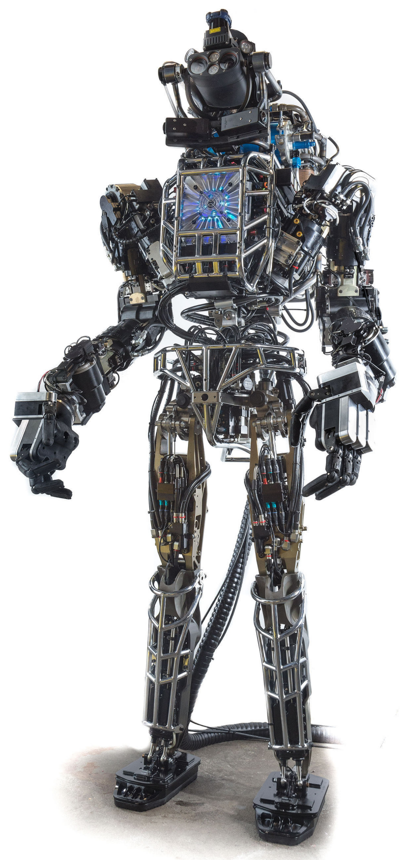
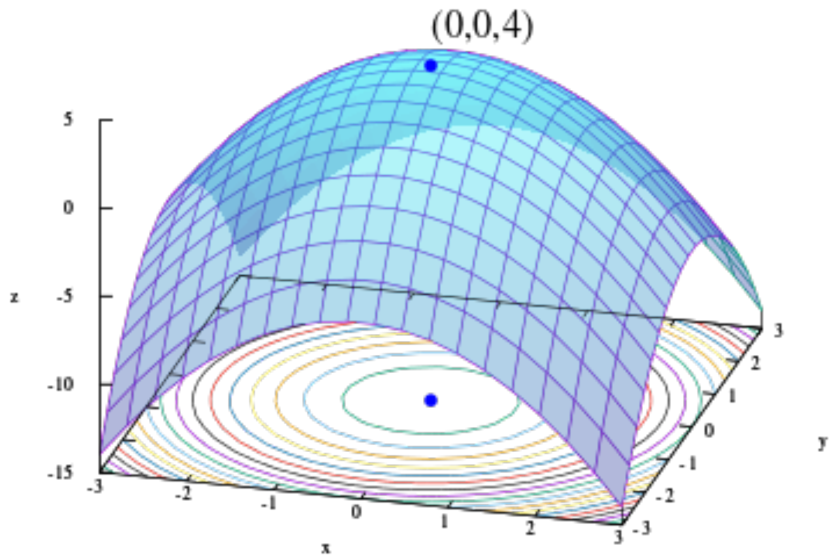
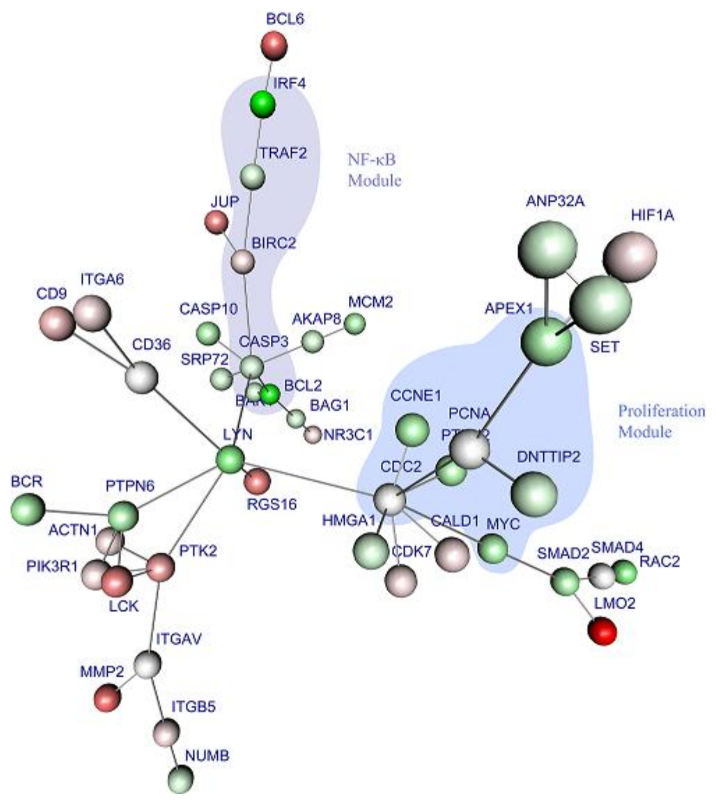
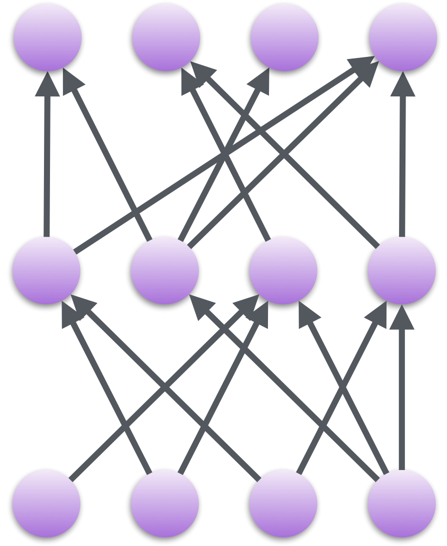
Compilers for Sparse Applications





