INFORMATION THEORY AND APPLICATIONS IN COMPUTER SCIENCE

Lecture 7 | 26 October 2026

Student notes

# 7 Hadamard codes: unique, list, and local decoding

**Overview.** We study the Hadamard code as a concrete setting for unique, list, and local decoding. We prove its distance, bound the number of nearby codewords using Parseval and the binary Johnson bound, and recover a requested coordinate using two random queries. The Goldreich–Levin construction then produces a list of local decoding procedures from agreement slightly above one-half. Optional sections develop Reed–Solomon interpolation, unique decoding, and list decoding, then use random lines to obtain Reed–Muller local decoding and local list decoding. A complete proof of the binary Johnson bound appears at the end.

**Prerequisites.** Binary inner products, finite expectations, independence, the union bound, and the second-moment inequality. No Fourier analysis is assumed: the orthogonality and Parseval calculation are developed here. The optional Reed–Solomon/Reed–Muller sections use polynomial division and Gaussian elimination over a finite field. Logarithms are base two, and binary inner products and code-bit addition are computed modulo two.

## 7.1 A truth table as an error-correcting code

A message is a vector $a \in \mathbb{F}_2^k$, with $k \geq 1$. Its Hadamard encoding is the truth table of its linear function:

$$H(a) : \mathbb{F}_2^k \longrightarrow \mathbb{F}_2, \qquad H(a)(r) = a \cdot r = \sum_{i=1}^{k} a_i r_i \pmod 2.$$

The $N = 2^k$ coordinates are indexed by masks $r \in \mathbb{F}_2^k$. The coordinate at $r = 0$ is always zero. At the unit vector $e_i$ the coordinate equals $a_i$, so the map is injective. There are $2^k = N$ codewords, each of length $N$, and the binary rate is $k/2^k$.

An exponentially long word is unattractive for ordinary data transmission, but useful for a computational reduction that can query selected coordinates. Given $a$ and $r$, one encoded bit takes $k$ bit multiplications and additions to compute.

**Example 7.1** (The complete two-bit code)**.** With coordinates ordered $00, 01, 10, 11$, the four codewords are

| $a$ | $H(a)$ |
|---|---|
| 00 | 0000 |
| 01 | 0101 |
| 10 | 0011 |
| 11 | 0110 |

Every pair differs in two coordinates. The received word 0001 is at distance one from the first three codewords and distance three from the fourth. There is genuine ambiguity after a single error.

**Proposition 7.2** (Hadamard parameters)**.** *The Hadamard code is a binary linear* $[2^k, k, 2^{k-1}]_2$ *code. Every nonzero message has exactly half of its encoded coordinates equal to one.*

*Proof.* Linearity follows from $(a+b)\cdot r = a\cdot r + b\cdot r$. To count the *Hamming weight* of $H(a)$, we count the masks $r$ for which its encoded bit $H(a)(r)$ is one. Fix one particular nonzero message $a$ and keep it fixed throughout this count; only the coordinate index $r$ varies. Choose an index $i$ with $a_i = 1$.

Fix all $k-1$ mask coordinates other than $r_i$. There are exactly two masks with these fixed coordinates: one with $r_i = 0$ and one with $r_i = 1$. They form a pair $\{r, r+e_i\}$, where $e_i$ has a one in position $i$ and zeros elsewhere. Since the coefficient $a_i$ is one, flipping $r_i$ flips the encoded bit:

$$H(a)(r+e_i) = a\cdot(r+e_i) = a\cdot r + a_i = H(a)(r) + 1 \pmod 2.$$

The two bits are therefore 0 and 1, in some order. There are $2^{k-1}$ choices of the other coordinates, giving $2^{k-1}$ disjoint pairs that cover all $2^k$ masks. Each pair contributes a single 1 to the codeword, so $\|H(a)\|_0 = 2^{k-1}$. This argument applies separately to every nonzero message. The zero message has the all-zero codeword.

For distinct messages $a, b$, a coordinate differs exactly when the corresponding bit of $H(a)+H(b) = H(a+b)$ is one. Because $a+b \neq 0$, the weight calculation gives $\operatorname{dist}(H(a), H(b)) = \|H(a+b)\|_0 = 2^{k-1}$. □

It follows that fewer than $N/4$ corrupted coordinates permit unique recovery: the integer radius is $\lfloor (N/2-1)/2 \rfloor$. At $N/4$ errors, when this is an integer, take two codewords and split their $N/2$ differing positions into equal halves to obtain a received word at that distance from both. The rate tends to zero as $k$ grows, but relative distance stays exactly $1/2$.

## 7.2 List decoding

For a specified error budget, *list decoding* asks for all messages whose codewords differ from the received word in at most that many positions. The output is a list of possible messages. The goal is to keep this list small even when there are too many errors for unique recovery. Identifying the originally sent message within the list requires additional information or a further promise.

For the Hadamard code, let the received word be $y : \mathbb{F}_2^k \to \{0,1\}$. For $0 < \varepsilon \leq 1/2$, consider the candidate list

$$\mathcal{L}_\varepsilon(y) = \left\{ a : \Pr_{r \text{ uniform}} \{ y(r) = a\cdot r \} \geq \frac{1}{2} + \varepsilon \right\}.$$

These are exactly the messages whose codewords differ from $y$ in at most $(1/2-\varepsilon)N$ positions, with integer rounding implicit. A combinatorial list bound controls $|\mathcal{L}_\varepsilon(y)|$ for every $y$. A decoding algorithm must additionally produce all its elements from its allowed access to $y$. The next theorem bounds the list size at corruption $1/2-\varepsilon$ using the orthogonality of distinct linear functions.

## 7.3 From agreement to Fourier energy

Represent the received bits as signs $g(r) = (-1)^{y(r)}$, and represent a message by the character $\chi_a(r) = (-1)^{a\cdot r}$. Define the normalized Fourier coefficient

$$\widehat{g}(a) = \mathbb{E}_r[g(r)\chi_a(r)] = 2^{-k} \sum_{r \in \mathbb{F}_2^k} g(r)\chi_a(r).$$

On an agreeing coordinate the product is $+1$; on a disagreeing coordinate it is $-1$. Therefore

$$\widehat{g}(a) = 2 \Pr_r \{ y(r) = a\cdot r \} - 1. \tag{7.1}$$

In particular the desired agreement is exactly the one-sided condition $\widehat{g}(a) \geq 2\varepsilon$. A large negative coefficient corresponds to agreement with the complement of the linear function. The complement is not one of our linear codewords, so the sign must not be forgotten.

