Electronic Colloquium on Computational Complexity, Report No. 159 (2024)

# Binary Codes with Distance Close to Half

Dean Doron*

We survey recent and classical results and techniques concerning binary codes in the large distance (or, high-noise) regime, and the closely related notion of $\varepsilon$-balanced codes. Our (hopefully small-biased) column will mainly discuss encoding, and decoding from adversarial errors.

*A previous version of this text originally appeared as an ACM SIGACT News Complexity Theory Column [Dor24].*

## 1 Introducing Our Codes

The hero of this column is a linear error correcting code $\mathcal{C} \subseteq \mathbb{F}_2^n$ of large relative distance, of the form $\frac{1}{2} - \varepsilon$ for some small $\varepsilon > 0$. That is, we require $\Pr_{i \sim [n]}[c_i \neq c'_i] \geq \frac{1}{2} - \varepsilon$ for any distinct $c, c' \in \mathcal{C}$. Since our set $\mathcal{C}$ is always a linear subspace, it suffices to require that the Hamming weight of every nonzero codeword is at least $(\frac{1}{2} - \varepsilon)n$.

As the reader is probably aware, large distance allows us to communicate even in the presence of many corruptions. However, there is an obvious tension: How large can $\mathcal{C}$ be? This is captured by the code's *rate*, defined as $R = k/n$, where $k$ is the dimension of $\mathcal{C}$. Viewing $\mathcal{C}$ as an encoding map that encodes a *message* $x \in \mathbb{F}_2^k$ as a *codeword* $\mathcal{C}(x)$ (and we will do this implicitly from now on, identifying the subspace $\mathcal{C}$ with the image of the encoding map $\mathcal{C}$), a large $\mathcal{C}$ translates to adding little *redundancy* to the original information. The Gilbert–Varshamov bound tells us that there exist large-distance codes with rate $\Omega(\varepsilon^2)$.

**Theorem 1.1** ([Gil52, Var57])**.** *For every $\delta \in [0, 1/2)$ there exists a family of linear binary codes with rate $R \geq 1 - h_2(\delta)$ and relative distance $\delta$, where $h_2$ is the binary entropy function. When $\delta = \frac{1}{2} - \varepsilon$, we have $1 - h_2(\delta) = \Theta(\varepsilon^2)$. Allowing some slackness $\eta > 0$, a uniformly random code of rate $R \geq 1 - h_2(\delta) - \eta$ will have relative distance at least $\delta$ with probability $1 - 2^{-\eta n}$.*

Studying rate vs. distance tradeoffs is one of the most fundamental questions, which makes it natural to ask for a lower bound on $R$:

**Theorem 1.2** ([MRRW77], see also [NS09])**.** *The rate $R$ of a family of binary linear codes of distance $\frac{1}{2} - \varepsilon$ satisfies $R = O(\varepsilon^2 \cdot \log \frac{1}{\varepsilon})$.*

This already brings us to our first open problem: Investigating the $O(\log \frac{1}{\varepsilon})$ gap.

**Open Problem 1.** *Is the GV bound tight for binary codes with distance close to half? Or is it the case that there exists a family of codes with distance $\frac{1}{2} - \varepsilon$ and rate $\omega(\varepsilon^2)$?*

*Department of Computer Science, Ben-Gurion University of the Negev. deand@bgu.ac.il.



ISSN 1433-8092

In this column, we will discuss the encoding (that is, computing $\mathcal{C}$) and also the *decoding* of such codes—where an adversary can corrupt some (arbitrary) $p$ fraction of the symbols of $\mathcal{C}(x)$, and the goal is to recover the message $x$ itself. It is not hard to see that *unique* decoding is impossible whenever $p$ is greater than half the distance of $\mathcal{C}$, which puts a bound of $p < \frac{1}{4}$. Unique decoding can be relaxed to *list* decoding by allowing to output a list of potential candidates, one of which is $x$. We say $\mathcal{C}$ is $(\rho, L)$ list-decodable if for any word $w \in \mathbb{F}_2^n$ there are at most $L$ codewords in $\mathcal{C}$ with relative distance at most $\rho$ from $w$.

It turns out that in our regime, list decoding can almost double the number of corruptions we can handle! The Johnson bound (see, e.g., [GRS, Section 7.3]) tells us that *any* code of distance $\frac{1}{2} - \varepsilon$ is $(\rho, L)$ list-decodable for $\rho = \frac{1}{2} - \sqrt{\varepsilon}$, and $L = O(1/\varepsilon)$. Better yet, the list decoding capacity theorem tells us that there exist codes of rate roughly $\varepsilon^2$ that are list decodable all the way up to radius $\frac{1}{2} - 2\varepsilon$ (even a random one will be good with high probability).

### 1.1 Small-Biased Codes in Pseudorandomness

Beyond being a fundamental object in coding theory (and an interesting mathematical structure in itself), our codes have great importance in pseudorandomness.

**Small-Bias Sets.** We say that a set $S \subseteq \{0,1\}^n$ is *$\varepsilon$-biased* if the uniform distribution over the elements of $S$ is indistinguishable from uniform by every *linear test*. That is, $S$ is $\varepsilon$-biased if for every nonempty $T \subseteq [n]$ it holds that $\Pr_{s \sim S}[\bigoplus_{i \in T} s_i = 1] \in [\frac{1-\varepsilon}{2}, \frac{1+\varepsilon}{2}]$. Now, looking at our code $\mathcal{C}$, assume that not only is the relative Hamming weight of each nonzero codeword at least $\frac{1}{2} - \varepsilon$, but it is also bounded from above by $\frac{1}{2} + \varepsilon$. Such codes are called *$\varepsilon$-balanced*,[1] and we know that (linear) $\varepsilon$-balanced codes and $\varepsilon$-biased sets are essentially the same: $S$ is $2\varepsilon$-biased if and only if the $|S| \times n$ matrix $A_S$ whose rows are the elements of $S$ is a generating matrix of an $\varepsilon$-balanced code (so $\mathcal{C}(x) = A_S \cdot x$).

Efficiently generated $\varepsilon$-biased sets (also called $\varepsilon$-biased generators) are one of the most fundamental objects in pseudorandomness, and serve as building blocks in countless constructions, from two-source extractors to pseudorandom generators that fool various kinds of branching programs. We refer the reader to the excellent survey by Hatami and Hoza [HH24] for examples of pseudorandom generators that use $\varepsilon$-biased generators.

**Randomness Extractors.** The theory of randomness extractors (and related objects such as condensers and dispersers) aims to utilize very weak sources of randomness, both for practical and theoretical applications. For example, whenever randomness is used—whether because it is necessary or just because it is faster and simpler in practice, as in many randomized algorithms, protocols, and other applications—an unlimited supply of independent, unbiased bits is often assumed. It is essential, therefore, that the crude randomness generated by such sources be *purified*. A function $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (strong, seeded) extractor for min-entropy $\ell$, if for any $\ell$-source[2] $X$ and an independent uniform seed $Y$, it holds that $\mathsf{Ext}(X, Y)$ is close to uniform in statistical

[1] The added restriction on the maximal weight does not seem to make much difference. All the bounds above still hold, and the best large distance code constructions we have are also balanced.

[2] A distribution $X \sim \{0,1\}^n$ is an $\ell$-source if it has $\ell$ min-entropy, namely the probability of each $x \sim X$ is at most $2^{-\ell}$.

distance, even conditioned on a typical $Y$.[3] When $m = 1$, which is already a challenging setting, we have a (near) equivalence between extractors and list-decodable codes.

**Proposition 1.3.** *For a code $\mathcal{C}\colon \mathbb{F}_2^k \to \mathbb{F}_2^n$ we define $\mathsf{Ext}_{\mathcal{C}}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$ by $\mathsf{Ext}_{\mathcal{C}}(x,y) = \mathcal{C}(x)_y$. If $\mathcal{C}$ is $(\rho = \frac{1}{2} - \tau, L)$ list decodable then $\mathsf{Ext}_{\mathcal{C}}$ is a strong $(\ell = \log \frac{L}{\tau} + 1, 2\tau)$ extractor, and if $\mathsf{Ext}_{\mathcal{C}}$ is a strong $(\ell, \tau)$ extractor then $\mathcal{C}$ is $(\rho = \frac{1}{2} - \tau, L = 2^\ell - 1)$ list decodable.*

Note that by the Johnson bound, a code with distance $\frac{1}{2} - \varepsilon$ and rate $R = \varepsilon^{O(1)}$ readily gives an extractor with $\ell = O(\log(1/\varepsilon))$, error $O(\varepsilon)$, and seed length $d = \log n + O(\log(1/\varepsilon))$. By working over larger alphabets, one can extend Proposition 1.3 to $m > 1$ (see also [TZ04] for a complete equivalence between extractors and soft-decision list-decoding).

**Hardness Amplification.** Given a function $f\colon \{0,1\}^n \to \{0,1\}$ that is hard for Boolean circuits of size $s$ (namely, no circuits of size $s$ compute $f$ correctly on all inputs), *hardness amplification* is the worst-case to average-case procedure of transforming it to some $f'\colon \{0,1\}^{n'} \to \{0,1\}$ such that for any size-$s'$ circuit $C$ it holds that $\Pr_x[C(x) = f(x)] \leq \frac{1}{2} + \varepsilon$. Ideally, we want the procedure to be efficient and with minor loss in parameters (that is, $s' \approx s$ and $n' \approx n$, as a function of $\varepsilon$ of course).

In pseudorandomness and derandomization, the classical and prevalent approach for constructing pseudorandom generators for polynomial-sized circuits (that suffices to derandomize **BPP**, the complexity class of languages solved by a polynomial-time randomized algorithms) goes via hardness amplification, and lets us derandomize randomized algorithms assuming only worst-case hardness.