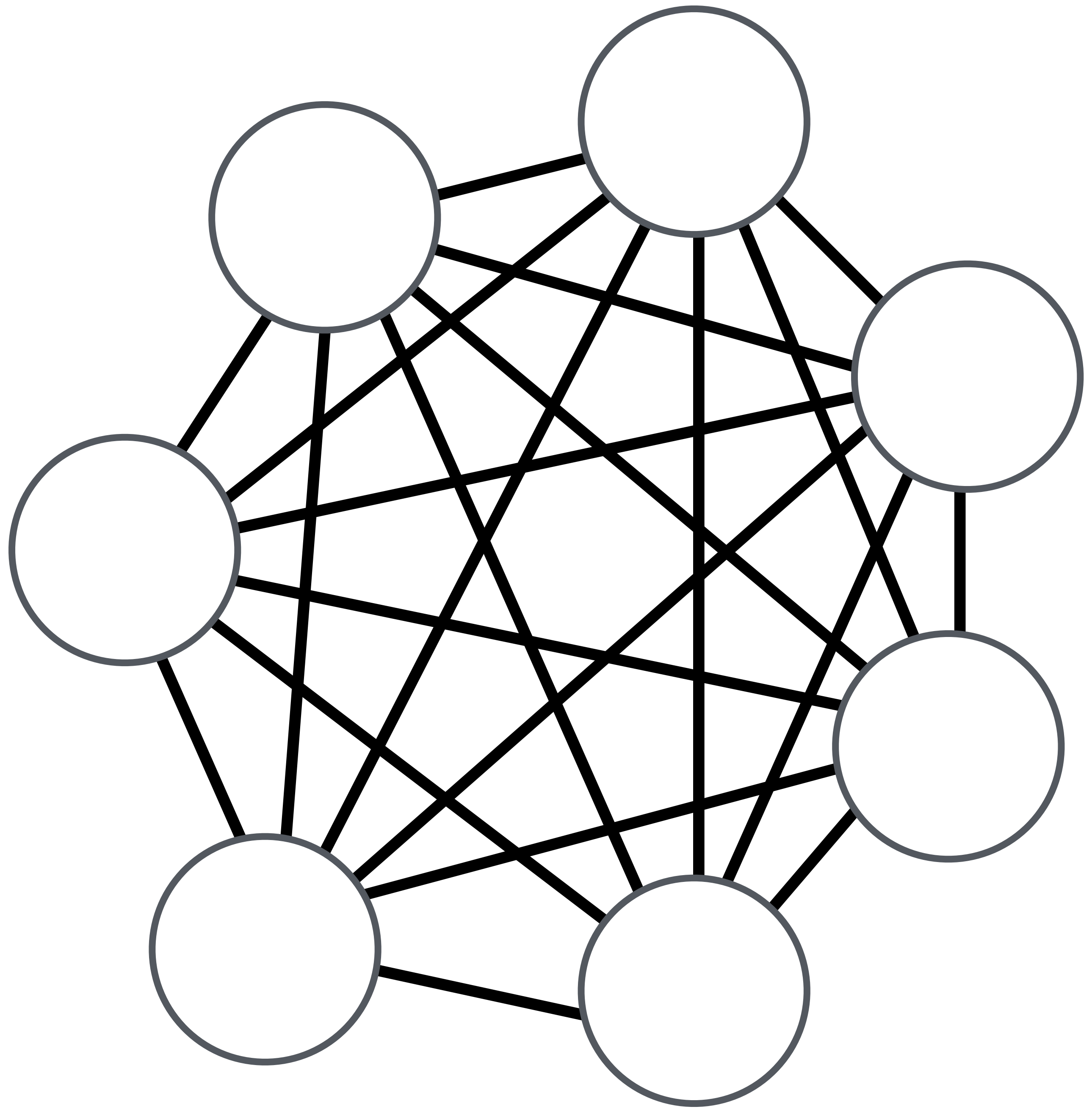




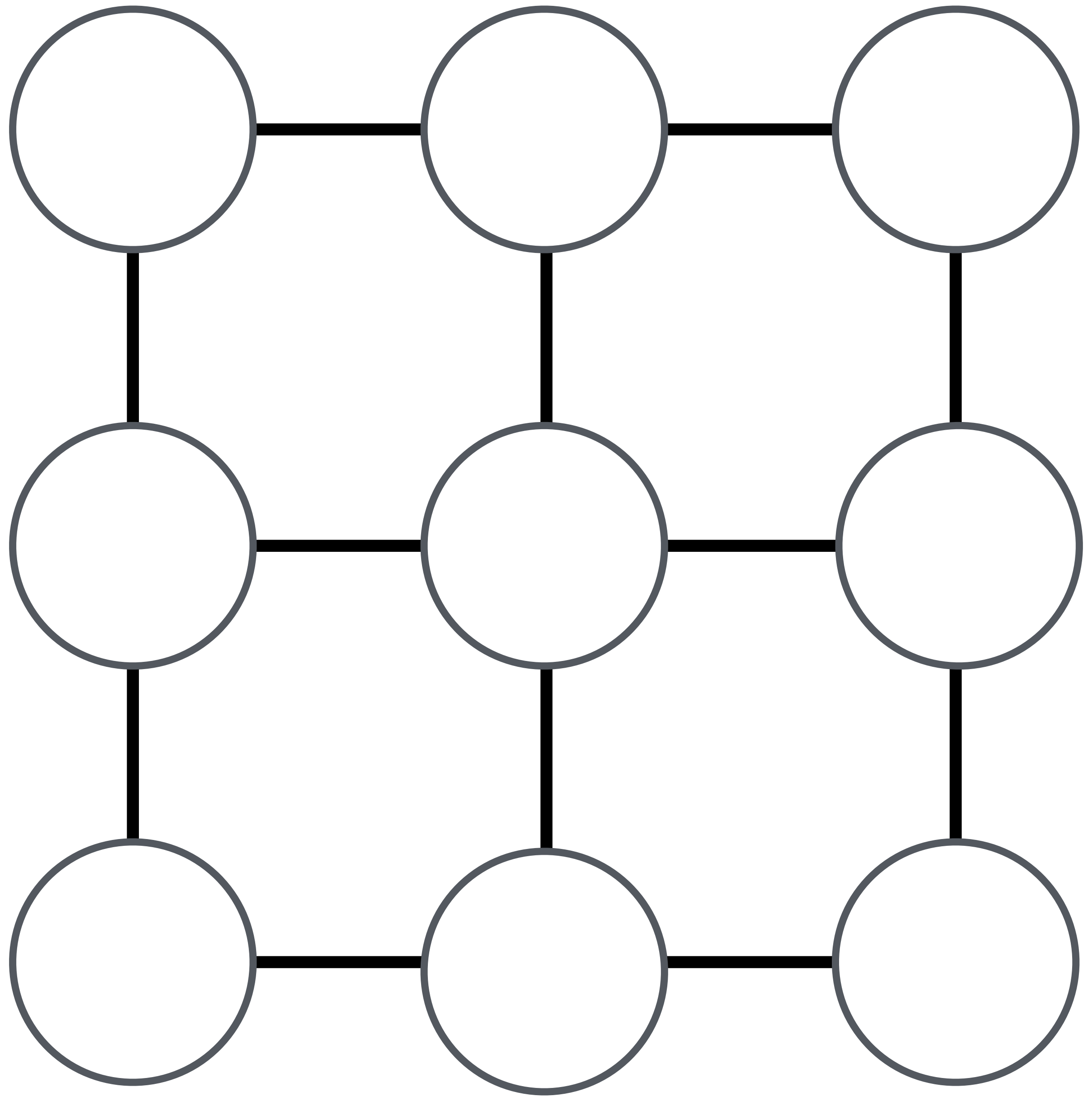
Fred Kjolstad
Assistant Professor

fredrikbk.com

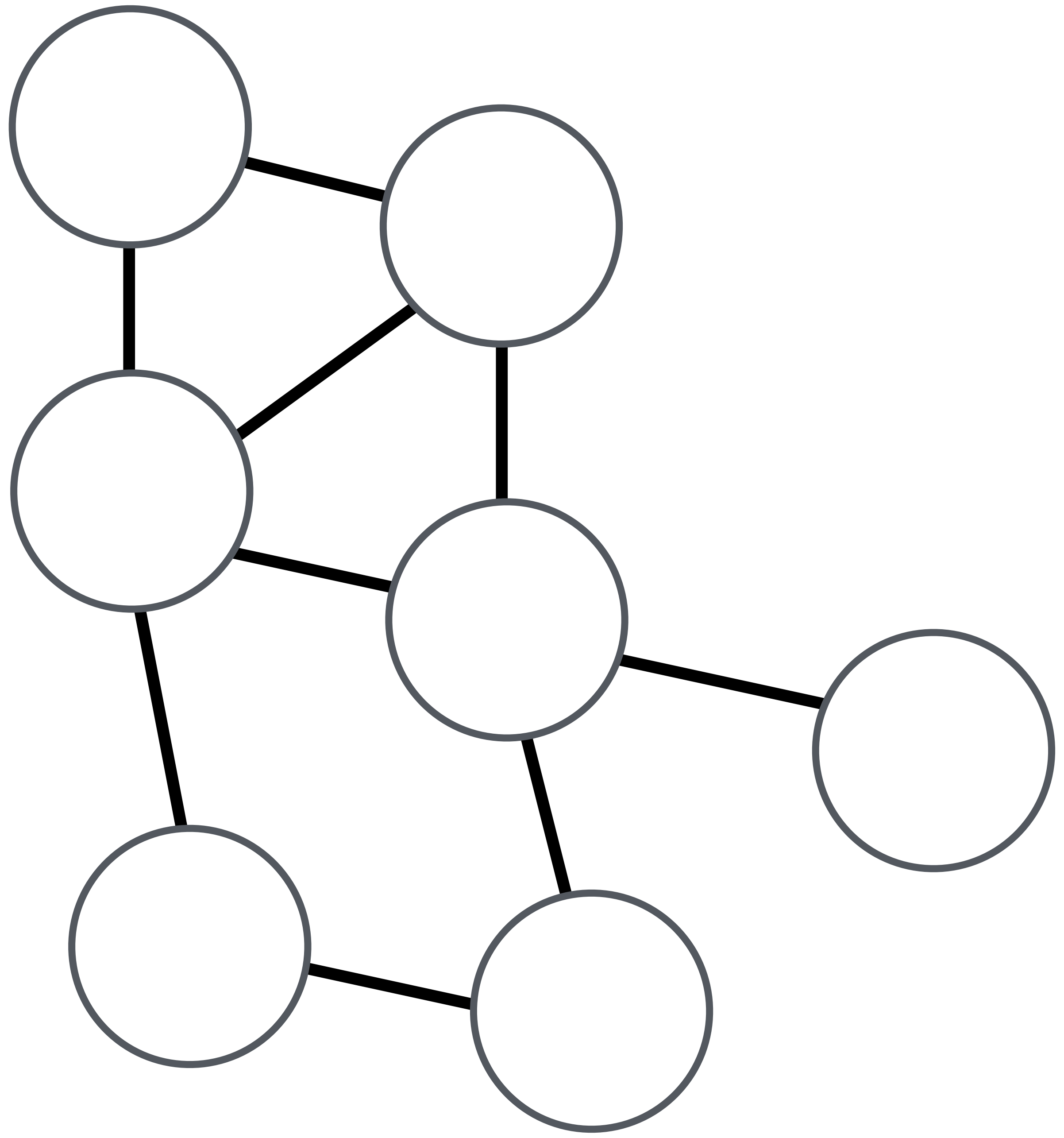
Sparsity and Irregularity of Systems



Dense System

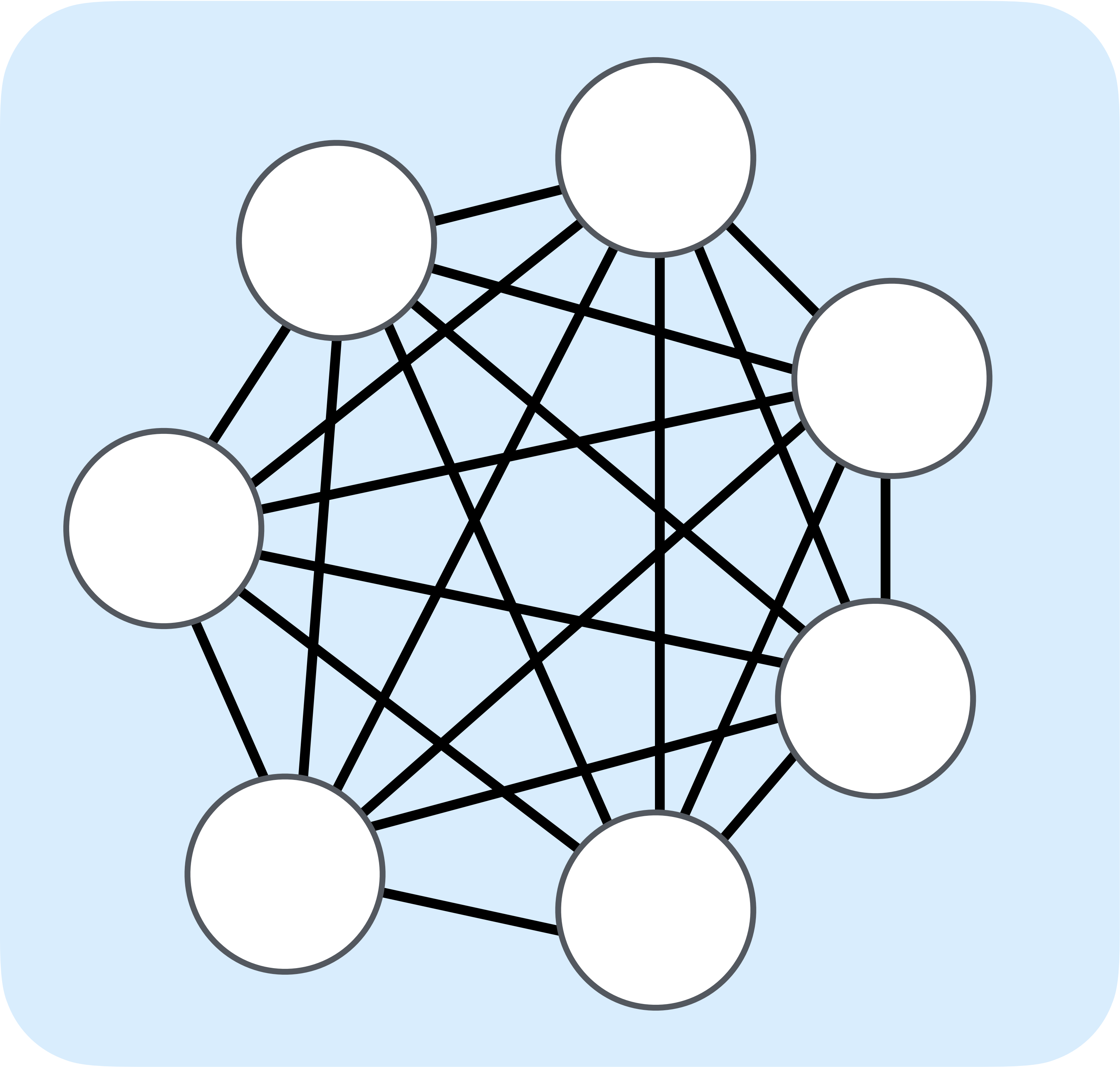


Sparse Regular System

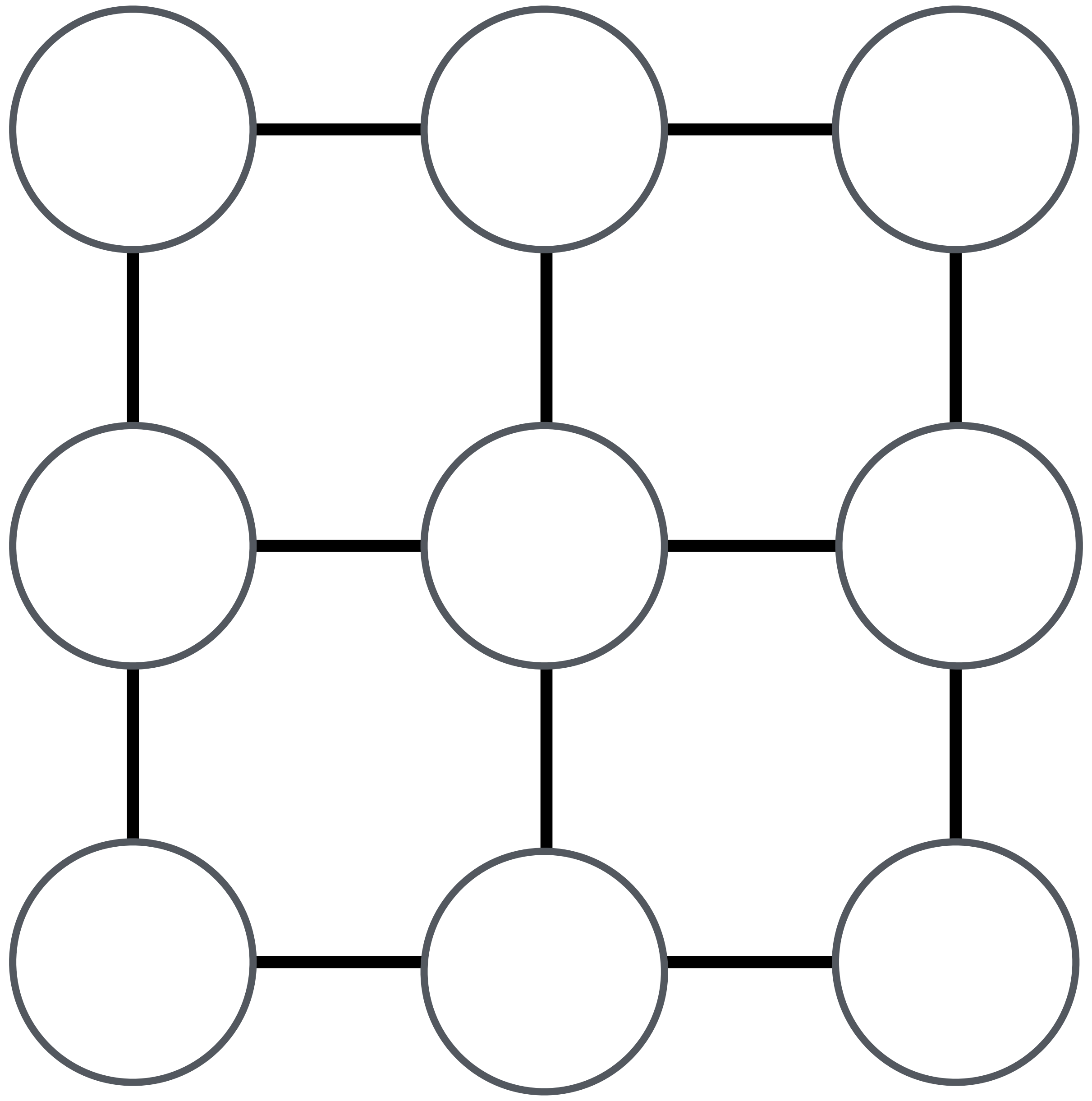


Sparse Irregular System

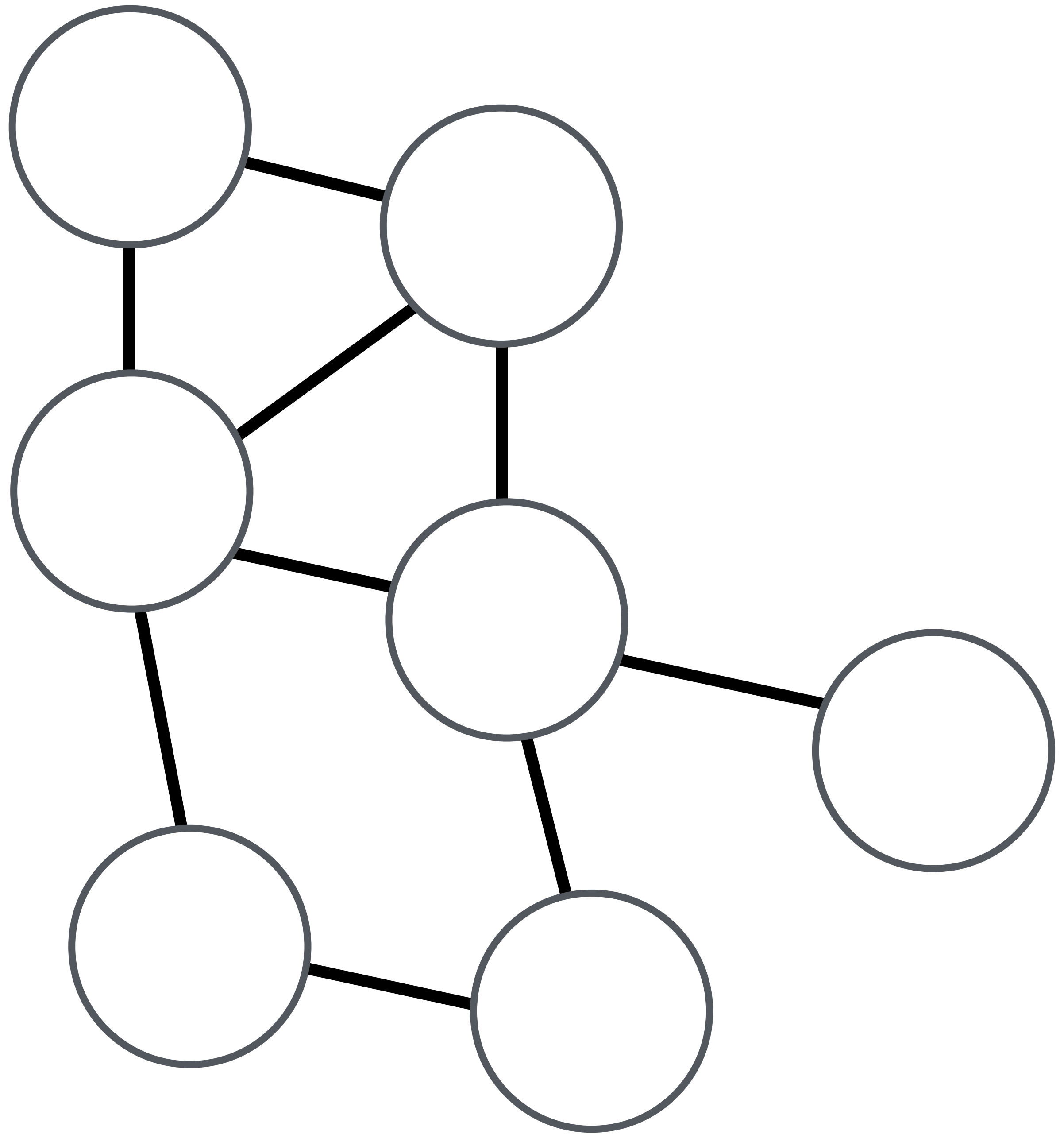
Sparsity and Irregularity of Systems



Dense System

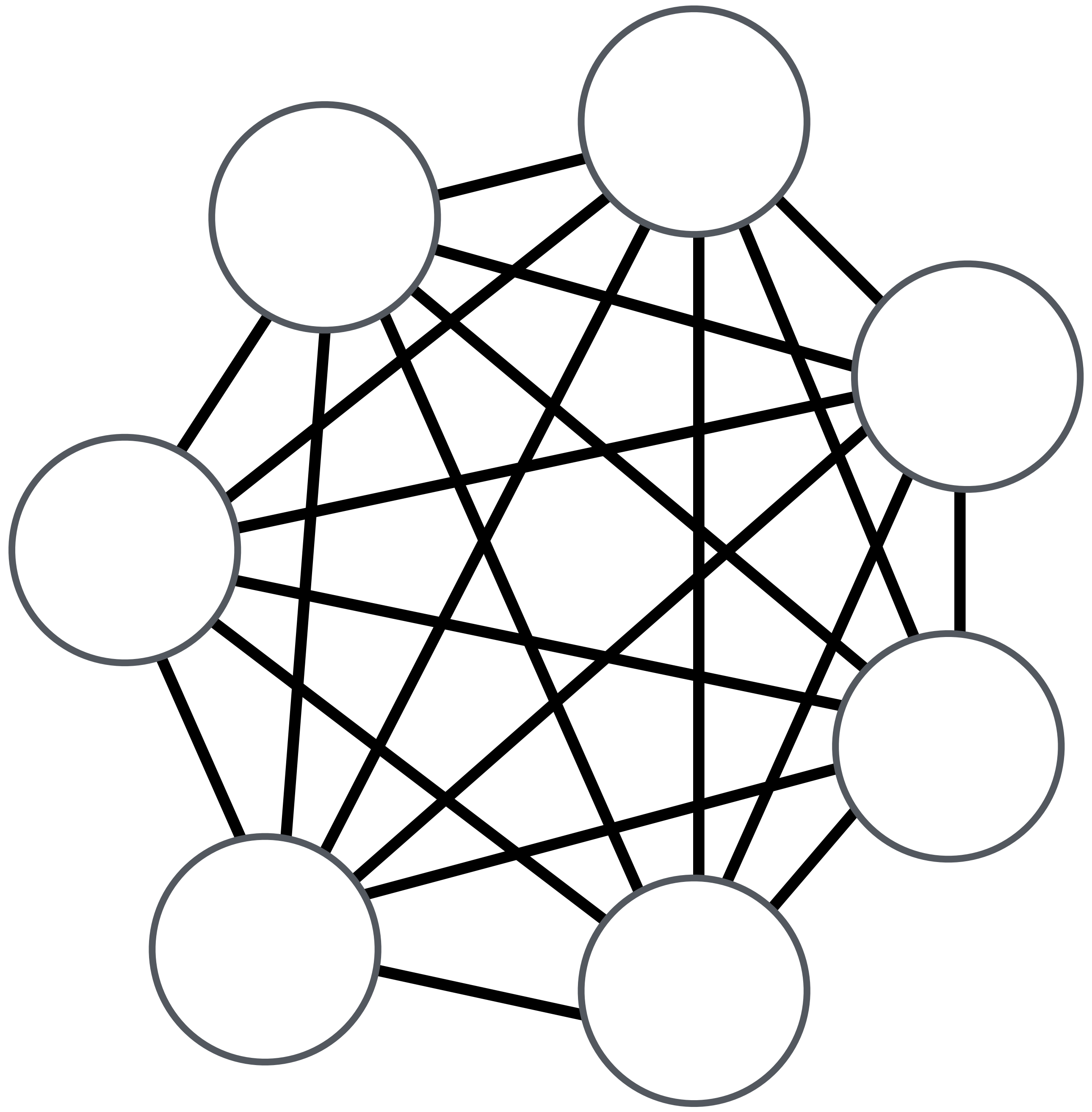


Sparse Regular System

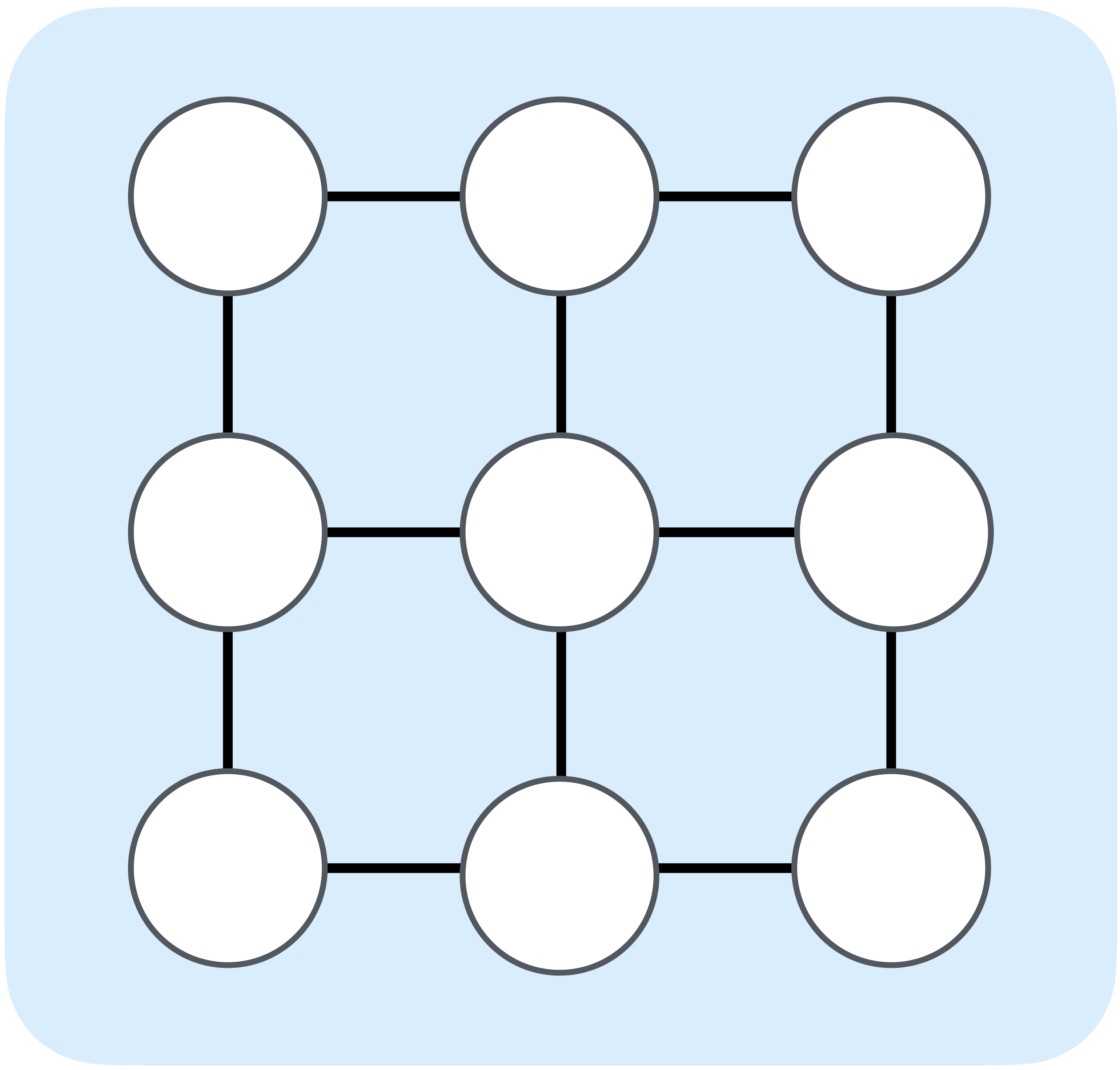


Sparse Irregular System

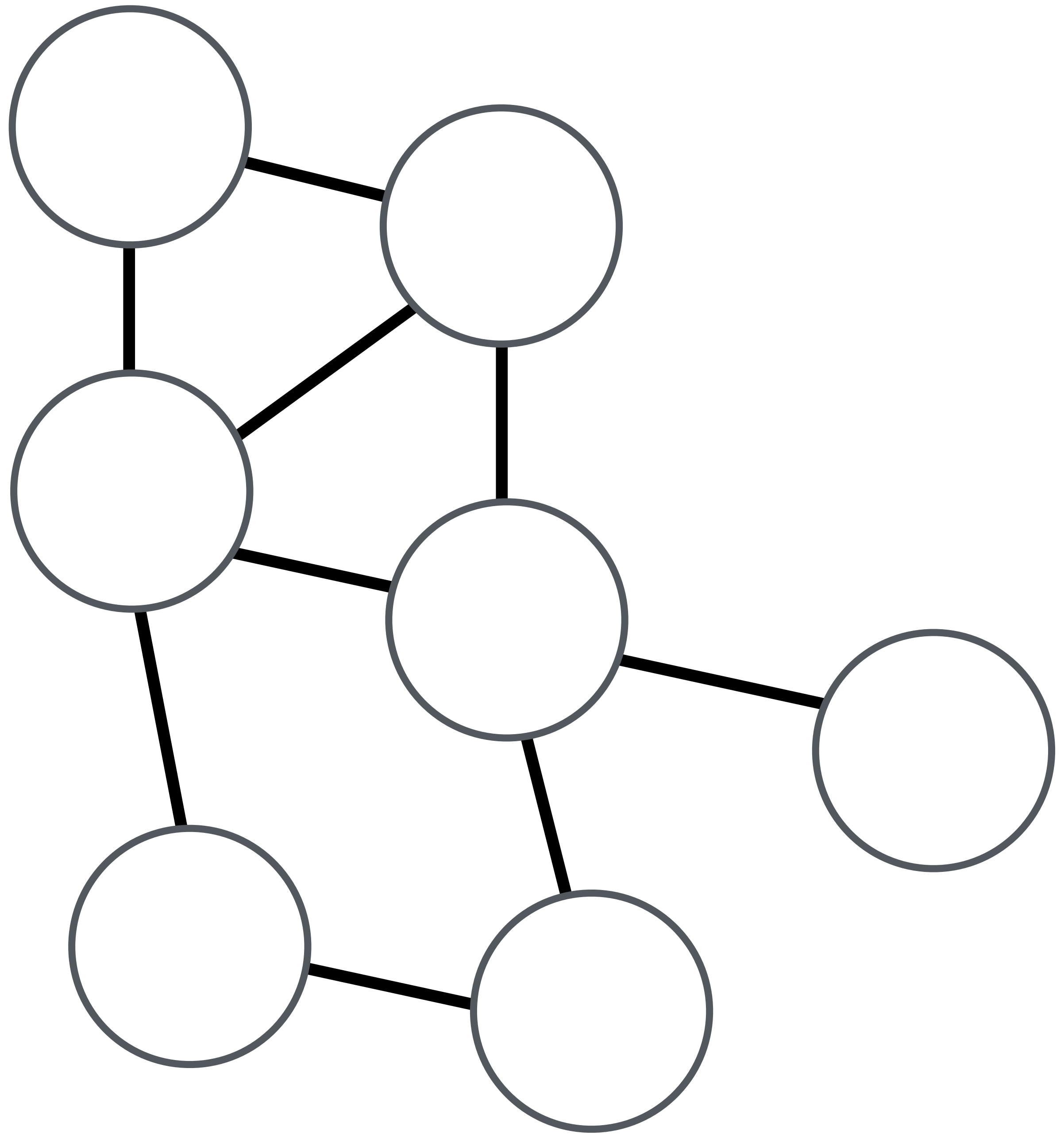
Sparsity and Irregularity of Systems



Dense System

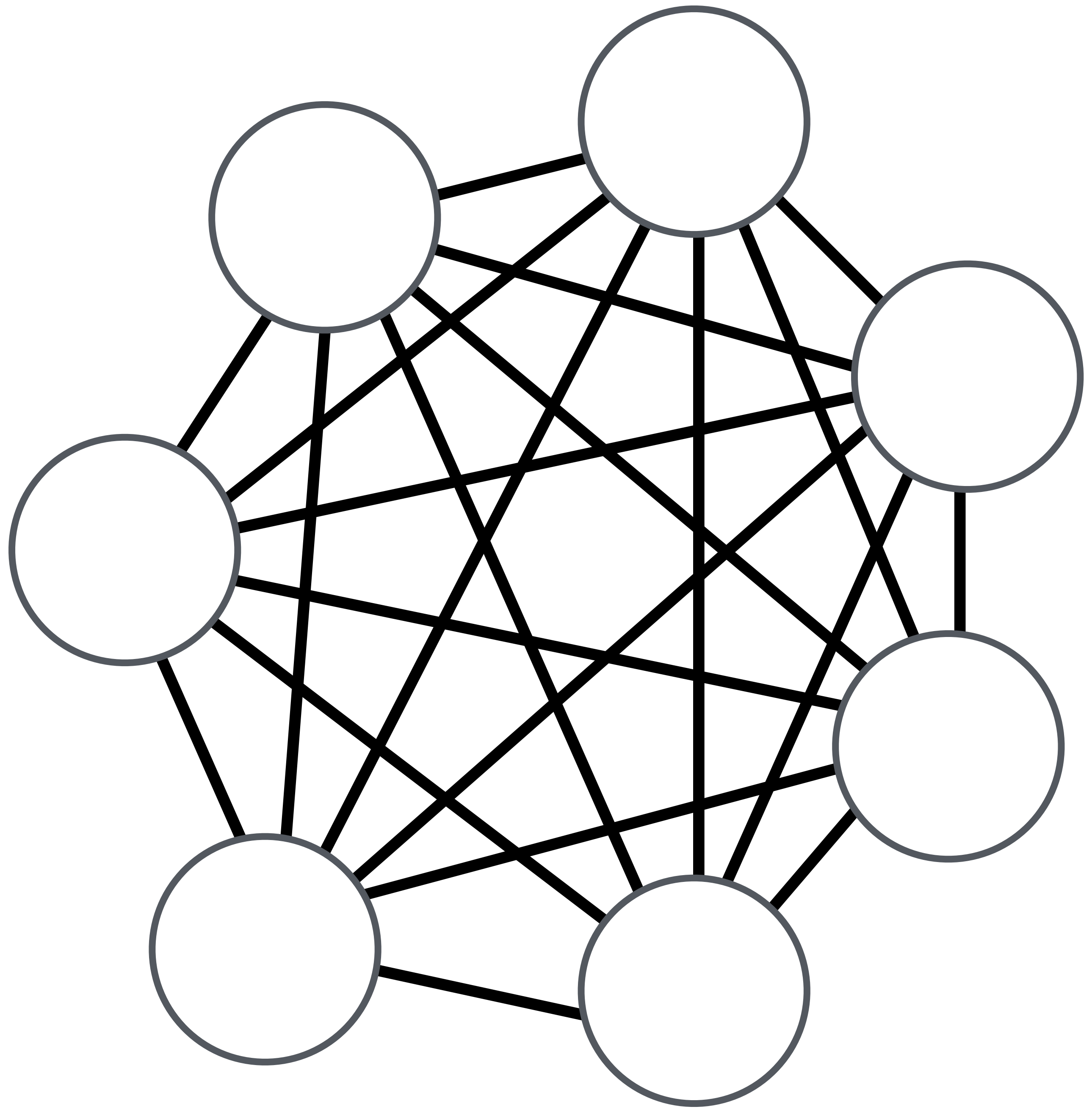


Sparse Regular System

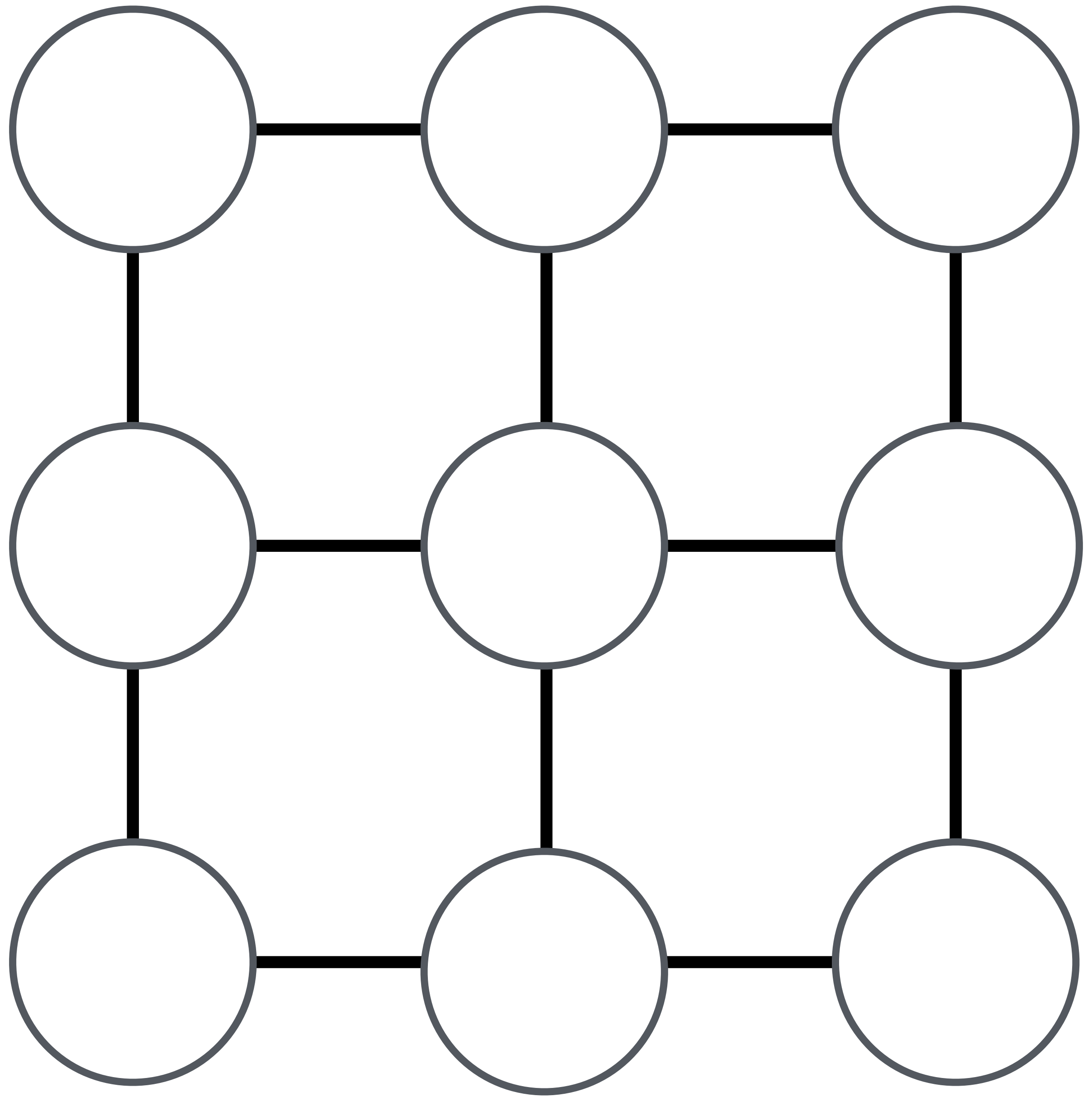


Sparse Irregular System

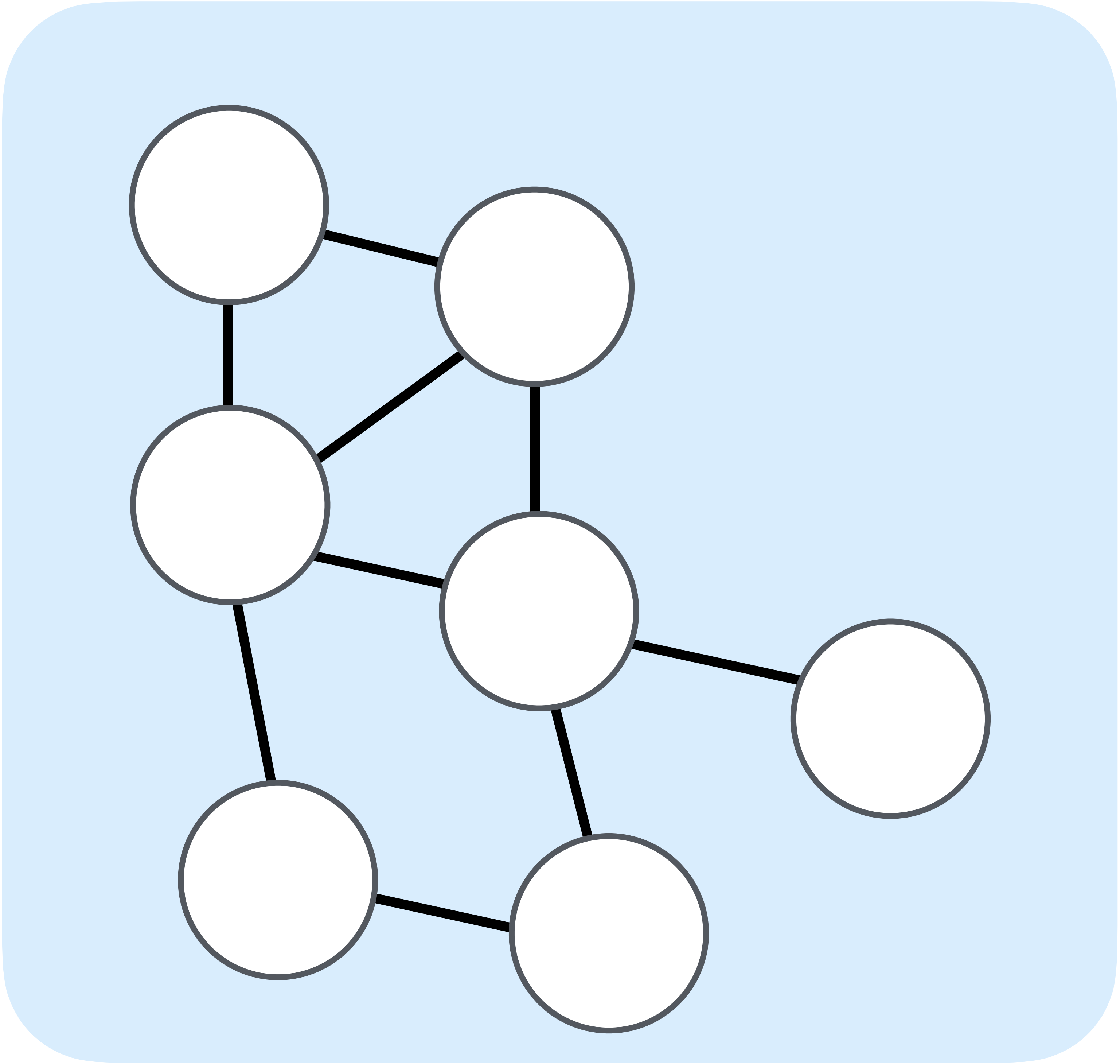
Sparsity and Irregularity of Systems



Dense System



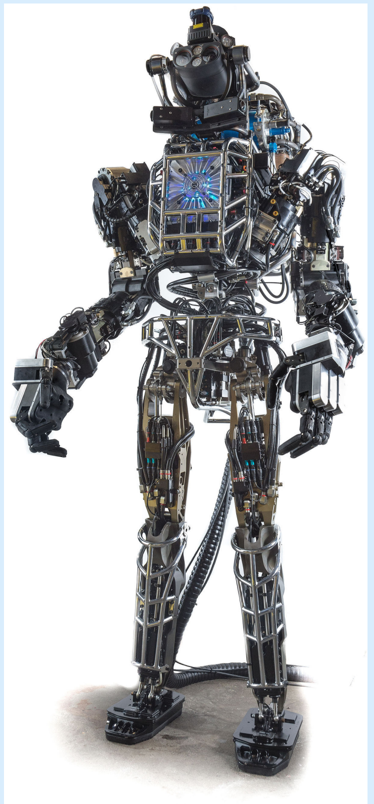
Sparse Regular System



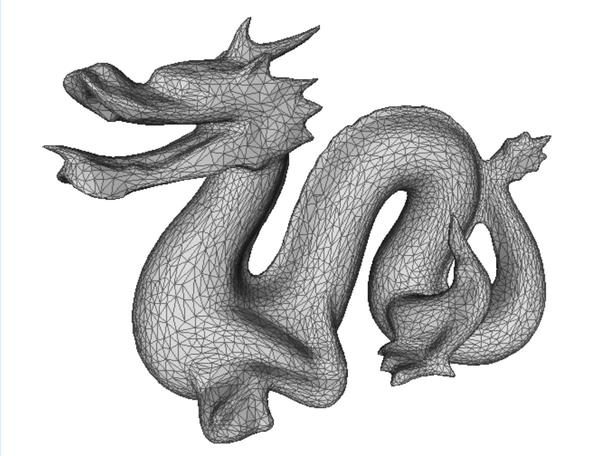
Sparse Irregular System

Sparse Irregular Computing Applications

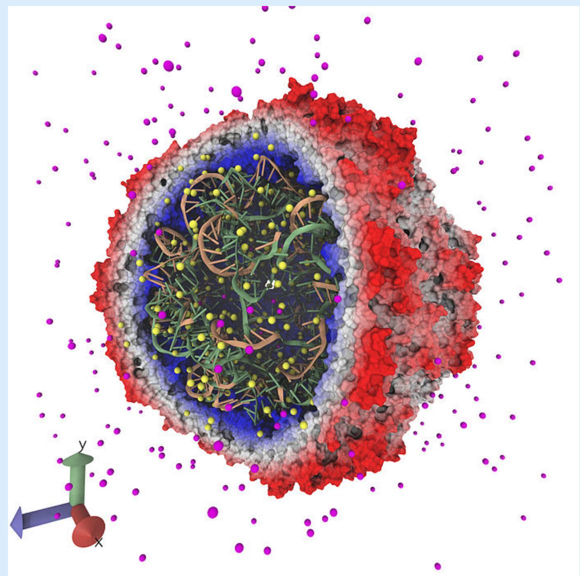
Simulation and Optimization



Robotics



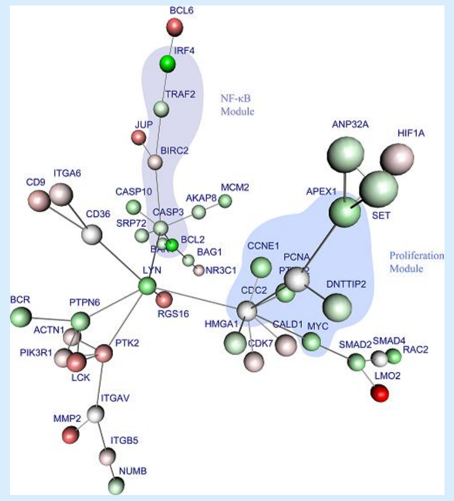
FEM Simulations



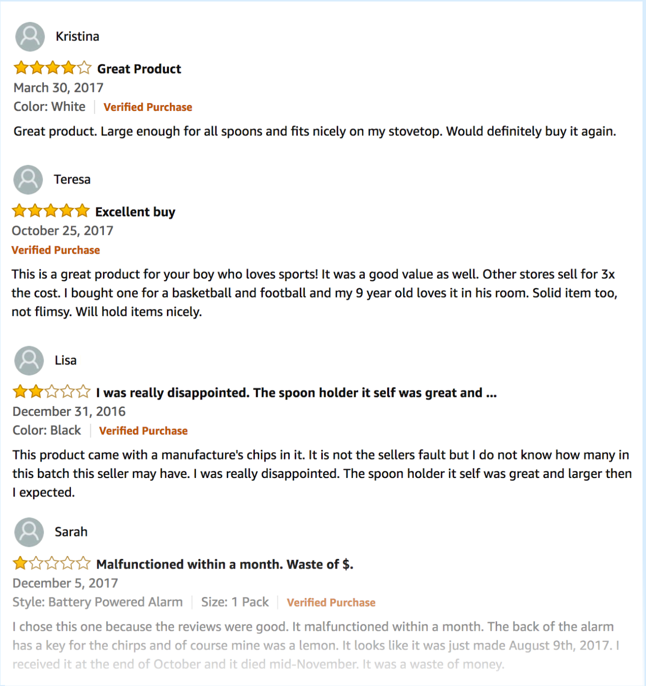
Molecular Dynamics

Data Analytics

Social Networks



Computational Biology



Kristina
Great Product
March 30, 2017
Color: White | Verified Purchase
Great product. Large enough for all spoons and fits nicely on my stovetop. Would definitely buy it again.

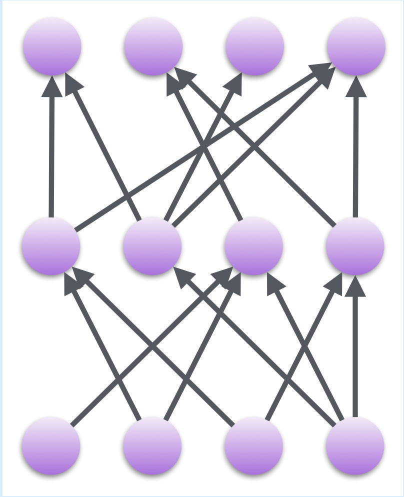
Teresa
Excellent buy
October 25, 2017
Verified Purchase
This is a great product for your boy who loves sports! It was a good value as well. Other stores sell for 3x the cost. I bought one for a basketball and football and my 9 year old loves it in his room. Solid item too, not flimsy. Will hold items nicely.

Lisa
I was really disappointed. The spoon holder it self was great and ...
December 31, 2016
Color: Black | Verified Purchase
This product came with a manufacture's chips in it. It is not the sellers fault but I do not know how many in this batch this seller may have. I was really disappointed. The spoon holder it self was great and larger then I expected.

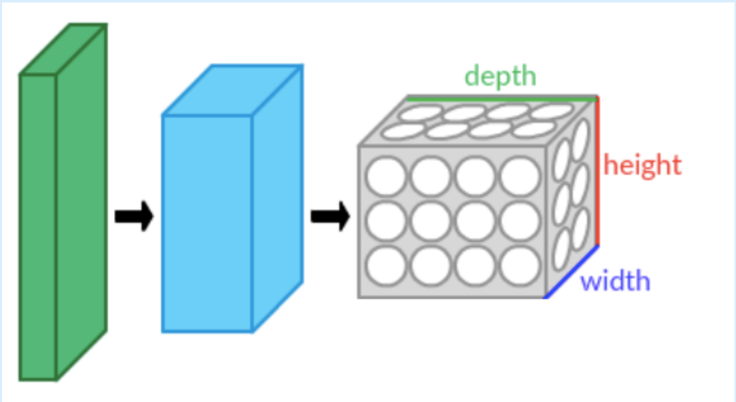
Sarah
Malfunctioned within a month. Waste of $.
December 5, 2017
Style: Battery Powered Alarm | Size: 1 Pack | Verified Purchase
I chose this one because the reviews were good. It malfunctioned within a month. The back of the alarm has a key for the chirps and of course mine was a lemon. It looks like it was just made August 9th, 2017. I received it at the end of October and it died mid-November. It was a waste of money.