**Lemma 7.3** (Character orthogonality)**.** *For $a, b \in \mathbb{F}_2^k$,*

$$\mathbb{E}_r[\chi_a(r)\chi_b(r)] = \mathbf{1}\{a = b\}.$$

*Also, for $r, s \in \mathbb{F}_2^k$,*

$$\sum_{a \in \mathbb{F}_2^k} \chi_a(r)\chi_a(s) = N\mathbf{1}\{r = s\}.$$

*Proof.* The product in the first expectation is $(-1)^{(a+b)\cdot r}$. If $a = b$ it is identically one. Otherwise the pairing argument used in the distance proof makes its sum zero. For the second identity interchange the roles of the mask and the message: the summand is $(-1)^{a\cdot(r+s)}$. If $r = s$ all $N$ terms equal one; otherwise choosing a coordinate where $r + s$ is nonzero pairs terms of opposite sign. □

**Lemma 7.4** (Parseval with the normalization shown)**.** *For every real function $g : \mathbb{F}_2^k \to \mathbb{R}$,*

$$\sum_{a \in \mathbb{F}_2^k} \widehat{g}(a)^2 = \mathbb{E}_r[g(r)^2].$$

*Proof.* Expand both copies of the coefficient and exchange the finite sums:

$$\begin{aligned}\sum_a \widehat{g}(a)^2 &= \frac{1}{N^2} \sum_a \sum_{r,s} g(r)g(s)\chi_a(r)\chi_a(s) \\ &= \frac{1}{N^2} \sum_{r,s} g(r)g(s) \sum_a \chi_a(r)\chi_a(s) \\ &= \frac{1}{N^2} \sum_r Ng(r)^2 = \frac{1}{N} \sum_r g(r)^2.\end{aligned}$$

The second orthogonality identity kills all terms with $r \neq s$ and supplies the factor $N$ for the diagonal terms. □

**Theorem 7.5** (Hadamard list-size bound)**.** *For every $k \geq 1$, every received word $y$, and $0 < \varepsilon \leq 1/2$,*

$$|\mathcal{L}_\varepsilon(y)| \leq \frac{1}{4\varepsilon^2}.$$

*Proof.* The sign function $g$ has $g(r)^2 = 1$, so Parseval gives $\sum_a \widehat{g}(a)^2 = 1$. Each $a \in \mathcal{L}_\varepsilon(y)$ contributes at least $(2\varepsilon)^2$ to this nonnegative sum, by (7.1). Hence $1 \geq |\mathcal{L}_\varepsilon(y)|\, 4\varepsilon^2$, proving the claim. □

**The general Johnson bound (binary form).** Let $C \subseteq \{0,1\}^n$ be any binary code whose distinct codewords have Hamming distance at least $\delta n$, where $0 < \delta \leq 1/2$. For every received word $z \in \{0,1\}^n$ and every $0 \leq \rho < 1/2$ with $2\rho(1-\rho) < \delta$, the *binary Johnson bound* states that

$$|\{c \in C : \operatorname{dist}(c, z) \leq \rho n\}| \leq \frac{\delta}{\delta - 2\rho(1-\rho)}.$$

Thus a fixed positive denominator gives a list-size bound independent of $n$. The condition on $\rho$ is equivalently

$$\rho < J_2(\delta), \qquad J_2(\delta) = \frac{1 - \sqrt{1-2\delta}}{2},$$

where $J_2(\delta)$ is the *binary Johnson radius*. Linearity is not required. This bounds the number of candidates; it does not provide an algorithm for finding them. A complete proof appears at the end of these notes.

**The Hadamard specialization.** Its relative distance is exactly $\delta = 1/2$, so $J_2(\delta) = 1/2$. Substituting $\rho = 1/2 - \varepsilon$ into the general bound gives

$$\frac{1/2}{1/2 - 2(1/2 - \varepsilon)(1/2 + \varepsilon)} = \frac{1}{4\varepsilon^2}.$$

This is exactly the preceding Hadamard bound: the numerical guarantee already follows from its minimum distance. Hadamard's Fourier structure provides the short Parseval proof above and supports the later decoding construction. For a code with $\delta < 1/2$, the Johnson radius is smaller than $1/2$; the same general theorem does not guarantee a short list arbitrarily close to one-half corruption.

Using the binary alphabet matters here. The weaker, alphabet-independent Johnson bound has radius $1 - \sqrt{1 - \delta}$, which at $\delta = 1/2$ is $1 - 1/\sqrt{2} \approx 0.293$. Applying that version to a binary code discards information about its alphabet. The binary Johnson bound gives the full Hadamard radius and list-size guarantee.

**Reference.** Venkatesan Guruswami, *Algorithmic Results in List Decoding*, §3.1, Theorem 3.1 and its proof (printed pp. 124–126), specialized to $q = 2$. Theorem 3.3 gives the alphabet-independent version.

At agreement 0.6, take $\varepsilon = 0.1$: the list contains at most twenty-five messages for every $k$. In the two-bit example, the received word 0001 has three messages agreeing on $3/4$ of coordinates. The theorem allows at most four when $\varepsilon = 1/4$, consistent with that ambiguity. The bound is independent of the blocklength; it concerns how many noticeably correlated functions one fixed word can support.

The same proof gives a useful variant for real-valued predictions. If $g(r) \in [-1, 1]$ for every $r$, then $\mathbb{E}g^2 \leq 1$, and the number of coefficients with $|\widehat{g}(a)| \geq \tau$ is at most $1/\tau^2$. The absolute-value version counts positive and negative coefficients together.

## 7.4 Local self-correction below one-quarter corruption

Suppose a fixed received word $y$ differs from a particular Hadamard word $H(a)$ on at most a $\rho$ fraction of coordinates. The adversary may choose those coordinates knowing $a$. It fixes $y$ before the decoder chooses its random masks; the oracle does not change answers during a run. The task is to recover the bit $a \cdot x$ at one requested coordinate $x$, while reading only a few locations of the corrupted word.

The algebraic identity

$$a \cdot x = a \cdot r + a \cdot (r + x)$$

holds for every $r$. It lets us express the target as the sum of two randomly located encoded bits. Define the one-trial algorithm:

1. Draw $R$ uniformly from $\mathbb{F}_2^k$ using $k$ fresh random bits.
2. Query $y(R)$ and $y(R + x)$.
3. Return $y(R) + y(R + x)$ modulo two.