Following a sequence of beautiful works starting from [STV01], we know that (black-box) hardness amplification procedures are tightly connected to *local* list decoding of large-distance binary codes (see, e.g., [GGH+07, GSV18, SV22, DPT24] for more recent constructions and insights). Specifically, we can take $f' = \mathcal{C}(f)$, where we view $f$ as a truth-table of length $2^n$, and $\mathcal{C}$ maps $2^n$ bits to $2^{n'}$ bits. While a relative distance of $\frac{1}{2} - \varepsilon$ is necessary for hardness amplification via local list decoding (since the decoding should be from only $\frac{1}{2} + \varepsilon$ fraction of agreement), local list decoding does not readily follow from the distance property, and the notion of locality warrants its own separate discussion.

## 2 Explicit Constructions

Constructing efficiently *encodable* codes of distance $\frac{1}{2} - \varepsilon$—even without an efficient decoding algorithm—is already challenging, and has been the subject of extensive and fruitful research over the past decades.[4] A prominent and natural way of constructing such codes is via *code concatenation* (e.g., as in [AGHP92, BT13]), and we discuss this in Section 2.1. In code concatenation, we have some outer code $\mathcal{C}_{\text{out}}$ over a large field, say $\mathbb{F}_q$, and an inner binary code $\mathcal{C}_{\text{in}}$ that encodes messages of length $\log_2 q$. The resulting code $\mathcal{C} = \mathcal{C}_{\text{out}} \circ \mathcal{C}_{\text{in}}$ is obtained by encoding the input message $x$ with $\mathcal{C}_{\text{out}}$, and then encoding each symbol of $\mathcal{C}_{\text{out}}(x)$ using $\mathcal{C}_{\text{in}}$. Explicit concatenation-based constructions have proven useful (and influential), yet they do not approach rate $R = \varepsilon^2$.

In a breakthrough result, Ta-Shma [Ta-17] was able to achieve a rate of $\varepsilon^{2+o(1)}$ using a *bias amplification* method, which we discuss in Section 2.2. This approach, first used by Naor and

[3] Formally, $\mathsf{Ext}$ is a strong $(\ell, \varepsilon)$ extractor if $(\mathsf{Ext}(X,Y), Y)$ is $\varepsilon$-close, in total variation distance, to $U_m \times Y$, where $U_m$ is the uniform distribution over $m$ bits.

[4] Note that for small-bias sets and randomness extractors, efficient encoding is all we need.

Naor [NN93], starts with a code of bias $\varepsilon_0$ and then transforms it (mainly using expander-based techniques) to a code of bias $\varepsilon \ll \varepsilon_0$, hopefully without hurting the rate too much. Bias amplification constructions turned out to be amenable to highly efficient decoding algorithms, which we discuss in Section 3.

A third approach, which we will not cover in depth here, is trace codes. This method also start with an outer code $\mathcal{C}_{\text{out}}$, but then "traces down" each symbol by applying a linear transformation from $\mathbb{F}_q$ to $\mathbb{F}_2$ on each symbol. When $\mathcal{C}_{\text{out}}$ is Reed–Solomon, this gives the dual BCH code,[5] and a variant of this construction was used in [AGHP92], giving a vanishing rate of $R = \Omega(\varepsilon^2 \cdot k^{-1})$, where $k$ is the code's dimension. Very recently, this approach was revisited by Kopparty, Ta-Shma, and Yakirevich, that asked what happens when one uses an Algebraic-Geometric (AG) code as $\mathcal{C}_{\text{out}}$, and made some partial progress when $\mathcal{C}_{\text{out}}$ is the Hermitian code (see [Ta-24]).

Finally, one can also consider "semi random" constructions, where structured randomness can help both in reducing the amount of randomness used for the constructing the code, and in facilitating efficient decoding. We defer the discussion on structured randomness to Section 4.

## 2.1 Concatenation-Based Constructions

Let $\mathcal{C}_{\text{out}}$ be a linear code[6] $\mathcal{C}_{\text{out}} \subseteq \mathbb{F}_q^{n_{\text{out}}}$ of dimension $k_{\text{out}}$ for some large $q$ of characteristic 2, and let $\mathcal{C}_{\text{in}}$ be a smaller inner binary linear code $\mathcal{C}_{\text{in}} \subseteq \mathbb{F}_2^{n_{\text{in}}}$ with dimension $k_{\text{in}} = \log q$. The concatenated code $\mathcal{C} = \mathcal{C}_{\text{out}} \circ \mathcal{C}_{\text{in}} \subseteq \mathbb{F}_2^n$ with $n = n_{\text{out}} n_{\text{in}}$ has dimension $k = k_{\text{out}} k_{\text{in}}$, and every message $x \in \mathbb{F}_q^{k_{\text{out}}}$ (which we can view as a binary message of length $k$,

$$\mathcal{C}(x) = (\mathcal{C}_{\text{in}}(\mathcal{C}_{\text{out}}(x)_1), \ldots, \mathcal{C}_{\text{in}}(\mathcal{C}_{\text{out}}(x)_{n_{\text{out}}})) \in \mathbb{F}_2^n.$$

It is not hard to see that the distance of $\mathcal{C}$ is at least the product of the distances, and we refer to this fact as the "concatenation property".

The natural approach to constructing a good concatenated code is to start with $\mathcal{C}_{\text{out}}$ that has an optimal rate vs. distance tradeoff, which is indeed easy for a large $q$. Alon, Goldreich, Håstad, and Peralta [AGHP92] chose $\mathcal{C}_{\text{out}}$ to be the Reed–Solomon code (which achieves the GV bound) and $\mathcal{C}_{\text{in}}$ to be the Hadamard code, resulting in a code with a vanishing rate of $R = \Omega(\varepsilon^2 \cdot k^{-1})$. The correctness follows from the "degree mantra" – the fundamental yet highly useful fact that a degree-$d$ polynomial has at most $d$ roots.

Ben-Aroya and Ta-Shma [BT13] used the Hermitian code as $\mathcal{C}_{\text{out}}$ and obtained $R = \Omega(\varepsilon^{5/2} \cdot k^{-1/4})$. There, instead of evaluating univariate polynomials over an arbitrary subset of $\mathbb{F}_q$ (as we do with Reed–Solomon), they choose the set of evaluation points more carefully, and evaluate low-degree bivariate polynomials on the Hermitian curve $x^p + x - y^{p+1} = 0$ for $q = p^2$. One can show that no low-degree polynomial intersects the set of points along the curve too much, which leads to a better rate-distance tradeoff in most settings of parameters. Instead of the Hermitian curve, one can use a more subtly chosen set of evaluation points by considering curves of higher dimensions. Using the Riemann–Roch theorem, one can show that evaluating rational functions (rather than just polynomials) over the Garcia–Stichtenoth curve [GS96] gives $R = \Omega(\varepsilon^3)$. We will not get into more details regarding AG codes, but the interested reader can refer to [GS06, Sti09, Coh22].

[5] This is not completely accurate, as one needs to exclude certain polynomials on which the Weil bound cannot be applied. Also, [AGHP92] worked with the quadratic residue (a multiplicative character) rather than the trace function (an additive one), which somewhat changes the choice of polynomials, and the choice of $\mathbb{F}_q$, but does not affect the overall result.

[6] In fact, it suffices for $\mathcal{C}_{\text{out}}$ to only be $\mathbb{F}_2$-linear for the concatenated code $\mathcal{C}$ to be linear.

However, in general, this approach will not achieve the GV bound. If we do not assume any additional properties of $\mathcal{C}_{\text{out}}$ and $\mathcal{C}_{\text{in}}$ and simply use the concatenation property, then all we can hope for is $R = O(\varepsilon^3)$. This is known as the Zyablov bound [Zya71] (so in this sense, using AG codes gives the optimal rate vs. distance tradeoff). Note, however, that the structure of concatenated codes lends itself to natural decoding schemes. We discuss the possibility of attaining the GV bound via concatenated codes in Section 4, where we also mention their decoding.

## 2.2 Bias Amplification, and Ta-Shma Codes

Suppose we already have some code $\mathcal{C}_0 \subseteq \mathbb{F}_2^n$ with bias $\varepsilon_0$ and rate $R_0$, and we want to amplify its bias to some $\varepsilon \ll \varepsilon_0$, hopefully with a good rate. A successful way of doing so is via *direct sum codes*. Let $\mathcal{W} = \{W_1, \ldots, W_{\overline{n}}\} \subseteq [n]^t$ be a family of subsets, and for each $z \in \mathbb{F}_2^n$, we define $\text{dsum}_{\mathcal{W}}(z)$ to be the string $y \in \mathbb{F}_2^{\overline{n}}$, where $y_i = \sum_{j \in W_i} z_j$, and the sum is taken modulo 2. The lifted code $\mathcal{C}$ is defined as

$$\mathcal{C} = \text{dsum}_{\mathcal{W}}(\mathcal{C}_0) = \{\text{dsum}_{\mathcal{W}}(z) : z \in \mathcal{C}_0\}.$$

The notion of *parity sampler*[7] captures how well $\mathcal{W}$ helps in reducing the bias of a code.

**Definition 2.1.** *$\mathcal{W} \subseteq [n]^t$ is an $(\varepsilon_0, \varepsilon)$ parity sampler if for every $z \in \mathbb{F}_2^n$ that is $\varepsilon_0$-biased[8], it holds that $\text{dsum}_{\mathcal{W}}(z)$ is $\varepsilon$-biased.*

When $\mathcal{W} = [n]^t$, it is not hard to see that $\mathcal{W}$ improves any bias $\varepsilon_0$ to bias $\varepsilon_0^t$. But this would lead to a (rapidly) vanishing rate. Our goal is thus to *sparsify* this trivial $\mathcal{W}$.

