Fast approximation of kernel matrices

with
Chenhan Yu, Bill March, and Bo Xiao

GEORGE BIROS
padas.ices.utexas.edu

INSTITUTE FOR COMPUTATIONAL ENGINEERING & SCIENCES

THE UNIVERSITY OF TEXAS AT AUSTIN

Fast approximation of kernel matrices

with
Chenhan Yu, Bill March, and Bo Xiao

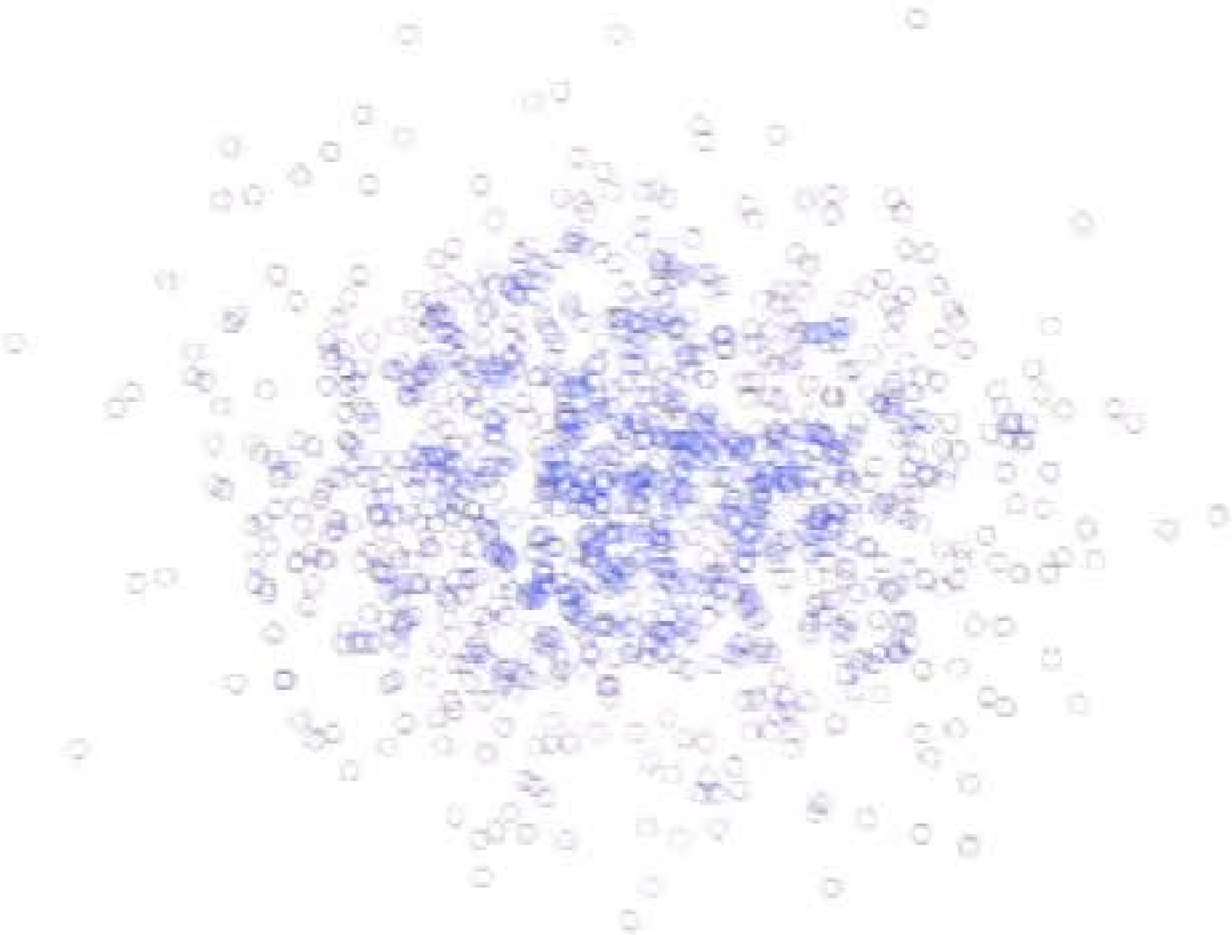
NSF DARPA

GEORGE BIROS
padas.ices.utexas.edu

INSTITUTE FOR COMPUTATIONAL ENGINEERING & SCIENCES

THE UNIVERSITY OF TEXAS AT AUSTIN

Kernel matrices



Input

N points in $\mathbb{R}^d$: $\quad x_1, \dots, x_N$

N densities in $\mathbb{R}$: $\quad w_1, \dots, w_N$

Output

N potentials in $\mathbb{R}$: $\quad u_1, \dots, u_N$

$$u_i = \sum_{j=1}^{N} G(x_i, x_j) w_j$$

$$G(x_i, x_j) = \exp\left(-\frac{1}{2}\frac{\|x_i - x_j\|_2^2}{h^2}\right)$$

Gaussian	$\exp(-\|x - x_j\|^2/(2h^2))$
Laplace	$\|x - x_j\|^{2-d},\ d > 2$
Matern	$\left(\sqrt{2\nu}\|x - x_j\|\right)^{\nu} K_\nu\left(\sqrt{2\nu}\|x - x_j\|\right)$
Polynomial	$(x^T x_j/h + c)^p$
Ornstein-Uhlenbeck	$\exp(-c\|x - x_j\|)$
Multiquadratic	$\sqrt{c^2 + \|x - x_j\|_2^2}$
Inverse multiquadratic	$1/\sqrt{c^2 + \|x - x_j\|_2^2}$

