Introduction to Coding Theory **IST-UL, 2025/2026**

Week 5: List-decoding

Lecturer: João Ribeiro

Recommended reading: Essential Coding Theory, Chapter 7.

1 Hamming vs. Shannon

So far we have studied bounds for codes against adversarial errors (the *Hamming* setting) and probabilistic errors (the *Shannon* setting). Now is a great time to compare the results we have seen, and we will do that by comparing the Hamming setting for binary codes with what we know about binary symmetric channels.

Adversarial errors. Fix $p < 1/4$. We know that any binary code with block length n that corrects $t = pn$ adversarial errors must have relative distance $\delta > 2p$. Therefore, the asymptotic Hamming bound says that the rate of this code satisfies

$$R \leq 1 - h(\delta/2) \leq 1 - h(p).$$

On the other hand, the asymptotic Gilbert-Varshamov bound guarantees the existence of codes correcting t adversarial errors with rate at least $1 - h(\delta) \approx 1 - h(2p)$.

Random errors. In this case, Shannon's noisy channel coding theorem states that we can achieve any rate $R < 1 - h(p)$ with vanishing decoding error probability over the BSC with error probability p, and that we cannot do this for rates $R > 1 - h(p)$.

It is also not hard to show that correcting slightly more than pn adversarial errors is an easier task than coding reliably over the BSC with error probability p. More precisely, we have the following.

Theorem 1 *Fix an arbitrary $\gamma > 0$. Then, any family of binary codes with relative distance $\delta \geq 2(p + \gamma)$ can be used for reliable communication over the BSC with error probability p.*

Proof: You will figure this out in your homework! ■

The take-home message. There is a gap between what we know we can achieve in the Hamming setting (rate $1 - h(2p)$) and what we can achieve in the easier Shannon setting (rate $1 - h(p)$).

2 List-decoding

We will now study a relaxed decoding notion for adversarial errors, called *list-decoding*, that allows us to, in a sense, bridge the Hamming and Shannon settings. Our notion of decoding for adversarial errors thus far is *unique decoding*: given an output y corrupted by t errors, there should be exactly one codeword c such that $d_H(c, y)$. This allows the decoder to recover the correct transmission with certainty. But what if we allow the decoder to output a small list of candidate codewords?

Definition 1 (List-decodable code) *A code $\mathcal{C} \subseteq \Sigma^n$ is* (ρ, L)-list-decodable *if for any $y \in \Sigma^n$ we have*

$$|\{c \in \mathcal{C} : d_H(c, y) \leq \rho n\}| \leq L.$$

In words, if a code $\mathcal{C}$ is (ρ, L)-list-decodable, then given an output y corrupted by $t = \rho n$ errors, the decoder will output a list of at most L candidate codewords.

List-decoding was first studied by Elias and Wozencraft in the 1950s. When $L = 1$ we recover the notion of unique decoding we have studied so far. On the other extreme, when $L = |\Sigma|^n$ list-decoding is trivial. The following question arises: Can we achieve better rates than in the unique-decoding setting by allowing small lists of size $L > 1$?

A small list size is less useful than list size $L = 1$, but it still seems to be a useful guarantee. First, if the errors are random and the alphabet size is large enough, then with high probability the actual list size will be 1, in which case we manage to recover the correct codeword. Second, if we have some side information about the message being transmitted then we can use it to eliminate candidates from the list and find the correct codeword. If the list is small, then the amount of side information required to do this is also small.

Beyond being a natural coding-theoretic notion, list-decodable codes have found important applications beyond their original motivation in coding theory and in other areas, such as complexity theory. We may see some of these applications later on. If you would like to explore this now, see this nice overview.

3 The Johnson bound

We begin by proving the Johnson bound (for binary codes). This bound relates the minimum distance of a code to its list-decoding properties.

Theorem 2 (Johnson bound) *Consider the function*

$$J_2(\delta) = \frac{1}{2}(1 - \sqrt{1 - 2\delta}).$$

Let $\mathcal{C} \subseteq \{0,1\}^n$ be a code with minimum distance at least d. If $\rho < J_2(d/n)$, then $\mathcal{C}$ is $(\rho, 2dn)$-list-decodable.

Figure 1 plots $J_2(\delta)$ in blue. In particular, we can see that $J_2(\delta) > \delta/2$. This is relevant because at relative distance δ we can uniquely decode from a $\delta/2$-fraction of adversarial errors. So the Johnson bounds tells us that we can push beyond the unique decoding radius $\delta/2$ by accepting a list of size $O(dn)$.

