Multidimensional Included and Excluded Sums

Helen Xu

Sean Fraser

Charles E. Leiserson

ACDA
June 21, 2021

CSAIL
MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

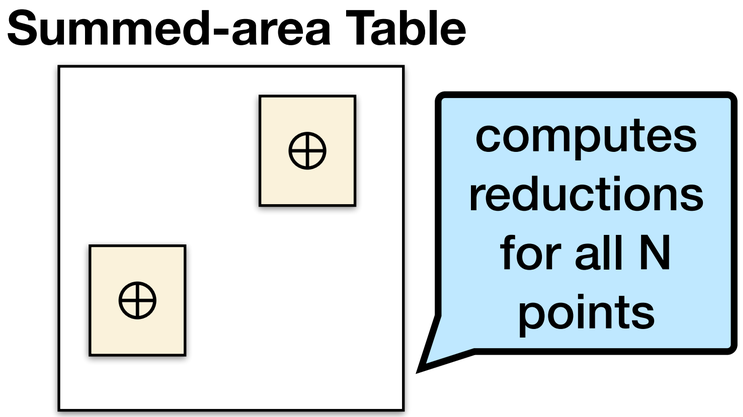
Tensor Region Sums

Several scientific computing applications involve reducing **many (potentially overlapping) regions** of a tensor to a single value for each region, using a binary associative operator $\bigoplus$.

Tensor Region Sums

Several scientific computing applications involve reducing **many (potentially overlapping) regions** of a tensor to a single value for each region, using a binary associative operator $\oplus$.

Inclusion: The **summed-area table** (SAT) method preprocesses an image to answer queries about the sum in rectangular subregions of a tensor [C84].



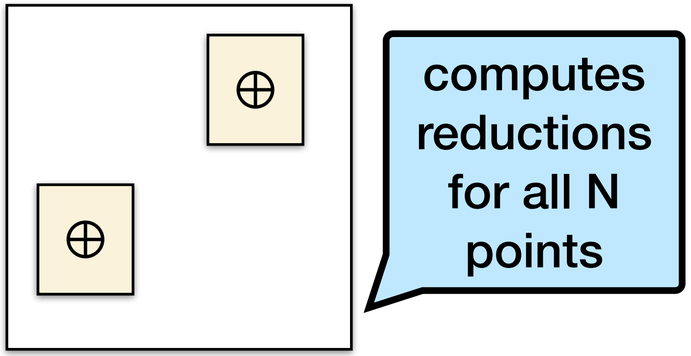
Tensor Region Sums

Several scientific computing applications involve reducing **many (potentially overlapping) regions** of a tensor to a single value for each region, using a binary associative operator $\oplus$.

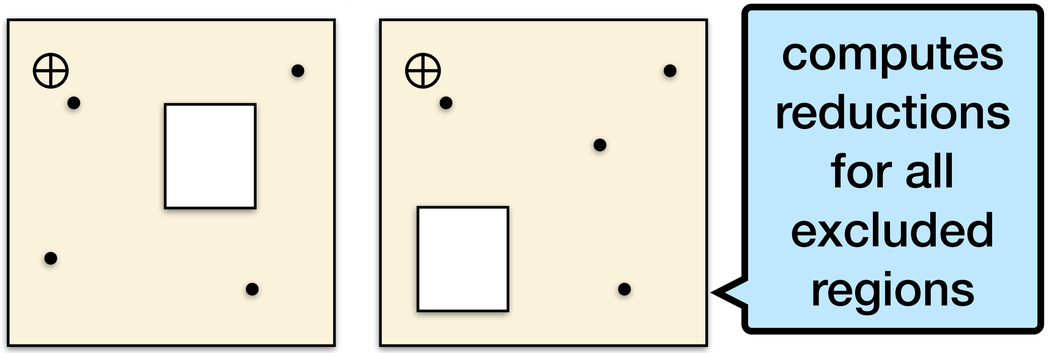
Inclusion: The **summed-area table** (SAT) method preprocesses an image to answer queries about the sum in rectangular subregions of a tensor [C84].

Exclusion: The essence of the **fast multipole method** (FMM) is a reduction of a subregion's elements, excluding elements too close [BG97, CRW93, D00].

Summed-area Table



Fast Multipole Method



Binary associative operator

Excluded-sums Problem

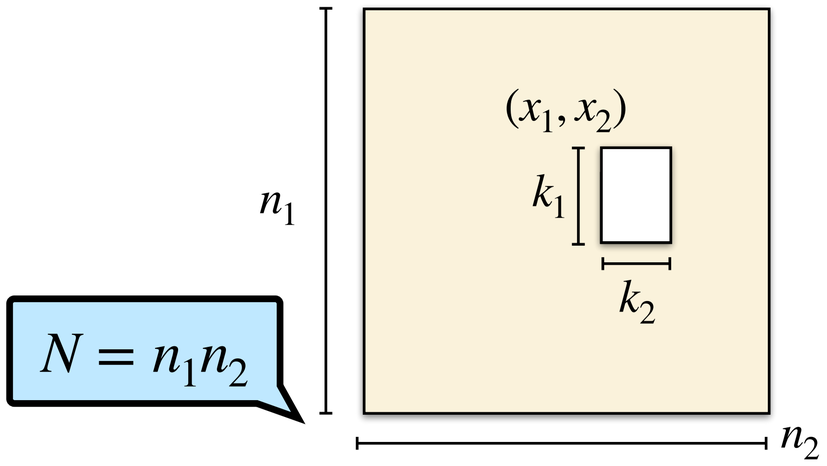
The excluded-sums problem [DDELP05] underlies applications that require **reducing regions of a tensor** to a single value using $\bigoplus$.

Binary associative operator

Excluded-sums Problem

The excluded-sums problem [DDELP05] underlies applications that require **reducing regions of a tensor** to a single value using $\bigoplus$.

In 2D, it takes as input an $n_1 \times n_2$ matrix A and "box size" $\mathbf{k} = (k_1, k_2)$ where $k_1 \leq n_1, k_2 \leq n_2$.



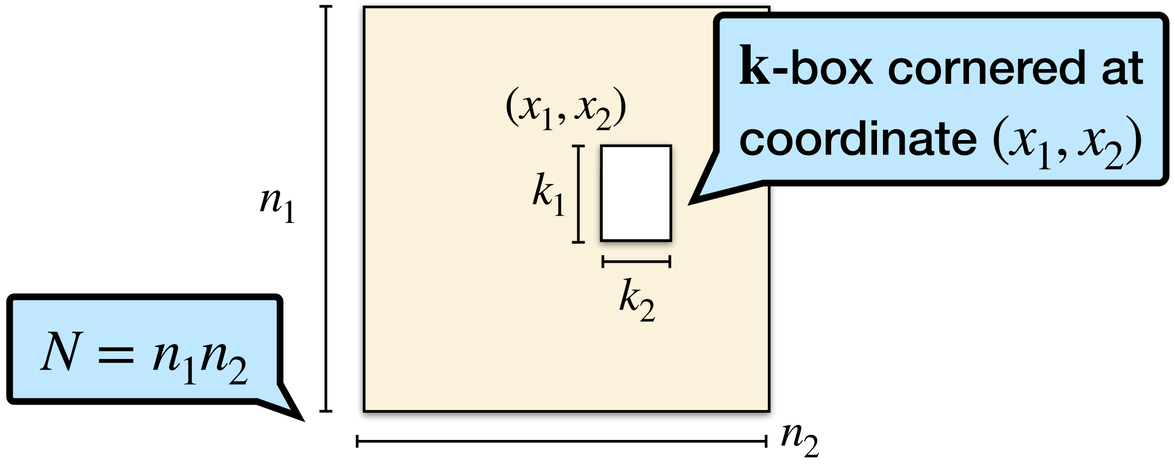
Binary associative operator

Excluded-sums Problem

The excluded-sums problem [DDELP05] underlies applications that require **reducing regions of a tensor** to a single value using $\bigoplus$.

