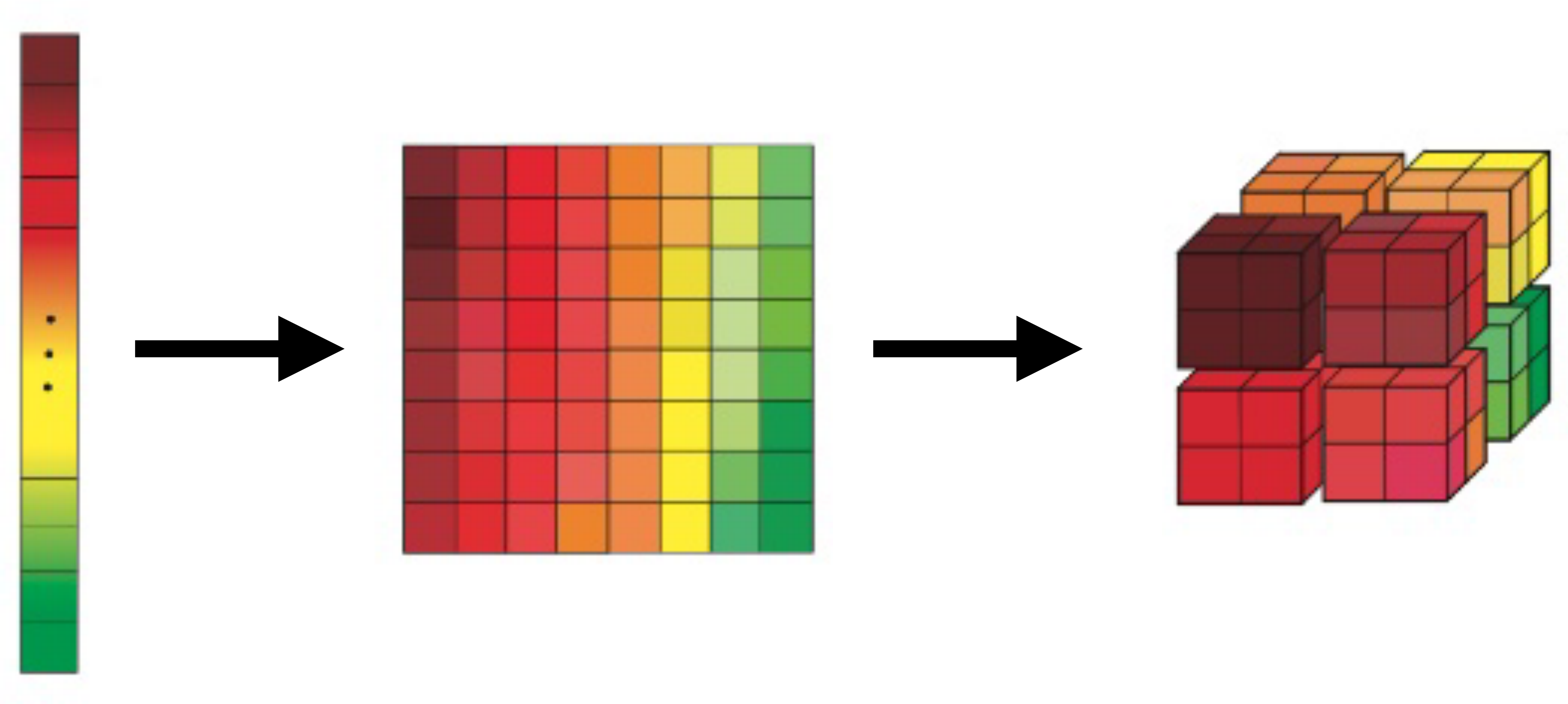


HiCOO: Hierarchical Storage of Sparse Tensors

Jiajia Li [1,2], Jimeng Sun [1], Richard Vuduc [1]

[1] Georgia Institute of Technology

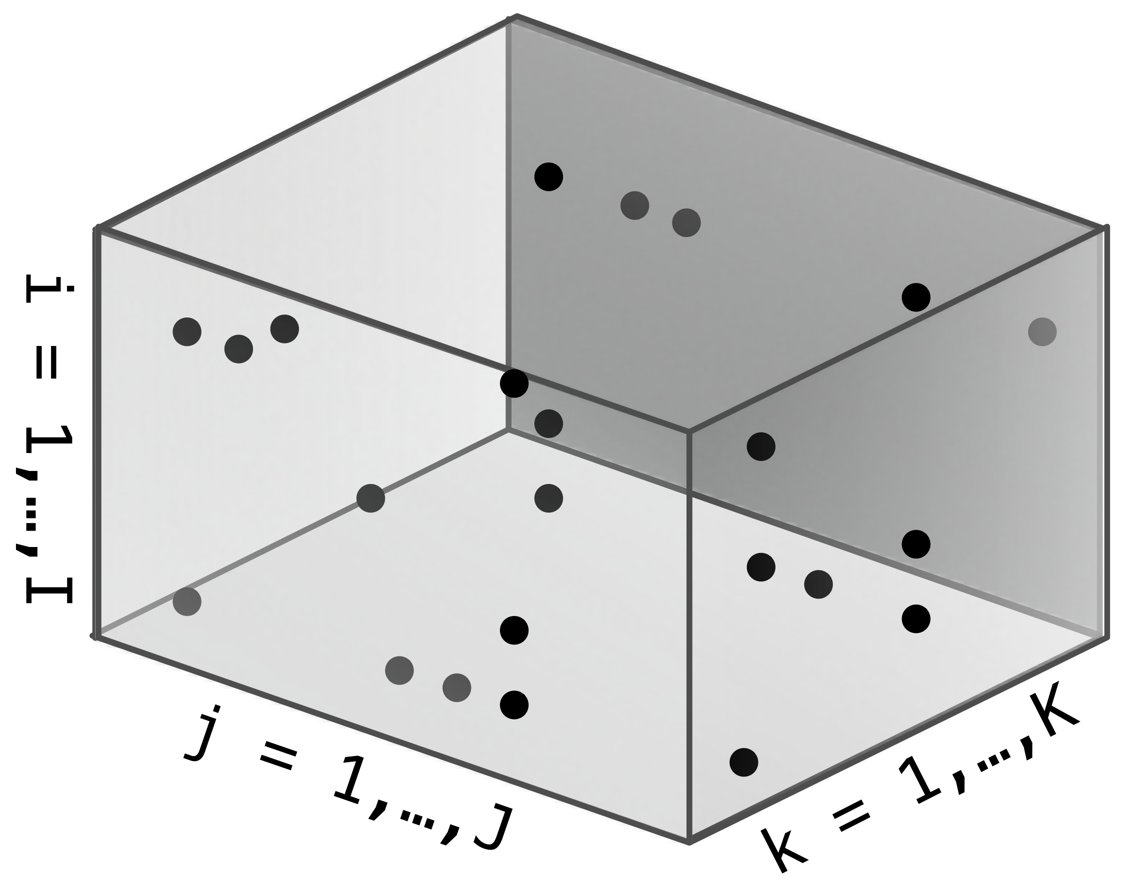
[2] Pacific Northwest National Laboratory

November 13, 2018 @ SC18

Figure sources: “A brief survey of tensors” by Berton Earnshaw and NVIDIA Tensor Cores

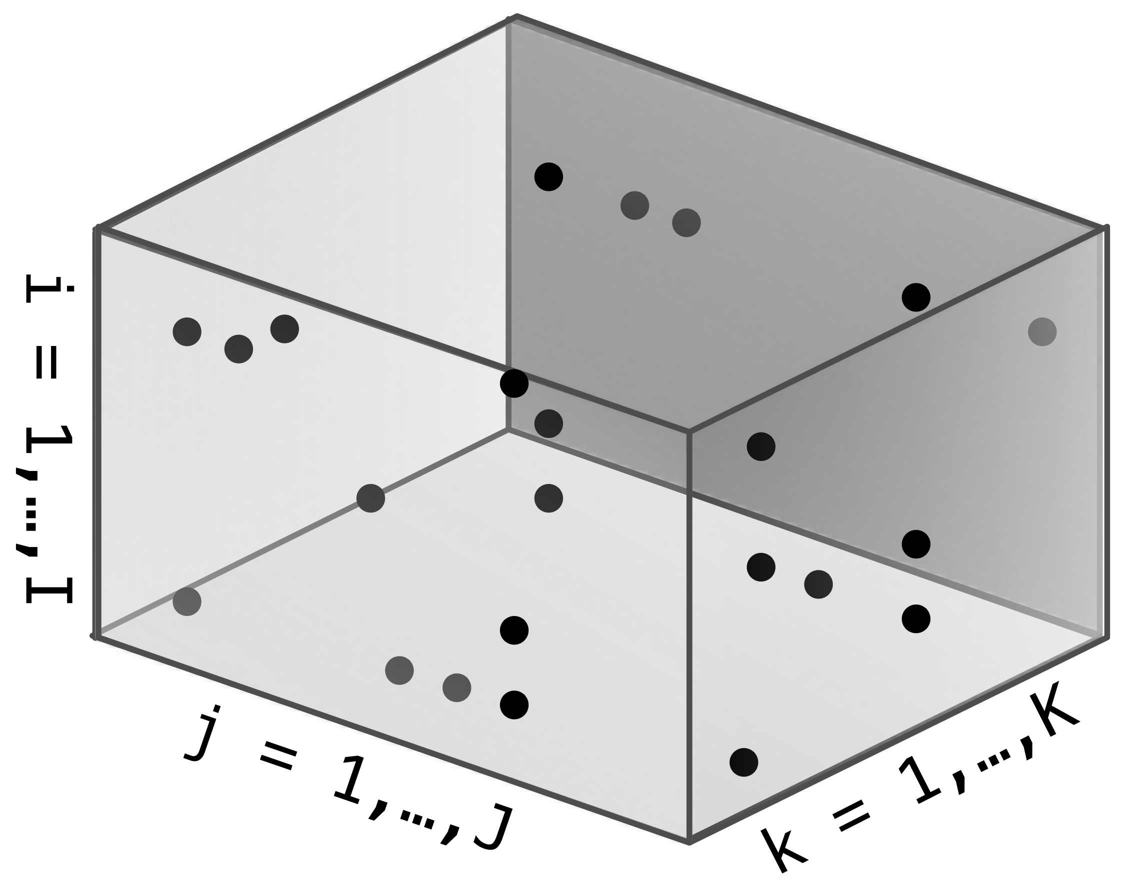
Tensors & Decompositions

- Tensors, multi-way arrays, provide a natural way to represent multi-relational data.
  - Special cases: matrices, vectors
  - Tensor mode or order: tensor dimension.
  - Data tensors in applications are usually SPARSE, meaning consisting mostly of zero entries.



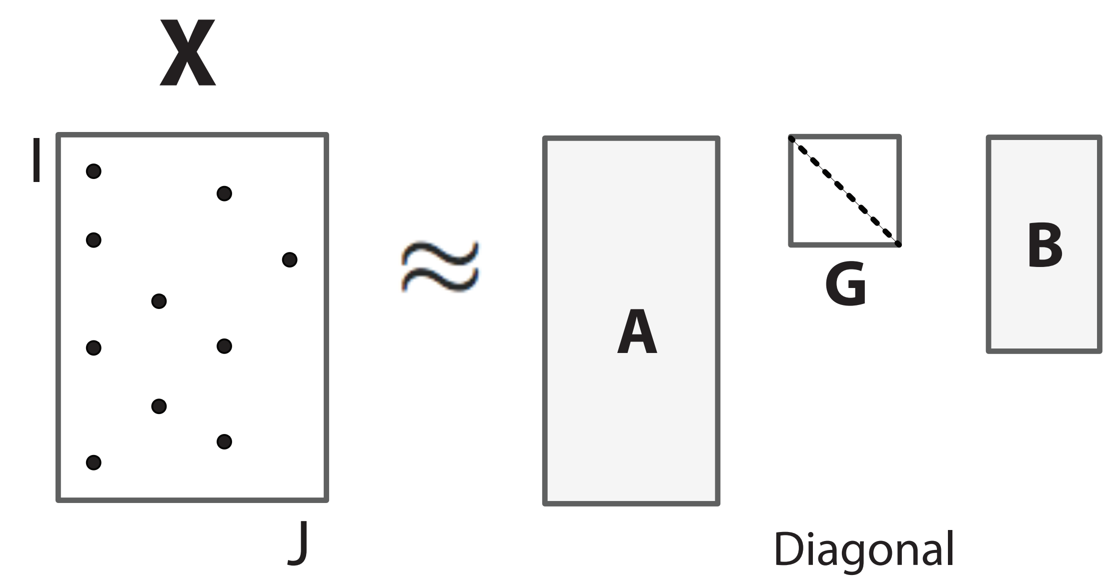
Tensors & Decompositions

- Tensors, multi-way arrays, provide a natural way to represent multi-relational data.
  - Special cases: matrices, vectors
  - Tensor mode or order: tensor dimension.
  - Data tensors in applications are usually SPARSE, meaning consisting mostly of zero entries.



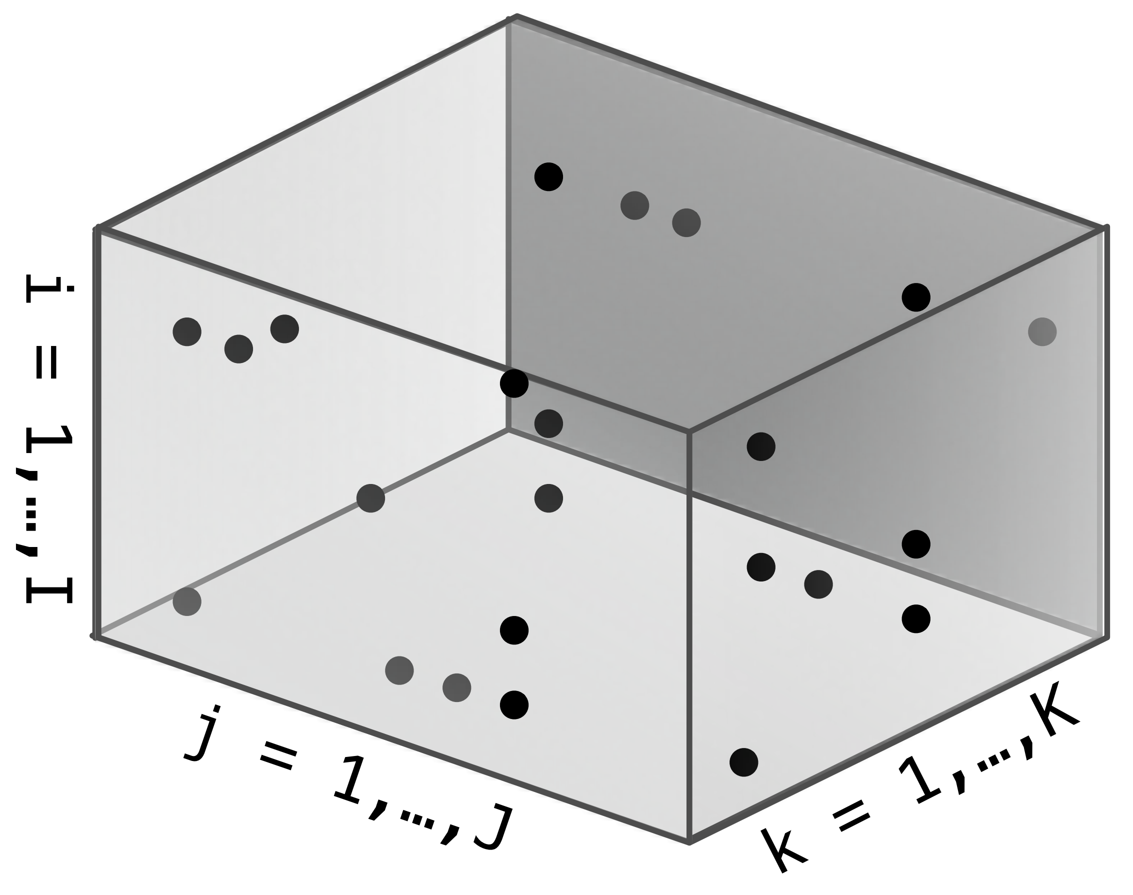
- Tensor decompositions: the natural generalization of matrix decompositions to tensors.

Singular Value Decomposition (SVD)



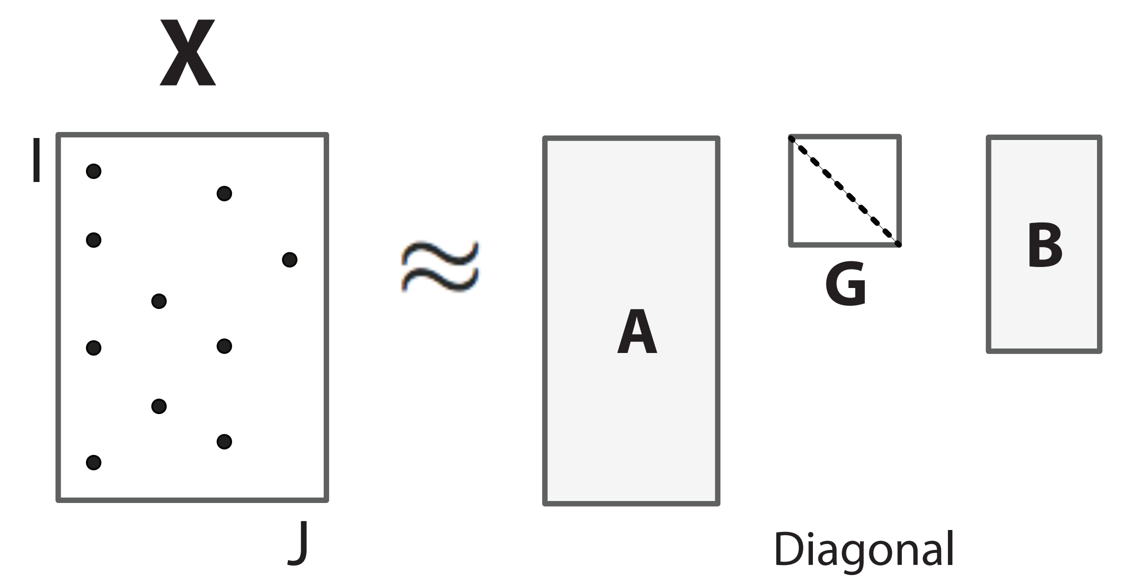
Tensors & Decompositions

- Tensors, multi-way arrays, provide a natural way to represent multi-relational data.
  - Special cases: matrices, vectors
  - Tensor mode or order: tensor dimension.
  - Data tensors in applications are usually <u>SPARSE</u>, meaning consisting mostly of zero entries.

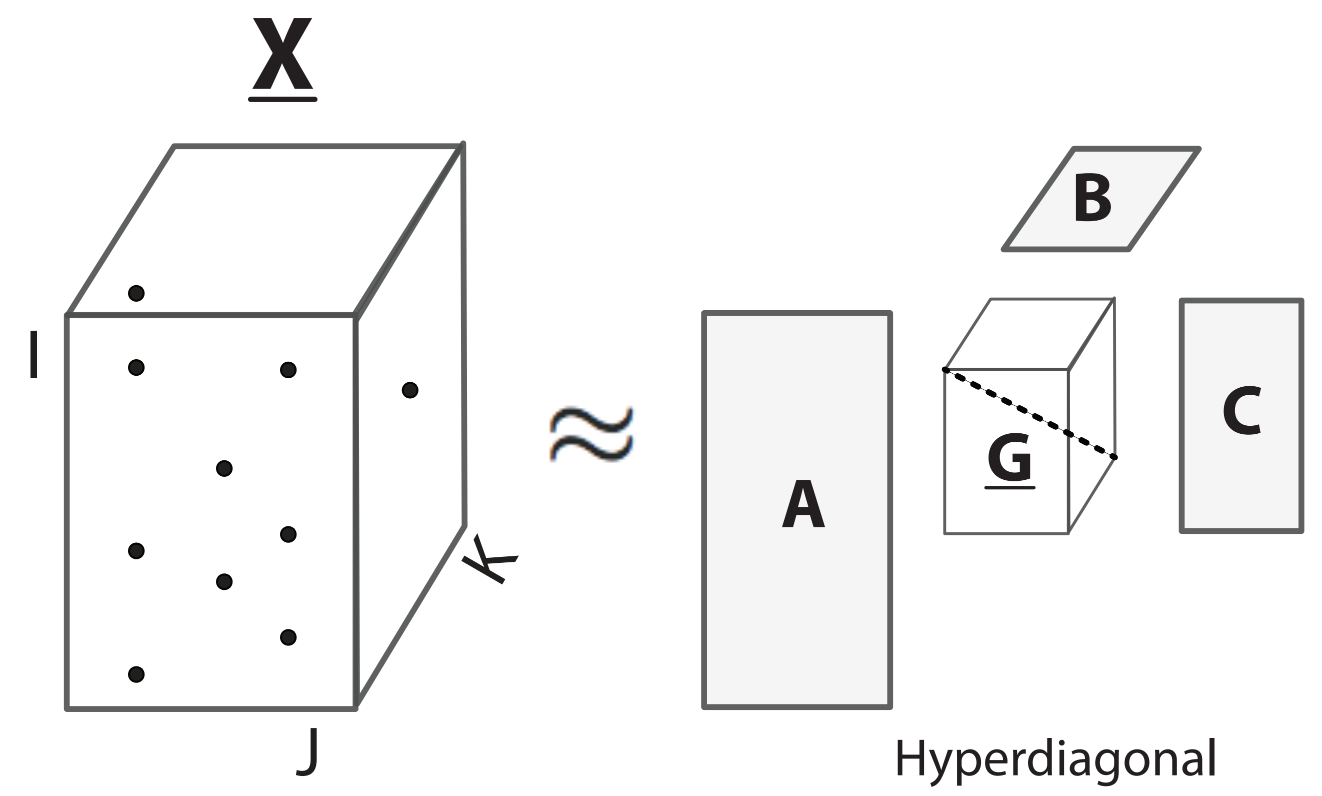


- Tensor decompositions: the natural generalization of matrix decompositions to tensors.

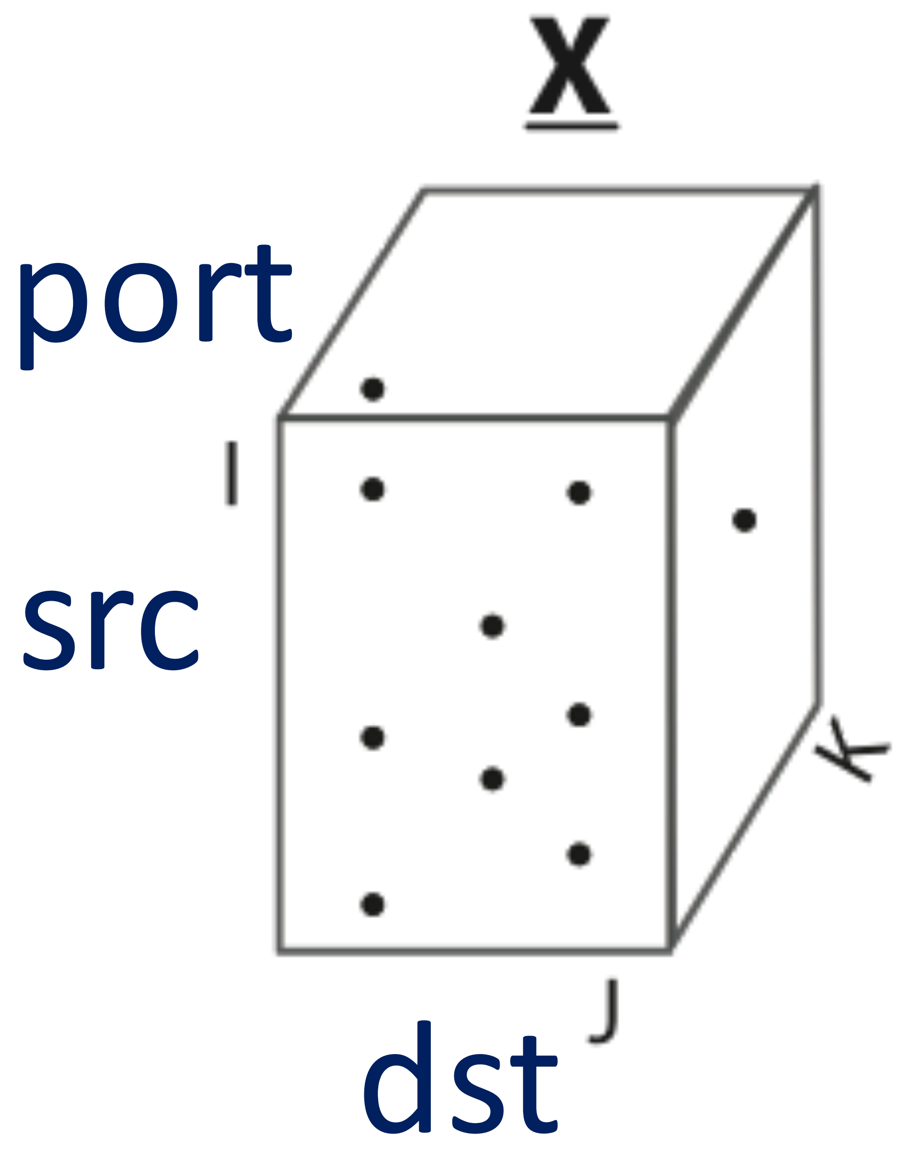
Singular Value Decomposition (SVD)



CANDECOMP/PARAFAC Decomposition (CPD)

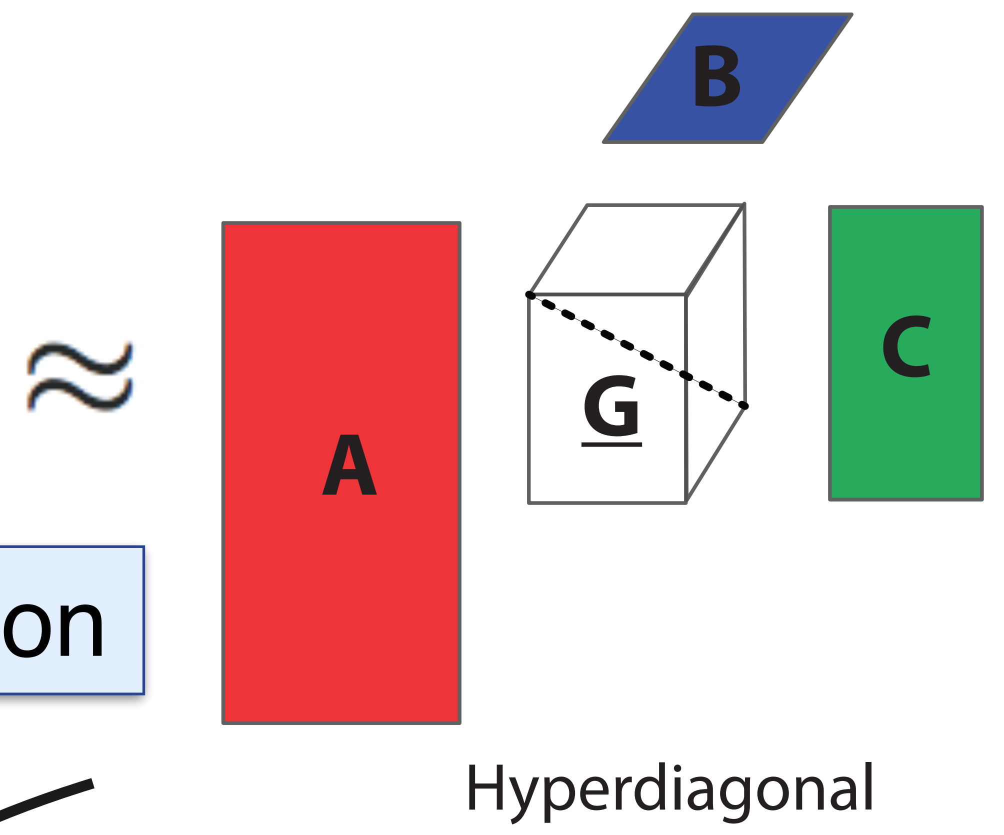


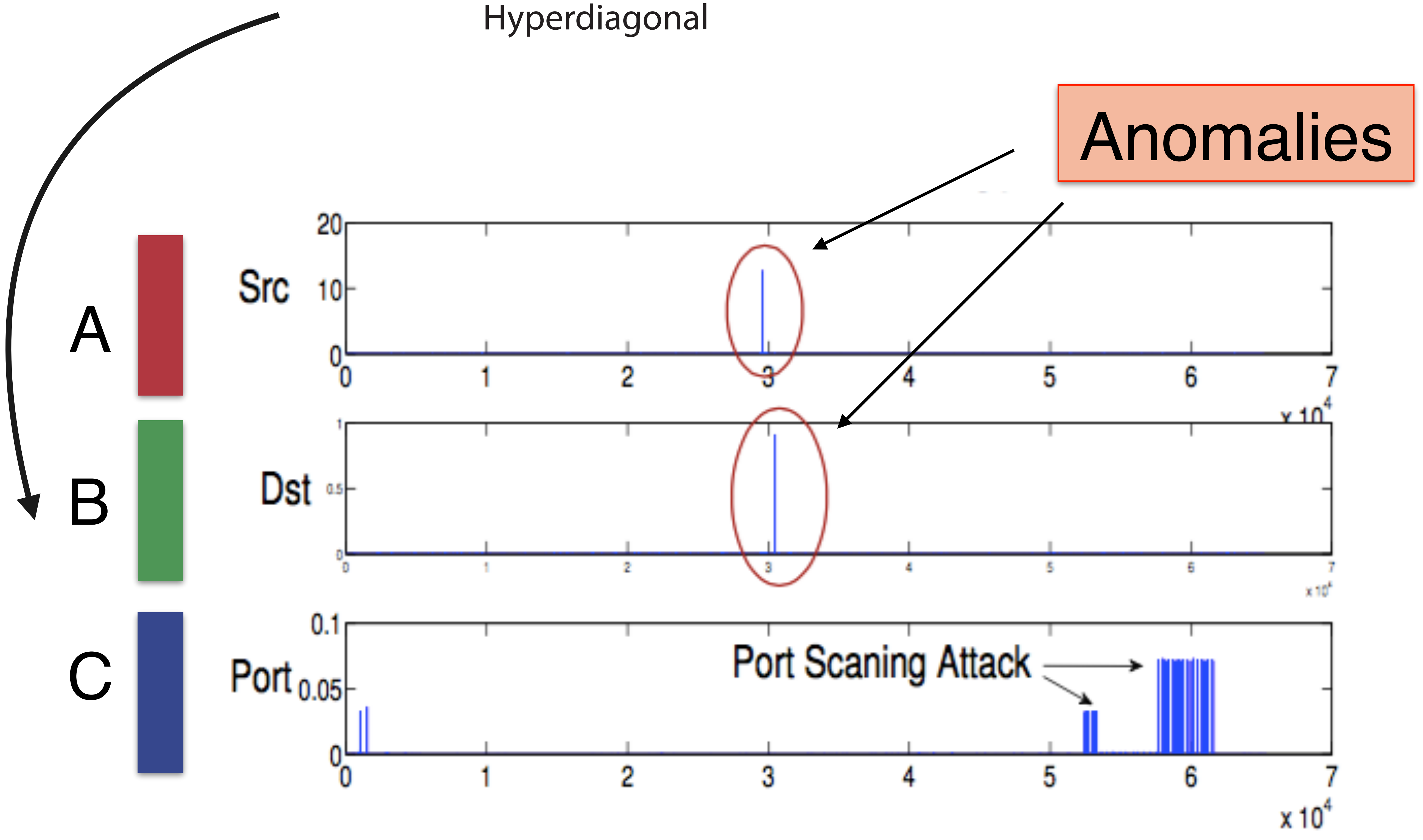
Tensor Decomposition for Anomaly Detection



Network data

Tensor decomposition





Source: ParCube, by Papalexakis et al. ECML-PKDD 2012

Outline

Definition → **Challenges** → HiCOO → Experiments

Challenges

- Compactness: A space-efficient data structure
- Mode-Genericity: Efficient traversals of the data structure for computations

The concept “mode-genericity” is inherited from [Baskaran et al. 2012].

