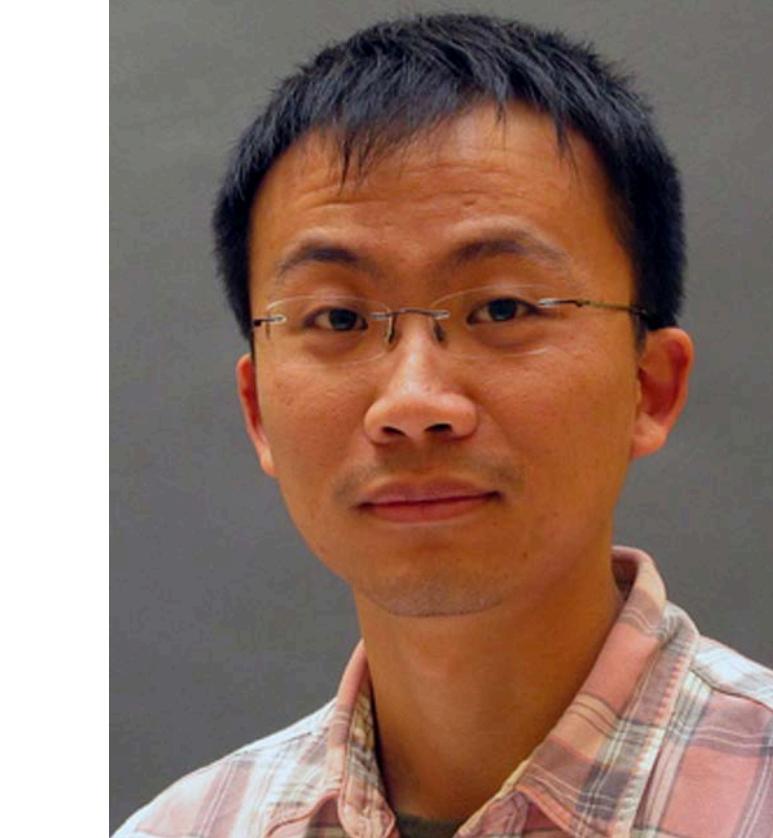


Fast Multipole Methods in Arbitrary Dimensions

with

Chenhan Yu James Levitt Severin Riez Bill March Bo Xiao



GEORGE BIROS
padas.ices.utexas.edu

INSTITUTE FOR COMPUTATIONAL
ENGINEERING & SCIENCES



THE UNIVERSITY OF
TEXAS
AT AUSTIN

QUALCOMM

Problem statement and contributions

- FMM for kernel matrices given points in high D
- FMM for SPD matrices—no points given
- Four components
 - Matrix permutations to expose low-rank structure — $O(N)$
 - Compress blocks — $O(N \log N)$
 - Fast Matvec — $O(N)$
 - HPC implementation (MPI async + ARM, x86/KNL, GPUs)

Motivation: Kernel classification

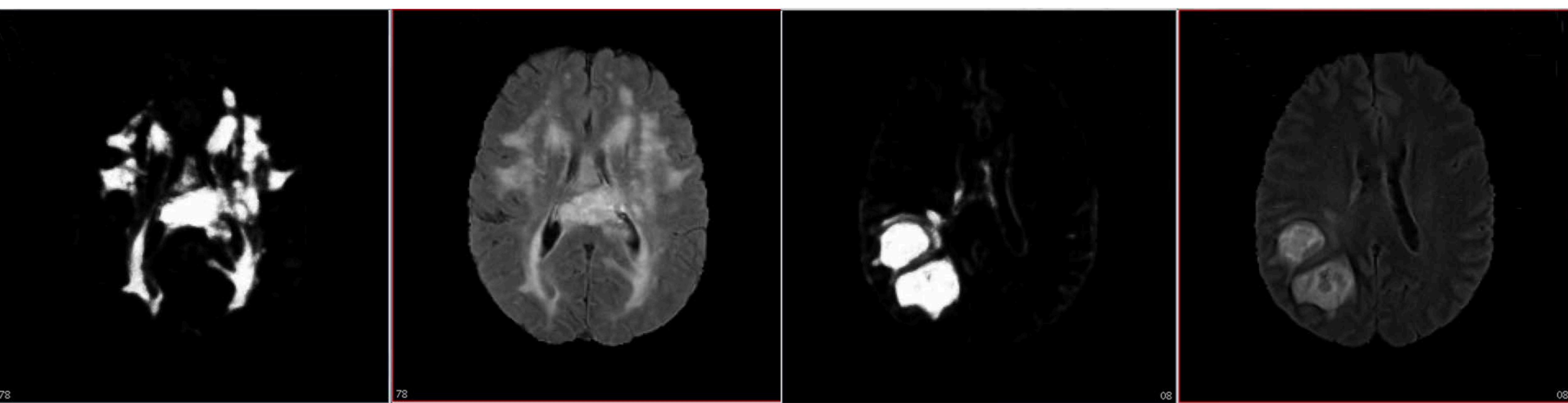
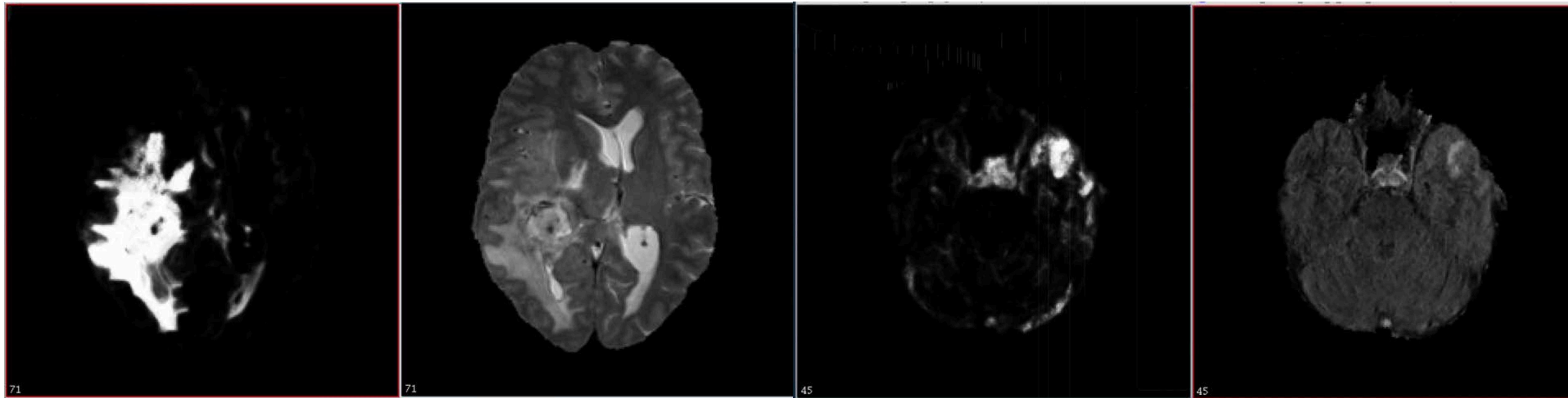
Train:

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

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

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

		COVTYPE		SUSY		MNIST2M	
		h	ϵ_c	h	ϵ_c	h	ϵ_c
low rank	↑	0.35	71.6	0.50	65.7	4	95.0
	↓	0.22	74.0	0.15	72.1	2	97.4
full rank	↑	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



Motivation: arbitrary SPD matrices

- Hessian matrix for large scale optimization
- Schur-complement operators for computing inverse graph Laplacians
- Factorization of dense covariance matrices

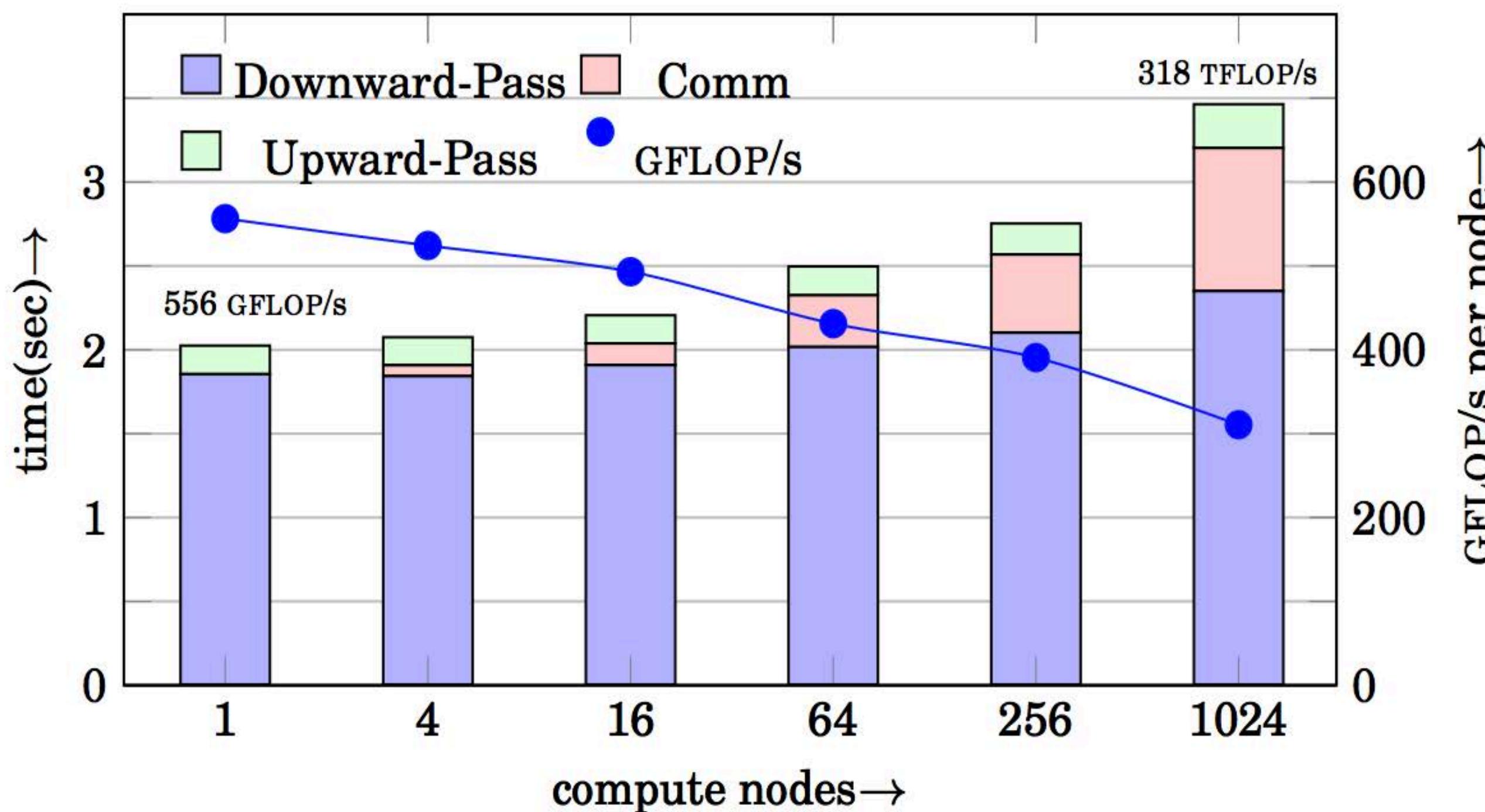
No available geometry information (i.e., points)

Two algorithms

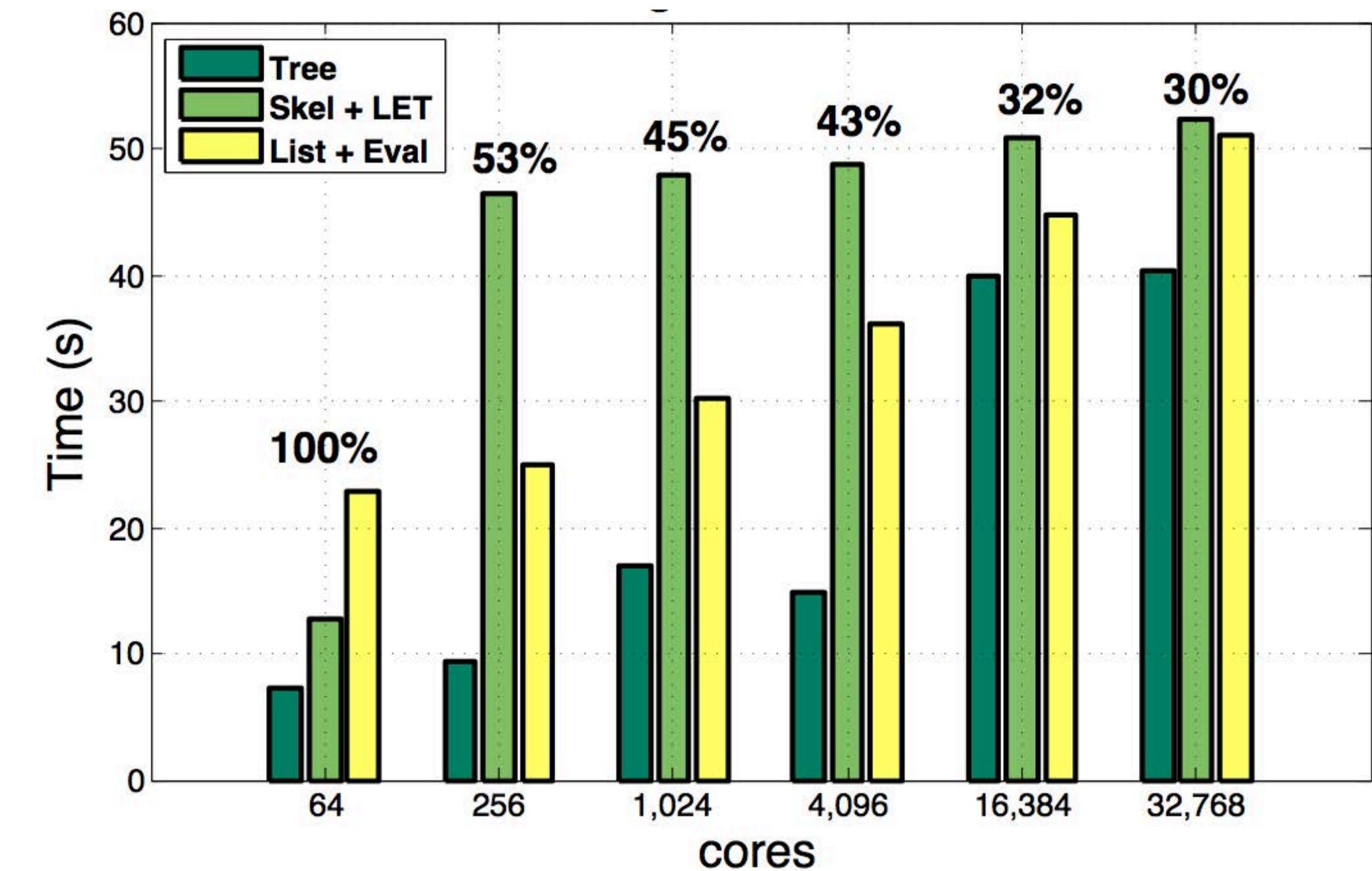
- ASKIT: Algebraically Skeletonized Kernel Independent Treecode
- GOFMM: Geometry Oblivious Fast Multipole Method

