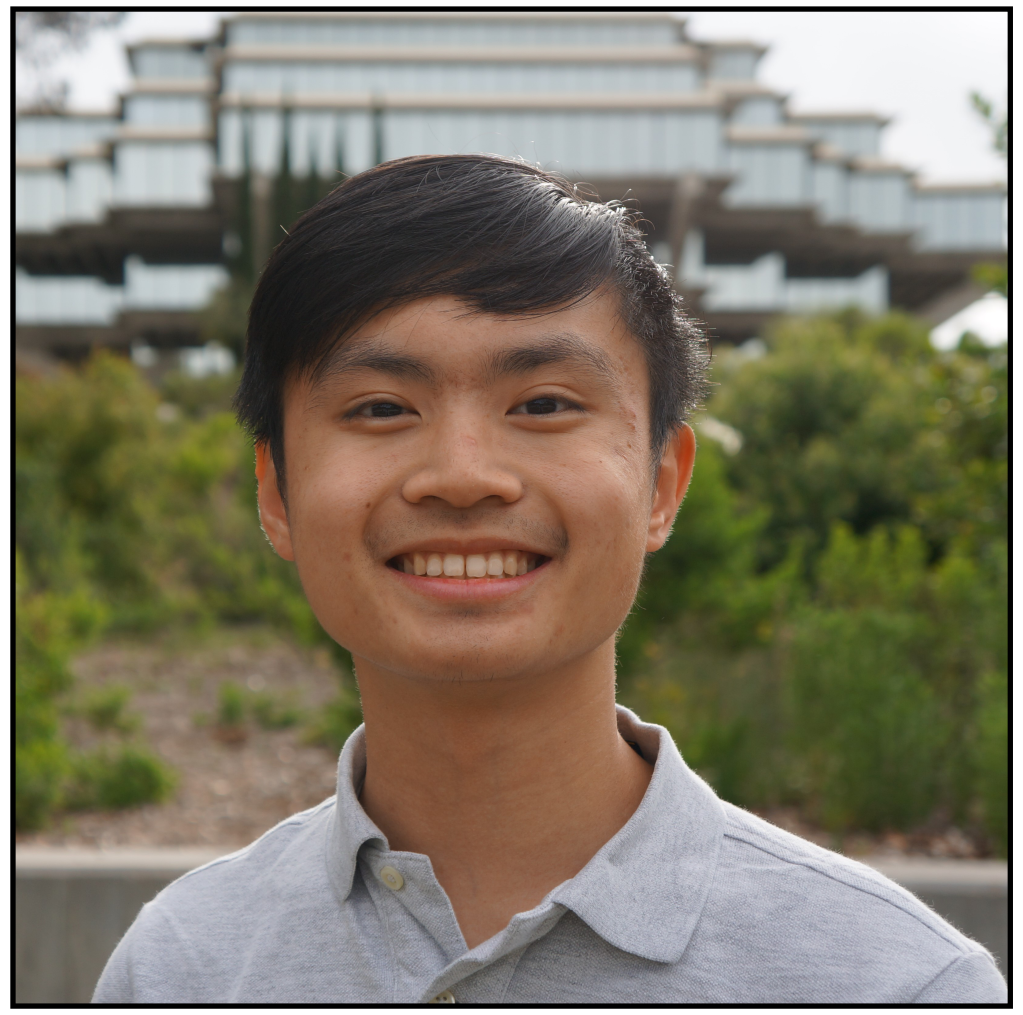
Efficient Algorithm for Sparse Fourier Transform of Generalized q-ary Functions

Darin Tsui*



Kunal Talreja*

Amirali Aghazadeh

Georgia Tech College of Engineering
School of Electrical and Computer Engineering

* = equal contributions

Scaling machine learning models

120B parameters

Scaling machine learning models

120B parameters

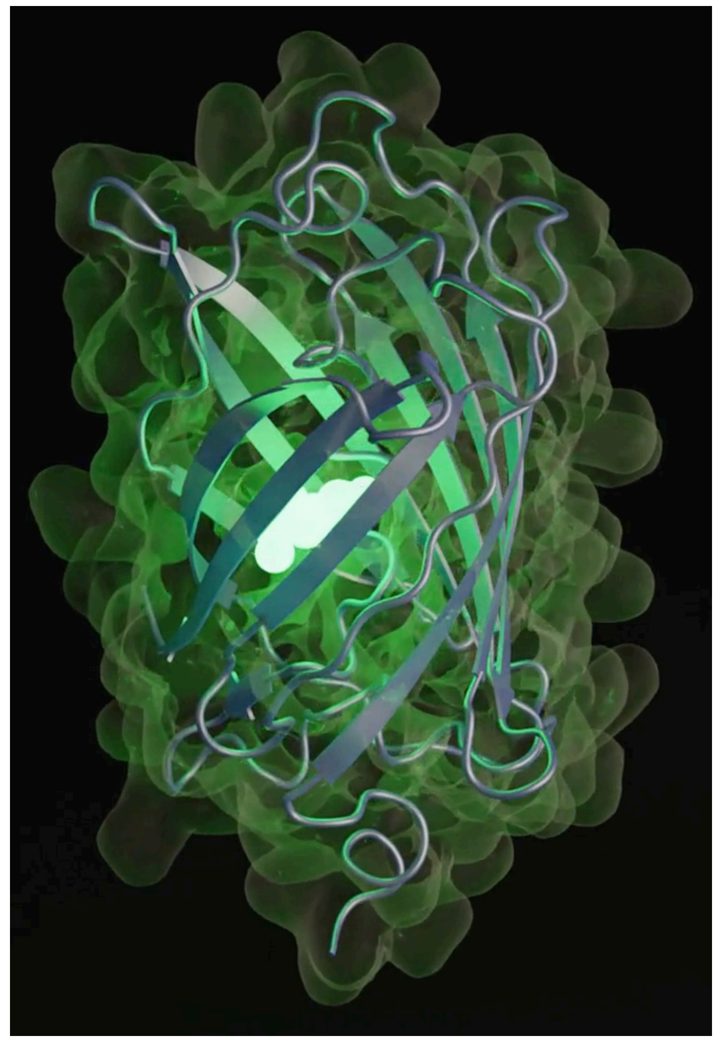
ESM3: Simulating 500 million years of evolution with a language model

Preview our paper ↗

Trained on **2.78 Billion** natural proteins

Scaling machine learning models

120B parameters



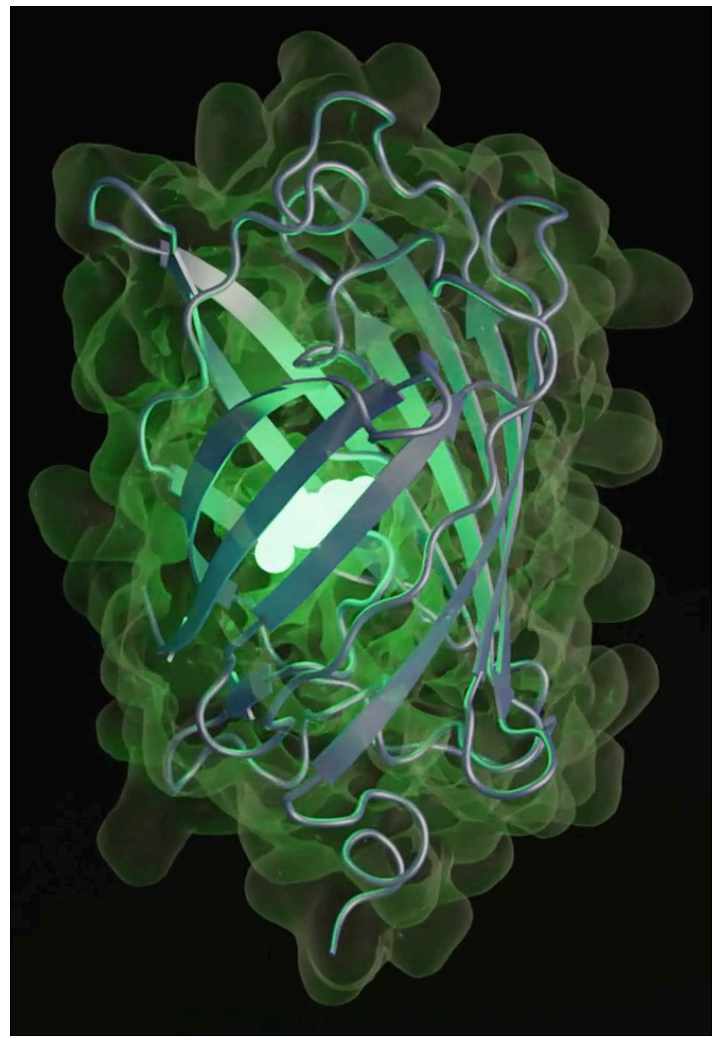
ESM3: Simulating 500 million years of evolution with a language model

Preview our paper ↗

Trained on **2.78 Billion** natural proteins

Scaling machine learning models

120B parameters



ESM3: Simulating 500 million years of evolution with a language model

Preview our paper ↗

Trained on **2.78 Billion** natural proteins

But does our power to **explain** them also **scale**?

These functions can be modeled as *q*-ary functions

$$f : \mathbb{Z}_q^n \rightarrow \mathbb{R}$$

$$f(x_1, x_2, x_3, \cdots, x_n) \rightarrow y$$

These functions can be modeled as *q*-ary functions

$$f : \mathbb{Z}_q^n \to \mathbb{R}$$

$$f(x_1, x_2, x_3, \cdots, x_n) \to y$$

$$x_i \in \{0, 1, \cdots, q - 1\}$$

These functions can be modeled as *q*-ary functions

$$f : \mathbb{Z}_{20}^{n} \to \mathbb{R}$$

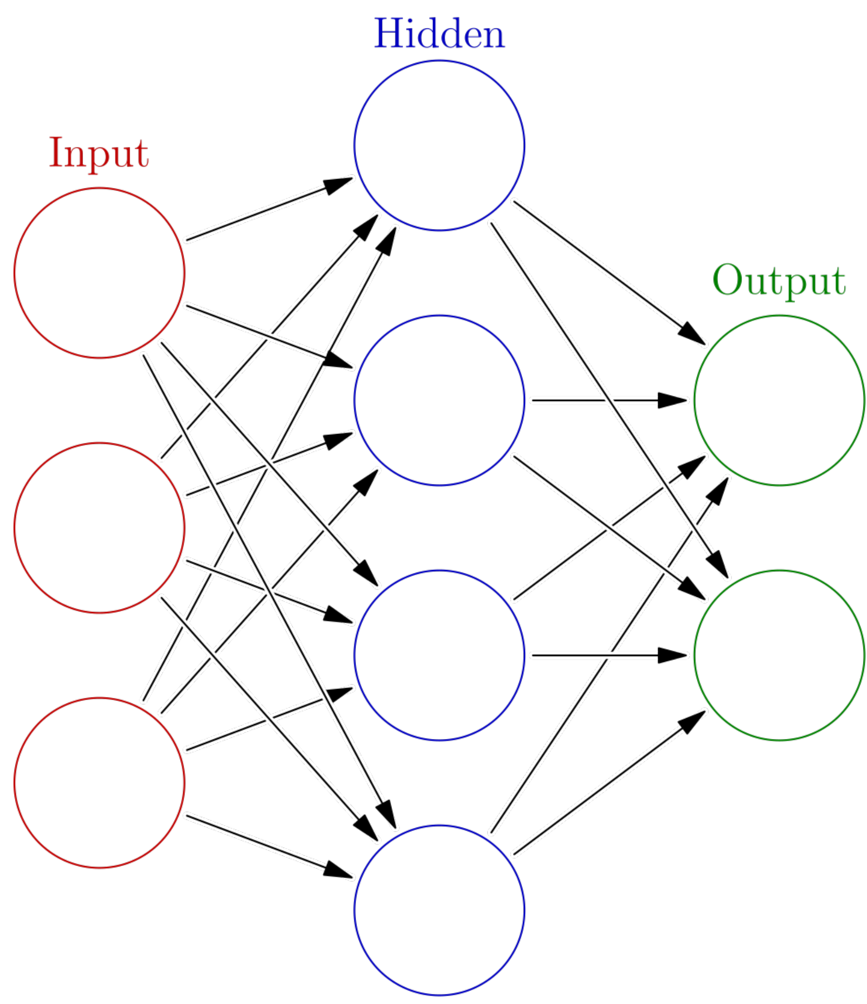
$$f(x_1, x_2, x_3, \cdots, x_n) \to$$ 

$$x_i \in \{P, L, S, \cdots, C\}$$

These functions can be modeled as *q*-ary functions

$$f : \mathbb{Z}_3^n \to \mathbb{R}$$

$$f(x_1, x_2, x_3, \cdots, x_n) \to$$



$$x_i \in \{\text{Class } 1, \text{Class } 2, \text{Class } 3\}$$

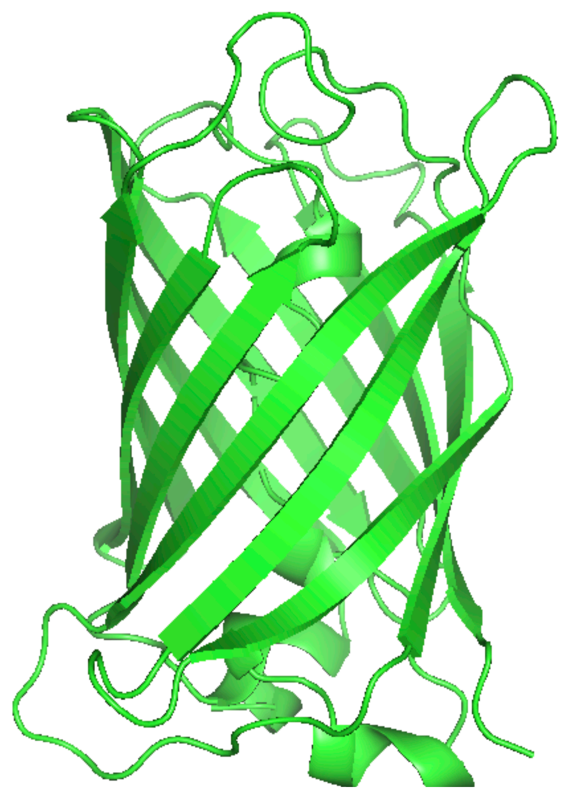
These functions can be modeled as *q*-ary functions

