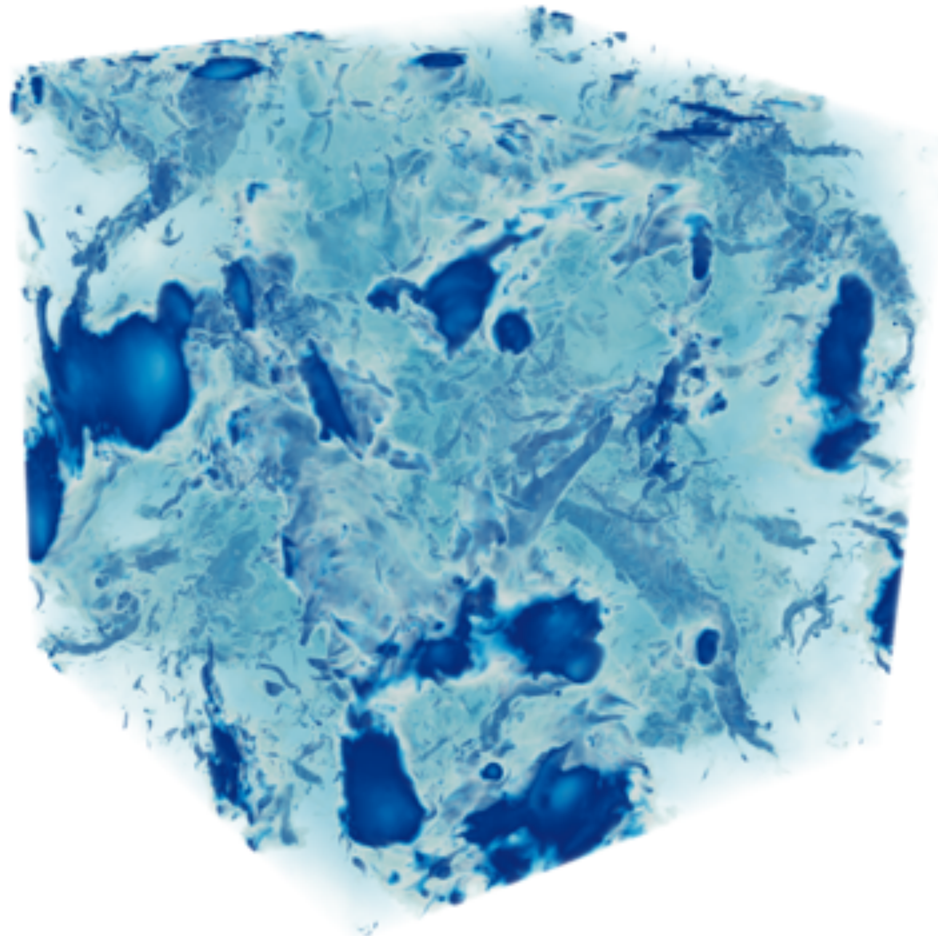
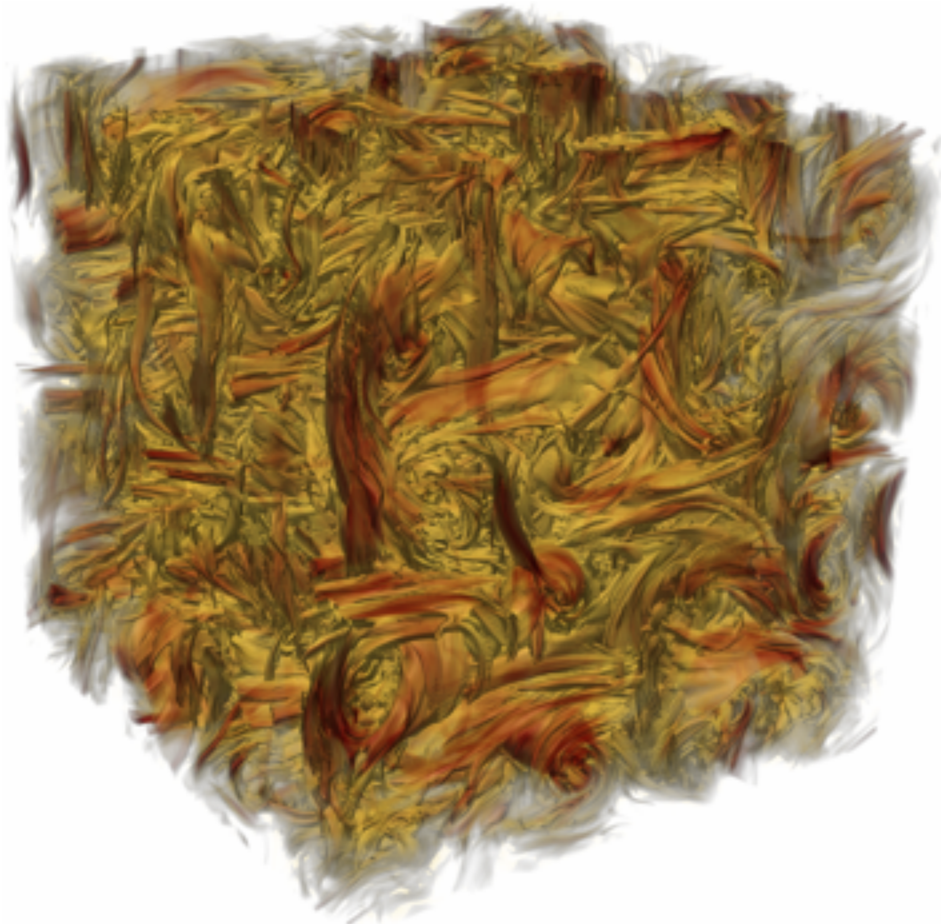
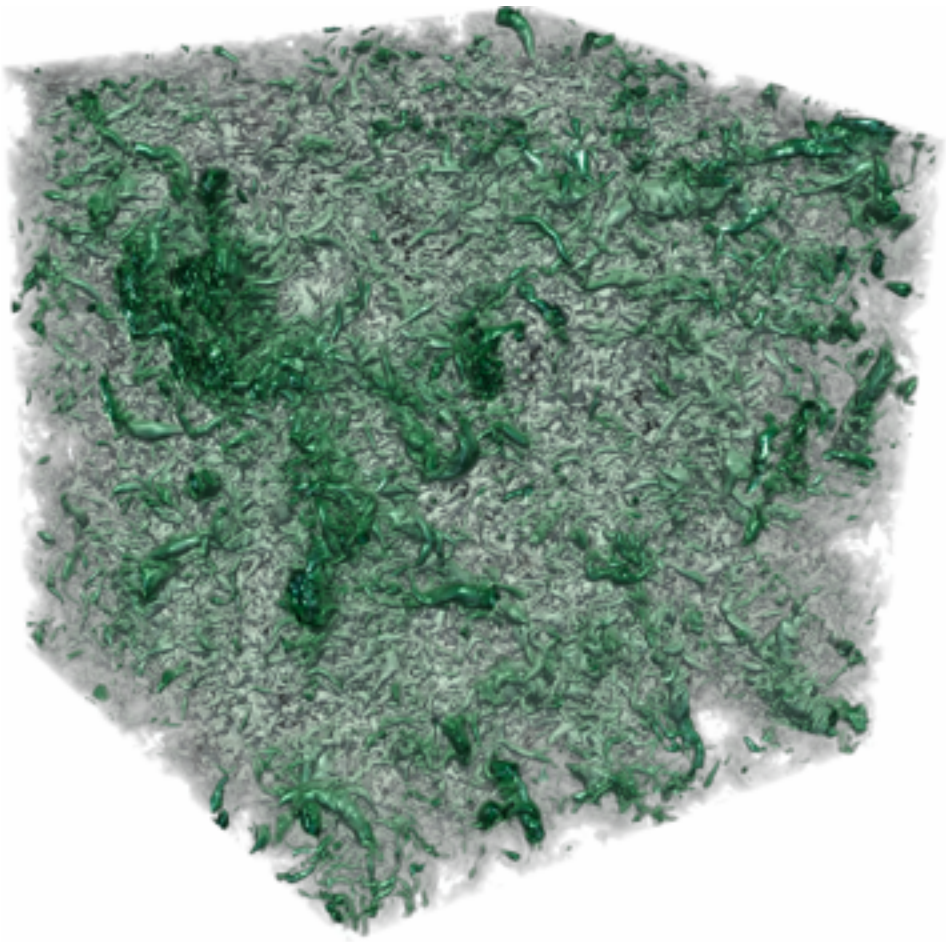
# Tensor Methods for High-Dimensional Analysis and Visualization

Rafael Ballester-Ripoll

26th September 2017

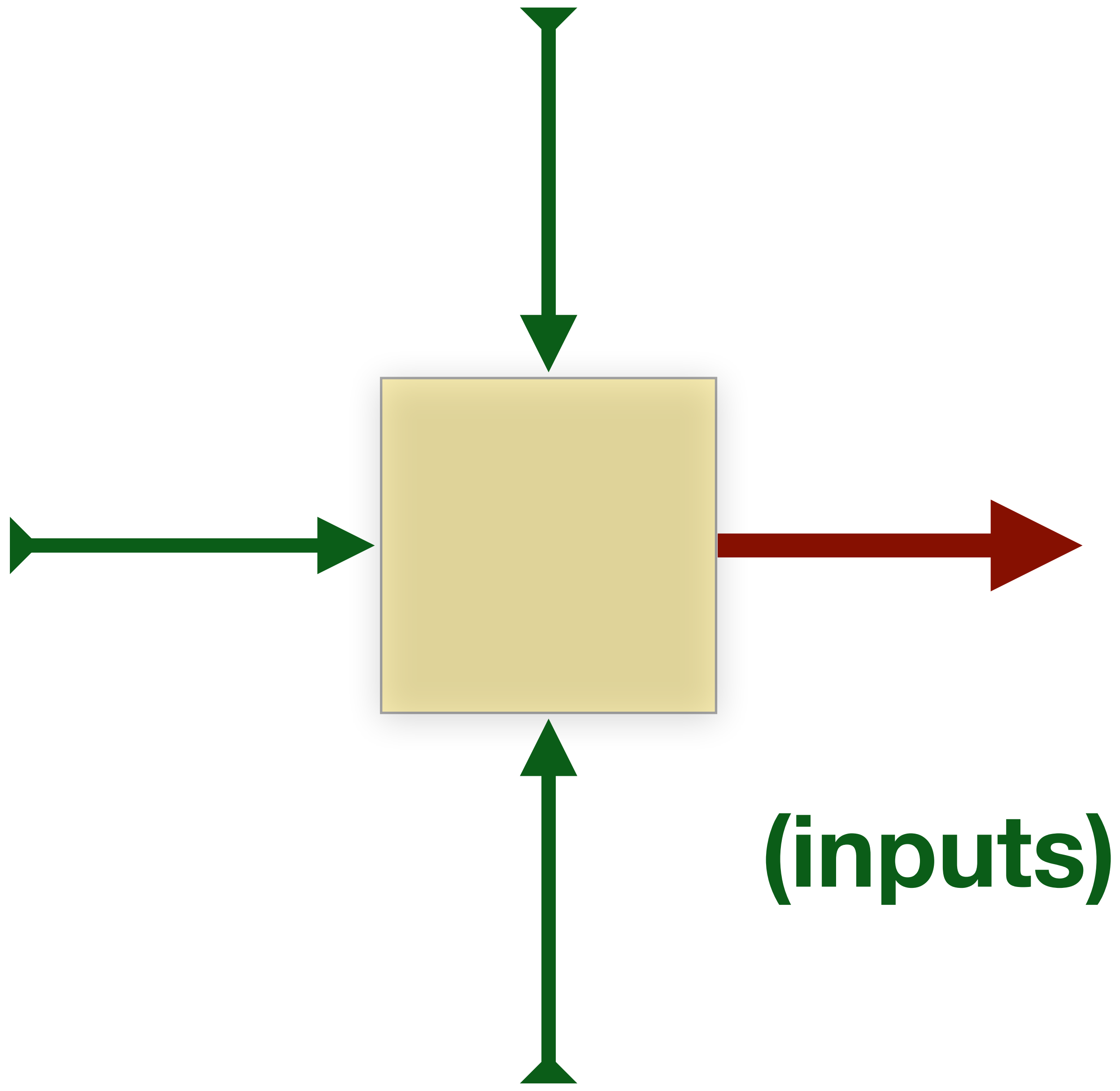
University of Zurich UZH

VISUALIZATIONANDMULTIMEDIALAB

# Motivation

**Data is more abundant than ever**

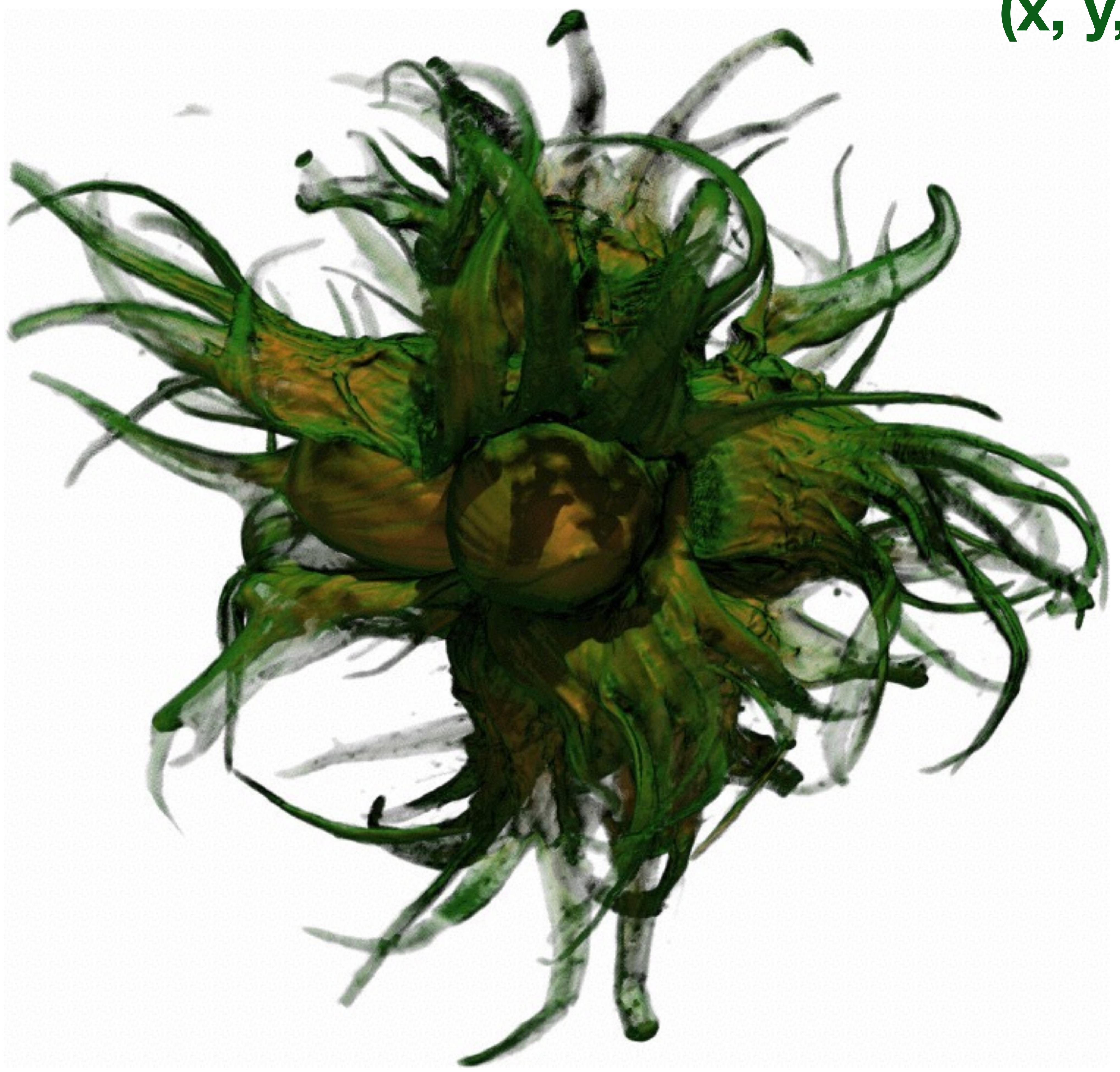
- Growing **data size**:
  - Computing power
  - Acquisition technology
- Many **dimensions**
- Complex **dependence on variables**