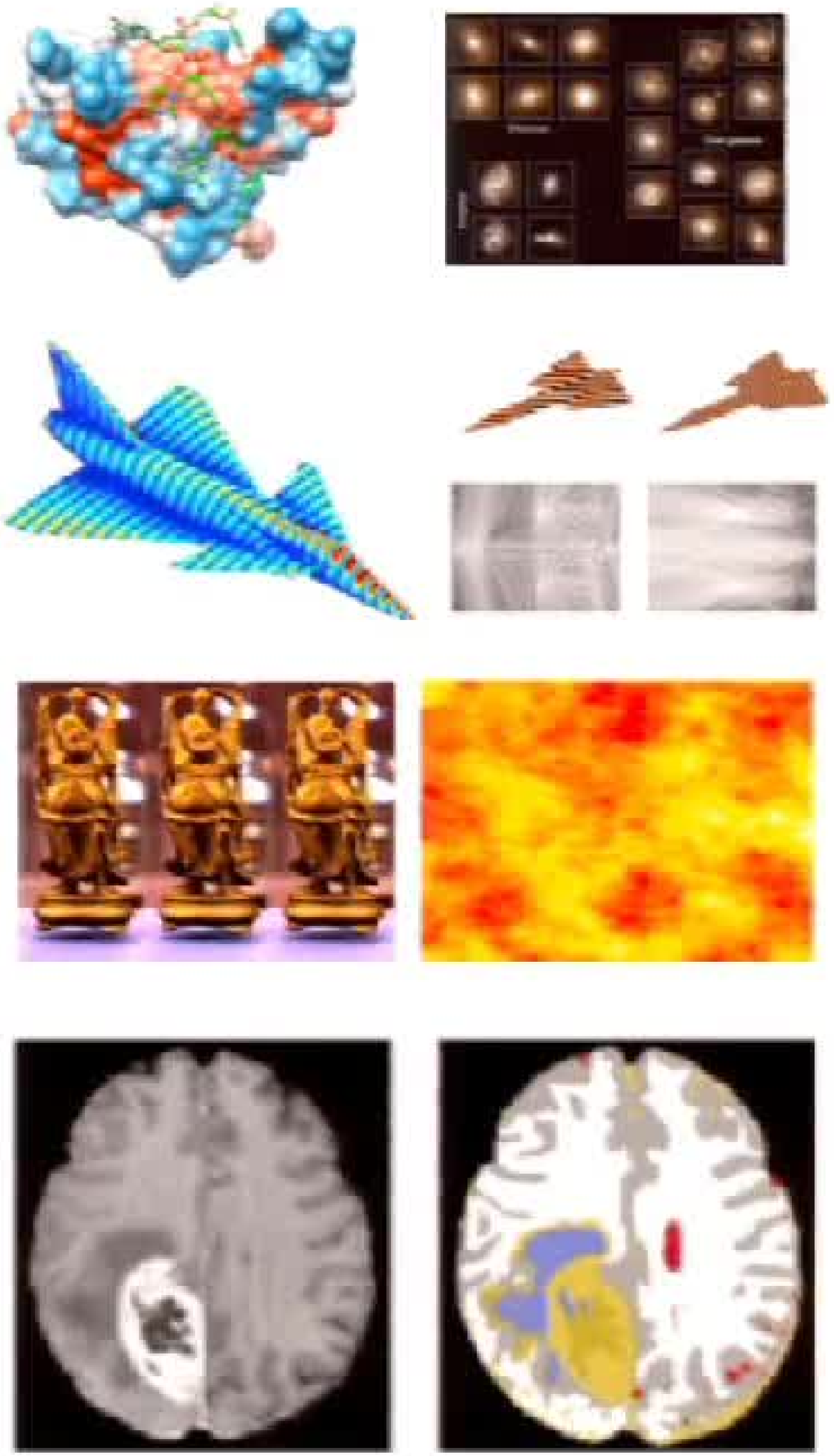
Applications

Simulation

- Gravity & Coulomb
- Waves & scattering
- Fluids & transport

Data analysis

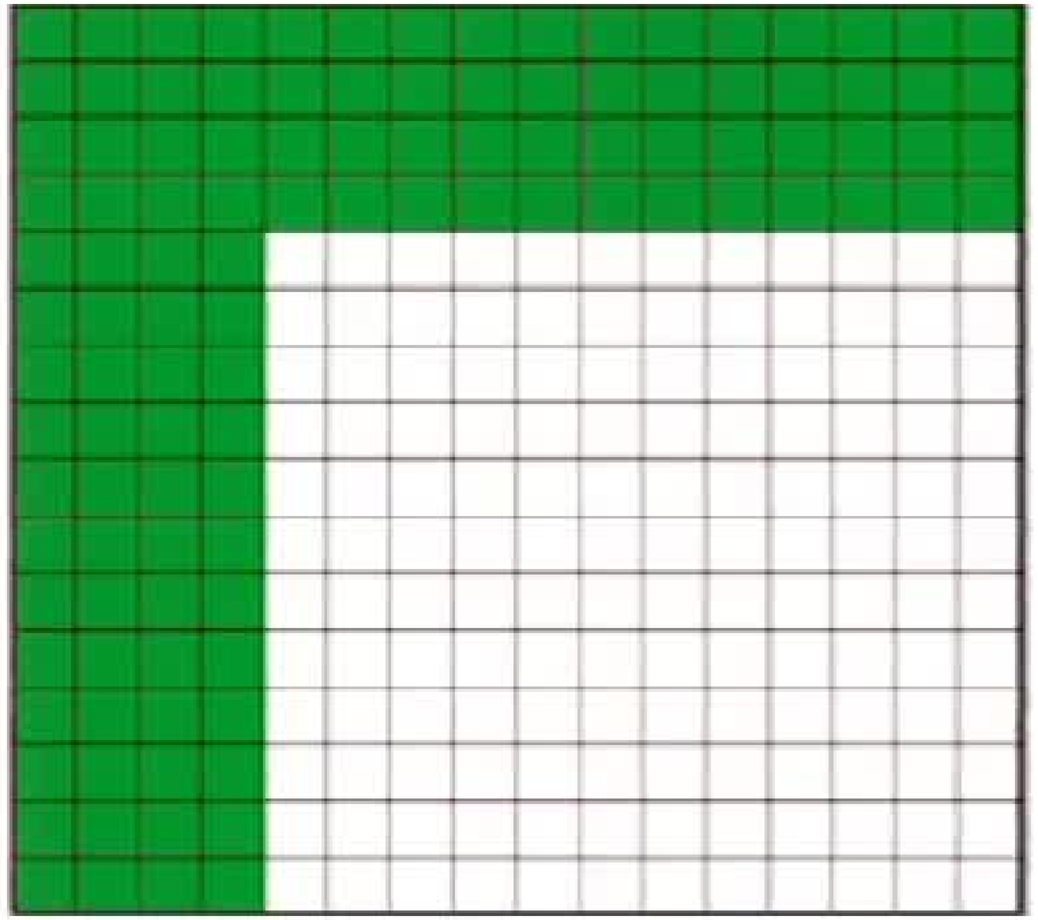
- Kernel methods in machine learning
- Approximation/Geostatistics
- Non-parametric statistics