In 2D, it takes as input an $n_1 \times n_2$ matrix A and "box size" $\mathbf{k} = (k_1, k_2)$ where $k_1 \leq n_1, k_2 \leq n_2$.

The problem involves reducing the excluded region outside of **every** $\mathbf{k}$-box in the matrix.

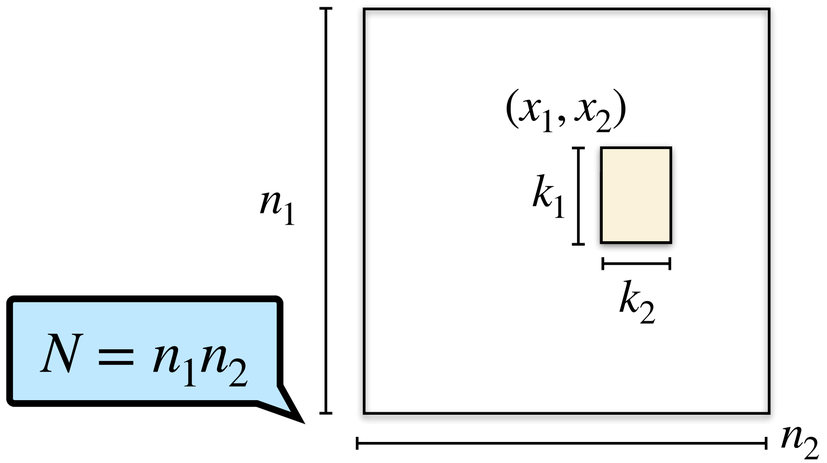


Binary associative operator

Included-sums Problem

The included-sums problem takes the same input as the excluded-sums problem.

In 2D, the included sum at coordinate (x_1, x_2) involves reducing (accumulating with $\bigoplus$) all elements **in the k-box** cornered at (x_1, x_2).

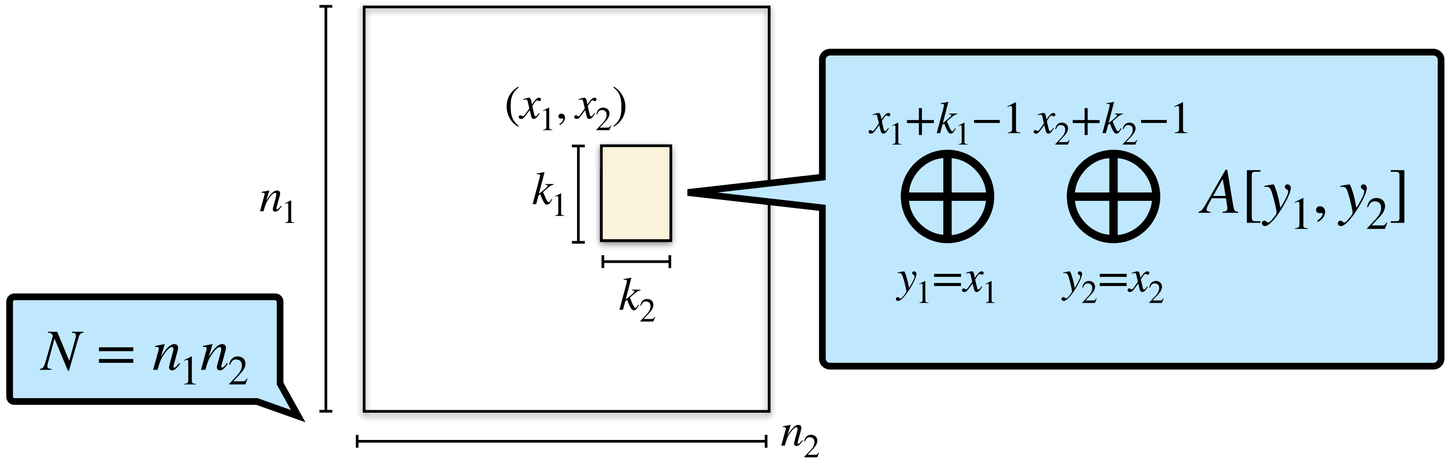


Binary associative operator

Included-sums Problem

The included-sums problem takes the same input as the excluded-sums problem.

In 2D, the included sum at coordinate (x_1, x_2) involves reducing (accumulating with $\oplus$) all elements **in the k-box** cornered at (x_1, x_2).



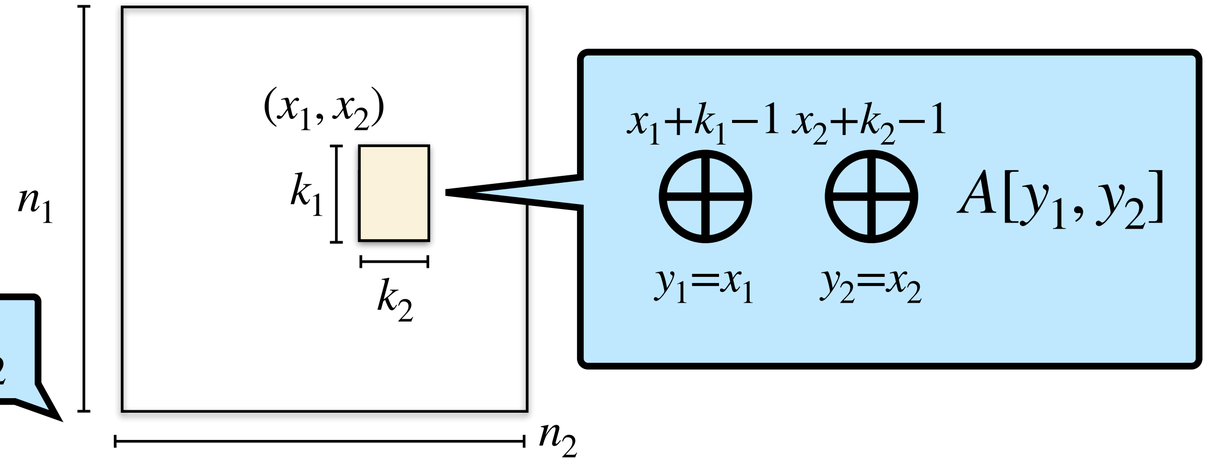
Binary associative operator

Included-sums Problem

The included-sums problem takes the same input as the excluded-sums problem.

In 2D, the included sum at coordinate (x_1, x_2) involves reducing (accumulating with $\oplus$) all elements **in the k-box** cornered at (x_1, x_2).

Can be computed straightforwardly with four nested loops in $\Theta(n_1 n_2 k_1 k_2)$ time.



