Learning to Understand: Identifying Interactions via the Mobius Transform

Justin S. Kang[1], Yigit E. Erginbas[1], Landon Butler[1], Prof. Ramtin Pedarsani[2], Prof. Kannan Ramchandran[1]





[1]UC Berkeley

[2]UC Santa Barbara

Algorithmic Structures for Uncoordinated Communications and Statistical Inference in Exceedingly Large Spaces

Motivation: Sentiment Analysis

Review

The concept of information entropy was introduced by Claude Shannon in his 1948 paper "A Mathematical Theory of Communication", and is also referred to as Shannon entropy. Shannon's theory defines a data communication system composed of three elements: a source of data, a communication channel, and a receiver. The "fundamental problem of communication" – as expressed by Shannon – is for the receiver to be able to identify what data was generated by the source, based on the signal it receives through the channel …

Sentiment score

5/10

Motivation: Sentiment Analysis

Review

The concept of information entropy was introduced by Claude Shannon in his 1948 paper "A Mathematical Theory of Communication", and is also referred to as Shannon entropy. Shannon's theory defines a data communication system composed of three elements: a source of data, a communication channel, and a receiver. The "fundamental problem of communication" – as expressed by Shannon – is for the receiver to be able to identify what data was generated by the source, based on the signal it receives through the channel …

→ →

Sentiment score

5/10

Can we understand what part of the text triggers the model to produce the (erroneous) output?

Typical Solution: Mask and Try Again

The concept of information [redacted] by Claude Shannon in his 1948 paper "A Mathematical Theory of Communication" and is also referred to as Shannon entropy. [redacted] communication [redacted] source of data, a communication channel, and a receiver. The "fundamental problem of communication" – as expressed by Shannon – is for the receiver to be able to identify what data was [redacted] …

[redacted] was introduced by Claude Shannon in his 1948 paper "A Mathematical Theory of Communication"[redacted]of three elements: a source of data, a communication channel, and a receiver. The [redacted] of communication" – as expressed by Shannon – is for the receiver to be able to identify what data was generated by the source …

The concept [redacted]"A Mathematical Theory of Communication" and is also referred to as Shannon entropy. S[redacted] data communication system composed of three elements: a source of data, a communication channel, and a receiver. The "fundamental problem of [redacted]hannon – is for the receiver to be able to identify what data was generated by the source …

7/10

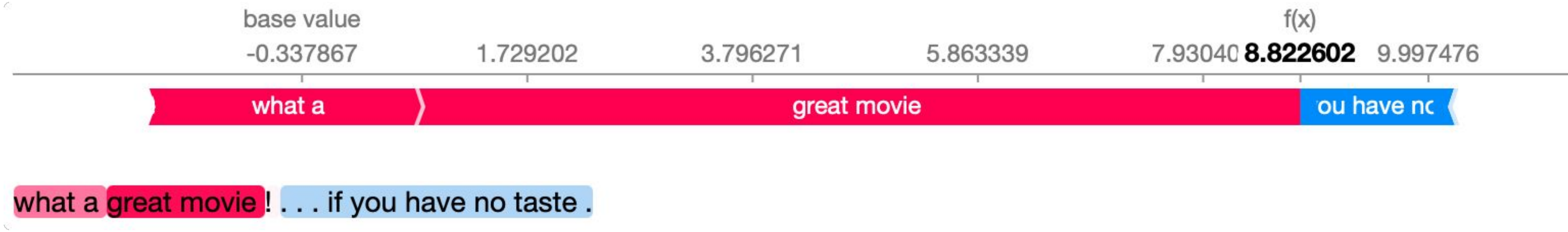
5/10

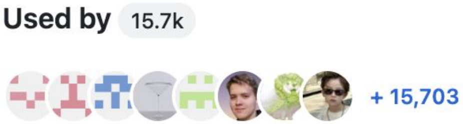
10/10

Shapley Value and SHAP

- SHAP software package: game-theoretic ***Shapley Value***
- Assigns a score to each (group of) word related to its average marginal contribution to the overall score

SHAP

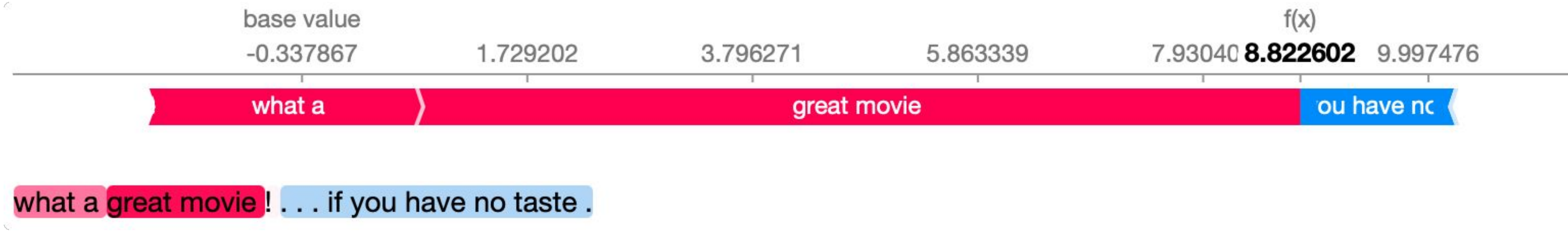




Shapley Value and SHAP

- SHAP software package: game-theoretic ***Shapley Value***
- Assigns a score to each (group of) word related to its average marginal contribution to the overall score

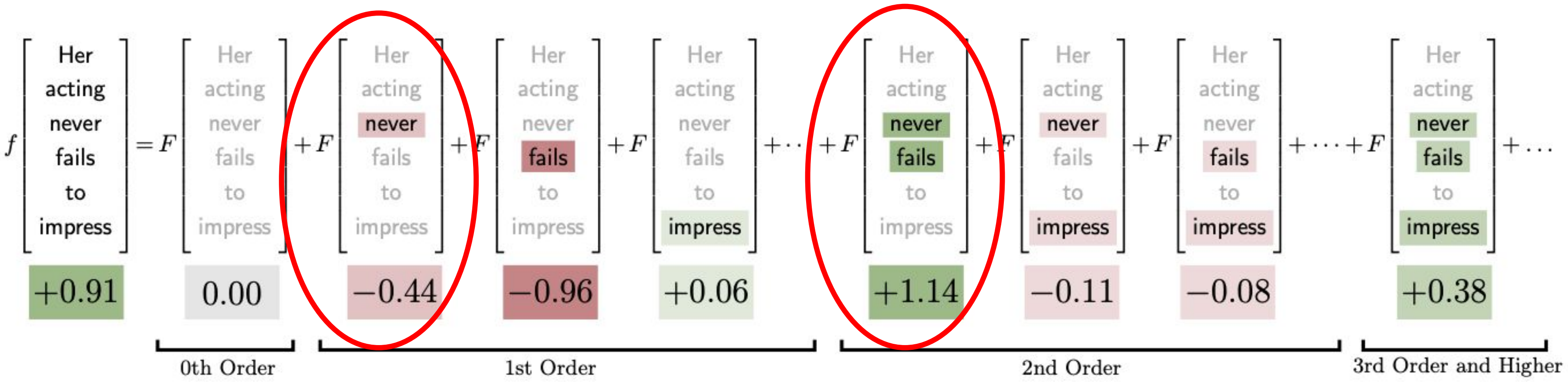
SHAP



Improving on SHAP:

1. Faster - Fewer masking patterns
2. Higher order information

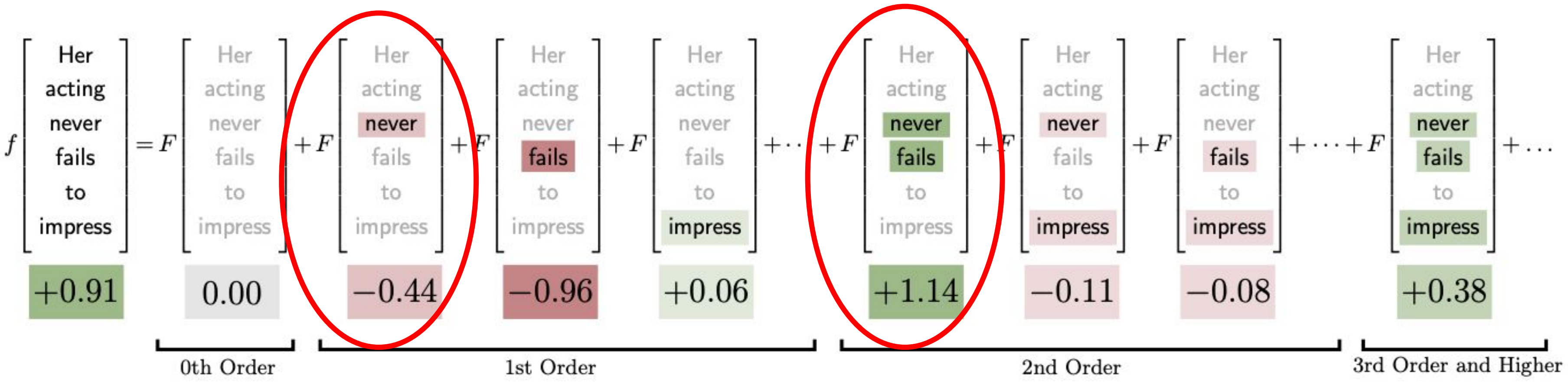
Higher order information is useful



- First order information is deceptive: “**never**” is negative on its own.
- If “**never**” appears before “**fails**” connotation is positive.

Sentiment analyzer uses pretrained BERT fine-tuned on IMDB review dataset

Higher order information is useful



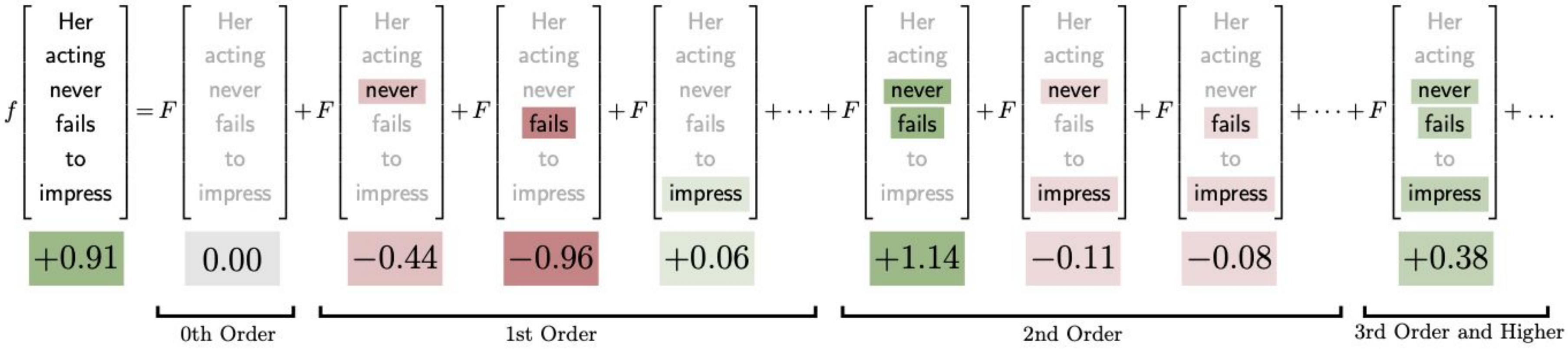
- First order information is deceptive: “**never**” is negative on its own.
- If “**never**” appears before “**fails**” connotation is positive.

Decomposing a function into constituent parts - just like a **Fourier Transform**

Signal Processing Approach for Explanations!

- Decompose the function in terms of effects of sets of inputs: polynomial

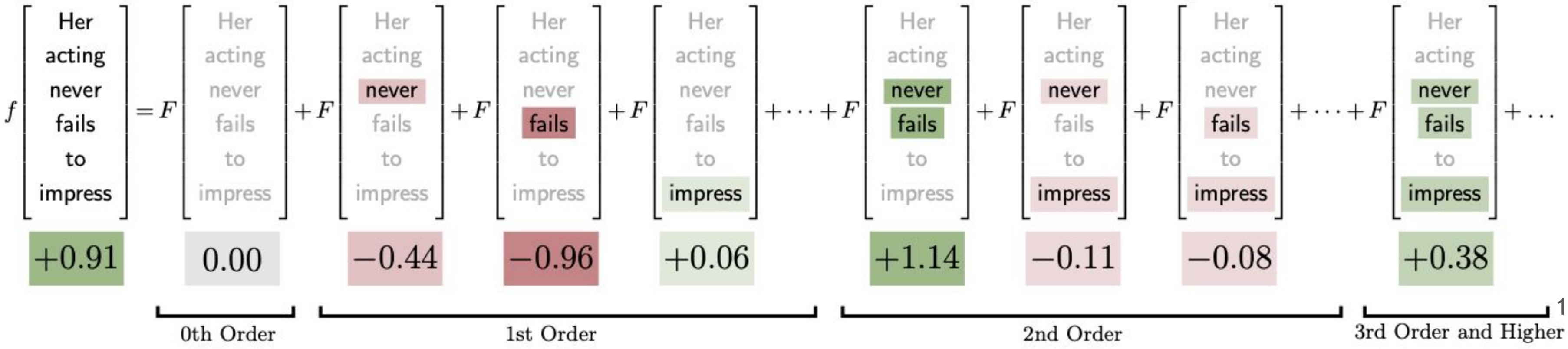
$$f(\mathbf{m}) =$$



Signal Processing Approach for Explanations!

- Decompose the function in terms of effects of sets of inputs: polynomial

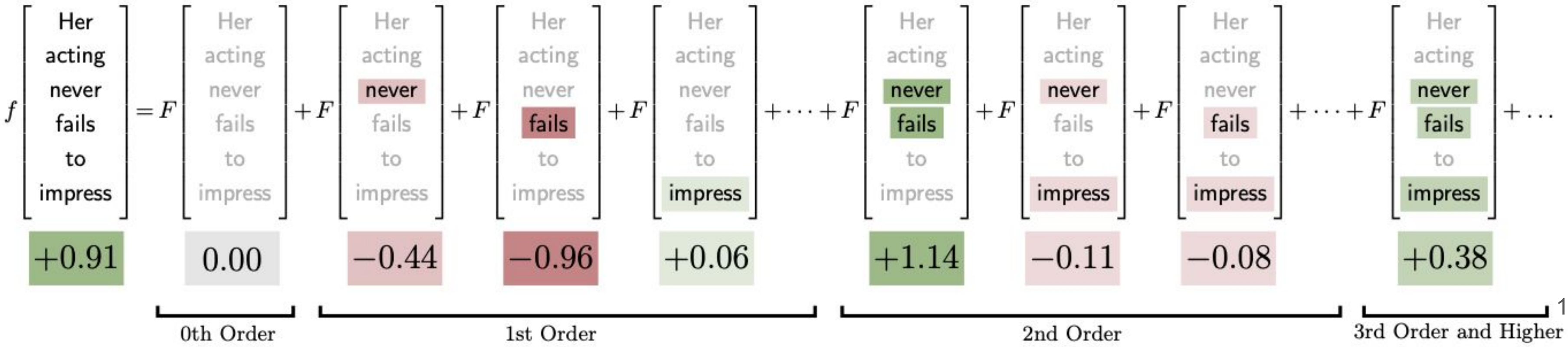
$$f(\mathbf{m}) = -0.44 m_3$$



Signal Processing Approach for Explanations!

