

Fast Matrix Multiplication

Ran Duan¹.

Hongxun Wu².

Renfei Zhou¹

¹ IIS, Tsinghua University

² UC Berkeley

$$\begin{matrix} & A & & B & & \\ \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} & \times & \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} & = & \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix} \end{matrix}$$

$$1 \times 6 + 2 \times 8 = 22$$

$$1 \times 5 + 2 \times 7 = 19$$

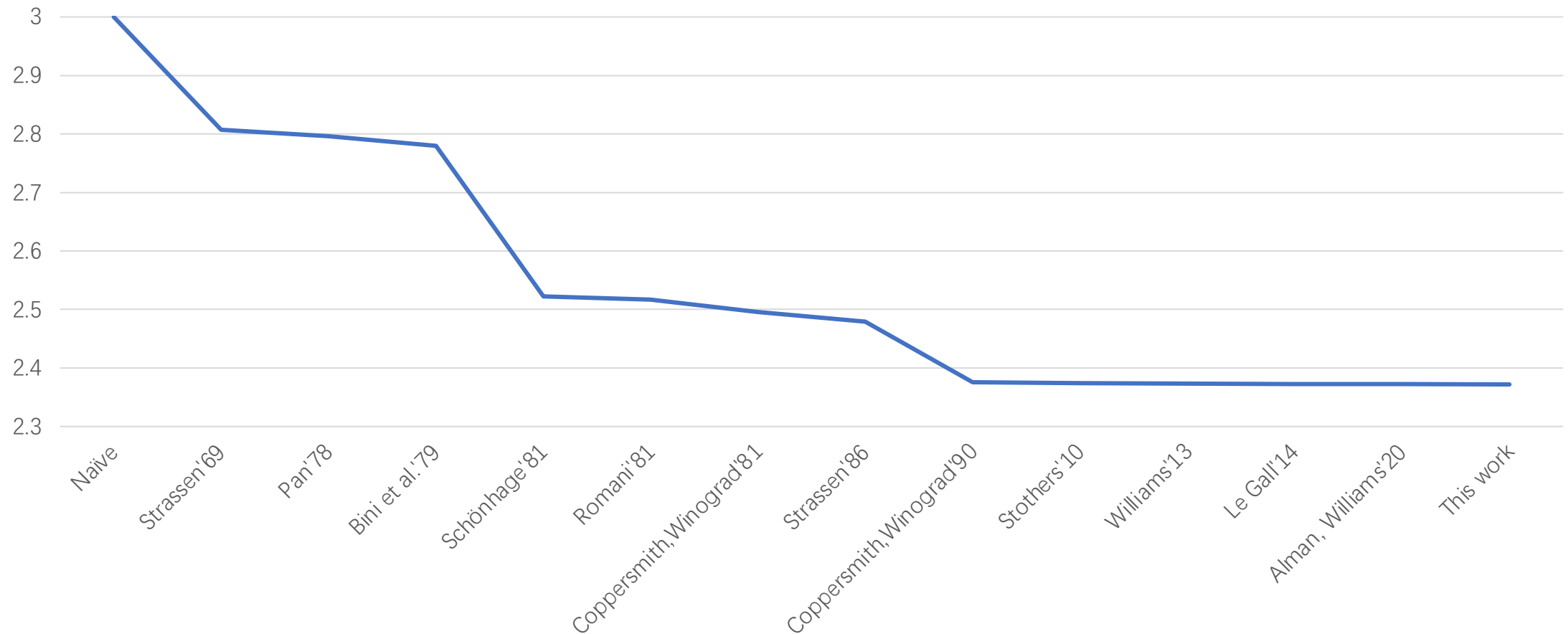
$$3 \times 5 + 4 \times 7 = 43$$

$$3 \times 6 + 4 \times 8 = 50$$

8 multiplications

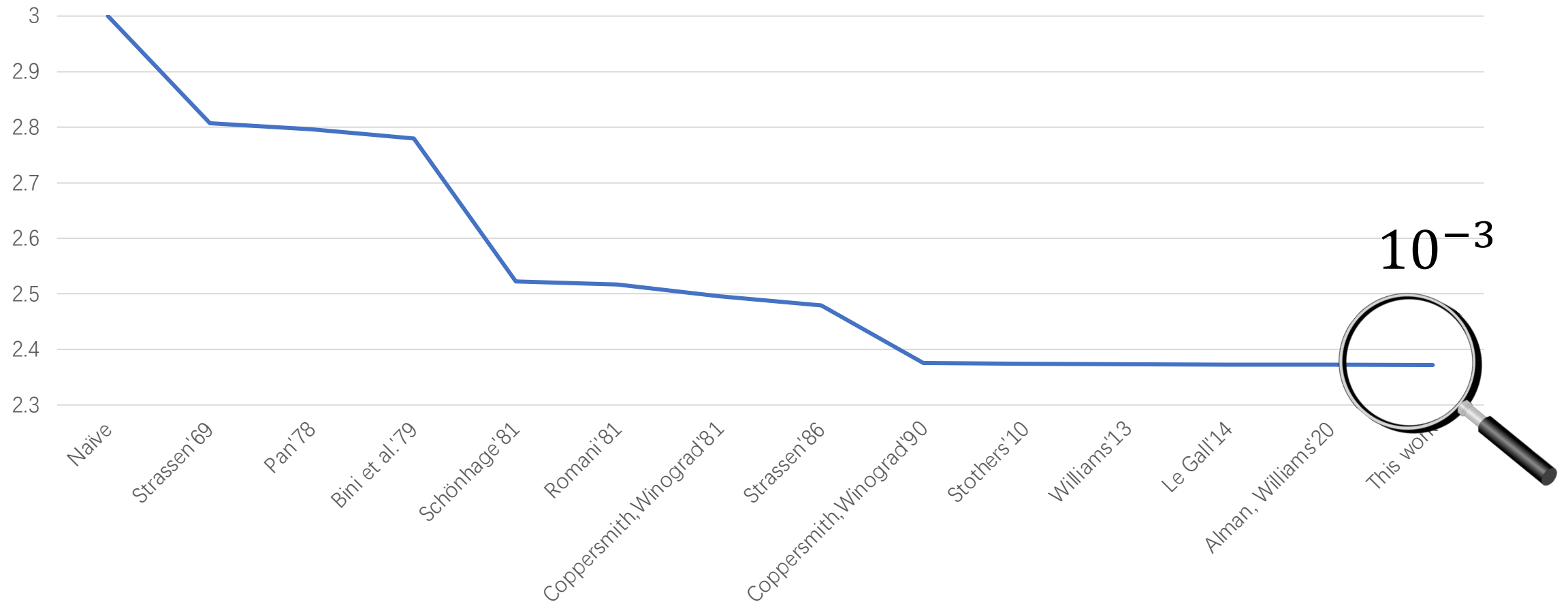
Fast Matrix Multiplication

Complexity. $O(n^\omega)$. $2 \leq \omega \leq 3$



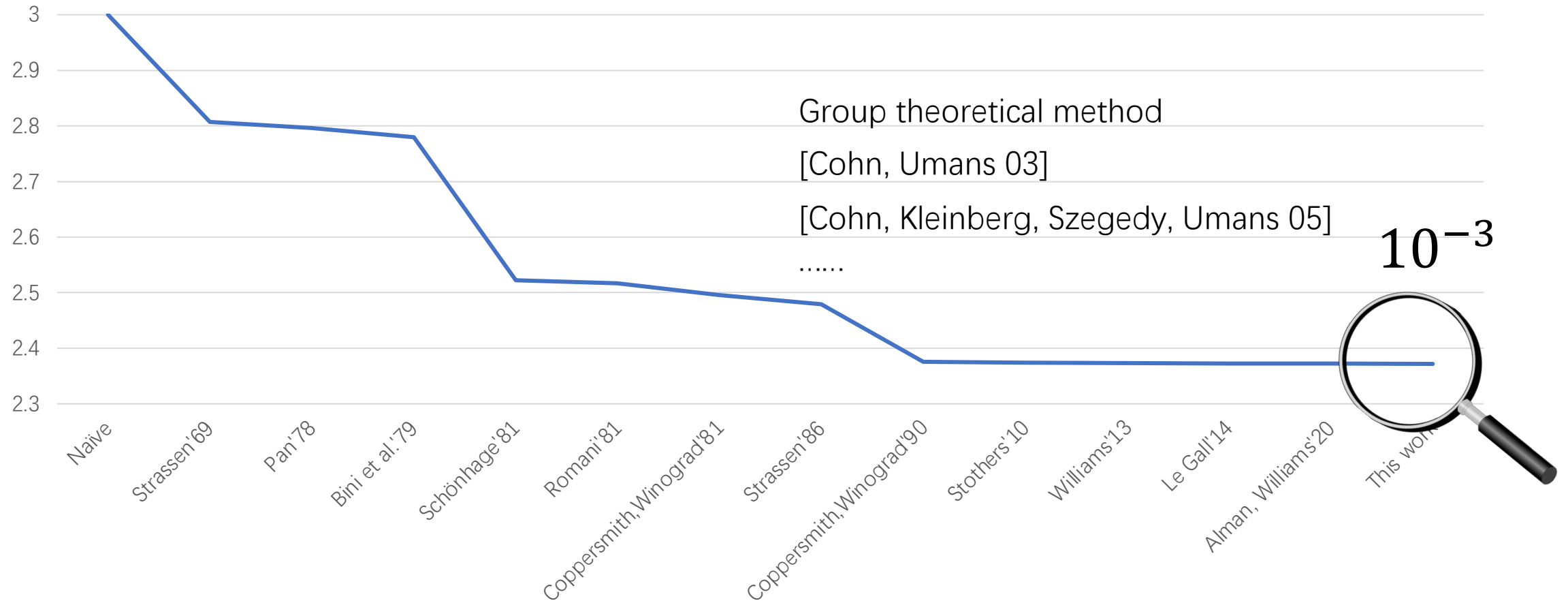
Fast Matrix Multiplication

Complexity. $O(n^\omega)$. $2 \leq \omega \leq 3$

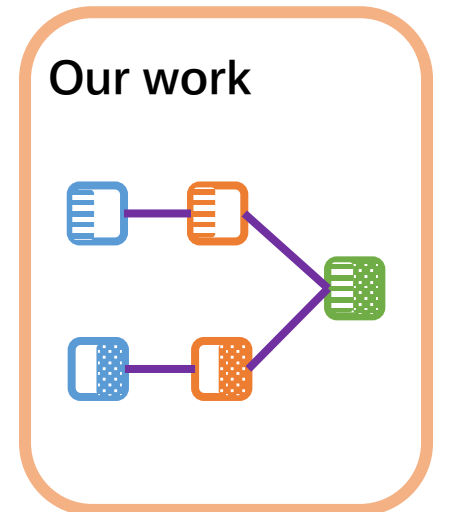


Fast Matrix Multiplication

Complexity. $O(n^\omega)$. $2 \leq \omega \leq 3$



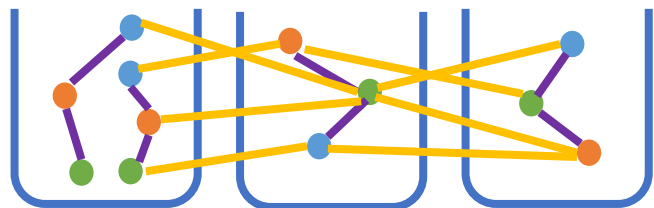
A Messy Storyline



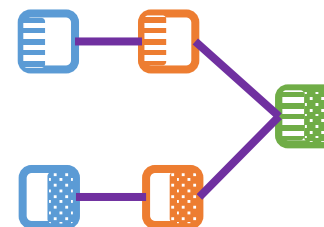
A Messy Storyline

Hashing

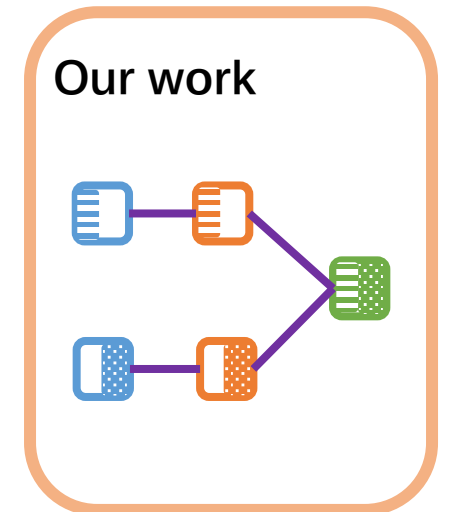
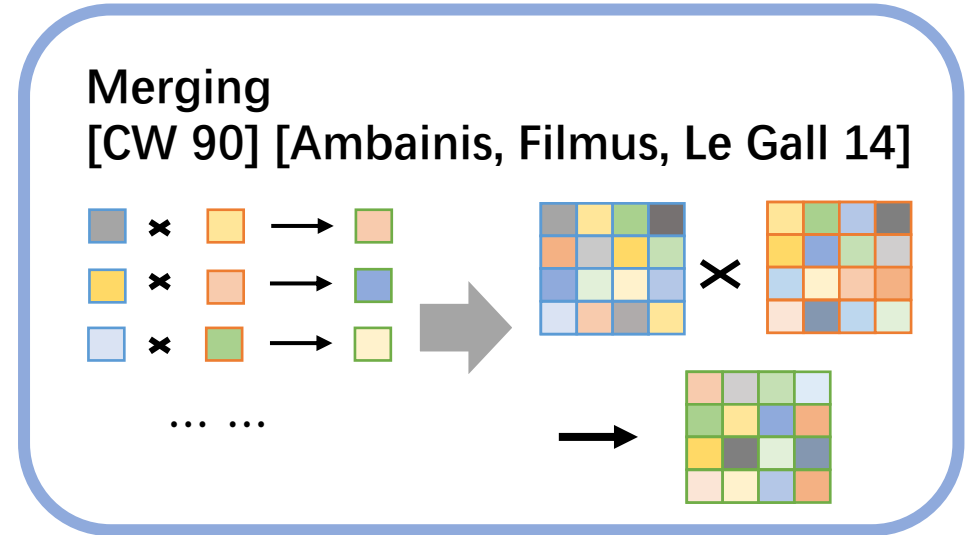
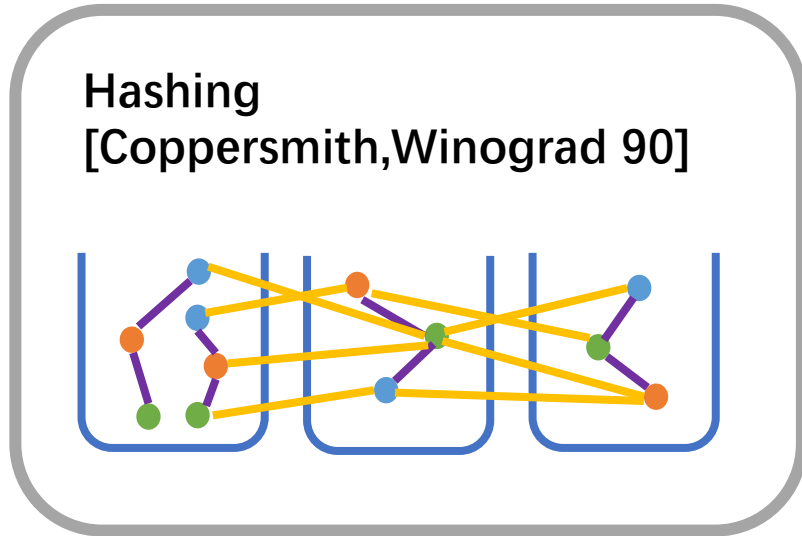
[Coppersmith, Winograd 90]



Our work

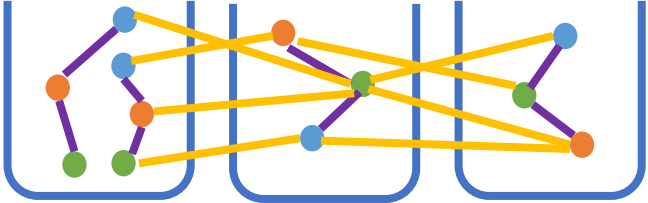


A Messy Storyline



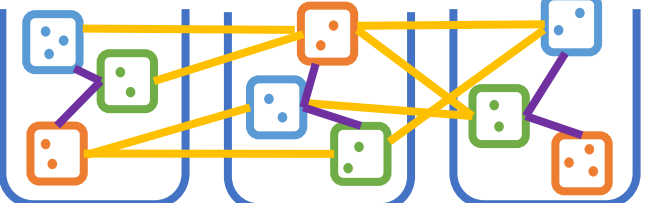
A Messy Storyline

Hashing
[Coppersmith, Winograd 90]



A diagram showing a network of nodes and connections. On the left, there are three vertical blue bars. Nodes are represented by colored circles (orange, blue, green) connected by yellow and purple lines. The connections are dense and cross between the bars.

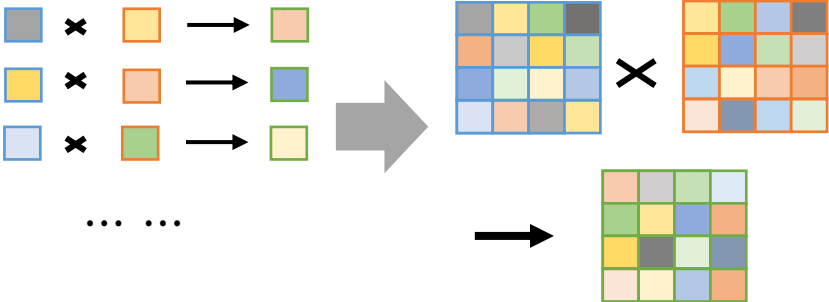
2/4/8/16/32-th Power
[CW 90][Stothers 10]
[Williams 13] [Le Gall 14]



A diagram showing a network of nodes and connections. On the left, there are three vertical blue bars. Nodes are represented by colored squares (blue, orange, green) with dots, connected by yellow and purple lines. The connections are dense and cross between the bars.

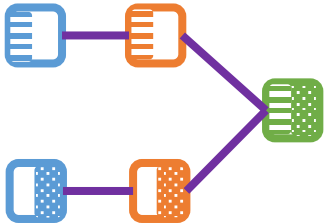


Merging
[CW 90] [Ambainis, Filmus, Le Gall 14]



A diagram illustrating a merging process. It shows a sequence of operations: a 3x3 grid of colored squares (grey, orange, yellow, blue, green) multiplied by another 3x3 grid of colored squares (orange, blue, green, grey, yellow, blue, orange, blue, green). The result is a larger 6x6 grid of colored squares. The process is shown as a sequence of operations: a 3x3 grid multiplied by a 3x3 grid, resulting in a 6x6 grid.

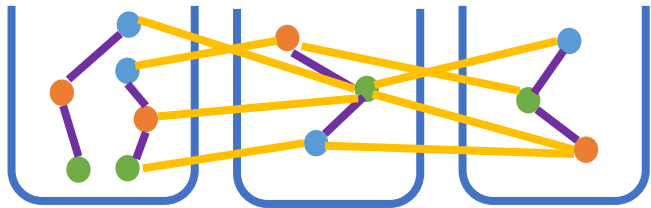
Our work



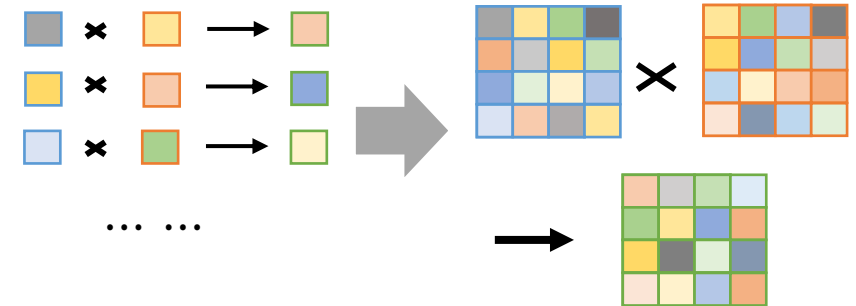
A diagram showing a network structure. It consists of four nodes on the left (two blue, two orange) connected by purple lines to a single green node on the right. The nodes are arranged in two rows of two.

A Messy Storyline

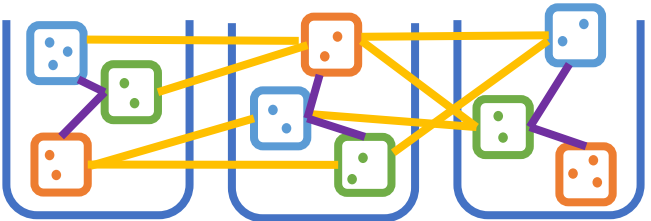
Hashing
[Coppersmith, Winograd 90]



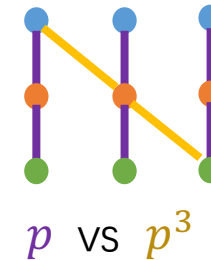
Merging
[CW 90] [Ambainis, Filmus, Le Gall 14]



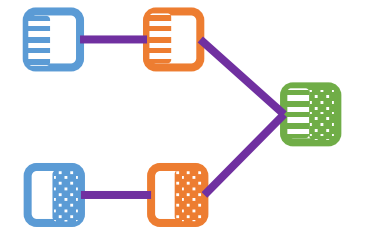
2/4/8/16/32-th Power
[CW 90][Stothers 10]
[Williams 13] [Le Gall 14]



Refined Laser Method
[Alman, Williams 21]

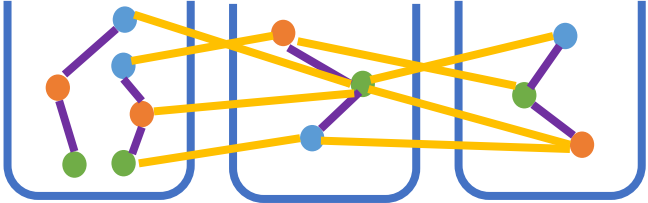


Our work



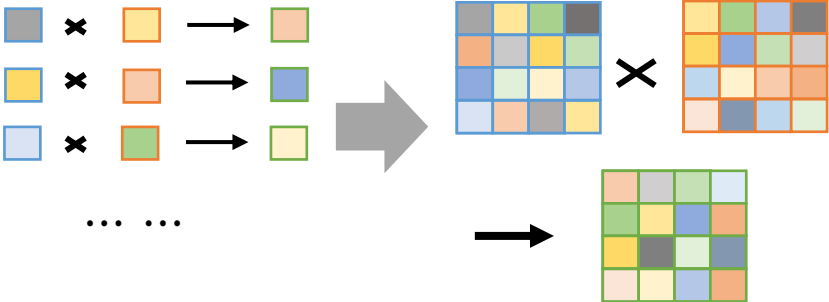
A Messy Storyline

Hashing
[Coppersmith, Winograd 90]



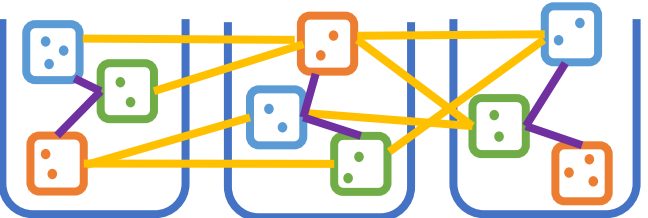
A diagram showing a hashing circuit with three stages. Each stage has two columns of nodes. The nodes are connected by yellow and purple lines, representing a complex network of operations.

Merging
[CW 90] [Ambainis, Filmus, Le Gall 14]



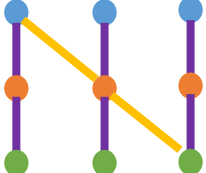
A diagram illustrating the merging process. It shows a sequence of operations: a grey block multiplied by an orange block results in a light brown block. A yellow block multiplied by an orange block results in a blue block. A light blue block multiplied by a green block results in a light green block. These operations are then combined into a larger grid structure, with a final arrow pointing to a single merged grid.

2/4/8/16/32-th Power
[CW 90][Stothers 10]
[Williams 13] [Le Gall 14]



A diagram showing a circuit for computing powers. It features three stages with nodes represented by dice faces. The nodes are connected by yellow and purple lines, forming a complex network.

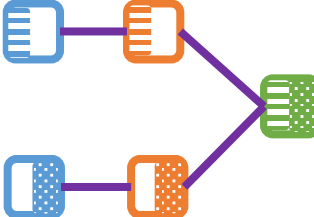
Refined Laser Method
[Alman, Williams 21]



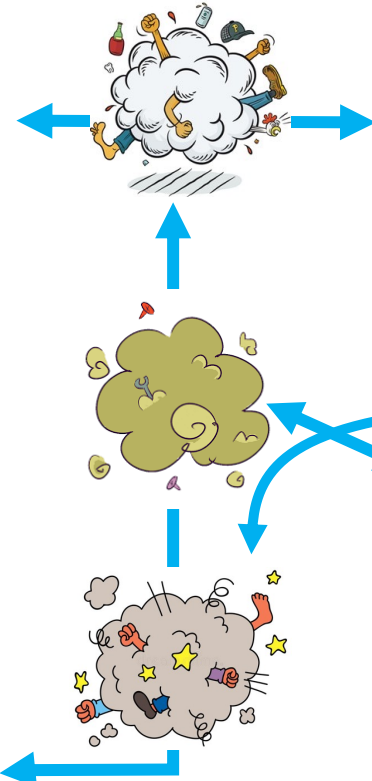
p vs p^3

A diagram illustrating the refined laser method. It shows three vertical columns of nodes. The nodes are connected by yellow and purple lines, forming a network. Below the diagram, the text p vs p^3 is written.

Our work

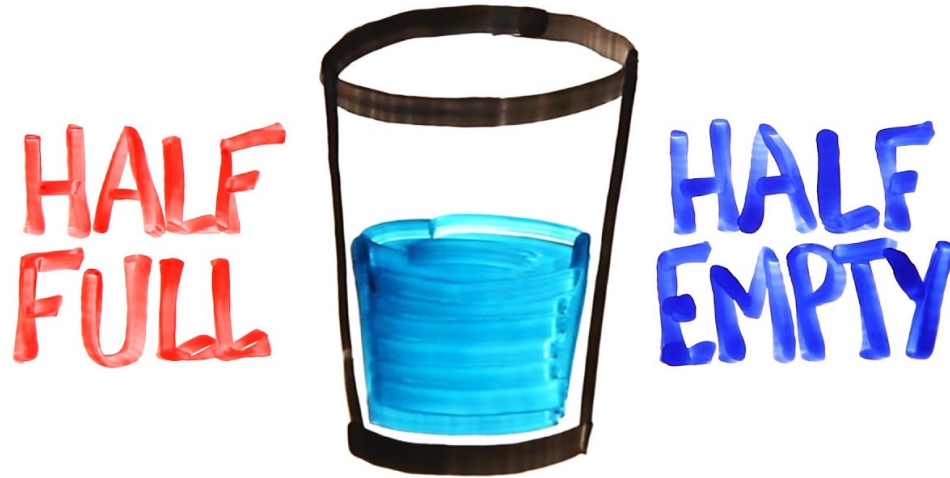


A diagram showing a network of nodes and connections. It features two columns of nodes on the left, connected by purple lines to a single node on the right. The nodes are represented by dice faces.

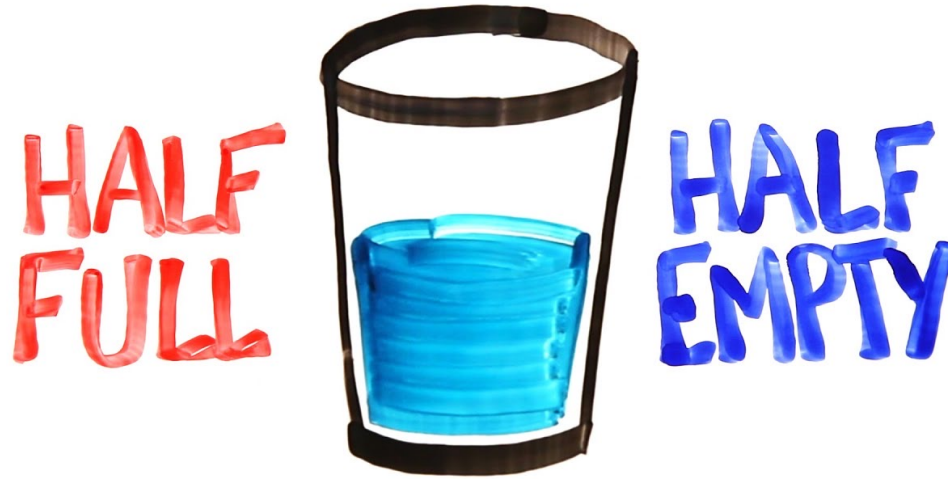


Half full or Half empty?