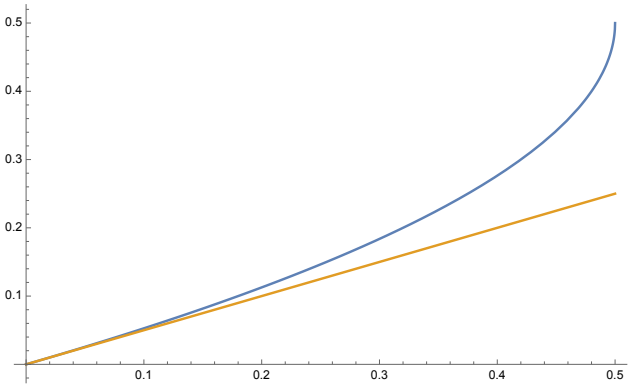

Figure 1: The function $J_2(\delta)$ plotted for $\delta \in [0, 1/2]$ (in blue) vs. $\delta/2$ (in orange).

Proof: We proceed by double counting.

Define $e = \rho n$ and fix an arbitrary $y \in \{0,1\}^n$. Then, our goal is to show that

$$M = |B(y, e) \cap \mathcal{C}| \leq 2dn.$$

Enumerate the elements of $B(y, e) \cap \mathcal{C}$ as $c_1, \ldots, c_M$. We will center the c_i's by defining $c_i' = c_i - y$. Note that $w_H(c_i') \leq e$ for all i, and also that $d_H(c_i', c_j') \geq d$ for all $i \neq j$. We will

$$S = \sum_{i \neq j} d_H(c_i', c_j')$$

in two different ways. This will yield the desired bound $M \leq 2dn$.

Since $d_H(c_i', c_j') \geq d$ for all $i \neq j$, we have

$$S \geq \binom{M}{2} \cdot d. \tag{1}$$

We now obtain an upper bound on S. Let A be the $n \times M$ matrix whose columns are $c_1, \ldots, c_M$. Let m_i denote the number of 1s in the i-th row of A. Then, the i-th row contributes $m_i(M - m_i)$ to S, corresponding to all 0-1 pairs in that row, and so

$$S = \sum_{i=1}^{n} m_i(M - m_i) = M \cdot \sum_{i=1}^{n} m_i - \sum_{i=1}^{n} m_i^2. \tag{2}$$

Define $\overline{e} = \frac{1}{M} \sum_{i=1}^{n} m_i$. Note that $\overline{e} \leq e$, since

$$\overline{e} = \frac{1}{M} \sum_{i=1}^{n} m_i = \frac{1}{M} \sum_{j=1}^{M} w_H(c_j') \leq \frac{eM}{M} = e. \tag{3}$$

We may now rewrite Equation (2) as

$$S = \overline{e}M^2 - \sum_{i=1}^{n} m_i^2.$$

An application of the Cauchy-Schwarz inequality yields[1]

$$\sum_{i=1}^{n} m_i^2 \geq \frac{1}{n}\left(\sum_{i=1}^{n} m_i\right)^2 = \frac{(\overline{e}M)^2}{n}. \tag{4}$$

Combining Equations (1), (2) and (4), we get that

$$\overline{e}M^2 - \frac{(\overline{e}M)^2}{n} \leq \binom{M}{2} \cdot d.$$

Together with Equation (3), this implies that

$$M \leq \frac{dn}{dn - 2n\overline{e} + 2\overline{e}^2} \leq \frac{2dn}{(n-2\overline{e})^2 - n(n-2d)} \leq \frac{2dn}{(n-2e)^2 - n(n-2d)}.$$

To wrap up the proof, it suffices to note that $\rho < J_2(d/n)$ implies that

$$(n-2e)^2 - n(n-2d) \geq 1,$$

and so $M \leq 2dn$, as desired. ∎

4 List-decoding capacity

The Johnson bound guarantees list size polynomial in n whenever the list-decoding radius ρ is small enough as a function of the relative distance δ of the code. However, the only information that the Johnson bound uses is the relative distance of the code, so there is hope that we can get stronger achievability results (larger rates and smaller list size).

We will now work out what rates are achievable with constant-sized lists.

