

Reed-Muller Codes Achieve Capacity on Erasure Channels

In the previous lectures, you have seen that LDPC codes achieve capacity via spatial coupling. The proof of this fact is based on the analysis of the “density evolution” of the BP decoder. In the next lecture(s), you will also see that polar codes achieve capacity. There, the proof is baked into the construction of the code which transforms the i.i.d. copies of the transmission channel into polarized synthetic channels (i.e., either noiseless or completely noisy channels). In this lecture, we present a different recipe in order to achieve capacity which is uniquely based on the symmetry of the code. Morally, we will show that **symmetry alone guarantees asymptotically optimal performance**. A corollary of this fact is that **Reed-Muller codes are capacity-achieving** for the transmission over the BEC under MAP decoding.

1 Definition of Reed-Muller codes

Reed-Muller (RM) codes are among the oldest known codes. They were introduced by Muller in 1954 and, shortly thereafter, Reed proposed a majority logic decoder. Given m and $v \in \mathbb{N}$, a Reed-Muller code $\text{RM}(m, v)$ is a linear code of block length $n = 2^m$ and rate R , given by

$$R = \frac{\sum_{i=0}^v \binom{m}{i}}{2^m}.$$

It is well known that the minimum distance of this code is 2^{m-v} .

The generator matrix (i.e., the matrix whose rows are the generators of the linear space formed by the code) can be defined recursively as follows.

Let

$$G_m = \begin{bmatrix} G_{m-1} & 0 \\ G_{m-1} & G_{m-1} \end{bmatrix}, \quad (1)$$

with initial condition $G_0 = [1]$. Then, the generator matrix of the code $\text{RM}(m, v)$ is obtained by taking all the rows of G_m with Hamming weight at least 2^{m-v} .

The generator matrix of polar codes is also obtained by taking rows of G_m , but the selection rule is different.

2 Main result

The aim of this lecture is to prove the following result.

Theorem 1 (Kudekar, Kumar, Mondelli, Pfister, Sasoglu, Urbanke, 2016). *Any sequence of Reed-Muller codes with block lengths $n_m \rightarrow \infty$, and rates $R_m \rightarrow R$, for $R \in (0, 1)$, is capacity-achieving for the transmission over the BEC under bit-MAP decoding.*

Actually, we will prove something stronger: we will show that any code with “sufficient amount of symmetry” achieves capacity. Let us start by clarifying exactly what we mean by “sufficient amount of symmetry”.

3 Ingredient 1: Symmetry

Denote by S_n the symmetric group on n elements and recall that, for $n \in \mathbb{N}$, $[n]$ is a shorthand for $\{1, \dots, n\}$.

Definition 2 (Permutation Group). *The permutation group $\mathcal{G}$ of a binary code $\mathcal{C} \subseteq \{0, 1\}^n$ is defined as*

$$\mathcal{G} \triangleq \{\pi \in S_n \mid \pi(x) \in \mathcal{C} \text{ for all } x \in \mathcal{C}\}. \quad (2)$$

With an abuse of notation, we denote by $\pi(x)$ the vector of length n that is obtained by permuting the positions of x according to π . In words, the permutation group of a code is the set of permutations that map the code into itself.

Definition 3 (Transitivity). *Let $\mathcal{G}$ be a permutation group. Then,*

- a) $\mathcal{G}$ is transitive if for any $i, j \in [n]$, there exists $\pi \in \mathcal{G}$ such that $\pi(i) = j$;
- b) $\mathcal{G}$ is doubly transitive if for $i, j, k, \ell \in [n]$ with $i \neq k$ and $j \neq \ell$, there exists $\pi \in \mathcal{G}$ such that $\pi(i) = j$, $\pi(k) = \ell$.

The following is a classic result for Reed-Muller codes.

Lemma 4 (Kasami, Lin, Peterson, 1968). *The permutation group $\mathcal{G}$ of the Reed-Muller code $RM(m, v)$ is doubly transitive for any $m, v \in \mathbb{N}$.*

This is the only property of Reed-Muller codes that we need. In fact, we will show that any family of doubly transitive codes achieves capacity.