(inputs) ⟶ output

# Computer Tomography

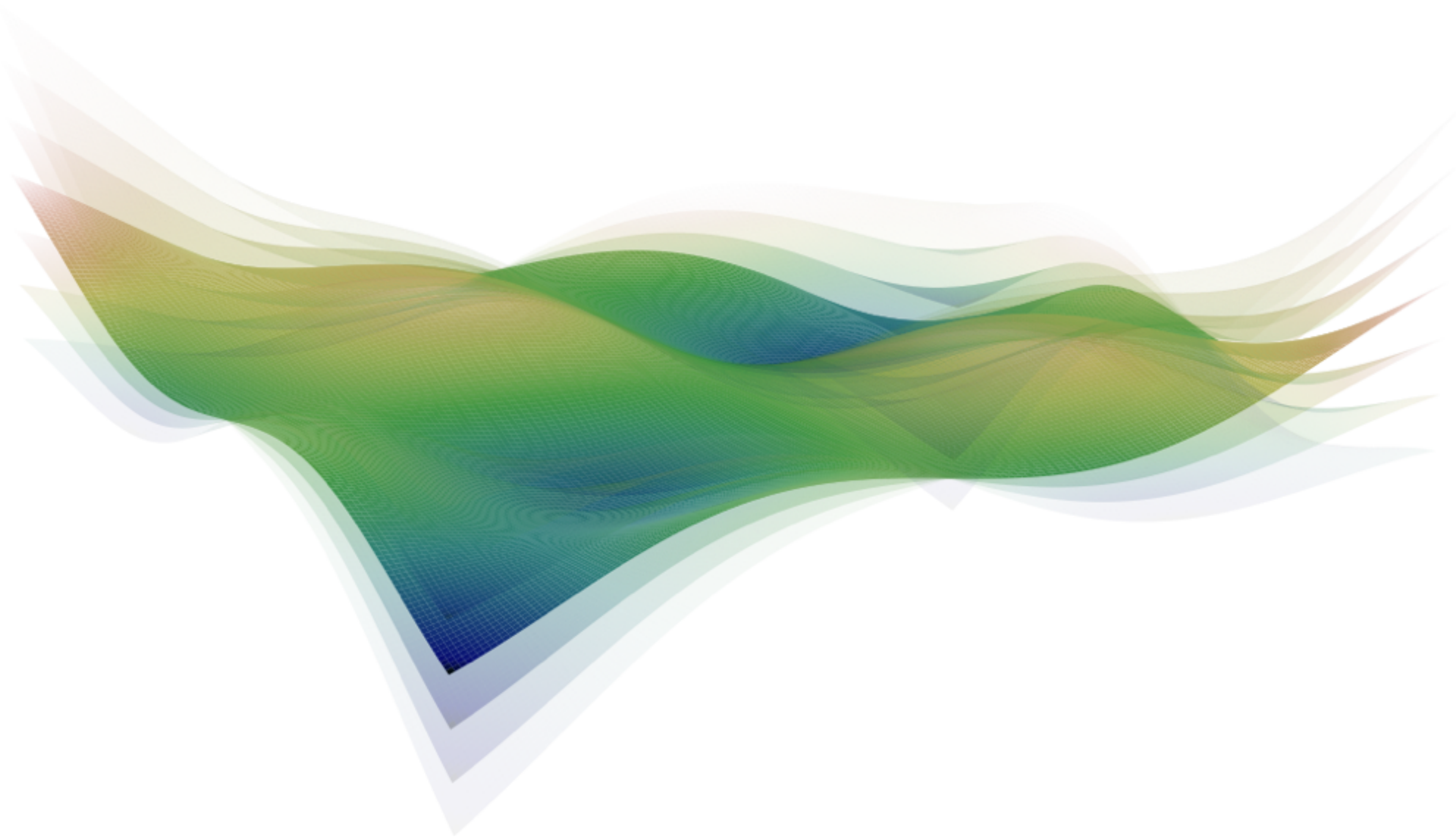
(x, y, z) ⟶ **density**

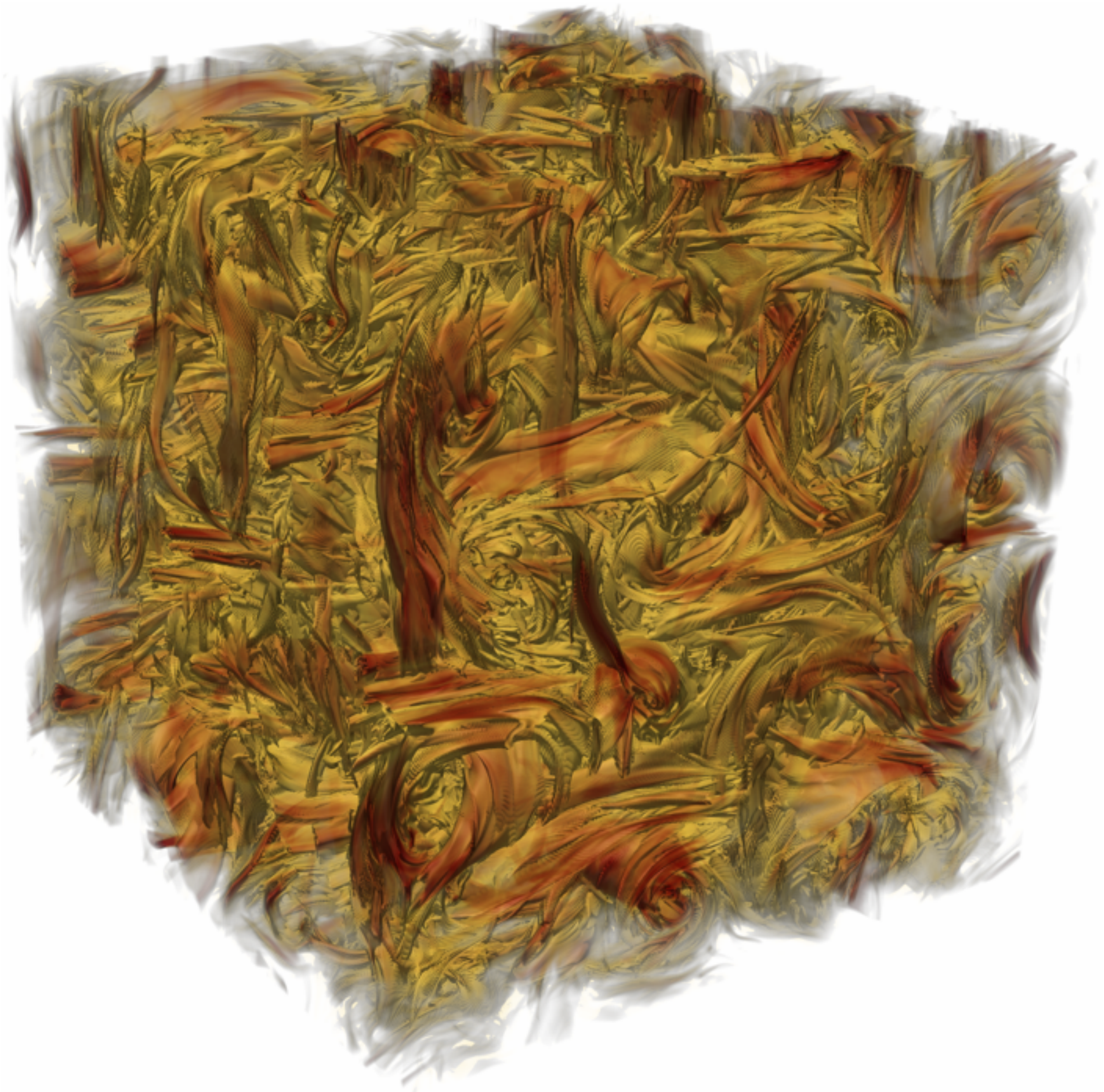


VISUALIZATIONAND
MULTIMEDIALAB

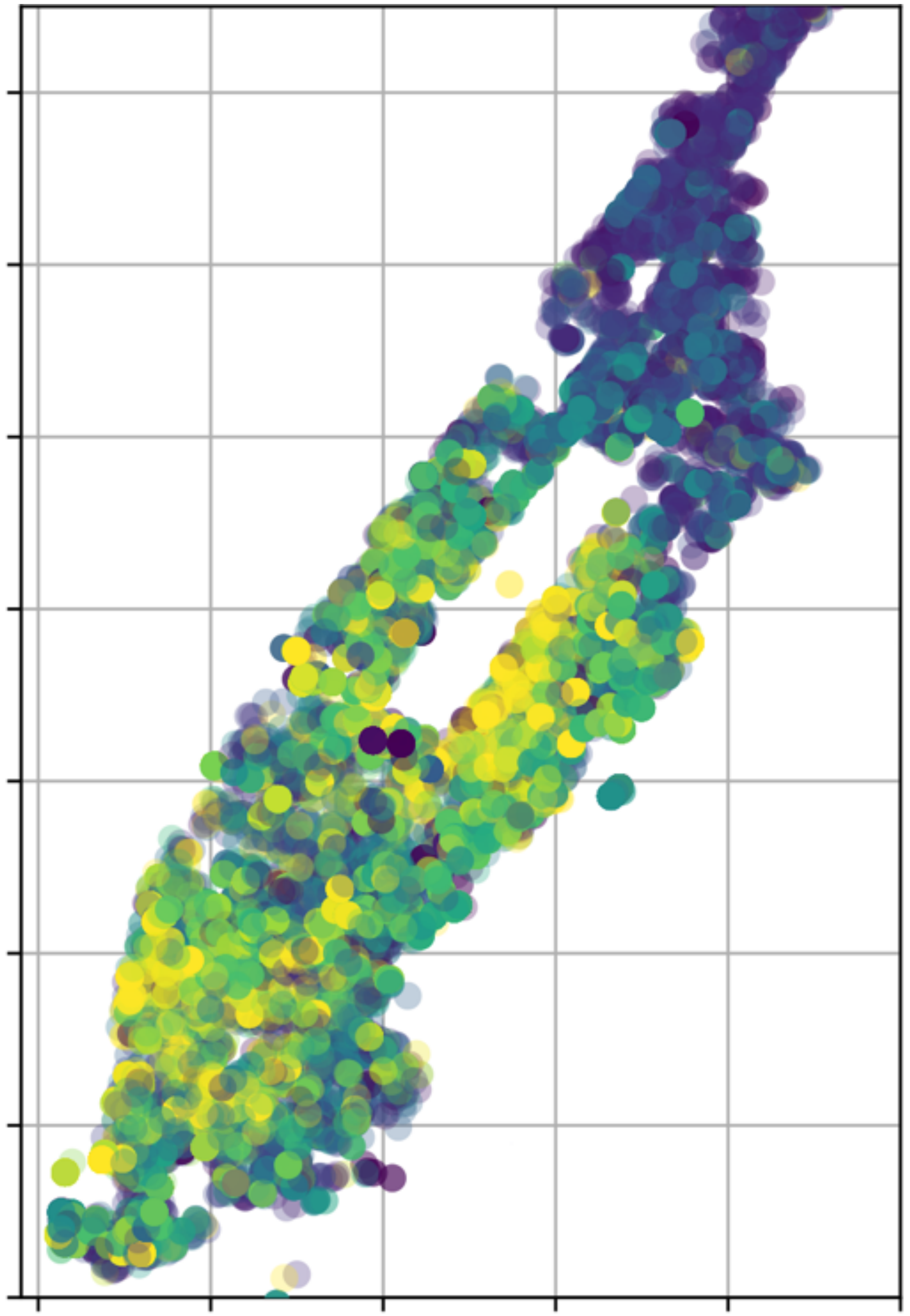
# Engineering Processes

**parameters** ⟶ **result**





# Machine Learning

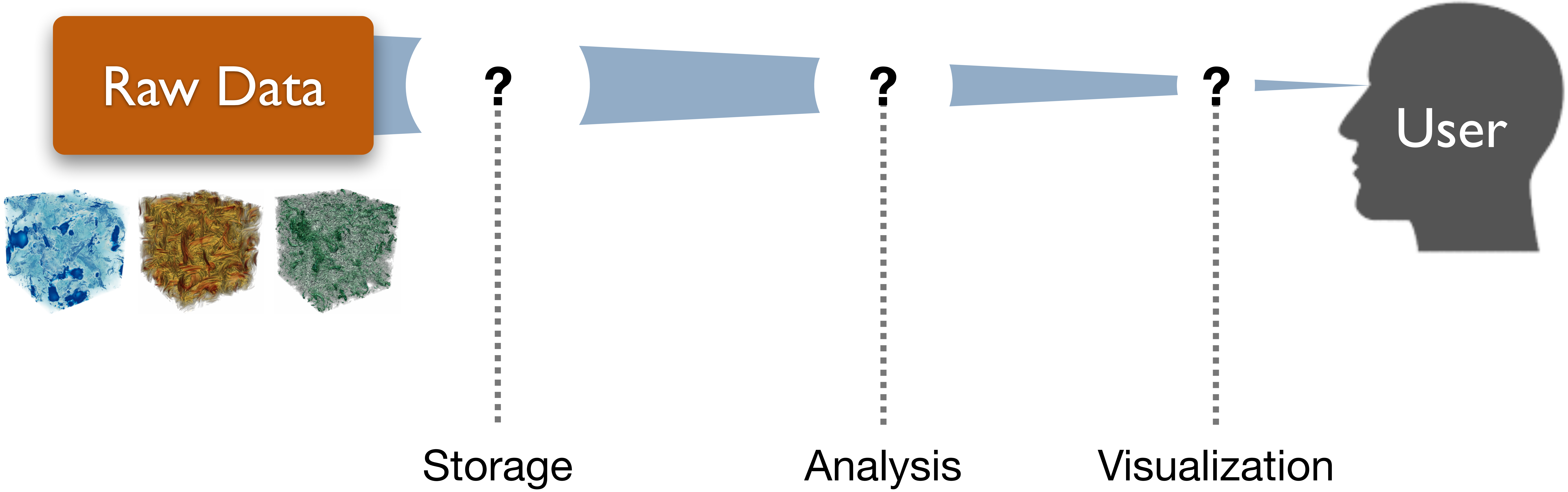


**features ⟶ output**

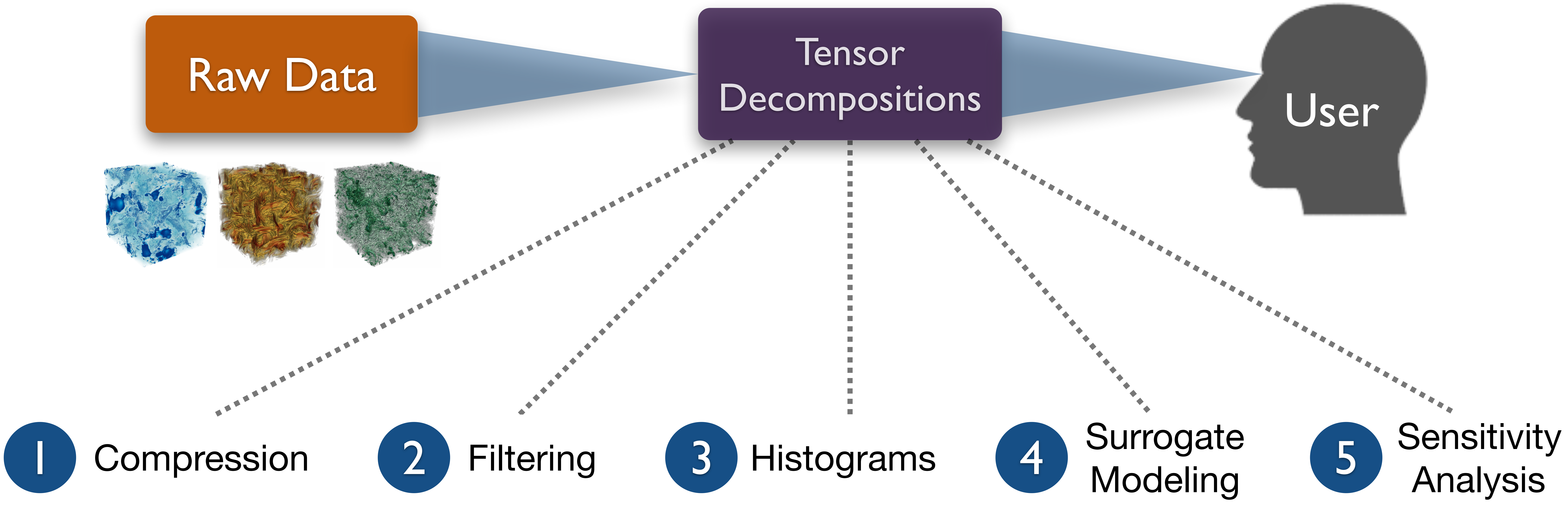
| Latitude | Longitude | Square feet | Year built | Sale date | Sale price |
|---|---|---|---|---|---|
| 40.7225368 | -73.9776789 | 3542 | 1899 | 21 | 1800000 |
| 40.7241446 | -73.9760039 | 5725 | 1910 | 295 | 426000 |
| 40.722516 | -73.985741 | 8722 | 1900 | 250 | 1600000 |
| 40.724563 | -73.982373 | 6200 | 1920 | 119 | 1060000 |
| 40.72923 | -73.9783678 | 12613 | 1900 | 321 | 3200000 |
| 40.7267061 | -73.977258 | 21328 | 1910 | 36 | 3000000 |
| 40.7240413 | -73.9782925 | 5197 | 2005 | 273 | 1110000 |
| 40.7263358 | -73.9799891 | 600 | 1928 | 287 | 660000 |
| 40.7263358 | -73.9799891 | 700 | 1928 | 85 | 685000 |
| 40.7263358 | -73.9799891 | 1042 | 1928 | 181 | 652500 |
| 40.7263358 | -73.9799891 | 401 | 1928 | 29 | 200000 |
| 40.7127837 | -74.0059413 | 850 | 1928 | 112 | 2250000 |
| 40.7251057 | -73.9783557 | 3540 | 1900 | 280 | 1100000 |
| 40.743722 | -74.0028449 | 2340 | 1930 | 152 | 1500000 |
| 40.745718 | -74.0000878 | 5400 | 1901 | 342 | 3400000 |
| 40.743898 | -74.002215 | 3264 | 1910 | 252 | 2525000 |
| 40.743847 | -73.997739 | 4825 | 1901 | 177 | 2700000 |
| 40.74518 | -74.004588 | 3232 | 1910 | 303 | 1850000 |
| 40.742158 | -74.0028033 | 8525 | 1910 | 2 | 18750 |
| 40.742158 | -74.0028033 | 8525 | 1910 | 2 | 18750 |
| 40.744349 | -74.001472 | 5480 | 1900 | 216 | 1500000 |
| 40.74452 | -74.000037 | 9690 | 1900 | 2 | 40312 |
| 40.74452 | -74.000037 | 9690 | 1900 | 2 | 40312 |
| 40.740689 | -74.000017 | 5304 | 1910 | 315 | 2300000 |
| 40.7434121 | -73.9995002 | 8955 | 1944 | 154 | 2470000 |
| 40.743065 | -73.99781 | 8540 | 1920 | 197 | 659360 |
| 40.743094 | -73.997885 | 11580 | 1920 | 197 | 659360 |
| 40.7433573 | -73.9983275 | 4600 | 1901 | 225 | 2600500 |
| 40.743284 | -73.997123 | 9880 | 1901 | 2 | 43312 |
| 40.743284 | -73.997123 | 9880 | 1901 | 2 | 43312 |
| 40.745495 | -73.997492 | 2499 | 1920 | 41 | 2650000 |
| 40.7462831 | -73.9973351 | 3307 | 1900 | 29 | 665000 |
| 40.738686 | -73.997248 | 4165 | 1910 | 295 | 1600000 |

