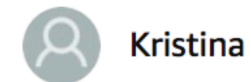


# Unified Compilation for Lossless Compression and Sparse Computing

Daniel Donenfeld, Stephen Chou, Saman Amarasinghe



# Real world data is often redundant



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.



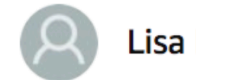
Teresa

★★★★★ **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.



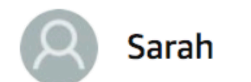
Lisa

★★★☆☆ **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.

# Real world data is often redundant

 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.

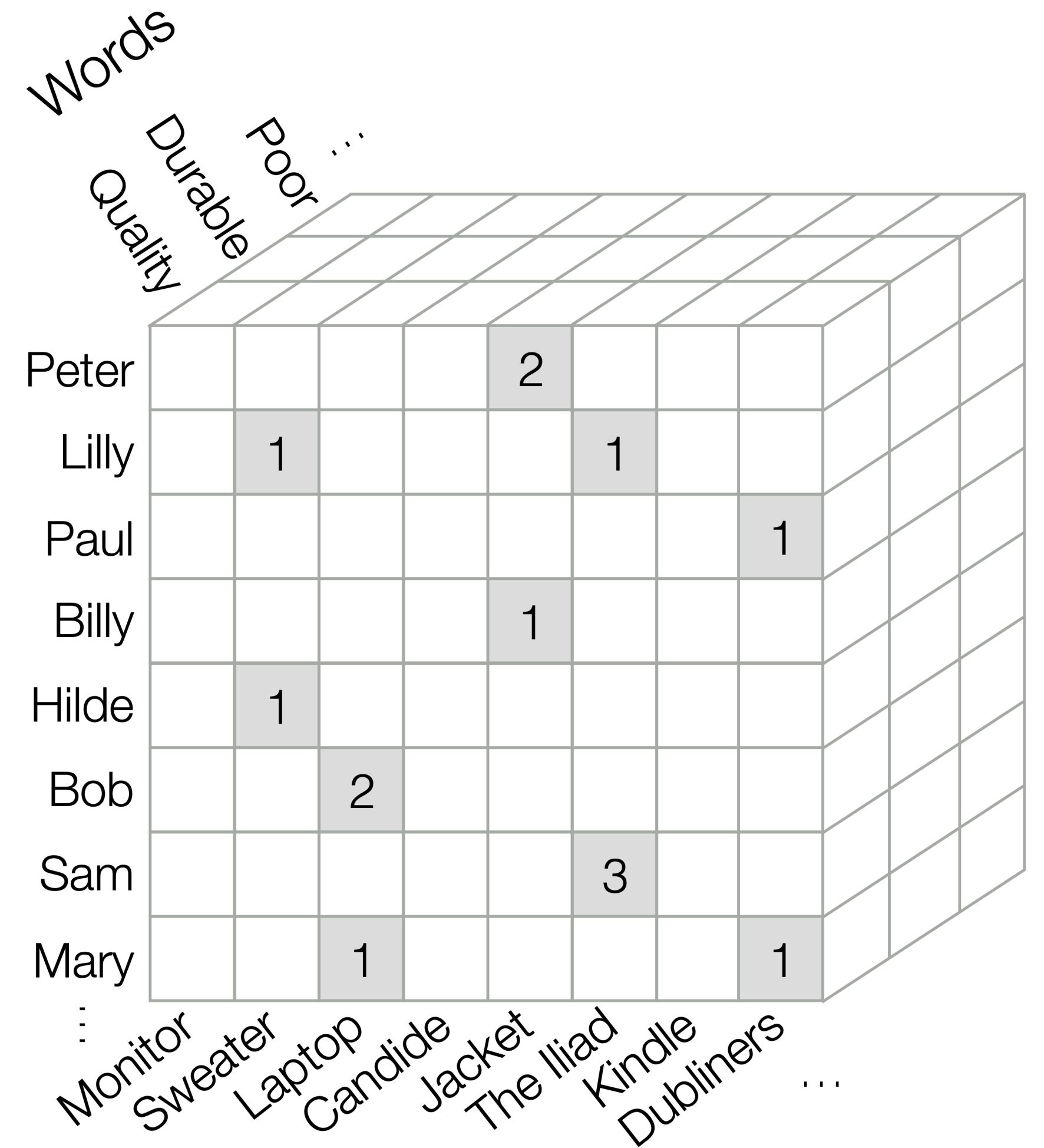
 Teresa  
 ★★★★★ **Excellent buy**  
 October 25, 2017  
**Verified Purchase**  
 This is a great product for your boy who loves sports! It was a good value as well. Other stores sell for 3x the cost. I bought one for a basketball and football and my 9 year old loves it in his room. Solid item too, not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





Users




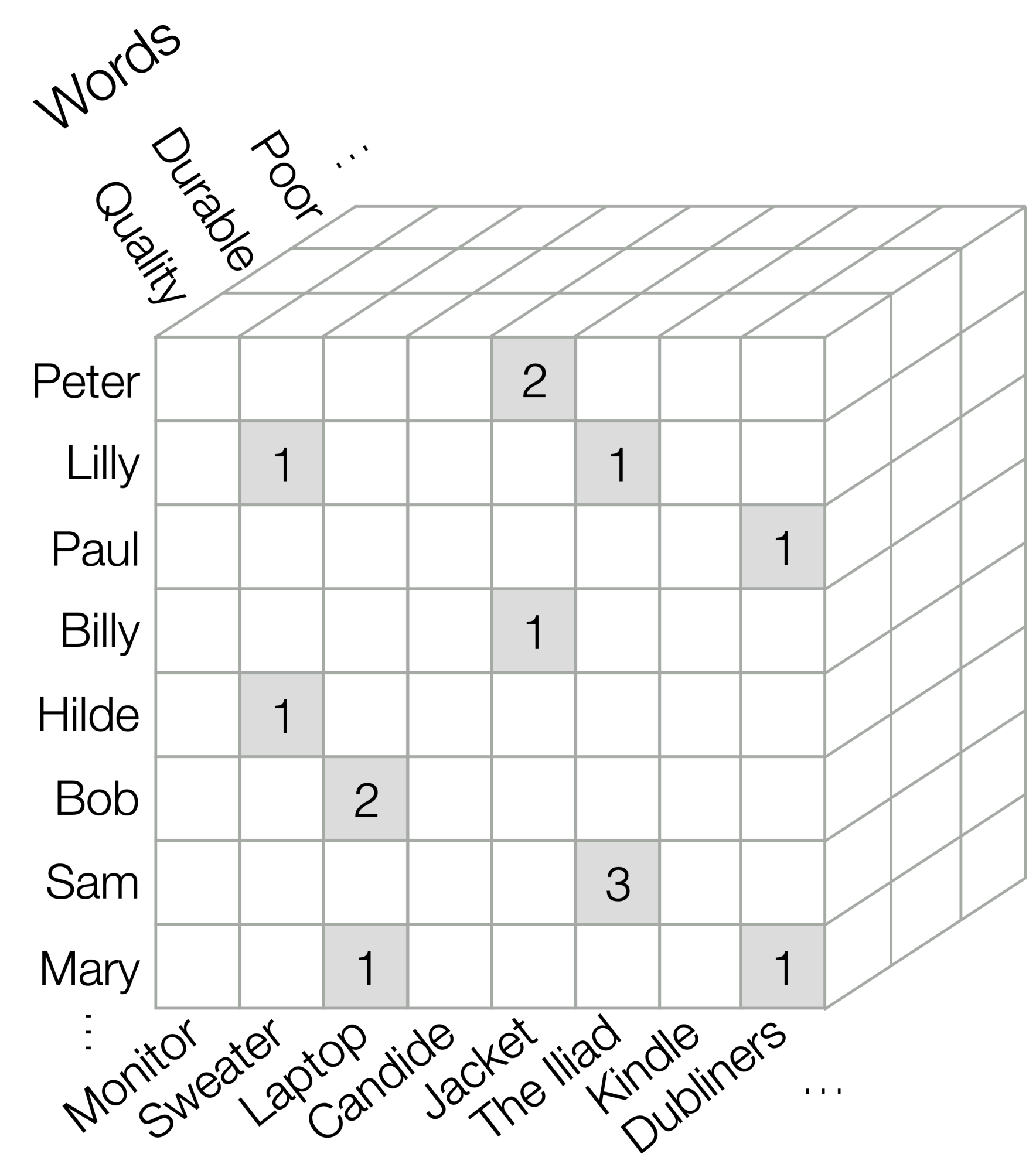
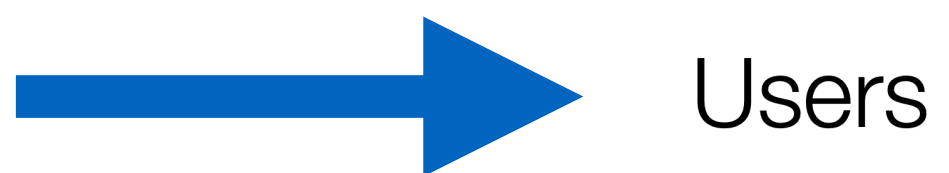
# Real world data is often redundant

 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.



Dense storage: 107 exabytes  
 Sparse storage: 13 gigabytes



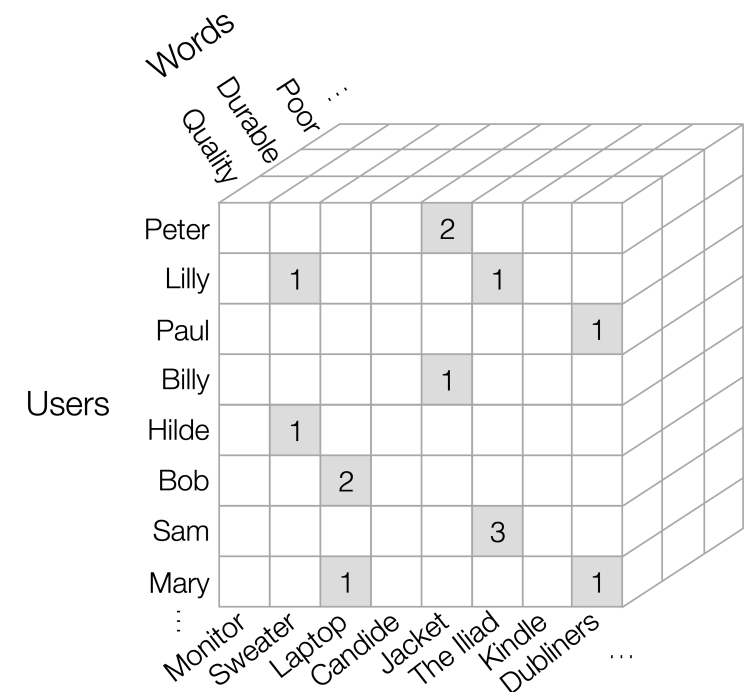
# Real world data is often redundant

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.