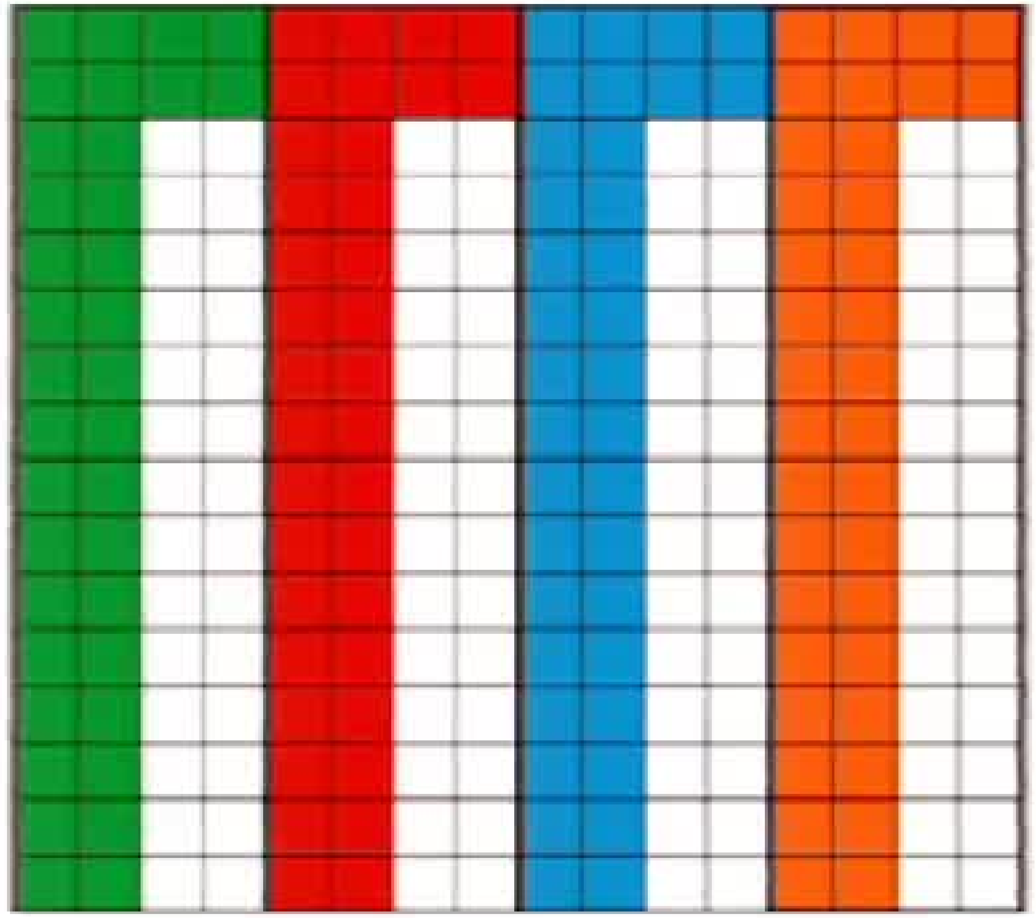
Computational challenges

- *N* points
  - N^2 work for matvec
  - N^3 work for factorization

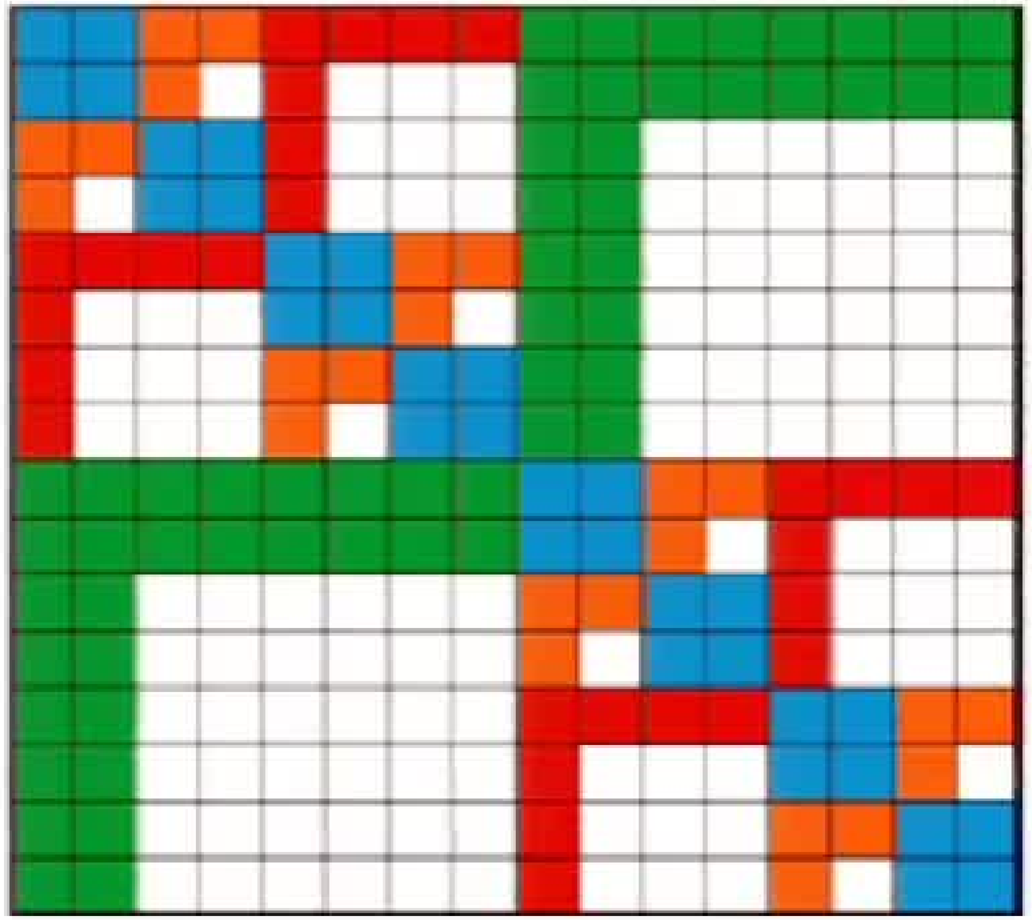
Achieving $O(N\ log^a N)$ complexity



NYSTROM



ENSEMBLE NYSTROM



HIERARCHICAL
MATRICES

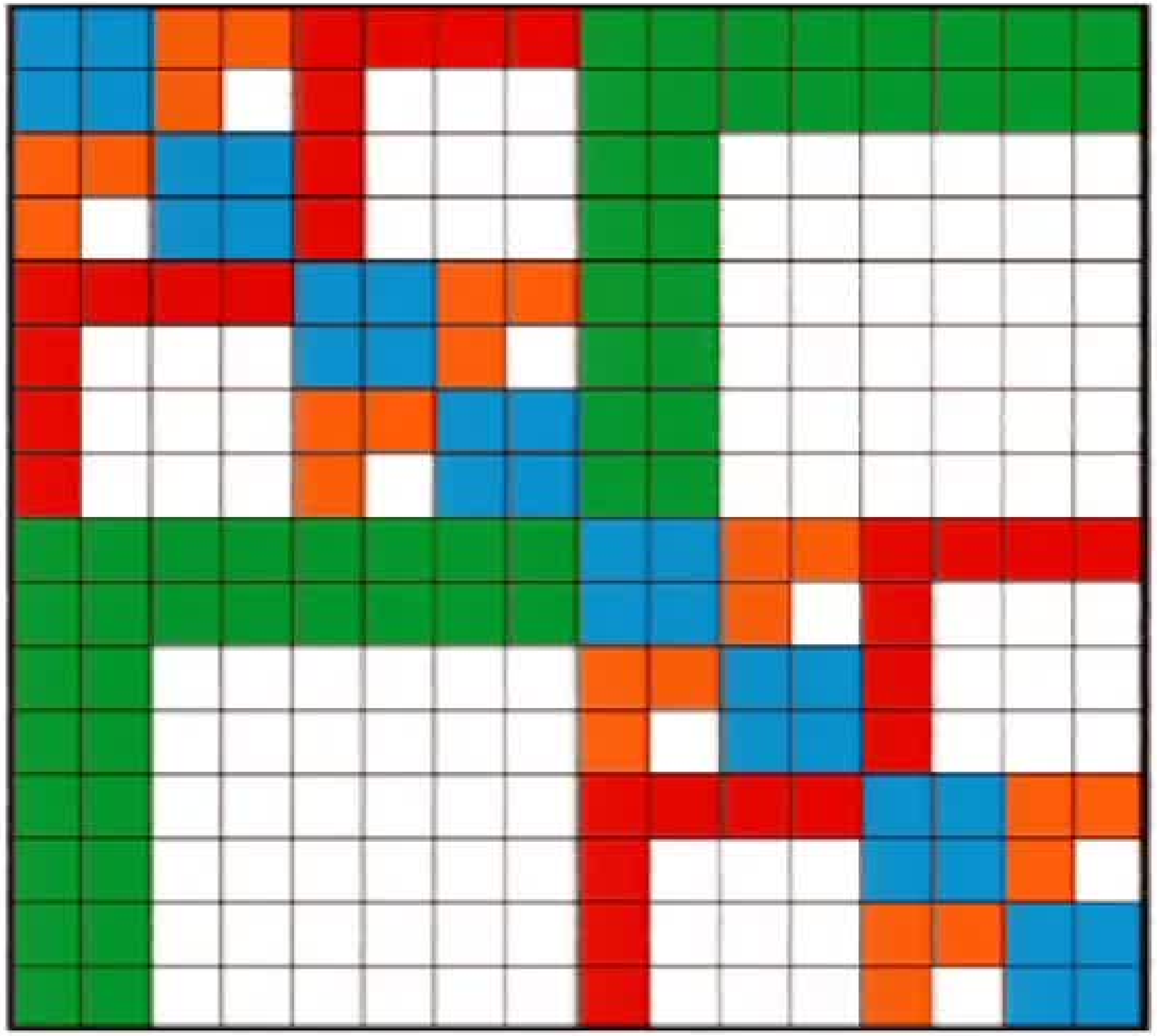
Hierarchical matrices, basic idea

$$\begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

$$= \begin{bmatrix} G_{11} & 0 \\ 0 & G_{22} \end{bmatrix} + \begin{bmatrix} 0 & G_{12} \\ G_{21} & 0 \end{bmatrix}$$