[Baskaran et al. 2012] M. Baskaran et al., “Efficient and scalable computations with sparse tensors,” HPEC2012

Mode Genericity

- Matrix case:

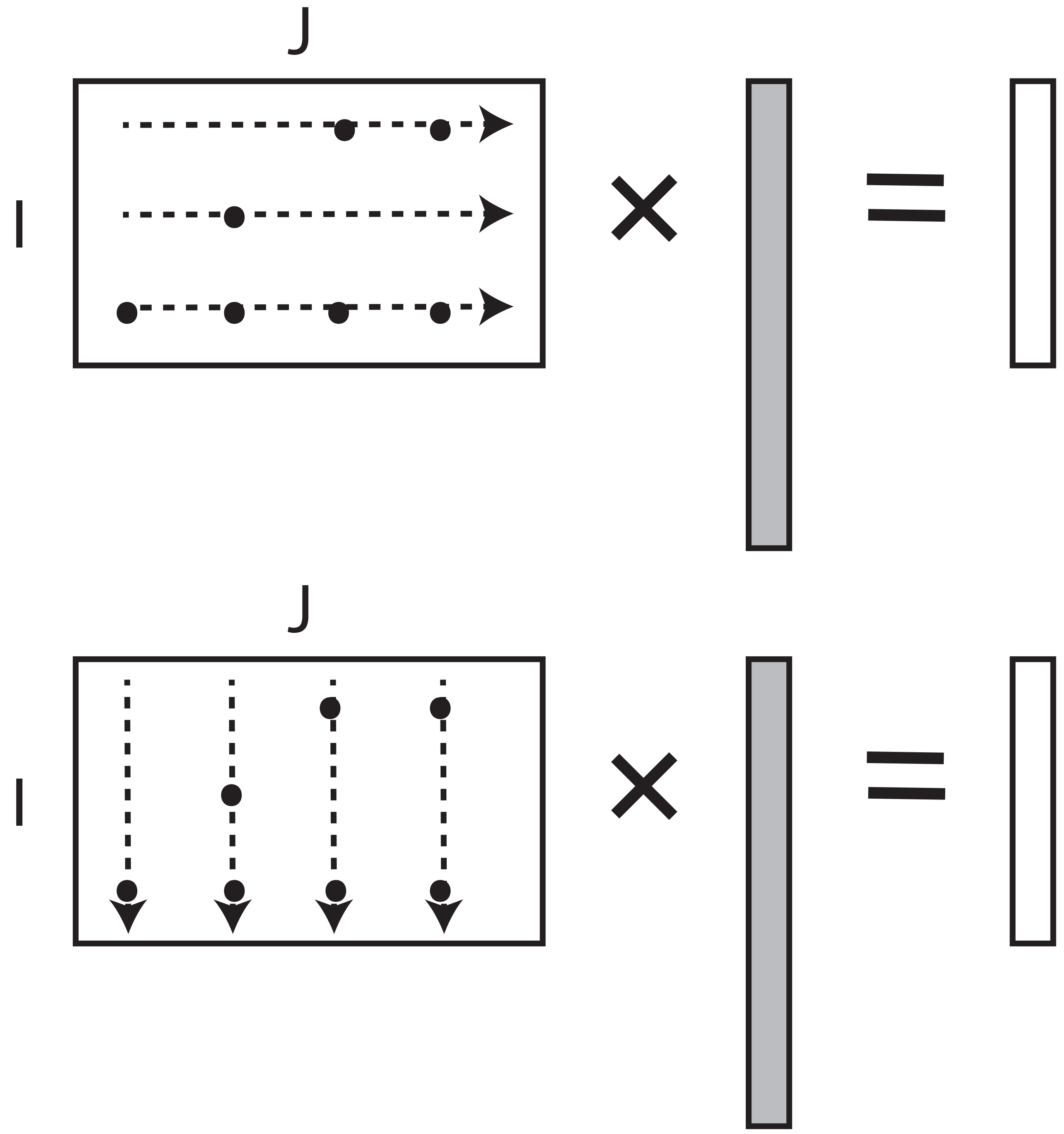
  Do both matrix-vector multiplication and matrix-transpose-vector multiplication.

Mode-Specific

Row-oriented (CSR)

SpMV

SpMTV

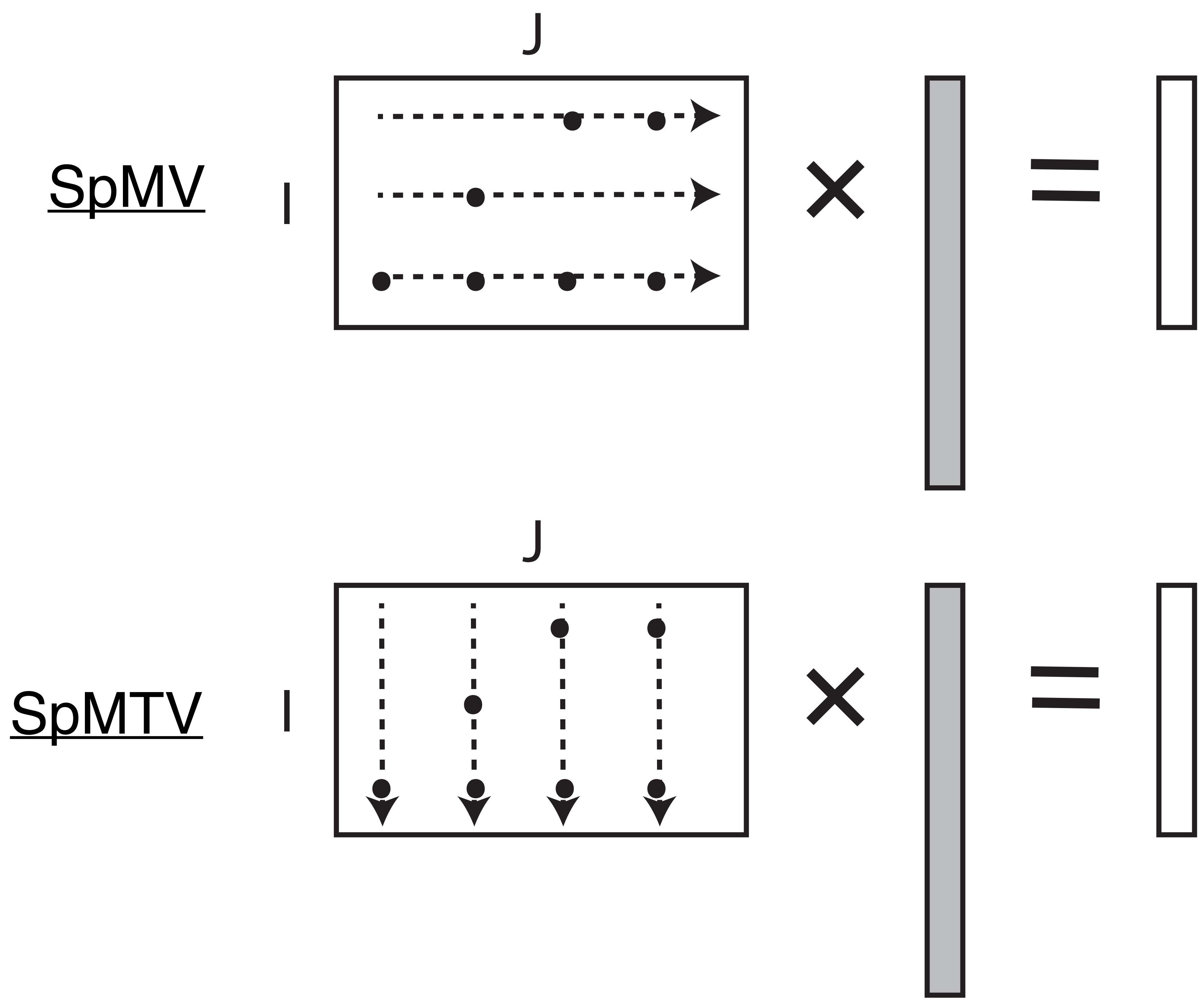


 Efficient  In-efficient

Mode Genericity

- Matrix case:

Do both matrix-vector multiplication and matrix-transpose-vector multiplication.



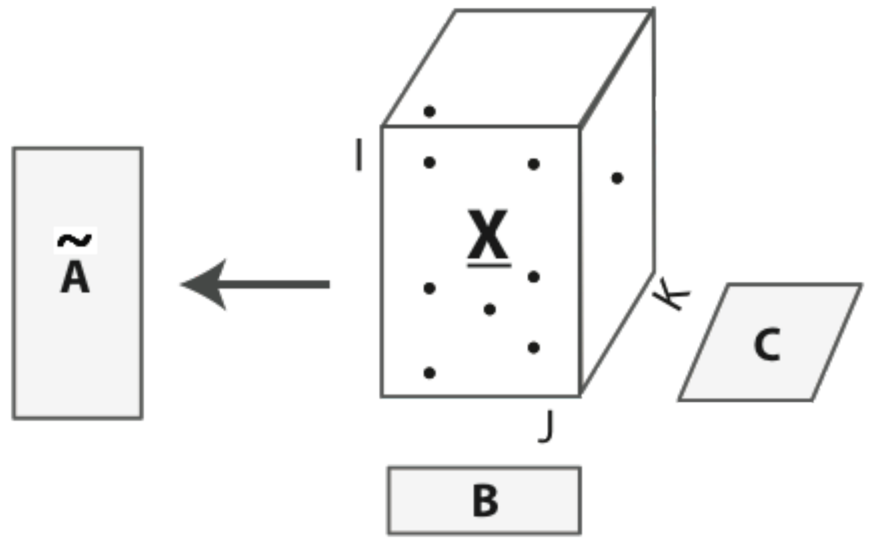
	Mode-Specific	Mode-Generic
	Row-oriented (CSR)	Coordinate (COO)
SpMV	✓	✗
SpMTV	✗	✗

✓ Efficient ✗ In-efficient

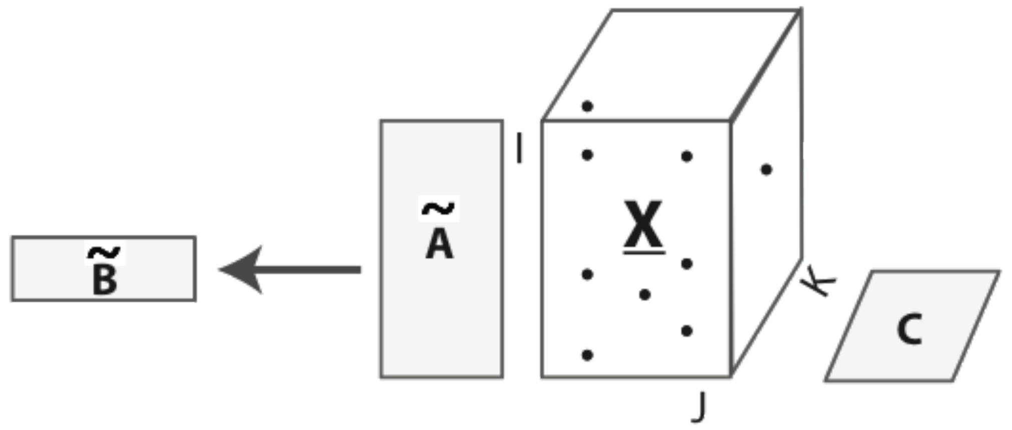
Mode Genericity

Tensor decomposition

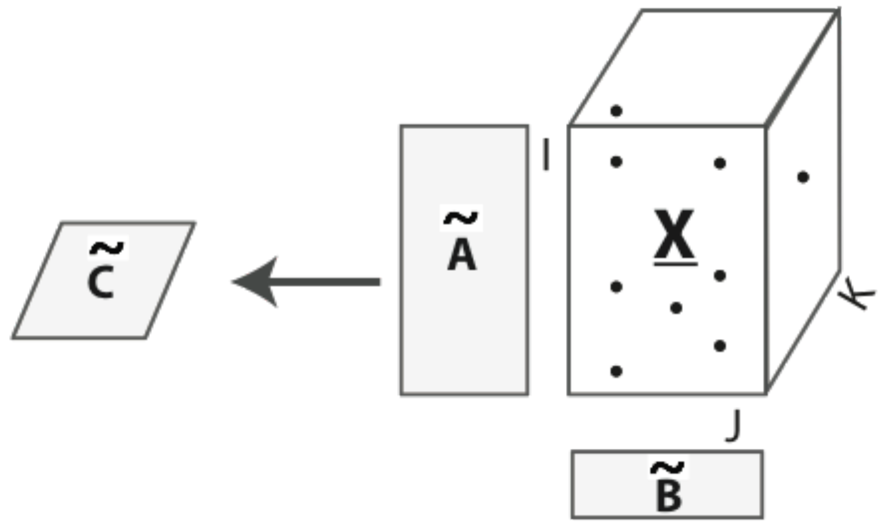
Kernel in Mode-1



Kernel in Mode-2



Kernel in Mode-3



Tensor decomposition	Mode-Specific	Mode-Generic
	Mode-1 oriented (CSF/FCOO)	Coordinate (COO)
Kernel in Mode-1	✓	✗
Kernel in Mode-2	✗	✗
Kernel in Mode-3	✗	✗

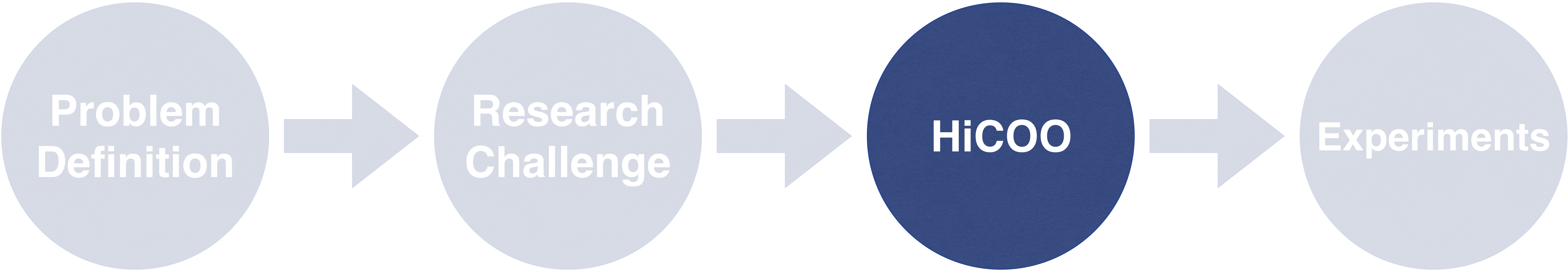
✓ Efficient ✗ In-efficient

Outline

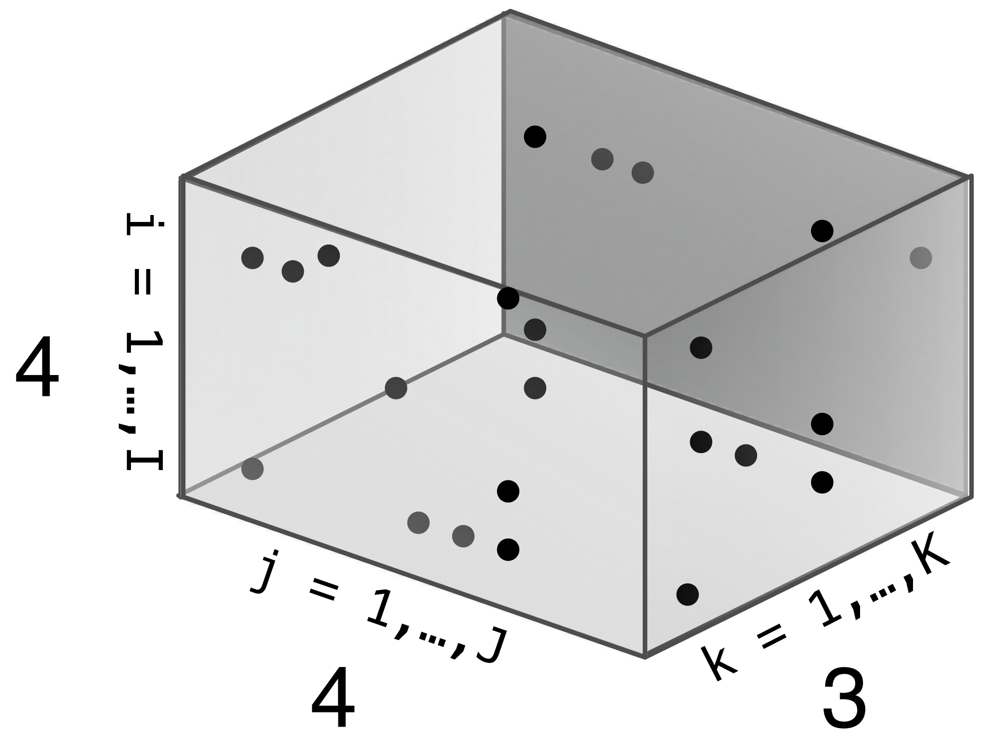
QUESTION: Is there a data structure that is BOTH compact and mode-generic?

Outline

QUESTION: Is there a data structure that is BOTH compact and mode-generic?

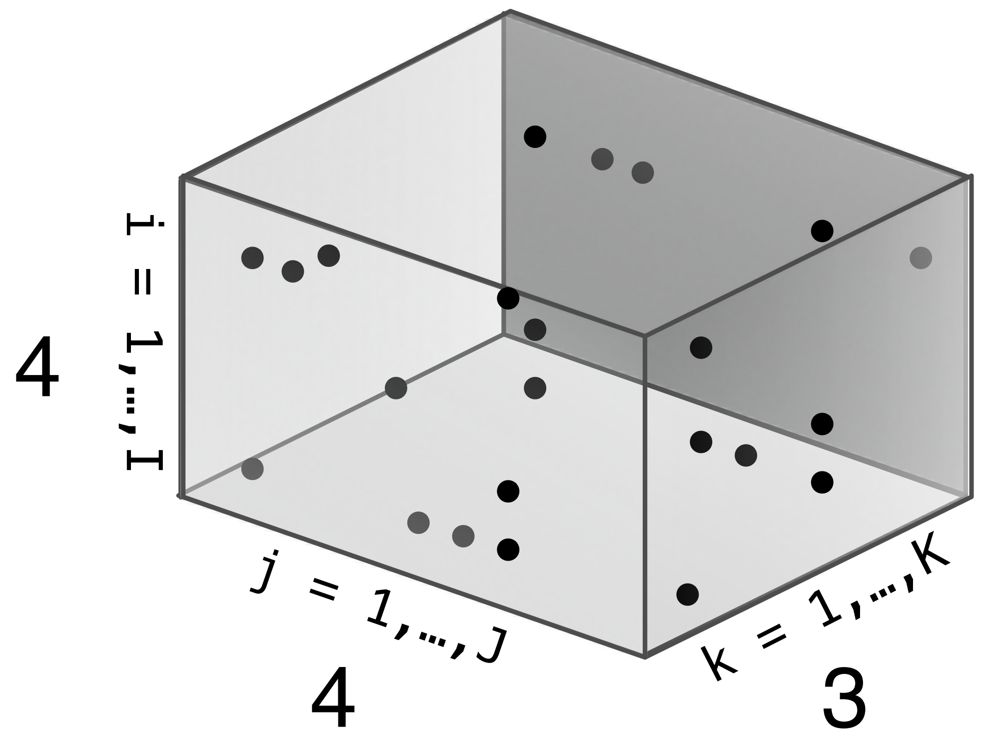


Baseline Sparse Tensor Formats in This Work



Baseline Sparse Tensor Formats in This Work

- COO: coordinate formats [Bader et al., 2006]



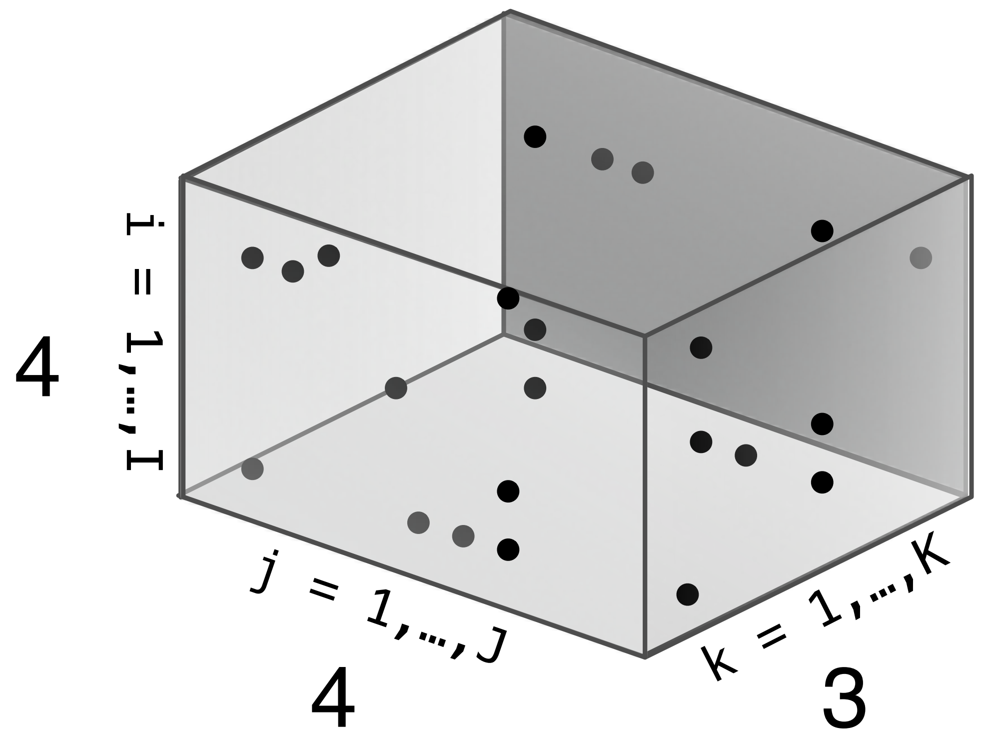
i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

(a) COO

Mode-Generic

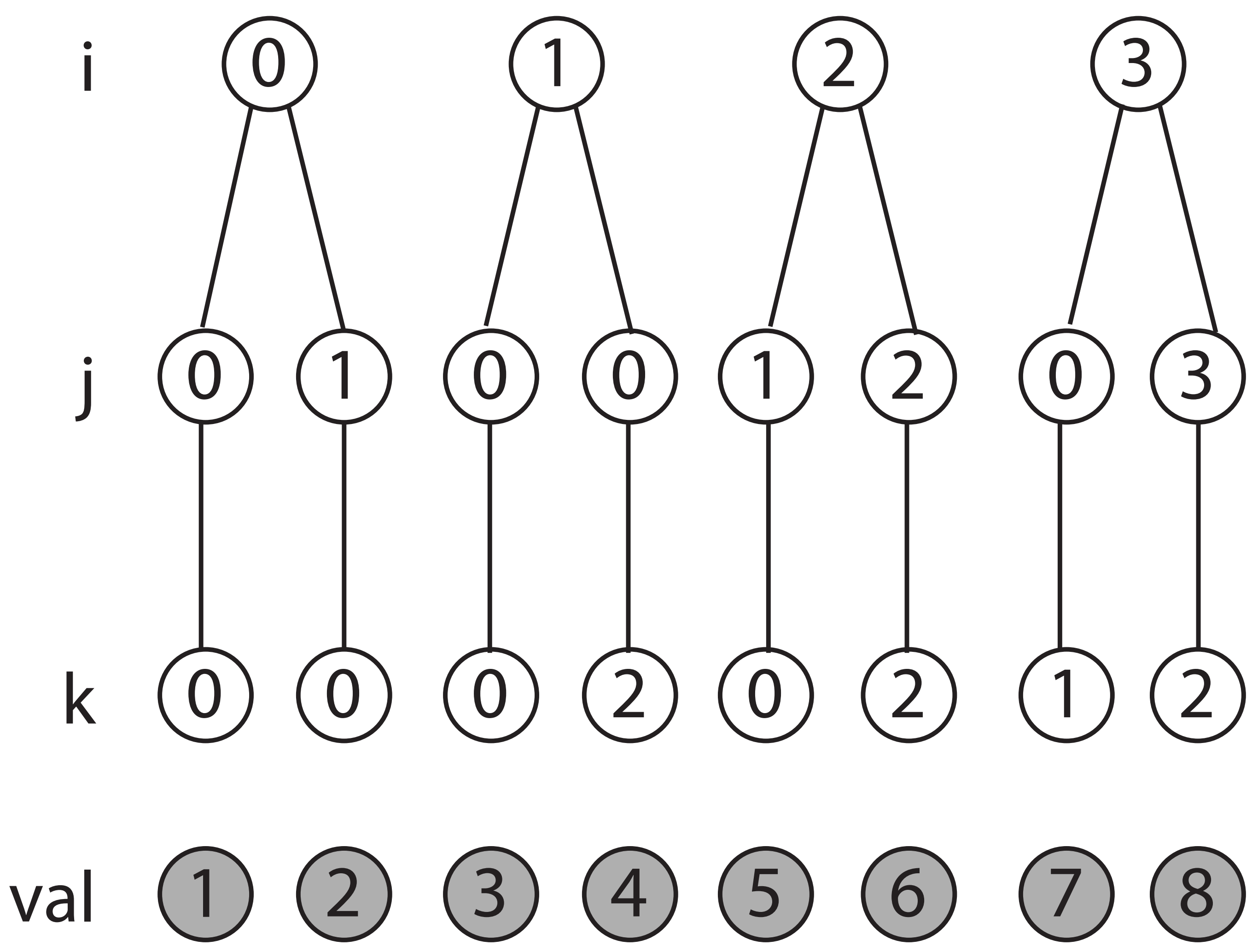
Baseline Sparse Tensor Formats in This Work

- COO: coordinate formats [Bader et al., 2006]
- CSF: Compressed Sparse Fibers, extension of CSR. [Smith et al. 2015]



i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

(a) COO



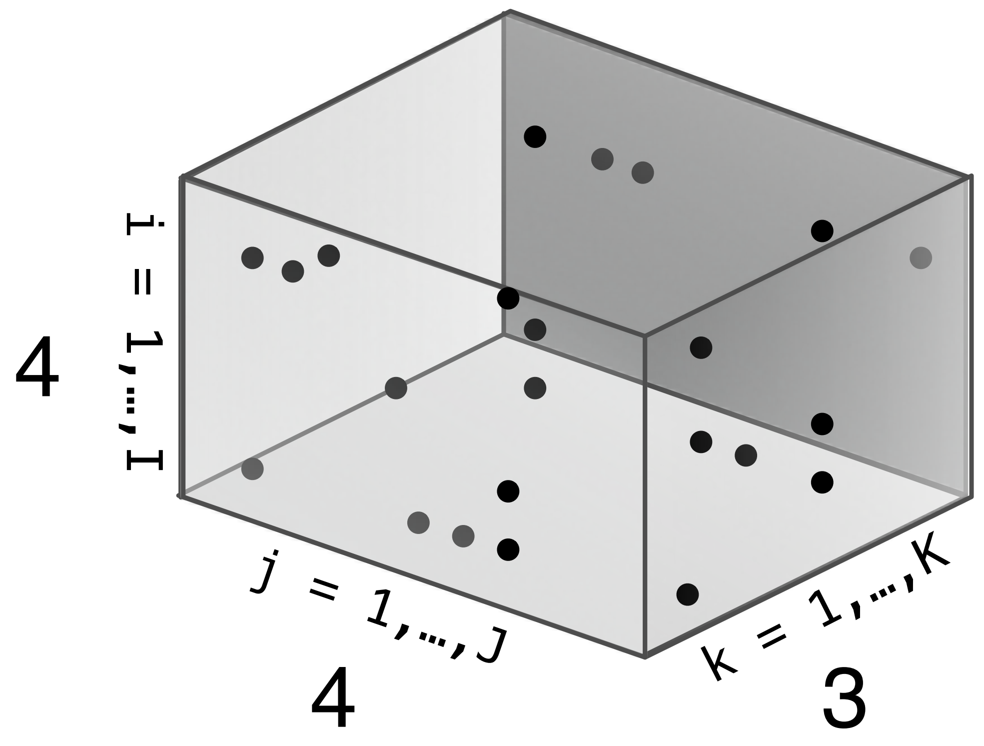
(b) CSF

Mode-Generic

Mode-Specific
prefer different representations for different modes.

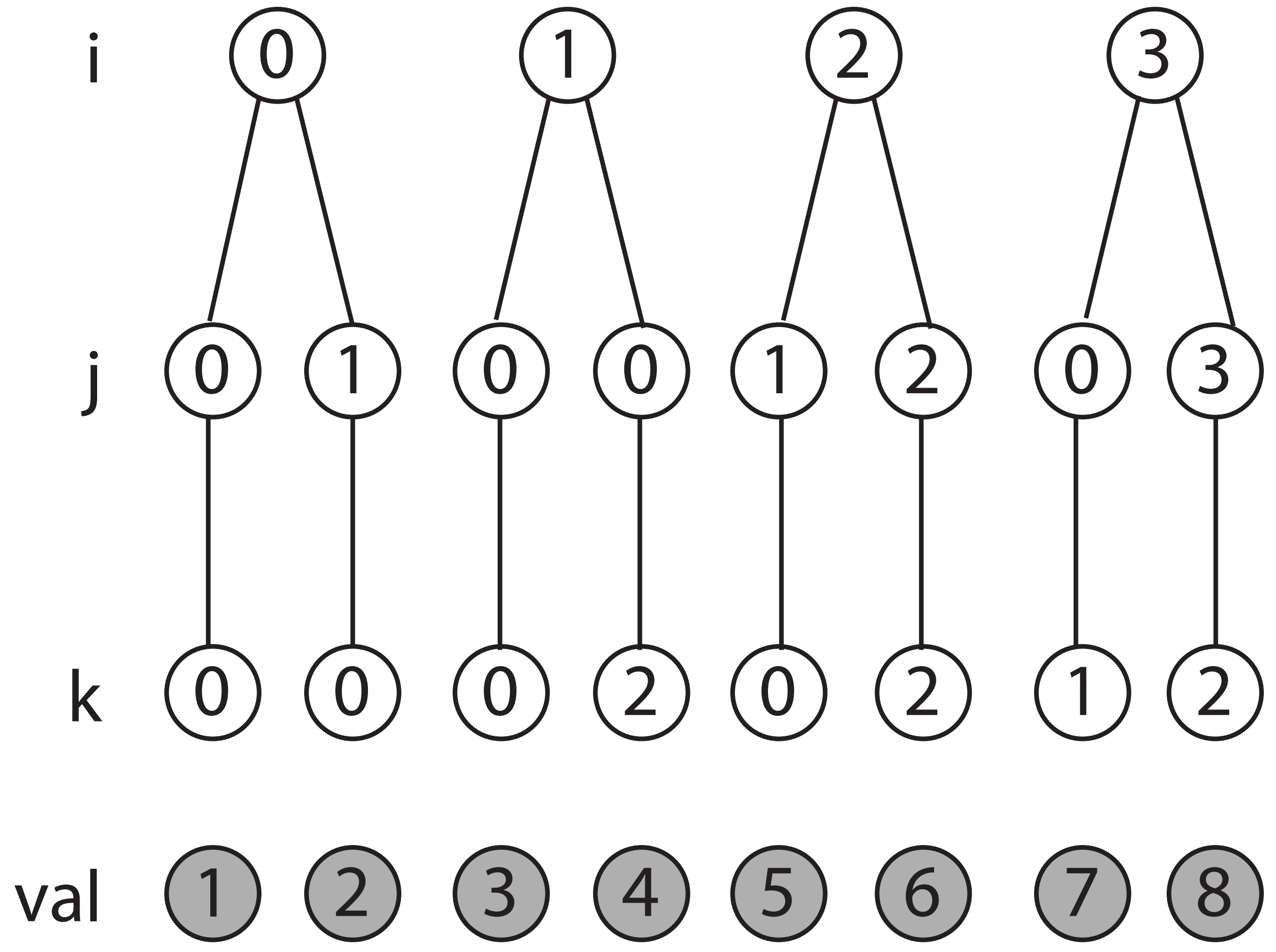
Baseline Sparse Tensor Formats in This Work

- COO: coordinate formats [Bader et al., 2006]
- CSF: Compressed Sparse Fibers, extension of CSR. [Smith et al. 2015]
- F-COO: Flagged COO format [Liu et al., 2017]



i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

(a) COO



(b) CSF

	bf	j	k	val
sf[0]=1	1	0	0	1
	0	1	0	2
	1	0	0	3
	0	0	2	4
sf[1]=1	1	1	0	5
	0	2	2	6
	1	0	1	7
	0	3	2	8

(c) F-COO

Mode-Generic

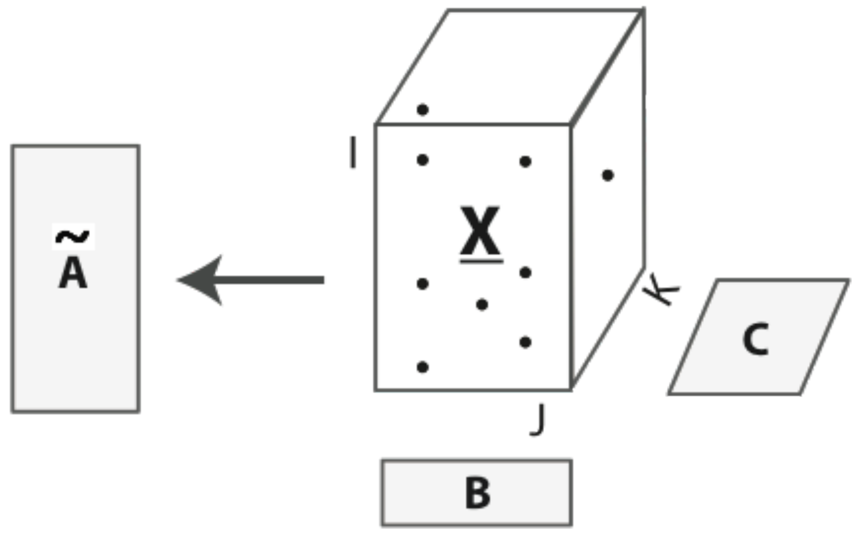
Mode-Specific
prefer different representations for different modes.

