An Adaptive Sublinear Time Block Sparse Fourier Transform

Volkan Cevher **Michael Kapralov** Jonathan Scarlett
Amir Zandieh

EPFL

February 8th 2017

Given $x \in \mathbb{C}^N$, compute the Discrete Fourier Transform (DFT) of x:

$$\widehat{x}_i = \frac{1}{N} \sum_{j \in [N]} x_j \cdot \omega^{-ij},$$

where $\omega = e^{2\pi i/N}$ is the N-th root of unity.

Given $x \in \mathbb{C}^N$, compute the Discrete Fourier Transform (DFT) of x:

$$\widehat{x}_i = \frac{1}{N} \sum_{j \in [N]} x_j \cdot \omega^{-ij},$$

where $\omega = e^{2\pi i/N}$ is the N-th root of unity.

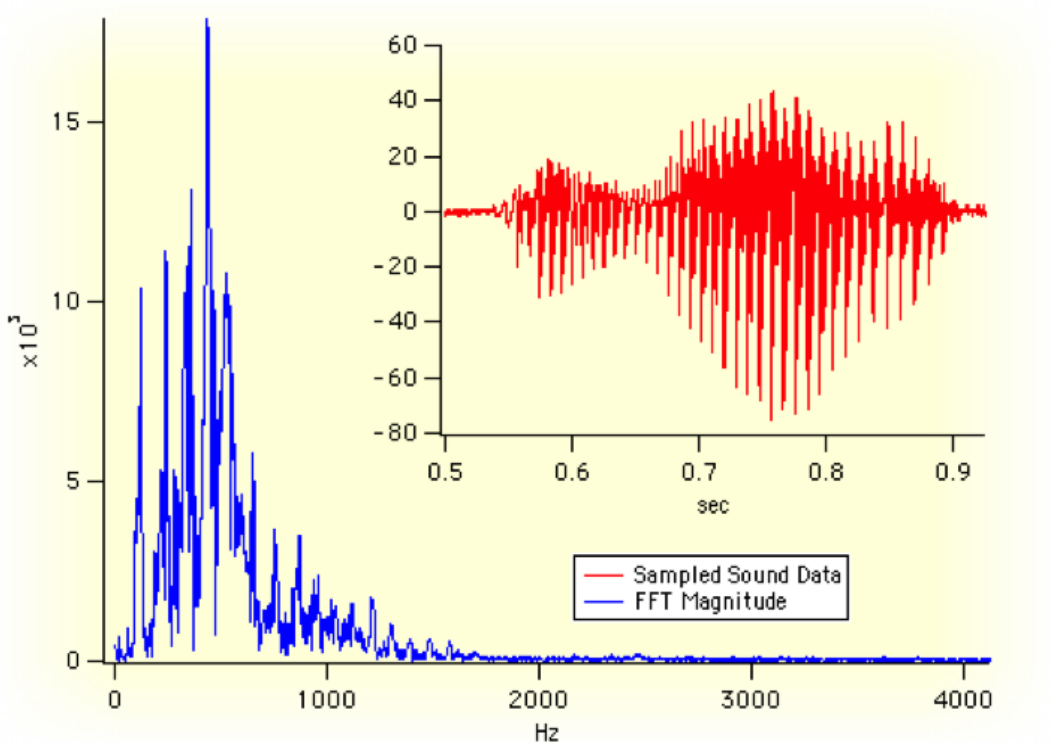
Assume that N is a power of 2.

Given $x \in \mathbb{C}^N$, compute the Discrete Fourier Transform (DFT) of x:

$$\widehat{x}_i = \frac{1}{N} \sum_{j \in [N]} x_j \cdot \omega^{-ij},$$

where $\omega = e^{2\pi i/N}$ is the N-th root of unity.

Assume that N is a power of 2.



compression schemes (JPEG, MPEG)
signal processing
data analysis
imaging (MRI, NMR)

Fast Fourier Transform (FFT)

Computes Discrete Fourier Transform (DFT) of a length N signal in $O(N \log N)$ time

Fast Fourier Transform (FFT)

Computes Discrete Fourier Transform (DFT) of a length N signal in $O(N \log N)$ time

Cooley and Tukey, 1964

Fast Fourier Transform (FFT)

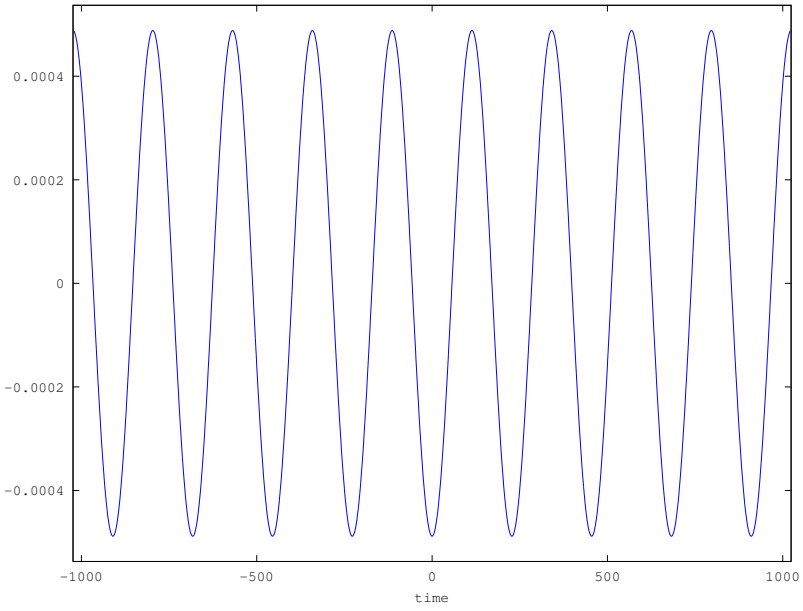
Computes Discrete Fourier Transform (DFT) of a length N signal in $O(N \log N)$ time

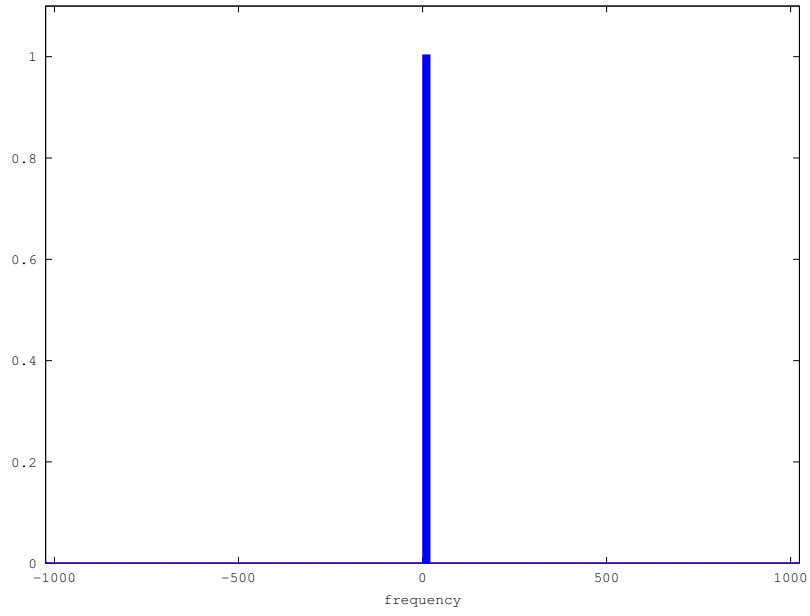
Cooley and Tukey, 1964

Gauss, 1805

Sparse FFT

Say that $\hat{x}$ is *k*-sparse if $\hat{x}$ has k nonzero entries

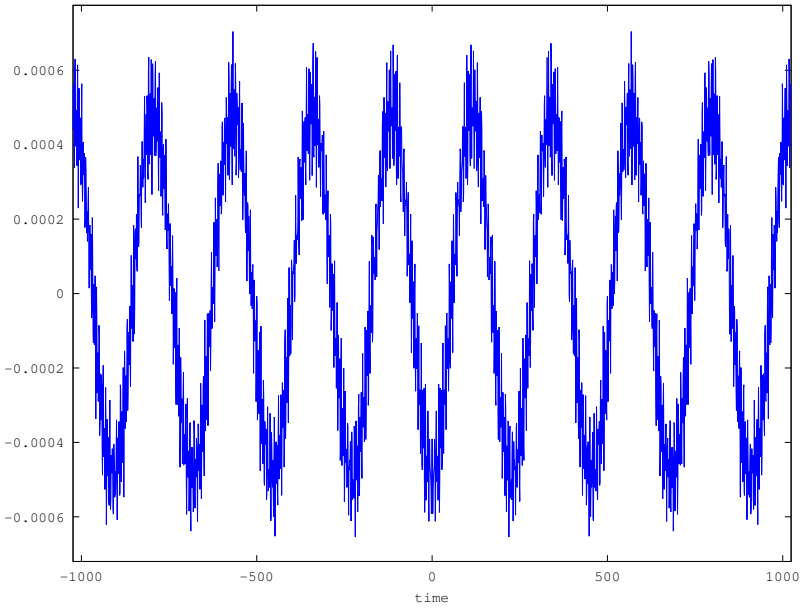


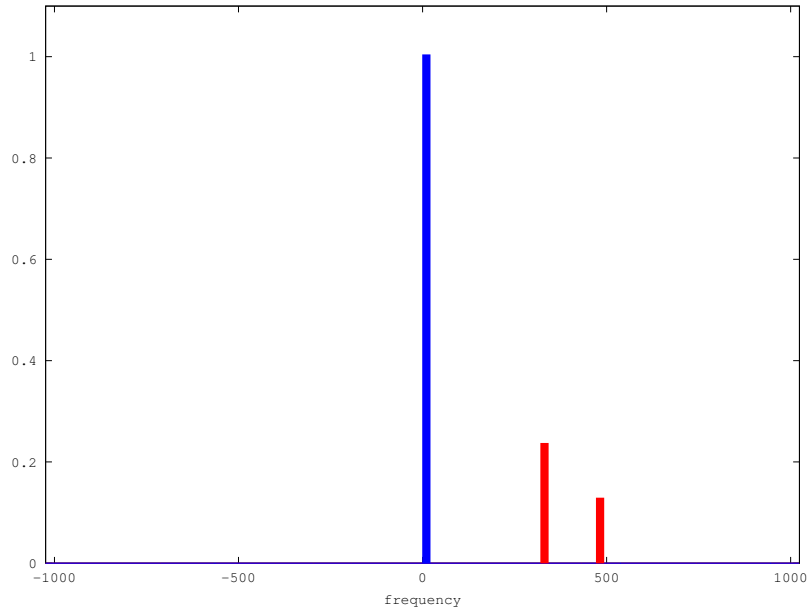


Sparse FFT

Say that $\hat{x}$ is *k*-sparse if $\hat{x}$ has *k* nonzero entries

Say that $\hat{x}$ is approximately *k*-sparse if $\hat{x}$ is close to *k*-sparse in some norm





Sparse approximations

JPEG
$\Longrightarrow$

Image and video compression schemes
(e.g. JPEG, MPEG)

Compute top k coefficients of $\hat{x}$ faster than FFT?

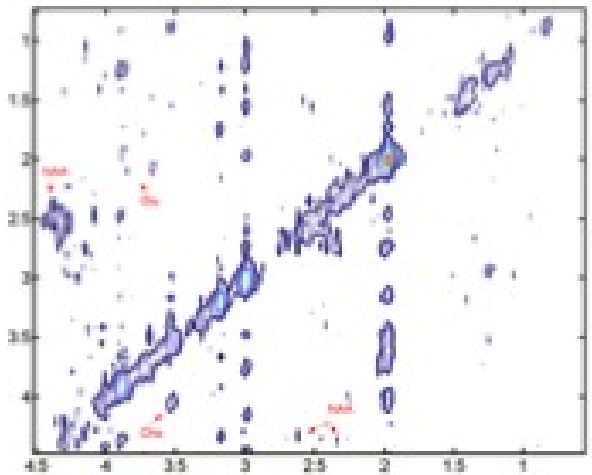
Sample complexity

Sample complexity=number of samples accessed in time domain.

Sample complexity

Sample complexity=number of samples accessed in time domain.

In medical imaging (MRI, NMR), one measures Fourier coefficients $\hat{x}$ of imaged object x (which is often sparse)



Given access to signal x in time domain, (approximately) compute top k coefficients of $\widehat{x}$

Minimize runtime and sample complexity

Given access to signal x in time domain, (approximately) compute top k coefficients of $\widehat{x}$

Minimize runtime and sample complexity

Formally, want to find $\widehat{y}$ such that

$$||\widehat{x} - \widehat{y}||^2 \leq (1+\varepsilon) \cdot \min_{k-\text{sparse } \widehat{z}} ||\widehat{x} - \widehat{z}||^2$$

(ℓ_2/ℓ_2 sparse recovery guarantees)

Uniform bounds (for all):

Candes-Tao'06
Rudelson-Vershynin'08
Cheraghchi-Guruswami-Velingker'12
Bourgain'14
Haviv-Regev'15

Deterministic, $\Omega(N)$ runtime

$O(k \log^2 k \log N)$

Non-uniform bounds (for each):

Goldreich-Levin'89
Kushilevitz-Mansour'91, Mansour'92
Gilbert-Guha-Indyk-Muthukrishnan-Strauss'02
Gilbert-Muthukrishnan-Strauss'05
Hassanieh-Indyk-Katabi-Price'12a
Hassanieh-Indyk-Katabi-Price'12b
Indyk-K.-Price'14 $(k \log n (\log\log n)^C$, but only in 1d)
Indyk-K.'14
K'16
K'??

Randomized, $O(k \cdot \mathrm{poly}(\log N))$ runtime

$O(k \log N)$

Lower bound: $k \log(N/k)$ for non-adaptive algorithms Do-Ba-Indyk-Price-Woodruff'10

(Also Boufounos-Cevher-Gilbert-Li-Strauss'12, Prince-Song'14 on continuous Sparse FFT)

Uniform bounds (for all):

Candes-Tao'06
Rudelson-Vershynin'08
Cheraghchi-Guruswami-Velingker'12
Bourgain'14
Haviv-Regev'15

Deterministic, $\Omega(N)$ runtime

$O(k \log^2 k \log N)$

Non-uniform bounds (for each):

Goldreich-Levin'89
Kushilevitz-Mansour'91, Mansour'92
Gilbert-Guha-Indyk-Muthukrishnan-Strauss'02
Gilbert-Muthukrishnan-Strauss'05
Hassanieh-Indyk-Katabi-Price'12a
Hassanieh-Indyk-Katabi-Price'12b
Indyk-K.-Price'14 $(k \log n (\log\log n)^C$, but only in 1d)
Indyk-K.'14
K'16
K'??

Randomized, $O(k \cdot \mathrm{poly}(\log N))$ runtime

$O(k \log N)$

Lower bound: $k \log(N/k)$ for non-adaptive algorithms Do-Ba-Indyk-Price-Woodruff'10

(Also Boufounos-Cevher-Gilbert-Li-Strauss'12, Prince-Song'14 on continuous Sparse FFT)

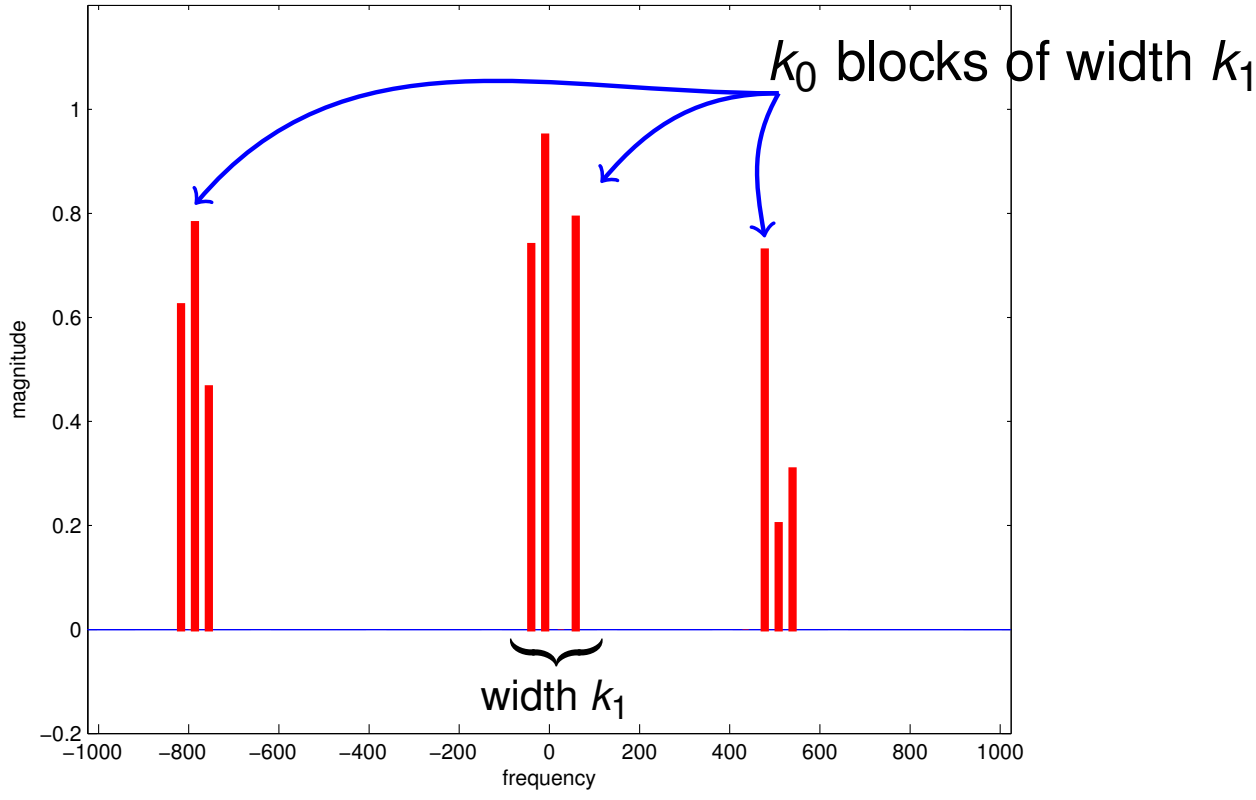
Nearly optimal results under sparsity assumption alone – can we exploit structure beyond sparsity?