# Sparsity

Dense Matrix

CSR

COO

DCSR

DIA

CSF

DCSC

Sparse vector

Blocked COO

Coordinate tensor

CSB

Blocked Tensors

Hash Maps

BCSR

CSC

ELLPACK

# Lossless Compression

RLE

LZ78

LZ77

Huffman Coding

Arithmetic Coding

7z

GZIP

ZIP

FLAC

ALAC

bzip2

VP9

H.264

H.265

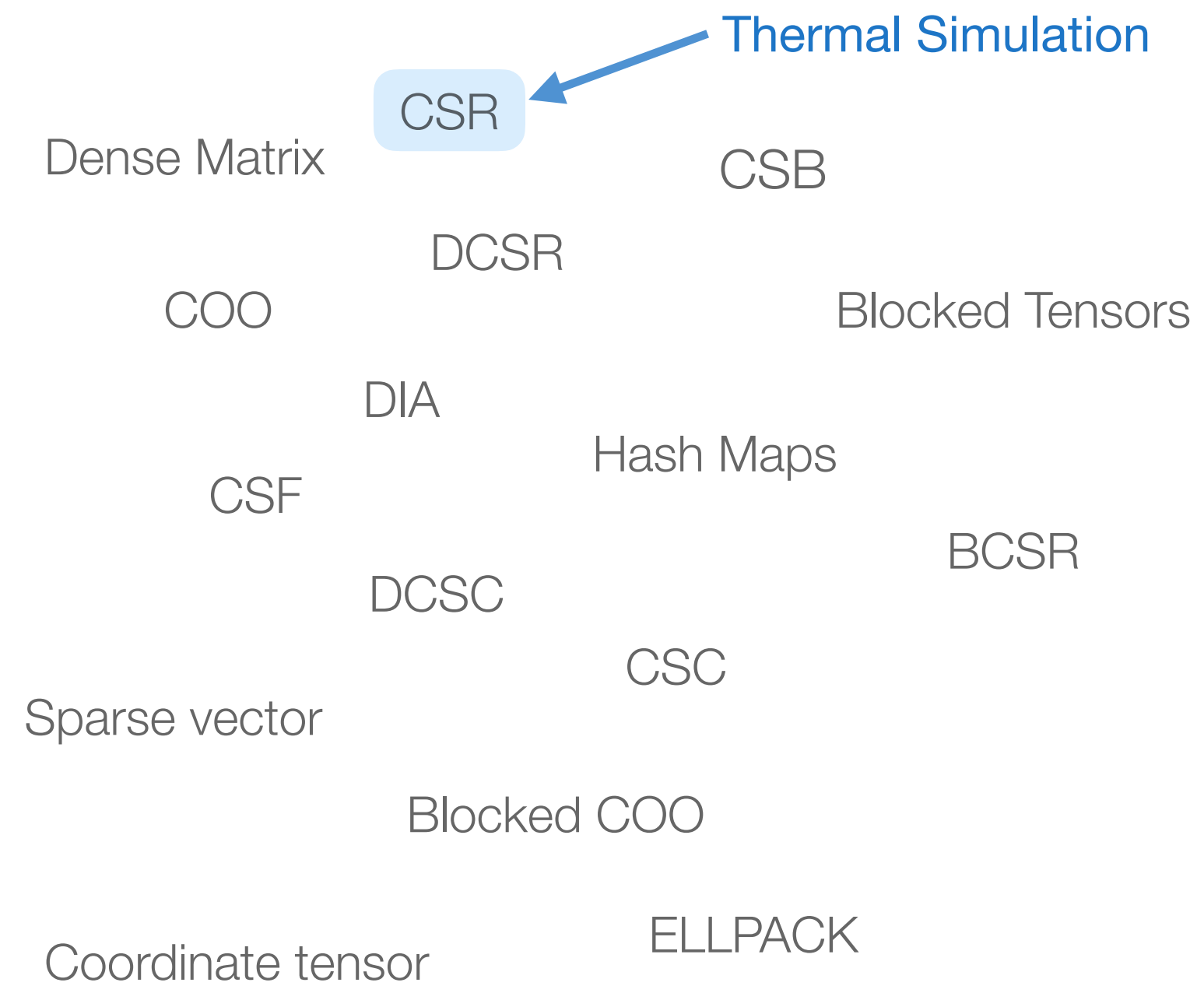
JPEG-LS

PNG

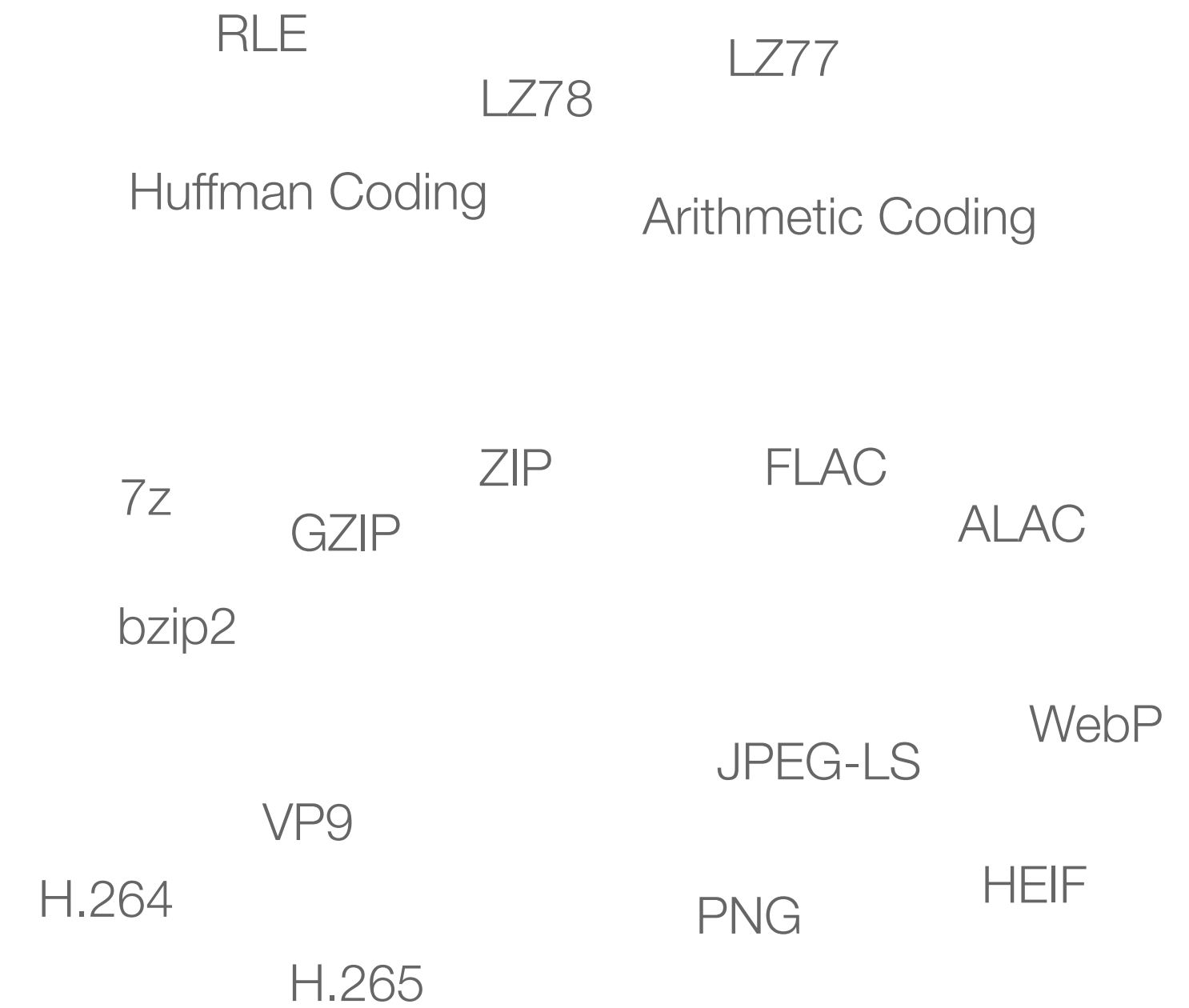
WebP

HEIF

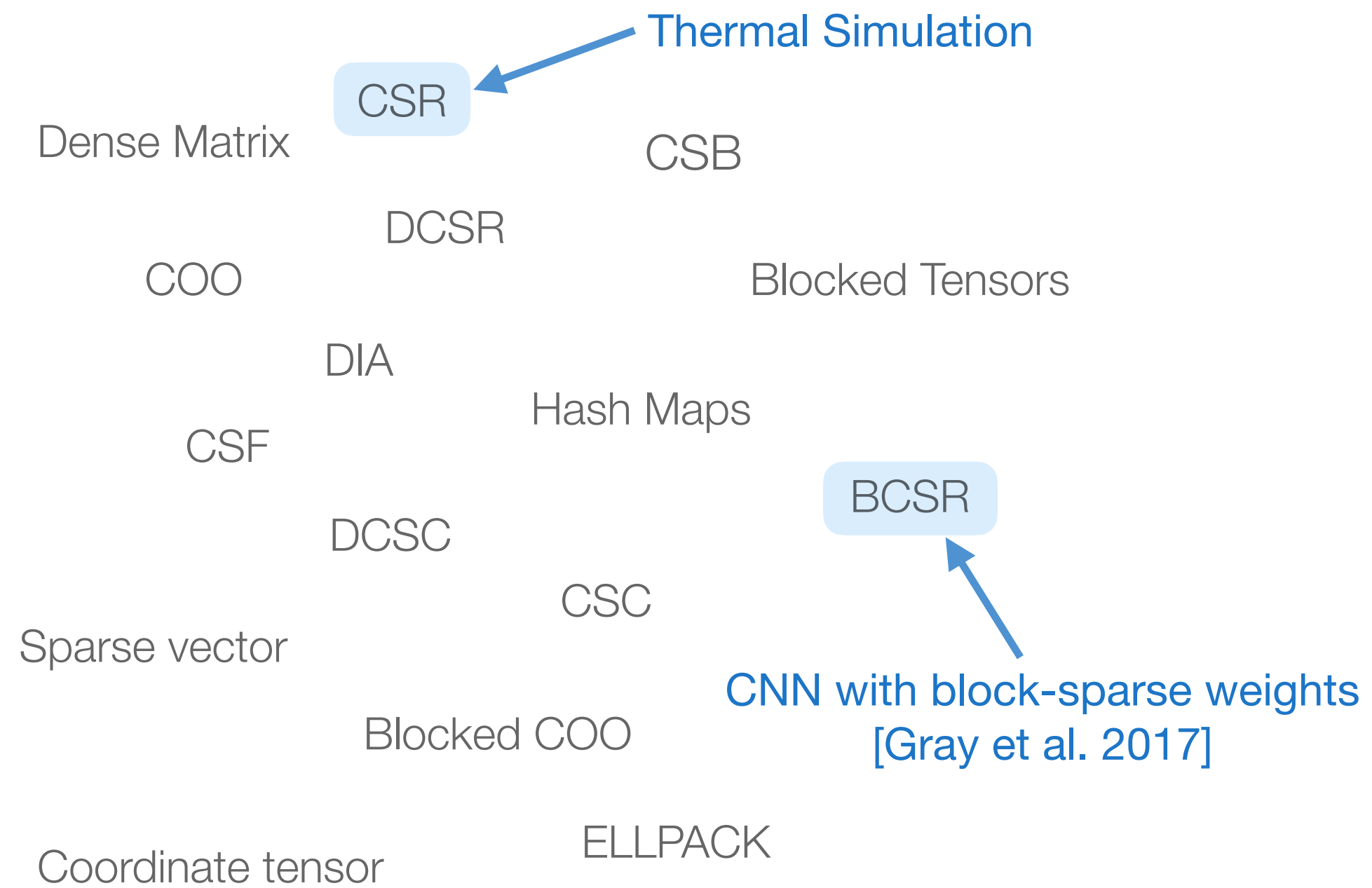
# Sparsity



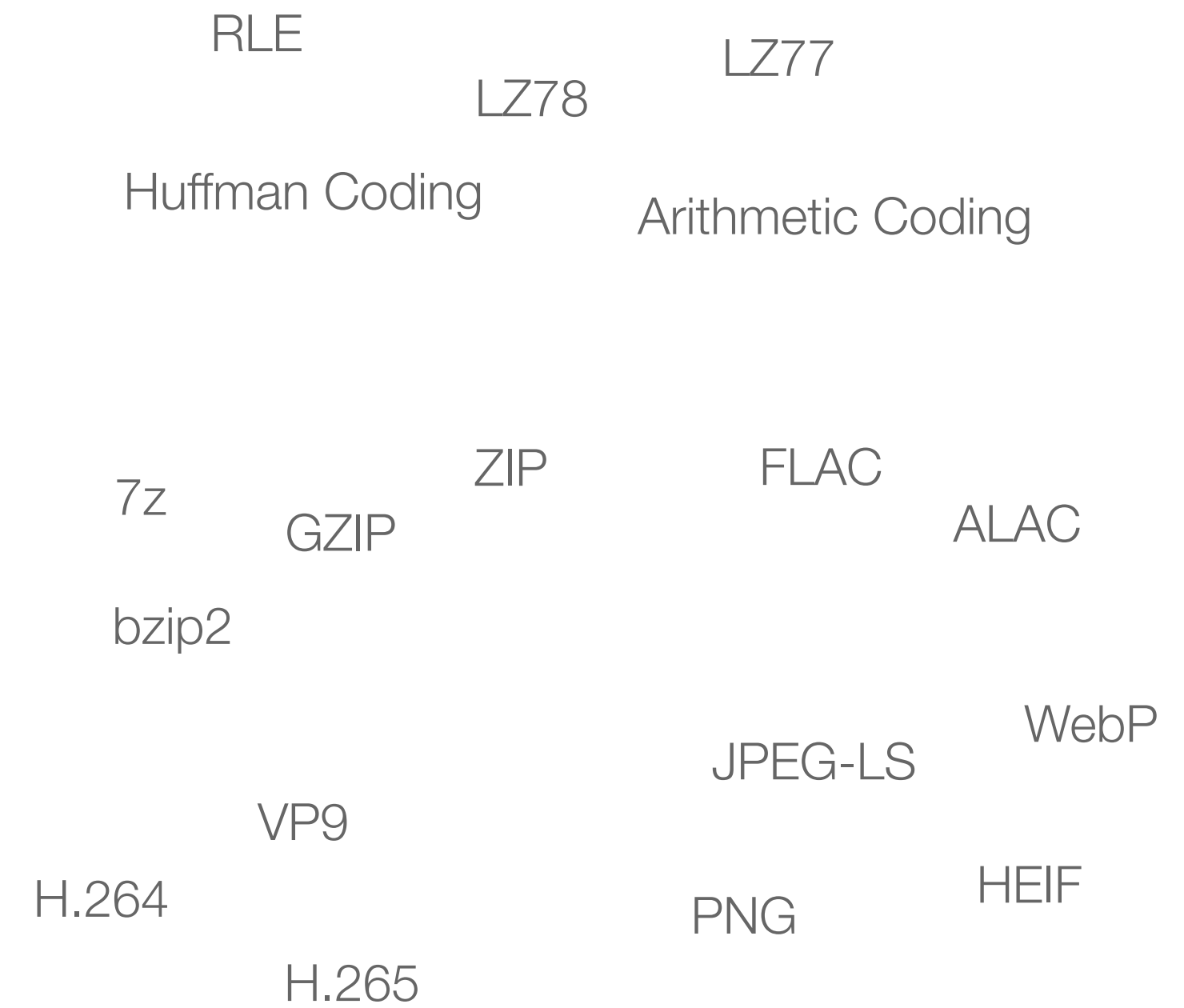
# Lossless Compression



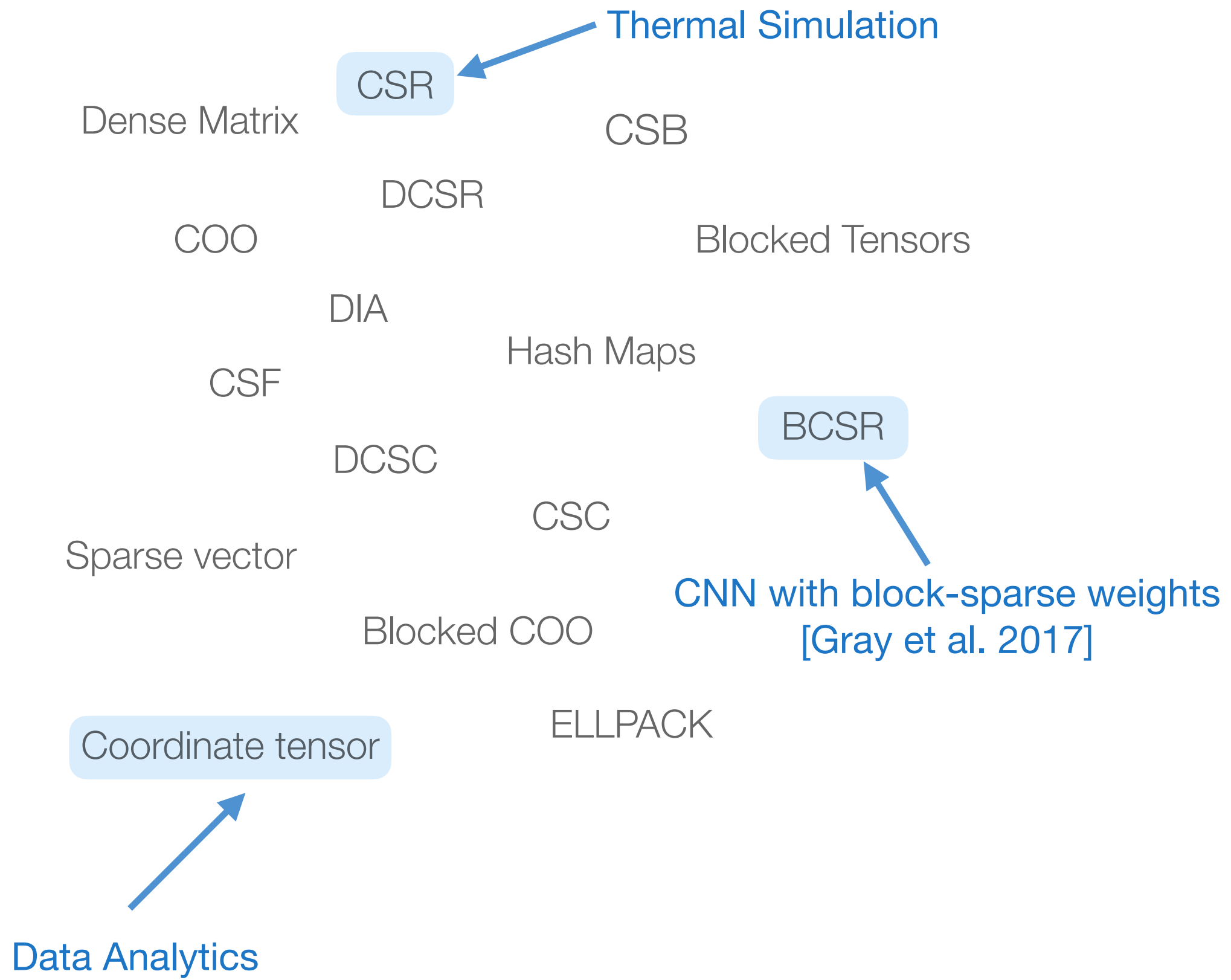
# Sparsity



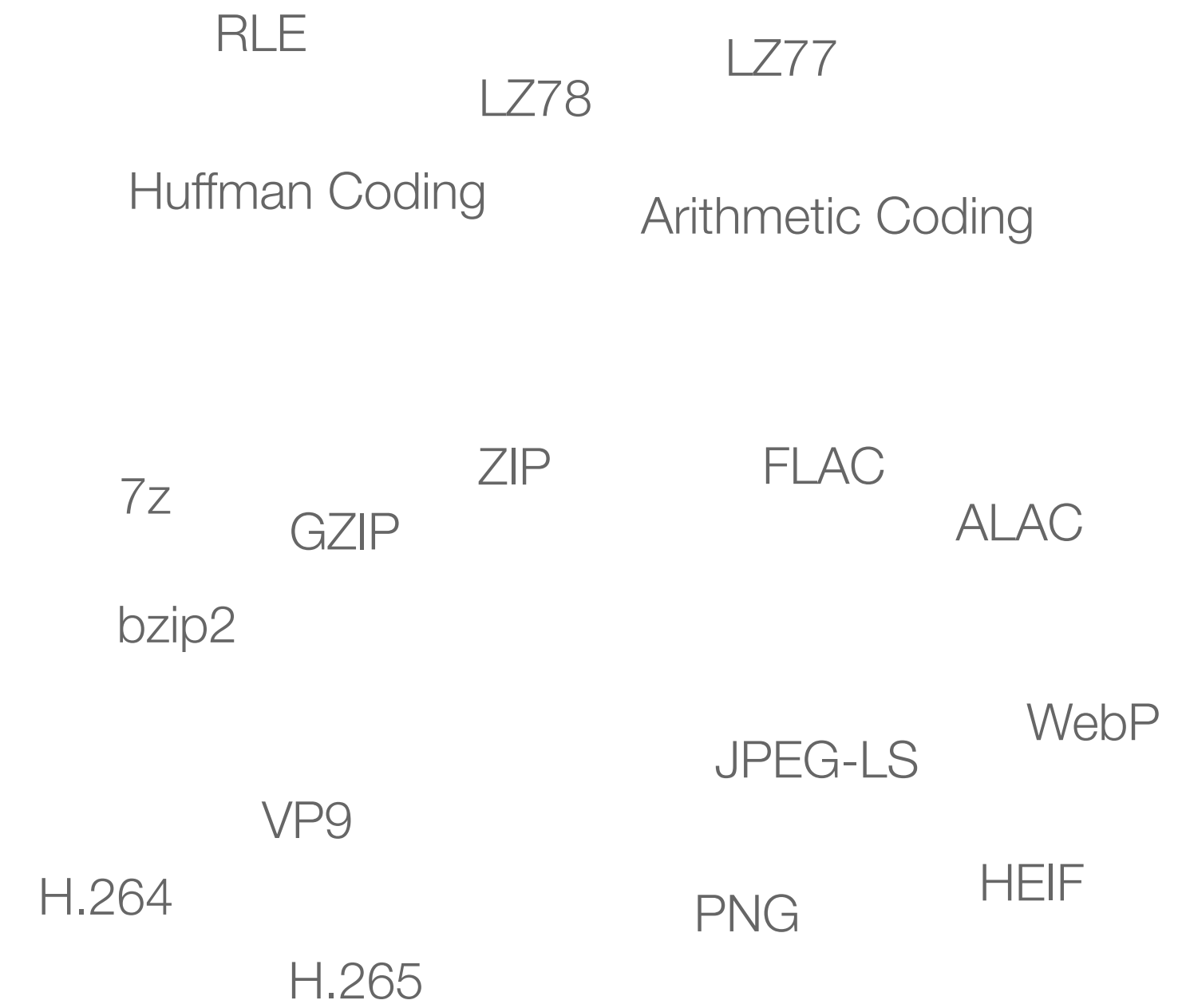
# Lossless Compression



# Sparsity

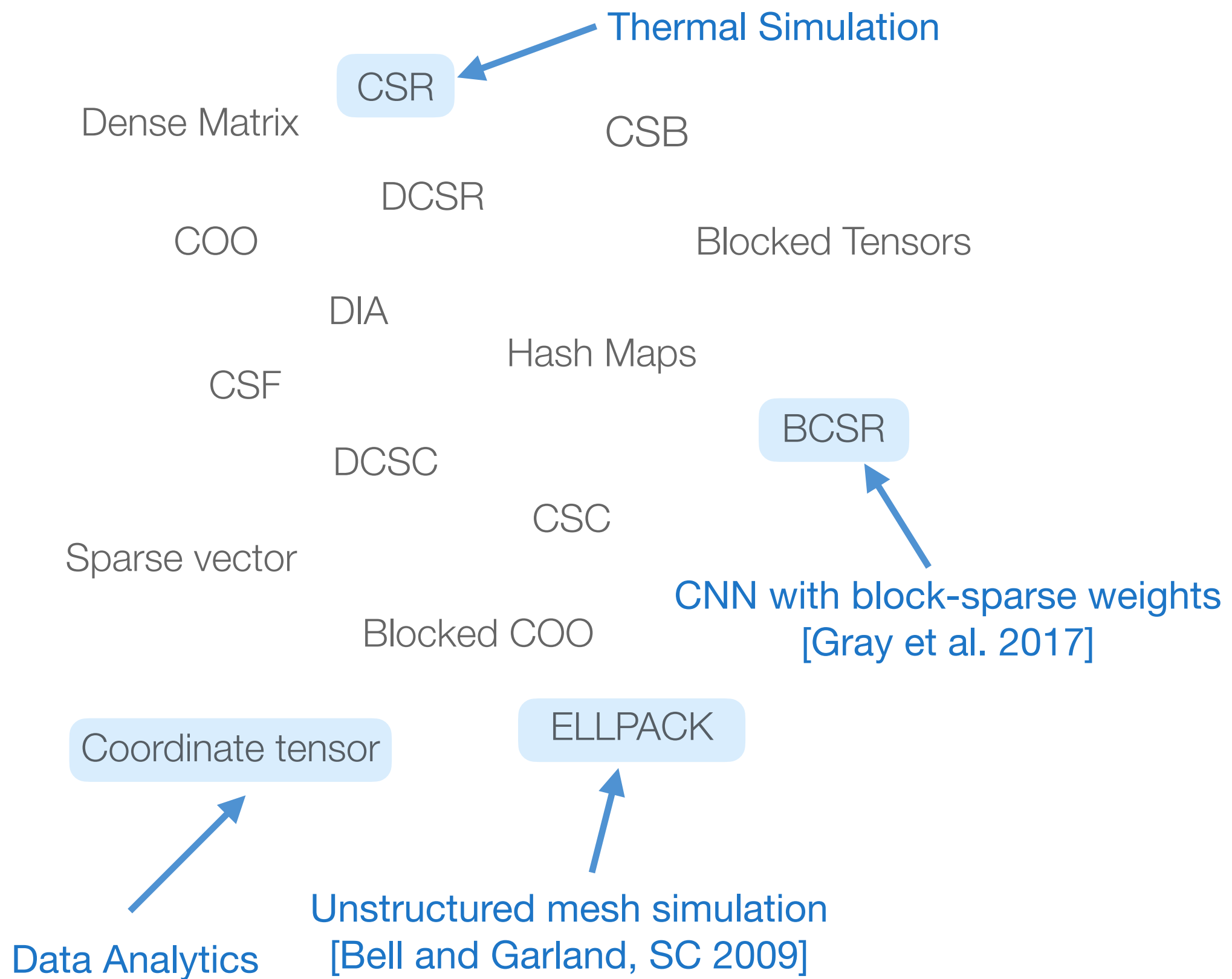


# Lossless Compression

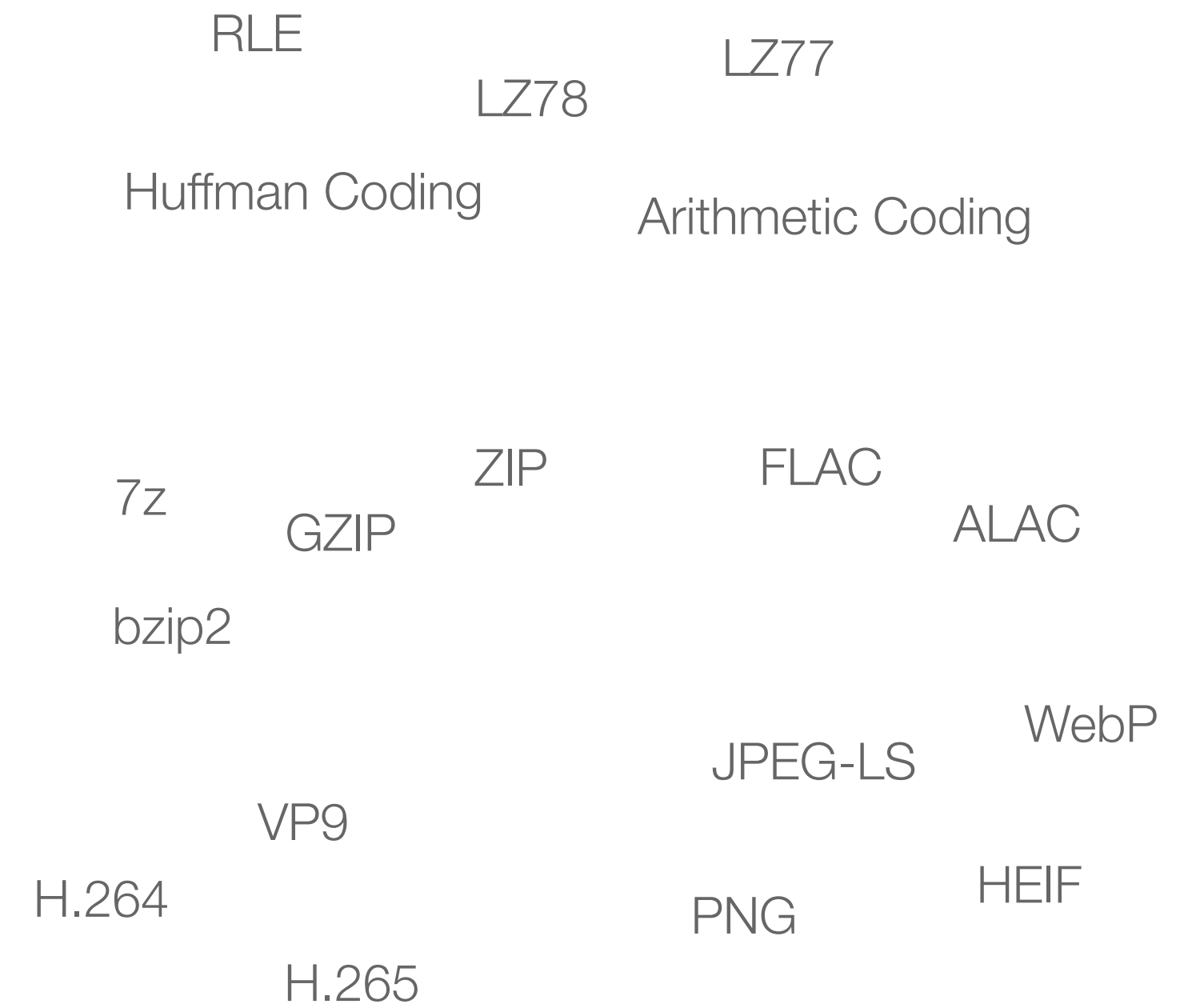




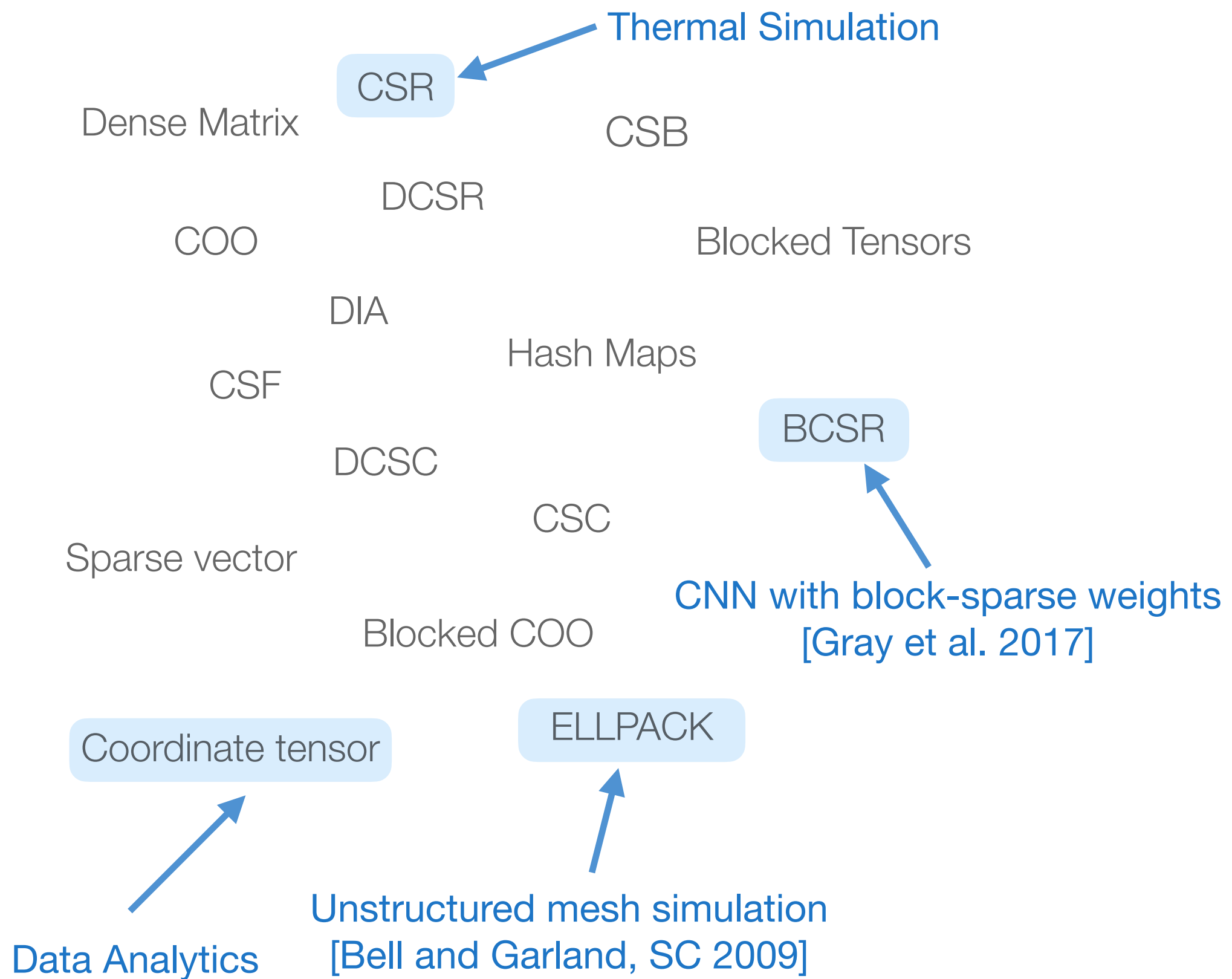
# Sparsity



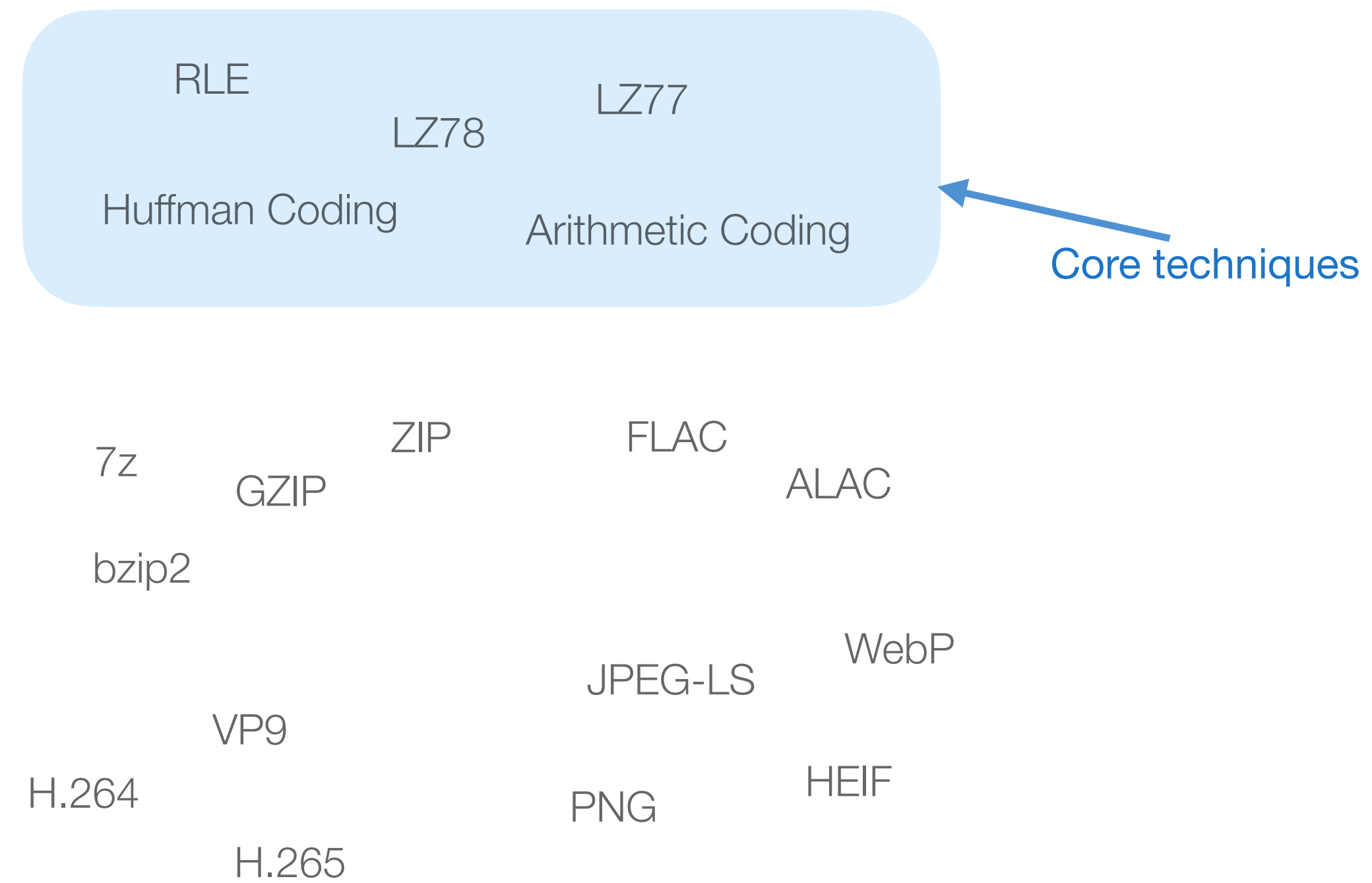
# Lossless Compression



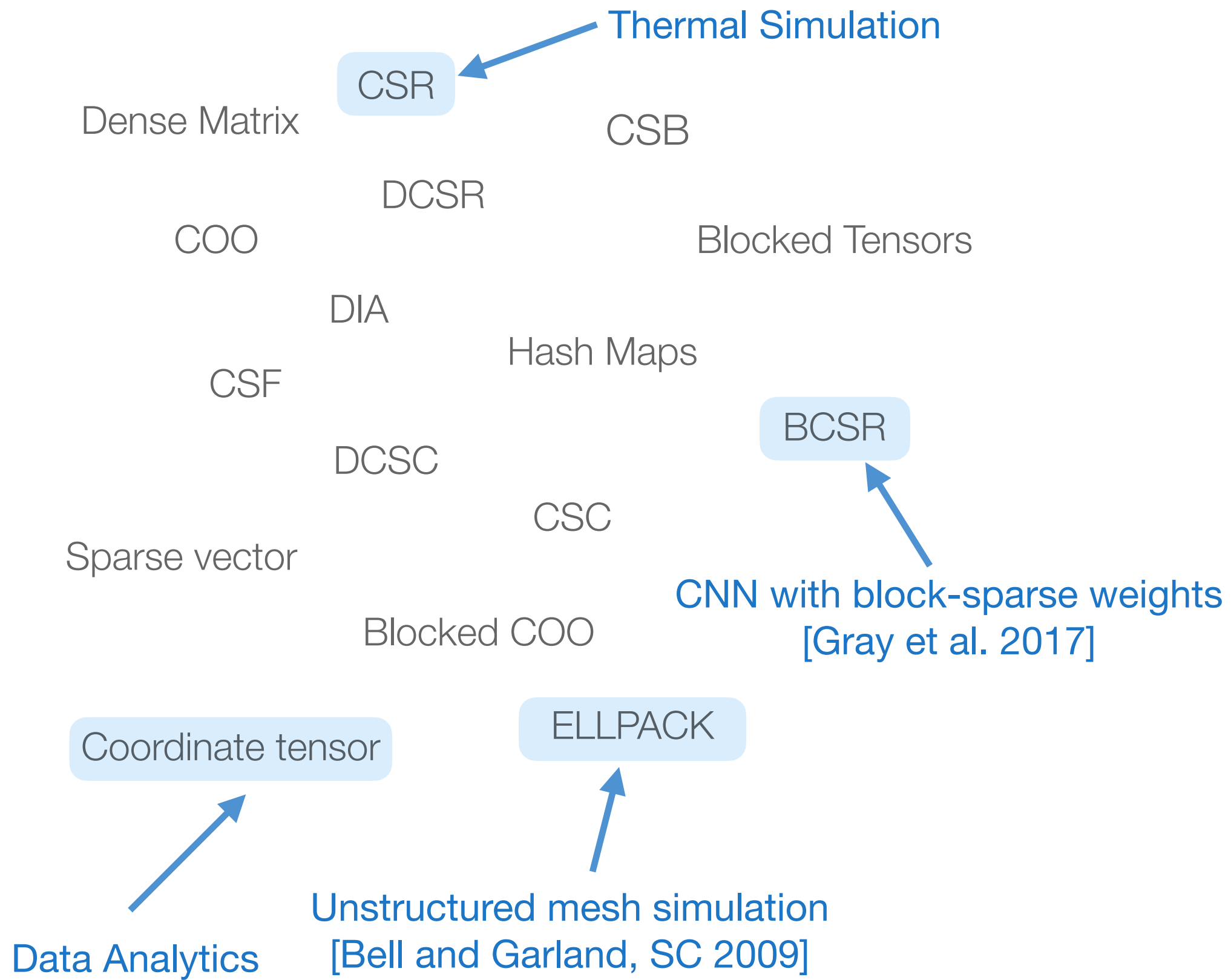
# Sparsity



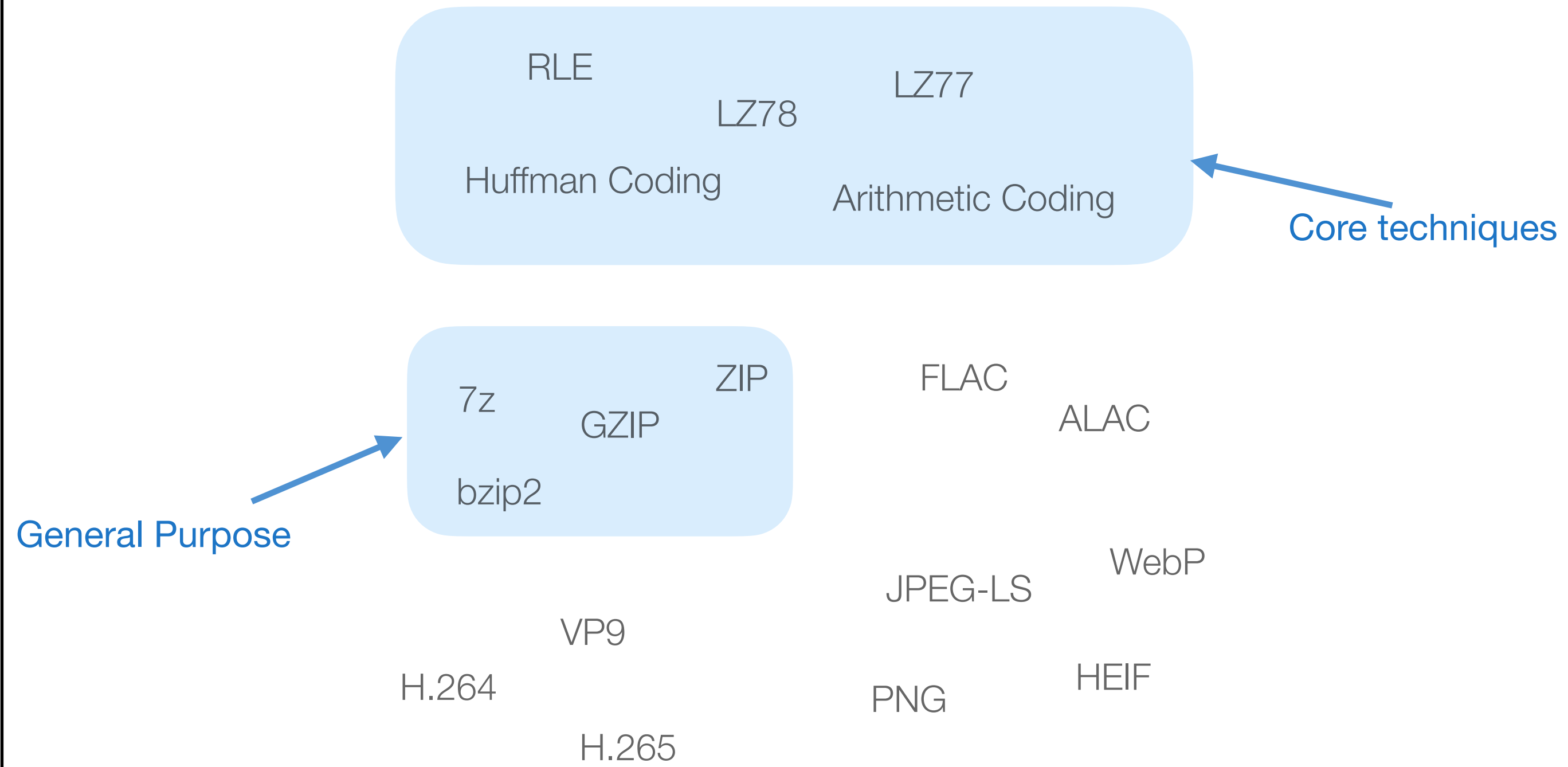
# Lossless Compression



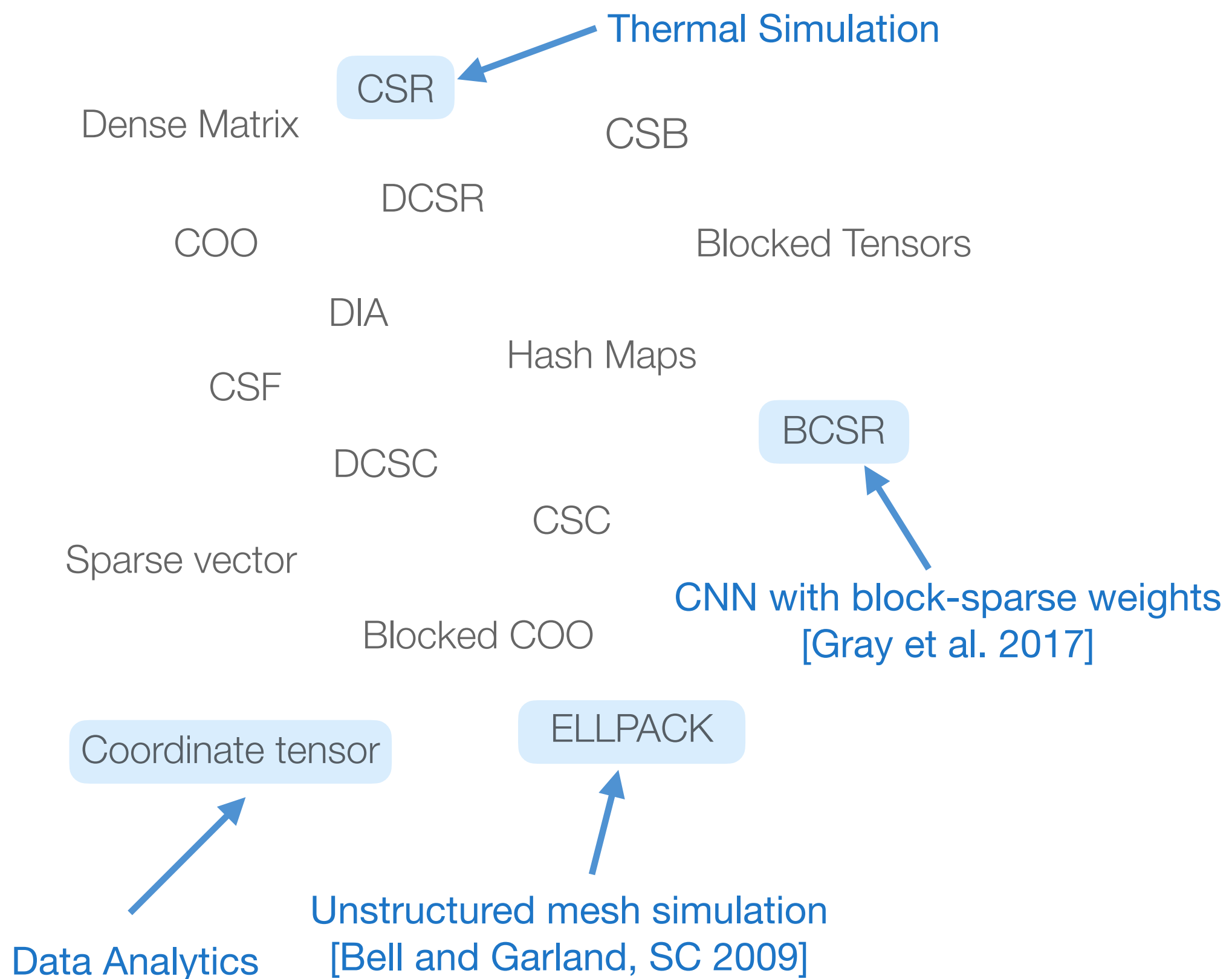
# Sparsity



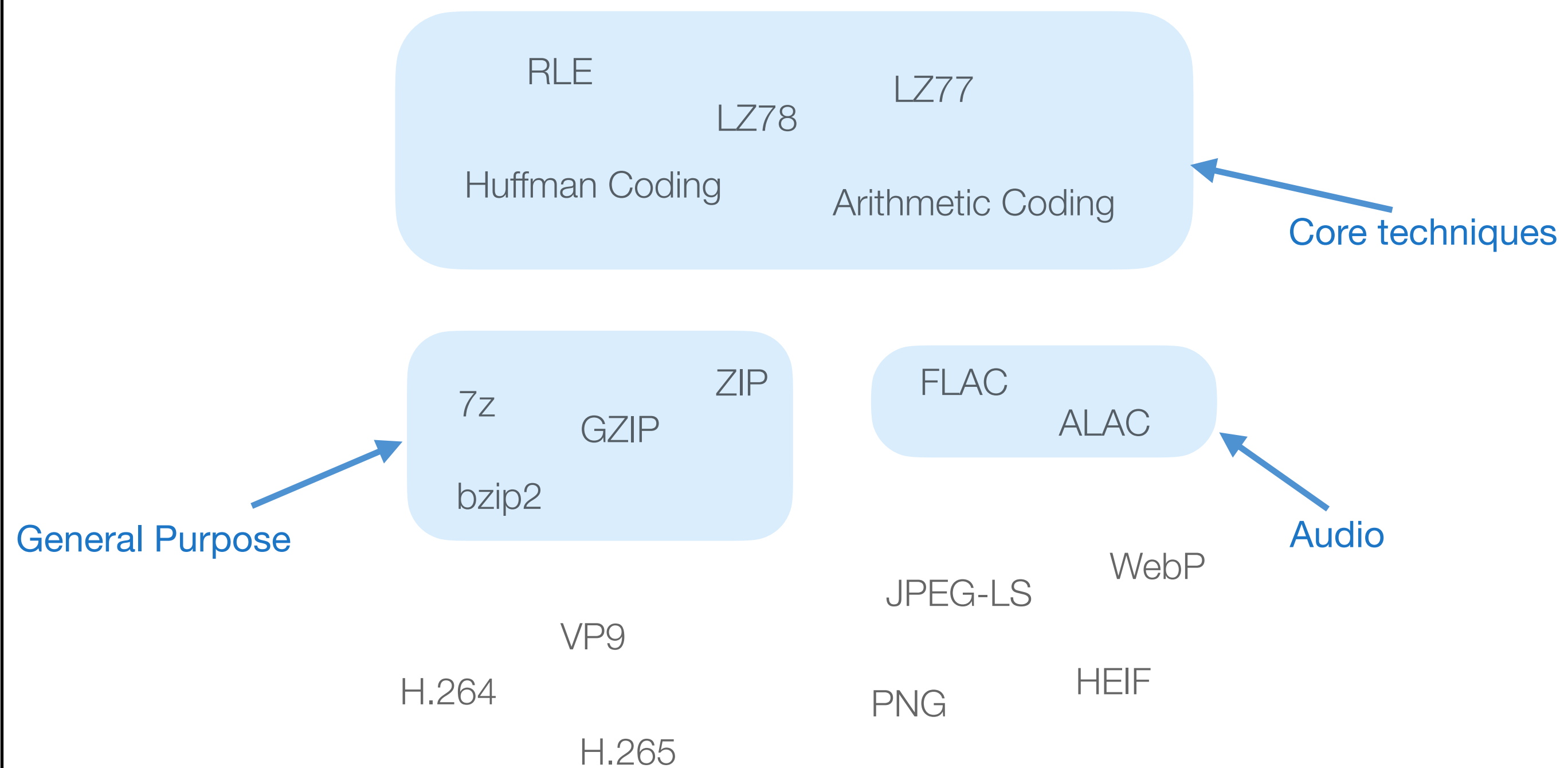
# Lossless Compression



# Sparsity

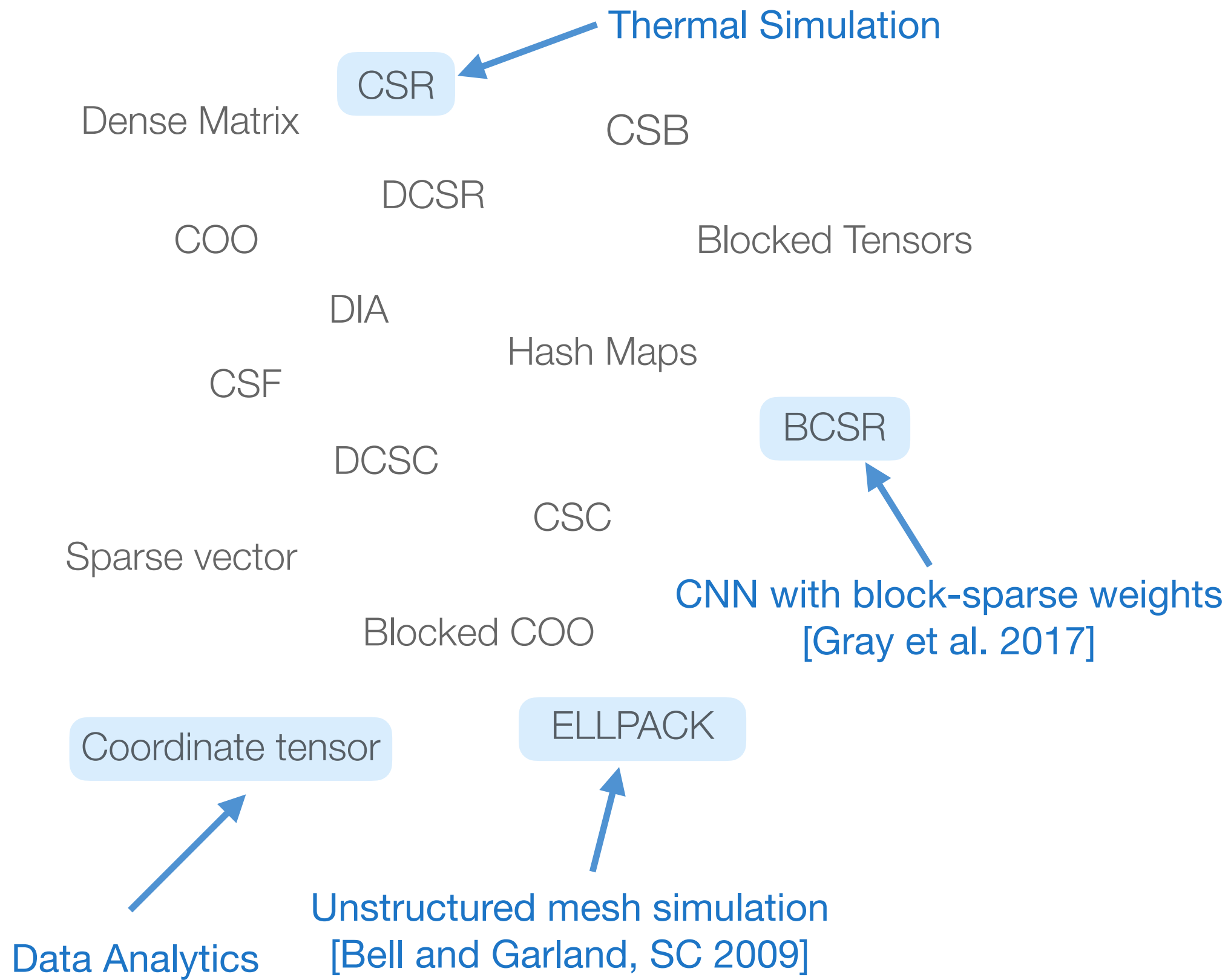


# Lossless Compression

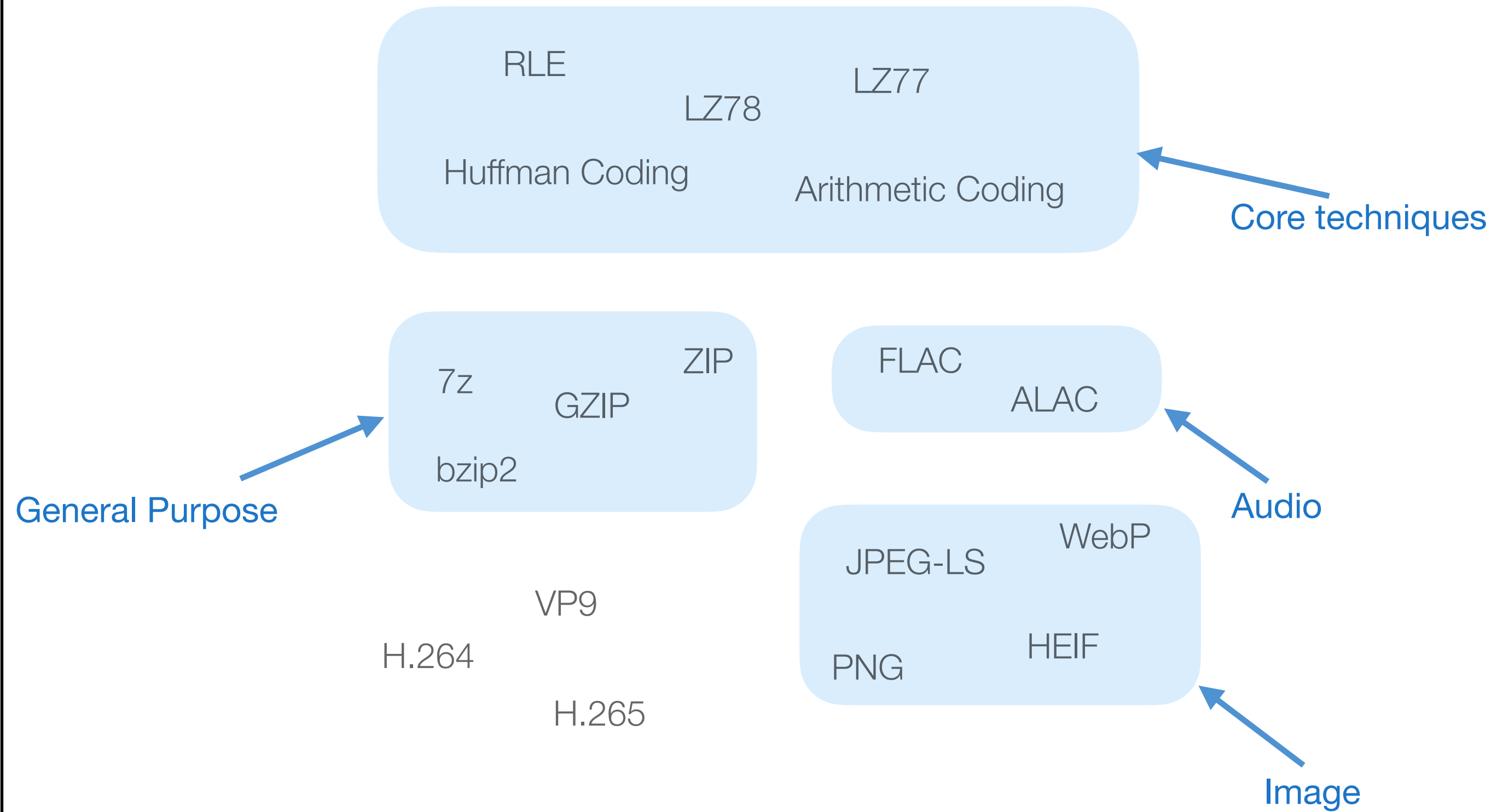




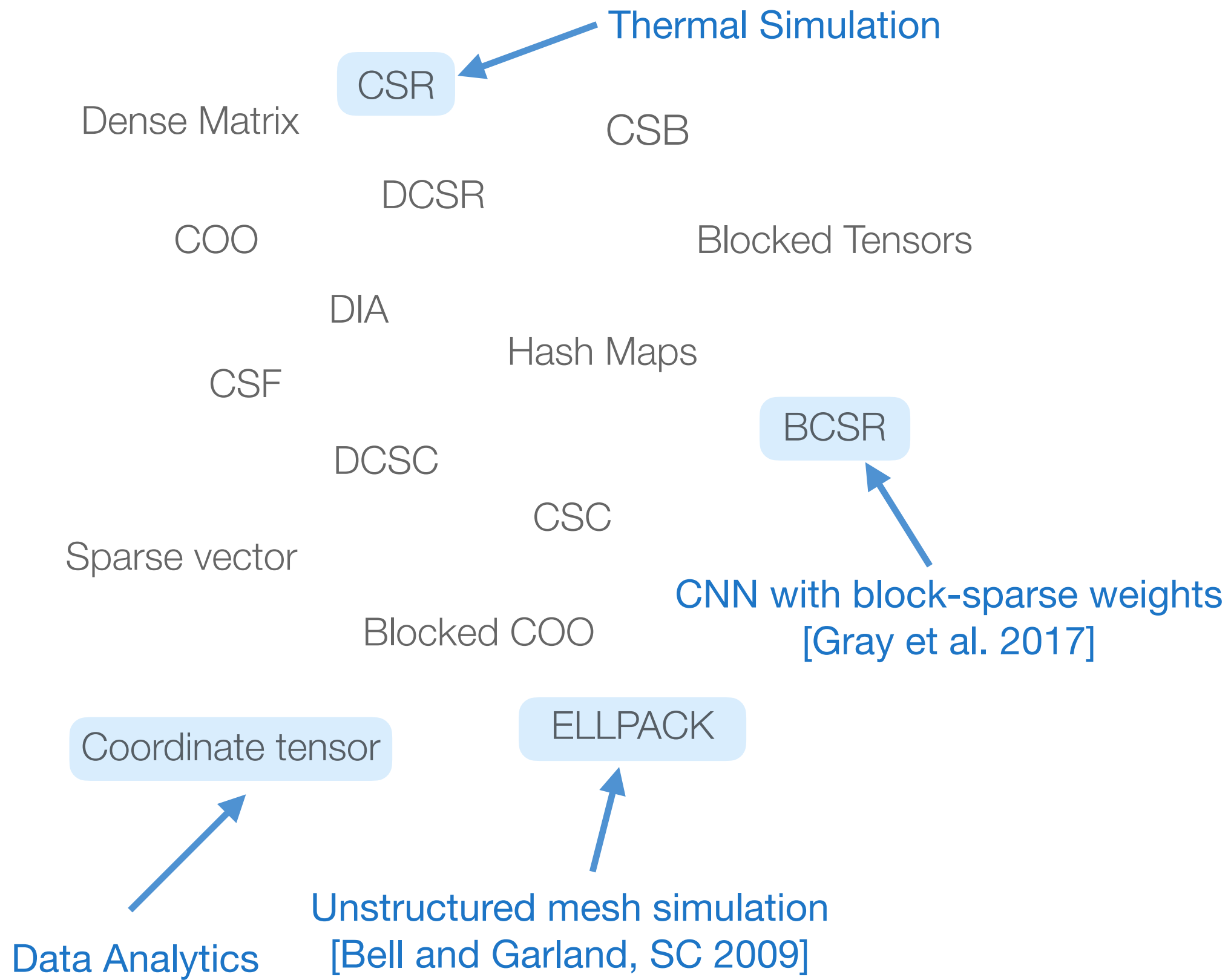
# Sparsity



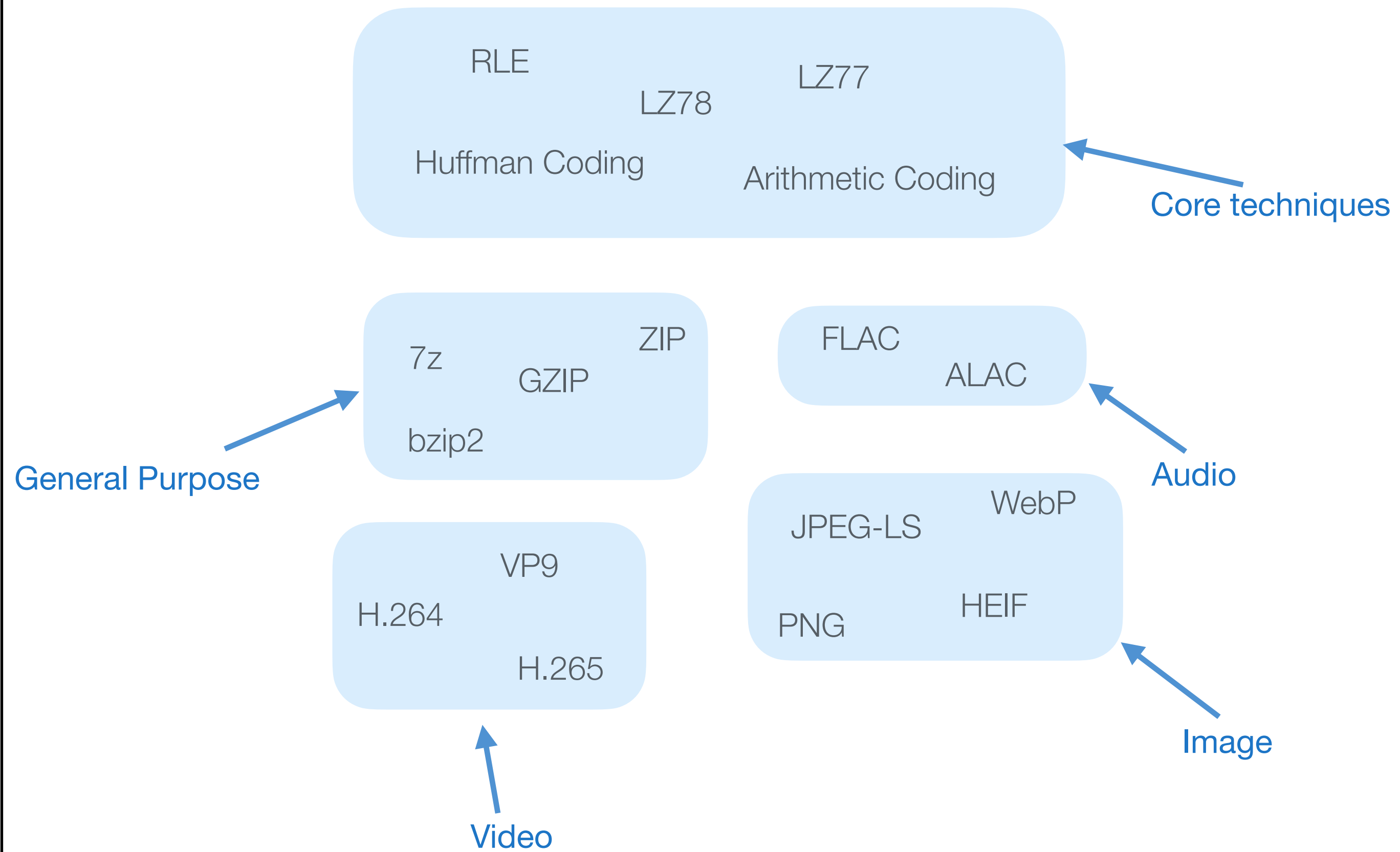
# Lossless Compression



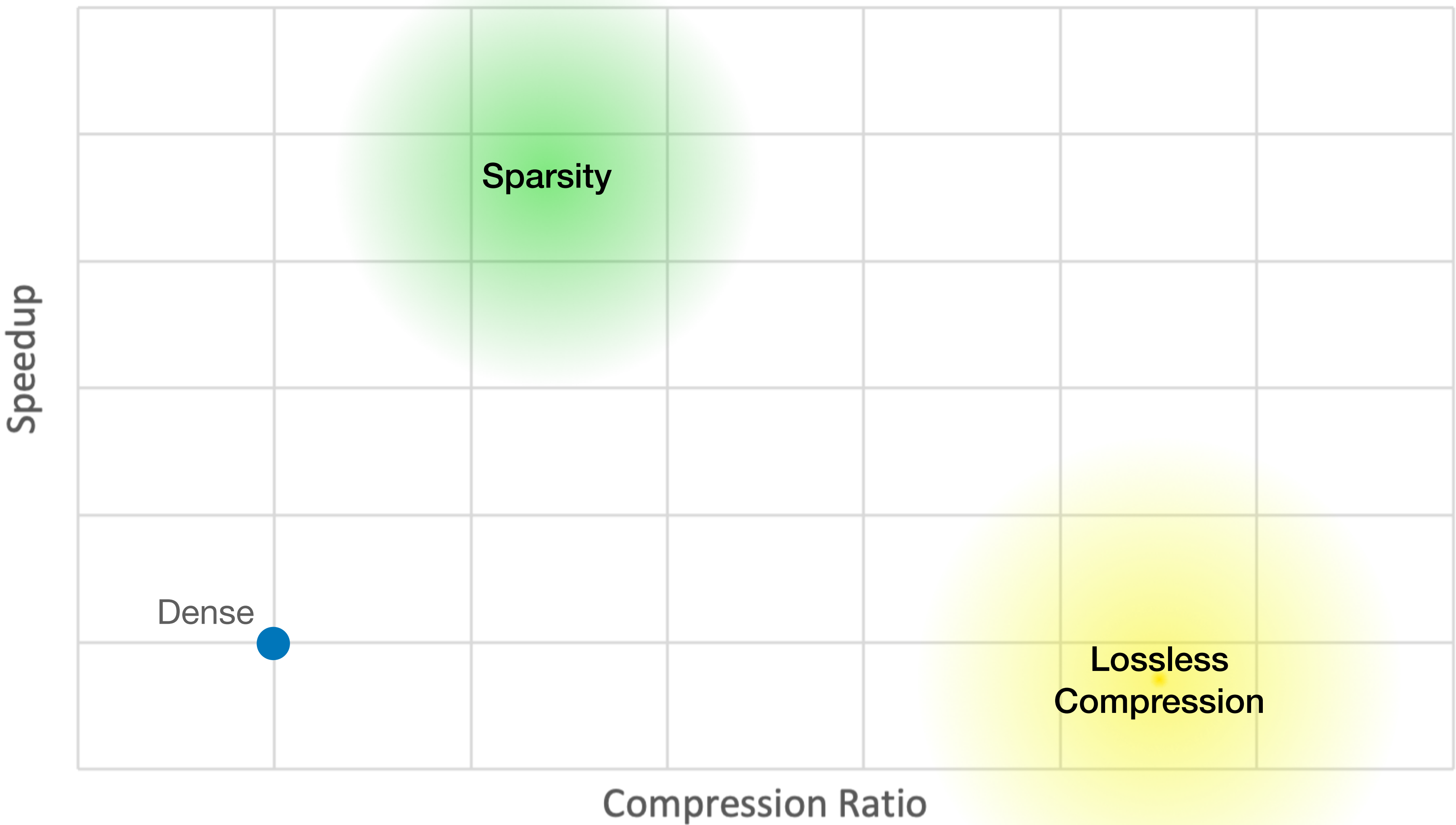
# Sparsity



# Lossless Compression

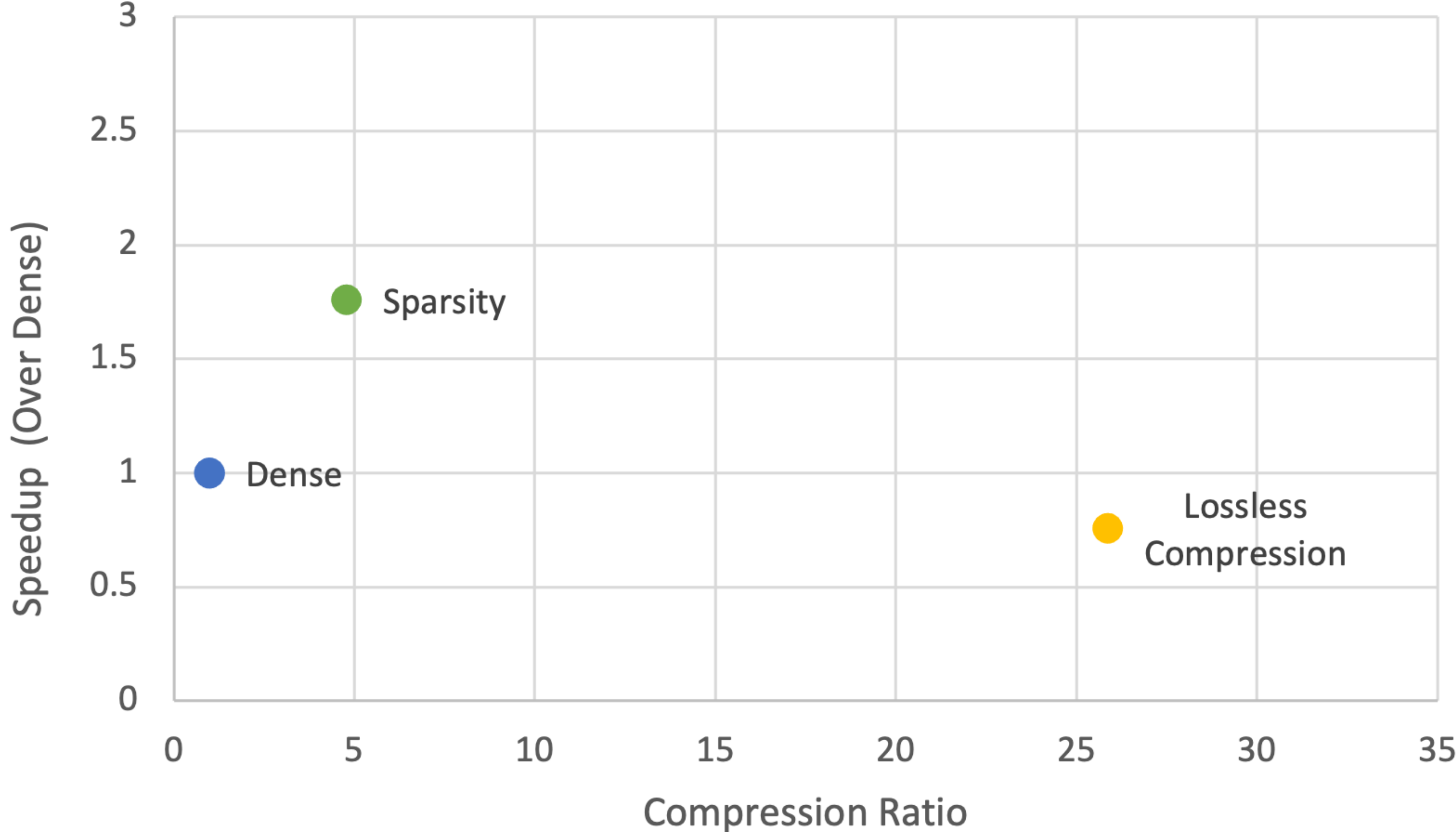


# Sparsity and Compression have traditionally had different goals



# Sparsity and Compression have traditionally had different goals

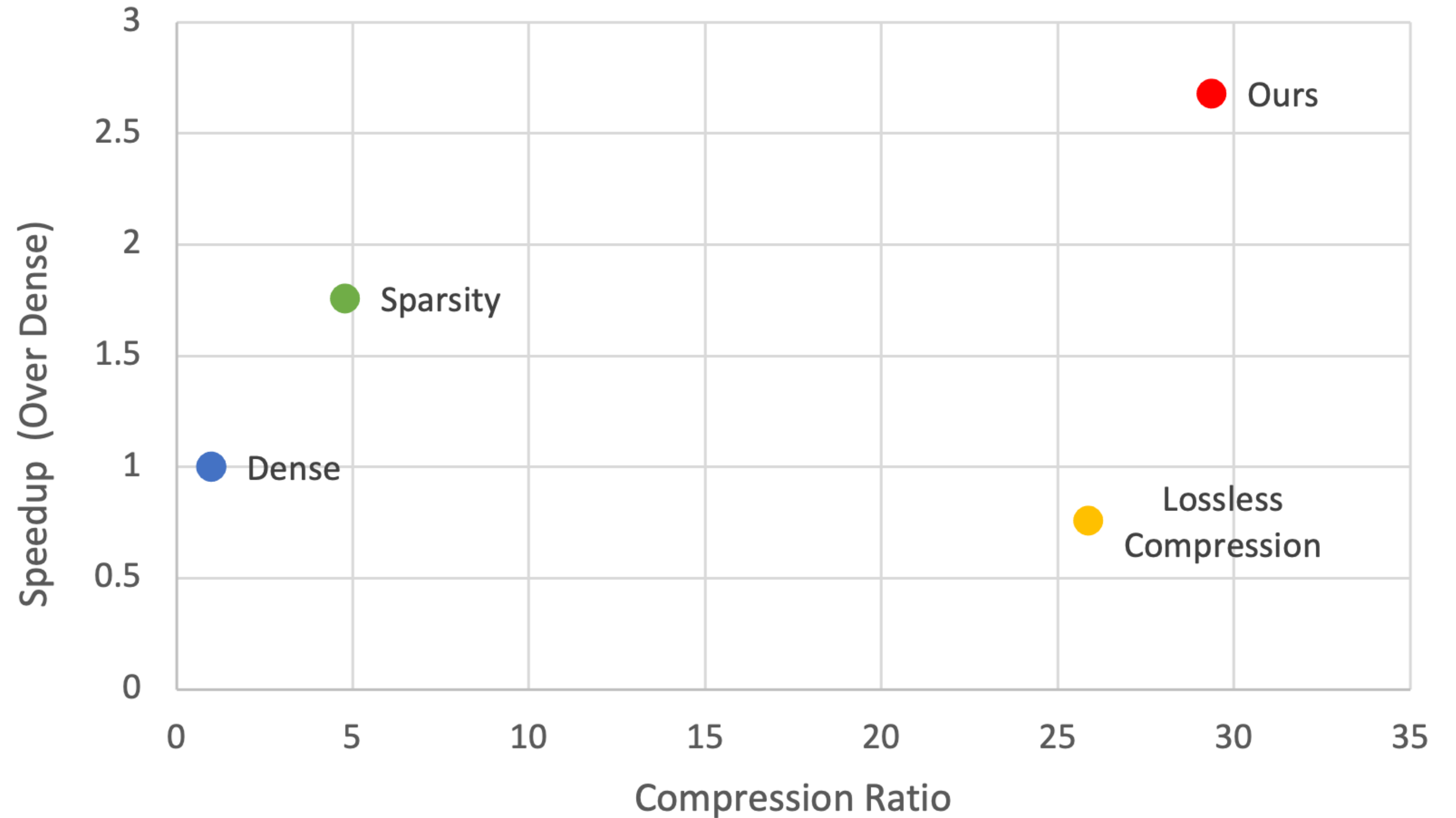
Edge Detection





# Unifying both can result in the best of both compression and performance

Edge Detection

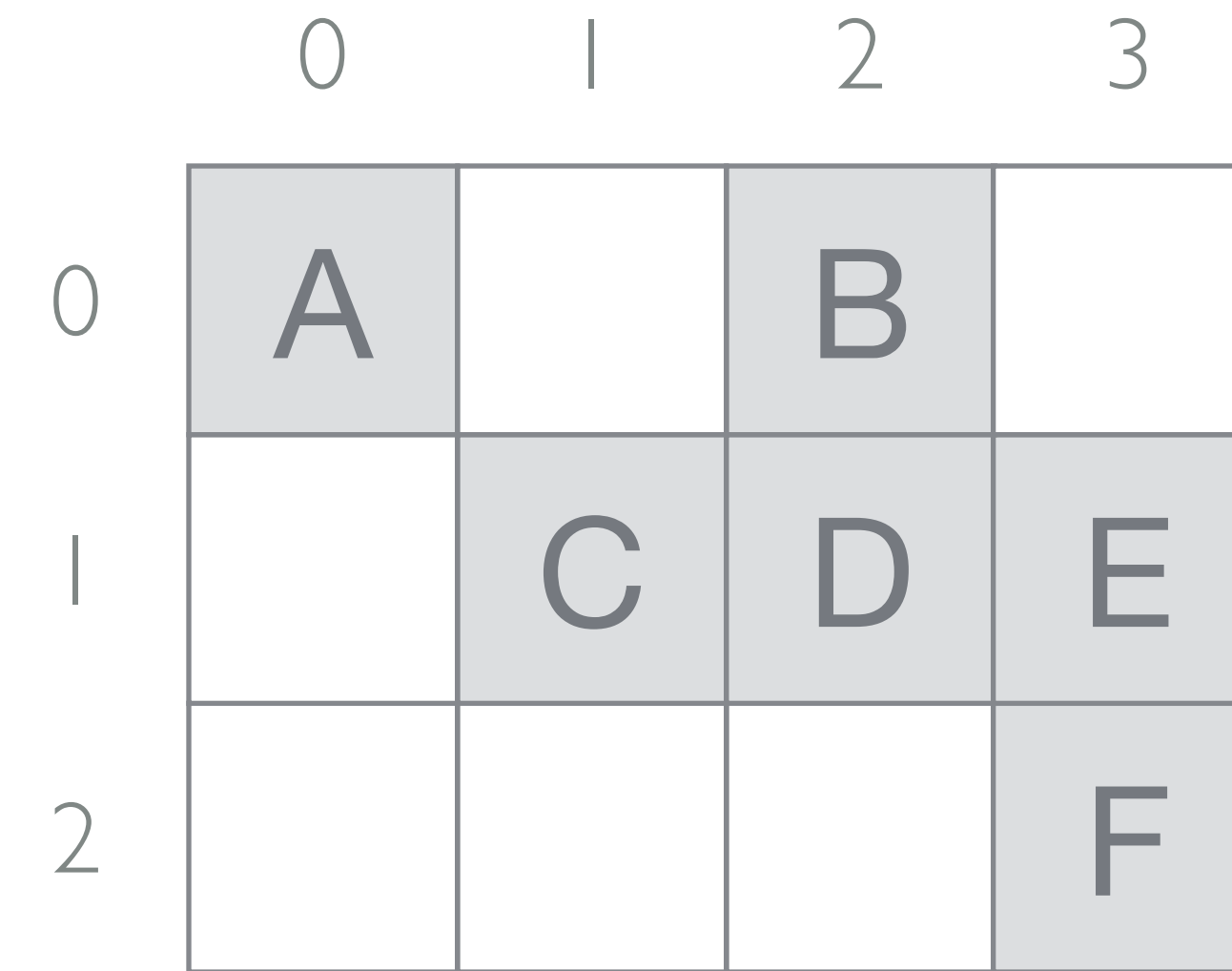
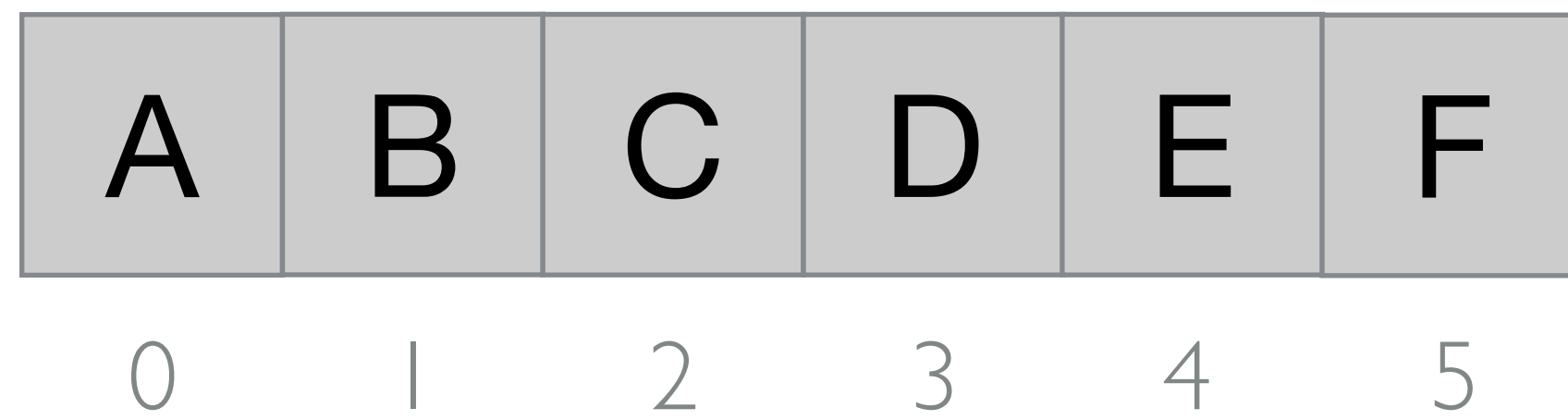


# Sparse tensors can be compressed by adding metadata

	0	1	2	3
0	A		B	
1		C	D	E
2				F

A		B			C	D	E				F
0	1	2	3	4	5	6	7	8	9	10	11

# Sparse tensors can be compressed by adding metadata



# Sparse tensors can be compressed by adding metadata

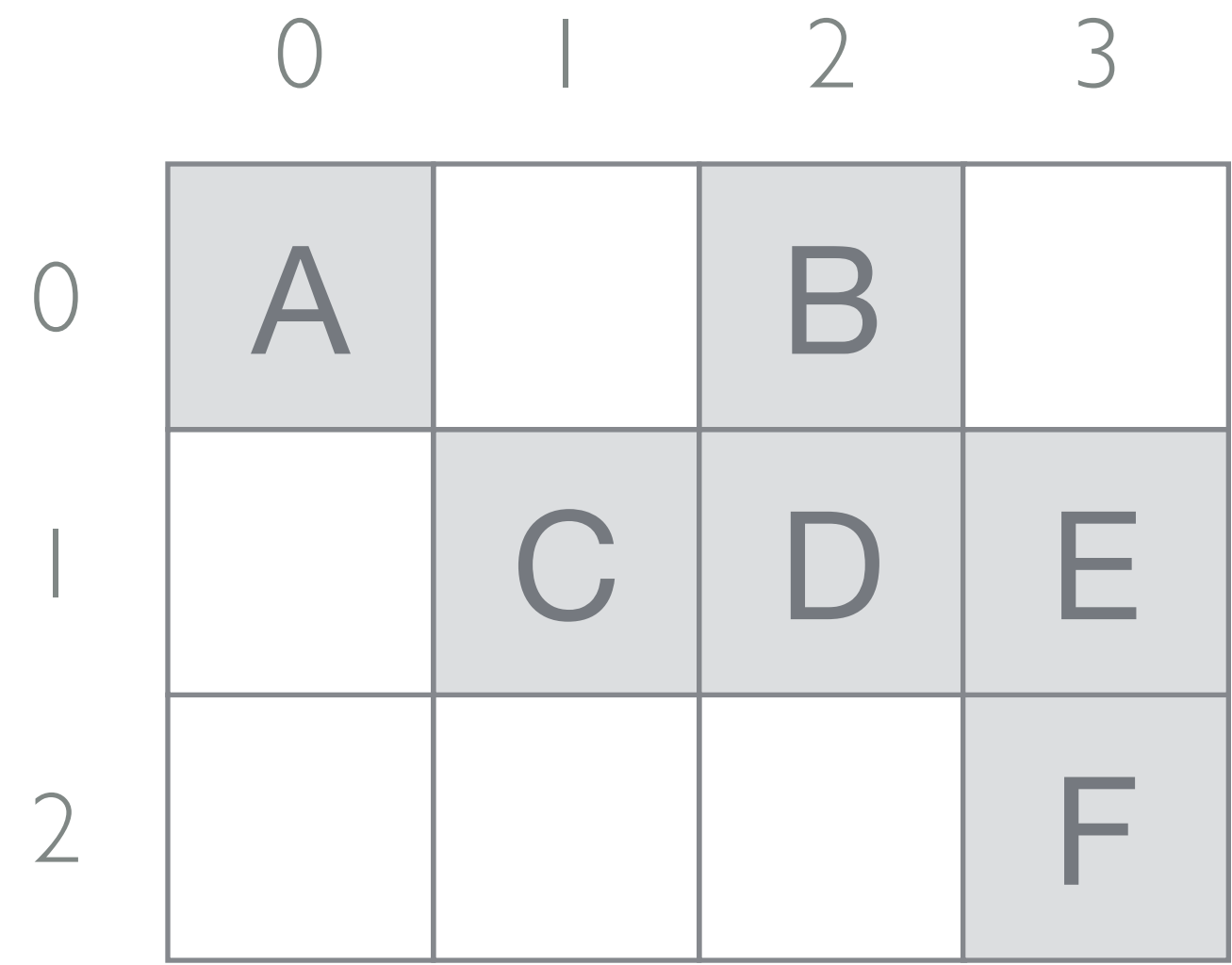
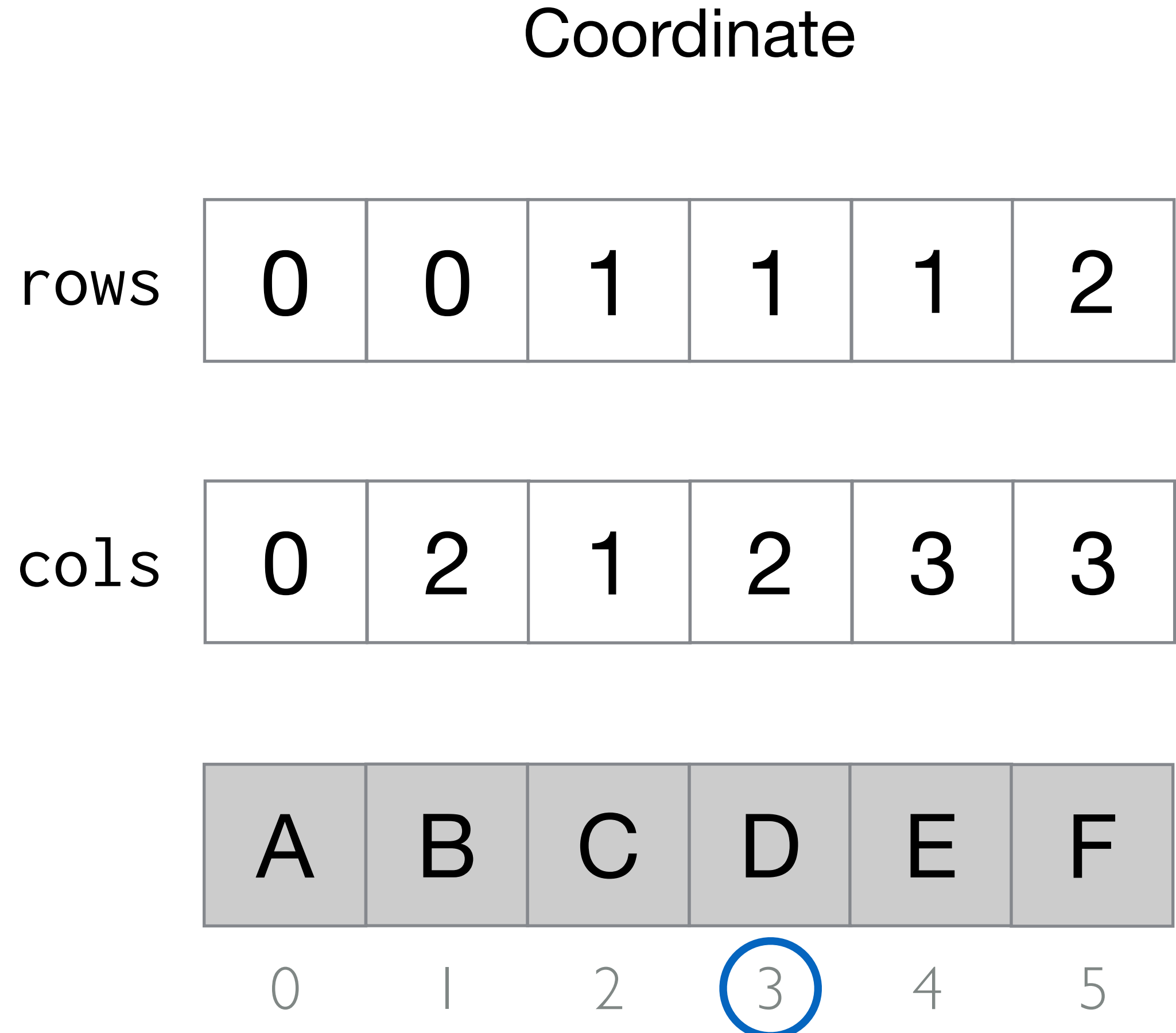
row(3) = ???

col(3) = ???

