Nearly Optimal Hardness for CVP and NCP with Preprocessing

Giacomo Fenzi
giacomo.fenzi@epfl.ch
EPFL

Guy Weissenberg
guy.weissenberg@epfl.ch
EPFL

October 25, 2026

Abstract

The closest vector problem (CVP) and the nearest codeword problem (NCP) are fundamental computational problems with connections to learning, coding theory, and cryptography. We study hardness of approximation in the *preprocessing model*: an algorithm receives the lattice or code in advance, may perform unrestricted computation to produce polynomial-size advice, and must then solve the problem on any target in polynomial time.

Unless NP $\subseteq$ P/poly, no such algorithm achieves an approximation factor of

- $M^{1-\varepsilon}$ for NCP over any fixed finite field, or
- $M^{1/\min\{p,2\}-\varepsilon}$ for CVP in any ℓ_p norm,

for every fixed $\varepsilon > 0$, where M denotes the code block length or lattice dimension, respectively. We improve the subpolynomial factors of Khot, Popat, and Vishnoi [KPV14] to polynomial ones, matching the exponents that Song [Son26] proved without preprocessing for binary codes and the ℓ_2 norm. We further extend Song's evaluation tables to every fixed prime characteristic and the lattice reduction to ℓ_p with $p > 2$. Our main technical contribution is an extension of the moment method, introduced in an OpenAI report [Ope26] and improved by Song [Son26], to the preprocessing setting. In addition, we give a direct reduction from probabilistically checkable proofs (PCPs) whose query locations and families of possible predicates do not depend on the formula. The resulting factor of $M^{\Omega(1/(\log\log M)^2)}$ is already larger than that of [KPV14], and the reduction may be of independent interest.

Keywords: closest vector, nearest codeword, preprocessing, hardness of approximation

Contents

1 Introduction **3**

2 Technical overview **5**
- 2.1 The moment reduction without preprocessing . . . 6
- 2.2 Moving the formula to the right-hand side . . . 9
- 2.3 A PCP reduction without smoothness . . . 13
- 2.4 From codes to lattices . . . 15

3 Preliminaries **16**
- 3.1 Codes and lattices with preprocessing . . . 16
- 3.2 Conversions to codes and lattices . . . 17
- 3.3 Verifiers . . . 19

4 The moment reduction with a fixed matrix **19**
- 4.1 Definitions and setup . . . 19
- 4.2 The construction and its guarantee . . . 21
- 4.3 Completeness . . . 22
- 4.4 Soundness . . . 24
- 4.5 System size and weight gap . . . 25

5 Codes, lattices, and hardness with preprocessing **26**
- 5.1 Approximation hardness with preprocessing . . . 26
- 5.2 Hardness within $M^{1/s-o(1)}$ . . . 28

6 A reduction from query-oblivious PCPs **29**
- 6.1 Query-oblivious verifiers . . . 29
- 6.2 The PCP reduction . . . 30
- 6.3 Hardness from known PCPs . . . 33

A Explicit moment representations **34**
- A.1 Constructing the moment representation . . . 34
- A.2 Rank and moment reconstruction . . . 36
- A.3 Properties of the construction . . . 39
- A.4 Proof of the bounds on system size . . . 40

B Code and lattice conversions **41**
- B.1 A fixed code . . . 41
- B.2 Lattice gaps . . . 42
- B.3 The lattice reduction . . . 43

C Proof of Corollary 5.5 **44**

D Comparison with the Label Cover reductions **45**

AI Methodology **47**

References **47**

1 Introduction

In the nearest codeword problem (NCP) and the closest vector problem (CVP), the goal is to find a codeword or lattice point closest to a given target:

- In NCP, one is given a linear code $C \subseteq \mathbb{F}^M$ of block length M over a finite field $\mathbb{F}$ and a target word $\boldsymbol{y} \in \mathbb{F}^M$, and must find a codeword minimizing the Hamming distance to $\boldsymbol{y}$.
- In CVP, one is given a basis of a full-rank lattice $\mathcal{L} \subseteq \mathbb{R}^M$ and a target vector $\boldsymbol{t} \in \mathbb{R}^M$, and must find a lattice point minimizing the ℓ_p distance to $\boldsymbol{t}$, for a fixed $p \in [1, \infty]$.

Both problems have close connections to learning and coding theory, and suitable variants underpin code-based and lattice-based cryptography, respectively. In this work, we investigate the hardness of approximating NCP and CVP in the *preprocessing model*.

Preprocessing. An algorithm in this model operates in two phases:

- In the ***preprocessing phase***, the algorithm receives a description of the code C or lattice $\mathcal{L}$ and performs unrestricted computation to produce a string, the ***preprocessing output***.
- In the ***online phase***, the algorithm receives this output and a target, and must solve the problem on this target.

We study the approximation factors achievable when the output length and online running time are polynomial, or more generally quasipolynomial, in the relevant encoding lengths. In coding theory and cryptography, the code or lattice is usually known in advance and only the target changes [FM04; AR05].

Algorithms for NCP and CVP. Both problems are NP-hard to solve exactly [BMT78; EB81], so we seek an approximation within a factor $f \geq 1$. In the *search* version, one must return a codeword or lattice point whose distance to the target is at most f times the optimum. In the *gap* version, one is also given a radius R and must decide whether the distance from the target to the code or lattice is at most R or greater than $f \cdot R$, promised that one of the two holds.

- For NCP, the algorithm that returns the target if it is a codeword and any codeword otherwise achieves $f = M$. Over any fixed finite field, polynomial-time algorithms achieve $f = O(M/\log M)$ [BK02; APY10], and no better factor is known even with preprocessing.

- For CVP without preprocessing, polynomial-time algorithms achieve $f = 2^{O\left(M \frac{\log\log M}{\log M}\right)}$ [LLL82; Bab86; Sch87; MV13]. With preprocessing, the algorithm of Dadush, Regev, and Stephens-Davidowitz [DRSD14] achieves $f = O(M/\sqrt{\log M})$ for search in the ℓ_2 norm. For the gap version, the algorithm of Aharonov and Regev [AR05] achieves $f = O(\sqrt{M/\log M})$ in the ℓ_2 norm, and the algorithm of Peikert [Pei08] achieves $f = O(\sqrt{M})$ in the ℓ_p norm for every fixed $p \in [2, \infty]$.

Hardness of approximation. Hardness is known for polynomial factors without preprocessing, but only for subpolynomial factors with preprocessing:

- Without preprocessing, Arora, Babai, Stern, and Sweedyk [ABSS97] proved NP-hardness for every constant factor, and hardness for larger subpolynomial factors under a quasipolynomial-time assumption. Dinur, Kindler, Raz, and Safra [DKRS03] proved NP-hardness of CVP within $M^{c/\log\log M}$ for a constant $c > 0$. Recently, NP-hardness within polynomial factors was proved in an OpenAI report [Ope26, Chapter 7]: $M^{1/200}$ for binary codes and $M^{1/400}$ for lattices in the ℓ_2 norm. Song [Son26] improved these to $M^{1-\varepsilon}$ and $M^{1/2-\varepsilon}$, respectively, for every fixed $\varepsilon > 0$.

- With preprocessing, the largest known factor is $2^{(\log M)^{1-\varepsilon}}$, due to Khot, Popat, and Vishnoi [KPV14], assuming that NP does not have circuits of size $2^{(\log n)^{O(1/\varepsilon)}}$ on inputs of length n (Table 2).

Our results. We prove hardness of approximation within polynomial factors for NCP and CVP with preprocessing.

Theorem 1.1. *Unless* NP $\subseteq$ P/poly*, no preprocessing algorithm with polynomial output length and a polynomial-time online phase achieves an approximation factor of*
- $M^{1-\varepsilon}$ *for NCP over any fixed finite field, for any fixed* $0 < \varepsilon < 1$*, or*
- $M^{1/\min\{p,2\}-\varepsilon}$ *for CVP in the* ℓ_p *norm, for any fixed* $p \in [1, \infty]$ *and* $0 < \varepsilon < 1/\min\{p, 2\}$.

This holds for both the search and the gap versions. If the preprocessing phase runs in polynomial time, the same holds unless P = NP.

These are the first polynomial hardness factors with preprocessing, and they hold under NP $\not\subseteq$ P/poly, a weaker assumption than that of [KPV14]. For NCP and for the gap version of CVP, the factors nearly match those achieved by the best known preprocessing algorithms (Table 1). The approximation factors match those established by Song [Son26] without preprocessing. We also extend the results to codes over every fixed finite field and to lattices in every ℓ_p norm. For codes over $\mathbb{F}_{2^k}$ and lattices with $p < 2$, we reduce from binary codes. For odd characteristic, we construct evaluation tables over $\mathbb{F}_{\mathsf{p}}$; for $p > 2$, we apply Hadamard matrices block by block, following Hair and Sahai's work on the shortest vector problem (SVP) [HS26b] (Sections 2.1 and 2.4).

Problem	Hardness	Algorithmic upper bound
Codes, any fixed finite field	$M^{1-\varepsilon}$	$O(M/\log M)$ [APY10]
Lattices, gap, $1 \le p < 2$	$M^{1/p-\varepsilon}$	$O(M^{1/p}/\sqrt{\log M})$ [AR05]
Lattices, gap, $p = 2$	$M^{1/2-\varepsilon}$	$O(\sqrt{M/\log M})$ [AR05]
Lattices, gap, $2 < p < \infty$	$M^{1/2-\varepsilon}$	$O(\sqrt{M})$ [Pei08]
Lattices, gap, $p = \infty$	$M^{1/2-\varepsilon}$	$O(\sqrt{M})$ [Pei08]
Lattices, search, $p = 2$	$M^{1/2-\varepsilon}$	$O(M/\sqrt{\log M})$ [DRSD14]

Table 1: Approximation with preprocessing. The hardness factors are those of Theorem 1.1, and the algorithms run in polynomial time.

Since Theorem 1.1 holds for every fixed $\varepsilon > 0$, hardness also holds within a single computable factor of the form $M^{1-o(1)}$ for NCP and $M^{1/\min\{p,2\}-o(1)}$ for CVP in ℓ_p norm with rational $p \ge 1$ or $p = \infty$ (Corollary 5.5). The ***nonuniform exponential time hypothesis*** (nonuniform ETH), a circuit version of the hypothesis of [IP01], is the assertion that 3SAT on v variables has no Boolean circuits of size $2^{o(v)}$. Assuming this hypothesis, for every fixed integer $c \ge 1$ we prove hardness for these problems with the $o(1)$ term equal to $C_c \cdot \sqrt{\log\log M/\log M}$ for a constant $C_c > 0$, even when both the output length and the online running time are $M^{O((\log M)^c)}$ (Theorem 5.4).

Lean verification. We formally verified Theorem 5.1 (except for the running-time claims) and the search test of Lemma 5.2 in Lean [FW26].

Organization. Our main contribution is an extension of the moment method of [Ope26; Son26] to the preprocessing setting. In those reductions, the matrix of the linear system depends on the formula. We make the matrix independent of the formula through what we call *telescoping translations* (Section 2.2). We also improve upon prior hardness results with preprocessing through a

reduction from query-oblivious PCPs (input oblivious in [CMS19]) without a smoothness requirement, though the resulting factors are weaker than our main bounds. The constructions and their analysis are quite technical, so we give a detailed overview of both approaches in Section 2 before turning to the formal proofs. After the preliminaries in Section 3, we present the moment reduction in Section 4 and prove hardness of approximation for codes and lattices in Section 5. The PCP reduction is presented in Section 6. We defer the explicit moment representations and supporting proofs to Appendices A to C.

	Codes	Lattices in ℓ_p	Assumption
[FM04]	$5/3-\varepsilon$	$(5/3)^{1/p}-\varepsilon$	NP $\not\subseteq$ P/poly
[Reg04]	$3-\varepsilon$	$3^{1/p}-\varepsilon$	NP $\not\subseteq$ P/poly
[AKKV11]	any constant	any constant	NP $\not\subseteq$ P/poly
[KPV14]	$2^{(\log M)^{1-\varepsilon}}$	$2^{(\log M)^{1-\varepsilon}}$	NP $\not\subseteq$ SIZE$(2^{(\log n)^{O(1/\varepsilon)}})$
Corollary 6.6 (PCP reduction)	$M^{\Omega(1/(\log\log M)^2)}$	$M^{\Omega(1/(p\cdot(\log\log M)^2))}$	NP $\not\subseteq$ SIZE$(n^{O(\log\log n)})$
Theorem 1.1	$M^{1-\varepsilon}$	$M^{1/\min\{p,2\}-\varepsilon}$	NP $\not\subseteq$ P/poly

Table 2: Hardness factors with preprocessing and nonuniform assumptions. For lattices, we take $1 \le p < \infty$ in the first two rows and fixed rational $p \ge 1$ in the PCP row.

2 Technical overview

Throughout, an instance of 3SAT of length $n_{\text{𝔵}}$ is the binary encoding $\text{𝔵} \in \{0,1\}^{n_{\text{𝔵}}}$ of a 3CNF formula, and $\text{𝔵} \in$ 3SAT means that the formula is satisfiable. The prime p is fixed throughout; $\mathbb{F}_{\mathsf{p}}$ is the field with p elements. Unless specified otherwise, the linear systems and codes below are over $\mathbb{F}_{\mathsf{p}}$.

Reductions as linear systems. For the moment reduction, as in [KPV14], we construct the code and the targets from linear systems $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\text{𝔵}}$, one for each formula 𝔵. The code is $\ker \boldsymbol{G}_0$, the target for 𝔵 is any solution $\boldsymbol{v}^*$,[1] and the distance from the target to the code is the least Hamming weight of a solution. For the PCP reduction, we instead minimize the weight after applying a fixed linear map to the solution (Section 2.3). The matrix $\boldsymbol{G}_0$ and the radius D are computed from $1^{n_{\text{𝔵}}}$ alone, and the right-hand side $\boldsymbol{b}_{\text{𝔵}}$ from 𝔵, in polynomial time (quasipolynomial time for the PCP reduction, Section 2.3). For a *gap* Γ, we require (Definitions 3.4 and 3.10):

$$\begin{aligned} \text{Completeness:}\quad & \text{𝔵} \in \text{3SAT} \implies \boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\text{𝔵}} \text{ has a solution of weight at most } D, \\ \text{Soundness:}\quad & \text{𝔵} \notin \text{3SAT} \implies \text{every solution of } \boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\text{𝔵}} \text{ has weight more than } \Gamma \cdot D. \end{aligned} \tag{1}$$

The block length M is within a constant factor of the number of columns of $\boldsymbol{G}_0$ (Lemma 3.11). Crucially, to prove Theorem 1.1, we need a gap $\Gamma \ge M^{1-\varepsilon}$ in terms of the block length M, which will be much larger than $n_{\text{𝔵}}$.

[1] In the formal construction, we use padding to handle systems with no solution, at the cost of a constant factor in the block length (proof of Lemma 3.11, Section B.1).

2.1 The moment reduction without preprocessing

Using the moment method of [Ope26], Song [Son26] proved hardness within the factors of Theorem 1.1 for binary codes and lattices in ℓ_2 without preprocessing. In that reduction, the linear system is derived from a quadratic system encoding the formula's clauses; the clause equations depend on the formula, whereas with preprocessing only the right-hand side may depend on it. We derive our system instead from a verifier circuit that takes the formula as input. The circuit depends only on the formula's length, and so does every equation except those for the input wires. This is our first step, though it is not yet enough: in this subsection, the formula still appears in the matrix, through the equations that set the input wires to the formula's bits. We then remove the formula from the matrix entirely, leaving it only in the right-hand side (Section 2.2).

Initial verifier. For each instance length $n_{\mathbb{x}}$, we fix a Boolean circuit $\mathbf{V}$ with gates of fan-in at most two and $N = \mathrm{poly}(n_{\mathbb{x}})$ wires such that $\mathbf{V}(\mathbb{x}, \mathbb{w}) = 1$ exactly when the witness $\mathbb{w}$ encodes a satisfying truth assignment of $\mathbb{x}$ (Definition 3.13). An *assignment* to the wires is a vector $\boldsymbol{z} \in \{0,1\}^N$ of their values, ordered so that the $n_{\mathbb{x}}$ formula wires come first, followed by the output wire. The restriction $\mathrm{prefix}(\boldsymbol{z})$ of $\boldsymbol{z}$ to these $n_{\mathbb{x}}+1$ wires is the *prefix* of $\boldsymbol{z}$. An assignment is *valid* if every gate computes the correct value from its incoming wires, and it is an *accepting computation on* $\mathbb{x}$ if it is valid with prefix $(\mathbb{x}, 1)$ (Definition 3.14). Such a computation exists exactly when $\mathbb{x} \in$ 3SAT. With a constant wire $z_0 \coloneqq 1$, validity and the prefix condition can be expressed by homogeneous quadratic equations over $\mathbb{F}_{\mathsf{p}}$ in $(z_0, \boldsymbol{z})$, in which every term is a product of two wire values (Claim 3.15):

$$\begin{aligned} z_0 z_o &= z_l z_{l'} && \text{for an AND gate with incoming wires } z_l, z_{l'} \text{ and outgoing wire } z_o, \\ z_0 z_i &= \mathbb{x}_i \cdot z_0 z_0 && (1 \le i \le n_{\mathbb{x}}), \\ z_0 z_{n_{\mathbb{x}}+1} &= z_0 z_0. \end{aligned} \tag{2}$$

These are the *gate equations* (shown for AND gates), the *input equations*, and the *output equation*, respectively.

The encoding. These equations (2) are linear in the products $z_i z_j$, one for each pair $0 \le j \le i \le N$, since $z_i z_j = z_j z_i$ and $z_i^2 = z_i$. We introduce a *product variable* for each distinct product. The values assigned to these variables may satisfy the resulting linear system even if no single Boolean wire assignment has those product values.

The first step in overcoming this is the following. Fix a finite extension $\mathbb{E}$ of $\mathbb{F}_{\mathsf{p}}$ and elements $\beta_1, \ldots, \beta_N \in \mathbb{E}$ that are linearly independent over $\mathbb{F}_{\mathsf{p}}$. The *encoding* of an assignment (Section 4.2) is the field element

$$\mathrm{enc}(\boldsymbol{z}) \coloneqq \sum_{i=1}^{N} \beta_i \cdot z_i \in \mathbb{E}. \tag{3}$$

The products of one assignment satisfy the *encoding identities*

$$\mathrm{enc}(\boldsymbol{z}) \cdot z_j = \sum_{i=1}^{N} \beta_i \cdot z_i z_j \qquad (0 \le j \le N). \tag{4}$$

For a fixed value of $\mathrm{enc}(\boldsymbol{z})$, these identities are linear in the product variables, with z_j on the left-hand side interpreted as the variable for $z_0 z_j$. Conversely, fix any $\alpha \in \mathbb{E}$ in place of $\mathrm{enc}(\boldsymbol{z})$ and assign values in $\mathbb{F}_{\mathsf{p}}$ to the product variables so that these identities hold. One can show that there

are a Boolean assignment $\boldsymbol{z}$ and a scalar $c \in \mathbb{F}_\mathsf{p}$ such that the value assigned to the variable for $z_i z_j$ is $c\, z_i z_j$ for every pair (i, j) (see the proof of Lemma 4.7).

In the reduction, however, the encoding $\alpha = \mathrm{enc}(\boldsymbol{z})$ (see (3)) is not known in advance. If we treat α as another variable, the encoding identities (4) are no longer linear because of the products αz_j. To *linearize* these equations, we introduce variables over $\mathbb{E}$ intended to equal $z_i z_j \cdot \mathrm{enc}(\boldsymbol{z})^k$, for $k \geq 0$ up to a bound to be chosen later. After multiplying each equation above by $\mathrm{enc}(\boldsymbol{z})^k$, we can write it as a linear equation in these variables, with fixed coefficients. In the encoding identities, multiplication by the unknown encoding thus becomes a shift of the exponent k by one.

Evaluation tables. For this linearization, we use *evaluation tables*, following [Ope26; Son26]. These tables are vectors $\boldsymbol{y}$ over $\mathbb{F}_\mathsf{p}$ of length $D_{\mathrm{tab}} \cdot Q$, split into D_{tab} blocks of size Q. Together with the tables, we are given a fixed matrix $\boldsymbol{T}$ over $\mathbb{F}_\mathsf{p}$, whose equations $\boldsymbol{T} \cdot \boldsymbol{y} = \boldsymbol{0}$ are the *evaluation table constraints*, and fixed $\mathbb{F}_\mathsf{p}$-linear functions called *moment maps*

$$\hat{\mu}_k \colon \mathbb{F}_\mathsf{p}^{D_{\mathrm{tab}} \cdot Q} \to \mathbb{E} \qquad (0 \leq k \leq 2\kappa - 2), \tag{5}$$

where the *moment parameter* $\kappa \geq 2$ is chosen according to the desired gap. We call $\hat{\mu}_k(\boldsymbol{y})$ the kth *moment* of the table $\boldsymbol{y}$.

We require the following two properties (Definition 4.4). *Completeness.* For every $\alpha \in \mathbb{E}$, there is a table $\boldsymbol{q}_\alpha \in \{0,1\}^{D_{\mathrm{tab}} \cdot Q}$ with exactly one 1 in each block, $\boldsymbol{T} \cdot \boldsymbol{q}_\alpha = \boldsymbol{0}$, and

$$\hat{\mu}_k(\boldsymbol{q}_\alpha) = \alpha^k \qquad (0 \leq k \leq 2\kappa - 2). \tag{6}$$

To state *soundness*, we first define the *moments* of a function. For a function f from a finite extension $\mathbb{E}'$ of $\mathbb{E}$ to itself, its kth *moment* is

$$\mu_k(f) \coloneqq \sum_{\alpha \in \mathbb{E}'} f(\alpha) \cdot \alpha^k \qquad (k \geq 0). \tag{7}$$

A *coefficient function* is a function $f \colon \mathbb{E}' \to \mathbb{F}_\mathsf{p}$ with *support* $\operatorname{supp} f \coloneqq \{\alpha : f(\alpha) \neq 0\}$, whose elements are its *support points* (Definitions 4.1 and 4.2). The *indicator* $\mathbf{1}_a$ is 1 at a and 0 elsewhere, so $\mu_k(\mathbf{1}_a) = a^k$, and $\boldsymbol{q}_\alpha$ has the moments of $\mathbf{1}_\alpha$. Going back to the soundness condition, we require that every table $\boldsymbol{y}$ with $\boldsymbol{T} \cdot \boldsymbol{y} = \boldsymbol{0}$ and weight below the *weight threshold* $\kappa \cdot D_{\mathrm{tab}}/2$ *represents* a coefficient function f on a finite extension of $\mathbb{E}$ with bounded support:

$$\hat{\mu}_k(\boldsymbol{y}) = \mu_k(f) \quad (0 \leq k \leq 2\kappa - 2), \qquad |\operatorname{supp} f| \leq \frac{2\,\mathrm{wt}(\boldsymbol{y})}{D_{\mathrm{tab}}}. \tag{8}$$

We could already define the linear system using a separate coefficient function $f \colon \mathbb{E} \to \mathbb{F}_\mathsf{p}$ for each product $z_i z_j$, with one variable $f(\alpha)$ for each $\alpha \in \mathbb{E}$. Each moment would then be a fixed linear expression in these variables, by the definition of function moments (7). For each product, however, we would need $|\mathbb{E}| \geq \mathsf{p}^N$ variables, since $\beta_1, \ldots, \beta_N$ are linearly independent over $\mathbb{F}_\mathsf{p}$. We therefore need much shorter evaluation tables satisfying the completeness and soundness requirements above.

Adapting the construction of Song [Son26] to $\mathbb{F}_\mathsf{p}$ (Appendix A), we construct evaluation tables of length

$$D_{\mathrm{tab}} \cdot Q = (\mathsf{a} \cdot N)^{O(\mathsf{a})} \cdot \kappa^{1+1/\mathsf{a}}, \tag{9}$$

for an integer parameter $\mathsf{a} \geq 1$ (Theorem A.8). By choosing a larger a, we can bring the exponent of κ closer to 1, at the cost of a larger factor $(\mathsf{a} \cdot N)^{O(\mathsf{a})}$.

The system. We now combine this linearization with the evaluation tables. The variables of the linear system are the entries of evaluation tables over $\mathbb{F}_\mathsf{p}$ (Construction 4.5): one table $\boldsymbol{v}_{0,0}$ for the constant product $z_0 z_0 = 1$, and one table $\boldsymbol{v}_{i,j}$ for each product $z_i z_j$ with $0 \le j < i \le N$. Symmetric and diagonal indices are identified as for the products, $\boldsymbol{v}_{j,i} \coloneqq \boldsymbol{v}_{i,j}$ and $\boldsymbol{v}_{i,i} \coloneqq \boldsymbol{v}_{i,0}$, and $\boldsymbol{v}$ is the vector of all tables.

The equations are:

- the evaluation table constraints $\boldsymbol{T} \cdot \boldsymbol{v}_{i,j} = \boldsymbol{0}$ for every table,
- the *normalization* $\hat{\mu}_0(\boldsymbol{v}_{0,0}) = 1$ for the constant product $z_0 z_0 = 1$,
- the *encoding equations* $\hat{\mu}_{k+1}(\boldsymbol{v}_{0,j}) = \sum_{i=1}^{N} \beta_i \cdot \hat{\mu}_k(\boldsymbol{v}_{i,j})$ for $0 \le j \le N$,
- the gate equations $\hat{\mu}_k(\boldsymbol{v}_{o,0}) = \hat{\mu}_k(\boldsymbol{v}_{l,l'})$ for each AND gate as above, with the analogous equations for every other gate,
- the input equations $\hat{\mu}_k(\boldsymbol{v}_{i,0}) = \mathbb{x}_i \cdot \hat{\mu}_k(\boldsymbol{v}_{0,0})$ for $1 \le i \le n_\mathbb{x}$ and the output equation $\hat{\mu}_k(\boldsymbol{v}_{n_\mathbb{x}+1,0}) = \hat{\mu}_k(\boldsymbol{v}_{0,0})$,

the last three for every $0 \le k < \kappa$. Thus, for each j, we impose κ encoding equations over $\mathbb{E}$. In a fixed basis of $\mathbb{E}$ over $\mathbb{F}_\mathsf{p}$, each equation between elements of $\mathbb{E}$ is a set of linear equations over $\mathbb{F}_\mathsf{p}$.

Completeness. For $\mathbb{x} \in$ 3SAT, fix an accepting computation $\boldsymbol{z}$ on $\mathbb{x}$ and define the *honest solution* (Section 4.3) by

$$\boldsymbol{v}_{i,j} \coloneqq z_i z_j \cdot \boldsymbol{q}_{\mathrm{enc}(\boldsymbol{z})}, \qquad \text{so that} \qquad \hat{\mu}_k(\boldsymbol{v}_{i,j}) = z_i z_j \cdot \mathrm{enc}(\boldsymbol{z})^k. \tag{10}$$

It satisfies all the equations. Since there are at most $(N+1)^2$ tables, each with at most one 1 in each of its D_{tab} blocks, the weight of this solution satisfies $\mathrm{wt}(\boldsymbol{v}) \le (N+1)^2 \cdot D_{\mathrm{tab}}$.

Soundness. We now show that if $\mathbb{x} \notin$ 3SAT, every solution $\boldsymbol{v}$ has weight $\mathrm{wt}(\boldsymbol{v}) \ge \kappa \cdot D_{\mathrm{tab}}/2$, by extracting an accepting computation on $\mathbb{x}$ from any solution of smaller weight.