**Random Walks on Expanders.** A natural first attempt, suggested already by Rozenman and Wigderson (see [Bog12]), is to take random walks over an expander. Let $G$ be a $\lambda$-spectral expander[9] over the vertex set $V = [n]$, and let $\mathcal{W}$ be the set of length-$t$ walks over $G$. If $G$ is $D$-regular (say that $D$ and $\lambda < 1$ are constants), then $|\mathcal{W}| = n \cdot D^{t-1}$, a substantial improvement over the trivial $n^t$. The next theorem shows that random walks do reduce the bias pretty well.

**Theorem 2.2** (see [Ta-17])**.** *For any $\varepsilon_0$ and an even $t$, $\mathcal{W}$ is an $(\varepsilon_0, \varepsilon)$ parity sampler for $\varepsilon = (\varepsilon_0 + 2\lambda)^{t/2}$.*

If one chooses $\mathcal{C}_0$ with any constant bias and rate that depends only on $\varepsilon_0$ (say, the one in [NN93]), then the parameters can be set so that $\mathcal{C} = \text{dsum}_{\mathcal{W}}(\mathcal{C}_0)$ is $\varepsilon$-biased with rate $R = \Omega(\varepsilon^4)$.

Establishing Theorem 2.2 can go as follows. Given $z \in \mathbb{F}_2^n$, let $\Pi_z$ denote the diagonal $n \times n$ matrix that has $(-1)^{z_i}$ in $\Pi_z[i,i]$. Then, one can verify that the bias of $\text{dsum}_{\mathcal{W}}(z)$ is given by

$$\left| \frac{1}{n} \mathbf{1}^\dagger (\Pi_z G)^t \mathbf{1} \right| \leq \left\| (\Pi_z G)^t \right\|,$$

[7] To picture $\mathcal{W}$ as a sampler, one can think of a bipartite graph, where each vertex on the left-hand side corresponds to an element of $\mathcal{W}$, the right-hand side is simply $[n]$, and each $W_i$ on the left-hand side is connected to its $t$ elements. While we will soon see that the standard random walks sampler is also a parity sampler, not every standard sampler (see [Gol11] for the definition) is a parity sampler. For example, the bounded-independence sampler is not a parity sampler.

[8] The bias of a string $z \in \mathbb{F}_2^n$ is defined as $\left|\mathbb{E}_{i \sim [n]}\left[(-1)^{z_i}\right]\right|$.

[9] Letting $\lambda_n \leq \ldots \leq \lambda_1 = 1$ be the eigenvalues of the normalized adjacency matrix of $G$, we say that $G$ is a $\lambda$-spectral expander if $\max\{\lambda_2, -\lambda_n\} \leq \lambda$.

where $\mathbf{1}$ is the all-ones vector, and $\|\cdot\|$ is the operator norm $\|A\| = \max_{x\neq 0} \|Ax\|_2 / \|x\|_2$. As a first attempt we could try to bound $\left\|(\Pi_z G)^t\right\| \leq \|\Pi_z G\|^t$. When a vector $v$ is perpendicular to $\mathbf{1}$, we have that $\|\Pi_z G v\|_2 \leq \|Gv\|_2 \leq \lambda \|v\|_2$. But when $v$ is parallel to $\mathbf{1}$, we have that $\|\Pi_z G v\|_2 = \|Gv\|_2 = \|v\|_2$ because $G\mathbf{1} = \mathbf{1}$, meaning that $\|\Pi_z G\| = 1$.

The key observation is that, in the case where $v$ is parallel to $\mathbf{1}$, the *second* step works in our favor, because $\Pi_z\mathbf{1}$ is mostly perpendicular to $\mathbf{1}$. In particular, $\|\Pi_z G \Pi_z G \mathbf{1}\|_2 \leq (\lambda + \varepsilon_0) \|\mathbf{1}\|_2$. Intuitively, at least one out of every two steps "work", and it's not just a mere artifact of the proof. Indeed, the first application of $\Pi_z$ might map the current vector to $\mathbf{1}$, and if that happens, the second application of $G$ is wasted.

**Random Walks on the Wide Replacement Product.** To break the $R = \Omega(\varepsilon^4)$ barrier, Ta-Shma used an intricately designed random walk on a graph product called the *s-wide replacement product*, originally introduced by Ta-Shma and Ben-Aroya in [BT11]. Let us first discuss the standard replacement product.

We have two graphs: A $D$-regular "large" graph $G = (V_1 = [n], E_1)$, and a $d$-regular "small" graph $H = (V_2 = [D], E_2)$. The vertices of the replacement product $G(\mathrm{r})H$ are simply $V_1 \times V_2$, and we think of the vertices $(v, 1), \ldots, (v, D)$ as the "cloud" of $v$, and we put a copy of $H$ on each cloud. The edges can then be partitioned into *intercloud* edges and *intracloud* edges. The intracloud edges are determined by $H$, and the intercloud edges connect the clouds in the natural way: each original edge $\{u, v\} \in E_1$ is mapped to some $\{(u, i), (v, j)\}$ in a way that there is only one intercloud edge connected to each vertex. A step on the replacement product $G(\mathrm{r})H$ amounts to taking a (random) intracloud edge, followed by a (deterministic) intercloud one. Our bias can then be similarly bounded by $\left\|(\Pi_z GH)^t\right\|$, where each linear operator acts only on the corresponding component.

Ta-Shma observed that this walk still does not "protect" from unruly applications of $\Pi_z$, as described earlier. But since (the adversarial) $\Pi_z$ does not act on the $H$ component, there's hope that a clever "$H$ mechanism" would make it such that if a certain step fails to reduce the bias, many following steps will work well.

To make the idea work, we make $V_2$ larger, namely $V_2 = D^s$, and treat each vertex of $H$ as comprising $s$ registers, each containing an instruction in $[D]$. Since we now have $|V_2| > D$, we need to explain how to map a vertex of $u = (u_0, \ldots, u_{s-1}) \in V_2$ to an instruction in $[D]$ for $G$: at the $i$-th step of the walk, we choose the instruction $u_{i \bmod s}$. That is, while in the standard replacement product, the intercloud steps were precisely determined by the current vertex of $H$, now each intercloud step is determined by some register, depending on $i$, of the current vertex of $H$. Crucially, the amount of randomness invested in each step remains $\log d$.

Algebraically, bounding the bias amounts to bounding the spectral norm of

$$\prod_{i=0}^{t-1} \Pi_z G_{i \bmod s} H,$$

where $G_i$ specifies one intercloud edge for each vertex $(v, u) \in V_1 \times V_2$, which goes to the cloud whose $G$-component is $v[u_i]$ (and again, each operator acts only on the corresponding component).

The hope here is that now, if we do get mapped to $\mathbf{1}$, we are uniform over the cloud and potentially $H$ mixes so well that the labels we get in the *next* few steps are completely uniform, and independent of $\Pi_z$. Indeed, by choosing the $\lambda$-expander $H$ carefully, if we fail once, the next

$s - O(1)$ steps would work perfectly. Thus, instead of an $\varepsilon_0 \to (\varepsilon_0 + O(\lambda))^{t/2}$ amplification, we get, for a small enough $\varepsilon_0$, an $\varepsilon_0 \to (\varepsilon_0 + O(\lambda))^{\frac{s-O(1)}{s} \cdot t}$ amplification.

Working out the parameters, as well as the expanders and the base code $\mathcal{C}_0$, we get the following $\varepsilon$-balanced code.

**Theorem 2.3** (informal; see [Ta-17])**.** *There exists an explicit $\varepsilon$-balanced code $\mathcal{C} \subseteq \mathbb{F}_2^n$ with rate*

$$R = \varepsilon^2 \cdot 2^{-\widetilde{O}(\log(1/\varepsilon)^{2/3})}$$

*Moreover, $\mathcal{C} = \mathrm{dsum}_{\mathcal{W}}(\mathcal{C}_0)$ for some base code $\mathcal{C}_0$, where $\mathcal{W}$ is a collection of length-$t$ walks over a suitable $s$-wide replacement product.*

**Random Walks on Hypergraphs.** In the [Ta-17] construction, we used fresh randomness for every step (that is, $\log d$ uniform bits), and ensured that most steps work. Another approach, taken in [BD22], aims only for one out of every two steps to work, but *shares randomness* between the two steps in order to make them as cheap as a single step.

Specifically, let $G_1$ and $G_2$ be two degree-$d$ expanders on the same set of $n$ vertices. In order to take two *correlated* steps from some vertex $v_1$, we draw a random $i \in [d]$, move to $v_2$, the $i$-th neighbor of $v_1$ in $G_2$, and then move to $v_3$, the $i$-th neighbor of $v_2$ in $G_2$. Since we use the same label $i$ for both steps, this walk can take $t$ "double steps" at the cost of only $t$ steps. If we can guarantee that a double step is as productive as two independent steps, the rate of the resulting code would be $\varepsilon^{2+o(1)}$.

For the double step to work, clearly there must be some relation between the two expanders. Otherwise, $G_2$, for example, could always reverse the step taken by $G_1$. Hence, we would like to think of $G_1$ and $G_2$ together as a single primitive: for each vertex $v_1$, there are $d$ choices for the pair $(v_2, v_3)$. As a result, one can think of $G_1$ and $G_2$ together as a single $d$-regular 3-*uniform hypergraph*, and consider walks on that hypergraph, $H = (V = [n], E_H)$. A step on $H$, according to an instruction $i \in [d]$, starting from some vertex $v$, amounts to choosing the $i$-th hyperedge $e$ that touches $v$ according to some fixed ordering in which $v = e[1]$, *recording* $w = e[2]$, and *moving* to $u = e[3]$. A length-$t$ walk is the sequence of recorded $w$-s.