**Theorem 7.6** (Two-query self-correction)**.** *If $d_H(y, H(a)) \le \rho N$, the algorithm returns $a \cdot x$ with probability at least $1 - 2\rho$ for every fixed requested coordinate $x$. In particular, if $\rho < 1/4$, its success probability exceeds $1/2$.*

*Proof.* Let $B = \{r : y(r) \neq a \cdot r\}$, so $|B|/N \le \rho$. The first query $R$ is uniform. Translation by $x$ is a bijection, so $R + x$ is uniform as well. Their locations need not be independent; for fixed $x$ one determines the other. Nevertheless the union bound gives

$$\Pr\{R \in B \text{ or } R + x \in B\} \le \Pr\{R \in B\} + \Pr\{R + x \in B\} \le 2\rho.$$

If neither query is corrupted, linearity makes the returned bit exactly $a \cdot x$. The error event is thus contained in the displayed event. □

This guarantee is pointwise in the requested coordinate and probabilistic over the decoder's masks. It is not an average guarantee over target coordinates. Even if the requested coordinate itself is corrupted, its value can be recovered from other positions.

### Amplification and recovery of a whole message

Fix $\rho < 1/4$ and set $\eta = 1/2 - 2\rho > 0$. Repeat the trial independently $s$ times and take a majority vote, using any fixed rule on a tie. Let $F_j$ indicate failure in trial $j$. For the fixed oracle and target, the $F_j$ are independent because they are functions of independent fresh masks, even though queried coordinates may accidentally coincide. Each has mean at most $1/2 - \eta$, and variance at most $1/4$. A wrong majority implies $\sum_j F_j \ge s/2$, and hence

$$\sum_j F_j - \mathbb{E}\sum_j F_j \ge s\eta.$$

The elementary second-moment inequality therefore gives

$$\Pr\{\text{wrong majority}\} \le \frac{\operatorname{Var}(\sum_j F_j)}{s^2\eta^2} \le \frac{1}{4s\eta^2}. \tag{7.2}$$

For a target $0 < \beta < 1$, $s \ge 1/(4\eta^2\beta)$ suffices for failure probability at most $\beta$, using $2s$ queries. Section 7.6 gives a sharper amplification bound with logarithmic dependence on $1/\beta$.

To recover all $k$ message bits, request $x = e_i$ for each $i$. Using $s \ge k/(4\eta^2\beta)$ trials for each coordinate makes each failure probability at most $\beta/k$. A union bound makes the probability of any wrong message bit at most $\beta$, using at most $2ks$ queries. With straightforward mask generation and dot-product indexing, the running time is polynomial in $k, 1/(1-4\rho), 1/\beta$ under random access to the oracle.

For example, at $\rho = 1/8$ we have $\eta = 1/4$. The one-trial success is at least $3/4$, and $s = 40$ gives the elementary amplified bound $1/10$ for one requested coordinate. The query count is eighty regardless of the word length $2^k$.

## 7.5 From a list-size bound to efficient decoding

There are three different conclusions on the table:

1. Distance gives uniqueness below a corruption fraction of $1/4$.
2. Parseval bounds the number of candidate messages at any corruption fraction $1/2 - \varepsilon$, for $\varepsilon > 0$.
3. Self-correction recovers a coordinate efficiently when the received word is promised to be closer than $1/4$ to one codeword.

The third conclusion does not extend automatically to the second regime. If $\rho = 1/2 - \varepsilon$, its one-trial lower bound becomes $2\varepsilon$, which can be below $1/2$; majority amplification then has no proved advantage to amplify. A short list is also insufficient by itself when finding the list would require searching all $2^k$ messages.

**What exhaustive list finding does establish.** Read all $N$ bits of $y$. For each of the $N$ messages $a$, evaluate its agreement with $y$ on all $N$ coordinates and retain it if the threshold holds. This returns the exact list in at most $O(kN^2)$ bit operations by straightforward dot products. The size bound limits the output, not this search cost. Even a faster transform running in $O(N \log N)$ arithmetic operations is exponential in $k$.

For completeness, that transform follows a short recurrence. Split $r = (r', b)$ and $a = (a', c)$, where $b, c \in \{0, 1\}$. If $g_0, g_1$ are the restrictions of $g$ to last coordinate zero and one, then

$$\widehat{g}(a', c) = \frac{1}{2} \left( \widehat{g_0}(a') + (-1)^c \widehat{g_1}(a') \right).$$

Compute both transforms on $N/2$ entries recursively, then perform $O(N)$ additions and subtractions. The recurrence $T(N) = 2T(N/2) + O(N)$ solves to $O(N \log N)$. This is efficient in the truth-table length $N$, but still exponential in the message length $k$.

**Efficient reconstruction.** The Goldreich–Levin construction below produces local decoding procedures in time polynomial in $k, 1/\varepsilon$. Querying them at the unit vectors recovers candidate messages. The same mechanism underlies the Goldreich–Levin hardcore-predicate reduction.

**A short list can be disambiguated if extra bits are available.** Suppose a fixed list contains the true message $a$ and at most $L - 1$ wrong messages. After the list is fixed, independently sample a uniform binary $t \times k$ matrix $A$ and send the fingerprint $Aa$. For each $b \neq a$, $A(a - b)$ is uniform on $\mathbb{F}_2^t$: in each row a nonzero vector has a fair inner product, and the rows are independent. Thus

$$\Pr_A\{Ab = Aa\} = 2^{-t}, \qquad \Pr_A\{\text{some wrong candidate survives}\} \leq (L-1)2^{-t}.$$

Choose $t = \lceil \log(L/\beta) \rceil$ for error at most $\beta$, with $t = 0$ allowed if the list is already a singleton. This counts $t$ trusted fingerprint bits, not the $tk$ seed bits of $A$. Those seed bits must be shared or communicated separately. This requires a reliable side channel and a matrix chosen independently of the list.

## 7.6 Goldreich–Levin: a list of local decoding procedures

**The task and the order of the quantifiers.** Keep the fixed oracle $y : \mathbb{F}_2^k \to \{0, 1\}$ and the agreement list $\mathcal{L}_\varepsilon(y)$ from above. The setup algorithm receives $k, \varepsilon$ and oracle access to $y$, but no target message $a$. It outputs a list of descriptions of randomized oracle procedures. A description consists of shared setup data $\sigma$ and a short advice string $b$. After setup, $D^y_{\sigma,b}(x)$ should recover the requested codeword coordinate $H(a)(x) = a \cdot x$, using fresh randomness and few queries to $y$. Taking $x = e_i$ also recovers message bit $a_i$. Thus our construction is local list *correction*, which immediately gives local list decoding of message bits for this systematic code.