Good News:
This is the chance
for you to clean it
up!

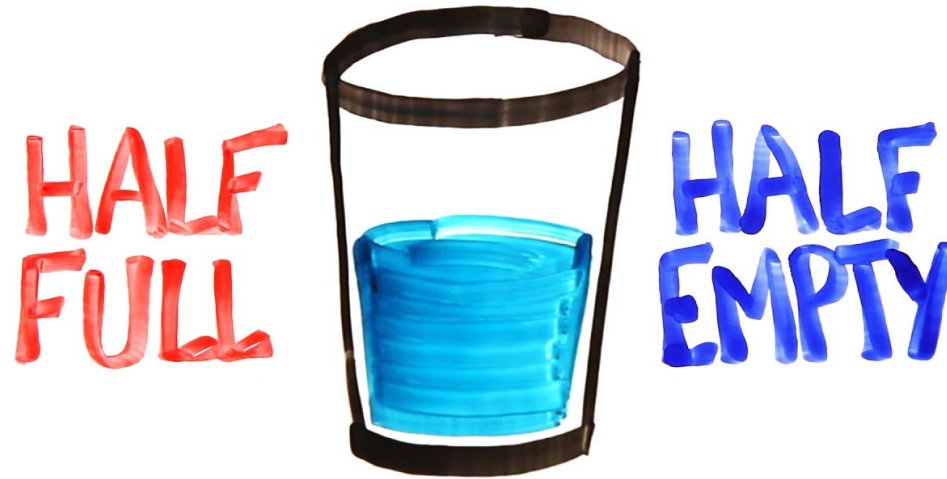


Half full or Half empty?



Maybe there is no simple & fast & elegant algo

Half full or Half empty?



Maybe there is no simple & fast & elegant algo

Vassilevska Williams remembers a conversation she once had with Strassen about this: “I asked him if he thinks you can get [exponent 2] for matrix multiplication and he said, ‘no, no, no, no, no.’”

“Matrix Multiplication Inches Closer to Mythic Goal”, Quanta Magazine

Our Plan

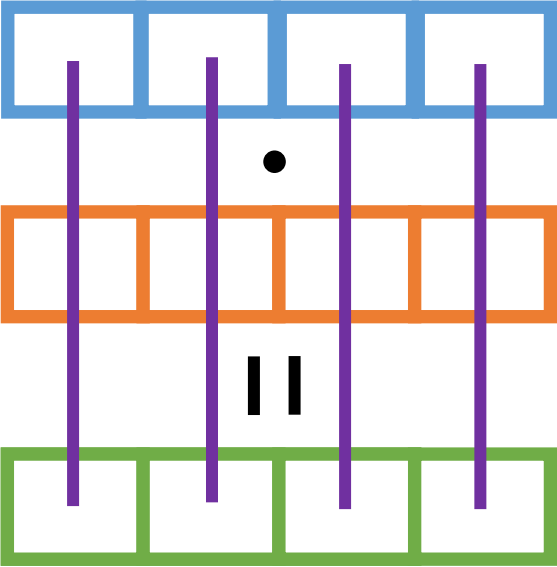
- 1. Tensor Formulation**
2. CW Algorithm From scratch
3. Main idea of our improvement

Tensor Formulation

Entry-wise Product

For $i = 1 \dots n$

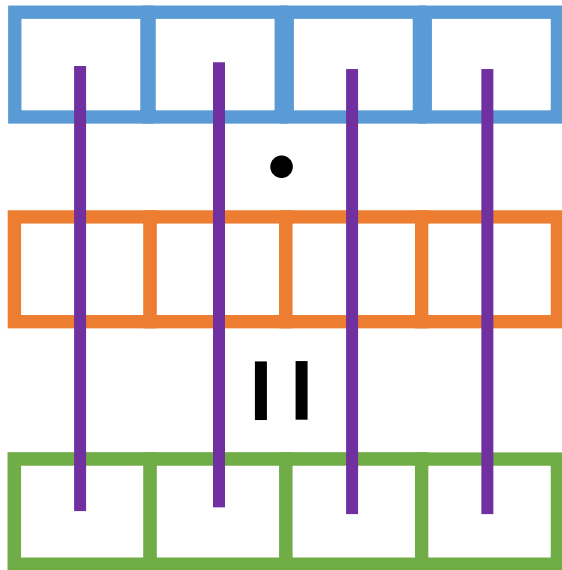
$$c_i \leftarrow a_i \cdot b_i$$



Tensor Formulation

Entry-wise Product

For $i = 1 \dots n$
 $c_i \leftarrow a_i \cdot b_i$



Multilinear Polynomial

Formal Variables

$$X = \{x_1, x_2, \dots, x_n\},$$

$$Y = \{y_1, y_2, \dots, y_n\},$$

$$Z = \{z_1, z_2, \dots, z_n\}.$$

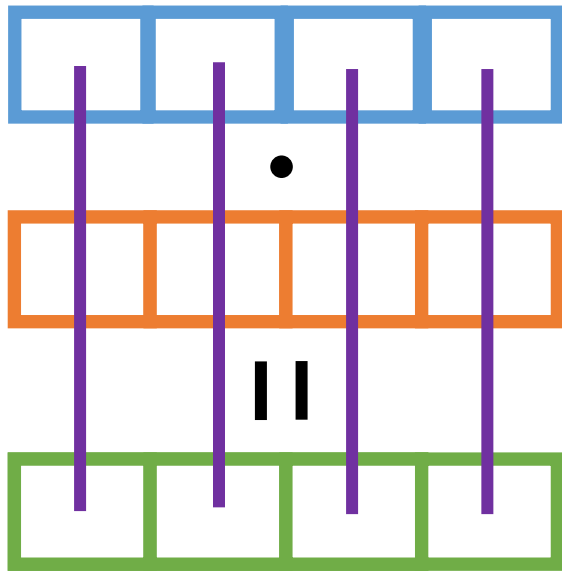
Polynomial

$$p(x, y, z) = \sum_{i=1}^n x_i y_i z_i$$

Tensor Formulation

Entry-wise Product

$$\text{For } i = 1 \dots n \\ c_i \leftarrow a_i \cdot b_i$$



Multilinear Polynomial

Formal Variables

$$X = \{x_1, x_2, \dots, x_n\},$$

$$Y = \{y_1, y_2, \dots, y_n\},$$

$$Z = \{z_1, z_2, \dots, z_n\}.$$

Polynomial

$$p(x, y, z) = \sum_{i=1}^n x_i y_i z_i$$

Task

Substitute $x \leftarrow a, y \leftarrow b$.

What is the univariate polynomial $p_{a,b}(z)$?

Tensor Formulation

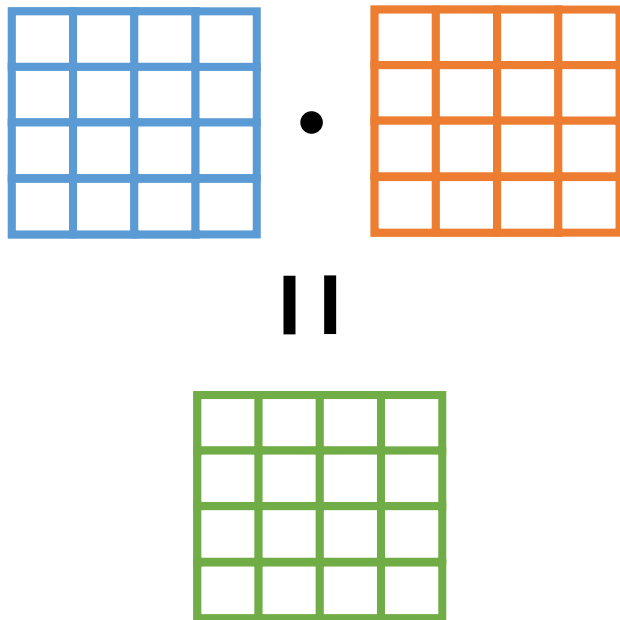
Matrix Multiplication

For $i = 1 \dots n$

For $j = 1 \dots n$

For $k = 1 \dots n$

$$c_{i,k} \leftarrow c_{i,k} + a_{i,j} \cdot b_{j,k}$$



Multilinear Polynomial

Polynomial

$$\sum_{i,j,k \in [n]} x_{i,j} y_{j,k} z_{i,k}$$

Task

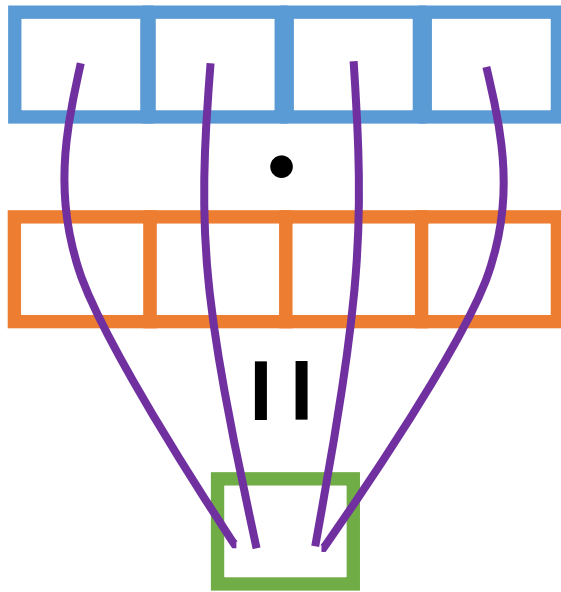
Substitute $x \leftarrow A, y \leftarrow B$.

What is the univariate polynomial $p_{a,b}(z)$?

Two more operations

Inner Product

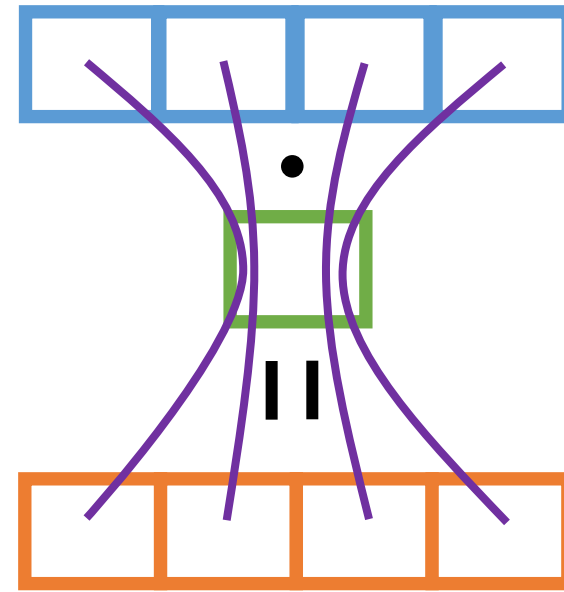
$$c_0 \leftarrow c_0 + a_i \cdot b_i$$



$$\sum_i x_i y_i z_0$$

“Inner Product”

$$c_i \leftarrow a_i \cdot b_0$$



$$\sum_i x_i y_0 z_i$$

Tensor Product

$$[a]_{i \in [n]} \otimes [a]_{i' \in [n]} \rightarrow [a_{i,i'}]_{i,i' \in [n]}$$

$$[b]_{i \in [n]} \otimes [b]_{i' \in [n]} \rightarrow [b_{i,i'}]_{i,i' \in [n]}$$

$$[c]_{i \in [n]} \otimes [c]_{i' \in [n]} \rightarrow [c_{i,i'}]_{i,i' \in [n]}$$

Tensor Product

Example

$$\begin{array}{ccc} \text{For } i = 1 \dots n & \otimes & \text{For } j = 1 \dots n \\ c_i \leftarrow a_i \cdot b_0 & & c_0 \leftarrow a_j \cdot b_j \\ & & \otimes & & \text{For } k = 1 \dots n \\ & & & & c_k \leftarrow a_0 \cdot b_k \end{array}$$

Tensor Product

Example

$$\begin{array}{l} \text{For } i = 1 \dots n \\ c_i \leftarrow a_i \cdot b_0 \end{array} \otimes \begin{array}{l} \text{For } j = 1 \dots n \\ c_0 \leftarrow a_j \cdot b_j \end{array} \otimes \begin{array}{l} \text{For } k = 1 \dots n \\ c_k \leftarrow a_0 \cdot b_k \end{array}$$

For $i = 1 \dots n$

For $j = 1 \dots n$

For $k = 1 \dots n$

$$c_{i,0,k} += a_{i,j,0} \cdot b_{0,j,k}$$

Tensor Product

Example

$$\begin{array}{l} \text{For } i = 1 \dots n \\ c_i \leftarrow a_i \cdot b_0 \end{array} \otimes \begin{array}{l} \text{For } j = 1 \dots n \\ c_0 \leftarrow a_j \cdot b_j \end{array} \otimes \begin{array}{l} \text{For } k = 1 \dots n \\ c_k \leftarrow a_0 \cdot b_k \end{array}$$

$$\begin{array}{l} \text{For } i = 1 \dots n \\ \quad \text{For } j = 1 \dots n \\ \quad \quad \text{For } k = 1 \dots n \\ \quad \quad \quad c_{i,k} += a_{i,j} \cdot b_{j,k} \end{array}$$

