University of Arkansas, Fayetteville
ScholarWorks@UARK

Mathematical Sciences Spring Lecture Series | Mathematical Sciences

4-9-2021

Lecture 09: Hierarchically Low Rank and Kronecker Methods

Rio Yokota
Tokyo Institute of Technology, rioyokota@gsic.titech.ac.jp

Follow this and additional works at: https://scholarworks.uark.edu/mascsls

Part of the Algebra Commons, Control Theory Commons, Non-linear Dynamics Commons, Numerical Analysis and Computation Commons, and the Numerical Analysis and Scientific Computing Commons

Citation

Yokota, R. (2021). Lecture 09: Hierarchically Low Rank and Kronecker Methods. *Mathematical Sciences Spring Lecture Series.* Retrieved from https://scholarworks.uark.edu/mascsls/16

This Video is brought to you for free and open access by the Mathematical Sciences at ScholarWorks@UARK. It has been accepted for inclusion in Mathematical Sciences Spring Lecture Series by an authorized administrator of ScholarWorks@UARK. For more information, please contact scholar@uark.edu.

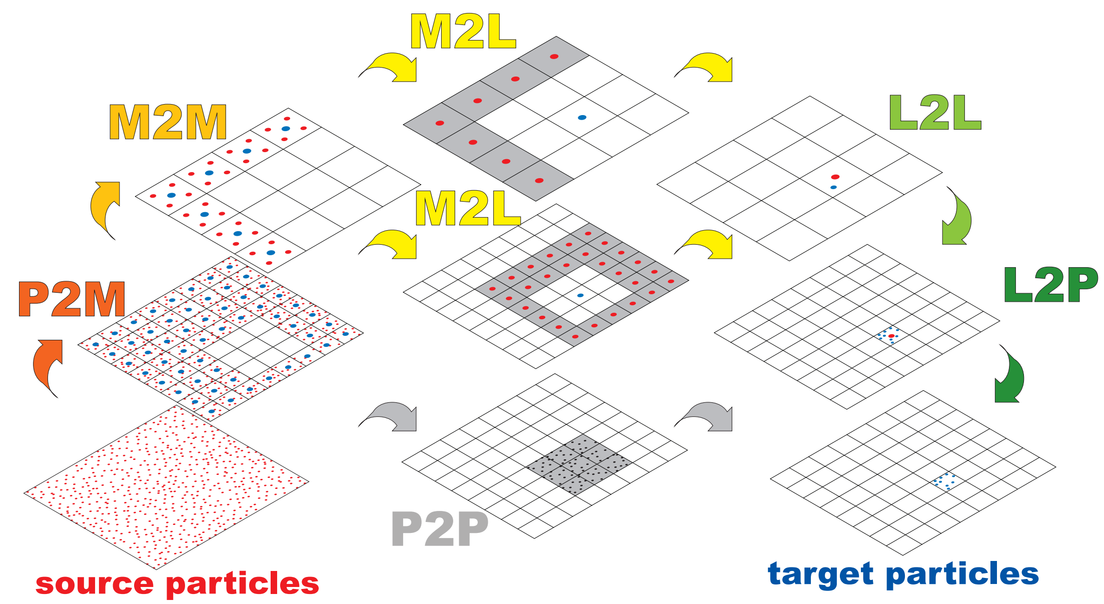
University of Arkansas Department of Mathematical Sciences
46th Spring Lecture Series

Hierarchically Low Rank and Kronecker Methods

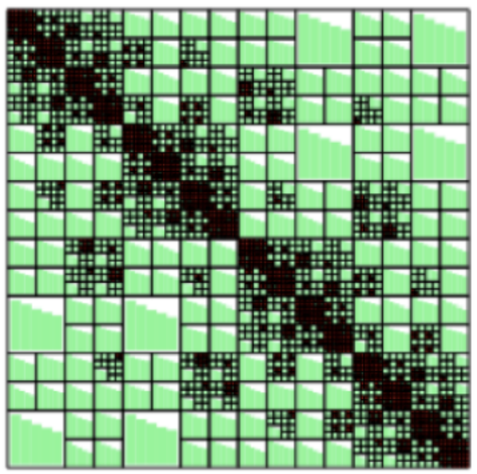
Tokyo Institute of Technology
Rio Yokota
rioyokota@gsic.titech.ac.jp

What I will be talking about today

1. Fast multipole methods



2. Hierarchical low-rank matrices

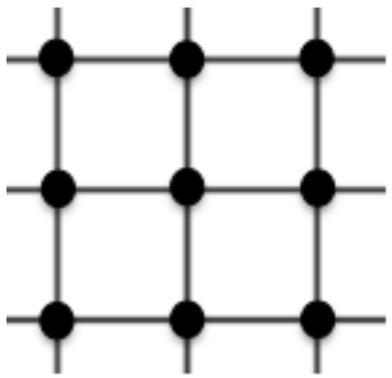


3. Kronecker factorization

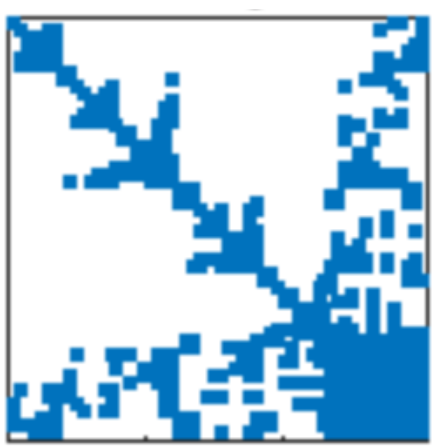
Fast Multipole Methods

Structure of matrices

Sparse

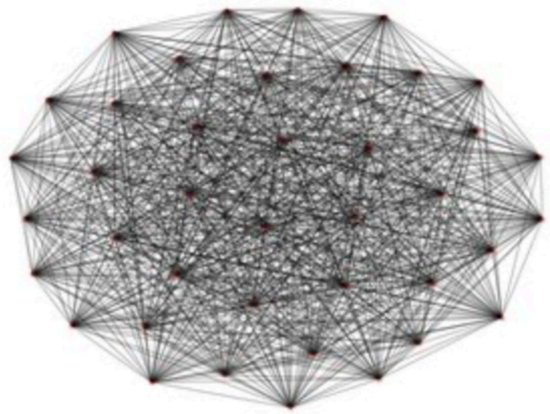


locally connected

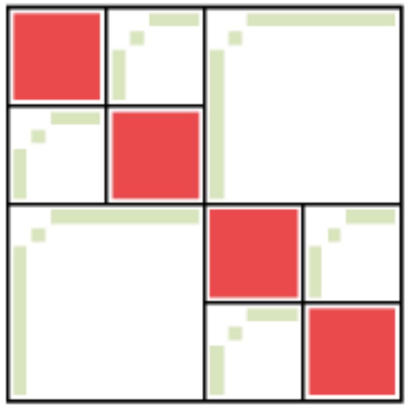


grouping based on connectivity

Dense



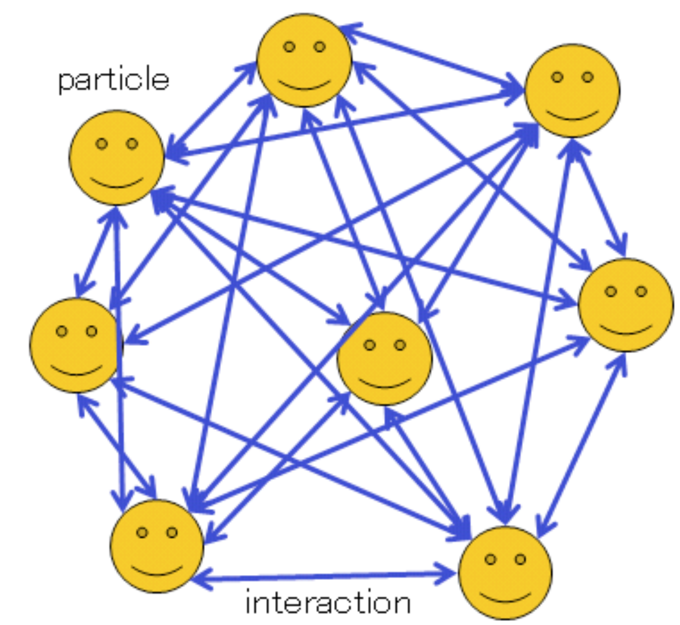
fully connected



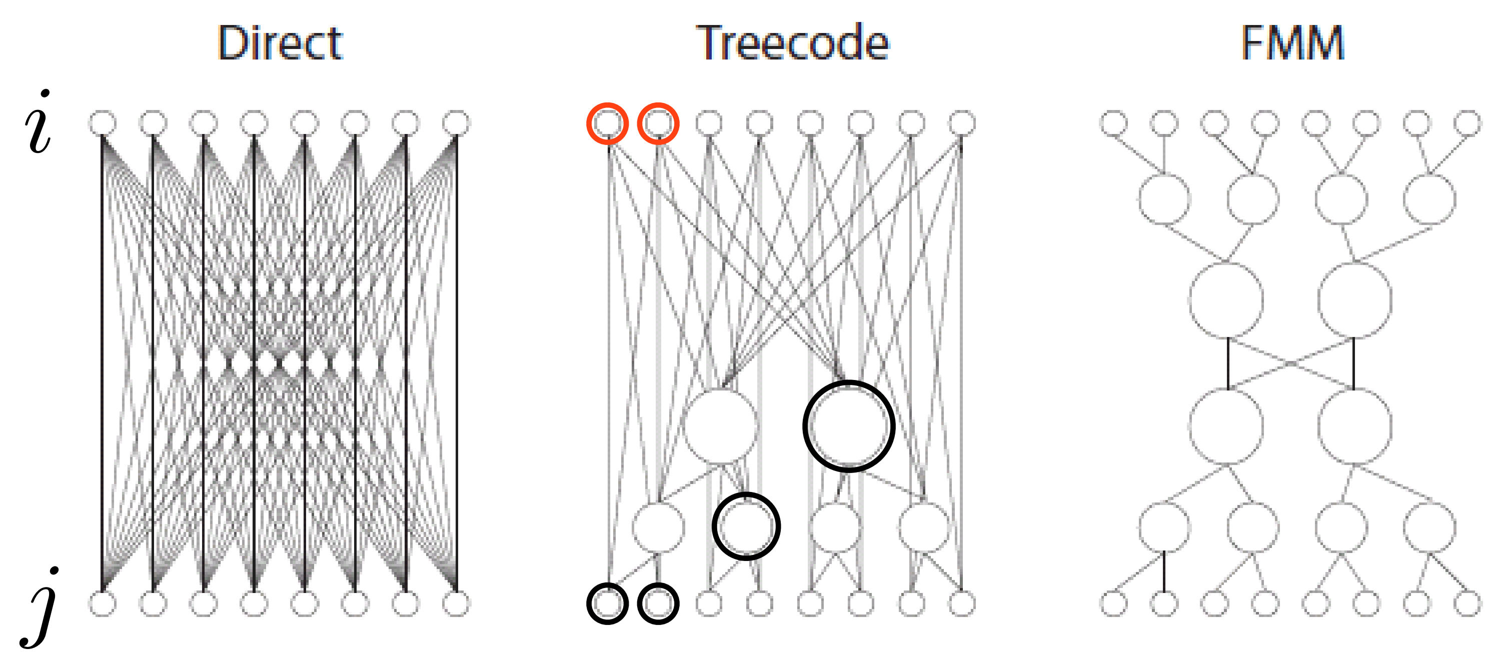
grouping based on proximity

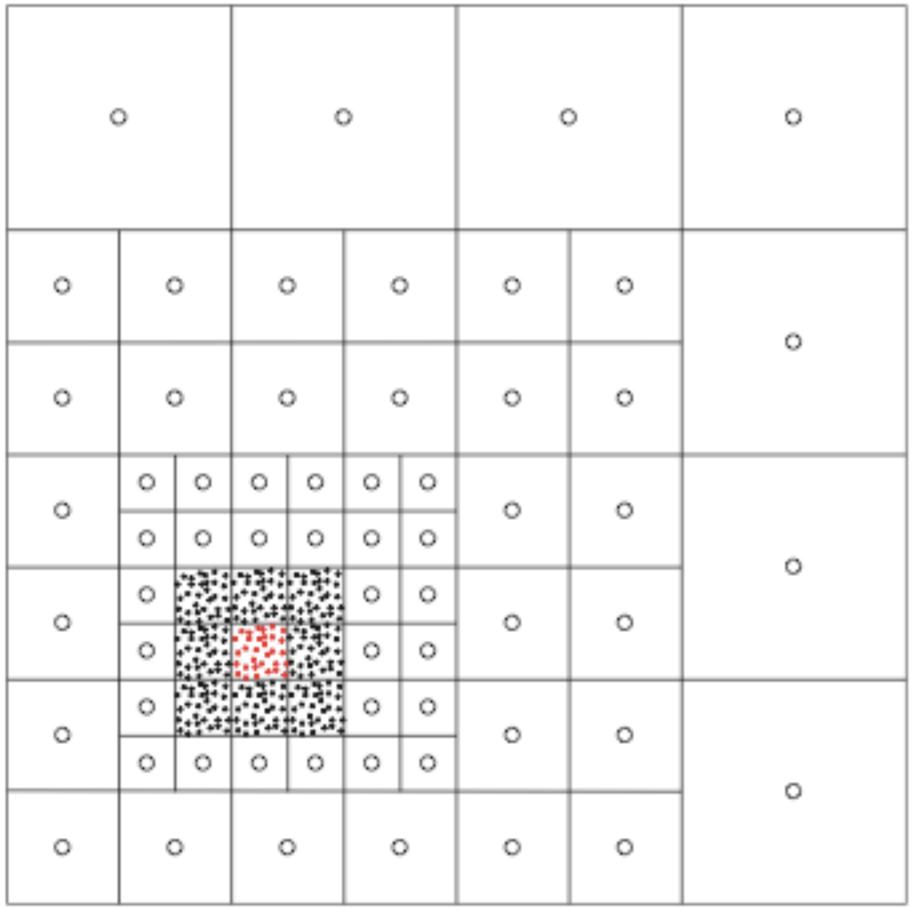
Hierarchical N-body methods

Particles interact with each other
Stars, Galaxies, Atoms, etc.

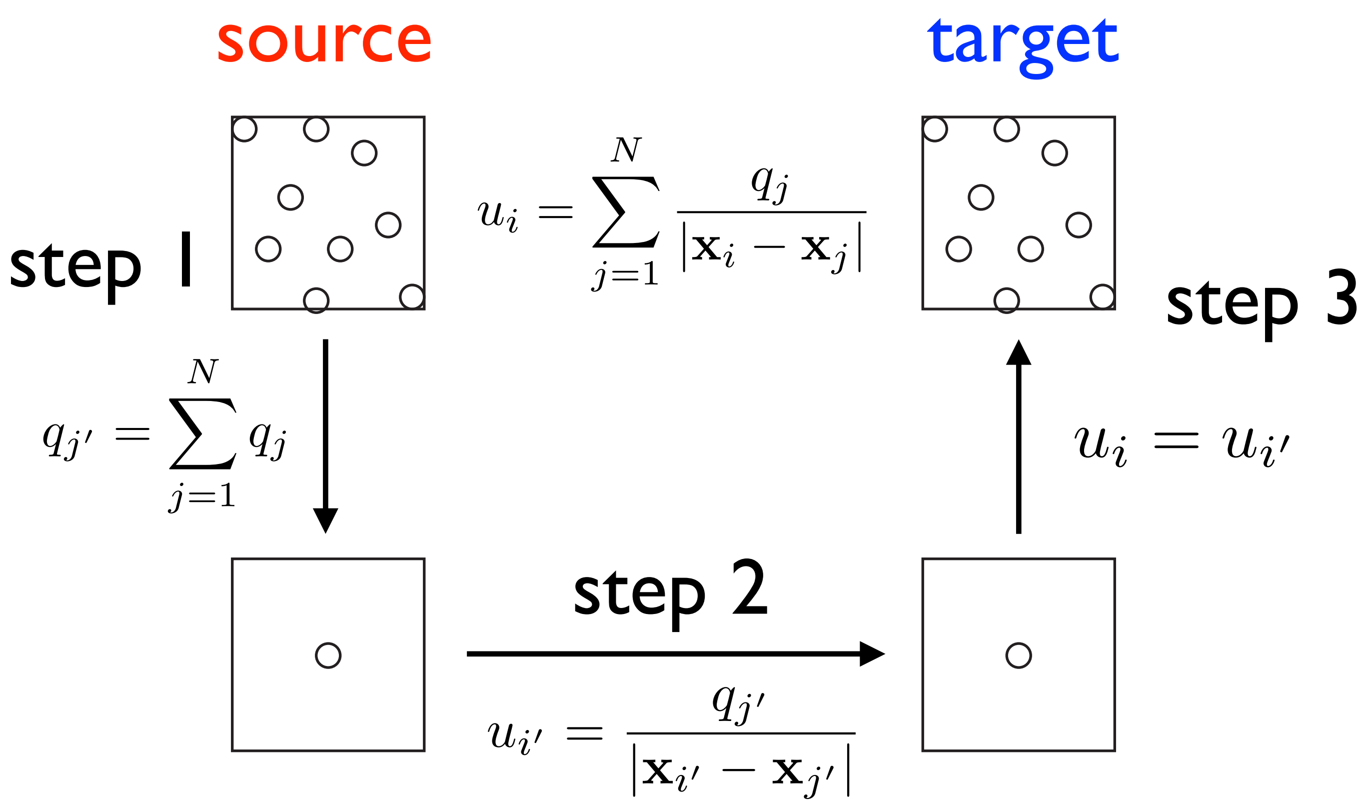


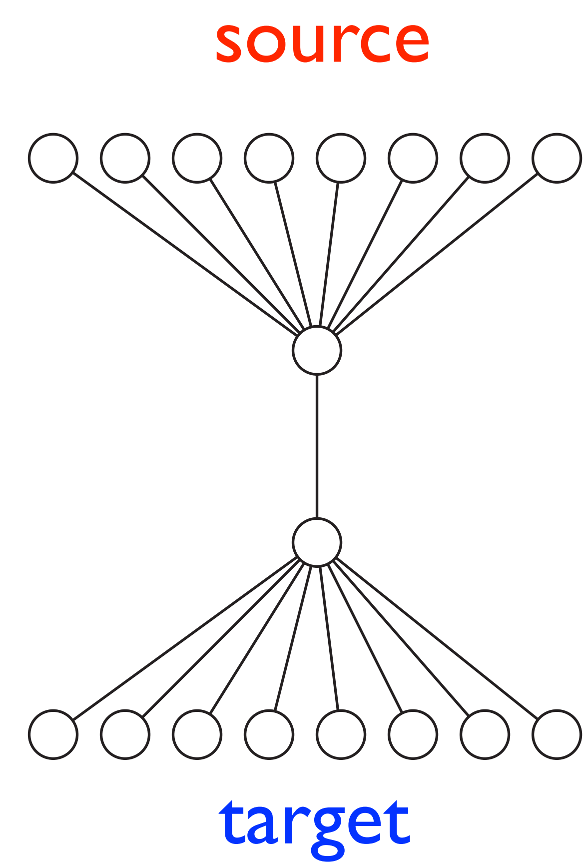
Computational cost
Direct sum - $O(N^2)$
Treecode - $O(N\log N)$
Fast Multipole Method - $O(N)$