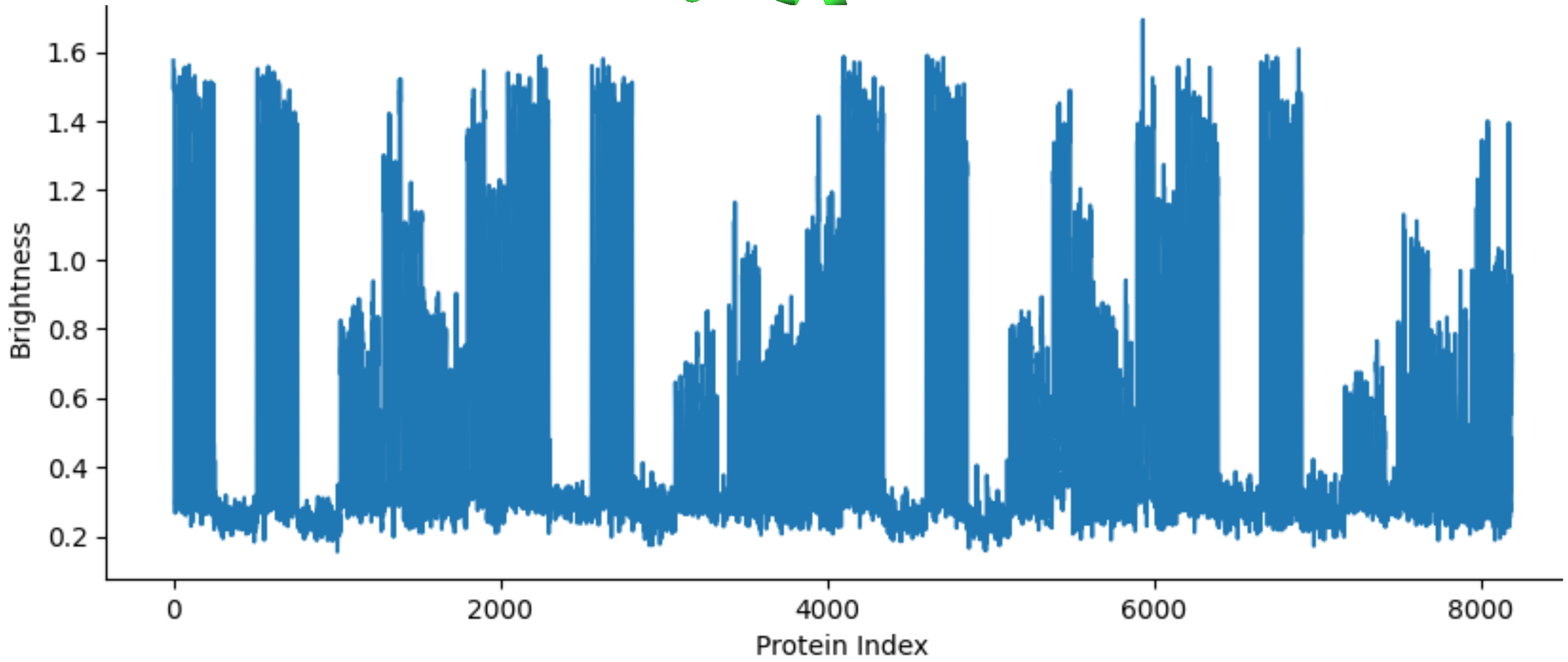
$$f : \mathbb{Z}_4^n \rightarrow \mathbb{R}$$

$$f(x_1, x_2, x_3, \cdots, x_n) \rightarrow$$ 

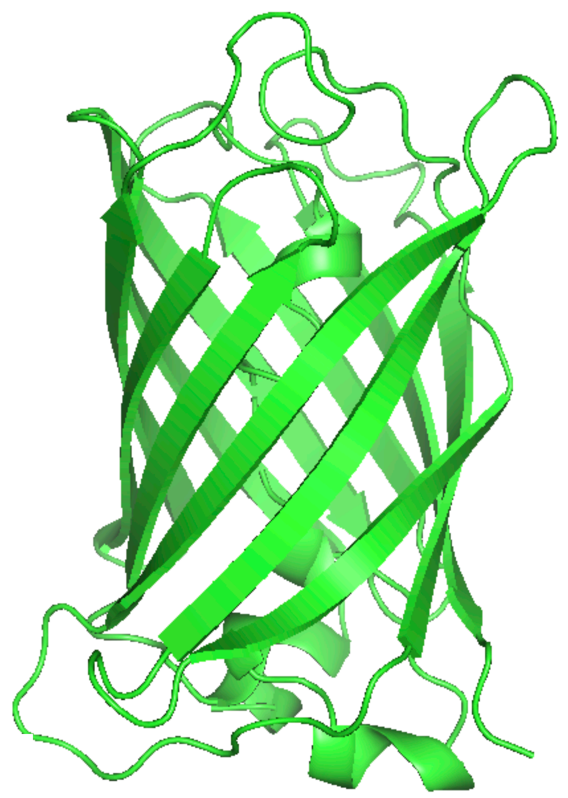
$$x_i \in \{\text{"love"}, \text{"hate"}, \text{"regret"}, \text{"miss"}\}$$

Many functions of interest have *sparse Fourier transforms*

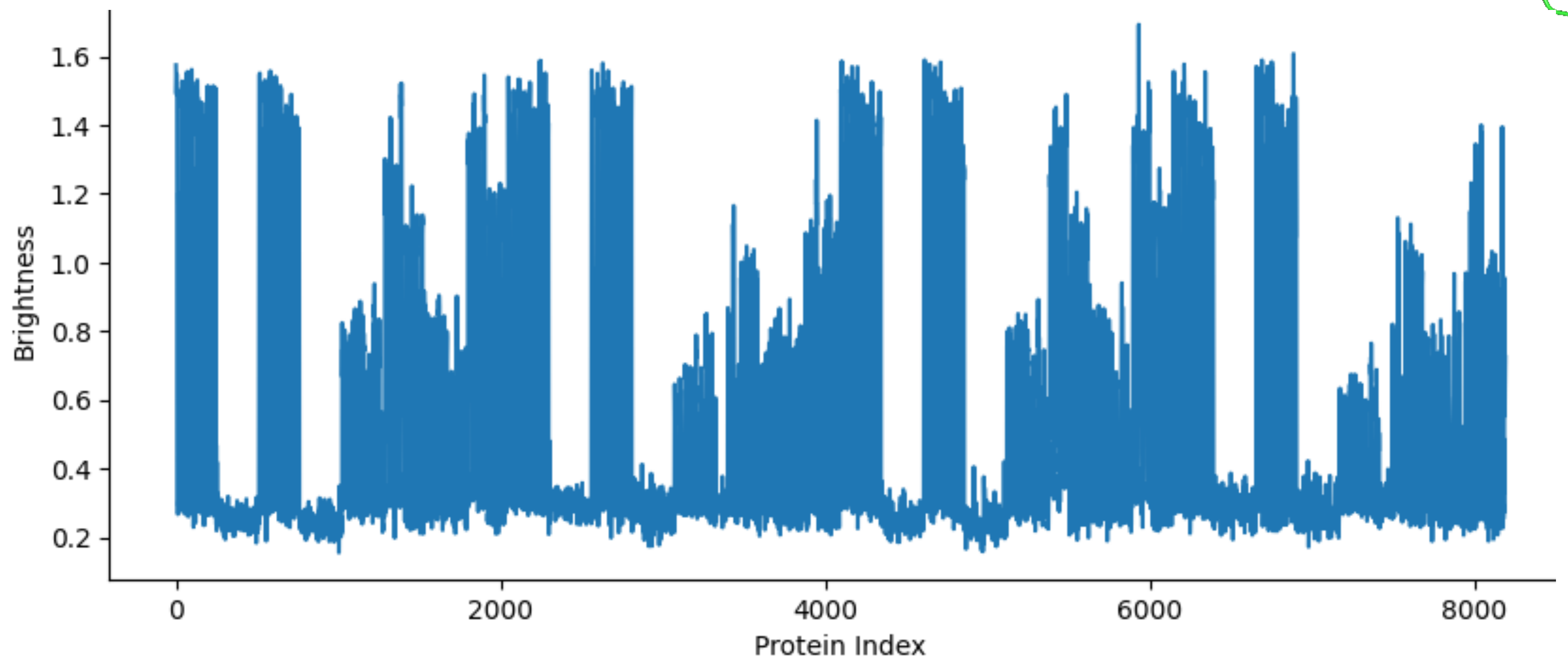


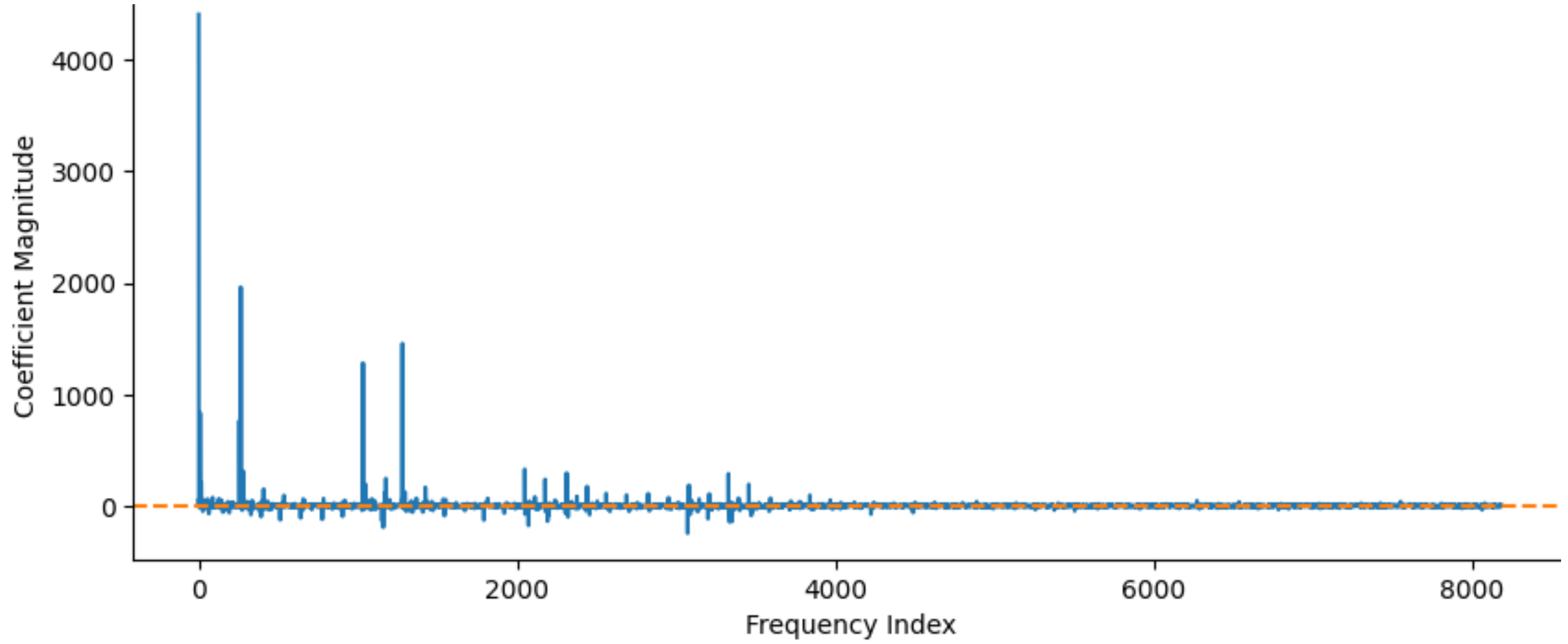


Many functions of interest have *sparse Fourier transforms*



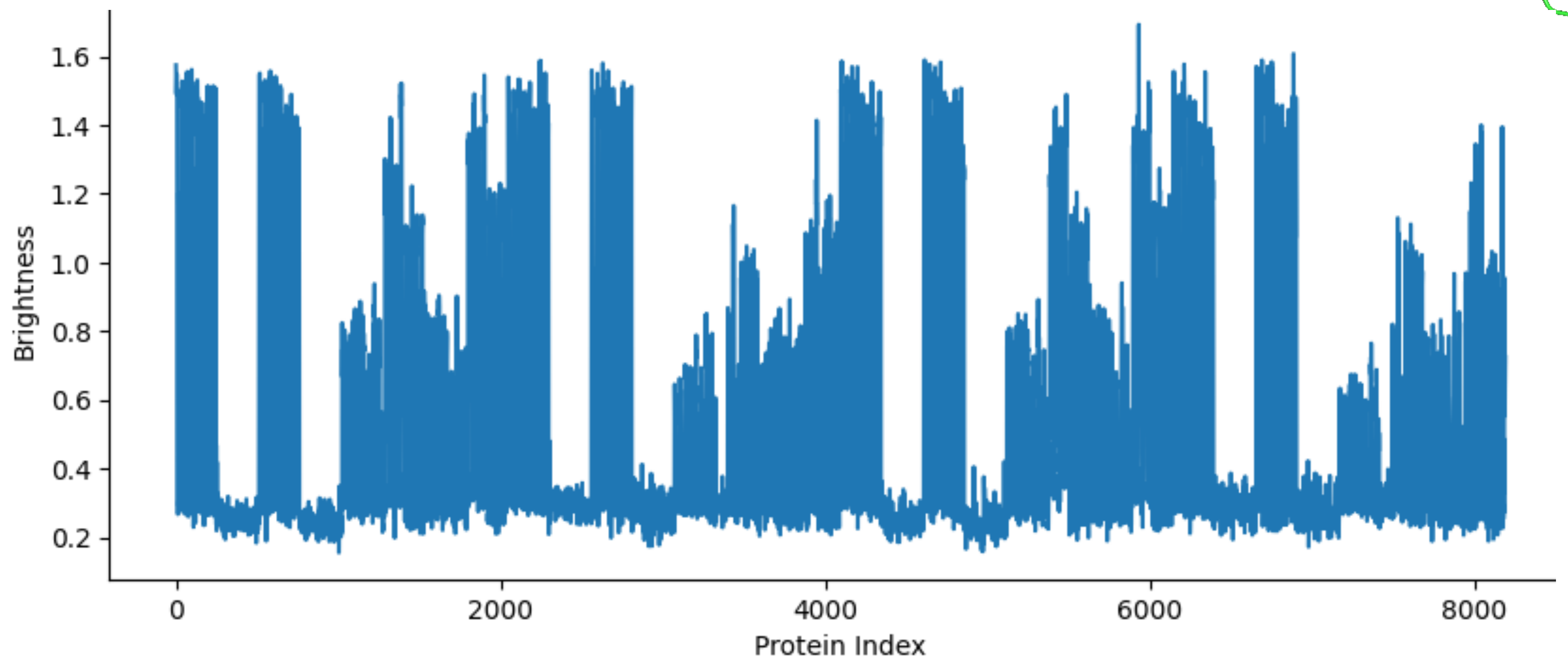
$$f(\mathbf{m}) \xrightarrow{\mathcal{F}} F[\mathbf{k}]$$

