

SPIN: Fast Codes for Correlation Generation and Polynomial Commitments

Stanislav Peceny¹, Rahul Rachuri², Srinivasan Raghuraman², and
Peter Rindal³

¹ Stealth Software Technologies, Inc.

² Visa Research

³ Category Labs

Abstract. Binary codes used in correlation generation and proof systems need strong distance, fast ordinary and transposed encoding, and regular memory access. We introduce Single-Permutation INTERLEAVED (SPIN) codes, combining a blockwise outer encoder, one global interleaver, and a recursive inner.

Our main result is a Structured SPIN family with rate $1/2$ and $O(N)$ bit operations for ordinary and transposed encoding at its native output lengths N . With probability $1 - o(1)$ over the random choice of the code, its minimum distance exceeds $0.11N$. The proof combines outer spectrum bounds with an analysis of structured support spreading and recursive state mixing. We formally verify these distance and rate guarantees in Lean.

For finite lengths, a BCH-derived outer constituent and a fixed recursive inner give rate $1/2$ and relative distance greater than 0.10 , with setup-failure probability below 2^{-40} . We certify this bound at message lengths 2^{16} , 2^{18} , 2^{20} , 2^{22} , and 2^{24} using rigorous spectrum inequalities, without assuming the constituent’s exact weight spectrum.

For pseudorandom correlation generators (PCGs), SPIN’s transposed encoder computes 2^{20} 128-bit correlation blocks in about 9.3 ms on one Ryzen 9 7950X thread after precomputation. This is approximately $3.4\times$ faster than the original rate- $1/2$ BAA codes of Kolesnikov et al. (CRYPTO 2026) [20]. Ordinary encoding has similar cost. With a precomputed code, libOTe’s regular-noise Silent OT sender achieves 89.0 million hashed OTs per second on one core, in batches of 2^{18} . Timings exclude initial setup, base-correlation generation, and network transport.

We also use SPIN in a Brakedown-style polynomial commitment scheme and integrate it into Flock. Commitment and one opening of 512 MiB take 509 ms on one core, with a 100-bit interactive fixed-matrix testing target; this excludes code-setup failure, extraction, and Fiat-Shamir reductions. Within an optimized Flock implementation, replacing Ligerito with SPIN-Brakedown improves prover throughput by $1.63\text{--}2.29\times$ at the measured workloads, with larger proofs and slower verification. The comparison gives both backends the applicable shared implementation optimizations.⁴

⁴ Generative AI tools were used extensively in this work. See Section 10 for details.

Table of Contents

1	Introduction	4
1.1	Results and Approach	4
1.2	Related Work	5
2	Preliminaries	6
2.1	Notation	6
2.2	Linear Encoders and Codes	7
2.3	Sampled Families and Asymptotics	7
3	A Common First-Moment Framework	7
4	Building Intuition: Accumulation and Random Mixing	9
5	Structured SPIN	11
5.1	From Random Mixing to Structured Mixing	11
5.2	Construction at a Glance	12
5.3	Block-Diagonal Outer	12
5.4	Structured Interleaver	13
5.5	Structured Recursive Inner	14
5.6	Composition and Rate	15
6	Asymptotic Distance and Linear Work	15
6.1	Scalable Constituent Family	15
6.2	Selecting the Shared Outer	16
6.3	From Row Weights to an iid Comparison	17
6.4	IMT Moments at Positive Occupation	19
6.5	Sparse Occupation without a Large Conditioning Loss	20
6.6	Fixed Occupation: Mixing between Isolated Impulses	20
6.7	Completing the Argument	21
7	Finite-Length Structured SPIN	22
7.1	The Selected Construction	22
7.2	Finite Distance Guarantee	23
7.3	A Rate-1/4 Operating Point	24
8	Encoder Implementation and Performance	24
9	Applications	26
9.1	Pseudorandom Correlation Generation	26
9.2	Polynomial Commitments	26
9.3	Flock	28
10	Disclosure of Generative AI Use	28
A	Supporting Constructions and Their Proofs	31
A.1	Uniform Interleaving and Random-Block Outers	31
A.2	Accumulator SPIN	33
A.3	Random SPIN	36
A.4	The Bazzi–Mahdian–Spielman Question	41
B	Asymptotic Structured SPIN Certificate	42
B.1	Certificate for the Scalable Family	42
C	Finite BCH-256 Certificate	54
C.1	The Fixed Outer and Its Spectrum	54
C.2	Selected IMT Maps and Positive Transfers	56

C.3	One Active Outer Row	57
C.4	Sparse Occupations and Constant-Row Compositions	57
C.5	Dense Occupations by a Change of Input Law	58
C.6	The Quarter-Rate Outer and Feedback	61
C.7	Complete Finite Margins	63
C.8	Certificate Completion	63
D	Parameter Selection and Engineering Curves	64
D.1	How the Parameters Affect the Margin	64
D.2	The Certified Engineering Curve	69
D.3	Length, Work, and Parameter Selection	70
E	Encoder Implementation and Benchmark Details	72
E.1	Ordinary Encoding	73
E.2	Comparison with Other Transposed Encoders	74
F	Application Details and Additional Comparisons	75
F.1	Silent OT Integration	75
F.2	Polynomial Commitment Details	78
F.3	Flock Integration and Full Results	82

1 Introduction

Binary encoding is a major cost in correlation generators and code-based proof systems. Pseudorandom correlation generators (PCGs) apply public generator matrices to expand sparse correlations into OT and VOLE correlations [5,8]. Code-based proof systems encode rows of large arrays, with distance supporting soundness [16,10]. These applications need constant rate and linear distance, but also fast encoding, efficient transposed encoding, and regular memory access.

Random linear codes provide strong distance but require dense computation under direct encoding. Structure can reduce arithmetic and data movement; it also introduces dependencies that complicate distance proofs. We introduce *Single-Permutation INterleaved* (SPIN) codes to combine quantitative distance guarantees with efficient evaluation in both directions.

For an admissible output length N , a SPIN encoder has the form

$$E_N := I_N \circ \Pi_N \circ O_N. \quad (1)$$

The outer O_N maps the message to N intermediate bits, Π_N permutes these coordinates, and the recursive linear inner I_N preserves length. The outer gives sparse messages an initial weight floor; the interleaver spreads the resulting support before recursive encoding. The phrase *single permutation* describes this interface: block shuffles, a transpose, and region shuffles may implement Π_N , but their composition is one global interleaver.

1.1 Results and Approach

Distance and Encoding Work. Every realization of our main construction, Structured SPIN, has rate $1/2$. Its minimum distance exceeds $0.11N$ with probability $1 - o(1)$ over setup. Ordinary and transposed encoding require $O(N)$ bit operations at the family’s native lengths. The distance and rate claims are formally verified in Lean, including the construction’s random setup, the analytic estimates, and the required numerical certificates. Appendix B.1 describes the formalization. For finite lengths, we choose constituents for concrete performance rather than require the asymptotic outer. A fixed BCH-derived $[256, 128, \geq 38]$ constituent gives relative distance greater than 0.10 with setup-failure probability below 2^{-40} at message lengths $2^{16}, 2^{18}, 2^{20}, 2^{22}, 2^{24}$. The certificate uses rigorous spectrum inequalities, not an assumed exact BCH spectrum. Section 7.3 also gives two certified rate- $1/4$ operating points.

Analysis and Structure. For this class of encoder, sparse messages typically cause low minimum distance. The encoder must spread their nonzero bits widely without dense matrix multiplication. First, the outer encodes short message blocks independently, so even a block with a single nonzero bit produces several nonzero bits. Next, the permutation spreads the bits of each encoded block across distinct regions. Finally, the inner processes this stream with a small running state. Cheap XOR updates let an input bit influence many later outputs, spreading

local changes over a long codeword. Our proof establishes the distance guarantees above by bounding the probability that any nonzero message produces too few nonzero output bits. Our contribution is this efficient combination and its distance analysis; its ingredients build on established ideas from concatenated codes.

Section 4 develops the idea through two simpler supporting constructions: Accumulator SPIN uses a running sum as its inner, and Random SPIN illustrates what stronger random mixing can achieve. Structured SPIN matches Random SPIN’s 11% distance guarantee while reducing the direct encoding work from $\Theta(N \log N)$ to $O(N)$. The complete proofs for these two supporting constructions are in Appendix A. The minimal accumulator instance also answers the logarithmic-outer-memory question of Bazzi, Mahdian, and Spielman [1] in the terminated-encoder interpretation of Appendix A.4.

Implementation and Applications. On one Ryzen 9 7950X thread, SPIN’s transposed encoder computes 2^{20} 128-bit correlation blocks in about 9.3 ms after precomputation, approximately $3.4\times$ faster than the original block–accumulate–accumulate (BAA) codes [20]. Ordinary encoding has similar cost. Our same-host encoder comparison also includes newer chosen-block BAA, whose fastest measured configuration takes about 24 ms. In libOTe’s regular-noise Silent OT path, the sender produces 2^{18} hashed OTs in 2.944 ms, or 89.0 million OTs per second on one core. This experiment reuses the full precomputed SPIN code and does not refresh its permutations between batches; such refresh is a heuristic outside our distance certificates. It measures local computation with supplied base correlations, excluding initial setup and network transport; Section 9.1 specifies the timing scope.

As application evidence, we substitute SPIN into a Brakedown-style PCS [16] and integrate it into Flock [11]. Standalone commitment and one opening of 512 MiB take 509 ms on one core. Replacing Ligerito [21] within Flock improves throughput by $1.63\text{--}2.29\times$ at the measured workloads, with larger proofs and slower verification. Both backends receive the applicable shared optimizations. Sections 9.2 and 9.3 state the security targets, what they exclude, and the comparison scope.

1.2 Related Work

Serial Concatenation. Turbo codes use randomized interleaving [3]. Benedetto et al. analyzed serial concatenation with a uniform interleaver [2]. Repeat–accumulate codes and their enumerators [14], distance analyses of parallel and serial concatenations [18], and near-GV repeat-multiple-accumulate ensembles [19] form the coding-theoretic background. Bazzi, Mahdian, and Spielman give limitations for depth-two constant-memory concatenations and positive results with two accumulators [1]. Block-Accumulate codes use random block-diagonal maps, detailed expected spectra, and optimized CPU/GPU encoders [20]. Accumulator SPIN truncates BAA to a single accumulator stage. Chosen-block BAA reuses

a spectrum-controlled constituent [22]; Structured SPIN combines that shared design idea with a structured interleaver and a different recursive inner.

Correlation Generation. PCGs and VOLE use linear codes [5,8,7,27,9]. Silver uses structured LDPC matrices and a banded triangular solve [12]; its heuristic linear-distance conjecture was later refuted [23]. Expand–Accumulate combines a sparse expander and accumulator [6]. Expand–Convolute uses random time-varying convolution and proves distance through coupling and concentration [23]; its faster empirical parameter choices go beyond those guarantees. Random SPIN uses that inner but replaces the expander with independent random-block injections and uses an exact weight-enumerator transform. Structured SPIN instead combines a spectrum-controlled outer, structured interleaving, and a recursive inner to prove strong distance with linear encoding work.

Proof Systems. Linear encoding is a first-order cost in constant-overhead arguments [24]. Orion uses code switching [26]; Brakedown gives a field-agnostic SNARK from linear-time encodable codes [16]. Other constructions use Expand–Accumulate [4] or interleaved RAA with improved binary distance bounds [10]. SPIN could also instantiate Blaze’s generic code-switching framework, which gives asymptotically smaller proofs than our Brakedown-style PCS. Bolt uses sketched codes to reduce commitment cost [17].

Organization and Artifact. Sections 2–3 give the common framework; Section 4 builds intuition before the structured construction, asymptotic proof, and finite certificates in Sections 5–7. Sections 8 and 9 evaluate the encoder and applications. The appendices contain the complete supporting proofs, parameter studies, and detailed experiments. The source repository [25] includes the core SPIN encoder implementation and build instructions. The accompanying Lean development includes proof sources, numerical certificates, and reproduction instructions.

2 Preliminaries

2.1 Notation

For an integer $n \geq 0$, let $[n] := \{1, \dots, n\}$, with $[0] := \emptyset$. In the coding sections, all vector spaces are over $\mathbb{F}_2$. The applications explicitly introduce an extension field. Unless stated otherwise, we write vectors as rows. For $u \in \mathbb{F}_2^n$, let $\text{wt}(u)$ be its Hamming weight and let $\text{supp}(u) := \{i \in [n] : u_i \neq 0\}$ be its support. Let S_n denote the symmetric group on $[n]$; a permutation acts on vectors by permuting their coordinates.

If S is a finite set, then $X \leftarrow S$ denotes a uniform sample from S . If $\mathcal{D}$ is a distribution, then $X \leftarrow \mathcal{D}$ denotes a sample from $\mathcal{D}$. Separate sampling statements use independent randomness unless a joint distribution is declared.

For $q \in [0, 1]$, define the binary entropy function $H_2(q) := -q \log_2 q - (1 - q) \log_2 (1 - q)$, where $0 \log_2 0 := 0$. For integers a, b , we set $\binom{a}{b} := 0$ unless $0 \leq b \leq a$. These conventions shorten the enumerator formulas later.

2.2 Linear Encoders and Codes

Fix integers $1 \leq K \leq N$. A binary linear encoder is a map $E : \mathbb{F}_2^K \rightarrow \mathbb{F}_2^N$. Under the row-vector convention, there is a unique generator matrix $G_E \in \mathbb{F}_2^{K \times N}$ satisfying $E(x) = xG_E$. The associated linear code is the image $\mathcal{C}(E) := E(\mathbb{F}_2^K) \subseteq \mathbb{F}_2^N$. Its rate is $\dim(\mathcal{C}(E))/N$. In particular, an injective encoder has rate K/N .

For a nonzero linear code $\mathcal{C} \subseteq \mathbb{F}_2^N$, define

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

By convention, set $d_{\min}(\{0\}) := +\infty$, and write $d_{\min}(E) := d_{\min}(\mathcal{C}(E))$. The relative distance is $d_{\min}(\mathcal{C})/N$.

We keep injectivity separate from the distance of the image. A noninjective map can have an image of positive minimum distance, while a nonzero kernel message encodes to zero. Accordingly, a claim that a K -to- N encoder has rate K/N and relative distance δ includes injectivity and the bound $d_{\min}(E) \geq \delta N$.

The transposed encoder is the linear map

$$E^\top : \mathbb{F}_2^N \rightarrow \mathbb{F}_2^K, \quad E^\top(q) := qG_E^\top.$$

It satisfies $\langle E(x), q \rangle = \langle x, E^\top(q) \rangle$. Ordinary encoding work refers to evaluating E ; transposed work refers to evaluating $E^\top$. Unless stated otherwise, work bounds count bit operations. We discuss long-range data movement separately because the bit-operation count does not capture it.

2.3 Sampled Families and Asymptotics

A sampled encoder is specified together with the distribution from which setup draws its components. After setup, the realized linear map and its image code are fixed. Expectations and probabilities refer to the declared setup distribution unless a subscript specifies a smaller probability space.

For a family indexed by native lengths $N_m \rightarrow \infty$, the notation $o(1)$ refers to the limit $m \rightarrow \infty$. An event holds *with high probability* if its failure probability is $o(1)$ over setup. Other output lengths are obtained by zero extension of the largest smaller native member (Section D.3).

3 A Common First-Moment Framework

This section isolates the calculation shared by all SPIN families. We first define the complete sampled encoder. We then count low-weight outputs of nonzero messages.

Fix lengths $K, N \geq 1$. The construction independently samples an outer encoder $O : \mathbb{F}_2^K \rightarrow \mathbb{F}_2^N$, a permutation Π of $[N]$, and a linear inner encoder $I : \mathbb{F}_2^N \rightarrow \mathbb{F}_2^N$. A fixed component is sampled from a point mass. The realized SPIN encoder is $E := I \circ \Pi \circ O$.

Fix a finite class set $\mathcal{T}$ and a class map $c : \mathbb{F}_2^N \rightarrow \mathcal{T}$. The class map records the information about an outer word that remains relevant after the outer stage.

For a fixed outer word $u \in \mathbb{F}_2^N$, define its conditional failure probability

$$q_d(u) := \mathbb{P}_{\Pi, I}[\text{wt}(I(\Pi(u))) \leq d]. \quad (2)$$

The probability is over the interleaver and inner setup; the word u is fixed. For every class $\tau \in \mathcal{T}$, let $Q_\tau(d)$ be any *failure envelope*, that is, an upper bound satisfying

$$q_d(u) \leq Q_\tau(d) \quad \text{for every } u \text{ with } c(u) = \tau. \quad (3)$$

Set $Q_\tau(d) := 0$ when the class has no preimage. The class need not determine the exact failure probability. It need only expose enough information to prove the uniform bound in (3).

For $\tau \in \mathcal{T}$, define the expected outer multiplicity

$$A_\tau^{\text{out}} := \mathbb{E}_O \left[\sum_{x \in \mathbb{F}_2^K \setminus \{0\}} \mathbf{1}\{c(O(x)) = \tau\} \right]. \quad (4)$$

Theorem 1 (Class-indexed first-moment envelope). *For $d \in \{0, \dots, N\}$, define*

$$Z_d := |\{x \in \mathbb{F}_2^K \setminus \{0\} : \text{wt}(E(x)) \leq d\}|.$$

Then

$$\begin{aligned} \mathbb{P}[E \text{ is not injective or } d_{\min}(E) \leq d] &\leq \mathbb{E}[Z_d] = \mathbb{E}_O \left[\sum_{x \in \mathbb{F}_2^K \setminus \{0\}} q_d(O(x)) \right] \\ &\leq \sum_{\tau \in \mathcal{T}} A_\tau^{\text{out}} Q_\tau(d). \end{aligned} \quad (5)$$

If q_d is constant on every class and $Q_\tau(d)$ equals that constant, then the final inequality is an equality.

Proof. The bad event in (5) occurs exactly when $Z_d \geq 1$. Indeed, a nonzero kernel message contributes output weight zero, and every nonzero image word of weight at most d has a nonzero preimage. Markov's inequality gives $\mathbb{P}[Z_d \geq 1] \leq \mathbb{E}[Z_d]$.

For the identity, expand Z_d as a sum over nonzero messages and condition on the outer realization. Independence of O from (Π, I) gives

$$\mathbb{E}[Z_d] = \mathbb{E}_O \left[\sum_{x \neq 0} q_d(O(x)) \right].$$

Group the terms by $\tau = c(O(x))$ and apply (3). Equation (4) gives the final inequality. Equality holds under the condition in the theorem statement. $\square$

If $\mathcal{G}$ is an event determined only by the outer setup, the theorem also applies with O drawn from its conditional distribution given $\mathcal{G}$. The pair (II, I) remains independent of the conditioned outer encoder. A proof can therefore first restrict to an event $\mathcal{G}$ on which the outer spectrum obeys a required bound, and then certify the failure envelopes $Q_\tau(d)$ on that event.

For a uniform interleaver independent of the inner setup, the failure probability $q_d(u)$ depends only on $\text{wt}(u)$. Section 4 uses this specialization to explain the two supporting constructions; Appendix A.1 gives its full statement. Structured SPIN instead uses occupation and row-weight classes, with a uniform failure envelope for every word in each class (Section 6).

4 Building Intuition: Accumulation and Random Mixing

The outer gives sparse messages an initial weight floor; the inner must turn this local protection into distance proportional to the full codeword length. We compare two constructions with the same outer and permutation to see what helps, and what obstructs, this spreading.

Choose an even block length B and split the message into M blocks of $B/2$ bits each. The outer encodes block i using a linear injection $C_i : \mathbb{F}_2^{B/2} \rightarrow \mathbb{F}_2^B$. Setup samples these maps independently and uniformly from all such injections. For each fixed nonzero block input, its output is therefore uniform over the nonzero B -bit vectors. Under the logarithmic block schedules below, with probability $1 - o(1)$, every nonzero block output has weight proportional to B . In particular, this rules out weight-one outer outputs. But a message supported on one block still produces at most B nonzero bits, not a constant fraction of the full length $N = MB$.

An independently sampled uniform permutation scatters the concatenated outer outputs across the N positions. Conditional on any fixed outer word of weight w , every support of size w is equally likely. The inner then processes this scattered input. All setup choices are sampled once and reused for every message.

The Minimal Inner: An Accumulator. For interleaved input u , set $y_0 := 0$ and compute

$$y_i := y_{i-1} + u_i, \quad i = 1, \dots, N.$$

Write $\text{Acc}_N(u) := y$. With this inner, SPIN truncates BAA [20] to a single accumulator stage. The inner has one bit of state. An input one starts a run of output ones, but the next input one stops it. For example, input 00110000 produces output 00100000: two adjacent ones leave only one nonzero output. The outer's weight floor does not prevent this kind of cancellation. An even-weight outer word of weight w can be interleaved into adjacent pairs, producing only $w/2$ output ones. More generally, each run of output ones starts at an odd-numbered input one and ends just before the next input one. Unfavorable permutations place the two input ones of each such pair close together, leaving short runs of output ones separated by long runs of zeros.

The question is how often these unfavorable arrangements occur across all nonzero messages. Counting the lengths of the output runs of ones and zeros gives an exact input–output weight enumerator for the accumulator. Averaging over the permutation and outer gives the expected number of low-weight codewords. As the outer weight increases, a low-weight output requires more short runs. Logarithmically growing outer blocks make these events rare enough that the expected number of low-weight codewords tends to zero at a positive relative distance.

Theorem 5 makes this concrete: with its block schedule $B \geq 3 \log_2 N$, the rate-1/2 family has relative distance greater than 0.0037 with probability $1 - o(1)$ over setup. Thus a one-bit inner suffices for linear distance, despite its easy cancellation patterns. With a terminated streaming outer, it also answers the question of Bazzi, Mahdian, and Spielman [1] (Appendix A.4).

To obtain larger distance, one can permute and accumulate again, as in BAA codes [20]. The additional permutation breaks up the first accumulator’s runs, but incurs another pass of irregular data movement. A stronger inner can instead sustain spreading after a single permutation by making it harder to return to the zero state.

A Stronger Inner: Random Recursive Convolution. We keep the same outer and permutation, but use the recursive inner from Expand–Convolute [23]. We derive a new, exact expected input–output weight enumerator for this inner; it combines directly with the outer spectrum in our proof framework. Rather than storing one parity bit, the inner remembers the last m outputs. Sample independent $\alpha_i \leftarrow \mathbb{F}_2^m$ at setup, independently of the outer and interleaver, and set $y_j := 0$ for $j \leq 0$. Compute

$$y_i := u_i + \langle \alpha_i, (y_{i-1}, \dots, y_{i-m}) \rangle.$$

For a fixed input, condition on the preceding outputs. Whenever the state is nonzero, the independent coefficient vector makes the next output bit uniform, regardless of the current input bit. A second input one therefore does not simply end the spreading started by the first. Immediately after an output one, returning to the zero state requires m consecutive output zeros. Each of these steps still has nonzero state, so this reset has probability 2^{-m} , provided m positions remain. Once the state is zero, the next input one makes it nonzero again.

This gives the proof two cases to control. A low-weight output either spends substantial time in the zero state or contains unusually few ones during its nonzero-state steps. Long memory makes resets rare, while the outer and permutation supply scattered input ones that restart spreading. Because each nonzero-state output bit is uniform, few ones during those steps is also unlikely. Tracking the number of trailing output zeros, up to m , and recording input and output weights gives this enumerator. Summing over outer words then bounds the probability that any nonzero message has low-weight output. As for a random rate-1/2 code, the lower tails must be small enough to overcome the $2^{N/2} - 1$ nonzero messages.

Theorem 6 combines this calculation with logarithmic outer block size and memory to recover nearly the full random-code distance. At rate $1/2$, it proves relative distance greater than 0.11002 with probability $1 - o(1)$, close to the Gilbert–Varshamov value of approximately 0.110028 . The complete enumerator and the bounds for sparse and linear weights appear together in Appendix A.3.

Toward Efficient Encoding. Random maps simplify the analysis but are costly to encode. A dense random outer block takes $\Theta(B^2)$ bit operations; over N/B blocks with $B = \Theta(\log N)$, this is $\Theta(N \log N)$ work. The accumulator itself is linear-time, whereas the random recursive inner evaluates $m = \Theta(\log N)$ feedback taps at each position, adding another $\Theta(N \log N)$ term. The uniform permutation also requires irregular data movement.

Section 5 replaces each costly component with a structured alternative of linear work.

5 Structured SPIN

5.1 From Random Mixing to Structured Mixing

The examples in Section 4 motivate three changes: use a fast outer constituent with a certified spectrum, control how the outer support reaches the inner, and amortize random mixing over groups of coordinates. A sparse message illustrates the design; the proof must also control larger supports because their multiplicities can dominate the first moment.

First, change the outer. Following chosen-block BAA [22], replace the outer’s independently sampled dense maps with repeated copies of a chosen constituent code. We can select this code for both efficient encoding and a favorable weight distribution, then reuse its encoding circuit across blocks. The proof needs more than its minimum distance: we must also bound how many codewords occur at each low weight. Equal-weight words can still have different coordinate patterns. Independently permuting the coordinates of each block removes this pattern: for each fixed message, each block’s support is uniform given its weight, even when all positions share one constituent.

Second, replace the uniform global interleaver. It can concentrate an active block near the end of the recursion, leaving too little time to spread its support. Requiring heavier outer words can reduce that risk but may increase encoding cost. Instead, transpose the independently shuffled blocks into regions and shuffle each region independently. This structure also speeds up encoding: the transpose admits tiled data movement, and each regional shuffle accesses only part of the full vector. We call an outer block a row. A row of weight w then reaches exactly w distinct regions, with at most one bit per region. This constrains late activation without excluding it; the shuffles randomize the remaining alignment.

Third, process the inner in steps that each consume a group of coordinates, rather than one coordinate at a time. With m -bit memory, the random convolution uses $\Theta(m^2)$ operations to produce m output bits. Grouping lets us replace these dense inner products with fast fixed circuits: for example, an $O(m \log m)$

transform can process m coordinates together. The work is shared across the group rather than repeated at every coordinate.

Grouping also lets us reduce randomness, which is expensive to apply. Ideally, the inner would use only fixed maps, but we do not have a tractable distance proof for that choice. Our compromise keeps the expansion and feedback maps fixed and introduces random state mixing once per step. The Independent-Map Transvection (IMT) inner implements this mixing with a cheap transvection sampled at setup for each step. This retains the randomness needed by the proof without paying for dense random feedback at every coordinate. The following construction specifies these maps and their setup distributions.

5.2 Construction at a Glance

We first define one native-length member of the family. Choose a constituent length b , a constituent dimension $k \leq b$, a number L of outer-block positions, an inner step width t , and a state dimension s , with $t \mid L$. Set

$$N := Lb, \quad K := Lk, \quad e := L/t, \quad R := N/t = be. \quad (6)$$

Thus the interleaver will have b regions, each region will contain L coordinates, and the inner will process each region in e steps.

The encoder has the following dataflow:

$$\mathbb{F}_2^{Lk} \xrightarrow{O} (\mathbb{F}_2^b)^L \xrightarrow{\Pi_{\text{blk}}} (\mathbb{F}_2^b)^L \xrightarrow{T} (\mathbb{F}_2^L)^b \xrightarrow{\Pi_{\text{reg}}} (\mathbb{F}_2^L)^b \xrightarrow{\text{serialize}} \mathbb{F}_2^N \xrightarrow{I} \mathbb{F}_2^N. \quad (7)$$

Here O applies the constituent encoders, Π_{blk} shuffles within blocks, T transposes, and Π_{reg} shuffles within regions. The inner I processes the serialized regions in t -bit steps. The next three subsections define these arrows.

There are two kinds of construction data. The fixed data are the integers (L, b, k, t, s) , two linear maps $A : \mathbb{F}_2^s \rightarrow \mathbb{F}_2^t$ and $C : \mathbb{F}_2^t \rightarrow \mathbb{F}_2^s$, a sampling distribution for invertible state maps $M_i : \mathbb{F}_2^s \rightarrow \mathbb{F}_2^s$, and a joint sampling distribution for injective constituent maps $G_j : \mathbb{F}_2^k \rightarrow \mathbb{F}_2^b$ for $j \in [L]$. The construction must specify whether the positions use one fixed constituent, one shared sampled constituent, or independently sampled constituents. The distinction changes the expected outer enumerator.

Setup samples the constituent maps from this joint distribution. Independently, it samples one uniform coordinate permutation for every outer block, one uniform permutation for every region, and one state map for every inner step. All routing permutations and state maps are mutually independent. After setup, every arrow in (7) is fixed. A family selector is a sequence $(\theta_m)_m$ of these parameter choices and sampling distributions; we suppress m until the scalable instantiation in Section 6.1.