Let $\boldsymbol{v}$ be a solution of weight $W < \kappa \cdot D_{\mathrm{tab}}/2$. By the soundness requirement for evaluation tables (8), each table $\boldsymbol{v}_{i,j}$ represents a coefficient function $\omega_{i,j}$ with at most $2\,\mathrm{wt}(\boldsymbol{v}_{i,j})/D_{\mathrm{tab}}$ support points. We use the same function for indices that refer to the same table and extend all functions by zero to a common finite extension $\mathbb{E}'$ of $\mathbb{E}$. Since W is the sum of the weights of the distinct tables, $\left|\bigcup_{i,j} \operatorname{supp} \omega_{i,j}\right| \le \frac{2W}{D_{\mathrm{tab}}} < \kappa$. The functions thus have fewer than κ support points in total, while each encoding, gate, input, and output equation is imposed for κ moment orders. Since there are more moment orders than support points, we can deduce the corresponding identities between the functions $\omega_{i,j}$ at every $a \in \mathbb{E}'$. For example, after replacing table moments by function moments using (8), the encoding equations for a fixed $0 \le j \le N$ become

$$\sum_{a \in \mathbb{E}'} \left(a \cdot \omega_{0,j}(a) - \sum_{i=1}^{N} \beta_i \cdot \omega_{i,j}(a) \right) a^k = 0 \qquad (0 \le k < \kappa).$$

The expression in parentheses is zero outside the union of the supports, so it has fewer than κ nonzero values, and these κ equations are a homogeneous Vandermonde system in those values, with more equations than unknowns. Hence the values are all zero (Lemma 4.3), and $a \cdot \omega_{0,j}(a) = \sum_{i=1}^{N} \beta_i \cdot \omega_{i,j}(a)$ for every $a \in \mathbb{E}'$. Similarly, from the gate, input, and output equations, $\omega_{o,0}(a) = \omega_{l,l'}(a)$, $\omega_{i,0}(a) = \mathbb{x}_i \cdot \omega_{0,0}(a)$, and $\omega_{n_\mathbb{x}+1,0}(a) = \omega_{0,0}(a)$ for every a.

By normalization, $\sum_a \omega_{0,0}(a) = \hat{\mu}_0(\boldsymbol{v}_{0,0}) = 1$, so there is an $\alpha \in \mathbb{E}'$ with $\omega_{0,0}(\alpha) \neq 0$. Fix such an α and define

$$z_i \coloneqq \frac{\omega_{i,0}(\alpha)}{\omega_{0,0}(\alpha)} \qquad (1 \leq i \leq N).$$

By the encoding identity with $j = 0$ at α, $\alpha = \sum_{i=1}^{N} \beta_i \cdot z_i$. Then $\boldsymbol{z} = (z_1, \ldots, z_N)$ is a valid Boolean assignment (Lemma 4.7). By the input and output identities, it is an accepting computation on $\mathbb{x}$.

Thus, if $\mathbb{x} \notin \mathrm{3SAT}$, every solution has weight at least $\kappa \cdot D_{\mathrm{tab}}/2$. The honest solution has weight at most $(N+1)^2 \cdot D_{\mathrm{tab}}$, so (1) holds with this radius and every gap $\Gamma < \kappa/(2(N+1)^2)$. For a desired gap Γ, it therefore suffices to choose $\kappa > 2\Gamma \cdot (N+1)^2$.

2.2 Moving the formula to the right-hand side

The system so far is the moment reduction of [Ope26; Son26] with a verifier circuit in place of the clause system. Going back to our goal, which is that the matrix determining the system of equations must be computed from $1^{n_{\mathbb{x}}}$ alone, we did make some progress by using that circuit: the system now depends on $\mathbb{x}$ only through the input equations

$$\hat{\mu}_k(\boldsymbol{v}_{i,0}) = \mathbb{x}_i \cdot \hat{\mu}_k(\boldsymbol{v}_{0,0}) \qquad (1 \leq i \leq n_{\mathbb{x}},\ 0 \leq k < \kappa),$$

and the right-hand side, zero except in the normalization, is the same for every formula. In the input equations, however, $\hat{\mu}_k(\boldsymbol{v}_{0,0})$ is a linear expression in the entries of $\boldsymbol{v}_{0,0}$, with fixed coefficients, and we multiply it by $\mathbb{x}_i$. Thus the matrix depends on the formula. We cannot move the term $\mathbb{x}_i \cdot \hat{\mu}_k(\boldsymbol{v}_{0,0})$ to the right-hand side, which must be computed from $\mathbb{x}$ alone: the moment value $\mathrm{enc}(\boldsymbol{z})^k$ in an honest solution depends, in general, on the witness. We need a different condition on $\boldsymbol{v}_{0,0}$, with fixed coefficients and with $\mathbb{x}$ only on the right-hand side, under which we can recover a computation with prefix $(\mathbb{x}, 1)$. An assignment has this prefix exactly when its encoding lies in one coset of a fixed subspace. We impose, by moment equations with fixed coefficients, that $\omega_{0,0}$ sums to 1 over this coset.

Telescoping translations. In the honest solution, $\omega_{0,0} = \mathbf{1}_{\mathrm{enc}(\boldsymbol{z})}$, and the prefix of $\boldsymbol{z}$ is $(\mathbb{x}, 1)$ by the input and output equations. The right-hand side may depend on $\mathbb{x}$ but not on the witness. The encoding of the prefix alone,

$$\gamma_{\mathbb{x}} \coloneqq \mathrm{enc}(\mathbb{x}, 1, 0, \ldots, 0)$$

(Construction 4.5), and its powers $\gamma_{\mathbb{x}}^k$ can be computed from $\mathbb{x}$. The question is therefore how to relate the moments $\mathrm{enc}(\boldsymbol{z})^k$ of $\omega_{0,0}$ to these powers with fixed coefficients.

For a coefficient function f and a direction $u \in \mathbb{E}$, the *translation difference* of f by u is the function

$$\alpha \mapsto f(\alpha - u) - f(\alpha).$$

For each moment order k, we can express the kth moment of this difference as a linear combination of the lower-order moments of f:

$$\sum_{\alpha} (f(\alpha - u) - f(\alpha)) \cdot \alpha^k = \sum_{\ell < k} \binom{k}{\ell} \cdot u^{k-\ell} \cdot \mu_\ell(f). \tag{11}$$

The coefficient of $\mu_\ell(f)$ is $\binom{k}{\ell}u^{k-\ell}$. For fixed k and u, these coefficients do not depend on f, even when f itself depends on the witness. For the change of variables and binomial calculation, see Section 4.3.

Now $\mathbf{1}_{\mathrm{enc}(\boldsymbol{z})} - \mathbf{1}_{\gamma_{\boldsymbol{x}}}$ is the translation difference of $\mathbf{1}_{\gamma_{\boldsymbol{x}}}$ by $u = \mathrm{enc}(\boldsymbol{z}) - \gamma_{\boldsymbol{x}} = \sum_{i>n_{\boldsymbol{x}}+1} z_i \cdot \beta_i$. This direction lies in the fixed subspace $U \coloneqq \mathrm{span}_{\mathbb{F}_\mathsf{p}}\{\beta_i : i > n_{\boldsymbol{x}}+1\}$, but it depends on the witness, so the multipliers $\binom{k}{\ell}u^{k-\ell}$ of the moments $\mu_\ell(f)$ in (11) are not fixed.

The key idea is to *replace this single translation by a walk in fixed directions*. We can then keep the moment coefficients fixed and encode the witness dependence in auxiliary functions. Concretely, we reach $\mathrm{enc}(\boldsymbol{z})$ from $\gamma_{\boldsymbol{x}}$ by steps in the directions β_i, along the walk

$$\alpha_{n_{\boldsymbol{x}}+1} \coloneqq \gamma_{\boldsymbol{x}}, \qquad \alpha_i \coloneqq \alpha_{i-1} + z_i \cdot \beta_i \quad (n_{\boldsymbol{x}}+1 < i \le N),$$

which ends at $\alpha_N = \mathrm{enc}(\boldsymbol{z})$, with one step for each wire $z_i = 1$ beyond the prefix. The element α_i is the encoding of the assignment with prefix $(\boldsymbol{x}, 1)$, wire values $z_{n_{\boldsymbol{x}}+2}, \dots, z_i$, and zeros elsewhere. This intermediate assignment need not be valid. The translation difference of $z_i \cdot \mathbf{1}_{\alpha_{i-1}}$ by β_i is $\mathbf{1}_{\alpha_i} - \mathbf{1}_{\alpha_{i-1}}$. Summing along the walk, we derive the *telescoping identity* (Figure 1):

$$\mathbf{1}_{\mathrm{enc}(\boldsymbol{z})}(\alpha) - \mathbf{1}_{\gamma_{\boldsymbol{x}}}(\alpha) = \sum_{i=n_{\boldsymbol{x}}+2}^{N} \left(z_i \cdot \mathbf{1}_{\alpha_{i-1}}(\alpha - \beta_i) - z_i \cdot \mathbf{1}_{\alpha_{i-1}}(\alpha)\right). \tag{12}$$

This is almost what we need: the left-hand side has the moments $\mathrm{enc}(\boldsymbol{z})^k - \gamma_{\boldsymbol{x}}^k$, and by (11) the moments of the right-hand side are linear combinations of the moments of the functions $z_i \cdot \mathbf{1}_{\alpha_{i-1}}$ with coefficients that depend only on $n_{\boldsymbol{x}}$. These functions still depend on the witness, so their moments must be variables of the system, with one auxiliary evaluation table for each $i > n_{\boldsymbol{x}}+1$.

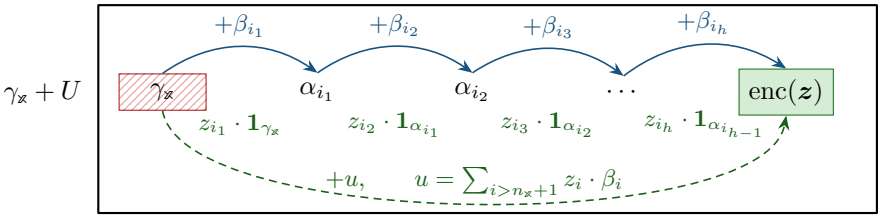

Figure 1: The telescoping identity (12) inside the coset $\gamma_{\boldsymbol{x}} + U$, which contains the encodings of all assignments with prefix $(\boldsymbol{x}, 1)$. The indices $i_1 < \cdots < i_h$ are the coordinates $i > n_{\boldsymbol{x}}+1$ with $z_i = 1$. The dashed arrow is the translation by u, whose direction depends on the witness. We replace this translation by steps in the fixed directions β_i. For a step from a to b, the corresponding summand is $\mathbf{1}_b - \mathbf{1}_a$, the translation difference of the auxiliary function $z_i \cdot \mathbf{1}_a$ shown below the step. These auxiliary functions depend on the witness.

The system. We keep the product tables $\boldsymbol{v}_{i,j}$ and their evaluation table constraints from Section 2.1, together with the encoding and gate equations for $0 \le k < \kappa$. For each $n_{\boldsymbol{x}}+1 < i \le N$, we add the entries of an *auxiliary evaluation table* $\boldsymbol{v}^{(i)} \in \mathbb{F}_\mathsf{p}^{D_{\mathrm{tab}} \cdot Q}$ as variables and impose $\boldsymbol{T} \cdot \boldsymbol{v}^{(i)} = \mathbf{0}$ (Construction 4.5). We replace the input and output equations by the *translation equations*

$$\hat{\mu}_k(\boldsymbol{v}_{0,0}) - \sum_{i=n_{\boldsymbol{x}}+2}^{N} \sum_{\ell<k} \binom{k}{\ell} \cdot \beta_i^{k-\ell} \cdot \hat{\mu}_\ell(\boldsymbol{v}^{(i)}) = \gamma_{\boldsymbol{x}}^k \qquad (0 \le k \le 2\kappa - 2). \tag{13}$$

Each $\hat{\mu}_\ell$ is a fixed $\mathbb{F}_\mathsf{p}$-linear map, and the multipliers $\binom{k}{\ell}\beta_i^{k-\ell}$ depend only on $n_\mathbb{x}$. We expand the equations in a fixed $\mathbb{F}_\mathsf{p}$-basis of $\mathbb{E}$ to form the matrix over $\mathbb{F}_\mathsf{p}$, whose entries depend only on $n_\mathbb{x}$. This is exactly the form we needed: $\mathbb{x}$ appears only through the powers $\gamma_\mathbb{x}^k$ on the right-hand side (Equation 18). For $k = 0$, we have the normalization $\hat{\mu}_0(\boldsymbol{v}_{0,0}) = 1$. The auxiliary tables occur only in their evaluation table constraints and the translation equations (Figure 2).

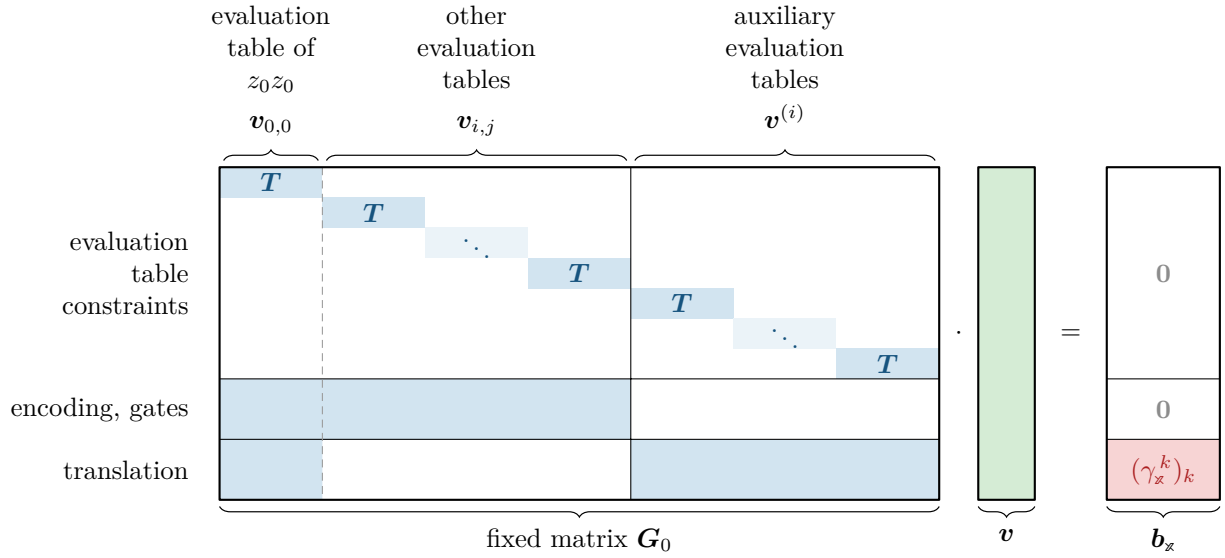

Figure 2: The system $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_\mathbb{x}$. The formula $\mathbb{x}$ appears only on the right-hand side of the translation equations, as the coordinates of the powers $\gamma_\mathbb{x}^k$ in the fixed basis of $\mathbb{E}$ over $\mathbb{F}_\mathsf{p}$.

Completeness. For an accepting computation $\boldsymbol{z}$ on $\mathbb{x}$, the honest solution consists of the tables $\boldsymbol{v}_{i,j} = z_i z_j \cdot \boldsymbol{q}_{\mathrm{enc}(\boldsymbol{z})}$ as before and of the auxiliary tables

$$\boldsymbol{v}^{(i)} \coloneqq z_i \cdot \boldsymbol{q}_{\alpha_{i-1}}, \qquad \text{which represent} \qquad \omega^{(i)} = z_i \cdot \mathbf{1}_{\alpha_{i-1}}$$

(Section 4.3). All tables satisfy the evaluation table constraints, and the tables $\boldsymbol{v}_{i,j}$ satisfy the encoding and gate equations, for the same reasons as before. The table moments are

$$\hat{\mu}_k(\boldsymbol{v}_{0,0}) = \mathrm{enc}(\boldsymbol{z})^k, \qquad \hat{\mu}_k(\boldsymbol{v}^{(i)}) = z_i \cdot \alpha_{i-1}^k.$$

Taking moments of orders $0 \le k \le 2\kappa - 2$ in the telescoping identity (12), and using (11) with $u = \beta_i$, we verify the translation equations (13). With

$$\sigma \coloneqq 1 + \binom{N+1}{2} + N - n_\mathbb{x} - 1 < (N+1)^2$$

tables in all, each again with at most one 1 in each of its D_{tab} blocks, $\mathrm{wt}(\boldsymbol{v}) \le \sigma \cdot D_{\mathrm{tab}} = D$, where D is the radius in (1) (Construction 4.5).

Soundness. For a solution below the weight threshold, every table, including the auxiliary tables, represents a coefficient function. As before, we write $\omega_{i,j}$ and $\omega^{(i)}$ for the functions represented by $\boldsymbol{v}_{i,j}$ and $\boldsymbol{v}^{(i)}$, respectively, extended by zero to a common finite extension $\mathbb{E}'$ of $\mathbb{E}$. The sum of their support sizes over the distinct tables is again less than κ. As in Section 2.1, the encoding and gate identities hold for every α, so $\omega_{0,0}$ is supported on encodings of valid assignments. It remains to find a support point of $\omega_{0,0}$ in the coset $\gamma_{\mathbb{x}} + U$, since the encoding of an assignment lies in this coset exactly when the assignment has prefix $(\mathbb{x}, 1)$. The crucial point is that the auxiliary functions need not describe a walk: the translation difference of *every* function by an element of U sums to zero over each coset of U. By the translation equations and this cancellation, $\omega_{0,0}$ sums to 1 over $\gamma_{\mathbb{x}} + U$.

We next verify the *translation identity*

$$\omega_{0,0}(\alpha) - \mathbf{1}_{\gamma_{\mathbb{x}}}(\alpha) = \sum_{i=n_{\mathbb{x}}+2}^{N} \left(\omega^{(i)}(\alpha - \beta_i) - \omega^{(i)}(\alpha)\right) \qquad (\alpha \in \mathbb{E}') \tag{14}$$

(Equation 19). Its two sides have the same moments of orders up to $2\kappa - 2$ by the translation equations, and their difference has at most

$$|\operatorname{supp} \omega_{0,0}| + 1 + 2 \sum_i |\operatorname{supp} \omega^{(i)}| \le 2\kappa - 1$$

support points, since a translation difference of f has at most $2|\operatorname{supp} f|$ support points, so we apply Lemma 4.3 as before, now with $t = 2\kappa - 1$. This is why the translation equations go up to order $2\kappa - 2$, roughly twice as many moment orders as the encoding and gate equations.

Every coset of U is invariant under translation by $\beta_i \in U$, so for every function f and every $\gamma \in \mathbb{E}'$

$$\sum_{\alpha \in \gamma + U} \left(f(\alpha - \beta_i) - f(\alpha)\right) = 0$$

(Section 4.4). Summing the translation identity over $\gamma_{\mathbb{x}} + U$, we obtain

$$\sum_{\alpha \in \gamma_{\mathbb{x}} + U} \omega_{0,0}(\alpha) = 1.$$

Hence there is an $\alpha \in \gamma_{\mathbb{x}} + U$ with $\omega_{0,0}(\alpha) \neq 0$. Fix such an α. As in Section 2.1, it is the encoding of a valid assignment $\boldsymbol{z} = (z_1, \ldots, z_N)$. Since $\alpha - \gamma_{\mathbb{x}} \in U$ is a $\mathbb{F}_{\mathsf{p}}$-linear combination of the β_i with $i > n_{\mathbb{x}} + 1$, we can compare coefficients of the linearly independent β_i in $\operatorname{enc}(\boldsymbol{z}) = \alpha$: $z_i = \mathbb{x}_i$ for $1 \le i \le n_{\mathbb{x}}$ and $z_{n_{\mathbb{x}}+1} = 1$. Thus $\boldsymbol{z}$ is an accepting computation on $\mathbb{x}$, and if $\mathbb{x} \notin$ 3SAT, every solution has weight at least $\kappa \cdot D_{\mathrm{tab}}/2 = \frac{\kappa}{2\sigma} \cdot D$, so (1) holds for every gap $\Gamma < \kappa/(2\sigma)$ (Theorem 4.6).

Parameters. For any fixed integer $\mathsf{a} \ge 1$, we can construct tables of length $\operatorname{poly}(N) \cdot \kappa^{1+1/\mathsf{a}}$ as before, provided κ is an a-th power of an integer (Construction A.4). With κ chosen as the least such power above $2\sigma \cdot \Gamma$, we have $\kappa = O(\sigma \cdot \Gamma)$ for fixed a. Since there are $\sigma \le (N+1)^2$ tables, the code block length satisfies

$$M \le \operatorname{poly}(N) \cdot \Gamma^{1+1/\mathsf{a}}.$$

The polynomial factor may depend on the fixed choice of a (Theorem 4.8).

The two factors beyond Γ in this bound are $\Gamma^{1/\mathsf{a}}$ and a polynomial in N. For every fixed $0 < \varepsilon < 1$, we can make each at most $\Gamma^{\varepsilon/2}$ by choosing $\mathsf{a} = \lceil 2/\varepsilon \rceil$ in the evaluation-table construction

and Γ as a sufficiently large fixed power of N. We then have $M \le \Gamma^{1+\varepsilon}$, and hence $\Gamma \ge M^{1-\varepsilon}$, while the code length and construction time remain polynomial in N. This is the required gap for Theorem 1.1 for codes over $\mathbb{F}_{\mathsf{p}}$. With the auxiliary tables and the doubled moment orders in the translation equations, only the polynomial factor is larger. The gap exponent is the same as for codes without preprocessing [Son26].

2.3 A PCP reduction without smoothness

We also give a new reduction from a *query-oblivious* PCP [CMS19], with efficiently enumerable, instance-independent families of acceptance predicates. We use the instance-independent queries and predicate families to construct a code that is fixed before the instance is known. We bound the answer-list sizes using a linear encoding, without a smoothness requirement. The improved gap then follows from the small query complexity and low soundness error of the IOP constructions below. With the IOPs of Arnon, Chiesa, and Yogev [ACY23] and Minzer and Zheng [MZ25], we prove hardness within $M^{\Omega(1/(\log\log M)^2)}$, an improvement over the earlier factors but still far from the $M^{1-\varepsilon}$ of the moment method.

Views and class sums. We use a query-oblivious PCP (Definition 6.1) for $\{(\mathbb{x}, \mathbb{w}) : \mathbf{V}(\mathbb{x}, \mathbb{w}) = 1\}$ with perfect completeness, soundness error s, R random bits, and q queries over Σ. For each random string r, write $I_r = (u_{r,1}, \dots, u_{r,\mathsf{q}})$ for the query tuple and $\mathcal{D}_r$ for the family of labeled acceptance predicates $\Sigma^{\mathsf{q}} \to \{0,1\}$, which contains the predicate $V_{\mathbb{x},r}$ used for instance $\mathbb{x}$. An *accepting view* for r is a pair $(V, \boldsymbol{a}) \in \mathcal{D}_r \times \Sigma^{\mathsf{q}}$ with $V(\boldsymbol{a}) = 1$ and $\boldsymbol{a} = \pi|_{I_r}$ for some proof π, that is, $a_j = a_{j'}$ whenever $u_{r,j} = u_{r,j'}$. Write $\mathcal{V}_r$ for these views and $\mathsf{m} \coloneqq \sum_r |\mathcal{V}_r|$. There is one variable $v_{r,V,\boldsymbol{a}}$ over a fixed finite field $\mathbb{F}$ for each accepting view, so a tuple accepted by two predicates has two variables. For every r and $V \in \mathcal{D}_r$, the views $(V, \boldsymbol{a})$ form a *class*, and we impose only the class sums

$$\sum_{\boldsymbol{a}:\,(V,\boldsymbol{a})\in\mathcal{V}_r} v_{r,V,\boldsymbol{a}} = \begin{cases} 1 & \text{if } V = V_{\mathbb{x},r}, \\ 0 & \text{otherwise,} \end{cases} \qquad (r \in \{0,1\}^{\mathsf{R}},\ V \in \mathcal{D}_r), \tag{15}$$

written as $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\mathbb{x}}$, with one row for every pair (r, V). There is one class sum equal to 1 per random string, whereas a single coset sum equals 1 for the whole computation in the translation equations (Section 2.2). In each class with sum 1, at least one coefficient is nonzero, so every random string r has an accepting view of $V_{\mathbb{x},r}$ with a nonzero variable. Whenever the system is feasible, we can choose one accepting view of this predicate for each random string, set its variable to 1, and set all other variables to 0. Then $\mathrm{wt}(\boldsymbol{v}) = 2^{\mathsf{R}}$, whether or not the chosen views agree at shared locations. To measure agreement, we instead consider, for each proof location, how many distinct answers occur among views with nonzero coefficients.

Encoding the answer lists. Let $\mathcal{U}$ be the set of queried locations. For $(u, a) \in \mathcal{U} \times \Sigma$, let $\boldsymbol{v}^{u,a} \in \mathbb{F}^{\mathsf{m}}$ agree with $\boldsymbol{v}$ on the views with answer a at u, from all random strings and classes, and be zero elsewhere. The *answer list* at u is

$$\Lambda_u(\boldsymbol{v}) \coloneqq \{a \in \Sigma : \boldsymbol{v}^{u,a} \neq \boldsymbol{0}\}.$$

To bound the list size, we encode each restriction rather than summing its coordinates. Let $\iota\colon \mathbb{F}^{\mathsf{m}} \to \mathbb{F}^B$ be an injective linear encoding map whose code $\iota(\mathbb{F}^{\mathsf{m}})$ has relative distance δ. Then, for every

$\boldsymbol{z} \in \mathbb{F}^{\mathsf{m}}$,

$$\delta \cdot B \cdot \mathbf{1}[\boldsymbol{z} \neq \mathbf{0}] \leq \mathrm{wt}(\iota(\boldsymbol{z})) \leq B \cdot \mathbf{1}[\boldsymbol{z} \neq \mathbf{0}].$$

Thus the encoding of $\boldsymbol{v}^{u,a}$ has weight between $\delta \cdot B$ and B if $a \in \Lambda_u(\boldsymbol{v})$, whatever the coefficients of the views with answer a, and weight 0 otherwise.

In the Label Cover reduction of [AKKV05, Sections 4–5] and the reduction of [KPV14], projection sums can vanish by cancellation. Smoothness is used to bound projection collisions within short lists of labels with nonzero coefficients (Appendix D). Here every nonzero restriction has a nonzero encoding, so soundness can be proved from the answer lists without a smoothness assumption (Remark 6.3).

The use of a code with constant relative distance to measure nonzero restrictions already occurs in [AKKV05, Lemma 3.5]. Here the restrictions are indexed by $(u, a) \in \mathcal{U} \times \Sigma$ and include the views of all random strings. Let d_u be the number of query occurrences $(r, j) \in \{0,1\}^{\mathsf{R}} \times [\mathsf{q}]$ with $u_{r,j} = u$. To match the query frequencies, we define $\boldsymbol{G}_1 \cdot \boldsymbol{v}$ to contain d_u copies of $\iota(\boldsymbol{v}^{u,a})$ for every $(u, a) \in \mathcal{U} \times \Sigma$. Consequently,

$$\delta \cdot \sum_{u \in \mathcal{U}} d_u \cdot |\Lambda_u(\boldsymbol{v})| \leq \frac{\mathrm{wt}(\boldsymbol{G}_1 \cdot \boldsymbol{v})}{B} \leq \sum_{u \in \mathcal{U}} d_u \cdot |\Lambda_u(\boldsymbol{v})|. \tag{16}$$

The code and the target are

$$C = \boldsymbol{G}_1(\ker \boldsymbol{G}_0), \qquad \boldsymbol{y}_{\mathbb{x}} = \boldsymbol{G}_1 \cdot \boldsymbol{v}^*, \qquad \boldsymbol{G}_0 \cdot \boldsymbol{v}^* = \boldsymbol{b}_{\mathbb{x}},$$

as in [KPV14, Section 2.1], and the distance from the target to the code is the minimum of $\mathrm{wt}(\boldsymbol{G}_1 \cdot \boldsymbol{v})$ over the solutions (Lemma 3.8). Both matrices are fixed, and only $\boldsymbol{b}_{\mathbb{x}}$ and $\boldsymbol{y}_{\mathbb{x}}$ depend on the instance.

Completeness and soundness. In a YES instance, for a proof π accepted with probability 1, we define the *honest solution* by setting $v_{r,V,\boldsymbol{a}} = 1$ exactly when $V = V_{\mathbb{x},r}$ and $\boldsymbol{a} = \pi|_{I_r}$, with all other variables zero. It satisfies (15), and at each queried location u its answer list is the singleton $\{\pi(u)\}$, so by (16), $\mathrm{wt}(\boldsymbol{G}_1 \cdot \boldsymbol{v}) \leq B \cdot \sum_u d_u = Y$, where $Y \coloneqq B \cdot 2^{\mathsf{R}} \cdot \mathsf{q}$. We append $H = \Gamma \cdot Y + 1$ zero coordinates to the code and to targets of feasible systems. For an infeasible system, the target is one on these coordinates and zero elsewhere, at distance $H > \Gamma \cdot Y$ from the code (Construction 6.2).