Sparse FFT beyond the sparsity assumption

Signals arising in applications often have structure beyond sparsity

Can we exploit further structure to reduce sample complexity and runtime?

Block sparsity in Fourier domain



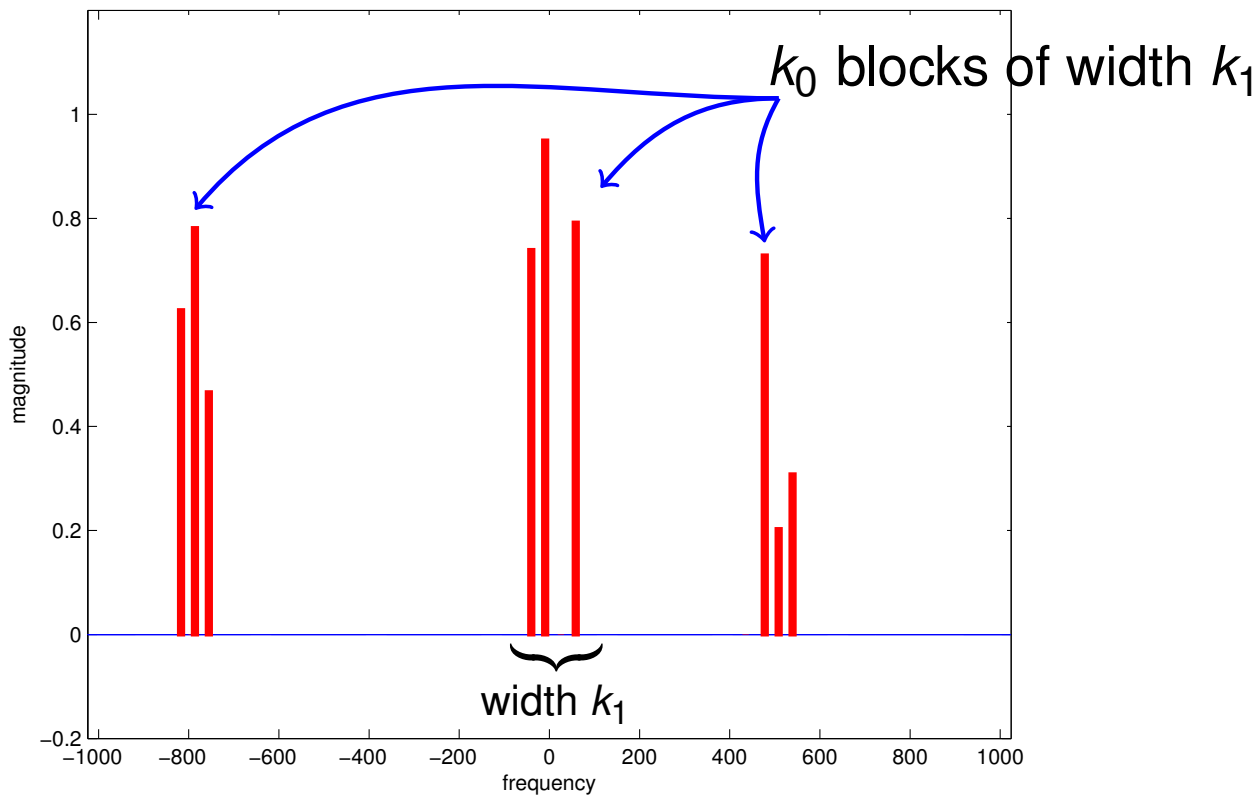
This work: suppose that dominant frequencies are contained in k_0 clusters (intervals) of length k_1 each?

$O(k_0 \log n + k_0 k_1)$ sample complexity in sublinear time?

Model based compressed sensing

Framework for exploiting structured sparsity to reduce sample complexity, introduced by Baraniuk, Cevher, Duarte and Hegde'10

Baraniuk et al'10: optimal $O(k_0 \log n + k_0 k_1)$ sample complexity using Gaussian measurements in $\widetilde{O}(k_0 k_1 n)$ time



Adaptivity in compressed sensing

An algorithm is adaptive its signal access pattern is guided by samples taken

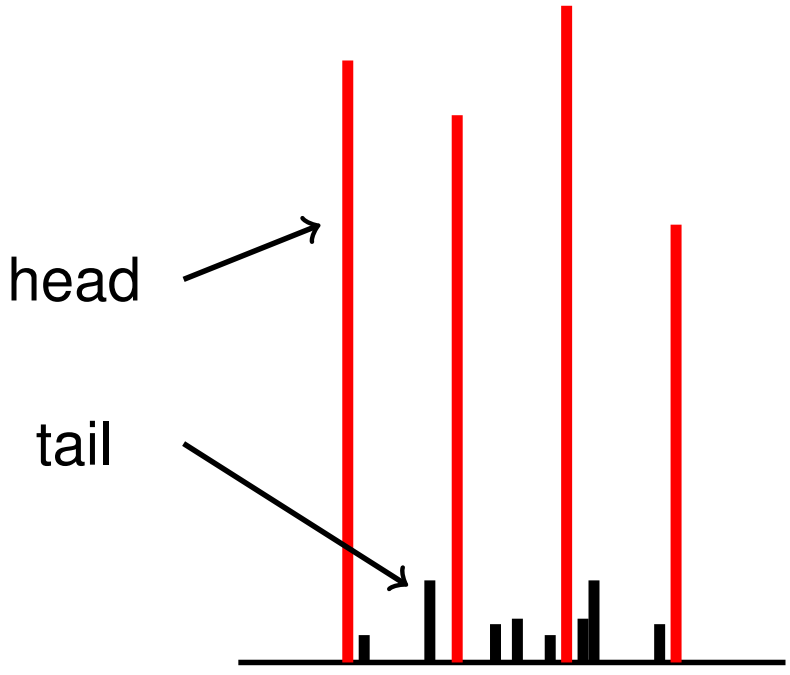
All Sparse FFT and compressed sensing results mentioned so far are non-adaptive

Main result

Theorem

There exists an adaptive (k_0, k_1)*-block sparse Fourier transform with sample complexity* $O^*(k_0 \log(1+k_0) \log n + k_0 k_1) \cdot \log SNR$ *samples and* $O^*(k_0 k_1 \log^3 n) \cdot \log SNR$ *runtime.*

Sample complexity optimal up to lower order terms for constant SNR



SNR=ratio of total signal energy to total noise energy

Main result

Theorem

There exists an adaptive (k_0, k_1)*-block sparse Fourier transform with sample complexity* $O^*(k_0 \log(1+k_0) \log n + k_0 k_1) \cdot \log SNR$ *samples and* $O^*(k_0 k_1 \log^3 n) \cdot \log SNR$ *runtime.*

Sample complexity optimal up to lower order terms for constant SNR

Main result

Theorem

There exists an adaptive (k_0, k_1)*-block sparse Fourier transform with sample complexity* $O^*(k_0 \log(1+k_0) \log n + k_0 k_1) \cdot \log SNR$ *samples and* $O^*(k_0 k_1 \log^3 n) \cdot \log SNR$ *runtime.*

Sample complexity optimal up to lower order terms for constant SNR

Crucially use adaptivity!

Main result

Theorem

There exists an adaptive (k_0, k_1)*-block sparse Fourier transform with sample complexity* $O^*(k_0 \log(1+k_0) \log n + k_0 k_1) \cdot \log SNR$ *samples and* $O^*(k_0 k_1 \log^3 n) \cdot \log SNR$ *runtime.*

Sample complexity optimal up to lower order terms for constant SNR

Crucially use adaptivity!

Theorem

Any non-adaptive Sparse FFT algorithm must take $\Omega(k_0 k_1 \log \frac{n}{k_0 k_1})$ *samples.*

Main result

Theorem

There exists an adaptive (k_0, k_1)*-block sparse Fourier transform with sample complexity* $O^*(k_0 \log(1+k_0) \log n + k_0 k_1) \cdot \log SNR$ *samples and* $O^*(k_0 k_1 \log^3 n) \cdot \log SNR$ *runtime.*

Sample complexity optimal up to lower order terms for constant SNR

Crucially use adaptivity!

Theorem

Any non-adaptive Sparse FFT algorithm must take $\Omega(k_0 k_1 \log \frac{n}{k_0 k_1})$ *samples.*

No improvement upon vanilla sparsity possible with non-adaptive Fourier measurements

First result along the following directions:

- First sublinear time model based compressed sensing primitive
- Separation between adaptive vs non-adaptive Sparse FFT
- Structured (Fourier) vs unstructured (Gaussian) measurements in model based compressed sensing

1. Sparse recovery from Fourier measurements

2. Main result

1. **Sparse recovery from Fourier measurements**

2. Main result

Structure of Sparse FFT algorithms

Input: access to signal $x \in \mathbb{C}^N$ in time domain
Output: top k coefficients of $\widehat{x}$, approximately

Structure of Sparse FFT algorithms

Input: access to signal $x \in \mathbb{C}^N$ in time domain
Output: top k coefficients of $\widehat{x}$, approximately

Iterative process:

- recover dominant coefficients of input signal

Structure of Sparse FFT algorithms

Input: access to signal $x \in \mathbb{C}^N$ in time domain
Output: top k coefficients of $\widehat{x}$, approximately

Iterative process:

- recover dominant coefficients of input signal
- subtract recovered coefficients (e.g. in time domain)

Structure of Sparse FFT algorithms

Input: access to signal $x \in \mathbb{C}^N$ in time domain
Output: top k coefficients of $\widehat{x}$, approximately

Iterative process:

- recover dominant coefficients of input signal
- subtract recovered coefficients (e.g. in time domain)
- repeat

Structure of Sparse FFT algorithms

Input: access to signal $x \in \mathbb{C}^N$ in time domain
Output: top k coefficients of $\widehat{x}$, approximately

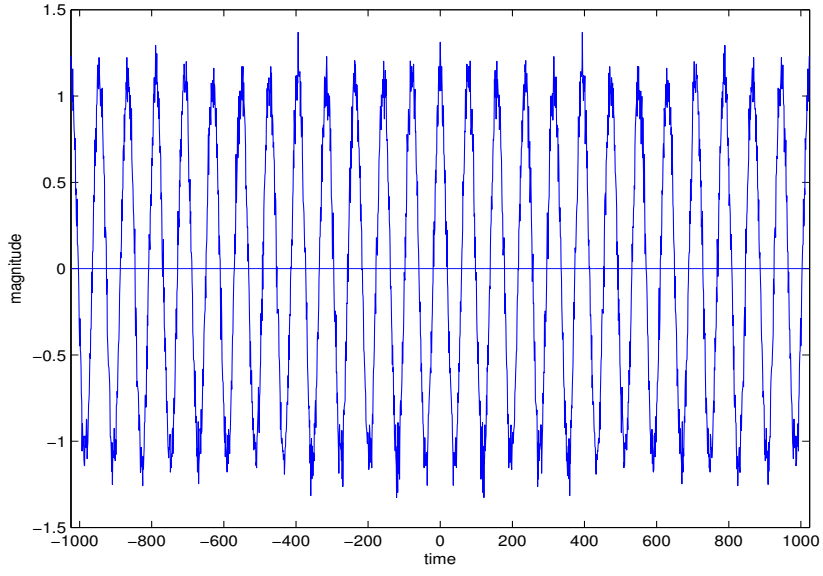
Iterative process:

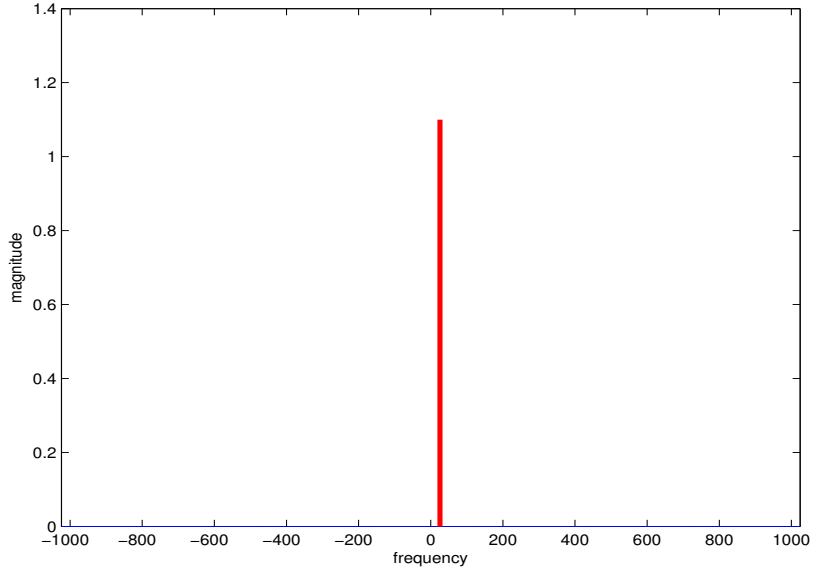
- recover dominant coefficients of input signal
- subtract recovered coefficients (e.g. in time domain)
- repeat

Summary of techniques from

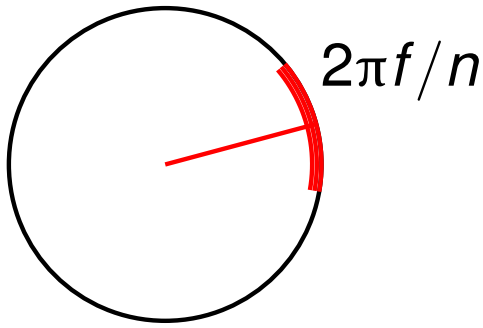
Gilbert-Guha-Indyk-Muthukrishnan-Strauss'02, Akavia-Goldwasser-Safra'03,
Gilbert-Muthukrishnan-Strauss'05, Iwen'10, Akavia'10,
Hassanieh-Indyk-Katabi-Price'12a, Hassanieh-Indyk-Katabi-Price'12b