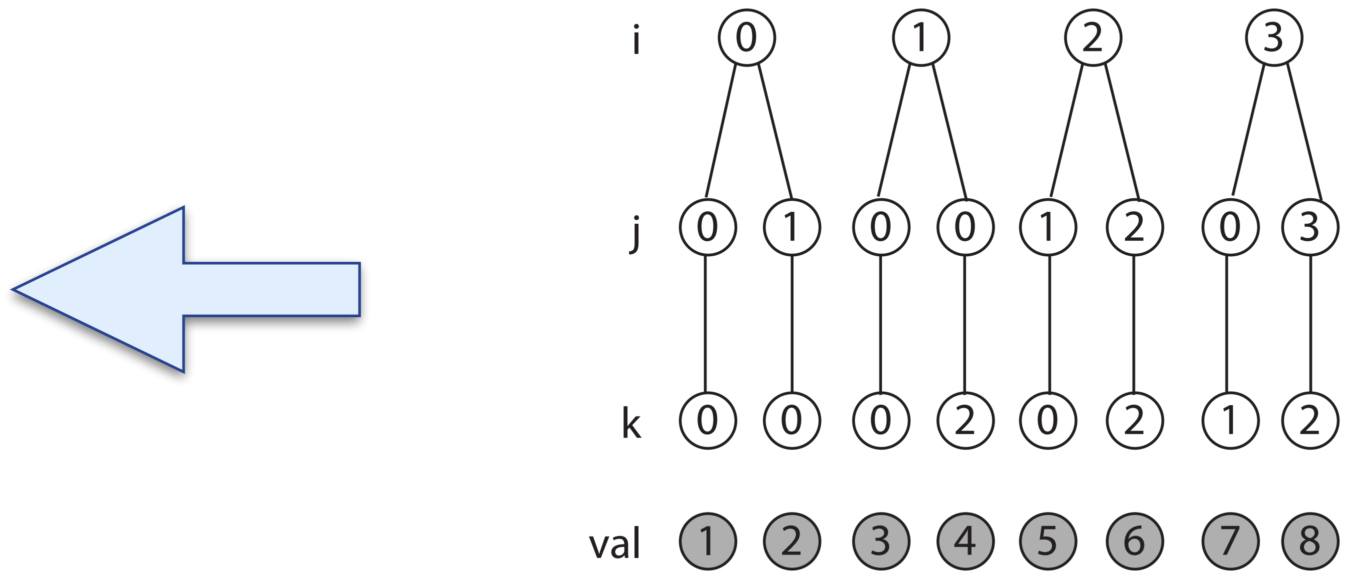
Mode-Specific Tensor Formats

- Three CSF/F-COO representations are required/preferred for three kernels.

Tensor Decomposition

Kernel in Mode-1

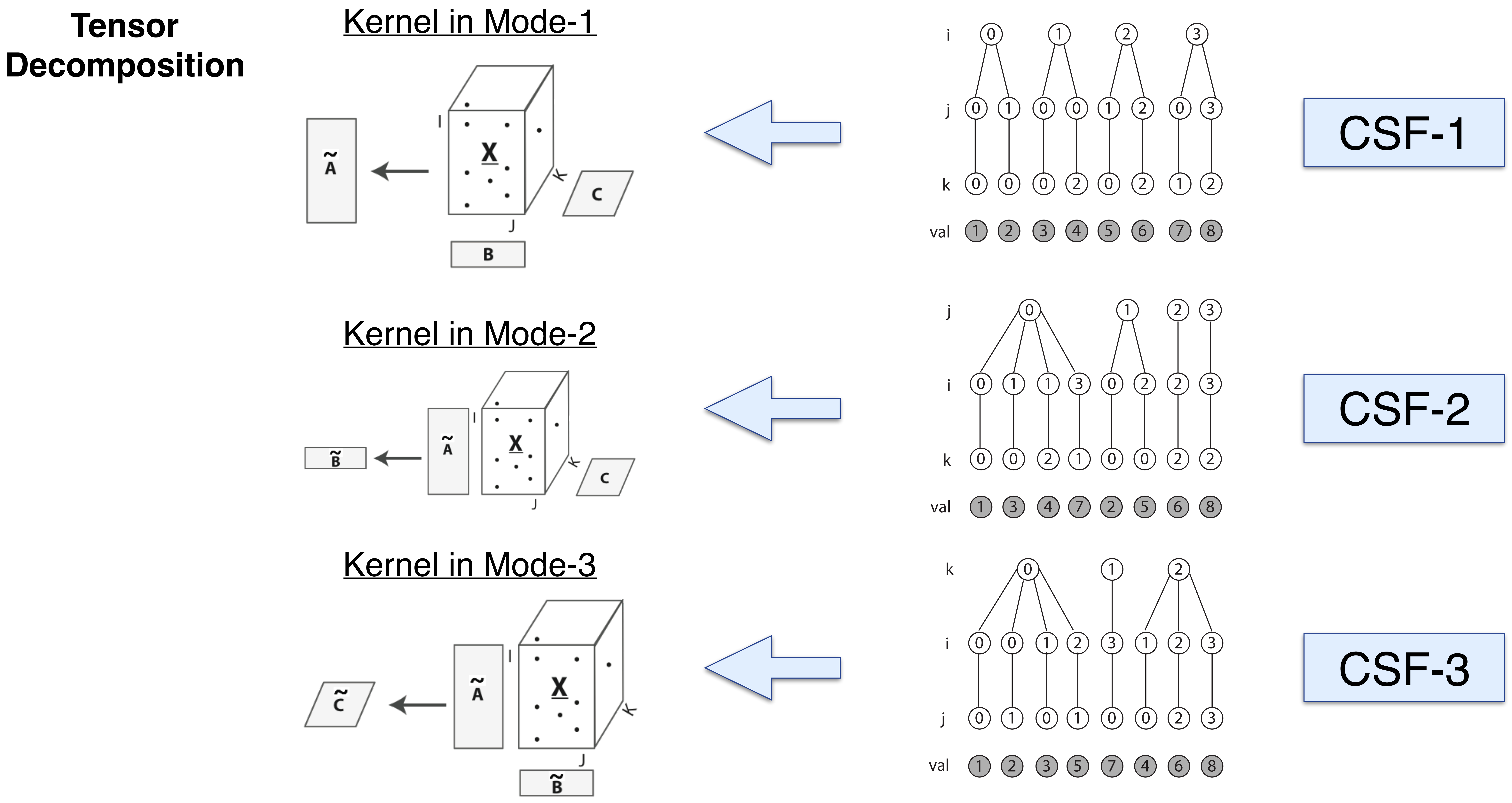




CSF-1

Mode-Specific Tensor Formats

- Three CSF/F-COO representations are required/preferred for three kernels.

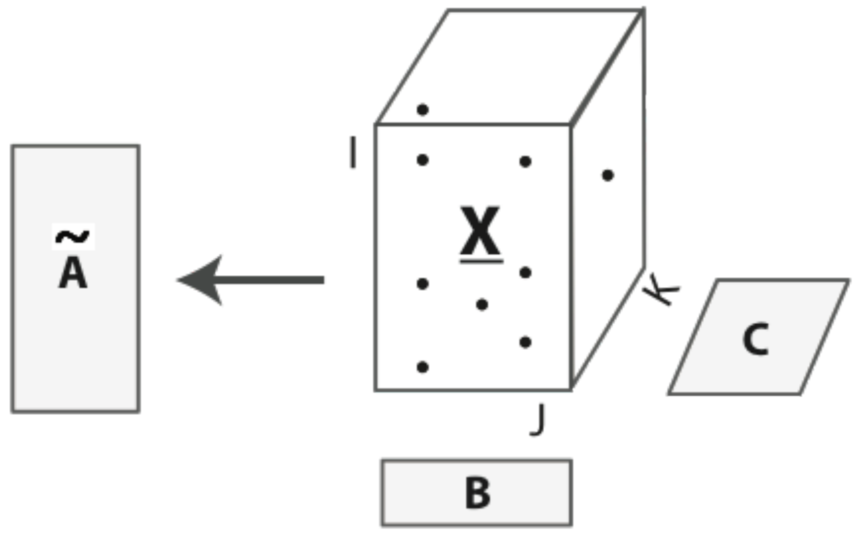


Mode-Specific Tensor Formats

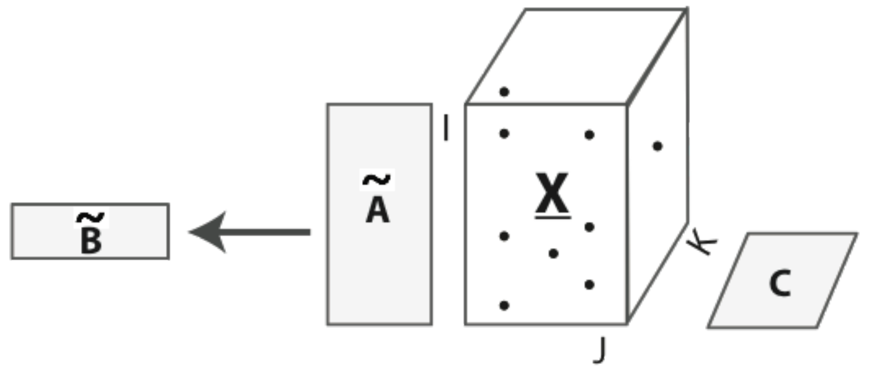
- Three CSF/F-COO representations are required/preferred for three kernels.

Tensor Decomposition

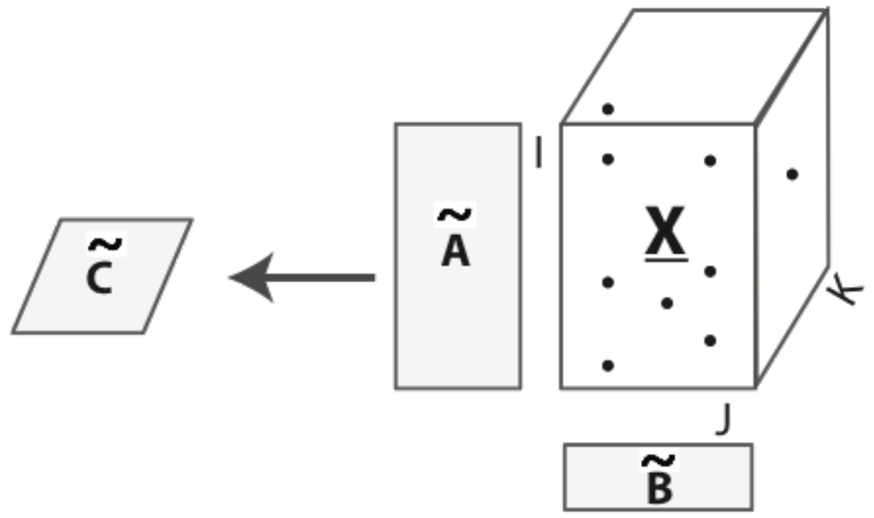
Kernel in Mode-1

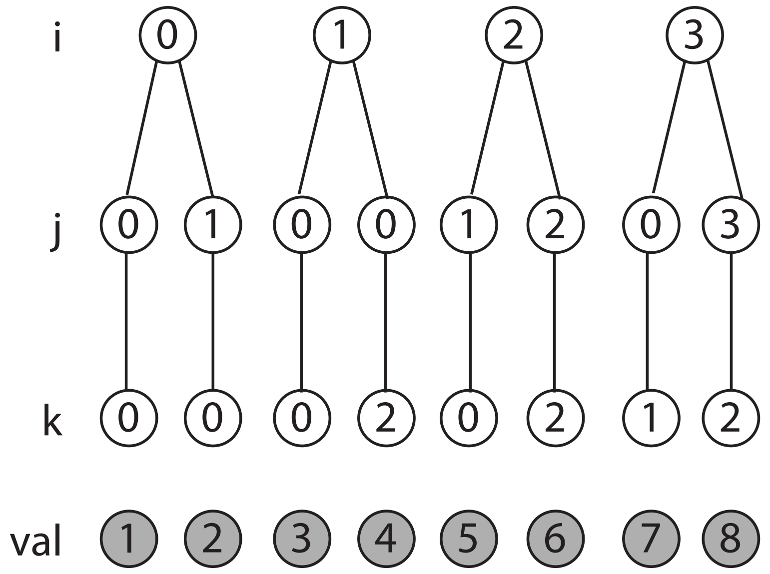


Kernel in Mode-2



Kernel in Mode-3

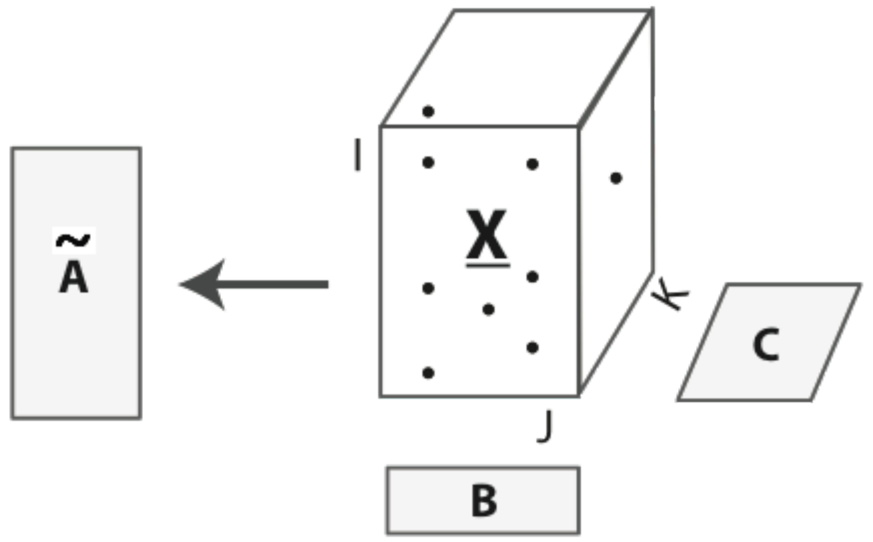
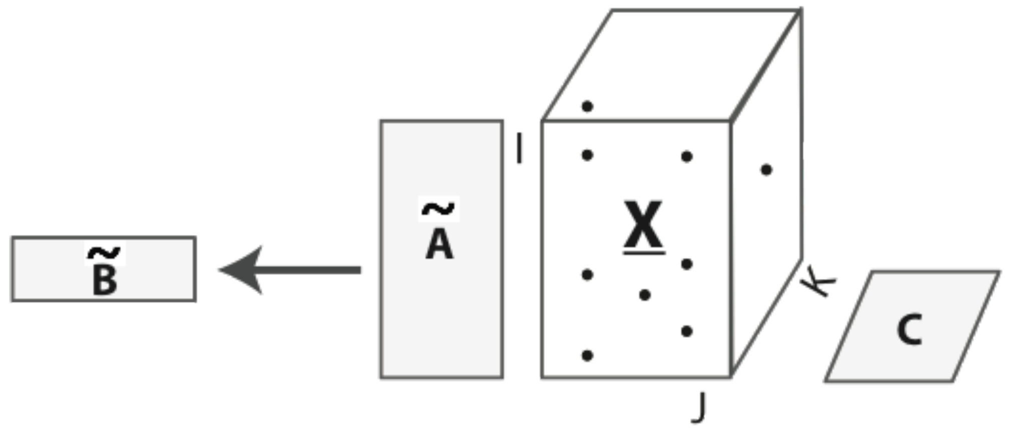
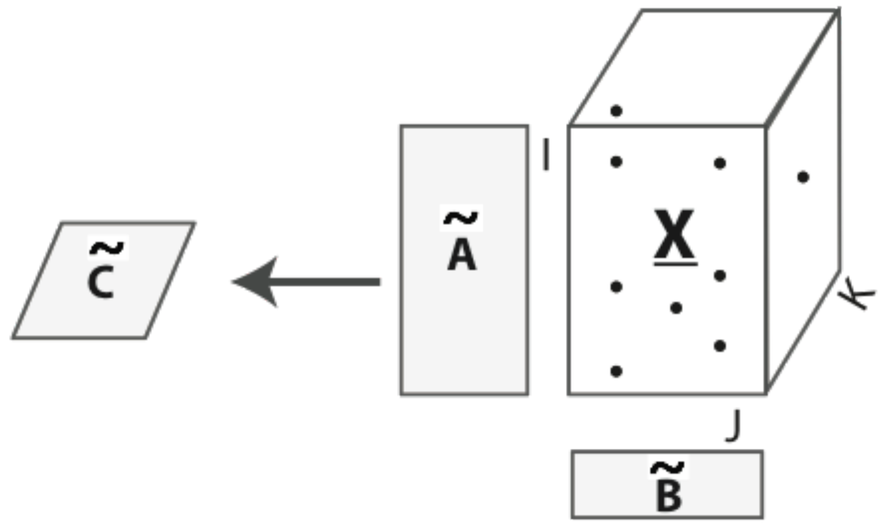




CSF-1

Performance payoff

Mode Orientation

Tensor decomposition	Mode-Specific	Mode-Generic	
	Mode-1 oriented (CSF/FCOO)	Coordinate (COO)	**HiCOO**
Kernel in Mode-1 	✓	✗	✓
Kernel in Mode-2 	✗	✗	✓
Kernel in Mode-3 	✗	✗	✓

✓ Efficient ✗ In-efficient

HiCOO Format

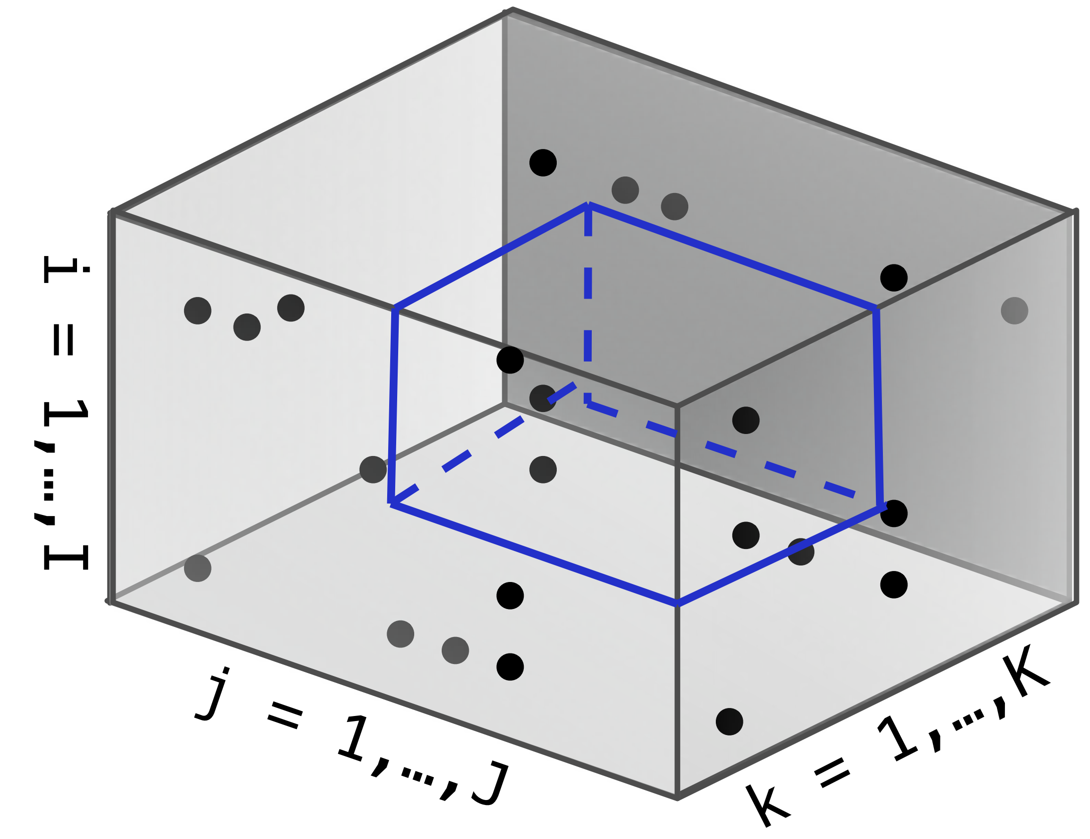
- Store a sparse tensor in units of small sparse blocks

i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

COO

	bptr	bi	bj	bk	ei	ej	ek	val
B1	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

HiCOO



Block size: 2*2*2

Extension from Compressed Sparse Blocks (CSB) format by Buluc et al. SPAA. 2009.

HiCOO Format

- Store a sparse tensor in units of small sparse blocks

i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

COO

	bptr	bi	bj	bk	ei	ej	ek	val
	0	0	0	0	0	0	0	1
B1					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

HiCOO

Extension from Compressed Sparse Blocks (CSB) format by Buluc et al. SPAA. 2009.

HiCOO Format

- Store a sparse tensor in units of small sparse blocks
  - Shorten the bit-length of element indices

block indices | element indices

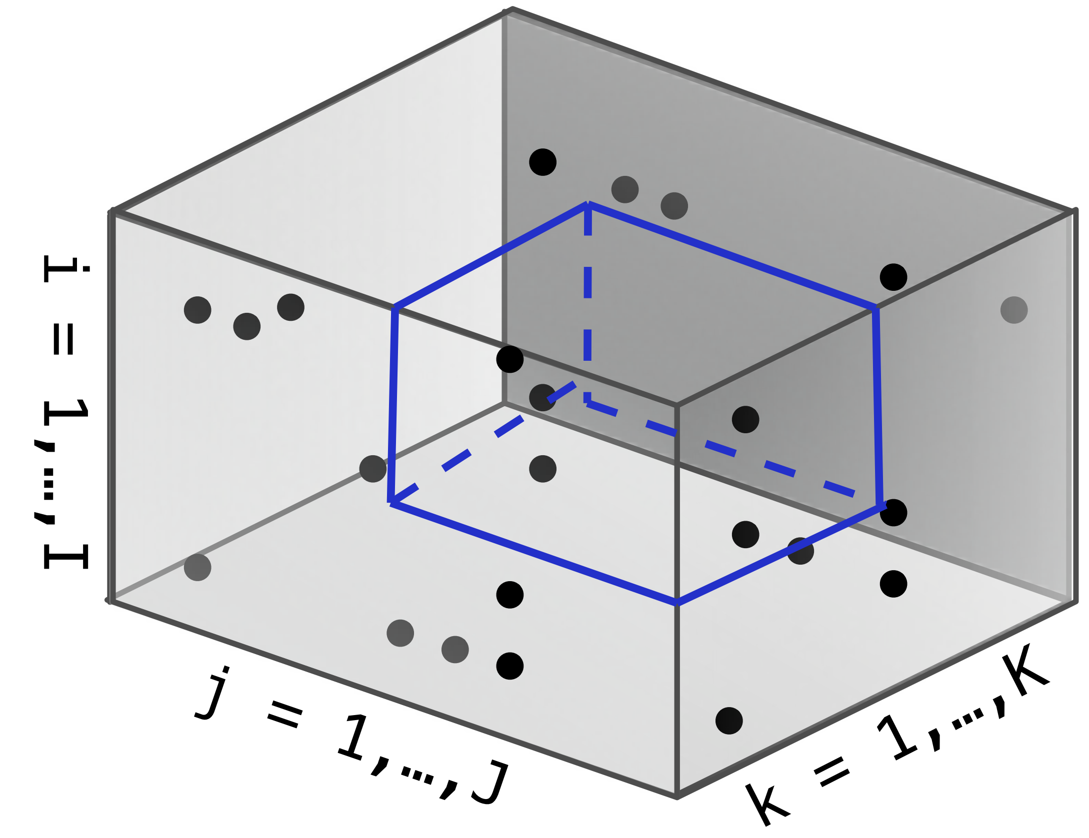
COO (32-bit):

i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

HiCOO (block indices: 32-bit; element indices: **8-bit**):

	bptr	bi	bj	bk	ei	ej	ek	val
B1	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

i = bi * B + ei



Block size: 2*2*2

HiCOO Format

- Store a sparse tensor in units of small sparse blocks
  - Shorten the bit-length of element indices
  - Compress the number of block indices

COO (32-bit)

i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

HiCOO (block indices: 32-bit; element indices: 8-bit)

	bptr	bi	bj	bk	ei	ej	ek	val
B1	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

HiCOO Format

- Store a sparse tensor in units of small sparse blocks
  - Shorten the bit-length of element indices
  - Compress the number of block indices

COO (i, j, k: 32-bit)

i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

HiCOO (block indices bi, bj, bk: 32-bit; element indices ei, ej, ek: 8-bit)

	bptr	bi	bj	bk	ei	ej	ek	val
B1	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

i = bi * B + ei

COO indices:
= nnz * 3 * 32

HiCOO indices:
= nnz * 3 * **8** + **nnb** * (3 * 32 + 32)

nnz: #Nonzeros; nnb: #Non-zero blocks

HiCOO Format

- Store a sparse tensor in units of small sparse blocks
  - Shorten the bit-length of element indices
  - Compress the number of block indices
  - For arbitrary-order sparse tensors.

