Linear Time Encodable Binary Code Achieving GV Bound with Linear Time Encodable Dual Achieving GV Bound

Martijn Brehm
Informatics Institute, University of Amsterdam, The Netherlands

Nicolas Resch
Informatics Institute, University of Amsterdam, The Netherlands

Abstract

We initiate the study of what we term "fast good codes" with "fast good duals." Specifically, we consider the task of constructing a binary linear code $\mathcal{C} \leq \mathbb{F}_2^n$ such that both it and its *dual* $\mathcal{C}^\perp := \{x \in \mathbb{F}_2^n : \forall c \in \mathcal{C}, \langle x, c\rangle = 0\}$ are asymptotically good (in fact, have rate-distance tradeoff approaching the GV bound), and are encodable in $O(n)$ time. While we believe such codes should find applications more broadly, as motivation we describe how such codes can be used the secure computation task of *encrypted matrix-vector product*, as studied by Behhamouda et al (CCS 2025).

Our main contribution is a construction of such a fast good code with fast good dual. Our construction is inspired by the repeat multiple accumulate (RMA) codes of Divsalar, Jin and McEliece (Allerton, 1998). To create the rate 1/2 code, after repeating each message coordinate, we perform *accumulation steps* – where first a uniform coordinate permutation is applied, and afterwards the prefix-sum modulo 2 is applied – which are alternated with *discrete derivative steps* – where again a uniform coordinate permutation is applied, and afterwards the previous two coordinates are summed modulo 2. Importantly, these two operations are inverse of each other. In particular, the dual of the code is very similar, with the accumulation and discrete derivative steps reversed.

Our analysis is inspired by a prior analysis of RMA codes due to Ravazzi and Fagnani (IEEE Trans. Info. Theory, 2009). The main idea is to bound the *input-output weight-enumerator function*: the expected number of messages of a given weight that are encoded into a codeword of a given weight. We face new challenges in controlling the behaviour of the discrete derivative matrix (which can significantly drop the weight of a vector), which we overcome by careful case analysis.

2012 ACM Subject Classification Theory of computation → Error-correcting codes

Keywords and phrases Binary error-correcting codes, dual codes, fast encoding, repeat-multiple-accumulate codes

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.28

Related Version *Full Version*: `https://arxiv.org/abs/2509.07639` [7]

Funding *Nicolas Resch*: Research supported by an NWO (Dutch Research Council) grant with number C.2324.0590. This work was done in part while visiting the Simons Institute for the Theory of Computing, supported by DOE grant #DE-SC0024124.

Acknowledgements The authors would like to thank Yuval Ishai, for patient discussions concerning potential cryptographic applications of our codes; Geoffroy Couteau, for sharing the construction of the codes that we analyze; Divya Ravi, for kindly reading over the cryptographic sections and suggesting relevant citations; the anonymous reviewers for helpful comments; and Henri Pfister for suggesting overlooked citations on RMA codes.

1 Introduction

The theory of error-correcting codes is largely concerned with constructing linear subspaces of finite vector spaces satisfying interesting combinatorial properties. In this work, we focus on the most popular setting of binary codes, i.e., subspaces $\mathcal{C} \leq \mathbb{F}_2^n$. A first desirable

property of a code $\mathcal{C}$ is that the codewords (i.e., elements of $\mathcal{C}$) are well-spread. To quantify this, one typically uses the *minimum distance* $\delta(\mathcal{C})$, defined as the minimum fraction of coordinates one needs to change in order to transform one codeword into another one. In other words, $\delta(\mathcal{C}) := \min\{d(c,c') : c,c' \in \mathcal{C}, c \neq c'\}$, where $d(x,y) := \frac{1}{n}|\{i \in [n] : x_i \neq y_i\}|$ denotes the (relative) *Hamming distance*. In fact, for linear codes, it suffices to look at the minimum distance of a nonzero codeword from 0, i.e., $\delta(\mathcal{C}) = \min\{\mathrm{wt}(c) : c \in \mathcal{C} \setminus \{0\}\}$, where $\mathrm{wt}(x) = \frac{1}{n}|\{i \in [n] : x_i \neq 0\}|$ is the (relative) *Hamming weight*.

On the other hand, one would like the linear code to be fairly large. This is typically quantified by the code's *rate*, defined as $R(\mathcal{C}) = \frac{k}{n}$ where $k = \dim(\mathcal{C})$. The well-known *Gilbert-Varshmov (GV)* bound states that such binary linear codes of rate R and distance δ exist whenever $R \leq 1 - h(\delta)$, where $h(x) := x \log \frac{1}{x} + (1-x) \log \frac{1}{1-x}$ denotes the binary entropy function, which has a continuous inverse $h^{-1} : [0, 1/2] \to [0, 1]$. For binary codes, the GV bound remains the best known achievable tradeoff between rate and distance. We will be looking for codes that get close to the GV bound, as in the following definition.

▶ **Definition 1** (GV Bound). *Fix $R \in (0,1)$ and $\varepsilon > 0$. We define $\delta^{(\mathrm{GV})}(R) := h^{-1}(1-R)$, and say a code $\mathcal{C}$ of rate R is* ε-close *to the GV bound if $\delta(\mathcal{C}) \geq \delta^{(\mathrm{GV})}(R) - \varepsilon$.*

Beyond hoping for large rate and large distance, one can make other demands on a code. For example, one could hope for *encoding* to be very fast. That is, recalling that for every linear code $\mathcal{C}$ one can define a *generator matrix* $G \in \mathbb{F}_2^{n \times k}$ for which $\mathcal{C} = \{Gm : m \in \mathbb{F}_2^k\}$, one could hope that the time it takes to compute Gm from m is as small as possible, ideally $O(n)$.[1]

Additionally, especially in the context of cryptography the *dual* code of a binary linear code $\mathcal{C}$ is also often crucial (being connected to, e.g., the secrecy threshold in secret-sharing schemes). Recall that the dual of a code $\mathcal{C}$ is defined as $\mathcal{C}^\perp := \{x \in \mathbb{F}_2^n : \langle x, c\rangle = 0,\ \forall c \in \mathcal{C}\}$, where for $x, y \in \mathbb{F}_2^n$ we have $\langle x, y\rangle = \sum_{i=1}^n x_i y_i$, the standard inner-product modulo 2. Recall that if $\dim(\mathcal{C}) = k$ then $\dim(\mathcal{C}) = n - k$, so if $\mathcal{C}$ has rate R then $\mathcal{C}^\perp$ has rate $1 - R$. We thus find it most natural to consider codes of rate $1/2$, so that $\mathcal{C}$ and $\mathcal{C}^\perp$ are of the same rate (and could potentially achieve the same distance).

In this work, we consider the task of constructing binary linear codes of rate $1/2$ meeting the following list of desiderata:

- Firstly, we would like $\delta(\mathcal{C})$ to approach the GV bound $\delta^{(\mathrm{GV})}(1/2) \approx 0.1100$.
- We would like the dual distance $\delta(\mathcal{C}^\perp)$ to also approach the GV bound $\delta^{(\mathrm{GV})}(1/2) \approx 0.1100$.
- Lastly, we would like both $\mathcal{C}$ and $\mathcal{C}^\perp$ to be encodable in $O(n)$ time (which, up to constants, is clearly optimal).

To coin a term, we call such a code a *fast good code* with *fast good dual*.

1.1 Our results

In this work, we prove that such a fast good code with fast good dual indeed exists.

▶ **Theorem 2.** *For all large enough (even) $n \in \mathbb{N}$ and all $\varepsilon > 0$, there exists a binary code $\mathcal{C} \leq \mathbb{F}_2^n$ of rate $1/2$ such that:*

- *$\mathcal{C}$ and $\mathcal{C}^\perp$ both have distance at least $\delta^{(\mathrm{GV})}(1/2) - \varepsilon$;*
- *$\mathcal{C}$ and $\mathcal{C}^\perp$ are both encodable in $O_\varepsilon(n)$ time.*

[1] We quantify encoding time in a circuit model, where all fan-in 2 gates are available. For additional details on this point, see Section 2.

As we discuss later, the encoding time is in fact practically fast. For example, with $\varepsilon = 0.0001$ there is a boolean circuit with binary gates implementing the encoding map with size $8(n-1)$. Unfortunately we do not now how to bound more precisely the encoding time of $\mathcal{C}$ and $\mathcal{C}^\perp$ in terms of ε. However, empirically the constant in front of n is very small; see Table 1 and the surrounding discussion. We additionally note that Boolean circuits of size $O_\varepsilon(n)$ and depth $O_\varepsilon(\log n)$ can implement the encoding map, should low depth be desired.

While we will provide a much broader overview of our proof technique later, we now briefly outline the idea. Firstly, we emphasize that our construction is *randomized*. Both $\mathcal{C}$ and $\mathcal{C}^\perp$ will be sampled in such a way that they are reminiscent of *repeat multiple accumulate (RMA)* codes. Essentially, *RMA* codes are defined by multiple rounds, where in each round a coordinate permutation is applied, followed by an *accumulation* step, which is just the prefix-sum modulo 2. In particular, these rounds can be implemented very quickly (just $n-1$ XOR's). The idea is to consider *random RMA* codes, where the coordinate permutations are chosen uniformly at random, and demonstrate they are likely to have very good distance.