5.3 Block-Diagonal Outer

Write a message as $u = (u_1, \dots, u_L)$ with $u_j \in \mathbb{F}_2^k$. The outer encoder is the direct sum

$$O(u) := (G_1(u_1), \dots, G_L(u_L)) \in (\mathbb{F}_2^b)^L. \quad (8)$$

Equivalently, a generator matrix for O places the local generator matrices on the diagonal. The most basic instantiation repeats one constituent $G : \mathbb{F}_2^k \rightarrow \mathbb{F}_2^b$ in every position. The asymptotic construction in Section 6.1 uses a growing, Golay-based BAA constituent: extended Golay blocks followed by two permute–accumulate stages. The finite-length constructions in Section 7 instead repeat fixed BCH constituents.

The outer proof interface is its local weight enumerator. For a realized local map G_j , define

$$P_{G_j}(z) := \sum_{v \in \mathbb{F}_2^k \setminus \{0\}} z^{\text{wt}(G_j(v))}. \quad (9)$$

The variable z records weight: the coefficient of z^w counts the nonzero block messages whose encoded output has weight w . These coefficients are ordinary integer counts, not elements of $\mathbb{F}_2$. We write $[z^h]F(z)$ for the coefficient of z^h in a polynomial $F(z)$. If one deterministic map G is repeated, the number of messages with exactly q active blocks and outer weight h is

$$\binom{L}{q} [z^h] P_G(z)^q. \quad (10)$$

Here $\binom{L}{q}$ chooses the q nonzero message-block positions. Expanding $P_G(z)^q$ chooses one nonzero message for each of these blocks; multiplication adds their output weights in the exponent of z . The coefficient $[z^h]$ therefore counts exactly the choices with total outer weight h . For example, if $P_G(z) = 2z^2 + z^4$, then $[z^6]P_G(z)^2 = 4$: either block can supply weight two, with two choices for its message, while the other supplies weight four.

For independently sampled, identically distributed local maps, expectation replaces P_G in (10) by $\mathbb{E}[P_{G_1}]$. If one sampled map is reused in all positions, the correct quantity is instead

$$\binom{L}{q} \mathbb{E}[[z^h] P_G(z)^q]. \quad (11)$$

In particular, an expectation over one sampled constituent must not be moved inside the q th power.

The distance proof need not know how the local encoder is implemented. It may consume an exact enumerator, an expected enumerator under the declared setup, or a coefficientwise certified envelope. Minimum distance alone is generally insufficient because nearby low-weight shells can have much larger multiplicity.

5.4 Structured Interleaver

Index the outer coordinates by pairs $(j, r) \in [L] \times [b]$, where j is the outer-block position. For every $j \in [L]$, setup samples $\sigma_j \leftarrow S_b$ and defines $\Pi_{\text{blk}}(j, r) := (j, \sigma_j(r))$. The deterministic transpose changes the coordinate view by $T(j, r) := (r, j) \in [b] \times [L]$. For every region $r \in [b]$, setup samples $\pi_r \leftarrow S_L$ and defines $\Pi_{\text{reg}}(r, j) := (r, \pi_r(j))$. The complete interleaver is

$$\Pi := \Pi_{\text{reg}} \circ T \circ \Pi_{\text{blk}}. \quad (12)$$

We serialize the output of (12) in region-major order. Thus region r occupies the consecutive coordinates $(r-1)L+1, \dots, rL$.

Each outer block contributes exactly one coordinate to each of the b regions. Conditional on the local output weight, σ_j makes its active coordinates a uniform subset of the regions. Conditional on every region weight, π_r makes the support uniform within that region. These two symmetries remove the original coordinate pattern while preserving the deterministic spreading supplied by the transpose.

5.5 Structured Recursive Inner

The inner processes the serialized interleaver output $X = (X_0, \dots, X_{R-1})$ in t -bit steps, with $X_i \in \mathbb{F}_2^t$. For the IMT recurrences, step inputs, step outputs, and state-space vectors are columns. The state space is $\mathbb{F}_2^s$. The matrices of A and C have dimensions $t \times s$ and $s \times t$, respectively. They need not satisfy $C = A^\top$ or $CA = 0$. For IMT, setup samples $u_i \leftarrow \mathbb{F}_2^s \setminus \{0\}$ and then $v_i \leftarrow \{v \in \mathbb{F}_2^s : \langle u_i, v \rangle = 0\}$, including zero, and sets $M_i := I + u_i v_i^\top$. The pairs are independent over $0 \leq i < R$. Since $(u_i v_i^\top)^2 = 0$, each M_i is invertible. Starting from $Q_0 := 0$, the inner computes

$$Y_i := X_i + A(Q_i), \quad Q_{i+1} := M_i Q_i + C(X_i) \quad (0 \leq i < R). \quad (13)$$

It outputs $I(X) := (Y_0, \dots, Y_{R-1})$ and discards Q_R without a flush. State persists across region boundaries. For each fixed $q \neq 0$,

$$\mathcal{L}(M_i q) = \frac{1}{2} \delta_q + \frac{1}{2} \text{Unif}(\mathbb{F}_2^s \setminus \{0\}).$$

Here the right side denotes a mixture of probability measures. If $u_i = q$, then $\langle v_i, q \rangle = 0$ and $M_i q = q$; for every other u_i , the bit $\langle v_i, q \rangle$ is fair. Thus any nonzero target other than q has probability $1/(2(2^s - 1))$. The finite constructions below use this same transvection law.

The transpose $I^\top$ processes input groups $U_i \in \mathbb{F}_2^t$ in reverse order. Starting from $r_R = 0 \in \mathbb{F}_2^s$, it emits groups $V_i \in \mathbb{F}_2^s$:

$$V_i = U_i + C^\top r_{i+1}, \quad r_i = A^\top U_i + M_i^\top r_{i+1}. \quad (14)$$

Both feedback computations consume the raw input, not the emitted output.

Lemma 1 (Inner bijectivity). *For every fixed state-map schedule and every fixed pair of linear maps A, C , the map $I : (\mathbb{F}_2^t)^R \rightarrow (\mathbb{F}_2^s)^R$ defined in (13) is bijective.*

Proof. Fix $Y_0, \dots, Y_{R-1}$. The initial state is known. At step i , recover $X_i = Y_i + A(Q_i)$ and then compute Q_{i+1} from (13). This procedure recovers exactly one input sequence from every output sequence. $\square$

5.6 Composition and Rate

Definition 1 (Structured SPIN). For a selector θ_m and one realization of its setup, the Structured SPIN encoder is

$$E_m := I_m \circ \Pi_m \circ O_m : \mathbb{F}_2^{K_m} \longrightarrow \mathbb{F}_2^{N_m}. \quad (15)$$

Every realized local map is injective. The outer direct sum is therefore injective. The interleaver is a permutation, and Lemma 1 makes the inner bijective. Hence every realized Structured SPIN encoder is injective and has rate

$$\frac{K_m}{N_m} = \frac{k_m}{b_m}. \quad (16)$$

The distance analysis can now vary the constituent family without changing the routing or inner definitions. It must supply a certified outer spectrum or envelope and a conditional failure envelope for the structured route classes used in Theorem 1.

The scalable family below lets the base constituent and b_m vary while retaining the fixed IMT step and state dimensions. Finite-length instantiations may tune these choices at one target length. Both cases use Definition 1.

6 Asymptotic Distance and Linear Work

We now instantiate Structured SPIN with a growing outer and fixed IMT maps. The constituent length b grows logarithmically, but the outer encoding work is linear. The proof must handle both messages concentrated in a few outer blocks and messages occupying a constant fraction of the blocks.

6.1 Scalable Constituent Family

The scalable outer begins with an even base length a and an injective rate-1/2 map $H_a : \mathbb{F}_2^{a/2} \longrightarrow \mathbb{F}_2^a$. Its weight enumerator

$$W_a(z) := \sum_{v \in \mathbb{F}_2^{a/2}} z^{\text{wt}(H_a(v))}$$

must be known exactly or bounded by a certified envelope. For any multiple b of a , let $H_{a,b}$ be the direct sum of b/a copies of H_a . Setup samples two independent permutations $\tau_1, \tau_2 \leftarrow S_b$ and constructs the rate-1/2 BAA constituent

$$\mathcal{O}_{a,b} := \text{Acc}_b \circ \tau_2 \circ \text{Acc}_b \circ \tau_1 \circ H_{a,b}. \quad (17)$$

It samples $\mathcal{O}_{a,b}$ once and reuses that realization in every outer position. This is the chosen-block form of the BAA construction [20,22].

The base length a is a construction parameter. A distance certificate must fix a particular H_a , its spectrum, and an admissible schedule for b . Changing a or H_a changes the outer enumerator and may change the certified schedule. It does not change the route or the structured inner.

The Asymptotic Instantiation. For the theorem below, set $a := 24$ and let $H_{24} : \mathbb{F}_2^{12} \rightarrow \mathbb{F}_2^{24}$ encode the extended binary Golay code. Fix a positive integer m and set $L_m := 128m$. Let b_m be the least positive multiple of 24 satisfying

$$b_m \geq \frac{39}{4} \log_2(L_m b_m), \quad N_m := L_m b_m, \quad K_m := N_m/2. \quad (18)$$

Then $b_m = \frac{39}{4} \log_2 N_m + O(1)$, and consecutive native lengths have relative gap $o(1)$.

Use $\mathcal{O}_{24,b_m}$ from (17). Setup reuses this sampled map in all L_m diagonal positions: $G_{m,1} = \dots = G_{m,L_m} := \mathcal{O}_{24,b_m}$. The expectation over (τ_1, τ_2) remains outside the q th enumerator power, as in (11).

Setup then samples the block shuffles, region shuffles, and transvections as in Section 5, independently of (τ_1, τ_2) . The inner fixes $(t, s) = (128, 19)$ and the IMT maps A, C in Table 4.

The selected maps have a small, explicit weight structure. For nonzero states q , the only values of $\text{wt}(Aq)$ are 48, 56, 64, 72, 80. The transfer analysis tracks these five shells rather than all 2^{19} states. The 128 columns of C are distinct nonzero weight-five vectors spanning $\mathbb{F}_2^{19}$, so each input coordinate contributes to only five state coordinates. This gives sparse feedback. The proof does not treat the feedback syndromes $C(X_i)$ as uniform: it uses the exact spectrum of $C^\top$ to bound cancellations. The appendix supplies the map bases, shell multiplicities, and transfer entries; Section 8 describes the circuits used to apply the maps.

Theorem 2 (Scalable Structured SPIN). *For the setup above,*

$$\mathbb{P}[d_{\min}(E_m) \leq \lfloor 0.11N_m \rfloor] = o(1) \quad (m \rightarrow \infty). \quad (19)$$

The probability is over the two BAA permutations, all route permutations, and all IMT transvections. Every realized code has rate $1/2$. Its ordinary and transposed encoders use $O(N_m)$ bit operations.

The distance and rate claims of Theorem 2 are formally verified in Lean; see Appendix B.1.

The argument has three parts. First, control the realized spectrum of the one sampled outer. Next, bound the route-inner failure probability for each number of active blocks. Finally, sum those bounds before applying the first-moment theorem. Appendix B.1 gives the spectrum calculations, transfer entries, and numerical witnesses used below. The block constant $39/4$ is the value certified here; a smaller certified constant would change the schedule, not the construction.

6.2 Selecting the Shared Outer

Suppress the family index m , and write $\mathcal{O}_b := \mathcal{O}_{24,b}$. Let $A_{\mathcal{O}_b}(w)$ count its nonzero words of weight w , and let $\bar{A}_b(w) := \mathbb{E}_{\tau_1, \tau_2} A_{\mathcal{O}_b}(w)$. The exact formula in Lemma 12 composes the Golay enumerator with two accumulator weight

transitions. It supplies both the tail bound and the spectrum envelope. Let $\mathcal{W} := [0.104, 0.896]$. Lemma 13 proves

$$\sum_{w/b \notin \mathcal{W}} \bar{A}_b(w) = o(1). \quad (20)$$

Lemma 2 (One-sample selection). *There is an event $\mathcal{G}_b$, determined by the two BAA permutations, such that*

$$\mathbb{P}[\mathcal{G}_b] = 1 - o(1). \quad (21)$$

On $\mathcal{G}_b$, every nonzero constituent word has relative weight in $\mathcal{W}$, and

$$A_{\mathcal{O}_b}(w) \leq b^2 \bar{A}_b(w) \quad (0 \leq w \leq b). \quad (22)$$

Proof. The expected number of nonzero tail words equals the left side of (20). Markov's inequality shows that no such word exists with probability $1 - o(1)$. For each remaining weight w , another application of Markov's inequality gives

$$\mathbb{P}[A_{\mathcal{O}_b}(w) > b^2 \bar{A}_b(w)] \leq b^{-2}.$$

A union bound over at most $b + 1$ weights costs $O(1/b)$. Intersecting the two events proves the lemma. $\square$

The $o(1)$ loss from Lemma 2 is incurred once. After setup samples $\mathcal{O}_b$, the construction reuses that realization at every outer position.

Fix any realization satisfying $\mathcal{G}_b$. All subsequent bounds are uniform over this choice; the remaining randomness consists of the independent route permutations and IMT transvections. The spectrum calculation also gives a concave function $\hat{a}_{\text{BA}}$ on $\mathcal{W}$ such that

$$A_{\mathcal{O}_b}(w) \leq \exp\{b\hat{a}_{\text{BA}}(w/b) + O(\ln b)\}.$$

Appendix B.1 derives this majorant from the exact expected spectrum, including the uniform remainder. Selection lets us multiply bounds on the realized spectrum across rows. It does not replace the expectation of a power by a power of the mean.

For $0 \leq d \leq N$ and $1 \leq Q \leq L$, let $Z_{d,Q}$ count nonzero messages with exactly Q active outer blocks and output weight at most d . Our goal is to show that $\sum_{Q=1}^L \mathbb{E}[Z_{\lfloor 0.11N \rfloor, Q} \mid \mathcal{G}_b] = o(1)$.

6.3 From Row Weights to an iid Comparison

View an outer word as an L -tuple of rows in $\mathbb{F}_2^b$. For the framework in Section 3, take its class to be the ordered row-weight profile $c(u) := (\text{wt}(u_1), \dots, \text{wt}(u_L))$. This finite class records the information preserved by the independent block shuffles. The analysis below constructs a uniform route-inner failure envelope

for every such profile. It then aggregates profiles by occupation and mean row weight where those coarser parameters suffice.

Fix Q active outer positions. Let their realized output weights be $w_1, \dots, w_Q$, and set

$$x_j := \frac{w_j}{b}, \quad \alpha := \frac{Q}{L}, \quad x := \frac{1}{Q} \sum_{j=1}^Q x_j, \quad q := \alpha x.$$

When α has a positive lower bound, the next lemma reduces the routing law to a comparison that depends only on q , and the concavity step (24) reduces the outer spectrum to (α, x) . We use binary relative entropy with natural logarithms,

$$D_{\text{KL}}(u||v) := u \ln(u/v) + (1-u) \ln((1-u)/(1-v)),$$

with the usual continuous endpoint conventions.

Lemma 3 (Route domination). *Let F be any nonnegative function of the routed word, viewed as b regions of L bits. Conditioned on the row weights $w_1, \dots, w_Q$, the structured route satisfies*

$$\mathbb{E}_{\text{route}}[F] \leq (b+1)^Q (L+1)^b \mathbb{E}_{\text{Ber}(q)^{Lb}}[F]. \quad (23)$$

If α is bounded below by a positive constant, the multiplicative factor in (23) is $\exp(o(N))$.

Proof. For row j , begin with b independent Bernoulli- x_j variables and condition their sum to equal w_j . The value w_j is a mode of the binomial distribution. Hence the inverse conditioning probability is at most $b+1$. Applying this reduction to all active rows costs at most $(b+1)^Q$.

Before the region shuffle, one fixed region contains independent Bernoulli variables whose average success probability is q . Let S be their sum. For every $0 \leq k \leq L$, the Chernoff bound for a Poisson-binomial sum gives

$$\mathbb{P}[S = k] \leq \exp\{-LD_{\text{KL}}(k/L||q)\}.$$

The method-of-types lower bound gives

$$\mathbb{P}[\text{Bin}(L, q) = k] \geq \frac{1}{L+1} \exp\{-LD_{\text{KL}}(k/L||q)\}.$$

After a uniform region shuffle, both laws are uniform on every weight- k slice. The routed region law is therefore pointwise at most $L+1$ times the iid Bernoulli- q law. Multiplying over the b independent region shuffles proves (23).

Finally,

$$\ln((b+1)^Q (L+1)^b) = Q \ln(b+1) + b \ln(L+1) = o(Lb)$$

when Q/L is bounded below by a positive constant and $b, L \rightarrow \infty$. $\square$

Concavity of $\hat{a}_{\text{BA}}$ now reduces the selected outer spectrum to the mean row weight:

$$\frac{1}{Q} \sum_{j=1}^Q \hat{a}_{\text{BA}}(x_j) \leq \hat{a}_{\text{BA}}(x). \quad (24)$$

The region shuffles are essential for Lemma 3. The deterministic transpose alone spreads each row, but it does not erase the row's coordinate pattern.

6.4 IMT Moments at Positive Occupation

The iid comparison is useful because the inner has a finite-state description. For iid Bernoulli- β input and $0 < z < 1$, weight each transition by z to the emitted Hamming weight. Track mass at zero, arbitrary nonzero mass, and uniform mass on each of the five nonzero image-weight shells of A . This gives seven nonnegative coordinates. For a live state, the transvection law of Section 5 is a mixture of two branches. The lazy branch, $M_i Q_i = Q_i$, can cancel through $CX = Q_i$; the refresh branch spreads a live state uniformly. Both effects enter the transfer bound.

Appendix B.1 constructs nonnegative matrices $T(\beta, z)$ that dominate these weighted measures step by step. The occupation envelope $T_{\text{occ}}(\beta, z)$ averages the transfer bounds for uniform fixed-weight inputs with weights from $\text{Bin}(128, \beta)$. Starting with the zero-state unit row vector e_Z , induction gives

$$\mathbb{E}[z^{\text{wt}(Y)}] \leq e_Z T(\beta, z)^{N/128} \mathbf{1},$$

where $\mathbf{1}$ is the all-ones column vector. A positive column vector $\mathbf{v}$ satisfying $T\mathbf{v} \leq \lambda\mathbf{v}$ therefore gives a bound $C_{\mathbf{v}} \lambda^{N/128}$, with $C_{\mathbf{v}} := v_Z / \min_i v_i$.

Fix $\alpha \geq 10^{-4}$ and mean row weight $x \in \mathcal{W}$. Markov's inequality contributes $z^{-0.11N}$. The outer spectrum and concavity contribute $\exp\{N\alpha\hat{a}_{\text{BA}}(x) + o(N)\}$. The factors for choosing active positions, row weights, and routing are also $\exp(o(N))$, uniformly in this range.

We may choose a reference density $\beta = py$ rather than αx , where p and y are witness parameters in $(0, 1)$ playing the roles of α and x . Every routed array in the profile has weight $N\alpha x$. On that slice, changing the reference law costs $\exp\{ND_{\text{KL}}(\alpha x \| py)\}$. The chain rule bounds this divergence by $D_{\text{KL}}(\alpha \| p) + \alpha D_{\text{KL}}(x \| y)$. The resulting exponent per output bit is

$$\alpha\hat{a}_{\text{BA}}(x) + D_{\text{KL}}(\alpha \| p) + \alpha D_{\text{KL}}(x \| y) + \frac{\ln \lambda}{128} - 0.11 \ln z.$$

The certificate fixes witnesses on finitely many boxes covering $[10^{-4}, 1] \times \mathcal{W}$. On each affine segment of the outer majorant, this expression is convex in (α, x) ; vertex bounds suffice. Outward-rounded interval arithmetic bounds this exponent above by $-4.10335 \cdot 10^{-7}$ on every box. The finite set of witnesses has bounded prefactors, so the total positive-occupation contribution is at most $L \exp(-\eta N + o(N))$ for some $\eta > 0$.

6.5 Sparse Occupation without a Large Conditioning Loss

The dense comparison loses a factor $(L + 1)^b$. Although subexponential in N , this factor is too large when the useful decay is only exponential in Qb . Sparse inputs need a different reference measure.

Write $h(x) := -x \ln x - (1-x) \ln(1-x)$. The outer majorant satisfies $\widehat{a}_{\text{BA}}(x) - h(x) + (\ln 2)/2 < \ell$ on $\mathcal{W}$, with $\ell := 0.01281$. After summing messages and shuffling each active row, comparison with a fair b -bit row thus costs at most $\exp\{b((\ln 2)/2 + \ell + \epsilon_b)\}$ per row, where $\epsilon_b = O(\ln b/b)$. This comparison concerns the message counting measure, not the law of a single fixed word.

Under the fair-row reference, a region contains Q uniformly placed marks carrying independent fair bits. Generate marks independently with probability $p = (8/5)\alpha$, then condition their count to equal Q . Before conditioning, the bits have density $\beta = (4/5)\alpha$. The inverse conditioning probability per region is at most $8\sqrt{Q} \exp\{LD_{\text{KL}}(\alpha\|p)\}$. For this occupation envelope and $z = 1 - (8/5)\alpha$, an exact polynomial check gives

$$T_{\text{occ}}(\beta, z)\mathbf{v}(\alpha) \leq (1 - 96\alpha)\mathbf{v}(\alpha) \quad (0 < \alpha \leq 10^{-4}),$$

with moment prefactor at most 2048. Multiplying the row-counting, conditioning, and moment bounds yields

$$\begin{aligned} \mathbb{E}[Z_{\lfloor 0.11N \rfloor, Q} \mid \mathcal{G}_b] &\leq 2048 \binom{L}{Q} e^{Qb((\ln 2)/2 + \ell + \epsilon_b)} (8\sqrt{Q})^b \\ &\quad \cdot e^{ND_{\text{KL}}(\alpha\|p)} (1 - 96\alpha)^{N/128} z^{-0.11N}. \end{aligned}$$

The schedule (18) gives $\ln L/b \leq 4(\ln 2)/39$, which bounds the cost of choosing active positions. The leading coefficient per Qb is below -0.01340384 . When $Q \geq 4096$, the remaining conditioning cost per Qb is below 0.002. The other remainders vanish uniformly as b grows. Consequently, for all sufficiently large lengths,

$$\mathbb{E}[Z_{\lfloor 0.11N \rfloor, Q} \mid \mathcal{G}_b] \leq e^{-0.006Qb}, \quad 4096 \leq Q \leq 10^{-4}L.$$

Appendix B.1 gives the polynomial witness and every remainder. The explicit cutoff makes the bound uniform, even when Q grows slowly.

6.6 Fixed Occupation: Mixing between Isolated Impulses

It remains to handle the finite set $Q < 4096$. Each region then contains at most Q active bits. As L grows, those bits typically lie in distinct steps separated by long empty intervals. A live state mixes during an empty interval: after g steps its law is 2^{-g} times its entering law plus $(1 - 2^{-g})$ times the uniform nonzero law. Mixing does not make the state zero.

Put $M := 2^{19} - 1$ and $p_0 := 2^{18}/M$. Every row of A is nonzero, so a uniform live state emits mean weight $128p_0$ per empty step. The geometric mixing law bounds the variance over g empty steps by $3t^2g$. With the tilt $z_L = e^{-\theta/L}$, the

weighted empty-interval kernel therefore converges to $D_\gamma(u) := \text{diag}(1, e^{-\gamma u})$ on zero/live states, where $\gamma = p_0\theta$ and u is the interval's fraction of a region.

An isolated input bit moves a zero state to a live state because the columns of C are nonzero. From a uniform live state, it cancels with probability $1/M$. Its limiting transition is thus

$$P_0 = \begin{pmatrix} 0 & 1 \\ 1/M & 1 - 1/M \end{pmatrix}.$$

For a uniformly placed impulses, the limiting weighted region kernel is

$$K_a(\theta) = a! \int_{\substack{u_i \geq 0 \\ \sum_{i=0}^a u_i = 1}} D_\gamma(u_0) P_0 D_\gamma(u_1) \cdots P_0 D_\gamma(u_a) d\mathbf{u}.$$

The integral uses a independent coordinates. Collisions and short gaps have probability $O_Q(\log L/L)$. The empty-interval estimate makes the remaining error $O_Q(L^{-1/2} + \log L/L)$. Appendix B.1 proves a componentwise multiplicative comparison, including the entries that vanish exactly. Across $b = O(\log L)$ regions, its cost is $\exp(o(1))$.

For $Q = 1, 2$, combine these two-state kernels with the fair-row comparison. The resulting exponents per Qb , including active-position choices, are below -0.10918 and -0.10914 , respectively. For $Q \geq 3$, retain each row weight: assign a positive variable u_j to row j and form the matrix polynomial

$$T_Q(\mathbf{u}, \theta) := \sum_{\mathbf{e} \in \{0,1\}^Q} \left(\prod_j u_j^{e_j} \right) K_{\sum_j e_j}^+(\theta).$$

Here K_a^+ replaces the live-to-live entry of P_0 by 1, giving an entrywise upper bound. Nonnegative coefficient extraction bounds the fixed-row-weight moment by

$$\left(\prod_j \begin{pmatrix} b \\ w_j \end{pmatrix}^{-1} u_j^{-w_j} \right) e_0^\top T_Q(\mathbf{u}, \theta)^b \mathbf{1}_2 \exp(o_Q(b)).$$

A weighted matrix norm separates this expression into one factor per row; the outer majorant then bounds each factor. The appendix proves that separation and verifies its endpoint inequalities. For every fixed $3 \leq Q < 4096$, it gives

$$\mathbb{E}[Z_{\lfloor 0.11N \rfloor, Q} \mid \mathcal{G}_b] \leq \exp\{-0.0008679Qb + o_Q(b)\}.$$

The remainder may depend on Q : only finitely many such bounds are needed.

6.7 Completing the Argument

Proof (Proof of Theorem 2). Fix a shared outer realization in $\mathcal{G}_b$. The route-
inner bounds hold uniformly over this realization. Averaging under the outer law

conditioned on $\mathcal{G}_b$ gives the conditional first-moment bounds below. The fixed-occupation arguments give a vanishing sum over the finite set $1 \leq Q < 4096$. Equation (69) gives a uniform geometric sum for $4096 \leq Q \leq 10^{-4}L$. The positive-occupation certificate bounds the remaining sum by $L \exp(-\eta N + o(N))$ for some $\eta > 0$. Thus $\sum_{Q=1}^L \mathbb{E}[Z_{\lfloor 0.11N \rfloor, Q} \mid \mathcal{G}_b] = o(1)$. Markov's inequality and $\mathbb{P}[\mathcal{G}_b^c] = o(1)$ prove the distance claim.

Rate follows from injectivity. Every outer row costs $O(b)$ in either circuit direction: it uses fixed Golay circuits, two permutations, and two accumulators. The route costs $O(N)$, and the fixed-size binary IMT steps cost $O(1)$ each. The transposed recurrence (14) gives the same bound for transposed encoding. $\square$

For any sufficiently large requested output length n , let $m(n)$ be the largest index satisfying $N_{m(n)} \leq n$. Define

$$\tilde{E}_n(x) := E_{m(n)}(x) \| 0^{n-N_{m(n)}}. \quad (25)$$

This zero extension preserves injectivity and minimum distance. Consecutive native lengths have relative gap $o(1)$, so $N_{m(n)}/n = 1 - o(1)$. The wrapped family therefore has rate $1/2 - o(1)$, relative distance greater than $0.11 - o(1)$ with high probability, and $O(n)$ ordinary and transposed work.

7 Finite-Length Structured SPIN

The growing outer supplies the spectral control needed by the asymptotic theorem. At a fixed length, the goal is an explicit setup failure bound, such as 2^{-40} , at low encoding cost. Reusing a sampled constituent requires control of its realized spectrum, not just its ensemble mean; see (11). Concentration or selection can consume a significant part of the finite failure budget. Randomized outers can also admit finite guarantees; the difficulty is proof loss, not the design.

A fixed algebraic constituent removes the additional outer-spectrum failure event. An exact spectrum is convenient, but rigorous spectrum inequalities can serve the same interface. This section instantiates that approach with fixed BCH constituents and the same IMT architecture. The rate-1/2 construction uses the same fixed inner maps. The finite proof replaces the growing-outer argument with bounds tailored to that selected constituent and inner.

We give certified rate-1/2 and rate-1/4 BCH constructions. For an upper bound U on distance failure, its margin is $-\log_2 U$ bits. Appendix D studies how outer size, message length, state size, and step width affect the bound. Its diagnostic curves, such as Figure 1, guide parameter searches; the guarantees here cover all nonzero messages.

