factor hardness is unconditional, whereas the $2^{(\log n)^{1-\epsilon}}$ factor assumes NP $\not\subseteq$ DTIME($2^{(\log n)^{O(1)}}$). In the 2003 row, $a > 0$ is an absolute constant. The final hardness row holds for every fixed $0 < \epsilon < 1/2$.

(a) Hardness lower bounds

Year	Authors	Reference	Ratio
1997	Arora, Babai, Stern, and Sweedyk	[ABSS97]	Every constant; $2^{(\log n)^{1-\epsilon}}$
2003	Dinur, Kindler, Raz, and Safra	[DKRS03]	$n^{a/\log\log n}$
2026	OpenAI	[Ope26]	$n^{1/400}$
2026	This paper	Theorem 1.3	$n^{1/2-\epsilon}$

(b) Algorithmic upper bounds

Year	Authors	Reference	R/D	Time	Ratio
1986	Babai	[Bab86]	D	poly	$2^{n/2}$
1987	Schnorr	[Sch87]	D	poly	$2^{O(n(\log\log n)^2/\log n)}$
2001	Ajtai, Kumar, and Sivakumar	[AKS01]	R	poly	$2^{O(n\log\log n/\log n)}$
2013	Micciancio and Voulgaris	[MV13]	D	poly	$2^{O(n\log\log n/\log n)}$

2 Technique Overview

This section compares the mechanism behind the OpenAI reduction [Ope26, Chapter 7] with the two quadratic estimates that permit a sharper choice of parameters. We focus on the reduction to binary nearest codeword and syndrome decoding (Theorem 1.8). The Euclidean closest-vector result (Theorem 1.3) then follows from the dimension-preserving parity-lift reduction, which converts a binary weight gap into the square root of that gap in Euclidean distance. Section 2.1 reviews the approach in the OpenAI report, whereas Section 2.2 explains how our sharper analysis improves the resulting hardness factors.

2.1 Overview of the OpenAI Approach

Chapter 7 of the OpenAI report [Ope26, Chapter 7, Section 1.2] factors the reduction as

$$\text{3SAT} \longrightarrow \text{binary affine decoding} \longrightarrow \text{binary nearest codeword} \longrightarrow \text{Euclidean CVP}.$$

The first arrow contains the main algebraic construction. Let φ be the input formula, let s be its encoding length, and let m and ℓ be its numbers of variables and clauses. We define $N := 100 + s + m + \ell$ and $D_{\mathrm{RS}} := m$. The variables are indexed by $[m] := \{1, \dots, m\}$ and represented by distinct anchors $a_1, \dots, a_m$ in a characteristic-two field $\mathbb{F}_q$. A Boolean assignment $\sigma \in \{0,1\}^m$ determines the unique polynomial $Q_\sigma \in \mathbb{F}_q[X]$ of degree less than m satisfying $Q_\sigma(a_i) = \sigma_i$, where σ_i is the bit assigned to variable i. The reduction evaluates this polynomial on $P := \mathbb{F}_q \setminus \{a_1, \dots, a_m\}$.

The binary variables are arranged into evaluation tables. For a clause C, let $I_C \subseteq [m]$ be the set of its variable indices and let $B_C \subseteq \{0,1\}^{I_C}$ be its satisfying local assignments. The table types are $\Theta := \{0\} \cup \{(C, \beta) : \beta \in B_C\}$: type 0 is the global table, and (C, β) is the subtype asserting that clause C receives the local assignment β. Write $x_{\tau,p,w}$ for the binary coordinate selecting the field value $w \in \mathbb{F}_q$ in table $\tau \in \Theta$ at $p \in P$, and define its fiber by $S_\tau(p) := \{w \in \mathbb{F}_q : x_{\tau,p,w} = 1\}$.

Table 2: Historical hardness lower bounds and algorithmic upper bounds for binary nearest codeword. In the first panel, the 2005 and 2014 rows concern nearest codeword with preprocessing; all other rows concern ordinary nearest codeword or the equivalent syndrome-decoding formulation. The constant-factor statements in the 1997 and 2025 rows are unconditional. The $2^{(\log n)^{1-\epsilon}}$ statements in those rows, as well as the 2005 row, assume NP $\not\subseteq$ DTIME($2^{(\log n)^{O(1)}}$); the 2014 row assumes NP $\not\subseteq$ DTIME($2^{(\log n)^{O(1/\epsilon)}}$). Both 2026 hardness results are unconditional, and the final row holds for every fixed $0 < \epsilon < 1$. In the second panel, n denotes the binary block length, k denotes the code dimension, and R and D denote randomized and deterministic algorithms, respectively. The randomized 2002 row records the sharper $O(k/\log n)$ guarantee that Alon, Panigrahy, and Yekhanin observed from Berman and Karpinski's analysis; Berman and Karpinski themselves stated $O(k/\log k)$. In the first row of the second panel, $c > 0$ is an arbitrary fixed constant. The parameter s is a fixed positive integer, and $\log^{(s)} n$ denotes the s-fold iterated logarithm.

(a) Hardness lower bounds

Year	Authors	Reference	Ratio
1997	Arora, Babai, Stern, and Sweedyk	[ABSS97]	Every constant; $2^{(\log n)^{1-\epsilon}}$
2005	Alekhnovich, Khot, Kindler, and Vishnoi	[AKKV05]	$(\log n)^{1-\epsilon}$
2014	Khot, Popat, and Vishnoi	[KPV14]	$2^{(\log n)^{1-\epsilon}}$
2025	Bhattiprolu, Guruswami, and Ren	[BGR25]	Every constant; $2^{(\log n)^{1-\epsilon}}$
2026	OpenAI	[Ope26]	$n^{1/200}$
2026	This paper	Theorem 1.8	$n^{1-\epsilon}$

(b) Algorithmic upper bounds

Year	Authors	Reference	R/D	Time	Ratio
2002	Berman and Karpinski	[BK02]	D	poly	$O(k/c)$
2002	Berman and Karpinski	[BK02, APY10]	R	poly	$O(k/\log n)$
2010	Alon, Panigrahy, and Yekhanin	[APY10]	D	poly	$O(n/\log n)$
2010	Alon, Panigrahy, and Yekhanin	[APY10]	D	$n^{O(s)}$	$O(k \log^{(s)} n/\log n)$
2010	Alon, Panigrahy, and Yekhanin	[APY10]	D	$n^{O(\log^* n)}$	$O(k/\log n)$

Fix a moment budget T. For $0 \le j \le T$, the ordinary power sum of table τ at p is

$$\mu_{\tau,j}(p) := \sum_{w \in S_\tau(p)} w^j.$$

For $\tau = (C, \beta)$, $i \in I_C$, and $0 \le j \le T$, the shifted power sum is

$$\eta_{\tau,i,j}(p) := \sum_{w \in S_\tau(p)} \left(\frac{w - \beta_i}{p - a_i}\right)^j,$$

where β_i is the bit assigned to variable i by β. The affine constraints require these pointwise quantities to be evaluations of low-degree polynomials and require the clause tables to reproduce the global table modulo two. Let $Hx = b$ denote the resulting binary affine system, let $M := |\Theta||P|q$ be its number of binary coordinates, and, when the system is consistent, define $W(H, b) := \min\{\mathrm{wt}(x) : Hx = b\}$. If σ satisfies the formula, each active fiber is the singleton $\{Q_\sigma(p)\}$, exactly one subtype is active for every clause, and the resulting binary solution has weight $R_1 := (\ell + 1)|P|$.

For soundness, start with a low-weight binary solution. For a fiber cutoff κ, define the good evaluation set of table τ by $P_\tau := \{p \in P : |S_\tau(p)| \le \kappa\}$. Markov's inequality shows that P_τ contains all but a small set of evaluation points. Reed–Solomon interpolation turns the pointwise

moments into polynomials $\mu_{\tau,j}(X)$ in the indeterminate X. Lemma 10 of the OpenAI report [Ope26, Chapter 7, Lemma 10] uses the associated Hankel matrix to construct a monic separable polynomial $G_\tau \in \mathbb{F}_q(X)[Y]$. Its roots in a common splitting field have the prescribed power sums. The shifted moments are then examined using a valuation above $X - a_i$. They force every root associated with subtype (C, β) to reduce to the bit β_i at the anchor a_i. Finally, the clause decomposition and a Vandermonde argument match a global root with a satisfying subtype for every clause. Using the same valuation for a variable wherever it occurs makes these local assignments consistent, producing a global satisfying assignment.

The three parameters κ, T, and q serve different purposes, but the soundness proof couples them. The cutoff κ must be large enough that a low-weight solution has few heavy fibers: if $\mathrm{wt}(x) \le 4M^{1/200}R_1$, then

$$|P \setminus P_\tau| \le \frac{4M^{1/200}R_1}{\kappa}.$$

Increasing κ improves this estimate, but it makes both algebraic steps more expensive. Thus one cannot choose the cutoff, the moment budget, and the field size independently.

Lemma 12 of the OpenAI report [Ope26, Chapter 7, Lemma 12] uses a large moment budget because of its local valuation argument. For a reconstructed polynomial of degree $h \le \kappa$, it first bounds the coefficients of an inverse Vandermonde matrix. In the notation of that argument, the successive bounds are

$$D_0 := D_{\mathrm{RS}}h^2 + h, \qquad U_h := h^2 D_0 + D_{\mathrm{RS}}h^2 + 1, \qquad z := hU_h + 1.$$

With $D_{\mathrm{RS}} \le N$ and $h \le \kappa = N^4$, these quantities have scales N^9, N^{17}, and N^{21}, respectively. The proof uses the moments with indices from z through $z + h - 1$ to isolate one root at a time. Consequently, the moment budget must be larger than this range; the choice $T = N^{30}$ provides ample slack. This is the local-valuation moment cost; it does not come from the Reed–Solomon encoding itself.

Lemma 10 of the OpenAI report [Ope26, Chapter 7, Lemma 10] imposes two further costs, both quadratic in the fiber cutoff. Its Hankel determinant Δ_h has degree at most $D_{\mathrm{RS}}h(h-1)$; hence, uniformly for $h \le \kappa$, as many as $D_{\mathrm{RS}}\kappa(\kappa-1)$ retained sample points may fail to have maximum fiber size. Moreover, after clearing the common denominator Δ_h, the j-th root-moment discrepancy has degree at most $2D_{\mathrm{RS}}h^2 j$, which is bounded by $2D_{\mathrm{RS}}\kappa^2 T$ for $j \le T$. Thus the reconstruction lemma requires

$$|P_\tau| - D_{\mathrm{RS}}\kappa(\kappa-1) > 2D_{\mathrm{RS}}\kappa^2 T. \tag{1}$$

We call the subtractive term the determinant-zero cost and the right-hand term the denominator-clearing cost. For $D_{\mathrm{RS}} \le N$, $\kappa = N^4$, and $T = N^{30}$, the right-hand side is of order N^{39}. Hence the field must be substantially larger than N^{39}. Moreover, it must remain large after the heavy fibers are discarded. Taking $q = \Theta(N^{200})$ makes both requirements hold with considerable uniform slack: since $M = O(Nq^2)$ and $R_1 = O(Nq)$, the preceding discarded-point bound is $o(q)$ for the target factor $M^{1/200}$. The exact exponents 30 and 200 are therefore deliberately generous choices, but the original proof genuinely requires a large separation among these scales.

Finally, $q = \Theta(N^{200})$ gives $M = O(Nq^2) = O(N^{401})$. Soundness against solutions of weight $O(M^{1/200}R_1)$ yields the $M^{1/200}$ coding gap. The standard parity lift outputs a lattice basis B and a target t satisfying $\mathrm{dist}_2(t, \mathcal{L}(B))^2 = W(H, b)$, so this becomes the $M^{1/400}$ Euclidean gap.

2.2 Overview of Our Approach

We retain the outer reduction through a sparse binary affine system, but replace its inner encoding and soundness mechanism. Multi-axis exponent reconstruction replaces anchor-based interpolation,

and soundness is proved by a high-rank/low-rank dichotomy combining rank charging with short-recurrence atomization.

Sparse quadratic encoding and assignment compression. Instead of interpolating an assignment by a polynomial through anchors, we first replace the formula by a system of homogeneous quadratic forms in which each form has at most five monomials (Lemma 3.2), and we then compress an assignment into one field element, $\mathrm{enc}(x) := \sum_{i=1}^{n} y_i x_i \in \mathbb{E}$ where the y_i are independent over $\mathbb{F}_2$, so that $x \mapsto \mathrm{enc}(x)$ is injective. Consistency between occurrences of a variable is enforced by linear equations in the quadratic encoding rather than by a valuation at an anchor, so no local analysis at anchors is needed at any point below.

Axis-wise interpolation and multi-axis exponent reconstruction. Fix a power of two d and a number of axes r, and put $\kappa := d^r$. For each axis j the reduction uses the polynomial $P_{x,j}$ of degree less than $[\mathbb{E} : \mathbb{F}_q]$ with $P_{x,j}(\zeta) = \mathrm{enc}(x)^{d^j}$, which exists because $\{1, \zeta, \ldots\}$ is a basis, and whose coefficients are $\mathbb{F}_2$-linear in x because d^j is a power of two and the Frobenius map is additive (Lemma 3.3). Writing an exponent in base d as $k = \sum_{h=0}^{r-1} p_h d^h$ gives $\prod_{h=0}^{r-1} P_{x,h}(\zeta)^{p_h} = \mathrm{enc}(x)^k$ by Lemma 3.4. This is the quantitative heart of the construction: r axis polynomials represent $\kappa = d^r$ consecutive powers of the assignment encoding, while the interpolation degree needed to carry them grows only linearly in d. Thus the construction represents d^r consecutive powers while the interpolation-degree dependence on the radix remains linear in d.

