Sample-optimal average-case sparse Fourier Transform in two dimensions

Haitham Hassanieh

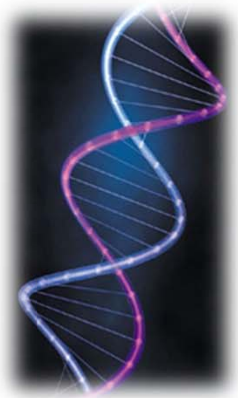
Joint work with Badih Ghazi , Piotr Indyk , Dina Katabi ,
Eric Price, & Lixin Shi

Massachusetts
Institute of
Technology

wireless
@mit

The Discrete Fourier Transform

Video / Audio



DNA

Medical Imagining

Astronomy

Given: A signal $x(t)$ $0 \leq t < n$

Goal: Compute the frequency representation $\hat{x}(f)$

$$\hat{x}(f) = \sum_{t=0}^{n} x(t) e^{-j\frac{2\pi f t}{N}}$$

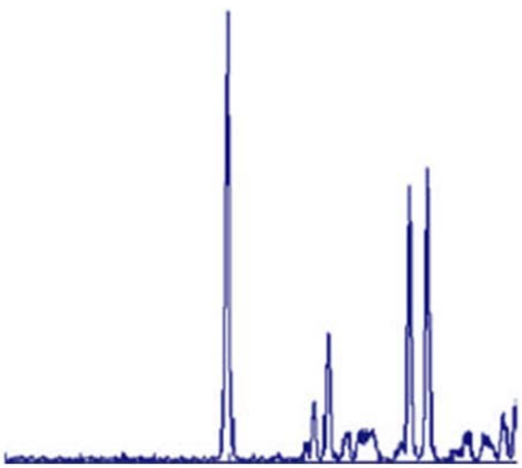
Computing the DFT

- Fast Fourier Transform (1965): FFT $\rightarrow$ $O(n \log n)$

Can we do better? Sub-linear time?

Leverage Sparsity

- Compute only the few large frequencies
- Sparsity appears in video, audio, telescope/satellite data, genomics …



The Sparse Fourier Transform

- For signals of length n, compute the k “large” frequencies
- Sub-linear algorithm:**

Algorithm	Time	Samples		Lower bound
Exactly sparse	$O(k \log n)$	$O(k \log n)$	$\neq$	$O(k)$
Approximately sparse	$O(k \log(n) \log(n/k))$	$O(k \log(n) \log(n/k))$	$\neq$	$O(k \log(n/k))$

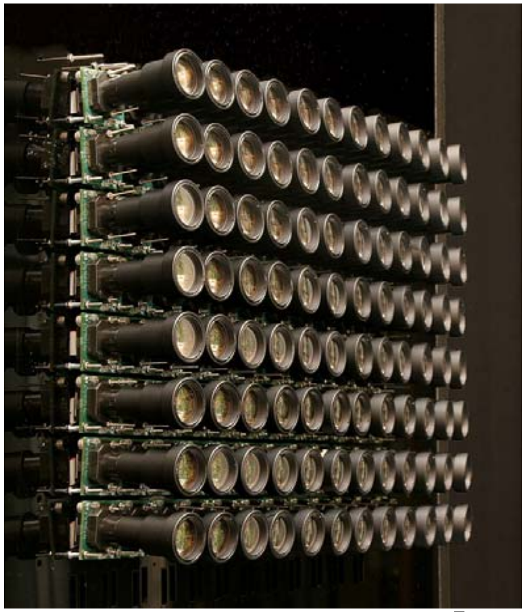
Current algorithms do not match lower bounds on sample complexity

**Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. “Nearly Optimal Sparse Fourier Transform” *STOC'12, ACM Symposium on Theory of Computing*, New York USA, May 2012.

In many applications, collecting the samples is costly

- **More samples → More Time**
  - MRI: Time patient spends in machine
  - Spectroscopy: experiments run for weeks

- **More samples → More Hardware**
  - Light field camera arrays
  - Astronomy Radar arrays



