

Chosen-Block BAA Codes: Fast, Near the Gilbert–Varshamov Bound, and Field-Agnostic

Stanislav Peceny¹ and Peter Rindal²

¹ Stealth Software Technologies, Inc.

² Category Labs

Abstract. Block–Accumulate–Accumulate (BAA) codes provide fast encoding for pseudorandom correlation generators and code-based zero-knowledge. We replace their independently sampled local maps with repeated copies of one fixed short code. This chosen-block construction retains the two permutation–prefix-sum rounds while allowing the constituent code to be selected for distance and efficient implementation.

Over $\mathbb{F}_2$, exact weight-distribution propagation gives finite-length distance guarantees. Suitable constituents yield numerical asymptotic distance estimates reaching 99.99% of the Gilbert–Varshamov (GV) distance at rate $1/2$. Our practical binary instantiations encode about a million message symbols in 23–24 ms on one CPU thread and under 6 ms on 24–32 threads.

Over larger fields, we add random nonzero coordinate scalings and account for cancellations. For every field of size $q \geq 2^{127}$, including 128-bit prime fields, Reed–Solomon constituents certify relative distances 0.30 and 0.60 at rates $1/2$ and $1/4$, respectively. At message dimension $k = 2^{20}$, each distance bound fails with probability below 2^{-137} under independent uniform sampling of the permutations and nonzero scalings. The rate- $1/4$ asymptotic distance estimate reaches 81% of large-field GV. To investigate the gap between this estimate and GV, we identify permutation patterns that force low-weight codewords regardless of the nonzero coordinate scalings.³

Keywords: linear codes · block–accumulate–accumulate codes · pseudorandom correlation generators · zero-knowledge proofs · Lean

³ This paper made use of GPT 5.6.

Table of Contents

1	Introduction.....	3
2	Preliminaries	10
3	The chosen-block BAA ensemble	12
4	Binary analysis	14
5	Large-field support analysis	17
6	Generic rank and the structural obstruction	20
7	Concrete distance evaluation	23
8	Implementation and performance evaluation.....	28
9	Lean formalization and proof audit	31
A	Explicit binary [32,8,12] subcode.....	34
B	Exact rate-1/2 QR spectrum	35
C	Additional binary analysis	35
D	Additional large-field analysis.....	39
E	Structural-container calculations	44
F	Implementation and benchmark details	46
G	Formalization and proof-audit details.....	48
H	Complete finite-length RS certificate	50

1 Introduction

Fast linear codes are used in two main cryptographic settings. In syndrome-decoding-based pseudorandom correlation generators (PCGs), a noninteractive local step multiplies shares of a sparse noise vector by a structured generator matrix. This step helps expand short seeds into OTs, VOLEs, or multiplication correlations [9,11]. High minimum distance prevents any nontrivial linear test from depending on only a few noise coordinates, as required by the associated syndrome-decoding assumption. At cryptographic vector lengths of roughly a million coordinates, this matrix–vector product is often the main local computational bottleneck [22].

In code-based succinct arguments and their zero-knowledge variants, the prover encodes a witness vector as part of a polynomial commitment, and a proximity test checks that the committed word is close to a valid codeword. The code’s distance supplies the separation needed for soundness, while its encoding cost contributes directly to prover time [3,13,16,18,34]. The same two properties also appear in private server-aided proving, where a repeat–accumulate–accumulate (RAA) code can provide a masking layer for encrypted multi-scalar multiplication: its distance determines the linear-test security parameters, and multiplication by the code contributes to the client’s cost [1].

These applications do not all require codes over the same field. Binary codes directly serve syndrome-decoding-based PCGs. A binary matrix can also be used over a binary extension field by applying it coordinatewise to the binary representation of each field element. Blaze does this with RAA, then uses code switching to reduce a multilinear-evaluation claim to a proximity claim for the resulting packed RAA code [13]. Other PCGs and proof systems instead require encoders that are linear over larger or arbitrary finite fields [1,3,8,16,18,25]. A fast binary encoder therefore does not resolve the nonbinary setting. Moreover, protocol instances often contain 2^{20} or more symbols. Even among $O(n)$ encoders, memory traffic and the constant work per symbol can determine the practical cost.

Block–Accumulate codes were introduced by Kolesnikov, Pecený, Rachuri, Raghuraman, Rindal, and Shah as an implementation-friendly turbo-like family for PCGs and zero knowledge [22]. For a code with dimension k and length n , their main two-accumulator construction, called BA-3, first applies a random systematic block-diagonal map $\mathbf{B} \in \mathbb{F}_2^{k \times n}$, whose small local blocks are sampled independently, and then applies two interleaved prefix-sum accumulators:

$$\mathbf{B} \Pi_1 A \Pi_2 A,$$

where Π_1, Π_2 are random interleavers and A is an accumulator. The composition is shown in Figure 1.

The original BAA work analyzes the binary construction using detailed enumeration, proves an asymptotic linear-distance theorem, gives efficient CPU and GPU implementations, and reports a proof-of-concept integration with Blaze over a binary extension field. It also sketches an extension over $\mathbb{F}_p$ that places

random nonzero diagonal scalings, called weighters, around the accumulators. These scalings randomize the nonzero coordinate values conditional on their support, so the proposed weight analysis does not depend on the original symbol values. The original work sketches this weight-analysis extension but leaves its large-field distance guarantee open [22].

Passing from a probability bound for one message to a useful minimum-distance bound is nontrivial: the number of possible nonzero symbol assignments grows with the field size. Akhiani and Zhang explain why inserting the nonbinary accumulator enumerator into the binary-style union bound can give a vacuous bound, and develop a finer analysis for large-field RAA [2]. Khabbazian further sharpens the RAA failure bounds by accounting jointly for the cancellation constraints in both accumulators [20]. Extending these arguments to chosen-block BAA requires more than changing the outer weight enumerator. Repetition fixes the support within each active block and contributes one independent scalar per block. A chosen block can have several support patterns with different numbers of degrees of freedom. Our refined union bound therefore tracks both support size and shortening dimension, the dimension of the subcode confined to that support, and relates these quantities to cancellations across both accumulator stages.

The original BAA work also observes that the local-distance condition can be enforced by selecting a good subcode and repeating it in every block. This paper develops that suggestion at finite length and asks a second question: over a large finite field $\mathbb{F}_q$, where q is the field size, how close can the same two-accumulator architecture come to the q -ary Gilbert–Varshamov (GV) distance? This question is relevant both to PCGs that generate correlations over finite fields and to proof systems that encode directly over their computation field. To expose the practical tradeoffs without collapsing distance, failure probability, and runtime into a single matching criterion, we also benchmark a representative menu of chosen-block BAA and weighted repeat–accumulate^t configurations at a common message dimension.

1.1 Contributions

Our contributions combine stronger distance guarantees with efficient encoding, and identify what changes when BAA is used over larger fields.

- **Chosen-block BAA over $\mathbb{F}_2$.** We replace independently sampled local maps with repeated copies of one certified short code, preserving its rate and linear-time encoding. Exact propagation of the constituent’s weight distribution

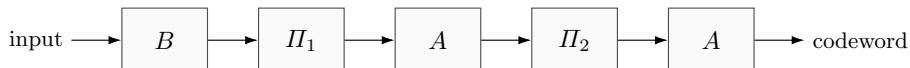


Fig. 1. The BAA encoder. The block-diagonal map **B** is followed by two permutation–accumulator rounds. Here Π_i permutes coordinates and A computes prefix sums.

gives finite-length expected counts of low-weight codewords and explicit sampling-failure bounds. Suitable constituents yield numerical asymptotic distance estimates that nearly meet binary GV at rates $1/2$ and $1/4$. Our practical instantiations also achieve failure probabilities below 2^{-40} at the distance used in the original-style CPU comparison, while encoding faster.

- **Extension to $\mathbb{F}_q$ and a structural obstruction.** We generalize chosen-block BAA to larger fields using Reed–Solomon blocks and random nonzero coordinate scalings. A useful distance bound requires extending the prior large-field RAA techniques [2, 20]: chosen blocks can contribute several degrees of freedom on a local support, rather than the single scalar supplied by each active repetition block. We account for these degrees of freedom jointly with cancellations in both accumulators, obtaining complete finite-length guarantees, including over large prime fields. At about a million message symbols, the rate- $1/2$ and rate- $1/4$ uniform ensembles achieve relative distances 0.30 and 0.60 with sampling-failure probabilities below 2^{-137} and 2^{-209} , respectively. We also prove that some permutation patterns force low-weight codewords for every choice of nonzero scalings. The rate- $1/4$ asymptotic distance estimate reaches about 81% of large-field GV; numerical counts of these bad patterns suggest an architectural source for the remaining gap, although a typical-code distance upper bound remains open.
- **Fast implementations.** We implement and optimize chosen-block BAA over binary and large fields. In our matched 20-bit failure-target comparison, the binary encoders are $1.22\times$ – $1.30\times$ faster than an original-BAA-style CPU control. Our binary and large-field implementations encode about a million message symbols in tens of milliseconds on one CPU thread; the binary implementations run in under 6 ms with multiple threads. We also benchmark weighted repeat–accumulate codes to compare encoding costs across the two families.
- **Formal verification.** In Lean 4, we verify the exact finite-length counting identities, the generic-rank characterization and structural-container count, and the chosen constituent codes. The analytic reductions, asymptotic estimates, and numerical certificates remain outside the formalization.

1.2 Technical overview

BAA combines a cheap block-diagonal outer encoding with two global permute–accumulate stages. The original construction samples the local map in each block independently and analyzes the resulting binary ensemble. We ask whether one carefully chosen short code can instead be reused in every block, and whether the same architecture remains competitive when the ambient field allows cancellations. The binary answer is favorable: chosen constituents are faster to encode in our 20-bit-matched comparison and can approach the binary GV distance. At the measured length, their complete first moments certify 46.38 and 47.09 bits, although we do not present a baseline-matched binary 40-bit benchmark. Over large fields we obtain strong absolute symbol distance and fast single-thread encoders, but uncover an architectural obstruction that leaves a substantial

gap to the corresponding q -ary GV benchmark. We present BAA and weighted repeat-accumulate^t measurements as a configuration menu so that rate, target distance, certification status, and runtime remain visible.

Construction. Let C_0 be a fixed $[m, \sigma, d]_q$ code of length m , dimension σ , and minimum distance d . Chosen-block BAA first encodes n/m independent input blocks with C_0 , rather than sampling a new local map for every block as in the original BAA construction. It then applies two independent permute-accumulate stages; over $\mathbb{F}_q$ with $q > 2$, each stage also multiplies the coordinates by independent random nonzero field elements. In symbols,

$$C_{\text{out}} = C_0^{\oplus n/m}, \quad \mathcal{C}_{\text{BAA}} = \text{Acc } M_2 \text{ Acc } M_1(C_{\text{out}}),$$

where M_i is the i th random permutation and scaling, and Acc maps a word to its running sums. The permutations disperse the local supports, while the accumulators convert the resulting changes into stretches of running-sum state. All maps after the local encoding are invertible, so the resulting code has dimension $k = (n/m)\sigma$ and rate σ/m for every sample. For constant m and σ , encoding remains linear in n .

Fixing C_0 removes the randomness in the local stage and makes the ensemble depend only on the two global stages. It also lets us use a constituent whose distance, complete weight distribution, and implementation are known in advance. In the binary case, the original asymptotic BAA proof already allows a fixed outer map and needs only that each active block contribute weight greater than four. Thus every binary constituent used here retains the high-probability linear-distance guarantee. Our finite-length analysis below determines how large that distance can be at the concrete lengths of interest.

Binary analysis. Write $W_0(z)$ for the weight enumerator of C_0 . Because the outer code is a direct sum, its exact enumerator is $W_0(z)^{n/m}$. Conditional on an input of weight w , a uniform interleaver makes its support uniform among the w -subsets of $[n]$. Over $\mathbb{F}_2$, that support completely determines the accumulator trajectory: its state alternates between zero-runs and one-runs. Counting these runs gives the exact probability of every output weight. Composing this transition for the two accumulators and summing against the outer enumerator gives the exact expected number $\mu_{\text{bin}}(n, L; C_0)$ of nonzero codewords of weight at most L . Markov's inequality then gives the concrete sampling guarantee

$$\Pr[d_{\min}(\mathcal{C}_{\text{BAA}}) \leq L] \leq \mu_{\text{bin}}(n, L; C_0).$$

This calculation produces fixed-length failure bounds for a named constituent, rather than an average over newly sampled local matrices. Optimizing its asymptotic exponent gives transitions of 0.105 and 0.106 for the compact Golay and extended BCH / RM(2, 5) blocks, and 0.110 for the longer $[168, 84, 24]$ quadratic-residue block. The last is within 1.33×10^{-5} of the binary rate-1/2 GV distance.

Large-field analysis. The binary argument does not extend by simply replacing the alphabet. Over a larger field, different nonzero values on the same support can cancel in an accumulator, and all scalar multiples of a codeword describe the same bad direction. We therefore summarize the local code by its shortening dimension: for a support S , let r be the dimension of the outer subcode whose nonzero coordinates are confined to S . That shortening contains $(q^r - 1)/(q - 1)$ projective codeword directions.

For each accumulator we additionally record its zero-return trace, the positions where its running sum returns to zero. A prescribed return imposes a linear condition on a fresh random multiplier. Consequently, the first $r - 1$ such conditions can be absorbed by the projective degrees of freedom, whereas every additional condition costs a factor about $1/q$. Combining this cancellation bound with exact support-placement counts for both accumulators yields a finite-length union bound for every $q > 2$. For Reed–Solomon (RS) constituents, the MDS shortening profile is explicit, so the bound reduces asymptotically to an entropy optimization. At $q = 2^{128}$ it gives estimated transitions 0.305 at rate $1/2$ and 0.604 at rate $1/4$, compared with GV distances 0.492 and 0.744.

For comparison, at rate $1/4$ the large-field RAA analyses prove relative distance 0.50 for repetition-4 codes $[2, 20]$; our chosen-block construction has a complete finite-length guarantee at 0.60 (Theorem 7.1). At rate $1/2$, those growing-field theorems do not cover repetition-2 RAA. Applying our trace analysis gives a numerical estimate of 0.169 for that ensemble, compared with 0.305 for chosen-block BAA (Section 7.6). The RAA guarantee of 0.50 leaves open whether larger distances hold.

Why the large-field gap persists. The trace bound can count several internal cancellation patterns leading to the same final support. To determine whether this multiplicity explains the gap to GV, we instead fix a proposed final support T and propagate its parity checks backward through both accumulators. A codeword supported in T exists exactly when the resulting restricted parity-check matrix loses rank. After replacing the random diagonal entries by indeterminates, we show that its generic rank equals a maximum path packing in a two-layer graph. The parity-check rank of each local MDS block becomes a capacity on paths entering that block.

If the two-step backward neighborhood of T overloads these capacities, the matrix is rank deficient for every specialization of the diagonal entries. Such a *structural container* therefore contains a nonzero codeword no matter how the nonzero scaling coefficients are chosen. We derive an exact expected count of these containers over the two permutations. Its estimated transitions, 0.308 and 0.609, lie only 0.003 and 0.005 above the large-field trace transitions, but well below q -ary GV. A concentration theorem is still needed to turn these structural transitions into typical-code upper bounds. Nevertheless, the obstruction identifies the two-accumulator architecture, rather than the field size or the choice of local MDS code, as the likely source of the observed gap.

Implementation and verification. At $n = 2,097,024$, the chosen Golay and Reed–Muller implementations take 22.810 and 24.278 ms on one CPU thread, compared with 29.596 ms for our same-host original-BAA-style control; their multithreaded latencies are below 6 ms. Over $\mathbb{F}_{2^{128}}$ at $k = 2^{20}$, RS [16, 8, 9] BAA takes 33.633 ms on one thread at rate 1/2, and RS [16, 4, 13] BAA takes 73.739 ms at rate 1/4. The corresponding uniform ensembles have complete finite-length failure bounds $2^{-137.56}$ at distance 0.30 and $2^{-209.60}$ at distance 0.60. We additionally report representative weighted RA^t runtimes, whose failure bounds remain uncertified. Finally, our Lean 4 development checks the finite combinatorial identities, probability bounds, constituent-code facts, and generic-rank results on which the analysis rests.

Organization. Section 2 reviews the mathematical background, and Section 3 defines the ensemble. Sections 4 and 5 give the binary and large-field analyses. Section 6 develops the structural obstruction, and Section 7 gives the concrete distance evaluation and its design implications. Section 8 describes the implementation and reports its performance, while Section 9 describes the machine-checked proof. The appendices provide constituent spectra, detailed calculations, numerical certificates, and reproduction instructions.

1.3 Related work

Coding theory. Turbo codes originated with Berrou, Glavieux, and Thitimajshima [7]. Serial concatenation and the uniform-interleaver method were developed and analyzed by Benedetto, Divsalar, Montorsi, and Pollara [5], while Divsalar, Jin, and McEliece introduced repeat-accumulate codes and their weight-enumerator analysis [15]. Kahale and Urbanke studied minimum distance in random parallel and serial concatenations [19]. Bazzi, Mahdian, and Spielman proved that depth-two constant-memory serial concatenations are asymptotically bad but that repetition followed by two accumulators can be asymptotically good [4]. Kliever, Zigangirov, Koller, and Costello gave coding theorems for repeat-multiple-accumulate codes, including punctured and multi-accumulator ensembles approaching binary GV [21].