The table has one block $\mathcal{T}_{i,j,\ell,s}$ for every pair $0 \le i, j \le n$, every label ℓ in the label set $\mathcal{L} := \mathbb{F}_q^r$, and every sample point s in the sample set $\mathcal{S} := \mathbb{F}_q$; for fixed ℓ and s these form a block $\mathcal{T}_{\ell,s} \in \mathbb{F}_2^{(n+1)\times(n+1)}$.

Definitions 3.7 and 3.9 introduce the derived quantities f, F, and u.

From interpolants to exponent-indexed moments. Definition 3.8 controls how the table is converted into scalar moments. The degree bound keeps interpolation within the available budget, while the consistency condition ensures that $F_{i,j,p}(\zeta)$ depends only on the scalar exponent $\mathrm{ind}(p) = \sum_{h=0}^{r-1} p_h d^h$. Thus the exponent-indexed moments $u_{i,j,k}$ are well defined.

Algebraic identities of honest moments. Definition 3.10 imposes the three algebraic identities satisfied by honest moments. Doubling an exponent squares the moment, the linear-combination identity advances the exponent through the assignment encoding, and matrix-inner-product vanishing enforces every quadratic encoding form. These constraints are affine over $\mathbb{F}_2$; in particular, squaring is affine because it is the Frobenius automorphism (Lemma 3.12).

Direct constraints on table blocks. Definition 3.11 acts directly on the table blocks. Symmetry makes each block behave like a matrix of pairwise products, while anchor normalization fixes the homogenizing coordinate and removes the scaling ambiguity. Accordingly, the completeness witness places one rank-one block $zz^\top$ at the honest label of each sample point (Lemma 3.13).

Charging rank instead of discarding a discriminant. Soundness splits on the rank of the polynomial moment matrix $\mathbf{U}(X)$ over $\mathbb{K}(X)$. If that rank is at least κ, we do not ask any determinant to be nonzero at any particular sample point. Instead we choose one $\kappa \times \kappa$ minor with nonzero determinant Δ, note that the rank lost at a sample point s divides Δ to that order, and sum:

$$\sum_{s\in\mathcal{S}} \delta_s \le \deg \Delta \le \kappa W_{\deg}, \qquad \mathrm{wt}(\mathcal{T}) \ge \sum_{s\in\mathcal{S}} \mathrm{rank}\, \mathbf{U}(s) \ge \kappa(q - W_{\deg}),$$

Here the Kronecker factorization $\mathbf{U}(s) = \sum_{\ell\in\mathcal{L}} \mathcal{T}_{\ell,s} \otimes w_\ell (w'_\ell)^\top$ gives the first inequality (Lemma 4.1). Determinantal divisibility gives the second: if $\delta_s := \kappa - \mathrm{rank}\, C(s)$, then $(X-s)^{\delta_s} \mid \Delta$, and therefore $\sum_s \delta_s \le \deg \Delta \le \kappa W_{\deg}$ (Lemma 4.3). For fixed r, $W_{\deg} = 2r(n+2)(d-1)(c-1)$ is linear in the radix d, equivalently proportional to $\kappa^{1/r}$ up to the displayed n, c, r factors; in particular, no quadratic κ-factor appears. Thus the high-rank branch replaces the determinant-zero cost $D_{\mathrm{RS}}\kappa(\kappa-1)$ in

Eq. (1). Rather than discarding every sample point where a maximal-fiber Hankel determinant vanishes, it charges each pointwise rank deficit to the vanishing order of one fixed nonzero minor.

Atomization instead of local valuation. In the complementary case the rank is below κ, so every scalar moment sequence obeys a short linear recurrence (Lemma 5.2). View the sequence as a functional λ on the finite-dimensional recurrence algebra it generates. Before passing to the reduced algebra, generalized nilpotent components may still occur, while arbitrary coefficients could permit cancellation between candidate assignments. The doubling-is-squaring identity in Definition 3.10-(a) removes exactly these freedoms: the identity $\lambda(\psi^2) = \lambda(\psi)^2$ kills the nilradical and forces the idempotent values to satisfy $c^2 = c$, hence to lie in $\{0, 1\}$. So every moment sequence is a plain sum $\sum_{\xi \in \Xi_{i,j}} \xi^k$ over an entrywise set of distinct atoms with all coefficients equal to one (Lemma 5.3). Constraint-local Vandermonde isolation then synchronizes these entrywise atom sets and decodes a satisfying assignment (Lemma 6.2). This low-rank branch avoids the denominator-clearing cost $2D_{\mathrm{RS}}\kappa^2 T$ in Eq. (1): the atom identities are proved inside the recurrence algebra and synchronized by a constraint-local Vandermonde system, rather than by inverse-Vandermonde local valuation followed by interpolation of a denominator-cleared global root identity.

Nearest codeword and syndrome decoding. A light table therefore yields a satisfying assignment, so an unsatisfiable formula forces weight at least $R_0 = \kappa(q - W_{\deg})$, whereas completeness costs only $R_1 = q(n+1)^2$. The block length is $M = (n+1)^2 q^{r+1}$, because the table has one sample coordinate, r label coordinates, and one matrix block. The actual gap satisfies $R_0/R_1 \geq 3\kappa/[4(n+1)^2]$. With $d = \Theta(n^B)$ and $q = \Theta(rdn^2)$, the block length is $M = n^{B(r+1)+2r+4+o(1)}$, whereas $R_0/R_1 = n^{Br-2+o(1)}$. Thus the gap exponent is $\delta_{r,B} = (Br-2)/(B(r+1)+2r+4)$, which approaches $r/(r+1)$ as $B \to \infty$ for fixed r and then can be made arbitrarily close to 1 by choosing r large. Consequently, for every fixed $0 < \epsilon < 1$, the reduction gives an $M^{1-\epsilon}$ gap for binary nearest codeword and syndrome decoding (Theorem 7.1).

Euclidean GapCVP**.** The parity-kernel lattice lift identifies Hamming weight with squared Euclidean distance exactly, through $\mathrm{dist}_2(t, \mathcal{L}(B))^2 = W(H, b)$. Taking square roots halves the binary gap exponent. Because the binary theorem permits arbitrary slack, reserving part of it for radius rounding yields an $M^{1/2-\epsilon}$ hardness factor for Euclidean GapCVP for every fixed $0 < \epsilon < 1/2$ (Theorem 8.1). That endpoint is the square-root scale of Aharonov and Regev [AR05], which no reduction of this kind can reach unless NP = coNP.

Roadmap. Section 3 defines basic notation, reduces 3SAT to a sparse quadratic system, introduces the assignment encoding and the folded axis polynomials, defines the binary affine system, and proves completeness. Section 4 proves the rank-charging bound for tables whose moment matrix has large rank. Section 5 proves that in the complementary case the moment sequences are sums over entrywise sets of fewer than κ atoms with unit coefficients. Section 6 combines the two cases and decodes an atom into a satisfying assignment. Section 7 chooses the parameters and proves the $n^{1-\epsilon}$ hardness result for binary nearest codeword and binary syndrome decoding. Section 8 lifts the binary instance to a full-rank integer lattice while preserving the optimum as a squared Euclidean distance, which gives the $n^{1/2-\epsilon}$ hardness result for Euclidean GapCVP.

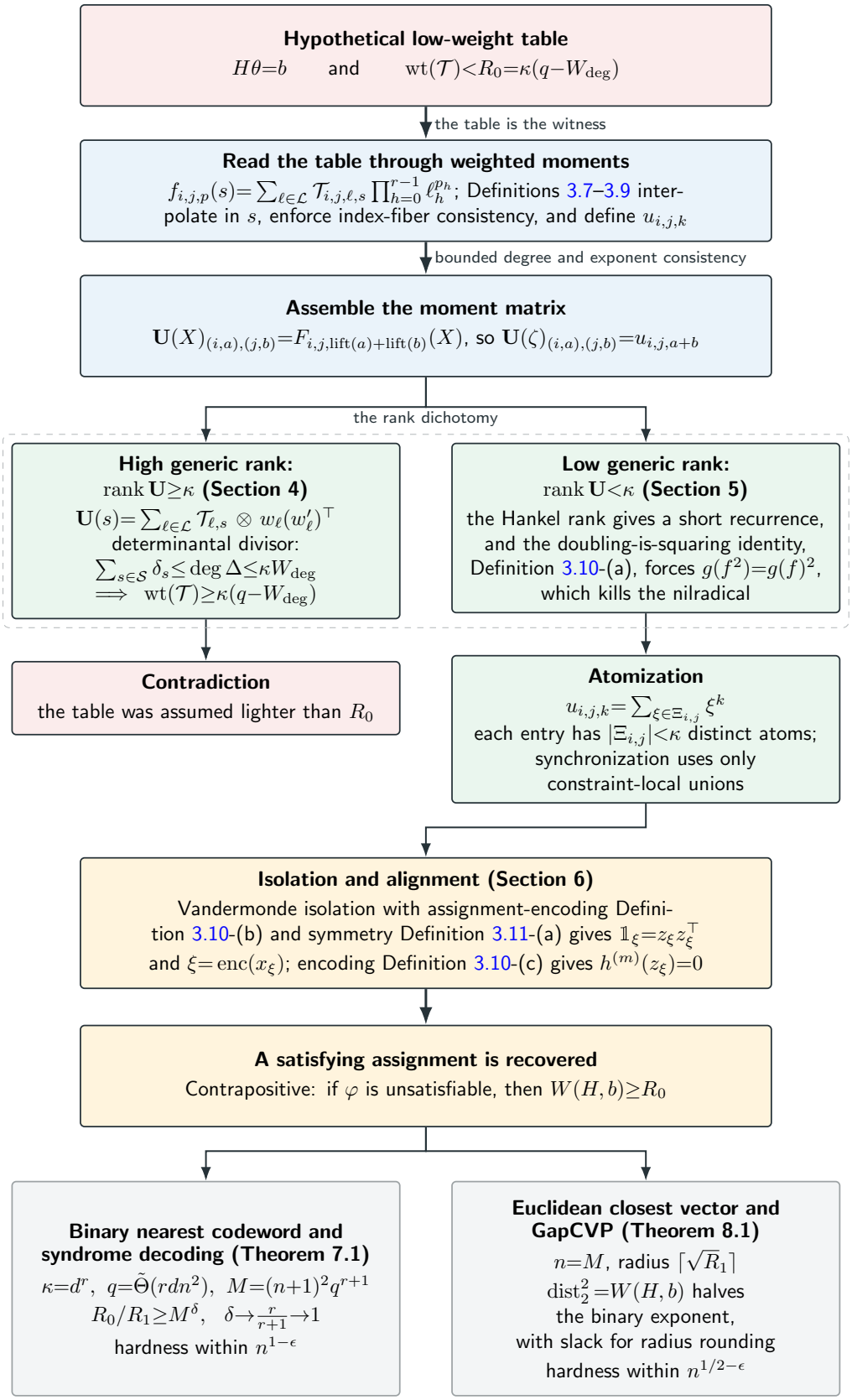


Figure 1: The soundness pipeline.

3 Preliminaries and The Binary Affine System

Section 3.1 defines basic notation and field conventions. Section 3.2 introduces index maps and bounded lifts; Section 3.3 converts 3SAT into a sparse quadratic encoding; Section 3.4 develops axis-wise interpolation and multi-axis exponent reconstruction; and Section 3.5 assembles these ingredients into the binary affine system and establishes its completeness properties.

3.1 Notation

For a positive integer n, we define $[n] := \{1, \dots, n\}$. We write $\mathbb{Z}$ for the integers and $\mathbb{R}$ for the real numbers. For a matrix A, $A^\top$ denotes its transpose and $\operatorname{rank}(A)$ its rank. If A is square, $\det(A)$ denotes its determinant. For matrices A and B, $A \otimes B$ denotes their Kronecker product, so that $(A \otimes B)_{(i,a),(j,b)} = A_{i,j} B_{a,b}$.

For a vector $x = (x_1, \dots, x_n)$, its Hamming weight is $\operatorname{wt}(x) := |\{i \in [n] : x_i \neq 0\}|$, and the same notation is used for the number of nonzero entries of a matrix. For binary vectors $x, y \in \mathbb{F}_2^n$, their Hamming distance is $\operatorname{dist}_H(x, y) := \operatorname{wt}(x - y)$. For a nonempty set $C \subseteq \mathbb{F}_2^n$, we define $\operatorname{dist}_H(x, C) := \min_{y \in C} \operatorname{dist}_H(x, y)$. For $x \in \mathbb{R}^n$, its Euclidean norm is $\|x\|_2 := \sqrt{\sum_{i=1}^n x_i^2}$, and for $t \in \mathbb{R}^n$ and a nonempty $V \subseteq \mathbb{R}^n$ we define $\operatorname{dist}_2(t, V) := \inf_{v \in V} \|t - v\|_2$. In every application below, V is a lattice or a finite set, so the infimum is attained.

Throughout, q is a power of two and $\mathbb{K} := \mathbb{F}_q$. We write $\mathbb{K}[X]$ and $\mathbb{K}(X)$ for the polynomial ring and the rational-function field over $\mathbb{K}$. For a nonzero $\psi \in \mathbb{K}[X]$, $\deg \psi$ denotes its degree, and we set $\deg 0 := -\infty$. defines basic notation and field conventionsFor $\psi \in \mathbb{K}[X]$, $\operatorname{ord}_{X-s}(\psi)$ denotes the order of vanishing of ψ at $X = s$.

3.2 Index Maps and Bounded Lifts

To formulate the affine system, we need to translate bounded multi-indices into scalar exponents and back. The following index and lift maps provide this translation while respecting the prescribed coordinate range.