- Decompose the function in terms of effects of sets of inputs: polynomial

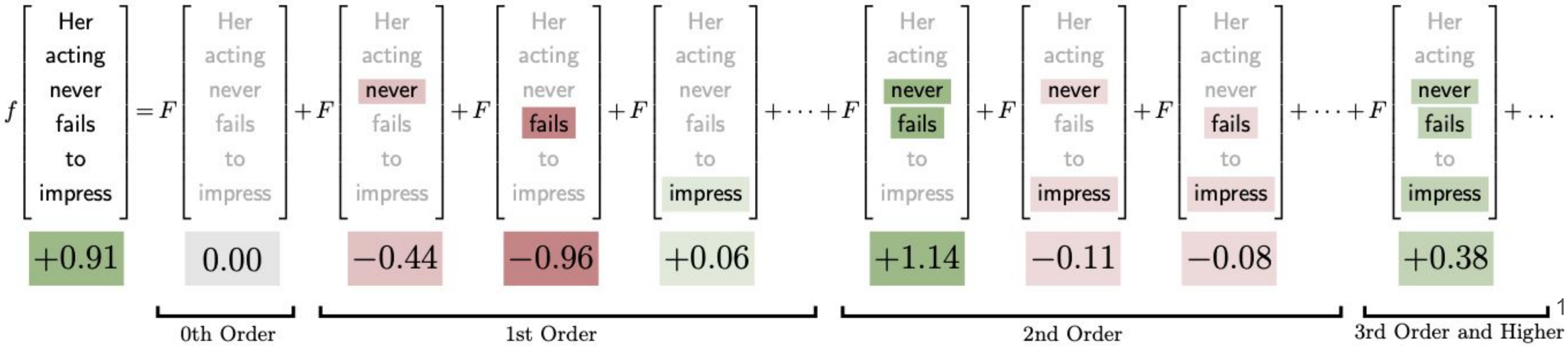
$$f(\mathbf{m}) = -0.44 m_3 + -0.96 m_4$$



Signal Processing Approach for Explanations!

- Decompose the function in terms of effects of sets of inputs: polynomial

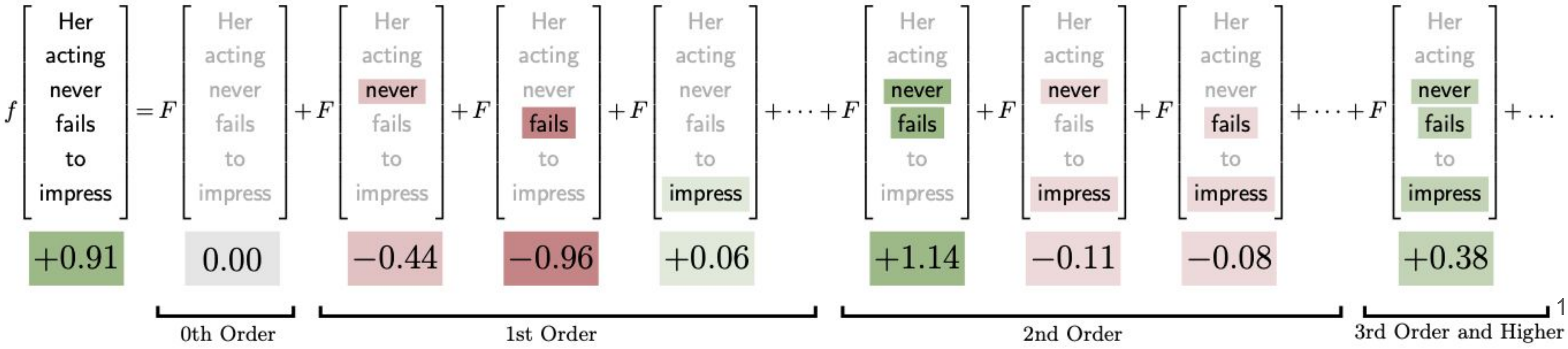
$$f(\mathbf{m}) = -0.44m_3 + -0.96m_4 + 0.06m_6 + \cdots + 1.14m_3m_4 + \\ -0.11m_3m_6 + -0.18m_4m_6 + 0.38m_3m_4m_6 + \ldots$$



Signal Processing Approach for Explanations!

- Decompose the function in terms of effects of sets of inputs: polynomial

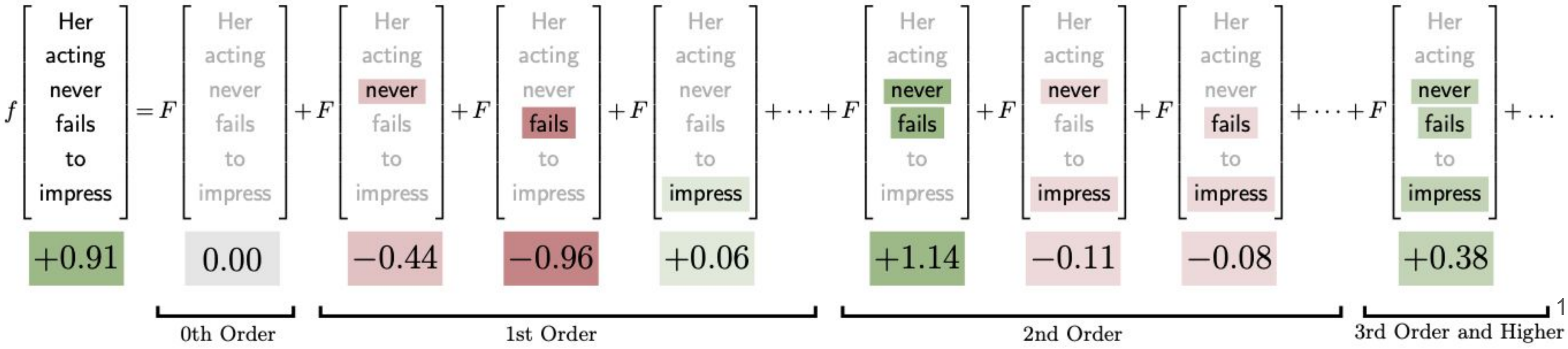
$$f(\mathbf{m}) = -0.44(1) + -0.96(1) + 0.06(1) + \cdots + 1.14(1)(1) + \\ -0.11(1)(1) + -0.18(1)(1) + 0.38(1)(1)(1) + \ldots$$



Signal Processing Approach for Explanations!

- Decompose the function in terms of effects of sets of inputs: polynomial

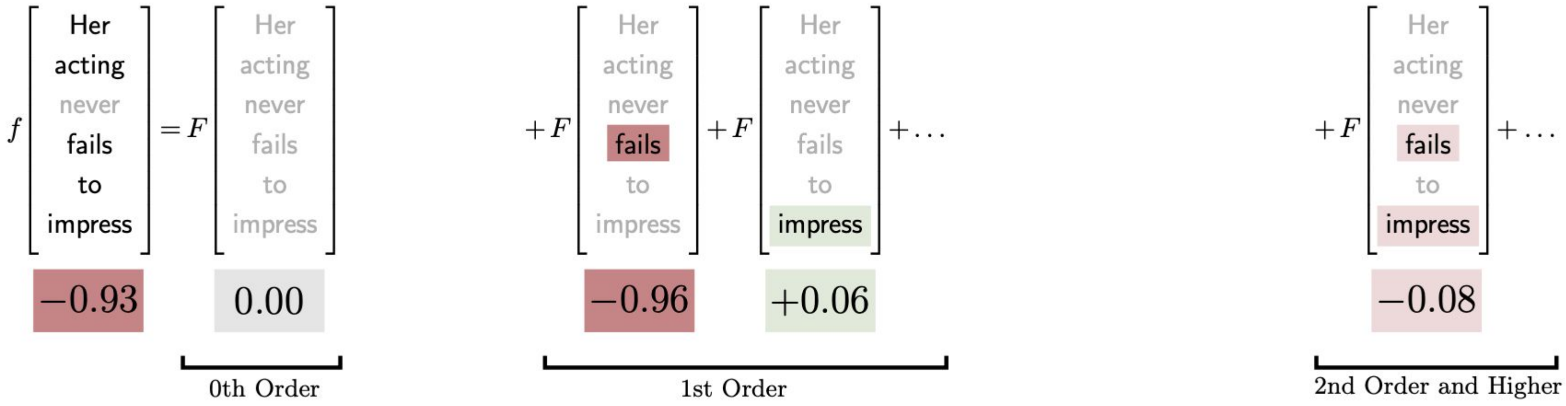
$$f(\mathbf{m}) = 0.91$$



Signal Processing Approach for Explanations!

- Decompose the function in terms of effects of sets of inputs: polynomial

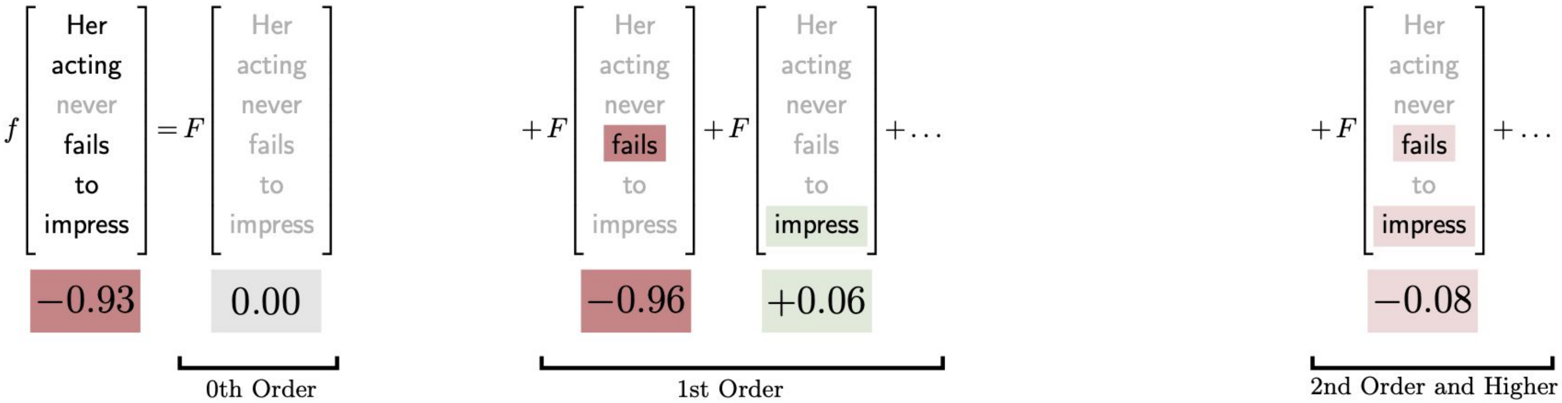
$$f(\mathbf{m}) = -0.44(0) + -0.96(1) + 0.06(1) + \cdots + 1.14(0)(1) + \\ -0.11(0)(1) + -0.18(1)(1) + 0.38(0)(1)(1) + \ldots$$



Signal Processing Approach for Explanations!

- Decompose the function in terms of effects of sets of inputs: polynomial

$$f(\mathbf{m}) = -0.93$$



The Mobius Transform (AND basis)

- The Mobius transform is defined as:

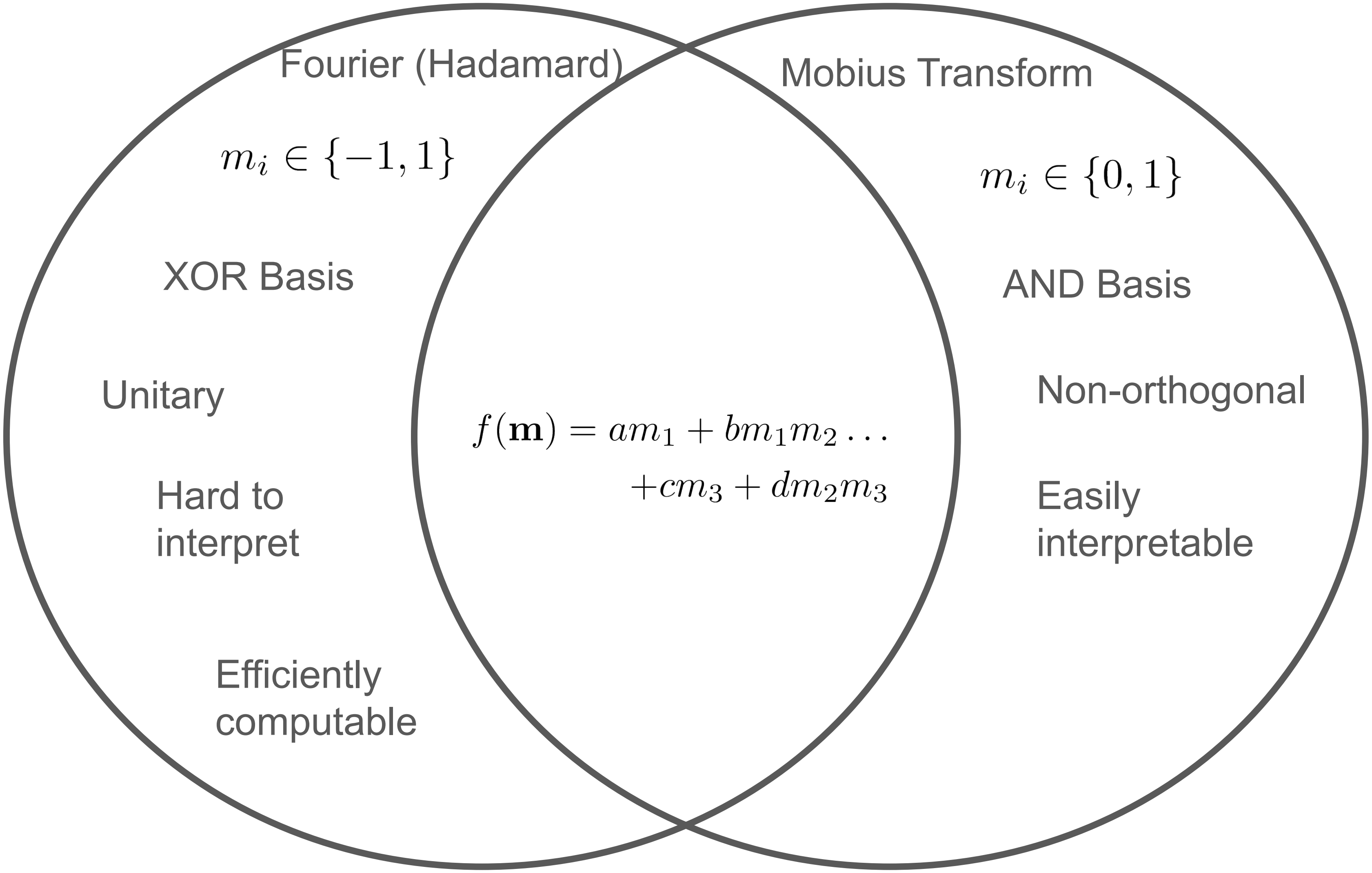
$$F(\mathbf{k}) = \sum_{\mathbf{m} \leq \mathbf{k}} (-1)^{\mathbf{1}^{\mathrm{T}}(\mathbf{k}-\mathbf{m})} f(\mathbf{m})$$