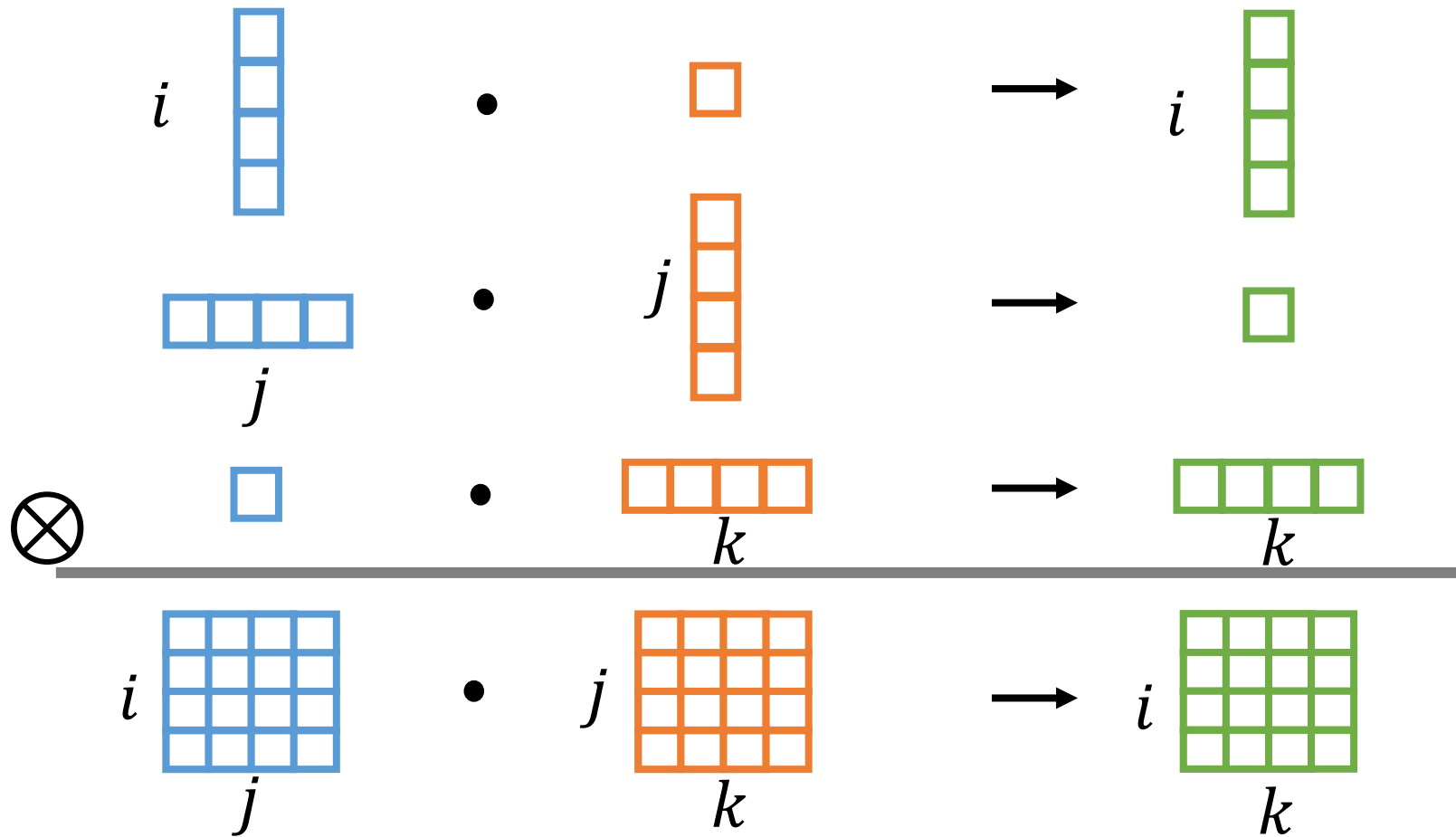
Tensor Product

For $i = 1 \dots n$

For $j = 1 \dots n$

For $k = 1 \dots n$

$$a_{i,j,0} \cdot b_{0,j,k} \rightarrow c_{i,0,k}$$



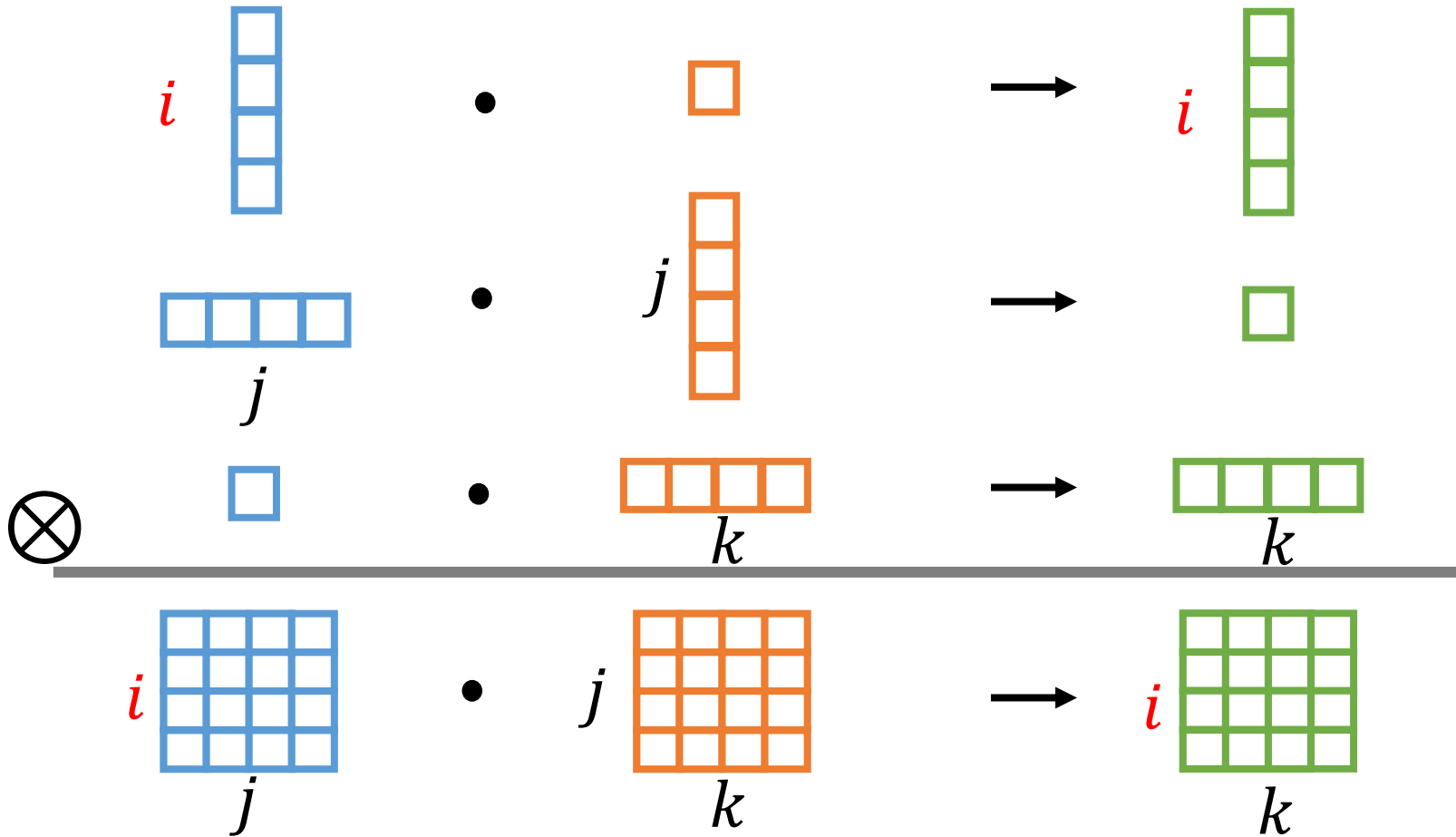
Tensor Product

For $i = 1 \dots n$

For $j = 1 \dots n$

For $k = 1 \dots n$

$$a_{i,j,0} \cdot b_{0,j,k} \rightarrow c_{i,0,k}$$



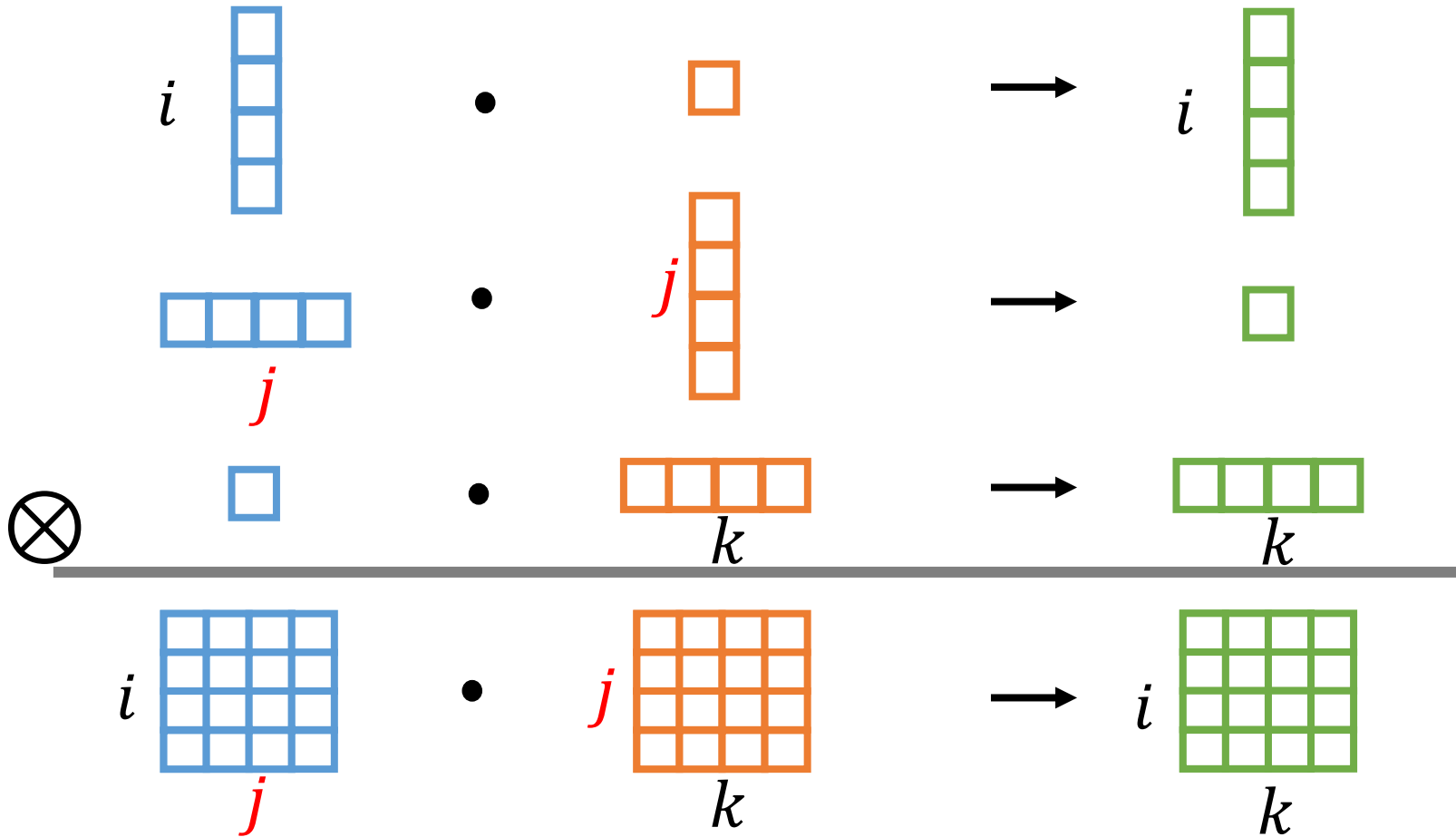
Tensor Product

For $i = 1 \dots n$

For $j = 1 \dots n$

For $k = 1 \dots n$

$$a_{i,j,0} \cdot b_{0,j,k} \rightarrow c_{i,0,k}$$



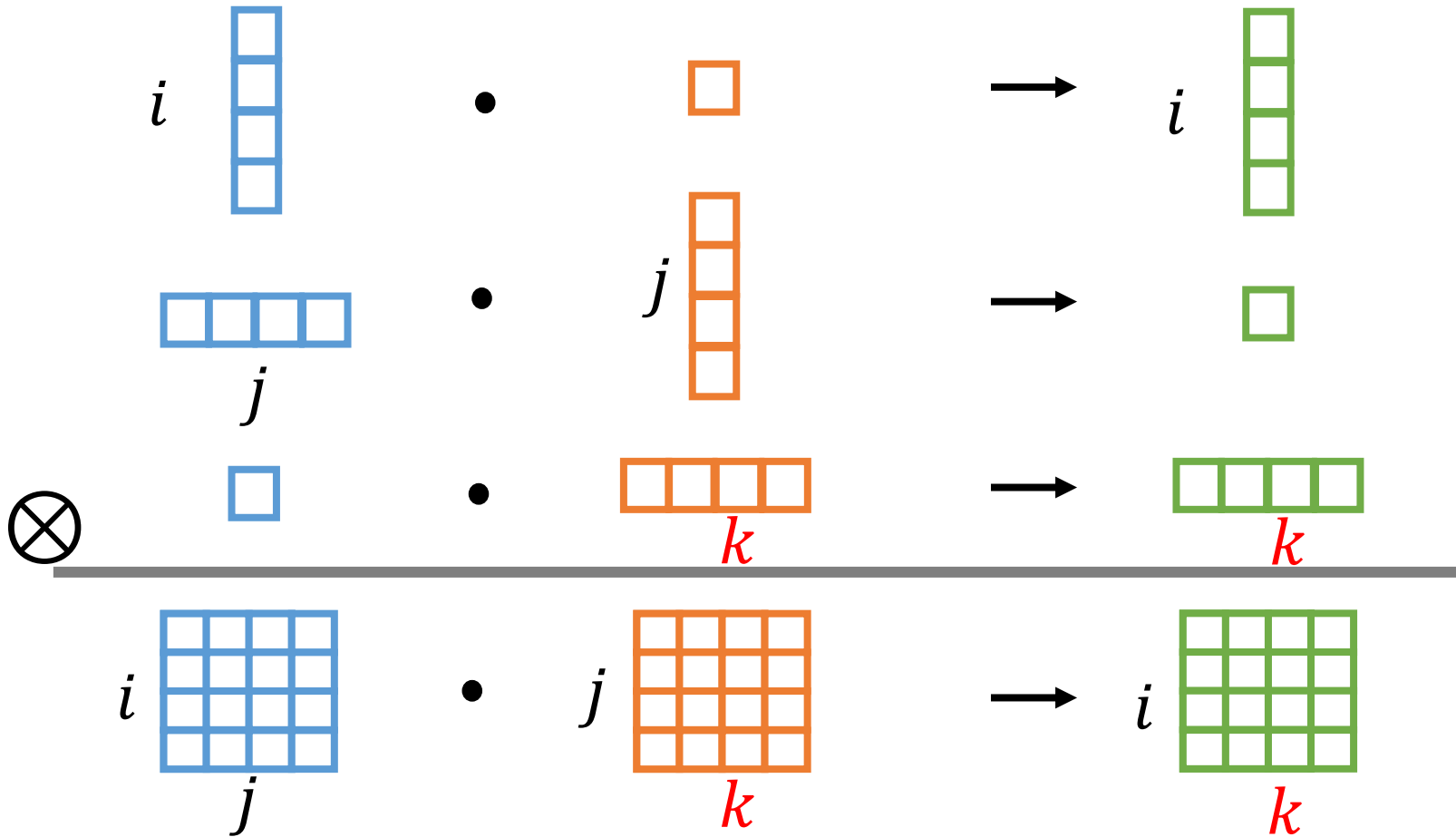
Tensor Product

For $i = 1 \dots n$

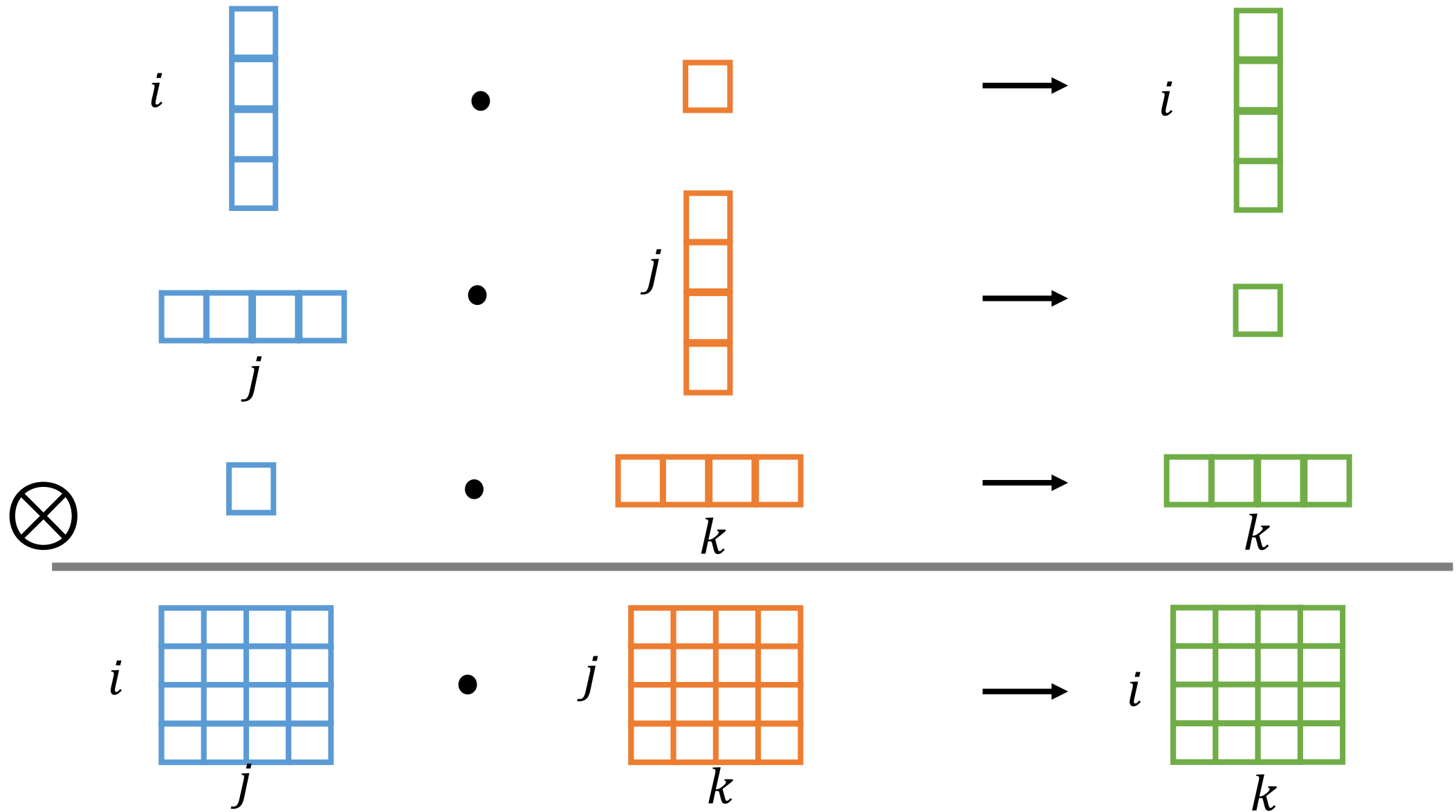
For $j = 1 \dots n$

For $k = 1 \dots n$

$$a_{i,j,0} \cdot b_{0,j,k} \rightarrow c_{i,0,k}$$



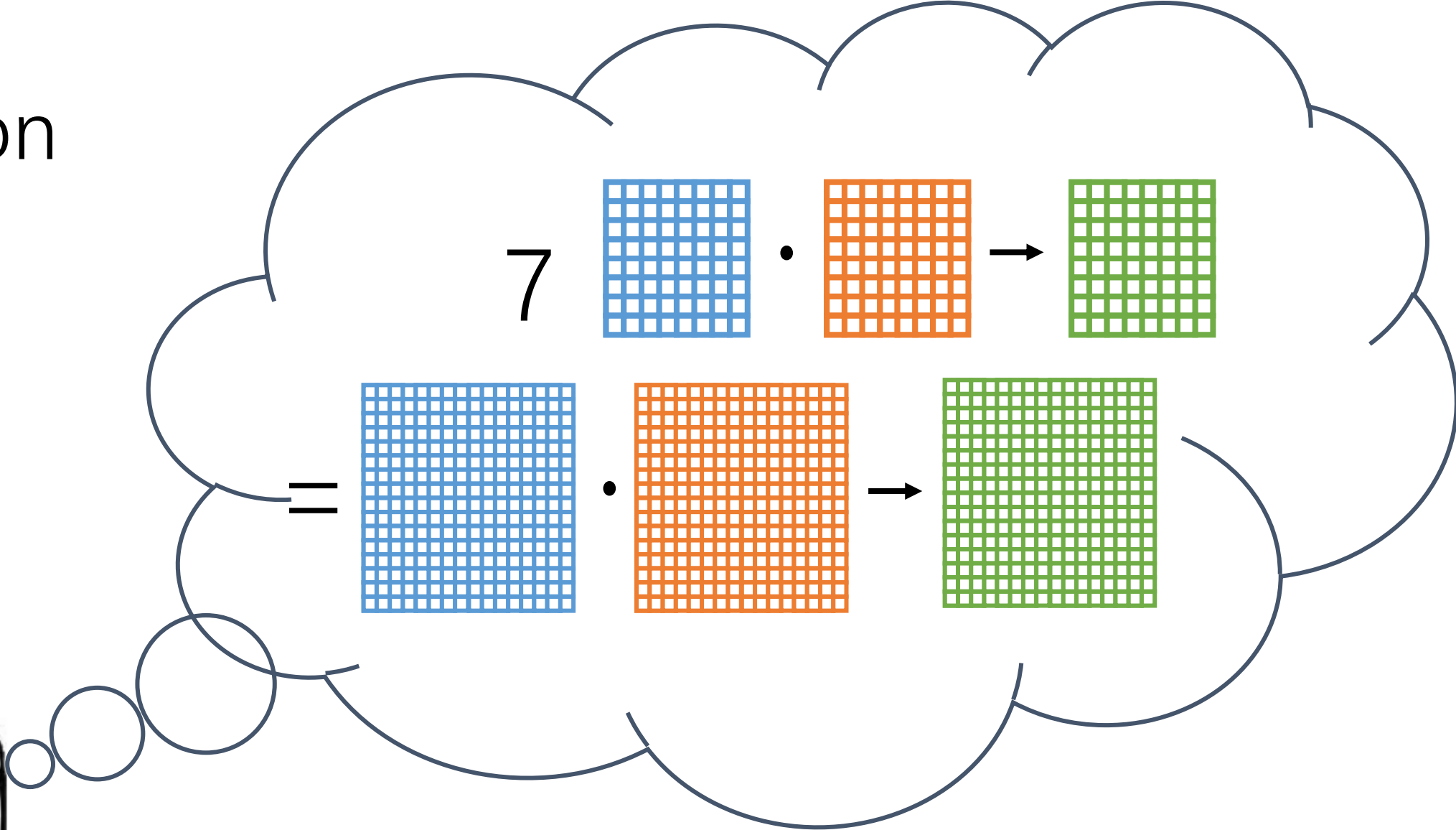
Matrix Multiplication = 3 Inner Products



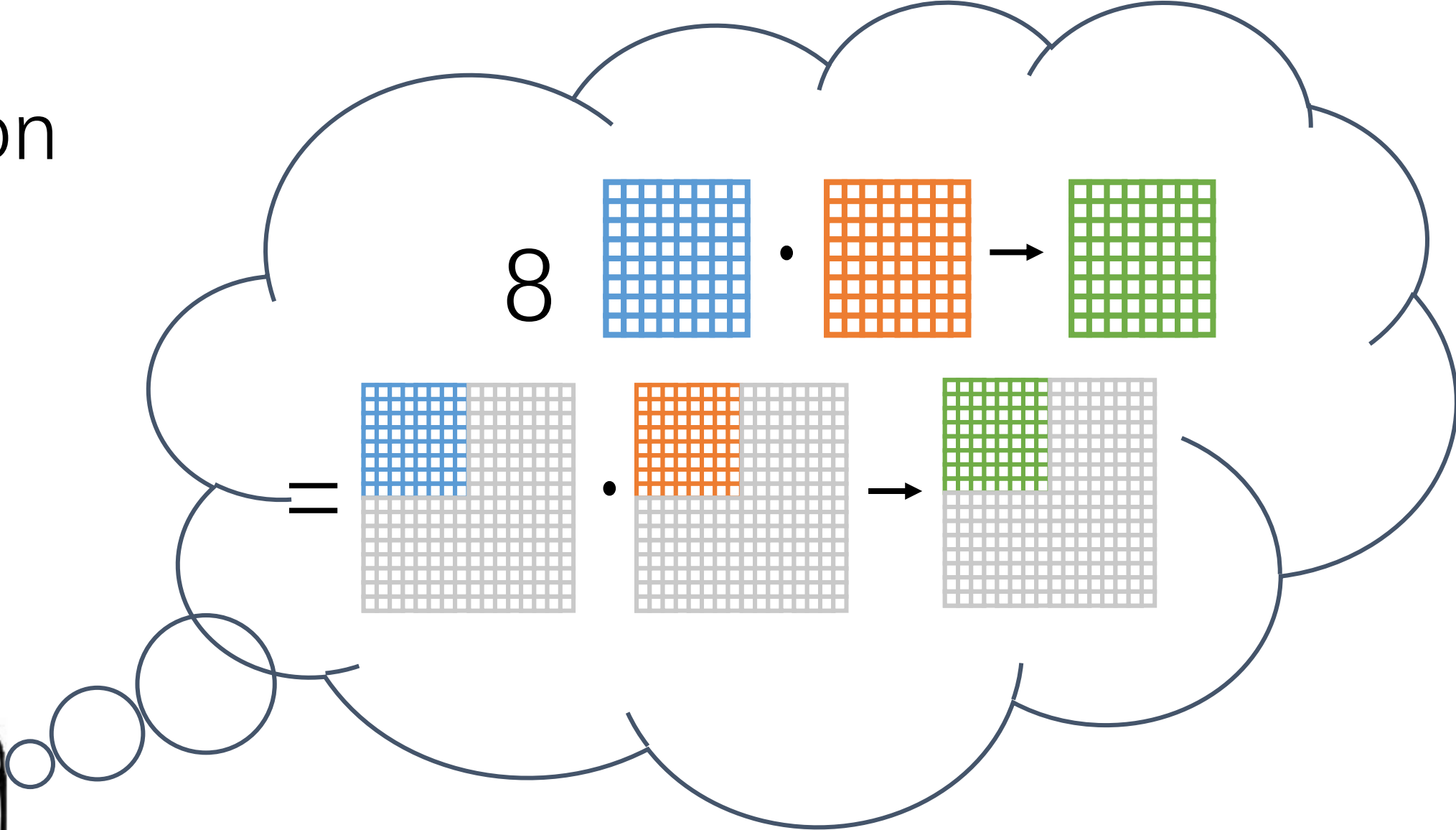
Our Plan

1. Tensor Formulation
2. **CW algorithm from scratch**
3. Main idea of our improvement

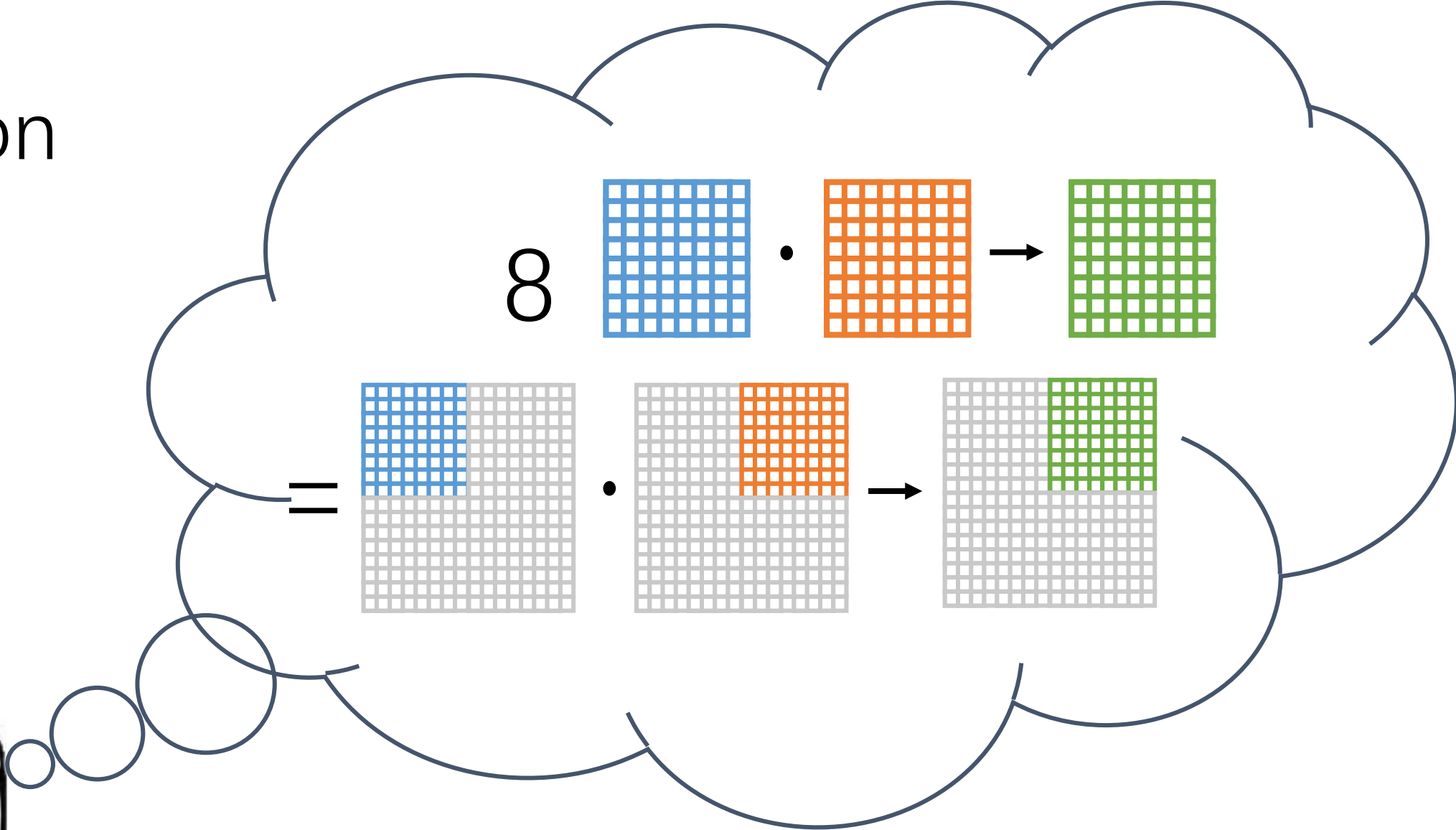
Intuition



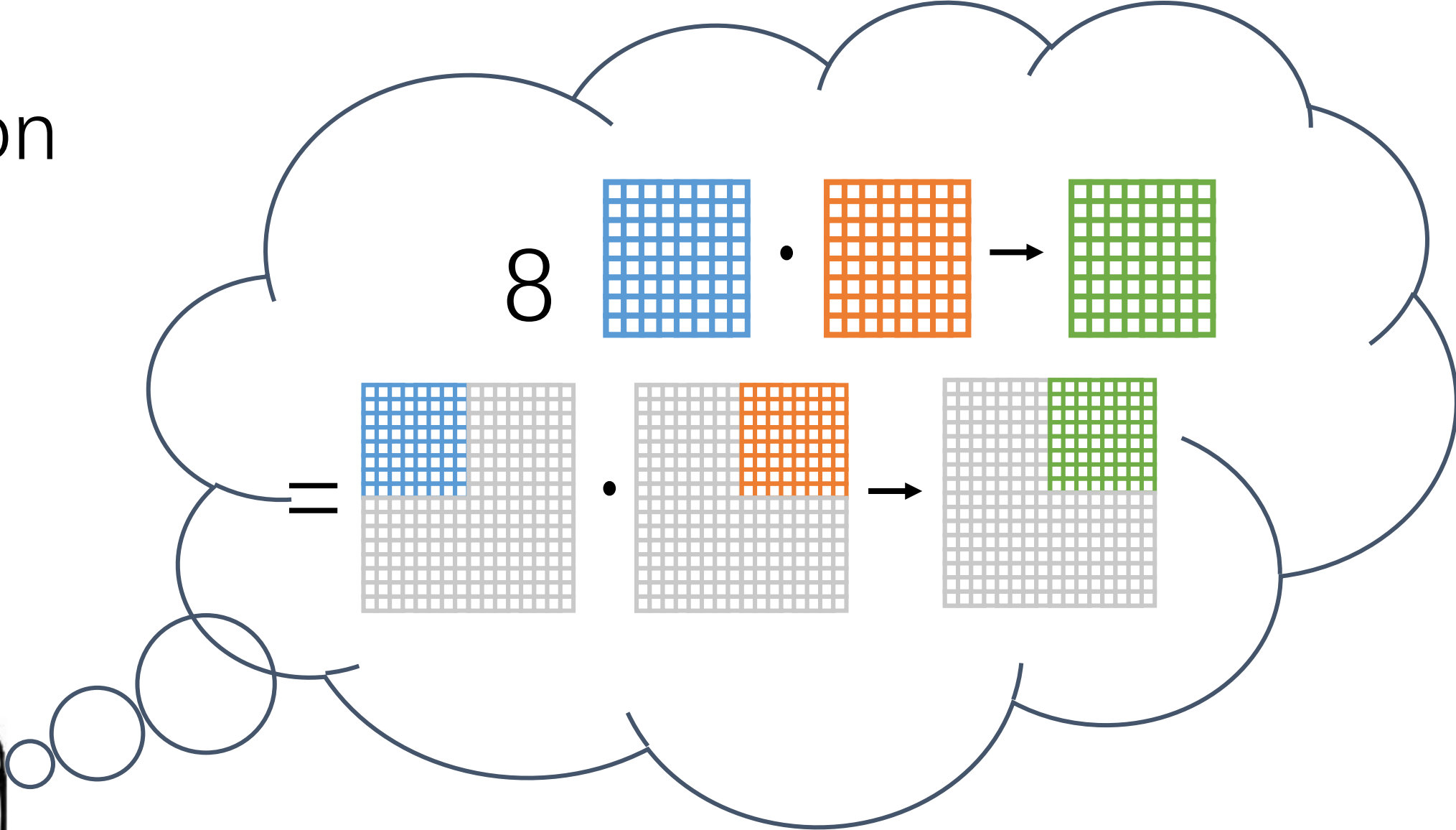
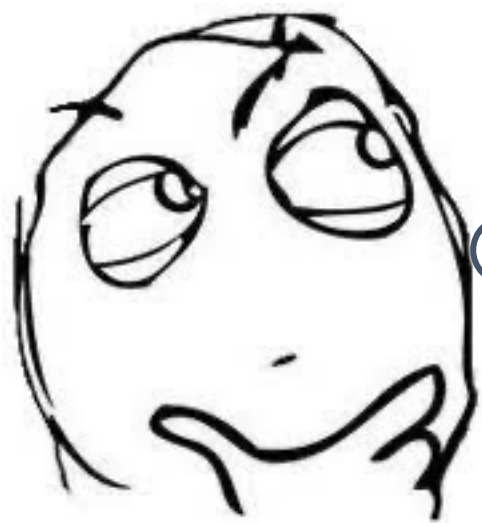
Intuition



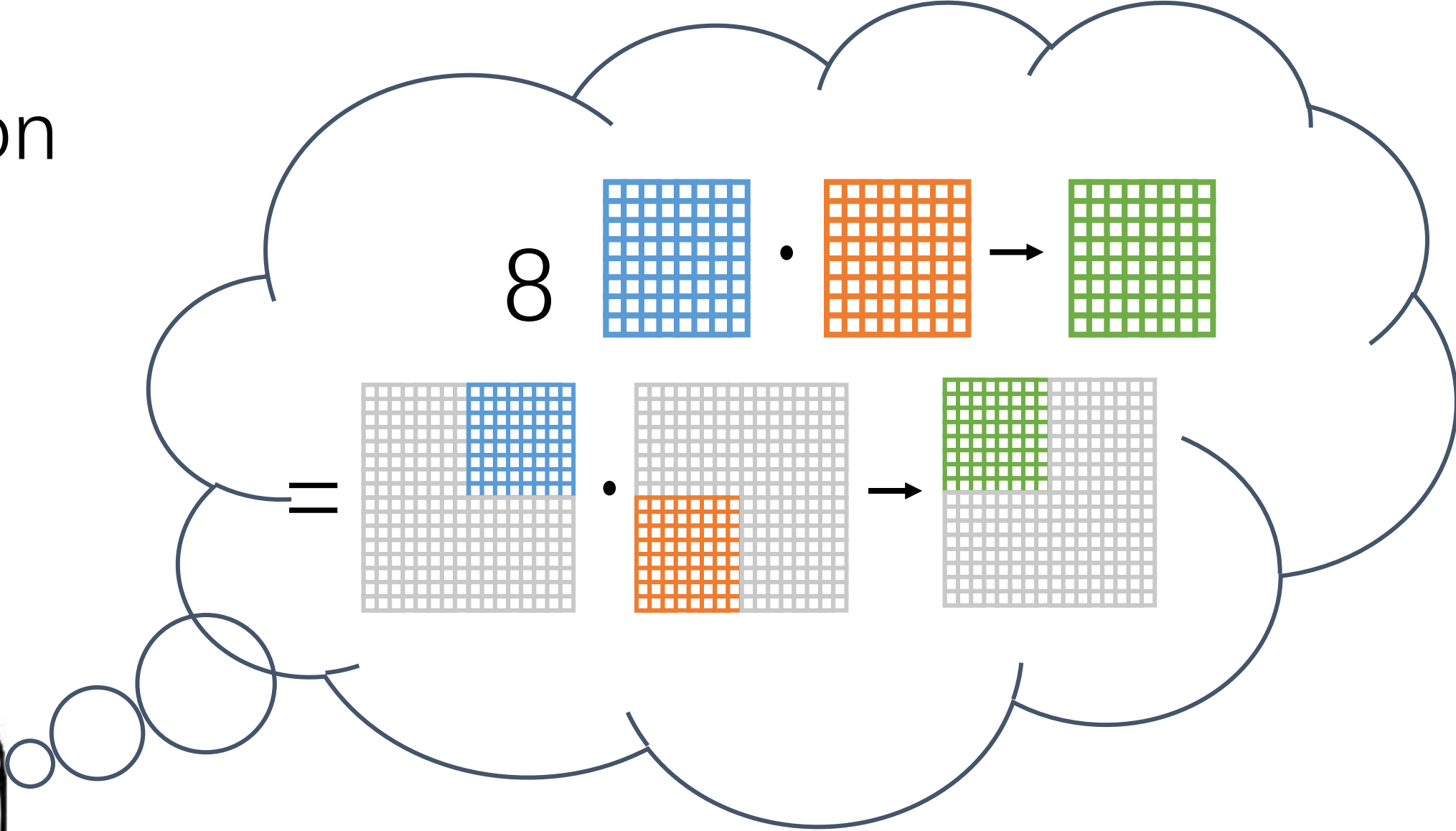
Intuition



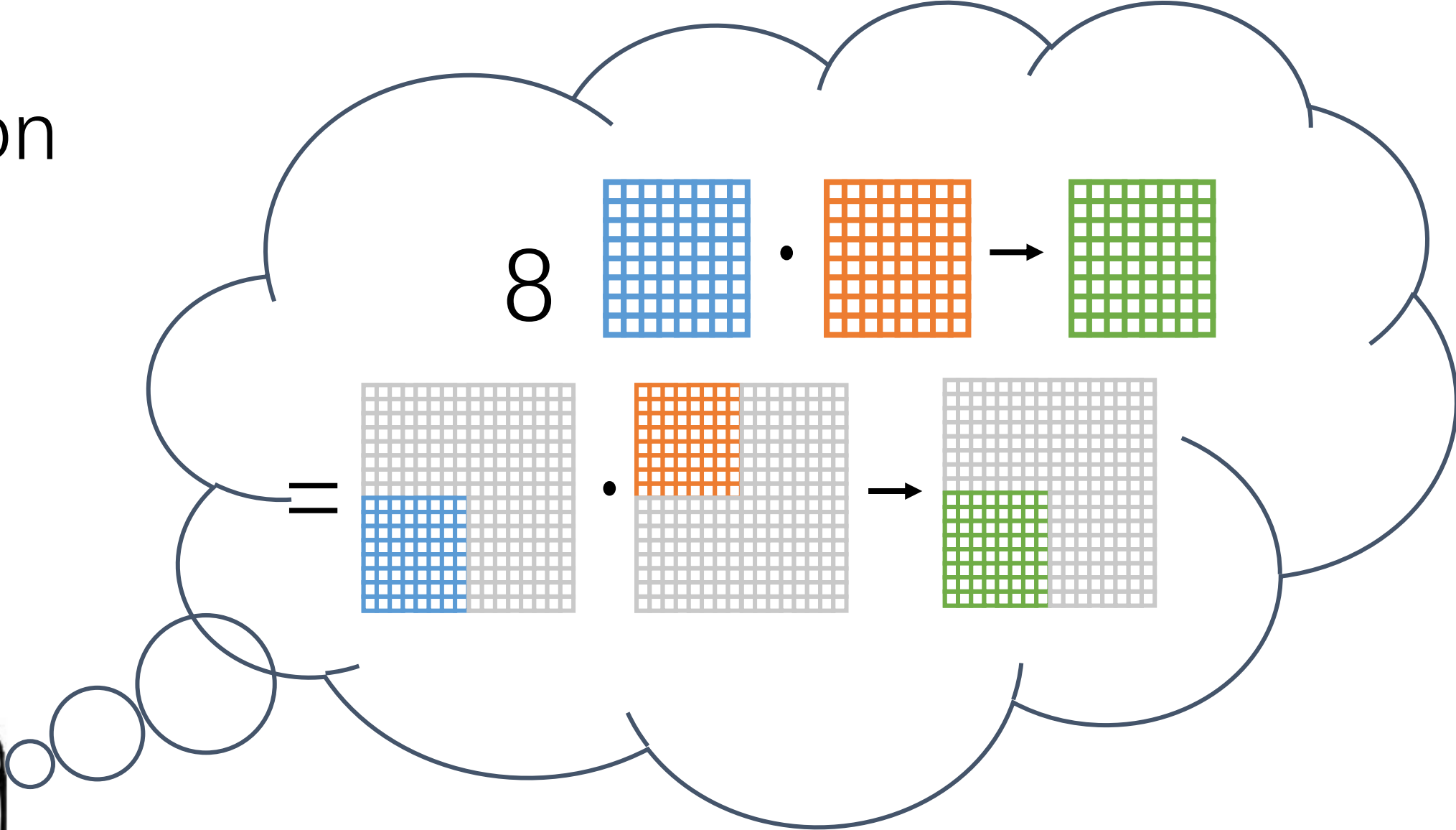
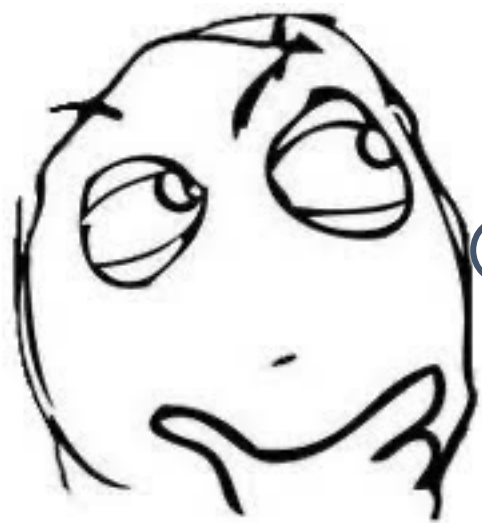
Intuition