Highlights ASKIT

CFD:12B/3D ~ 700 GB



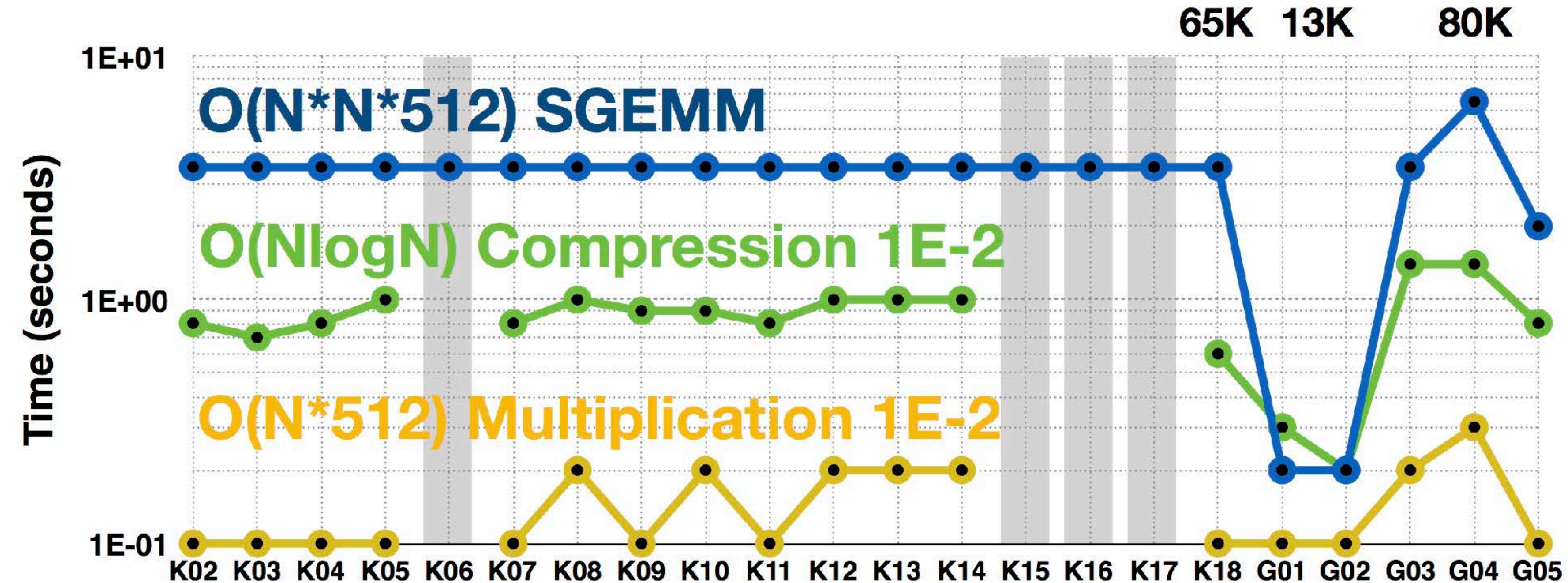
Kernels: 1B/128D ~ 1TB



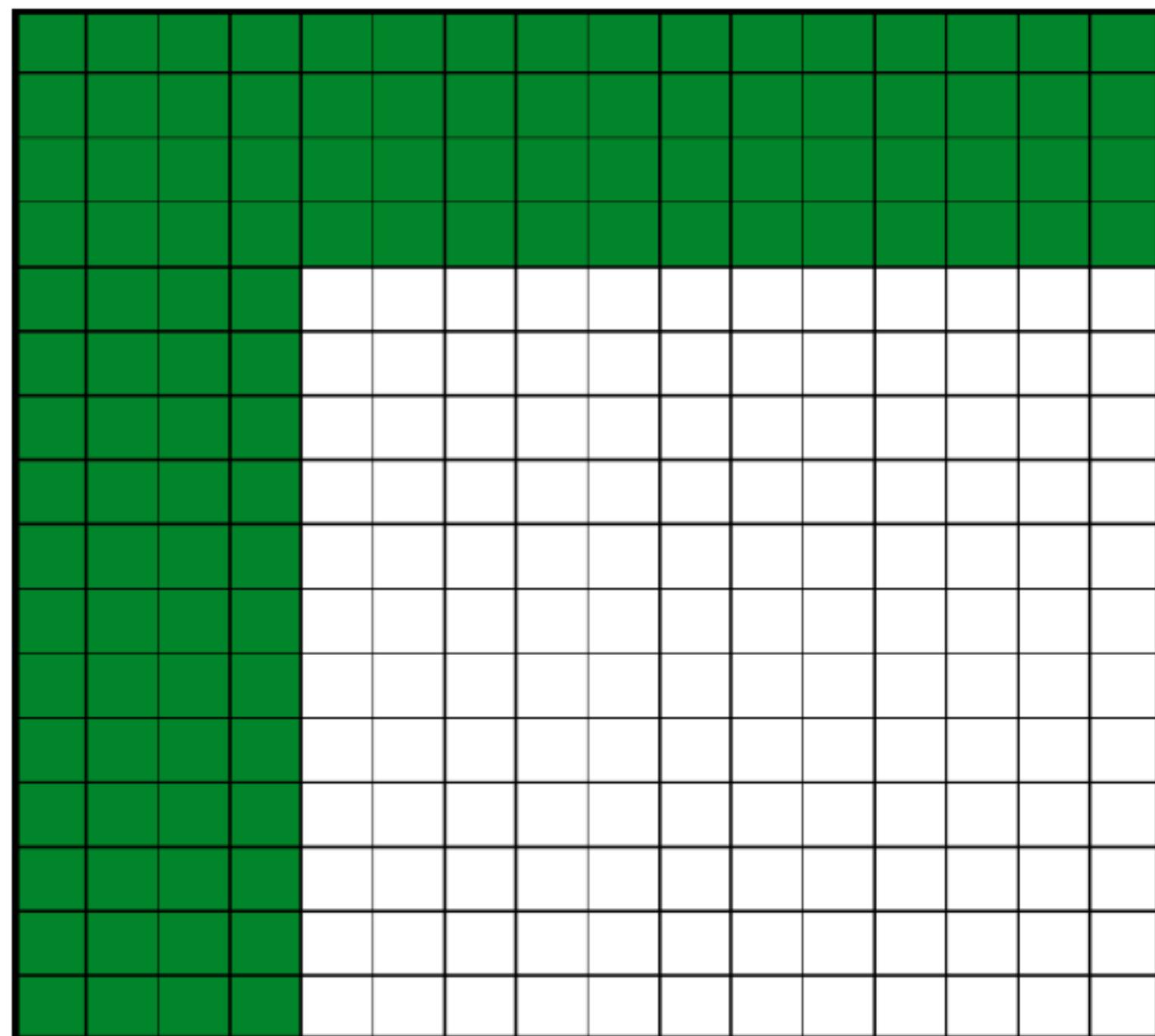
Malhotra, Gholami, & B' SC14

March, Yu, Xiao, B. KDD'15

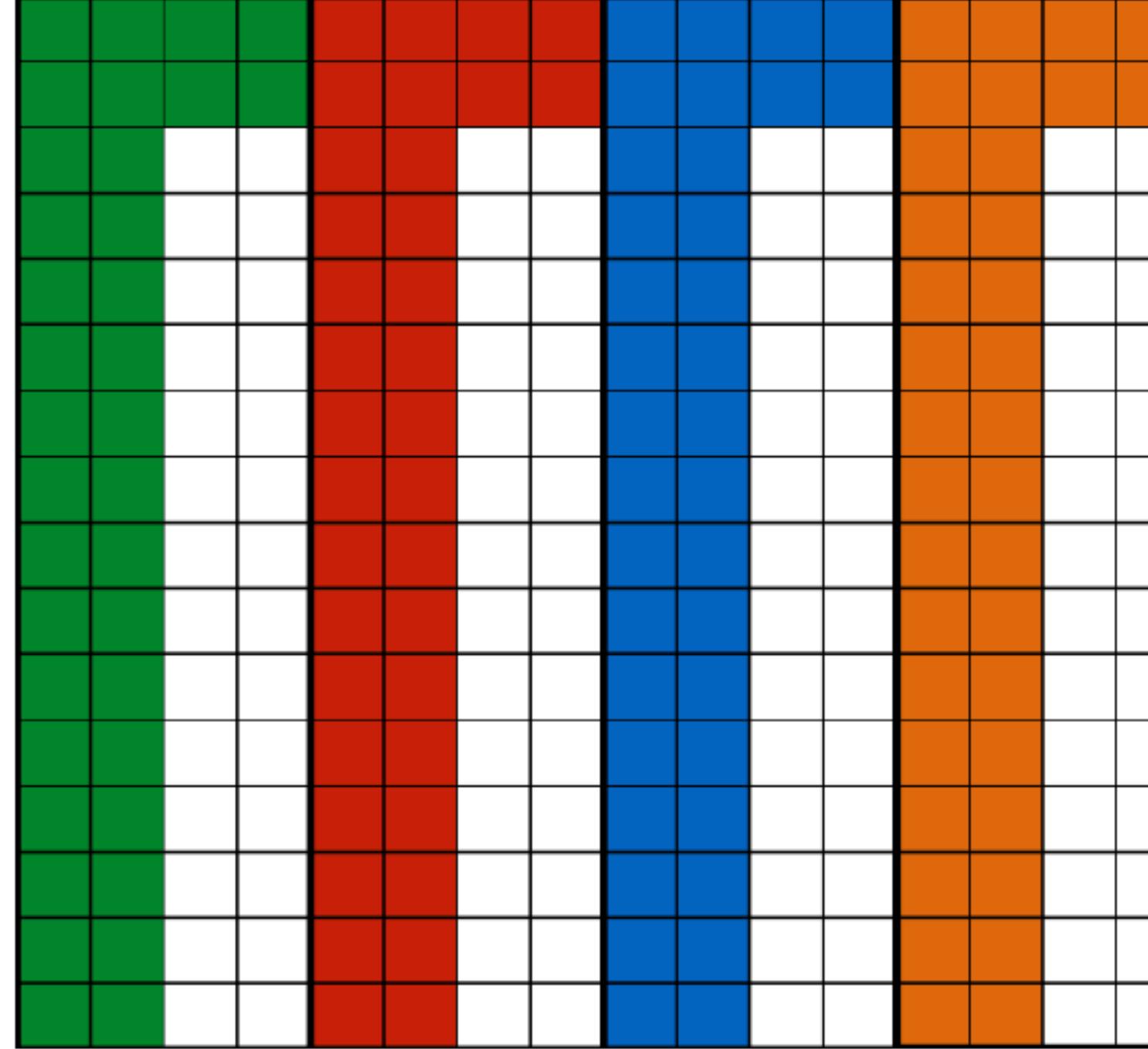
Highlights GOFMM



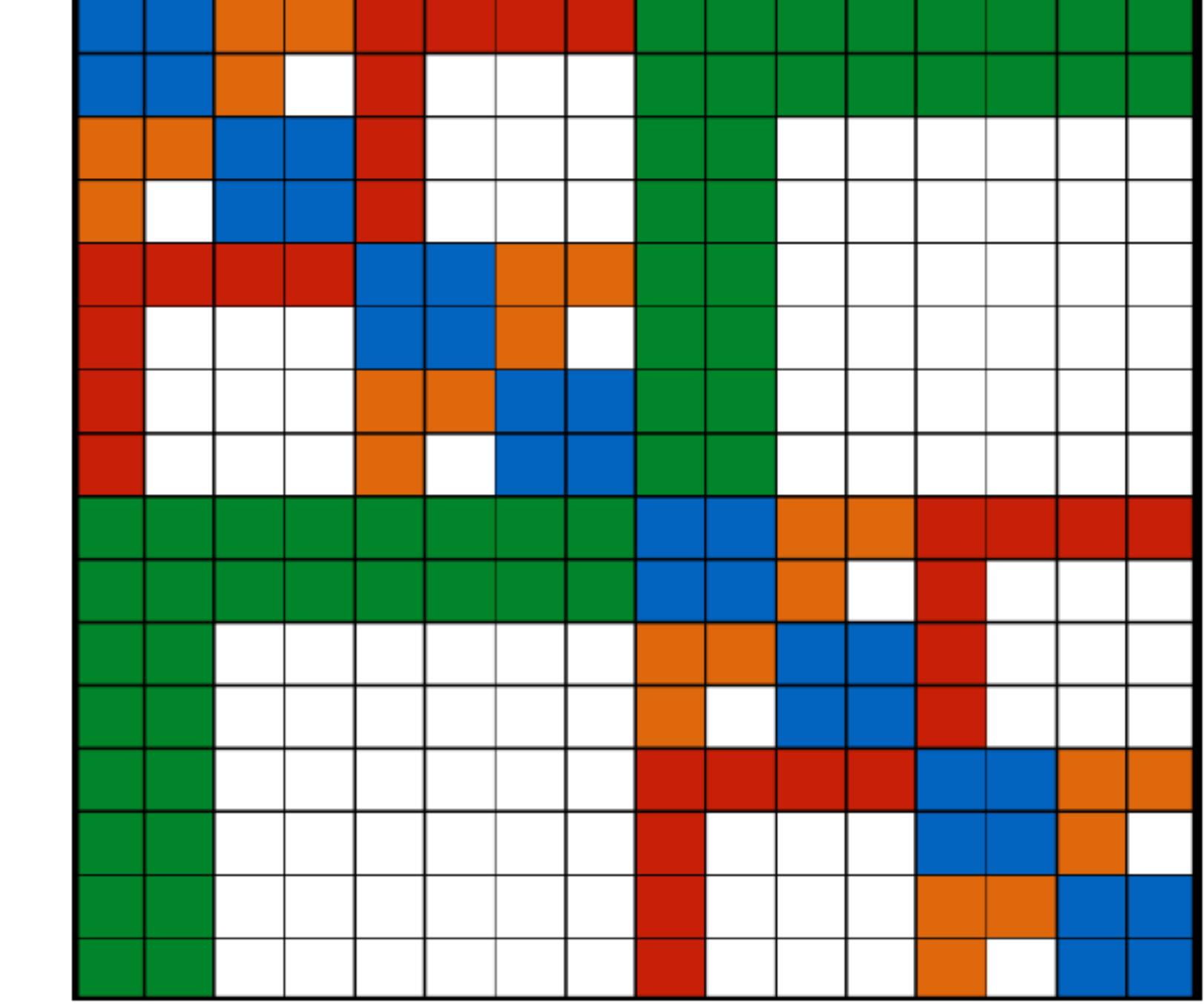
Achieving $O(N \log^a N)$ complexity



NYSTROM



ENSEMBLE NYSTROM



HIERARCHICAL
MATRICES

Sparse + low-rank

1								
	1							
		1						
			1					
				1				
					1			
						1		
							1	
								1