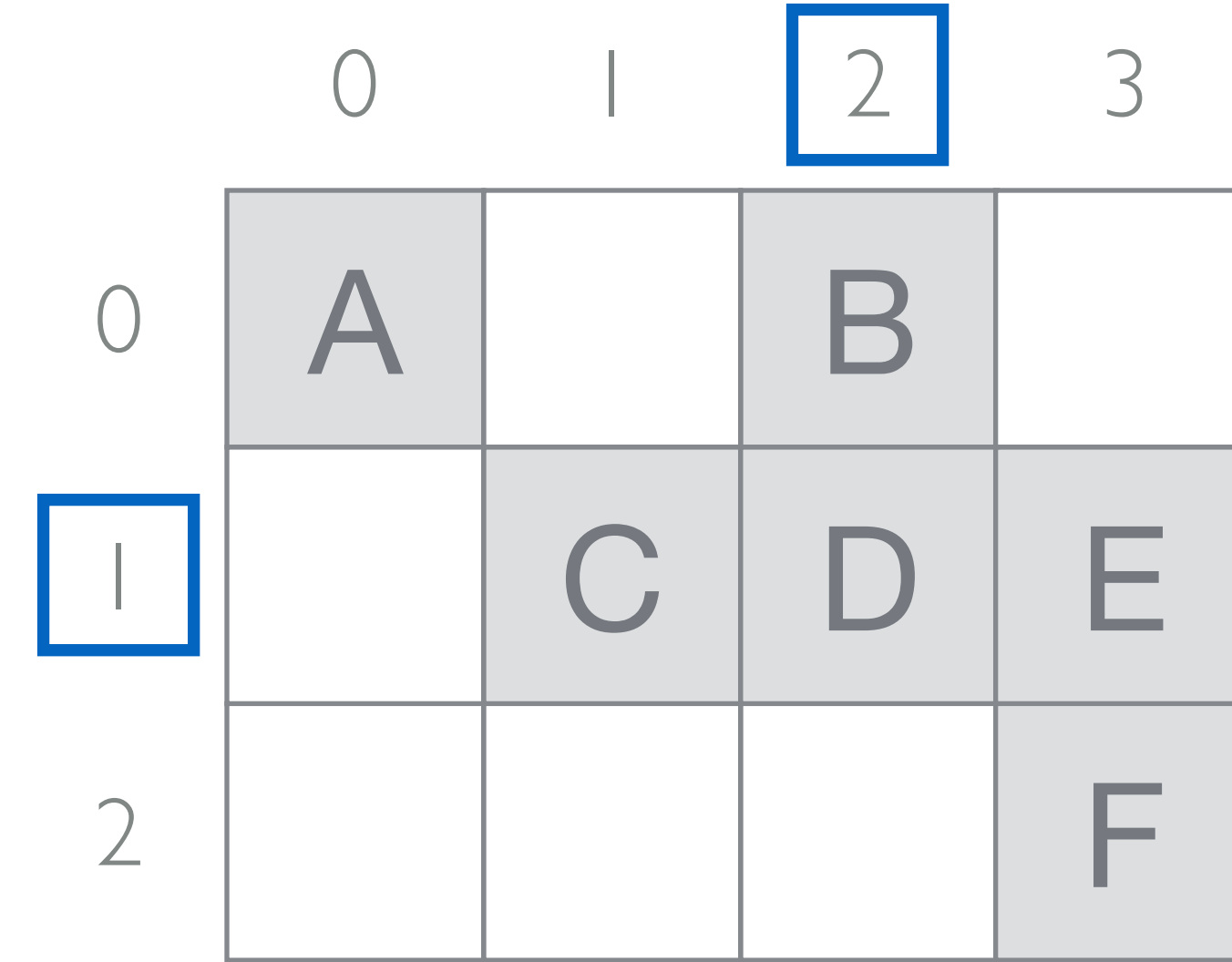
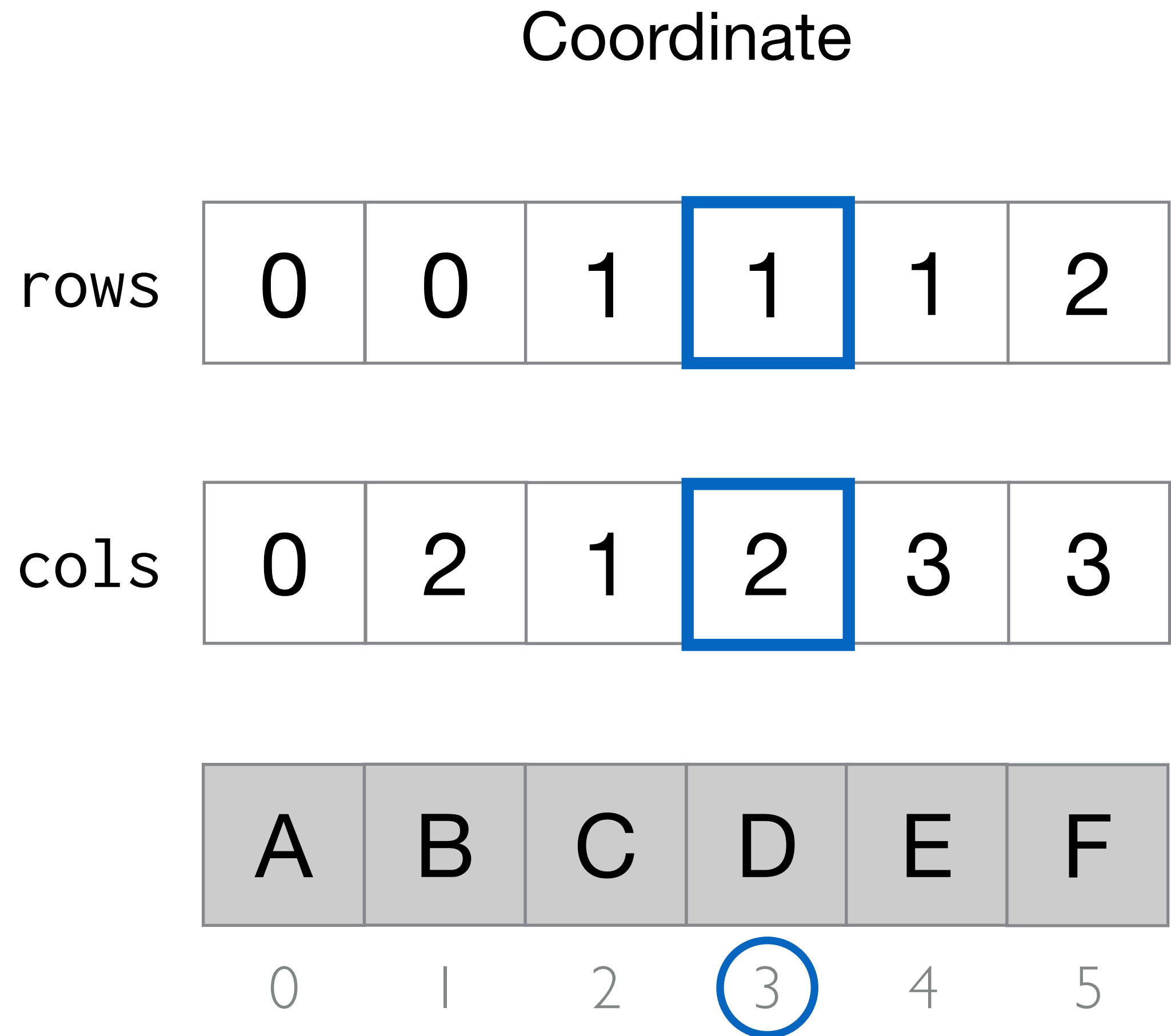
	0	1	2	3
0	A		B	
1		C	D	E
2				F

A	B	C	D	E	F
0	1	2	3	4	5

# Sparse tensors can be compressed by adding metadata

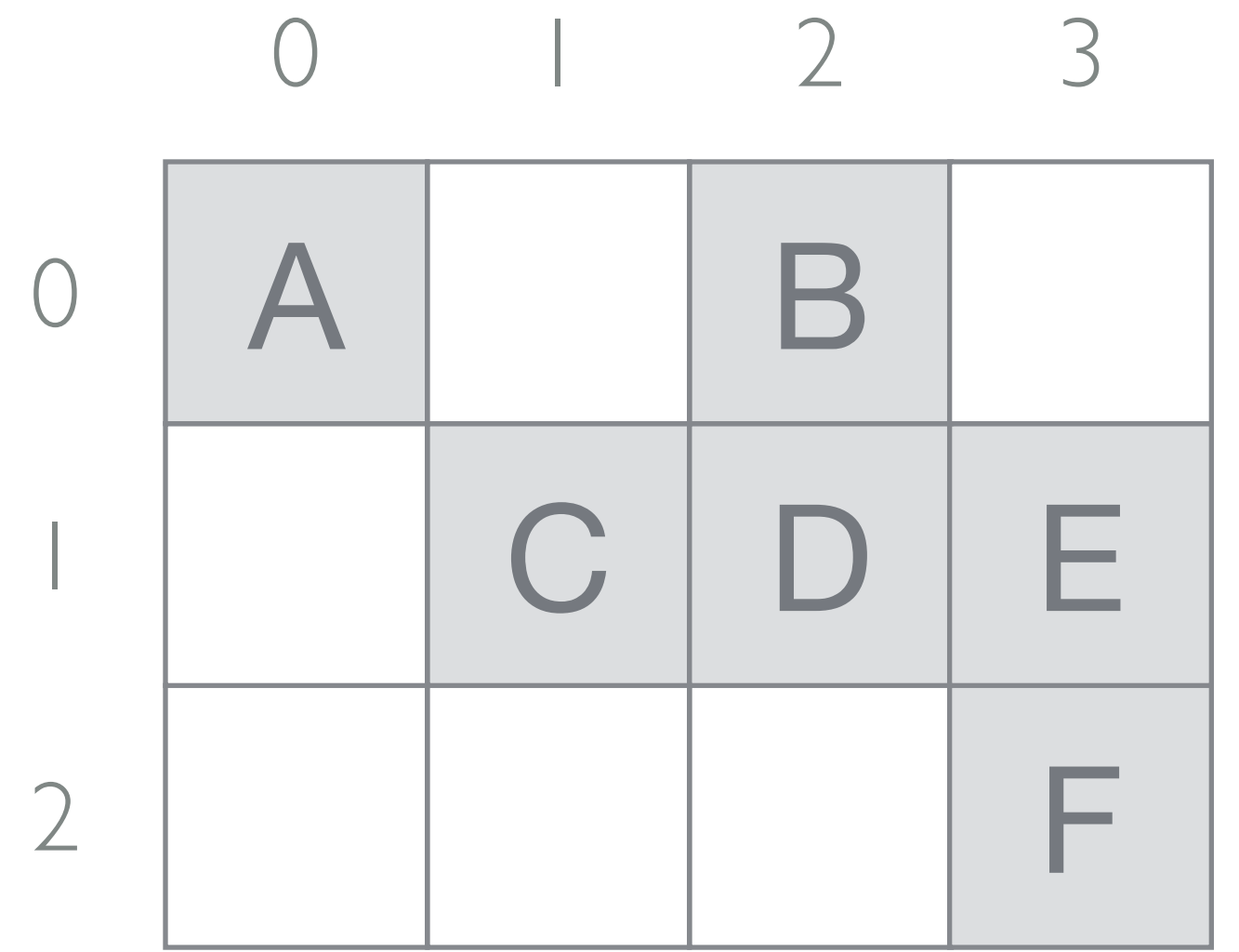
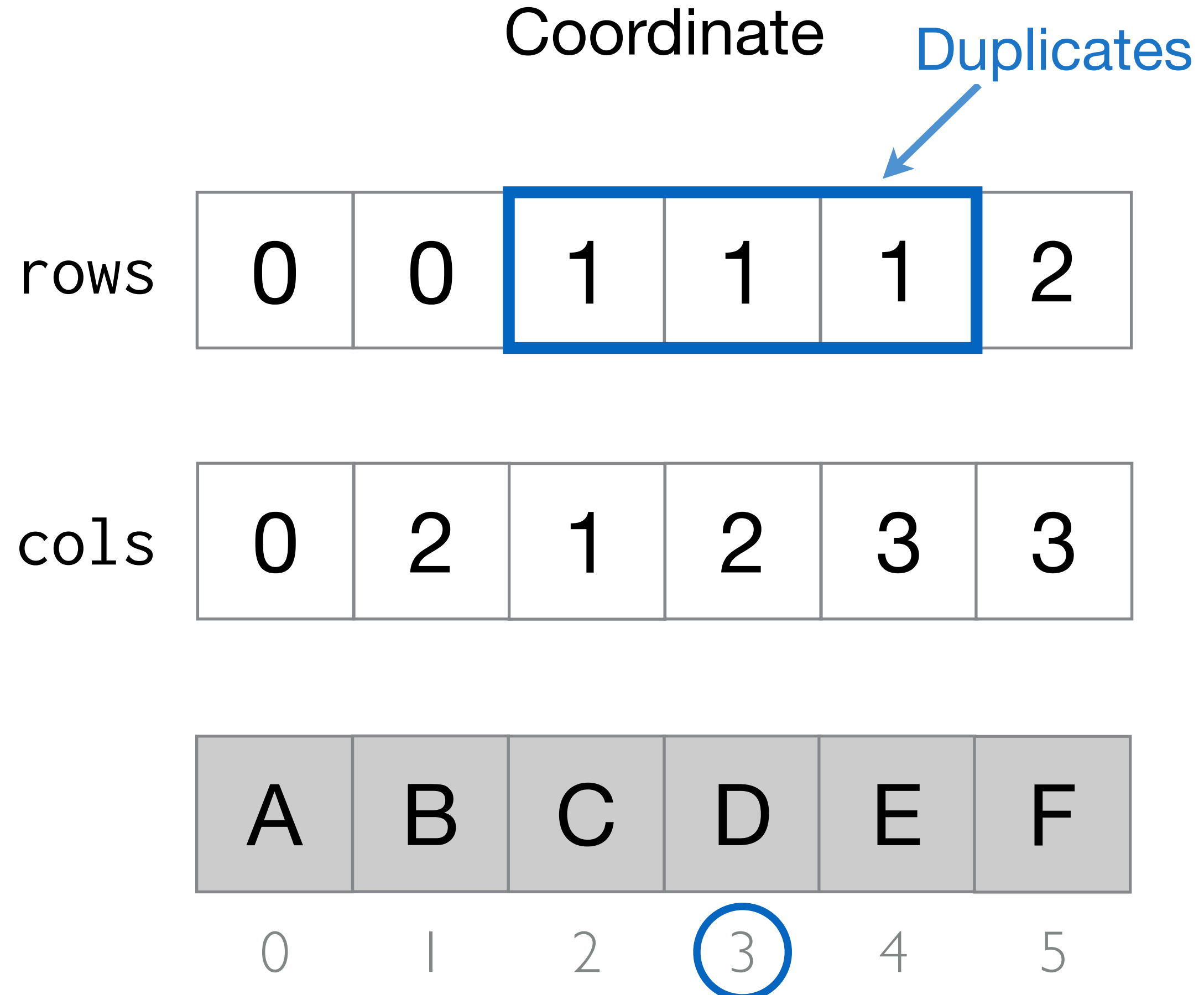


# Sparse tensors can be compressed by adding metadata

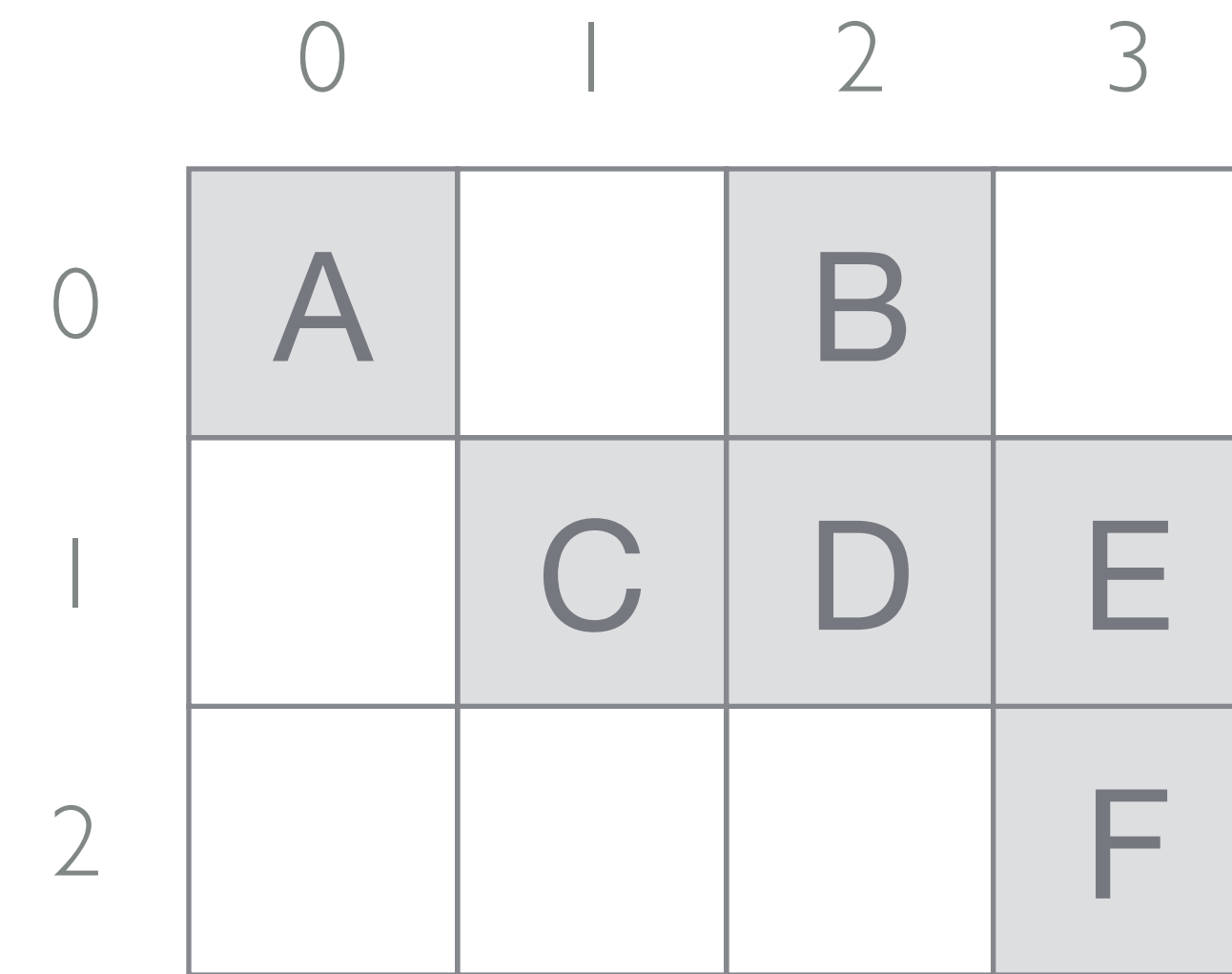
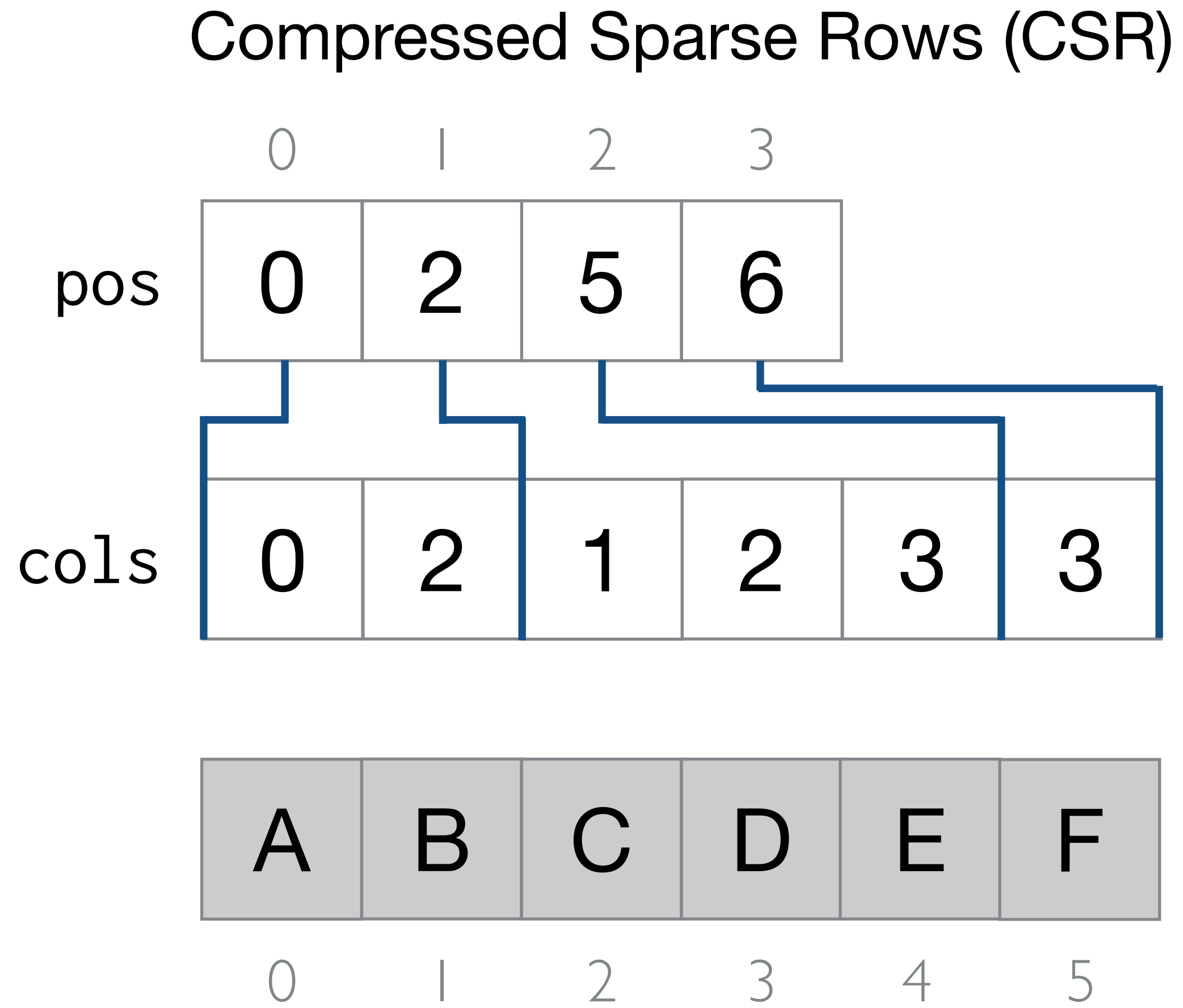




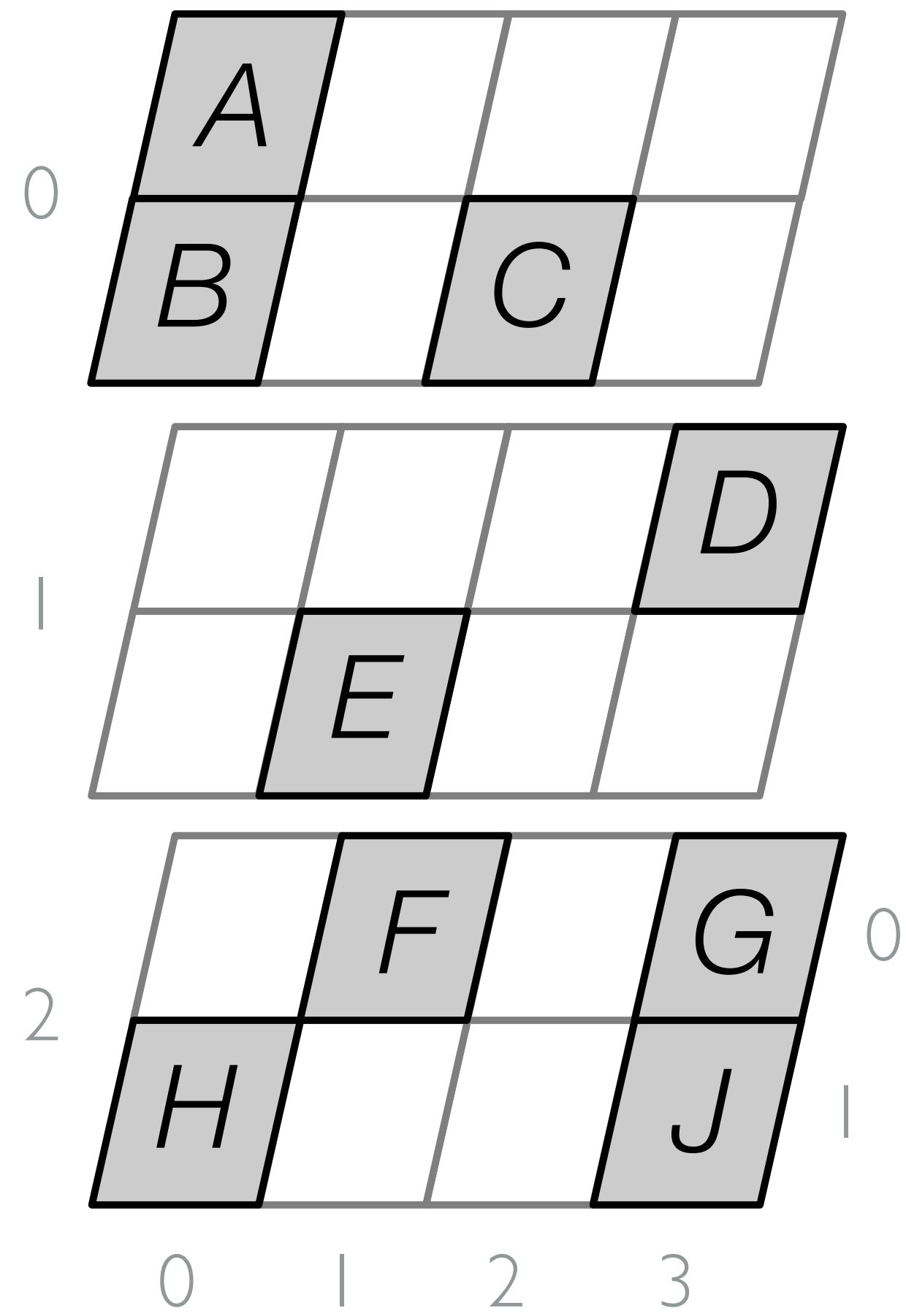
# Sparse tensors can be compressed by adding metadata



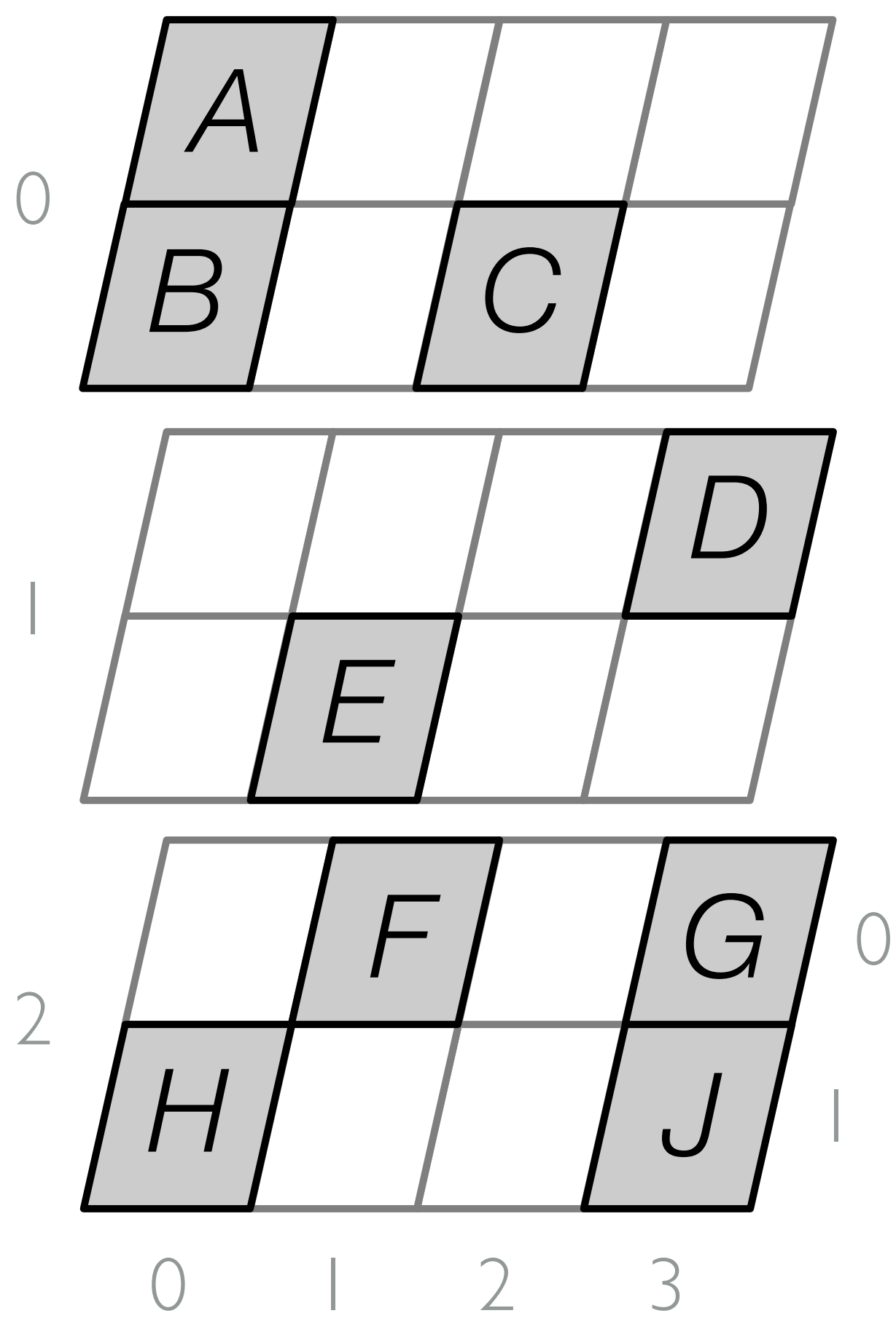
# Sparse tensors can be compressed by adding metadata



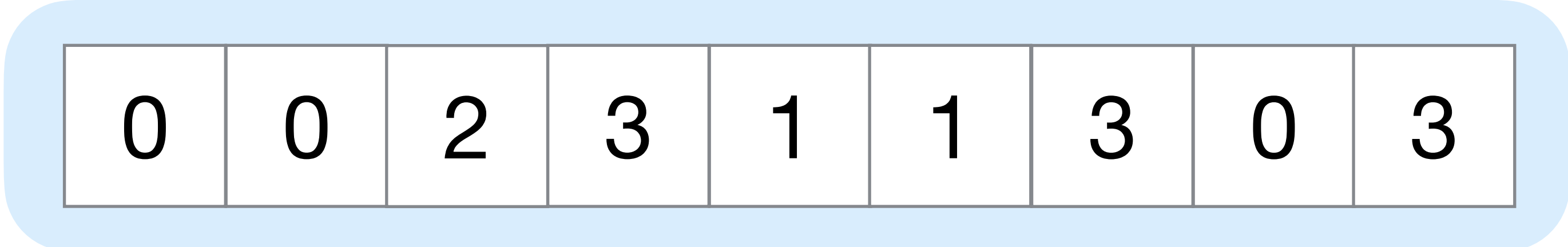
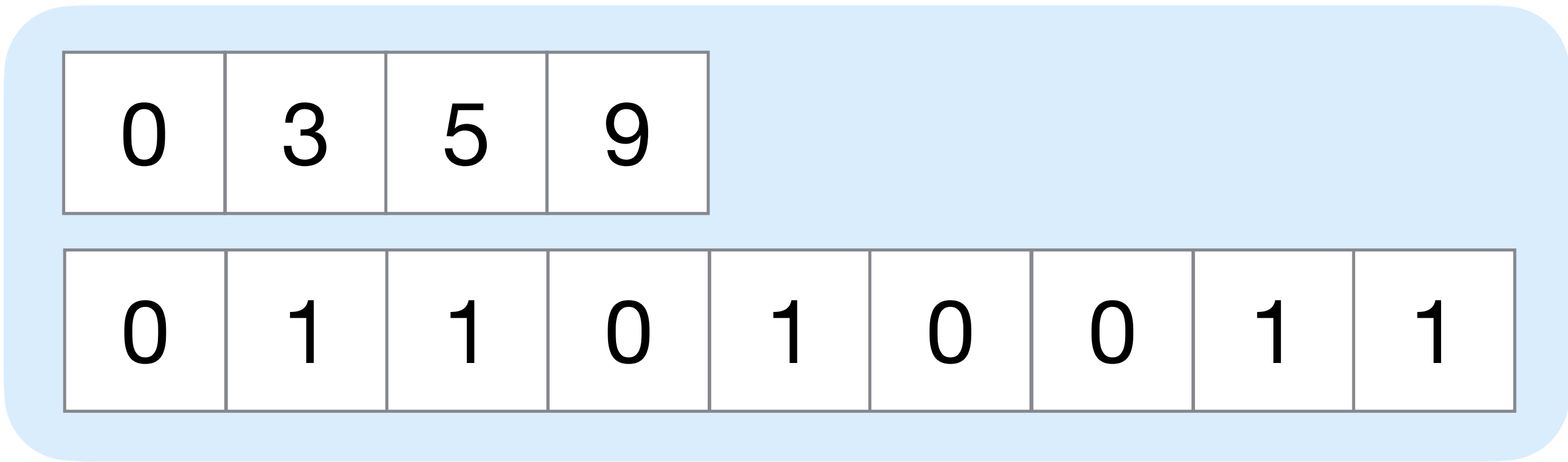
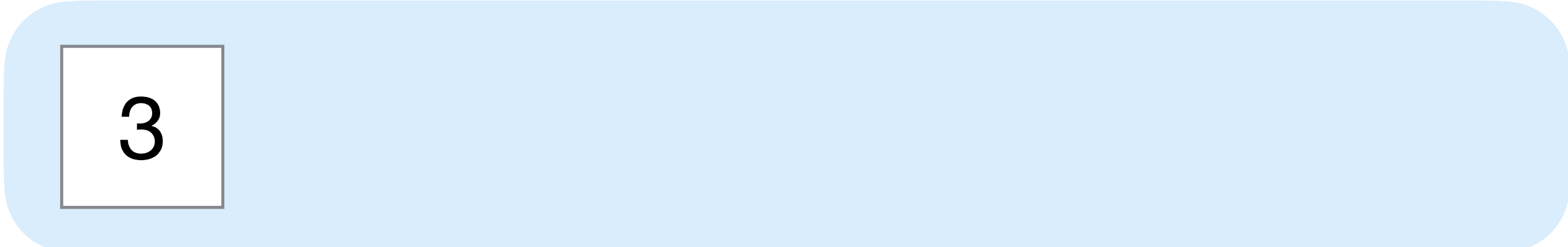
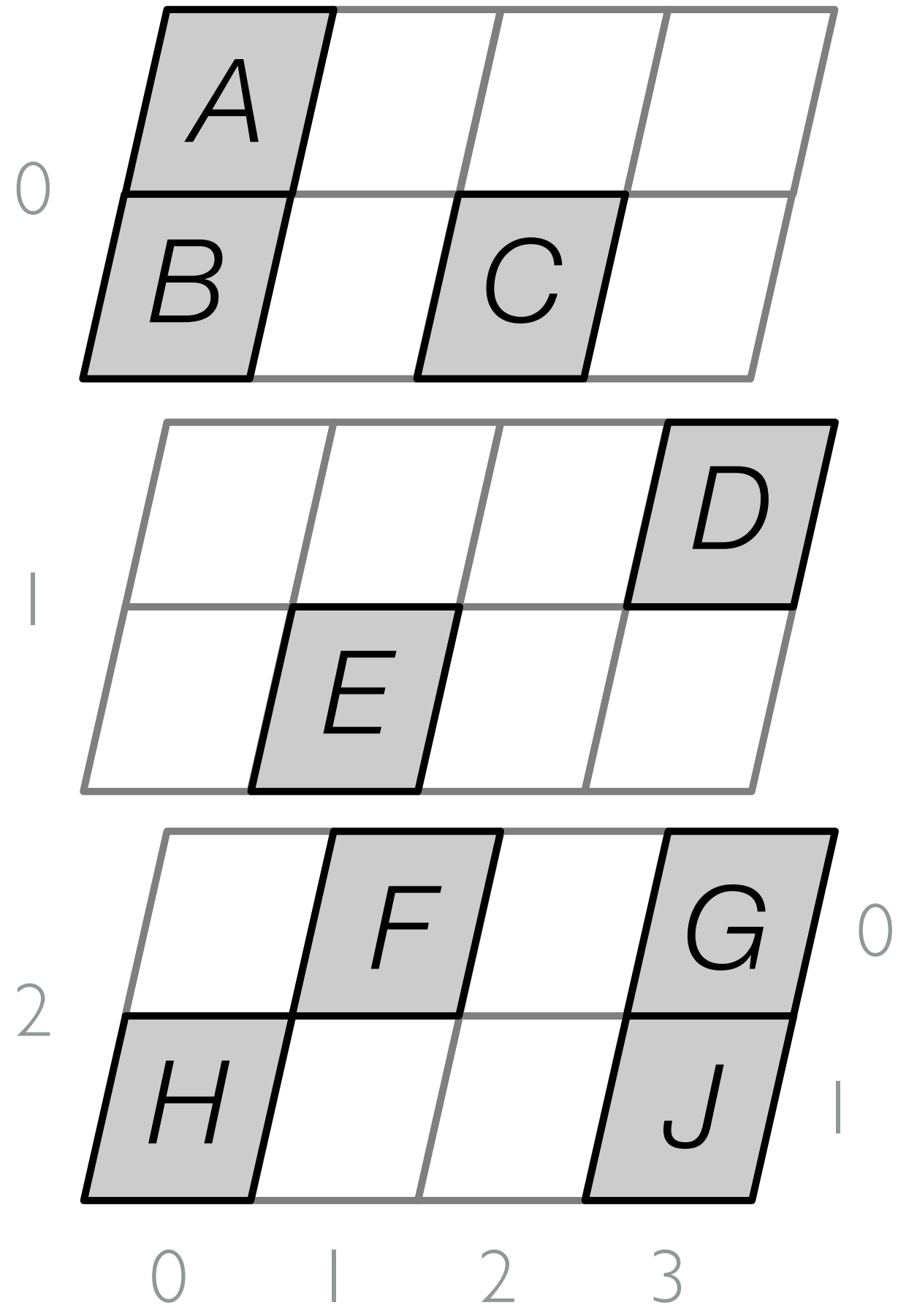
# TACO models tensor formats as a hierarchy of per-dimension formats



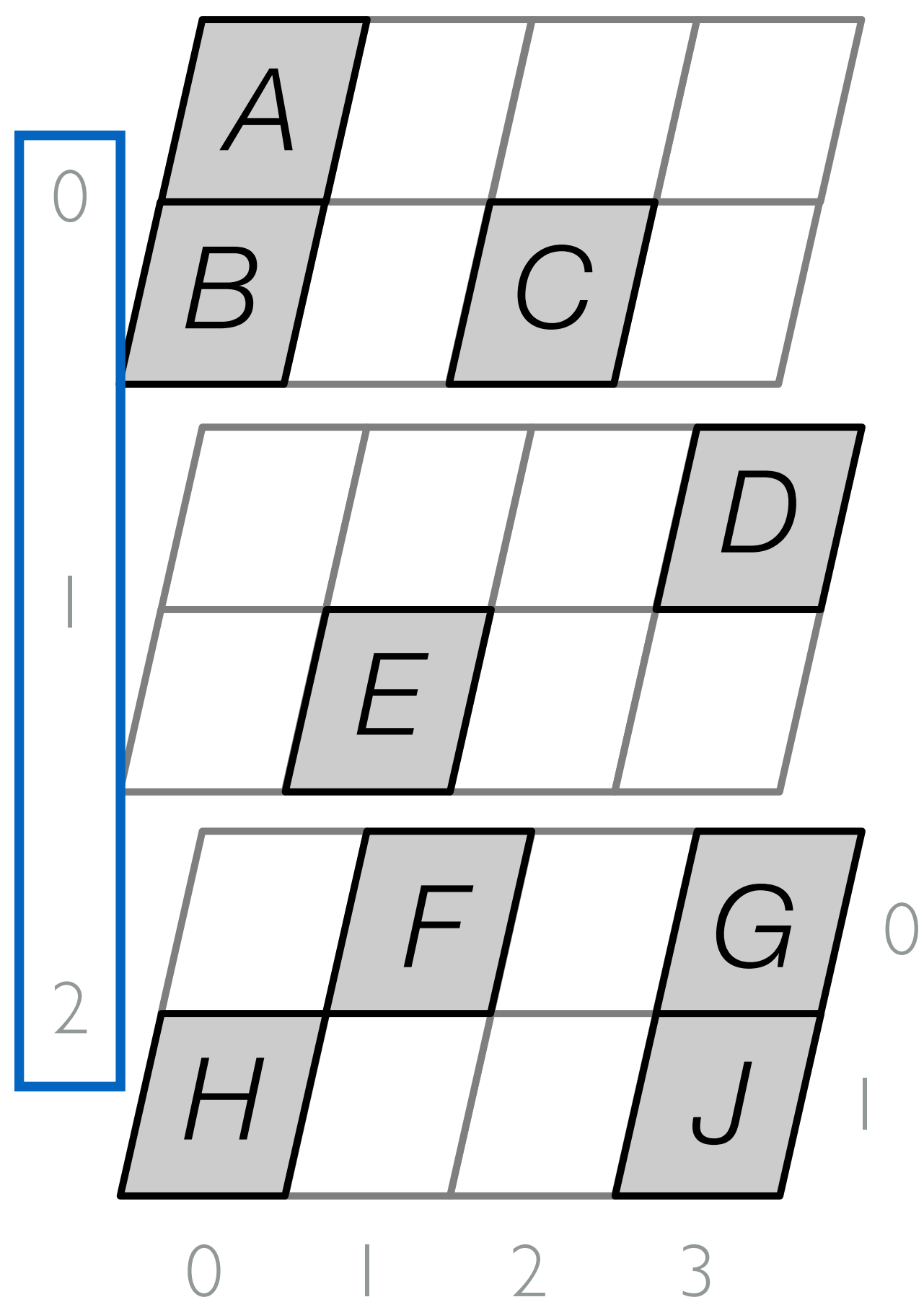
# TACO models tensor formats as a hierarchy of per-dimension formats



# TACO models tensor formats as a hierarchy of per-dimension formats



# TACO models tensor formats as a hierarchy of per-dimension formats



Slice Coordinates

3

0 3 5 9

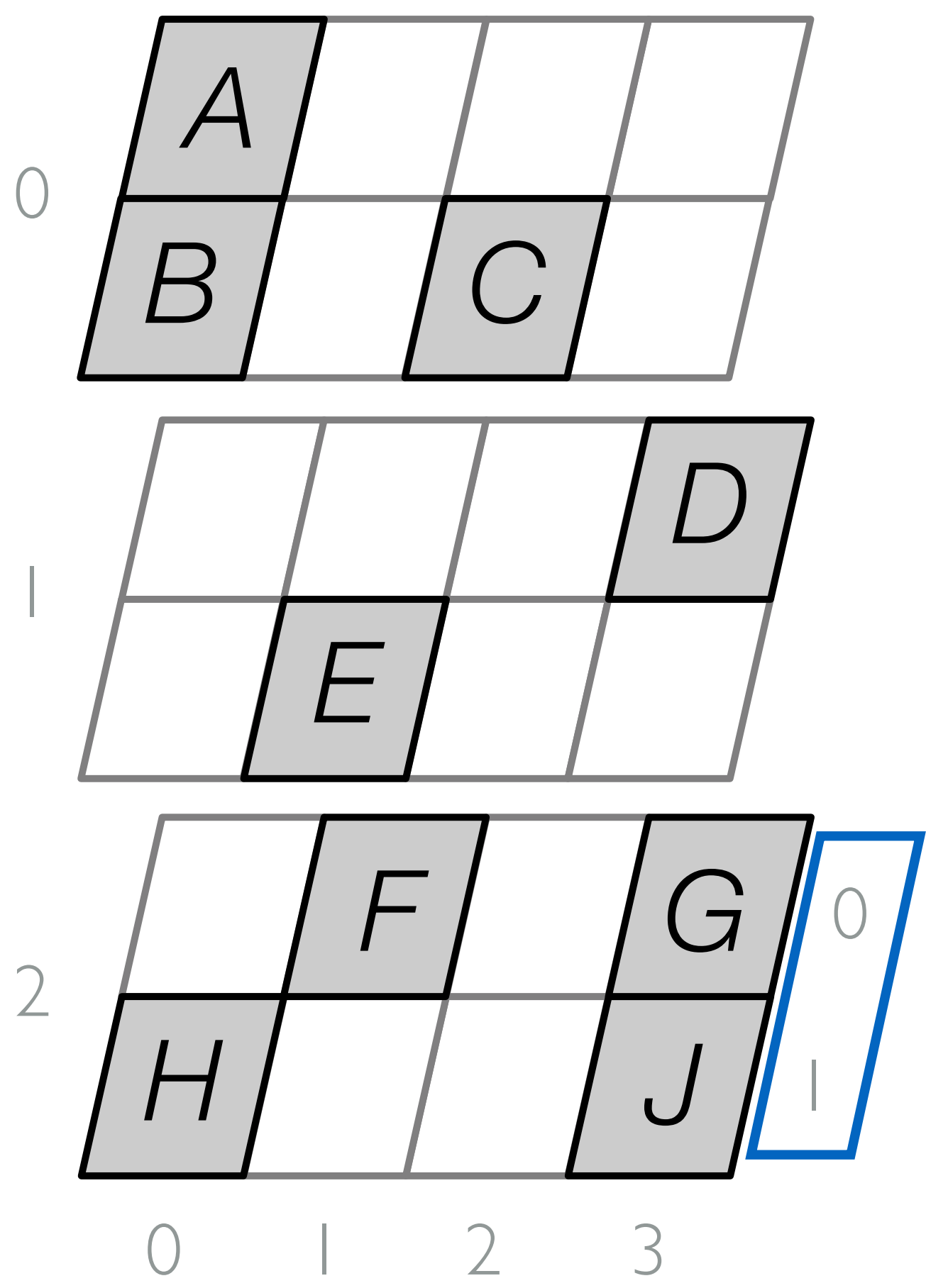
0 1 1 0 1 0 0 1 1

0 0 2 3 1 1 3 0 3

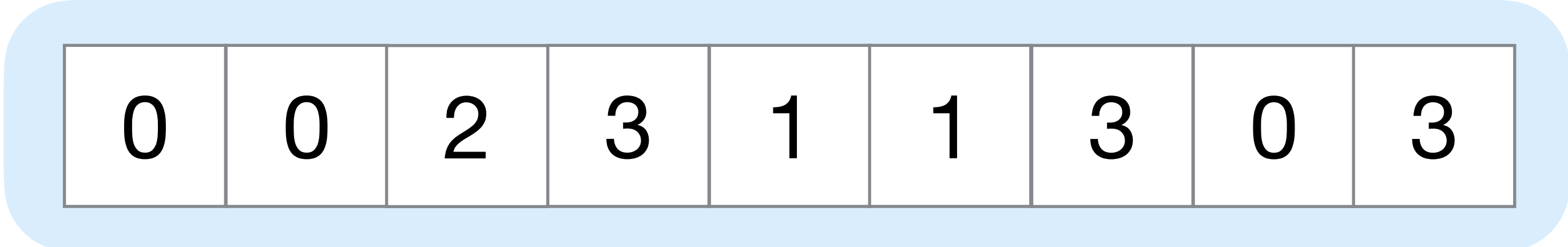
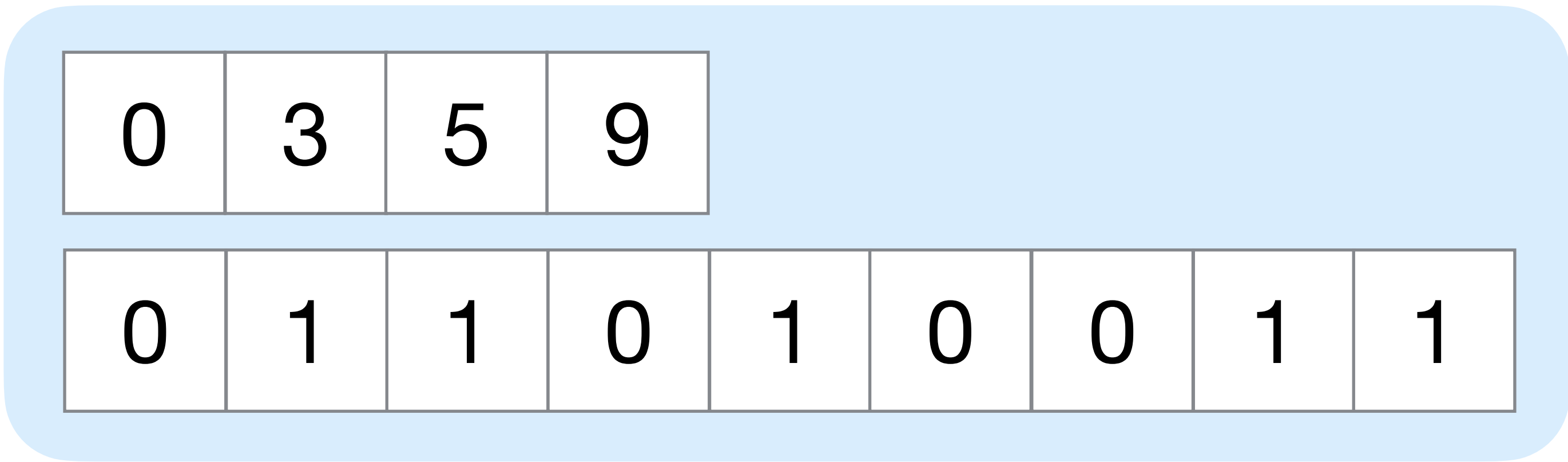
A B C D E F G H J

0 1 2 3 4 5 6 7 8

# TACO models tensor formats as a hierarchy of per-dimension formats

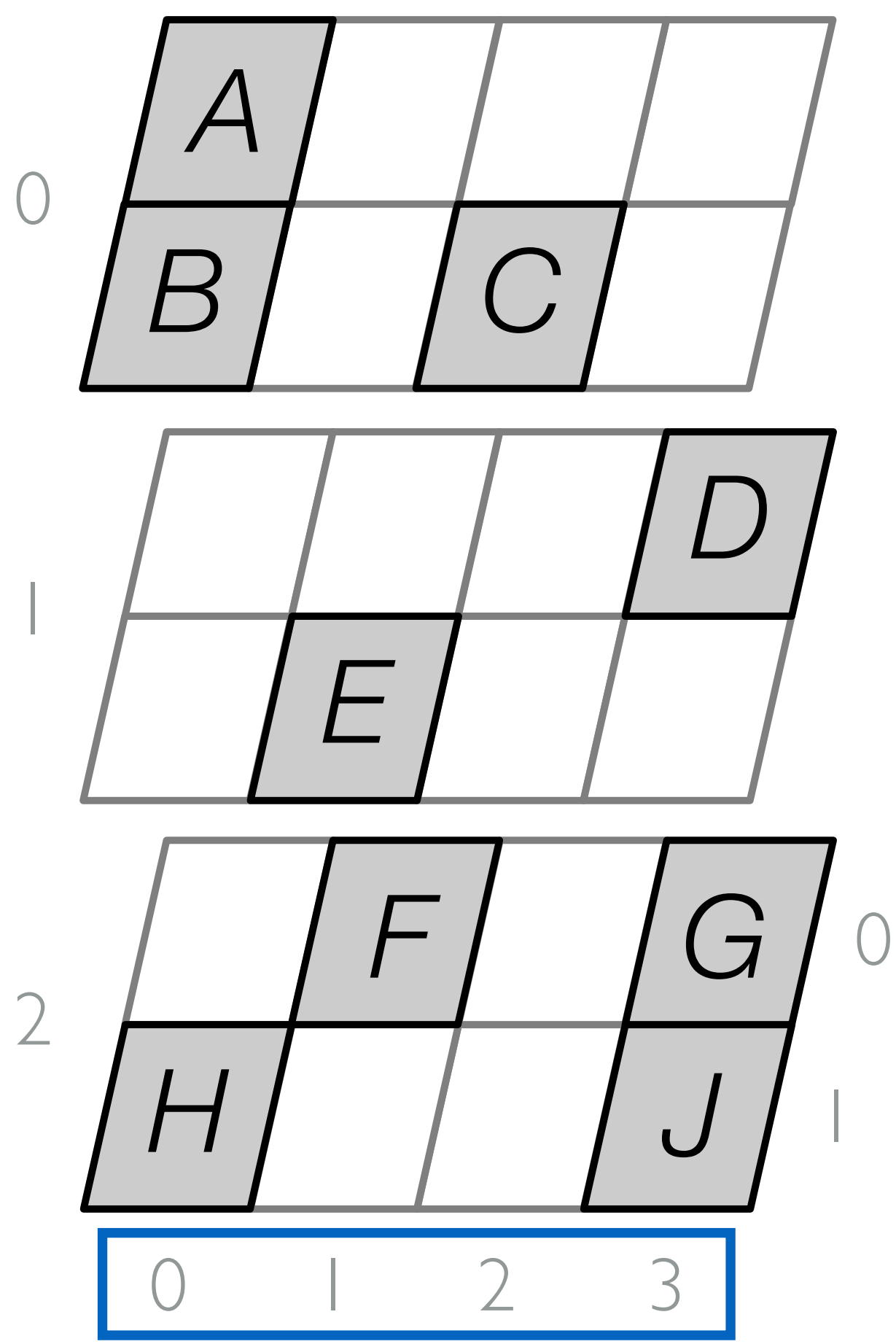


Row  
Coordinates

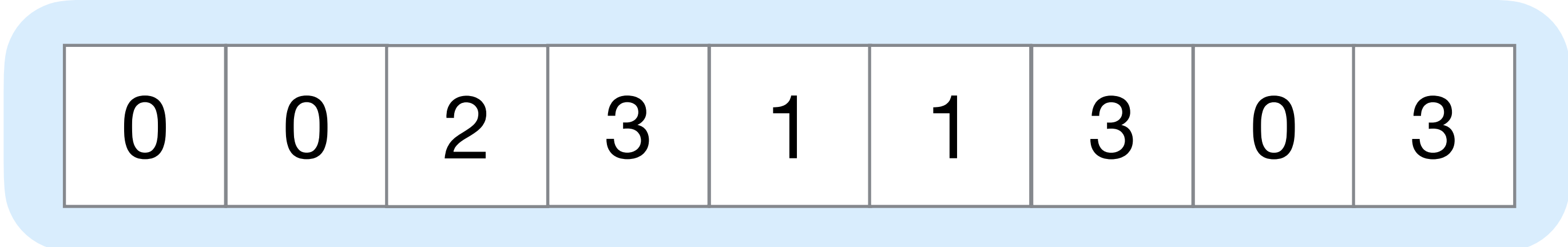
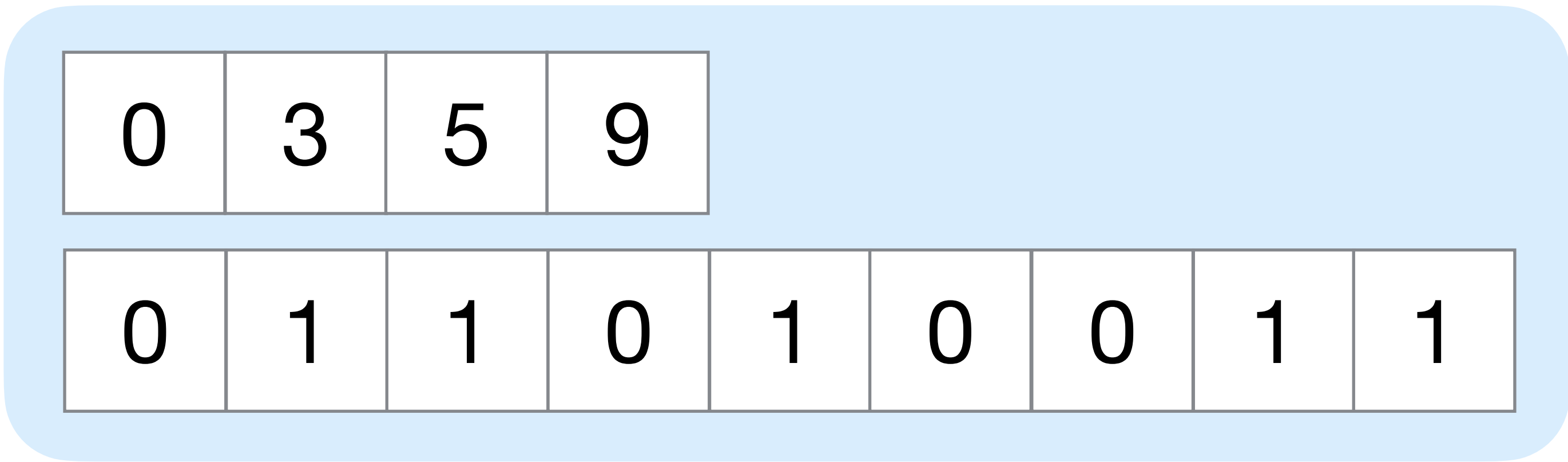
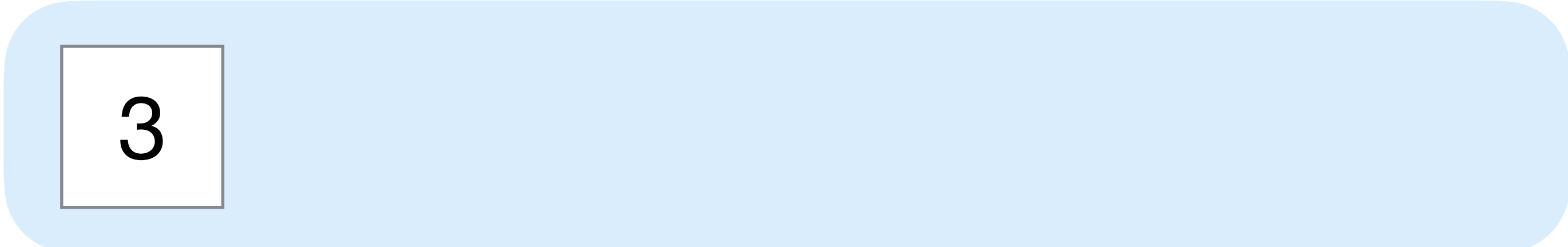