Let $\boldsymbol{v}$ be a solution in a NO instance, and abbreviate $\Lambda_u = \Lambda_u(\boldsymbol{v})$. By (15), every random string r has an accepting view $(V_{\mathbb{x},r}, \boldsymbol{a}) \in \mathcal{V}_r$ with $v_{r,V_{\mathbb{x},r},\boldsymbol{a}} \neq 0$, and its answers lie in the lists $\Lambda_{u_{r,1}}, \ldots, \Lambda_{u_{r,\mathsf{q}}}$. Distinct answers at the same location are thus unavoidable: if every list had a single answer, there would be a proof accepted by the verifier for every random string. Independently of r, we sample a proof π by choosing $\pi(u)$ independently and uniformly from Λ_u for each queried location u; the remaining locations are arbitrary. For a fixed r, the sampled proof agrees with this view with probability at least $\prod_{j=1}^{\mathsf{q}} |\Lambda_{u_{r,j}}|^{-1}$. We write

$$\overline{\Lambda} \coloneqq \frac{\sum_{u \in \mathcal{U}} d_u \cdot |\Lambda_u|}{2^{\mathsf{R}} \cdot \mathsf{q}},$$

and apply Jensen's inequality twice:

$$\Pr_{\pi,r}[V_{\mathbb{x},r}(\pi|_{I_r}) = 1] \geq \mathbb{E}_r \left[\prod_{j=1}^{\mathsf{q}} |\Lambda_{u_{r,j}}|^{-1} \right] \geq \overline{\Lambda}^{-\mathsf{q}}.$$

Every fixed proof is accepted with probability at most s, so the same bound holds after averaging over π. Hence $\overline{\Lambda} \geq \mathsf{s}^{-1/\mathsf{q}}$, and by (16),

$$\mathrm{wt}(\boldsymbol{G}_1 \cdot \boldsymbol{v}) \geq \delta \cdot Y \cdot \overline{\Lambda} \geq \delta \cdot \mathsf{s}^{-1/\mathsf{q}} \cdot Y.$$

We therefore prove soundness for every gap $\Gamma < \delta \cdot \mathsf{s}^{-1/\mathsf{q}}$ (Theorem 6.4).

Parameters. Arnon, Chiesa, and Yogev [ACY23] construct IOPs with perfect completeness, soundness error $1/\mathrm{poly}(N)$, and $O(\log \log N)$ queries. By unrolling as in Arnon, Bhangale, Chiesa, and Yogev [ABCY22, Lemma 4.6], with the query locations determined by the verifier's randomness, we construct query-oblivious PCPs with soundness error $\mathsf{s} = 1/N$ and

$$\mathsf{q} = O(\log \log N), \quad |\Sigma| = \mathrm{poly}(N), \quad \mathsf{R} = O(\log N \cdot \log \log N), \quad |\mathcal{D}_r| = \mathrm{poly}(N)$$

(Theorem 6.5). With a linear code of constant rate and constant relative distance, the block length is $M = (|\Sigma| + \Gamma) \cdot Y + 1 = N^{O(\log \log N)}$, and for a polynomial-size verifier the reduction runs in time $n_{\mathscr{x}}{}^{O(\log \log n_{\mathscr{x}})}$. Since $\delta \cdot \mathsf{s}^{-1/\mathsf{q}} = N^{\Omega(1/\log \log N)}$, the gap is $M^{\Omega(1/(\log \log M)^2)}$ (Corollary 6.6).

An upper bound on the gap. Suppose $\mathsf{q} \geq 2$, and let the system be feasible for a NO instance. Then some predicate $V_{\mathscr{x},r}$ has an accepting view, and a proof consistent with that view is accepted whenever the verifier chooses r, so $\mathsf{s} \geq 2^{-\mathsf{R}}$. Since the block length M exceeds the number 2^{R} of random strings,

$$\delta \cdot \mathsf{s}^{-1/\mathsf{q}} \leq 2^{\mathsf{R}/\mathsf{q}} < M^{1/\mathsf{q}} \leq M^{1/2}.$$

With the moment method, we recover an accepting computation from any solution below the weight threshold (Section 4.4), without sampling a proof, and the factor is $M^{1-\varepsilon}$.

2.4 From codes to lattices

We also prove hardness for lattices in every ℓ_p norm, within $M^{1/\min\{p,2\}-\varepsilon}$, by applying known conversions to the moment reduction in characteristic 2 (Section 2.2).

Construction A. For the binary code $C = \ker \boldsymbol{G}_0$, we use the lattice $\mathcal{L} \coloneqq \{\boldsymbol{u} \in \mathbb{Z}^M : \boldsymbol{u} \bmod 2 \in C\}$ of Construction A [LS71, Section 2.1], as in [FM04]. The target remains $\boldsymbol{v}^*$, now viewed as an integer vector. For finite p, its ℓ_p distance to $\mathcal{L}$ is the p-th root of its Hamming distance to C (Lemma 3.9). The gap is therefore $\Gamma^{1/p}$, as desired for $1 \leq p \leq 2$.

Why the block structure matters. For $p > 2$, we need a gap of $\sqrt{\Gamma}$. The difficulty is clearest in ℓ_∞: every binary target has distance at most one from $\mathcal{L}$, regardless of its Hamming distance from C. We therefore transform the lattice and target so that their ℓ_∞ distance is $\Omega(\sqrt{\Gamma})$ in NO instances and at most 1 in YES instances. For this, we use more than a bound on the total weight: the honest solution has at most one 1 in each evaluation-table block, including those of the auxiliary tables in Section 2.2.

Hadamard blocks. Following Hair and Sahai's work on SVP [HS26b; HS26a], we multiply each block by a Sylvester–Hadamard matrix; the block size Q is a power of two. Writing $\mathcal{H}$ for the resulting block-diagonal matrix, we take $\mathcal{H} \cdot \mathcal{L}$ and $\mathcal{H} \cdot \boldsymbol{v}^*$ as the new lattice and target.

In a YES instance, the honest solution $\boldsymbol{v}$ satisfies $\boldsymbol{v}^* - \boldsymbol{v} \in \mathcal{L}$. Each block of $\mathcal{H} \cdot \boldsymbol{v}$ is zero or a column of the Hadamard matrix, so every entry is 0 or ± 1. The completeness radius is therefore at most $M^{1/p}$. In a NO instance, every integer residual $\boldsymbol{d} = \boldsymbol{v}^* - \boldsymbol{u}$, with $\boldsymbol{u} \in \mathcal{L}$, has $\Omega(\Gamma)$ odd entries per block on average. By orthogonality of the columns of the Hadamard matrix, $\|\mathcal{H} \cdot \boldsymbol{d}\|_2 / \sqrt{M} = \Omega(\sqrt{\Gamma})$. For every $2 \le p \le \infty$,

$$\frac{\|\mathcal{H} \cdot \boldsymbol{d}\|_p}{M^{1/p}} \ge \frac{\|\mathcal{H} \cdot \boldsymbol{d}\|_2}{\sqrt{M}}.$$

The gap is therefore of order $\sqrt{\Gamma}$ for $2 < p \le \infty$, with $M^{1/p} = 1$ when $p = \infty$ (Lemma 3.12).

In both cases, the gap is of order $\Gamma^{1/\min\{p,2\}}$. Since the code, its block partition, and $\mathcal{H}$ depend only on the formula length, the transformed lattice is fixed and only the target depends on the formula.

3 Preliminaries

Let $1 \le p \le \infty$ be a norm exponent, where $1/p$ is read as 0 for $p = \infty$, and let $\|\cdot\|_p$ denote the ℓ_p norm. We write $\mathrm{wt}(\boldsymbol{w})$ for the number of nonzero coordinates of a vector $\boldsymbol{w}$ and $\Delta(\boldsymbol{v}, \boldsymbol{w})$ for the number of coordinates that differ between $\boldsymbol{v}$ and $\boldsymbol{w}$.

Let $\mathbb{F}$ denote a field of characteristic p.

- A ***linear code*** C over $\mathbb{F}$ is a linear subspace of $\mathbb{F}^M$.
- A ***lattice*** $\mathcal{L} \subseteq \mathbb{R}^M$ is a discrete additive subgroup of $\mathbb{R}^M$. Equivalently, $\mathcal{L} = \boldsymbol{B} \cdot \mathbb{Z}^k$ for a matrix $\boldsymbol{B} \in \mathbb{R}^{M \times k}$ with linearly independent columns. These columns form a basis, and k is the rank of the lattice. Throughout this paper, lattices have full rank $k = M$, so their bases are nonsingular $M \times M$ matrices and their dimension is M.

Definition 3.1. *For a dimension $M \in \mathbb{N}$, a norm exponent $1 \le p \le \infty$, a code $C \subseteq \mathbb{F}^M$ over a finite field $\mathbb{F}$, a lattice $\mathcal{L} \subseteq \mathbb{R}^M$, and targets $\boldsymbol{y} \in \mathbb{F}^M$ and $\boldsymbol{t} \in \mathbb{R}^M$, define*

$$\Delta(\boldsymbol{y}, C) \coloneqq \min_{\boldsymbol{c} \in C} \Delta(\boldsymbol{y}, \boldsymbol{c}), \qquad \mathrm{dist}_p(\boldsymbol{t}, \mathcal{L}) \coloneqq \min_{\boldsymbol{v} \in \mathcal{L}} \|\boldsymbol{t} - \boldsymbol{v}\|_p .$$

3.1 Codes and lattices with preprocessing

Definition 3.2. *A* **preprocessing algorithm** $\mathcal{A}$ *consists of a pair of algorithms $\mathcal{A} \coloneqq (\mathcal{A}_0, \mathcal{A}_1)$*

- $\mathcal{A}_0(\mathsf{pp}) \to \mathsf{aux}$ *receives a* **fixed input** pp *and outputs a string* aux*, the* **preprocessing output***, with length $|\mathsf{aux}|$ polynomial in $|\mathsf{pp}|$.*
- $\mathcal{A}_1(\mathsf{aux}, \mathbb{x})$ *receives the preprocessing output and an* **online input** $\mathbb{x}$*.*

The preprocessing phase $\mathcal{A}_0$ may run for arbitrary time, while the online phase $\mathcal{A}_1$ must run in time polynomial in $|\mathsf{aux}| + |\mathbb{x}|$. We write $\mathcal{A}(\mathsf{pp}, \mathbb{x})$ for the output of running $\mathcal{A}_0, \mathcal{A}_1$ in sequence. If $\mathcal{A}_0$ also runs in time polynomial in $|\mathsf{pp}|$, we say $\mathcal{A}$ is a **polynomial-time preprocessing algorithm***.*

Definition 3.3. *A* **promise problem with preprocessing** *is a pair* $\Pi \coloneqq (\Pi_{\text{YES}}, \Pi_{\text{NO}})$ *of disjoint sets of instances* $(\mathsf{pp}, \mathbb{x})$*. A preprocessing algorithm* $\mathcal{A}$ solves Π *if*

$$\mathcal{A}(\mathsf{pp}, \mathbb{x}) = \begin{cases} 1, & (\mathsf{pp}, \mathbb{x}) \in \Pi_{\text{YES}}, \\ 0, & (\mathsf{pp}, \mathbb{x}) \in \Pi_{\text{NO}}\,. \end{cases}$$

No correctness requirement is imposed on instances outside $\Pi_{\text{YES}} \cup \Pi_{\text{NO}}$*.*

Definition 3.4 (Preprocessing reduction)**.** *A* **preprocessing reduction** *from* $\Pi \coloneqq (\Pi_{\text{YES}}, \Pi_{\text{NO}})$ *to* $\Pi' \coloneqq (\Pi'_{\text{YES}}, \Pi'_{\text{NO}})$ *is a polynomial-time preprocessing algorithm* $\mathcal{A} \coloneqq (\mathcal{A}_0, \mathcal{A}_1)$ *whose preprocessing output* aux *contains a specified* pp'*, and whose online output is* $\mathbb{x}'$*, such that:*

- *if* $(\mathsf{pp}, \mathbb{x}) \in \Pi_{\text{YES}}$*, then* $(\mathsf{pp}', \mathbb{x}') \in \Pi'_{\text{YES}}$*, and*
- *if* $(\mathsf{pp}, \mathbb{x}) \in \Pi_{\text{NO}}$*, then* $(\mathsf{pp}', \mathbb{x}') \in \Pi'_{\text{NO}}$*.*

In particular, pp' *depends only on* pp*.*

Claim 3.5 (Preprocessing as advice, cf. [FM04, Section 1])**.** *For the decision problem* $\mathbb{x} \in \mathsf{L}$*, where* $\mathsf{L} \subseteq \{0,1\}^*$*, the fixed input is* $1^{n_{\mathbb{x}}}$ *and the online input is* $\mathbb{x} \in \{0,1\}^{n_{\mathbb{x}}}$*. Suppose there is a preprocessing reduction from this problem to a promise problem* Π *with preprocessing. If a preprocessing algorithm* $\mathcal{A}$ *solves* Π*, then* $\mathsf{L} \in \mathrm{P/poly}$*. If* $\mathcal{A}$ *is a polynomial-time preprocessing algorithm, then* $\mathsf{L} \in \mathrm{P}$*.*

We define the nearest codeword problem with preprocessing following [KPV14, full version, Definition 2.5].

Definition 3.6 (Nearest codeword with preprocessing)**.** *In the* **nearest codeword problem with preprocessing (NCPP)** *over* $\mathbb{F}$*, we seek a codeword nearest to an online target. In the preprocessing phase, the algorithm receives a generator matrix of a linear code* $C \subseteq \mathbb{F}^M$*. In the online phase, it receives* $\boldsymbol{y} \in \mathbb{F}^M$ *and outputs* $\boldsymbol{c} \in C$*. For* $f \geq 1$*, in the search problem* f**-approximate NCPP***, the task is to find* $\boldsymbol{c} \in C$ *such that* $\Delta(\boldsymbol{y}, \boldsymbol{c}) \leq f \cdot \Delta(\boldsymbol{y}, C)$*. In* f**-gap NCPP** *(*$f \geq 1$*), the online input also includes a positive radius* R*, and the task is to distinguish* $\Delta(\boldsymbol{y}, C) \leq R$ *from* $\Delta(\boldsymbol{y}, C) > f \cdot R$*, under the promise that one holds.*

For the closest vector problem with preprocessing, we follow the formulation in [FM04; KPV14].

Definition 3.7 (Closest vector with preprocessing)**.** *In the* **closest vector problem with preprocessing (CVPP)***, distance is measured in the* ℓ_p *norm, for* $1 \leq p \leq \infty$*. In the preprocessing phase, the algorithm receives a basis* $\boldsymbol{B} \in \mathbb{Q}^{M \times M}$ *of a lattice* $\mathcal{L} \coloneqq \boldsymbol{B} \cdot \mathbb{Z}^M$*. In the online phase, it receives* $\boldsymbol{t} \in \mathbb{Q}^M$ *and outputs* $\boldsymbol{v} \in \mathcal{L}$*. For* $f \geq 1$*, in the search problem* f**-approximate CVPP***, the task is to find* $\boldsymbol{v} \in \mathcal{L}$ *such that* $\|\boldsymbol{t} - \boldsymbol{v}\|_p \leq f \cdot \mathrm{dist}_p(\boldsymbol{t}, \mathcal{L})$*. In* f**-gap CVPP** *(*$f \geq 1$*), the online input also includes a positive radius* R*, and the task is to distinguish* $\mathrm{dist}_p(\boldsymbol{t}, \mathcal{L}) \leq R$ *from* $\mathrm{dist}_p(\boldsymbol{t}, \mathcal{L}) > f \cdot R$*, under the promise that one holds.*

3.2 Conversions to codes and lattices

Khot, Popat, and Vishnoi call the binary version of the following optimization problem *Minimum Weight Solution Problem with Preprocessing* (MWSPP) [KPV14, Definition 2.4]; their B_f, B_v, t correspond to our $\boldsymbol{G}_0, \boldsymbol{G}_1, \boldsymbol{b}$, respectively.

Lemma 3.8 (adapted from [KPV14, Section 2.1])**.** *Fix a finite field* $\mathbb{F}$*. There exists a polynomial-time preprocessing algorithm that:*

- *in the preprocessing phase, given matrices* $\boldsymbol{G}_0 \in \mathbb{F}^{h\times m}$ *and* $\boldsymbol{G}_1 \in \mathbb{F}^{M\times m}$*, outputs a generator matrix of a linear code* $C \subseteq \mathbb{F}^M$*, and*
- *in the online phase, given* $\boldsymbol{b} \in \mathbb{F}^h$*, outputs a target* $\boldsymbol{y}_{\boldsymbol{b}} \in \mathbb{F}^M$*,*

such that (whenever $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}$ *is feasible):*

$$\Delta(\boldsymbol{y}_{\boldsymbol{b}}, C) = \min_{\boldsymbol{G}_0\cdot\boldsymbol{v}=\boldsymbol{b}} \mathrm{wt}(\boldsymbol{G}_1 \cdot \boldsymbol{v})\,.$$

Lemma 3.9 ([FM04, Section 4])**.** *Fix a norm exponent* $1 \le p < \infty$*. There exists a polynomial-time preprocessing algorithm that:*

- *in the preprocessing phase, given a generator matrix of a binary linear code* $C \subseteq \mathbb{F}_2^M$*, outputs a basis of a full-rank lattice* $\mathcal{L} \subseteq \mathbb{Z}^M$*, and*
- *in the online phase, given* $\boldsymbol{y} \in \mathbb{F}_2^M$*, outputs a target* $\boldsymbol{t}_{\boldsymbol{y}} \in \mathbb{Z}^M$*,*

with the following guarantees:

1. **Distance.** $\mathrm{dist}_p(\boldsymbol{t}_{\boldsymbol{y}}, \mathcal{L})^p = \Delta(\boldsymbol{y}, C)$*.*
2. **Solution conversion.** *Given any* $\boldsymbol{v} \in \mathcal{L}$*, one can compute in polynomial time a codeword* $\boldsymbol{c} \in C$ *such that* $\Delta(\boldsymbol{y}, \boldsymbol{c}) \le \|\boldsymbol{t}_{\boldsymbol{y}} - \boldsymbol{v}\|_p^p$*.*

We reduce the following problem to gap CVPP and gap NCPP. The lattice conversion is adapted from [HS26a].

Definition 3.10 (Block-sparse linear systems with preprocessing)**.** *Fix a prime* p *and integers* $D, Q, \Gamma \ge 1$ *with* $Q > \Gamma$*. In the* **block-sparse linear-system problem with preprocessing***, denoted* (D, Q, Γ)**-BSLSP***, the fixed input is a matrix* $\boldsymbol{A} \in \mathbb{F}_{\mathsf{p}}^{h\times n}$*, whose* $n \coloneqq D \cdot Q$ *columns are partitioned into* D *blocks of size* Q*. The online input is* $\boldsymbol{b} \in \mathbb{F}_{\mathsf{p}}^h$*. The task is to distinguish:*

- **YES:** $\boldsymbol{A} \cdot \boldsymbol{v} = \boldsymbol{b}$ *has a solution* $\boldsymbol{v} \in \{0,1\}^n$ *with at most one nonzero coordinate per block,*
- **NO:** *every solution* $\boldsymbol{v} \in \mathbb{F}_{\mathsf{p}}^n$ *to* $\boldsymbol{A} \cdot \boldsymbol{v} = \boldsymbol{b}$ *has* $\mathrm{wt}(\boldsymbol{v}) > \Gamma \cdot D$*,*

under the promise that one holds. We call Γ *the gap.*

Lemma 3.11. *Fix a finite field* $\mathbb{F}$ *of characteristic* p*. There is a preprocessing reduction from* (D, Q, Γ)*-BSLSP over* $\mathbb{F}_{\mathsf{p}}$ *to* Γ*-gap NCPP over* $\mathbb{F}$*. The output code has block length* $M = 2 \cdot D \cdot Q$*, and the output radius is* $R = D$*.*

Lemma 3.12 (adapted from [HS26a, Sections 8 and 10])**.** *For a prime* p *and rational* $p \ge 1$ *or* $p = \infty$*, let* $s \coloneqq \min\{p, 2\}$*. For every* $f \ge 1$ *and* $\Gamma > 16 \cdot f^s$*, there is a preprocessing reduction from* (D, Q, Γ)*-BSLSP over* $\mathbb{F}_{\mathsf{p}}$ *to* f*-gap CVPP in* ℓ_p*. The construction is independent of* f*. The output consists of an integer basis and target* $\boldsymbol{t}_{\boldsymbol{b}}$ *for a full-rank lattice* $\mathcal{L} \subseteq \mathbb{Z}^M$*, with* $2\cdot D\cdot Q \le M < 4\cdot D\cdot Q$*, and an integer radius* $1 \le R \le M$ *depending only on the fixed input.*

In polynomial time, one can check membership in $\mathcal{L}$ *and, given* $\boldsymbol{v} \in \mathcal{L}$*, compute an integer* $w \ge 0$ *such that*

$$w \le 16 \cdot D \cdot \left(\frac{\|\boldsymbol{t}_{\boldsymbol{b}} - \boldsymbol{v}\|_p}{R}\right)^s \quad \textit{for all instances,} \qquad w > \Gamma \cdot D \quad \textit{for NO instances}\,.$$

Hair and Sahai [HS26a] convert a code built from the instance into a lattice for SVP in ℓ_p with $p > 2$. Here the lattice depends only on the fixed input. We cover every rational $p \ge 1$ and $p = \infty$, with $s = \min\{p, 2\}$, and use an integer radius depending only on the fixed input (Section B.3).

3.3 Verifiers

Definition 3.13. *A* **verifier** *is a Boolean circuit* $\mathbf{V}\colon \{0,1\}^{n_{\mathbb{x}}} \times \{0,1\}^{n_{\mathbb{w}}} \to \{0,1\}$ *with instance length* $n_{\mathbb{x}} \geq 1$*, witness length* $n_{\mathbb{w}}$*, gates of fan-in at most two, and a designated output gate. The instance is* $\mathbb{x} \in \{0,1\}^{n_{\mathbb{x}}}$ *and the witness is* $\mathbb{w} \in \{0,1\}^{n_{\mathbb{w}}}$*. The accepted language is*

$$\mathsf{L}_{\mathbf{V}} \coloneqq \{\mathbb{x} \,:\, \exists \mathbb{w}\ \mathbf{V}(\mathbb{x},\mathbb{w}) = 1\}\ .$$

An instance $\mathbb{x}$ *is a* **YES instance** *if* $\mathbb{x} \in \mathsf{L}_{\mathbf{V}}$*, and a* **NO instance** *otherwise.*

Definition 3.14. *For a verifier* $\mathbf{V}$ *with* $n_{\mathbb{x}}$ *input bits,* $n_{\mathbb{w}}$ *witness bits, and* $g_{\mathbf{V}}$ *gates, an* **assignment** $\boldsymbol{z} \in \{0,1\}^N$*, where* $N \coloneqq n_{\mathbb{x}} + n_{\mathbb{w}} + g_{\mathbf{V}}$*, lists, in order, the* **input** *bits* $\mathbb{x}$*, the output of the designated gate, the* **witness** *bits* $\mathbb{w}$*, and the remaining gate outputs. Its first* $n_{\mathbb{x}} + 1$ *coordinates form the* **prefix***. The assignment is* **valid** *if every gate is evaluated correctly, and an* **accepting computation on** $\mathbb{x}$ *if it is valid, its input is* $\mathbb{x}$*, and its output is* 1 *(so its prefix is* $(\mathbb{x}, 1)$*).*

For each gate, we interpolate its Boolean truth table over $\mathbb{F}_{\mathsf{p}}$ and homogenize the gate equation with the constant coordinate $z_0 \coloneqq 1$.

Claim 3.15 (Quadratic gate equations, cf. [DHK15, Lemma 6.8])**.** *Fix a prime* p *and let* $\mathbf{V}$ *be a verifier with* N *assignment coordinates and* $g_{\mathbf{V}}$ *gates. In deterministic polynomial time, one can compute coefficients* $\mathsf{c}^{(b)}_{(i,j)} \in \mathbb{F}_{\mathsf{p}}$ *of quadratic* **gate equations** *such that, for every* $\boldsymbol{z} \in \{0,1\}^N$*, setting* $z_0 \coloneqq 1$*,*

$$\boldsymbol{z} \text{ is valid} \quad\Longleftrightarrow\quad \hat{g}_b(1,\boldsymbol{z}) \coloneqq \sum_{i,j=0}^{N} \mathsf{c}^{(b)}_{(i,j)} \cdot z_i \cdot z_j = 0 \quad (1 \leq b \leq g_{\mathbf{V}})\,.$$

4 The moment reduction with a fixed matrix

For a fixed verifier $\mathbf{V}$, we construct $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\mathbb{x}}$, with $\boldsymbol{G}_0$ independent of $\mathbb{x}$. Correctness is proved for any moment representation satisfying Definition 4.4. We construct one with suitable parameters later (Section 4.5 and appendix A).

4.1 Definitions and setup

Definition 4.1 (Coefficient functions)**.** *For a field* $\mathbb{F}$ *and an extension field* $\mathbb{E}$ *of* $\mathbb{F}$*, a* **coefficient function** *is a function* $\omega\colon \mathbb{E} \to \mathbb{F}$ *with finite support* $\operatorname{supp}\omega \coloneqq \{\alpha : \omega(\alpha) \neq 0\}$*.*

Definition 4.2 (Moments, cf. [Ope26, Chapter 7, Section 3.3])**.** *For a coefficient function* $\omega\colon \mathbb{E} \to \mathbb{F}$ *and an integer* $k \geq 0$*, its* k**-th moment** *is*

$$\mu_k(\omega) \coloneqq \sum_{\alpha \in \mathbb{E}} \omega(\alpha) \cdot \alpha^k \in \mathbb{E}\,,$$

with the convention $0^0 \coloneqq 1$*.*

For an extension field $\mathbb{E}/\mathbb{F}$ and $\alpha \in \mathbb{E}$, we define the ***indicator*** $\mathbf{1}_\alpha\colon \mathbb{E} \to \mathbb{F}$ by

$$\mathbf{1}_\alpha(\alpha') \coloneqq \begin{cases} 1, & \alpha' = \alpha, \\ 0, & \alpha' \neq \alpha \end{cases}\ .$$

In particular, $\mu_k(\mathbf{1}_\alpha) = \alpha^k$ for every $\alpha \in \mathbb{E}$ and $k \geq 0$.

Lemma 4.3 ([Son26, Lemma 6.1])**.** *Let $\mathbb{E}/\mathbb{F}$ be a field extension, $t \geq 1$ an integer, and $\omega, \omega' \colon \mathbb{E} \to \mathbb{F}$ be coefficient functions. If $|\operatorname{supp}(\omega - \omega')| \leq t$, then*

$$\omega = \omega' \quad \Longleftrightarrow \quad \mu_k(\omega) = \mu_k(\omega') \quad (0 \leq k < t)\,.$$

Proof. Equal coefficient functions have equal moments. For the converse, we write $\operatorname{supp}(\omega - \omega') = \{\alpha_1, \ldots, \alpha_s\}$, where $s \leq t$. If $s = 0$, then $\omega = \omega'$. Otherwise, the first s moment equations are

$$(\alpha_j^k)_{\substack{0 \leq k < s \\ 1 \leq j \leq s}} \cdot \big(\omega(\alpha_j) - \omega'(\alpha_j)\big)_{1 \leq j \leq s} = \mathbf{0}\,.$$

The Vandermonde matrix is invertible over $\mathbb{E}$ because the support points are distinct, so $\omega(\alpha_j) = \omega'(\alpha_j)$ for every j. □

We adapt the evaluation tables and moment maps of [Son26, Definitions 3.7–3.9 and Lemma 5.3] to represent single field elements by evaluation tables over $\mathbb{F}_{\mathsf{p}}$.