Definition 3.1 (Index map and bounded lift)**.** *Fix a power of two $d \geq 2$, an integer $r \geq 1$, and an integer $J \geq 1$. For a multi-index $p \in \mathbb{Z}_{\geq 0}^r$, its exponent index is*

$$\operatorname{ind}(p) := \sum_{h=0}^{r-1} p_h d^h.$$

For every integer $0 \leq a \leq J(d^r - 1)$, the bounded lift

$$\operatorname{lift}_J(a) \in \{0, \dots, J(d-1)\}^r$$

is the multi-index obtained by greedily expanding a from the highest base-d position to the lowest. It is chosen so that

$$\operatorname{ind}(\operatorname{lift}_J(a)) = a.$$

Thus ind *converts a multi-index into its scalar exponent, while* lift_J *selects a bounded multi-index representing a given scalar exponent.*

The bounded lift exists throughout the stated range. Indeed, the largest attainable index is $J(d-1)\sum_{h=0}^{r-1} d^h = J(d^r - 1)$. Set $a_{r-1} := a$ and process $h = r-1, r-2, \dots, 0$, maintaining $0 \leq a_h \leq J(d^{h+1} - 1)$. At position h, choose

$$p_h := \min\{J(d-1), \lfloor a_h / d^h \rfloor\}, \qquad a_{h-1} := a_h - p_h d^h.$$

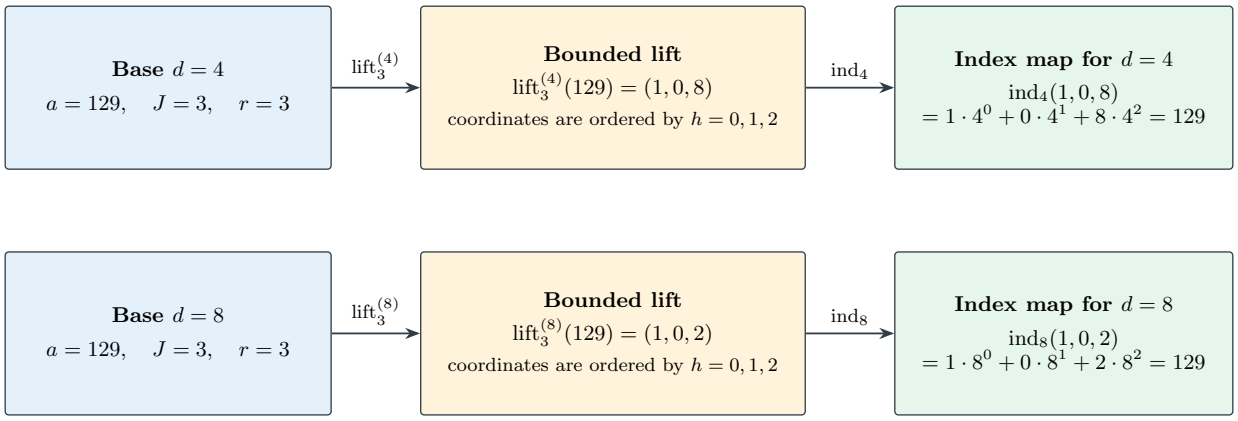


Figure 2: A comparison of the bounded lift under two bases. With the same scalar $a = 129$, lift bound $J = 3$, and length $r = 3$, base $d = 4$ gives $\mathrm{lift}_3^{(4)}(129) = (1, 0, 8)$ and $\mathrm{ind}_4(1, 0, 8) = 129$, whereas base $d = 8$ gives $\mathrm{lift}_3^{(8)}(129) = (1, 0, 2)$ and $\mathrm{ind}_8(1, 0, 2) = 129$. The superscripts and subscripts record the base only for this comparison; in the construction, d is fixed and therefore suppressed from the notation.

If the cap is inactive, then $a_{h-1} \le d^h - 1 \le J(d^h - 1)$. If it is active, then $a_{h-1} \le J(d^{h+1} - 1) - J(d-1)d^h = J(d^h - 1)$. Thus the invariant propagates, and at $h = 0$ it gives $a_{-1} = 0$. When $J = 1$, this is the ordinary base-d expansion. Moreover, linearity of the index map gives

$$\mathrm{ind}(\mathrm{lift}_J(a) + \mathrm{lift}_J(b)) = a + b$$

whenever both bounded lifts are defined.

3.3 Sparse Quadratic Encoding

Let φ be a 3SAT formula of encoding length N. Delete tautological clauses, repeated literals, and unused variables. If deleting repeated literals leaves a clause with fewer than three literals, repeat one of the survivors until it has exactly three; this changes neither the truth value of the clause nor the analysis below, in which two of the three falsity bits then coincide. The cases of an empty clause and of no remaining clauses are handled by fixed promise instances.

Give every occurrence of a variable its own bit, and tie the occurrences of one global variable to its lexicographically first occurrence by equalities. This keeps the number of equalities linear in the number of occurrences, rather than quadratic. For a clause with falsity bits $\beta_0, \beta_1, \beta_2$, introduce one auxiliary bit y and impose $y + \beta_0\beta_1 = 0$ and $y\beta_2 = 0$. Adjoin one homogenizing coordinate and let $z = (z_0, z_1, \ldots, z_n)$ collect it together with all occurrence and auxiliary bits, where z_0 is the homogenizing coordinate. For each falsity bit β_t, let $\widetilde{\beta}_t$ be z_a or $z_a + z_0$, according as the corresponding literal is negated or unnegated. The clause equations become $yz_0 + \widetilde{\beta}_0\widetilde{\beta}_1 = 0$ and $y\widetilde{\beta}_2 = 0$, while an occurrence equality becomes $z_0(z_a + z_b) = 0$. Thus every constraint above becomes a homogeneous quadratic form $h(z) := \sum_{i=0}^{n}\sum_{j=0}^{n} h_{i,j}z_iz_j$ with at most five nonzero coefficients, since each falsity bit is either a coordinate or the sum of a coordinate and z_0. Let $h^{(1)}, \ldots, h^{(L)}$ be the encoding forms. Both n and L are $O(N)$.

Lemma 3.2 (Sparse quadratic encoding)**.** *The formula φ is satisfiable if and only if there is $x \in \mathbb{F}_2^n$ such that $z := (1, x)$ satisfies $h^{(m)}(z) = 0$ for every $m \in [L]$.*

Proof. Set $z_0 = 1$. The equalities force all occurrences of one global variable to agree, so the occurrence bits define a Boolean assignment, and conversely. Fix a clause and its falsity bits. If $\beta_0\beta_1 = 0$, then the first equation forces $y = 0$ and the second holds for every β_2. If $\beta_0 = \beta_1 = 1$, then the first equation forces $y = 1$ and the second forces $\beta_2 = 0$, so the third literal is true. If $\beta_0 = \beta_1 = \beta_2 = 1$, the two equations are inconsistent. Hence the pair is solvable in y exactly when $\beta_0\beta_1\beta_2 = 0$, which is exactly the condition that the clause is satisfied. □

3.4 Axis-wise Interpolation and Multi-axis Exponent Reconstruction

Let $\mathbb{E}$ be an extension of $\mathbb{K}$ of degree $c := [\mathbb{E} : \mathbb{K}]$, generated by ζ, and chosen so that $c \log_2 q \geq n$. Fix $y_1, \ldots, y_n \in \mathbb{E}$ that are linearly independent over $\mathbb{F}_2$; such elements exist because $[\mathbb{E} : \mathbb{F}_2] = c \log_2 q \geq n$. For $x \in \mathbb{F}_2^n$, we define the assignment encoding

$$\mathrm{enc}(x) := \sum_{i=1}^{n} y_i x_i \in \mathbb{E}.$$

Since the y_i are independent over $\mathbb{F}_2$, the map $x \mapsto \mathrm{enc}(x)$ is injective.

We define the moment threshold $\kappa := d^r$. The point of the reconstruction is that κ consecutive powers of $\mathrm{enc}(x)$ are carried by r axis polynomials whose degrees grow only with d.

Lemma 3.3 (Axis-wise interpolation)**.** *For every $x \in \mathbb{F}_2^n$ and every $0 \leq j < r$ there is a unique $P_{x,j} \in \mathbb{K}[X]$ with $\deg P_{x,j} < c$ and $P_{x,j}(\zeta) = \mathrm{enc}(x)^{d^j}$. Its coefficients are $\mathbb{F}_2$-linear functions of x.*

Proof. The set $\{1, \zeta, \ldots, \zeta^{c-1}\}$ is a $\mathbb{K}$-basis of $\mathbb{E}$, so every element of $\mathbb{E}$ is $\psi(\zeta)$ for a unique $\psi \in \mathbb{K}[X]$ of degree less than c, which gives existence and uniqueness. Since d^j is a power of two and $x_i^{d^j} = x_i$, the Frobenius map gives

$$\mathrm{enc}(x)^{d^j} = \sum_{i=1}^{n} y_i^{d^j} x_i.$$

Writing $y_i^{d^j} = \sum_{l=0}^{c-1} \gamma_{i,j,l} \zeta^l$ with $\gamma_{i,j,l} \in \mathbb{K}$ gives $P_{x,j}(X) = \sum_{l=0}^{c-1} (\sum_{i=1}^{n} \gamma_{i,j,l} x_i) X^l$, whose coefficients are $\mathbb{F}_2$-linear in x. □

Lemma 3.4 (Multi-axis exponent reconstruction)**.** *For every $x \in \mathbb{F}_2^n$ and every $p \in \mathbb{Z}_{\geq 0}^r$,*

$$\prod_{h=0}^{r-1} P_{x,h}(\zeta)^{p_h} = \mathrm{enc}(x)^{\mathrm{ind}(p)}.$$

Proof. By Lemma 3.3, the h-th factor contributes exponent $p_h d^h$ to $\mathrm{enc}(x)$. The total exponent is therefore $\sum_{h=0}^{r-1} p_h d^h = \mathrm{ind}(p)$, proving the claim. □

3.5 The Binary Affine System

We define the sample set $\mathcal{S} := \mathbb{K}$ and the label set $\mathcal{L} := \mathbb{K}^r$. The synchronization step of Section 6 isolates atoms against a union of at most $n + 1$ atom sets, so the moment window has to be wider than $(n + 1)\kappa$. We collect the resulting budgets in one definition.

Definition 3.5 (Window and degree parameters)**.** *We use four named budgets because different stages of the proof require different ranges.*

First, define the synchronization factor by

$$W_{\mathrm{syn}} := n + 2.$$

It provides enough slack to combine the atom sets arising from the at most $n+1$ matrix entries in a linear-combination equation.

Second, define the isolation-window length by

$$W_{\mathrm{iso}} := W_{\mathrm{syn}}(\kappa - 1) + 1.$$

This is the number of Hankel rows. The soundness proof uses the consecutive moments with indices $0 \le k < W_{\mathrm{iso}}$ for recurrence propagation and Vandermonde isolation.

Third, define the largest exposed moment index by

$$W_{\mathrm{ind}} := 2W_{\mathrm{syn}}(\kappa - 1).$$

The affine system exposes moments with indices $0 \le k \le W_{\mathrm{ind}}$. Thus W_{ind} is an endpoint, rather than the length of an isolation window.

Finally, define the degree budget by

$$W_{\deg} := 2rW_{\mathrm{syn}}(d-1)(c-1).$$

It bounds the degrees of the interpolants obtained from products of the r axis polynomials: each admissible exponent is at most $2W_{\mathrm{syn}}(d-1)$ and each axis polynomial has degree less than c.

We require $q > 2W_{\deg}$ and $\kappa \ge n+1$. The latter condition gives $(n+1)\kappa \le W_{\mathrm{iso}}$.

These four budgets determine both the allowed digit range and the moment indices exposed by the affine system. We now collect the multi-indices whose coordinates respect that range.

Definition 3.6 (Admissible multi-indices)**.** *The admissible multi-index set is*

$$\mathcal{P} := \{p \in \mathbb{Z}^r_{\ge 0} : p_h \le 2W_{\mathrm{syn}}(d-1) \text{ for all } h\}.$$

For every $p \in \mathcal{P}$, we have $\mathrm{ind}(p) \le W_{\mathrm{ind}}$, so every admissible exponent lies in the exposed moment range.

For the row and column indices used below, we abbreviate $\mathrm{lift}(a) := \mathrm{lift}_{W_{\mathrm{syn}}}(a)$ for $0 \le a < W_{\mathrm{iso}}$.

For every $\ell \in \mathcal{L}$ and $s \in \mathcal{S}$ the system has one binary variable

$$\mathcal{T}_{i,j,\ell,s} \in \mathbb{F}_2 \qquad (0 \le i, j \le n),$$

so a coordinate of the system is named by a row index i, a column index j, a label ℓ, and a sample point s. For fixed ℓ and s these variables form the block $\mathcal{T}_{\ell,s} \in \mathbb{F}_2^{(n+1)\times(n+1)}$ whose (i,j) entry is $\mathcal{T}_{i,j,\ell,s}$, and the whole table is denoted $\mathcal{T}$. The number of binary coordinates is therefore

$$M := (n+1)^2 q^{r+1}. \tag{2}$$

Definition 3.7 (Weighted interpolants)**.** *The following quantities are derived linearly from the table $\mathcal{T}$. For $0 \le i, j \le n$, $p \in \mathcal{P}$, and $s \in \mathcal{S}$, the table is read through*

$$f_{i,j,p}(s) := \sum_{\ell \in \mathcal{L}} \mathcal{T}_{i,j,\ell,s} \prod_{h=0}^{r-1} \ell_h^{p_h} \in \mathbb{K}.$$

For every i, j and every $p \in \mathcal{P}$, let $F_{i,j,p} \in \mathbb{K}[X]$ be the unique polynomial of degree less than q satisfying $F_{i,j,p}(s) = f_{i,j,p}(s)$ for every $s \in \mathcal{S}$. Such a polynomial exists and is unique because $\mathcal{S} = \mathbb{K}$ has q distinct points.

These interpolants package the dependence on the sample point s into the polynomial variable X. We next bound their degrees and require that evaluation at ζ depend only on the exponent encoded by the multi-index.

