Efficiently Learning Fourier Sparse Set Functions

Andisheh Amrollahi*, Amir Zandieh*, Michael Kapralov, Andreas Krause

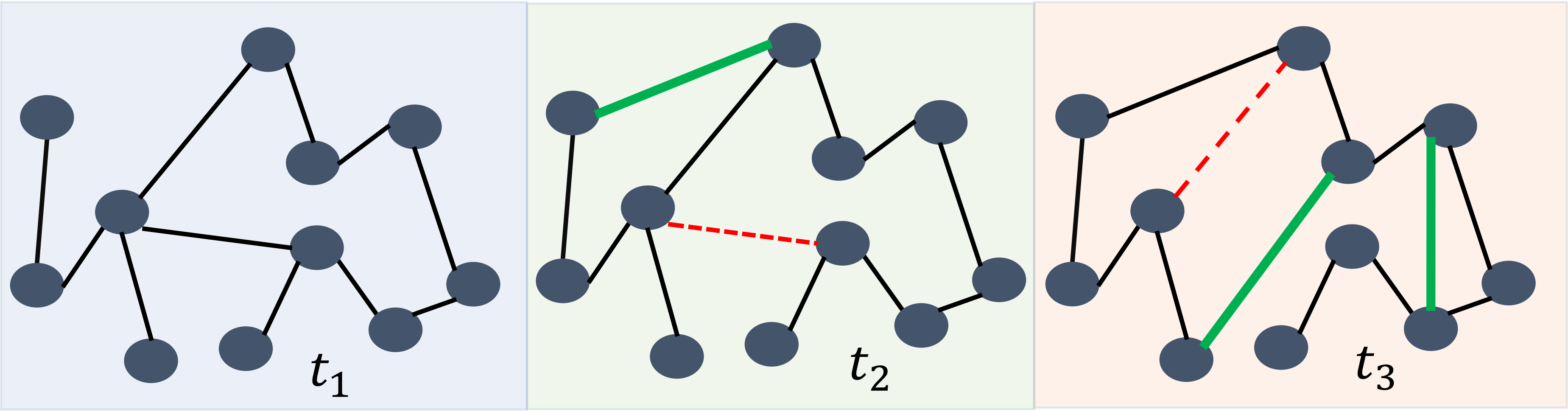



ETH zürich

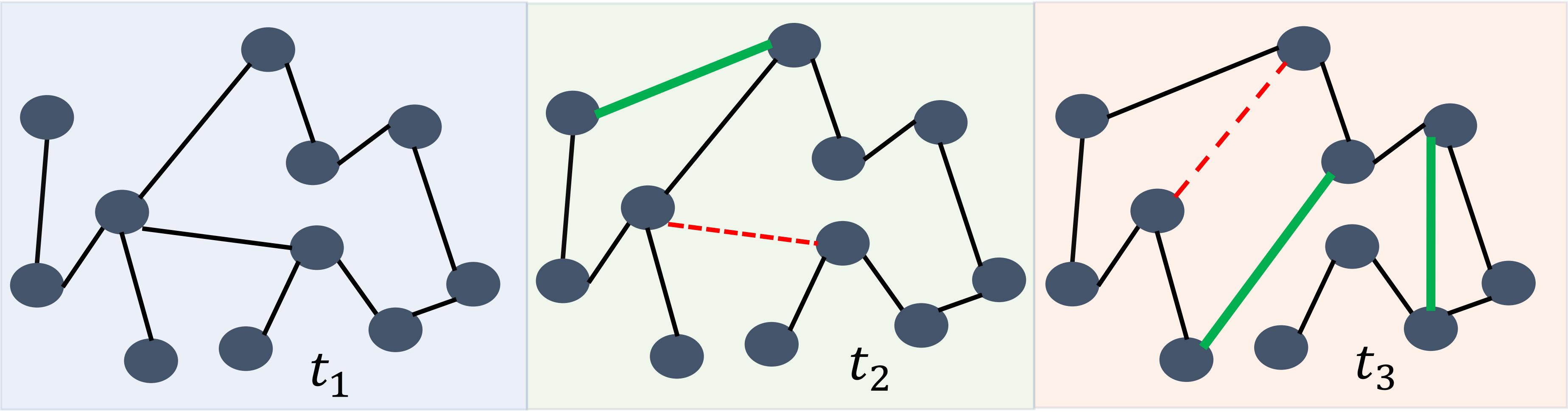
EPFL
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