Yang introduced weighted nonbinary RA codes for iterative decoding over $\mathbb{F}_q$ [35]. Rosnes and Graell i Amat then derived exact nonbinary accumulator enumerators and gave conditions under which a typical weighted repeat-multiple-accumulate code has linear minimum distance; they checked the required positive distance threshold numerically for the reported field sizes [31]. Abbaszadeh et al. give an arbitrary-field distance calculation for unscaled RAA, but extrapolate its finite-size evaluation to the zk-SNARK instance lengths of interest [1]. Akhiani and Zhang prove constant relative distance up to 1/2 with high probability for RAA codes over sufficiently large finite fields, improving on the corresponding binary guarantee [2]. Khabbaziian obtains sharp minimum-distance lower tails for scaled RAA: when $r \geq 4$, $0 < \delta \leq 1/2$, and $q - 1 \geq \gamma n$, the failure probability is $\Theta_{r,\delta,\gamma}(n^{1-r})$ [20]. In the complementary fixed-field regime, Huang et al. prove that for every fixed $q \geq 3$, a weighted RAA ensemble has some constant relative

distance with failure $o(1/n)$ once the repetition factor exceeds a q -dependent threshold [18]. Our large-field bound follows this line of work: it identifies scalar multiples and charges only the cancellation equations left after the codeword direction’s available degrees of freedom are exhausted. A chosen local block adds a complication: its supports can have different shortening dimensions. Repetition contributes exactly one dimension per active block, so a support meeting t repetition blocks has dimension t . Our implementation comparison uses the randomly scaled RAA ensemble of Khabbazzian [20]. We specialize the same projective support/trace count to its repetition outer code and match the dominant weight-one contribution at the concrete length.

Baweja et al. introduce Encode–Repeat–Accumulate (ERA) codes over arbitrary sufficiently large fields [3]. ERA first encodes with an arbitrary base code, repeats its output, and then applies two permute–multiply–accumulate stages. Their concrete distance theorem reuses one diagonal multiplier in both stages and supports a tailored proximity proof. Chosen-block BAA instead feeds a direct sum of fixed local codes directly into the accumulators and samples the two diagonal layers independently.

The original BAA work [22] samples systematic block-diagonal local matrices, computes detailed expected enumerators, proves BA-3 asymptotically good when each activated block contributes constant distance greater than four, and gives optimized CPU/GPU implementations. It also suggests selecting a good subcode and repeating it across all blocks to enforce the local-distance condition. The binary contribution here is an exact finite-length first moment for each chosen local spectrum, together with certified named blocks. The large-field contribution is an explicit support-and-shortening analysis and a generic-rank obstruction for that specialization. The original paper also notes an $\mathbb{F}_p$ extension using random coefficients and weighters. Our monomial maps—permutations followed by nonzero coordinate scalings—let the analysis group scalar multiples explicitly and expose rank defects that no choice of scaling coefficients can remove.

Cryptographic applications. PCGs. The use of structured LPN and syndrome-decoding assumptions for silent correlation generation grew from vector-OLE compression and foundational PCG work [9, 11]. Subsequent constructions use Ring-LPN [12], structured LDPC codes [14], Expand–Accumulate codes [10], and Expand–Convolute codes [29]. Stationary syndrome decoding keeps the sparse support fixed while resampling its field values across instances. It thereby amortizes the OTs and tree expansions used to generate shared noise [23]. This optimization is complementary to accelerating the subsequent linear map. Li et al. construct programmable PCGs for OLE over every prime field and extend the construction to correlations over arbitrary finite fields using extension-field trace maps under Ring-LPN or quasi-abelian syndrome-decoding assumptions [25]. Their construction does not use BAA, but it makes the nonbinary PCG case explicit rather than treating it as a binary encoding problem. BAA contributes a hardware-friendly encoder to the structured-code lineage [22], and chosen-block BAA preserves its linear-time structure.

Succinct proofs and zero knowledge. Here the code is typically part of a polynomial commitment, an interactive oracle proof, or a proximity test. Its distance enforces separation from invalid words, and its encoder often dominates the prover’s linear work. Ron-Zewi and Rothblum give a theoretical framework with constant prover overhead [30]. Orion uses code switching to obtain a linear-time prover and polylogarithmic proof size [34]. Brakedown builds a field-agnostic SNARK from a practical linear-time code; the paper does not report a zero-knowledge variant [16]. Block et al. generalize Expand–Accumulate codes to arbitrary finite fields with quasi-linear encoding [8]. Blaze lifts a binary RAA matrix to a packed RAA code over a binary extension field and uses it inside a multilinear polynomial commitment [13]; the original BAA work reports a direct encoding and memory improvement in that system [22]. Abbaszadeh et al. use unscaled RAA over large fields to instantiate dual-LPN for encrypted MSM, which privately outsources KZG commitments and proving steps for Nova, Groth16, and Plonk [1]. Their dual-LPN claim remains an assumption; the distance calculation supports parameter selection against linear tests. ERA combines a linear-time field-agnostic encoder with a tailored proximity proof [3]. Huang et al. instead use independently weighted RAA over a fixed nonbinary field in a blind PCS and an end-to-end blind SNARK [18]. Bolt designs a different code for the same code-switching paradigm [17], and PIPFRI combines a linear-time code layer with Reed–Solomon FRI [24]. These systems motivate both halves of our analysis: near-GV binary codes for packed code switching over binary extension fields, and field-valued codes for protocols that avoid a binary lifting or conversion layer.

2 Preliminaries

This section fixes notation and recalls the background on coding theory and formal verification used below. Standard background on the named code families can be found in MacWilliams and Sloane [26].

2.1 Notation

Throughout, q is a prime power, $\mathbb{F}_q$ is the field of order q , $\mathbb{F}_q^* := \mathbb{F}_q \setminus \{0\}$, and $[n] := \{1, \dots, n\}$. Vectors are row vectors unless transposed; for a vector x , $\text{supp}(x) := \{i : x_i \neq 0\}$ and $\text{wt}(x) := |\text{supp}(x)|$. For a finite set S , $x \leftarrow S$ denotes uniform sampling, with separate samples independent unless stated otherwise; $\Pr$ and $\mathbb{E}$ refer to the displayed randomness. Unsubscripted logarithms are natural, and asymptotics are as $n \rightarrow \infty$ with local parameters fixed.

2.2 Error-correcting codes

Coding theory studies how to represent information with controlled redundancy. The redundancy permits recovery from errors in the classical communication setting; in this paper, the same algebraic structure supplies fast linear maps with few low-weight outputs.

A q -ary linear code of length n is a linear subspace $C \subseteq \mathbb{F}_q^n$. If C has dimension k and minimum distance d , it has parameters $[n, k, d]_q$ and rate $R = k/n$. The Hamming distance between x and y is $\text{wt}(x - y)$, so linearity gives

$$d_{\min}(C) = \min_{0 \neq c \in C} \text{wt}(c).$$

Thus proving that a linear code has large minimum distance amounts to ruling out nonzero codewords with small support.

A full-rank matrix $G \in \mathbb{F}_q^{k \times n}$ is a generator matrix when $C = \{uG : u \in \mathbb{F}_q^k\}$. A full-rank $H \in \mathbb{F}_q^{(n-k) \times n}$ is a parity-check matrix when $C = \{x : Hx^\top = 0\}$. The direct sum $C_0^{\oplus b}$ encodes b independent blocks with C_0 . Its dimension and length are multiplied by b , while its generator matrix is block diagonal.

For a binary code, the weight enumerator

$$W_C(z) := \sum_{c \in C} z^{\text{wt}(c)} = \sum_{w=0}^n A_w z^w$$

records the number A_w of codewords of every weight; see MacWilliams and Sloane [26] for this standard framework. The binary analysis in Section 4 uses the complete enumerator of the fixed local block. For $T \subseteq [n]$, the shortening supported on T is

$$C[T] := \{c \in C : \text{supp}(c) \subseteq T\}.$$

Its dimension measures how many codewords can share a prescribed containing support. This is the corresponding local statistic in the large-field analysis of Section 5.

Every $[n, k, d]_q$ code satisfies the Singleton bound $d \leq n - k + 1$. A code meeting it is maximum-distance separable (MDS). Reed–Solomon codes are the principal examples here: for $n \leq q$, an $\text{RS}[n, k, n - k + 1]$ code evaluates all polynomials of degree less than k at n distinct field points. Their optimal shortening profile makes them useful large-field outer blocks.

For asymptotically long codes of rate R , the Gilbert–Varshamov (GV) bound is the reference distance benchmark. Its relative distance δ_{GV} satisfies

$$H_q(\delta_{\text{GV}}) = 1 - R, \quad H_q(\delta) = \delta \log_q(q - 1) - \delta \log_q \delta - (1 - \delta) \log_q(1 - \delta).$$

For $q = 2$, this reduces to the usual binary entropy. The GV value is a benchmark for the numerical transitions reported here, not a claim about any particular finite code.

The binary instantiations use Golay, Reed–Muller, BCH, and extended quadratic-residue codes. Appendix C.1 recalls their standard constructions and spectra; Sections 7 and 8 compare their distance and encoding performance.

2.3 Machine-checked proofs with Lean

Lean 4 is an interactive theorem prover and functional programming language based on dependent type theory [27]. Mathematical definitions, theorem statements, and proofs are represented in a formal language, and a small trusted

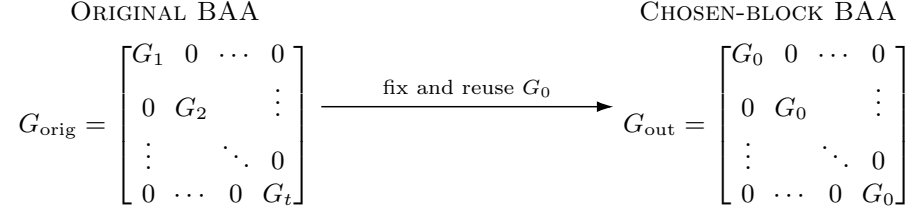


Fig. 2. The block-diagonal stage before and after the chosen-block specialization. Original BAA samples independent systematic generators $G_1, \dots, G_t$. Chosen-block BAA fixes one certified constituent generator G_0 and reuses it in every diagonal position.

kernel checks that each proof term has the type asserted by its theorem. We build on mathlib, Lean’s community-developed library of formal mathematics [33]. Thus “verified in Lean” below means that the claimed finite statement has a kernel-checked proof from the explicitly audited foundations and dependencies; it does not mean that every asymptotic or floating-point calculation in the paper has been formalized.

3 The chosen-block BAA ensemble

The ensemble repeats a fixed local code and randomizes only the two recursive stages. Its binary and large-field specializations use the same linear maps but different methods for counting low-weight outputs.

3.1 Definition and sampling

Following the original BAA notation, let k and n be the global code dimension and length, and let σ be the local input-block size (called the state size in the original paper). Let $C_0 \subseteq \mathbb{F}_q^m$ be a fixed $[m, \sigma, d]_q$ linear code with full-rank generator $G_0 \in \mathbb{F}_q^{\sigma \times m}$. Assume $m \mid n$ and

$$\frac{k}{n} = \frac{\sigma}{m}, \quad \frac{k}{\sigma} = \frac{n}{m}.$$

The second local parameter m is necessary because we allow rates other than $1/2$; for the original rate- $1/2$ setting, $m = 2\sigma$. Define the block-diagonal generator and the outer code it generates by

$$\begin{aligned}
 \mathbf{B} &:= \text{diag}(G_0, \dots, G_0) \in \mathbb{F}_q^{k \times n}, \\
 C_{\text{out}} &:= \{u\mathbf{B} : u \in \mathbb{F}_q^k\} = C_0^{\oplus k/\sigma} \subseteq \mathbb{F}_q^n.
 \end{aligned}$$

Figure 2 isolates the change to the local block-diagonal stage: independently sampled blocks are replaced by repeated copies of one fixed constituent generator.

Sample independent permutations $\Pi_1, \Pi_2 \leftarrow \text{Sym}([n])$. For $q > 2$, let

$$D_i = \text{diag}(\alpha_{i,1}, \dots, \alpha_{i,n}), \quad \alpha_{i,j} \leftarrow \mathbb{F}_q^*,$$

independently, and set

$$M_i := \Pi_i D_i.$$

For the binary specialization, set $D_i = I$, so $M_i = \Pi_i$. Permuting before or after diagonal scaling induces the same nonbinary ensemble. The formulas below use the displayed order.

Define the prefix-sum accumulator

$$(\text{Acc } x)_j := \sum_{\ell \leq j} x_\ell.$$

The BAA code is

$$\boxed{\mathcal{C}_{\text{BAA}} := \text{Acc } M_2 \text{Acc } M_1 (C_{\text{out}}).}$$

With row-vector convention, a generator may be written

$$G_{\text{BAA}} = \mathbf{B} M_1 A M_2 A \in \mathbb{F}_q^{k \times n},$$

where A is the lower-triangular all-ones matrix.

Proposition 3.1 (Dimension preservation). *For every choice of permutations and nonzero diagonal entries,*

$$\dim \mathcal{C}_{\text{BAA}} = k$$

deterministically.

Proof. The accumulator is lower triangular with unit diagonal, and each monomial map is invertible. Their composition is a linear equivalence of $\mathbb{F}_q^n$. Restricting that equivalence to C_{out} preserves dimension, while the direct sum has dimension $(k/\sigma)\sigma = k$.

Encoding cost. For constant σ and m , local block encoding costs $O(n)$ field operations. Each permutation, diagonal scaling, and accumulator also costs $O(n)$. Hence the complete encoder uses

$$\boxed{O(n).}$$

3.2 Why two accumulators are necessary

The obstruction is classical in the binary setting: Bazzi, Mahdian, and Spielman prove that depth-two serial concatenations with constant-memory encoders have sublinear minimum distance for every interleaver [4, Theorem 3]. We give a short probabilistic proof for our chosen-block ensemble over arbitrary fields, including nonzero coordinate scalings.

For comparison, define the one-accumulator code

$$\mathcal{C}_{\text{BA}} := \text{Acc } M_1 (C_{\text{out}}).$$

Proposition 3.2 (Failure of one accumulation). *Fix a nonzero word in C_0 and a constant $0 < \delta < 1$. Over the random monomial map M_1 , and along multiples of m , the one-accumulator ensemble satisfies*

$$\Pr[d_{\min}(\mathcal{C}_{\text{BA}}) \leq \delta n] \rightarrow 1.$$

Proof. Fix any nonzero local codeword $c_0 \in C_0$ of weight w . Place one copy in each of the $n/m = k/\sigma$ local blocks.

Let

$$H = \{n - L + 1, \dots, n\}, \quad L = \lfloor \delta n \rfloor.$$

For block i , let E_i be the event that all w support coordinates of its copy of c_0 land in H under Π_1 . Then

$$\Pr[E_i] = \frac{\binom{L}{w}}{\binom{n}{w}} = \delta^w + O(n^{-1}).$$

On E_i , the accumulator input is zero outside H , so its output is also supported in H . Thus E_i produces a codeword of weight at most L .

Let $X := \sum_i \mathbf{1}_{E_i}$. For distinct blocks $i \neq j$, their chosen supports are disjoint and

$$\Pr[E_i \cap E_j] = \frac{\binom{L}{2w}}{\binom{n}{2w}} = \Pr[E_i] \Pr[E_j] + O(n^{-1}).$$

Because $n/m = \Theta(n)$ and $\Pr[E_i] = \Theta(1)$, $\mathbb{E}X = \Theta(n)$ and $\text{Var}(X) = O(n)$. Chebyshev's inequality therefore gives $\Pr[X = 0] = O(n^{-1})$. Hence

$$\Pr[d_{\min}(\mathcal{C}_{\text{BA}}) \leq \delta n] \rightarrow 1.$$

Thus one accumulator cannot give asymptotically linear distance with a constant-size direct-sum outer code. The second randomized accumulator is an essential part of the construction.

4 Binary analysis

Set $q = 2$. Then the diagonal randomizers are trivial and the construction is

$$\boxed{C_0^{\oplus k/\sigma} \xrightarrow{\Pi_1} \text{Acc} \xrightarrow{\Pi_2} \text{Acc} .}$$

The local block $C_0 \subseteq \mathbb{F}_2^m$ is deterministic. Its complete weight spectrum is the only local-code input to the binary analysis.

4.1 Finite-length first moment

We use the classical uniform-interleaver method of Benedetto and Montorsi [6], developed for serial concatenations by Benedetto, Divsalar, Montorsi, and Polara [5]. The chosen-block input to this calculation is the exact spectrum of a fixed direct-sum outer code.

Write the local homogeneous weight enumerator as

$$W_0(z) = \sum_{w=0}^m A_w^{(0)} z^w, \quad A_0^{(0)} = 1.$$

For the direct-sum outer code,

$$W_{\text{out}}(z) = W_0(z)^{k/\sigma}, \quad A_w^{\text{out}} = [z^w] W_0(z)^{k/\sigma}. \quad (1)$$

For constant m , enumerating the 2^σ local messages gives the complete spectrum. A BCH-like block therefore needs neither an asymptotic spectrum theorem nor a random local-code failure event.

The following accumulator enumerator is due to Divsalar, Jin, and McEliece [15]. We recall its run-counting proof in Appendix C.2.