1-sparse recovery from Fourier measurements





$$x_a = \omega^{a \cdot f} + \text{noise}$$

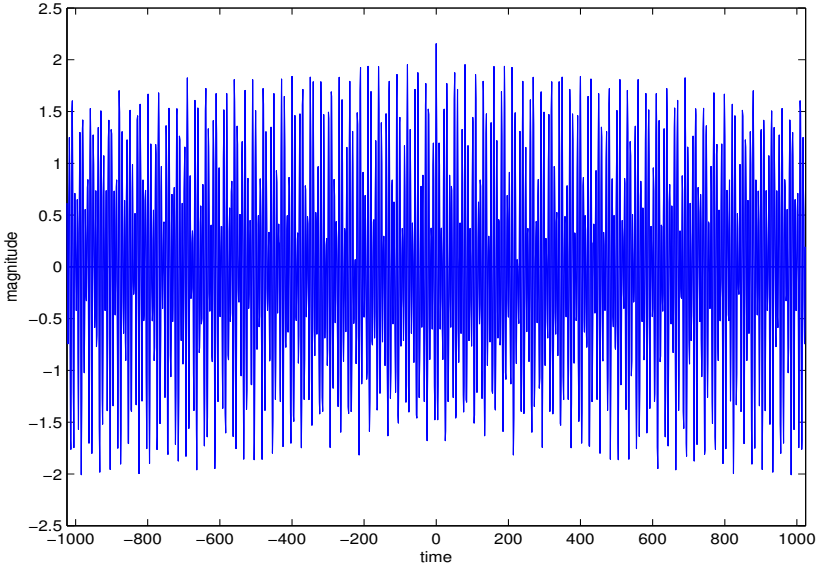


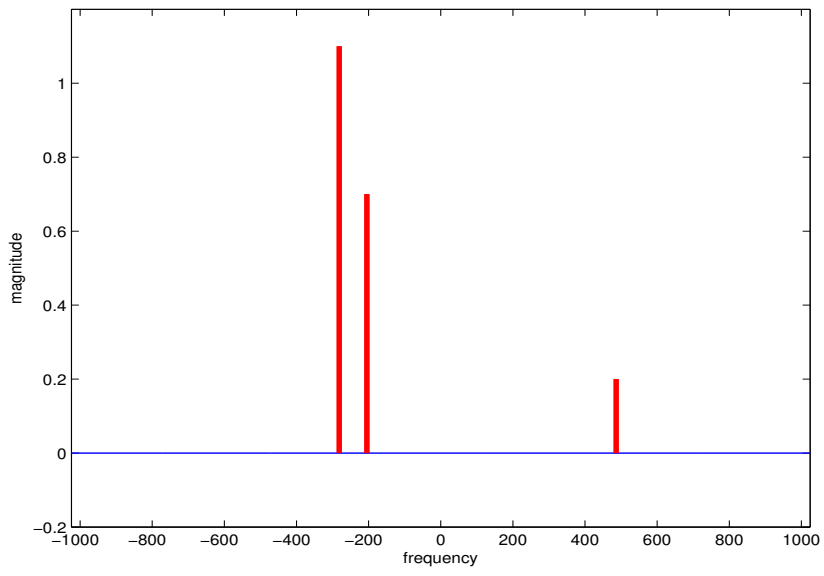
$O(\log n)$ measurements

for random a

Reducing k-sparse recovery to 1-sparse recovery

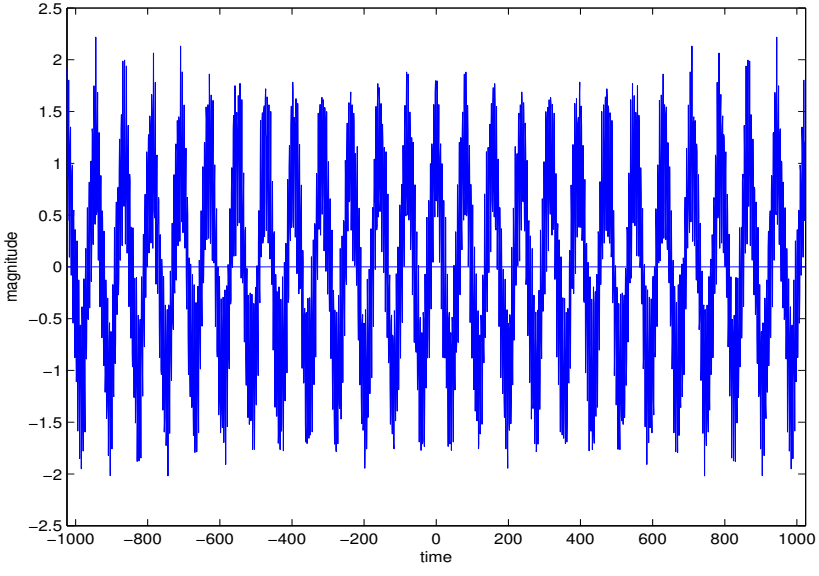
Permute with a random linear transformation and phase shift

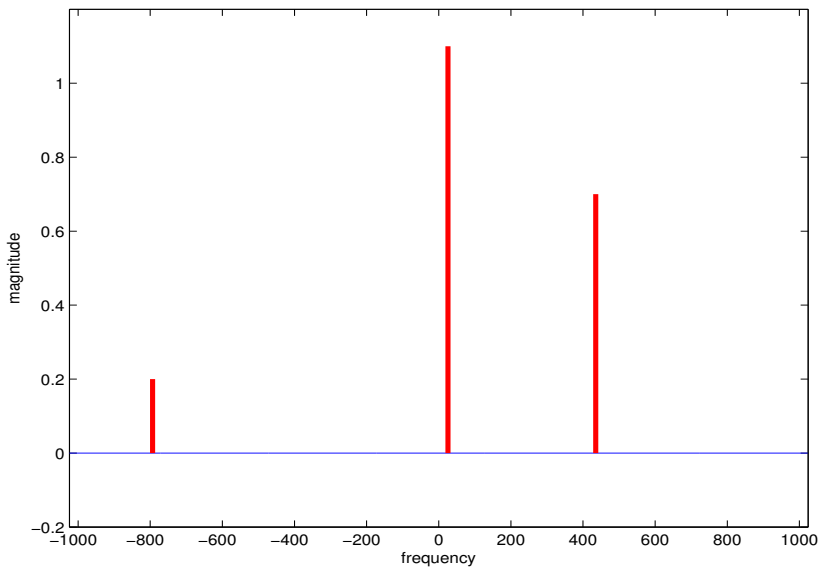




Reducing *k*-sparse recovery to 1-sparse recovery

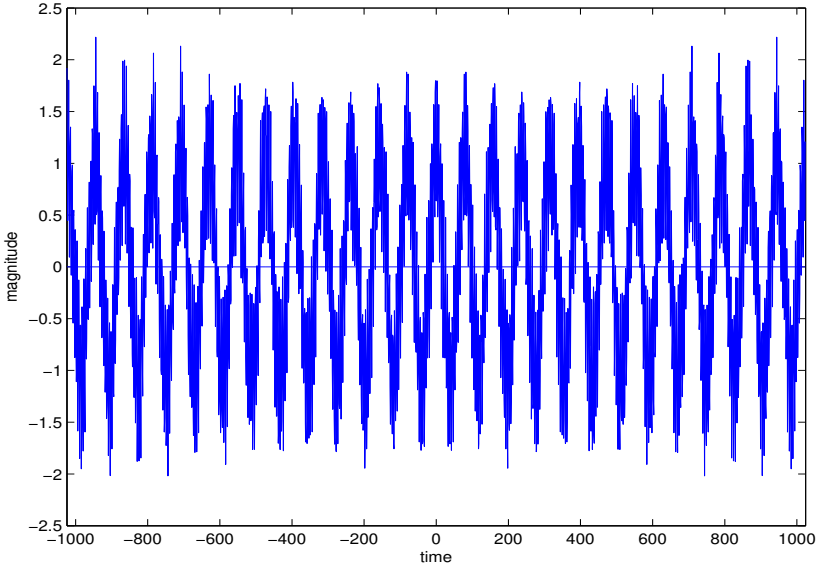
Permute with a random linear transformation and phase shift

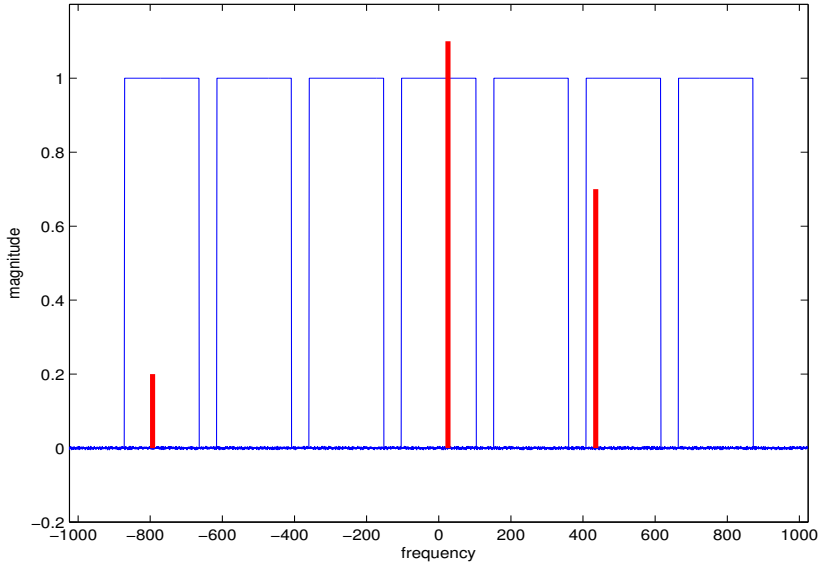




Reducing k-sparse recovery to 1-sparse recovery

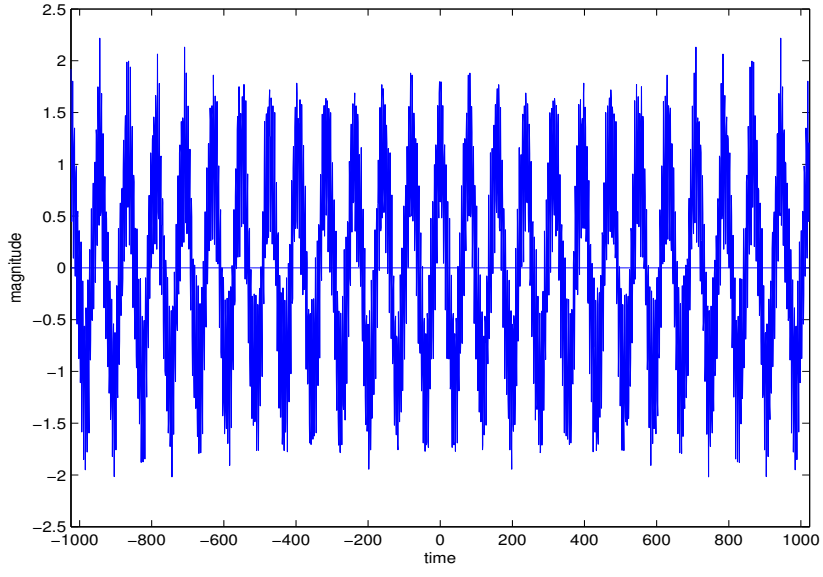
Permute with a random linear transformation and phase shift

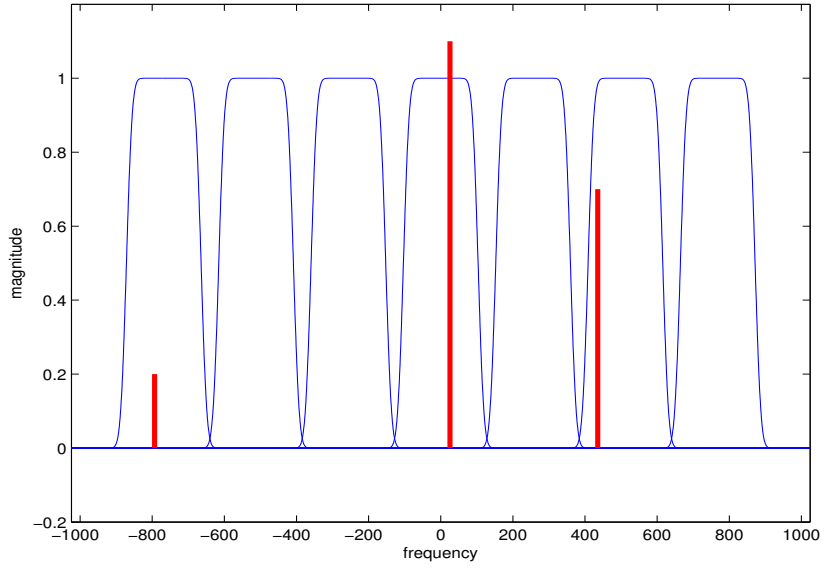




Reducing k-sparse recovery to 1-sparse recovery

Permute with a random linear transformation and phase shift





Choose a filter $G, \widehat{G}$ such that

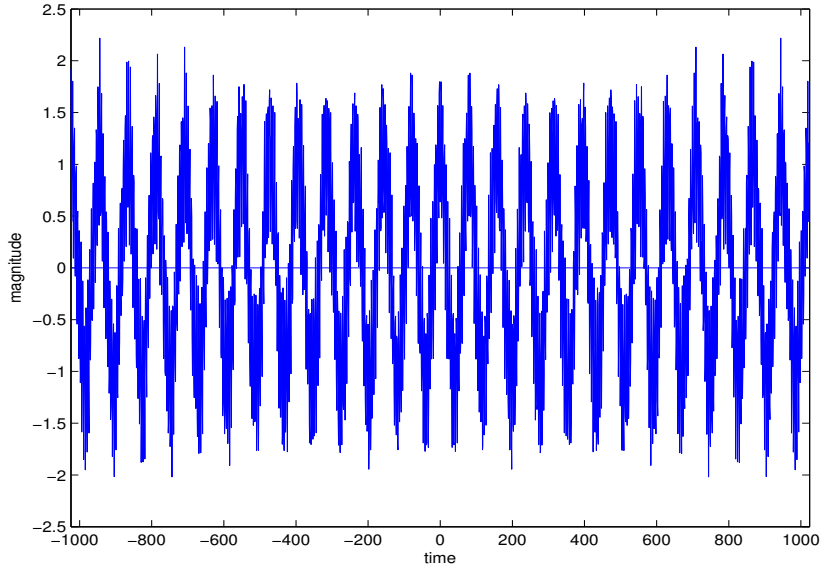
- $\widehat{G} \approx$ rectangle
- G has support $\approx k$

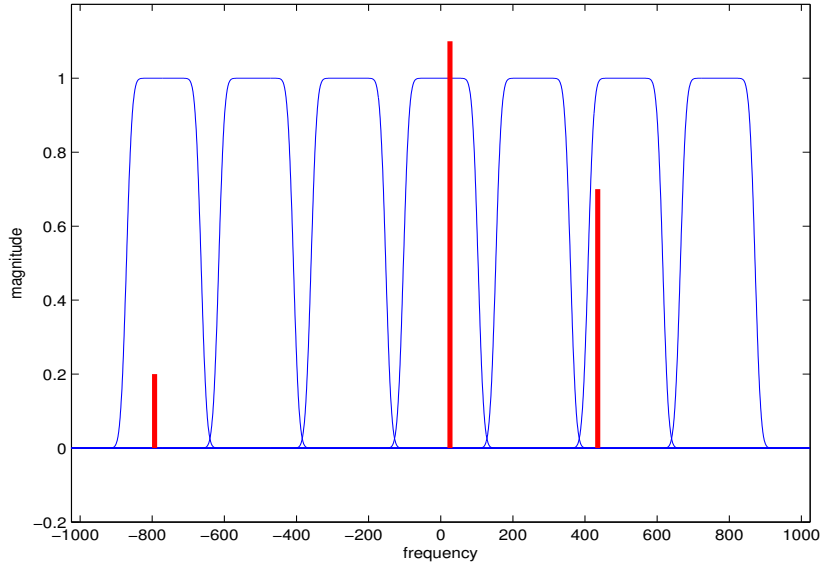
Compute $\widehat{x} * \widehat{G} = \widehat{(x \cdot G)}$

Sample complexity=supp G!

Reducing k-sparse recovery to 1-sparse recovery

Permute with a random linear transformation and phase shift





Choose a filter $G, \widehat{G}$ such that

- $\widehat{G} \approx$ rectangle
- G has support $\approx k$

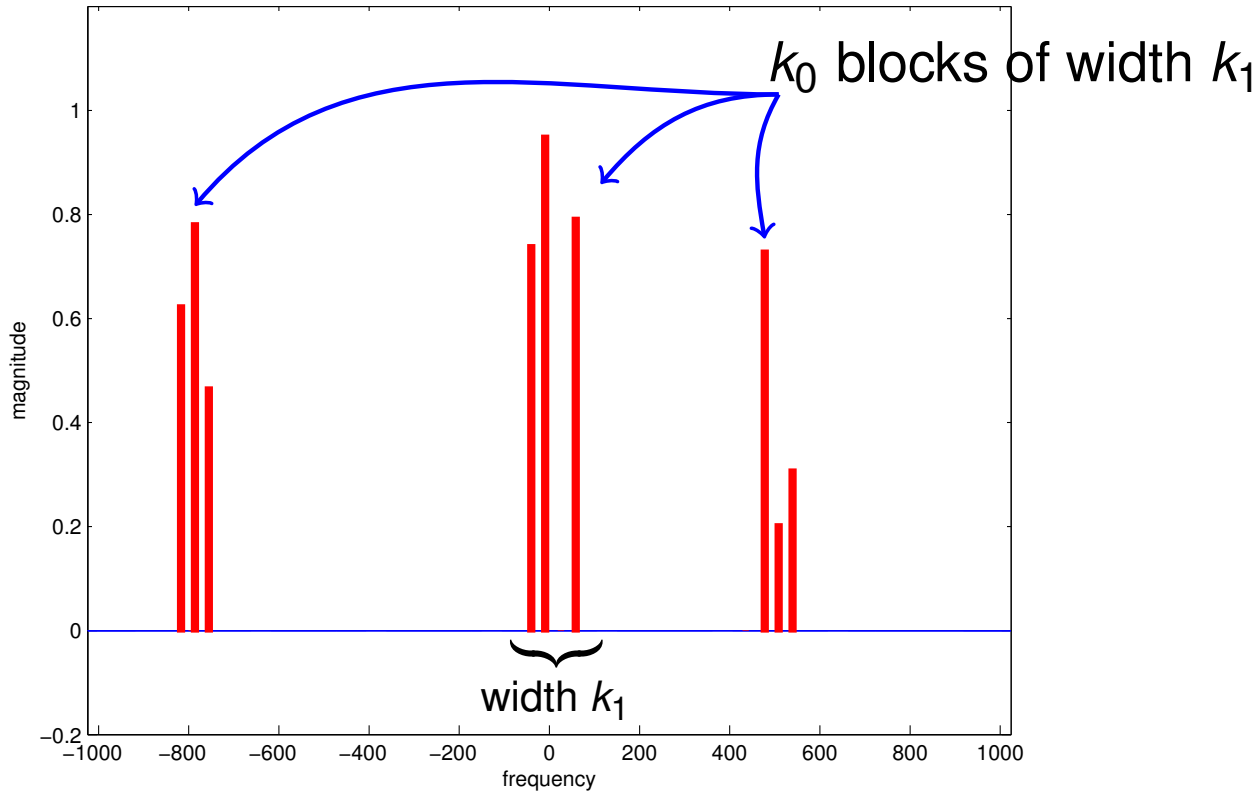
Compute $\widehat{x} * \widehat{G} = \widehat{(x \cdot G)}$

Sample complexity=supp G!