Definition 4.4 (Moment representations and evaluation tables)**.** *Let $\mathbb{E}$ be a finite field of prime characteristic p, with prime subfield $\mathbb{F}_{\mathsf{p}}$. A* **moment representation** *over $\mathbb{E}$ is a tuple*

$$(\kappa, D, Q, \mathcal{P}, \boldsymbol{T}, (\hat{\mu}_k)_{k=0}^{2\cdot\kappa-2})\,,$$

where

- *κ, D, Q are positive integers called the* **moment parameter***, the* **number of blocks per table***, and the* **block size***, respectively. We denote this count by D_{tab} in Construction 4.5. The* **evaluation table length** *is $m_{\mathrm{tab}} \coloneqq D \cdot Q$,*
- *$\mathcal{P}$ is a partition of the coordinates into D blocks of size Q,*
- *$\boldsymbol{T}$ is a matrix over $\mathbb{F}_{\mathsf{p}}$ with m_{tab} columns,*
- *the* **moment maps** *$\hat{\mu}_k \colon \mathbb{F}_{\mathsf{p}}^{m_{\mathrm{tab}}} \to \mathbb{E}$ are $\mathbb{F}_{\mathsf{p}}$-linear for $0 \leq k \leq 2 \cdot \kappa - 2$. These maps are marked with a hat to distinguish them from the moments μ_k of coefficient functions.*

A vector $\boldsymbol{v} \in \mathbb{F}_{\mathsf{p}}^{m_{\mathrm{tab}}}$ is an **evaluation table***, and the equations $\boldsymbol{T} \cdot \boldsymbol{v} = \mathbf{0}$ are its* **evaluation table constraints***. We require the following two properties.*

1. **Completeness.** *For every $\alpha \in \mathbb{E}$, there is $\boldsymbol{v} \in \{0,1\}^{m_{\mathrm{tab}}}$ with exactly one nonzero coordinate in each block such that*

$$\boldsymbol{T} \cdot \boldsymbol{v} = \mathbf{0}\,, \qquad \hat{\mu}_k(\boldsymbol{v}) = \alpha^k \quad (0 \leq k \leq 2 \cdot \kappa - 2)\,.$$

2. **Soundness.** *For every $\boldsymbol{v} \in \mathbb{F}_{\mathsf{p}}^{m_{\mathrm{tab}}}$ with $\boldsymbol{T} \cdot \boldsymbol{v} = \mathbf{0}$ and weight below the* weight threshold *$\kappa \cdot D/2$, there are a finite extension $\mathbb{E}_{\boldsymbol{v}}/\mathbb{E}$ and a coefficient function $\omega \colon \mathbb{E}_{\boldsymbol{v}} \to \mathbb{F}_{\mathsf{p}}$ such that*

$$\hat{\mu}_k(\boldsymbol{v}) = \mu_k(\omega) \quad (0 \leq k \leq 2 \cdot \kappa - 2)\,, \qquad |\operatorname{supp}\omega| \leq \frac{2 \cdot \mathrm{wt}(\boldsymbol{v})}{D}\,.$$

We say that an evaluation table $\boldsymbol{v}$ **represents** *a coefficient function $\omega \colon \mathbb{E}' \to \mathbb{F}_{\mathsf{p}}$ if $\hat{\mu}_k(\boldsymbol{v}) = \mu_k(\omega)$ for $0 \leq k \leq 2 \cdot \kappa - 2$, where $\mathbb{E}'$ is a finite extension of $\mathbb{E}$.*

4.2 The construction and its guarantee

We use product evaluation tables to encode a Boolean assignment $\boldsymbol{z}$, with coordinates ordered as in Definition 3.14. Set $z_0 \coloneqq 1$ and write $\alpha \coloneqq \sum_{i=1}^{N} \beta_i \cdot z_i$ for its encoding. We intend to choose each product evaluation table $\boldsymbol{v}_{i,j}$ so that

$$\hat{\mu}_k(\boldsymbol{v}_{i,j}) = z_i \cdot z_j \cdot \alpha^k \quad (0 \le k \le 2 \cdot \kappa - 2)\,.$$

Since $z_i^2 = z_i$ and $z_i \cdot z_j = z_j \cdot z_i$, we use the same evaluation table for diagonal and linear terms, and for the two orders of a product. We use auxiliary evaluation tables for translations in the coordinates outside the prefix. For solutions below the weight threshold, the coefficients of valid assignments with prefix $(\mathbb{x}, 1)$ sum to 1 (Section 4.4).

Construction 4.5 (Fixed-verifier linear system)**.** The ingredients are:

- A Boolean verifier $\mathbf{V}$ with $n_{\mathbb{x}}$ input bits, $n_{\mathbb{w}}$ witness bits, and $g_{\mathbf{V}}$ gates of fan-in at most two.
- A moment representation $(\kappa, D_{\mathrm{tab}}, Q, \mathcal{P}, \boldsymbol{T}, (\hat{\mu}_k)_{k=0}^{2\cdot\kappa-2})$ over a finite field $\mathbb{E}$ of characteristic p, where $\kappa \ge 2$. The evaluation table length is $m_{\mathrm{tab}} \coloneqq D_{\mathrm{tab}} \cdot Q$.
- A tuple $\boldsymbol{\beta} \coloneqq (\beta_1, \ldots, \beta_N)$ of $\mathbb{F}_{\mathsf{p}}$-linearly independent elements of $\mathbb{E}$.

We construct a linear system $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\mathbb{x}}$ with a partition of its columns into blocks, as follows:

1. ***Variables.*** Let $N \coloneqq n_{\mathbb{x}} + n_{\mathbb{w}} + g_{\mathbf{V}}$. The product and auxiliary evaluation tables are

$$\begin{aligned} &\boldsymbol{v}_{0,0}, \quad \boldsymbol{v}_{i,j} \in \mathbb{F}_{\mathsf{p}}^{m_{\mathrm{tab}}} && (0 \le j < i \le N), \\ &\boldsymbol{v}^{(i)} \in \mathbb{F}_{\mathsf{p}}^{m_{\mathrm{tab}}} && (n_{\mathbb{x}} + 1 < i \le N)\,. \end{aligned}$$

   Symmetric indices and diagonal indices denote the same evaluation tables as follows:

$$\boldsymbol{v}_{0,i} \coloneqq \boldsymbol{v}_{i,0}, \qquad \boldsymbol{v}_{i,i} \coloneqq \boldsymbol{v}_{i,0} \quad (1 \le i \le N), \qquad \boldsymbol{v}_{i,j} \coloneqq \boldsymbol{v}_{j,i} \quad (i < j)\,.$$

   The vector $\boldsymbol{v}$ is the concatenation of the distinct product evaluation tables and the auxiliary evaluation tables. The number of evaluation tables is $\sigma \coloneqq 1 + \binom{N+1}{2} + (N - n_{\mathbb{x}} - 1) \le (N+1)^2$.

2. ***Blocks and dimensions.*** Each evaluation table has a copy of the partition $\mathcal{P}$ from the moment representation. The coordinates of $\boldsymbol{v}$ are partitioned by these copies into $\sigma \cdot D_{\mathrm{tab}}$ blocks, each of size Q. The dimensions are

$$D \coloneqq \sigma \cdot D_{\mathrm{tab}}, \qquad Q = m_{\mathrm{tab}}/D_{\mathrm{tab}}, \qquad n_{\mathrm{sys}} \coloneqq \sigma \cdot m_{\mathrm{tab}} = D \cdot Q\,.$$

3. ***Affine constraints, matrix, and right-hand side.*** For $\mathbb{x} \in \{0,1\}^{n_{\mathbb{x}}}$, the affine constraints are as follows.

   (a) ***Evaluation table constraints.*** For every product evaluation table and auxiliary evaluation table, we impose

$$\begin{aligned} &\boldsymbol{T} \cdot \boldsymbol{v}_{0,0} = \mathbf{0}\,, \qquad \boldsymbol{T} \cdot \boldsymbol{v}_{i,j} = \mathbf{0} && (0 \le j < i \le N), \\ &\boldsymbol{T} \cdot \boldsymbol{v}^{(i)} = \mathbf{0} && (n_{\mathbb{x}} + 1 < i \le N) \end{aligned}\,.$$

(b) ***Encoding and gate equations.*** We compute the coefficients $(\mathsf{c}^{(b)}_{(i,j)})_{i,j,b}$ of the gate equations as in Claim 3.15. For every $0 \le k < \kappa$, we impose the *moment equations* of order k

$$\begin{aligned} \hat{\mu}_{k+1}(\boldsymbol{v}_{0,j}) - \sum_{i=1}^{N} \beta_i \cdot \hat{\mu}_k(\boldsymbol{v}_{i,j}) = 0 \quad (0 \le j \le N)\,, \\ \sum_{i=0}^{N}\sum_{j=0}^{N} \mathsf{c}^{(b)}_{(i,j)} \cdot \hat{\mu}_k(\boldsymbol{v}_{i,j}) = 0 \quad (1 \le b \le g_{\mathbf{V}})\,. \end{aligned} \tag{17}$$

(c) ***Translation equations.*** We encode the prefix $(\mathbb{x}, 1)$ as $\gamma_{\mathbb{x}} \coloneqq \sum_{i=1}^{n_{\mathbb{x}}} \beta_i \cdot \mathbb{x}_i + \beta_{n_{\mathbb{x}}+1}$ and impose

$$\hat{\mu}_k(\boldsymbol{v}_{0,0}) - \sum_{i=n_{\mathbb{x}}+2}^{N} \sum_{j=0}^{k-1} \binom{k}{j} \cdot \beta_i^{k-j} \cdot \hat{\mu}_j(\boldsymbol{v}^{(i)}) = \gamma_{\mathbb{x}}^k \quad (0 \le k \le 2\cdot\kappa - 2)\,. \tag{18}$$

In particular, the $k = 0$ translation equation is $\hat{\mu}_0(\boldsymbol{v}_{0,0}) = 1$.

In a fixed $\mathbb{F}_{\mathsf{p}}$-basis of $\mathbb{E}$, we expand these equations into $\mathbb{F}_{\mathsf{p}}$ coordinates and collect all variable terms on the left. The resulting system is $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\mathbb{x}}$, with columns ordered as in $\boldsymbol{v}$. The coefficient matrix $\boldsymbol{G}_0$ has n_{sys} columns and is independent of $\mathbb{x}$. The right-hand side $\boldsymbol{b}_{\mathbb{x}}$ consists of zeros for the evaluation table constraints and the encoding and gate equations, and the $\mathbb{F}_{\mathsf{p}}$ coordinates of $\gamma_{\mathbb{x}}^k$ for the translation equations, in the corresponding row order.

Theorem 4.6. *Consider Construction 4.5, and let $\Gamma \ge 1$ be an integer satisfying*

$$\Gamma < Q, \qquad 2\cdot\Gamma\cdot\sigma < \kappa\,.$$

Then the construction is a preprocessing reduction from $\mathsf{L}_{\mathbf{V}}$ to (D, Q, Γ)-BSLSP over $\mathbb{F}_{\mathsf{p}}$.

The reduction satisfies:

- **Completeness.** *For every $\mathbb{x} \in \mathsf{L}_{\mathbf{V}}$, there is a binary vector $\boldsymbol{v} \in \{0,1\}^{n_{\mathrm{sys}}}$ satisfying $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\mathbb{x}}$ with at most one nonzero coordinate per block. In particular, $\mathrm{wt}(\boldsymbol{v}) \le D$.*
- **Soundness.** *For every $\mathbb{x} \in \{0,1\}^{n_{\mathbb{x}}} \setminus \mathsf{L}_{\mathbf{V}}$ and every $\boldsymbol{v} \in \mathbb{F}_{\mathsf{p}}^{n_{\mathrm{sys}}}$ satisfying $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\mathbb{x}}$, $\mathrm{wt}(\boldsymbol{v}) > \Gamma \cdot D$.*
- **Time.** *If the ingredients are given in a fixed $\mathbb{F}_{\mathsf{p}}$-basis of $\mathbb{E}$ in which field arithmetic takes polynomial time, then $\boldsymbol{G}_0$ and $\boldsymbol{b}_{\mathbb{x}}$ can be computed in time polynomial in $n_{\mathbb{x}}$ and the description length of the ingredients.*

4.3 Completeness

Proof of completeness in Theorem 4.6. For an accepting computation $\boldsymbol{z}$ on $\mathbb{x}$, with $z_0 \coloneqq 1$, we define $\alpha_i \coloneqq \sum_{j=1}^{i} \beta_j \cdot z_j$ for $n_{\mathbb{x}} + 1 \le i \le N$. Then $\alpha_{n_{\mathbb{x}}+1} = \gamma_{\mathbb{x}}$. For each $\alpha \in \{\alpha_i : n_{\mathbb{x}} + 1 \le i \le N\}$, there exists (by Definition 4.4) an evaluation table $\boldsymbol{q}_\alpha \in \{0,1\}^{m_{\mathrm{tab}}}$ with exactly one nonzero coordinate in each block satisfying

$$\boldsymbol{T} \cdot \boldsymbol{q}_\alpha = \boldsymbol{0}, \qquad \hat{\mu}_k(\boldsymbol{q}_\alpha) = \alpha^k \quad (0 \le k \le 2\cdot\kappa - 2)\,.$$

We define the *honest solution* $\boldsymbol{v}$ by

$$\boldsymbol{v}_{i,j} \coloneqq z_i \cdot z_j \cdot \boldsymbol{q}_{\alpha_N}, \qquad \boldsymbol{v}^{(i)} \coloneqq z_i \cdot \boldsymbol{q}_{\alpha_{i-1}}\,.$$

We verify each family of constraints and the weight bound.

- ***Evaluation table constraints.*** Each evaluation table is a scalar multiple of an evaluation table in the kernel of $\boldsymbol{T}$, so it satisfies the evaluation table constraints. The product evaluation tables are symmetric in i, j, and $\boldsymbol{v}_{i,i} = \boldsymbol{v}_{i,0}$ because $z_i^2 = z_i$ and $z_0 = 1$. By linearity of the moment maps,
$$\hat{\mu}_k(\boldsymbol{v}_{i,j}) = z_i \cdot z_j \cdot \alpha_N^k, \qquad \hat{\mu}_k(\boldsymbol{v}^{(i)}) = z_i \cdot \alpha_{i-1}^k \,.$$

- ***Encoding equations.*** For each j and $0 \le k < \kappa$, the left-hand side of the encoding equation in Equation 17 is
$$z_j \cdot \alpha_N^k \cdot \left(\alpha_N - \sum_{i=1}^{N} \beta_i \cdot z_i \right) = 0 \,,$$
since α_N is the encoding of $\boldsymbol{z}$.

- ***Gate equations.*** For each gate b and $0 \le k < \kappa$, the left-hand side of the gate equation in Equation 17 is
$$\alpha_N^k \cdot \sum_{i=0}^{N} \sum_{j=0}^{N} \mathsf{c}_{(i,j)}^{(b)} \cdot z_i \cdot z_j = \alpha_N^k \cdot \hat{g}_b(1, \boldsymbol{z}) = 0 \,,$$
because $\boldsymbol{z}$ is an accepting computation.

- ***Translation equations.*** We set $\omega_{0,0} \coloneqq \mathbf{1}_{\alpha_N}$ and $\omega^{(i)} \coloneqq z_i \cdot \mathbf{1}_{\alpha_{i-1}}$. Since $z_i \in \{0, 1\}$,
$$\omega^{(i)}(\alpha - \beta_i) - \omega^{(i)}(\alpha) = \mathbf{1}_{\alpha_i}(\alpha) - \mathbf{1}_{\alpha_{i-1}}(\alpha) \,.$$
Summing over $i > n_{\mathsf{x}} + 1$, we find the *translation identity*
$$\omega_{0,0}(\alpha) - \sum_{i=n_{\mathsf{x}}+2}^{N} \left(\omega^{(i)}(\alpha - \beta_i) - \omega^{(i)}(\alpha) \right) = \mathbf{1}_{\gamma_{\mathsf{x}}}(\alpha) \qquad (\alpha \in \mathbb{E}) \,. \tag{19}$$
The identity also holds after extending the functions by zero to any finite extension $\mathbb{E}'/\mathbb{E}$.

  We now compute the moments of a translation difference. Fix a coefficient function $f \colon \mathbb{E}' \to \mathbb{F}_{\mathsf{p}}$, a direction $u \in \mathbb{E}$, and an integer $k \ge 1$. Substituting $t = \alpha - u$ and expanding $(t + u)^k$, we obtain
$$\begin{aligned} \sum_{\alpha} \left(f(\alpha - u) - f(\alpha) \right) \cdot \alpha^k &= \sum_{t} f(t) \left((t + u)^k - t^k \right) \\ &= \sum_{j=0}^{k-1} \binom{k}{j} \cdot u^{k-j} \cdot \sum_{t} f(t) \cdot t^j . \end{aligned} \tag{20}$$
  The inner sum is $\mu_j(f)$, and its coefficient $\binom{k}{j} u^{k-j}$ depends only on k, j, u. In particular, for fixed k and u, the coefficients are independent of f. For $k = 0$, the sum on the right is empty and both sides are zero.

  We apply (20) with $f = \omega^{(i)}$ and $u = \beta_i$. The moments of $\omega_{0,0}$ and $\omega^{(i)}$ are those of $\boldsymbol{v}_{0,0}$ and $\boldsymbol{v}^{(i)}$, respectively. Taking moments in Equation 19, we therefore verify Equation 18.

- ***Binary entries and weight bound.*** Each evaluation table is either zero or a binary evaluation table with exactly one nonzero coordinate per block. Hence every evaluation table is binary with at most one nonzero coordinate per block, and the total weight is at most $\sigma \cdot D_{\mathrm{tab}} = D$.

□

4.4 Soundness

We recover a valid assignment wherever a reconstructed product coefficient is nonzero, even when the coefficients lie in an arbitrary prime field.

Lemma 4.7 (cf. [Son26, proof of Lemma 6.2])**.** *Consider Construction 4.5. Let Ξ be a finite subset of a finite extension of $\mathbb{E}$, and let $\omega_{i,j}\colon \Xi \to \mathbb{F}_{\mathsf{p}}$ $(0 \le i, j \le N)$ satisfy the following equations:*

$$\begin{aligned} \omega_{0,i} = \omega_{i,i} = \omega_{i,0} \quad & (1 \le i \le N), \\ \omega_{i,j} = \omega_{j,i} \quad & (0 \le i < j \le N)\,. \end{aligned}$$

Further assume that, for every $\alpha \in \Xi$,

$$\begin{aligned} \alpha \cdot \omega_{0,j}(\alpha) - \sum_{i=1}^{N} \beta_i \cdot \omega_{i,j}(\alpha) = 0 \quad & (0 \le j \le N), \\ \sum_{i,j=0}^{N} \mathsf{c}^{(b)}_{(i,j)} \cdot \omega_{i,j}(\alpha) = 0 \quad & (1 \le b \le g_{\mathbf{V}})\,. \end{aligned} \tag{21}$$

Then, for every $\alpha \in \Xi$ such that $\omega_{i,j}(\alpha) \neq 0$ for some $0 \le i, j \le N$, $\omega_{0,0}(\alpha) \neq 0$ and there is a unique valid assignment $\boldsymbol{z} \in \{0,1\}^N$ satisfying

$$\begin{aligned} \alpha &= \sum_{i=1}^{N} \beta_i \cdot z_i, \\ \omega_{i,j}(\alpha) &= \omega_{0,0}(\alpha) \cdot z_i \cdot z_j \qquad (0 \le i, j \le N)\,, \end{aligned}$$

where $z_0 \coloneqq 1$.

Proof. If $\omega_{0,0}(\alpha) = 0$, then $\omega_{i,0}(\alpha) = 0$ for every i by the $j = 0$ encoding equation and the independence of the β_i over $\mathbb{F}_{\mathsf{p}}$. By symmetry, $\omega_{0,j}(\alpha) = \omega_{j,0}(\alpha) = 0$ for every j. Thus, for each $1 \le j \le N$, $\sum_{i=1}^{N} \beta_i \cdot \omega_{i,j}(\alpha) = 0$. Since each $\omega_{i,j}(\alpha)$ lies in $\mathbb{F}_{\mathsf{p}}$, $\omega_{i,j}(\alpha) = 0$ for all $0 \le i, j \le N$ by the same linear independence.

Otherwise, we define $\boldsymbol{z} \in \mathbb{F}_{\mathsf{p}}^N$ by

$$z_i \coloneqq \frac{\omega_{i,0}(\alpha)}{\omega_{0,0}(\alpha)} \in \mathbb{F}_{\mathsf{p}} \quad (1 \le i \le N)\,.$$

By the $j = 0$ encoding equation, $\alpha = \sum_i \beta_i \cdot z_i$. After substituting this identity into the remaining encoding equations, we compare coefficients of the $\mathbb{F}_{\mathsf{p}}$-linearly independent elements β_i:

$$\omega_{i,j}(\alpha) = \omega_{0,0}(\alpha) \cdot z_i \cdot z_j \quad (0 \le i, j \le N)\,.$$

Since $\omega_{i,i} = \omega_{i,0}$, $z_i^2 = z_i$ and hence $z_i \in \{0,1\}$. After substitution and division by $\omega_{0,0}(\alpha)$ in the gate equations, $\hat{g}_b(1, \boldsymbol{z}) = 0$ for every b. Thus $\boldsymbol{z}$ is valid, and it is unique by the independence of the β_i. □

Proof of soundness in Theorem 4.6. Suppose that $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\boldsymbol{z}}$ and $\mathrm{wt}(\boldsymbol{v}) < \kappa \cdot D_{\mathrm{tab}}/2$. For each distinct evaluation table, we choose a coefficient function that it represents, on a finite extension of $\mathbb{E}$, as in Definition 4.4. We embed these extensions into a common finite extension $\mathbb{E}'$ of $\mathbb{E}$ and

extend each function by zero outside the image of its domain. The embeddings are the identity on $\mathbb{E}$, so the represented moments and the individual support sizes are unchanged.

Write the extended functions as $\omega_{i,j}, \omega^{(i)} \colon \mathbb{E}' \to \mathbb{F}_{\mathsf{p}}$, using the same function whenever the indices refer to the same evaluation table. By summing the support bounds for the individual evaluation tables,

$$|\operatorname{supp}\omega_{0,0}| + \sum_{i=1}^{N}\sum_{j=0}^{i-1} |\operatorname{supp}\omega_{i,j}| + \sum_{i=n_{\mathbb{x}}+2}^{N} |\operatorname{supp}\omega^{(i)}| \le \frac{2\cdot \operatorname{wt}(\boldsymbol{v})}{D_{\mathrm{tab}}} < \kappa\,.$$

The difference between the two sides of each equation in Equation 21 is an $\mathbb{E}'$-valued function on $\mathbb{E}'$ with fewer than κ support points. Its first κ moments vanish by Equation 17. Here $\hat{\mu}_{k+1}(\boldsymbol{v}_{0,j})$ is the k-th moment of $\alpha \mapsto \alpha \cdot \omega_{0,j}(\alpha)$. We apply Lemma 4.3 over $\mathbb{E}'$, since after multiplication by α or β_i the values may lie outside $\mathbb{F}_{\mathsf{p}}$. Hence these differences vanish pointwise. Thus $\omega_{0,0}$ is supported on encodings of valid assignments (Lemma 4.7).

The difference between the two sides of Equation 19 has support of size at most

$$|\operatorname{supp}\omega_{0,0}| + 1 + 2\cdot \sum_{i=n_{\mathbb{x}}+2}^{N} |\operatorname{supp}\omega^{(i)}| \le 2\cdot\kappa - 1\,.$$

Its first $2\cdot\kappa - 1$ moments vanish by Equations 18 and 20, so the identity holds pointwise by Lemma 4.3. We define $U \coloneqq \operatorname{span}_{\mathbb{F}_{\mathsf{p}}}\{\beta_i : n_{\mathbb{x}} + 1 < i \le N\}$. The coset $\gamma_{\mathbb{x}} + U$ consists of the encodings of assignments over $\mathbb{F}_{\mathsf{p}}$ with prefix $(\mathbb{x}, 1)$. For every $i > n_{\mathbb{x}} + 1$, this coset is invariant under translation by β_i. Hence, after summing Equation 19 over the coset,

$$\sum_{\alpha \in \gamma_{\mathbb{x}} + U} \omega_{0,0}(\alpha) = 1\,.$$

So $\omega_{0,0}(\alpha) \neq 0$ for some α in the coset. By Lemma 4.7, α encodes a valid assignment, and its prefix is $(\mathbb{x}, 1)$ since α lies in the coset. Hence $\mathbb{x} \in \mathsf{L}_{\mathbf{V}}$. □

Proof of the time bound in Theorem 4.6. The evaluation table partition and all coefficients of $\boldsymbol{G}_0$ depend only on the ingredients. Only the powers $\gamma_{\mathbb{x}}^k$ in $\boldsymbol{b}_{\mathbb{x}}$ depend on the instance. We compute the coefficients by composing the moment maps with multiplication in $\mathbb{E}$, in the fixed basis. Besides $\sigma \le (N+1)^2$ copies of the evaluation table constraints, the system has at most $(2\cdot N + 3)\cdot\kappa$ equations over $\mathbb{E}$, each with at most $(N+1)^2 + 2\cdot N\cdot\kappa$ terms. Since N and κ are at most the description length of the ingredients, $\boldsymbol{G}_0$ and $\boldsymbol{b}_{\mathbb{x}}$ can be computed in time polynomial in this length and $n_{\mathbb{x}}$. □

4.5 System size and weight gap

We use the moment representation of Construction A.4 in Construction 4.5. The column and time bounds are proved later, together with the choice of a as a function of the verifier size and the gap Γ (Section A.4).

Theorem 4.8. *Let p be a prime, let $\mathbf{V}$ be a verifier with N assignment coordinates, and let $\Gamma, \mathsf{a} \ge 1$ be integers. There are integers D and Q and a preprocessing reduction from $\mathsf{L}_{\mathbf{V}}$, with fixed input $(1^N, 1^\Gamma)$, to (D, Q, Γ)-BSLSP over $\mathbb{F}_{\mathsf{p}}$, whose matrix $\boldsymbol{G}_0$ has $n_{\mathrm{sys}} \coloneqq D\cdot Q$ columns and whose right-hand side for $\mathbb{x}$ is $\boldsymbol{b}_{\mathbb{x}}$. The reduction satisfies:*

- **Completeness.** *For every* $\mathbb{x} \in \mathsf{L}_{\mathbf{V}}$*, there is a binary vector* $\boldsymbol{v} \in \{0,1\}^{n_{\mathrm{sys}}}$ *satisfying* $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\mathbb{x}}$ *with at most one nonzero coordinate per block. In particular,* $\mathrm{wt}(\boldsymbol{v}) \le D$*.*
- **Soundness.** *For every* $\mathbb{x} \in \{0,1\}^{n_{\mathbb{x}}} \setminus \mathsf{L}_{\mathbf{V}}$*, every* $\boldsymbol{v} \in \mathbb{F}_{\mathsf{p}}^{n_{\mathrm{sys}}}$ *satisfying* $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\mathbb{x}}$ *has* $\mathrm{wt}(\boldsymbol{v}) > \Gamma \cdot D$*.*
- **Size.** *The number of columns satisfies*

$$n_{\mathrm{sys}} \le \big(8 \cdot \mathsf{p}^4 \cdot \mathsf{a} \cdot (N+1)\big)^{\mathsf{a}+7} \cdot \Gamma^{\,1+1/\mathsf{a}}\,. \tag{22}$$

  In particular, for a constant $C_{\mathsf{p},\mathsf{a}}$ *depending only on* p, a*,*

$$n_{\mathrm{sys}} \le (N+1)^{C_{\mathsf{p},\mathsf{a}}} \cdot \Gamma^{\,1+1/\mathsf{a}}\,. \tag{23}$$

- **Time.** *The matrix* $\boldsymbol{G}_0$ *and its partition depend only on* $\mathbf{V}, \mathsf{p}, \mathsf{a}, \Gamma$*. Together with* $\boldsymbol{b}_{\mathbb{x}}$ *for each instance, they can be computed in deterministic time* $(N + n_{\mathrm{sys}})^{O_{\mathsf{p}}(1)}$*, with an exponent independent of* a*. For fixed* p, a*, this time is polynomial in* $N + \Gamma$*.*

If $1 \le \Gamma \le 2^N$*, there is a choice* $\mathsf{a} = \mathsf{a}(N, \Gamma)$ *for which, in addition,*

$$n_{\mathrm{sys}} \le \Gamma \cdot (N+1)^{O_{\mathsf{p}}(1)} \cdot \exp\Big(O_{\mathsf{p}}\Big(\sqrt{\log \Gamma \cdot \log N}\Big)\Big)\,, \tag{24}$$

and $\boldsymbol{G}_0$*, its partition, and* $\boldsymbol{b}_{\mathbb{x}}$ *for each instance can be computed in deterministic time* $(N+\Gamma)^{O_{\mathsf{p}}(1)}$*.*

The size–gap tradeoff. For the $\sigma \le (N+1)^2$ evaluation tables of Construction 4.5, the moment representation is that of Construction A.4, with $D_{\mathrm{tab}} = \tau$ and

$$\kappa = \mathsf{d}^{\mathsf{a}}, \qquad m_{\mathrm{tab}} = \tau^{\mathsf{a}+1}, \qquad \mathsf{d} \le \tau = O_{\mathsf{p}}(\mathsf{a} \cdot N \cdot \mathsf{d})\,.$$

The hypotheses of Theorem 4.6 hold when $\mathsf{d} \ge 2$ is the least integer with $\mathsf{d}^{\mathsf{a}} \ge 8 \cdot \sigma \cdot \Gamma$, and the dependence on the gap is then $\Gamma^{1+1/\mathsf{a}}$ in Equation 23. For the fixed-ε factors of Theorem 5.1, a is a sufficiently large constant. For the explicit factor of Theorem 5.4, we choose a as a function of N and Γ to balance the prefactor in Equation 22 against $\Gamma^{1/\mathsf{a}}$ (Section A.4).