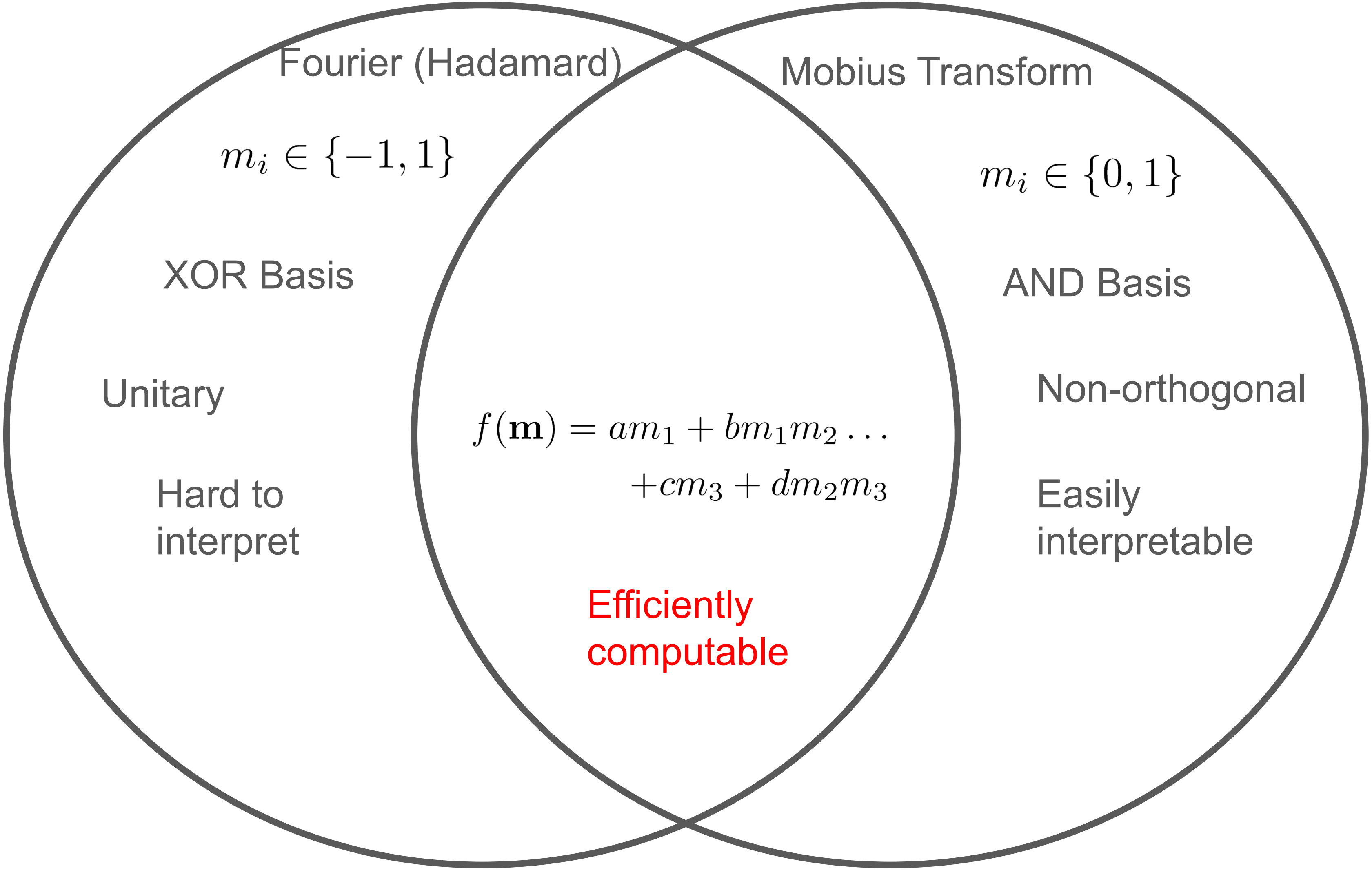
- The “Backwards” transform is:

$$f(\mathbf{m}) = \sum_{\mathbf{k} \leq \mathbf{m}} F(\mathbf{k})$$

August Möbius

Gian-Carlo Rota


Fourier (Hadamard)
$m_i \in \{-1, 1\}$
XOR Basis
Unitary
Hard to interpret
Efficiently computable
Mobius Transform
$m_i \in \{0, 1\}$
AND Basis
Non-orthogonal
Easily interpretable
$f(\mathbf{m}) = am_1 + bm_1m_2 \dots + cm_3 + dm_2m_3$


Fourier (Hadamard)
$m_i \in \{-1, 1\}$
XOR Basis
Unitary
Hard to interpret
Mobius Transform
$m_i \in \{0, 1\}$
AND Basis
Non-orthogonal
Easily interpretable
$f(\mathbf{m}) = am_1 + bm_1m_2 \ldots + cm_3 + dm_2m_3$
Efficiently computable

Long History of *Fast* Fourier Transforms

The Fast Fourier Transform

(1805) - Gauss uses the FFT to compute the orbits of astronomical objects

(1930) - Frank Yates develops the Fast Hadamard Transform

(1965) - Cooley and Tukey re-discover the FFT

The Sparse Fourier Transform

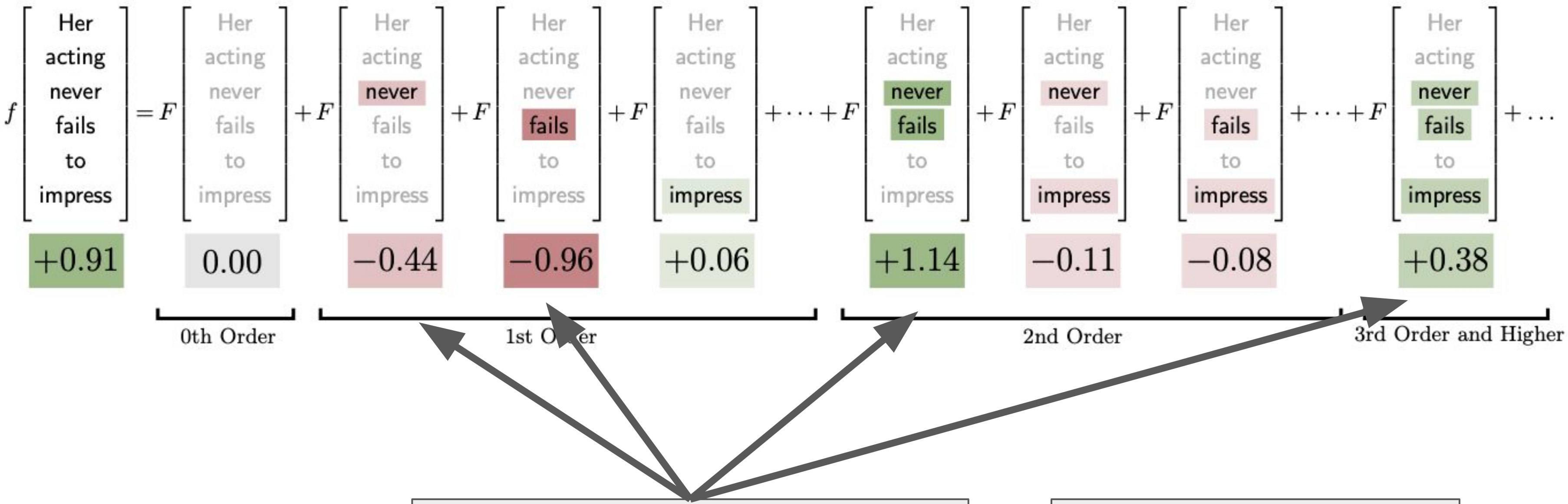
(2012-Present) - Many algorithms, including:

Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price - Sparse DFT

Sameer Pawar, Kannan Ramchandran - FFAST

Many others!

“Signal” Model - Structure of Explainable Representations



1. Only a small number of coefficients are large

2. Big terms are generally low-order

Theorems

Theorem 1. (Noiseless Decoding) When there are K non-zero interactions chosen uniformly at random from all 2^n interaction, with $K = O(2^{n\delta})$ for $\delta < 1/3$ our algorithm exactly computes the Mobius transform:

- with sample complexity $O(Kn)$ and
- with time $O(Kn^2)$

with probability $1 - O(1/K)$.

Theorem 2. (Robust Low-Decoding, Informal) When there are K non-zero interactions chosen uniformly over all $|\mathbf{k}| \leq t$, with $t = \Theta(n^\alpha)$, $\alpha \leq 0.407$ our algorithm computes the Mobius transform:

- with sample complexity $O(Kt\log(n))$ and
- with time $O(K\text{poly}(n))$

with probability $1 - O(1/K)$ with any fixed SNR.

Theorems

Theorem 1. (Uniform Prior, Noiseless) When there are K non-zero interactions chosen uniformly at random from all 2^n interaction, with $K = O(2^{n\delta})$ for $\delta < 1/3$ our algorithm exactly computes the Mobius transform:

- with sample complexity $O(Kn)$ and
- with time $O(Kn^2)$

with probability $1 - O(1/K)$.

Theorem 2. (t-Degree, Robust) When there are K non-zero interactions chosen uniformly over all $|\mathbf{k}| \leq t$, with $t = \Theta(n^{\alpha})$, $\alpha \leq 0.407$ our algorithm computes the Mobius transform:

- with sample complexity $O(Kt\log(n))$ and
- with time $O(K\text{poly}(n))$

with probability $1 - O(1/K)$ with any fixed SNR.

Example: $K = 1000$ interactions, $n = 100$ words, $t = 4$ degree.

$$\text{Naive} : 2^n \approx 10^{30}$$

$$\text{Theorem 1} : Kn = 10^5$$

$$\text{Theorem 2} : Kt\log(n) \approx 2.6 \times 10^4$$

Step 1 - Subsampling for Optimal Aliasing/Hashing

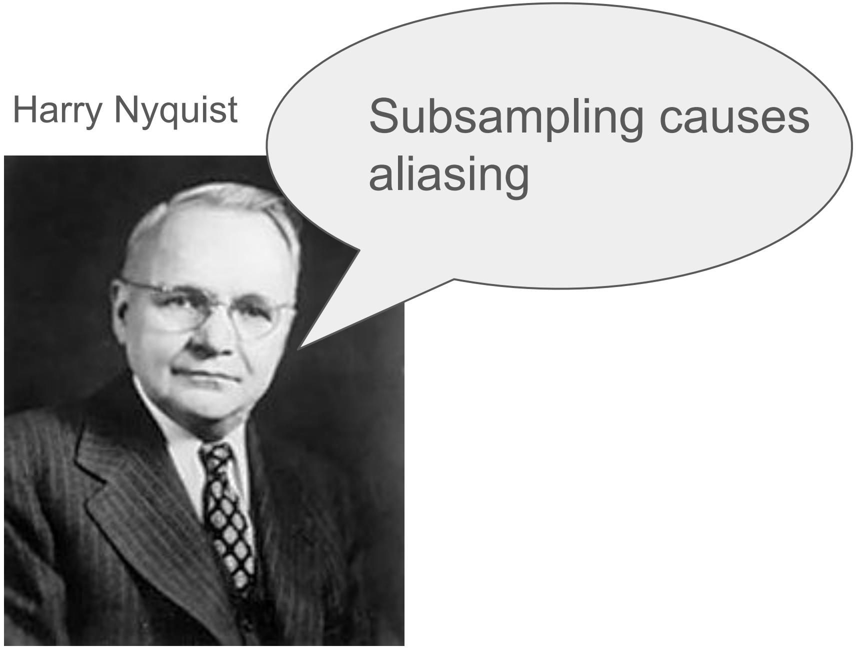
- **Inescapable fact** of signal processing (Nyquist Sampling Theorem):

Harry Nyquist

Step 1 - Subsampling for Optimal Aliasing/Hashing

- **Inescapable fact** of signal processing (Nyquist Sampling Theorem):

Harry Nyquist



Step 1 - Subsampling for Optimal Aliasing/Hashing

- **Inescapable fact** of signal processing (Nyquist Sampling Theorem):

Harry Nyquist

Subsampling causes aliasing

Embrace and understand the aliasing!