4 Ingredient 2: Sharp Thresholds

Consider a family of binary linear codes $\{\mathcal{C}_m\}$ and let $P_B(n, \varepsilon)$ be the error probability for the transmission of a code of block length n over a channel with parameter ε . The parameter ε represents the quality of the transmission channel and, to be concrete, we can think of the binary erasure channel with erasure probability $\varepsilon \in [0, 1]$, i.e., the BEC(ε), and to the binary symmetric channel with crossover probability $\varepsilon \in [0, 1/2]$, i.e., the BSC(ε).

For any reasonable decoding algorithm, $P_B(n, \varepsilon)$ is increasing in ε , as we expect that the error probability increases when the channel introduces more erasures or errors. Define $\varepsilon^*(n, \delta)$ as the channel parameter such that the error probability for the code of block length n is equal to δ , i.e., $P_B(n, \varepsilon^*(n, \delta)) = \delta$. We say that the error probability experiences a sharp threshold when

$$\lim_{n \rightarrow \infty} \varepsilon^*(n, 1 - \delta) - \varepsilon^*(n, \delta) = 0. \quad (3)$$

In words, (3) means that the error probability passes from δ to $1 - \delta$ in a window whose size vanishes with n .

Theorem 5 (Tillich, Zemor, 2000). *Consider the transmission of a binary linear code with block length n and minimum distance $d_{\min}$ over the BEC(ε) or over the BSC(ε). Let $P_B(n, \varepsilon)$ be the error probability under block-MAP decoding. Then,*

$$\varepsilon^*(n, 1 - \delta) - \varepsilon^*(n, \delta) \leq \frac{c_1(\delta)}{\sqrt{d_{\min}}}, \quad (4)$$

where $c_1(\delta)$ is a universal constant depending only on δ .

Note that the block-MAP decoder outputs the most likely codeword, as opposed to the bit-MAP decoder that outputs the most likely value for each bit position.

The result above essentially says that the error probability under block-MAP decoding experiences a sharp threshold for any family of codes such that $d_{\min}$ tends to infinity, as n grows large (see also Figure

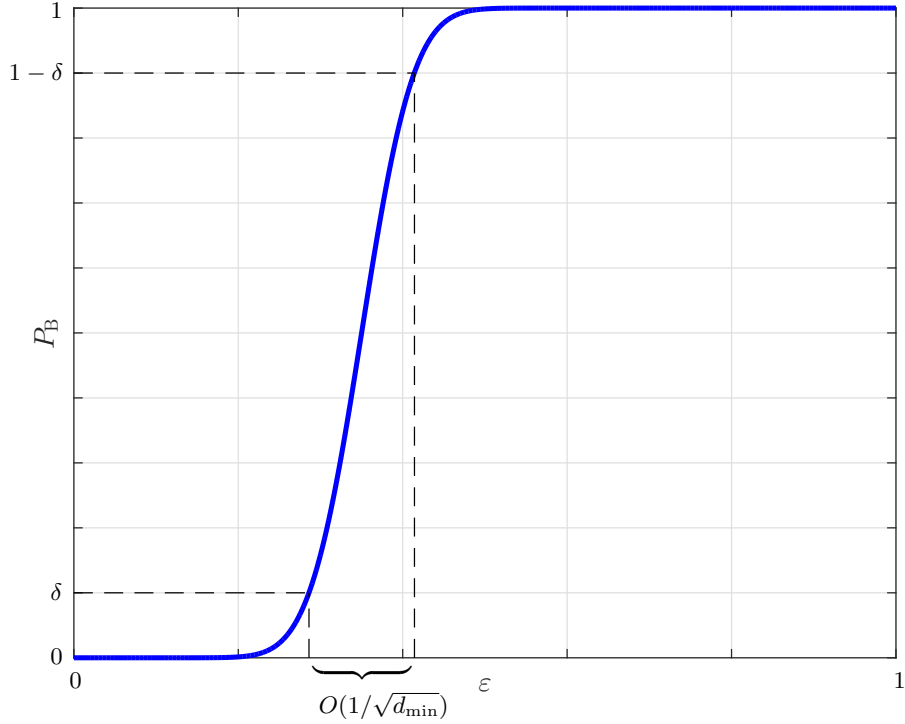


Figure 1: Illustration of the result of Theorem 5: the error probability under block-MAP decoding P_B passes from δ to $1 - \delta$ in a window of size $O(1/\sqrt{d_{\min}})$.