7.1 The Selected Construction

We now use a binary BCH-derived constituent $\mathcal{B}$ of length 256, dimension 128, and minimum distance at least 38. Its exact weight spectrum is unknown. The

proof instead uses deterministic inequalities for that spectrum, with no exceptional event for the choice of outer.

Let $\mathcal{P}$ and $\mathcal{Q}$ be the parity extensions of the primitive binary BCH codes of length 255 and designed distances 37 and 39, respectively. Their dimensions are 131 and 123. The selected constituent satisfies $\mathcal{Q} \subset \mathcal{B} \subset \mathcal{P}$ and has dimension 128. Appendix C.1 specifies its generators and explains why the spectrum bounds apply to the implemented code.

For K message bits, choose

$$L := K/128, \quad N := 256L = 2K, \quad (t, s) := (128, 19).$$

Use $\mathcal{B}$ in every outer row and the structured interleaver of Section 5. Use the IMT maps $A : \mathbb{F}_2^{19} \rightarrow \mathbb{F}_2^{128}$ and $C : \mathbb{F}_2^{128} \rightarrow \mathbb{F}_2^{19}$ in Table 4, also used by the asymptotic construction. Setup θ consists of independent uniform row permutations, independent uniform region permutations, and independent transvections $M_i = I + u_i v_i^\top$: sample u_i uniformly from the nonzero state vectors and v_i uniformly from $u_i^\perp$, including zero. One setup is shared by all messages.

The inner applies (13) to the routed 128-bit groups X_i :

$$Q_0 = 0, \quad Y_i = X_i + A(Q_i), \quad Q_{i+1} = M_i Q_i + C(X_i).$$

The inner and the routing distribution are therefore those of Section 6; only the outer and the first-moment calculation differ.

7.2 Finite Distance Guarantee

Theorem 3 (Certified BCH-256 Structured SPIN).

For each $K \in \{2^{16}, 2^{18}, 2^{20}, 2^{22}, 2^{24}\}$, let E_θ be the selected encoder above and put $H := \lfloor N/10 \rfloor$. Every realization is a binary linear code of rate $1/2$, and

$$\mathbb{P}_\theta[\exists x \neq 0 : \text{wt}(E_\theta(x)) \leq H] \leq U_K < 2^{-40}. \quad (26)$$

The certified bounds U_K have the margins reported in Table 6.

The margins range from 41.818 to 50.189 bits; at $K = 2^{20}$ the margin is 50.062 bits.

Thus, except with probability below 2^{-40} over setup, every nonzero message has output weight greater than $N/10$. The margin measures a certified upper bound on distance failure, not the true failure probability or the security level of a complete application. The theorem covers the five listed lengths, rather than every length between them. It does not assert an asymptotic distance guarantee for fixed $\mathcal{B}$ and fixed inner parameters.

Proof Overview. Let Z_H count nonzero messages whose output weight is at most H . Let $\mathcal{T}$ be the ordered row-weight classes of nonzero messages; each class determines the number q of nonzero outer rows. Theorem 1 gives

$$\mathbb{P}_\theta[Z_H > 0] \leq \mathbb{E}_\theta Z_H \leq \sum_{\tau \in \mathcal{T}} \hat{A}_\tau^{\text{out}} \hat{Q}_\tau(H) =: U_K. \quad (27)$$

Here $\hat{A}_7^{\text{out}}$ bounds the class multiplicity and $\hat{Q}_\tau(H)$ bounds its conditional failure probability. No independence between different codewords is required.

The $q = 1$ contribution combines a nonnegative inner transfer bound with weighted BCH spectrum inequalities. Sparse occupations use a positive polynomial recurrence that keeps the region-weight constraint. Dense occupations use a comparison with independent Bernoulli inputs, together with input and output tilts. The all-ones outer word is counted separately: its exact multiplicity and deterministic support must not be inflated by an ordinary weight-band envelope.

For the three largest instances, the sparse recurrence covers $2 \leq q \leq 511$, and interval certificates cover $512 \leq q \leq L$. The instances $K = 2^{16}$ and $K = 2^{18}$ end the sparse recurrence at $q = 63$ and $q = 127$, respectively, and use sharper routing-density comparisons, with separate bounds for assignments containing high-weight rows. All contributions are summed in exact rational arithmetic. Appendix C gives the mathematical reductions and identifies the bound inputs and coverage checks.

7.3 A Rate-1/4 Operating Point

At rate 1/4, use the fixed BCH-derived [128, 32, 32] constituent specified in Appendix C.6. Its spectrum is known exactly. Set $K = 2^{20}$, $L = K/32$, and $N = 128L = 4K$. Keep the same routing distribution, expansion A , $(t, s) = (128, 19)$, and transvection law. Use the weight-three feedback map in Table 5. Only the outer code and the feedback map change; the inner recurrence and end conditions are unchanged.

The certified targets are $(\delta, \lambda) = (0.165, 40)$ and $(0.190, 30)$.

Theorem 4 (Certified BCH-128 Structured SPIN). *For this encoder, every setup gives a binary linear code of rate 1/4. For either of these two pairs (δ, λ) ,*

$$\mathbb{P}_\theta[\exists x \neq 0 : \text{wt}(E_\theta(x)) \leq \lfloor \delta N \rfloor] \leq U_\delta < 2^{-\lambda}.$$

The margins are 41.048 bits at $\delta = 0.165$ and 30.033 bits at $\delta = 0.190$. Appendix C.7 records the exact integer cutoffs and additional digits.

The proof uses the same IMT transfer with the separately enumerated feedback spectrum and cancellation data, followed by a complete sum over $1 \leq q \leq 32768$. These two bounds concern different distance thresholds for one setup distribution, not two independently selected inner designs.

8 Encoder Implementation and Performance

We implement the certified BCH-IMT encoders from Section 7, with an additional small-length variant described in Appendix E. The transposed encoder maps N input blocks to K output blocks; each 128-bit block carries 128 parallel binary instances. It applies the transposed inner, reverses the routing, and applies the transposed BCH outer.

Table 1. Same-host transposed-encoder latency in milliseconds. Entries are medians of three process medians, with no input resets, except for the approximate original-BAA reference. The pair (δ, λ) gives the relative distance and margin in bits at the largest listed length, except for the RAA reference marked * below; – denotes no bound or no timing reported. EC uses the conservative, weaker distance and margin target.

Encoder	Rate	(δ, λ)	$K \approx 2^{16}$	$K \approx 2^{18}$	$K \approx 2^{20}$
SPIN, IMT	1/2	(0.10, 50.062)	0.340	1.460	9.289
Original BAA	1/2	(0.10, 20)	–	–	≈ 32
Chosen Golay BAA	1/2	(0.10, 46.386)	0.595	2.781	23.997
Chosen RM BAA	1/2	(0.10, 47.093)	0.866	3.807	27.660
EC (33, 25)	1/2	(0.05, 20)	3.189	13.128	181.825
Binary RAA, repetition 3	1/3	–	2.834	14.171	87.643
Binary RAA, repetition 4	1/4	(0.19, ≈ 13)*	1.009	6.643	45.176

*RAA reference bound at $K = 2^{22}$, not the timed $K = 2^{20}$: approximately 13 bits without setup testing [10, Sec. 3.1]. One-time testing of low-weight messages can improve the margin to about 40 bits (41.5 bits at this size), with near-quadratic setup work [10, Fig. 2]. No such test is used in our benchmark.

The configurations share an IMT template and fixed XOR circuits for the inner maps. Small instances use direct routing; larger instances use tiled routing. Packed indices and workspace-local page routing reduce data movement. Online calls allocate no memory. Symbolic checks, dense reference implementations, and adjoint tests check the optimized maps. Appendix E gives the kernel details, memory costs, correctness checks, and source-to-certificate bindings.

Transposed Comparison. Table 1 compares binary encoders on one Ryzen 9 7950X thread, pinned to CPU 15. We compile with GCC 15.2.0, release optimization, and Zen 4 tuning, with boost disabled; compiler flags differ per implementation (Appendix E). The rate-1/2 BCH circuit uses AVX-512. Except for the original-BAA reference, which is the authors’ measurement of the original BAA implementation, each cell is the median of three serial process medians; each process has three warmups and 101 timed calls. Setup, allocation, and output checks are excluded. Input buffers are initialized once, with no copy or reset between calls. These are steady-state encoder latencies.

We include original BAA [20] and the newer chosen-block Golay and RM implementations [22]. The chosen-block message lengths are $2^m - 64$ for $m \in \{16, 18, 20\}$; the other rows use $K = 2^m$. Expand–Convolute (EC) uses the conservative profile $(w, m) = (33, 25)$ [23], with its weaker distance and margin. Binary RAA uses repetition three or four [10]. Routing is included: BAA and RAA use Feistel permutations, whereas SPIN uses packed tables. The SPIN cells use the certificate-matched configurations under the same host and timing policy. Rate-1/2 SPIN uses $(t, s) = (64, 12)$ with two transvections per step at $K = 2^{16}$, and the original (128, 19) inner at the larger lengths. All SPIN setup is precomputed; these timings do not use heuristic code refresh.

At $K = 2^{20}$, rate-1/2 SPIN takes 9.289 ms, versus 23.997 ms and 27.660 ms for chosen Golay and RM BAA. The original rate-1/2 BAA baseline takes approximately 32 ms, giving the approximately $3.4\times$ comparison in the abstract. The table reports each construction’s own distance guarantee, not a common application-security level. In particular, the RAA reference bound is not transferred to our smaller timed lengths.

Other Selected Configurations. Rate-1/4 SPIN takes 15.254 ms at $K = 2^{20}$. Rate-1/2 ordinary encoding takes 9.031 ms at that same K , under the same precomputed, repeated-call timing policy. Thus the two directions have similar cost. Appendix E gives the full tables and protocols.

9 Applications

We evaluate SPIN as a component of existing protocols. The three applications use different parts of the encoder interface. We distinguish measured computation costs from projected costs and conditional security calculations.

9.1 Pseudorandom Correlation Generation

PCGs use the transposed encoder to compress pseudorandom tree outputs into correlation blocks. At rate 1/2, the standalone SPIN encoder maps 2^{21} input blocks to 2^{20} output blocks of 128 bits in about 9.3 ms after precomputation. At $K = 2^{18}$, in-place transposed encoding takes 1.419 ms, corresponding to about 185 million correlation blocks per second on one core. This is the linear-encoding stage of OT generation. It excludes tree expansion and output hashing. We also integrate SPIN into libOTe’s semi-honest Silent OT path [7]. In regular-noise mode, the parties reuse the full precomputed SPIN code and expand fresh trees for each batch. At $K = 2^{18}$, the sender takes 2.944 ms per batch with VAES and streaming (non-temporal) output stores, or 89.0 million hashed OTs per second on one core. The timing includes tree expansion, compression, and hashing both sender messages. It excludes initial setup, generation of the supplied base correlations, and network transport; it is local sender computation. Stationary mode instead retains tree seeds and refreshes a precomputed permutation bank; it achieves 84.5 million hashed OTs per second. This heuristic refresh is outside our uniform-setup distance certificates. Appendix F.1 gives both modes’ stage breakdowns, ordinary-store comparisons, and timing contracts.

9.2 Polynomial Commitments

We use SPIN as the row encoder in a Brakedown-style PCS [16]. Arrange a Boolean evaluation table in R rows of K bits, encode each row to $N = 2K$ bits, and Merkle-commit to the columns. In the interactive protocol, the commitment and evaluation claim precede the verifier’s uniform testing coefficients in $\mathbb{F}_{2^{128}}^R$. The prover returns the testing combination and an evaluation-weighted

combination of the message rows. Only then does the verifier sample columns and check both encoded combinations against authenticated openings. The measured implementation derives these challenges from a Fiat–Shamir transcript; Appendix F.2 specifies the order. Keeping the input packed, batching columns, and using table-based row combinations avoid expanding the full Boolean matrix into field elements.

We condition once on the code satisfying Theorem 3; extending its binary generator to $\mathbb{F}_{2^{128}}$ preserves minimum distance. For $K = 2^{16}$ or 2^{18} , one testing fold and 2045 column samples give fixed-matrix testing and evaluation error below 2^{-100} [13]. This conditional target excludes the one-time code-setup event and the commitment-extraction and Fiat–Shamir reductions. It is not a complete composed security theorem: the 100-bit target bounds only the testing term conditional on a good code.

Table 2. Standalone PCS at 512 MiB input volume. Prover time includes commitment and one evaluation opening. SPIN uses the conditional 100-bit fixed-matrix testing target; Ligerito uses its Fast100 and Slim100 profiles. Bolt uses its nominal 100-bit parameters.

Configuration	Prover (ms)	Verify (ms)	Proof (MiB)
SPIN–Brakedown (square)	509	17	18.2
SPIN–Brakedown (longer rows)	525	13	12.4
Ligerito (Fast100)	4331	2	0.5
Ligerito (Slim100)	5567	1	0.3
Bolt-max (amortized)*	2614*	—	11.1*

*Estimated: measured commitment plus a calibrated opening projection; proof size uses a communication proxy. Both assign negligible per-opening cost to the shared matrix proof (amortized limit). Bolt uses $\mathbb{F}_{2^{32}}$ inputs; the other rows use packed Boolean inputs. Verification was not measured. See Appendix F.2.

Table 2 matches input bytes, not coefficient fields or polynomial dimensions. It compares the measured implementations: SPIN reuses a preallocated workspace, whereas Ligerito’s internal allocations remain timed. It does not isolate an encoder-only substitution. The square SPIN shape has $R = K = 2^{16}$; longer rows use $(R, K) = (2^{14}, 2^{18})$. Both encode to 1 GiB. SPIN commits and opens in 509 ms for the square shape. Longer rows reduce proof size by 31.6% at a 3.1% prover-time cost. Ligerito [21] provides much smaller proofs (0.5 versus 18.2 MiB for Fast100 against square SPIN) and faster verification, but its prover takes 4331 ms, about $8.5\times$ longer.

These runs use one worker on the same Ryzen host, with boost disabled and a requested 4.5 GHz frequency. Each measured SPIN or Ligerito entry pools ten trials from two processes, each excluding one warmup. Setup and input preparation are excluded; Appendix F.2 states the implementation-specific allocation and serialization boundaries and the Fast100/Slim100 parameters.

The large SPIN proofs reflect this PCS’s explicit folded messages. Blaze’s generic code-switching framework [10] permits a theoretical composition with asymptotically smaller proofs; we do not instantiate or benchmark it here. Bolt-max [17] provides a comparison at a similar proof size: its estimated 11.1 MiB is close to the longer-row SPIN configuration’s measured 12.4 MiB. Its prover entry is a calibrated projection. Appendix F.2 gives the accounting.

9.3 Flock

We replace Ligerito with SPIN–Brakedown inside Flock [11], which reduces batches of Boolean constraints to claims about a committed witness. The adapter incorporates Flock’s interpolation weights into the row combinations and generates the packed witness directly in the PCS layout. It uses two openings with 2110 column samples each and $K = 2^{16}$. Their conditional fixed-matrix error bounds sum to less than 2^{-102} ; composition with adaptive Flock claims still requires the extraction argument.

For 16,384 and 65,536 BLAKE3-compression constraints, SPIN’s measured prover times are 94 and 351 ms, versus 215 and 571 ms for Ligerito. These are $2.29\times$ and $1.63\times$ prover speedups. At the larger workload, SPIN’s proof is 10.5 MiB and verification takes 42 ms, versus 0.4 MiB and 28 ms for Ligerito. The gain therefore trades communication and verification for prover speed.

Both backends receive the applicable optimizations to arithmetic, workspace reuse, and witness processing. Prover time includes witness generation, commitment, and proving, but excludes setup, verification, and outer serialization. The checked relation is the committed batch’s compression constraints; binding public hash inputs and outputs needs a separate wrapper. Appendix F.3 gives the full measured table, balanced serial timing protocol, phase breakdowns, and separately labeled Bolt projections. No Bolt–Flock implementation was benchmarked.

10 Disclosure of Generative AI Use

The core construction predates this AI-assisted work. The proofs, concrete instantiations, experiments, and implementations were developed through close human–AI collaboration over many author-directed iterations. The manuscript was drafted with AI assistance and revised substantially by the authors. We used GPT-5.6 for most of this work and GPT-6.0 only during the final phase.

AI tools also assisted the Lean proof development, numerical-certificate generation, and verification orchestration. The Lean development included Claude-assisted work and GPT-assisted completion.

The process involved thousands of prompts; a complete transcript is not included because of its volume. The authors retain responsibility for all claims, attribution, and correctness, including AI-generated material.

References

1. Bazzi, L., Mahdian, M., Spielman, D.A.: The minimum distance of turbo-like codes. *IEEE Transactions on Information Theory* **55**(1), 6–15 (2009). <https://doi.org/10.1109/TIT.2008.2008114>
2. 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). <https://doi.org/10.1109/18.669119>
3. 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). <https://doi.org/10.1109/ICC.1993.397441>
4. 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*. *Lecture Notes in Computer Science*, vol. 14929, pp. 276–307. Springer (2024). https://doi.org/10.1007/978-3-031-68403-6_9
5. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y.: Compressing vector OLE. In: *ACM CCS 2018*. pp. 896–912 (2018). <https://doi.org/10.1145/3243734.3243868>
6. 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*. *Lecture Notes in Computer Science*, vol. 13508, pp. 603–633. Springer (2022). https://doi.org/10.1007/978-3-031-15979-4_21
7. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Rindal, P., Scholl, P.: Efficient two-round OT extension and silent non-interactive secure computation. In: *ACM CCS 2019*. pp. 291–308 (2019). <https://doi.org/10.1145/3319535.3354255>, <https://eprint.iacr.org/2019/1159>
8. 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*. *Lecture Notes in Computer Science*, vol. 11694, pp. 489–518. Springer (2019). https://doi.org/10.1007/978-3-030-26954-8_16
9. 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*. Springer (2020). https://doi.org/10.1007/978-3-030-56880-1_14, <https://eprint.iacr.org/2022/1035>, full version: *Cryptology ePrint Archive*, Report 2022/1035
10. Brehm, M., Chen, B., Fisch, B., Resch, N., Rothblum, R.D., Zeilberger, H.: Blaze: Fast SNARKs from interleaved RAA codes. In: *Advances in Cryptology – EUROCRYPT 2025*. pp. 123–152 (2025). https://doi.org/10.1007/978-3-031-91134-7_5, also *IACR Cryptology ePrint Archive*, Report 2024/1609
11. Bünz, B., Rothblum, R., Wang, W.: Flock: Fast proving for batch boolean computations. *Cryptology ePrint Archive*, Paper 2026/1329 (2026), <https://eprint.iacr.org/2026/1329>
12. 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*. *Lecture Notes in Computer Science*, vol. 12827, pp. 502–534. Springer (2021). https://doi.org/10.1007/978-3-030-84252-9_17, <https://eprint.iacr.org/2021/1150>
13. Diamond, B.E., Posen, J.: Proximity testing with logarithmic randomness. *IACR Communications in Cryptology* **1**(1) (2024). <https://doi.org/10.62056/aksdkp10>

14. 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)
15. Fujiwara, T., Kusaka, T.: The weight distributions of the $(256, k)$ extended binary primitive BCH codes with $k \leq 71$ and $k \geq 187$. *IEICE Transactions on Fundamentals* **E104-A**(9), 1321–1328 (2021). <https://doi.org/10.1587/transfun.2020EAP1119>, https://www.jstage.jst.go.jp/article/transfun/E104.A/9/E104.A_2020EAP1119/_article/-char/en
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. Lecture Notes in Computer Science*, vol. 14082, pp. 193–226. Springer (2023). https://doi.org/10.1007/978-3-031-38545-2_7
17. Gurkan, K., Novakovic, A., Rothblum, R.D.: Bolt: Faster SNARKs from sketched codes. *Cryptology ePrint Archive*, Paper 2026/310 (2026), <https://eprint.iacr.org/2026/310>
18. Kahale, N., Urbanke, R.: On the minimum distance of parallel and serially concatenated codes. In: *IEEE International Symposium on Information Theory*. p. 31 (1998). <https://doi.org/10.1109/ISIT.1998.708611>
19. Kliewer, J., Zigangirov, K.S., Koller, C., Costello, Jr., D.J.: Coding theorems for repeat multiple accumulate codes. *arXiv:0810.3422* (2008)
20. 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. Lecture Notes in Computer Science*, vol. 16807, pp. 397–429. Springer (2026). https://doi.org/10.1007/978-3-032-35418-1_13, <https://eprint.iacr.org/2025/1828>, a minor revision is available as *Cryptology ePrint Archive*, Report 2025/1828
21. Novakovic, A., Angeris, G.: Ligerito: A small and concretely fast polynomial commitment scheme. *Cryptology ePrint Archive*, Paper 2025/1187 (2025), <https://eprint.iacr.org/2025/1187>
22. Peceny, S., Rindal, P.: Chosen-block BAA codes: Fast, near the Gilbert–Varshamov bound, and field-agnostic. *Cryptology ePrint Archive*, Paper 2026/1903 (2026), <https://eprint.iacr.org/2026/1903>
23. Raghuraman, S., Rindal, P., Tanguy, T.: Expand-convolute codes for pseudorandom correlation generators from LPN. In: *Advances in Cryptology – CRYPTO 2023, Part IV. Lecture Notes in Computer Science*, vol. 14084, pp. 602–632. Springer (2023). https://doi.org/10.1007/978-3-031-38551-3_19, <https://eprint.iacr.org/2023/882>
24. 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>
25. SPIN contributors: SPIN codes: Source repository. *GitHub* (2026), https://github.com/ladnir/spin_codes
26. Xie, T., Zhang, Y., Song, D.: Orion: Zero knowledge proof with linear prover time. In: *Advances in Cryptology – CRYPTO 2022*. pp. 299–328 (2022). https://doi.org/10.1007/978-3-031-15985-5_11, also *IACR Cryptology ePrint Archive*, Report 2022/1010
27. Yang, K., Weng, C., Lan, X., Zhang, J., Wang, X.: Ferret: Fast extension for correlated OT with small communication. In: *ACM CCS 2020* (2020). <https://doi.org/10.1145/3372297.3417276>, <https://eprint.iacr.org/2020/924>

Supplementary Material

A Supporting Constructions and Their Proofs

This appendix gives the complete analysis of the constructions used for intuition in Section 4; Table 3 summarizes the three families. We first give the shared uniform-interleaver calculation and random-block enumerators, then analyze the accumulator and random recursive inner. Appendix A.3 closes the Random SPIN exponent estimates. The final subsection gives the terminated-encoder interpretation of the BMS connection.

Table 3. From minimal recursion to structured encoding.

Family	Outer	Interleaver Inner	
Accumulator SPIN	random blocks	uniform	accumulator
Random SPIN	random blocks	uniform	random recursive
Structured SPIN	spectrum-controlled	structured	structured recursive

A.1 Uniform Interleaving and Random-Block Outers

The Weight-Indexed Calculation All three families use the same first-moment idea. Fix a target distance d , and count nonzero messages whose encoded words have weight at most d . If the expected count is below one, then some sampled encoder is injective and has minimum distance greater than d .

For a uniform interleaver, the outer Hamming weight is often a sufficient interface. The expected bad-word count then has the form

$$\sum_w A_w^{\text{out}} p_w^{\text{in}}(d), \quad (28)$$

where A_w^{out} is the expected outer spectrum and $p_w^{\text{in}}(d)$ is an inner slice-to-tail probability.

A structured interleaver need not produce a uniform Hamming slice. We therefore allow a class to record block occupation or another routing profile. The interleaver–inner analysis supplies a uniform failure envelope for each class. The weight-indexed sum in (28) is the special case in which the conditional law depends only on weight.

Uniform-Interleaver Specialization Suppose Π is uniform over all permutations of $[N]$. If u has weight w , then $\Pi(u)$ is uniform over the Hamming slice

$$\mathcal{H}_{N,w} := \{v \in \mathbb{F}_2^N : \text{wt}(v) = w\}.$$

Indeed, every vector in $\mathcal{H}_{N,w}$ has exactly $w!(N-w)!$ preimages among the permutations of u .

Let U_w be uniform over $\mathcal{H}_{N,w}$, independent of the inner setup, and define

$$p_w^{\text{in}}(d) := \mathbb{P}_{U_w, I}[\text{wt}(I(U_w)) \leq d]. \quad (29)$$

Also define

$$A_w^{\text{out}} := \mathbb{E}_O[|\{x \neq 0 : \text{wt}(O(x)) = w\}|].$$

Corollary 1 (Weight-indexed first moment). *Under a uniform interleaver independent of O and I ,*

$$\mathbb{E}[Z_d] = \sum_{w=0}^N A_w^{\text{out}} p_w^{\text{in}}(d).$$

Accumulator SPIN and Random SPIN use Corollary 1. Structured SPIN uses Theorem 1 with occupation and row-weight classes; the uniform envelope does not require the conditional failure probability to be identical for all words in a class.

Relation to Block-Accumulate Codes Block-Accumulate (BA) codes concatenate a random block-diagonal outer with one or more stages, each consisting of a uniform interleaver and an accumulator [20]. Accumulator SPIN keeps only the first such stage and therefore may be viewed as a truncated BA code. We use the SPIN name to place this construction in the common outer–interleaver–inner interface, but the block-accumulate architecture and the random-block enumerator below are due to Kolesnikov et al.

Accumulator SPIN follows the BA ensemble at this level: setup samples the local maps independently. Theorem 5 conditions each local map to be injective. This condition makes every realized outer encoder injective.

Chosen-block BAA studies the related fixed-constituent choice [22]. It fixes a short constituent code and uses the same code in every outer position. Its concrete binary constituents include Golay, Reed–Muller, and a BCH subcode. That reuse option is not the ensemble analyzed here.

Random-Block Outer Lineage The following local input–output enumerator is the random-block calculation used for BA codes. We state it in the present notation because it is also the local input to the Accumulator SPIN analysis.

Lemma 4 (Uniform random-block enumerator). *Suppose that G_B is uniform over $\mathbb{F}_2^{K_B \times B}$. Let $E_{a,b}^{\text{blk}}(G_B)$ be the number of inputs of weight a whose images have weight b . Then*

$$\mathbb{E}_{G_B}[E_{a,b}^{\text{blk}}(G_B)] = \begin{cases} 1, & a = 0 \text{ and } b = 0, \\ 0, & a = 0 \text{ and } b > 0, \\ \binom{K_B}{a} \binom{B}{b} 2^{-B}, & a > 0. \end{cases} \quad (30)$$

Proof. The zero input maps to the zero output. Fix a nonzero input u . Each column of G_B is uniform and independent, so the coordinates of uG_B are independent uniform bits. Thus uG_B is uniform over $\mathbb{F}_2^B$. There are $\binom{K_B}{a}$ inputs of weight a and $\binom{B}{b}$ outputs of weight b . Summing the probability 2^{-B} over these pairs gives (30). $\square$

The BA independent-block ensemble has a closed global input-output enumerator. Let $E_{a,b}^{\text{BA}}$ count total input weight a and total outer-output weight b across M blocks. Specializing the BA formula [20] to local dimensions $K_B \times B$ gives

$$\mathbb{E}[E_{a,b}^{\text{BA}}] = \sum_{q=0}^M 2^{-Bq} \binom{M}{q} \binom{Bq}{b} \sum_{j=0}^q (-1)^j \binom{q}{j} \binom{K_B(q-j)}{a}. \quad (31)$$

Here q is the number of nonzero input blocks. The inner alternating sum counts weight- a inputs for which exactly those q blocks are nonzero. For a fixed such input, the q independently sampled local matrices produce a uniform word on the corresponding Bq output coordinates. This yields the factor $2^{-Bq} \binom{Bq}{b}$.

The following exact generating function uses the same independence pattern, but samples uniform injections rather than unconditioned matrices.

Uniform Injections Fix a rational rate $R \in (0, 1)$ and a block length B with RB integral. For M block positions, let $N := MB$ and sample independent uniform linear injections $C_i : \mathbb{F}_2^{RB} \rightarrow \mathbb{F}_2^B$. Let $A_{N,w}^{\text{out}}$ count nonzero messages whose blockwise image has weight w .

For one local injection, define

$$G_{B,R}(z) := (2^{RB} - 1) \frac{(1+z)^B - 1}{2^B - 1}. \quad (32)$$

Every nonzero input has a uniform nonzero image, so independence across block positions gives

$$\mathbb{E} \left[1 + \sum_{w=1}^N A_{N,w}^{\text{out}} z^w \right] = (1 + G_{B,R}(z))^M. \quad (33)$$

A.2 Accumulator SPIN

Accumulator SPIN is the transparent instantiation of the framework. It uses a random block-diagonal outer, a uniform interleaver, and the binary accumulator.