VISUALIZATIONANDMULTIMEDIALAB

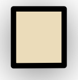
# Challenges



# Contributions



# Tensors



$x$

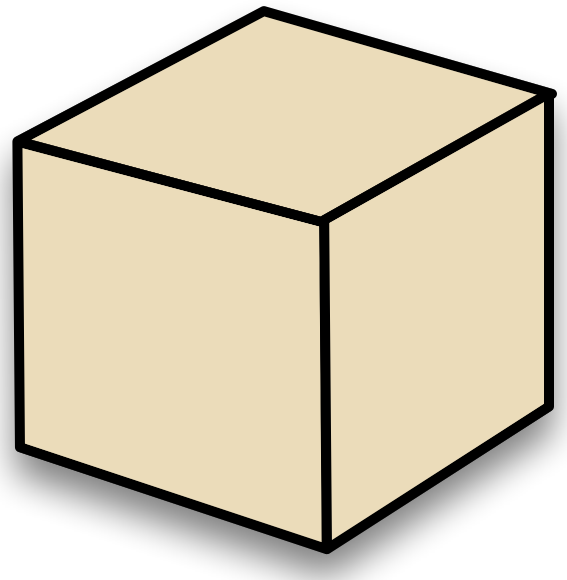
Number: 0D tensor

$i \mapsto x$

Vector: 1D tensor

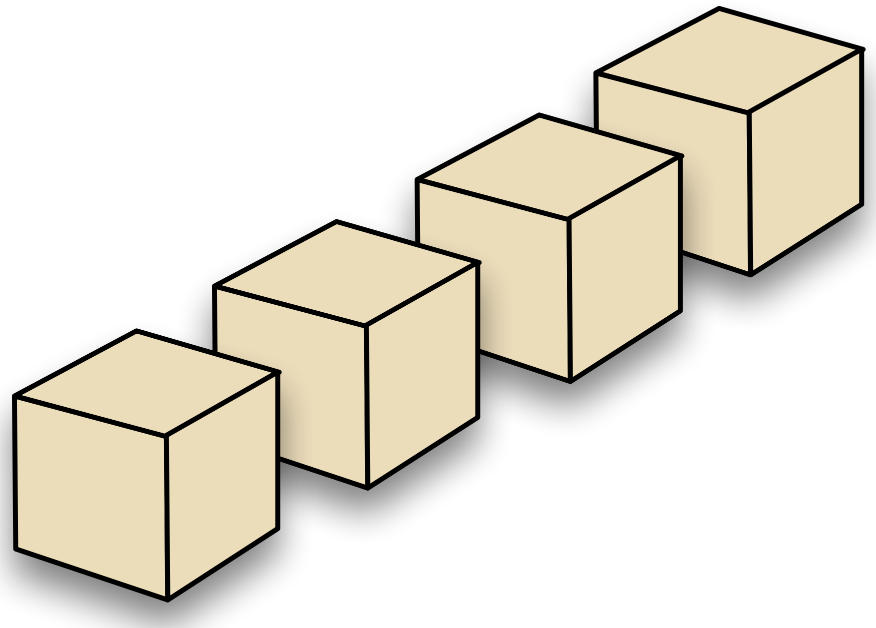
$(i, j) \mapsto x$

Matrix: 2D tensor



Cube: 3D tensor

$(i, j, k) \mapsto x$



Hypercube: 4D tensor

$(i, j, k, l) \mapsto x$

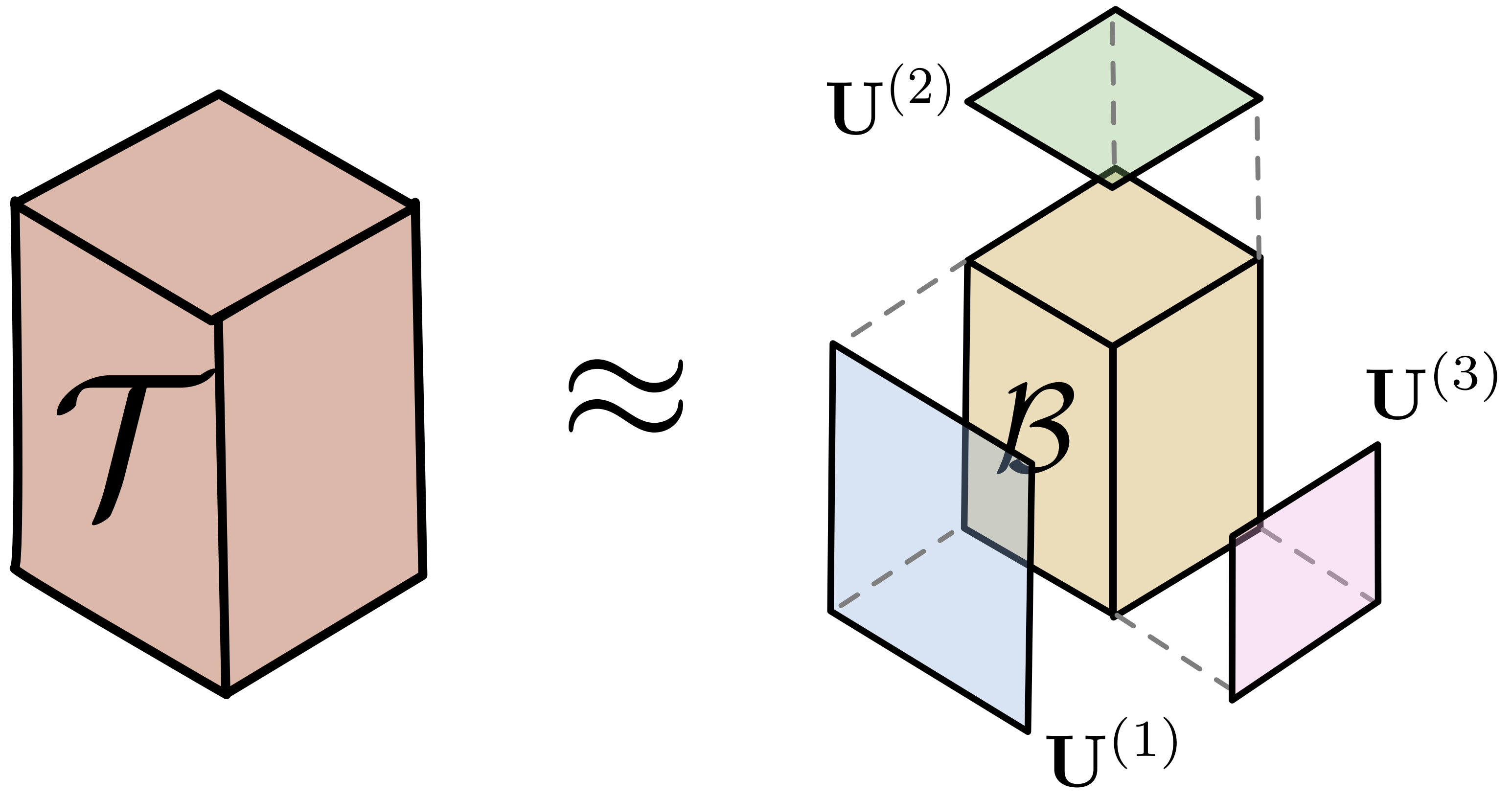
# Tensors: Decomposition

- **Tensors** $\longrightarrow$ **smaller tensors** that interact together

Tucker decomposition

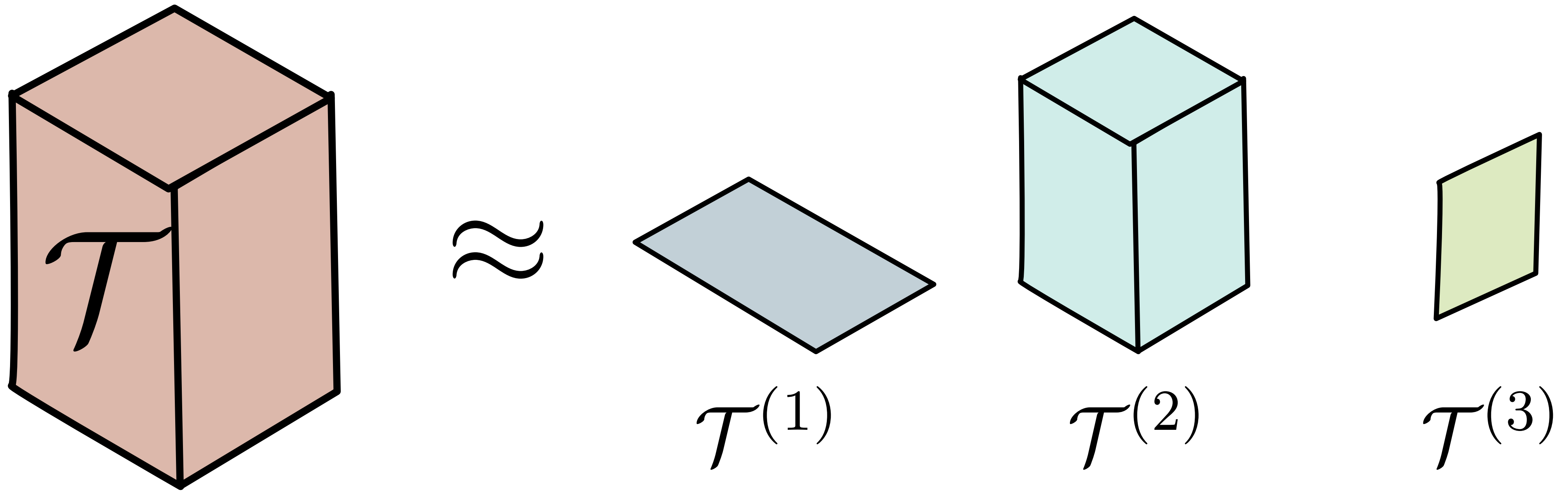
**[Tucker '66]**

**[de Lathauwer et al. '00]**




Tensor train decomposition

**[Oseledets '11]**



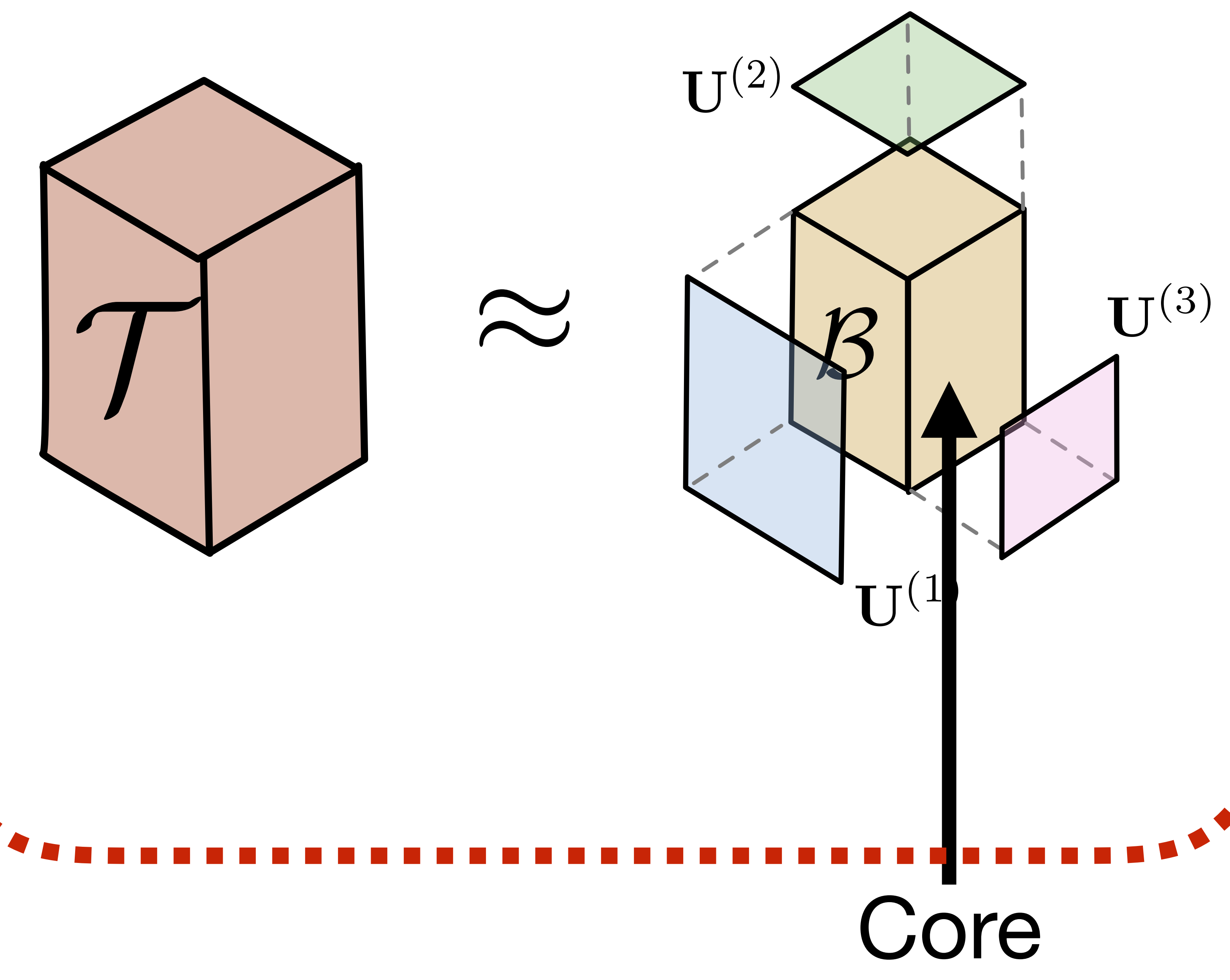
# Tensors: Decomposition

- **Tensors** ⟶ **smaller tensors** that interact together

Tucker decomposition

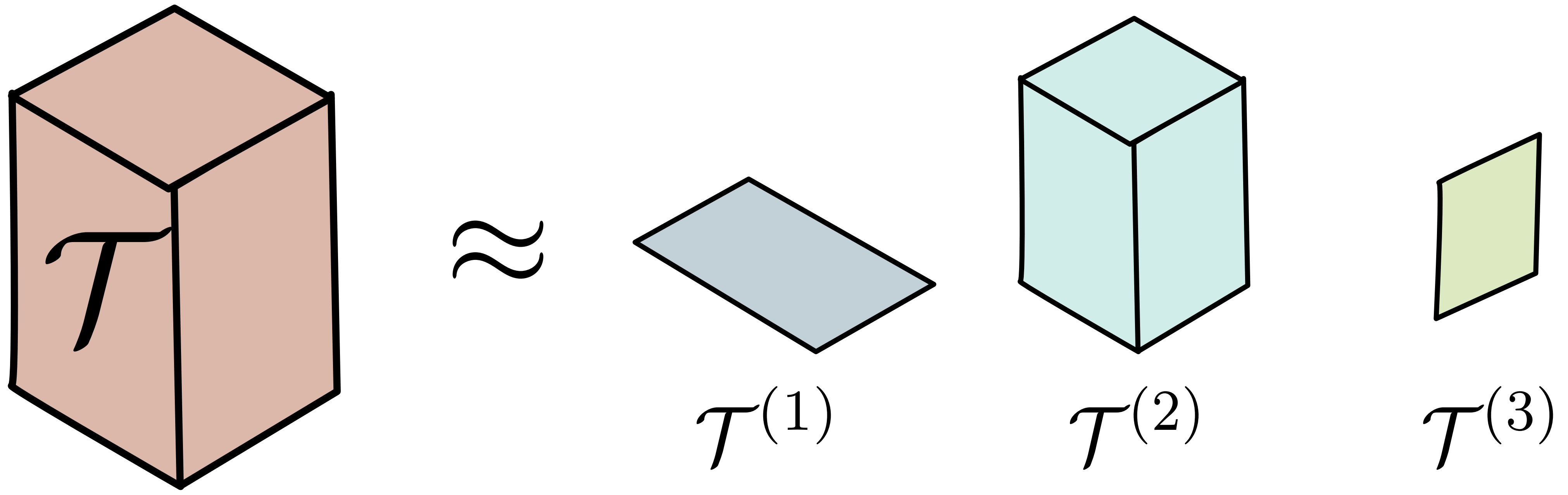
**[Tucker '66]**

**[de Lathauwer et al. '00]**




Tensor train decomposition

**[Oseledets '11]**




VISUALIZATIONAND MULTIMEDIALAB

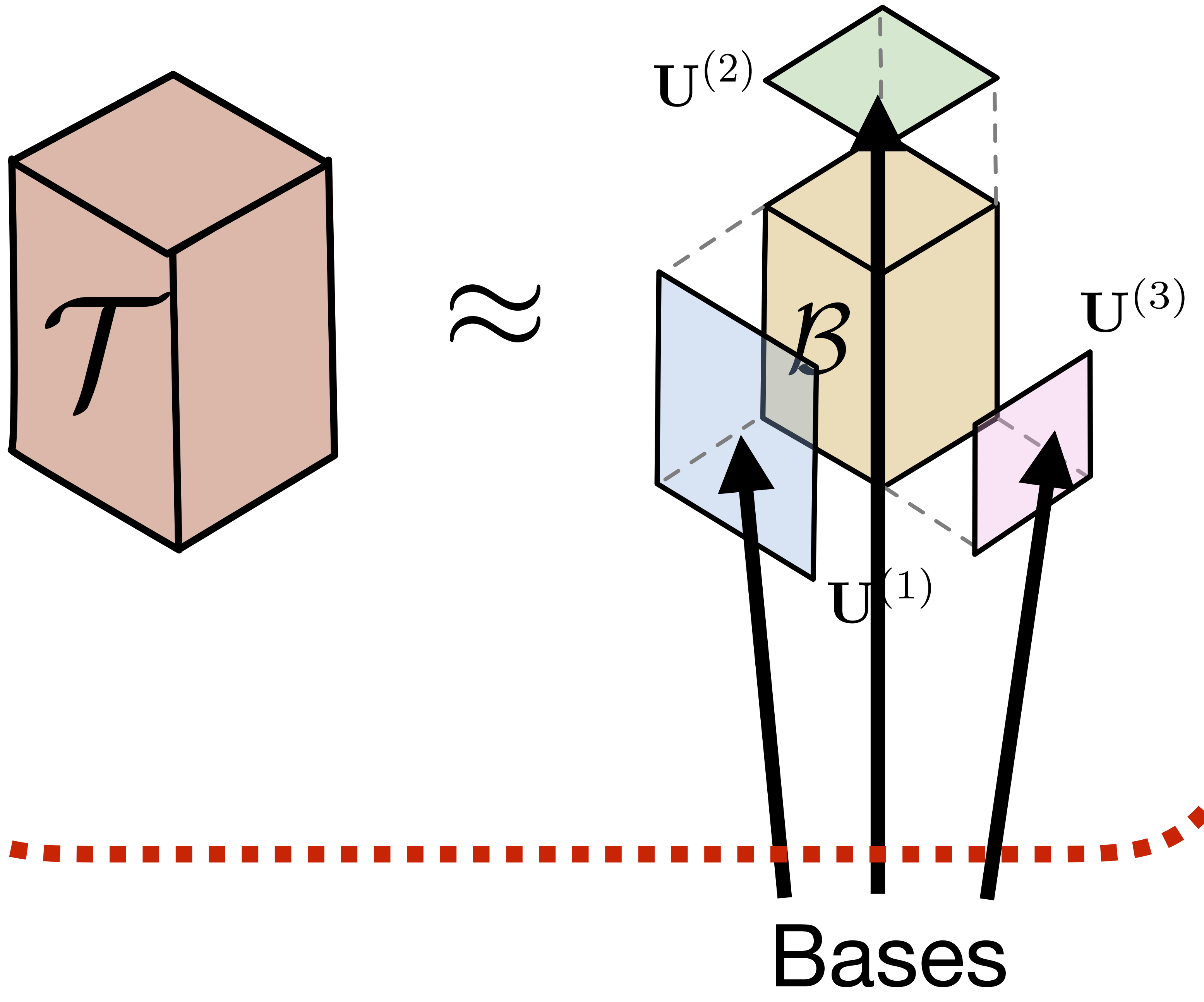
# Tensors: Decomposition

- **Tensors** $\longrightarrow$ **smaller tensors** that interact together

## Tucker decomposition

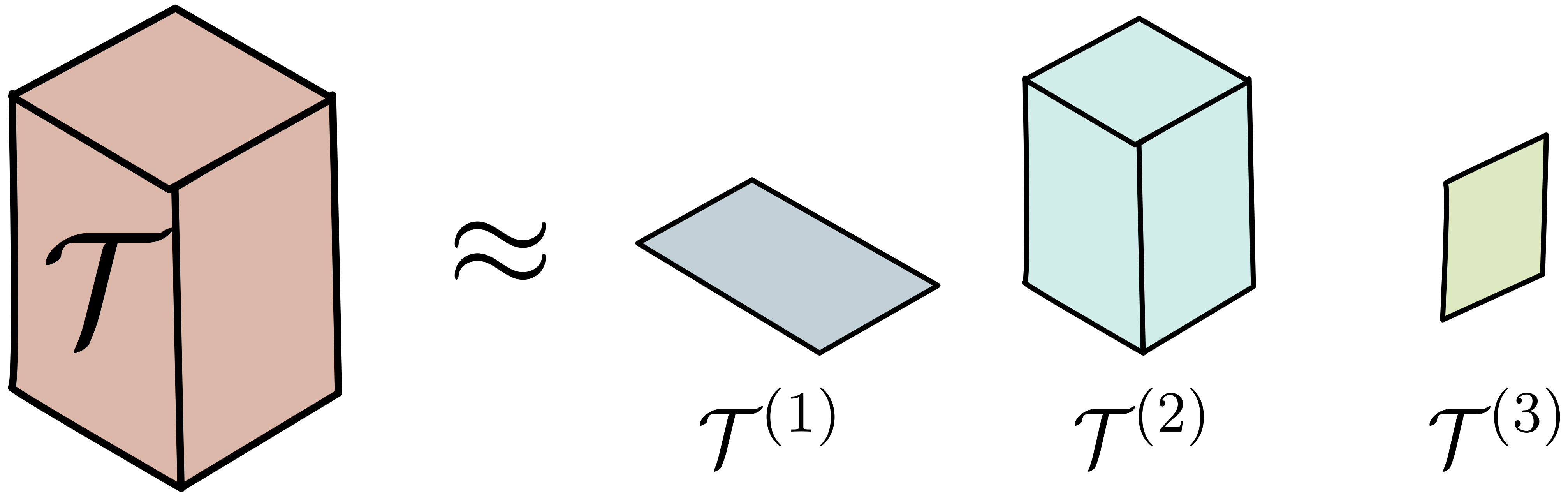
**[Tucker ’66]**

**[de Lathauwer et al. ’00]**




## Tensor train decomposition

**[Oseledets ’11]**



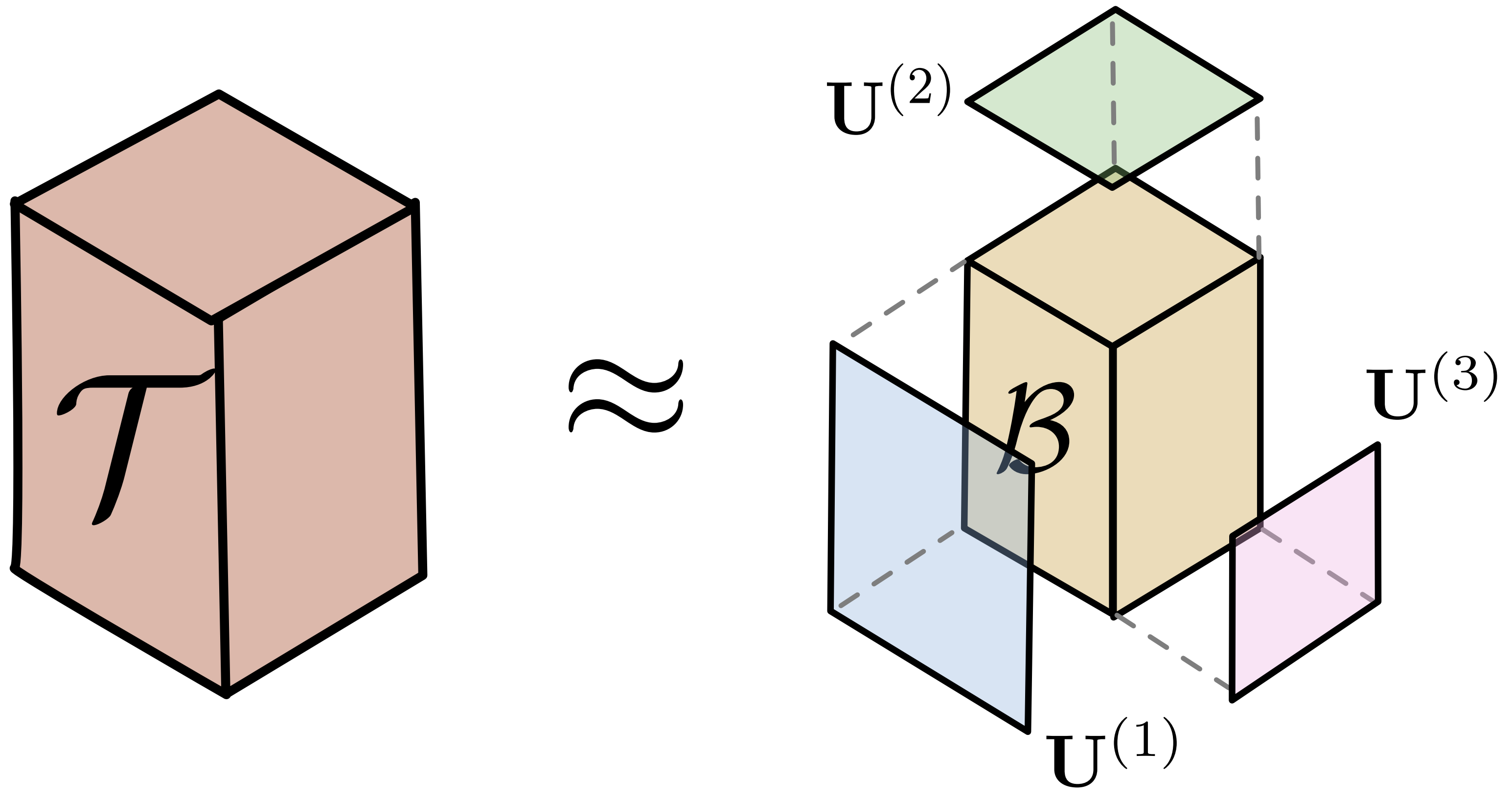
# Tensors: Decomposition

- **Tensors** $\longrightarrow$ **smaller tensors** that interact together

Tucker decomposition

**[Tucker ’66]**

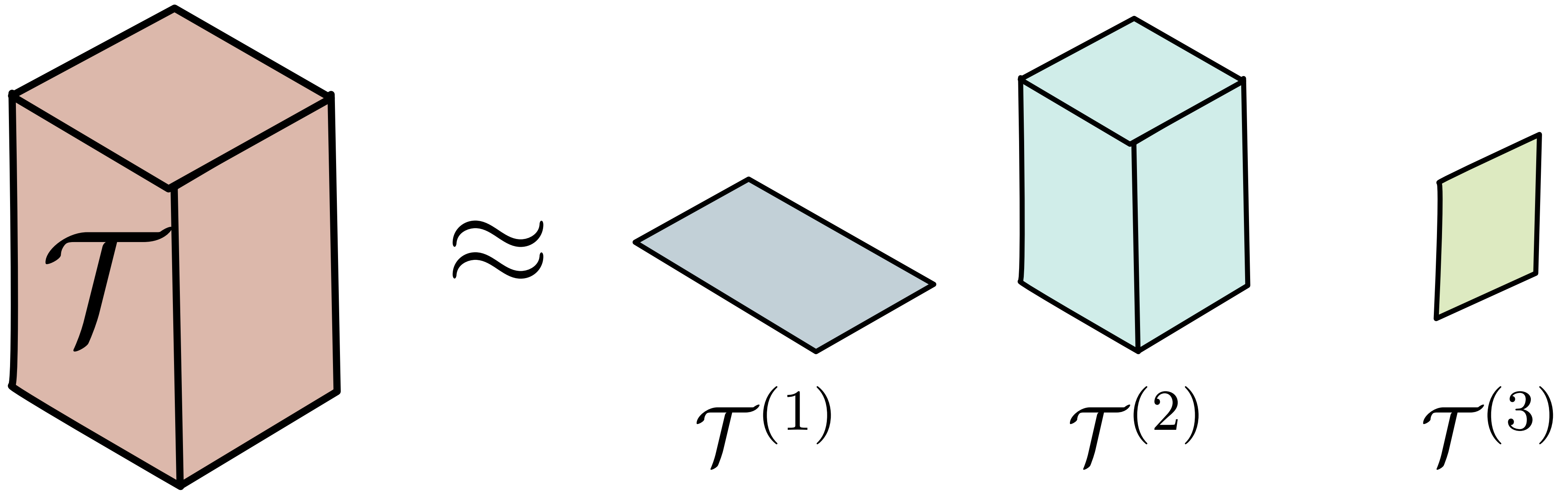
**[de Lathauwer et al. ’00]**




Best for **few dimensions**

Tensor train decomposition

**[Oseledets ’11]**



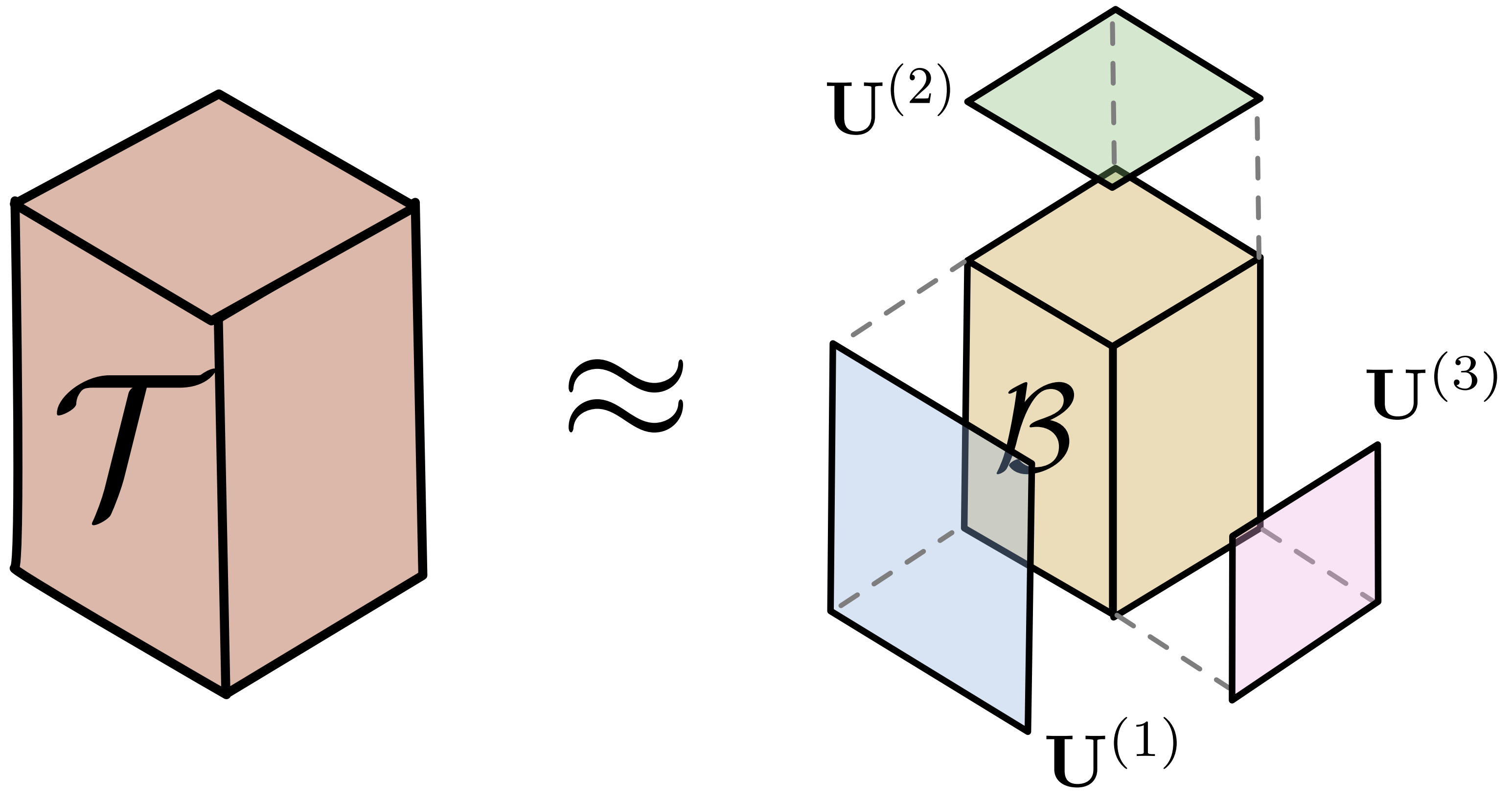