To analyze the parity sampling capabilities of $\mathcal{W} \subseteq [n]^t$ that consists of length-$t$ walks over 3-uniform hypergraphs, one can define a linear operator $A = A(\Pi_z, H)$, under which the bias can be bounded by $\left\|A^t\right\|$. Which property of $H$ leads to a good amplification? Blanc and Doron study two expansion notions. When $H$ is "$\lambda$-mixing", they get an $\varepsilon_0 \to (\varepsilon_0 + O(\lambda \log(1/\lambda)))^t$ amplification. Under a stronger notion of "$\lambda$-spectral", they get an $\varepsilon_0 \to (\varepsilon_0 + \lambda)^t$ amplification.

Clearly, there's still the issue of the dependence between the degree $d$ and the expansion parameter $\lambda$. Unfortunately, we do not currently have good enough explicit hypergraphs, but one can show that a random one is $\lambda$-mixing, with $\lambda = O(1/\sqrt{d})$, with very high probability. This leads to the following (conditional) construction.

**Theorem 2.4** (informal; see [BD22])**.** *There exists an $\varepsilon$-balanced code $\mathcal{C} \subseteq \mathbb{F}_2^n$ with rate*

$$R = \varepsilon^2 \cdot 2^{-\widetilde{O}\left(\sqrt{\log(1/\varepsilon)}\right)}$$

*that can be constructed in probabilistic polynomial time. Moreover, $\mathcal{C} = \mathrm{dsum}_{\mathcal{W}}(\mathcal{C}_0)$ for some base code $\mathcal{C}_0$, where $\mathcal{W}$ is a collection of length-$t$ walks over a mixing 3-regular hypergraph.*

In [BD22], they also showed that assuming sufficiently good $\lambda$-spectral hypergraphs (which we don't know exist), one can get extremely close to the optimal rate, namely $R = \varepsilon^2 \cdot \frac{1}{\text{polylog}(1/\varepsilon)}$.

To conclude this section, we ask:

**Open Problem 2.** *Can the bias amplification method be pushed further? Specifically, is there a sparse set $\mathcal{W}$ of random walks (over an expander, hypergraph, or a high-dimensional expander), and an explicit base code $\mathcal{C}_0$, for which $\text{dsum}_{\mathcal{W}}(\mathcal{C}_0)$ achieves the GV bound?*

# 3 Decoding Small-Biased Codes

Our focus in this section is on unique decoding and list decoding from *adversarial errors*, i.e., when an adversary is allowed to corrupt any $\delta$-fraction of the coordinates of a received codeword $c = \mathcal{C}(x) \in \mathbb{F}_2^n$. We will concentrate on decoding from direct sum codes, as described above in Section 2.2. Thus, we will assume that we have a base code $\mathcal{C}_0$ which is unique- or list-decodable with sufficiently good parameters (and when we do not insist on the optimal rate vs. distance tradeoff, we do have such codes), and a family of subsets $\mathcal{W}$, and we will ask the following question: Under which conditions on $\mathcal{W}$, can we come up with efficient decoding algorithms? This question will give rise to interesting properties.

We leave the discussion about decoding concatenated codes to Section 4, where we discuss (more) probabilistic constructions.

## 3.1 Decoding from Regularity

For many combinatorial objects, one can define what it means for them to be *pseudorandom*, and then a dual notion of *structure* naturally emerges. This phenomenon gives rise to *regularity lemmas* applicable to, e.g., graphs and certain families of matrices.

Consider a (structured) family of functions $\mathcal{F} \subseteq \mathcal{X} \to [-1, 1]$ where $\mathcal{X}$ is some finite space. We want to approximate *any* function $g \colon \mathcal{X} \to [-1, 1]$ by a function $g_{\mathsf{simple}}$ which consists only of weighted sums of functions from $\mathcal{F}$. While impossible in general,[10] what if we put on our pseudorandomness lens, and only wish $g_{\mathsf{simple}}$ to approximate $g$ from a point of view of only functions from $\mathcal{F}$? That is, we want

$$\mathop{\mathbb{E}}_{x \sim \mu} \left[ f(x) \cdot (g(x) - g_{\mathsf{simple}}(x)) \right] \leq \delta \tag{1}$$

for some associated probability measure $\mu$. By a gradient-descent like argument, one can show that existentially, we can construct $g_{\mathsf{simple}}$ as a combination of only $\frac{1}{\delta^2}$ functions from $\mathcal{F}$.[11]

How can we harness regularity for the task of decoding, and specifically for the goal of reducing the unique- or list-decoding of $\text{dsum}_{\mathcal{W}}(\mathcal{C}_0)$ to that of decoding $\mathcal{C}_0$? The approach suggested by Jeronimo, Srivastava, and Tulsiani [JST21] goes as follows. Given a corrupt $w \in \mathbb{F}_2^{\overline{n}}$, where we identify $[\overline{n}]$ with the elements of $\mathcal{W}$, we wish to find $x$ (or $x$-s) for which $\Delta(w, \mathcal{C}(x))$ is smaller than the decoding radius. For now, we aim to find $x$ that minimizes this Hamming distance, or

[10] It *is* possible, in general, if one wishes to write $g$ as $g_{\mathsf{simple}} + h$, where $h$ being the "pseudorandom" part with respect to the structure of $\mathcal{F}$.

[11] A well-known instantiation of this fact is weak regularity lemmas for dense graphs, wherein one decomposes the adjacency matrix as a weighted sum of a small number of cut matrices, and one can use this decomposition to approximate the number of edges between any two subsets. The algorithmic problem of finding those cut matrices can be reduced to approximating a solution to a certain semidefinite program.

alternatively, find $z \in \mathcal{C}_0$ that minimizes $\Delta(w, \mathrm{dsum}_{\mathcal{W}}(z))$. Let $\mathcal{X} = [n]^t$, and let $g\colon \mathcal{X} \to [-1,1]$ be the function

$$g(i) = \begin{cases} (-1)^{z_i} & i \in \mathcal{W}, \\ 0 & \text{otherwise.} \end{cases}$$

Following the definition of $\mathrm{dsum}_{\mathcal{W}}$, it's not hard to see that for any $z \in \mathbb{F}_2^n$ (not necessarily $z \in \mathcal{C}_0$),

$$1 - 2 \cdot \Delta(w, \mathrm{dsum}_{\mathcal{W}}(z)) = \frac{n^t}{|\mathcal{W}|} \mathop{\mathbb{E}}_{i \sim [n]^t} \left[ g(i) \cdot \chi_z(i_1) \cdots \chi_z(i_t) \right] \triangleq \frac{n^t}{|\mathcal{W}|} \mathop{\mathbb{E}}_{i \sim [n]^t} \left[ g(i) \cdot \chi_z^{\otimes t}(i) \right],$$

where $\chi_z(i) = (-1)^{z_i}$. While $g$ is (somewhat) arbitrary, the function we want to "fool" is structured. Specifically, $\chi_z^{\otimes t}$ belongs to the class

$$\mathcal{F} = \{ \pm \chi_{z_1} \otimes \ldots \otimes \chi_{z_t} : z_1, \ldots, z_t \in \mathbb{F}_2^n \}, \tag{2}$$

so the regularity lemma will tell us that there exists $g_{\mathsf{simple}} = \sum_{k \in [\ell]} c_k \cdot \chi_{z_{i,1}} \otimes \cdots \otimes \chi_{z_{i,t}}$ such that

$$\mathop{\mathbb{E}}_{i \sim [n]^t} \left[ (g(i) - g_{\mathsf{simple}}(i)) \cdot \chi_z^{\otimes t}(i) \right]$$

is small. Now, if indeed we manage to *efficiently find* such a $g_{\mathsf{simple}}$, and the error above is *sufficiently small*, we are left with the task of finding $z \in \mathcal{C}_0$ that maximizes $\mathbb{E}_{i \sim [n]^t} \left[ g_{\mathsf{simple}}(i) \cdot \chi_z^{\otimes t}(i) \right]$. Due to the parity sampling properties of $\mathcal{W}$, we can even range over all $z \in \mathbb{F}_2^n$. In fact, [JST21] ranges over all $z \in [-1,1]^n$ and then deduce a $z \in \mathbb{F}_2^n$ by a random rounding.

*Why is our life easier now?* Our objective only depends on the inner product between $g_{\mathsf{simp}}$ and $\chi_z^{\otimes t}$, so in particular, it only depends on $2^{\ell t}$ indicator functions! We can then partition $[n]$ into $2^{\ell t}$ subsets, and only iterate over $z \in [-1,1]^n$ (after a suitable discretization), each of which gets the same value on every subset. After deducing a $z \in \mathbb{F}_2^n$ that (approximately) maximizes our objective, we simply unique-decode $\mathcal{C}_0$. List decoding is similar: For any $z \in \mathbb{F}_2^n$ that is obtained by that process, try to uniquely decode it according to $\mathcal{C}_0$ (so say $z_0 \in \mathcal{C}_0$ is sufficiently close to $z$), and add it to the output list if $\Delta(w, \mathrm{dsum}_{\mathcal{W}}(z_0))$ is small enough. The Johnson bound guarantees a small output list.[12]

Notice that we still have not used any property of $\mathcal{W}$ besides its parity sampling properties, and indeed, the existential regularity in (1) is neither strong enough for us nor efficient. In order to facilitate efficient decoding, Jeronimo, Srivastava, and Tulsiani use *splittable* families of subsets.