Theorem 3 *Fix any $\rho \in (0, 1/2)$ and $\varepsilon > 0$. Then, the following hold for sufficiently large block length n:*

1. *If $R \leq 1 - h(\rho) - \varepsilon$, there exists a $(\rho, L = O(1/\varepsilon))$-list-decodable code of rate R;*
2. *If $R > 1 - h(\rho) + \varepsilon$, every (ρ, L)-list-decodable code of rate R has list size $L \geq 2^{\Omega(\varepsilon n)}$.*

Theorem 3 can be interpreted as identifying the "list-decoding capacity" as $1 - h(\rho)$:

[1] Cauchy-Schwarz states that $\langle x, z\rangle^2 \leq \|x\|_2^2 \cdot \|z\|_2^2$. Apply it to $x = (m_1, \ldots, m_n)$ and $z = (1/n, \ldots, 1/n)$.

- At rates below $1 - h(\rho)$ we can construct list-decodable codes from ρn errors with list size *independent of the block length* n;
- At rates above $1 - h(\rho)$ every code has exponentially large list size against ρn errors.

Similarly to the noisy channel coding theorem, this result is non-constructive. This immediately motivates the quest of constructing efficiently encodable and list-decodable codes with rates close to the list-decoding capacity and small list size. We will see more about the algorithmic challenges behind list-decoding later. If you would like to explore more about this now, see this nice (but relatively old) survey.

Note that the list-decoding capacity equals the capacity of the BSC with error probability ρ!

Proof: We divide the proof in two parts.

Part 1. We begin by establishing the first bullet via the probabilistic method. We sample a code $\mathcal{C} \subseteq \{0,1\}^n$ as follows: For each message $i \in \{1, \ldots, 2^{Rn}\}$, we sample $\mathsf{Enc}(i)$ uniformly at random from $\{0,1\}^n$. We will now show that this code has the desired list-decoding properties with positive probability.

Let the list size L be a free parameter. We wish to upper bound the probability that there exists $y \in \{0,1\}^n$ such that $|B(y, \rho n) \cap \mathcal{C}| > L$. Towards that, call $(y, i_1, \ldots, i_{L+1})$ a *bad event* if $\mathsf{Enc}(i_j) \in B(y, \rho n)$ for all $j \in \{1, \ldots, L+1\}$. Note that $\mathcal{C}$ fails to be (ρ, L)-list-decodable if and only if there exists such a bad event. For each j, we have

$$\Pr[\mathsf{Enc}(i_j) \in B(y, \rho n)] \leq \frac{\mathrm{Vol}_2(\rho n, n)}{2^n} \leq 2^{-(1-h(\rho))n}.$$

Since the encodings $\mathsf{Enc}(i_1), \ldots, \mathsf{Enc}(i_{L+1})$ are sampled independently, we have

$$\Pr[\forall j, \mathsf{Enc}(i_j) \in B(y, \rho n)] = \prod_{j=1}^{L+1} \Pr[\mathsf{Enc}(i_j) \in B(y, \rho n)] \leq 2^{-(1-h(\rho))n(L+1)}.$$

There are at most

$$2^n \cdot \binom{2^{Rn}}{L+1} \leq 2^{n+Rn(L+1)}$$

possible events $(y, i_1, \ldots, i_{L+1})$. Here, we used the inequality $\binom{a}{b} \leq a^b$. Therefore, by a union bound, the probability that there exists a bad event is at most

$$2^{n+Rn(L+1)} \cdot 2^{-(1-h(\rho))n(L+1)} = 2^{n-(1-h(\rho)-R)n(L+1)} \leq 2^{n-\varepsilon n(L+1)},$$

where we used the fact that $R \leq 1 - h(\rho) - \varepsilon$. To ensure that $2^{n-\varepsilon n(L+1)} < 1$, and hence that a (ρ, L)-list-decodable code of rate R exists, it suffices to take $L \geq 1/\varepsilon$.

We glossed over the fact that our sampling process may produce repeated codewords. If we are unlucky, $\mathcal{C}$ could end up having much fewer than 2^{Rn} codewords. In the problem set you will show that the probability that this happens is extremely small. This means that this sampling process produces, with high probability, a code $\mathcal{C}$ that simultaneously has rate R satisfying, say, $R \geq 1 - h(\rho) - 2\varepsilon$, and is $(\rho, L = 1/\varepsilon)$-list-decodable.