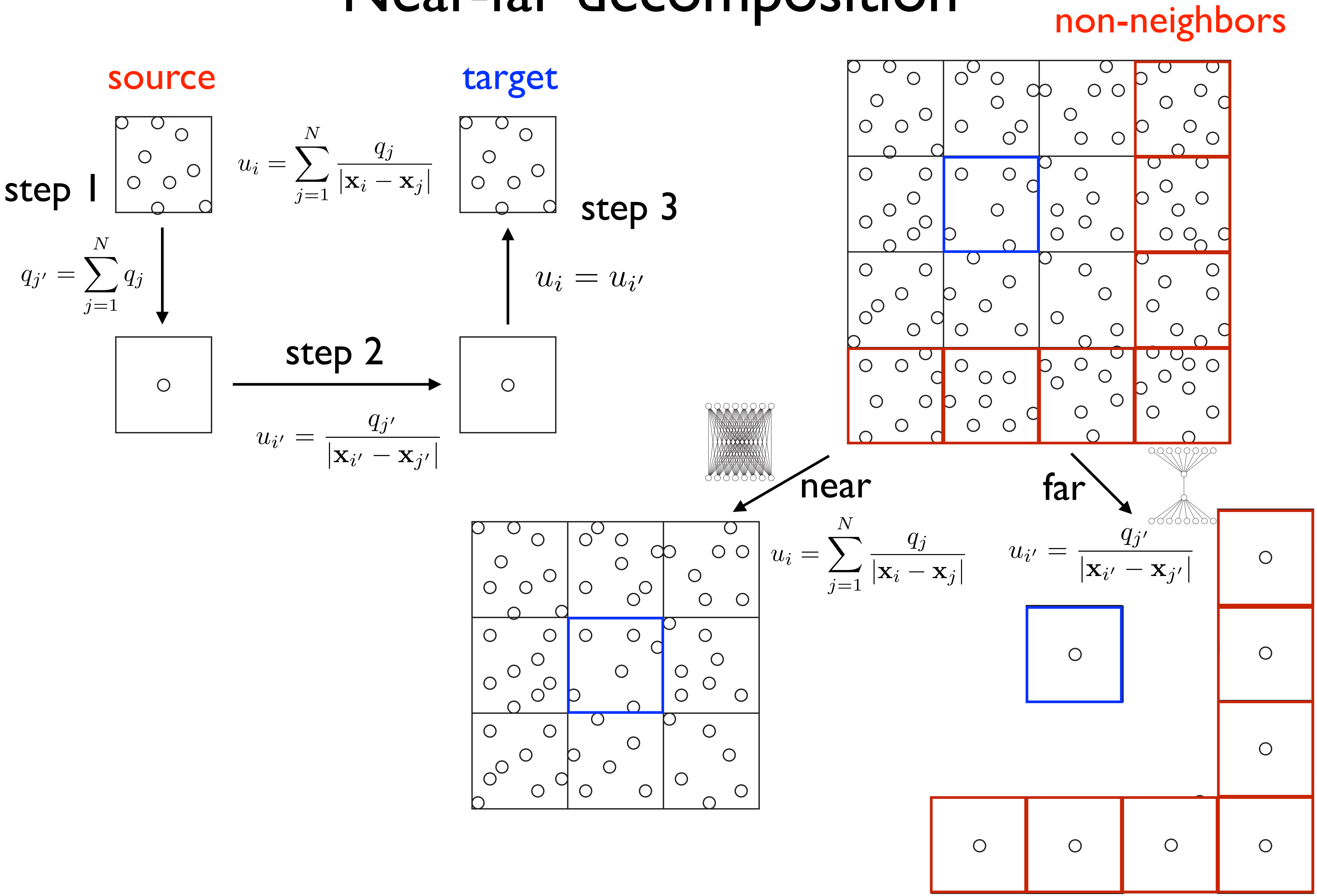
Approximating the interaction



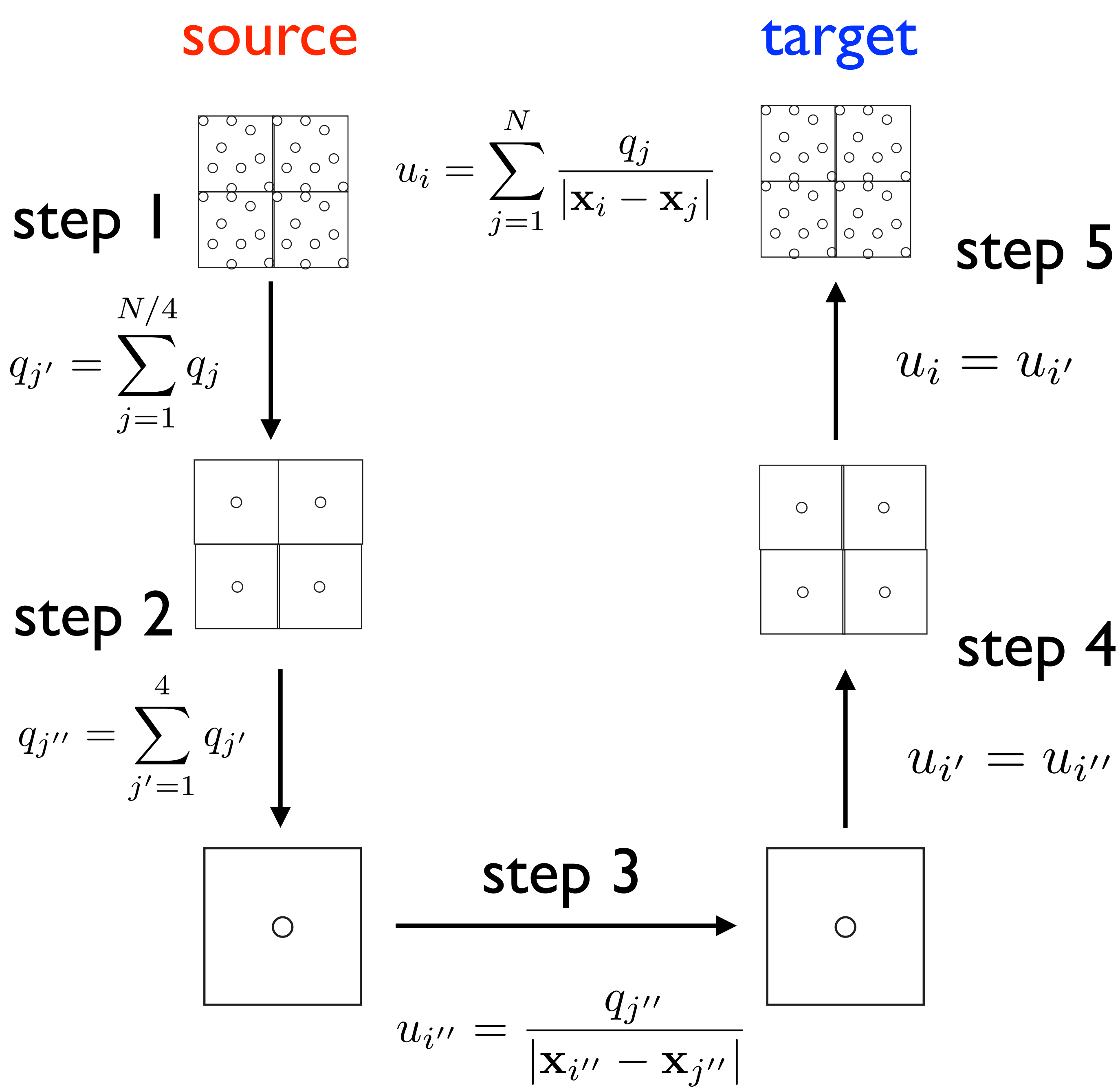


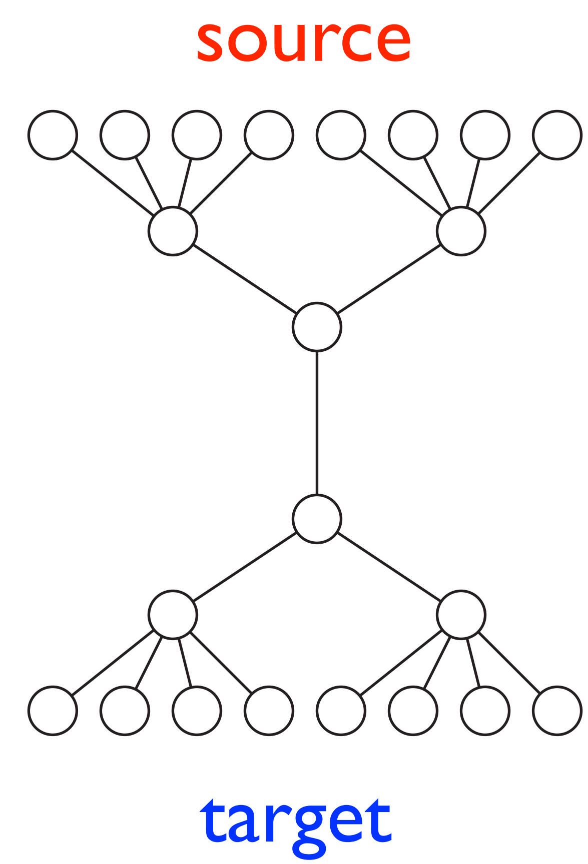
1. Sum all charges
2. Calculate effect of center source on center target
3. Assume all targets in the box have equal potential

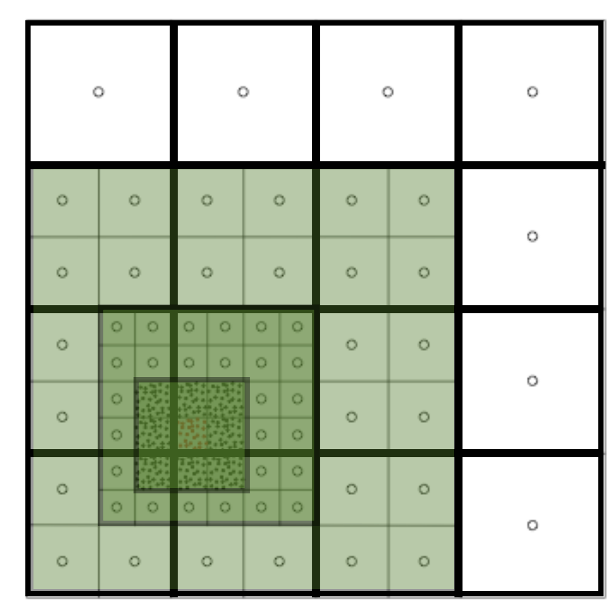
Near-far decomposition



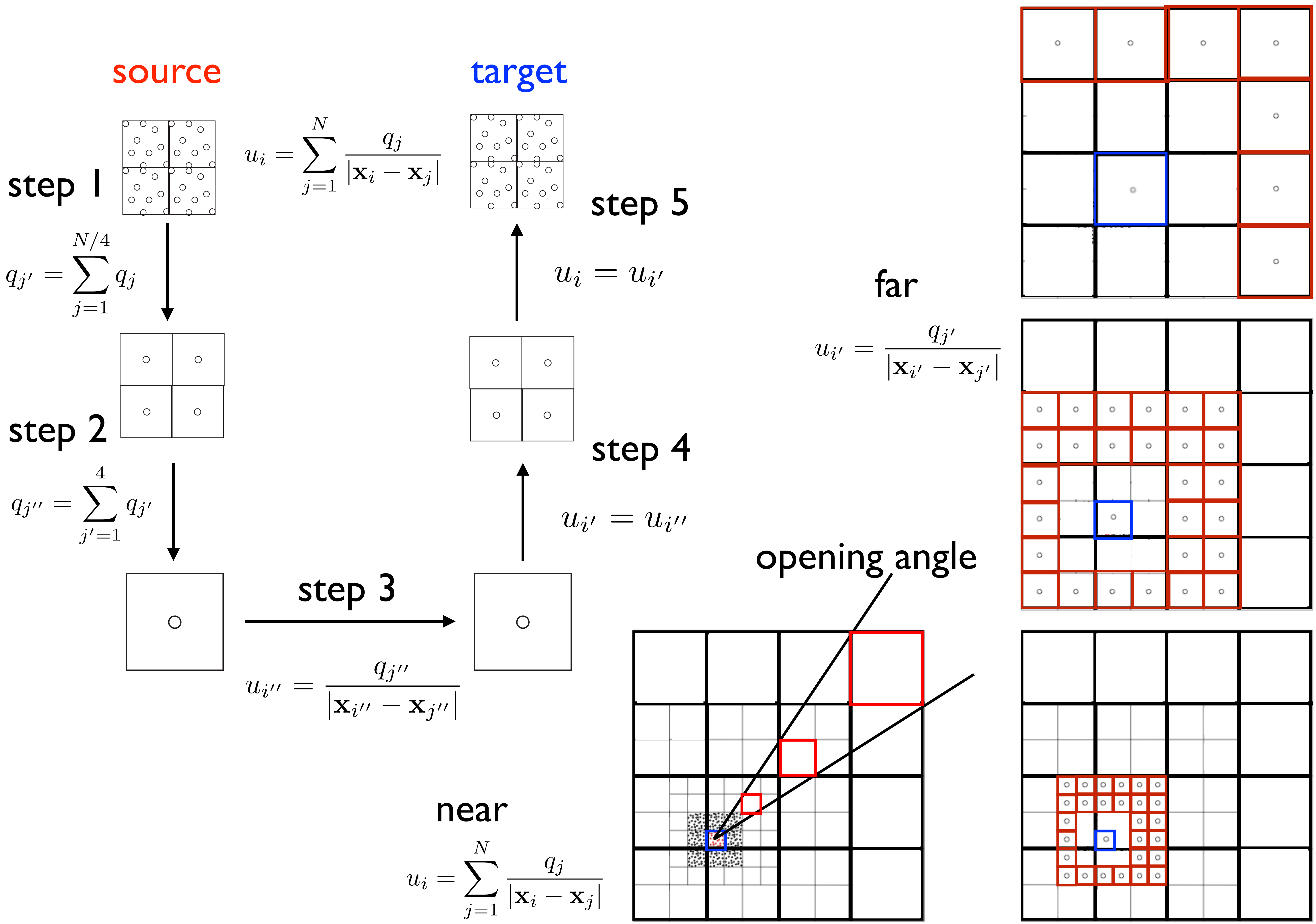
Hierarchical decomposition



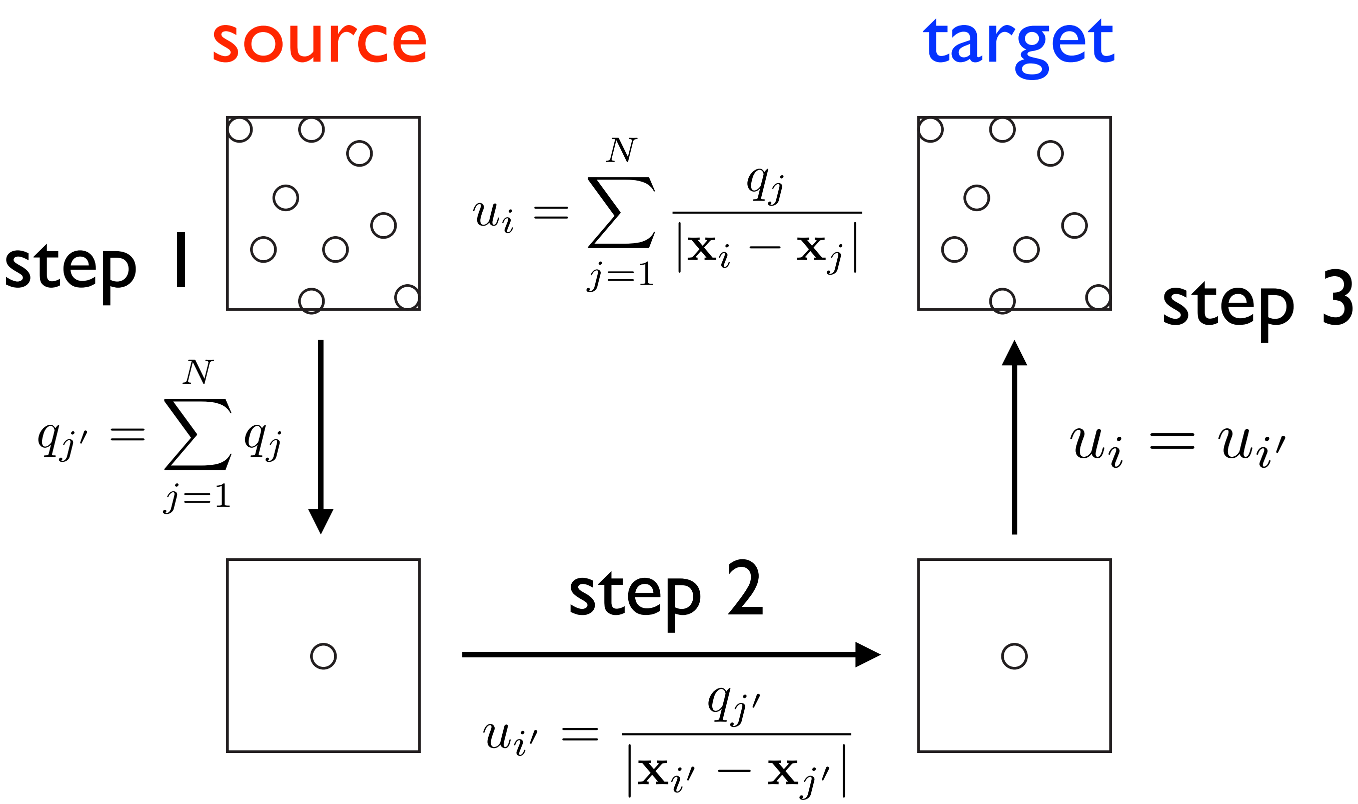


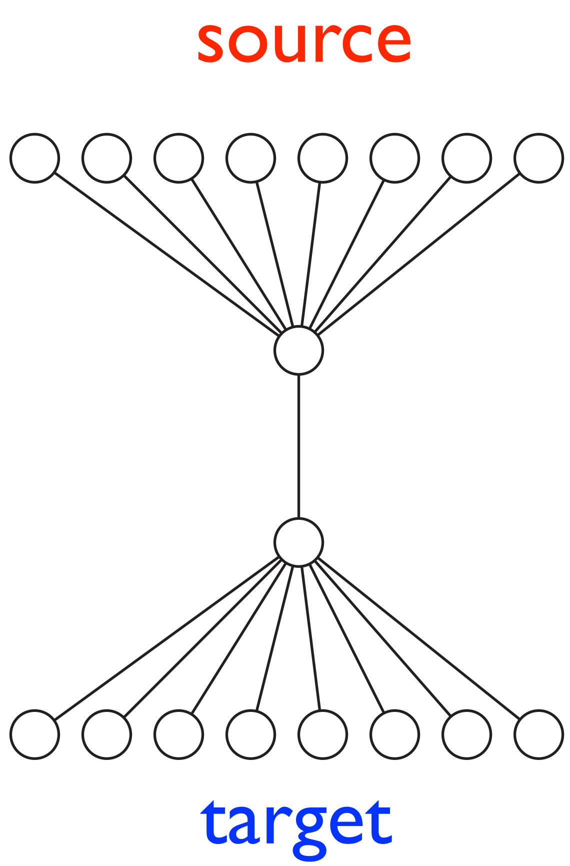


Hierarchical near-far decomposition



Approximating the interaction





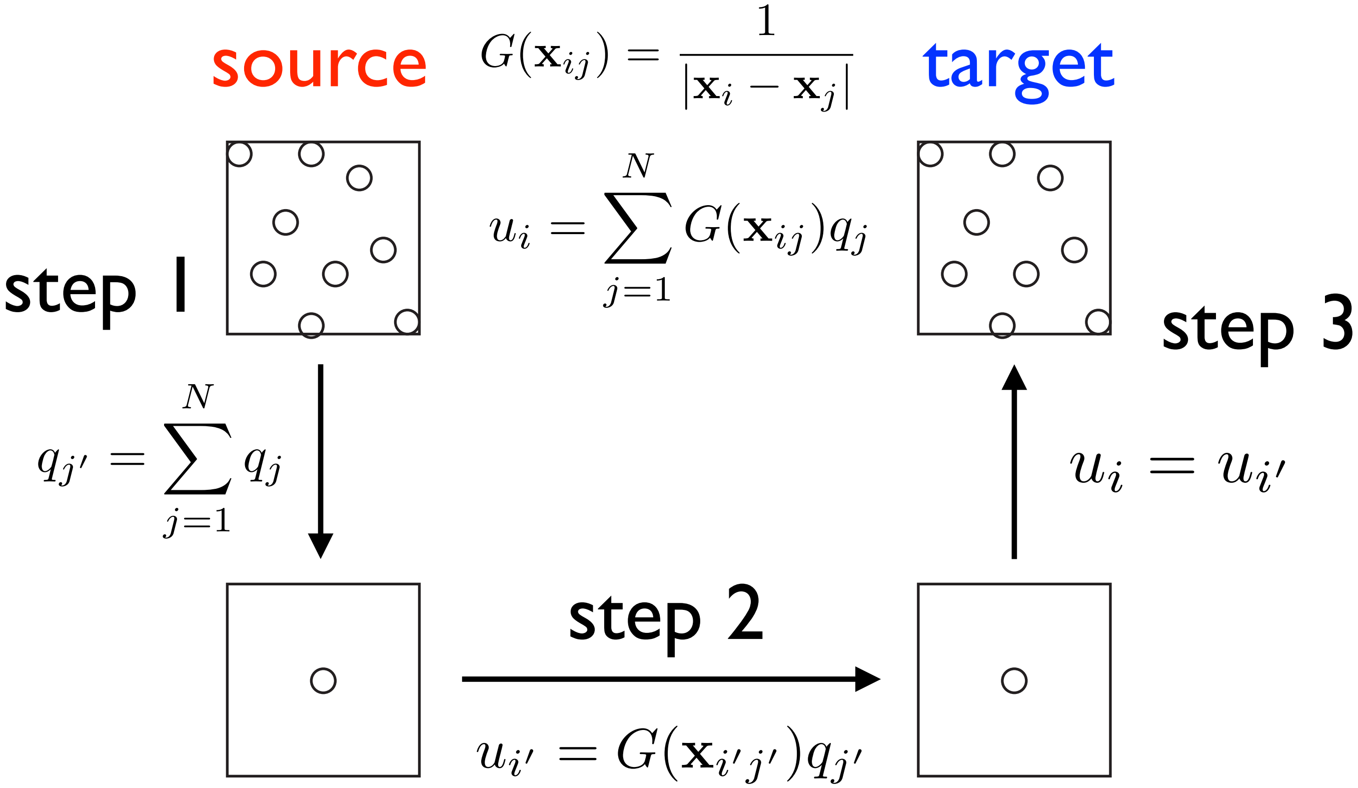
How accurate is the solution?