Fortunately for us, a sequence of works [22, 23, 2, 18, 24, 6] have shown these codes achieve good distance. The basic idea of these works is, given two weights α and β, to analyze the expected number of message vectors of weight α that are mapped to a codeword of weight β. Using some coding-theoretic jargon, this is called the *input-output weight-enumerator function*, or *IOWEF* for short. Concretely, for a fixed generator matrix $G \in \mathbb{F}_2^{k\times n}$ we define

$$N_G(\alpha,\beta) := \{m \in \mathbb{F}_2^k : \mathrm{wt}(m) = \alpha \text{ and } \mathrm{wt}(Gm) = \beta\} .$$

Then, if $\mathbf{G}$ denotes a random generator matrix (sampled according to some given distribution), the IOWEF will be $\mathbb{E}_{\mathbf{G}}[N_{\mathbf{G}}(\alpha,\beta)]$. The pleasant feature of this code ensemble is that its IOWEF has a fairly simple expression. To show a code achieves minimum distance δ with probability $1-p$ for some $p \in (0,1)$, by Markov's inequality it suffices to show that the sum of the IOWEF over all α, β with $\beta \le \delta$ is $\le p$.

Now, it is not difficult to see that an *RMA* code has constant weight dual codewords (assuming the number of rounds is constant, as we always do). Hence, they are unsuitable for our target. Nonetheless, we note that the inverse of the accumulation step is quite simple: namely, one just takes the XOR of the previous two coordinates. We view this operation as a *discrete derivative*. While this operation is not very helpful in terms of achieving good distance (it can at most double the weight of a vector, and in the worst case can drop it all the way down to $1/n$), as it corresponds to doing an accumulation step in the dual, it increases the dual distance! Thus, we can hope for both distance and dual distance to be large. A main challenge is to argue that these discrete derivative rounds are unlikely to *harm* the minimum distance.

Thus, our task is to bound the IOWEF of codes where accumulation rounds are interleaved with discrete derivative rounds. As in prior works [18, 24, 6], roughly speaking we break the analysis into the case where (α,β) are small or large. The first case is handled largely by combinatorial reasoning concerning binomial coefficients; the latter case is handled by looking at the exponent of the IOWEF, i.e., looking at $\frac{1}{n}\log$ of the IOWEF and bounding it via analytic means: this function is often called the *spectral shape function*. In some sense, we manage to bound the spectral shape functions of our code ensembles by that of an *RMA* code, allowing us to obtain the same distance guarantees. This is the sense in which we manage to show the discrete derivative rounds don't harm minimum distance. In particular, we can get distance ε-close to the GV bound, where the closeness to the GV bound is determined by the number of "rounds" of our encoding maps.

We believe that the existence of such a code is interesting, and likely to find further applications in coding theory, cryptography, or theoretical computer science more broadly. As a small proof of concept, we show their utility for the task of computing *encrypted matrix-vector product (EMVP)*, as studied by Benhamouda et al [4]. The task is as follows. In an initial offline phase, a client sends an encryption $\hat{M}$ of a matrix $M \in \mathbb{F}^{m \times \ell}$ to a server, while keeping a (short) secret key. In the online phase, the client may send encryptions $\hat{q}$ of query vectors $q \in \mathbb{F}^{\ell}$; based on these encryptions, the server computes a value M', which the client can use to determine the matrix-vector product Mq. Importantly, the server learns nothing about the matrix M or the query vectors q. As Benhamouda et al [4] discuss, this is an important subroutine for efficient secure computation tasks in a number of domains, including encrypted fuzzy search and secure ML. Additionally, note that the special case of each q being a unit vector (i.e., having a single 1 entry) is equivalent to (single-server) private information retrieval (PIR). While we provide more details later of this application in Section 4, the fast encoding time for the $\mathcal{C}$ and $\mathcal{C}^{\perp}$ we construct yield improved running times for the protocol of Benhamouda et al [4].

Again, we emphasize that this is just a small proof-of-concept for our codes. We consider the main contribution of this work to be the code construction itself, along with the (fairly involved) analysis. Nonetheless, we flesh out this application further in Section 4.

1.2 Related work

To the best of our knowledge, ours is the first work to construct "fast good codes" with "fast good duals." However, our codes are heavily inspired by so-called *repeat-multiple-accumulate (RMA)* codes. These codes were initially introduced as containing only a single round of permuting and accumulating by Divsalar, Jin and McEliece [11], and then extended to contain multiple rounds by Pfister and Siegel in an 1999 conference paper [22]. A series of works [23, 18, 24, 6] (that we build upon) have managed to show that such codes can have minimum distance approaching the GV bound[2] along with linear-time encoding. We will introduce these codes more formally later (Section 2.1) and discuss in detail how we build upon the analyses of these works. We remark that the codes in these works are, like ours, non-explicit.

Regarding the task of constructing *explicit* linear-time encodable binary codes that are *asymptotically good* (namely, rate and minimum distance are positive constants), Spielman [27] provided a construction based on expander codes; these codes can also be *decoded* in linear time. Many works have built off this construction, allowing for more and more sophisticated decoding guarantees; however, this is somewhat tangential from our main focus. Note that the proven rate/distance tradeoff of Spielman's code is quite far from the GV bound. Guruswami and Indyk [13] are the first to construct linear-time encodable codes achieving the GV bound, but only over sufficiently large fields (and in fact, they work in the regime where the distance can be $1 - R - \varepsilon$, i.e., roughly the Singleton bound).

A work in the same vein as ours is by Druk and Ishai [12]; therein, the authors give a (randomized, non-explicit) construction of a code meeting the GV bound which is linear-time encodable, and additionally outline some cryptographic applications of these codes. Finally, we mention a work by Rudra and Wootters [25] which, among other results, demonstrates the existence of linear-time encodable binary codes of rate $\Omega(\varepsilon^2)$ that are list-decodable up

[2] There are some minor caveats concerning the speed at which one approaches the GV bound which we discuss later, but for practical purposes such codes achieve essentially this rate/distance tradeoff.

to radius $1/2 - \varepsilon$, where we recall that a code $\mathcal{C} \subseteq \mathbb{F}_2^n$ is called (ρ, L)-list-decodable if from every vector $z \in \mathbb{F}_2^n$ there are at most L codewords $c \in \mathcal{C}$ satisfying $d(c, z) \leq \rho$ (the list-size L in this work is a constant depending only on ε). In fact, this code is obtained by randomly folding Spielman's linear-time encodable code.

We remark that such prior constructions of linear time encodable codes typically have sparse parity-checks; in particular, the duals cannot be asymptotically good. This is a challenge that we overcome, by carefully adding encoding steps specifically designed to improve the dual distance while also not harming the original code's distance.

1.3 Future directions

Before moving onto the technical portion of our paper, we now take some time to highlight some open problems that we consider worthy of study. Firstly, our analysis is inherently limited to the binary field case. One could naturally define similar codes over larger fields: the main thing which should change is that instead of just randomly permuting, one should also randomly *rescale* the entries, i.e., multiply by a uniformly random full-rank diagonal matrix.[3] However, even for the case of RMA codes most analyses up till now have focused on the binary case. The simple reason for this is that, once $q > 3$ the IOWEF is in fact *exponentially large*;[4] thus, an argument simply applying Markov's inequality cannot succeed. This is elucidated in the work of Block et al [5], where the authors managed to develop some nontrivial analysis of the minimum distance of RA codes over large fields that *does not* directly use Markov's inequality. Nonetheless, the current best argument proving that RMA codes over large alphabets have nontrivial distance follows from simply analyzing the code over $\mathbb{F}_2$, and then using the same generator matrix to define a code over $\mathbb{F}_{2^t}$, i.e., some (large) extension of $\mathbb{F}_2$ [6]. It would be very interesting to find an analysis over large fields that leads to improved rate/distance tradeoffs.

Continuing with questions concerning RMA codes, we now have a fairly good understanding of their minimum distance; in particular, Brehm et al [6] provide a fairly thorough investigation of the probability they fail to achieve good distance, demonstrating they achieve good distance for concrete values of n. A next task could be to determine what *list-decodability* they might possess. A reasonable approach for this could be to show that they are *locally similar to random linear codes* [15, 21], which would (roughly) imply that they inherit the list-decodability of a typical linear code.[5] It would additionally be very interesting to find *explicit* RMA codes achieving nontrivial distance; to the best of our knowledge the only result in this vein is due to Guruswami and Machmouchi [14], which essentially partially derandomizes the construction of an asymptotically good RAA code (namely, it requires only the second permutation to be sampled randomly; the first can be deterministically chosen). Finally, we remark that every question asked above could additionally be asked for the codes we define and study.

