

Efficiently decoding Reed-Muller codes from random errors

Ramprasad Saptharishi

TIFR

Amir Shpilka

Tel Aviv University

Ben Lee Volk

Tel Aviv University

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f , i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0, 1\}^m$.

1	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f , i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0, 1\}^m$.

1	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Can you recover f ?

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f , i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0, 1\}^m$.

1	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Can you recover f ?

Of course! Just interpolate.

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f , i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0, 1\}^m$. But she corrupts 49% of the bits.

1	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Can you recover f ?

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f , i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0, 1\}^m$. But she corrupts 49% of the bits.

1	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Can you recover f ?

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f , i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0, 1\}^m$. But she corrupts 49% of the bits.

0	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Can you recover f ?

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f , i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0, 1\}^m$. But she corrupts 49% of the bits.

0	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Can you recover f ?

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f , i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0, 1\}^m$. But she corrupts 49% of the bits.

0	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Can you recover f ?

Impossible if errors are adversarial...

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f , i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0, 1\}^m$. But she corrupts 49% of the bits **randomly**.

0	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Can you recover f ?

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f , i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0, 1\}^m$. But she corrupts 49% of the bits randomly.

0	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Can you recover f ?

This talk: Yes, and we can do it efficiently!

A Puzzle

Your friend picks a polynomial $f(\mathbf{x}) \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree $r \approx \sqrt[3]{m}$.

She gives you the entire truth-table of f , i.e. the value of $f(\mathbf{v})$ for every $\mathbf{v} \in \{0, 1\}^m$. But she corrupts 49% of the bits randomly.

0	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Can you recover f ?

This talk: Yes, and we can do it efficiently!

(“Efficiently decoding Reed-Muller codes from random errors”)

Reed-Muller Codes: $RM(m, r)$

Message: A degree polynomial $f \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree at most r .

Encoding: The evaluation of f on all points in $\{0, 1\}^m$.

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 + x_3 x_4$$



0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Reed-Muller Codes: $RM(m, r)$

Message: A degree polynomial $f \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree at most r .

Encoding: The evaluation of f on all points in $\{0, 1\}^m$.

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 + x_3 x_4$$



0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

A **linear code**, with

Reed-Muller Codes: $RM(m, r)$

Message: A degree polynomial $f \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree at most r .

Encoding: The evaluation of f on all points in $\{0, 1\}^m$.

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 + x_3 x_4$$



0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

A **linear code**, with

- ▶ **Block Length:** $2^m := n$

Reed-Muller Codes: $RM(m, r)$

Message: A degree polynomial $f \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree at most r .

Encoding: The evaluation of f on all points in $\{0, 1\}^m$.

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 + x_3 x_4$$



0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

A **linear code**, with

- ▶ **Block Length:** $2^m := n$
- ▶ **Distance:** 2^{m-r} (lightest codeword: $x_1 x_2 \dots x_r$)

Reed-Muller Codes: $RM(m, r)$

Message: A degree polynomial $f \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree at most r .

Encoding: The evaluation of f on all points in $\{0, 1\}^m$.

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 + x_3 x_4$$



0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

A **linear code**, with

- ▶ **Block Length:** $2^m := n$
- ▶ **Distance:** 2^{m-r} (lightest codeword: $x_1 x_2 \dots x_r$)
- ▶ **Dimension:** $\binom{m}{0} + \binom{m}{1} + \dots + \binom{m}{r} =: \binom{m}{\leq r}$

Reed-Muller Codes: $RM(m, r)$

Message: A degree polynomial $f \in \mathbb{F}_2[x_1, \dots, x_m]$ of degree at most r .

Encoding: The evaluation of f on all points in $\{0, 1\}^m$.

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 + x_3 x_4$$



0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

A **linear code**, with

- ▶ **Block Length:** $2^m := n$
- ▶ **Distance:** 2^{m-r} (lightest codeword: $x_1 x_2 \dots x_r$)
- ▶ **Dimension:** $\binom{m}{0} + \binom{m}{1} + \dots + \binom{m}{r} =: \binom{m}{\leq r}$
- ▶ **Rate:** dimension/block length = $\binom{m}{\leq r} / 2^m$

Reed-Muller Codes: $RM(m, r)$

Generator Matrix: evaluation matrix of $\deg \leq r$ monomials:

$$M \xrightarrow{\quad \text{---} \quad} \left(\begin{array}{c} \mathbf{v} \in \mathbb{F}_2^m \\ \downarrow \\ M(\mathbf{v}) \end{array} \right) := E(m, r)$$

Reed-Muller Codes: $RM(m, r)$

Generator Matrix: evaluation matrix of $\deg \leq r$ monomials:

$$M \xrightarrow{\quad \text{---} \quad} \left(\begin{array}{c} \mathbf{v} \in \mathbb{F}_2^m \\ \vdots \\ M(\mathbf{v}) \end{array} \right) := E(m, r)$$

(Every codeword is spanned by the rows.)

Reed-Muller Codes: $RM(m, r)$

Generator Matrix: evaluation matrix of $\deg \leq r$ monomials:

$$M \dashrightarrow \left(\begin{array}{c} \mathbf{v} \in \mathbb{F}_2^m \\ \downarrow \\ M(\mathbf{v}) \end{array} \right) := E(m, r)$$

(Every codeword is spanned by the rows.)

Also called the *inclusion matrix*
($M(\mathbf{v}) = 1$ if and only if " $M \subset \mathbf{v}$ ").

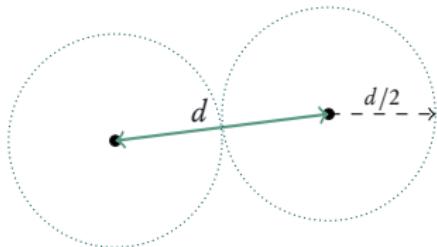
Decoding $RM(m, r)$

Worst Case Errors:



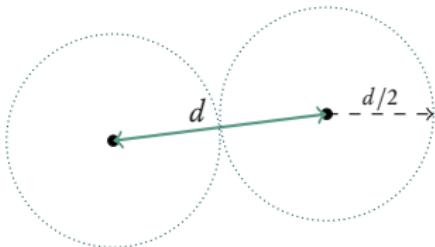
Decoding $RM(m, r)$

Worst Case Errors: Up to $d/2$ ($d = 2^{m-r}$ is minimal distance).



Decoding $RM(m, r)$

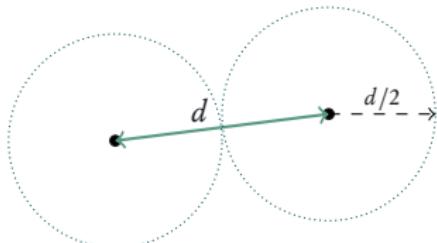
Worst Case Errors: Up to $d/2$ ($d = 2^{m-r}$ is minimal distance).



(algorithm by [Reed54])

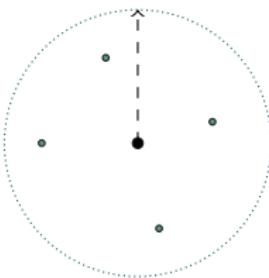
Decoding $RM(m, r)$

Worst Case Errors: Up to $d/2$ ($d = 2^{m-r}$ is minimal distance).



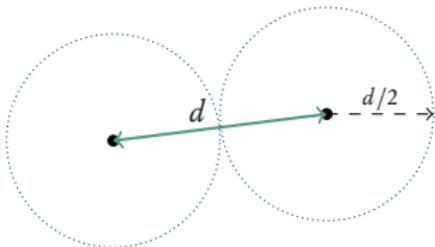
(algorithm by [Reed54])

List Decoding: max radius with constant # of words



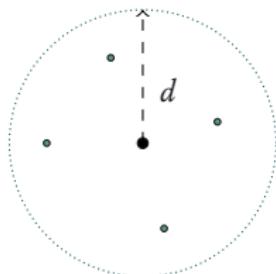
Decoding $RM(m, r)$

Worst Case Errors: Up to $d/2$ ($d = 2^{m-r}$ is minimal distance).



(algorithm by [Reed54])

List Decoding: max radius with constant # of words



[Gopalan-Klivans-Zuckerman08,
Bhowmick-Lovett15]

List decoding radius = d .