1). In addition, it gives a tight upper bound on the size of the window in which the transition takes place. The upper bound is tight in the following sense. If the code sequence has a minimum distance that is linear in the block length (up to logarithmic factors), then the transition occurs in roughly $O(1/\sqrt{n})$. This is as sharp as it can be, since the random variations of the channel are already of order $1/\sqrt{n}$: the number of erasures and errors tends to a Gaussian distribution with mean $n\varepsilon$ and standard deviation $\sqrt{n\varepsilon(1-\varepsilon)}$.

However, Theorem 5 does not let us to establish the location of the threshold. This happens quite frequently in theoretical computer science. On the one hand, we can apply the sharp transition framework in order to deduce that the transition width of certain functions goes to 0. On the other hand, establishing that the threshold exists, i.e., that the limit (3) exists, and determining its precise location is notoriously difficult.

In order to overcome these difficulties, we do not consider directly the error probability under MAP decoding, but a function closely related to it, i.e., the extrinsic information transfer (EXIT) function, as detailed in the next section. Furthermore, to show that the EXIT function exhibits a sharp transition, we resort to a more general result valid for *Bernoulli product measures of monotone symmetric sets*. Before stating this result, let us give some definitions. For $\omega, \omega' \in \{0, 1\}^M$, we write $\omega \preceq \omega'$ when $\omega_i \leq \omega'_i$ for all $i \in [M]$.

Definition 6 (Monotonicity). *Let $\Omega \subseteq \{0, 1\}^M$. We say that Ω is monotone if $\omega \in \Omega$ and $\omega \preceq \omega'$ imply that $\omega' \in \Omega$.*

In words, a subset Ω of the Hamming hypercube is monotone when, by adding more 1s to one of the elements of Ω , we remain in Ω .

Definition 7 (Symmetry). Let $\Omega \subseteq \{0, 1\}^M$. We say that Ω is symmetric if it is transitive in the sense of Definition 3.

In words, a subset Ω of the Hamming hypercube is transitive when, for any pair of indices $i, j \in [M]$, there exists a permutation that maps i into j and that keeps Ω invariant.

Definition 8 (Bernoulli Product Measure). Let $\Omega \subseteq \{0, 1\}^M$. The Bernoulli product measure of Ω with parameter ε is denoted by $\mu_\varepsilon(\Omega)$ and it is defined as

$$\mu_\varepsilon(\Omega) = \sum_{\omega \in \Omega} \varepsilon^{w_H(\omega)} (1 - \varepsilon)^{M - w_H(\omega)}, \quad (5)$$

where w_H denotes the Hamming weight.

In words, the Bernoulli product measure of a subset Ω of the Hamming hypercube is the probability that a vector whose components are i.i.d. Bernoulli(ε) random variables is contained in Ω .

Theorem 9 (Friedgut, Kalai, 1996). Let $\Omega \subseteq \{0, 1\}^M$ be monotone and symmetric and consider the Bernoulli product measure $\mu_\varepsilon(\Omega)$. Define $\varepsilon^*(\Omega, \delta)$ as the parameter such that $\mu_{\varepsilon^*}(\Omega) = \delta$. Then,

$$\varepsilon^*(\Omega, 1 - \delta) - \varepsilon^*(\Omega, \delta) \leq c_2 \frac{\ln(1/\delta)}{\ln M}, \quad (6)$$

where c_2 is a universal constant.

In words, consider the Bernoulli product measure $\mu_\varepsilon(\Omega)$ as a function of ε . Then, if Ω is monotone and symmetric, $\mu_\varepsilon(\Omega)$ passes from δ to $1 - \delta$ in a window of size $O(\ln(1/\delta)/\ln M)$.

5 Ingredient 3: EXIT Functions and Area Theorem

EXIT charts were introduced by ten Brink in the context of turbo decoding as a visual tool for understanding iterative decoding. For a given input bit, the EXIT function is defined to be the conditional entropy of the input bit, given the outputs associated with all other input bits. The average EXIT function is formed by averaging all of the bit EXIT functions. As you have seen in the previous lectures, these functions are also instrumental in the design and analysis of LDPC codes. The crucial property we exploit is the so called *area theorem*, originally proven by Ashikhmin, Kramer, and ten Brink and further generalized by Méasson, Montanari, and Urbanke. This result says that the area under the average EXIT function equals the rate of the code.