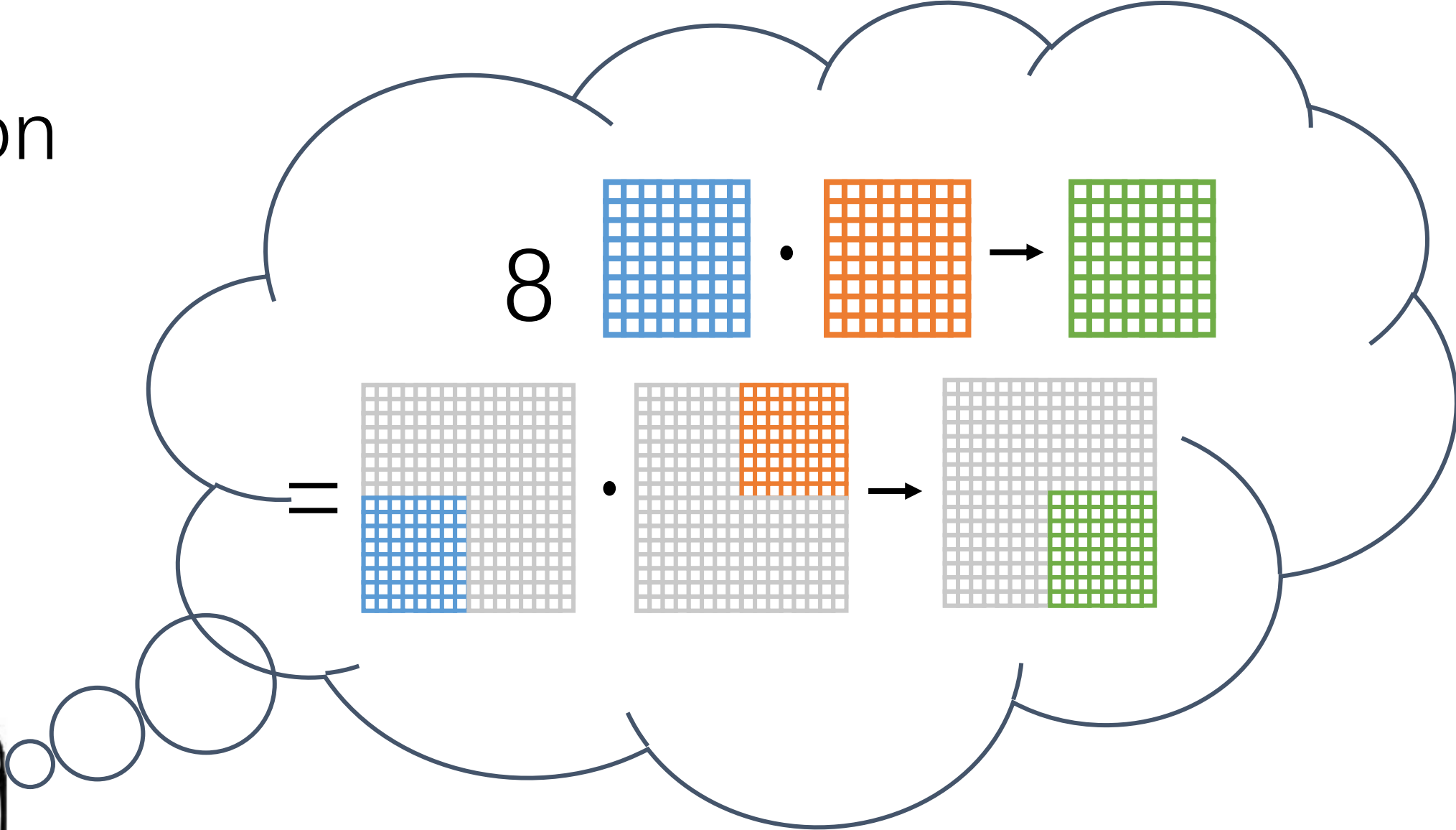
Intuition



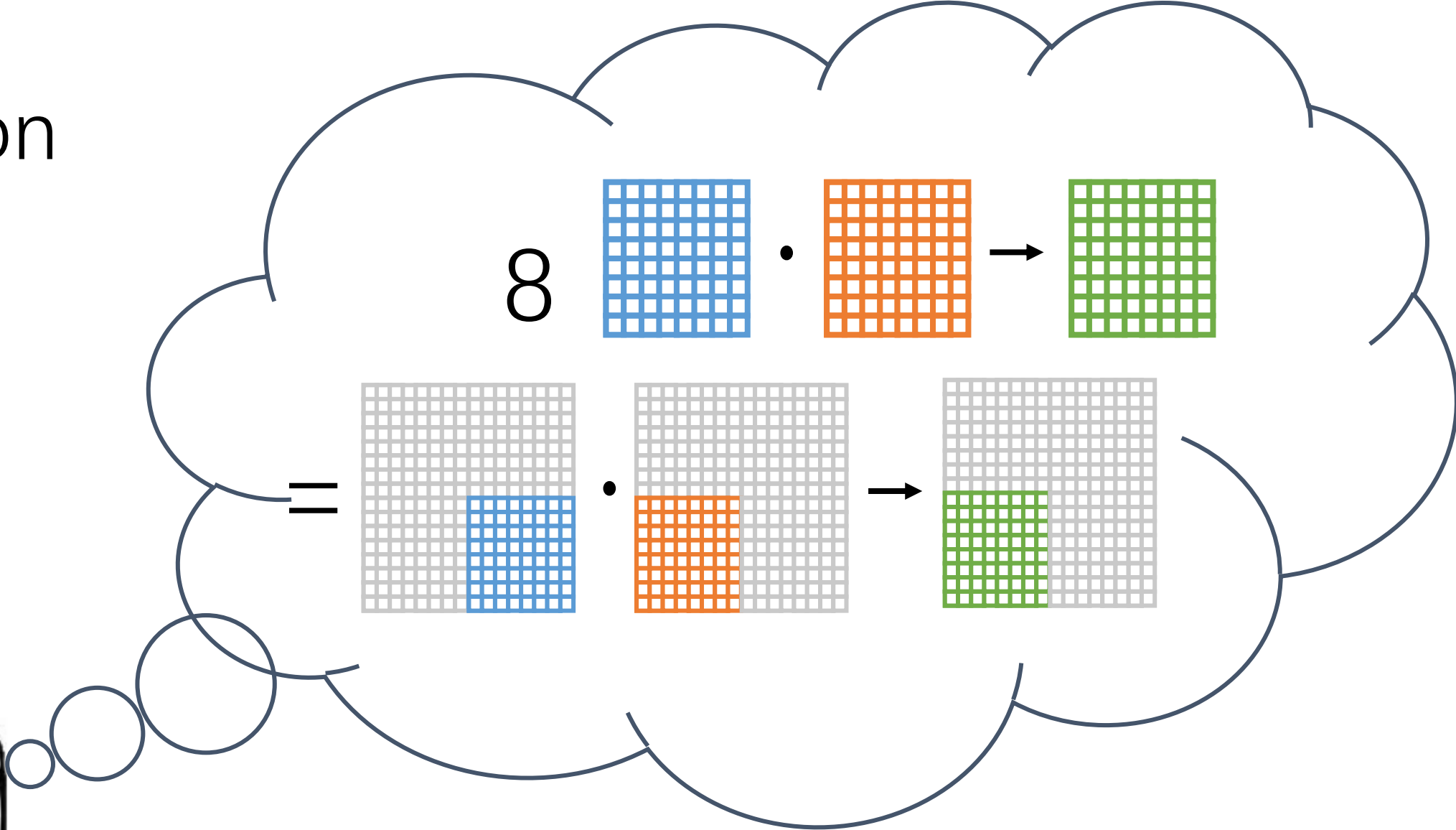
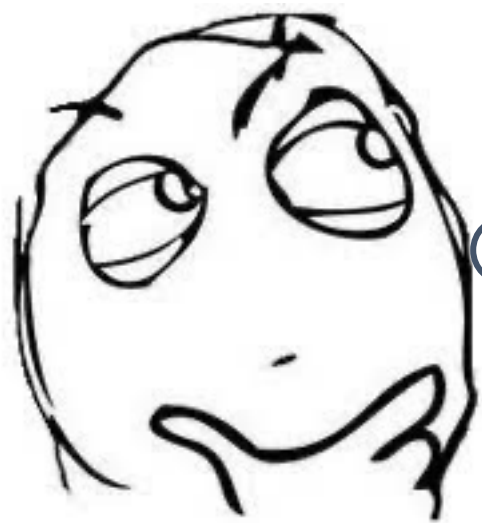
Intuition



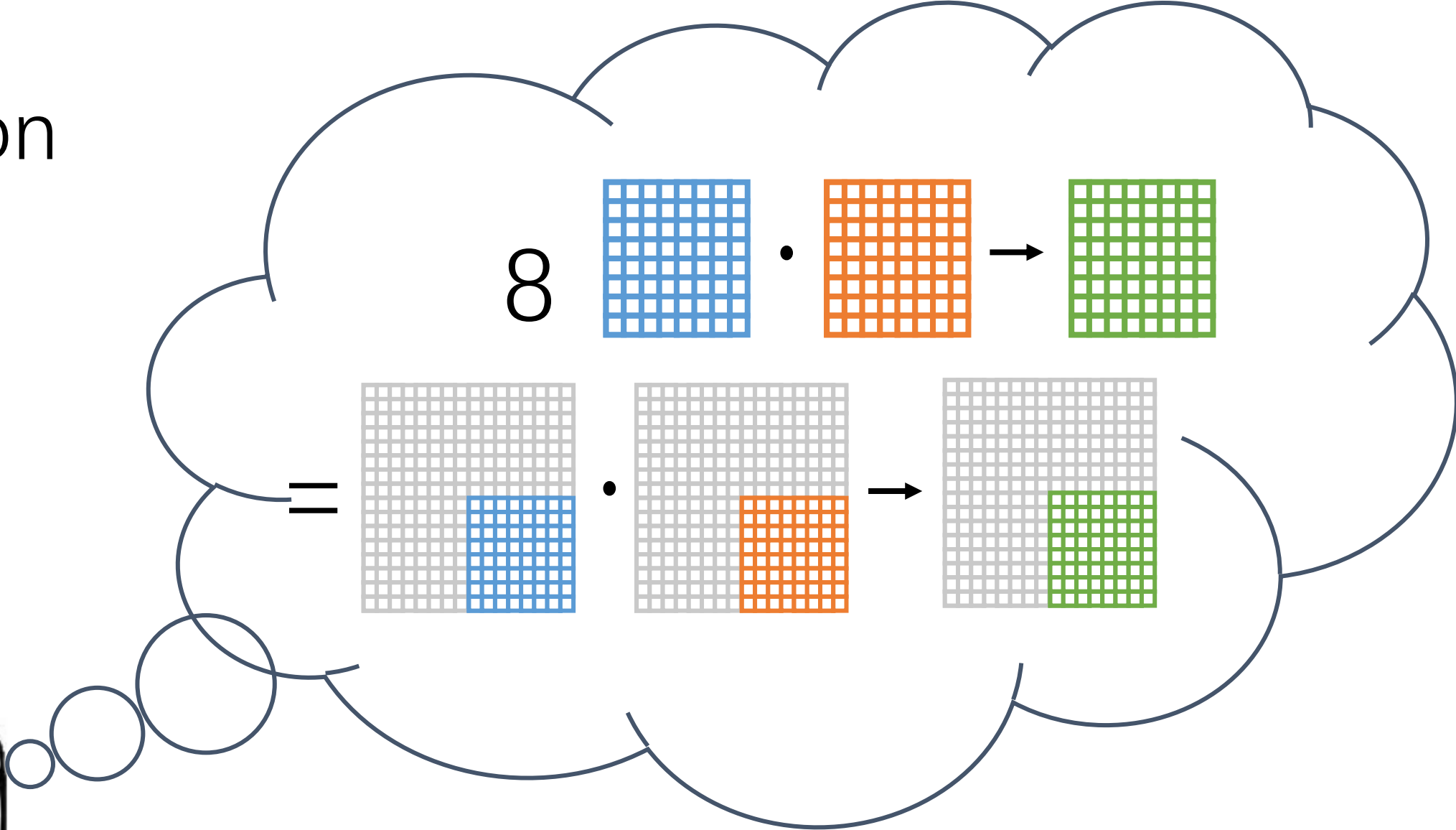
Intuition



Intuition



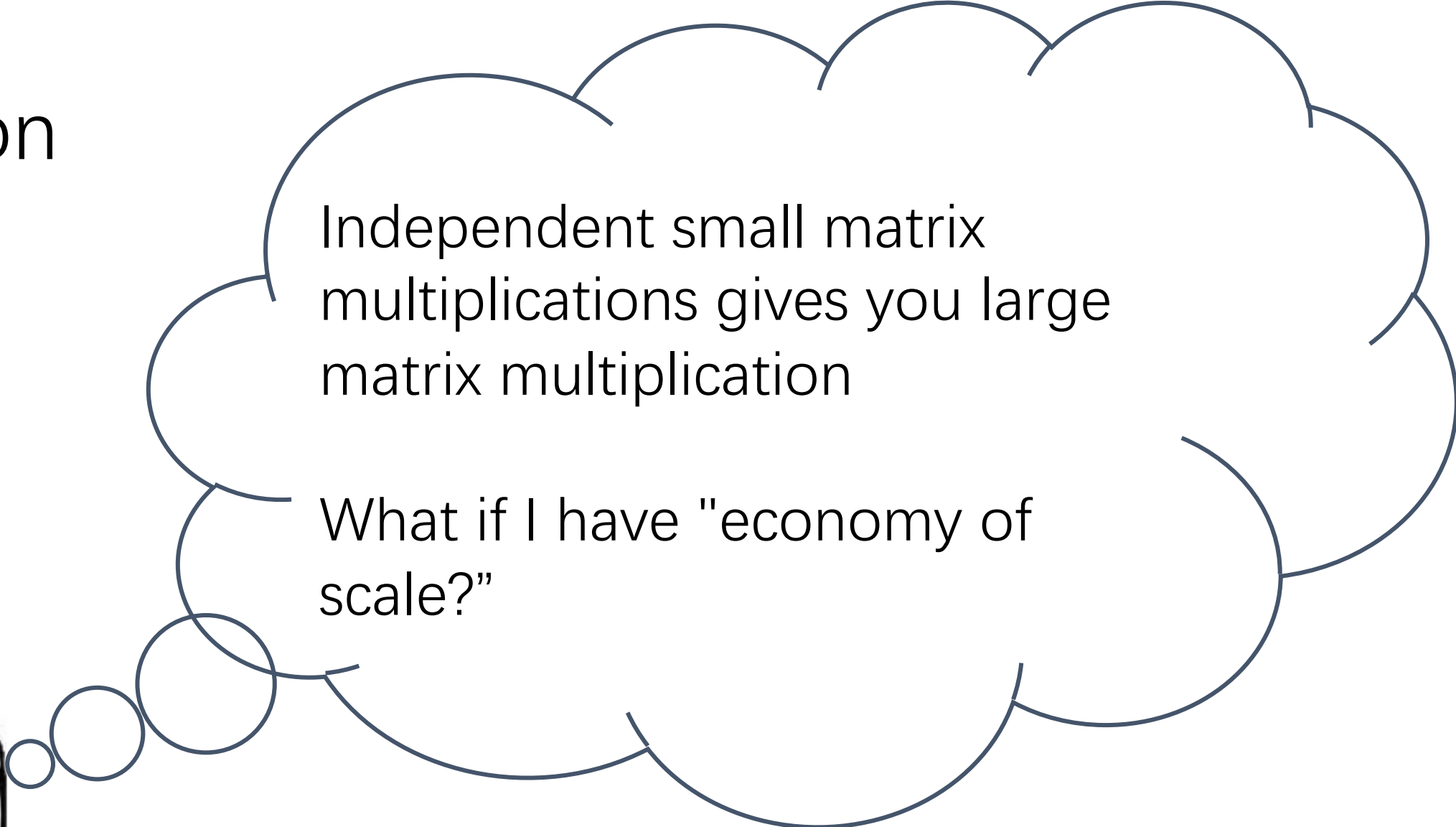
Intuition



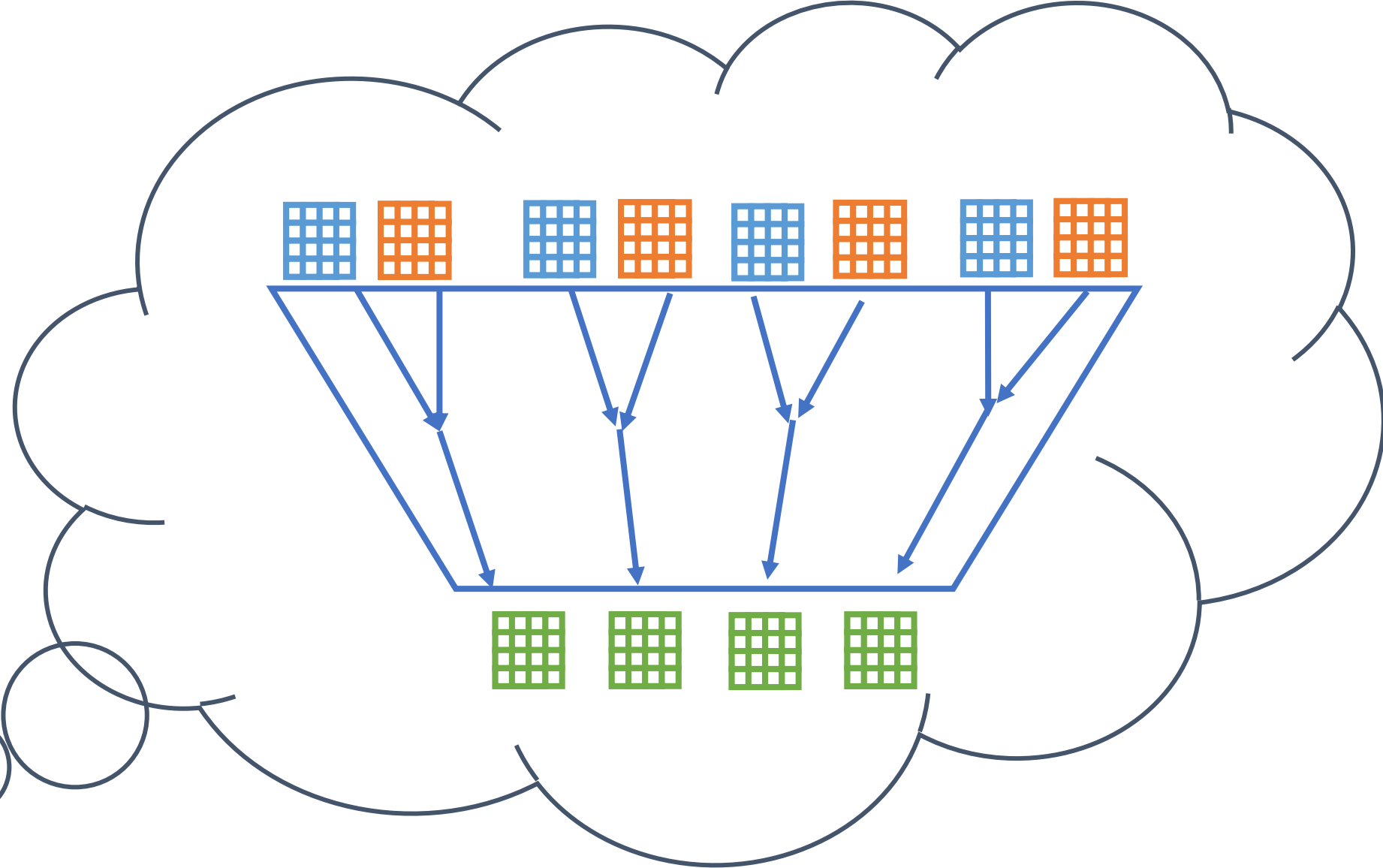
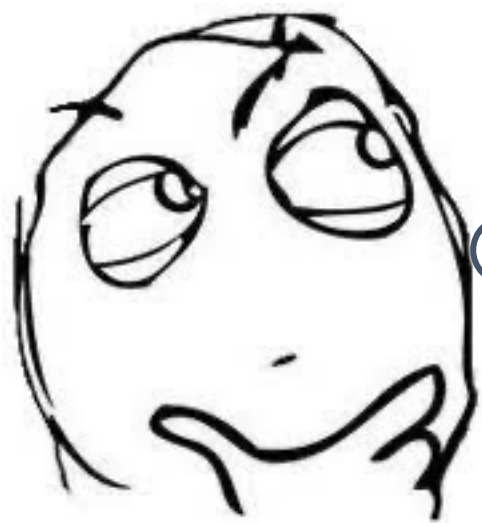
Intuition

Independent small matrix
multiplications gives you large
matrix multiplication

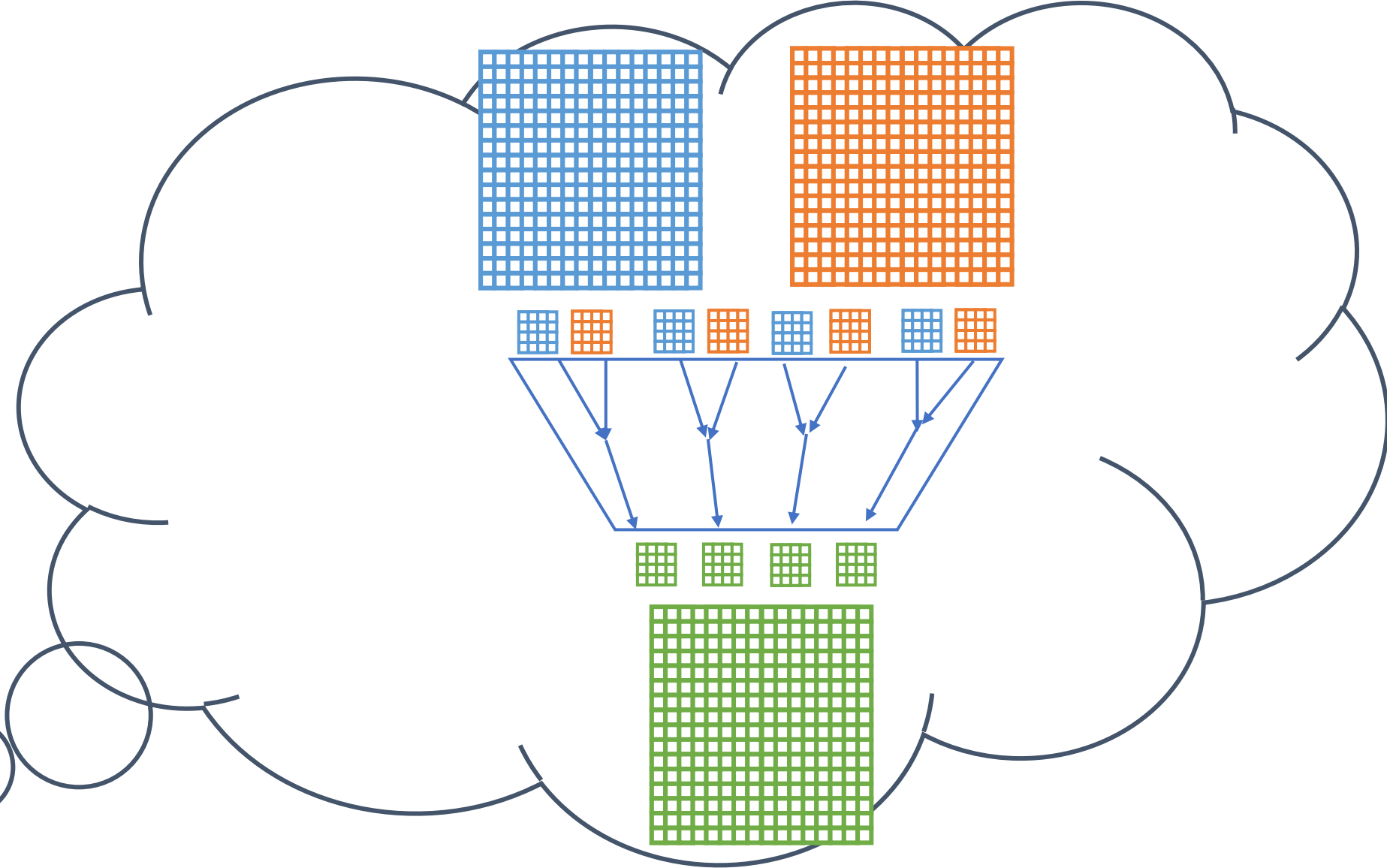
What if I have "economy of
scale?"