Even worse for multi-dimensional DFT

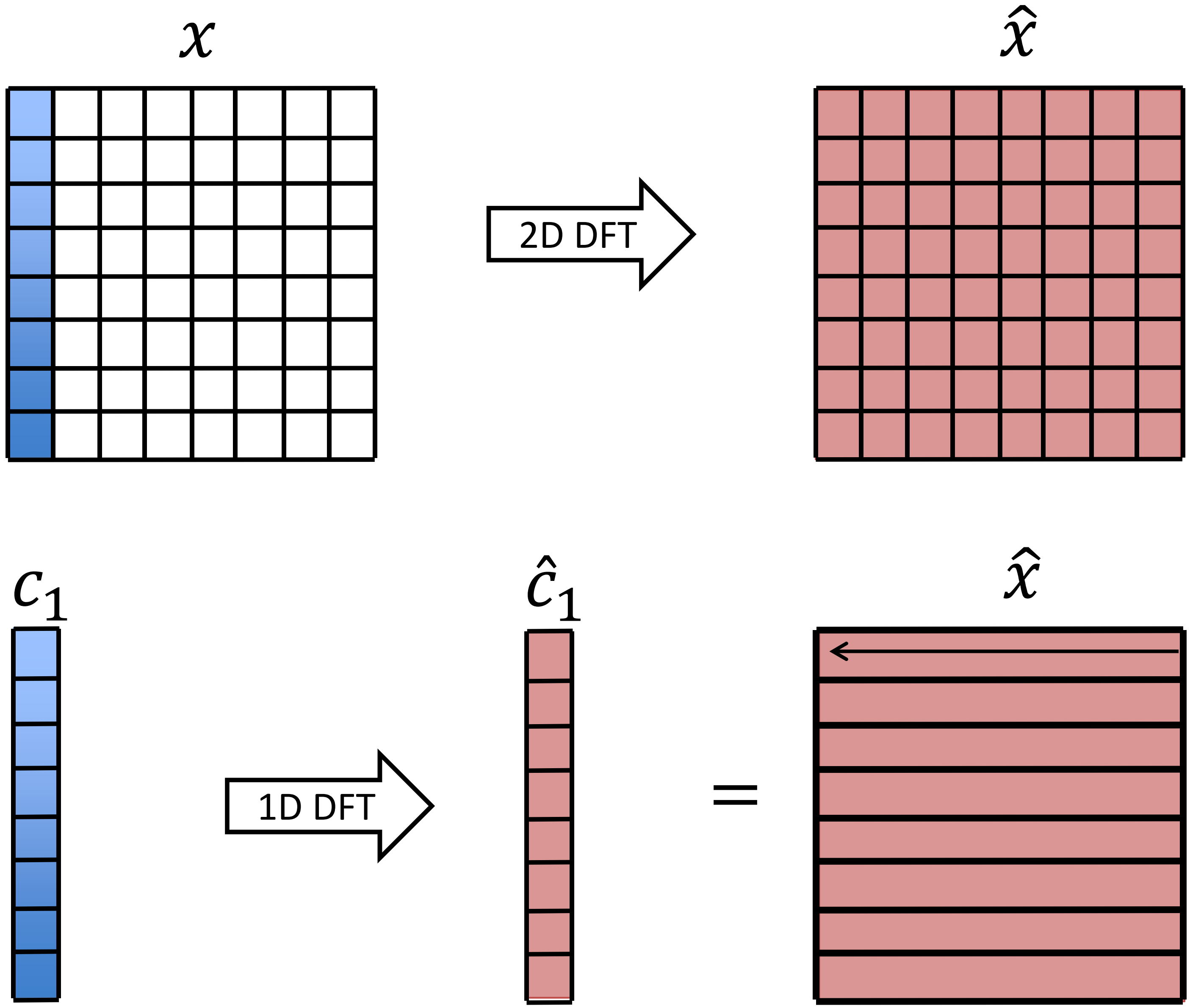
- Most applications require multi-dimensional DFT:
  - Medical Imaging: 2D – 6D
  - Spectroscopy: 2D – 7D
  - Light fields: 4D

- Current Algorithms:
  - Sample complexity worse: $k\ (\log n)^{d+1}$ instead of $k\ \log(n^{d+1})$

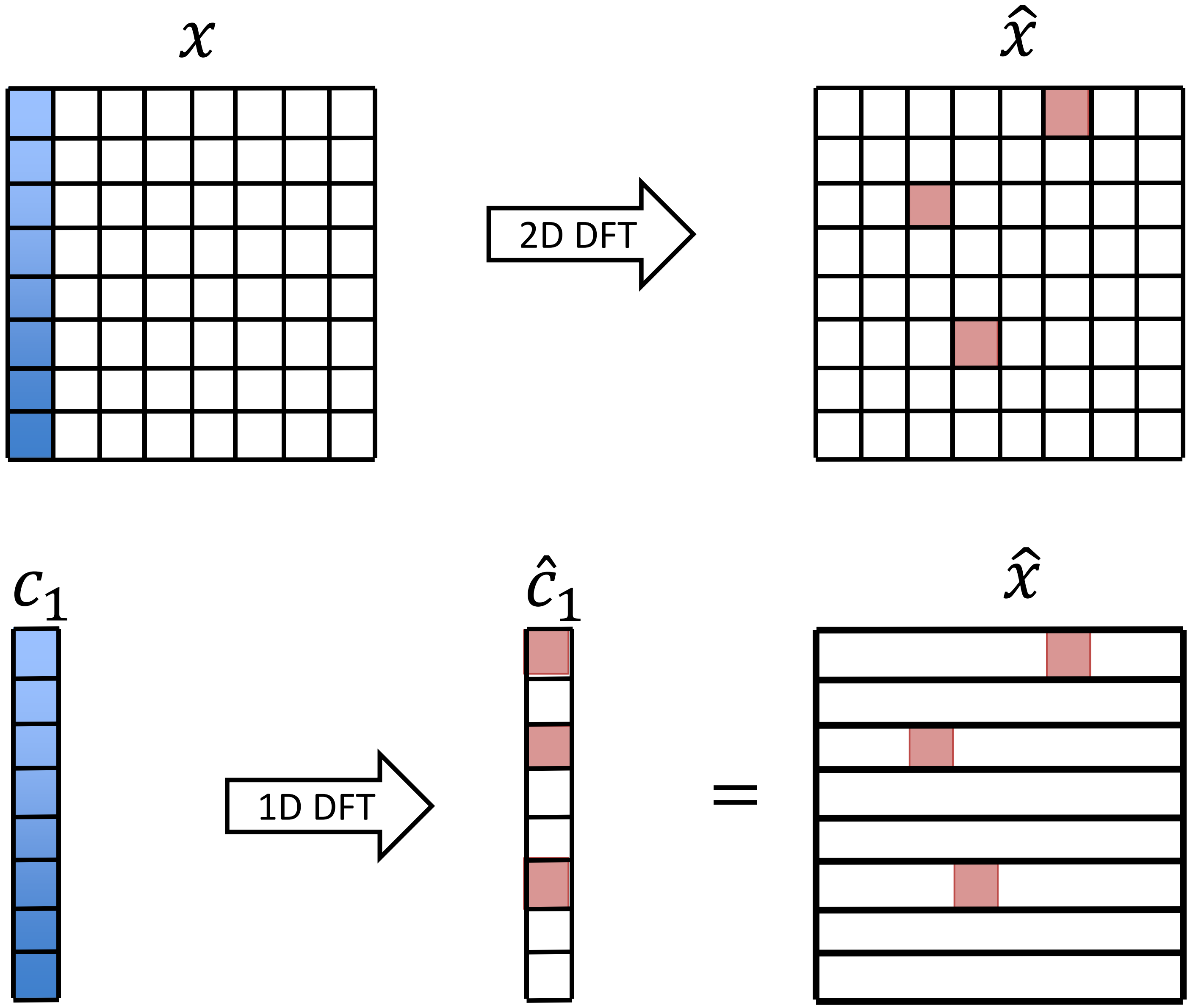
How can we match the lower bounds on sample complexity while maintaining fast run time for multi-dimensional DFT?