**Theorem 7.7** (Hadamard local list decoding: a precise per-target guarantee)**.** *For every $k \geq 1$ and $0 < \varepsilon \leq 1/2$, there is a randomized setup algorithm that outputs fewer than $16/\varepsilon^2 + 2$ candidate procedures. For every fixed $a \in \mathcal{L}_\varepsilon(y)$, with probability at least $3/4$ over setup, one of the returned descriptions $(\sigma, b)$ satisfies*

$$\text{for every } x \in \mathbb{F}_2^k, \qquad \Pr_U\{D^y_{\sigma,b}(x) = a \cdot x\} \geq \frac{3}{4}.$$

*The same description works for all requested coordinates; $U$ is fresh local randomness for each call. Setup makes no queries to $y$. One local call makes $O(\varepsilon^{-2})$ queries to $y$, and setup and local evaluation take time polynomial in $k, 1/\varepsilon$, not in $N = 2^k$.*

The theorem is per fixed target $a$. It does not say that one setup simultaneously covers every member of $\mathcal{L}_\varepsilon(y)$; that stronger statement will require repetition below. Other returned descriptions may be junk, and the algorithm does not identify which description represents the originally transmitted message.

### Construction: guess a few parities, not the whole message

Choose the smallest positive integer $\ell$ such that

$$m = 2^\ell - 1 \geq \frac{8}{\varepsilon^2}.$$

Sample independent uniform masks $R_1, \ldots, R_\ell \in \mathbb{F}_2^k$; these form the shared setup seed $\sigma$. For each nonzero $v \in \mathbb{F}_2^\ell$, put $R_v = \sum_j v_j R_j$. For every guess $b \in \mathbb{F}_2^\ell$, define the intermediate Boolean function

$$h_{\sigma,b}(x) = \text{majority}_{v \neq 0}(y(x + R_v) + b \cdot v).$$

All additions of bits are modulo two. There are $m$ votes, so no tie is possible. This equation describes an oracle procedure; setup does not evaluate the function on all $2^k$ inputs.

The candidate local procedure is the old two-query corrector applied to this intermediate function. Given a requested coordinate $x$, sample $U$ uniformly from $\mathbb{F}_2^k$ and return

$$\boxed{D^y_{\sigma,b}(x) = h_{\sigma,b}(U) + h_{\sigma,b}(U + x).}$$

Evaluate each of the two majorities using queries to $y$. Setup simply lists all $2^\ell$ guesses $b$, with the same $\sigma$ shared by them. No individual one of the unknown parities $a \cdot R_j$ needs to be predictable: the correct joint guess is somewhere in this short list.

### Proof: a weak predictor becomes a close word, then a local decoder

Fix $a \in \mathcal{L}_\varepsilon(y)$. The advice we want is $b^*_j = a \cdot R_j$, so $b^* \cdot v = a \cdot R_v$. It is among the enumerated guesses for every setup seed. We analyze this seed-dependent guess without conditioning on its value.

*First, the shifts are pairwise independent over setup.* For distinct nonzero $v, w \in \mathbb{F}_2^\ell$, the two coefficient vectors are linearly independent over $\mathbb{F}_2$. Therefore the linear map taking $(R_1, \ldots, R_\ell)$ to $(R_v, R_w)$ is onto $\mathbb{F}_2^k \times \mathbb{F}_2^k$, with equally many preimages for every pair. Hence $R_v, R_w$ are independent uniform masks. The collection need not be jointly independent. Do not require the map $v \mapsto R_v$ to have full rank or distinct outputs: $\ell$ can exceed $k$.

*Second, bound the error at each fixed input.* For fixed $x$, the signed correctness of vote $v$ for the right advice is

$$Z_v(x) = (-1)^{y(x+R_v)+a\cdot R_v+a\cdot x} = (-1)^{y(x+R_v)+a\cdot(x+R_v)}.$$

Uniformity gives $\mathbb{E}_\sigma Z_v(x) = 2\Pr_r\{y(r) = a \cdot r\} - 1 \geq 2\varepsilon$. Pairwise independence gives

$$\mathrm{Var}_\sigma\left(\frac{1}{m}\sum_{v\neq 0} Z_v(x)\right) \leq \frac{1}{m}.$$

An incorrect majority requires this average to be at most zero. Chebyshev therefore yields

$$\Pr_\sigma\{h_{\sigma,b^*}(x) \neq a \cdot x\} \leq \frac{1}{4m\varepsilon^2}.$$

These are probabilities over the setup masks, not independent fresh votes after those masks have been fixed. Pairwise independence is enough for this variance bound; it does not justify a Chernoff bound for these votes.

*Third, turn the fixed-input bound into a close intermediate word.* Let $e(\sigma) = \Pr_x\{h_{\sigma,b^*}(x) \neq a\cdot x\}$, where $x$ is uniform. Average the last inequality over $x$, then apply Markov:

$$\mathbb{E}_\sigma e(\sigma) \leq \frac{1}{4m\varepsilon^2}, \qquad \Pr_\sigma\{e(\sigma) > 1/8\} \leq 8\mathbb{E}_\sigma e(\sigma) \leq \frac{2}{m\varepsilon^2} \leq \frac{1}{4}.$$

Thus with setup probability at least $3/4$, the *fixed function* $h_{\sigma,b^*}$ is within relative distance $1/8$ of $H(a)$. For every such setup, Theorem 7.6 applies to that function: for every requested $x$, the fresh random masks $U, U+x$ are both uniform, so their two error probabilities sum to at most $1/4$. Consequently $D^y_{\sigma,b^*}(x)$ succeeds with probability at least $3/4$ for every $x$. This proves the theorem's quantifier order.

**Resource accounting.** Minimality of $\ell$ gives $2^\ell < 16/\varepsilon^2 + 2$, bounding the number of descriptions. The shared masks use $k\ell$ random bits. Precomputing the $m$ subset sums takes $O(km)$ bit operations, with no oracle calls. Each local call uses at most $2m$ queries and $O((k+\ell)m)$ bit operations to form query addresses, compute guessed parities, and count votes.

### Amplification and the two different simultaneous guarantees

Fix a good setup and a good description for one target. Repeat a local call $s$ times with independent fresh masks and take an odd majority. Each failure indicator $F_j$ has probability at most $1/4$, and these indicators are independent conditional on the fixed setup and oracle. The same exponential-moment argument as for repetition coding gives

$$\Pr\{\text{wrong majority}\} \leq 3^{-s/2}\mathbb{E}3^{\sum_j F_j} \leq 3^{-s/2}(3/2)^s = (\sqrt{3}/2)^s \leq e^{-s/8}.$$

Thus an odd $s \geq 8\ln(1/\beta)$ gives error at most $\beta$ for any requested coordinate, using $2ms = O(\varepsilon^{-2}(1+\log(1/\beta)))$ queries. For recovering all $k$ message bits, query $x = e_i$ and use error at most $\beta/k$ per bit. A union bound gives whole-message error at most $\beta$ with $O(k\varepsilon^{-2}(1+\log(k/\beta)))$ queries to a good procedure. The procedure's setup success probability and this conditional decoding success probability are separate guarantees.