Lemma 4.1 (Binary accumulator transition). *For $1 \leq w \leq n$, put*

$$r_w := \left\lceil \frac{w}{2} \right\rceil, \quad s_w := \left\lfloor \frac{w}{2} \right\rfloor.$$

For $1 \leq h \leq n$, the exact binary accumulator input–output enumerator is

$$E_{w,h}^{\text{bin}} = \binom{h-1}{r_w-1} \binom{n-h}{s_w}. \quad (2)$$

If $w > 0$ and $h = 0$, then $E_{w,0}^{\text{bin}} = 0$ because the accumulator is invertible. This separate boundary case avoids interpreting $\binom{h-1}{r_w-1}$ at $h = 0$.

The run-counting proof is recalled in Appendix C.2.

Let $\mathcal{H}_{n,w} := \{u \in \mathbb{F}_2^n : \text{wt}(u) = w\}$. Consequently, for $U \leftarrow \mathcal{H}_{n,w}$,

$$P_n^{\text{bin}}(w, h) := \Pr_{U \leftarrow \mathcal{H}_{n,w}} [\text{wt}(\text{Acc } U) = h] = \frac{E_{w,h}^{\text{bin}}}{\binom{n}{w}}. \quad (3)$$

In contrast with the large-field trace formula below, the binary accumulator state alternates deterministically between zero and one at the active coordinates; there is no independently chosen zero-return trace.

Define the second-stage low-weight tail

$$Q_n^{\text{bin}}(h, L) := \sum_{j=1}^L P_n^{\text{bin}}(h, j).$$

Composing these transitions and bounding the probability of a nonzero low-weight count is the standard RAA/RMA first-moment argument [4, 21]. Here we substitute the chosen-block spectrum A_w^{out} for the repetition-code spectrum.

Theorem 4.2 (Exact binary first moment). *Let $1 \leq L \leq n$. Over independent $\Pi_1, \Pi_2 \leftarrow \text{Sym}([n])$, the expected number of nonzero BAA codewords of weight at most L is*

$$\mu_{\text{bin}}(n, L; C_0) = \sum_{w=1}^n A_w^{\text{out}} \sum_{h=1}^n P_n^{\text{bin}}(w, h) Q_n^{\text{bin}}(h, L). \quad (4)$$

Both interleavers are uniform, so conditioned on its weight, the word entering each accumulator is uniform on the corresponding Hamming slice. Since the permutations and accumulators are invertible,

$$\Pr[d_{\min}(\mathcal{C}_{\text{BAA}}) \leq L] \leq \mu_{\text{bin}}(n, L; C_0). \quad (5)$$

Unlike the large-field case, no projective-support correction is needed: $\mathbb{F}_2^*$ contains only one scalar.

Proof. Fix an outer word of weight w . A uniform permutation sends it uniformly to the weight- w Hamming slice, so (3) gives the first output weight. Conditional on an intermediate weight h , the independent second permutation is uniform on the weight- h slice, and $Q_n^{\text{bin}}(h, L)$ gives the final tail. Summing over the A_w^{out} outer words proves (4) by linearity of expectation. If the minimum distance is at most L , at least one nonzero bad word exists. Markov's inequality therefore gives (5).

Corollary 4.3 (Asymptotic linear minimum distance). *Let $C_0 \subseteq \mathbb{F}_2^m$ be a fixed $[m, \sigma, d]$ code with constant m and σ and with $d > 4$. Along admissible lengths n with $k/n = R := \sigma/m$, there is a constant $\varepsilon = \varepsilon(d, \sigma, R) > 0$ such that*

$$\Pr[d_{\min}(\mathcal{C}_{\text{BAA}}) \geq \varepsilon n] \xrightarrow{n \rightarrow \infty} 1.$$

Proof. The linear-distance proof in the original BAA work [22] fixes the outer block-diagonal map and assumes that every activated input block contributes output weight at least a constant $d' > 4$; it does not require the local blocks to be sampled independently. Here the full-rank local generator maps every nonzero input block to a nonzero word of C_0 , hence to one of weight at least d . The original theorem therefore applies with $d' = d$ to the same two independent uniform interleavers and accumulators.

4.2 Asymptotic consequences

The local spectrum controls both sparse failures and the linear-weight transition. A single active block of local distance d contributes at most $O(n^{1-\lceil d/2 \rceil})$ at a fixed target relative distance. For linear weights, we combine the direct-sum coefficient exponent with the two accumulator transition exponents. Appendix C.3 gives these calculations and the restricted negative-exponent criterion. The numerical transitions in Section 7 use this criterion; finite-length guarantees use the complete sum in (4).

5 Large-field support analysis

Over a large field, many codewords can have the same support, and every nonzero scalar multiple of a codeword is present automatically. Counting these scalar multiples as separate bad events can greatly inflate a first-moment bound. We instead treat scalar multiples as one codeword direction, fix its support, and compare the accumulator cancellation constraints with the scalar degrees of freedom available on that support.

5.1 Accumulator traces and projective cancellation

The use of random nonzero diagonal scalings, or weighters, around nonbinary accumulators follows the weighted nonbinary RA/RMA literature [31, 35]. Rosnes and Graell i Amat show that these weighters allow ensemble-average enumerators to be composed using Hamming weights alone [31, Theorem 1]. Recent nonbinary RAA analyses likewise distinguish between counting all field-valued codewords and bounding whether a sampled code contains a bad codeword direction, meaning a nonzero word modulo scalar multiplication [2, 18, 20]. Each active repetition block contributes one scalar degree of freedom. A support meeting t such blocks therefore has shortening dimension t , and can contain many codeword directions when $t > 1$. For a chosen local code, each block's contribution depends on its local support. The RAA argument thus requires a new shortening analysis to give our refined chosen-block bound.

We use the cancellation-trace and random-gap decomposition of Khabbazian [20, Section 3]. Consider an accumulator input with exactly s nonzero coordinates at

$$p_1 < \cdots < p_s, \quad p_{s+1} := n + 1.$$

Let

$$z_i := \sum_{\ell=1}^i u_{p_\ell}$$

be the accumulator state immediately after the i th active coordinate. Define the zero-return trace

$$Z := \{i : z_i = 0\}, \quad a := |Z|.$$

Because every active input symbol is nonzero, $1 \notin Z$ and Z cannot contain consecutive indices.

The placement count below separates support geometry from symbol choices. It is related to the nonbinary run enumeration of Rosnes and Graell i Amat [31, Theorem 2], whose formula counts symbol assignments and requires termination at zero. Our prefix-sum accumulator has no terminal-state constraint, so we count its placements directly.

Lemma 5.1 (Nonbinary accumulator placements). *Suppose the active input coordinates form a uniform s -subset of $[n]$. The support-placement probability for*

output weight h , summed over all zero-return traces of cardinality a , is

$$\mathcal{G}_n(s, h, a) := \frac{\binom{s-a}{a} \binom{n-h}{a} \binom{h-1}{s-a-1}}{\binom{n}{s}}. \quad (6)$$

for feasible s, h, a ; the value is zero otherwise.

Proof. The three factors count, respectively:

- the nonadjacent zero-return trace;
- the zero-state gaps;
- the positive nonzero-state gaps.

Dividing their product by the $\binom{n}{s}$ possible input supports gives (6). The gap construction is reversible, so every placement is counted once.

Fix an exact outer support $S \subseteq [n]$. Write

$$s := |S|, \quad C_{\text{out}}[S] := \{c \in C_{\text{out}} : \text{supp}(c) \subseteq S\}, \quad r := \dim C_{\text{out}}[S].$$

The shortening contains exactly

$$P_r = \frac{q^r - 1}{q - 1}$$

projective lines, of which at most P_r can have exact support S .

Following the large-field RAA analyses [2, 20], we charge cancellations against the available scalar degrees of freedom across both stages. Here that budget is determined by the shortening dimension r . For one fixed projective word, a prescribed zero return in one accumulator forces one fresh diagonal multiplier to one value in $\mathbb{F}_q^*$. Hence a prescribed total of c zero returns across the two stages succeeds with probability at most $(q - 1)^{-c}$ for that fixed projective word.

Lemma 5.2 (Projective cancellation bound). *Let $q > 2$, let an outer shortening have dimension $r \geq 1$, and prescribe c zero returns across the two accumulators. Over the independent nonzero diagonal entries, the probability that some projective word in the shortening realizes all c returns is at most*

$$\chi_q(r, c) := \min \left\{ 1, \frac{q^r - 1}{q - 1} (q - 1)^{-c} \right\}. \quad (7)$$

Proof. Normalize each nonzero word by its first nonzero coordinate. An r -dimensional shortening has $(q^r - 1)/(q - 1)$ normalized representatives. For a fixed representative, each prescribed return fixes a fresh independent diagonal entry and therefore costs a factor $(q - 1)^{-1}$. The union bound over representatives gives the second argument of the minimum in (7); probability is at most one.

For large q ,

$$\chi_q(r, c) \approx (q - 1)^{-(c - (r - 1))_+}.$$

Thus the first $r - 1$ cancellation constraints can be paid for by projective degrees of freedom; each surplus constraint costs approximately $1/q$.

5.2 Finite-length master bound

Theorem 5.3 (Finite large-field master bound). *Let $q > 2$ and $1 \leq L \leq n$. Over independent uniform permutations and independent uniform nonzero diagonal entries, the chosen-block BAA ensemble satisfies*

$$\Pr[d_{\min}(\mathcal{C}_{\text{BAA}}) \leq L] \leq \sum_{\emptyset \neq S \subseteq [n]} \sum_{a=0}^{|S|-1} \sum_{h=1}^n \sum_{b=0}^{h-1} \sum_{j=1}^L \mathcal{G}_n(|S|, h, a) \cdot \mathcal{G}_n(h, j, b) \cdot \chi_q(r_S, a + b), \quad (8)$$

where $r_S := \dim C_{\text{out}}[S]$. We set $\chi_q(0, c) = 0$. The outer sum may be restricted to exact supports of nonzero outer codewords, since every other support event is empty.

Proof. Partition a nonzero outer word by its exact support S , and partition each accumulator trajectory by (a, h, b, j) . Lemma 5.1 supplies the two independent support-placement factors. For each fixed support, Lemma 5.2 bounds the union over its normalized projective words by $\chi_q(r_S, a + b)$. A final union bound over the displayed finite index set proves (8).

Although S indexes an exact-support event, r_S is the dimension of the larger shortening whose words have support contained in S . It therefore upper-bounds the number of projective candidates with exact support S . Equation (8) groups each shortened projective space before applying the union bound, so every codeword retains its unique exact support.

5.3 Shortenings and asymptotic consequences

For a local coordinate set $T \subseteq [m]$, define

$$C_0[T] := \{c \in C_0 : \text{supp}(c) \subseteq T\}, \quad r_0(T) := \dim C_0[T].$$

Call T admissible when $C_0[T]$ is nontrivial.

Proposition 5.4 (Shortening profile). *Suppose that C_0 has minimum distance d . Every admissible set T of size s_T satisfies*

$$s_T - r_0(T) \geq d - 1.$$

Consequently, let a global support have size s , shortening dimension r , and nonempty intersection with t blocks. If each of those t local intersections is admissible, then

$$s - r \geq t(d - 1),$$

and its support excess $E := s - r + 1$ satisfies

$$E \geq t(d - 1) + 1. \quad (9)$$

If C_0 is an $[m, \sigma, d]$ MDS code, then every $T \subseteq [m]$ satisfies

$$r_0(T) = (|T| - d + 1)_+. \quad (10)$$

Proof. The shortening $C_0[T]$ has length s_T , dimension $r_0(T)$, and minimum distance at least d . The classical Singleton bound gives the local inequality [26]. Shortening of a direct sum decomposes blockwise, so dimensions and support sizes add; summing over the t admissible intersections proves (9).

For an MDS code, restriction to any σ coordinates is an isomorphism. The kernel of restriction to the complement of T therefore has dimension $\max\{0, \sigma - (m - |T|)\} = (|T| - d + 1)_+$, which proves (10).

The admissibility condition in (9) is necessary: an arbitrary small set can support only the zero shortening and need not satisfy the local Singleton inequality. Exact supports of outer codewords are admissible in every block they meet.

This pointwise optimal shortening profile makes a small Reed–Solomon block a natural local code over a large field.

Consequences for distance. The shortening profile supplies the local input to both the sparse-support and linear-weight analyses. In the no-surplus regime, a support meeting t blocks has at least $t(d - 1) + 1$ surviving nonzero-state gaps across the two stages. For linear supports, we aggregate blocks by support size and shortening dimension and optimize the resulting entropy bound. Appendix D.1 gives the sparse estimates and the full optimization. The complete finite-length RS guarantees in Theorem 7.1 also control the intermediate support sizes and surplus cancellations.

6 Generic rank and the structural obstruction

The master bound in Section 5 distinguishes zero-return traces even when they produce the same final support. Its expected trace count therefore cannot, by itself, show that sampled codes above the threshold typically contain a low-weight word. We instead fix a proposed final support and ask directly whether the code contains a nonzero word confined to it. A parity-check matrix propagated backward through the two accumulators groups all internal traces and scalar assignments that lead to that support. Treating the diagonal coefficients as algebraically independent variables defines its generic rank; we characterize that rank by a path-packing problem and derive an obstruction that holds for every coefficient choice.

6.1 Backward parity-check formulation

Switch temporarily to column-vector convention. Let H be a parity-check matrix for C_{out} and let

$$\Delta := A^{-1}$$

be the first-difference operator. After renaming the inverses of the two random monomial maps, a parity-check matrix for BAA is

$$\boxed{Q = H D_1 P_1 \Delta D_2 P_2 \Delta.} \tag{11}$$

For a final coordinate set $T \subseteq [n]$, a nonzero BAA codeword supported in T exists exactly when

$$\boxed{\text{rank } Q_T < |T|}. \quad (12)$$

This event groups every outer support, projective assignment, and pair of accumulator traces that leads to the same final support.

Replace the entries of D_1 and D_2 by two disjoint families of indeterminates, and let K be the fraction field of the resulting polynomial ring. All ranks in the remainder of this subsection are over K . For an $[m, \sigma]$ MDS block, define the local capacity

$$h := m - \sigma$$

for the number of paths that may terminate in that block.

The layered graph associated with (11) has three copies V_2, V_1, V_0 of $[n]$. Add an edge from $j \in V_2$ to $i \in V_1$ when $(P_2 \Delta)_{i,j} \neq 0$, and add an edge from $j \in V_1$ to $i \in V_0$ when $(P_1 \Delta)_{i,j} \neq 0$. Join each vertex of V_0 to the local block that contains its coordinate. Coordinate vertices have unit capacity, and every block terminal has capacity h . For $T \subseteq V_2$, let $\nu(T)$ be the maximum number of capacity-respecting paths from distinct vertices of T to the block terminals.

Theorem 6.1 (Generic rank as path packing). *Let C_0 be an $[m, \sigma]$ MDS code, and let H be a block-diagonal parity-check matrix for C_{out} . For every final support $T \subseteq [n]$,*

$$\text{rank}_K Q_T = \nu(T).$$

Equivalently, the generic rank equals the minimum cut capacity in the layered graph above.

Proof. Apply rectangular Cauchy–Binet at both diagonal layers. Each pair of selected coordinate sets contributes a distinct diagonal monomial, so nonzero terms cannot cancel across selections. A minor of Δ is nonzero exactly when its selected row and column sets admit the unique vertex-disjoint linkage in which each step stays in place or moves to its successor. Finally, an MDS parity-check matrix has selected-column rank $\min\{s, h\}$ on a block with s selected columns. A full-rank minor therefore exists exactly when the layered network admits a capacity-respecting path packing of the same size.

This theorem is a trace-free analogue of Khabbazzian’s shared-rank method for large-field RAA [20].

6.2 A coefficient-independent obstruction

For $S \subseteq [n]$, define

$$S + 1 := \{i + 1 : i \in S, i < n\}$$

and the one-step path neighborhood

$$\partial S := S \cup (S + 1),$$

so the right endpoint is handled explicitly. Define the two-step backward neighborhood

$$U(T) := P_1 \partial(P_2 \partial T). \quad (13)$$

For an outer-coordinate set U , define its MDS block rank

$$\kappa(U) := \sum_{\text{blocks } J} \min\{|U \cap J|, h\}. \quad (14)$$

Every backward path from T terminates in $U(T)$, hence

$$\text{rank } Q_T \leq \kappa(U(T)).$$

Consequently, the rank criterion gives the following corollary.

Proposition 6.2 (Structural obstruction). *Let C_0 be an $[m, \sigma]$ MDS code and put $h = m - \sigma$. For every $T \subseteq [n]$,*

$$\boxed{\kappa(U(T)) < |T| \implies T \text{ supports a nonzero BAA codeword}} \quad (15)$$

for every choice of diagonal coefficients. The obstruction is purely a defect of the two permutations, the two difference layers, and the local block-rank caps. Increasing q cannot remove it.

Proof. The selected columns of the transported parity-check matrix lie in the direct sum of the blockwise column spaces indexed by $U(T)$. That space has dimension at most $\kappa(U(T))$. If this value is smaller than $|T|$, then Q_T has a nonzero kernel vector for every specialization of the diagonal entries. By (12), the resulting codeword is supported in T .