Accumulator Inner Define $\text{Acc}_N : \mathbb{F}_2^N \rightarrow \mathbb{F}_2^N$ by

$$(\text{Acc}_N(u))_t := \sum_{i=1}^t u_i \pmod{2}, \quad t \in [N]. \quad (34)$$

Equivalently, if $y := \text{Acc}_N(u)$ and $y_0 := 0$, then $y_t = y_{t-1} \oplus u_t$.

For $w, h \in \{0, \dots, N\}$, define

$$A_{w,h}^{\text{Acc}} := |\{u \in \mathbb{F}_2^N : \text{wt}(u) = w, \text{wt}(\text{Acc}_N(u)) = h\}|.$$

Lemma 5 (Exact accumulator enumerator). Fix $w \in \{1, \dots, N-1\}$ and set $k := \lceil w/2 \rceil$. Then

$$A_{w,h}^{\text{Acc}} = \binom{h-1}{k-1} \binom{N-h}{w-k}. \quad (35)$$

Proof. Let the ones of u occur at $1 \leq p_1 < \dots < p_w \leq N$. Set $p_0 := 0$, $p_{w+1} := N+1$, and $\Delta_i := p_i - p_{i-1}$. The positive spacings satisfy $\sum_{i=1}^{w+1} \Delta_i = N+1$.

The accumulator output equals one between its first and second flips, between its third and fourth flips, and so on. Hence its weight is the sum of the even-index spacings. There are k such spacings. The number of positive compositions of h into those k parts is $\binom{h-1}{k-1}$. The remaining $w+1-k$ spacings sum to $N+1-h$, which gives $\binom{N-h}{w-k}$ choices. Multiplication proves (35). $\square$

If U_w is uniform over the weight- w slice, then

$$\mathbb{P}[\text{wt}(\text{Acc}_N(U_w)) \leq d] = \frac{1}{\binom{N}{w}} \sum_{h=0}^d A_{w,h}^{\text{Acc}}. \quad (36)$$

Lemma 6 (A simple accumulator tail).

Fix $\delta \in (0, 1/4)$ and $w \in \{1, \dots, \lfloor N/2 \rfloor\}$. Then

$$\mathbb{P}[\text{wt}(\text{Acc}_N(U_w)) \leq \lfloor \delta N \rfloor] \leq (4e\delta)^{\lceil w/2 \rceil}.$$

Proof. Set $k := \lceil w/2 \rceil$. Lemma 5 and the hockey-stick identity give

$$\sum_{h \leq \lfloor \delta N \rfloor} A_{w,h}^{\text{Acc}} \leq \binom{N}{w-k} \binom{\lfloor \delta N \rfloor}{k}.$$

After division by $\binom{N}{w}$, bound the binomial term by $(e\delta N/k)^k$. Since $w \leq N/2$, the remaining binomial ratio is at most $(2w/N)^k$. Finally, $w \leq 2k$, which gives the claim. $\square$

The next contraction is uniform in the input weight. It is the form used by the asymptotic theorem.

Lemma 7 (Sharp accumulator contraction). Fix $\delta \in (0, 1/2)$ and define

$$\rho(\delta) := 2\sqrt{\delta(1-\delta)}.$$

For every $w \in \{1, \dots, N\}$,

$$\mathbb{P}[\text{wt}(\text{Acc}_N(U_w)) \leq \lfloor \delta N \rfloor] \leq \rho(\delta)^w. \quad (37)$$

Proof. Set $D := \lfloor \delta N \rfloor$ and $k := \lceil w/2 \rceil$. For $w < N$, Lemma 5 identifies the probability in (37) with $\mathbb{P}[X \geq k]$, where X is hypergeometric with sample size w and D marked positions among N . For $t > 0$, the moment bound for sampling without replacement gives

$$\mathbb{E}[e^{tX}] \leq (1 - \delta + \delta e^t)^w.$$

Markov's inequality and $k \geq w/2$ give

$$\mathbb{P}[X \geq k] \leq \inf_{t>0} \left(e^{-t/2} (1 - \delta + \delta e^t) \right)^w.$$

The minimizer satisfies $e^t = (1 - \delta)/\delta$, which yields the claim. When $w = N$, the accumulator output has weight $\lceil N/2 \rceil > D$, so the left side is zero. $\square$

Asymptotic Theorem Fix a rational rate $R \in (0, 1)$ and let

$$\mathcal{B}_R := \{B \geq 1 : RB \in \mathbb{Z}\}.$$

For $B \in \mathcal{B}_R$ and each $i \in [M]$, setup samples an independent uniform linear injection

$$C_i : \mathbb{F}_2^{RB} \longrightarrow \mathbb{F}_2^B.$$

It then samples $\Pi \leftarrow S_N$, where $N := MB$. For $x = (x^{(1)}, \dots, x^{(M)})$, define

$$E_N^{\text{acc}}(x) := \text{Acc}_N \left(\Pi(C_1(x^{(1)}) \parallel \dots \parallel C_M(x^{(M)})) \right). \quad (38)$$

The setup remains fixed for all messages. Every realized encoder is injective.

Define

$$\lambda_A(R, \delta) := 1 - R - \log_2(1 + \rho(\delta)). \quad (39)$$

Fix $c > 0$. For each sufficiently large M , let B_M be the least member of $\mathcal{B}_R$ satisfying

$$B_M \geq c \log_2(MB_M),$$

and set $N_M := MB_M$.

Theorem 5 (Accumulator SPIN). *Suppose*

$$\lambda_A(R, \delta) > 0, \quad c\lambda_A(R, \delta) \geq 1.$$

Then

$$\mathbb{P}[d_{\min}(E_{N_M}^{\text{acc}}) \leq \delta N_M] = o(1) \quad (M \rightarrow \infty). \quad (40)$$

The family has rate R , block size $\Theta(\log N_M)$, and $\Theta(N_M \log N_M)$ direct encoding cost.

Proof. Let $D := \lfloor \delta N_M \rfloor$ and let Z_D count nonzero messages whose encoded words have weight at most D . Corollary 1, Lemma 7, and (33) give

$$\mathbb{E}[Z_D] \leq (1 + G_{B_M, R}(\rho(\delta)))^M - 1.$$

For $z > 0$, equation (32) gives

$$G_{B, R}(z) \leq 2 \left(2^{R-1} (1 + z) \right)^B.$$

Set $\lambda := \lambda_A(R, \delta)$. The block schedule implies

$$MG_{B_M, R}(\rho(\delta)) \leq \frac{2}{B_M} N_M^{1-c\lambda} = o(1).$$

Thus $\mathbb{E}[Z_D] = o(1)$, and Markov's inequality proves (40). $\square$

At rate $R = 1/2$, the choice $c = 3$ and $\delta = 37/10000$ satisfies the theorem with a strict interval-certified margin. The corresponding relative-distance guarantee is small but positive. For this proof, the largest possible rate-1/2 limit as $c \rightarrow \infty$ is $0.0449101394\dots$

A.3 Random SPIN

Random SPIN replaces the accumulator with a random recursive convolution. Its outer uses the independent random-block injections from Appendix A.2.

Relation to Silver and Expand-Convolute Silver codes predate Expand-Convolute and use a banded lower-triangular solve for linear-time encoding [12]. The solve is itself a structured recursive encoder of convolutional form. Silver does not sample independent convolution taps at every time step.

Expand-Convolute makes the random convolution explicit. It combines a sparse expander with a non-wrapping recursion whose time-dependent feedback coefficients are independent uniform bits [23]. The Random SPIN theorem uses that inner distribution and replaces the expander with independent random-block injections followed by one uniform interleaver.

We give an exact expected input-output weight enumerator for this inner. We mark input and output weight in the transitions of its trailing-zero state. This preserves the averaged weight law without state-process relaxations and fits the first-moment framework directly: we extract lower-tail bounds and combine them with the outer spectrum.

Random Recursive Inner Fix a memory $m \geq 1$. For every $t \in [N]$, setup samples

$$\alpha_t \leftarrow \mathbb{F}_2^m$$

independently. Set $y_j := 0$ for $j \leq 0$. On input $u \in \mathbb{F}_2^N$, define

$$y_t := u_t + \langle \alpha_t, (y_{t-1}, \dots, y_{t-m}) \rangle, \quad t \in [N]. \quad (41)$$

Write $I_{N,m}^{\text{rnd}}(u) := y$. Once setup fixes $(\alpha_t)_{t=1}^N$, this map is linear and invertible because y_t contains u_t with coefficient one.

Exact Convolution Transfer The random inner admits an exact finite-state transform. Let X mark input weight and let S mark output weight. Index the matrix rows and columns by $z \in \{0, \dots, m\}$, where z is the number of trailing output zeros, clipped at m . Thus $z = m$ means that the convolution state is zero.

Define the nonnegative matrix $T_m(X, S)$ by

$$T_m(X, S)[z, z'] := \begin{cases} (1+X)S/2, & z < m, z' = 0, \\ (1+X)/2, & z < m, z' = z+1, \\ XS, & z = m, z' = 0, \\ 1, & z = m, z' = m, \\ 0, & \text{otherwise.} \end{cases} \quad (42)$$

Let e_m be the unit vector at index m , and let $\mathbf{1}$ be the all-ones vector.

Lemma 8 (Exact random-convolution transform). *For the setup in (41),*

$$\mathbb{E} \left[\sum_{u \in \mathbb{F}_2^N} X^{\text{wt}(u)} S^{\text{wt}(I_{N,m}^{\text{rnd}}(u))} \right] = e_m^T T_m(X, S)^N \mathbf{1}. \quad (43)$$

Proof. Suppose $z < m$. The convolution state is nonzero, so the fresh uniform α_t makes the output bit uniform for either input bit. Summing the two input choices with weights 1 and X gives the first two nonzero transitions in (42). If $z = m$, input zero stays in the zero state, while input one produces output one and moves to state zero. These are the remaining two transitions. Multiplying the one-step transforms and summing over the terminal state proves the identity. $\square$

Let U_h be uniform on the weight- h slice, independently of the inner setup. For $0 < S < 1$, $a > 0$, and $D \in \{0, \dots, N\}$, Lemma 8 gives

$$\mathbb{P}[\text{wt}(I_{N,m}^{\text{rnd}}(U_h)) \leq D] \leq S^{-D} a^{-h} \frac{e_m^T T_m(a, S)^N \mathbf{1}}{\binom{N}{h}}. \quad (44)$$

The first step is the lower-tail moment bound. The second bounds the X^h coefficient by evaluating the nonnegative polynomial at a .

For $\delta \in (0, 1/2)$, define

$$p_{N,h}(\delta) := \mathbb{P}[\text{wt}(I_{N,m}^{\text{rnd}}(U_h)) \leq \lfloor \delta N \rfloor]. \quad (45)$$

Asymptotic Theorem For an even block length B dividing N , set $M := N/B$. Independently sample uniform linear injections

$$C_i : \mathbb{F}_2^{B/2} \longrightarrow \mathbb{F}_2^B, \quad i \in [M],$$

one permutation $\Pi \leftarrow S_N$, and the convolution setup above. Define

$$E_{N,m}^{\text{rnd}}(x) := I_{N,m}^{\text{rnd}} \left(\Pi(C_1(x^{(1)}) \parallel \dots \parallel C_M(x^{(M)})) \right). \quad (46)$$

All setup objects are independent and remain fixed for all messages.

For each sufficiently large M , let B_M^* be the least positive even integer satisfying

$$B_M^* \geq 17 \log_2(MB_M^*).$$

Set

$$N_M^* := MB_M^*, \quad m_M^* := \left\lceil \frac{51}{50} \log_2 N_M^* \right\rceil, \quad \delta_* := \frac{5501}{50000}.$$

Theorem 6 (Random SPIN). *For the ensemble in (46),*

$$\mathbb{P}[d_{\min}(E_{N_M^*, m_M^*}^{\text{rnd}}) \leq \delta_* N_M^*] = o(1) \quad (M \rightarrow \infty). \quad (47)$$

The family has rate 1/2 and relative distance greater than 0.11002; direct encoding costs $\Theta(N_M^* \log N_M^*)$.

Proof. Let $A_{N,h}^{\text{out}}$ be the number of outer words of weight h , and let $p_{N,h}(\delta_*)$ be the probability in (45), which (44) bounds with $D = \lfloor \delta_* N \rfloor$. Split the first-moment sum at $h = N/5000$ and $h = 0.9N$.

Lemma 11 shows that the first range is $o(1)$. Lemma 10 is uniform for $h/N \in [1/5000, 0.9]$, so the middle range is $2^{-\Omega(N)}$. For the final range, use $p_{N,h} \leq 1$ and the rate-1/2 random-block outer exponent:

$$\sum_{h \geq 0.9N} \mathbb{E}[A_{N,h}^{\text{out}}] \leq 2^{N(H_2(0.9) - 1/2 + o(1))} = 2^{-\Omega(N)}.$$

The complete first moment tends to zero. Markov's inequality proves (47). $\square$

The admissible lengths have relative gaps $o(1)$. For a requested length, zero-extend the largest preceding native member. This preserves injectivity and absolute minimum distance while changing rate and relative distance by only $o(1)$.

Random-Family Encoding Costs Under the schedules of Theorems 5 and 6, the outer encoder processes N/B blocks of length $B = \Theta(\log N)$. Direct dense block multiplication therefore costs $\Theta(N \log N)$ bit operations. The accumulator costs $O(N)$. The Random SPIN recurrence evaluates $m = \Theta(\log N)$ taps per output coordinate and hence costs $\Theta(N \log N)$. These bounds concern direct encoders; preprocessing or word-level batching may reduce practical constants.

Random SPIN Exponent Closure This subsection proves the two estimates used by Theorem 6. The outer exponent follows from the exact injection law in (33). For a sequence $h/N \rightarrow \eta \in (0, 1)$, coefficient extraction gives

$$\Phi_{\text{blk}}(\eta) = \begin{cases} \eta \log_2(1 + \sqrt{2}), & 0 < \eta \leq 1 - 2^{-1/2}, \\ H_2(\eta) - 1/2, & 1 - 2^{-1/2} \leq \eta < 1. \end{cases} \quad (48)$$

The inner exponent comes from the transfer matrix in (42). For $a > 0$ and $S \in (0, 1)$, define

$$\lambda_\infty(a, S) := \max \left\{ 1, \frac{(1+a)(1+S)}{2} \right\}.$$

Lemma 9 (Limiting convolution growth). *If $m_N = \Theta(\log N)$, then*

$$\limsup_{N \rightarrow \infty} \frac{1}{N} \log_2 (e_{m_N}^\top T_{m_N}(a, S)^N \mathbf{1}) \leq \log_2 \lambda_\infty(a, S). \quad (49)$$

Proof. Set $b := (1 + a)/2$ and let λ_m be the Perron root of $T_m(a, S)$. The right Perron-vector equations give

$$1 = \frac{bS}{\lambda_m} \frac{1 - (b/\lambda_m)^m}{1 - b/\lambda_m} + \left(\frac{b}{\lambda_m}\right)^m \frac{aS}{\lambda_m - 1}. \quad (50)$$

If $b(1 + S) < 1$, this identity gives $\lambda_m \rightarrow 1$. If $b(1 + S) > 1$, the active-state principal submatrix and (50) give $\lambda_m \rightarrow b(1 + S)$. At equality, any subsequential limit greater than one contradicts (50).

Normalize the Perron vector at state zero. The same equations show that the ratio of its largest and smallest coordinates is $\exp(O(m))$. Comparing the all-ones vector with coordinatewise multiples of this Perron vector bounds the matrix power by $\lambda_m^N \exp(O(m))$. Since $m_N = O(\log N)$, this factor is subexponential and proves (49). $\square$

For $\delta \in (0, 1/2)$, equations (44) and (49) give the inner exponent

$$F_\delta(\eta) := \inf_{\substack{a \geq 0 \\ 0 < S < 1}} [\log_2 \lambda_\infty(a, S) - \eta \log_2 a - \delta \log_2 S - H_2(\eta)]. \quad (51)$$

Set

$$\eta_{\text{conv}}(\delta) := 1 - \frac{1}{2(1 - \delta)}, \quad a_\eta := \frac{\eta}{\sqrt{\eta^2 + \delta^2} + \delta}, \quad S_\eta := \frac{1 - a_\eta}{1 + a_\eta}.$$

Direct minimization yields

$$F_\delta(\eta) = \begin{cases} -\eta \log_2 a_\eta - \delta \log_2 S_\eta - H_2(\eta), & 0 < \eta \leq \eta_{\text{conv}}(\delta), \\ H_2(\delta) - 1, & \eta_{\text{conv}}(\delta) \leq \eta < 1. \end{cases} \quad (52)$$

Lemma 10 (Linear-weight closure). *Set $\delta_* := 5501/50000$ and $J_{\delta_*} := \Phi_{\text{blk}} + F_{\delta_*}$. Then*

$$J_{\delta_*}(\eta) < 0 \quad (0 < \eta < 1).$$

The inequality is uniform on every compact subinterval of $(0, 1)$.

Proof. Set $\eta_{\text{blk}} := 1 - 2^{-1/2}$. On $(0, \eta_{\text{blk}}]$, the derivative of J_{δ_*} is increasing, so J_{δ_*} is convex and $J_{\delta_*}(0) = 0$. Outward interval arithmetic gives

$$J_{\delta_*}(\eta_{\text{blk}}) \in [-0.074030, -0.074029].$$

Convexity proves strict negativity on the first region. On $[\eta_{\text{blk}}, \eta_{\text{conv}}(\delta_*)]$, the entropy terms cancel and the remaining expression is increasing. Its right endpoint lies in $[-0.011076, -0.011075]$. On the final region,

$$J_{\delta_*}(\eta) = H_2(\eta) + H_2(\delta_*) - \frac{3}{2} \leq H_2(\delta_*) - \frac{1}{2},$$

and the last value lies in $[-0.000023719, -0.000023718]$. These strict intervals prove the claim. $\square$

The remaining issue is uniformity as $\eta \downarrow 0$. Define

$$\eta_* := \frac{1}{5000}, \quad \chi := \frac{101}{100}, \quad \gamma := \frac{51}{50}.$$

Set

$$\begin{aligned} a_* &:= \frac{\eta_*}{\delta_*(\chi + 1)}, & b_* &:= \frac{1 + a_*}{2}, \\ \epsilon_* &:= \frac{\chi^2 a_*}{(\chi + 1)(1 - \chi a_*)} + \frac{\eta_*}{2(1 - \eta_*)}, & \rho_* &:= \delta_*(\chi + 1)e^{\epsilon_*}. \end{aligned}$$

Lemma 11 (Sparse-weight closure). *Let $m_N := \lceil \gamma \log_2 N \rceil$. There are constants C, N_0 such that, for $N \geq N_0$ and $1 \leq h \leq \eta_* N$,*

$$p_{N,h}(\delta_*) \leq CN^{2\gamma} \sqrt{h} \rho_*^h. \quad (53)$$

If B_N is even, divides N , and satisfies $B_N \geq 17 \log_2 N$, then

$$\sum_{h=1}^{\lfloor \eta_* N \rfloor} \mathbb{E}[A_{N,h}^{\text{out}}] p_{N,h}(\delta_*) = o(1). \quad (54)$$

Proof. For $\eta := h/N$, evaluate (44) at

$$a := \frac{\eta}{\delta_*(\chi + 1)}, \quad S := \frac{1 - \chi a}{1 + a}.$$

The active-state row sum equals $1 - (\chi - 1)a/2 < 1$. The Perron-root identity then bounds the matrix power by $CN^{2\gamma}$ uniformly over the stated range. A standard lower bound on $\binom{N}{h}$ gives (53); the exponent correction increases with η and is at most ϵ_* . Interval arithmetic certifies

$$\begin{aligned} \rho_* &\in [0.2212639483446276, 0.2212639483446278], \\ \gamma \log_2(1/b_*) - 1 &> 0.0186697. \end{aligned}$$

In particular, $\rho_* < \sqrt{2} - 1$.

The exact independent-block outer law gives

$$\sum_{h \leq \eta_* N} \mathbb{E}[A_{N,h}^{\text{out}}] p_{N,h}(\delta_*) \leq CN^{2\gamma+1/2} \left[(1 + G_{B_N,1/2}(\rho_*))^{N/B_N} - 1 \right].$$

Using (32), the right side is bounded by a constant times

$$\frac{N^{2\gamma+3/2}}{B_N} \left(\frac{1 + \rho_*}{\sqrt{2}} \right)^{B_N}.$$

Finally, outward arithmetic gives

$$17 \log_2 \left(\frac{\sqrt{2}}{1 + \rho_*} \right) - (2\gamma + 3/2) > 0.0576243.$$

Thus the last expression tends to zero. $\square$

A.4 The Bazzi–Mahdian–Spielman Question

When Does One Interleaver Suffice? Bazzi, Mahdian, and Spielman (BMS) show that a depth-two serial concatenation cannot be asymptotically good when the outer has constant memory and the inner has sublinear memory [1, Sec. III-A]. Their conclusion asks whether allowing the outer memory to grow logarithmically with block length permits an asymptotically good code [1, Sec. IV]. Here, asymptotically good means that both rate and relative minimum distance stay bounded away from zero. The question concerns the composition of an outer encoder, one interleaver, and an inner encoder.

The SPIN Progression. Accumulator SPIN supplies the basic positive construction in the block-encoder setting. Its outer blocks have length $\Theta(\log N)$, and its inner accumulator carries one bit of state. At rate $1/2$, Theorem 5 gives relative distance greater than 0.0037 with probability $1 - o(1)$. Thus a growing inner state is unnecessary for linear distance once the outer blocks grow logarithmically. Random SPIN increases the inner memory to $\Theta(\log N)$ and proves relative distance greater than 0.11002. Structured SPIN then returns to constant recursive state: with better constituents and structured interleaving, it proves relative distance greater than 0.11 and linear-time encoding. The Random and Structured SPIN distance guarantees are both close to the rate- $1/2$ GV value of approximately 0.110028.

Logarithmic Outer Memory Suffices. The block description admits a streaming realization with logarithmic memory and a short termination. For a rate- $1/2$ outer with block length b , buffer one block while emitting the preceding block, at two output bits per input bit. Initially the output buffer contains b zeros. After the message, $b/2$ fixed zero input bits flush the last block. This produces $0^b \| O(x)$ using $O(b + \log N)$ memory, including buffers and a position counter. Extend the interleaver to send these startup zeros to the end and apply the original permutation to $O(x)$. The inner therefore receives $\Pi(O(x)) \| 0^b$. Since each inner is causal, the first N output bits are exactly the original SPIN codeword. The added outputs cannot reduce absolute minimum distance. For $b = \Theta(\log N)$, the rate becomes $1/2 - o(1)$ and the relative-distance guarantee changes by only $o(1)$.

This gives a positive answer in the terminated-encoder interpretation of the BMS question: logarithmic outer memory and a one-bit accumulator suffice for an asymptotically good code. BMS mention termination but omit it from their displayed encoder definition [1, Sec. I-A]; the fixed termination inputs are explicit in the realization above. The structured interleaver is still one permutation between the two encoders; permutations internal to the outer act only within its blocks.

B Asymptotic Structured SPIN Certificate

B.1 Certificate for the Scalable Family

This appendix supplies the detailed estimates for the proof of Theorem 2 in Section 6. The argument first selects one Golay BAA constituent whose realized spectrum obeys a certified envelope. It then reduces the structured route to occupation and mean row weight. The final first-moment bound separates four regimes: $Q = 1, 2$, fixed $Q \geq 3$, growing sparse Q , and positive relative occupation. Finite IMT transfers handle the last two regimes. Continuum transfers handle $Q = 1, 2$ and fixed $Q \geq 3$.

For $0 \leq d \leq N$ and $1 \leq Q \leq L$, let $Z_{d,Q}$ count nonzero messages with exactly Q active outer blocks and encoded weight at most d . Unless a display states otherwise, expectations and probabilities include the shared BAA constituent, the route permutations, and the IMT transvections. The finite-length specialization is not part of this probability space.

The One-Sampled Golay BAA Outer We begin with the exact expected spectrum before selecting the shared constituent. Let

$$G(u) := 1 + 759u^8 + 2576u^{12} + 759u^{16} + u^{24}$$

be the weight polynomial of the extended Golay code, and define

$$G_b(a) := [u^a]G(u)^{b/24}.$$

For $a \geq 1$, put $r_a := \lceil a/2 \rceil$ and

$$T_b(a, c) := \binom{c-1}{r_a-1} \binom{b-c}{a-r_a}, \quad P_b(a, c) := \frac{T_b(a, c)}{\binom{b}{a}}. \quad (55)$$

We set $T_b(0, 0) := 1$ and $T_b(0, c) := 0$ for $c \neq 0$. Binomial coefficients outside their natural range are zero. Thus $P_b(a, c)$ is the exact weight-transition probability of a uniformly interleaved accumulator.

For a realized BAA constituent $\mathcal{O}_b$, let $A_{\mathcal{O}_b}(w)$ count its nonzero words of weight w , and define

$$\bar{A}_b(w) := \mathbb{E}_{\tau_1, \tau_2} [A_{\mathcal{O}_b}(w)].$$

Lemma 12 (Exact expected BAA spectrum). *For every $0 \leq w \leq b$,*

$$\bar{A}_b(w) = \sum_{a=1}^b \sum_{c=0}^b G_b(a) P_b(a, c) P_b(c, w). \quad (56)$$

Proof. Fix a Golay direct-sum word of weight a . The first uniform permutation and accumulator produce weight c with probability $P_b(a, c)$. Conditioned on that weight, the second permutation makes the word uniform on its Hamming slice. The second accumulator then produces weight w with probability $P_b(c, w)$. Summing over the $G_b(a)$ input words proves (56). $\square$

We use this formula to bound both tails and the central spectrum. The tails require a separate estimate for messages supported on few Golay blocks; a limiting exponent alone does not control those messages. Let

$$h(x) := -x \ln x - (1-x) \ln(1-x)$$

with its continuous endpoint values. Define

$$g(a) := \inf_{u>0} \left\{ \frac{1}{24} \ln G(u) - a \ln u \right\}.$$

On the accumulator feasibility region $a/2 \leq c \leq 1 - a/2$, set

$$\pi(a, c) := ch\left(\frac{a}{2c}\right) + (1-c)h\left(\frac{a}{2(1-c)}\right) - h(a),$$

and set $\pi(a, c) := -\infty$ outside that region. The limiting exponent is

$$a_{\text{BA}}(x) := \sup_{0 \leq a, c \leq 1} \{g(a) + \pi(a, c) + \pi(c, x)\}. \quad (57)$$

Uniform coefficient and Stirling bounds in (56) give

$$\frac{1}{b} \ln \bar{A}_b(w) \leq a_{\text{BA}}(w/b) + O\left(\frac{\ln b}{b}\right). \quad (58)$$

Lemma 13 (BAA outer tails). *As $b \rightarrow \infty$ through multiples of 24,*

$$\sum_{w/b \notin [0.104, 0.896]} \bar{A}_b(w) = o(1).$$

The expectation is over the two independent outer permutations τ_1, τ_2 .

Proof. Put $\delta_o := 13/125$ and $\eta := 1/1000$. We split nonzero messages by the number j of active Golay blocks. There are at most

$$N_j := \binom{b/24}{j} 4095^j \leq \left(\frac{4095eb}{24j}\right)^j$$

such messages. Their input weight to the first accumulator is even, say $a = 2\ell$, with $4j \leq \ell \leq 12j$.

First consider $j \leq \eta b$. For a uniform weight- v input V , the transition formula (55) bounds the lower-tail probability by

$$p_v := \sum_{0 \leq h \leq \delta_o b} P_b(v, h) \leq c_o^{\lfloor v/2 \rfloor}, \quad c_o := \frac{4\delta_o}{1 - \delta_o} = \frac{13}{28}. \quad (59)$$

Indeed, for $v = 2\ell$ and $D := \lfloor \delta_o b \rfloor$, summing $\binom{h-1}{\ell-1}$ over $h \leq D$ gives

$$p_{2\ell} \leq \frac{\binom{b}{\ell} \binom{D}{\ell}}{\binom{b}{2\ell}} = \binom{2\ell}{\ell} \frac{\binom{D}{\ell}}{\binom{b-\ell}{\ell}} \leq c_o^\ell.$$

If $\ell > D$, the probability is zero. The odd case has an additional factor at most $2\delta_o/(1 - \delta_o) < 1$. The bound also holds for $v = 0$.