For a single setup to cover *all nearby target messages*, repeat setup independently $T$ times and unite the lists of descriptions. A designated target is missed with probability at most $4^{-T}$; Parseval bounds the number of targets by $1/(4\varepsilon^2)$. Hence

$$\Pr\{\text{some } a \in \mathcal{L}_\varepsilon(y) \text{ has no good description}\} \leq \frac{4^{-T}}{4\varepsilon^2}.$$

Choose $T \geq 1$ to make this at most $\delta$. Write $L$ for the total number of descriptions, so $L = O(\varepsilon^{-2}\log(1/(\varepsilon\delta)))$; each local call retains the same query bound. For constant $\delta$, this simultaneous-target guarantee costs an extra $O(\log(1/\varepsilon))$ setups. This is still a statement about

a list of *local procedures*, not simultaneously correct answers to all coordinates. To form explicit message candidates from all $L$ descriptions, use an odd $s \geq 8\ln(kL/\eta)$ trials for each of the $kL$ bit calls. For every good description, each bit then has error at most $\eta/(kL)$; there is no correctness promise for junk descriptions. On the good-setup event, choose one good description per target before drawing fresh local coins. A union bound over their at most $kL$ required bit calls gives an explicit message list containing every qualifying target except with additional probability $\eta$. The overall failure is at most $\delta + \eta$, and the list may contain extras. The total query count is $O(kL\varepsilon^{-2}(1 + \log(kL/\eta)))$.

**References.** Vadhan's published December 2012 monograph is available from the author's site; Definition 7.54 is on printed pp. 254–255 (PDF pp. 257–258), and Problem 7.12 is on printed p. 276 (PDF p. 279). The original Goldreich–Levin paper is available from Goldreich's site.

## 7.7 Optional: Reed–Solomon codes and unique decoding

The Hadamard arguments used orthogonality and linearity. A different code uses the fact that a nonzero polynomial has few roots. We begin with interpolation and unique decoding, then extend the method to list decoding and local decoding of multivariate polynomial codes.

Fix a finite field $\mathbb{F}_q$, distinct points $\alpha_1, \ldots, \alpha_n$ in that field, and integers $1 \leq k \leq n \leq q$. A message is a polynomial $f \in \mathbb{F}_q[X]$ of degree less than $k$, represented by its $k$ coefficients. Define

$$\mathrm{RS}(n,k) = \{(f(\alpha_1), \ldots, f(\alpha_n)) : \deg f < k\}.$$

We use the elementary root bound: a nonzero polynomial of degree $d$ over a field has at most $d$ distinct roots. To prove it, a root $\alpha$ gives a factor $X - \alpha$ by division with remainder; divide it out and induct on the degree. Distinct roots of the original polynomial other than $\alpha$ remain roots of the quotient, because their difference from $\alpha$ is nonzero and hence invertible in the field.

**Proposition 7.8** (Reed–Solomon parameters)**.** *The evaluation code is linear with parameters* $[n, k, n-k+1]_q$.

*Proof.* Evaluation is a linear map. If two degree-less-than-$k$ polynomials agree at all $n \geq k$ points, their difference has at least $k$ roots and degree at most $k-1$, so it is zero. Thus evaluation is injective and the code has size $q^k$ and dimension $k$. Distinct polynomials agree at at most $k-1$ positions, so their words have distance at least $n-k+1$. For equality, when $k \geq 2$ take the nonzero polynomial $\prod_{j=1}^{k-1}(X - \alpha_j)$ and compare it with zero. Its evaluations vanish at exactly those $k-1$ positions; at the remaining distinct points all factors are nonzero. When $k = 1$, any nonzero constant differs from zero at all $n$ positions. Thus the distance is exactly $n-k+1$. □

For erasures, any $k$ surviving values suffice. If their points are $\beta_1, \ldots, \beta_k$ and values $v_1, \ldots, v_k$, the polynomial is

$$f(X) = \sum_{i=1}^{k} v_i \prod_{j \neq i} \frac{X - \beta_j}{\beta_i - \beta_j}. \tag{7.3}$$

Each denominator is nonzero, so field division is valid. Each product has degree $k-1$ and evaluates to one at its own $\beta_i$ and zero at the other points. Hence the displayed polynomial has the prescribed values; uniqueness follows from the root bound.

**Theorem 7.9** (A complete unique decoder)**.** *Fix an integer $t \geq 0$ with $2t \leq n - k$. Given a received word $y \in \mathbb{F}_q^n$, the following procedure recovers every transmitted Reed–Solomon word corrupted in at most $t$ positions, using a polynomial number of field operations:*

1. *Find polynomials $E, A$ with $E$ monic of degree exactly $t$ and $\deg A \leq k + t - 1$, satisfying $A(\alpha_i) = y_i E(\alpha_i)$ for every $i$.*
2. *If the linear system is inconsistent, report failure. Otherwise perform polynomial division $A/E$. If it is a polynomial $f$ of degree less than $k$ whose evaluation differs from $y$ at at most $t$ positions, return it; otherwise report failure.*

*Proof.* Suppose $y$ differs from the true evaluations of $f$ in $s \leq t$ positions, whose set is $B$. Their error-locator polynomial is $E_0(X) = \prod_{i \in B}(X - \alpha_i)$, a monic degree-$s$ polynomial. Multiply it by any monic polynomial of degree $t - s$, for example $X^{t-s}$, to obtain a monic $E$ of degree $t$. Set $A = fE$. At an uncorrupted position $A(\alpha_i) = f(\alpha_i)E(\alpha_i) = y_i E(\alpha_i)$; at a corrupted one both sides are zero. Thus a solution exists even if the actual number of errors is below $t$.

Now consider any solution returned by the linear system. The polynomial $A - fE$ has degree at most $k + t - 1$. At each of the at least $n - s$ uncorrupted positions it vanishes. But

$$n - s \geq n - t \geq k + t > k + t - 1.$$

There are more roots than its degree, so $A - fE = 0$ identically. Since $E$ is monic it is nonzero, and polynomial division returns exactly $f$. The validation therefore accepts. This proves that every solution works under the promised error budget, not only the specially constructed one.

Finally, write $E = X^t + \sum_{j=0}^{t-1} e_j X^j$ and $A = \sum_{j=0}^{k+t-1} a_j X^j$. The constraints are affine linear equations in the $t + (k + t) = k + 2t \leq n$ unknown coefficients. Gaussian elimination, polynomial division, and evaluation at $n$ points use polynomially many field operations. An explicit representation and arithmetic for $\mathbb{F}_q$ are assumed. □

**Example 7.10** (Five evaluations over $\mathbb{F}_7$)**.** Take $n = 5$, $k = 3$, points $0, 1, 2, 3, 4$, and $f(X) = 1 + 2X + X^2$. The word is $(1, 4, 2, 2, 4)$ modulo seven. Its distance is three, so two erasures or one unknown error are correctable. If the last value changes to zero, receive $y = (1, 4, 2, 2, 0)$. For $t = 1$ a solution is

$$E(X) = X - 4 = X + 3, \qquad A(X) = f(X)E(X) = X^3 + 5X^2 + 3 \quad \text{in } \mathbb{F}_7[X].$$

The last constraint holds because $E(4) = 0$; the first four hold because they are uncorrupted. Dividing $A$ by $E$ recovers $1 + 2X + X^2$, and checking its evaluations confirms one disagreement. For two erasures, any three surviving pairs instead go directly into formula (7.3). The two procedures exploit different information about the error locations.