Higher order approximations

$$\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$$

$$G(\mathbf{x}_{ij}) = \frac{1}{|\mathbf{x}_i - \mathbf{x}_j|}$$

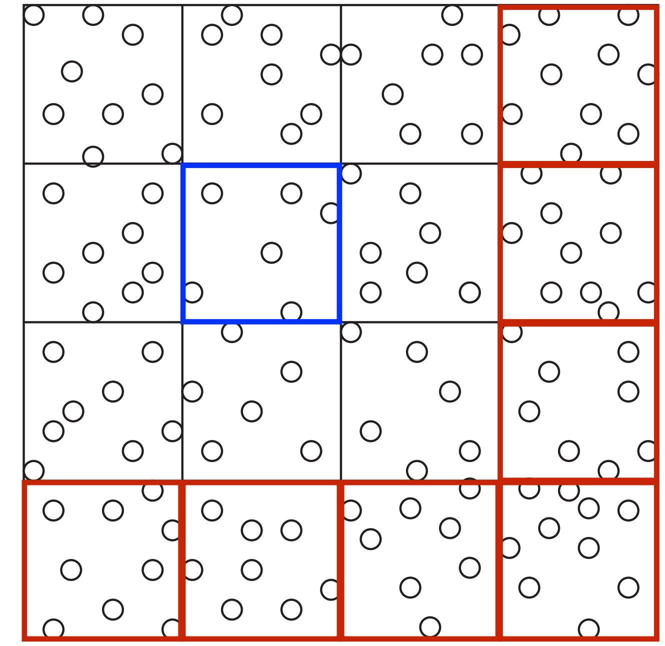
$$u_i = \sum_{j=1}^{N} G(\mathbf{x}_{ij}) q_j$$



$$q_{j'} = \sum_{j=1}^{N} q_j$$

$$u_{i'} = G(\mathbf{x}_{i'j'}) q_{j'}$$

$$u_i = u_{i'}$$



$$\mathbf{x}_{ij} = \mathbf{x}_{ii'} + \mathbf{x}_{i'j'} + \mathbf{x}_{j'j}$$

$$\mathbf{x}_{ii'} + \mathbf{x}_{j'j} \ll \mathbf{x}_{i'j'}$$

$$G(\mathbf{x}_{ij}) = \sum_{\mathbf{n}=0}^{\infty} \frac{1}{\mathbf{n}!} (\mathbf{x}_{ii'} + \mathbf{x}_{j'j})^{\mathbf{n}} \nabla^{(\mathbf{n})} G(\mathbf{x}_{i'j'})$$

Taylor expansion $f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n$

$$G(\mathbf{x}_{ij}) = \sum_{\mathbf{n}=0}^{p} \frac{1}{\mathbf{n}!} (\mathbf{x}_{ii'} + \mathbf{x}_{j'j})^{\mathbf{n}} \nabla^{(\mathbf{n})} G(\mathbf{x}_{i'j'})$$

Binomial theorem $(x+y)^n = \sum_{k=0}^{n} \frac{n!}{(n-k)!k!} x^{n-k} y^k$ →

$$= \sum_{\mathbf{n}=0}^{p} \frac{1}{\mathbf{n}!} \sum_{\mathbf{k}=0}^{\mathbf{n}} \frac{\mathbf{n}!}{(\mathbf{n}-\mathbf{k})!\mathbf{k}!} \mathbf{x}_{ii'}^{\mathbf{k}} \mathbf{x}_{j'j}^{\mathbf{n}-\mathbf{k}} \nabla^{(\mathbf{n})} G(\mathbf{x}_{i'j'})$$

Cancel $\mathbf{n}!$ →

$$= \sum_{\mathbf{n}=0}^{p} \sum_{\mathbf{k}=0}^{\mathbf{n}} \frac{1}{(\mathbf{n}-\mathbf{k})!\mathbf{k}!} \mathbf{x}_{ii'}^{\mathbf{k}} \mathbf{x}_{j'j}^{\mathbf{n}-\mathbf{k}} \nabla^{(\mathbf{n})} G(\mathbf{x}_{i'j'})$$

Swap loop order between $\mathbf{n}$ and $\mathbf{k}$ →

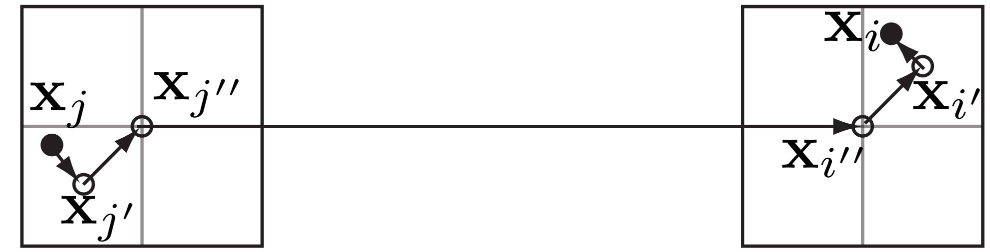
$$= \sum_{\mathbf{k}=0}^{p} \sum_{\mathbf{n}=\mathbf{k}}^{p} \frac{1}{(\mathbf{n}-\mathbf{k})!\mathbf{k}!} \mathbf{x}_{ii'}^{\mathbf{k}} \mathbf{x}_{j'j}^{\mathbf{n}-\mathbf{k}} \nabla^{(\mathbf{n})} G(\mathbf{x}_{i'j'})$$

Redefine $\mathbf{n}-\mathbf{k}$ to $\mathbf{n}$ →

$$= \sum_{\mathbf{k}=0}^{p} \sum_{\mathbf{n}=0}^{p-\mathbf{k}} \frac{1}{\mathbf{n}!\mathbf{k}!} \mathbf{x}_{ii'}^{\mathbf{k}} \mathbf{x}_{j'j}^{\mathbf{n}} \nabla^{(\mathbf{n}+\mathbf{k})} G(\mathbf{x}_{i'j'})$$

$$= \sum_{\mathbf{k}=0}^{p} \frac{1}{\mathbf{k}!} \mathbf{x}_{ii'}^{\mathbf{k}} \underbrace{\sum_{\mathbf{n}=0}^{p-\mathbf{k}} \nabla^{(\mathbf{n}+\mathbf{k})} G(\mathbf{x}_{i'j'}) \underbrace{\frac{1}{\mathbf{n}!} \mathbf{x}_{j'j}^{\mathbf{n}}}_{\mathbf{M}}}_{\mathbf{L}}$$

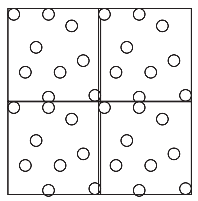
Multi-level case



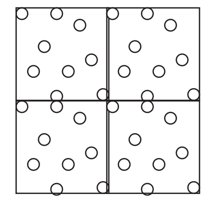
source

target

P2M



$$u_i = \sum_{j=1}^{N} G(\mathbf{x}_{ij}) q_j$$



L2P

$$G(\mathbf{x}_{ij}) = \sum_{\mathbf{k}=0}^{p} \frac{1}{\mathbf{k}!}\mathbf{x}_{ii''}^{\mathbf{k}} \underbrace{\sum_{\mathbf{n}=0}^{p-\mathbf{k}} \nabla^{(\mathbf{n}+\mathbf{k})} G(\mathbf{x}_{i''j''}) \underbrace{\frac{1}{\mathbf{n}!}\mathbf{x}_{j''j}^{\mathbf{n}}}_{\mathbf{M}}}_{\mathbf{L}}$$

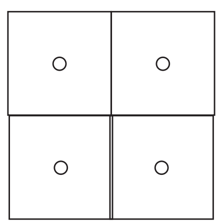
$$\mathbf{M}^{\mathbf{n}}(\mathbf{x}_{j'}) = \sum_{j=1}^{N/4} \frac{1}{\mathbf{n}!}\mathbf{x}_{j'j}^{\mathbf{n}} q_j$$

$$u_i = \sum_{\mathbf{k}=0}^{p} \frac{1}{\mathbf{k}!}\mathbf{x}_{ii'}^{\mathbf{k}} \mathbf{L}^{\mathbf{k}}(\mathbf{x}_{i'})$$

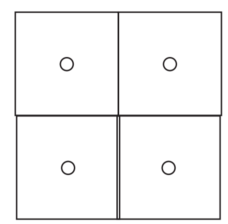
$$\mathbf{x}_{j''j} = \mathbf{x}_{j''j'} + \mathbf{x}_{j'j}$$

$$\mathbf{M}^{\mathbf{n}}(\mathbf{x}_{j''}) = \frac{1}{\mathbf{n}!}(\mathbf{x}_{j''j'} + \mathbf{x}_{j'j})^{\mathbf{n}} q_j$$

$$= \frac{1}{\mathbf{n}!}\sum_{\mathbf{k}=0}^{\mathbf{n}} \frac{\mathbf{n}!}{(\mathbf{n}-\mathbf{k})!\mathbf{k}!}\mathbf{x}_{j''j'}^{\mathbf{n}-\mathbf{k}}\mathbf{x}_{j'j}^{\mathbf{k}} q_j$$



M2M



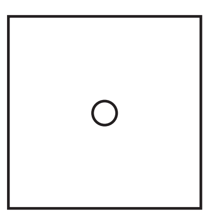
L2L

$$= \sum_{\mathbf{k}=0}^{\mathbf{n}} \frac{1}{(\mathbf{n}-\mathbf{k})!}\mathbf{x}_{j''j'}^{\mathbf{n}-\mathbf{k}}\mathbf{M}^{\mathbf{k}}(\mathbf{x}_{j'})$$

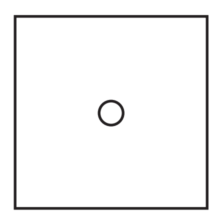
$$\mathbf{M}^{\mathbf{n}}(\mathbf{x}_{j''}) = \sum_{j'=1}^{4}\sum_{\mathbf{k}=0}^{\mathbf{n}} \frac{1}{(\mathbf{n}-\mathbf{k})!}\mathbf{x}_{j''j'}^{\mathbf{n}-\mathbf{k}}\mathbf{M}^{\mathbf{k}}(\mathbf{x}_{j'})$$

$$\mathbf{L}^{\mathbf{n}}(\mathbf{x}_{i'}) = \sum_{\mathbf{k}=\mathbf{n}}^{p} \frac{1}{(\mathbf{k}-\mathbf{n})!}\mathbf{x}_{i'i''}^{\mathbf{k}-\mathbf{n}}\mathbf{L}^{\mathbf{k}}(\mathbf{x}_{i''})$$

$$\mathbf{x}_{ii''} = \mathbf{x}_{ii'} + \mathbf{x}_{i'i''}$$



M2L



$$u_i = \sum_{\mathbf{k}=0}^{p} \frac{1}{\mathbf{k}!}(\mathbf{x}_{ii'} + \mathbf{x}_{i'i''})^{\mathbf{k}}\mathbf{L}^{\mathbf{k}}(\mathbf{x}_{i''})$$

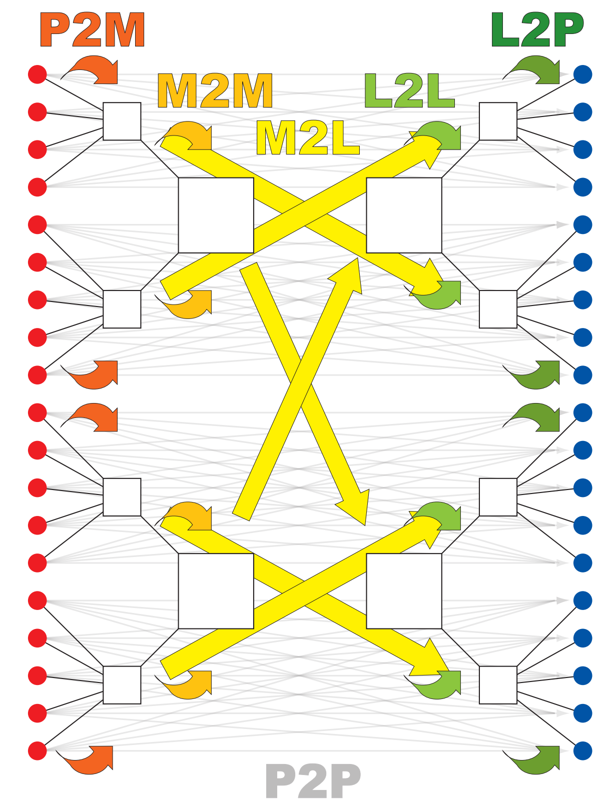
$$\mathbf{L}^{\mathbf{k}}(\mathbf{x}_{i''}) = \sum_{\mathbf{n}=0}^{p-\mathbf{k}} \nabla^{(\mathbf{n}+\mathbf{k})} G(\mathbf{x}_{i''j''})\mathbf{M}^{\mathbf{n}}(\mathbf{x}_{j''})$$

$$= \sum_{\mathbf{k}=0}^{p}\sum_{\mathbf{n}=0}^{\mathbf{k}} \frac{1}{(\mathbf{k}-\mathbf{n})!\mathbf{n}!}\mathbf{x}_{ii'}^{\mathbf{n}}\mathbf{x}_{i'i''}^{\mathbf{k}-\mathbf{n}}\mathbf{L}^{\mathbf{k}}(\mathbf{x}_{i''})$$

$$= \sum_{\mathbf{n}=0}^{p}\sum_{\mathbf{k}=\mathbf{n}}^{p} \frac{1}{(\mathbf{k}-\mathbf{n})!\mathbf{n}!}\mathbf{x}_{ii'}^{\mathbf{n}}\mathbf{x}_{i'i''}^{\mathbf{k}-\mathbf{n}}\mathbf{L}^{\mathbf{k}}(\mathbf{x}_{i''})$$

$$\mathbf{L}^{\mathbf{n}}(\mathbf{x}_{i'}) = \sum_{\mathbf{k}=\mathbf{n}}^{p} \frac{1}{(\mathbf{k}-\mathbf{n})!}\mathbf{x}_{i'i''}^{\mathbf{k}-\mathbf{n}}\mathbf{L}^{\mathbf{k}}(\mathbf{x}_{i''}) \longrightarrow = \sum_{\mathbf{n}=0}^{p} \frac{1}{\mathbf{n}!}\mathbf{x}_{ii'}^{\mathbf{n}}\mathbf{L}^{\mathbf{n}}(\mathbf{x}_{i'})$$

Flow of Calculation



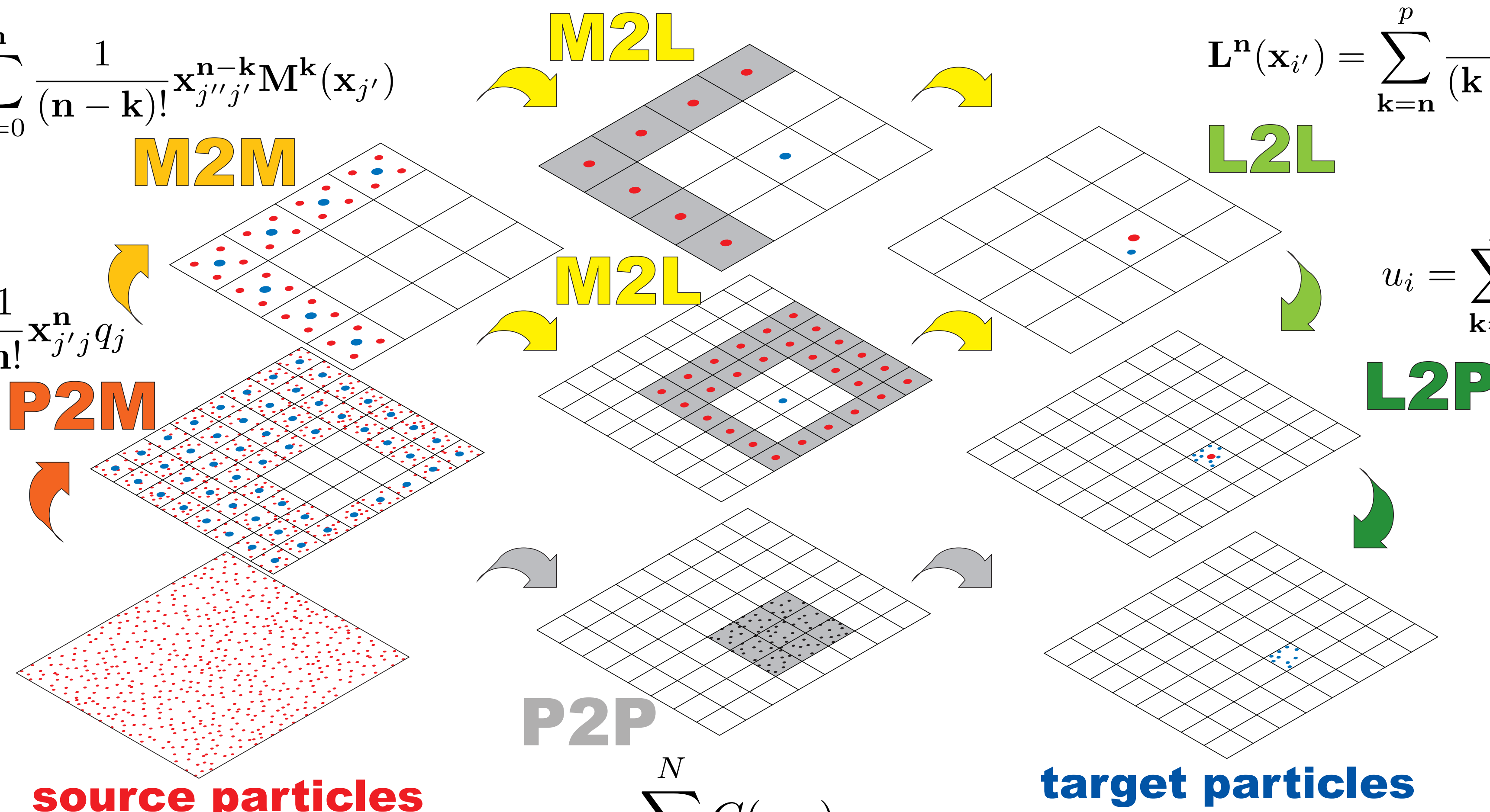
$$\mathbf{L}^{\mathbf{k}}(\mathbf{x}_{i''}) = \sum_{\mathbf{n}=0}^{p-\mathbf{k}} \nabla^{(\mathbf{n}+\mathbf{k})} G(\mathbf{x}_{i''j''}) \mathbf{M}^{\mathbf{n}}(\mathbf{x}_{j''})$$

$$\mathbf{M}^{\mathbf{n}}(\mathbf{x}_{j''}) = \sum_{j'=1}^{4} \sum_{\mathbf{k}=0}^{\mathbf{n}} \frac{1}{(\mathbf{n}-\mathbf{k})!} \mathbf{x}_{j''j'}^{\mathbf{n}-\mathbf{k}} \mathbf{M}^{\mathbf{k}}(\mathbf{x}_{j'})$$

$$\mathbf{L}^{\mathbf{n}}(\mathbf{x}_{i'}) = \sum_{\mathbf{k}=\mathbf{n}}^{p} \frac{1}{(\mathbf{k}-\mathbf{n})!} \mathbf{x}_{i'i''}^{\mathbf{k}-\mathbf{n}} \mathbf{L}^{\mathbf{k}}(\mathbf{x}_{i''})$$