Definition 3.8 (Degree and exponent consistency)**.** *The following two families constrain the interpolants.*

(a) Bounded degree. For every i, j and every $p \in \mathcal{P}$,

$$\deg F_{i,j,p} \le W_{\deg}.$$

(b) Index-fiber consistency. For every $0 \le i, j \le n$ and all $p, p' \in \mathcal{P}$ with $\mathrm{ind}(p) = \mathrm{ind}(p')$,

$$F_{i,j,p}(\zeta) = F_{i,j,p'}(\zeta).$$

When these constraints hold, any two multi-indices with the same ind give the same value at ζ. This allows us to replace the multi-indexed evaluations by a single sequence indexed by the scalar exponent k.

Definition 3.9 (Exponent-indexed moments)**.** *For each $0 \le k \le W_{\mathrm{ind}}$, fix once and for all a representative $p^{(k)} \in \mathcal{P}$ with $\mathrm{ind}(p^{(k)}) = k$; such representatives exist by the greedy construction above with $J = 2W_{\mathrm{syn}}$. For every $0 \le i, j \le n$, we define*

$$u_{i,j,k} := F_{i,j,p^{(k)}}(\zeta).$$

Whenever the index-fiber consistency constraint holds, this equals $F_{i,j,p}(\zeta)$ for every $p \in \mathcal{P}$ with $\mathrm{ind}(p) = k$.

The quantities $u_{i,j,k}$ are the scalar moments used in the soundness argument. We now impose the algebraic identities satisfied by the honest moments, coming from the doubling-is-squaring identity, the linear-combination identity, and the quadratic encoding.

Definition 3.10 (Algebraic moment constraints)**.** *These three families constrain the moments. Let W_{ind} be the index budget defined in Definition 3.5.*

(a) Doubling is squaring. For every $0 \le i, j \le n$ and every k with $2k \le W_{\mathrm{ind}}$,

$$u_{i,j,2k} = u_{i,j,k}^2.$$

(b) Linear combination. For every $0 \le j \le n$ and every k with $k + 1 \le W_{\mathrm{ind}}$,

$$u_{0,j,k+1} = \sum_{i=1}^{n} y_i u_{i,j,k}.$$

(c) Matrix inner product is zero. For every $m \in [L]$ and every $k \le W_{\mathrm{ind}}$,

$$\sum_{i=0}^{n} \sum_{j=0}^{n} h_{i,j}^{(m)} u_{i,j,k} = 0.$$

The preceding conditions constrain the table only through its derived moments. Two final conditions act directly on the table blocks, enforcing symmetry and normalizing the anchor entry.

Definition 3.11 (Table symmetry and normalization)**.** *The following two families constrain the entries of the table $\mathcal{T}$ directly, without passing through the moments.*

(a) Symmetry in the matrix indices. $\mathcal{T}_{i,j,\ell,s} = \mathcal{T}_{j,i,\ell,s}$ *for all* i, j, $\ell \in \mathcal{L}$ *and* $s \in \mathcal{S}$.

(b) Row summation in $\mathcal{L}$ is one. $\sum_{\ell\in\mathcal{L}} \mathcal{T}_{0,0,\ell,s} = 1$ *for every* $s \in \mathcal{S}$.

The binary affine system $H\theta = b$ consists of the two families of Definition 3.8, the three families of Definition 3.10, and the two families of Definition 3.11.

Lemma 3.12 (The system is affine)**.** *Every condition in Definitions 3.8, 3.10, and 3.11 is an affine condition over $\mathbb{F}_2$ on the table bits.*

Proof. Each $f_{i,j,p}(s)$ is a fixed $\mathbb{K}$-linear combination of table bits, hence an $\mathbb{F}_2$-linear function of them; interpolation on $\mathcal{S}$ and evaluation at ζ are $\mathbb{K}$-linear and therefore $\mathbb{F}_2$-linear. Writing

$$F_{i,j,p}(X) = \sum_{a=0}^{q-1} \widehat{F}_{i,j,p,a} X^a,$$

each coefficient $\widehat{F}_{i,j,p,a} \in \mathbb{K}$ is an $\mathbb{F}_2$-linear function of the table bits. Family Definition 3.8-(a) is precisely the vanishing of these coefficients for $W_{\mathrm{deg}} < a < q$. Evaluation also shows that every $u_{i,j,k}$ is an $\mathbb{F}_2$-linear function of the table bits. The remaining families Definition 3.8-(b), Definition 3.10-(b), Definition 3.10-(c), Definition 3.11-(a), and Definition 3.11-(b) are then $\mathbb{F}_2$-linear or affine by inspection. For Definition 3.10-(a), the map $\lambda \mapsto \lambda^2$ is the Frobenius automorphism of $\mathbb{E}$, which is $\mathbb{F}_2$-linear, so $u_{i,j,k}^2$ is an $\mathbb{F}_2$-linear function of the table bits as well. Expanding each $\mathbb{E}$-valued equation in an $\mathbb{F}_2$-basis of $\mathbb{E}$ turns every family into binary affine equations. □

Let $H\theta = b$ denote the resulting binary affine system, where θ collects the table bits.

Lemma 3.13 (Completeness)**.** *If φ is satisfiable, then $H\theta = b$ has a binary solution of weight at most*

$$R_1 := q(n+1)^2. \tag{3}$$

Proof. Let $x \in \mathbb{F}_2^n$ be as in Lemma 3.2 and put $z := (1, x)$. At each sample point s, we define the honest label

$$\ell_x(s) := (P_{x,0}(s), \ldots, P_{x,r-1}(s)) \in \mathcal{L},$$

and we set $\mathcal{T}_{\ell_x(s),s} := zz^\top$ and $\mathcal{T}_{\ell,s} := 0$ for $\ell \neq \ell_x(s)$. Exactly one block is nonzero at each sample point, so the weight is $q\,\mathrm{wt}(z)^2 \le R_1$, which gives Eq. (3).

We verify the seven families. For every $p \in \mathcal{P}$,

$$f_{i,j,p}(s) = z_i z_j \prod_{h=0}^{r-1} P_{x,h}(s)^{p_h},$$

and the polynomial $z_i z_j \prod_{h=0}^{r-1} P_{x,h}^{p_h}$ has degree at most $(\sum_{h=0}^{r-1} p_h)(c-1) \le 2rW_{\mathrm{syn}}(d-1)(c-1) = W_{\mathrm{deg}}$ by Lemma 3.3, so it is the unique interpolant of Definition 3.7, so $F_{i,j,p} = z_i z_j \prod_{h=0}^{r-1} P_{x,h}^{p_h}$; in particular, Definition 3.8-(a) holds. Evaluating at ζ and applying Lemma 3.4 gives

$$u_{i,j,k} = z_i z_j\, \mathrm{enc}(x)^k \qquad (0 \le k \le W_{\mathrm{ind}}), \tag{4}$$

which depends only on $\mathrm{ind}(p) = k$ and hence gives Definition 3.8-(b). For Definition 3.10-(a),

$$u_{i,j,2k} = z_i z_j \,\mathrm{enc}(x)^{2k} = (z_i z_j \,\mathrm{enc}(x)^k)^2 = u_{i,j,k}^2,$$

where the first step follows from Eq. (4), the second step follows from $z_i^2 = z_i$ and $z_j^2 = z_j$ together with the fact that squaring is additive in characteristic two, and the third step follows again from Eq. (4). For Definition 3.10-(b),

$$\sum_{i=1}^{n} y_i u_{i,j,k} = z_j \,\mathrm{enc}(x)^k \sum_{i=1}^{n} y_i x_i = z_j \,\mathrm{enc}(x)^{k+1} = u_{0,j,k+1},$$

where the first step follows from Eq. (4) and $z_i = x_i$ for $i \geq 1$, the second step follows from the definition of $\mathrm{enc}(x)$, and the third step follows from Eq. (4) and $z_0 = 1$. For Definition 3.10-(c), $\sum_{i=0}^{n} \sum_{j=0}^{n} h_{i,j}^{(m)} u_{i,j,k} = \mathrm{enc}(x)^k h^{(m)}(z) = 0$ by Lemma 3.2. Family Definition 3.11-(a) holds because $zz^\top$ is symmetric, and Definition 3.11-(b) holds because $z_0^2 = 1$. □

4 Rank Loss and Determinantal Charging

A low-weight table cannot support a moment matrix of large rank. The reason is that rank is subadditive and is bounded by the number of nonzero bits, while a determinant of a large minor cannot vanish to high order at too many sample points. Charging the amount of rank lost at each sample point, rather than merely whether rank is lost there, is what keeps the estimate linear in the moment threshold.

We define the polynomial block matrix $\mathbf{U}(X)$, with rows indexed by (i, a) and columns by (j, b) for $0 \leq i, j \leq n$, $0 \leq a < W_{\mathrm{iso}}$, and $0 \leq b < \kappa$, by

$$\mathbf{U}(X)_{(i,a),(j,b)} := F_{i,j,\mathrm{lift}(a)+\mathrm{lift}(b)}(X).$$

This is well defined: the entries of $\mathrm{lift}(a) + \mathrm{lift}(b)$ are at most $2W_{\mathrm{syn}}(d-1)$, so $\mathrm{lift}(a) + \mathrm{lift}(b) \in \mathcal{P}$, and $\mathrm{ind}(\mathrm{lift}(a) + \mathrm{lift}(b)) = a + b$. Consequently, whenever the index-fiber consistency family Definition 3.8-(b) holds, $\mathbf{U}(\zeta)_{(i,a),(j,b)} = u_{i,j,a+b}$. The row window is wider than the column window, which is what the synchronization step of Section 6 consumes.

Lemma 4.1 (Specialization)**.** *For every $\ell \in \mathcal{L}$, define $w_\ell \in \mathbb{K}^{W_{\mathrm{iso}}}$ and $w'_\ell \in \mathbb{K}^{\kappa}$ by $(w_\ell)_a := \prod_{h=0}^{r-1} \ell_h^{\mathrm{lift}(a)_h}$ and $(w'_\ell)_b := \prod_{h=0}^{r-1} \ell_h^{\mathrm{lift}(b)_h}$. Then, for every $s \in \mathcal{S}$,*

$$\mathbf{U}(s) = \sum_{\ell \in \mathcal{L}} \mathcal{T}_{\ell,s} \otimes w_\ell (w'_\ell)^\top, \qquad \textit{and hence} \qquad \mathrm{rank}\, \mathbf{U}(s) \leq \sum_{\ell \in \mathcal{L}} \mathrm{wt}(\mathcal{T}_{\ell,s}).$$

Proof. By Definition 3.7, the polynomial $F_{i,j,p}$ agrees with $f_{i,j,p}$ at every $s \in \mathcal{S}$, so

$$\mathbf{U}(s)_{(i,a),(j,b)} = \sum_{\ell \in \mathcal{L}} \mathcal{T}_{i,j,\ell,s} \prod_{h=0}^{r-1} \ell_h^{\mathrm{lift}(a)_h + \mathrm{lift}(b)_h} = \sum_{\ell \in \mathcal{L}} \mathcal{T}_{i,j,\ell,s} (w_\ell)_a (w'_\ell)_b,$$

where the first step follows from the definition of $f_{i,j,p}$ applied to $p = \mathrm{lift}(a) + \mathrm{lift}(b)$, and the second step follows from the definitions of w_ℓ and w'_ℓ. This is the asserted Kronecker decomposition. Taking ranks,

$$\mathrm{rank}\, \mathbf{U}(s) \leq \sum_{\ell \in \mathcal{L}} \mathrm{rank}(\mathcal{T}_{\ell,s} \otimes w_\ell (w'_\ell)^\top) \leq \sum_{\ell \in \mathcal{L}} \mathrm{rank}(\mathcal{T}_{\ell,s}) \leq \sum_{\ell \in \mathcal{L}} \mathrm{wt}(\mathcal{T}_{\ell,s}),$$

where the first step follows from subadditivity of rank, the second step follows because a Kronecker product with a rank-one matrix does not increase rank, and the third step follows because every matrix is the sum of as many rank-one matrices as it has nonzero entries. □

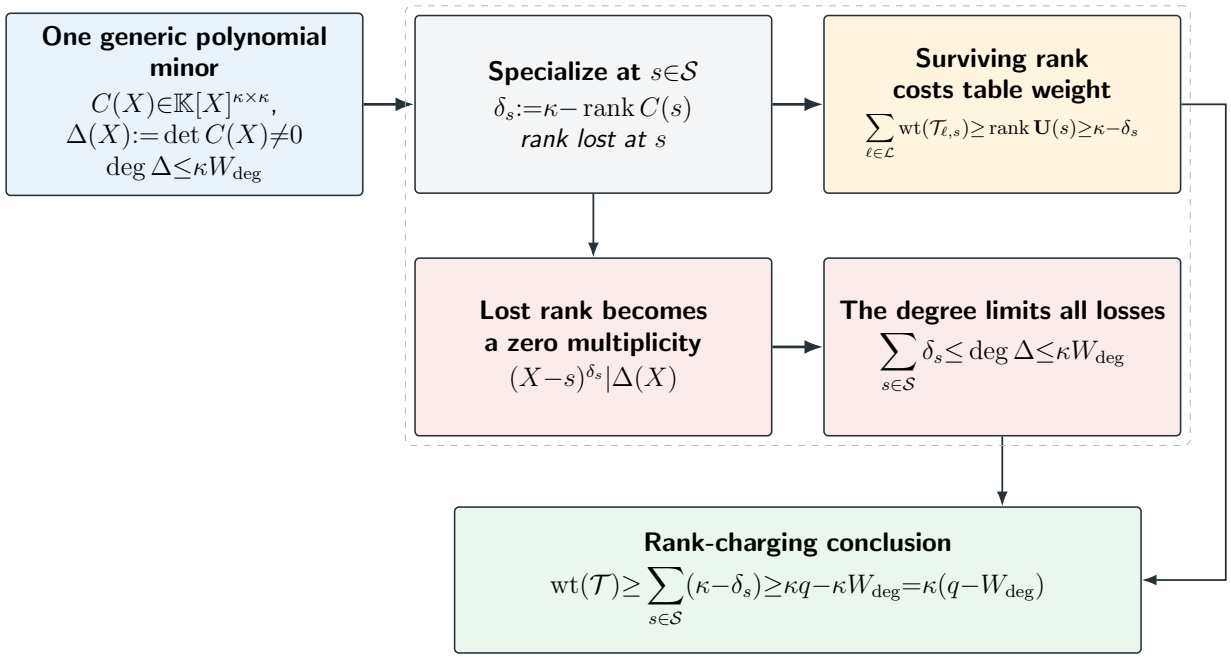


Figure 3: Rank charging. The same deficit δ_s appears both as rank lost in the local weight bound and as the multiplicity of the zero of Δ at s. The degree of Δ therefore controls the total loss over all sample points.