Next, we list a couple interesting questions motivated by cryptographic applications. Specifically, we consider the code property of *multiplicativity*. Given two vectors $x, y \in \mathbb{F}^n$, we define their *(Schur) product* as $x * y := (x_1 \cdot y_1, \ldots, x_n \cdot y_n)$ as the component-wise

[3] Note that over $\mathbb{F}_2$, the only full-rank diagonal matrix is the identity; hence, this is a natural generalization.
[4] For $q = 3$ as well it is unclear how to bound the IOWEF; already for this field size, new ideas appear to be required.
[5] Technically, RMA will not be fully locally similar to random linear codes, as they contain constant-weight codewords with an overly large probably $1/\text{poly}(n)$. The same issue arises with LDPC codes, but can be resolved [20].

product, and then given two linear codes $\mathcal{C}$ and $\mathcal{D}$ their (Schur) product is defined as $\mathcal{C} * \mathcal{D} := \text{span}\{c * d : c \in \mathcal{C}, d \in \mathcal{D}\}$. A code pair $(\mathcal{C}, \mathcal{D})$ is said to be *multiplicative* if $\mathcal{C} * \mathcal{D} \neq \mathbb{F}^n$; namely, it does not span the whole space. Firstly, it can be observed that a binary linear code $\mathcal{C} \leq \mathbb{F}_2^n$ and its dual $\mathcal{C}^\perp$ are multiplicative, as $\mathcal{C} * \mathcal{C}^\perp$ only contains even weight vectors. It is known [9] that multiplicative codes yield multiplicative secret-sharing schemes,[6] and additionally that such a multiplicative secret-sharing scheme can be used for general-purpose secure multiparty computation (MPC) [3, 28]. The encoding efficiency of the codes $\mathcal{C}$ and $\mathcal{C}^\perp$ directly translate into the complexity of sharing a secret [12], which must be done by each party for each multiplication gate of the circuit representing the function one is securely computing. Thus, in the quest for "constant-overhead cryptography [16]" (where the hope is that, if the cost of computing a functionality *without* security is N, the cost of computing it *with* security is just $O(N)$) multiplicative secret-sharing scheme with linear-time encoding can play an important role. We observe that typical multiplicative secret-sharing schemes are "algebraic" (e.g., Shamir secret-sharing [26], which is based on polynomial evaluations); here, one can deal the shares in *quasilinear* time (using fast Fourier-transform-type ideas), but getting linear-time dealing appears to require more "combinatorial" approaches.

To summarize the above, our work provides a multiplicative secret sharing scheme over $\mathbb{F}_2$ with linear-time encoding. One could hope for more. Firstly, one can hope for higher-degree multiplicative secret-sharing [1] allowing for multiplying more secrets together, again with linear-time algorithms for dealing the shares. Next, one could additionally hope for *strong* multiplicativity of a secret-sharing scheme, which informally would follow from a pair of asymptotically good codes $\mathcal{C}, \mathcal{D}$ such that $\mathcal{C} * \mathcal{D}$ has *constant* distance (such codes are said to satisfy the *multiplication property*). Note that for a linear code $\mathcal{C}$ and its dual $\mathcal{C}^\perp$, as $\mathcal{C} * \mathcal{C}^\perp$ only contains even weight vectors, it has minimum distance $\geq 2/n$, which is nontrivial, but far from constant. Strongly multiplicative secret-sharing schemes lead to general MPC protocols with *malicious* security [9], i.e., security even in the presence of parties that may actively deviate from the prescribed protocol. One could hope to have such multiplication codes with linear time encoding towards getting general-purpose MPC with constant-computational overhead and malicious security. We remark that prior work has already considered the possibility of constructing somewhat "combinatorial" codes (i.e., codes that could admit linear time encoding) that additionally have the multiplication property [10].

Broadly speaking, we view our work as a stepping stone towards a broader suite of "fast codes" with interesting combinatorial properties. We emphasize that from a cryptographic perspective, decodability is often not required,[7] but instead ask for nontrivial conditions concerning, e.g., the dual code or multiplicativity.

1.4 Organization

The remainder of this paper is organized as follows. In Section 2 we set notation, define RMA codes and review their existing analysis, after which we define our pair of dual codes. In Section 3 we give a broad overview of our proof strategy for Theorem 2. We leave the details

[6] Briefly, secret-sharing schemes allow a dealer holding to a secret s to generate shares $(s_1, \ldots, s_n)$ to be given to n parties such that *authorized* sets of parties can reconstruct the secret, but no *unauthorized set* of parties can. As for multiplicativity, this can be viewed as an additional property allowing parties to locally convert shares of secrets a, b into a new value c_i such that the product ab is a linear combination of the c_i's.

[7] And in fact, in certain cases the security of the scheme rests on the assumed *hardness* of decoding; this is (essentially) the case for the encrypted matrix-vector product protocol we sketch in Section 4.

of this proof to Sections 4 and 5 in the full version of this paper [7]. Finally, in Section 4, we discuss the utility of these codes in the context of the encrypted matrix-vector product problem.

2 Preliminaries

Throughout this work we will use Latin letters $a, b, c, \ldots$ to refer to absolute weights (integers from 1 to n) and Greek letters α, β, γ to refer to relative weights (reals in $[0, 1]$). By default, log denotes the base-2 logarithm.

We use standard Landau notation, e.g., $O(\cdot)$, $\Omega(\cdot)$, $o(n)$, $\omega(n)$, etc. In all contexts the growing parameter will be n. We use $\mathrm{poly}(n)$ to refer to a function of the form $n^{O(1)}$ and $\mathrm{negl}(n)$ a function of the form $n^{-\omega(n)}$. The notation $\tilde{O}(\cdot)$ suppresses terms of the form $\log^{O(1)} n$.

Next, we recall Markov's inequality, and the specific way in which we use it in this work.

▶ **Theorem 3** (Markov's inequality). *Let X be a non-negative random variable. Then for any $\alpha > 0$,*

$$\mathbb{P}\big[X \geq \alpha\big] \leq \frac{\mathbb{E}[X]}{\alpha} \ .$$

In particular, if X takes values in $\mathbb{N} \cup \{0\}$ then

$$\mathbb{P}\big[X > 0\big] \leq \mathbb{E}[X] \ .$$

Now, recall that we defined the *dual* of a linear code $\mathcal{C} \leq \mathbb{F}_2^n$ of dimension k as

$$\mathcal{C}^\perp := \{x \in \mathbb{F}_2^n : \forall c \in \mathcal{C},\ \langle x, c\rangle = 0\} \ .$$

We recall the following standard facts. Firstly, $\dim(\mathcal{C}^\perp) = n - k$ and $(\mathcal{C}^\perp)^\perp = \mathcal{C}$. Additionally, if linear codes $\mathcal{C}$ and $\mathcal{D}$ are generated by matrices $G \in \mathbb{F}_2^{n\times k}$ and $H \in \mathbb{F}_2^{n\times(n-k)}$, respectively, in the sense that $\mathcal{C} = \{Gm : m \in \mathbb{F}_2^k\}$ and $\mathcal{D} = \{Hm : m \in \mathbb{F}_2^{n-k}\}$, then $\mathcal{C}^\perp = \mathcal{D}$ if and only if $H^\top G = 0$.

Lastly, as we wish to discuss encoding complexity of binary linear codes, we choose the standard model of Boolean circuits where gates of fan-in 2 are available.[8] We remark that this circuit model is robust to the precise set of available gates, but for simplicity we assume all gates are available. This should be contrasted with more liberal models such as arithmetic circuits or RAM models, which are more sensitive to ring or word size, respectively. Additional discussion of this model choice can be found in [27].

2.1 Repeat-multiple-accumulate RMA codes

2.1.1 Definition

The codes we study in this paper are directly inspired by the well-studied *repeat-multiple-accumulate (RMA) codes* over the binary field $\mathbb{F}_2$. These are parametrised by three integers $m, n, r \in \mathbb{N}$, where m is the number of *rounds* of the code, r is the repetition factor which fixes the rate $1/r$ of the code, and finally n is the *block-length*. We assume n is divisible by r.

[8] Of course, any other constant fan-in only affects the encoding complexities up to constants; the choice of 2 is just for concreteness.

The encoding function of an RMA code is simple to describe. First, take your message vector of length n/r and repeat it r times. Then apply a permutation to the resulting length n vector. Next, apply the accumulator operation to this vector, which is simply a prefix sum: bit i of the output will be equal to the XOR of the first i bits in the input vector. We repeat this permute-accumulate step m times. More formally, the three operations in the encoding can be described in terms of the following linear operations.

- For a constant $r \in \mathbb{N}$ dividing n, let $F_r \in \mathbb{F}_2^{n \times n/r}$ denote the *repetition matrix* corresponding to the linear operator that repeats each entry in the vector r times. That is, $F_r[i,j] = 1$ if and only if $\lfloor i/r \rfloor + 1 = j$.
- Let $A \in \mathbb{F}_2^{n \times n}$ be the *accumulator matrix*, $A_{ij} = 1$ if and only if $i \geq j$.
- For a permutation $\pi : [n] \to [n]$, let $M_\pi \in \mathbb{F}_2^{n \times n}$ be the *permutation matrix* corresponding to π. That is, $(M_\pi)_{ij} = 1$ if and only if $\pi(i) = j$.