+

1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

=

2	1	1	1	1	1	1	1	1
1	2	1	1	1	1	1	1	1
1	1	2	1	1	1	1	1	1
1	1	1	2	1	1	1	1	1
1	1	1	1	2	1	1	1	1
1	1	1	1	1	2	1	1	1
1	1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	2	1
1	1	1	1	1	1	1	1	2

Identity matrix

rank-1 matrix

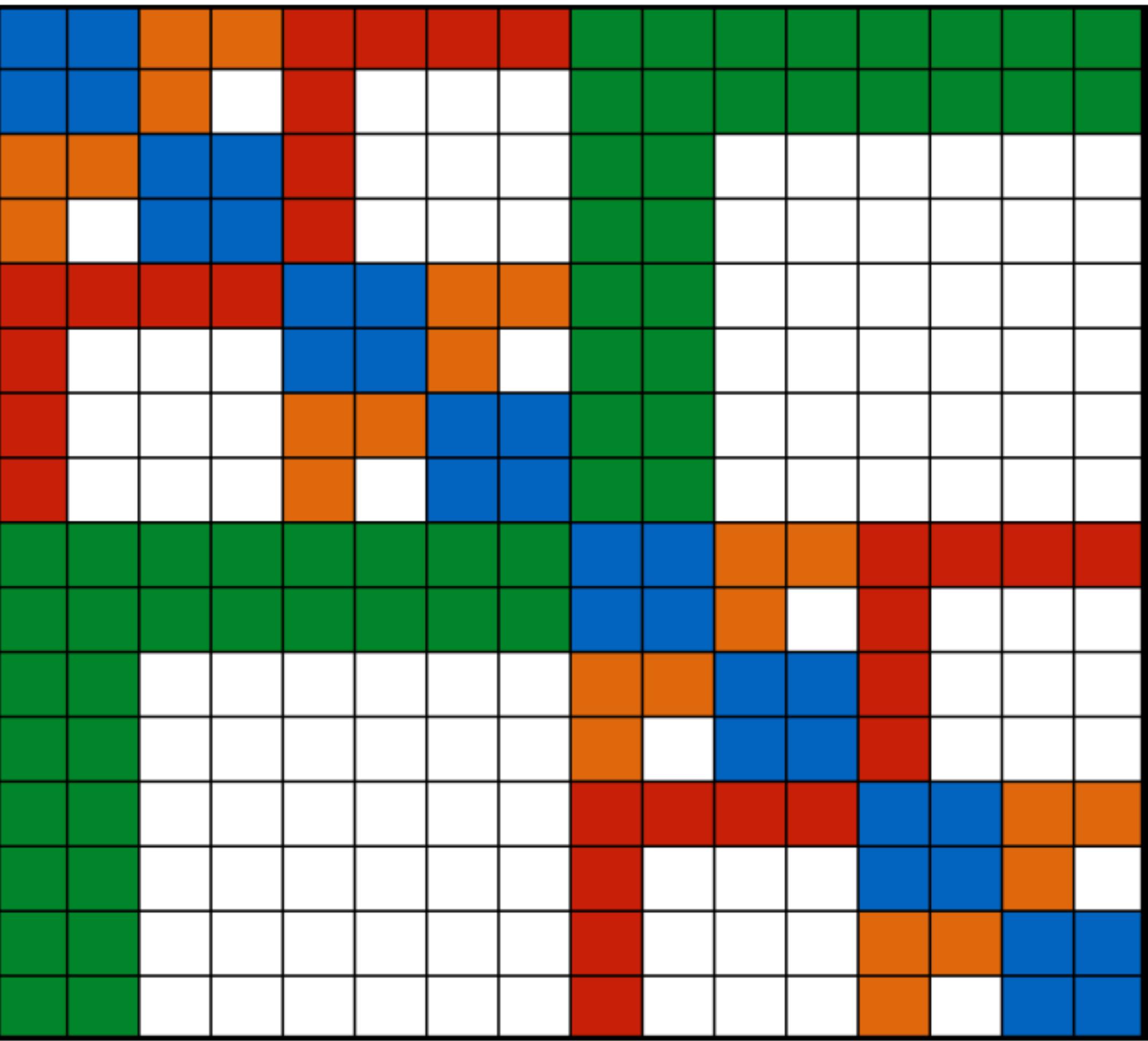
Neither sparse
nor low-rank

Hierarchical matrices, basic idea

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

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

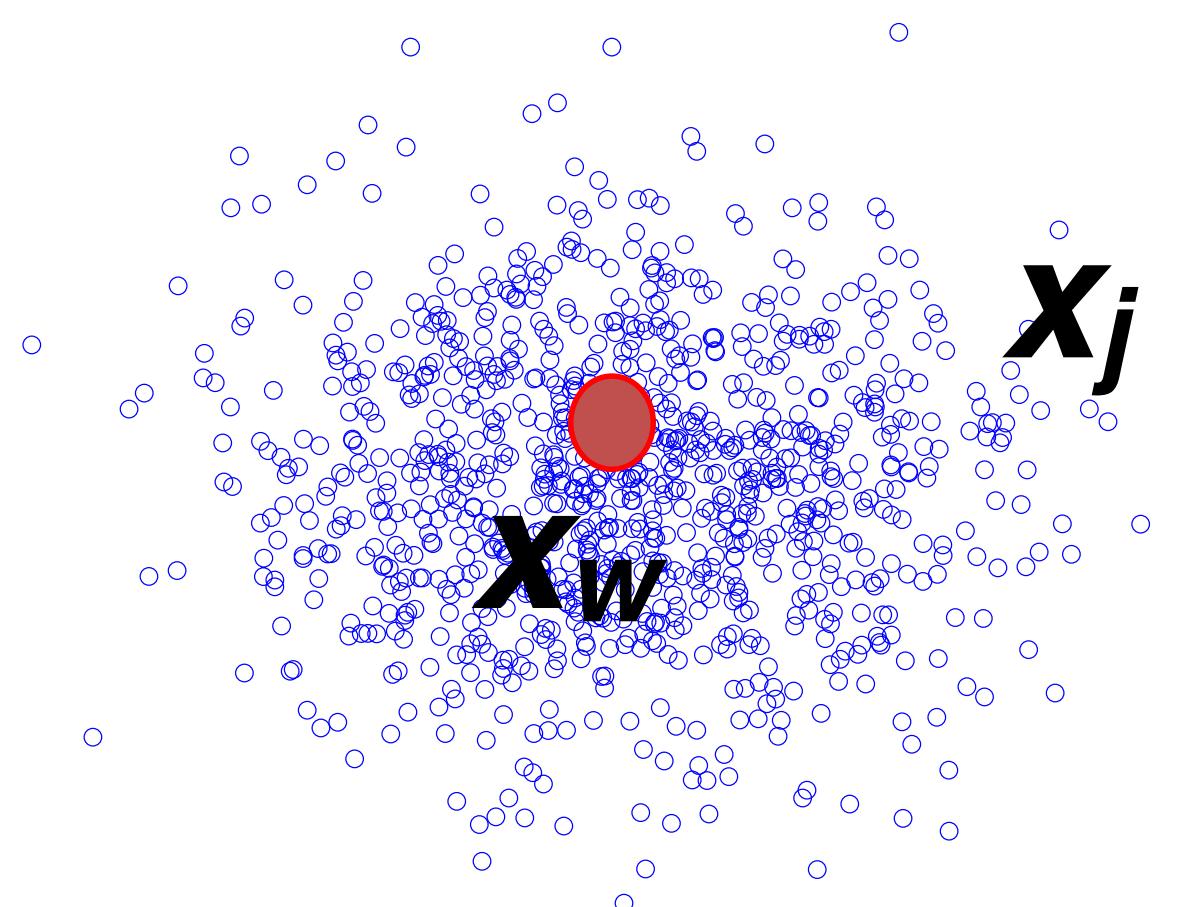
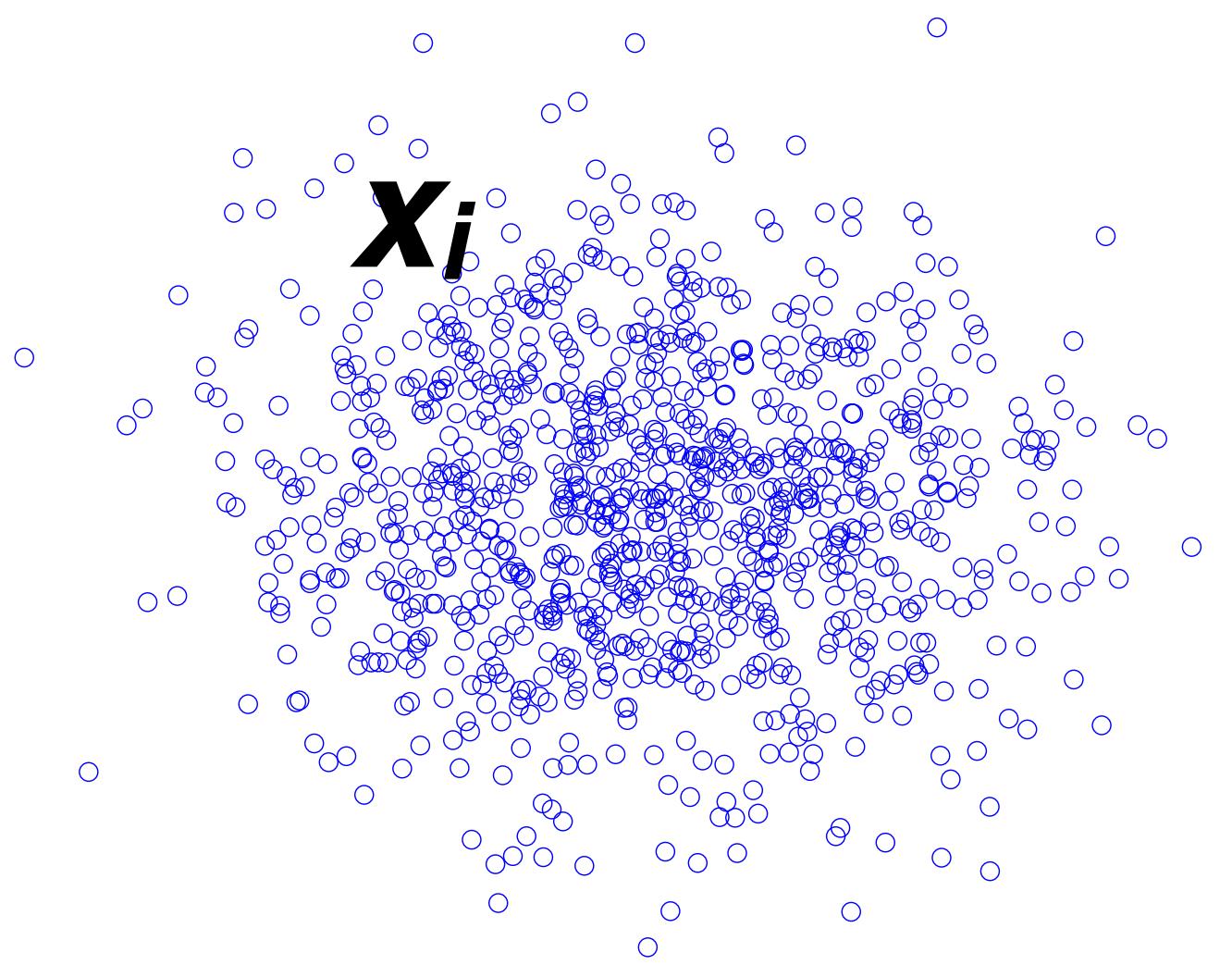
$$\frac{D + UV}{D + UV}$$