To aggregate rank losses across the sample points, we need a local connection between specialization rank and determinant vanishing. The next lemma provides this connection: a rank deficit δ_s forces the determinant to vanish to order at least δ_s at $X = s$.

Lemma 4.2 (Determinantal divisor)**.** *Let $C \in \mathbb{K}[X]^{\kappa\times\kappa}$ and $s \in \mathbb{K}$, and define $\delta_s := \kappa - \operatorname{rank} C(s)$. If $\det C \neq 0$, then $(X - s)^{\delta_s}$ divides $\det C$.*

Proof. Localize at $X - s$. The ring $\mathbb{K}[X]_{(X-s)}$ is a discrete valuation ring, so C admits a Smith normal form $C = E_1\Lambda E_2$ with E_1, E_2 invertible over that ring and Λ diagonal with entries $(X - s)^{e_1}, \ldots, (X - s)^{e_\kappa}$. Specializing at $X = s$ gives $\operatorname{rank} C(s) = |\{t : e_t = 0\}|$, so at least δ_s of the exponents are positive, and therefore $\operatorname{ord}_{X-s}(\det C) = \sum_{t=1}^{\kappa} e_t \geq \delta_s$. □

The preceding lemma measures the rank lost at a sample point by the zero multiplicity of a nonzero minor. Summing these multiplicities over all sample points charges the total rank deficit to the degree of that minor, yielding the global weight bound below.

Lemma 4.3 (Rank charging)**.** *Suppose the table satisfies Definition 3.8-(a) and $\operatorname{rank}_{\mathbb{K}(X)}\mathbf{U}(X) \geq \kappa$. Define $R_0 := \kappa(q - W_{\deg})$. Then*

$$\operatorname{wt}(\mathcal{T})\geq R_0. \tag{5}$$

Proof. Since the rank is at least κ, there is a $\kappa \times \kappa$ submatrix $C(X)$ of $\mathbf{U}(X)$ with $\Delta := \det C \neq 0$. Every entry of $\mathbf{U}(X)$ has degree at most $W_{\deg}$ by Definition 3.8-(a), so $\deg\Delta \leq \kappa W_{\deg}$. Setting $\delta_s := \kappa - \operatorname{rank} C(s)$, Lemma 4.2 gives $(X - s)^{\delta_s} \mid \Delta$ for every $s \in \mathcal{S}$, and distinct sample points give coprime factors, so

$$\sum_{s\in\mathcal{S}} \delta_s \leq \deg\Delta \leq \kappa W_{\deg}.$$

Therefore

$$\mathrm{wt}(\mathcal{T}) = \sum_{s\in\mathcal{S}}\sum_{\ell\in\mathcal{L}} \mathrm{wt}(\mathcal{T}_{\ell,s}) \geq \sum_{s\in\mathcal{S}} \operatorname{rank}\mathbf{U}(s) \geq \sum_{s\in\mathcal{S}} \operatorname{rank} C(s) = \kappa q - \sum_{s\in\mathcal{S}} \delta_s \geq \kappa(q - W_{\deg}),$$

where the first step is the definition of the total weight, the second step follows from Lemma 4.1, the third step follows because $C(s)$ is a submatrix of $\mathbf{U}(s)$, the fourth step follows from the definition of δ_s and $|\mathcal{S}| = q$, and the fifth step follows from the displayed bound on $\sum_{s\in\mathcal{S}} \delta_s$. □

5 Short Recurrences and Atomic Moments

We now treat the complementary case, in which the moment matrix has rank below κ. A short recurrence gives a finite-dimensional recurrence algebra, which may initially contain generalized nilpotent components and arbitrary coefficient weights. The doubling-is-squaring condition in Definition 3.10-(a) kills the nilradical and forces every surviving coefficient to be one, so each entrywise sequence is the plain sum over a set of fewer than κ atoms. No bound on the global union is needed; synchronization later uses only constraint-local unions.

Lemma 5.1 (Atomization)**.** *Let $\mathbb{E}$ have characteristic two and let $(u_k)_{k\geq 0}$ be a sequence in $\mathbb{E}$ satisfying a monic linear recurrence with characteristic polynomial $\chi \in \mathbb{E}[T]$ of degree $\rho \geq 1$, and suppose $u_{2k} = u_k^2$ for every $k < \rho$. Then there is a set Ξ of at most ρ distinct roots of χ in a splitting field with*

$$u_k = \sum_{\xi\in\Xi} \xi^k \qquad (k \geq 0).$$

Proof. Let $\overline{\mathbb{E}}$ be a splitting field of χ and put $\mathcal{B} := \overline{\mathbb{E}}[T]/(\chi)$. Define $g : \mathcal{B} \to \overline{\mathbb{E}}$ by $g(T^k \bmod \chi) := u_k$. This is well defined, because the map $\overline{\mathbb{E}}[T] \to \overline{\mathbb{E}}$ sending T^k to u_k annihilates every multiple of χ, which is precisely the recurrence at every shift.

We first show $g(\psi^2) = g(\psi)^2$ for every $\psi \in \mathcal{B}$. Writing $\psi = \sum_{i=0}^{\rho-1} a_i T^i$ and using that squaring is additive in characteristic two, $\psi^2 = \sum_{i=0}^{\rho-1} a_i^2 T^{2i}$, so

$$g(\psi^2) = \sum_{i=0}^{\rho-1} a_i^2 u_{2i} = \sum_{i=0}^{\rho-1} a_i^2 u_i^2 = (\sum_{i=0}^{\rho-1} a_i u_i)^2 = g(\psi)^2,$$

where the first step follows from the definition of g, the second step follows from the hypothesis $u_{2i} = u_i^2$, available because $i < \rho$, the third step follows again from additivity of squaring, and the fourth step follows from the definition of g.

Next, g vanishes on the nilradical of $\mathcal{B}$. Indeed, let $\psi \in \mathcal{B}$ be nilpotent. Some power ψ^{2^a} with $a \geq 0$ vanishes, and iterating the previous paragraph gives $g(\psi)^{2^a} = g(\psi^{2^a}) = 0$, so $g(\psi) = 0$ because $\overline{\mathbb{E}}$ is a field.

Consequently g factors through $\mathcal{B}_{\mathrm{red}} := \mathcal{B}/\mathrm{nil}(\mathcal{B})$, where $\mathrm{nil}(\mathcal{B})$ denotes the ideal of nilpotent elements of $\mathcal{B}$. By the Chinese remainder theorem, $\mathcal{B}_{\mathrm{red}}$ is a product of t copies of $\overline{\mathbb{E}}$, one for each of the t distinct roots $\xi_1, \ldots, \xi_t$ of χ, where $t \leq \rho$. Let $\varepsilon_1, \ldots, \varepsilon_t$ be the primitive idempotents and $c_\nu := g(\varepsilon_\nu)$ for $1 \leq \nu \leq t$. From $\varepsilon_\nu^2 = \varepsilon_\nu$ we get $c_\nu^2 = c_\nu$, and in a field this forces $c_\nu \in \{0, 1\}$.

Finally, the image of T in $\mathcal{B}_{\mathrm{red}}$ is $\sum_{\nu=1}^{t} \xi_\nu \varepsilon_\nu$, so T^k maps to $\sum_{\nu=1}^{t} \xi_\nu^k \varepsilon_\nu$ and $u_k = g(T^k) = \sum_{\nu=1}^{t} c_\nu \xi_\nu^k$. Taking $\Xi := \{\xi_\nu : c_\nu = 1\}$ completes the proof, and $|\Xi| \leq t \leq \rho$. □

When the polynomial moment matrix has rank $\rho < \kappa$, its entries satisfy algebraic dependencies of bounded order. The next lemma converts this rank deficiency into a short linear recurrence for each scalar moment sequence, providing the starting point for the atomization argument.

Lemma 5.2 (Short recurrence)**.** *Suppose the table satisfies the index-fiber consistency family Definition 3.8-(b) and* $\rho := \operatorname{rank}_{\mathbb{K}(X)} \mathbf{U}(X) < \kappa$*. Then for every* i, j *there is a monic linear recurrence of order* $\rho_{i,j} \le \rho$ *(where order zero means that the sequence vanishes) satisfied by* $(u_{i,j,k})_k$ *at every index*

$$\rho_{i,j} \le k < \rho_{i,j} + W_{\text{iso}}.$$

Proof. All minors of $\mathbf{U}(X)$ of size $\rho+1$ vanish identically, hence also after substituting $X = \zeta$, so $\operatorname{rank}_{\mathbb{E}} \mathbf{U}(\zeta) \le \rho$. By the folding hypothesis, fixing i and j and letting $a < W_{\text{iso}}$ and $b < \kappa$ range gives the Hankel submatrix $(u_{i,j,a+b})$ of $\mathbf{U}(\zeta)$, whose rank is therefore at most ρ. Hence its first $\rho+1$ columns are linearly dependent. Let $\rho_{i,j} \le \rho$ be minimal such that its columns $c_0, \dots, c_{\rho_{i,j}}$ are linearly dependent; such an index exists because the rank is at most $\rho < \kappa$. By minimality $c_0, \dots, c_{\rho_{i,j}-1}$ are independent, so $c_{\rho_{i,j}}$ is a combination of them, which is a monic recurrence of order $\rho_{i,j}$ holding at every row index $a < W_{\text{iso}}$, that is, at every index $k = a + \rho_{i,j}$ with $a < W_{\text{iso}}$. □

Because the recurrence of Lemma 5.2 is verified at every row index below W_{iso}, the sequence it extends agrees with the true moments on the whole window $0 \le k < W_{\text{iso}}$, and Lemma 5.1 applies with $\rho_{i,j} \le \rho < \kappa \le W_{\text{iso}}$. This gives the following form of the moments. The atom sets are indexed entrywise, and no claim is made that one set serves all entries; the synchronization is carried out constraint by constraint in Section 6, where each constraint couples at most $n+1$ entries.

Lemma 5.2 gives a short recurrence for each moment sequence, but a recurrence alone may still involve arbitrary coefficients and repeated roots. The Frobenius identity removes this ambiguity by forcing the sequence to be a sum of pure exponentials with distinct atoms and unit coefficients, as formalized next.

Lemma 5.3 (Atomic moments)**.** *Suppose the table satisfies Definition 3.8-(b) and Definition 3.10-(a), and suppose* $\rho := \operatorname{rank}_{\mathbb{K}(X)} \mathbf{U}(X) < \kappa$*. Then for every* i, j *there is a set* $\Xi_{i,j}$ *of at most* ρ *distinct elements of a fixed algebraic closure of* $\mathbb{E}$ *with*

$$u_{i,j,k} = \sum_{\xi \in \Xi_{i,j}} \xi^k \qquad (0 \le k < W_{\text{iso}}). \tag{6}$$

Consequently, for any set $\mathcal{F}$ *of at most* $n+1$ *index pairs, the union* $\Xi_{\mathcal{F}} := \bigcup_{(i,j)\in\mathcal{F}} \Xi_{i,j}$ *satisfies* $|\Xi_{\mathcal{F}}| \le (n+1)\rho < (n+1)\kappa \le W_{\text{iso}}$.

Proof. Fix i, j and let $\rho_{i,j}$ be the order supplied by Lemma 5.2. If $\rho_{i,j} = 0$, then the relation produced there reads $u_{i,j,k} = 0$ for every $k < W_{\text{iso}}$, and $\Xi_{i,j} := \emptyset$ satisfies Eq. (6) on the whole window, so we may assume $\rho_{i,j} \ge 1$ below. Its characteristic polynomial is called $\chi_{i,j}$, and the recurrence is valid at every index $\rho_{i,j} \le k < \rho_{i,j} + W_{\text{iso}}$. Let $(\widetilde{u}_k)_{k\ge 0}$ be the sequence determined by the first $\rho_{i,j}$ moments and by that recurrence. By induction on k, $\widetilde{u}_k = u_{i,j,k}$ for every $k < W_{\text{iso}} + \rho_{i,j}$, and in particular on the window $0 \le k < W_{\text{iso}}$: the two agree below $\rho_{i,j}$ by construction, and for $\rho_{i,j} \le k < W_{\text{iso}} + \rho_{i,j}$ the recurrence expresses both in the same way from earlier terms.

Since $\rho_{i,j} \le \kappa - 1$ and $W_{\text{syn}} \ge 1$, $W_{\text{iso}} + \rho_{i,j} - 1 \le (W_{\text{syn}}+1)(\kappa-1) \le W_{\text{ind}}$. Moreover, whenever $0 \le k < \rho_{i,j}$, $2k \le 2\rho_{i,j} - 2 \le 2(\kappa-2) \le W_{\text{ind}}$.

For every $0 \le k < \rho_{i,j}$, both k and $2k$ lie in the range on which $\widetilde{u}$ agrees with $u_{i,j,\cdot}$, and $2k \le W_{\text{ind}}$. Hence the doubling-is-squaring condition gives

$$\widetilde{u}_{2k} = u_{i,j,2k} = u_{i,j,k}^2 = \widetilde{u}_k^2.$$

Thus Lemma 5.1 applies to $\widetilde{u}$ and produces a set $\Xi_{i,j}$ of at most $\rho_{i,j}$ distinct roots of $\chi_{i,j}$ with $\widetilde{u}_k = \sum_{\xi\in\Xi_{i,j}} \xi^k$ for every $k \ge 0$, which is Eq. (6) on the window where $\widetilde{u}$ and $u_{i,j,\cdot}$ agree.