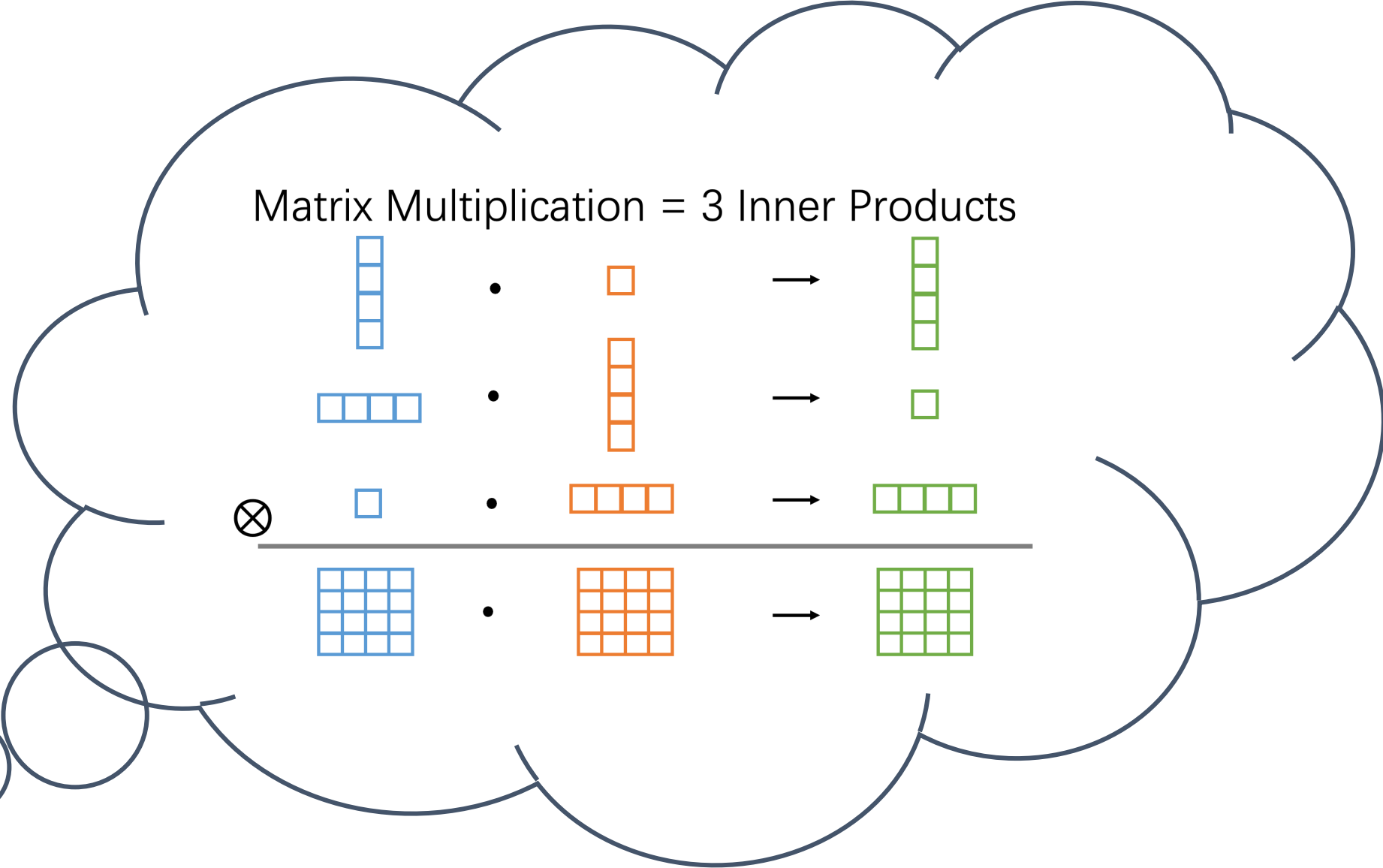
Intuition



Intuition



Intuition



Intuition

$$\text{Let } p = \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} \cdot \begin{array}{|c|} \hline \square \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array}$$

$$q = \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} \cdot \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline \square \\ \hline \end{array}$$

$$r = \begin{array}{|c|} \hline \square \\ \hline \end{array} \cdot \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \end{array}$$

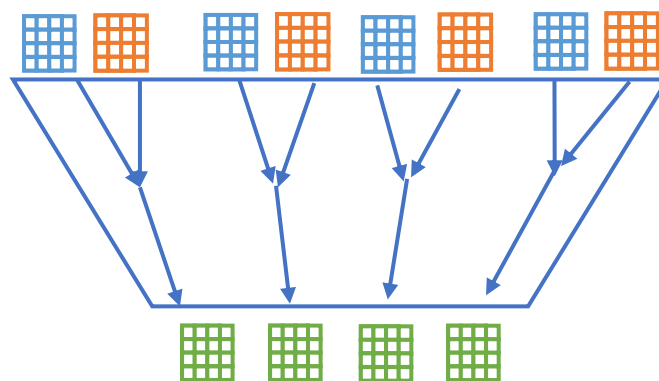
$$\text{Then } m = p q r = \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \cdot \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array}$$



Intuition

Known: $pqr = m$

Task: Get **100** copies of m



Intuition

Known: $pqr = m$

Task: Get 100 copies of m

$$(p + q + r)^3 = 6m + \dots$$



Intuition

Known: $pqr = m$

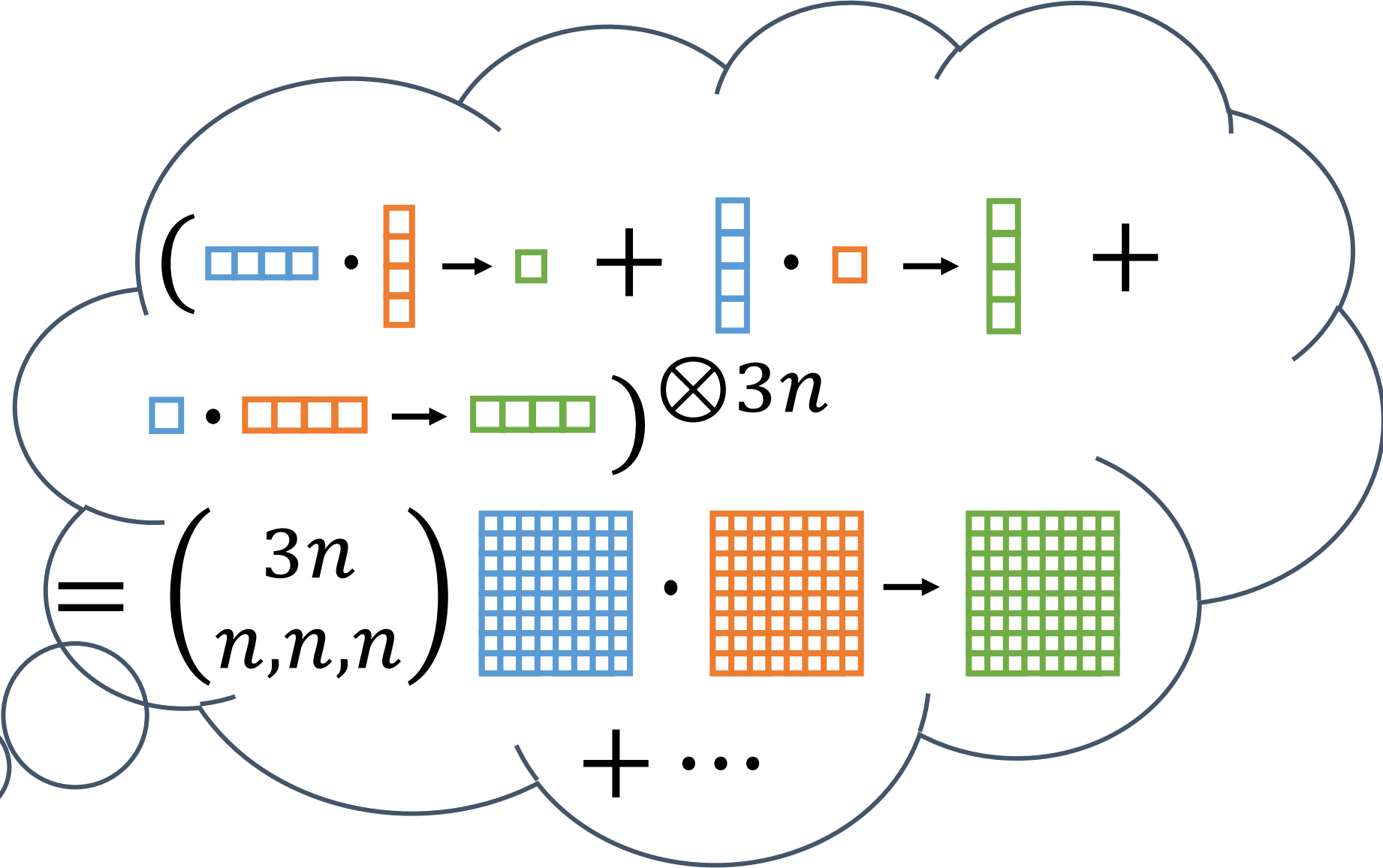
Task: Get 100 copies of m

$$(p + q + r)^3 = 6m + \dots$$

$$(p + q + r)^{3n} = \binom{3n}{n, n, n} m^n + \dots$$



Intuition



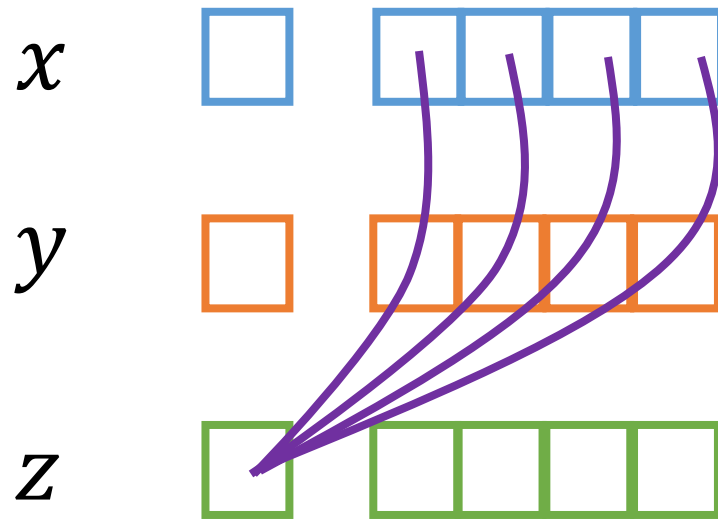
(Small) Coppersmith Winograd Tensor

$$T_{cw} = \sum_{i=1}^q x_i y_i z_0 + x_i y_0 z_i + x_0 y_i z_i$$



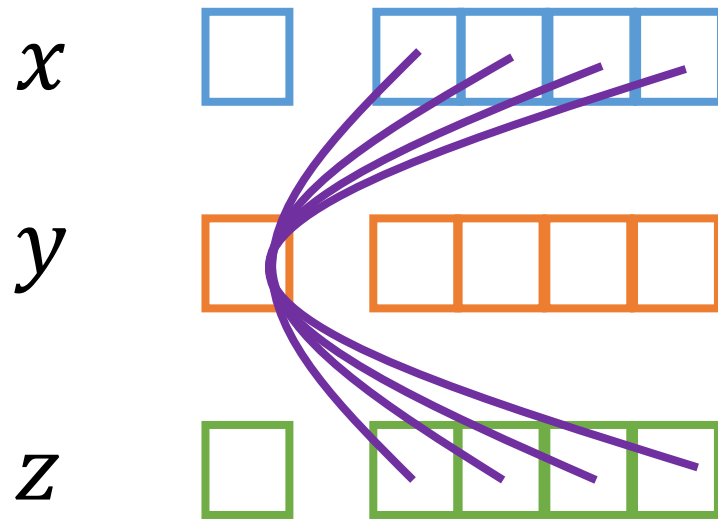
(Small) Coppersmith Winograd Tensor

$$T_{cw} = \sum_{i=1}^q \boxed{x_i y_i z_0} + x_i y_0 z_i + x_0 y_i z_i$$



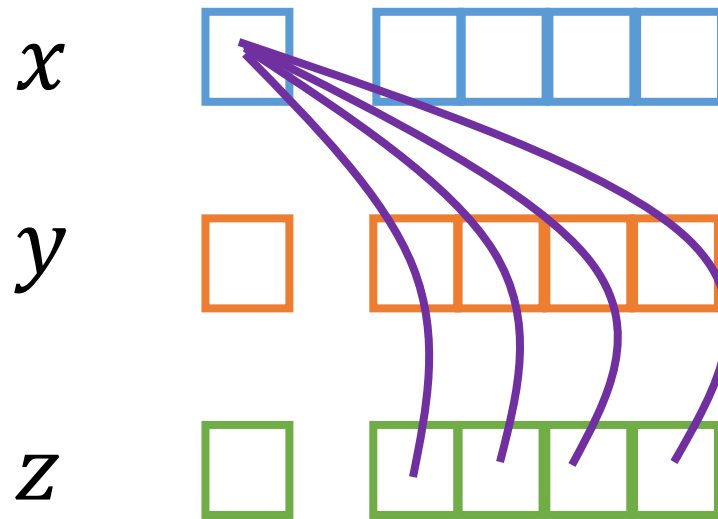
(Small) Coppersmith Winograd Tensor

$$T_{cw} = \sum_{i=1}^q x_i y_i z_0 + \boxed{x_i y_0 z_i} + x_0 y_i z_i$$



(Small) Coppersmith Winograd Tensor

$$T_{cw} = \sum_{i=1}^q x_i y_i z_0 + x_i y_0 z_i + \boxed{x_0 y_i z_i}$$



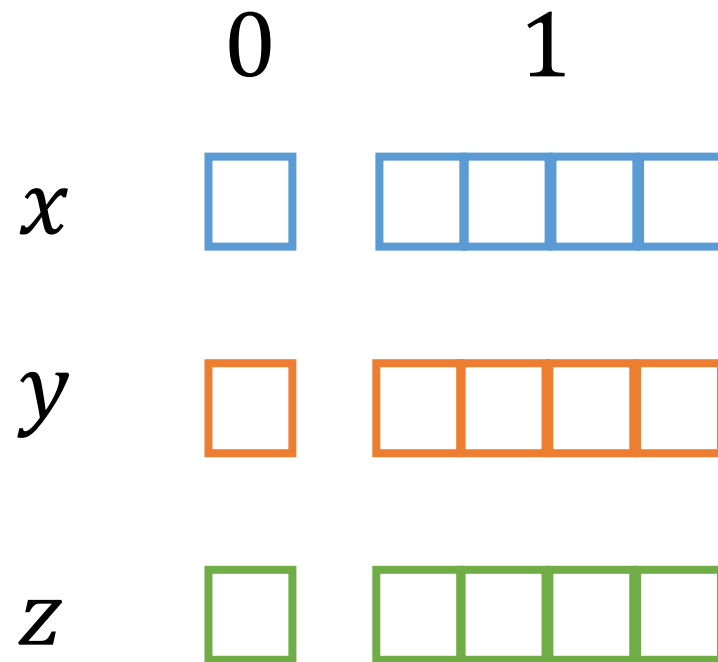
(Small) Coppersmith Winograd Tensor

$$T_{cw} = \sum_{i=1}^q x_i y_i z_0 + x_i y_0 z_i + x_0 y_i z_i$$

$T_{cw}^{\otimes 3n}$ needs $(q + 2)^{3n+o(n)}$
multiplications instead $(3q)^{3n+o(n)}$

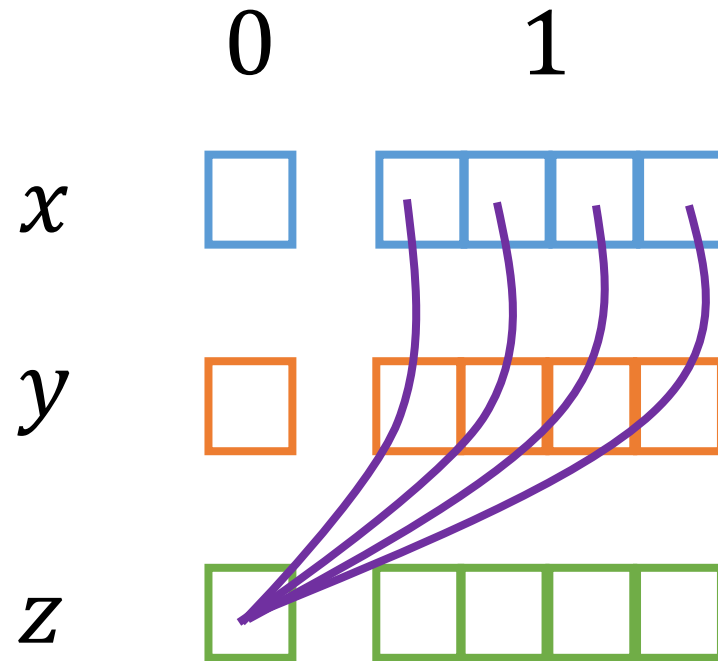
Partition of the variables

$$X_0 = \{x_0\}, \quad X_1 = \{x_1, x_2, \dots, x_q\}$$



Partition of the variables

$$X_0 = \{x_0\}, \quad X_1 = \{x_1, x_2, \dots, x_q\}$$

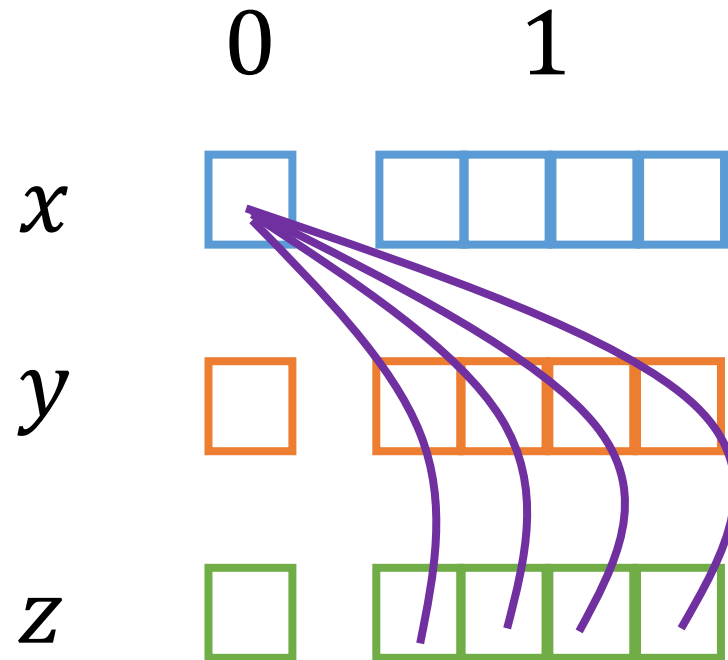


$$T_{110} = \sum_{i=1}^q x_i y_i z_0$$

1 + 1 + 0

Partition of the variables

$$X_0 = \{x_0\}, \quad X_1 = \{x_1, x_2, \dots, x_q\}$$

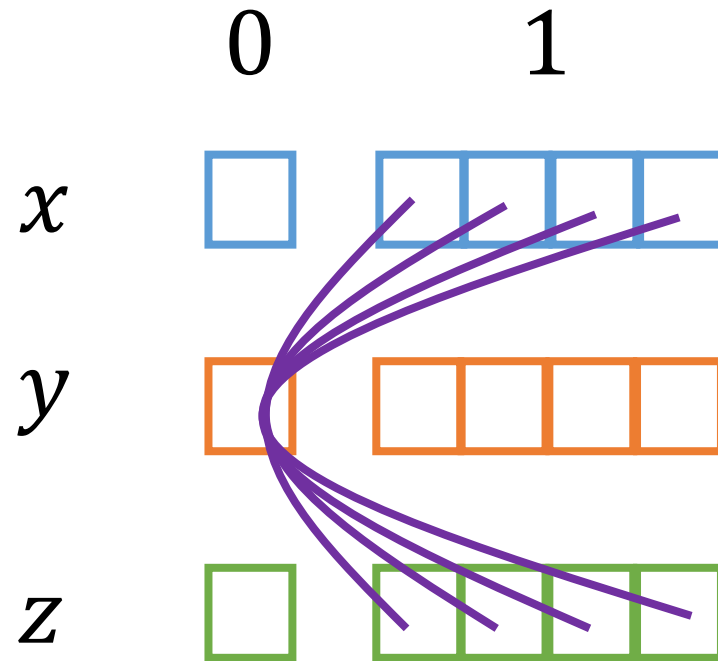


$$T_{011} = \sum_{i=1}^q x_0 y_i z_i$$

0 + 1 + 1

Partition of the variables

$$X_0 = \{x_0\}, \quad X_1 = \{x_1, x_2, \dots, x_q\}$$



$$T_{101} = \sum_{i=1}^q x_i y_0 z_i$$

1 + 0 + 1

Partition of the variables

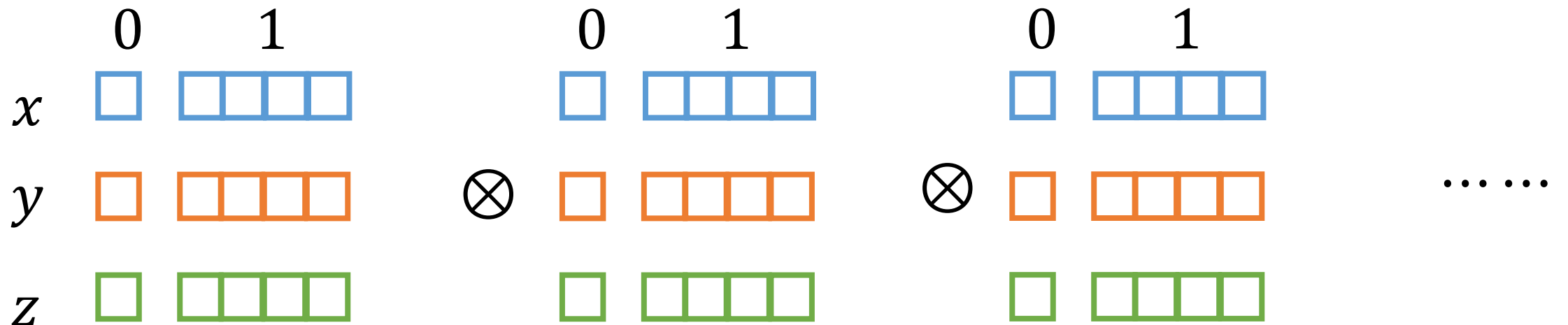
$$X_0 = \{x_0\}, \quad X_1 = \{x_1, x_2, \dots, x_q\}$$

$$T_{110}, T_{101}, T_{011}$$

There is an (hyper)edge between X_i, Y_j, Z_k only if $i + j + k = 2$

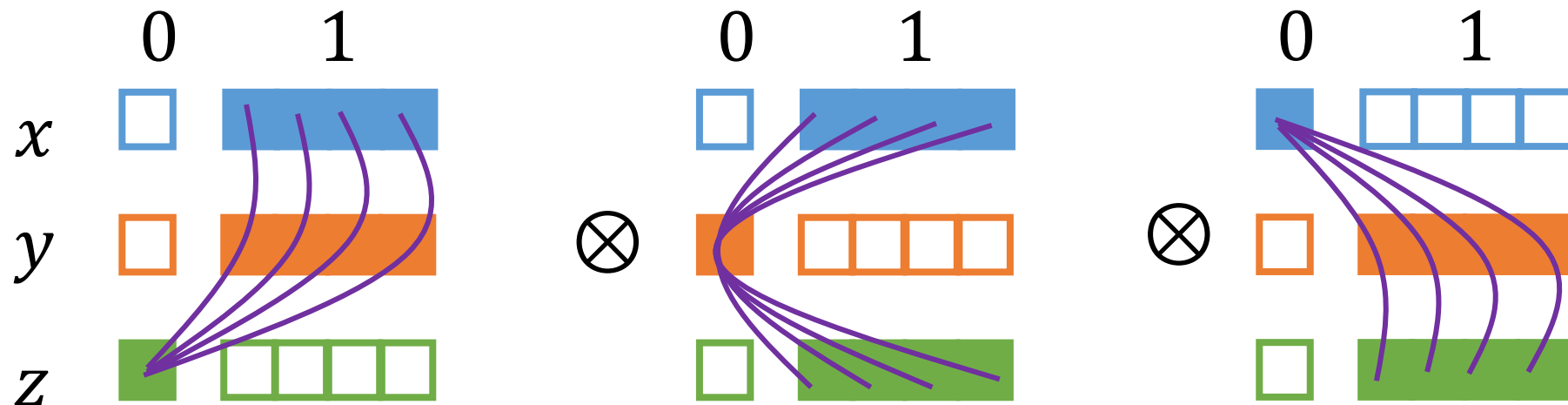
Taking n -th power

$$T_{cw}^{\otimes n} = (T_{110} + T_{101} + T_{011}) \otimes (T_{110} + T_{101} + T_{011}) \otimes \dots$$



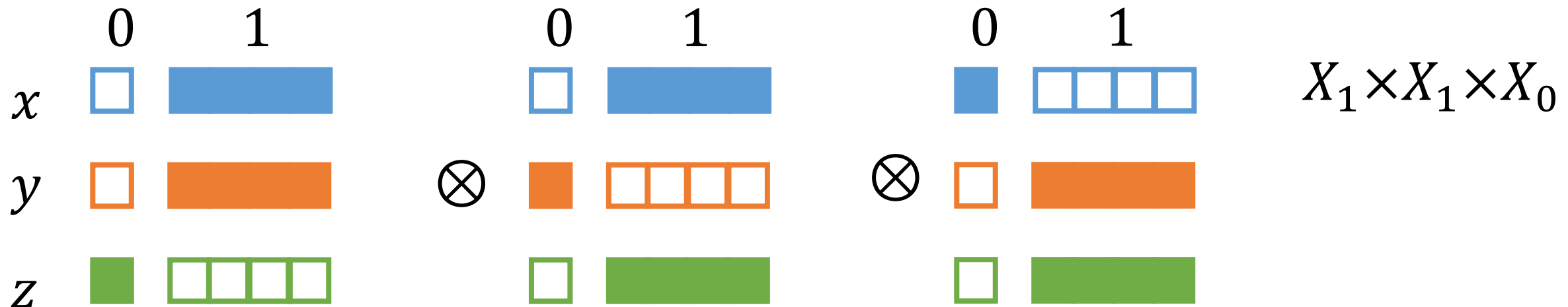
Taking n -th power

$$T_{cw}^{\otimes n} = (T_{110} + T_{101} + T_{011}) \otimes (T_{110} + T_{101} + T_{011}) \otimes \dots$$



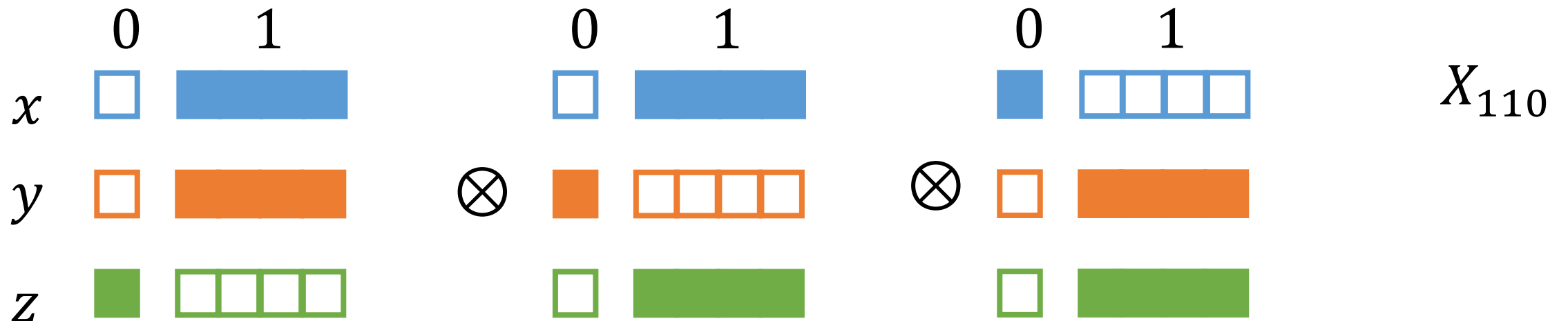
Taking n -th power

$$T_{cw}^{\otimes n} = (T_{110} + T_{101} + T_{011}) \otimes (T_{110} + T_{101} + T_{011}) \otimes \dots$$



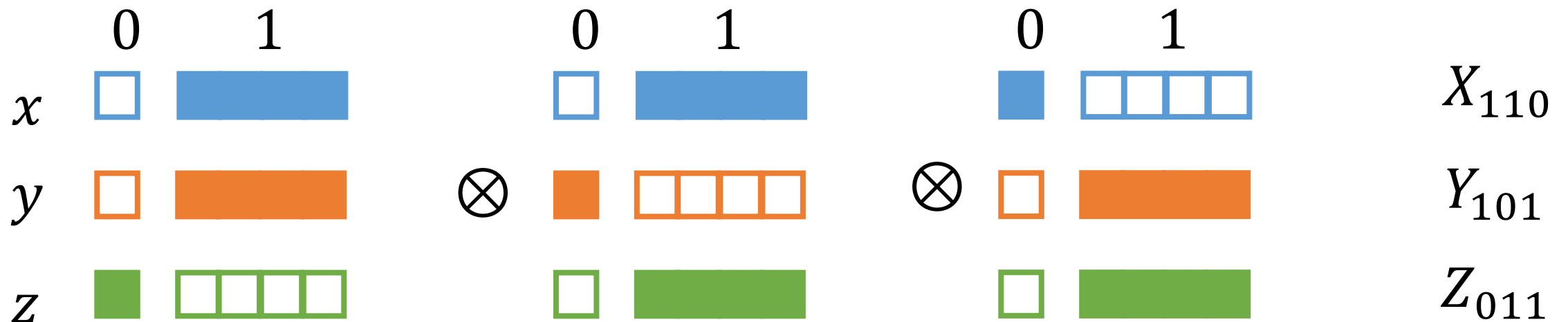
Taking n -th power

$$T_{cw}^{\otimes n} = (T_{110} + T_{101} + T_{011}) \otimes (T_{110} + T_{101} + T_{011}) \otimes \dots$$



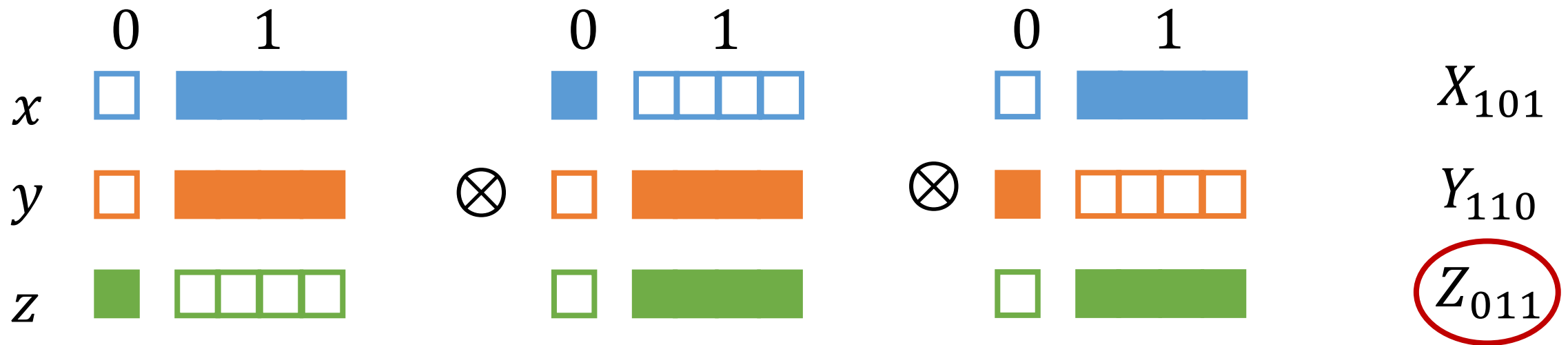
Taking n -th power

$$T_{cw}^{\otimes n} = (T_{110} + T_{101} + T_{011}) \otimes (T_{110} + T_{101} + T_{011}) \otimes \dots$$