Product Reviews

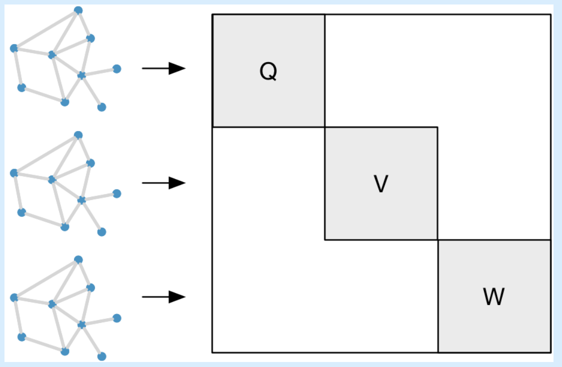
Machine Learning



Sparse Networks



Sparse Convolutional Networks



Graph Convolutional Network

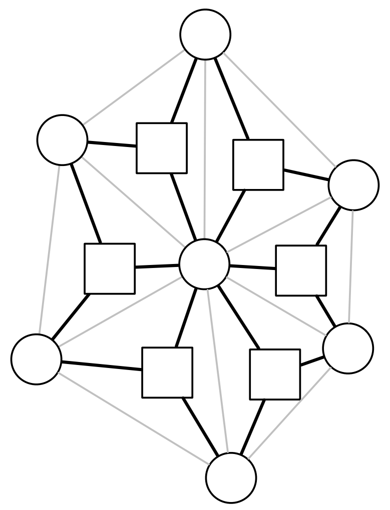
Three Mathematical Languages

Relations

Names	City	Age
Peter	Boston	54
Mary	San Fransisco	35
Paul	New York	23
Adam	Seattle	84
Hilde	Boston	19
Bob	Chicago	76
Sam	Portland	32
Angela	Los Angeles	62

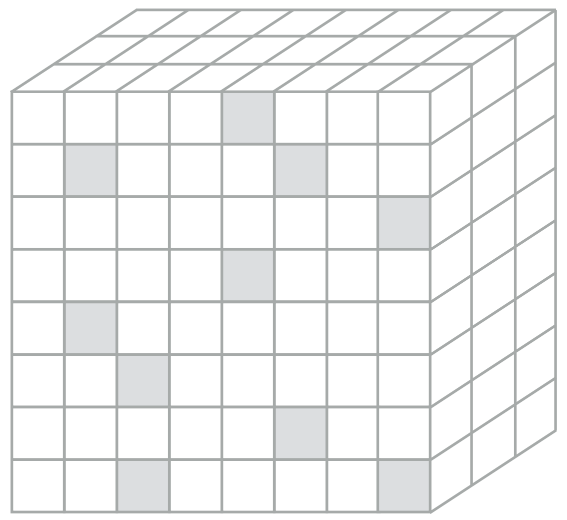
Ideal for combining data to form systems

Graphs



Ideal for local operations

Tensors



Ideal for global operations

Too many implementation variants to write by hand

$a = Bc$

$a = Bc + a$

$a = Bc + b$ $\quad A = B + C$ $\quad a = \alpha Bc + \beta a$

$a = B^T c$ $\quad A = \alpha B$ $\quad a = B(c + d)$

$a = B^T c + d$ $\quad A = B + C + D$ $\quad A = BC$

$A = B \odot C$ $\quad a = b \odot c$ $\quad A = 0$ $\quad A = B \odot (CD)$

$A = BCd$ $\quad A = B^T$ $\quad a = B^T Bc$

$a = b + c$ $\quad A = B$ $\quad K = A^T CA$

$$A_{ij} = \sum_{kl} B_{ikl} C_{lj} D_{kj} \qquad A_{kj} = \sum_{il} B_{ikl} C_{lj} D_{ij}$$

$$A_{lj} = \sum_{ik} B_{ikl} C_{ij} D_{kj} \qquad A_{ij} = \sum_{k} B_{ijk} c_k$$

$$A_{ijk} = \sum_{l} B_{ikl} C_{lj} \qquad A_{ik} = \sum_{j} B_{ijk} c_j$$

$$A_{jk} = \sum_{i} B_{ijk} c_i \qquad A_{ijl} = \sum_{k} B_{ikl} C_{kj}$$

$$\tau = \sum_i z_i (\sum_j z_j \theta_{ij})(\sum_k z_k \theta_{ik})$$

$$C = \sum_{ijkl} M_{ij} P_{jk} \overline{M_{lk}} \, \overline{P_{il}}$$

$$a = \sum_{ijklmnop} M_{ij} P_{jk} M_{kl} P_{lm} \overline{M_{nm}} P_{no} \overline{M_{po}} \, \overline{P_{ip}}$$

Too many implementation variants to write by hand

$a = Bc$

$a = Bc + a$

$a = Bc + b$ $\quad A = B + C$ $\quad a = \alpha Bc + \beta a$

$a = B^T c$ $\quad A = \alpha B$ $\quad a = B(c + d)$

$a = B^T c + d$ $\quad A = B + C + D$ $\quad A = BC$

$A = B \odot C$ $\quad a = b \odot c$ $\quad A = 0$ $\quad A = B \odot (CD)$

$A = BCd$ $\quad A = B^T$ $\quad a = B^T Bc$

$a = b + c$ $\quad A = B$ $\quad K = A^T CA$

Linear Algebra

$$A_{ij} = \sum_{kl} B_{ikl} C_{lj} D_{kj} \qquad A_{kj} = \sum_{il} B_{ikl} C_{lj} D_{ij}$$

$$A_{lj} = \sum_{ik} B_{ikl} C_{ij} D_{kj} \qquad A_{ij} = \sum_{k} B_{ijk} c_k$$

$$A_{ijk} = \sum_{l} B_{ikl} C_{lj} \qquad A_{ik} = \sum_{j} B_{ijk} c_j$$

$$A_{jk} = \sum_{i} B_{ijk} c_i \qquad A_{ijl} = \sum_{k} B_{ikl} C_{kj}$$

$$\tau = \sum_{i} z_i (\sum_{j} z_j \theta_{ij}) (\sum_{k} z_k \theta_{ik})$$

$$C = \sum_{ijkl} M_{ij} P_{jk} \overline{M_{lk}}\, \overline{P_{il}}$$

$$a = \sum_{ijklmnop} M_{ij} P_{jk} M_{kl} P_{lm} \overline{M_{nm}} P_{no} \overline{M_{po}}\, \overline{P_{ip}}$$