Two schools of study

- ▶ **Hamming Model** a.k.a worst-case errors
 - ▶ Generally the model of interest for complexity theorists,
 - ▶ Reed-Muller codes are not the best for these (far from optimal rate-distance tradeoffs).
- ▶ **Shannon Model** a.k.a random errors
 - ▶ The standard model for coding theorists,
 - ▶ Recent breakthroughs (e.g. Arikan's polar codes),

Two schools of study

- ▶ **Hamming Model** a.k.a worst-case errors
 - ▶ Generally the model of interest for complexity theorists,
 - ▶ Reed-Muller codes are not the best for these (far from optimal rate-distance tradeoffs).
- ▶ **Shannon Model** a.k.a random errors
 - ▶ The standard model for coding theorists,
 - ▶ Recent breakthroughs (e.g. Arikan's polar codes),
 - ▶ An ongoing research endeavor:
How do Reed-Muller codes perform in the Shannon model?

Models for random corruptions (channels)

Binary Erasure Channel — BEC(p)

Each bit independently replaced by ‘?’ with probability p

Models for random corruptions (channels)

Binary Erasure Channel — BEC(p)

Each bit independently replaced by ‘?’ with probability p

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

Models for random corruptions (channels)

Binary Erasure Channel — BEC(p)

Each bit independently replaced by ‘?’ with probability p

0	?	0	1	?	1	?	0
---	---	---	---	---	---	---	---

Models for random corruptions (channels)

Binary Erasure Channel – BEC(p)

Each bit independently replaced by ‘?’ with probability p

0	?	0	1	?	1	?	0
---	---	---	---	---	---	---	---

Binary Symmetric Channel – BSC(p)

Each bit independently flipped with probability p

Models for random corruptions (channels)

Binary Erasure Channel – BEC(p)

Each bit independently replaced by ‘?’ with probability p

0	?	0	1	?	1	?	0
---	---	---	---	---	---	---	---

Binary Symmetric Channel – BSC(p)

Each bit independently flipped with probability p

0	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

Models for random corruptions (channels)

Binary Erasure Channel – BEC(p)

Each bit independently replaced by ‘?’ with probability p

0	?	0	1	?	1	?	0
---	---	---	---	---	---	---	---

Binary Symmetric Channel – BSC(p)

Each bit independently flipped with probability p

0	1	0	1	1	1	0	0
---	---	---	---	---	---	---	---

Models for random corruptions (channels)

Binary Erasure Channel – BEC(p)

Each bit independently replaced by ‘?’ with probability p

0	?	0	1	?	1	?	0
---	---	---	---	---	---	---	---

Binary Symmetric Channel – BSC(p)

Each bit independently flipped with probability p

0	1	0	1	1	1	0	0
---	---	---	---	---	---	---	---

(almost) equiv: fixed number $t \approx pn$ of random errors

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Binary Erasure Channel – BEC(p)

Each bit independently replaced by ‘?’ with probability p

0	?	0	1	?	1	?	0
---	---	---	---	---	---	---	---

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Binary Erasure Channel – BEC(p)

Each bit independently replaced by ‘?’ with probability p

0	?	0	1	?	1	?	0
---	---	---	---	---	---	---	---

If X^n is transmitted to the channel and received as Y^n , how many bits of information about X^n do we get from Y^n ?

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Binary Erasure Channel – BEC(p)

Each bit independently replaced by ‘?’ with probability p

0	?	0	1	?	1	?	0
---	---	---	---	---	---	---	---

If X^n is transmitted to the channel and received as Y^n , how many bits of information about X^n do we get from Y^n ?

Intuitively, $(1 - p)n$.

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Binary Symmetric Channel – BSC(p)

Each bit independently flipped with probability p

0	0	0	1	1	1	1	0
*			*		*		*

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Binary Symmetric Channel – BSC(p)

Each bit independently flipped with probability p

0	0	0	1	1	1	1	0
*	*	*	*	*	*	*	

If X^n is transmitted to the channel and received as Y^n , how many bits of information about X^n do we get from Y^n ?

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

Binary Symmetric Channel – BSC(p)

Each bit independently flipped with probability p

0	0	0	1	1	1	1	0
*	*	*	*	*	*	*	

If X^n is transmitted to the channel and received as Y^n , how many bits of information about X^n do we get from Y^n ?

Intuitively, $(1 - H(p))n$. (as $\binom{n}{p_n} \approx 2^{H(p) \cdot n}$)

Channel Capacity

Question: Given a channel, what is the best rate we can hope for?

[Shannon48] Maximum rate that enables decoding (w.h.p.) is:

$$\begin{aligned} 1-p & \quad \text{for BEC}(p), \\ 1-H(p) & \quad \text{for BSC}(p). \end{aligned}$$

Codes achieving this bound called **capacity achieving**.

Category:Capacity-achieving codes

?

Help

From Wikipedia, the free encyclopedia

Pages in category "Capacity-achieving codes"

This category contains only the following page. This list may not reflect recent changes ([learn more](#)).

P

- [Polar code \(coding theory\)](#)

Categories: [Error detection and correction](#)

Motivating questions for this talk

How well does Reed-Muller codes perform in the Shannon Model?

ln BEC(p)?

ln BSC(p)?

Motivating questions for this talk

How well does Reed-Muller codes perform in the Shannon Model?

ln BEC(p)?

ln BSC(p)?

Are they as good as polar codes?

Dual of a code

Dual of a code

Any linear space can be specified by a **generating basis**, or a solution to a system of constraints.

Dual of a code

Any linear space can be specified by a **generating basis**, or a solution to a system of constraints.

$$\mathcal{C}^\perp = \{\mathbf{u} : \langle \mathbf{v}, \mathbf{u} \rangle = 0 \text{ for every } \mathbf{v} \in \mathcal{C}\}$$

Dual of a code

Any linear space can be specified by a **generating basis**, or a solution to a system of constraints.

$$\mathcal{C}^\perp = \{\mathbf{u} : \langle \mathbf{v}, \mathbf{u} \rangle = 0 \text{ for every } \mathbf{v} \in \mathcal{C}\}$$

Parity Check Matrix

(A basis for \mathcal{C}^\perp stacked as rows)

$$\boxed{PCM} \quad \boxed{v} = 0 \iff v \in \mathcal{C}$$

Linear codes and erasures

0	?	0	1	?	1	?	0
---	---	---	---	---	---	---	---

Question: When can we decode from a pattern of erasures?

Linear codes and erasures

0	?	0	1	?	1	?	0
---	---	---	---	---	---	---	---

Question: When can we decode from a pattern of erasures?

0	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---

0	1	0	1	0	1	0	0
---	---	---	---	---	---	---	---

Linear codes and erasures

0	?	0	1	?	1	?	0
---	---	---	---	---	---	---	---

Question: When can we decode from a pattern of erasures?

0	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---

0	1	0	1	0	1	0	0
---	---	---	---	---	---	---	---

0	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---

Linear codes and erasures

0	?	0	1	?	1	?	0
---	---	---	---	---	---	---	---

Question: When can we decode from a pattern of erasures?

0	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---

Linear codes and erasures

0	?	0	1	?	1	?	0
---	---	---	---	---	---	---	---

Question: When can we decode from a pattern of erasures?

0	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---

Decodable *if and only if* no non-zero codeword supported on erasures.

Linear codes and erasures

	?			?		?	
--	---	--	--	---	--	---	--

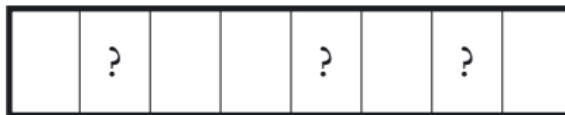
Question: When can we decode from a pattern of erasures?

0	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---

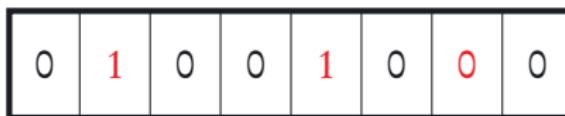
Decodable *if and only if* no non-zero codeword supported on erasures.

Depends only the erasure pattern

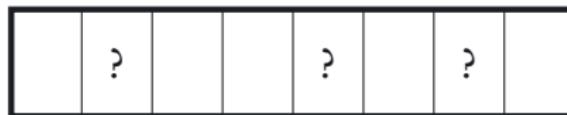
Linear codes and erasures



Question: When can we decode from a pattern of erasures?



Linear codes and erasures

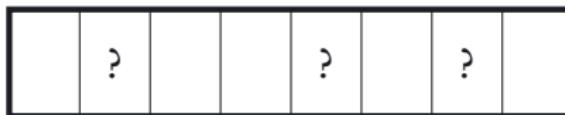


Question: When can we decode from a pattern of erasures?