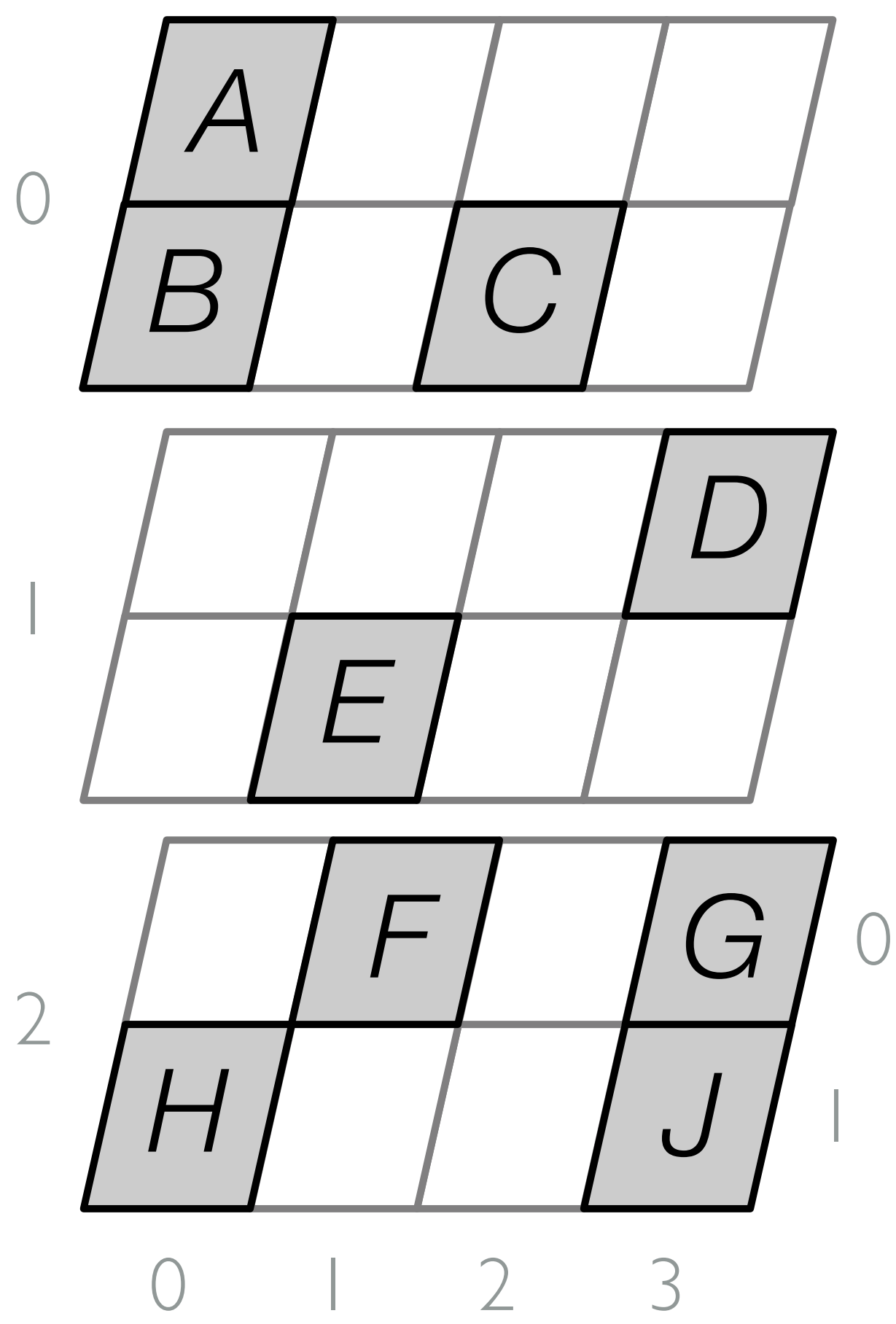
# TACO models tensor formats as a hierarchy of per-dimension formats



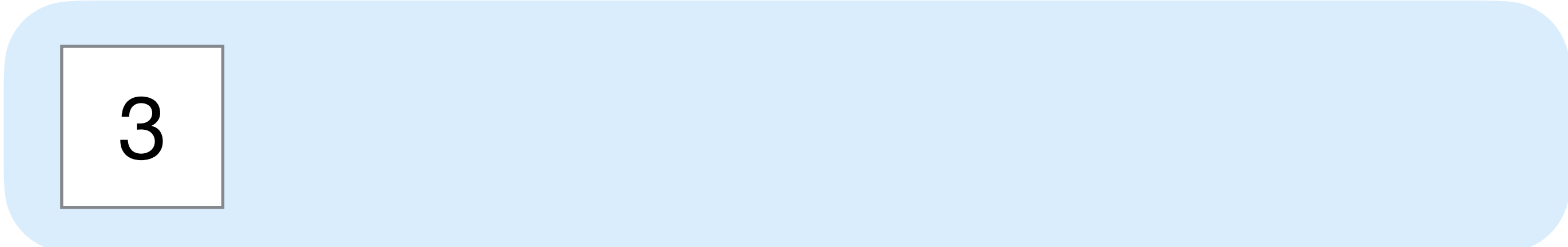
Column Coordinates



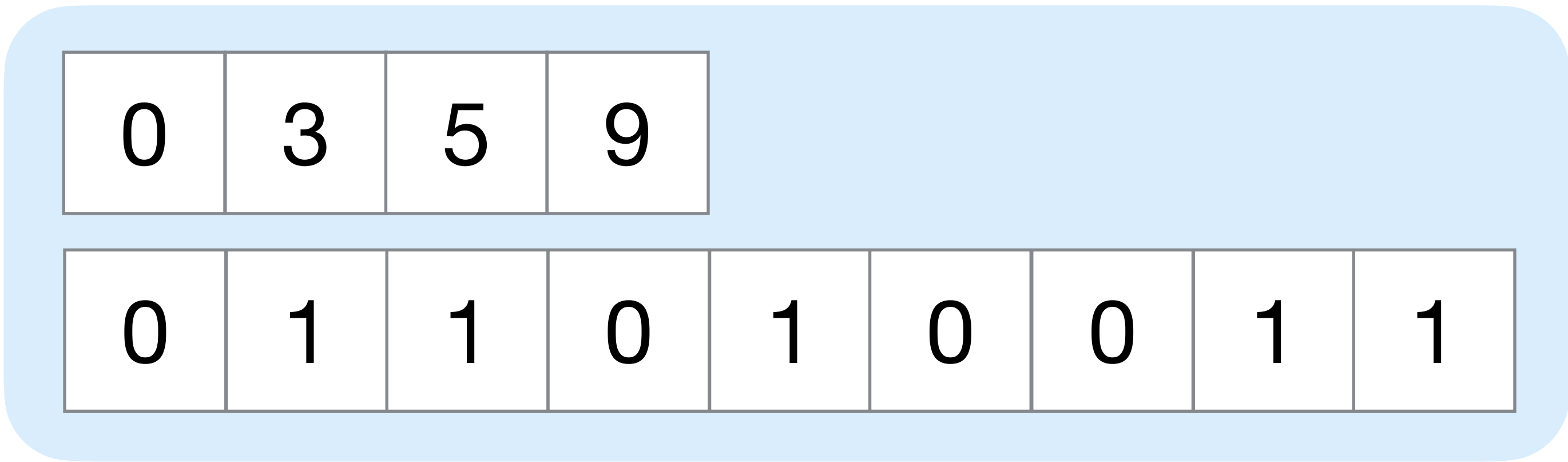
# TACO models tensor formats as a hierarchy of per-dimension formats



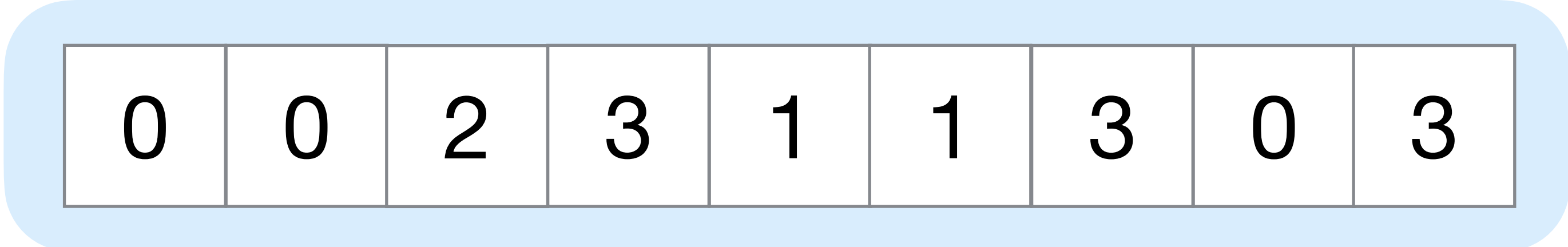
Dense



Compressed

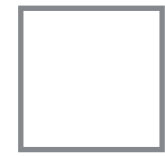


Singleton

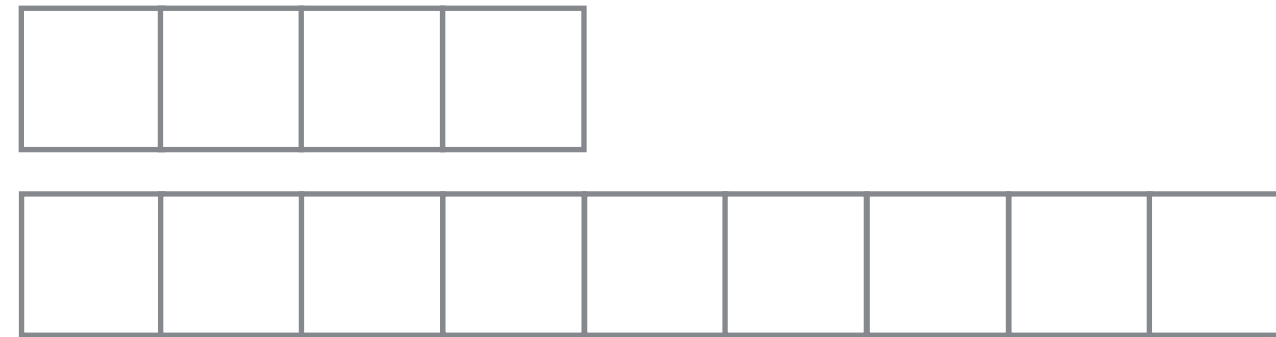


# Per-dimension formats can be composed in many ways

Dense



Compressed

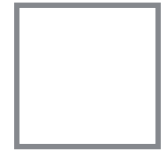


Singleton

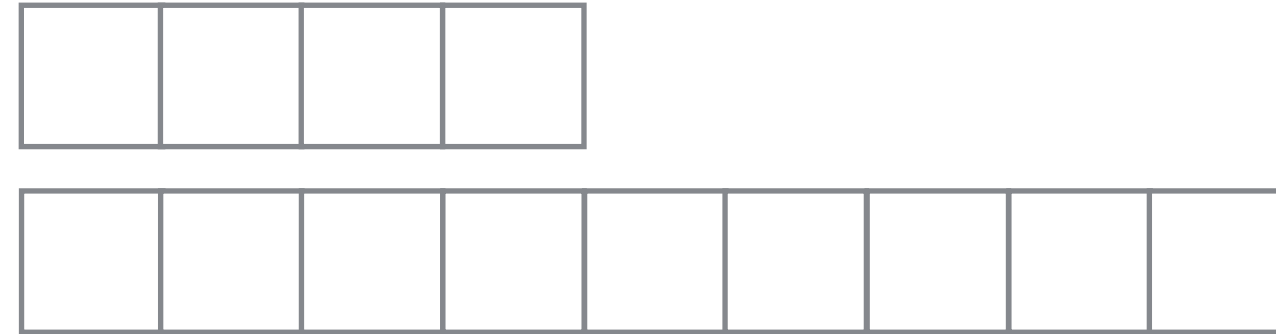


# Per-dimension formats can be composed in many ways

Dense



Compressed



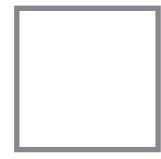
Singleton



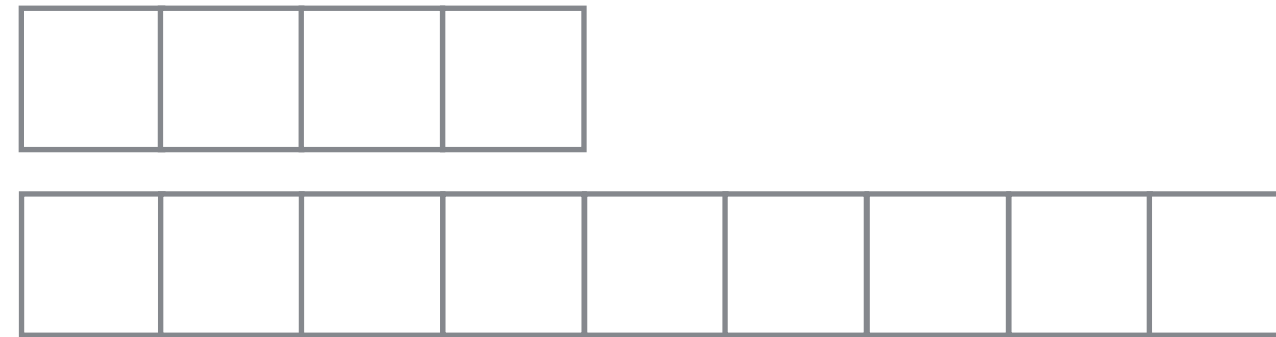
	0	1	2	3
0	A		B	
1		C	D	E
2				F

# Per-dimension formats can be composed in many ways

Dense



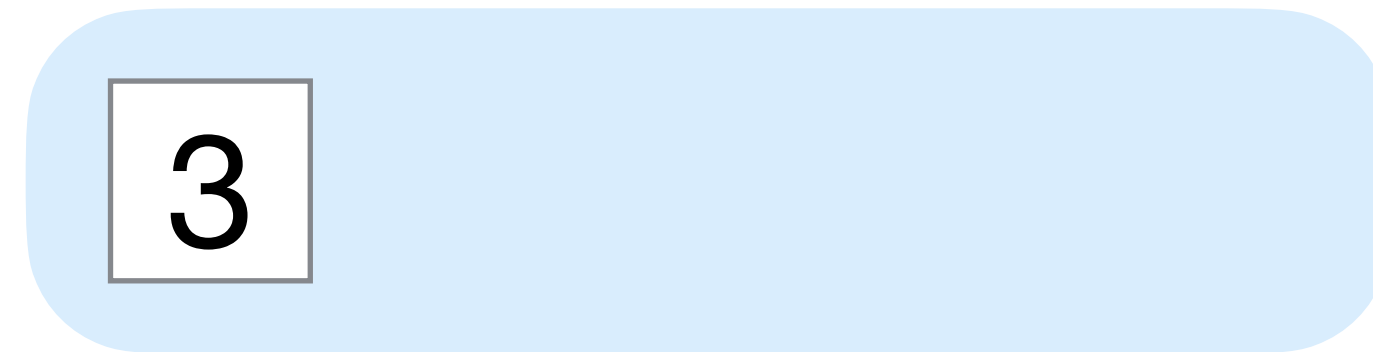
Compressed



Singleton



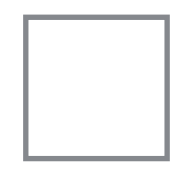
Dense



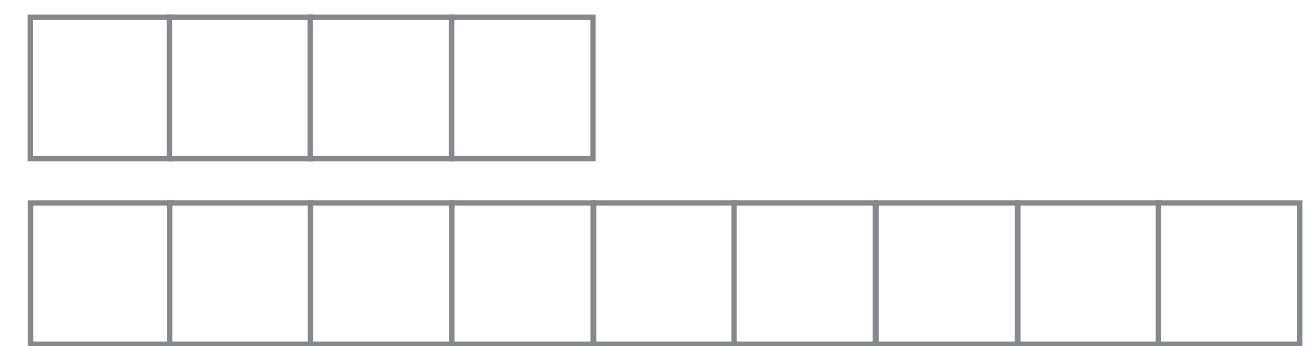
	0	1	2	3
0	A		B	
1		C	D	E
2				F

# Per-dimension formats can be composed in many ways

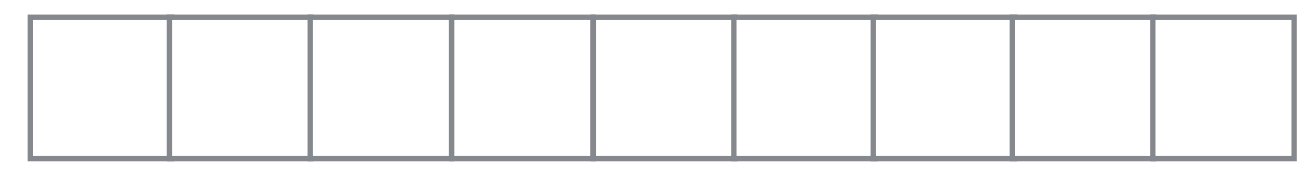
Dense



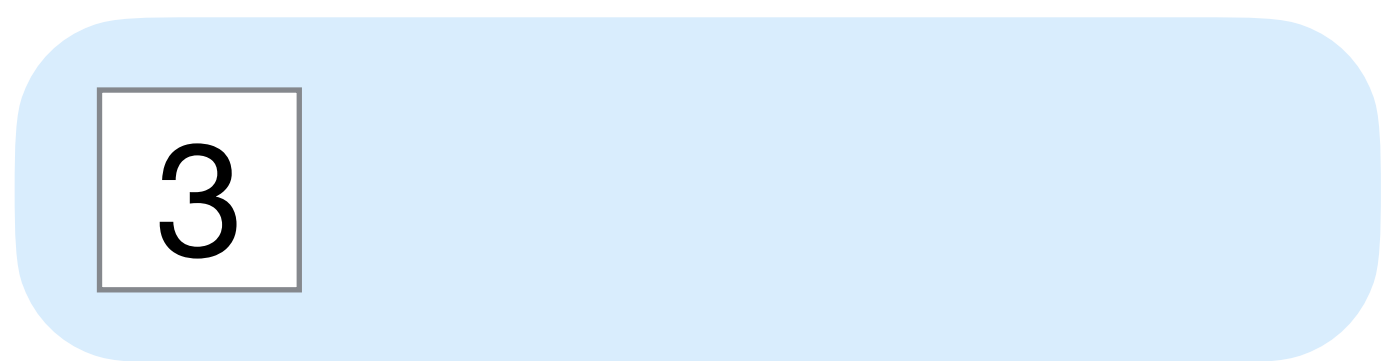
Compressed



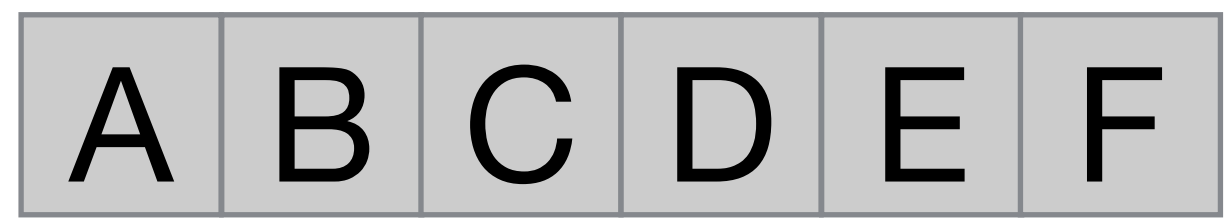
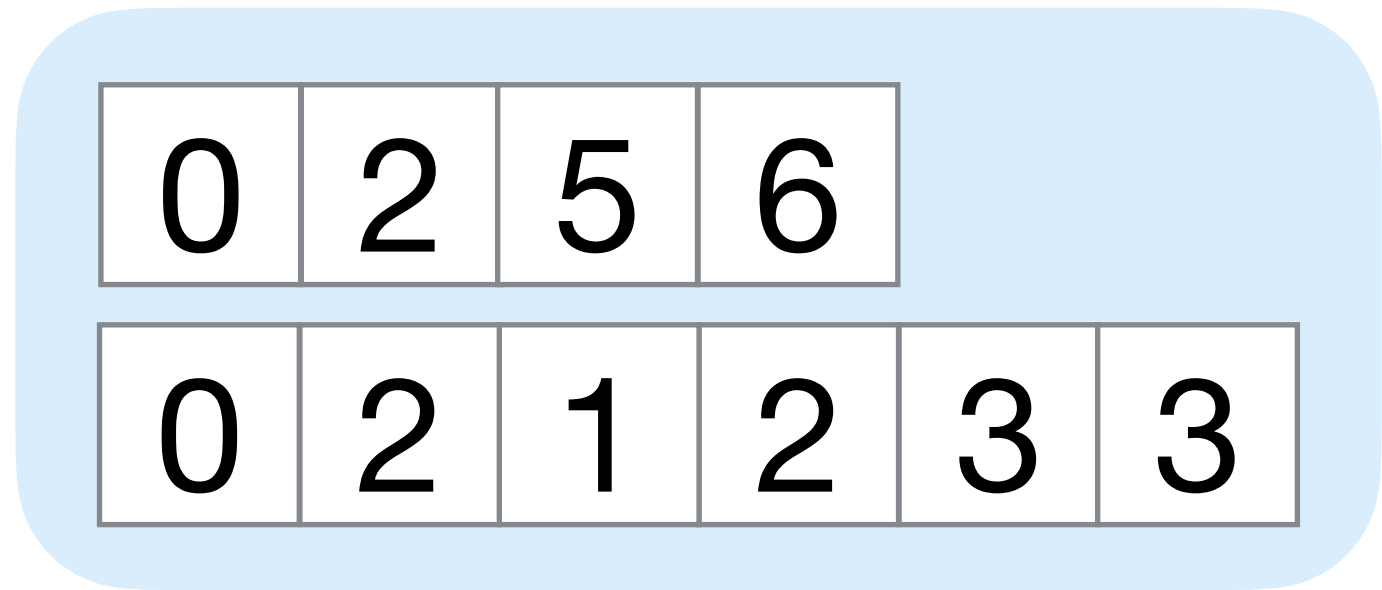
Singleton



Dense



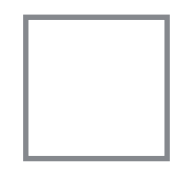
Compressed



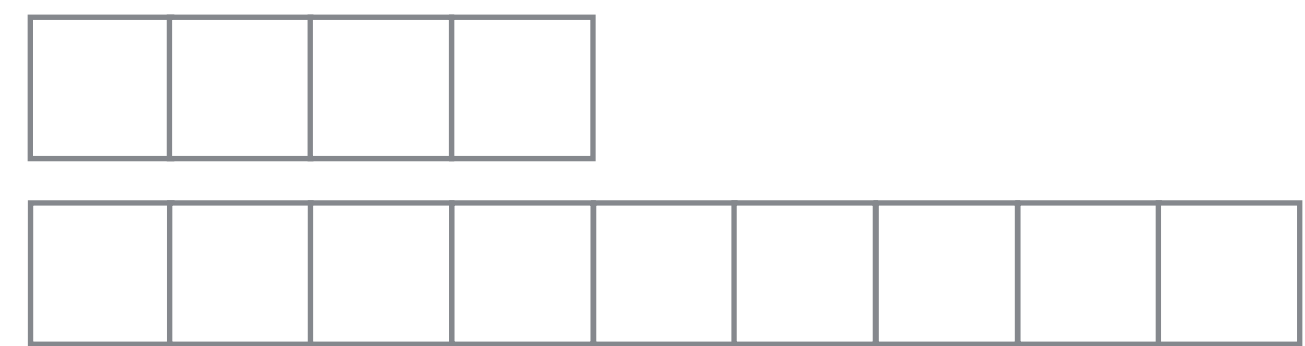
	0	1	2	3
0	A		B	
1		C	D	E
2				F

# Per-dimension formats can be composed in many ways

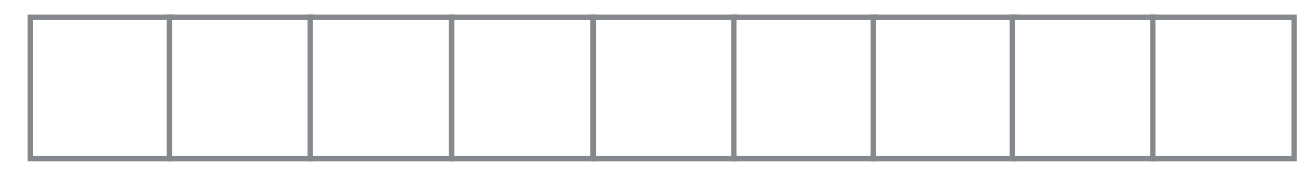
Dense



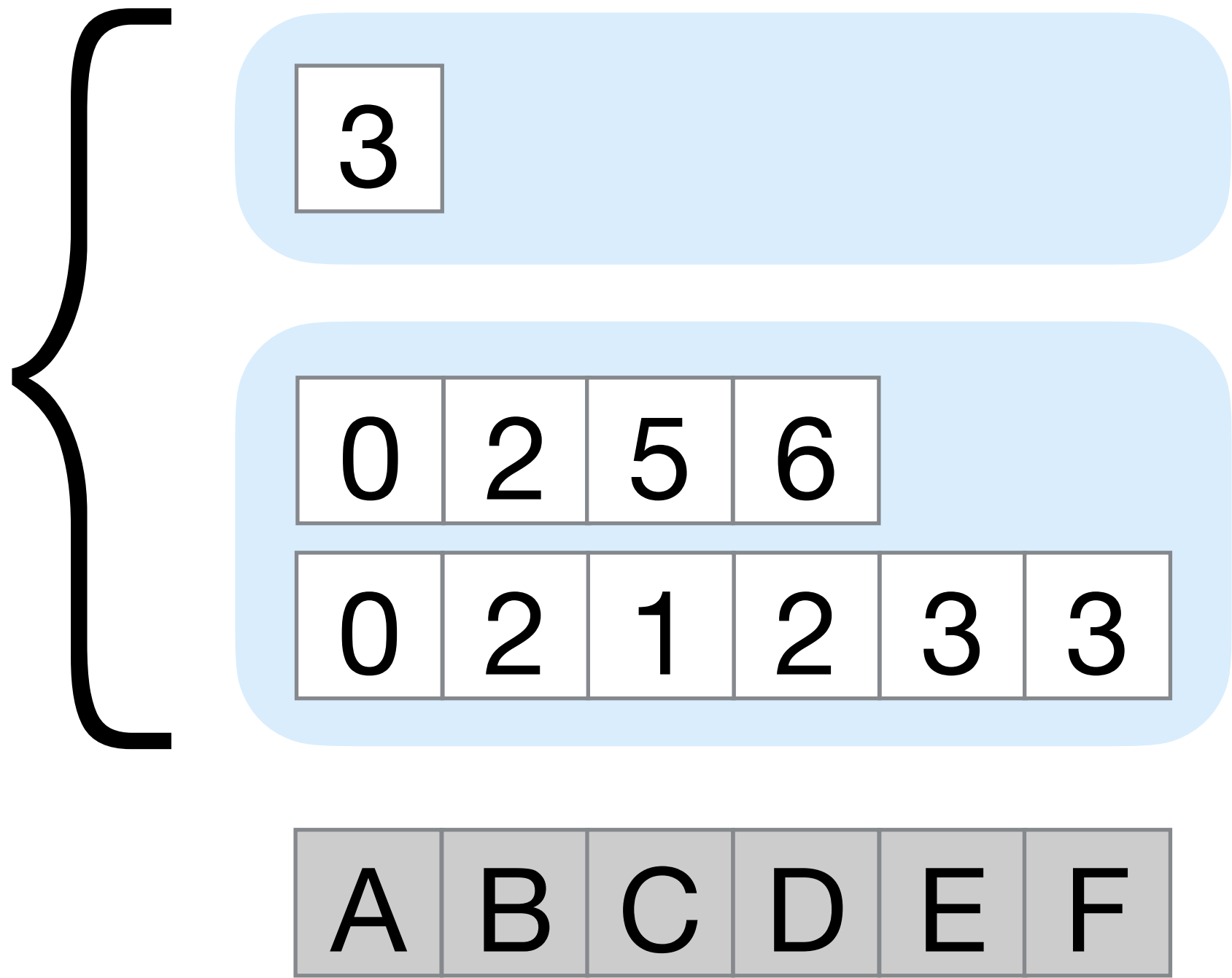
Compressed



Singleton



CSR

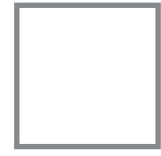


	0	1	2	3
0	A		B	
1		C	D	E
2				F

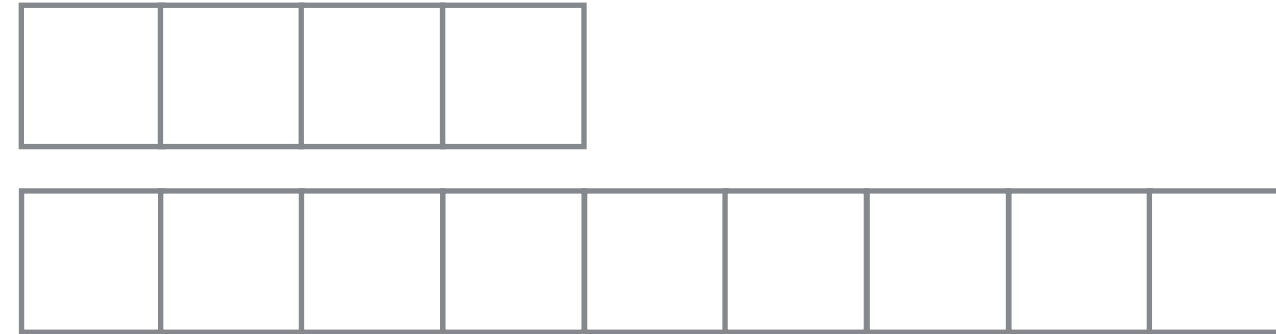


# Per-dimension formats can be composed in many ways

Dense



Compressed



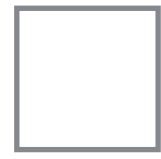
Singleton



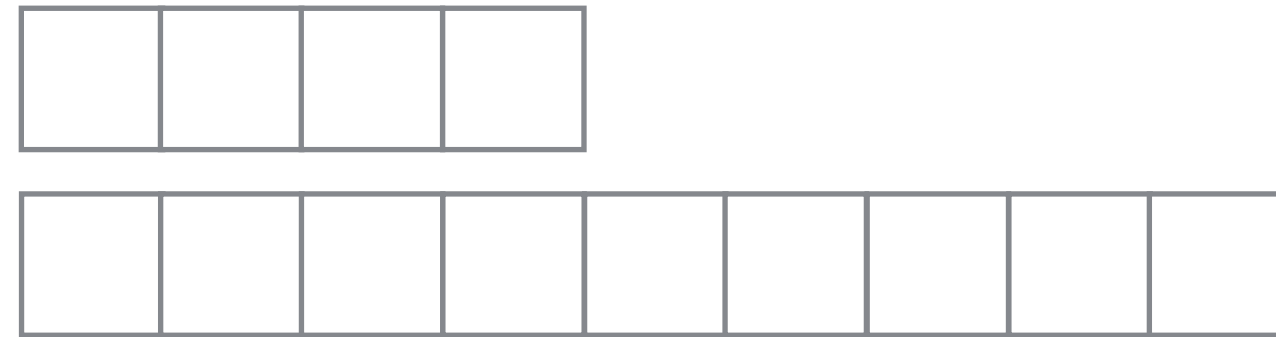
	0	1	2	3
0	A		B	
1		C	D	E
2				F

# Per-dimension formats can be composed in many ways

Dense



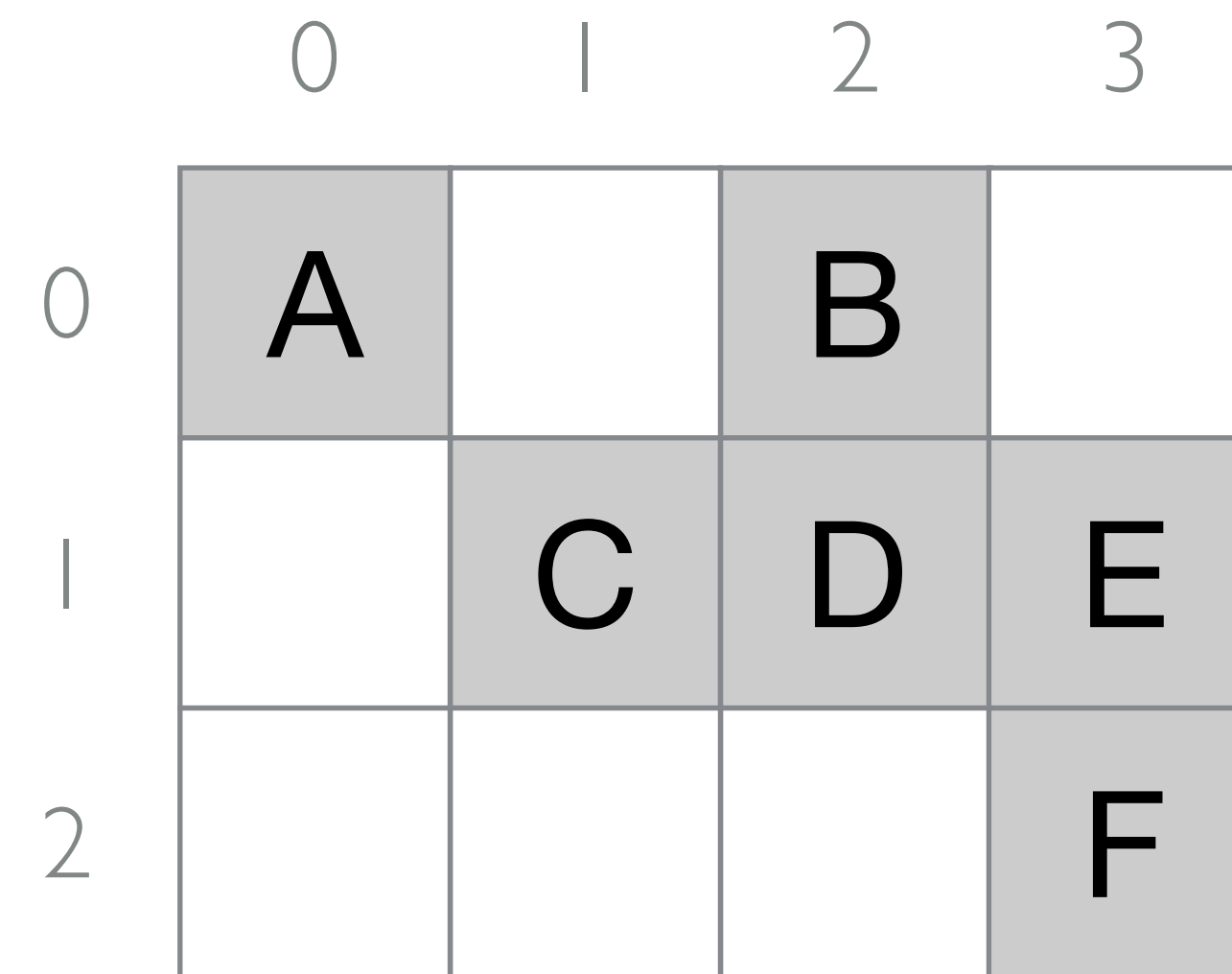
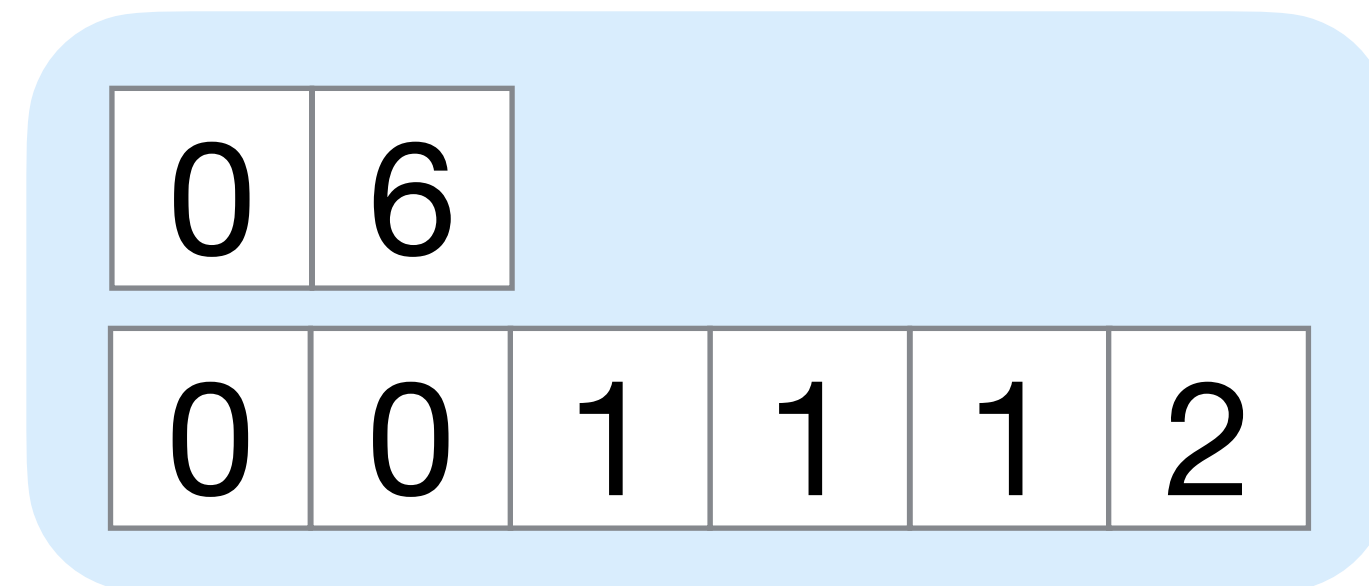
Compressed



Singleton

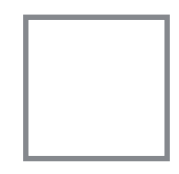


Compressed

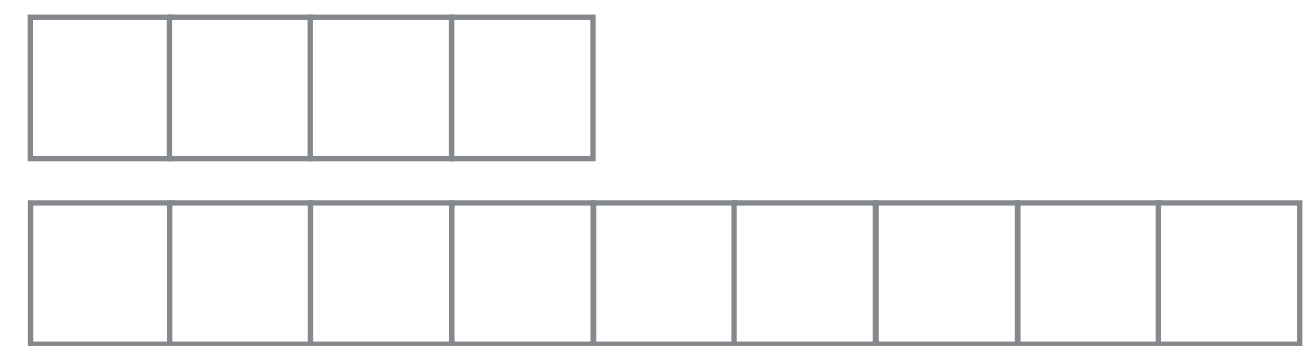


# Per-dimension formats can be composed in many ways

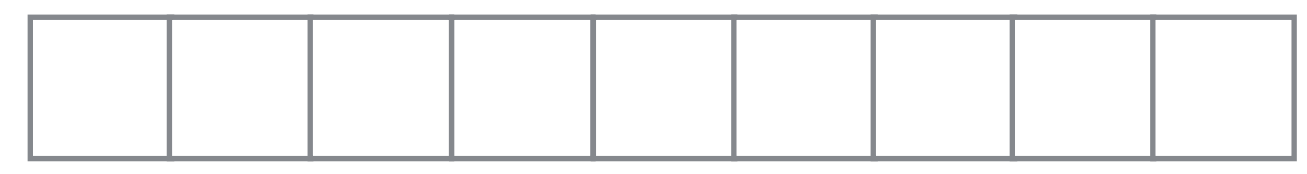
Dense



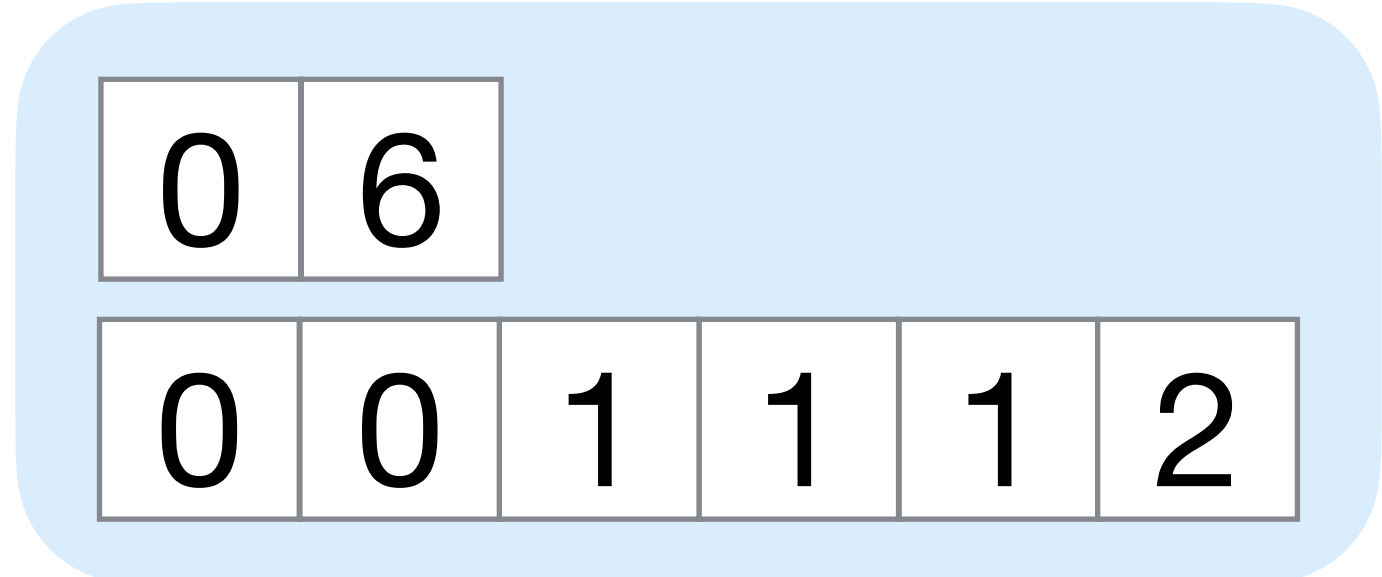
Compressed



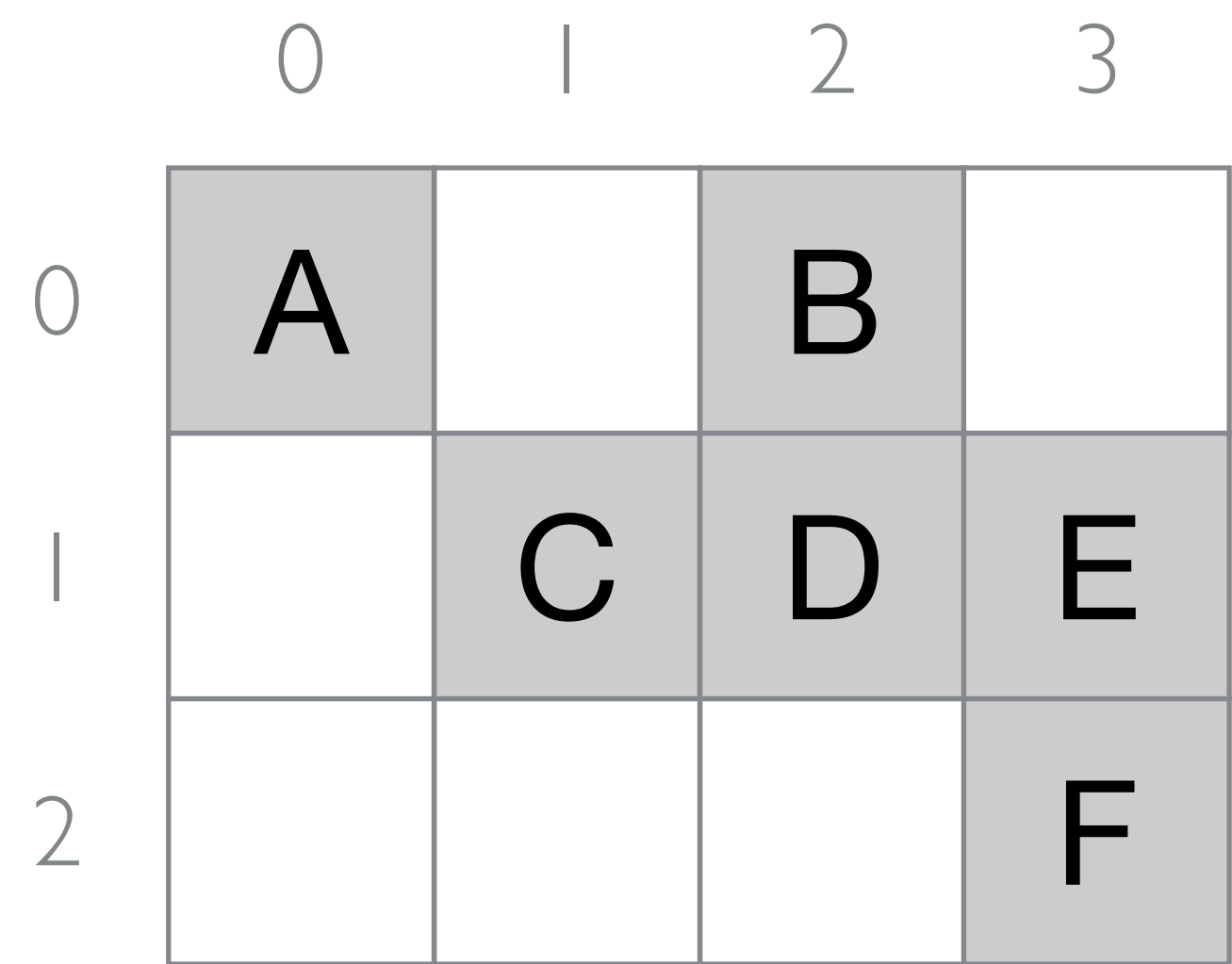
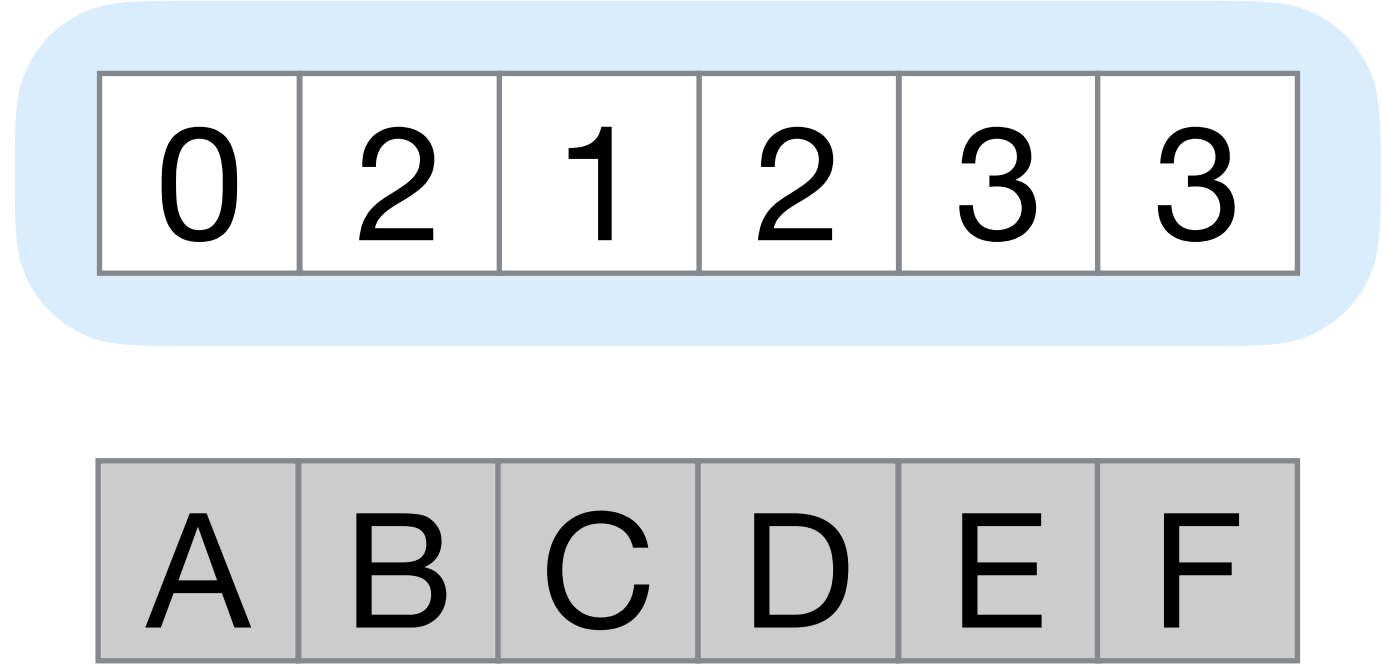
Singleton