Say we fix $m = 2$ and have $r \in \mathbb{N}$. We then obtain a rate $1/r$ RMA code with two permute-accumulate rounds, which we refer to as an RAA code. Its generator matrix $RAA_{\pi_1,\pi_2} : \mathbb{F}_2^{n/r} \to \mathbb{F}_2^n$ code can be defined as $RAA_{\pi_1,\pi_2}(x) = AM_{\pi_2}AM_{\pi_1}F_r\,x$. More generally, we will refer to an RMA code of m rounds as an RA^m code.

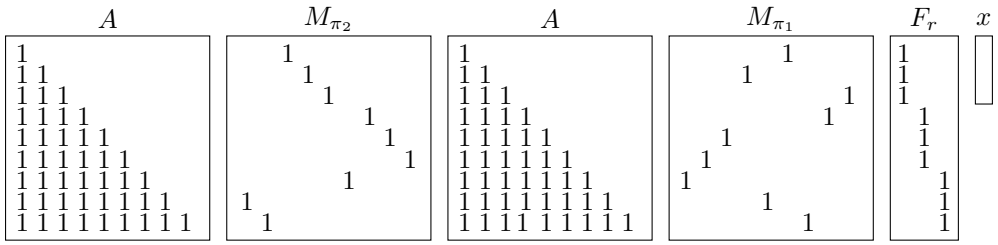

Figure 1 A pictorial representation of (a generator matrix for) an RAA code with rate $R = 1/3$ and block-length $n = 9$.

Note that computing the RA^m encoding can be done by a Boolean circuit consisting of just $m(n-1)$ XOR gates of fan-in 2; additionally, classical work of Ladner and Fischer [19] gives an implementation with $O(n)$ gates and depth $O(\log n)$ (and the constants are very reasonable).

Rather than considering a fixed choice of RA^m code, we will consider a *random* RA^m codes defined by sampling uniformly at random the m permutations making up the code. Thus, in the analysis, for a fixed vector weight one can imagine that going into each accumulator round we in fact have a *uniformly random* vector of that weight. To establish that a random RA^m code achieves good distance, the important fact is that a uniformly random vector of *any* weight has expected weight $1/2$ after accumulating. Thus, a random RA^m code typically "pushes the weights towards the middle," which is exactly what we want when we recall that, for a linear code, achieving minimum distance δ is equivalent to having no codewords of weight below weight δ.

2.1.2 Distance analysis

As the analysis of the minimum distance of our codes is inspired by prior analyses of RA^m codes, we now sketch these approaches. As mentioned in the introduction, RA codes (with only a single round of permute and accumulate) were introduced by Divsalar, Jin and McEliece [11]. Beyond the efficiency of the encoding, part of the appeal of these codes is the fairly simple expression of their IOWEF: the expression telling you the expected number

of message vectors of a given weight a that are mapped to a codeword of weight b. This expression is very useful because, as noted in the introduction, to show that a code achieves relative minimum distance δ with probability $1-p$ for some $p \in (0,1)$, by Markov's inequality it suffices to show that the expected number of codewords of relative weight at most δ (which is simply the sum of the IOWEF for absolute codeword weights $b \leq \delta n$). When introducing RA codes, the authors indeed show that we can express the probability that a weight a vector is mapped to weight b after permuting and accumulation as follows (where we rewrite the binomials as this will be useful in the later analysis):

$$p_A(a,b) = \frac{\binom{n-b}{\lfloor a/2 \rfloor}\binom{b-1}{\lceil a/2 \rceil - 1}}{\binom{n}{a}} = \frac{\binom{n-b}{\lfloor a/2 \rfloor}\binom{b}{\lceil a/2 \rceil}}{\binom{n}{a}} \frac{\lceil a/2 \rceil}{b} = \frac{\binom{n-a}{b-\lceil a/2 \rceil}\binom{a}{\lceil a/2 \rceil}}{\binom{n}{b}} \frac{\lceil a/2 \rceil}{b} \ .$$

With $p_A(a,b)$ in hand, one can easily express the expectation that we are interested in.[9] For example, consider a random RA code of rate $1/r$. After repeating a message of absolute weight a, there are $\binom{n/r}{a}$ vectors of absolute weight ra. We then expect $p_A(ra,b)$ of these vectors to be mapped to absolute weight b after applying the first permute-accumulate operation. Now sum over all permitted values of a and b, but restrict $b \leq \delta n$.

Using this, Kahale and Urbanke proved that these single-round codes are not asymptotically good [17]. This led Pfister and Siegel, in a 1999 conference paper, to first study RA^m codes with multiple rounds of permute and accumulate [22]. They numerically estimate the expected number of low weight codewords, which suggests that with at least two rounds, RA^m codes achieve asymptotic goodness, and in fact distance relatively close to the GV-bound. This work was extended into chapter 3 of Pfister's dissertation [23], which contains the first formal proof of the asymptotic goodness of RA^m codes with at least two rounds, as well as precise numerical estimates of the minimum distance attained by RA^m codes of varying rates and number of rounds (similar to our Table 1). Importantly, the provably attained minimum distance is not equal to these numerical estimates, and is in fact far below the GV-bound. Thus, this does not yet prove distance close to the GV-bound.

It takes a few more years before another group of authors formally prove that RA^m codes obtain minimum distance according to these numerical estimates [18]. They achieve this by taking a more analytic approach. Instead of analyzing the IOWEF directly (a complicated expression with many binomial coefficients), they use Stirling's approximation,[10] to say that $p_A(a,b) \leq O(1) \cdot 2^{n \cdot f_A(a/n, b/n)}$, where

$$f_A(\alpha,\beta) = \alpha - h(\beta) + (1-\alpha) h\left(\frac{\beta - \alpha/2}{1-\alpha}\right) \ .$$

This, along with the standard bound $\binom{n/r}{a} \leq 2^{n \cdot h(a/n)/r}$, gives a fairly natural representation for the exponent of the terms appearing in the IOWEF, which is called the *(asymptotic) spectral shape function* (which we formally define below, see Definition 4). This spectral shape function admits an analysis of its critical points, allowing us to maximize over the non-final weights. This reveals a tradeoff between the repetition factor r (which recall determines the rate as $R = 1/r$), and the target minimum distance δ, reminiscent of the tradeoff appearing in the proof of the GV bound (although the expression involved is more complicated). Using this method, they reproduce Pfister's numerical estimates of the minimum distance of RA^m codes, but now proving this distance can be achieved.

[9] Note also that a glance at $p_A(a,b)$ reveals the previous claim that the expected outcome is roughly $b = n/2$.

[10] There are some subtleties regarding floors and ceiling, which can be addressed, as we do in Lemma 5.3 of the full version of this paper [7].

This work, however, only proves that the expectation (and hence probability of attaining this minimum distance quite close to the GV-bound) goes to 1 with the block length n; it does not explicate how fast this happens. Building on this work, Ravazzi and Fagnani first prove that this probability has the form $1 - 1/\text{poly}(n)$ and specify precisely what this polynomial is. Moreover, they prove that the minimum distance converges to the GV-bound as the number of rounds goes to infinity (this was only conjectured before based on numerical estimates) [24]. Many years later, another group of authors characterise the expectation more precisely, showing for the first time that we can expect RA^m codes to have good distance for concrete block lengths n [6].

Let us now formally define the spectral shape function, as we will require it in the analysis of our codes.

▶ **Definition 4.** *Let $r, m \in \mathbb{N}$. The asymptotic spectral shape function of a rate $1/r$ RA^m code is defined recursively as*

$$\hat{r}_{A,r}^{(m)}(\gamma) := \max_{0 \leq \alpha \leq 1} \{\hat{r}_{A,r}^{(m-1)}(\alpha) + f_A(\alpha, \gamma)\} \quad \textit{and} \quad \hat{r}_{A,r}^{(1)}(\gamma) = h(\gamma)/r \ ,$$

and we use it to define

$$\delta_{A,r}^{(m)} := \max\{\delta \in [0, 1/2) : \forall \varepsilon \leq \delta, \hat{r}_A^{(m)}(\varepsilon) = 0\} \ ,$$

where we will use the shorthands $\hat{r}_A^{(m)}(\gamma) := \hat{r}_{A,2}^{(m)}(\gamma)$ and $\delta^{(m)} := \delta_{A,2}^{(m)}$, as we will only consider codes of rate $1/2$ throughout.