## 7.8 Optional: list decoding Reed–Solomon codes

Write $d = k - 1$, so the message polynomial has degree at most $d$. Unique decoding identifies one polynomial from more than roughly half correct evaluations. List decoding can work with far fewer correct values. The idea is to fit one *bivariate* polynomial through all the received pairs: every candidate message then becomes a polynomial root of that same equation. We use Sudan's interpolation method without multiplicities.

**Theorem 7.11** (An elementary Reed–Solomon list decoder)**.** *Let $1 \le d < n \le q$, with distinct evaluation points $\alpha_1, \dots, \alpha_n \in \mathbb{F}_q$ and received values $y_1, \dots, y_n$. Choose integers $u \ge 0$, $\ell \ge 1$ such that $(u+1)(\ell+1) > n$. For any integer $t > u + d\ell$, one can find* all *polynomials $f$ of degree at most $d$ agreeing with the received word at at least $t$ positions. The list has at most $\ell$ elements; the algorithm uses a polynomial number of field operations in $n, q, d, u, \ell$.*

*Proof.* Find a nonzero polynomial

$$Q(X,Y) = \sum_{i=0}^{u} \sum_{j=0}^{\ell} c_{ij} X^i Y^j \quad \text{such that} \quad Q(\alpha_s, y_s) = 0 \quad (1 \le s \le n).$$

These are $n$ homogeneous linear equations in more than $n$ unknowns, so Gaussian elimination finds a nonzero solution.

Suppose $f$ is a candidate. Each agreement position is a root of the univariate polynomial $Q(X, f(X))$. But

$$\deg Q(X, f(X)) \le u + d\ell < t.$$

The root bound forces $Q(X, f(X)) = 0$ identically. Division by the monic polynomial $Y - f(X)$, treating $Y$ as the variable, therefore has zero remainder: $Y - f(X)$ divides $Q(X, Y)$.

There are at most $\ell$ such distinct polynomials $f$. Indeed, after factoring out $Y - f_1(X)$, any other root $f_2$ is still a root of the quotient: $(f_2 - f_1)Q_1(X, f_2(X)) = 0$ and $f_2 - f_1 \neq 0$. Repeating would otherwise give more linear factors than the $Y$-degree of $Q$.

The coefficient procedure below finds every degree-at-most-$d$ polynomial root of $Q$ in polynomial time. Finally check each candidate's agreement with the received word, discarding those with fewer than $t$ matches. Thus the output contains exactly the requested list. □

**Finding the polynomial roots, without a factoring theorem.** We recover the coefficients of $f(X) = a_0 + a_1 X + \cdots + a_d X^d$ in order. First divide $Q$ by the largest power of $X$ dividing every term. This does not change its polynomial roots, and now $A(Y) := Q(0, Y)$ is nonzero. If $Q(X, f(X)) = 0$, then $A(a_0) = 0$. Try each of the at most $q$ field elements to find these possible constant coefficients.

For each root $a$, substitute $Y = a + XY'$ in $Q$, divide out the largest common power of $X$ again, and repeat with $Y'$ as the new variable. A root $f$ with constant coefficient $a$ becomes the root $(f(X) - a)/X$ of this new equation. After $d + 1$ steps, test each complete coefficient sequence by substitution in the original $Q$.

The search tree stays small. If $a$ has multiplicity $s$ in $A$, write $Q$ in powers of $X$ and $Y - a$. After substitution a term $X^i(Y - a)^j$ becomes $X^{i+j}(Y')^j$. The nonzero term with $i = 0, j = s$ shows that the smallest common $X$-power is some $v \le s$. After dividing by $X^v$, specialization at $X = 0$ retains only terms with $i + j = v$, so its $Y'$-degree is at most $v \le s$. The root multiplicities of $A$ sum to at most $\deg A$. Hence the specialized degrees summed over an entire level never exceed those of the preceding level, and are at most $\ell$. Each node has at most its specialized degree many children, so each level has at most $\ell$ nodes. After $j$ substitutions, the $X$-degree is at most $u + j\ell$ and the $Y$-degree at most $\ell$. There are $d + 1$ levels; enumerating field elements at each node gives the stated polynomial bound.

**How many errors can this correct?** Take $u = \lfloor \sqrt{nd} \rfloor$ and $\ell = \lfloor \sqrt{n/d} \rfloor$. Then

$$(u+1)(\ell+1) > n, \qquad u + d\ell \le 2\sqrt{nd}.$$

Consequently agreement strictly greater than $2\sqrt{d/n}$ suffices. For example, take $n = 100, d = 1$ over $\mathbb{F}_{101}$. With $u = \ell = 10$, 121 coefficients satisfy 100 constraints. The decoder finds every affine polynomial agreeing at 21 or more positions, and outputs at most ten candidates: up to 79 errors, compared with 49 for unique decoding.

**A form used below.** Let $0 < \alpha \le 1$ and suppose $q \ge 4096(d+1)/\alpha^2$. For evaluations at every point of $\mathbb{F}_q$, put

$$\ell = \lceil 4/\alpha \rceil, \qquad u = \lceil \alpha q/4 \rceil - 1.$$

Here $(u+1)(\ell+1) > q$ and, using $\ell \le 5/\alpha$,

$$u + d\ell < \frac{\alpha q}{4} + \frac{5\alpha q}{4096} < \frac{\alpha q}{2}.$$

The decoder therefore returns exactly the degree-at-most-$d$ polynomials with agreement at least $\alpha/2$, with list size at most $\ell \le 5/\alpha$. The integer threshold is $\lceil \alpha q/2 \rceil$.

## 7.9 Optional: local decoding Reed–Muller codes

A Reed–Muller code evaluates a multivariate polynomial:

$$P : \mathbb{F}_q^m \longrightarrow \mathbb{F}_q, \qquad \deg P \le d < q,$$

where degree means *total degree*. Its blocklength is $N = q^m$. Along an affine line $L(t) = x + tv$, the restriction $P(L(t))$ is a univariate polynomial of degree at most $d$. Thus a line is a short Reed–Solomon word inside the much larger Reed–Muller word.