Set $\zeta := \sqrt{c_o}$ and $\kappa := 4\zeta/(1 - \zeta)$. Let H_1, H_2 be the weights after the first and second accumulators. Conditional on H_1 , the second permutation makes its input uniform; thus (59) bounds the lower-tail probability by $\zeta^{-1}\mathbb{E}[\zeta^{H_1}]$. For a fixed first input of weight 2ℓ ,

$$\begin{aligned}\mathbb{E}[\zeta^{H_1}] &\leq \frac{\binom{b}{\ell}}{\binom{b}{2\ell}} \sum_{h \geq \ell} \binom{h-1}{\ell-1} \zeta^h \\ &= \frac{\binom{b}{\ell}}{\binom{b}{2\ell}} \left(\frac{\zeta}{1-\zeta} \right)^\ell \leq \left(\frac{\kappa\ell}{b-2\ell+1} \right)^\ell.\end{aligned}\tag{60}$$

For $\ell/b \leq 12\eta$, the last expression decreases with ℓ . To check this, its logarithmic derivative is at most $\ln(\kappa x/(1-2x)) + 1 + 2x/(1-2x)$ at $x = 12\eta$, which is below -1.22 . Consequently, every message with $j \leq \eta b$ satisfies

$$\mathbb{E}[\zeta^{H_1}] \leq \left(\frac{4\kappa j}{b(1-8\eta)} \right)^{4j}.$$

Summing over these messages gives

$$S_b := \sum_{1 \leq j \leq \eta b} \left[C_* \left(\frac{j}{b} \right)^3 \right]^j, \quad C_* := \frac{4095e}{24} \left(\frac{4\kappa}{1-8\eta} \right)^4.\tag{61}$$

The expected number of sparse lower-tail words is at most $\zeta^{-1}S_b$. Outward interval arithmetic gives $C_*\eta^3 < 0.657$. For $j \leq \sqrt{b}$, the summands are bounded by a geometric series with base $C_*b^{-3/2}$; the remaining summands are at most 0.657^j . Hence $S_b = O(b^{-3/2})$.

The sparse upper tail needs an explicit bound as well. Let $e_1 = (1, 0, \dots, 0)$. Since $\text{Acc}(V) + \mathbf{1} = \text{Acc}(V + e_1)$, conditioning on the first bit of a uniform weight- v input gives

$$\mathbb{P}[\text{wt}(\text{Acc}(V)) \geq (1 - \delta_o)b] \leq \frac{v}{b-v+1}p_{v-1} + \frac{b-v}{v+1}p_{v+1}.$$

At $v = b$ the second term is absent. Each prefactor is at most b . Using (59), the right side is at most $b(\zeta^{-2} + 1)\zeta^v$ for $v \geq 1$. Apply this bound with $v = H_1$, which is nonzero for a nonzero input. Equation (61) bounds the expected number of sparse upper-tail words by $b(\zeta^{-2} + 1)S_b = O(b^{-1/2}) = o(1)$.

It remains to bound messages with $j > \eta b$. Write $\alpha := a/b$, $\beta := H_1/b$, and $x := H_2/b$. Then $\alpha \geq 8\eta = 0.008$. Accumulator support constraints imply $x \geq \alpha/4 \geq 0.002$. For the lower tail, our interval verifier [25] proves

$$g(\alpha) + \pi(\alpha, \beta) + \pi(\beta, x) < -7.68 \cdot 10^{-8}\tag{62}$$

at every feasible point with $\alpha \in [0.008, 1]$, $\beta \in [0, 1]$, and $x \in [0.002, 0.104]$. The branch-and-bound check processes 6,747 boxes and leaves no unresolved boxes.

Uniform coefficient and Stirling bounds apply to each summand of (56); the polynomial number of weights and the $O(\ln b)$ remainder preserve exponential decay. This argument restricts the initial weight before taking a supremum: the unrestricted exponent a_{BA} also includes sparse paths.

For the dense upper tail, reflect x to $1 - x$ and use $\pi(\beta, x) = \pi(\beta, 1 - x)$. The finite support constraints can differ from the limiting feasibility region by $O(1/b)$ because of parity. The entropy terms are uniformly continuous, so these shifts contribute $o(1)$ to the normalized exponent. The strict negative gap in (62) therefore also gives exponential decay for the dense upper tail. Combining the sparse and dense estimates proves the lemma. $\square$

The monotonicity and geometric constants and the interval check of (62) are checked by the same verifier. Lemma 2 now selects the shared outer. We next bound its spectrum on $\mathcal{W} = [0.104, 0.896]$.

The certificate represents a concave majorant $\hat{a}_{\text{BA}}$ as the lower envelope of rational affine supports. It proves

$$a_{\text{BA}}(x) \leq \hat{a}_{\text{BA}}(x) \quad (x \in \mathcal{W}). \quad (63)$$

The majorant has 39 affine segments. It contains the supports of the 31-segment majorant of Appendix B.1 and five further left-half supports with slopes 0.09, 0.07, 0.05, 0.03, and 0.01, together with their reflections. Each of these ten intercepts is an outward rational upper bound on $\ln(1 + e^{-s}) - (\ln 2)/2 + 10^{-5}$ for its slope s . This expression proposes the line; validity follows from outward checks of (57) over its entire active interval. The check of these supports accepts 5,363 leaves after 10,721 box checks, with largest upper residual below $-1.70 \cdot 10^{-9}$. Exact partition replay and higher working precision confirm every leaf. Reflection uses $\pi(c, x) = \pi(c, 1 - x)$, not a symmetry of each realized constituent. Adding supports only shrinks the active interval of each 31-segment support, so its validity checks carry over. The central support is an outward rational rounding of the exact code-size bound $(\ln 2)/2$. Taking the lower envelope preserves concavity. Equations (58) and (63) give the outer exponent used below. The certificate records all affine supports and box witnesses.

Routing and the Inner Estimates Lemma 3 and its proof are in Section 6. The estimates below use its notation: Q active rows, $\alpha = Q/L$, mean relative row weight x , and $q = \alpha x$. The concavity bound (24) reduces the selected spectrum to that mean.

The Selected IMT Maps The state dimension is $s = 19$ and the step width is $t = 128$ throughout this appendix. Table 4 specifies both maps in (13). Bit i of each hexadecimal word is coordinate i , starting with the least significant bit. Write $M := 2^{19} - 1$. The nonzero image weights of A and their multiplicities are

$$(d_i)_{i=1}^5 = (48, 56, 64, 72, 80), \quad (c_i)_{i=1}^5 = (5166, 110288, 293455, 110128, 5250).$$

Table 4. Fixed IMT expansion and feedback maps. The words specify $A(e_j)$ and $C^\top(e_j)$ for the standard state basis.

j	$A(e_j)$	$C^\top(e_j)$
0	55555555555555555555555555555555	800003d868100423210234004341a004
1	33333333333333333333333333333333	010d0102568441441a6212120884488c
2	0f0f0f0f0f0f0f0f0f0f0f0f0f0f0f0f	40482008104050000312449182000a43
3	ff00ff00ff00ff00ff00ff00ff00ff00	c60682141218101b9a382140d1011330
4	0000ffff0000ffff0000ffff0000ffff	02800a0201093ca2cc121a20200c0408
5	ffffff00000000ffffff0000000000	81212844806ef00068055044102ee101
6	ffffffffffffffff0000000000000000	1e200091046309f08018083800400061
7	ca9ff6a3c56f06ac0653c590f65cca60	91750062141381019100a60c30a810aa
8	8e8e718e2bd4d4d4b2b24db217e8e8e8	01043040c08a882804068c12246500d0
9	b2b2b2b218e7e71842bdbd42e8e8e8e8	2089148c8300089c8504044184424100
10	a00a05af366393c6396c633650fa0aa0	044340810132201506e1012001322008
11	9350635fa063506c5f9c506c93509ca0	42001d281e80022101ec18224403cc12
12	12e2747b121d7484de2e4748212eb848	2834da0440600280021100010b9a9780
13	4411e14b11bb4b1e2d87772287d22288	2118a12209240f06640041cacb0884c5
14	59953003cfca66a0cc0655665560cc0	80122401c000804c00c9821294803a04
15	7447b88b74b8b874b7847b4848848448	6880cc1f224046004080818c3814003b
16	11ddd21e447787b42d1e112278b44488	1cf284802d0ca24a000820040c904700
17	72e48214e48deb82d8b1d7be4ed8be28	7c405360a885745070a0c0a122702002
18	4e824e82e428e4284e824e82e428e428	128a40312091018038456bcd40011874

Every coordinate of A is a nonzero linear functional. The 128 columns of C are distinct nonzero vectors of weight five and span $\mathbb{F}_2^{19}$. The interval verifier of Appendix B.1 checks these properties and both spectra by exact enumeration.

Finite IMT Transfers We represent a weighted state measure by seven non-negative coordinates $(Z, D, S_1, \dots, S_5)$. The Z coordinate is mass at zero. The D coordinate bounds arbitrary mass on nonzero states. Coordinate S_i dominates S_i times the uniform probability measure on $\{q : \text{wt}(Aq) = d_i\}$. Terminal mass is bounded by the sum of coordinates. Rows of a transfer matrix index the entering coordinate, so weighted measures propagate as row vectors. Write e_Z for the zero-state unit row vector and $\mathbf{1}$ for the all-ones column vector.

First fix an input weight j . Let $n_j := \binom{128}{j}$, let k_j be the number of weight- j vectors in $\ker C$, and put $\nu_j := 1 - k_j/n_j$. For $z \in (0, 1)$, the emitted moment at image weight d is

$$m_{d,j}(z) := \frac{1}{n_j} \sum_h \binom{d}{h} \binom{128-d}{j-h} z^{d+j-2h}.$$

Set $m_{D,j} := \max_i m_{d_i,j}$. For $q \neq 0$, define the lazy cancellation moment

$$L_j(q; z) := \frac{1}{n_j} \sum_{\substack{\text{wt}(X)=j \\ CX=q}} z^{\text{wt}(X+Aq)}.$$

Choose upper bounds $\ell_{D,j} \geq \max_{q \neq 0} L_j(q; z)$ and $\ell_{i,j} \geq c_i^{-1} \sum_{\text{wt}(Aq)=d_i} L_j(q; z)$. Let f_j bound the size of every nonzero weight- j syndrome fiber. Valid choices

are the minima of the following bounds:

$$\begin{aligned}\ell_{D,j} &:= \min \left\{ m_{D,j}, \nu_j, \frac{f_j}{n_j} z^{\min_i |d_i - j|} \right\}, \\ \ell_{i,j} &:= \min \left\{ m_{d_i,j}, \frac{\min(n_j - k_j, c_i f_j)}{c_i n_j} z^{|d_i - j|} \right\}.\end{aligned}$$

For $j = 1, 2$ the verifier also uses the exact cancellation sums, enumerating the input supports and grouping them by CX and $\text{wt}(ACX)$. No relation between A and $C^\top$ is assumed.

The integer fiber bounds require only the spectrum of $C^\top$. Let b_w count its nonzero words of weight w , and let $\mathcal{K}_j(w)$ be the binary Krawtchouk polynomial. Fourier inversion and Parseval give

$$k_j = 2^{-19} \left(n_j + \sum_w b_w \mathcal{K}_j(w) \right), \quad V_j = 2^{-19} \left(n_j^2 + \sum_w b_w \mathcal{K}_j(w)^2 \right),$$

where V_j is the sum of squares of all syndrome fiber sizes. Put $a_j := n_j - k_j$. A nonzero fiber is at most each of

$$\begin{aligned}a_j, \quad & \left\lfloor 2^{-19} \left(n_j + \sum_w b_w |\mathcal{K}_j(w)| \right) \right\rfloor, \\ & \left\lfloor \frac{a_j + \lceil \sqrt{(M-1)(M(V_j - k_j^2) - a_j^2)} \rceil}{M} \right\rfloor.\end{aligned}$$

The last bound follows by minimizing the other $M - 1$ squares for a fixed maximum fiber. There is also a constant-weight packing bound. If d_C is the kernel distance, set $r = \lfloor (d_C - 1)/2 \rfloor$ and $h = \min(j, 128 - j)$. The bound is 1 for $h < r$ and $\lfloor \binom{128}{h-r} / \binom{h}{r} \rfloor$ otherwise: two words in one fiber cannot contain the same $(h - r)$ -subset after complementing when needed. The verifier takes the minimum of these integer upper bounds.

Define the fixed-weight transfer $T_j(z)$ by the following nonzero entries:

$$\begin{aligned}(T_j)_{ZZ} &= (1 - \nu_j) z^j, & (T_j)_{ZD} &= \nu_j z^j, \\ (T_j)_{DZ} &= \frac{1}{2} \ell_{D,j} + \frac{1}{2M} \min(m_{D,j}, \nu_j), & (T_j)_{DD} &= \frac{1}{2} m_{D,j}, \\ (T_j)_{DS_i} &= \frac{c_i}{2M} m_{D,j}, & (T_j)_{S_i Z} &= \frac{1}{2} \ell_{i,j} + \frac{1}{2M} \min(m_{d_i,j}, \nu_j), \\ (T_j)_{S_i S_h} &= \frac{c_h}{2M} m_{d_i,j}.\end{aligned}$$

Additionally put mass $m_{d_i,j}/2$ in entry $S_i D$ when $\nu_j > 0$, or add it to entry $S_i S_i$ when $\nu_j = 0$.

These entries dominate the weighted state measure. The zero row is exact. The lazy branch pays its cancellation bound and places the remaining mass in D ; if all syndromes vanish, it preserves the entering shell. On the refresh branch, every specified target has probability at most $1/M$. Termination also requires a nonzero syndrome, giving the minimum in the zero column. Keeping the

full surviving-mass upper bounds deliberately overcounts cancellation; it never subtracts an unproved lower bound. For iid Bernoulli- β input, the occupation transfer is

$$T_{\text{occ}}(\beta, z) := \sum_{j=0}^{128} \binom{128}{j} \beta^j (1-\beta)^{128-j} T_j(z).$$

A second envelope bounds every target by Fourier inversion of a tilted input law. Set

$$\begin{aligned} g_0 &:= 1 - \beta + \beta z, & g_1 &:= \beta + (1 - \beta)z, & F_d &:= g_0^{128-d} g_1^d, \\ r_0 &:= |1 - 2\beta z/g_0|, & r_1 &:= |1 - 2(1 - \beta)z/g_1|, \end{aligned}$$

and define

$$H_d := 2^{-19} \left(1 + \sum_w b_w \max_{\max(0, d+w-128) \leq h \leq \min(d, w)} r_0^{w-h} r_1^h \right).$$

The maximum is attained at an endpoint. Tilting by $z^{\text{wt}(X+AQ)}$ keeps the input coordinates independent. Fourier inversion bounds each syndrome point probability by H_d . Thus every target, including zero, receives weighted mass at most $F_d(H_d/2 + 1/(2M))$. Define T_F by placing this value in column Z and c_h times this value in column S_h for an entering shell S_i with $d = d_i$. For an entering D , maximize over d_i . The zero row of T_F is the exact binomial mixture of the zero rows of T_j .

Finally the state-independent scalar envelope is

$$T_{\text{sc}}(\beta, z) := \max_{d \in \{0, 48, 56, 64, 72, 80\}} F_d.$$

At $\beta = 1/2$ it equals $((1+z)/2)^{128}$ exactly. For either matrix envelope, induction gives

$$\mathbb{E}[z^{\text{wt}(Y)}] \leq e_Z T(\beta, z)^{N/128} \mathbf{1}. \quad (64)$$

The scalar bound gives $T_{\text{sc}}^{N/128}$. A witness selects one complete envelope; the proof never takes an entrywise minimum of representations.

Positive Occupation Fix $\alpha \geq 10^{-4}$ and mean active-row weight $x \in \mathcal{W}$. All arrays in a fixed profile have $N\alpha x$ ones. Apply route domination to the moment restricted to that weight slice, then change its iid reference probability from αx to py . The likelihood cost is $\exp(ND_{\text{KL}}(\alpha x \| py))$. The chain rule bounds its exponent by $D_{\text{KL}}(\alpha \| p) + \alpha D_{\text{KL}}(x \| y)$. Consequently the bad-message exponent per output bit is at most

$$\alpha \hat{a}_{\text{BA}}(x) + D_{\text{KL}}(\alpha \| p) + \alpha D_{\text{KL}}(x \| y) + \frac{\ln \lambda}{128} - 0.11 \ln z, \quad (65)$$

where λ is a scalar bound or a positive Collatz bound $T(py, z)\mathbf{w} \leq \lambda \mathbf{w}$.

The verifier covers $[10^{-4}, 1] \times \mathcal{W}$ by 1,023 rational boxes within the 39 outer segments. It fixes a witness on each box and checks four vertices. The exponent is convex in $(\alpha, \alpha x)$ on each affine segment. All bounds pass outward arithmetic at 256 bits and replay at 512 bits, with maximum less than $-4.10335 \cdot 10^{-7}$. The witnesses use 433 occupation transfers, 307 Fourier transfers, and 283 scalar transfers. The values p, y and the positive vectors are rational; the weight tilt is specified by a rational $\log(-\log z)$ and enclosed outward. Exact rational geometry checks containment, disjoint interiors, and total area. No numerical eigenvalue is trusted.

There are finitely many witnesses, so their moment prefactors are bounded. Counting and route-domination costs are $\exp(o(N))$ uniformly in this range. They include active-position choices, row-weight tuples, and the selected spectrum's polynomial factors. The summed first moment is therefore $o(1)$.

Uniform Sparse Occupation For sparse profiles, the factor $(L+1)^b$ would be too costly. We instead compare the counting measure on each permuted active outer row with fair bits. The coarser 31-segment outer majorant $\hat{a}_{31}$ already proves

$$\sup_{x \in \mathcal{W}} (\hat{a}_{31}(x) - h(x) + \ln(2)/2) < \ell, \quad \ell := 0.01281.$$

Hence, uniformly over all selected outers and weights, for some $\epsilon_b = O(\log(b)/b)$,

$$\frac{A_{\mathcal{O}_b}(w)}{\binom{b}{w}} \leq \exp\{b(-\ln(2)/2 + \ell + \epsilon_b)\}. \quad (66)$$

Relative to a fair b -bit row, the counting measure thus costs $\exp\{b(\ln(2)/2 + \ell + \epsilon_b)\}$. This comparison is made after summing messages; it does not assert that a fixed constituent word is uniform.

Fix Q active positions and put $\alpha = Q/L \leq 10^{-4}$. Under the fair-row reference, each region has Q uniform marked positions carrying independent fair bits. Sample iid marks with probability $p = (8/5)\alpha$ and condition their count to equal Q . Without that conditioning, all input bits are iid with $\beta = p/2 = (4/5)\alpha$. The reference regions are independent. Since Q is a mode of $\text{Bin}(L, \alpha)$, Chebyshev's inequality gives modal mass at least $1/(8\sqrt{Q})$. Indeed, probability at least $3/4$ lies in at most $4\sqrt{Q} + 1 \leq 5\sqrt{Q}$ integer locations about the mean, so a mode has mass at least $3/(20\sqrt{Q})$. The exact binomial likelihood ratio yields

$$\mathbb{P}[\text{Bin}(L, p) = Q] \geq \frac{e^{-LD_{\text{KL}}(\alpha||p)}}{8\sqrt{Q}}.$$

Put $z = 1 - (8/5)\alpha$. An exact rational polynomial certificate proves

$$T_{\text{occ}}(\beta, z)\mathbf{w}(\alpha) \leq (1 - 96\alpha)\mathbf{w}(\alpha), \quad 0 < \alpha \leq 10^{-4}. \quad (67)$$

Here $w_Z = 1$, and $w_i = 2^{-10}(1 + h_i\alpha)$ for the six live coordinates, with

$$h_D = 900, \quad h_{S_1} = \frac{116184452864}{2257055535}, \quad h_{S_2} = \frac{92653723520}{3613910291},$$

$$h_{S_3} = -\frac{26853143552}{769273207925}, \quad h_{S_4} = -\frac{461216096896}{18043337105}, \quad h_{S_5} = -\frac{351546952448}{6881266875}.$$

The shell-weighted average of the corrections is zero. Every coordinate is at least $1/2048$. In the variable $x = 10000\alpha$, the verifier constructs degree-257 rational polynomial upper bounds for the seven residuals. It certifies over $[0, 1]$ that each polynomial substituted for a maximum in a residual dominates that maximum. Each residual vanishes at zero. After removing that factor, its constant coefficient plus the sum of its positive remaining coefficients is strictly negative. This proves the whole interval, including arbitrarily small positive α .

Combining (66), the conditioning bound, (67), and Markov's inequality gives

$$\mathbb{E}[Z_{\lfloor 0.11N \rfloor, Q} \mid \mathcal{G}_b] \leq 2048 \binom{L}{Q} e^{Qb(\ln(2)/2 + \ell + \epsilon_b)} (8\sqrt{Q})^b \cdot e^{ND_{\text{KL}}(\alpha \| p)} (1 - 96\alpha)^{N/128} z^{-0.11N}. \quad (68)$$

Use $\ln \binom{L}{Q} \leq Q \ln L + Q$, $b \geq c \log_2 N$ with $c = 39/4$, and $\ln(1 - u) \leq -u$. The uniform coefficient per active-row bit is at most

$$C_* := \frac{\ln 2}{2} + \ln(5/8) + \frac{3/5 + 0.11(8/5)}{1 - 0.00016} - \frac{96}{128} + \ell + \frac{\ln 2}{c} < -0.01340384.$$

The remaining terms are $\epsilon_b + 1/b + \ln(8\sqrt{Q})/Q + \ln(2048)/(Qb)$. For $Q \geq 4096$, the third is below 0.002 and decreases with Q . For sufficiently large b , the other terms together are below 0.004 uniformly. Thus

$$\mathbb{E}[Z_{\lfloor 0.11N \rfloor, Q} \mid \mathcal{G}_b] \leq e^{-0.006Qb}, \quad 4096 \leq Q \leq 10^{-4}L. \quad (69)$$

This explicit cutoff avoids any assumption about how fast Q grows.

Fixed Occupation and the Continuum Limit It remains to control finitely many $Q < 4096$. We derive a limit for each fixed Q , allowing constants to depend on Q . Define

$$r := 1/M, \quad p_0 := 2^{18}/M, \quad P_0 := \begin{pmatrix} 0 & 1 \\ r & 1 - r \end{pmatrix}.$$

During empty steps, whose t input bits are all zero, a live state has transition $P = (I + \Pi)/2$, where Π replaces it by a uniform nonzero state. Write $f(q) := \text{wt}(Aq)$. Its stationary mean is $\mu = 128p_0$. For any initial live state,

$$\mathbb{E}[f(q_j) \mid q_i] = \mu + 2^{-(j-i)}(f(q_i) - \mu), \quad j \geq i.$$

For $S_g := \sum_{i=0}^{g-1} f(q_i)$ this implies

$$|\mathbb{E}S_g - \mu g| \leq 2t, \quad \text{Var}(S_g) \leq 3t^2g, \quad \mathbb{E}|S_g - \mu g| \leq t\sqrt{3g} + 2t.$$

The variance bound follows from $\text{Cov}(f(q_i), f(q_j)) = 2^{-(j-i)} \text{Var}(f(q_i))$. It does not require a stationary initial state.

Fix $\theta > 0$, and set $z_L = e^{-\theta/L}$ and $\gamma = p_0\theta$. Let $W_0(z_L)$ be the full-state weighted empty-step kernel. The Lipschitz bound for e^{-x} and $P^g = 2^{-g}I + (1 - 2^{-g})II$ give, for every live q, q' ,

$$\left| W_0(z_L)^g(q, q') - \frac{e^{-\theta\mu g/L}}{M} \right| \leq \frac{\theta t(\sqrt{3g} + 2)}{L} + 2^{-g}. \quad (70)$$

The zero state stays zero. Define the lift J by $J(q, 0) = \mathbf{1}_{q=0}$, $J(q, 1) = \mathbf{1}_{q \neq 0}$, and the projection R by $R(0, q) = \mathbf{1}_{q=0}$, $R(1, q) = \mathbf{1}_{q \neq 0}/M$. Thus $RJ = I_2$. Put $D_\gamma(u) := \text{diag}(1, e^{-\gamma u})$. For $\lceil 3 \log_2 L \rceil \leq g \leq L/t$, equation (70) approximates W_0^g by $JD_\gamma(tg/L)R$ with entry error $O(L^{-1/2})$.

Let $W_1(z_L)$ be the kernel for one input bit at a uniformly chosen local position. Its difference from $W_1(1)$ is $O(1/L)$ in row norm, since the emitted weight is at most t . Invertibility of the mixer and the nonzero columns of C give the exact identity

$$RW_1(1)J = P_0.$$

For a fixed $a \leq Q$ uniformly placed impulses in one region, two impulses in one step have probability $O_Q(1/L)$. A boundary or inter-impulse gap shorter than $\lceil 3 \log_2 L \rceil$ has probability $O_Q(\log(L)/L)$. On the complement, replace each empty gap using (70) and each impulse by $W_1(1)$. The finite products telescope with bounded row norms. The omitted impulse durations total $at/L = O_Q(1/L)$. Riemann sums for the ordered uniform sites then give

$$K_a(\theta) := a! \int_{\substack{u_i \geq 0 \\ \sum_{i=0}^a u_i = 1}} D_\gamma(u_0) P_0 D_\gamma(u_1) \cdots P_0 D_\gamma(u_a) du. \quad (71)$$

For $a = 0$, $K_0 = D_\gamma(1)$. The simplex integral uses a independent coordinates. The region kernel $F_{a,L}(z_L)$ differs from $JK_a R$ by $O_{Q,M,\theta}(L^{-1/2} + \log(L)/L)$ in every entry, uniformly over initial and final states. Bad placement events cost at most their probability because weighted kernels have row sums at most one.

The additive comparison becomes multiplicative. For $a = 0$, both zero/live cross entries vanish exactly. For $a = 1$, zero-to-zero vanishes exactly. All other coarse entries are positive, and for $a \geq 2$ every entry is positive. Since Q, M, θ are fixed, there exists $\eta_L = O_{Q,M,\theta}(L^{-1/2} + \log(L)/L)$ such that

$$F_{a,L}(z_L) \leq (1 + \eta_L) JK_a(\theta) R \quad (0 \leq a \leq Q). \quad (72)$$

This inequality averages over the within-region placements. It is uniform over the choices of which active rows contribute a bit and over boundary states. Multiplication over $b = O(\log L)$ regions costs $\exp(o(1))$, also when coefficient extraction enforces the row weights.

We may enlarge P_0 to $P_+ := \begin{pmatrix} 0 & 1 \\ r & 1 \end{pmatrix}$ in (71); write K_a^+ for the resulting upper matrices. They are nonnegative, but not stochastic.

One and Two Active Rows. Use (66) and fair reference rows. The continuum region matrix is $2^{-Q}T_Q$, with $T_Q := \sum_{a=0}^Q \binom{Q}{a} K_a^+$. Set $\gamma = Q \ln 2$ and $\theta = \gamma/p_0$.

With

$$a_0 = e^{-\gamma}, \quad f = \frac{1-a_0}{\gamma}, \quad g = \frac{2(\gamma-1+a_0)}{\gamma^2}, \quad e = \frac{2(1-(1+\gamma)a_0)}{\gamma^2},$$

the matrices are

$$K_0^+ = \text{diag}(1, a_0), \quad K_1^+ = \begin{pmatrix} 0 & f \\ rf & a_0 \end{pmatrix}, \quad K_2^+ = \begin{pmatrix} rg & e \\ re & re + a_0 \end{pmatrix}.$$

The exponent per outer bit, including active-position choices, is

$$\ln \rho(T_Q) + 0.11\theta - Q \ln(2)/2 + Q\ell + Q \ln(2)/c.$$

Outward checks at 256 bits, replayed at 512 bits, bound this expression divided by Q by -0.10918 for $Q = 1$ and -0.10914 for $Q = 2$. Finite positive-matrix prefactors and the uniform outer remainder are $\exp(o_Q(b))$, so both first moments tend to zero.