$$D + UV$$

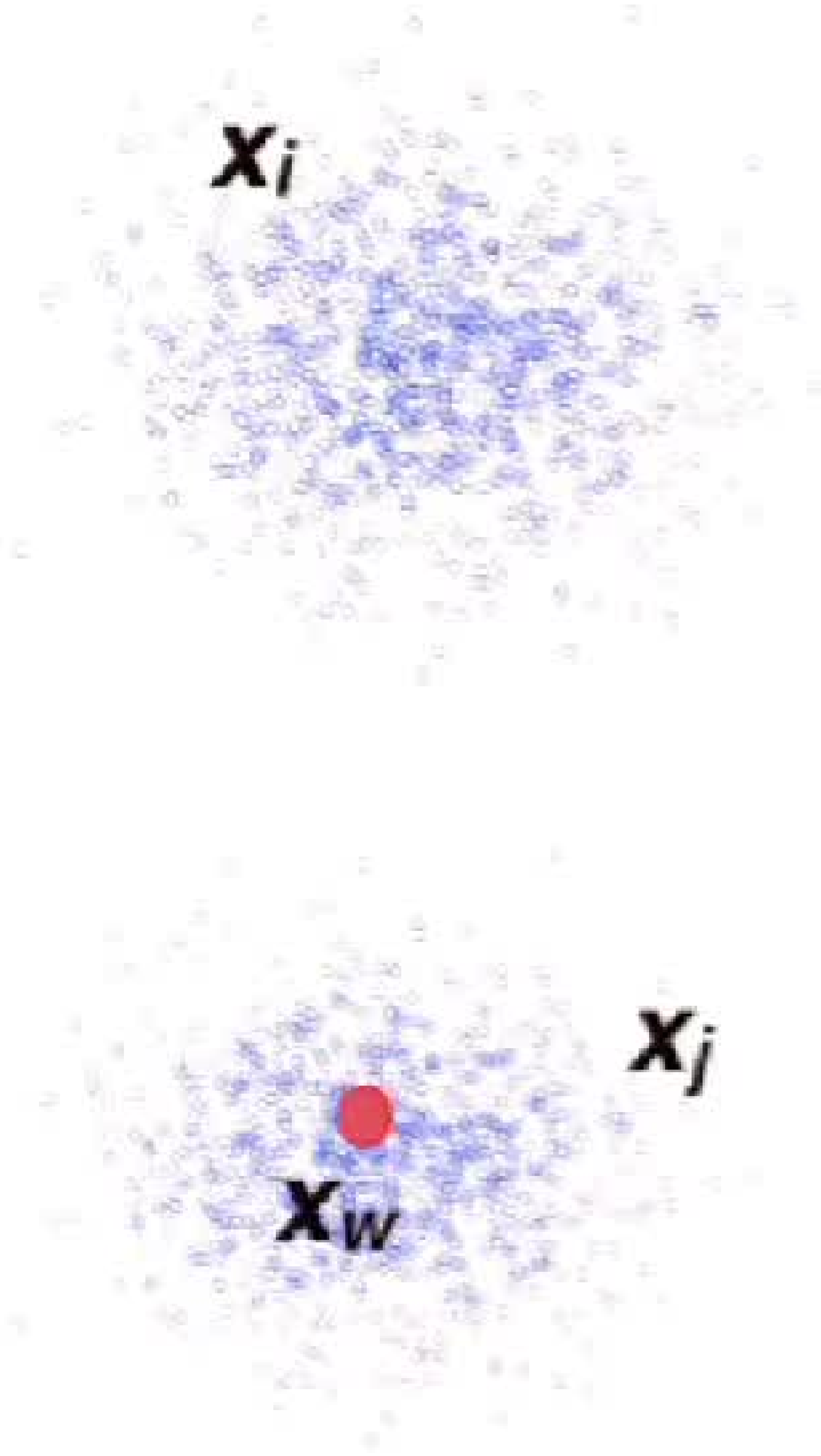
$$D + UV$$



$$\mathcal{O}(N^2) \rightarrow \mathcal{O}(N \log N)$$

Constructing the approximation

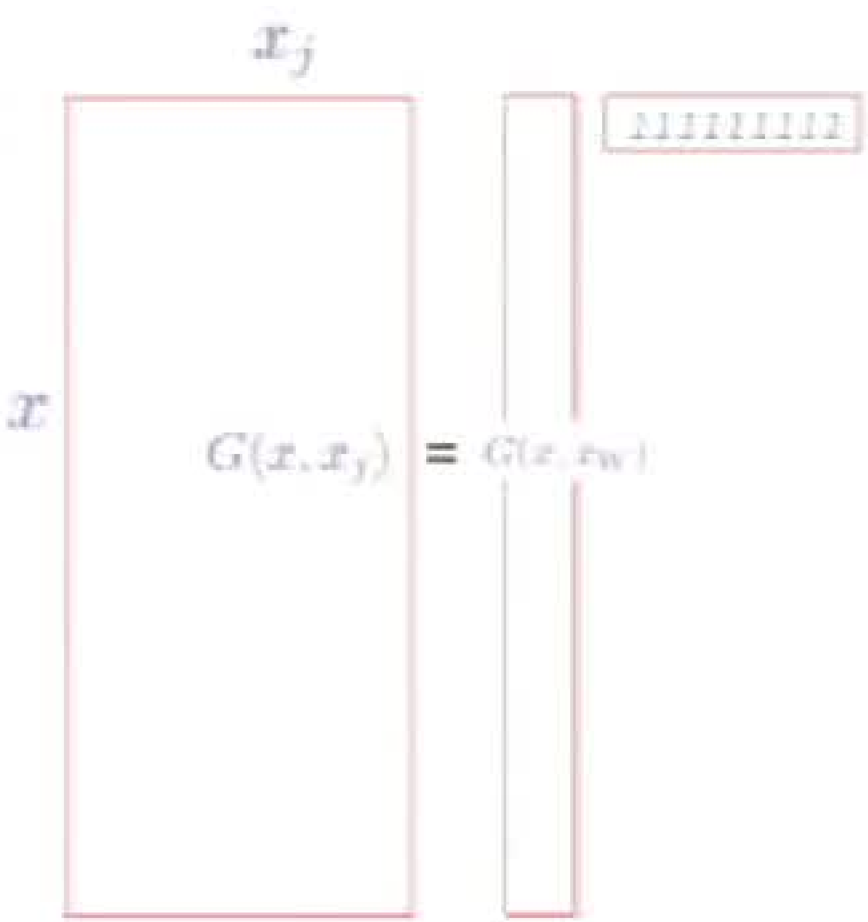
Idea I: far-field —> low rank