# Tensors: Decomposition

- **Tensors** $\longrightarrow$ **smaller tensors** that interact together

Tucker decomposition

**[Tucker '66]**

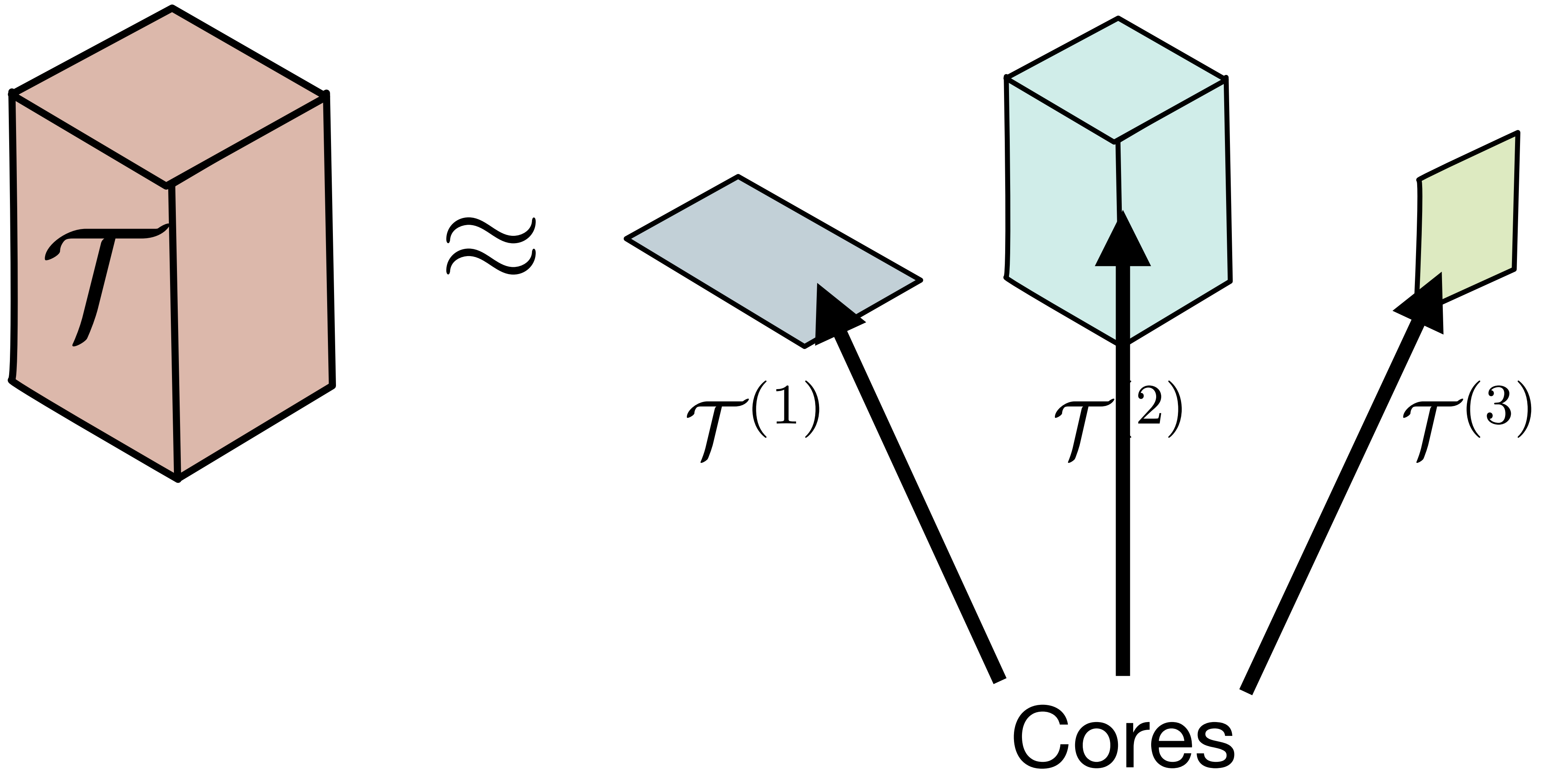
**[de Lathauwer et al. '00]**




Best for **few dimensions**

Tensor train decomposition

**[Oseledets '11]**



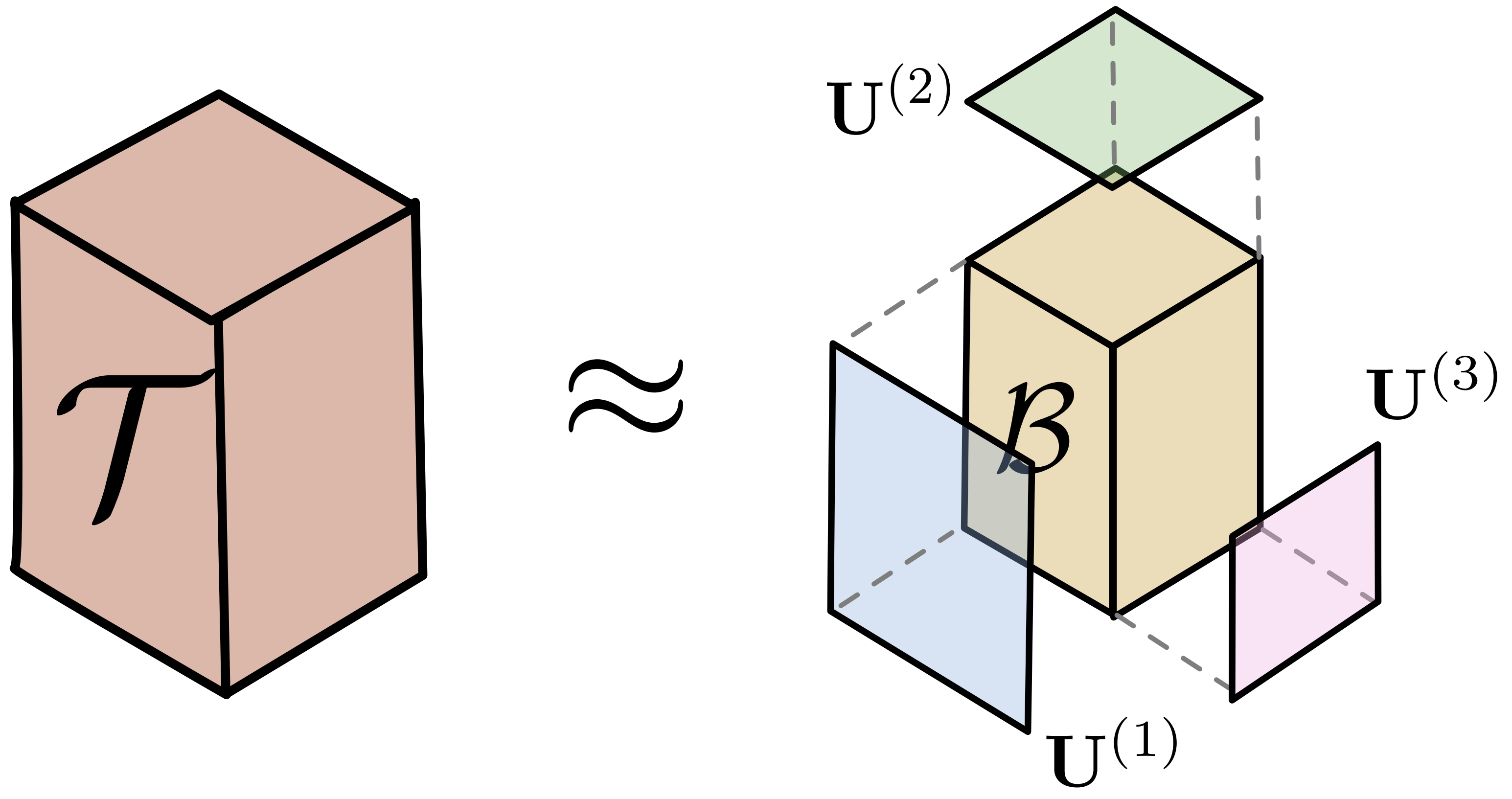
# Tensors: Decomposition

- **Tensors** $\longrightarrow$ **smaller tensors** that interact together

## Tucker decomposition

**[Tucker '66]**

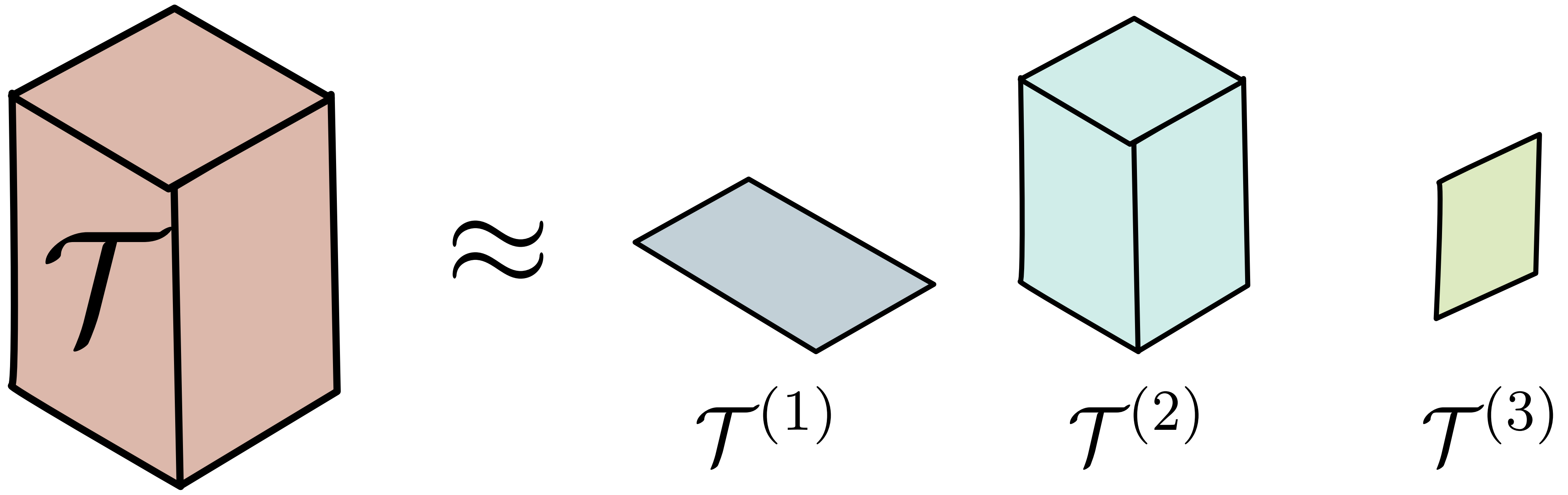
**[de Lathauwer et al. '00]**




Best for **few dimensions**

## Tensor train decomposition

**[Oseledets '11]**

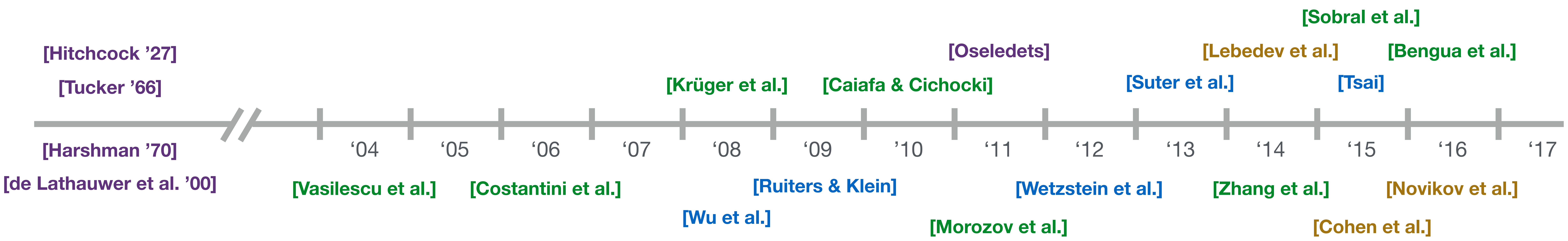



Best for **many dimensions**

VISUALIZATIONAND MULTIMEDIALAB

# Tensors: Timeline

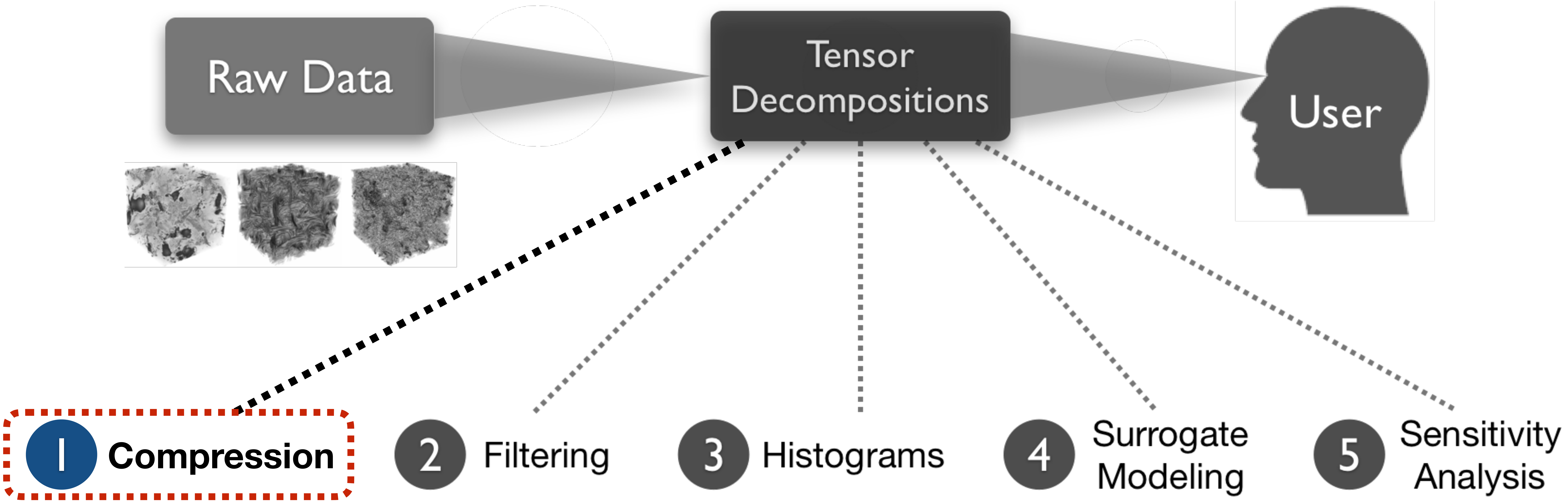
- Designed to avoid the **curse of dimensionality**
- Increasingly used in many fields




- Theory and models
- Multilinear learning and synthesis
- Compression and rendering
- Deep learning

VISUALIZATIONAND MULTIMEDIALAB

# Compression

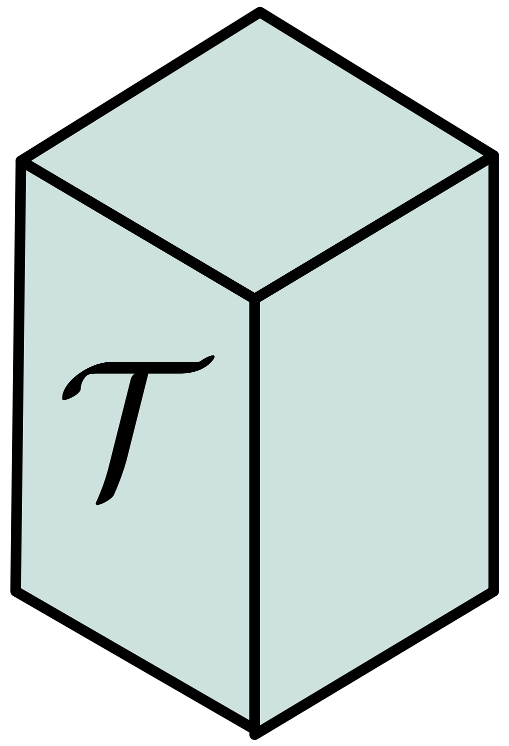



*“Lossy Volume Compression Using Tucker Truncation and Thresholding”*

VISUALIZATIONAND MULTIMEDIALAB

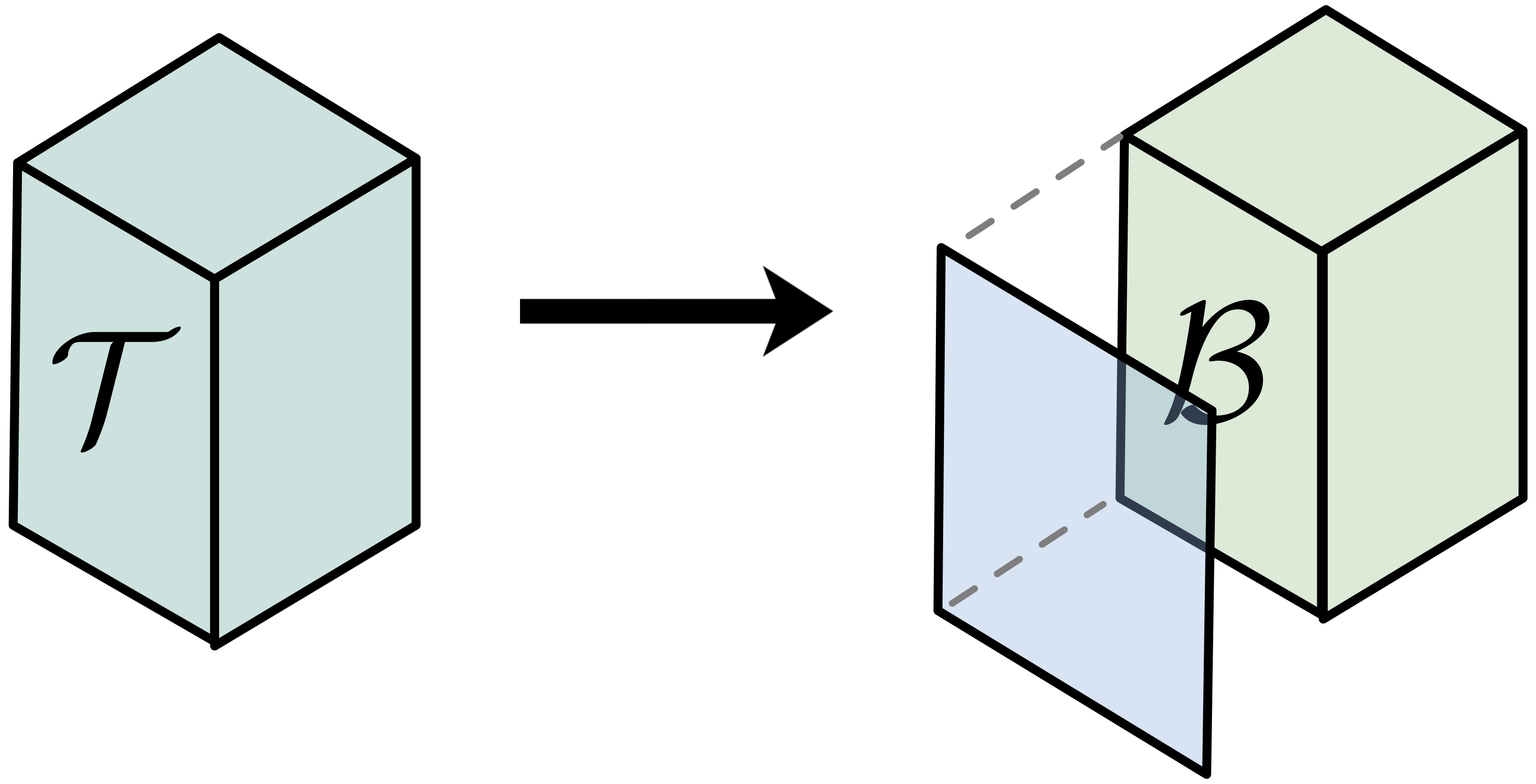
# State-of-the-Art

- Take your multidimensional data set (a tensor)
- Compute principal component analysis
  - Once per dimension



# State-of-the-Art

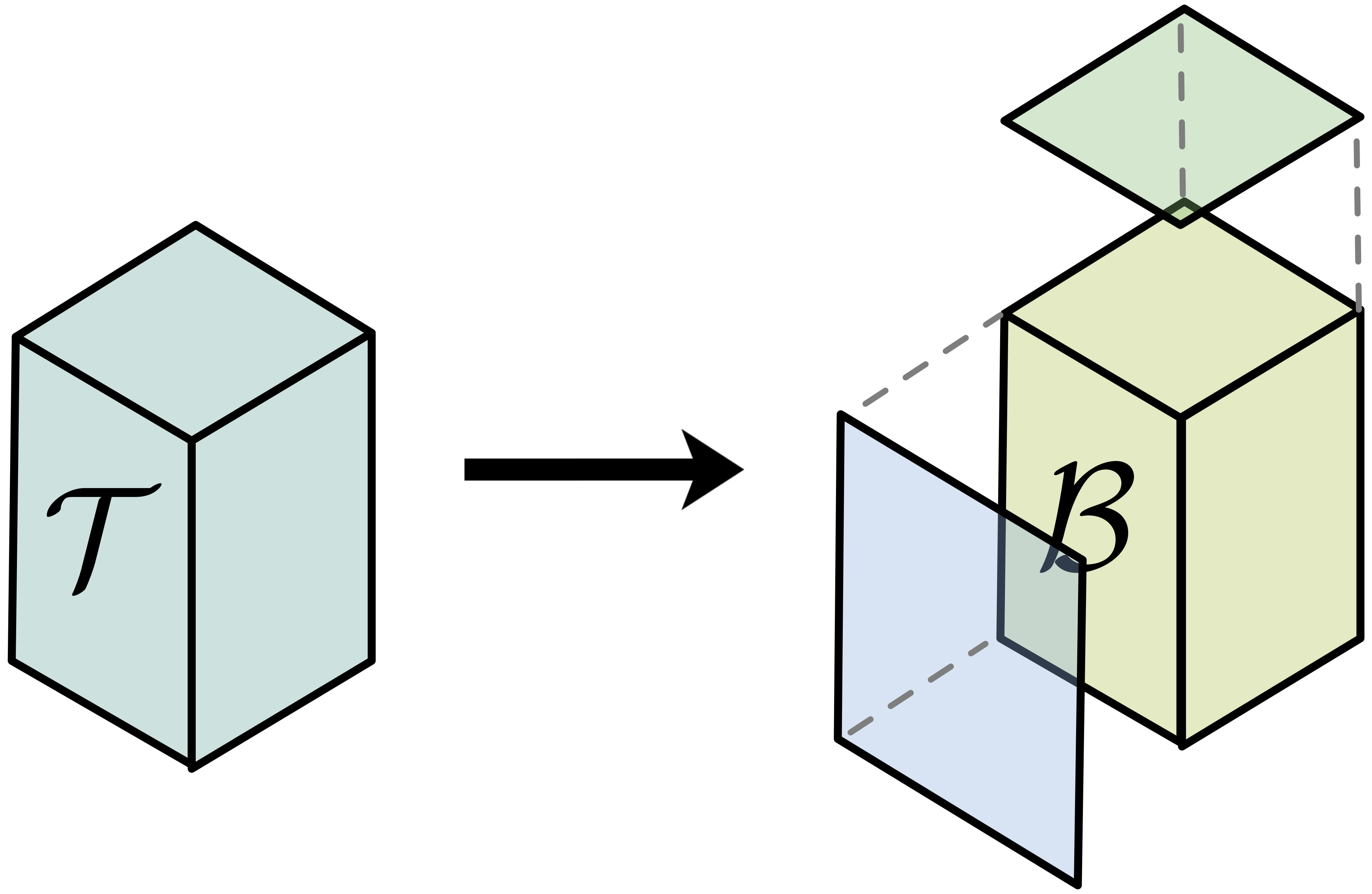
- Take your multidimensional data set (a tensor)
- Compute principal component analysis
  - Once per dimension




VISUALIZATIONAND
MULTIMEDIALAB

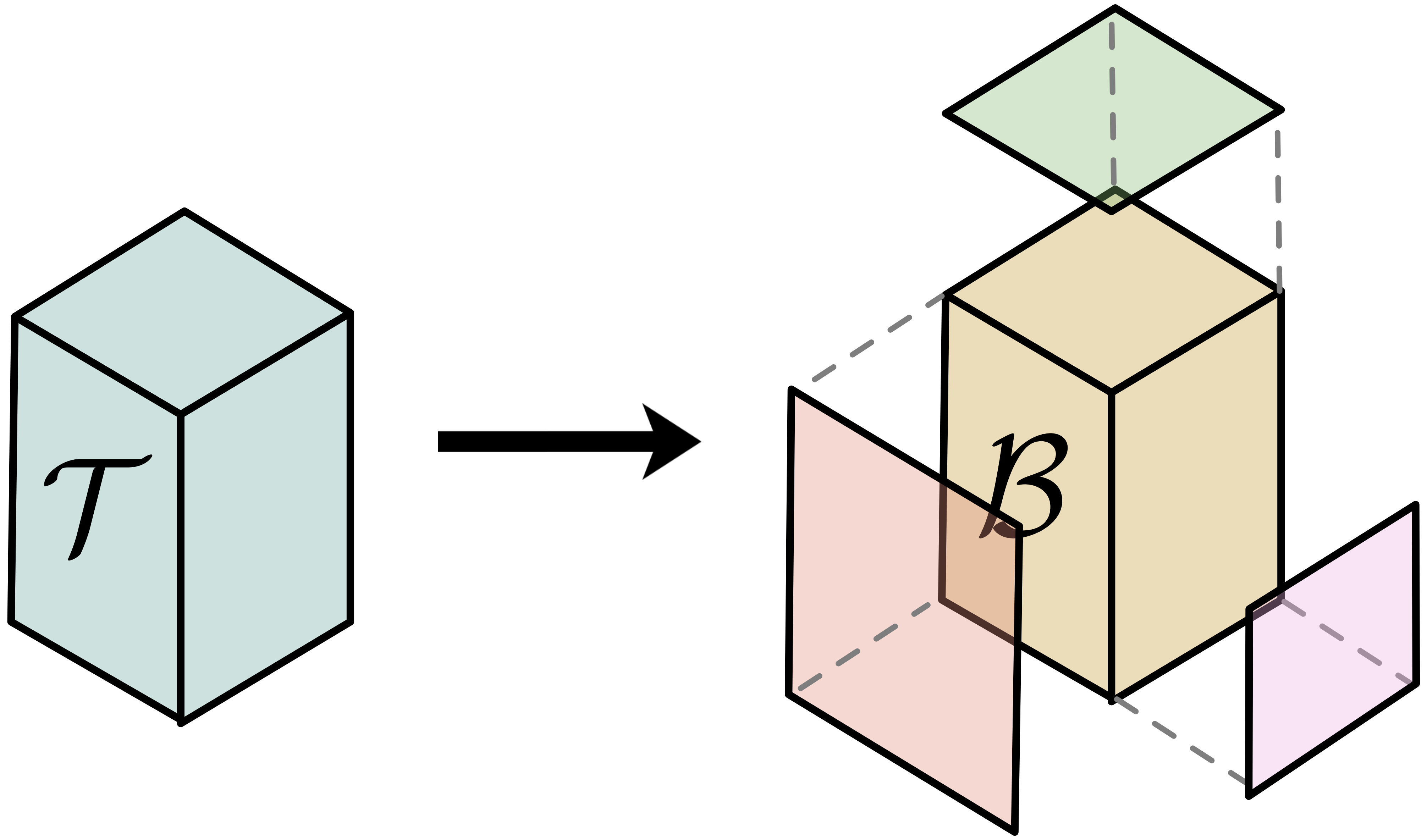
# State-of-the-Art

- Take your multidimensional data set (a tensor)
- Compute principal component analysis
  - Once per dimension