Compressed

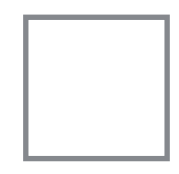


Singleton

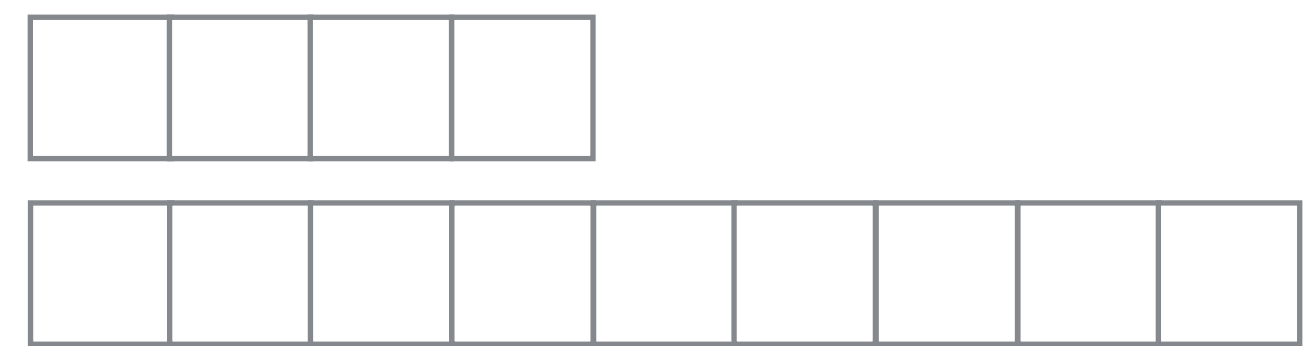


# Per-dimension formats can be composed in many ways

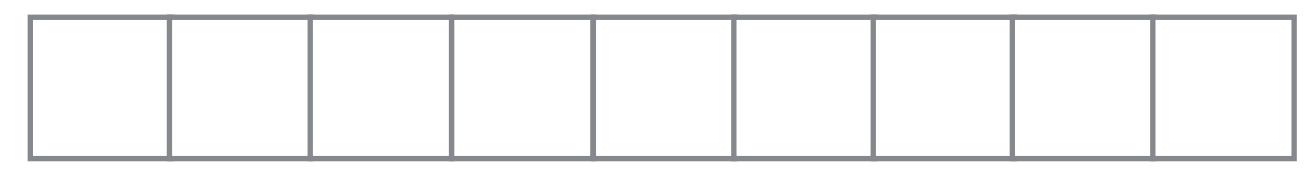
Dense



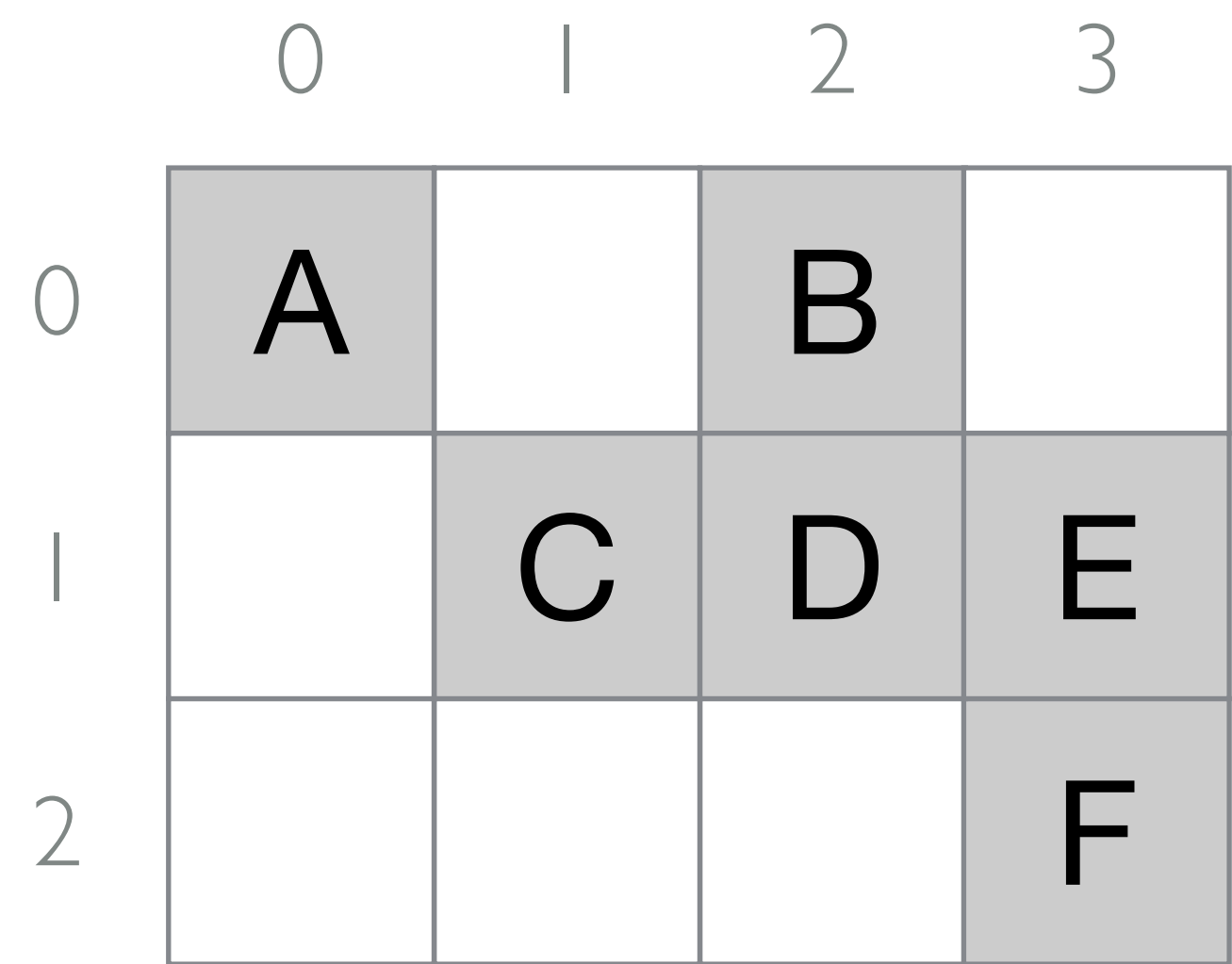
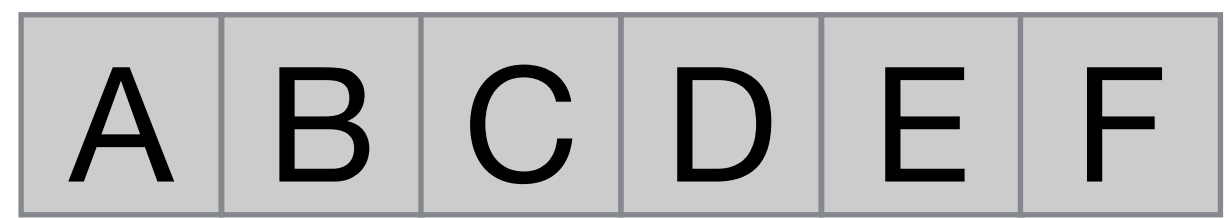
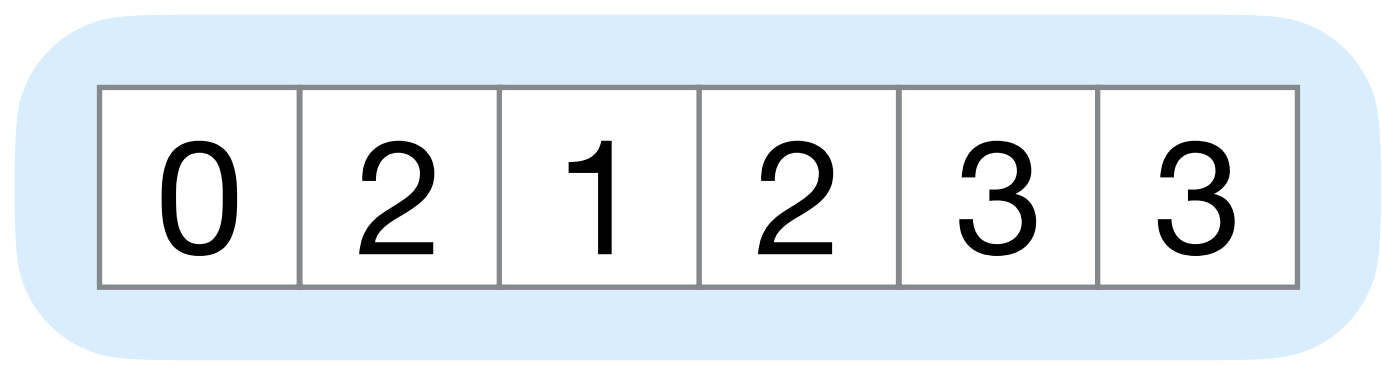
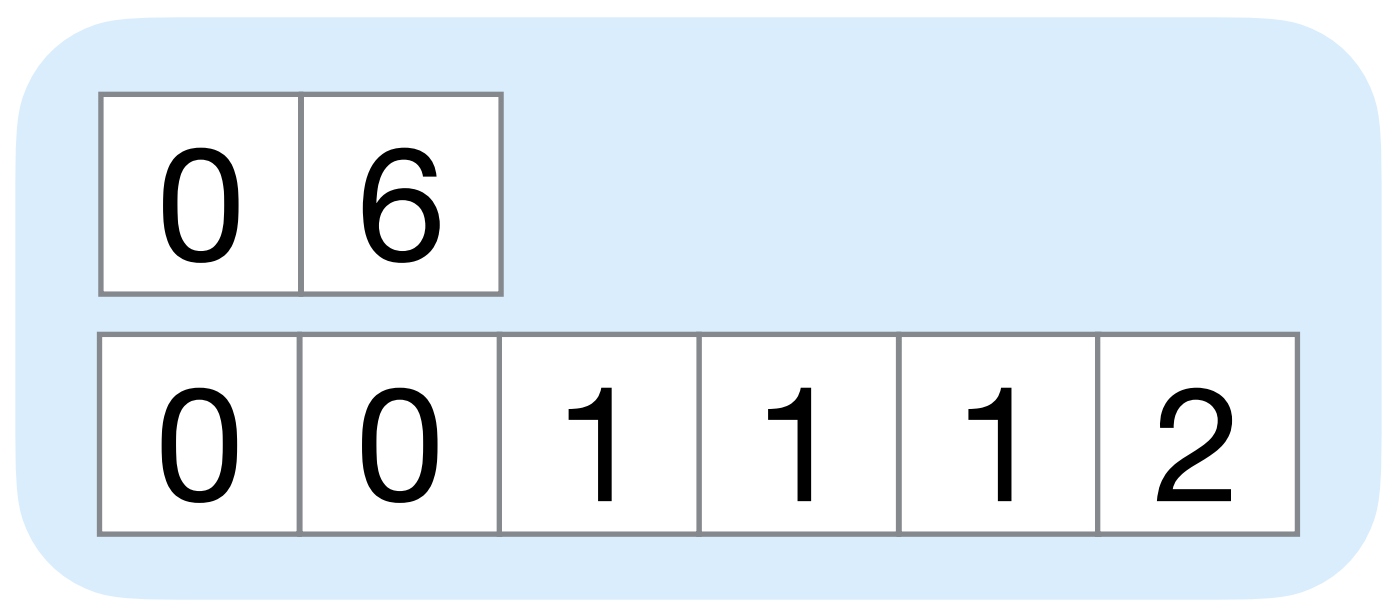
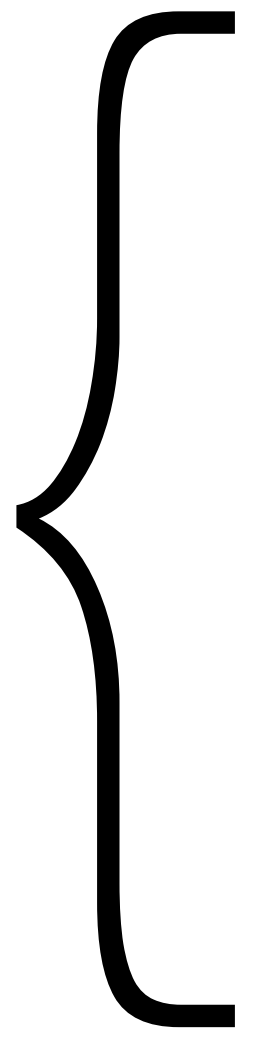
Compressed



Singleton



Coordinates





# Level formats can be composed in many ways

Dense

Compressed

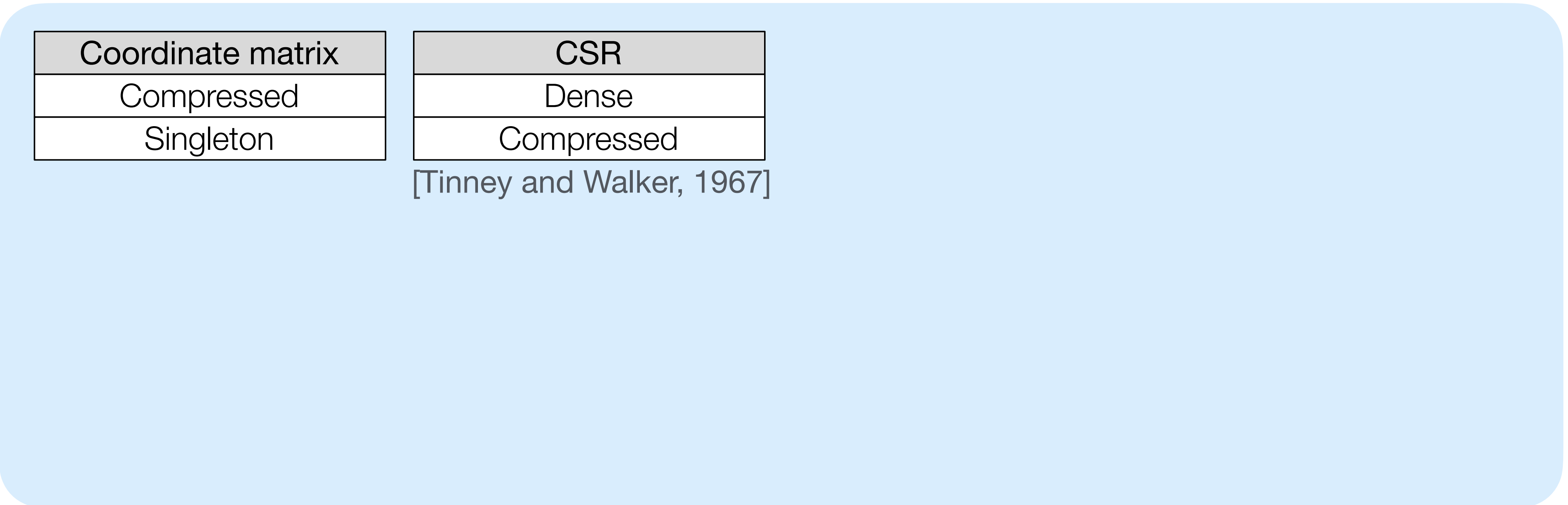
Singleton

# Level formats can be composed in many ways

Level  
formats

Dense    Compressed    Singleton

Tensor  
formats



# Level formats can be composed in many ways

Level formats: Dense, Compressed, Singleton

Tensor formats

Coordinate matrix
Compressed
Singleton

CSR
Dense
Compressed

Dense array tensor
Dense
Dense
Dense

Coordinate tensor
Compressed
Singleton
Singleton

[Tinney and Walker, 1967]

Mode-generic tensor
Compressed
Singleton
Dense
Dense

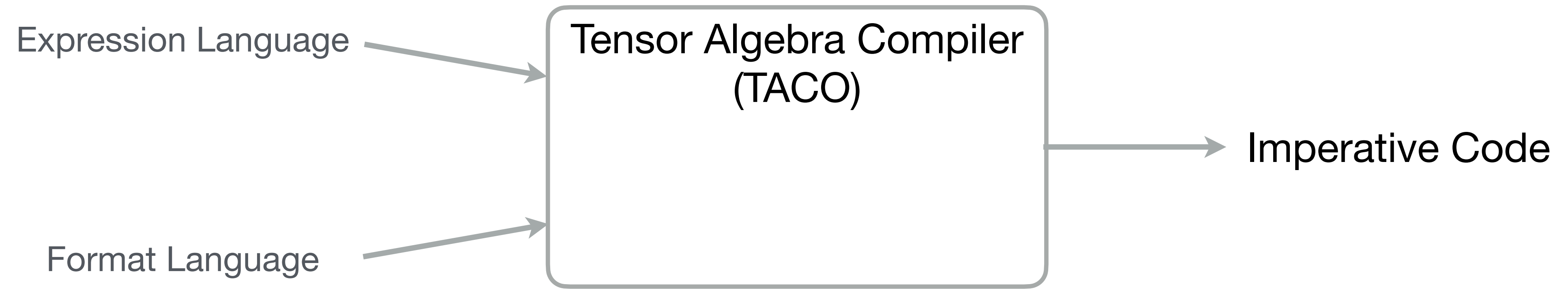
[Baskaran et al. 2012]

# Level formats can be composed in many ways

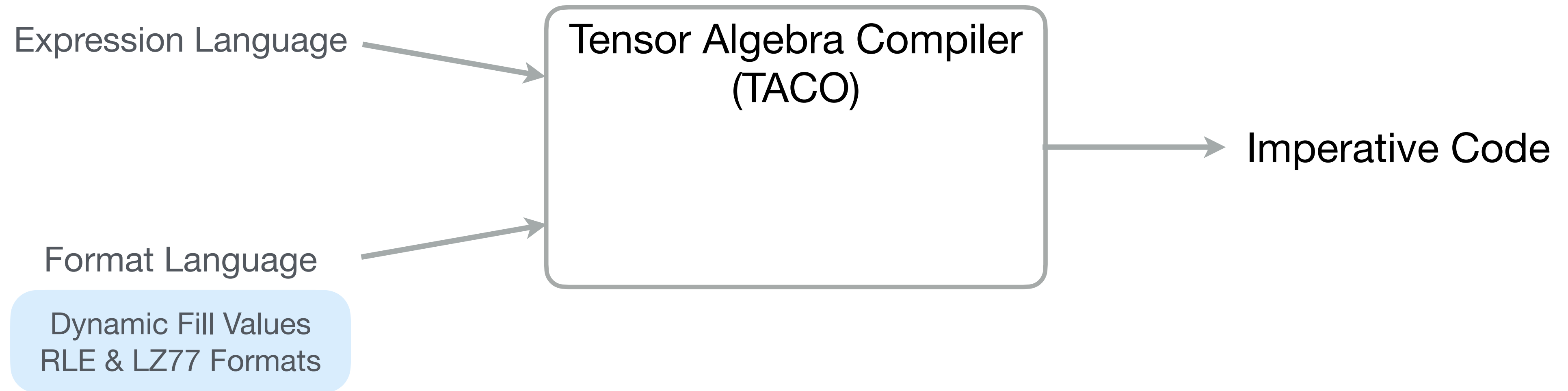
Level formats  
 Dense    Compressed    Singleton

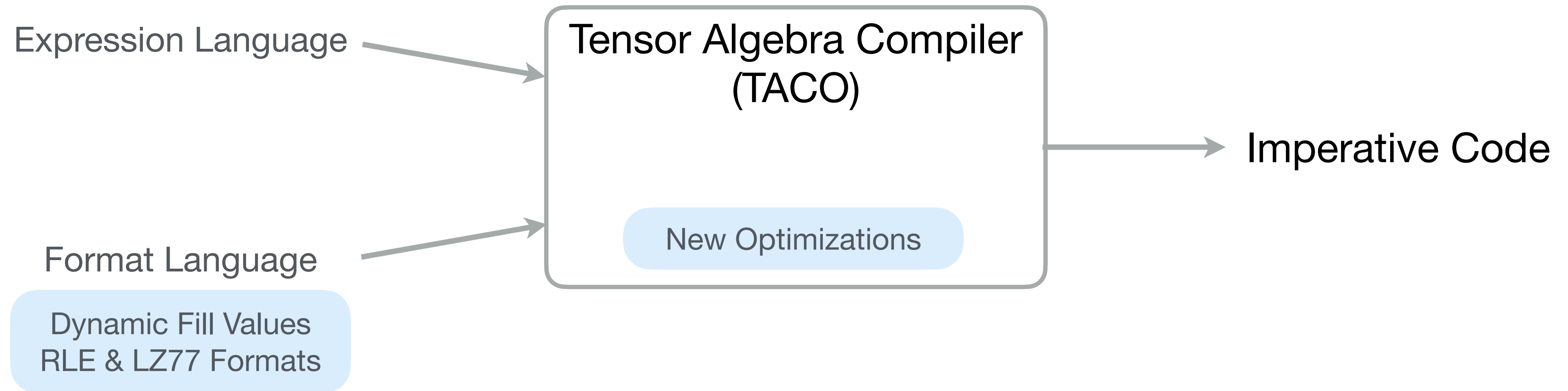
Tensor formats

<table border="1"> <tr><th>Coordinate matrix</th></tr> <tr><td>Compressed</td></tr> <tr><td>Singleton</td></tr> </table>	Coordinate matrix	Compressed	Singleton	<table border="1"> <tr><th>CSR</th></tr> <tr><td>Dense</td></tr> <tr><td>Compressed</td></tr> </table> <p>[Tinney and Walker, 1967]</p>	CSR	Dense	Compressed	<table border="1"> <tr><th>Dense array tensor</th></tr> <tr><td>Dense</td></tr> <tr><td>Dense</td></tr> <tr><td>Dense</td></tr> </table>	Dense array tensor	Dense	Dense	Dense	<table border="1"> <tr><th>Coordinate tensor</th></tr> <tr><td>Compressed</td></tr> <tr><td>Singleton</td></tr> <tr><td>Singleton</td></tr> </table>	Coordinate tensor	Compressed	Singleton	Singleton					
Coordinate matrix																						
Compressed																						
Singleton																						
CSR																						
Dense																						
Compressed																						
Dense array tensor																						
Dense																						
Dense																						
Dense																						
Coordinate tensor																						
Compressed																						
Singleton																						
Singleton																						
<table border="1"> <tr><th>Mode-generic tensor</th></tr> <tr><td>Compressed</td></tr> <tr><td>Singleton</td></tr> <tr><td>Dense</td></tr> <tr><td>Dense</td></tr> </table> <p>[Baskaran et al. 2012]</p>	Mode-generic tensor	Compressed	Singleton	Dense	Dense	<table border="1"> <tr><th>BCSR</th></tr> <tr><td>Dense</td></tr> <tr><td>Compressed</td></tr> <tr><td>Dense</td></tr> <tr><td>Dense</td></tr> </table> <p>[Im and Yelick 1998]</p>	BCSR	Dense	Compressed	Dense	Dense	<table border="1"> <tr><th>CSB</th></tr> <tr><td>Dense</td></tr> <tr><td>Dense</td></tr> <tr><td>Compressed</td></tr> <tr><td>Singleton</td></tr> </table> <p>[Buluç et al. 2009]</p>	CSB	Dense	Dense	Compressed	Singleton	<table border="1"> <tr><th>ELLPACK</th></tr> <tr><td>Dense</td></tr> <tr><td>Dense</td></tr> <tr><td>Singleton</td></tr> </table> <p>[Kincaid et al. 1989]</p>	ELLPACK	Dense	Dense	Singleton
Mode-generic tensor																						
Compressed																						
Singleton																						
Dense																						
Dense																						
BCSR																						
Dense																						
Compressed																						
Dense																						
Dense																						
CSB																						
Dense																						
Dense																						
Compressed																						
Singleton																						
ELLPACK																						
Dense																						
Dense																						
Singleton																						









# Sparsity beyond zero Fill Values

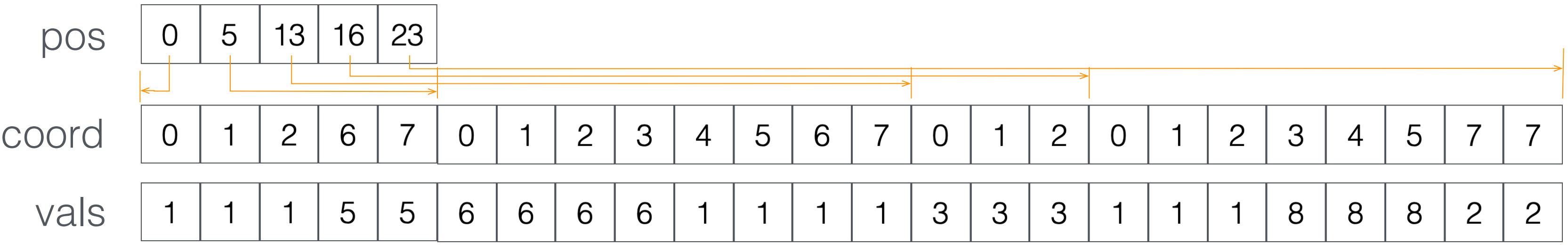
	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

# Sparsity beyond zero Fill Values

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

# Sparsity beyond zero Fill Values

## Compressed Level Format



	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

# Sparsity beyond zero Fill Values

## Compressed Level Format

pos	0	5	13	16	23																			
coord	0	1	2	6	7	0	1	2	3	4	5	6	7	0	1	2	0	1	2	3	4	5	7	7
vals	1	1	1	5	5	6	6	6	6	1	1	1	1	3	3	3	1	1	1	8	8	8	2	2

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2



# Sparsity beyond zero Fill Values

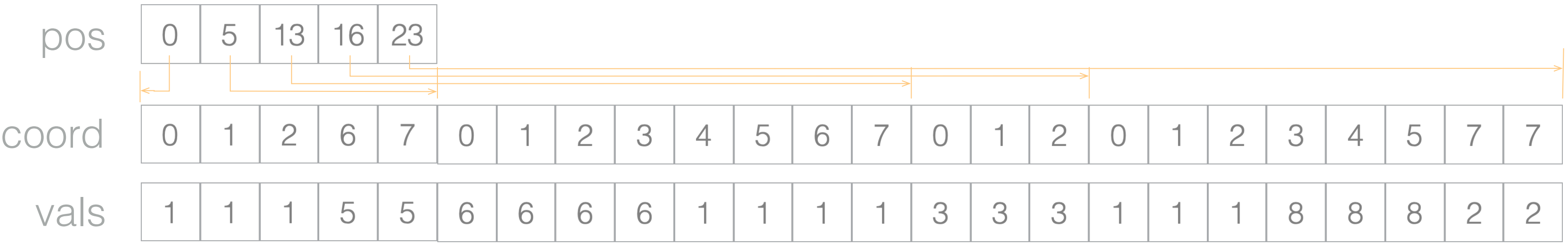
## Compressed Level Format

pos	0	5	13	16	23																			
coord	0	1	2	6	7	0	1	2	3	4	5	6	7	0	1	2	0	1	2	3	4	5	7	7
vals	1	1	1	5	5	6	6	6	6	1	1	1	1	3	3	3	1	1	1	8	8	8	2	2

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

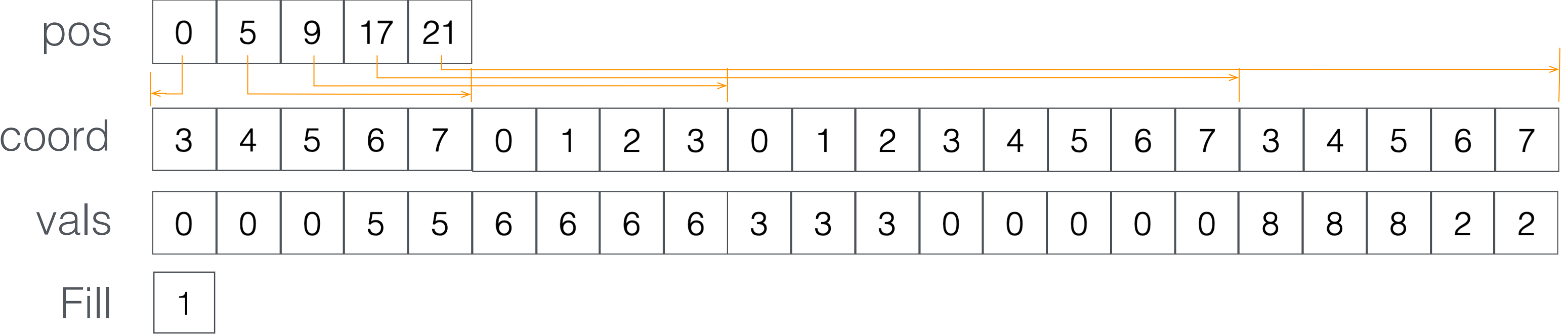
# Sparsity beyond zero Fill Values

## Compressed Level Format



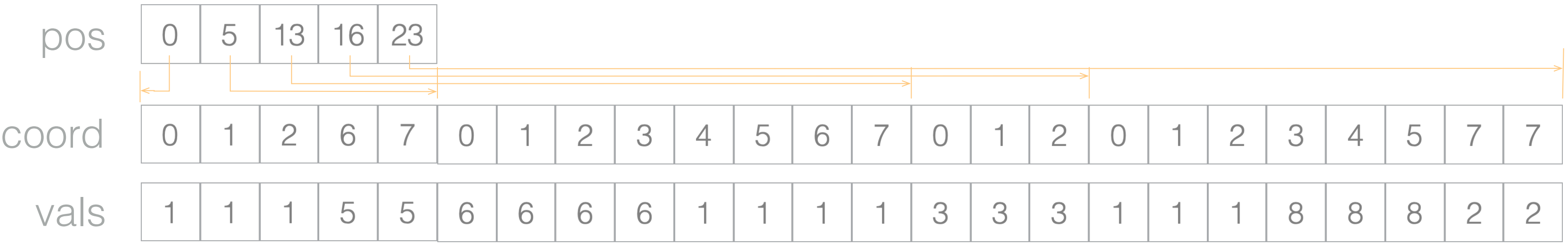
	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

## Compressed Level Format with a Fill Value [\[Henry et al. OOPSLA, 2021\]](#)



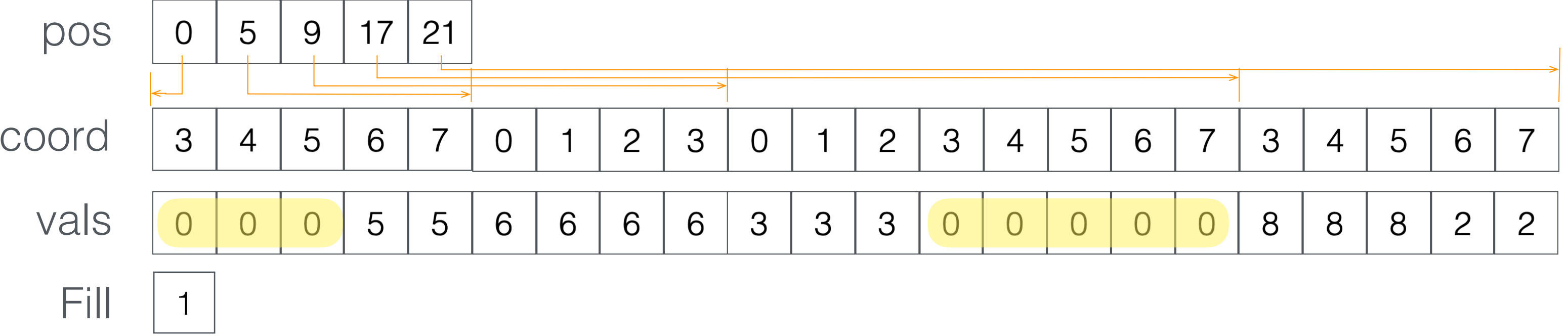
# Sparsity beyond zero Fill Values

## Compressed Level Format



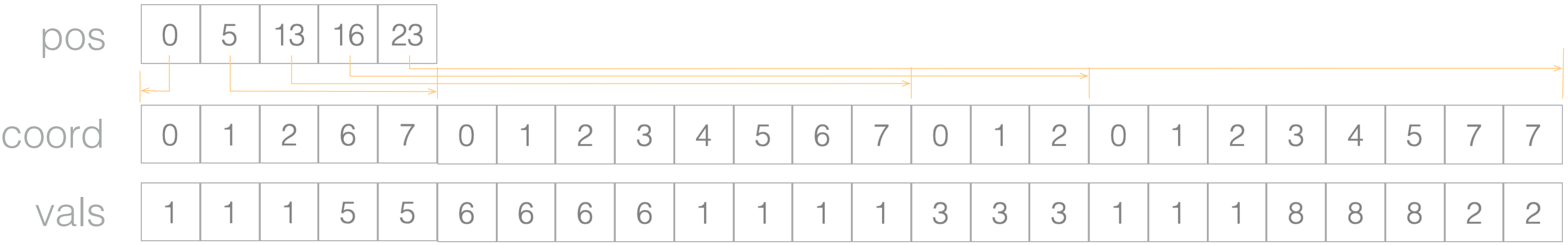
	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

## Compressed Level Format with a Fill Value [\[Henry et al. OOPSLA, 2021\]](#)



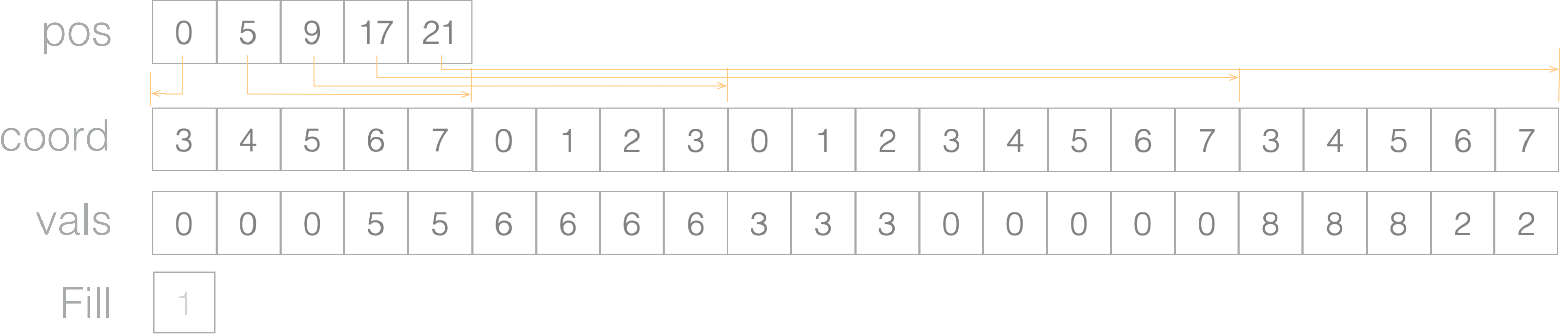
# Sparsity beyond zero Fill Values

## Compressed Level Format



	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

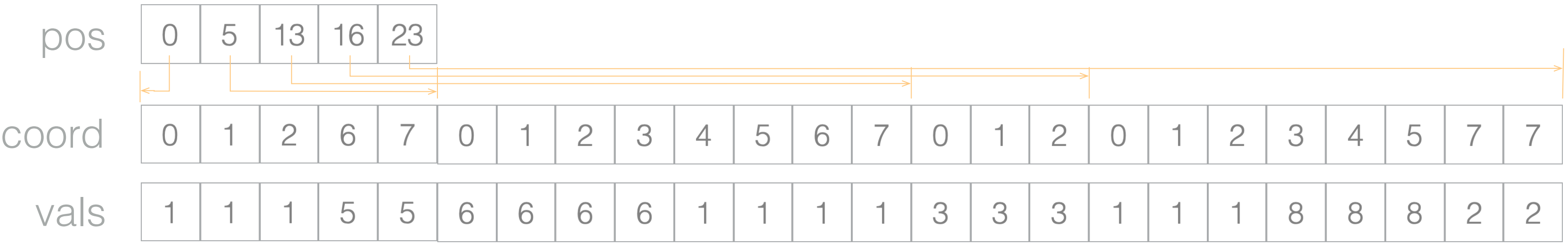
## Compressed Level Format with a Fill Value [Henry et al. OOPSLA, 2021]



Why are we limited to a single repeated value?

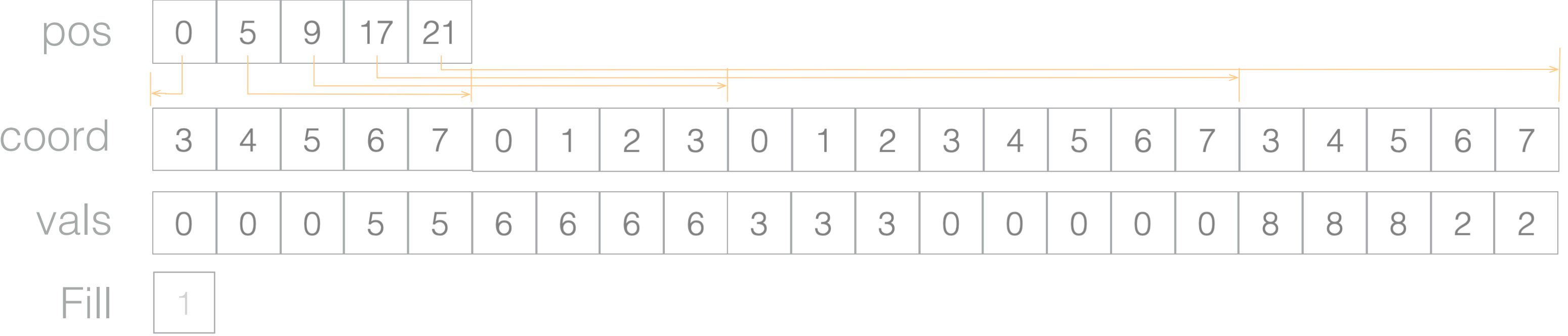
# Sparsity beyond zero Fill Values

## Compressed Level Format



	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

## Compressed Level Format with a Fill Value [Henry et al. OOPSLA, 2021]



Why are we limited to a single repeated value?

Run Length Encoding (RLE)

# Dynamic Fill Values

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2



# Dynamic Fill Values

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

# Dynamic Fill Values

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

# Dynamic Fill Values

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

Fill

?

# Dynamic Fill Values

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

coord	0	3	6
-------	---	---	---

vals	1	0	5
------	---	---	---

Fill

# Dynamic Fill Values

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5

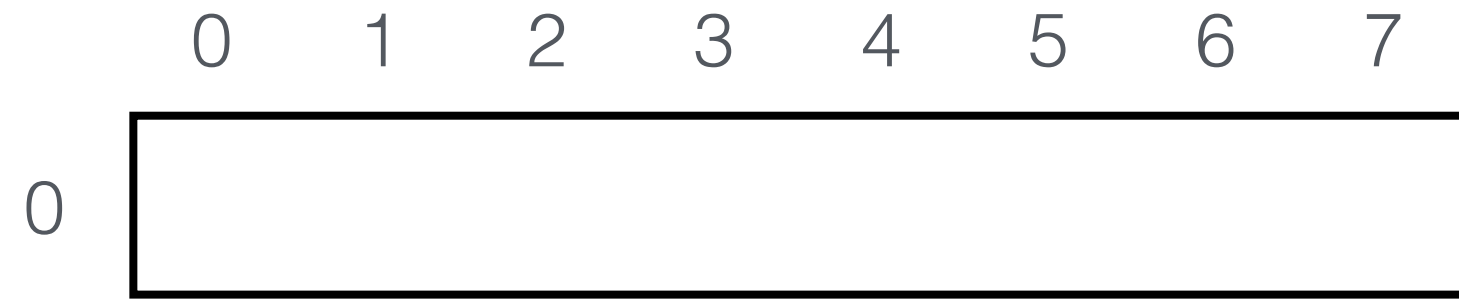
	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

coord	0	3	6
-------	---	---	---

vals	1	0	5
------	---	---	---

Fill

# Dynamic Fill Values



	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

coord

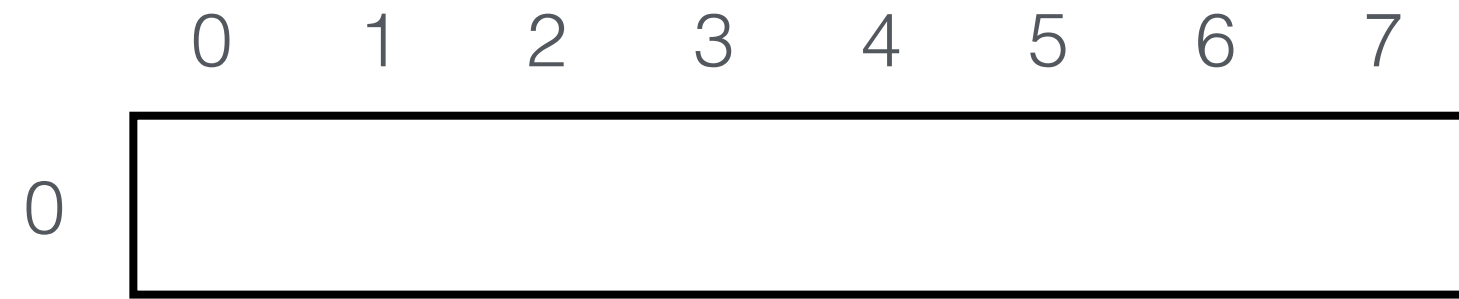
0	3	6
---	---	---

vals

1	0	5
---	---	---

Fill

# Dynamic Fill Values

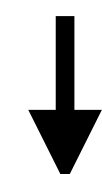
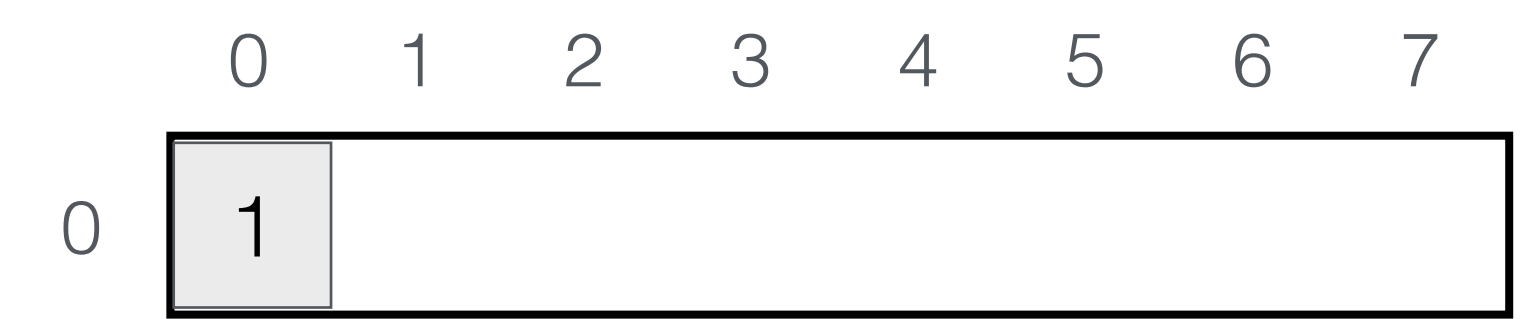


coord	0	3	6
vals	1	0	5
Fill			

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2



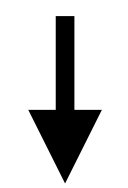
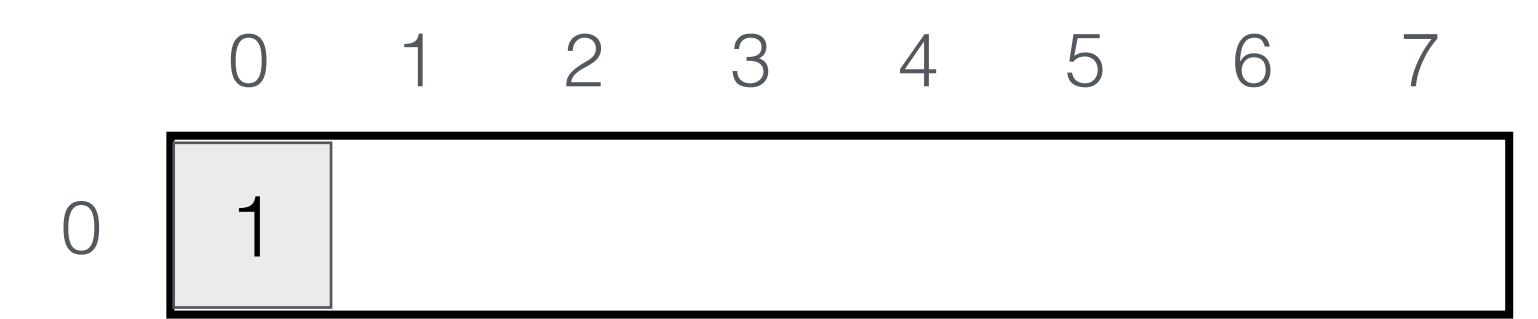
# Dynamic Fill Values



coord	0	3	6
vals	1	0	5
Fill			

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

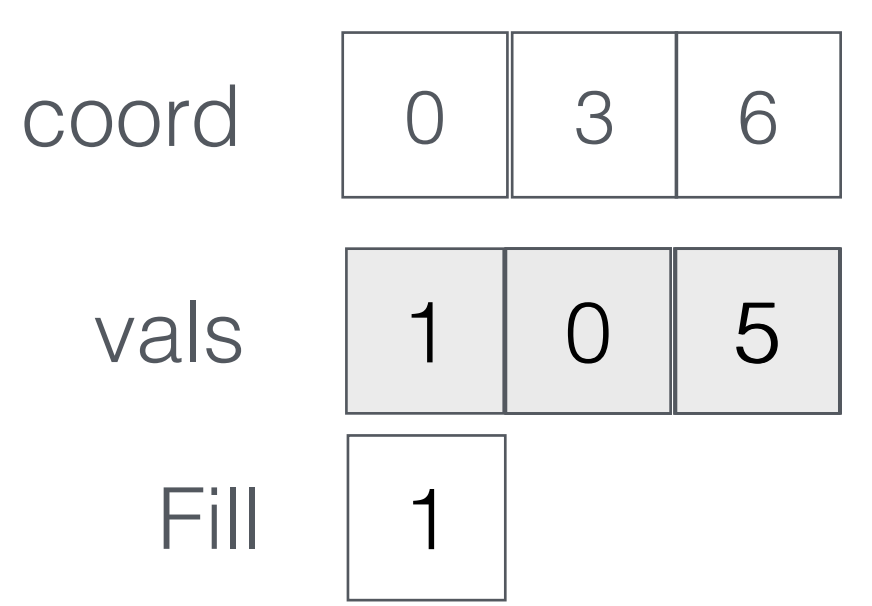
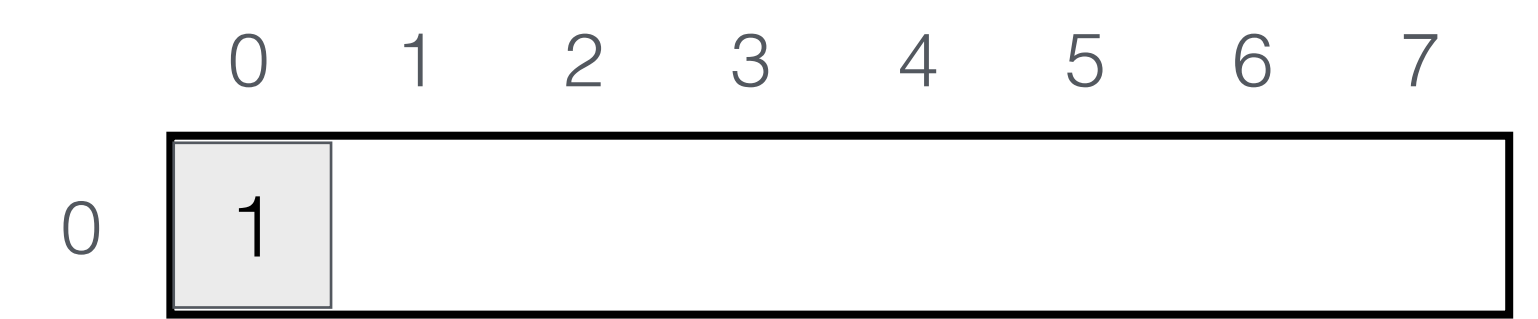
# Dynamic Fill Values



coord	0	3	6
vals	1	0	5
Fill	1		

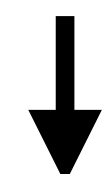
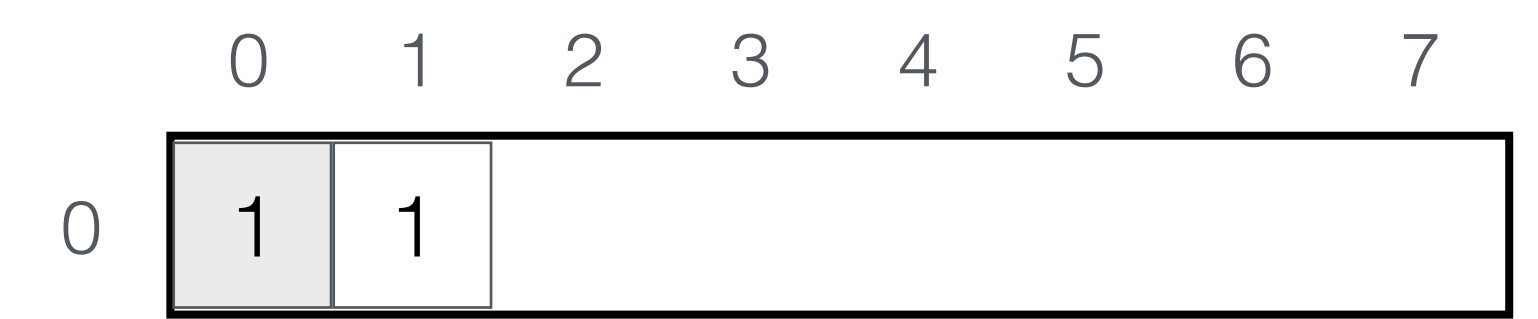
	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

# Dynamic Fill Values



	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

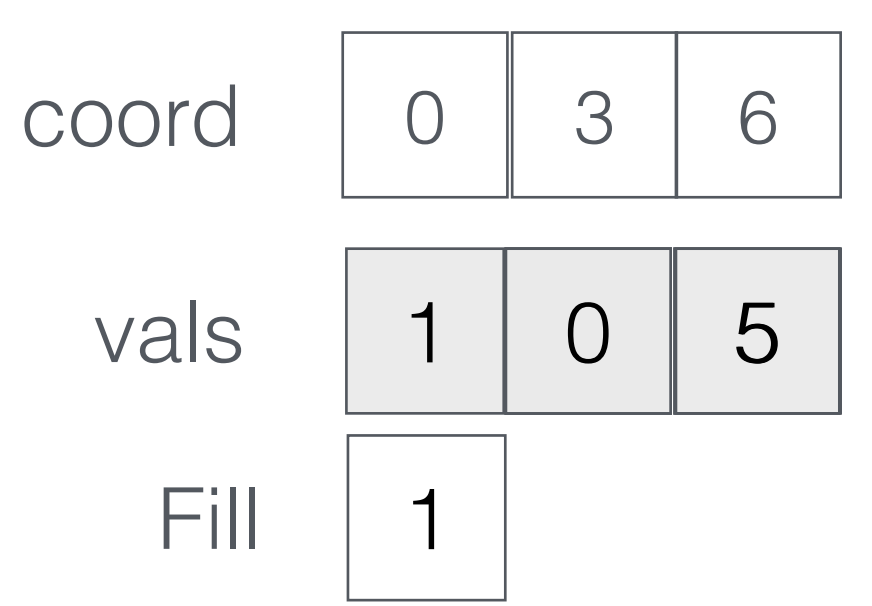
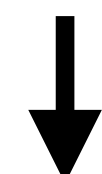
# Dynamic Fill Values



coord	0	3	6
vals	1	0	5
Fill	1		

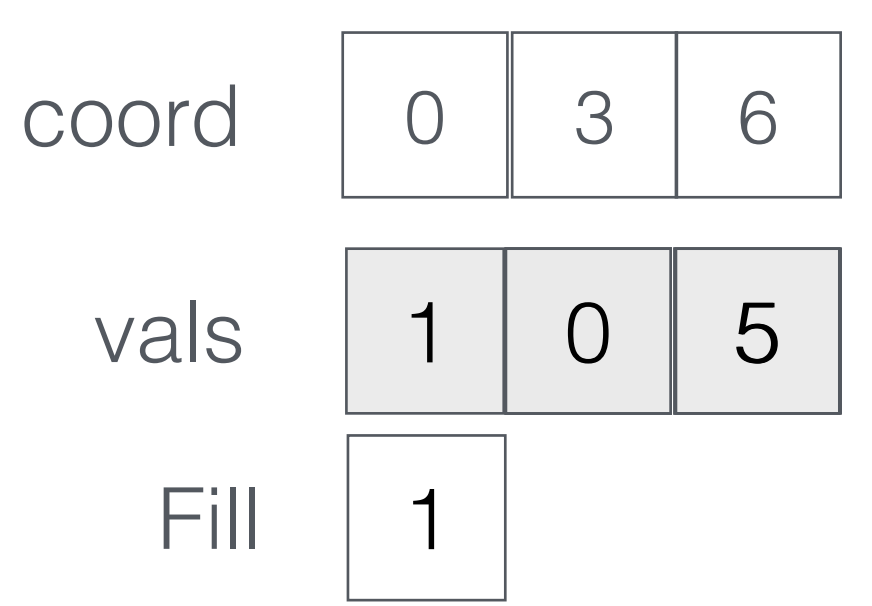
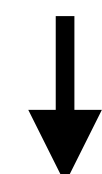
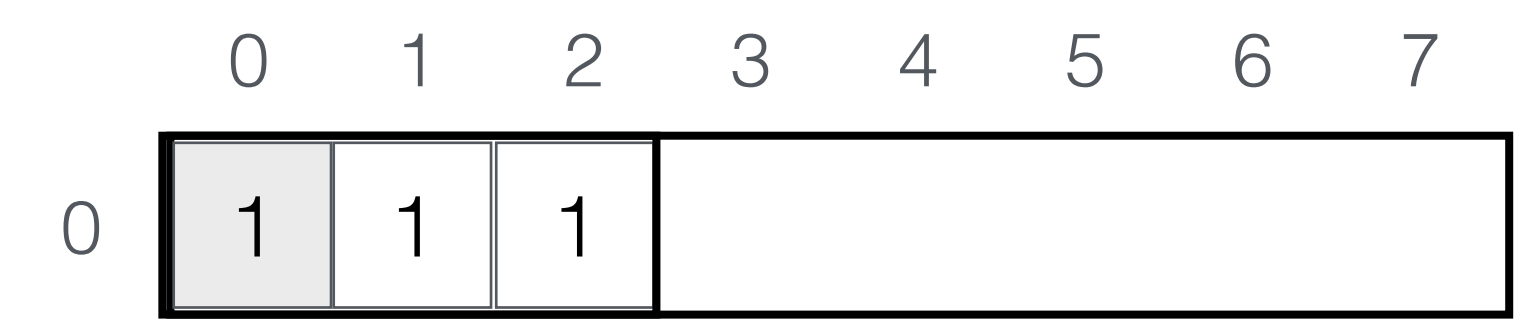
	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

# Dynamic Fill Values



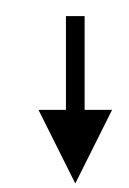
	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

# Dynamic Fill Values



	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

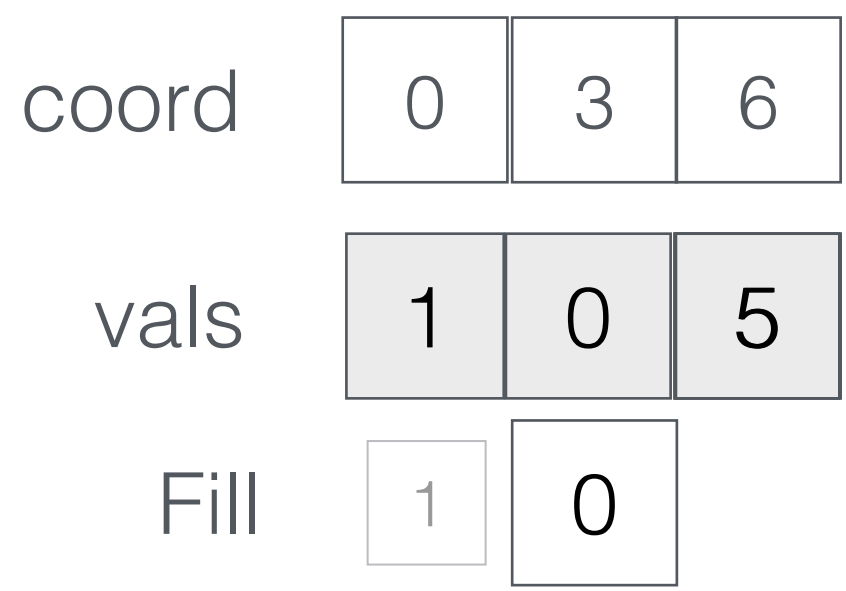
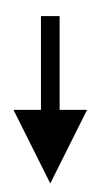
# Dynamic Fill Values