1. Sparse recovery from Fourier measurements

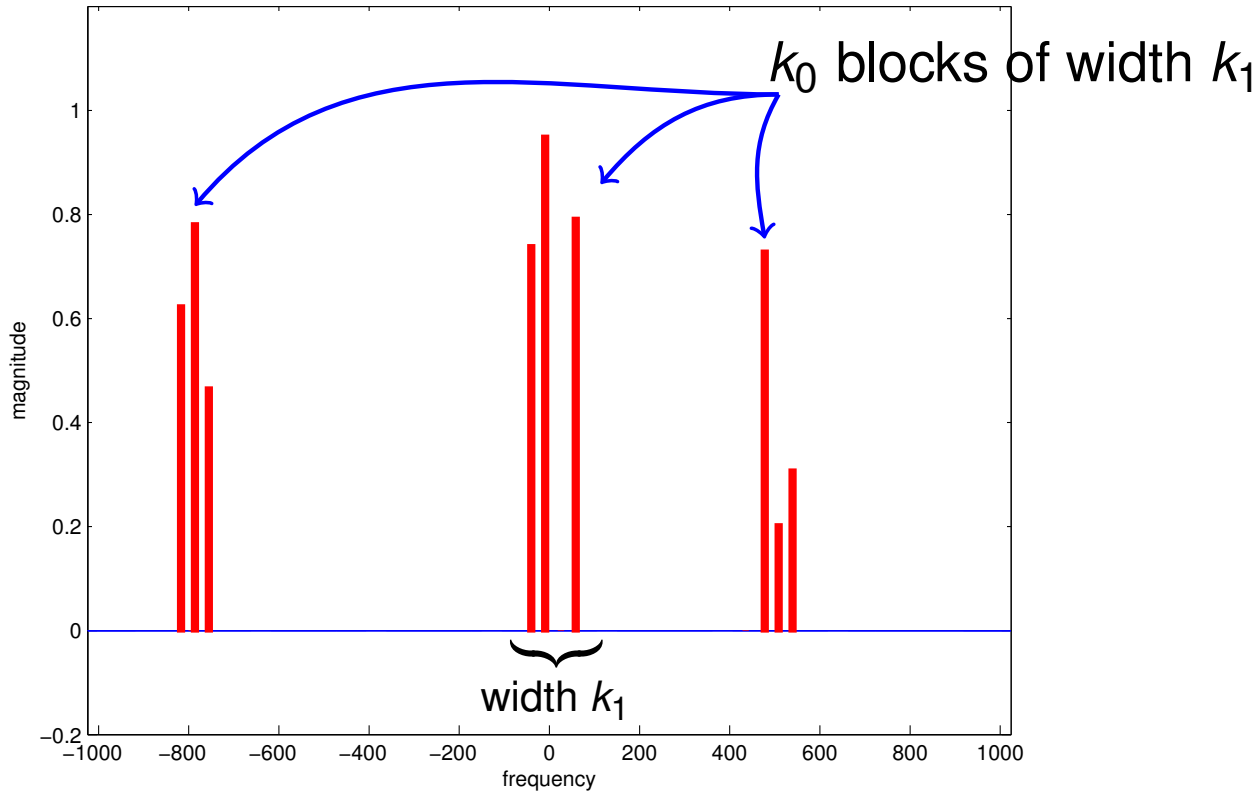
2. **Main result**

Block sparsity in Fourier domain



This work: suppose that dominant frequencies are contained in k_0 clusters (intervals) of length k_1 each?

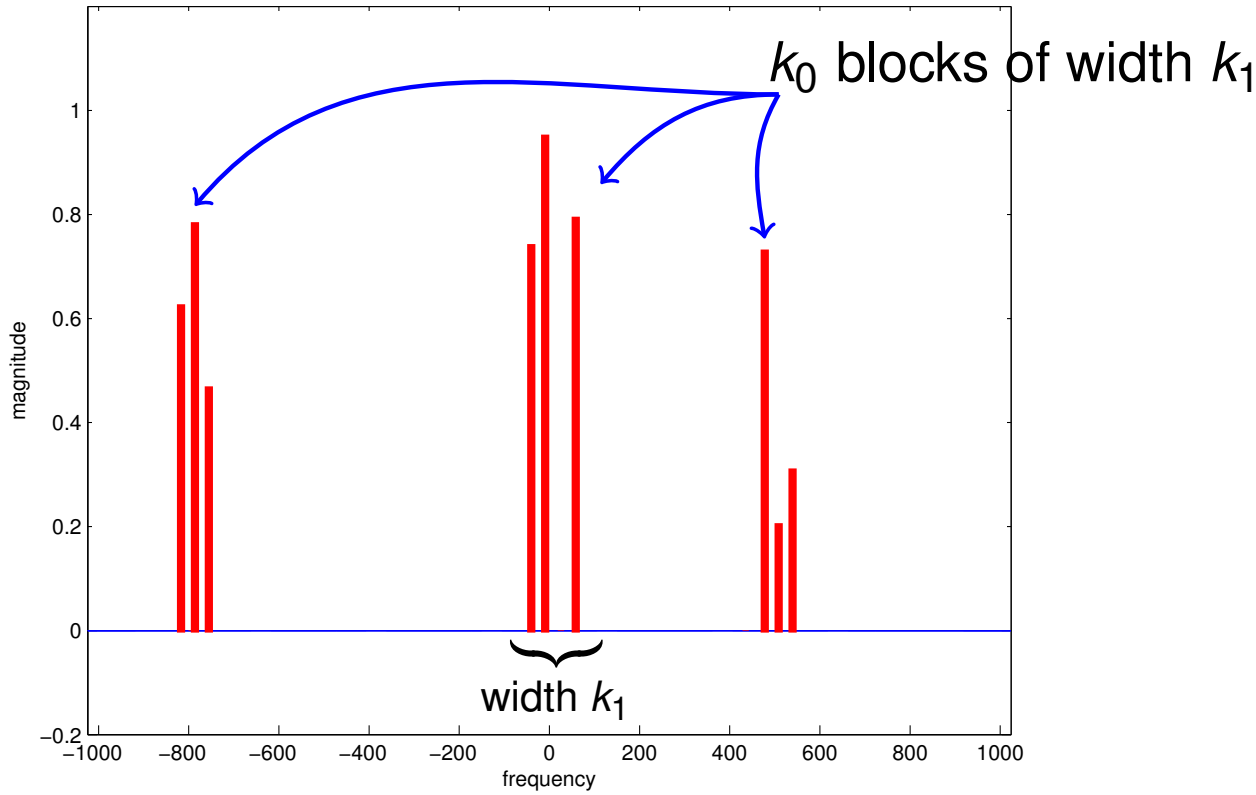
Block sparsity in Fourier domain



This work: suppose that dominant frequencies are contained in k_0 clusters (intervals) of length k_1 each?

Standard techniques destroy structure (by a random permutation), and need $\Omega(k_0 k_1 \log n)$ samples

Block sparsity in Fourier domain



This work: suppose that dominant frequencies are contained in k_0 clusters (intervals) of length k_1 each?

Standard techniques destroy structure (by a random permutation), and need $\Omega(k_0 k_1 \log n)$ samples

$O(k_0 \log n + k_0 k_1)$ sample complexity in sublinear time?

Main result

Theorem

There exists an adaptive (k_0, k_1)*-block sparse Fourier transform with sample complexity* $O^*(k_0 \log(1+k_0) \log n + k_0 k_1) \cdot \log SNR$ *samples and* $O^*(k_0 k_1 \log^3 n) \cdot \log SNR$ *runtime.*

Sample complexity optimal up to lower order terms for constant SNR

Main result

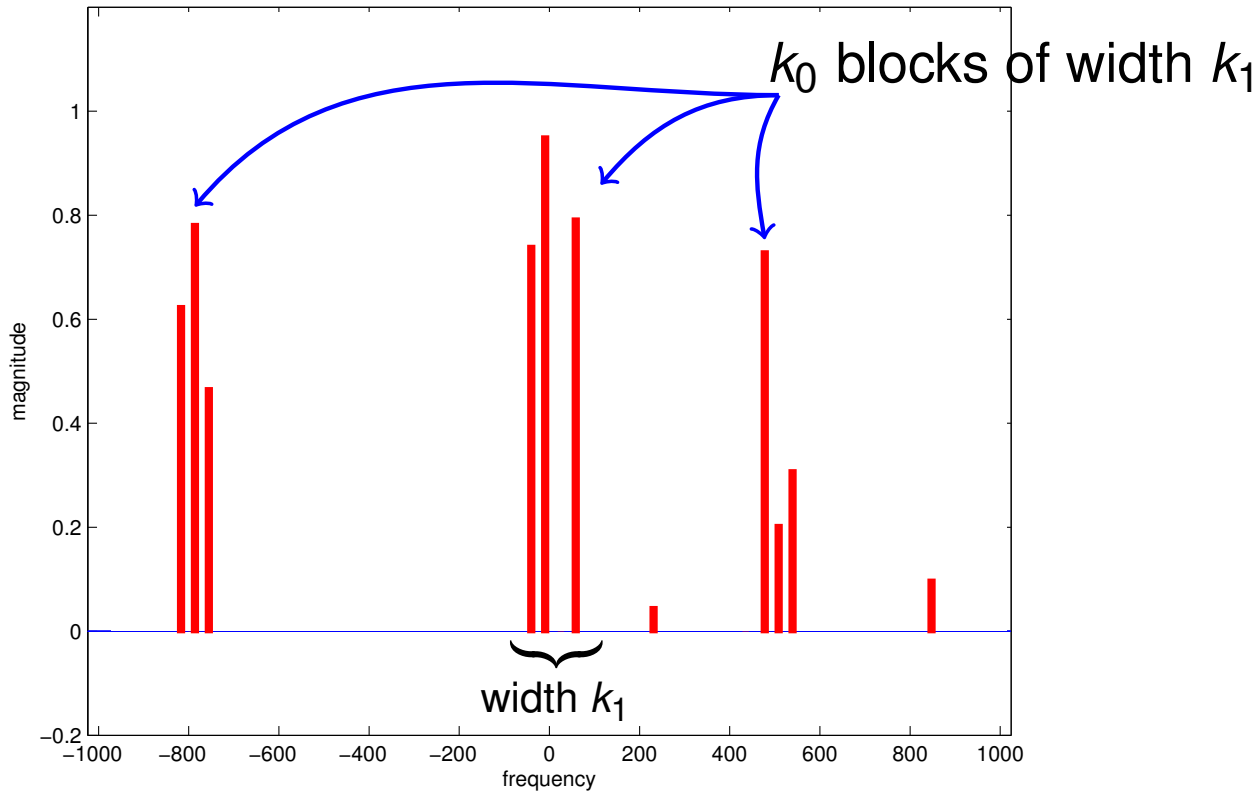
Theorem

There exists an adaptive (k_0, k_1)*-block sparse Fourier transform with sample complexity* $O^*(k_0 \log(1+k_0) \log n + k_0 k_1) \cdot \log SNR$ *samples and* $O^*(k_0 k_1 \log^3 n) \cdot \log SNR$ *runtime.*

Sample complexity optimal up to lower order terms for constant SNR

Crucially use adaptivity!

Spectrum is well approximated by k_0 blocks of length k_1:

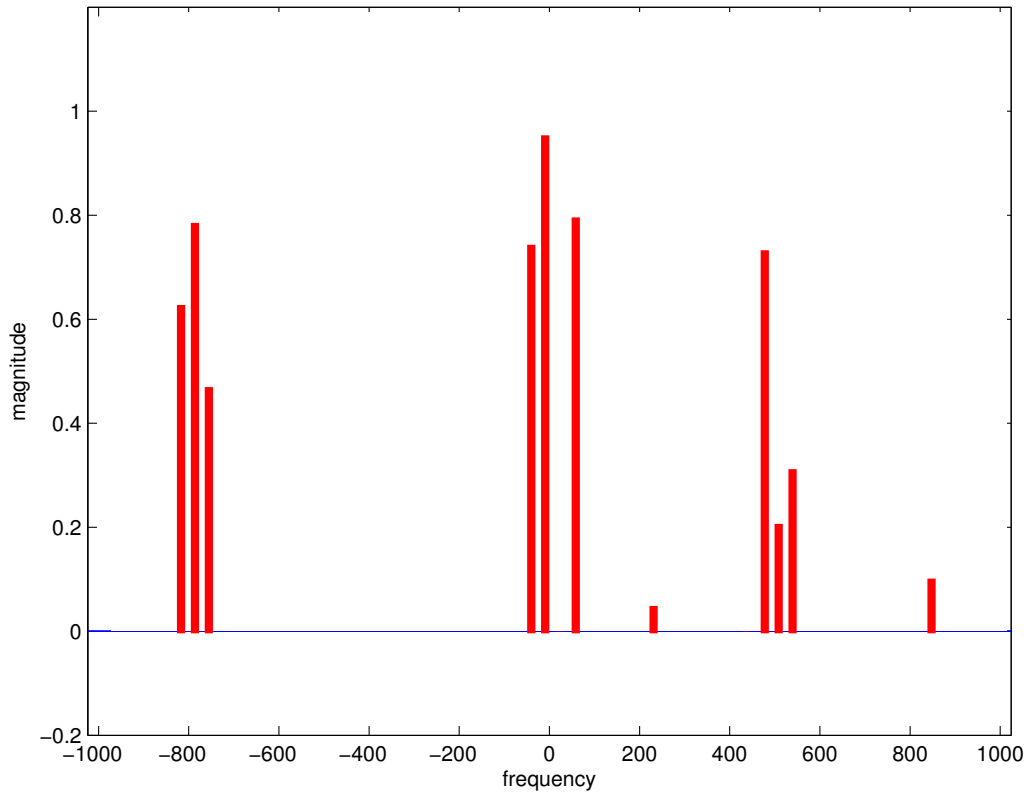


$k_0 = 3$ in the example

Natural idea: can we treat blocks as individual frequencies, and thus reduce to standard k_0-sparse recovery?

YES, to some extent...

Spectrum is well approximated by k_0 blocks of length k_1:

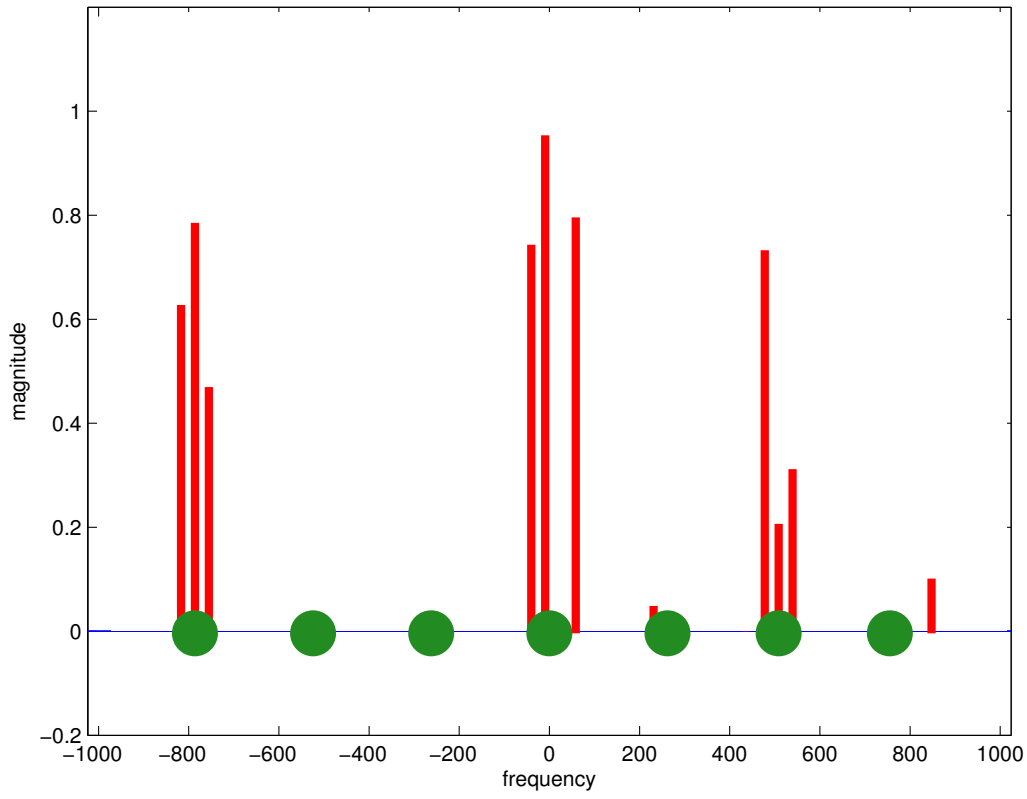


$k_0 = 3$ in the example

Natural idea: can we treat blocks as individual frequencies, and thus reduce to standard k_0-sparse recovery?

YES, to some extent...