Observation: A pattern of erasures are decodeable *if and only if* the corresponding columns of the **Parity Check Matrix** are linearly independent.

Linear codes and erasures



Question: When can we decode from a pattern of erasures?



Observation: A pattern of erasures are decodeable if and only if the corresponding columns of the **Parity Check Matrix** are linearly independent.

In order for a code to be good for BEC(p), the Parity Check Matrix of the code must be “robustly high-rank”.

Reed-Muller codes under erasures

Cool Fact

The **dual** of $RM(m, r)$ is $RM(m, r')$ where $r' = m - r - 1$.

Reed-Muller codes under erasures

Cool Fact

The **dual** of $RM(m, r)$ is $RM(m, r')$ where $r' = m - r - 1$.

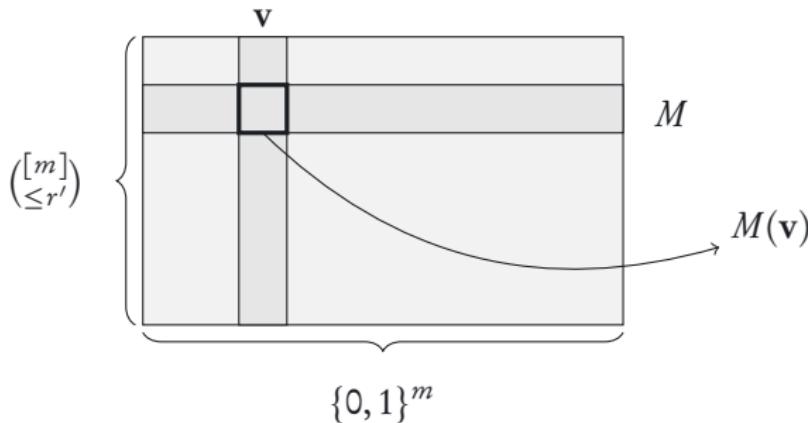
Hence, the **Parity Check Matrix** of $RM(m, r)$ is the generator matrix of $RM(m, r')$.

Reed-Muller codes under erasures

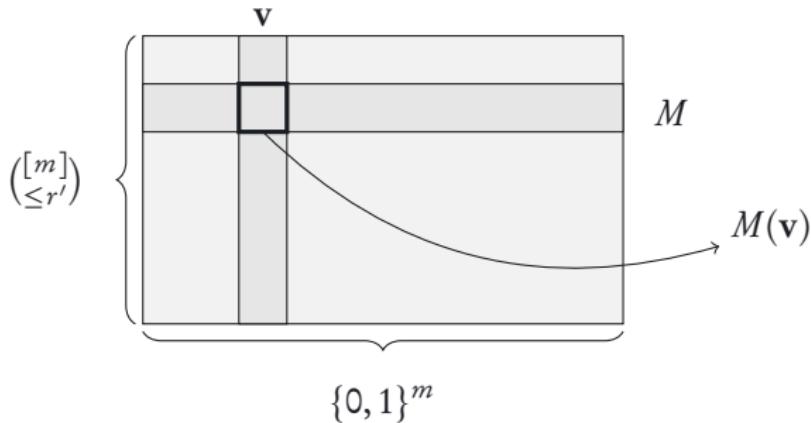
Cool Fact

The **dual** of $RM(m, r)$ is $RM(m, r')$ where $r' = m - r - 1$.

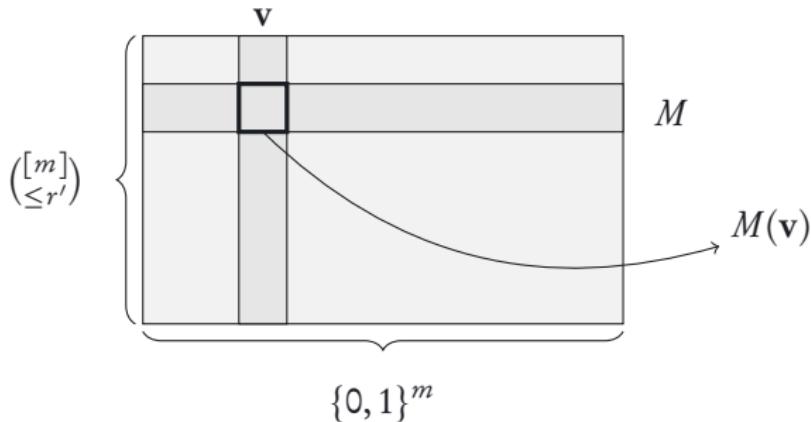
Hence, the **Parity Check Matrix** of $RM(m, r)$ is the generator matrix of $RM(m, r')$.



Reed-Muller codes under erasures

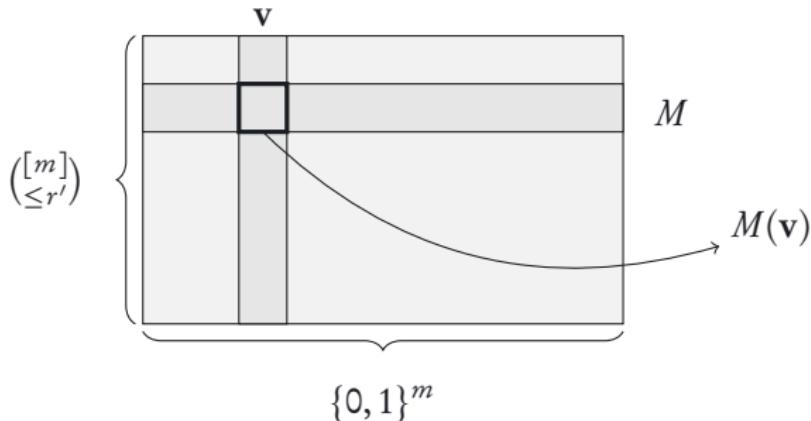


Reed-Muller codes under erasures

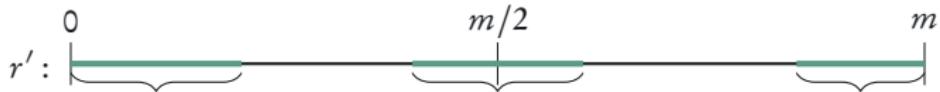


Question: Let $R = \binom{m}{\leq r'}$. Suppose you pick $(0.99)R$ columns at random. Are they linearly independent with high probability?

Reed-Muller codes under erasures



Question: Let $R = \binom{m}{\leq r'}$. Suppose you pick $(0.99)R$ columns at random. Are they linearly independent with high probability?

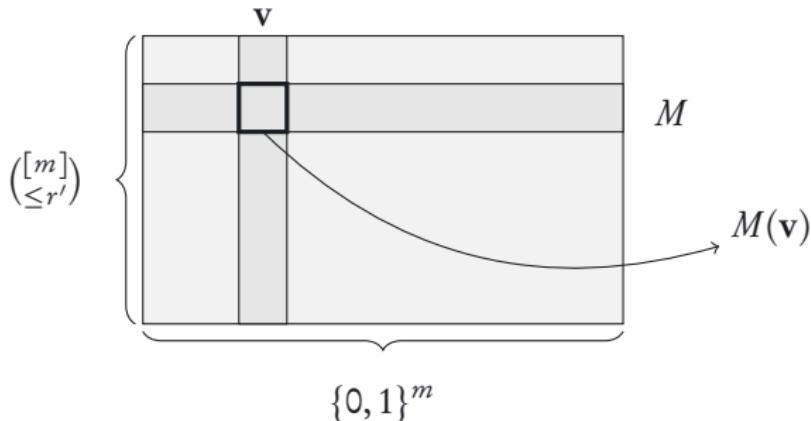


$o(\sqrt{m/\log m})$
[ASW-15]

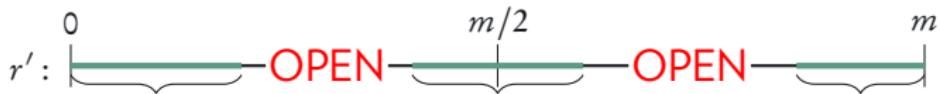
$O(\sqrt{m})$
[KMSU+KP-16]

$o(m)$
[ASW-15]

Reed-Muller codes under erasures



Question: Let $R = \binom{m}{\leq r'}$. Suppose you pick $(0.99)R$ columns at random. Are they linearly independent with high probability?