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

Constructing the approximation

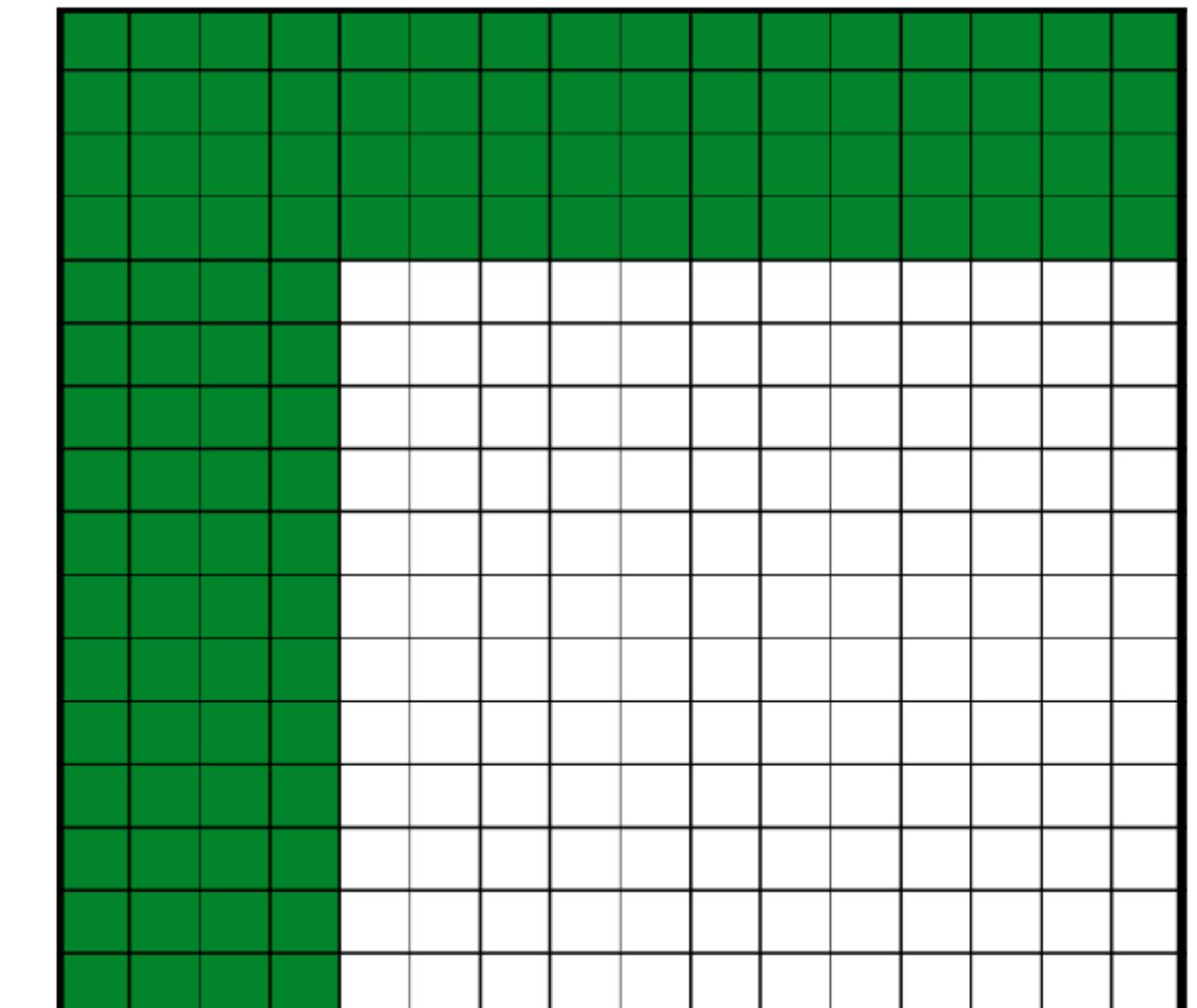
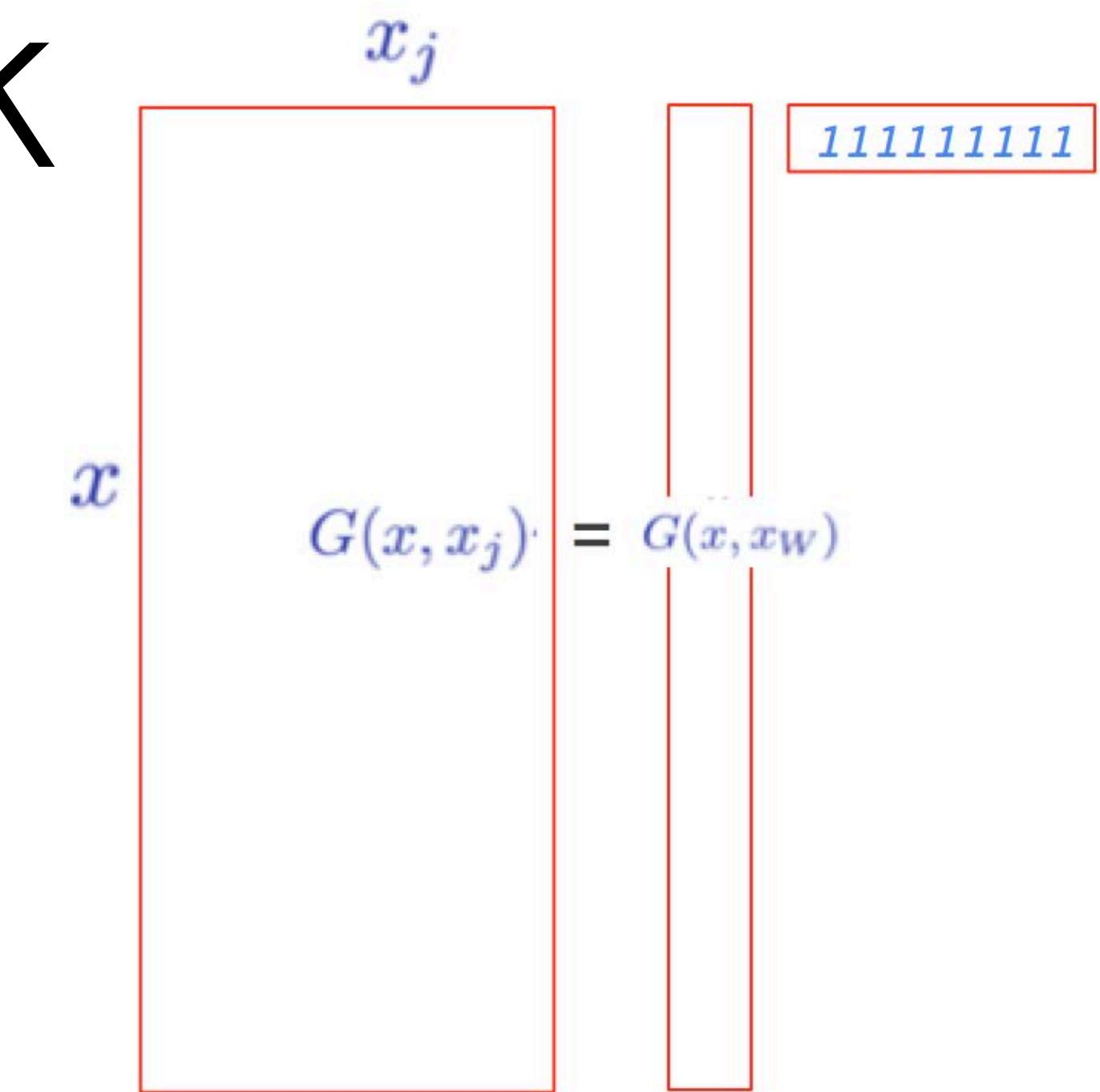
Idea I: far-field \rightarrow low rank



x : Target, x_j : sources
 w_j : weights

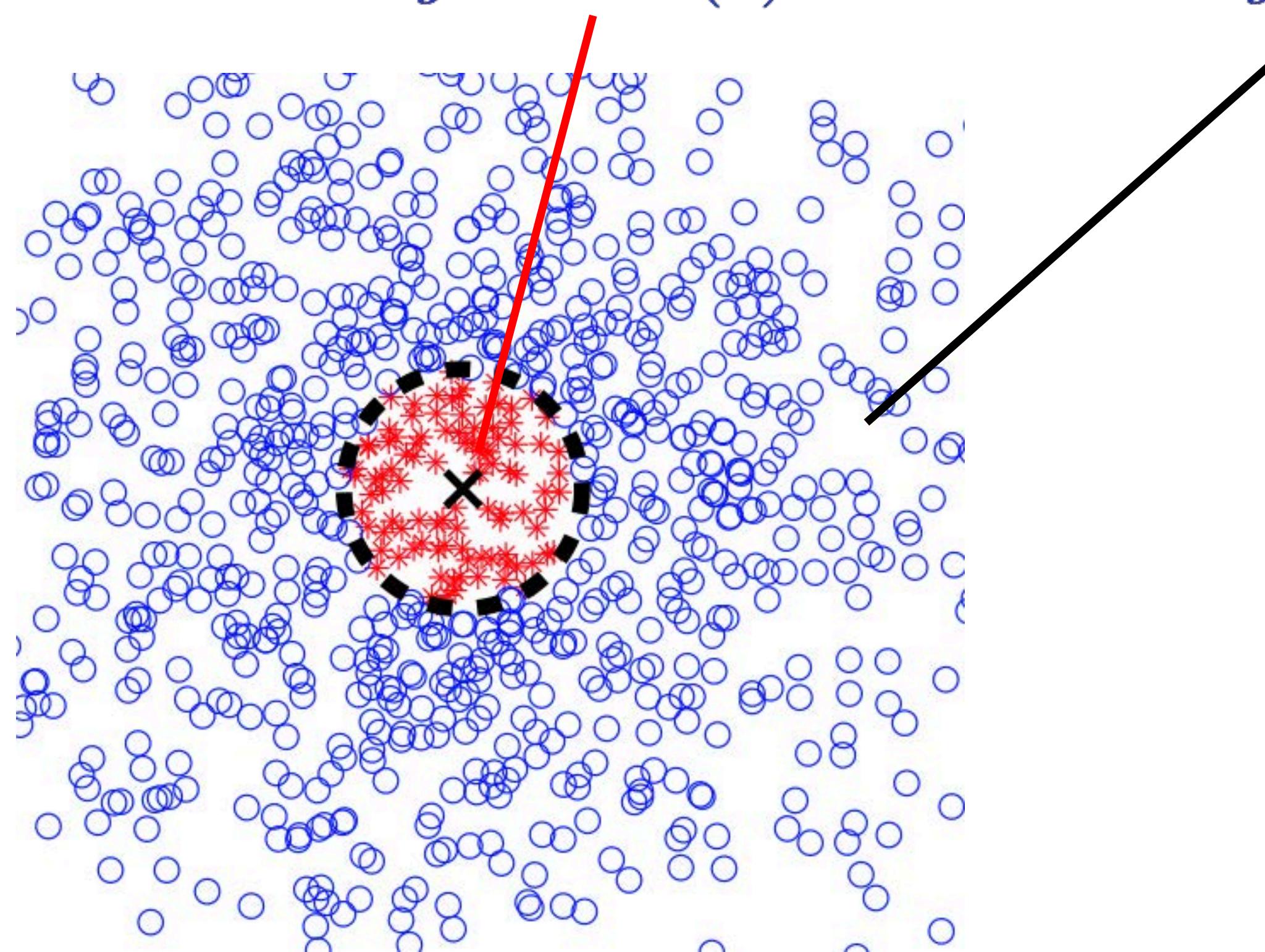
$$u(x) = \sum_j G(x, x_j) w_j$$

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

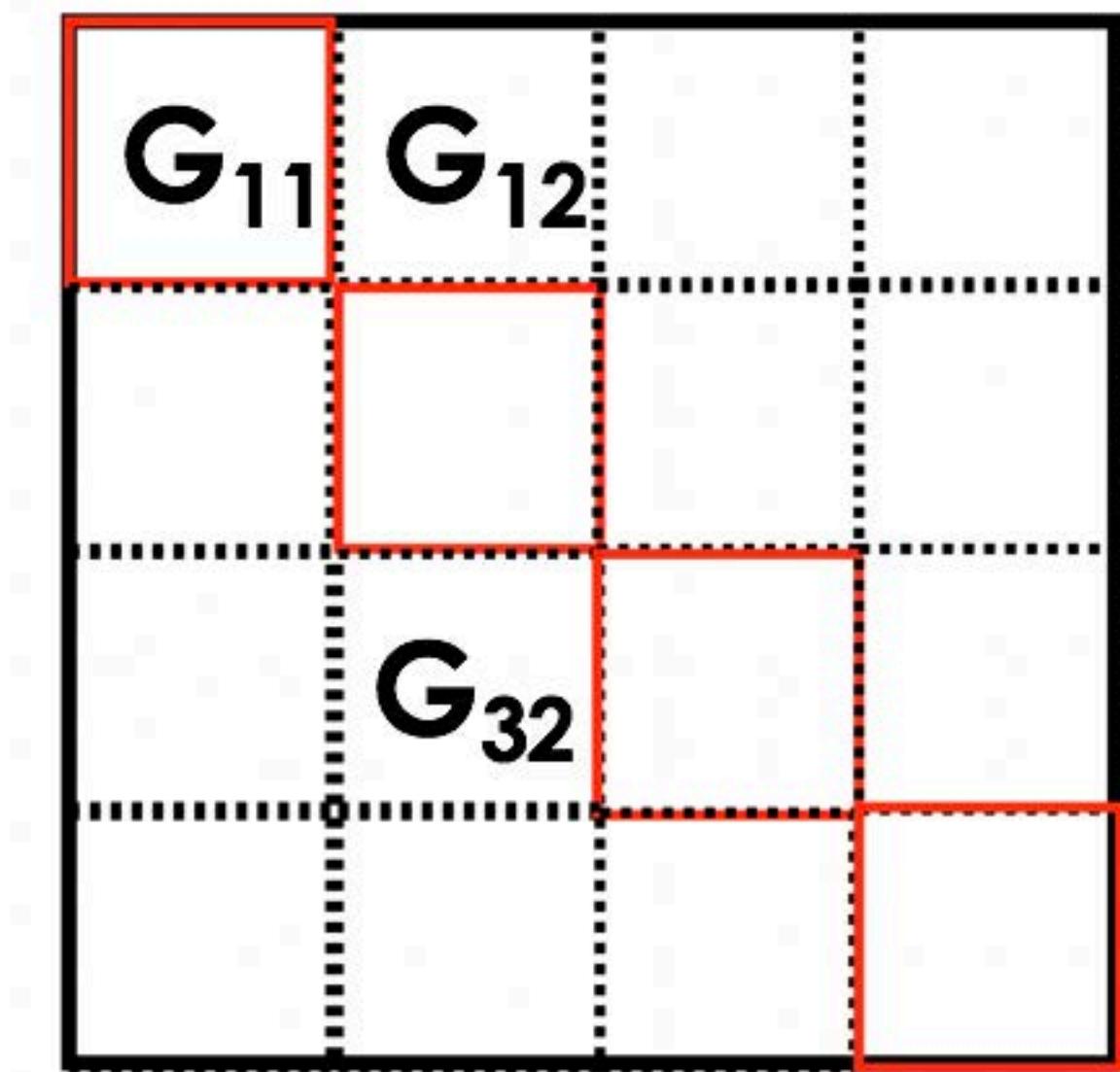
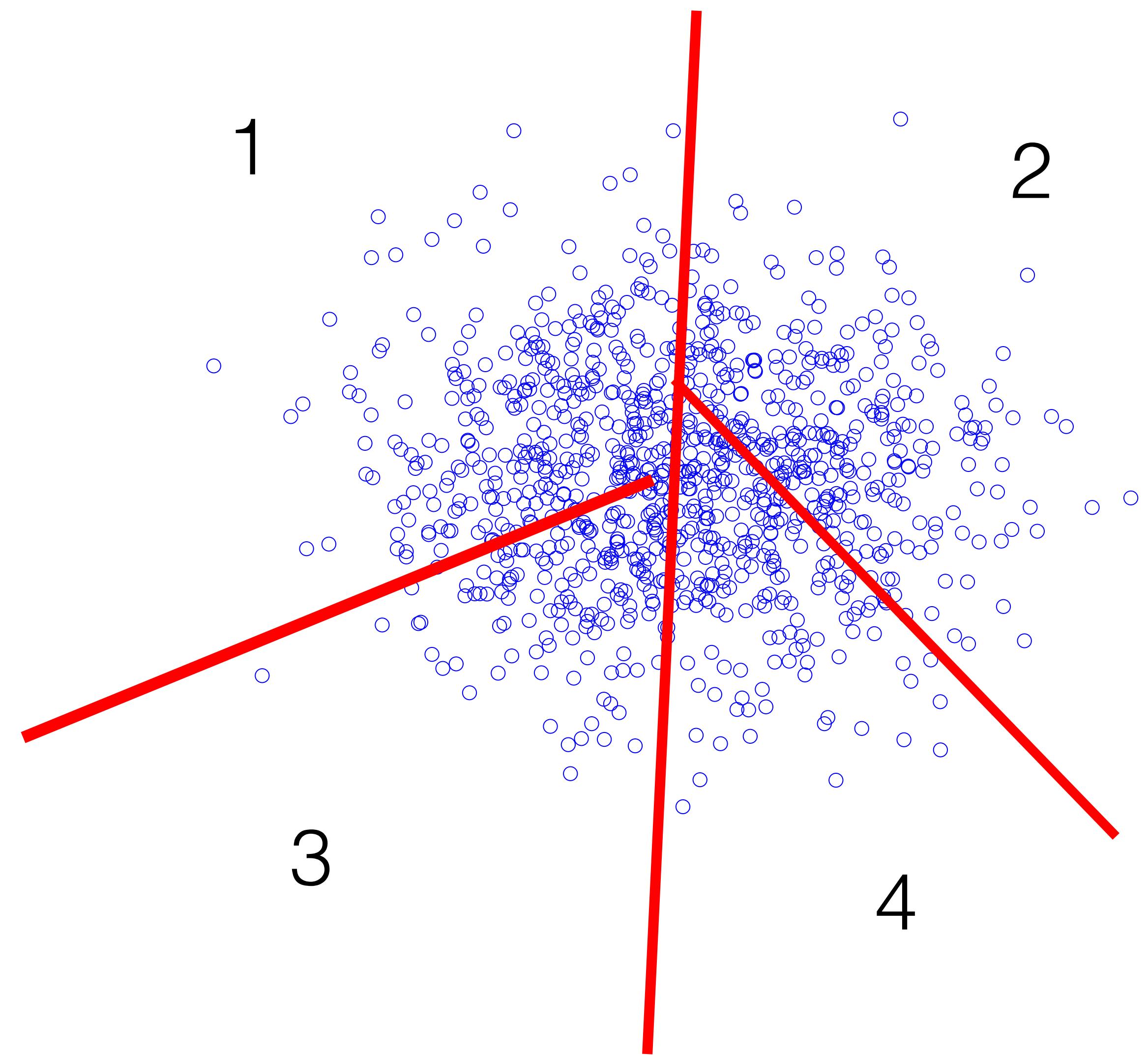