**Splittable $\mathcal{W}$-s.** A family of tuples is $\tau$-splittable if various graphs that originate from partitions of $\mathcal{W}$ are $\tau$-expanders. A bit more formally:

**Definition 3.1** ($\tau$-splittability)**.** *For $1 \le a \le b \le t$, let $\mathcal{W}_{[a,b]} = \{(i_a, \ldots, i_b) : (i_1, \ldots, i_t) \in \mathcal{W}\}$. We say that $\mathcal{W}$ is $\tau$-splittable, if for any triple $1 \le a \le r < b \le t$, whenever we consider the bipartite graph $S_{(a,r,b)}$ with vertex sets $\mathcal{W}_{[a,r]}$ and $\mathcal{W}_{[r+1,b]}$, and edges $(w_1, w_2)$ whenever $w_1 \circ w_2 \in \mathcal{W}_{[a,b]}$, then $S_{(a,r,b)}$ is a $\tau$-expander.*

[12] While a runtime which is exponential in $2^{\ell t}$ does not sound too promising (and that is given $g_{\mathsf{simple}}$, which we have yet to discuss how to achieve), it is only a function of $\varepsilon$, and independent of $n$. Thus, when $\varepsilon$ is large, this gives us an efficient algorithm (see Theorem 3.2 for the details).

When $t = 2$, this is simply a bipartite expander. It is also fairly easy to see that the collection of length-$t$ random walks is splittable. An important milestone was achieved by Jeronimo, Quintana, Srivastava, and Tulsiani [JQST20], where they showed that the collection of random walks on the wide replacement product (discussed in Section 2.2), is also $\tau$-splittable! A bit more specifically, if we set the parameters in (a slight modification of) Ta-Shma's code so that the rate is $\Omega(\varepsilon^{2+o(1)})$, then the corresponding parity sampler is $\tau$-splittable for $\tau$ which is exponential in $-\log(1/\varepsilon)^{1/6}$.

Where does splittability help with regularity? Utilizing the splittable structure, [JST21] prove a "splittable mixing lemma", which is a higher-order analogue of the expander mixing lemma. Using the splittable mixing lemma, they show a much stronger (existential) weak regularity lemma for families of "split functions", such as $\mathcal{F}$ in (2). Still, in order to use $g_{\mathsf{simple}}$ above, we need to find it. The tensor structure of $\mathcal{F}$ allows [JST21] to devise an *algorithmic* weak regularity lemma. Combining matrix cut norm approximation [AN04] with fast SDP solvers for sparse matrices, they get an algorithm which runs in time $\widetilde{O}_{t,\tau}(|\mathcal{W}|)$.

Getting back to our decoding algorithm, after finding $g_{\mathsf{simple}}$, recall that we go over all (discretized) $z \in [-1,1]^n$ which are constant over $2^{\ell t}$ subsets of $[n]$, and apply the unique decoding algorithm of our inner code $\mathcal{C}_0$. Setting parameters appropriately, we can get the following decoding of Ta-Shma codes.

**Theorem 3.2** ([JST21])**.** *There exists an explicit family of $\varepsilon$-balanced binary linear codes $\mathcal{C} \subseteq \mathbb{F}_2^{\overline{n}}$ of rate*

$$\varepsilon^2 \cdot 2^{-O((\log(1/\varepsilon))^{5/6})}$$

*such that:*

1. *Unique decoding: There exists a randomized algorithm that uniquely decodes $\mathcal{C}$ up to half the distance in time $c_1(\varepsilon) \cdot \widetilde{O}(\overline{n})$, where $c_1(\varepsilon)$ is doubly-exponential in $\log^{\alpha}(1/\varepsilon)$ for some $\alpha < 1$.*

2. *List decoding: There exists a randomized algorithm that list-decodes $\mathcal{C}$ up to radius*

$$\rho = \frac{1}{2} - 2^{-O((\log(1/\varepsilon))^{1/6})}$$

*in time $c_2(\varepsilon) \cdot \widetilde{O}(\overline{n})$, where $c_2(\varepsilon)$ is triply-exponential in $\log^{\alpha}(1/\varepsilon)$ for some $\alpha < 1$.*

While the algorithm runs in nearly-linear time in the code's length $\overline{n}$, the dependence on $\varepsilon$ is quite bad. The regime of small (or even mildly-small) $\varepsilon$ is also of great interest. For example, for the pseudorandomness applications we saw in Section 1.1, a small $\varepsilon$ (even polynomially-small in the code's dimension) is often crucial. Moreover, note that the list decoding radius is far from the Johnson bound of $\frac{1}{2} - \sqrt{\varepsilon}$, let alone from the "list decoding capacity", that says we can potentially get arbitrarily close to $\frac{1}{2} - \varepsilon$.

**Sampling $\mathcal{W}$-s.** Splittability is a "structured pseudorandomness" property, and does not hold for a sparse random $\mathcal{W}$. To see this, consider for example the $t = 4$ case. For splittability, we require, in particular, that the bipartite graphs between pairs $(i_1, i_2)$ and $(i_3, i_4)$, which are connected if $(i_1, i_2, i_3, i_4) \in \mathcal{W}$, are expanders. However, as also observed in [JST21], for a random $\mathcal{W}$ of size $O(n)$, such a bipartite graph is a matching with high probability. Recall that in [BD22], the parity sampler is based on walks on 3-regular hypergraphs, and a random such hypergraph is sufficiently good. This suggests that the [BD22] parity sampler is not splittable.

However, in [BD22] they identify that a weaker property suffices to enact the decoding framework of [JST21], which they dub "$\tau$-sampling". This property tells us that we can use $\mathcal{W}$ to sample any set, starting from any prefix.

**Definition 3.3** ($\tau$-sampling)**.** *$\mathcal{W}$ is $\tau$-sampling if for any $S \subseteq [n]$, $j \in [t]$, and $X \subseteq [n]^{j-1}$, it holds that*

$$\left| \Pr_{i \sim \mathcal{W}} [i_j \in S \mid (i_1, \ldots, i_{j-1}) \in X] - \rho(S) \right| \leq \frac{\tau}{\rho(X)},$$

*where $\rho(X)$ is the density of $X$.*

Using the strong sampling property of random walks on hypergraphs, [BD22] provide an analogue of Theorem 3.2, wherein both the rate and the list decoding radius are slightly better (namely, one can replace the 5/6 and 1/6 with 1/2), but the code itself is constructible in randomized polynomial time.

In a very recent work, Dikstein and Hopkins [DH24] defined a similar expansion notion, *complete splittability*, on uniform partite complexes, and used completely splittable complexes for efficient decoding of "ABNNR codes" [ABN+92] (see also [DHK+21], who were the first to instantiate list decoding of ABNNR codes on high-dimensional expanders). Dikstein and Hopkins observed that $\mathcal{W}$ is splittable if and only if it is completely splittable, and this in turn implies that the complex is $\tau$-sampling.

### 3.2 Semidefinite-Programming Based Decoding

Another successful approach for decoding direct sum codes is Sum-of-Squares (SoS) decoding, initiated by Alev, Jeronimo, Quintana, Srivastava, and Tulsiani in [AJQ+20], and further developed in [JQST20].[13] A unifying theme in decoding-based SoS algorithms is to reduce the task of decoding to the task of solving instances of constraint satisfaction problems using SDP solvers. While the decoding parameters of [JQST20, AJQ+20] are somewhat comparable to the regularity-based decoding in Theorem 3.2 (and in particular do not reach the Johnson bound), the runtime in those works is worse.

More recently, Richelson and Roy were able to list-decode Ta-Shma's code, using SDPs, all the way up to the Johnson bound!

**Theorem 3.4** ([RR23])**.** *There exists an explicit family of $\varepsilon$-balanced binary linear codes $\mathcal{C} \subseteq \mathbb{F}_2^{\overline{n}}$ with rate $\Omega(\varepsilon^{2+o(1)})$ such that for any $\theta > 0$, there exists a randomized algorithm that list-decodes $\mathcal{C}$ up to radius*

$$\rho = \frac{1}{2} - \sqrt{\varepsilon} - \theta$$

*in time* $\mathrm{poly}(\overline{n}^{\mathrm{poly}(1/\varepsilon)}, \log(1/\theta))$.

Recall that since we list-decode up to the Johnson bound, the list size is guaranteed to be $O(1/\varepsilon)$. It is also important to note that the seminal work of Guruswami and Rudra [GR08b] gives binary

[13] The SoS framework can itself be seen as a (major) extension of decoding based on approximating $k$-CSPs on expanders [DHK+21, AJT19]. We also note that the splittability notion used in the regularity-based framework was already used in [JQST20, AJQ+20], and one can draw some similarities between the two approaches.

codes of rate $\Omega(\varepsilon^3)$ that are list decodable up to *capacity*.[14] Thus, if one is only interested in the list decoding radius of binary codes with non-vanishing rate, [GR08b] still outperforms Theorem 3.4.

As mentioned above, the algorithm given in [RR23] is an SDP-based one, and it follows the framework esablished in [AJQ+20]. We will not get into the machinery of SDP hierarchies, and only give a very brief outline of the approach. Given a corrupt word $w \in \mathbb{F}_2^{\overline{n}}$, note the our goal is to recover $x \in \mathcal{C}_0$ such that

$$\left| \mathop{\mathbb{E}}_{i \sim \mathcal{W}} [\overline{w}_i \overline{x}_{i_1} \cdots \overline{x}_{i_t}] \right| \geq 2\sqrt{\varepsilon} + 2\theta,$$

where $\overline{w}_i$ is simply $(-1)^{w_i}$, and likewise for $\overline{x}$. In SDP-based decoding, the steps are roughly as follows.