Inclusion and Exclusion Example

Input

1	3	6	2	5
3	9	1	1	2
5	1	5	3	2
4	3	2	0	9
6	2	1	7	8

Inclusion

16	19	10	10	7
18	16	10	8	4
13	11	10	14	11
15	8	10	24	17
8	3	8	15	8

Inclusion and Exclusion Example

Input

1	3	6	2	5
3	9	1	1	2
5	1	5	3	2
4	3	2	0	9
6	2	1	7	8

Inclusion

16	19	10	10	7
18	16	10	8	4
13	11	10	14	11
15	8	10	24	17
8	3	8	15	8

Exclusion

75	72	81	81	84
73	75	81	83	87
78	80	81	77	80
76	83	81	67	74
83	78	83	76	83

Inclusion and Exclusion Example

Input

1	3	6	2	5
3	9	1	1	2
5	1	5	3	2
4	3	2	0	9
6	2	1	7	8

Inclusion

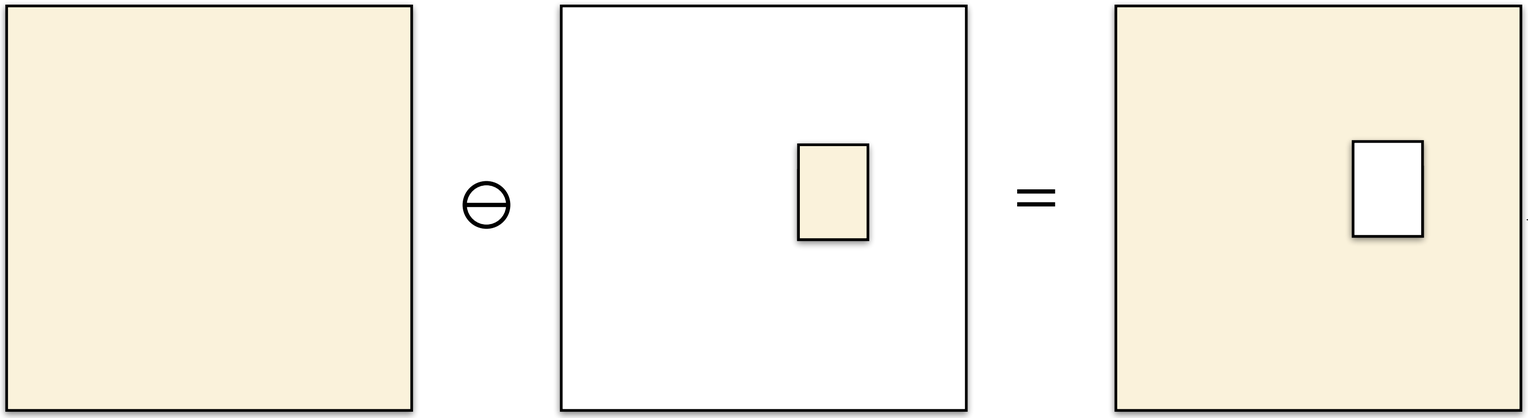
16	19	10	10	7
18	16	10	8	4
13	11	10	14	11
15	8	10	24	17
8	3	8	15	8

Exclusion

75	72	81	81	84
73	75	81	83	87
78	80	81	77	80
76	83	81	67	74
83	78	83	76	83

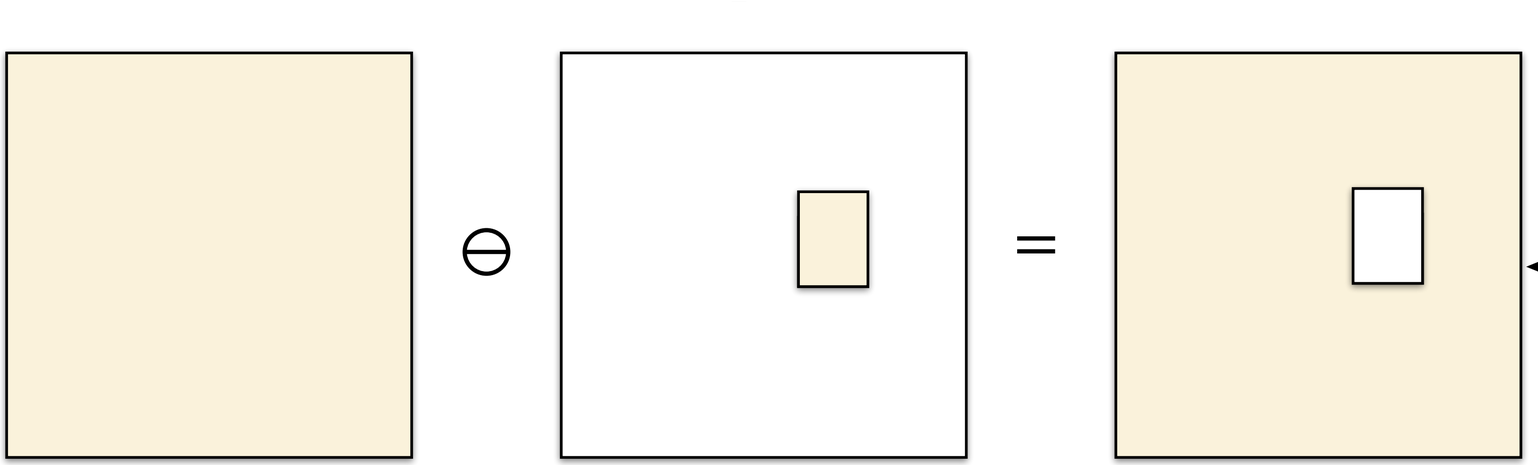
We present an example with addition for ease of understanding, but in general an algorithm for these problems should work with general operators.

Included and Excluded Sums With and Without Operator Inverse



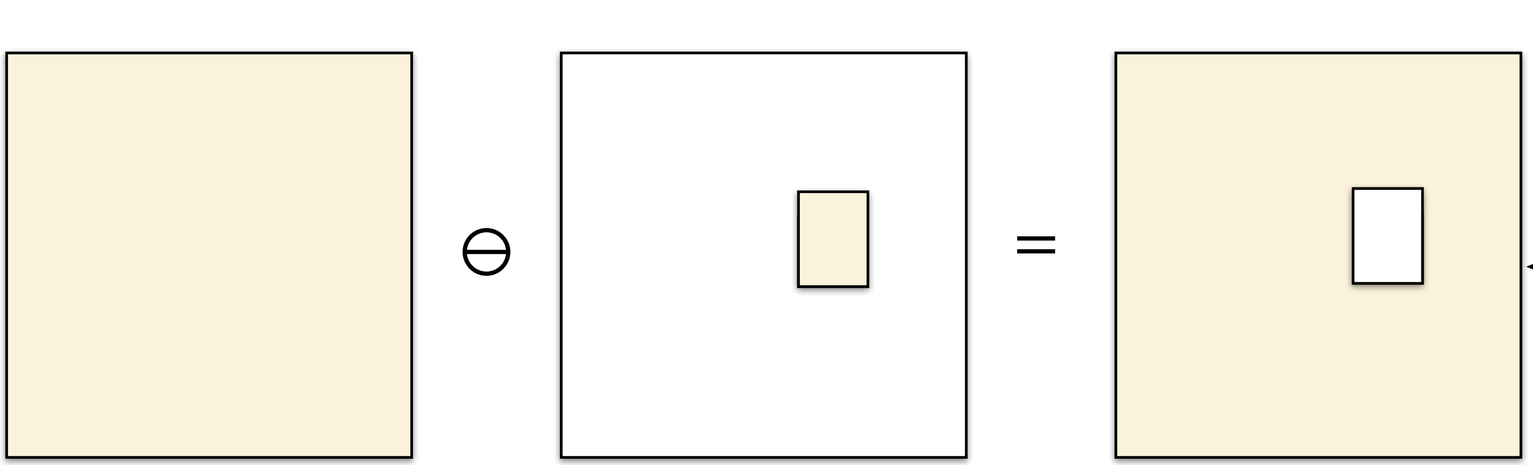
Given an operator inverse, can solve excluded by subtracting out included

Included and Excluded Sums With and Without Operator Inverse



This approach fails for **operators without inverse** such as max, or the FMM's functions, which may exhibit singularities [DDELP05].

Included and Excluded Sums With and Without Operator Inverse



This approach fails for **operators without inverse** such as max, or the FMM's functions, which may exhibit singularities [DDELP05].

We refine the included- and excluded-sums problems into **weak** and **strong** versions. The weak version requires an operator inverse, while the strong version does not.