**What is recovered?** Given oracle access to a corrupted word $w : \mathbb{F}_q^m \to \mathbb{F}_q$, we want $P(x)$ at a requested point $x$, without reading all $q^m$ values. This is formally *local correction* of an encoded coordinate. It is also local decoding of message symbols if the message is stored as values at a fixed set of independent evaluation positions (a systematic encoding). It does not directly return an arbitrary coefficient of $P$.

The code has relative distance at least $1 - d/q$: distinct polynomials agree on at most a $d/q$ fraction of the space. To prove this, induct on $m$ for a nonzero difference polynomial. Write it as a polynomial in the last variable, of degree $s$, whose nonzero leading coefficient has total degree at most $d - s$. For a uniform choice of the other variables, the leading coefficient vanishes with probability at most $(d-s)/q$ by induction. Otherwise the univariate root bound gives probability at most $s/q$. The sum is at most $d/q$.

**Theorem 7.12** (Local correction by decoding a random line)**.** *Let $m \ge 1$ and $1 \le d \le (q-1)/3$. If $w$ differs from $P$ on at most a $1/12$ fraction of points, there is a randomized procedure that, for* every *requested $x$, returns $P(x)$ with probability at least $3/4$. It makes at most $q-1$ queries to $w$ and uses polynomially many field operations in $m, q, d$.*

*Proof.* Choose $v$ uniformly from all of $\mathbb{F}_q^m$, including 0. Query

$$w(x + tv) \qquad \text{for all } t \in \mathbb{F}_q \setminus \{0\}.$$

Apply the unique Reed–Solomon decoder to these $q-1$ values at the nonzero evaluation points $t$, for degree at most $d$ and error budget $\lfloor (q-d-2)/2 \rfloor$. Return the decoded polynomial's value at 0, using a fixed default on failure.

For each nonzero $t$, $x + tv$ is uniform in $\mathbb{F}_q^m$ even though $x$ is fixed. If $B$ is the number of erroneous queried values, linearity of expectation gives $\mathbb{E}B \le (q-1)/12$. No independence among the errors is needed. Unique decoding succeeds whenever $2B < (q-1) - d$, by Theorem 7.9 with $n = q-1, k = d+1$. Markov's inequality bounds the failure probability by

$$\Pr\left[B \ge \frac{q-1-d}{2}\right] \le \frac{2\mathbb{E}B}{q-1-d} \le \frac{q-1}{6(q-1-d)} \le \frac{1}{4}.$$

On success the decoded polynomial is $P(x+tv)$, whose value at 0 is $P(x)$. The argument holds separately for every $x$, including points where $w(x)$ itself is corrupted. □

The queries exclude $t = 0$: the decoder reconstructs the value at $x$ from other evaluations on the restriction. Allowing $v = 0$ keeps every query uniform; the degenerate restriction is covered by the same estimate.

## 7.10 Optional: local list decoding Reed–Muller codes

Suppose now that $w$ agrees with a polynomial $P$ on only an $\alpha$ fraction of the space. Different restrictions may have different lists of nearby polynomials. To make their choices consistent, give the decoder one value $a = P(z)$ at a fixed *anchor* point $z$. The setup stage produces a short list of possible anchor values. One of them will select an approximate copy of $P$; local correction then removes its remaining errors.

**Theorem 7.13** (A list of local decoding procedures)**.** *Let $m \geq 1$, $d \geq 1$, $0 < \alpha \leq 1$, and $q \geq 4096(d+1)/\alpha^2$. Given oracle access to any $w : \mathbb{F}_q^m \to \mathbb{F}_q$, a randomized setup makes at most $q$ queries and outputs at most $\ell = \lceil 4/\alpha \rceil$ decoding procedures. For every polynomial $P$ of total degree at most $d$ with $\Pr_x[w(x) = P(x)] \geq \alpha$, the setup has probability at least $15/16$ of outputting* one fixed procedure $A$ *such that*

$$\text{for every } x \in \mathbb{F}_q^m, \qquad \Pr[A^w(x) = P(x)] \geq 3/4.$$

*The last probability is over fresh randomness of $A$. Each call makes at most $q(q-1)$ queries to $w$ and runs in time polynomial in $m, q, d, 1/\alpha$.*

**The setup and the decoding procedures.**

1. Choose independent uniform $z, v \in \mathbb{F}_q^m$. Query the restriction $t \mapsto w(z + tv)$ at all $q$ field elements. Use the Reed–Solomon list decoder at agreement $\alpha/2$. Let $S$ be the set of values $h(0)$ taken by its candidate polynomials. There are at most $\ell$ such values.
2. For each $a \in S$, define a deterministic approximate oracle $g_{z,a}$. At $x = z$, return $a$. Otherwise query the line $L(t) = z + t(x - z)$ at all $q$ points and list-decode at agreement $\alpha/2$. If exactly one candidate $h$ satisfies $h(0) = a$, return $h(1)$; otherwise return 0.
3. Output, for each $a \in S$, the local corrector of Theorem 7.12 applied to the oracle $g_{z,a}$. Each of its $q - 1$ oracle calls is answered by the procedure in step 2, using at most $q$ queries to $w$.

All candidate lists are checked for their actual agreement, as in the Reed–Solomon decoder. Thus the list on a line is intrinsic to that line, independent of its affine parametrization.

*Proof.* Fix one qualifying polynomial $P$. We bound three ways the construction can fail.

*1. A random restriction has too little agreement.* For independent uniform $Z, V \in \mathbb{F}_q^m$, let

$$T = \frac{1}{q} \sum_{t \in \mathbb{F}_q} \mathbf{1}\{w(Z + tV) = P(Z + tV)\}.$$

Its mean is at least $\alpha$. For distinct $s, t$, the pair $(Z + sV, Z + tV)$ is uniform on $(\mathbb{F}_q^m)^2$: the linear change of variables is invertible. The indicators are therefore pairwise independent, so $\mathrm{Var}(T) \leq 1/q$. The second-moment inequality gives

$$\Pr[T < \alpha/2] \leq \frac{4}{q\alpha^2}. \tag{7.4}$$

In particular the setup omits the value $P(z)$ with probability at most this bound.

*2. The correct anchor can still leave an ambiguous line.* For a uniform anchor $z$, put $g_z = g_{z,P(z)}$. Also choose a uniform query point $x$, independently. When $x = z$, $g_z(x) = P(x)$. Otherwise,

condition on the affine line through $z, x$. Conditional on this line, the anchor $z$ is uniform on its $q$ points. Each incorrect candidate polynomial on the line agrees with the true restriction of $P$ at at most $d$ points. There are at most $\ell$ candidates, so the probability of any incorrect candidate matching $P(z)$ is at most $\ell d/q$. If there is no such collision and the true restriction is listed, step 2 returns $P(x)$.