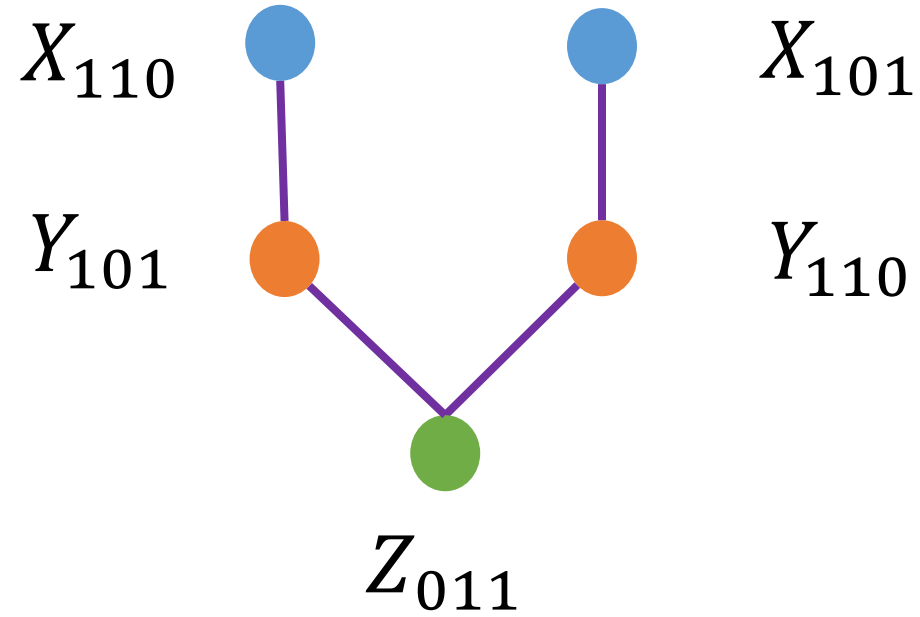


Taking n -th power

$$T_{cw}^{\otimes n} = (T_{110} + T_{101} + T_{011}) \otimes (T_{110} + T_{101} + T_{011}) \otimes (T_{110} + T_{101} + T_{011}) \dots$$



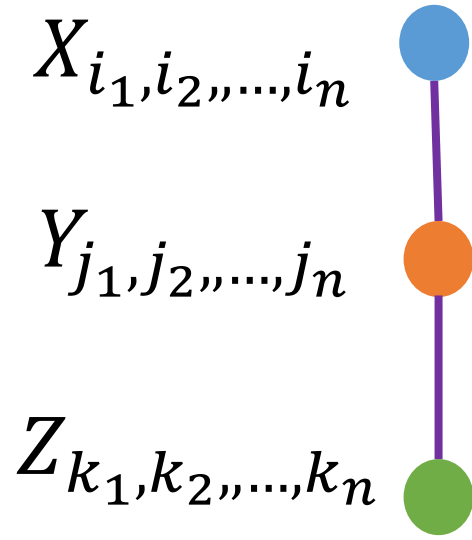
Conflict



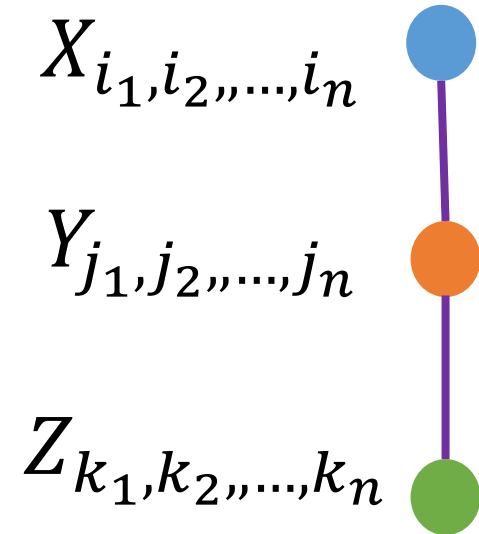
$$C_{011} = A_{110} \cdot B_{101} + A_{101} \cdot B_{110}$$

Garbage output

In general

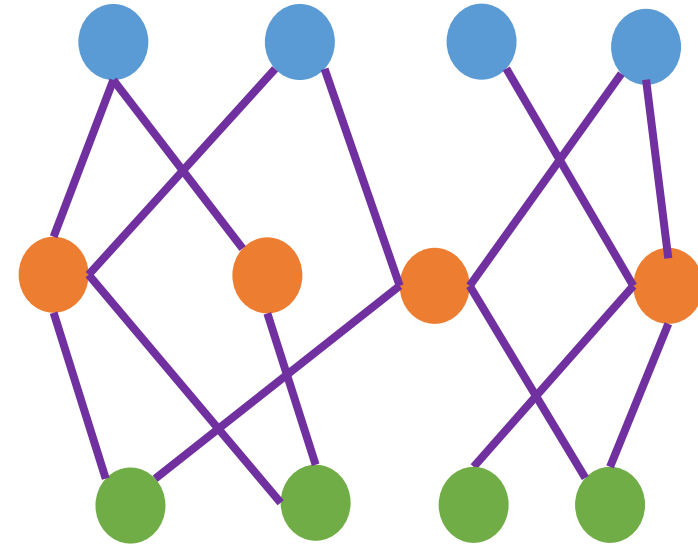
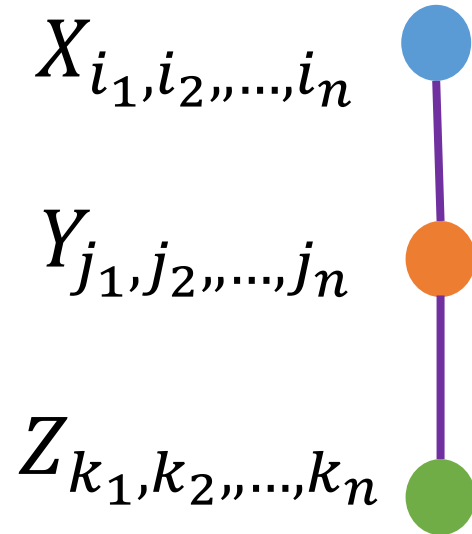


In general



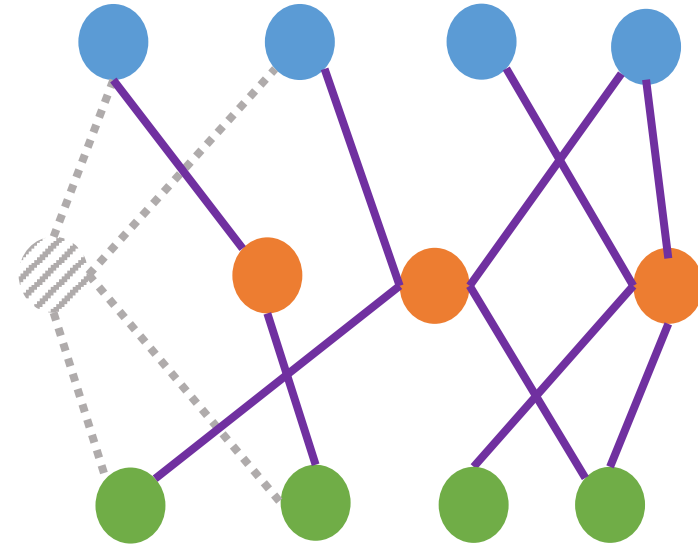
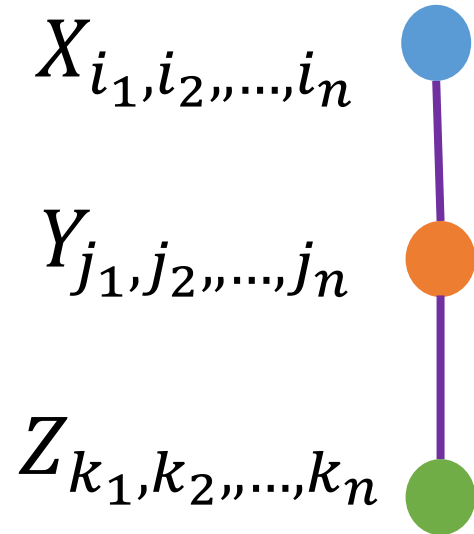
There is an (hyper)edge only if $i + j + k = 22222222 \dots$

In general



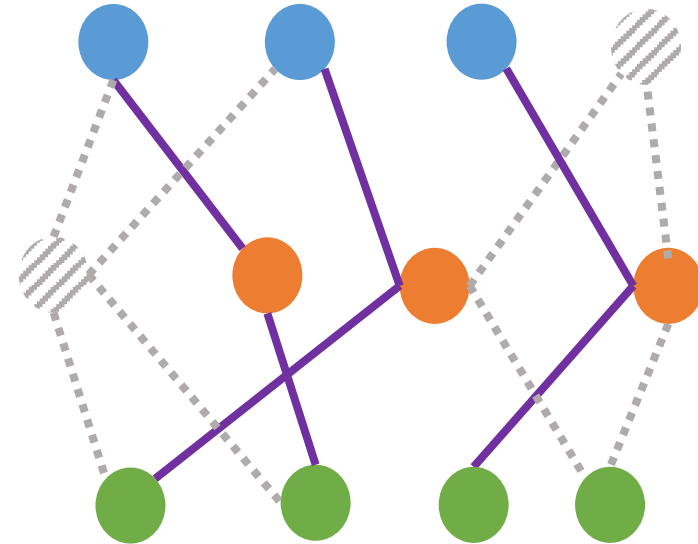
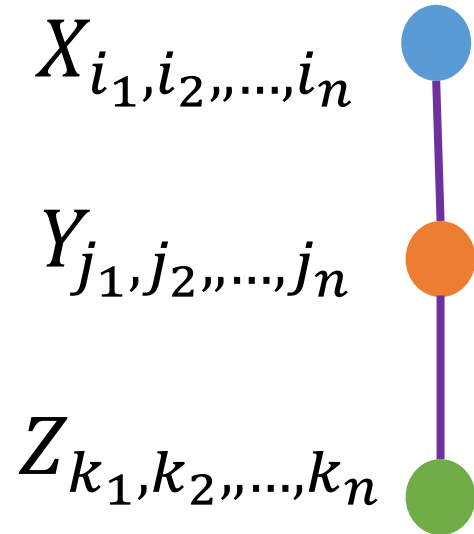
There is an (hyper)edge only if $i + j + k = 22222222 \dots$

Zero Out



There is an (hyper)edge only if $i + j + k = 22222222 \dots$

Zero Out



There is an (hyper)edge only if $i + j + k = 22222222 \dots$

Strassen's Laser Method

Main idea: Take a cheap tensor T . Show that it is useful for MM.

Cheap. There is a non-trivial algorithm for T .

Useful. By zeroing-out variables of T , turn it into a disjoint union of MM tensors.



Strassen's Laser Method

Main idea: Take a cheap tensor T . Show that it is useful for MM.

Cheap. There is a non-trivial algorithm for T .

Useful. By zeroing-out variables of T , turn it into a disjoint union of MM tensors.



Strassen's Laser Method

Main idea: Take a cheap tensor T . Show that it is useful for MM.

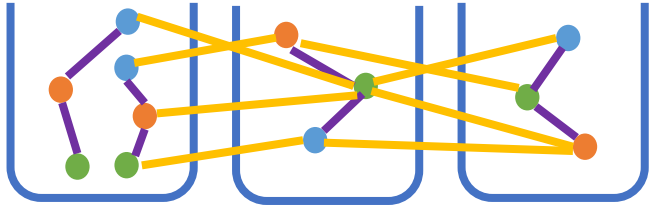
Cheap. There is a non-trivial algorithm for T .

Useful. By zeroing-out variables of T , turn it into a disjoint union of MM tensors.



Hashing

[Coppersmith, Winograd 90]



X_{i_1, i_2, \dots, i_n}



Y_{j_1, j_2, \dots, j_n}



Z_{k_1, k_2, \dots, k_n}



$h_X(i) \in [0, p)$

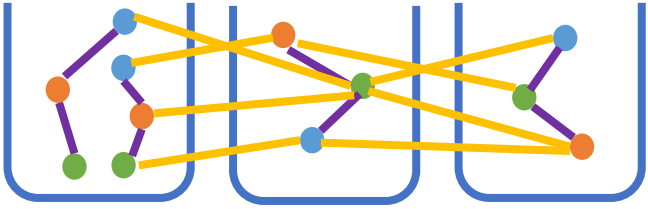
$h_Y(j) \in [0, p)$

$h_Z(k) \in [0, p)$

There is an (hyper)edge only if $h_X(i) + h_Z(k) = 2h_Y(j)$

Hashing

[Coppersmith, Winograd 90]



$$h_X(i) = \langle w, i \rangle \bmod p$$

$$h_Z(k) = (\langle w, k \rangle + b) \bmod p$$

$$h_Y(j) = (\langle w, 222 \dots - j \rangle + b) / 2 \bmod p$$

X_{i_1, i_2, \dots, i_n}



Y_{j_1, j_2, \dots, j_n}



Z_{k_1, k_2, \dots, k_n}



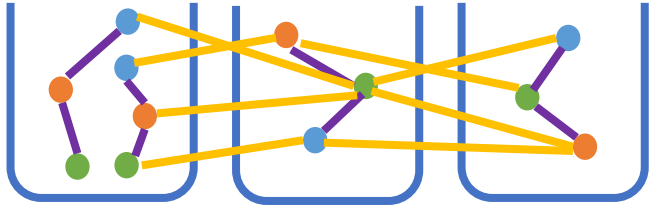
$$h_X(i) \in [0, p)$$

$$h_Y(j) \in [0, p)$$

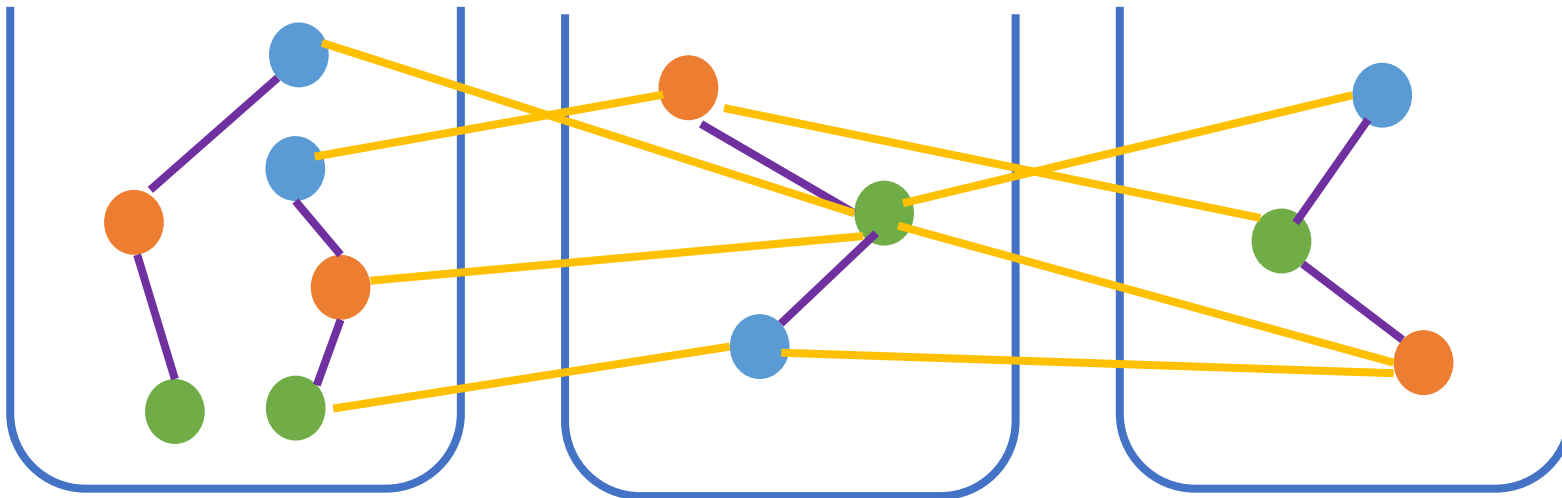
$$h_Z(k) \in [0, p)$$

There is an (hyper)edge only if $h_X(i) + h_Z(k) = 2h_Y(j)$

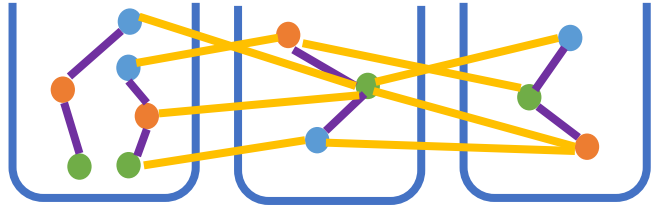
Hashing
[Coppersmith, Winograd 90]



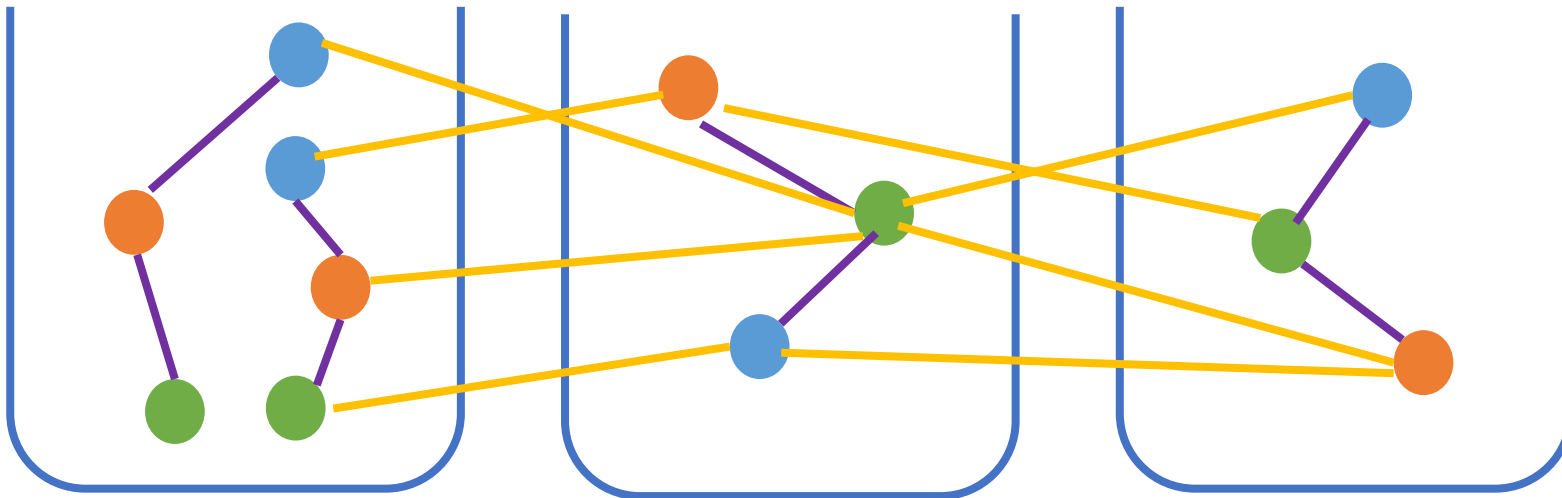
- Throw nodes into p buckets.



Hashing
[Coppersmith, Winograd 90]

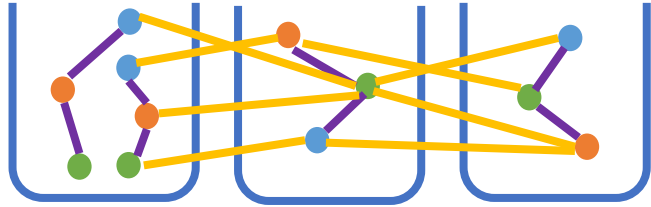


- Throw nodes into p buckets.
- Pick p so that each node has degree 1 within its bucket.

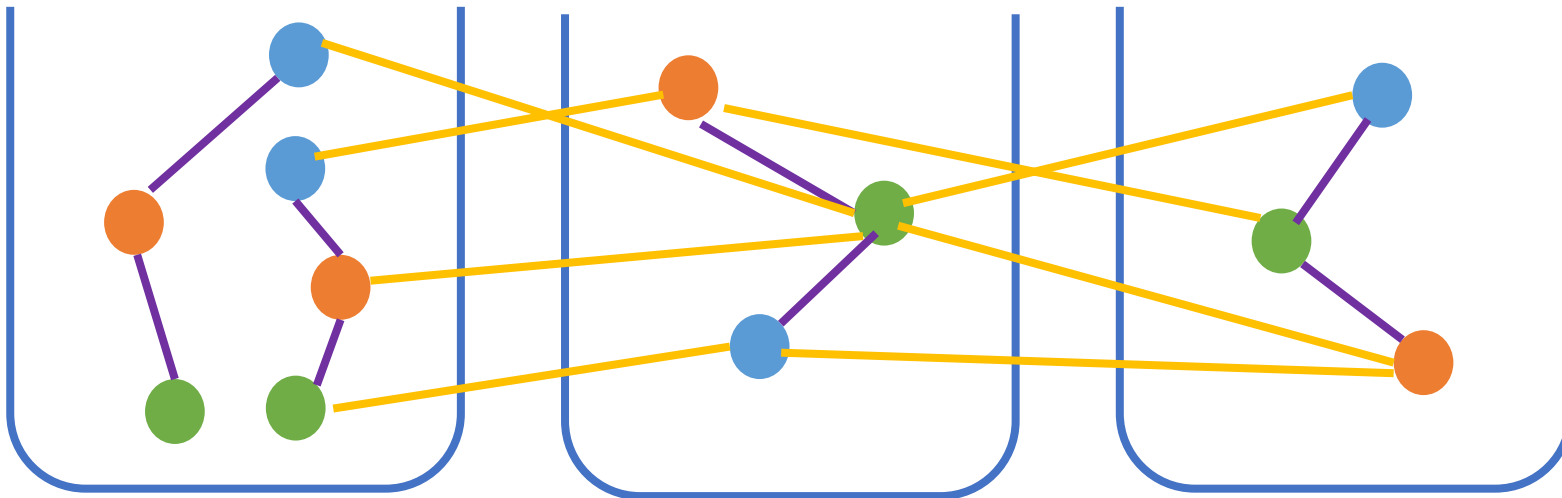


.....

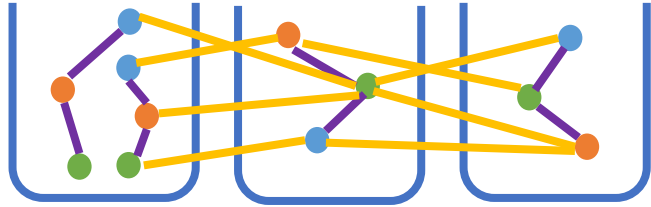
Hashing
[Coppersmith, Winograd 90]



- Triples we want
- Interfering cross terms

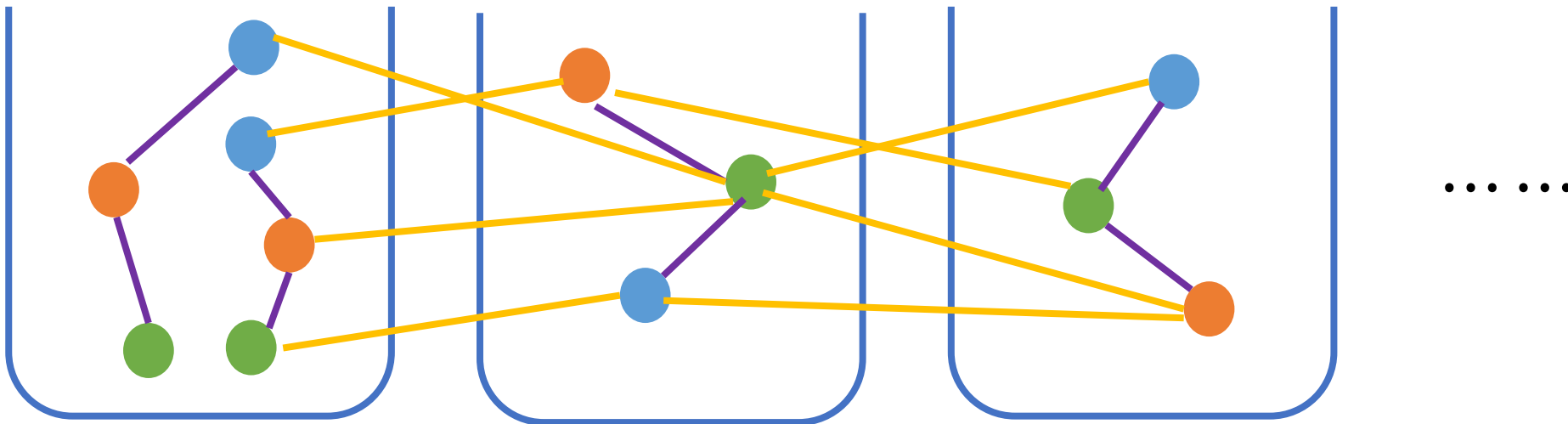


Hashing
[Coppersmith, Winograd 90]

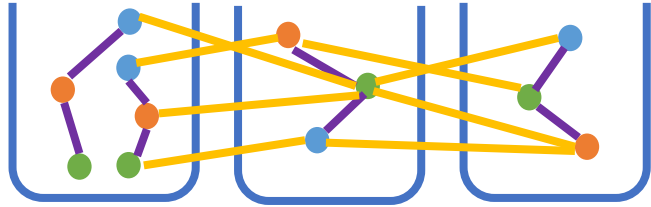


- Triples we want
- Interfering cross terms

There is an (hyper)edge only if $h_X(i) + h_Z(k) = 2h_Y(j)$

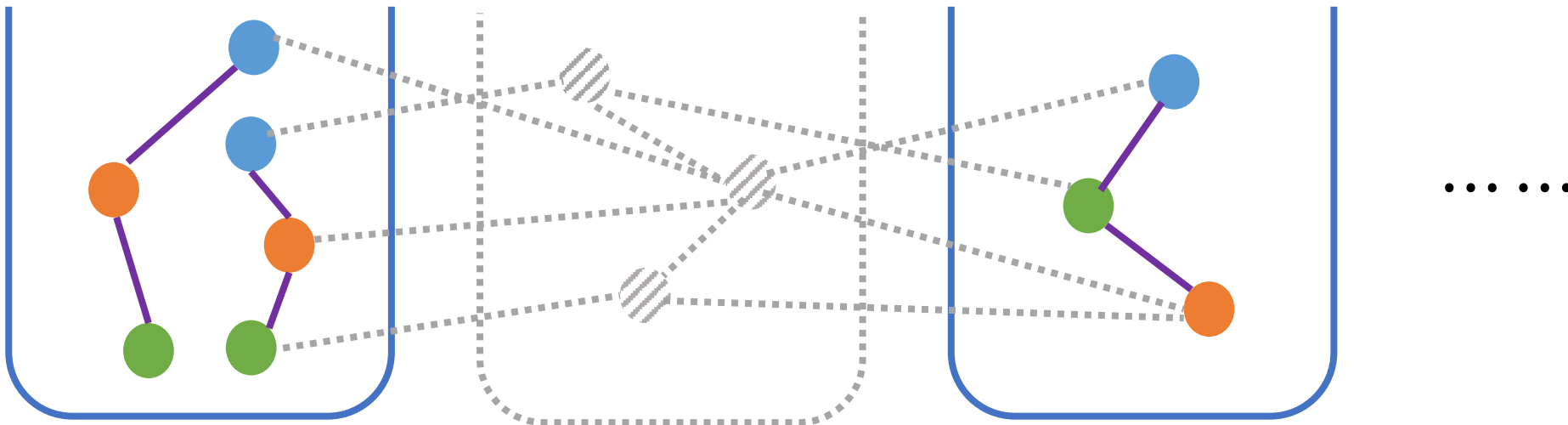


Hashing
[Coppersmith, Winograd 90]

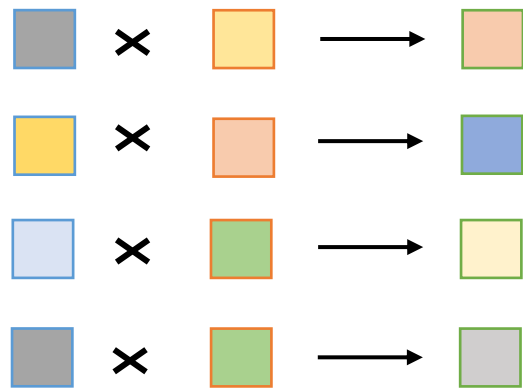


- Take a Salem-Spencer set of size $p^{1-o(1)}$. Zero-out the rest.