5 Codes, lattices, and hardness with preprocessing

5.1 Approximation hardness with preprocessing

Theorem 5.1 (Fixed code and lattice gaps)**.** *Fix the nearest codeword problem over a finite field* $\mathbb{F}$*, or the closest vector problem in* ℓ_p *for a rational* $p \ge 1$ *or* $p = \infty$*. The exponent parameter is*

$$s := \begin{cases} 1 & \text{for nearest codeword,} \\ \min\{p, 2\} & \text{for closest vector.} \end{cases}$$

For every fixed $0 < \varepsilon < 1/s$*, there are deterministic polynomial-time maps*

$$1^{n_{\mathbb{x}}} \longmapsto (\mathcal{S}_{n_{\mathbb{x}}}, R_{n_{\mathbb{x}}}), \qquad \mathbb{x} \in \{0,1\}^{n_{\mathbb{x}}} \longmapsto \boldsymbol{t}_{\mathbb{x}}.$$

Here $\mathcal{S}_{n_{\mathbb{x}}}$ *denotes the code or lattice, and* dist *is Hamming distance or* ℓ_p *distance, respectively. The code block length or lattice dimension is* $M = \mathrm{poly}(n_{\mathbb{x}})$*. For every* $\mathbb{x} \in \{0,1\}^{n_{\mathbb{x}}}$*,*

$$\begin{aligned} \mathbb{x} \in \mathrm{3SAT} \quad &\Longrightarrow \quad \mathrm{dist}(\boldsymbol{t}_{\mathbb{x}}, \mathcal{S}_{n_{\mathbb{x}}}) \le R_{n_{\mathbb{x}}}, \\ \mathbb{x} \notin \mathrm{3SAT} \quad &\Longrightarrow \quad \mathrm{dist}(\boldsymbol{t}_{\mathbb{x}}, \mathcal{S}_{n_{\mathbb{x}}}) > M^{1/s-\varepsilon} \cdot R_{n_{\mathbb{x}}}. \end{aligned}$$

The code is given by a generator matrix, and the lattice by an integer basis. Radii are positive integers, and lattice targets are integer vectors. The fixed input and target have binary description lengths $O(M^2 \cdot \log(M+1))$ and $O(M \cdot \log(M+1))$, respectively.

Proof. We apply Theorem 4.8, followed by Lemma 3.11 for codes and Lemma 3.12 for lattices. Set $\mathsf{p} = \operatorname{char} \mathbb{F}$ for codes and $\mathsf{p} = 2$ for lattices. The intermediate system has $n_{\mathrm{sys}} = D \cdot Q$ columns, and $2 \cdot n_{\mathrm{sys}} \le M < 4 \cdot n_{\mathrm{sys}}$. For a gap $f(M)$, it suffices to choose $\Gamma > 16 \cdot f(M)^s$.

Take $\mathbf{V} = \mathbf{V}_{n_\varkappa}^{\mathrm{3SAT}}$, a verifier for 3SAT constructible from $1^{n_\varkappa}$ in polynomial time, with $N = \mathrm{poly}(n_\varkappa) \ge 2$, and set $\delta \coloneqq 1 - s \cdot \varepsilon \in (0,1)$. We choose integers $\mathsf{a}, K \ge 1$ such that

$$\delta \cdot (1 + 1/\mathsf{a}) < 1, \qquad K \cdot \big(1 - \delta \cdot (1 + 1/\mathsf{a})\big) > \delta \cdot C_{\mathsf{p},\mathsf{a}} + 4,$$

and set $\Gamma \coloneqq (N+1)^K$, so that the fixed input $(1^N, 1^\Gamma)$ of Theorem 4.8 is computable from $1^{n_\varkappa}$ in polynomial time. By Equation 23,

$$\frac{16 \cdot M^\delta}{\Gamma} \le 16 \cdot 4^\delta \cdot (N+1)^{\delta \cdot C_{\mathsf{p},\mathsf{a}} - K \cdot (1 - \delta \cdot (1 + 1/\mathsf{a}))} \le \frac{64}{(N+1)^4} < 1. \tag{25}$$

Since $(M^{1/s-\varepsilon})^s = M^\delta$, the required gap holds. The fixed input and the radius depend only on $\mathbf{V}_{n_\varkappa}^{\mathrm{3SAT}}$ and the construction parameters, while the target depends on $\varkappa$. Both maps are computable in polynomial time. For lattices, the fixed input and target have the claimed description lengths (Section B.3). □

Lemma 5.2 (Testing search outputs)**.** *For the code and lattice instances constructed in Lemmas 3.11 and 3.12, let $s = 1$ for codes and $s = \min\{p, 2\}$ for lattices. If $f(M) \ge 1$ and $\Gamma > 16 \cdot f(M)^s$, there is a test with running time polynomial in the number of bits needed to represent the instance and the returned vector, such that*

$$\begin{aligned} \textit{YES instance} \quad &\Longrightarrow \quad \textit{every } f(M)\textit{-approximate solution is accepted,} \\ \textit{NO instance} \quad &\Longrightarrow \quad \textit{every returned vector is rejected.} \end{aligned}$$

Proof. For target $\boldsymbol{t}$ and radius R, membership in the code or lattice is checked first, and nonmembers are rejected. For a codeword $\boldsymbol{v}$, we define $w \coloneqq \mathrm{wt}(\boldsymbol{t} - \boldsymbol{v})$, and for a lattice vector we compute w as in Lemma 3.12. We accept a returned vector if $w \le \Gamma \cdot D$ and reject it otherwise. For every $f(M)$-approximate solution $\boldsymbol{v}$ to a YES instance, by Lemmas 3.11 and 3.12,

$$\begin{aligned} &w \le f(M) \cdot D < \Gamma \cdot D && \text{for codes,} \\ &w \le 16 \cdot D \cdot \left(\frac{\|\boldsymbol{t} - \boldsymbol{v}\|_p}{R} \right)^s \le 16 \cdot D \cdot f(M)^s < \Gamma \cdot D && \text{for lattices.} \end{aligned}$$

For NO instances, $w > \Gamma \cdot D$ by the same lemmas. Membership and both acceptance conditions can be checked using exact field or integer arithmetic, in time polynomial in the number of bits needed to represent the instance and the returned vector. □

Proof of Theorem 1.1. First suppose that p is rational or infinite in the lattice case. For the code or lattice $\mathcal{S}_{n_\varkappa}$ of Theorem 5.1, the fixed input matrix $S_{n_\varkappa}$ is its generator matrix or basis. We hardwire the construction parameters, $S_{n_\varkappa}$, the radius $R_{n_\varkappa}$, and $\mathcal{A}_0(S_{n_\varkappa})$. Given $\varkappa$, we compute $\boldsymbol{t}_\varkappa$ and run $\mathcal{A}_1$, with the radius included for the gap problem. For the gap problem, the output is the answer

of $\mathcal{A}_1$. For search, we test the returned vector as in Lemma 5.2. 3SAT is therefore decidable by circuits of size

$$\big(T(n_{\varkappa}) + M\big)^{O(1)}, \tag{26}$$

where T is the target construction time. Here $T, M = \mathrm{poly}(n_{\varkappa})$, so NP $\subseteq$ P/poly. If preprocessing takes polynomial time, we compute $S_{n_{\varkappa}}$, $\mathcal{A}_0(S_{n_{\varkappa}})$, and $R_{n_{\varkappa}}$ in polynomial time as well. Thus 3SAT $\in$ P.

For any fixed real $1 \le p < \infty$, choose a rational $1 \le \widehat{p} \le p$ and let $\eta \coloneqq 1/\widehat{p} - 1/p$. In dimension M,

$$\|\boldsymbol{v}\|_p \le \|\boldsymbol{v}\|_{\widehat{p}} \le M^{\eta} \cdot \|\boldsymbol{v}\|_p .$$

For $p \le 2$, take $\widehat{p} = 1$ and $\varepsilon' = \varepsilon$. For $p > 2$, choose a rational $2 < \widehat{p} < p$ with $\eta < \varepsilon/2$ and take $\varepsilon' = \varepsilon/2$. We apply the preceding proof in $\ell_{\widehat{p}}$ with ε' in place of ε. The same integer radius is valid for YES instances in ℓ_p, and the NO gap exponent satisfies

$$1/\min\{\widehat{p}, 2\} - \varepsilon' - \eta \ge 1/\min\{p, 2\} - \varepsilon.$$

For search, let $f(M) = M^{1/\min\{p,2\} - \varepsilon}$. An $f(M)$-approximate solution in ℓ_p is an $M^{\eta} \cdot f(M)$-approximate solution in $\ell_{\widehat{p}}$, so we apply the test of Lemma 5.2 in $\ell_{\widehat{p}}$. □

Remark 5.3 (Randomized online algorithms)**.** The conclusion of Theorem 1.1 under NP $\not\subseteq$ P/poly also holds for randomized online algorithms that err with probability at most 1/3 on every instance, where for search an error is a returned vector that is not an f-approximate solution. For fixed $n_{\varkappa}$, we run the online algorithm $O(n_{\varkappa})$ times with independent random strings. For the gap problem, we take the majority answer. For search, we accept if some returned vector passes the test of Lemma 5.2, and reject otherwise. Every returned vector is rejected on NO instances. Each procedure decides 3SAT with error probability below $2^{-n_{\varkappa}-1}$ on every instance, so by a union bound over the $2^{n_{\varkappa}}$ targets $\boldsymbol{t}_{\varkappa}$, some random string is correct for all of them. We hardwire this string with the preprocessing output. Then 3SAT is decidable by circuits of size polynomial in $n_{\varkappa}$, so NP $\subseteq$ P/poly. With polynomial-time preprocessing, we instead conclude NP $\subseteq$ BPP.

5.2 Hardness within $M^{1/s - o(1)}$

Nonuniform ETH is the assumption that no family of circuits of size $2^{o(v)}$ decides 3SAT on v variables.

Theorem 5.4. *Fix a problem and s as in Theorem 5.1. For every fixed integer $c \ge 1$, there is a constant $C_c > 0$ such that, for*

$$f(M) \coloneqq M^{1/s} \cdot \exp\left(-C_c \cdot \sqrt{\log M \cdot \log\log M}\right),$$

if nonuniform ETH holds, neither the f-gap problem nor f-approximate search has a preprocessing algorithm with output length and online time $M^{O((\log M)^c)}$ for all sufficiently large M. This holds for fixed inputs of binary length $O(M^2 \cdot \log(M+1))$ and targets of binary length $O(M \cdot \log(M+1))$. Here we replace the definition's polynomial bounds by the stated bounds; preprocessing time remains unrestricted. Without preprocessing, under the exponential time hypothesis of [IP01], no deterministic algorithm solves the f-gap problem or f-approximate search in time $M^{O((\log M)^c)}$ for all sufficiently large M.

Proof. We encode v-variable 3CNF formulas as strings of a common length polynomial in v, using the $O(v^3)$ possible clauses. The verifier has size $v \le N = v^{O(1)}$ and depends only on v. We set

$$\Gamma := 2^{\lceil v^{1/(c+2)} \rceil} \le 2^N.$$

We choose a as in the optimized bound of Theorem 4.8 and apply that theorem and the conversions used above, with the same field characteristic. By Equation 24 and $\Gamma < Q \le M < 4 \cdot n_{\mathrm{sys}}$,

$$\log M = \log \Gamma + O\left(\sqrt{\log \Gamma \cdot \log v} + \log v\right) = \Theta(v^{1/(c+2)}).$$

Hence $\log v = \Theta(\log \log M)$ and, for a constant $C_c' > 0$,

$$M \le \Gamma \cdot \exp\left(C_c' \cdot \sqrt{\log M \cdot \log \log M}\right).$$

We choose $C_c > C_c'$. For all sufficiently large v, $f(M) \ge 1$ and

$$16 \cdot f(M)^s \le 16 \cdot \Gamma \cdot \exp\left((C_c' - s \cdot C_c) \cdot \sqrt{\log M \cdot \log \log M}\right) < \Gamma.$$

The target can be constructed in time $T = \mathrm{poly}(N+\Gamma) = 2^{O(v^{1/(c+2)})}$. We hardwire the construction parameters, fixed matrix, radius, and preprocessing output, and apply Lemma 5.2 for search. Then 3SAT is decidable by circuits of size

$$M^{O((\log M)^c)} = 2^{O(v^{(c+1)/(c+2)})},$$

contradicting nonuniform ETH. For lattices, the fixed input and target have the claimed description lengths (Section B.3). Without preprocessing, the fixed matrix and radius are also constructed in deterministic time $\mathrm{poly}(N + \Gamma) = 2^{O(v^{1/(c+2)})}$, so an algorithm with running time $M^{O((\log M)^c)} = 2^{O(v^{(c+1)/(c+2)})}$ decides 3SAT in time $2^{o(v)}$, contradicting the exponential time hypothesis. □

We prove the following corollary of Theorem 5.1 in Appendix C.

Corollary 5.5. *Assume* NP $\not\subseteq$ P/poly*, and fix a problem and s as in Theorem 5.1. There is a computable nondecreasing $f\colon \mathbb{N} \to \mathbb{N}$ with $f(M) = M^{1/s - o(1)}$ such that neither the f-gap problem nor f-approximate search has a preprocessing algorithm with output length and online time* $\mathrm{poly}(M)$.

6 A reduction from query-oblivious PCPs

In this section, we reduce languages with query-oblivious PCPs to nearest codeword with preprocessing, without the smoothness assumption of [KPV14]. Using the IOPs of Arnon, Chiesa, and Yogev [ACY23] and Minzer and Zheng [MZ25], we obtain hardness within subpolynomial factors larger than those previously known.

6.1 Query-oblivious verifiers

Definition 6.1 (Query-oblivious PCP, [BGHSV06, Section 2.1], [CMS19, Section 1.1])**.** *Let $\mathcal{R}$ be a relation, and let $\mathsf{L}(\mathcal{R}) := \{\mathbb{x} : \exists \mathbb{w}\ (\mathbb{x}, \mathbb{w}) \in \mathcal{R}\}$. A* **PCP** *for $\mathcal{R}$ on instances of length $n_{\mathbb{x}}$ has the following components.*

- **Parameters.** *An alphabet Σ, a proof length ℓ, a number R of random bits, and a number $\mathsf{q} \geq 1$ of queries. For a proof $\pi \in \Sigma^{\ell}$ and a tuple $I = (i_1, \ldots, i_{\mathsf{q}}) \in [\ell]^{\mathsf{q}}$, where $[\ell] \coloneqq \{1, \ldots, \ell\}$, we write $\pi|_I \coloneqq (\pi(i_1), \ldots, \pi(i_{\mathsf{q}}))$.*

- **Verifier.** *Given an instance $\mathbb{x} \in \{0,1\}^{n_{\mathbb{x}}}$ and a random string $r \in \{0,1\}^{\mathsf{R}}$, the verifier computes a tuple $I_{\mathbb{x},r} \in [\ell]^{\mathsf{q}}$ of proof locations and an* **acceptance predicate** *$V_{\mathbb{x},r} \colon \Sigma^{\mathsf{q}} \to \{0,1\}$. It accepts a proof π if $V_{\mathbb{x},r}(\pi|_{I_{\mathbb{x},r}}) = 1$.*

The PCP has **perfect completeness** *and* **soundness error** s *if the following hold.*

- **Completeness.** *For every $\mathbb{x} \in \mathsf{L}(\mathcal{R}) \cap \{0,1\}^{n_{\mathbb{x}}}$, there is a proof that the verifier accepts for every random string.*

- **Soundness.** *For every $\mathbb{x} \in \{0,1\}^{n_{\mathbb{x}}} \setminus \mathsf{L}(\mathcal{R})$ and every proof π, the verifier accepts π with probability at most s over the random string.*

The PCP is **query-oblivious** *(*input oblivious *in [CMS19]) if the tuple $I_{\mathbb{x},r}$ does not depend on $\mathbb{x}$. Write $I_r = (u_{r,1}, \ldots, u_{r,\mathsf{q}})$ for this tuple. For the reduction, we additionally specify the following data and algorithms:*

- **Predicates.** *For each r, there is an indexed family $\mathcal{D}_r$ of labeled predicates $\Sigma^{\mathsf{q}} \to \{0,1\}$ that does not depend on $\mathbb{x}$ and contains $V_{\mathbb{x},r}$ for every $\mathbb{x}$. Different labels may represent the same Boolean function. Equality of predicates below means equality of their labels. The* **maximum number of predicates per random string** *is $\mathsf{k} \coloneqq \max_r |\mathcal{D}_r|$. No polynomial bound on k is assumed in this definition. In Theorem 6.4, we bound the running time in terms of this parameter. For the PCP used in Theorem 6.5, $\mathsf{k} = \mathrm{poly}(N)$.*

- **Accepting view.** *An accepting view for $r \in \{0,1\}^{\mathsf{R}}$ is a pair $(V, \boldsymbol{a}) \in \mathcal{D}_r \times \Sigma^{\mathsf{q}}$ such that $V(\boldsymbol{a}) = 1$ and $\boldsymbol{a} = \pi|_{I_r}$ for some proof π, that is, $a_j = a_{j'}$ whenever $u_{r,j} = u_{r,j'}$.*

  We denote the set of accepting views for r by $\mathcal{V}_r$, and their total number by $\mathsf{m} \coloneqq \sum_r |\mathcal{V}_r|$.

- **Running time.** *We use vt as a common bound for two separate algorithms. Given $\mathbb{x}$ and r, the verifier computes I_r, the label and circuit of $V_{\mathbb{x},r}$, and its decision on the queried answers in time vt. Given $n_{\mathbb{x}}$ and r, the predicate enumerator computes I_r and lists the labels and circuits of all predicates in $\mathcal{D}_r$ in time vt.*

6.2 The PCP reduction

Construction 6.2 (Accepting-view code)**.** The ingredients are:
- A query-oblivious PCP for a relation $\mathcal{R}$ on instances of length $n_{\mathbb{x}}$, with alphabet Σ, R random bits, q queries, query tuples $I_r = (u_{r,1}, \ldots, u_{r,\mathsf{q}})$, predicate families $\mathcal{D}_r$, and m accepting views.
- A finite field $\mathbb{F}$ and an $\mathbb{F}$-linear encoding map $\iota \colon \mathbb{F}^{\mathsf{m}} \to \mathbb{F}^{B}$.
- An integer $\Gamma \geq 2$.

We construct an instance of Γ-gap NCPP over $\mathbb{F}$ as follows.

1. ***Variables.*** For each $r \in \{0,1\}^{\mathsf{R}}$ and each $(V, \boldsymbol{a}) \in \mathcal{V}_r$, there is a variable $v_{r,V,\boldsymbol{a}}$. The vector of all variables is $\boldsymbol{v} \in \mathbb{F}^{\mathsf{m}}$.

2. ***Constraints.*** Given $\mathbb{x} \in \{0,1\}^{n_{\mathbb{x}}}$, we compute $V_{\mathbb{x},r}$ for every r. For every r and every $V \in \mathcal{D}_r$, we impose

$$\sum_{\boldsymbol{a}:\,(V,\boldsymbol{a})\in\mathcal{V}_r} v_{r,V,\boldsymbol{a}} = \begin{cases} 1, & V = V_{\mathbb{x},r}\,, \\ 0, & V \neq V_{\mathbb{x},r}\,. \end{cases} \tag{27}$$

   In matrix notation, this system is $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\mathbb{x}}$, with one row for every pair (r, V). The matrix $\boldsymbol{G}_0$ does not depend on $\mathbb{x}$.

3. ***Restrictions.*** Let $\mathcal{U}$ be the set of queried locations $u_{r,j}$. For $u \in \mathcal{U}$, let d_u be the number of pairs $(r, j) \in \{0,1\}^{\mathsf{R}} \times [\mathsf{q}]$ with $u_{r,j} = u$. For $(u, a) \in \mathcal{U} \times \Sigma$, the restriction vector $\boldsymbol{v}^{u,a} \in \mathbb{F}^{\mathsf{m}}$ is defined by

$$(\boldsymbol{v}^{u,a})_{r,V,\boldsymbol{a}} \coloneqq \begin{cases} v_{r,V,\boldsymbol{a}}, & (u_{r,j}, a_j) = (u, a) \text{ for some } j \in [\mathsf{q}]\,, \\ 0, & \text{otherwise}\,. \end{cases}$$

4. ***Code.*** Let $Y \coloneqq B \cdot 2^{\mathsf{R}} \cdot \mathsf{q}$, $M_0 \coloneqq |\Sigma| \cdot Y$, $H \coloneqq \Gamma \cdot Y + 1$, and $M \coloneqq M_0 + H$. For each $(u, a) \in \mathcal{U} \times \Sigma$, we stack d_u copies of the matrix of $\boldsymbol{v} \mapsto \iota(\boldsymbol{v}^{u,a})$. The resulting matrix is $\boldsymbol{G}_1 \in \mathbb{F}^{M_0 \times \mathsf{m}}$. The code is

$$C \coloneqq (\boldsymbol{G}_1 \cdot \ker \boldsymbol{G}_0) \times \{\boldsymbol{0}^H\} \subseteq \mathbb{F}^M\,,$$

   and the radius is Y.

5. ***Target.*** Given $\mathbb{x}$, we solve $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\mathbb{x}}$ by Gaussian elimination. If there is a solution $\boldsymbol{v}_{\mathbb{x}}$, the target is $\boldsymbol{y}_{\mathbb{x}} \coloneqq (\boldsymbol{G}_1 \cdot \boldsymbol{v}_{\mathbb{x}}, \boldsymbol{0}^H)$. Otherwise, it is $\boldsymbol{y}_{\mathbb{x}} \coloneqq (\boldsymbol{0}^{M_0}, \boldsymbol{1}^H)$.

Remark 6.3 (Comparison with the Label Cover reductions)**.** The reductions of [AKKV05; KPV14] pass through Label Cover with preprocessing (Appendix D). There, a variable corresponds to a label of a vertex, class sums as in (27) are imposed at the left vertices, and projection equations equate each right variable with the sum of the left variables whose labels project to it. Such a sum can vanish although its summands are nonzero, and smoothness is used to bound the fraction of edges on which two labels with nonzero coefficients collide. Here we use one variable per accepting view, so a tuple accepted by two predicates has two variables, and impose no projection equations. The restriction vector $\boldsymbol{v}^{u,a}$ consists of the coefficients of all views with answer a at u, and its injective encoding is nonzero whenever $\boldsymbol{v}^{u,a} \neq \boldsymbol{0}$, even if these coefficients sum to zero. Every view with a nonzero coefficient therefore has all its answers in the lists $\{a : \boldsymbol{v}^{u,a} \neq \boldsymbol{0}\}$, without a smoothness assumption.

Theorem 6.4. *For Construction 6.2, suppose that the PCP has instance-independent query locations and predicate families enumerable as in Definition 6.1, perfect completeness, and soundness error $0 < \mathsf{s} < 1$, that ι is injective and the code $\iota(\mathbb{F}^{\mathsf{m}})$ has relative distance δ, and that*

$$\Gamma < \delta \cdot \mathsf{s}^{-1/\mathsf{q}}\,. \tag{28}$$

Then the following hold.

- **Completeness.** *For every $\mathbb{x} \in \mathsf{L}(\mathcal{R}) \cap \{0,1\}^{n_{\mathbb{x}}}$, $\Delta(\boldsymbol{y}_{\mathbb{x}}, C) \leq Y$.*
- **Soundness.** *For every $\mathbb{x} \in \{0,1\}^{n_{\mathbb{x}}} \setminus \mathsf{L}(\mathcal{R})$, $\Delta(\boldsymbol{y}_{\mathbb{x}}, C) > \Gamma \cdot Y$.*

- **Size and time.** *The code has block length* $M = (|\Sigma|+\Gamma)\cdot Y+1$ *and depends only on the PCP, the encoding map, and* Γ*. The construction can be carried out in time polynomial in* $n_{\mathbb{x}}$*,* $2^{\mathsf{R}}\cdot\mathsf{k}\cdot|\Sigma|^{\mathsf{q}}$*,* B*,* Γ*, and the common running-time bound* vt *for verification and predicate enumeration, plus the time to construct the encoding map.*

In particular, if these parameters are polynomial in $n_{\mathbb{x}}$ *and the encoding map can be constructed in polynomial time, the construction is a preprocessing reduction from* $\mathsf{L}(\mathcal{R})$*, with fixed input* $1^{n_{\mathbb{x}}}$*, to* Γ*-gap NCPP over* $\mathbb{F}$*.*

Proof. For every random string, the verifier makes q queries, so $\sum_{u\in\mathcal{U}} d_u = 2^{\mathsf{R}}\cdot\mathsf{q}$. Hence $\boldsymbol{G}_1$ has $B\cdot|\Sigma|\cdot 2^{\mathsf{R}}\cdot\mathsf{q} = M_0$ rows, and for every $\boldsymbol{v}\in\mathbb{F}^{\mathsf{m}}$,

$$W(\boldsymbol{v}) \coloneqq \mathrm{wt}(\boldsymbol{G}_1\cdot\boldsymbol{v}) = \sum_{u\in\mathcal{U}} d_u\cdot\sum_{a\in\Sigma}\mathrm{wt}(\iota(\boldsymbol{v}^{u,a}))\,. \tag{29}$$

If $\boldsymbol{G}_0\cdot\boldsymbol{v} = \boldsymbol{b}_{\mathbb{x}}$ has a solution, we delete the common zero padding from the code and target and apply Lemma 3.8:

$$\Delta(\boldsymbol{y}_{\mathbb{x}}, C) = \min_{\boldsymbol{G}_0\cdot\boldsymbol{v}=\boldsymbol{b}_{\mathbb{x}}} W(\boldsymbol{v})\,. \tag{30}$$

- ***Completeness.*** Let $\mathbb{x}\in\mathsf{L}(\mathcal{R})$, and let π be a proof that the verifier accepts for every random string. Then $(V_{\mathbb{x},r}, \pi|_{I_r})$ is an accepting view for every r, since $\pi(u_{r,j}) = \pi(u_{r,j'})$ whenever $u_{r,j} = u_{r,j'}$. Let the *honest solution* $\boldsymbol{v}$ be the indicator vector of these views.
  - *Constraints.* For each r, the only nonzero variable of $\boldsymbol{v}$ with index r is $v_{r,V_{\mathbb{x},r},\pi|_{I_r}} = 1$. Hence $\boldsymbol{v}$ satisfies Equation 27.
  - *Restrictions.* Every view in the support of $\boldsymbol{v}$ has answer $\pi(u)$ at each of its queried locations u. Hence $\boldsymbol{v}^{u,a} = \boldsymbol{0}$ for $a\neq\pi(u)$, and the encoding of $\boldsymbol{v}^{u,\pi(u)}$ has weight at most B.
  - *Distance.* By Equations 29 and 30,
$$\Delta(\boldsymbol{y}_{\mathbb{x}}, C) \le W(\boldsymbol{v}) \le B\cdot\sum_{u\in\mathcal{U}} d_u = Y\,.$$
- ***Soundness.*** Let $\mathbb{x}\notin\mathsf{L}(\mathcal{R})$. If $\boldsymbol{G}_0\cdot\boldsymbol{v} = \boldsymbol{b}_{\mathbb{x}}$ has no solution, then $\boldsymbol{y}_{\mathbb{x}} = (\boldsymbol{0}^{M_0}, \boldsymbol{1}^H)$. Every codeword is zero on the last H coordinates, so $\Delta(\boldsymbol{y}_{\mathbb{x}}, C) = H > \Gamma\cdot Y$. Otherwise, by Equation 30, it suffices to show that $W(\boldsymbol{v}) > \Gamma\cdot Y$ for every solution $\boldsymbol{v}$. Fix a solution $\boldsymbol{v}$.
  - *Lists.* For $u\in\mathcal{U}$, let $\Lambda_u \coloneqq \{a\in\Sigma : \boldsymbol{v}^{u,a}\neq\boldsymbol{0}\}$. The average list size over all queries is
$$\overline{\Lambda} \coloneqq \frac{1}{2^{\mathsf{R}}\cdot\mathsf{q}}\cdot\sum_{u\in\mathcal{U}} d_u\cdot|\Lambda_u|\,.$$
    By Equation 27, for each r the variables $v_{r,V_{\mathbb{x},r},\boldsymbol{a}}$ sum to 1. Hence there is an accepting view $(V_{\mathbb{x},r}, \boldsymbol{a}^{(r)})$ with $v_{r,V_{\mathbb{x},r},\boldsymbol{a}^{(r)}}\neq 0$, and $a_j^{(r)}\in\Lambda_{u_{r,j}}$ for every j. In particular, every list is nonempty.
  - *Sampling a proof.* We sample a proof π whose entries $\pi(u)$, $u\in\mathcal{U}$, are independent and uniformly distributed on Λ_u. For a fixed random string r, the sampled proof is accepted if $\pi|_{I_r} = \boldsymbol{a}^{(r)}$, which happens with probability at least $\prod_{j=1}^{\mathsf{q}}|\Lambda_{u_{r,j}}|^{-1}$, since each proof entry is sampled only once.