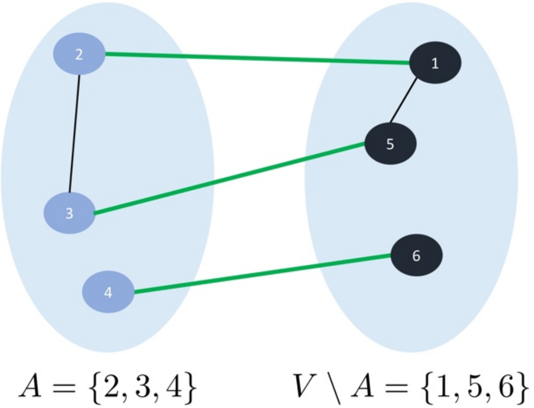
* The first two authors contributed equally

Motivation – sketching graphs



Motivation – sketching graphs

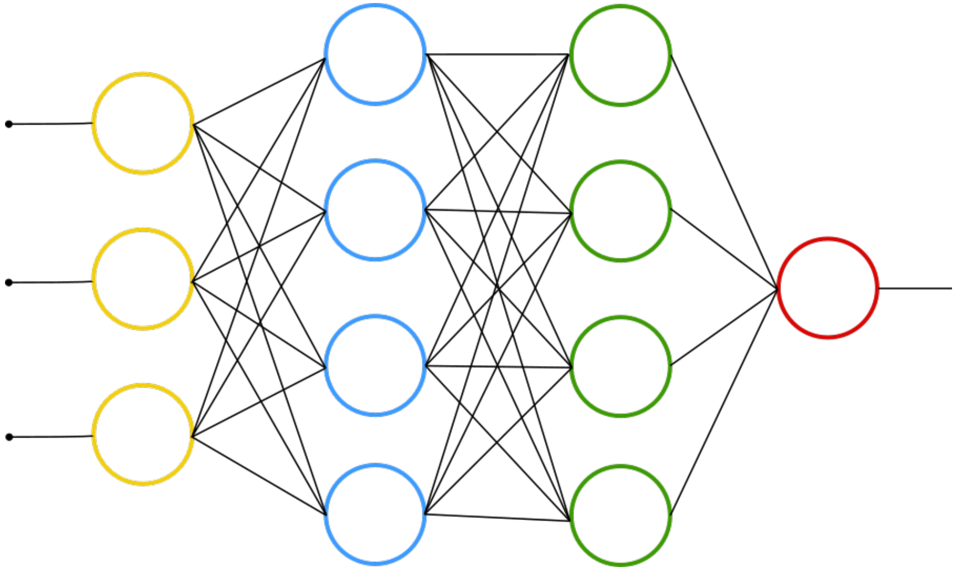




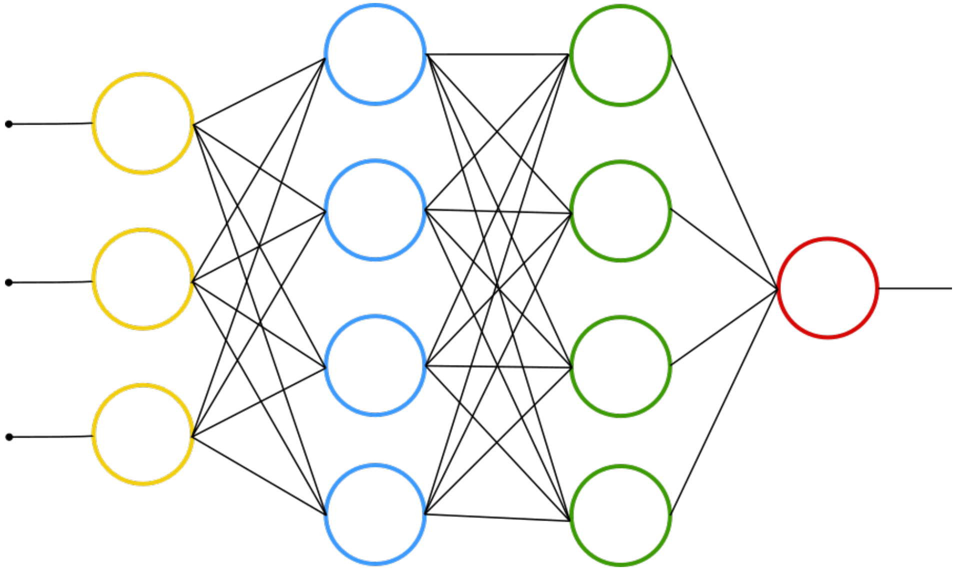
$$\underbrace{F(A)} = 3$$

Size of the cut between A and V \ A

Motivation – hyperparameter optimization



Motivation – hyperparameter optimization



$F(x_1\, x_2\, x_3\, x_4\, x_5)$ = Validation error using hyperparameters **x**

$x_1 = \begin{cases} 0 \text{ if optimizer is ADAM} \\ 1 \text{ if optimizer is SGD} \end{cases}$