$o(\sqrt{m / \log m})$
[ASW-15]

$O(\sqrt{m})$
[KMSU+KP-16]

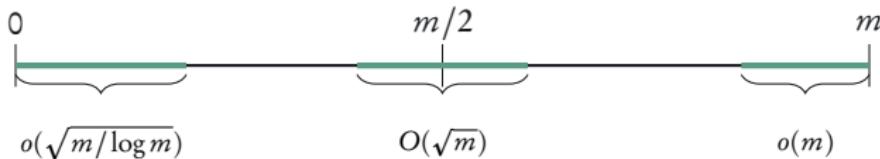
$o(m)$
[ASW-15]

From erasures to errors

Theorem: [ASW] Any pattern correctable from **erasures** in $RM(m, m - r - 1)$ is correctable from **errors** in $RM(m, m - 2r - 2)$.

From erasures to errors

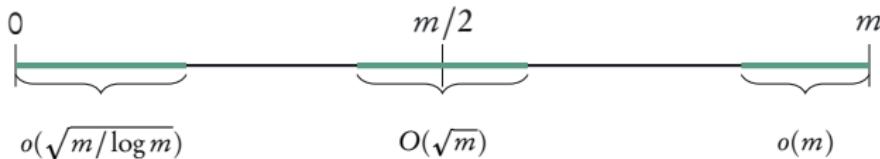
Remark: If r falls in the **green zone**, then $RM(m, m - r - 1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.



Theorem: [ASW] Any pattern correctable from **erasures** in $RM(m, m - r - 1)$ is correctable from **errors** in $RM(m, m - 2r - 2)$.

From erasures to errors

Remark: If r falls in the **green zone**, then $RM(m, m - r - 1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.

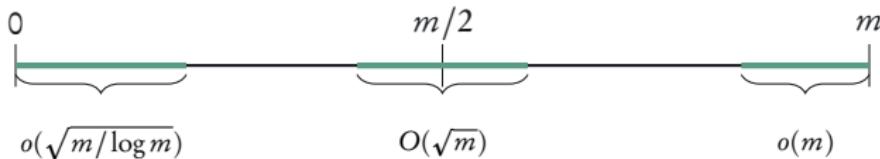


Theorem: [ASW] Any pattern correctable from **erasures** in $RM(m, m - r - 1)$ is correctable from **errors** in $RM(m, m - 2r - 2)$.

Corollary #1: (high-rate) Decodable from $(1 - o(1)) \binom{m}{\leq r}$ random errors in $RM(m, m - 2r)$ if $r = o(\sqrt{m / \log m})$
(min distance of $RM(m, m - 2r)$ is 2^{2r}).

From erasures to errors

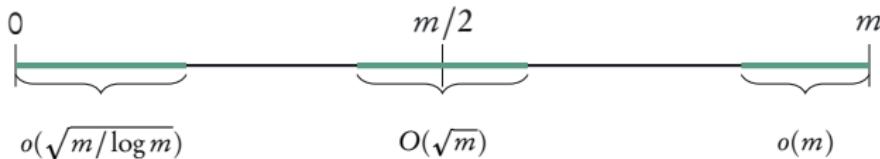
Remark: If r falls in the **green zone**, then $RM(m, m - r - 1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.



Theorem: [ASW] Any pattern correctable from **erasures** in $RM(m, m - r - 1)$ is correctable from **errors** in $RM(m, m - 2r - 2)$.

From erasures to errors

Remark: If r falls in the **green zone**, then $RM(m, m - r - 1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.



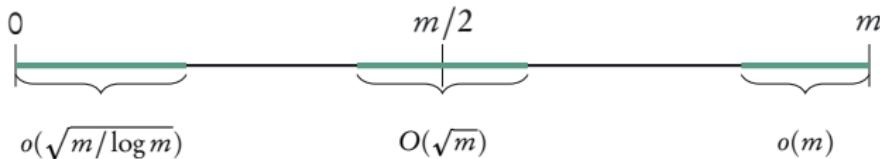
Theorem: [ASW] Any pattern correctable from **erasures** in $RM(m, m - r - 1)$ is correctable from **errors** in $RM(m, m - 2r - 2)$.

(If $r = \frac{m}{2} - o(\sqrt{m})$, then $m - 2r - 2 = o(\sqrt{m})$)

Corollary #2: (low-rate) Decodable from $(\frac{1}{2} - o(1))2^m$ random errors in $RM(m, o(\sqrt{m}))$ (min distance of $RM(m, \sqrt{m})$ is $2^{m-\sqrt{m}}$).

From erasures to errors

Remark: If r falls in the **green zone**, then $RM(m, m - r - 1)$ can correct $\approx \binom{m}{\leq r}$ random erasures.



Theorem: [ASW] Any pattern correctable from **erasures** in $RM(m, m - r - 1)$ is correctable from **errors** in $RM(m, m - 2r - 2)$.

(If $r = \frac{m}{2} - o(\sqrt{m})$, then $m - 2r - 2 = o(\sqrt{m})$)

Corollary #2: (low-rate) Decodable from $(\frac{1}{2} - o(1))2^m$ random errors in $RM(m, o(\sqrt{m}))$ (min distance of $RM(m, \sqrt{m})$ is $2^{m - \sqrt{m}}$).

[S-Shpilka-Volk]: Efficient decoding from errors.

What we want to prove

Theorem [S-Shpilka-Volk]

There exists an efficient algorithm with the following guarantee:

Given a corrupted codeword $\mathbf{w} = \mathbf{v} + \text{err}_S$ of $RM(m, m - 2r - 1)$,

if S happens to be a correctable erasure pattern in $RM(m, m - r - 1)$,

then the algorithm correctly decodes \mathbf{v} from \mathbf{w} .

What we have access to

Received word is $\mathbf{w} := \mathbf{v} + \text{err}_S$ for some $\mathbf{v} \in RM(m, m - 2r - 2)$ and $S = \{u_1, \dots, u_t\}$.

What we have access to

Received word is $\mathbf{w} := \mathbf{v} + \text{err}_S$ for some $\mathbf{v} \in RM(m, m - 2r - 2)$ and $S = \{u_1, \dots, u_t\}$.

Parity Check Matrix for $RM(m, m - 2r - 2)$ is the generator matrix for $RM(m, 2r + 1)$.

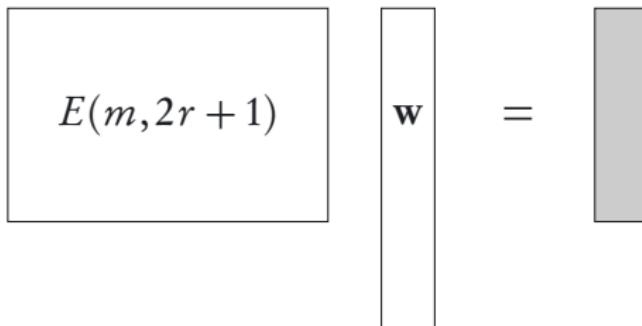
What we have access to

Received word is $\mathbf{w} := \mathbf{v} + \text{err}_S$ for some $\mathbf{v} \in RM(m, m - 2r - 2)$ and $S = \{u_1, \dots, u_t\}$.

$$\boxed{E(m, 2r + 1)} \quad \boxed{\mathbf{v}} = \mathbf{0}$$

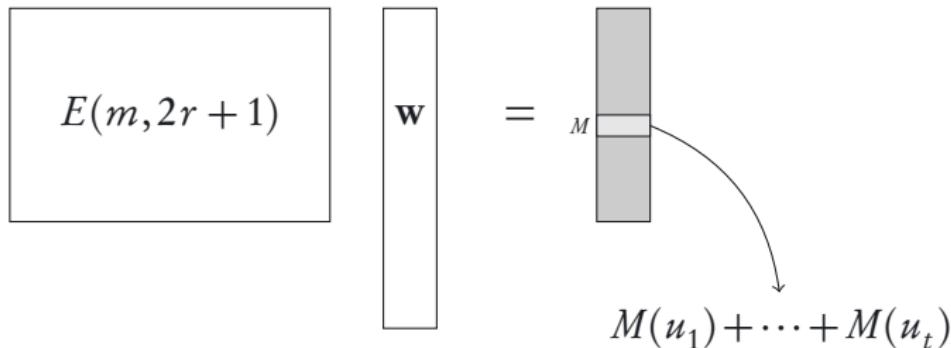
What we have access to

Received word is $\mathbf{w} := \mathbf{v} + \text{err}_S$ for some $\mathbf{v} \in RM(m, m - 2r - 2)$ and $S = \{u_1, \dots, u_t\}$.