We will consider the transmission over the BEC(ε). For this channel, we can define EXIT functions as follows.

Definition 10 (EXIT Functions). Consider the transmission of a binary linear code $\mathcal{C}$ of block length n over the BEC(ε). Denote by $X = (X_1, \dots, X_n)$ and $Y = (Y_1, \dots, Y_n)$ the input and the output, respectively, of the channel and let $Y_{\sim i}$ be a shortcut for the vector $(Y_1, \dots, Y_{i-1}, Y_{i+1}, \dots, Y_n)$ that contains all outputs except the one at position i . Then, the EXIT function associated with bit i , denoted by $h_i(\varepsilon)$, and the average EXIT function, denoted by $h(\varepsilon)$, are defined by

$$\begin{aligned} h_i(\varepsilon) &= H(X_i | Y_{\sim i}), \\ h(\varepsilon) &= \frac{1}{n} \sum_{i=1}^n h_i(\varepsilon). \end{aligned} \quad (7)$$

Let us now investigate the relation between the error probability and the average EXIT function. We say that an erasure pattern covers a codeword of the code $\mathcal{C}$ when all the positions in which the codeword is 1 are erased. Let P_b be the error probability of the bit-MAP decoder. Clearly, for $i \in [n]$, the bit-MAP

decoder recovers bit i from the output $y = (y_1, \dots, y_n)$ if and only if the erasure pattern does not cover any codeword where bit i is non-zero. In this case, $H(X_i|Y = y) = 0$. Whenever bit i cannot be recovered uniquely, the linearity of the code implies that the set of codewords matching the unerased channel outputs has an equal number of 0s and 1s in bit position i . In this case, $H(X_i|Y = y) = 1$. Thus, the probability $P_{b,i}$ that the bit-MAP decoder cannot recover position i is equal to $H(X_i | Y)$. Since the average EXIT function can also be written in terms of entropies, it is not surprising that it is closely related to P_b , as proven by the following lemma.

Lemma 11 (EXIT Function and Bit-MAP Error Probability). *Consider the transmission of a code $\mathcal{C}$ of block length n over the $\text{BEC}(\varepsilon)$. Let $h(\varepsilon)$ be the average EXIT function and let P_b be the error probability under bit-MAP decoding. Then,*

$$P_b = \varepsilon \cdot h(\varepsilon) \quad (8)$$

Proof Recall that the $\text{BEC}(\varepsilon)$ outputs a “?” with probability ε and, otherwise, leaves the input unchanged. Thus, a simple calculation shows that

$$\begin{aligned} H(X_i | Y) &= \mathbb{P}(Y_i = X_i)H(X_i | Y_{\sim i}, Y_i = X_i) + \mathbb{P}(Y_i = ?)H(X_i | Y_{\sim i}, Y_i = ?) \\ &= (1 - \varepsilon) \cdot 0 + \varepsilon \cdot h_i(\varepsilon). \end{aligned} \quad (9)$$

Since $P_{b,i} = H(X_i | Y)$, by taking the average over the position i , the thesis follows. $\square$

We consider the EXIT function rather than the error probability because there is a conservation law that says the area under the average EXIT function is always equal to the rate of the code. Therefore, the area under $h(\varepsilon)$ is an invariant for any code of a given rate: changing the code just modifies the shape of the average EXIT function, while keeping fixed the area under the curve. These considerations are formalized by the so called *area theorem*, stated and proven below.