$x_2 = \begin{cases} 0 \text{ if filtersize} = 3x3 \\ 1 \text{ if filtersize} = 5x5 \end{cases}$

…

Motivation – Learning set functions

- Given: set function $F: 2^V \to \mathbb{R}$. Can make (noisy) value queries

Motivation – Learning set functions

- Given: set function $F : 2^V \to \mathbb{R}$. Can make (noisy) value queries
- Goal: "learn" F with as few many queries as possible

Motivation – Learning set functions

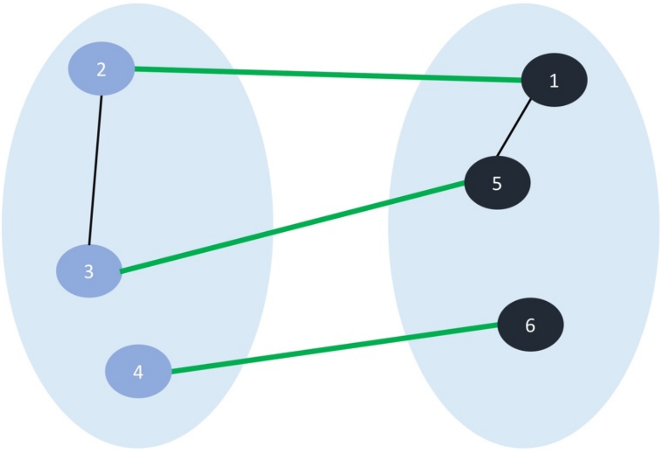
- Given: set function $F : 2^V \rightarrow \mathbb{R}$. Can make (noisy) value queries
- Goal: "learn" F with as few many queries as possible
- Assumptions: 1) sparsity - at most k non-zero Fourier coefficients

Motivation – Learning set functions

- Given: set function $F : 2^V \to \mathbb{R}$. Can make (noisy) value queries
- Goal: "learn" F with as few many queries as possible
- Assumptions: 1) sparsity - at most k non-zero Fourier coefficients
  2) low frequency - degree of Fourier polynomials at most d

Motivation – Learning set functions

- Given: set function $F : 2^V \to \mathbb{R}$. Can make (noisy) value queries
- Goal: "learn" F with as few many queries as possible
- Assumptions: 1) sparsity - at most k non-zero Fourier coefficients
  2) low frequency - degree of Fourier polynomials at most d

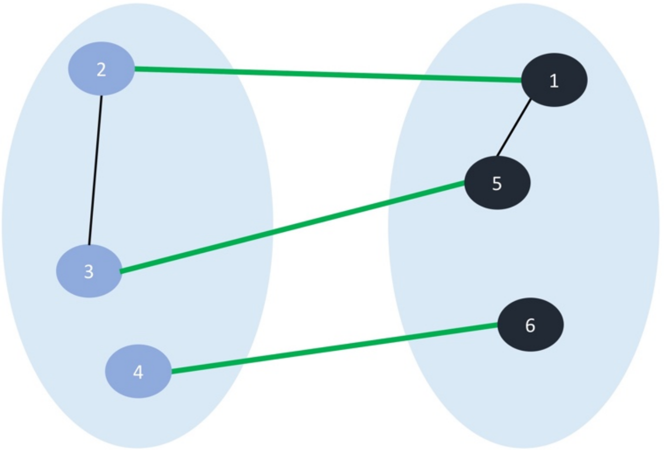


$A = \{2, 3, 4\}$ $V \setminus A = \{1, 5, 6\}$

$k = |E| + 1 = 6$ $d = 2$

Motivation – Learning set functions

- Given: set function $F : 2^V \to \mathbb{R}$. Can make (noisy) value queries
- Goal: "learn" F with as few many queries as possible
- Assumptions: 1) sparsity - at most k non-zero Fourier coefficients
  2) low frequency - degree of Fourier polynomials at most d