State of the Art for Included and Excluded Sums

Given a d-dimensional tensor with N elements:

Algorithm	Problem	Weak/Strong	Time	Space
Summed-area table	Included	Weak 😐	$\Theta(2^d N)$ 😐	$\Theta(N)$ 💚

State of the Art for Included and Excluded Sums

Given a d-dimensional tensor with N elements:

Algorithm	Problem	Weak/Strong	Time	Space
Summed-area table	Included	Weak 😐	$\Theta(2^d N)$ 😐	$\Theta(N)$ 💚
Corners(c)	Excluded	Strong 💚	$\Omega(2^d N)$ 😐	$\Theta(cN)$ 😡

Our Contributions

Given a d-dimensional tensor with N elements:

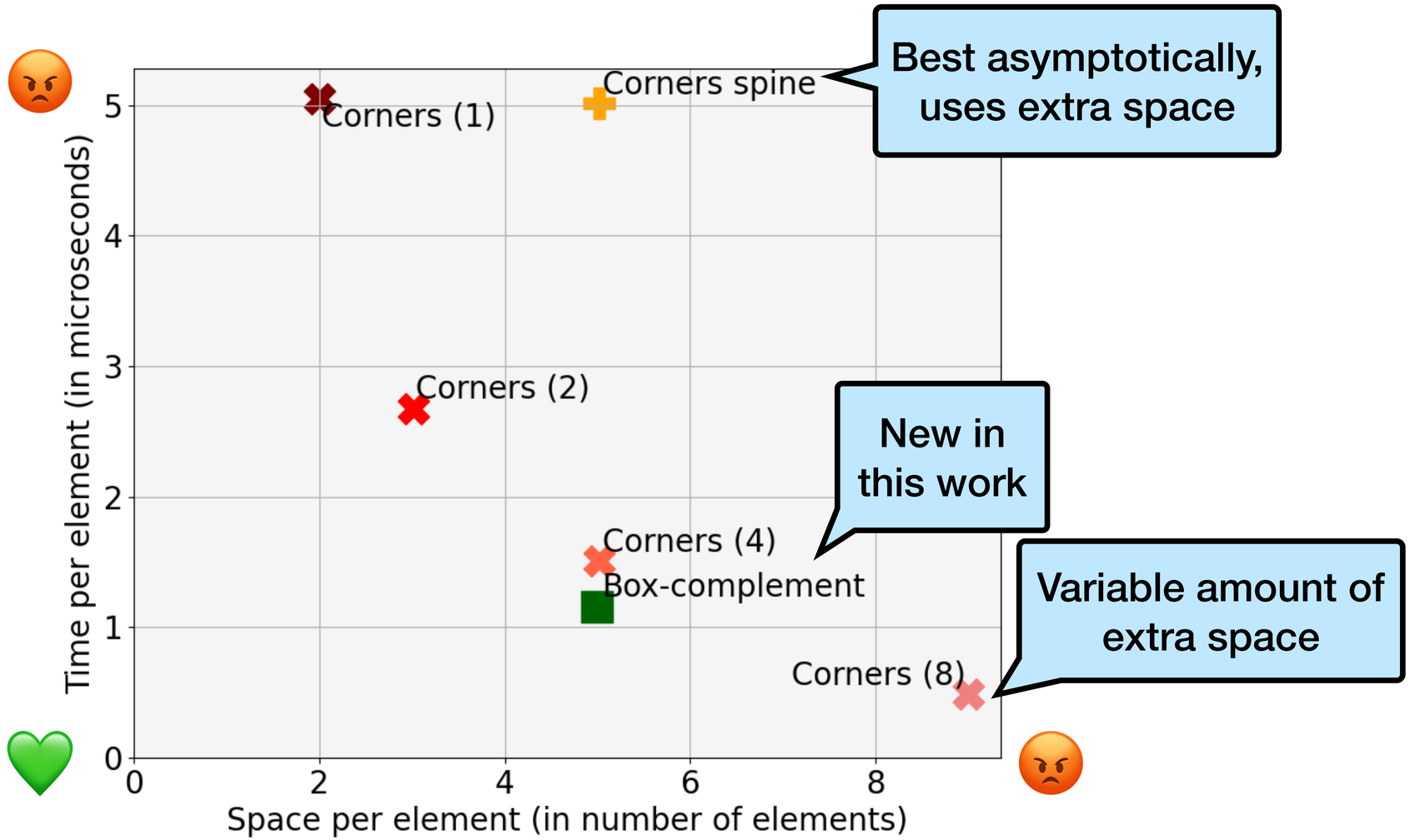
Algorithm	Problem	Weak/Strong	Time	Space
Summed-area table	Included	Weak 😐	$\Theta(2^d N)$ 😐	$\Theta(N)$ 💚
Corners(c)	Excluded	Strong 💚	$\Omega(2^d N)$ 😐	$\Theta(cN)$ 😡
This work				
Bidirectional box-sum (BDBS)	Included	Strong 💚	$\Theta(dN)$ 💚	$\Theta(N)$ 💚

Our Contributions

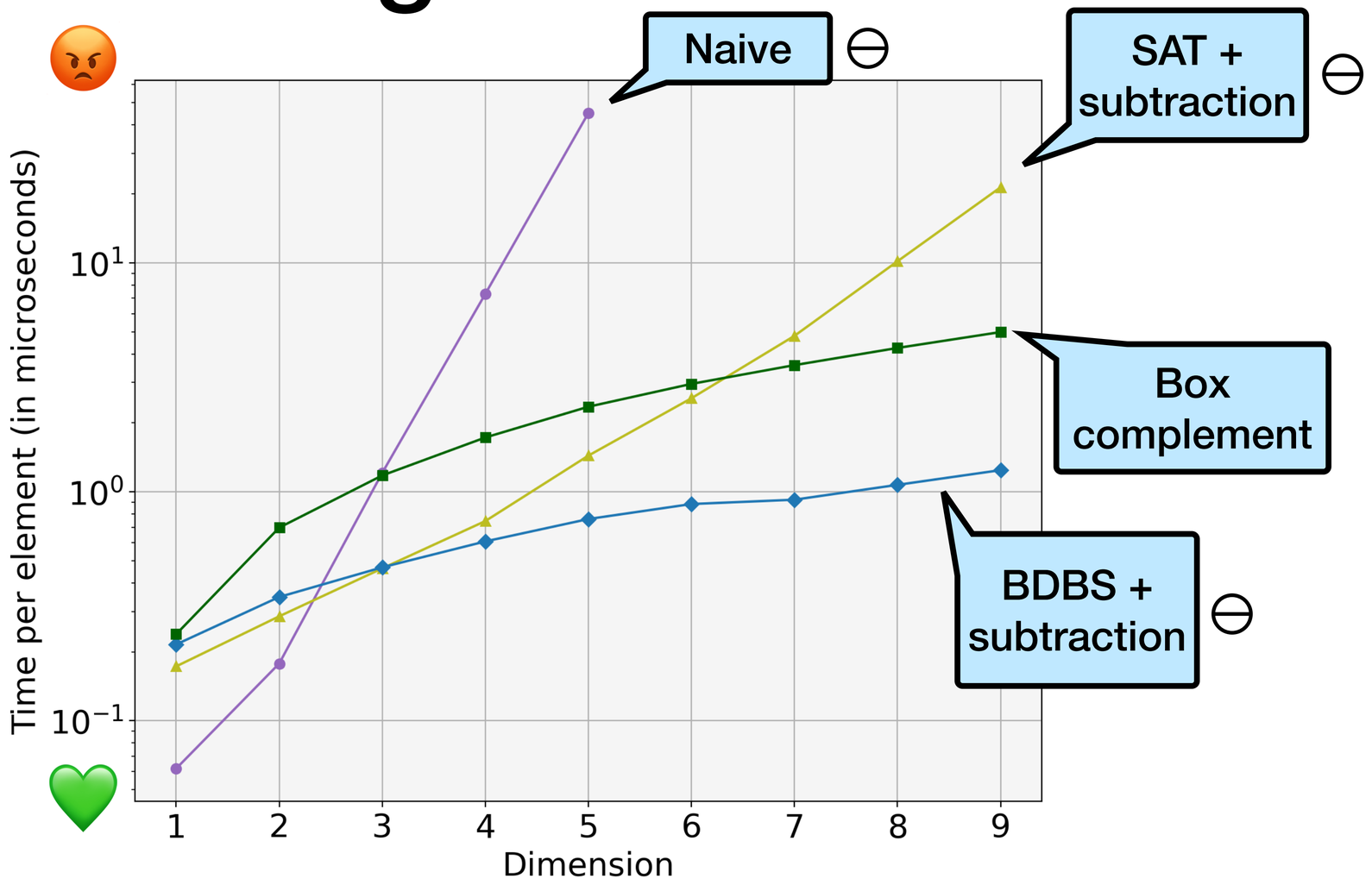
Given a d-dimensional tensor with N elements:

Algorithm	Problem	Weak/Strong	Time	Space
Summed-area table	Included	Weak 😐	$\Theta(2^d N)$ 😐	$\Theta(N)$ 💚
Corners(c)	Excluded	Strong 💚	$\Omega(2^d N)$ 😐	$\Theta(cN)$ 😡
This work				
Bidirectional box-sum (BDBS)	Included	Strong 💚	$\Theta(dN)$ 💚	$\Theta(N)$ 💚
Box complement	Excluded	Strong 💚	$\Theta(dN)$ 💚	$\Theta(N)$ 💚

Strong Excluded Sums in 3D



Weak and Strong Excluded Sums in Higher Dimensions



Bidirectional Box-sum Algorithm for Strong Included Sums

We will start with the bidirectional box-sum algorithm (BDBS) in one dimension then show how to extend the technique to higher dimensions.