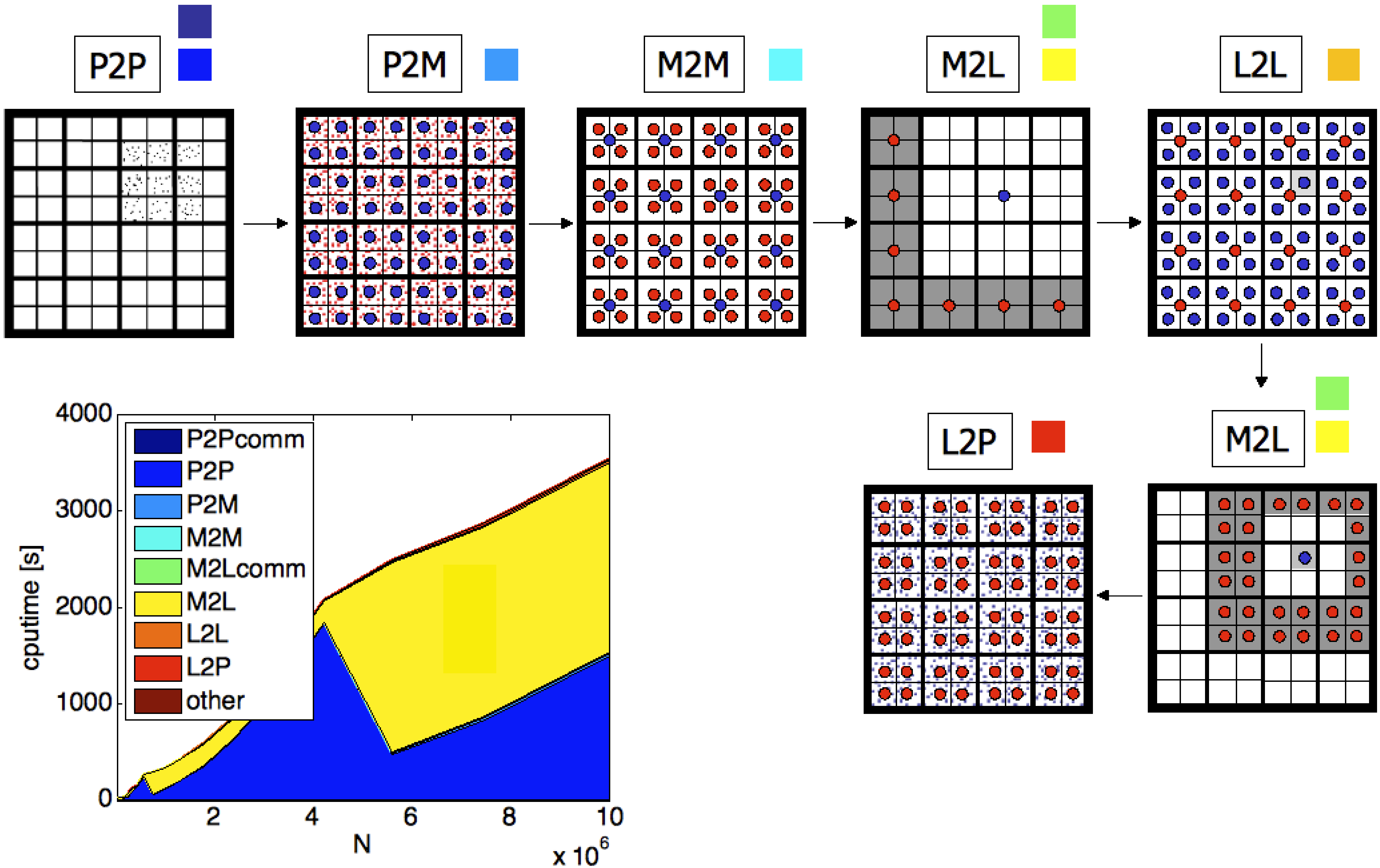
$$\mathbf{M}^{\mathbf{n}}(\mathbf{x}_{j'}) = \sum_{j=1}^{N/4} \frac{1}{\mathbf{n}!} \mathbf{x}_{j'j}^{\mathbf{n}} q_j$$

$$u_i = \sum_{\mathbf{k}=0}^{p} \frac{1}{\mathbf{k}!} \mathbf{x}_{ii'}^{\mathbf{k}} \mathbf{L}^{\mathbf{k}}(\mathbf{x}_{i'})$$



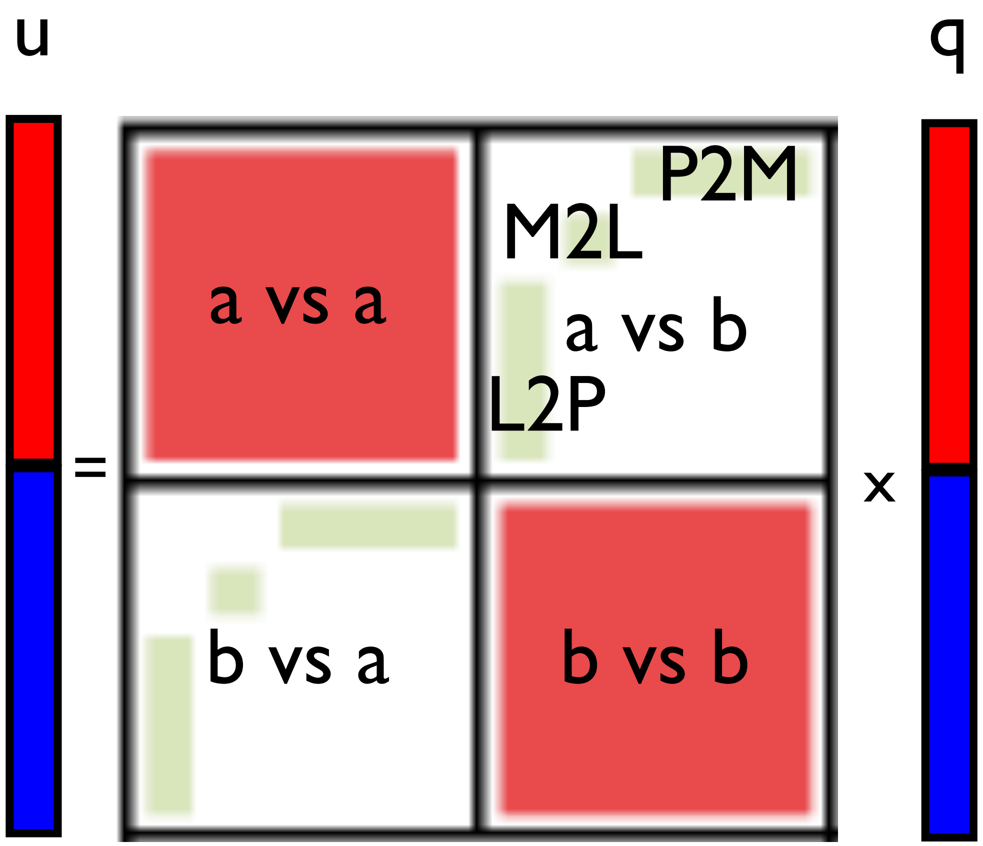
$$u_i = \sum_{j=1}^{N} G(\mathbf{x}_{ij}) q_j$$

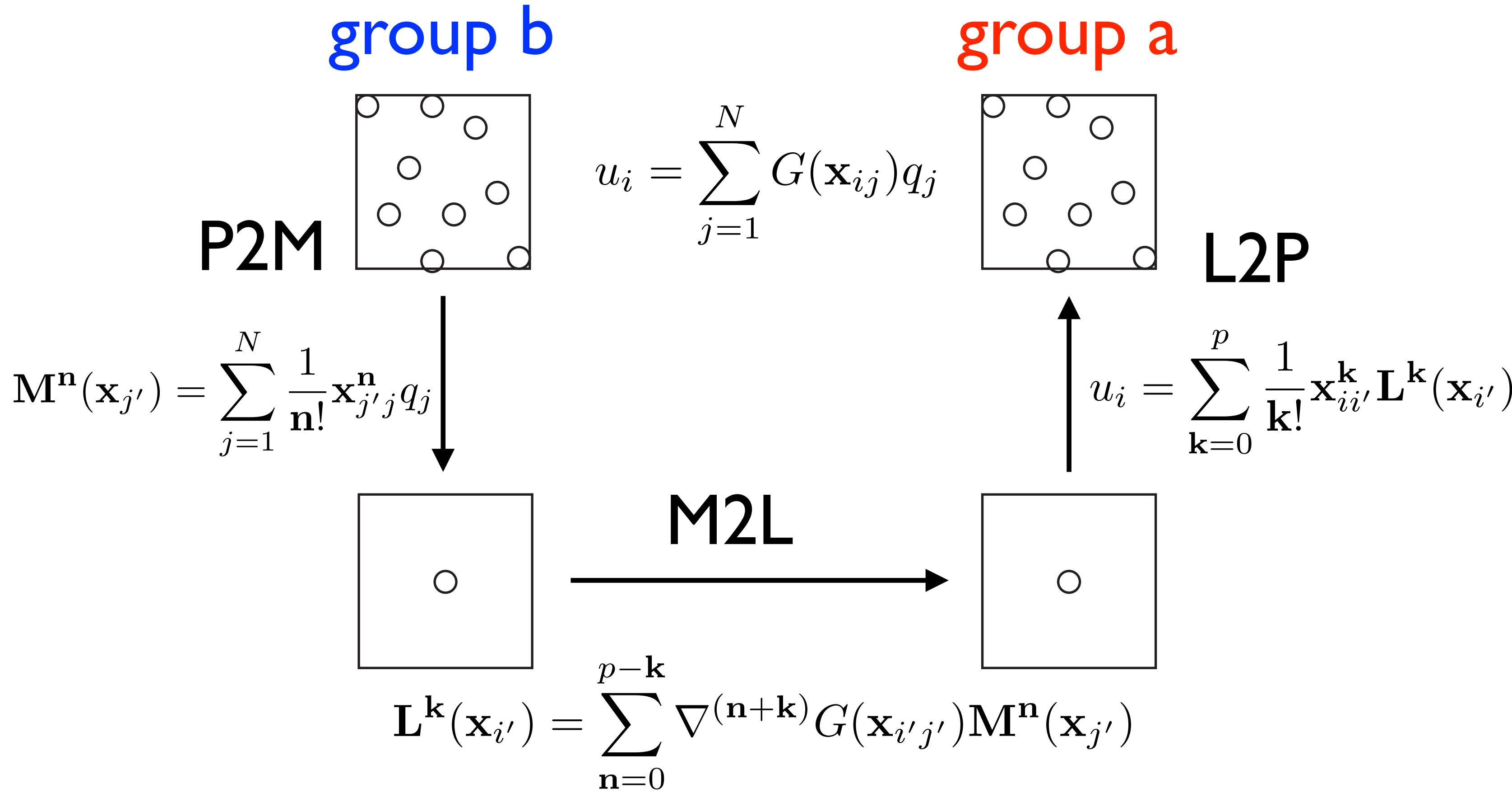
How much time does each part take?



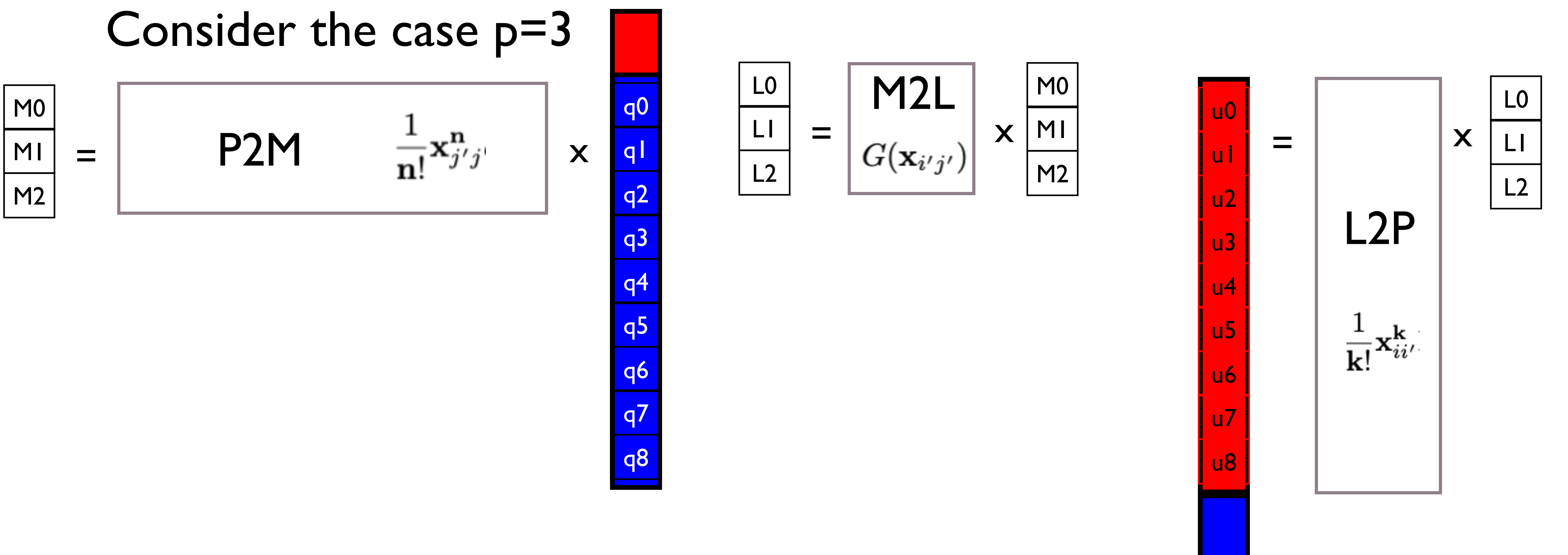
Hierarchical low-rank matrices

FMM as a hierarchical matrix-vector multiplication





Consider the case p=3



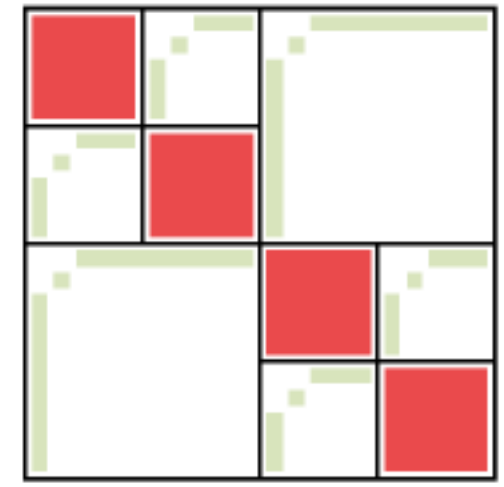
Hierarchical low-rank matrices

Replace dense linear algebra

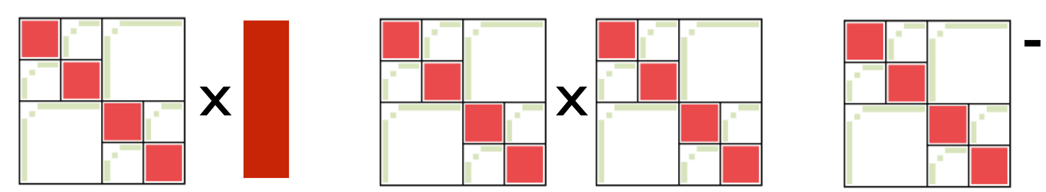
Compute : $\mathcal{O}(N^3) \longrightarrow \mathcal{O}(N)$

Memory : $\mathcal{O}(N^2) \longrightarrow \mathcal{O}(N)$

Hierarchical off-diagonal blocks
Approximated with low rank



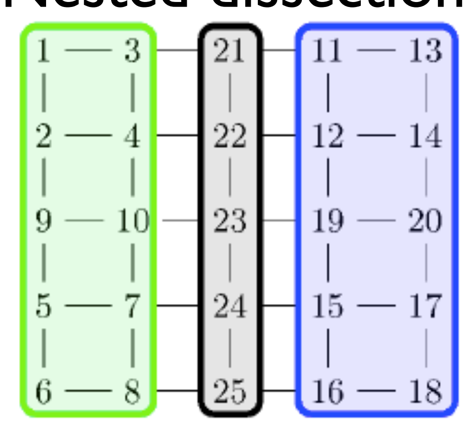
Augment sparse linear algebra

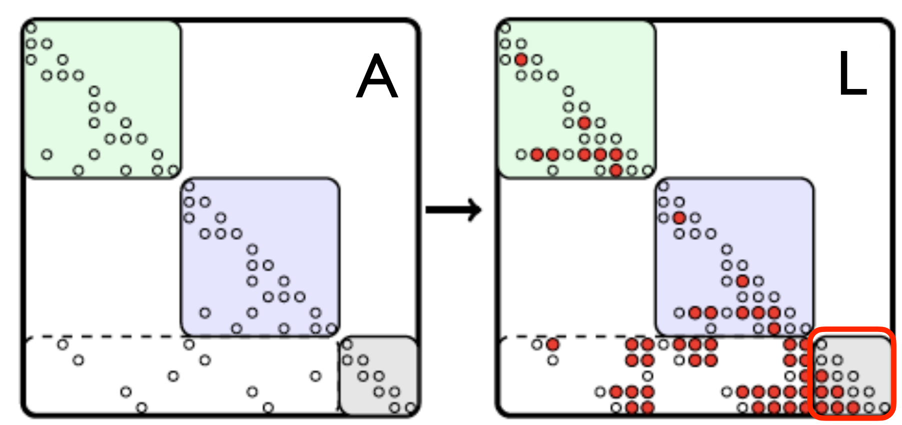


Sparse direct solvers

Schur complement (frontal matrix) is dense but numerically low-rank

Nested dissection





Schur complement

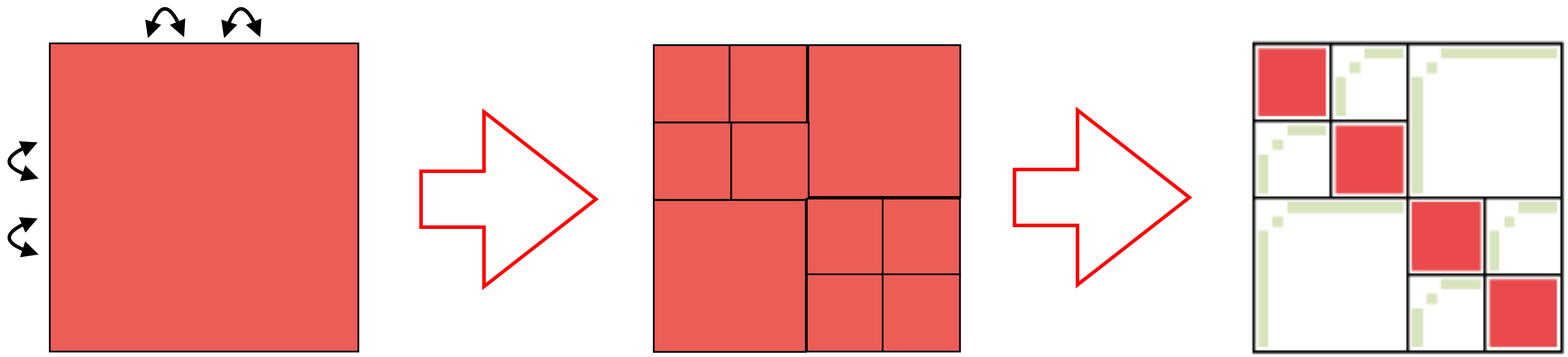
Iterative solvers

Use small rank to precondition

Less sensitive to matrix condition than multigrid

Three Stages of H-matrix Compression

Reordering | Subdivision | Low-Rank



<u>What to minimize？</u>

Fill-in (Graph connections)
Rank (Geometric distance)
Communication (Locality)
→Close nodes are usually connected, so minimizing rank will minimize fill-in