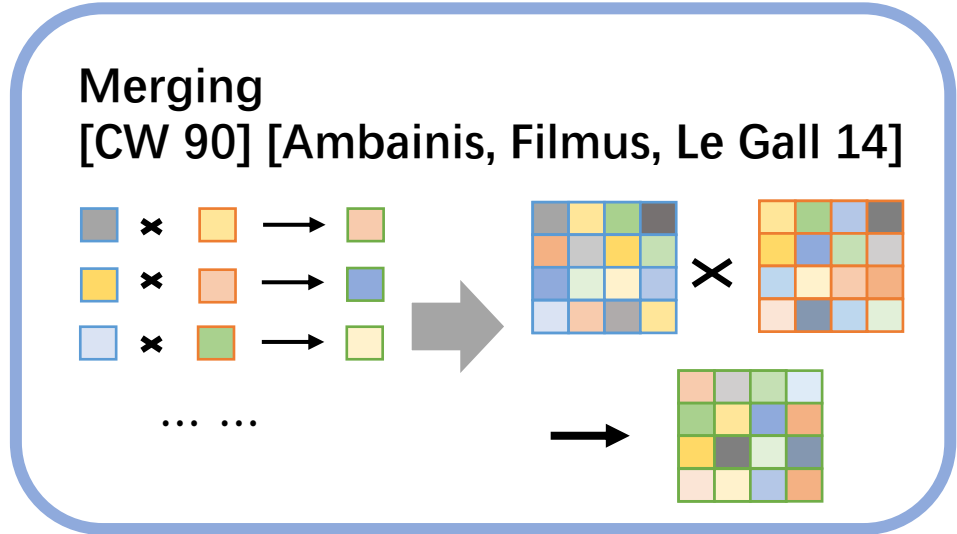
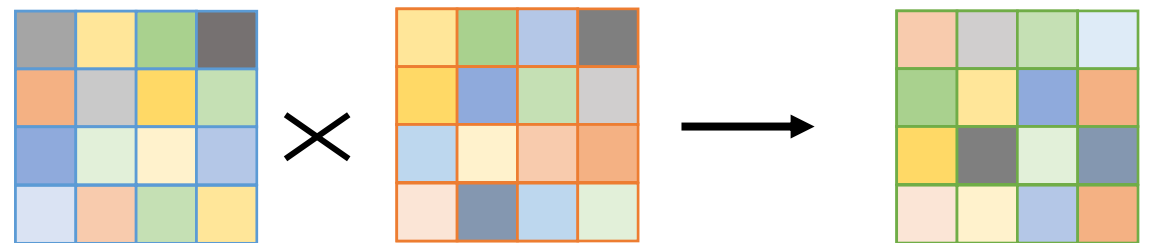
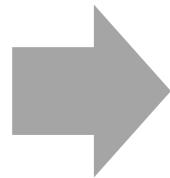
Salem-Spencer Set: A subset of $[p - 1]$ that has no arithmetic progressions.

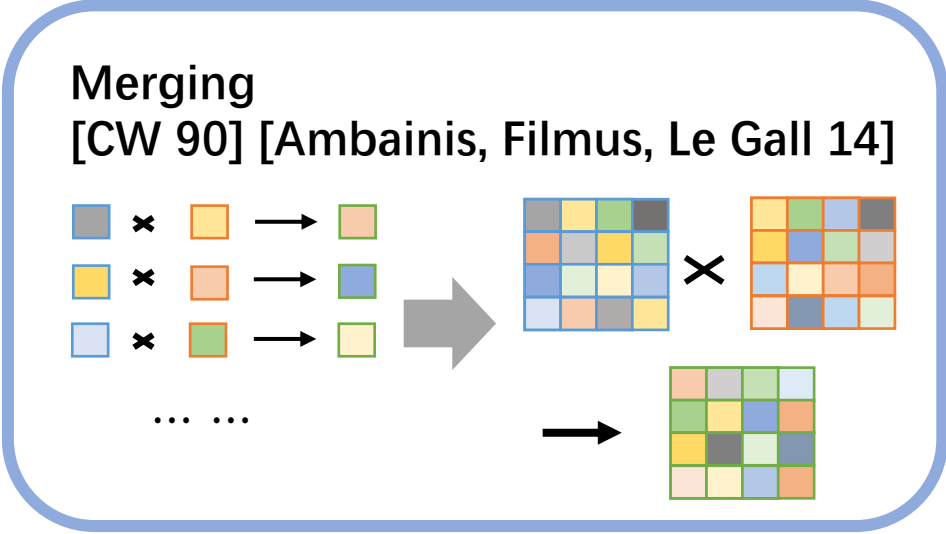
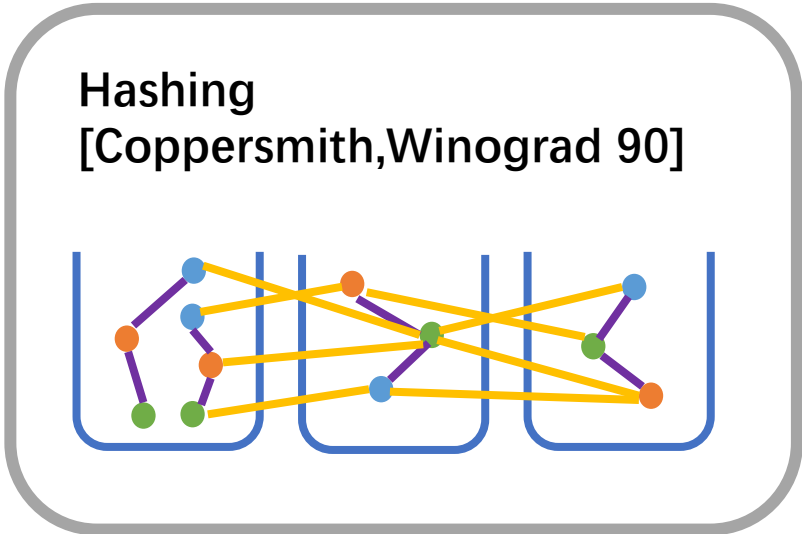


- Sometimes conflicting matrix multiplications could be all kept.



... ..





- Randomness destroy structure.

$$X_{i_1, i_2, \dots, i_n}$$

$$\tilde{X}_{i_1+i_2, i_3+i_4, \dots, i_{n-1}+i_n}$$

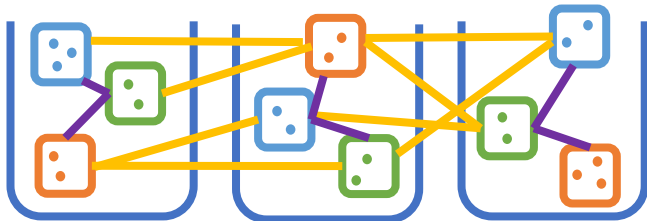
$$Y_{j_1, j_2, \dots, j_n} \in$$

$$\tilde{Y}_{j_1+j_2, j_3+j_4, \dots, j_{n-1}+j_n}$$

$$Z_{k_1, k_2, \dots, k_n}$$

$$\tilde{Z}_{k_1+k_2, k_3+k_4, \dots, k_{n-1}+k_n}$$

2/4/8/16/32-th Power
[CW 90][Stothers 10]
[Williams 13] [Le Gall 14]



$$X_{i_1, i_2, \dots, i_n}$$

$$\tilde{X}_{i_1+i_2, i_3+i_4, \dots, i_{n-1}+i_n}$$

$$Y_{j_1, j_2, \dots, j_n}$$

∈

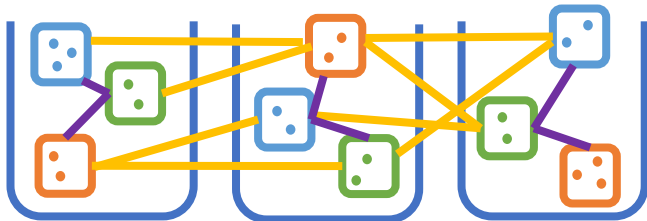
$$\tilde{Y}_{j_1+j_2, j_3+j_4, \dots, j_{n-1}+j_n}$$

$$Z_{k_1, k_2, \dots, k_n}$$

$$\tilde{Z}_{k_1+k_2, k_3+k_4, \dots, k_{n-1}+k_n}$$

There is an (hyper)edge iff $\tilde{i} + \tilde{j} + \tilde{k} = 44444444 \dots$

2/4/8/16/32-th Power
[CW 90][Stothers 10]
[Williams 13] [Le Gall 14]



$$X_{i_1, i_2, \dots, i_n}$$

$$\tilde{X}_{i_1+i_2, i_3+i_4, \dots, i_{n-1}+i_n}$$

$$Y_{j_1, j_2, \dots, j_n} \in$$

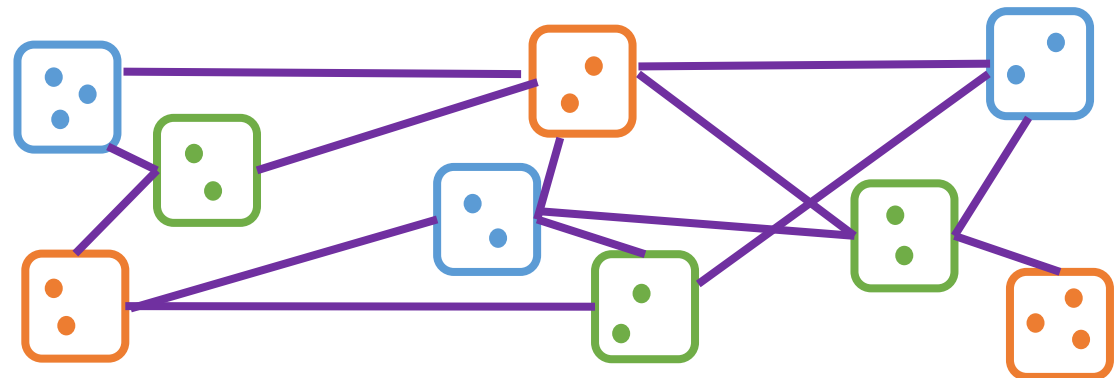
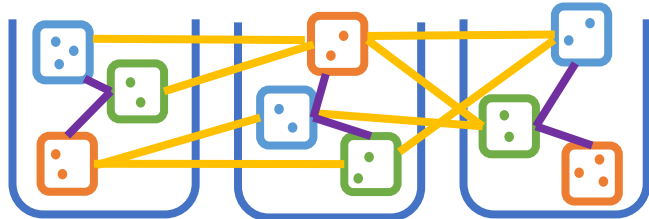
$$\tilde{Y}_{j_1+j_2, j_3+j_4, \dots, j_{n-1}+j_n}$$

$$Z_{k_1, k_2, \dots, k_n}$$

$$\tilde{Z}_{k_1+k_2, k_3+k_4, \dots, k_{n-1}+k_n}$$

There is an (hyper)edge iff $\tilde{i} + \tilde{j} + \tilde{k} = 44444444 \dots$

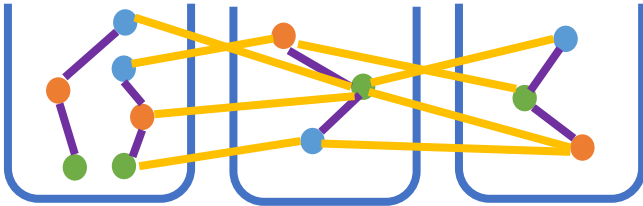
2/4/8/16/32-th Power
 [CW 90][Stothers 10]
 [Williams 13] [Le Gall 14]



- Apply Hashing and zeroing out.

Hashing

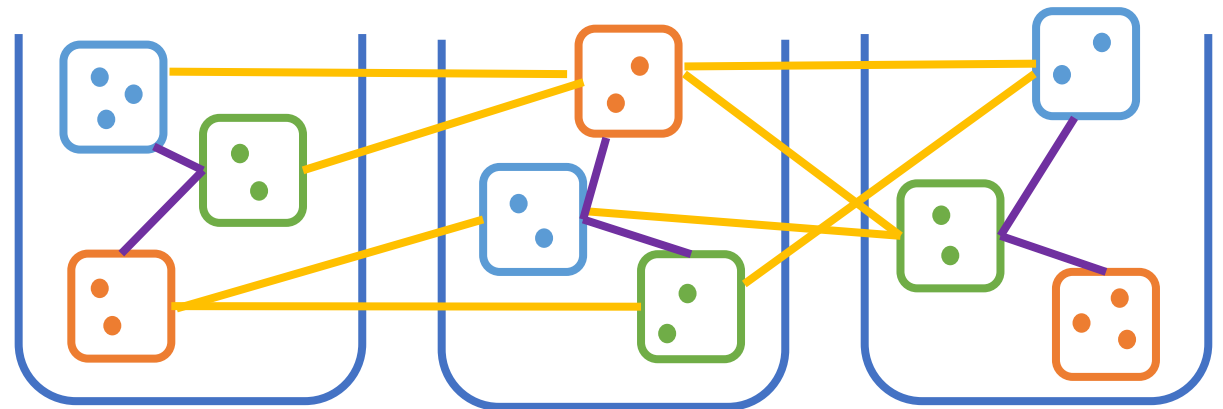
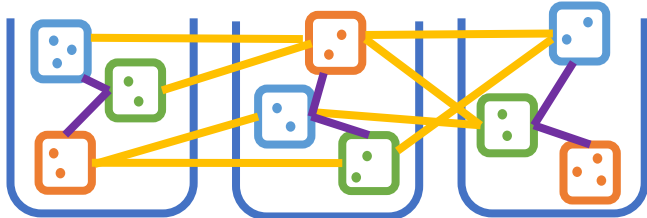
[Coppersmith, Winograd 90]



2/4/8/16/32-th Power

[CW 90][Stothers 10]

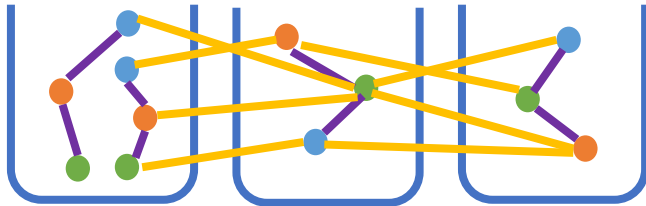
[Williams 13] [Le Gall 14]



- Apply Hashing and zeroing out.

Hashing

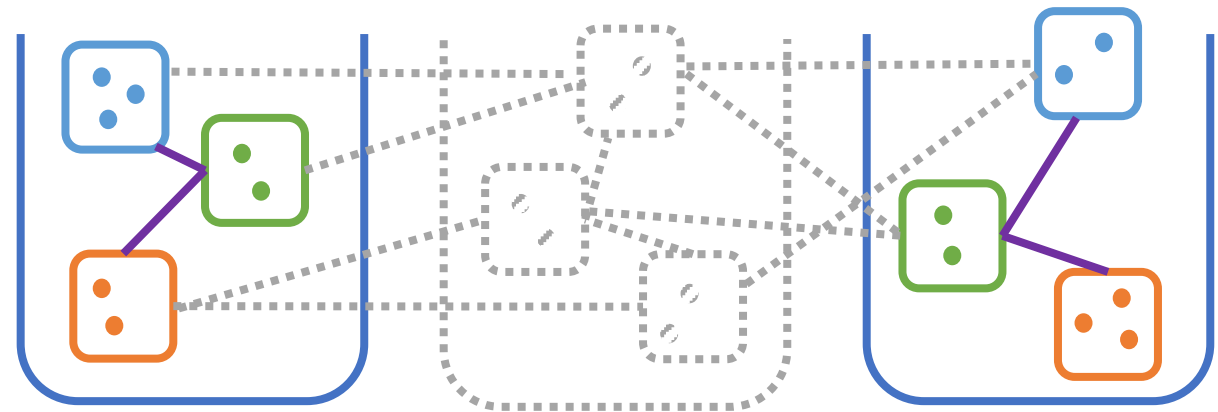
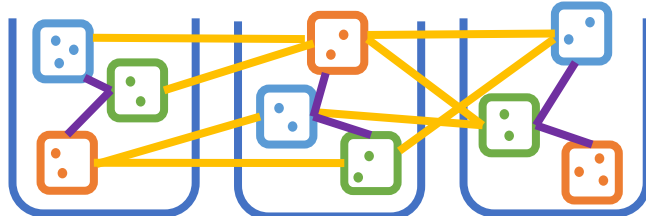
[Coppersmith, Winograd 90]



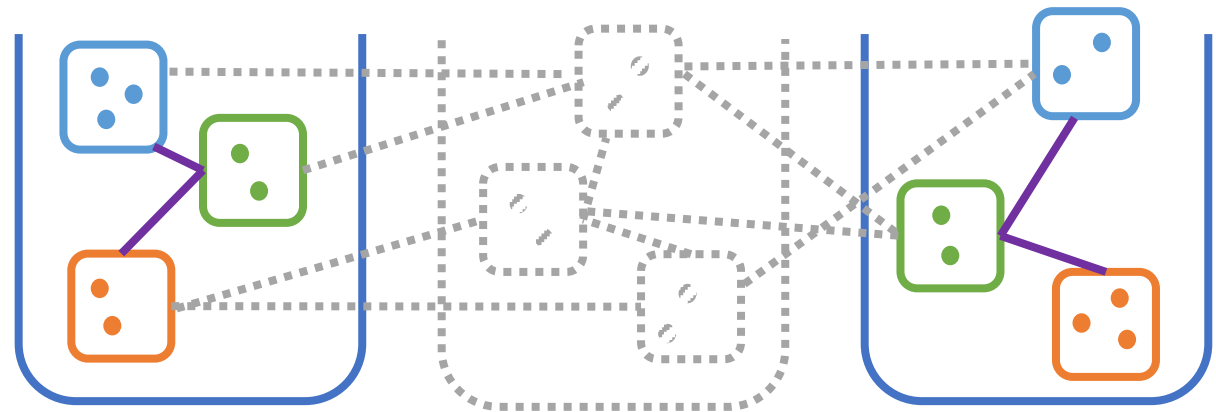
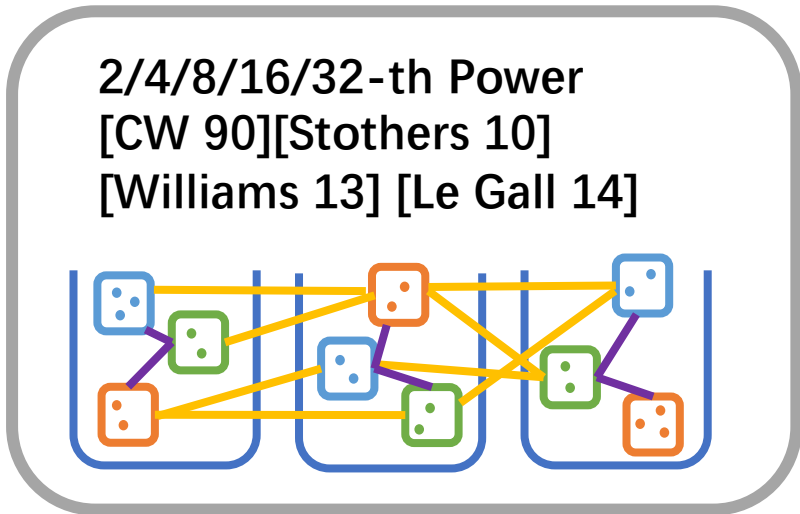
2/4/8/16/32-th Power

[CW 90][Stothers 10]

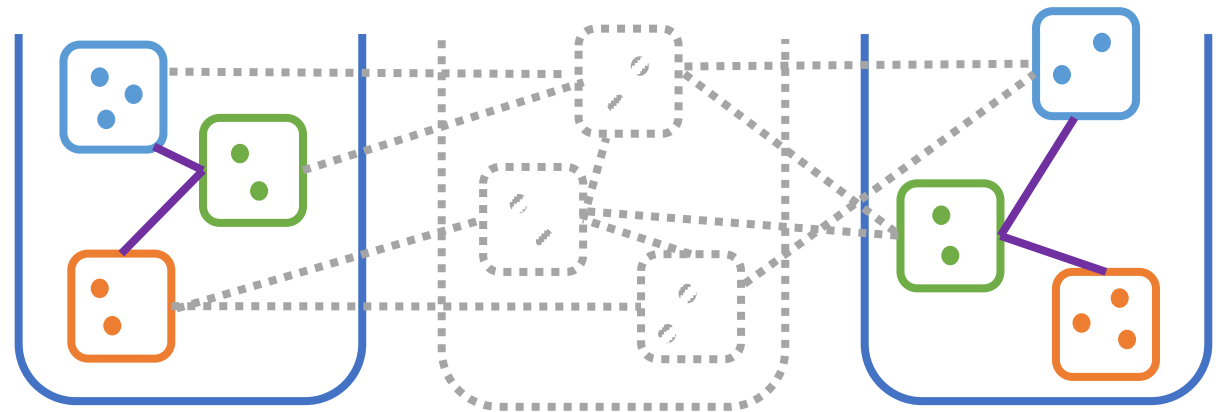
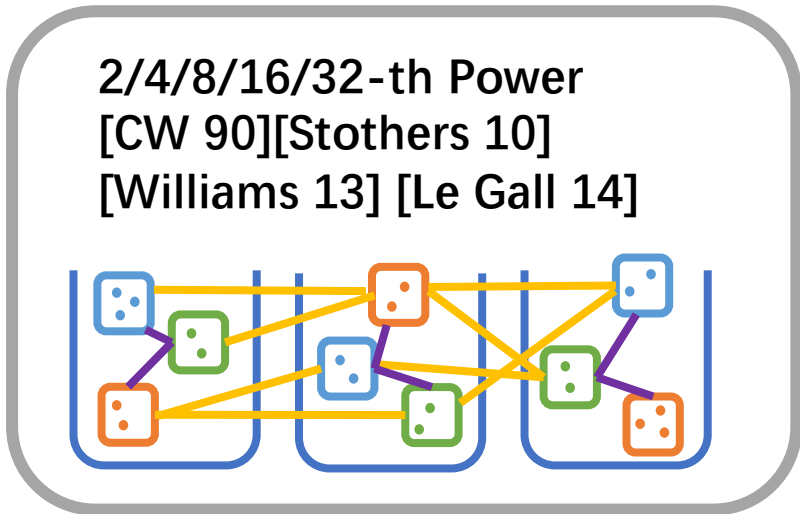
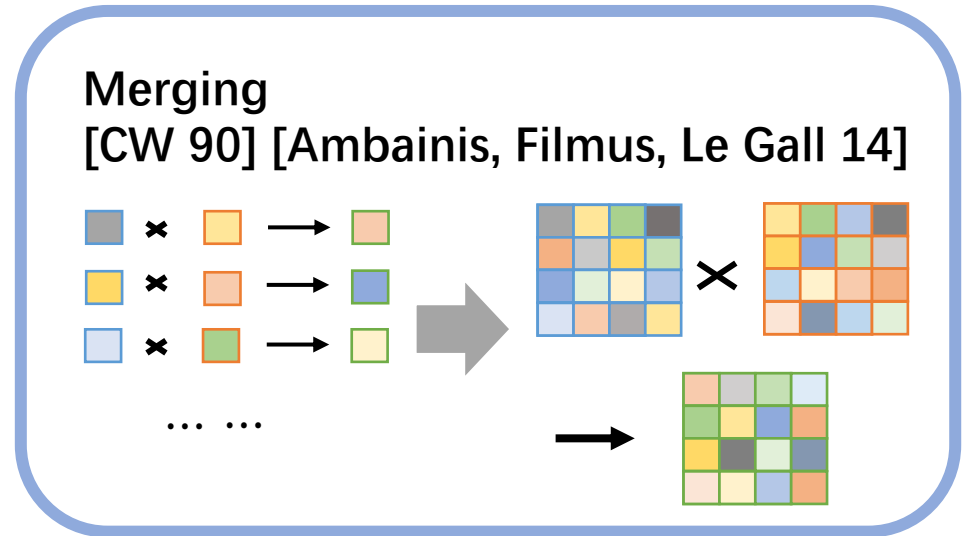
[Williams 13] [Le Gall 14]



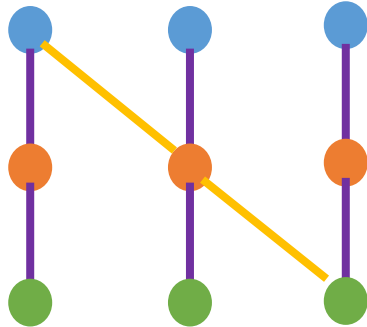
- Apply Hashing and zeroing out.
- Within each triple the structure is preserved.



- Apply Hashing and zeroing out.
- Within each triple the structure is preserved.
 - Apply merging.

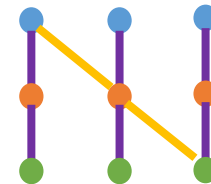


Correlated Zero-out



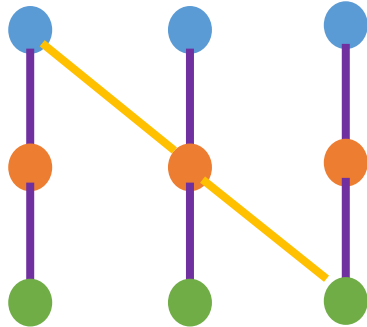
- In higher order, sometimes, arithmetic-progression based zeroing-out fails.

Refined Laser Method
[Alman, Williams 21]



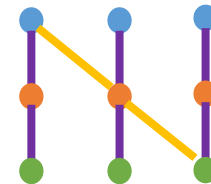
p vs p^3

Correlated Zero-out



- If independently zero out each vertex w.p. $1 - p$.
- Each triple is kept w.p. p^3 .

Refined Laser Method
[Alman, Williams 21]



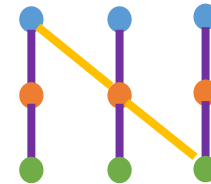
p vs p^3

Correlated Zero-out



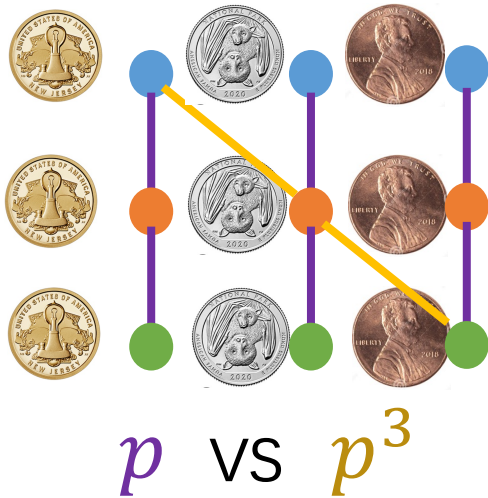
- If independently zero out each vertex w.p. $1 - p$.
- Each triple is kept w.p. p^3 .