Given a list A of length N and a (scalar) box size k, output a list A' of included sums.

$k = 4$

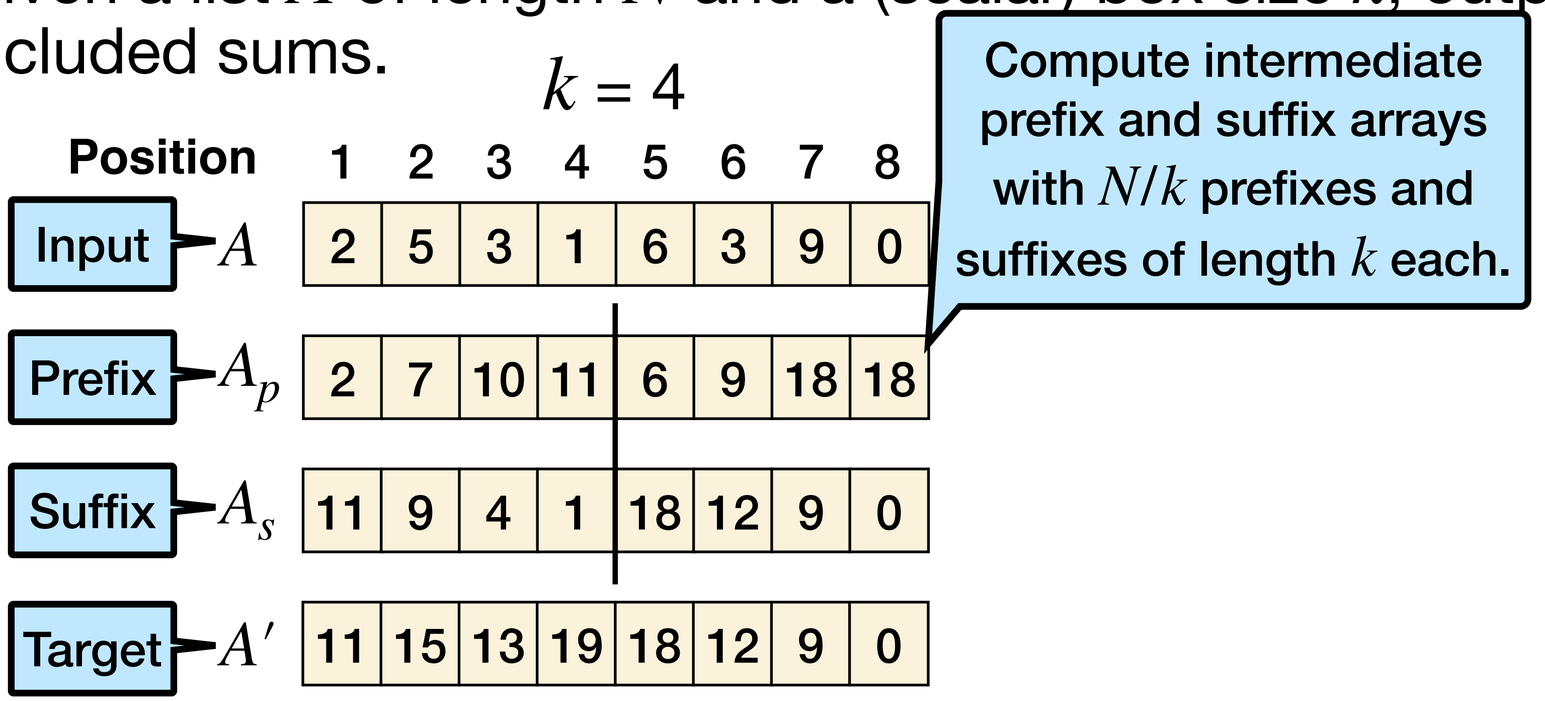
Position	1	2	3	4	5	6	7	8
Input A	2	5	3	1	6	3	9	0

Target A'	11	15	13	19	18	12	9	0

Bidirectional Box-sum Algorithm for Strong Included Sums

We will start with the bidirectional box-sum algorithm (BDBS) in one dimension then show how to extend the technique to higher dimensions.

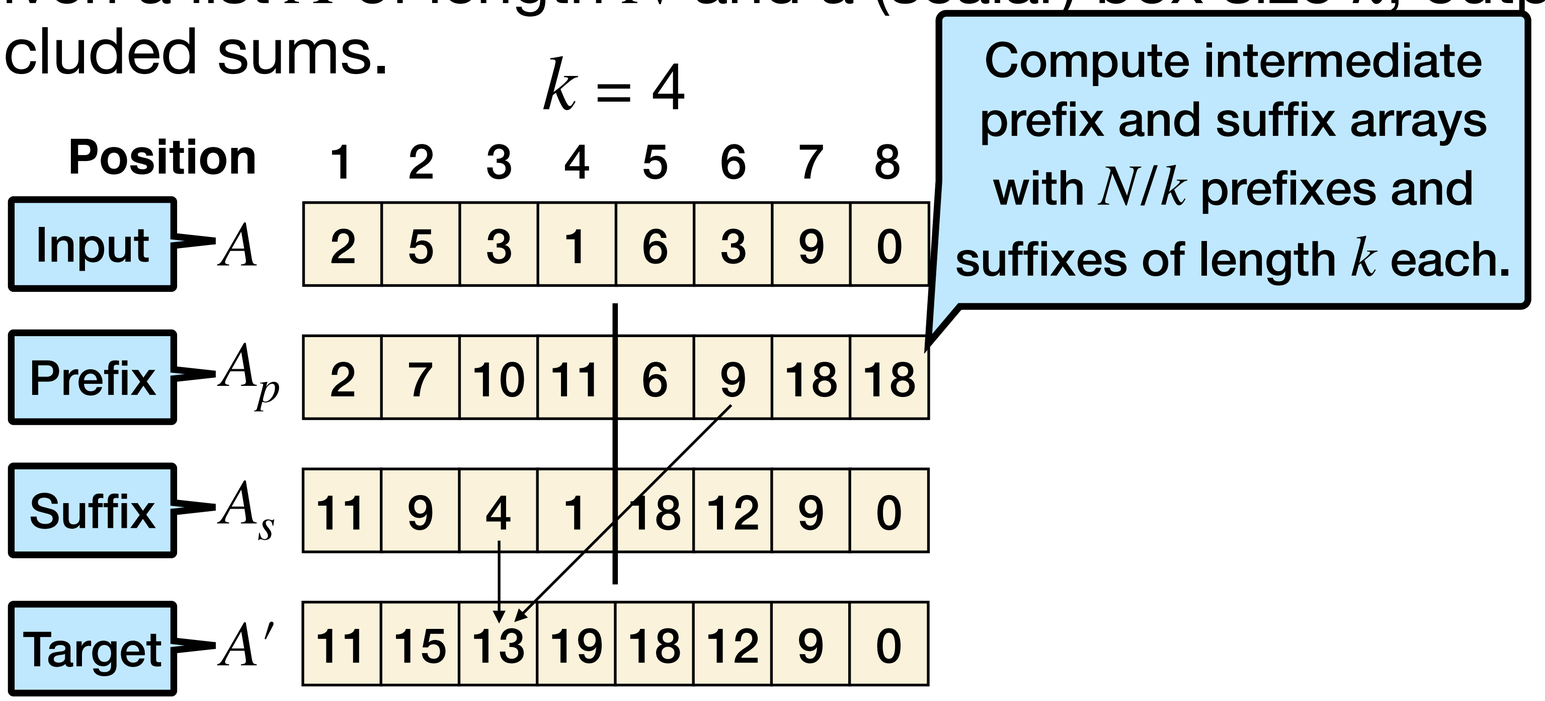
Given a list A of length N and a (scalar) box size k, output a list A' of included sums.