A Weight-Coupled Bound for Fixed $Q \geq 3$. For positive row fugacities u_j , define

$$T_Q(\mathbf{u}, \theta) := \sum_{\mathbf{e} \in \{0,1\}^Q} \left(\prod_j u_j^{e_j} \right) K_{\sum_j e_j}^+(\theta).$$

Nonnegative coefficient extraction and (72) bound the moment for row weights w_j by

$$\left(\prod_j \binom{b}{w_j}^{-1} u_j^{-w_j} \right) e_0^\top T_Q(\mathbf{u}, \theta)^b \mathbf{1}_2 \exp(o_Q(b)).$$

Set $\sigma = 127/250$, $\tau = 133/125$, $v_1 = 3/1600$, and $\theta = Q\tau/p_0$. Define

$$G_Q := \sup_{0 \leq y \leq 1} \{-\tau y + (1 + 1/Q) \ln(1 - y + y/\sigma)\},$$

$$m_v(u) := \max\{1 + u\sigma v_1, ur/v_1 + \sigma(1 + u)\}.$$

For the weighted row norm with $v = (1, v_1)^\top$,

$$\|T_Q(\mathbf{u}, Q\tau/p_0)\|_v \leq e^{QG_Q} \prod_j m_v(u_j), \quad G_Q \leq G_3. \quad (73)$$

To prove it, insert all Q potential impulse sites, using I for an unused site and P_+ for a used site. If a fixed class path has l live spacings, their total duration Y has distribution $\text{Beta}(l, Q + 1 - l)$, with deterministic endpoint cases. The identity

$$\mathbb{E}[(1 - Y + Y/\sigma)^{-(Q+1)}] = \sigma^l$$

follows by the substitution $u = y/[\sigma + (1 - \sigma)y]$ in the beta integral. By the definition of G_Q , the path's integrated weight is at most $e^{QG_Q} \sigma^l$. Summing paths yields the average over orders π of

$$D_\sigma(I + u_{\pi(1)}P_+)D_\sigma \cdots (I + u_{\pi(Q)}P_+)D_\sigma,$$

times e^{QG_Q} , where $D_\sigma = \text{diag}(1, \sigma)$. Submultiplicativity, $\|D_\sigma\|_v \leq 1$, and $\|(I + uP_+)D_\sigma\|_v = m_v(u)$ prove (73). This reasoning requires nonnegative coefficients, not stochasticity of P_+ .

The 31-segment majorant $\hat{a}_{31}$ suffices here. For each segment choose one rational $u > 0$. Convexity in x reduces the following inequality to its two endpoints:

$$\hat{a}_{31}(x) - h(x) - x \ln u + \ln m_v(u) + G_3 + 0.11\tau/p_0 + \ln(2)/c < -0.0008679786.$$

All 62 outward checks pass. The row-weight tuples and finite coefficient remainders contribute $o_Q(b)$. Therefore, for each fixed $Q \geq 3$,

$$\mathbb{E}[Z_{\lfloor 0.11N \rfloor, Q} \mid \mathcal{G}_b] \leq \exp\{-0.0008679Qb + o_Q(b)\}.$$

Completion and Certificate Boundary The proof of Theorem 2 in Section 6 combines these regimes and accounts for ordinary and transposed work.

Theorem 2 uses only the probability space and inequalities stated in Section 6 and this appendix. The certificates for the outer tails, the 31- and 39-segment majorants, and the IMT estimates, including the 512-bit replay of the positive-occupation cover, are documented with their verification commands [25]; the verifier sources and hash-bound arithmetic witnesses lie outside the core encoder artifact. The numerical certificates check the finite spectra, interval covers, Collatz inequalities, exact sparse polynomials, and fixed-occupation constants. The analytic steps are the measure comparisons, transfer induction, continuum limit, and final union proved above. Higher-precision replay is not an independent proof of those steps. The finite BCH-256 certificate of Appendix C uses the same transfer interface, with separate finite-length evidence.

Formal Verification The Lean development defines the binary encoder and setup distribution used in Theorem 2. It proves the distance statement (19). It also proves that every setup realization has rate $1/2$. One pair of BAA permutations is shared by all outer rows; the route permutations and IMT transvections are sampled independently of that pair. The final probability is over this original setup distribution. Conditioning on $\mathcal{G}_b$ is internal to the proof.

The proof includes the shared outer spectrum, the routed IMT process, the three occupation regimes, and the final first-moment argument. Lean checks both the analytic arguments and the required numerical inequalities. The encoding-work claim, subsequent padding corollary, and separate finite-length BCH results are outside the final checked statement.

The formal proof uses some weaker intermediate bounds than those displayed above. Its positive-occupation certificate covers all 1,023 indexed boxes with margin $4 \cdot 10^{-7}$, rather than the sharper numerical margin $4.10335 \cdot 10^{-7}$. For each fixed positive occupation Q , and all sufficiently large family indices m , it gives, with $b = b_m$ and $N = N_m$,

$$\mathbb{E}[Z_{\lfloor 0.11N \rfloor, Q} \mid \mathcal{G}_b] \leq \begin{cases} 1000e^{-b/50}, & Q = 1, \\ 1000e^{-b/20}, & Q = 2, \\ (1600/3)e^{-Qb/2000}, & Q \geq 3. \end{cases}$$

The threshold index may depend on Q . For $Q = 1, 2$, the formalization uses the tilts $\theta = 2, 4$, respectively. For larger fixed occupations it reserves a fixed positive margin in the continuum transfer. These bounds suffice for the same limit; they do not establish every sharper intermediate estimate above. Fixed- Q convergence is summed only over the finite set $1 \leq Q < 4096$. The growing sparse range uses its separate uniform bound.

The public import is `SpinCodes.Native`. The module `SpinCodes.NativePin` exposes the final statements and their recursive axiom audits, which report only `propext`, `Classical.choice`, and `Quot.sound`. The development’s reproduction guide, `FINAL_REPRODUCTION.md`, describes the verification commands and their scope.

C Finite BCH-256 Certificate

This appendix connects the fixed encoder to the numerical inequalities in Theorem 3. The argument has three interfaces: a deterministic outer spectrum bound, a positive inner transfer, and a complete sum over outer occupations. Computational witnesses supply finite upper bounds at these interfaces; they do not replace the probability comparisons.

C.1 The Fixed Outer and Its Spectrum

Represent $\mathbb{F}_{256}$ in the polynomial basis with modulus $X^8 + X^6 + X^3 + X^2 + 1$ (hexadecimal 14d), and let γ be the element represented by X . For $\delta \in \{37, 39\}$, let g_δ be the binary minimal polynomial product for the roots $\gamma, \dots, \gamma^{\delta-1}$. Write `ext` for even parity extension from 255 to 256 coordinates. Explicitly, the outer is

$$\mathcal{B} = \text{span}_{\mathbb{F}_2}(\{\text{ext}(X^i g_{39}) : 0 \leq i < 123\} \cup \{\text{ext}(X^j g_{37}) : 0 \leq j < 5\}).$$

The polynomials, with coefficient i stored in bit i , are

$$\begin{aligned} g_{37} & \text{ 1c3b42fe115aa90747020d79ac738adb} \\ g_{39} & \text{ 1f3126ff25af19628f815685cb83da770f.} \end{aligned}$$

The quotient g_{39}/g_{37} is primitive irreducible of degree eight. Thus the five added residues are independent modulo the code generated by g_{39} . The outer has dimension $123 + 5 = 128$, is contained in $\mathcal{P}$, and contains $\mathcal{Q}$ (the codes $\mathcal{P} \supset \mathcal{Q}$ of Section 7). Even parity and the BCH bound give $d(\mathcal{B}) \geq 38$. The all-ones word belongs to $\mathcal{Q}$, hence to $\mathcal{B}$.

Lemma 14 (Common spectrum of the intermediate codes).

Every dimension-128 code between $\mathcal{Q}$ and $\mathcal{P}$ has the same weight spectrum. If q_w counts weight- w words of $\mathcal{Q}$ and h_w counts them in one nonzero $\mathcal{Q}$ -coset in $\mathcal{P}$, then

$$A_w(\mathcal{P}) = q_w + 255h_w, \quad A_w(\mathcal{B}) = q_w + 31h_w. \quad (74)$$

Proof. A cyclic shift on the punctured coordinates acts on $\mathcal{P}/\mathcal{Q}$ as multiplication by X modulo g_{39}/g_{37} . Primitivity makes this action transitive on its 255 nonzero residues. Parity extension preserves this weight-preserving action, so all nonzero cosets have the same spectrum. Each intermediate dimension-128 code is a union of $\mathcal{Q}$ and 31 nonzero cosets. $\square$

This lemma connects the implemented generators to the certificate's definition of the outer by syndrome subspaces. It does not identify the two coordinate codes. Independent row permutations make their identical spectra sufficient for the same first-moment bounds.

Spectrum Inequalities. Write $A_w := A_w(\mathcal{B})$. Evenness, minimum distance, and complementation give $A_0 = A_{256} = 1$, $A_w = A_{256-w}$, and nonzero weights restricted to

$$\mathcal{W} := \{38, 40, \dots, 218\} \cup \{256\}.$$

The bound inputs comprise individual caps $A_w \leq a_w$ and a joint inequality on the three lowest complementary shell pairs. They are derived from linear constraints, rather than a model for the unknown spectrum.

The constraints use the exact endpoint spectra of Fujiwara and Kusaka [15]: the spectrum of the extended BCH code of dimension 71 and the dual spectrum of the extended BCH code of dimension 187. Generator containments and MacWilliams transforms relate these known spectra to $\mathcal{P}$, $\mathcal{Q}$, and their duals. The polynomial

$$\mathcal{K}_j(w) := \sum_v (-1)^v \binom{w}{v} \binom{256-w}{j-v}$$

gives the transform $A_j(D^\perp) = |D|^{-1} \sum_w A_w(D) \mathcal{K}_j(w)$ for a binary linear code D . Containment gives coefficientwise inequalities between the corresponding spectra.

Two additional sources sharpen this relaxation. First, the hulls $R_{\mathcal{P}} = \mathcal{P} \cap \mathcal{P}^\perp$ and $R_{\mathcal{Q}} = \mathcal{Q} \cap \mathcal{Q}^\perp$ have dimensions 85 and 93, with

$$R_{\mathcal{P}} \subset R_{\mathcal{Q}} \subset \mathcal{Q}, \quad \mathcal{P} \cap R_{\mathcal{Q}}^\perp = \mathcal{Q}.$$

The nonzero cosets of $R_{\mathcal{P}}$ in $R_{\mathcal{Q}}$ also have a common spectrum. Writing their spectra as r and s gives $A(R_{\mathcal{Q}}) = r + 255s$. For example, the displayed intersection and MacWilliams identity imply

$$h \leq 2^{-93} \mathcal{K}(r-s)$$

coefficientwise. This is a set-difference argument, not subtraction of unrelated upper bounds. Signed Fourier transforms of $(-1)^{\text{wt}(c)/2}$ on the even codes give further linear inequalities through their hulls. Our verification code derives these constraints and checks the requisite binary ranks and intersections.

Second, a finite Fourier case cover proves $d(\mathcal{Q}^\perp) \geq 30$. For a punctured dual word, partition cases by its first nonzero Fourier cyclotomic coset. In each case, a root run or an independent-set witness bounds the support size. In the latter

witness, Frobenius and coordinate scaling preserve independence; appending a vector with nonzero coordinate sum to vectors with zero coordinate sums increases the rank. The final rank cannot exceed the support size. The exceptional case is split exhaustively on a further Fourier value. Parity and affine invariance transfer the resulting bound to the extended dual.

Consequently, projection of $\mathcal{Q}$, and of every $\mathcal{Q}$ -coset, onto any 29 coordinates is uniform. Otherwise an annihilating linear functional would give a nonzero dual word of weight at most 29. Thus

$$\sum_w q_w \mathcal{K}_j(w) = \sum_w h_w \mathcal{K}_j(w) = 0 \quad (1 \leq j \leq 29).$$

These moment equations also hold for $\mathcal{B}$. For example, squaring the degree-14 reproducing kernel for the binomial measure gives the cap

$$A_w \leq \frac{2^{128}}{\sum_{j=0}^{14} \mathcal{K}_j(w)^2 / \binom{256}{j}}. \quad (75)$$

Indeed, normalize that kernel to equal one at w ; its square is nonnegative, has degree 28, and has binomial expectation equal to the reciprocal denominator. The moment identities transfer that expectation to a uniform word of $\mathcal{B}$.

The caps take the minimum of such bounds, constant-weight packing bounds, and exact linear-programming witnesses. The joint relaxation has 196 nonnegative variables and 1163 normalized rows. Its checker reconstructs the rows and verifies rational dual feasibility and the objective bound. In particular, if $Fv \leq b$, $v \geq 0$, and $y \geq 0$ satisfies $F^\top y \geq c$, then $c^\top v \leq b^\top y$. Equality rows have unrestricted multipliers. This exact test, not a solver's floating feasibility tolerance, supplies the inequality consumed below. The published endpoint spectra remain mathematical inputs; empirical spectrum estimates do not.

C.2 Selected IMT Maps and Positive Transfers

The half-rate construction uses exactly the maps in Table 4. In particular, the image of A does not contain the all-ones word; the feedback columns have weight five. No field-multiplier update or identity $CA = 0$ is used. The seven-state fixed-weight transfers $T_j(z)$ in Appendix B.1 apply without a limiting argument. They distinguish zero, arbitrary nonzero mass, and five uniform image-weight shells. Their cancellation bounds use the feedback map C separately from the expansion A .

Fix an output tilt $z \in (0, 1)$ and write $E_j(z) := T_j(z)$. For a shuffled region of L bits, form

$$\mathcal{E}(u, z) := \sum_{j=0}^{128} \binom{128}{j} E_j(z) u^j, \quad R_j(z) := \frac{[u^j] \mathcal{E}(u, z)^{L/128}}{\binom{L}{j}}.$$

Conditional on its weight, the region is a uniform slice. The binomial factors count its stepwise supports, proving this coefficient formula. Multiplication retains the inner state across steps and regions; the matrices need not commute. From the zero initial state, the conditional bad-word probability is at most $z^{-H} e_Z R_{j_1} \cdots R_{j_{256}} \mathbf{1}$, where e_Z is a row vector. Truncation above degree q cannot change any coefficient through degree q .

For dense input laws the verifier may instead use the complete occupation or Fourier moment bound of Appendix B.1. Every selection bounds the same terminal moment; an entrywise minimum of different measure representations would not be justified.

C.3 One Active Outer Row

A row of weight w reaches a uniform w -subset of the 256 regions. Its single position in each reached region is independently uniform. Use the same E_0, E_1 above, including the exact weight-one cancellation data for the selected independent maps. Put $\ell = L/t$, $R_0 = E_0^\ell$, and $R_1 = \ell^{-1} \sum_{i=0}^{\ell-1} E_0^i E_1 E_0^{\ell-1-i}$. The coefficient bound including the L possible active row positions is

$$c_w(z) := \frac{Lz^{-H}}{\binom{256}{w}} e_Z[u^w](R_0 + uR_1)^{256} \mathbf{1}.$$

Let $Z_{H,q}$ count bad messages with exactly q active rows and put $F_q := \mathbb{E} Z_{H,q}$. Each shell may choose its own certified tilt; write c_w for the resulting upper bound. Then $F_1 \leq \sum_w c_w A_w$.

Let $J = \{38, 40, 42\}$. The outer witness proves $\sum_{w \in J} b_w^* A_w \leq B^*$ for fixed positive rational b_w^* . Complement symmetry and nonnegativity imply

$$F_1 \leq B^* \max_{w \in J} \frac{c_w + c_{256-w}}{b_w^*} + \sum_{w \in \mathcal{W} \setminus (J \cup (256-J))} c_w a_w. \quad (76)$$

This reuses a weighted outer inequality, not a probability computed for another inner. All c_w are evaluated for the selected IMT instance.

C.4 Sparse Occupations and Constant-Row Compositions

Partition ordinary weights into twelve bands:

$$\begin{aligned} & [38, 42], [44, 48], [50, 54], [56, 64], [66, 78], [80, 100], \\ & [102, 128], [130, 154], [156, 176], [178, 198], [200, 210], [212, 218], \end{aligned}$$

retaining only even weights. Keep the all-ones word as a thirteenth label. For an ordinary band g , choose $p_g \in (0, 1)$ and define

$$\Gamma_g = \max_{w \in g} \frac{a_w}{\binom{256}{w} p_g^w (1-p_g)^{256-w}}, \quad \rho_g \geq \Gamma_g^{1/256}.$$

The counting measure of a randomly permuted row in band g is pointwise dominated by Γ_g times product Bernoulli(p_g) measure. Indeed, at a support of weight w its mass is $A_w / \binom{256}{w}$. The comparison makes different regions independent before their inner states are propagated. Split Γ_g into 256 equal factors, one per region.

For fixed occupation q , start with $V_j^{(0)} = R_j$ for $0 \leq j \leq q$. For $0 \leq r < q$ and $0 \leq j < q - r$, set

$$V_j^{(r+1)} = \max_g \rho_g ((1 - p_g) V_j^{(r)} + p_g V_{j+1}^{(r)}).$$

The maximum is entrywise and includes the exact label $(p_*, \rho_*) = (1, 1)$. Induction dominates every fixed sequence of band choices. There are 13^q such sequences, giving

$$F_q \leq \binom{L}{q} 13^q z^{-H} e_Z(V_0^{(q)})^{256} \mathbf{1}. \quad (77)$$

An entrywise maximum may select different bands for different entries. That only enlarges the envelope; it is not a claim that the actual rows make those adaptive choices.

For $K = 2^{16}, 2^{18}$, a sharper composition split keeps h all-ones rows exact and applies the recurrence only to d ordinary rows. Initialize $V_j^{(0)} = R_{h+j}$ for $0 \leq j \leq d$ and apply d recurrence steps, maximizing over the twelve ordinary bands. The composition contribution is bounded by

$$U_{d,h} = \binom{L}{d} \binom{L-d}{h} 12^d z^{-H} e_Z(V_0^{(d)})^{256} \mathbf{1}.$$

For $d = 0$ there is no density cost. Summing every $h = 0, \dots, q$ with $d = q - h$ covers occupation q . Witness choices may vary with the composition. Only complete, checked sums enter the final union bound.

C.5 Dense Occupations by a Change of Input Law

The dense bound compresses the ordinary/all-ones compositions into occupation and mean reference density. It needs the following pointwise comparison.

Lemma 15 (Shuffled Bernoulli comparison). *Let $n \geq 1$ and let S be a sum of n independent Bernoulli variables with mean nr . For $D_n := \min\{n+1, 4\lceil\sqrt{n}\rceil\}$ and $0 \leq j \leq n$,*

$$\mathbb{P}[S = j] \leq D_n \binom{n}{j} r^j (1-r)^{n-j}.$$

The same factor compares the uniformly shuffled bit vector with n independent Bernoulli(r) bits.

Proof. The logarithm of the moment generating function is bounded by that of $\text{Bin}(n, r)$, by concavity in each success probability and Jensen's inequality. Chernoff's bound gives $\mathbb{P}[S = j] \leq \exp(-n D_{\text{KL}}(j/n \| r))$. The binomial mass

on the right without D_n equals this exponential times the mass $b_{n,j}$ at the mean of $\text{Bin}(n, j/n)$. That mean is a mode, so $b_{n,j} \geq 1/(n+1)$. The latter binomial has variance at most $n/4$; Chebyshev places at least $3/4$ of its mass within $\sqrt{n}$ of its mean. There are at most $2\sqrt{n} + 1$ integers there, giving $b_{n,j} \geq 3/[4(2\sqrt{n} + 1)] \geq 1/(4\sqrt{n})$. The endpoint cases are deterministic. Finally, divide the weight probabilities by $\binom{n}{j}$ to compare uniformly shuffled vectors. $\square$

Fix reference coordinates $\xi_g \in (0, 1)$ for the twelve ordinary labels, and $\xi_* = 1$. These coordinates do not change across the density cover. For an ordered assignment a of q labels, let $\nu_a = q^{-1} \sum_i \xi_{a_i}$ and $\vartheta_a = q\nu_a/L$. Choose an input tilt $x > 0$, with row probabilities $p_g = \xi_g/[x + (1-x)\xi_g]$, and define

$$C_g(x) := \max_{w \in g} \frac{a_w x^w}{\binom{256}{w} \xi_g^w (1 - \xi_g)^{256-w}}, \quad C_*(x) = x^{256}, \quad B(\vartheta, x) = 1 - \vartheta + \vartheta/x.$$

Changing from row probabilities p_g to ξ_g multiplies the density of a region of weight j by a row-normalization factor and x^{-j} . The shuffle preserves j . Apply Lemma 15, then absorb x^{-j} into independent Bernoulli inputs of probability $r = \vartheta/[x + (1-x)\vartheta]$. The product of row costs and normalizers is exactly $\prod_i C_{a_i}(x) B(\vartheta_a, x)^N$.

For $\lambda > 0$, use a valid seven-state IMT transfer to define

$$T(r, z) := \sum_{j=0}^t \binom{t}{j} r^j (1-r)^{t-j} E_j(z), \quad \mathcal{M}(r, \lambda) := e^{\lambda H} e_Z T(r, e^{-\lambda})^{N/t} \mathbf{1}.$$

The fixed-assignment first moment is at most $D_L^{256} \prod_i C_{a_i}(x) B(\vartheta_a, x)^N \mathcal{M}(r, \lambda)$. For an auxiliary real slope η , put $S(x, \eta) = \sum_g C_g(x) e^{-\eta \xi_g}$, including the all-ones label. Expanding S^q sums all ordered assignments. Therefore, for a density interval I , let $U_{q,I}$ be the contribution to $\mathbb{E}Z_H$ from assignments with q active rows and $\nu_a \in I$. Then

$$U_{q,I} \leq \binom{L}{q} D_L^{256} S(x, \eta)^q \cdot \sup_{\nu \in I} \left\{ e^{\eta q \nu} B(q\nu/L, x)^N \mathcal{M}\left(\frac{q\nu/L}{x + (1-x)q\nu/L}, \lambda\right) \right\}. \quad (78)$$

Every interval can choose its own witnesses. Fixed coordinates ξ_g ensure that the intervals still cover every assignment. At $q = L, \nu = 1$, the all-ones cost cancels the normalization exactly; no ordinary-band multiplicity is charged to that word.

Finite Rectangle Certificates. The checker fixes $r \in (0, 1)$ within each rectangle and chooses $x(\vartheta) = \vartheta(1-r)/(r(1-\vartheta))$. Since x now depends on ν , it applies the label sum separately at each distinct ν , and pays the composition count $\binom{q+12}{12}$. This factor is necessary; the fixed- x bound cannot simply be reused.

For $u = \log x$, the function $\log S(e^u, \eta)$ is convex, because each band cost is a maximum of exponentials and log-sum-exp preserves convexity. A secant upper

line $a + bu$, with $0 \leq b \leq 256$, bounds it on the interval. After substitution, the scalar exponent becomes at most

$$qa + \eta q\nu - qb \log \frac{r}{1-r} - N \log(1-r) + qb \log \vartheta + (N - qb) \log(1 - \vartheta).$$

Both logarithm coefficients are nonnegative. Replacing the logarithms by tangent upper bounds at an interior ϑ_0 leaves a function affine in ν and convex in q : its quadratic coefficient in q is $b\nu L^{-1}(\vartheta_0^{-1} + (1 - \vartheta_0)^{-1}) \geq 0$. Thus its maximum occurs at a rectangle corner. For intervals ending at $\vartheta = 1$, use slope $b = 256$ and extend by continuity at the feasible endpoint $q = L, \nu = 1$.

A finite partition tree covers every integer q and every reference density in its declared range. Each leaf includes the number of integer occupations it covers, an upper bound on the composition factor, and the fixed- r matrix bound. The exact union sums all leaf bounds. The leaf evaluations use the factor $(L+1)^{256}$; the checker multiplies their bounds by the exact rational factor $(D_L/(L+1))^{256}$. This changes only the density comparison, not any transfer or cover.

Short-Length Refinements. The two shortest instances need sharper bounds, not a different encoder. First, activation from zero can retain uniform-shell mass. For a weight- j input, let f_j bound every nonzero syndrome fiber. A valid complete zero row retains $E_{ZZ} = k_j z^j / \binom{128}{j}$ and sends $a_v f_j z^j / \binom{128}{j}$ to the shell of size a_v , with no arbitrary component. Here k_j counts weight- j kernel words and a_v counts states with image weight v . For Bernoulli input, let b_w count nonzero weight- w words in the image of $C^\top$. Fourier inversion similarly bounds each weighted nonzero syndrome by

$$h = 2^{-19}(1 - r + rz)^{128} \left(1 + \sum_w b_w \left| 1 - \frac{2rz}{1 - r + rz} \right|^w \right).$$

Keeping the exact zero entry and sending $a_v h$ to each shell is valid. The verifier compares complete moment bounds, not their individual entries.

Second, the factor D_L can be replaced by a smaller composition-sensitive bound. Fix an assignment of reference probabilities ξ_i (zero for inactive rows), with mean ϑ . To compare its shuffled weight- j probability to $\text{Bin}(L, \vartheta)$, assume $0 < \vartheta < 1$ and $0 < j < L$, and put $y = j/L$ and $a = y(1 - \vartheta)/(\vartheta(1 - y))$. Exponential tilting gives the exact ratio

$$\frac{\prod_i (1 - \xi_i + \xi_i a)}{(1 - \vartheta + \vartheta a)^L} \frac{\mathbb{P}[\sum_i X'_i = j]}{\mathbb{P}[\text{Bin}(L, y) = j]}, \quad \mathbb{P}[X'_i = 1] = \frac{\xi_i a}{1 - \xi_i + \xi_i a}.$$

Here the X'_i are independent Bernoulli variables. Concavity bounds the product in the numerator by $(1 - \nu + \nu a)^q$. If the tilted variance is at least V , Fourier inversion gives the point-mass bound $e^{-V} I_0(V)$, where $I_0(V) := \pi^{-1} \int_0^\pi e^{V \cos \phi} d\phi$. Indeed, the absolute characteristic function is at most $\exp(-2V \sin^2(\phi/2))$, and integration proves the claim. This bound decreases with V .

For an entire (q, ν) rectangle, minimize each tilted variance $\xi(1 - \xi)a/(1 - \xi + \xi a)^2$ over the allowed tilt interval. Its minimum is at an endpoint. A finite

linear program then minimizes the average of these lower bounds subject to the allowed mean reference density. Multiplication by the minimum q gives a uniform V . Maximizing the preceding likelihood ratio over integer j gives a uniform routing factor for the rectangle. Endpoint weights are bounded directly. This calculation is also valid with a restricted set of labels.

Third, fixing x throughout a rectangle permits the label sum $S(x, \eta)^q$ in (78) without the composition-count factor required by the varying- x calculation. The iid probability r then varies with ϑ . To keep the moment and normalizer coupled, fix the setup-averaged nonnegative function $f(u) = \mathbb{E}_\theta[z^{\text{wt}(I_\theta(u))}]$ on N -bit inputs and put $M(r) = \mathbb{E}_{U \sim \text{Bern}(r)^N} f(U)$. For $D(\vartheta) = 1 - \vartheta + \vartheta/x$,

$$D(\vartheta)^N M\left(\frac{\vartheta}{x + (1-x)\vartheta}\right) = \sum_u f(u) x^{-\text{wt}(u)} \vartheta^{\text{wt}(u)} (1 - \vartheta)^{N - \text{wt}(u)}.$$

Thus, as a function of $v = \log(\vartheta/(1 - \vartheta))$,

$$F(v) = N \log D(\vartheta) + \log M(r(\vartheta)) + \eta L \vartheta \quad \text{satisfies} \quad F''(v) \geq -(N + |\eta L|)/4.$$

Indeed, the logarithm of the positive exponential sum is convex, $(\log(1 + e^v))'' \leq 1/4$, and $|\vartheta''(v)| \leq 1/4$. The resulting secant bound is

$$F(v) \leq \max(F(v_0), F(v_1)) + (N + |\eta L|)(v_1 - v_0)^2/32.$$

Outward endpoint moment bounds can replace $F(v_0), F(v_1)$. The verifier adds $q \log S + \log \binom{L}{q} + \lambda H$, sums over the integers q in the leaf, and applies its routing factor.

Finally, split assignments with no high-reference label from those with at least one, taking a label to be high when $\xi_g > 3/4$. The first part uses S_{low} and the restricted routing comparison. For the second part, choose $\zeta \leq 0$, use $S_{\text{low}} + e^{-\zeta} S_{\text{high}}$ and multiply by e^ζ . If $h \geq 1$ labels are high, $e^{\zeta h} \leq e^\zeta$ proves this upper bound after expanding the label sum. An analogous split isolates the all-ones label. The two subset bounds are added before comparison with an unsplit bound. No separately rounded normalizers are subtracted. Full derivations of the rectangle optimizations and their exact-arithmetic tests accompany the code [25].