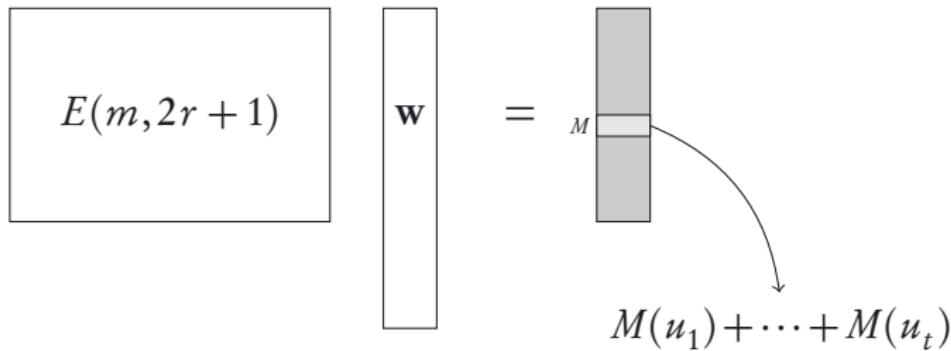
What we have access to

Received word is $\mathbf{w} := \mathbf{v} + \text{err}_S$ for some $\mathbf{v} \in RM(m, m - 2r - 2)$ and $S = \{u_1, \dots, u_t\}$.



What we have access to

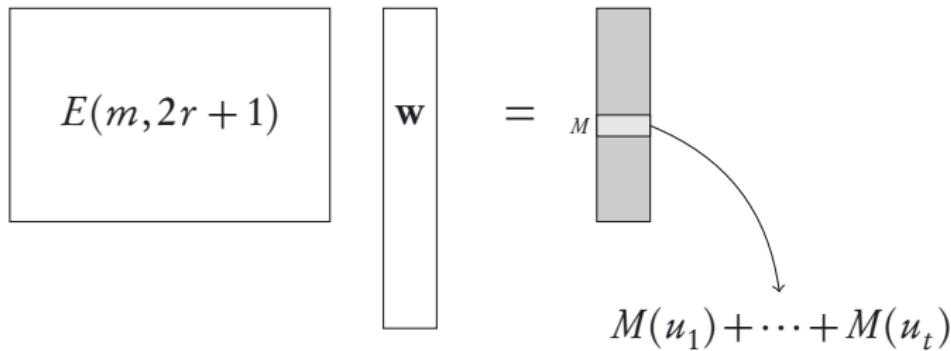
Received word is $\mathbf{w} := \mathbf{v} + \text{err}_S$ for some $\mathbf{v} \in RM(m, m - 2r - 2)$ and $S = \{u_1, \dots, u_t\}$.



Have access to $\sum_{i \in S} M(u_i)$ for every monomial M with $\deg(M) \leq 2r + 1$.

What we have access to

Received word is $\mathbf{w} := \mathbf{v} + \text{err}_S$ for some $\mathbf{v} \in RM(m, m - 2r - 2)$ and $S = \{u_1, \dots, u_t\}$.



Have access to $\sum_{i \in S} f(u_i)$ for every polynomial f with $\deg(f) \leq 2r + 1$.

Erasure Correctable Patterns

Erasure Correctable Patterns

A pattern of erasures is correctable *if and only if* the corresponding columns in the parity check matrix are linearly independent.

Erasure Correctable Patterns

A pattern of erasures is correctable *if and only if* the corresponding columns in the parity check matrix are linearly independent.

The **parity check matrix** for $RM(m, m - r - 1)$ is the **generator matrix** for $RM(m, r)$.

Erasure Correctable Patterns

A pattern of erasures is correctable *if and only if* the corresponding columns in the parity check matrix are linearly independent.

The **parity check matrix** for $RM(m, m - r - 1)$ is the **generator matrix** for $RM(m, r)$.

Corollary

A set of patterns $S = \{u_1, \dots, u_t\}$ is **erasure-correctable** in $RM(m, m - r - 1)$ *if and only if* $\{u_1^r, \dots, u_t^r\}$ are **linearly independent**.

u_i^r is just the vector of degree r monomials evaluated at u_i

The Decoding Algorithm

Input: Received word w ($= v + \text{err}_S$)

The Decoding Algorithm

Input: Received word w ($= v + \text{err}_S$)

Lemma

Assume that S is a *pattern of erasures correctable* in $RM(m, m - r - 1)$.

The Decoding Algorithm

Input: Received word \mathbf{w} ($= \mathbf{v} + \text{err}_S$)

Lemma

Assume that S is a *pattern of erasures correctable* in $RM(m, m - r - 1)$.

For any arbitrary $u \in \{0, 1\}^m$, we have $u \in S$ if and only if there exists a polynomial g with $\deg(g) \leq r$ such that

$$\sum_{i \in S} (f \cdot g)(u_i) = f(u) \text{ for every } f \text{ with } \deg(f) \leq r + 1.$$

The Decoding Algorithm

Input: Received word w ($= v + \text{err}_S$)

Lemma

Assume that S is a *pattern of erasures correctable* in $RM(m, m - r - 1)$.

For any arbitrary $u \in \{0, 1\}^m$, we have $u \in S$ if and only if there exists a polynomial g with $\deg(g) \leq r$ such that

$$\sum_{i \in S} (f \cdot g)(u_i) = f(u) \text{ for every } f \text{ with } \deg(f) \leq r + 1.$$

Can be checked by solving a system of linear equations.
Algorithm is straightforward.

Proof of Lemma

Claim 1

Let $u_1, \dots, u_t \in \{0, 1\}^m$ such that $\{u_1^r, \dots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- ▶ $\deg(h_i) \leq r$,
- ▶ $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Proof of Lemma

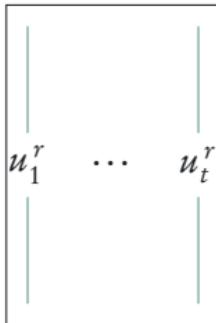
Claim 1

Let $u_1, \dots, u_t \in \{0, 1\}^m$ such that $\{u_1^r, \dots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- ▶ $\deg(h_i) \leq r$,
- ▶ $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Proof.



Proof of Lemma

Claim 1

Let $u_1, \dots, u_t \in \{0, 1\}^m$ such that $\{u_1^r, \dots, u_t^r\}$ are linearly independent.

Then, for each $i \in [t]$, there is a polynomial h_i such that:

- ▶ $\deg(h_i) \leq r$,
- ▶ $h_i(u_j) = 1$ if and only if $i = j$, and 0 otherwise.

Proof.

$$\begin{array}{|c|c|c|} \hline & & \\ \hline u_1^r & \dots & u_t^r \\ \hline & & \end{array}$$