Lemma 1. *Consider* $\mathbf{H} \in \mathbf{Z}_2^{b \times n}$, $b < n$ *and* $f : \mathbb{Z}_2^n \mapsto \mathbb{R}$*. Let*

$$u(\boldsymbol{\ell}) = f\left(\overline{\mathbf{H}^T \overline{\boldsymbol{\ell}}}\right), \ \forall \boldsymbol{\ell} \in \mathbb{Z}_2^b.$$

If U is the Mobius transform of u, and F is the Mobius transform of f we have:

$$U(\mathbf{j}) = \sum_{\mathbf{H}\mathbf{k}=\mathbf{j}} F(\mathbf{k}).$$

Step 1 - Subsampling for Optimal Aliasing/Hashing

- **Inescapable fact** of signal processing (Nyquist Sampling Theorem):

Harry Nyquist

Subsampling causes aliasing

Embrace and understand the aliasing!

$$U(\mathbf{j}) = \sum_{\mathbf{Hk}=\mathbf{j}} F(\mathbf{k})$$

Step 1 - Subsampling for Optimal Aliasing/Hashing

- **Inescapable fact** of signal processing (Nyquist Sampling Theorem):

Harry Nyquist

Subsampling causes aliasing

Embrace and understand the aliasing!

$$U(\mathbf{j}) = \sum_{\mathbf{Hk}=\mathbf{j}} F(\mathbf{k})$$

monoid

Step 1 - Subsampling for Optimal Aliasing/Hashing

Non-zero Interactions

$$\begin{bmatrix} \text{Her} \\ \text{acting} \\ \text{never} \\ \text{fails} \\ \text{to} \\ \text{impress} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \mathbf{k}_1 \qquad \begin{bmatrix} \text{Her} \\ \text{acting} \\ \text{never} \\ \text{fails} \\ \text{to} \\ \text{impress} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{k}_2$$

$$\begin{bmatrix} \text{Her} \\ \text{acting} \\ \text{never} \\ \text{fails} \\ \text{to} \\ \text{impress} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{k}_3 \qquad \begin{bmatrix} \text{Her} \\ \text{acting} \\ \text{never} \\ \text{fails} \\ \text{to} \\ \text{impress} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \mathbf{k}_4$$

Transform

Aliasing 1

$$\begin{aligned} u_1(00) &= f(110011) \\ u_1(01) &= f(110111) \\ u_1(10) &= f(111011) \\ u_1(11) &= f(111111) \end{aligned} \longrightarrow \begin{aligned} U_1(00) &= 0 \\ U_1(01) &= F(\mathbf{k}_3) \\ U_1(10) &= F(\mathbf{k}_1) \\ U_1(11) &= F(\mathbf{k}_2) + F(\mathbf{k}_4) \end{aligned}$$

Aliasing 2

$$\begin{aligned} u_2(00) &= f(111100) \\ u_2(01) &= f(111101) \\ u_2(10) &= f(111110) \\ u_2(11) &= f(111111) \end{aligned} \longrightarrow \begin{aligned} U_2(00) &= F(\mathbf{k}_1) + F(\mathbf{k}_2) + F(\mathbf{k}_3) \\ U_2(01) &= F(\mathbf{k}_4) \\ U_2(10) &= 0 \\ U_2(11) &= 0 \end{aligned}$$

Zeroton

Singleton

Multiton

Step 1 - Subsampling for Optimal Aliasing/Hashing

Non-zero Interactions

$$\begin{bmatrix}\text{Her}\\\text{acting}\\\text{never}\\\text{fails}\\\text{to}\\\text{impress}\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix} = \mathbf{k}_1 \qquad \begin{bmatrix}\text{Her}\\\text{acting}\\\text{never}\\\text{fails}\\\text{to}\\\text{impress}\end{bmatrix} = \begin{bmatrix}0\\0\\1\\1\\0\\0\end{bmatrix} = \mathbf{k}_2$$

$$\begin{bmatrix}\text{Her}\\\text{acting}\\\text{never}\\\text{fails}\\\text{to}\\\text{impress}\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\\0\\0\end{bmatrix} = \mathbf{k}_3 \qquad \begin{bmatrix}\text{Her}\\\text{acting}\\\text{never}\\\text{fails}\\\text{to}\\\text{impress}\end{bmatrix} = \begin{bmatrix}0\\0\\1\\1\\0\\1\end{bmatrix} = \mathbf{k}_4$$

Transform

Aliasing 1

$$\begin{aligned} u_1(00) &= f(110011) \\ u_1(01) &= f(110111) \\ u_1(10) &= f(111011) \\ u_1(11) &= f(111111) \end{aligned} \longrightarrow \begin{aligned} U_1(00) &= 0 \\ U_1(01) &= F(\mathbf{k}_3) \\ U_1(10) &= F(\mathbf{k}_1) \\ U_1(11) &= F(\mathbf{k}_2) + F(\mathbf{k}_4) \end{aligned}$$

Aliasing 2

$$\begin{aligned} u_2(00) &= f(111100) \\ u_2(01) &= f(111101) \\ u_2(10) &= f(111110) \\ u_2(11) &= f(111111) \end{aligned} \longrightarrow \begin{aligned} U_2(00) &= F(\mathbf{k}_1) + F(\mathbf{k}_2) + F(\mathbf{k}_3) \\ U_2(01) &= F(\mathbf{k}_4) \\ U_2(10) &= 0 \\ U_2(11) &= 0 \end{aligned}$$

Zeroton

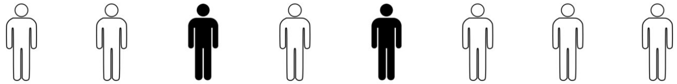
Singleton

Multiton

Q: How do we ensure good hashing? (Design H)

Step 2: Group Testing

- Originally proposed by Dorfman (1940s)
- Finds efficient ways to test soldiers for syphilis
- Pooling test allows you to identify infected individuals with fewer tests



								Outcome
1	1	1	1	0	0	0	0	Positive
0	0	0	0	1	1	1	1	Positive
1	1	0	0	0	0	0	0	Negative
0	0	1	0	0	0	0	0	Positive
0	0	1	0	1	1	0	0	Positive
0	0	0	0	1	0	0	0	Positive

t infected individuals

n total individuals

Step 2: Group Testing

- Originally proposed by Dorfman (1940s)
- Finds efficient ways to test soldiers for syphilis
- Pooling test allows you to identify infected individuals with fewer tests

$\mathbf{k}_1 =$	Her	acting	never	fails	to	impress	$\mathbf{j}$
	0	0	0	1	1	1	0
$\mathbf{H} =$	0	1	1	0	0	1	1
	1	0	1	0	1	0	1

t important words

n total words

Step 2: Group Testing

- Originally proposed by Dorfman (1940s)
- Finds efficient ways to test soldiers for syphilis
- Pooling test allows you to identify infected individuals with fewer tests

$\mathbf{k}_1 =$	Her	acting	never	fails	to	impress	$\mathbf{j}$
	0	0	0	1	1	1	0
$\mathbf{H} =$	0	1	1	0	0	1	1
	1	0	1	0	1	0	1

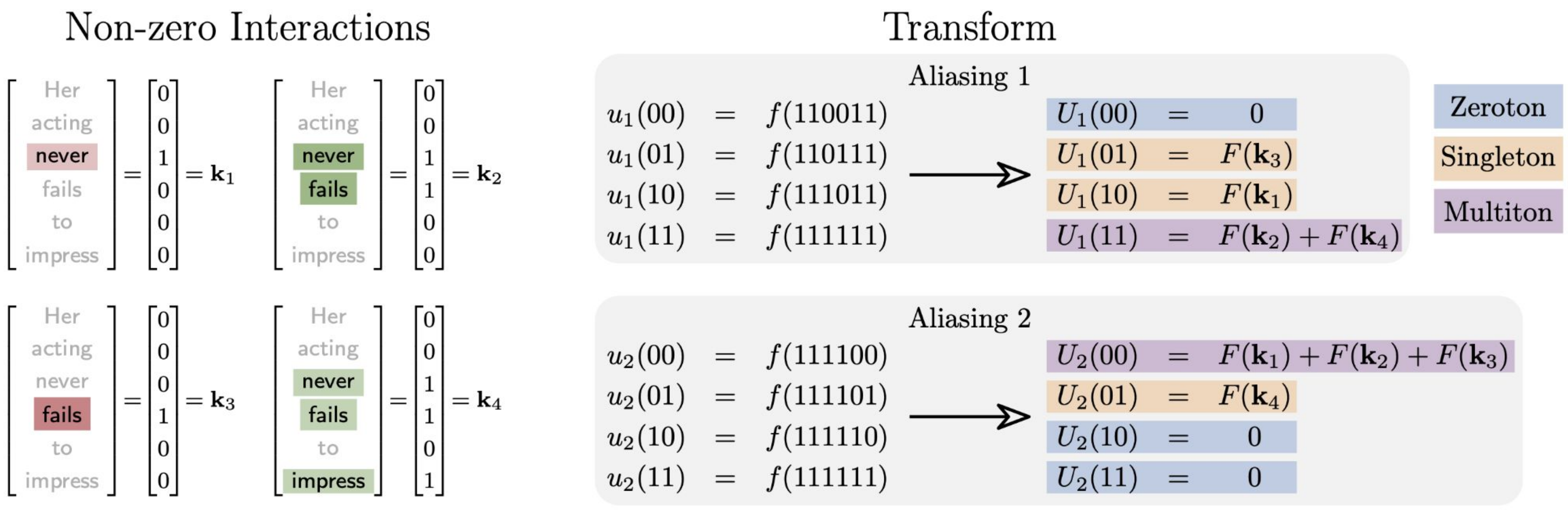
t important words

n total words

Group testing has **the same arithmetic** as our hashing rule!

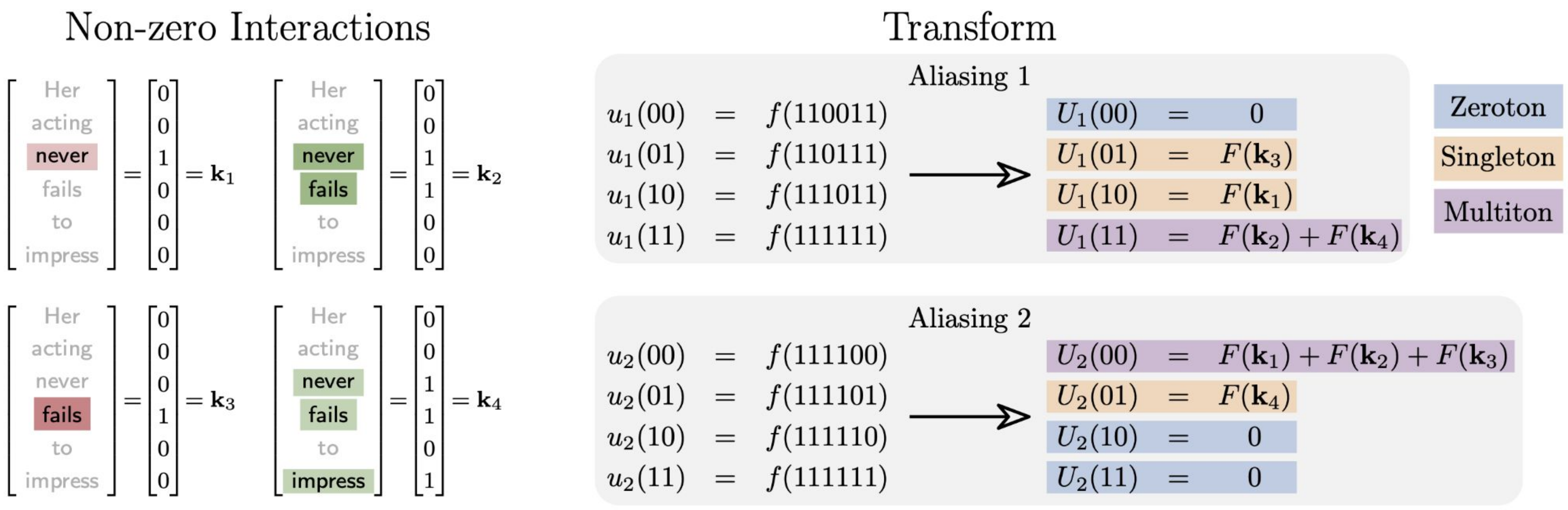
$$U(\mathbf{j}) = \sum_{\mathbf{Hk}=\mathbf{j}} F(\mathbf{k})$$

Step 2: Group Testing



Q: How do we ensure good hashing?
A: Exploit group testing

Step 2: Group Testing



Q: How do we identify singleton/multitons, and find the k_i?

Step 2: Group Testing

Non-zero Interactions

$$\begin{bmatrix}\text{Her}\\ \text{acting}\\ \text{never}\\ \text{fails}\\ \text{to}\\ \text{impress}\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix} = \mathbf{k}_1 \qquad \begin{bmatrix}\text{Her}\\ \text{acting}\\ \text{never}\\ \text{fails}\\ \text{to}\\ \text{impress}\end{bmatrix} = \begin{bmatrix}0\\0\\1\\1\\0\\0\end{bmatrix} = \mathbf{k}_2$$

$$\begin{bmatrix}\text{Her}\\ \text{acting}\\ \text{never}\\ \text{fails}\\ \text{to}\\ \text{impress}\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\\0\\0\end{bmatrix} = \mathbf{k}_3 \qquad \begin{bmatrix}\text{Her}\\ \text{acting}\\ \text{never}\\ \text{fails}\\ \text{to}\\ \text{impress}\end{bmatrix} = \begin{bmatrix}0\\0\\1\\1\\0\\1\end{bmatrix} = \mathbf{k}_4$$

Transform

Aliasing 1

$$\begin{aligned} u_1(00) &= f(110011) \\ u_1(01) &= f(110111) \\ u_1(10) &= f(111011) \\ u_1(11) &= f(111111) \end{aligned} \longrightarrow \begin{aligned} U_1(00) &= 0 \\ U_1(01) &= F(\mathbf{k}_3) \\ U_1(10) &= F(\mathbf{k}_1) \\ U_1(11) &= F(\mathbf{k}_2) + F(\mathbf{k}_4) \end{aligned}$$

Aliasing 2

$$\begin{aligned} u_2(00) &= f(111100) \\ u_2(01) &= f(111101) \\ u_2(10) &= f(111110) \\ u_2(11) &= f(111111) \end{aligned} \longrightarrow \begin{aligned} U_2(00) &= F(\mathbf{k}_1) + F(\mathbf{k}_2) + F(\mathbf{k}_3) \\ U_2(01) &= F(\mathbf{k}_4) \\ U_2(10) &= 0 \\ U_2(11) &= 0 \end{aligned}$$

Zeroton

Singleton

Multiton

Q: How do we identify singleton/multitons, and find the k_i?

A: Exploit group testing again!

Step 2: Group Testing

Lemma 2. *Consider* $\mathbf{H} \in \mathbf{Z}_2^{b \times n}$, $b < n$ *and* $f : \mathbb{Z}_2^n \mapsto \mathbb{R}$. *Let*

$$u(\boldsymbol{\ell}) = f\left(\overline{\mathbf{H}^T \overline{\boldsymbol{\ell}} + \mathbf{d}}\right), \ \forall \boldsymbol{\ell} \in \mathbb{Z}_2^b.$$

If U *is the Mobius transform of* u, *and* F *is the Mobius transform of* f *we have:*

$$U(\mathbf{j}) = \sum_{\substack{\mathbf{Hk}=\mathbf{j} \\ \mathbf{k} \leq \overline{\mathbf{d}}}} F(\mathbf{k}).$$

If we construct a matrix $\mathbf{D} \in \mathbb{Z}_2^{P \times n}$, *and repeat this process for each row of* $\mathbf{D}$. *We can construct* $\mathbf{y} = \mathbf{Dk}$.