C.6 The Quarter-Rate Outer and Feedback

Over $\mathbb{F}_{128}$ with modulus $X^7 + X + 1$ and primitive element X , let p and q generate the primitive length-127 BCH codes of designed distances 31 and 43. In coefficient-bit notation,

$$p = \text{d5306d6bfdbc8574719e70d}, \quad q = \text{5106dae17a61e520c606a4e29}.$$

The quarter-rate constituent is the span of the parity extensions of $X^i q$ for $0 \leq i < 29$ and $p, Xp, X^2 p$. The quotient $q/p = \text{d5}$ is primitive of degree seven. The same coset-orbit argument as above proves rank 32 and spectrum

$$W_{32} = W_{29} + \frac{7}{127}(W_{36} - W_{29}).$$

Table 5. Quarter-rate feedback map: bit i of the listed word is coordinate i of $C^\top(e_j)$.

j	$C^\top(e_j)$
0	808019020040c0800400032500101090
1	01008021009102020030601020308206
2	020006020e0200220400001205680121
3	00100002410a18010108190188020009
4	006040080801003c2000900080604442
5	e1060040101080201088844080801000
6	a0000010020800002064008920020c34
7	4010b19080200090406201800c000000
8	409802c8300001880104400220c40000
9	2808000401442400804120405408a000
10	044140048004110008004c0612100048
11	02202118200840020003264008080210
12	10060025440204005004808802004248
13	00a0aa01990002400290001042002800
14	0001548000c0110c020002284001d800
15	15000c0000240644950a000000020082
16	00000060001128016a000824000121a1
17	0843000024a008508011100001840100
18	1e0c00004200e0010880000011050404

Here W_{29}, W_{36} are the published weight distributions of the length-128 extended BCH codes of dimensions 29 and 36; the code documentation records their source [25]. The resulting spectrum has minimum weight 32, multiplicity 588 at weight 32, total mass 2^{32} , and a single all-ones word.

Use A from Table 4 and C from Table 5. Its columns have weight three. The fixed-weight and Fourier transfers apply with this map's own dual spectrum, kernel counts, fiber caps, and weight-one/two cancellation sums. The quarter-rate calculation uses the exact outer spectrum, so no BCH-256 spectrum inequality is transferred to it.

For the two quarter-rate cutoffs, the verifier separately sums all outer weight classes for $q = 1$, band compositions for $q = 2$, and region bounds for $3 \leq q \leq q_{\max}$. It covers the remaining integer types (zero, ordinary-weight bands, and all-ones rows) by a disjoint dense partition. Here $q_{\max} = 64$ for $\delta = 0.165$ and 128 for $\delta = 0.19$. Every dense type uses a product reference law and a complete IMT moment; the type likelihood ratio is bounded at the vertices of its box. The union covers $q = 1, \dots, 32768$ at each cutoff. The dense comparison can be written directly in terms of these types. Let n_g count rows of label g , so $\sum_g n_g = L$, and put $J(\mathbf{n}) = L! / \prod_g n_g!$. Let Γ_g dominate the row counting measure by Bernoulli(p_g) bits, as in the sparse calculation. Include zero and all-ones labels with $(\Gamma_0, p_0) = (1, 0)$ and $(\Gamma_*, p_*) = (1, 1)$. Choose a positive probability vector $\boldsymbol{\pi}$ on the labels and put $r = \sum_g \pi_g p_g$. Independent labels in a region have type $\mathbf{n}$ with probability $J(\mathbf{n}) \prod_g \pi_g^{n_g}$. Conditioning on this type gives the permuted reference rows exactly. Pay that conditioning cost in each of the $B = 128$ regions (one per outer coordinate) and count the $J(\mathbf{n})$ outer assignments once. If $\mathcal{U}(r, z)$ upper-bounds the complete IMT moment, the

contribution of this type is at most

$$z^{-H} \mathcal{U}(r, z) J(\mathbf{n})^{1-B} \prod_g (\Gamma_g \pi_g^{-B})^{n_g}.$$

Its logarithm is convex in $\mathbf{n}$ for a fixed witness: the only nonlinear term is $(B-1) \sum_g \log \Gamma(n_g + 1)$, which is convex. Thus vertices of a box intersected with $\sum_g n_g = L$ suffice. Multiply its maximum by an upper bound on the number of integer types in the box, then sum the disjoint boxes. This is an auxiliary input-law comparison, not additional randomness in the encoder setup. A binary64 search only proposes witnesses; the bounds come from outward evaluation, higher-precision replay, and the exact-union checker.

C.7 Complete Finite Margins

Tables 6 and 7 give the complete numerical results for the two finite theorems in Section 7. Their margins are certified bounds on setup failure, not full application-security levels.

Table 6. Complete finite certificates for the selected (128, 19) IMT inner. Margins are rounded for display; the theorem uses exact rational upper bounds.

$\log_2 K$	outer rows	L	cutoff	$H - \log_2 U_K$ (bits)
16	512	13 107		41.818327
18	2 048	52 428		50.189076
20	8 192	209 715		50.062088
22	32 768	838 860		48.390492
24	131 072	3 355 443		46.457255

Table 7. Two distance thresholds for the same quarter-rate encoder at $K = 2^{20}$. The exact unions, not the rounded margins, establish the targets.

δ	cutoff	$\lfloor \delta N \rfloor$	target	λ	margin (bits)
0.165	692 060		40		41.048168
0.190	796 917		30		30.033491

C.8 Certificate Completion

Table 8 gives the component margins of the half-rate exact unions. Here $q_{\max}$ is the last sparse occupation. In each case the checked occupation set is exactly

Table 8. Component margins of the half-rate exact unions, in bits. Each entry is $-\log_2$ of the bound on one part of the union: one active row ($q = 1$), the sparse occupations ($2 \leq q \leq q_{\max}$), and the dense occupations ($q > q_{\max}$). Values are rounded for display.

$\log_2 K$	$q_{\max}$	$q = 1$	$2 \leq q \leq q_{\max}$	$q > q_{\max}$
16	63	43.592834	60.988751	42.317109
18	127	50.189181	63.938924	79.903280
20	511	50.062272	63.000000	78.909867
22	511	48.390495	67.000000	84.999997
24	511	46.457257	66.000000	191.512160

$\{1, \dots, L\}$. Summing the three rational bounds and applying (27) proves (26). The quarter-rate unions prove Theorem 4 by the same first-moment inequality. Injectivity follows independently from sequential inversion of the inner and injectivity of the block-diagonal outer.

Verification of the five half-rate instances and the quarter-rate certificate is documented with the code [25]. The verification records bind the selected maps, code parameters, numerical inputs, and verifier sources.

All selected bounds have passed 512-bit replay, followed by source authentication, exact coverage checks, and rational summation. The $K = 2^{16}$ dense partition has 1096 leaves. Directed arithmetic and upward dyadic conversion supply the numerical bounds; floating-point search scores do not enter the unions. Higher-precision replay shares producer formulas, so it is not an independent proof implementation.

For the timing measurements, the records also check map identity, the outer implementation, correctness tests, and the identities of the measured sources and binaries. These verification records are separate from the core encoder artifact, which does not distribute them or the raw measurements. This is a computer-assisted proof, not a proof-assistant formalization.

D Parameter Selection and Engineering Curves

This appendix supports the finite constructions in Section 7. It is intended for choosing and implementing other parameter sets: the plots separate diagnostic bounds from the complete certificates at the selected operating points.

D.1 How the Parameters Affect the Margin

Consider a rate-1/2 outer constituent of length B , repeated over $L = 2K/B$ rows to encode K message bits. The output length is $N = 2K$. The IMT inner processes t bits per step, carries an s -bit state, and applies one transvection per update. We ask how to choose t and s as K changes, rather than prescribing one inner for every length. Throughout the plots, the bad-word cutoff is $H = \lfloor N/10 \rfloor$. For an upper bound U on the expected number of nonzero messages with output

weight at most H , call $-\log_2 U$ its *margin* in bits. A full bound includes every number q of active outer rows. Write U_1 for an upper bound on the contribution from messages with exactly one active row, and call $-\log_2 U_1$ the *Q1 margin*. We use Q1 to explore parameter sensitivity, then compare it with full certificates at five selected operating points in Appendix D.2. Q1 alone does not control distance failure: messages with two or more active rows remain to be bounded.

Figure 1 centers this comparison on the BCH-derived $[256, 128]$ constituent and implemented $(t, s) = (128, 19)$ maps specified in Section 7.1. Two smaller exact-spectrum constituents provide outer-size comparisons: a shortened BCH-derived $[64, 32, 12]$ code and an extended BCH $[128, 64, 22]$ code, called BCH-64 and BCH-128. Their fixed-step curves use $(t, s) = (64, 20)$. For each fixed curve, an ideal mixing reference replaces every transvection by an independently sampled uniform invertible linear map. Every fixed nonzero state then has a uniform nonzero image; zero stays zero. This *full refresh* keeps the outer, routing, expansion, and feedback fixed. Its Q1 bound is a reference, not an absolute outer-only ceiling.

Why the Length Curve Turns Down. The Q1 sum contains L possible active rows. Once the bounds on conditional low-weight probabilities stabilize, doubling K , and hence L , costs approximately one margin bit. The larger-length portions of Figure 1 follow this trend. At short lengths, however, the conditional probabilities also change: a region contains only $h := L/t = 2K/(Bt)$ inner steps. At $K = 2^{16}$, the BCH-256 fixed-step configuration has $h = 4$, compared with $h = 64$ at $K = 2^{20}$. Increasing B at fixed K, t also reduces h : a stronger outer can require a different inner calibration, not just a vertical shift of the curve.

One mechanism is a short excursion away from state zero. The first input bit injects a nonzero state; a later bit can cancel it before much weight has been emitted. A transvection retains any fixed nonzero incoming state with mixture weight $1/2$. After d empty updates, the retained component therefore has weight 2^{-d} ; it is not a fresh uniform state. Consider two adjacent regions, each containing one input bit, with zero state before the first. Let their input positions be independent and uniform, and suppose the feedback columns are distinct and nonzero. The probability that the state returns to zero just after the second input is

$$p_{\text{cancel}} = \frac{\rho_h}{t} + \frac{1 - \rho_h}{2^s - 1}, \quad \rho_h := \frac{2(1 - 2^{-h})^2}{h^2}.$$

The first term comes from repeating the same feedback column while the state is retained. Increasing s reduces the second term, not the first. For large h , the first term is approximately $2t/L^2$. These retained cancellations favor inputs near the shared region boundary: their mean separation approaches three epochs even as h grows. They can therefore produce short output pulses, rather than benefiting from the typical number of mixing updates. This is a local cancellation probability, not a whole-code distance bound; the emitted weights and all activation patterns still matter.

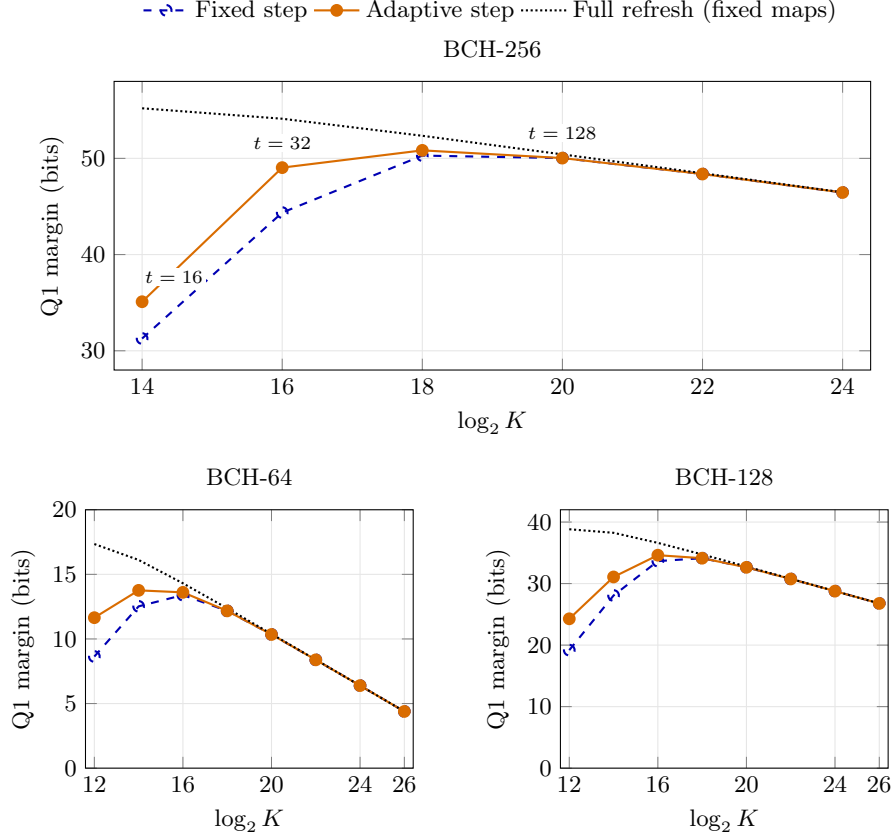


Fig. 1. Length dependence with one transvection per IMT update. The fixed curves use $(t, s) = (128, 19)$ for BCH-256 and $(64, 20)$ for the smaller outers. The adaptive curves choose the largest tested step within 0.1 bits of the best tested Q1 margin. Top labels mark step changes; the smaller outers use $t = 16, 32, 32, 64, \dots$ at successive markers. The full-refresh references retain the fixed curves' maps. All points are replayed binary64 Q1 evaluations, not full certificates; segments only connect tested lengths.

Additional mixing within each update also suppresses retention. Applying r independent transvections before adding feedback reduces the retained component's weight from $1/2$ to 2^{-r} , without changing the step or maps. We evaluated this with the BCH-256 outer, the $(t, s) = (128, 19)$ maps, feedback, routing, and cutoff all fixed, varying only K and r . Each value is an outward Q1 bound from a small bank of tilt witnesses chosen for the low BCH weights. Increasing r from one to two raises the Q1 margin from 44.26 to 51.21 bits at $K = 2^{16}$. At $K = 2^{20}$, the same change gives only 50.08 to 50.35 bits: the benefit diminishes once one transvection per update provides sufficient mixing for these Q1 bounds. The maps and replay of this study accompany the code [25]. These gains concern Q1, not full certificates. We keep one transvection per update in the plotted constructions and instead vary the step, which also changes feedback grouping and the expansion maps.

Reducing the Step at Short Lengths. The adaptive curves keep one transvection per update and select among smaller steps. At each tested K , we take the largest t within 0.1 bits of the best tested Q1 margin. This tolerance avoids selecting a smaller step for a negligible bound improvement; it is not a runtime optimum. For BCH-256, the selected pairs are $(16, 10)$ at $K = 2^{14}$, $(32, 15)$ at $K = 2^{16}, 2^{18}$, and the implemented $(128, 19)$ pair at the three larger tested lengths. At $K = 2^{16}$, the smaller step improves Q1 from 44.36 to 49.04 bits in the binary64 evaluations of Figure 1. For BCH-128 at $K = 2^{12}$, it improves Q1 from 19.18 to 24.27 bits. The short-length loss shrinks but does not disappear.

Shrinking t has a competing cost in this map family. With $t = 2^m$, the non-constant degree-at-most-two generators span only $m + \binom{m}{2}$ dimensions. This is one less than the bound (79), because the family omits the constant generator, whose image would contain the all-ones word. Our $t = 32, 16, 8$ candidates consequently use $s = 15, 10, 6$. Smaller state spaces increase fresh-state cancellation probabilities, and different expansion maps change the emitted-weight distribution. In our tests, $t = 8$ is selected at none of the tested lengths. There is no evidence here for a universal rule based only on K/t .

All maps, the derivation of p_{cancel} , and the 110-cell independent log-domain replay accompany the code [25]. Figure 1 tracks the expansion-weight shell of the first nonzero state at activation, rather than treating that state as arbitrary. At the shortest BCH-128 point this alone improves the bound by 2.90 bits; that gain is separate from changing t . The selected BCH-256 certificates do not certify the different BCH-64/128 diagnostic configurations. Nor do they certify the smaller-step BCH-256 alternatives.

State Size and Step Size. Figures 2 and 3 vary the state size for BCH-64/128. At each t , fixed ordered map chains give nested state maps; the same maps serve both outer lengths and all K . These slices treat the first nonzero state after activation as arbitrary, without the shell tracking of Figure 1. Their 130 geometries and replay accompany the code [25]. Figure 2 fixes $K = 2^{20}$. At $t = 64$, additional state first buys substantial margin and then gives diminishing returns. For BCH-128, increasing s from 10 to 16 raises the Q1 margin from

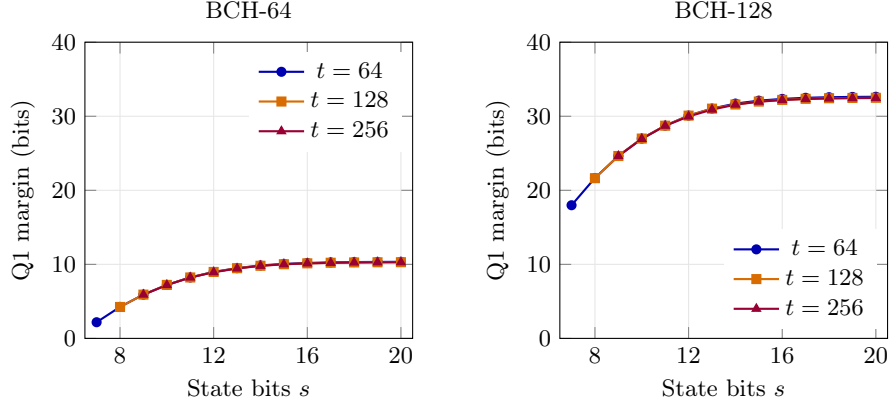


Fig. 2. IMT state and step size at $K = 2^{20}$. Each t uses its own nested map chain. More state gives diminishing Q1 gains; the step-size curves nearly coincide at shared states.

26.99 to 32.34 bits; increasing it to 20 adds only 0.29 bits. Increasing t reduces the number of inner steps, while the Q1 curves nearly coincide at shared states in this study. At $s = 20$, the BCH-128 Q1 margins for $t \in \{64, 128, 256\}$ are 32.63, 32.49, and 32.46 bits. Their closeness does not establish comparable full margins: larger steps also change the multi-row activation and cancellation bounds.

State Size at Larger Message Lengths. Figure 3 compares the state curve at $K \in \{2^{16}, 2^{20}, 2^{24}\}$, holding $t = 64$ fixed. The nearly saturated part shifts downward as K grows, but the loss need not be a uniform vertical shift. For BCH-128 at $K = 2^{24}$, raising s from 10 to 16 improves Q1 from 23.04 to 28.50 bits; raising it to 20 adds only 0.27 bits.

Together, the slices suggest calibrating t at the intended K , then checking which compatible state maps approach the available Q1 margin. The selected choice $s = 19$ is a demonstrated operating point, not a proved minimum or optimum. The final tests are the full bound over all occupations and the cost of the resulting encoder; neither follows from Q1 alone.

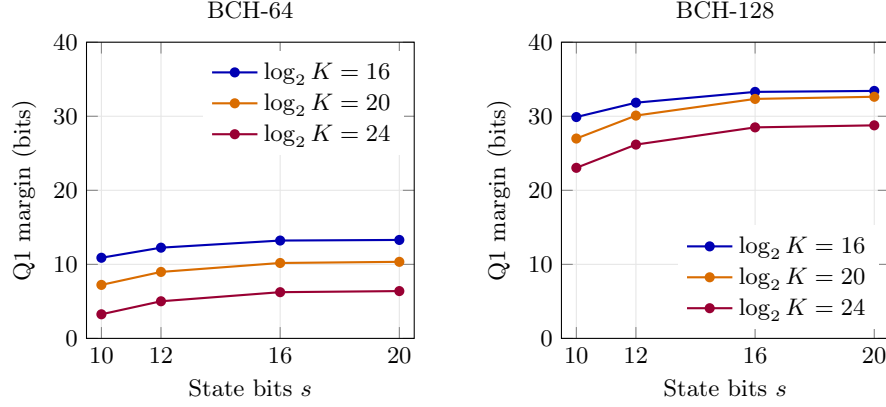


Fig. 3. IMT state slices at three message lengths, with $t = 64$ fixed. Only the four marked states are compared at every length. Increasing message length lowers the Q1 plateau; increasing state gives diminishing gains.

D.2 The Certified Engineering Curve

Figure 4 holds the outer and selected inner fixed and compares the Q1 component with the complete certificate at each length. Between 2^{20} and 2^{24} , the full bound falls from 50.062 to 46.457 bits, roughly 0.90 bits per doubling. This is a local description of the certified points, not a bound at an untested length. At $K = 2^{16}$, the Q1 margin is already lower, at 43.593 bits. This is the Q1 component of the complete outward certificate. The 44.26 bits of the r -comparison above and the 44.36 bits of Figure 1 bound the same one-row contribution with different tilt witnesses and activation envelopes; at $K = 2^{20}$ the corresponding values are 50.08, 50.03, and the certified 50.062 bits. Adding the higher-occupation bounds costs another 1.775 bits, giving the full margin of 41.818 bits; the dense contribution dominates this additional loss. The short-length dip therefore occurs in Q1 as well as in the full bound. At each of the four larger tested lengths, the Q1-to-full loss is below 0.0002 bits. Thus Q1 closely describes the selected certified curve beyond its short-length transient. This comparison concerns upper bounds, not the tightness of either bound against the true failure probability.

Appendix D.1 investigates smaller steps as an alternative at short lengths. All certificates here and the measured implementations use $(t, s) = (128, 19)$ and one transvection per update.

These certified margins are conservative inputs to the parameter selector of Section D.3.

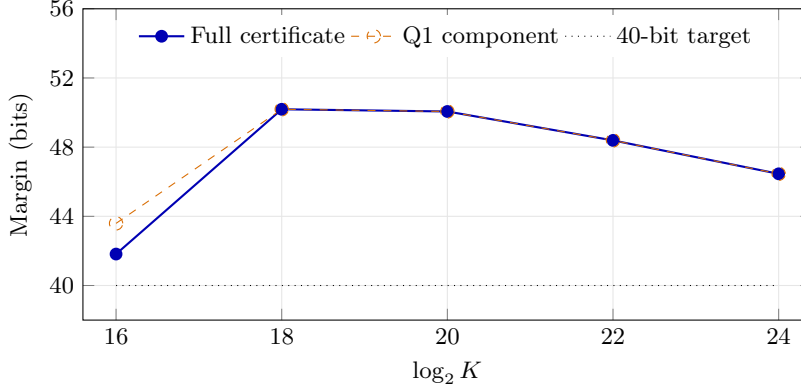


Fig. 4. Matched Q1 components and complete outward bounds for the selected BCH-256, IMT (128, 19) construction. Both curves use exactly the same maps and setup distribution. The four larger-length pairs nearly coincide. Connecting segments do not certify intermediate lengths.

D.3 Length, Work, and Parameter Selection

We state finite validity, asymptotic distance, and encoding complexity separately for each SPIN family.

Native and Requested Lengths The direct constructions apply at native lengths for which their block dimensions are integral. Their logarithmic schedules permit admissible lengths with relative gaps $o(1)$. For Structured SPIN, the zero extension in (25) maps the largest native member below any sufficiently large requested length into that output length. This wrapper preserves injectivity and absolute minimum distance. It changes rate and relative distance by only $o(1)$ and preserves linear work.

Asymptotic Statements Theorems 5 and 6 give positive relative distance with high probability for the independent-block ensembles. They specify the block schedule, setup failure probability, and rate. At rate one half, the stated examples use block constants 3 and 17, respectively; Random SPIN also uses memory constant 51/50.

Theorem 2 uses direct lengths $N_m = L_m b_m$, where $L_m = 128m$ and b_m is the least multiple of 24 satisfying $b_m \geq \frac{39}{4} \log_2 N_m$. These lengths have relative gaps $o(1)$. The requested-length wrapper gives rate $1/2 - o(1)$, relative distance $0.11 - o(1)$, and linear work.

Linear Work and Working State

Why the Structured Inner Takes Linear Time. In the scalable family, the inner step width $t = 128$ and state dimension $s = 19$ are fixed independently of N . Each of the N/t steps evaluates the fixed linear maps A, C , adds fixed-length vectors, and applies a sampled transvection as specified in (13). Even a direct bit implementation takes a constant number of operations per step, so the total inner work is $O(N)$. The sampled transvection changes with the step, but neither its width nor the arithmetic cost grows. The recursive state therefore has constant size. If the encoder also tracks its position to select the next transvection, its counter takes $O(\log N)$ bits, giving $O(\log N)$ total working state. If the position is supplied externally, the working state remains constant. This distinction does not change the linear encoding work. The stored transvection schedule and interleaver description are setup data, separate from this working state. The full encoder also needs a linear-time outer, as accounted for in (80) below.

Finite Parameter Selection The finite theorem and Figure 4 provide certified choices, rather than an extrapolated parameter schedule. The selector’s objective is measured encoding cost subject to a complete first-moment bound at the requested length and margin. A diagnostic calculation can rank candidates, but acceptance requires outward arithmetic and coverage of every nonzero occupation.

The inner parameters control different costs. Increasing t reduces the number of recursive steps; increasing s can suppress cancellations but increases the local arithmetic. The selected map also matters, not merely its dimensions. For $t = 2^m$, an s -dimensional subcode of $\text{RM}(2, m)$ requires

$$s \leq 1 + m + \binom{m}{2}. \quad (79)$$

This dimension constraint is an admissibility test, not a distance guarantee. Changing the outer or selected map requires a matching spectrum calculation and inner analysis. The fixed (128, 19) inner, certified at all five tested lengths, shows that useful finite coverage need not require retuning at every message length.

Cost Model Let $C_{\text{out}}(b)$ be the work to encode one outer block, and let $C_{\text{in}}(t, s)$ be the work for one structured inner step. A Structured SPIN encoder has the schematic cost

$$T(N) = \frac{N}{b}C_{\text{out}}(b) + \frac{N}{t}C_{\text{in}}(t, s) + O(N), \quad (80)$$

where the last term covers routing and permutation work.

For any fixed base constituent with a constant-size encoding circuit, tiling, two permutations, and two accumulators give $C_{\text{out}}(b) = O(b)$. The fixed (128, 19) IMT step has constant cost. Every term in (80) is therefore $O(N)$. The asymptotic BAA constituent uses extended Golay base blocks, but the cost argument does not depend on their 24-bit length.

E Encoder Implementation and Benchmark Details

This appendix supports Section 8. It gives the full implementation description, memory accounting, measurement protocols, and ordinary-encoding comparison. We evaluate the transposed BCH-IMT encoders of Section 8 on batches of N input blocks, with $N = 2K$ at rate 1/2 and $N = 4K$ at rate 1/4.

Shared Implementation and Optimized Kernels. The configurations share the routing machinery and an IMT template. The transpose uses

$$V_i = U_i + C^\top r_{i+1}, \quad r_i = A^\top U_i + M_i^\top r_{i+1}.$$

In particular, $A^\top$ consumes the raw input U_i . The balanced expansion is evaluated by pruned SIMD zeta stages and fixed nibble tables. Sparse feedback compiles to unrolled XORs; each output block combines five state blocks for the (128, 19) rate-1/2 inner, or three at rate 1/4. The small-length variant uses its own fixed maps. The transvection update uses XORs on selected state blocks. The selected BCH-256 circuit evaluates four rows together with AVX-512, using a generated 3758-XOR circuit; the AVX2 fallback uses a 3029-XOR circuit on two rows. The rate-1/4 constituent has its own generated transpose circuit. The online call has no allocation or type-erased callback dispatch; configuration and layout dispatch occur once per encoding.

Packed 24-bit routing indices and workspace-local page routing reduce data-movement costs. At $K = 2^{20}$ the rate-1/2 implementation uses 2048-row tiles and the rate-1/4 implementation uses 4096-row tiles. The smaller rate-1/2 measurements use direct routing and four-row SIMD packing, avoiding the intermediate tile scatter. These choices were tuned on the evaluation host.

Correctness and Certificate Binding. Symbolic checks verify the generated outer matrices and transpose circuits, the pruned zeta schedule, and both independent inner maps. Runtime tests compare the complete optimized output with dense outer multiplication and a separately implemented scalar inner recurrence. The checks cover routing layouts, tile sizes, setup compaction, boundary inputs, buffer validation, and agreement between SIMD backends. Source and binary identities, exact map columns, and the outer construction bind each reported timing to its corresponding finite certificate. A timing alone does not establish distance.

Measurement Method. Measurements use one Ryzen 9 7950X thread on Linux, pinned to CPU 15, with GCC 15.2.0 and boost disabled. The rate-1/2 SPIN build uses `-O3 -DNDEBUG -march=x86-64-v3 -mtune=znver4`, carry-less multiplication instructions, and AVX-512 for the BCH circuit. The rate-1/4 and external comparison builds use `-march=znver4`. Each reported time is the median of three independent serial process medians. Each process performs three warmups and 101 timed calls. Buffers are initialized once per process, with no copy or reset between calls. Setup, allocation, correctness checks, and output hashing are outside the online interval. These are steady-state encoder latencies, not cold-cache or full-application timings. All setup is precomputed; no heuristic refresh is included.