The mechanism is block saturation. Once $U(T)$ occupies at least h positions inside a block, additional positions in that block add no parity-check rank. Two backward neighborhood expansions can therefore contain many coordinates but concentrate them in a smaller number of almost-full blocks.

We call a set T satisfying (15) a direct structural container: “container” because the codeword’s exact support may be a strict subset of T , and “direct” because the block-capacity test (15) detects it without solving the full path-packing problem.

6.3 Counting structural containers

Let Z_t count the t -subsets satisfying the structural obstruction (15). We obtain an exact expression for $\mathbb{E}Z_t$ by tracking the runs in each backward neighborhood and the rank contributed by each local block; Appendix E.1 gives the count and its proof. This calculation groups all traces and coefficient assignments leading to the same final support. Its numerical transition helps assess the gap in the trace bound, but a growing expectation alone does not establish a typical-code distance upper bound: concentration remains open.

7 Concrete distance evaluation

The calculations below evaluate the binary first moment, the large-field trace bound, and the structural-container count. All transition points come from floating-point optimization; none is an interval-certified theorem. Displayed values are rounded to at most three decimal places unless a smaller gap is shown in scientific notation; the optimizations retain full precision internally.

From an enumerator to a concrete distance guarantee. For the binary construction at a fixed length n , we choose an acceptable code-sampling failure probability ε and evaluate the cumulative first moment $\mu_{\text{bin}}(n, L; C_0)$ in (4). The largest L for which this quantity is at most ε is a concrete ensemble guarantee: by (5), a fresh choice of the two interleavers has minimum distance greater than L except with probability at most ε . The local weight enumerator is therefore an input to the calculation, not itself the claimed global minimum distance.

The original BAA calculation also propagated an expected weight enumerator through the accumulators, but it first averaged over a fresh random local matrix in every block. Here the named constituent C_0 is deterministic and repeated in every block, so its exact spectrum W_0 enters (1) directly and the binary expectation is only over the two interleavers. There is no additional event that a sampled local code is weak. The asymptotic values δ^* reported below are the zero crossings of the corresponding first-moment exponents; they should not be confused with a finite- n certificate unless (4) is evaluated at the stated n and failure target. Over large fields, the analogous calculation uses shortening dimensions and zero-return traces rather than a binary weight enumerator, while the structural-container transition is a separate diagnostic.

7.1 Binary instantiations

The binary Gilbert–Varshamov benchmarks solve

$$H_2(\delta_{\text{GV}}) = 1 - R.$$

Numerically,

$$\delta_{\text{GV}}(R = 1/2) \approx 0.110, \quad \delta_{\text{GV}}(R = 1/4) \approx 0.215.$$

The short-code parameters and enumerators are standard or taken from the cited exact weight-distribution calculation; see MacWilliams and Sloane [26] and Su et al. [32]. For each local spectrum, the calculation maximizes (23) over its feasible domain and finds the value of δ where the nontrivial interior maximum crosses zero. Equations (18)–(20) handle sparse weights separately.

The compact rate-1/2 constituents give the following results at $n = 2,097,024$ and $L = \lfloor 0.10n \rfloor$:

C_0	δ^*	$\mu_{\text{bin}}(n, L; C_0)$
Golay [24, 12, 8]	0.105	$\leq 2^{-46.386}$
RM(2, 5) [32, 16, 8]	0.106	$\leq 2^{-47.093}$

Both therefore exceed a 40-bit sampling-failure target at relative distance 0.10. Appendix C.4 gives the exact spectra and complete calculation, including its one-bit numerical rounding allowance. The asymptotic transitions are estimates; the failure bounds evaluate the finite-length first moment.

Rate 1/2: extended quadratic-residue spectra. To improve the central rate-1/2 distance result, we next evaluate longer extended binary quadratic-residue blocks. Table 1 reports the transition obtained from each exact spectrum.

Local code	δ^*	Fraction of binary GV	One-block scale
QR [48, 24, 12]	0.109	98.7%	n^{-5}
QR [80, 40, 16]	0.110	99.7%	n^{-7}
QR [104, 52, 20]	0.110	99.9%	n^{-9}
QR [168, 84, 24]	0.110	99.99%	n^{-11}

Table 1. Rate-1/2 binary BAA transitions from exact extended quadratic-residue spectra. A block labeled $[m, \sigma, d]$ has length m , dimension σ , and minimum Hamming distance d . The GV value is 0.110. The transition points are floating-point estimates, not interval-certified bounds.

For the strongest row, the numerical transition is

$$\delta_{168,84,24}^* \approx 0.110.$$

It lies about 1.33×10^{-5} below binary GV. The local distance 24 makes the one-active-block contribution $O(n^{-11})$. These longer blocks strengthen the distance calculation but are not included in the CPU benchmarks or the Lean certificate.

At rate 1/4, the explicit BCH subcode [32, 8, 12] gives an estimated transition 0.213, about 99.4% of binary GV, and a one-block contribution of order n^{-5} . Appendix A gives its generator and exact spectrum. All these constructions retain linear encoding cost when the local block size is fixed.

7.2 Large-field benchmarks and method

Using the q -ary entropy convention from Section 2, the Gilbert–Varshamov relative distance solves $H_q(\delta_{\text{GV}}) = 1 - R$. For $q = 2^{128}$, the rate-1/2 and rate-1/4 benchmarks are

$$\delta_{\text{GV}}(1/2) \approx 0.492, \quad \delta_{\text{GV}}(1/4) \approx 0.744. \quad (16)$$

For the finite-length estimates below, we hold the global dimension at $k = 2^{20}$ across rates. This keeps the application payload fixed, giving $n = 2^{21}$ at rate 1/2 and $n = 2^{22}$ at rate 1/4; it does not equalize encoding work, which scales with n . The corresponding Singleton ceilings are 0.5 and 0.75. For each local block below, the calculation maximizes (36) over its feasible domain and finds the distance at which the maximum crosses zero.

7.3 Large-field instantiations

The RS $[16, 8, 9]$ and $[16, 4, 13]$ constituents give estimated transitions 0.305 and 0.604 at rates $1/2$ and $1/4$, respectively. The latter is about 81% of large-field GV. At the optimizing points, the cancellation constraints nearly exhaust the projective dimension; at rate $1/4$, dense local shortenings dominate rather than minimum-weight local words. Appendix D.2 records the substitutions, optimizer coordinates, and sparse-term diagnostics. The following theorem gives the complete finite-length guarantees at distances 0.30 and 0.60.

7.4 Certified finite-length RS bounds

The following theorem bounds every support size at finite length, including all surplus cancellations. Its proof combines explicit sparse sums with finite envelopes for the remaining supports.

Theorem 7.1 (Certified finite-length RS bounds). *Let $q \geq 2^{127}$ be a prime power and $k = 2^{20}$. Fix an RS constituent $C_0 = \text{RS}[16, \sigma, 17 - \sigma]$ over $\mathbb{F}_q$, and construct chosen-block BAA with two independent uniform permutations and independent uniform nonzero diagonal entries. Then*

$$\begin{array}{ccc} \sigma & n & \delta \\ 4 & 2^{22} & 0.60 \\ 8 & 2^{21} & 0.30 \end{array} \quad \begin{array}{l} \Pr[d_{\min}(C_{\text{BAA}}) \leq \lfloor \delta n \rfloor] \\ < 2^{-209.60} \\ < 2^{-137.56}. \end{array}$$

Proof. Restrict the master union bound to possible exact outer supports. For a support of size s meeting t local blocks, the MDS shortening dimension is $r = s - (16 - \sigma)t$. Put $T = \lfloor n/1000 \rfloor$. Appendix H evaluates the one-block terms by finite binomial sums and bounds the omitted intermediate weights by a convergent positive series. Geometric envelopes cover every $2 \leq t \leq T$ and all remaining cancellation layers. For $t > T$, finite entropy inequalities and an interval-verified covering bound the full support/trace sum, with explicit counting prefactors. Exact rational arithmetic and 192-bit Arb intervals verify the two displayed comparisons. The complete executable and certificate data are provided in the accompanying repository [28].

The theorem applies to every 128-bit prime field as well as $\mathbb{F}_{2^{128}}$. It concerns the uniform ensemble; the timings below use a Feistel-based permutation implementation, whose distribution is not certified by this theorem. The new analytic reductions and numerical certificates are outside the Lean formalization. They do not certify the estimated asymptotic transitions.

7.5 Structural transitions

Entropy optimization of (47) gives the following estimated transitions:

scheme	trace first moment direct structural container	
RS[16, 8, 9]	0.305	0.308
RS[16, 4, 13]	0.604	0.609

(17)

At the trace thresholds, the direct-container exponents remain approximately -0.011 and -0.021 bits per coordinate, respectively. Thus trace clustering creates real slack, but the recovered relative distance is only about 0.003 and 0.005 .

Appendix E.2 records the optimizing support profiles and the associated moderate-length experiments.

A positive exponent in (47) is not a typical-code upper bound without a concentration argument for structural containers. Still, the coefficient-independent transition lies only 0.003 or 0.005 beyond the trace threshold. The numerical gap to q -ary GV is much larger.

7.6 Representative large-field repeat-accumulate configurations

For construction-level context over the same field, we use weighted repeat-accumulate ^{t} (RA ^{t}), whose generator has the form

$$G_{\text{RA}^t} = R_r M_1 A \cdots M_t A.$$

Here R_r repeats each of k_0 message symbols r times, $N = rk_0$, and each M_i is an independent permutation followed by independent nonzero coordinate scalings. For $t = 2$ this is the standard randomly scaled large-field RAA ensemble [2, 20, 31].

A single accumulator is not asymptotically good, so we consider $t \geq 2$. Each additional accumulator adds one full length- N scan, but also suppresses the no-return weight-one event by another polynomial factor. Numerical optimization of the support/trace exponent gives transition estimates 0.169 for repetition-2 RAA and 0.447 for repetition-3 RAA; a third accumulator raises the repetition-2 estimate to 0.415 . The sharp growing-field theorem currently assumes $r \geq 4$ [20]. The representative repetition-2 configurations therefore require a separate finite-length calculation.

At rate $1/4$, repetition-4 RAA does fall within the large-field theorems. Akhiani and Zhang report relative distance $1/2$ with failure probability 2^{-20} at message dimension 2^{20} over a 31-bit prime field [2]. Khabbaziyan proves the asymptotic failure order $\Theta(N^{-3})$ for this repetition factor and any fixed target distance up to $1/2$, when the field size grows at least linearly with the output length N [20]. Our rate- $1/4$ chosen-block result instead certifies distance 0.60 with failure below $2^{-209.60}$ at the same message dimension, over fields of size at least 2^{127} . These concrete failure figures use different field sizes; the asymptotic RAA theorem alone does not supply a concrete failure constant at our parameters.

Appendix D.3 gives the full support/trace sum for weighted RA ^{t} , obtained by composing the same placement and cancellation bounds.

Theorem 7.1 certifies the two chosen-block RS configurations. We have not established a complete numerical upper bound for (43) at the RA ^{t} configurations below. Direct sparse contributions and asymptotic margins do not control the omitted message weights and cancellation layers. We retain these rows as runtime and parameter-search diagnostics, and leave their failure bounds uncertified.

At $k_0 = 2^{20}$, Section 8 tabulates representative RA ^{t} and chosen-block BAA configurations. We do not designate pairs as “matched”: changing repetition,

accumulator count, or local code jointly changes rate, target distance, failure probability, and encoding work. Each row therefore uses a meaningful distance below its own estimated transition; the lower-rate rows use correspondingly larger distances.

7.7 Comparison across alphabets

Table 2 places the first-moment and structural transitions on one scale. Distances use the natural symbol Hamming metric for each alphabet.

Alphabet	Rate	Local code	Trace/enumerator δ^*	Rank-container δ_{str}	GV	Sparse scale
$\mathbb{F}_2$	1/2	QR [168, 84, 24]	0.110	–	0.110	n^{-11}
$\mathbb{F}_2$	1/2	Golay [24, 12, 8]	0.105	–	0.110	n^{-3}
$\mathbb{F}_2$	1/2	extended BCH / RM(2, 5) [32, 16, 8]	0.106	–	0.110	n^{-3}
large $\mathbb{F}_q$	1/2	RS [16, 8, 9]	0.305	0.308	0.492	n^{-8}
$\mathbb{F}_2$	1/4	BCH subcode [32, 8, 12]	0.213	–	0.215	n^{-5}
large $\mathbb{F}_q$	1/4	RS [16, 4, 13]	0.604	0.609	0.744	n^{-12}

Table 2. Numerical transitions. Binary rows use the exact weight-enumerator first moment. Large-field trace rows use the support/shortening first moment, while rank-container rows count the coefficient-independent obstruction. The q -ary GV values use $q = 2^{128}$. A local code labeled $[m, \sigma, d]$ has block length m , dimension σ , and minimum Hamming distance d .

At equal rate, large-field BAA has about 2.8 times the relative symbol distance of binary BAA and smaller sparse code-generation tails. Relative to the alphabet-specific benchmark, however, the binary schemes are much closer to GV. At the primary rate-1/2 target, QR [168, 84, 24] reaches 99.99% of binary GV; the compact implemented extended BCH / RM(2, 5) block reaches 96.5%. Thus the paper already contains a rate-1/2 BCH result, while the secondary rate-1/4 BCH subcode reaches 99.4%. The large-field trace transitions reach 62.0% and 81.3% of their q -ary GV values; the structural thresholds improve these fractions only to 62.7% and 81.9%.

The difference comes from scalar freedom. Over $\mathbb{F}_2$, support parity fixes the accumulator state. Over a large field, an r -dimensional shortening has about q^{r-1} projective candidates and can absorb as many as $r - 1$ homogeneous cancellation equations without a $1/q$ penalty. Projectivization removes duplicate scalar words, but it does not remove block saturation.

7.8 Design implications and open questions

To test whether a stronger local code alone can close the large-field gap, we repeated the trace optimization with rate-matched MDS constituents of increasing size. Table 3 shows rapidly diminishing gains.

Rate 1/2		Rate 1/4	
MDS block $[m, \sigma, d]$	δ_{trace}	MDS block $[m, \sigma, d]$	δ_{trace}
[8, 4, 5]	0.285	[8, 2, 7]	0.588
[12, 6, 7]	0.299	[12, 3, 10]	0.599
[16, 8, 9]	0.305	[16, 4, 13]	0.604
[20, 10, 11]	0.309	[20, 5, 16]	0.607
[24, 12, 13]	0.312	[24, 6, 19]	0.609
[32, 16, 17]	0.315	[32, 8, 25]	0.612
[64, 32, 33]	0.319	[64, 16, 49]	0.615

Table 3. Estimated trace transitions for rate-matched large-field MDS constituents. In $[m, \sigma, d]$, m is the local block length, σ its dimension, and d its minimum Hamming distance; δ_{trace} is the estimated relative-distance transition from the trace first moment. The numerical large-block trends are approximately 0.323 at rate 1/2 and 0.618 at rate 1/4.

The intuition is that a larger MDS block reduces the number of ways to choose active local blocks, but it does not change the one-field-element state carried by each accumulator. Two backward neighborhoods can still overload the same local parity-check capacities. Since an MDS code already minimizes every shortening dimension for fixed (m, σ) , the flattening in Table 3 points to the standard BAA architecture, rather than the local code or trace multiplicity, as the main gap to q -ary GV.

The first open problem is to certify the optimized transition itself, uniformly in block length and without relying on floating-point saddle searches. The pointwise finite-length evaluations in Table 5 certify distances 0.30 and 0.60 for the two implemented BAA rows, but they do not turn the estimated transition values 0.305 and 0.604 into interval-certified thresholds. Closing that gap requires interval optimization of the trace exponents and uniform control of the finite- n prefactors.

A second open problem is concentration of the structural-container count Z_t around the transitions in (17), including cuts that use intermediate-layer vertices rather than only outer-block capacity. A matching upper and lower analysis would characterize the ensemble distance. On the construction side, two natural BAA variants are adding a third independent accumulator layer and encoding below the target rate before puncturing. Both remain asymptotically linear, but increase the amount of work or data processed. Whether either retains BAA’s practical advantage while improving the structural cut is open.

8 Implementation and performance evaluation

8.1 Implementation

We implemented multiplication by $G_{\text{BAA}} = \mathbf{B}M_1AM_2A$ and the weighted RA^t construction in about 1,000 lines of C++20 in libOTe, excluding tests and benchmark drivers. Source code and reproduction instructions are provided

in the accompanying repository [28]. The implementation supports the binary Golay [24, 12, 8], $\text{RM}(2, 5) = [32, 16, 8]$, and BCH [32, 8, 12] constituents and original-BAA random systematic blocks, as well as systematic $[2, 1, 2]$, $[4, 2, 3]$, $[16, 8, 9]$, and $[16, 4, 13]$ Reed–Solomon constituents over $\mathbb{F}_{2^{128}}$. It provides single- and multi-threaded execution of the two accumulator layers, global interleavers, and optimized local-code maps; diagonal weighters are omitted in the binary case, while the large-field path uses carry-less multiplication. The binary path is bitsliced over libOTe’s 128-bit blocks: one call on n blocks evaluates 128 independent length- n binary inputs in parallel, rather than packing one binary input into $\lceil n/128 \rceil$ blocks. The weighted RA^t implementation accepts arbitrary positive accumulator counts, uses the same fused large-field scan–multiply–scatter loop with an independently sampled interleaver and weighter in every stage, and provides optimized segment reductions for repetition 2, 3, and 7. The low-security BAA paths similarly use straight-line local maps; in particular, RS $[2, 1, 2]$ reduces to one XOR and performs no needless multiplication by the field identity.