1. Use $w$ to set up and solve an SoS hierarchy. We will not specify the SDP variables in detail, and only mention that they correspond to subsets of the vertices of the expander up to a certain size (say $r$), and an assignment to the elements of those subsets. We also have "distance constraints" variables which are the ones that are problem-specific. Solving the SDP hierarchy will give a *pseudodistribution*.[15]

2. Using SDP *rounding*, round the pseudodistributions to a real distribution on $\mathbb{F}_2^n$.

3. Sample $z$ from that distribution. With sufficiently high probability, that sample has a good agreement with the base code, and we can unique decode to find $x$. Indeed, the SDP is set such that $|\mathbb{E}_{i \sim \mathcal{W}} [\overline{z}_{i_1} \overline{x}_{i_1} \cdots \overline{z}_{i_t} \overline{x}_{i_t}]| \geq 2\varepsilon$ will imply $\left|\mathbb{E}_{i \sim [n]} [\overline{z}_i \overline{x}_i]\right| \geq 2\varepsilon_0$.

   The runtime of the solver is dominated by, roughly, $n^{O(r)}$. Thus, while the SoS framework is powerful, the runtime of SoS-based decoding algorithms seems to be inherently slower than regularity-based decoding algorithms.

The analysis of the decoding algorithm can be seen as mimicking the analysis of expander-based *encoding* algorithms; i.e., we need to interpret the distance of the direct sum code of [Ta-17] as an "SoS proof".[16] Generally speaking, Richelson and Roy come up with an improved rounding step that utilizes the observation that the same argument Ta-Shma used to bound the distance of his code also works in proving the correctness of the list-decoding algorithm. They show that Ta-Shma's proof can be phrased as an SoS proof and can be applied to the pseudodistributions that are obtained from the SDP-solving part. Moreover, they manage to show that there is essentially no loss in making the standard distance analysis into an SoS version (thereby improving the "parity sampling proof" step of [AJQ+20]) . We also remark that recently, Jeronimo, Srivastava, and Tulsiani [JST23] gave SoS distance proofs for other families of codes, such as LDPC Tanner codes and expander-based AEL distance amplification codes over large alphabets, leading to better list decoding algorithms for those codes.

We can summarize the current state of affairs regarding decoding as follows. While existing constructions have near-optimal rate, $\Omega(\varepsilon^{2+o(1)})$, where the $o(1)$ term goes to 0 as $\varepsilon \to 0$, the

[14] There too, the dependence on $\varepsilon$ is bad, not only in the list decoding runtime, but also in the encoding (unless one resorts to randomized constructions).

[15] We can think of a pseudodistribution as a collection of local distributions over subsets of $[n]$. Or, a bit more formally, as an oracle that gets any subset $S \subseteq [n]$ of size at most $r$, and (randomly) generates an assignment to the elements of $S$. The properties of the SDP will ensure certain consistency requirements.

[16] More basically, the random-walks based parity sampler of Section 2.2 can be proven using relatively basic inequalities (such as Cauchy–Schwarz), and it has been known that those can often be applied to the *pseudodistributions* which result from solving the SDP hierarchy.

$\varepsilon$-dependence on other parameters, like the running time of algorithms or the list-decoding radius, can be quite bad, sometimes doubly or even triply exponential. Thus, even given the exciting recent progress, there is still much room for improvement!

**Open Problem 3.** *Construct an explicit code $\mathcal{C} \subseteq \mathbb{F}_2^n$ or relative distance $\frac{1}{2} - \varepsilon$ and rate approaching $\varepsilon^2$, that is list decodable up to the Johnson bound, in time $\mathrm{poly}(1/\varepsilon) \cdot \widetilde{O}(n)$, or even $\mathrm{poly}(n/\varepsilon)$. The decoding (or even the encoding) can be randomized.*

What about codes that achieve the list decoding capacity? There, even the combinatorial list decoding question is open.

**Open Problem 4.** *Construct an explicit code $\mathcal{C} \subseteq \mathbb{F}_2^n$ of relative distance $\frac{1}{2} - \varepsilon$ and rate approaching $\varepsilon^2$, that is $(\rho, L)$ list decodable up to radius $\frac{1}{2} - 2\varepsilon$ and $L = \mathrm{poly}(1/\varepsilon)$.*

# 4 Encoding and Decoding with Structured Randomness

We already saw one example for structured randomness in Section 2.2 where we talked about the [BD22] parity sampler. In this section, we concentrate on the natural framework of code concatenation, introduced in Section 2.1, and ask whether randomness can facilitate optimal constructions and good decoding algorithms.

As mentioned earlier, a prominent concatenation-based approach for constructing good codes is to choose $\mathcal{C}_{\mathrm{out}} \subseteq \mathbb{F}_q^{n_{\mathrm{out}}}$, of dimension $k_{\mathrm{out}}$, with the optimal rate-vs.-distance tradeoff (say, a Reed–Solomon code). Letting $\mathcal{C}_{\mathrm{in}} \subseteq \mathbb{F}_2^{n_{\mathrm{in}}}$ be our inner code of dimension $k_{\mathrm{in}} = \log q$, if $n_{\mathrm{in}}$ is sufficiently small, we can use brute force to find a $\mathcal{C}_{\mathrm{in}}$ that sits on the GV bound. But if we do not assume any additional properties on $\mathcal{C}_{\mathrm{out}}$ and $\mathcal{C}_{\mathrm{in}}$, and simply use the concatenation property, then setting the parameters so that $\mathcal{C} = \mathcal{C}_{\mathrm{out}} \circ \mathcal{C}_{\mathrm{in}}$ has distance $\frac{1}{2} - \varepsilon$, the rate of $\mathcal{C}$ will be at most roughly $\varepsilon^3$. This is known as the *Zyablov bound* [Zya71] (see also [GRS]).[17]

**The Thomessen Construction.** Instead of using a single inner code, several works have focused on a related construction, originally due to Thomessen [Tho83], which uses i.i.d. random inner codes for each coordinate. It can be shown that the resulting code does lie on the GV bound with high probability over the independent $\mathcal{C}_{\mathrm{in}}^{(1)}, \ldots, \mathcal{C}_{\mathrm{in}}^{(n_{\mathrm{out}})}$. While this construction requires $n \log q$ random bits, not too far from a uniformly random linear code, its structure allows for efficient decoding, and this was already used in [GI04, Rud07, GR10, HRW19].

To see this, let $w \in \mathbb{F}_2^n$ be the corrupt word we wish to decode, and break it up into $w = (w_1, \ldots, w_{n_{\mathrm{out}}})$, each $w_i \in \mathbb{F}_2^{n_{\mathrm{in}}}$. Since we think of $n_{\mathrm{in}}$ as small, we can brute-force list-decode, and find a list $S_i \subseteq \mathbb{F}_q$ such that for any $\alpha \in S_i$, $\mathcal{C}_{\mathrm{in}}^{(i)}(\alpha)$ is close to $w_i$, for some suitable chosen closeness parameter. Since originally, $w$ was close to some $\mathcal{C}(x)$ (or to some $\mathcal{C}(x_1), \ldots, \mathcal{C}(x_L)$ in the case of list decoding), and since most inner codes are good in the sense that they also achieve list decoding capacity, one can show that many of the $S_i$-s will be small, say $1 - \rho$ fraction of them.

We are thus left with the following task: Given $S_1, \ldots, S_{n_{\mathrm{out}}}$, where at least $1 - \rho$ fraction of them have size at most $\ell = \ell(\varepsilon)$, return the set of $x$-s for which $\mathcal{C}_{\mathrm{out}}(x)_i \in S_i$. This is (one variant of) the *list recovery* problem, a fascinating notion which warrants its own discussion. Hemenway,

[17] Notice that the concatenation property is pessimistic, and lower bounds the weights of each $\mathcal{C}_{\mathrm{in}}(\alpha)$, where $\alpha \in \mathbb{F}_q \setminus \{0\}$, by the *minimal* distance of $\mathcal{C}_{\mathrm{in}}$. Soon, we will ask whether codewords of $\mathcal{C}_{\mathrm{out}}$ can be so adversarial.

Ron-Zewi, and Wootters [HRW19] constructed good list recoverable codes, which led to the following decoding result of the Thomessen construction.

**Theorem 4.1** ([HRW19])**.** *There exists a family of $\varepsilon$-balanced binary codes $\mathcal{C} \subseteq \mathbb{F}_2^n$ of rate $\Omega(\varepsilon^2)$ that can be constructed in probabilistic polynomial time, and there exists a randomized algorithm that uniquely decodes $\mathcal{C}$ up to half the distance in time $c(\varepsilon) \cdot n^{1+1/t}$, where $t \approx \log\log\log n$, and $c(\varepsilon)$ is triply-exponential in* $\mathrm{poly}(1/\varepsilon)$*. The decoding algorithm was later derandomized in [KRRZ+20].*

The [HRW19] construction can also be used for list decoding up to the Zyablov bound. We note that *unique* decoding up to the Zyablov bound[18] can by done using the classical Generalized Minimum Distance (GMD) decoding due to Forney [For66].

**A Single Inner Code.** Being unsatisfied with a construction that uses many random bits, and hoping to have some path towards explicit construction of rate $\Omega(\varepsilon^2)$, we can go back to the idea of using a single inner code, and ask: Are there any concatenated (linear) codes $\mathcal{C}_{\text{out}} \circ \mathcal{C}_{\text{in}}$ that meet the GV bound with high probability over $\mathcal{C}_{\text{in}}$? If so, what are the conditions on $\mathcal{C}_{\text{out}}$ that will guarantee this?

Concatenating a random $\mathcal{C}_{\text{out}}$ with a *fixed-sized* random $\mathcal{C}_{\text{in}}$ was initially studied by Barg, Justesen, and Thommesen [BJT01], who showed that $\mathcal{C}_{\text{out}} \circ \mathcal{C}_{\text{in}}$, for random $\mathcal{C}_{\text{out}}$ and $\mathcal{C}_{\text{in}}$, approaches the GV bound in some cases (see also [BM10]). They also demonstrate suitable $\mathcal{C}_{\text{in}}$-s of small constant size. For low-rate codes of arbitrary lengths, the recent work of Doron, Mosheiff, and Wootters [DMW24], shows that most codes $\mathcal{C}_{\text{out}}$ are good.