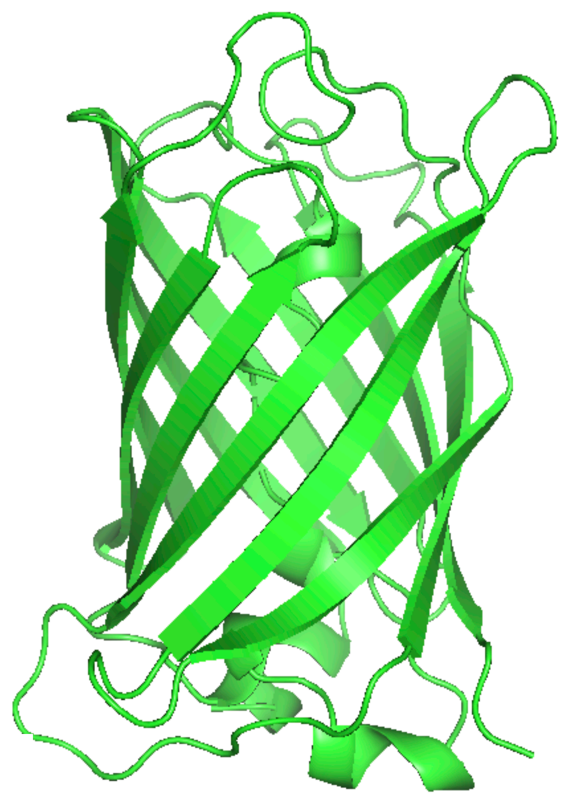


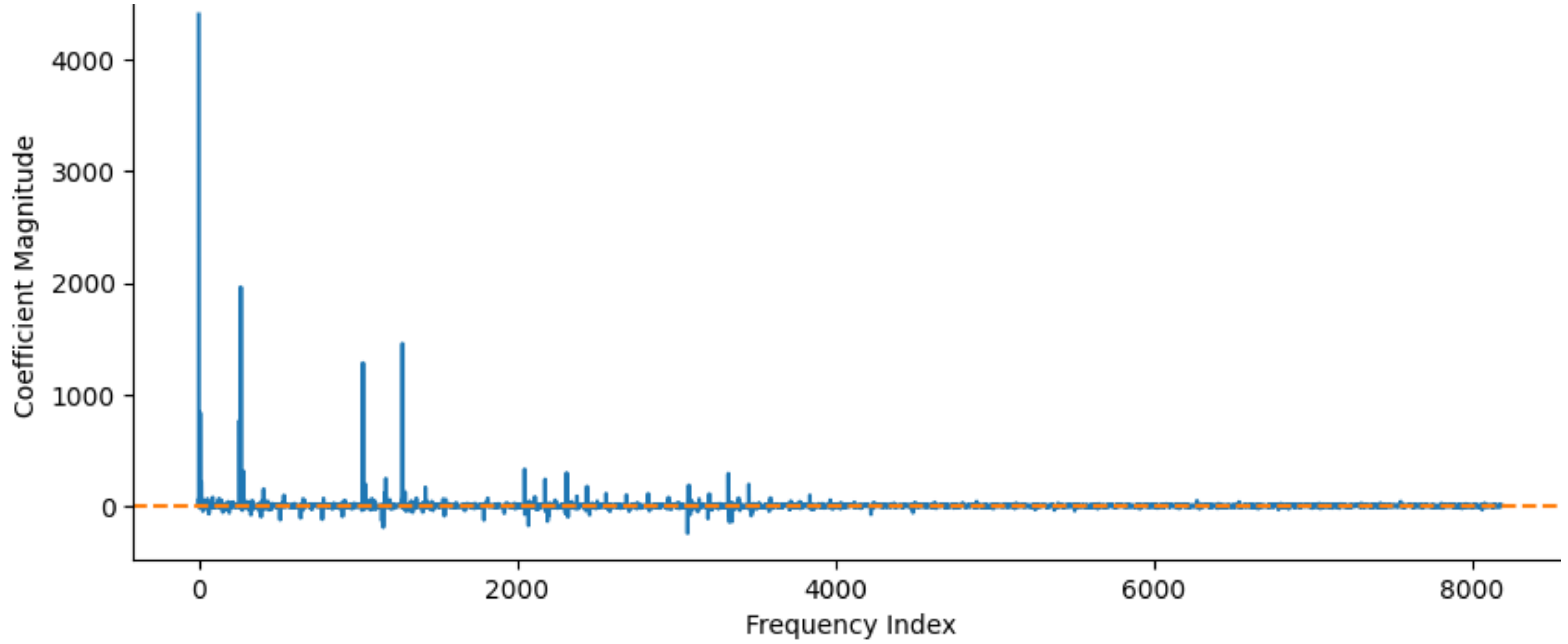


Many functions of interest have *sparse Fourier transforms*

$$f(\mathbf{m}) \xrightarrow{\mathcal{F}} F[\mathbf{k}]$$







Top 40 Fourier coefficients explain **99%** of energy

Previous works

q-sft [1]: We can compute the Fourier transform of an *s*-sparse *q*-ary function with sublinear sample and computational complexity, with vanishing error probability as the function gets larger

$$F[\mathbf{k}] = \frac{1}{q^n} \sum_{\mathbf{m} \in \mathbb{Z}_q^n} f(\mathbf{m}) e^{-\frac{2j\pi}{q}}$$

[1] *Y. E. Erginbas, J. S. Kang, A. Aghazadeh, and K. Ramchandran,"Efficiently computing sparse Fourier transforms of q-ary functions," in 2023 IEEE International Symposium on Information Theory (ISIT), 2023.*

Previous works

q-sft [1]: We can compute the Fourier transform of an *s*-sparse *q*-ary function with sublinear sample and computational complexity, with vanishing error probability as the function gets larger

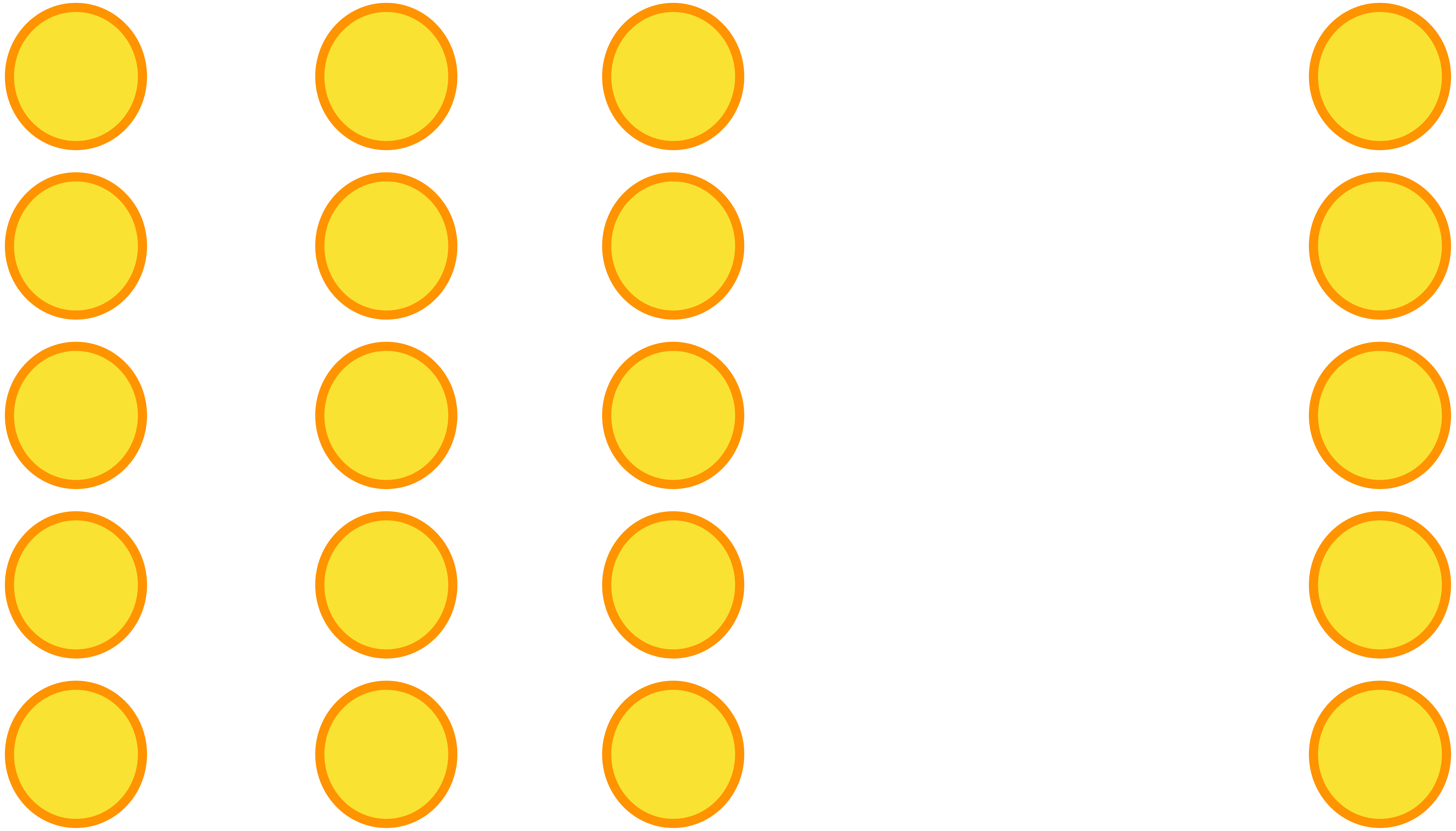
$$F[\mathbf{k}] = \frac{1}{q^n} \sum_{\mathbf{m} \in \mathbb{Z}_q^n} f(\mathbf{m}) e^{-\frac{2j\pi}{q}}$$

q is fixed

[1] *Y. E. Erginbas, J. S. Kang, A. Aghazadeh, and K. Ramchandran,"Efficiently computing sparse Fourier transforms of q-ary functions," in 2023 IEEE International Symposium on Information Theory (ISIT), 2023.*