Step 2: Group Testing

$$U(\mathbf{j}) = \sum_{\substack{\mathbf{Hk}=\mathbf{j} \\ \mathbf{k} \leq \overline{\mathbf{d}}}} F(\mathbf{k})$$

$\mathbf{D} \in \mathbb{Z}_2^{P \times n}$, repeat for each row of $\mathbf{D}$. We can construct

$$\mathbf{y} = \mathbf{D}\mathbf{k}$$

Uniform Prior

$$\mathbf{D} = \mathbf{I} \implies \mathbf{y} = \mathbf{k}$$

t-degree

$$\begin{matrix} |\mathbf{k}| < t \\ \mathbf{D} \text{ is } t \text{ group testing} \end{matrix} \implies \mathrm{dec}(\mathbf{y}) = \mathbf{k}$$

Step 3: Message Passing - (Erasure Decoding)

Non-zero Interactions

$$\begin{bmatrix}\text{Her}\\ \text{acting}\\ \text{never}\\ \text{fails}\\ \text{to}\\ \text{impress}\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix} = \mathbf{k}_1 \qquad \begin{bmatrix}\text{Her}\\ \text{acting}\\ \text{never}\\ \text{fails}\\ \text{to}\\ \text{impress}\end{bmatrix} = \begin{bmatrix}0\\0\\1\\1\\0\\0\end{bmatrix} = \mathbf{k}_2$$

$$\begin{bmatrix}\text{Her}\\ \text{acting}\\ \text{never}\\ \text{fails}\\ \text{to}\\ \text{impress}\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\\0\\0\end{bmatrix} = \mathbf{k}_3 \qquad \begin{bmatrix}\text{Her}\\ \text{acting}\\ \text{never}\\ \text{fails}\\ \text{to}\\ \text{impress}\end{bmatrix} = \begin{bmatrix}0\\0\\1\\1\\0\\1\end{bmatrix} = \mathbf{k}_4$$

Transform

Aliasing 1

$$\begin{aligned} u_1(00) &= f(110011) \\ u_1(01) &= f(110111) \\ u_1(10) &= f(111011) \\ u_1(11) &= f(111111) \end{aligned} \longrightarrow \begin{aligned} U_1(00) &= 0 \\ U_1(01) &= F(\mathbf{k}_3) \\ U_1(10) &= F(\mathbf{k}_1) \\ U_1(11) &= F(\mathbf{k}_2) + F(\mathbf{k}_4) \end{aligned}$$

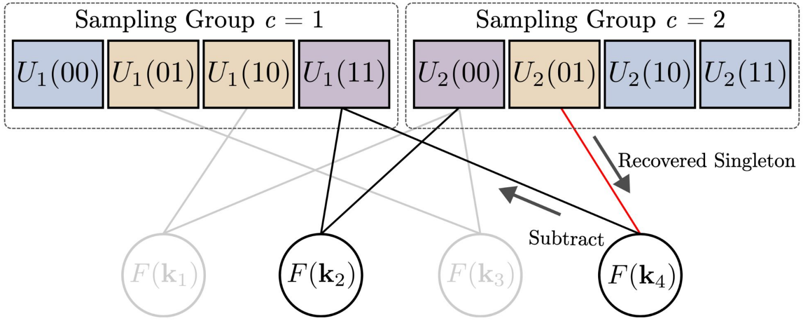
Aliasing 2

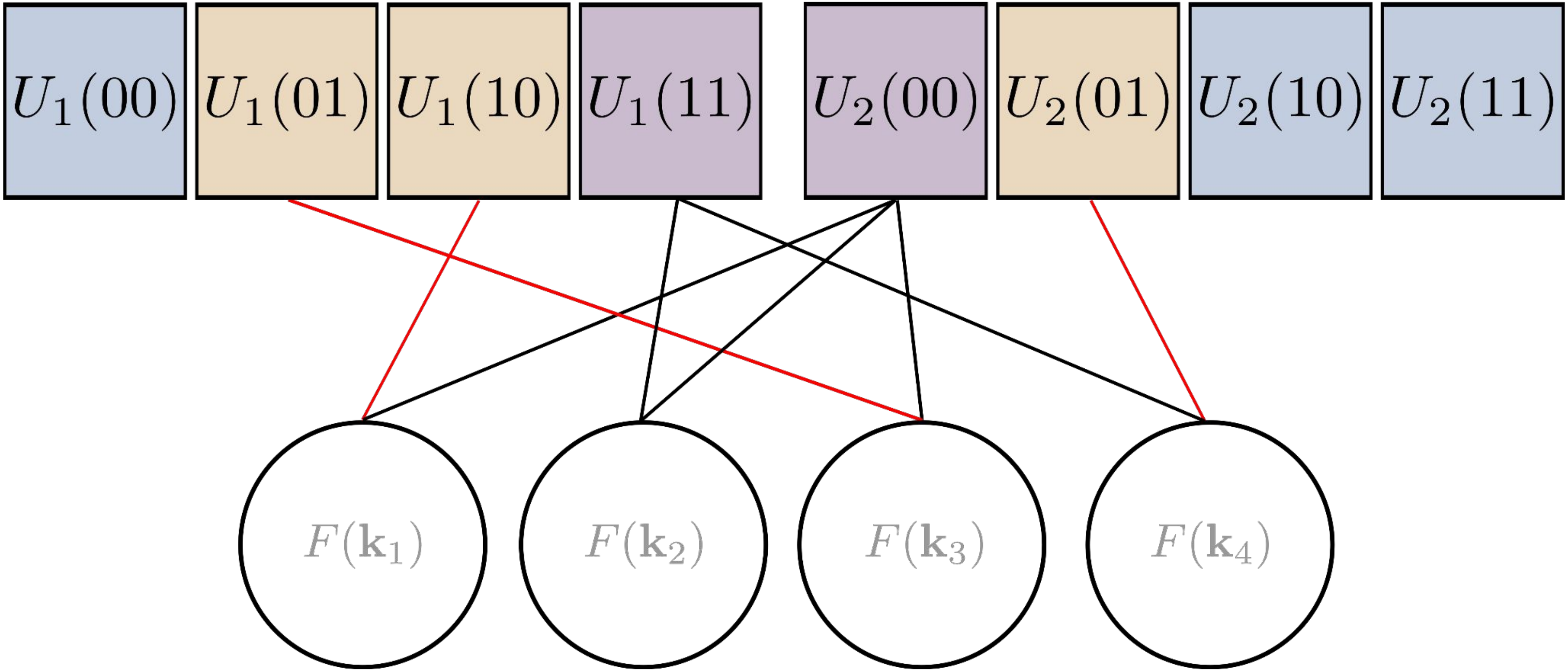
$$\begin{aligned} u_2(00) &= f(111100) \\ u_2(01) &= f(111101) \\ u_2(10) &= f(111110) \\ u_2(11) &= f(111111) \end{aligned} \longrightarrow \begin{aligned} U_2(00) &= F(\mathbf{k}_1) + F(\mathbf{k}_2) + F(\mathbf{k}_3) \\ U_2(01) &= F(\mathbf{k}_4) \\ U_2(10) &= 0 \\ U_2(11) &= 0 \end{aligned}$$