Spectrum is well approximated by k_0 blocks of length k_1:

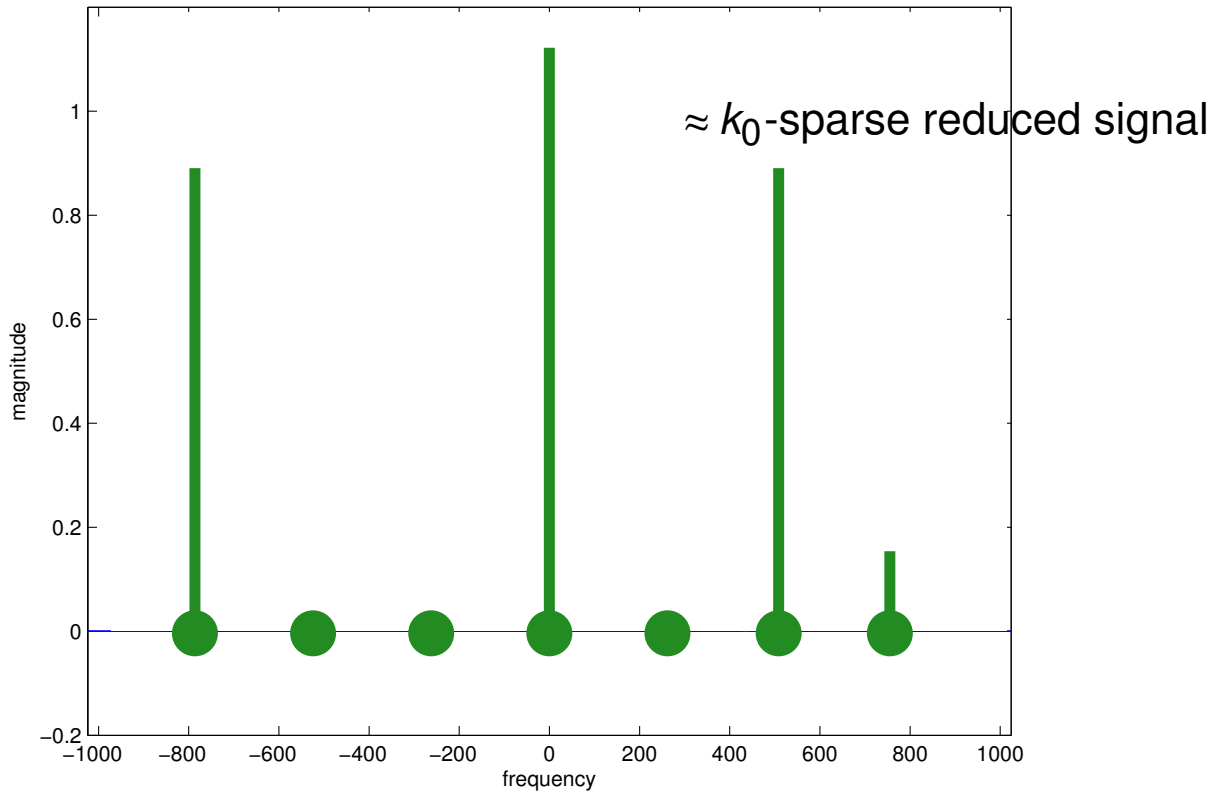


$k_0 = 3$ in the example

Natural idea: can we treat blocks as individual frequencies, and thus reduce to standard k_0-sparse recovery?

YES, to some extent...

Spectrum is well approximated by k_0 blocks of length k_1:

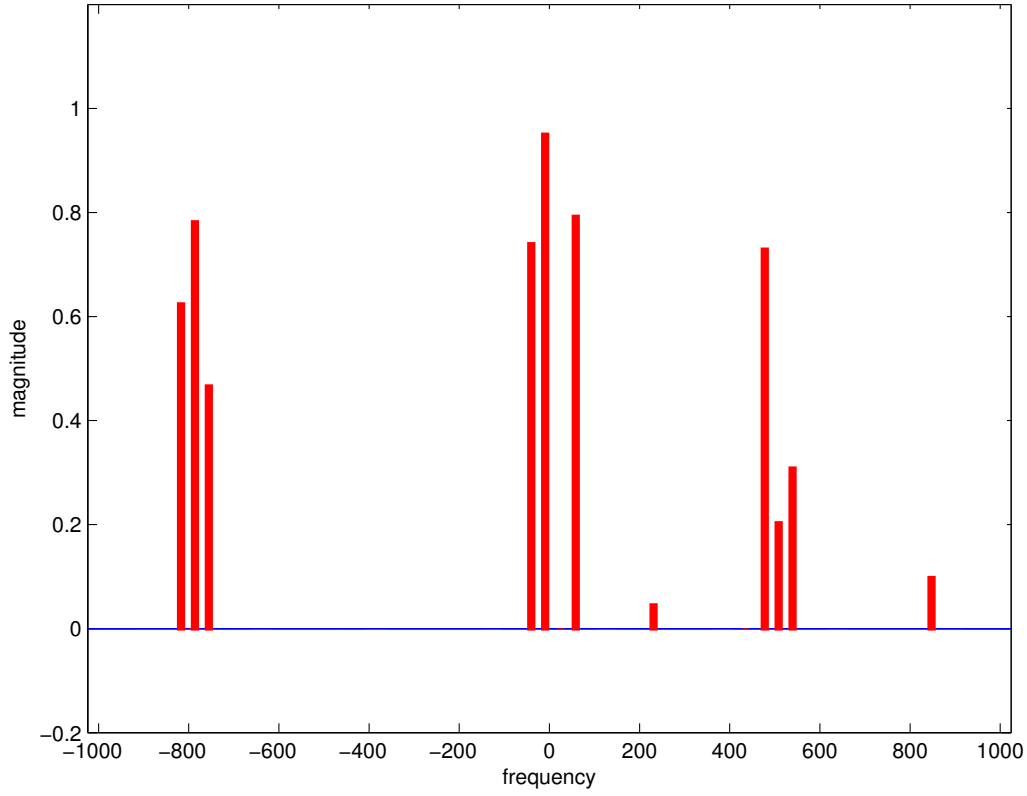


$k_0 = 3$ in the example

Natural idea: can we treat blocks as individual frequencies, and thus reduce to standard k_0-sparse recovery?

YES, to some extent...

Reduced signals

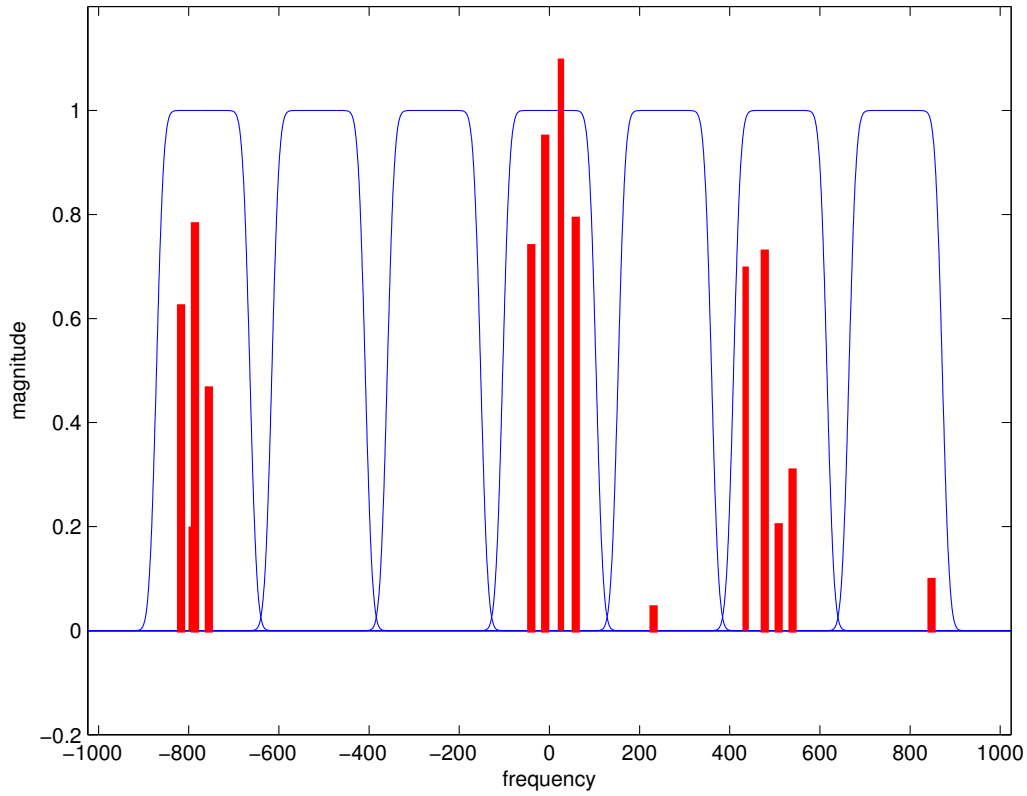


For each $j = 1, \ldots, n/k_1$ define reduced signal Z by

$$\widehat{Z}_j := (\widehat{X} * \widehat{G})_{j \cdot k_1}$$

(G has small support, and $\widehat{G} \approx$ indicator of a block)

Reduced signals

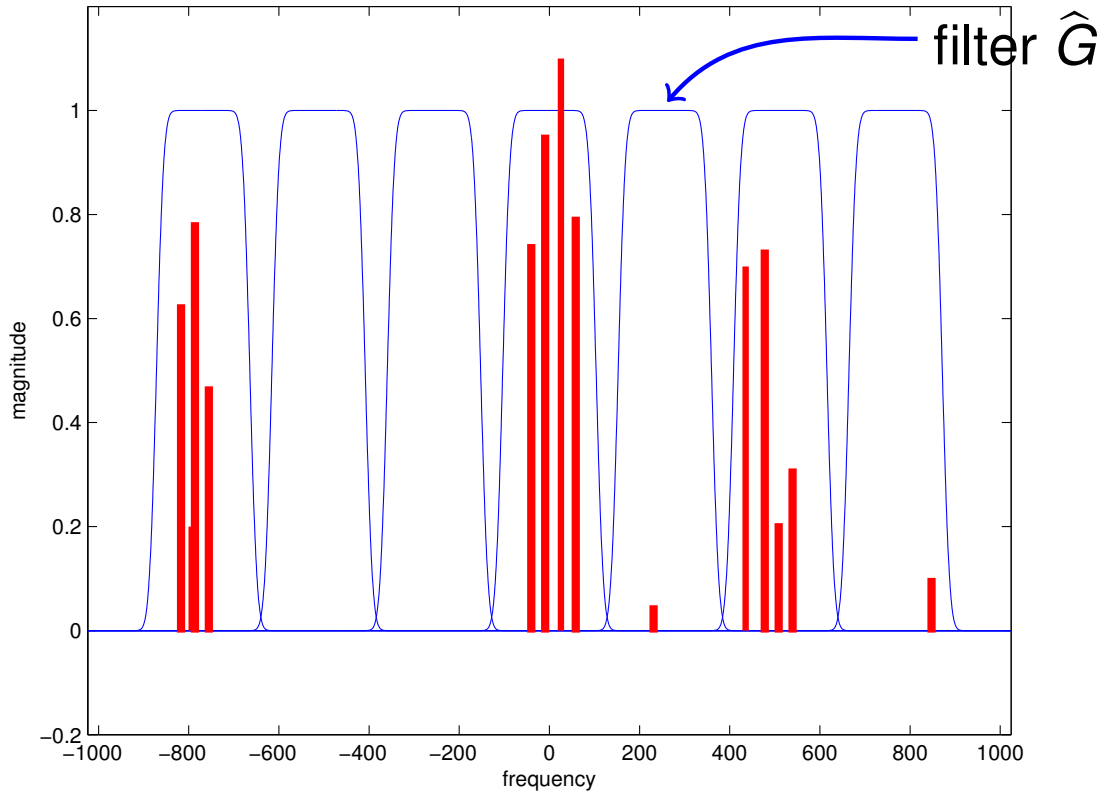


For each $j = 1, \ldots, n/k_1$ define reduced signal Z by

$$\widehat{Z}_j := (\widehat{X} * \widehat{G})_{j \cdot k_1}$$

(G has small support, and $\widehat{G} \approx$ indicator of a block)

Reduced signals

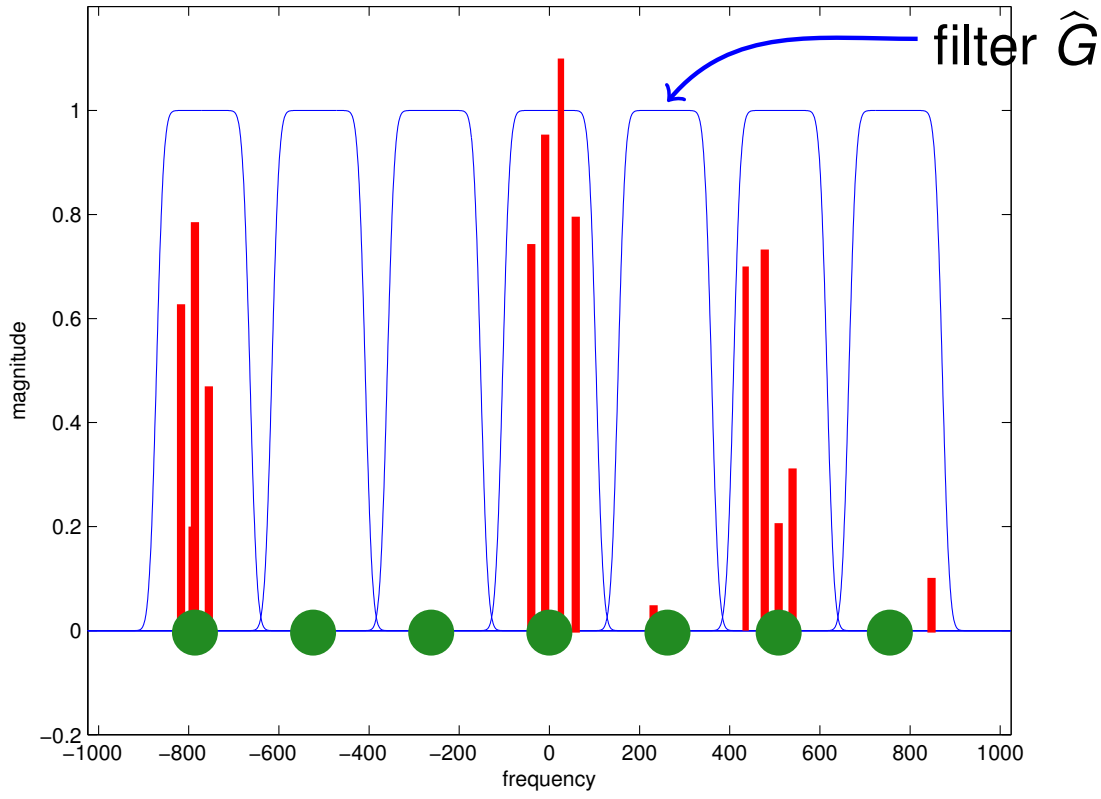


For each $j = 1, \ldots, n/k_1$ define reduced signal Z by

$$\widehat{Z}_j := (\widehat{X} * \widehat{G})_{j \cdot k_1}$$

(G has small support, and $\widehat{G} \approx$ indicator of a block)

Reduced signals

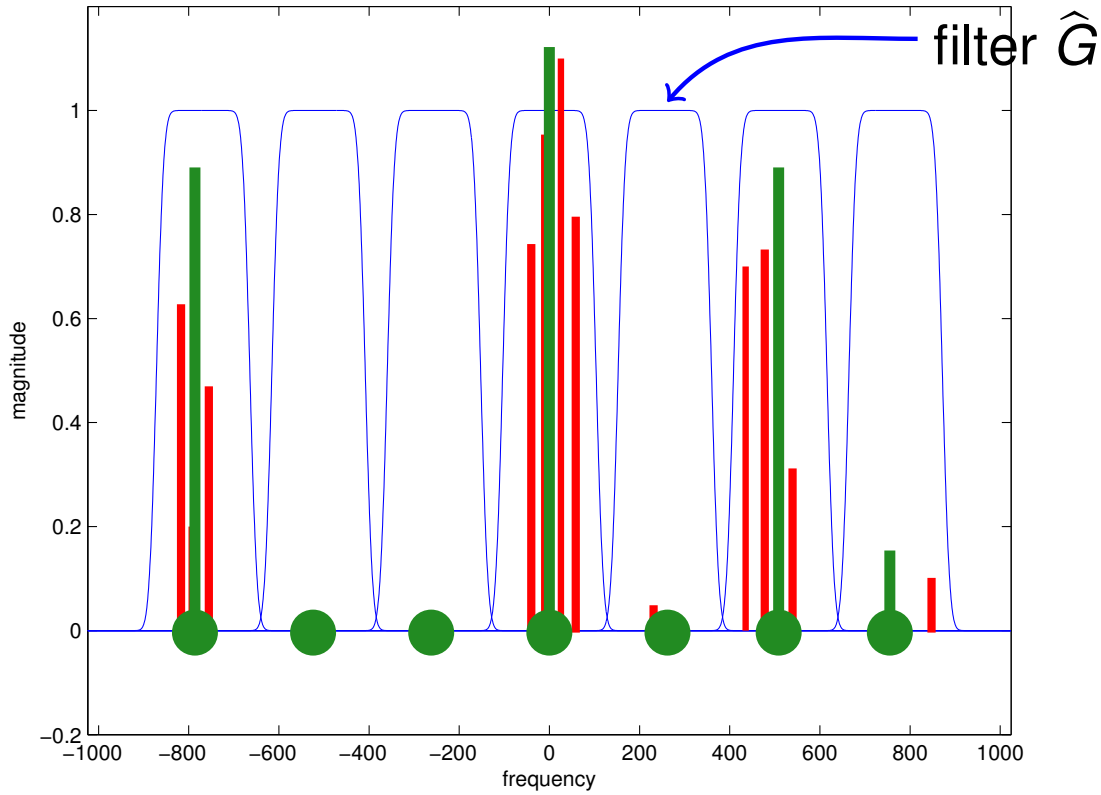


For each $j = 1, \ldots, n/k_1$ define reduced signal Z by

$$\widehat{Z}_j := (\widehat{X} * \widehat{G})_{j \cdot k_1}$$

(G has small support, and $\widehat{G} \approx$ indicator of a block)