	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

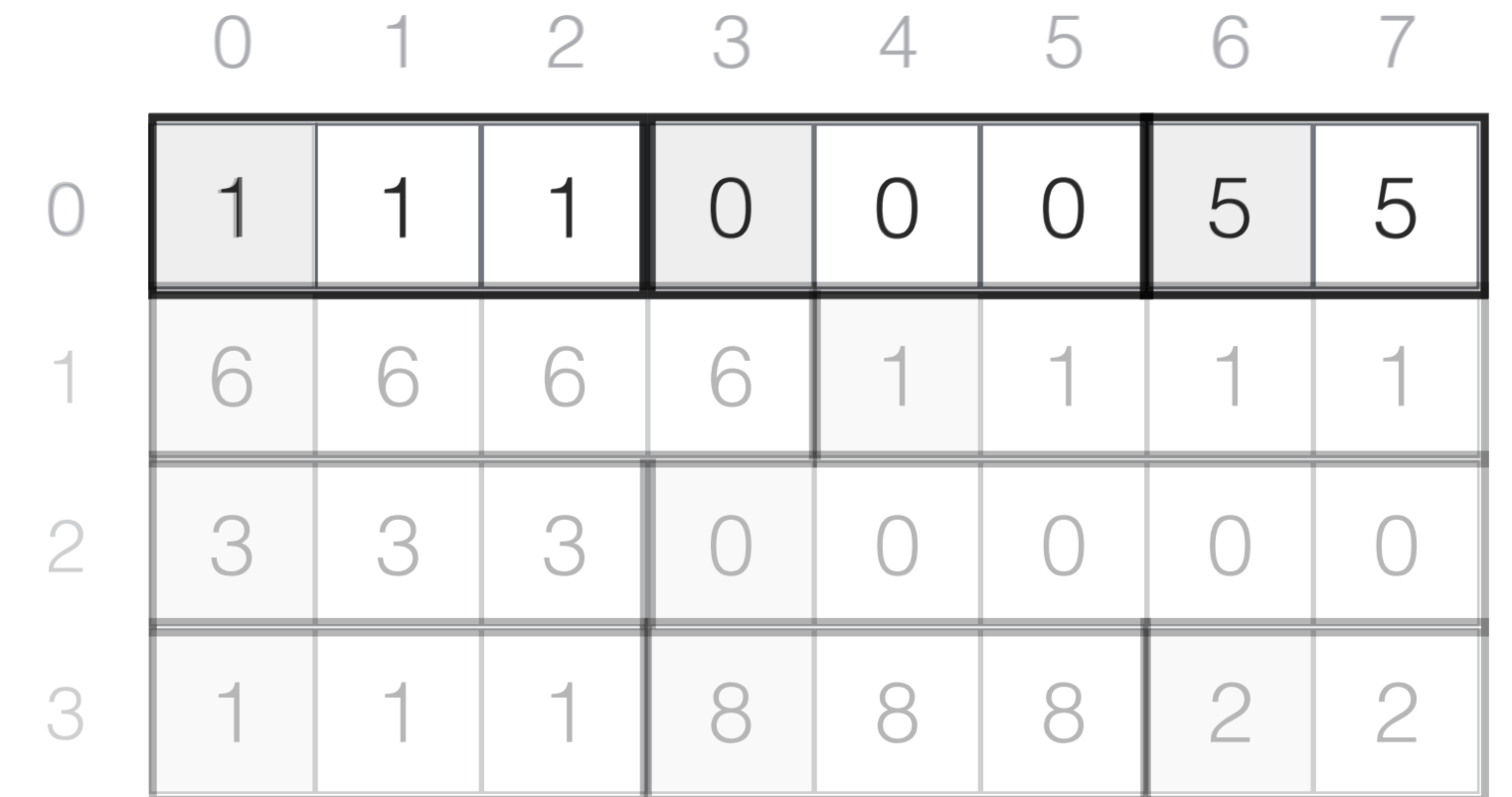
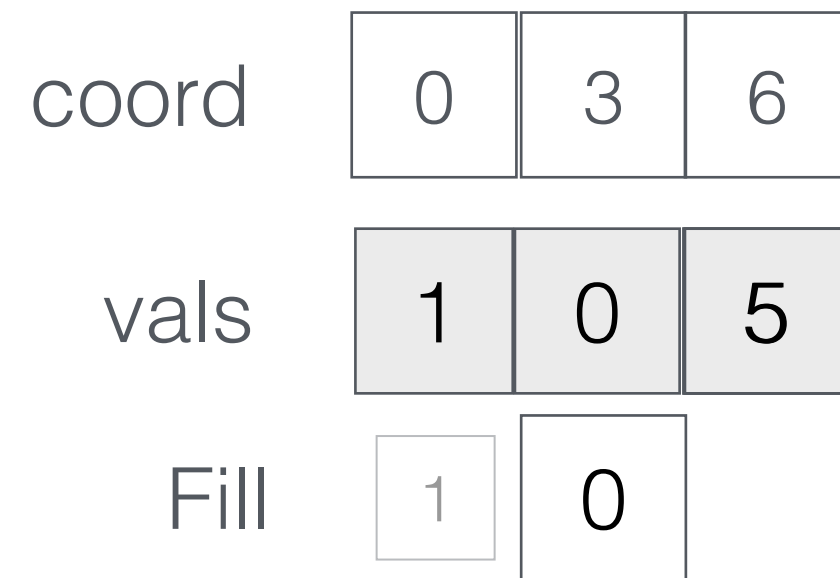
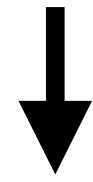


# Dynamic Fill Values



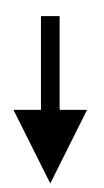
	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

# Dynamic Fill Values



# Dynamic Fill Values

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0		

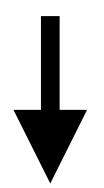


coord	0	3	6
vals	1	0	5
Fill	1	0	

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

# Dynamic Fill Values

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0		

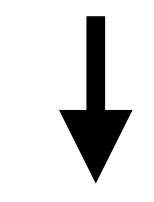


coord	0	3	6
vals	1	0	5
Fill	1	0	

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

# Dynamic Fill Values

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	

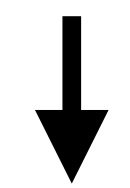


coord	0	3	6
vals	1	0	5
Fill	1	0	5

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

# Dynamic Fill Values

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5



coord	0	3	6
vals	1	0	5
Fill	1	0	5

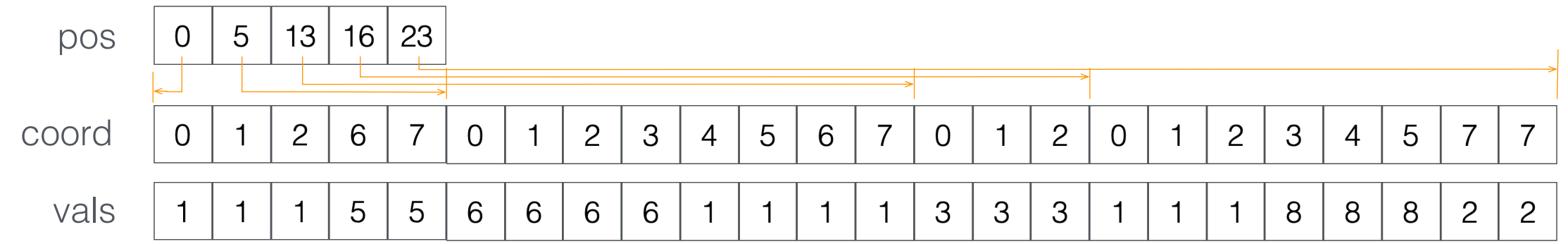
	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

# Dynamic Fill Values

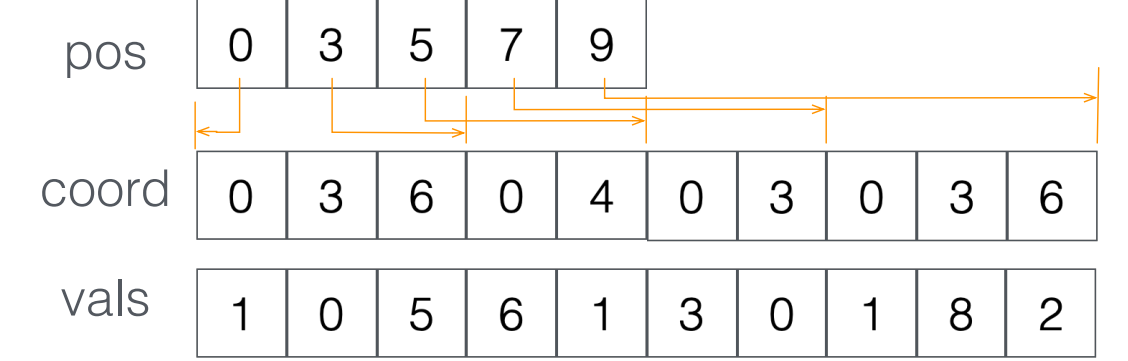
	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	5	5
1	6	6	6	6	1	1	1	1
2	3	3	3	0	0	0	0	0
3	1	1	1	8	8	8	2	2

## Compressed Level Format

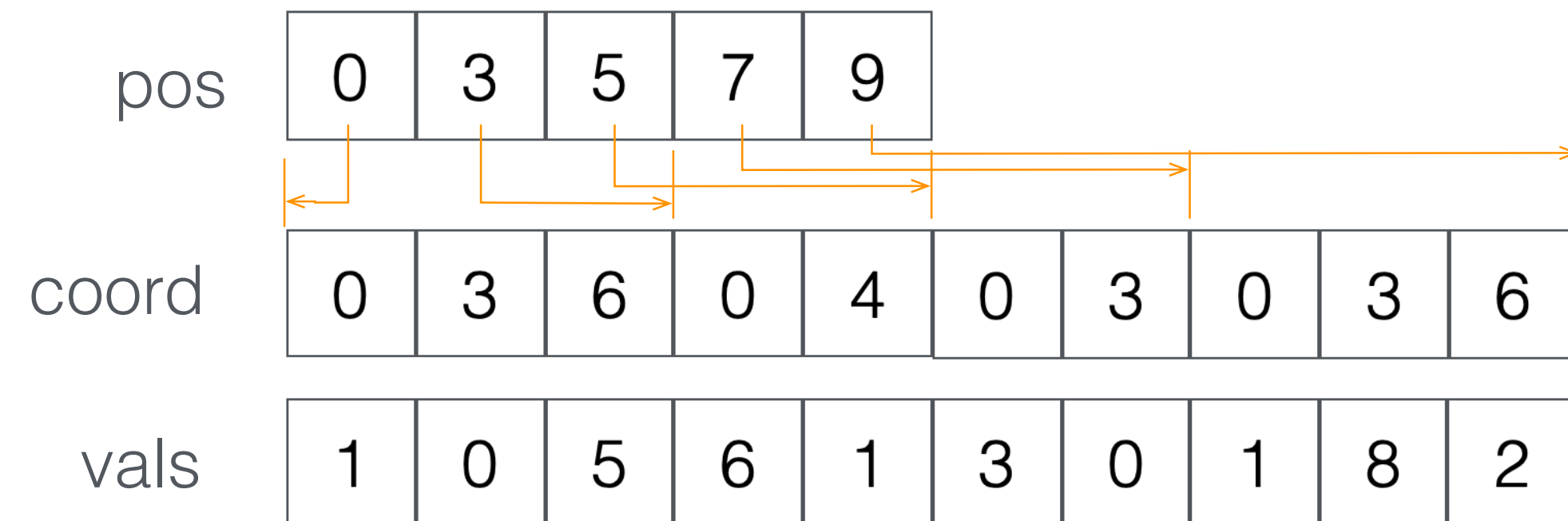


## Run Length Encoding (RLE) Level Format



# New Level Formats

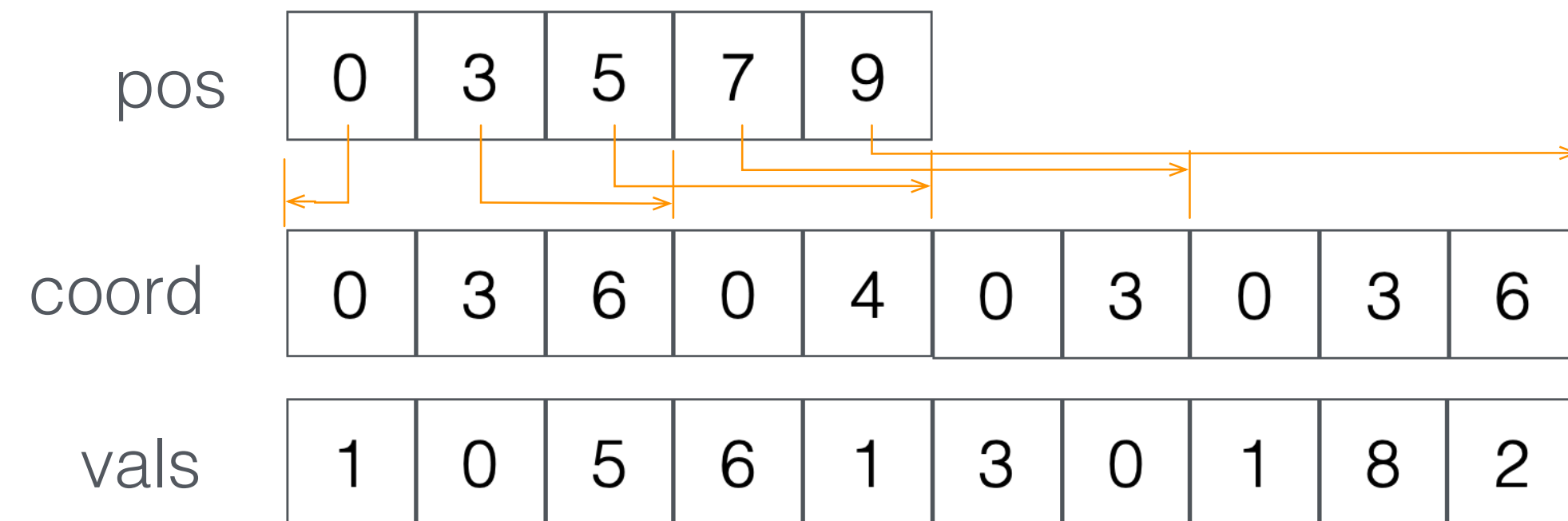
Run Length Encoding (RLE) Level Format





# New Level Formats

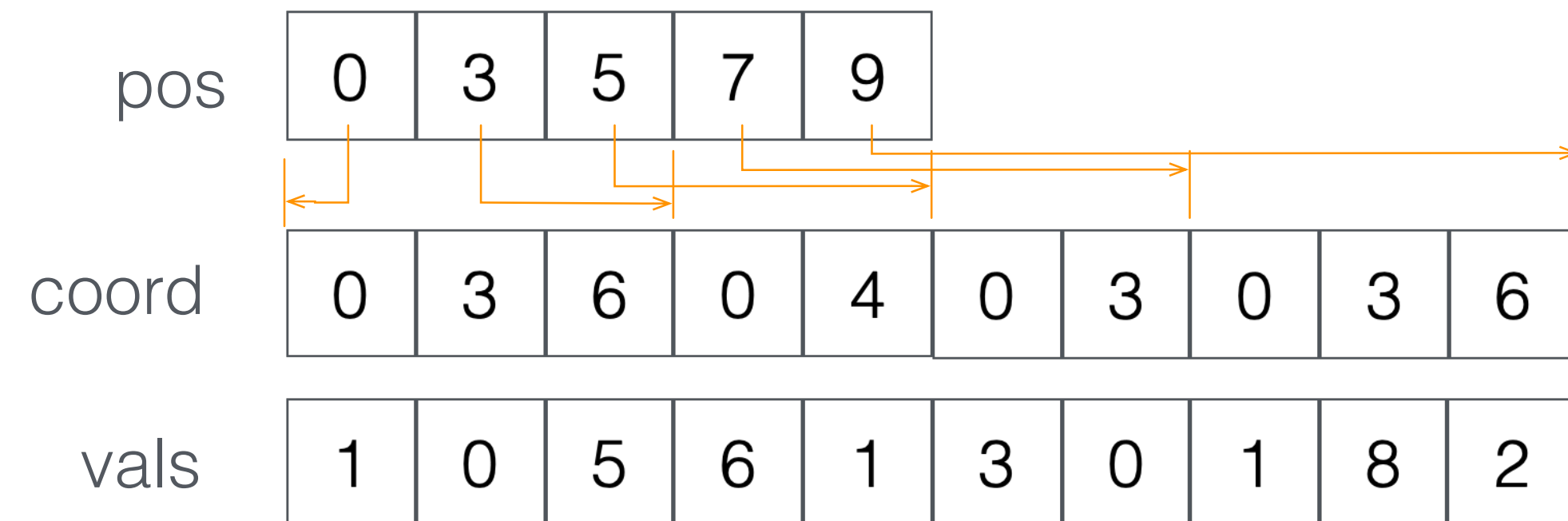
Run Length Encoding (RLE) Level Format



LZ77 Level Format

# New Level Formats

## Run Length Encoding (RLE) Level Format

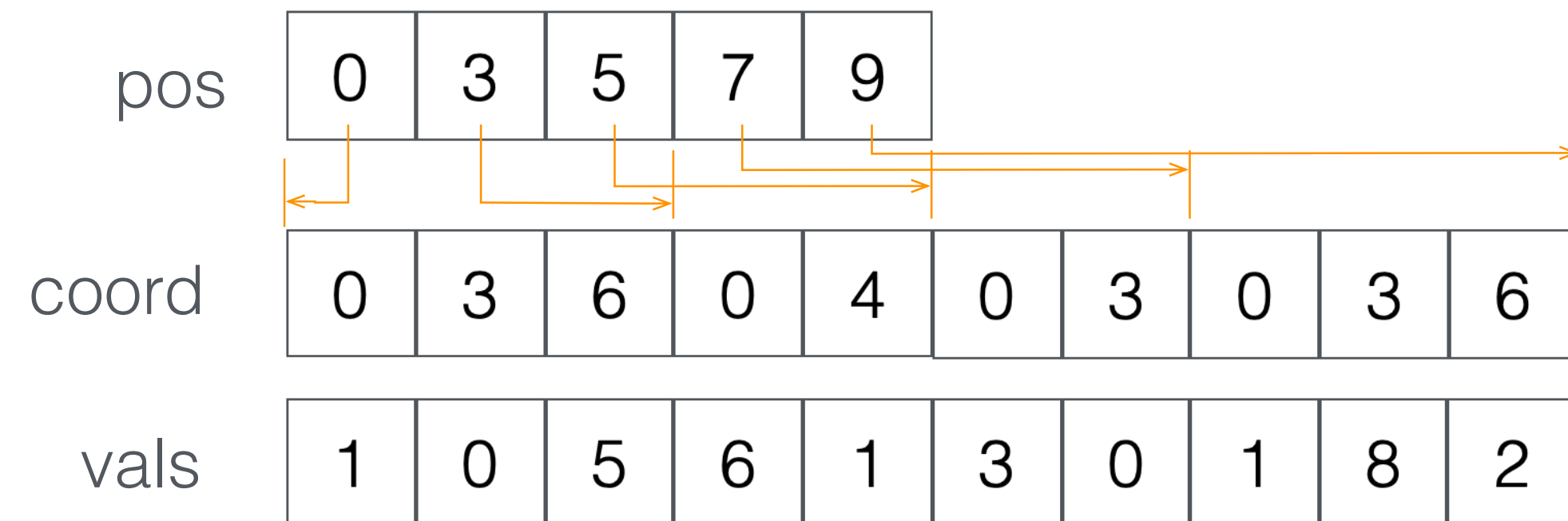


## LZ77 Level Format

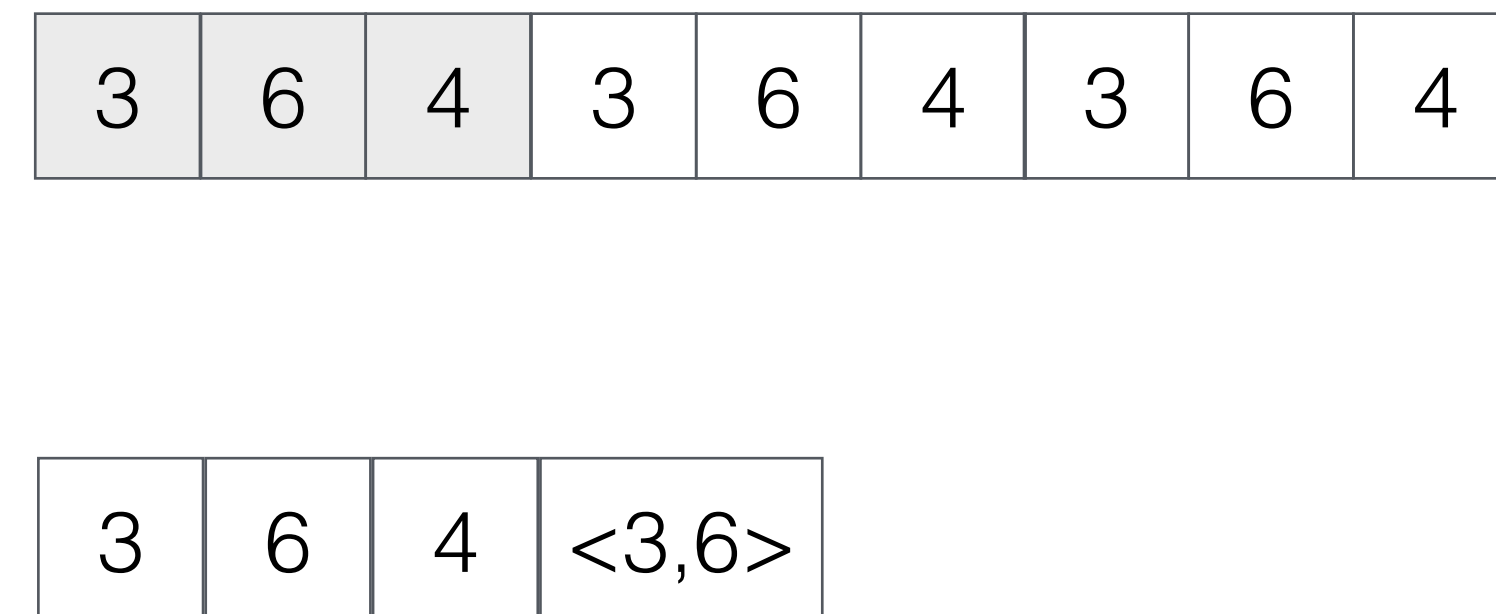


# New Level Formats

## Run Length Encoding (RLE) Level Format

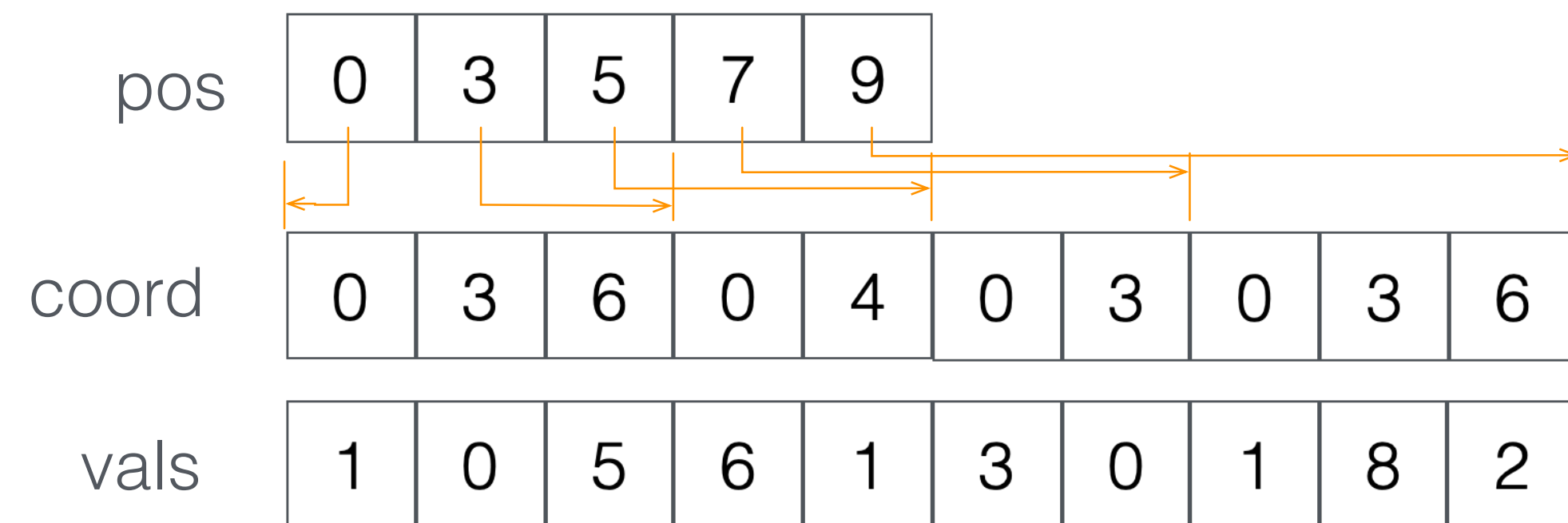


## LZ77 Level Format

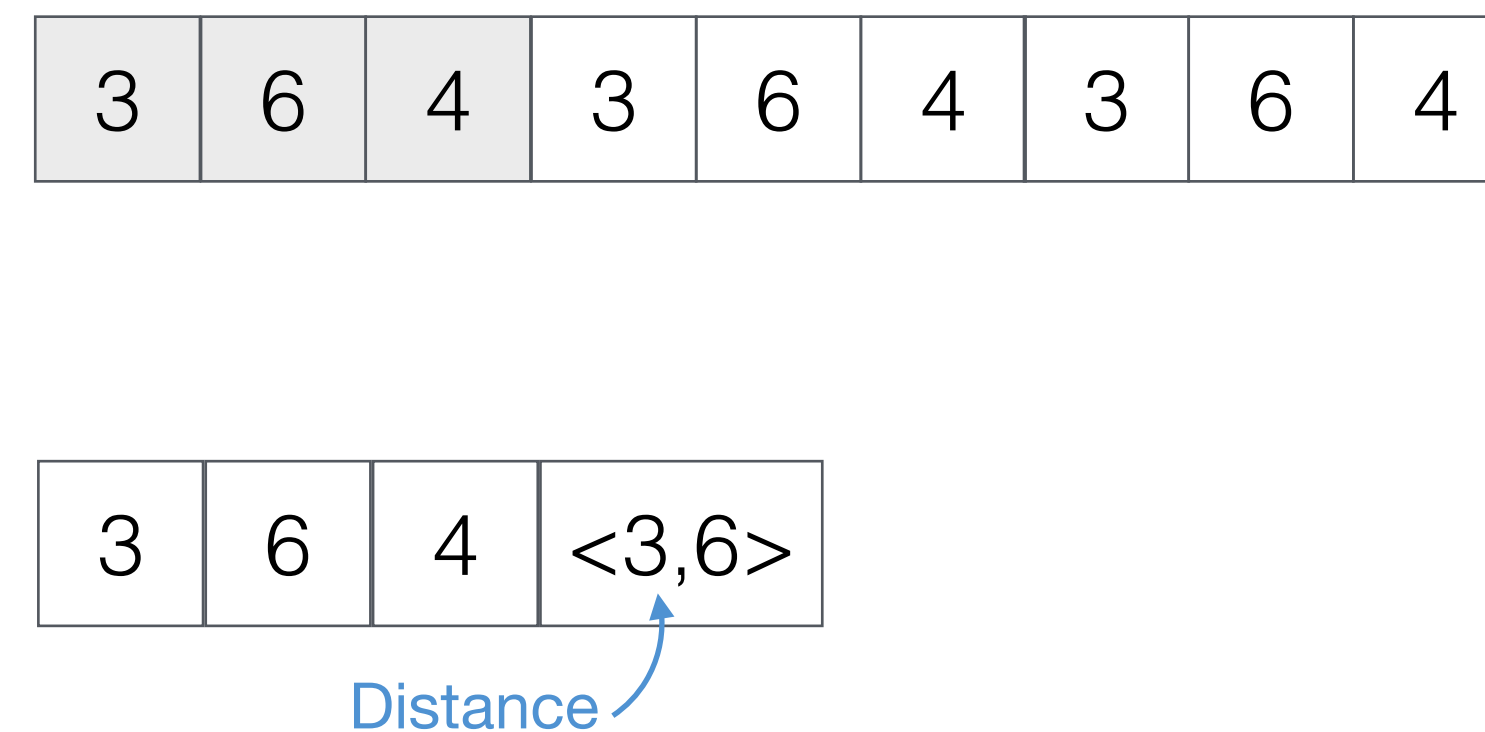


# New Level Formats

## Run Length Encoding (RLE) Level Format

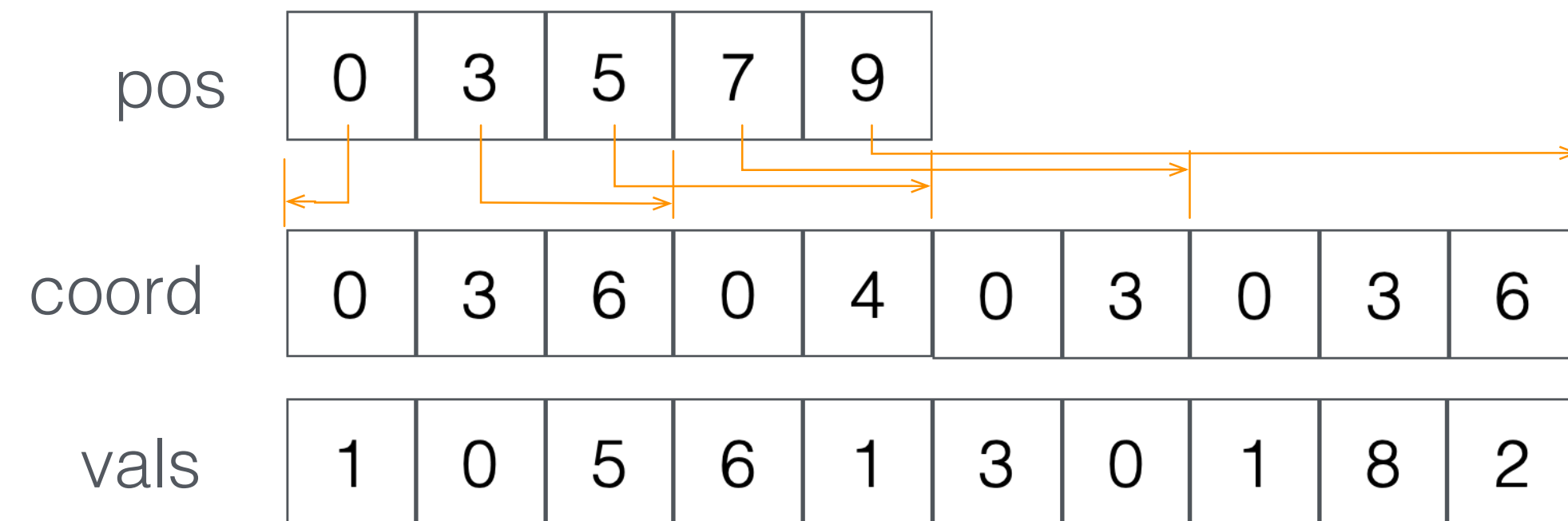


## LZ77 Level Format

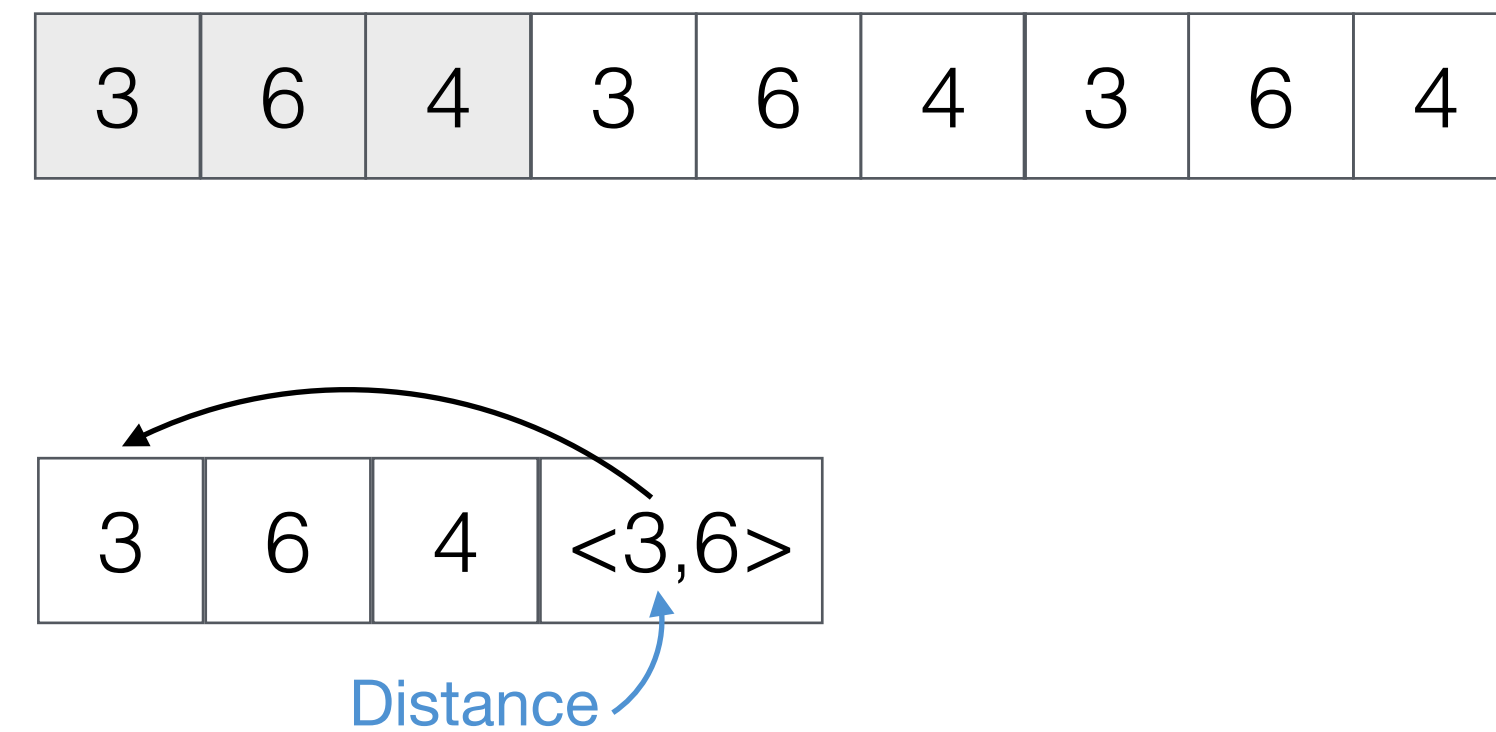


# New Level Formats

## Run Length Encoding (RLE) Level Format

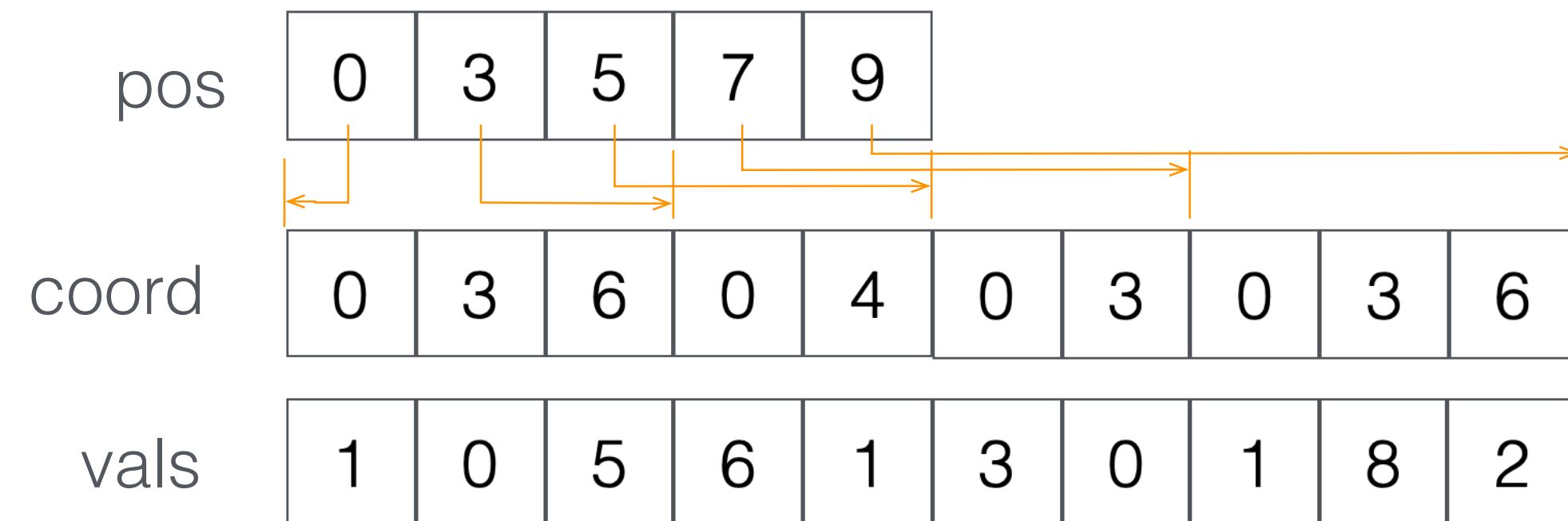


## LZ77 Level Format

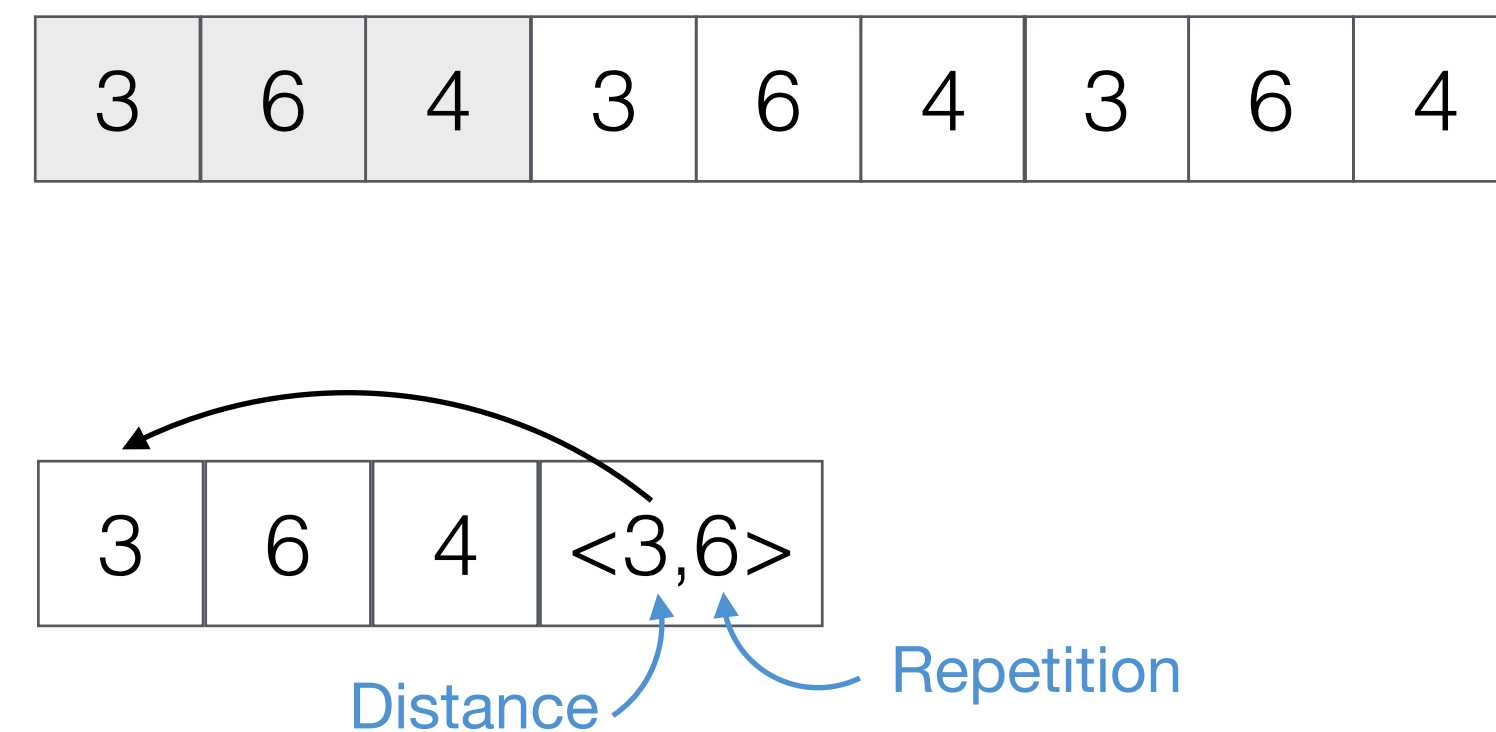


# New Level Formats

## Run Length Encoding (RLE) Level Format

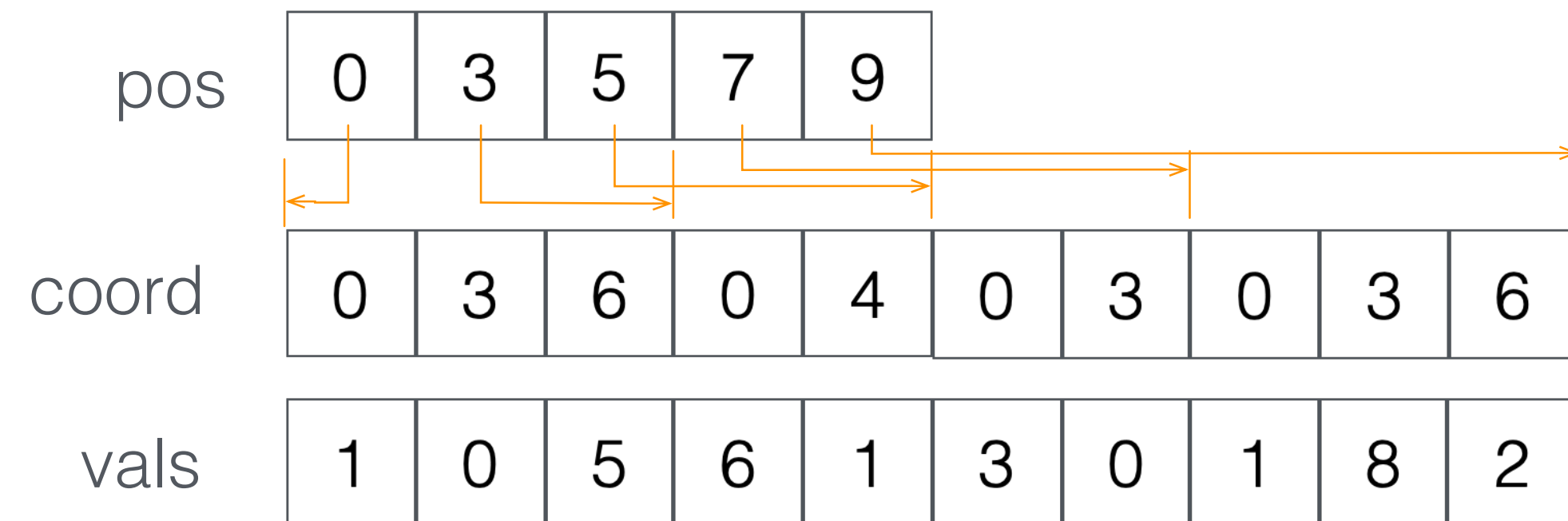


## LZ77 Level Format

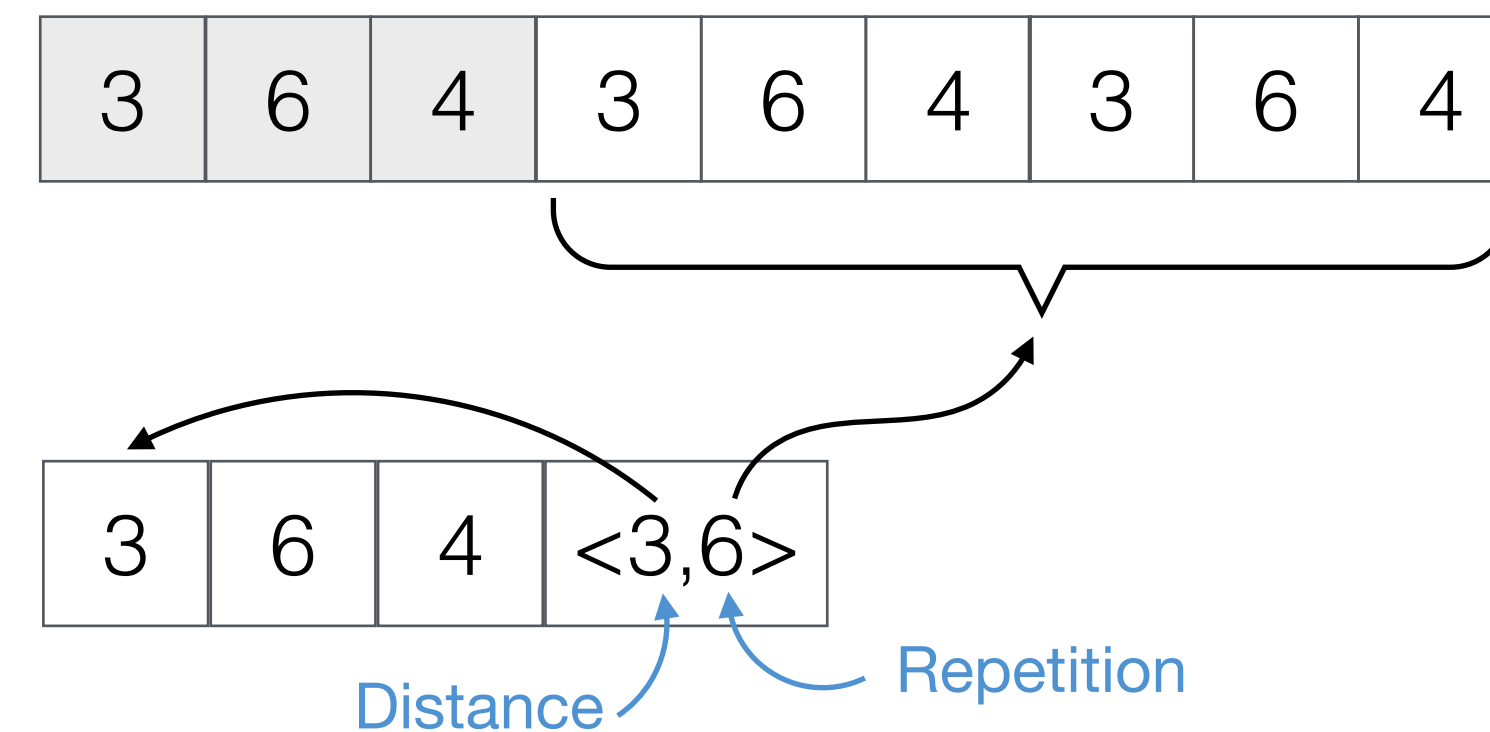


# New Level Formats

## Run Length Encoding (RLE) Level Format

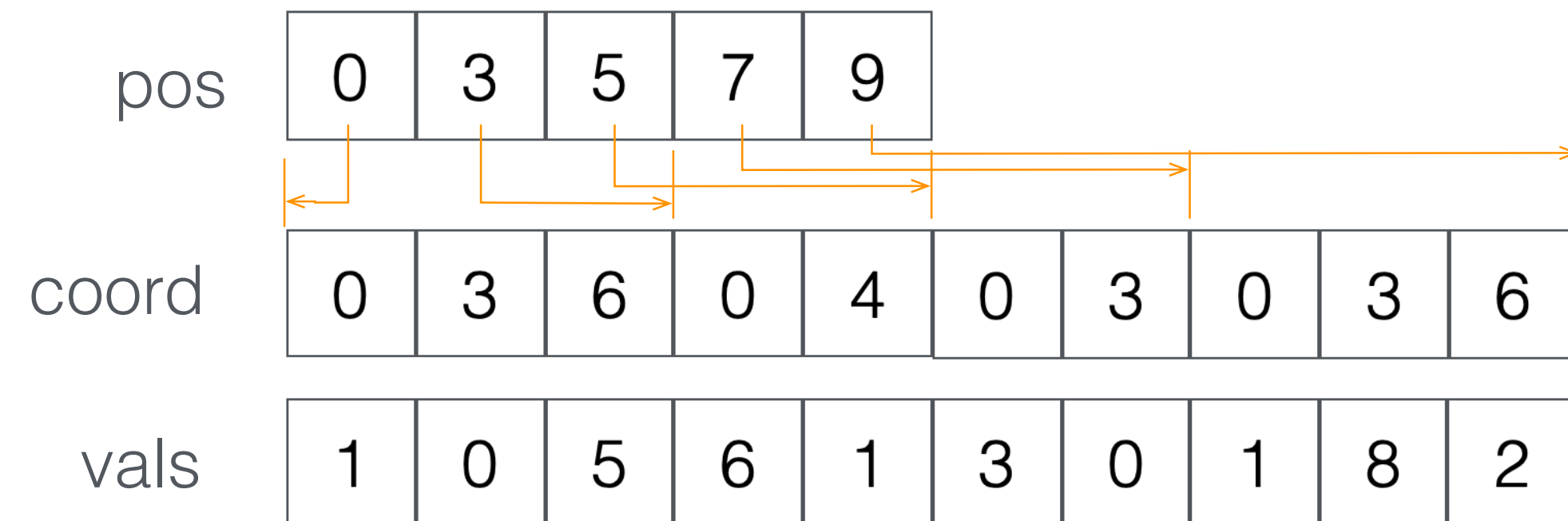


## LZ77 Level Format

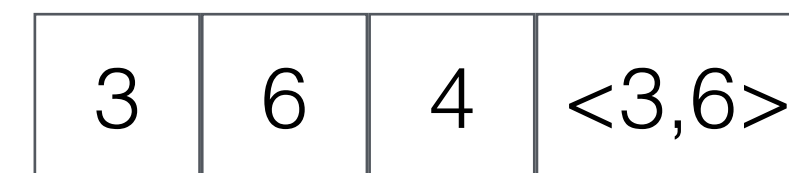


# New Level Formats

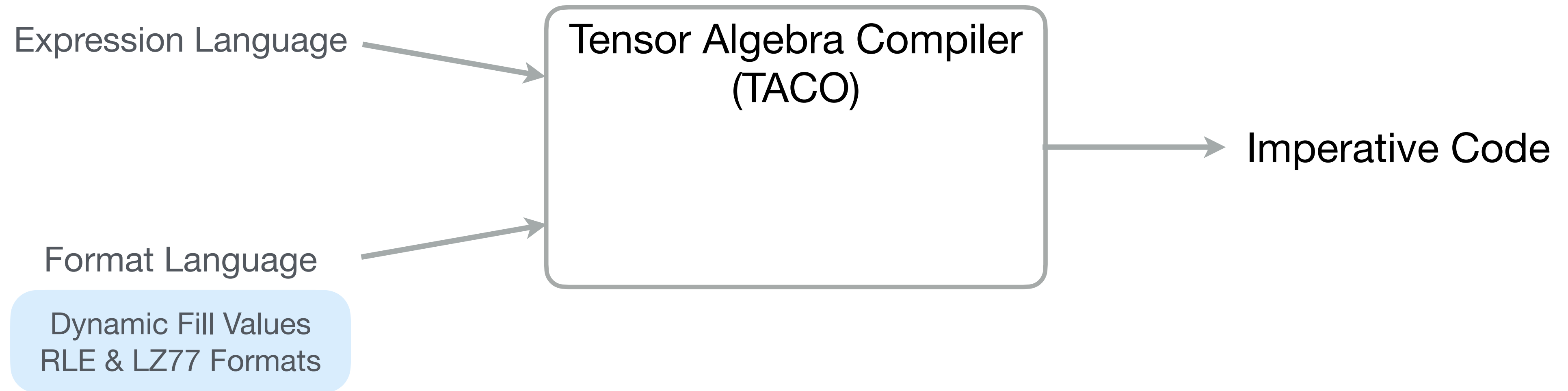
## Run Length Encoding (RLE) Level Format

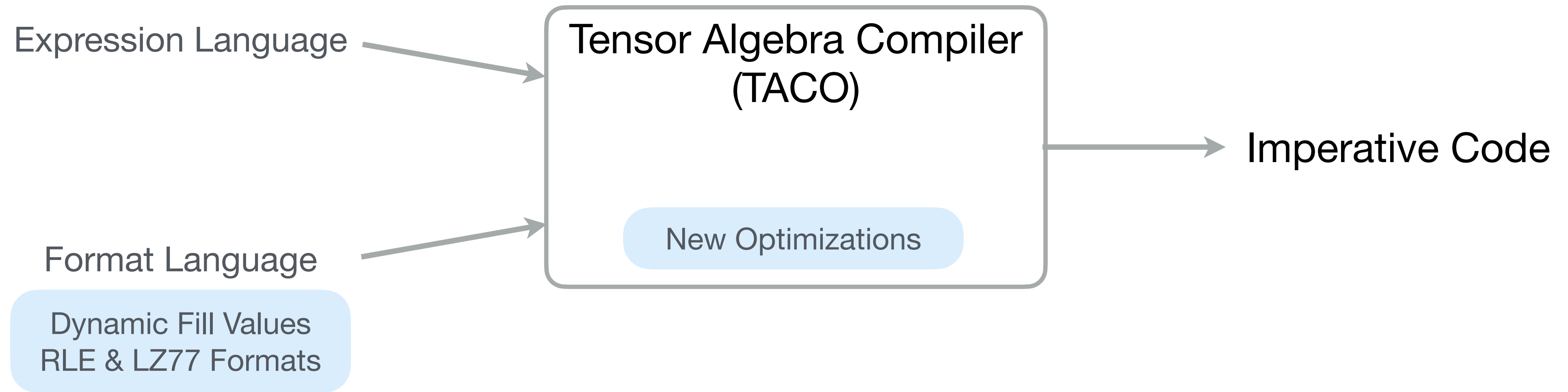


## LZ77 Level Format









# Optimizing computations on lossless compression formats

## Elementwise computation

1	1	1	2	2	2	5	5
---	---	---	---	---	---	---	---

RLE

×

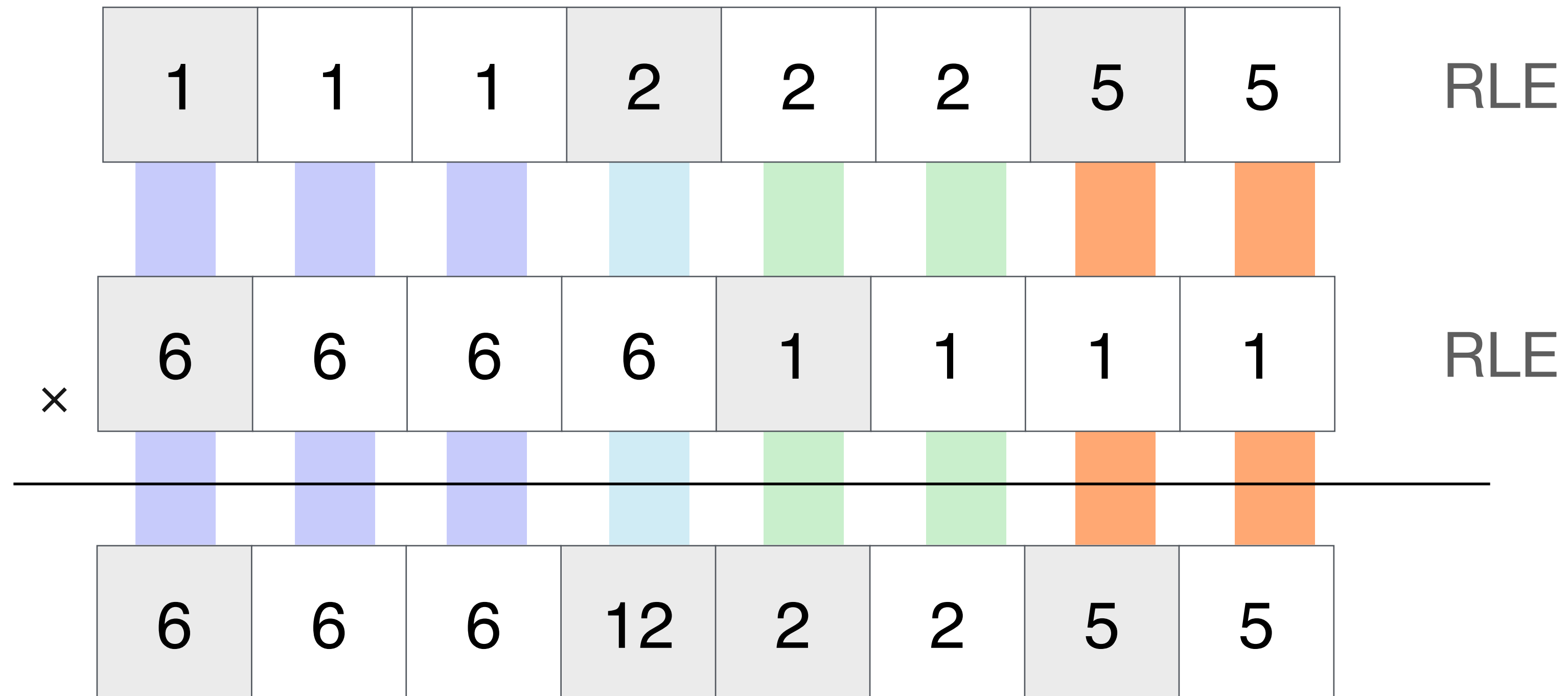
6	6	6	6	1	1	1	1
---	---	---	---	---	---	---	---