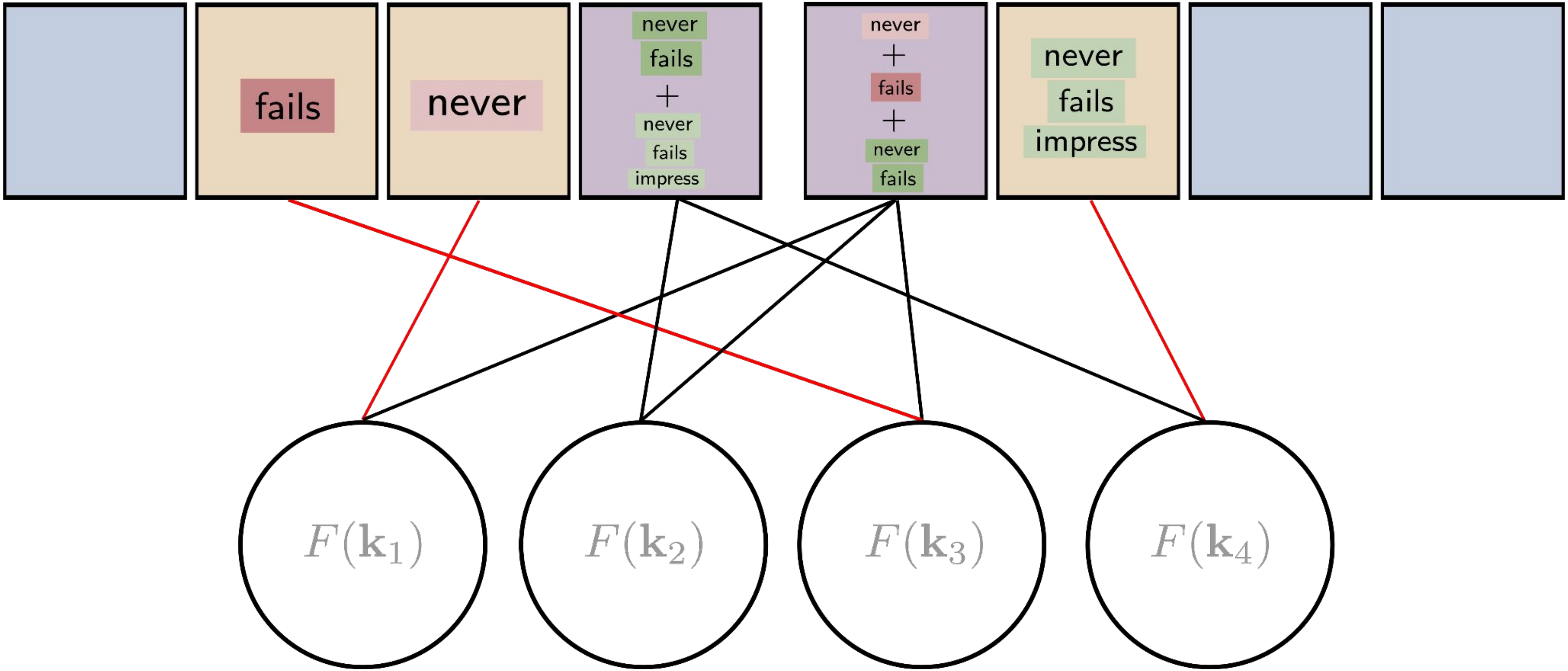
Zeroton

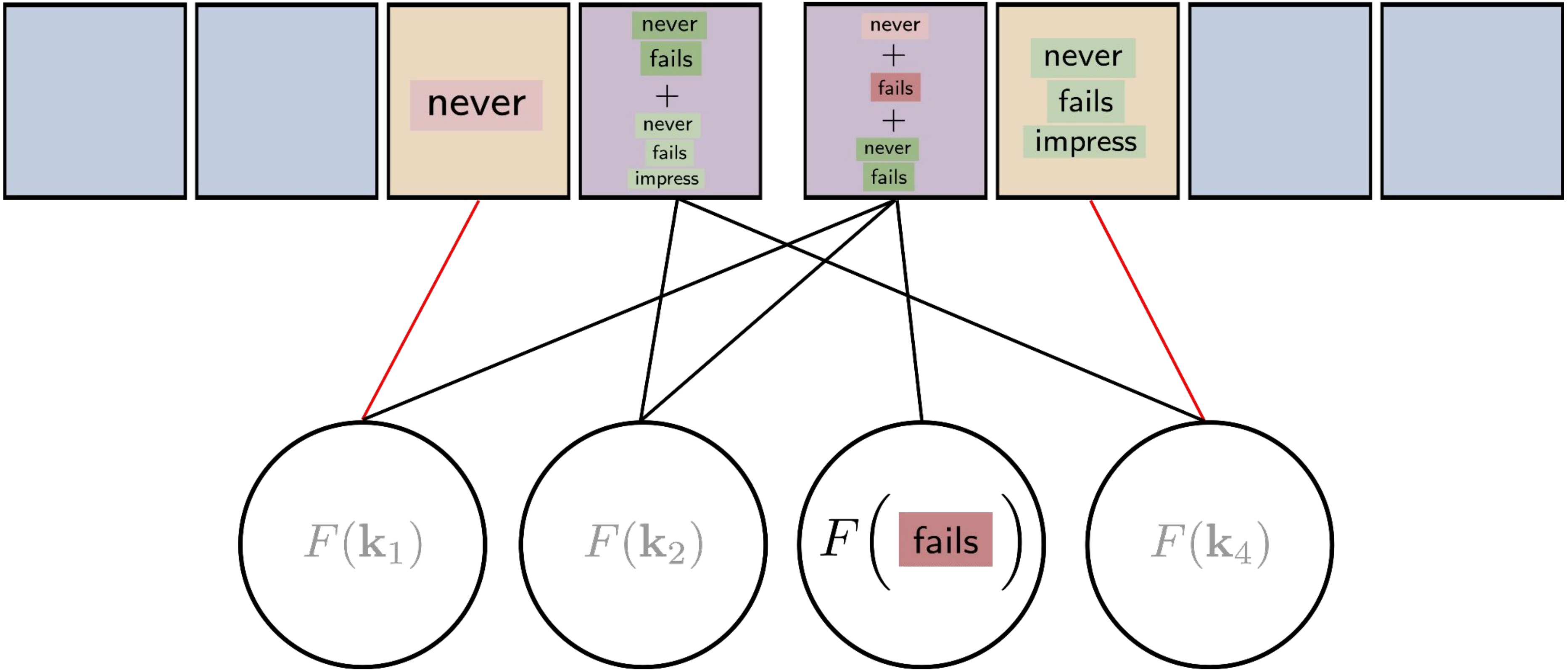
Singleton

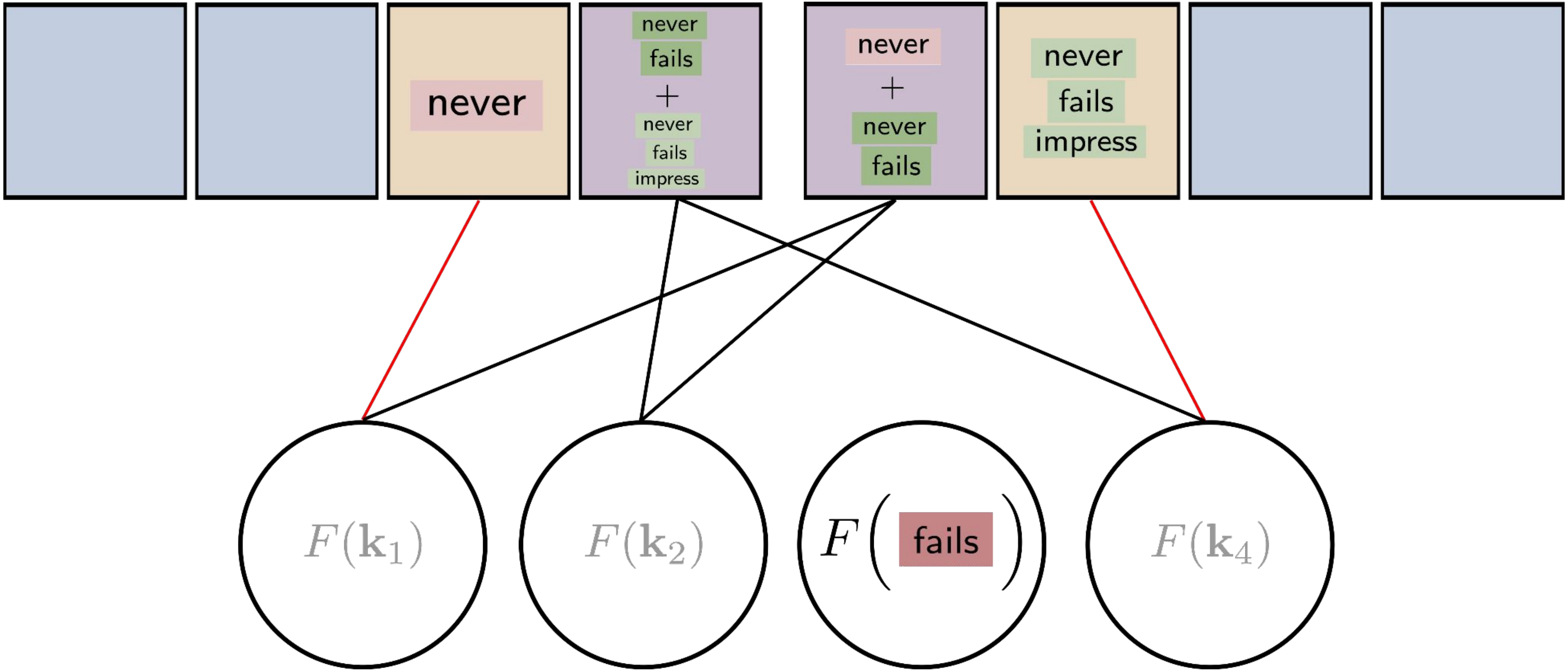
Multiton

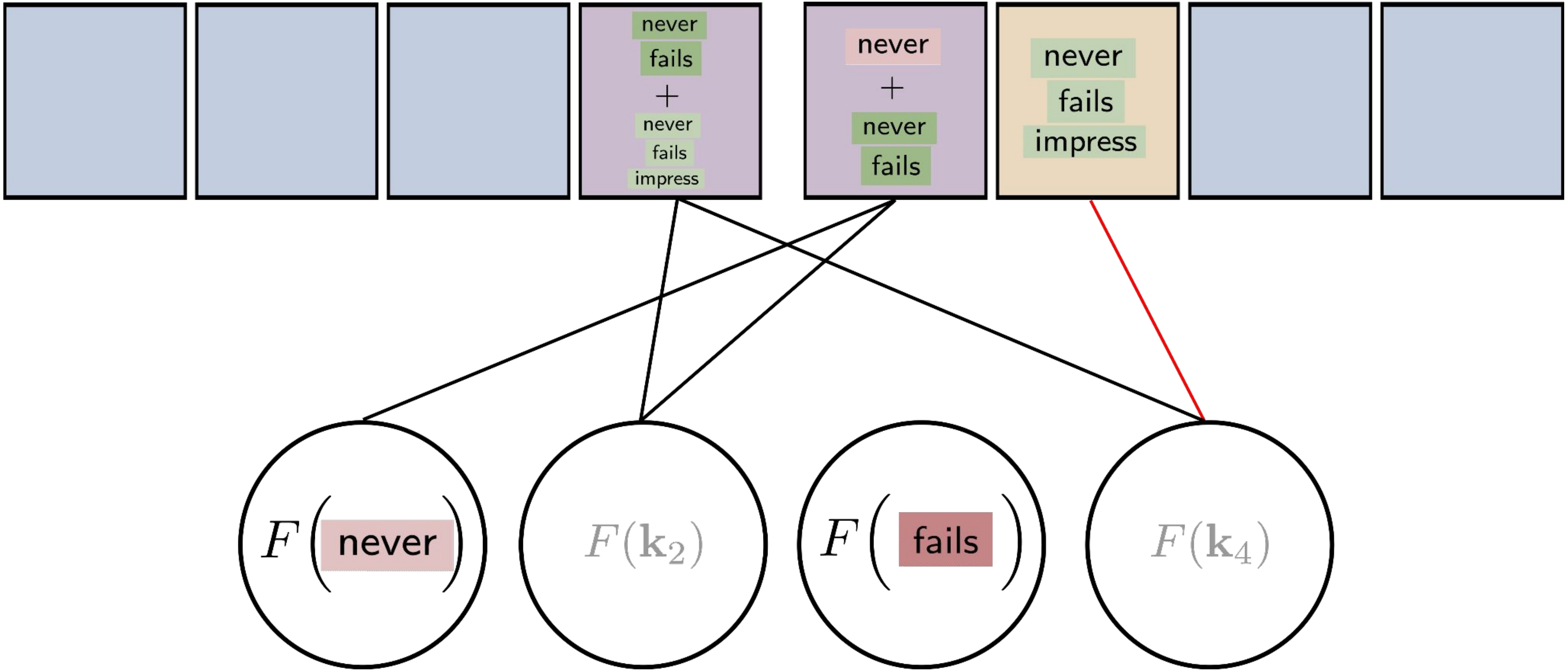


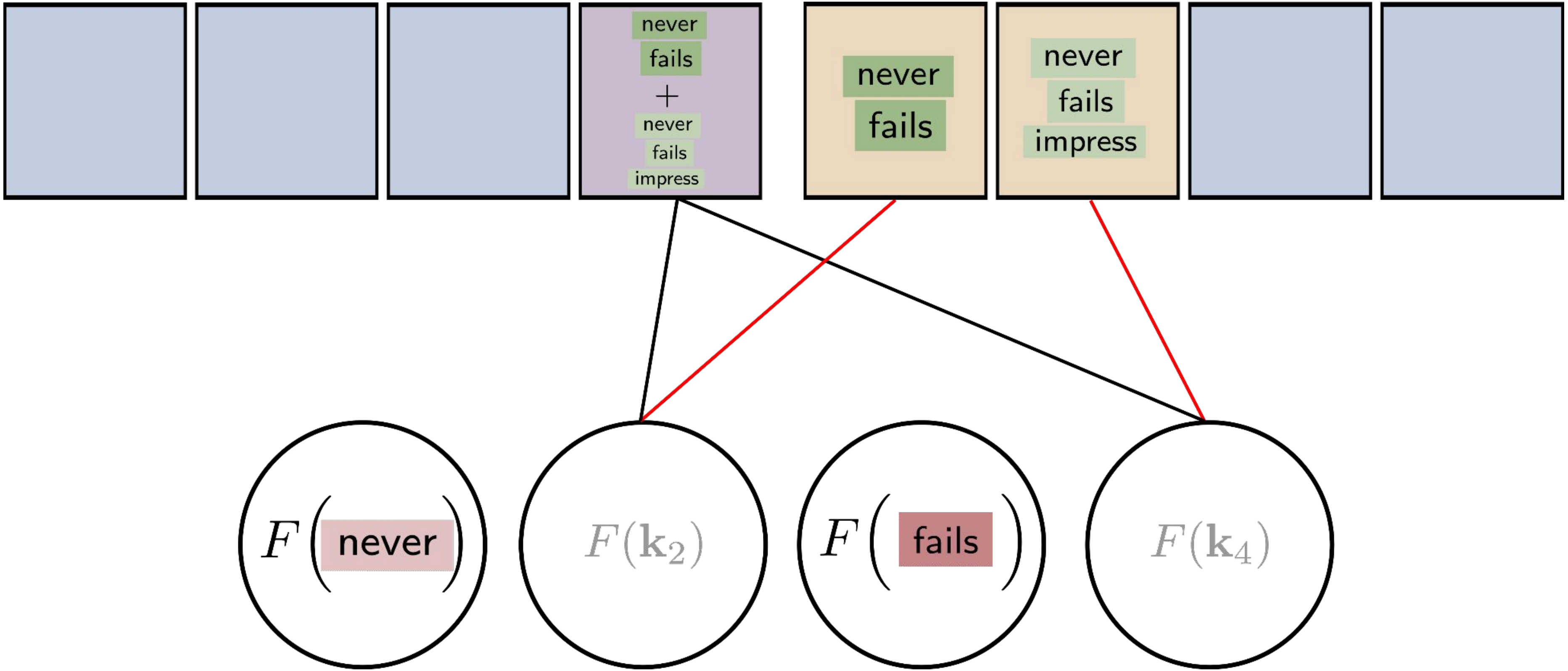

$U_1(00)$
$U_1(01)$
$U_1(10)$
$U_1(11)$
$U_2(00)$
$U_2(01)$
$U_2(10)$
$U_2(11)$
$F(\mathbf{k}_1)$
$F(\mathbf{k}_2)$
$F(\mathbf{k}_3)$
$F(\mathbf{k}_4)$

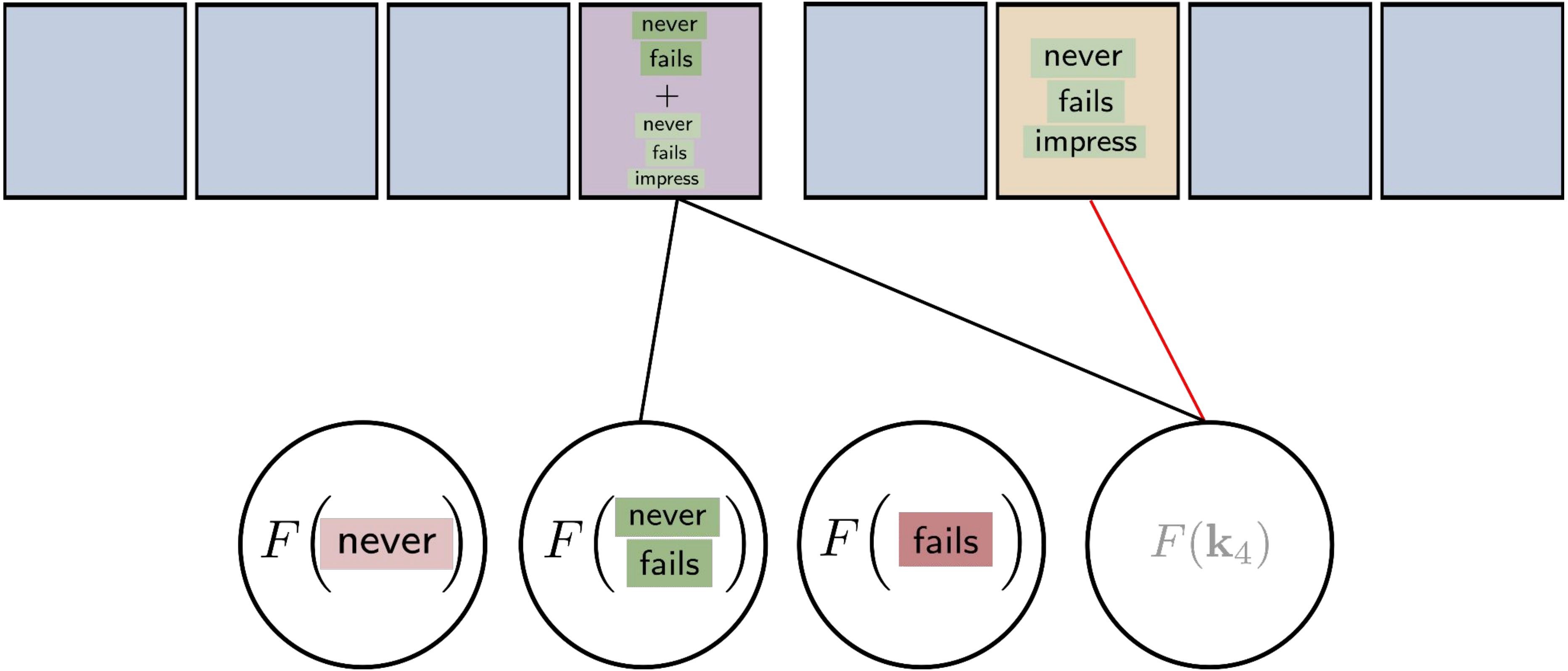

fails
never
never
fails
+
never
fails
impress
never
+
fails
+
never
fails
never
fails
impress
$F(\mathbf{k}_1)$
$F(\mathbf{k}_2)$
$F(\mathbf{k}_3)$
$F(\mathbf{k}_4)$

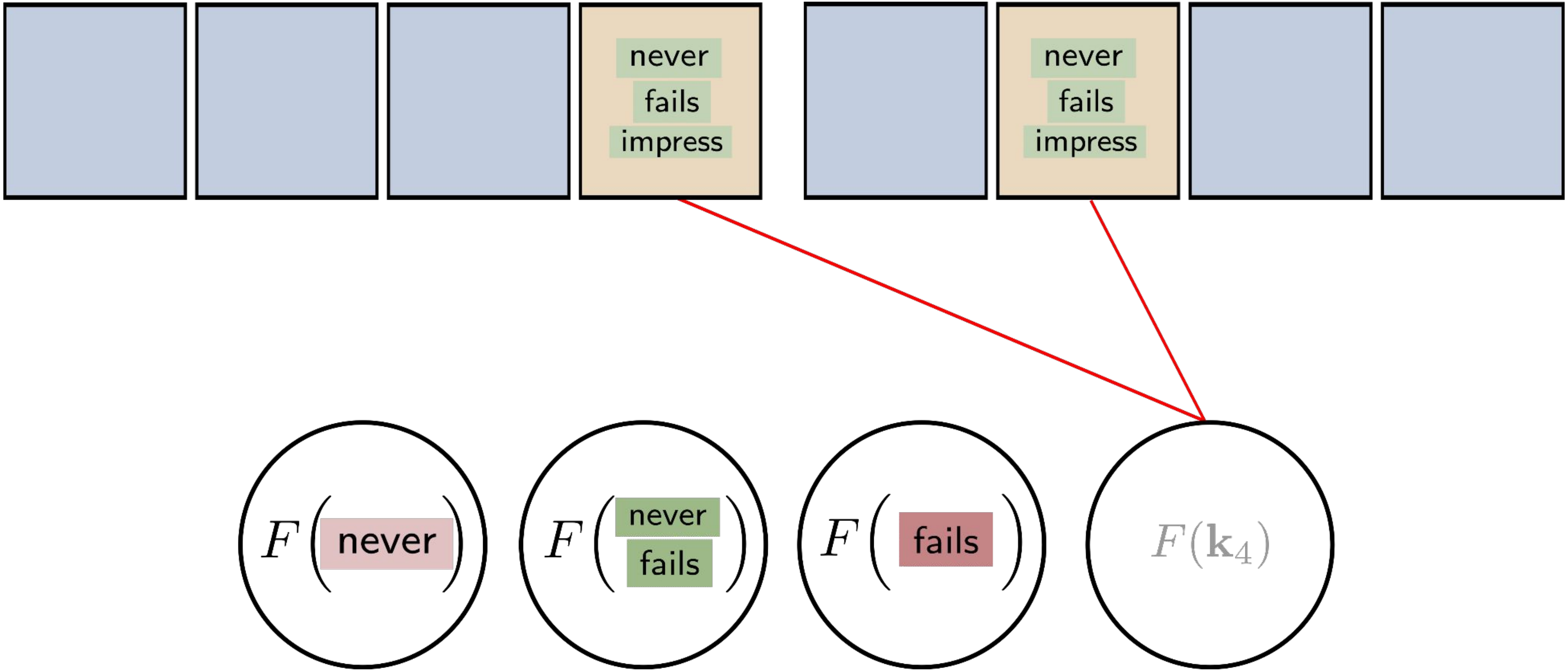

never
never
fails
+
never
fails
impress
never
+
fails
+
never
fails
never
fails
impress
$F(\mathbf{k}_1)$
$F(\mathbf{k}_2)$
$F($ fails $)$
$F(\mathbf{k}_4)$