$A = \{2,3,4\}$ $\quad V \setminus A = \{1,5,6\}$

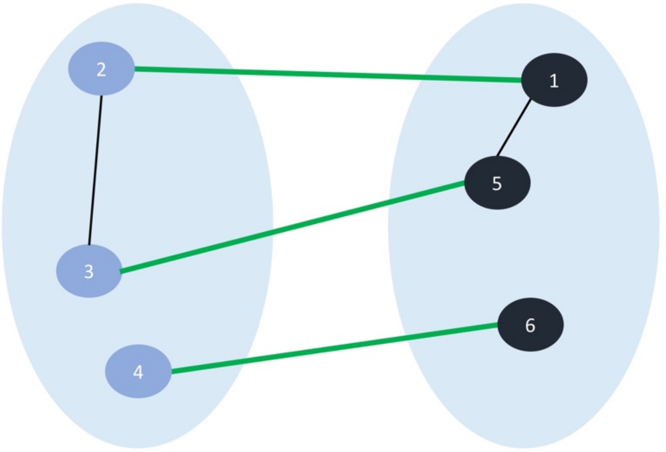
$k = |E| + 1 = 6 \quad d = 2$

$$F(x_1\, x_2\, x_3\, x_4\, x_5) =$$

$$F_1(x_1) + F_2(\underbrace{x_3\, x_4}_{d=2}) + F_3(x_2\, x_5)$$

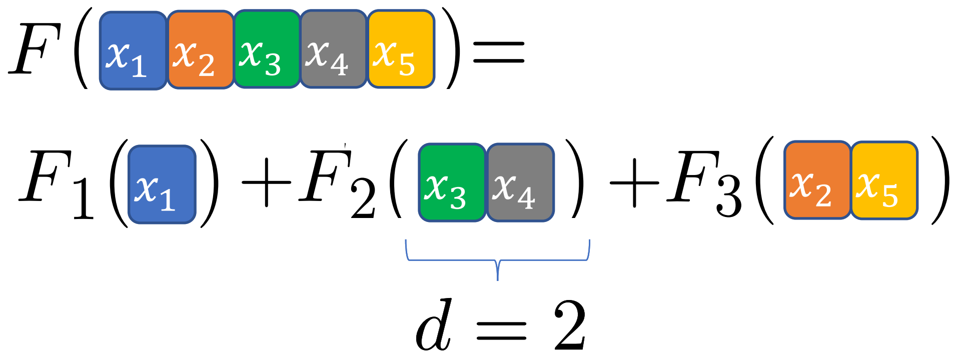
Motivation – Learning set functions

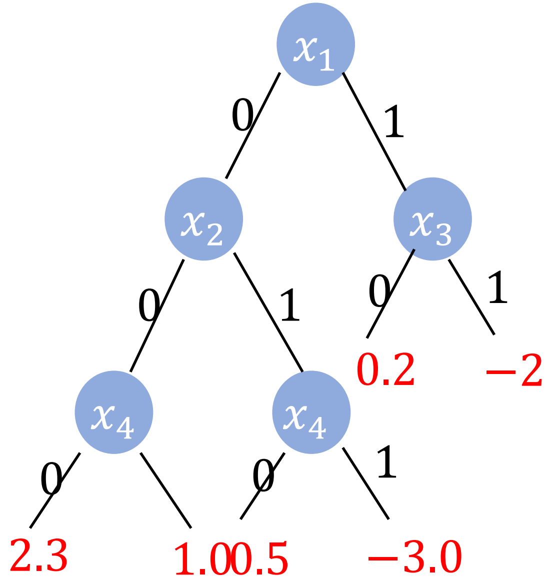
- Given: set function $F : 2^V \to \mathbb{R}$. Can make (noisy) value queries
- Goal: "learn" F with as few many queries as possible
- Assumptions: 1) sparsity - at most k non-zero Fourier coefficients
  2) low frequency - degree of Fourier polynomials at most d