Theorem 12 (Area Theorem). *Consider the transmission of a code $\mathcal{C}$ of block length n and rate R over the $\text{BEC}(\varepsilon)$ and let $h(\varepsilon)$ be the average EXIT function. Then,*

$$\int_0^\varepsilon h(x) dx = \frac{H(X | Y)}{n}, \quad (10)$$

where $H(X | Y)$ is the conditional entropy of the codeword X given the observation Y at the receiver. In particular,

$$\int_0^1 h(x) dx = R. \quad (11)$$

Proof Although all bits are sent through the same channel $\text{BEC}(\varepsilon)$, it is convenient to imagine that bit i is sent through a $\text{BEC}(\varepsilon_i)$, where $\varepsilon_i = \varepsilon$ for all $i \in [n]$. Then, the derivative of the conditional entropy

$H(X | Y)$ can be written as

$$\begin{aligned}
\frac{dH(X | Y(\varepsilon_1, \dots, \varepsilon_n))}{d\varepsilon} &\stackrel{(a)}{=} \sum_{i=1}^n \frac{\partial H(X | Y(\varepsilon_1, \dots, \varepsilon_n))}{\partial \varepsilon_i} \Big|_{\varepsilon_j = \varepsilon, \forall j \in [n]} \\
&\stackrel{(b)}{=} \sum_{i=1}^n \frac{\partial H(X_i | Y(\varepsilon_1, \dots, \varepsilon_n))}{\partial \varepsilon_i} + \frac{\partial H(X_{\sim i} | X_i, Y(\varepsilon_1, \dots, \varepsilon_n))}{\partial \varepsilon_i} \Big|_{\varepsilon_j = \varepsilon, \forall j \in [n]} \\
&\stackrel{(c)}{=} \sum_{i=1}^n \frac{\partial H(X_i | Y(\varepsilon_1, \dots, \varepsilon_n))}{\partial \varepsilon_i} \Big|_{\varepsilon_j = \varepsilon, \forall j \in [n]} \\
&\stackrel{(d)}{=} \sum_{i=1}^n \frac{\partial (\varepsilon_i \cdot H(X_i | Y_{\sim i}, Y_i = ?))}{\partial \varepsilon_i} \Big|_{\varepsilon_j = \varepsilon, \forall j \in [n]} \\
&\stackrel{(e)}{=} \sum_{i=1}^n H(X_i | Y_{\sim i}, Y_i = ?) \\
&\stackrel{(f)}{=} nh(\varepsilon),
\end{aligned} \tag{12}$$

where (a) comes from the definition of total derivative, (b) comes from the chain rule for the conditional entropy, (c) uses that $H(X_{\sim i} | X_i, Y)$ does not depend on ε_i , (d) follows by expanding $H(X_i | Y)$ as in (9), (e) uses that $H(X_i | Y_{\sim i}, Y_i = ?)$ does not depend on ε_i ; and (f) follows from the definition (7) of average EXIT function.

By applying the fundamental theorem of calculus to (12), the result (10) immediately follows. In order to obtain (11), note that, when $\varepsilon = 1$, Y is independent from X , hence $H(X | Y) = H(X) = nR$. $\square$

6 The Proof

As described in the previous section, the EXIT function $h_i(\varepsilon)$ associated with bit i is the entropy of the input bit i given the outputs associated with all other input bits. This corresponds to the indirect recovery of x_i given the $n - 1$ received bits $y_{\sim i}$. We denote an erasure pattern by a binary vector $\omega \in \{0, 1\}^{n-1}$ that indicates the locations of the erased positions: a 1 denotes an erasure and a 0 denotes a non-erasure. The central object of our study is the set Ω_i of “bad” erasure patterns covering a codeword of $\mathcal{C}$ equal to 1 at position i . These erasure patterns are bad in the sense that they do not allow indirect recovery of the input bit i , i.e., the bit-MAP decoder cannot recover x_i from $y_{\sim i}$. Consequently, $h_i(\varepsilon)$ is encoded by Ω_i , as it is equal to the Bernoulli product measure of this set. These concepts are formalized by the definition and the lemma that immediately follow.