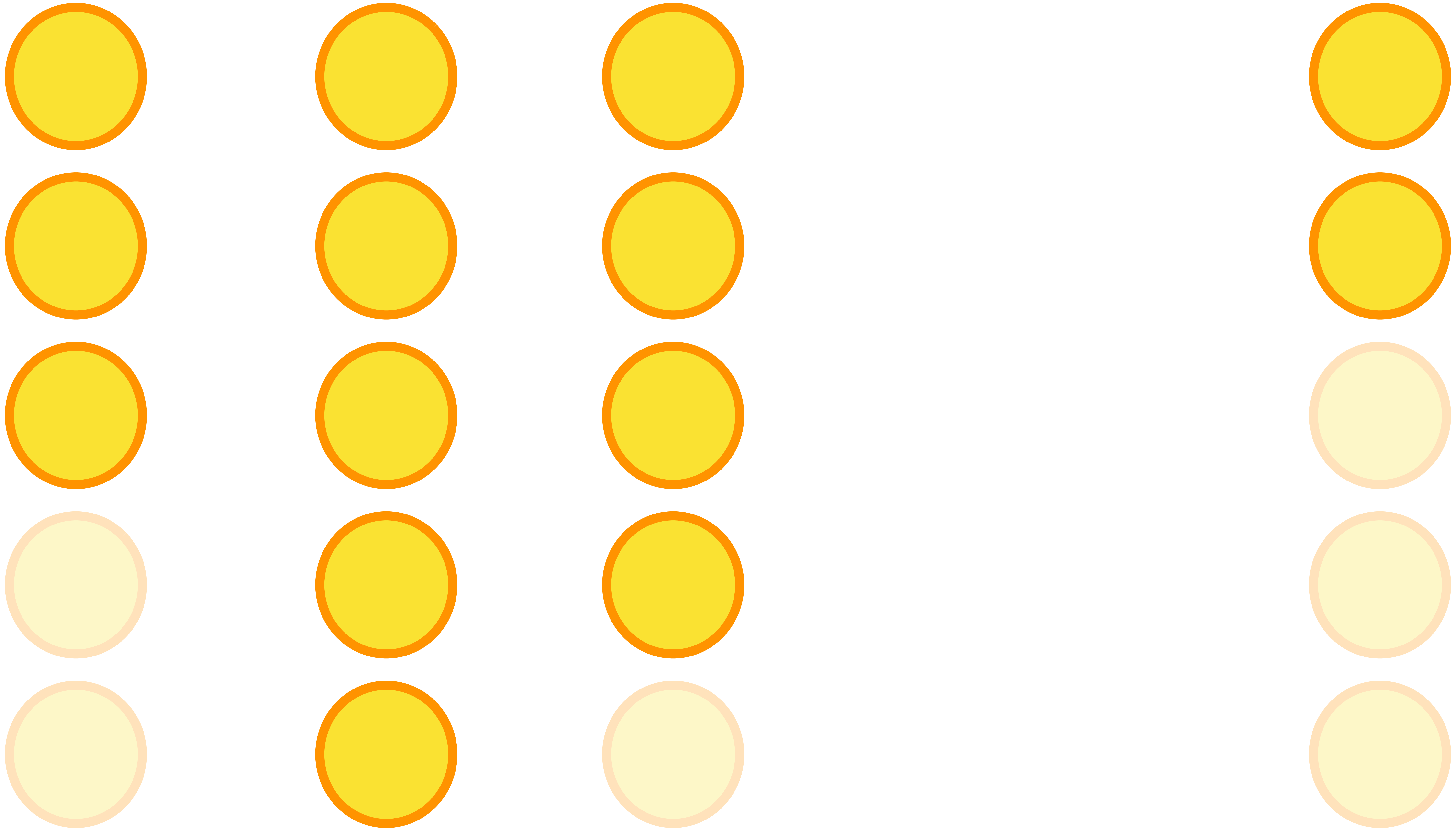
q may vary

$$f(x_1, x_2, x_3, \cdots, x_n) \rightarrow y$$



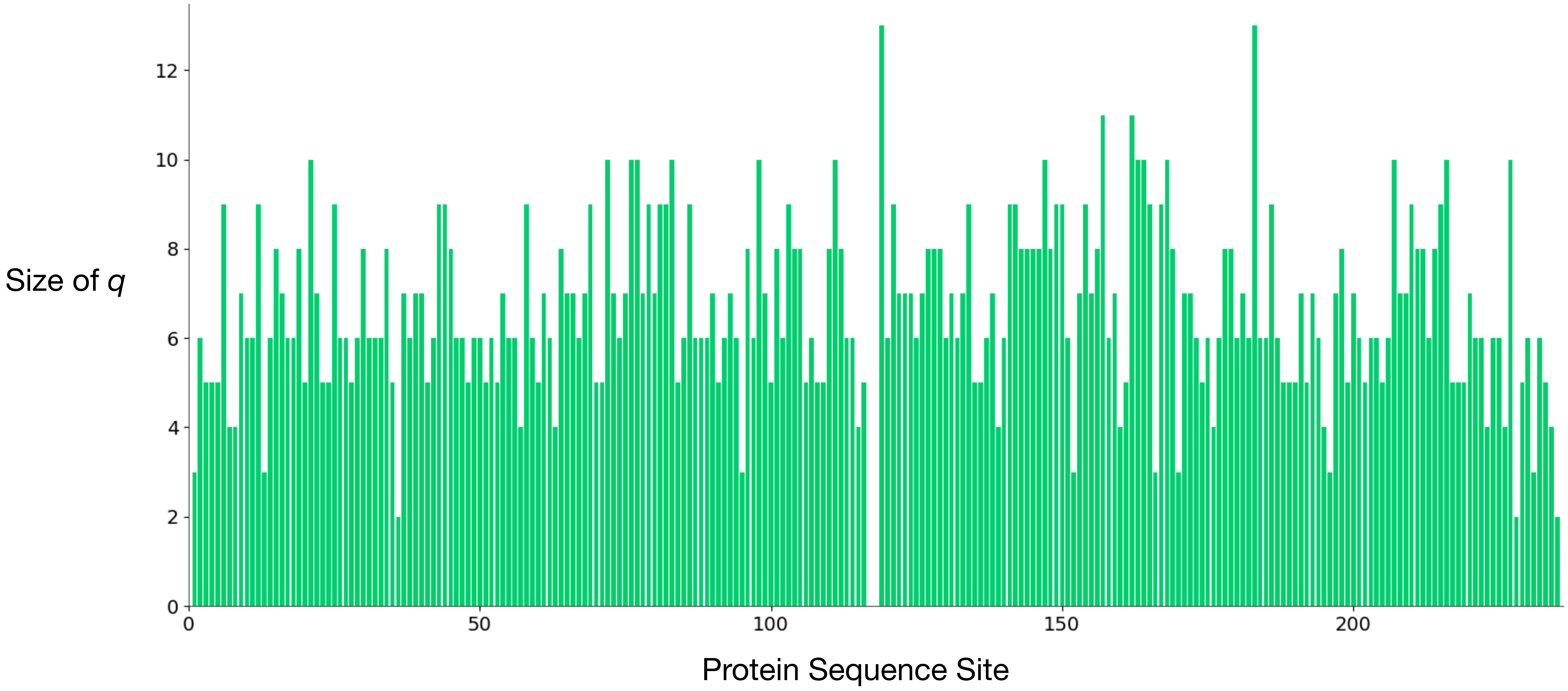
q may vary

$$f(x_1, x_2, x_3, \cdots, x_n) \rightarrow y$$



Generalized *q*-ary functions

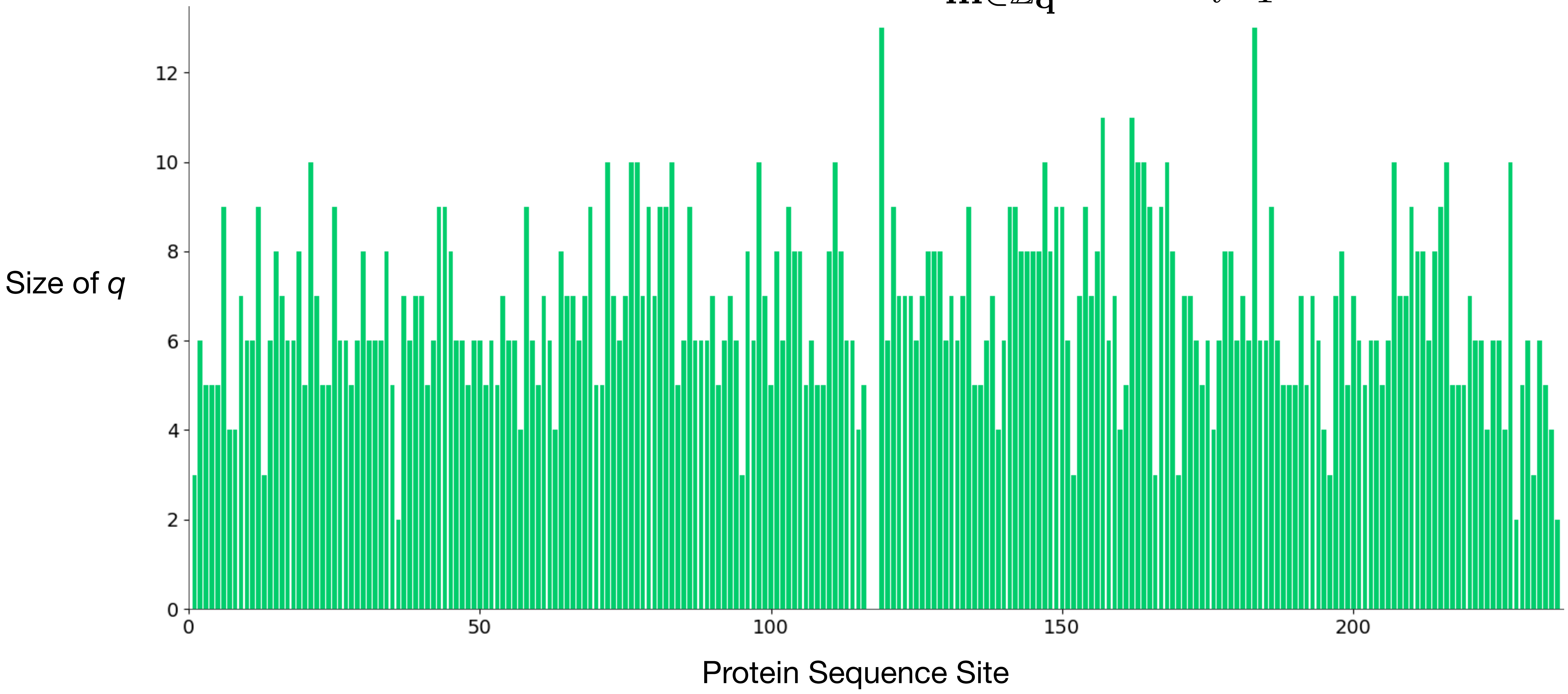
$$\mathbb{Z}_{\mathbf{q}} \triangleq \mathbb{Z}_{q_1} \times \mathbb{Z}_{q_2} \times \cdots \times \mathbb{Z}_{q_n}$$



Generalized *q*-ary functions

$$\mathbb{Z}_{\mathbf{q}} \triangleq \mathbb{Z}_{q_1} \times \mathbb{Z}_{q_2} \times \cdots \times \mathbb{Z}_{q_n}$$

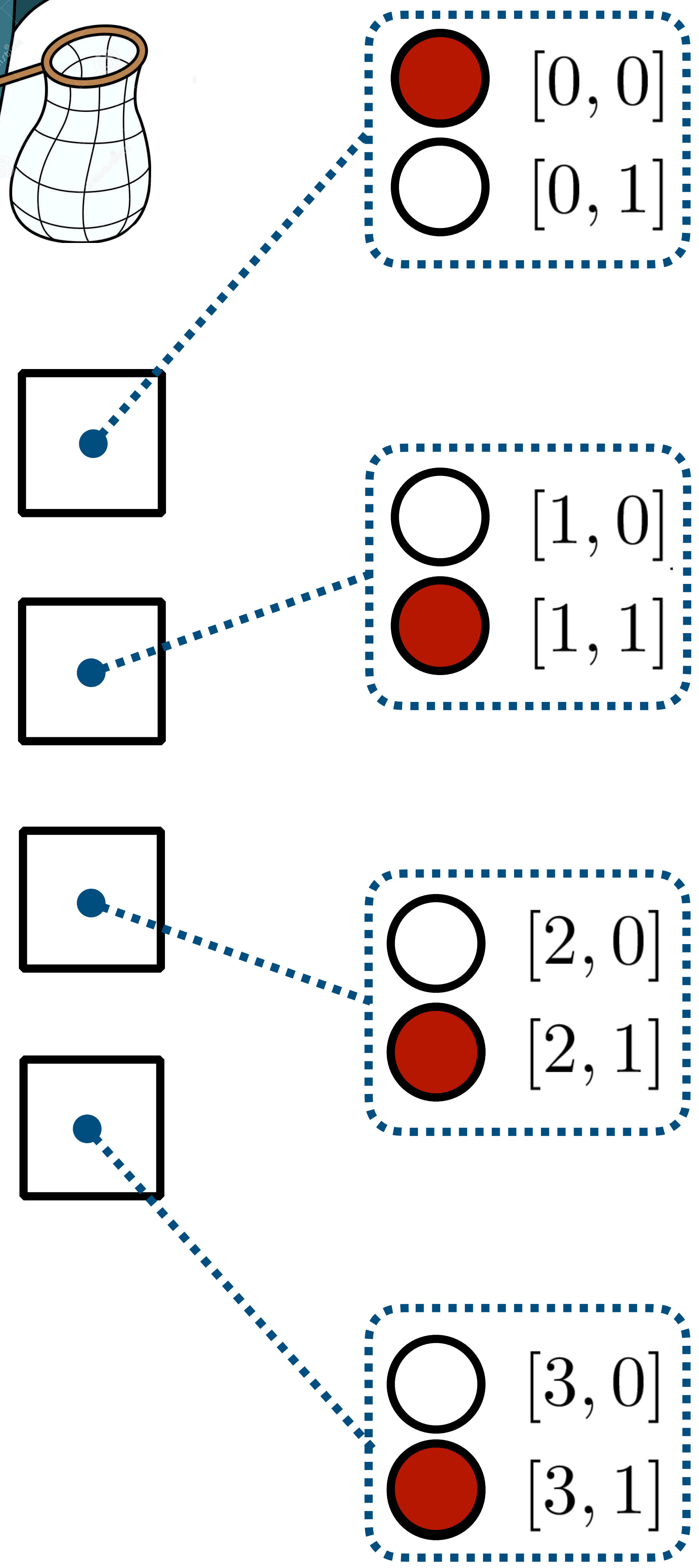
$$F[\mathbf{k}] = \frac{1}{\prod_{i=1}^{n} q_i} \sum_{\mathbf{m} \in \mathbb{Z}_{\mathbf{q}}} f(\mathbf{m}) \prod_{i=1}^{n} e^{-\frac{2j\pi}{q_i}}$$



GFast: Efficiently computing generalized *q*-ary Fourier transforms

Theorem. We can recover a $N = \prod_{i=1}^{n} q_i$ -dimensional generalized *q*-ary function whose Fourier is s-sparse

- with **sample complexity** $O(sn^2)$
- with **computational complexity** $O(sn^2 \log N)$
- assumption: interactions chosen **uniformly** at random
- high probability guarantee + robust to noise