Part 2. We now establish the converse. Set $R = 1 - h(\rho) + \varepsilon$ and let $\mathcal{C}$ be a code of rate R. Our goal is to show that there exists $y \in \{0,1\}^n$ such that $|B(y, \rho n) \cap \mathcal{C}| \geq 2^{\Omega(\varepsilon n)}$. We will employ the probabilistic method and analyze a "random ball".

Sample y uniformly at random from $\{0,1\}^n$. For every fixed codeword $c \in \mathcal{C}$,

$$\Pr[c \in B(y, \rho n)] = \Pr[y \in B(c, \rho n)] = \frac{\mathrm{Vol}_2(\rho n, n)}{2^n} \geq 2^{-(1-h(\rho))n - o(n)}.$$

For each $c \in \mathcal{C}$, let Z_c denote the random variable that is 1 if $c \in B(y, \rho n)$ and is 0 otherwise. Note that

$$|B(y, \rho n) \cap \mathcal{C}| = \sum_{c \in \mathcal{C}} Z_c.$$

We now compute the expectation of this quantity. By linearity of expectation,

$$\begin{aligned}
\mathrm{E}[|B(y, \rho n) \cap \mathcal{C}|] &= \sum_{c \in \mathcal{C}} \mathrm{E}[Z_c] \\
&= \sum_{c \in \mathcal{C}} \Pr[c \in B(y, \rho n)] \\
&\geq |\mathcal{C}| \cdot 2^{-(1-h(\rho))n - o(n)} \\
&= 2^{\varepsilon n - o(n)} \\
&= 2^{\Omega(\varepsilon n)}.
\end{aligned}$$

We conclude that the expected list size over a uniformly random choice of center y is $2^{\Omega(\varepsilon n)}$. Therefore, there exists at least a y such that $|B(y, \rho n) \cap \mathcal{C}| = 2^{\Omega(\varepsilon n)}$. ∎

List-decoding capacity vs. capacity of the BSC. We have noted above that the list-decoding capacity at radius ρ, which is $1 - h(\rho)$, equals the capacity of the BSC with error probability ρ. This holds more generally over q-ary alphabets, where we replace $h(\rho)$ by $h_q(\rho)$ (where $h_q(\cdot)$ is the q-ary entropy function) and the BSC by the q-ary symmetric channel (qSC) with error probability ρ which independently replaces each input symbol $x \in \{0, \ldots, q-1\}$ by a uniformly random symbol distinct from x with probability ρ.

It is natural to wonder whether we can connect codes achieving list-decoding capacity and codes achieving capacity on the qSC. Very recently, Pernice, Sprumont, and Wootters [PSW25] obtained some results about this connection. At a high level, linear codes with not-too-small minimum distance that achieve list-decoding capacity also achieve capacity on the BSC. The converse to this statement is not true.

5 Do linear codes achieve list-decoding capacity?

We know that for any $\varepsilon > 0$ and decoding radius ρ there exists a (ρ, L)-list-decodable code of rate $1 - h(\rho) - \varepsilon$ and list size $L \leq 2/\varepsilon$, and that codes with rate above $1 - h(\rho)$ have exponentially large list size. Like we did for the GV bound, we can ask whether binary linear codes achieve

list-decoding capacity with small list size. This turns out to be a surprisingly difficult question. Guruswami, Håstad, Sudan, and Zuckerman [GHSZ02] gave a non-obvious but beautifully simple argument showing that there exist binary linear codes achieving list-decoding capacity, which we will see here. Surprisingly, this argument does not extend to larger alphabets. Understanding the list-decoding properties of linear codes (and other related properties, such as *list-recovery*) is still a topic of very active research.

Theorem 4 ([GHSZ02]) *Fix any $\rho \in (0, 1/2)$ and $\varepsilon > 0$. Then, for all sufficiently large block lengths n there exists a binary linear code $\mathcal{C}$ of rate $1 - h(\rho) - \varepsilon$ that is $(\rho, L = 1 + 1/\varepsilon)$-list-decodable.*

Proof: The proof is a non-obvious application of the probabilistic method. In short, we iteratively randomly sample new basis vectors for our linear code, and analyze the behavior of a carefully crafted potential as the sampling process progresses.