$A = \{2,3,4\}$ $\quad V \setminus A = \{1,5,6\}$

$k = |E| + 1 = 6 \quad d = 2$



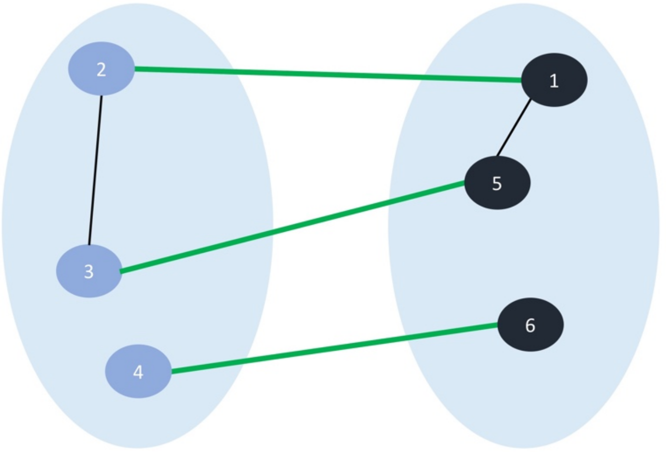


$d = 3$

Motivation – Learning set functions

Approximate Fourier transform of

- Given: set function $F : 2^V \to \mathbb{R}$. Can make (noisy) value queries
- Goal: "learn" F with as few many queries as possible
- Assumptions: 1) sparsity - at most k non-zero Fourier coefficients
  2) low frequency - degree of Fourier polynomials at most d

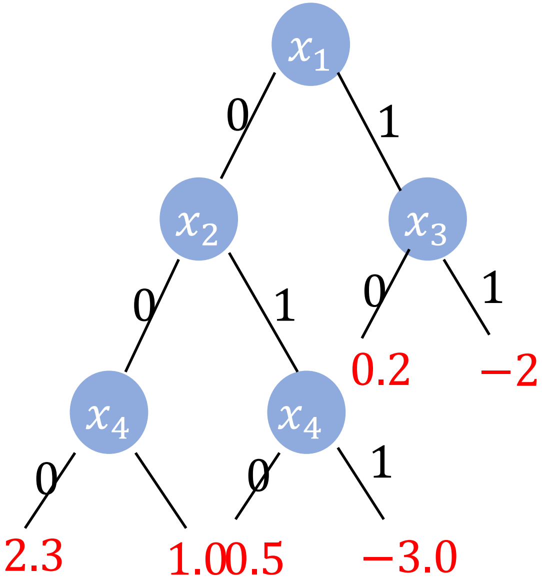


$A = \{2, 3, 4\}$ $V \setminus A = \{1, 5, 6\}$

$k = |E| + 1 = 6$ $d = 2$

$$F(x_1\, x_2\, x_3\, x_4\, x_5) =$$

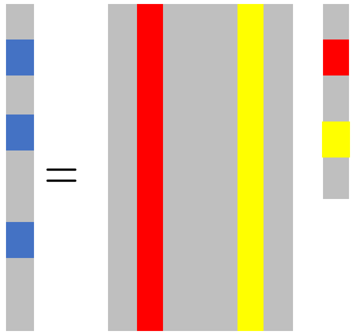
$$F_1(x_1) + \underbrace{F_2(x_3\, x_4)}_{d=2} + F_3(x_2\, x_5)$$

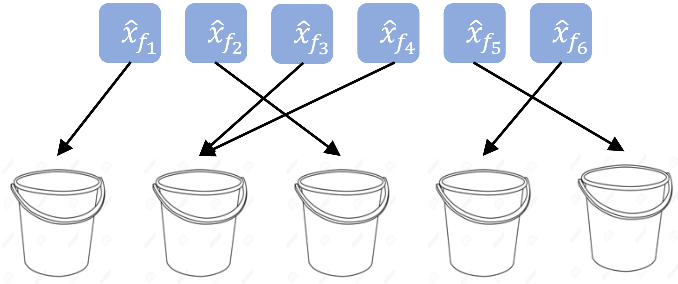


$d = 3$

Previous work and our contributions

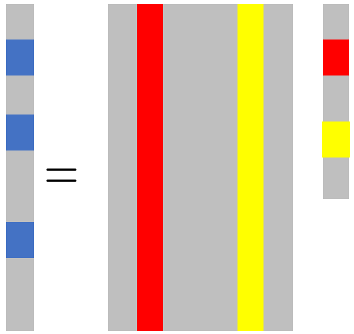
	Compressive sensing	Sparse FFT
Runtime		
Sampling complexity		
Assumptions		
Robustness		