- Key strategy
  - smart subsampling
  - small Fourier transform
  - sparse graph codes

fix one position, enumerate the rest

$$\begin{array}{l} \quad\;\; q_1 \;\; q_2 \\ \mathbf{q} = [4, 2] \\ n = 2 \text{ sites} \\ N = 4 \times 2 = 8 \end{array}$$

$$\underset{\mathbf{y}}{\begin{bmatrix} 6 \\ -2 \\ 0 \\ 4 \\ 2 \\ 2 \\ 0 \\ 4 \end{bmatrix}} = \underbrace{\overset{\mathbf{W}}{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w_2 & 1 & w_2 & 1 & w_2 & 1 & w_2 \\ 1 & 1 & w_4 & w_4 & w_4^2 & w_4^2 & w_4^3 & w_4^3 \\ 1 & w_2 & w_4 & w_4 w_2 & w_4^2 & w_4^2 w_2 & w_4^3 & w_4^3 w_2 \\ 1 & 1 & w_4^2 & w_4^2 & 1 & 1 & w_4^2 & w_4^2 \\ 1 & w_2 & w_4^2 & w_4^2 w_2 & 1 & w_2 & w_4^2 & w_4^2 w_2 \\ 1 & 1 & w_4^3 & w_4^3 & w_4^2 & w_4^2 & w_4 & w_4 \\ 1 & w_2 & w_4^3 & w_4^3 w_2 & w_4^2 & w_4^2 w_2 & w_4 & w_4 w_2 \end{bmatrix}}}_{\text{generalized } q\text{-ary Fourier transform matrix}} \overset{\mathbf{x}}{\begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix}}$$

$$\omega_{q_i} = e^{\frac{2\pi j}{q_i}}$$

$y = f([x_1, x_2])$

x_1	x_2	y
0	0	2
0	1	2
1	0	0
1	1	4
2	0	6
2	1	-2
3	0	0
3	1	4

fix one position, enumerate the rest

$$\begin{array}{c} \quad q_1 \; q_2 \\ \mathbf{q} = [4, 2] \end{array}$$

$n = 2$ sites

$N = 4 \times 2 = 8$

$$\underset{\mathbf{y}_T}{\begin{bmatrix} 6 \\ 0 \\ 2 \\ 0 \end{bmatrix}} = \underset{\mathbf{W}_T}{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & w_4 & w_4 & w_4^2 & w_4^2 & w_4^3 & w_4^3 \\ 1 & 1 & w_4^2 & w_4^2 & 1 & 1 & w_4^2 & w_4^2 \\ 1 & 1 & w_4^3 & w_4^3 & w_4^2 & w_4^2 & w_4 & w_4 \end{bmatrix}} \underset{\mathbf{x}}{\begin{bmatrix} \color{red}{2} \\ 0 \\ 0 \\ \color{red}{1} \\ 0 \\ \color{red}{2} \\ 0 \\ \color{red}{1} \end{bmatrix}}$$

regularly subsampled FT matrix

$y = f([x_1, x_2])$

x_1	x_2	y
0	0	2
0	1	2
1	0	0
1	1	4
2	0	6
2	1	-2
3	0	0
3	1	4

take small Fourier transform

$$\begin{array}{cc} q_1 & q_2 \end{array}$$
$$\mathbf{q} = [4, 2]$$
$$n = 2 \text{ sites}$$
$$N = 4 \times 2 = 8$$

$$\mathbf{y}_T \quad \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & w_4 & w_4^2 & w_4^3 \\ 1 & w_4^2 & 1 & w_4^2 \\ 1 & w_4^3 & w_4^2 & w_4 \end{bmatrix} \times \quad \mathbf{W}_T \quad \mathbf{x}$$

$$\begin{bmatrix} 6 \\ 0 \\ 2 \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & w_4 & w_4 & w_4^2 & w_4^2 & w_4^3 & w_4^3 \\ 1 & 1 & w_4^2 & w_4^2 & 1 & 1 & w_4^2 & w_4^2 \\ 1 & 1 & w_4^3 & w_4^3 & w_4^2 & w_4^2 & w_4 & w_4 \end{bmatrix}}_{\textbf{regularly}\text{ subsampled FT matrix}} \begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

$y = f([x_1, x_2])$

x_1	x_2	y
0	0	2
0	1	2
1	0	0
1	1	4
2	0	6
2	1	-2
3	0	0
3	1	4

take small Fourier transform

$$\begin{array}{cc} & q_1 \;\; q_2 \\ \mathbf{q} = & [4, 2] \end{array}$$

$n = 2$ sites

$N = 4 \times 2 = 8$

$$\frac{1}{4}\mathbf{W}_{4\times 4}\mathbf{y}_T$$

$$\downarrow$$

$$\begin{bmatrix} 2 \\ 1 \\ 2 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\text{induced sparse graph}} \overset{\mathbf{x}}{\begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix}}$$

$y = f([x_1, x_2])$

x_1	x_2	y
0	0	2
0	1	2
1	0	0
1	1	4
2	0	6
2	1	-2
3	0	0
3	1	4

take small Fourier transform

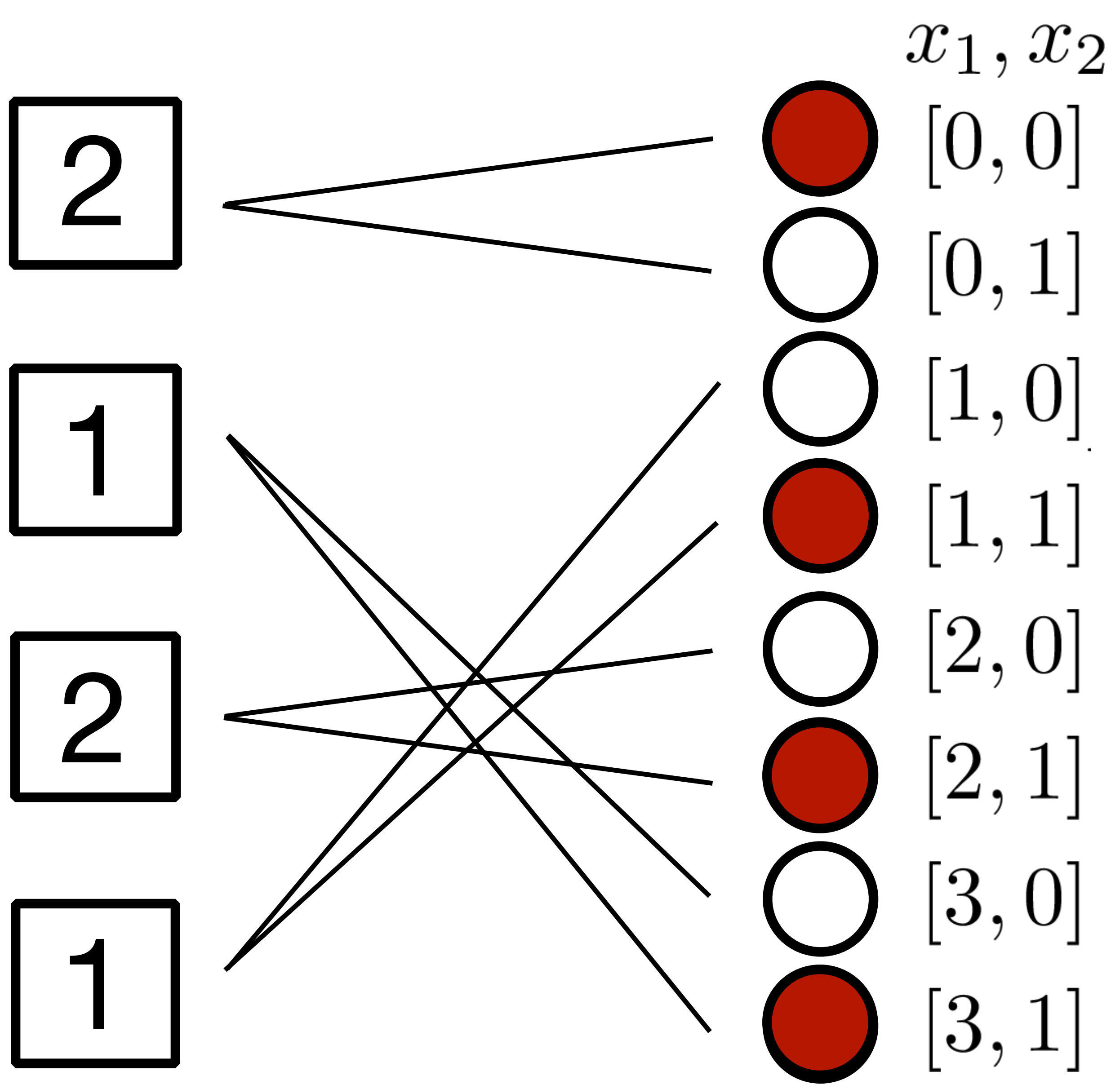
Fourier

transformed samples

$$\frac{1}{4}\mathbf{W}_{4\times 4}\mathbf{y}_T$$

$$\downarrow$$

$$\begin{bmatrix}2\\1\\2\\1\end{bmatrix} = \underbrace{\begin{bmatrix}1&1&0&0&0&0&0&0\\0&0&0&0&0&0&1&1\\0&0&0&0&1&1&0&0\\0&0&1&1&0&0&0&0\end{bmatrix}}_{\text{induced sparse graph}} \overset{\mathbf{x}}{\begin{bmatrix}\color{red}{2}\\0\\0\\\color{red}{1}\\0\\\color{red}{2}\\0\\\color{red}{1}\end{bmatrix}} \longrightarrow$$



<u>sparse bipartite graph</u>
#edges ~ #nodes

new set of samples

$$\mathbf{q} = [\overset{q_1}{4}, \overset{q_2}{2}]$$

$$n = 2 \text{ sites}$$

$$N = 4 \times 2 = 8$$

$$\underset{\mathbf{y}}{\begin{bmatrix} 6 \\ -2 \\ 0 \\ 4 \\ 2 \\ 2 \\ 0 \\ 4 \end{bmatrix}} = \underbrace{\overset{\mathbf{W}}{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w_2 & 1 & w_2 & 1 & w_2 & 1 & w_2 \\ 1 & 1 & w_4 & w_4 & w_4^2 & w_4^2 & w_4^3 & w_4^3 \\ 1 & w_2 & w_4 & w_4 w_2 & w_4^2 & w_4^2 w_2 & w_4^3 & w_4^3 w_2 \\ 1 & 1 & w_4^2 & w_4^2 & 1 & 1 & w_4^2 & w_4^2 \\ 1 & w_2 & w_4^2 & w_4^2 w_2 & 1 & w_2 & w_4^2 & w_4^2 w_2 \\ 1 & 1 & w_4^3 & w_4^3 & w_4^2 & w_4^2 & w_4 & w_4 \\ 1 & w_2 & w_4^3 & w_4^3 w_2 & w_4^2 & w_4^2 w_2 & w_4 & w_4 w_2 \end{bmatrix}}}_{\text{generalized } q\text{-ary Fourier transform matrix}} \underset{\mathbf{x}}{\begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix}}$$

$y = f([x_1, x_2])$

x_1	x_2	y
0	0	2
0	1	2
1	0	0
1	1	4
2	0	6
2	1	-2
3	0	0
3	1	4

new set of position patterns

$$q_1 \;\; q_2$$
$$\mathbf{q} = [4, 2]$$
$$n = 2 \text{ sites}$$
$$N = 4 \times 2 = 8$$

$$\mathbf{y}_T = \mathbf{W}_T \, \mathbf{x}$$

$$\begin{bmatrix} 6 \\ -2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w_2 & 1 & w_2 & 1 & w_2 & 1 & w_2 \end{bmatrix}}_{\text{regularly subsampled FT matrix}} \begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

regularly subsampled FT matrix

$y = f([x_1, x_2])$

x_1	x_2	y
0	0	2
0	1	2
1	0	0
1	1	4
2	0	6
2	1	-2
3	0	0
3	1	4

take small Fourier transform

$$\begin{array}{cc} q_1 & q_2 \end{array}$$

$$\mathbf{q} = [4, 2]$$

$$n = 2 \text{ sites}$$

$$N = 4 \times 2 = 8$$

$$\mathbf{y}_T \qquad \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & w_2 \end{bmatrix} \times \qquad \mathbf{W}_T \qquad \mathbf{x}$$

$$\begin{bmatrix} 6 \\ -2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w_2 & 1 & w_2 & 1 & w_2 & 1 & w_2 \end{bmatrix}}_{} \begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