Set $k = (1 - h(\rho) - \varepsilon)n$ to be the dimension of our binary linear code. We sample the binary linear code $\mathcal{C}$ by iteratively sampling each basis vector v_i for $i = 1, \ldots, k$ uniformly at random from $\{0,1\}^n$ subject to $v_i \notin \mathcal{C}_i = \mathrm{span}(v_1, \ldots, v_{i-1})$, where

$$\mathrm{span}(v_1, \ldots, v_{i-1}) = \left\{ \sum_{j=1}^{i-1} \alpha_j v_j : \alpha_1, \ldots, \alpha_{i-1} \in \{0,1\} \right\}$$

is the subspace spanned by $v_1, \ldots, v_{i-1}$. Initially, $\mathcal{C}_0 = \{0\}$ is the trivial linear code. This guarantees that the v_i's are linearly independent, and so, after k iterations, $\mathcal{C}$ is indeed a binary linear code of dimension k.

To a code $\mathcal{C}$ we associate the potential $\Phi_{\mathcal{C}}$ defined as

$$\Phi_{\mathcal{C}} = 2^{-n} \sum_{x \in \{0,1\}^n} 2^{\varepsilon n L_{\mathcal{C}}(x)},$$

where $L_{\mathcal{C}}(x) = |B(x, \rho n) \cap \mathcal{C}|$. Note that

$$\Phi_{\mathcal{C}_0} = 2^{-n}(2^{\varepsilon n} \cdot |B(0, \rho n)| + 1 \cdot (2^n - |B(0, \rho n)|) \leq 1 + \frac{2^{\varepsilon n}|B(0, \rho n)|}{2^n} \leq 1 + 2^{-n(1-h(\rho)-\varepsilon)}. \quad (5)$$

Our goal will be to bound the expected growth of $\Phi_{\mathcal{C}}$ as we increase the dimension of $\mathcal{C}$. This will then allow us to bound the expected maximum list size.

Fix $i \in \{1, \ldots, k\}$ and suppose we have already sampled $v_1, \ldots, v_i$. We now bound $\mathrm{E}[\Phi_{\mathcal{C}_{i+1}}]$ as a

function of $\Phi_{\mathcal{C}_i}$, where the expectation is taken over the sampling of v_{i+1}. We have

$$\begin{aligned}
\mathrm{E}[\Phi_{\mathcal{C}_{i+1}}] &= 2^{-n} \sum_{x\in\{0,1\}^n} \mathrm{E}_{v_{i+1}}\left[2^{\varepsilon n L_{\mathcal{C}_{i+1}}(x)}\right] \\
&\le 2^{-n} \sum_{x\in\{0,1\}^n} \mathrm{E}_{v_{i+1}}\left[2^{\varepsilon n(|B(x,\rho n)\cap\mathcal{C}_i|+|B(x,\rho n)\cap(\mathcal{C}_i+v_{i+1})|)}\right] \\
&= 2^{-n} \sum_{x\in\{0,1\}^n} \mathrm{E}_{v_{i+1}}\left[2^{\varepsilon n|B(x,\rho n)\cap\mathcal{C}_i|}\cdot 2^{\varepsilon n|B(x,\rho n)\cap(\mathcal{C}_i+v_{i+1})|}\right] \\
&= 2^{-n} \sum_{x\in\{0,1\}^n} 2^{\varepsilon n|B(x,\rho n)\cap\mathcal{C}_i|}\cdot \mathrm{E}_{v_{i+1}}\left[2^{\varepsilon n|B(x,\rho n)\cap(\mathcal{C}_i+v_{i+1})|}\right] \\
&= 2^{-n} \sum_{x\in\{0,1\}^n} 2^{\varepsilon n|B(x,\rho n)\cap\mathcal{C}_i|}\cdot \mathrm{E}_{v_{i+1}}\left[2^{\varepsilon n|B(x+v_{i+1},\rho n)\cap\mathcal{C}_i|}\right] \\
&= 2^{-n} \sum_{x\in\{0,1\}^n} 2^{\varepsilon n L_{\mathcal{C}_i}(x)}\cdot \mathrm{E}_{v_{i+1}}\left[2^{\varepsilon n L_{\mathcal{C}_i}(x+v_{i+1})}\right].
\end{aligned}$$