Bidirectional Box-sum Algorithm for Strong Included Sums

We will start with the bidirectional box-sum algorithm (BDBS) in one dimension then show how to extend the technique to higher dimensions.

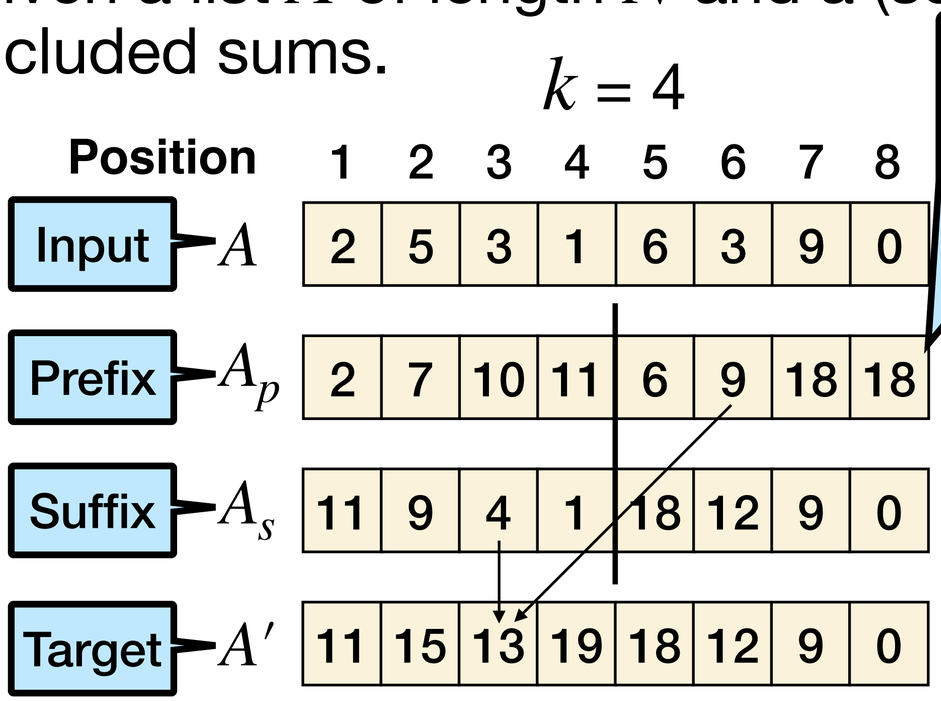
Given a list A of length N and a (scalar) box size k, output a list A' of included sums.



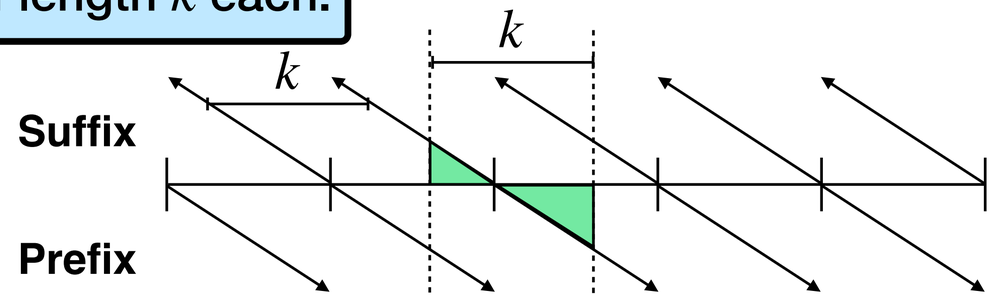
Bidirectional Box-sum Algorithm for Strong Included Sums

We will start with the bidirectional box-sum algorithm (BDBS) in one dimension then show how to extend the technique to higher dimensions.

Given a list A of length N and a (scalar) box size k, output a list A' of included sums.



Compute intermediate prefix and suffix arrays with N/k prefixes and suffixes of length k each.

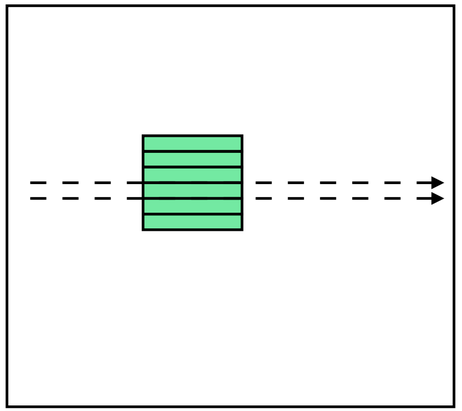


$\Theta(N)$ time, space in 1D

Multidimensional Bidirectional Box Sum

The BDBS technique extends into arbitrary dimensions by performing the prefixes and suffixes along each dimension in turn.

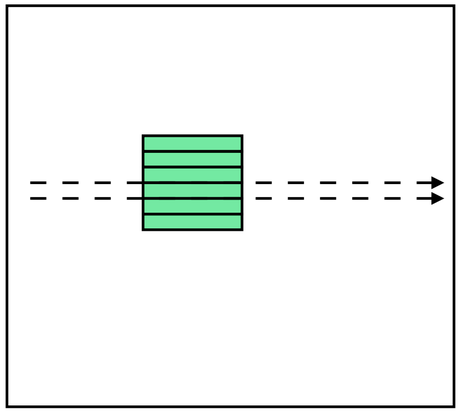
Bidirectional box sum along first dimension



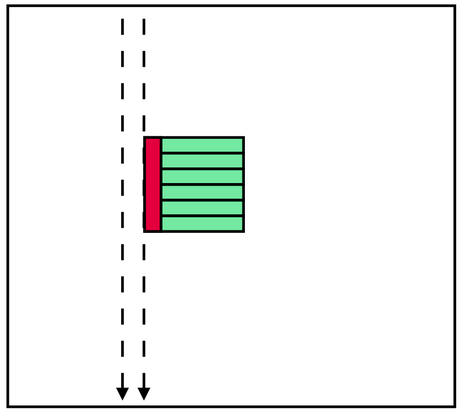
Multidimensional Bidirectional Box Sum

The BDBS technique extends into arbitrary dimensions by performing the prefixes and suffixes along each dimension in turn.

Bidirectional box sum along first dimension



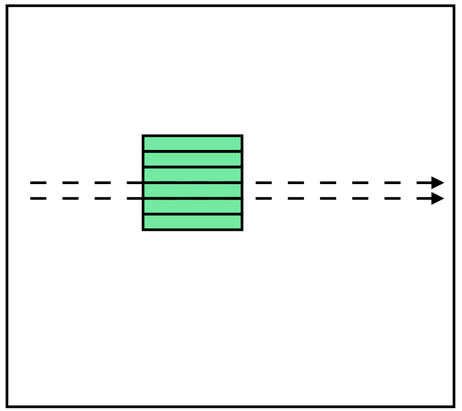
Bidirectional box sum along second dimension on result



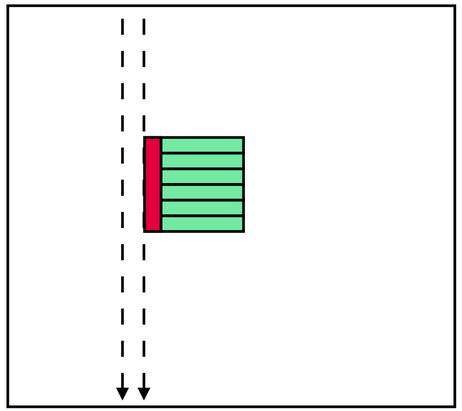
Multidimensional Bidirectional Box Sum

The BDBS technique extends into arbitrary dimensions by performing the prefixes and suffixes along each dimension in turn.