Definition 13 (Ω_i). *Consider the transmission of a binary linear code $\mathcal{C}$ of block length n over the BEC(ε). Then, Ω_i is defined as the set of all erasure patterns covering a codeword of $\mathcal{C}$ equal to 1 at position i , i.e.,*

$$\Omega_i = \{\omega \in \{0, 1\}^{n-1} \mid x_{\sim i} \preceq \omega, x_i = 1 \text{ for some } x \in \mathcal{C}\}. \tag{13}$$

Lemma 14 (Ω_i Encodes $h_i(\varepsilon)$). *Consider the transmission of a binary linear code $\mathcal{C}$ of block length n over the BEC(ε). Then, the set Ω_i defined in (13) contains all the erasure patterns such that it is not possible to recover the input bit x_i from the outputs $y_{\sim i}$ corresponding to all other positions. Furthermore, let $h_i(\varepsilon)$ be the EXIT function associated with bit i and $\mu_\varepsilon(\cdot)$ the Bernoulli product measure defined in (5). Then,*

$$h_i(\varepsilon) = \mu_\varepsilon(\Omega_i). \tag{14}$$

Proof As the code is linear and the channel is memoryless and symmetric, we can assume that the all-zero codeword was transmitted. Given an erasure pattern $\omega \in \{0, 1\}^{n-1}$, let $\mathcal{C}'$ denote the set of all codewords x that are compatible with the observation $y_{\sim i}$, i.e., all codewords for which $x_{\sim i} \preceq \omega$.

As the code is linear, so is $\mathcal{C}'$. This implies that if there exists an $x \in \mathcal{C}'$ with $x_i = 1$, then half of all codewords in $\mathcal{C}'$ have a 0 at position i , and the other half have a 1, which means that the indirect recovery of x_i given $y_{\sim i}$ fails. Whereas, if there is no $x \in \mathcal{C}'$ with $x_i = 1$, then all compatible codewords have a 0 at position i , which means that the indirect recovery of x_i succeeds. This argument proves that Ω_i is the set of all the erasure patterns that do not allow the indirect recovery of x_i from $y_{\sim i}$.

Since the channel is memoryless, an erasure pattern ω occurs with probability $\mu_\varepsilon(\omega)$. Hence, the claim (14) immediately follows. $\square$

As the discussion of Section 4 focuses on Bernoulli product measures of monotone symmetric sets, it is not surprising that the next step consists in proving that Ω_i is monotone and symmetric. The monotonicity follows basically by definition, whereas the symmetry comes from the fact that the code has a doubly transitive permutation group.

Lemma 15 (Ω_i Monotone). *Consider the transmission of a binary linear code $\mathcal{C}$ of block length n over the $\text{BEC}(\varepsilon)$. Then, the set Ω_i defined in (13) is monotone for any $i \in [n]$.*

Proof By Definition 6, we need to prove that if $\omega \in \Omega_i$ and $\omega \preceq \omega'$, then $\omega' \in \Omega_i$.

If $\omega \in \Omega_i$, then there exists $x \in \mathcal{C}$ so that $x_i = 1$ and $x_{\sim i} \preceq \omega$. Since, by assumption, $\omega \preceq \omega'$, it follows that $x_{\sim i} \preceq \omega'$, which implies that $\omega' \in \Omega_i$. $\square$

Lemma 16 (Ω_i Symmetric). *Consider the transmission of a binary linear code $\mathcal{C}$ of block length n with doubly transitive permutation group over the $\text{BEC}(\varepsilon)$. Then, the set Ω_i defined in (13) is symmetric for any $i \in [n]$.*

Proof By Definition 7, we need to prove that Ω_i is transitive.

As $\mathcal{C}$ has a doubly transitive permutation group, for any $j_1, j_2 \in [n] \setminus \{i\}$, there exists a permutation $\pi : [n] \rightarrow [n]$ such that $\pi(i) = i$, $\pi(j_1) = j_2$, and $\pi(x) \in \mathcal{C}$ for any $x \in \mathcal{C}$.

Consider the permutation $\hat{\pi}$ obtained by viewing the restriction of π to $[n] \setminus \{i\}$ as a permutation from $[n-1]$ to $[n-1]$. More formally, let $S_1 : [n-1] \rightarrow [n] \setminus \{i\}$ be defined as $S_1(k) = k$ for $k \in \{1, \dots, i-1\}$ and $S_1(k) = k+1$ for $k \in \{i, \dots, n-1\}$. Let $S_2 : [n] \setminus \{i\} \rightarrow [n-1]$ be defined as $S_2(k) = k$ for $k \in \{1, \dots, i-1\}$ and $S_2(k) = k-1$ for $k \in \{i+1, \dots, n\}$. Then, $\hat{\pi}(k) = S_2(\pi(S_1(k)))$.