# State-of-the-Art

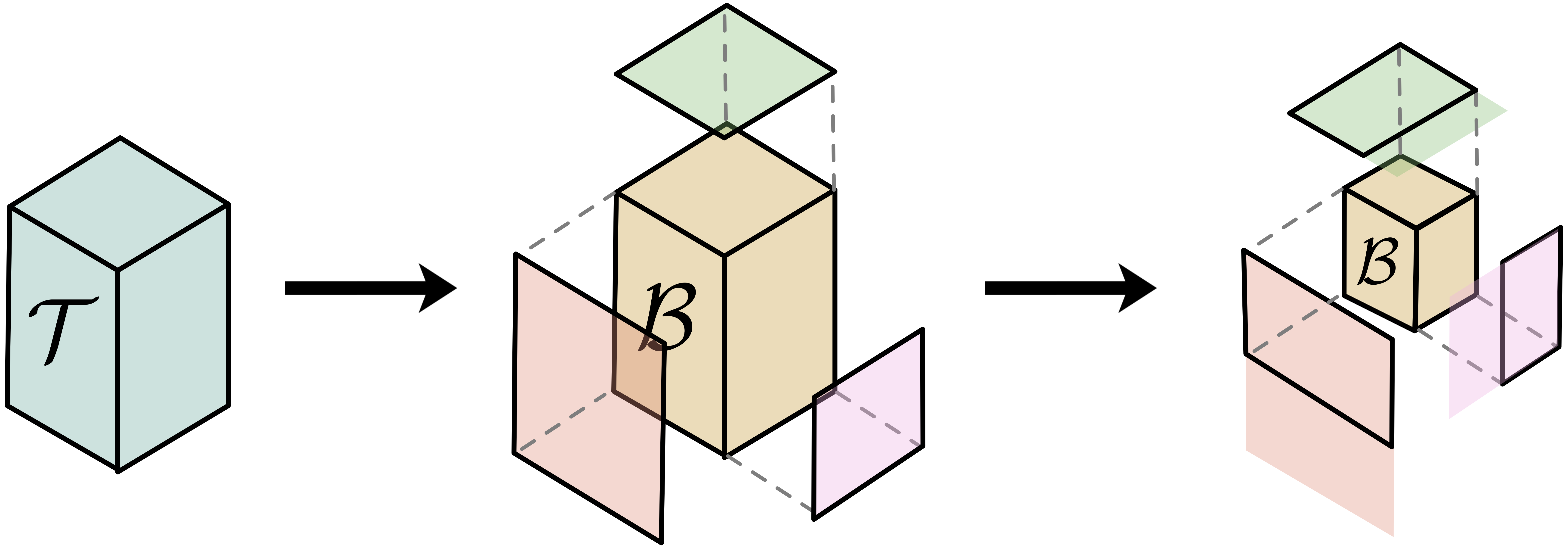
- Take your multidimensional data set (a tensor)
- Compute principal component analysis
  - Once per dimension



Tucker model

# State-of-the-Art

- Take your multidimensional data set (a tensor)
- Compute principal component analysis
  - Once per dimension
- Truncate the result




Tucker model

[Wu et al. '08]

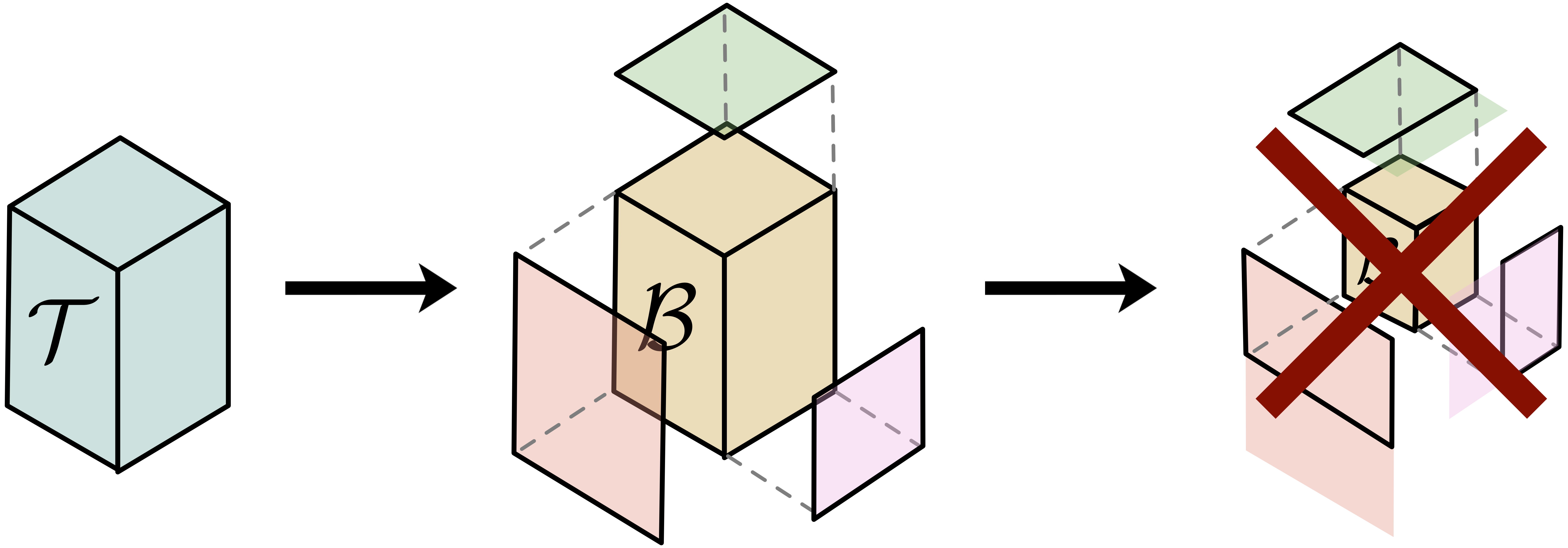
[Suter et al. '11]

[Suter et al. '13]

VISUALIZATIONAND MULTIMEDIALAB

# Proposed Approach

- Do not truncate
- Treat coefficients **based on their magnitude**, not position



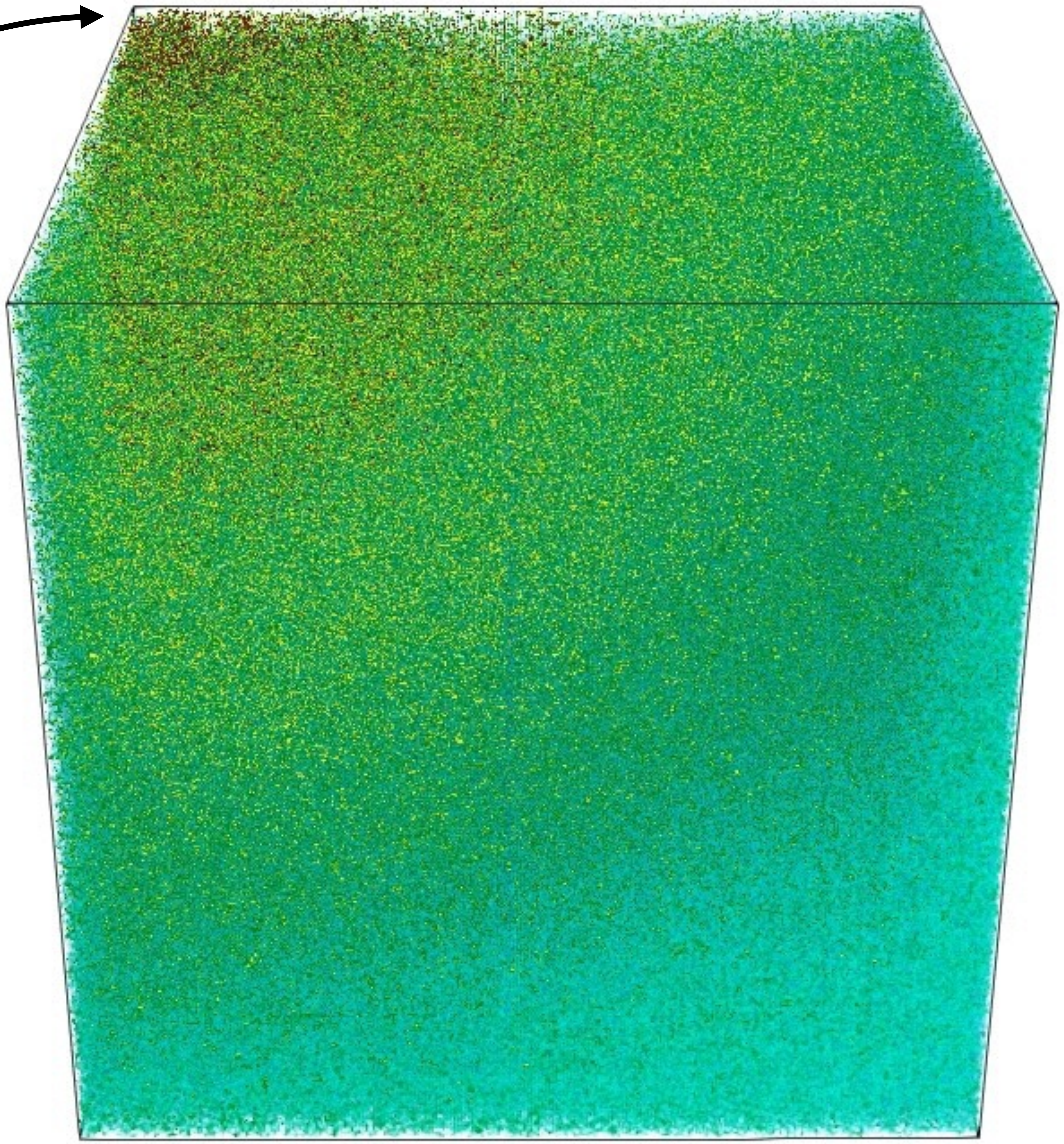

[Wu et al. '08]

[Suter et al. '11]

[Suter et al. '13]

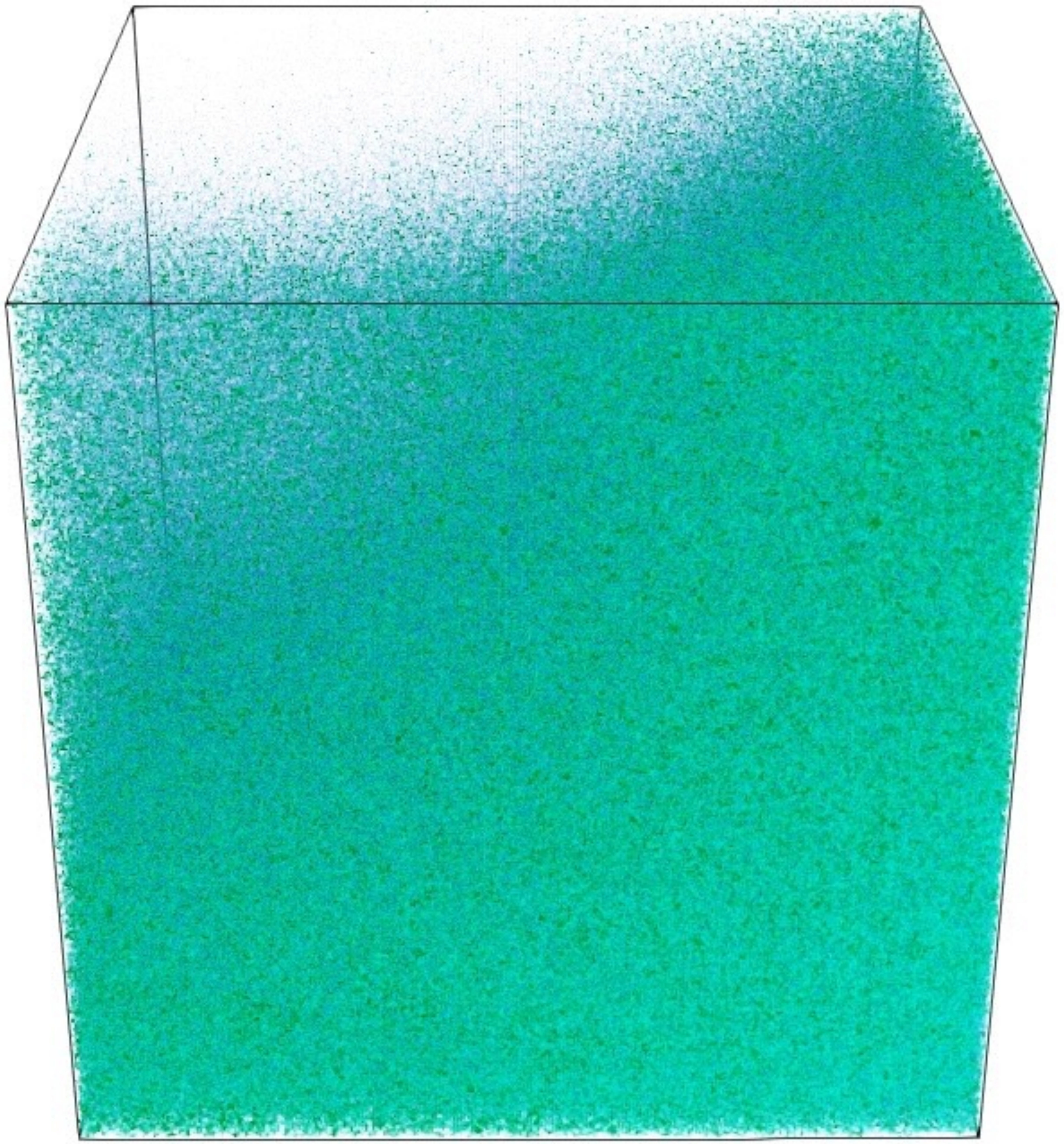
VISUALIZATIONAND MULTIMEDIALAB

“Hot corner”