The final bound is immediate: a union of at most $n+1$ sets of size at most ρ has at most $(n+1)\rho$ elements, and $(n+1)\kappa \le W_{\text{iso}}$ holds by the choice of W_{iso} and the requirement $\kappa \ge n+1$. □

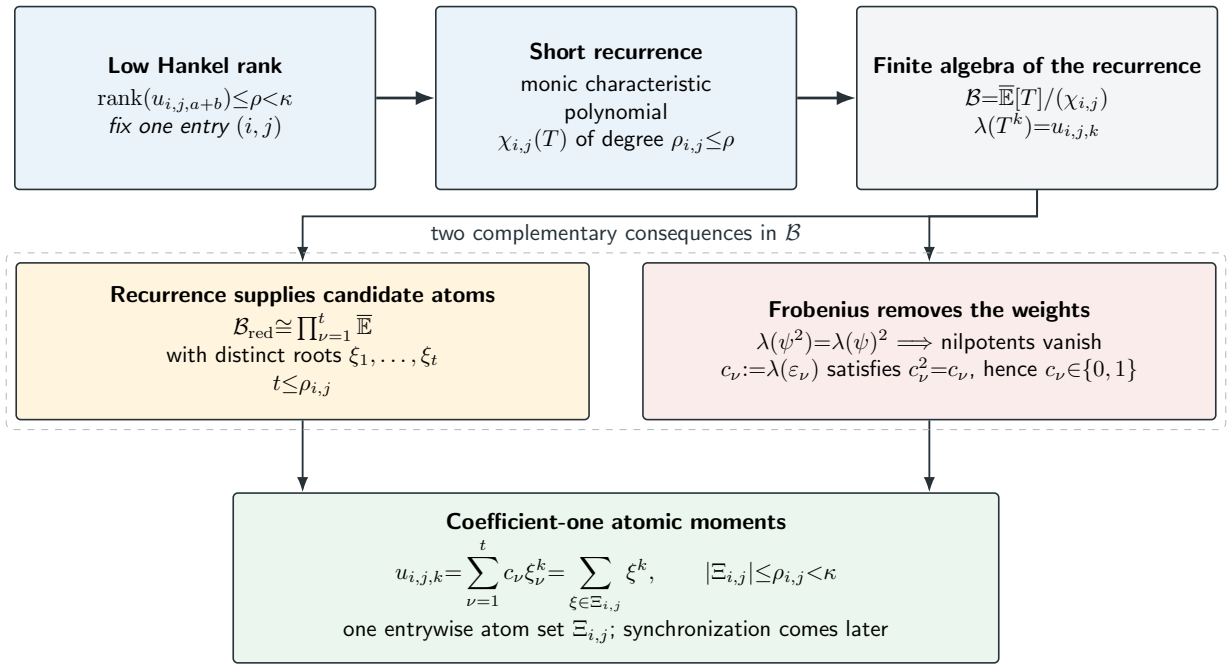


Figure 4: Short-recurrence atomization for a fixed matrix entry. Low Hankel rank gives a short recurrence and candidate atoms. The doubling-is-squaring identity kills nilpotents and forces every surviving coefficient to equal one, producing the entrywise set $\Xi_{i,j}$.

Remark 5.4. It is tempting to shorten the argument by writing the block Hankel matrix as $\mathbf{U}(\zeta) = \sum_{\xi \in \Xi} \mathbb{1}_\xi \otimes v_\xi v_\xi^\top$ with $v_\xi = (\xi^a)_a$ and concluding $\operatorname{rank} \mathbf{U}(\zeta) = \sum_{\xi \in \Xi} \operatorname{rank} \mathbb{1}_\xi$, which would bound the total number of atoms by ρ and make the synchronization immediate. The standard factorization proving that identity uses linear independence of the vectors v_ξ. Without this hypothesis the identity need not hold. Over $\mathbb{F}_4$, take the atoms 0, 1, and p with $p^2 + p + 1 = 0$, two moment coordinates $v_\xi = (1, \xi)^\top$, and $\mathbb{1}_\xi = [1]$ for each of the three atoms. Then $\sum_{\xi \in \{0,1,p\}} \operatorname{rank} \mathbb{1}_\xi = 3$, whereas

$$\sum_{\xi \in \{0,1,p\}} v_\xi v_\xi^\top = \begin{pmatrix} 1 & p^2 \\ p^2 & p \end{pmatrix}$$

has determinant $p + p^4 = 0$ and hence rank one. This is why the argument above keeps the atom sets entrywise and pays for the union with the synchronization factor W_{syn}.

6 Soundness Criterion

We now combine the two cases. If the table is light, rank charging forces the moment matrix to have rank below κ. Atomization then represents every matrix entry through its own set of fewer than κ atoms, and constraint-local Vandermonde isolation synchronizes precisely the entries in each linear-combination, symmetry, encoding, or anchor equation and recovers a satisfying assignment.

Lemma 6.1 (Isolation)**.** *Let $\mathbb{L}$ be a field, let $\Xi \subseteq \mathbb{L}$ be finite with distinct elements, and let $(c_\xi)_{\xi \in \Xi} \subseteq \mathbb{L}$ satisfy $\sum_{\xi \in \Xi} c_\xi \xi^k = 0$ for every $0 \le k < |\Xi|$. Then $c_\xi = 0$ for every ξ.*

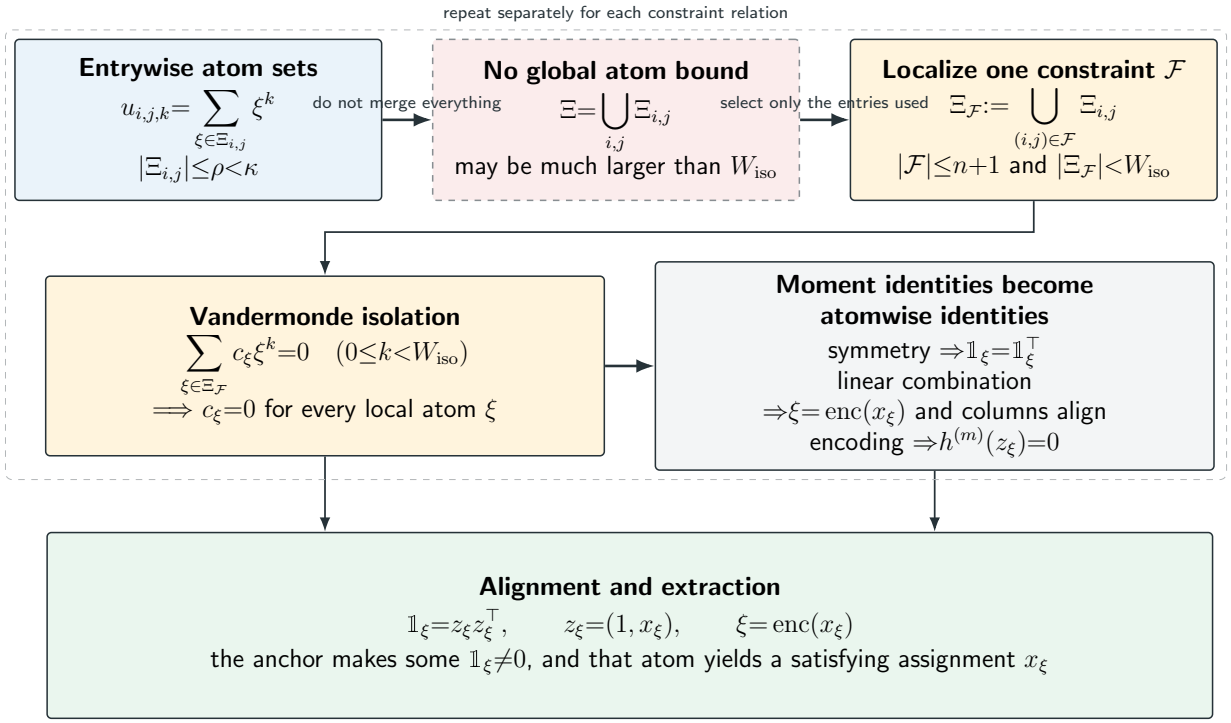


Figure 5: Constraint-local synchronization. No bound is imposed on the global union of the entrywise atom sets. Instead, each constraint uses fewer than W_{iso} atoms, so Vandermonde isolation converts its moment identity into atomwise symmetry, linear-combination, encoding, or anchor conditions.

Proof. The system has coefficient matrix $(\xi^k)_{\xi\in\Xi,0\le k<|\Xi|}$, which is a Vandermonde matrix on the distinct elements of Ξ and hence invertible. □

The preceding lemmas provide two complementary tools: rank charging rules out low-weight solutions in the high-rank case, while atomization and Vandermonde isolation decode a satisfying assignment in the low-rank case. The next lemma combines these two branches. It shows that every solution lighter than R_0 must arise from a satisfiable formula.

Lemma 6.2 (Soundness criterion)**.** *Assume $n \ge 4$. Suppose a binary solution of $H\theta = b$ has weight less than $R_0 = \kappa(q - W_{\text{deg}})$. Then φ is satisfiable.*

Proof. Because the binary vector solves $H\theta = b$, it satisfies Definition 3.8-(a). Lemma 4.3 in contrapositive form therefore gives $\rho := \mathrm{rank}_{\mathbb{K}(X)}\,\mathbf{U}(X) < \kappa$. The solution also satisfies Definition 3.8-(b) and Definition 3.10-(a), so Lemma 5.3 gives entrywise atom sets $\Xi_{i,j}$ of size at most ρ with Eq. (6) valid for $0 \le k < W_{\text{iso}}$. We define

$$\mathbb{1}_\xi(i,j) := 1 \text{ if } \xi \in \Xi_{i,j} \quad \text{and} \quad \mathbb{1}_\xi(i,j) := 0 \text{ otherwise},$$

and we let $\Xi := \bigcup_{0\le i,j\le n}\Xi_{i,j}$, so that $u_{i,j,k} = \sum_{\xi\in\Xi}\mathbb{1}_\xi(i,j)\xi^k$ for $0 \le k < W_{\text{iso}}$. No bound on the total number of atoms is claimed. Each constraint below couples at most $n+1$ entries, so by Lemma 5.3 the union $\Xi_{\mathcal{F}}$ of the atom sets it involves has fewer than W_{iso} elements, and Lemma 6.1 applies to it on the window $0 \le k < W_{\text{iso}}$.

Symmetry. By Definition 3.11-(a), $f_{i,j,p}(s) = f_{j,i,p}(s)$ for every $p \in \mathcal{P}$ and $s \in \mathcal{S}$. Uniqueness of interpolation gives $F_{i,j,p} = F_{j,i,p}$, and Definition 3.9 hence gives $u_{i,j,k} = u_{j,i,k}$ for every $k < W_{\mathrm{iso}}$. The two entries involve at most $2\rho < W_{\mathrm{iso}}$ atoms, so Lemma 6.1 applied to the differences gives $\mathbb{1}_\xi(i,j) = \mathbb{1}_\xi(j,i)$ for every ξ.

Linear combination. By Definition 3.10-(b), for every j and every $0 \le k < W_{\mathrm{iso}} - 1$,

$$\sum_{\xi \in \Xi} (\mathbb{1}_\xi(0,j)\xi + \sum_{i=1}^{n} y_i \mathbb{1}_\xi(i,j))\xi^k = 0,$$

where we used Eq. (6) on both sides of Definition 3.10-(b) and collected the coefficient of ξ^k. This relation involves the $n+1$ entries $(0,j)$ and (i,j) for $1 \le i \le n$, whose atom sets have union of size at most $(n+1)\rho \le (n+1)(\kappa-1) \le W_{\mathrm{iso}}-1$. Thus the $W_{\mathrm{iso}}-1$ moments indexed by $0 \le k < W_{\mathrm{iso}}-1$ suffice, and Lemma 6.1 gives

$$\mathbb{1}_\xi(0,j)\xi = \sum_{i=1}^{n} y_i \mathbb{1}_\xi(i,j) \qquad (\xi \in \Xi,\ 0 \le j \le n). \tag{7}$$

Structure of the coefficient matrices. Fix ξ with $\mathbb{1}_\xi \neq 0$. If $\mathbb{1}_\xi(0,j) = 0$, then Eq. (7) gives $\sum_{i=1}^{n} y_i \mathbb{1}_\xi(i,j) = 0$, and the y_i are independent over $\mathbb{F}_2$, so the whole column j vanishes. Hence there is some j with $\mathbb{1}_\xi(0,j) = 1$, and for every such j Eq. (7) gives $\xi = \sum_{i=1}^{n} y_i \mathbb{1}_\xi(i,j) = \mathrm{enc}(x)$ with $x_i := \mathbb{1}_\xi(i,j)$. Since $x \mapsto \mathrm{enc}(x)$ is injective, this vector is the same for all such j; call it x_ξ and put $z_\xi := (1, x_\xi)$. Every column of $\mathbb{1}_\xi$ is therefore $\mathbb{1}_\xi(0,j) z_\xi$, that is, $\mathbb{1}_\xi = z_\xi a_\xi^\top$ with $(a_\xi)_j := \mathbb{1}_\xi(0,j)$. Symmetry gives $(a_\xi)_j = \mathbb{1}_\xi(0,j) = \mathbb{1}_\xi(j,0) = (z_\xi)_j (a_\xi)_0$. If $(a_\xi)_0 = 0$ then $a_\xi = 0$ and $\mathbb{1}_\xi = 0$, contrary to assumption, so $(a_\xi)_0 = 1$ and

$$\mathbb{1}_\xi = z_\xi z_\xi^\top, \qquad \xi = \mathrm{enc}(x_\xi), \qquad \mathbb{1}_\xi(0,0) = 1. \tag{8}$$

Encoding equations. By Definition 3.10-(c), for every encoding form $h^{(m)}$ and every $0 \le k < W_{\mathrm{iso}}$, using that $h^{(m)}$ has at most five nonzero coefficients and hence involves at most five entries, $\sum_{\xi \in \Xi} (\sum_{i=0}^{n} \sum_{j=0}^{n} h_{i,j}^{(m)} \mathbb{1}_\xi(i,j))\xi^k = 0$ with at most $5\rho < W_{\mathrm{iso}}$ atoms involved, so Lemma 6.1 gives $\sum_{i=0}^{n} \sum_{j=0}^{n} h_{i,j}^{(m)} \mathbb{1}_\xi(i,j) = 0$ for every $\xi \in \Xi$. Substituting Eq. (8) turns this into $h^{(m)}(z_\xi) = 0$.