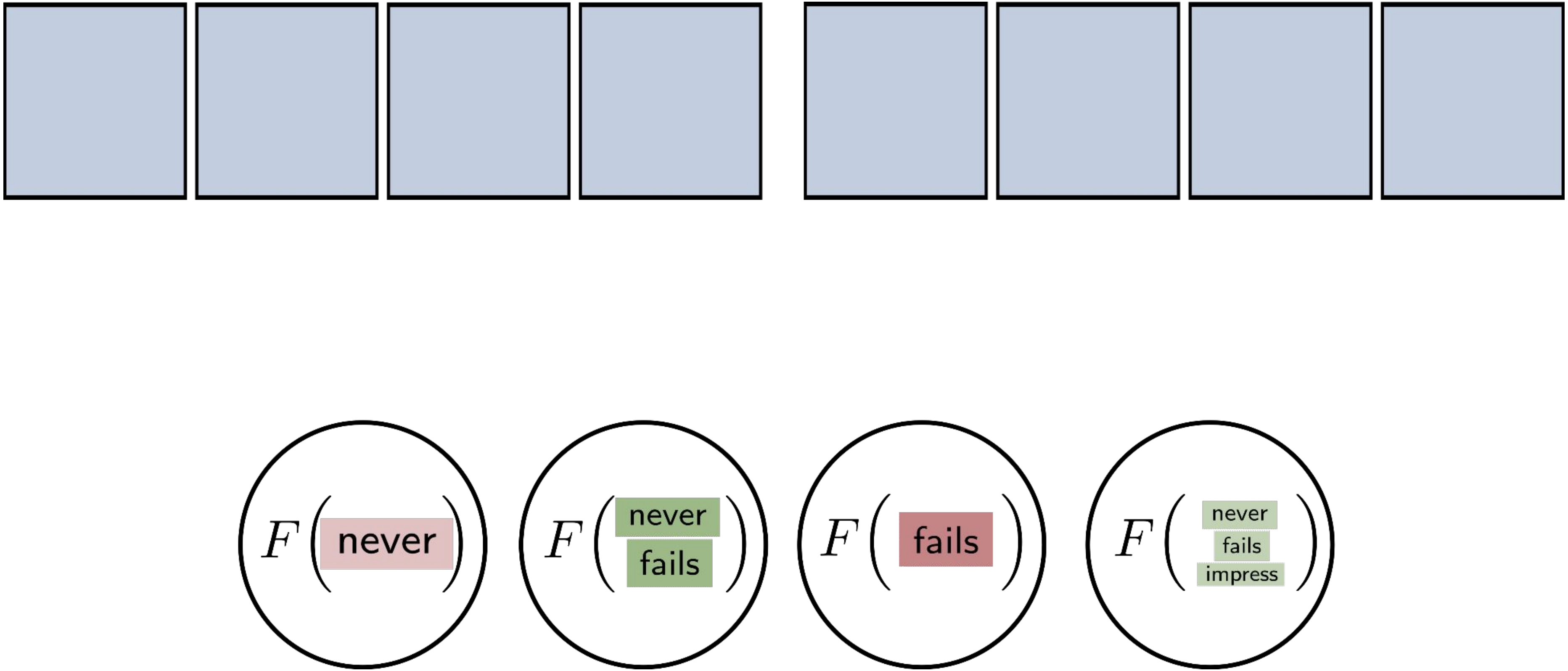

never
never
fails
+
never
fails
impress
never
+
never
fails
never
fails
impress
$F(\mathbf{k}_1)$
$F(\mathbf{k}_2)$
$F($ fails $)$
$F(\mathbf{k}_4)$


never
fails
+
never
fails
impress
never
+
never
fails
never
fails
impress
$F($never$)$
$F(\mathbf{k}_2)$
$F($fails$)$
$F(\mathbf{k}_4)$


never
fails
+
never
fails
impress
never
fails
never
fails
impress
$F($never$)$
$F(\mathbf{k}_2)$
$F($fails$)$
$F(\mathbf{k}_4)$


never
fails
+
never
fails
impress
never
fails
impress
$F($never$)$
$F($never fails$)$
$F($fails$)$
$F(\mathbf{k}_4)$


never
fails
impress
never
fails
impress
$F($never$)$
$F($never fails$)$
$F($fails$)$
$F(\mathbf{k}_4)$



F(never)
F(never fails)
F(fails)
F(never fails impress)

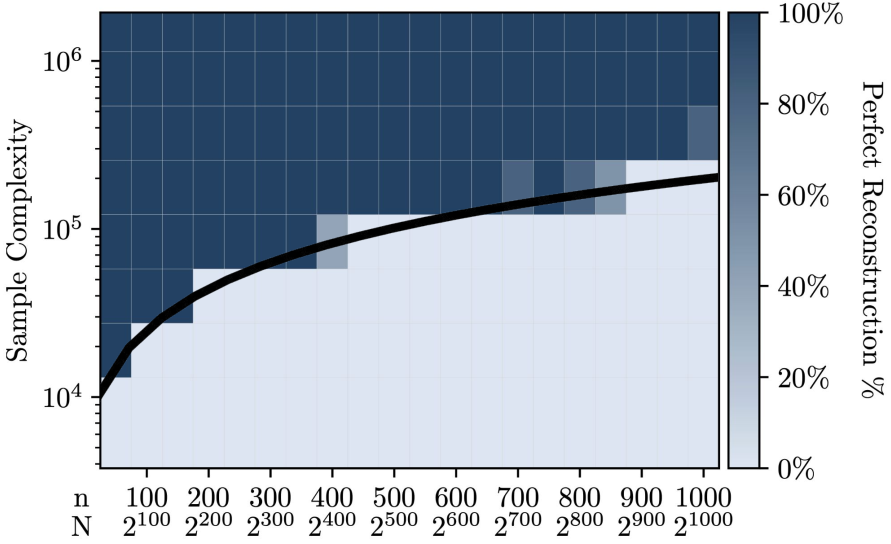
Overview

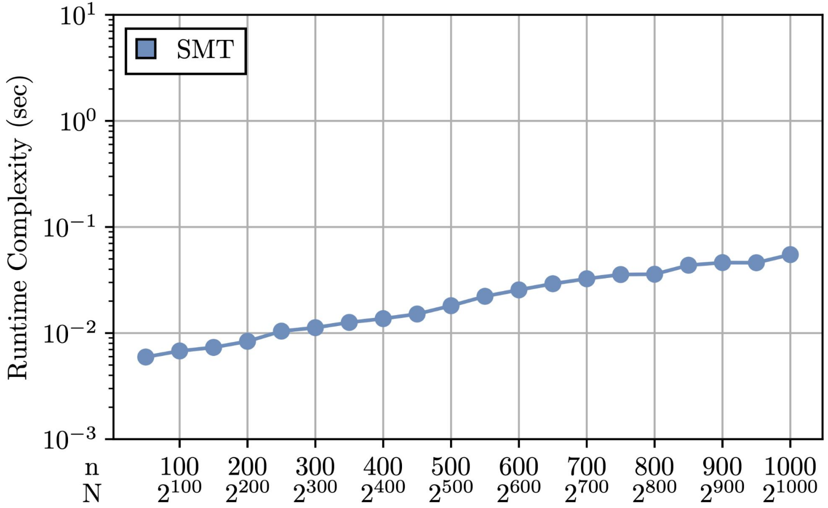
- **Hashing:** Take $O(K)$ samples of the function according to group testing matrix
- **Singleton Identification/Detection:** Repeat this process
  - $O(n)$ times under uniform prior
  - $O(t\log(n))$ times under the t-degree assumption
- **Message Passing Decoding:** Repeat the entire process only $O(1)$ times for density evolution to work

Uniform: $O(Kn)$ samples

t-degree: $O(Kt\log(n))$ samples

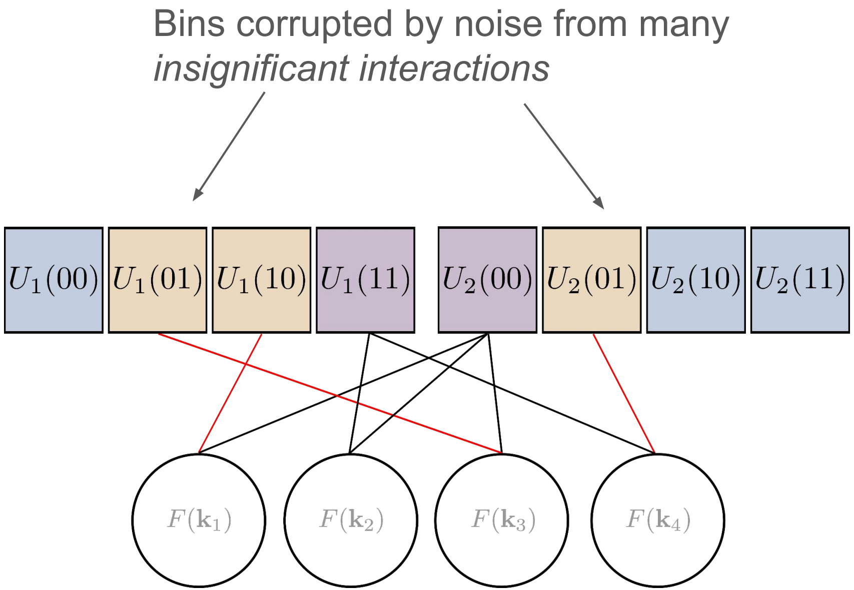
Simulation Results





Sample complexity O(K*n)
Plotted against 2*K*n for K=100

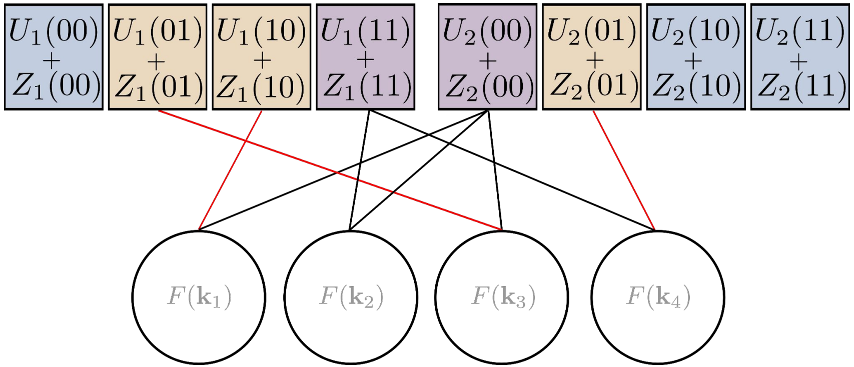
Robustness?



Bins corrupted by noise from many *insignificant interactions*
$U_1(00)$
$U_1(01)$
$U_1(10)$
$U_1(11)$
$U_2(00)$
$U_2(01)$
$U_2(10)$
$U_2(11)$
$F(\mathbf{k}_1)$
$F(\mathbf{k}_2)$
$F(\mathbf{k}_3)$
$F(\mathbf{k}_4)$

Robustness?

Bins corrupted by noise from many
insignificant interactions



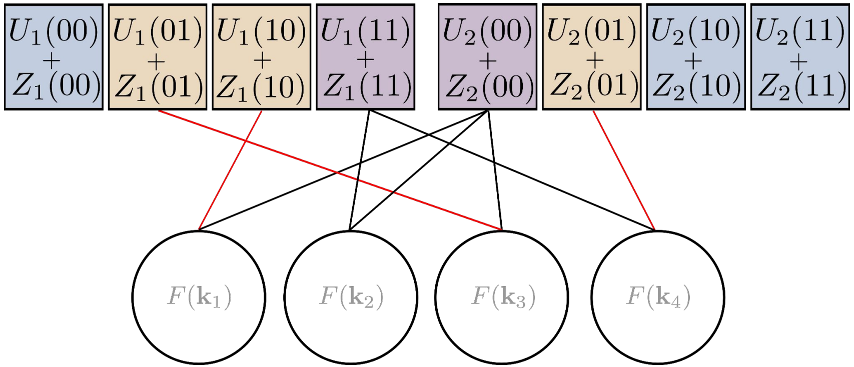
$$\mathrm{SNR} = \frac{\min F(\mathbf{k})^2}{\sigma^2}$$

$$Z_i(\cdot) \sim \mathcal{N}(0, \sigma^2)$$

Robustness?

Bins corrupted by noise from many
insignificant interactions

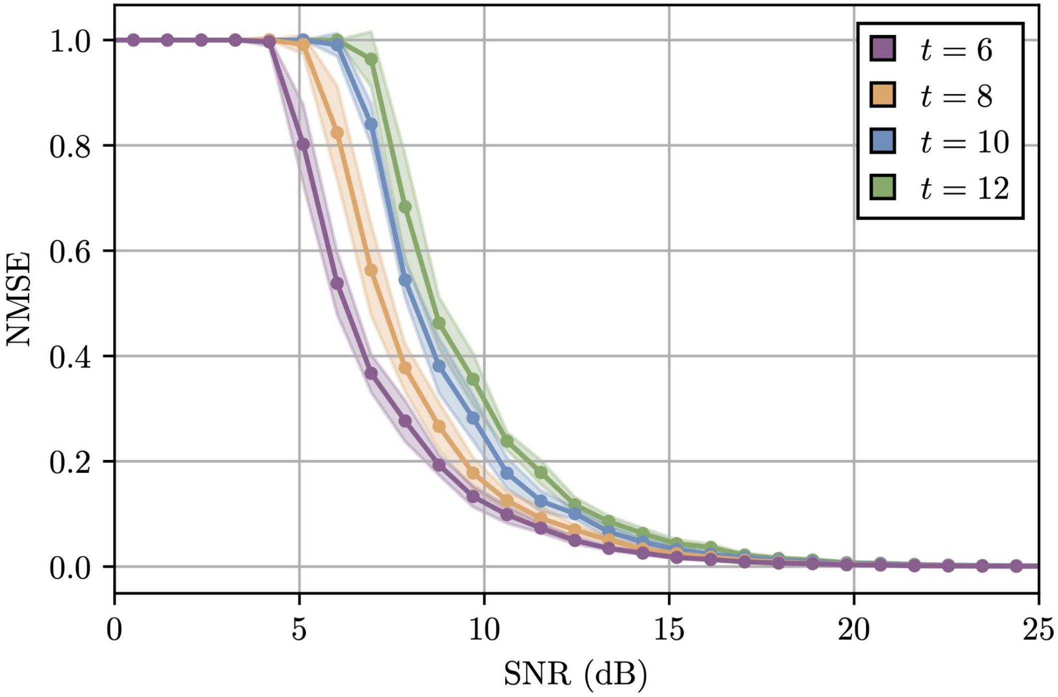
$$\mathrm{SNR} = \frac{\min F(\mathbf{k})^2}{\sigma^2}$$



$$Z_i(\cdot) \sim \mathcal{N}(0, \sigma^2)$$

- To address noise we take additional **redundant group tests**
- Noisy group testing theory (Scarlett, 2022), gives us a theoretical guarantee

Robust Algorithm Simulations



$$\mathrm{NMSE} = \|\hat{F} - F\|^2 / \|F\|^2$$

$$n = 500, K = 500, P = 1000$$

Limitations: Theorem Revisited, Open Questions

Theorem 2. **(Robust Low-Decoding, Informal)** When there are K non-zero interactions chosen uniformly over all $|\mathbf{k}| \leq t$, with $t = \Theta(n^{\alpha})$, $\alpha \leq 0.407$ our algorithm computes the Mobius transform:

- with sample complexity $O(Kt\log(n))$ and
- with time $O(K\text{poly}(n))$

with probability $1 - O(1/K)$ with any fixed SNR.