Too many implementation variants to write by hand

$a = Bc$

$a = Bc + a$

$a = Bc + b$ $\quad A = B + C$ $\quad a = \alpha Bc + \beta a$

$a = B^T c$ $\quad A = \alpha B$ $\quad a = B(c + d)$

$a = B^T c + d$ $\quad A = B + C + D$ $\quad A = BC$

$A = B \odot C$ $\quad a = b \odot c$ $\quad A = 0$ $\quad A = B \odot (CD)$

$A = BCd$ $\quad A = B^T$ $\quad a = B^T Bc$

$a = b + c$ $\quad A = B$ $\quad K = A^T CA$

$$A_{ij} = \sum_{kl} B_{ikl} C_{lj} D_{kj} \qquad A_{kj} = \sum_{il} B_{ikl} C_{lj} D_{ij}$$

$$A_{lj} = \sum_{ik} B_{ikl} C_{ij} D_{kj} \qquad A_{ij} = \sum_{k} B_{ijk} c_k$$

$$A_{ijk} = \sum_{l} B_{ikl} C_{lj} \qquad A_{ik} = \sum_{j} B_{ijk} c_j$$

$$A_{jk} = \sum_{i} B_{ijk} c_i \qquad A_{ijl} = \sum_{k} B_{ikl} C_{kj}$$

Data analytics
(tensor factorization)

$$\tau = \sum_i z_i (\sum_j z_j \theta_{ij})(\sum_k z_k \theta_{ik})$$

$$C = \sum_{ijkl} M_{ij} P_{jk} \overline{M_{lk}}\, \overline{P_{il}}$$

$$a = \sum_{ijklmnop} M_{ij} P_{jk} M_{kl} P_{lm} \overline{M_{nm}} P_{no} \overline{M_{po}}\, \overline{P_{ip}}$$

Too many implementation variants to write by hand

$a = Bc$

$a = Bc + a$

$a = Bc + b$ $\quad A = B + C$ $\quad a = \alpha Bc + \beta a$

$a = B^T c$ $\quad A = \alpha B$ $\quad a = B(c + d)$

$a = B^T c + d$ $\quad A = B + C + D$ $\quad A = BC$

$A = B \odot C$ $\quad a = b \odot c$ $\quad A = 0$ $\quad A = B \odot (CD)$

$A = BCd$ $\quad A = B^T$ $\quad a = B^T Bc$

$a = b + c$ $\quad A = B$ $\quad K = A^T CA$

$$A_{ij} = \sum_{kl} B_{ikl} C_{lj} D_{kj} \qquad A_{kj} = \sum_{il} B_{ikl} C_{lj} D_{ij}$$

$$A_{lj} = \sum_{ik} B_{ikl} C_{ij} D_{kj} \qquad A_{ij} = \sum_{k} B_{ijk} c_k$$

$$A_{ijk} = \sum_{l} B_{ikl} C_{lj} \qquad A_{ik} = \sum_{j} B_{ijk} c_j$$

$$A_{jk} = \sum_{i} B_{ijk} c_i \qquad A_{ijl} = \sum_{k} B_{ikl} C_{kj}$$

$$\tau = \sum_{i} z_i (\sum_{j} z_j \theta_{ij})(\sum_{k} z_k \theta_{ik})$$

$$C = \sum_{ijkl} M_{ij} P_{jk} \overline{M_{lk}}\, \overline{P_{il}}$$

$$a = \sum_{ijklmnop} M_{ij} P_{jk} M_{kl} P_{lm} \overline{M_{nm}} P_{no} \overline{M_{po}}\, \overline{P_{ip}}$$

Quantum Chromodynamics

Too many implementation variants to write by hand

$a = Bc$

$a = Bc + a$

$a = Bc + b$ $\quad A = B + C$ $\quad a = \alpha Bc + \beta a$

$a = B^T c$ $\quad A = \alpha B$ $\quad a = B(c + d)$

$a = B^T c + d$ $\quad A = B + C + D$ $\quad A = BC$

$A = B \odot C$ $\quad a = b \odot c$ $\quad A = 0$ $\quad A = B \odot (CD)$

$A = BCd$ $\quad A = B^T$ $\quad a = B^T Bc$

$a = b + c$ $\quad A = B$ $\quad K = A^T CA$

$A_{ij} = \sum_{kl} B_{ikl} C_{lj} D_{kj}$ $\quad A_{kj} = \sum_{il} B_{ikl} C_{lj} D_{ij}$

$A_{lj} = \sum_{ik} B_{ikl} C_{ij} D_{kj}$ $\quad A_{ij} = \sum_{k} B_{ijk} c_k$

$A_{ijk} = \sum_{l} B_{ikl} C_{lj}$ $\quad A_{ik} = \sum_{j} B_{ijk} c_j$

$A_{jk} = \sum_{i} B_{ijk} c_i$ $\quad A_{ijl} = \sum_{k} B_{ikl} C_{kj}$

$\tau = \sum_{i} z_i (\sum_{j} z_j \theta_{ij})(\sum_{k} z_k \theta_{ik})$

$C = \sum_{ijkl} M_{ij} P_{jk} \overline{M_{lk}}\, \overline{P_{il}}$

$a = \sum_{ijklmnop} M_{ij} P_{jk} M_{kl} P_{lm} \overline{M_{nm}} P_{no} \overline{M_{po}}\, \overline{P_{ip}}$

$\times$

Dense Matrix

CSR DCSR BCSR

COO ELLPACK CSB

Blocked COO CSC

DIA Blocked DIA DCSC

Sparse vector Hash Maps

Coordinates

Dense Tensors

CSF

Blocked Tensors

Linked Lists Database

Compression Schemes

Cloud Storage

Too many implementation variants to write by hand

$a = Bc$
$a = Bc + a$
$a = Bc + b$ $A = B + C$ $a = \alpha Bc + \beta a$
$a = B^T c$ $A = \alpha B$ $a = B(c + d)$
$a = B^T c + d$ $A = B + C + D$ $A = BC$
$A = B \odot C$ $a = b \odot c$ $A = 0$ $A = B \odot (CD)$
$A = BCd$ $A = B^T$ $a = B^T Bc$
$a = b + c$ $A = B$ $K = A^T CA$

$$A_{ij} = \sum_{kl} B_{ikl} C_{lj} D_{kj} \quad A_{kj} = \sum_{il} B_{ikl} C_{lj} D_{ij}$$

$$A_{lj} = \sum_{ik} B_{ikl} C_{ij} D_{kj} \quad A_{ij} = \sum_{k} B_{ijk} c_k$$

$$A_{ijk} = \sum_{l} B_{ikl} C_{lj} \quad A_{ik} = \sum_{j} B_{ijk} c_j$$

$$A_{jk} = \sum_{i} B_{ijk} c_i \quad A_{ijl} = \sum_{k} B_{ikl} C_{kj}$$

$$\tau = \sum_{i} z_i (\sum_{j} z_j \theta_{ij})(\sum_{k} z_k \theta_{ik})$$

$$C = \sum_{ijkl} M_{ij} P_{jk} \overline{M_{lk}}\, \overline{P_{il}}$$

$$a = \sum_{ijklmnop} M_{ij} P_{jk} M_{kl} P_{lm} \overline{M_{nm}} P_{no} \overline{M_{po}}\, \overline{P_{ip}}$$

$\times$

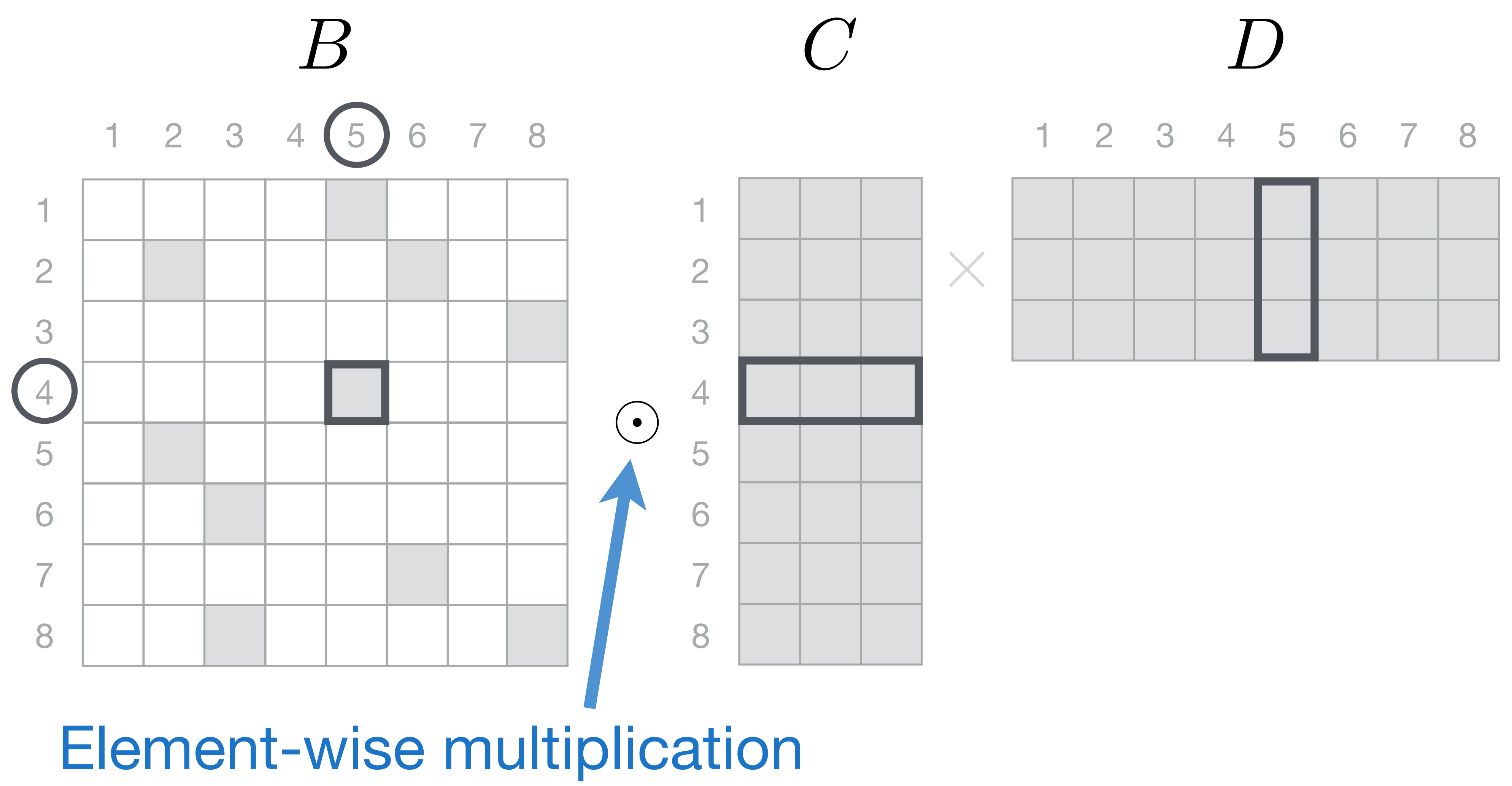
Dense Matrix
CSR DCSR BCSR
COO ELLPACK CSB
Blocked COO CSC
DIA Blocked DIA DCSC
Sparse vector Hash Maps
Coordinates
Dense Tensors
CSF
Blocked Tensors
Linked Lists Database
Compression Schemes
Cloud Storage

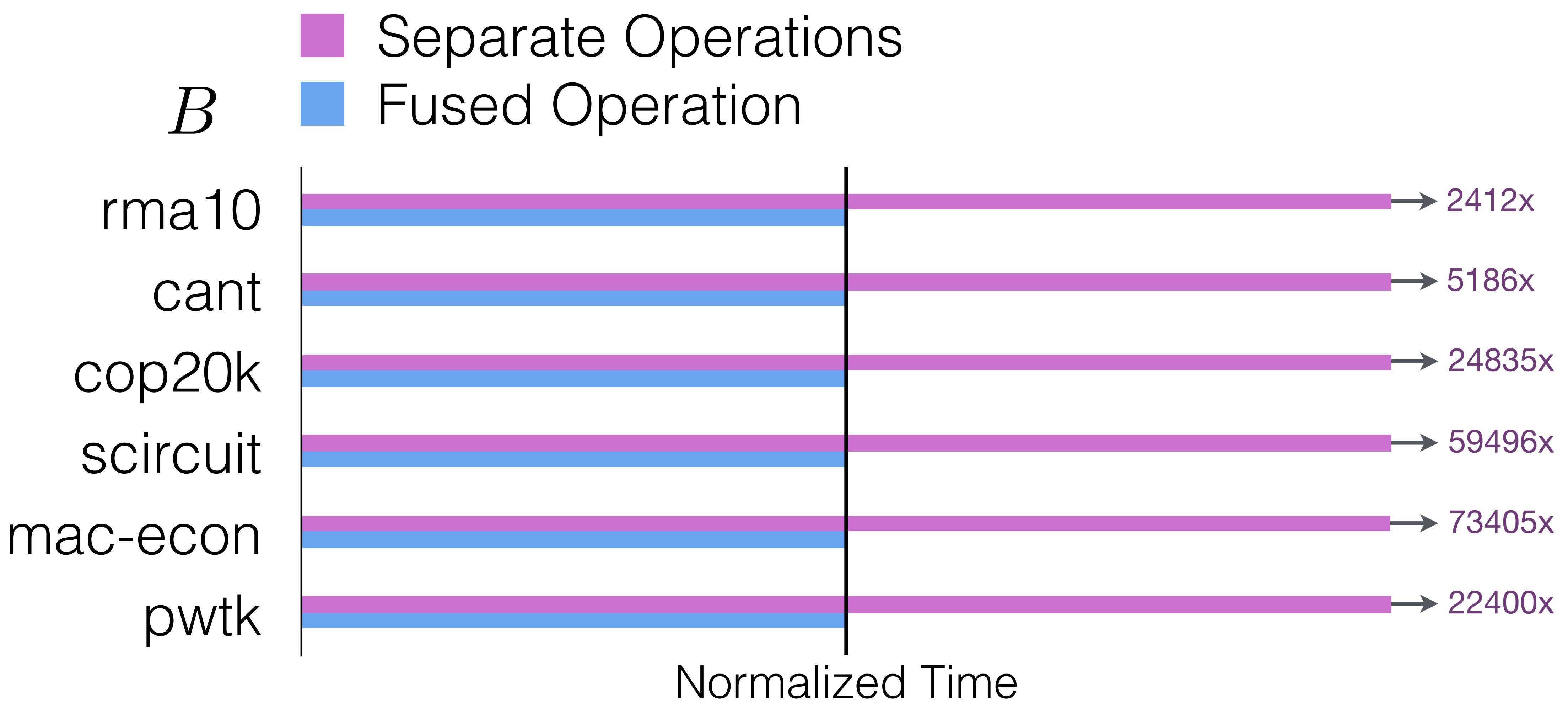
$\times$

CPU
GPUs TPUs
FPGA
Sparse Tensor Hardware
Cloud Computers
Supercomputers

Compound Example 1: Sampled Distance Metrics with Linear Algebra

$$A = B \odot (CD)$$





Compound Example 2: Triangle Counting with Relational Algebra

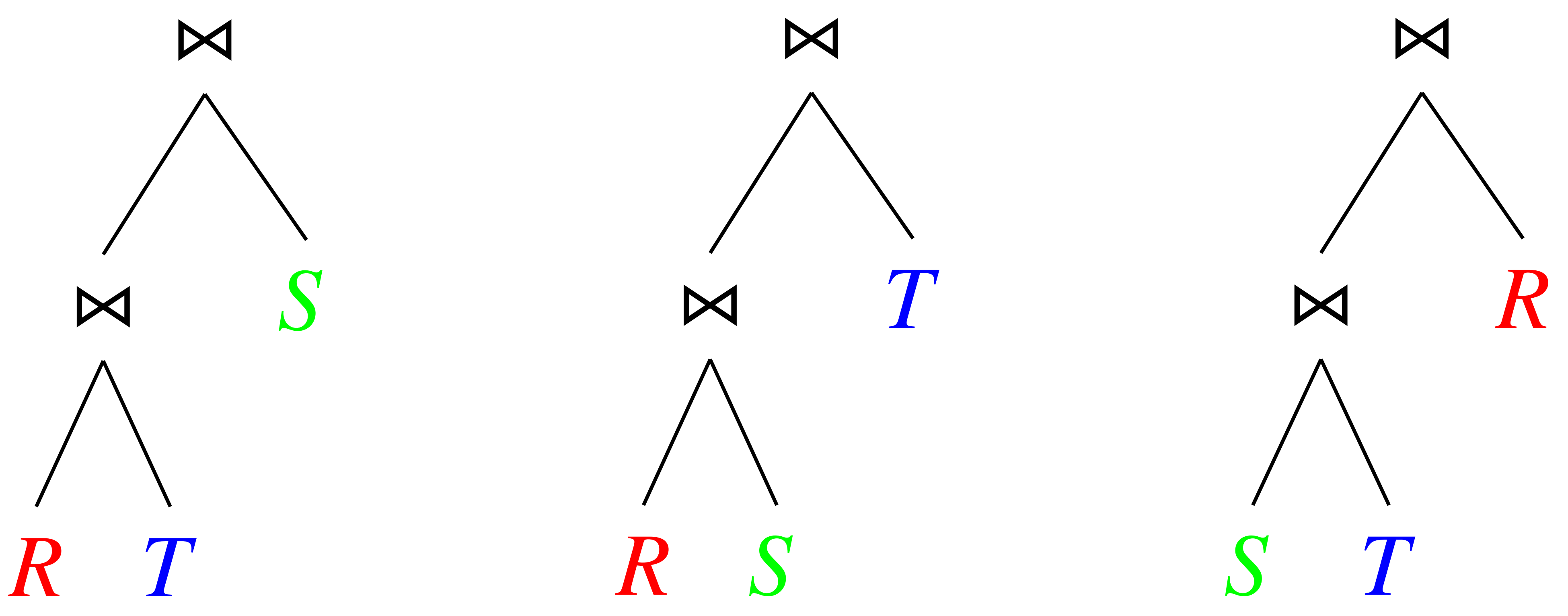
$$Q_{\triangle} = R(A, B) \bowtie S(B, C) \bowtie T(A, C)$$

Compound Example 2: Triangle Counting with Relational Algebra

$$Q_{\triangle} = R(A, B) \bowtie S(B, C) \bowtie T(A, C) \qquad O(N^{2/3})$$

Compound Example 2: Triangle Counting with Relational Algebra

$$Q_{\triangle} = R(A,B) \bowtie S(B,C) \bowtie T(A,C) \qquad O(N^{2/3})$$

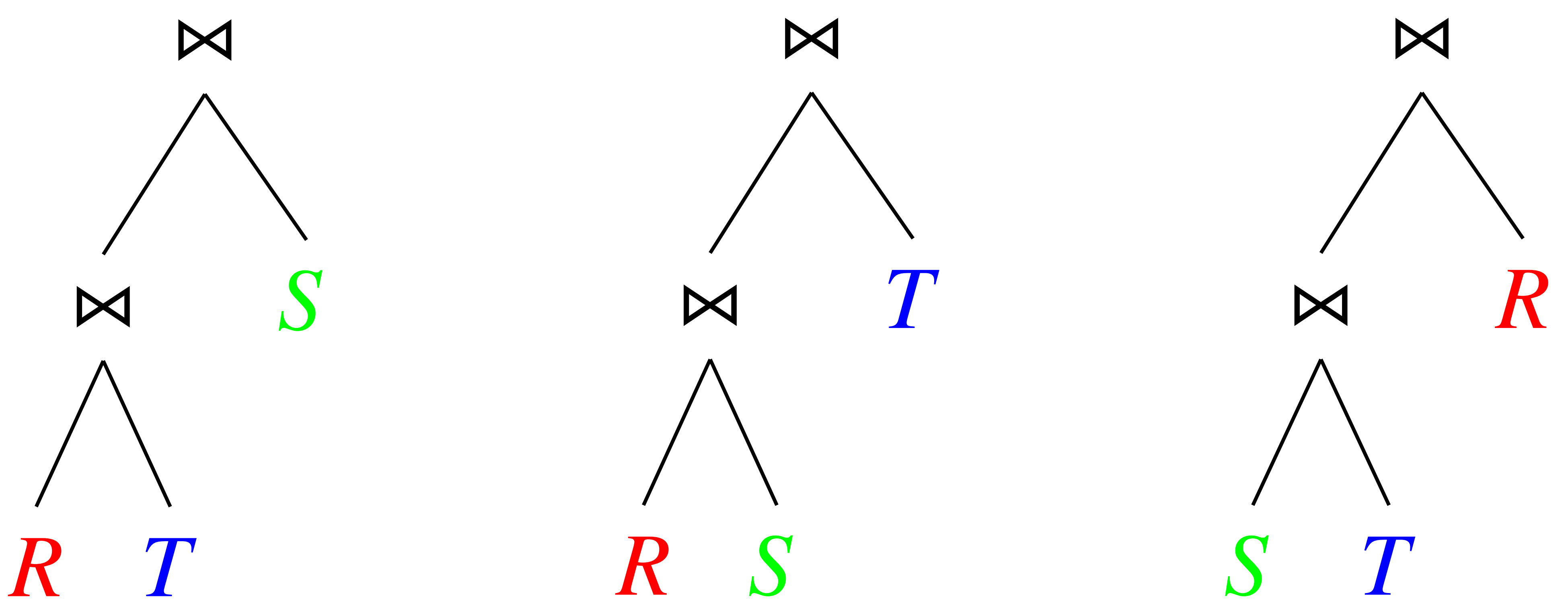


$O(N^2)$

Figures from Ngo, Ré and Rudra (2013),
with algorithm from Veldhuizen (2014)

Compound Example 2: Triangle Counting with Relational Algebra

$$Q_\triangle = R(A,B) \bowtie S(B,C) \bowtie T(A,C) \qquad O(N^{2/3})$$



$O(N^2)$

Algorithm 2 Computing $Q_\triangle$ by delaying computation.