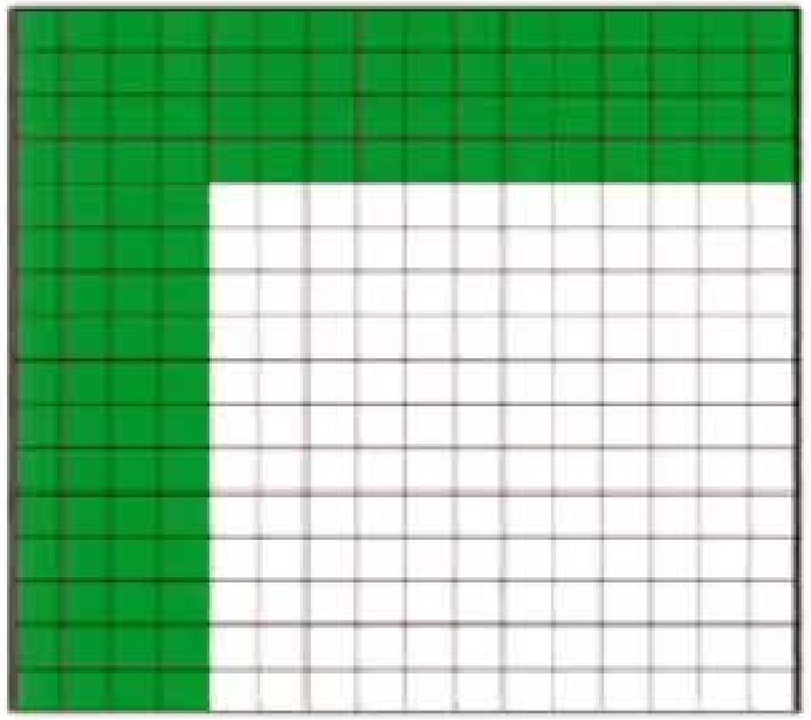


x: Target, x_j: sources
w_j: weights

$$u(x) = \sum_j G(x, x_j) w_j$$

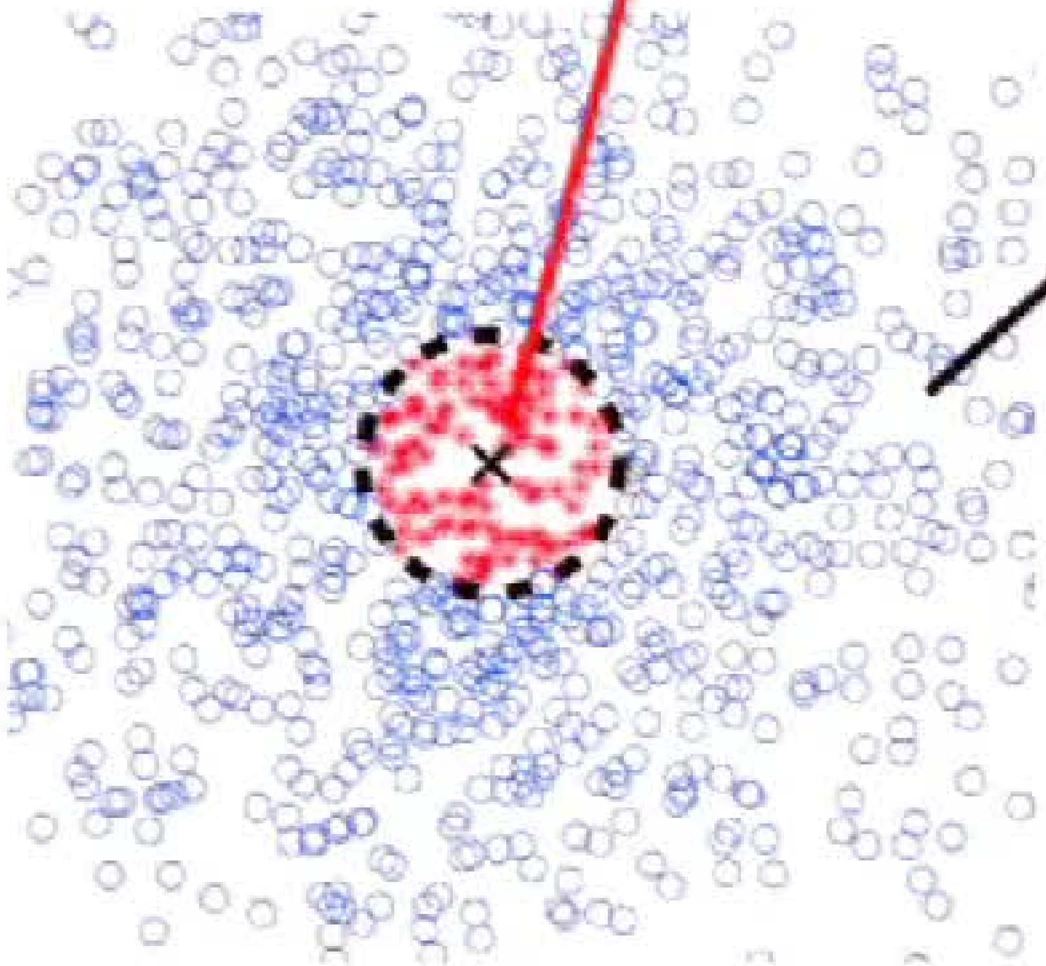
1. compute $W = \sum_i w_j$
2. choose x_W
3. $u(x) \approx G(x, x_W) W$