RLE

---

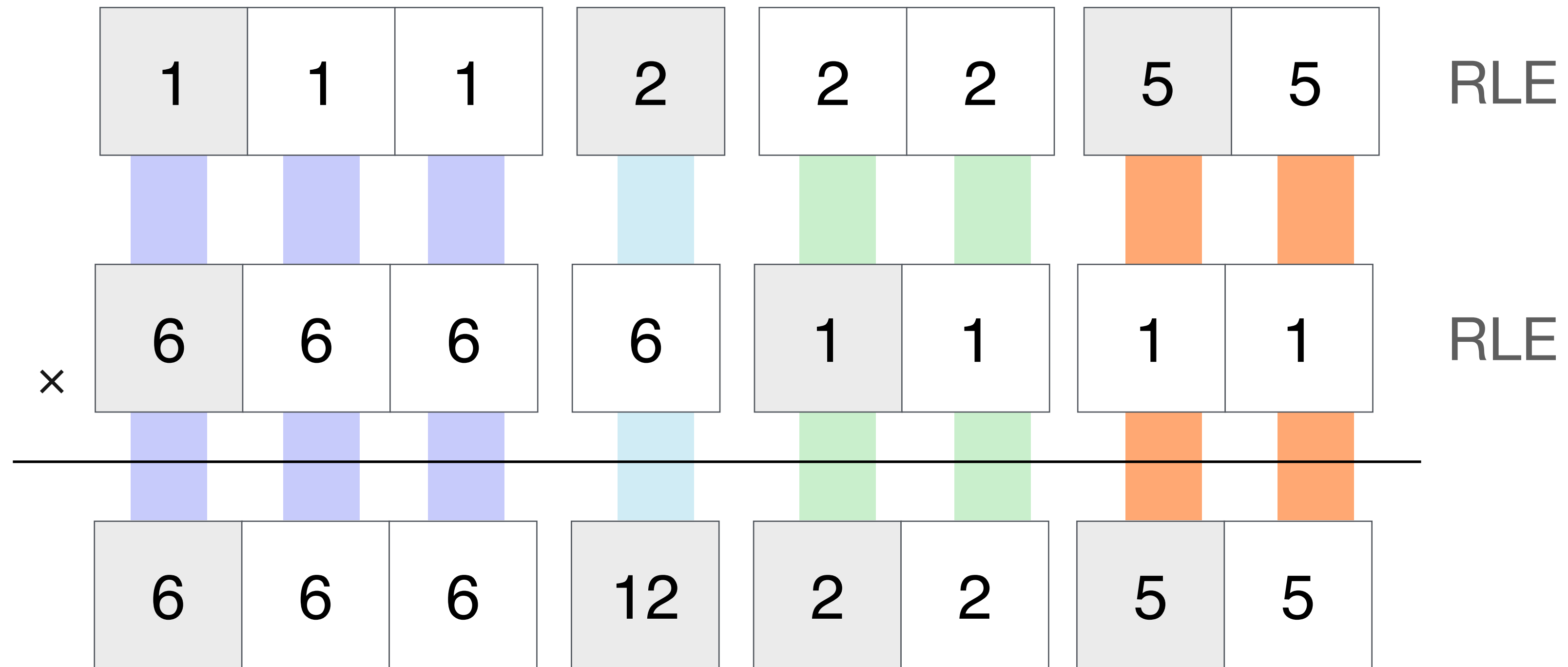
# Optimizing computations on lossless compression formats

## Elementwise computation



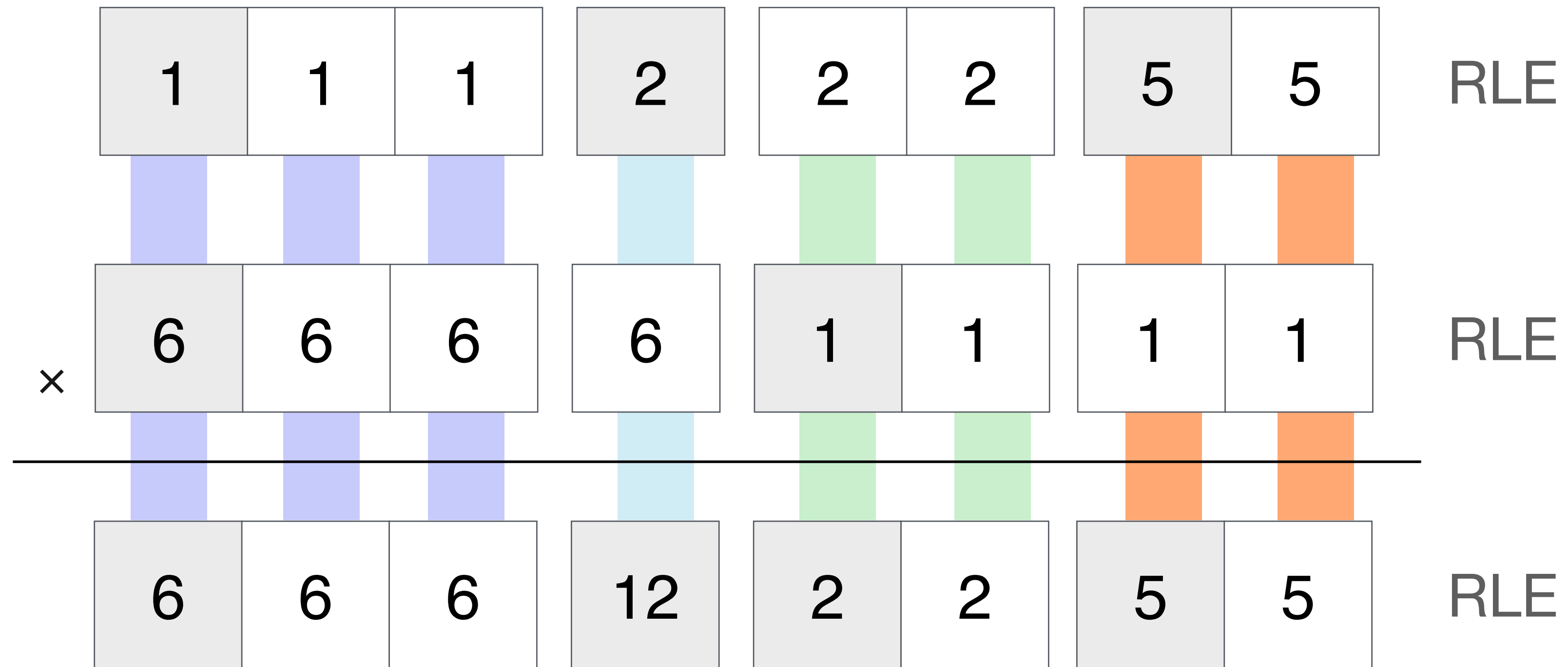
# Optimizing computations on lossless compression formats

## Elementwise computation



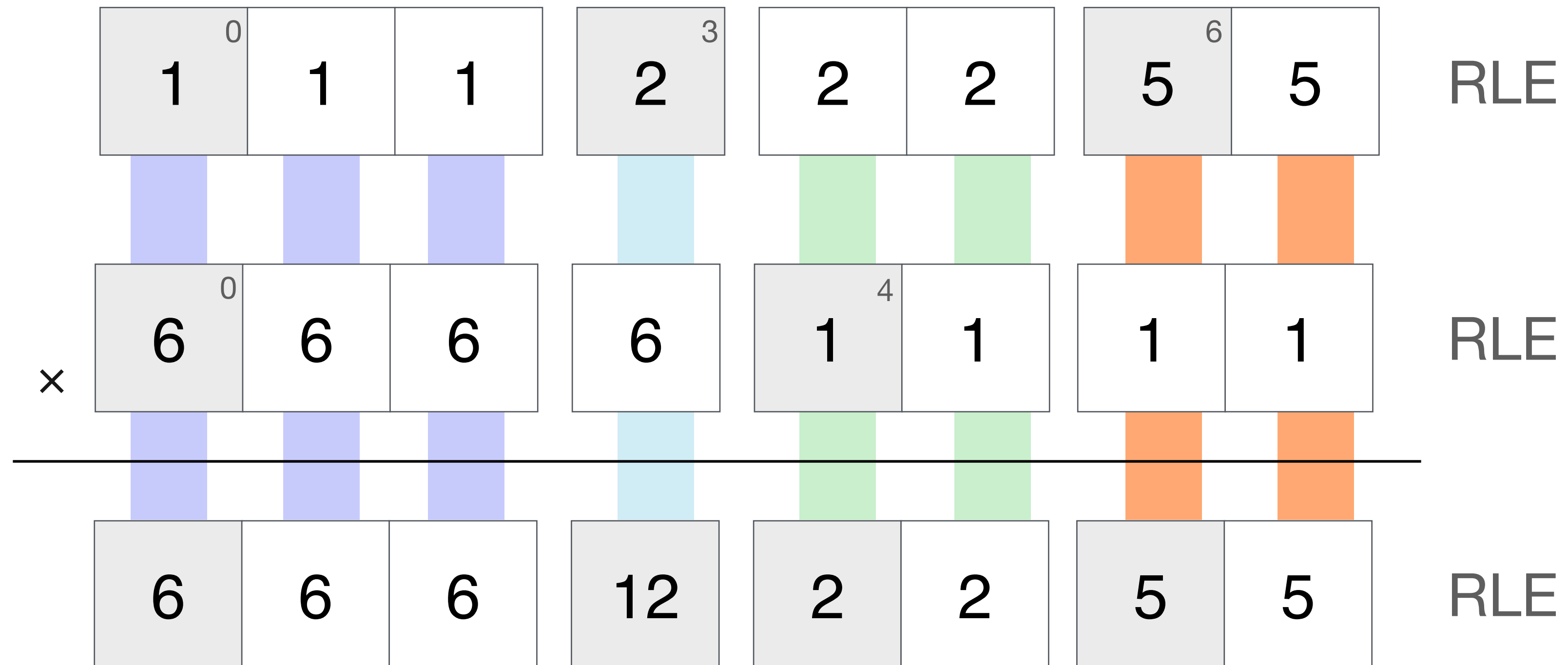
# Optimizing computations on lossless compression formats

## Elementwise computation



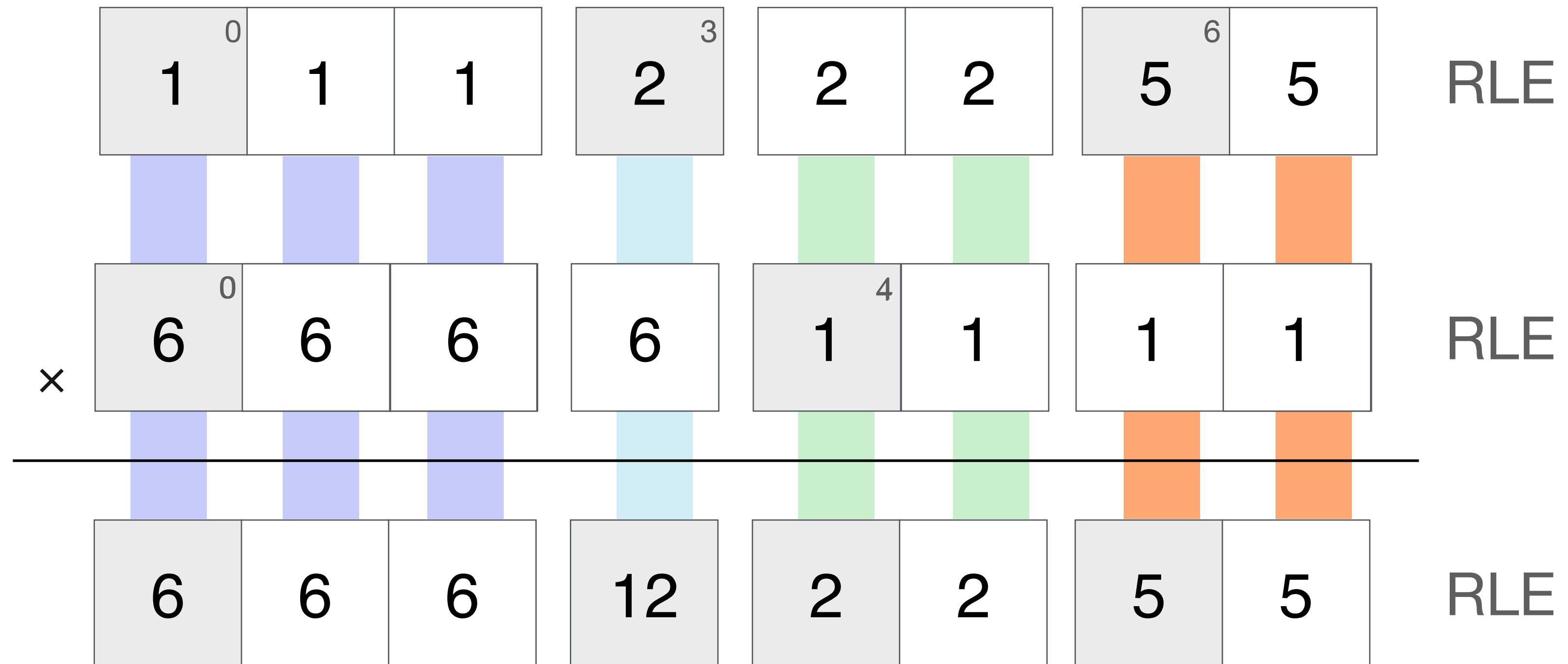
# Optimizing computations on lossless compression formats

## Elementwise computation



# Optimizing computations on lossless compression formats

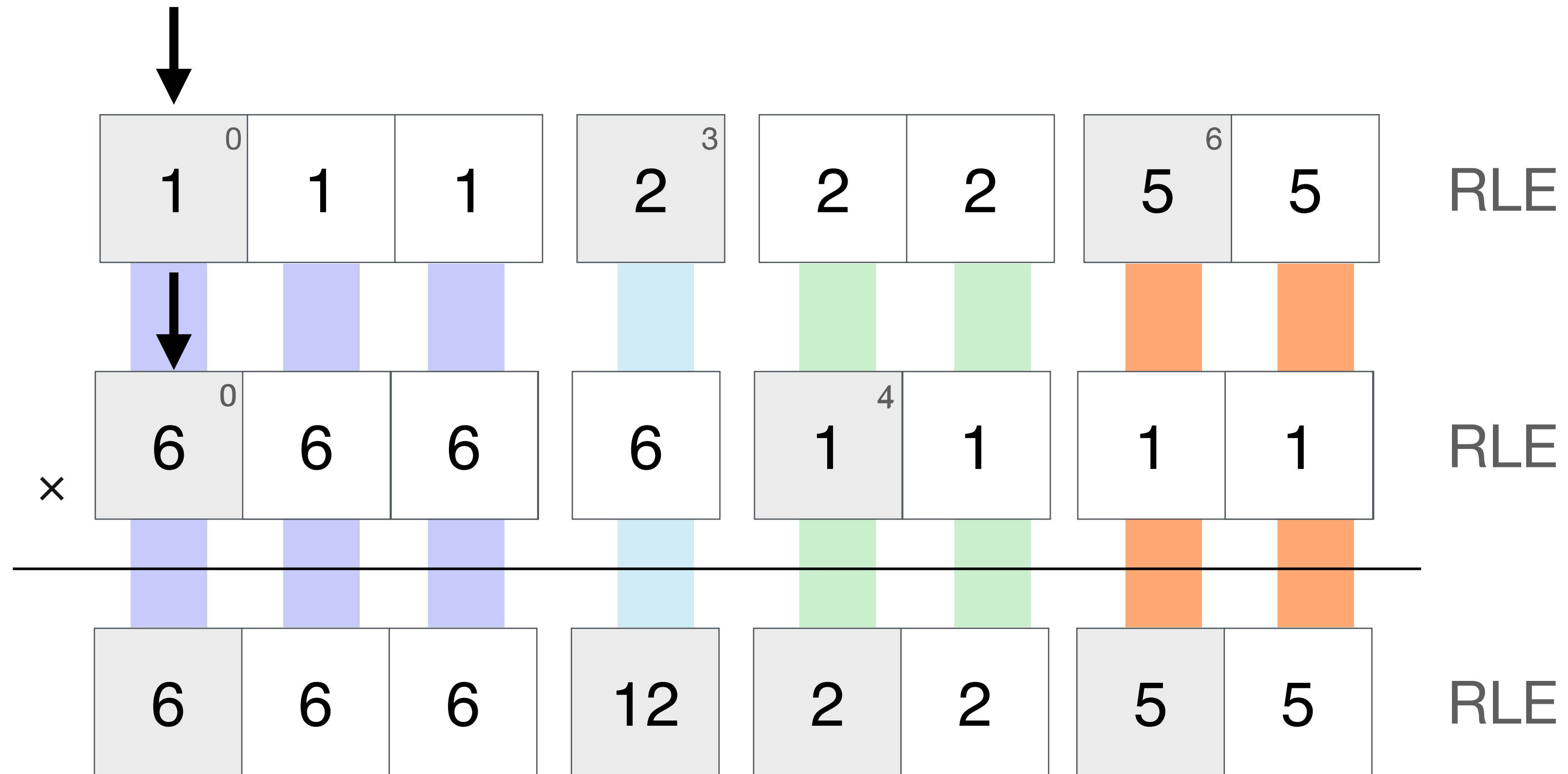
## Elementwise computation





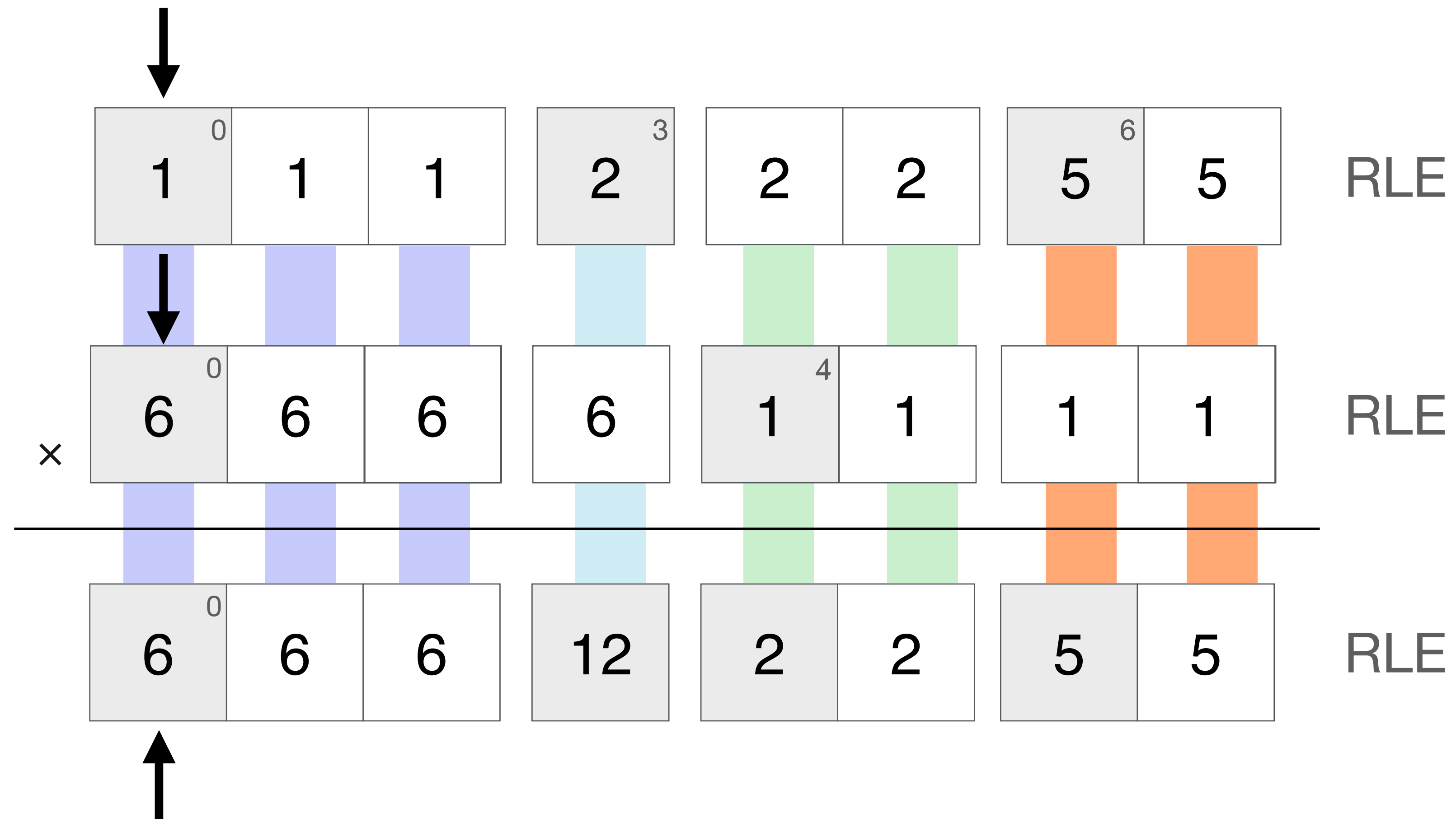
# Optimizing computations on lossless compression formats

## Elementwise computation



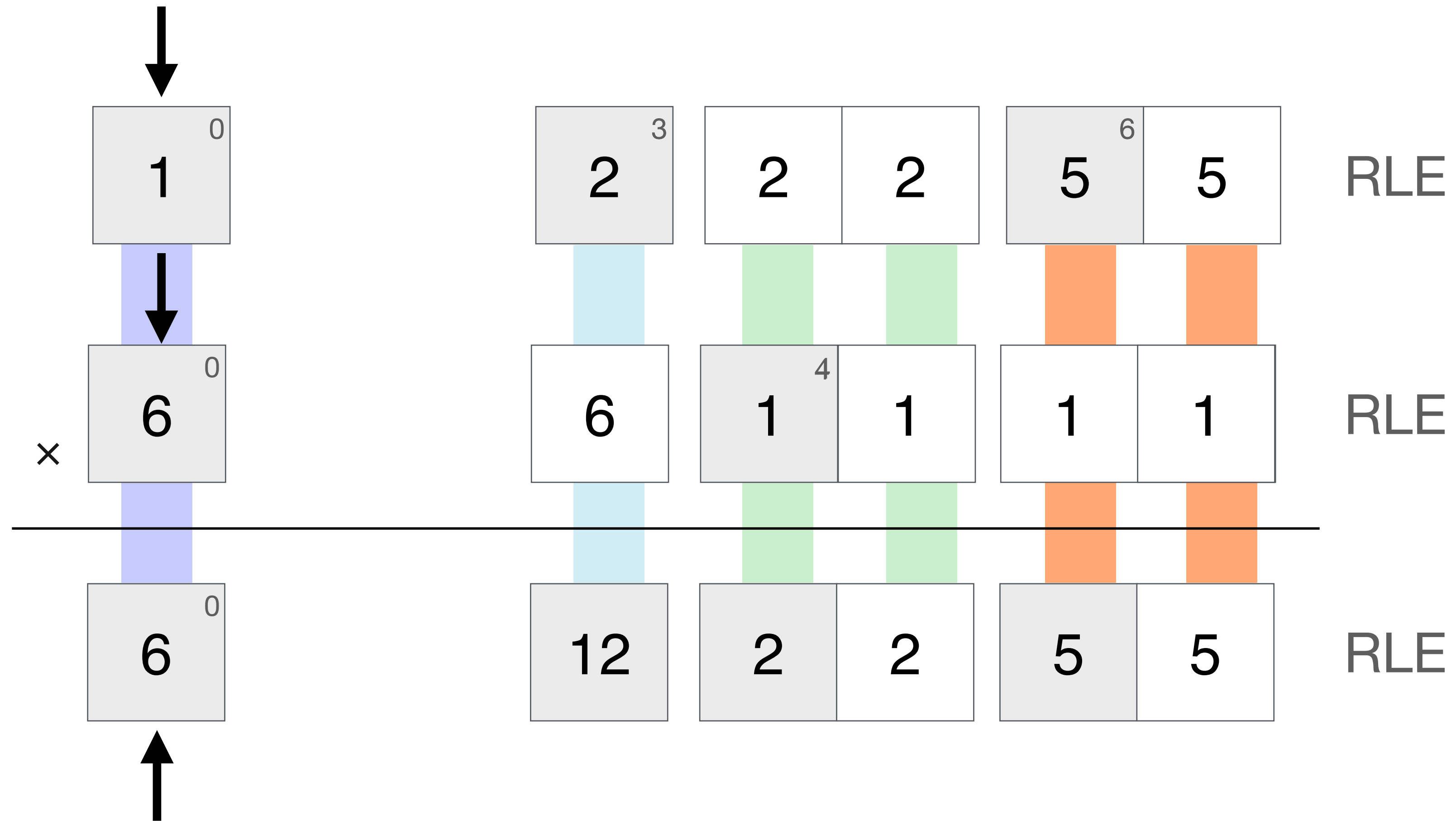
# Optimizing computations on lossless compression formats

## Elementwise computation



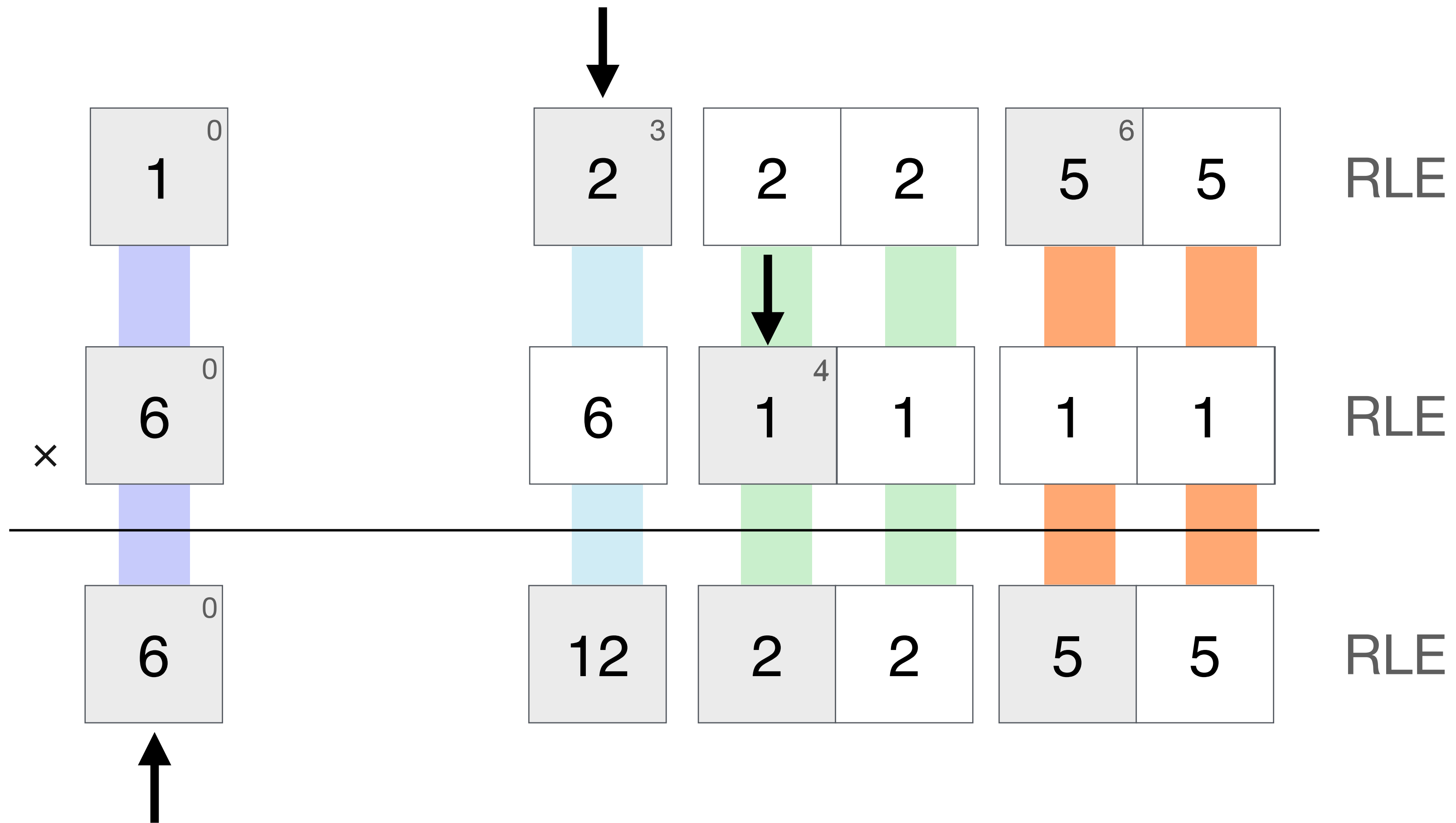
# Optimizing computations on lossless compression formats

## Elementwise computation



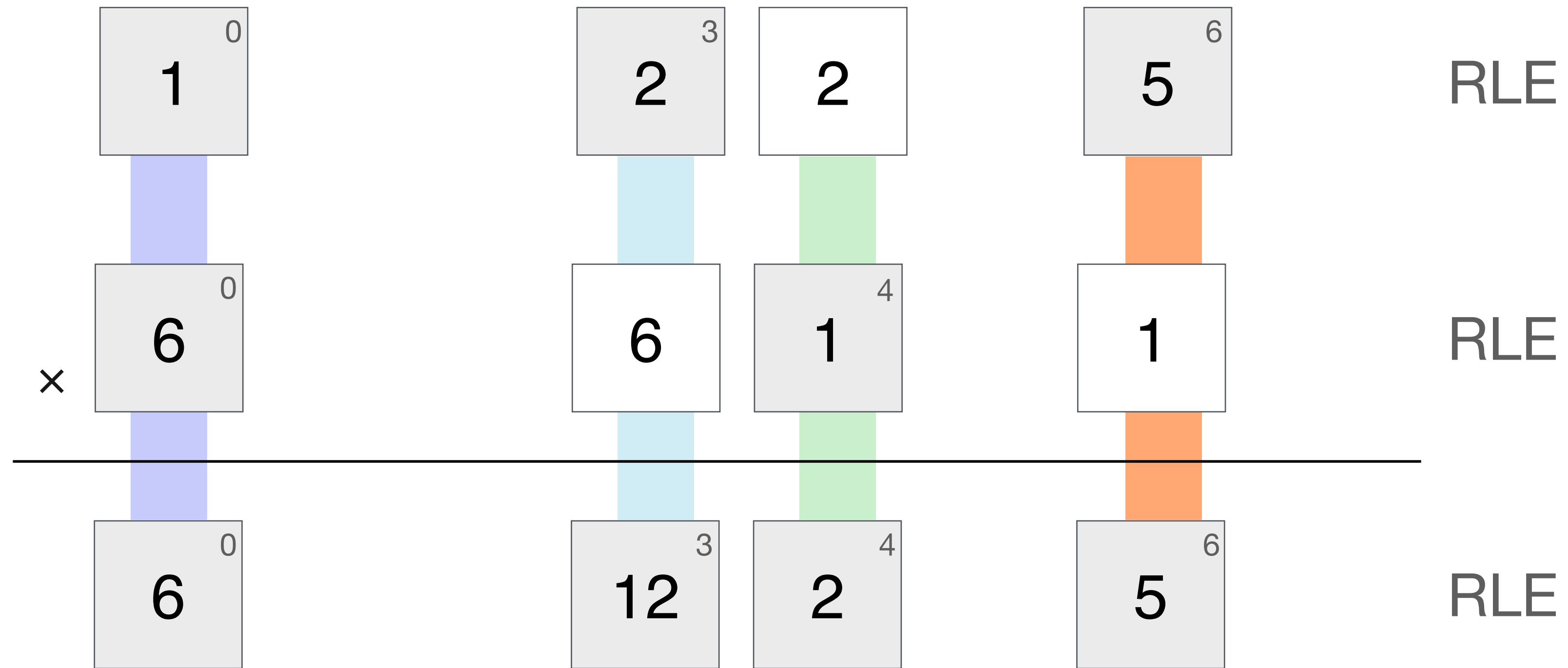
# Optimizing computations on lossless compression formats

## Elementwise computation



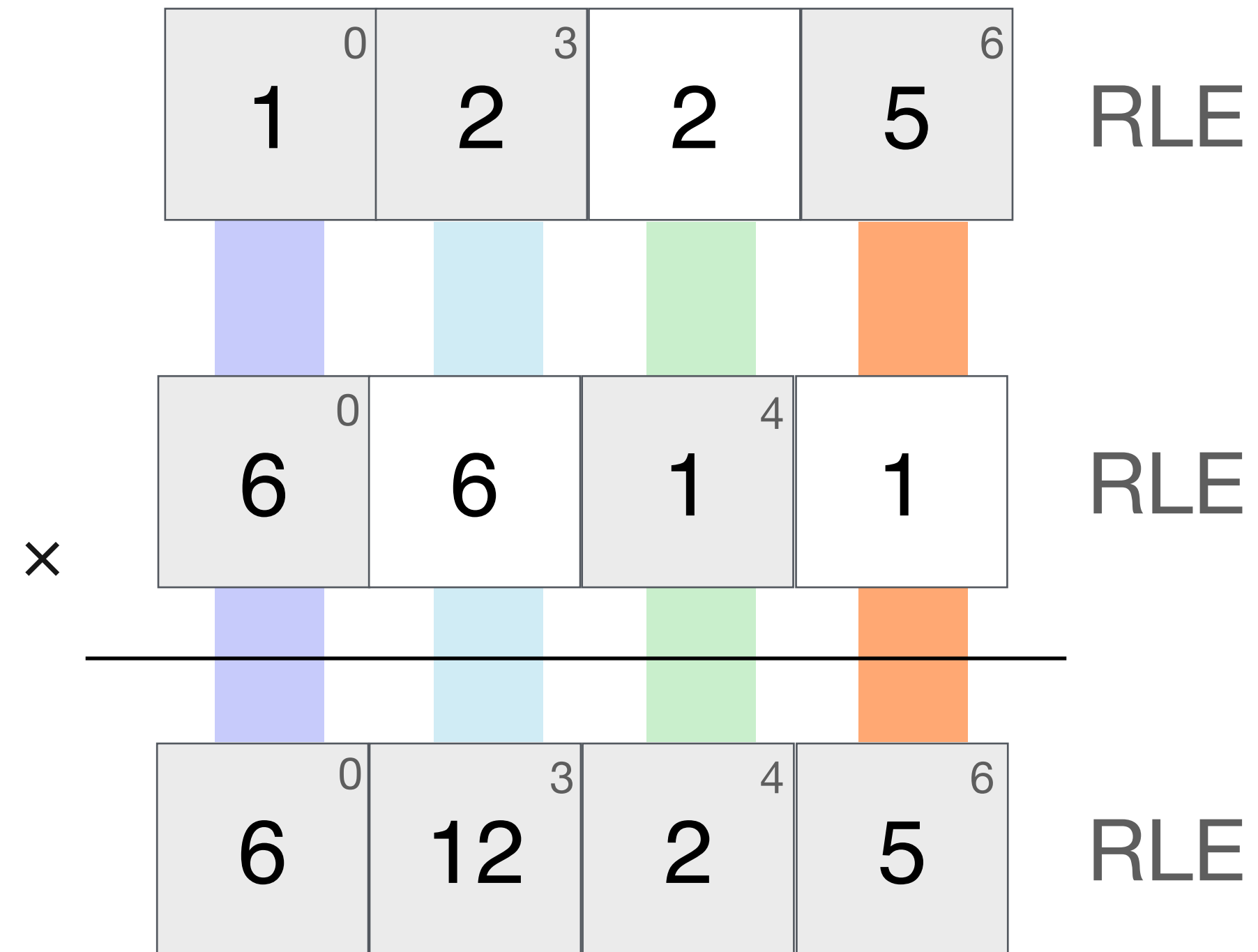
# Optimizing computations on lossless compression formats

## Elementwise computation



# Optimizing computations on lossless compression formats

## Elementwise computation



# Optimizing computations on lossless compression formats

## Reduction

1	1	1	2	2	2	5	5
---	---	---	---	---	---	---	---

RLE

3	7	2	4	8	3	3	1
---	---	---	---	---	---	---	---

Dense



# Optimizing computations on lossless compression formats

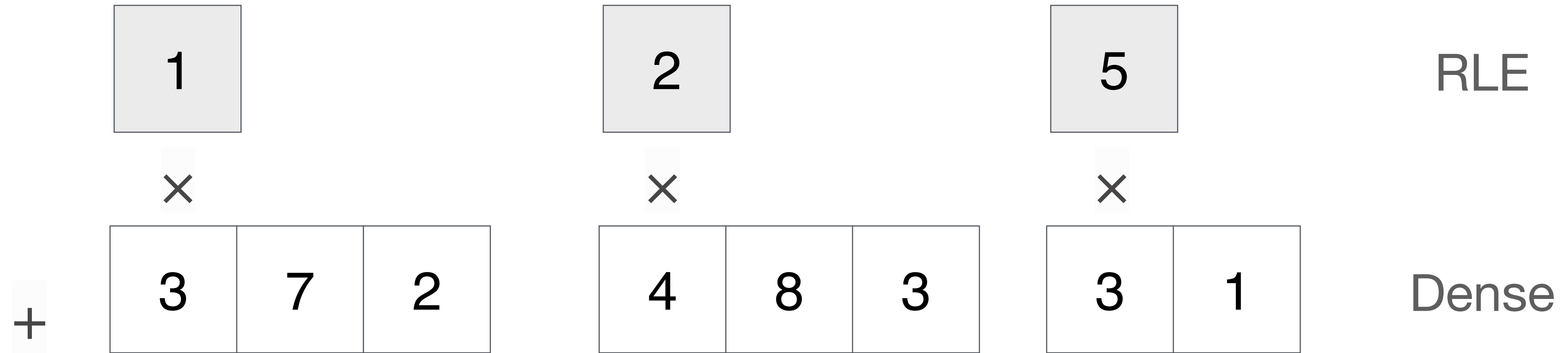
## Reduction





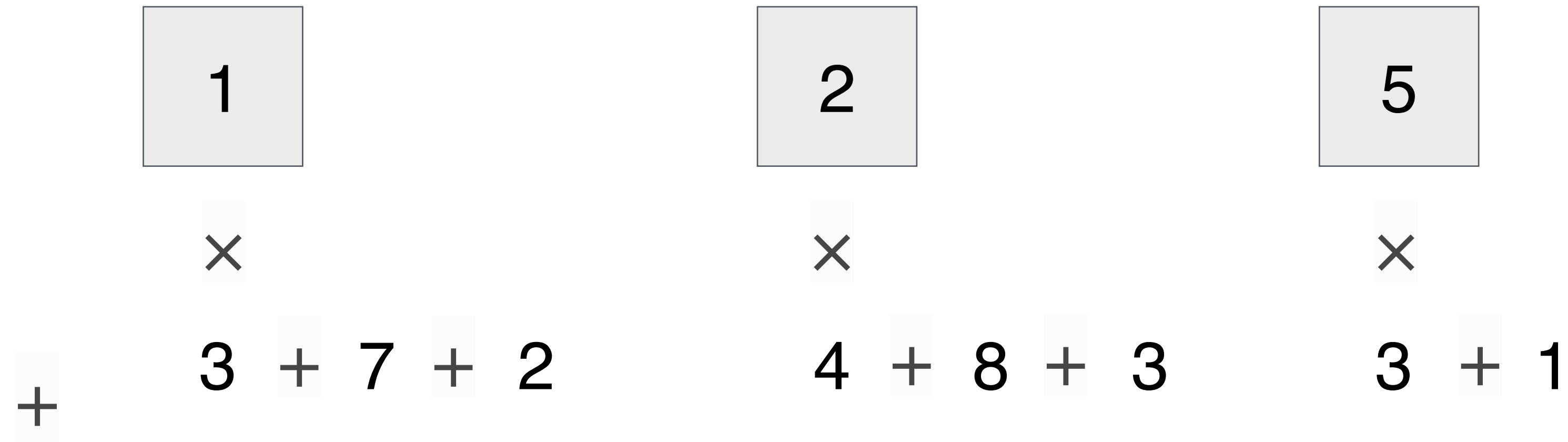
# Optimizing computations on lossless compression formats

## Reduction



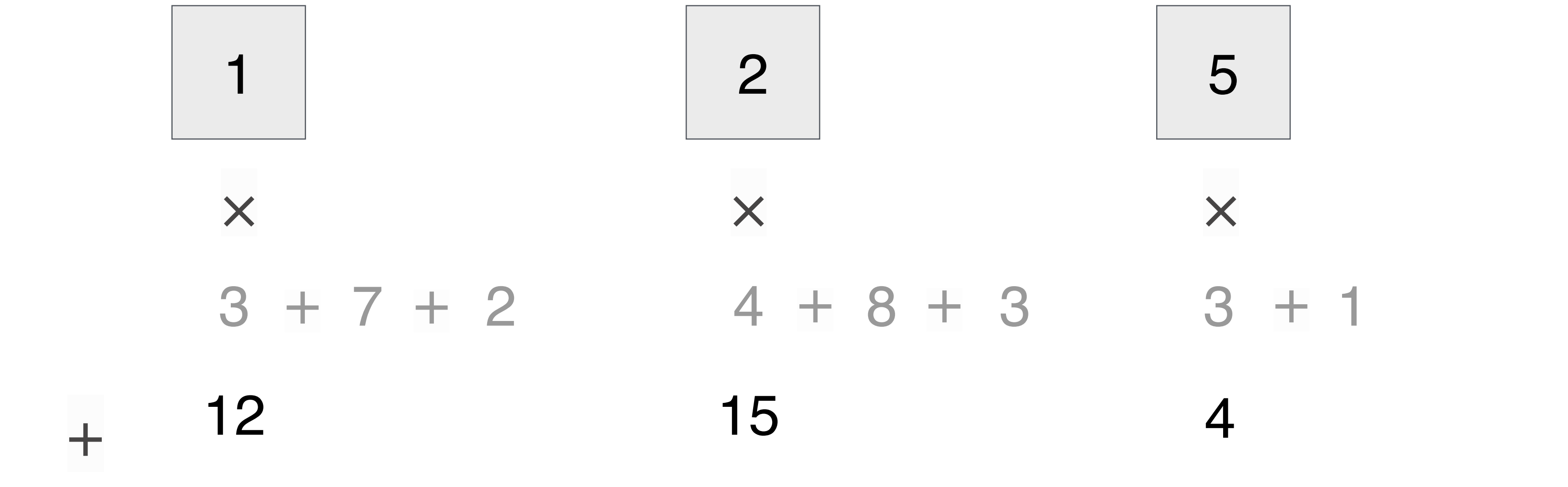
# Optimizing computations on lossless compression formats

## Reduction



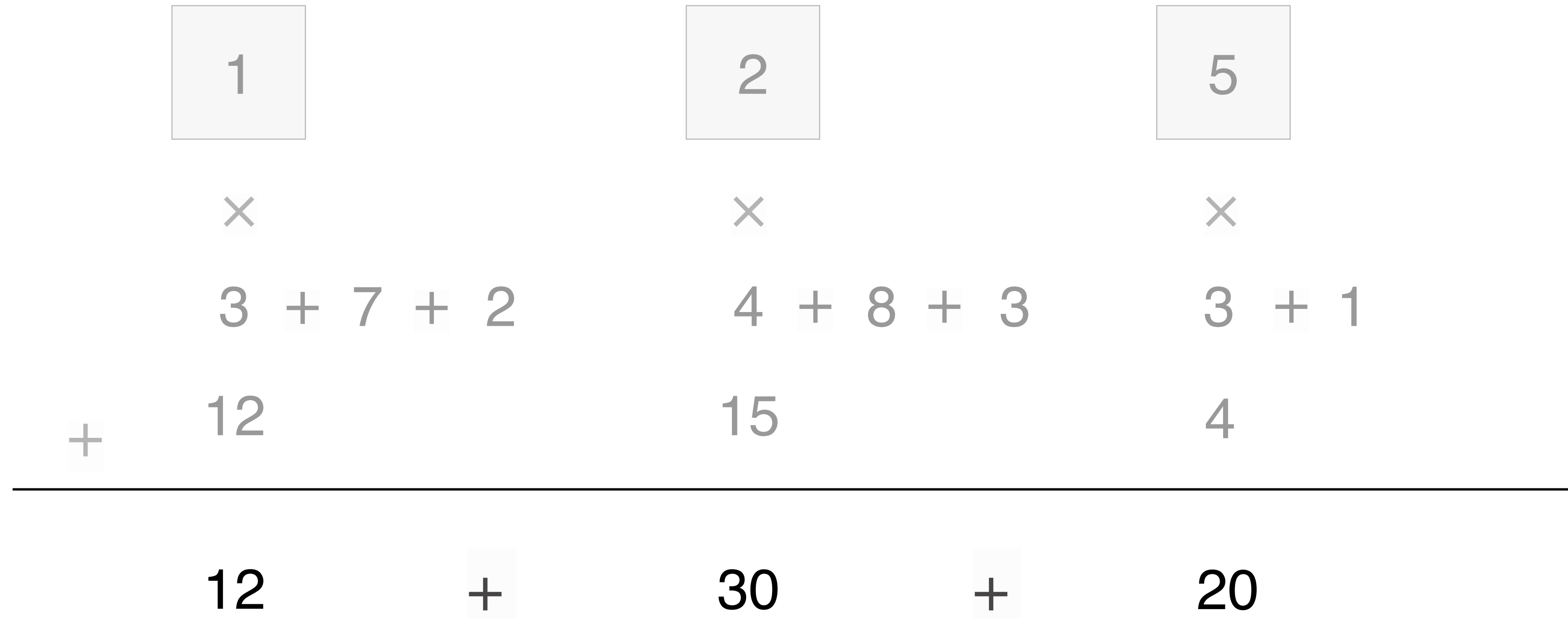
# Optimizing computations on lossless compression formats

## Reduction



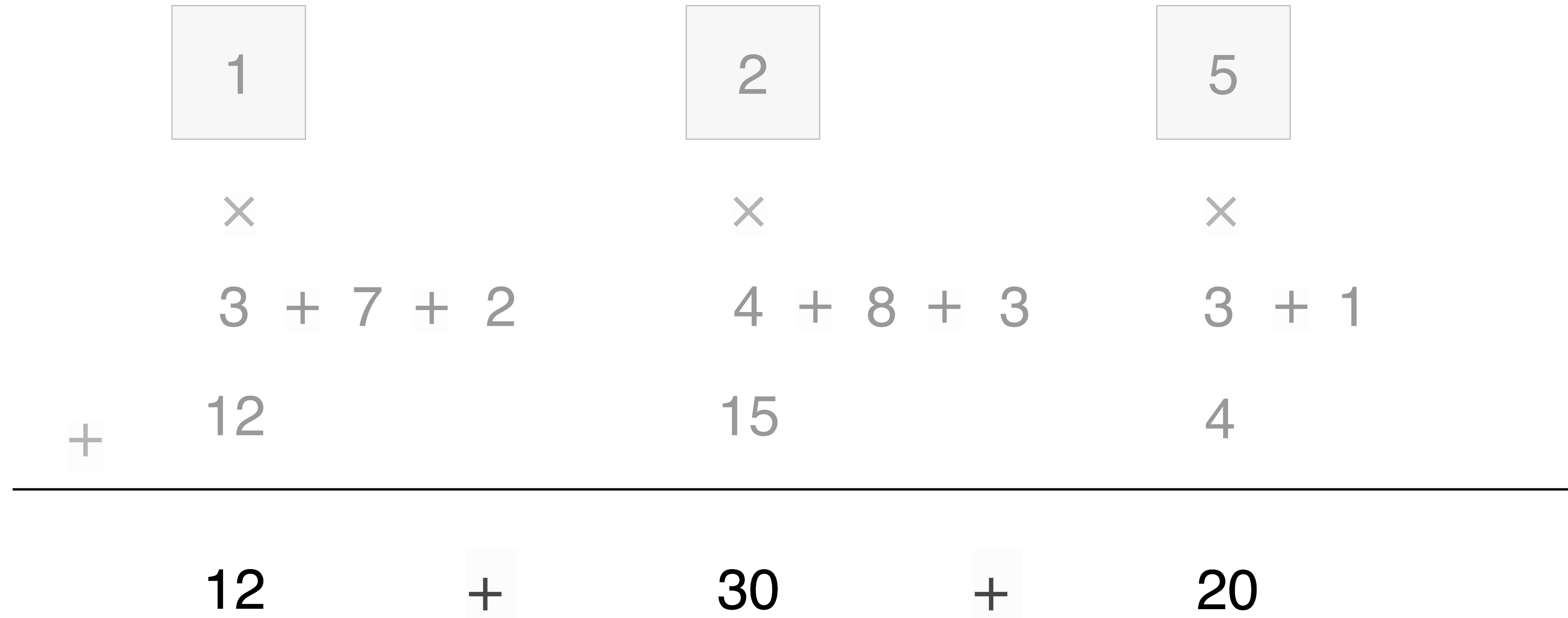
# Optimizing computations on lossless compression formats

## Reduction



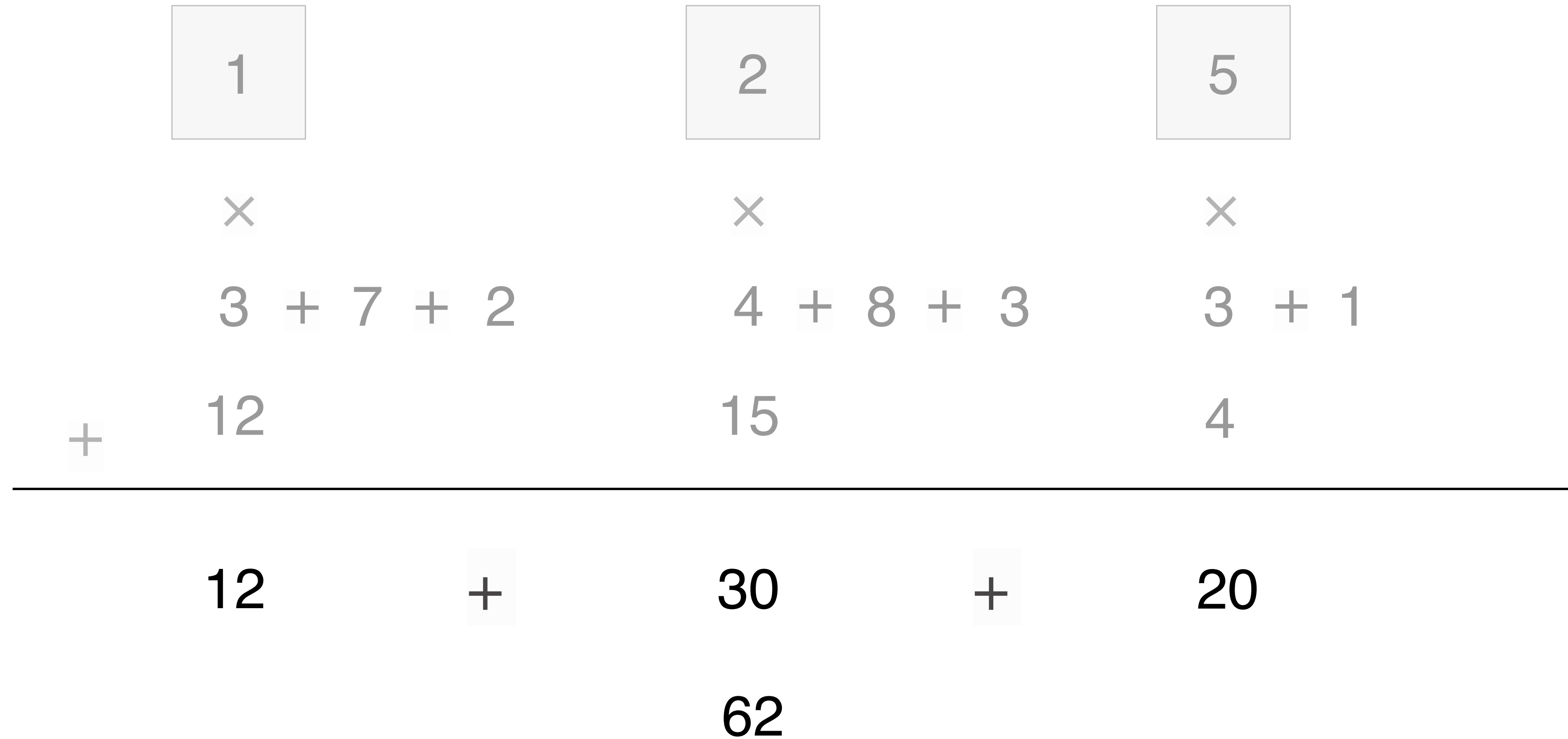
# Optimizing computations on lossless compression formats