Reduced signals

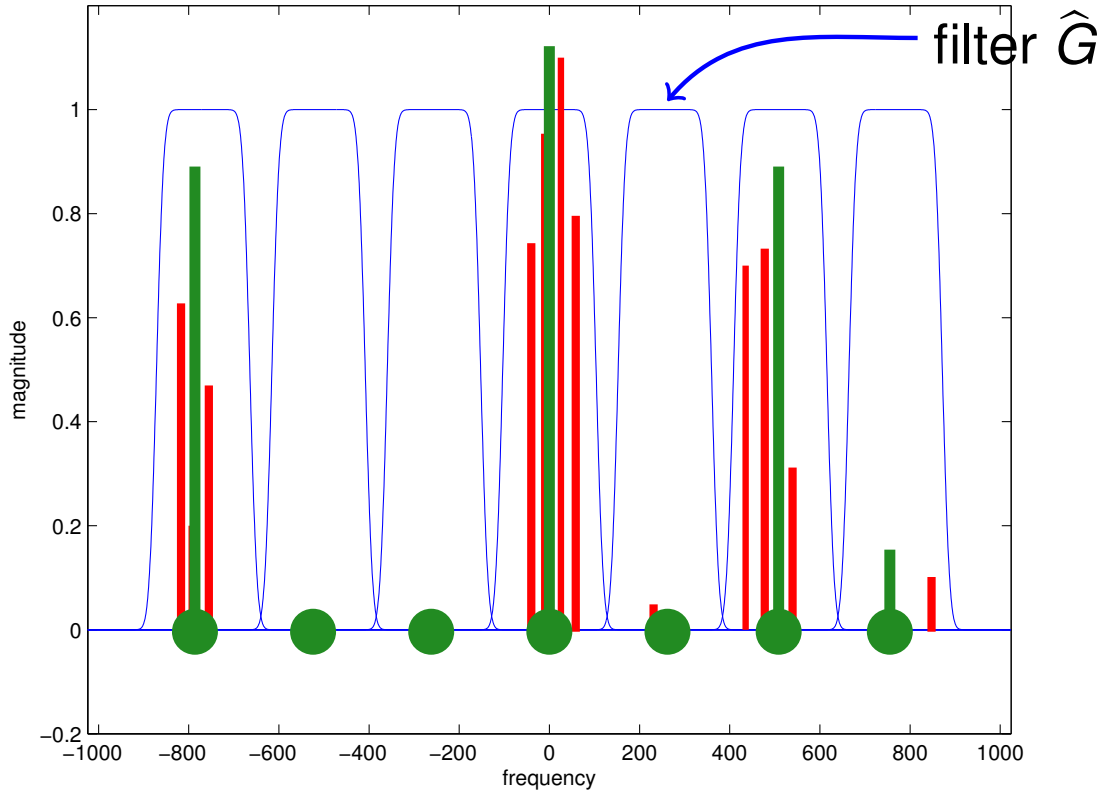


For each $j = 1, \ldots, n/k_1$ define reduced signal Z by

$$\widehat{Z}_j := (\widehat{X} * \widehat{G})_{j \cdot k_1}$$

Good news: if X is approximately (k_0, k_1)-block sparse, then Z is approximately $O(k_0)$-sparse.

Reduced signals



For each $j = 1, \ldots, n/k_1$ define reduced signal Z by

$$\widehat{Z}_j := (\widehat{X} * \widehat{G})_{j \cdot k_1}$$

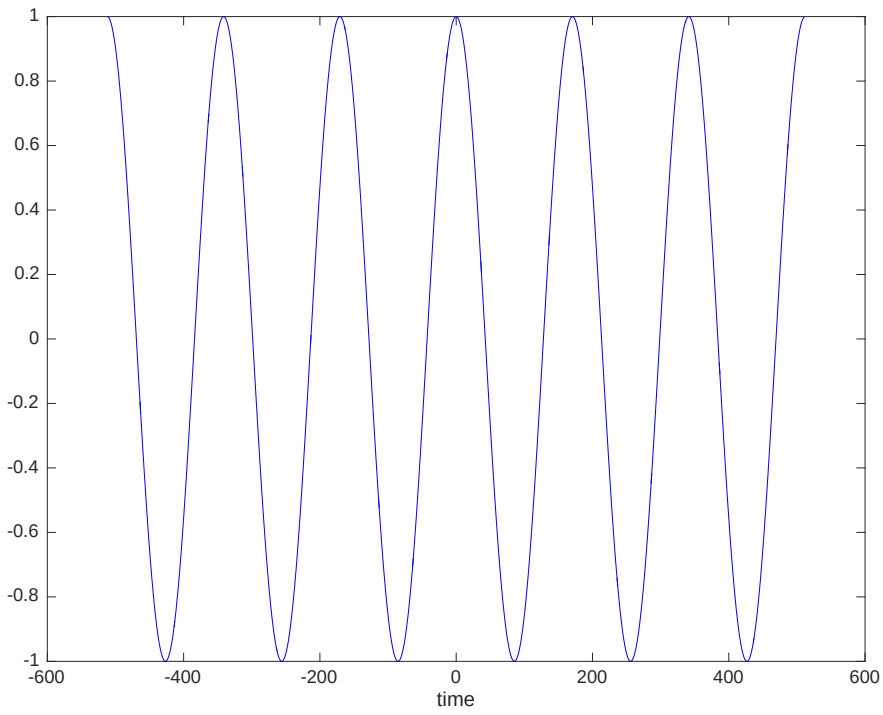
Major problem: some blocks of X might have essentially zero contribution to Z due to cancellations!

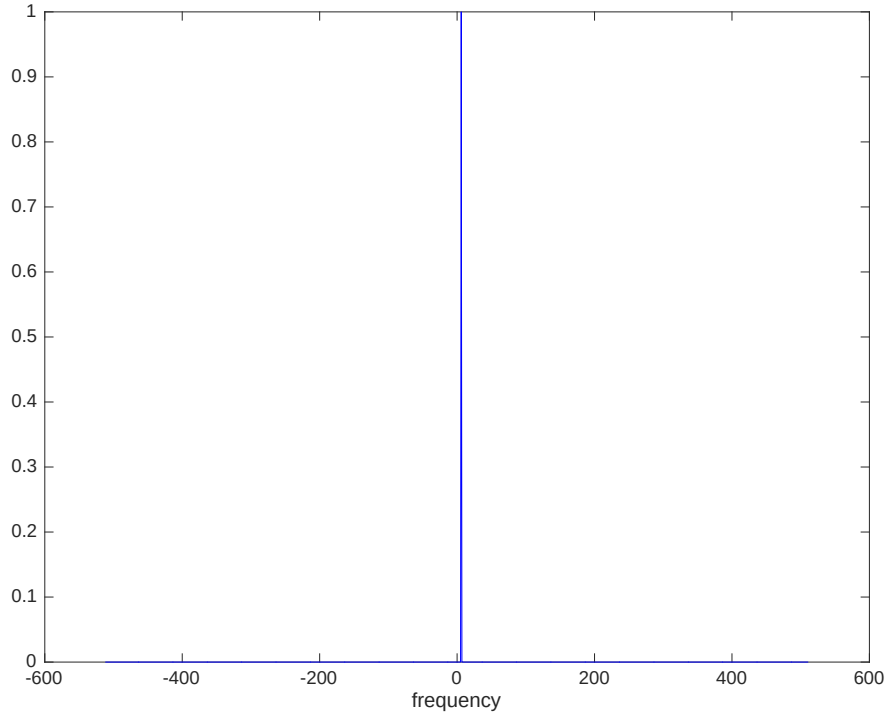
Energy concentration of 1-block sparse signals

In standard Sparse FFT $k_1 = 1$, so energy of a 'block' is uniformly spread over time.

Energy concentration of 1-block sparse signals

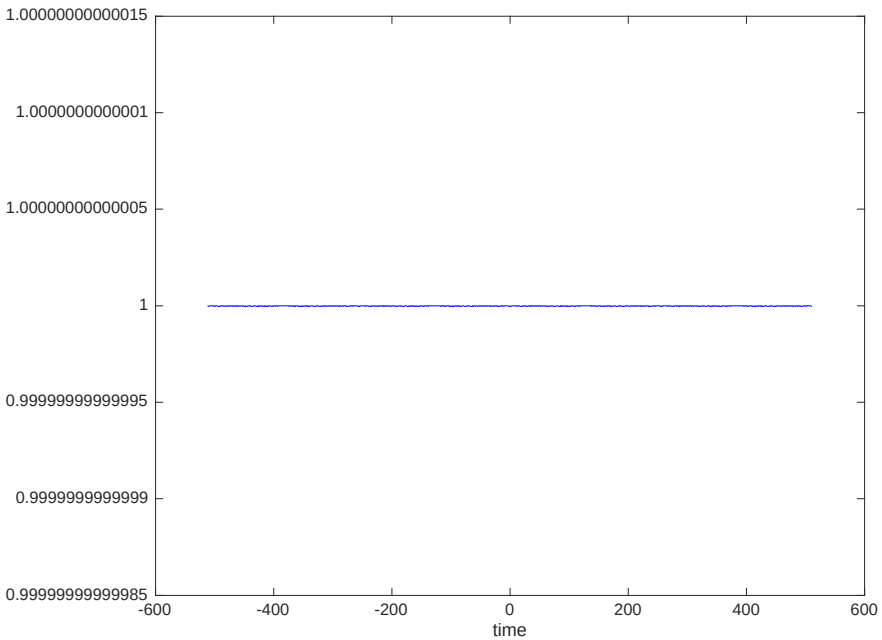
In standard Sparse FFT $k_1 = 1$, so energy of a 'block' is uniformly spread over time.

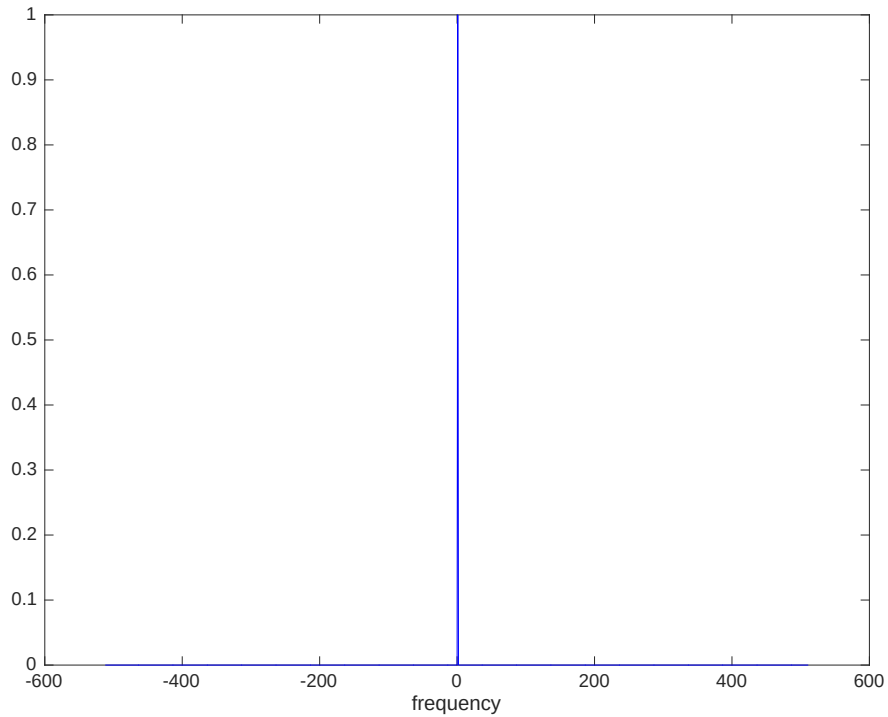




Energy concentration of 1-block sparse signals

In standard Sparse FFT $k_1 = 1$, so energy of a ‘block’ is uniformly spread over time.

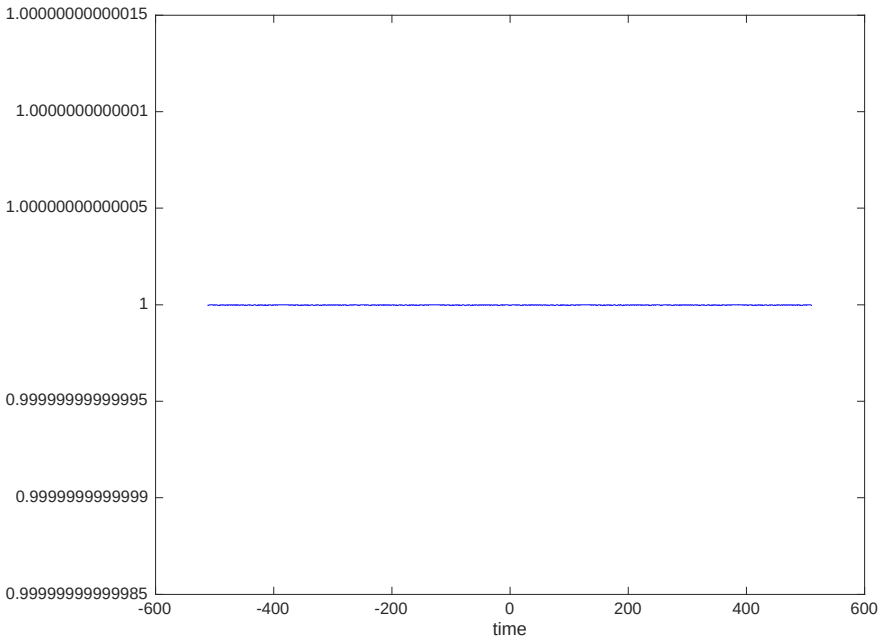


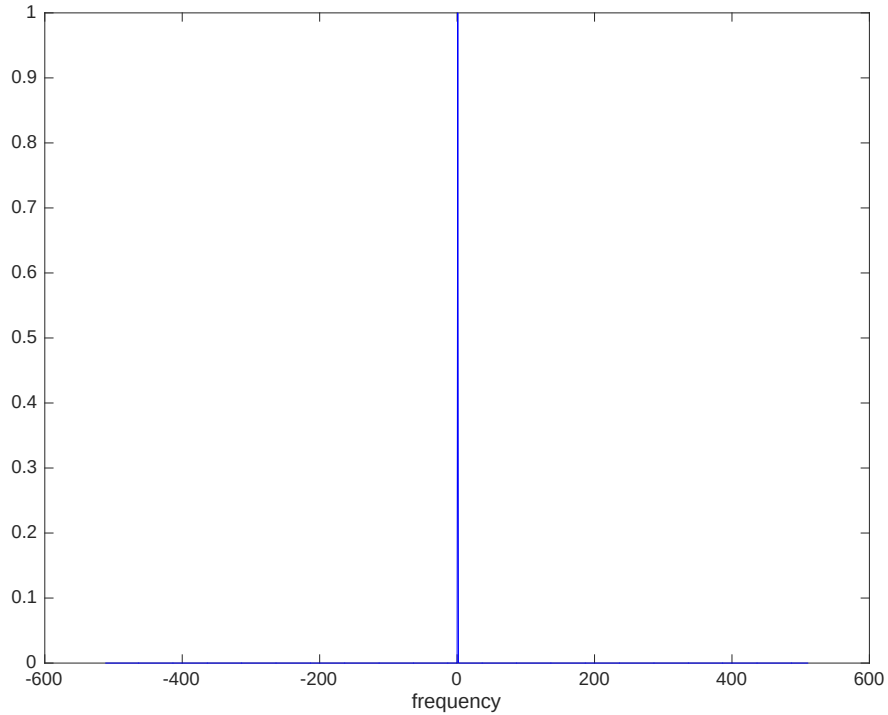


Energy concentration of 1-block sparse signals

In standard Sparse FFT $k_1 = 1$, so energy of a 'block' is uniformly spread over time.

If $\widehat{X}$ is 1-block sparse with block size k_1, its energy could be entirely concentrated on only a $1/k_1$ fraction of the time domain!