**Theorem 4.2** ([DMW24])**.** *Suppose that $\mathcal{C}_{\text{out}} \subseteq \mathbb{F}_q^{n_{\text{out}}}$ and $\mathcal{C}_{\text{in}} \subseteq \mathbb{F}_2^{n_{\text{in}}}$ are random linear codes of rate $\varepsilon$ so that $q \geq 2^{\Omega(\varepsilon^{-3})}$. Then, $\mathcal{C} = \mathcal{C}_{\text{out}} \circ \mathcal{C}_{\text{in}}$ has rate $\varepsilon^2$, and with high probability, the relative distance of $\mathcal{C}$ is at least $\frac{1}{2} - O(\varepsilon)$.*

Note that while both codes are random, a codeword $c \in \mathcal{C}_{\text{out}} \circ \mathcal{C}_{\text{in}}$ is *not* uniform over $\mathbb{F}_2^n$, and hence Theorem 4.2 is (seemingly) not trivial.

Proving Theorem 4.2 uses a moments-based argument. Let $\{b_1, \ldots, b_{n_{\text{in}}}\}$ be the rows of the generating matrix $G_0$ of $\mathcal{C}_{\text{in}}$,[19] each of length $\log q$. For any nonzero message $m \in \mathbb{F}_q^{k_{\text{out}}}$, let $c = (\mathcal{C}_{\text{out}} \circ \mathcal{C}_{\text{in}})(m) \in \mathbb{F}_2^n$, and one can verify that the bias of $c$ can be expressed as

$$X_m = \sum_{i \in [n_{\text{out}}]} \sum_{j \in [n_{\text{in}}]} (-1)^{\langle \mathcal{C}_{\text{out}}(m)_i, b_j \rangle},$$

where we think of each $\mathbb{F}_q$-symbol as a $k_{\text{in}} = \log q$ bit-string. The goal is now to show that $|X_m| = O(\varepsilon n)$, and this is done by bounding $\mathbb{E}_{m \neq 0}[X_m^r]$ for some large enough $r$. If the bound is sufficiently small, a simple application of Markov's inequality, followed by a union bound, would show that there are no messages with bias above our desired $O(\varepsilon n)$ bound. In fact, in Theorem 4.2, we don't need $\mathcal{C}_{\text{in}}$ to be random, we just need $\mathcal{C}_{\text{in}}^{\perp}$ to have, approximately the "right" weight distribution.[20]

[18] Or more generally, up to half the distance that is given to us by the concatenation property, namely $d(\mathcal{C}_{\text{out}}) \cdot d(\mathcal{C}_{\text{in}})$.

[19] That is, $G_0 \in \mathbb{F}_2^{n_{\text{in}} \times k_{\text{in}}}$ represents the linear transformation $\mathcal{C}_{\text{in}} \colon \mathbb{F}_2^{k_{\text{in}}} \to \mathbb{F}_2^{n_{\text{in}}}$, and recall that $k_{\text{in}} = \log q$.

[20] Namely, that the number of $c \in \mathcal{C}_{\text{in}}^{\perp}$ of weight $i$ is roughly $\binom{n_{\text{in}}}{i} \cdot 2^{\varepsilon n_{\text{in}}}$. Establishing the bound on $\mathbb{E}_{m \neq 0}[X_m^r]$ first goes through expressing it in terms of $\mathcal{C}_{\text{out}}^{\perp}$, the dual subspace of $\mathcal{C}_{\text{out}}$, as is often the case in Fourier-analytic proofs. A bit more specifically, we need to count the number vectors in $\mathcal{C}_{\text{out}}^{\perp}$ that can arise from certain combinations of at most $r$ rows of $G_0$ in each coordinate.

Importantly, the proof technique in [DMW24] suggests future avenues towards making $\mathcal{C}_{\text{out}}$ *explicit*. Note that by making $\mathcal{C}_{\text{out}}$ explicit and choosing $\mathcal{C}_{\text{in}}$ uniformly at random, we only need to invest $n_{\text{in}} \cdot \log q$ random bits in order to generate $\mathcal{C}$. The first avenue towards constructing an explicit $\mathcal{C}_{\text{out}}$, is requiring that $\mathcal{C}_{\text{out}}^{\perp}$ satisfy some good list decodability from *soft information*, wherein one gets a distribution $\mathcal{D}_i \sim \mathbb{F}_q$ for each coordinate, representing some prior information about the $i$-coordinate, and the goal is bounding the probability that a word drawn from $\mathcal{D}_1 \times \ldots \times \mathcal{D}_n$ is a codeword. In [DMW24], they define a distribution $\mathcal{D} \sim \mathbb{F}_q$, so that if

$$\Pr_{x \sim \mathcal{D}^n}\left[x \in \mathcal{C}_{\text{out}}^{\perp} \setminus \{0\}\right] \leq (1+\Delta)q^{-k_{\text{out}}}$$

for some small $\Delta$, then $\mathcal{C}_{\text{out}} \circ \mathcal{C}_{\text{in}}$ lies on the GV bound with high probability over just a random $\mathcal{C}_{\text{in}}$. Note that the $q^{-k_{\text{out}}}$ term is the probability that a completely random vector lies in $\mathcal{C}_{\text{out}}^{\perp}$.

The second sufficient condition on $\mathcal{C}_{\text{out}}$, an arguably very natural one, concerns the symbol distribution of codewords, and requires the codewords of $\mathcal{C}_{\text{out}}$ to be smooth enough, meaning roughly, that every nonzero codeword has a fairly uniform distribution of symbols from $\mathbb{F}_q$. To illustrate why smoothness is desirable, let us consider the two extremes. The bad extreme is when there exists a codeword $c$ that is supported on only a single symbol, say $c = (\sigma, \ldots, \sigma)$ for some $\sigma \in \mathbb{F}_q$. Then, the relative weight of $c \circ \mathcal{C}_{\text{in}}$, for a random $\mathcal{C}_{\text{in}}$ of rate $\varepsilon$, might be $\frac{1}{2} - \Omega(\sqrt{\varepsilon})$, much worse than the desired $\frac{1}{2} - O(\varepsilon)$. The other, optimistic, extreme is where the symbol distribution of each nonzero codeword of $\mathcal{C}_{\text{out}}$ is uniform over $\mathbb{F}_q$. In this case, one can show that $\mathcal{C}_{\text{out}} \circ \mathcal{C}_{\text{in}}$ will approach the GV bound for a random $\mathcal{C}_{\text{in}}$.

It turns out that it suffices for every nonzero codeword $c$ to have a symbol distribution that has $\Theta(\varepsilon n_{\text{out}})$ copies of the same symbol (say, the zero symbol), while the remaining symbols in $c$ are uniformly distributed over a set of size only $q^{1-\varepsilon}$. A more general condition can be phrased in terms of the symbol distribution's "smooth min-entropy" (see [DMW24] for the details).

We note that instead of aiming for a single $\mathcal{C}_{\text{in}}$, one can consider an explicit $\mathcal{C}_{\text{out}}$, and *derandomize* the Thomessen construction, that is, generate (possibly distinct) $\mathcal{C}_{\text{in}}^{(1)}, \ldots, \mathcal{C}_{\text{in}}^{n_{\text{out}}}$ using few random bits. A randomness-efficient ensemble of codes is given in the well-known Justesen construction of asymptotically good codes [Jus72], but it is not clear how to utilize this construction to get the binary code we want. We thus conclude with the following open question.

**Open Problem 5.** *Find an explicit $\mathcal{C}_{\text{out}}$ such that $\mathcal{C}_{\text{out}} \circ \mathcal{C}_{\text{in}}$ has rate $\Omega(\varepsilon^2)$ and relative distance $\frac{1}{2} - O(\varepsilon)$ with high probability over a uniform $\mathcal{C}_{\text{in}}$, in the regime where $n_{\text{in}} \ll n_{\text{out}}$. More generally, give an efficient probabilistic construction of a binary code that approaches the GV bound and uses $o(n)$ random bits.*

Interestingly, any such randomness-efficient construction would need to evade the naive union bound over all $q^{k_{\text{out}}}$ codewords. Finally, we note that a Thomessen-like construction was shown to achieve *list-decoding capacity* by Guruswami and Rudra [GR08a], where $\mathcal{C}_{\text{out}}$ is a uniformly random code, or the folded Reed–Solomon code. It would thus be interesting to study the capabilities of single inner code concatenations to achieve list decoding capacity.

## 5 Conclusion

We surveyed some of the recent developments, and some important classical results, related to binary codes in the large distance regime, focusing on *combinatorial* properties (i.e., rate-distance

tradeoff), and *decoding* from adversarial errors. We note that other exciting, and sometimes related, progress has been recently made in studying *locality* (e.g., locally testable codes and relaxed locally correctable codes), decoding of good codes over larger alphabets, and decoding good binary codes in other corruption models. We did not attempt to cover those topics here.

We presented several open problems, starting from purely combinatorial, through coming up with explicit constructions, and designing efficient decoding algorithms. One important takeaway is that while the dependence on $\varepsilon$ in the rate of the code is now near-optimal, the dependence on $\varepsilon$ in things such as the runtime or list-decoding parameters, still leaves much to be desired. The small $\varepsilon$ regime is important for pseudorandomness, and we believe that making progress on that front will come with novel techniques, potentially contributing to other areas in coding theory.