Note that, by changing the choice of j_1 and j_2 , we generate the transitive group of permutations on $[n-1]$. Hence, in order to prove the claim, it suffices to show that if $\omega \in \Omega_i$, then $\hat{\pi}(\omega) \in \Omega_i$.

Recall that $\omega \in \Omega_i$ if there exists a codeword $x \in \mathcal{C}$ so that $x_i = 1$ and $x_{\sim i} \preceq \omega$. By construction of π , we have that $\pi(x) \in \mathcal{C}$ and $(\pi(x))_i = x_i = 1$. By construction of $\hat{\pi}$, we have that $\pi(x) \preceq \hat{\pi}(\omega)$. As a result, $\hat{\pi}(\omega) \in \Omega_i$ and the proof is complete. $\square$

Then, we show that the EXIT functions associated with the various bits of a transitive code are all identical.

Lemma 17 (h_i Independent of i). *Consider the transmission of a transitive binary linear code $\mathcal{C}$ of block length n over the $\text{BEC}(\varepsilon)$. Let $h_i(\varepsilon)$ be the EXIT function associated with bit i . Then, $h_i(\varepsilon) = h_j(\varepsilon)$ for all $i, j \in [n]$, i.e., $h_i(\varepsilon)$ is independent of i .*

Proof Since $\mathcal{C}$ is transitive, there exists a permutation $\pi : [n] \rightarrow [n]$ so that $\pi(i) = j$, and $\pi(x) \in \mathcal{C}$ for any $x \in \mathcal{C}$. The idea is that π maps the elements of Ω_i into the elements of Ω_j .

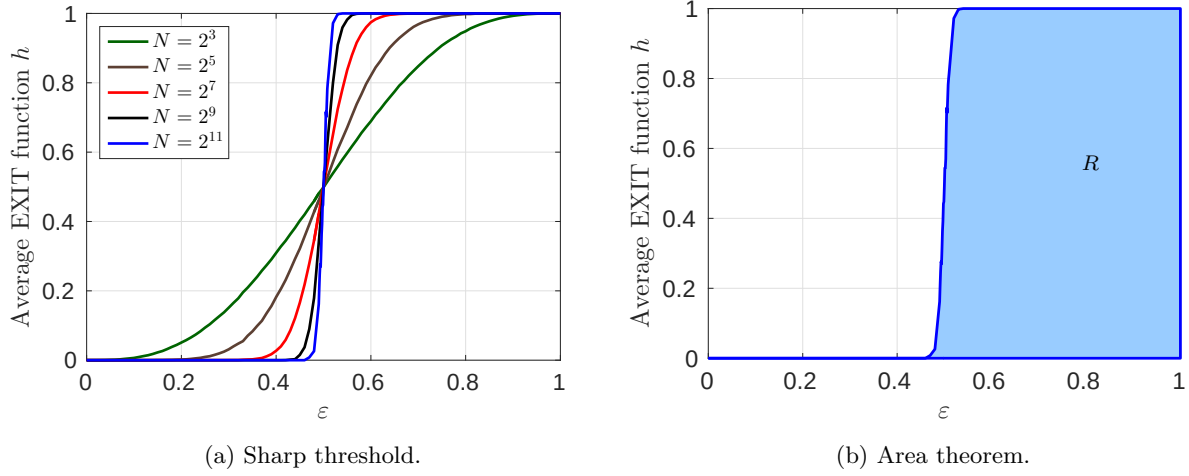


Figure 2: Proof by pictures of Theorem 1: in the left figure, we plot the average EXIT function of a sequence of codes with doubly transitive permutation group, e.g., of Reed-Muller codes, as the block length n increases; in the right figure, we indicate that the area under the average EXIT function is equal to the rate of the code.

More specifically, pick $\omega \in \Omega_i$. Note that ω comes from an erasure pattern on the transmitted codeword, call it $\hat{\omega} \in \{0, 1\}^n$, from which we have removed the observation i . Define $\hat{\pi}(\omega) \in \{0, 1\}^{n-1}$ as the erasure pattern obtained by removing the observation j to $\pi(\hat{\omega})$. Since $\omega \in \Omega_i$, there exists a codeword x so that $x_i = 1$ and $x_{\sim i} \preceq \omega$. By definition of π and $\hat{\pi}$, we have that $(\pi(x))_j = 1$ and $(\pi(x))_{\sim j} \preceq \hat{\pi}(\omega)$. As a result, $\hat{\pi}(\omega) \in \Omega_j$.