COO (32-bit)

i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

HiCOO (bptr, bi, bj, bk: 32-bit; ei, ej, ek: **8-bit**)

	bptr	bi	bj	bk	ei	ej	ek	val
B1	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

For the tensor: Reduce its storage and memory footprints

For matrices: Better data locality

Format Conversion

$|— \beta_{int} —| \beta_{float}$

i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

COO

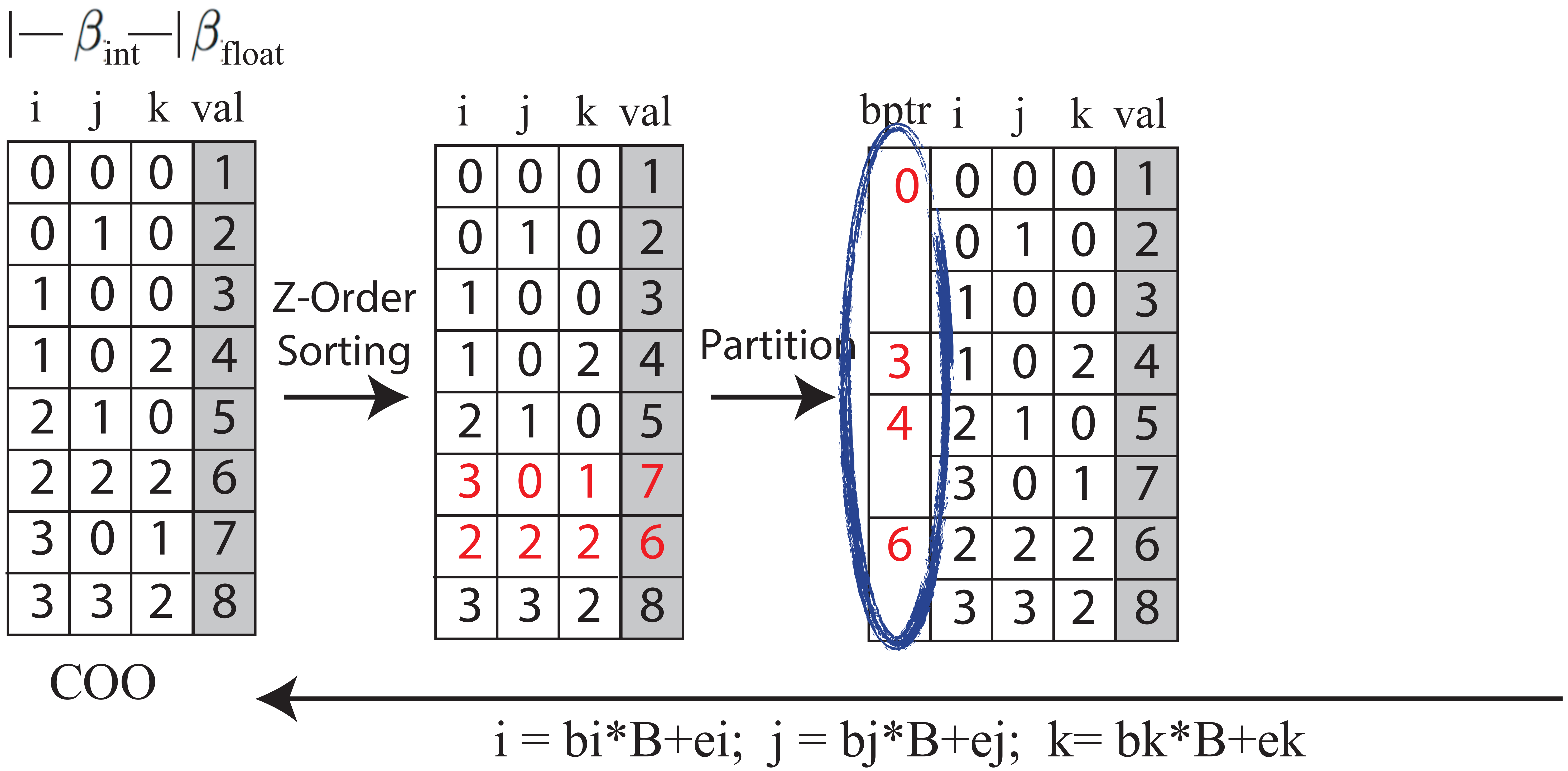
Z-Order Sorting →

i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
3	0	1	7
2	2	2	6
3	3	2	8

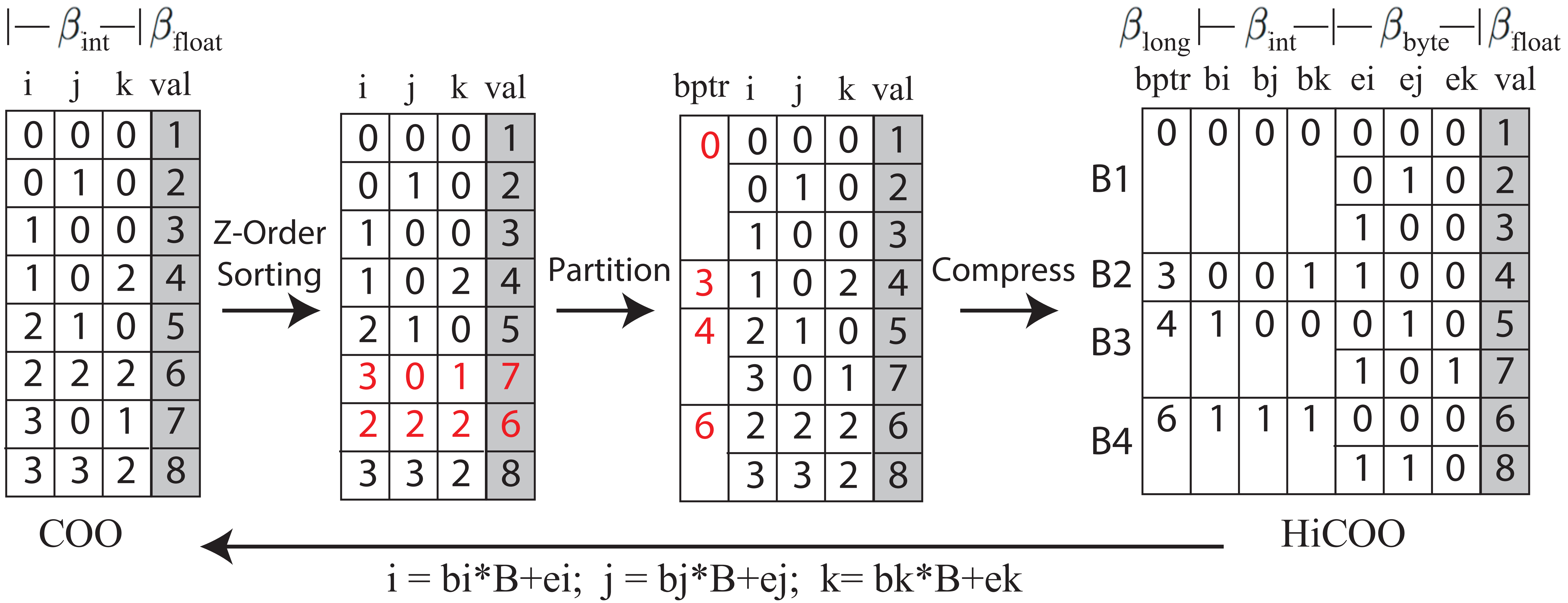
←

i = bi*B+ei; j = bj*B+ej; k= bk*B+ek

Format Conversion



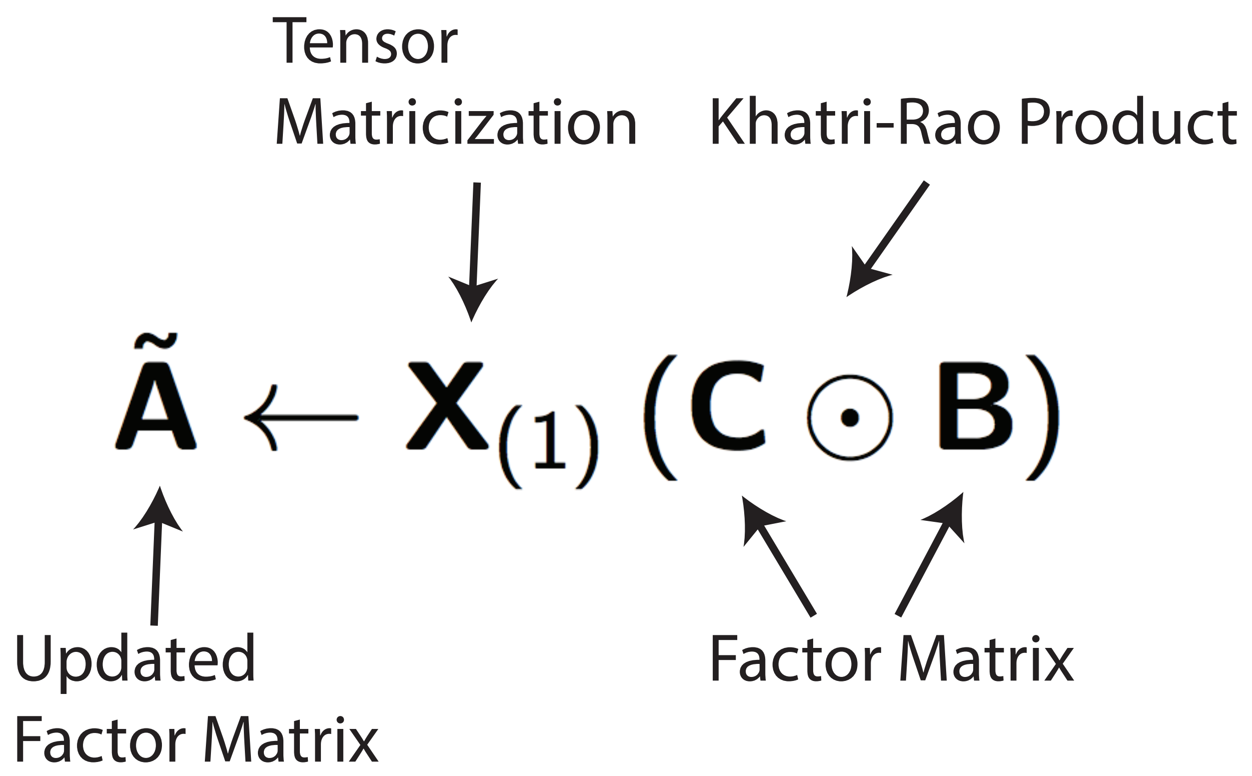
Format Conversion

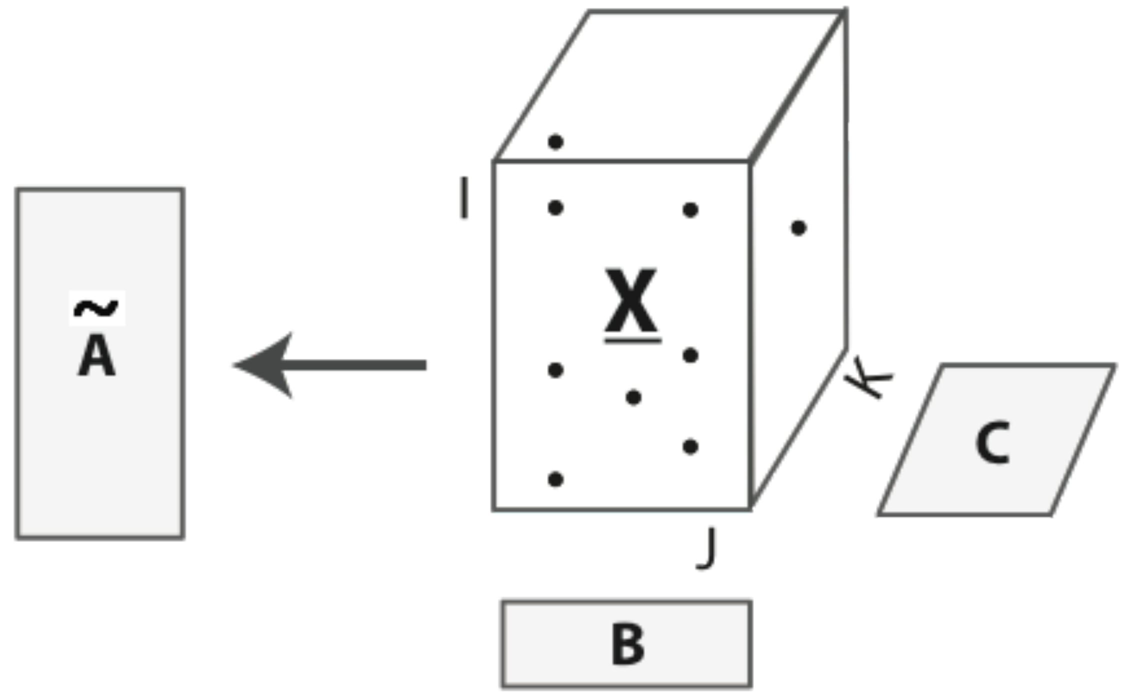


MTTKRP Operation

- Matriced Tensor Times Khatri-Rao Product (MTTKRP)

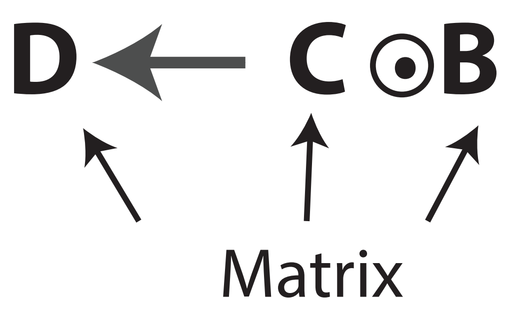
$$\tilde{\mathbf{A}} \leftarrow \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B})$$

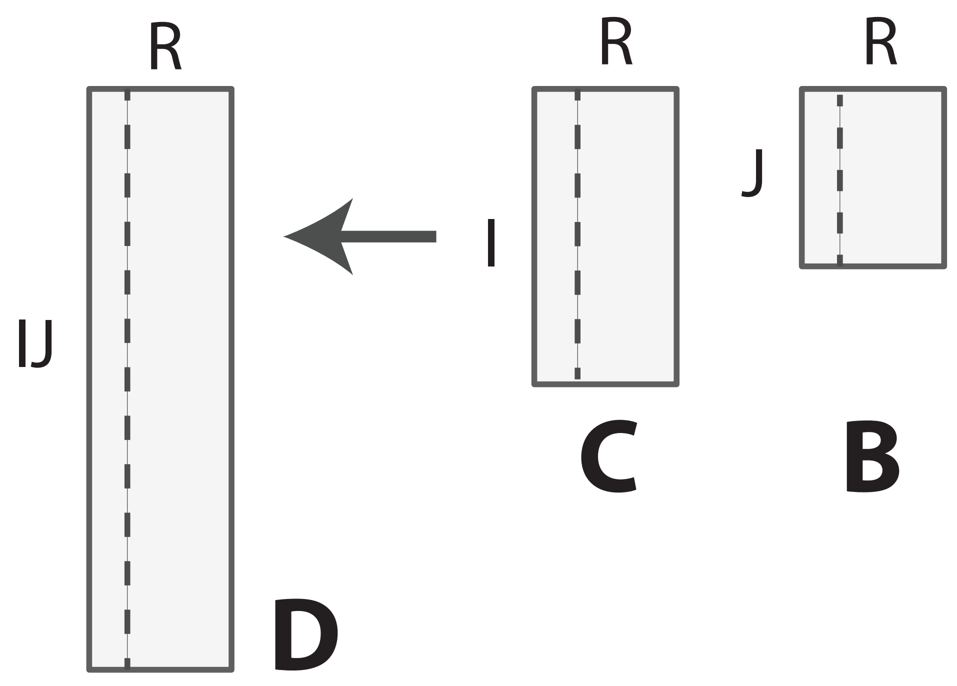




- Khatri-Rao Product

$$\mathbf{D} \leftarrow \mathbf{C} \odot \mathbf{B}$$





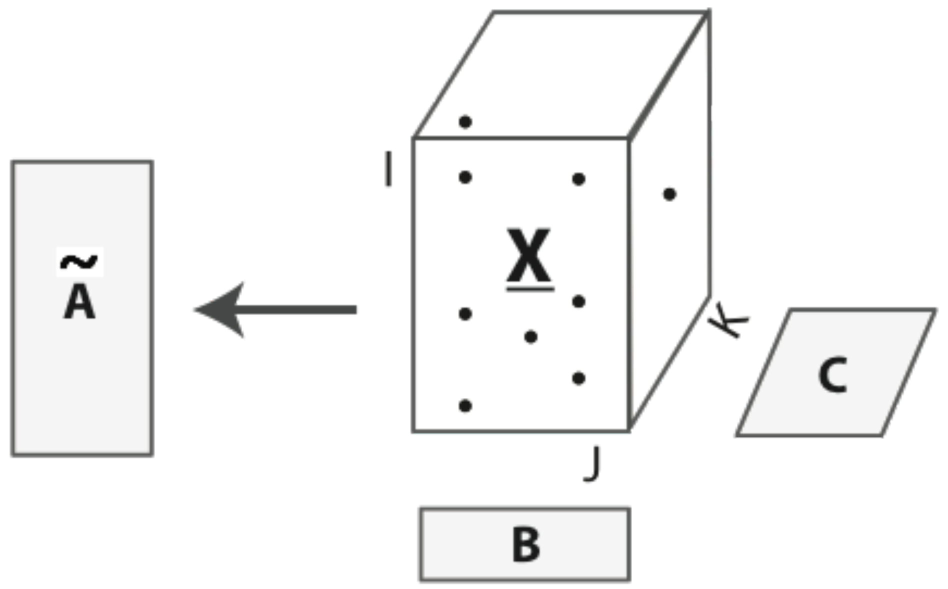
MTTKRP is the performance bottleneck of CP decomposition.

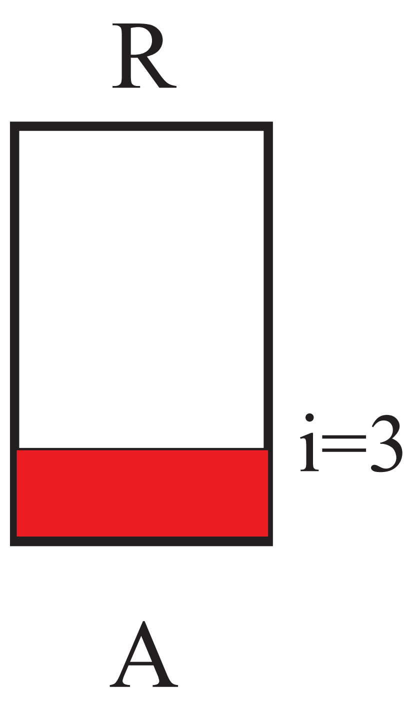
COO-MTTKRP

Tensor Matricization — Khatri-Rao Product

$$\tilde{\mathbf{A}} \leftarrow \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B})$$

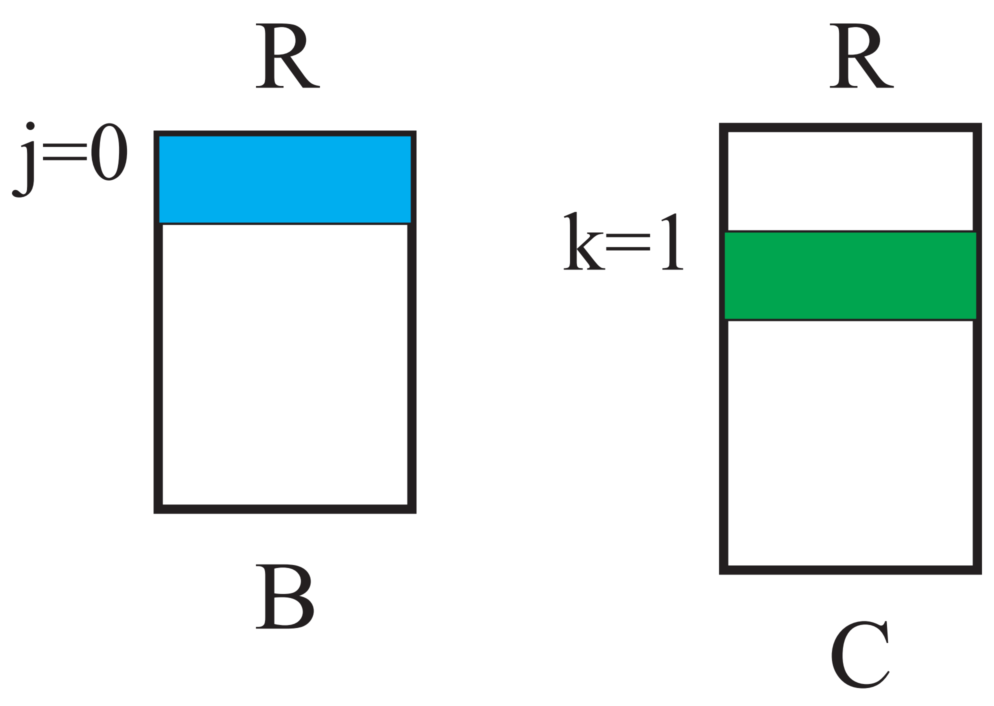
Updated Factor Matrix — Factor Matrix





i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

COO



COO-MTTKRP algorithm in mode-1

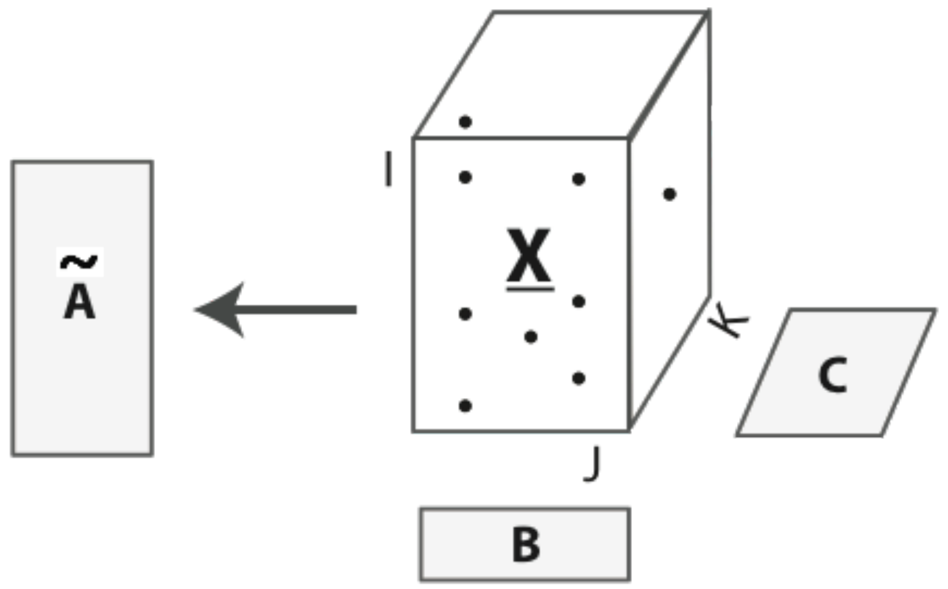
Entry-wise

COO-MTTKRP

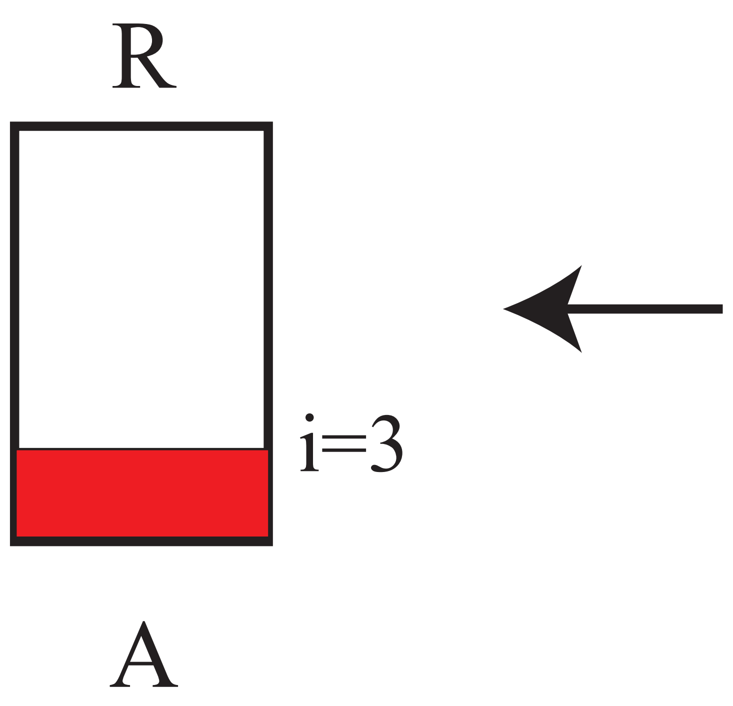
Tensor Matricization   Khatri-Rao Product

$$\tilde{\mathbf{A}} \leftarrow \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B})$$

Updated Factor Matrix   Factor Matrix

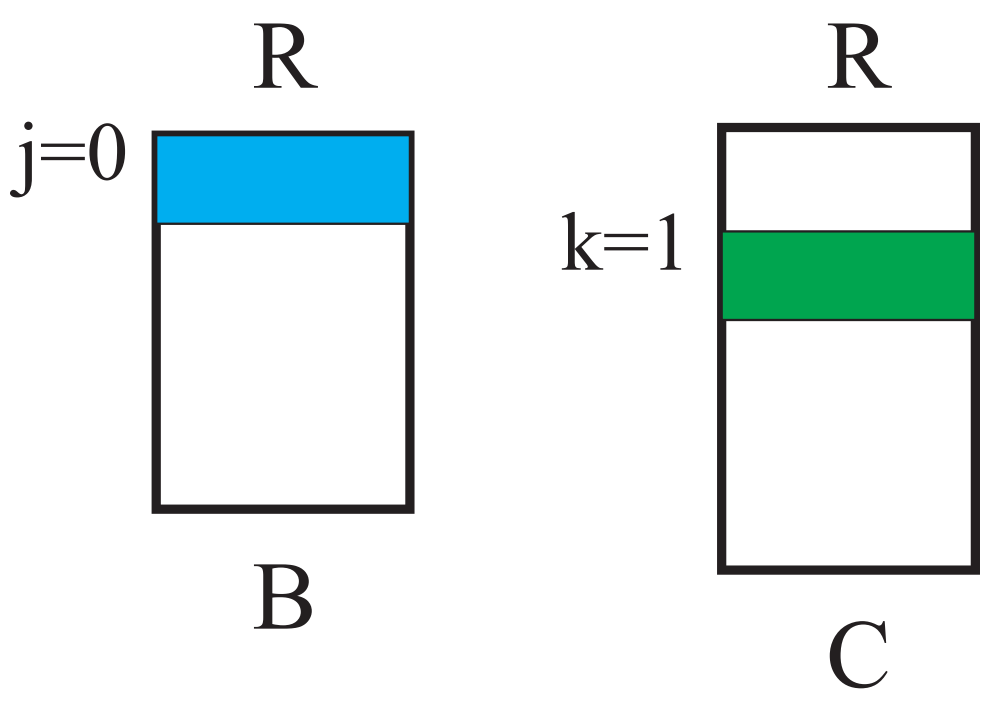


COO-MTTKRP algorithm in mode-1

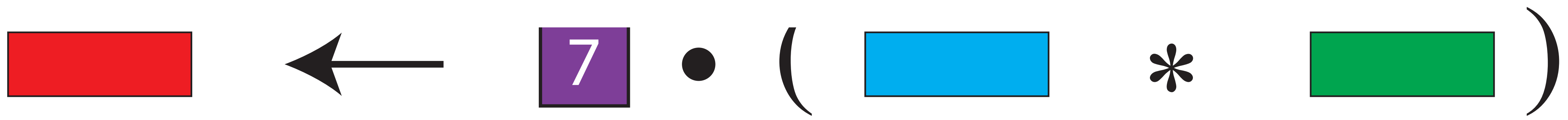


i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

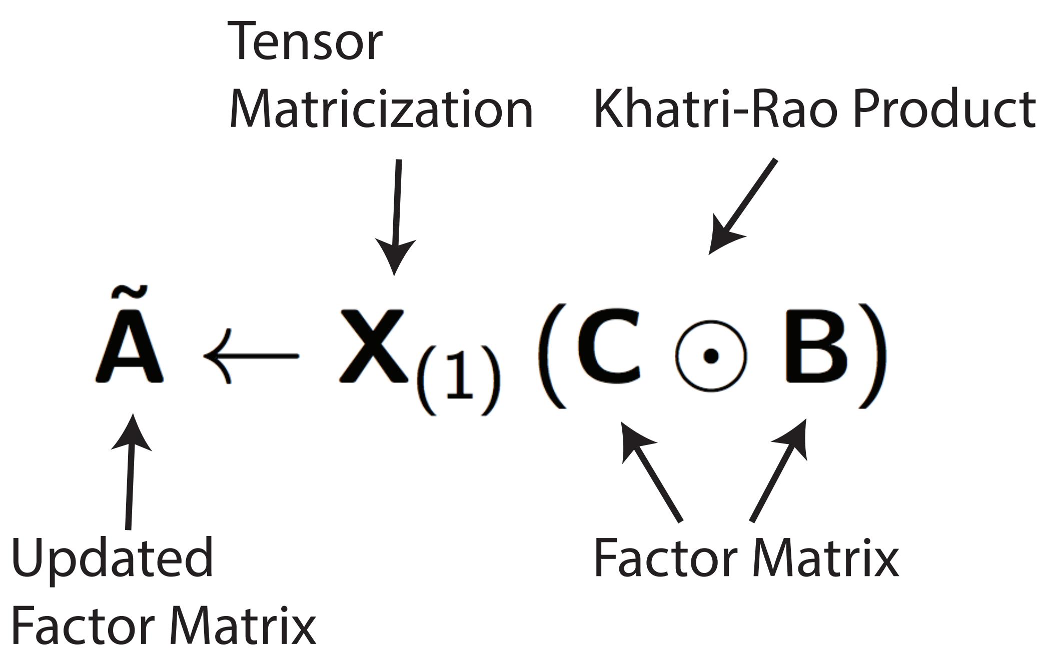
COO

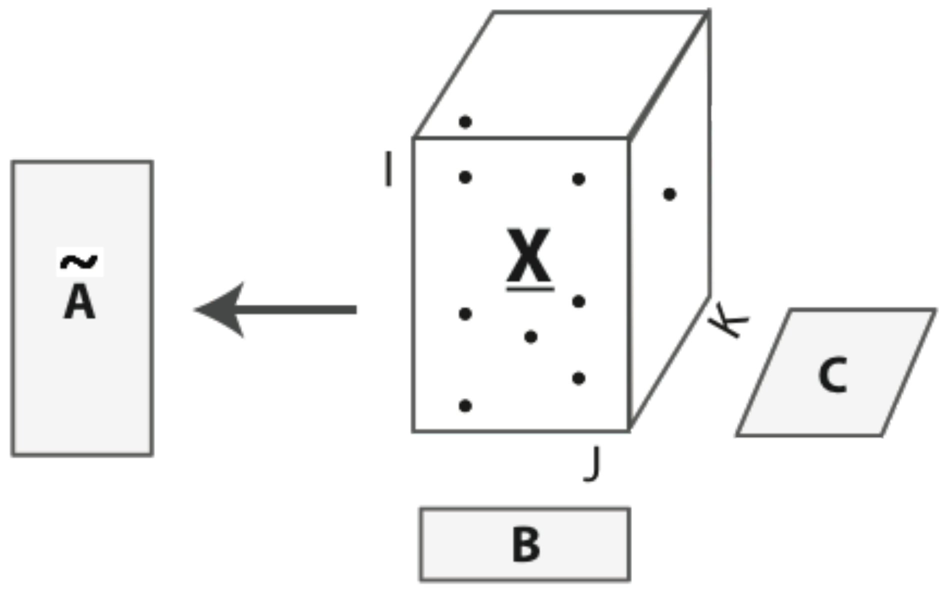


Entry-wise

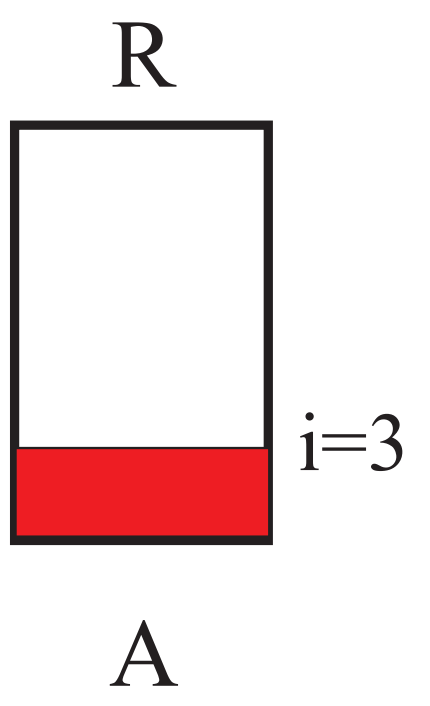


COO-MTTKRP



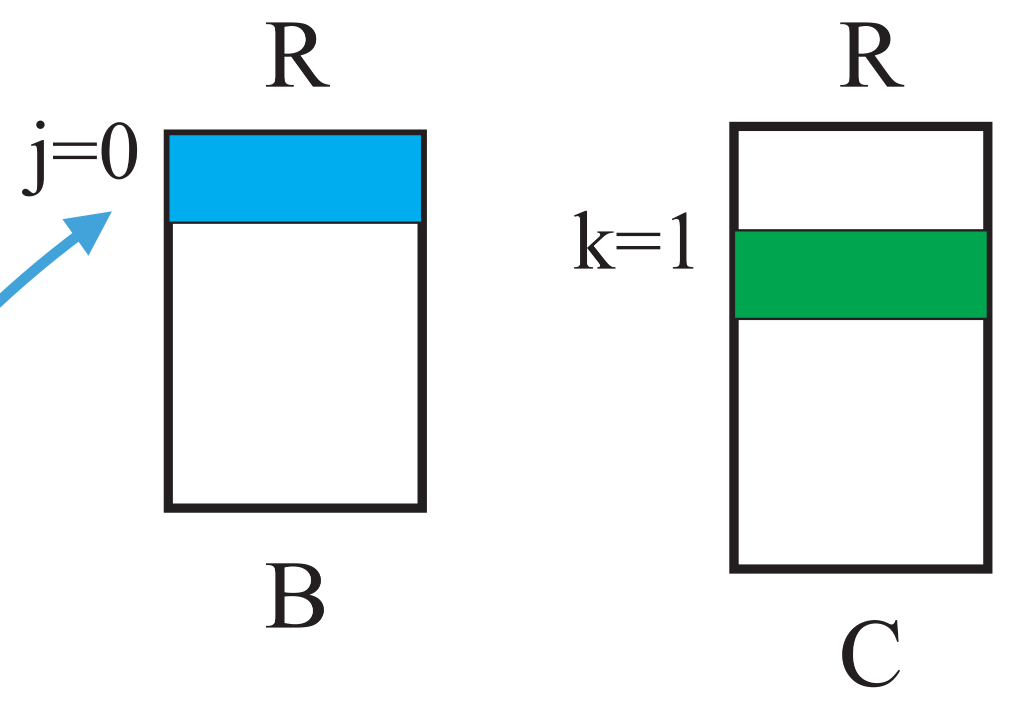


COO-MTTKRP
algorithm in mode-1



i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

COO

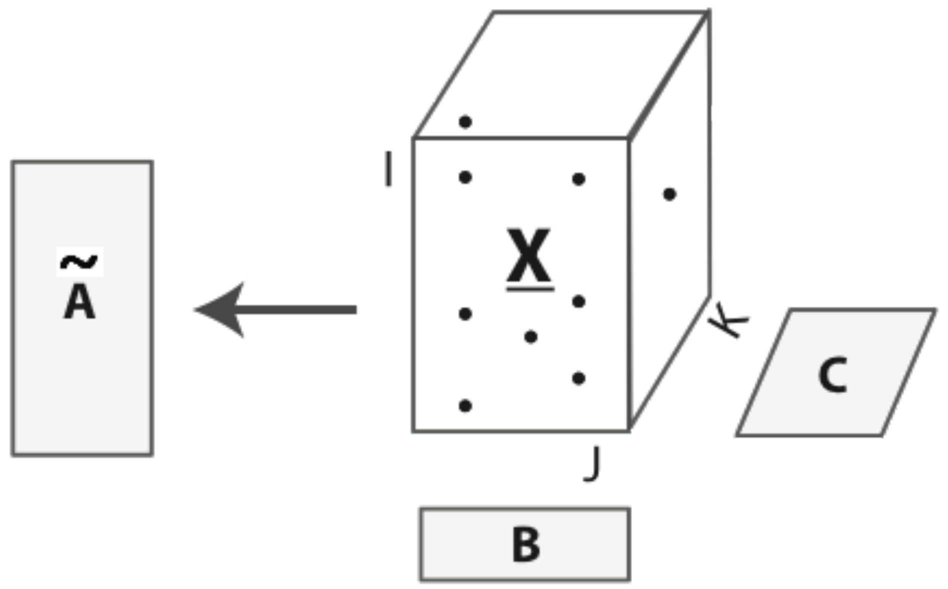


COO-MTTKRP

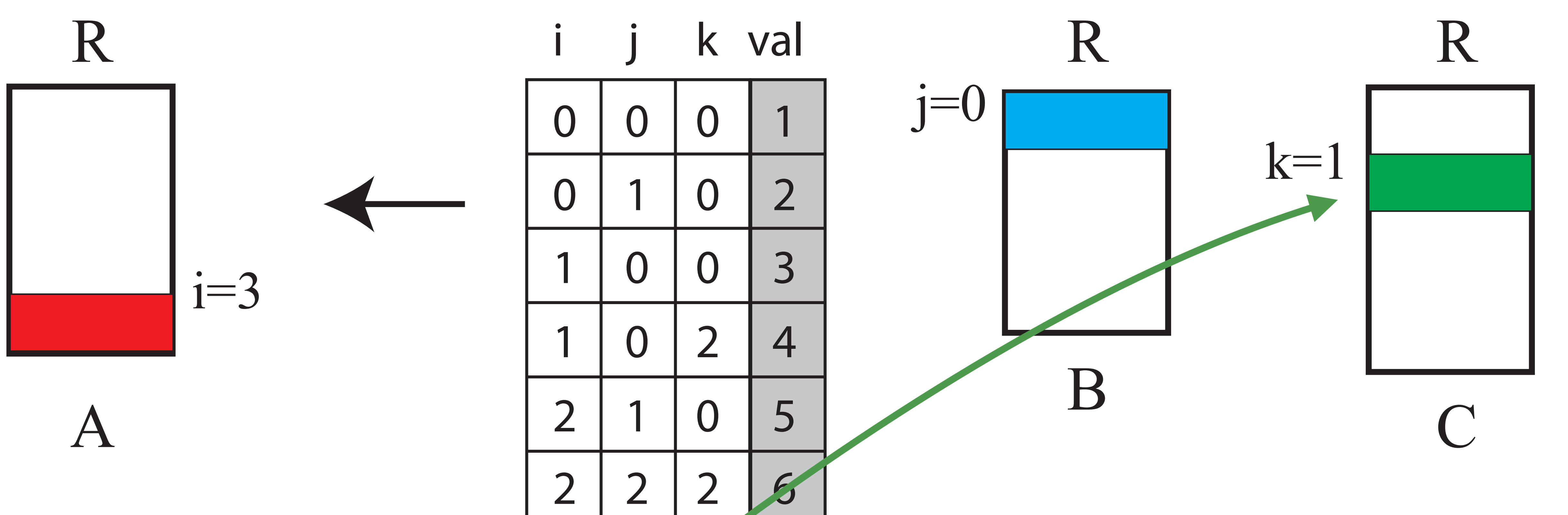
Tensor Matricization — Khatri-Rao Product

$$\tilde{\mathbf{A}} \leftarrow \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B})$$