Nonemptiness. By Definition 3.11-(b), $f_{0,0,0}(s) = 1$ for every s, so the interpolant of Definition 3.7 is the constant polynomial 1 and Definition 3.9 gives $u_{0,0,0} = 1$. By Eq. (6) and Eq. (8), $u_{0,0,0} = \sum_{\xi \in \Xi} \mathbb{1}_\xi(0,0)$ counts modulo two the atoms with $\mathbb{1}_\xi \neq 0$, so that number is odd and in particular nonzero.

Pick any ξ with $\mathbb{1}_\xi \neq 0$. Then $z_\xi = (1, x_\xi)$ satisfies every encoding form, so φ is satisfiable by Lemma 3.2. □

7 Hardness of Binary Nearest Codeword and Syndrome Decoding

It remains to choose the parameters. The radix d controls both the moment threshold $\kappa = d^r$ and the degree budget $W_{\deg}$, but it enters them very differently: κ grows like d^r, whereas $W_{\deg}$ grows only linearly in d. Since the sample field is taken greater than $2W_{\deg}$, the gap κ can be made almost as large as q^r, and the block length pays q^{r+1}. The ratio of these two exponents is what produces $r/(r+1)$.

Theorem 7.1 (Main result, formal version of Theorem 1.8)**.** *Fix an integer $r \ge 1$ and a real $0 < \epsilon < r/(r+1)$. Then* $\mathrm{GapNCP}_{M^{\frac{r}{r+1}-\epsilon}}$ *and* $\mathrm{GapSD}_{M^{\frac{r}{r+1}-\epsilon}}$ *are NP-hard under deterministic*

polynomial-time many-one reductions, where M is the binary block length. Consequently, binary nearest codeword and binary syndrome decoding are NP-hard to approximate within $M^{\frac{r}{r+1}-\epsilon}$. For every fixed $0 < \epsilon < 1$, their corresponding gap problems are NP-hard with factor $M^{1-\epsilon}$, and the optimization problems are NP-hard to approximate within the same factor.

Proof. Because r and ϵ are fixed rather than part of the input, all constants depending on them are hardwired into the reduction. If ϵ is irrational, choose a rational $0 < \epsilon' < \epsilon$; hardness for the larger factor obtained with ϵ' implies the claimed factor by monotonicity.

Let φ have encoding length N, and let n and L be as in Lemma 3.2, so $n, L = O(N)$. Fix an integer $B \geq 3$, depending only on the fixed r and ϵ, as chosen below. All sufficiently-large-n thresholds below are therefore constants. By adjoining unused Boolean variables before fixing the remaining parameters, we may assume that $n \geq 8$ and that n exceeds all these thresholds. This preserves satisfiability and the sparsity of the encoding forms, and still gives $n = O(N)$. Every parameter below is a function of n alone, and the exponent is measured against M, so no lower bound on n in terms of N is needed. We define d to be the least power of two with $d \geq n^B$, and

$$\kappa := d^r, \qquad q := \text{the least power of two with } q \geq 12rdn^2, \qquad c := \lceil n/\log_2 q \rceil.$$

Then $d \geq 2$ and $\kappa \geq n+1$, as Section 3 requires. We let $\mathbb{E}$ be the field with q^c elements, which contains $\mathbb{K} = \mathbb{F}_q$, we let ζ generate $\mathbb{E}$ over $\mathbb{K}$, and we let $y_1, \ldots, y_n$ be the first n members of an $\mathbb{F}_2$-basis of $\mathbb{E}$, which exist because $[\mathbb{E} : \mathbb{F}_2] = c \log_2 q \geq n$.

The parameter inequalities. We first check $q > 2W_{\text{deg}}$. Indeed

$$W_{\text{deg}} = 2rW_{\text{syn}}(d-1)(c-1) < \frac{4rdn^2}{\log_2 q} \leq \frac{q}{3\log_2 q} \leq \frac{q}{12} < \frac{q}{4},$$

where the first step follows from Definition 3.5, together with $W_{\text{syn}} = n+2 \leq 2n$, $d-1 \leq d$, and $c - 1 < n/\log_2 q$, the second step follows from $q \geq 12rdn^2$, and the third step follows from $\log_2 q \geq 4$. Hence $q - W_{\text{deg}} \geq 3q/4$, and by Eq. (5)

$$R_0 = \kappa(q - W_{\text{deg}}) \geq \frac{3\kappa q}{4}.$$

The gap. By Eq. (2) and Eq. (3),

$$M = (n+1)^2 q^{r+1}, \qquad R_1 = q(n+1)^2, \qquad \frac{R_0}{R_1} \geq \frac{3\kappa}{4(n+1)^2}.$$

Since $d = \Theta(n^B)$ and $q = \Theta(rdn^2)$ up to one factor of two, taking logarithms base n gives

$$\log_n M = (r+1)(B+2) + 2 + o(1), \qquad \log_n \frac{R_0}{R_1} = Br - 2 + o(1),$$

where r and B are constants and the $o(1)$ terms tend to zero as n grows. We define the fixed exponent

$$\delta_{r,B} := \frac{Br-2}{B(r+1)+2r+4}, \tag{9}$$

which is the ratio of the two displayed quantities in the limit. As B tends to infinity $\delta_{r,B}$ tends to $r/(r+1)$, so we may fix B with $\delta_{r,B} > r/(r+1) - \epsilon/2$. Writing $\delta := r/(r+1) - \epsilon$ for the target exponent, the strict inequality $\delta < \delta_{r,B}$ leaves positive slack, so

$$\frac{R_0}{R_1} \;\geq\; 2M^{\delta}$$

once n exceeds a fixed threshold. By the preceding padding, every instance entering the construction satisfies this threshold.

Completeness and soundness. Lemma 3.13 gives a solution of weight at most R_1 whenever φ is satisfiable, and Lemma 6.2 gives, in contrapositive form, that if φ is unsatisfiable then every solution has weight at least R_0. Writing $W(H,b) := \min\{\mathrm{wt}(\theta) : H\theta = b\}$ for a consistent system,

$$\begin{aligned} \varphi \text{ satisfiable} &\implies W(H,b) \le R_1, \\ \varphi \text{ unsatisfiable and } H\theta = b \text{ consistent} &\implies W(H,b) \ge 2M^{\delta}R_1 > M^{\delta}R_1. \end{aligned}$$

Conversion. Gaussian elimination computes a solution ℓ of $H\ell = b$ and a generator matrix of the binary linear code $\mathcal{C} := \ker_{\mathbb{F}_2} H$. The solution set of $H\theta = b$ is the coset $\ell + \mathcal{C}$, so

$$\mathrm{dist}_H(\ell, \mathcal{C}) = \min_{c \in \mathcal{C}} \mathrm{wt}(\ell + c) = \min\{\mathrm{wt}(\theta) : H\theta = b\} = W(H,b),$$

where the first step follows from the definition of Hamming distance together with $\ell - c = \ell + c$ over $\mathbb{F}_2$, the second step follows because the solution set is $\ell + \mathcal{C}$, and the third step follows from the definition of $W(H,b)$. Output $\mathcal{C}$, the target ℓ, and the radius R_1; its block length is M. For syndrome decoding, output H, the syndrome b, and the radius R_1, whose optimum is $W(H,b)$ and whose block length is again M.

Exceptional branches. If preprocessing finds an empty clause, or if $H\theta = b$ is inconsistent, then φ is unsatisfiable, the second case because a satisfiable formula produces a solution by Lemma 3.13. For nearest codeword output the fixed code $\{00\}$, the target 11, and the radius 1, whose distance is 2; for syndrome decoding output $H := I_2$, the syndrome $b := 11$, and the radius 1, whose optimum is also 2. If no clauses remain, then φ is satisfiable; output the code $\{00\}$ with target 00 and radius 1, or the syndrome instance $H := I_2$ with $b := 00$ and radius 1, both of optimum zero.

Running time. For fixed r and B we have $d, q = n^{O(1)}$ and $c = O(n)$, so $M = (n+1)^2 q^{r+1} = n^{O(1)}$, which is $N^{O(1)}$ because $n = O(N)$. Write $q = 2^m$. By Shoup's deterministic construction [Sho90], first over $\mathbb{F}_2$ in degree m and then over the resulting representation of $\mathbb{F}_q$ in degree c, we can construct $\mathbb{F}_q$ and a monic irreducible polynomial $h \in \mathbb{F}_q[T]$ of degree c in time polynomial in $\log q$ and c; the small-characteristic guarantee applies because the characteristic is two. Any polynomial-factorization steps over the fixed prime field $\mathbb{F}_2$ are deterministic and polynomial-time, so this construction introduces no randomized subroutine. Set $\mathbb{E} := \mathbb{F}_q[T]/(h)$ and $\zeta := T \bmod h$. The product of the standard bases of $\mathbb{F}_q$ over $\mathbb{F}_2$ and of $\mathbb{E}$ over $\mathbb{F}_q$ gives the explicit $\mathbb{F}_2$-basis from which $y_1, \dots, y_n$ are selected, and elements of $\mathbb{E}$ are stored as c-tuples over $\mathbb{F}_q$. The interpolation matrix on $\mathcal{S} = \mathbb{K}$ and its inverse are computed over $\mathbb{K}$, so all coefficients $\widehat{F}_{i,j,p,a}$ are explicit linear forms. We impose $\widehat{F}_{i,j,p,a} = 0$ for $a > W_{\deg}$. There are $O((n+1)^2|\mathcal{P}|q)$ coefficient conditions, $O((n+1)^2|\mathcal{P}|^2)$ equal-exponent conditions, $O((n+1)^2 W_{\mathrm{ind}})$ Frobenius conditions, $O(nW_{\mathrm{ind}})$ linear-combination conditions, $O(LW_{\mathrm{ind}})$ encoding conditions, and $O(M)$ direct table conditions. A $\mathbb{K}$-valued condition expands into $\log_2 q$ binary equations, and an $\mathbb{E}$-valued condition expands into $c \log_2 q$ binary equations. Since $L = O(N)$ and $|\mathcal{P}|, q, W_{\mathrm{ind}}, M = N^{O(1)}$, all conditions can be enumerated in polynomial time. Gaussian elimination is polynomial. Hence the whole mapping is deterministic polynomial time.

Conclusion. An M^{δ}-approximation algorithm for either problem would compare its returned cost with $M^{\delta}R_1$ and thereby distinguish the two cases, so no such algorithm exists unless P = NP. Since $\delta = r/(r+1) - \epsilon$, this proves the first claim. For the second, given $0 < \epsilon < 1$ choose $r \ge 2/\epsilon$ and apply the first claim with $\epsilon/2$. Then $\frac{r}{r+1} - \frac{\epsilon}{2} = 1 - \frac{1}{r+1} - \frac{\epsilon}{2} \ge 1 - \epsilon$. Therefore an $M^{1-\epsilon}$-approximation would also be an $M^{\,r/(r+1)-\epsilon/2}$-approximation, contradicting the first claim. □

8 Hardness of Euclidean GapCVP

The preceding section produces a gap in the minimum Hamming weight of a binary affine system. To obtain a Euclidean closest-vector instance, we realize the kernel code as the residue modulo two of a full-rank integer lattice and use a representative of the affine coset as the target. The minimum squared Euclidean distance to this lattice is exactly the minimum Hamming weight in the coset: odd coordinates cost at least one, and a minimum-weight binary representative attains equality. Consequently, a Hamming gap of M^δ becomes a Euclidean-distance gap of $M^{\delta/2}$, which explains the exponent in this section.

Theorem 8.1 (Euclidean closest vector, formal version of Theorem 1.3)**.** *Fix a real $0 < \epsilon < 1/2$. There is a deterministic polynomial-time mapping that assigns to each 3SAT formula φ a nonsingular matrix $B \in \mathbb{Z}^{n\times n}$, a target $t \in \mathbb{Z}^n$, and a radius $r \in \mathbb{Q}_{>0}$ such that*

$$\begin{aligned} \varphi \text{ satisfiable} &\implies \operatorname{dist}_2(t, \mathcal{L}(B)) \le r, \\ \varphi \text{ unsatisfiable} &\implies \operatorname{dist}_2(t, \mathcal{L}(B)) > n^{1/2-\epsilon} r. \end{aligned}$$

Consequently, $\mathrm{GapCVP}^{(2)}_{n^{1/2-\epsilon}}$ *is NP-hard under deterministic polynomial-time many-one reductions.*

Proof. Fix the theorem's ϵ. The squared-distance identity below halves a binary gap exponent. We apply Theorem 7.1 with

$$\delta := 1 - \epsilon,$$

which that theorem supplies for every fixed $0 < \epsilon < 1$. The target exponent $1/2 - \epsilon$ is weaker than $\delta/2 = 1/2 - \epsilon/2$, leaving the slack needed to absorb the rounding of the radius. We use the binary affine system constructed in the proof of Theorem 7.1. Before its conversion to nearest codeword, that proof deterministically produces a binary matrix $H \in \mathbb{F}_2^{\rho\times M}$, a syndrome $b \in \mathbb{F}_2^\rho$, and an integer $R_1 \ge 1$, where ρ is the number of binary constraints. For a consistent system, write

$$W(H, b) := \min\{\mathrm{wt}(x) : x \in \mathbb{F}_2^M,\ Hx = b\}.$$

We use the unused-variable padding in that proof with one additional fixed threshold, depending only on ϵ, so that $M > 4^{1/\epsilon}$. This preserves satisfiability and polynomial output size, and ensures $4M^{-\epsilon} < 1$. Lemma 3.13 supplies the completeness guarantee. For soundness, we apply Lemma 6.2 and reuse the parameter verification from the proof of Theorem 7.1. These completeness and soundness conclusions are

$$\begin{aligned} \varphi \text{ satisfiable} &\implies Hx = b \text{ is consistent and } W(H, b) \le R_1, \\ \varphi \text{ unsatisfiable and } Hx = b \text{ consistent} &\implies W(H, b) \ge M^\delta R_1. \end{aligned}$$

We convert every consistent affine system to a full-rank integer lattice without changing the dimension.