Sample optimal 2D sparse Fourier transform

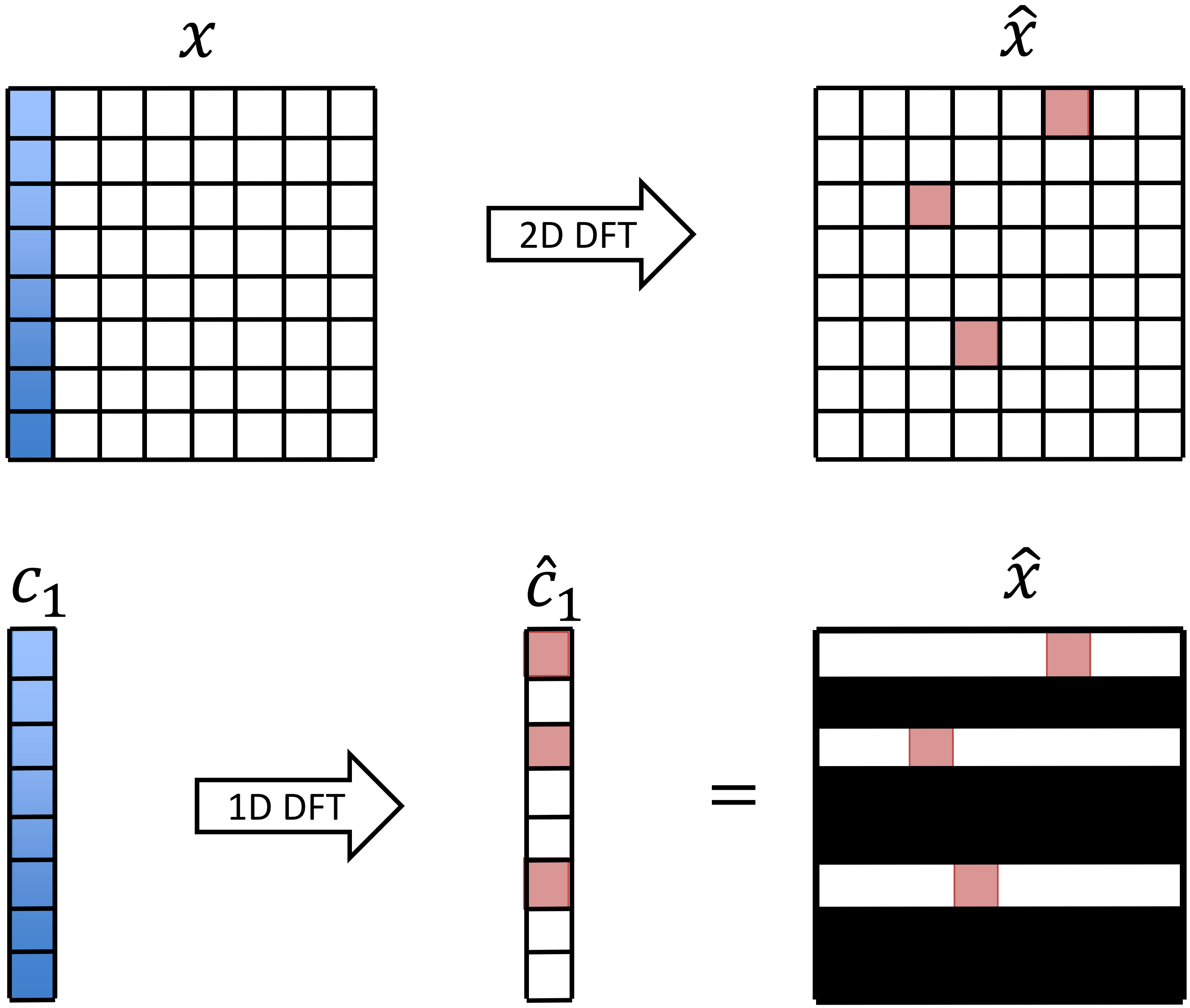
2D Sparse Fourier Transform



2D Sparse Fourier Transform



2D Sparse Fourier Transform



2D Sparse Fourier Transform

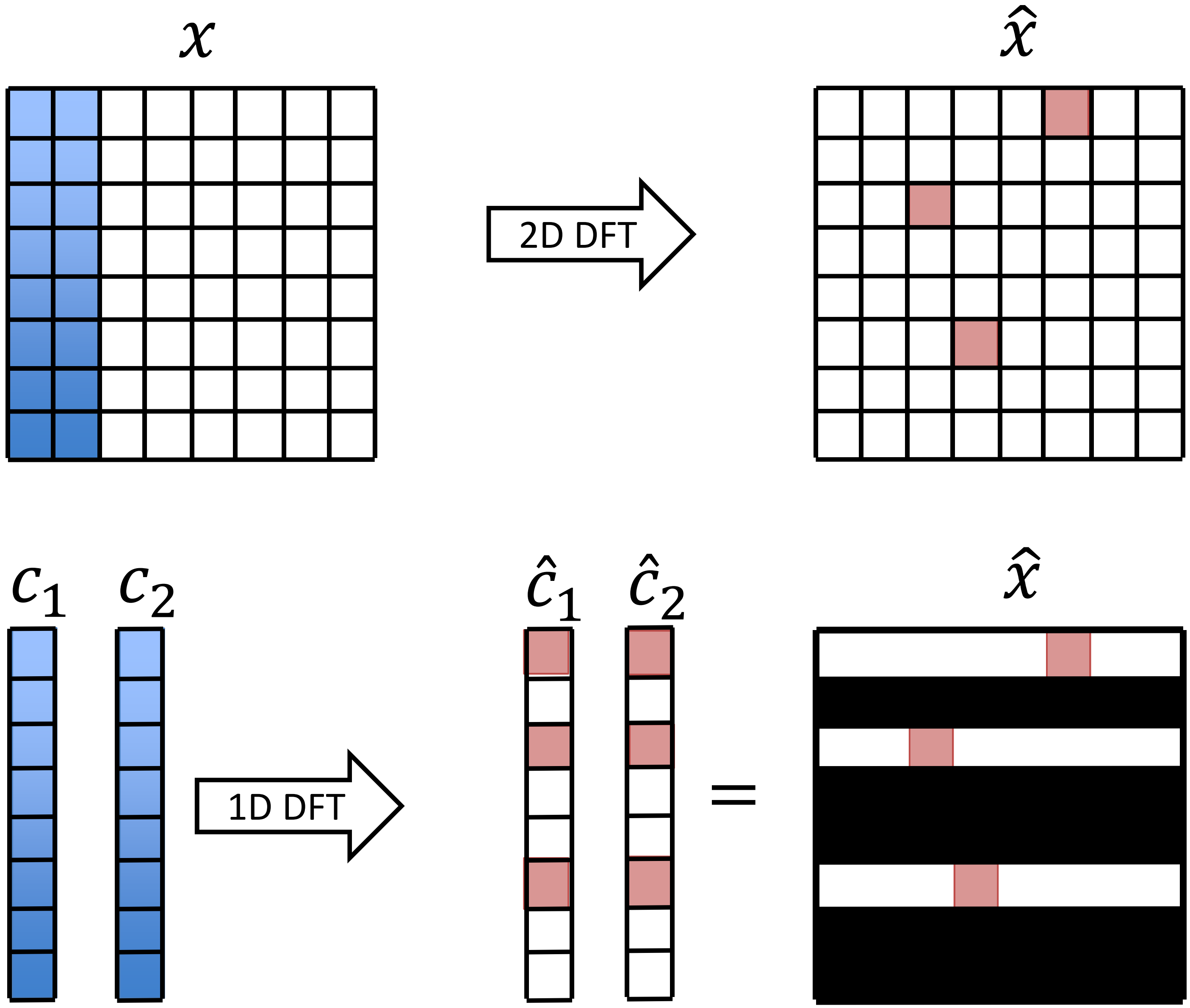
- Recall time shift property of DFT:

1D : $x(t-\tau) \rightarrow \hat{x}(f)e^{j\frac{2\pi f\tau}{N}}$

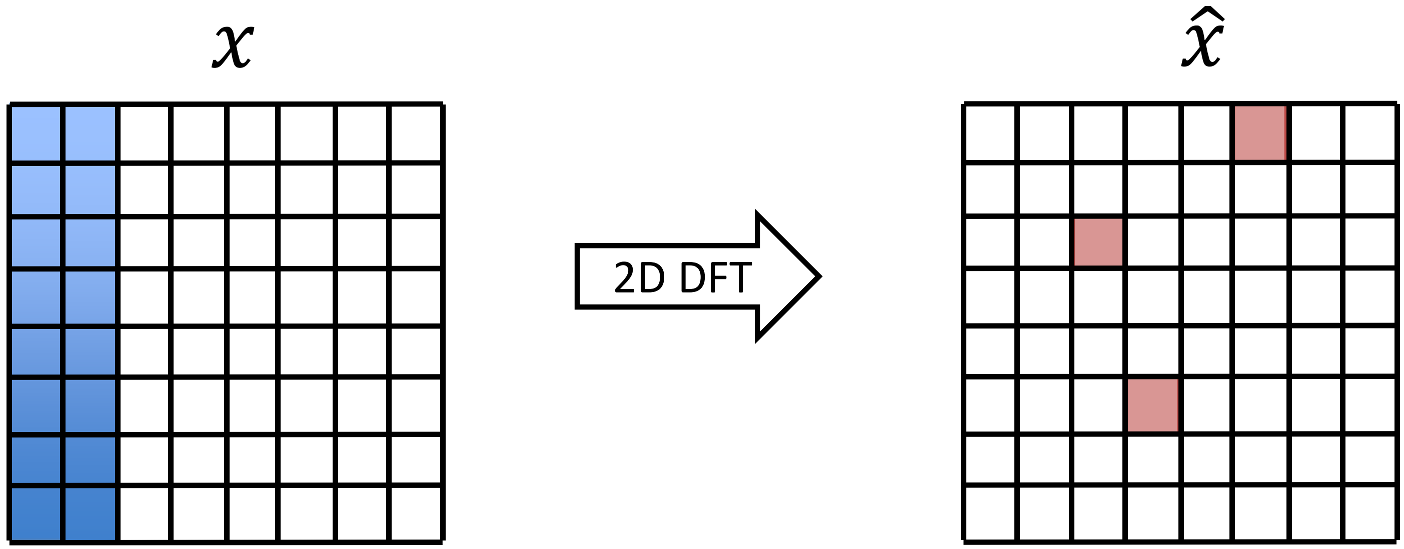
2D : $x(t_r-\tau_r, t_c-\tau_c) \rightarrow \hat{x}(f_r, f_c)e^{j2\pi\frac{f_r\tau_r+f_c\tau_c}{N}}$

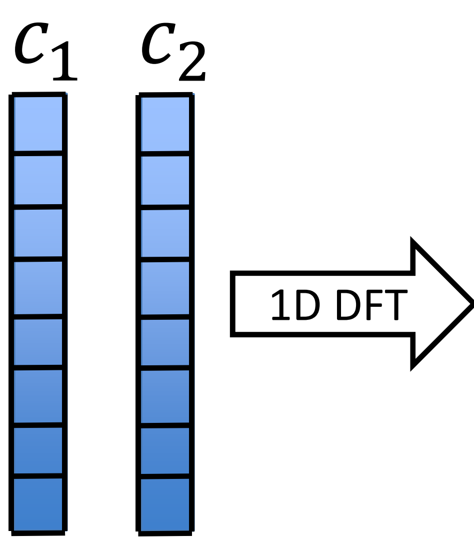
- $x(t_r-1, t_c) \rightarrow \hat{x}(f_r, f_c)e^{j2\pi\frac{f_r}{N}}$ ➔ phase α row freq. index
- $x(t_r, t_c-1) \rightarrow \hat{x}(f_r, f_c)e^{j2\pi\frac{f_c}{N}}$ ➔ phase α column freq. index