Let's try to simplify this last expression. If v_{i+1} was sampled uniformly at random from $\{0,1\}^n$, we would get that

$$\mathrm{E}_{v_{i+1}}\left[2^{\varepsilon n L_{\mathcal{C}_i}(x+v_{i+1})}\right] = 2^{-n}\sum_{y\in\{0,1\}^n} 2^{\varepsilon n L_{\mathcal{C}_i}(x+y)} = 2^{-n}\sum_{z\in\{0,1\}^n} 2^{\varepsilon n L_{\mathcal{C}_i}(z)} = \Phi_{\mathcal{C}_i},$$

and so we would conclude that $\mathrm{E}[\Phi_{\mathcal{C}_{i+1}}] \le \Phi_{\mathcal{C}_i}^2$. However, this is not quite true, since v_{i+1} is sampled uniformly from $\{0,1\}^n$ *conditioned on* $v_{i+1} \notin \mathcal{C}_i$. Nevertheless, since the probability that a uniformly random $y \in \{0,1\}^n$ lands in $\mathcal{C}_i$ is at most 2^{i-n}, the true conditional expectation satisfies

$$\mathrm{E}_{v_{i+1}}\left[2^{\varepsilon n L_{\mathcal{C}_i}(x+v_{i+1})}\right] \le \frac{\Phi_{\mathcal{C}_i}}{1-2^{i-n}},$$

and so

$$\mathrm{E}[\Phi_{\mathcal{C}_{i+1}}] \le \frac{\Phi_{\mathcal{C}_i}^2}{1-2^{i-n}} \tag{6}$$

for all i.

Applying Equation (6) iteratively k times starting at $\mathcal{C}_0$, we get that

$$\mathrm{E}[\Phi_{\mathcal{C}_k}] \le \frac{\Phi_{\mathcal{C}_0}^{2^k}}{\prod_{i=0}^{k-1}(1-2^{i-n})^{2^{n-i}}} \le \frac{(1+2^{-n(1-h(\rho)-\varepsilon)})^{2^k}}{\prod_{i=0}^{k-1}(1-2^{i-n})^{2^{n-i}}}, \tag{7}$$

where we used Equation (5).

We now simplify Equation (7). Using the fact that $(1+x)^a \le e^{ax}$ for all $x \in \mathbb{R}$ and $a > 0$ and $(1-x)^a \ge 1-ax$ for all $x < 1$ and $a \ge 1$, we have

$$\mathrm{E}[\Phi_{\mathcal{C}_k}] \le \frac{e^{2^{k-n(1-h(\rho)-\varepsilon)}}}{(1-2^{k-n})^k} \le \frac{e}{1-k2^{k-n}}.$$

For large enough n we have $1-k2^{k-n} > 1/2$, in which case we conclude that $\mathrm{E}[\Phi_{\mathcal{C}_k}] \leq 6$. In particular, this means that for n large enough there exists a linear code $\mathcal{C}$ of dimension $k = (1 - h(\rho) - \varepsilon)n$ such that

$$\Phi_{\mathcal{C}} = 2^{-n} \sum_{x \in \{0,1\}^n} 2^{\varepsilon n L_{\mathcal{C}}(x)} \leq 6. \tag{8}$$

We now show that this code has the desired list-decoding properties. From Equation (8), we conclude in particular that for every $x \in \{0,1\}^n$ we have

$$2^{\varepsilon n L_{\mathcal{C}}(x)} \leq 6 \cdot 2^n.$$

Taking logarithms on both sides and rearranging, we get that

$$L_{\mathcal{C}}(x) \leq \frac{1}{\varepsilon} + \frac{3}{\varepsilon n}$$

for every $x \in \{0,1\}^n$. When n is large enough the right-hand side is at most $1 + 1/\varepsilon$, as desired. ■

It is instructive to reflect on where the choice of $\Phi_{\mathcal{C}}$ came from. Our end goal was to show that $\mathcal{C}$ is (ρ, L)-list-decodable. This is equivalent to showing that $\max_{x \in \{0,1\}^n} L_{\mathcal{C}}(x) \leq L$. However, analyzing the expectation of a maximum can be very complicated. Therefore, we try upper bound this maximum by a function that behaves better under expectation.

References

[GHSZ02] Venkatesan Guruswami, Johan Håstad, Madhu Sudan, and David Zuckerman. Combinatorial bounds for list decoding. *IEEE Transactions on Information Theory*, 48(5):1021–1034, 2002.

[PSW25] Francisco Pernice, Oscar Sprumont, and Mary Wootters. List-decoding capacity implies capacity on the q-ary symmetric channel. In *57th Annual ACM Symposium on Theory of Computing (STOC 2025)*, pages 855–866, 2025.