As $\omega \in \Omega_i$ implies that $\hat{\pi}(\omega) \in \Omega_j$, we can think of the map $\hat{\pi}$ as going from Ω_i to Ω_j . This map is injective and it preserves the Hamming weight, i.e., $w_H(\omega) = w_H(\hat{\pi}(\omega))$ for any $\omega \in \Omega_i$, which implies that $\mu_\varepsilon(\Omega_i) \leq \mu_\varepsilon(\Omega_j)$. By repeating the same argument with the indices i and j exchanged, we conclude that $\mu_\varepsilon(\Omega_i) = \mu_\varepsilon(\Omega_j)$ and the thesis follows from (14). $\square$

Finally, we are ready to prove our main result, i.e., Theorem 1. In a nutshell, the proof follows by assembling correctly the ingredients that we have described so far and, at a high level, it can be summarized by Figure 2. On the left side, we show that the average EXIT function becomes steeper and steeper as the block length increases, i.e., it experiences a sharp transition as predicted by Theorem 9. On the right side, we show that the threshold must occur at capacity from the area theorem, i.e., Theorem 12.

Proof Let $\{\mathcal{C}_m\}$ be a sequence of codes with doubly transitive permutation groups such that their block lengths $n_m \rightarrow \infty$ and rates $R_m \rightarrow R$. Consider the transmission over the $\text{BEC}(\varepsilon)$ and bit-MAP decoding. We say that the family $\{\mathcal{C}_m\}$ achieves capacity if the error probability tends to 0 for any $R < 1 - \varepsilon$.

By Lemma 11, it suffices to show that the sequence of average EXIT functions tends to 0 for any $\varepsilon < 1 - R$.

By Lemma 15 and 16, the set Ω_i of Definition 13 is monotone and symmetric. Furthermore, by Lemma 14, its Bernoulli product measure $\mu_\varepsilon(\Omega_i)$ is equal to the EXIT function $h_i^{(m)}(\varepsilon)$ associated with the bit i of the code $\mathcal{C}_m$. Therefore, Theorem 9 bounds the window size in which $h_i^{(m)}(\varepsilon)$ passes from δ_m to $1 - \delta_m$. More specifically, we have that if $h_i^{(m)}(\bar{\varepsilon}) = 1 - \delta_m$, then $h_i^{(m)}(\underline{\varepsilon}) \leq \delta_m$ for

$$\bar{\varepsilon} = \underline{\varepsilon} + c_2 \frac{\ln(1/\delta_m)}{\ln(n_m - 1)},$$

where c_2 is a universal constant.

Lemma 17 implies that $h_i^{(m)}(\varepsilon)$ is independent of i , thus it is equal to the average EXIT function $h^{(m)}(\varepsilon)$ of the code $\mathcal{C}_m$. By definition of $\bar{\varepsilon}$, we have that

$$\int_0^1 h^{(m)}(\varepsilon) \, d\varepsilon \geq (1 - \bar{\varepsilon})(1 - \delta_m) \geq \left(1 - \underline{\varepsilon} - c_2 \frac{\ln(1/\delta_m)}{\ln(n_m - 1)}\right) (1 - \delta_m). \quad (15)$$

Furthermore, Theorem 12 gives that

$$\int_0^1 h^{(m)}(\varepsilon) \, d\varepsilon = R_m. \quad (16)$$

Combining (15) and (16), we obtain

$$\underline{\varepsilon} \geq 1 - \frac{R_m}{1 - \delta_m} - c_2 \frac{\ln(1/\delta_m)}{\ln(n_m - 1)}. \quad (17)$$

As $m \rightarrow \infty$, we have that $n_m \rightarrow \infty$ and $R_m \rightarrow R$. Thus, we can have $\delta_m \rightarrow 0$ and, at the same time, $\underline{\varepsilon}$ arbitrarily close to $1 - R$, which suffices to prove the claim. $\square$