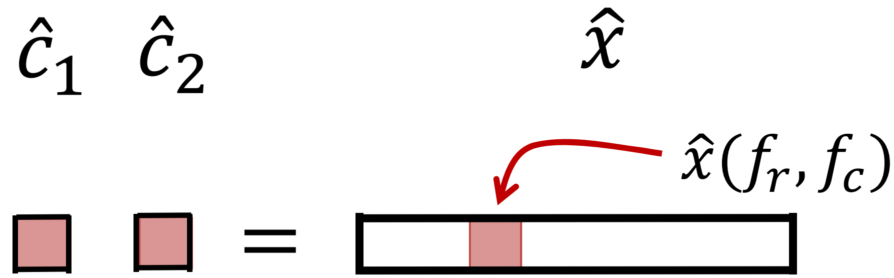
2D Sparse Fourier Transform



2D Sparse Fourier Transform





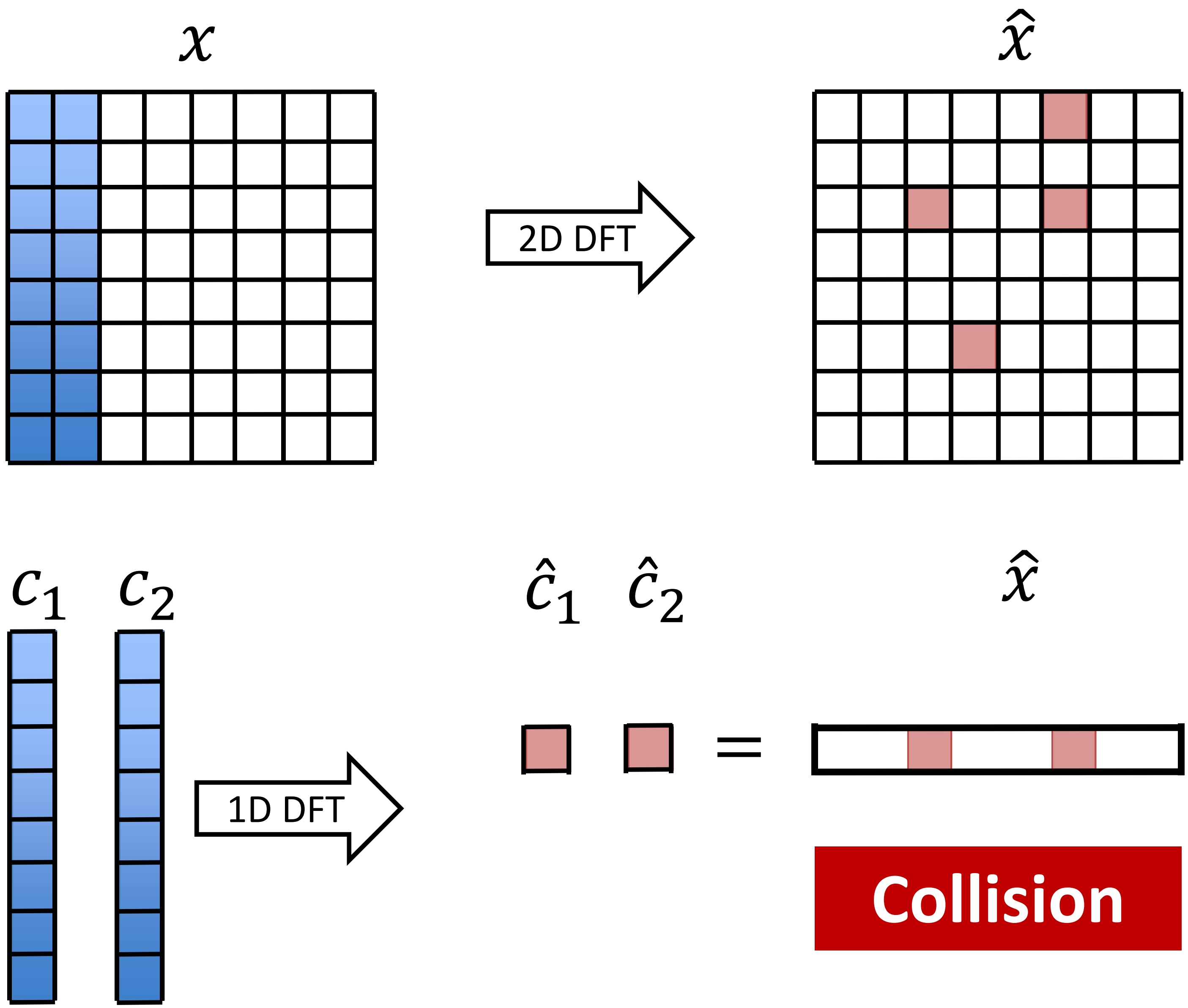