Bidirectional box sum along first dimension



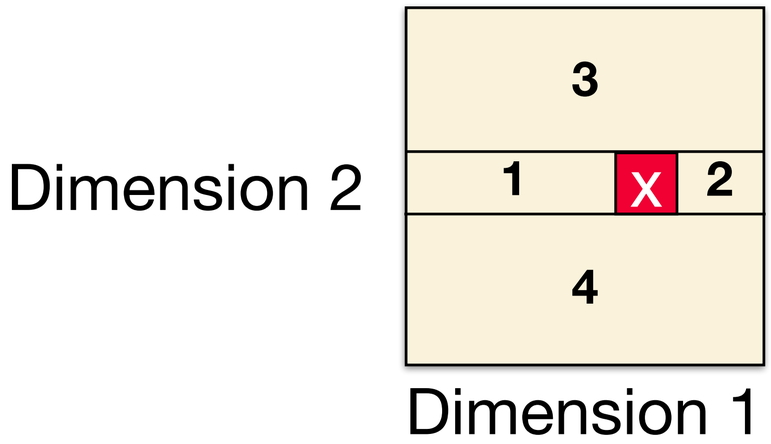
Bidirectional box sum along second dimension on result



Given a d-dimensional tensor with N elements, BDBS solves the strong included-sums problem in $\Theta(dN)$ time and $\Theta(N)$ space.

Formulating the Excluded Sum as the Box Complement

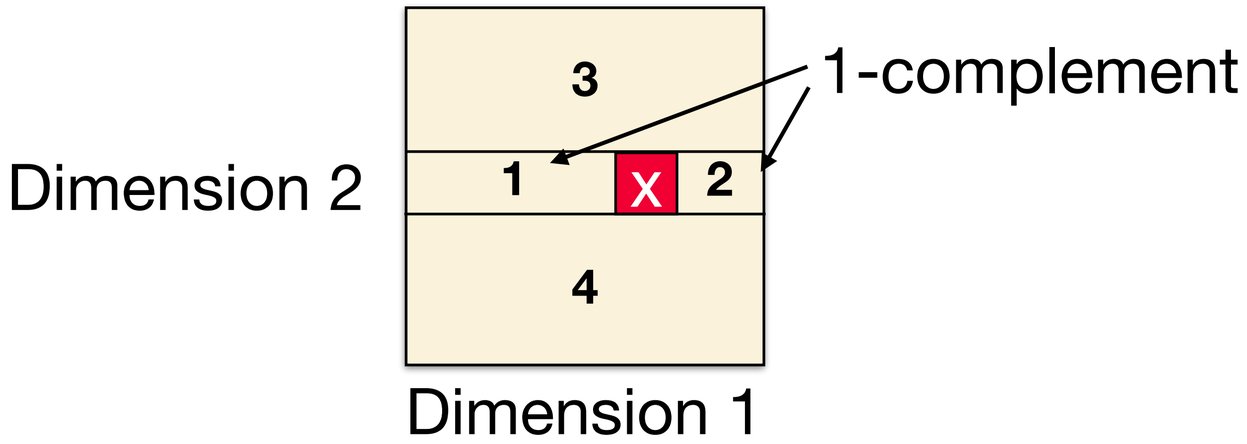
Given a d-dimensional tensor and a “box size”, we will first sketch how to decompose the excluded region for each point into $2d$ disjoint regions.



Formulating the Excluded Sum as the Box Complement

Given a d-dimensional tensor and a "box size", we will first sketch how to decompose the excluded region for each point into $2d$ disjoint regions.

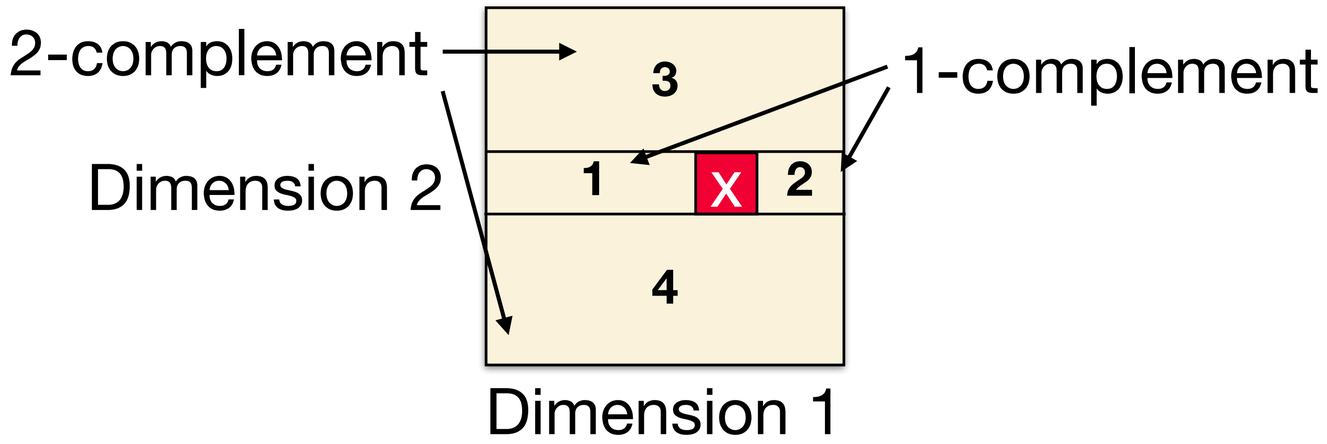
At a high level, the "i-complement" of a box such that there is some coordinate in dimension $j \in [1,i]$ that is "out of range" in dimension j, and the coordinates are "in range" for all dimensions $m \in [i+1,d]$.



Formulating the Excluded Sum as the Box Complement

Given a d-dimensional tensor and a "box size", we will first sketch how to decompose the excluded region for each point into $2d$ disjoint regions.

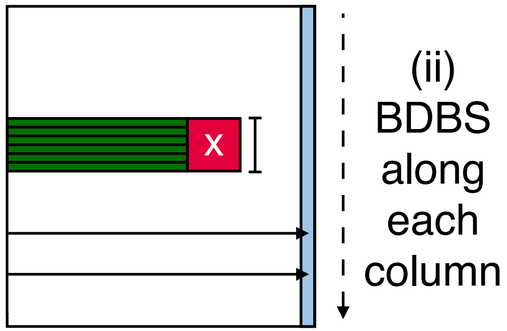
At a high level, the "i-complement" of a box such that there is some coordinate in dimension $j \in [1,i]$ that is "out of range" in dimension j, and the coordinates are "in range" for all dimensions $m \in [i+1,d]$.



Box Complement Algorithm for Strong Excluded Sums

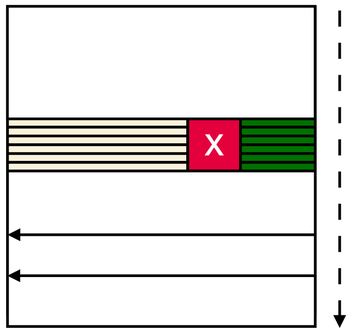
The box-complement algorithm uses **dimension reduction** to compute the "i-complement" for all $i = 1,\ldots,d$.

The **BDBS algorithm** for included sums is a major subroutine in the box-complement algorithm to sum up elements "in the range."