Idea II: Near/Far field split

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



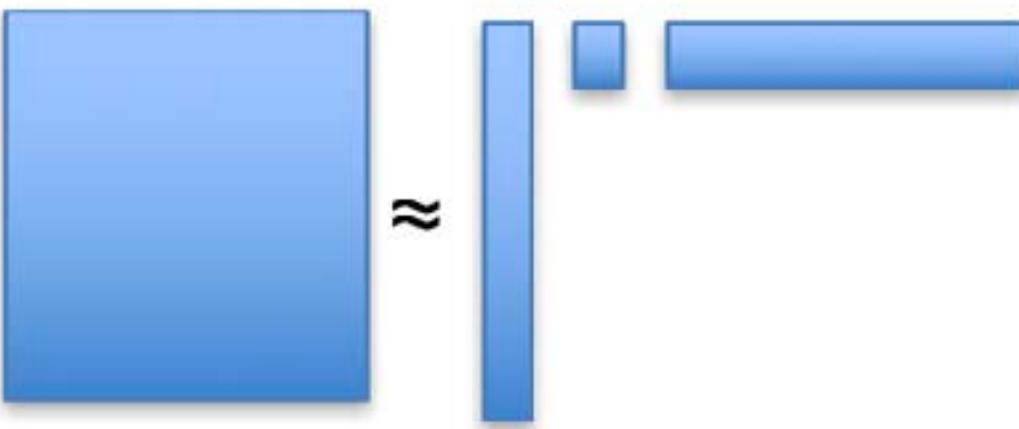
Idea III: recursion



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
probably not the case with increasing N

Randomized linear algebra — Nystrom method



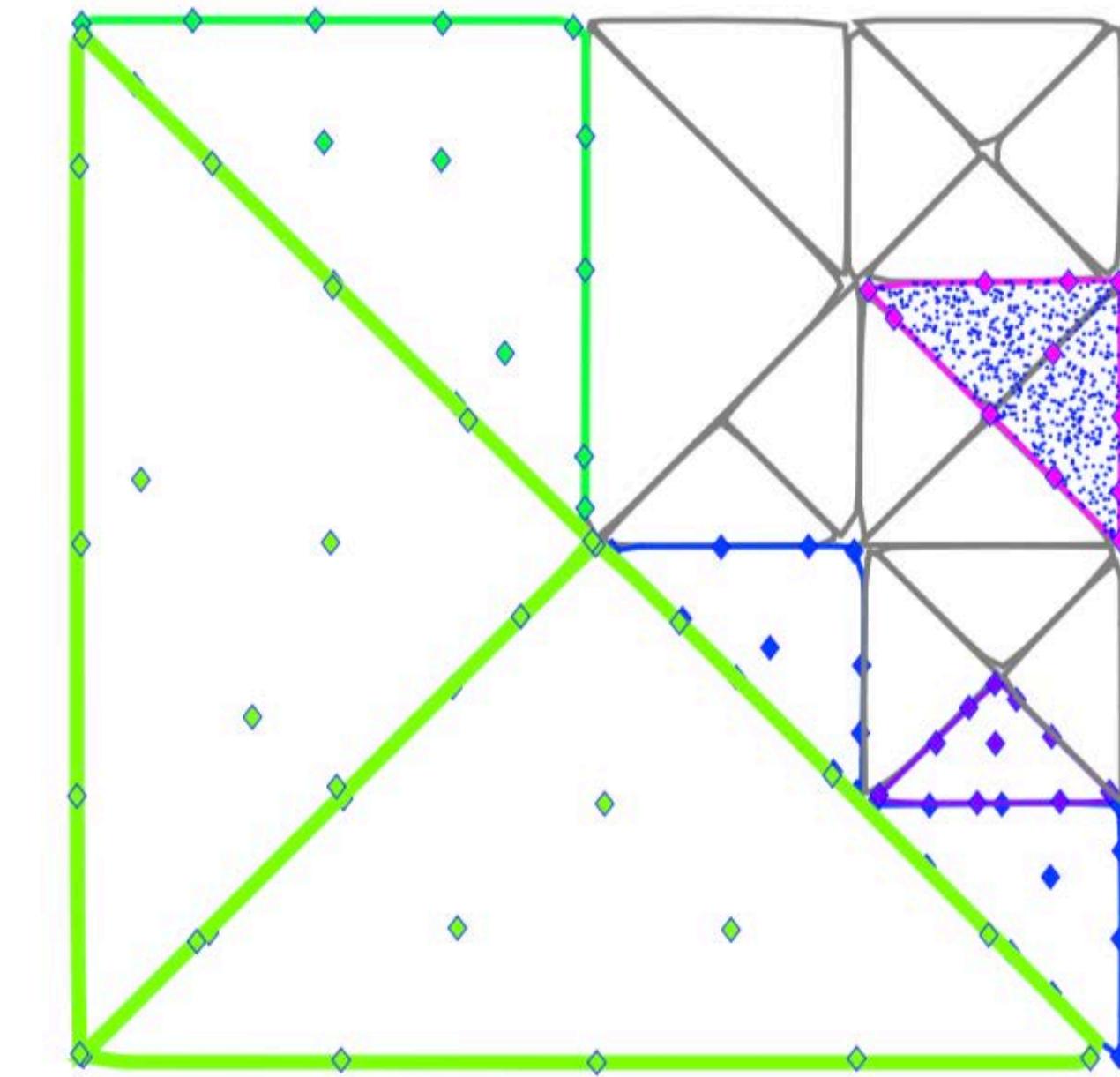
- **Low-rank** decomposition of G
- Random sampling of $O(s)$ points, \mathbf{s} : target **rank**

$$G \approx \tilde{G} = G_{Ns} G_{ss}^{-1} G_{sN}$$

- Work $Ns + s^3$
- Error $\|G - \tilde{G}\| \leq \sqrt{1 + 6N/s} \sigma_{s+1}(G)$