$$\hat{c}_1(f_r) = \hat{x}(f_r, f_c)$$

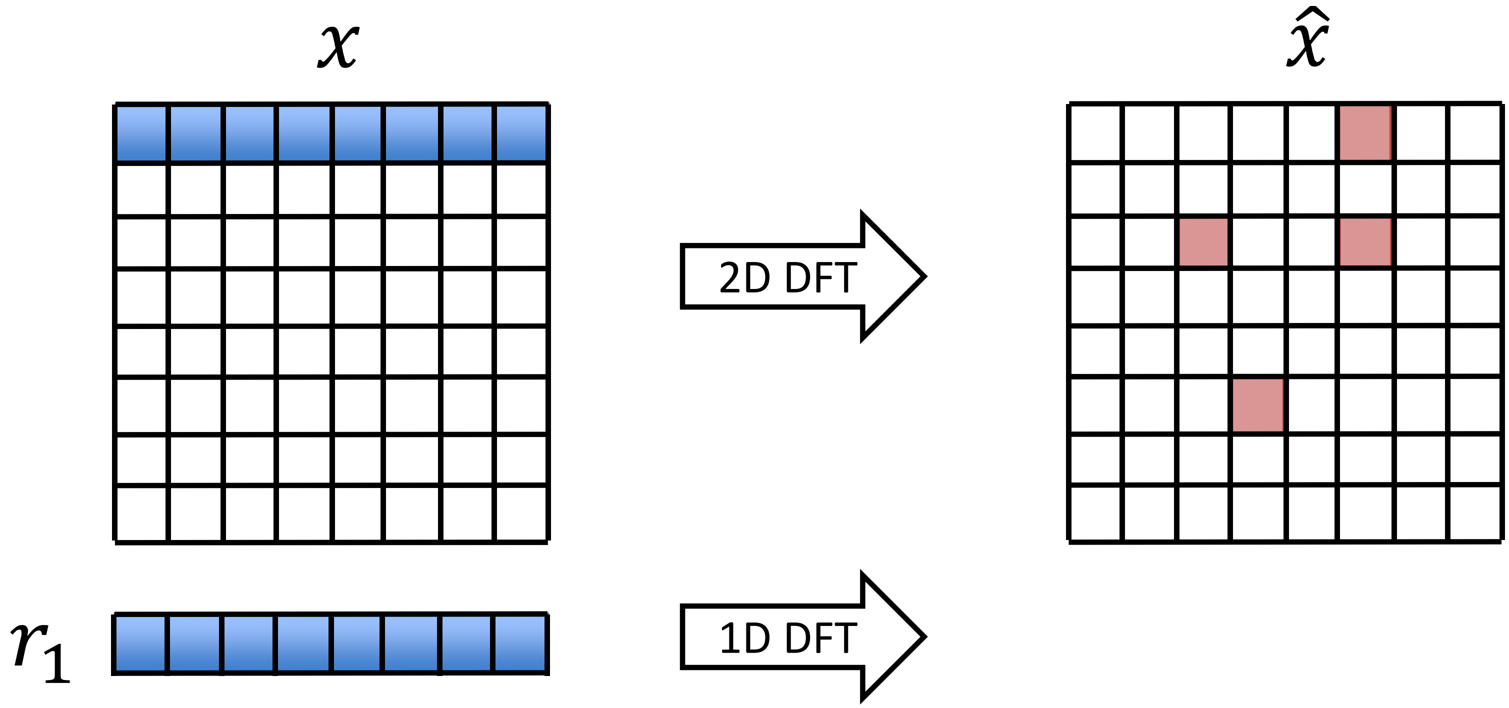
$$\hat{c}_2(f_r) = \hat{x}(f_r, f_c)e^{-j2\pi \frac{f_c}{N}}$$