Refined Laser Method
[Alman, Williams 21]



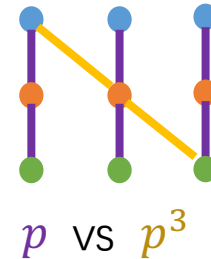
p vs p^3

Correlated Zero-out

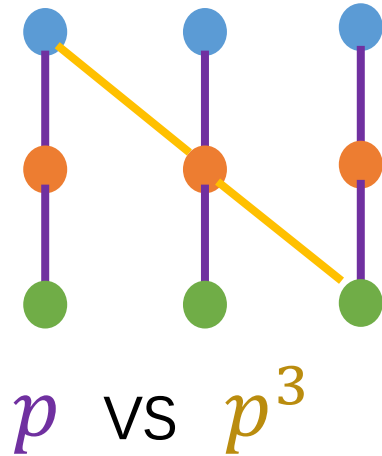


- If we use same randomness for every desired triple.
- Each **desired triple** is kept w.p. p .
- Each **cross term** is kept w.p. p^3

Refined Laser Method
[Alman, Williams 21]

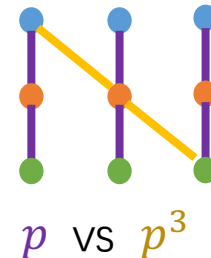


Correlated Zero-out



- If we use same randomness for every desired triple.
 - Each **desired triple** is kept w.p. p .
 - Each **cross term** is kept w.p. p^3

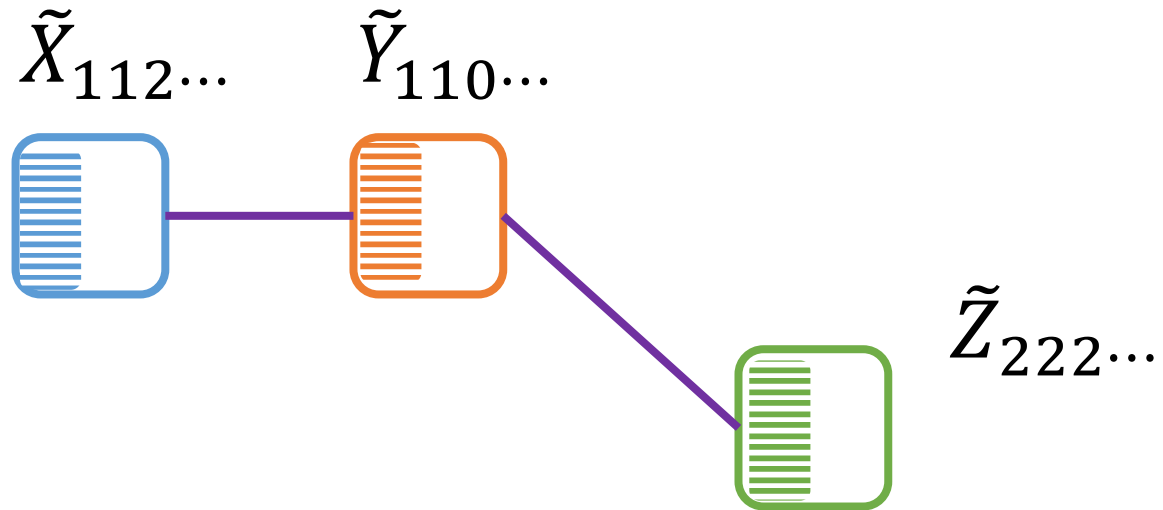
Refined Laser Method
[Alman, Williams 21]



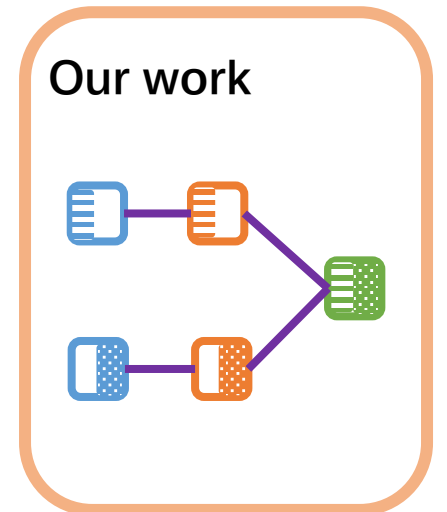
Our Plan

1. Tensor Formulation
2. CW Algorithm From scratch
- 3. Main idea of our improvement**

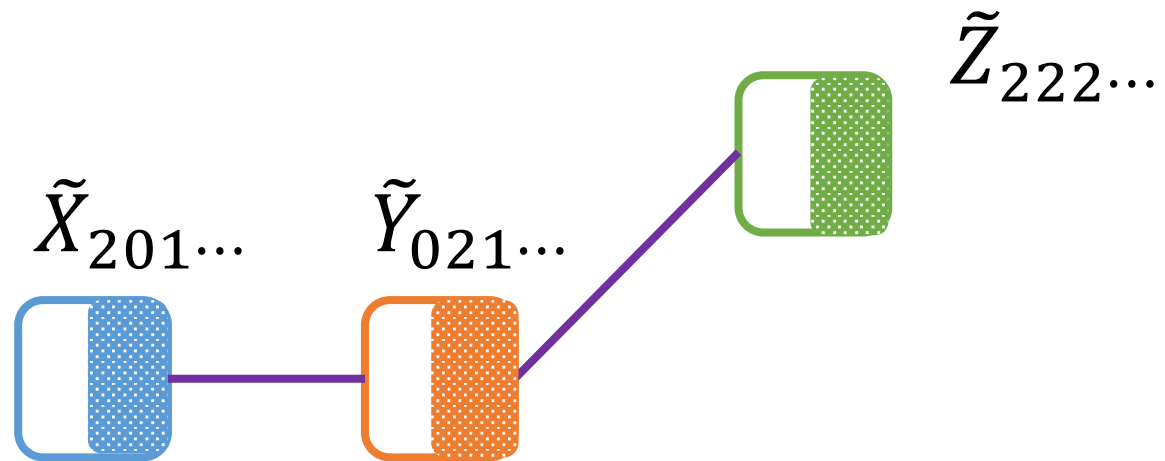
Combination Loss



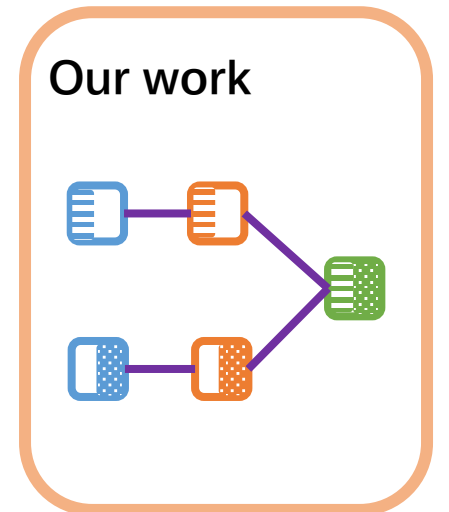
- Structure depends on the matching.



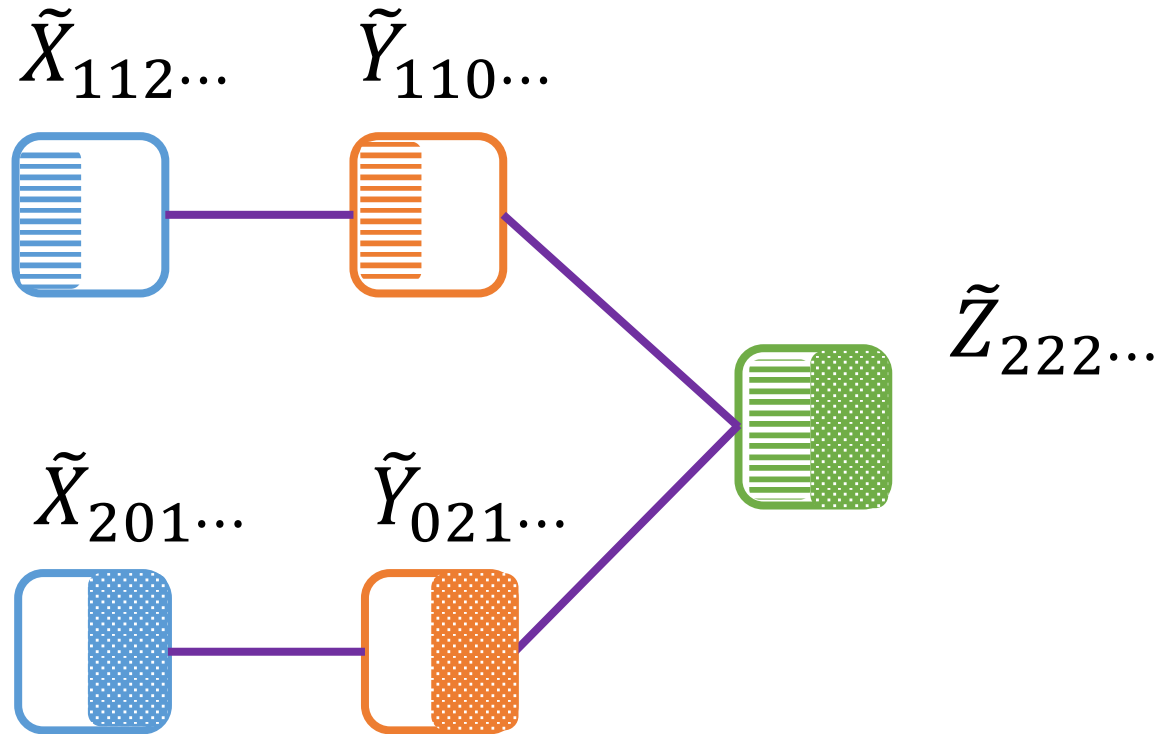
Combination Loss



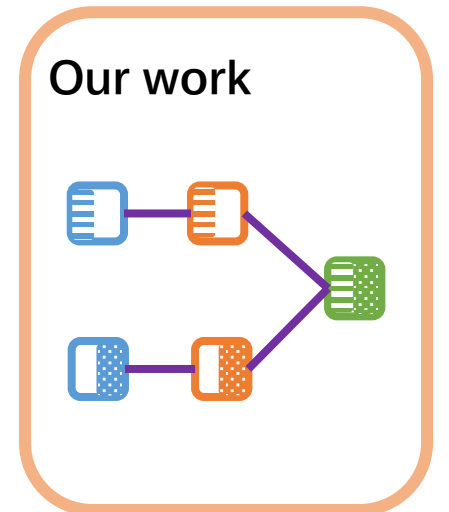
- Structure depends on the matching.



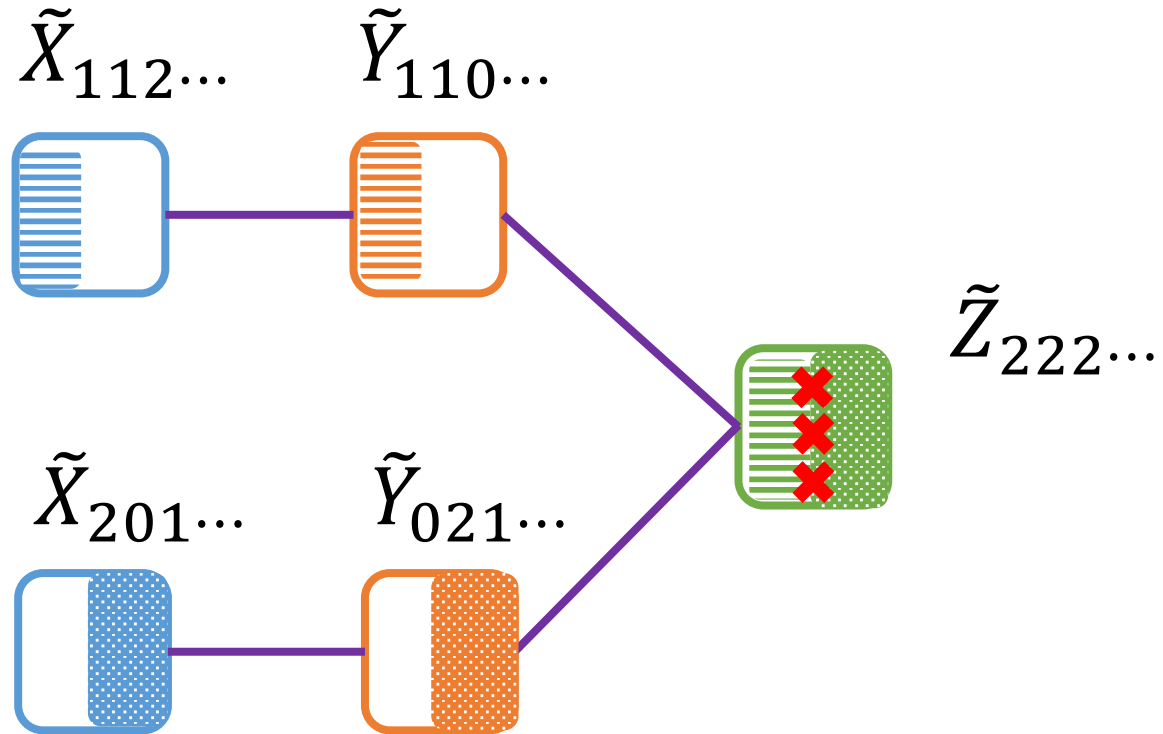
Combination Loss



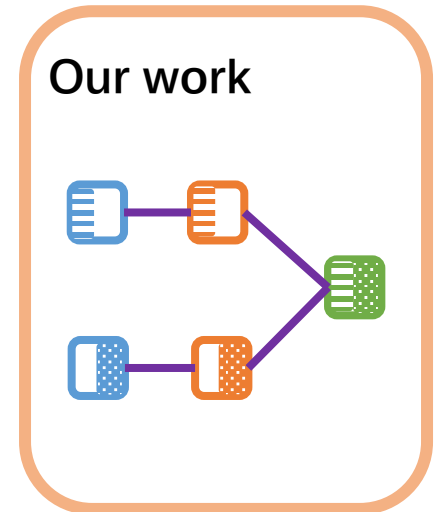
- Structure depends on the matching.
- Idea: Match each Z multiple times



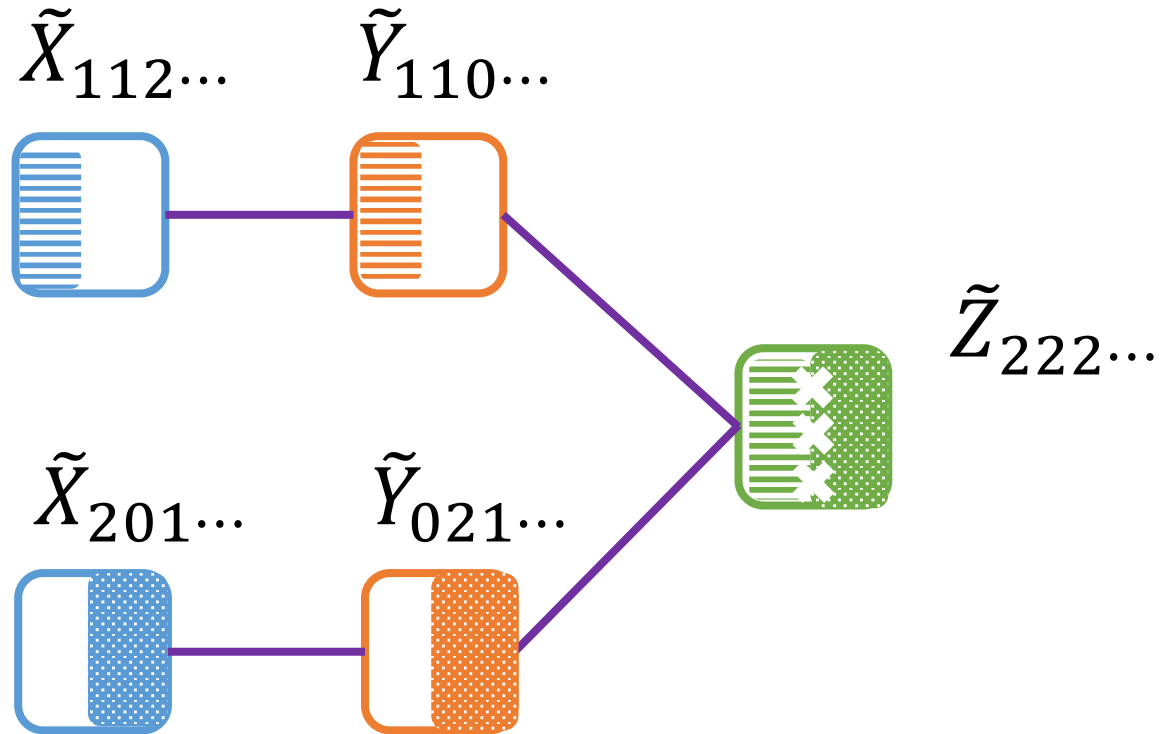
Fixing holes



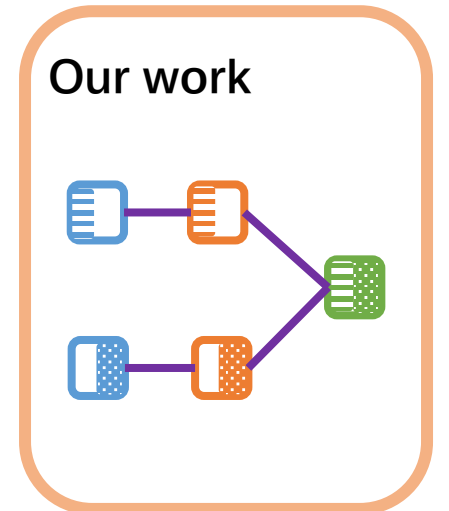
- Still there could be a small number of conflicts.



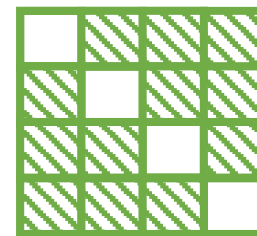
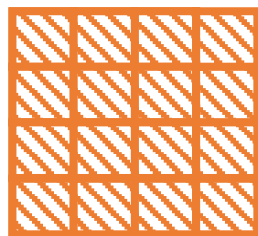
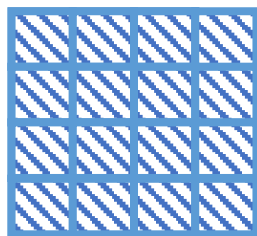
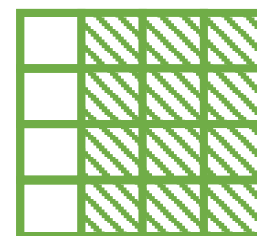
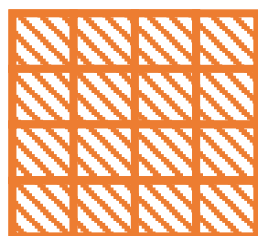
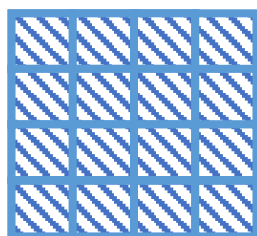
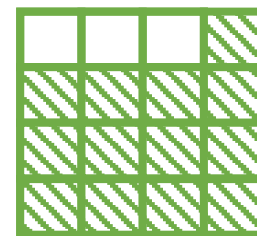
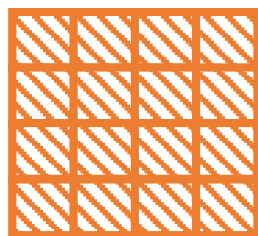
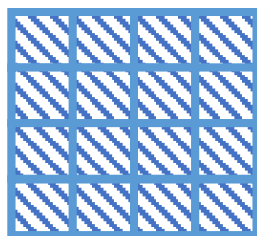
Fixing holes



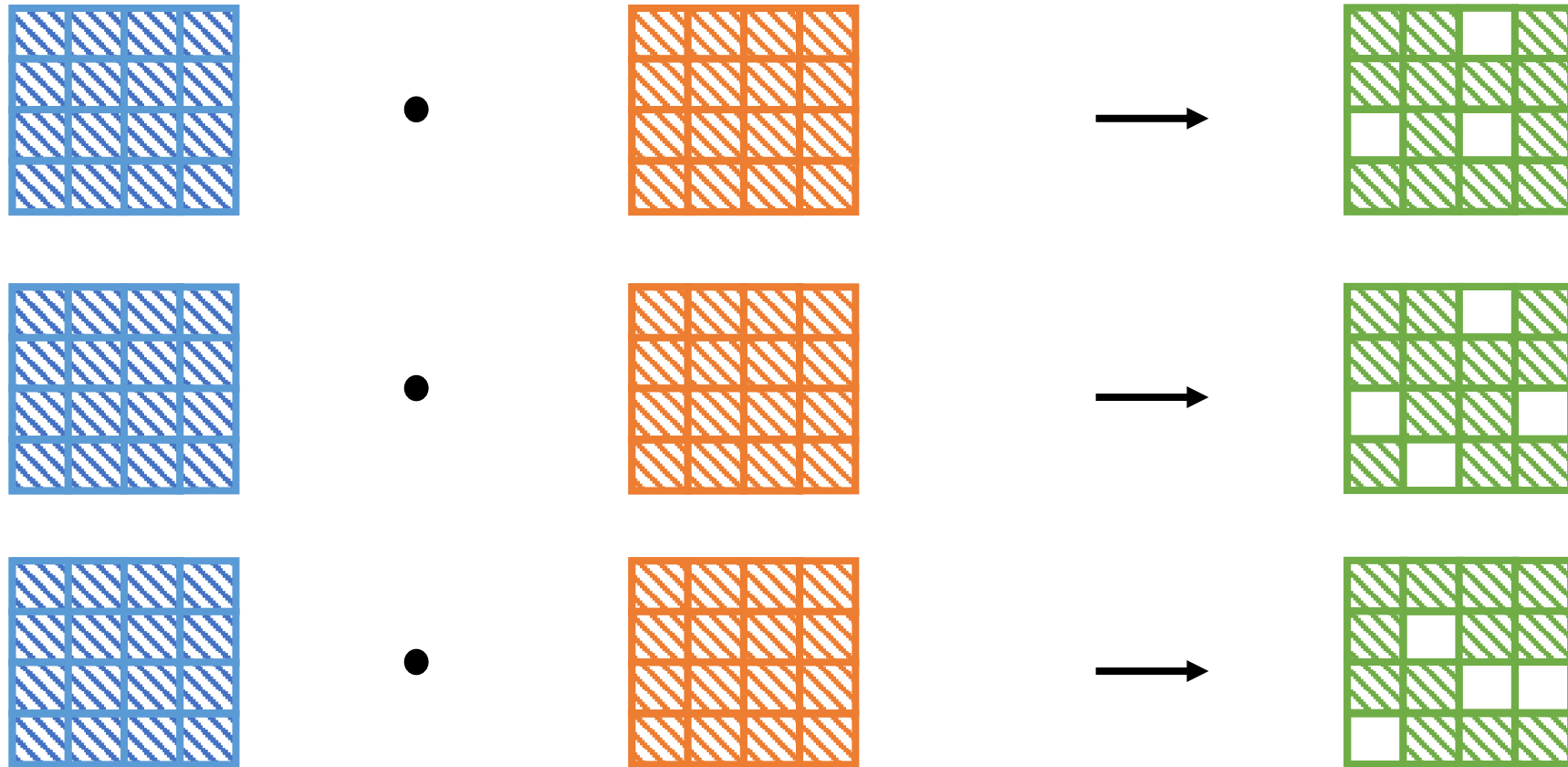
- Still there could be a small number of conflicts. (fix: zero-out)



Fixing holes

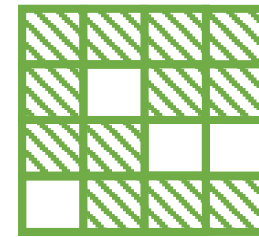
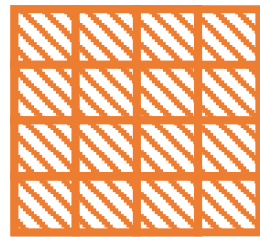
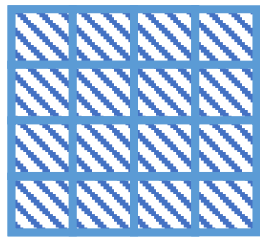
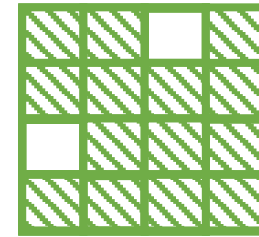
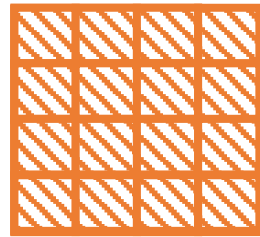
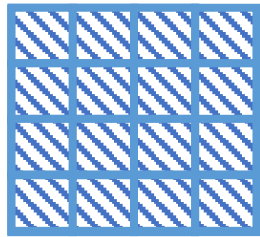


Fixing holes : Shuffle

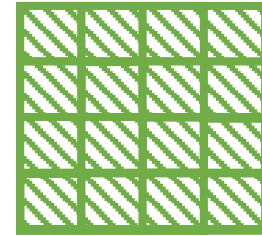
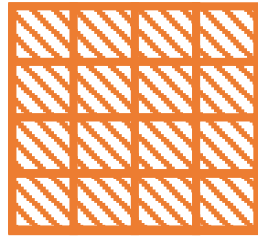
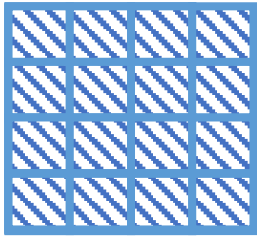


[Karppa, Kaski 19] Random shuffling

Fixing holes : Glue together

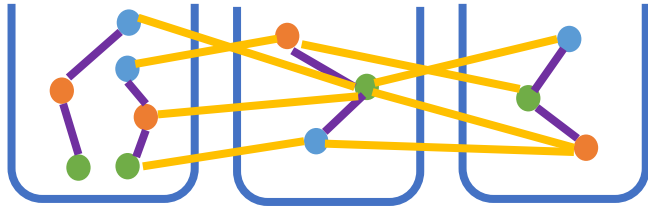


Fixing holes : Glue together

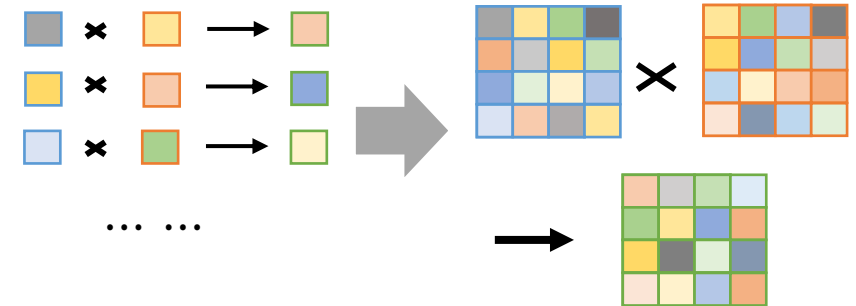


Take-away

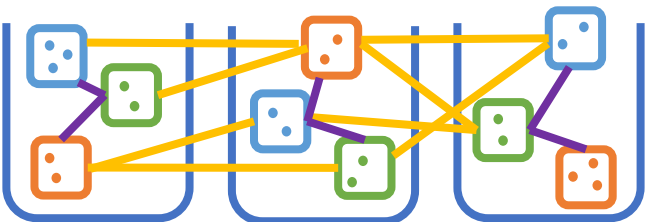
Hashing
[Coppersmith, Winograd 90]



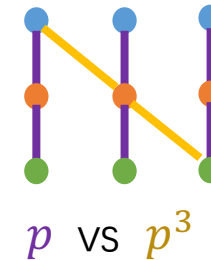
Merging
[CW 90] [Ambainis, Filmus, Le Gall 14]



2/4/8/16/32-th Power
[CW 90][Stothers 10]
[Williams 13] [Le Gall 14]

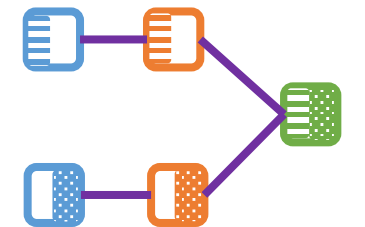


Refined Laser Method
[Alman, Williams 21]

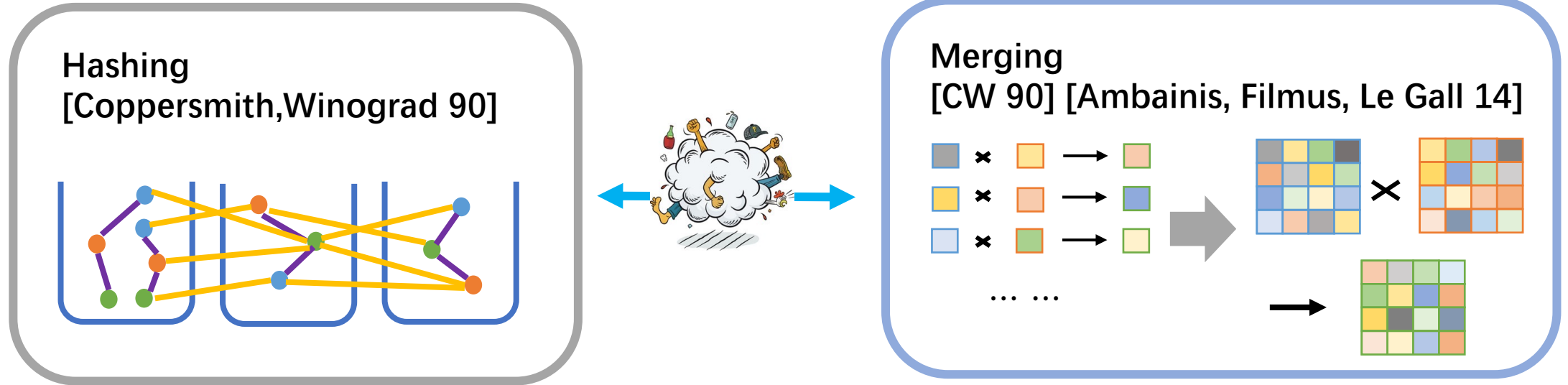


p vs p^3

Our work



Future Directions



- Can we resolve such conflict in some other way?
(Merging cannot prove $\omega < 2.3078$ [Ambainis, Filmus, Le Gall 14])
- A better tensor than T_{CW} ?
(T_{CW} Cannot prove $\omega < 2.16805$
[Alman 19][Christandl, Vrana, Zuiddam 19])

Thanks!