### Acknowledgments

We would like to thank Fernando Granha Jeronimo and João Ribeiro for helpful comments on an earlier draft of this column. We also thank Mary Wootters for various discussions on related topics. Dean Doron is supported in part by NSF-BSF grant #2022644.

## 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. *Information Theory, IEEE Transactions on*, 38(2):509–516, 1992.

[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.

[AJQ+20] Vedat Levi Alev, Fernando Granha Jeronimo, Dylan Quintana, Shashank Srivastava, and Madhur Tulsiani. List decoding of direct sum codes. In *Proceedings of the 14th Annual Symposium on Discrete Algorithms (SODA)*, pages 1412–1425. SIAM, 2020.

[AJT19] Vedat Levi Alev, Fernando Granha Jeronimo, and Madhur Tulsiani. Approximating constraint satisfaction problems on high-dimensional expanders. In *Proceedings of the 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 180–201. IEEE, 2019.

[AN04] Noga Alon and Assaf Naor. Approximating the cut-norm via Grothendieck's inequality. In *Proceedings of the 36th Annual Symposium on Theory of Computing (STOC)*, pages 72–80. ACM, 2004.

[BD22] Guy Blanc and Dean Doron. New near-linear time decodable codes closer to the GV bound. In *Proceedings of the 37th Computational Complexity Conference (CCC)*, volume 234, pages 10:1–10:40. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2022.

[BJT01] Alexander Barg, Jørn Justesen, and Christian Thommesen. Concatenated codes with fixed inner code and random outer code. *IEEE Transactions on Information Theory*, 47(1):361–365, 2001.

[BM10] Alexander Barg and Arya Mazumdar. Small ensembles of sampling matrices constructed from coding theory. In *IEEE International Symposium on Information Theory (ISIT)*, pages 1963–1967. IEEE, 2010.

[Bog12] Andrej Bogdanov. Topics in (and out) the theory of computing: Lecture notes. https://andrejb.net/csc5060/notes/12L12.pdf, 2012. [Online; accessed July-2024].

[BT11] Avraham Ben-Aroya and Amnon Ta-Shma. A combinatorial construction of almost-Ramanujan graphs using the zig-zag product. *SIAM Journal on Computing*, 40(2):267–290, 2011.

[BT13] Avraham Ben-Aroya and Amnon Ta-Shma. Constructing small-bias sets from algebraic-geometric codes. *Theory of Computing*, 9(5):253–272, 2013.

[Coh22] Gil Cohen. Algebraic geometric codes: Lecture notes. https://www.gilcohen.org/2022-ag-codes, 2022. [Online; accessed July-2024].

[DH24] Yotam Dikstein and Max Hopkins. Chernoff bounds and reverse hypercontractivity on HDX. In *Proceedings of the 65th Annual Symposium on Foundations of Computer Science (FOCS), to appear*. IEEE, 2024.

[DHK+21] Irit Dinur, Prahladh Harsha, Tali Kaufman, Inbal Livni Navon, and Amnon Ta-Shma. List-decoding with double samplers. *SIAM Journal on Computing*, 50(2):301–349, 2021.

[DMW24] Dean Doron, Jonathan Mosheiff, and Mary Wootters. When do low-rate concatenated codes approach the Gilbert–Varshamov bound? In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM), to appear*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2024.

[Dor24] Dean Doron. Complexity theory column: Binary codes with distance close to half. *ACM SIGACT News*, 55(3):32–51, 2024.

[DPT24] Dean Doron, Edward Pyne, and Roei Tell. Opening up the distinguisher: A hardness to randomness approach for $\mathbf{BPL} = \mathbf{L}$ that uses properties of $\mathbf{BPL}$. In *Proceedings of the 56th Annual Symposium on Theory of Computing (STOC)*, pages 2039–2049. ACM, 2024.

[For66] G. David Forney. Generalized minimum distance decoding. *IEEE Transactions on Information Theory*, 12(2):125–131, 1966.

[GGH+07] Shafi Goldwasser, Dan Gutfreund, Alexander Healy, Tali Kaufman, and Guy Rothblum. Verifying and decoding in constant depth. In *Proceedings of the 39th Annual Symposium on Theory of Computing (STOC)*, pages 440–449. ACM, 2007.

[GI04] Venkatesan Guruswami and Piotr Indyk. Efficiently decodable codes meeting gilbert-varshamov bound for low rates. In *Proceedings of the 15th Symposium on Discrete Algorithms (SODA)*, pages 756–757. ACM-SIAM, 2004.

[Gil52] Edgar N. Gilbert. A comparison of signalling alphabets. *The Bell system technical journal*, 31(3):504–522, 1952.

[Gol11] Oded Goldreich. A sample of samplers: A computational perspective on sampling. In *Studies in Complexity and Cryptography*, pages 302–332. Springer, 2011.

[GR08a] Venkatesan Guruswami and Atri Rudra. Concatenated codes can achieve list-decoding capacity. In *Proceedings of the 19th Symposium on Discrete Algorithms (SODA)*, pages 258–267. ACM-SIAM, 2008.

[GR08b] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on Information Theory*, 54(1):135–150, 2008.

[GR10] Venkatesan Guruswami and Atri Rudra. The existence of concatenated codes list-decodable up to the hamming bound. *IEEE Transactions on Information Theory*, 56(10):5195–5206, 2010.

[GRS] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory*. http://www.cse.buffalo.edu/faculty/atri/courses/coding-theory/book.

[GS96] Arnaldo Garcia and Henning Stichtenoth. On the asymptotic behaviour of some towers of function fields over finite fields. *Journal of Number Theory*, 61(2):248–273, 1996.

[GS06] Arnaldo Garcia and Henning Stichtenoth. *Topics in geometry, coding theory and cryptography*, volume 6. Springer Science & Business Media, 2006.

[GSV18] Aryeh Grinberg, Ronen Shaltiel, and Emanuele Viola. Indistinguishability by adaptive procedures with advice, and lower bounds on hardness amplification proofs. In *Proceedings of the 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 956–966. IEEE, 2018.

[HH24] Pooya Hatami and William Hoza. Paradigms for unconditional pseudorandom generators. *Foundations and Trends® in Theoretical Computer Science*, 16(1-2):1–210, 2024.

[HRW19] Brett Hemenway, Noga Ron-Zewi, and Mary Wootters. Local list recovery of high-rate tensor codes and applications. *SIAM Journal on Computing*, pages FOCS17–157, 2019.

[JQST20] Fernando Granha Jeronimo, Dylan Quintana, Shashank Srivastava, and Madhur Tulsiani. Unique decoding of explicit $\varepsilon$-balanced codes near the Gilbert–Varshamov bound. In *Proceedings of the 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 434–445. IEEE, 2020.

[JST21] Fernando Granha Jeronimo, Shashank Srivastava, and Madhur Tulsiani. Near-linear time decoding of Ta-Shma's codes via splittable regularity. In *Proceedings of the 53rdth Annual Symposium on Theory of Computing (STOC)*, pages 1527–1536. ACM, 2021.

[JST23] Fernando Granha Jeronimo, Shashank Srivastava, and Madhur Tulsiani. List decoding of tanner and expander amplified codes from distance certificates. In *Proceedings of the 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1682–1693. IEEE, 2023.

[Jus72] Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18(5):652–656, 1972.

[KRRZ+20] Swastik Kopparty, Nicolas Resch, Noga Ron-Zewi, Shubhangi Saraf, and Shashwat Silas. On list recovery of high-rate tensor codes. *IEEE Transactions on Information Theory*, 67(1):296–316, 2020.

[MRRW77] Robert McEliece, Eugene Rodemich, Howard Rumsey, and Lloyd Welch. New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE Transactions on Information Theory*, 23(2):157–166, 1977.

[NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, 1993.

[NS09] Michael Navon and Alex Samorodnitsky. Linear programming bounds for codes via a covering argument. *Discrete & Computational Geometry*, 41:199–207, 2009.

[RR23] Silas Richelson and Sourya Roy. Gilbert and Varshamov meet Johnson: List-decoding explicit nearly-optimal binary codes. In *Proceedings of the 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 194–205. IEEE, 2023.

[Rud07] Atri Rudra. *List decoding and property testing of error-correcting codes*. University of Washington, 2007.

[Sti09] Henning Stichtenoth. *Algebraic function fields and codes*, volume 254. Springer Science & Business Media, 2009.

[STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001.

[SV22] Ronen Shaltiel and Emanuele Viola. On hardness assumptions needed for "extreme high-end" PRGs and fast derandomization. In *Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 215, pages 116:1–116:17. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2022.

[Ta-17] Amnon Ta-Shma. Explicit, almost optimal, $\varepsilon$-balanced codes. In *Proceedings of the 49th Annual Symposium on Theory of Computing (STOC)*, pages 238–251. ACM, 2017.

[Ta-24] Amnon Ta-Shma. The hermitian trace code: A lecute given at the "Advances in the theory of error-correcting codes" Simons workshop. https://simons.berkeley.edu/talks/amnon-ta-shma-tel-aviv-university-2024-04-08, 2024. [Online; accessed July-2024].

[Tho83] Christian Thommesen. The existence of binary linear concatenated codes with Reed–Solomon outer codes which asymptotically meet the Gilbert–Varshamov bound. *IEEE Transactions on Information Theory*, 29(6):850–853, 1983.

[TZ04] Amnon Ta-Shma and David Zuckerman. Extractor codes. *IEEE Transactions on Information Theory*, 50(12):3015–3025, 2004.

[Var57] Rom Rubenovich Varshamov. Estimate of the number of signals in error correcting codes. *Docklady Akad. Nauk, SSSR*, 117:739–741, 1957.

[Zya71] Victor Vasilievich Zyablov. An estimate of the complexity of constructing binary linear cascade codes. *Problemy Peredachi Informatsii*, 7(1):5–13, 1971.



ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il