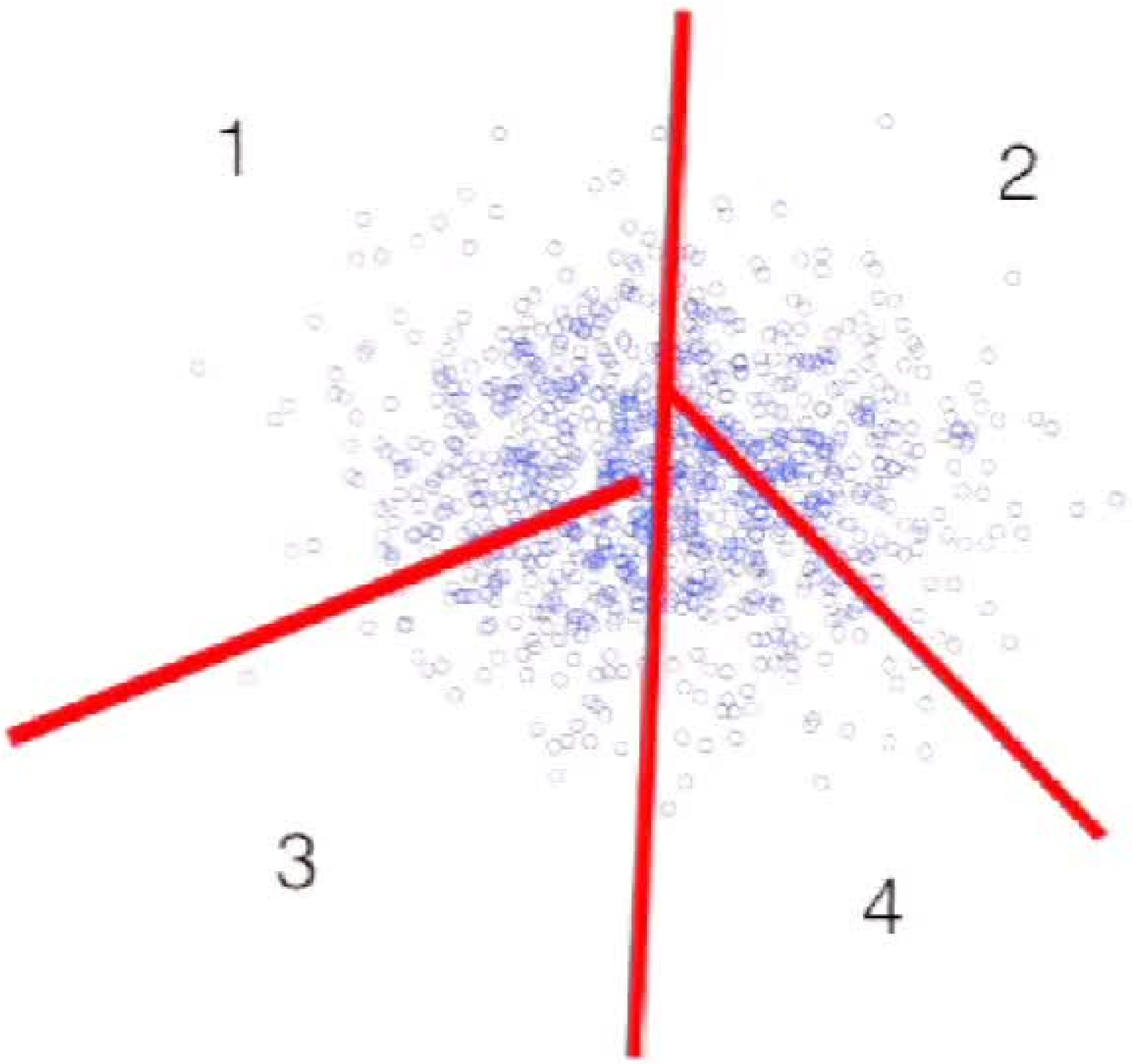


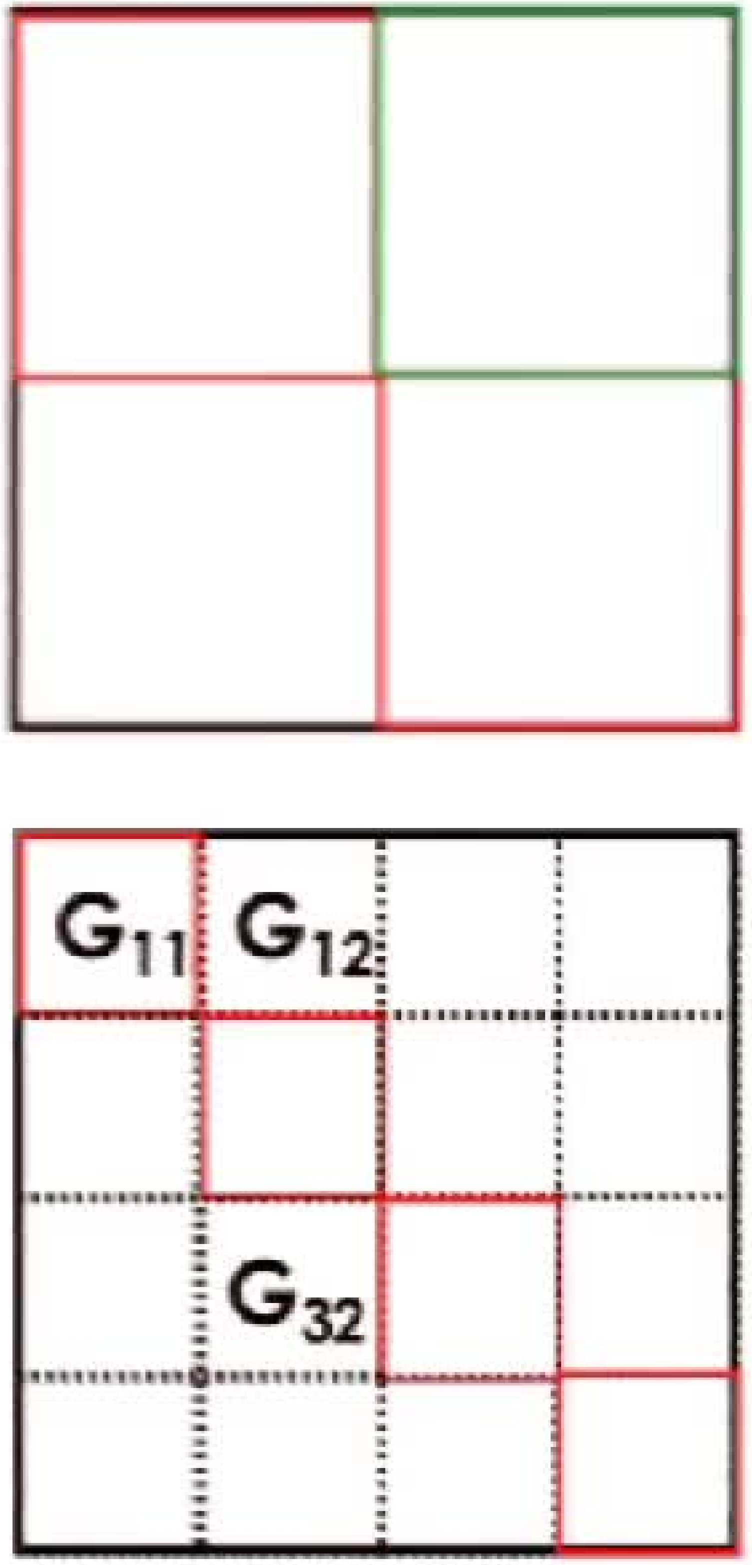
Idea II: Near/Far field split

$$u_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} G(x_i, x_j) w_j = \sum_{j \in \text{near}(i)} G_{ij} w_j + \sum_{j \in \text{far}(i)} G_{ij} w_j$$



Idea III: recursion





Questions

- Accurate far-field approximation
- Optimal complexity
- Error bounds
- HPC

Questions

- Accurate far-field approximation
- Optimal complexity
- Error bounds
- HPC
- **For d=4 these have been answered**

Related work — low dimensions

- Barnes & Hut'86 — treecodes
- Greengard & Rokhlin'87 — FMM
- Rokhlin'90 — high-frequency FMM
- Hackbush & Novak'89 — panel clustering
- Benderdorf'08 & Hackbush'99,'15 — H-matrices
- Greengard & Gropp'91 — parallel shared memory
- Warren & Salmon'93 — parallel distributed memory

Related work — high dimensions

- Griebel et al'12 — Fast Gauss transform
- Duraiswami'06 — Improved Fast Gauss transform
- Lee, Vuduc & Gray'12 — Treecode (parallel)
- Kondor et al'16 — Wavelets in high dimensions
- Mahoney & Darve'15 — HSS matrices
- Williams & Seeger'00 — Nystrom methods/global low rank

Challenges in high-dimensions