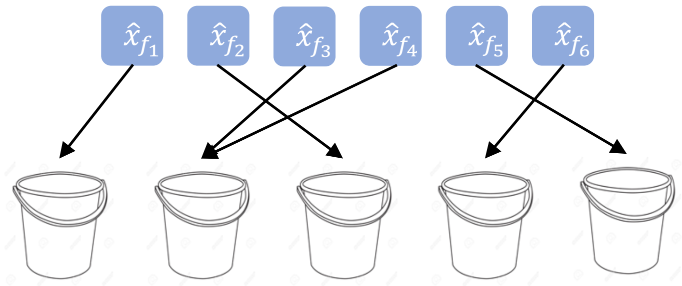




Previous work and our contributions

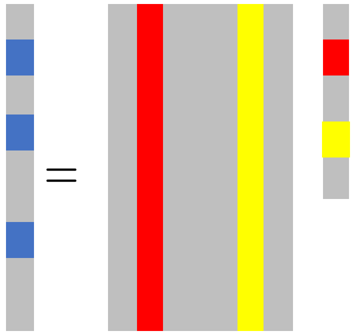
	Compressive sensing	Sparse FFT
Runtime	$\tilde{O}(kn^d)$	$\tilde{O}(kn^{(2)})$
Sampling complexity		
Assumptions		
Robustness		

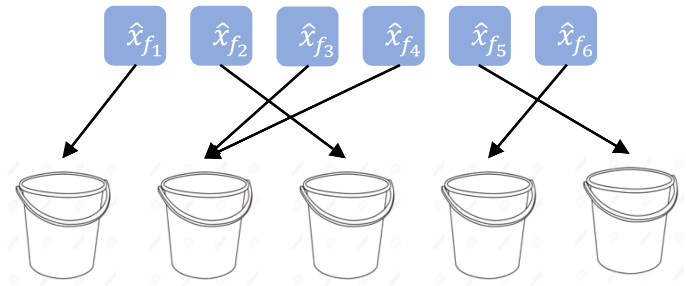




Previous work and our contributions

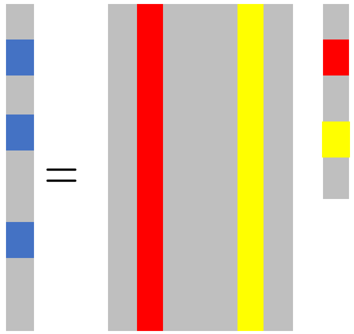
	Compressive sensing	Sparse FFT
Runtime	$\tilde{O}(kn^{d})$	$\tilde{O}(kn^{(2)})$
Sampling complexity	$\tilde{O}(kd)$	$\tilde{O}(kn)$
Assumptions		
Robustness		

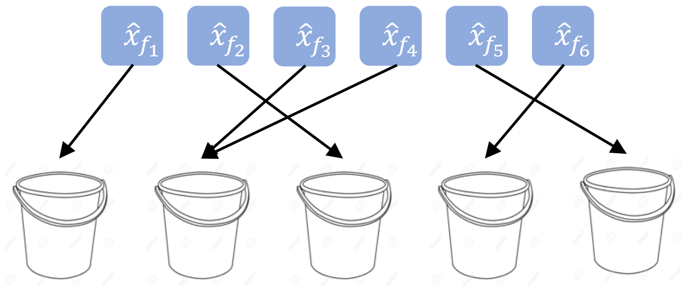




Previous work and our contributions

	Compressive sensing	Sparse FFT
Runtime	$\tilde{O}(kn^d)$	$\tilde{O}(kn^{(2)})$
Sampling complexity	$\tilde{O}(kd)$	$\tilde{O}(kn)$
Assumptions	None	Randomness of support
Robustness	Worst case noise	Gaussian noise + …