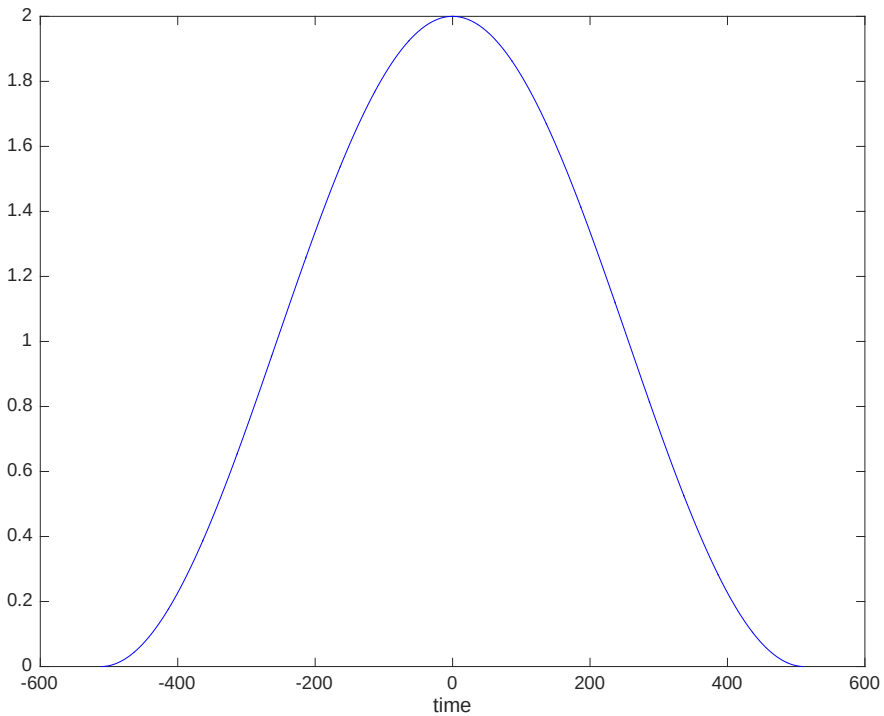


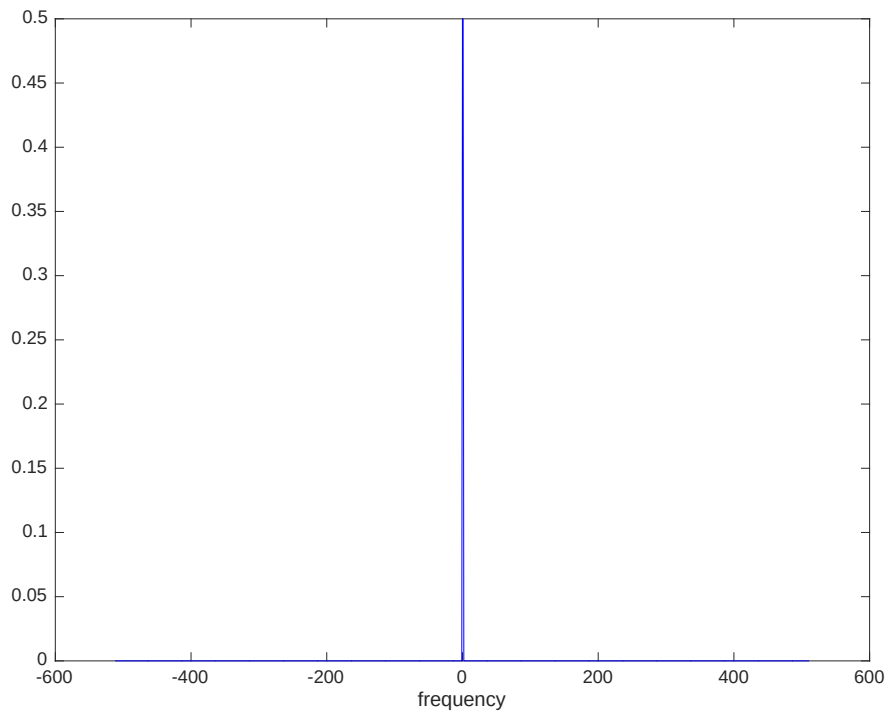


Energy concentration of 1-block sparse signals

In standard Sparse FFT $k_1 = 1$, so energy of a 'block' is uniformly spread over time.

If $\widehat{X}$ is 1-block sparse with block size k_1, its energy could be entirely concentrated on only a $1/k_1$ fraction of the time domain!

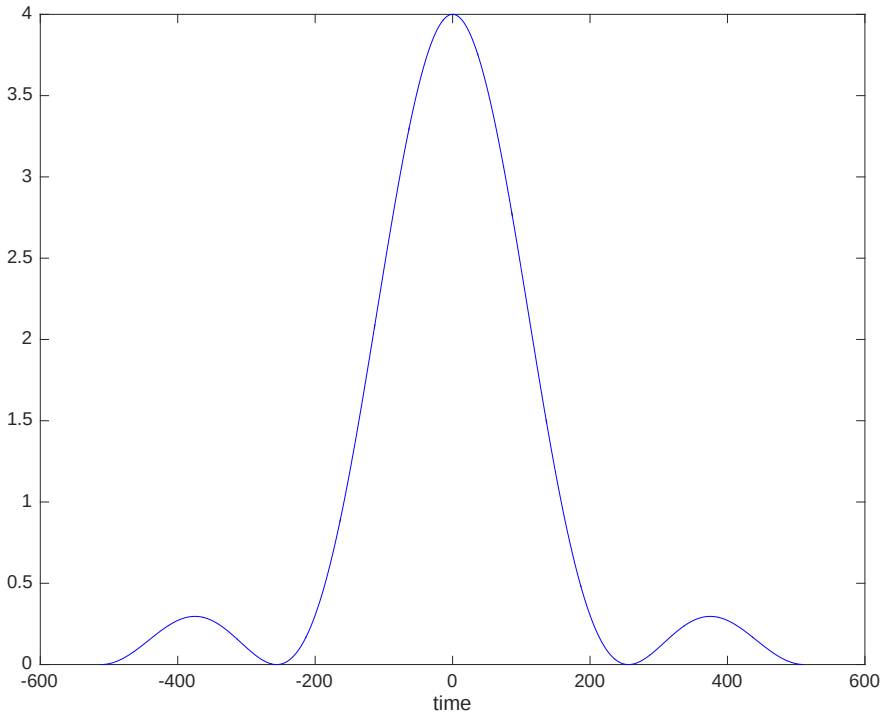


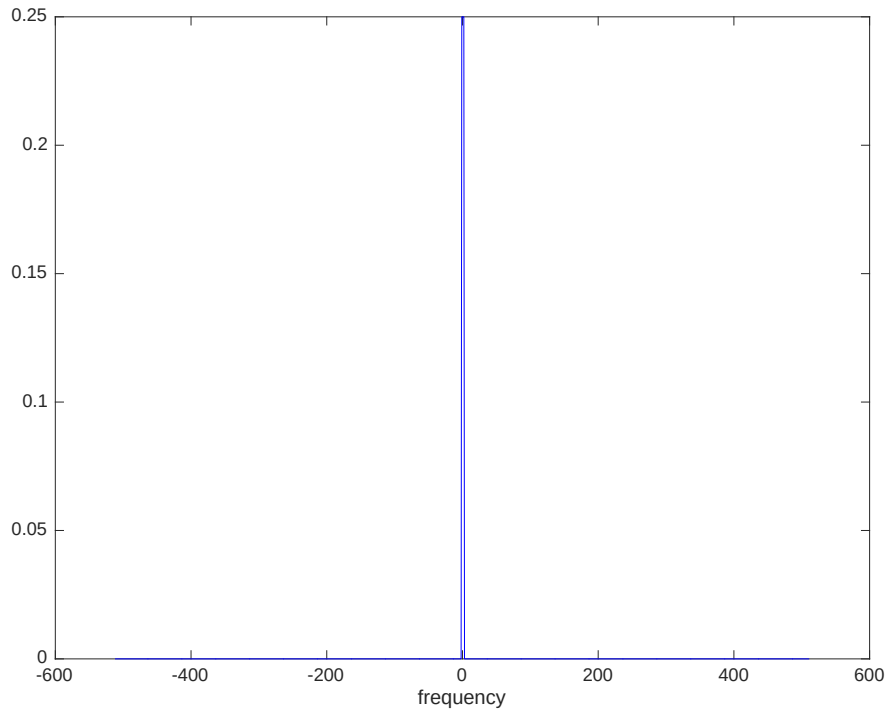


Energy concentration of 1-block sparse signals

In standard Sparse FFT $k_1 = 1$, so energy of a 'block' is uniformly spread over time.

If $\widehat{X}$ is 1-block sparse with block size k_1, its energy could be entirely concentrated on only a $1/k_1$ fraction of the time domain!

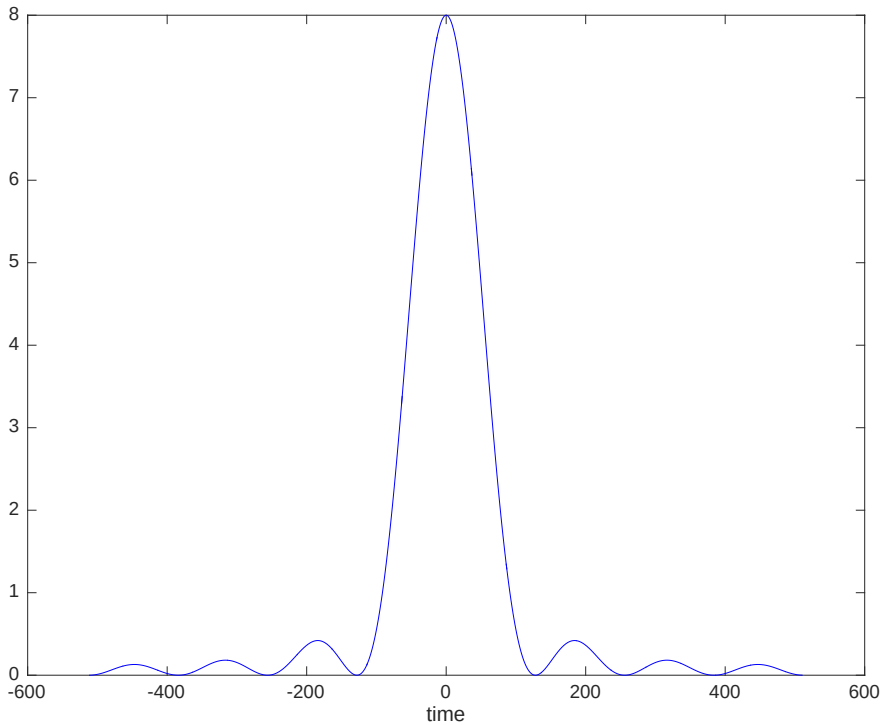


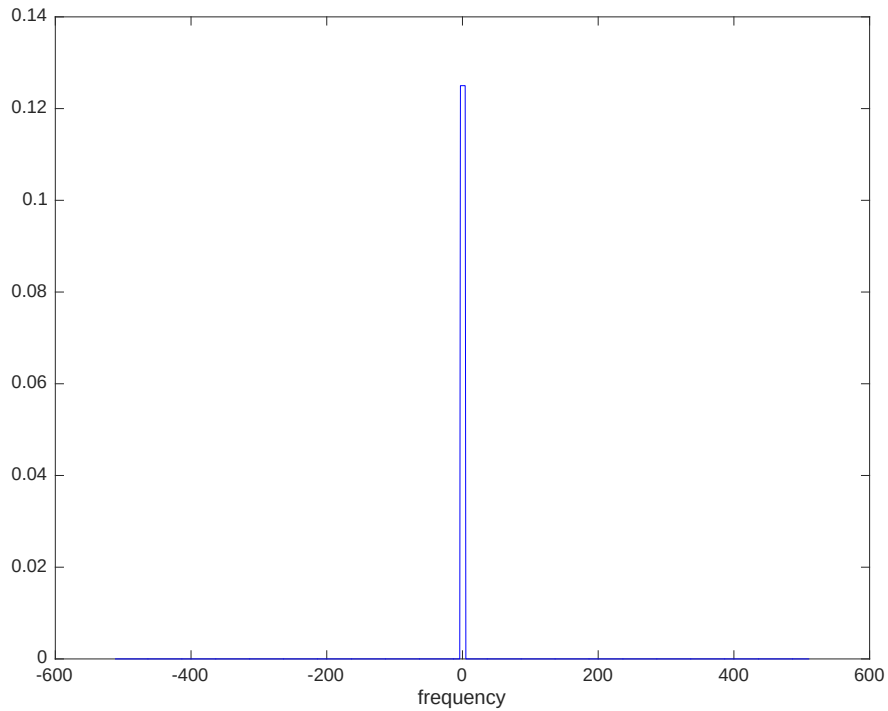


Energy concentration of 1-block sparse signals

In standard Sparse FFT $k_1 = 1$, so energy of a 'block' is uniformly spread over time.

If $\widehat{X}$ is 1-block sparse with block size k_1, its energy could be entirely concentrated on only a $1/k_1$ fraction of the time domain!

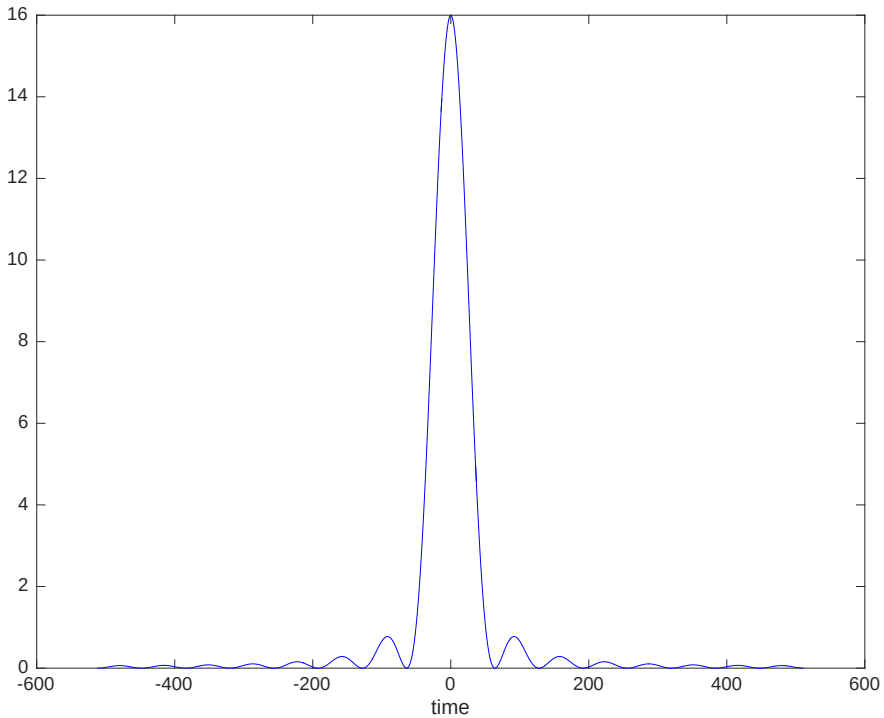


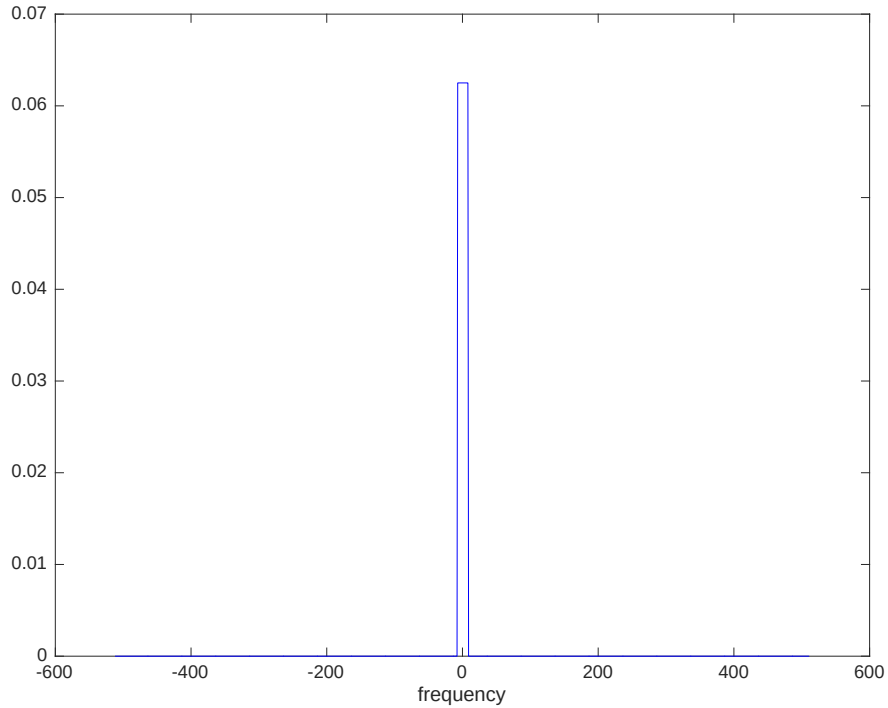


Energy concentration of 1-block sparse signals

In standard Sparse FFT $k_1 = 1$, so energy of a 'block' is uniformly spread over time.

If $\widehat{X}$ is 1-block sparse with block size k_1, its energy could be entirely concentrated on only a $1/k_1$ fraction of the time domain!