- In “real” signals/functions important/big interactions are correlated
- Adaptive versions of the transform can eliminate this assumption
- Can we eliminate this assumption and remain non-adaptive?

Applications - Explaining Images

- Transformers can help us generate interpretable orthogonal features

- Masking these features can give us more interpretable explanations

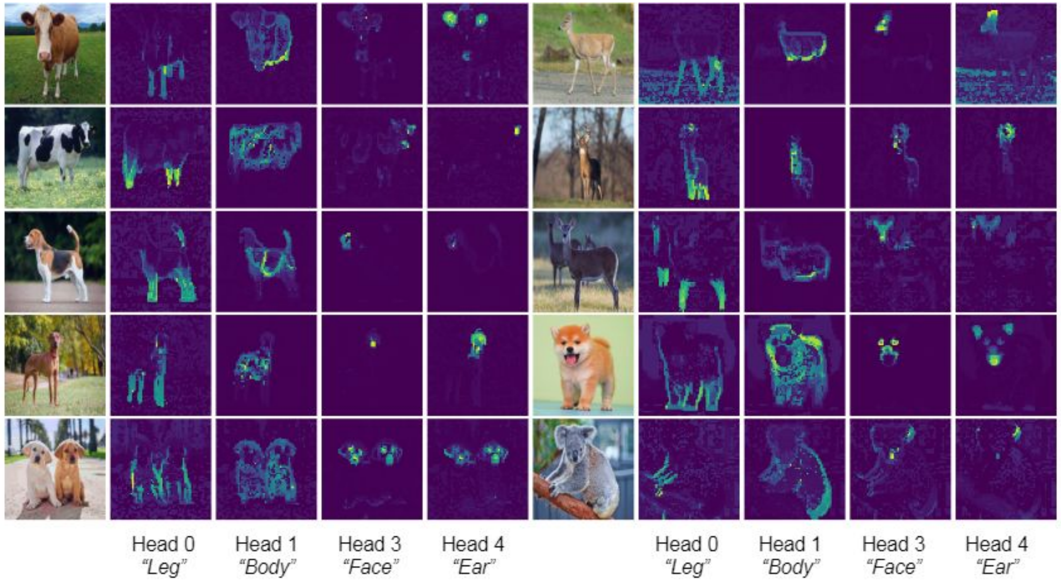
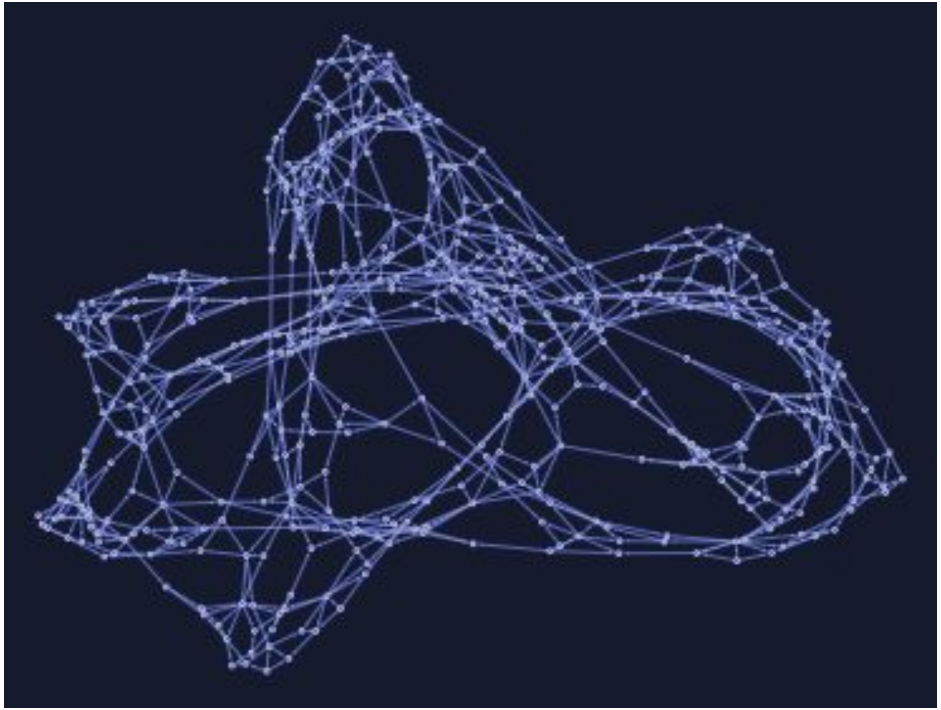


Figure 5: Visualization of semantic heads. We forward a mini-batch of images through a supervised CRATE and examine the attention maps from all the heads in the penultimate layer. We visualize a selection of attention heads to show that certain heads convey specific semantic meaning, i.e. *head 0* $\leftrightarrow$ *"Legs"*, *head 1* $\leftrightarrow$ *"Body"*, *head 3* $\leftrightarrow$ *"Face"*, *head 4* $\leftrightarrow$ *"Ear"*.

Emergence of Segmentation with Minimalistic White-Box Transformers [paper link]. By Yaodong Yu* (UC Berkeley), Tianzhe Chu* (UC Berkeley & ShanghaiTech U), Shengbang Tong (UC Berkeley & NYU), Ziyang Wu (UC Berkeley), Druv Pai (UC Berkeley), Sam Buchanan (TTIC), and Yi Ma (UC Berkeley & HKU). 2023. (* equal contribution)

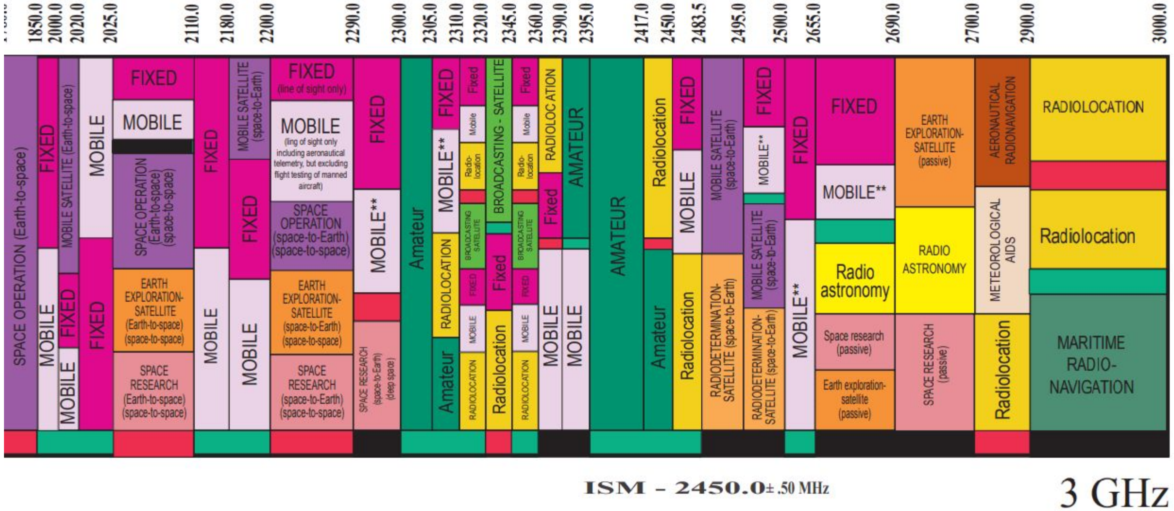
Applications - Sketching Large Hypergraphs

- Hypergraph structures emerge in many different applications
- The Mobius Transform can learn a hypergraph efficiently from looking at the number of hyperedges in subgraphs
- Direct application of this work may be state-of-art for some of these problems



Applications - Auctions and Game Theory

- Auctions may have complex combinatorial structure i.e., **Spectrum Auctions**
- Finding optimal allocations is hard in general, but there is structure
- Can signal processing help scale up auctions?



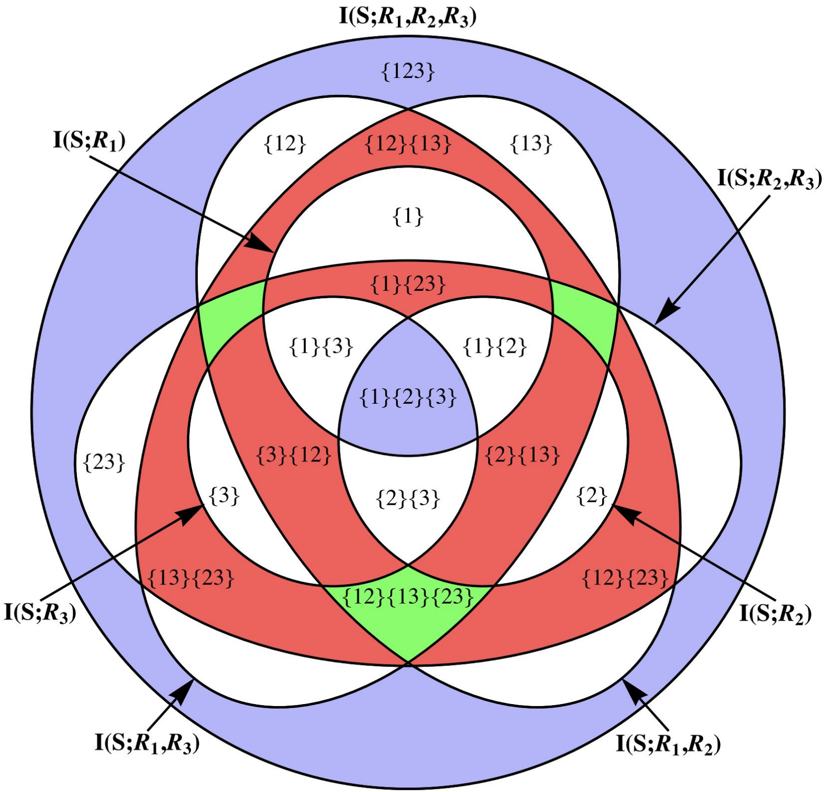
Application - Non-Negative Information Decomposition

- Mobius Transforms are also useful for decomposing mutual information

Example:

$$Z = X \oplus Y, \quad I(Z;X,Y) = 1$$

$$\begin{aligned} \mathrm{Syn}(Z;X,Y) &= 1 \\ \mathrm{Unq}(Z;X) &= 0 \\ \mathrm{Unq}(Z;Y) &= 0 \\ \mathrm{Rdn}(Z;X,Y) &= 0 \end{aligned}$$

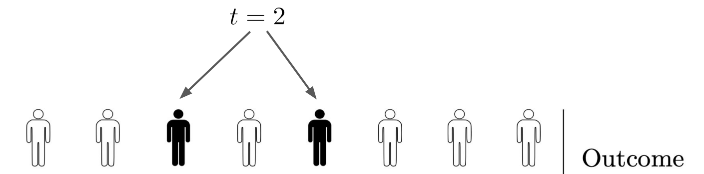


Williams PL, Beer RD. Nonnegative decomposition of multivariate information. arXiv preprint arXiv:1004.2515. 2010 Apr 14.

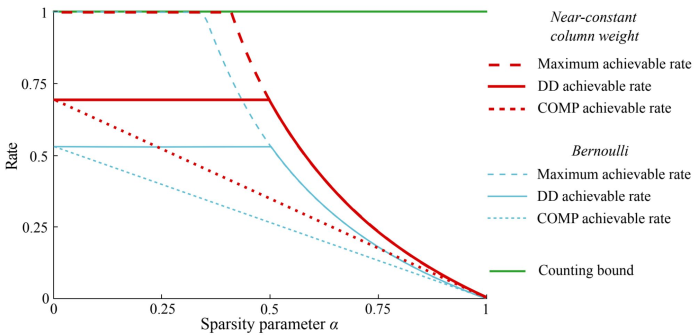
Conclusion

- Explaining deep models can be cast as functional decomposition
- Signal processing and communications ideas can provide a new perspective
- Lots of open problems:
  - Can we leverage white-box access to the model?
  - Can we exploit the connection between attention and Mobius transform?
  - How do we improve robustness in real-world noise models?

Step 2: Group Testing



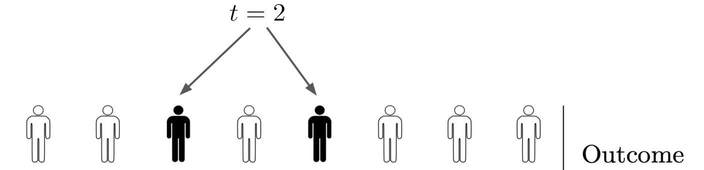
								Outcome
1	1	1	1	0	0	0	0	Positive
0	0	0	0	1	1	1	1	Positive
1	1	0	0	0	0	0	0	Negative
0	0	1	0	0	0	0	0	Positive
0	0	1	0	1	1	0	0	Positive
0	0	0	0	1	0	0	0	Positive



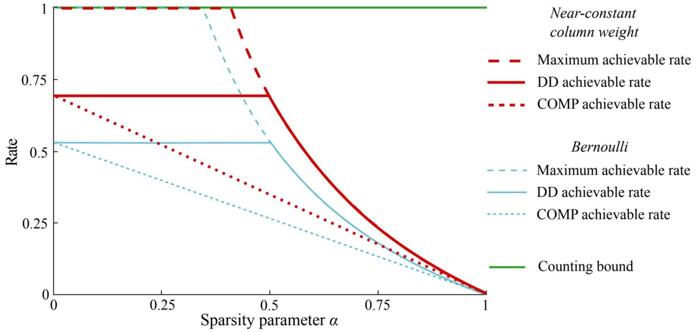
$$t \approx n^{\alpha}$$

- Rate 1 group testing matrices uniformly hash asymptotically
- If α is less than 0.4, a Rate 1 group testing matrix exists
- If interactions are not low degree (uniform prior) *individual testing* is Rate 1

Step 2: Group Testing



								Outcome
1	1	1	1	0	0	0	0	Positive
0	0	0	0	1	1	1	1	Positive
1	1	0	0	0	0	0	0	Negative
0	0	1	0	0	0	0	0	Positive
0	0	1	0	1	1	0	0	Positive
0	0	0	0	1	0	0	0	Positive



$$t \approx n^{\alpha}$$

Rules for designing H:

1. (Low Degree, α < 0.4) Choose O(log(K)) rows of a NCCW matrix
2. (Uniform Prior) Choose O(log(K)) rows of an Identity matrix

This ensures asymptotically uniform hashing