The spectral shape function is thus flat at 0 for $\gamma \leq \delta_A^{(m)}(r)$, where it was proven that $\delta_A^{(m)}(r)$ converges to the GV-bound [24]. After this point, the spectral shape function grows and becomes strictly positive. This implies that we expect exponentially many codewords of weight $\gamma > \delta^{(m)}$ in a rate $1/r$ RA^m code, showing that one cannot hope to achieve distance beyond $\delta^{(m)}$ in an RA^m code.[11] When the spectral shape function is equal to 0, for $\gamma \leq \delta^{(m)}$, we do not expect exponentially many vectors. But observe we can't conclude there are 0 such vectors: just subexponentially many. Despite this, previous authors established that we expect few [18], and eventually more precisely only $1/\text{poly}(n)$ [24, 6], such vectors. How did they prove this?

To answer this question, we need to ask why the spectral shape function is flat at 0 for $\gamma \leq \delta_A^{(m)}(r)$ in the first place. This happens because for such small γ, the expression over α (recall that we maximize over α) is in fact decreasing with α. To maximize, we are then forced to fix $\alpha = 0$, which yields the function $\hat{r}_A^{(m-1)}(0) + f_A(0, \gamma) = 0$. Suppose now that one has a non-zero lower bound on α, say $h/n \leq \alpha$. In that case, one would be forced to pick $\alpha = h/n$ to maximize and this would yield a strictly negative value for the spectral shape function. One can use this negative value of the spectral shape function to argue that there will only be about $\exp(-h)$ codewords of weight at most $\delta_A^{(m)}(r)$ in expectation.

Why can we assume we have a lower bound on α? Well, α described the weight of the codeword preceding the final round of permute-accumulate. Hence, if we can argue that only very few (say, some inverse polynomial number) vectors have relative weight below h/n before entering this last round of the encoding, then we can consider that case dealt with, and assume for the rest of the argument that $h/n \leq \alpha$.

[11] At least, not when using Markov's inequality; a stronger inequality may be able to prove better distance, but this seems unlikely to us.

We now have two conflicting forces. We need to pick h large enough so that $\exp(-h)$ is decreasing in n, but also pick h small enough so that we can realistically argue that there are only few vectors of weight at most h after $m-1$ rounds of permute-accumulate. A sensible assignment is $h = \log^2 n$, so that $\exp(-h) = n^{-\log n}$ is negligible in n, but turns out to be small enough to argue that the expected number of vectors of weight at most $\log^2 n$ after $m-1$ rounds is $1/\text{poly}(n)$. We will use this same general proof strategy to analyse our codes: split the computation of the expectation based on whether the weight of the codeword before going into the final round of the encoding is at least $h/n = \log^2 n/n$ or not.

2.1.3 Convergence of distance to GV-bound

Let us end this section by making a note about the convergence of $\delta_A^{(m)}(r)$ to the GV-bound with m, which as we said was proven by [24]. The closer we wish to approximate $\delta^{(\text{GV})}$, the higher m needs to be. Unfortunately, we lack a simple analytical expression for $\delta^{(m)}$, and thus have no theoretical understanding of how fast $\delta_A^{(m)}(r)$ converges to the GV-bound with m. Therefore, if one wishes to get $\varepsilon = 1/10000$-close to the GV-bound, we know this is possible for large enough m, but whether this is already true for $m = 3$ or requires $m = 10$, or even $m = 100$, is not something we can theoretically argue.

What we are left with is numerically estimating $\delta^{(m)}$. To do so, we need to estimate the spectral shape function. Suppose you have a rate $1/r$ RA^m code where we describe the weights in between the rounds using $\alpha_1, \alpha_2, ..., \alpha_{m+1}$, and the non-maximized asymptotic spectral shape function is described by $f_{A^m}(\alpha_1, ..., \alpha_{m+1})$. To determine the value of the spectral shape function, we need to maximize over the α_i's. Pfister first did this by a grid search over the domains of these variables [23]. A more complicated approach was taken by [18]. They gave the constraints

$$\frac{\partial f_{A^m}(\alpha_1, ..., \alpha_{m+1})}{\partial \alpha_1} = 0 \iff \alpha_2 = \frac{1}{2}\left(1 \pm (1-\alpha_1)\sqrt{1 - \left(\frac{\alpha_1}{1-\alpha_1}\right)^{2/r}}\right) ,$$

and

$$\frac{\partial f_{A^m}(\alpha_1, ..., \alpha_{m+1})}{\partial \alpha_i} = 0 \iff \alpha_{i+1} = \frac{1}{2} \pm \frac{1-\alpha_i}{2}\sqrt{1 - \left(\frac{\alpha_i - \frac{\alpha_{i-1}}{2}}{1-\alpha_i-\frac{\alpha_{i-1}}{2}}\frac{1-\alpha_i}{\alpha_i}\right)^2} .$$

Say we hope to achieve some relative minimum distance δ. Then we can use the above expressions to fix $\delta = \alpha_{m+1} = f(\alpha_m, \alpha_{m-1})$, and so on, to eventually obtain an expression of the form $\delta = f(\alpha_1)$, which one can solve for α_1. This yields assignments to $\alpha_1, ..., \alpha_m$ which guarantee the partial derivatives to all these variables are 0. In short, this lets one find critical points of the asymptotic spectral shape function, for some target minimum distance. Find the local maxima, and if these are negative for your choice of δ, r and m, then this means that $\delta \leq \delta^{(m)}$. We used the latter method to fill up Table 1 which contains lower bounds on various $\delta_m^{(m)}$. For example, to get 10^{-4}-close to the GV-bound you need $m \geq 4$.

2.2 Our codes

The codes introduced in this paper are directly inspired by RA^m codes and their analysis as introduced by the previously cited works. As hinted in the introduction, our idea is to "fix" the distance of the dual code, and to do that we introduce rounds where we perform the inverse of the accumulation operation. This inverse operation is the following:

- Let $D \in \mathbb{F}_2^{n\times n}$ denote the *discrete derivative matrix*, i.e., $D_{ij} = 1$ iff $i - 1 = j$ or $i = j$.

To see this is the inverse, note that if $y = Ax$ we have $y_i = \sum_{j=1}^{i} x_j$, so $(Dy)_i = (\sum_{j=1}^{i} x_j) + (\sum_{j=1}^{i-1} x_j) = x_i$. Again, this D matrix can be implemented with $n-1$ XOR gates (and in fact here the depth is just 1!).

With this D operation in hand, we are able to construct pairs of dual codes of rate $1/2$. For instance the RAD and RDA codes are dual to one another when $r = 2$, assuming the accumulation orders are "reversed" (i.e., in one code, A and D are transposed). More precisely, we have the following.

▶ **Proposition 5.** *Let $\pi_1, \ldots, \pi_{2m} : [n] \to [n]$ be permutations. Consider the generator matrices*

$$G = AM_{\pi_{2m}} DM_{\pi_{2m-1}} \cdots AM_{\pi_2} DM_{\pi_1} F_2$$

and

$$H = D^\top M_{\pi_{2m}} A^\top M_{\pi_{2m-1}} \cdots D^\top M_{\pi_2} A^\top M_{\pi_1} F_2 \ .$$

Then $H^\top G = 0$. In particular, they generate dual codes.

Proof. Since for matrices A, B we have $(AB)^\top = B^\top A^\top$ and the transpose of a permutation matrix is its inverse, it follows that

$$H^\top = F_2^\top M_{\pi_1}^\top A M_{\pi_2}^\top D \cdots M_{\pi_{2m-1}}^\top A M_{\pi_{2m}}^\top D = F_2^\top M_{\pi_1}^{-1} D^{-1} M_{\pi_2}^{-1} A^{-1} \cdots M_{\pi_{2m-1}}^{-1} D^{-1} M_{\pi_{2m}}^{-1} A^{-1}.$$

Thus $H^\top G = F_2^\top F_2 = 0$, this last equality being immediate from the definition. ◀

Our goal will be to study the pair of codes $R(AD)^m$ and $R(DA)^m$ for rate $1/2$ over a uniformly random choice of $\pi_1, \ldots, \pi_{2m}$ and prove that both achieve distance converging to the GV-bound as m grows.

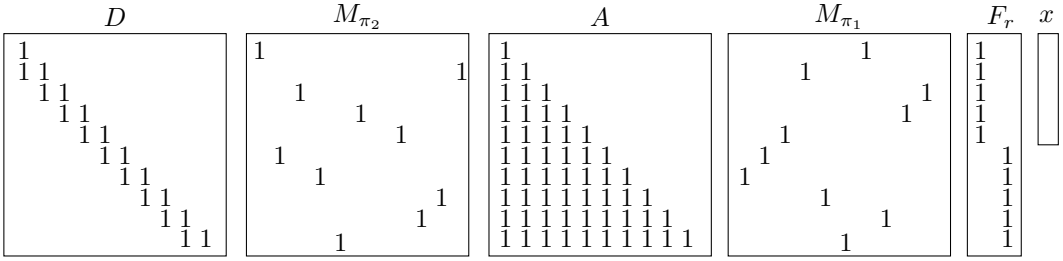

Figure 2 A pictorial representation of (a generator matrix for) an RAD code with rate $R = 1/2$ and block-length $n = 10$.