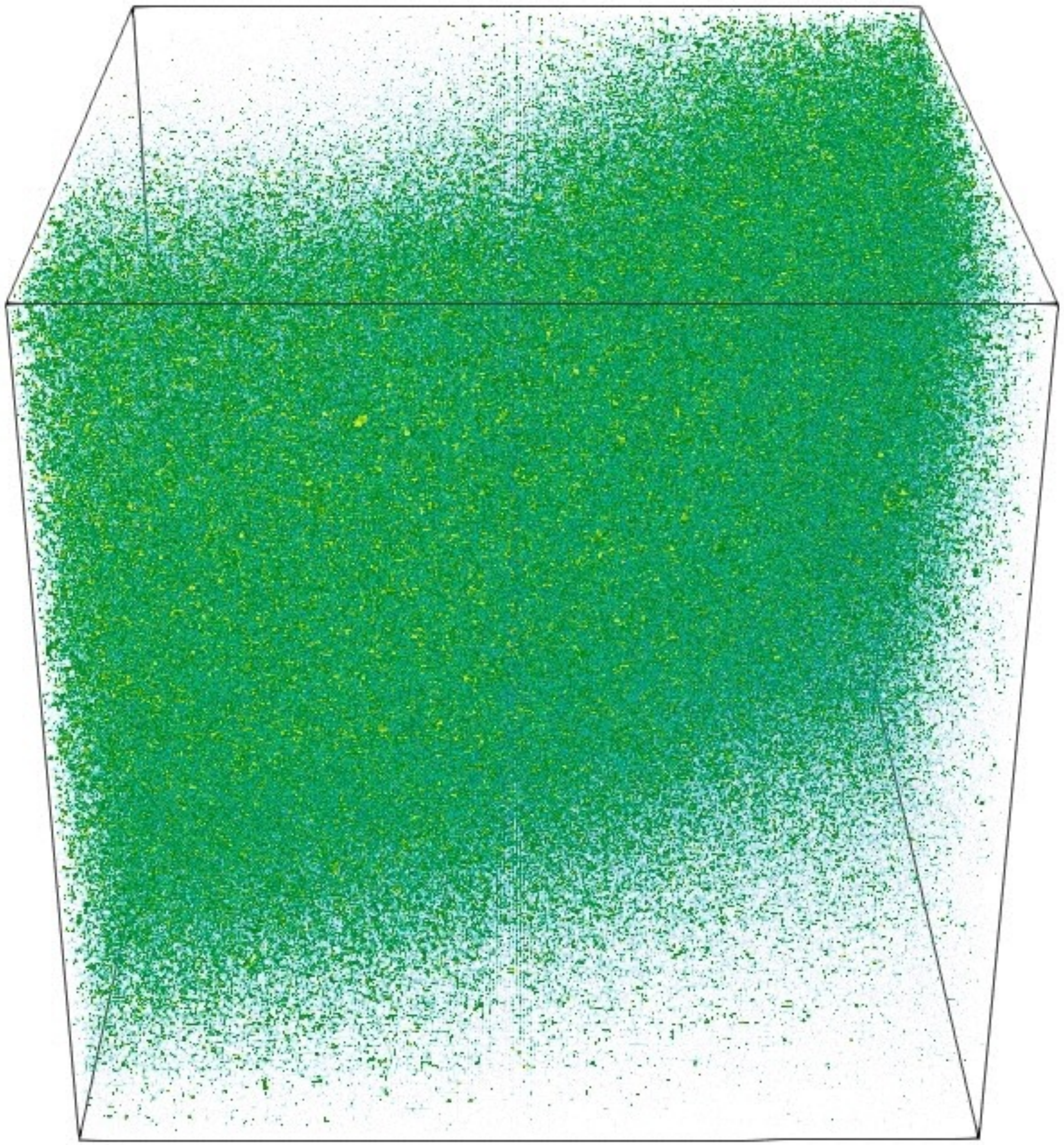


Transformed coefficients

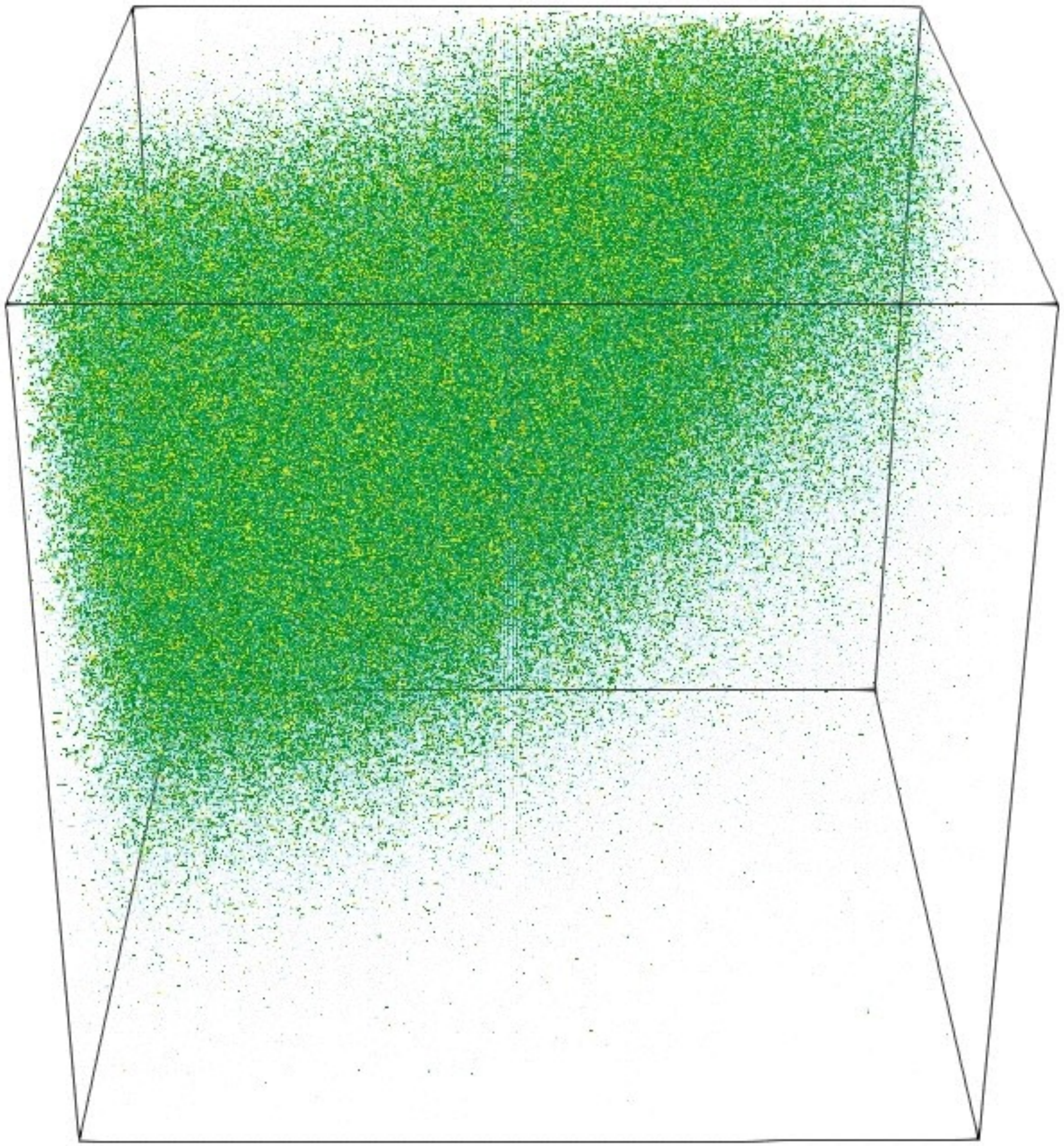
0 bits



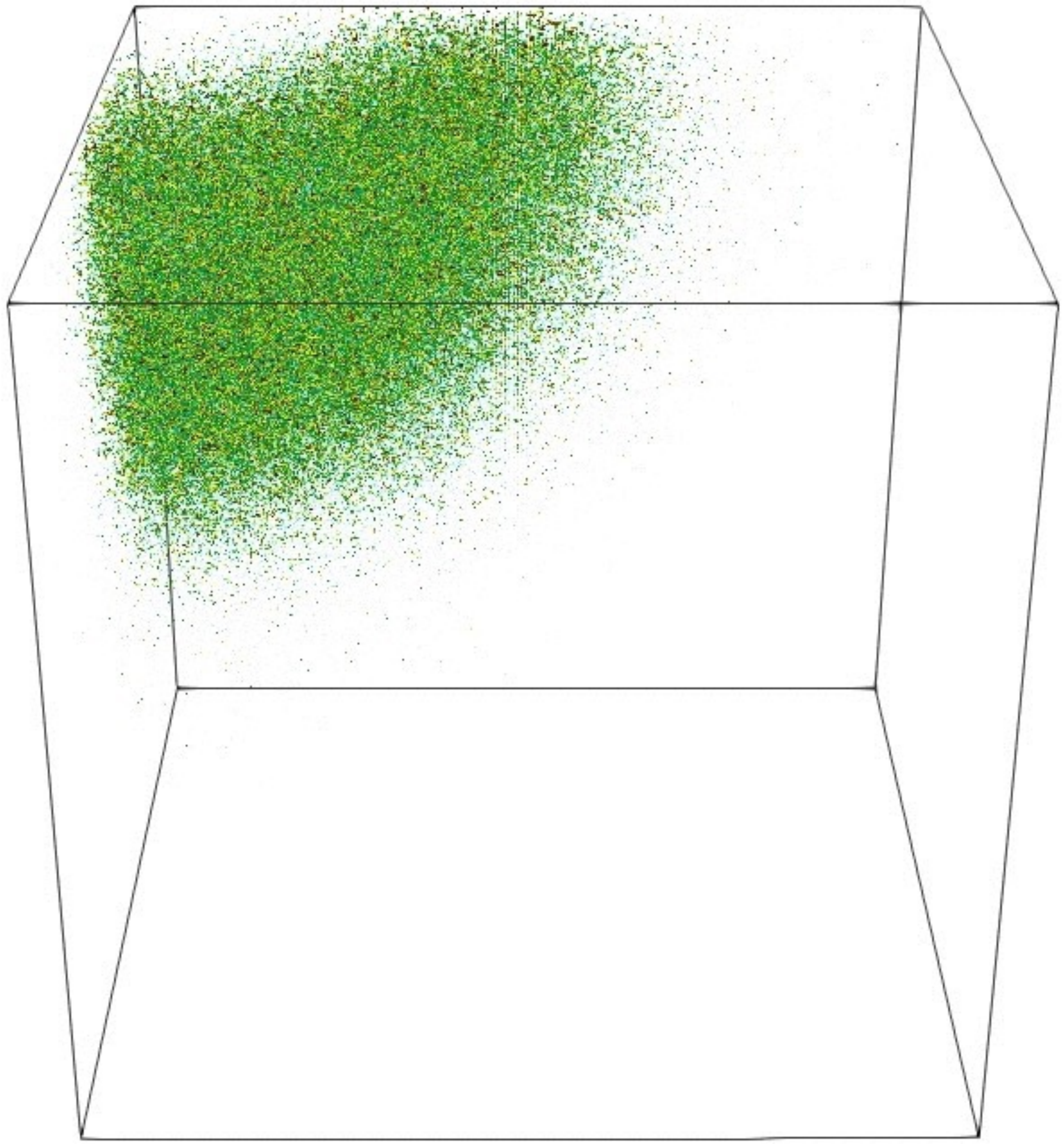
1 bit



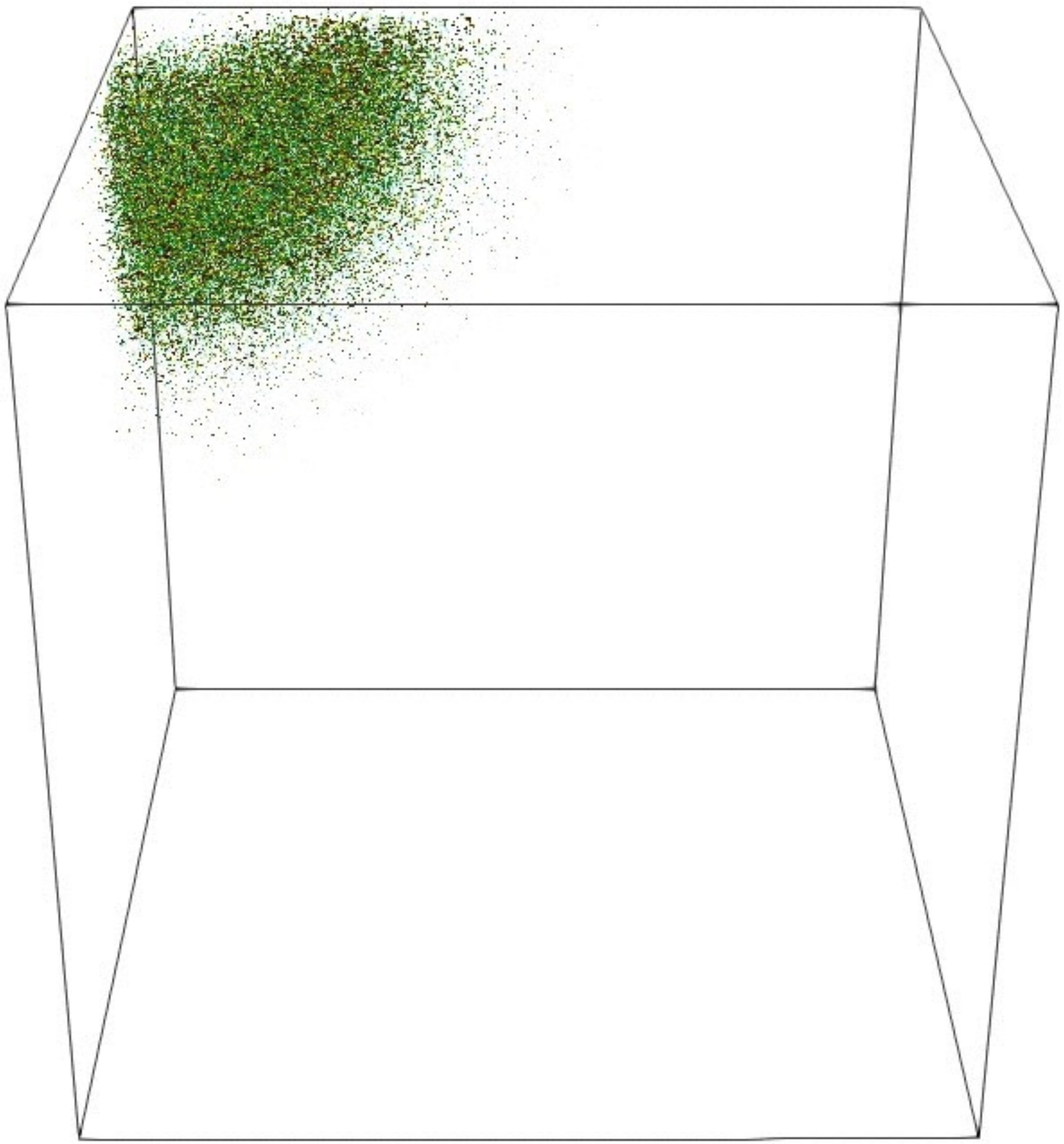
2 bits



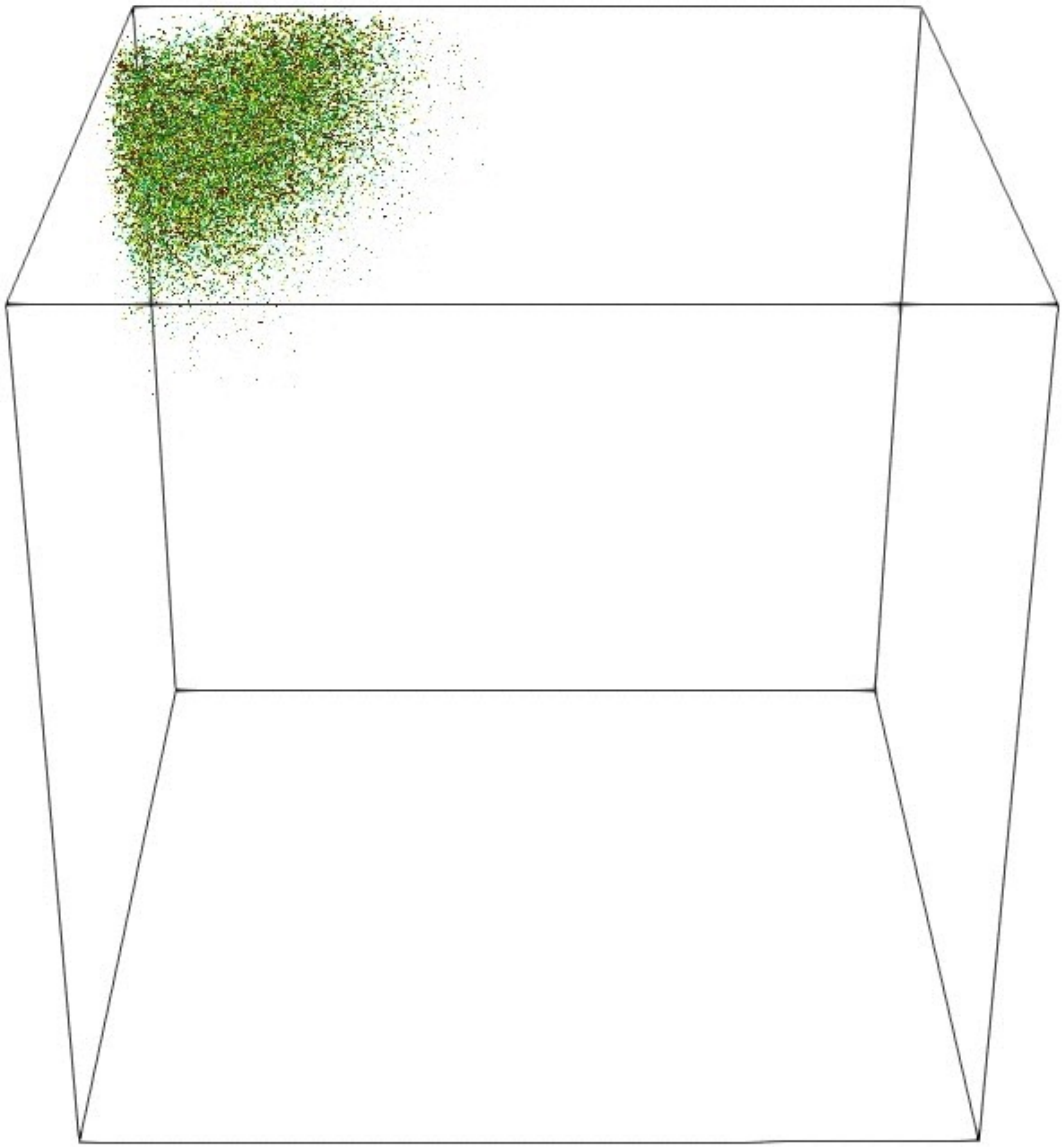
3 bits



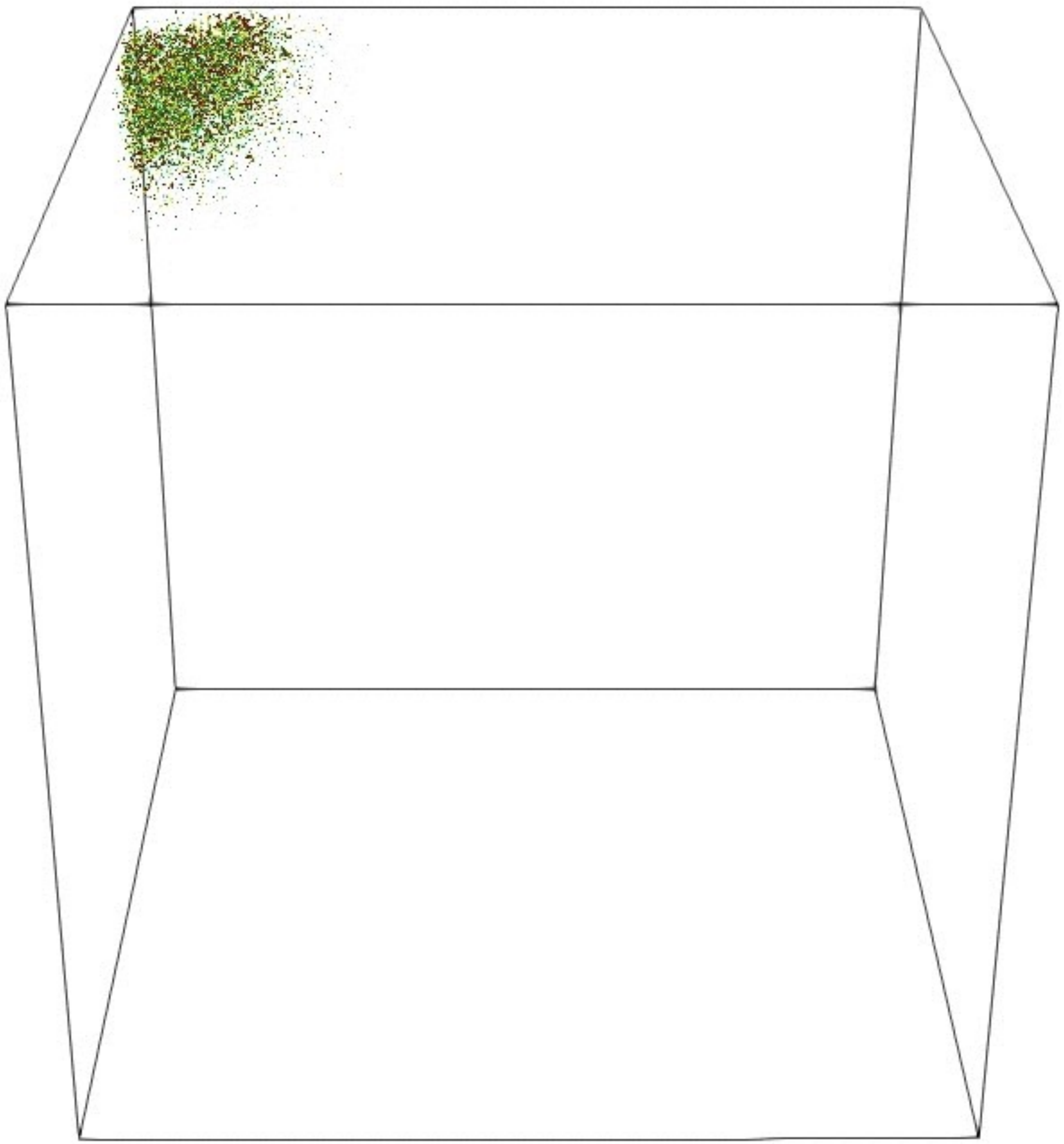
4 bits



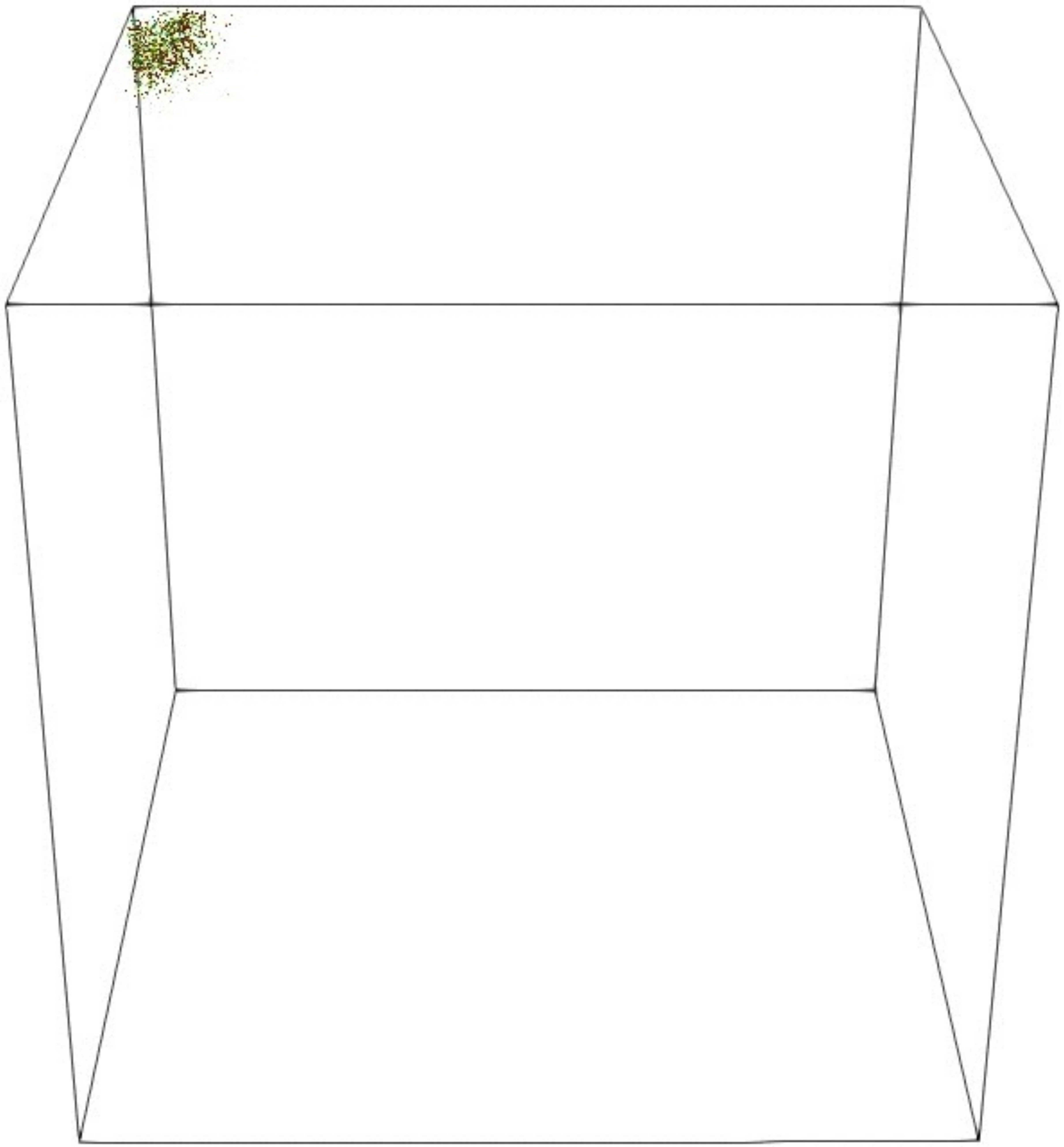
5 bits



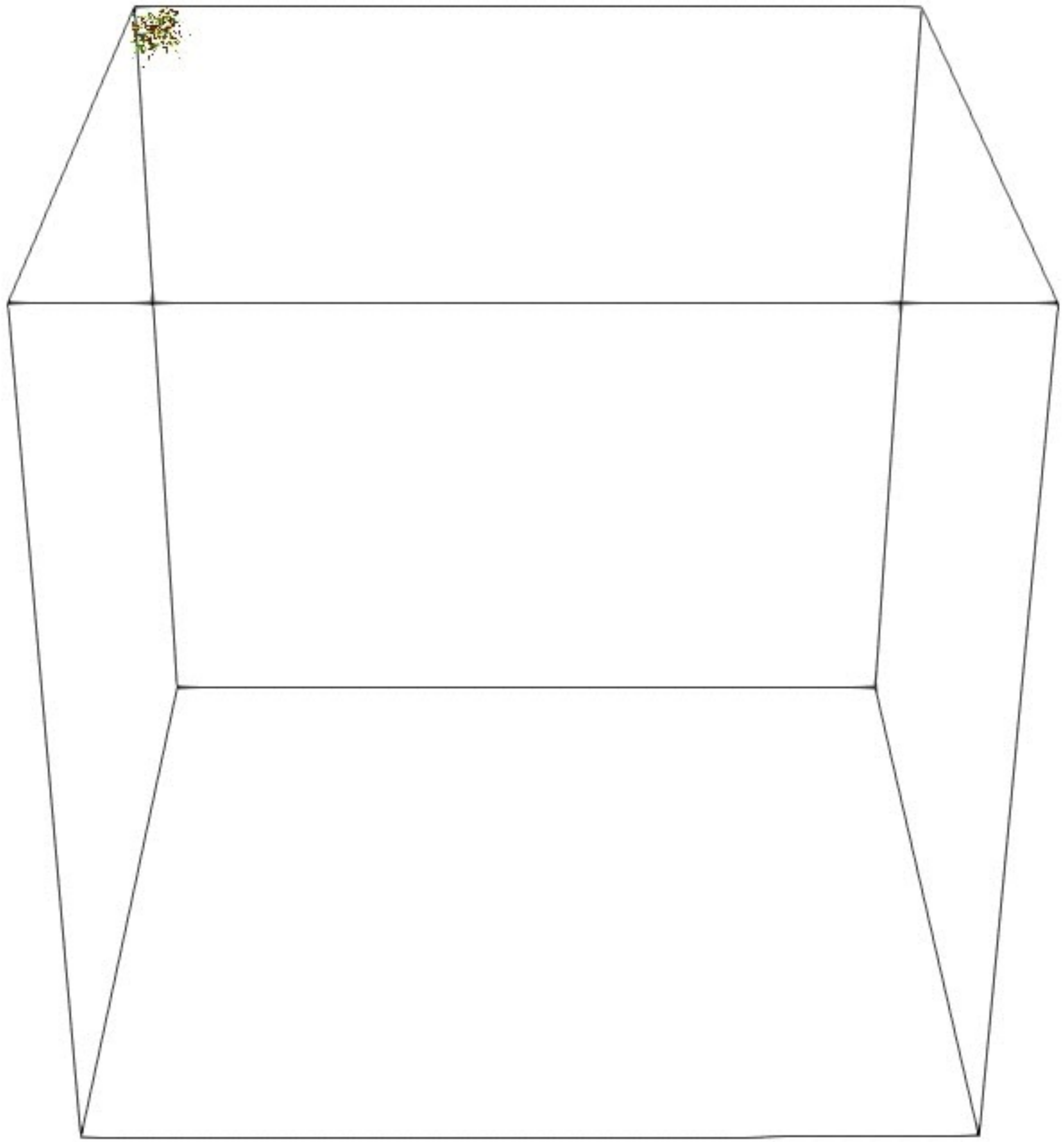
6 bits

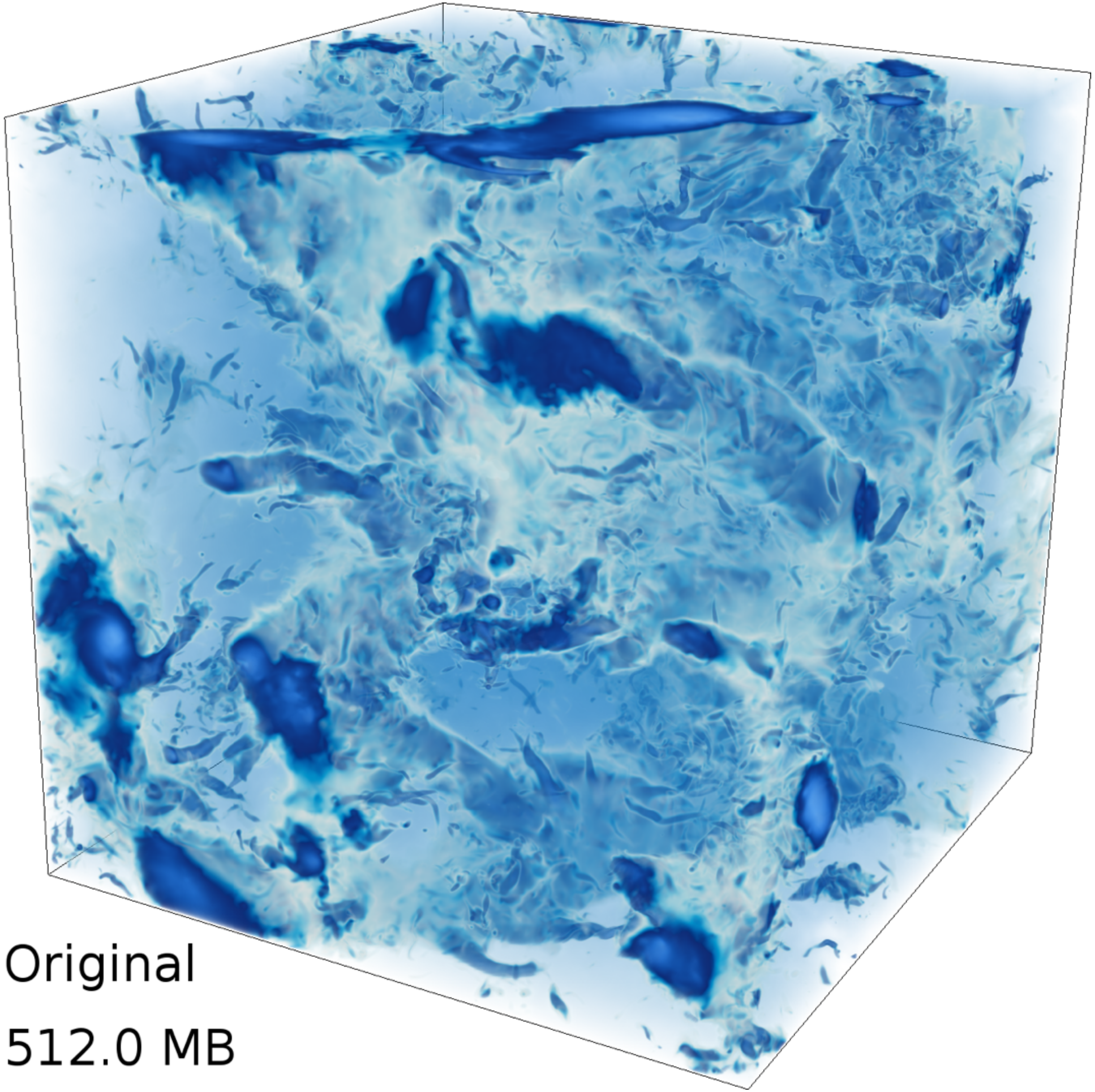


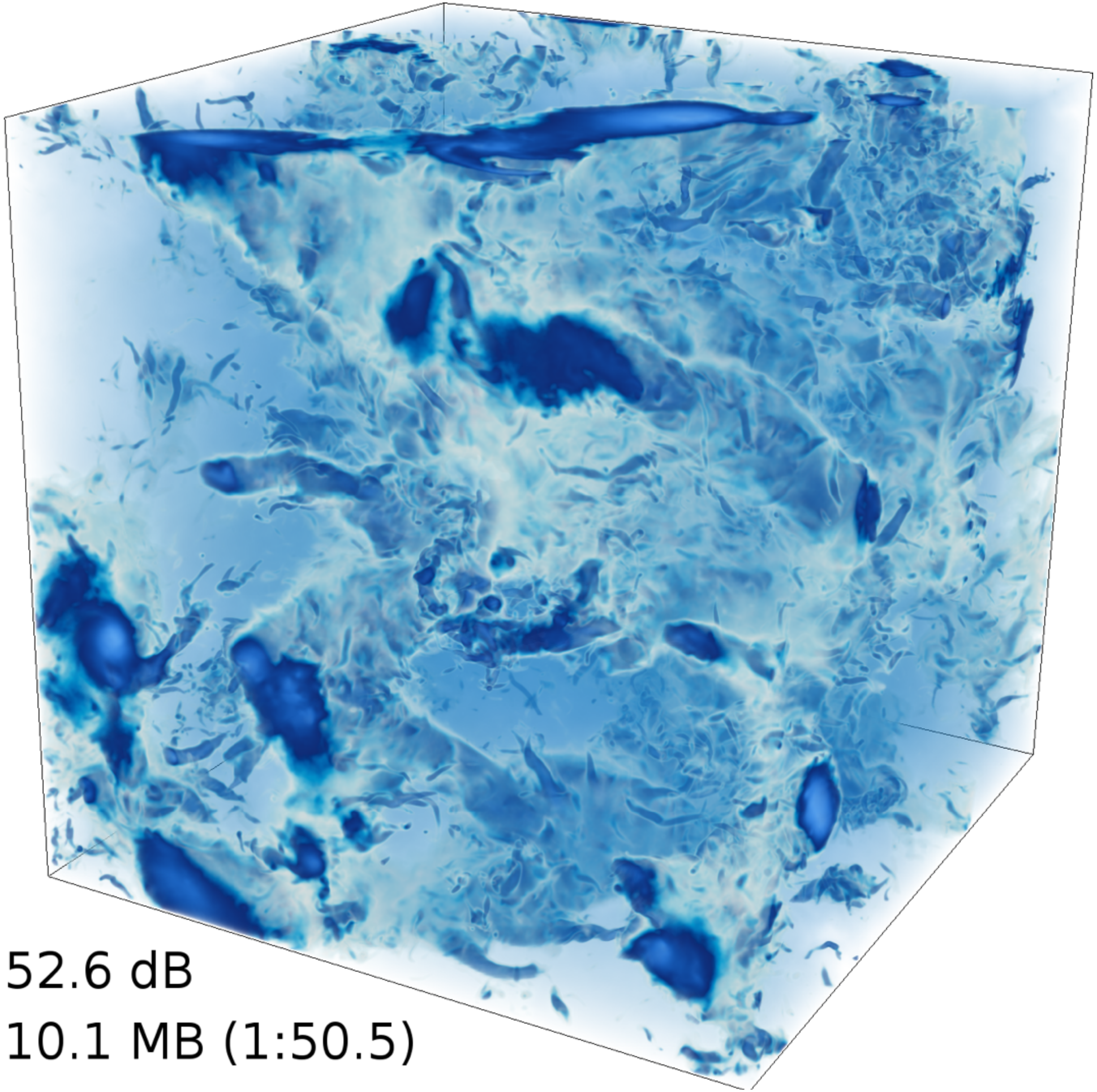
7 bits



8 bits

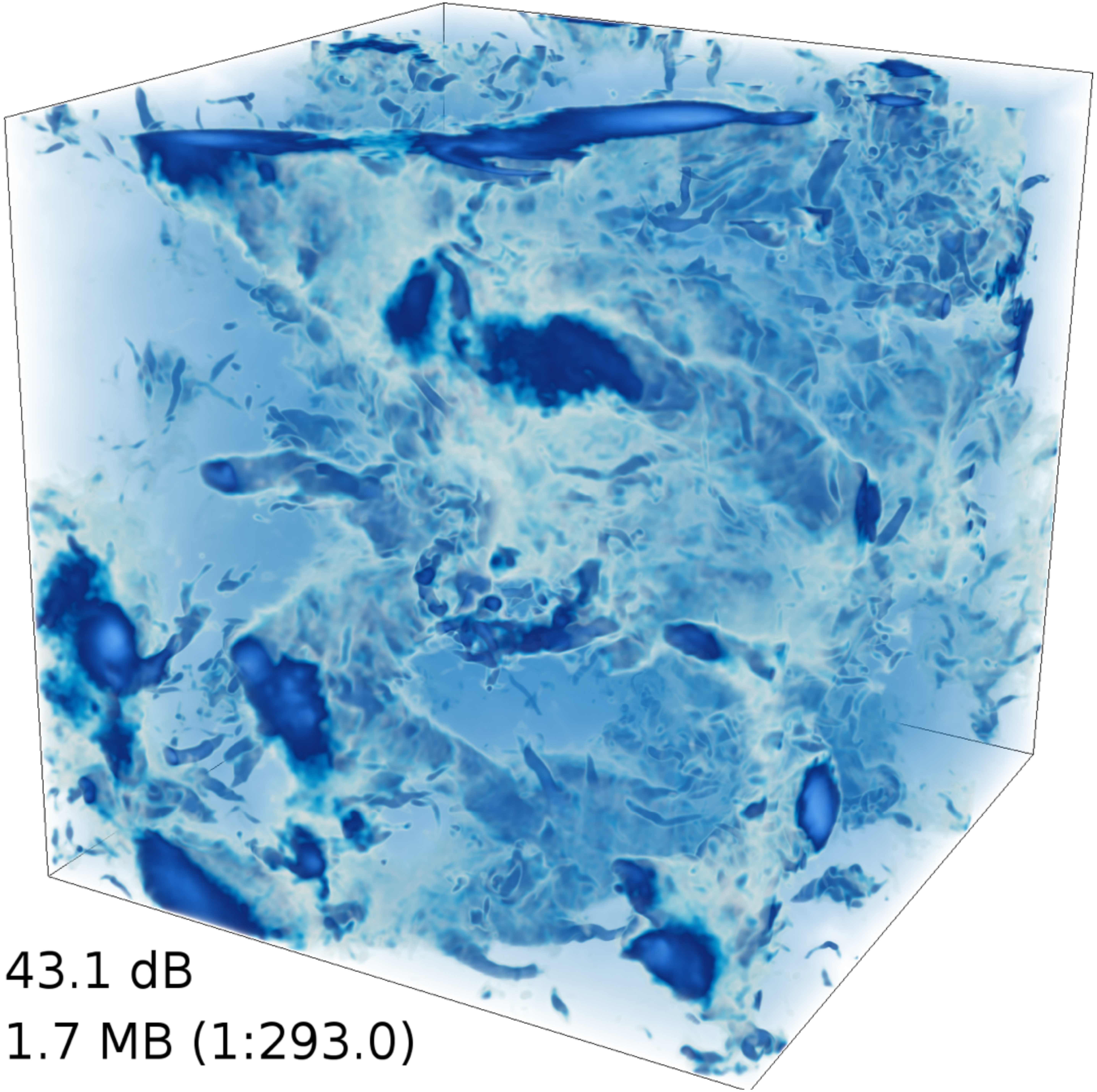



Original
512.0 MB



52.6 dB

10.1 MB (1:50.5)



43.1 dB

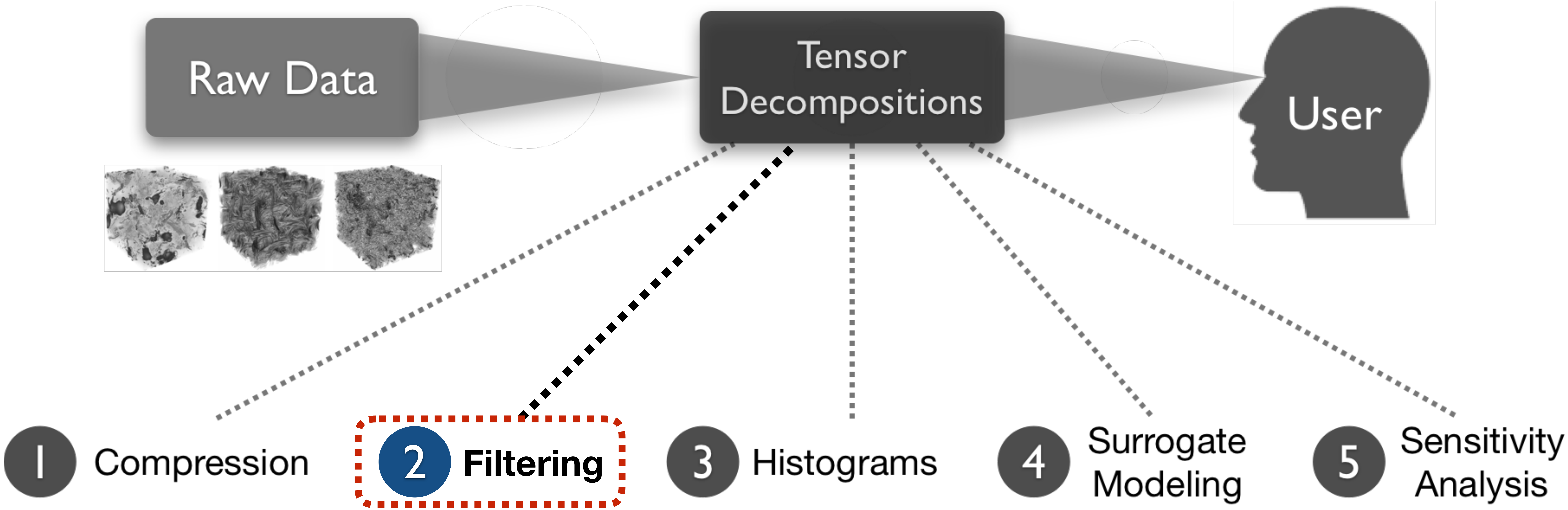
1.7 MB (1:293.0)


Original
512.0 MB
52.6 dB
10.1 MB (1:50.5)
43.1 dB
1.7 MB (1:293.0)