regularly subsampled FT matrix

$y = f([x_1, x_2])$

x_1	x_2	y
0	0	2
0	1	2
1	0	0
1	1	4
2	0	6
2	1	-2
3	0	0
3	1	4

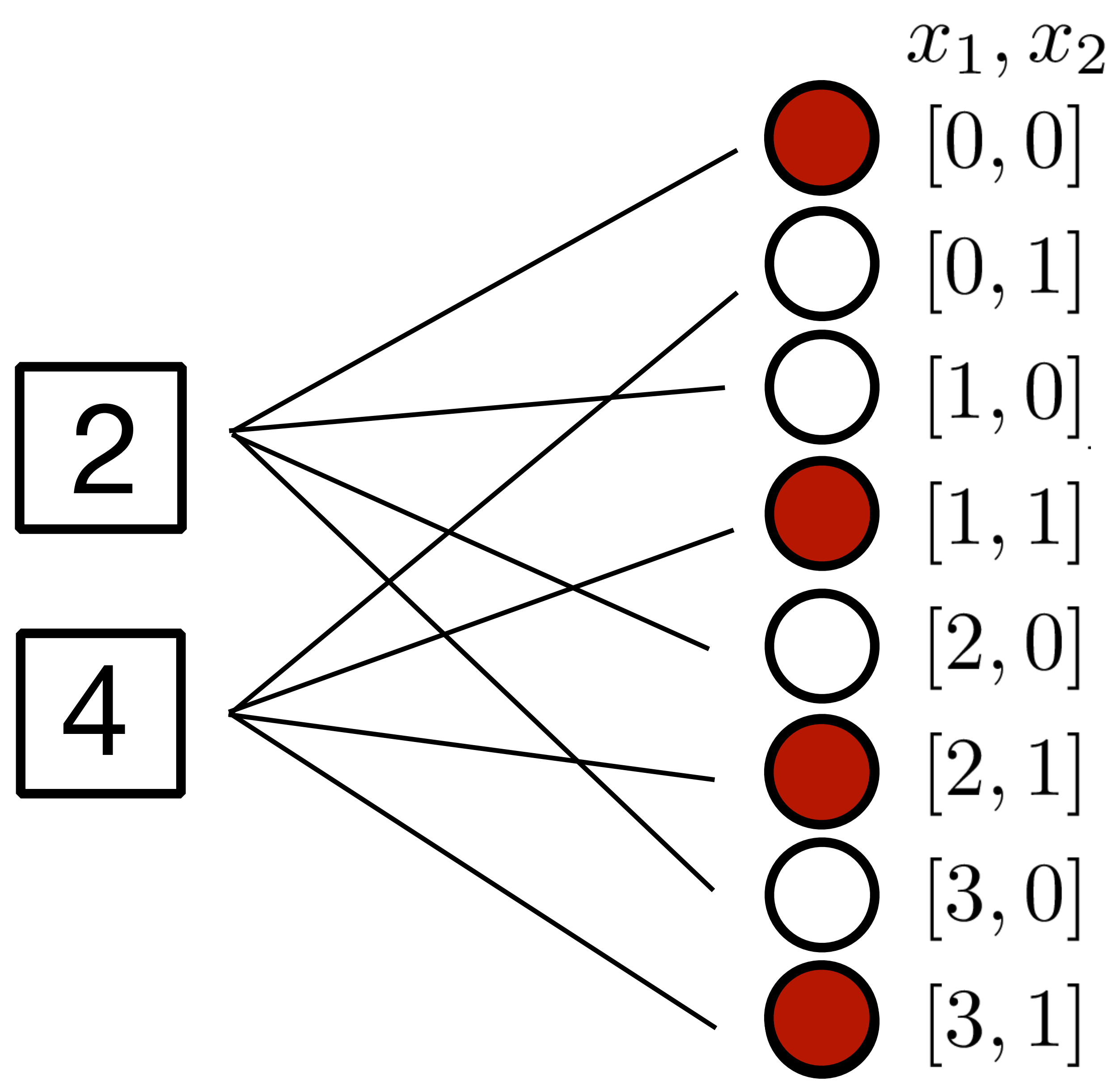
take small Fourier transform

Fourier

transformed samples

$$\frac{1}{2}\mathbf{W}_{2\times 2}\mathbf{y}_T \rightarrow \begin{bmatrix}2\\4\end{bmatrix} = \underbrace{\mathbf{W}_T}_{} \mathbf{x}$$

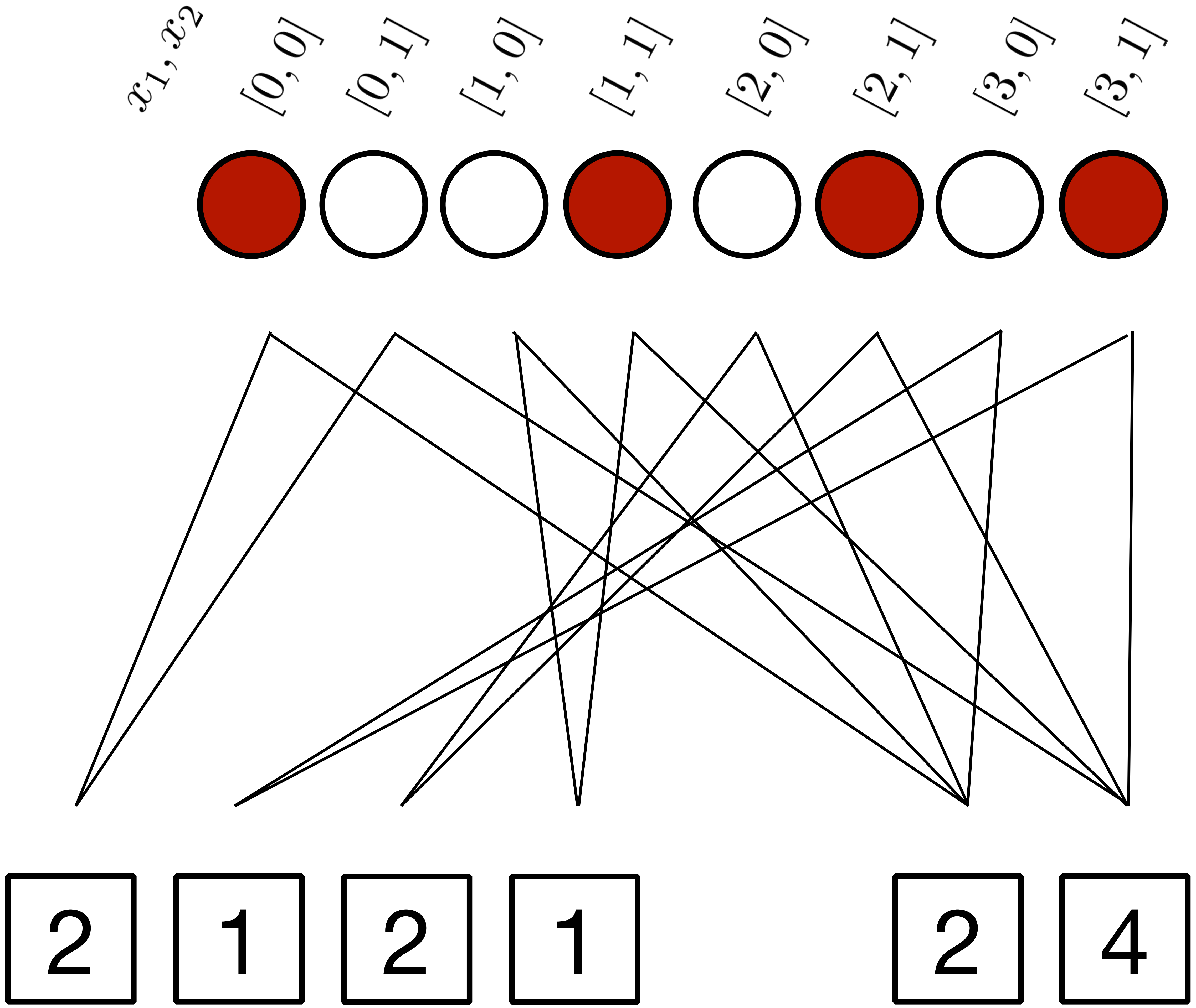
$$\begin{bmatrix}2\\4\end{bmatrix} = \underbrace{\begin{bmatrix}1 & 0 & 1 & 0 & 1 & 0 & 1 & 0\\0 & 1 & 0 & 1 & 0 & 1 & 0 & 1\end{bmatrix}}_{\text{induced sparse graph}} \begin{bmatrix}\color{red}{2}\\0\\0\\\color{red}{1}\\0\\\color{red}{2}\\0\\\color{red}{1}\end{bmatrix}$$