Since D is the inverse of A, it is not hard to describe $p_D(a, b)$, the probability that a uniformly random vector of absolute weight a is mapped to a vector of absolute weight b under D, in terms of $p_A(a, b)$: we can simply swap a and b in the numerator:

$$p_D(a,b) = \frac{\binom{n-a}{\lfloor b/2 \rfloor}\binom{a-1}{\lceil b/2 \rceil - 1}}{\binom{n}{a}} = \frac{\binom{n-a}{\lfloor b/2 \rfloor}\binom{a}{\lceil b/2 \rceil}}{\binom{n}{a}} \frac{\lceil b/2 \rceil}{a} \ .$$

We can then also define

$$f_D(\alpha, \beta) = (1-\alpha) h\left(\frac{\beta}{2(1-\alpha)}\right) + \alpha h\left(\frac{\beta}{2\alpha}\right) - h(\alpha) \ .$$

such that $p_D(a, b) \leq O(1) \cdot 2^{n \cdot f_D(a/n, b/n)}$. We will often use the shorthands $f_{DA}(\alpha, \beta, \gamma) := f_D(\alpha, \beta) + f_A(\beta, \gamma)$ and $f_{AD}(\alpha, \beta, \gamma) := f_A(\alpha, \beta) + f_D(\beta, \gamma)$. With this, we can define the asymptotic spectral shape function for our codes of interest.

▶ **Definition 6.** *Let $m \in \mathbb{N}$. The* asymptotic spectral shape function *of the rate* $1/2$ $R(AD)^m$ *and* $R(DA)^m$ *codes are defined, respectively, as*

$$\hat{r}^{(m)}_{AD}(\gamma) \leq \max_{\alpha,\beta}\{\hat{r}^{(m-1)}_{AD}(\alpha) + f_{AD}(\alpha,\beta,\gamma)\} \quad \textit{and} \quad \hat{r}^{(1)}_{AD}(\gamma) = h(\gamma)/2 \ ,$$

$$\hat{r}^{(m)}_{DA}(\gamma) \leq \max_{\alpha,\beta}\{\hat{r}^{(m-1)}_{DA}(\alpha) + f_{DA}(\alpha,\beta,\gamma)\} \quad \textit{and} \quad \hat{r}^{(1)}_{DA}(\gamma) = h(\gamma)/2 \ .$$

Lastly, we note that the probability distribution of $\mathrm{wt}(A^\top x)$ for a uniformly random x of absolute weight b is the same as the probability distribution of $\mathrm{wt}(Ax)$, and similarly for D and $D^\top$. This follows from the property that $A^\top$ is obtained by flipping each column of A, which means operationally that instead of accumulating "down," $A^\top$ now accumulates "up" (and similarly for D and $D^\top$). Thus, while technically one of the generator matrices needs to use the transpose of these matrices, the functions appearing in the analysis (p_A, p_D, f_{AD}, etc.) are unchanged. That is, our analysis applies regardless of whether or not the A and D matrices are transposed.

3 Proof strategy

Our goal is to prove Theorem 2: to find a pair of binary linear codes that are encodable with a linear sized circuit (in the block-length of the code), that are dual to each other, and that can attain minimum distance arbitrarily close to the GV-bound. We will do so for the pair of $R(DA)^m$ and $R(AD)^m$ codes of rate $1/2$, as introduced in Section 2.2. By Proposition 5, these can be chosen dual to one another and are encodable with a linear sized circuit: in fact a circuit of size $2m(n-1)$. The task that remains is thus to prove that both of these codes can attain minimum distance arbitrarily close to the GV-bound. We will show this is the case for a randomly sampled such code, except with $1/\mathrm{poly}(n)$ probability. Getting closer to the GV-bound will require larger m, and thus a longer encoding time. In other words, we aim to prove the following theorem, which implies the main result stated as Theorem 2.

▶ **Theorem 7.** *For all $\varepsilon > 0$ and all (even) $n \in \mathbb{N}$ large enough, there exists some $m = m(\varepsilon) \in \mathbb{N}$ such that $R(AD)^m$ and $R(DA)^m$ codes of rate $1/2$ and block-length n achieve relative minimum distance $\delta^{(\mathrm{GV})} - \varepsilon$, except with probability $\geq n^{1/2-m/4} = 1/\mathrm{poly}(n)$. The circuit size of the encoding function for either code is $2m(n-1)$, or $O(mn)$ with depth $O(m\log(n))$.*

The natural question raised by this theorem is what the function $m(\varepsilon)$ looks like. If $m(\varepsilon)$ is too big this could make the codes practically irrelevant (despite linear circuit size asymptotically). The way we will prove the above theorem is to prove instead that the $R(DA)^m$ and $R(AD)^m$ codes can attain relative distance ε-close to $\delta^{(m)}$, where we recall from Section 2.1 that this was the distance achievable by an RA^m code of rate $1/2$. We recall from that same section that $\delta^{(m)}$ converges to $\delta^{(\mathrm{GV})}$ with m [24]. This, in turn, would establish the above result, as it implies that our codes can achieve minimum distance arbitrarily close to $\delta^{(\mathrm{GV})}$.

However, as the reader may recall from that same section, we do not have a theoretical understanding of how m is related to ε. Instead, we have a program which takes m and computes ε. From its behaviour we can see that the convergence is quite fast: for example, to get 10^{-4}-close to the GV-bound you need $m \geq 4$ (see Table 1). Getting a better quantitative understanding of this convergence remains an interesting open question for future work.

Table 1 Lower bounds on the relative minimum distances $\delta^{(m)}$ achievable by RA^m codes of rate 1/2, as well as of the GV-bound (best known rate/distance trade-off for binary codes). We will prove that these relative minimum distances are also achieved by $R(AD)^m$ and $R(DA)^m$ codes of rate 1/2. The values of $\delta^{(m)}$ were computed using the method outlined in Section 2.1.

m	2	3	4	5	∞ ($\delta^{(\mathrm{GV})}$)
$\delta^{(m)}$	0.02859547585	0.1033989603	0.1099391081	0.1100278348	0.1100278644
ε to $\delta^{(\mathrm{GV})}$	$< 10^{-1}$	$< 10^{-2}$	$< 10^{-4}$	$< 10^{-7}$	0

Proof of Theorem 7. As explained above, to establish this theorem, it suffices to prove that rate 1/2 $R(AD)^m$ and $R(DA)^m$ codes achieve relative minimum distance ε-close to $\delta^{(m)}$, except with probability $\tilde{O}(n^{1/2-m/4}) = 1/\mathrm{poly}(n)$. By a union bound over the failure probabilities for both codes, we then obtain the desired result.

Our strategy, which is directly inspired by the method used to prove that RA^m codes achieve this same minimum distance, is as follows. Recall $p_A(a,b)$ and $p_D(a,b)$, which describe the probability that a uniformly random vector of absolute weight a is mapped by A, D, respectively to a vector of absolute weight b. These let us express the expected number of codewords that have weight at most $\delta^{(m)} - \varepsilon$, a quantity which, by Markov's inequality, translates into an upper bound on the probability that a randomly sampled code fails to achieve the given distance $\delta^{(m)} - \varepsilon$.

We will prove this expectation asymptotically behaves like $1/\mathrm{poly}(n)$, proving the theorem. Just like with RA^m codes, we prove this by splitting the computation of the expectation based on the weight of a vector when entering the final round of the encoding (where we consider AD to be one round and DA to be one round). Specifically, we split based on whether the vector has a weight that is within $h := \log^2 n$ of the endpoints 1 and n of the weight range. To work this out formally, let us write $d = (\delta_n - \varepsilon)n$ and $A_w(\mathit{code})$ for the expected number of weight w codewords in the given code, e.g. $A_w(R(AD)^2)$. We can then upper bound the error probability we wish to bound as follows

$$
\begin{aligned}
&\mathbb{P}[R(DA)^m \text{ fails to have absolute distance } d] \\
&\leq \mathbb{E}[\text{number of } R(DA)^m \text{ codewords of absolute weight } \leq d] \\
&= \sum_{w_1=1}^{d} A_{w_1}(R(DA)^m) \\
&= \sum_{w_1=1}^{n}\sum_{w_2=1}^{n}\sum_{w_3=1}^{d} A_{w_1}(R(DA)^{m-1}) p_D(w_1,w_2) p_A(w_2,w_3) \\
&= \underbrace{\sum_{\substack{1\leq w_1\leq h \vee \\ n-h\leq w_1\leq n}}\sum_{w_2=1}^{n}\sum_{w_3=1}^{d} A_{w_1}(R(DA)^{m-1}) p_D(w_1,w_2) p_A(w_2,w_3)}_{(*)} \\
&\qquad + \underbrace{\sum_{w_1=h}^{n-h}\sum_{w_2=1}^{n}\sum_{w_3=1}^{d} A_{w_1}(R(DA)^{m-1}) p_D(w_1,w_2) p_A(w_2,w_3)}_{(**)} .
\end{aligned}
$$

Note that we can write out an analogous expression for the dual code $R(AD)^m$.