Row operations
~~~~~→

$$\begin{array}{|c|} \hline I \\ \hline 0 \\ \hline \end{array}$$



# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $h_i$  such that:

- ▶  $\deg(h_i) \leq r$ ,
- ▶  $h_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $h_i$  such that:

- ▶  $\deg(h_i) \leq r$ ,
- ▶  $h_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

*Main Lemma ( $\Rightarrow$ ): If  $u \in S$ , then there is a polynomial  $g$  with  $\deg(g) \leq r$  such that*

$$\sum_{u_i \in S} (f \cdot g)(u_i) = f(u) \text{ for every } f \text{ with } \deg(f) \leq r + 1.$$

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $h_i$  such that:

- ▶  $\deg(h_i) \leq r$ ,
- ▶  $h_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

*Main Lemma ( $\Rightarrow$ ): If  $u \in S$ , then there is a polynomial  $g$  with  $\deg(g) \leq r$  such that*

$$\sum_{u_i \in S} (f \cdot g)(u_i) = f(u) \text{ for every } f \text{ with } \deg(f) \leq r + 1.$$

*If  $u = u_i$ , then  $g = h_i$  satisfies the conditions.*

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $h_i$  such that:

- ▶  $\deg(h_i) \leq r$ ,
- ▶  $h_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $b_i$  such that:

- ▶  $\deg(b_i) \leq r$ ,
- ▶  $b_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $b_i$  such that:

- ▶  $\deg(b_i) \leq r$ ,
- ▶  $b_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

## Claim 2

If  $u \notin \{u_1, \dots, u_t\}$ , then there is a polynomial  $f$  such that  $\deg(f) \leq r + 1$  and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \dots, t.$$

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $b_i$  such that:

- ▶  $\deg(b_i) \leq r$ ,
- ▶  $b_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

## Claim 2

If  $u \notin \{u_1, \dots, u_t\}$ , then there is a polynomial  $f$  such that  $\deg(f) \leq r + 1$  and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \dots, t.$$

*Main Lemma ( $\Leftarrow$ ): If  $u \notin S$ , then there is no polynomial  $g$  with  $\deg(g) \leq r$  such that*

$$\sum_{u_i \in S} (f \cdot g)(u_i) = f(u) \quad \text{for every } f \text{ with } \deg(f) \leq r + 1.$$

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $b_i$  such that:

- ▶  $\deg(b_i) \leq r$ ,
- ▶  $b_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

## Claim 2

If  $u \notin \{u_1, \dots, u_t\}$ , then there is a polynomial  $f$  such that  $\deg(f) \leq r + 1$  and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \dots, t.$$

## Proof

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $b_i$  such that:

- ▶  $\deg(b_i) \leq r$ ,
- ▶  $b_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

## Claim 2

If  $u \notin \{u_1, \dots, u_t\}$ , then there is a polynomial  $f$  such that  $\deg(f) \leq r + 1$  and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \dots, t.$$

## Proof

**Case 1:** Suppose  $b_i(u) = 1$  for some  $i$ .

| $u_1$ | $\dots$ | $u_i$           | $u_t$ | $u$ |
|-------|---------|-----------------|-------|-----|
| 0     | $\dots$ | 0 1 0 $\dots$ 0 |       | 1   |

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $b_i$  such that:

- ▶  $\deg(b_i) \leq r$ ,
- ▶  $b_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

## Claim 2

If  $u \notin \{u_1, \dots, u_t\}$ , then there is a polynomial  $f$  such that  $\deg(f) \leq r + 1$  and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \dots, t.$$

## Proof

**Case 1:** Suppose  $b_i(u) = 1$  for some  $i$ .

$u \neq u_i$  hence  $u_{(\ell)} \neq (u_i)_{(\ell)}$  for some coordinate  $\ell$ .

| $u_1$ | $\dots$ | $u_i$           | $u_t$ | $u$ |
|-------|---------|-----------------|-------|-----|
| 0     | $\dots$ | 0 1 0 $\dots$ 0 |       | 1   |

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $b_i$  such that:

- $\deg(b_i) \leq r$ ,
- $b_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

## Claim 2

If  $u \notin \{u_1, \dots, u_t\}$ , then there is a polynomial  $f$  such that  $\deg(f) \leq r + 1$  and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \dots, t.$$

## Proof

**Case 1:** Suppose  $b_i(u) = 1$  for some  $i$ .

$u \neq u_i$  hence  $u_{(\ell)} \neq (u_i)_{(\ell)}$  for some coordinate  $\ell$ .

$$f(\mathbf{x}) = b_i(\mathbf{x}) \cdot (x_\ell - (u_i)_{(\ell)}) \text{ works.}$$

|       |         |       |           |     |
|-------|---------|-------|-----------|-----|
| $u_1$ | $\dots$ | $u_i$ | $u_t$     | $u$ |
| 0     | $\dots$ | 0 1 0 | $\dots$ 0 | 1   |

|       |         |       |           |     |
|-------|---------|-------|-----------|-----|
| $u_1$ | $\dots$ | $u_i$ | $u_t$     | $u$ |
| 0     | $\dots$ | 0     | $\dots$ 0 | 1   |

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $b_i$  such that:

- ▶  $\deg(b_i) \leq r$ ,
- ▶  $b_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

## Claim 2

If  $u \notin \{u_1, \dots, u_t\}$ , then there is a polynomial  $f$  such that  $\deg(f) \leq r + 1$  and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \dots, t.$$

## Proof

**Case 2:** Suppose  $b_i(u) = 0$  for all  $i$ .

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $b_i$  such that:

- $\deg(b_i) \leq r$ ,
- $b_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

## Claim 2

If  $u \notin \{u_1, \dots, u_t\}$ , then there is a polynomial  $f$  such that  $\deg(f) \leq r + 1$  and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \dots, t.$$

## Proof

**Case 2:** Suppose  $b_i(u) = 0$  for all  $i$ .

Look at  $\sum b_i(x)$ .

$$\begin{array}{cccccc} & u_1 & & & u_t & \\ \hline & 1 & & \cdots & 1 & \\ & 0 & & & & \end{array}$$

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $b_i$  such that:

- $\deg(b_i) \leq r$ ,
- $b_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

## Claim 2

If  $u \notin \{u_1, \dots, u_t\}$ , then there is a polynomial  $f$  such that  $\deg(f) \leq r + 1$  and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \dots, t.$$

## Proof

**Case 2:** Suppose  $b_i(u) = 0$  for all  $i$ .

Look at  $\sum b_i(x)$ .

$$f(x) = 1 - \sum b_i(x) \text{ works.}$$

|       |         |       |     |
|-------|---------|-------|-----|
| $u_1$ | $\dots$ | $u_t$ | $u$ |
| 1     |         | 1     | 0   |

|       |         |       |     |
|-------|---------|-------|-----|
| $u_1$ | $\dots$ | $u_t$ | $u$ |
| 0     |         | 0     | 1   |

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $b_i$  such that:

- $\deg(b_i) \leq r$ ,
- $b_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

## Claim 2

If  $u \notin \{u_1, \dots, u_t\}$ , then there is a polynomial  $f$  such that  $\deg(f) \leq r + 1$  and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \dots, t.$$

## Proof

**Case 2:** Suppose  $b_i(u) = 0$  for all  $i$ .

Look at  $\sum b_i(x)$ .

$$f(x) = 1 - \sum b_i(x) \text{ works.}$$

|       |         |       |     |
|-------|---------|-------|-----|
| $u_1$ | $\dots$ | $u_t$ | $u$ |
| 1     |         | 1     | 0   |

|       |         |       |     |
|-------|---------|-------|-----|
| $u_1$ | $\dots$ | $u_t$ | $u$ |
| 0     |         | 0     | 1   |



# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $b_i$  such that:

- ▶  $\deg(b_i) \leq r$ ,
- ▶  $b_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

## Claim 2

If  $u \notin \{u_1, \dots, u_t\}$ , then there is a polynomial  $f$  such that  $\deg(f) \leq r + 1$  and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \dots, t.$$

# Proof of Lemma

## Claim 1

Let  $u_1, \dots, u_t \in \{0, 1\}^m$  such that  $\{u_1^r, \dots, u_t^r\}$  are linearly independent.

Then, for each  $i \in [t]$ , there is a polynomial  $b_i$  such that:

- ▶  $\deg(b_i) \leq r$ ,
- ▶  $b_i(u_j) = 1$  if and only if  $i = j$ , and 0 otherwise.

## Claim 2

If  $u \notin \{u_1, \dots, u_t\}$ , then there is a polynomial  $f$  such that  $\deg(f) \leq r + 1$  and

$$f(u) = 1 \quad , \quad \text{but } f(u_i) = 0 \text{ for all } i = 1, \dots, t.$$

Therefore, if  $\{u_1^r, \dots, u_t^r\}$  are linearly independent, then there exists a polynomial  $g$  with  $\deg(g) \leq r$  satisfying

$$\sum_{u_i \in S} (f \cdot g)(u_i) = f(u), \text{ for every polynomial } f \text{ with } \deg(f) \leq r + 1,$$

if and only if  $u = u_i$  for some  $i$ .

□

# Decoding Algorithm

- ▶ **Input:** Received word  $w$  ( $= v + \text{err}_S$ )

# Decoding Algorithm

- ▶ **Input:** Received word  $\mathbf{w}$  ( $= \mathbf{v} + \text{err}_S$ )
- ▶ Compute  $E(m, 2r + 1)\mathbf{w}$  to get the value of  $\sum_{u_i \in S} f(u_i)$  for every polynomial  $f$  with  $\deg(f) \leq 2r + 1$ .

# Decoding Algorithm

- ▶ **Input:** Received word  $\mathbf{w}$  ( $= \mathbf{v} + \text{err}_S$ )
- ▶ Compute  $E(m, 2r + 1)\mathbf{w}$  to get the value of  $\sum_{u_i \in S} f(u_i)$  for every polynomial  $f$  with  $\deg(f) \leq 2r + 1$ .
- ▶ For each  $u \in \{0, 1\}^m$ , solve for a polynomial  $g$  with  $\deg(g) \leq r$  satisfying

$$\sum_{u_i \in S} (f \cdot g)(u_i) = f(u), \text{ for every } f \text{ satisfying } \deg(f) \leq r + 1.$$

If there is a solution, then add  $u$  to Corruptions.

# Decoding Algorithm

- ▶ **Input:** Received word  $\mathbf{w}$  ( $= \mathbf{v} + \text{err}_S$ )
- ▶ Compute  $E(m, 2r + 1)\mathbf{w}$  to get the value of  $\sum_{u_i \in S} f(u_i)$  for every polynomial  $f$  with  $\deg(f) \leq 2r + 1$ .
- ▶ For each  $u \in \{0, 1\}^m$ , solve for a polynomial  $g$  with  $\deg(g) \leq r$  satisfying

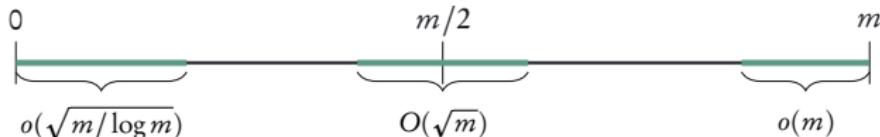
$$\sum_{u_i \in S} (f \cdot g)(u_i) = f(u), \text{ for every } f \text{ satisfying } \deg(f) \leq r + 1.$$

If there is a solution, then add  $u$  to Corruptions.

- ▶ Flip the coordinates in Corruptions and interpolate.

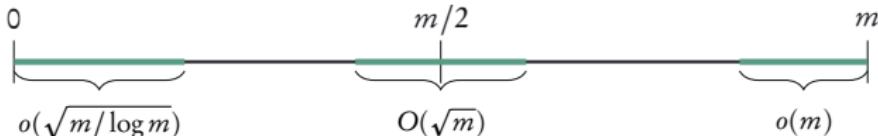
# ICYMI

**Remark:** If  $r$  falls in the **green zone**, then  $RM(m, m - r - 1)$  can correct  $\approx \binom{m}{\leq r}$  random erasures.



# ICYMI

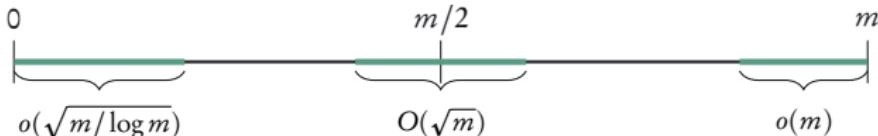