By soundness, every fixed proof is accepted with probability at most s, so

$$\mathsf{s} \geq \frac{1}{2^{\mathsf{R}}} \cdot \sum_{r \in \{0,1\}^{\mathsf{R}}} \prod_{j=1}^{\mathsf{q}} |\Lambda_{u_{r,j}}|^{-1} \geq \exp\left(-\frac{1}{2^{\mathsf{R}}} \cdot \sum_{u \in \mathcal{U}} d_u \cdot \ln|\Lambda_u|\right) \geq \overline{\Lambda}^{-\mathsf{q}}, \tag{31}$$

where the second inequality is Jensen's inequality for the exponential, and the third is Jensen's inequality for the logarithm with weights $d_u/(2^{\mathsf{R}} \cdot \mathsf{q})$.

– *Weight.* For $a \in \Lambda_u$, the vector $\boldsymbol{v}^{u,a}$ is nonzero, so its encoding has weight at least $\delta \cdot B$. By Equations 28, 29 and 31,

$$W(\boldsymbol{v}) \geq \delta \cdot B \cdot \sum_{u \in \mathcal{U}} d_u \cdot |\Lambda_u| = \delta \cdot Y \cdot \overline{\Lambda} \geq \delta \cdot Y \cdot \mathsf{s}^{-1/\mathsf{q}} > \Gamma \cdot Y\,.$$

- ***Size and time.*** The block length is $M = M_0 + H = (|\Sigma| + \Gamma) \cdot Y + 1$. The code is determined by $\boldsymbol{G}_0$, $\boldsymbol{G}_1$, and H, so it depends only on the PCP, the encoding map, and Γ. Since $\mathcal{V}_r \subseteq \mathcal{D}_r \times \Sigma^{\mathsf{q}}$,

$$\mathsf{m} \leq 2^{\mathsf{R}} \cdot \mathsf{k} \cdot |\Sigma|^{\mathsf{q}}\,.$$

To list $\mathcal{V}_r$, we compute I_r and circuits for the labeled predicates in $\mathcal{D}_r$ and evaluate every circuit on every answer tuple in Σ^{q}. We list only accepting tuples whose entries agree at repeated query locations. Over all r, this takes time $O(2^{\mathsf{R}} \cdot \mathsf{vt} \cdot |\Sigma|^{\mathsf{q}})$. We then form $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$, which have m columns and at most $2^{\mathsf{R}} \cdot \mathsf{k}$ and M_0 rows, respectively, and compute a generator matrix of C by Gaussian elimination. Given $\mathbb{x}$, we compute $V_{\mathbb{x},r}$ for every r, identify its label in $\mathcal{D}_r$, and solve $\boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}_{\mathbb{x}}$. Only queried locations are listed, so the running time does not depend on the proof length ℓ. □

6.3 Hardness from known PCPs

We use the constructions for rank-one constraint systems (R1CS) of Arnon, Chiesa, and Yogev [ACY23] and Minzer and Zheng [MZ25, Section 10.2] for a fixed Boolean verifier $\mathbf{V}$. The IOP verifier computes the query sets from its random tape and checks any supplied query-set descriptions against them, so that the unrolled PCP is query-oblivious. For fixed randomness, we enumerate the acceptance circuits by substituting all tuples of values for the constantly many instance-dependent field elements over a field of size $\mathrm{poly}(N)$, using the tuples as labels. By unrolling these IOPs as in [ABCY22, Lemma 4.6], we obtain the following:

Theorem 6.5 (Query-oblivious PCPs with enumerable predicate families)**.** *For every verifier $\mathbf{V}$ with N assignment coordinates, there exists a query-oblivious PCP for its accepting relation $\mathcal{R}_{\mathbf{V}} := \{(\mathbb{x}, \mathbb{w}) : \mathbf{V}(\mathbb{x}, \mathbb{w}) = 1\}$, with:*

- *perfect completeness and soundness error $1/N$,*
- *$O(\log N \cdot \log\log N)$ random bits,*
- *$O(\log\log N)$ queries over an alphabet of size $\mathrm{poly}(N)$,*
- *for each random string r, a query tuple I_r and a labeled family $\mathcal{D}_r$ of at most $\mathrm{poly}(N)$ predicates $\Sigma^{\mathsf{q}} \to \{0,1\}$, both determined by $\mathbf{V}$ and r, with $V_{\mathbb{x},r} \in \mathcal{D}_r$ for every instance $\mathbb{x}$.*

The PCP verifier and predicate enumerator can be constructed from $\mathbf{V}$ in time polynomial in N. Given r, the enumerator computes I_r and lists the labels and circuits in $\mathcal{D}_r$ in time polynomial in N. Given $\mathbb{x}$ and r, the verifier computes the label and circuit of $V_{\mathbb{x},r}$ and evaluates it on the queried answers in time polynomial in N.

The PCP of Theorem 6.5 yields the following corollary.

Corollary 6.6 (Hardness from known PCPs)**.** *Unless* NP $\subseteq \bigcup_{k>0} \mathrm{SIZE}(n^{k\cdot\log\log n})$*, nearest codeword with preprocessing over any fixed finite field and closest vector with preprocessing in* ℓ_p*, for every fixed rational* $p \geq 1$*, are hard to approximate within a factor of* $M^{c/(\log\log M)^2}$ *and* $M^{c/(p\cdot(\log\log M)^2)}$*, respectively, for some constant* $c > 0$*. Here* M *is the code block length or lattice dimension.*

Proof sketch. We apply Theorem 6.4 with a linear code of constant rate and constant relative distance [Jus72; GRS25], followed by Lemma 3.9 for lattices and the advice conversion in the proof of Theorem 1.1. For search, we check membership and test $\Delta(\boldsymbol{y}_{\mathbb{Z}}, \boldsymbol{c}) \leq \Gamma \cdot Y$, where for a lattice output the codeword $\boldsymbol{c}$ is given by the solution conversion of Lemma 3.9. For gap CVPP, we use the integer radius $R = \lceil Y^{1/p} \rceil \leq 2 \cdot Y^{1/p}$, and the constant c absorbs the resulting factor of 2. □

A Explicit moment representations

We construct the moment representation for Construction 4.5 and prove the size and time bounds of Theorem 4.8. The construction and the rank and recurrence proofs are adapted from [Son26, Sections 3–5].

A.1 Constructing the moment representation

We write moment orders as exponent vectors with small coordinates. The required orders extend beyond $\mathsf{d}^{\mathsf{a}} - 1$, so we allow larger digits than in the ordinary base-d expansion.

Definition A.1 (cf. [Son26, Definition 3.1])**.** *Let* $\mathsf{a} \geq 1$ *and* $\mathsf{d} \geq 2$ *be integers. For* $\boldsymbol{\eta} = (\eta_h)_{h=0}^{\mathsf{a}-1} \in \mathbb{Z}_{\geq 0}^{\mathsf{a}}$ *and a tuple* $\boldsymbol{\lambda} = (\lambda_0, \ldots, \lambda_{\mathsf{a}-1})$ *of field elements, we define*

$$\mathrm{val}_{\mathsf{d}}(\boldsymbol{\eta}) \coloneqq \sum_{h=0}^{\mathsf{a}-1} \eta_h \cdot \mathsf{d}^h\,, \qquad \boldsymbol{\lambda}^{\boldsymbol{\eta}} \coloneqq \prod_{h=0}^{\mathsf{a}-1} \lambda_h^{\eta_h}\,,$$

with the convention $0^0 \coloneqq 1$*. For an integer* $k \geq 0$*, let* t *and* b *be the integers with* $k = t \cdot (\mathsf{d}^{\mathsf{a}} - 1) + b$ *and* $0 \leq b < \mathsf{d}^{\mathsf{a}} - 1$*, and let* $b_0, \ldots, b_{\mathsf{a}-1}$ *be the base-*d *digits of* b*. The generalized digit vector is*

$$\mathrm{digits}_{\mathsf{d},\mathsf{a}}(k) \coloneqq \big(t \cdot (\mathsf{d} - 1) + b_h\big)_{h=0}^{\mathsf{a}-1}\,.$$

Claim A.2. *For every integer* $k \geq 0$*,* $\mathrm{val}_{\mathsf{d}}(\mathrm{digits}_{\mathsf{d},\mathsf{a}}(k)) = k$*. Moreover, for every integer* $J \geq 1$*,*

$$0 \leq k \leq J \cdot (\mathsf{d}^{\mathsf{a}} - 1) \quad \Longrightarrow \quad \mathrm{digits}_{\mathsf{d},\mathsf{a}}(k) \in \{0, \ldots, J \cdot (\mathsf{d} - 1)\}^{\mathsf{a}}\,.$$

Proof. Since $\sum_{h=0}^{\mathsf{a}-1} (\mathsf{d} - 1) \cdot \mathsf{d}^h = \mathsf{d}^{\mathsf{a}} - 1$,

$$\mathrm{val}_{\mathsf{d}}(\mathrm{digits}_{\mathsf{d},\mathsf{a}}(k)) = t \cdot (\mathsf{d}^{\mathsf{a}} - 1) + \sum_{h=0}^{\mathsf{a}-1} b_h \cdot \mathsf{d}^h = t \cdot (\mathsf{d}^{\mathsf{a}} - 1) + b = k\,.$$

If $k \leq J \cdot (\mathsf{d}^{\mathsf{a}} - 1)$, then either $t \leq J - 1$, or $t = J$ and $b = 0$. In the first case, every entry of $\mathrm{digits}_{\mathsf{d},\mathsf{a}}(k)$ is at most $(J-1) \cdot (\mathsf{d} - 1) + \mathsf{d} - 1 = J \cdot (\mathsf{d} - 1)$. In the second case, every entry equals $J \cdot (\mathsf{d} - 1)$. □

Lemma A.3 (cf. [LN97, Chapter 2])**.** *For a fixed prime* p*, an integer* $e \geq 1$*, and a power* c *of* p*, let* $\tau \coloneqq \mathsf{p}^e$*. In time polynomial in* $\tau + c$*, one can construct*

1. *an irreducible polynomial* f *of degree* e *over* $\mathbb{F}_\mathsf{p}$*, together with the field* $\mathbb{K} \coloneqq \mathbb{F}_\mathsf{p}[X]/(f)$ *and its* $\mathbb{F}_\mathsf{p}$*-basis* $1, X, \ldots, X^{e-1}$*, and*
2. *an extension* $\mathbb{E} \coloneqq \mathbb{K}(\zeta)$ *of degree* c*, given by a* $\mathbb{K}$*-basis in which each arithmetic operation takes time polynomial in* $c \cdot e$*, together with the change of basis to the power basis* $1, \zeta, \ldots, \zeta^{c-1}$*.*

Proof. For the first part, it suffices to test the τ monic polynomials of degree e over $\mathbb{F}_\mathsf{p}$ for irreducibility. A candidate f is irreducible if and only if

$$\gcd\left(f(X), X^{\mathsf{p}^j} - X\right) = 1 \quad (1 \leq j \leq \lfloor e/2 \rfloor) \ .$$

Any reducible candidate has an irreducible factor of some degree $j \leq e/2$, and this factor divides $X^{\mathsf{p}^j} - X$. Conversely, the irreducible factors of $X^{\mathsf{p}^j} - X$ have degree dividing $j < e$, so for an irreducible candidate all the gcds equal one. The tests can be carried out in polynomial time by repeated p-th powering modulo f.

For the second part, if $c = 1$, we take $\mathbb{E} \coloneqq \mathbb{K}$ and $\zeta \coloneqq 0$, with power basis $\{1\}$. Otherwise, $\mathbb{E}$ is a tower of Artin–Schreier extensions, each constructed by adjoining a root of an irreducible polynomial of the form $X^\mathsf{p} - X - \theta$. The element θ is chosen using the absolute trace. For a field $\mathbb{F}$ of size p^b, the absolute trace is

$$\mathrm{Tr}_{\mathbb{F}/\mathbb{F}_\mathsf{p}}(x) \coloneqq \sum_{j=0}^{b-1} x^{\mathsf{p}^j} \ .$$

This map is $\mathbb{F}_\mathsf{p}$-linear with values in $\mathbb{F}_\mathsf{p}$, and it is nonzero, since its defining polynomial has degree $\mathsf{p}^{b-1} < |\mathbb{F}|$ and so cannot vanish on all of $\mathbb{F}$.

The $\mathbb{F}_\mathsf{p}$-linear map $x \mapsto x^\mathsf{p} - x$ has kernel $\mathbb{F}_\mathsf{p}$ and image contained in the trace kernel. Both its image and the trace kernel have codimension one in $\mathbb{F}$. Some element of an $\mathbb{F}_\mathsf{p}$-basis of $\mathbb{F}$ has nonzero trace. After rescaling, the resulting element θ has trace one. Since θ lies outside the image of $x \mapsto x^\mathsf{p} - x$, the polynomial $X^\mathsf{p} - X - \theta$ has no root in $\mathbb{F}$. Its roots differ by elements of $\mathbb{F}_\mathsf{p}$, so its irreducible factors over $\mathbb{F}$ all have the same degree, which divides the prime p and is not one. Hence this polynomial is irreducible.

After $\log_\mathsf{p} c$ steps, each of degree p, the field has degree c over $\mathbb{K}$. Let ζ be the root adjoined in the last step. It lies outside the preceding field. Since c is a prime power, every proper intermediate field between $\mathbb{K}$ and $\mathbb{E}$ is contained in the preceding field. Thus ζ has degree c over $\mathbb{K}$. The change of basis can be computed by expressing $1, \zeta, \ldots, \zeta^{c-1}$ in the basis of the tower.

For fixed p, the tower can be constructed in time polynomial in $\tau + c$. Each arithmetic operation in $\mathbb{E}$ can be performed in time polynomial in $c \cdot e$. □

Construction A.4 (Polynomial evaluation tables, cf. [Son26, Section 3])**.** Let p be a prime, and let $N \geq 2$, $\mathsf{a} \geq 1$, and $\mathsf{d} \geq 2$ be integers. We construct a finite field $\mathbb{E}$ of characteristic p, a tuple $\boldsymbol{\beta} \in \mathbb{E}^N$, and a moment representation over $\mathbb{E}$, and define a binary evaluation table $\boldsymbol{q}_\alpha$ for every $\alpha \in \mathbb{E}$, as follows.

1. ***Fields and encoding tuple.*** Let c be the least power of p with $c \geq N$, and let $\tau \coloneqq \mathsf{p}^e$ be the least power of p with $\tau \geq 8 \cdot \mathsf{p} \cdot \mathsf{a} \cdot \mathsf{d} \cdot c$. Let $f \in \mathbb{F}_\mathsf{p}[X]$ be a monic irreducible polynomial of degree e, and let the sample field be $\mathbb{K} \coloneqq \mathbb{F}_\mathsf{p}[X]/(f)$, with $\xi \coloneqq X \bmod f$. Let $\mathbb{E} \coloneqq \mathbb{K}(\zeta)$ be

an extension of degree c. The fixed $\mathbb{F}_{\mathsf{p}}$-bases of $\mathbb{K}$ and $\mathbb{E}$ are, respectively, $\{\xi^i : 0 \le i < e\}$ and $\{\xi^i \cdot \zeta^j : 0 \le i < e,\ 0 \le j < c\}$. Let $\boldsymbol{\beta} \coloneqq (1, \zeta, \ldots, \zeta^{N-1})$.

2. ***Parameters.*** Let

$$\kappa \coloneqq \mathsf{d}^{\mathsf{a}}, \qquad m_{\mathrm{tab}} \coloneqq \tau^{\mathsf{a}+1}, \qquad \nu \coloneqq \mathsf{p} \cdot (\kappa - 1) + 1,$$
$$\mathcal{I} \coloneqq \{0, \ldots, 2 \cdot \mathsf{p} \cdot (\mathsf{d} - 1)\}^{\mathsf{a}}, \qquad W_{\deg} \coloneqq 2 \cdot \mathsf{p} \cdot \mathsf{a} \cdot (\mathsf{d} - 1) \cdot (c - 1)\,.$$

   Here κ is the moment parameter and m_{tab} is the evaluation table length. The output representation includes moment maps up to order $2 \cdot \kappa - 2$. For reconstruction, moments are defined up to order $2 \cdot \nu - 2$ and arranged in a $\nu \times \nu$ Hankel matrix.

3. ***Evaluation table coordinates.*** The coordinates of an evaluation table $\boldsymbol{v} \in \mathbb{F}_{\mathsf{p}}^{m_{\mathrm{tab}}}$ are indexed by pairs $(\boldsymbol{\lambda}, s) \in \mathbb{K}^{\mathsf{a}} \times \mathbb{K}$. The entry at $(\boldsymbol{\lambda}, s)$ is $v_{(\boldsymbol{\lambda}, s)}$. Let $\mathcal{P}$ be the partition of the coordinates into the τ blocks $\{(\boldsymbol{\lambda}, s) : \boldsymbol{\lambda} \in \mathbb{K}^{\mathsf{a}}\}$ $(s \in \mathbb{K})$. Each block is regarded as a column of length τ^{a}.

4. ***Interpolation and moment maps.*** For $\boldsymbol{\eta} \in \mathcal{I}$, let $P^{\boldsymbol{v}}_{\boldsymbol{\eta}} \in \mathbb{K}[X]$ be the unique polynomial of degree less than τ that interpolates the weighted column sums

$$P^{\boldsymbol{v}}_{\boldsymbol{\eta}}(s) = \sum_{\boldsymbol{\lambda} \in \mathbb{K}^{\mathsf{a}}} v_{(\boldsymbol{\lambda}, s)} \cdot \boldsymbol{\lambda}^{\boldsymbol{\eta}} \qquad (s \in \mathbb{K})\,,$$

   and let

$$\hat{\mu}_k(\boldsymbol{v}) \coloneqq P^{\boldsymbol{v}}_{\mathrm{digits}_{\mathsf{d},\mathsf{a}}(k)}(\zeta) \qquad (0 \le k \le 2 \cdot \nu - 2)\,.$$

5. ***Constraint matrix.*** Let $\boldsymbol{T} \cdot \boldsymbol{v} = \boldsymbol{0}$ be the system of low-degree, index-consistency, and Frobenius constraints

$$\begin{aligned} \deg P^{\boldsymbol{v}}_{\boldsymbol{\eta}} &\le W_{\deg} && (\boldsymbol{\eta} \in \mathcal{I})\,, \\ P^{\boldsymbol{v}}_{\boldsymbol{\eta}}(\zeta) &= \hat{\mu}_{\mathrm{val}_{\mathsf{d}}(\boldsymbol{\eta})}(\boldsymbol{v}) && (\boldsymbol{\eta} \in \mathcal{I})\,, \\ \hat{\mu}_{\mathsf{p} \cdot k}(\boldsymbol{v}) &= \hat{\mu}_k(\boldsymbol{v})^{\mathsf{p}} && (0 \le \mathsf{p} \cdot k \le 2 \cdot \nu - 2)\,, \end{aligned}$$

   expanded in the fixed $\mathbb{F}_{\mathsf{p}}$-bases. For the low-degree constraints, the coefficients of X^u in $P^{\boldsymbol{v}}_{\boldsymbol{\eta}}$ are required to vanish for $W_{\deg} < u < \tau$.

6. ***Evaluation table of a field element.*** For $\alpha \in \mathbb{E}$ and $0 \le h < \mathsf{a}$, let $\varphi_h \in \mathbb{K}[X]$ be the polynomial with $\deg \varphi_h < c$ and $\varphi_h(\zeta) = \alpha^{\mathsf{d}^h}$, and let $\boldsymbol{q}_\alpha \in \{0,1\}^{m_{\mathrm{tab}}}$ be the evaluation table with entries

$$(\boldsymbol{q}_\alpha)_{(\boldsymbol{\lambda}, s)} \coloneqq \begin{cases} 1 & \text{if } \boldsymbol{\lambda} = (\varphi_0(s), \ldots, \varphi_{\mathsf{a}-1}(s)), \\ 0 & \text{otherwise} \end{cases} \qquad ((\boldsymbol{\lambda}, s) \in \mathbb{K}^{\mathsf{a}} \times \mathbb{K})\,.$$

The output is the field $\mathbb{E}$ with its $\mathbb{F}_{\mathsf{p}}$-basis, the tuple $\boldsymbol{\beta}$, and the tuple $(\kappa, \tau, \tau^{\mathsf{a}}, \mathcal{P}, \boldsymbol{T}, (\hat{\mu}_k)_{k=0}^{2\cdot\kappa-2})$.

A.2 Rank and moment reconstruction

Definition A.5 (Polynomial matrix of an evaluation table)**.** *In Construction A.4, the* **polynomial matrix** *of an evaluation table $\boldsymbol{v} \in \mathbb{F}_{\mathsf{p}}^{m_{\mathrm{tab}}}$ is the $\nu \times \nu$ matrix $\boldsymbol{H}(X)$ over $\mathbb{K}[X]$ with entries*

$$H_{h,h'}(X) \coloneqq P^{\boldsymbol{v}}_{\mathrm{digits}_{\mathsf{d},\mathsf{a}}(h) + \mathrm{digits}_{\mathsf{d},\mathsf{a}}(h')}(X) \qquad (0 \le h, h' < \nu)\,.$$

The vectors $\mathrm{digits}_{\mathsf{d},\mathsf{a}}(h) + \mathrm{digits}_{\mathsf{d},\mathsf{a}}(h')$ *belong to $\mathcal{I}$ by Claim A.2 with $J = \mathsf{p}$, since $\nu - 1 = \mathsf{p} \cdot (\mathsf{d}^{\mathsf{a}} - 1)$.*

Lemma A.6 (cf. [Son26, Section 4])**.** *In Construction A.4, let $\boldsymbol{v} \in \mathbb{F}_{\mathsf{p}}^{m_{\mathrm{tab}}}$ satisfy $\boldsymbol{T} \cdot \boldsymbol{v} = \boldsymbol{0}$, let $\boldsymbol{H}(X)$ be its polynomial matrix, and let ρ be the rank of $\boldsymbol{H}(X)$ over the rational function field $\mathbb{K}(X)$. Then the following hold.*

1. **Weight.** $(\tau - W_{\deg}) \cdot \rho \le \mathrm{wt}(\boldsymbol{v})$.
2. **Moment matrix.** $\boldsymbol{H}(\zeta)$ *is the Hankel matrix* $(\hat{\mu}_{h+h'}(\boldsymbol{v}))_{0 \le h,h' < \nu}$ *of the moments of the evaluation table, and its rank over* $\mathbb{E}$ *is at most* ρ.

Proof. The entries of $\boldsymbol{H}(X)$ have degree at most $W_{\deg}$ by the low-degree constraints, and

$$\sum_{s \in \mathbb{K}} \operatorname{rank} \boldsymbol{H}(s) \ge \rho \cdot (\tau - W_{\deg}) \,. \tag{32}$$

If $\rho = 0$, the right-hand side vanishes. For $\rho > 0$, let $\boldsymbol{C}(X)$ be a $\rho \times \rho$ submatrix of $\boldsymbol{H}(X)$ with nonzero determinant, and for $s \in \mathbb{K}$, let $b_s \coloneqq \operatorname{rank} \boldsymbol{C}(s)$. There is an invertible matrix $\boldsymbol{U}$ over $\mathbb{K}$ such that $\rho - b_s$ rows of $\boldsymbol{U} \cdot \boldsymbol{C}(s)$ vanish. Every entry in these rows of $\boldsymbol{U} \cdot \boldsymbol{C}(X)$ is divisible by $X - s$, and $\det(\boldsymbol{U} \cdot \boldsymbol{C}(X)) = \det \boldsymbol{U} \cdot \det \boldsymbol{C}(X)$. Thus $(X - s)^{\rho - b_s}$ divides $\det \boldsymbol{C}(X)$, as in [FS14, Lemma 30]. Since the factors $X - s$ for distinct s are coprime,

$$\prod_{s \in \mathbb{K}} (X - s)^{\rho - b_s} \mid \det \boldsymbol{C}(X) \,, \qquad \sum_{s \in \mathbb{K}} (\rho - b_s) \le \deg \det \boldsymbol{C}(X) \le \rho \cdot W_{\deg} \,.$$

Since $\operatorname{rank} \boldsymbol{H}(s) \ge b_s$, Equation 32 holds after summing over s.

For $(h, \boldsymbol{\lambda}) \in \{0, \dots, \nu - 1\} \times \mathbb{K}^{\mathsf{a}}$, let $E_{h,\boldsymbol{\lambda}} \coloneqq \boldsymbol{\lambda}^{\mathrm{digits}_{\mathsf{d},\mathsf{a}}(h)}$. Then

$$\boldsymbol{H}(s) = \boldsymbol{E} \cdot \operatorname{diag}(v_{(\boldsymbol{\lambda},s)})_{\boldsymbol{\lambda}} \cdot \boldsymbol{E}^{\mathsf{T}} \qquad (s \in \mathbb{K}) \,.$$

The rank of $\boldsymbol{H}(s)$ is at most the Hamming weight of column s, so

$$\sum_{s \in \mathbb{K}} \operatorname{rank} \boldsymbol{H}(s) \le \mathrm{wt}(\boldsymbol{v}) \,.$$

By Equation 32 and the last inequality, the weight bound holds.

By Claim A.2 and the additivity of $\mathrm{val}_{\mathsf{d}}$, $\mathrm{val}_{\mathsf{d}}(\mathrm{digits}_{\mathsf{d},\mathsf{a}}(h) + \mathrm{digits}_{\mathsf{d},\mathsf{a}}(h')) = h + h'$. Hence, by the index-consistency constraints, the entries of $\boldsymbol{H}(\zeta)$ are the moments $\hat{\mu}_{h+h'}(\boldsymbol{v})$. The rank of $\boldsymbol{H}(\zeta)$ is at most ρ because every $(\rho+1) \times (\rho+1)$ minor of $\boldsymbol{H}(X)$ is the zero polynomial. □

Theorem A.7 (cf. [Son26, Lemmas 5.1 and 5.2])**.** *Let $\mathbb{E}$ be a finite field of characteristic p, with prime subfield $\mathbb{F}_{\mathsf{p}}$. For an integer $\kappa \ge 2$, let $\nu \coloneqq \mathsf{p} \cdot (\kappa - 1) + 1$. If $y_0, \dots, y_{2 \cdot \nu - 2} \in \mathbb{E}$ satisfy*

$$y_{\mathsf{p} \cdot k} = y_k^{\mathsf{p}} \quad (0 \le \mathsf{p} \cdot k \le 2 \cdot \nu - 2) \,, \qquad t \coloneqq \operatorname{rank}_{\mathbb{E}} (y_{i+j})_{0 \le i,j < \nu} < \kappa \,,$$

then there are a finite extension $\mathbb{E}'/\mathbb{E}$ and a coefficient function $\omega \colon \mathbb{E}' \to \mathbb{F}_{\mathsf{p}}$ with $|\operatorname{supp} \omega| \le t$ such that

$$y_k = \mu_k(\omega) \qquad (0 \le k < \nu) \,.$$

Proof. We first find a linear recurrence of order at most t for the given moments. Using the identities $y_{\mathsf{p} \cdot k} = y_k^{\mathsf{p}}$, we then prove that the characteristic polynomial of this recurrence has distinct roots and that the reconstructed coefficients lie in the prime subfield. Let $\boldsymbol{H} \coloneqq (y_{i+j})_{0 \le i,j < \nu}$. If its first column is zero, the conclusion holds with $\omega \coloneqq 0$. Otherwise, let h be the least index for which

columns $0, \ldots, h$ are linearly dependent. Such an index exists because $t < \kappa \leq \nu$. Let $\chi_0, \ldots, \chi_{h-1}$ be the coefficients of columns $0, \ldots, h-1$ in a dependence in which column h has coefficient one, and let

$$\chi(Y) \coloneqq Y^h + \sum_{i<h} \chi_i \cdot Y^i \,.$$

Then $1 \leq h \leq t < \kappa$, the first h columns are linearly independent, and row by row,

$$y_{j+h} = -\sum_{i<h} \chi_i \cdot y_{j+i} \qquad (0 \leq j < \nu) \,. \tag{33}$$

Let $\mathbb{E}'/\mathbb{E}$ be a splitting field of χ. For $f = a \cdot \chi + \sum_{i<h} b_i \cdot Y^i \in \mathbb{E}'[Y]$, we define a linear functional by its value on the remainder modulo χ:

$$\psi(f) \coloneqq \sum_{i<h} b_i \cdot y_i \,.$$

By uniqueness of the remainder modulo χ, ψ is a well-defined $\mathbb{E}'$-linear map with $\psi(\chi \cdot f) = 0$. For monomials,

$$\psi(Y^k) = y_k \qquad (0 \leq k \leq \nu + h - 1) \,. \tag{34}$$

These values are defined, since $h < \kappa < \nu$ and hence $\nu + h - 1 \leq 2 \cdot \nu - 2$. For $k < h$, Equation 34 is the definition of ψ. For larger k, it follows by induction on k from $Y^k \equiv -\sum_{i<h} \chi_i \cdot Y^{k-h+i} \pmod{\chi}$ and Equation 33 with $j = k - h$.

Let $b \coloneqq \sum_{i<h} b_i \cdot Y^i$ with $b_i \in \mathbb{E}'$. Since $h < \kappa$, $\mathsf{p} \cdot i < \nu$ for every $i < h$. By Equation 34 and the assumed identities $y_{\mathsf{p} \cdot k} = y_k^{\mathsf{p}}$,

$$\psi(b^{\mathsf{p}}) = \sum_{i<h} b_i^{\mathsf{p}} \cdot y_{\mathsf{p} \cdot i} = \sum_{i<h} (b_i \cdot y_i)^{\mathsf{p}} = \psi(b)^{\mathsf{p}} \,.$$

For arbitrary $f = a \cdot \chi + b$, both $f - b$ and $f^{\mathsf{p}} - b^{\mathsf{p}}$ are divisible by χ. Therefore

$$\psi(f^{\mathsf{p}}) = \psi(f)^{\mathsf{p}} \qquad (f \in \mathbb{E}'[Y]) \,. \tag{35}$$

If χ had repeated roots, we could construct a recurrence of lower order, contradicting the minimality of h. Let $\alpha_1, \ldots, \alpha_u$ be the distinct roots of χ. For $G \coloneqq \prod_{j=1}^{u} (Y - \alpha_j)$ and an integer $e \geq 0$ with $\mathsf{p}^e \geq h$, $\chi \mid G^{\mathsf{p}^e}$, so by repeated use of Equation 35, for every $f \in \mathbb{E}'[Y]$,

$$\psi(f \cdot G)^{\mathsf{p}^e} = \psi((f \cdot G)^{\mathsf{p}^e}) = 0 \quad \Longrightarrow \quad \psi(f \cdot G) = 0 \,.$$

If $u < h$, then $G = \sum_{i=0}^{u} g_i \cdot Y^i$, where $g_u = 1$, satisfies

$$0 = \psi(Y^j \cdot G) = \sum_{i=0}^{u} g_i \cdot y_{j+i} \qquad (0 \leq j < \nu) \,,$$

by Equation 34, since $j + i \leq \nu + h - 1$. The first $u + 1 \leq h$ columns of $\boldsymbol{H}$ would thus be linearly dependent over $\mathbb{E}'$, and hence over $\mathbb{E}$, contrary to their independence. Thus $u = h$, and the roots of χ are distinct.

For $1 \leq j \leq h$, let

$$\ell_j(Y) \coloneqq \prod_{\substack{1 \leq i \leq h \\ i \neq j}} \frac{Y - \alpha_i}{\alpha_j - \alpha_i} \,, \qquad \omega(\alpha_j) \coloneqq \psi(\ell_j) \,,$$

and $\omega(\alpha) \coloneqq 0$ for every other element of $\mathbb{E}'$. By Lagrange interpolation,

$$f \equiv \sum_{j=1}^{h} f(\alpha_j) \cdot \ell_j \pmod{\chi}, \qquad \psi(f) = \sum_{j=1}^{h} \omega(\alpha_j) \cdot f(\alpha_j)\,.$$

Since $\ell_j(\alpha_i) \in \{0,1\}$ for $1 \le i \le h$, the polynomial χ divides $\ell_j^{\mathsf{p}} - \ell_j$, and hence $\psi(\ell_j^{\mathsf{p}}) = \psi(\ell_j)$. By Equation 35, $\omega(\alpha_j)^{\mathsf{p}} = \psi(\ell_j^{\mathsf{p}}) = \omega(\alpha_j)$, so $\omega(\alpha_j) \in \mathbb{F}_{\mathsf{p}}$.

By Equation 34 and the formula for ψ,

$$y_k = \psi(Y^k) = \sum_{j=1}^{h} \omega(\alpha_j) \cdot \alpha_j^k = \mu_k(\omega) \qquad (0 \le k < \nu)\,,$$

and $|\operatorname{supp}\omega| \le h \le t$. □

A.3 Properties of the construction

Theorem A.8. *Let* p *be a prime, and let* $N \ge 2$, $\mathsf{a} \ge 1$, *and* $\mathsf{d} \ge 2$ *be integers. For the field, encoding tuple, and evaluation tables defined in Construction A.4 with these parameters,* $\boldsymbol{\beta}$ *is* $\mathbb{F}_{\mathsf{p}}$*-linearly independent, and the following hold.*

1. **Representation.** *The tuple* $(\kappa, \tau, \tau^{\mathsf{a}}, \mathcal{P}, \boldsymbol{T}, (\hat{\mu}_k)_{k=0}^{2\cdot\kappa-2})$ *is a moment representation over* $\mathbb{E}$*. For every* $\alpha \in \mathbb{E}$*, the evaluation table* $\boldsymbol{q}_\alpha$ *has exactly one nonzero coordinate in each block and satisfies*

$$\boldsymbol{T} \cdot \boldsymbol{q}_\alpha = \boldsymbol{0}\,, \qquad \hat{\mu}_k(\boldsymbol{q}_\alpha) = \alpha^k \quad (0 \le k \le 2\cdot\kappa - 2)\,.$$

2. **Size.** $\mathsf{d} \le \tau < 8 \cdot \mathsf{p}^3 \cdot \mathsf{a} \cdot N \cdot \mathsf{d}$.

3. **Construction time.** *The output of Construction A.4 can be computed in time* $(N + m_{\mathrm{tab}})^{O_{\mathsf{p}}(1)}$*, and so can* $\boldsymbol{q}_\alpha$ *from* $\alpha \in \mathbb{E}$ *given in the fixed* $\mathbb{F}_{\mathsf{p}}$*-basis.*

Proof. Since $[\mathbb{E} : \mathbb{K}] = c \ge N$, the elements $\beta_i = \zeta^{i-1}$ $(1 \le i \le N)$ are linearly independent over $\mathbb{K}$, hence over $\mathbb{F}_{\mathsf{p}}$.

By the choice of τ in Step 1,

$$\tau \ge 8 \cdot \mathsf{p} \cdot \mathsf{a} \cdot \mathsf{d} \cdot c > 4 \cdot W_{\deg}, \qquad 2 \cdot \mathsf{p} \cdot (\mathsf{d}-1) + 1 \le 2 \cdot \mathsf{p} \cdot \mathsf{d} \le \tau, \qquad \mathsf{d} \le \tau\,. \tag{36}$$

For $0 \le k \le 2 \cdot \nu - 2 = 2 \cdot \mathsf{p} \cdot (\mathsf{d}^{\mathsf{a}} - 1)$,

$$\operatorname{digits}_{\mathsf{d},\mathsf{a}}(k) \in \{0, \ldots, 2 \cdot \mathsf{p} \cdot (\mathsf{d}-1)\}^{\mathsf{a}} = \mathcal{I}\,,$$

by Claim A.2. Conversely, for $\boldsymbol{\eta} \in \mathcal{I}$,

$$\operatorname{val}_{\mathsf{d}}(\boldsymbol{\eta}) \le 2 \cdot \mathsf{p} \cdot (\kappa - 1) = 2 \cdot \nu - 2\,.$$

Hence the polynomials and moments used in Steps 4 and 5 are defined.

In Step 4, the weighted column sums are $\mathbb{F}_{\mathsf{p}}$-linear in the evaluation table entries. Interpolation and evaluation are linear, so each $\hat{\mu}_k$ is $\mathbb{F}_{\mathsf{p}}$-linear. In Step 5, the low-degree and index-consistency constraints are homogeneous $\mathbb{F}_{\mathsf{p}}$-linear equations in the evaluation table entries. The same holds for the Frobenius constraints, since $a \mapsto a^{\mathsf{p}}$ is $\mathbb{F}_{\mathsf{p}}$-linear.

For every $\alpha \in \mathbb{E}$, the evaluation table $\boldsymbol{q}_\alpha$ of Step 6 is binary with exactly one nonzero entry per column. For every $\boldsymbol{\eta} \in \mathcal{I}$, its weighted column sums are $\prod_h \varphi_h(s)^{\eta_h}$. Since $\deg \prod_h \varphi_h^{\eta_h} \le (c-1)\cdot\sum_h \eta_h \le W_{\deg} < \tau$,

$$P_{\boldsymbol{\eta}}^{\boldsymbol{q}_\alpha} = \prod_{h=0}^{\mathsf{a}-1} \varphi_h^{\eta_h}\,, \qquad P_{\boldsymbol{\eta}}^{\boldsymbol{q}_\alpha}(\zeta) = \alpha^{\mathrm{val}_{\mathsf{d}}(\boldsymbol{\eta})}\,.$$

By Claim A.2, $\hat{\mu}_k(\boldsymbol{q}_\alpha) = \alpha^k$ for $0 \le k \le 2\cdot\nu - 2$. The low-degree and index-consistency constraints hold, as do the Frobenius constraints because $\alpha^{\mathsf{p}\cdot k} = (\alpha^k)^{\mathsf{p}}$.

For soundness, let $\boldsymbol{v} \in \mathbb{F}_{\mathsf{p}}^{m_{\mathrm{tab}}}$ satisfy $\boldsymbol{T}\cdot\boldsymbol{v} = \boldsymbol{0}$ and $\mathrm{wt}(\boldsymbol{v}) < \kappa\cdot\tau/2$, let $\boldsymbol{H}(X)$ be its polynomial matrix, and let ρ be the rank of $\boldsymbol{H}(X)$ over $\mathbb{K}(X)$. By Equation 36, $\tau - W_{\deg} > 3\cdot\tau/4$. Hence, by Lemma A.6,

$$\operatorname{rank}\boldsymbol{H}(\zeta) \le \rho \le \frac{\mathrm{wt}(\boldsymbol{v})}{\tau - W_{\deg}} \le \frac{4\cdot\mathrm{wt}(\boldsymbol{v})}{3\cdot\tau} < \frac{2\cdot\kappa}{3} < \kappa\,.$$

By Lemma A.6, $\boldsymbol{H}(\zeta)$ is the Hankel matrix of the moments $\hat{\mu}_k(\boldsymbol{v})$, which satisfy the Frobenius constraints of Step 5. By Theorem A.7 with $y_k \coloneqq \hat{\mu}_k(\boldsymbol{v})$, there are a finite extension $\mathbb{E}_{\boldsymbol{v}}/\mathbb{E}$ and a coefficient function $\omega\colon \mathbb{E}_{\boldsymbol{v}} \to \mathbb{F}_{\mathsf{p}}$ with

$$\hat{\mu}_k(\boldsymbol{v}) = \mu_k(\omega) \quad (0 \le k < \nu)\,, \qquad |\operatorname{supp}\omega| \le \frac{4\cdot\mathrm{wt}(\boldsymbol{v})}{3\cdot\tau} \le \frac{2\cdot\mathrm{wt}(\boldsymbol{v})}{\tau}\,.$$

These identities hold for all $0 \le k \le 2\cdot\kappa - 2$ because $\nu - 1 = \mathsf{p}\cdot(\kappa-1) \ge 2\cdot\kappa - 2$.

By Step 1 of Construction A.4,

$$c < \mathsf{p}\cdot N\,, \qquad \tau < 8\cdot\mathsf{p}^2\cdot\mathsf{a}\cdot\mathsf{d}\cdot c < 8\cdot\mathsf{p}^3\cdot\mathsf{a}\cdot N\cdot\mathsf{d}\,.$$

By Lemma A.3, the fields can be constructed in time polynomial in $\tau + c$ for fixed p. Interpolation, coefficient extraction, evaluation, and Frobenius are $\mathbb{F}_{\mathsf{p}}$-linear operations on explicit coefficient vectors, and only $\mathbb{K}$ and $\mathbb{K}^{\mathsf{a}}$ are enumerated, not the τ^c elements of $\mathbb{E}$. There are $|\mathcal{I}| \le \tau^{\mathsf{a}}$ low-degree constraints, as many index-consistency constraints, and at most $2\cdot\mathsf{d}^{\mathsf{a}} \le 2\cdot\tau^{\mathsf{a}}$ Frobenius constraints. After expansion in the fixed $\mathbb{F}_{\mathsf{p}}$-bases, there are at most $\tau\cdot e \le \tau^2$ rows of $\boldsymbol{T}$ per constraint. Hence $\boldsymbol{T}$ has m_{tab} columns and at most $4\cdot\tau^{\mathsf{a}+2} \le 4\cdot m_{\mathrm{tab}}^2$ rows, and the output is constructible in time $(N + m_{\mathrm{tab}})^{O_{\mathsf{p}}(1)}$.

Given $\alpha \in \mathbb{E}$, the powers α^{d^h} $(0 \le h < \mathsf{a})$ can be computed by successive powering, and their coordinates in the power basis are the coefficients of the polynomials φ_h. Since $c, \mathsf{a}, \mathsf{d}, \tau \le m_{\mathrm{tab}}$ and field arithmetic takes polynomial time, $\boldsymbol{q}_\alpha$ can be constructed within the same time bound by evaluating the polynomials φ_h on $\mathbb{K}$. □

A.4 Proof of the bounds on system size

Proof of Theorem 4.8. Given a. Since an assignment contains the input bits and the output bit, $N \ge n_{\mathbb{x}} + 1 \ge 2$. The number σ of evaluation tables in Construction 4.5 is at most $(N+1)^2$. Let $\mathsf{d} \ge 2$ be the least integer with $\mathsf{d}^{\mathsf{a}} \ge 8\cdot\sigma\cdot\Gamma$. In Construction 4.5, we use the tuple $\boldsymbol{\beta}$ and the moment representation of Construction A.4 with parameters $N, \mathsf{a}, \mathsf{d}$. By Theorem A.8, $\boldsymbol{\beta}$ is $\mathbb{F}_{\mathsf{p}}$-linearly independent, and the moment representation has $\kappa = \mathsf{d}^{\mathsf{a}}$ and $D_{\mathrm{tab}} = \tau$ blocks of size $Q = \tau^{\mathsf{a}}$. The parameters of Construction 4.5 are

$$D = \sigma\cdot\tau\,, \qquad n_{\mathrm{sys}} = D\cdot Q = \sigma\cdot\tau^{\mathsf{a}+1}\,.$$

Since $\tau \geq \mathsf{d}$, the block size satisfies $Q \geq \mathsf{d}^{\mathsf{a}} \geq 8 \cdot \sigma \cdot \Gamma > \Gamma$, and $2 \cdot \Gamma \cdot \sigma < \kappa$. Completeness and soundness therefore hold by Theorem 4.6, with gap Γ.

The size parameters satisfy

$$\sigma \leq (N+1)^2\,, \qquad \mathsf{d} < \mathsf{p} \cdot (8 \cdot \sigma \cdot \Gamma)^{1/\mathsf{a}}\,, \qquad \tau < 8 \cdot \mathsf{p}^3 \cdot \mathsf{a} \cdot N \cdot \mathsf{d}\,.$$

The bound on τ is that of Theorem A.8. For the bound on d, either $\mathsf{d} = 2$ or, by its minimality, $(\mathsf{d}-1)^{\mathsf{a}} < 8 \cdot \sigma \cdot \Gamma$. In both cases $\mathsf{d} < 2 \cdot (8 \cdot \sigma \cdot \Gamma)^{1/\mathsf{a}} \leq \mathsf{p} \cdot (8 \cdot \sigma \cdot \Gamma)^{1/\mathsf{a}}$. Hence, with $C'_{\mathsf{p}} \coloneqq 8 \cdot \mathsf{p}^4$,

$$n_{\mathrm{sys}} = \sigma \cdot \tau^{\mathsf{a}+1} \leq \sigma \cdot (C'_{\mathsf{p}} \cdot \mathsf{a} \cdot N)^{\mathsf{a}+1} \cdot (8 \cdot \sigma \cdot \Gamma)^{1+1/\mathsf{a}}\,. \tag{37}$$

Since $\sigma \leq (N+1)^2$, $\mathsf{a} \geq 1$, and $8^{1+1/\mathsf{a}} \leq 64 \leq (C'_{\mathsf{p}} \cdot \mathsf{a})^6$,

$$n_{\mathrm{sys}} \leq 8^{1+1/\mathsf{a}} \cdot (C'_{\mathsf{p}} \cdot \mathsf{a})^{\mathsf{a}+1} \cdot (N+1)^{\mathsf{a}+5+2/\mathsf{a}} \cdot \Gamma^{\,1+1/\mathsf{a}} \leq \left(C'_{\mathsf{p}} \cdot \mathsf{a} \cdot (N+1)\right)^{\mathsf{a}+7} \cdot \Gamma^{\,1+1/\mathsf{a}}\,,$$

as claimed in Equation 22. For fixed a, Equation 23 holds with $C_{\mathsf{p},\mathsf{a}} \coloneqq (\mathsf{a}+7) \cdot (1 + \log_3(C'_{\mathsf{p}} \cdot \mathsf{a}))$, since $N + 1 \geq 3$.

By Theorem A.8, the moment representation and $\boldsymbol{\beta}$, which depend only on $\mathsf{p}, N, \mathsf{a}, \mathsf{d}$, can be computed in time $(N + m_{\mathrm{tab}})^{O_{\mathsf{p}}(1)}$, which also bounds their description length. Field arithmetic in $\mathbb{E}$ takes polynomial time by Lemma A.3. By the time bound of Theorem 4.6, and since $m_{\mathrm{tab}} \leq n_{\mathrm{sys}}$, the matrix and the right-hand side for each instance can be computed in time $(N + n_{\mathrm{sys}})^{O_{\mathsf{p}}(1)}$, with an exponent independent of a. This is polynomial in $N + \Gamma$ for fixed p, a.

Choosing a. For $1 \leq \Gamma \leq 2^N$, let $K \coloneqq 8 \cdot (N+1)^2 \cdot \Gamma \geq 8 \cdot \sigma \cdot \Gamma$. To balance $(C'_{\mathsf{p}} \cdot \mathsf{a} \cdot N)^{\mathsf{a}+1}$ and $K^{1/\mathsf{a}}$ in Equation 37, we choose

$$L \coloneqq \lceil \log_2 K \rceil, \qquad A \coloneqq \lceil \log_2(C'_{\mathsf{p}} \cdot N \cdot (L+1)) \rceil, \qquad \mathsf{a} \coloneqq \left\lceil \sqrt{L/A} \right\rceil\,.$$

Then $\mathsf{a} \leq L + 1$, so $C'_{\mathsf{p}} \cdot \mathsf{a} \cdot N \leq 2^A$. Also, $L/\mathsf{a} \leq \sqrt{L \cdot A}$ and $(\mathsf{a}+1) \cdot A \leq \sqrt{L \cdot A} + 2 \cdot A$. Thus, by Equation 37,

$$n_{\mathrm{sys}} \leq \sigma \cdot K \cdot (C'_{\mathsf{p}} \cdot \mathsf{a} \cdot N)^{\mathsf{a}+1} \cdot K^{1/\mathsf{a}} \leq \sigma \cdot K \cdot 2^{2 \cdot \sqrt{L \cdot A} + 2 \cdot A}\,.$$

Since $\Gamma \leq 2^N$, $A = O_{\mathsf{p}}(\log N)$ and $L = O(\log \Gamma + \log N)$, so Equation 24 holds. Also $2 \cdot \sqrt{L \cdot A} \leq L + A$, so $n_{\mathrm{sys}} \leq \sigma \cdot K \cdot 2^{L+3 \cdot A} = \mathrm{poly}(N + \Gamma)$. For fixed p, we can therefore construct these instances in time polynomial in $N + \Gamma$. □

B Code and lattice conversions

B.1 A fixed code

Proof of Lemma 3.8. Let $C \coloneqq \boldsymbol{G}_1 \cdot \ker \boldsymbol{G}_0$. If the system is feasible, we compute a solution $\boldsymbol{v}_0$ and define $\boldsymbol{y_b} \coloneqq \boldsymbol{G}_1 \cdot \boldsymbol{v}_0$. Then

$$\boldsymbol{y_b} - C = \{\boldsymbol{G}_1 \cdot \boldsymbol{v} : \boldsymbol{G}_0 \cdot \boldsymbol{v} = \boldsymbol{b}\},$$

so the minimum Hamming weight in this coset is $\Delta(\boldsymbol{y_b}, C)$. Otherwise, we set $\boldsymbol{y_b} \coloneqq \boldsymbol{0}^M$. A generator matrix and the targets are computable by Gaussian elimination. □

Proof of Lemma 3.11. For $n \coloneqq D \cdot Q$ and $M \coloneqq 2 \cdot n$, let $C \coloneqq \ker \boldsymbol{A} \times \{\mathbf{0}^n\}$. The first half has the given block partition, and the second has a copy of that partition.

If the system is feasible, we compute a solution $\boldsymbol{v}_0$ and set $\boldsymbol{y_b} \coloneqq (\boldsymbol{v}_0, \mathbf{0}^n)$. Then

$$\boldsymbol{y_b} - C = \{(\boldsymbol{v}, \mathbf{0}^n) : \boldsymbol{A} \cdot \boldsymbol{v} = \boldsymbol{b}\}.$$

Otherwise, we set $\boldsymbol{y_b} \coloneqq (\mathbf{0}^n, \mathbf{1}^n)$. In this case, the system is a NO instance, and $\Delta(\boldsymbol{y_b}, C) = n = D \cdot Q > \Gamma \cdot D$. Thus, by Definition 3.10,

$$\begin{aligned} \text{YES instance} \quad &\Longrightarrow \quad \exists \boldsymbol{e} \in \{0,1\}^M : \quad \boldsymbol{y_b} - \boldsymbol{e} \in C, \quad \mathrm{wt}(\boldsymbol{e}) \le D, \\ \text{NO instance} \quad &\Longrightarrow \quad \Delta(\boldsymbol{y_b}, C) > \Gamma \cdot D. \end{aligned} \tag{38}$$

In a YES instance, $\boldsymbol{e}$ can be chosen with at most one nonzero entry per block.

We choose an $\mathbb{F}_\mathsf{p}$-linear map $\rho\colon \mathbb{F} \to \mathbb{F}_\mathsf{p}$ fixing $\mathbb{F}_\mathsf{p}$, and define $C_\mathbb{F} \coloneqq \mathrm{span}_\mathbb{F} C$. Applied coordinatewise, $\rho(C_\mathbb{F}) \subseteq C$, $\rho(\boldsymbol{y_b}) = \boldsymbol{y_b}$, and $\mathrm{wt}(\rho(\boldsymbol{z})) \le \mathrm{wt}(\boldsymbol{z})$. Thus $\Delta(\boldsymbol{y_b}, C_\mathbb{F}) = \Delta(\boldsymbol{y_b}, C)$. The output code $C_\mathbb{F}$ depends only on $\boldsymbol{A}$. Its generator matrix and each target $\boldsymbol{y_b}$ are computable in polynomial time from the affine systems. The radius is $R \coloneqq D$. □

B.2 Lattice gaps

Lemma B.1 (From codes to lattices)**.** *Let* p *be prime and let* $C \subseteq \mathbb{F}_\mathsf{p}^M$ *be a linear code, and define the lattice*

$$\mathcal{L}_C \coloneqq \{\boldsymbol{v} \in \mathbb{Z}^M : \boldsymbol{v} \bmod \mathsf{p} \in C\}.$$

For each target $\boldsymbol{y} \in \mathbb{F}_\mathsf{p}^M$*, let* $\widetilde{\boldsymbol{y}} \in \{0, \dots, \mathsf{p}-1\}^M$ *be the vector with* $\widetilde{\boldsymbol{y}} \bmod \mathsf{p} = \boldsymbol{y}$*. For every* $1 \le p < \infty$*,*

$$\mathrm{dist}_p(\widetilde{\boldsymbol{y}}, \mathcal{L}_C)^p \ge \Delta(\boldsymbol{y}, C).$$

If $\boldsymbol{y} - \boldsymbol{e} \in C$ *for a binary vector* $\boldsymbol{e} \in \{0,1\}^M$*, then*

$$\mathrm{dist}_p(\widetilde{\boldsymbol{y}}, \mathcal{L}_C)^p \le \mathrm{wt}(\boldsymbol{e}).$$

Proof. For every $\boldsymbol{v} \in \mathcal{L}_C$,

$$\Delta(\boldsymbol{y}, C) \le \mathrm{wt}((\widetilde{\boldsymbol{y}} - \boldsymbol{v}) \bmod \mathsf{p}) \le \|\widetilde{\boldsymbol{y}} - \boldsymbol{v}\|_p^p. \tag{39}$$

If $\boldsymbol{y} - \boldsymbol{e} \in C$ with $\boldsymbol{e} \in \{0,1\}^M$, then $\widetilde{\boldsymbol{y}} - \boldsymbol{e} \in \mathcal{L}_C$ and $\|\boldsymbol{e}\|_p^p = \mathrm{wt}(\boldsymbol{e})$. The lattice has full rank, since $\mathsf{p} \cdot \mathbb{Z}^M \subseteq \mathcal{L}_C \subseteq \mathbb{Z}^M$.

Let $k \coloneqq \dim_{\mathbb{F}_\mathsf{p}} C$. By Gaussian elimination and a coordinate permutation, we put a given generator matrix in systematic form $[I_k \ G]$. As in [HS26a, Section 8.1, Lemma 8.1], we use the basis

$$\boldsymbol{B}_C \coloneqq P \cdot \begin{pmatrix} I_k & 0 \\ G^\mathsf{T} & \mathsf{p} \cdot I_{M-k} \end{pmatrix}, \tag{40}$$

where P is the permutation matrix for the original coordinate order and the entries of G are represented by integers in $\{0, \dots, \mathsf{p}-1\}$. The matrix $\boldsymbol{B}_C$ is a basis of $\mathcal{L}_C$, since in systematic order every vector of $\mathcal{L}_C$ can be written uniquely as $(\boldsymbol{u}, G^\mathsf{T} \cdot \boldsymbol{u} + \mathsf{p} \cdot \boldsymbol{u}')$ with $\boldsymbol{u} \in \mathbb{Z}^k$ and $\boldsymbol{u}' \in \mathbb{Z}^{M-k}$. □

Lemma B.2 (Hadamard blocks)**.** *Let $C \subseteq \mathbb{F}_{\mathsf{p}}^{M}$ and $\boldsymbol{y_b}$ satisfy Equation 38, with $M = 2 \cdot D \cdot Q$ and a partition into blocks of size Q, where Q is a power of two. Suppose that in the YES case $\boldsymbol{e}$ can be chosen with at most one nonzero entry per block. Let $\mathcal{H}$ be block diagonal with a $Q \times Q$ Sylvester–Hadamard matrix on each block. For every $2 \le p \le \infty$, let*

$$\mathcal{L} \coloneqq \mathcal{H} \cdot \mathcal{L}_C, \qquad \boldsymbol{t_b} \coloneqq \mathcal{H} \cdot \widetilde{\boldsymbol{y}}_{\boldsymbol{b}}, \qquad R_0 \coloneqq (Q \cdot D)^{1/p}, \qquad c_p \coloneqq 2^{1-2/p},$$

where $1/\infty \coloneqq 0$. Then

$$\begin{aligned} \textit{YES instance} \quad &\Longrightarrow \quad \operatorname{dist}_p(\boldsymbol{t_b}, \mathcal{L}) \le R_0, \\ \textit{NO instance} \quad &\Longrightarrow \quad \operatorname{dist}_p(\boldsymbol{t_b}, \mathcal{L}) > \sqrt{\Gamma / c_p} \cdot R_0. \end{aligned}$$

Proof. We follow Hair and Sahai [HS26b, Section 3 and Lemma 6.2] in applying Hadamard matrices separately to the coordinate blocks, with square matrices in place of the rectangular transformation in their later paper [HS26a, Section 8.2]. The Sylvester matrices $H_1 = (1)$ and $H_{2 \cdot b} = \left(\begin{smallmatrix} H_b & H_b \\ H_b & -H_b \end{smallmatrix}\right)$ have entries ± 1 and satisfy $H_Q^{\mathsf{T}} \cdot H_Q = Q \cdot I_Q$. Since $\mathcal{H}$ is nonsingular, we take

$$\boldsymbol{B} \coloneqq \mathcal{H} \cdot \boldsymbol{B}_C \tag{41}$$

as an integer basis of $\mathcal{L}$, with $\boldsymbol{B}_C$ as in Equation 40.