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

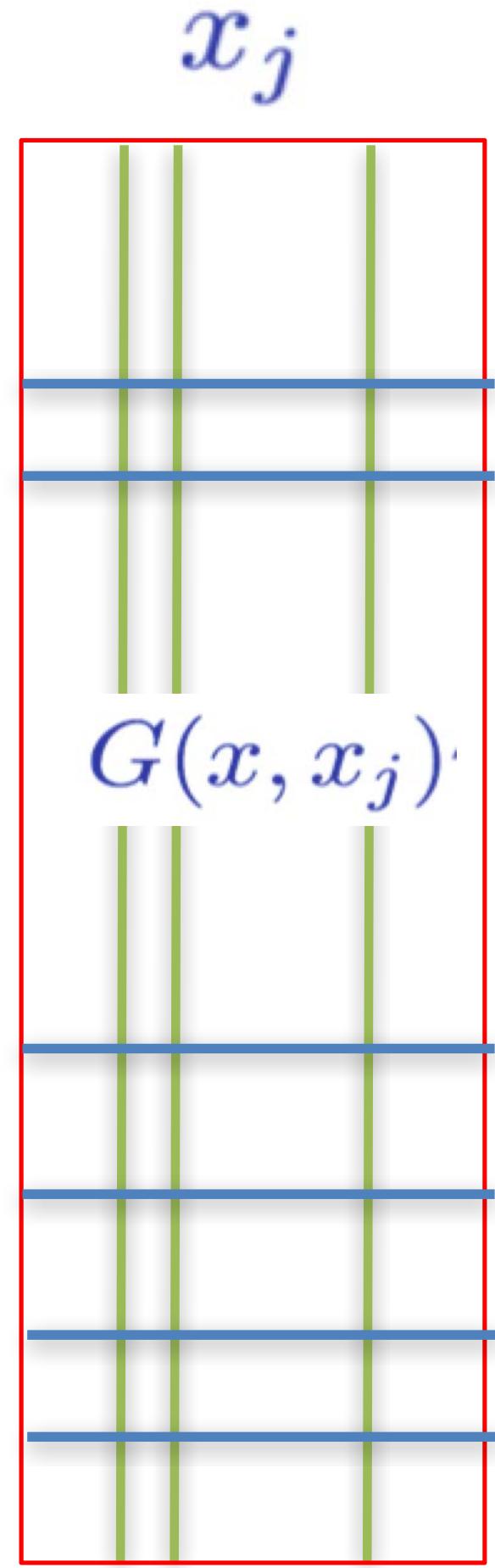


SISC'15,16
ACHA'15
KDD'15
SC'15
IPDPS'15,16,17

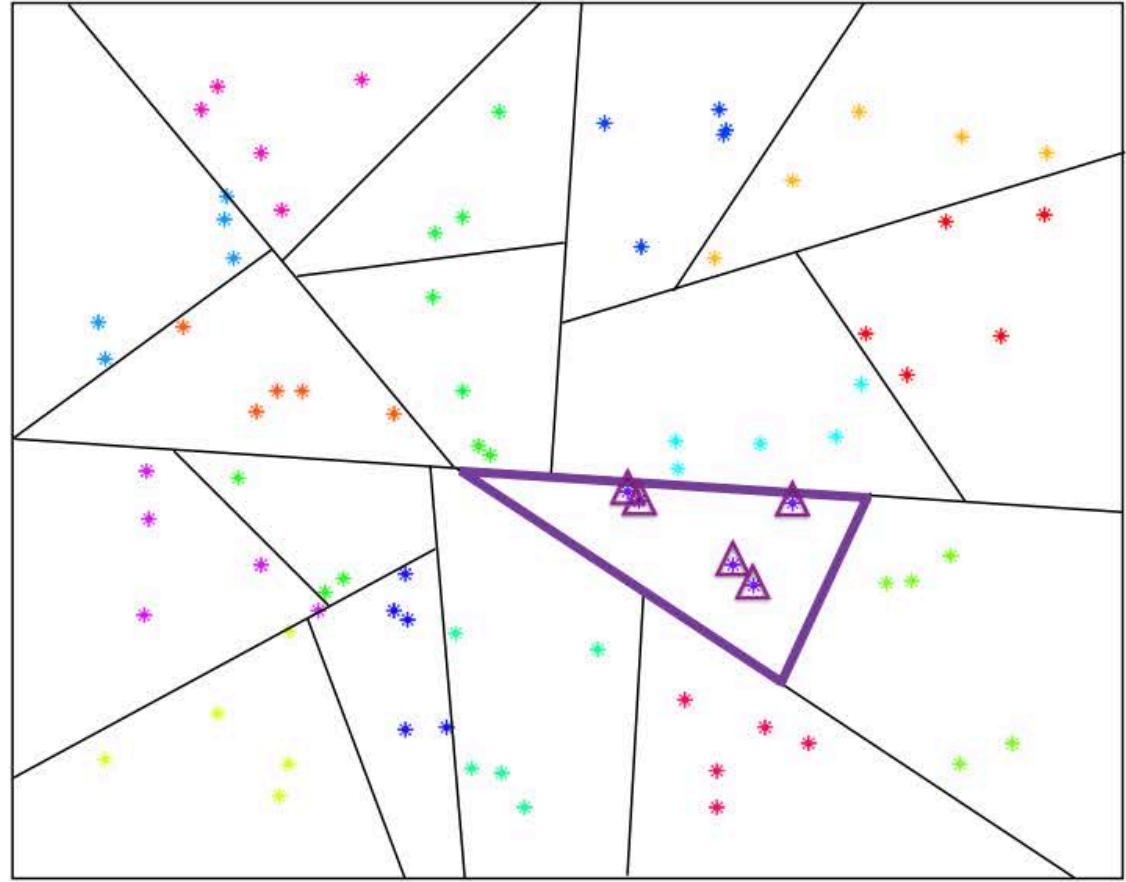
Far-field s -rank approximation

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

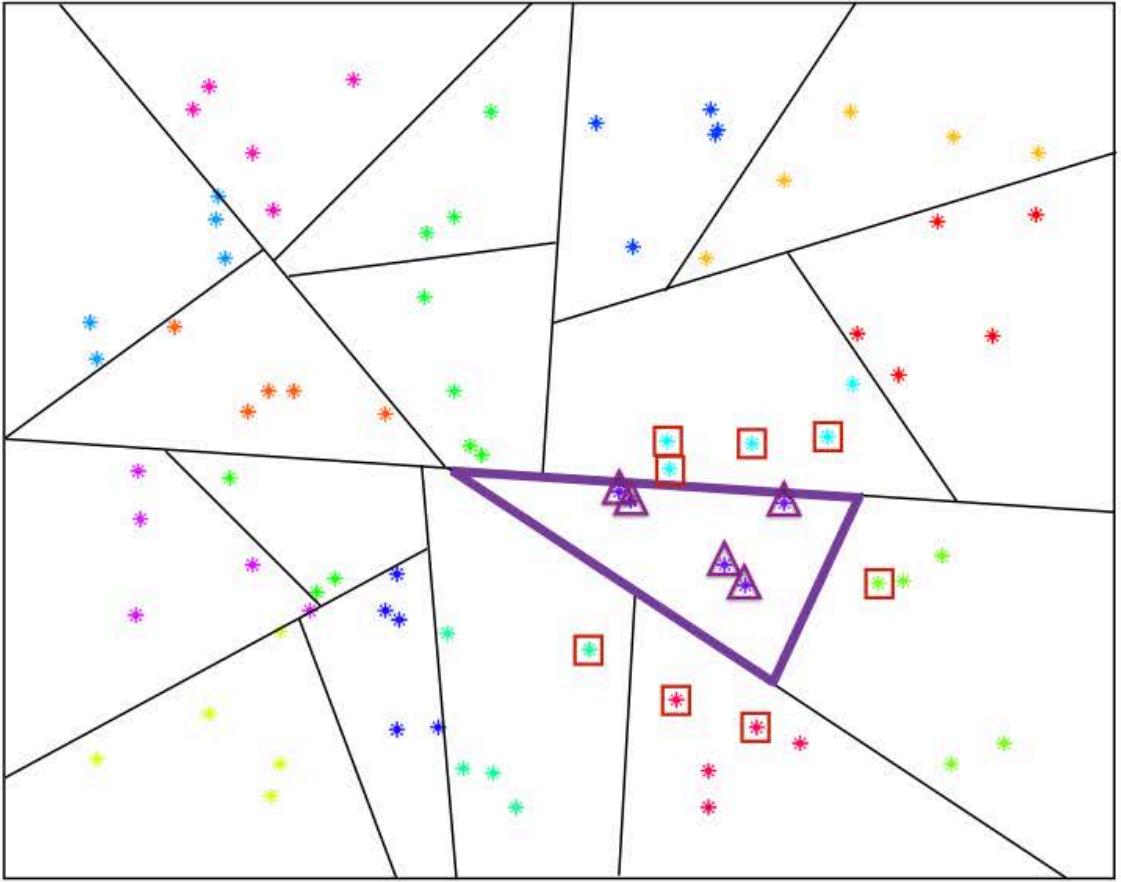
- SVD is too expensive — use sampling
- Sample rows
leverage, norm, range-space
- Interpolative decomposition
- ASKIT: approximate norm *adaptive* sampling
using nearest-neighbors + *adaptive* rank selection



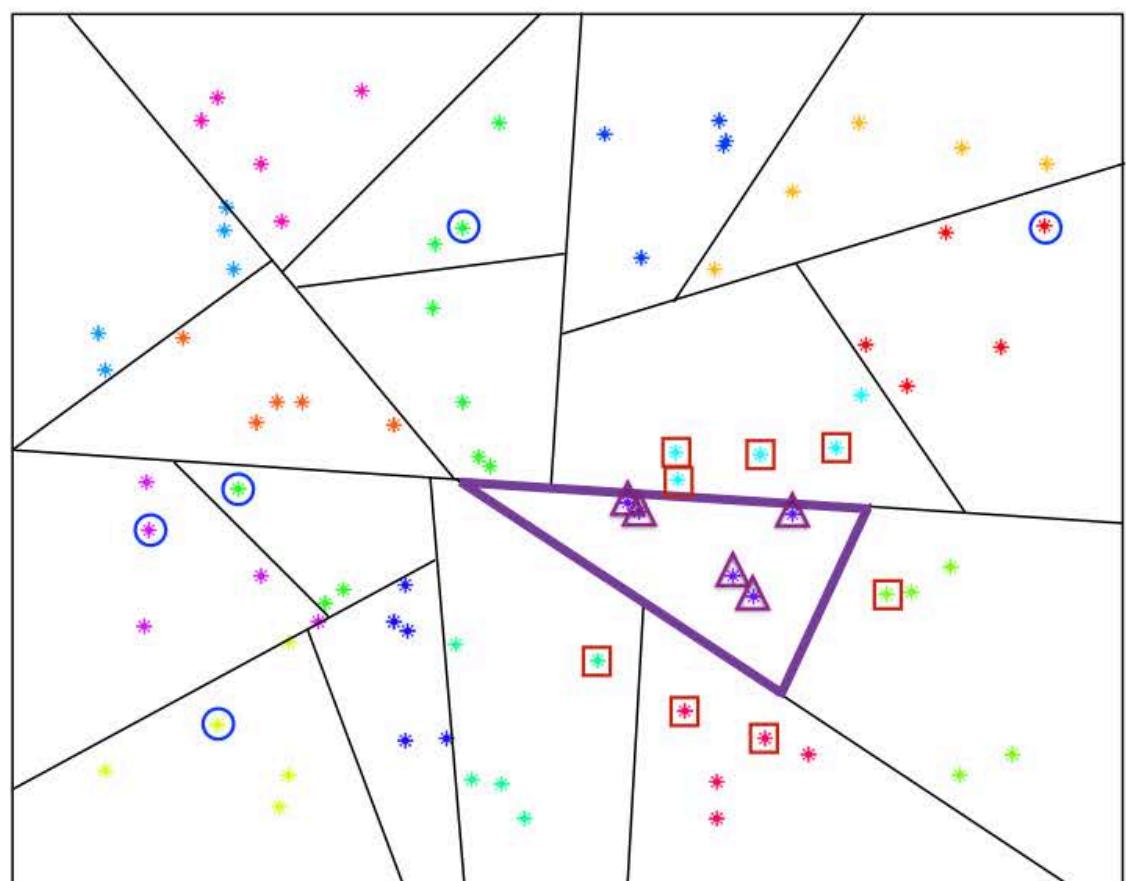
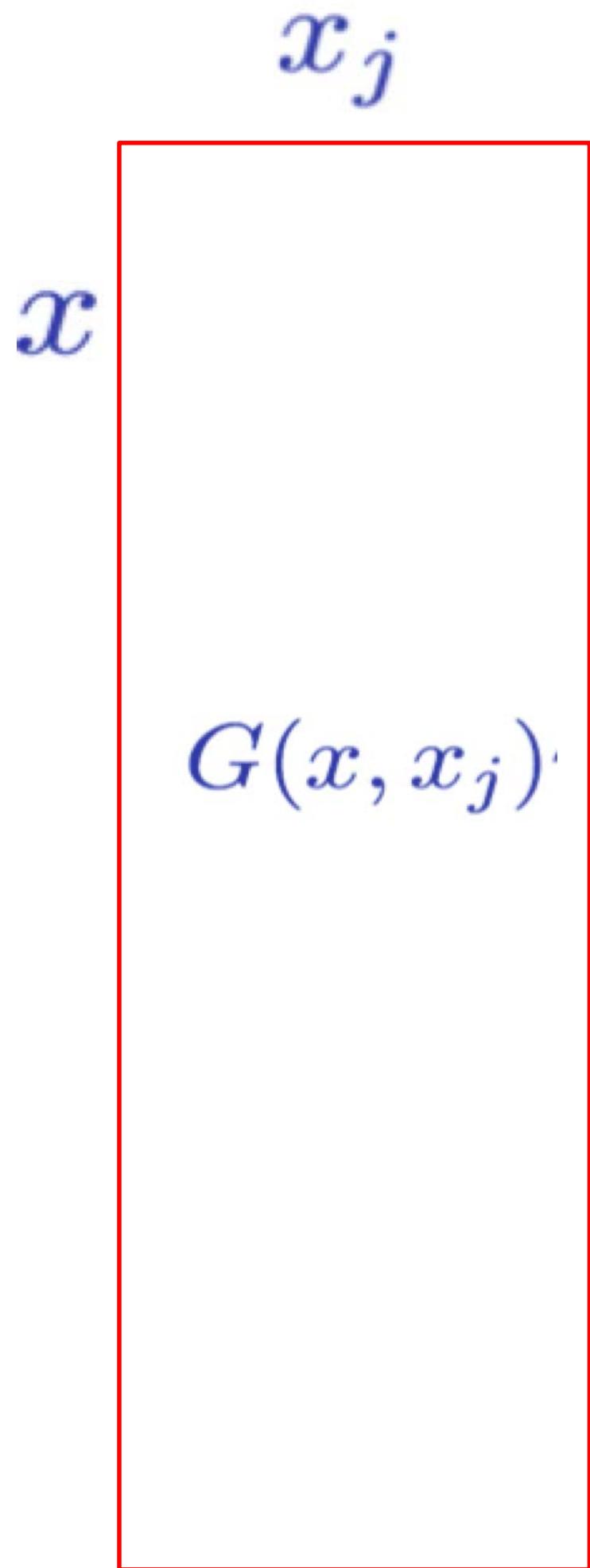
Skeletonization (far field approximation)



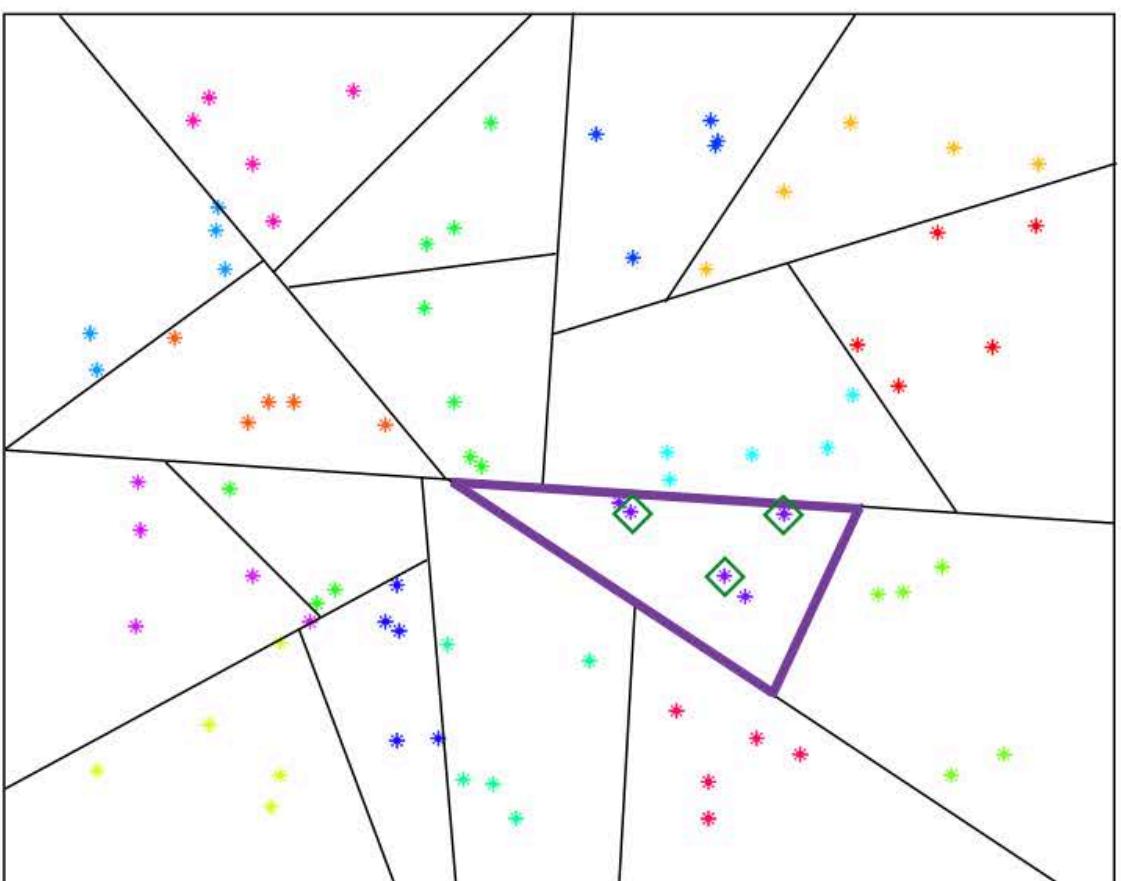
(a) A leaf node.



(b) The node's nearest neighbors.