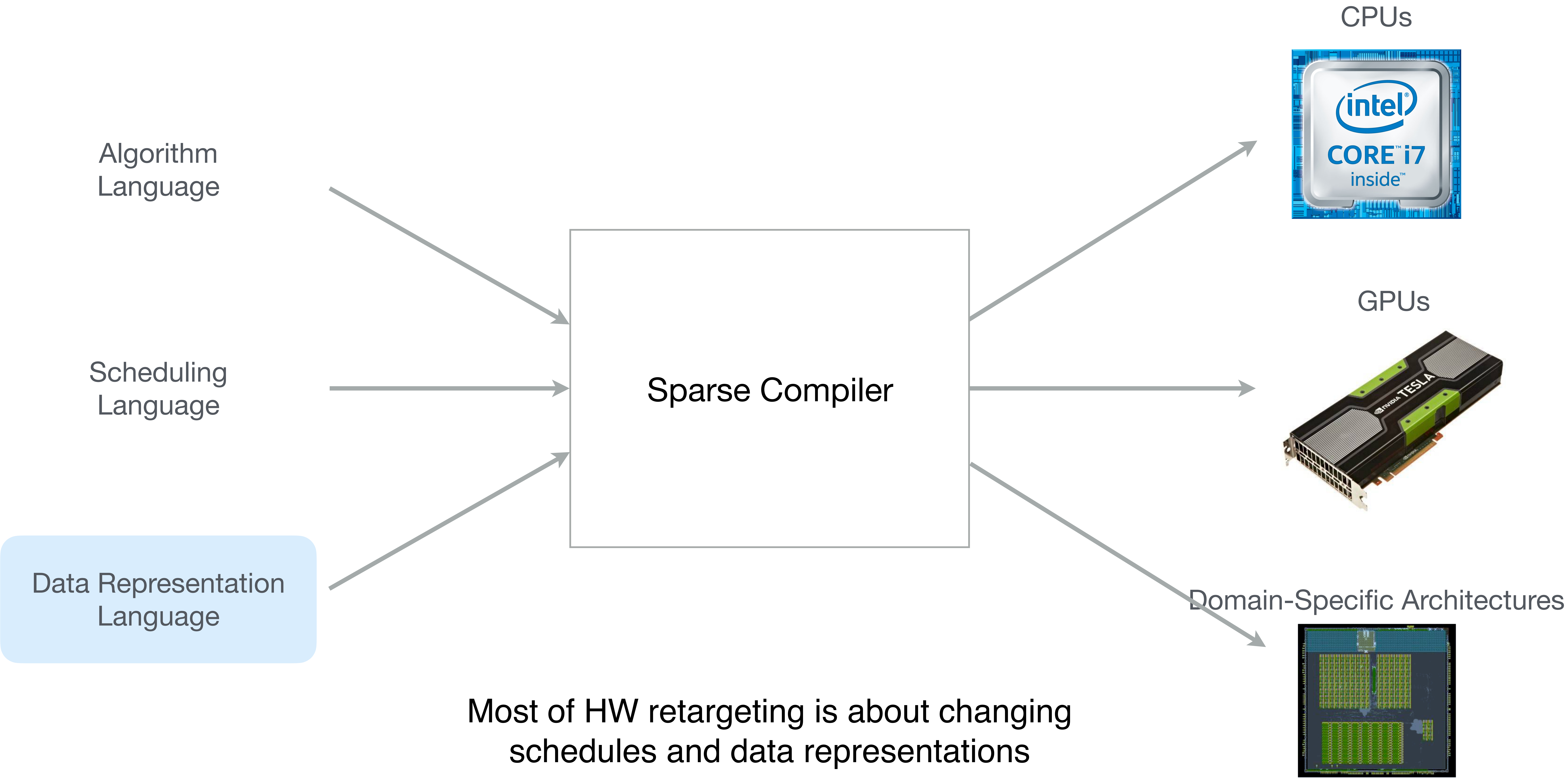
Input: $R(A,B), S(B,C), T(A,C)$ in sorted order

```
Q ← ∅
L_A ← π_A(R) ∩ π_A(T)
For each a ∈ L_A do
    L_B^a ← π_B(σ_{A=a}(R)) ∩ π_B(S)
    For each b ∈ L_B^a do
        L_C^{a,b} ← π_C(σ_{B=b}(S)) ∩ π_C(σ_{A=a}(T))
        For each c ∈ L_C^{a,b} do
            Add (a, b, c) to Q
Return Q
```

$O(N^{2/3})$

Figures from Ngo, Ré and Rudra (2013),
with algorithm from Veldhuizen (2014)

Separation of Algorithm, Schedule, and Data Representation



Most of HW retargeting is about changing schedules and data representations

The Sparse Tensor Algebra Compiler (TACO)

Tensor Algebra Expression Language

Scheduling Language

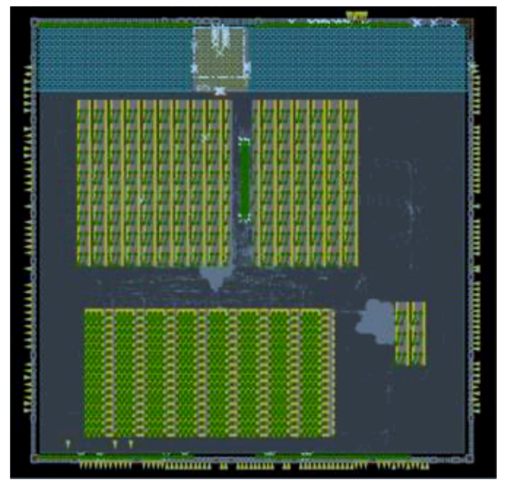
Data Representation Language

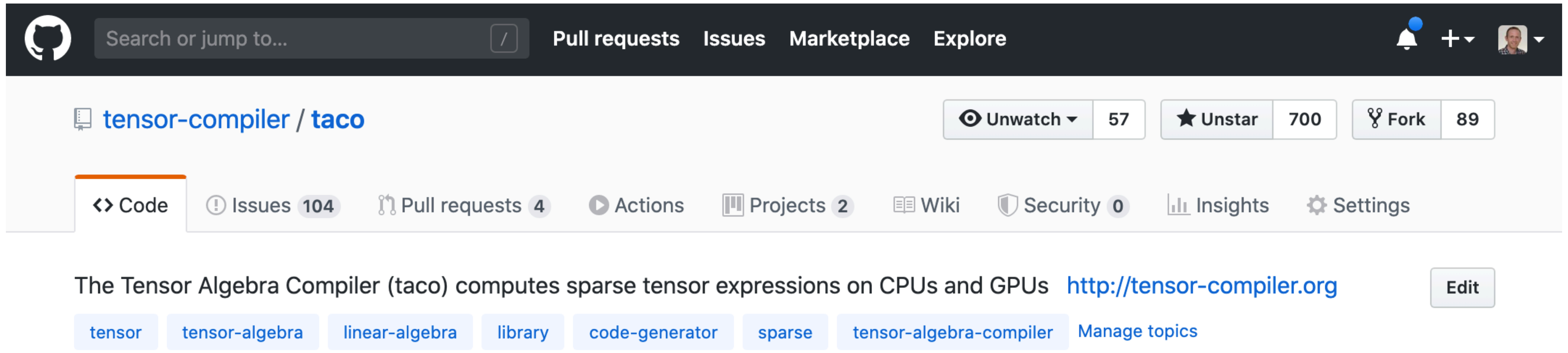
The Sparse Tensor Algebra Compiler (TACO)