8.2 Experimental methodology

Common-workload original-BAA control. The baseline in Table 4 is a same-host implementation control based on the original `ppcg-peter` CPU branch, not a replay of the exact executable timed in the original paper. That branch’s benchmark constructs one accumulator object and invokes it twice, thereby reusing one permutation; its block step also has overlapping mask-buffer writes and the wrong systematic-dual orientation. Our control instead uses independent permutations, consumes every matrix bit once, and computes the intended $x \leftarrow x + Ry$ map. These corrections preserve the construction and its work profile, but change the exact instruction stream. At $\sigma = 32$, every local factor is an independently sampled 32×64 systematic binary generator matrix $[I_{32} \mid R]$, equivalently a random $[64, 32]$ binary code. This is the conservative rate-1/2 parameter set for which the original evaluation reports relative distance 0.10 and one-time code-sampling failure at most 2^{-20} [22]. We use the same target for the chosen-block rows. Their numerical distance transitions, 0.105 for Golay and 0.106 for Reed–Muller, lie above 0.10. At the measured length, the complete numerical evaluation described in Section 7 gives $\mu_{\text{bin}} \leq 2^{-46.386}$ and $2^{-47.093}$, respectively. Thus both chosen rows themselves exceed 40 bits at relative distance 0.10 with no parameter or runtime change. The comparison criterion is the complete exact binary first moment in Theorem 4.2, rather than equality of local dimensions. The original $\sigma = 32$ row, however, is documented at only 20 bits. We therefore report Table 4 as a matched 20-bit comparison. A matched binary 40-bit speedup would still require benchmarking a stronger original-BAA parameter set; the original paper’s empirical rule $\sigma \approx 4\lambda/3$ suggests $\sigma \approx 54$ before rounding to a convenient block size when $\lambda = 40$ [22].

Workload and platform. The binary comparison uses $n = 2,097,024$ and $k = n/2$, so every constituent tiles the same input. All rows use the same input and permutation seeds. The timed operation includes both accumulator/permutation

layers and the local multiplication, including generation of the original control’s random local matrices; setup and workspace allocation are excluded. Large-field rows fix $k = 2^{20}$ and use the same timing boundary.

Measurements use an Intel Core i9-14900HX, MSVC 14.44 in Release mode, and one CPU thread unless stated otherwise. We report medians over repeated processes and calls. The implementation uses pseudorandom permutations; the mathematical guarantees concern uniform permutations. Appendix F.1 specifies seeds, affinity, scheduling, and the complete measurement procedure.

8.3 Performance results

Construction	Local outer code	δ	$\Pr[d_{\min} \leq \delta n]$	Threads	ms	vs. control
original-BAA-style control	random systematic $[64, 32]$, $\sigma = 32$	0.10	$\leq 2^{-20}$	1	29.596	1.00×
chosen-block BAA	Golay $[24, 12, 8]$	0.10	$\leq 2^{-46.386}$	1	22.810	1.30×
chosen-block BAA	RM(2, 5) $[32, 16, 8]$	0.10	$\leq 2^{-47.093}$	1	24.278	1.22×

Table 4. Common-workload, single-CPU-thread, rate-1/2 comparison. Every row uses $k = 1,048,512$, the same input and permutation seeds, the same two independently seeded pseudorandom permutations, and the same timed operation. The failure column reports the original construction’s documented code-sampling target and the complete chosen-constituent first-moment bounds in (27). A constituent labeled $[m, \sigma, d]$ has block length m , dimension σ , and minimum Hamming distance d ; $[m, \sigma]$ omits the distance.

Table 4 reports construction-level improvement over our original-BAA-style control, using one CPU thread throughout. Golay reduces latency by 22.9% (1.30×), and Reed–Muller reduces it by 18.0% (1.22×). The control performs the $\sigma = 32$ random local map, whereas chosen-block BAA uses a 56-XOR Golay circuit or a five-variable Reed–Muller zeta transform. The common accumulator and permutation work is present in all rows and is therefore not counted as a chosen-block speedup. The chosen rows exceed the 40-bit target at these same parameters; the control carries its documented 20-bit target. The speedup therefore compares the displayed constructions rather than equal failure targets.

Appendix F.2 compares the common-workload control with the original paper’s timing setup and checks the effect of the small length adjustment. Absolute times from different hosts and campaigns are not directly comparable.

Representative $\mathbb{F}_{2^{128}}$ configurations. Table 5 reports target distances, measured runtimes, and complete finite-length failure bounds where certified.

The configurations differ in repetition, accumulator count, local code, rate, target distance, and length. The RS $[16, 8, 9]$ and $[16, 4, 13]$ uniform ensembles have complete failure bounds at distances 0.30 and 0.60, respectively. The RA rows provide runtime context, but their uncertified probabilities do not support a comparison at equal failure targets.

Construction	Outer map	t	Rate	δ	$\Pr[d_{\min} \leq \delta n]$	Time (ms)
RA ²	repetition 2	2	1/2	0.10	uncertified	22.832
chosen-block BAA	RS [16, 8, 9]	2	1/2	0.30	$\leq 2^{-137.56}$	33.633
RA ³	repetition 2	3	1/2	0.40	uncertified	32.872
RA ²	repetition 3	2	1/3	0.40	uncertified	48.222
chosen-block BAA	RS [16, 4, 13]	2	1/4	0.60	$\leq 2^{-209.60}$	73.739

Table 5. Representative large-field configurations at $k = 2^{20}$ over $\mathbb{F}_{2^{128}}$; the output length is determined by the displayed rate. The RS failure bounds are certified for the uniform ensemble by Theorem 7.1, including all support classes and cancellation traces. RA target distances are parameter choices without certified failure bounds. Time is the median of six process medians, each with 31 calls after warm-up, on one CPU thread; setup and allocation are excluded. The rows are representative configurations, not matched pairs or a ranking.

Multithreaded scaling. The headline construction comparison uses one CPU thread. The implementation also supports a persistent CPU worker team: each worker scans a contiguous range, the range totals determine prefix offsets, and the workers rescan and permute their ranges in parallel. In a separate paired scaling campaign, chosen-block Golay drops from 22.632 ms on one thread to 5.605 ms on 24 threads, a $4.04\times$ improvement. Chosen-block Reed–Muller drops from 24.455 ms to 5.385 ms on 32 threads, a $4.54\times$ improvement. These are CPU-scaling results, not part of the comparison with the original-BAA-style control.

9 Lean formalization and proof audit

Using the Lean 4 proof assistant described in Section 2.3, we formalized the finite algebraic and combinatorial arguments in mathlib. The development focuses on points where informal notation can hide a boundary case: sample spaces, index ranges, endpoint conventions, and shortening hypotheses. The Lean development is provided in the accompanying repository [28].

9.1 Verified scope

The Lean development verifies the exact binary first moment, the nonbinary placement and projective bounds, and the finite large-field master bound. It also verifies the chosen Golay, Reed–Muller, and BCH constituents, the RS parameters and shortening profile, and the generic-rank and structural-container identities. Appendix G.1 gives the representation choices and a statement-by-statement inventory. The analytic reductions, numerical certificates, asymptotic transitions, and QR spectra remain outside Lean.

9.2 Audit findings and trust boundary

Formalization required three corrections to the informal statements.

1. Formula (2) requires $h \geq 1$; a separate theorem handles the impossible output weight $h = 0$.
2. The Singleton step before (9) requires every touched block to have a nontrivial local shortening. Exact supports of outer codewords meet this condition, which is the case used later.
3. Equation (47) requires two endpoint flags, e_1, e_2 , to give an exact finite equality. Suppressing the flags remains valid only at the entropy-exponent level.

None of the corrections changes the BAA construction, the finite master bound, or the reported asymptotic exponents. No contradiction was found within the formalized finite scope. The analytic and numerical claims lie outside that scope.

Trust boundary. The formalized theorems use only the standard Lean/mathlib foundations, with no admitted proofs or project-local axioms. Concrete constituent spectra and distances have kernel-checkable proofs. Appendix G.2 records dependencies, commands, regression checks, and the detailed boundary of the formalization.

References

1. Abbaszadeh, K., Hafezi, H., Katz, J., Meiklejohn, S.: Single-server private outsourcing of zk-SNARKs. In: IEEE Symposium on Security and Privacy. pp. 3075–3090 (2026). <https://doi.org/10.1109/SP63933.2026.00194>, also IACR Cryptology ePrint Archive, Report 2025/2113
2. Akhiani, P., Zhang, Y.: Distance of RAA codes over large finite fields (with applications in zkSNARKs and PCGs). IACR Cryptology ePrint Archive, Report 2026/524 (2026)
3. Baweja, A., Fenzi, G., Mishra, P., Mopuri, T.: Field-agnostic SNARKs with small proofs via encode-repeat-accumulate (ERA) codes. IACR Cryptology ePrint Archive, Report 2026/864 (2026)
4. Bazzi, L., Mahdian, M., Spielman, D.A.: The minimum distance of turbo-like codes. IEEE Transactions on Information Theory **55**(1), 6–15 (2009)
5. Benedetto, S., Divsalar, D., Montorsi, G., Pollara, F.: Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding. IEEE Transactions on Information Theory **44**(3), 909–926 (1998)
6. Benedetto, S., Montorsi, G.: Unveiling turbo codes: Some results on parallel concatenated coding schemes. IEEE Transactions on Information Theory **42**(2), 409–428 (1996). <https://doi.org/10.1109/18.485713>
7. Berrou, C., Glavieux, A., Thitimajshima, P.: Near Shannon limit error-correcting coding and decoding: Turbo codes. In: Proceedings of ICC '93. pp. 1064–1070 (1993)

8. Block, A.R., Fang, Z., Katz, J., Thaler, J., Waldner, H., Zhang, Y.: Field-agnostic SNARKs from expand-accumulate codes. In: *Advances in Cryptology — CRYPTO 2024*. LNCS, vol. 14929, pp. 276–307 (2024)
9. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y.: Compressing vector OLE. In: *ACM CCS 2018*. pp. 896–912 (2018)
10. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Resch, N., Scholl, P.: Correlated pseudorandomness from expand-accumulate codes. In: *Advances in Cryptology — CRYPTO 2022, Part II*. LNCS, vol. 13508, pp. 603–633 (2022)
11. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Scholl, P.: Efficient pseudorandom correlation generators: Silent OT extension and more. In: *Advances in Cryptology — CRYPTO 2019, Part III*. LNCS, vol. 11694, pp. 489–518 (2019)
12. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Scholl, P.: Efficient pseudorandom correlation generators from Ring-LPN. In: *Advances in Cryptology — CRYPTO 2020, Part II*. LNCS, vol. 12171, pp. 387–416 (2020)
13. Brehm, M., Chen, B., Fisch, B., Resch, N., Rothblum, R.D., Zeilberger, H.: Blaze: Fast SNARKs from interleaved RAA codes. In: *Advances in Cryptology — EURO-CRYPT 2025*. pp. 123–152 (2025), also IACR Cryptology ePrint Archive, Report 2024/1609
14. Couteau, G., Rindal, P., Raghuraman, S.: Silver: Silent VOLE and oblivious transfer from hardness of decoding structured LDPC codes. In: *Advances in Cryptology — CRYPTO 2021, Part III*. LNCS, vol. 12827, pp. 502–534 (2021)
15. Divsalar, D., Jin, H., McEliece, R.J.: Coding theorems for “turbo-like” codes. In: 36th Allerton Conference on Communication, Control, and Computing. pp. 201–210 (1998)
16. Golovnev, A., Lee, J., Setty, S., Thaler, J., Wahby, R.S.: Brakedown: Linear-time and field-agnostic SNARKs for R1CS. In: *Advances in Cryptology — CRYPTO 2023, Part II*. LNCS, vol. 14082, pp. 193–226 (2023)
17. Gurkan, K., Novakovic, A., Rothblum, R.D.: Bolt: Faster SNARKs from sketched codes. In: *Advances in Cryptology — CRYPTO 2026, Part IX*. LNCS, vol. 16808, pp. 396–427 (2026). https://doi.org/10.1007/978-3-032-35424-2_14, also IACR Cryptology ePrint Archive, Report 2026/310
18. Huang, K., Guo, Y., Qu, W., Zhang, J.: Linear-time, constant-depth blind polynomial commitments from generalized RAA codes, with an end-to-end blind SNARK implementation. IACR Cryptology ePrint Archive, Report 2026/487 (2026)
19. Kahale, N., Urbanke, R.: On the minimum distance of parallel and serially concatenated codes. In: *IEEE International Symposium on Information Theory*. p. 31 (1998)
20. Khabbazi, M.: Sharp minimum-distance lower tails for RAA codes. IACR Cryptology ePrint Archive, Report 2026/1075 (2026)
21. Kliewer, J., Zigangirov, K.S., Koller, C., Costello, Jr., D.J.: Coding theorems for repeat multiple accumulate codes. arXiv:0810.3422 (2008)
22. Kolesnikov, V., Peceny, S., Rachuri, R., Raghuraman, S., Rindal, P., Shah, H.: Block-accumulate codes: Accelerated linear codes for PCGs and ZK. In: *Advances in Cryptology — CRYPTO 2026, Part VIII*. LNCS, vol. 16807, pp. 397–429 (2026), also IACR Cryptology ePrint Archive, Report 2025/1828
23. Kolesnikov, V., Peceny, S., Raghuraman, S., Rindal, P.: Stationary syndrome decoding for improved PCGs. In: *Advances in Cryptology — CRYPTO 2025*. pp. 284–317 (2025), also IACR Cryptology ePrint Archive, Report 2025/295
24. Li, W., Zhang, Z., Chow, S.S.M., Guo, Y., Gao, B., Song, X., Deng, Y., Liu, J.: Shred-to-shine metamorphosis of (distributed) polynomial commitments. In: *USENIX Security 2026* (2026)

25. Li, Z., Xing, C., Yao, Y., Yuan, C.: Efficient pseudorandom correlation generators for any finite field. In: *Advances in Cryptology — EUROCRYPT 2025, Part V*. LNCS, vol. 15605, pp. 145–175 (2025), also IACR Cryptology ePrint Archive, Report 2025/169
26. MacWilliams, F.J., Sloane, N.J.A.: *The Theory of Error-Correcting Codes*. North-Holland (1977)
27. de Moura, L., Ullrich, S.: The Lean 4 theorem prover and programming language. In: *Automated Deduction — CADE-28*. LNCS, vol. 12699, pp. 625–635. Springer (2021). https://doi.org/10.1007/978-3-030-79876-5_37
28. Peceny, S., Rindal, P.: Chosen-block BAA: Software and proof artifacts. <https://github.com/ladnir/gen-BAA> (2026)
29. Raghuraman, S., Rindal, P., Tanguy, T.: Expand-convolute codes for pseudorandom correlation generators from LPN. In: *Advances in Cryptology — CRYPTO 2023, Part IV*. LNCS, vol. 14084, pp. 602–632 (2023)
30. Ron-Zewi, N., Rothblum, R.D.: Proving as fast as computing: Succinct arguments with constant prover overhead. *Journal of the ACM* **72**(2), 15:1–15:54 (2025). <https://doi.org/10.1145/3721477>
31. Rosnes, E., Graell i Amat, A.: On the analysis of weighted nonbinary repeat multiple-accumulate codes. arXiv:1101.5966 (2011), a preliminary version appeared at the 2011 Information Theory and Applications Workshop
32. Su, W.K., Lee, C.D., Lin, T.C., Truong, T.K., Chang, Y.: On determination of the weight distribution of binary (168, 84, 24) extended quadratic residue code. In: *IEEE International Symposium on Information Theory*. pp. 644–646 (2008). <https://doi.org/10.1109/ISIT.2008.4595065>
33. The mathlib Community: The Lean mathematical library. In: *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*. pp. 367–381. ACM (2020). <https://doi.org/10.1145/3372885.3373824>
34. Xie, T., Zhang, Y., Song, D.: Orion: Zero knowledge proof with linear prover time. In: *Advances in Cryptology — CRYPTO 2022*. pp. 299–328 (2022), also IACR Cryptology ePrint Archive, Report 2022/1010
35. Yang, K.: Weighted nonbinary repeat-accumulate codes. *IEEE Transactions on Information Theory* **50**(3), 527–531 (2004). <https://doi.org/10.1109/TIT.2004.825255>

A Explicit binary [32,8,12] subcode

The following eight 32-bit rows generate the rate-1/4 binary block used in Section 4. Hexadecimal words are interpreted with the least-significant bit as coordinate zero:

```
FFFFFFFF B7DC4AE7 AA9D2399 60E73723
6844CC86 180D3C68 52610E07 19DE6790.
```

The rows are linearly independent. Exhaustive enumeration of the resulting 2^8 words gives minimum distance 12 and the exact spectrum

$$1 + 48z^{12} + 158z^{16} + 48z^{20} + z^{32}.$$

The code is a subcode of the extended primitive BCH code [32, 11, 12].