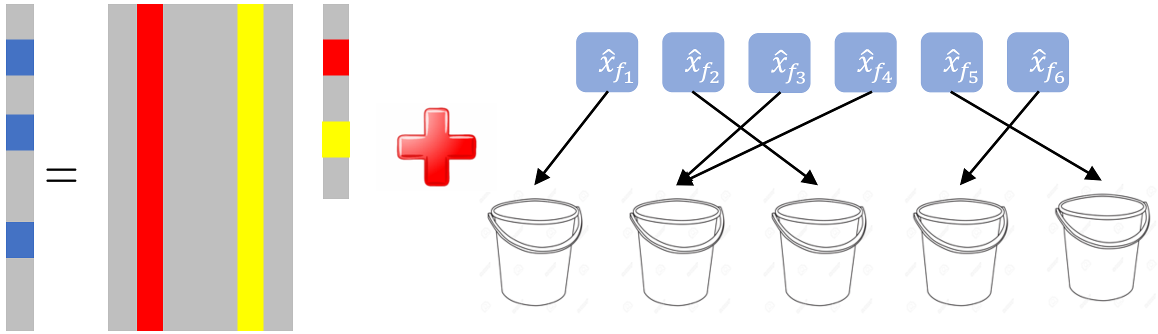




Previous work and our contributions

	Compressive sensing	Sparse FFT	Ours
Runtime	$\tilde{O}(kn^d)$	$\tilde{O}(kn^{(2)})$	$\tilde{O}(kn)$
Sampling complexity	$\tilde{O}(kd)$	$\tilde{O}(kn)$	$\tilde{O}(kd)$
Assumptions	None	Randomness of support	
Robustness	Worst case noise	Gaussian noise + …	

Compressive sensing over finite fields

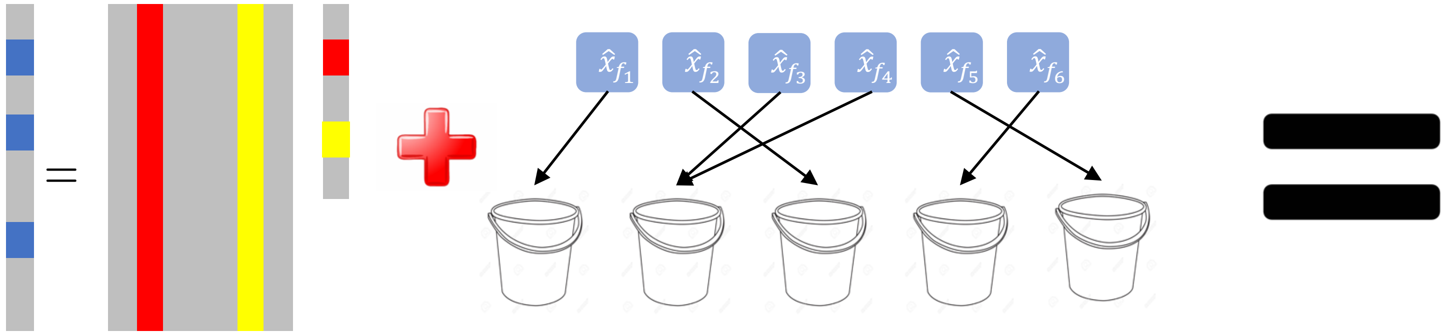


Previous work and our contributions

	Compressive sensing	Sparse FFT	Ours
Runtime	$\tilde{O}(kn^d)$	$\tilde{O}(kn^{(2)})$	$\tilde{O}(kn)$
Sampling complexity	$\tilde{O}(kd)$	$\tilde{O}(kn)$	$\tilde{O}(kd)$
Assumptions	None	Randomness of support	None
Robustness	Worst case noise	Gaussian noise + …	Worst case noise

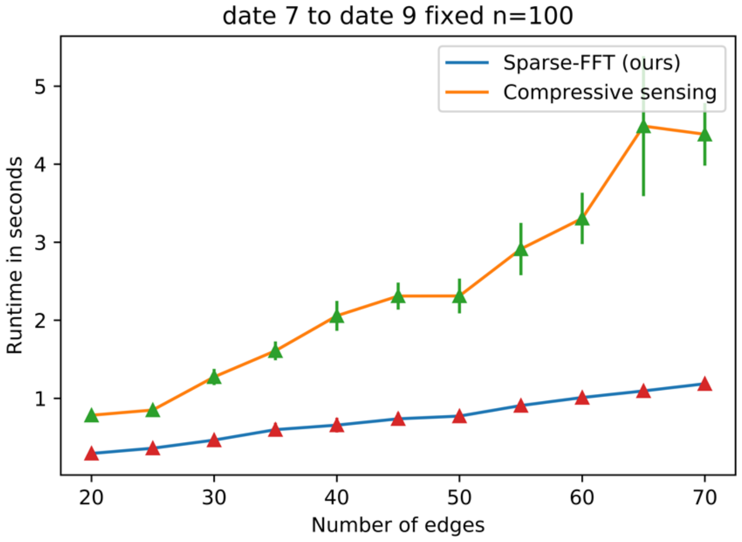
Compressive sensing over finite fields

New hashing schemes



Best of both worlds!

Please visit our poster for experimental results, more applications, and details of our algorithms



#44