CPUs

GPUs

DSAs TBD





Sparse Iteration Theory

$$A_{ij} = \sum_k B_{ik} C_{kj}$$ ← Matrix Multiplication

Sparse Iteration Theory

$$A_{ij} = \sum_k B_{ik} C_{kj}$$ ← Matrix Multiplication

Sparse Iteration Space

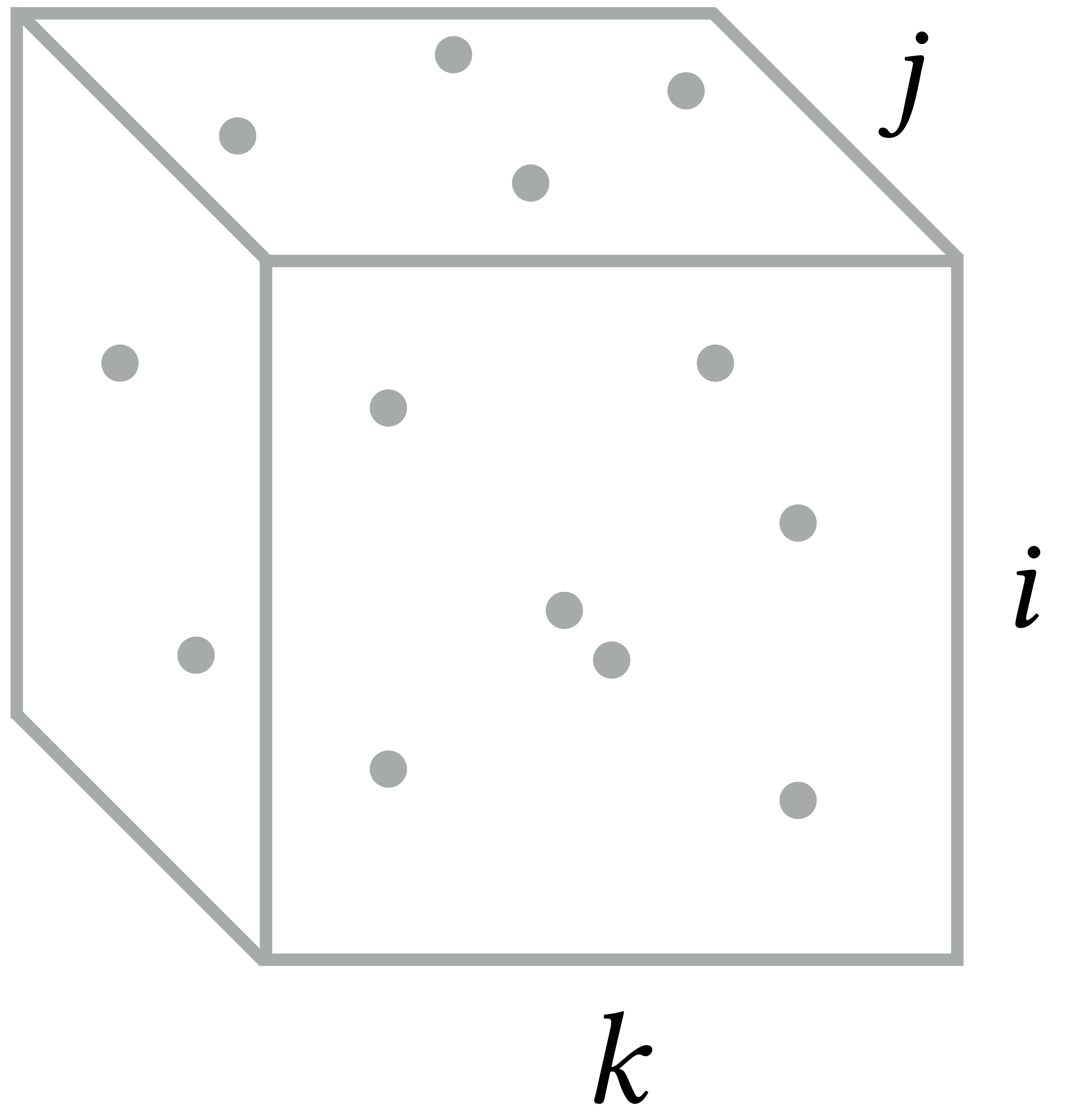
$$B_{ik} \cap C_{kj}$$

Sparse Iteration Theory

$$A_{ij} = \sum_k B_{ik} C_{kj}$$ ← Matrix Multiplication

Sparse Iteration Space

$$B_{ik} \cap C_{kj}$$

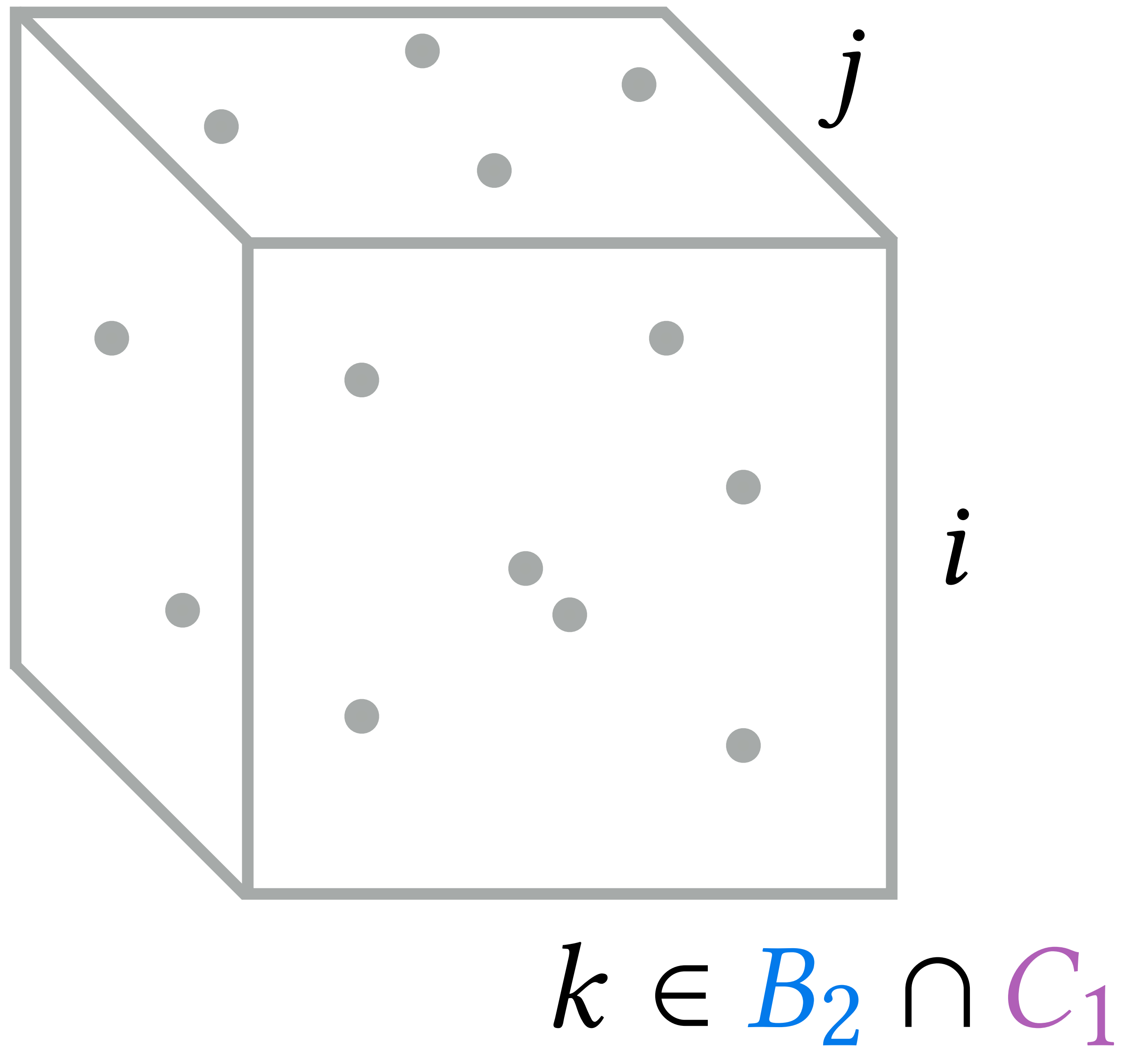


Sparse Iteration Theory

$$A_{ij} = \sum_k B_{ik} C_{kj}$$ ← Matrix Multiplication

Sparse Iteration Space

$$B_{ik} \cap C_{kj}$$

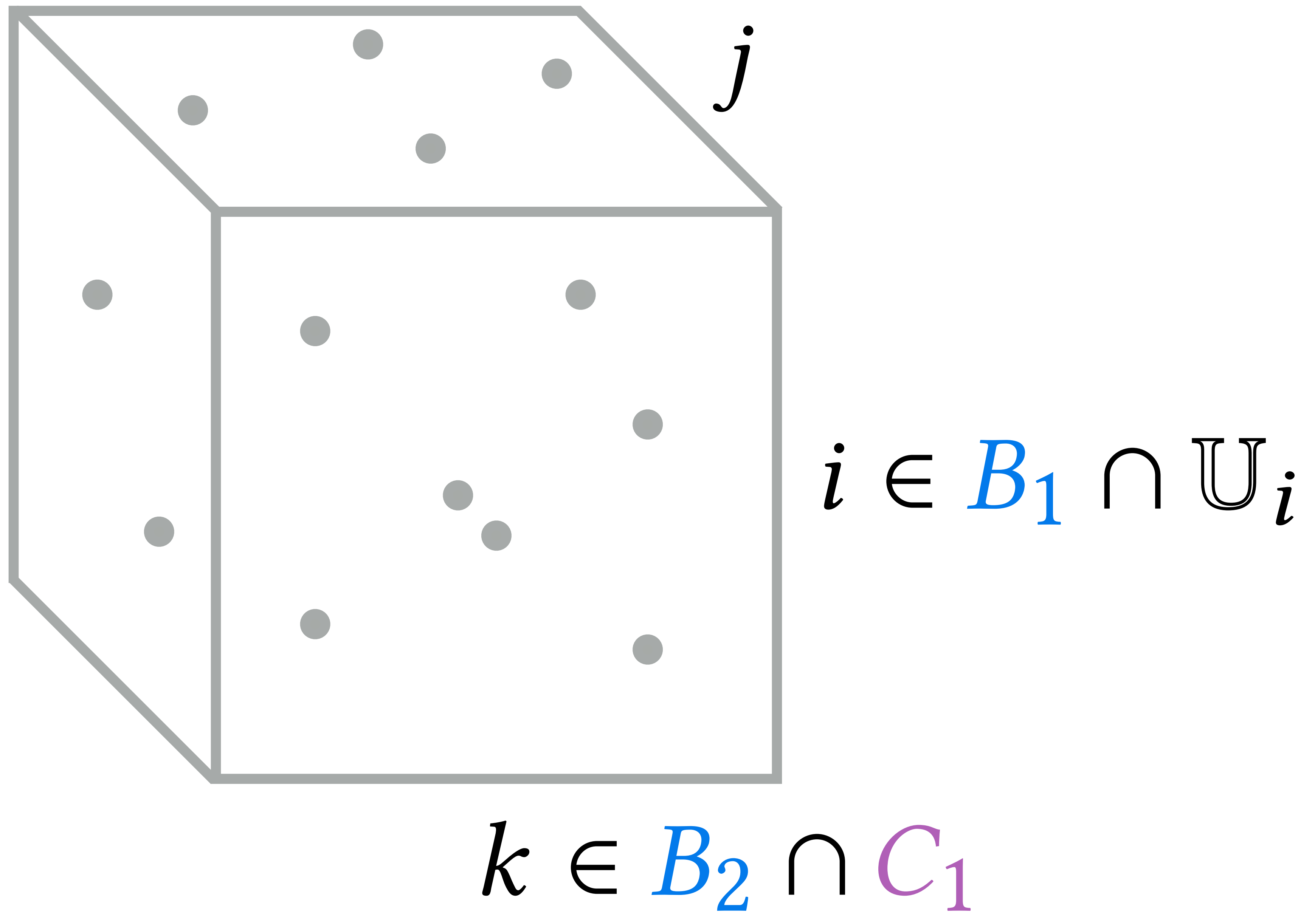


Sparse Iteration Theory

$$A_{ij} = \sum_k B_{ik} C_{kj}$$ ← Matrix Multiplication

Sparse Iteration Space

$$B_{ik} \cap C_{kj}$$



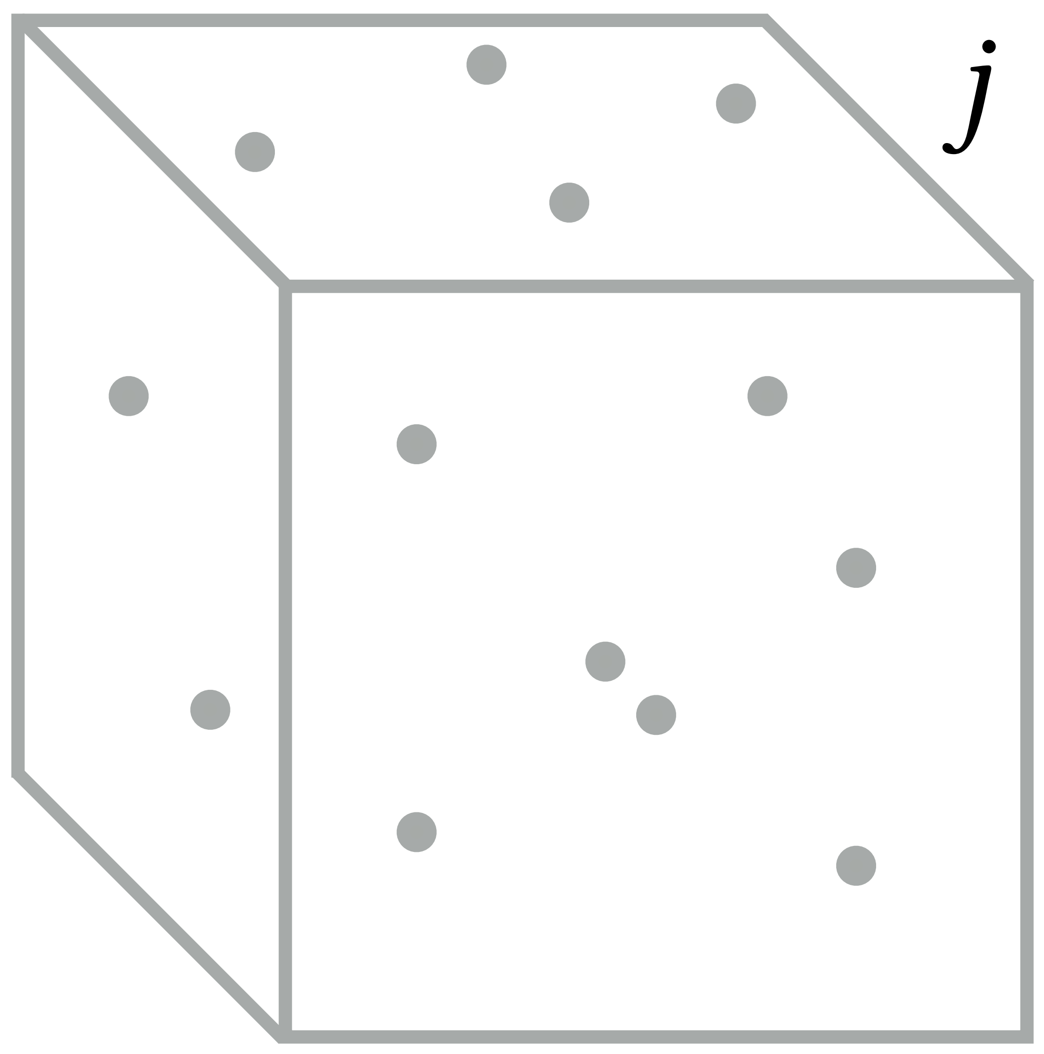
Sparse Iteration Theory

$$A_{ij} = \sum_k B_{ik} C_{kj}$$

Matrix Multiplication

Sparse Iteration Space

$$B_{ik} \cap C_{kj}$$



$j \in \mathbb{U}_j \cap C_2$

$i \in B_1 \cap \mathbb{U}_i$

$k \in B_2 \cap C_1$

Sparse Iteration Theory

$$A_{ij} = \sum_k B_{ik} C_{kj}$$ Matrix Multiplication

Union of Coordinate Trees

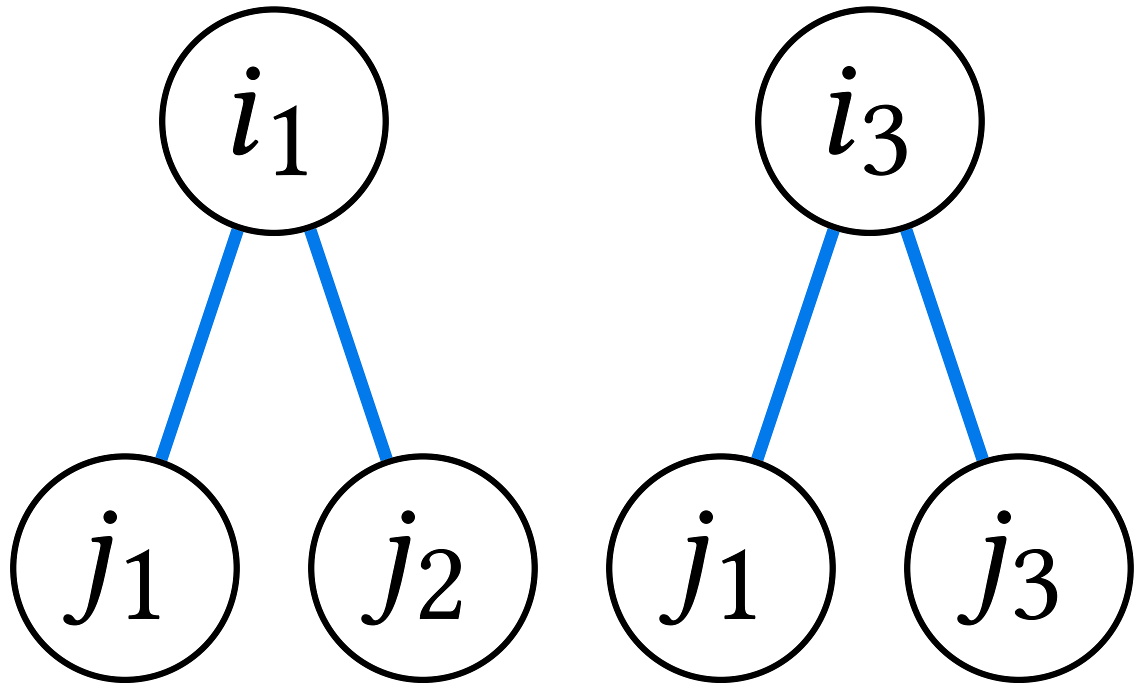
$$B_{ik} \cap C_{kj}$$

Sparse Iteration Theory

$$A_{ij} = \sum_k B_{ik} C_{kj}$$ ← Matrix Multiplication

Union of Coordinate Trees

$$B_{ik} \cap C_{kj}$$

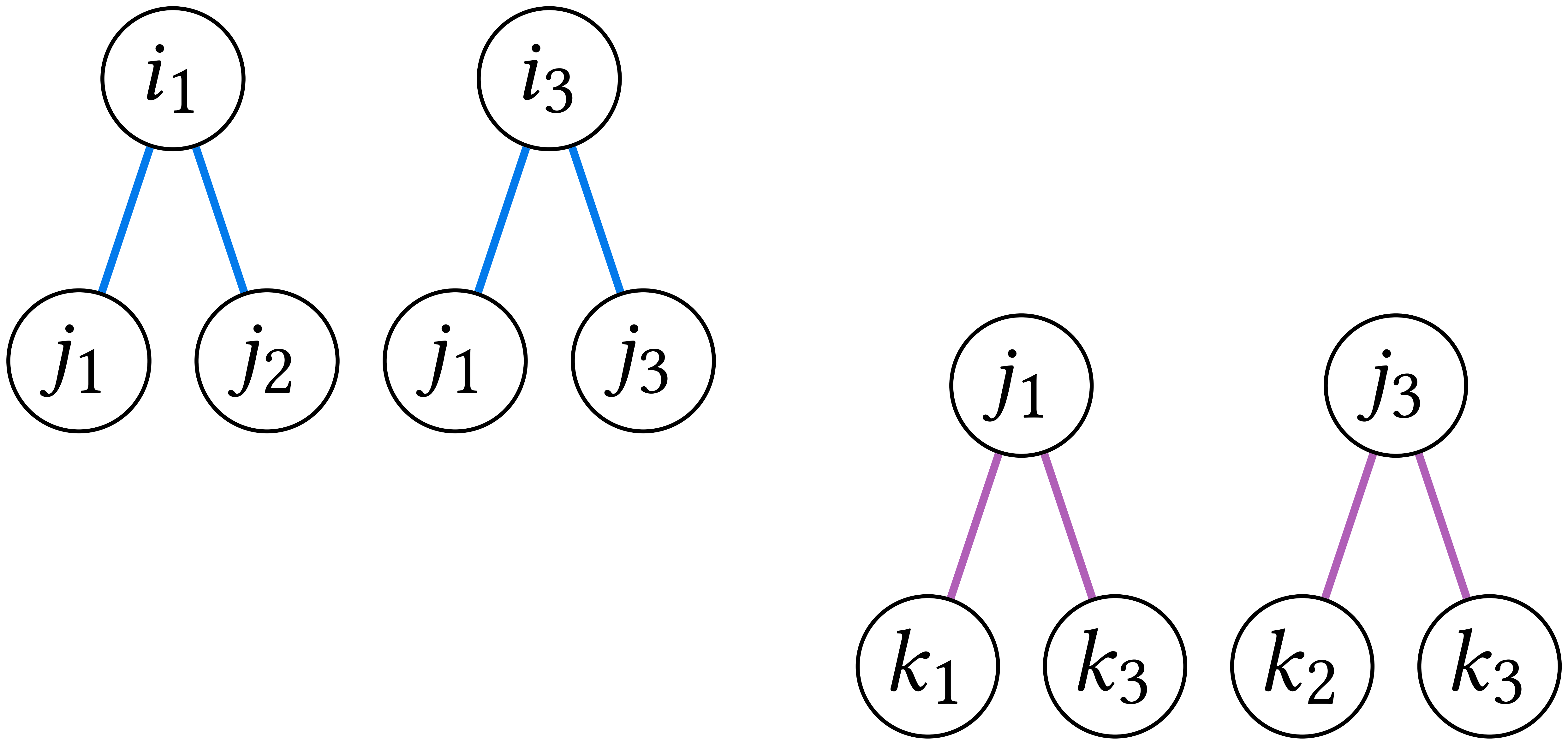


Sparse Iteration Theory

$$A_{ij} = \sum_k B_{ik} C_{kj}$$ Matrix Multiplication

Union of Coordinate Trees

$B_{ik} \cap C_{kj}$

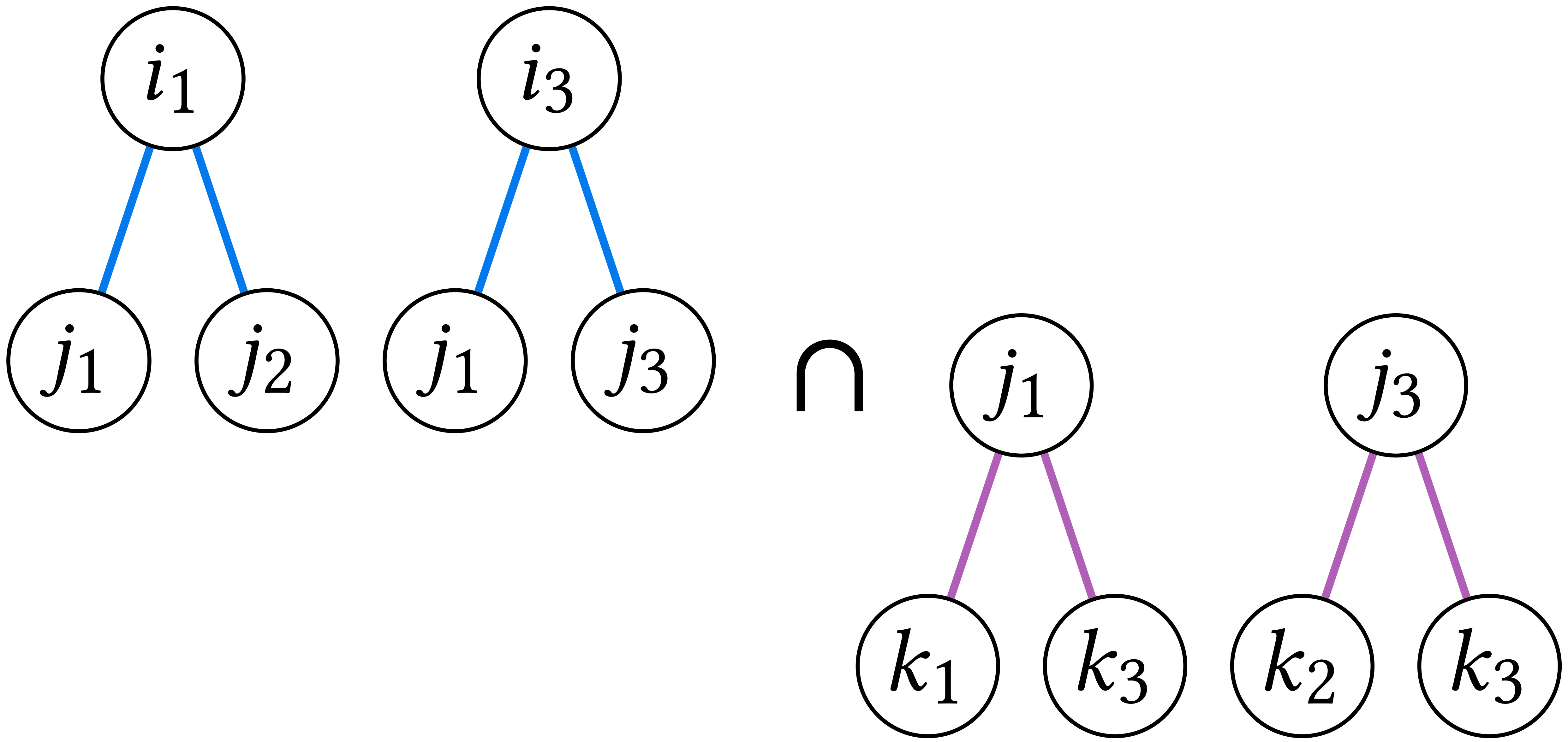


Sparse Iteration Theory

$$A_{ij} = \sum_k B_{ik} C_{kj}$$ ← Matrix Multiplication

Union of Coordinate Trees

$$B_{ik} \cap C_{kj}$$



Sparse Iteration Theory

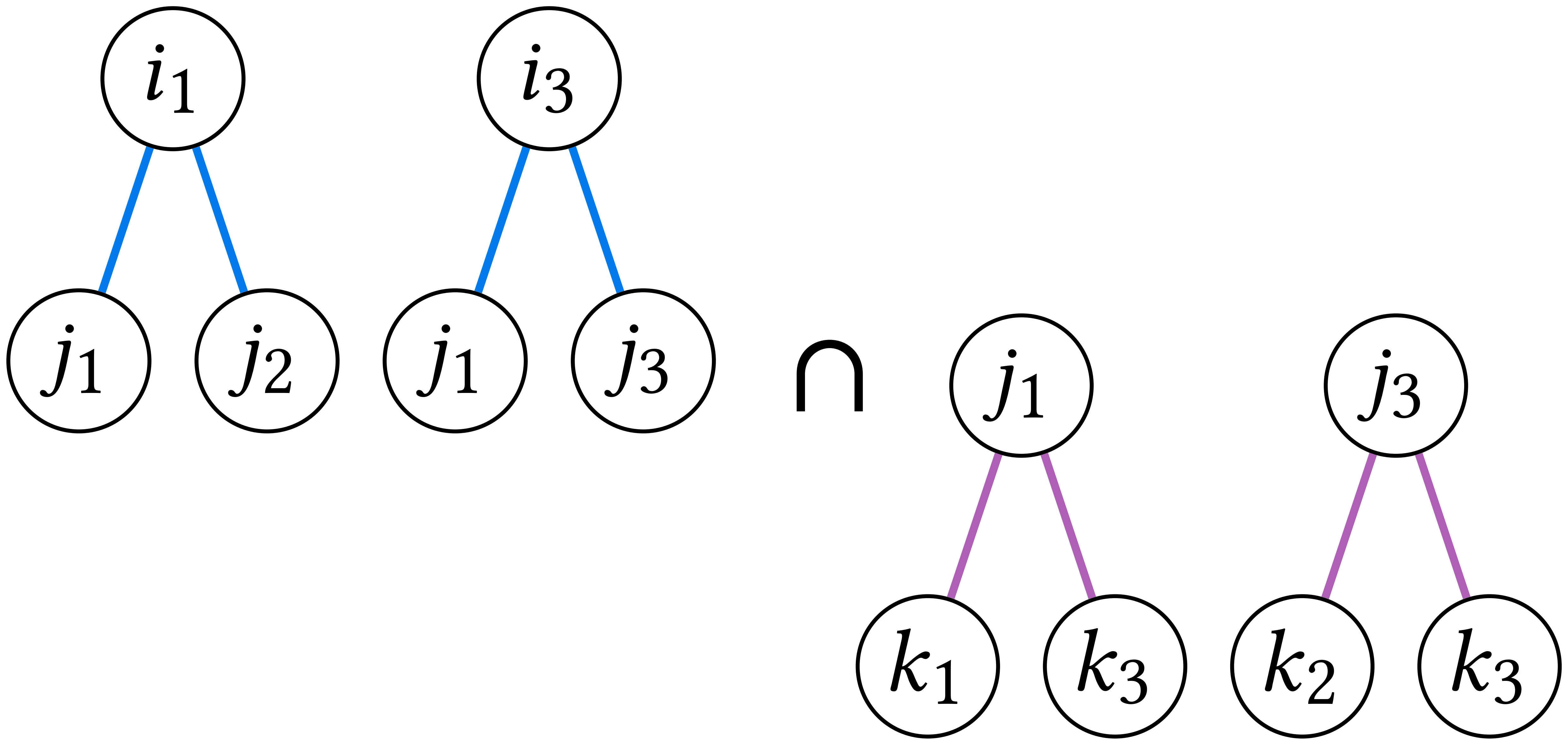
$$A_{ij} = \sum_k B_{ik} C_{kj}$$ ← Matrix Multiplication