## Reduction



# Optimizing computations on lossless compression formats

## Reduction

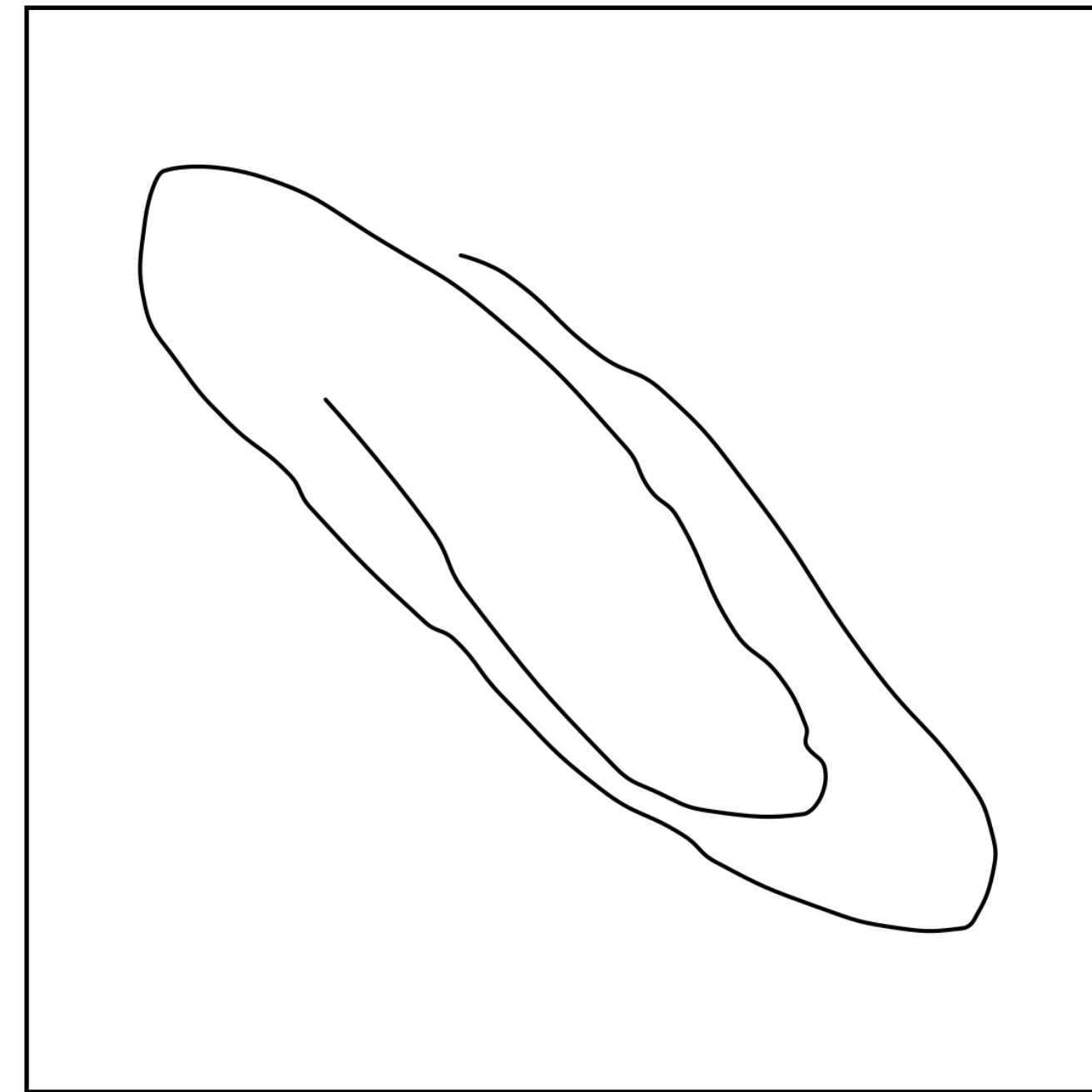
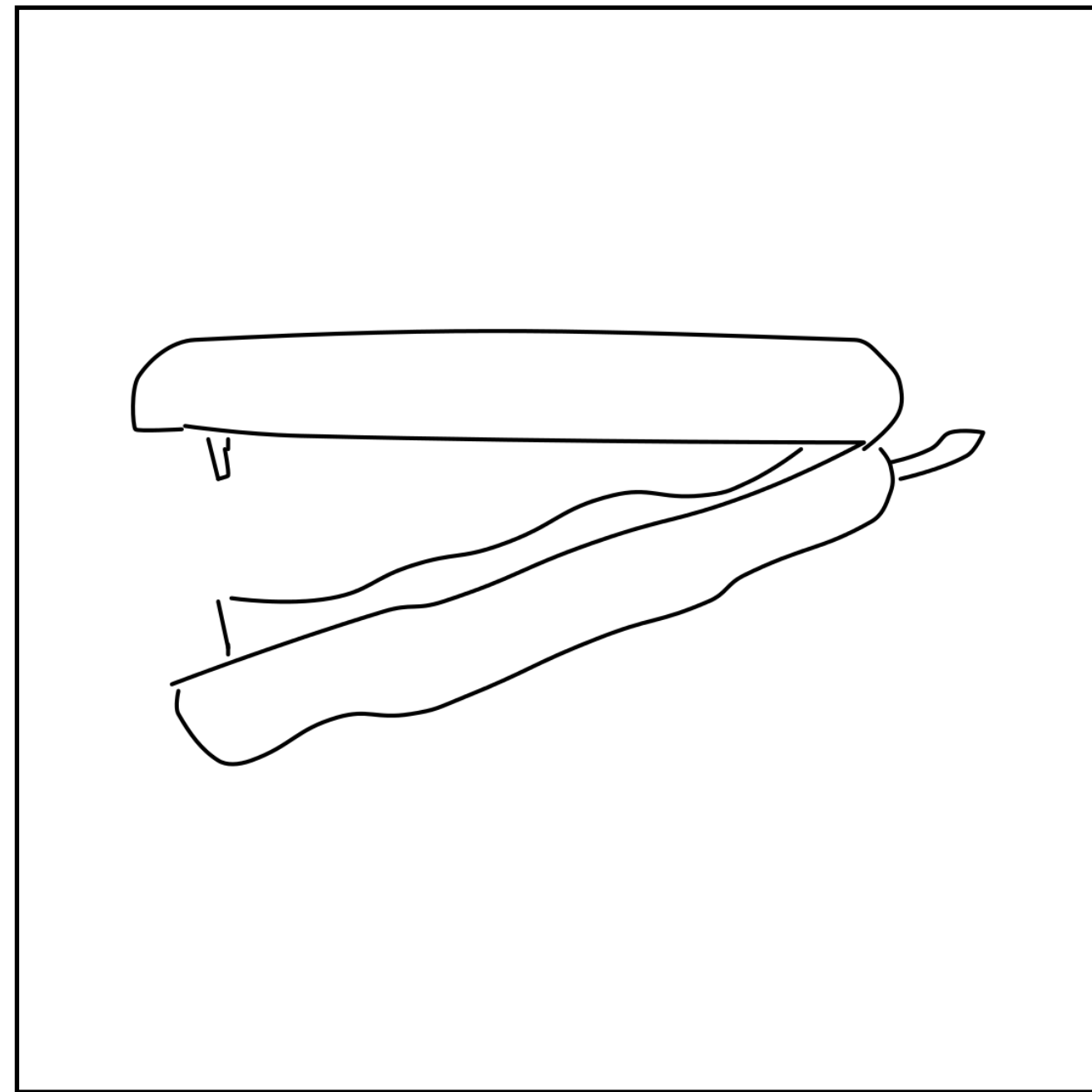


# Code Generation

- Implemented as an extension to TACO
- Integrated support for our new abstractions into the format language
- Added compiler support for generating efficient code using our abstractions
- Added our new level formats

# Results

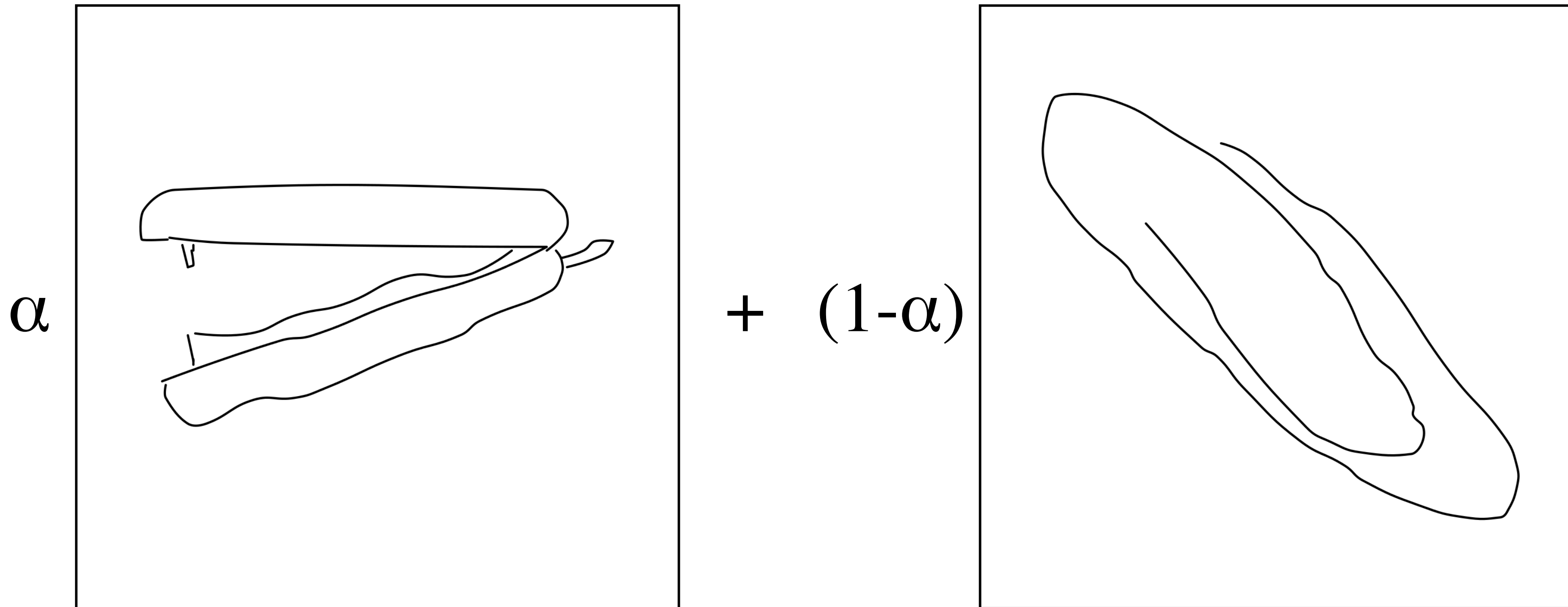
$$A_{ij} = \alpha B_{ij} + (1 - \alpha)C_{ij}$$





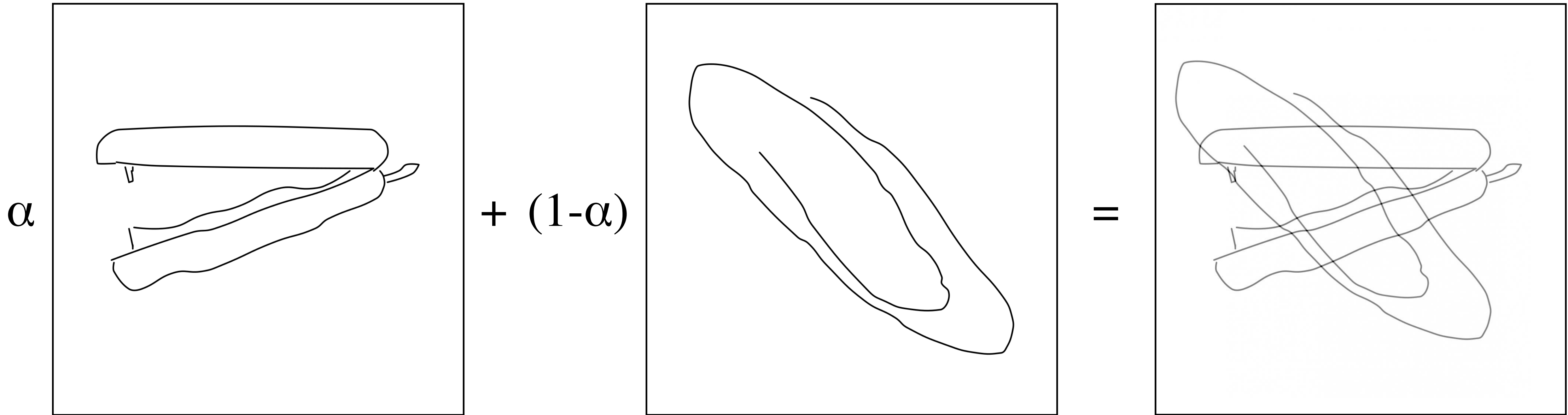
# Results

$$A_{ij} = \alpha B_{ij} + (1 - \alpha)C_{ij}$$



# Results

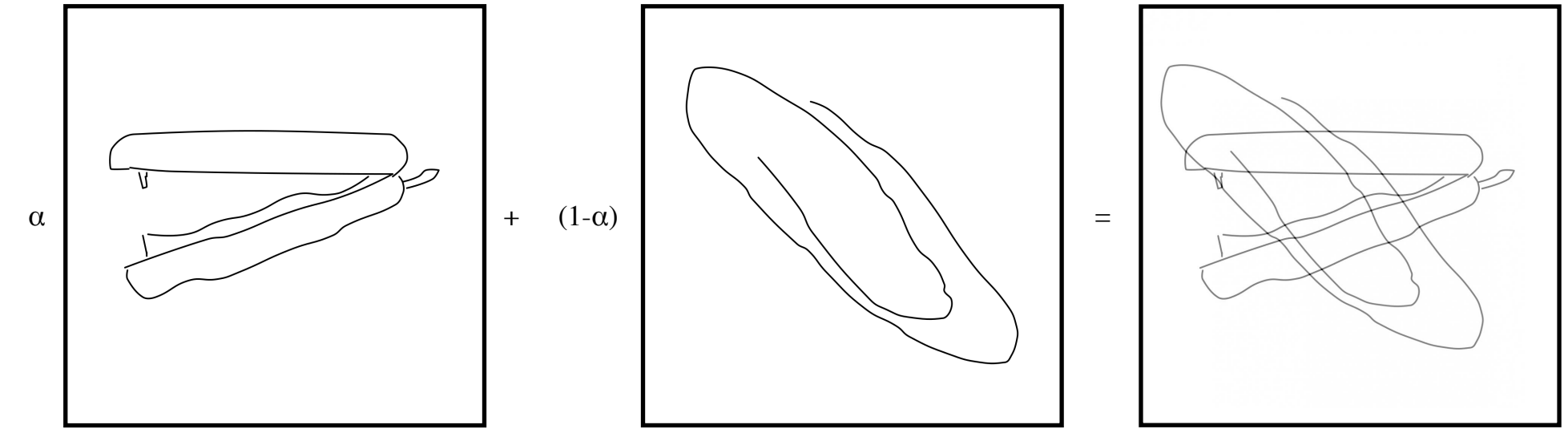
$$A_{ij} = \alpha B_{ij} + (1 - \alpha)C_{ij}$$



# Results

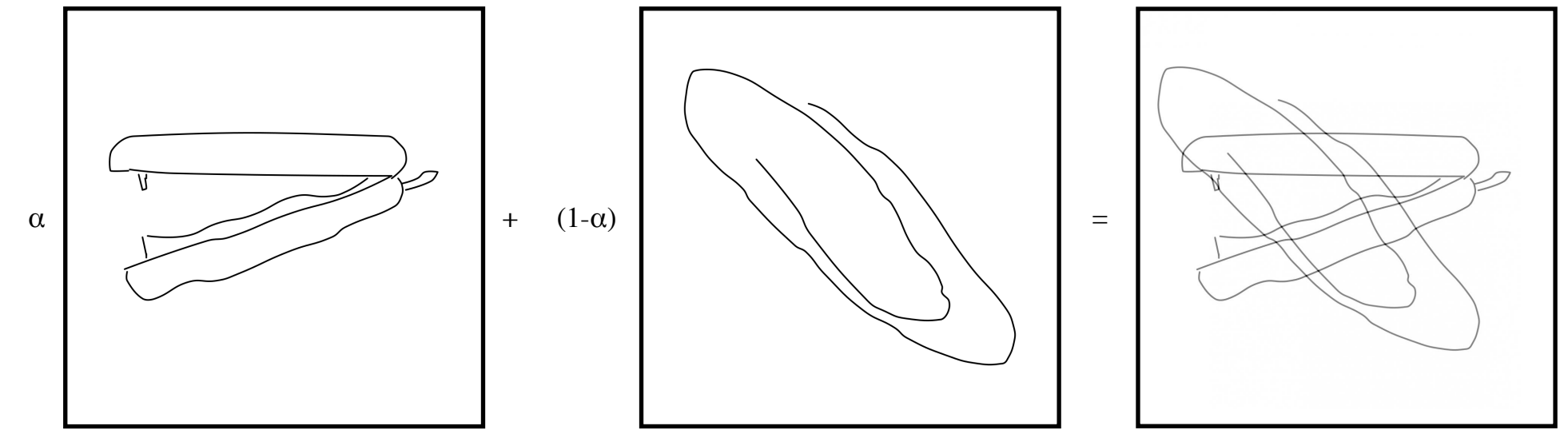
---

$$A_{ij} = \alpha B_{ij} + (1 - \alpha)C_{ij}$$

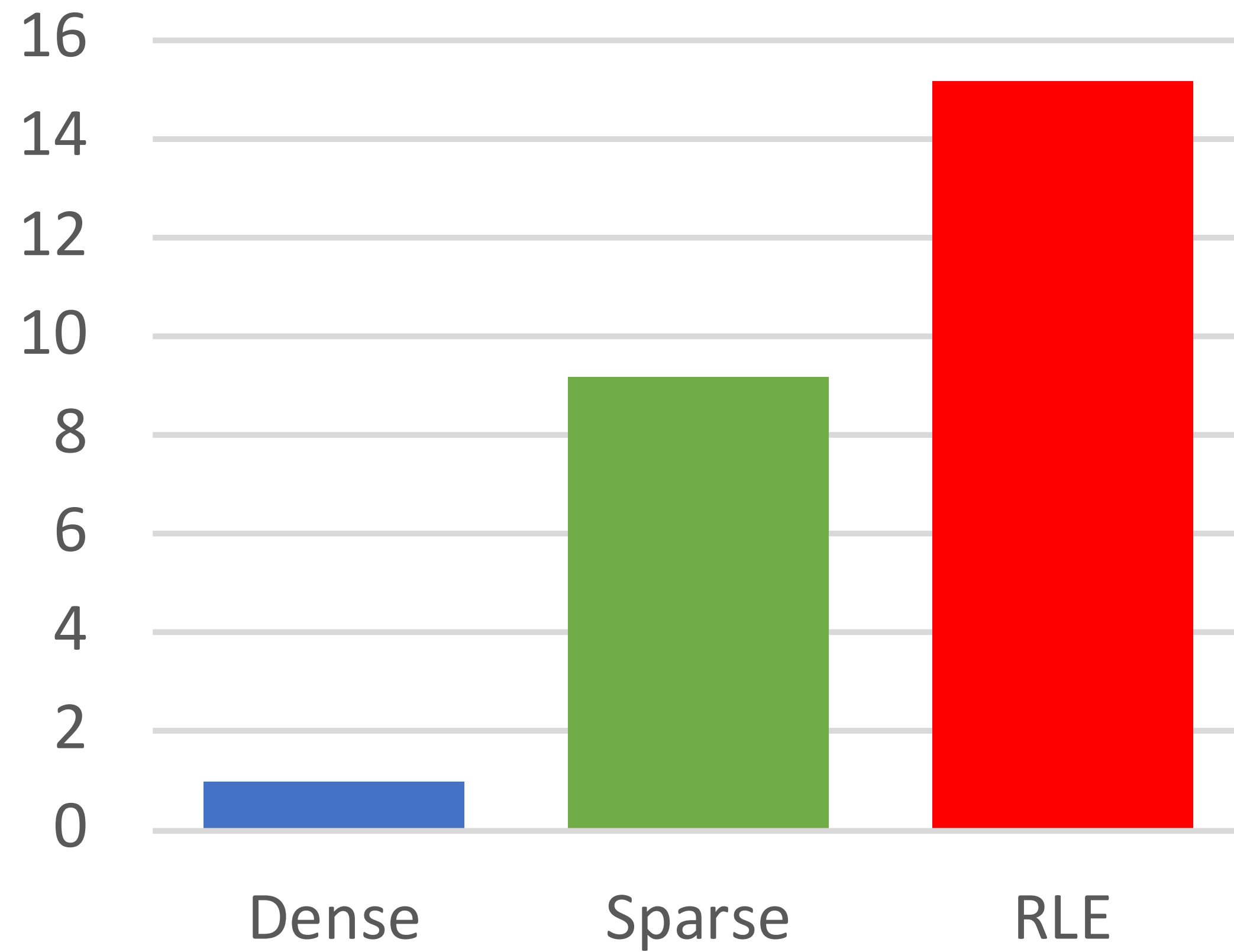


# Results

$$A_{ij} = \alpha B_{ij} + (1 - \alpha)C_{ij}$$

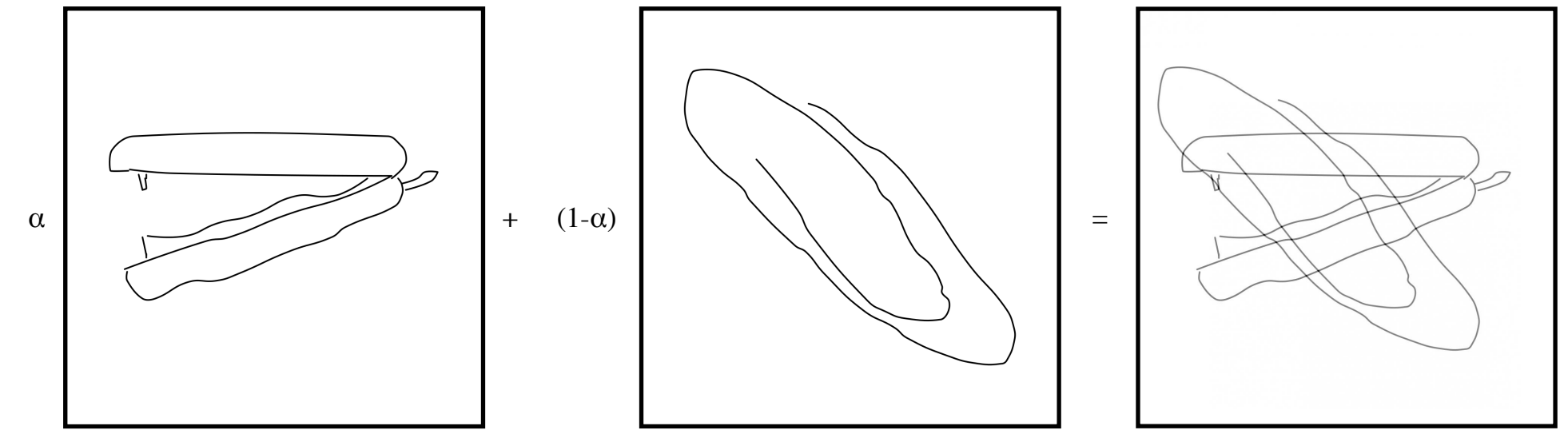


Data Compression Ratio

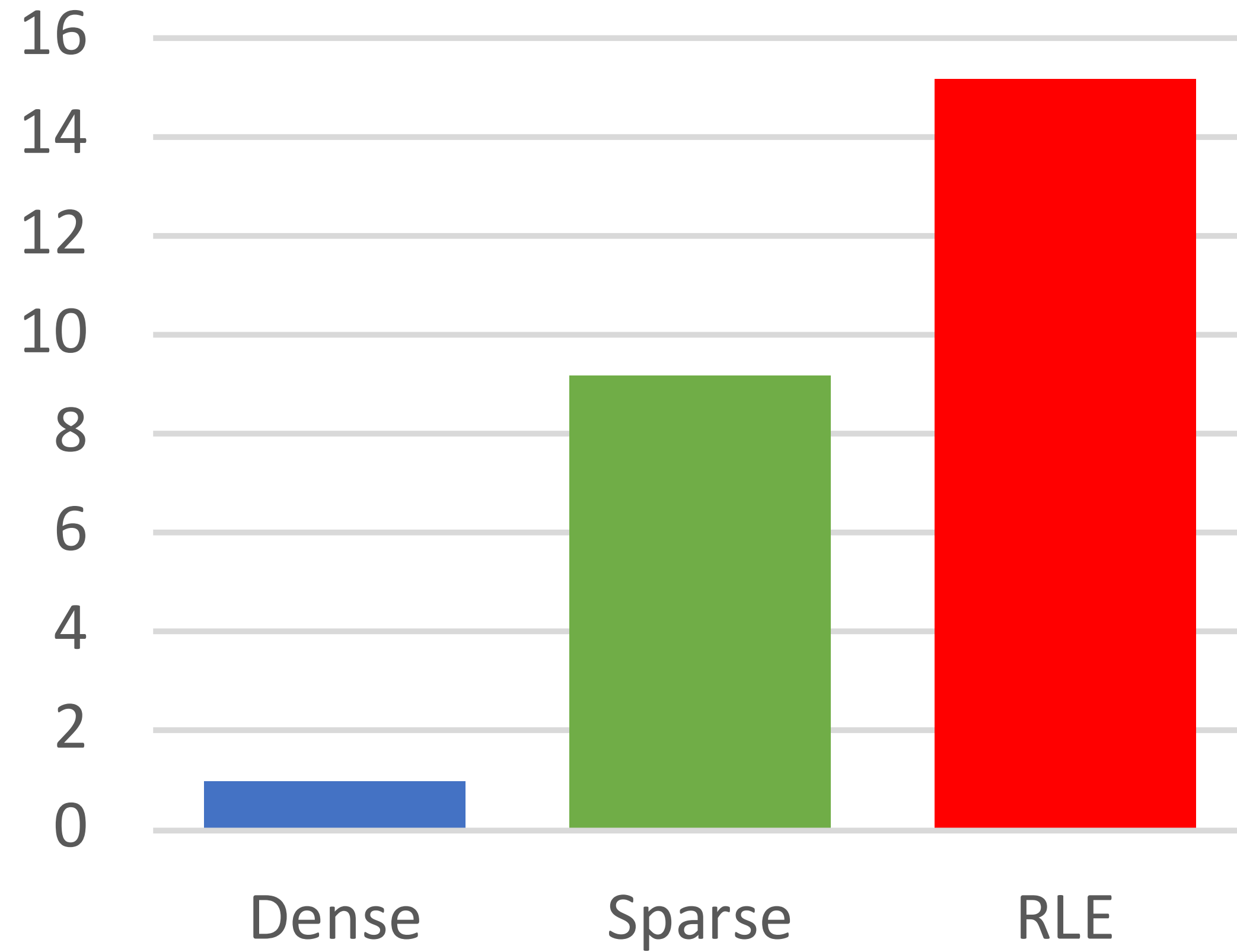


# Results

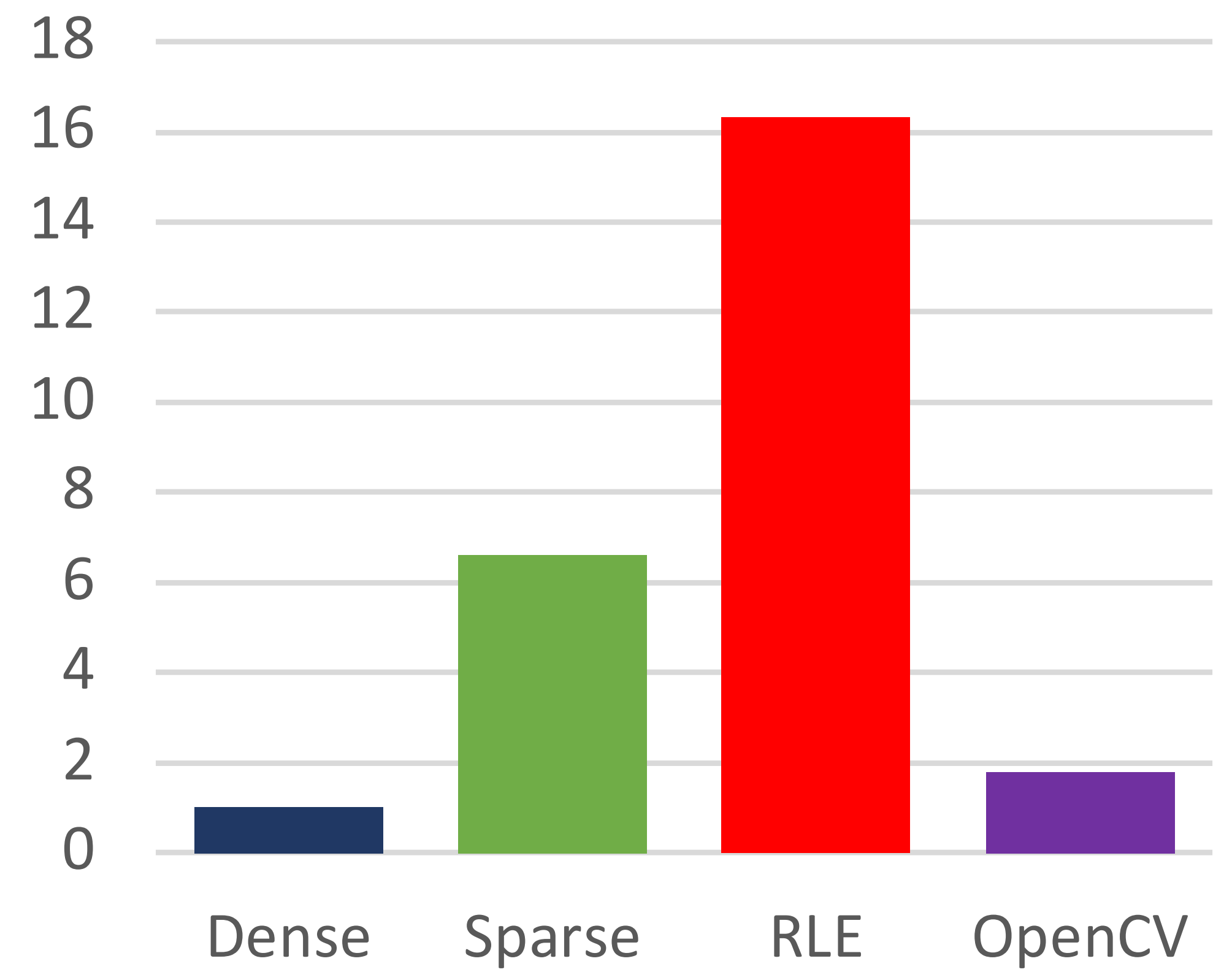
$$A_{ij} = \alpha B_{ij} + (1 - \alpha)C_{ij}$$



Data Compression Ratio



Geomean Speedup over Dense



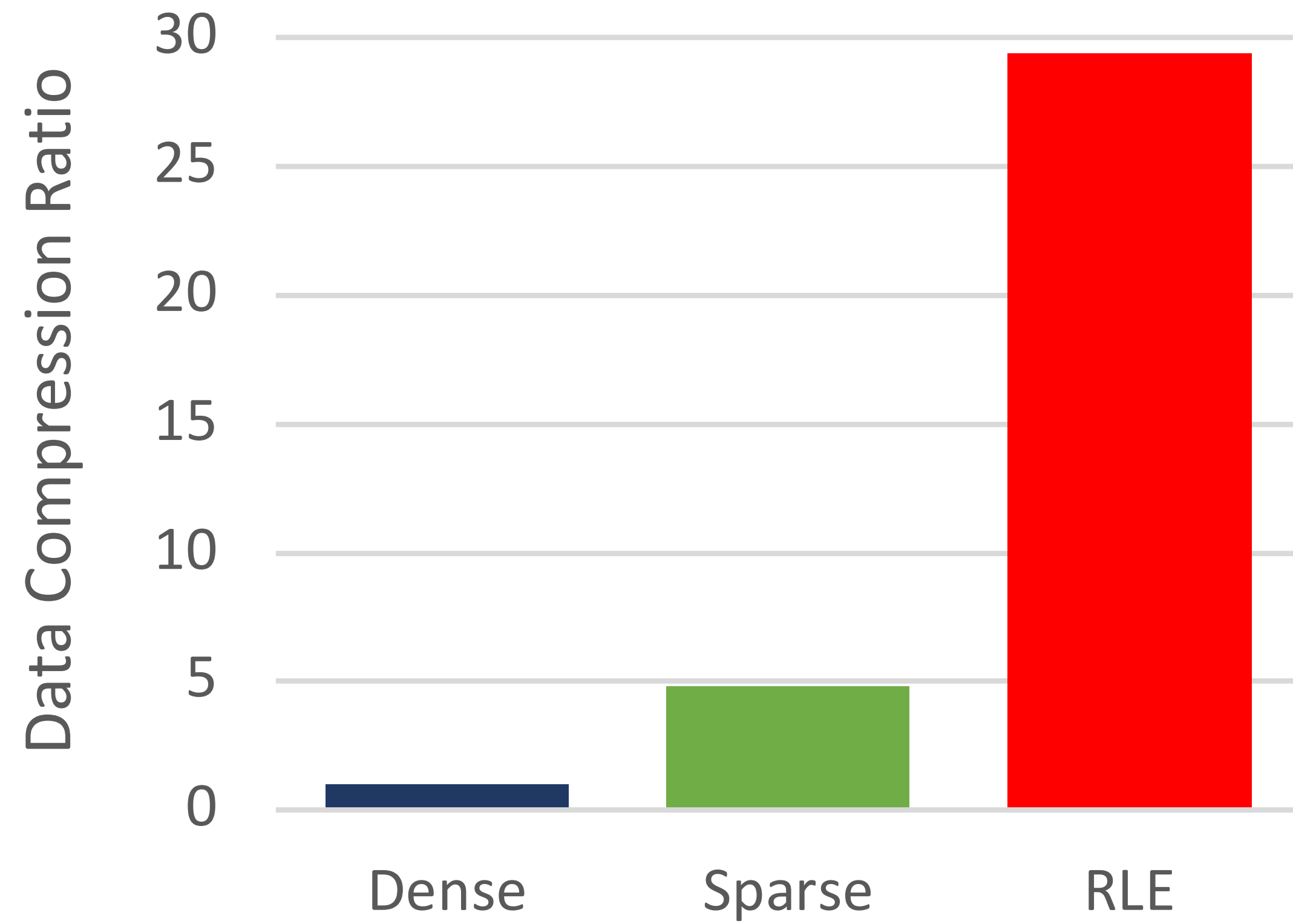
# Results

$$Out_{ij} = (A_{ij} \wedge ROI_{ij}) \oplus (B_{ij} \wedge ROI_{ij})$$



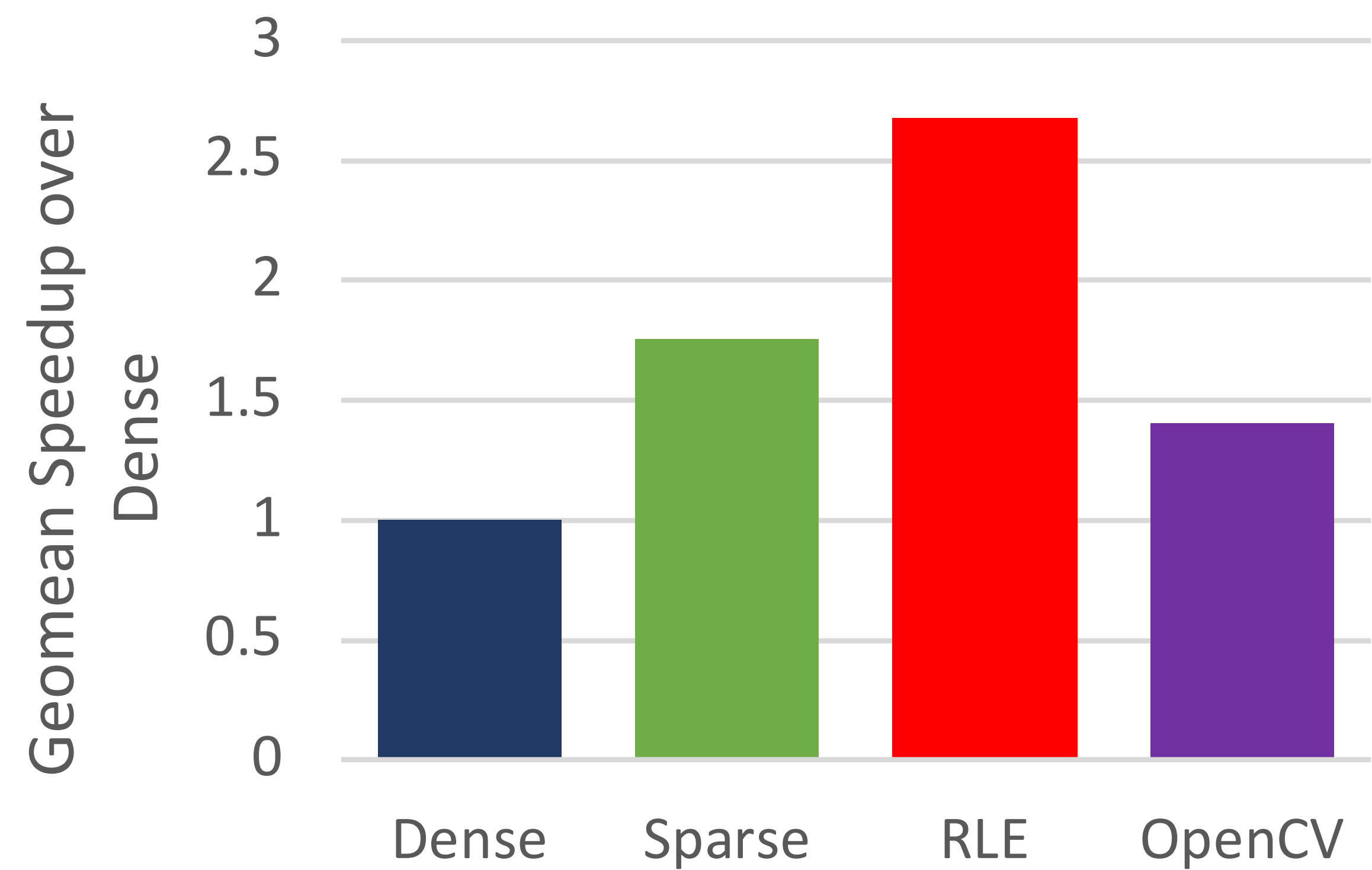
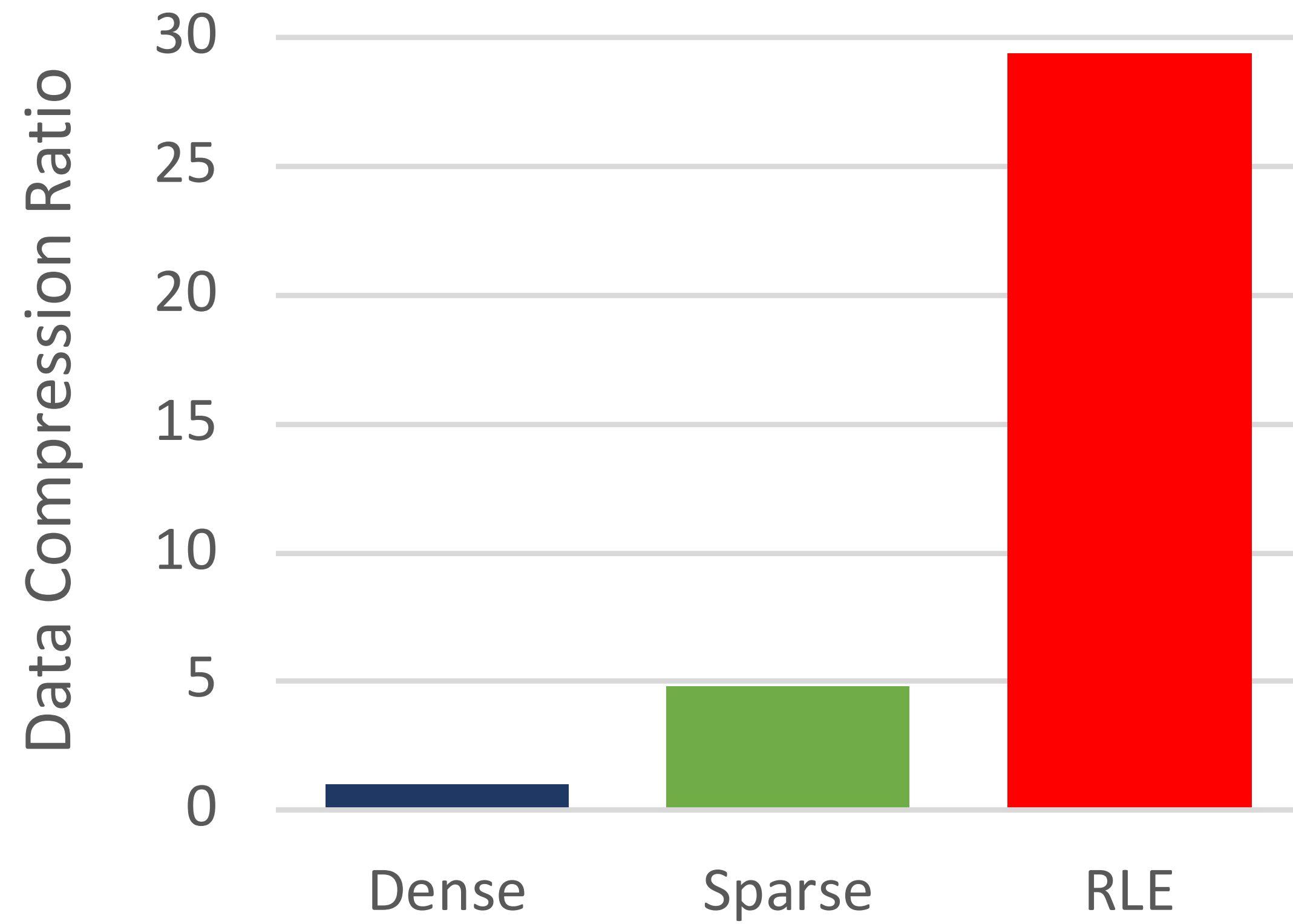
# Results

$$Out_{ij} = (A_{ij} \wedge ROI_{ij}) \oplus (B_{ij} \wedge ROI_{ij})$$



# Results

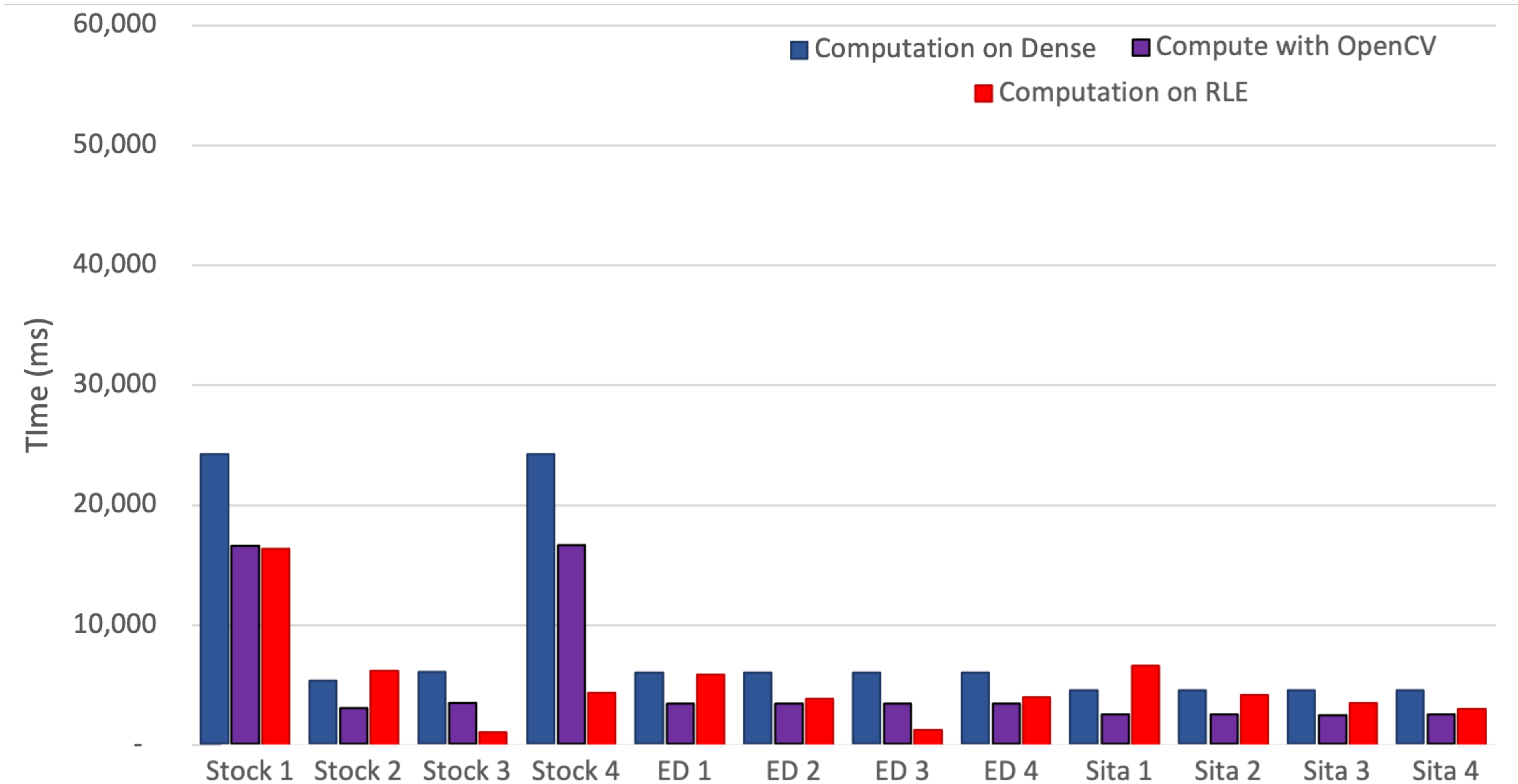
$$Out_{ij} = (A_{ij} \wedge ROI_{ij}) \oplus (B_{ij} \wedge ROI_{ij})$$





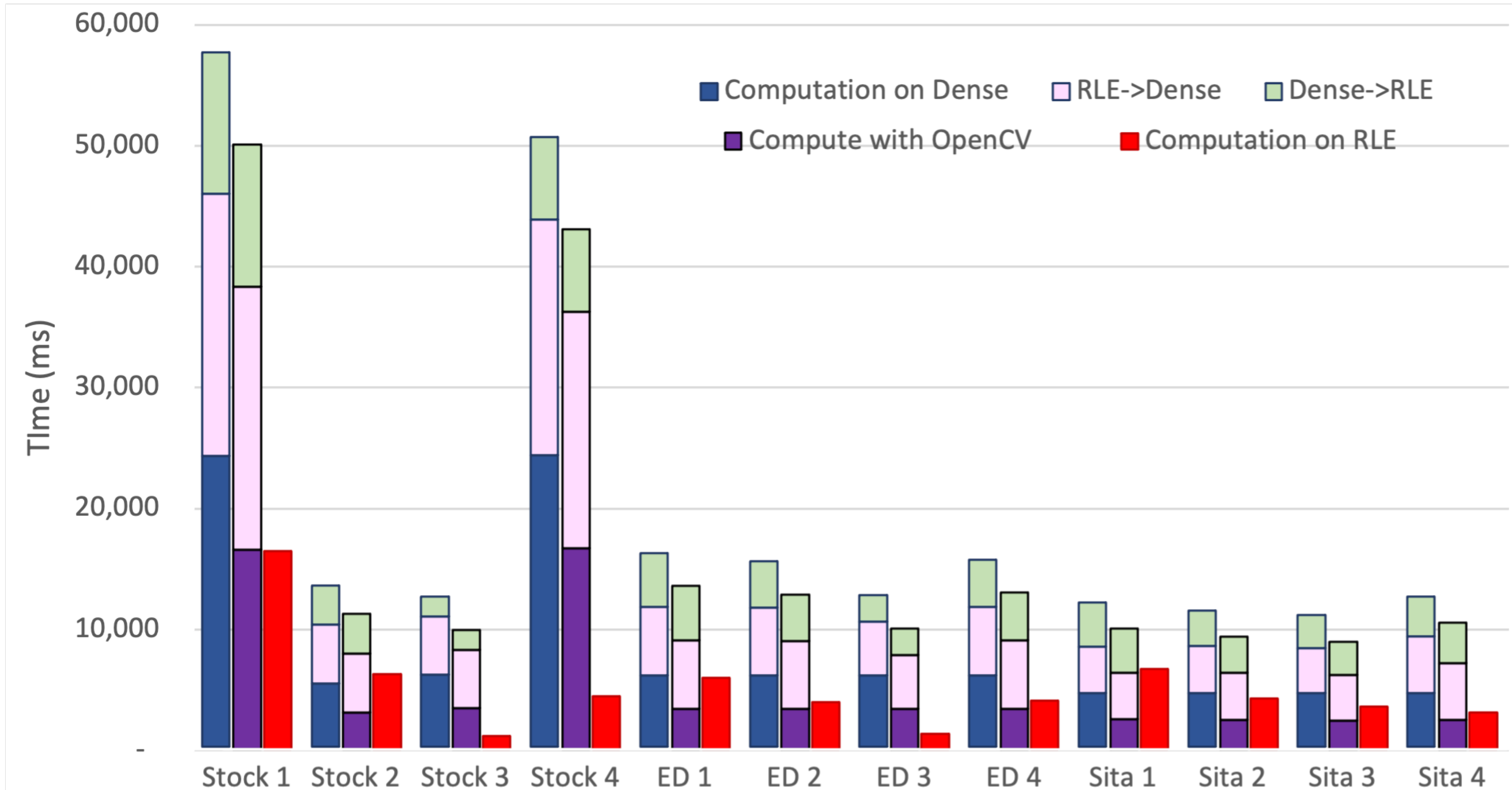
# Results

$$Out_{ijc} = (F_{ijc} * M_{ij}) + (S_{ij} * !M_{ij})$$



# Results

$$Out_{ijc} = (F_{ijc} * M_{ij}) + (S_{ij} * !M_{ij})$$



# In conclusion

- We can automatically generate kernels that compute with both lossless compressed and sparse formats



<http://tensor-compiler.org/>

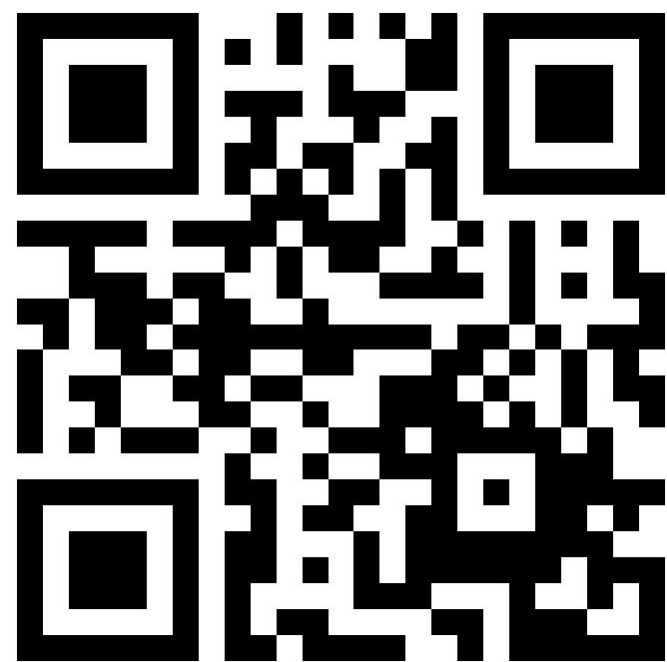


This Work Supported By:



# In conclusion

- We can automatically generate kernels that compute with both lossless compressed and sparse formats
- Our new abstractions enable implementation of many lossless compression formats



<http://tensor-compiler.org/>

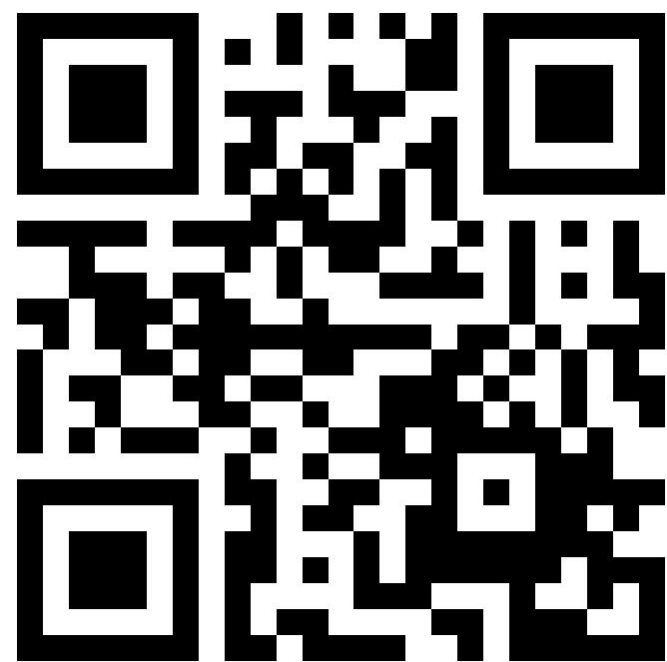


This Work Supported By:



# In conclusion

- We can automatically generate kernels that compute with both lossless compressed and sparse formats
- Our new abstractions enable implementation of many lossless compression formats
- It is possible to get both the best compression and performance



<http://tensor-compiler.org/>



This Work Supported By:

