

Decoding Balanced Linear Codes With Preprocessing

Andrej Bogdanov* Rohit Chatterjee† Yunqi Li‡ Prashant Nalini Vasudevan§

Abstract

Prange’s information set algorithm is a decoding algorithm for arbitrary linear codes. It decodes corrupted codewords of any $\mathbb{F}_2$ -linear code C of message length n up to relative error rate $O(\log n/n)$ in $\text{poly}(n)$ time. We show that the error rate can be improved to $O((\log n)^2/n)$, provided: (1) the decoder has access to a polynomial-length advice string that depends on C only, and (2) C is $n^{-\Omega(1)}$ -balanced.

As a consequence we improve the error tolerance in decoding random linear codes if inefficient preprocessing of the code is allowed. This reveals potential vulnerabilities in cryptographic applications of Learning Noisy Parities with low noise rate.

Our main technical result is that the Hamming weight of Hw , where H is a random sample of *short dual* codewords, measures the proximity of a word w to the code in the regime of interest. Given such H as advice, our algorithm corrects errors by locally minimizing this measure. We show that for most codes, the error rate tolerated by our decoder is asymptotically optimal among all algorithms whose decision is based on thresholding Hw for an arbitrary polynomial-size advice matrix H .

Contents

1	Introduction	2
2	Concepts and notation	6
2.1	Notations	6
2.2	Nearest Codeword Problem	6
2.3	Random Linear Codes	7
3	Algorithm with Preprocessing	7
3.1	Decision Algorithm	7
3.2	Search Algorithm	12
3.3	Search to Decision Reduction	15
4	On the Optimality of Our Algorithms	16
4.1	Limits of Threshold Distinguishers	16
4.2	Limits of Interval Distinguishers	17
4.3	Optimality of our Distribution	18

*abogdano@uottawa.ca. University of Ottawa.

†rochat@nus.edu.sg. Department of Computer Science, National University of Singapore.

‡yunqili@comp.nus.edu.sg. Department of Computer Science, National University of Singapore.

§prashvas@nus.edu.sg. Department of Computer Science, National University of Singapore.

1 Introduction

Decoding corrupted codewords is a challenging algorithmic task. Its complexity is far from being understood. The choice of the code is crucial in this context. Crafted codes like Reed-Solomon can be decoded optimally, all the way up to half the minimum distance. In contrast, for random linear codes no known algorithm can substantially outperform brute force.

What is the algorithmic complexity of decoding a generic code of given rate and distance? This question is captured by the *nearest codeword problem* (NCP).

NCP takes two inputs: A linear code C of message length n and blocklength $m \geq n$, and a target word w . (We restrict attention to codes over the binary alphabet.) In the *decision* version, the goal is to distinguish whether w is close to C or far from C . In the *estimation* version, the goal is to calculate the distance from w to C up to some approximation factor. The *search* version asks for the codeword closest to w under the promise that w is sufficiently close to C .

Of the three variants, the search one is the hardest. Decoding up to relative distance η yields an estimation algorithm with approximation ratio at most $1/\eta$. It also distinguishes those words that are η -close to C from all others, provided that η is within the decoding radius.

The randomized “information set decoding” algorithm of Prange [Pra62] is essentially still the best asymptotically efficient one for NCP.¹ It finds the codeword closest to w in expected time $\exp(O(\eta n))$ under the promise that w is within relative distance η of C . Setting η to $O((\log m)/n)$ yields a polynomial-time algorithm with approximation ratio $n/c \log m$ for any constant c .

No asymptotic improvement to Prange’s algorithm is known even in the average-case, i.e., for random linear codes C . In the vanishing rate regime $m \gg n$, which will be of our main interest, most such codes are $O(\sqrt{n/m})$ -balanced: all codewords have Hamming weight in the range $1/2 \pm \Theta(\sqrt{n/m})$, which is close to the best possible [Wel74, Lev83, Alo03, MRRW06, FT05].

On the hardness side, there is no polynomial-time algorithm that approximates the distance to the nearest codeword within an arbitrary constant factor unless P equals NP, or within a $O(2^{\log^{1-\epsilon} n})$ factor unless NP has quasi-polynomial-time algorithms [ABSS93, BGR25]. In addition, there is no sub-exponential-time approximation within *some* constant factor under the Exponential-Time Hypothesis [BHI⁺24].

Does this hardness stem from the choice of the code C or of the corrupted codeword w ? The nearest codeword problem *with pre-processing* is useful for studying their relative contribution. In this problem, there is a pre-processing phase, in which the algorithm sees only the code C and produces a bounded advice string H that depends on C . There is no restriction on the complexity of computing H . In the online phase, the algorithm reads w and produces its answer. The estimation variant of NCP with pre-processing is known to be NP-hard to approximate to within a fixed constant factor [FM04, Reg04], and SETH-hard to solve exactly in $2^{(1-\epsilon)n}$ time [SV19].

Our results

Our main result (Theorem 3.2) is that pre-processing improves upon Prange’s algorithm for highly balanced codes. Under the promise that C is β -balanced, the Nearest Codeword at relative distance η can be found in time $\text{poly}(m) \cdot \exp[O(\eta n / \log(1/\max\{\beta, \eta\}))]$ using advice of comparable length.

When the code is $n^{-\Omega(1)}$ -balanced, we obtain a polynomial-time algorithm with preprocessing for decoding near-codewords at relative distance $(\log n)^2/n$. In contrast, Prange’s algorithm would take quasipolynomial time to decode at this distance.

¹It was rediscovered by Berman and Karpinski [BK02] as an approximation algorithm, and partially derandomized by Alon, Panigrahy, and Yekhanin [APY09]. There have also been multiple optimizations that improve on it [BJMM12, MO15, BM18], but these ultimately rely on similar principles and have similar asymptotic behavior.

The balance assumption is satisfied by most linear codes of rate $n^{-\Omega(1)}$ (as n grows), as well as by many explicit constructions [NN93, AGHP92, ABN⁺92, BATS13, TS17]. Is it necessary? In Theorem 4.1 we show that it is for a certain class of algorithms with preprocessing.

Specifically, the advice in Algorithm 2 that solves the decision version of NCP is a matrix H that depends on the code. The algorithm decides proximity to the code based on the Hamming weight of Hw (modulo 2). This weight is small if w is close to the code and large if it is far.

Our Corollary 4.2 shows that for most codes C , any algorithm whose decision is the threshold of Hw for *some* matrix H cannot separate words that are $\omega((\log^2 n)/n)$ -close to the code from those that are $(1/2 - 3\sqrt{n/m})$ -far, provided H has size polynomial in n . Thus even under almost extremal assumptions on the balance of C , the decoding distance cannot be improved.

Theorem 4.1 is restrictive in assuming that proximity to the code is decided by a threshold of parities. In Corollary 4.3 we show that the same conclusion extends to a polynomial threshold function of parities. While further generalizations are an interesting open question, we cannot expect to fully eliminate the complexity assumption on the distinguisher in an unconditional lower bound. For instance, the possibility that every code can be decoded by a polynomial-size *threshold of thresholds* of parities is consistent with the current sorry state of computational complexity theory.

Ideas and techniques

The algorithm Our advice H consists of independent samples from a carefully chosen distribution D on length- m binary strings, conditioned on each sample being a dual codeword in $C^\perp$. We denote (the density of) this conditional distribution by D_C .

A sample of D is a sum $h = u_1 + \dots + u_\ell$ of independent standard basis (one-hot) vectors in $\mathbb{F}_2^m$ chosen with replacement. When ℓ is even, the all-zero codeword is in the support of D and the conditional distribution D_C is well-defined.

The distribution D_C favors light dual codewords, with the weight controlled by the parameter ℓ . When ℓ is slightly larger than the distance of $C^\perp$, we expect D_C to be concentrated on dual codewords of weight close to the minimum distance of $C^\perp$.

In Lemma 3.4 we show that for a suitable choice of ℓ the Fourier transform

$$\hat{D}_C(w) = \mathbb{E}_{h \sim D_C} \left[(-1)^{\langle h, w \rangle} \right] = \mathbb{E}_{h \sim D} \left[(-1)^{\langle h, w \rangle} \mid h \in C^\perp \right]$$

distinguishes close from far codewords: $\hat{D}_C(w)$ is bounded away from zero when w is close to the code and extremely close to zero when it is at relative distance $1/2 - n^{-\Omega(1)}$ from all codewords and their complements.

Our Algorithm 2 decides proximity based on an empirical estimate of $\hat{D}_C(w)$. By a large deviation bound, the efficiently computable function

$$\tilde{D}_H(w) = \mathbb{E}_{\text{row } h \text{ of } H} \left[(-1)^{\langle h, w \rangle} \right]$$

uniformly approximates $\hat{D}_C(w)$ for at least one choice (in fact, for most choices) of H . This H is the advice provided to the algorithm.

Our search algorithm (Algorithm 3) is based on one additional idea. For sufficiently light w , $\hat{D}_C(w)$ is monotonically decreasing. Thus the 1-entries in w are precisely those whose flip results in an increase of $\hat{D}_C(w)$, or of its proxy $\tilde{D}_H(w)$. As $\hat{D}_C(w)$ is periodic modulo C , the errors in w can be corrected by flipping those bits w_i that increase the value of $\tilde{D}_H(w)$.

Lower bound To prove Theorem 4.1 we show that for any sufficiently short advice H , the function $\tilde{D}_H$ cannot distinguish *random* codewords $w = Cx + e$ corrupted by random noise e exceeding the noise rate supported by our algorithm from truly random strings w in expectation. The former are typically close to the code, while the latter are very far. Thus it is impossible for a threshold distinguisher to tell them apart.

As a consequence, our algorithms cannot be improved by optimizing the advice H , and in particular by a different choice of distribution D . The key property of D used in our analysis is that the Fourier coefficients $\hat{D}(w)$ are noticeable when w is light (its relative Hamming weight is $O((\log n)^2/n)$) and vanishing when w is heavy (see (6)). In words, D is noticeably biased under all light linear tests but almost unbiased under all heavy linear tests.

Lemma 4.5 states that this is optimal: If D is noticeably biased under all light tests of relative Hamming weight $\omega((\log n)^2/n)$ then it must be somewhat biased under a heavy test of weight $(1 \pm n^{-\Omega(1)})/2$. This consequence may be of interest in pseudorandomness, specifically the study of small-biased sets. Lemma 4.5 says that one cannot ignore the contribution of light tests, which are generally easy to fool.

Lemma 4.5 is a consequence of Theorem 4.1 and our proof of Proposition 3.1. We find this argument somewhat unsatisfying as it detours into coding theory and algorithms to establish a statement about distributions. In Section 4.3 we provide an alternative direct analytical proof of it.

Parallels and tangents to decoding in lattices

The closest vector problem for lattices is the analogue of the nearest codeword problem for codes. Aharonov and Regev [AR05] gave an efficient algorithm with preprocessing that approximates the distance to any lattice of dimension n up to a $\sqrt{n/\log n}$ approximation factor. Liu, Lyubashevsky, and Micciancio [LLM06] gave a decoding algorithm with preprocessing for lattices under the promise that the distance between the target vector w and the lattice is within $O(\sqrt{\log n/n})$ times the length of its shortest vector.

There are strong parallels between their algorithms and ours. In the lattice setting, the analogue of our distribution D is the multivariate standard normal, H is a matrix of samples from D conditioned on membership in the dual lattice, and the proximity decision is made based on the norm of the vector Hw .

One important difference is that the algorithm of Aharonov and Regev works without any assumption on the length of the dual shortest vector, which is the analogue of dual distance for lattices. As we show in Theorem 4.1, a bounded dual distance assumption is necessary in the code setting.² For codes, the length of the advice must be exponential in the product of the relative proximity parameter η and the dual distance d .

Interestingly, Aharonov and Regev's $\sqrt{n/\log n}$ approximation ratio for the closest vector problem with preprocessing is matched by Goldreich and Goldwasser's statistical zero-knowledge proof [GG00] for the same problem. Analogously, our $n/(\log n)^2$ approximation ratio for the nearest codeword problem for balanced codes with preprocessing is matched by the statistical zero-knowledge proof of Brakerski et al. [BLVW19] for the same problem. As in the lattice setting, the two algorithms are quite different. (At a technical level, the sampling of short combinations of codewords in the code and its dual, respectively, and the Fourier analysis of this process is a commonality.)

One intriguing open question is whether Aharonov and Regev's $\sqrt{n}$ -approximate noninteractive (coNP) refutation for closest vector in lattices has an analogue for balanced codes.

²This still leaves open whether the balance assumption used in our algorithms can be replaced by a weaker one like bounded dual distance.

Implications on learning noisy parities and cryptography

Learning Noisy Parities (LPN) [BFKL93] is an average-case variant of NCP. The code C is random, and the decision problem is to distinguish a randomly corrupted random codeword from a uniformly random string. The hardness of LPN has found several uses in cryptography, including public-key encryption [Ale03], collision-resistant hashing [AHI⁺17, BLVW19, YZW⁺19], and more [BLSV18, BF22, AMR25]. LPN-based schemes are attractive for their computational simplicity and plausible post-quantum security.

Some of these constructions [BLVW19, BLSV18, BF22, YZW⁺19, AMR25] assume security of LPN at noise rate $O((\log n)^2/n)$. Our Algorithm 3 does not render them insecure owing to its inefficient preprocessing phase. However, it highlights potential concerns in settings where the code may be used as a public parameter, like a hash key as in [AMR25] or a common random string as in [BLSV18]. Such schemes may invoke security of some LPN instances where the code C is available *long term*, and not generated ephemerally by running some algorithm in the scheme. This would allow an adversary to mount a longer duration attack to calculate the advice H and then apply Algorithm 3 to break the scheme. Our results also rule out non-uniform security: If the adversary is only bounded in size but may otherwise depend arbitrarily on C then noise of rate $O((\log n)^2/n)$ is insecure.

This non-uniform security model is in particular captured by the *linear test framework* of Boyle et al. [BCG⁺20, CRR21, BCG⁺22]. They quantify the insecurity of a code C by the maximum bias

$$\max_{h \in \mathbb{F}_2^m} \mathbb{E} \left[(-1)^{\langle h, w \rangle} \right],$$

where w is a randomly corrupted random codeword of C . Couteau et al. [CRR21, Lemma 6] observe that the insecurity becomes negligible when the dual distance of C is sufficiently large. The same logic underlies the proof of our lower bound Theorem 4.1.

Other related work

Information set decoding algorithms Prange’s algorithm has been applied to cryptanalyze code-based schemes. There are several concrete improvements in the high-distance regime. None of them are asymptotic improvements in the exponent. Specifically, the expected running time of Prange’s algorithm on worst-case inputs (close to half the minimum distance) is asymptotically dominated by 2^{cn} with $c \approx 0.058$. In applications, decoding at lower distances is reduced to decoding near half the minimum distance on a code of lower dimension.

There has been considerable effort in improving this constant [MMT11, BJMM12, MO15, BM17]. Both and May [BM18] obtain $c \approx 0.047$. Bernstein [Ber10] obtains a quadratic *quantum* speedup of Prange’s algorithm, and Kachigar and Tillich [KT17] are able to get a quantum speedup for the approaches in [MMT11, BJMM12]. Ducas, Esser, Etinski and Kirshanova [DEEK24] provide further lower-order improvements motivated by cryptographically relevant concrete parameter settings building on the approach of [MMT11].

Worst-case to average-case reductions for balanced NCP The work of [BLVW19] also gives a reduction from the hardness of NCP for balanced codes to that of LPN, albeit with extreme parameters. Yu and Zhang [YZ21] somewhat improve and generalize this result to another restricted kind of codes they call *independent* (balanced) codes. Debris-Alazard and Resch [DAR25] give a reduction from worst case NCP on balanced codes (with inverse polynomial noise rate) to average case NCP (with noise rate inverse polynomially close to half).

2 Concepts and notation

2.1 Notations

Linear codes. A linear code $\mathcal{C}$ is a collection of vectors $\{C \cdot x : x \in \mathbb{F}_2^n\}$ with C being the generator matrix in $\mathbb{F}_2^{m \times n}$. By default, we assume C has full rank and $m > n$. Consider any linear code $\mathcal{C}$ with a generator matrix C , there exists a parity-check matrix $C^\perp \in \mathbb{F}_2^{m \times (m-n)}$, such that $(C^\perp)^T C = 0^{(m-n) \times n}$. $C^\perp$ also generates the dual code of $\mathcal{C}$, denoted by $\mathcal{C}^\perp = \{h \in \mathbb{F}_2^m : \langle h, v \rangle = 0, \forall v \in \mathcal{C}\}$.

Weight and distance. We denote the Hamming weight of a vector $v \in \mathbb{F}_2^m$ by $\text{wt}(v)$. For any code $\mathcal{C} \subseteq \mathbb{F}_2^m$ and vector $w \in \mathbb{F}_2^m$, let $\text{dist}(\mathcal{C}, w) = \min_{v \in \mathcal{C}} \text{wt}(v + w)$, indicate the minimum distance of w to a code $\mathcal{C}$. With these notations, we are ready to define the notion of *balanced codeword* and to characterize the distance of a vector from a linear code.

Definition 2.1 (Balanced codeword). For $\beta \in [0, 1]$, a length- m vector w is called β -balanced if $\text{wt}(w) \in \frac{1}{2}(1 \pm \beta)m$. Correspondingly, a code $\mathcal{C}$ is said to be β -balanced if every non-zero codeword is β -balanced.

Definition 2.2. For $\eta, \beta \in [0, 1]$, w is called η -close to a code $\mathcal{C}$ if $\text{dist}(\mathcal{C}, w) \leq \eta m$; w is called β -separated from the code $\mathcal{C}$, if $(w + v)$ is β -balanced for every $v \in \mathcal{C}$.

Distributions. We identify discrete distributions D with their probability mass functions, i.e., $D(x)$ is the probability that x is the outcome when sampling from D .

2.2 Nearest Codeword Problem

The search version of *nearest codeword problem* is defined as follows:

Definition 2.3 (Nearest Codeword Problem). For $m, n \in \mathbb{N}$, $0 < \eta < 1$ with $C \in \mathbb{F}_2^{m \times n}$, $w \in \mathbb{F}_2^m$, given input (C, w) with the promise that $\text{dist}(\mathcal{C}, w) < \eta m$, the (search) nearest codeword problem NCP_η is to find s , such that $s \in \arg \min_{x \in \mathbb{F}_2^n} \text{dist}(C \cdot x, w)$.

Beyond this general definition, we introduce a variant, called *balanced nearest codeword problem*, in which the code is restricted to meet the balance property.

Definition 2.4 (Balanced Nearest Codeword Problem). For $m, n \in \mathbb{N}$, $0 < \beta, \eta < 1/6$, consider $C \in \mathbb{F}_2^{m \times n}$, $w \in \mathbb{F}_2^m$, given input (C, w) with the promise that $\mathcal{C}$ is a β -balanced code and $\text{dist}(\mathcal{C}, w) < \eta m$, the (search) balanced nearest codeword problem $\text{BNCP}_{\beta, \eta}$ is to find $\arg \min_{x \in \mathbb{F}_2^n} \text{dist}(C \cdot x, w)$.

We note that when $\beta, \eta < 1/6$, with the promise that w is η -close to a β -balanced code $\mathcal{C}$, there must be a unique close codeword. The *decisional balanced nearest codeword problem* is given below. For simplicity, we use a single parameter β to capture both the balance of the code and the separation of the NO instances.

Definition 2.5 (Decisional Balanced Nearest Codeword Problem). For $m, n \in \mathbb{N}$, $0 < \beta, \eta < 1/6$, consider $C \in \mathbb{F}_2^{m \times n}$, $w \in \mathbb{F}_2^m$, given input (C, w) with the promise that $\mathcal{C}$ is a β -balanced code, the *decisional balanced nearest codeword problem* $\text{DBNCP}_{\beta, \eta}$ is to decide between the following two cases:

$$\begin{aligned} \text{YES} &= \{(C, w) : w \text{ is } \eta\text{-close to the code } \mathcal{C}\}; \\ \text{NO} &= \{(C, w) : w \text{ is } \beta\text{-separated from the code } \mathcal{C}\}. \end{aligned}$$

2.3 Random Linear Codes

A random linear code is a code specified by a uniformly random choice of generator matrix $C \in \mathbb{F}_2^{m \times n}$. By the Gilbert-Varshamov bound, a random linear code is $3\sqrt{n/m}$ -balanced except with probability $2^{-\Omega(n)}$.

We reproduce this argument. As each nonzero codeword of a random linear code is random, by Hoeffding's inequality it is β -balanced except with probability $2\exp(-\beta^2 m/2)$. By a union bound, the probability that there exists an unbalanced codeword is then at most

$$(2^n - 1) \cdot 2e^{-\beta^2 m/2} < 2^{n-\beta^2 m/2+1}. \quad (1)$$

which is $2^{-\Omega(n)}$ when $\beta = 3\sqrt{n/m}$.

3 Algorithm with Preprocessing

Our main result is Theorem 3.2, the search algorithm with preprocessing for the balanced codeword problem. As the algorithm for the decisional problem DBNCP is easier to describe and contains the main idea, we describe it first and prove its correctness in Proposition 3.1.

Both of these algorithms involve preprocessing. An algorithm with preprocessing for NCP is one that, in addition to the instance (C, w) , is also given an advice string that is a function of C (but not of w). This advice is not required to be efficiently computable.

In the following statements n and m are the message length and blocklength of the code, respectively.

Proposition 3.1. *Assuming $0 < \beta, \eta < 1/6$, there is an algorithm with preprocessing for DBNCP $_{\beta, \eta}$ with both advice size and running time $m^2 \cdot \exp[O(\eta n / \log(1/\beta))]$.*

Theorem 3.2. *Assuming $1/m \leq \beta, \eta < 1/8$, there is an algorithm with preprocessing for BNCP $_{\beta, \eta}$ with both advice size and running time $(m^4 \log^2(1/\alpha)/n^2) \cdot \exp[O(\eta n / \log(1/\alpha))]$, where $\alpha = \beta + 2\eta$.*

In particular, when $\beta = n^{-\Omega(1)}$ and $\eta = O(\log^2 n/n)$, Algorithm 3 (which proves Theorem 3.2) runs in polynomial time.

Corollary 3.3. *When $\beta = n^{-\Omega(1)}$, $\eta = O(\log^2 n/n)$, there is an algorithm with preprocessing for BNCP $_{\beta, \eta}$ with advice size and running time polynomial in n .*

Since the decisional variant DBNCP reduces to BNCP, Theorem 3.2 is effectively more general than Proposition 3.1. There is a gap in complexities in the regime $\eta \gg \beta$. This gap owes to our usage of β to represent both the balance and separation parameters in DBNCP.

In Section 3.3 we present an alternative proof of Theorem 3.2 (with slightly worse advice size) by reduction to Proposition 3.1. While the resulting algorithm is less natural, this argument explains the deterioration from $\log 1/\beta$ in Proposition 3.1 to $\log 1/(\beta + 2\eta)$ in Theorem 3.2.

3.1 Decision Algorithm

In this section, we present the decision algorithm and prove Proposition 3.1. The algorithm is as follows, where Pre is the preprocessing stage, and Decide is the algorithm itself. The algorithms are defined by parameters that we will set later in this section, based on the values of n , m , β , and η .

Algorithm 1: Preprocessing with parameters (ℓ, N)

Input: Generator matrix $C \in \mathbb{F}_2^{m \times n}$

Output: Advice $(h_1, \dots, h_N)$

Sampling $\text{Samp}(C)$:

 Sample length- m unit vectors $u_1, \dots, u_\ell$ conditioned on $u_1 + \dots + u_\ell \in \mathcal{C}^\perp$
 independently randomly;
 return $u_1 + \dots + u_\ell$;

Preprocessing $\text{Pre}(C)$:

for $i \in \{1, \dots, N\}$ **do**
 Sample $h_i \leftarrow \text{Samp}(C)$;
 return $h_1, \dots, h_N$;

Two Important Distributions Fix any n, m , and β as in the theorem statement, and any β -balanced code $C \in \mathbb{F}_2^{m \times n}$. Set the parameter $\ell \in \mathbb{N}$ to some fixed value (to be determined later). We define two distributions sampled as follows over the codespace, which will be crucial to the working of our algorithm:

- Distribution D : sample length- m unit vectors $u_1, \dots, u_\ell$ uniformly randomly, then output $u_1 + \dots + u_\ell$.
- Distribution D_C : sample $h \leftarrow D$ conditioned on $h \in \mathcal{C}^\perp$.

We will set ℓ to be an even integer (among other restrictions), so that the distribution D_C is well-defined. This is guaranteed by the fact that the zero vector 0^m lies in the dual space of any code and occurs with non-zero probability under the distribution D (so D_C has nonempty support).

In the *preprocessing* stage Pre , the algorithm samples vectors $(h_1, \dots, h_N)$ from distribution D_C , for $N \in \mathbb{N}$, which we refer to as advice. We note that since the preprocessing stage is allowed to be inefficient, it suffices to show the existence of good advice applicable to all inputs w (which implies that it can be found inefficiently). We present a uniform method for obtaining such advice, that succeeds with high probability, which more than meets this bar.

The *decision algorithm* Decide performs as follows: it takes the vector w and the advice $(h_1, \dots, h_N)$ associated with the code $\mathcal{C}$ as input, and compares the value $\sum_i (-1)^{\langle h_i, w \rangle}$ with a hard-coded threshold t to decide whether the vector is close to or separated from the code $\mathcal{C}$.

Algorithm 2: Decision algorithm with parameters (N, t)

Input: Vector $w \in \mathbb{F}_2^m$ and advice $(h_1, \dots, h_N)$

Output: YES or NO, indicating the vector w is closed to or separated from the code $\mathcal{C}$

Algorithm $\text{Decide}(w; h_1, \dots, h_N)$:

if $\sum_i (-1)^{\langle h_i, w \rangle} > t \cdot N$ **then**
 return YES;
 return NO;

To prove correctness we analyze the Fourier transform $\hat{D}_C(w) = \mathbb{E}(-1)^{\langle h, w \rangle}$ and show that it is a good measure of distance to the code (Lemma 3.4).

Fourier coefficient of D_C . We describe the probability mass function of D_C in terms of that of D , which is

$$D_C(h) = \frac{\mathbf{1}[h \in \mathcal{C}^\perp]}{Z_C} \cdot D(h),$$

where Z_C is a normalization factor that satisfies

$$Z_C = \sum_{h \in \mathcal{C}^\perp} D(h). \quad (2)$$

The Fourier coefficient of D is defined by

$$\hat{D}(w) = \mathbb{E}_{h \leftarrow D} \left[(-1)^{\langle h, w \rangle} \right] = \sum_{h \in \mathbb{F}_2^m} D(h) \cdot (-1)^{\langle h, w \rangle}.$$

For the distribution D_C , its Fourier coefficient $\hat{D}_C$ can be represented by $\hat{D}$,

$$\begin{aligned} \hat{D}_C(w) &= \mathbb{E}_{h \leftarrow D_C} \left[(-1)^{\langle h, w \rangle} \right] \\ &= \sum_{h \in \mathbb{F}_2^m} D_C(h) \cdot (-1)^{\langle h, w \rangle} \\ &= \frac{1}{Z_C} \sum_{h \in \mathbb{F}_2^m} \mathbf{1}[h \in \mathcal{C}^\perp] D(h) \cdot (-1)^{\langle h, w \rangle} \\ &= \frac{1}{Z_C \cdot 2^n} \sum_{v \in \mathcal{C}} \sum_{h \in \mathbb{F}_2^m} D(h) \cdot (-1)^{\langle h, w+v \rangle} \\ &= \frac{1}{Z_C \cdot 2^n} \sum_{v \in \mathcal{C}} \hat{D}(w+v), \end{aligned} \quad (3)$$

where the fourth equality follows the fact that

$$\mathbf{1}[h \in \mathcal{C}^\perp] = \frac{1}{2^n} \sum_{v \in \mathcal{C}} (-1)^{\langle h, v \rangle},$$

since if $h \in \mathcal{C}^\perp$, for all $v \in \mathcal{C}$, $(-1)^{\langle h, v \rangle} = 1$; if $h \notin \mathcal{C}^\perp$, there exists $v' \in \mathcal{C}$, such that $\langle h, v' \rangle = 1$, then

$$\sum_{v \in \mathcal{C}} (-1)^{\langle h, v \rangle} = \frac{1}{2} \sum_{v \in \mathcal{C}} \left((-1)^{\langle h, v \rangle} + (-1)^{\langle h, v+v' \rangle} \right) = 0.$$

Since $\hat{D}_C(0^m) = 1$, the normalization factor is

$$Z_C = \frac{1}{2^n} \sum_{v \in \mathcal{C}} \hat{D}(v). \quad (4)$$

Combining (3) and (4), we obtain that

$$\hat{D}_C(w) = \mathbb{E}_{h \leftarrow D_C} \left[(-1)^{\langle h, w \rangle} \right] = \frac{\sum_{v \in \mathcal{C}} \hat{D}(v+w)}{\sum_{v \in \mathcal{C}} \hat{D}(v)} \quad (5)$$

Fourier coefficient of D . Recall that the distribution D outputs the XOR of ℓ independent random unit vectors over $\mathbb{F}_2^m$,

$$\hat{D}(w) = \mathbb{E}_{h \leftarrow D} [(-1)^{\langle h, w \rangle}] = \mathbb{E}_{u_1, \dots, u_\ell} \left[\prod_i (-1)^{\langle u_i, w \rangle} \right] = \prod_i \mathbb{E}_{u_i} [(-1)^{\langle u_i, w \rangle}] = \left(1 - 2 \cdot \frac{\text{wt}(w)}{m} \right)^\ell. \quad (6)$$

The last equality is obtained by

$$\mathbb{E}_{u_i} [(-1)^{\langle u_i, w \rangle}] = \frac{m - \text{wt}(w)}{m} \cdot 1 + \frac{\text{wt}(w)}{m} \cdot (-1) = 1 - 2 \cdot \frac{\text{wt}(w)}{m}.$$

We now turn to establishing guarantees for our algorithm. We start by showing that the function $\mathbb{E}_{h \leftarrow D_C} [(-1)^{\langle h, w \rangle}]$ serves as an ‘ideal’ distinguishing function between inputs η -close to the code, and those β -separated from the code (hence helping us decide instances of the DBNCP problem). Note that this function is not necessarily efficient to compute, but we will later show that this is exactly what our preprocessing step will help us handle.

Lemma 3.4. *For $m, n \in \mathbb{N}$, $0 < \beta, \eta < 1/6$. Given any $\text{DBNCP}_{m, \beta, \eta}$ instance (C, w) , letting $\ell = 2 \cdot \lceil n / \log(1/\beta) \rceil$ be an even integer, we have that:*

If w is η -close to the code $\mathcal{C}$,

$$\mathbb{E}_{h \leftarrow D_C} [(-1)^{\langle h, w \rangle}] > \frac{1}{2} (1 - 2\eta)^{2(n/\log(1/\beta)+1)};$$

If w is β -separated from the code $\mathcal{C}$,

$$\mathbb{E}_{h \leftarrow D_C} [(-1)^{\langle h, w \rangle}] < 2^{-n}.$$

Proof. Consider any $\text{DBNCP}_{m, \beta, \eta}$ instance (C, w) , where C is the generator matrix of a β -balanced code $\mathcal{C}$, which means that for $v = C \cdot x$ with all non-zero $x \in \mathbb{F}_2^n$, the Hamming weight $\text{wt}(v) \in \frac{1}{2}(1 \pm \beta)m$, thus $0 \leq \hat{D}(v) \leq \beta^\ell$ according to (6). As $\hat{D}(0^m) = 1$, we have

$$1 \leq \sum_{v \in \mathcal{C}} \hat{D}(v) \leq 1 + (2^n - 1) \cdot \beta^\ell. \quad (7)$$

When $\ell = 2 \cdot \lceil n / \log(1/\beta) \rceil$,

$$\sum_{v \in \mathcal{C}} \hat{D}(v) \leq 1 + (2^n - 1) \cdot \beta^\ell < 1 + 2^n \cdot \beta^{2n/\log(1/\beta)} = 1 + 2^{-n}. \quad (8)$$

Suppose that (C, w) is a YES instance, which implies w is η -close to the code $\mathcal{C}$. Without loss of generality, assume $w = C \cdot x + e$ with $\text{wt}(e) \leq \eta m$,

$$\mathbb{E}_{h \leftarrow D_C} [(-1)^{\langle h, w \rangle}] = \frac{\sum_{v \in \mathcal{C}} \hat{D}(v + w)}{\sum_{v \in \mathcal{C}} \hat{D}(v)} > \frac{\hat{D}(e)}{1 + 2^{-n}} \geq \frac{(1 - 2\eta)^\ell}{1 + 2^{-n}} > \frac{1}{2} (1 - 2\eta)^{2(n/\log(1/\beta)+1)},$$

where the first inequality follows (8) and the positivity of the Fourier coefficients; the last inequality holds for all sufficiently large n . When $\beta, \eta < 1/6$, $(1 - 2\eta)^{2n/\log(1/\beta)} \gg 2^{-n}$.

For w being a NO instance, for all $v \in \mathcal{C}$, $\text{wt}(v + w) \in \frac{1}{2}(1 \pm \beta)m$, thus $\hat{D}(v + w) \leq \beta^\ell$,

$$\mathbb{E}_{h \leftarrow D_C} [(-1)^{\langle h, w \rangle}] = \frac{\sum_{v \in \mathcal{C}} \hat{D}(v + w)}{\sum_{v \in \mathcal{C}} \hat{D}(v)} \leq \sum_{v \in \mathcal{C}} \hat{D}(v + w) \leq 2^n \cdot \beta^\ell < 2^{-n}.$$

The first inequality follows the lower bound given by (7). □

Quality of advice. In the following, we show that with high probability the preprocessing stage outputs *good* advice $(h_1, \dots, h_N) \in (\mathbb{F}_2^m)^N$: i.e., advice on which the algorithm **Decide** yields the correct output on all inputs w that satisfy the promise (for some settings of parameters (ℓ, N, t) depending on n, m, β, η). Informally, this will be advice for which inputs that are close to the code lead to an evaluation by **Decide** to a relatively high value, while inputs that are far from the code in turn lead to a significantly lower evaluation.

Recall that Lemma 3.4 shows that the expected value $\mathbb{E}_h [(-1)^{\langle h, w \rangle}]$, for h being sampled from D_C , serves as an ideal test in the sense that it is large if the vector is close to the code $\mathcal{C}$; and is small when the vector is far. Using the advice, Algorithm 2 then essentially estimates this expectation using the N samples in the advice, namely $(h_1, \dots, h_N)$. Since h_i s are independent samples, standard concentration arguments ensure that, with high probability, the advice works for all possible w (via an union bound) under the promise related to C . This yields the following lemma.

Lemma 3.5. *Consider $m, n \in \mathbb{N}$ and $0 < \beta, \eta < 1/6$. Given any $\text{DBNCP}_{\beta, \eta}$ instance (C, w) , there is a setting of parameters (ℓ, N, t) , for which, with overwhelming probability, the preprocessing algorithm **Pre** outputs advice $(h_1, \dots, h_N)$ such that for all w that are either η -close to or β -separated from the code $\mathcal{C}$, the decision returned by Algorithm 2 is correct, where*

$$N = 72m \cdot (1 - 2\eta)^{-4(n/\log(1/\beta)+1)}.$$

Proof. Let $\ell = 2 \cdot \lceil n/\log(1/\beta) \rceil$, by Lemma 3.4, for any $\text{DBNCP}_{m, \beta, \eta}$ instance (C, w) , we have either

$$\mathbb{E}_{h \leftarrow D_C} [(-1)^{\langle h, w \rangle}] > \frac{1}{2}(1 - 2\eta)^{2(n/\log(1/\beta)+1)} \text{ or } \mathbb{E}_{h \leftarrow D_C} [(-1)^{\langle h, w \rangle}] < 2^{-n},$$

for w being close to or separated from C , respectively. As $\eta, \beta < 1/6$, when n is sufficiently large, $(1 - 2\eta)^{2n/\log(1/\beta)} \gg 2^{-n}$. Set the threshold to be

$$t = \frac{1}{3}(1 - 2\eta)^{2(n/\log(1/\beta)+1)}$$

and let $\delta = \frac{1}{6}(1 - 2\eta)^{2(n/\log(1/\beta)+1)}$.

For a fixed w being η -close to the code $\mathcal{C}$, the probability that Algorithm 2 decides w correctly with advice $(h_1, \dots, h_N)$ on the randomness of preprocessing is equivalent to

$$\Pr_{h_i} \left[\sum_i (-1)^{\langle h_i, w \rangle} > t \cdot N \right] > \Pr_{h_i} \left[\sum_i (-1)^{\langle h_i, w \rangle} > (\mathbb{E}_h (-1)^{\langle h, w \rangle} - \delta) \cdot N \right] \geq 1 - e^{-\delta^2 N/2}$$

where the first inequality is obtained from $t < \mathbb{E}_h (-1)^{\langle h, w \rangle} - \delta$ and the second one follows from Hoeffding's inequality; the probability is at least $(1 - e^{-m})$ when $N = 2m \cdot \delta^{-2} = 72m \cdot (1 - 2\eta)^{-4(n/\log(1/\beta)+1)}$.

Similarly, for w being far from the code $\mathcal{C}$, the success probability can be lower-bounded by

$$\Pr_{h_i} \left[\sum_i (-1)^{\langle h_i, w \rangle} \leq t \cdot N \right] > \Pr_{h_i} \left[\sum_i (-1)^{\langle h_i, w \rangle} \leq (\mathbb{E}_h (-1)^{\langle h, w \rangle} + \delta) \cdot N \right] \geq 1 - e^{-\delta^2 N/2}$$

where the first inequality follows from $t > \mathbb{E}_h (-1)^{\langle h, w \rangle} + \delta$ and the second inequality is ensured by Hoeffding's inequality.

Therefore, by the union bound, the probability that Algorithm 2 outputs correctly for all possible w is at least

$$\Pr_{h_i} [\forall w, \text{Decide}(w; h_1, \dots, h_N) \text{ is correct}] \geq 1 - 2^m \cdot e^{-m} > 1 - 2^{-0.4m},$$

when h_i is generated by the preprocessing algorithm. $\square$

Proposition 3.1. *Assuming $0 < \beta, \eta < 1/6$, there is an algorithm with preprocessing for $\text{DBNCP}_{\beta, \eta}$ with both advice size and running time $m^2 \cdot \exp[O(\eta n / \log(1/\beta))]$.*

Proof of Proposition 3.1. From Lemma 3.5, we conclude that, with overwhelming probability, $\text{Pre}(C)$ returns good advice $(h_1, \dots, h_N)$, which works for all w that satisfies the promise, when $N = 72m \cdot (1 - 2\eta)^{-4(n/\log(1/\beta)+1)}$. This proves that the algorithm with preprocessing works correctly. The advice size and running time of the algorithm is: $O(N \cdot m) = m^2 \cdot \exp O(\eta n / \log(1/\beta))$. $\square$

3.2 Search Algorithm

In this section, we describe the search algorithm and prove Theorem 3.2. With the same preprocessing procedure applied (as in Algorithm 1), the search algorithm proceeds as follows:

Algorithm 3: Search algorithm with parameter N

Input: Vector $w \in \mathbb{F}_2^m$ and advice $(h_1, \dots, h_N)$

Output: $\hat{x} \in \mathbb{F}_2^n$, such that $C \cdot \hat{x}$ is the nearest codeword to w

Algorithm Search($w; h_1, \dots, h_N$):

```

    Set  $S = \sum_k (-1)^{\langle h_k, w \rangle}$ ;
    for  $i \in \{1, \dots, m\}$  do
        Set  $w^{(i)} \leftarrow w$  with  $i$ -th bit flipped;
        if  $\sum_k (-1)^{\langle h_k, w^{(i)} \rangle} < S$  then
            Set  $\hat{e}_i \leftarrow 0$ ;
        else
            Set  $\hat{e}_i \leftarrow 1$ ;
    Set  $\hat{e} = (\hat{e}_1, \dots, \hat{e}_m)$ ;
    Solve the linear system  $(C \cdot \hat{x} + \hat{e} = w)$  for  $\hat{x}$ ;
    return  $\hat{x}$ 

```

Suppose the input is $w = C \cdot x + e$ with advice $(h_1, \dots, h_N)$; the algorithm proceeds to recover each bit of e by testing how the value $\sum_k (-1)^{\langle h_k, w \rangle}$ changes under corresponding bit flip. Intuitively, under the distribution D_C used to show Lemma 3.4, the value $\mathbb{E}_h [(-1)^{\langle h, w \rangle}]$ decreases monotonically when w is farther from $\mathcal{C}$. We can then toggle each coordinate of w to detect coordinates i for which $e_i = 1$. We show this formally below.

Lemma 3.6. *For $m, n \in \mathbb{N}$, $1/m \leq \beta, \eta < 1/8$, consider any $\text{BNCP}_{\beta, \eta}$ instance (C, w) with the promise that w is η -close to $\mathcal{C}$, and let $\ell = 2 \cdot \lceil n / \log(1/(\beta + 2\eta)) \rceil$ be an even integer. Assume that $w = C \cdot x + e$ with $\text{wt}(e) \leq \eta m$, denote $w^{(i)}, e^{(i)}$ as the vector w, e with their i -th bits flipped (respectively). Then, the sign of Δ_i indicates the value of the i -th bit e_i , where*

$$\Delta_i = \mathbb{E}_{h \leftarrow D_C} [(-1)^{\langle h, w \rangle}] - \mathbb{E}_{h \leftarrow D_C} [(-1)^{\langle h, w^{(i)} \rangle}].$$

In particular, if $e_i = 0$, $\Delta_i > \delta$; if $e_i = 1$, $\Delta_i < -\delta$; with

$$\delta > \frac{\ell}{m}(1 - 4\eta)^\ell.$$

Proof. Assume $\text{wt}(e) = \eta'm$ for $\eta' \leq \eta$.

Case 1: If the i -th bit of e is 0, the difference between $\hat{D}(e)$ and $\hat{D}(e^{(i)})$ is

$$\begin{aligned} \hat{D}(e) - \hat{D}(e^{(i)}) &= (1 - 2\eta')^\ell - \left(1 - 2\left(\eta' + \frac{1}{m}\right)\right)^\ell \\ &= \left(1 - 2\left(\eta' + \frac{1}{m}\right) + \frac{2}{m}\right)^\ell - \left(1 - 2\left(\eta' + \frac{1}{m}\right)\right)^\ell \\ &\geq \frac{2\ell}{m} \cdot \left(1 - 2\left(\eta' + \frac{1}{m}\right)\right)^{\ell-1} \\ &\geq \frac{2\ell}{m}(1 - 4\eta)^\ell. \end{aligned}$$

By (5), we have

$$\begin{aligned} \Delta_i &= \frac{\sum_{v \in \mathcal{C}} \hat{D}(v + w) - \sum_{v \in \mathcal{C}} \hat{D}(v + w^{(i)})}{\sum_{v \in \mathcal{C}} \hat{D}(v)} \\ &\geq \frac{\hat{D}(e) - \hat{D}(e^{(i)}) - (2^n - 1) \cdot (\beta + 2\eta)^\ell}{1 + (2^n - 1) \cdot \beta^\ell}. \end{aligned}$$

Let $\alpha = (\beta + 2\eta)$, $\ell = 2 \cdot \lceil n / \log(1/\alpha) \rceil$, then for sufficiently large n we have

$$2^n \cdot (\beta + 2\eta)^\ell < 2^{-n}.$$

When $\ell \approx 2n / \log(1/\alpha)$ and $\beta, \eta < 1/8$, $2^{-n} \ll (2\ell/m) \cdot (1 - 4\eta)^\ell$ for all sufficiently large n . Thus,

$$\Delta_i > \frac{1}{2} \left(\hat{D}(e) - \hat{D}(e^{(i)}) \right) \geq \frac{\ell}{m}(1 - 4\eta)^\ell,$$

which proves the claim for this case.

Case 2: If the i -th entry of e is 1,

$$\begin{aligned} \hat{D}(e^{(i)}) - \hat{D}(e) &= \left(1 - 2\left(\eta' - \frac{1}{m}\right)\right)^\ell - (1 - 2\eta')^\ell \\ &= \left(1 - 2\eta' + \frac{2}{m}\right)^\ell - (1 - 2\eta')^\ell \\ &> \frac{2\ell}{m}(1 - 2\eta')^\ell. \end{aligned}$$

Using similar arguments as above, we can obtain

$$\begin{aligned} -\Delta_i &= \frac{\sum_{v \in \mathcal{C}} \hat{D}(v + w^{(i)}) - \sum_{v \in \mathcal{C}} \hat{D}(v + w)}{\sum_{v \in \mathcal{C}} \hat{D}(v)} \\ &\geq \frac{\hat{D}(e^{(i)}) - \hat{D}(e) - (2^n - 1) \cdot (\beta + 2\eta)^\ell}{1 + (2^n - 1) \cdot \beta^\ell} \\ &> \frac{\ell}{m}(1 - 2\eta)^\ell, \end{aligned}$$

which proves the claim for this case as well. $\square$

We will also require the following concentration claim in our main proof. This follows from standard inequalities.

Lemma 3.7. *For any $w_1, w_2 \in \mathbb{F}_2^m$, if there exists $\delta > 0$, such that*

$$\mathbb{E}_{h \leftarrow D_C} \left[(-1)^{\langle h, w_1 \rangle} \right] - \mathbb{E}_{h \leftarrow D_C} \left[(-1)^{\langle h, w_2 \rangle} \right] > \delta,$$

then for $N \geq 8m \cdot \delta^{-2}$, we have that

$$\Pr_{h_1, \dots, h_N \leftarrow D_C} \left[\sum_k (-1)^{\langle h_k, w_1 \rangle} > \sum_k (-1)^{\langle h_k, w_2 \rangle} \right] > 1 - 2 \exp(-m).$$

Proof. By Hoeffding's inequality,

$$\Pr_{h_1, \dots, h_N} \left[\sum_k (-1)^{\langle h_k, w_1 \rangle} > \left(\mathbb{E}_h (-1)^{\langle h, w_1 \rangle} - \frac{\delta}{2} \right) \cdot N \right] > 1 - \exp(-\delta^2 N/8) = 1 - \exp(-m),$$

where $N = 8m \cdot \delta^{-2}$. Similarly,

$$\Pr_{h_1, \dots, h_N} \left[\sum_k (-1)^{\langle h_k, w_2 \rangle} < \left(\mathbb{E}_h (-1)^{\langle h, w_2 \rangle} + \frac{\delta}{2} \right) \cdot N \right] > 1 - \exp(-\delta^2 N/8) = 1 - \exp(-m).$$

Since $\mathbb{E}_h (-1)^{\langle h, w_1 \rangle} - \frac{\delta}{2} > \mathbb{E}_h (-1)^{\langle h, w_2 \rangle} + \frac{\delta}{2}$, we have

$$\Pr_{h_1, \dots, h_N} \left[\sum_k (-1)^{\langle h_k, w_1 \rangle} > \sum_k (-1)^{\langle h_k, w_2 \rangle} \right] > 1 - 2 \cdot \exp(-m). \quad \square$$

Finally we can show that the preprocessing step again generates good advice such that the search algorithm produces the correct output. This lemma plays a similar role to Lemma 3.5 for the decisional setting.

Lemma 3.8. *For $m, n \in \mathbb{N}$, $0 < \beta, \eta < 1/8$. Given any $\text{BNCP}_{\beta, \eta}$ instance (C, w) , there is a setting of parameters (ℓ, N) , for which, with high probability, the preprocessing algorithm outputs advice $(h_1, \dots, h_N)$, such that, for all w that is η -close to C , Algorithm 3 outputs x such that $C \cdot x$ is the closest codeword to w , where*

$$N = (2m^3 \log^2(1/\alpha)/n^2) \cdot (1 - 4\eta)^{-4(n/\log(1/\alpha)+1)}, \text{ with } \alpha = \beta + 2\eta.$$

Proof. Consider any fixed $w = C \cdot x + e$ with $\text{wt}(e) \leq \eta m$, denote $w^{(i)}, e^{(i)}$ as the vector w, e with their i -th bit flipped. Let $\ell = 2 \cdot \lceil n/\log(1/\alpha) \rceil$ and $\alpha = \beta + 2\eta$. By Lemma 3.6, when $e_i = 0$

$$\mathbb{E}_{h \leftarrow D_C} \left[(-1)^{\langle h, w \rangle} \right] - \mathbb{E}_{h \leftarrow D_C} \left[(-1)^{\langle h, w^{(i)} \rangle} \right] > \frac{\ell}{m} (1 - 4\eta)^\ell.$$

Let $\delta = \frac{\ell}{m} (1 - 4\eta)^\ell$ and $N = 8m \cdot \delta^{-2} = (2m^3 \log^2(1/\alpha)/n^2) \cdot (1 - 4\eta)^{-4(n/\log(1/\alpha)+1)}$, by Lemma 3.7, the probability that Search in Algorithm 3 sets $\hat{e}_i$ to be 0 for h_k 's being sampled from D_C can be lower-bounded by

$$\Pr_{h_1, \dots, h_N} [\hat{e}_i = 0 | e_i = 0] = \Pr_{h_1, \dots, h_N} \left[\sum_k (-1)^{\langle h_k, w \rangle} > \sum_k (-1)^{\langle h_k, w^{(i)} \rangle} \right] > 1 - 2 \cdot \exp(-m).$$

When $e_i = 1$, according to Lemma 3.6,

$$\mathbb{E}_{h \leftarrow D_C} \left[(-1)^{\langle h, w^{(i)} \rangle} \right] - \mathbb{E}_{h \leftarrow D_C} \left[(-1)^{\langle h, w \rangle} \right] > \frac{\ell}{m} (1 - 4\eta)^\ell,$$

then with the same parameters, Lemma 3.7 implies that

$$\Pr_{h_1, \dots, h_N} [\hat{e}_i = 1 | e_i = 1] = \Pr_{h_1, \dots, h_N} \left[\sum_k (-1)^{\langle h_k, w \rangle} \leq \sum_k (-1)^{\langle h_k, w^{(i)} \rangle} \right] > 1 - 2 \cdot \exp(-m).$$

For a fixed w , the probability that Algorithm 3 finds the correct x is at least

$$\Pr_{h_1, \dots, h_N} \left[\hat{x} = \arg \min_x \text{wt}(C \cdot x + w) \right] = \Pr_{h_1, \dots, h_N} [\forall i, \hat{e}_i = e_i] > 1 - 2m \cdot \exp(-m).$$

Consider all possible w that satisfies the promise, by taking the union bound,

$$\Pr_{h_1, \dots, h_N} \left[\forall w, \hat{x} = \arg \min_x \text{wt}(C \cdot x + w) \right] > 1 - 2m \cdot 2^m \cdot \exp(-m) > 1 - m \cdot 2^{-0.4m+1}. \quad \square$$

Proof of Theorem 3.2. Lemma 3.8 indicates that $\text{Pre}(C)$ outputs a good advice matrix H with overwhelming probability, given a balanced code $\mathcal{C}$, such that Algorithm 3 works correctly for all possible input w , when $N = (m^3 \log^2(1/\alpha)/n^2) \cdot \exp O(\eta n / \log(1/\alpha))$ with $\alpha = \beta + 2\eta$, then the advice size and running time are $O(m \cdot N) = (m^4 \log^2(1/\alpha)/n^2) \cdot \exp O(\eta n / \log(1/\alpha))$. $\square$

3.3 Search to Decision Reduction

Proposition 3.9. *If there exists an algorithm for $\text{DBNCP}_{\beta+2\eta, \eta}$ with message length $n-1$ and block length m with advice size a and time complexity t , then there exists an algorithm for $\text{BNCP}_{\beta, \eta}$ with advice size an and time complexity tn .*

As a consequence of Proposition 3.9, the search problem $\text{BNCP}_{\beta, \eta}$ can be solved using the decision algorithm in time $m^2 n \exp O(\eta n / \log(1/\alpha))$, where $\alpha = \beta + 2\eta$.

Proof. Given a $\text{BNCP}_{\beta, \eta}$ instance (C, w) , where $C \in \mathbb{F}_2^{m \times n}$ and $w \in \mathbb{F}_2^m$ the algorithm: (1) constructs n codes $C^{(i)} \in \mathbb{F}_2^{m \times (n-1)}$ by removing the i -th column of C , (2) queries the $\text{DBNCP}_{\beta+2\eta, \eta}$ oracle on $(C^{(i)}, w)$ for all i , and (3) outputs $\hat{x}$ where

$$\hat{x}_i = \begin{cases} 0, & \text{if } \text{DBNCP}_{\beta+2\eta, \eta}(C^{(i)}, w) \text{ answers YES,} \\ 1, & \text{if not.} \end{cases}$$

All codes $C^{(i)}$ are subcodes of C and therefore β -balanced.

Assuming w is η -close to C , it equals $C \cdot x + e$ for some e of weight less than ηm . We show that $\hat{x}$ must equal x . If $x_i = 0$ then Cx equals $C^{(i)}x^{(i)}$, where $x^{(i)}$ is x with its i -th entry removed. Therefore $w = C^{(i)}x^{(i)} + e$ is also η -close to $C^{(i)}$ and $\text{DBNCP}_{\beta+2\eta, \eta}$ answers YES.

If $x_i = 1$, then w equals $C^{(i)}x^{(i)} + c^{(i)} + e$, where $c^{(i)}$ is the i -th column of C . It remains to argue that $c^{(i)} + e$ is $(\beta + 2\eta)$ -separated from $C^{(i)}$. This guarantees a NO answer from the oracle. If, for contradiction, $c^{(i)} + e$ was $\frac{1}{2}(1 - \beta - 2\eta)$ -close or $\frac{1}{2}(1 + \beta + 2\eta)$ -far from some codeword c in $C^{(i)}$, then $c^{(i)}$ would be $\frac{1}{2}(1 - \beta)$ -close to or $\frac{1}{2}(1 + \beta)$ -far from c . $c + c^{(i)}$ would then be a β -unbalanced codeword of C , violating the promise. $\square$

4 On the Optimality of Our Algorithms

In this section, we show the optimality of the algorithm presented in the previous section. We emphasize that all our impossibility results in this section are stated for the *decision problem* $\text{DBNCP}_{\beta,\eta}$ with parameters set to $m = \text{poly}(n)$ and $\beta = 3\sqrt{n/m}$. This value of β allows an overwhelming fraction of linear codes to meet the balance requirement (see (1)).

4.1 Limits of Threshold Distinguishers

Recall that Algorithm 2 proceeds as follows: on input $w \in \mathbb{F}_2^m$, it takes as advice a matrix $H = (h_1, \dots, h_N) \in \mathbb{F}_2^{N \times m}$ associated with the code $\mathcal{C}$, it calculates $\sum_i (-1)^{\langle h_i, w \rangle}$ and compares it with a threshold T . We call this type of algorithm a *threshold distinguisher* for code $\mathcal{C}$ and refer to N as its size.

More generally, we allow a threshold distinguisher to apply some affine shift $b \in \mathbb{F}_2^N$. That is, it computes $\sum_i (-1)^{\langle h_i, w \rangle + b_i}$ where b_i denotes the i -th coordinate of b . In Section 4.2 we show that threshold distinguishers are somewhat more powerful than they seem.

Our main lower bound shows that threshold distinguishers have limited power.

Theorem 4.1. *For $n, m, d \in \mathbb{N}$, $\beta = 3\sqrt{n/m}$, for any code $\mathcal{C}$ of message length n , blocklength m , and dual distance d , no threshold distinguisher of size $\frac{1}{7} \exp[\min\{\eta d/6, 3n\}]$ correctly decides whether w is η -close to or β -separated from $\mathcal{C}$ for all w .*

All but $2^{-n/2}$ linear codes have dual distance at most $d = n/(2 \log(m))$. The reason is that there are at most m^d non-zero strings of weight at most d , and each of them is a dual codeword with probability 2^{-n} . By a union bound all of them fail to be in the dual except with probability $2^{-n/2}$. This gives the following corollary:

Corollary 4.2. *For all but a $2^{-n/2}$ fraction of linear codes $\mathcal{C}$, no threshold distinguisher of size $\frac{1}{7} \exp \min\{(\eta n/12 \log m), 3n\}$ decides whether w is η -close or $3\sqrt{n/m}$ -separated from $\mathcal{C}$ for all w .*

In particular, if m is polynomial in n and $\eta \gg (\log n)^2/n$, polynomial-size threshold distinguishers fail to solve $\text{DBNCP}_{\eta,\beta}$ with $\beta = 3\sqrt{n/m}$.

Proof of Theorem 4.1. By flipping all b_i if necessary, we may assume that the distinguisher accepts when $A_{H,b}(w) \geq T$, and rejects otherwise, where $A_{H,b}(w) = \sum_i (-1)^{\langle h_i, w \rangle + b_i}$.

We argue that $A_{H,b}$ cannot “tell apart” a randomly corrupted random codeword from a truly random string. Let $w = C \cdot x + e$ with random x and the bits of e independent $\text{Ber}(\eta/2)$. We first show that:

$$\mathbb{E} A_{H,b}(w) \leq \mathbb{E} A_{H,b}(r) + N \exp(-\eta d),$$

where r is a truly random string. By linearity of expectation, it is sufficient to argue that $\mathbb{E}(-1)^{\langle h, w \rangle + b}$ is $(1 - \eta)^d$ -close to $\mathbb{E}(-1)^{\langle h, r \rangle + b}$ for every row (h, b) . If h is the all zero string both expectations are one. If h not a dual codeword both are zero. The only difference comes from non-zero dual codewords. The difference is then $(1 - \eta)^{\text{wt}(h)} \leq (1 - \eta)^d \leq \exp(-\eta d)$.

Now assume $A_{H,b}$ outputs at least T for all strings η -close to the code and at most $T - 1$ for all β -separated strings. By the law of total probability,

$$\mathbb{E} A_{H,b}(w) \geq T(1 - \mathbb{P}(\text{FAR})) + \mathbb{E}[A_{H,b}(w) \mid \text{FAR}] \mathbb{P}(\text{FAR}) \geq T - 2N \mathbb{P}(\text{FAR}),$$

where FAR is the event that w is η -far from $\mathcal{C}$. By a Chernoff bound, this has probability at most $\exp(-\eta m/6)$. On the other hand,

$$\mathbb{E} A_{H,b}(r) \leq (T - 1)(1 - \mathbb{P}(\overline{\text{SEP}})) + \mathbb{E}[A_{H,b}(w) \mid \overline{\text{SEP}}] \mathbb{P}(\overline{\text{SEP}}) \leq (T - 1) + 2N \mathbb{P}(\overline{\text{SEP}}),$$

where $\overline{SEP}$ is the event that r is not β -separated from C . By a union bound and a Chernoff bound, $\overline{SEP}$ had probability at most $2^n \cdot 2 \exp(-\beta^2 m/2) \leq 2 \exp(-3n)$. In summary,

$$\begin{aligned} T - 2N \exp(-\eta m/6) &\leq \mathbb{E} A_{H,b}(w) \\ &\leq \mathbb{E} A_{H,b}(r) + N \exp(-\eta d) \\ &\leq (T - 1) + N \exp(-\eta d) + 2N \cdot 2 \exp(-3n). \end{aligned}$$

It follows that

$$\frac{1}{N} \leq \exp(-\eta d) + 2 \exp(-\eta m/6) + 4 \exp(-3n) \leq 7 \max\{\exp(-\eta d), \exp(-\eta m/6), \exp(-3n)\}.$$

As $d \leq m$, $\exp(-\eta d)$ and $\exp(-\eta m/6)$ are both bounded by $\exp(-\eta d/6)$. $\square$

4.2 Limits of Interval Distinguishers

Instead of a threshold, a distinguisher could potentially base its decision on some other function of the measure $A_{H,b}(w)$. For example, it may be sensible to accept inputs whose value is close to zero and reject those whose value is far from zero, be it positive or negative. This is a special case of an interval distinguisher. In general, a k -interval distinguisher partitions the range of $A_{H,b}$ into k intervals and decides based on the identity of the interval that $A_{H,b}(w)$ belongs to.

Corollary 4.3. *For $n, m \in \mathbb{N}$, $\beta = 3\sqrt{n/m}$, for any code C of message length n , blocklength m , and dual distance d , no k -interval distinguisher of size $\frac{1}{14} \exp[\min\{\eta d/6, 3n\}/(k-1)]$ correctly decides whether w is η -close to or β -separated from C for all w .*

We prove it by reducing an interval distinguisher to a threshold distinguisher.

Lemma 4.4. *Every k -interval distinguisher of size N can be simulated by a threshold distinguisher of size at most $(2N)^{k-1}$.*

Proof. For every partition of the line into k intervals there is a polynomial p of degree $k-1$ that alternates sign among the intervals. Given a k -interval distinguisher $A_{H,b}$ we construct a threshold distinguisher $A_{\hat{H},\hat{b}}$ so that

$$A_{\hat{H},\hat{b}}(w) = p(A_{H,b}(w)). \quad (9)$$

The value $A_{\hat{H},\hat{b}}(w)$ is positive precisely when $A_{H,b}(w)$ falls into a positive interval. The polynomial p factorizes as

$$p(z) = \prod_{i=1}^{k-1} (z - a_i),$$

where the breakpoints a_i are integers between $-N$ and N .

The advice $\hat{H}, \hat{b}$ will be constructed by “applying” p to H, b . To do so we describe how to subtract a constant from a threshold distinguisher, and how to multiply two threshold distinguishers.

To subtract a constant a from H, b we pad H with $|a|$ zero rows and pad b with $|a|$ ones if $a > 0$ and $|a|$ zeros if $a < 0$. The resulting distinguisher $\hat{H}, \hat{b}$ satisfies

$$A_{\hat{H},\hat{b}}(w) = A_{H,b}(w) - a.$$

To multiply two distinguishers H, b and H', b' , we create a new distinguisher $\hat{H}, \hat{b}$ whose rows are the XORS of all pairs of rows of (H, b) and (H', b') . Then

$$A_{\hat{H},\hat{b}}(w) = \sum_{h,h'} (-1)^{\langle h+h', w \rangle + b+b'} = \left(\sum_h (-1)^{\langle h, w \rangle + b} \right) \left(\sum_{h'} (-1)^{\langle h', w \rangle + b'} \right) = A_{H,b}(w) \cdot A_{H',b'}(w).$$

Applying N subtractions and $k - 2$ multiplications we obtain a distinguisher satisfying (9). Subtraction affects size by at most an additive N . As the original distinguisher has size N it at most doubles it. Multiplication results in a distinguisher whose size is the product of its parts. The size of the final distinguisher is therefore at most $(2N)^{k-1}$. $\square$

4.3 Optimality of our Distribution

The specific measure D over m -bit strings defined in Section 3.1 is key for our analysis. By (6), w has nonnegligible bias against tests of relative weight $O((\log n)^2/n)$ and tiny bias against all $n^{-\Omega(1)}$ balanced tests (assuming C is $n^{-\Omega(1)}$ -balanced). (In contrast, a symmetric product measure with the same expected Hamming weight has negligible bias beyond $w = O(\log n/n)$.)

Is there a better choice of D that remains biased against tests of weight $\omega((\log n)^2/n)$ yet remains pseudorandom against all balanced tests? Such a D would have improved the decoding radius η in Proposition 3.1. The improved algorithm would have then stood in contradiction to our lower bound Theorem 4.1. We summarize the conclusion:

Lemma 4.5. *Assume $m > c(\ln 2/\beta)(\ln 1/\gamma)/\beta^2\eta$ for some absolute constant c . For any distribution D over $\{-1, 1\}^m$, if $\hat{D}(w) \geq \gamma$ for all w of weight at most ηm (for even ηm and $\eta \leq 1/4$), there must exist a β -balanced w for which*

$$\hat{D}(w) \geq \exp \left[-\frac{\ln 2/\beta \cdot \ln 1/\gamma}{\eta} \right].$$

In our application to decoding, when $\beta = n^{-\Theta(1)}$ and $\gamma = n^{-\Theta(1)}$, the lower bound reads $\hat{D}(w) \geq \exp -O((\log n)^2/\eta)$. When η is $\Theta((\log n)^2/n)$ some β -balanced test has bias at least $\exp -O(n)$, matching (6).

Lemma 4.5 is a statement about distributions over m -bit strings that makes no reference to codes and algorithms. Qualitatively, it says that if a distribution has noticeable bias against all light tests, then it must have some bias against some balanced test. We give a direct analytical proof of this.

Proof of Lemma 4.5. Let X be a sample of D and $t = \eta m$. Let I be a uniformly random index in $\{1, \dots, m\}$. By monotonicity of moments, for every $K \geq t$,

$$\mathbb{E}(\mathbb{E} X_I | X)^K \geq (\mathbb{E}(\mathbb{E} X_I | X)^t)^{K/t}.$$

We can expand $\mathbb{E}(\mathbb{E} X_I | X)^t$ as $\mathbb{E} X_{I_1} \cdots X_{I_t}$, where $I_1, \dots, I_t$ are independent replicas of I . By our assumption on the light tests, for every fixing of $I_1, \dots, I_t$ this expression is at least γ . Therefore

$$(\mathbb{E}(\mathbb{E} X_I | X)^t)^{K/t} \geq \gamma^{K/t}.$$

Let K be a Poisson random variable of rate λm (to be determined). Averaging over K we obtain

$$\mathbb{E}(\mathbb{E} X_I | X)^K \geq \mathbb{E} \gamma^{K/t} \cdot \mathbb{1}(K \geq t) \geq \mathbb{E} \gamma^{K/t} - \mathbb{P}(K < t).$$

as $\gamma \leq 1$. Using the formula for the Poisson moment generating function,

$$\mathbb{E} \gamma^{K/t} = \exp \lambda m (\gamma^{1/t} - 1) \geq \gamma^{\lambda m/t} = \gamma^{\lambda/\eta}.$$

and so

$$\mathbb{E}(\mathbb{E} X_I | X)^K \geq \gamma^{\lambda/\eta} - \mathbb{P}(\text{Poisson}(\lambda m) < \eta m). \quad (10)$$

The left-hand side also expands into a product of a K independent replicas $\mathbb{E} X_{I_1} \cdots X_{I_K}$. Let N_i be the number of times X_i occurs in this expression, i.e. N_i is the number of indices j for which $I_j = i$. By the additivity of Poisson random variables, the N_i are independent Poissons of rate λ . Therefore

$$\mathbb{E}(\mathbb{E} X_I | X)^K = \mathbb{E} X_1^{N_1} \cdots X_m^{N_m} = \mathbb{E} X_1^{\oplus N_1} \cdots X_m^{\oplus N_m} = \mathbb{E} \hat{D}(\oplus N), \quad (11)$$

where $\oplus N_i = N_i \bmod 2$ and $\oplus N = (\oplus N_1, \dots, \oplus N_m)$, because $X_i^2 = 1$. The random variables $\oplus N_i$ inherit their independence from N_i . Marginally they are Bernoulli of bias

$$\mathbb{E}(-1)^{\oplus N_i} = \mathbb{E}(-1)^{N_i} = \exp -2\lambda.$$

again using the formula for the Poisson moment generating function.

We choose λ so that $\beta = 2 \exp -2\lambda$. By Chernoff bounds, $\oplus N$ is β -balanced except with probability $\exp -\Omega(\beta^2 m)$. By the total probability theorem,

$$\mathbb{E} \hat{D}(\oplus N) \leq \max_{w \text{ is } \beta\text{-balanced}} \hat{D}(w) + \exp -\Omega(\beta^2 m).$$

Plugging into (10) and (11),

$$\max_{w \text{ is } \beta\text{-balanced}} \hat{D}(w) \geq \gamma^{(\ln 2/\beta)/2\eta} - \exp -\Omega(\beta^2 m) - \mathbb{P}(\text{Poisson}(\lambda m) < \eta m).$$

Under our assumptions on m and η and the Poisson large deviation inequality [Can22, Theorem A.8] the leading term dominates and gives the desired bound. $\square$

Acknowledgements

Work supported by an NSERC Discovery Grant, and by the National Research Foundation, Singapore, under its NRF Fellowship programme, award no. NRF-NRFF14-2022-0010.

References

- [ABN⁺92] Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ron M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38(2):509–516, 1992. Earlier version in ISIT '91.
- [ABSS93] Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. In *34th Annual Symposium on Foundations of Computer Science, FOCS 1993*, pages 724–733. IEEE Computer Society, 1993.
- [AGHP92] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k -wise independent random variables. *Random Structures & Algorithms*, 3(3):289–304, 1992. Earlier version in FOCS '90.
- [AHI⁺17] Benny Applebaum, Naama Haramaty, Yuval Ishai, Eyal Kushilevitz, and Vinod Vaikuntanathan. Low-Complexity Cryptographic Hash Functions. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 7:1–7:31, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

- [Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *44th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2003*, pages 298–307. IEEE Computer Society, 2003.
- [Alo03] Noga Alon. Problems and results in extremal combinatorics—i. *Discrete Mathematics*, 273(1):31–53, 2003.
- [AMR25] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Trapdoor hash functions and PIR from low-noise LPN. *Cryptology ePrint Archive*, Paper 2025/416, 2025.
- [APY09] Noga Alon, Rina Panigrahy, and Sergey Yekhanin. Deterministic approximation algorithms for the nearest codeword problem. In *12th International Workshop on Randomization and Computation, APPROX-RANDOM 2009*, volume 5687 of *Lecture Notes in Computer Science*, pages 339–351. Springer, 2009.
- [AR05] Dorit Aharonov and Oded Regev. Lattice problems in NP intersect coNP. volume 52, pages 749–765, 2005.
- [BATS13] Avraham Ben-Aroya and Amnon Ta-Shma. Constructing small-bias sets from algebraic-geometric codes. *Theory of Computing*, 9(5):253–272, 2013.
- [BCG⁺20] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Correlated pseudorandom functions from variable-density LPN. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1069–1080. IEEE, 2020.
- [BCG⁺22] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Nicolas Resch, and Peter Scholl. Correlated pseudorandomness from expand-accumulate codes. In *Advances in Cryptology – CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part II*, page 603–633, Berlin, Heidelberg, 2022. Springer-Verlag.
- [Ber10] Daniel J. Bernstein. Grover vs. McEliece. In Nicolas Sendrier, editor, *Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010*, volume 6061 of *Lecture Notes in Computer Science*, pages 73–80. Springer, 2010.
- [BF22] Nir Bitansky and Sapir Freizeit. Statistically sender-private OT from LPN and derandomization. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022 – 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 625–653. Springer, 2022.
- [BFKL93] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO ’93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 278–291. Springer, 1993.
- [BGR25] Vijay Bhattiprolu, Venkatesan Guruswami, and Xuandi Ren. PCP-free APX-hardness of nearest codeword and minimum distance. *CoRR*, abs/2503.11131, 2025.

- [BHI⁺24] Nir Bitansky, Prahladh Harsha, Yuval Ishai, Ron D. Rothblum, and David J. Wu. Dot-product proofs and their applications. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024*, pages 806–825. IEEE Computer Society, 2024.
- [BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1+1=0$ improves information set decoding. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 520–536. Springer, 2012.
- [BK02] Piotr Berman and Marek Karpinski. Approximating minimum unsatisfiability of linear equations. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '02*, page 514–516. Society for Industrial and Applied Mathematics, 2002.
- [BLSV18] Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. Anonymous ibe, leakage resilience and circular security from new assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 535–564. Springer, 2018.
- [BLVW19] Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs. Worst-case hardness for LPN and cryptographic hashing via code smoothing. In *Advances in Cryptology - EUROCRYPT 2019*, volume 11478 of *Lecture Notes in Computer Science*, pages 619–635. Springer, 2019.
- [BM17] Leif Both and Alexander May. Optimizing bjmm with nearest neighbors: full decoding in $2^{22/21n}$ and mceliece security. In *WCC workshop on coding and cryptography*, volume 214, 2017.
- [BM18] Leif Both and Alexander May. Decoding linear codes with high error rate and its impact for LPN security. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - PQCrypto 2018*, volume 10786 of *Lecture Notes in Computer Science*, pages 25–46. Springer, 2018.
- [Can22] Clément L. Canonne. Topics and techniques in distribution testing: A biased but representative sample. *Foundations and Trends in Communications and Information Theory*, 19(6):1032–1198, 2022.
- [CRR21] Geoffroy Couteau, Peter Rindal, and Srinivasan Raghuraman. Silver: Silent vole and oblivious transfer from hardness of decoding structured ldpc codes. In *Advances in Cryptology - CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part III*, page 502–534, Berlin, Heidelberg, 2021. Springer-Verlag.
- [DAR25] Thomas Debris-Alazard and Nicolas Resch. Worst and average case hardness of decoding via smoothing bounds. In *Public-Key Cryptography - PKC 2025 - 28th IACR International Conference on Practice and Theory of Public-Key Cryptography, Røros, Norway, May 12-15, 2025, Proceedings, Part II*, volume 15675 of *Lecture Notes in Computer Science*, pages 363–392. Springer, 2025.

- [DEEK24] Léo Ducas, Andre Esser, Simona Etinski, and Elena Kirshanova. Asymptotics and improvements of sieving for codes. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024*, volume 14656 of *Lecture Notes in Computer Science*, pages 151–180. Springer, 2024.
- [FM04] Uriel Feige and Daniele Micciancio. The inapproximability of lattice and coding problems with preprocessing. *Journal of Computer and System Sciences*, 69(1):45–67, 2004.
- [FT05] Joel Friedman and Jean-Pierre Tillich. Generalized Alon–Boppana theorems and error-correcting codes. *SIAM Journal on Discrete Mathematics*, 19(3):700–718, 2005.
- [GG00] Oded Goldreich and Shafi Goldwasser. On the limits of nonapproximability of lattice problems. volume 60, pages 540–563, 2000. Preliminary version in STOC 98.
- [KT17] Ghazal Kachigar and Jean-Pierre Tillich. Quantum information set decoding algorithms. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography – PQCrypto 2017*, volume 10346 of *Lecture Notes in Computer Science*, pages 69–89. Springer, 2017.
- [Lev83] V. I. Levenshtein. Bounds for packings of metric spaces and some of their applications. *Problemy Kibernet.*, (40):43–110, 1983.
- [LLM06] Yi-Kai Liu, Vadim Lyubashevsky, and Daniele Micciancio. On bounded distance decoding for general lattices. In *9th International Workshop on Randomization and Computation, APPROX-RANDOM 2006*, volume 4110 of *Lecture Notes in Computer Science*, pages 450–461. Springer, 2006.
- [MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $O(2^{0.054n})$. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 107–124. Springer, 2011.
- [MO15] Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 203–228. Springer, 2015.
- [MRRW06] R. McEliece, E. Rodemich, H. Rumsey, and L. Welch. New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE Trans. Inf. Theor.*, 23(2):157–166, September 2006.
- [NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, 1993. Earlier version in STOC ’90.
- [Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Trans. Inf. Theory*, 8(5):5–9, 1962.
- [Reg04] Oded Regev. Improved inapproximability of lattice and coding problems with preprocessing. *IEEE Trans. Inf. Theory*, 50(9):2031–2037, 2004. Preliminary version in CCC 2003.

- [SV19] Noah Stephens-Davidowitz and Vinod Vaikuntanathan. SETH-hardness of coding problems. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019*, pages 287–301. IEEE Computer Society, 2019.
- [TS17] Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC '17*, pages 238–251, New York, NY, USA, 2017. ACM. Also available as ECCC Technical Report TR17-041.
- [Wel74] Lloyd R. Welch. Lower bounds on the maximum cross correlation of signals. *IEEE Trans. Inf. Theory*, 20:397–399, 1974.
- [YZ21] Yu Yu and Jiang Zhang. Smoothing out binary linear codes and worst-case sub-exponential hardness for LPN. In *Advances in Cryptology - CRYPTO 2021*, volume 12827 of *Lecture Notes in Computer Science*, pages 473–501. Springer, 2021.
- [YZW⁺19] Yu Yu, Jiang Zhang, Jian Weng, Chun Guo, and Xiangxue Li. Collision resistant hashing from sub-exponential learning parity with noise. In *Advances in Cryptology – ASIACRYPT 2019: 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8–12, 2019, Proceedings, Part II*, page 3–24, Berlin, Heidelberg, 2019. Springer-Verlag.