<u>How much to divide？</u>

Subdividing the block will decrease the rank

The rank can be kept constant while using the subdivision to control the accuracy
→SIMD friendly

<u>Speed or reliability？</u>

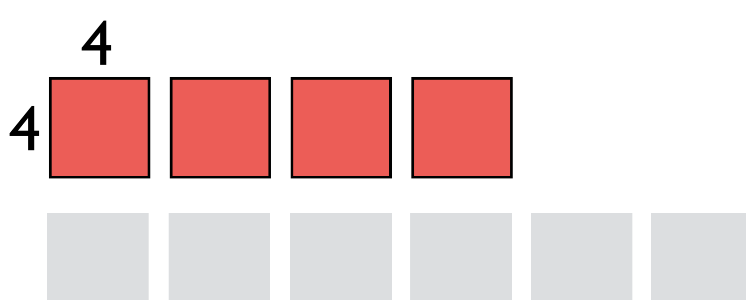
ACA is fast but unreliable
RSVD is reliable but slow

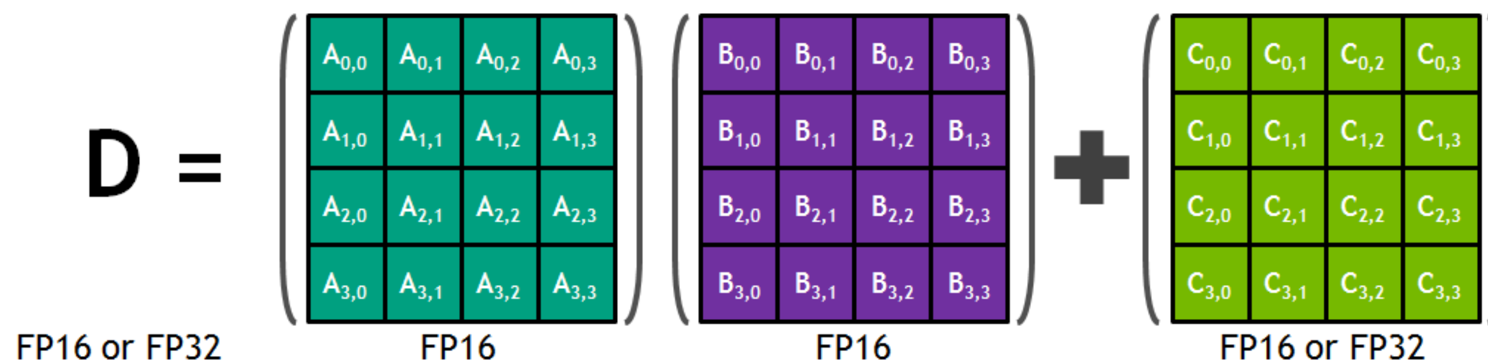
Many small RSVDs must be accelerated
→TSQR on Tensor Cores

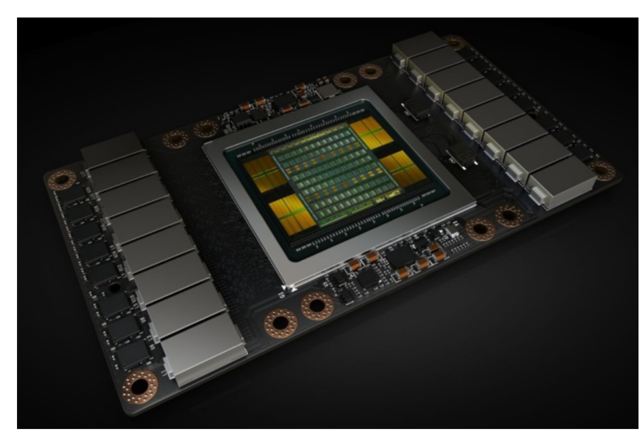
These methods run efficiently on modern architectures

Batch of many small dense matrices

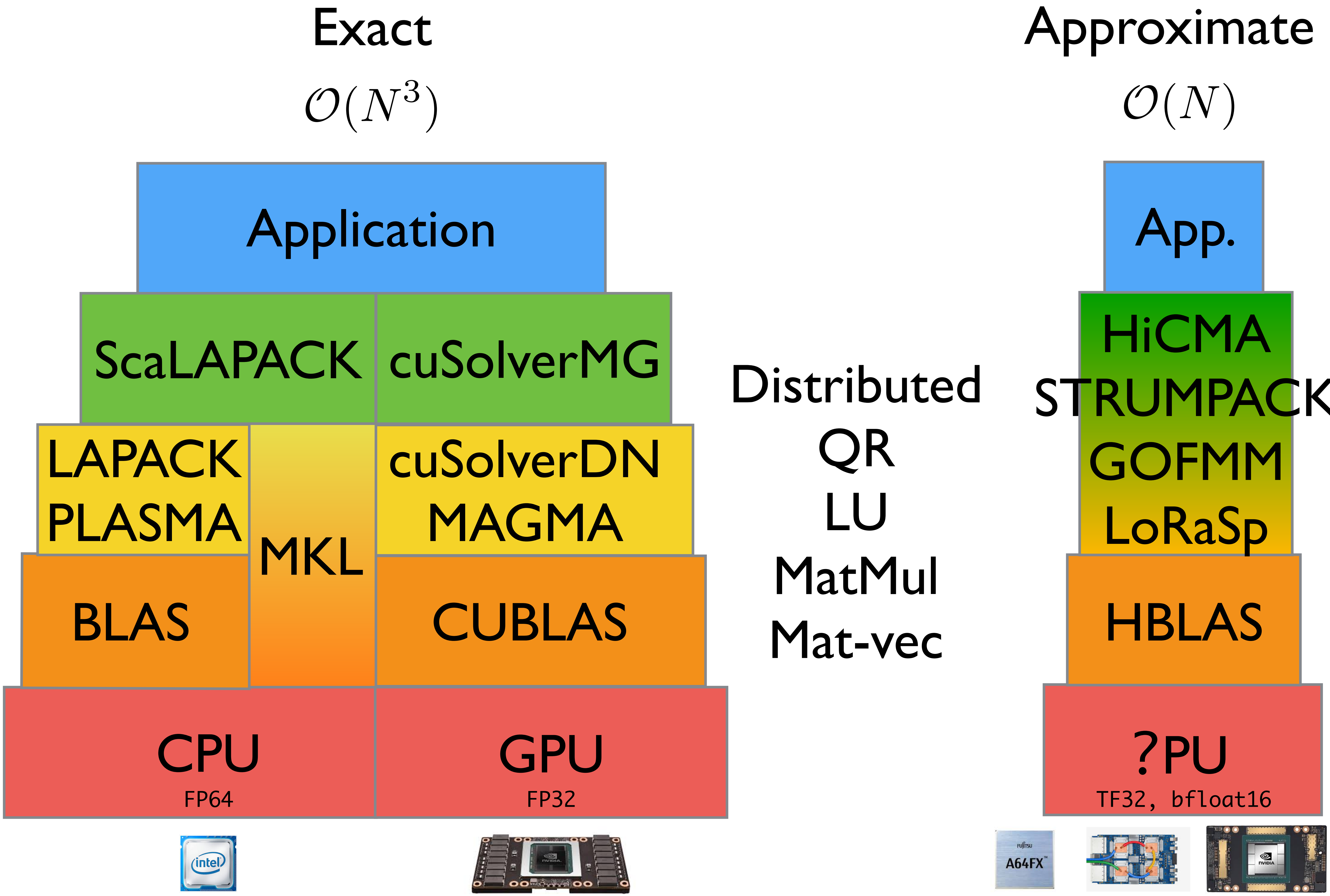
Low-rank approximation needs low arithmetic precision







Replacing Exact Linear Algebra with Low-Rank



List of implementations

	Method	Developer	url
AHMED	H-matrix	M. Bebendorf	https://github.com/xantares/ahmed
ASKIT	FMM	C. D. Yu	http://padas.ices.utexas.edu/libaskit
DMHM	H-matrix	J. Poulson	https://bitbucket.org/poulson/dmhm/src/default/
GOFMM	H^2-matrix	C. D. Yu	https://github.com/ChenhanYu/hmlp
H2Lib	H^2-matrix	S. Börm	https://github.com/H2Lib/H2Lib
H2Tools	H^2-matrix	A. Mikhalev	https://bitbucket.org/muxas/h2tools
HACApK	H-matrix	A. Ida	https://github.com/HLRA-JHPCN/HACApK-MAGMA
HiCMA	H-matrix	H. Ltaief	https://github.com/ecrc/hicma
HLib	H-matrix	L. Grasydyck	http://www.hlib.org
HLibPro	H-matrix	R. Kriemann	http://www.hlibpro.com
hmglib	H-matrix	P. Zaspel	https://github.com/zaspel/hmglib
HODLR	HODLR	A. Aminfar	https://github.com/amiraa127/Dense_HODLR
HSS	HSS	J. Xia	http://www.math.purdue.edu/~xiaj/
LoRaSp	H^2-matrix	H. Pouransari	https://bitbucket.org/hadip/lorasp
MUMPS-BLR	BLR	P. R. Amestoy	http://mumps.enseeiht.fr
STURMPACK	HSS	P. Ghysels	http://portal.nersc.gov/project/sparse/strumpack

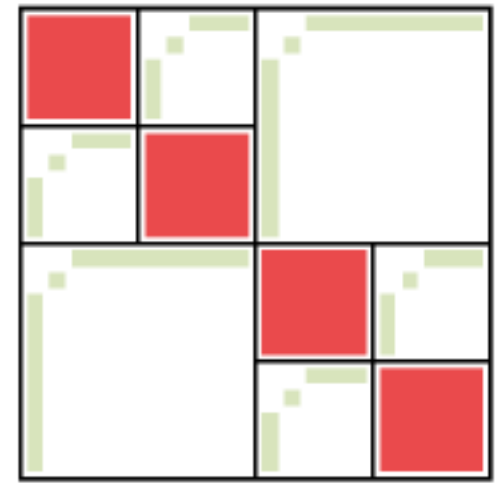
https://github.com/gchavez2/awesome_hierarchical_matrices

Differences between LRA methods

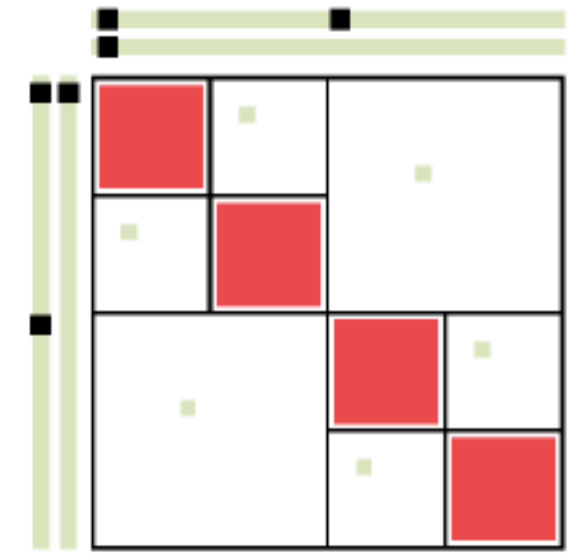
	Shared Basis	Admissibility
H-matrix	No	Strong
H^2-matrix	No	Strong
HODLR	No	Weak
HSS	Yes	Weak
RS/HIF	Yes	Strong
IFMM	Yes	FMM
(inv)-ASKIT	Yes	Strong
BLR	No	non-hierarchical
BLR^2	Yes	non-hierarchical