position: (f_r, f_c)
value: $\hat{x}(f_r, f_c)$

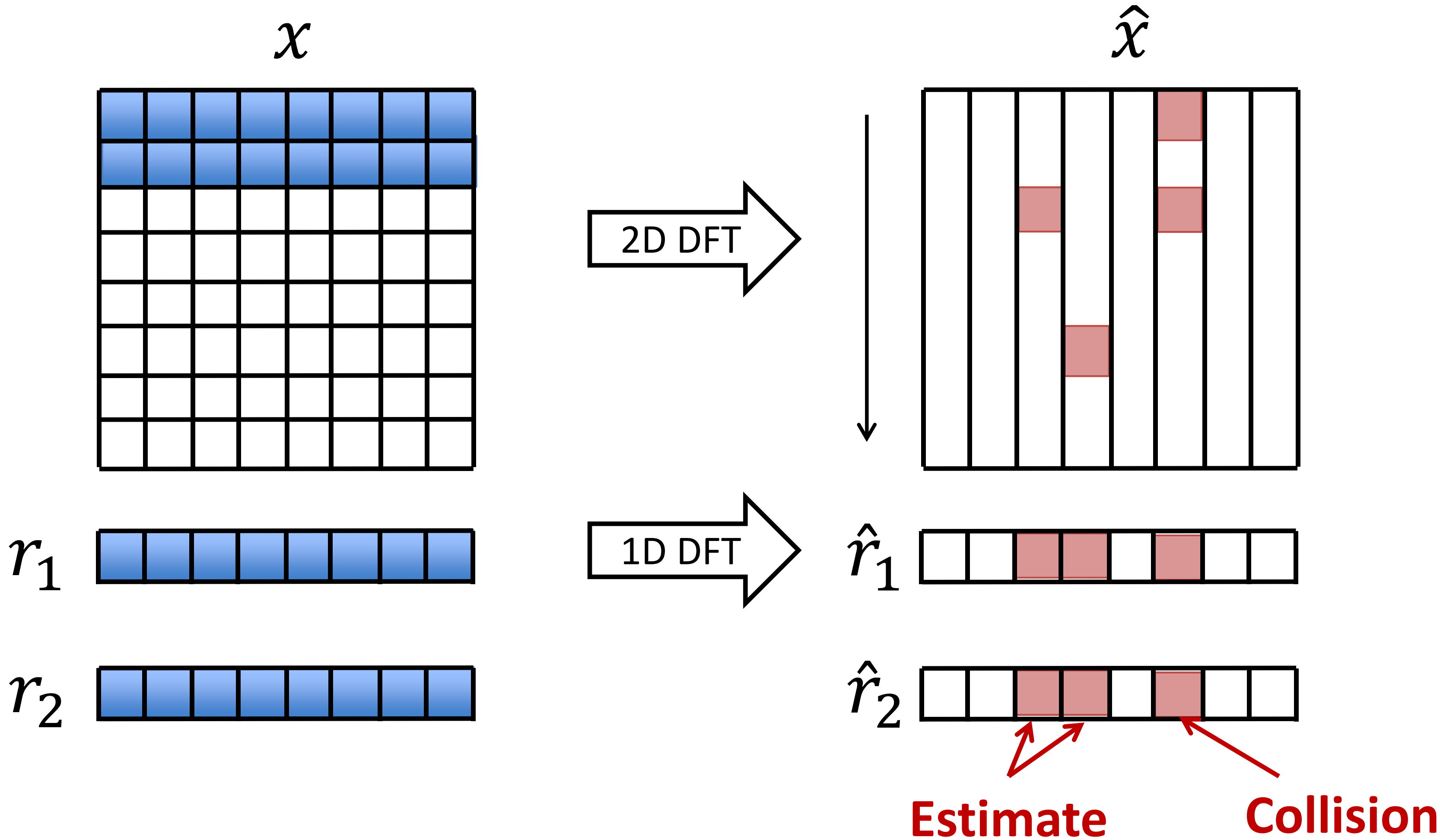
2D Sparse Fourier Transform



2D Sparse Fourier Transform



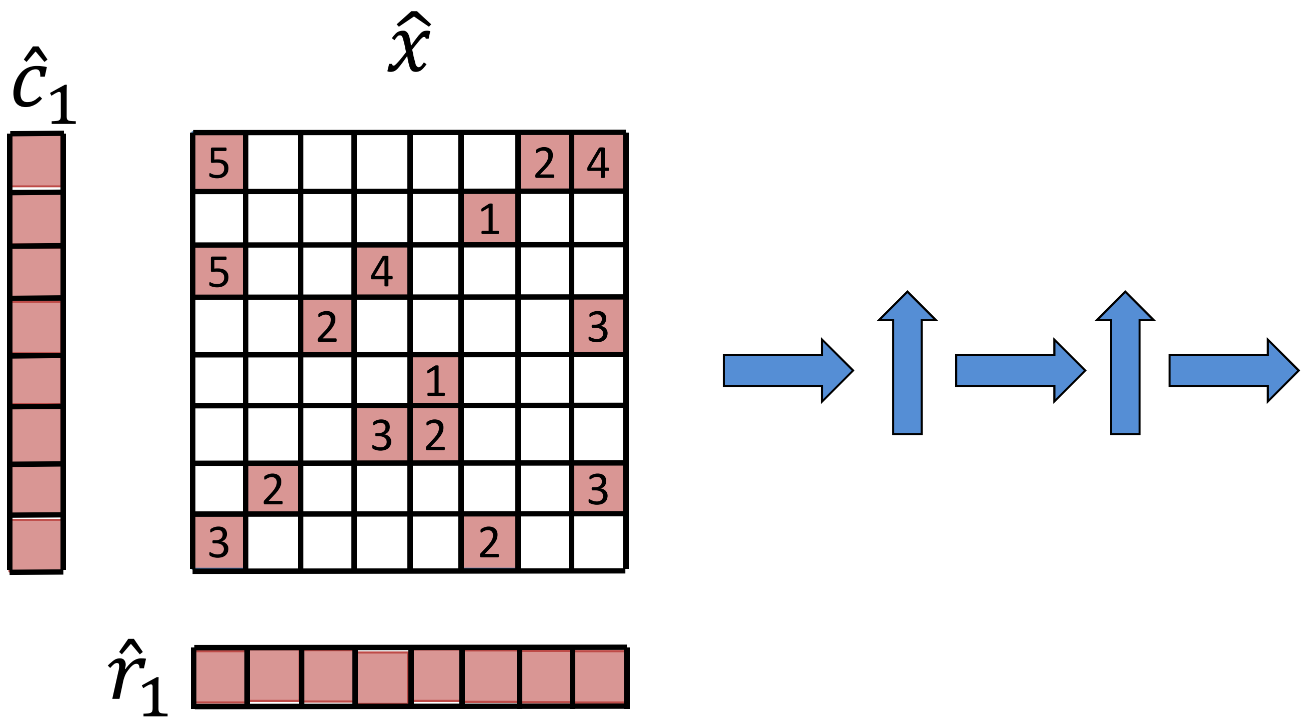
2D Sparse Fourier Transform



2D Sparse Fourier Transform

Algorithm:

- Take FFT of columns and rows
- Alternate columns/rows
- Estimate if one non-zero entry (needs collision detection)
- Subtract estimated frequencies



Analysis

- Setup:
  - 2D DFT Grid: $\sqrt{n} \times \sqrt{n}$
  - Average case: Non-zero freq. distributed uniformly at random
  - Sparsity: $k = \Theta(\sqrt{n})$
- Analysis:
  - At most one entry per column/row in expectation
  - With high probability: algorithm converges in $\mathrm{O}(\log n)$ steps
  - Each step requires columns/rows i.e. $\mathrm{O}(\sqrt{n}) = \mathrm{O}(k)$ samples
  - But we can use the same samples in all steps

- Exactly Sparse:

Sample Complexity : $\mathbf{O}(\boldsymbol{k})$

Time Complexity : $\mathbf{O}(\boldsymbol{k}\log \boldsymbol{n})$

Results

Algorithm	Time	Samples
Exactly Sparse	$O(k \log n)$	$O(k)$
Approximately Sparse	$O(k \log^2 n)$	$O(k \log n)$

- Generalize:
  - Approximately sparse case with Gaussian noise
  - Sparsity: $k = O(\sqrt{n})$
  - Dimensions > 2
  - 1D for $n = p \times q$, where p and q are co-prime**

**Similar result was recently and independently discovered by Pawar and Ramchandran

Conclusions

- Simple and practical multi-dimensional Sparse Fourier Transform algorithm
- Achieves sample complexity lower bounds
- Exactly sparse: $\mathrm{O}(k)$ samples in $\mathrm{O}(k \log n)$ time
- Approx. sparse : $\mathrm{O}(k \log n)$ samples in $\mathrm{O}(k \log^2 n)$ time
- Future directions:
  – Optimal sample complexity for worst case ?

2D Sparse Fourier Transform