Original
512.0 MB
52.6 dB
10.1 MB (1:50.5)
43.1 dB
1.7 MB (1:293.0)


Original
512.0 MB
52.6 dB
10.1 MB (1:50.5)
43.1 dB
1.7 MB (1:293.0)

# Filtering

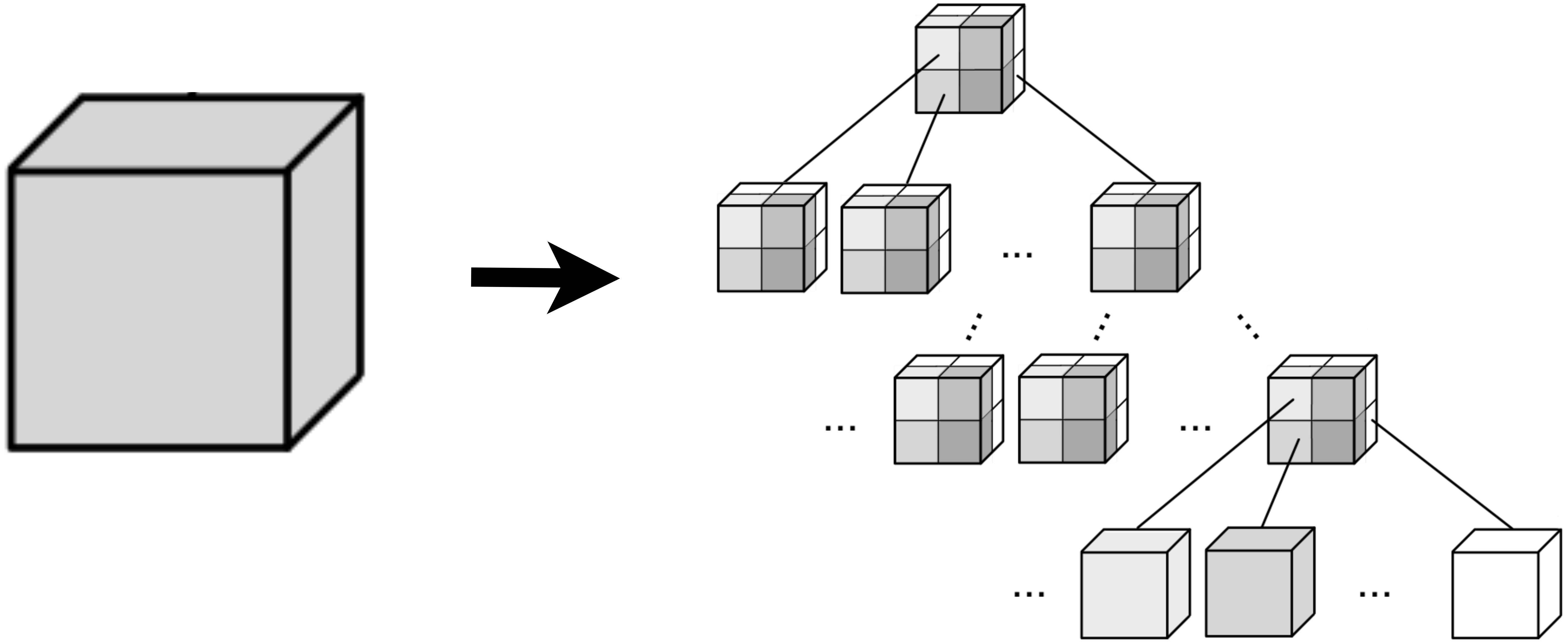



*"Multiresolution Volume Filtering in the Tensor Compressed Domain"*

VISUALIZATIONAND MULTIMEDIALAB

# Multiresolution Volumes

- **Octree:** recursive partition

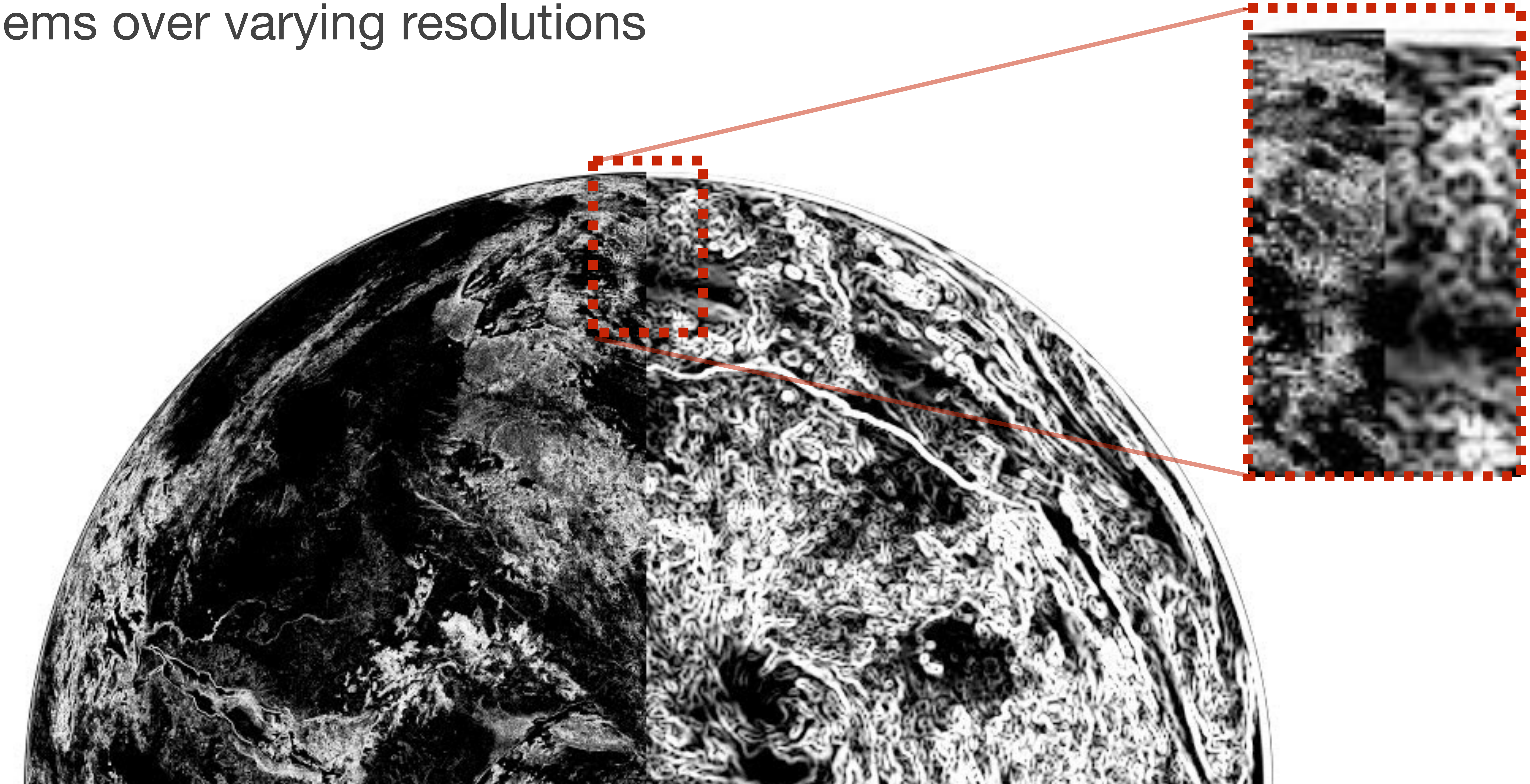


- Closer to the camera ⟶ higher resolution

VISUALIZATIONAND
MULTIMEDIALAB

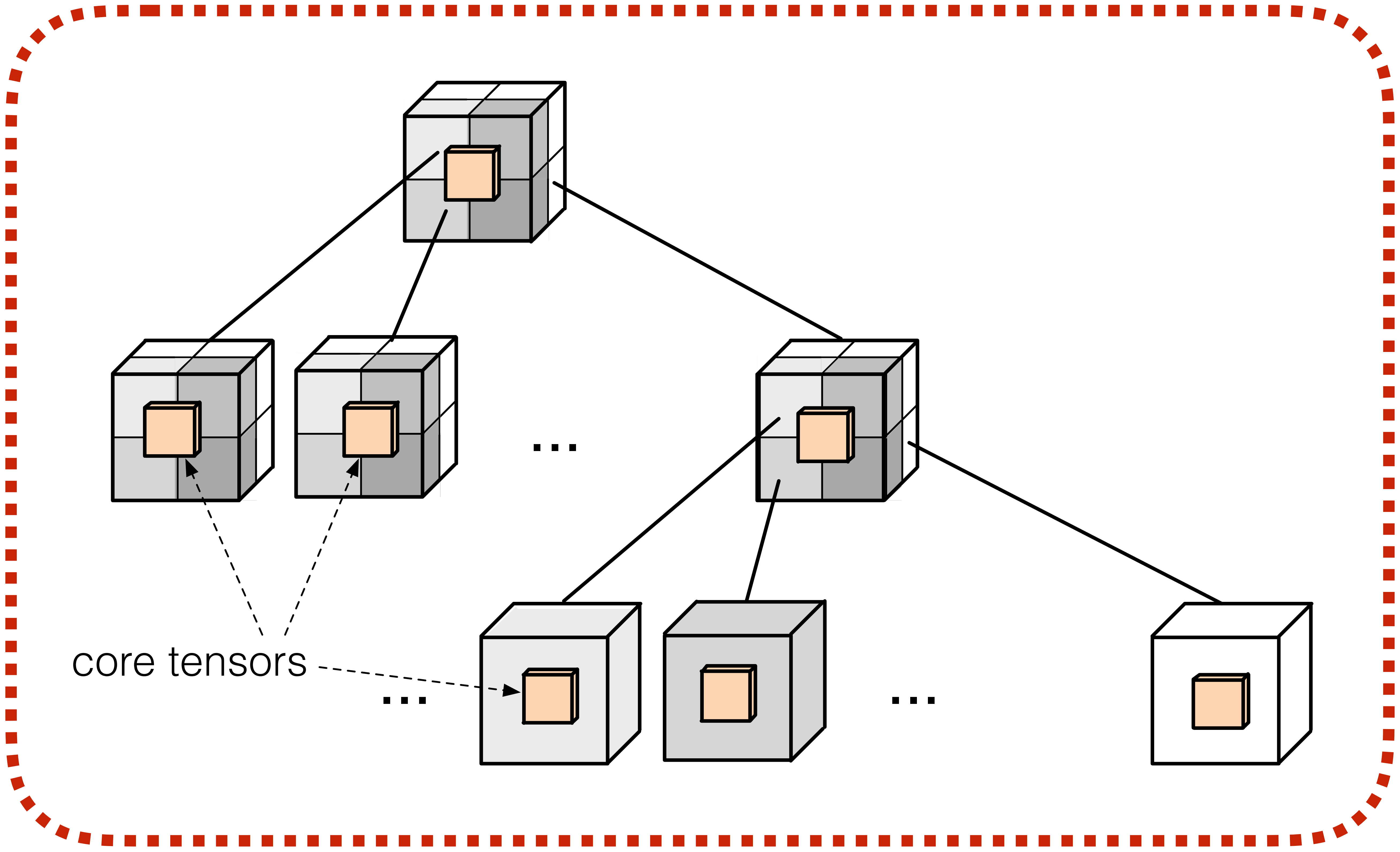
# Challenge

- How to filter an octree?
- Problems over varying resolutions



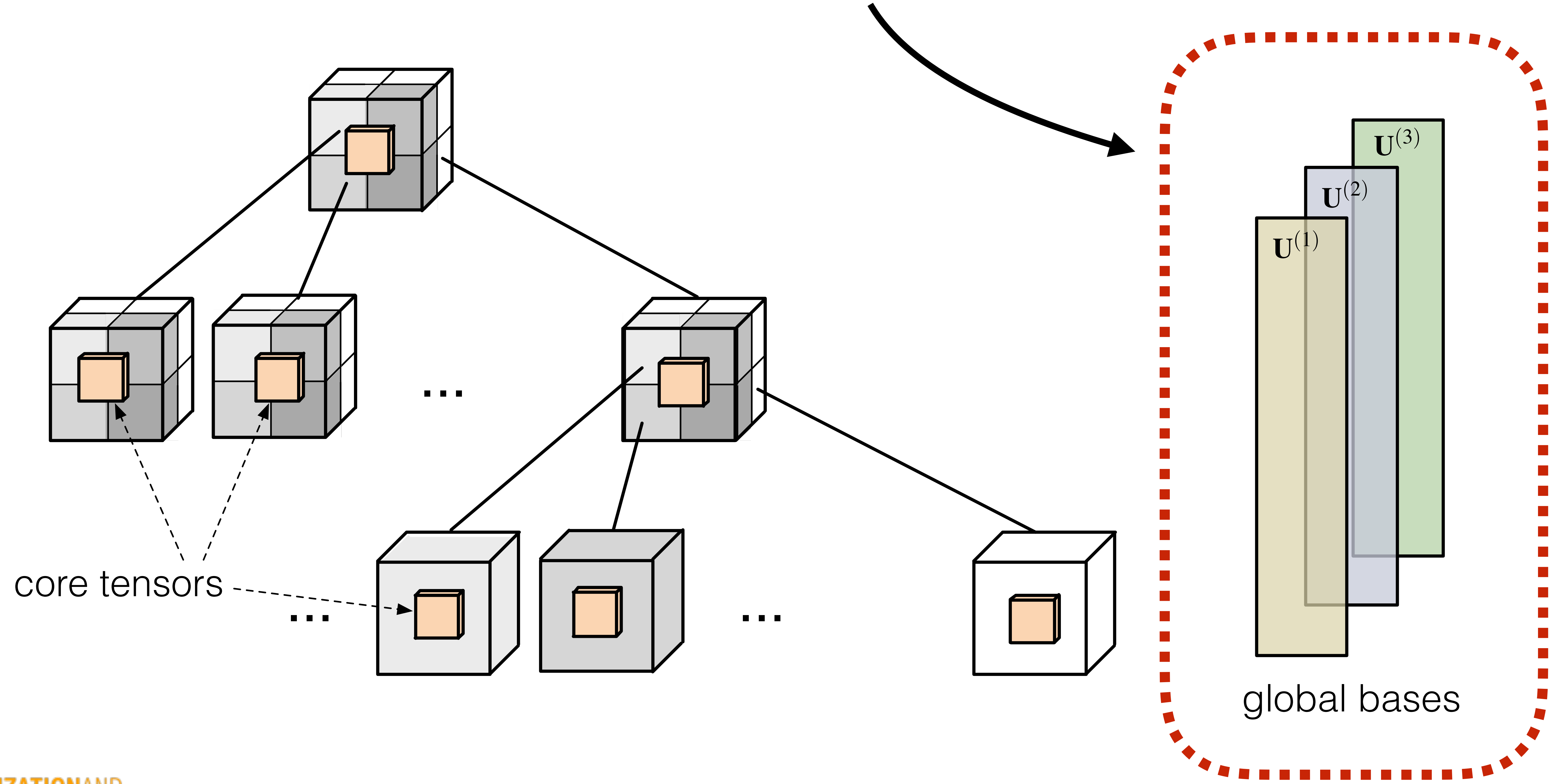
# Proposed Approach

- Novel **multiresolution Tucker** decomposition
  - Tailored to fast filtering operations