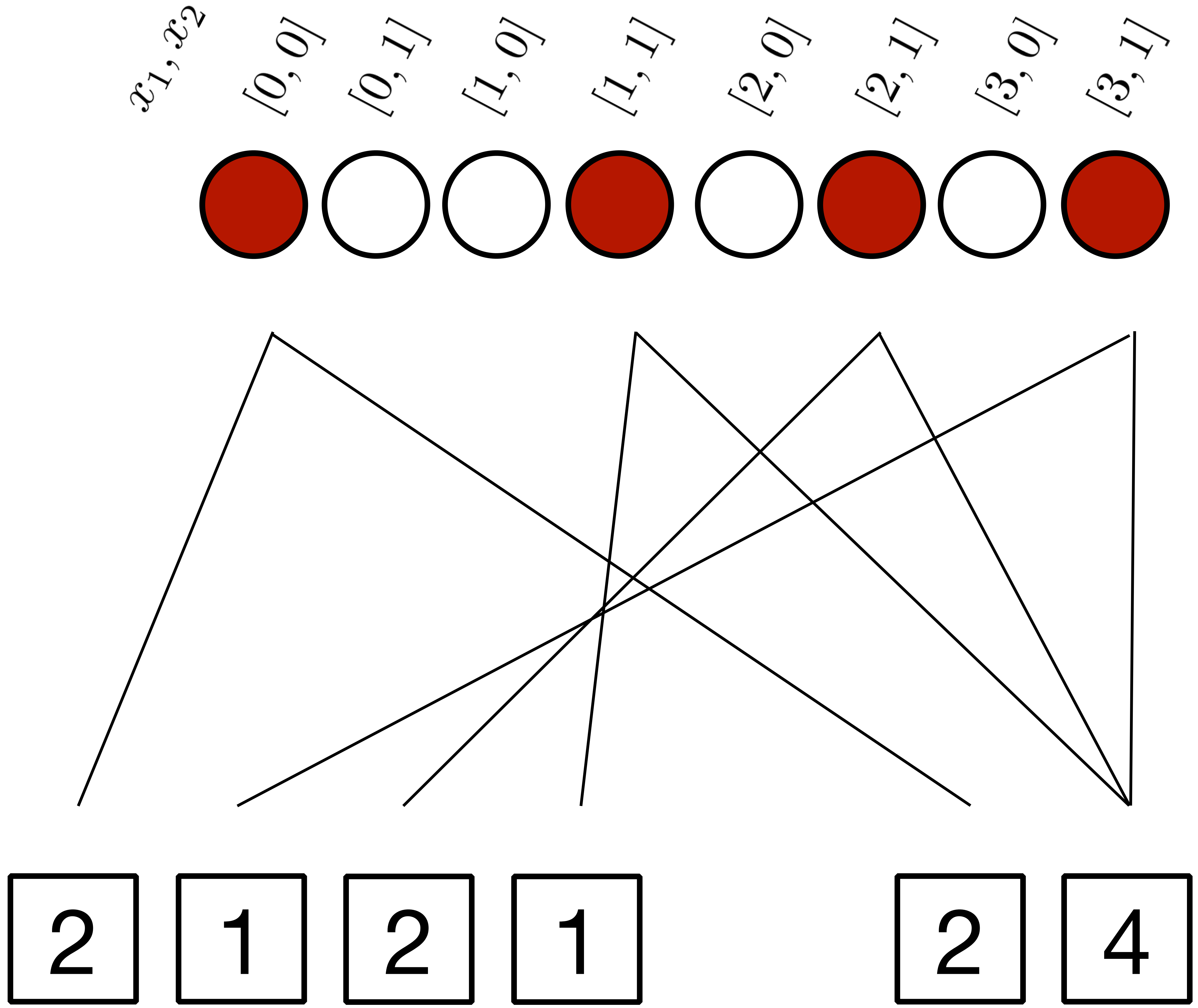
sparse bipartite graph

#edges ~ #nodes

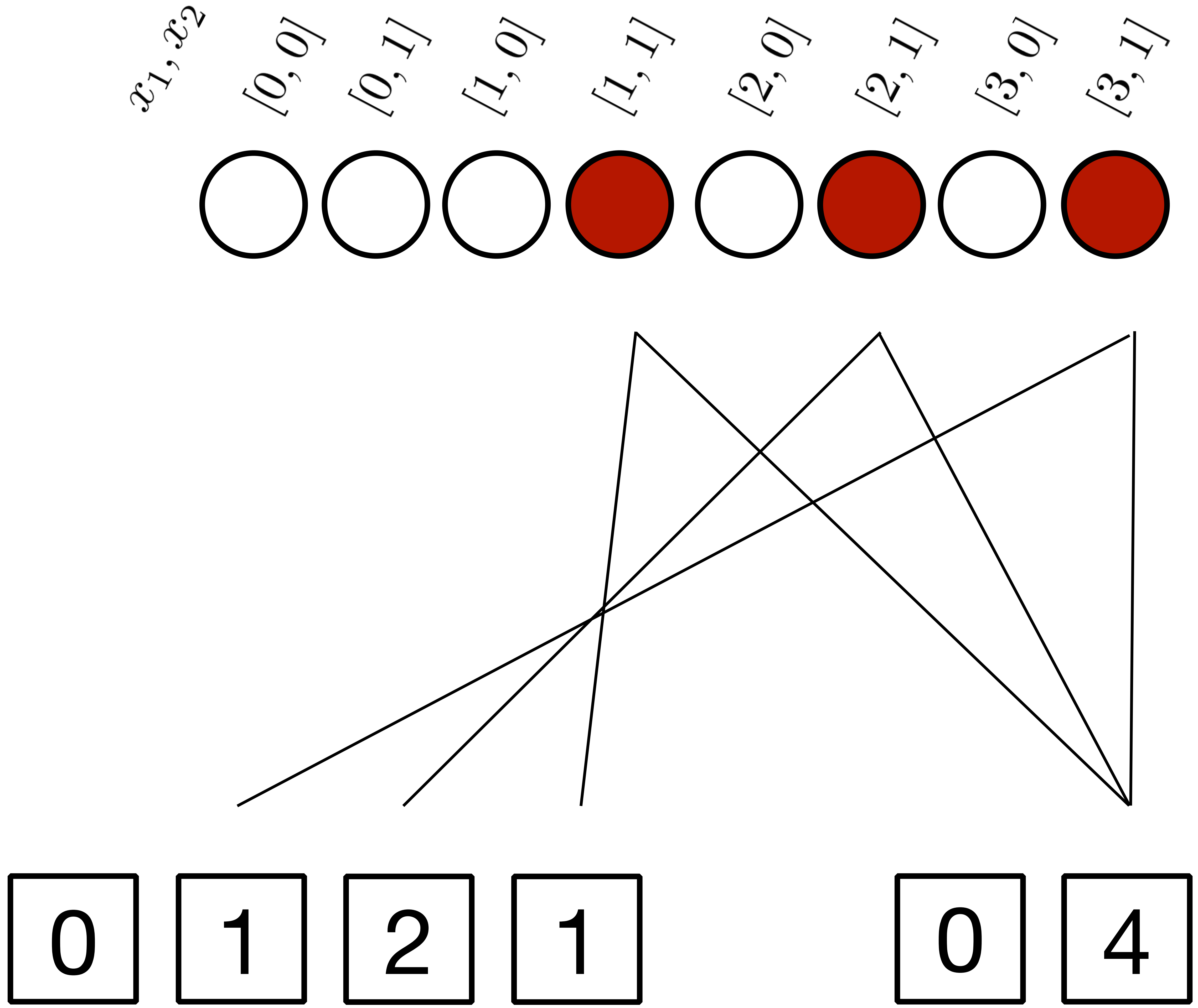
Peel interactions one by one



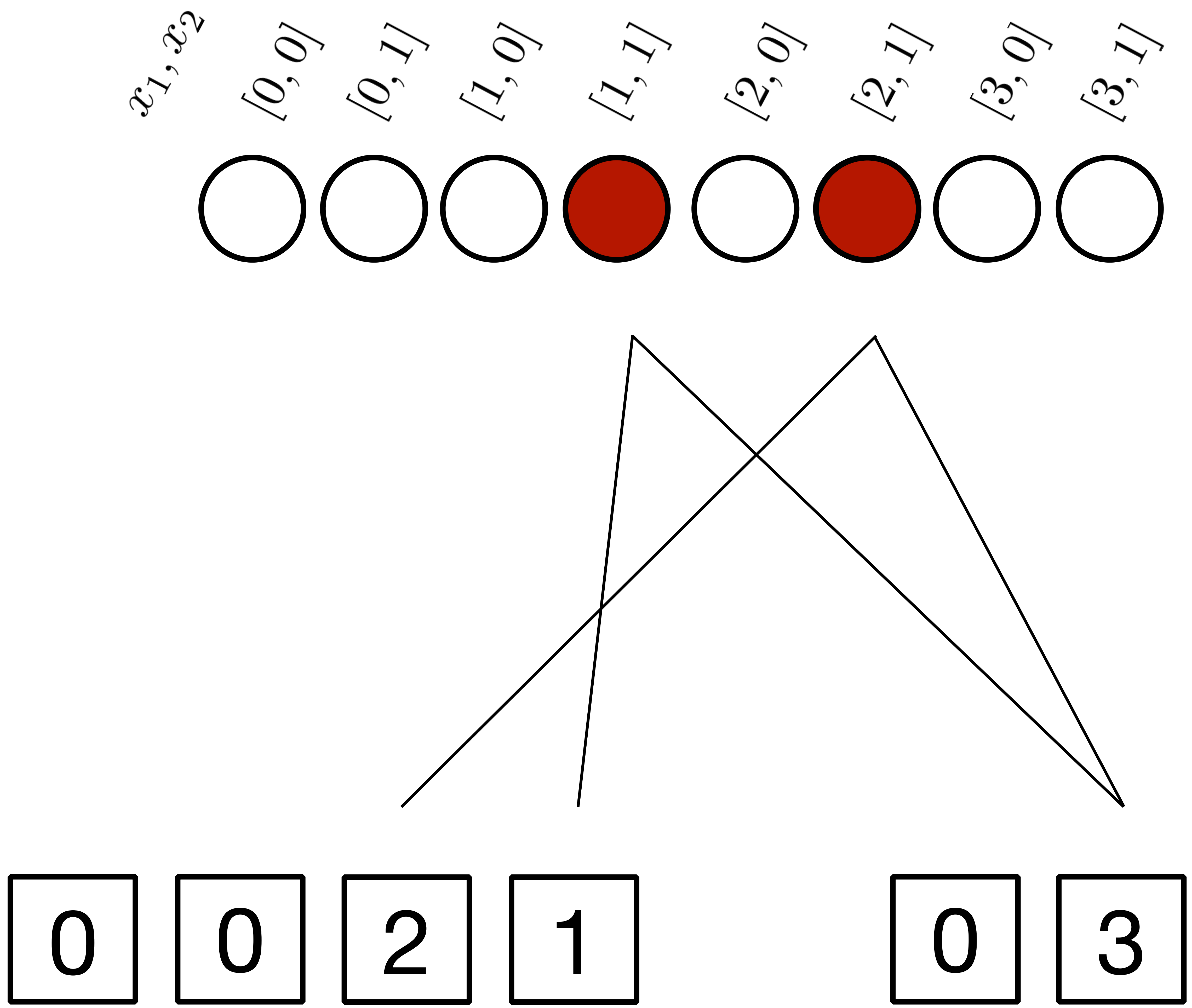
Peel interactions one by one



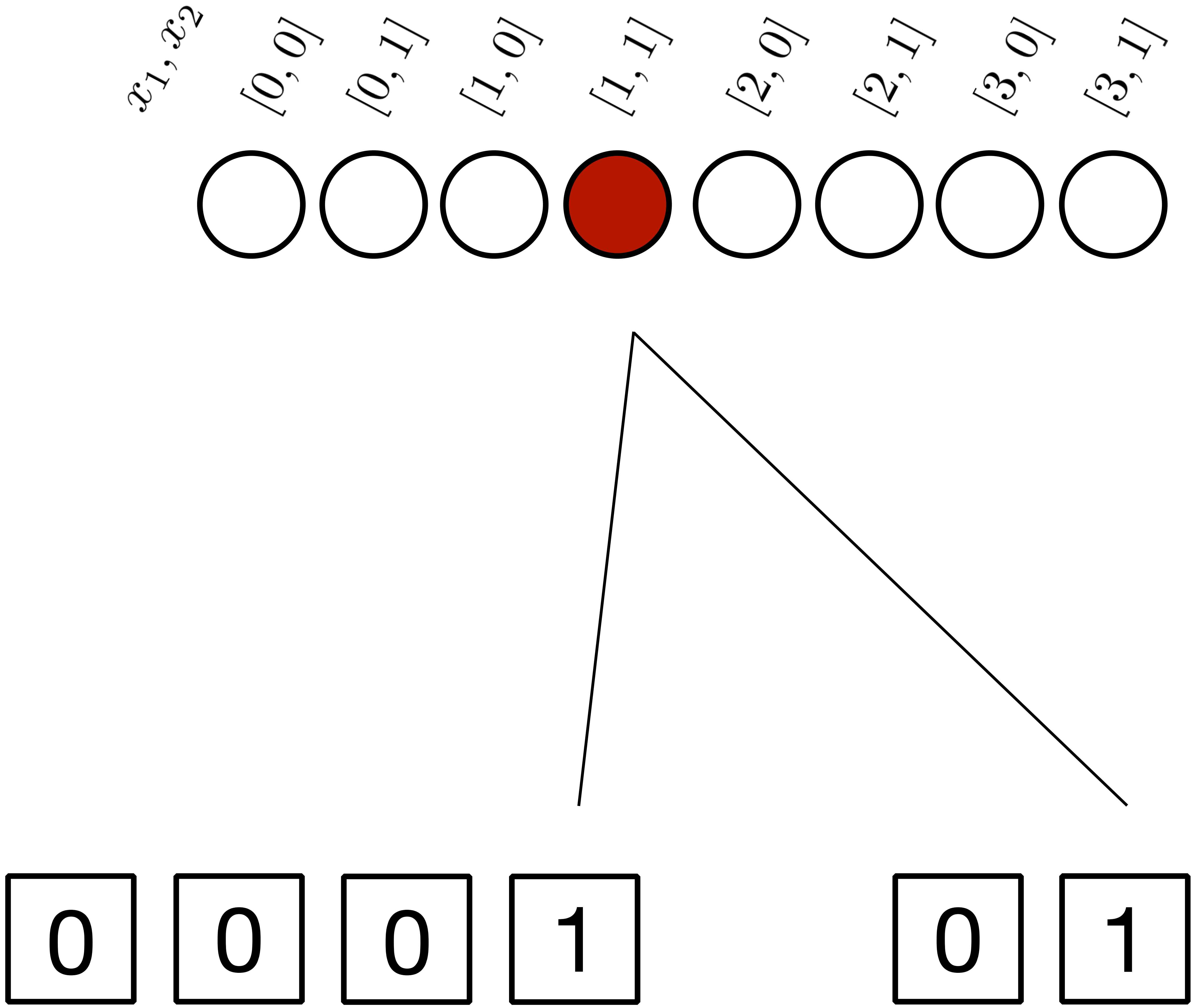
Peel interactions one by one



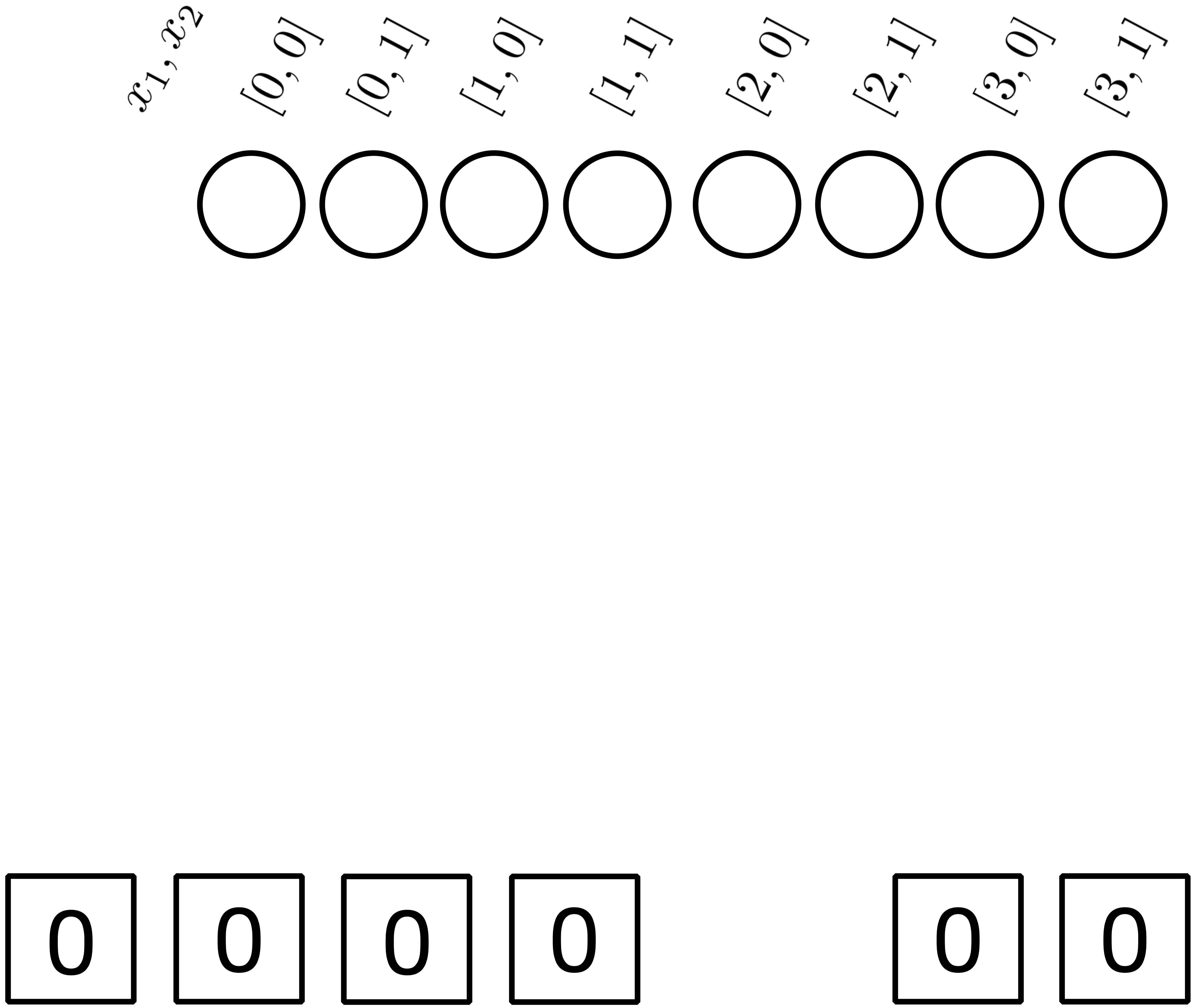
Peel interactions one by one



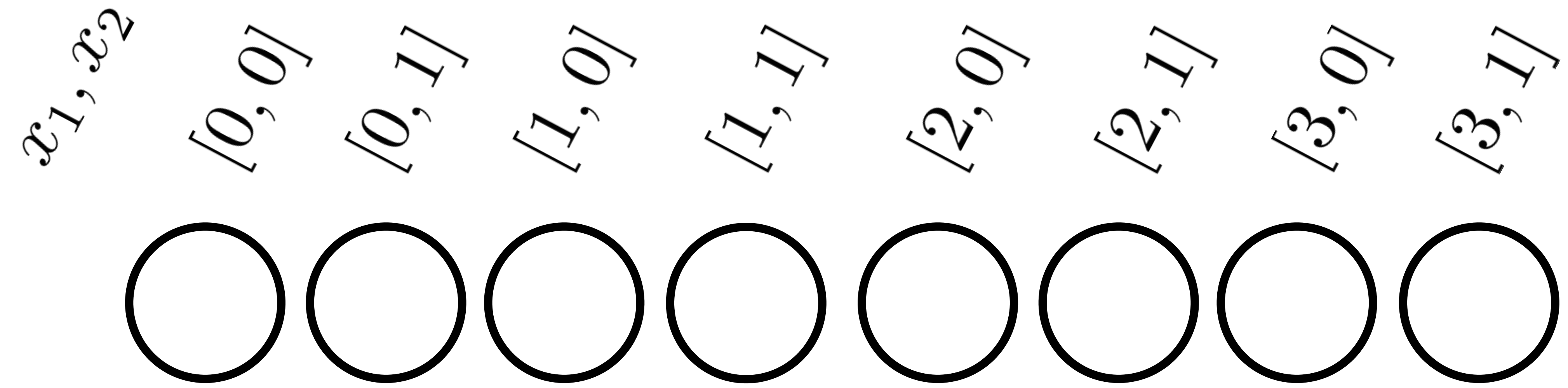
Peel interactions one by one



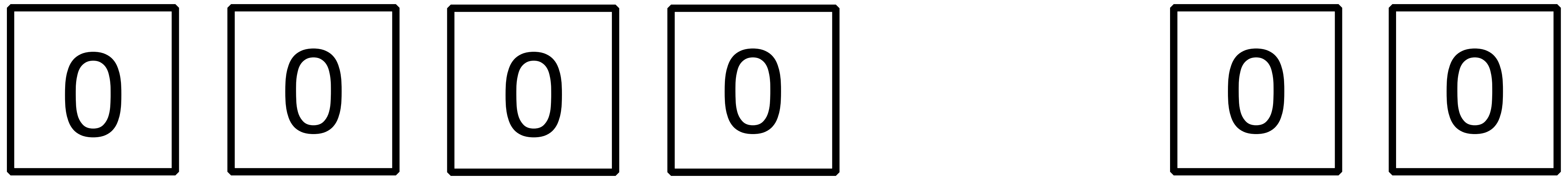
Peel interactions one by one



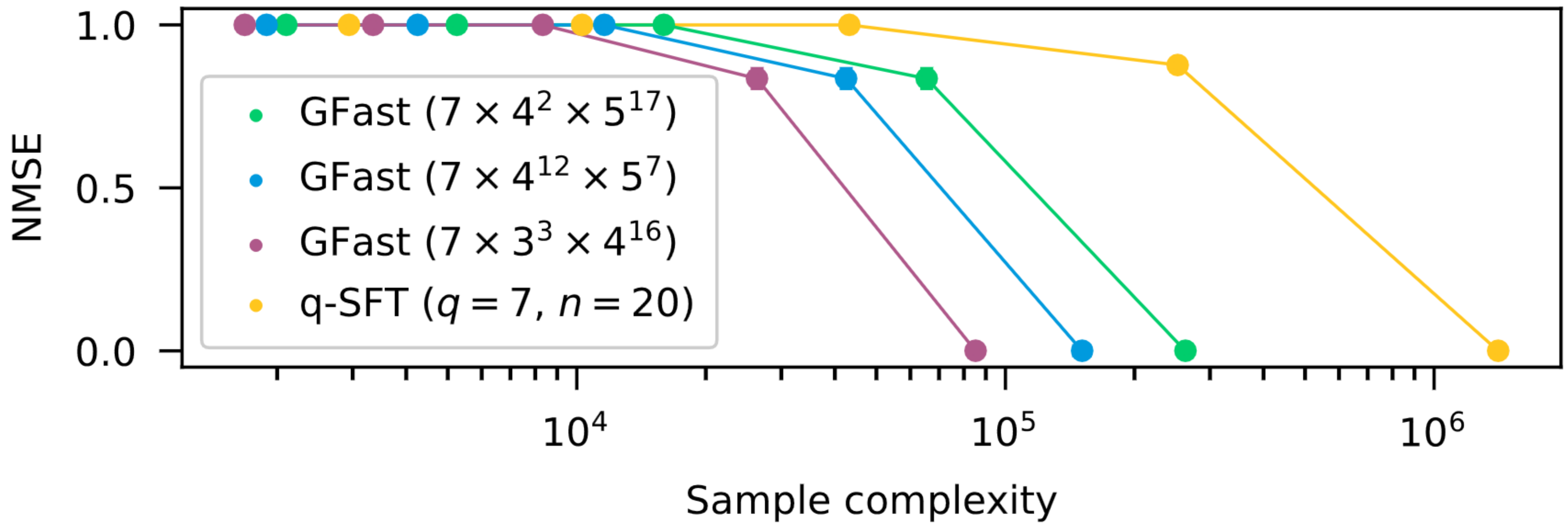
Peel interactions one by one



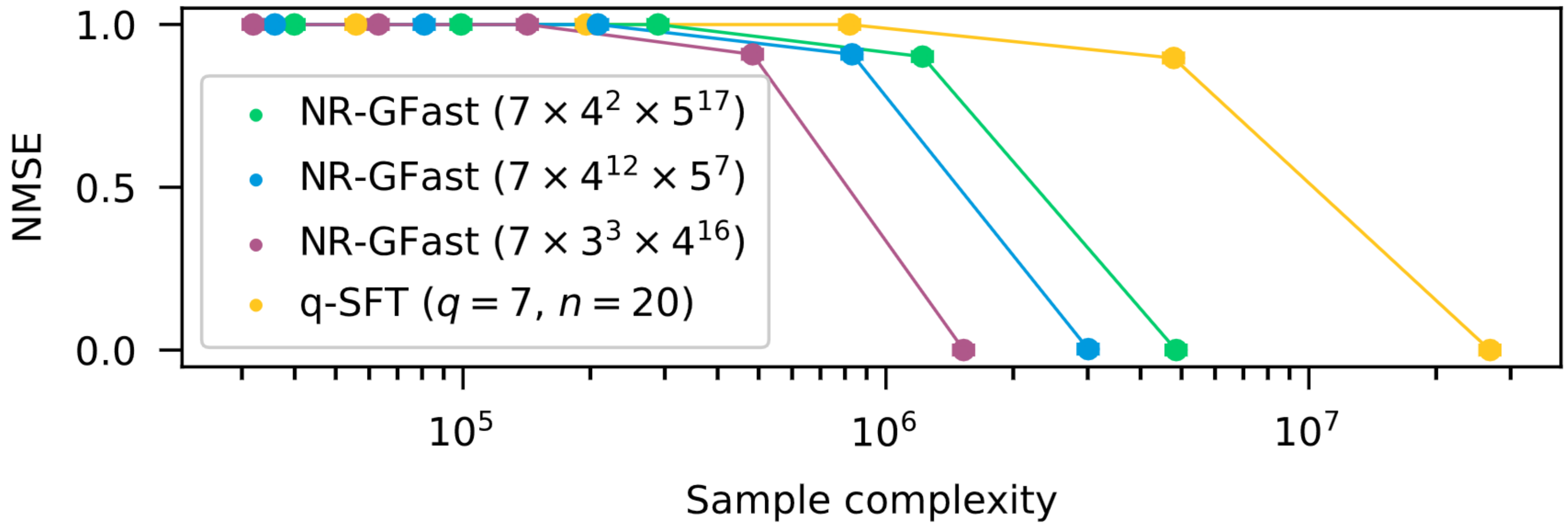
- High probability guarantees using density evolution