In Section 4 of the full version of this paper [7], we will bound the term called $(**)$ as something negligible in n for both codes. In Section 5 of the full version of this paper [7], we will bound the term called $(*)$ as $\tilde{O}(n^{1/2-m/4}) = 1/\mathrm{poly}(n)$ for both codes. Together, we

find $\mathbb{P}[R(DA)^m \text{ fails to have distance } \delta] \leq \tilde{O}(n^{1/2-m/4}) = 1/\text{poly}(n)$, and the same for the dual code. Of course, since we want both codes to attain this distance, we take a final union bound, which doubles this error probability.

Let us give a high-level overview of these two cases, starting with the bound $(**) \leq \text{negl}(n)$. Recall from the discussion in Section 2.1 that one encounters a similar case in the analysis of RA^m codes. There, we analysed the critical points of the spectral shape function, which led to the conclusion that the spectral shape function is flat at 0 for $\gamma \leq \delta^{(m)}$. Recall that the spectral shape function is built up round by round by adding a term $f_A(\alpha, \beta)$ to the previous function $\hat{r}_A^{(m-1)}(\alpha)$ and maximizing over α. What we found for this small initial range of $\gamma \leq \delta^{(m)}$ is that the function over α is decreasing. Hence, to maximize it, we should set $\alpha = 0$, making the spectral shape function equal to 0 for such γ. In this context, however, we have a lower bound $h/n \leq \alpha$ so that we should instead fix $\alpha = h/n$, making the spectral shape function strictly negative (instead of 0), and allowing us to argue that $(**) \leq \exp(-h)$: we only expect a negligible number of the vectors with relative weight at least h/n after $m-1$ rounds of accumulation, to then drop below the target distance of $\delta^{(m)} - \varepsilon$ after the m-th round.

We want to adapt this general strategy to bound $(**)$ for $R(AD)^m$ and $R(DA)^m$ codes. Unfortunately, due to having twice as many operations and weights, directly working out the critical points is infeasible.[12] A numerical approach, using a grid search to estimate maximizers, as Pfister applied to RA^m codes, could work to estimate the minimum distance of these codes. But this would not *prove* the codes attain this minimum distance, nor would it quantify the error probability.

We therefore came up with another approach: directly upper bound the spectral shape function of our codes by that of RA^m codes. If this is true, and if this upper bound remains true when the range over α is restricted, then the upper bound on $(**)$ will follow almost immediately (up to slightly higher polynomial factors from the union bound, which will be irrelevant anyway). This motivates us to consider spectral shape functions where the input weight α is bounded away from 0 and 1. Formally, we consider the following.

▶ **Definition 8** (Restricted Spectral Shape Function)**.** *Fix $\tau \in [0, 1/2)$. The* τ-restricted spectral shape function *for RA^m, $R(AD)^m$ and $R(DA)^m$ codes respectively, are*

$$\hat{r}_A^{(m)}(\tau,\gamma) := \max_{\tau \leq \alpha \leq \min\{1-\tau, 2\gamma, 2(1-\gamma)\}} \{\hat{r}_A^{(m-1)}(\alpha) + f_A(\alpha,\gamma)\} ,$$

$$\hat{r}_{AD}^{(m)}(\tau,\gamma) := \max_{\substack{\tau \leq \alpha \leq 1-\tau \\ \beta}} \{\hat{r}_{AD}^{(m-1)}(\alpha) + f_{AD}(\alpha,\beta,\gamma)\} ,$$

$$\hat{r}_{DA}^{(m)}(\tau,\gamma) := \max_{\substack{\tau \leq \alpha \leq 1-\tau \\ \beta}} \{\hat{r}_{DA}^{(m-1)}(\alpha) + f_{DA}(\alpha,\beta,\gamma)\} .$$

Note that the upper bound on the range of α for RA^m codes follows because $f_A(\alpha, \gamma)$ is only defined for $\alpha \leq \min\{2\gamma, 2(1-\gamma)\}$. This doesn't apply to the other two codes, as there any pair (α, γ) is a valid input to f_{AD} or f_{DA} (though the choice can limit the range over β). This is also the reason why we constrain $h/n \leq \alpha \leq 1 - h/n$ on both sides of the weight range, while in the analysis of RA^m codes it sufficed to just lower bound α. Because of these mismatched ranges we aim to prove an inequality between the restricted spectral

[12] At least, the analytic expressions for the critical points of the spectral shape function of an RA^m code (as found by Kliewer et al [18]) are obtained after making some very clever substitutions. It is unclear to us how this technique could be adapted for $R(AD)^m$ or $R(DA)^m$ codes.

shape functions $\hat{r}^{(m)}_{AD}(\tau,\gamma) \leq \hat{r}^{(m)}_{A}(\tau,\gamma)$ (and likewise for the dual code $R(DA)^m$), instead of directly proving $f_{AD}(\alpha,\beta,\gamma) \leq f_A(\alpha,\gamma)$ (as it is not meaningful to prove such a pointwise inequalities as the domains are different).

More specifically, we prove first that $f_{AD}(\alpha,\beta,\gamma)$ and $f_{DA}(\alpha,\beta,\gamma)$ can both be upper bounded by the same function $g(\alpha,\gamma)$, which is quite similar to $f_A(\alpha,\gamma)$. We then use this upper bound of g to prove that the restricted spectral shape function of both $R(AD)^m$ and $R(DA)^m$ codes are upper bounded by the restricted spectral shape function of RA^m codes. Finally, we show that this means that the existing argument that the analogue of $(**)$ for RA^m is $\mathrm{negl}(n)$ carries over to our codes.

Next, we consider the bound $(*) \leq 1/\mathrm{poly}(n)$. Recall that $(*)$ represents the expected number of low weight codewords which entered the final round of the encoding with a weight very close (within $h = \log^2 n$) to the endpoints of the weight range. We need to do this for both $R(AD)^m$ and $R(AD)^m$ codes. We will in fact simplify by counting *all* codewords with weight near the boundaries of the weight range after $m-1$ rounds (as opposed to only codewords which also have low weight after the entire encoding). Formally, we can express $(*)$ (here for $R(DA)^m$ codes) as:

$$\sum_{\substack{1\leq w_1\leq h \vee \\ n-h\leq w_1\leq n}} \sum_{w_2=1}^{n} \sum_{w_3=1}^{d} A_{w_1}(R(DA)^{m-1}) p_D(w_1,w_2) p_A(w_2,w_3) \leq \sum_{\substack{1\leq w_1\leq h \vee \\ n-h\leq w_1\leq n}} A_{w_1}(R(DA)^{m-1}) \,.$$

To bound the above, we will split it into two cases based on the weight of a vector in earlier rounds. The point is that the vectors with weight $\leq h$ or $\geq n-h$ after $m-1$ rounds can have one of two sources. Either, they had weight $\leq h$ or $\geq n-h$ before all preceding DA or AD rounds. Or, there was some earlier i'th AD or DA round before which the vector had weight in the middle of the weight range, and after which it dropped back down to weight $\leq h$ or $\geq n-h$, where it stayed until after round $m-1$.

In the latter case, we know from our bound on $(**)$ that there are only $\mathrm{negl}(n)$ vectors that can have weight in $[h, n-h]$ and then drop to below $\delta^{(m)}n$ or above $(1-\delta^{(m)})n$. This lets us argue that these cases are indeed negligible, but only from the second round on (as $\delta^{(1)} = 0$). It turns out that we can give a separate argument that there are only $\mathrm{negl}(n)$ vectors dropping from inside to outside weight range $[h, n-h]$ in round 1 as well (though for $R(AD)^m$ codes specifically this only seems to be true after two rounds, which means we can only expect these codes to attain good distance for $m \geq 3$). This then leaves the vectors which are in the weight range $[h.n-h]$ throughout the entire encoding, which is the "hard" case (in terms of contributing a non-negligible term to the expectation). We carefully deal with this case by analyzing the binomial coefficients, and note that these two cases are only $1/\mathrm{poly}(n)$ when $m \geq 3$. ◀

4 Cryptographic application

As mentioned in the introduction, such fast codes with fast good duals can be used to construct efficient secure multiparty computation (MPC) protocols for the problem of *encryption matrix-vector product* (EMVP). In this section, we sketch this application, but refrain from providing precise definitions, preferring to refer to [4] where appropriate.

Briefly, consider a client that wishes to delegate the task of computing matrix-vector products for a matrix M to a server in the cloud. In an initial setup phase, the client takes the matrix M and encrypts it into a matrix $\hat{M}$ (using a secret key), which is then sent to the server. Later, in the online phase, the client takes query vectors q and sends encryptions

$\hat{q}$ thereof to the server, keeping an associated secret key q'. The server then returns a value M'. The correctness requirement is that from M', q' and the initial secret key, the client learns the value of the matrix vector-product Mq. The security requirement is that the server learns nothing about the matrix M or any of the query vectors.