The probability that the true restriction is not listed is at most (7.4): for independent uniform $z, x$, the pair $(z, x - z)$ is uniform, exactly as $(Z, V)$ above. Hence

$$\mathbb{E}_z \Pr_x[g_z(x) \neq P(x)] \leq \beta := \frac{4}{q\alpha^2} + \frac{\ell d}{q} \leq \frac{9}{4096}.$$

The last inequality uses $q\alpha^2 \geq 4096(d+1)$ and $\ell \leq 5/\alpha$. Another application of Markov gives

$$\Pr_z[\Pr_x[g_z(x) \neq P(x)] > 1/12] \leq 12\beta.$$

*3. Turn a good approximate oracle into a local decoder.* Except with probability at most

$$12\beta + \frac{4}{q\alpha^2} \leq \frac{112}{4096} = \frac{7}{256} < \frac{1}{16},$$

the setup includes $a = P(z)$ and its deterministic oracle $g_z$ differs from $P$ on at most a $1/12$ fraction of points. The local-correction theorem now applies to this *fixed* oracle: its associated output procedure returns $P(x)$ with probability at least $3/4$ for every $x$. The field-size assumption implies $d \leq (q-1)/3$, as required. This proves both the order of the quantifiers and the query bound. □

The setup probability is stated for each fixed qualifying $P$. Once a successful setup is fixed, the same candidate index works for every requested coordinate. Fresh decoding randomness is used for each call. For $q$ polynomial in $m$, the query counts are polynomial in $\log N = m \log q$.

**Sources.** The bivariate interpolation method is due to Sudan, *Maximum Likelihood Decoding of Reed Solomon Codes* (FOCS 1996), Sections 2–3. The random-line and anchor constructions are presented in Vadhan's *Pseudorandomness*, Sections 5.2.2, 7.5.1, and 7.6.3. The local list-decoding construction is from Sudan–Trevisan–Vadhan. The coefficient search is the Roth–Ruckenstein method; see *Efficient decoding of Reed–Solomon codes beyond half the minimum distance* (2000). Here field roots are found by enumeration, with running time polynomial in $q$.

### Checks and exercises

1. For $k = 3$, write the word of message $a = (1, 0, 1)$ with coordinates in lexicographic order. Check its weight. Explain why corruption of the constant coordinate $r = 0$ is detectable by a single read, while its correct value can be recovered without reading the oracle at all.

2. Compute every Fourier coefficient of the sign representation of the received word 0001 in the two-bit example. Check Parseval and identify the three messages of agreement $3/4$. Explain why a negative coefficient does not count toward this list.

3. For a fixed corruption set $B$, show that the two-query algorithm fails exactly when one of $R, R + x$ belongs to $B$ and the other does not. Express the exact error probability as $|B \triangle (B + x)|/N$, where $\triangle$ denotes symmetric difference, and recover the bound $2|B|/N$. Explain why the exact value can depend on $x$.

4. In the Reed–Solomon decoder prove why making $E$ monic excludes the useless solution $E = A = 0$. Show that if the promise is violated, the final validation prevents returning a word outside the requested radius, although it does not identify the originally transmitted message.

5. Over $\mathbb{F}_5$, run the coefficient search on $Q(X, Y) = (Y - X)(Y - 1 - 2X)$ for roots of degree at most one. Draw its search tree and verify the candidates by substitution.

## Proof of the binary Johnson bound

Fix any binary code $C \subseteq \{0,1\}^n$ of minimum distance at least $\delta n$, with $n \geq 1$ and $0 < \delta \leq 1/2$. Fix any received word $z \in \{0,1\}^n$ and $0 \leq \rho < 1/2$ such that $\delta > 2\rho(1-\rho)$. We prove

$$L := |\{c \in C : \operatorname{dist}(c,z) \leq \rho n\}| \leq \frac{\delta}{\delta - 2\rho(1-\rho)}.$$

The bound is immediate for $L \leq 1$, since the right-hand side is at least one. Assume $L \geq 2$ and write the candidate codewords as $c^{(1)}, \ldots, c^{(L)}$.

**Idea: count all pairwise disagreements in two ways.** Minimum distance forces each pair of candidates to disagree often. But all candidates are close to the same received word, which limits their total pairwise disagreements. Comparing these two counts bounds $L$.

*Proof.* Replace every candidate $c^{(u)}$ by $c^{(u)} + z$, with addition modulo two. This preserves all pairwise distances, and the new candidates each have Hamming weight at most $\rho n$. We retain the same notation for them. Let

$$S = \sum_{1 \leq u < v \leq L} \operatorname{dist}(c^{(u)}, c^{(v)}).$$

There are $\binom{L}{2}$ unordered pairs, each at distance at least $\delta n$, so

$$S \geq \delta n \binom{L}{2} = \frac{\delta n L(L-1)}{2}.$$

For the other count, arrange the candidates as the rows of an $L$-by-$n$ binary matrix. Let $t_j$ be the number of ones in column $j$. A pair differs in that column exactly when one row is among its $t_j$ ones and the other among its $L - t_j$ zeros. Thus column $j$ contributes exactly $t_j(L - t_j)$ unordered pairs. If $T = \sum_j t_j$ is the total number of ones in the matrix, then

$$S = \sum_{j=1}^{n} t_j(L - t_j) = LT - \sum_{j=1}^{n} t_j^2, \qquad T \leq L\rho n.$$

The column counts need not be equal. The elementary inequality

$$0 \leq \sum_{j=1}^{n} (t_j - T/n)^2 = \sum_{j=1}^{n} t_j^2 - \frac{T^2}{n}$$

gives $S \leq LT - T^2/n$. Put $p = T/(Ln)$, so $0 \leq p \leq \rho < 1/2$. Since $\rho(1-\rho) - p(1-p) = (\rho - p)(1 - \rho - p) \geq 0$, we obtain

$$S \leq nL^2 p(1-p) \leq nL^2 \rho(1-\rho).$$

Combining the lower and upper bounds and dividing by $nL/2 > 0$ gives

$$\delta(L-1) \leq 2L\rho(1-\rho), \qquad L\big(\delta - 2\rho(1-\rho)\big) \leq \delta.$$

The coefficient of $L$ is positive by hypothesis, so division proves the claimed bound. □

This argument uses only the binary alphabet, the minimum distance, and the distance of each candidate from $z$; the code need not be linear. For Hadamard, $\delta = 1/2$ and $\rho = 1/2 - \varepsilon$, so the bound becomes $L \leq (1/2)/(2\varepsilon^2) = 1/(4\varepsilon^2)$, exactly as in Theorem 7.5.