For each nonzero block of $\boldsymbol{e}$, the corresponding block of $\mathcal{H} \cdot \boldsymbol{e}$ is a column of the Hadamard matrix, with entries ± 1. Thus $\boldsymbol{t_b} - \mathcal{H} \cdot \boldsymbol{e} \in \mathcal{L}$, and $\mathcal{H} \cdot \boldsymbol{e}$ has at most $Q \cdot D$ nonzero entries, all of magnitude one, as required for a YES instance.

For any $\boldsymbol{w} \in \mathcal{L}$, let $\boldsymbol{r} = \boldsymbol{t_b} - \boldsymbol{w}$. Before the transformation, the residual is $\boldsymbol{d} \coloneqq Q^{-1} \cdot \mathcal{H}^{\mathsf{T}} \cdot \boldsymbol{r} \in \widetilde{\boldsymbol{y}}_{\boldsymbol{b}} - \mathcal{L}_C$. Thus $\boldsymbol{d} \in \mathbb{Z}^M$ and $\boldsymbol{d} \bmod \mathsf{p} \in \boldsymbol{y_b} - C$. Thus

$$\begin{aligned} \Delta(\boldsymbol{y_b}, C) &\le \operatorname{wt}(\boldsymbol{d} \bmod \mathsf{p}) \le \|\boldsymbol{d}\|_2^2 = Q^{-1} \cdot \|\boldsymbol{r}\|_2^2 \\ &\le Q^{-1} \cdot M^{1-2/p} \cdot \|\boldsymbol{r}\|_p^2 = c_p \cdot D \cdot \left(\|\boldsymbol{r}\|_p / R_0\right)^2. \end{aligned} \tag{42}$$

For a NO instance, the left-hand side exceeds $\Gamma \cdot D$. □

Proof of Lemma 3.9. For a binary code $C \subseteq \mathbb{F}_2^M$, we use $\mathcal{L}_C$ and $\boldsymbol{t_y} \coloneqq \widetilde{\boldsymbol{y}} \in \{0,1\}^M$. For every $\boldsymbol{v} \in \mathcal{L}_C$, $\boldsymbol{c} \coloneqq \boldsymbol{v} \bmod 2 \in C$ and

$$\Delta(\boldsymbol{y}, C) \le \Delta(\boldsymbol{y}, \boldsymbol{c}) \le \|\boldsymbol{t_y} - \boldsymbol{v}\|_p^p.$$

Choose a nearest codeword $\boldsymbol{c}$, and let $\boldsymbol{e} \in \{0,1\}^M$ have $e_i = 1$ when $y_i \ne c_i$ and $e_i = 0$ otherwise. Then $\boldsymbol{t_y} - \boldsymbol{e} \in \mathcal{L}_C$ and $\|\boldsymbol{e}\|_p^p = \Delta(\boldsymbol{y}, C)$, so equality is attained. The basis in Equation 40 depends only on C. □

B.3 The lattice reduction

Proof of Lemma 3.12. We start with the code $C \subseteq \mathbb{F}_{\mathsf{p}}^{2 \cdot D \cdot Q}$ and targets $\boldsymbol{y_b}$ from Equation 38. Each block of every codeword and target is padded with zeros to size Q', where $Q' \coloneqq Q$ for $p \le 2$ and $Q' \coloneqq 2^{\lceil \log_2 Q \rceil} < 2 \cdot Q$ for $p > 2$. We pad $\boldsymbol{e}$ identically in YES instances and keep the notation $C, \boldsymbol{y_b}, \boldsymbol{e}$. Then Equation 38 still holds, and $\boldsymbol{e}$ still has at most one nonzero entry per block. The

padded code has block length $M \coloneqq 2 \cdot D \cdot Q'$, and $2 \cdot D \cdot Q \le M < 4 \cdot D \cdot Q$. For $p > 2$, let $\mathcal{H}$ be the matrix of Lemma B.2 with block size Q'. The radius is $R \coloneqq \lceil R_0 \rceil$, where

$$(\mathcal{L}, \boldsymbol{t_b}, R_0) \coloneqq \begin{cases} (\mathcal{L}_C,\ \widetilde{\boldsymbol{y}}_{\boldsymbol{b}},\ D^{1/p}) & \text{for } 1 \le p \le 2, \\ (\mathcal{H} \cdot \mathcal{L}_C,\ \mathcal{H} \cdot \widetilde{\boldsymbol{y}}_{\boldsymbol{b}},\ (Q' \cdot D)^{1/p}) & \text{for } 2 < p \le \infty. \end{cases}$$

Since $1 \le R_0 \le M$, the radius satisfies $1 \le R \le M$ and $R_0 \le R \le 2 \cdot R_0$. For $p = \infty$, $R = R_0 = 1$. For $p = u/v$ with positive integers u and v, we find the least $R \in \{1, \dots, M\}$ satisfying $R^u \ge (R_0^p)^v$ by binary search. Here $R_0^p \in \{D, Q' \cdot D\}$.

For $\boldsymbol{v} \in \mathcal{L}$, let $r \coloneqq \|\boldsymbol{t_b} - \boldsymbol{v}\|_p$ and

$$w(\boldsymbol{v}) \coloneqq \begin{cases} \mathrm{wt}((\boldsymbol{t_b} - \boldsymbol{v}) \bmod \mathsf{p}) & \text{for } 1 \le p \le 2, \\ \mathrm{wt}(Q'^{-1} \cdot \mathcal{H}^{\mathsf{T}} \cdot (\boldsymbol{t_b} - \boldsymbol{v}) \bmod \mathsf{p}) & \text{for } 2 < p \le \infty. \end{cases}$$

By Equations 39 and 42, and since $R \le 2 \cdot R_0$ and $c_p \le 2$,

$$\Delta(\boldsymbol{y_b}, C) \le w(\boldsymbol{v}) \le \begin{cases} D \cdot (r/R_0)^p & \text{for } 1 \le p \le 2 \\ c_p \cdot D \cdot (r/R_0)^2 & \text{for } 2 < p \le \infty \end{cases} \le 16 \cdot D \cdot (r/R)^s. \tag{43}$$

For YES instances, $\mathrm{dist}_p(\boldsymbol{t_b}, \mathcal{L}) \le R_0 \le R$ by Lemma B.1 with $\mathrm{wt}(\boldsymbol{e}) \le D$ for $p \le 2$, and by Lemma B.2 for $p > 2$. For NO instances, by Equations 38 and 43, every $\boldsymbol{v} \in \mathcal{L}$ satisfies $\Gamma \cdot D < 16 \cdot D \cdot (r/R)^s$. Since $\Gamma > 16 \cdot f^s$,

$$\mathrm{dist}_p(\boldsymbol{t_b}, \mathcal{L}) > (\Gamma/16)^{1/s} \cdot R > f \cdot R.$$

The bases in Equations 40 and 41 have integer entries of magnitude at most $\mathsf{p} \cdot Q'$, and $\boldsymbol{t_b}$ has integer entries of magnitude at most $(\mathsf{p} - 1) \cdot Q'$. Since $Q' \le M$, the basis and target have binary description lengths $O(M^2 \cdot \log(\mathsf{p} \cdot M))$ and $O(M \cdot \log(\mathsf{p} \cdot M))$, and both can be computed in polynomial time. The basis and R depend only on $\boldsymbol{A}$ and its partition, $\boldsymbol{t_b}$ depends on $\boldsymbol{b}$, and none of them depends on f. For $\boldsymbol{v} \in \mathbb{Z}^M$,

$$\boldsymbol{v} \in \mathcal{L}_C \iff \boldsymbol{v} \bmod \mathsf{p} \in C, \qquad \boldsymbol{v} \in \mathcal{H} \cdot \mathcal{L}_C \iff Q'^{-1} \cdot \mathcal{H}^{\mathsf{T}} \cdot \boldsymbol{v} \in \mathcal{L}_C.$$

Thus we can test membership and compute $w(\boldsymbol{v})$ using exact integer and field arithmetic, in time polynomial in the bit lengths of the instance and of $\boldsymbol{v}$. □

C Proof of Corollary 5.5

Proof. For $k \ge 3$, let $g_k(M) \coloneqq M^{1/s - 1/k}$. Fixed inputs and targets satisfy the description-length bounds of Theorem 5.1, with constants independent of k; radii are integers $1 \le R \le M$, as in Lemmas 3.11 and 3.12.

For each k, there are arbitrarily large M and fixed pairs $(\mathcal{S}, R)$ for which the $g_k(M)$-gap problem cannot be solved by circuits of size at most M^k. Otherwise, by Theorem 5.1 with $\varepsilon = 1/k$, we could decide 3SAT by polynomial-size circuits, using lookup tables indexed by targets for bounded M, contrary to NP $\not\subseteq$ P/poly.

We choose such pairs $(\mathcal{S}_k, R_k)$ at successive dimensions $M_3 < M_4 < \cdots$, taking each M_k as small as possible. The sequence is computable by exhaustive search over the finitely many encodings and

circuits at each dimension, since both promise conditions are decidable for codes and for lattices with rational p or $p = \infty$. The function

$$f(M) \coloneqq \begin{cases} 1 & \text{for } M < M_3, \\ \lceil g_k(M)/4 \rceil & \text{for } M_k \le M < M_{k+1} \end{cases}$$

is computable and nondecreasing, with $f(M) = \Theta(g_k(M))$ for $M_k \le M < M_{k+1}$. Since $k \to \infty$ as $M \to \infty$, $f(M) = M^{1/s - o(1)}$.

Suppose a preprocessing algorithm solves the f-gap problem or f-approximate search with output length and online time $\mathrm{poly}(M)$. The representation of $\mathcal{S}_k$, the output of the preprocessing phase on it, R_k, and $f(M_k)$ are hardwired into the online circuit. For search, the circuit rejects nonmembers and computes in polynomial time a number between the distance from the target to the returned vector and twice that distance. It accepts if this number is at most $2 \cdot f(M_k) \cdot R_k$.

Since $2 \cdot f(M_k) \le g_k(M_k)$ for all sufficiently large k, in either case there is a circuit of size $M_k^{O(1)}$ for the $g_k(M_k)$-gap problem, with exponent independent of k. For sufficiently large k, its size is at most M_k^k, contrary to the choice of $(\mathcal{S}_k, R_k)$. □

D Comparison with the Label Cover reductions

Alekhnovich, Khot, Kindler, and Vishnoi [AKKV05] and Khot, Popat, and Vishnoi [KPV14] reduce from smooth Label Cover. The bipartite formulation below is a schematic account of the approach in the conference version of [AKKV05]; Khot, Popat, and Vishnoi use layered hypergraph Label Cover with equalities between two projection sums [KPV14, Sections 3.2 and 3.3].

Label Cover with preprocessing. A Label Cover instance consists of a bipartite graph $G = (A \cup B, E)$ with left vertex set A and right vertex set B, finite alphabets Σ_A, Σ_B, and a projection $\pi_e \colon \Sigma_A \to \Sigma_B$ for each edge $e \in E$. We allow parallel edges. A labeling is a pair of maps $\lambda_A \colon A \to \Sigma_A$ and $\lambda_B \colon B \to \Sigma_B$. An edge $e = (r, u)$ is satisfied exactly when $\pi_e(\lambda_A(r)) = \lambda_B(u)$.

The graph, alphabets, and projections depend only on $n_{\mathbb{x}}$. For each $r \in A$, the alphabet Σ_A is partitioned into fixed classes $\mathcal{P}_r$. For each instance $\mathbb{x}$, one allowed class $P_{\mathbb{x},r} \in \mathcal{P}_r$ is specified at each left vertex, and a labeling must also satisfy $\lambda_A(r) \in P_{\mathbb{x},r}$.

From Label Cover to a linear system. Over a finite field $\mathbb{F}$, there is a variable $v_{r,a}$ for each $(r, a) \in A \times \Sigma_A$ and a variable $v_{u,b}$ for each $(u, b) \in B \times \Sigma_B$. For every left vertex and every class, we impose

$$\sum_{a \in P} v_{r,a} = \begin{cases} 1, & P = P_{\mathbb{x},r}, \\ 0, & P \ne P_{\mathbb{x},r}, \end{cases} \qquad (r \in A,\ P \in \mathcal{P}_r). \tag{44}$$

For every edge and every right label, we also impose the projection equation

$$v_{u,b} = \sum_{\substack{a \in \Sigma_A \\ \pi_e(a) = b}} v_{r,a} \qquad (e = (r, u) \in E,\ b \in \Sigma_B). \tag{45}$$

The matrix depends only on $n_{\mathbb{x}}$, and only the right-hand side depends on $\mathbb{x}$. For a satisfying labeling, set $v_{r,a} = 1$ exactly when $a = \lambda_A(r)$ and $v_{u,b} = 1$ exactly when $b = \lambda_B(u)$, with all other variables zero. The resulting solution has one nonzero variable per vertex.

To relate this formulation to a PCP, allow the left alphabet to depend on the vertex. Take the left vertices to be random strings r, with labels the answer tuples accepted by some predicate in $\mathcal{D}_r$, and the right vertices to be proof locations, with labels in Σ. Tuples must have equal answers at repeated locations. For each query occurrence $(r, j) \in \{0,1\}^{\mathsf{R}} \times [\mathsf{q}]$, there is an edge to $u_{r,j}$ with projection $\boldsymbol{a} \mapsto a_j$. The query locations and predicate families are fixed, and $V_{\varkappa,r}$ is selected according to the instance. If the sets of labels accepted by distinct predicates are disjoint and nonempty, they form the classes at r. A proof accepted on every random string then corresponds to a satisfying labeling. For PCP soundness, we measure the fraction of random strings for which all query edges are satisfied; for Label Cover, we measure the fraction of satisfied edges.

Cancellation and smoothness. For each vertex, form the list of labels whose variables are nonzero. For a solution of Hamming weight at most h times that of an honest solution, the average list size is at most h. For soundness, we seek a labeling from these lists whose fraction of satisfied edges exceeds the soundness error of a NO instance. If the label chosen at r is a, the edge $e = (r, u)$ can be satisfied only if $\pi_e(a)$ is in the list at u. Even when $v_{r,a} \neq 0$, however, we can have $v_{u,\pi_e(a)} = 0$ because of cancellation in (45). For example, two labels $a \neq a'$ with $\pi_e(a) = \pi_e(a')$ and $v_{r,a} = -v_{r,a'} \neq 0$ have zero total contribution to that projection sum.

Following Khot [Kho02], we call the bipartite instance *η-smooth* if, for every $r \in A$ and every pair of distinct labels $a, a' \in \Sigma_A$,

$$\Pr_{e \ni r}[\pi_e(a) = \pi_e(a')] \leq \eta,$$

where e is chosen uniformly among the edges incident to r [AKKV05, Section 4]. In the layered construction of [KPV14], the analogous condition is imposed for each vertex on its edges to the next layer [KPV14, Definition 3.7]. If the list at r has size t, then for each listed label a, its projection can be absent from the right list on at most a $(t-1)\eta$ fraction of incident edges, since cancellation is possible only after a collision with another listed label. By a union bound over all pairs, the fraction of incident edges with any collision is at most $\binom{t}{2}\eta$. Along every remaining edge, each nonzero left coefficient occurs alone in its projection sum, so the corresponding right coefficient is nonzero. For a Euclidean gap g, Alekhnovich et al. [AKKV05, Section 5] discard vertices whose lists have more than $200g^2$ labels. They fix an allowed label from each remaining left list and choose uniformly from each remaining right list, with $200\eta g^2 \leq 0.01$. The smooth instances of [Kho02] used in that reduction have size $n_{\varkappa}^{\Omega(1/\eta)}$. Since the output size M satisfies $\log M = \Omega((\log n_{\varkappa})/\eta)$, the gap is $g = O(\sqrt{\log M / \log n_{\varkappa}})$.

The parameters of Khot, Popat, and Vishnoi. Khot, Popat, and Vishnoi prove smoothness from agreement bounds for distinct low-degree polynomials [KPV14, Section 3.2]. For their layered construction, let n be the number of variables of the quadratic constraint satisfaction instance from which they reduce, and let $m = \log n$. Each hyperedge contains $2m + 3$ vertices. Writing θ for the soundness error measured over hyperedges, they prove a Hamming-weight gap h [KPV14, Definition 3.8, Theorem 3.10, and Claim 3.11] for sufficiently large m, subject to

$$h \leq m^{-3}\theta^{-1/(3m)}.$$

They take a field of size $2^{(\log n)^{4/\varepsilon}}$, soundness error θ polynomially small in the field size, and $\log h = (\log n)^{4/\varepsilon - 2}$ [KPV14, Section 3.4]. Since $\log M \leq (\log n)^{4/\varepsilon + 2}$, the gap is $2^{(\log M)^{1-\varepsilon}}$.

Our reduction without smoothness. In Construction 6.2, we keep the class equations, use one variable per accepting view $(V, \boldsymbol{a})$ instead of one per tuple $\boldsymbol{a}$, and replace the projection equations by the injective encoding of the restriction vectors $\boldsymbol{v}^{u,a}$ (Remark 6.3). Every view with a nonzero coefficient then has all its answers in the lists $\Lambda_u = \{a : \boldsymbol{v}^{u,a} \neq \boldsymbol{0}\}$, without a smoothness assumption. Using the relative distance of the image of ι, we bound the weighted average list size by the solution weight and then sample a proof from these lists, as in the proof of Theorem 6.4. We can choose any gap $\Gamma < \delta \cdot \mathtt{s}^{-1/\mathtt{q}}$. Without a smoothness requirement, we can use the PCPs constructed by unrolling the IOPs of [ACY23; MZ25] as in [ABCY22, Lemma 4.6], with $O(\log \log N)$ queries (Section 6.3).

AI Methodology

We began this work by studying a PCP-based approach to hardness with preprocessing, through which we could already improve upon prior results. We developed this reduction primarily without AI assistance (although the version in this paper is a simplified version of our approach, for which we used AI assistance). In discussing this approach with ChatGPT Sol 5.6 to investigate further extensions and limitations, we learned of the recent advances by OpenAI [Ope26, Chapter 7] and Song [Son26]. Starting from our PCP-based reduction, we developed the nearly optimal results presented here through an interactive discussion with the model. In addition, we used Claude Opus 5.5, Fable 5.1, and ChatGPT 6 Astra to assist with writing. The authors take full responsibility for the mathematical arguments, results, and presentation of this paper.

References

[ABCY22] Gal Arnon, Amey Bhangale, Alessandro Chiesa, and Eylon Yogev. "A Toolbox for Barriers on Interactive Oracle Proofs". In: *Proceedings of the 20th Theory of Cryptography Conference*. TCC '22. Full version: Cryptology ePrint Archive, Report 2022/790, `https://eprint.iacr.org/2022/790`. 2022, pp. 447–466.

[ABSS97] Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. "The Hardness of Approximate Optima in Lattices, Codes, and Systems of Linear Equations". In: *Journal of Computer and System Sciences* 54.2 (1997), pp. 317–331. DOI: `10.1006/jcss.1997.1472`.

[ACY23] Gal Arnon, Alessandro Chiesa, and Eylon Yogev. "IOPs with Inverse Polynomial Soundness Error". In: *64th IEEE Annual Symposium on Foundations of Computer Science , FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*. Full version: Cryptology ePrint Archive, Report 2023/1062, October 28, 2023 revision, `https://eprint.iacr.org/2023/1062`. IEEE, 2023, pp. 752–761.

[AKKV05] Mikhail Alekhnovich, Subhash A. Khot, Guy Kindler, and Nisheeth K. Vishnoi. "Hardness of Approximating the Closest Vector Problem with Pre-Processing". In: *46th Annual IEEE Symposium on Foundations of Computer Science*. 2005, pp. 216–225. DOI: `10.1109/SFCS.2005.40`.

[AKKV11] Mikhail Alekhnovich, Subhash A. Khot, Guy Kindler, and Nisheeth K. Vishnoi. "Hardness of approximating the closest vector problem with pre-processing". In: *Computational Complexity* 20 (2011), pp. 741–753. DOI: `10.1007/s00037-011-0031-3`.

[APY10] Noga Alon, Rina Panigrahy, and Sergey Yekhanin. "Deterministic approximation algorithms for the nearest codeword problem". In: *Dagstuhl Seminar Proceedings*. Vol. 09421. 2010, pp. 1–13. DOI: `10.4230/DagSemProc.09421.4`.

[AR05] Dorit Aharonov and Oded Regev. "Lattice problems in $\mathsf{NP} \cap \mathsf{coNP}$". In: *Journal of the ACM* 52.5 (2005), pp. 749–765. DOI: 10.1145/1089023.1089025. URL: https://cims.nyu.edu/~regev/papers/cvpconp.pdf.

[BGHSV06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. "Robust PCPs of Proximity, Shorter PCPs, and Applications to Coding". In: *SIAM Journal on Computing* 36.4 (2006), pp. 889–974.

[BK02] Piotr Berman and Marek Karpinski. "Approximating minimum unsatisfiability of linear equations". In: *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '02. 2002, pp. 514–516.

[BMT78] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. "On the inherent intractability of certain coding problems". In: *IEEE Transactions on Information Theory* 24.3 (1978), pp. 384–386. DOI: 10.1109/TIT.1978.1055873.

[Bab86] László Babai. "On Lovász' lattice reduction and the nearest lattice point problem". In: *Combinatorica* 6.1 (1986), pp. 1–13. DOI: 10.1007/BF02579403.

[CMS19] Alessandro Chiesa, Peter Manohar, and Igor Shinkar. "Probabilistic Checking Against Non-Signaling Strategies from Linearity Testing". In: *10th Innovations in Theoretical Computer Science Conference*. Vol. 124. Leibniz International Proceedings in Informatics (LIPIcs). 2019, 25:1–25:17. DOI: 10.4230/LIPIcs.ITCS.2019.25.

[DHK15] Irit Dinur, Prahladh Harsha, and Guy Kindler. "Polynomially Low Error PCPs with polyloglog n Queries via Modular Composition". In: *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*. STOC '15. 2015, pp. 267–276.

[DKRS03] Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. "Approximating CVP to Within Almost-Polynomial Factors is NP-Hard". In: *Combinatorica* 23.2 (2003), pp. 205–243. DOI: 10.1007/s00493-003-0019-y.

[DRSD14] Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. "On the closest vector problem with a distance guarantee". In: *IEEE 29th Conference on Computational Complexity*. 2014. arXiv: 1409.8063.

[EB81] Peter van Emde Boas. *Another NP-complete partition problem and the difficulty of computing short vectors in a lattice*. Tech. rep. 81-04. Mathematisch Instituut, Universiteit van Amsterdam, 1981.

[FM04] Uriel Feige and Daniele Micciancio. "The inapproximability of lattice and coding problems with preprocessing". In: *Journal of Computer and System Sciences* 69.1 (2004), pp. 45–67. DOI: 10.1016/j.jcss.2004.01.002.

[FS14] Szabolcs L. Fancsali and Péter Sziklai. *Higgledy-piggledy subspaces and uniform subspace designs*. 2014. arXiv: 1409.6227 [math.CO]. URL: https://arxiv.org/abs/1409.6227v1.

[FW26] Giacomo Fenzi and Guy Weissenberg. *Lean formalization of the hardness of CVP and NCP with preprocessing*. 2026. URL: https://github.com/WizardOfMenlo/preprocessing-approximations-lean.

[GRS25] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory*. Draft, August 26, 2025. 2025. URL: https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/web-coding-book.pdf.

[HS26a] Isaac M. Hair and Amit Sahai. *Polynomial-Factor Deterministic NP-Hardness for SVP in Every ℓ_p Norm with $p > 2$*. Version 3, August 19, 2026. 2026. arXiv: 2608.14529.

[HS26b] Isaac M. Hair and Amit Sahai. *SVP_p is Deterministically NP-Hard for all $p > 2$, Even to Approximate Within a Factor of $2^{\log^{1-\varepsilon} n}$*. Version 2, March 29, 2026. 2026. arXiv: 2511.04125. URL: https://arxiv.org/abs/2511.04125v2.

[IP01] Russell Impagliazzo and Ramamohan Paturi. "On the Complexity of k-SAT". In: *Journal of Computer and System Sciences* 62.2 (2001), pp. 367–375.

[Jus72] Jørn Justesen. "A class of constructive asymptotically good algebraic codes". In: *IEEE Transactions on Information Theory* 18.5 (1972), pp. 652–656. DOI: 10.1109/TIT.1972.1054893.

[KPV14] Subhash A. Khot, Preyas Popat, and Nisheeth K. Vishnoi. "Almost polynomial factor hardness for closest vector problem with preprocessing". In: *SIAM Journal on Computing* 43.3 (2014). Full version: arXiv:1109.2176v1 (2011), under the title $2^{\log^{1-\varepsilon} n}$ Hardness for Closest Vector Problem with Preprocessing, https://arxiv.org/abs/1109.2176v1, pp. 1184–1205. DOI: 10.1137/130919623.

[Kho02] Subhash Khot. "Hardness Results for Coloring 3-Colorable 3-Uniform Hypergraphs". In: *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*. FOCS '02. 2002, pp. 23–32. DOI: 10.1109/SFCS.2002.1181879.

[LLL82] Arjen K. Lenstra, Jr. Lenstra Hendrik W., and László Lovász. "Factoring polynomials with rational coefficients". In: *Mathematische Annalen* 261.4 (1982), pp. 515–534. DOI: 10.1007/BF01457454.

[LN97] Rudolf Lidl and Harald Niederreiter. *Finite Fields*. Second Edition. Cambridge University Press, 1997.

[LS71] John Leech and N. J. A. Sloane. "Sphere Packings and Error-Correcting Codes". In: *Canadian Journal of Mathematics* 23.4 (1971), pp. 718–745. DOI: 10.4153/CJM-1971-081-3.

[MV13] Daniele Micciancio and Panagiotis Voulgaris. "A Deterministic Single Exponential Time Algorithm for Most Lattice Problems Based on Voronoi Cell Computations". In: *SIAM Journal on Computing* 42.3 (2013), pp. 1364–1391.

[MZ25] Dor Minzer and Kai Zhe Zheng. "Improved Round-by-round Soundness IOPs via Reed-Muller Codes". In: *Proceedings of the 66th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '25. 2025.

[Ope26] OpenAI. *Ten Advances in Mathematics and Theoretical Computer Science*. Tech. rep. Chapter 7, "$n^{1/400}$-Hardness of the Euclidean Closest Vector Problem," pp. 183–218. 2026. URL: https://cdn.openai.com/pdf/ten-proofs-oai.pdf.

[Pei08] Chris Peikert. "Limits on the hardness of lattice problems in ℓ_p norms". In: *Computational Complexity* 17.2 (2008), pp. 300–351. DOI: 10.1007/s00037-008-0251-3.

[Reg04] Oded Regev. "Improved inapproximability of lattice and coding problems with preprocessing". In: *IEEE Transactions on Information Theory* 50.9 (2004), pp. 2031–2037.

[Sch87] Claus-Peter Schnorr. "A hierarchy of polynomial time lattice basis reduction algorithms". In: *Theoretical Computer Science* 53.2–3 (1987), pp. 201–224. DOI: 10.1016/0304-3975(87)90064-8.

[Son26] Zhao Song. *Hardness of Euclidean Closest Vector Within $n^{1/2-\epsilon}$ and Binary Nearest Codeword Within $n^{1-\epsilon}$*. Preprints.org. Version 2. Sept. 2, 2026. DOI: 10.20944/preprints202608.0796.v2. URL: https://www.preprints.org/manuscript/202608.0796.