Table 9. Online time in milliseconds for the batched transposed encoder. The rate-1/2 row uses the small-length variant at $K = 2^{16}$; the other cells use IMT $(t, s) = (128, 19)$ with their specified maps. A dash means that no certificate-matched timing is reported here.

Outer and rate	$K = 2^{16}$	$K = 2^{18}$	$K = 2^{20}$
BCH-256, 1/2	0.340	1.460	9.289
BCH-128, 1/4	–	–	15.254

A Small-Length Variant. At $K = 2^{16}$, we keep the BCH-256 outer and routing but use $(t, s) = (64, 12)$ with two independent transvections per inner step. The smaller maps reduce arithmetic cost; the extra transvection improves state mixing. A separate full first-moment certificate covers all 512 possible active-row counts and gives a 49.328-bit margin at relative distance 0.10. This is an additional finite operating point outside the one-transvection family of Theorem 3. The repository documentation describes its fixed maps and certificate [25].

At rate 1/2 and $K = 2^{20}$, the three process medians range from 9.158 to 9.321 ms. Precomputed setup occupies 12.13 MiB, workspace 40 MiB, and input/output buffers a separate 48 MiB. The rate-1/4 setup occupies 24.25 MiB and workspace 72 MiB; its process medians range from 15.204 to 15.313 ms. Setup latency is not reported. We report no encoding timings at $K = 2^{22}$ or 2^{24} . The repository provides the encoder implementation, its documentation, and the measurement and certificate data for both rates [25].

E.1 Ordinary Encoding

Polynomial commitments require the ordinary encoder: K input blocks expand to $2K$ output blocks. We implement this direction for the same selected SPIN map and measure it separately. Each block again contains 128 parallel binary instances. These measurements include the outer encoder, routing, and inner recurrence, but no commitment or opening work.

The ordinary encoder reuses the same local circuits and direct or tiled routing strategy. It gathers the inner input and emits the output in one pass, retaining the raw input for the independent feedback map. A generated XOR circuit evaluates that map. Both directions use the same transvection updates and workspace-local page routing. Dense-oracle checks and the identity $\langle E(x), q \rangle = \langle x, E^\top(q) \rangle$ check that the implementation preserves the linear map.

The experiment uses the same Ryzen host, CPU affinity, and GCC version as above, with boost disabled. Each direction has independent input and workspace buffers. Separate processes alternate direction order, each performing three warmups and 101 timed calls. Table 10 reports medians of three process medians. Setup, allocation, and output checks are outside the interval. All runs execute serially.

Table 10 shows similar cost for both directions (9.031 ms ordinary at $K = 2^{20}$). The same experiment measures in-place transposed encoding at $K = 2^{18}$

Table 10. Ordinary and transposed SPIN encoding, in milliseconds, for 128 parallel binary instances. Setup is fully precomputed, input is preserved, and output is separate; neither direction includes PCS work.

Direction	$K = 2^{16}$	$K = 2^{18}$	$K = 2^{20}$
Ordinary	0.348	1.601	9.031
Transposed	0.340	1.460	9.289

in 1.419 ms, versus 1.460 ms with separate output. Dividing 2^{18} output blocks by this in-place latency gives approximately 185 million 128-bit correlation blocks per second. The ordinary encoder uses 256-row tiles at $K = 2^{20}$ and direct routing at the smaller lengths. Its workspace occupies 33 MiB at $K = 2^{20}$. The OT, PCS, and Flock measurements use their own integrated encoder builds; we do not rescale them using these encoder timings.

E.2 Comparison with Other Transposed Encoders

We compare SPIN with BAA, introduced by Kolesnikov et al. at CRYPTO 2026 [20], and the newer chosen-block instantiations [22]. Chosen-block BAA revises BAA, including by reusing a spectrum-controlled outer constituent. Table 1 includes both BAA variants. The other baselines are Expand–Convolute [23] and binary repeat–accumulate–accumulate (RAA) [10].

Every call evaluates a binary transposed encoder on 128 parallel instances. No weighted extension-field RAA or ordinary-encoder timing is included.

The chosen-block BAA rows use the optimized Golay [24, 12, 8] and RM(2, 5) kernels from the chosen-block BAA implementation. Expand–Convolute uses the nonsystematic implementation with the conservative profile, expander weight $w = 33$ and memory $m = 25$. The implementation takes 24 random taps plus one fixed tap and applies one convolution pass. Its native interface modifies the input buffer and writes the result to a preallocated output buffer. Our binary RAA adapter uses two accumulator/permutation layers and a repetition transpose. Chosen-block BAA and RAA use four-round Feistel permutations, with cycle walking for non-power-of-two domains. SPIN uses its packed routing tables. The measured times include each implementation’s routing costs.

Common Timing Protocol. Except for the original-BAA reference, all rows use GCC 15.2.0, with the per-implementation flags above, on the same Ryzen host and CPU. Each cell uses three serial processes, each with three warmups and 101 timed calls. Each process allocates and initializes its buffers once, then repeatedly invokes the native encoder interface. There is no input copy or reset between calls. Expand–Convolute operates on the buffer left by the previous call; the other interfaces preserve their input. Code setup, caller buffer allocation, and output hashing are outside the interval; all work inside each encoder call is timed. The chosen-block BAA message lengths are $2^m - 64$ for $m \in \{16, 18, 20\}$,

allowing both constituents to tile the same input. All other rows use $K = 2^m$. RAA processes $3K$ or $4K$ input blocks, versus $2K$ for the rate-1/2 rows.

The combined results appear in the main comparison, Table 1. Its RAA footnote states the different reference length and the optional setup-testing guarantee.

At the largest measured length, rate-1/2 SPIN with IMT takes 9.289 ms. The original rate-1/2 BAA implementation takes approximately 32 ms, giving the approximately $3.4\times$ speedup stated in the abstract. The newer chosen-block implementations improve on that baseline: chosen Golay and RM BAA take 2.58 and 2.98 times as long as SPIN, respectively. The conservative EC profile takes 181.825 ms. The repetition-3 permutation domain requires cycle walking; the repetition-4 domain does not. The faster repetition-4 timings therefore do not show that repetition-4 RAA is intrinsically faster.

Distance Context. At the largest length, rate-1/2 SPIN’s certified margin is 50.062 bits at relative distance 0.10. The chosen-block BAA evaluation reports full first-moment margins of 46.386 and 47.093 bits for Golay and RM, respectively, at the same distance target and $K = 2^{20} - 64$. Original BAA’s conservative $\sigma = 32$ profile has relative distance 0.10 with a 20-bit margin at $K = 2^{20}$ [20, Fig. 10]. For EC, we use the conservative profile reported by Kolesnikov et al. [20]: relative distance 0.05 with a 20-bit margin at $K = 2^{20}$. These bounds concern each construction’s stated random ensemble. For repetition-4 RAA, we report Blaze’s published no-test reference bound at $K = 2^{22}$ separately from our smaller timing sizes. Its relative distance is 0.19, with an approximately 13-bit margin; setup testing can improve this margin as noted in the table. We do not transfer this bound to $K = 2^{20}$ or to repetition-3 RAA, nor extrapolate the BAA or EC margins to smaller lengths.

The repository documents the comparison methodology and provides the dependency revisions, correctness checks, actual lengths, and all 5454 timing samples for the six measured rows of Table 1 [25]. The SPIN row is the certificate-matched IMT series from Table 9, measured under the same host, compiler version, and timing policy.

F Application Details and Additional Comparisons

This appendix supports Section 9. It preserves the full PCS and Flock methodology, additional comparisons, and the distinction between measurements and projected costs.

F.1 Silent OT Integration

Regular-Noise Batches. The regular-noise integration reuses the full precomputed rate-1/2 BCH-IMT code, with $(t, s) = (128, 19)$ and weight-five feedback. There is no heuristic permutation bank or per-batch code refresh. Each batch expands fresh trees, compresses their outputs, and hashes both sender messages.

Table 11 reports sender computation at two batch sizes. All entries use VAES and streaming output stores.

Table 11. Regular-noise sender computation. Initial setup, base-correlation generation, and network transport are excluded.

Timed work	$K = 2^{18}$	$K = 2^{20}$
Tree expansion (ms)	1.033	4.093
SPIN compression (ms)	1.482	11.351
Hash both sender messages (ms)	0.428	1.728
Total (ms)	2.944	17.186
Million hashed OTs/s	89.0	61.0

Both sizes use 128 noise partitions, with 4096 and 16384 leaves per tree, respectively, and no padding. Each batch consumes 1536 or 1792 fresh base-OT correlations, supplied outside timing. The noise configurator uses an effective-distance tuning value of 0.25; this is not a certified relative distance or a new hardness claim. The runs use the hardware, compilers, and three-process measurement protocol described below for stationary mode. Setup, workspace and output allocation, base-correlation generation, transport, and output consumption are untimed. The measured calls include routine allocation and local message enqueueing. Separate paired runs check OT correctness.

With ordinary cached stores, totals are 3.066 and 17.495 ms. Including an immediate sequential scan of the streaming outputs raises the totals to 3.171 and 17.971 ms. A stationary run matched to these measurements takes 3.103 ms at $K = 2^{18}$, or 84.5 million hashed OTs per second; Table 12, measured separately, reports 3.101 ms. The modes consume different base correlations, whose generation is excluded; these timings do not establish which complete protocol has lower cost.

Stationary Batches. The stationary integration uses the rate-1/2 BCH-IMT encoder at $K = 2^{18}$, with $(t, s) = (128, 19)$ and weight-five feedback. The parties retain a precomputed permutation bank and tree seeds between batches. Each batch refreshes the routing parameters and IMT masks, generates fresh leaves, and applies the transposed encoder. The refreshed permutations form a heuristic family; the uniform-setup distance certificates do not apply.

The stationary configuration uses 312 trees of 1680 leaves and pads the remaining 128 coordinates to length 2^{19} with zeros. Cached leaf seeds occupy about 8 MiB per party. Each batch consumes 312 fresh base-VOLE correlations, supplied outside timing. Initial tree expansion, permutation-bank construction, and workspace and output allocation are also untimed. The timer covers fresh-leaf generation and sums, padding, code refresh, compression, and AES hashing of both sender messages. Routine allocation and local message enqueueing within

Table 12. Stationary sender computation for 2^{18} hashed OTs per batch. Both modes use VAES; only the output-store policy differs. Initial setup, base-correlation generation, and network transport are excluded.

Timed work	Cached stores (ms)	Streaming stores (ms)
Fresh leaves and padding	0.941	0.917
Code refresh and compression	2.213	1.752
Hash both sender messages	0.423	0.429
Total	3.577	3.101

these calls remain timed; transport, transcript draining, and output consumption do not.

For both modes in Table 12, we report the median of three process medians. Each process measures 101 batches after five warmups. Stage medians need not sum to the median total. The processes run serially on one Ryzen 9 7950X core at a requested 4.5 GHz with boost disabled. libOTe and cryptoTools use GCC 13.4 with VAES enabled; the SPIN library uses GCC 15.2.0 with Zen 4 tuning and runtime-dispatched AVX-512 BCH kernels. Before timing, the harness initializes both endpoints and checks their OT outputs. It then releases the receiver’s buffers to measure the sender alone. Separate paired runs check agreement across successive code refreshes.

Streaming stores avoid filling the cache with the 8 MiB sender-output buffer. Their main benefit appears in the next batch’s compression, not in hashing itself. Including an immediate sequential scan of all output bytes raises the streaming total to 3.425 ms, or 76.5 million OTs per second; cached stores take 3.884 ms with the same scan. This scan is a memory-access control, not a downstream protocol benchmark. Streaming stores remain opt-in because the benefit depends on batch size and the consumer’s access pattern.

Regular-Noise Integration at $K = 2^{20}$. We also report an integration measurement with a different length, noise configuration, and encoder implementation. It is not a matched baseline for Table 12, and its sender totals are not comparable with Table 11. We replace the linear map in libOTe’s semi-honest Silent OT path with the rate-1/2 BCH–IMT encoder at $K = 2^{20}$, $(t, s) = (128, 19)$ and weight-five feedback. Table 13 reports protocol computation, not an encoder-based projection.

Table 13. Measured time per party for 2^{20} outputs in the Silent OT integration. Base OT, setup, buffer allocation and network latency are excluded.

Timed work	Sender (ms)	Receiver (ms)
Expansion + SPIN	19.531	20.625
Expansion + SPIN + hashing	22.099	21.841

The regular binary noise has 400 partitions of 5242 coordinates each; the remaining 352 coordinates of the 2^{21} -coordinate input are zero. The configuration subtracts those unused coordinates from the distance input to libOTe’s noise-weight calculation. This benchmark specifies the implementation and its costs; it does not supply a new hardness claim for the resulting structured LPN instance.

The harness installs synthetic matching base correlations, constructs the public SPIN setup, and allocates and touches both endpoints’ buffers before timing. It runs the sender and receiver serially on the same core, using the sender’s actual buffered transcript for the receiver. Each endpoint has its own timer and workspace. The measured calls perform expansion and the in-place linear map; the receiver also packs its choice bits. Transcript handoff, network latency, output checks and object destruction are excluded. Online communication is 179,624 bytes, excluding base correlations.

Without hashing, the receiver stores its choices in the low bits of its blocks; the correlated-OT relation holds in the remaining 127 bits. With hashing, the harness checks each selected OT message against the corresponding sender output. All 2^{20} outputs pass in every trial, including warmups, in both modes. Independent dense-oracle encoder tests also pass at $K = 2^{16}, 2^{18}, 2^{20}$.

The runs use one Ryzen 9 7950X core, pinned to CPU 15 at a requested 4.5 GHz with boost disabled, and GCC 15.2.0 with `-O3 -march=znver4`. Each entry is the median of nine calls after one warmup. Hash-enabled totals are medians of per-call sums of protocol and hashing time, not sums of independently computed medians. The no-hash sender times range from 19.474 to 19.911 ms and receiver times from 20.532 to 20.754 ms.

For context, the chosen-block BAA evaluation reports 22.810 ms for Golay and 24.278 ms for RM for the same K and rate, using an Intel i9-14900HX and MSVC. Table 1 instead uses our same-host measurements and does not combine cross-machine timings.

F.2 Polynomial Commitment Details

We use SPIN as the row encoder in a Brakedown-style polynomial commitment scheme [16]. This application tests whether fast ordinary encoding translates into fast commitment and opening. The implementation has two principal changes: it instantiates the code with SPIN, and it organizes the computation around packed binary data. At 512 MiB of input, the resulting prover commits and opens in 509 ms on one core.

Construction. Let a Boolean evaluation table define a multilinear polynomial f over $\mathbb{F} := \mathbb{F}_{2^{128}}$. A public index mapping arranges its RK bits into a matrix $M \in \mathbb{F}_2^{R \times K}$. The prover applies the same SPIN encoder Enc to each row M_i , obtaining $U \in \mathbb{F}_2^{R \times N}$ with $N = 2K$, and Merkle-commits to its columns. We also write Enc for the extension of its binary generator to $\mathbb{F}$. For a requested evaluation point z , let $\beta \in \mathbb{F}^R$ and $\gamma \in \mathbb{F}^K$ be the Boolean Lagrange weights for the row and column variables under this mapping. Thus $f(z) = \sum_{i,j} \beta_i \gamma_j M_{ij}$.

In the interactive protocol, the commitment, z , and claimed value v are fixed before the verifier samples $\alpha \leftarrow \mathbb{F}^R$ and sends it to the prover. The honest prover returns the explicit folded messages

$$m_T := \sum_{i=1}^R \alpha_i M_i, \quad m_E := \sum_{i=1}^R \beta_i M_i \quad \text{in } \mathbb{F}^K.$$

After receiving both messages, the verifier draws q independent uniform column indices from $[N]$ with replacement and requests openings at the distinct sampled indices. The prover returns those columns and their Merkle authentication. The verifier rejects malformed messages or invalid authentication. It also checks $\sum_j \gamma_j (m_E)_j = v$ and, for every sampled column j ,

$$\text{Enc}(m_T)_j = \sum_i \alpha_i U_{ij}, \quad \text{Enc}(m_E)_j = \sum_i \beta_i U_{ij}.$$

It accepts only if all checks pass. Both folds share the query set; deduplication does not change the event of missing all inconsistent columns.

The measured implementation uses domain-separated BLAKE3 transcript challenges in place of verifier randomness. It absorbs the parameter identifier, commitment root, point, and claimed value before deriving α . It then absorbs m_T followed by m_E before deriving the column indices. The verifier reproduces this order.

Making the Binary Code Useful. Expanding every bit to an extension-field element would erase much of SPIN’s encoding advantage. We therefore keep the matrix packed and encode bounded groups of rows. The internal layout follows the input index mapping, avoiding a full-matrix bit transpose. Batched column processing and table-based row combinations reduce the data movement and arithmetic around the encoder. Workspaces are allocated before repeated proving. These changes make encoding and column hashing account for approximately 79% of the fastest standalone prover time.

Parameters. We condition once on the selected code satisfying the distance bound of Theorem 3. Extending its binary generator to $\mathbb{F}$ preserves minimum distance. At $K = 2^{16}$ and 2^{18} , we use 2045 column samples and one random testing fold. For a complete matrix fixed before the testing challenge, the generic-code proximity calculation [13] gives an error below 2^{-100} for the interactive matrix test and evaluation check. We call this the conditional 100-bit fixed-matrix testing target. It excludes the one-time setup event and the separate commitment-extraction and Fiat-Shamir reductions; the calculation does not establish a complete composed security theorem. Our verification code checks this bound and its parameters with exact arithmetic.

Standalone Performance. Table 2 compares SPIN–Brakedown with Ligerito [21] on the same 2^{32} -bit input. SPIN uses $R = K = 2^{16}$ for the square shape. The

longer-row shape has $R = 2^{14}$ and $K = 2^{18}$. Both encode to 1 GiB. Longer rows reduce the SPIN proof size by 31.6% for a 3.1% increase in prover time. The square configuration spends 221 ms encoding, 180 ms hashing columns, and approximately 96 ms computing the two row combinations.

All measurements use one worker on the Ryzen 9 7950X, pinned to CPU 15, with a requested 4.5 GHz frequency and boost disabled. Each SPIN entry is the pooled median of ten trials from two processes, each excluding one warmup. For SPIN, setup, workspace allocation, outer serialization, and independent evaluation for validation are outside the prover timer. Verification excludes decoding and workspace allocation. Every timed proof verifies.

Ligerito’s rows use the optimized implementation that also supplies the Flock baseline in Section 9.3. Its ring-switching layer opens the binary multilinear polynomial at an ordinary evaluation point; no Flock computation or cached partial evaluation is supplied. Fast100 and Slim100 use initial code rates $1/2$ and $1/4$, respectively, and their shipped query and grinding parameters. Their timings pool ten trials from two processes per profile, each with one excluded warmup. Input preparation, statement serialization, independent evaluation, outer proof serialization, and decoding are excluded; internal allocations remain timed. These profiles give much smaller proofs and faster verification, at higher prover cost.

Comparison with Bolt. At the same 512 MiB input volume and on one core, the fastest measured Bolt-max commitment takes 1342 ms, compared with 401 ms for SPIN–Brakedown: a $3.34\times$ difference. Both are measured on the same host. Bolt [17] encodes 2^{27} elements of $\mathbb{F}_{2^{32}}$; our input has 2^{32} Boolean entries. The comparison therefore matches input bytes, rather than polynomial dimension or coefficient field. It includes each implementation’s code, layout, and commitment processing. The authors’ public Bolt implementation provides commitment but does not implement the complete opening prover. We optimized Bolt’s x86 expander by specializing its 128-symbol columns and replacing each heap accumulator with a fixed-size array. The compiler keeps the accumulator in SIMD registers; no handwritten assembly is used. Against the unmodified baseline, expander time falls from 733 to 331 ms and commitment time from 1757 to 1342 ms. The graph, parameters, Reed–Solomon kernel, and hashing are unchanged; baseline and optimized commitment roots match. Timings are medians of six measured trials per binary, across two processes with one excluded warmup each. The repository records the source hashes and timing boundaries [25].

Opening Cost in Isolation. Table 14 separates opening from commitment. The two SPIN configurations both open in approximately 108 ms, including the row combinations and authenticated column queries. These are measured opening times from the same trials as Table 2; they exclude outer serialization. Sending the folded messages explicitly keeps the prover’s opening work small, at the proof sizes reported above.

For Bolt, we measured the dominant opening computations. We did not implement the full opening; Table 14 reports a calibrated projection.

Table 14. One opening after commitment, at 512 MiB input volume on one core. Bolt entries are calibrated projections, not complete prover measurements or runtime bounds.

Configuration	Evidence	Opening (ms)
SPIN–Brakedown (square)	Measured	108
SPIN–Brakedown (longer rows)	Measured	108
Ligerito (Fast100)	Measured	3389
Ligerito (Slim100)	Measured	3668
Bolt-max (amortized limit)	Calibrated projection	1273
Bolt-max (non-amortized)	Calibrated projection	1729

Bolt’s opening additionally uses inner Reed–Solomon proofs and a proof of the sparse-matrix relation [17]. To estimate its cost, we measure row evaluation, a random column fold, and sumchecks, and use complete Ligerito proximity runs as proxies for the inner proofs. The calibrated projection is 1273 ms in the amortized limit, which assigns negligible cost per opening to the matrix-relation proof. Adding a calibration of that proof’s leading multiplication work gives 1729 ms without amortization. The inner proxies differ from Bolt’s constrained and batched proofs; basis preparation, some matrix-proof work, outer authentication, and transcript integration remain unmodeled. The repository records these choices and the measured components [25]. These figures are the projected cost of the selected implementation approach, not a lower bound on Bolt’s opening time.

Estimated Bolt Communication. The Bolt-max row in Table 2 uses the same amortized-limit model for time and communication. Adding the measured commitment to the projected opening gives 2614 ms, using unrounded inputs. For communication, we use the paper’s nominal 100-bit query formulas, with parity-check distance $\gamma = 0.013$ and rate-1/2 Reed–Solomon encoding. They give 15,962 message queries and 241 syndrome queries [17, Section 7.3].

The measured commitment has 2^{20} message columns and 2^{18} syndrome columns, each containing 128 four-byte symbols. Uniform distinct queries within each part transmit 7.912 MiB of column data and an expected 2.600 MiB of batched Merkle authentication. The authentication calculation includes the two dummy sibling hashes in the implementation’s non-power-of-two tree. We add 451,184 and 177,200 bytes from the two verified inner-proof proxies used for timing, 2048 bytes of row evaluations, 1920 bytes for two sumcheck transcripts, and 128 bytes of scalar-message allowance. The resulting communication estimate is 11.115 MiB, rounded to 11.1.

The inner proxies are complete Ligerito proximity proofs, not Bolt’s constrained proofs. They include their own commitments and do not exploit Bolt’s virtual syndrome oracle or batching. The estimate assigns zero per-opening communication to the shared matrix proof in the amortized limit; finite batches must

pay their share. It excludes container framing and the original commitment root. These substitutions make the total a communication model, not a generated Bolt proof. As a secondary comparison, keeping the unresolved overhead in Bolt’s published 4 GiB proof gives 10.9–11.5 MiB at 512 MiB, depending on whether Bolt’s MB denotes 10^6 or 2^{20} bytes. That agreement does not independently validate the inner-proof model. The repository includes the calculation and its tests [25].

Compatibility with Blaze. SPIN is also compatible with Blaze’s generic code-switching framework [10]. Given the stated distance bound, its linear encoder can be combined with a generic interactive oracle proof of proximity for the encoding and multilinear-evaluation relation. This gives a theoretical SPIN–Blaze composition with asymptotically smaller proofs than our Brakedown-style implementation. The inner proof replaces the explicit folded messages, allowing longer rows and shorter authenticated columns. The large proofs in our measurements therefore come from the chosen PCS construction. We give no concrete SPIN–Blaze instantiation, parameters, or performance.

The opening comparison motivates testing the complete application: commitment speed alone does not determine the cost of a PCS inside a proof system. Section 9.3 measures both commitment and opening after integration into Flock, where the claim interface and available intermediate computations also affect their cost.

F.3 Flock Integration and Full Results

We integrate SPIN–Brakedown into Flock [11] to measure its effect inside a larger prover. Flock checks batches of Boolean constraints and reduces them to claims about a committed witness. Our integration replaces its Ligerito PCS [21] with the construction in Section 9.2. SPIN improves prover throughput by $2.29\times$ and $1.63\times$ at the two measured workloads.

Adapting the Opening Interface. Flock’s final claims carry interpolation weights on a small group of witness coordinates. We incorporate these weights into the PCS row combination, so the opening proves the functional that Flock actually requests. A public permutation of witness variables preserves the six-coordinate interpolation prefix and groups known padding into zero rows. The commitment omits these rows; the opening accounts for their zero contribution. Flock generates the packed witness directly in this compact layout, avoiding a separate permutation pass. The adapter uses two separate openings with 2110 column samples each and $K = 2^{16}$. Their conditional fixed-matrix error bounds sum to less than 2^{-102} . The benchmark proves the committed batch’s compression constraints; binding public hash inputs and outputs is a separate application wrapper.

Improving the Surrounding Prover. During integration, we found that field arithmetic, repeated witness passes, and temporary storage could obscure the PCS

Table 15. Flock proofs of BLAKE3-compression constraints on one Ryzen core. SPIN and Ligerito, using its Fast100 profile, are measured with the applicable shared optimizations. Bolt entries are optimistic projections assuming free claim conversion, shared opening work, and amortized matrix-proof cost; Bolt–Flock is not implemented. Dashes denote unestimated quantities.

Compressions PCS	Prover (ms)	Verify (ms)	Proof (MiB)
16,384 SPIN–Brakedown	94	37	6.2
16,384 Ligerito	215	28	0.3
16,384 Bolt (projection)	228	—	—
65,536 SPIN–Brakedown	351	42	10.5
65,536 Ligerito	571	28	0.4
65,536 Bolt (projection)	967	—	—

improvement. We optimized Flock’s x86 kernels, reused workspaces, and evaluated the fixed linear constraints directly. The Ligerito configuration receives the applicable improvements as well. In particular, its existing column buffers can supply an intermediate zerocheck value, avoiding duplicate work. These changes preserve the checked protocol messages. The comparison below uses both PCS backends within this optimized implementation, rather than attributing shared improvements to SPIN.

Measurement Method. The SPIN and Ligerito configurations run in one executable with the same deterministic input stream and compiler settings. They use the single-worker hardware environment of the standalone PCS experiment: one Ryzen 9 7950X core at 4.5 GHz, with boost disabled. The build uses Rust 1.96.0, GCC 15.2, native CPU instructions, and 64-byte function alignment. Each workload has two processes per backend in a balanced serial order. Each process excludes five warmup and settling proofs and measures four trials; each entry is the mean of two process medians. Every generated proof is verified. Prover time includes witness generation, commitment, and proving. Setup, verification, and outer proof serialization are outside the interval. Serialization inside the SPIN opening remains timed because its bytes enter the protocol transcript.

Prover Costs. At the smaller workload, commitment takes 15.7 ms with SPIN–Brakedown and 43.7 ms with Ligerito; opening takes 17.1 and 102.8 ms. At the larger workload, these costs are 62.2 versus 162.0 ms for commitment and 46.0 versus 141.7 ms for opening. The larger-workload PCS savings account for about 195 ms of the 221 ms total reduction. Encoding alone takes 7.9 and 28.0 ms with SPIN, versus 26.9 and 98.6 ms with Reed–Solomon. These are complete integrations with different layouts and hashing; the measurements do not isolate an encoder-only substitution.

Four times the workload raises SPIN commitment and opening from 32.7 to 108.2 ms, improving their cost per compression. The surrounding prover takes

242.7 ms at the larger size, about 69% of the total. Ligerito opening grows only from 102.8 to 141.7 ms, which helps explain the smaller relative prover speedup at that size. Within SPIN commitment, leaf hashing grows from 6.7 to 33.1 ms and is a remaining optimization target.

An Optimistic Bolt Projection. Table 15 also considers replacing the PCS with Bolt, using the calibrated opening costs from Appendix F.2. We give Bolt favorable assumptions: free conversion of Flock’s weighted claims, shared opening work across its two claims, and negligible matrix-proof cost per opening in the amortized limit. We keep the surrounding prover time measured with SPIN and substitute Bolt’s measured commitment and projected opening costs. The resulting totals are 228 and 967 ms, compared with the measured SPIN totals of 94 and 351 ms. Thus the projected Bolt configuration remains slower even with these allowances. We did not implement Bolt–Flock; the projection describes this substitution and is not a runtime bound.

The SPIN–Brakedown integration favors prover time at a substantial communication cost. At the larger workload, its proof is 10.5 MiB, compared with 0.4 MiB for Ligerito, which also verifies faster.