Updated Factor Matrix ($\tilde{\mathbf{A}}$) — Factor Matrix ($\mathbf{C}$, $\mathbf{B}$)



COO-MTTKRP algorithm in mode-1

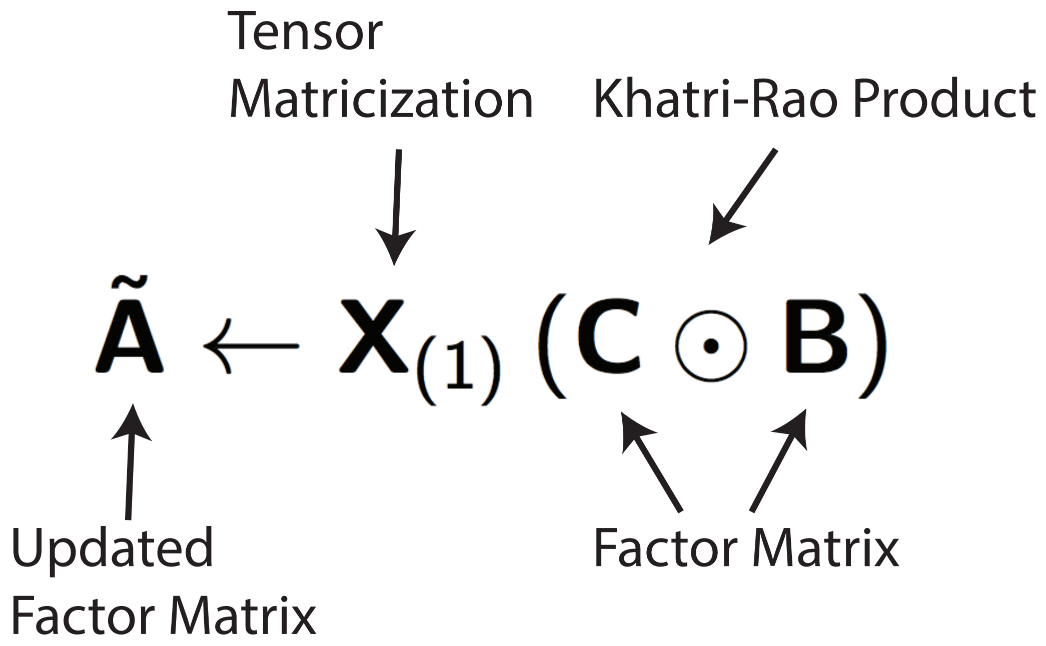


i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

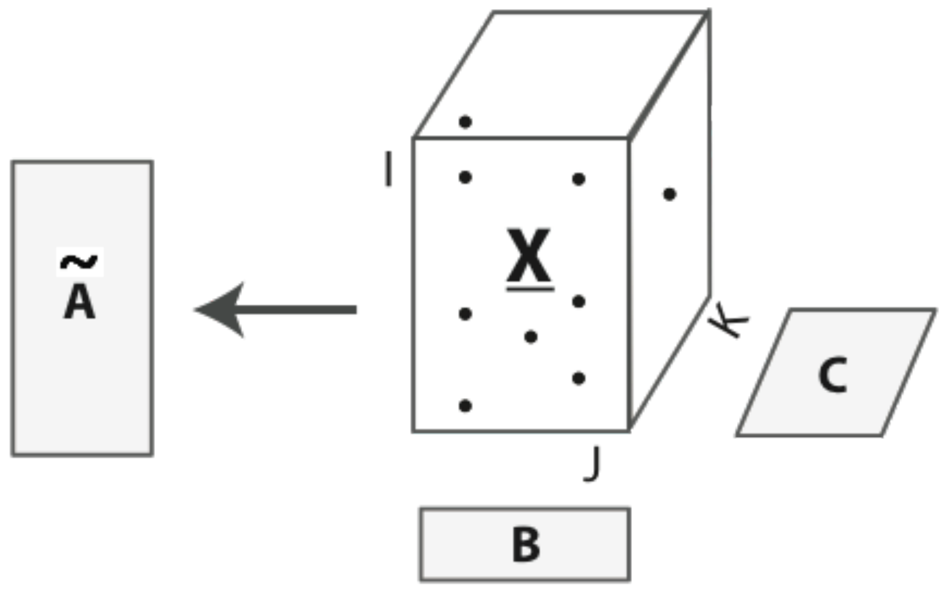
COO

Entry-wise

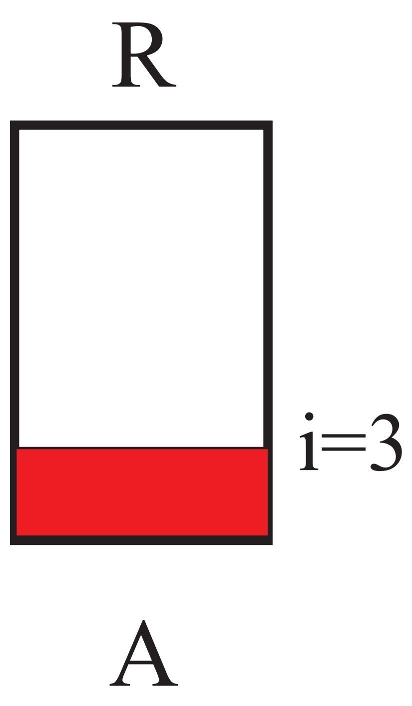
COO-MTTKRP



$$\tilde{\mathbf{A}} \leftarrow \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B})$$

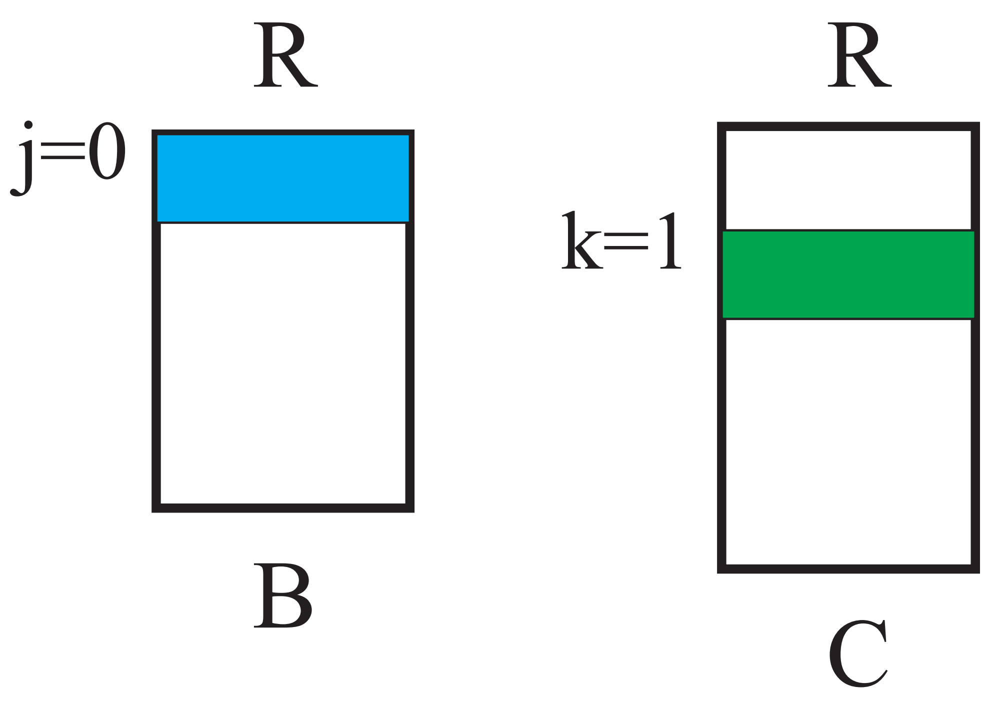


COO-MTTKRP algorithm in mode-1



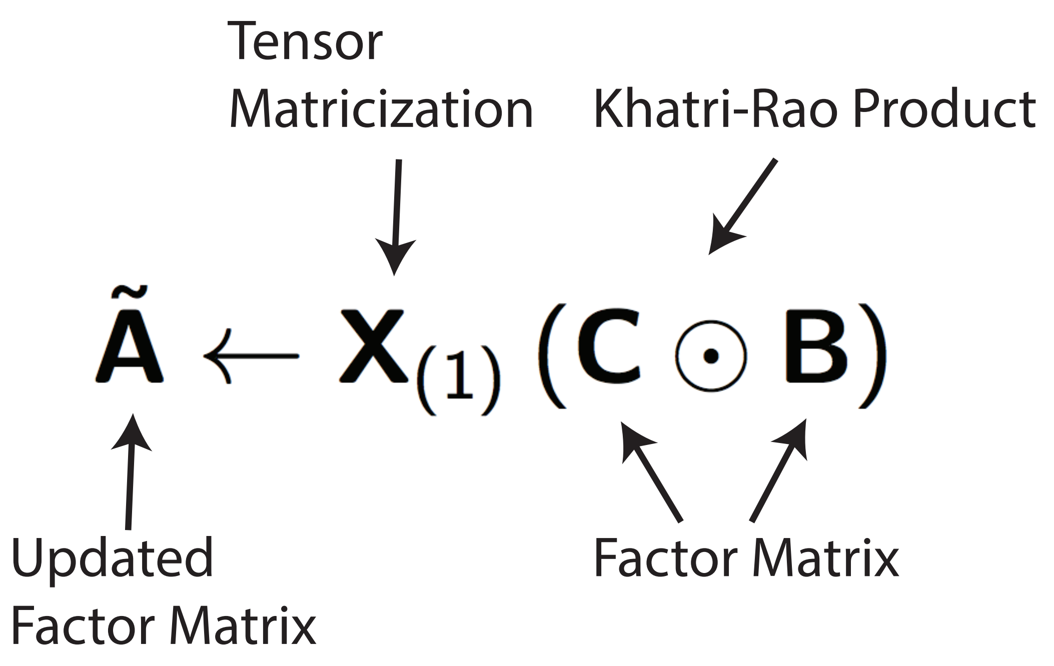
i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

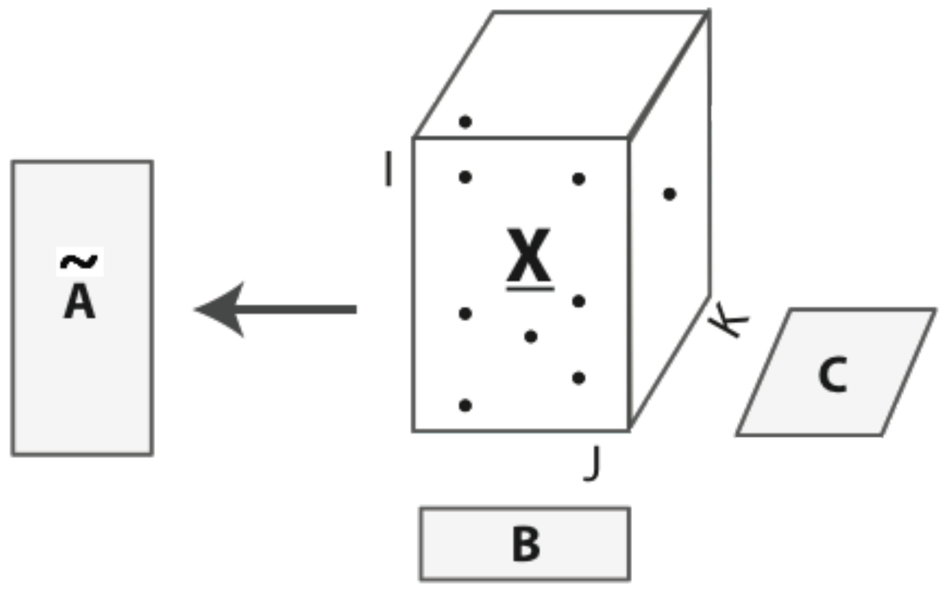
COO



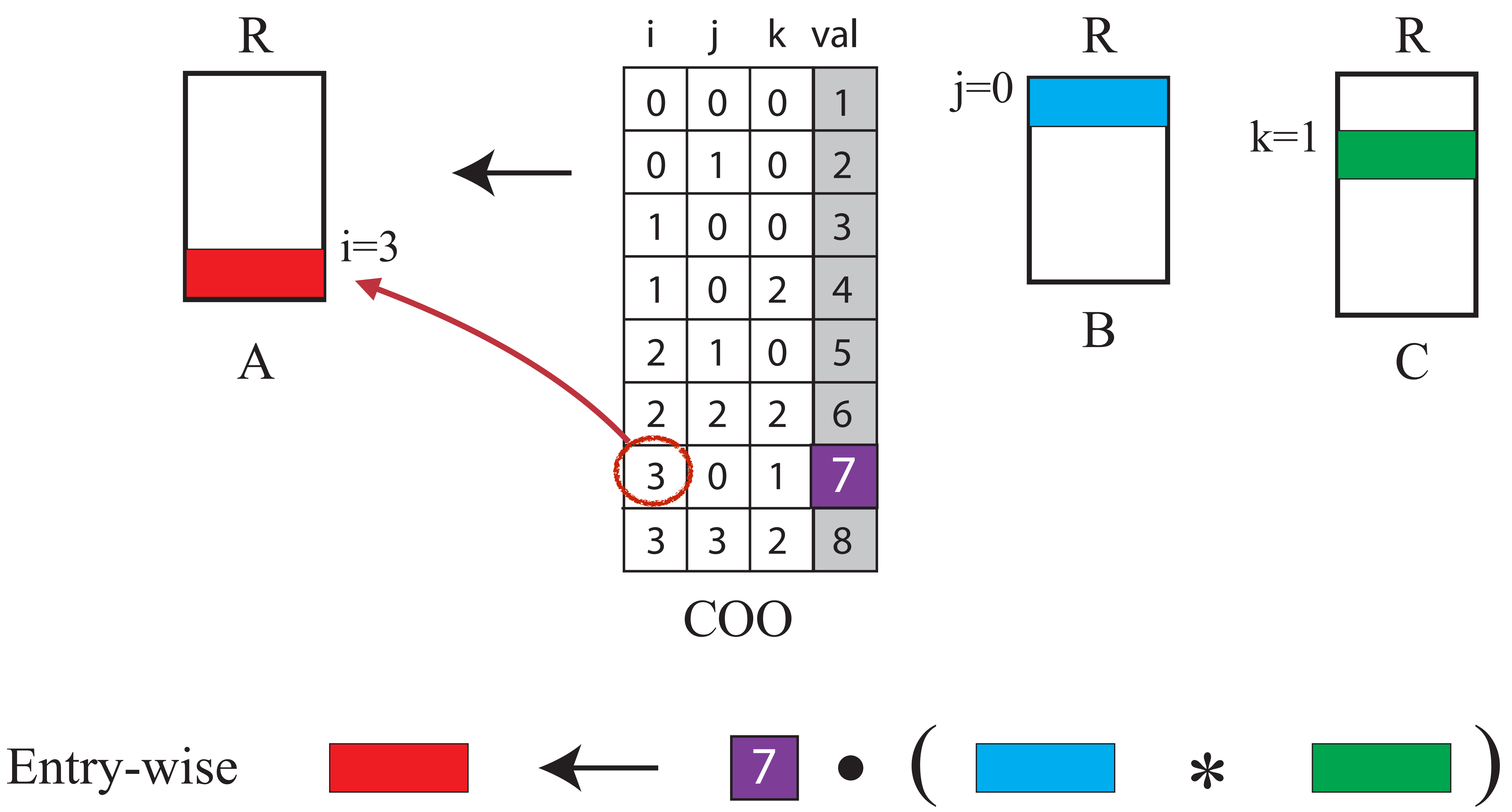
Entry-wise

COO-MTTKRP





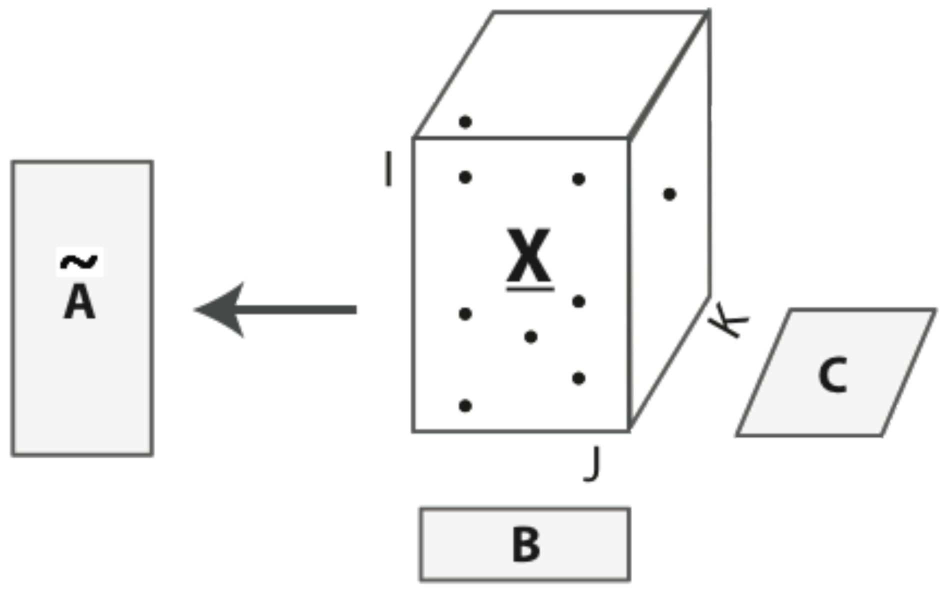
COO-MTTKRP algorithm in mode-1

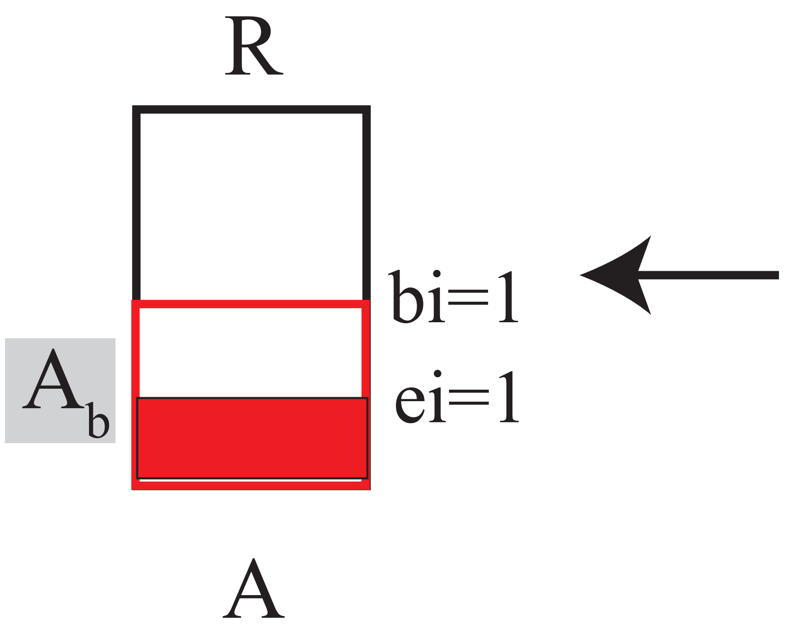


HiCOO-MTTKRP

$$\tilde{\mathbf{A}} \leftarrow \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B})$$

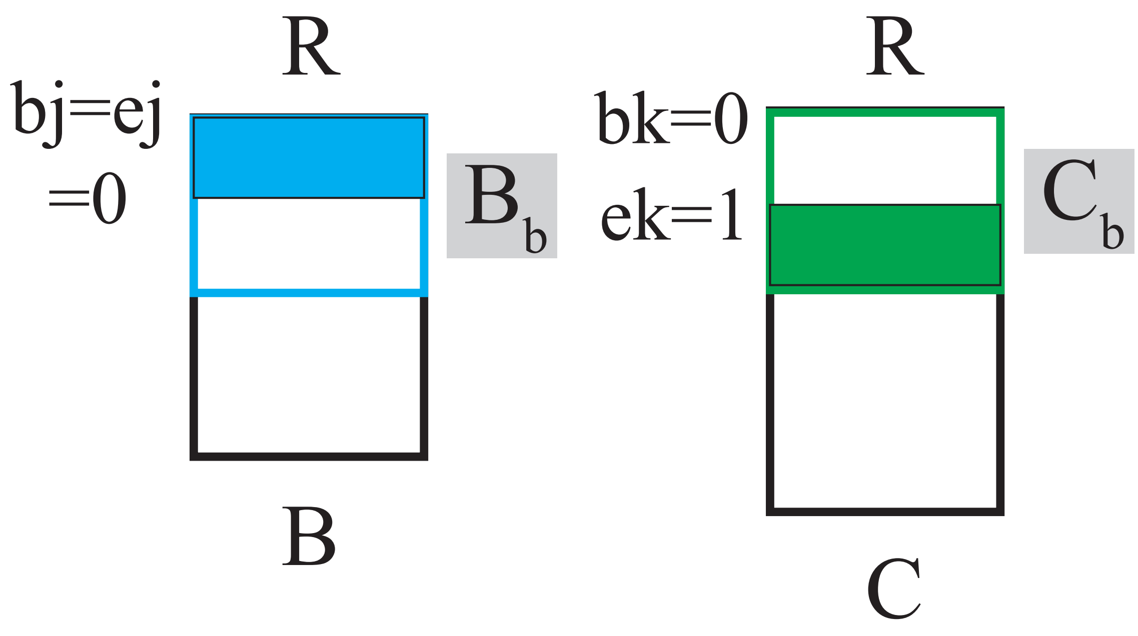
Tensor Matricization → $\mathbf{X}_{(1)}$; Khatri-Rao Product → $\odot$; Updated Factor Matrix → $\tilde{\mathbf{A}}$; Factor Matrix → $\mathbf{C}$, $\mathbf{B}$





	bptr	bi	bj	bk	ei	ej	ek	val
B1	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

HiCOO



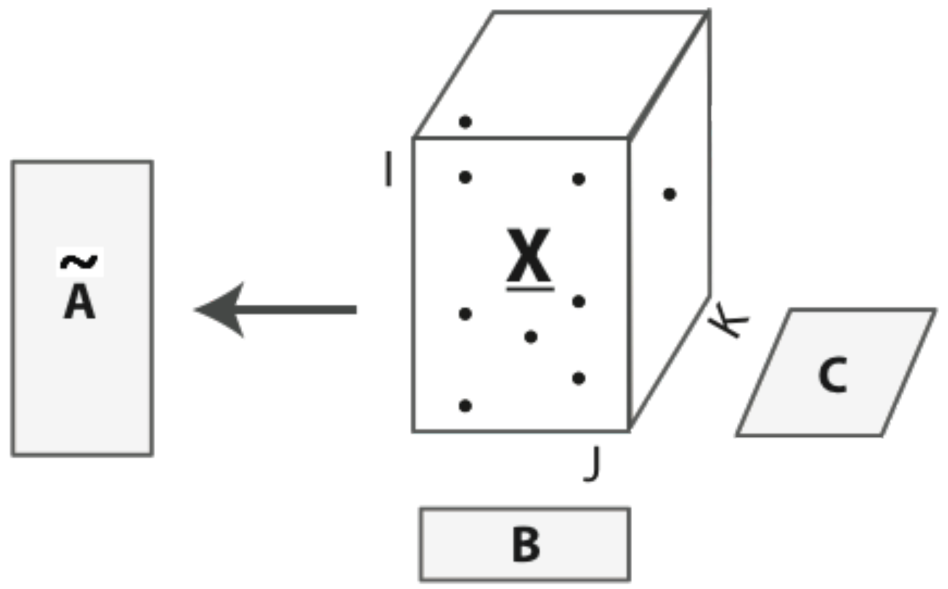
HiCOO-MTTKRP algorithm in mode-1

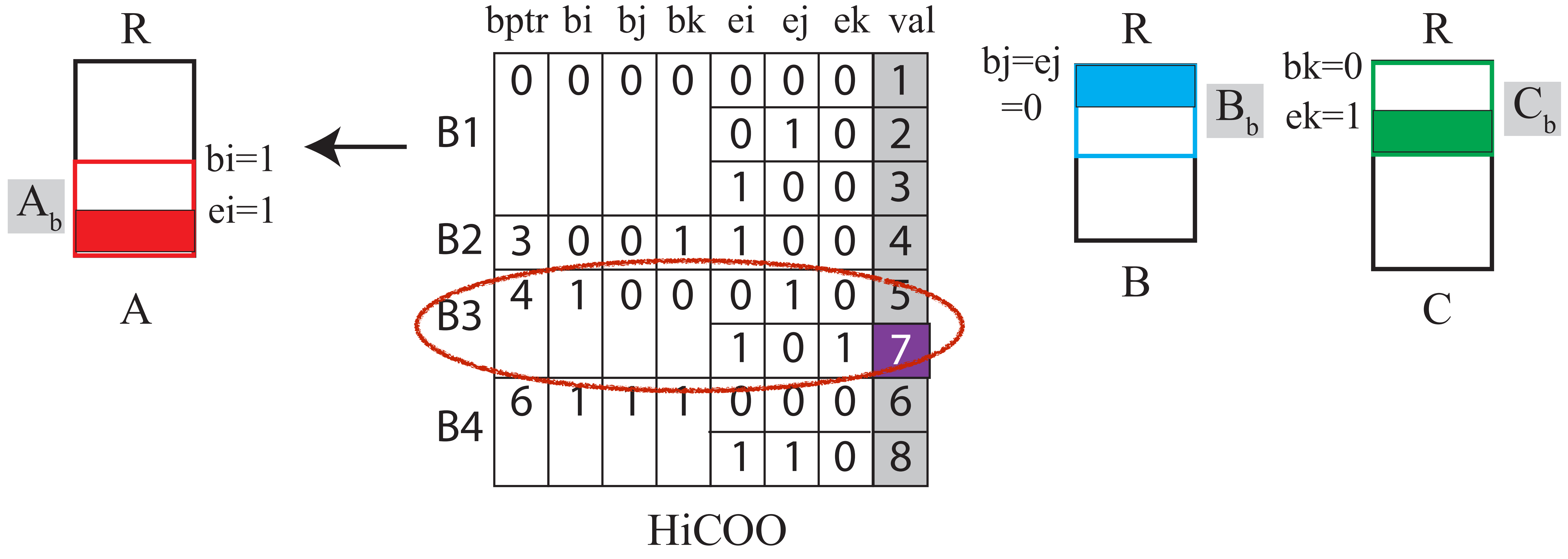
Entry-wise

HiCOO-MTTKRP

$$\tilde{\mathbf{A}} \leftarrow \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B})$$

- Tensor Matricization: $\mathbf{X}_{(1)}$
- Khatri-Rao Product: $\odot$
- Updated Factor Matrix: $\tilde{\mathbf{A}}$
- Factor Matrix: $\mathbf{C}$, $\mathbf{B}$





	bptr	bi	bj	bk	ei	ej	ek	val
B1	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