- Constructing the far-field approximations
  polynomial in ambient-D
- Near-far field decomposition
  polynomial in ambient-D
- No scalable algorithms (other than Nystrom)
- Nystrom method assumes low rank
  provably not the case with increasing N

ASKIT

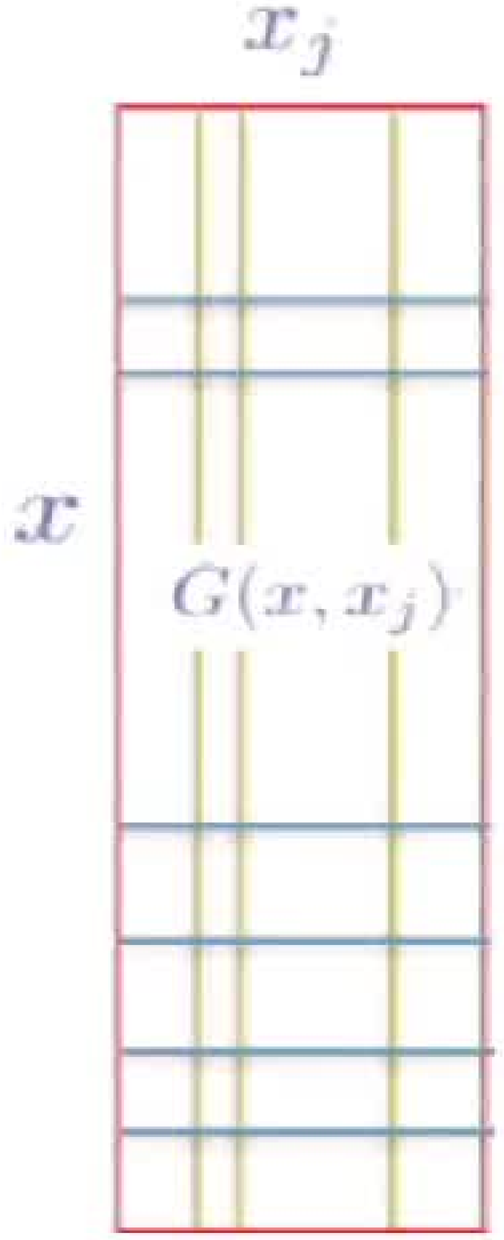
- Randomized Linear Algebra — far field approximation
- Parallel binary trees — permutation, partitioning
- Nearest neighbors — pruning and sampling
- Treecode / FMM
- MPI / OpenMP / SIMD / GPU acceleration
- Inspired by
  Ying & B. & Zorin'03
  Haiko & Martinsson & Tropp'11
  Drineas & Kahan & Mahoney'06

SISC'15,16
ACHA'15
KDD'15
SC'15
IPDPS'15,16,17

Far-field *s-rank* approximation

$$G(x, x_j) = G_{x,s}(G_{\ell,s})^{\dagger}G_{s,x_j}$$

- SVD is too expensive — use sampling
- Sample rows
  leverage, norm, range-space
- Interpolative decomposition
- ASKIT: approximate norm *adaptive* sampling
  using nearest-neighbors + *adaptive* rank selection



Complexity and error

- Work

	RAM	skeletonize	evaluate
	$(d+\kappa)N$	Ns^2	$dNs\kappa\log(\frac{N}{s})$

- Error $\|G-\tilde{G}\| \leq \sqrt{1+6N/s}\ \log(N/s)\ \gamma_{s+1}\ \sigma_{s+1}$ off-diagonal