B Exact rate-1/2 QR spectrum

For completeness, this appendix records the exact spectrum used for the strongest rate-1/2 row. Let A_w denote the number of weight- w words in the extended binary quadratic-residue code [168, 84, 24]. The distribution is symmetric, $A_{168-w} = A_w$, with $A_0 = A_{168} = 1$. All coefficients not implied by the following table are zero [32].

w	A_w	w	A_w
24	776216	56	372099690249351071112
28	18130188	60	4739291519495550245228
32	5550332508	64	39973673337590380474086
36	1251282702264	68	225696677727188690570184
40	166071600559137	72	860241108921860741947676
44	13047136918828740	76	2227390683565491780127428
48	629048543890724216	80	3935099586463594172460648
52	19087130695796615088	84	4755747412595715344169376

Equivalently,

$$W_{168}(z) = 1 + z^{168} + A_{84}z^{84} + \sum_{\substack{24 \leq w < 84 \\ A_w \neq 0}} A_w (z^w + z^{168-w}).$$

The coefficients sum to 2^{84} , as required by the dimension.

C Additional binary analysis

C.1 Binary constituents and their spectra

The binary evaluation uses four well-known families with exact parameters and weight spectra. The Golay, Reed–Muller, and BCH constituents are certified in the accompanying Lean development. The longer quadratic-residue spectra are standard algebraic or published enumerators and are not part of that formal certificate.

Reed–Muller codes. The code $\text{RM}(r, \ell)$ is obtained by evaluating every multilinear polynomial in ℓ binary variables of total degree at most r on all points of $\mathbb{F}_2^\ell$. It has parameters

$$\left[2^\ell, \sum_{i=0}^r \binom{\ell}{i}, 2^{\ell-r} \right]_2.$$

In particular, $\text{RM}(2, 5)$ is a $[32, 16, 8]_2$ code. Up to a permutation of coordinates, it is also the parity extension of the primitive triple-error-correcting BCH code $[31, 16, 7]_2$, and is therefore often called the extended BCH code $[32, 16, 8]_2$ [26]. Here “equivalent” means that the two descriptions differ only by a relabeling of coordinates; they have the same weight spectrum and give exactly the same BAA transition. Thus the $\text{RM}(2, 5)$ calculation below is already a rate-1/2 BCH result.

BCH codes. A primitive binary BCH code is a cyclic code of length $2^\ell - 1$ specified by requiring its code polynomials to vanish at a consecutive set of powers of a primitive element of $\mathbb{F}_{2^\ell}$. The chosen roots give a designed-distance lower bound; extending the code by an overall parity coordinate can increase its length and distance. A subcode is simply a linear subspace of the parent code. The rate-1/4 block used here is an explicitly generated $[32, 8, 12]_2$ subcode of an extended primitive $[32, 11, 12]_2$ BCH code. Appendix A gives its generator and its exact exhaustive weight enumeration, so the analysis does not rely only on the BCH designed distance.

The extended Golay code. The extended binary Golay code is the exceptional self-dual $[24, 12, 8]_2$ code. Along with the extended BCH code (equivalently $\text{RM}(2, 5)$), it supplies a compact rate-1/2 block with a known complete weight enumerator.

Extended quadratic-residue codes. Extending the binary quadratic-residue code of prime length $p \equiv -1 \pmod{8}$ gives a doubly even self-dual code of length $p+1$ and rate 1/2. We evaluate the concrete constituents $[48, 24, 12]$, $[80, 40, 16]$, $[104, 52, 20]$, and $[168, 84, 24]$. For the first three, Type-II extremality and Gleason's theorem fix the complete weight enumerator from the displayed parameters. For length 168, we use the exact weight distribution determined by Su et al. [32]; Appendix B records every nonzero coefficient used in our calculation.

C.2 Accumulator transition proof

The run-counting argument for Lemma 4.1 is recalled here for completeness.

Proof. A nonzero input coordinate toggles the accumulator state. Decomposing the output into alternating zero-runs and one-runs gives a bijection with two run compositions. The positive lengths of the r_w one-runs contribute the first binomial coefficient. The s_w required zero gaps contribute the second. The inverse difference map proves both injectivity and surjectivity of this correspondence. Invertibility excludes a zero output for nonzero input.

C.3 Binary asymptotic calculations

Fix w and a target $0 < \delta < 1/2$. The exact transition (3) gives, for fixed h ,

$$P_n^{\text{bin}}(w, h) = \Theta_{w,h} \left(n^{-\lceil w/2 \rceil} \right).$$

The second accumulator tail from fixed h converges to a nonzero beta-tail constant. Splitting the intermediate-weight sum into bounded and growing ranges therefore gives

$$\sum_h P_n^{\text{bin}}(w, h) Q_n^{\text{bin}}(h, \lfloor \delta n \rfloor) = \Theta_{w,\delta} \left(n^{-\lceil w/2 \rceil} \right). \quad (18)$$

Hence the one-active-block contribution is

$$\frac{k}{\sigma} \sum_{w=1}^m A_w^{(0)} \Theta_{w,\delta} \left(n^{-\lceil w/2 \rceil} \right). \quad (19)$$

If the local distance is d , this is at most

$$O\left(n^{1-\lceil d/2 \rceil}\right),$$

with the exact leading term determined by all local weights attaining the smallest value of $\lceil w/2 \rceil$.

Using only local distance, a fixed family touching t blocks has the coarse scale

$$O\left(n^{t-\lceil td/2 \rceil}\right). \quad (20)$$

Thus a constant local distance of at least three is consistent with a vanishing fixed- t sparse tail. The earlier binary BA proof assumes $d > 4$ because it uses coarser inequalities. For a fixed local block, its exact spectrum and the two-accumulator kernel (4) give the sharper bound.

Linear weights. We use the entropy analysis of accumulator transitions developed for RAA/RMA codes [21, Sections III–IV]. For the outer stage, the elementary coefficient bound $[z^w]W_0(z)^{n/m} \leq W_0(z)^{n/m} z^{-w}$ for $z > 0$ gives the direct-sum spectrum the upper exponent

$$\mathcal{A}_{C_0}(\alpha) := \inf_{z>0} \left\{ \frac{1}{m} \log W_0(z) - \alpha \log z \right\}. \quad (21)$$

Indeed,

$$A_{[\alpha n]}^{\text{out}} \leq \exp(n\mathcal{A}_{C_0}(\alpha) + O(\log n)).$$

For linear input and output weights un and vn , the binary accumulator transition exponent is

$$\Gamma_{\text{bin}}(u, v) := vH\left(\frac{u}{2v}\right) + (1-v)H\left(\frac{u}{2(1-v)}\right) - H(u), \quad (22)$$

with feasibility $u/2 \leq v \leq 1 - u/2$. For $\delta < 1/2$, this exponent is increasing in the final output fraction up to $1/2$, so the low-weight tail is maximized at the boundary $v = \delta$.

The binary BAA first-moment exponent is therefore

$$\Phi_{C_0,\delta}^{\text{bin}}(\alpha, \eta) = \mathcal{A}_{C_0}(\alpha) + \Gamma_{\text{bin}}(\alpha, \eta) + \Gamma_{\text{bin}}(\eta, \delta). \quad (23)$$

For a fixed $\varepsilon > 0$, a sufficient condition for the contribution from outer weights $w \geq \varepsilon n$ to vanish is

$$\sup_{\substack{\alpha \geq \varepsilon, \eta \\ \text{feasible}}} \Phi_{C_0,\delta}^{\text{bin}}(\alpha, \eta) < 0 \quad (24)$$

on this restricted domain. The unrestricted supremum is at least zero because the exponent tends to zero at the sparse boundary. For a fixed block C_0 , (4) gives the exact finite-length first moment, (19) controls fixed-block supports, and (24) controls linear weights.

C.4 Binary numerical certificates

Rate 1/2: extended Golay. The extended binary Golay code has parameters $[24, 12, 8]$ and spectrum

$$W_{\text{Golay}}(z) = 1 + 759z^8 + 2576z^{12} + 759z^{16} + z^{24}.$$

The numerical BAA transition is

$$\delta_{\text{Golay}}^* \approx 0.105.$$

The contribution from one active block has order n^{-3} .

Rate 1/2: extended BCH / RM(2, 5). The extended BCH code $[32, 16, 8]$, equivalently the second-order Reed–Muller code $\text{RM}(2, 5)$, has spectrum

$$W_{\text{RM}(2,5)}(z) = 1 + 620z^8 + 13888z^{12} + 36518z^{16} + 13888z^{20} + 620z^{24} + z^{32}.$$

The numerical BAA transition is

$$\delta_{\text{eBCH/RM}(2,5)}^* \approx 0.106.$$

This is our compact rate-1/2 BCH baseline. Among the two compact implemented blocks, it gives the larger transition: about 96.5% of the binary GV relative distance. The longer rate-1/2 quadratic-residue spectra below improve on this value. The one-block contribution again has order n^{-3} .

Finite-length 40-bit certificate for the implemented blocks. We evaluate the complete sum (4) at the benchmark length $n = 2,097,024$ and $L = \lfloor 0.10n \rfloor = 209,702$. To make the calculation both exhaustive and reproducible, first observe that $Q_n^{\text{bin}}(h, L)$ is the upper tail $\Pr[X \geq x_h]$ for $X \sim \text{Hypergeom}(n, L, h)$ and $x_h = \lceil h/2 \rceil$. If

$$\rho_h = \frac{L - x_h}{x_h + 1} \frac{h - x_h}{n - L - h + x_h + 1},$$

then successive terms of this tail have decreasing ratio, and hence

$$Q_n^{\text{bin}}(h, L) \leq \frac{\binom{L}{x_h} \binom{n-L}{h-x_h}}{\binom{n}{h} (1 - \rho_h)}. \quad (25)$$

We compute the coefficients A_w^{out} exactly through $w = 32$ and sum the resulting bound over every feasible intermediate weight h . For the remaining weights, the coefficient inequality

$$A_w^{\text{out}} \leq \exp(B \log W_0(e^\theta) - w\theta), \quad B = n/m, \quad (26)$$

holds for every real θ . We minimize this bound numerically. An explicit consecutive-ratio check shows that the bounded transition summand is unimodal on each of the even and odd h subsequences; we maximize the two subsequences separately and multiply their maximum by the number of feasible h . No $w > 4L$ term can contribute, because each binary accumulator has output weight at least half its nonzero input weight. A full one-bit outward-rounding allowance is applied independently to the $w \leq 32$ and $w > 32$ subtotals.

The resulting complete first-moment certificates are

$$\boxed{\begin{aligned} \mu_{\text{bin}}(n, L; C_{\text{Golay}}) &\leq 2^{-46.386}, \\ \mu_{\text{bin}}(n, L; \text{RM}(2, 5)) &\leq 2^{-47.093}. \end{aligned}} \quad (27)$$

For comparison, the $w > 32$ subtotal is at most $2^{-155.759}$ for Golay and $2^{-158.897}$ for Reed–Muller; the complete bound is controlled by $w = 8$. Repeating the calculation with cutoffs 24 and 32 gives the same displayed complete bounds. The binary first-moment script in the accompanying repository [28] reproduces the calculation. By (5), both implemented chosen-block ensembles therefore exceed a 40-bit one-time sampling-failure target at relative distance 0.10. This certifies the chosen rows themselves; it does not turn the original $\sigma = 32$ row, documented at 20 bits, into a matched 40-bit baseline.

D Additional large-field analysis

D.1 Large-field asymptotic calculations

Sparse supports. Suppose an admissible global support touches t local blocks. In the no-surplus regime,

$$a + b \leq r - 1.$$

The first accumulator output has at least $s - a$ nonzero-state gaps, and after the second stage any final output of weight j satisfies

$$j \geq s - a - b.$$

Using the full projective cancellation budget gives

$$s - a - b \geq s - r + 1 \geq t(d - 1) + 1.$$

A direct fixed- t asymptotic analysis of the exact kernel gives the scale

$$\boxed{P_t = O\left(n^{-t(d-2)-1}\right)} \quad (28)$$

when q is sufficiently large that surplus cancellations are negligible.

In particular,

$$P_1 = O(n^{1-d}).$$

The canonical one-block event gives a matching lower bound of the same polynomial order, so the natural sparse tail is

$$\boxed{\Pr[\text{sparse bad event}] = \Theta(n^{1-d}).} \quad (29)$$

For example:

d	sparse tail scale
2	n^{-1}
3	n^{-2}
5	n^{-4}
7	n^{-6}
9	n^{-8}

For $q - 1 \gtrsim n$, a surplus cancellation can recover at most one geometric power of n while costing $1/(q - 1)$, so surplus layers remain controlled. For $q \approx 2^{128}$ and the lengths considered here, the resulting surplus terms are negligible on the reported scale.

Linear weights. Let C_0 be an $[m, \sigma, d]$ MDS code, with

$$d = m - \sigma + 1, \quad R = \sigma/m = k/n.$$

Its local support-dimension polynomial is

$$\boxed{F_{m,\sigma}(z, u) = 1 + \sum_{\ell=d}^m \binom{m}{\ell} z^\ell u^{\ell-d+1}.} \quad (30)$$

Let $p_0, p_d, \dots, p_m$ be the fractions of local blocks in each exact support class, and put

$$\lambda := \sum_{\ell=d}^m p_\ell.$$

Define the normalized global support and shortening dimension by

$$\alpha := \frac{1}{m} \sum_{\ell=d}^m \ell p_\ell, \quad \beta := \frac{1}{m} \sum_{\ell=d}^m (\ell - d + 1) p_\ell.$$

Since $d - 1 = m - \sigma = m(1 - R)$,

$$\boxed{\beta = \alpha - (1 - R)\lambda.} \quad (31)$$

The support-pattern entropy per output coordinate is

$$\boxed{\mathcal{S}_{m,\sigma}(p) := \frac{1}{m} \left[H(p_0, p_d, \dots, p_m) + \sum_{\ell=d}^m p_\ell \log \binom{m}{\ell} \right].} \quad (32)$$

Here H uses natural logarithms unless a quantity is explicitly divided by $\log 2$.

Normalize the first- and second-stage zero-return counts and intermediate support by

$$x := a/n, \quad y := b/n, \quad \eta := h/n.$$

The first accumulator contributes

$$\begin{aligned} \Gamma_1(\alpha, \eta, x) = & (\alpha - x)H\left(\frac{x}{\alpha - x}\right) + (1 - \eta)H\left(\frac{x}{1 - \eta}\right) \\ & + \eta H\left(\frac{\alpha - x}{\eta}\right) - H(\alpha), \end{aligned} \quad (33)$$

and, at final relative weight ζ , the second contributes

$$\begin{aligned} \Gamma_2(\eta, \zeta, y) = & (\eta - y)H\left(\frac{y}{\eta - y}\right) + (1 - \zeta)H\left(\frac{y}{1 - \zeta}\right) \\ & + \zeta H\left(\frac{\eta - y}{\zeta}\right) - H(\eta). \end{aligned} \quad (34)$$

A direct derivative calculation gives the unconstrained maximizing final weight

$$\zeta_\star = 1 - \frac{y}{\eta}. \quad (35)$$

Therefore define

$$\widehat{\Gamma}_2(\eta, \delta, y) := \Gamma_2\left(\eta, \min\left\{\delta, 1 - \frac{y}{\eta}\right\}, y\right).$$

This distinction matters for target distances exceeding $1/2$.

In the large-field no-surplus region,

$$x + y \leq \beta,$$

and the complete support/trace first-moment exponent is

$$\Phi_{m,\sigma,\delta} = \mathcal{S}_{m,\sigma}(p) + \Gamma_1(\alpha, \eta, x) + \widehat{\Gamma}_2(\eta, \delta, y). \quad (36)$$

For finite q , the corresponding exponent includes

$$\min\{0, \beta \log q - (x + y) \log(q - 1)\}.$$

The exponent tends to zero as the active-block fraction tends to zero, so its unrestricted supremum cannot be negative. The numerical transition estimates below track the zero crossing of a nonzero stationary branch of (36). We write $\Phi_{m,\sigma,\delta}^{\text{int}}$ for the reported value on that branch. A negative branch value alone does not certify a uniform bound. The finite calculation in Appendix H instead bounds an explicitly specified domain away from the sparse boundary and covers the complementary supports separately.

D.2 Worked Reed–Solomon evaluations

Rate 1/2. Take

$$C_0 = \text{RS}[16, 8, 9].$$

Then

$$F_{16,8}(z, u) = 1 + \sum_{\ell=9}^{16} \binom{16}{\ell} z^\ell u^{\ell-8},$$

and

$$\beta = \alpha - \frac{1}{2}\lambda.$$

Numerical optimization of the support/trace exponent gives the transition

$$\boxed{\delta_{16,8}^* \approx 0.305.} \quad (37)$$

At the transition, a representative saddle is approximately

$$\begin{aligned} \lambda &\approx 0.500, & \alpha &\approx 0.448, & \beta &\approx 0.198, \\ x &\approx 0.108, & y &\approx 0.090, & \eta &\approx 0.367. \end{aligned}$$

At this saddle, the available projective dimension is almost exactly exhausted by the zero-return constraints:

$$\boxed{x + y \approx \beta.}$$

At $\delta = 0.30$, the reported interior-branch exponent is approximately

$$\Phi_{16,8,0.30}^{\text{int}} / \log 2 \approx -0.017$$

bits per output coordinate.

The local distance is $d = 9$, so the sparse one-block scale is n^{-8} . At the primary dimension $k = 2^{20}$ and output length $n = 2^{21}$, with $\delta = 0.30$, the marked one-block term is approximately $2^{-137.57}$. That term and the negative linear-weight exponent alone do not control every support size. Theorem 7.1 supplies the complete finite-length bound at this operating point.