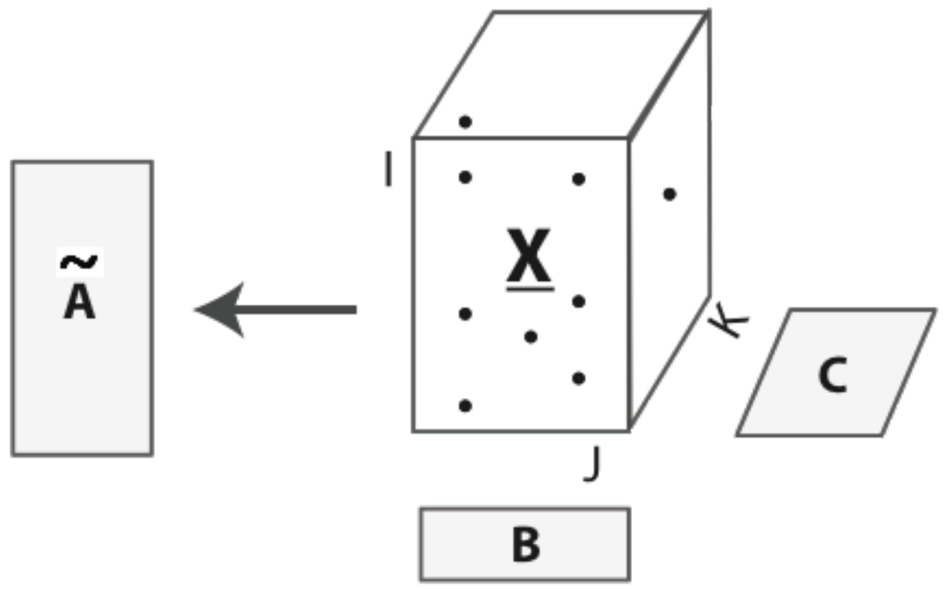
HiCOO-MTTKRP algorithm in mode-1

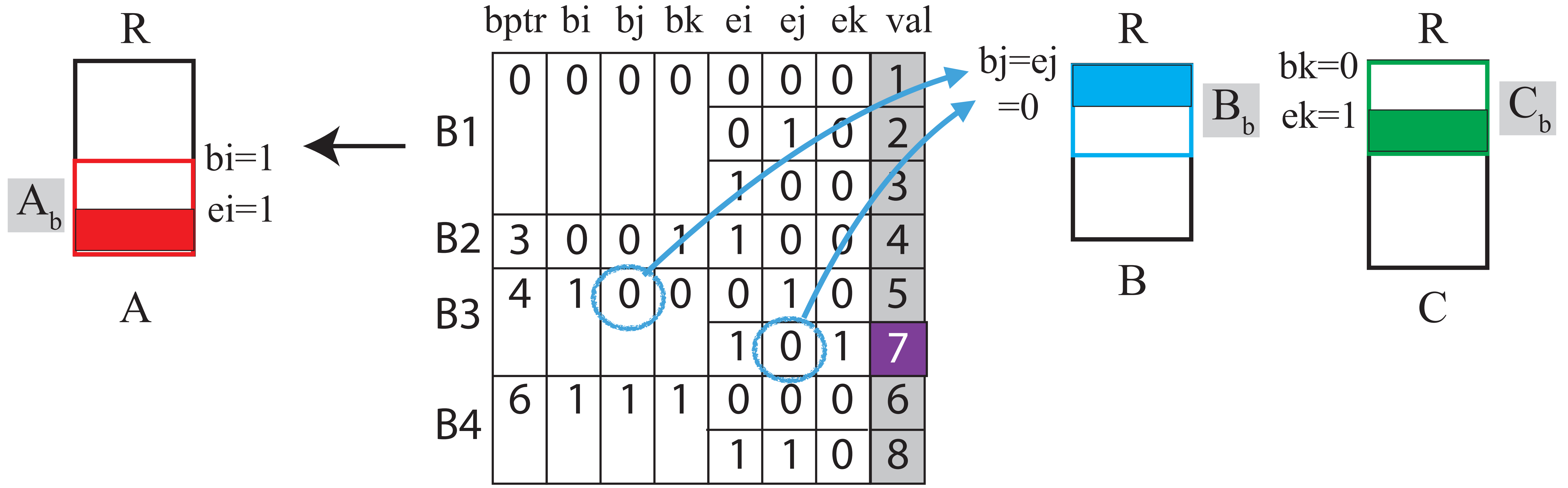
HiCOO-MTTKRP

Tensor Matricization · Khatri-Rao Product

$$\tilde{\mathbf{A}} \leftarrow \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B})$$

Updated Factor Matrix · Factor Matrix



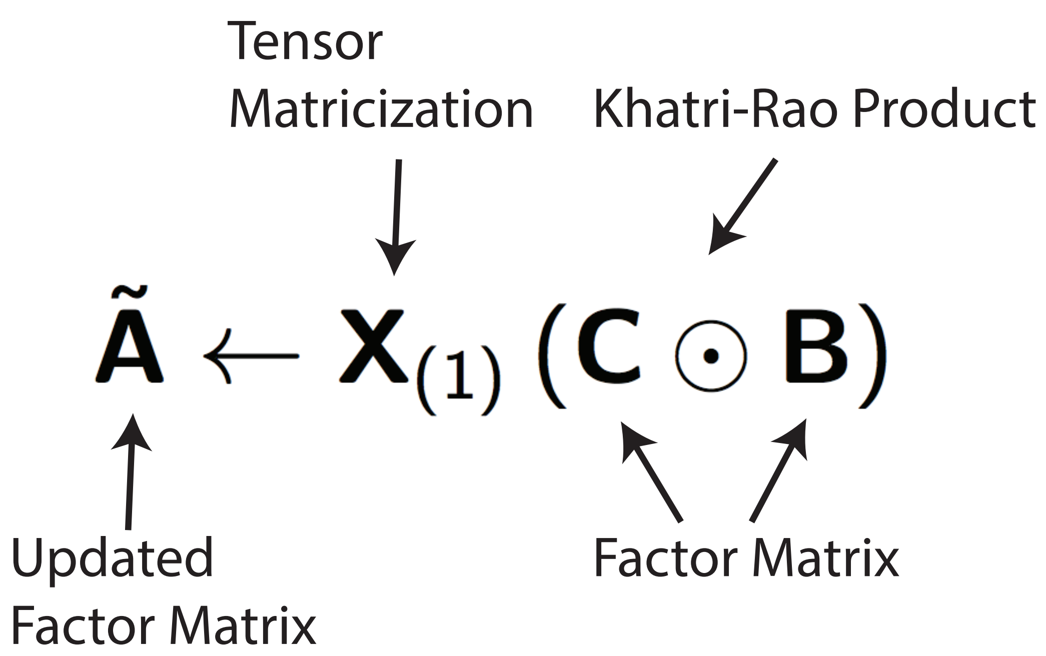


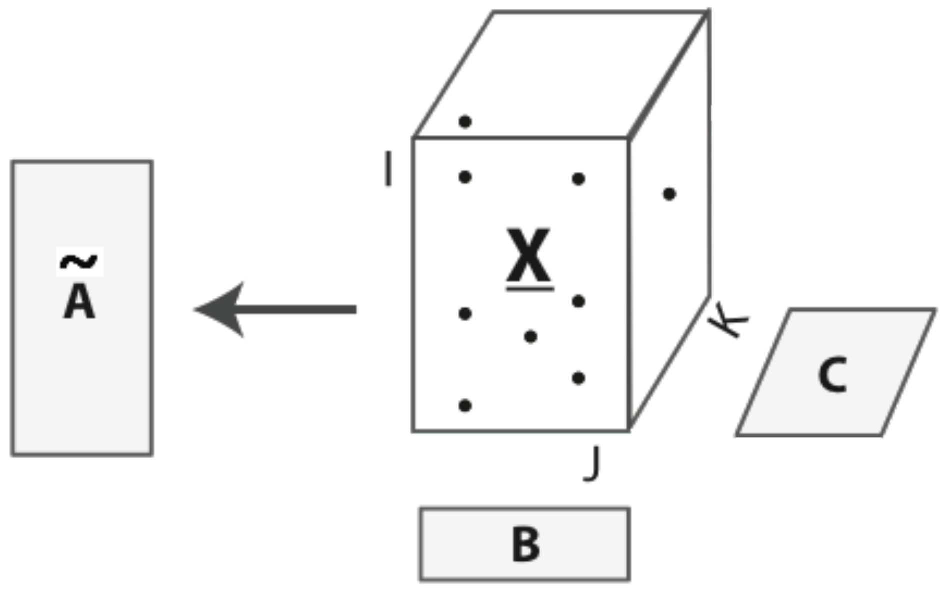
	bptr	bi	bj	bk	ei	ej	ek	val
B1	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

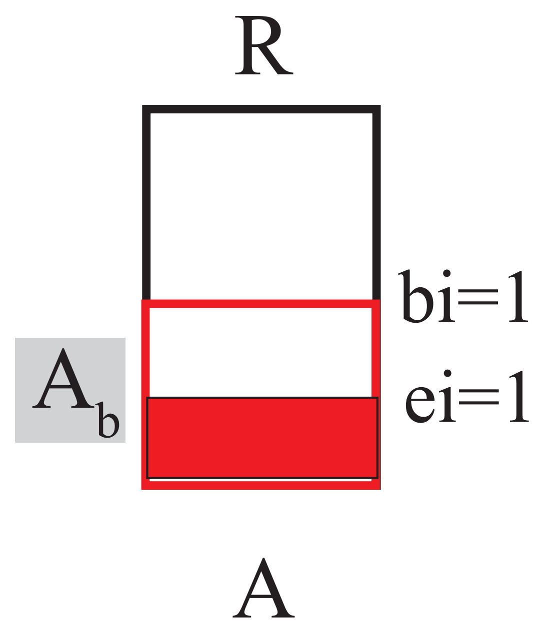
HiCOO

HiCOO-MTTKRP algorithm in mode-1

HiCOO-MTTKRP

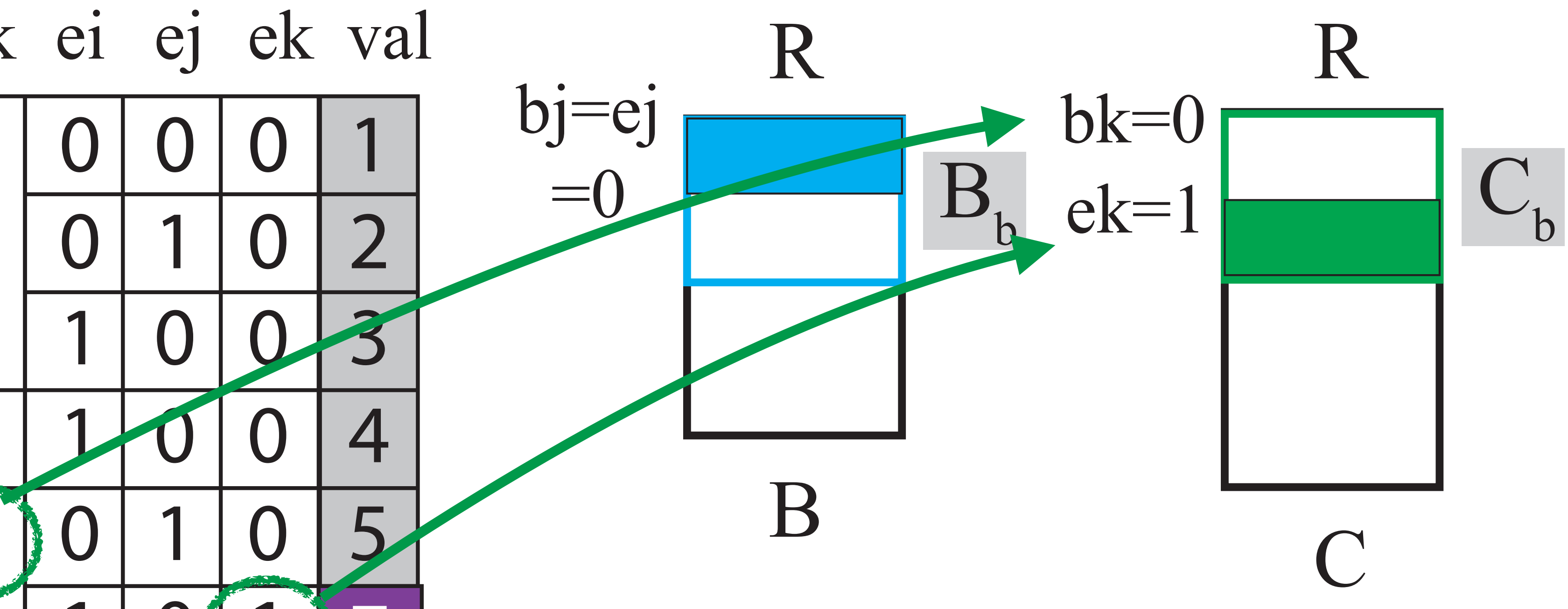






	bptr	bi	bj	bk	ei	ej	ek	val
B1	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

HiCOO



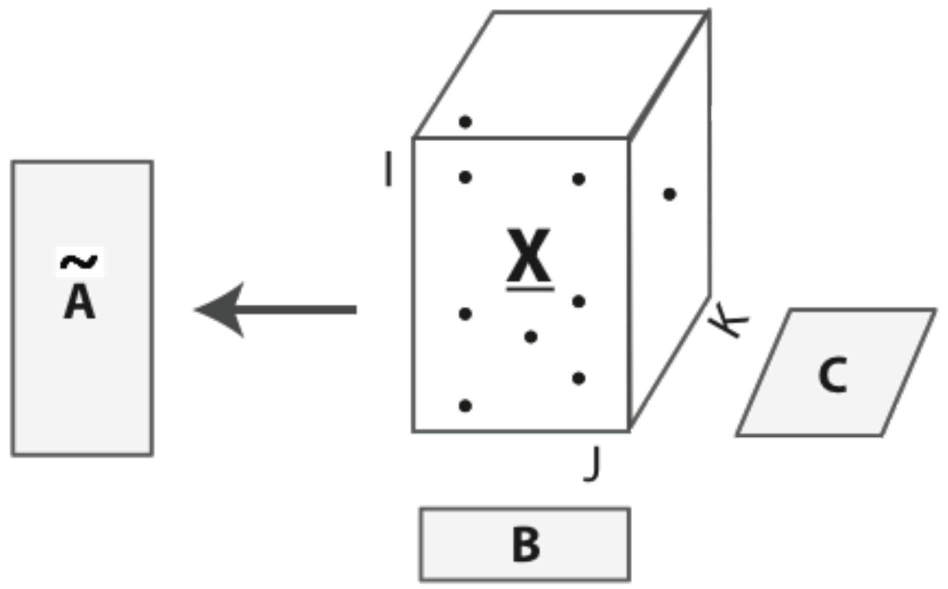
HiCOO-MTTKRP algorithm in mode-1

HiCOO-MTTKRP

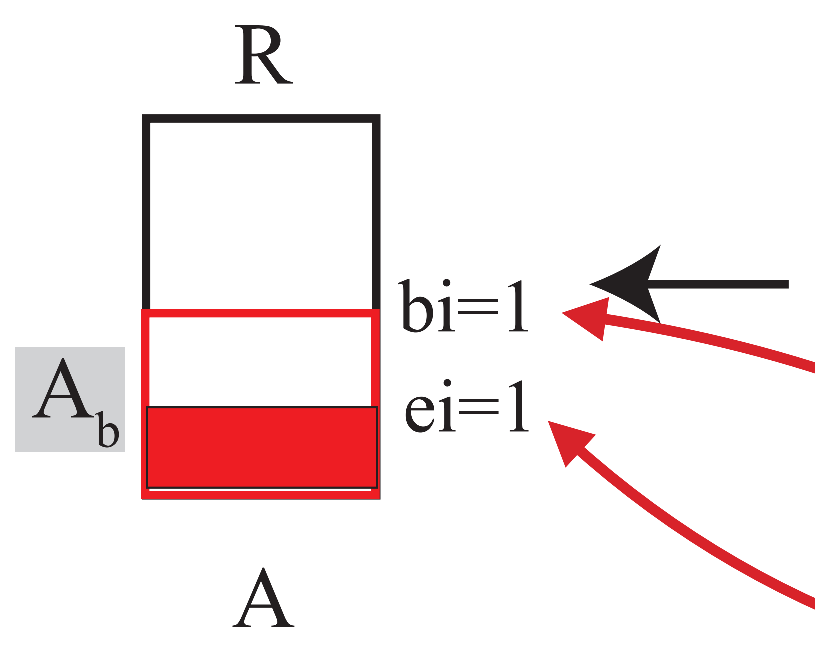
Tensor Matricization — Khatri-Rao Product

$$\tilde{\mathbf{A}} \leftarrow \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B})$$

Updated Factor Matrix — Factor Matrix

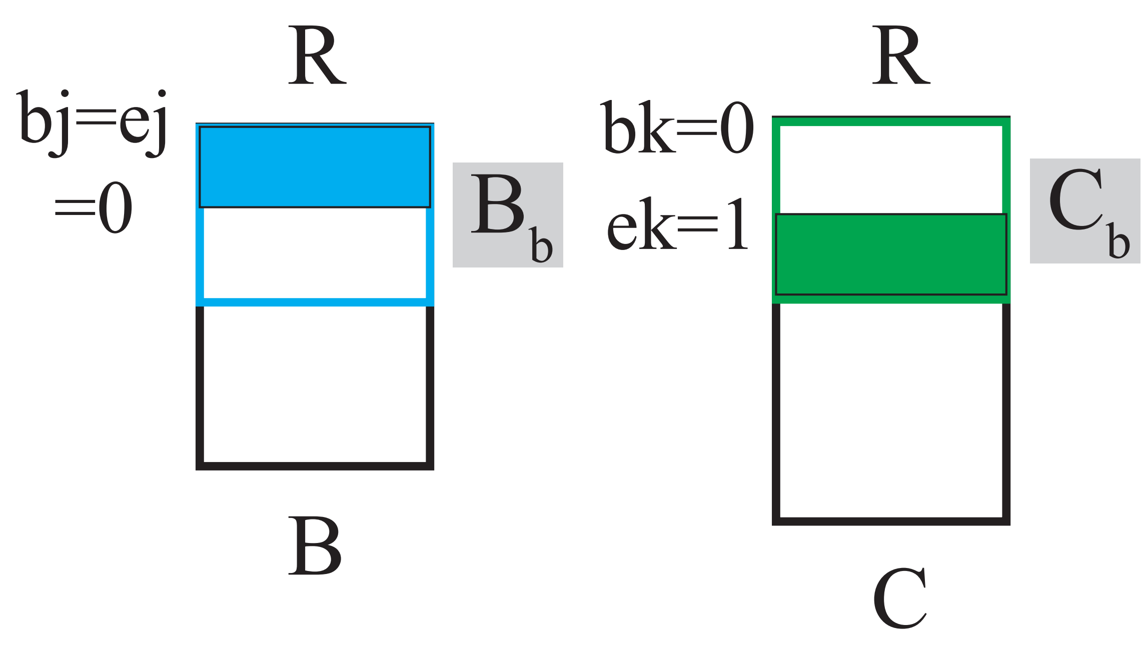


HiCOO-MTTKRP algorithm in mode-1



	bptr	bi	bj	bk	ei	ej	ek	val
B1	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B2	3	0	0	1	1	0	0	4
B3	4	1	0	0	0	1	0	5
					1	0	1	7
B4	6	1	1	1	0	0	0	6
					1	1	0	8

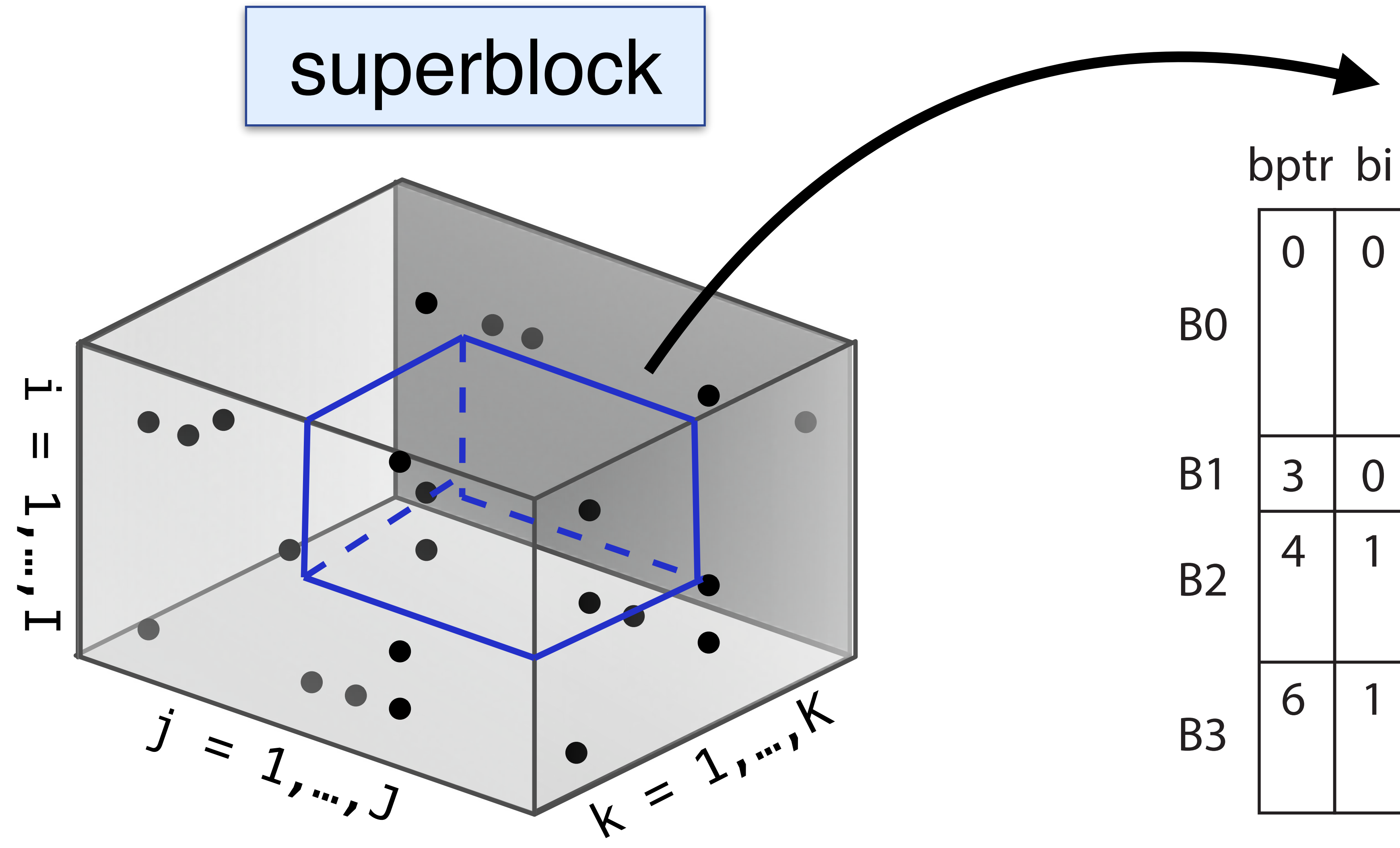
HiCOO



Entry-wise

Two-level Blocking for Efficient Thread Parallelism

- Use two-level blocking strategy
  - Large superblocks (logical) + small blocks (physical)

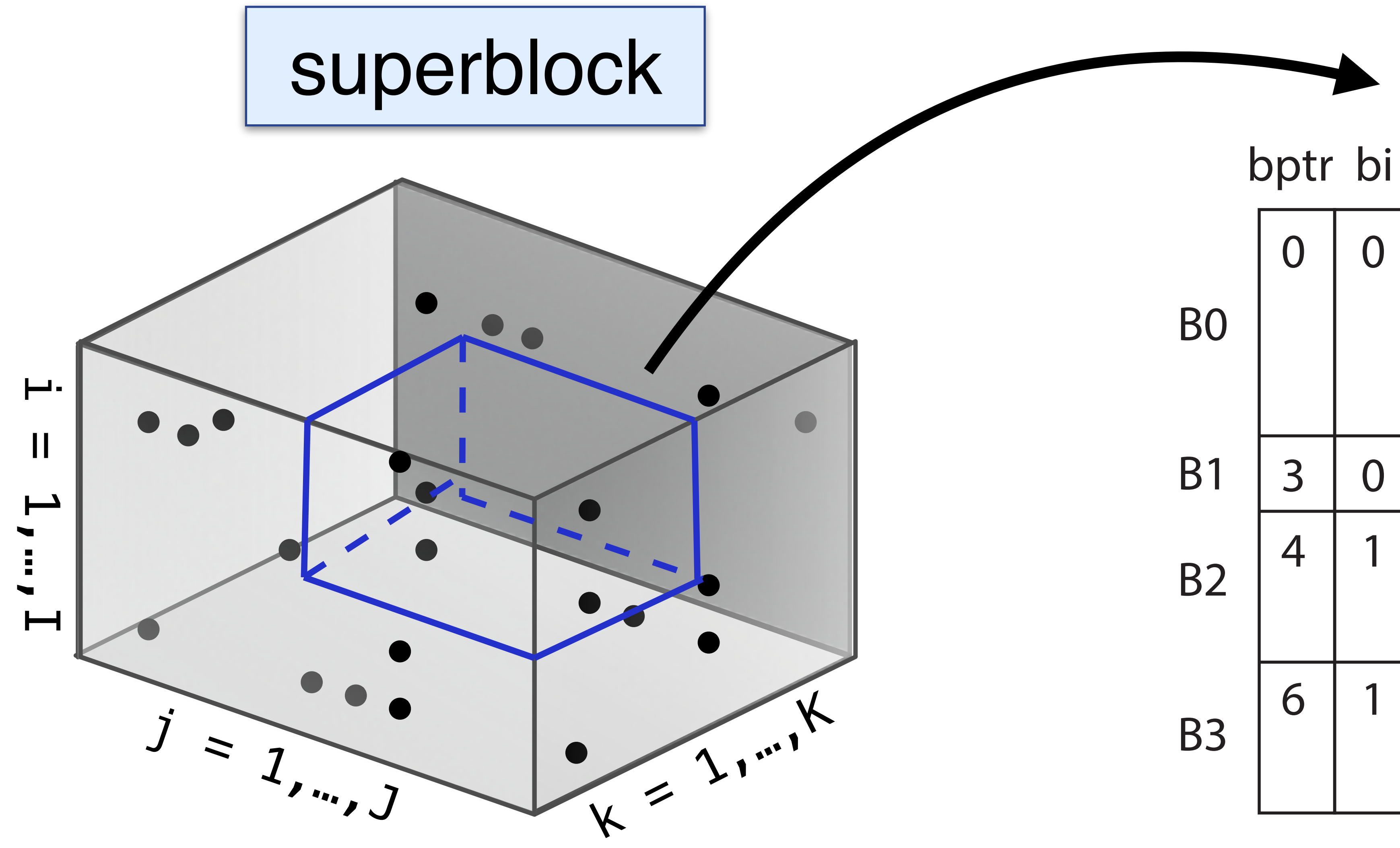


	bptr	bi	bj	bk	ei	ej	ek	val
B0	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B1	3	0	0	1	1	0	0	4
B2	4	1	0	0	0	1	0	5
					1	0	1	7
B3	6	1	1	1	0	0	0	6
					1	1	0	8

HiCOO

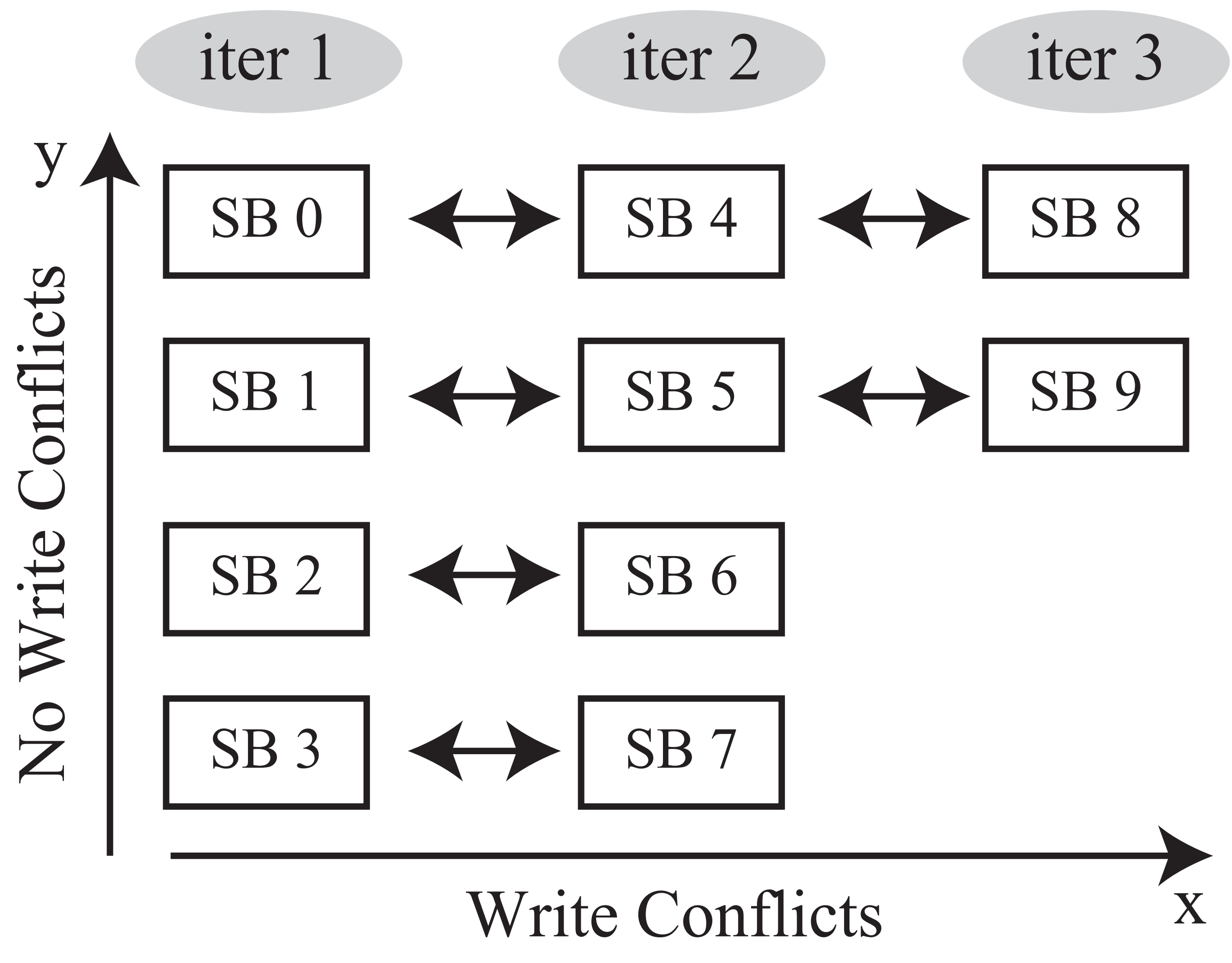
Two-level Blocking for Efficient Thread Parallelism

- Use two-level blocking strategy
  - Large superblocks (logical) + small blocks (physical)
  - To avoid using locks, we schedule superblocks according to scheduler with two parallel strategies (direct + privatization).
  - Increase only a bit extra storage.



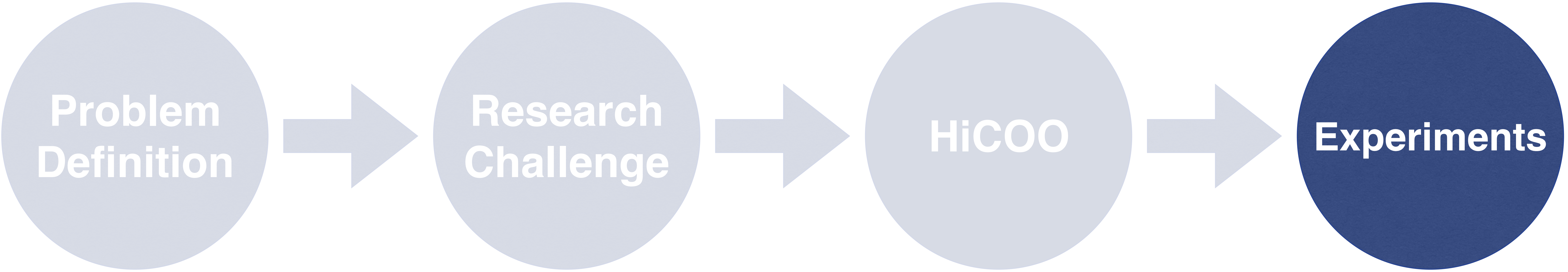
	bptr	bi	bj	bk	ei	ej	ek	val
B0	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B1	3	0	0	1	1	0	0	4
B2	4	1	0	0	0	1	0	5
					1	0	1	7
B3	6	1	1	1	0	0	0	6
					1	1	0	8

HiCOO



Superblock scheduler

Outline



Problem Definition
Research Challenge
HiCOO
Experiments

Platform and Dataset

- **Platform**: Intel Xeon CPU E7-4850 v3 platform consisting 56 physical cores with icc 18.0.2 and parallelized by OpenMP.
- **Dataset**: FROSTT [Smith et al. 2017], HaTen2 [Jeon et al. 2015], and healthcare data [Perros et al. 2017].