Union of Coordinate Trees

$B_{ik} \cap C_{kj}$

Sparse Iteration Graph

$\forall_i \forall_k \forall_j \; B_{ik} \cap C_{kj}$

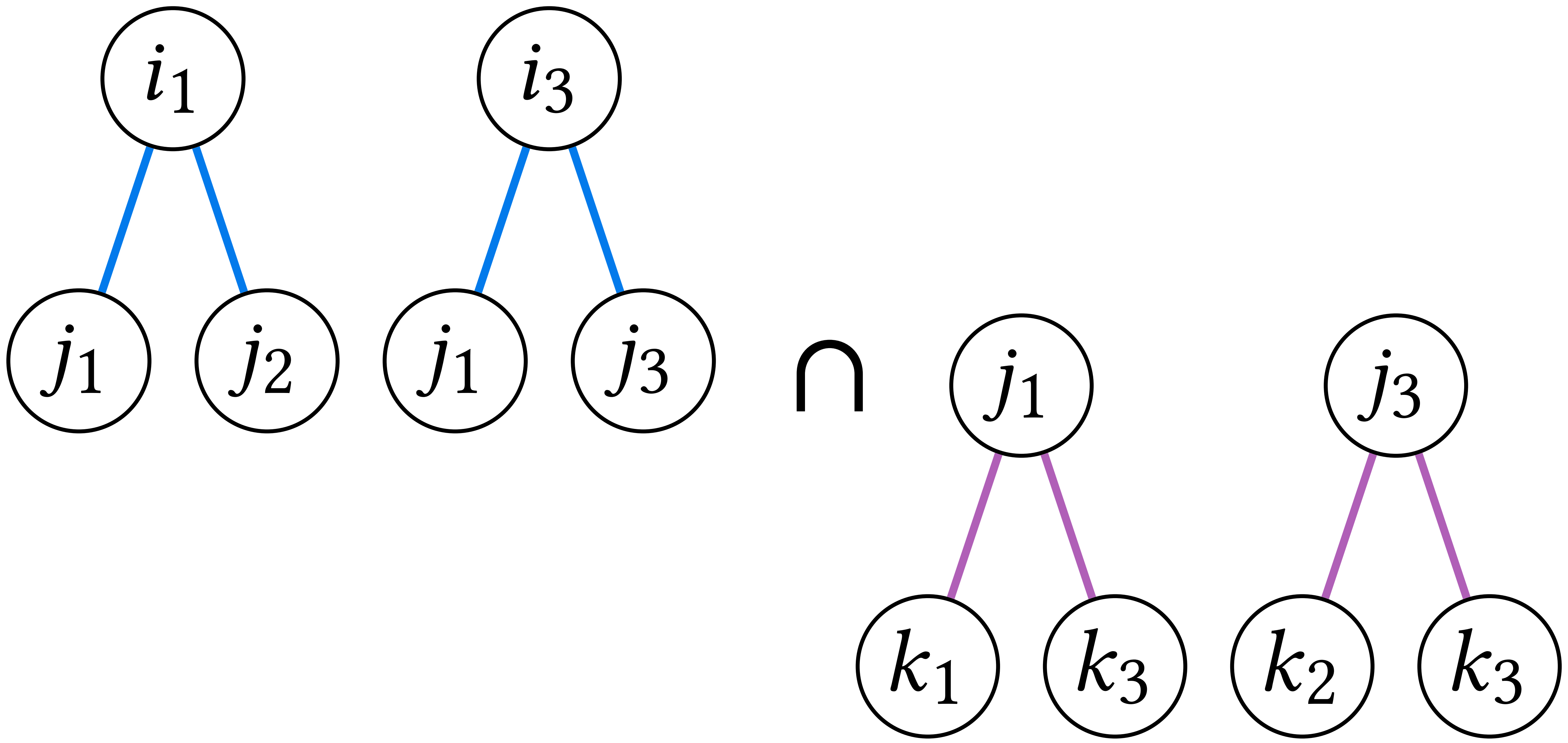


Sparse Iteration Theory

$$A_{ij} = \sum_k B_{ik} C_{kj}$$ ← Matrix Multiplication

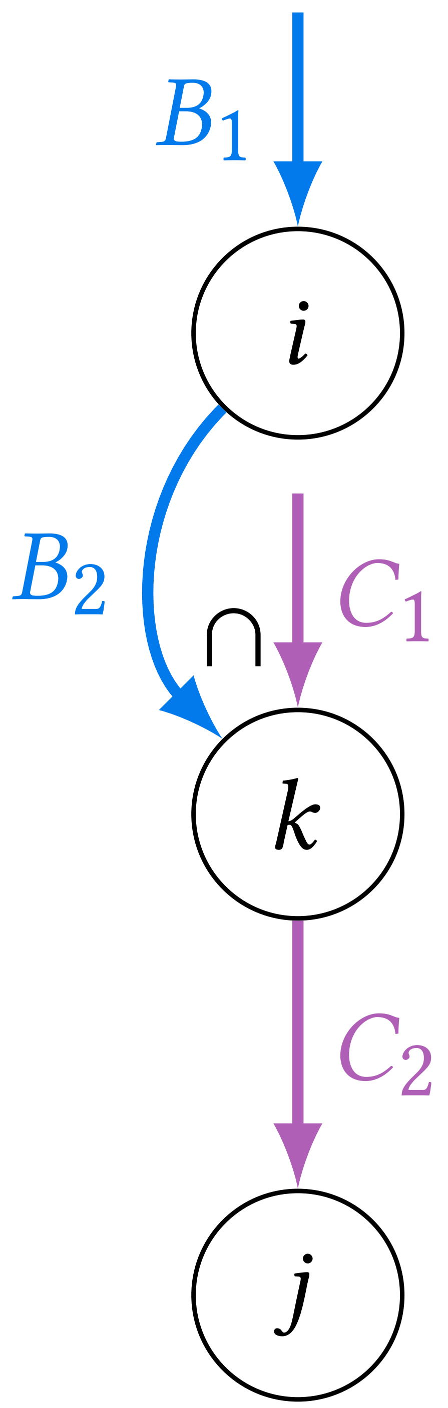
Union of Coordinate Trees

$B_{ik} \cap C_{kj}$



Sparse Iteration Graph

$\forall_i \forall_k \forall_j\ B_{ik} \cap C_{kj}$

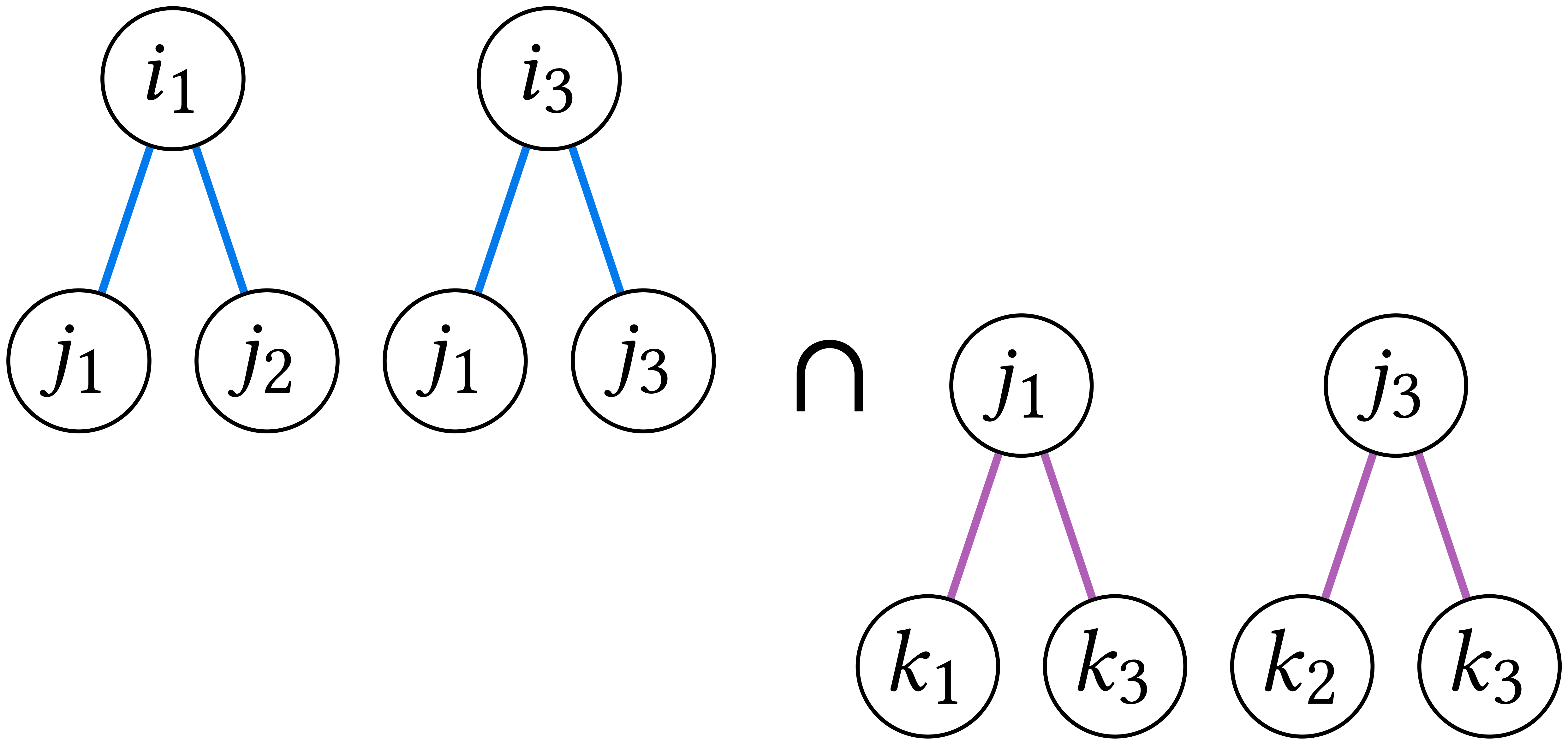


Sparse Iteration Theory

$$A_{ij} = \sum_k B_{ik} C_{kj}$$ ← Matrix Multiplication

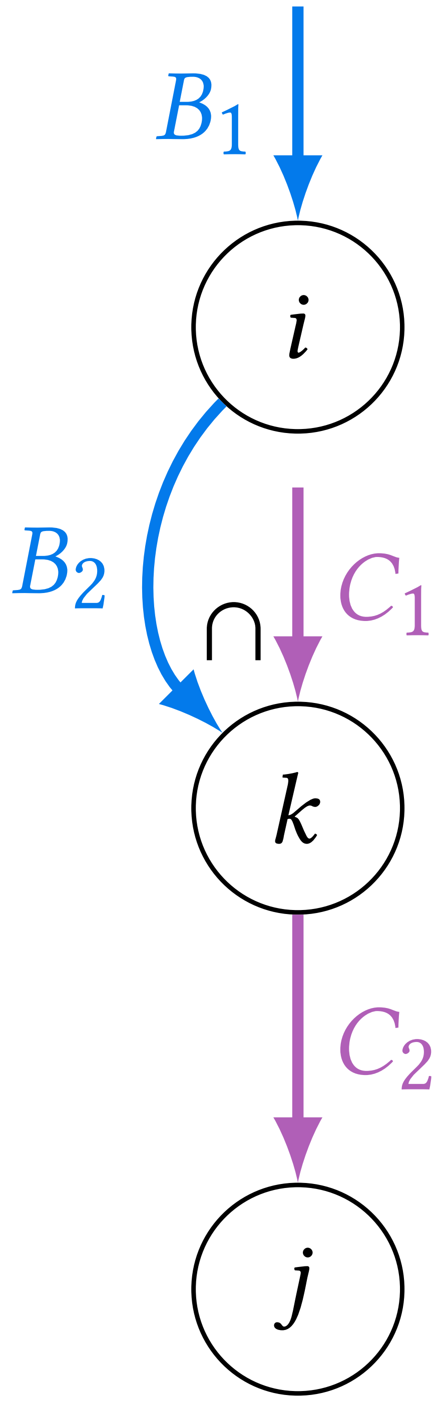
Union of Coordinate Trees

$B_{ik} \cap C_{kj}$



Sparse Iteration Graph

$\forall_i \forall_k \forall_j \; B_{ik} \cap C_{kj}$



$i \in B_1 \cap \mathbb{U}_i$

$k \in B_2 \cap C_1$

$j \in \mathbb{U}_j \cap C_2$

Sparse Iteration Theory

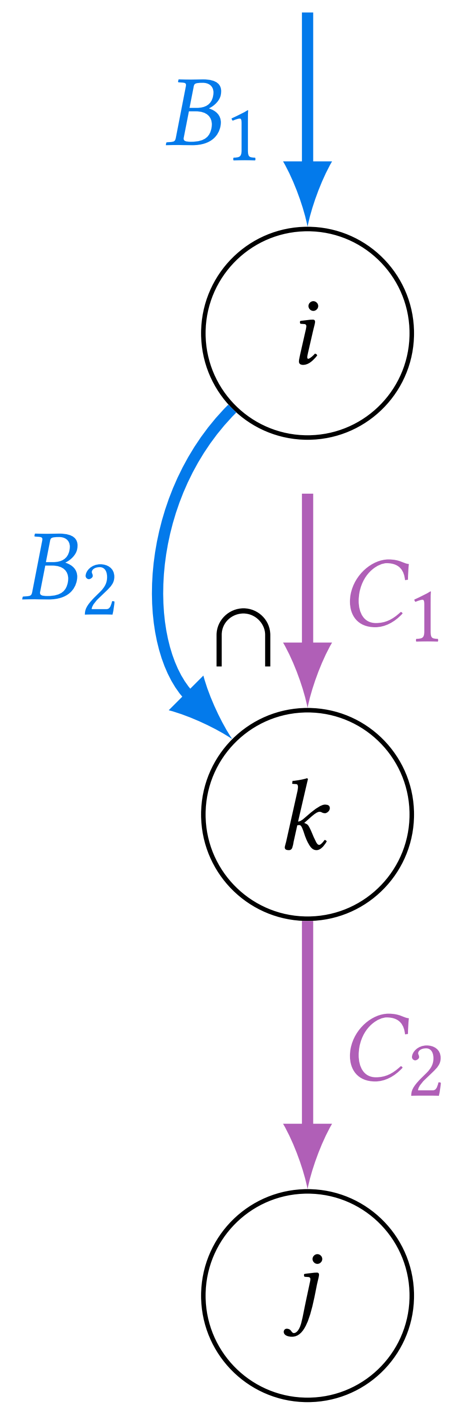
$$A_{ij} = \sum_k B_{ik} C_{kj}$$

Matrix Multiplication

Sparse Iteration Graph

$$\forall_i \forall_k \forall_j \; B_{ik} \cap C_{kj}$$

Must be able to generate code for any combination of intersections and unions of any number of data structures



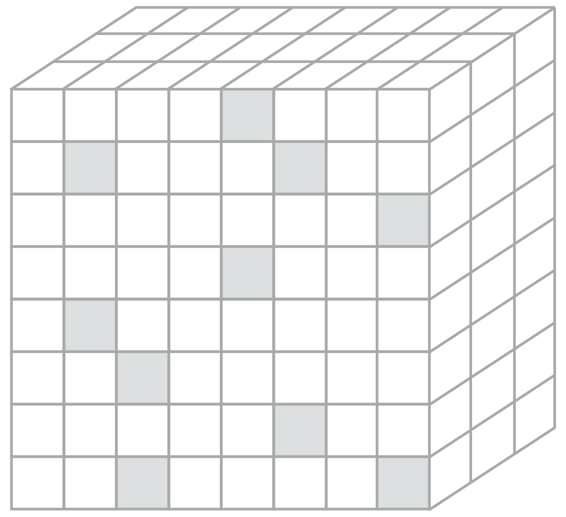
$i \in B_1 \cap \mathbb{U}_i$

$k \in B_2 \cap C_1$

$j \in \mathbb{U}_j \cap C_2$

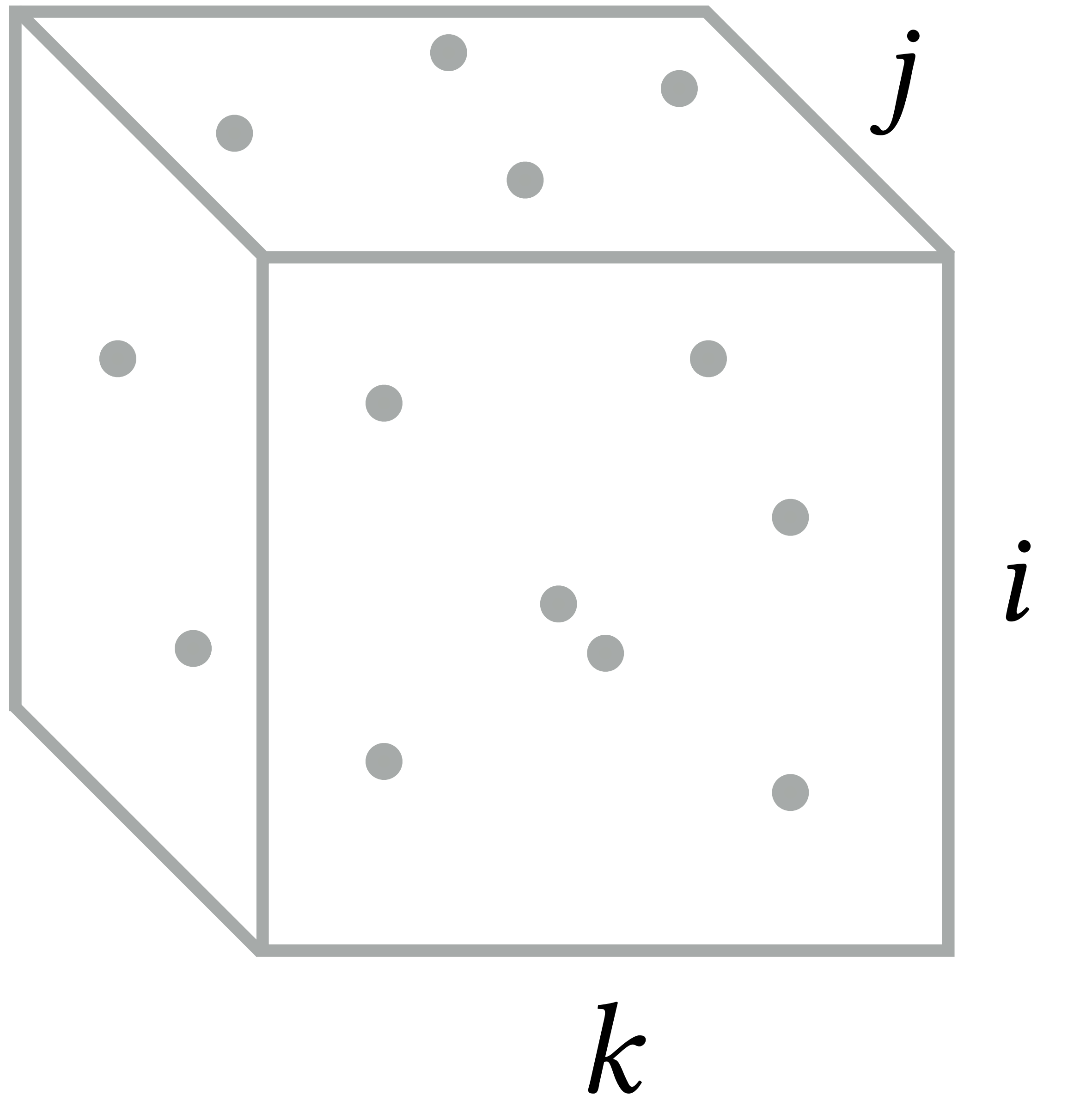
Sparse Iteration Theory

Tensors



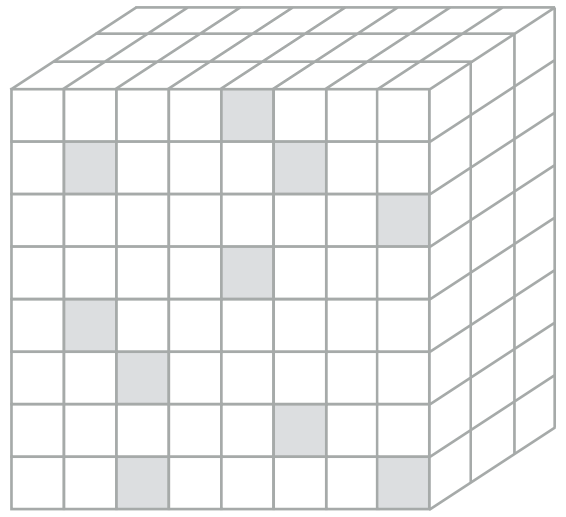
Nonzeros are a subset of the cartesian combination of sets

Sparse Iteration Spaces



Sparse Iteration Theory

Tensors



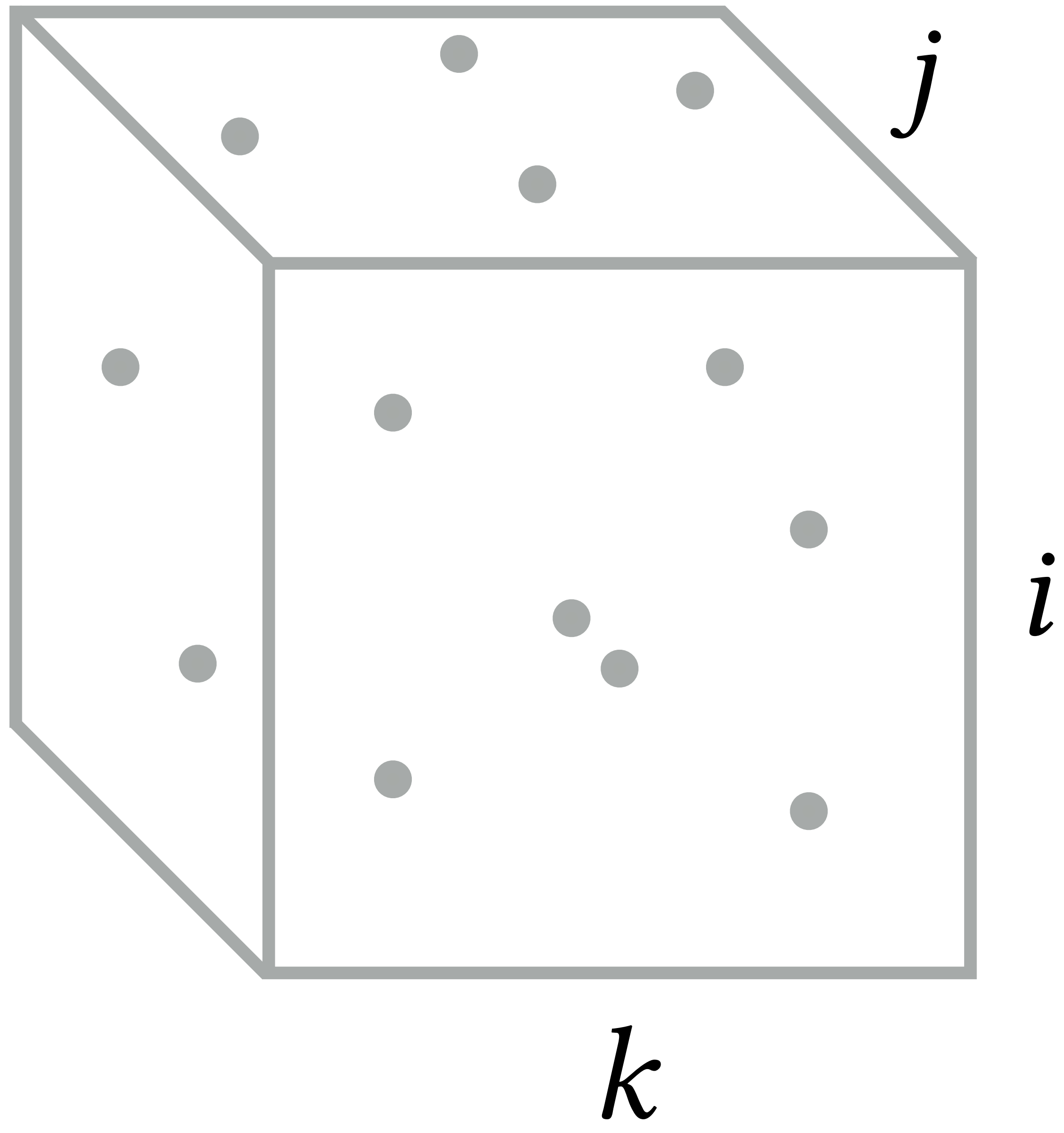
Nonzeros are a subset of the cartesian combination of sets

Relations

Names	City	Age
Peter	Boston	54
Mary	San Fransisco	35
Paul	New York	23
Adam	Seattle	84
Hilde	Boston	19
Bob	Chicago	76
Sam	Portland	32
Angela	Los Angeles	62

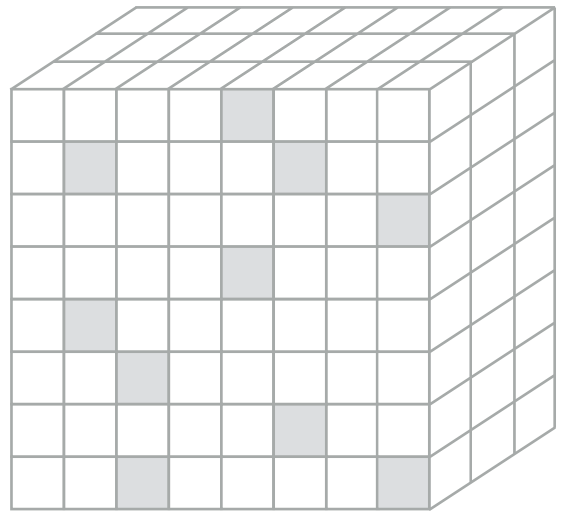
A relation is a subset of the cartesian combination of sets

Sparse Iteration Spaces



Sparse Iteration Theory

Tensors



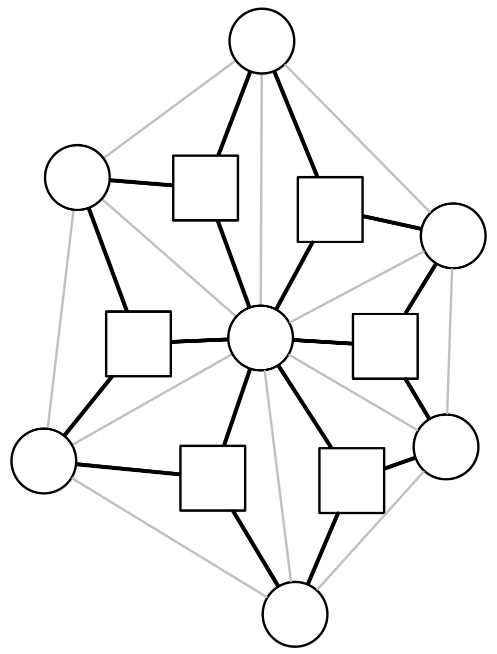
Nonzeros are a subset of the cartesian combination of sets

Relations

Names	City	Age
Peter	Boston	54
Mary	San Fransisco	35
Paul	New York	23
Adam	Seattle	84
Hilde	Boston	19
Bob	Chicago	76
Sam	Portland	32
Angela	Los Angeles	62