Nested Basis

Non-nested

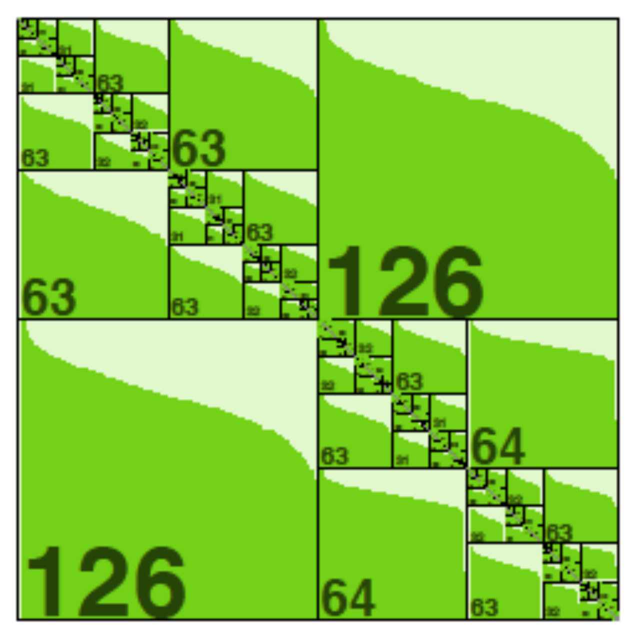


Nested



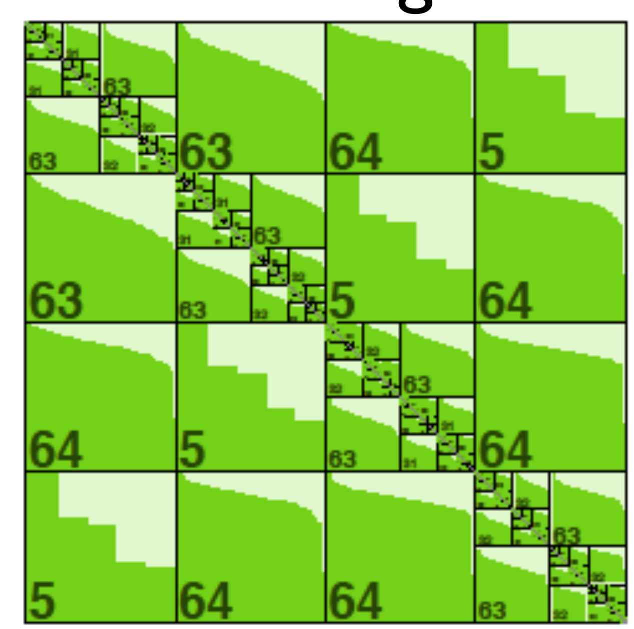
Admissibility

Weak



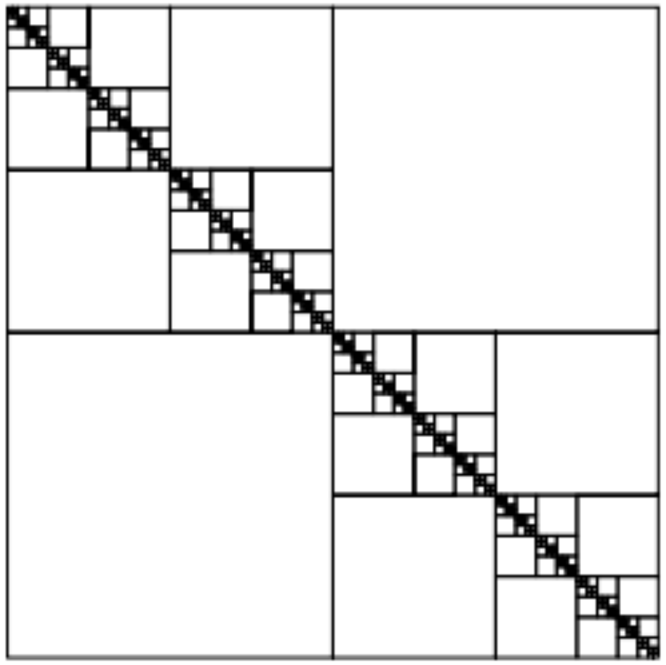
Strong

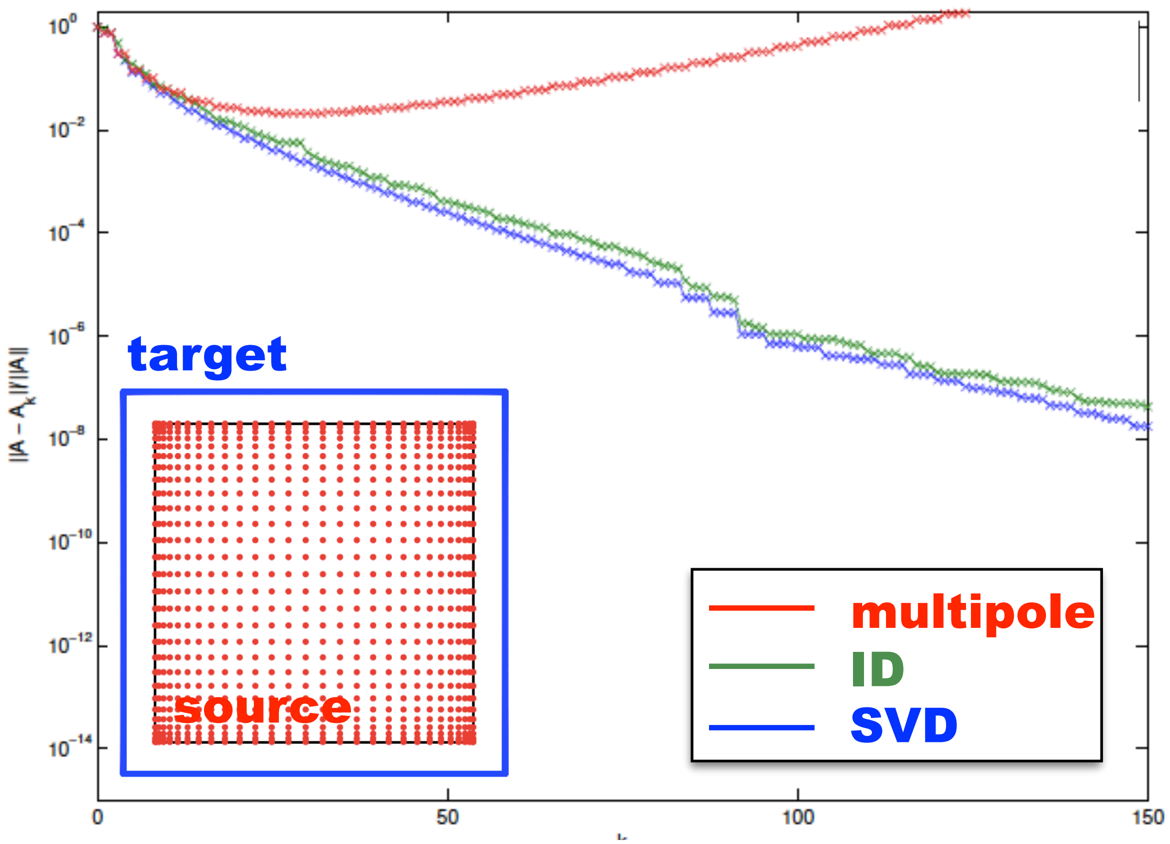
well separated



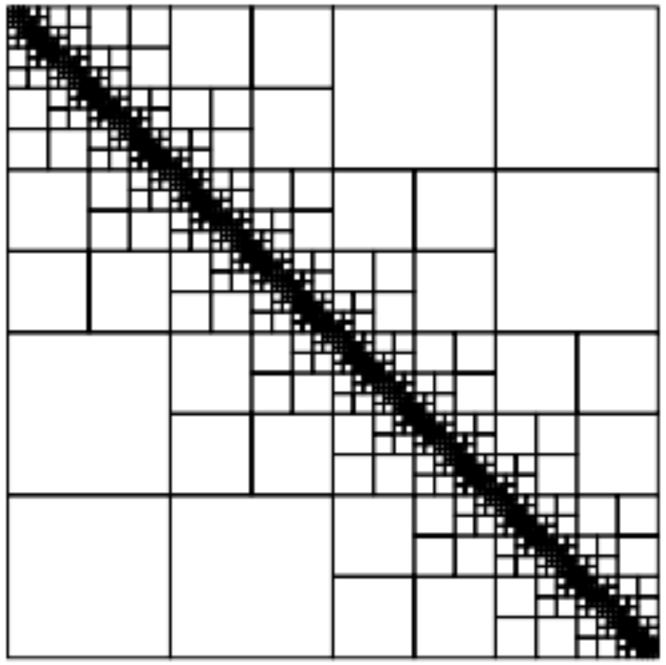
Admissibility condition

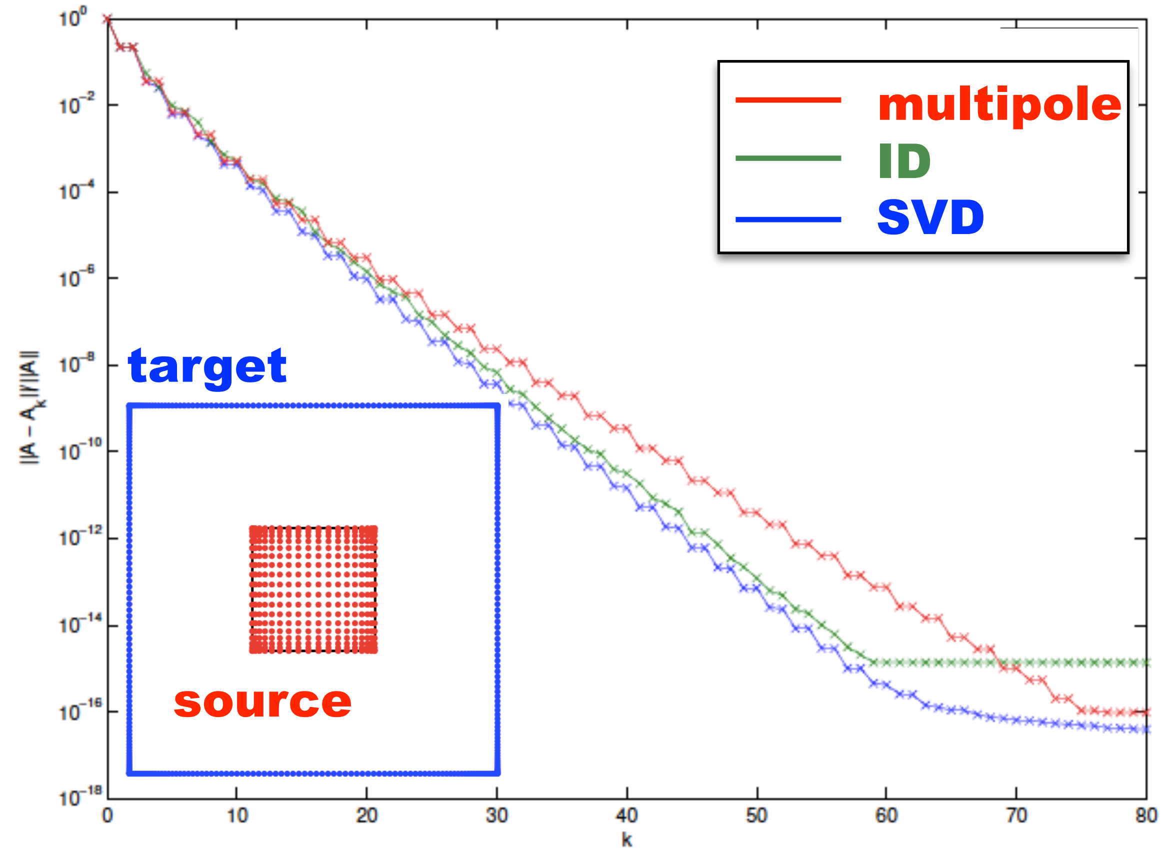
Weak admissibility





Standard admissibility





P. G. Martinsson, CBMS/NSF Conference on Fast Direct Solvers, June 2014

Nullity Theorem

David Keyes' lecture 2

Simple example of data sparsity with the 1D Laplacian

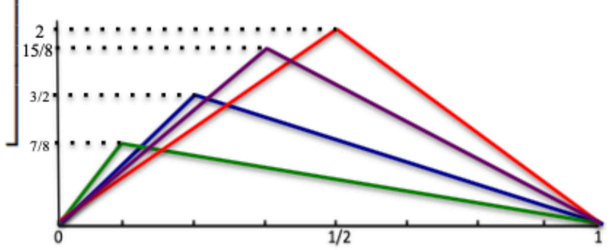
$$A = \begin{bmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & -1 & 2 & -1 & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \end{bmatrix}$$

Conformal off-diagonal blocks of A and A^{-1} admit low-rank representation with the same low rank
(Fiedler & Markham, 1986)

$$A^{-1} = \frac{1}{8} \times \begin{bmatrix} 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 6 & 12 & 10 & 8 & 6 & 4 & 2 \\ 5 & 10 & 15 & 12 & 9 & 6 & 3 \\ 4 & 8 & 12 & 16 & 12 & 8 & 4 \\ 3 & 6 & 9 & 12 & 15 & 10 & 5 \\ 2 & 4 & 6 & 8 & 10 & 12 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \begin{bmatrix} 4 & 3 & 2 & 1 \end{bmatrix}$$

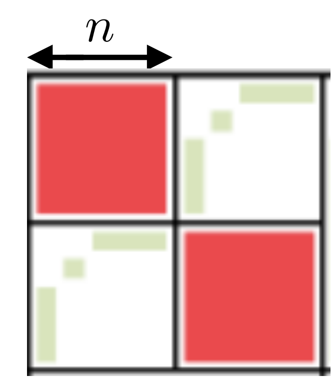


A is sparse but A^{-1} is dense

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

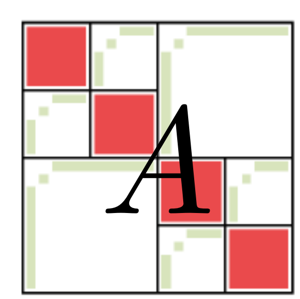
$$\begin{aligned} \text{nullity } A &= \text{nullity } H, \\ \text{nullity } B &= \text{nullity } F, \\ \text{nullity } C &= \text{nullity } G, \\ \text{nullity } D &= \text{nullity } E. \end{aligned}$$

$$\operatorname{rank}(A) + \operatorname{nullity}(A) = n.$$

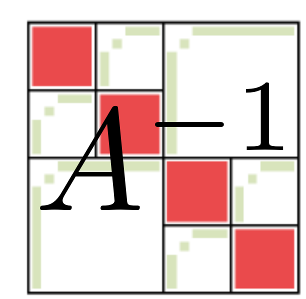


Apply it recursively

if



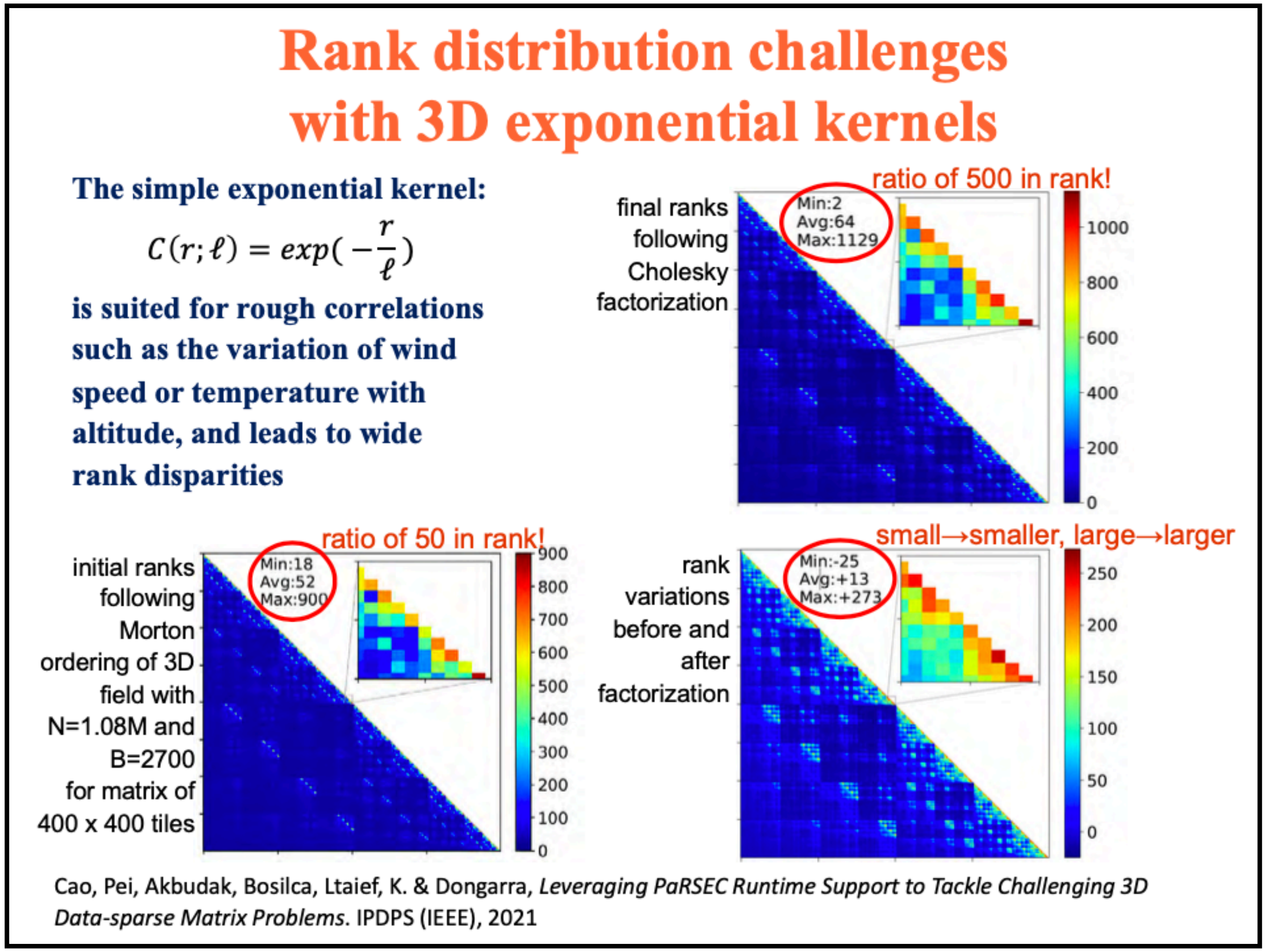
then

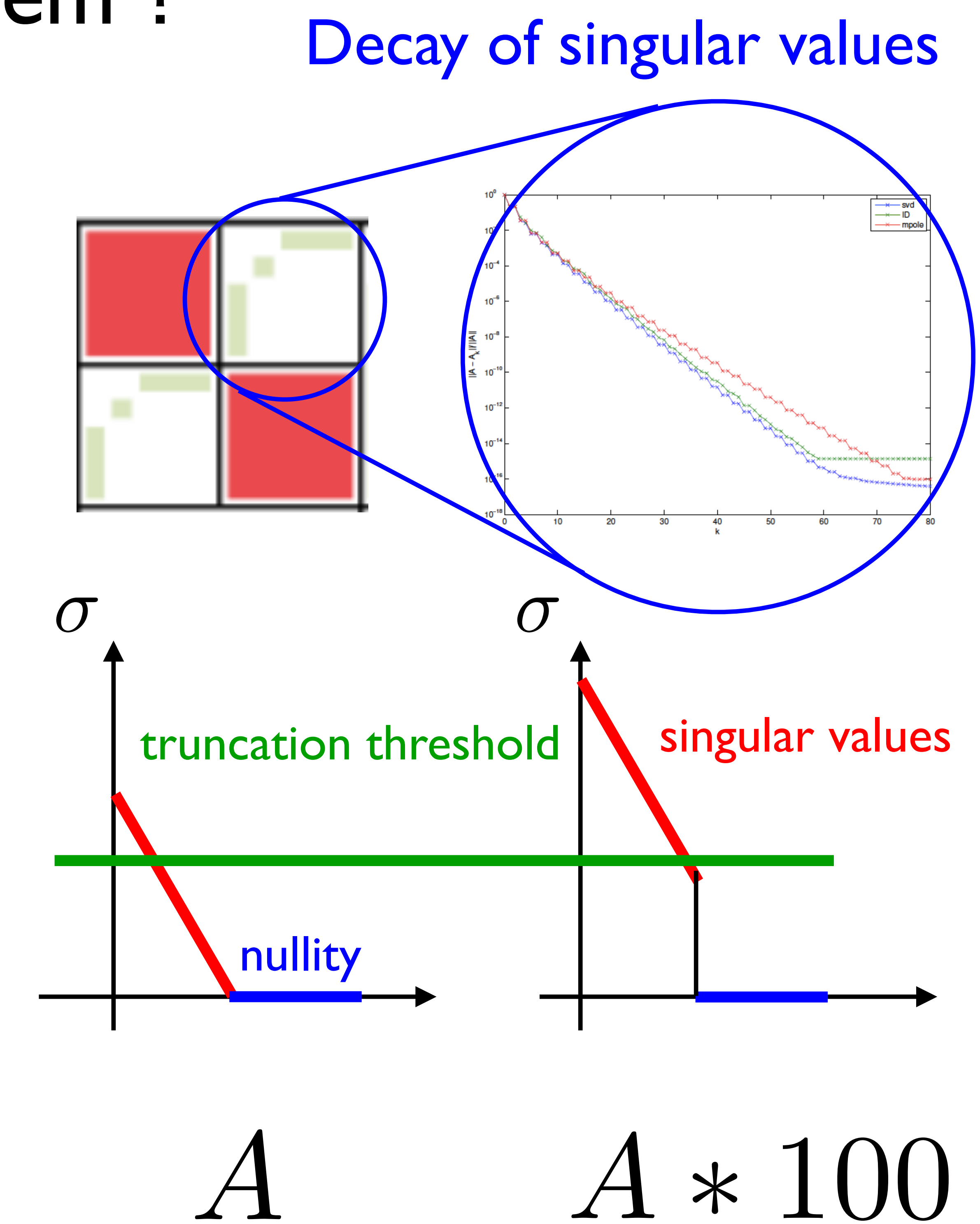


Gustafson, William H. (1984), "A note on matrix inversion", *Linear Algebra and Its Applications*, **57**: 71–73, doi:10.1016/0024-3795(84)90177-0
Strang, Gilbert; Nguyen, Tri (2004), "The interplay of ranks of submatrices" (PDF), *SIAM Review*, **46** (4): 637–646, doi:10.1137/S0036144503434381

Nullity Theorem ?

David Keyes' lecture 2





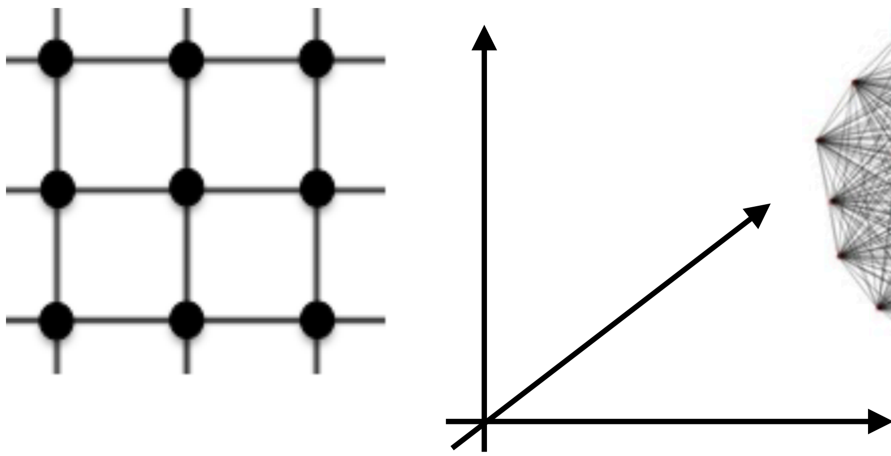
The exact rank does not grow, but the numerical/truncated rank does