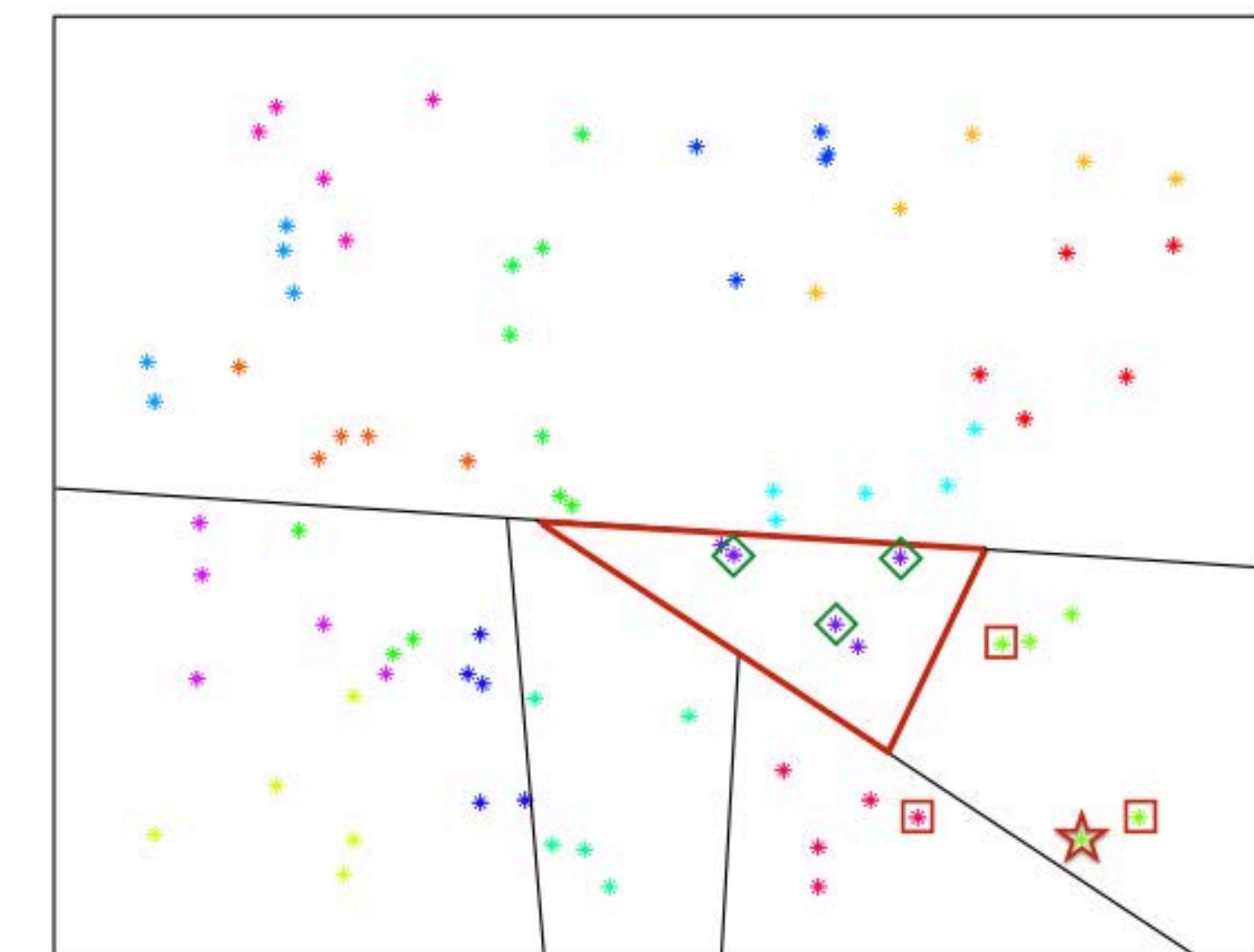
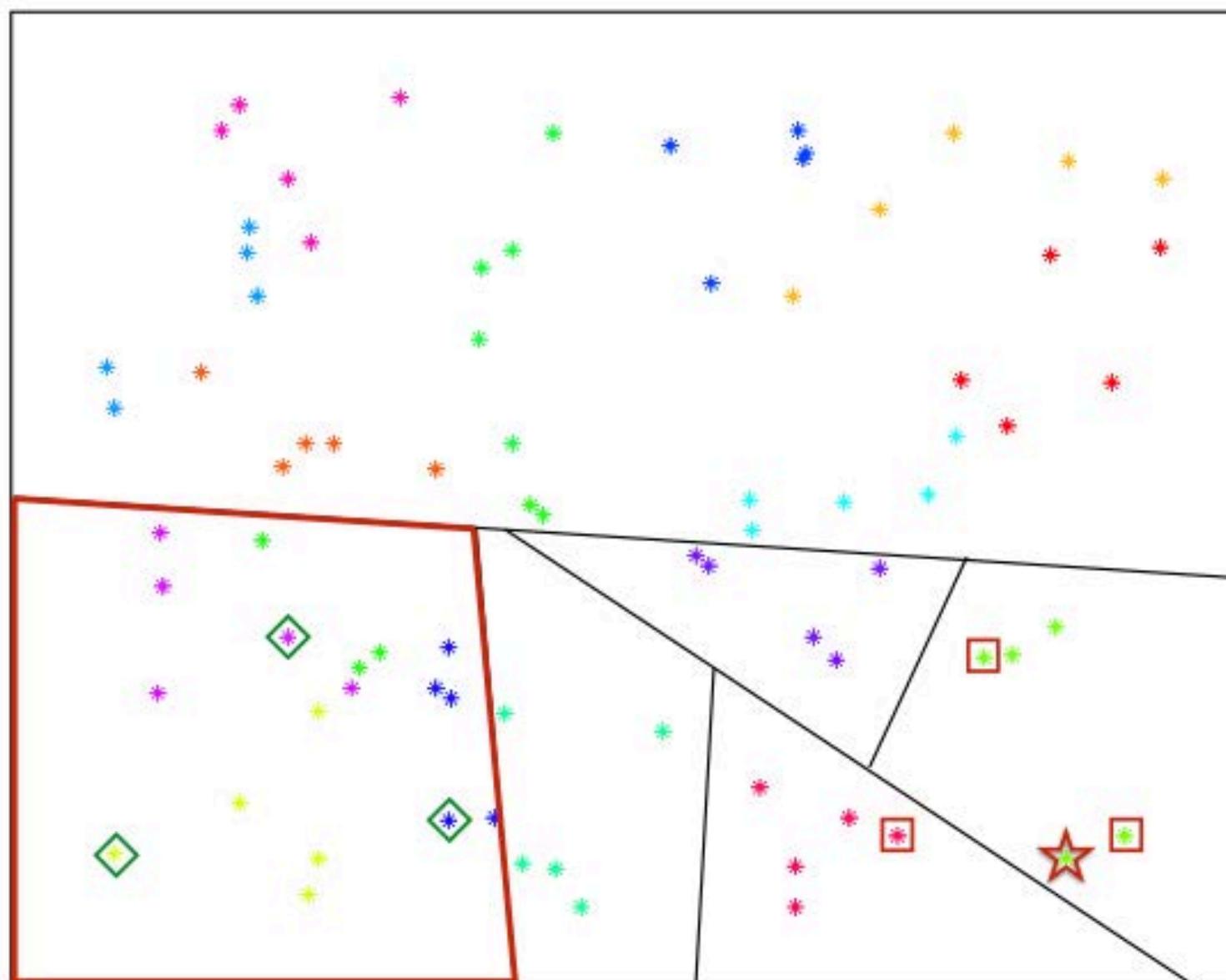
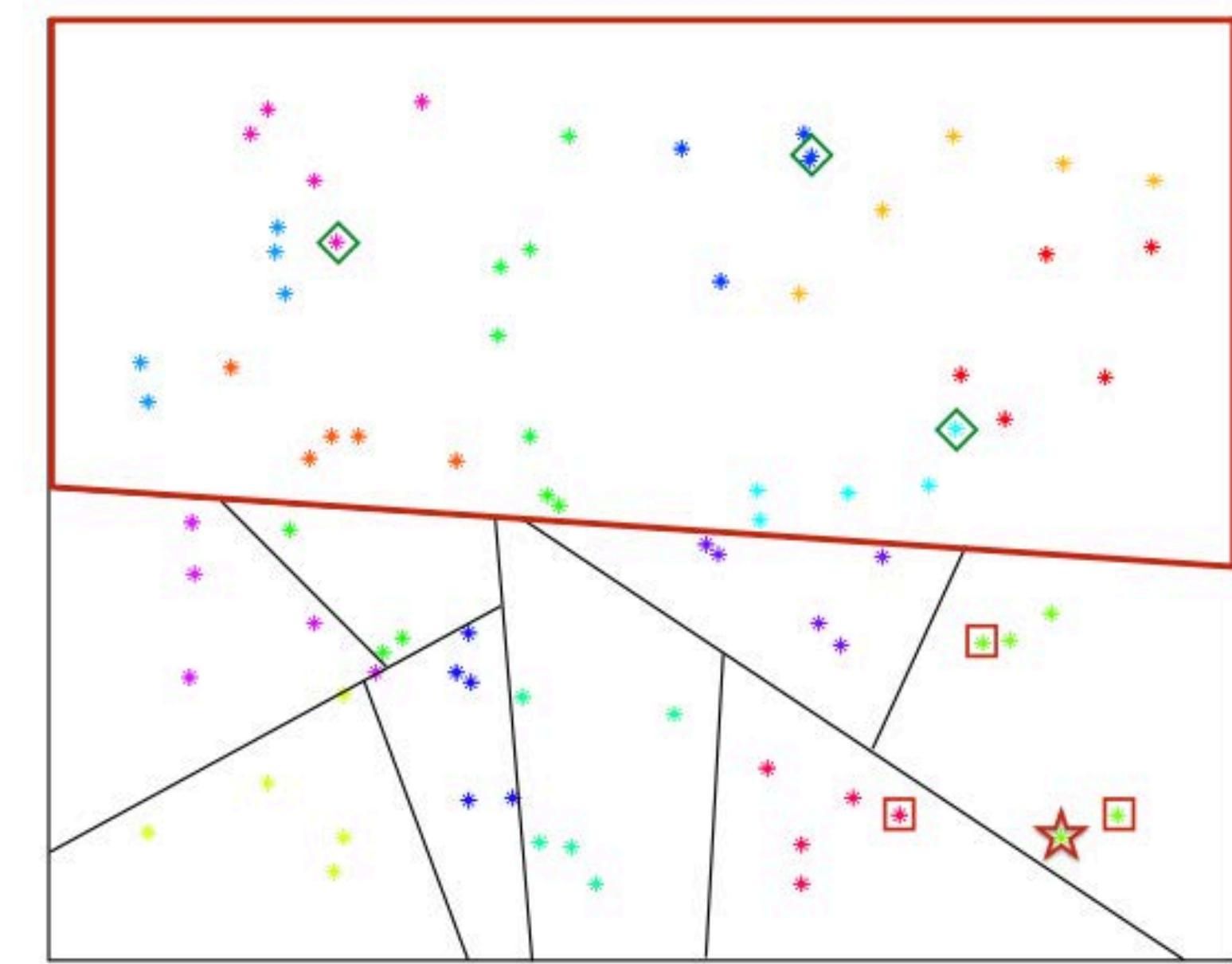
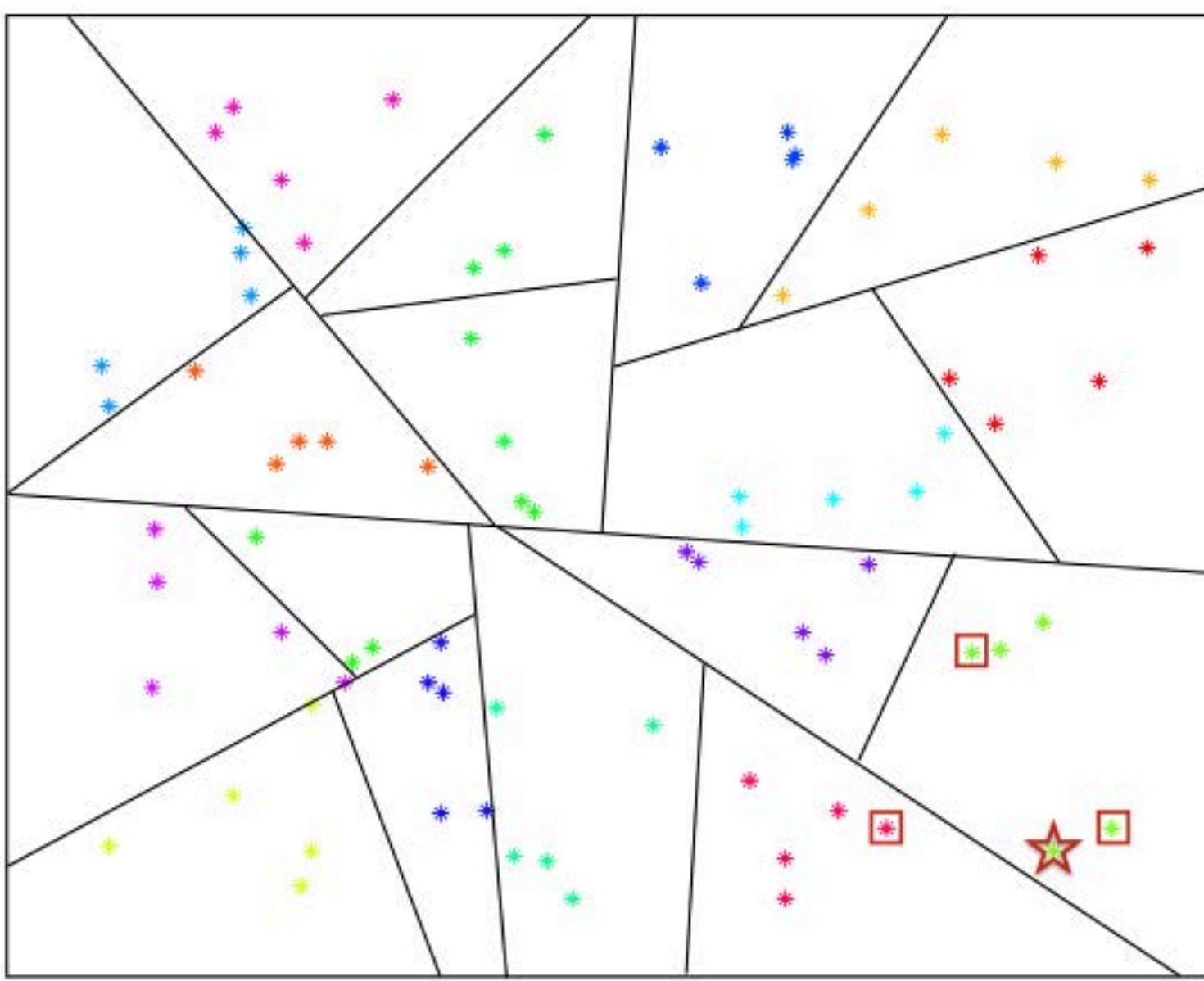


(c) Sampling distant points.