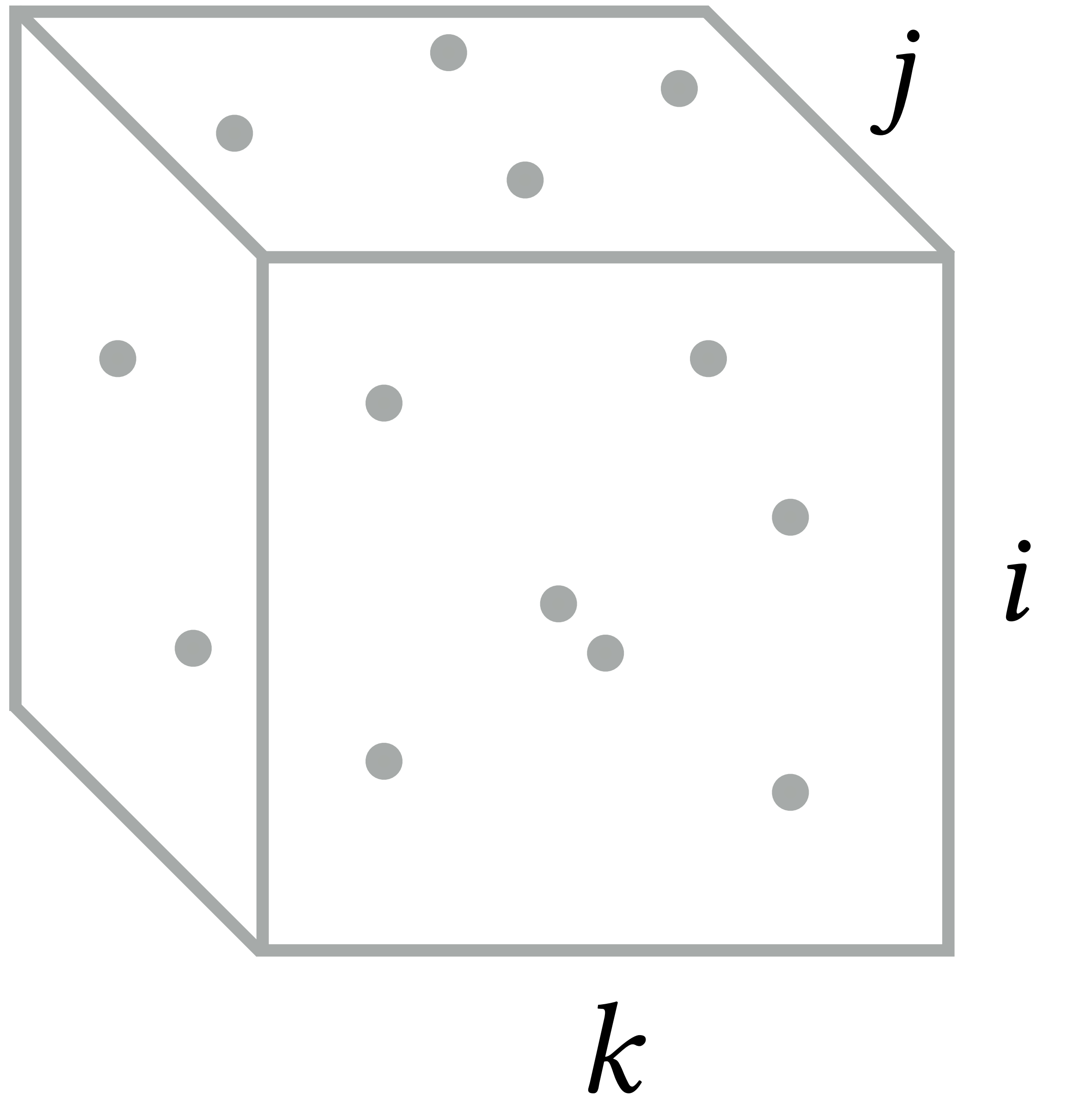
A relation is a subset of the cartesian combination of sets

Graphs



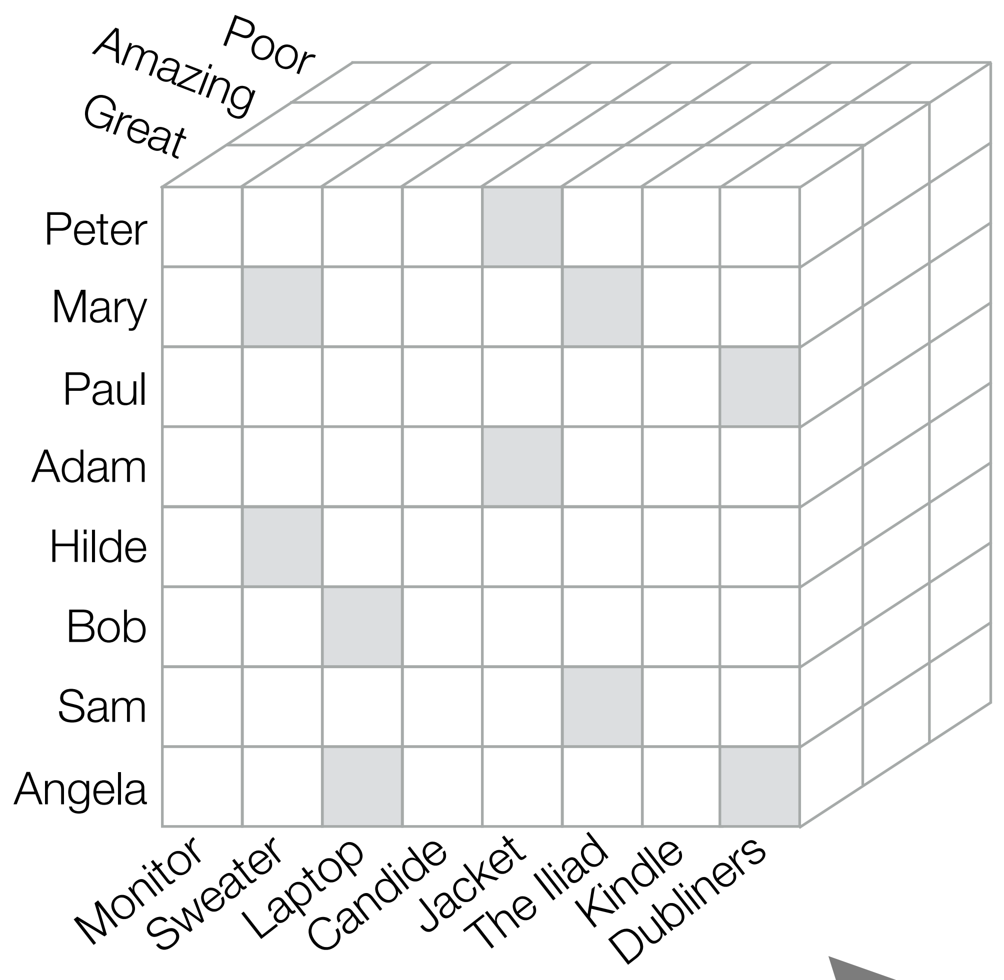
Graph edges are a subset of the cartesian combination of sets

Sparse Iteration Spaces



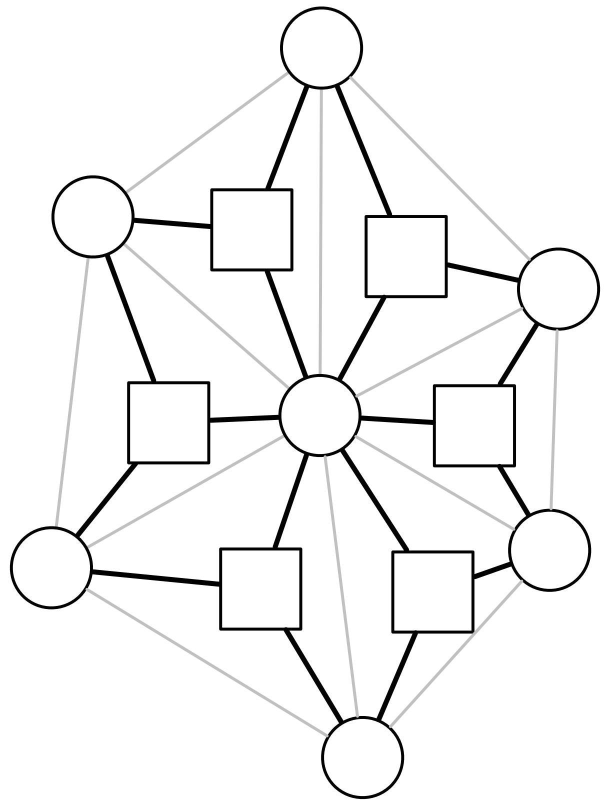
Three Abstractions

Tensors



Ideal for global operations on systems

Graphs



Ideal for local operations on systems

Relations

Names	City	Age
Peter	Boston	54
Mary	San Fransisco	35
Paul	New York	23
Adam	Seattle	84
Hilde	Boston	19
Bob	Chicago	76
Sam	Portland	32
Angela	Los Angeles	62

Ideal for combining data to form systems

Tensor Computations on Relational Data

Relations

Names	City	Age
Peter	Boston	54
Mary	San Fransisco	35
Paul	New York	23
Adam	Seattle	84
Hilde	Boston	19
Bob	Chicago	76
Sam	Portland	32
Angela	Los Angeles	62

Person	Product
Peter	Computer
Mary	Book
Paul	Pencil
Adam	Seattle
Hilde	Boston
Bob	Chicago
Sam	Portland
Angela	Los Angeles

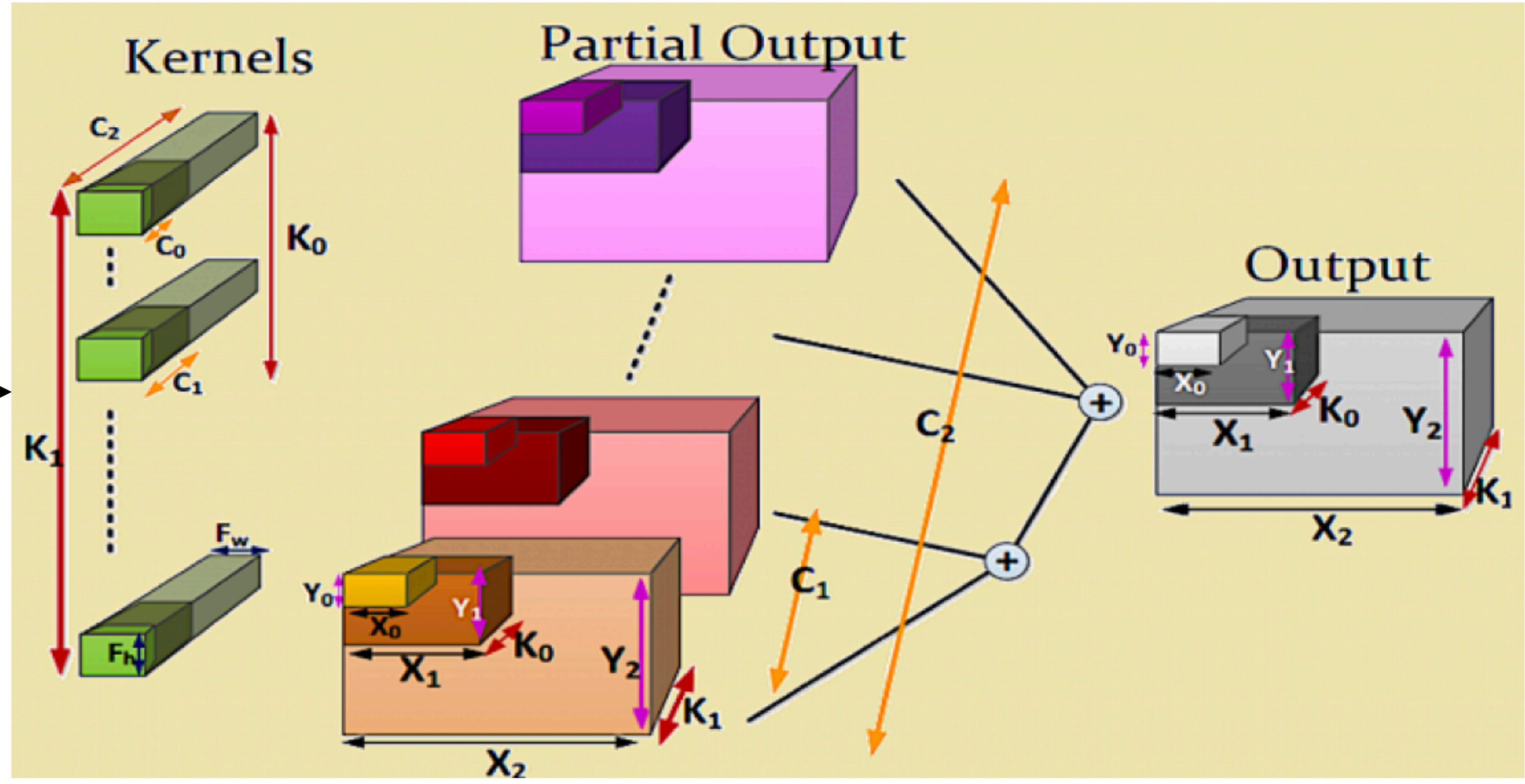
Data Processing

Filter

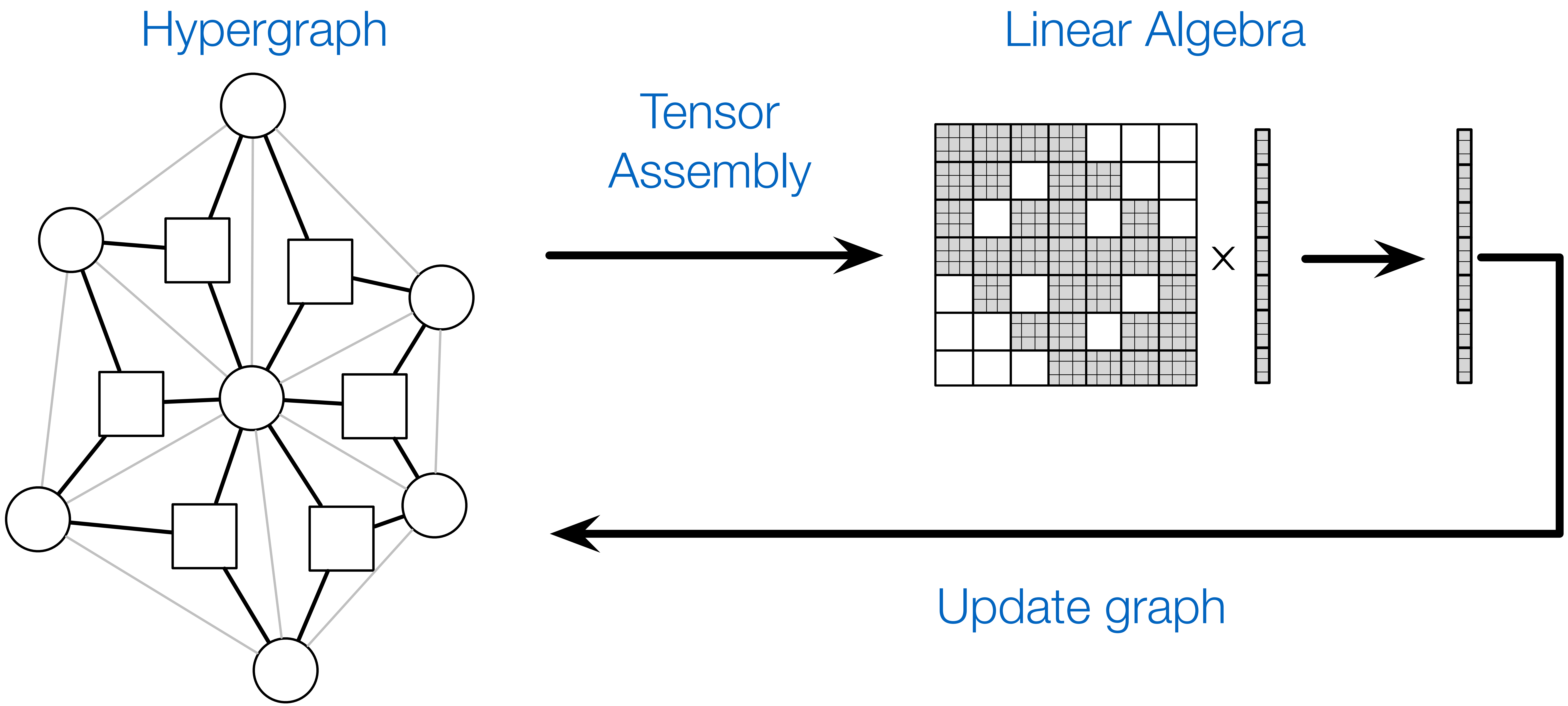
Filter

Join

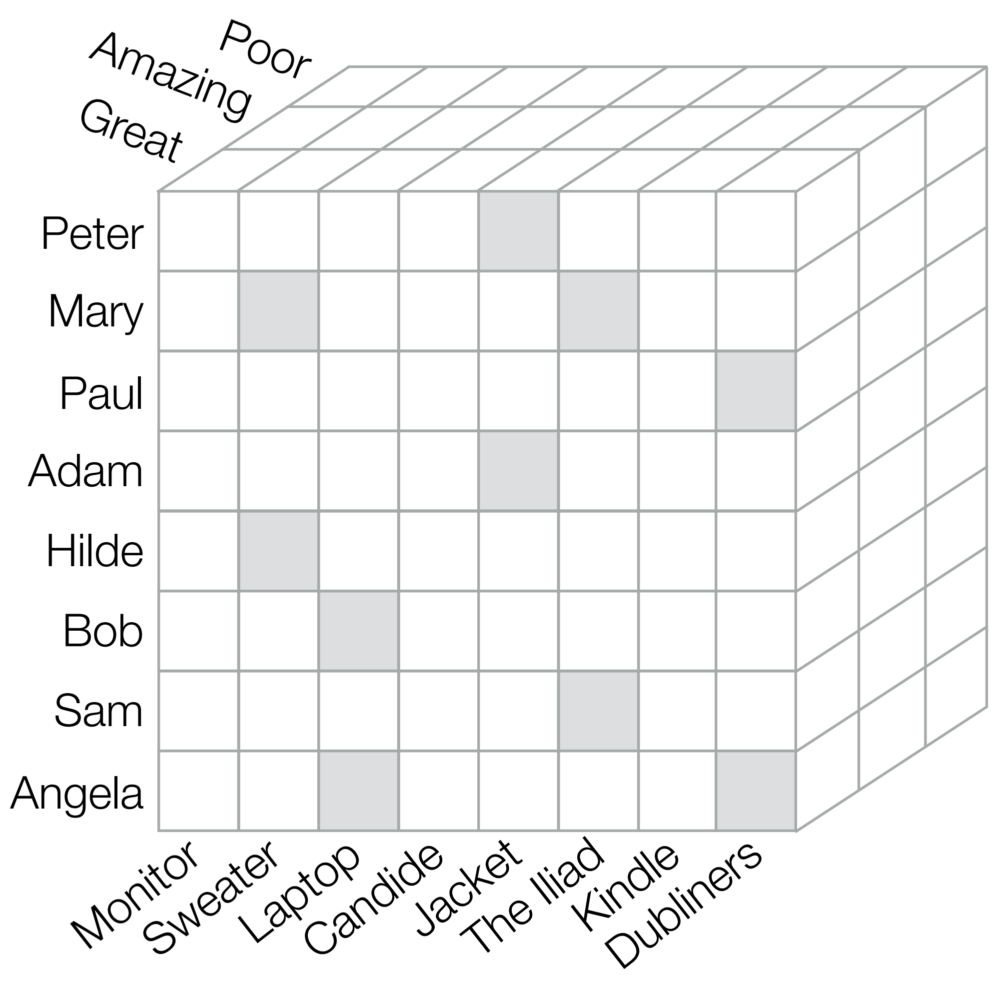
Tensor Operations

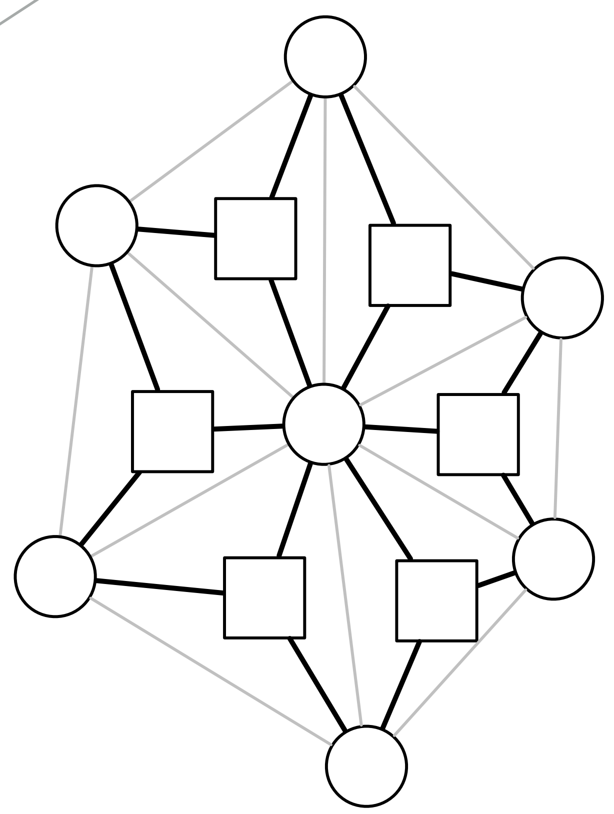


Simulation on Mesh or Rigid Bodies



Compilation to Domain-Specific Hardware





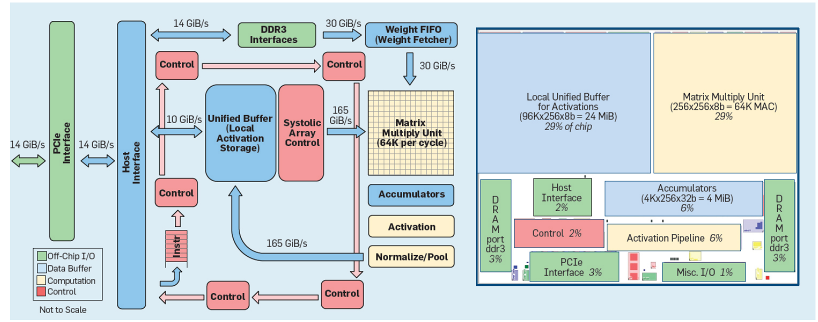
Names	City	Age
Peter	Boston	54
Mary	San Fransisco	35
Paul	New York	23
Adam	Seattle	84
Hilde	Boston	19
Bob	Chicago	76
Sam	Portland	32
Angela	Los Angeles	62

Three Languages Compiler

CPUs

GPUs

Cloud



DSAs