**Remark:** If  $r$  falls in the **green zone**, then  $RM(m, m - r - 1)$  can correct  $\approx \binom{m}{\leq r}$  random erasures.



**Theorem:** [S-Shpilka-Volk] Any pattern that is erasure-correctable in  $RM(m, m - r - 1)$  is **efficiently** error-correctable in  $RM(m, m - 2r - 2)$ .

# ICYMI

**Remark:** If  $r$  falls in the green zone, then  $RM(m, m - r - 1)$  can correct  $\approx \binom{m}{\leq r}$  random erasures.

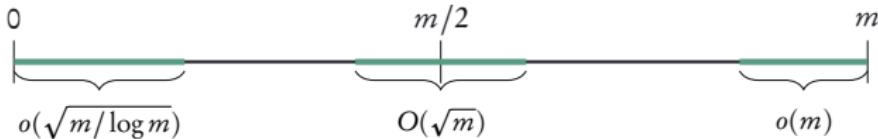


**Theorem:** [S-Shpilka-Volk] Any pattern that is erasure-correctable in  $RM(m, m - r - 1)$  is efficiently error-correctable in  $RM(m, m - 2r - 2)$ .

**Corollary #1:** (high-rate) Efficiently decodeable from  $(1 - o(1)) \binom{m}{\leq r}$  random errors in  $RM(m, m - 2r)$  if  $r = o(\sqrt{m / \log m})$ .  
(min distance of  $RM(m, m - 2r)$  is  $2^{2r}$ )

# ICYMI

**Remark:** If  $r$  falls in the green zone, then  $RM(m, m - r - 1)$  can correct  $\approx \binom{m}{\leq r}$  random erasures.

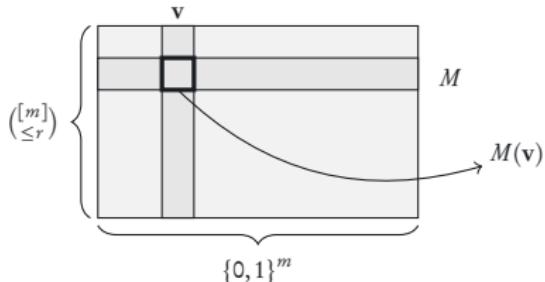


**Theorem:** [S-Shpilka-Volk] Any pattern that is erasure-correctable in  $RM(m, m - r - 1)$  is efficiently error-correctable in  $RM(m, m - 2r - 2)$ .

**Corollary #1:** (high-rate) Efficiently decodeable from  $(1 - o(1)) \binom{m}{\leq r}$  random errors in  $RM(m, m - 2r)$  if  $r = o(\sqrt{m / \log m})$ .  
(min distance of  $RM(m, m - 2r)$  is  $2^{2r}$ )

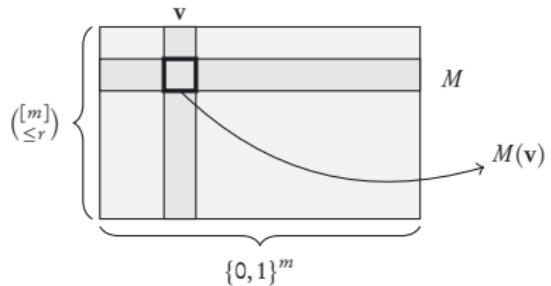
**Corollary #2:** (low-rate) Efficiently decodeable from  $(\frac{1}{2} - o(1)) 2^m$  random errors in  $RM(m, o(\sqrt{m}))$ .  
(min distance of  $RM(m, \sqrt{m})$  is  $2^{m - \sqrt{m}}$ )

# Robustness of evaluation matrix

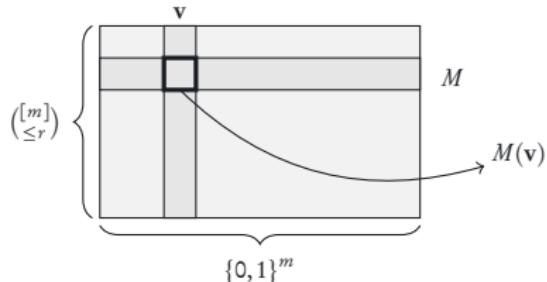


**Question:** Let  $R = \binom{m}{\leq r}$ . Suppose you pick  $(0.99)R$  columns at random. Are they linearly independent with high probability?

# Robustness of evaluation matrix

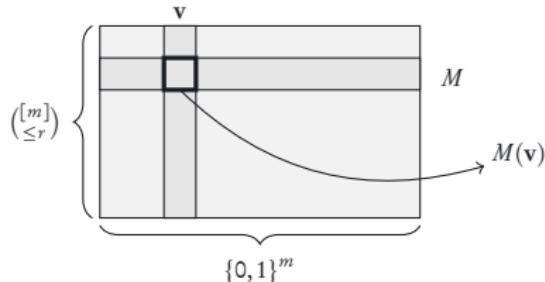


# Robustness of evaluation matrix



There are degree  $r$  polynomials that has only  $2^{m-r}$  non-zero evaluations (like  $x_1 \cdots x_r$ ). Suppose I tell you that  $f$  has only  $2^{m-r+t}$ , what do such  $f$  look like?

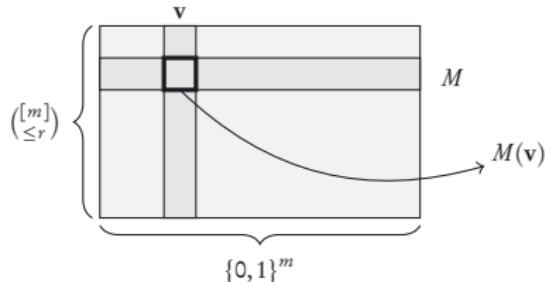
# Robustness of evaluation matrix



There are degree  $r$  polynomials that has only  $2^{m-r}$  non-zero evaluations (like  $x_1 \cdots x_r$ ). Suppose I tell you that  $f$  has only  $2^{m-r+t}$ , what do such  $f$  look like?

[Kaufman-Lovett-Porat]: Such functions *morally look like*  $\ell_1 \cdots \ell_t \cdot g$ .

# Robustness of evaluation matrix

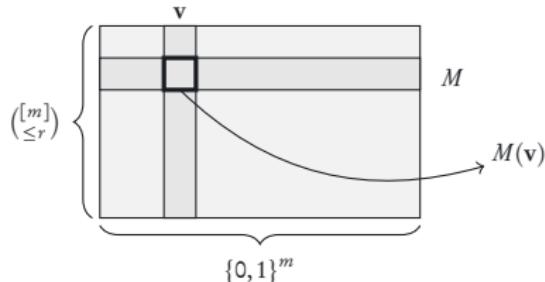


There are degree  $r$  polynomials that has only  $2^{m-r}$  non-zero evaluations (like  $x_1 \cdots x_r$ ). Suppose I tell you that  $f$  has only  $2^{m-r+t}$ , what do such  $f$  look like?

[Kaufman-Lovett-Porat]: Such functions *morally look like*  $\ell_1 \cdots \ell_t \cdot g$ .

Lets you *count* how many such functions there are — *weight distribution*.

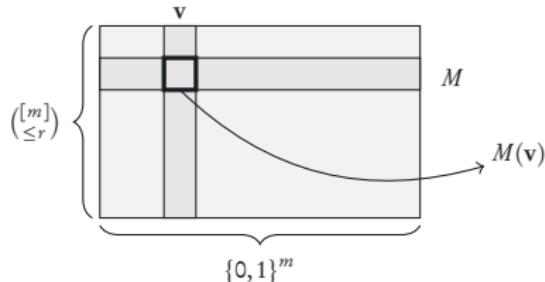
# Robustness of evaluation matrix



There are ways to pick many columns but have rank far from full.

For example, a sub-cube:  $\{*\cdots*000\ldots0\}$

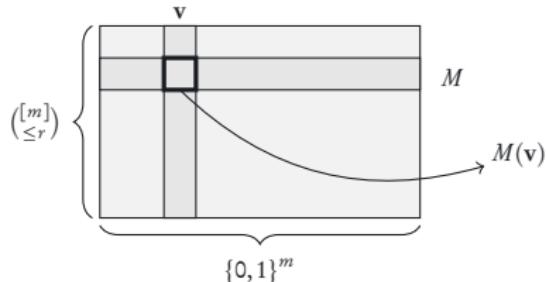
# Robustness of evaluation matrix



There are ways to pick  $2^{m-t}$  columns but have rank far from full.

For example, a sub-cube:  $\{*\dots*000\dots0\}$

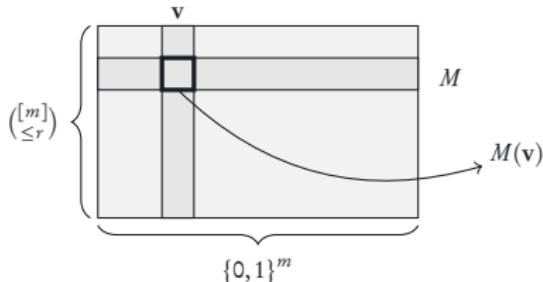
# Robustness of evaluation matrix



There are ways to pick  $2^{m-t}$  columns but have rank  $\binom{m-t}{\leq r}$ .

For example, a sub-cube:  $\{*\dots*000\dots0\}$

# Robustness of evaluation matrix

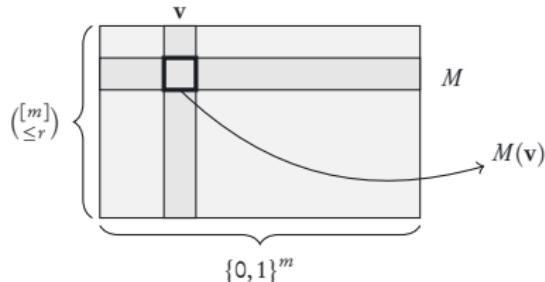


There are ways to pick  $2^{m-t}$  columns but have rank  $\binom{m-t}{\leq r}$ .

For example, a sub-cube:  $\{*\cdots*000\ldots0\}$

**Question:** If a large set of columns have very small rank, is it the case that it *morally looks like* a sub-cube?

# Robustness of evaluation matrix



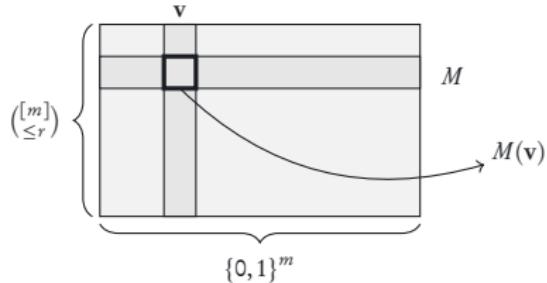
There are ways to pick  $2^{m-t}$  columns but have rank  $\binom{m-t}{\leq r}$ .

For example, a sub-cube:  $\{*\cdots*000\ldots0\}$

**Question:** If a large set of columns have very small rank, is it the case that it *morally looks like* a sub-cube?

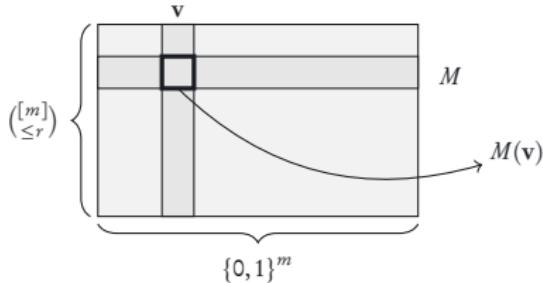
Hopefully this helps us count the number of sets of columns that yield a rank-deficiency.

# The obvious open question

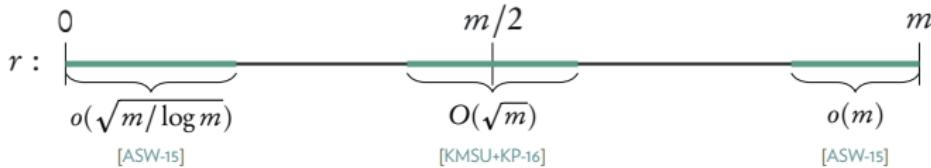


**Question:** Let  $R = \binom{m}{\leq r}$ . Suppose you pick  $(0.99)R$  columns at random. Are they linearly independent with high probability?

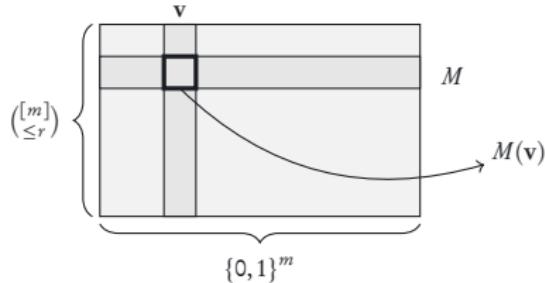
# The obvious open question



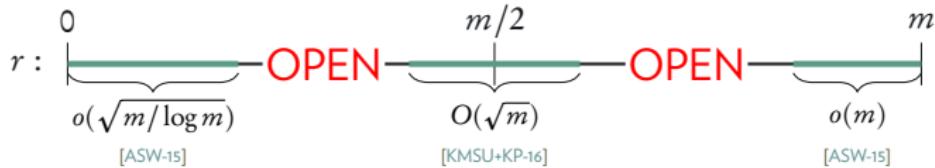
**Question:** Let  $R = \binom{m}{\leq r}$ . Suppose you pick  $(0.99)R$  columns at random. Are they linearly independent with high probability?



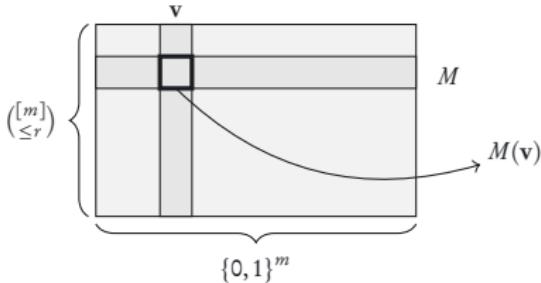
# The obvious open question



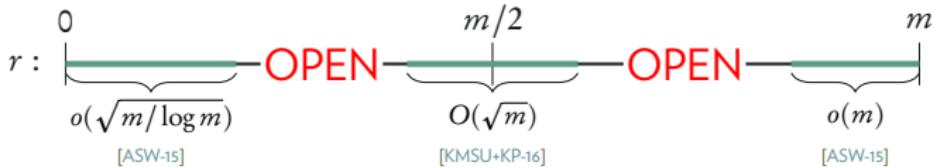
**Question:** Let  $R = \binom{m}{\leq r}$ . Suppose you pick  $(0.99)R$  columns at random. Are they linearly independent with high probability?



# The obvious open question



**Question:** Let  $R = \binom{m}{\leq r}$ . Suppose you pick  $(0.99)R$  columns at random. Are they linearly independent with high probability?



\end{document}