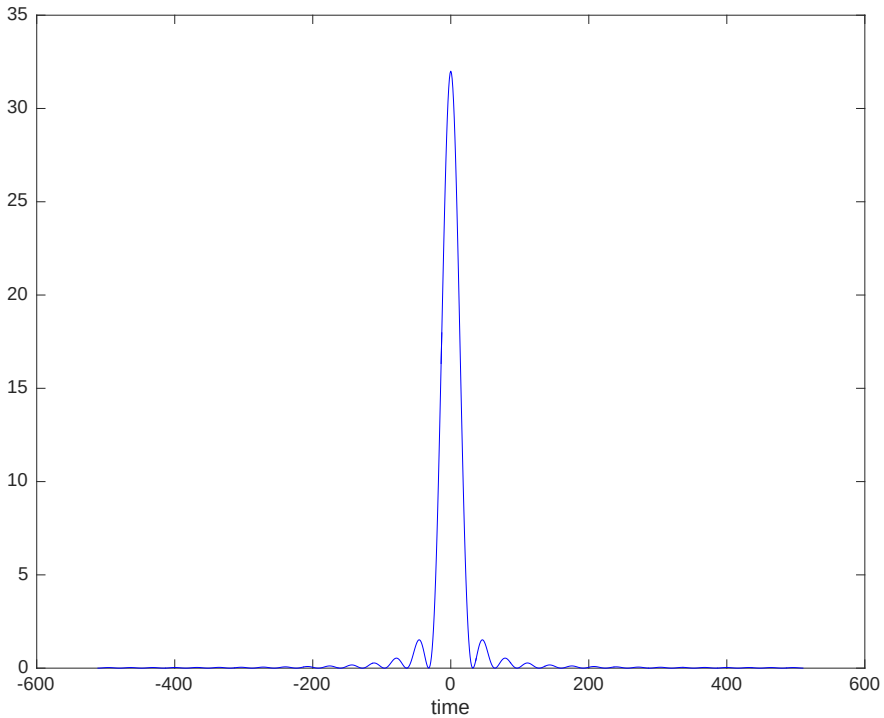


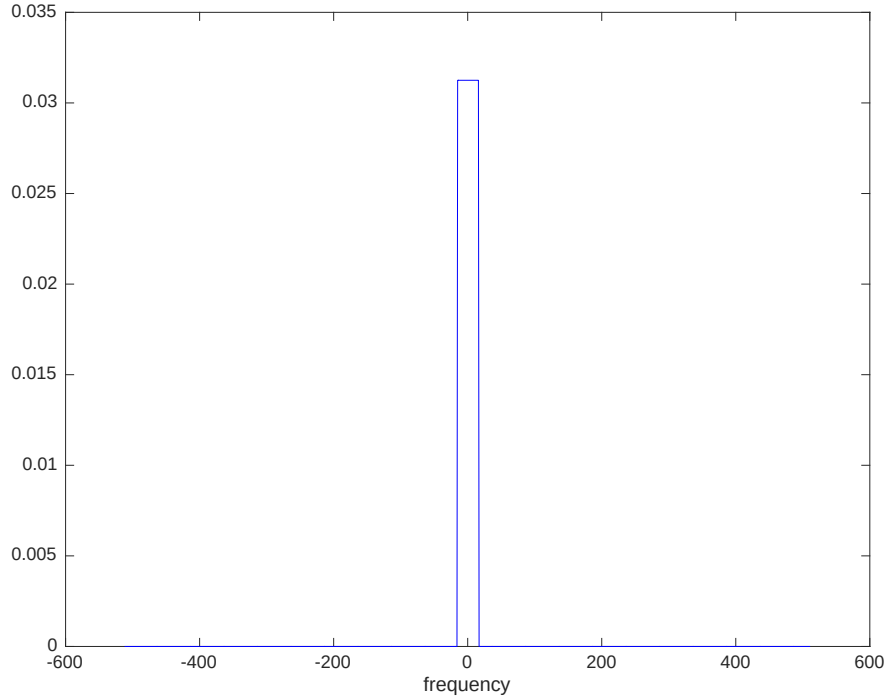


Energy concentration of 1-block sparse signals

In standard Sparse FFT $k_1 = 1$, so energy of a 'block' is uniformly spread over time.

If $\hat{X}$ is 1-block sparse with block size k_1, its energy could be entirely concentrated on only a $1/k_1$ fraction of the time domain!





Recovery of block sparse signals

For each $r = 1, \ldots, 2k_1$ define reduced signal $Z^{\mathbf{r}}$ by

$$\widehat{Z^{\mathbf{r}}_j} := \left(X \widehat{\left(\cdot - r \cdot \frac{n}{2k_1} \right)} * \widehat{G} \right)_{j \cdot k_1}$$

(Shift X first, use $2k_1$ regular shifts)

Recovery of block sparse signals

For each $r = 1, \ldots, 2k_1$ define reduced signal $Z^{\mathbf{r}}$ by

$$\widehat{Z^{\mathbf{r}}_j} := \left(X \widehat{\left(\cdot - r \cdot \frac{n}{2k_1} \right)} * \widehat{G} \right)_{j \cdot k_1}$$

(Shift X first, use $2k_1$ regular shifts)

If each reduction requires $k_0 \log n$ samples, we are back to the $k_0 k_1 \log n$ bound?

Recovery of block sparse signals

For each $r = 1, \ldots, 2k_1$ define reduced signal $Z^{\mathbf{r}}$ by

$$\widehat{Z^{\mathbf{r}}_j} := \left(X \widehat{\left(\cdot - r \cdot \frac{n}{2k_1} \right)} * \widehat{G} \right)_{j \cdot k_1}$$

(Shift X first, use $2k_1$ regular shifts)

If each reduction requires $k_0 \log n$ samples, we are back to the $k_0 k_1 \log n$ bound?

An adaptive solution can first probe which signals carry most energy, then allocate hashing budgets carefully

Importance sampling

AllocateBudgets(X, k_0, k_1)
For $t = 1, \ldots, O(k_0)$
  Sample $r \sim ||Z^r||_2^2$, $q \sim Geom$
  Set $s^t \leftarrow (r, 2^q)$
End For

$O(k_0 k_1)$ samples

Importance sampling

AllocateBudgets(X, k_0, k_1)
For $t = 1, \ldots, O(k_0)$
  Sample $r \sim ||Z^r||_2^2$, $q \sim Geom$
  Set $s^t \leftarrow (r, 2^q)$
End For

$O(k_0 k_1)$ samples

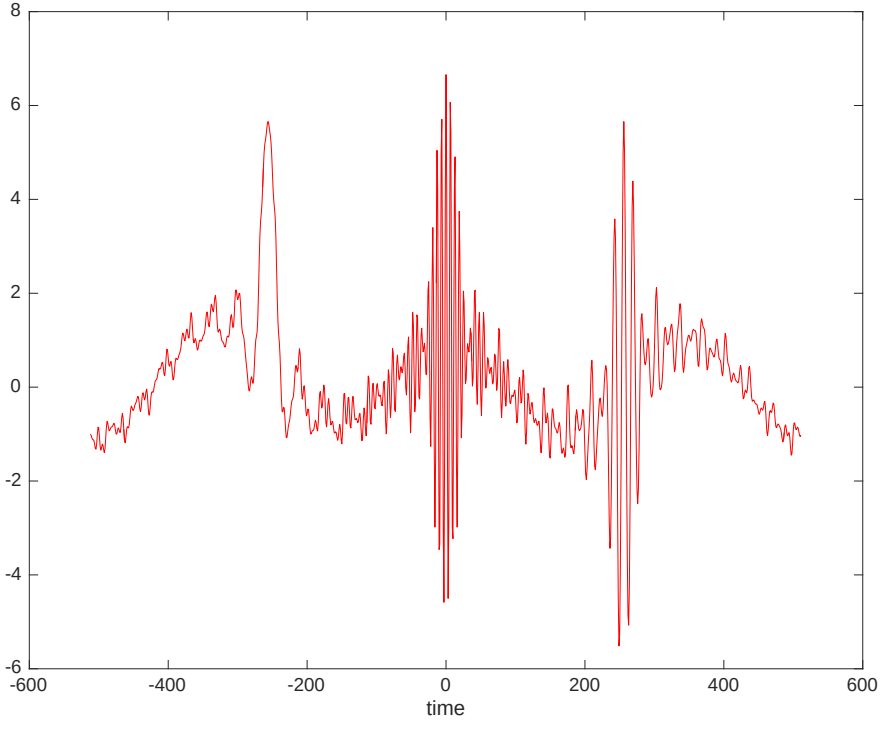
BudgetedRecovery$(X, \{s^t\})$
For $t = 1, \ldots, O(k_0)$
  Hash Z^r into $\approx s^t$ buckets,
  Recover top s^t frequencies
End For

$O(k_0 \log(1 + k_0) \log n)$ samples

Importance sampling

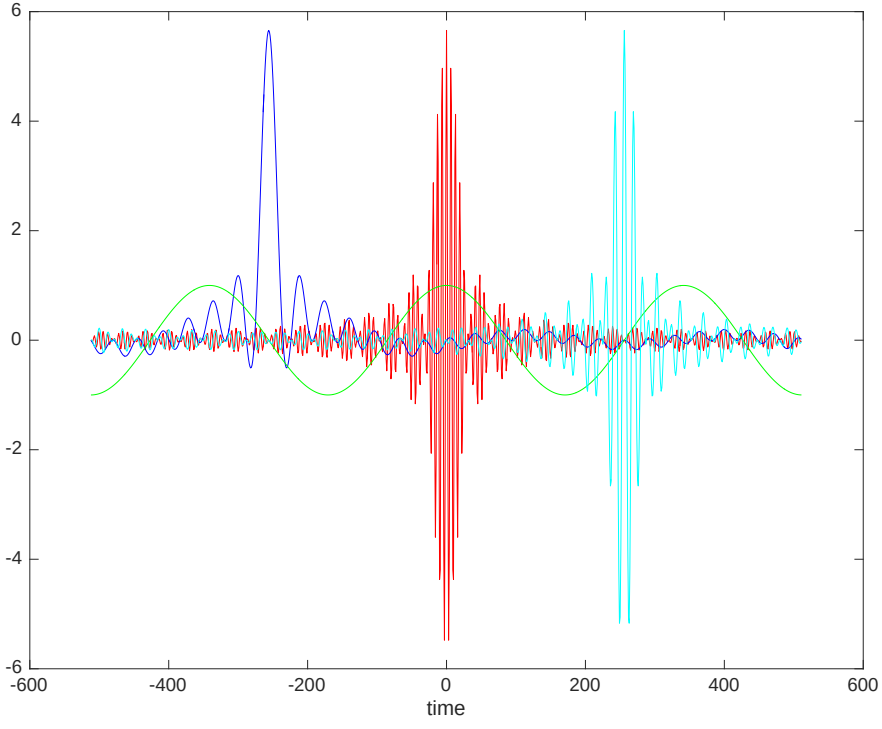
AllocateBudgets(X, k_0, k_1)
For $t = 1, \ldots, O(k_0)$
  Sample $r \sim ||Z^r||_2^2$, $q \sim Geom$
  Set $s^t \leftarrow (r, 2^q)$
End For

$O(k_0 k_1)$ samples



BudgetedRecovery$(X, \{s^t\})$
For $t = 1, \ldots, O(k_0)$
  Hash Z^r into $\approx s^t$ buckets,
  Recover top s^t frequencies
End For

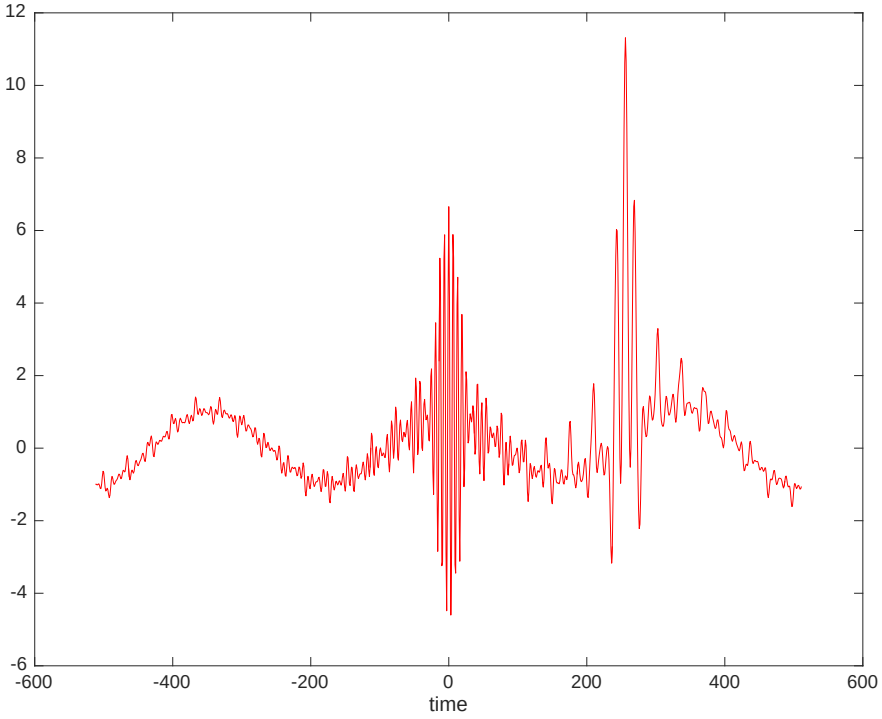
$O(k_0 \log(1 + k_0) \log n)$ samples



Importance sampling

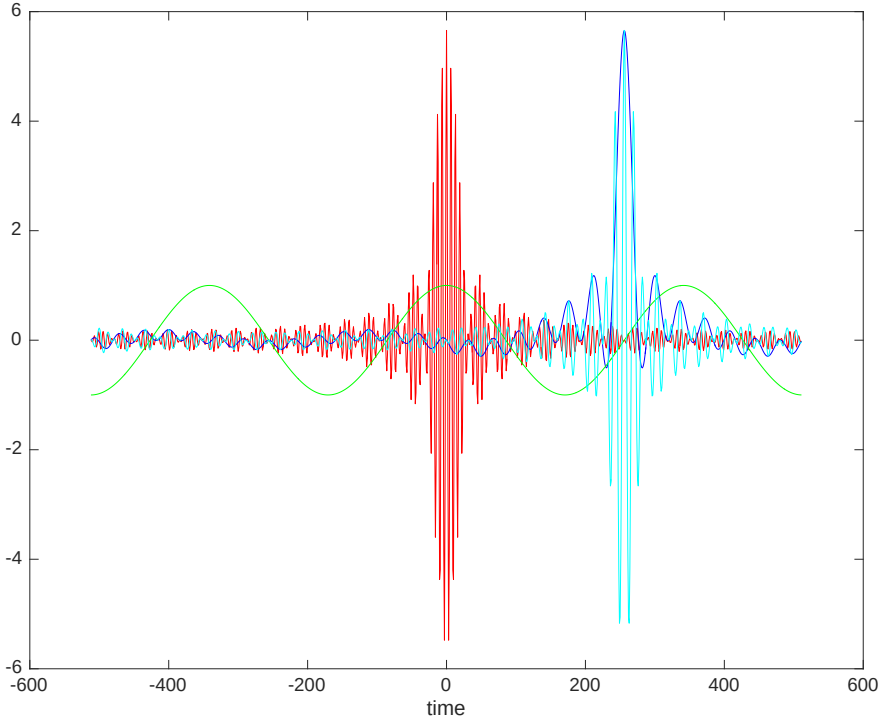
AllocateBudgets(X, k_0, k_1)
For $t = 1, \ldots, O(k_0)$
  Sample $r \sim \|Z^r\|_2^2$, $q \sim Geom$
  Set $s^t \leftarrow (r, 2^q)$
End For

$O(k_0 k_1)$ samples



BudgetedRecovery$(X, \{s^t\})$
For $t = 1, \ldots, O(k_0)$
  Hash Z^r into $\approx s^t$ buckets,
  Recover top s^t frequencies
End For

$O(k_0 \log(1 + k_0) \log n)$ samples



Adaptivity is necessary

Algorithm crucially uses adaptivity: sample a few points first to find a non-uniform sampling distribution for second stage

Theorem

Any non-adaptive Sparse FFT algorithm must take $\Omega(k_0 k_1 \log \frac{n}{k_0 k_1})$ *samples.*

Our results:

- first Sparse FFT algorithm that exploits structure beyond sparsity
- adaptivity is crucial

Our results:

- first Sparse FFT algorithm that exploits structure beyond sparsity
- adaptivity is crucial

Other forms of structured sparsity (e.g. tree sparsity, graph sparsity?)

Our results:

- first Sparse FFT algorithm that exploits structure beyond sparsity
- adaptivity is crucial

Other forms of structured sparsity (e.g. tree sparsity, graph sparsity?)

A new class of Sparse FFT techniques that adapt to structure of the spectrum?

Our results:

- first Sparse FFT algorithm that exploits structure beyond sparsity
- adaptivity is crucial

Other forms of structured sparsity (e.g. tree sparsity, graph sparsity?)

A new class of Sparse FFT techniques that adapt to structure of the spectrum?

Experimental evaluation?

Our results:

- first Sparse FFT algorithm that exploits structure beyond sparsity
- adaptivity is crucial

Other forms of structured sparsity (e.g. tree sparsity, graph sparsity?)

A new class of Sparse FFT techniques that adapt to structure of the spectrum?

Experimental evaluation?

Thank you!