Let $C := \ker_{\mathbb{F}_2} H$ and compute a solution $u \in \mathbb{F}_2^M$ of $Hu = b$. Gaussian elimination computes u and a basis for C. We define $h := \dim C$. Choose h linearly independent coordinate columns of a generator matrix, move them to the first h positions, and use invertible row operations to put the generator into systematic form $[I_h\ A]$ for some $A \in \mathbb{F}_2^{h\times(M-h)}$. The row operations only change the chosen basis of C, while the coordinate permutation preserves Hamming weight. The permuted code is therefore

$$C' := \{(y, A^\top y) : y \in \mathbb{F}_2^h\}.$$

All entries of A are henceforth represented by their lifts in $\{0,1\}$. The block form of B is chosen so that reduction modulo two remembers exactly the code C', while the factor $2I_{M-h}$ makes the lattice full rank and ensures that it contains $2\mathbb{Z}^M$. Consequently, parity records the Hamming support of a coset representative, and the squared Euclidean norm charges at least one for each nonzero binary coordinate.

Define

$$B := \begin{pmatrix} I_h & 0 \\ A^\top & 2I_{M-h} \end{pmatrix} \in \mathbb{Z}^{M\times M}.$$

We use the usual block-matrix conventions when $h = 0$ or $h = M$. The determinant of B is 2^{M-h}, so B is nonsingular.

We next identify its lattice. The first h columns of B, reduced modulo 2, generate C', and the remaining columns reduce to zero. Moreover, $2e_j$ is a column of B for $h < j \le M$. If $1 \le i \le h$ and g_i is the i-th column of B, then

$$2e_i = 2g_i - \sum_{j=1}^{M-h} A_{i,j}(2e_{h+j}).$$

Hence $2\mathbb{Z}^M \subseteq \mathcal{L}(B)$. Every column of B reduces modulo 2 to an element of C', so $\mathcal{L}(B) \subseteq \{z \in \mathbb{Z}^M : z \bmod 2 \in C'\}$. For the reverse containment, let $z \in \mathbb{Z}^M$ satisfy $z \bmod 2 \in C'$. The first h columns reduce to a basis of C', so there is $\lambda_0 \in \mathcal{L}(B)$ with $\lambda_0 \bmod 2 = z \bmod 2$. Then $z - \lambda_0 \in 2\mathbb{Z}^M \subseteq \mathcal{L}(B)$, and hence $z \in \mathcal{L}(B)$. Therefore

$$\mathcal{L}(B) = \{z \in \mathbb{Z}^M : z \bmod 2 \in C'\}.$$

Let $t \in \{0,1\}^M$ be the integer vector obtained by applying the same coordinate permutation to u. The affine coset $t + C'$ is the permuted solution set of $Hx = b$.

For any $\lambda \in \mathcal{L}(B)$, we define $x' := (t - \lambda) \bmod 2$. Then $x' \in t + C'$, so x' is the coordinate permutation of a solution to $Hx = b$. Since the permutation preserves Hamming weight, $\mathrm{wt}(x') \ge W(H,b)$. Every nonzero coordinate of x' comes from an odd integer coordinate of $t - \lambda$, whose absolute value is at least one. Therefore

$$\|t - \lambda\|_2^2 \ge \mathrm{wt}(x') \ge W(H,b),$$

where the first step follows because every coordinate on which x' is nonzero corresponds to an odd integer coordinate of $t - \lambda$, and the second step follows because x' represents a solution of $Hx = b$. Conversely, let x attain $W(H,b)$ and let x' be its coordinate permutation. Write $x' = t + c$ for some $c \in C'$. We define $\lambda := t - x'$. Then $\lambda \bmod 2 = c \in C'$, so the lattice characterization above gives $\lambda \in \mathcal{L}(B)$. Moreover, we can show

$$\|t - \lambda\|_2^2 = \|x'\|_2^2 = \mathrm{wt}(x') = \mathrm{wt}(x) = W(H,b),$$

where the first step follows from $t - \lambda = x'$, the second step follows because x' is binary, the third step follows because the coordinate permutation preserves Hamming weight, and the fourth step follows from the choice of x as a minimum-weight solution. We have proved the exact identity

$$\mathrm{dist}_2(t, \mathcal{L}(B))^2 = W(H,b). \tag{10}$$

The natural completeness radius is $\sqrt{R_1}$, which need not be rational. We round it up to the integer r so that the reduction has a valid rational output. This loses at most a factor two in

distance, or a factor four after squaring. The padding condition $4M^{-\epsilon} < 1$ above ensures that the unused exponent slack absorbs this factor.

Set $r := \lceil \sqrt{R_1} \rceil$. This is a positive integer and can be computed by deterministic integer binary search. We can show

$$\sqrt{R_1} \le r \le \sqrt{R_1} + 1 \le 2\sqrt{R_1}, \tag{11}$$

where the first step follows from $r = \lceil \sqrt{R_1} \rceil$, the second step follows from the defining property of the ceiling function, the third step follows from $R_1 \ge 1$.

If φ is satisfiable, we have

$$\mathrm{dist}_2(t, \mathcal{L}(B)) = \sqrt{W(H, b)} \le \sqrt{R_1} \le r,$$

where the first step follows from Eq. (10), the second step follows from completeness, and the third step follows from the definition of r.

Suppose that φ is unsatisfiable and $Hx = b$ is consistent. If

$$\mathrm{dist}_2(t, \mathcal{L}(B)) \le M^{1/2-\epsilon} r,$$

then we can show

$$W(H, b) \le M^{1-2\epsilon} r^2 \le 4M^{1-2\epsilon} R_1 < M^{\delta} R_1,$$

where the first step follows from the assumed upper bound on the Euclidean distance and Eq. (10), the second step follows from $r^2 \le 4R_1$ (Eq. (11)), and the third step follows from $\delta = 1 - \epsilon$ and the padding guarantee $4M^{-\epsilon} < 1$.

This contradicts soundness. Thus

$$\mathrm{dist}_2(t, \mathcal{L}(B)) > M^{1/2-\epsilon} r.$$

The lattice rank is M, which is denoted by n in the theorem statement, so this is precisely the claimed NO threshold.

It remains to specify the exceptional branches. If preprocessing finds an empty clause, or if $Hx = b$ is inconsistent, output

$$B = (2), \qquad t = (1), \qquad r = \frac{1}{2}.$$

This is a one-dimensional NO instance: its rank is $n = 1$, so $n^{1/2-\epsilon} r = 1/2$, whereas $\mathrm{dist}_2(1, 2\mathbb{Z}) = 1$. A satisfiable formula never reaches the inconsistent branch by completeness. If no clauses remain after preprocessing, output

$$B = (2), \qquad t = (0), \qquad r = \frac{1}{2},$$

which is a YES instance.

For fixed ϵ, Theorem 7.1 constructs H, b, and R_1 in deterministic polynomial time. Gaussian elimination, the coordinate permutation, construction of B, and integer square-root computation are also deterministic polynomial-time operations. The basis has dimension exactly M and entries in $\{0, 1, 2\}$, while t is binary. Equations (3) and (2) give $R_1 \le M$, and hence the bit length of r is $O(\log(r + 1)) = O(\log(R_1 + 1)) = O(\log(M + 1))$. Hence the complete mapping has polynomial output size. A polynomial-time algorithm for $\mathrm{GapCVP}^{(2)}_{n^{1/2-\epsilon}}$ composed with this mapping would decide 3SAT in polynomial time, proving the stated NP-hardness. □

References

[ABSS97] Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.

[AGMZ26] Divesh Aggarwal, Rishav Gupta, Aditya Morolia, and Chuanqi Zhang. Mind the gap? not for SVP hardness under ETH! In *53rd International Colloquium on Automata, Languages, and Programming*, volume 374 of *Leibniz International Proceedings in Informatics*, pages 8:1–8:24. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2026.

[AKKV05] Mikhail Alekhnovich, Subhash A. Khot, Guy Kindler, and Nisheeth K. Vishnoi. Hardness of approximating the closest vector problem with pre-processing. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 216–225, 2005.

[AKS01] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 266–275, 2001.

[APY10] Noga Alon, Rina Panigrahy, and Sergey Yekhanin. Deterministic approximation algorithms for the nearest codeword problem. In *Algebraic Methods in Computational Complexity*, volume 9421 of *Dagstuhl Seminar Proceedings*, pages 1–13, 2010.

[AR05] Dorit Aharonov and Oded Regev. Lattice problems in NP ∩ coNP. *Journal of the ACM*, 52(5):749–765, 2005.

[Aro94] Sanjeev Arora. *Probabilistic Checking of Proofs and Hardness of Approximation Problems*. PhD thesis, University of California, Berkeley, 1994. Revised version available as Princeton technical report CS-TR-476-94.

[Bab86] László Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[BGLR25] Vijay Bhattiprolu, Venkatesan Guruswami, Euiwoong Lee, and Xuandi Ren. Inapproximability of finding sparse vectors in codes, subspaces, and lattices. In *Proceedings of the 66th Annual IEEE Symposium on Foundations of Computer Science*, pages 1295–1303, 2025.

[BGR25] Vijay Bhattiprolu, Venkatesan Guruswami, and Xuandi Ren. PCP-free APX-hardness of nearest codeword and minimum distance. Electronic Colloquium on Computational Complexity, TR25-029, 2025.

[BHI+24] Nir Bitansky, Prahladh Harsha, Yuval Ishai, Ron D. Rothblum, and David J. Wu. Dot-product proofs and their applications. In *Proceedings of the 65th Annual IEEE Symposium on Foundations of Computer Science*, pages 806–825, 2024.

[BK02] Piotr Berman and Marek Karpinski. Approximating minimum unsatisfiability of linear equations. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 514–516. Society for Industrial and Applied Mathematics, 2002.

[BMT78] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[BP23] Huck Bennett and Chris Peikert. Hardness of the (approximate) shortest vector problem: A simple proof via Reed–Solomon codes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 275 of *Leibniz International Proceedings in Informatics*, pages 37:1–37:20, 2023.

[DKRS03] Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica*, 23(2):205–243, 2003.

[DKS98] Irit Dinur, Guy Kindler, and Shmuel Safra. Approximating CVP to within almost-polynomial factors is NP-hard. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 99–111, 1998.

[GG00] Oded Goldreich and Shafi Goldwasser. On the limits of nonapproximability of lattice problems. *Journal of Computer and System Sciences*, 60(3):540–563, 2000.

[HKW26] Jeremy Ahrens Huang, Young Kun Ko, and Chunhao Wang. On the (classical and quantum) fine-grained complexity of approximate CVP and Max-Cut. In *53rd International Colloquium on Automata, Languages, and Programming*, volume 374 of *Leibniz International Proceedings in Informatics*, pages 111:1–111:17. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2026.

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

[KPV14] Subhash A. Khot, Preyas Popat, and Nisheeth K. Vishnoi. Almost polynomial factor hardness for closest vector problem with preprocessing. *SIAM Journal on Computing*, 43(3):1184–1205, 2014.

[MG02] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: A Cryptographic Perspective*, volume 671 of *The Springer International Series in Engineering and Computer Science*. Springer, 2002.

[Mos15] Dana Moshkovitz. The projection games conjecture and the NP-hardness of $\ln n$-approximating set-cover. *Theory of Computing*, 11(7):221–235, 2015.

[Muk22] Partha Mukhopadhyay. The projection games conjecture and the hardness of approximation of Super-SAT and related problems. *Journal of Computer and System Sciences*, 123:186–201, 2022.

[MV13] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. *SIAM Journal on Computing*, 42(3):1364–1391, 2013.

[Ope26] OpenAI. Ten advances in mathematics and theoretical computer science. Technical report, 2026.

[RS60] Irving S. Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2–3):201–224, 1987.

[Sho90] Victor Shoup. New algorithms for finding irreducible polynomials over finite fields. *Mathematics of Computation*, 54(189):435–447, 1990.

[vEB81] Peter van Emde Boas. Another NP-complete partition problem and the complexity of computing short vectors in a lattice. Technical Report MI-UvA-81-04, Mathematisch Instituut, University of Amsterdam, 1981.

Acknowledgments and AI Disclosure

The AI tool used in preparing this paper is Codex 5.6 Sol. The author first provided the model with the OpenAI paper [Ope26], from which it derived a finite-k inapproximability exponent of $\frac{k-1}{10k+5}$ for the binary nearest codeword problem, tending to $1/10$ as $k \to \infty$. With further efforts, the author figured out the bound can be improved to $\frac{k-1}{6k+5}$, whose limit is $1/6$. After another level of deep discussion, the author and the AI together improved the bound to $1/4$. After using a different multi-axis construction and lifting, the author figures out how to obtain the present exponent $1-\epsilon$ for every fixed $0 < \epsilon < 1$. All proofs in this paper have been carefully verified by the author. This paper has been substantially revised by the author in response to comments and feedback from Johan Håstad and Omri Weinstein.