- Nystrom $\|G-\tilde{G}\| \leq \sqrt{1+6N/s}\ \sigma_{s+1}$

  $Ns+s^3$ diagonal

Summary of ASKIT features

- Binary tree for matrix perturbation
- Approximate randomized nearest neighbors
- Nearest neighbors for skeletonization
- Bottom-up recursive low-rank approximation
- Top-down pass for fast evaluation
- Adaptive sampling and rank selection

Gaussian

3D, 1M points

ϵ_2	T_S	T_{LET}	T_L	T_E	$\%K$
5E-10	439	53	7	4	2.1%
5E-05	73	16	1	1	0.6%
2E-04	29	15	1	1	0.4%
1E-03	14	15	1	1	0.3%
6E-03	10	15	1	1	0.2%

64D/20D intr, 1M points

ϵ_2	T_S	T_{LET}	T_L	T_E	$\%K$
9E-06	1068	395	149	260	56%
4E-04	486	67	11	29	6.2%
5E-03	57	30	1	9	1.6%

Kernel regression

Train:

$\{x_i \in \mathbb{R}^d, c_i \in \{-1,1\}_{i=1}^N\}$

$\{w_j\}_{j=1}^N : \sum_{j=1}^N G(x_i, x_j) w_j = c_i, \quad \forall i.$

Classify: $c(x) = \text{sign} \sum_{j=1}^N G(x, x_j) w_j$

low rank ↑

full rank ↓

COVTYPE		SUSY		MNIST2M	
h	ϵ_c	h	ϵ_c	h	ϵ_c
0.35	71.6	0.50	65.7	4	95.0
0.22	74.0	0.15	72.1	2	97.4
0.14	79.8	0.09	75.0	1	100
0.02	95.4	0.05	76.7	0.1	99.5
0.001	6.4	0.01	64.3	0.05	13.6

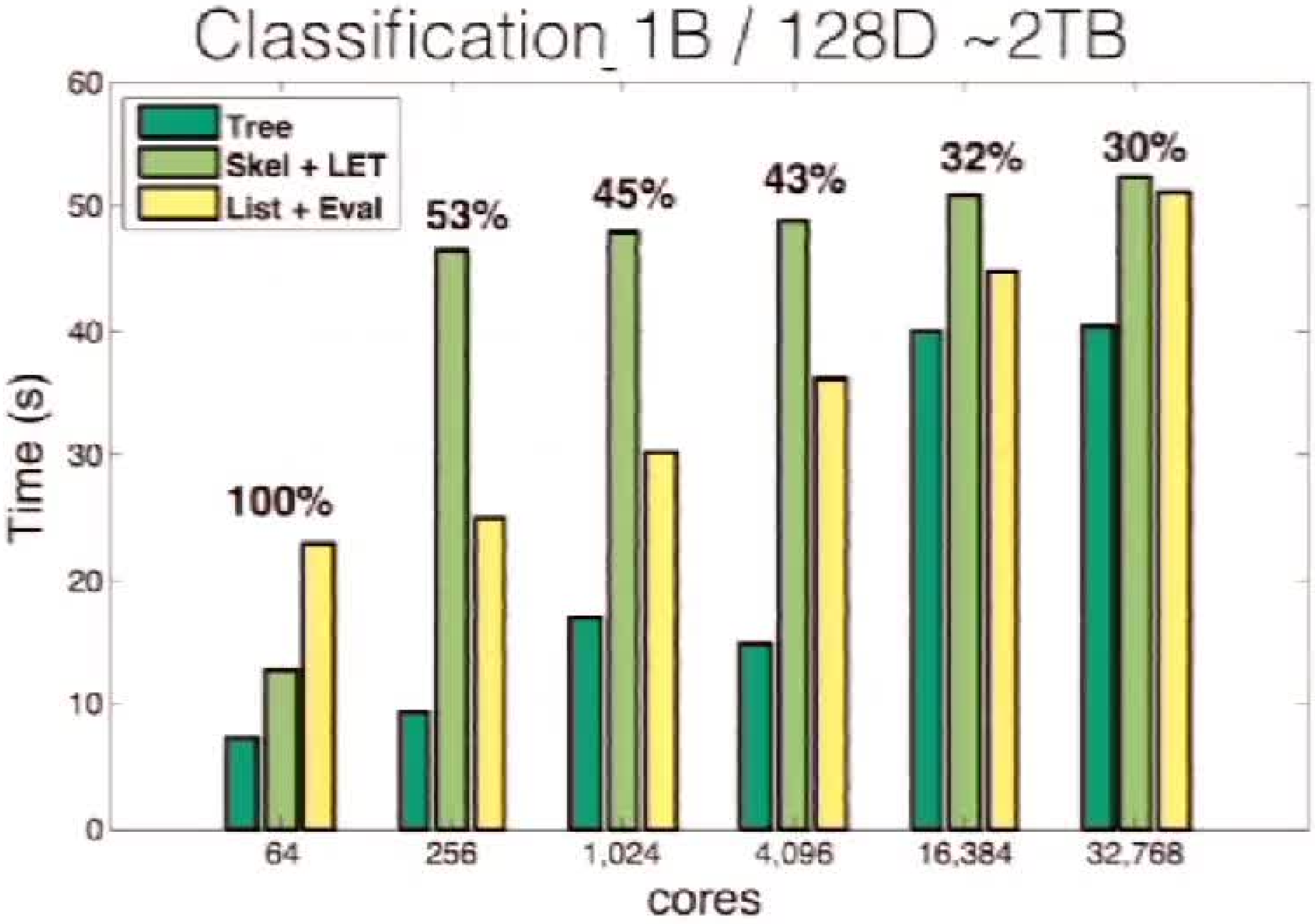
Kernel acceleration

Data	$\boldsymbol{N}$	$\boldsymbol{d}$	$\boldsymbol{\epsilon_2}$	$\boldsymbol{\%K}$
Uniform	1M	64	5E-3	1.6%
Covtype	500K	54	8E-2	2.7%
SUSY	4.5M	18	5E-3	0.4 %
HIGGS	10.5M	28	1E-1	11%
BRAIN	10.5M	246	5E-3	0.9%

Nystrom vs ASKIT (8M/784D)

	Param	$h = 0.5$			$h = 1$		
		ϵ_2	T	T_E	ϵ_2	T	T_E
NYSTROM	$r = 1024$	>9E-1	63	<1	>9E-1	63	<1
	$r = 2048$	>9E-1	122	<1	>9E-1	120	<1
	$r = 4096$	>9E-1	299	<1	>9E-1	301	<1
	$r = 8192$	mem	–	–	mem	–	–
ASKIT	$\kappa = 256$	1E-4	226	32	3E-2	154	31
	$\kappa = 512$	3E-5	243	39	2E-2	181	38
	$\kappa = 1024$	5E-6	306	50	2E-2	239	47
	$\kappa = 2048$	9E-7	410	65	8E-3	370	62

Weak scaling



TACC's Stampede
Largest run 144s
200 TFLOPs
30% peak