Benhamouda et al [4] suggest resolving this problem in the following way.[13] The first step is for the server to sample two secret, dual codes: one could use an $R(DA)^m$ code $\mathcal{C}$ and its dual $R(AD)^m$ code $\mathcal{C}^\perp$. The code $\mathcal{C}^*$ will be used to encrypt the matrix M: one sets $\hat{M} := MG^* + R$, where G^* is a generator matrix for $\mathcal{C}^*$ and R is a pseudorandom matrix (which serves to hide M). (One technical caveat is that the matrix G^* must be systematic for this application; while we would like to be able to guarantee that this step can be done in linear time since $\mathcal{C}^*$ admits a linear time encoding algorithm, we unfortunately cannot.)

Next, in the online phase, to mask the query q one must use a uniformly random codeword $c \in \mathcal{C}$. To sample such a codeword, one can sample a uniformly random $r \in \mathbb{F}_2^k$ and then output rG, where G generates $\mathcal{C}$. In particular, assuming $\mathcal{C}$ admits linear time encoding this step can be done in linear time!

Thus, we see that by finding self-dual codes with fast encoding we can speed up this algorithm, making our codes quite promising for this application. We remark that Benhamouda et al [4] already suggested using *quasi-cyclic* codes to get the desired dual codes, which admit quasilinear encoding (but not truly linear, as we obtain). In fact, the authors write that "one could potentially use other families of (dual) linear codes that admit fast encoding" [4, Section 3.1].

Of course, this entire discussion is moot if the resulting scheme is insecure. Naturally, the code we sample needs to satisfy a certain hardness assumption: namely, a variant of the *learning subspace with noise (LSN)* assumption [4, Definition 3.1] must hold, which informally states that if one can obtain random codewords from $\mathcal{C}$ which, with probability $\mu \in (0, 1)$, are rerandomized (i.e., one obtains a uniformly random vector a μ-fraction of the time), it is hard to recover the secret code $\mathcal{C}$. One can ask whether a secret $R(AD)^m$ (or $R(DA)^m$) code can be reasonably expected to satisfy this LSN assumption. For the close cousin *learning parity with noise (LPN)* assumption – where essentially the task is to decode – the general consensus is that unless a code displays notable "algebraic structure," the best general strategy is to look for low-weight dual codewords to hopefully recover information about the codeword untainted by the noise (see, e.g., [8, Section 3]). But, note that we have explicitly argued that $\mathcal{C}$ *does not* have any light dual codewords: $\mathcal{C}^\perp$ (quite likely) approaches the GV bound!

Of course, this is merely a sketch of this potential application. Nonetheless, we believe it is emblematic of the principle that such codes with fast encoding and non-trivial dual properties can be useful for cryptography, especially when targetting extremely efficient implementations. We anticipate there to be many further applications in the future.

References

1 Omer Barkol, Yuval Ishai, and Enav Weinreb. On d-multiplicative secret sharing. *Journal of cryptology*, 23(4):580–593, 2010. doi:10.1007/s00145-010-9056-z.

2 Louay Bazzi, Mohammad Mahdian, and Daniel A. Spielman. The minimum distance of turbo-like codes. *IEEE Transactions on Information Theory*, 55(1):6–15, 2009. doi:10.1109/TIT.2008.2008114.

[13] This, admittedly, simplifies certain details.

3 Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 1–10, New York, NY, USA, 1988. Association for Computing Machinery. doi:10.1145/62212.62213.

4 Fabrice Benhamouda, Caicai Chen, Shai Halevi, Yuval Ishai, Hugo Krawczyk, Tamer Mour, Tal Rabin, and Alon Rosen. Encrypted matrix-vector products from secret dual codes. In Chun-Ying Huang, Jyh-Cheng Chen, Shiuh-Pyng Shieh, David Lie, and Véronique Cortier, editors, *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security, CCS 2025, Taipei, Taiwan, October 13-17, 2025*, pages 394–408. ACM, 2025. doi:10.1145/3719027.3765194.

5 Alexander R Block, Zhiyong Fang, Jonathan Katz, Justin Thaler, Hendrik Waldner, and Yupeng Zhang. Field-agnostic snarks from expand-accumulate codes. In *Annual International Cryptology Conference*, pages 276–307. Springer, 2024. doi:10.1007/978-3-031-68403-6_9.

6 Martijn Brehm, Binyi Chen, Ben Fisch, Nicolas Resch, Ron D. Rothblum, and Hadas Zeilberger. Blaze: Fast SNARKs from interleaved RAA codes. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025*, pages 123–152, Cham, 2025. Springer Nature Switzerland. doi:10.1007/978-3-031-91134-7_5.

7 Martijn Brehm and Nicolas Resch. Linear time encodable binary code achieving gv bound with linear time encodable dual achieving gv bound, 2025. doi:10.48550/arXiv.2509.07639.

8 Geoffroy Couteau, Peter Rindal, and Srinivasan Raghuraman. Silver: silent vole and oblivious transfer from hardness of decoding structured ldpc codes. In *Annual International Cryptology Conference*, pages 502–534. Springer, 2021. doi:10.1007/978-3-030-84252-9_17.

9 Ronald Cramer, Ivan Damgård, and Ueli Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 316–334. Springer, 2000. doi:10.1007/3-540-45539-6_22.

10 Irit Dinur, Siqi Liu, and Rachel Yun Zhang. New codes on high dimensional expanders. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference, CCC 2025, Toronto, Canada, August 5-8, 2025*, volume 339 of *LIPIcs*, pages 27:1–27:42. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.CCC.2025.27.

11 Dariush Divsalar, Hui Jin, and Robert J McEliece. Coding theorems for "turbo-like" codes. In *Proceedings of the annual Allerton Conference on Communication control and Computing*, volume 36, pages 201–210. University Of Illinois, 1998.

12 Erez Druk and Yuval Ishai. Linear-time encodable codes meeting the Gilbert-Varshamov bound and their cryptographic applications. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 169–182, 2014. doi:10.1145/2554797.2554815.

13 Venkatesan Guruswami and Piotr Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Transactions on Information Theory*, 51(10):3393–3400, 2005. doi:10.1109/TIT.2005.855587.

14 Venkatesan Guruswami and Widad Machmouchi. Explicit interleavers for a repeat accumulate accumulate (raa) code construction. In *2008 IEEE International Symposium on Information Theory*, pages 1968–1972. IEEE, 2008. doi:10.1109/ISIT.2008.4595333.

15 Venkatesan Guruswami and Jonathan Mosheiff. Punctured low-bias codes behave like random linear codes. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 36–45. IEEE, 2022. doi:10.1109/FOCS54457.2022.00011.

16 Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 433–442, 2008. doi:10.1145/1374376.1374438.

17 N. Kahale and R. Urbanke. On the minimum distance of parallel and serially concatenated codes. In *Proceedings. 1998 IEEE International Symposium on Information Theory (Cat. No.98CH36252)*, page 31, Cambridge, MA, USA, 1998. IEEE. doi:10.1109/ISIT.1998.708611.

18 Jörg Kliewer, Kamil Sh. Zigangirov, Christian Koller, and Daniel J. Costello Jr. Coding theorems for repeat multiple accumulate codes. *CoRR*, abs/0810.3422(arXiv:0810.3422), October 2008. arXiv:0810.3422.

19 Richard E Ladner and Michael J Fischer. Parallel prefix computation. *Journal of the ACM (JACM)*, 27(4):831–838, 1980. doi:10.1145/322217.322232.

20 Jonathan Mosheiff, Nicolas Resch, Noga Ron-Zewi, Shashwat Silas, and Mary Wootters. Ldpc codes achieve list decoding capacity. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 458–469. IEEE, 2020. doi:10.1109/FOCS46700.2020.00050.

21 Jonathan Mosheiff, Nicolas Resch, Kuo Shang, and Chen Yuan. Randomness-efficient constructions of capacity-achieving list-decodable codes. *CoRR*, abs/2402.11533, 2024. doi:10.48550/arXiv.2402.11533.

22 H.D. Pfister and P.H. Siegel. The serial concatenation of rate-1 codes through uniform random interleavers. *IEEE Transactions on Information Theory*, 49(6):1425–1438, 2003. doi:10.1109/TIT.2003.811907.

23 Henry David Pfister. *On the capacity of finite state channels and the analysis of convolutional accumulate-m codes*. PhD thesis, University of California, San Diego, 2003.

24 Chiara Ravazzi and Fabio Fagnani. Spectra and Minimum Distances of Repeat Multiple–Accumulate Codes. *IEEE Transactions on Information Theory*, 55(11):4905–4924, November 2009. doi:10.1109/TIT.2009.2030459.

25 Atri Rudra and Mary Wootters. It'll probably work out: improved list-decoding through random operations. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 287–296, 2015. doi:10.1145/2688073.2688092.

26 Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979. doi:10.1145/359168.359176.

27 Daniel A Spielman. Linear-time encodable and decodable error-correcting codes. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 388–397, 1995. doi:10.1145/225058.225165.

28 Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th annual symposium on foundations of computer science (Sfcs 1986)*, pages 162–167. IEEE, 1986. doi:10.1109/SFCS.1986.25.