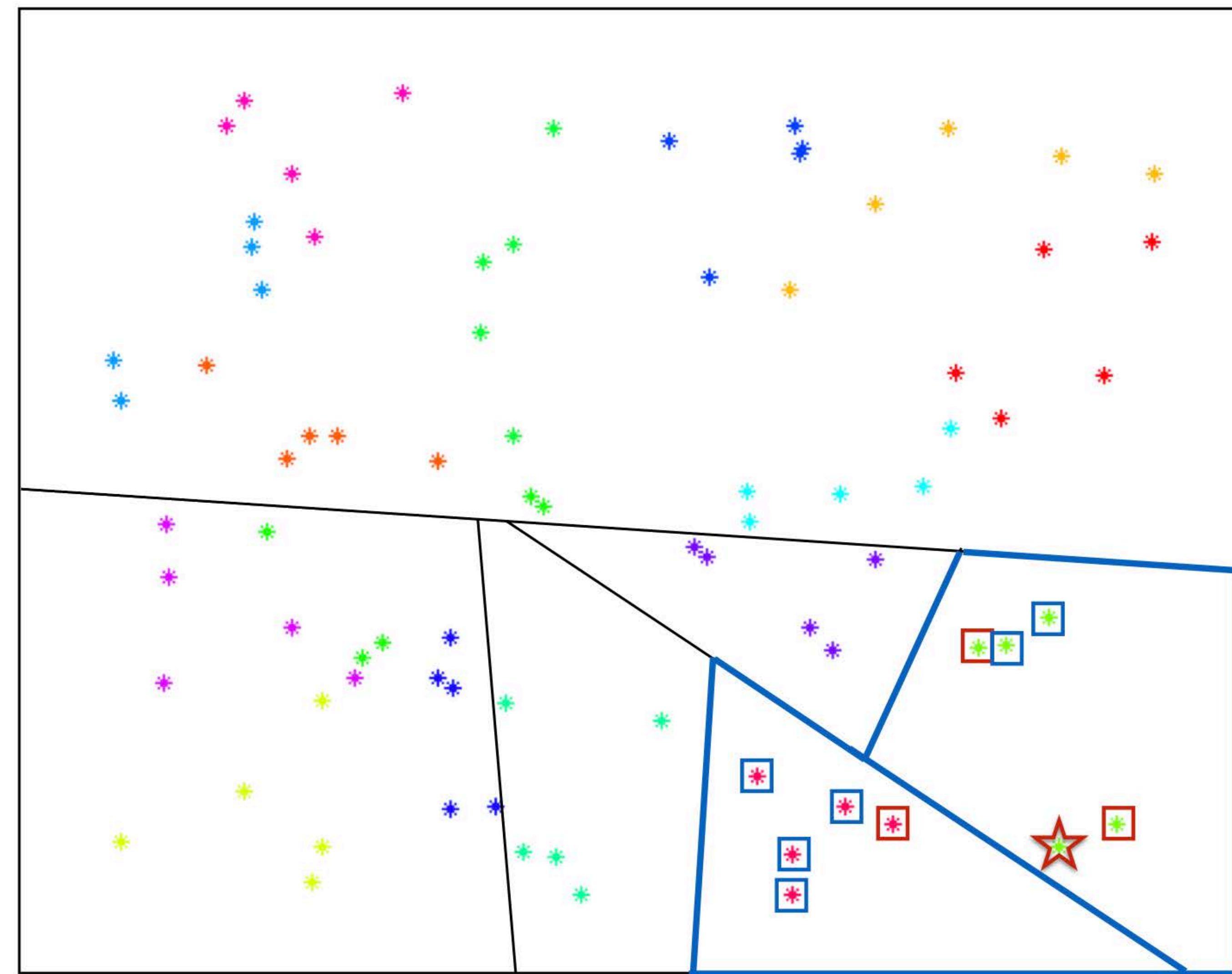


(d) The resulting skeleton.

Evaluation



Evaluation



Complexity and error

- Work RAM skeletonize evaluate
 $(d + \kappa)N$ Ns^2 $dNs\kappa \log\left(\frac{N}{s}\right)$
- Error $\|G - \tilde{G}\| \leq \sqrt{1 + 6N/s} \log(N/s)$ off-diagonal
 $\gamma_{s+1} \sigma_{s+1}$
- Nystrom $\|G - \tilde{G}\| \leq \sqrt{1 + 6N/s} \sigma_{s+1}$
 $Ns + s^3$ diagonal

Parallel complexity

$$\text{Points per MPI task } n = \frac{N}{p}$$

$$\text{Tree depth } D = \log \frac{N}{s}$$

$$\text{Tree construction} \leq (t_s + t_w) \log^2 p \log N + (t_w \log p) (d + k) n$$

$$\text{Skeletonization} \leq t_f \left(\frac{n}{s} + \log p \right) s^3$$

$$\text{Evaluation} \leq t_s p + (t_w + t_f) d k s D n$$

Summary of ASKIT features

- Binary tree for matrix permutation
- 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 acceleration

Data	N	d	ϵ_2	%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

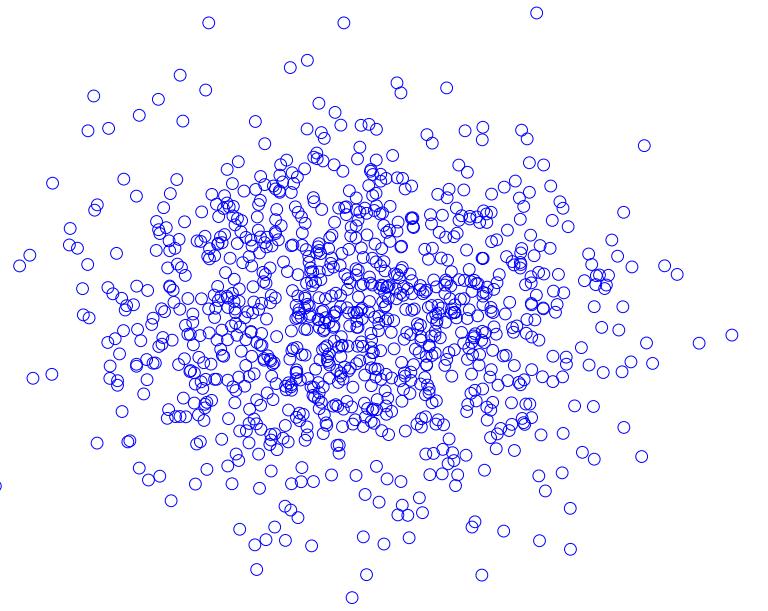
Kernel regression scaling

MNIST dataset for OCR
strong scaling, 8M points d=784

0000000000000000
1111111111111111
2222222222222222
3333333333333333
4444444444444444
5555555555555555
6666666666666666
7777777777777777
8888888888888888
9999999999999999

#cores	512	2,048	4,096	8,192	16,384
Skel. (Alg. 2)	1,295	465	370	305	269
Lists (Alg. 3)	729	177	87	46	23
LET (Eq. 7)	273	136	107	87	71
Eval. (Eq. 14)	157	67	42	28	23
Total	2,471	862	621	483	394
Efficiency	1.00	0.72	0.50	0.32	0.20

Related work



- 2D & 3D N-body
Barnes & Hut, Greengard & Rokhlin,
Darve, Hackbusch / Novak / Bebendorf
- High-D
Griebel, Duraiswami, Vuduc / Gray,
Kondor, Mahoney & Darve
March / Bo / Biros [ASKIT]
- Purely algebraic
Li et al / STRUMPACK
Darve et al / HODLR

$$K_{ij} = K(x_i, x_j) = K(x_i - x_j)$$

Gaussian	$\exp(-\ x - x_j\ ^2/(2h^2))$
Laplace	$\ x - x_j\ ^{2-d}, d > 2$
Matern	$(\sqrt{2\nu}\ x - x_j\)^{\nu} K_{\nu}(\sqrt{2\nu}\ x - x_j\)$
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	$1/\sqrt{c^2 + \ x - x_j\ _2^2}$

METHOD	MATRIX	LOW-RANK	PERM	S
FMM [10]	$\mathcal{K}(x_i, x_j)$	EXP	OCTREE	Y
KIFMM [45]	$\mathcal{K}(x_i, x_j)$	EQU	OCTREE	Y
BBFMM [15]	$\mathcal{K}(x_i, x_j)$	EQU	OCTREE	Y
HODLR [3]	K_{ij}	ALG	NONE	N
STRUMPACK [38]	K_{ij}	ALG	NONE	N
ASKIT [33]	$\mathcal{K}(x_i, x_j)$	ALG	TREE	Y
MLPACK [13]	$\mathcal{K}(x_i, x_j)$	EQU	TREE	Y
GOFMM	K_{ij}	ALG	TREE	Y

Geometry oblivious FMM

- Permute matrix to expose low-rank structure
- Geometry-based algorithms: need distance between indices i, j
- Gram vectors
distance(i, j) = function(K_{ii}, K_{ij}, K_{jj})

Geometry oblivious FMM

- Gram vectors (K is SPD): $K_{ij} = \phi_i \cdot \phi_j$
- Distances

Euclidean

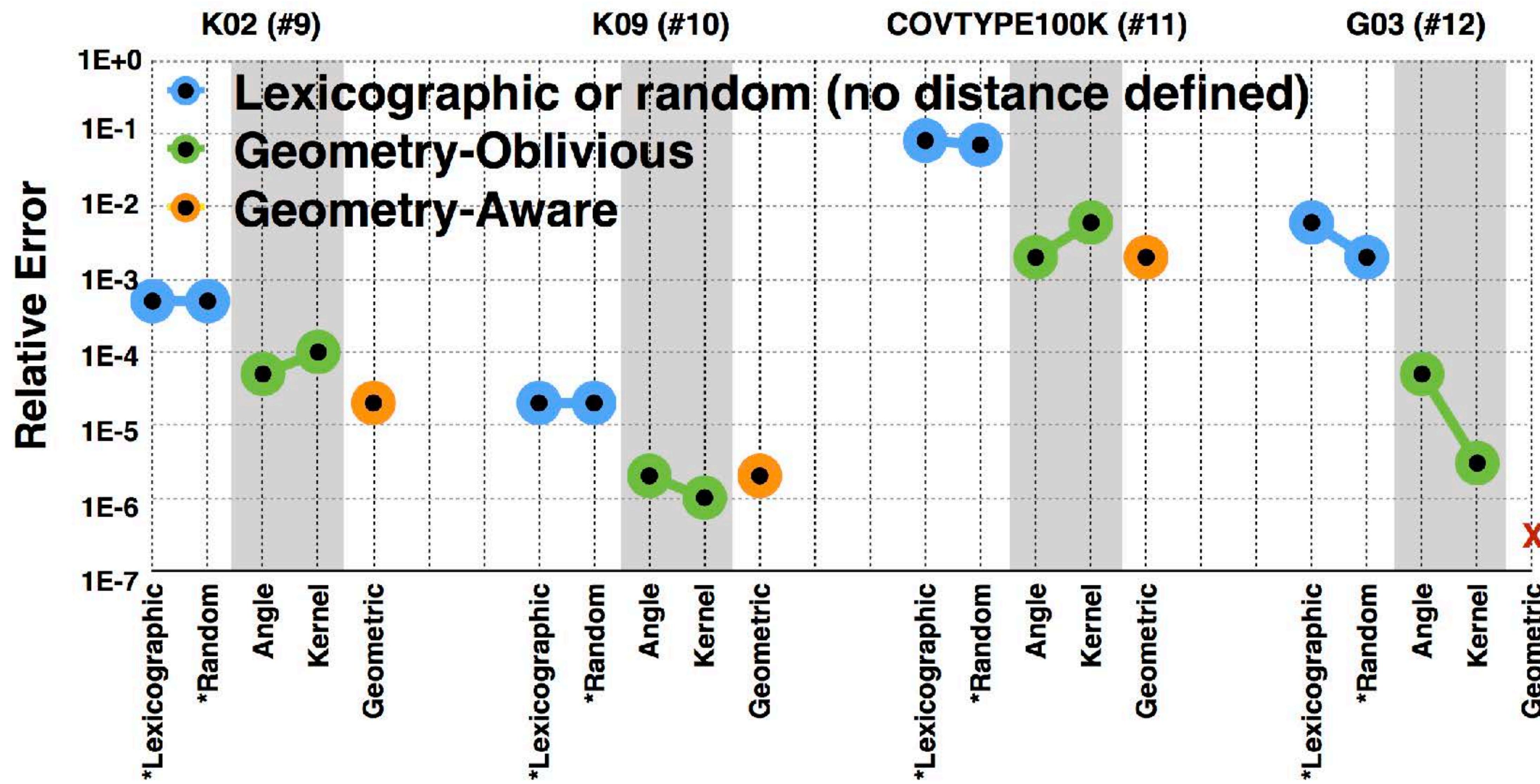
$$\|\phi_i - \phi_j\|_2^2$$

Angle

$$\sin\left(\frac{\phi_i \cdot \phi_j}{\|\phi_i\|_2 \|\phi_j\|_2}\right)$$

$$K_{ii} + K_{jj} - 2K_{ij}$$

$$1 - \frac{K_{ij}^2}{K_{ii} K_{jj}}$$



***Lexicographic and *Random does not use any distance (no sparse correction)**

K02 is a 2D regularized inverse Laplacian squared, resembling the Hessian operator of a PDE-constrained optimization problem.

**The Laplacian is discretized using a 5-stencil finite-difference scheme with Dirichlet boundary conditions on a regular grid.*

K04–K10 are kernel matrices in six dimensions (Gaussians, Laplacian, Green's function, polynomial and cosine-similarity).

G01–G05 are the inverse graph Laplacian of five graphs from UFL.

GOFMM vs Others

	HODLR				STRUMLPACK				GOFMM			
case	ϵ_2	Comp	Eval		ϵ_2	Comp	Eval		ϵ_2	Comp	Eval	
K02	6E-5	0.6	2.7		1E-4	9.2	0.6		2E-5	1.0	0.3	
K04	6E-5	0.7	2.7		1E-4	507.8	7.8		2E-5	1.0	0.5	
K07	7E-5	0.9	3.1		2E-4	528.4	8.2		4E-5	0.6	0.2	
K12	6E-5	0.7	2.7		2E-4	18.8	0.8		1E-4	0.6	0.2	
K17	1E-1	862.2	37.6		2E-1	663.4	8.2		9E-2	48.8	3.1	
G03	3E-4	12.9	9.7		3E-2	29.8	1.3		8E-5	0.5	0.8	

GOFMM vs Others

Parameters			ASKIT			GOFMM		
case	N	τ	ϵ_2	Comp	Eval	ϵ_2	Comp	Eval
K04	36 864	1E-3	2E-4	0.3	2E-2	2E-4	0.6	2E-2
K04	36 864	1E-6	8E-7	1.4	4E-2	7E-7	1.0	3E-2
K04	65 536	1E-3	2E-4	1.0	4E-2	2E-4	1.2	4E-2
K04	65 536	1E-6	7E-7	2.2	8E-2	6E-7	1.7	4E-2
K06	36 864	1E-3	4E-2	6.6	6E-2	3E-2	3.3	4E-2
K06	36 864	1E-6	2E-2	7.4	6E-2	3E-2	4.8	5E-2
K06	65 536	1E-3	4E-2	11.1	1E-1	4E-2	5.7	8E-2
K06	65 536	1E-6	5E-2	12.0	1E-1	4E-2	7.7	9E-2

Summary

- ASKIT for arbitrary dimension FMM using geometry
- Geometry Oblivious Fast Multipole Method for sketching dense SPD matrices; Gram distances