Rate 1/4. Take

$$C_0 = \text{RS}[16, 4, 13].$$

The local support–dimension polynomial is

$$\boxed{F_{16,4}(z, u) = 1 + 560z^{13}u + 120z^{14}u^2 + 16z^{15}u^3 + z^{16}u^4.} \quad (38)$$

With

$$\lambda = p_{13} + p_{14} + p_{15} + p_{16},$$

equation (31) becomes

$$\boxed{\beta = \alpha - \frac{3}{4}\lambda.} \quad (39)$$

The larger redundancy sharply reduces the projective cancellation budget.

Numerical optimization of (36) gives the estimated transition

$$\boxed{\delta_{16,4}^* \approx 0.604.} \quad (40)$$

At the transition, a representative saddle is approximately

$$\begin{aligned} \lambda &\approx 0.670, & \alpha &\approx 0.660, & \beta &\approx 0.158, \\ x &\approx 0.065, & y &\approx 0.093, & \eta &\approx 0.666. \end{aligned}$$

Here too, the zero-return constraints nearly exhaust the projective dimension:

$$\boxed{x + y \approx \beta.}$$

At this saddle,

$$1 - \frac{y}{\eta} \approx 0.861 > \delta_{16,4}^*,$$

so the final-weight constraint is active and the maximizing value is $\zeta = \delta$.

Conditional on a block being active, the optimizing local-support distribution is approximately

local support	conditional fraction
13	0.002
14	0.022
15	0.190
16	0.787

Thus the transition is dominated by dense local shortenings rather than minimum-weight words.

In bits per output coordinate, the exponent decomposes approximately as

$$\mathcal{S}_{16,4} \approx +0.132, \quad \Gamma_1 \approx -0.068, \quad \Gamma_2 \approx -0.064,$$

which sum to zero at the transition. This is about 81.3% of the large-field rate-1/4 GV relative distance.

Sparse tail and finite-length margins. Here $d = 13$, so the one-active-block structural tail has scale

$$\boxed{n^{-12}.}$$

The leading marked term is

$$\boxed{\mu_1^{\text{mark}} \sim \frac{13!}{16} \binom{16}{13}^2 \left(\frac{\delta}{1-\delta} \right)^{13} n^{-12}.} \quad (41)$$

At dimension $k = 2^{20}$ and output length $n = 2^{22}$, the numerical estimates are

δ	$-\log_2 \mu_1^{\text{mark}}$
0.550	213.44
0.580	211.15
0.600	209.60
0.604	209.26

The estimated interior-branch margins at this length are

δ	$\Phi_{16,4,\delta}^{\text{int}}/\log 2$	$-n\Phi_{16,4,\delta}^{\text{int}}/\log 2$ (bits)
0.600	-0.017	70828
near 0.604	$[-0.002, -5.15 \times 10^{-5}]$	216–6708

For $k = 2^{20}$ and $n = 2^{22}$, the high-precision point used in the first-moment calculation rounds to

$$\boxed{C_0 = \text{RS}[16, 4, 13], \quad \delta = 0.604} \quad (42)$$

gives a numerical margin of roughly 160 bits. This does not certify that $d_{\min} > 0.604n$. At $\delta = 0.60$, the estimated margin is much larger.

D.3 Repeat–accumulate comparison bounds

For a full finite-length comparison, put $h_0 = rw$. Applying Lemma 5.1 stage by stage and Lemma 5.2 to the w -dimensional shortening of w active repetition blocks gives

$$\begin{aligned} \Pr[d_{\min}(C_{\text{RA}^t}) \leq L] &\leq \mu_{\text{RA}^t}(N, L; r) \\ &:= \sum_{w=1}^{k_0} \binom{k_0}{w} \sum_{\substack{a_1, \dots, a_t \geq 0 \\ h_1, \dots, h_t \geq 1 \\ h_t \leq L}} \chi_q \left(w, \sum_{i=1}^t a_i \right) \prod_{i=1}^t \mathcal{G}_N(h_{i-1}, h_i, a_i). \end{aligned} \quad (43)$$

Infeasible trace tuples contribute zero through $\mathcal{G}_N$. The outer binomial coefficient counts the possible active message supports, and their shortening dimension is w . Thus this is the complete support/trace union bound, not only the weight-one no-return contribution. That leading contribution still explains the sparse orders N^{1-r} for RA^2 and $N^{1-r(t-1)}$ for fixed r, t .

E Structural-container calculations

E.1 Counting structural containers

For fixed permutations, define

$$Z_t := |\{T \subseteq [n] : |T| = t, \kappa(U(T)) < t\}|.$$

By (15), every set counted by Z_t supports a nonzero BAA codeword for every choice of diagonal coefficients.

Endpoint membership changes the size of a difference neighborhood. For $e \in \{0, 1\}$, let $R_n^e(t, r)$ count the t -subsets with r runs and with $e = 1$ exactly

when n belongs to the set. Then

$$\begin{aligned} R_n^0(t, r) &:= \binom{t-1}{r-1} \binom{n-t}{r}, \\ R_n^1(t, r) &:= \binom{t-1}{r-1} \binom{n-t}{r-1}, \\ R_n(t, r) &:= R_n^0(t, r) + R_n^1(t, r) = \binom{t-1}{r-1} \binom{n-t+1}{r}. \end{aligned} \quad (44)$$

For a set of type (t, r, e) , the neighborhood has exact size

$$|\partial T| = t + r - e.$$

For the local block-rank count define

$$B_{m,h}(x, y) := \sum_{\ell=0}^m \binom{m}{\ell} x^\ell y^{\min(\ell, h)}, \quad (45)$$

and

$$A_{n,m,h}(u, c) := \sum_{z \leq c} [x^u y^z] B_{m,h}(x, y)^{n/m}. \quad (46)$$

Theorem E.1 (Exact structural-container expectation). *Assume $m \mid n$, fix $1 \leq t \leq n$, and sample independent $P_1, P_2 \leftarrow \text{Sym}([n])$. For $r_1, r_2 \geq 1$ and $e_1, e_2 \in \{0, 1\}$, set*

$$u_1 := t + r_1 - e_1, \quad u_2 := u_1 + r_2 - e_2.$$

Let $\mathcal{I}_t$ contain the tuples (r_1, r_2, e_1, e_2) that satisfy $r_1 \leq t$, $r_2 \leq u_1$, and $u_1, u_2 \leq n$. Then the expected number of direct structural containers T of size t is

$$\mathbb{E}Z_t = \sum_{(r_1, r_2, e_1, e_2) \in \mathcal{I}_t} R_n^{e_1}(t, r_1) \frac{R_n^{e_2}(u_1, r_2)}{\binom{n}{u_1}} \cdot \frac{A_{n,m,h}(u_2, t-1)}{\binom{n}{u_2}}. \quad (47)$$

Proof. Partition the t -subsets by their first run count and endpoint flag. Equation (44) gives $R_n^{e_1}(t, r_1)$ choices, and the first neighborhood has size u_1 . The permutation P_2 maps this neighborhood to a uniform u_1 -subset. A second application of (44) gives the ratio $R_n^{e_2}(u_1, r_2)/\binom{n}{u_1}$ and a neighborhood of size u_2 . The permutation P_1 maps that neighborhood to a uniform u_2 -subset. Equation (46), divided by $\binom{n}{u_2}$, is exactly the probability that its local block rank is at most $t-1$. Multiplying the factors and summing the disjoint cases proves (47).

Impossible parameter choices contribute zero under the usual binomial convention. Equation (47) counts each final support once and contains no field values or zero-return choices. Omitting e_1, e_2 changes neighborhood sizes by at most two and gives the same entropy exponent, but not the same finite expectation.

E.2 Structural optimization details

The rate-1/2 structural saddle has normalized statistics

$$\begin{aligned} |T|/n &\approx 0.308, & r_1/n &\approx 0.101, \\ |\partial T|/n &\approx 0.409, & r_2/n &\approx 0.120, \\ |U(T)|/n &\approx 0.530, & \kappa(U(T))/n &\approx 0.308. \end{aligned} \tag{48}$$

At rate 1/4 the corresponding values are

$$\begin{aligned} |T|/n &\approx 0.609, & r_1/n &\approx 0.103, \\ |\partial T|/n &\approx 0.712, & r_2/n &\approx 0.070, \\ |U(T)|/n &\approx 0.782, & \kappa(U(T))/n &\approx 0.609. \end{aligned} \tag{49}$$

Moderate-length integer-program experiments on the exact min-cut formulation produce block-saturated witnesses consistent with these profiles. These experiments and all numbers in (17)–(49) are diagnostic rather than theorem statements.

F Implementation and benchmark details

F.1 Benchmark reproduction details

Controlled randomness and timed region. All three implementations use $n = 2,097,024$, the largest multiple of $\text{lcm}(24, 32, 64) = 192$ below 2^{21} , and $k = n/2 = 1,048,512$. Divisibility by 24, 32, and 64 lets the Golay, Reed–Muller, and original random-block constituents tile one common input without truncation. All rows therefore use the same input, two accumulator layers, two independently seeded pseudorandom permutations, one CPU thread, and timed n -to- k operation. The input and permutation seeds are generated before timing from the same deterministic seeds and in the same order.

The original-BAA-style control uses two prefix-sum/permutation layers followed by a dense random systematic local map. The two permutations are sampled independently, and the timed computation evaluates the n -to- k multiplication $x \leftarrow x + Ry$. Chosen-block BAA replaces the random local matrices with the chosen constituent and hence needs no local-matrix stream.

Input generation, permutation-seed derivation, and workspace allocation occur before timing. Both accumulator/permutation layers and the local outer multiplication remain inside it. The original baseline generates its random local matrices during that operation, whereas the chosen constructions evaluate their constituent-code circuits. The benchmark supports the common non-power-of-two length without a stored n -entry permutation table. The mathematical ensemble analysis continues to sample uniform permutations.

For the large-field configuration menu, every row fixes $k = 2^{20}$, $\mathbb{F}_{2^{128}}$, one CPU thread, and the same encode-only boundary, while rate, length, local map or repetition, and accumulator count vary intentionally. The rate-1/2 rows have

$n = 2^{21}$; repetition-3 RA^2 has $n = 3 \cdot 2^{20}$; and RS [16, 4, 13] BAA has rate 1/4 and $n = 2^{22}$. The reported target distances are 0.10 for repetition-2 RA^2 , 0.30 for rate-1/2 BAA, 0.40 for repetition-2 RA^3 and repetition-3 RA^2 , and 0.60 for rate-1/4 BAA. Each lies below that configuration’s estimated trace transition; lower-rate configurations are evaluated at larger distances rather than at the rate-1/2 reference 0.10. Each stage has independent diagonal weighters and interleavers. In every case, permutation and diagonal generation, input generation, and workspace allocation are outside the timed region; every full scan–multiply–scatter layer and the outer transpose multiplication or repetition reduction are inside it.

Platform and procedure. On an Intel Core i9-14900HX with 32 logical processors, we compiled libOTe in Release mode with MSVC 14.44 and C++20 on Windows NT 10.0.26200, with SSE and PCLMUL enabled. We exclude setup, input generation, permutation construction, and workspace allocation; all scan/permutation layers and the outer multiplication are timed. The headline binary construction campaign uses exactly one CPU thread and binds each process to logical processor 2 (Windows affinity mask 0x4). We report the median of six process medians, each over 31 calls after warm-up. Across the six processes, each of the three implementations occupies each benchmark-order position twice. The separate, unpinned two-mode scaling campaigns reverse order halfway. Turbo and thermal state are not controlled; results are host-specific. Each large-field row likewise uses six 31-call process medians after warm-up; paired BAA and RA^t campaigns alternate which implementation runs first.

F.2 CPU control and permutation ablations

For context, the original BAA paper reports 21.4 ms for its aggressive $\sigma = 8$ row and 30.2 ms for its conservative $\sigma = 32$ row at $n = 2^{21}$ on an Intel Core i7-13700H [22]. Table 4 instead uses $n = 2,097,024$, the largest multiple of $\text{lcm}(24, 32, 64)$ below 2^{21} , so that the Golay, Reed–Muller, and random-block constituents all tile the same input without a shortened final block. This common-workload choice is 128 symbols below the original length. In a separate affinity-controlled path ablation, the corrected $\sigma = 32$ control took 31.895 ms at the common length and 31.992 ms at $n = 2^{21}$ under the same on-the-fly permutation implementation; their six-process ranges, 29.147–32.997 and 28.999–33.039 ms, overlap. A precomputed-permutation variant at the common length took 33.688 ms in the same ablation. Thus the 128-symbol length reduction has no measurable penalty once the permutation implementation is held fixed. Table 4’s 29.596 ms control comes from the separate balanced three-way campaign and is used only for its within-campaign ratios. The original paper’s executable, compiler environment, and processor (i7-13700H rather than i9-14900HX) also differ from ours, so hardware labels alone do not make the absolute timings directly comparable.

F.3 Projected PCG throughput

Estimated single-thread PCG throughput. The original BAA evaluation reports about 8 ms of CPU expansion work per batch, in addition to the linear-code encoding, and excludes an approximately 10 ms one-time seed-tree cost from its throughput figure because that cost is amortized across later invocations [22]. Following the same convention, we combine its recurring 8 ms estimate with our one-thread Golay encoding time:

$$\frac{1,048,512}{(22.810 + 8) \text{ ms}} \approx 34.0 \text{ million OTs/s.}$$

This projected rate excludes the one-time 10 ms cost, as does the original paper; it is not an integrated measurement. We do not extrapolate an end-to-end multithreaded rate because the corresponding parallel expansion cost was not measured; the encoding-only scaling results remain below.

G Formalization and proof-audit details

G.1 Formalization details

Representation and proof architecture A length- n word over F is represented as a function $\text{Fin}(n) \rightarrow F$, and a linear code is a submodule of that vector space. Hamming weight uses `mathlib`’s `hammingNorm`. The accumulator, difference, permutation, diagonal, and monomial transformations are linear equivalences. The repeated outer code uses an explicit equivalence between $\text{Fin}(n)$ and $\text{Fin}(n/m) \times \text{Fin}(m)$, which removes hidden divisibility and coordinate-order assumptions.

Ensemble probabilities are represented as ratios of finite cardinalities in $\mathbb{Q}$. Both first-moment arguments are therefore exact finite counts; no measure-theoretic layer is needed. Generic rank is defined over the fraction field of a multivariate polynomial ring. The two diagonal layers use disjoint variable families, so different Cauchy–Binet selections yield different monomials.

The concrete binary codes also have a computation-oriented representation based on fixed-width bit vectors. Conversion lemmas connect these evaluators to the abstract submodules. Consequently, a computed spectrum certifies the code in the corresponding mathematical theorem.

Verified statements The development contains complete proofs of the following finite claims.

- The BAA transformations are invertible and preserve dimension. The block-direct-sum enumerator and the binary identities (1)–(5) are exact finite theorems.
- The extended Golay and $\text{RM}(2, 5)$ constructions have the dimensions, minimum distances, and spectra stated above. The eight-row $[32, 8, 12]$ code has the stated spectrum and lies in a concretely defined extended primitive BCH code.

- Reed–Solomon evaluation codes have the claimed dimension and minimum distance. Their MDS shortening dimensions give (10), (30), (31), (38), and (39).
- The zero-return placement count (6), projective bound (7), master bound (8), and corrected support-excess bound (9) are formal consequences of finite counting and linear algebra.
- The backward parity-check identities (11)–(15), run and block-rank counts (44)–(46), and exact structural-container expectation (47), including the endpoint cases, are proved. The generic rank equals the maximum size of a two-layer vertex-disjoint path packing with the local MDS capacities.

The quadratic-residue rows in Table 1 use exact external spectra and are deliberately outside this list of Lean-verified statements.

The path-packing result is stronger than the neighborhood upper bound used in (15). Its proof first establishes a rectangular Cauchy–Binet identity. It then identifies nonzero minors of the difference matrix with unique self-or-successor linkages. Finally, the MDS parity-check theorem converts selected columns in each local block into the capacity $\min\{|U \cap J|, m - \sigma\}$.

G.2 Formal verification audit and reproduction

Trust boundary. The formalization uses no `sorry`, `admit`, or `project-local` axiom. A dedicated module applies `#print axioms` to the theorems stated in the paper. Those theorems depend only on the standard Lean/mathlib foundations. The audit reports `propext`, `Classical.choice`, and `Quot.sound`. Native decision procedures are used only for labeled small-instance regression tests. No Lean-formalized theorem stated in the paper depends on a generated native-decision axiom. The fixed-code spectra and distances have kernel-checkable proofs.

The command `lake build BAA.AxiomAudit` checks the complete development and prints the dependency audit. Exhaustive regressions cover the binary accumulator and difference-linkage conventions through length eight and the nonbinary placement formula through length seven. These tests supplement the symbolic all-length theorems; they do not replace them.

The Lean development does not formalize the quadratic-residue constituents or their published spectra. It also does not claim the one-accumulator asymptotic limit, (18)–(24), (28)–(29), (32)–(36), any numerical transition, the $O(n)$ cost model, or concentration of structural containers. These claims require external coding-theory results, analytic estimates, certified optimization, a computational cost model, or new probabilistic arguments. The RA^t parameter selection and configuration measurements are likewise outside the Lean development. Theorem 7.1 additionally uses the analytic reductions in Appendix H and the accompanying rational/Arb verifier [28]; those components are not Lean-formalized. The RA^t finite sums have not received a complete numerical certificate.