Kronecker factorization

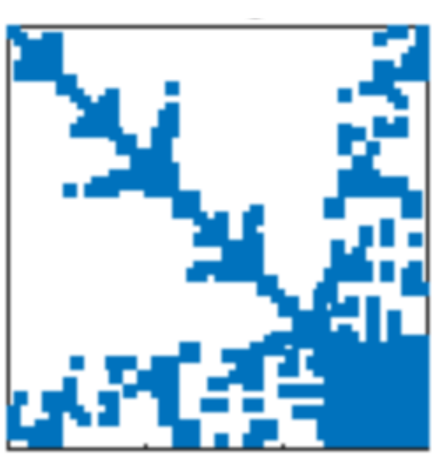
Structure of matrices

2-D or 3-D

Sparse

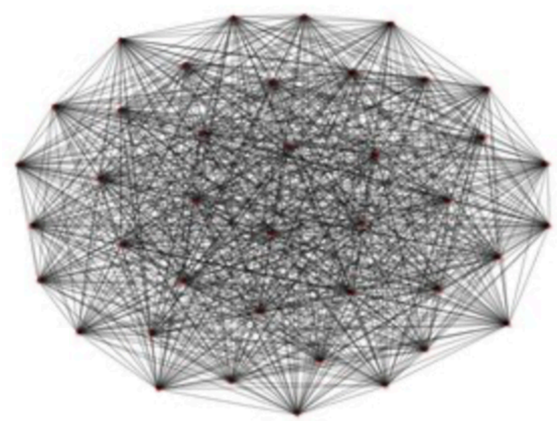


locally connected

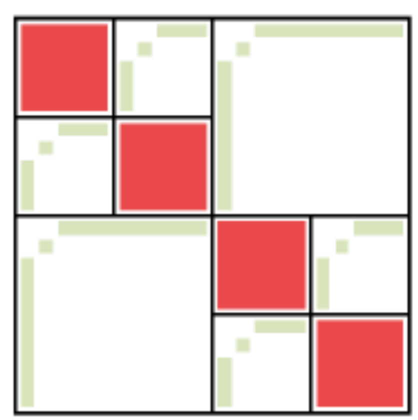


group based on connectivity

Dense

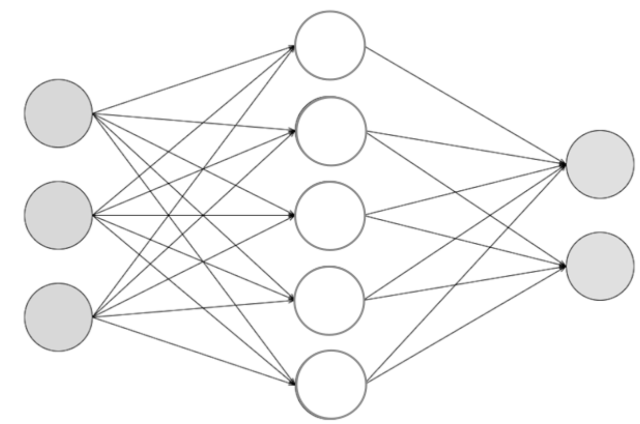


fully connected



group based on proximity

Million-D

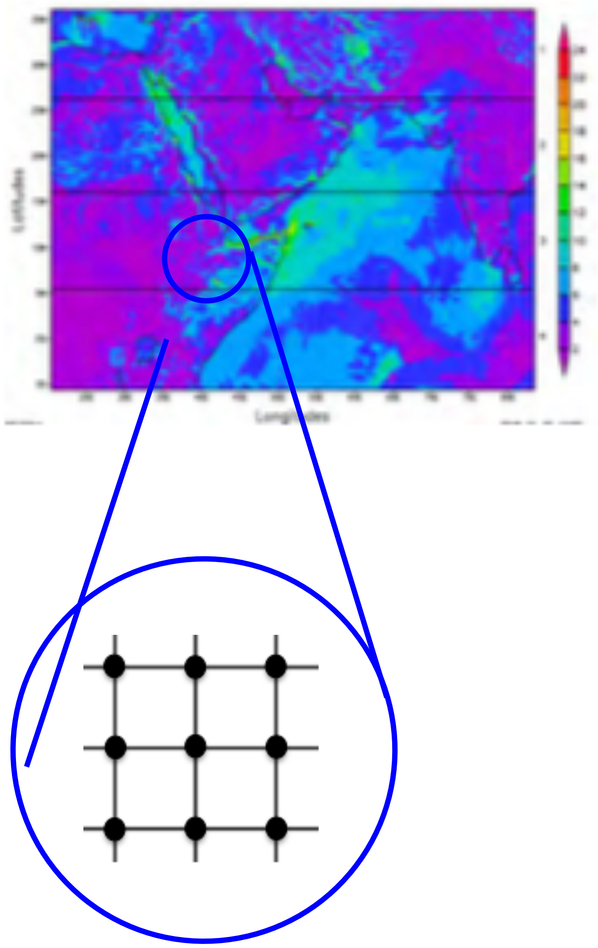


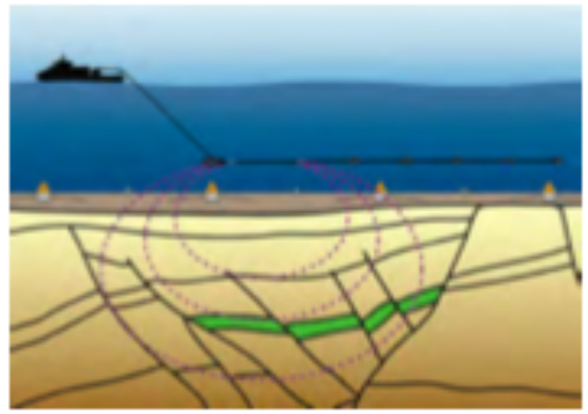
Each edge is a dimension

everything is close but dimensions are embedded

A different kind of piecewise linearity

Scientific computing





$$\int_\Omega f \phi d\Omega = 0$$

Conservation laws are integrated over low-D physical space

Local piecewise linearity

linear	linear	linear
linear	linear	linear
linear	linear	linear

$$Ax = b$$

Patch them together and you get something complex

Deep learning

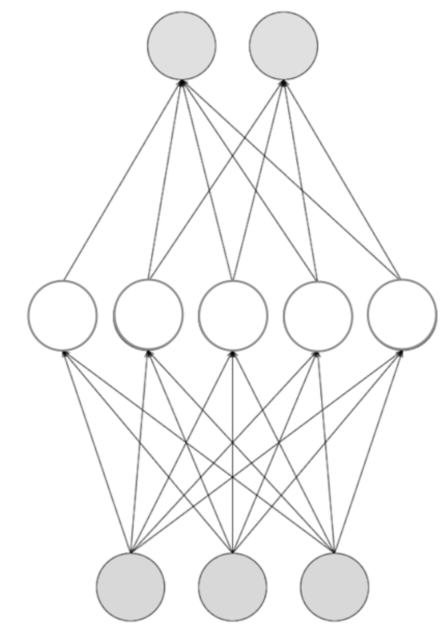
$f($ linear $)$

↑

$g($ linear $)$

↑

$h($ linear $)$

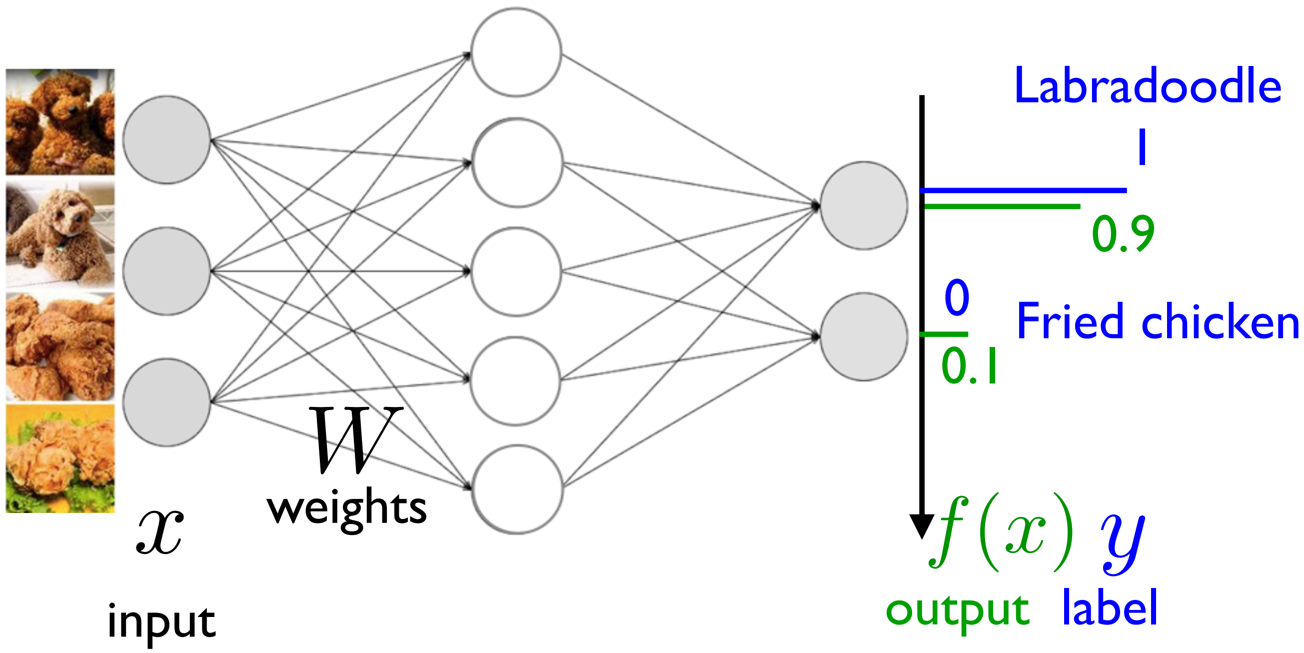


$$\mathbf{y} = f(g(h(\mathbf{x})))$$

Composite functions of of piecewise linear transformations are used to describe complex non-linear functions

Stack them up and you get something complex

Hessian, Fisher & Covariance Matrices



Negative log likelihood per class per data sample

$$l_{cd} = -\log f_c(x_d)$$

Loss per data sample

$$l_d = \sum_c^{class} y_c l_{cd}$$

Overall loss

$$L = \sum_d^{data} l_d$$

Gradient

$$\nabla L = \sum_d^{data} \frac{\partial l_d}{\partial W}$$

Hessian (Newton's method)

$$H = \sum_d^{data} \frac{\partial^2 l_d}{\partial W^2}$$

Covariance (KFAC)

$$C = \sum_d^{data} \left(\frac{\partial l_d}{\partial W}\right)^T \left(\frac{\partial l_d}{\partial W}\right)$$

Fisher (Natural gradient descent)

$$F = \sum_d^{data} \sum_c^{class} f_c(x_d) \left(\frac{\partial l_{cd}}{\partial W}\right)^T \left(\frac{\partial l_{cd}}{\partial W}\right)$$

I'm intentionally forgetting vector notation to decouple calculus from linear algebra in this slide

Applications of H,F,C Matrices

Predicting Hyperparameters

An Empirical Model of Large-Batch Training

Sam McCandlish* OpenAI sam@openai.com
Jared Kaplan Johns Hopkins University, OpenAI jaredk@jhu.edu
Dario Amodei OpenAI damodei@openai.com

and the **OpenAI Dota Team**†

$$\mathcal{B}_{noise} = \frac{tr(\mathbf{HC}^{-1})}{\mathbf{J}^T\mathbf{HJ}}$$

Optimizing Millions of Hyperparameters by Implicit Differentiation

Jonathan Lorraine, Paul Vicol, David Duvenaud
University of Toronto, Vector Institute
{lorraine, pvicol, duvenaud}@cs.toronto.edu

$$\frac{\partial \theta}{\partial \lambda} = -\mathbf{H}^{-1} \frac{\partial^2 \mathcal{L}}{\partial \theta \partial \lambda^\top}$$

Preconditioned Optimizers

When Does Preconditioning Help or Hurt Generalization?

*Shun-ichi Amari†, Jimmy Ba‡, Roger Grosse‡, Xuechen Li§,
Atsushi Nitanda¶, Taiji Suzuki¶, Denny Wu‡, Ji Xu‖

Gauss-Newton

$$\mathbf{F}(\theta)^{-1}\nabla\mathcal{L}(\theta) = \left\{\mathbf{J}_{f,\theta}^\top \mathcal{H}_{\ell,f} \mathbf{J}_{f,\theta}\right\}^{-1} \mathbf{J}_{f,\theta}^\top \frac{\partial \mathcal{L}(\theta)}{\partial f}$$

Gram-Gauss-Newton

$$\mathbf{F}(\theta)^{-1}\nabla\mathcal{L}(\theta) = \mathbf{J}_{f,\theta}^\top \left\{\mathcal{H}_{\ell,f} \mathbf{J}_{f,\theta} \mathbf{J}_{f,\theta}^\top\right\}^{-1} \frac{\partial \mathcal{L}(\theta)}{\partial f}$$

Bayesian Inference

Noisy Natural Gradient as Variational Inference

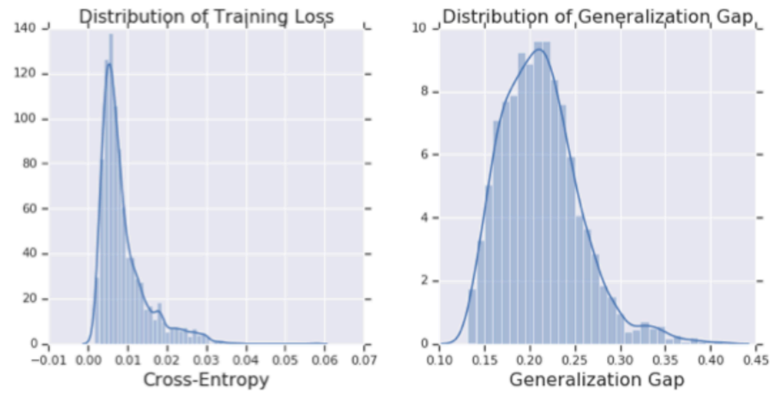
Guodong Zhang [*1 2] **Shengyang Sun** [*1 2] **David Duvenaud** [1 2] **Roger Grosse** [1 2]

$$\left\{\mathbf{F}(\theta) + \sigma^{-2}\mathbf{I}\right\}^{-1} \nabla\mathcal{L}(\theta)$$
$$= \left\{\mathbf{J}_{f,\theta}^\top \mathcal{H}_{\ell,f} \mathbf{J}_{f,\theta} + \sigma^{-2}\mathbf{I}\right\}^{-1} \mathbf{J}_{f,\theta}^\top \frac{\partial \mathcal{L}(\theta)}{\partial f}$$

Generalization Metrics

Fantastic Generalization Measures and Where to Find Them

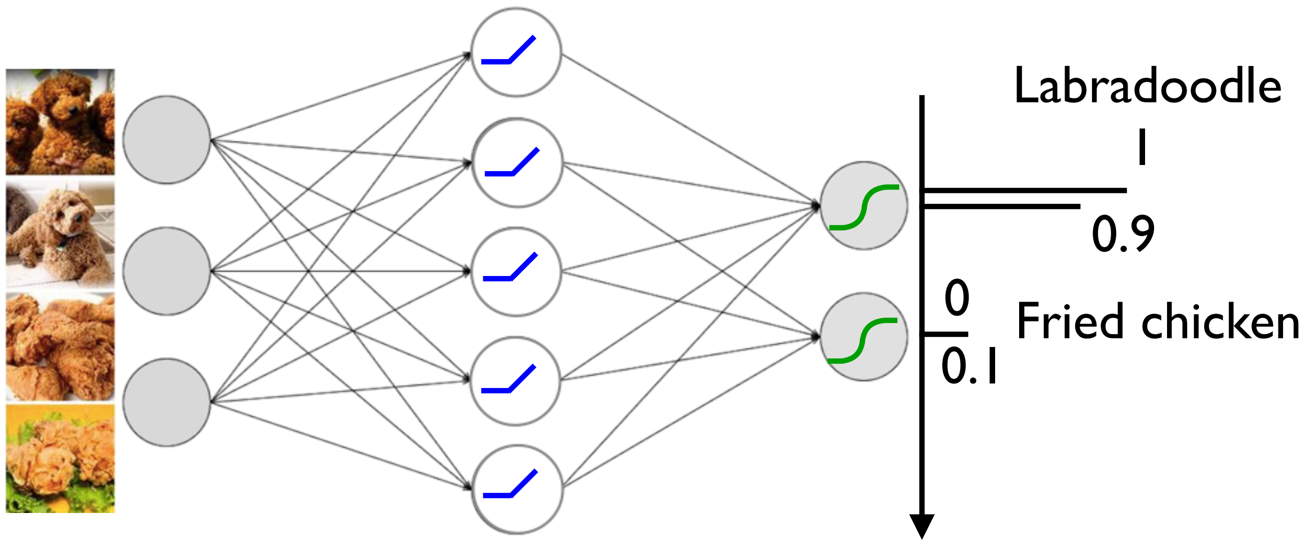
Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi
Dilip Krishnan, Samy Bengio
Google



- Spectral bound
- Path norm
- Fisher-Rao metric
- Variance of gradients
- Sharpness
- PAC-Baysian
- Takeuchi Information Criteria

$$\mathrm{TIC}(\theta) = -\log p(y|\theta) + \frac{1}{N}\mathrm{tr}\left(\boldsymbol{H}(\theta^*)^{-1}\boldsymbol{C}(\theta^*)\right)$$

Jacobian-Vector Product



$x = h_0 \quad h_1 = f(u_0) \quad p = f(u_1)$

$u_0 = W_0 h_0 \quad u_1 = W_1 h_1$

$$\frac{\partial h_1}{\partial u_0} * \frac{\partial u_1}{\partial h_1} * \frac{\partial l_d}{\partial u_1} = \frac{\partial l_d}{\partial W}$$

$$* \frac{\partial u_0}{\partial W_0} \qquad * \frac{\partial u_1}{\partial W_1}$$

Jacobian-vector product

$$\underbrace{\left(\frac{\partial u_1}{\partial W_0}\right)^T}_{25 \times 2} * \underbrace{\frac{\partial l_d}{\partial u_1}}_{2 \times 1} = \underbrace{\frac{\partial l_d}{\partial W}}_{25 \times 1}$$

Negative log likelihood per class per data sample

$$l_{cd} = -\log f_c(x_d)$$

Loss per data sample

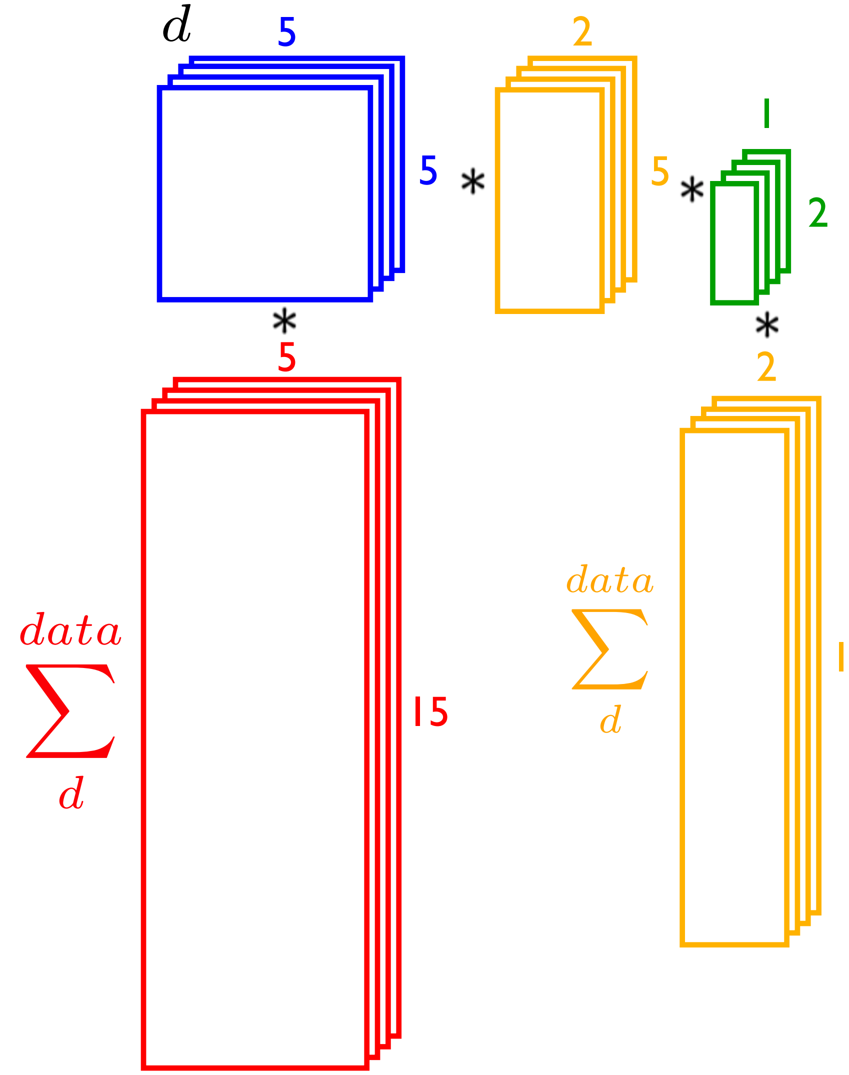
$$l_d = \sum_c^{class} y_c l_{cd}$$

Overall loss

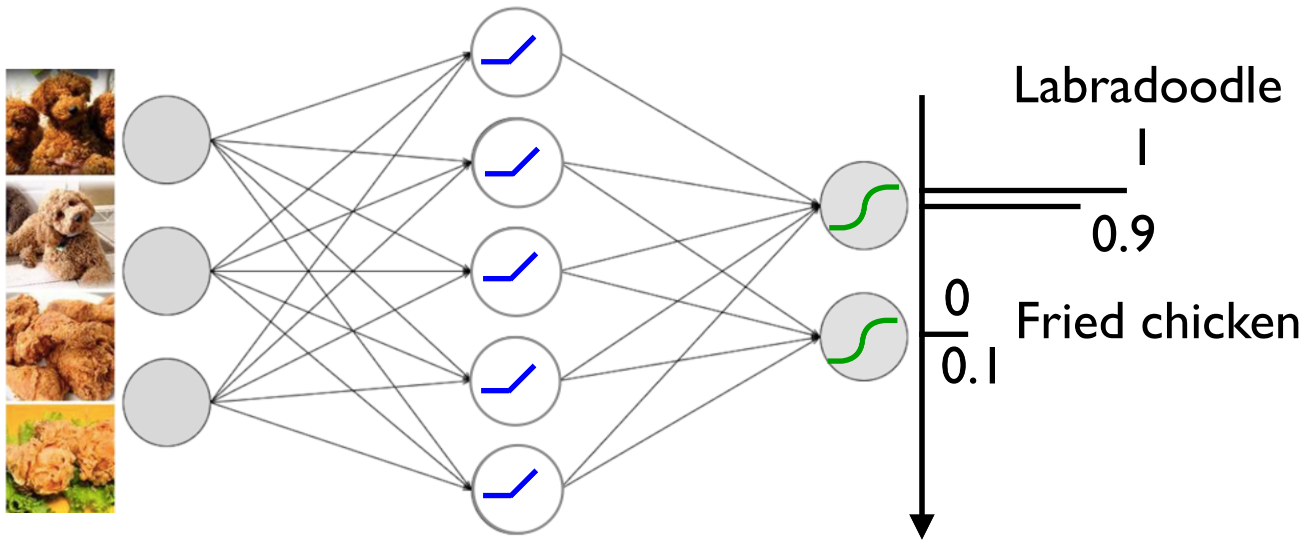
$$L = \sum_d^{data} l_d$$

Gradient

$$\nabla L = \sum_d^{data} \frac{\partial l_d}{\partial W}$$



Jacobian-Matrix Product



$$x = h_0 \qquad h_1 = f(u_0) \qquad p = f(u_1)$$

$$u_0 = W_0 h_0 \qquad u_1 = W_1 h_1$$

Gradient of first layer per data sample

$$\frac{\partial l_d}{\partial W_0} = \frac{\partial u_0}{\partial W_0} * \frac{\partial h_1}{\partial u_0} * \frac{\partial u_1}{\partial h_1} * \frac{\partial l_d}{\partial u_1}$$