DESCRIPTION OF SPARSE TENSORS.

Tensors	Order	Dimensions	#Nonzeros	Density
nell2	3	$12K \times 9K \times 29K$	77M	2.4×10^{-5}
choa	3	$712K \times 10K \times 767$	27M	5.0×10^{-6}
darpa	3	$22K \times 22K \times 24M$	28M	2.4×10^{-9}
fb-m	3	$23M \times 23M \times 166$	100M	1.1×10^{-9}
fb-s	3	$39M \times 39M \times 532$	140M	1.7×10^{-10}
deli	3	$533K \times 17M \times 2.5M$	140M	6.1×10^{-12}
nell1	3	$3M \times 2M \times 25M$	144M	9.1×10^{-13}
crime	4	$6K \times 24 \times 77 \times 32$	5M	1.5×10^{-2}
nips	4	$2K \times 3K \times 14K \times 17$	3M	1.8×10^{-6}
enron	4	$6K \times 6K \times 244K \times 1K$	54M	5.5×10^{-9}
flickr	4	$320K \times 28M \times 2M \times 731$	113M	1.1×10^{-14}
deli4d	4	$533K \times 17M \times 2M \times 1K$	140M	4.3×10^{-15}

Platform and Dataset

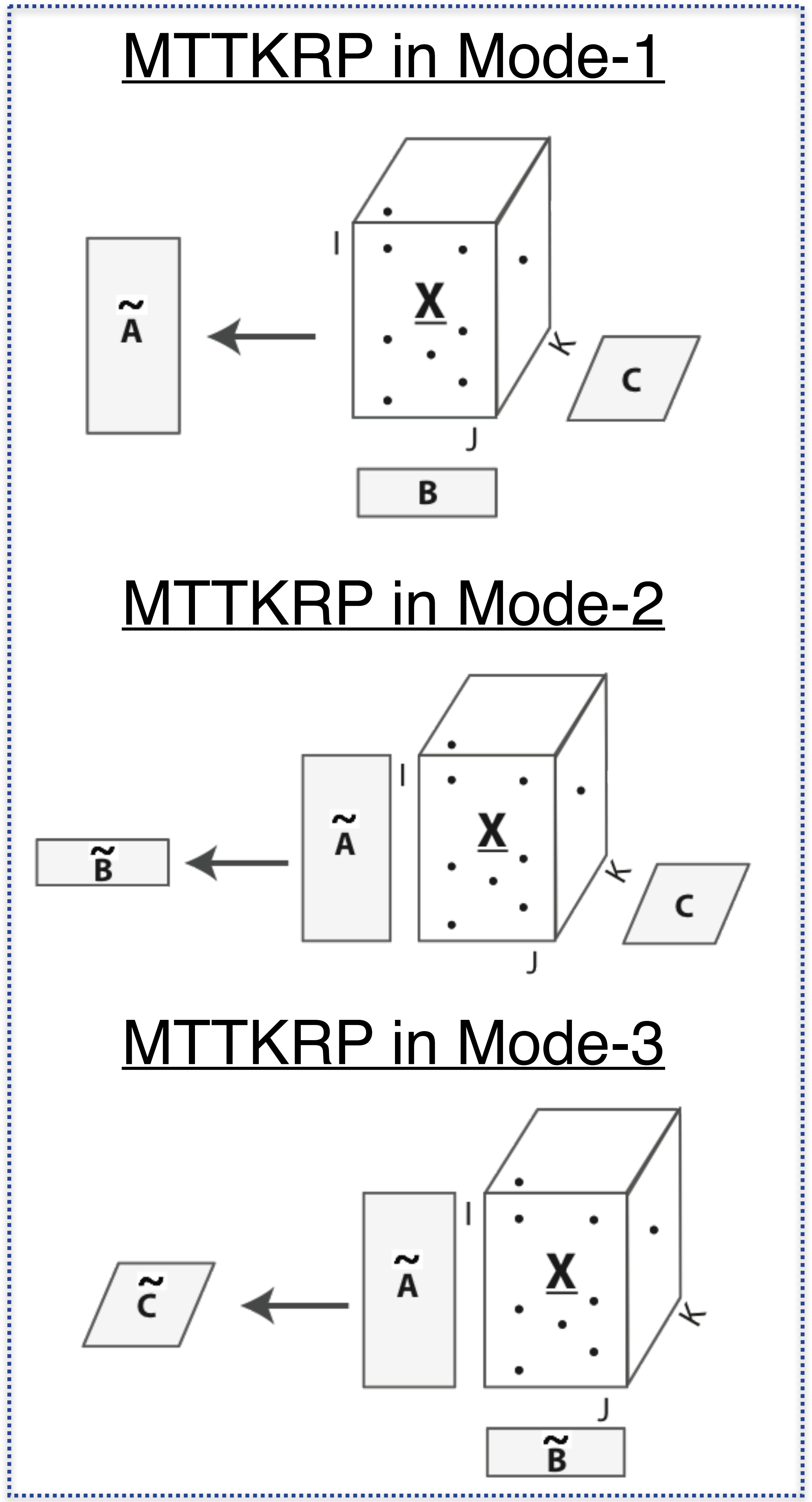
- **Platform**: Intel Xeon CPU E7-4850 v3 platform consisting 56 physical cores with icc 18.0.2 and parallelized by OpenMP.
- **Dataset**: FROSTT [Smith et al. 2017], HaTen2 [Jeon et al. 2015], and healthcare data [Perros et al. 2017].

DESCRIPTION OF SPARSE TENSORS.

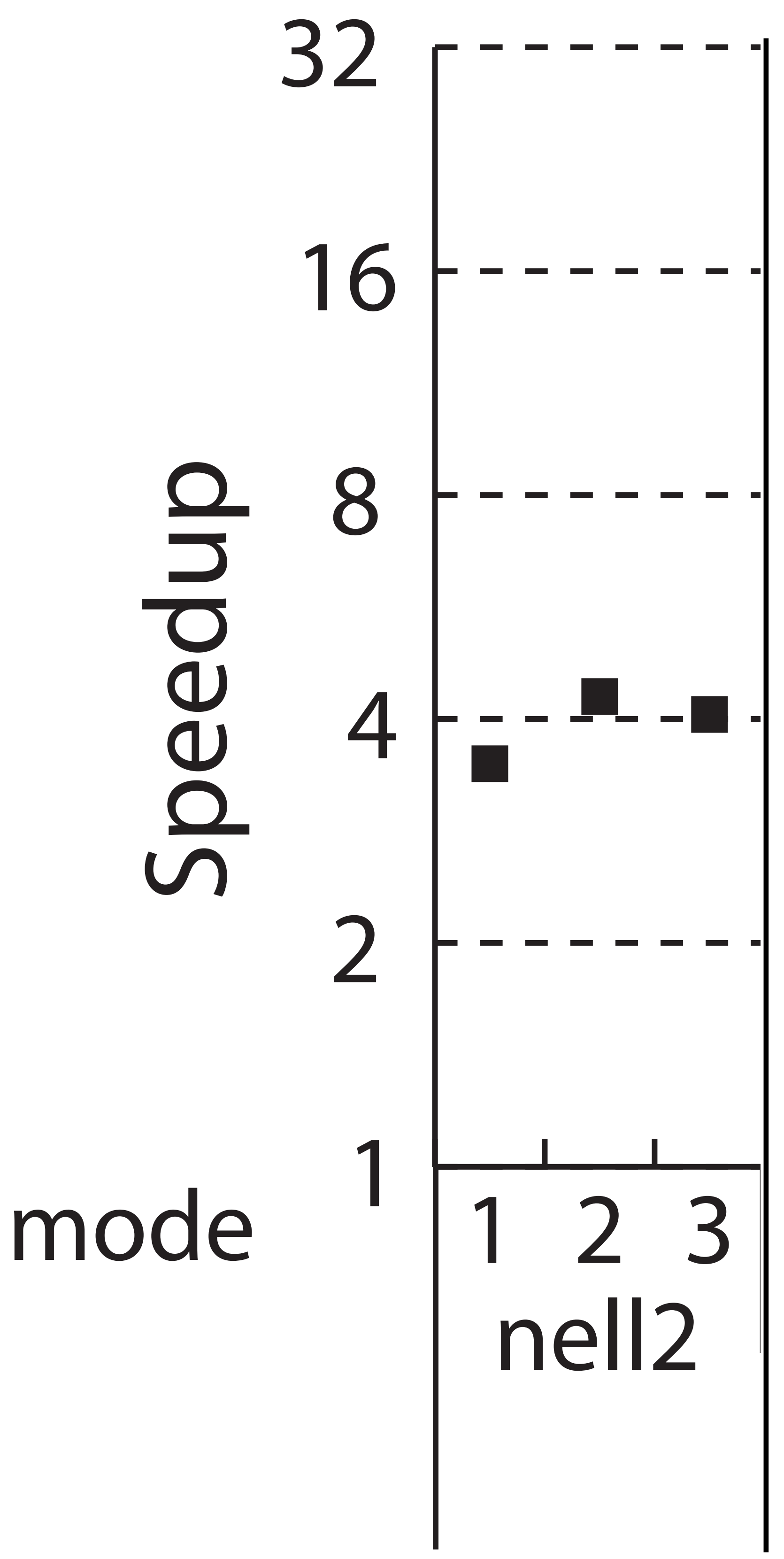
Tensors	Order	Dimensions	#Nonzeros	Density
nell2	3	$12K \times 9K \times 29K$	77M	2.4×10^{-5}
choa	3	$712K \times 10K \times 767$	27M	5.0×10^{-6}
darpa	3	$22K \times 22K \times 24M$	28M	2.4×10^{-9}
fb-m	3	$23M \times 23M \times 166$	100M	1.1×10^{-9}
fb-s	3	$39M \times 39M \times 532$	140M	1.7×10^{-10}
deli	3	$533K \times 17M \times 2.5M$	140M	6.1×10^{-12}
nell1	3	$3M \times 2M \times 25M$	144M	9.1×10^{-13}
crime	4	$6K \times 24 \times 77 \times 32$	5M	1.5×10^{-2}
nips	4	$2K \times 3K \times 14K \times 17$	3M	1.8×10^{-6}
enron	4	$6K \times 6K \times 244K \times 1K$	54M	5.5×10^{-9}
flickr	4	$320K \times 28M \times 2M \times 731$	113M	1.1×10^{-14}
deli4d	4	$533K \times 17M \times 2M \times 1K$	140M	4.3×10^{-15}

Multicore MTTKRP in all Modes

CP decomposition

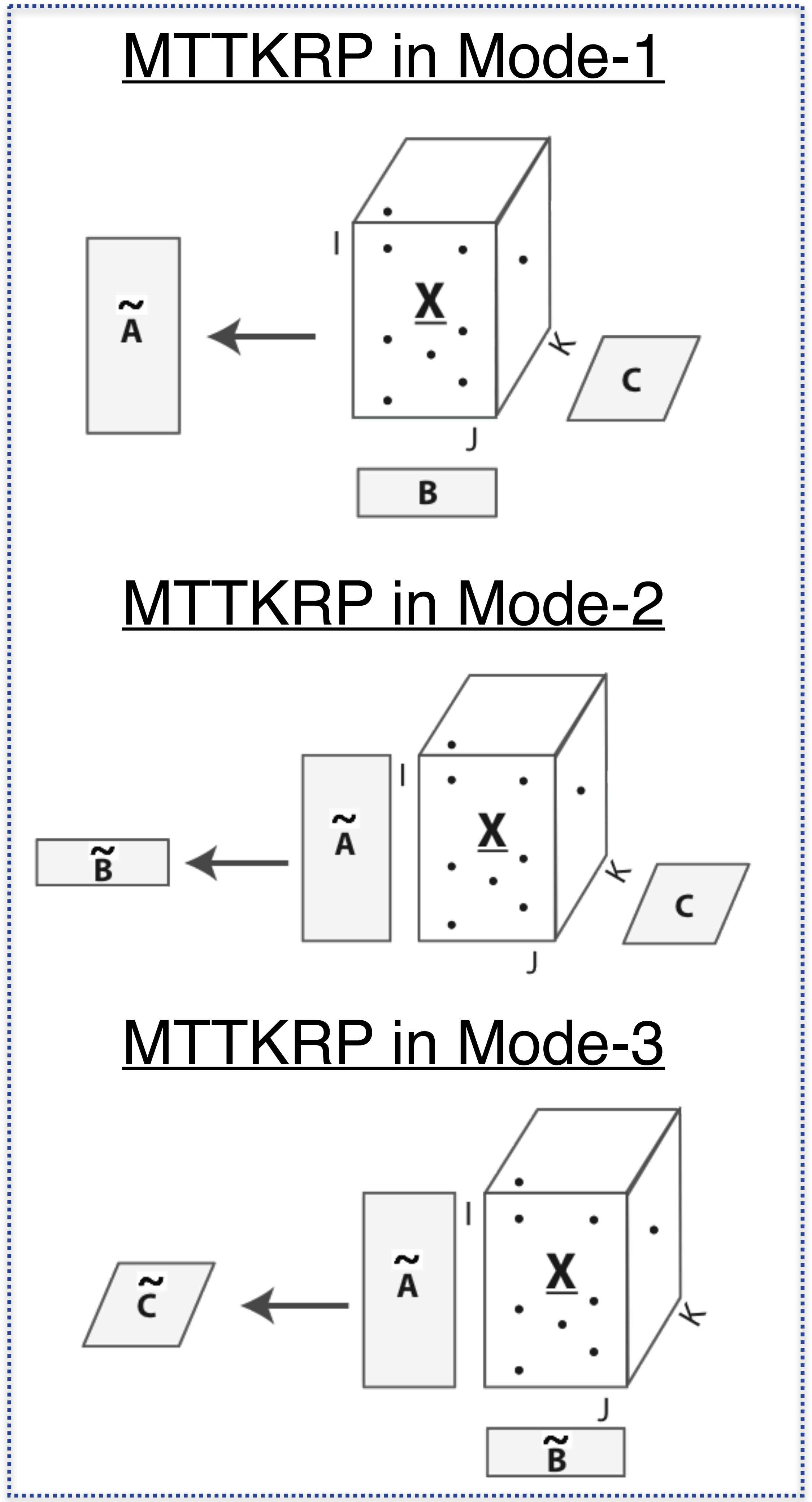


- ParTI! library: Speedups of HiCOO over COO

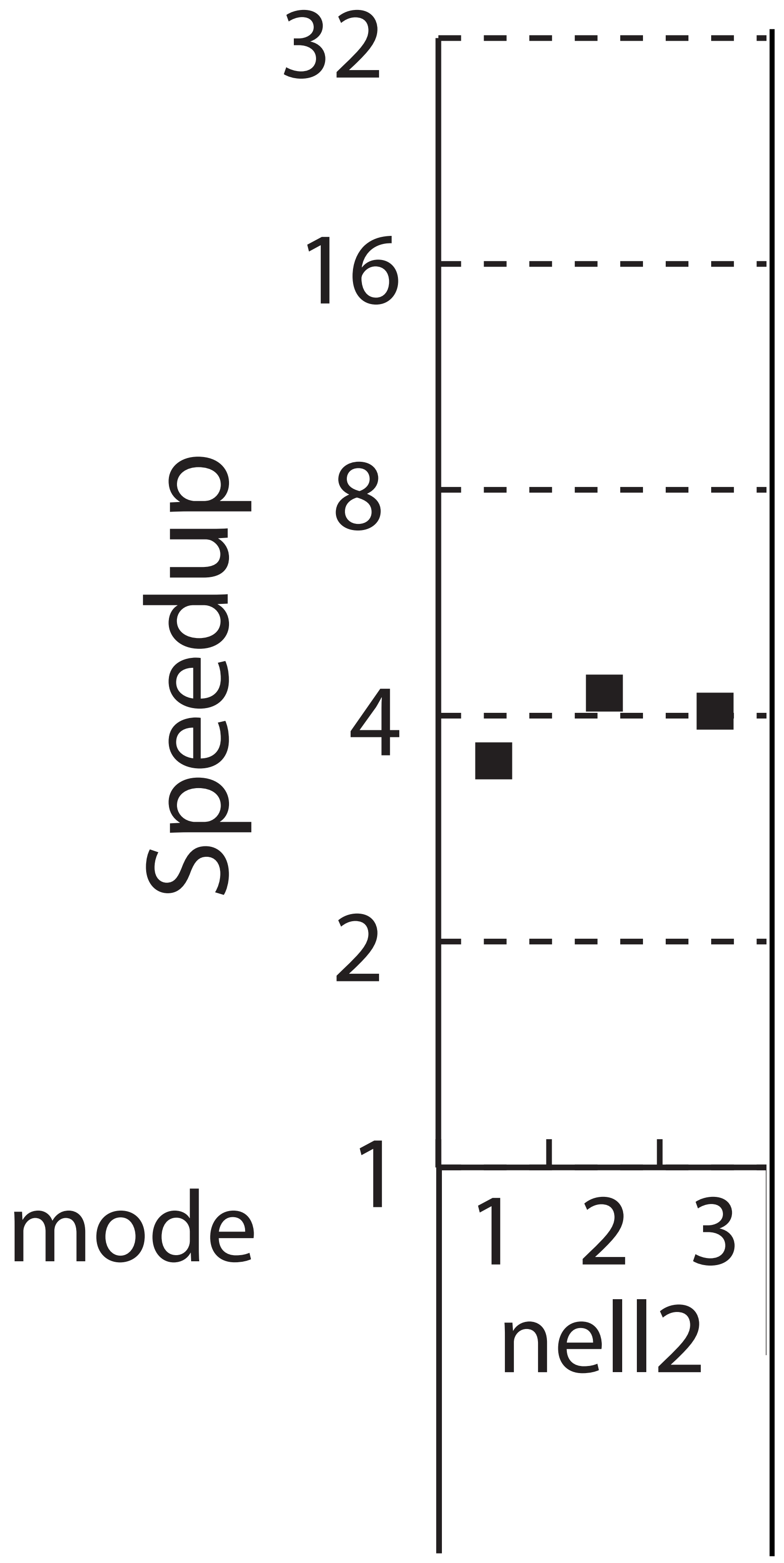


Multicore MTTKRP in all Modes

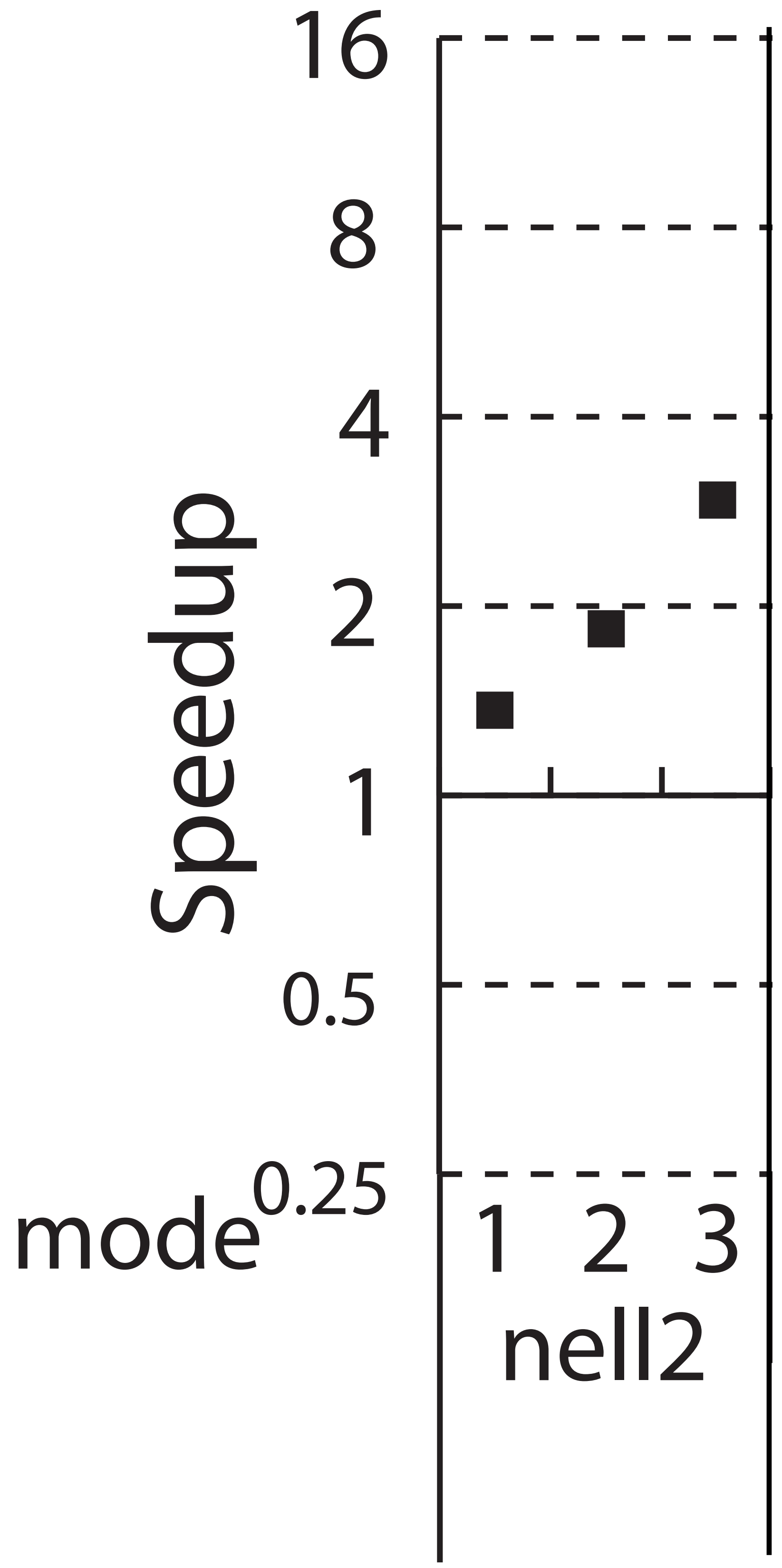
CP decomposition



- ParTI! library: Speedups of HiCOO over COO

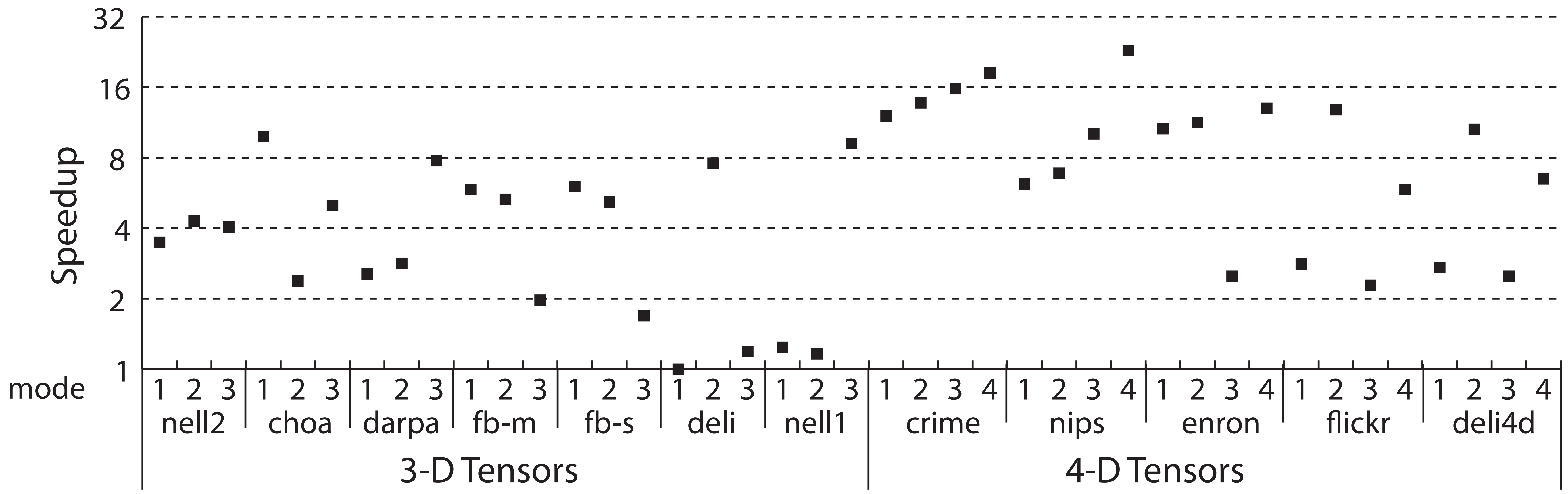


- SPLATT library: Speedups of HiCOO over CSF



Multicore MTTKRP in all Modes

- ParTI! library: Speedups of HiCOO over COO

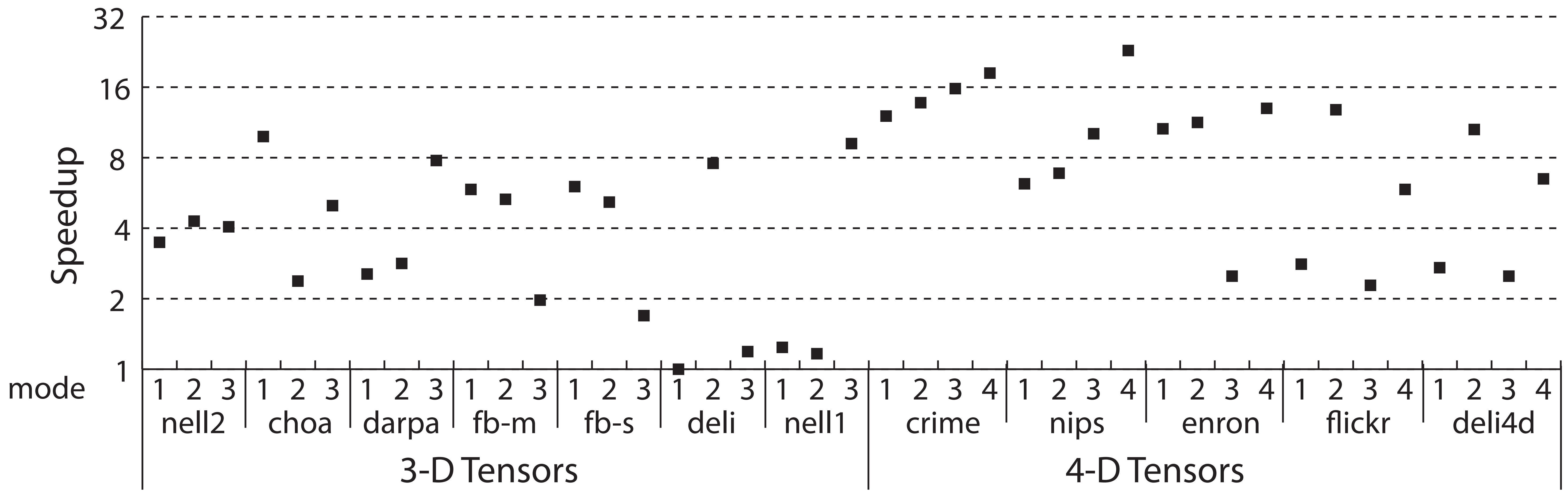


6.8x

Average speedup

Multicore MTTKRP in all Modes

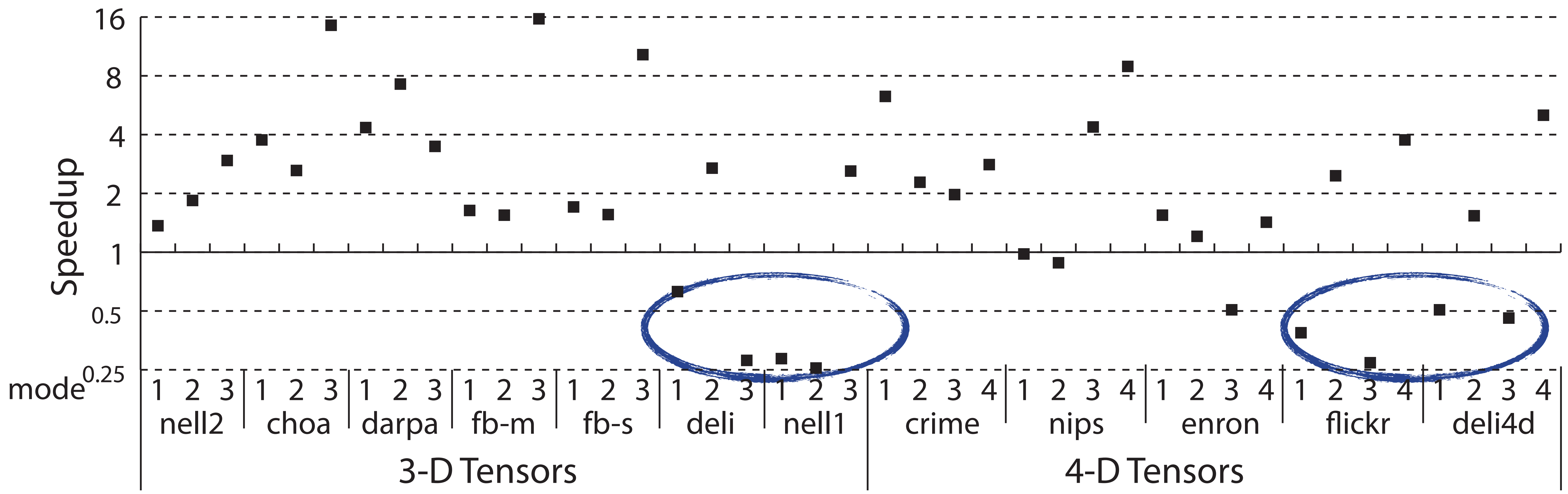
- ParTI! library: Speedups of HiCOO over COO