- Each small transform group has check node degrees that are Poisson distributed



We can save a lot if you know a bit about the model

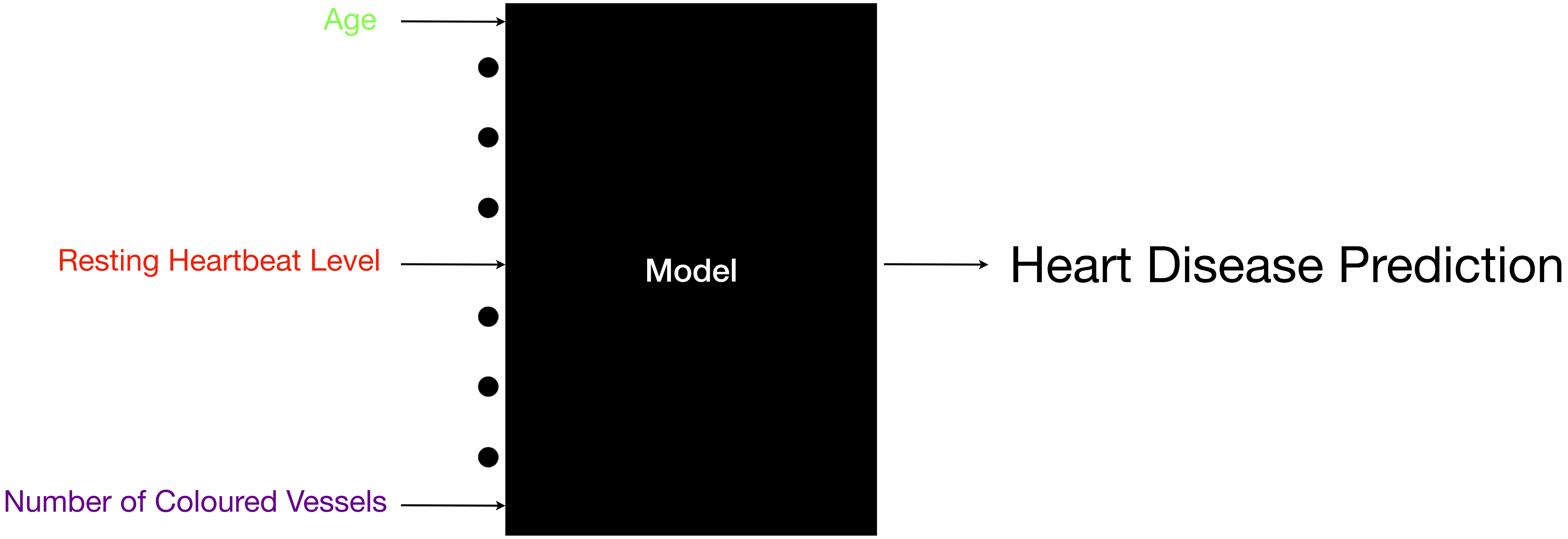


Noise-less



Noise-robust

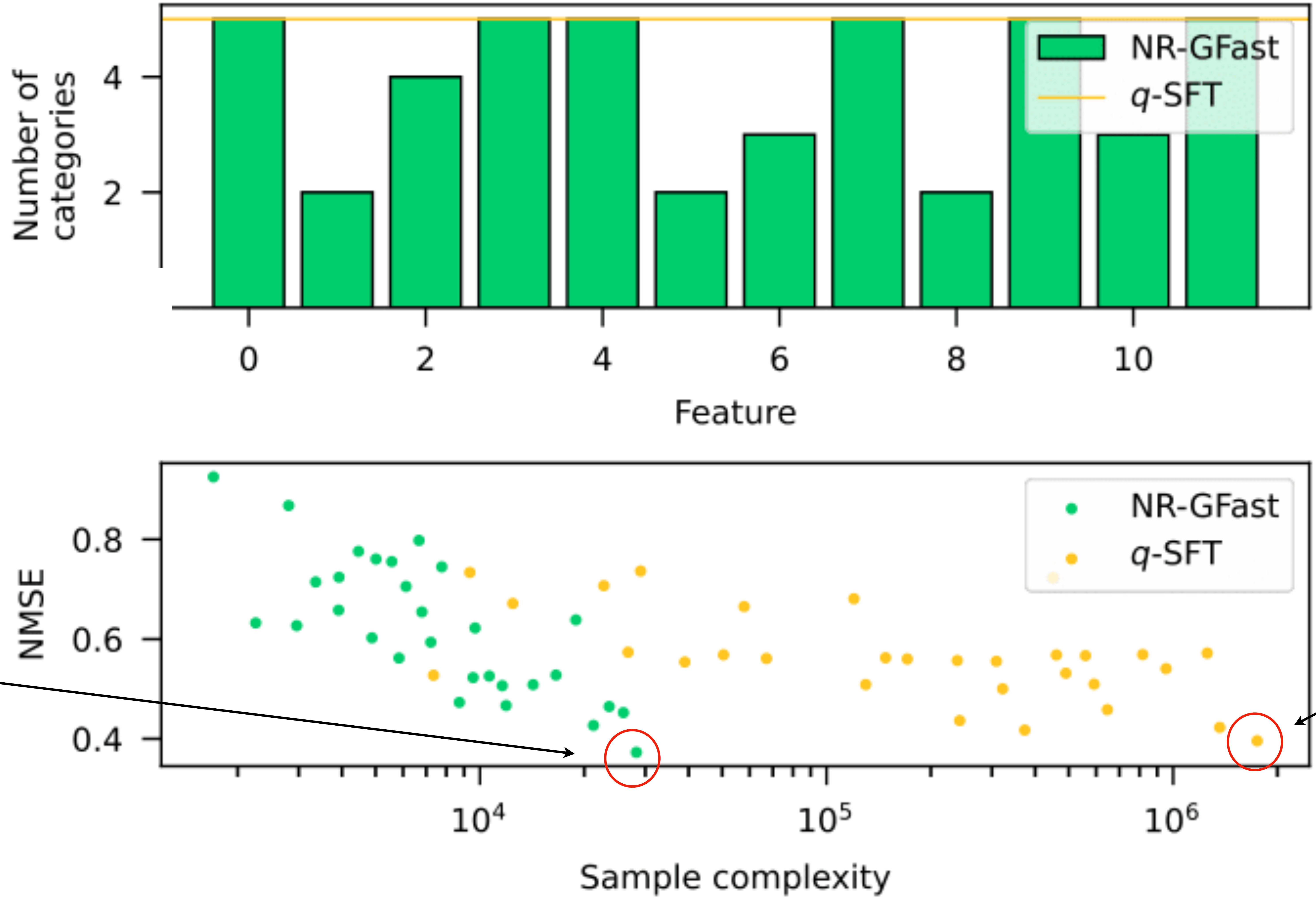
Machine learning example



$$\mathbb{Z}_5 \times \mathbb{Z}_2 \times \mathbb{Z}_4 \times \mathbb{Z}_5 \times \mathbb{Z}_5 \times \mathbb{Z}_2 \times \mathbb{Z}_3 \times \mathbb{Z}_5 \times \mathbb{Z}_2 \times \mathbb{Z}_5 \times \mathbb{Z}_3 \times \mathbb{Z}_5$$

q-ary: $\mathbb{Z}_5^{12}$

Treating the function as *q*-ary requires *61x* more samples to achieve similar accuracy



28,000 samples

1.75 million samples

Thanks!

arXiv

Github