H Complete finite-length RS certificate

This appendix proves the finite reductions used in Theorem 7.1 and specifies their executable verification. Fix $m = 16$, $\sigma \in \{4, 8\}$, $h_0 = m - \sigma$, $d = h_0 + 1$, and $n = mB = mk/\sigma$, where $k = 2^{20}$. Set $\delta = 3/5$ for $\sigma = 4$ and $\delta = 3/10$ for $\sigma = 8$, and put $L = \lfloor \delta n \rfloor$. All logarithms in the proof are natural. We use $Q = 2^{127} - 1$ as a lower bound on $q - 1$.

An exact nonzero outer support meeting t blocks has size s and shortening dimension r satisfying

$$dt \leq s \leq mt, \quad r = s - h_0 t, \quad 1 \leq r \leq \sigma t.$$

Its multiplicity is

$$N_{t,s} = \binom{B}{t} [z^s] \left(\sum_{\ell=d}^m \binom{m}{\ell} z^\ell \right)^t. \quad (50)$$

Every listed RS support is possible: a polynomial vanishing at its complement has degree $m - \ell \leq \sigma - 1$ and is nonzero on the support. Supports containing a nonempty local intersection smaller than d have empty exact-support events. We omit them before applying the union bound. Write $\hat{\mu}$ for this restriction of the master sum (8). It still bounds the failure probability.

H.1 A uniform projective envelope

For $q - 1 \geq 2n$ and $r \leq n$, writing $Q' = q - 1$ gives

$$\frac{q^r - 1}{q - 1} \leq (Q')^{r-1} (1 + 1/Q')^r \leq \frac{(Q')^{r-1}}{1 - r/Q'} \leq 2(Q')^{r-1}.$$

The middle inequality follows by bounding each binomial coefficient $\binom{r}{j} \leq r^j$ and extending the resulting geometric series. Consequently, for all $q \geq 2^{127}$,

$$\chi_q(r, c) \leq 2 \min\{1, Q^{r-1-c}\}. \quad (51)$$

Indeed, $\min\{1, u^e\}$ is constant for $e \geq 0$ and decreases with $u \geq 1$ for $e < 0$. For direct evaluation we use the sharper fact that $\chi_q(r, c) = 1$ for $c < r$. For $c \geq r$, its value is

$$\sum_{j=1}^r \binom{r}{j} (q-1)^{j-1-c},$$

which decreases with q . Thus evaluation at $q = 2^{127}$ supplies a uniform upper bound, even when that integer is only a lower bound on the actual field size.

H.2 An envelope for all sparse supports

Put $T = \lfloor n/1000 \rfloor$ and choose $z = 1/10$ for $\sigma = 4$ and $z = 1/5$ for $\sigma = 8$. Define

$$\rho = \delta(1+z)^2, \quad D = 1 - \delta - mT/n, \quad V = (\delta D z^2)^{-1}, \quad K_s = \frac{s}{n-s+1} \frac{\rho}{1-\rho}.$$

The verifier checks $0 < \rho < 1$, $D > 0$, $K_s < 1$, $K_s Q > 1$, and $V < Q$ using the extrema over $d \leq s \leq mT$.

First suppose $b \leq s$. Summing the last binomial in $\mathcal{G}_n(H, j, b)$ by the hockey-stick identity gives

$$\sum_{j=1}^L \mathcal{G}_n(H, j, b) \leq \frac{\binom{H-b}{b} \binom{n}{b} \binom{L}{H-b}}{\binom{n}{H}}.$$

A nonzero term requires $H \leq L + b \leq L + s$. The inequalities

$$\frac{\binom{L}{H-b}}{\binom{n}{H-b}} \leq \delta^{H-b}, \quad \frac{\binom{n}{b} \binom{n}{H-b}}{\binom{n}{H}} \leq \binom{H}{b} \left(\frac{n}{n-H+1} \right)^b$$

therefore imply

$$\sum_{j=1}^L \mathcal{G}_n(H, j, b) \leq \binom{H}{b}^2 D^{-b} \delta^{H-b} \leq \rho^H V^b. \quad (52)$$

The last inequality uses $\binom{H}{b} \leq (1+z)^H / z^b$. For the first stage,

$$\mathcal{G}_n(s, H, a) \leq 2^s \left(\frac{s}{n-s+1} \right)^{s-a} \binom{H-1}{s-a-1}.$$

This follows from $\binom{s-a}{a} \leq 2^s$, $\binom{n-H}{a} \leq \binom{n}{a}$, and $\binom{n}{a} / \binom{n}{s} \leq [s/(n-s+1)]^{s-a}$. Extending the positive series over H and using

$$\sum_{H \geq u} \binom{H-1}{u-1} \rho^H = \left(\frac{\rho}{1-\rho} \right)^u$$

now bounds the two-stage placement sum by $2^s K_s^{s-a} V^b$.

To include all cancellations, write $K = K_s$. For $b < r$ set $e = a + b - r + 1$. Then

$$K^{s-a} V^b = K^{h_0 t + 1} (KV)^b K^{-e}.$$

Apply (51) and extend e to all integers. The sums for $e \leq 0$ and $e \geq 1$ are $1/(1-K)$ and $1/(KQ-1)$. For $r \leq b \leq s$, every term has surplus cancellations; extending both geometric sums in a and $b-r$ gives the following bound for all $b \leq s$:

$$2^{s+1} \left[K^{h_0 t + 1} \left(\frac{1}{1-K} + \frac{1}{KQ-1} \right) \sum_{b=0}^{r-1} (KV)^b + \frac{K^{h_0 t} (KV)^r}{Q(1-1/(KQ))(1-V/Q)} \right]. \quad (53)$$

For $b > s$, discard the output-weight restriction. Vandermonde's identity gives

$$\sum_{j=1}^n \mathcal{G}_n(H, j, b) = \binom{H-b}{b} \leq \binom{n}{b}, \quad \sum_{H=1}^n \mathcal{G}_n(s, H, a) = \binom{s-a}{a}.$$

Since $\sum_a \binom{s-a}{a} \leq 2^s$, dropping the additional factor Q^{-a} and using $\binom{n}{b} \leq n^b$ yields the complete tail bound

$$\frac{2^{s+1}n^{s+1}Q^{-h_0t-2}}{1-n/Q}. \quad (54)$$

To aggregate (53), set

$$\begin{aligned} U_0 &= \sum_{\ell=d}^m \binom{m}{\ell}, & M_0 &= 2^m U_0, \\ C &= \frac{mn}{n-mT+1} \frac{\rho}{1-\rho}, & K_{\max} &= CT/n, \\ K_{\min} &= \frac{d}{n-d+1} \frac{\rho}{1-\rho}, & W &= \max\{1, K_{\max}V\}, \\ J_0 &= \frac{3}{m} M_0 C^{h_0} W^\sigma, & x(t) &= J_0(t/n)^{h_0-1}. \end{aligned}$$

There are at most $\binom{B}{t} U_0^t \leq (3B/t)^t U_0^t$ supports touching t blocks; the last inequality uses $e < 3$. Also $K_s \leq Ct/n$ and $\sum_{b < r} (K_s V)^b \leq \sigma t W^{\sigma t}$. Hence their combined $b \leq s$ contribution is at most

$$(c_2 t^2 + c_0) x(t)^t, \quad (55)$$

where

$$c_2 = \frac{2C\sigma}{n} \left(\frac{1}{1-K_{\max}} + \frac{1}{K_{\min}Q-1} \right), \quad c_0 = \frac{2}{Q(1-1/(K_{\min}Q))(1-V/Q)}.$$

Use the disjoint intervals $[2, 3]$, $[4, 7]$, $[8, 15]$, $[16, 31]$, $[32, 63]$, and $[64, T]$. On $[A, E]$, replace $x(t)$ by $x(E) < 1$ and extend the sum to all $t \geq A$. The resulting geometric sums are rational:

$$\begin{aligned} \sum_{t \geq A} x^t &= \frac{x^A}{1-x}, \\ \sum_{t \geq A} t^2 x^t &= x^A \left(\frac{A^2}{1-x} + \frac{2Ax}{(1-x)^2} + \frac{x(1+x)}{(1-x)^3} \right). \end{aligned}$$

This covers every $2 \leq t \leq T$, including growing sublinear block counts. Finally, summing (54) over all $1 \leq t \leq T$ gives at most

$$\frac{2n}{Q^2(1-n/Q)} \frac{Y}{1-Y}, \quad Y = \frac{3BM_0 n^m}{Q^{h_0}} < 1. \quad (56)$$

Here $s \leq mt$ and $1/t \leq 1$ justify each relaxation. The verifier checks all positivity and convergence conditions by exact rational comparisons.

H.3 Direct one-block terms and their intermediate tail

For $t = 1$, we retain the exact multiplicities (50). For $b < r$, the final-weight sum has the finite identity

$$\sum_{j=1}^L \mathcal{G}_n(H, j, b) = \frac{\binom{H-b}{b}}{\binom{n}{H}} \sum_{v=0}^b \binom{L}{H-v} \binom{n-L}{v}. \quad (57)$$

Indeed, an H -subset has its $(H-b)$ th order statistic at most L exactly when it has at most b elements above L . Counting first by the order statistic and then by that number of elements proves the identity. We evaluate every $H \leq 512$, $a \leq \lfloor s/2 \rfloor$, and $b < r$ using (57) and the exact first kernel.

For the remaining H , put $\delta_0 = L/n$ and $D_s = 1 - \delta_0 - s/n$. Keeping the first binomial coefficient in the preceding envelope gives

$$\mathcal{G}_n(s, H, a) \sum_{j \leq L} \mathcal{G}_n(H, j, b) \leq A_{s,a,b} \binom{H-1}{s-a-1} \binom{H}{2b} \delta_0^H,$$

where

$$A_{s,a,b} = \frac{\binom{s-a}{a} \binom{n}{a}}{\binom{n}{s}} \frac{\binom{2b}{b}}{(D_s \delta_0)^b}.$$

We used $\binom{H-b}{b} \binom{H}{b} = \binom{2b}{b} \binom{H}{2b}$. For an integer $w \geq 1$ and $0 < u < 1$, define the rational function

$$R_{w,b}(u) = \sum_{i=0}^{\min(w, 2b)} \frac{w}{w+2b-i} \frac{(w+2b-i)!}{i!(w-i)!(2b-i)!} \left(\frac{u}{1-u} \right)^{w+2b-i}. \quad (58)$$

It equals $\sum_{H \geq 1} \binom{H-1}{w-1} \binom{H}{2b} u^H$. To prove this, use $\binom{H-1}{w-1} = w \binom{H}{w} / H$, count a pair of subsets by its intersection size i , and sum the resulting union size $N = w+2b-i$ with $\binom{H}{N} / H = \binom{H-1}{N-1} / N$. Taking $u = (1 + \delta_0)/2$ bounds every omitted $H > 512$ by

$$A_{s,a,b} (\delta_0/u)^{513} R_{s-a,b}(u). \quad (59)$$

The script multiplies this expression by the projective factor and sums every a and $b < r$. For $r \leq b \leq s$, it uses the second term of (53), with the support's own D_s and K_s . Equation (56) already covers $b > s$.

H.4 Every remaining support type

For $t > T$, let $k_0, k_d, \dots, k_m$ count blocks of each local support size, and set $p_\ell = k_\ell/B$. The number of supports of that type is at most $\exp(n\mathcal{S}_{m,\sigma}(p))$ by the multinomial entropy inequality. Normalize s, H, a, b, j by n , writing $\alpha, \eta, x, y, \zeta$, and put $\beta = \sum_{\ell=d}^m (\ell - h_0) p_\ell / m$. Define $f(A, B) = AH(B/A)$ on $0 \leq B \leq A$, with $f(0, 0) = 0$. Let J_1 be the sum of

$$\begin{aligned} f(\alpha - x, x), & \quad f(1 - \eta, x), & \quad f(\eta, \alpha - x), \\ f(\eta - y, y), & \quad f(1 - \zeta, y), & \quad f(\zeta, \eta - y). \end{aligned}$$

The finite exponent used by the verifier is

$$F = \mathcal{S}_{m,\sigma}(p) + J_1 - H(\alpha) - H(\eta) - \log Q \max\{0, x + y - \beta\}.$$

The feasible region requires that p be a probability vector, $\alpha = \sum \ell p_\ell / m$, $p_0 \leq 1 - m(T+1)/n$, $0 \leq \zeta \leq \delta$, and that all six perspective terms have $0 \leq B \leq A$.

Use $\binom{v}{u} \leq \exp(vH(u/v))$ in the six numerators, after replacing $\binom{H-1}{s-a-1}$ by $\binom{H}{s-a}$ and its second-stage analogue. The lower bound $\binom{n}{u} \geq \exp(nH(u/n))/(n+1)$ controls both denominators. Equation (51), with its extra minus one discarded, therefore bounds each support-type/trace contribution by $2(n+1)^2 \exp(nF)$. There are at most $(B+1)^\sigma$ support types and $(n+1)^4$ trace tuples. Thus any uniform upper bound $F \leq U$ gives

$$\hat{\mu}_{t>T} \leq 2(B+1)^\sigma (n+1)^6 \exp(nU). \quad (60)$$

Only $-H(\alpha)$ and $-H(\eta)$ are nonconcave terms of F . The verifier covers

$$[d(T+1)/n, 1] \times [d(T+1)/(2n), 1]$$

by rectangles with rational boundaries, and replaces these two terms by their upper secants. Every other entropy term has a global linear upper bound. For any $0 < \theta < 1$ and strictly positive probability vector p^* ,

$$f(A, B) \leq -B \log \theta - (A - B) \log(1 - \theta), \quad H(p) \leq -\sum_\ell p_\ell \log p_\ell^*.$$

Introduce $c \geq 0$, $c \geq x + y - \beta$, with objective term $-c \log Q$. The choice $c = \max\{0, x + y - \beta\}$ lies in $[0, 1]$, since $x, y \leq 1/2$. Each rectangle consequently has a linear upper objective $c_* + v_* \cdot v$ over rational constraints $Av \leq b$, $Ev = e$, and $l \leq v \leq u$. For any rational multipliers $\mu \geq 0$ and ν , put $r_* = v_* - A^T \mu - E^T \nu$. Then

$$F \leq c_* + \mu \cdot b + \nu \cdot e + \sum_i ((r_*)_i l_i + \max\{0, (r_*)_i\} (u_i - l_i)). \quad (61)$$

Arb intervals enclose the logarithms and residuals in this expression. The numerical optimizer only proposes p^* , θ , and multipliers; their feasibility or optimality is not assumed. The verifier normalizes p^* exactly and checks all required signs.

The covering is encoded as a binary tree of exact midpoint splits. The verifier reconstructs every boundary and rejects missing or extraneous nodes. An infeasible leaf is accepted only if an exact necessary condition fails: $\alpha_{\text{upper}} \geq d(T+1)/n$, $\alpha_{\text{lower}} \leq 2\eta_{\text{upper}}$, or $\eta_{\text{lower}} \leq 2\delta$. The saved coverings have 21 and 19 leaves for $\sigma = 4$ and $\sigma = 8$, respectively.

H.5 Numerical result and reproduction

Table 6 lists separate bounds for every region. The complete verifier adds their interval enclosures before checking the reported failure bound. It obtains

$$\log_2 \hat{\mu} < -209.6008601171 \quad (\sigma = 4), \quad \log_2 \hat{\mu} < -137.5682486336 \quad (\sigma = 8).$$

Region	$\sigma = 4$	$\sigma = 8$
$t = 1, H \leq 512, b < r$	209.6008	137.5682
$t = 1, H > 512, b < r$	404	690
$t = 1, r \leq b \leq s$	296	243
$2 \leq t \leq T, b \leq s$	250	159
$1 \leq t \leq T, b > s$	1357	862
$t > T$, all traces	1078	5634
Complete bound	209.60	137.56

Table 6. Certified lower bounds on $-\log_2$ of each contribution. Every region includes all feasible first return counts and final weights. The lengths and distances are those of Theorem 7.1.

These inequalities imply Theorem 7.1.

The exact directly summed terms with $t = 1$, $H \leq 512$, and $a + b < r$ also give a lower bound on $\hat{\mu}$, because their projective factor is one. The ratio of the complete upper bound to this partial sum is less than $1 + 5 \cdot 10^{-13}$ for $\sigma = 4$ and $1 + 3.3 \cdot 10^{-7}$ for $\sigma = 8$. Further numerical tightening of this same sum can gain less than a millionth of a bit. This comparison does not lower-bound the true failure probability, since $\hat{\mu}$ is itself a union bound.

The certificate artifact is provided in the accompanying repository [28] and uses Python with `python-flint` 0.9.0. From the repository root, the complete verification commands are

```
python -m pip install -r rs-certification/requirements.txt
python rs-certification/verify_complete.py
```

The certificate directory contains both rational certificate proposals. Verification needs neither NumPy nor SciPy. Optional generation and small-instance tests have separate dependencies and commands in that directory’s README. Tests check the order-statistic identity, generating series, intermediate tail, finite prefactors, and rejection of malformed coverings. These checks support the written reductions; the new analytic and numerical certificate is not a Lean proof.