(ii) BDBS along each column

(i) Prefix along each row



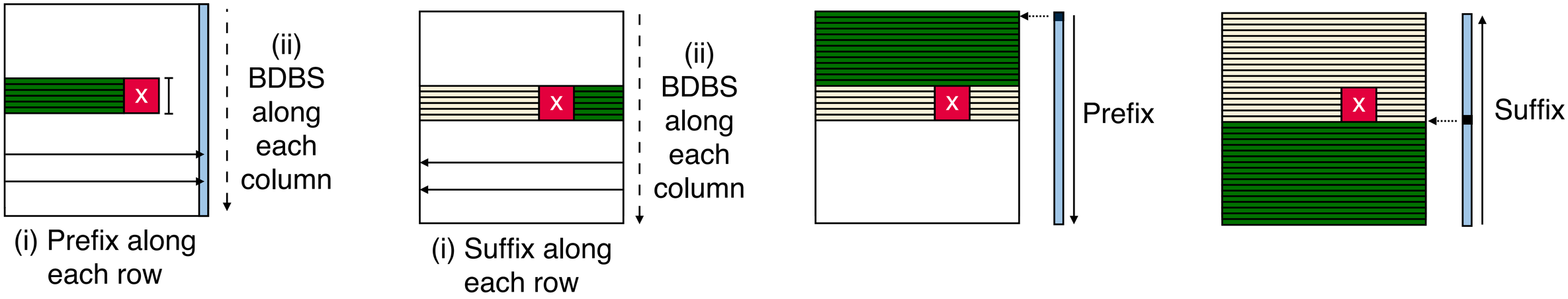
(ii) BDBS along each column

(i) Suffix along each row

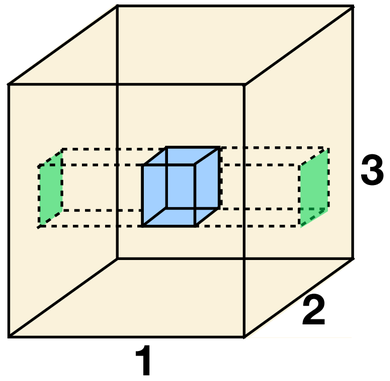
Box Complement Algorithm for Strong Excluded Sums

The box-complement algorithm uses **dimension reduction** to compute the "i-complement" for all $i = 1,\ldots,d$.

The **BDBS algorithm** for included sums is a major subroutine in the box-complement algorithm to sum up elements "in the range."



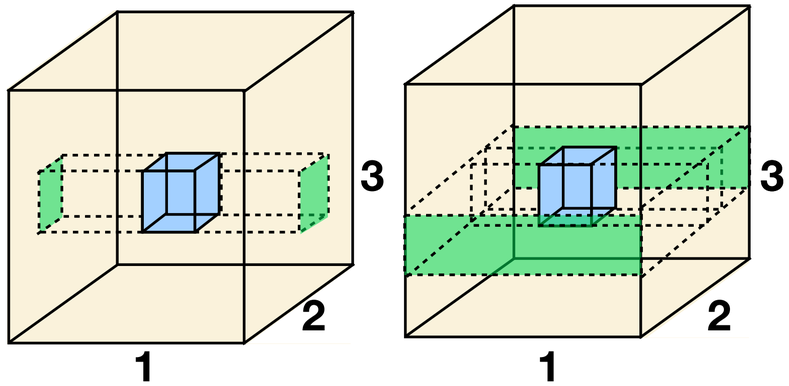
Extending the Box-complement Algorithm to Higher Dimensions



Full Prefix / Suffix Dimensions: 1

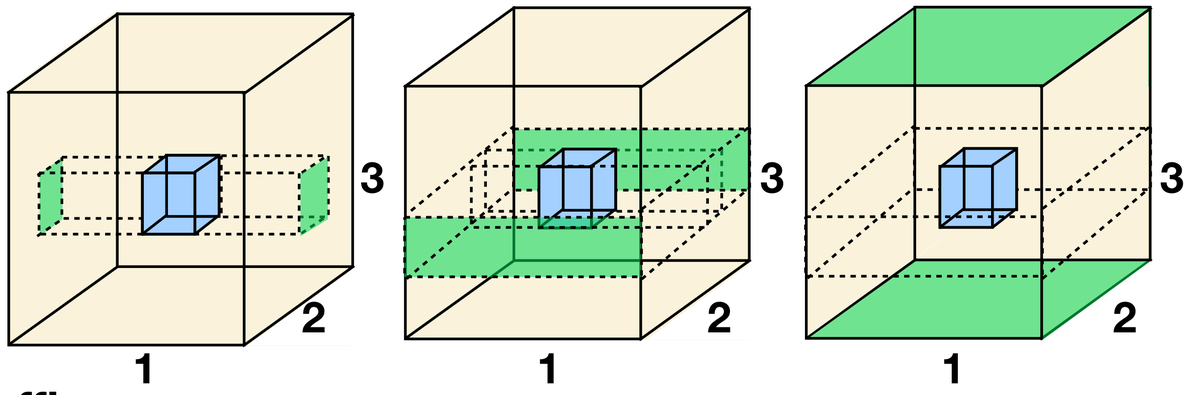
Included Sum Dimensions: 2, 3

Extending the Box-complement Algorithm to Higher Dimensions



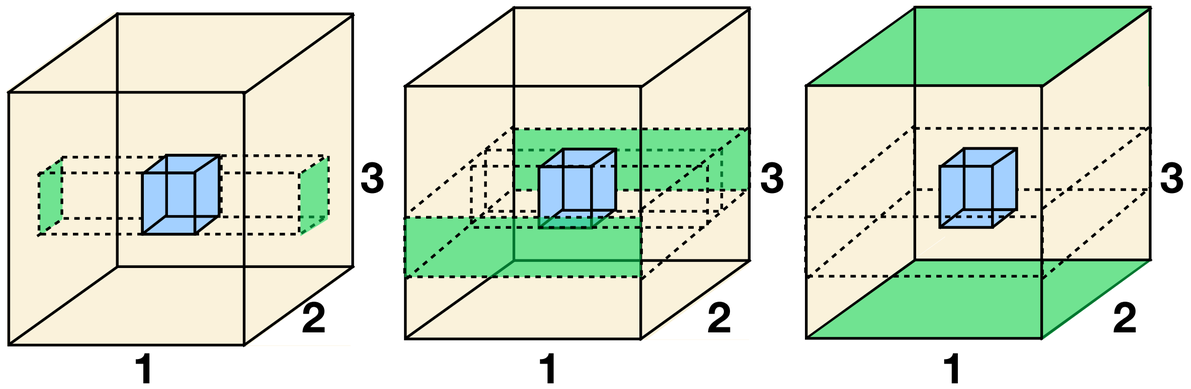
Full Prefix / Suffix Dimensions:	**1**	**1, 2**
Included Sum Dimensions:	**2, 3**	**3**

Extending the Box-complement Algorithm to Higher Dimensions



Full Prefix / Suffix Dimensions:	**1**	**1, 2**	**1, 2, 3**
Included Sum Dimensions:	**2, 3**	**3**	**None**

Extending the Box-complement Algorithm to Higher Dimensions



Full Prefix / Suffix Dimensions:	**1**	**1, 2**	**1, 2, 3**
Included Sum Dimensions:	**2, 3**	**3**	**None**

Each dimension-reduction step takes $\Theta(N)$ time and reuses the same temporaries, for a total of $\Theta(dN)$ time and $\Theta(N)$ space.

Conclusion

This work introduces the box-complement algorithm for the excluded-sums problem, which improves the running time of the state-of-the-art corners algorithm **from $\Omega(2^d N)$ to $\Theta(dN)$**.

Conclusion

This work introduces the box-complement algorithm for the excluded-sums problem, which improves the running time of the state-of-the-art corners algorithm **from $\Omega(2^d N)$ to $\Theta(dN)$**.

Since floating-point subtraction is less numerically stable than addition, our next step is to study the **numerical accuracy** of the different algorithms.

Conclusion

This work introduces the box-complement algorithm for the excluded-sums problem, which improves the running time of the state-of-the-art corners algorithm **from $\Omega(2^d N)$ to $\Theta(dN)$**.

Since floating-point subtraction is less numerically stable than addition, our next step is to study the **numerical accuracy** of the different algorithms.

Future work includes incorporating the BDBS and box-complement algorithm into the FMM.

Excluded Sums with Different Operators