Hessian of first layer per data sample

$$\frac{\partial^2 l_d}{\partial W_0^2} = \frac{\partial^2 u_0}{\partial W_0^2} * \frac{\partial h_1}{\partial u_0} * \frac{\partial u_1}{\partial h_1} * \frac{\partial l_d}{\partial u_1} \rightarrow 0$$

$$+ \left(\frac{\partial u_0}{\partial W_0}\right)^2 * \frac{\partial^2 h_1}{\partial u_0^2} * \frac{\partial u_1}{\partial h_1} * \frac{\partial l_d}{\partial u_1} \rightarrow 0$$

$$+ \left(\frac{\partial u_0}{\partial W_0}\right)^2 * \left(\frac{\partial h_1}{\partial u_0}\right)^2 * \frac{\partial^2 u_1}{\partial h_1^2} * \frac{\partial l_d}{\partial u_1} \rightarrow 0$$

Gauss-Newton approximation →

$$+ \left(\frac{\partial u_0}{\partial W_0}\right)^2 * \left(\frac{\partial h_1}{\partial u_0}\right)^2 * \left(\frac{\partial u_1}{\partial h_1}\right)^2 * \frac{\partial^2 l_d}{\partial u_1^2}$$

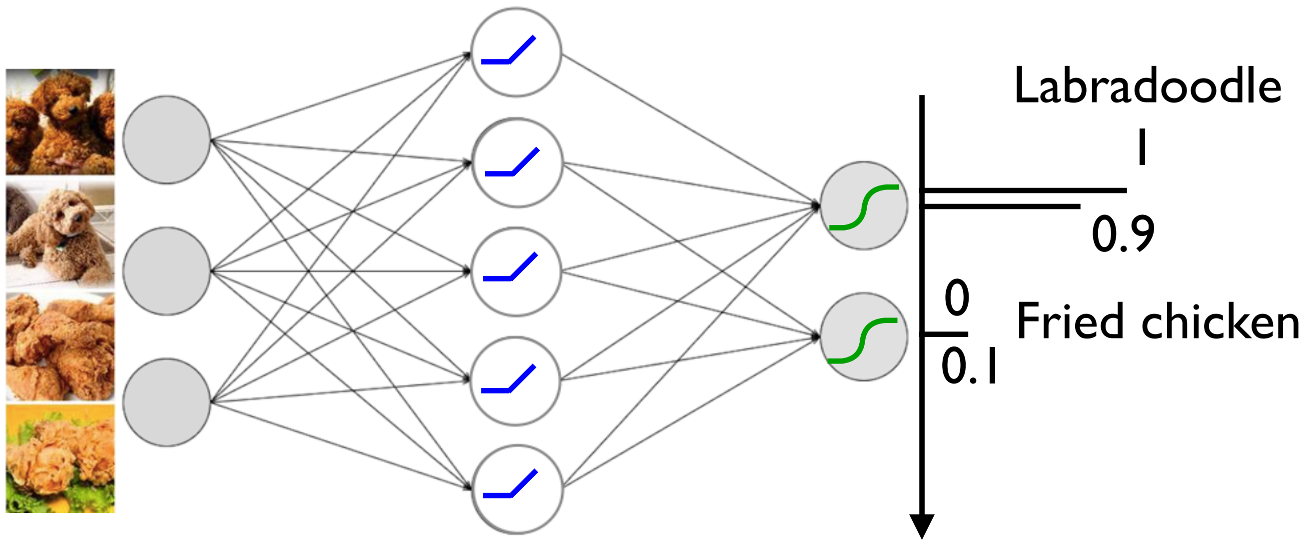
$$= \left(\frac{\partial u_1}{\partial W_0}\right)^T * \frac{\partial^2 l_d}{\partial u_1^2} * \left(\frac{\partial u_1}{\partial W_0}\right)$$

$$\left(\frac{\partial u_1}{\partial W_0} * \sqrt{\frac{\partial^2 l_d}{\partial u_1^2}}\right)^T \left(\sqrt{\frac{\partial^2 l_d}{\partial u_1^2}} * \frac{\partial u_1}{\partial W_0}\right)$$

$$\underbrace{\left(\frac{\partial u_1}{\partial W_0}\right)^T}_{25 \times 2} * \underbrace{\sqrt{\frac{\partial^2 l_d}{\partial u_1^2}}}_{2 \times 2}$$ Jacobian-matrix product

$$\underbrace{\left(\frac{\partial u_1}{\partial W_0}\right)^T}_{25 \times 2} * \underbrace{\frac{\partial^2 l_d}{\partial u_1^2}}_{2 \times 2} * \underbrace{\left(\frac{\partial u_1}{\partial W_0}\right)}_{2 \times 25}$$

Kronecker Factors



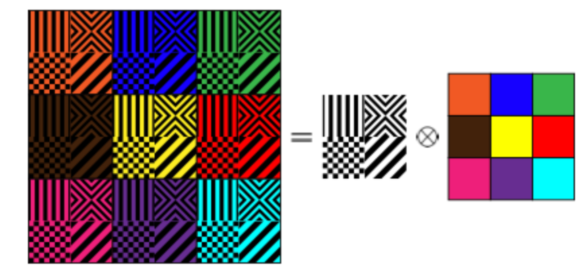
$$x = h_0 \quad h_1 = f(u_0) \quad p = f(u_1)$$

$$u_0 = W_0 h_0 \quad u_1 = W_1 h_1$$

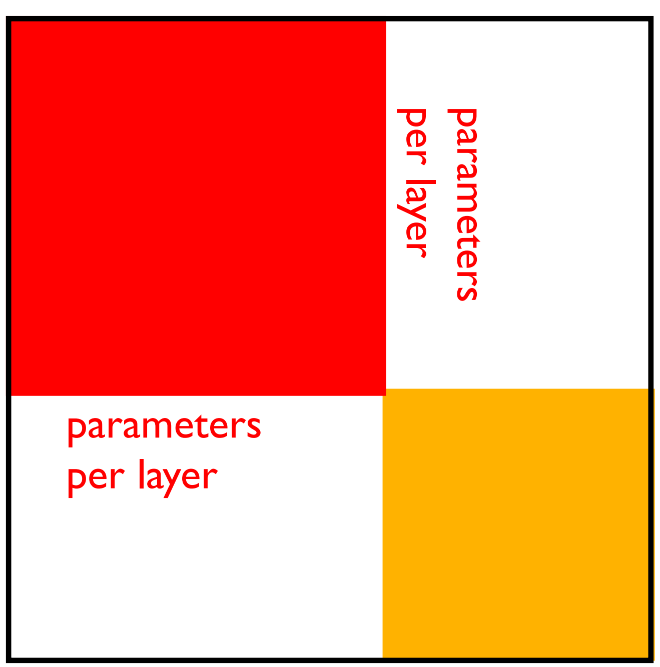
$$\frac{\partial l_d}{\partial W_0} = \frac{\partial u_0}{\partial W_0} \otimes \frac{\partial h_1}{\partial u_0} * \frac{\partial u_1}{\partial h_1} * \frac{\partial l_d}{\partial u_1}$$

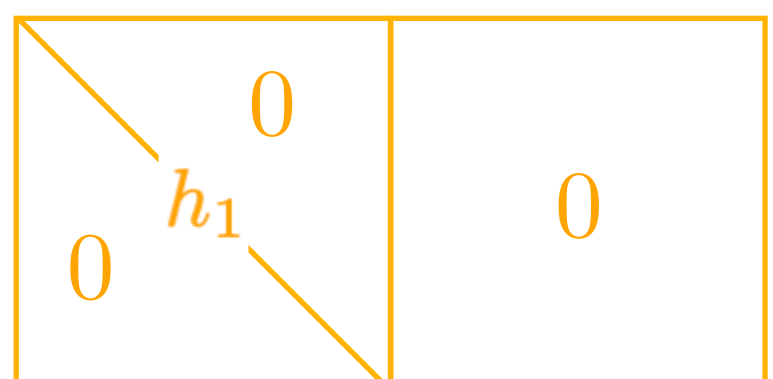
$$= \frac{\partial u_0}{\partial W_0} \otimes \frac{\partial l_d}{\partial u_0}$$

$$\frac{\partial l_d}{\partial W_1} = \frac{\partial u_1}{\partial W_1} \otimes \frac{\partial l_d}{\partial u_1}$$



$$\begin{pmatrix} \frac{\partial u_1}{\partial W_1} & \frac{\partial u_1}{\partial W_2} & \cdots & \frac{\partial u_1}{\partial W_{10}} \\ \frac{\partial u_2}{\partial W_1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial u_5}{\partial W_1} & \cdots & \cdots & \frac{\partial u_5}{\partial W_{10}} \end{pmatrix} \in \mathbb{R}^{10 \times 5}$$





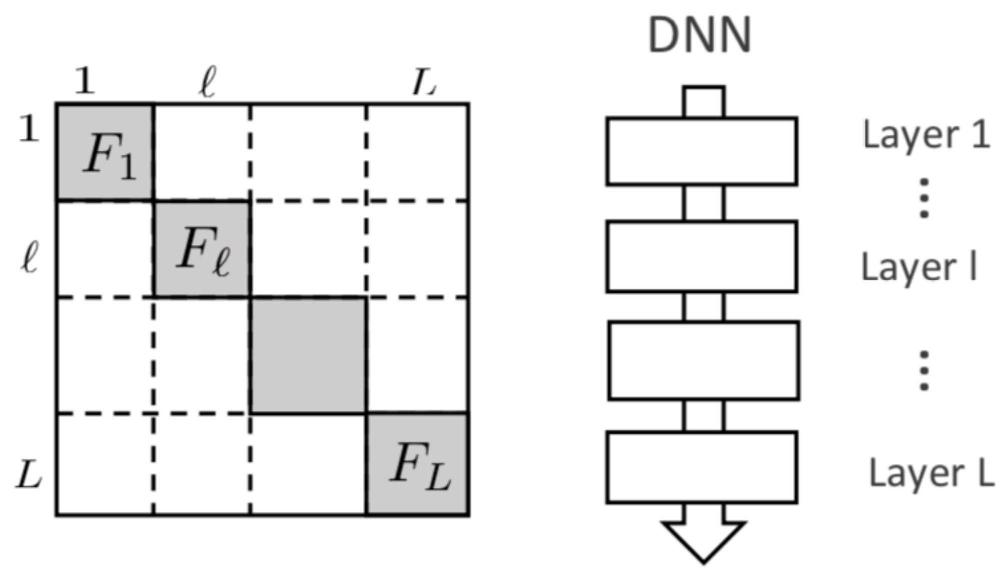
K-FAC

Kronecker-**F**actored **A**pproximate **C**urvature (J. Martens+, ICML 2015)

Step 1. Layer-wise block-diagonal approximation

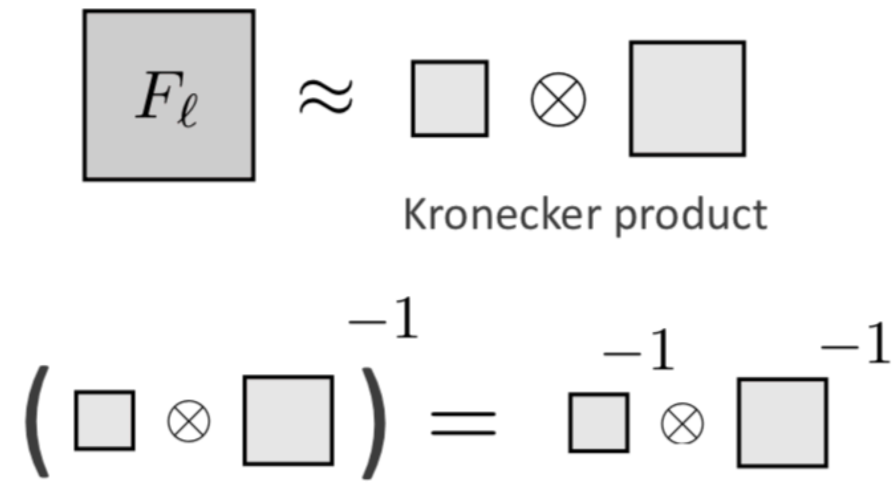
$C = F \approx$

FIM



easier to invert $O(N^3) \rightarrow O(N_\ell^3)$

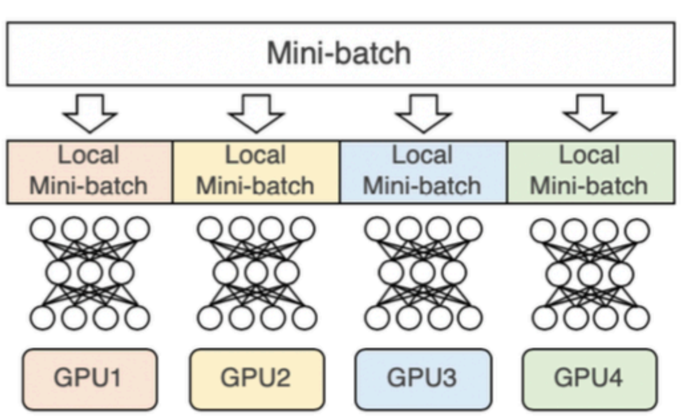
Step 2. Kronecker-factorization (for each layer)



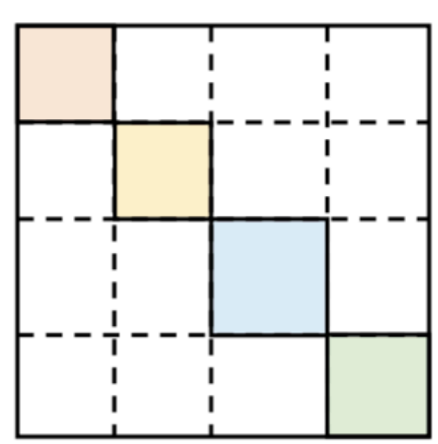
much easier to invert $O(N_\ell^3) \rightarrow O(N_\ell^{3/2})$

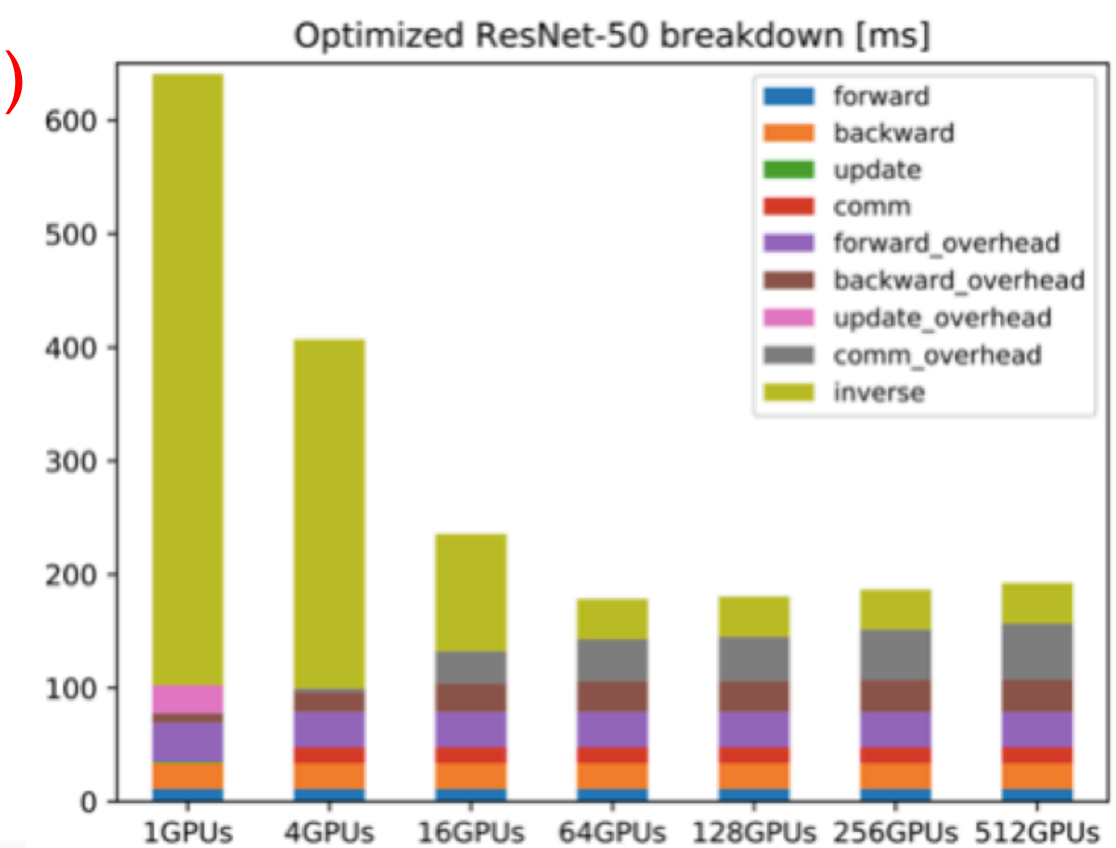
Distributed K-FAC (K. Osawa et el., CVPR2019, TPAMI)

data-parallelism × layers-parallelism



$F \approx$





Summary

- The structure of matrices depend on the underlying geometry
- Sparsity is related to connectivity, where as rank is related to proximity
- FMM is a matrix-free mat-vac of a hierarchical low-rank matrix
- Using the matrix form is useful for multiple right hand sides and factorization
- Nesting of bases and admissibility distinguishes the various hierarchical formats
- FMM and hierarchical matrices both rely on geometric separation
- Everything is close in high dimensions but dimensions may also have structure
- Embedding of dimensions leads to Kronecker structure
- Exploiting this structure is useful for very high-dimension problems

Thank you