6.8x

- SPLATT library: Speedups of HiCOO over CSF

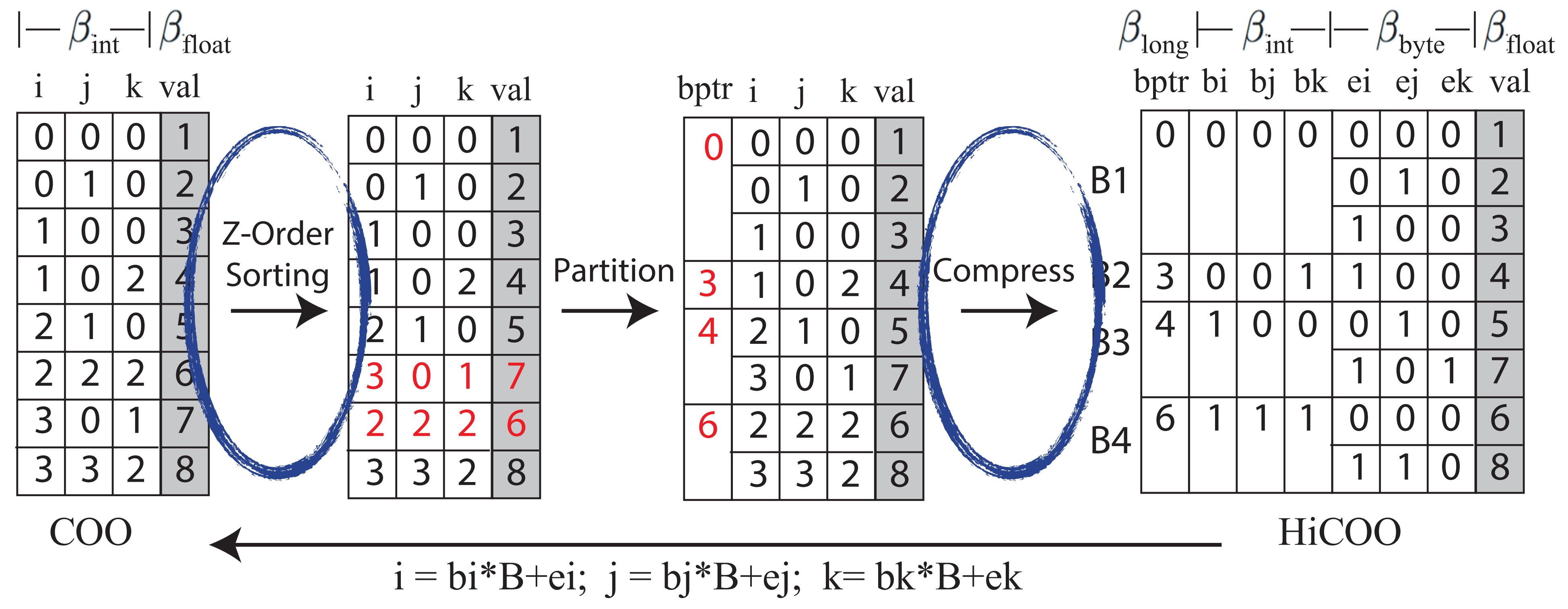
Average speedup

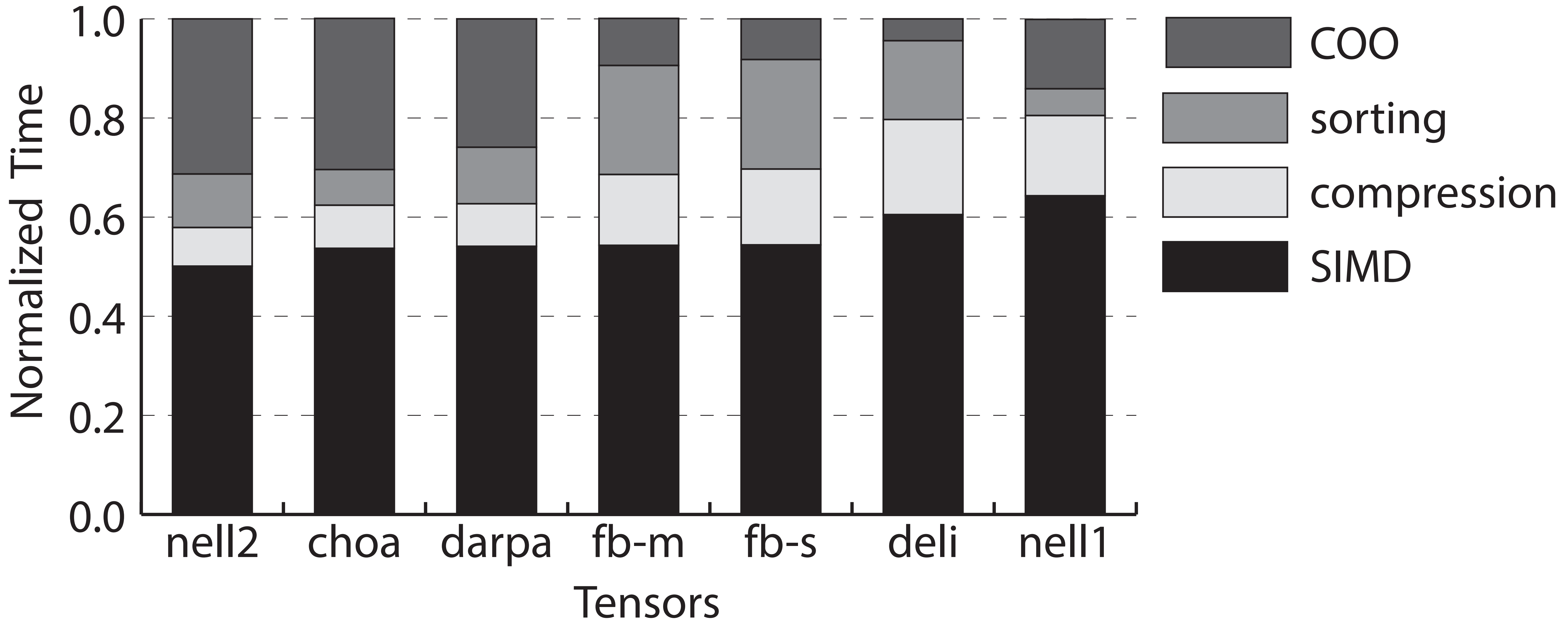


3.1x

Optimization Impact

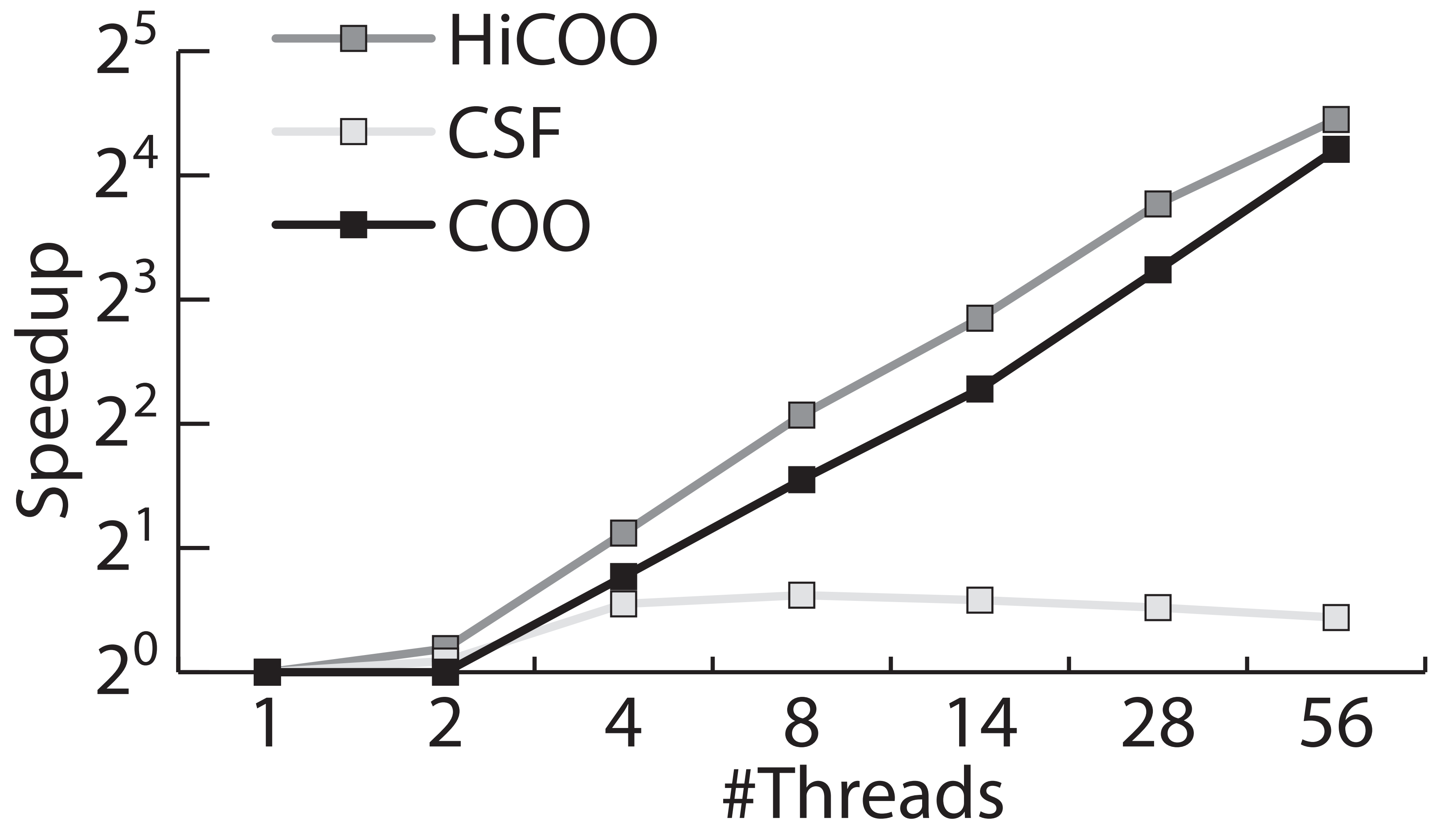
- Z-order sorting: +18%
- Index compression: +20%
- SIMD: +22%



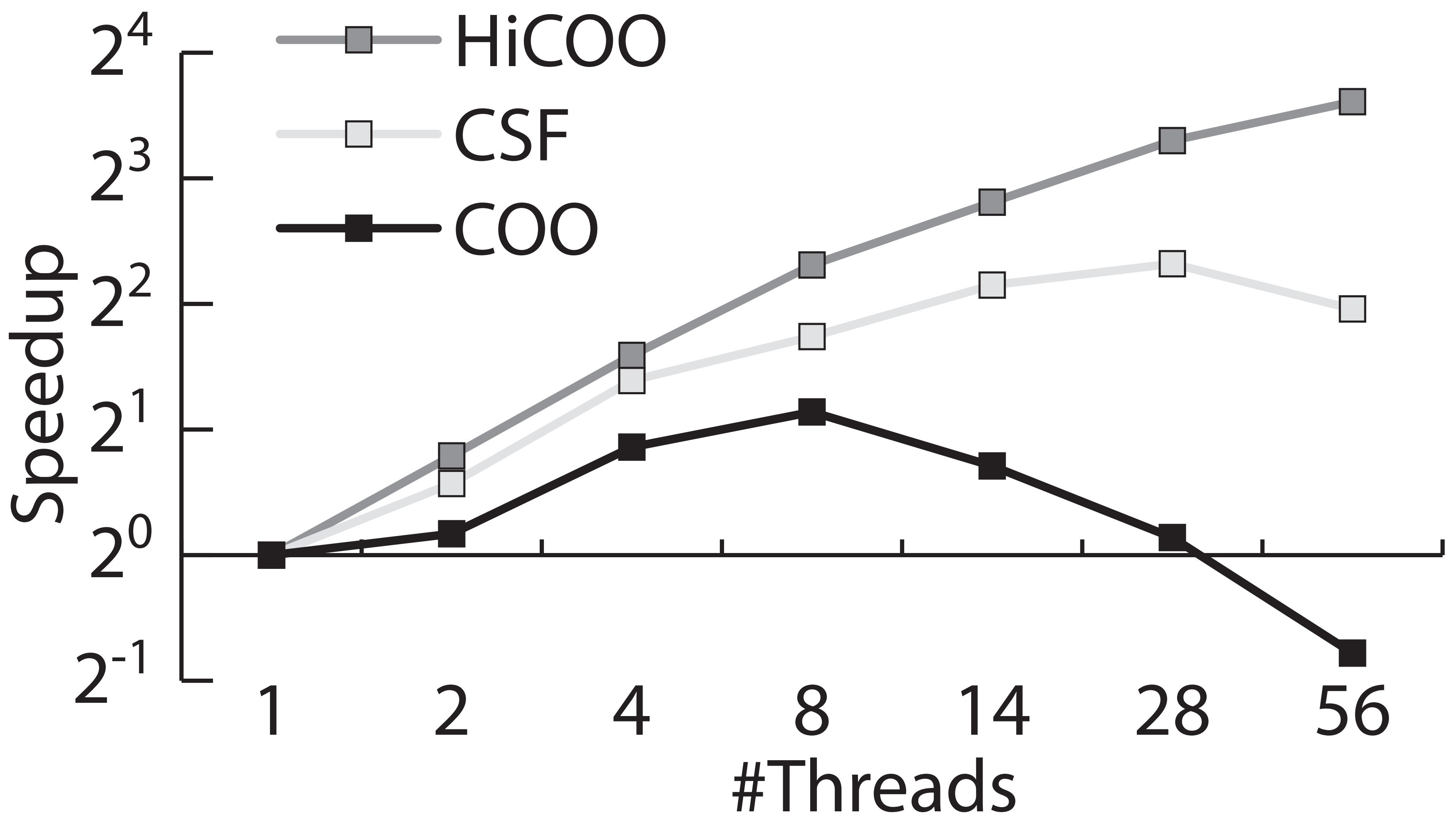


Thread Scalability

- Thread scalability of parallel COO, CSF, and HiCOO MTTKRPs on two representative cases.
- HiCOO achieves the best scalability.



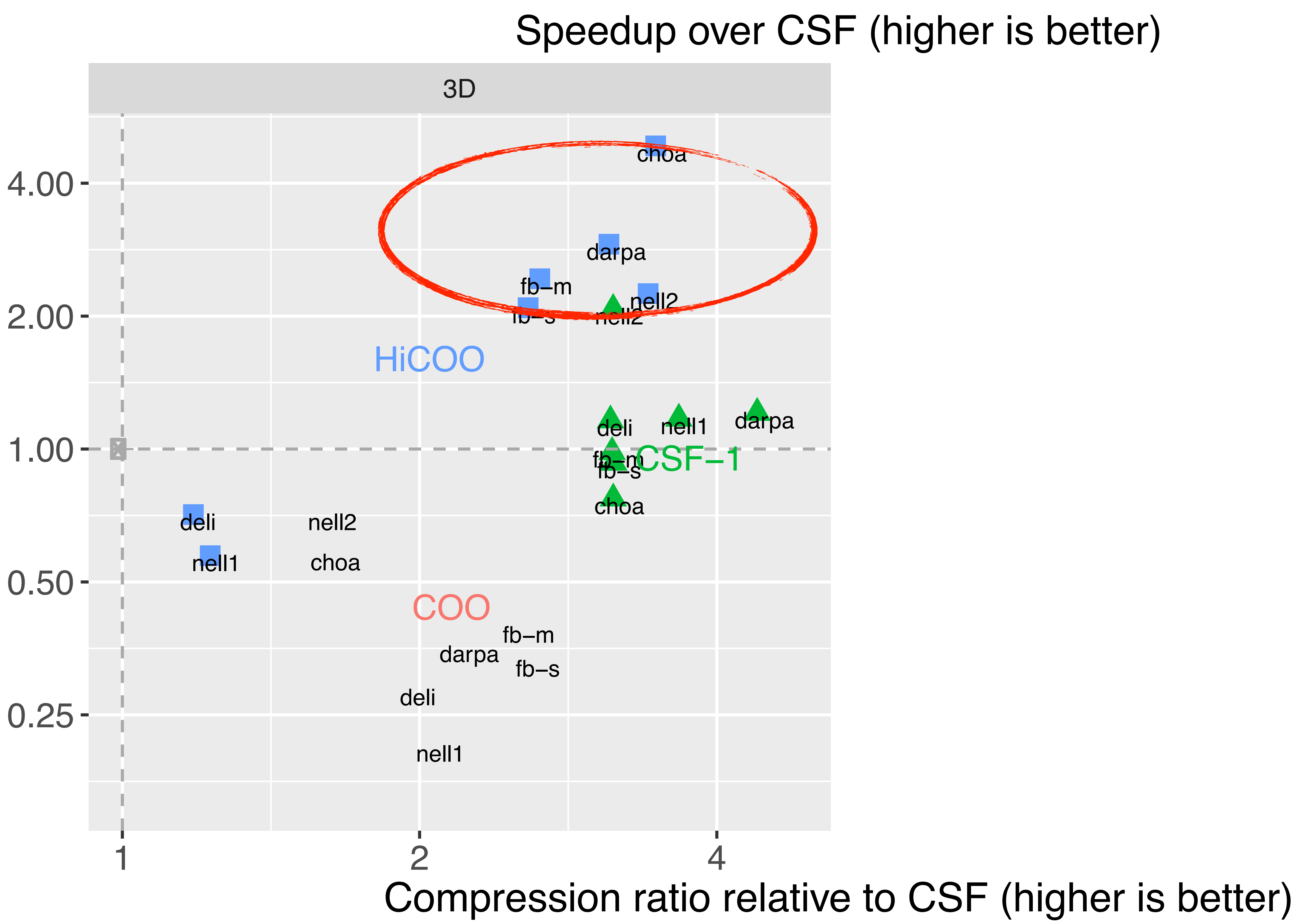
tensor fb-s in mode 3
(shortest mode)



tensor choa in mode 1
(longest mode)

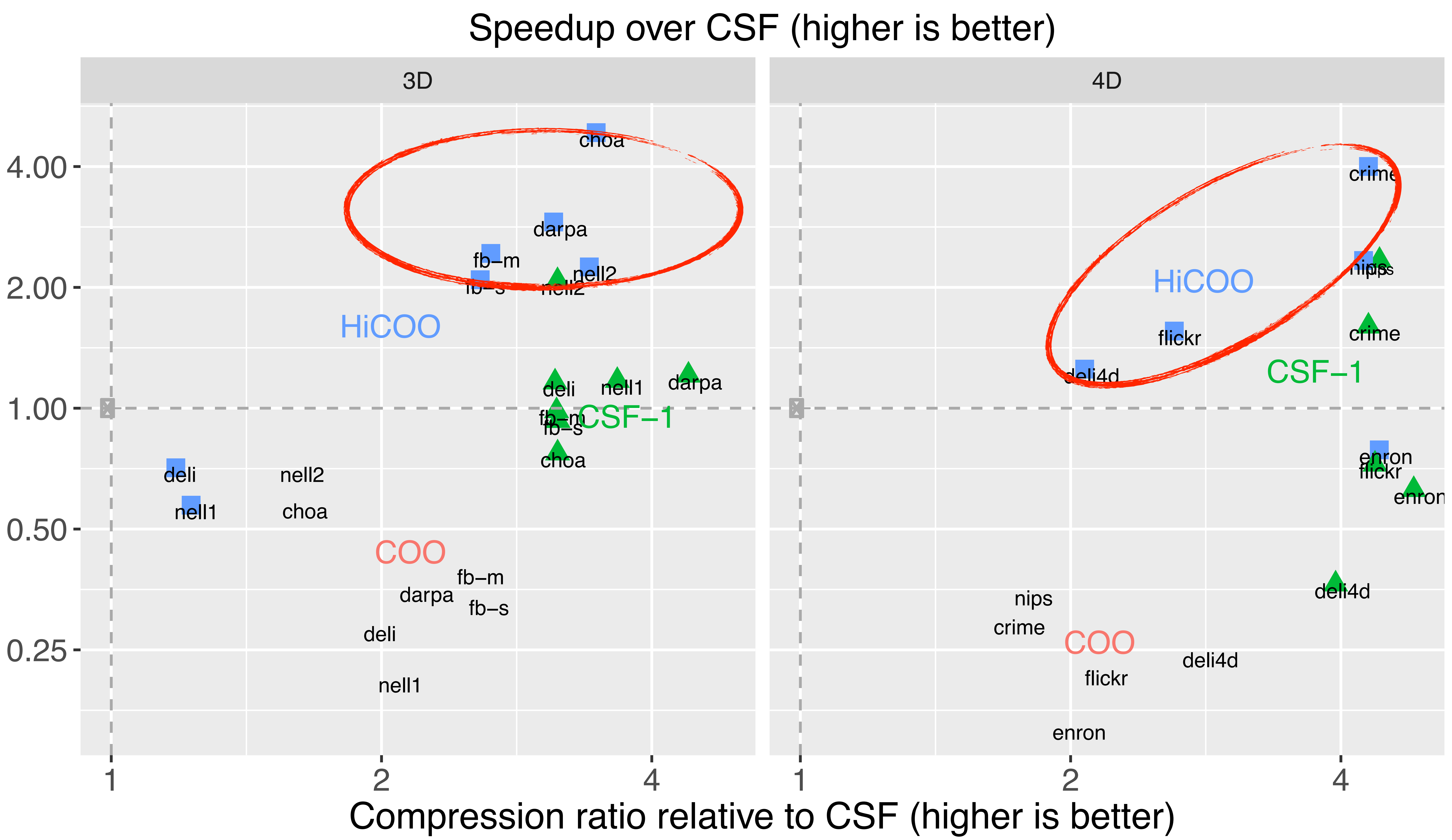
Multicore CP-ALS

- HiCOO outperforms COO by 6.2× and CSF up to 2.1× on average.



Multicore CP-ALS

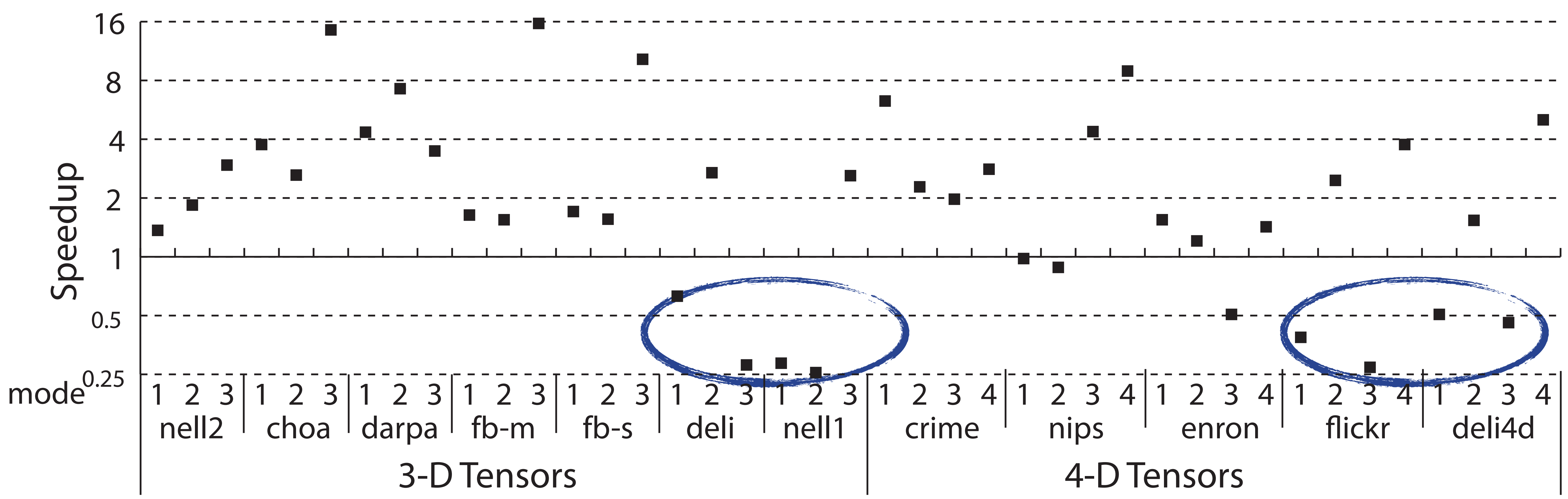
- HiCOO outperforms COO by 6.2× and CSF up to 2.1× on average.



Performance and Storage Analysis

Parameters	Meaning	Effect	Preferable values
B	Block size	Data locality	$B \le \frac{S_{cache}}{NR\beta_{float}}$
L	Superblock size	Parallel granularity	depends
α_b	Block ratio	Tensor format size	small $\alpha_b < \frac{\beta_{\text{int}} - \beta_{byte}}{\beta_{\text{int}} + \beta_{long} / N}$
$\overline{c_b}$	Average slice size per tensor block	Amount of Memory traffic	large

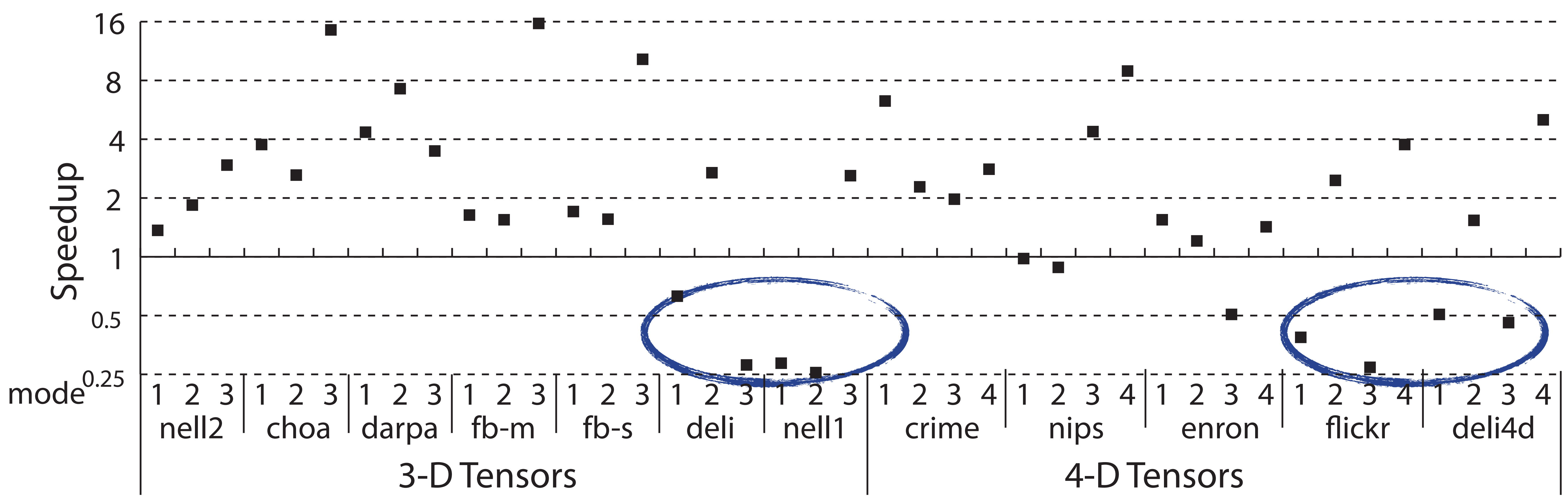
Performance and Storage Analysis cont.



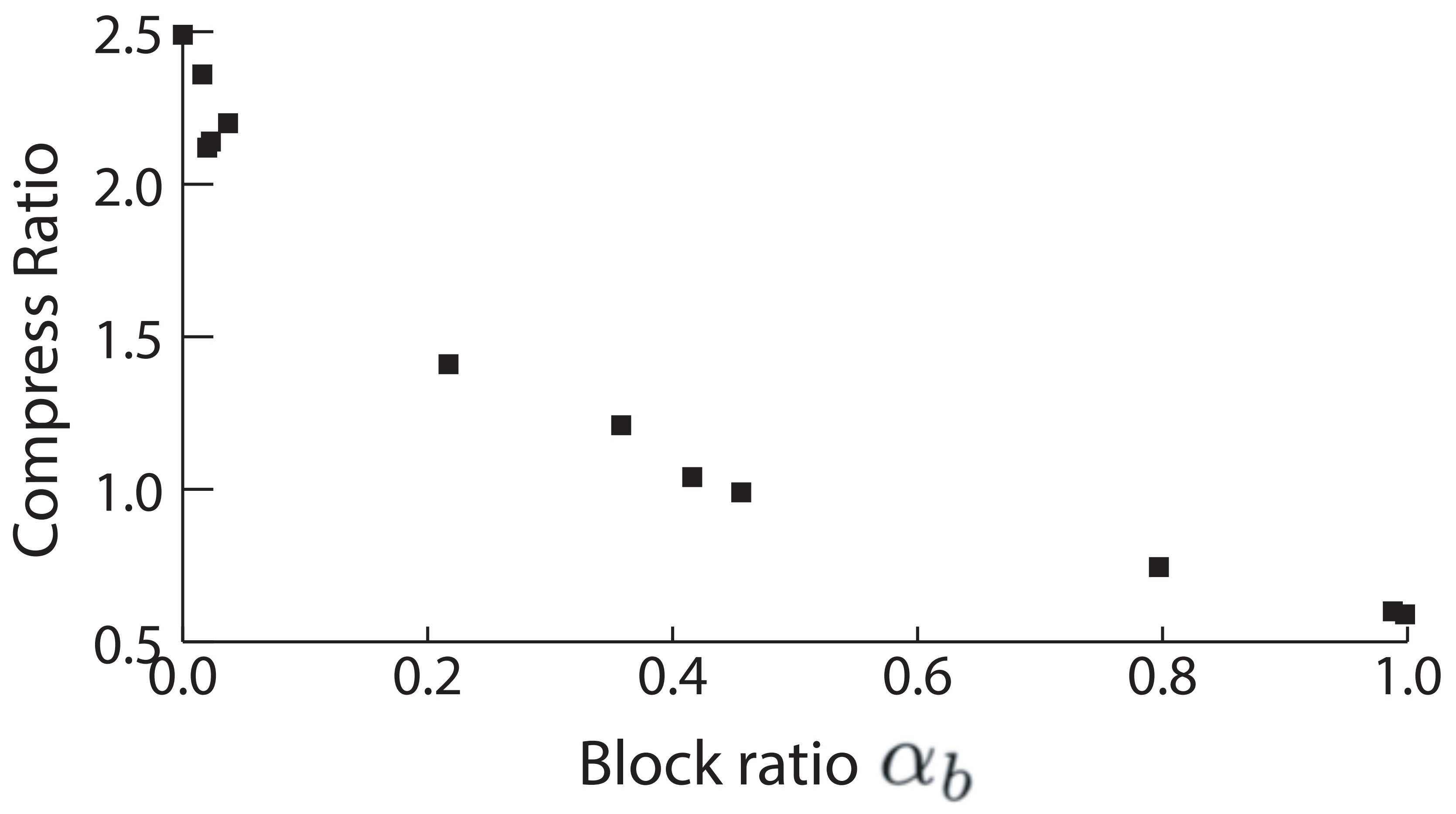
Speedups of HiCOO over CSF

Tensors	α_b	$\overline{c_b}$	Compress Ratio
nell2	0.020	0.302	2.12
choa	0.023	0.070	2.14
darpa	0.217	0.016	1.41
fb-m	0.416	0.011	1.04
fb-s	0.456	0.010	0.99
deli	0.988	0.008	0.60
nell1	0.998	0.008	0.59
crime	0.000	666.892	2.49
nips	0.016	0.416	2.36
enron	0.037	0.031	2.20
flickr	0.358	0.014	1.21
deli4d	0.797	0.009	0.74

Performance and Storage Analysis cont.



Speedups of HiCOO over CSF



Tensors	α_b	$\overline{c_b}$	Compress Ratio
nell2	0.020	0.302	2.12
choa	0.023	0.070	2.14
darpa	0.217	0.016	1.41
fb-m	0.416	0.011	1.04
fb-s	0.456	0.010	0.99
deli	0.988	0.008	0.60
nell1	0.998	0.008	0.59
crime	0.000	666.892	2.49
nips	0.016	0.416	2.36
enron	0.037	0.031	2.20
flickr	0.358	0.014	1.21
deli4d	0.797	0.009	0.74

HiCOO: Hierarchical Storage of Sparse Tensors

- Mode-generic format for arbitrary-order sparse tensors.
- Code: https://github.com/hpcgarage/ParTI (v1.0.0)
- Future steps:
  - Extend to sparse TTM and Tucker decomposition.
  - Optimize HiCOO-MTTKRP on GPUs.
  - Accelerate tensor reordering and format construction process.

32-bit			
i	j	k	val
0	0	0	1
0	1	0	2
1	0	0	3
1	0	2	4
2	1	0	5
2	2	2	6
3	0	1	7
3	3	2	8

(a) COO

		32-bit			8-bit			
	bptr	bi	bj	bk	ei	ej	ek	val
B0	0	0	0	0	0	0	0	1
					0	1	0	2
					1	0	0	3
B1	3	0	0	1	1	0	0	4
B2	4	1	0	0	0	1	0	5
					1	0	1	7
B3	6	1	1	1	0	0	0	6
					1	1	0	8

(b) HiCOO

A haiku for HiCOO

— By Richard W. Vuduc

Flexible format
Of hierarchical sparse blocks
Small, and often fast

Acknowledgement

SunLab
Enlighten the World

NSF

Sandia National Laboratories

Pacific Northwest NATIONAL LABORATORY