# Proposed Approach

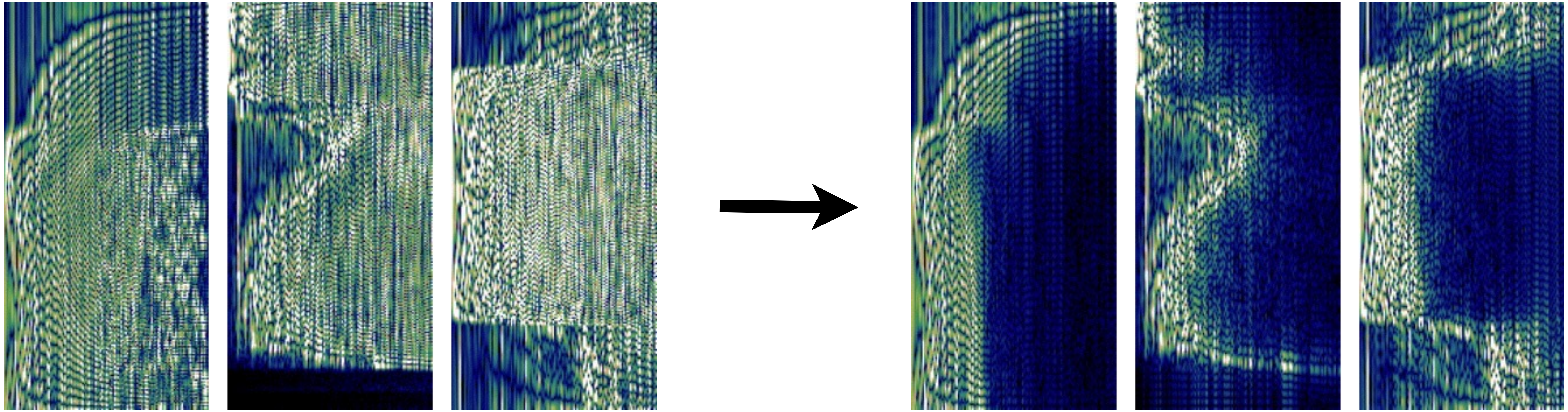
- Novel **multiresolution Tucker** decomposition
  - Tailored to fast filtering operations
- Keeping **full resolution structure**




VISUALIZATIONAND MULTIMEDIALAB

# Proposed Approach

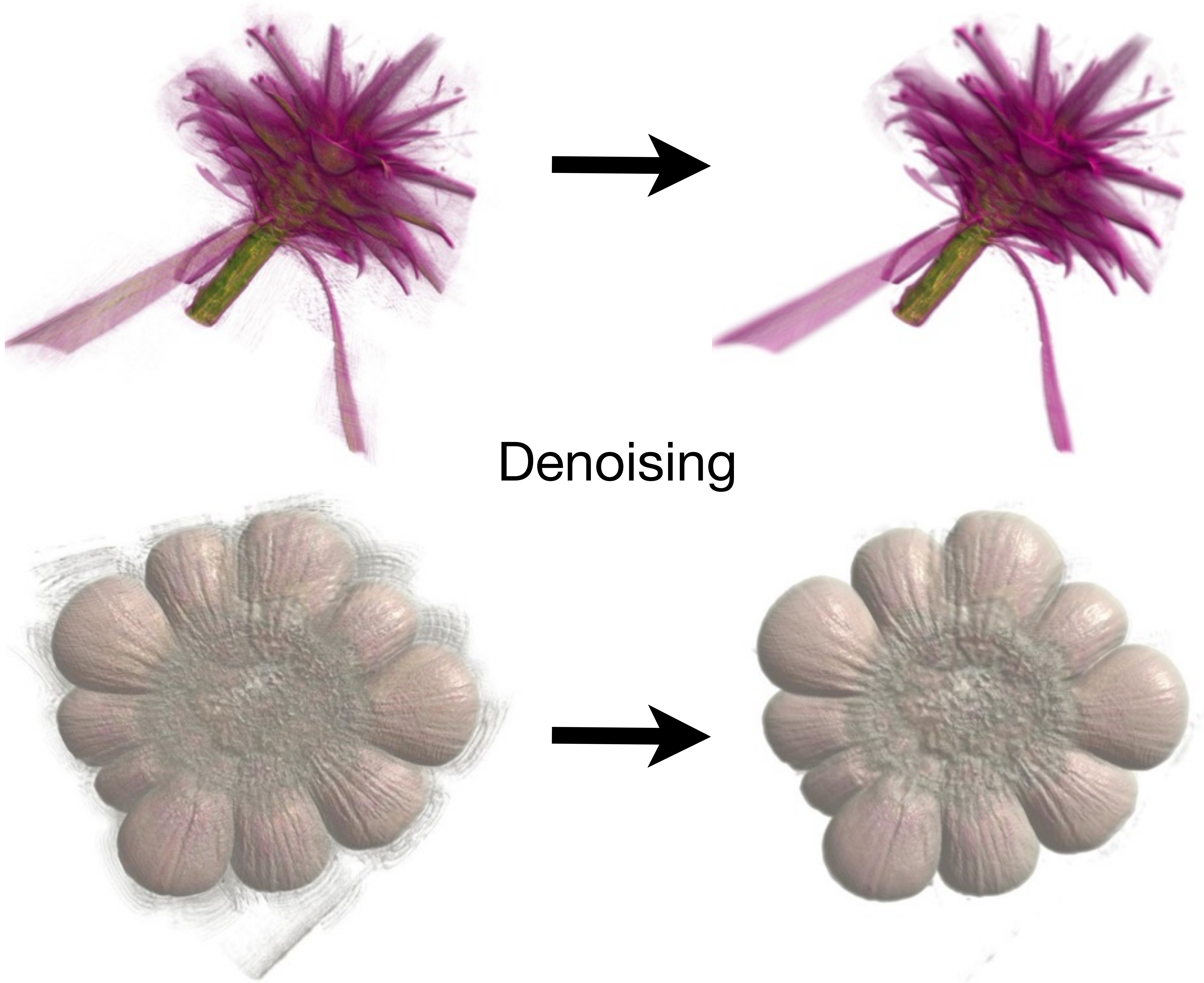
- To filter: **convolve all bases** column-wise



- Then, decompress: result is filtered!

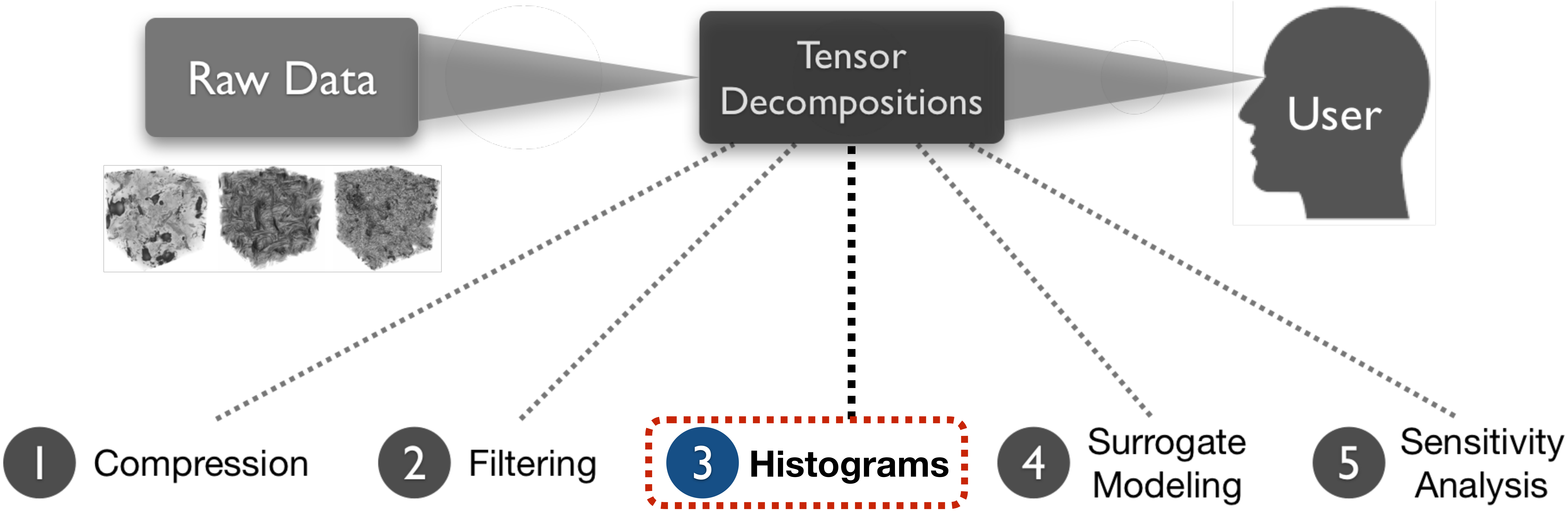
VISUALIZATIONAND MULTIMEDIALAB

# Results




- Smooth response
- Interactive rates
- Tested up to 8GB volumes
- Flexible:
  - Difference of Gaussians
  - Sobel operator
  - Guided filter, etc.

# Histograms



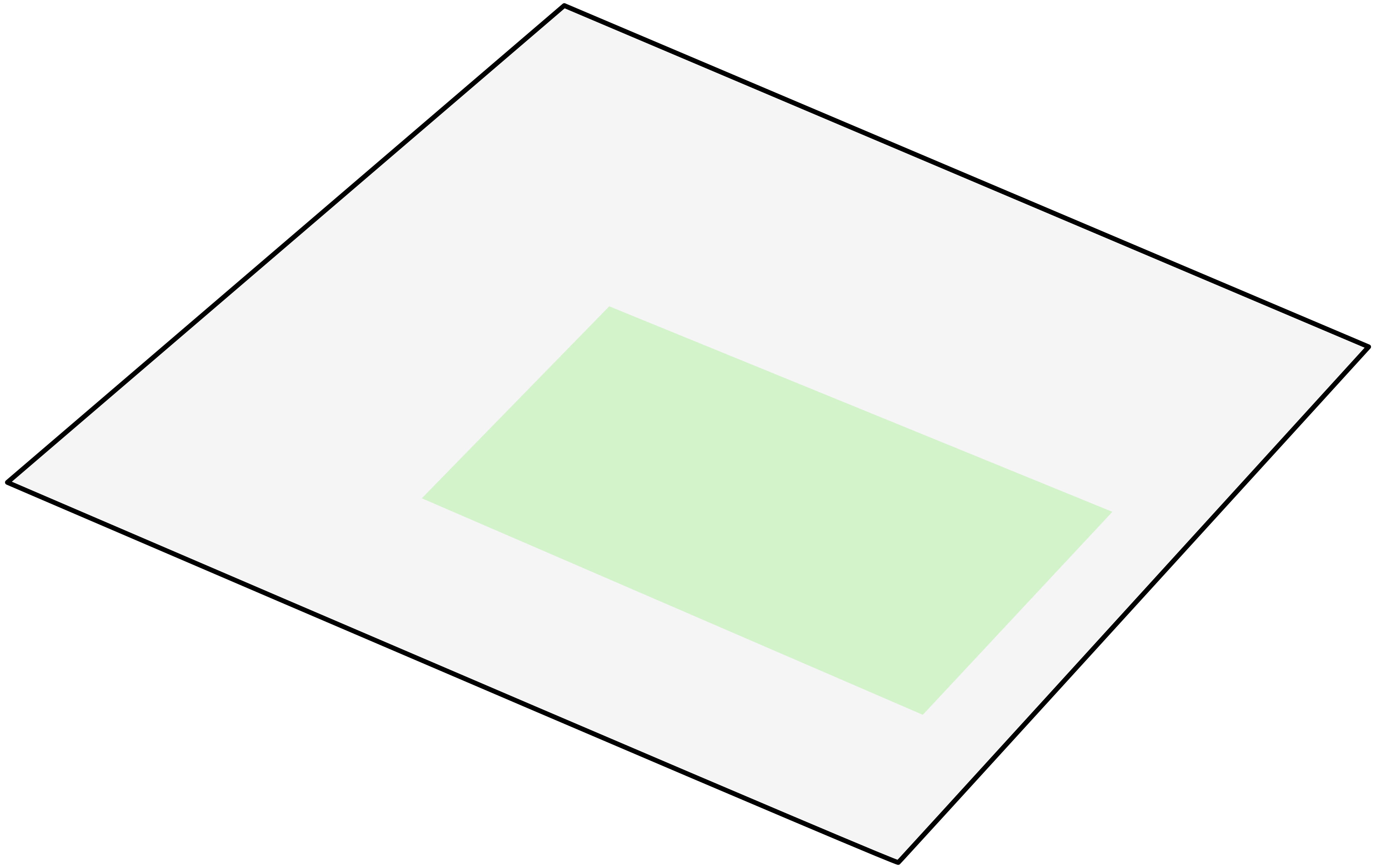

*"Tensor Decompositions for Integral Histogram Compression and Look-up"*

VISUALIZATIONAND MULTIMEDIALAB

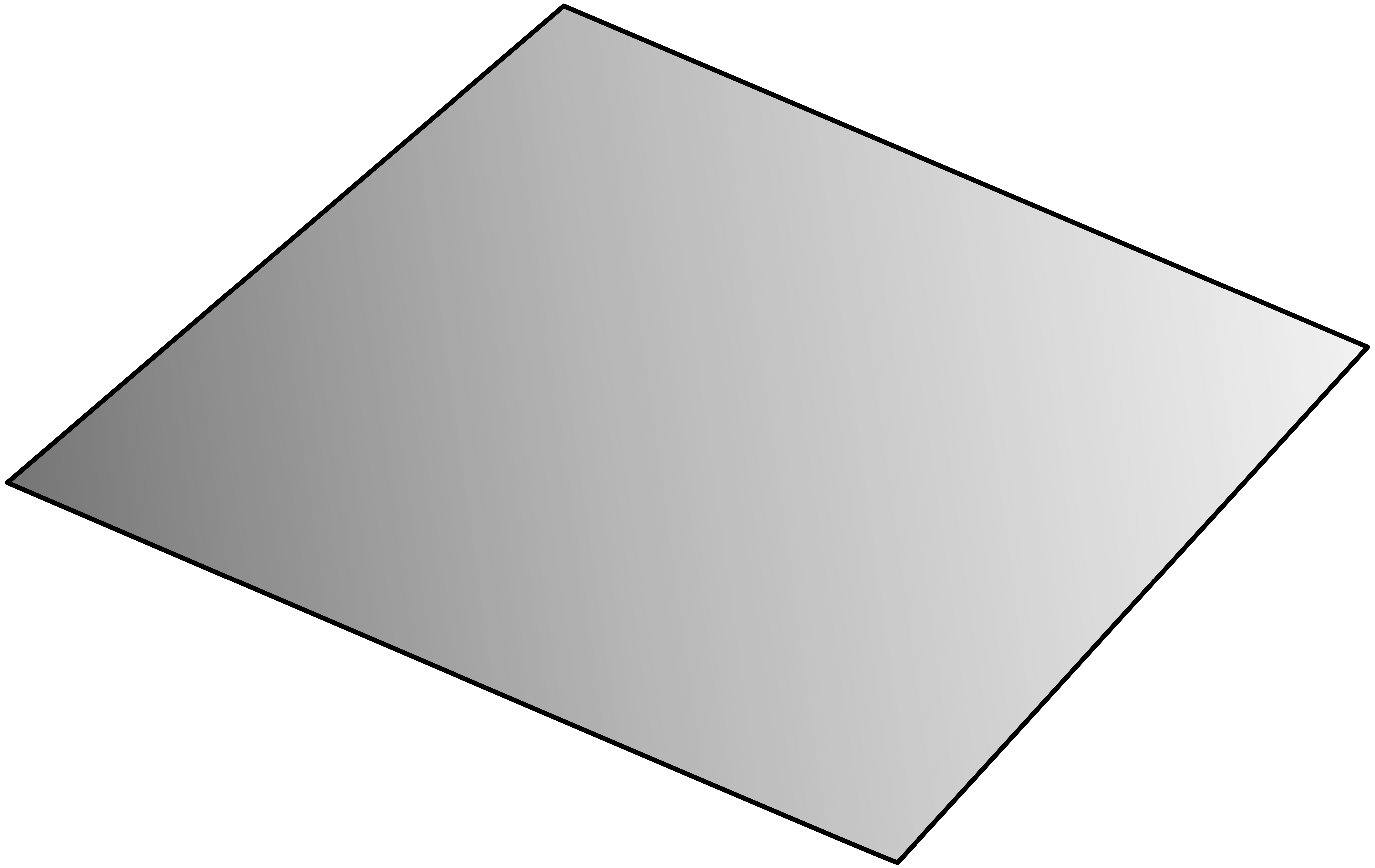
# Challenge

- Computing histograms over large regions
  - Can be slow
- **Popular method:**
  - Pre-integrate each histogram bin: *summed area tables*
  - Stack them up ⟶ *integral histogram* **[Porikli '05]**
  - Fast, **but huge**

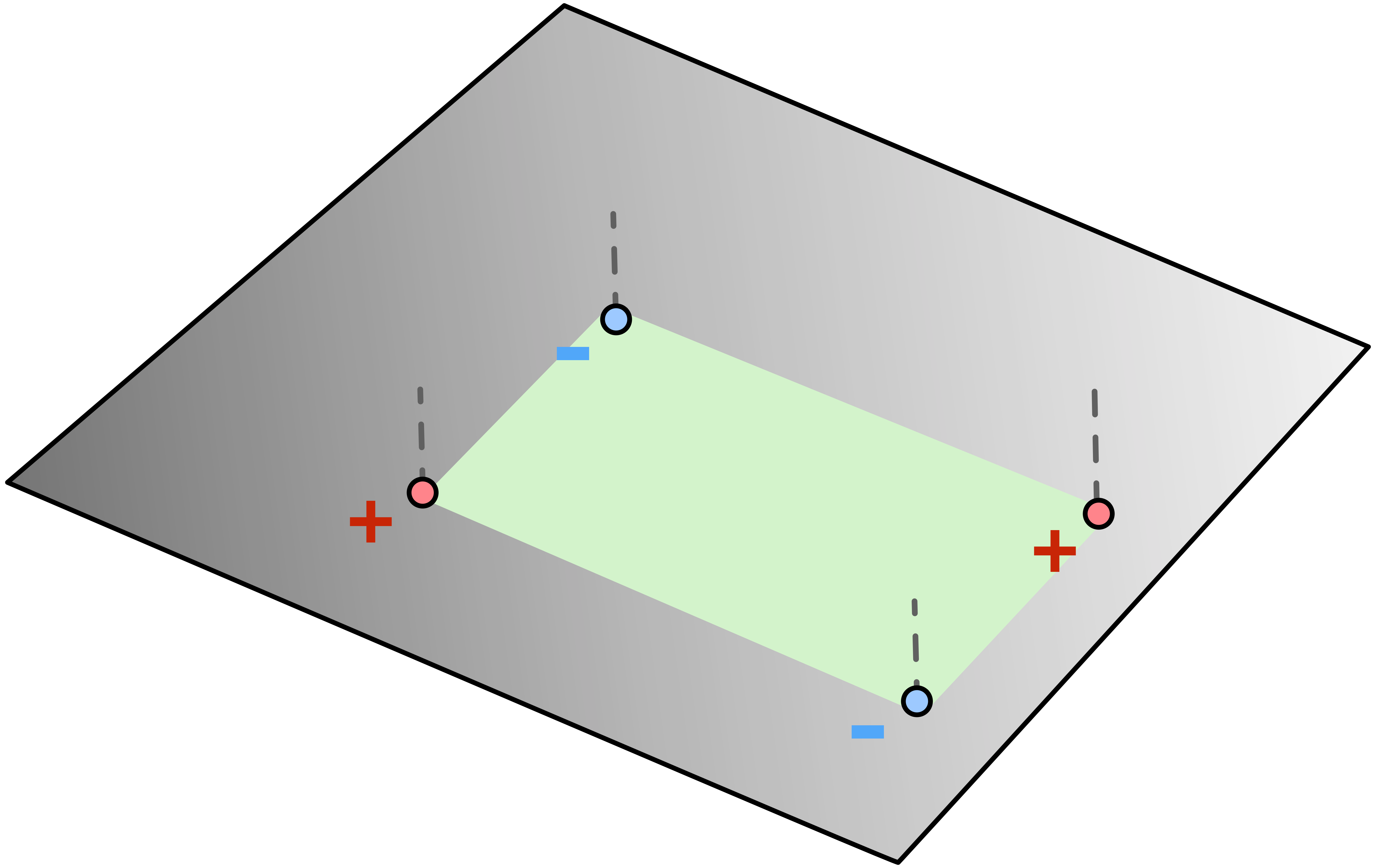
# Summed Area Tables



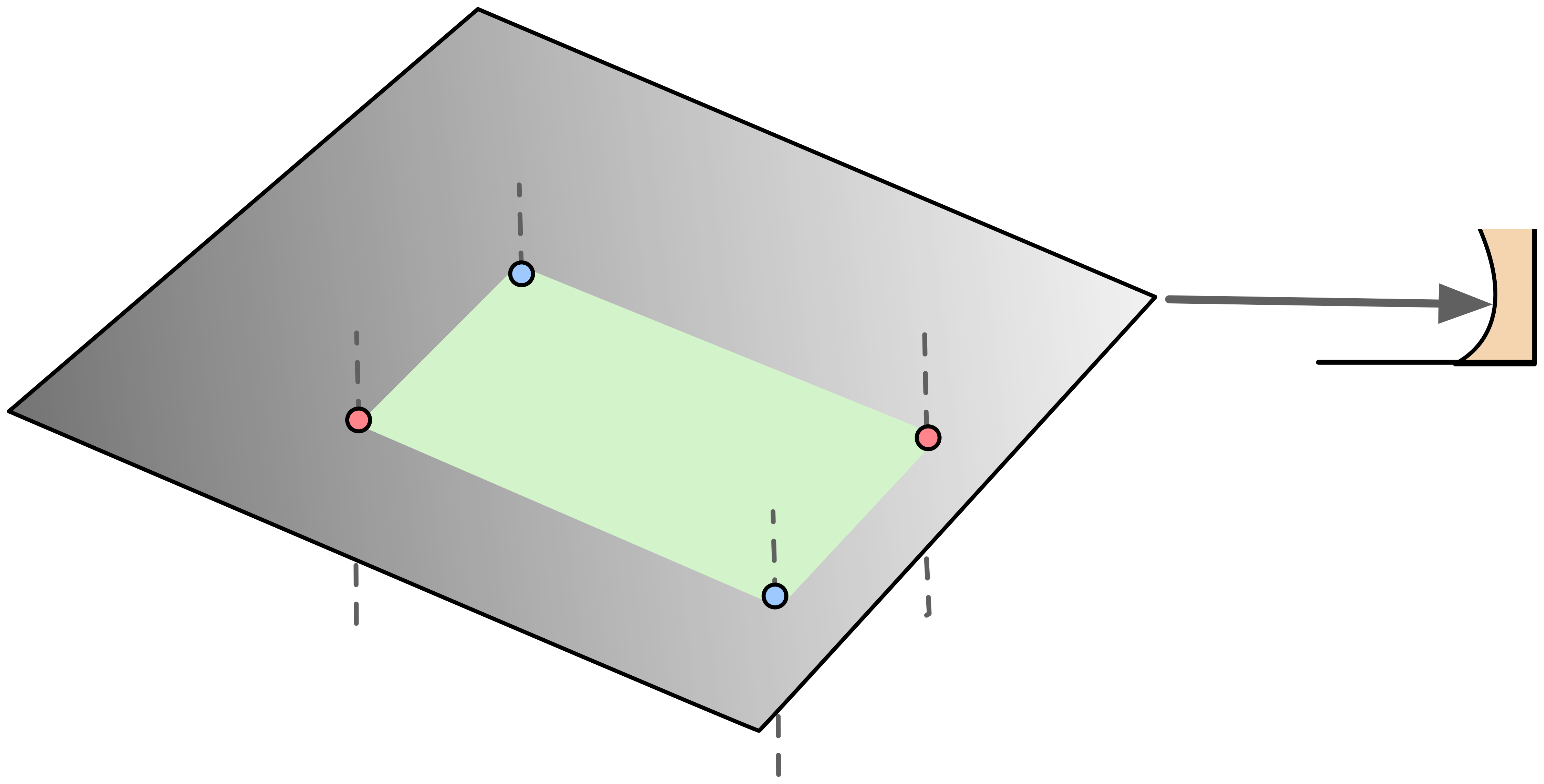
# Summed Area Tables



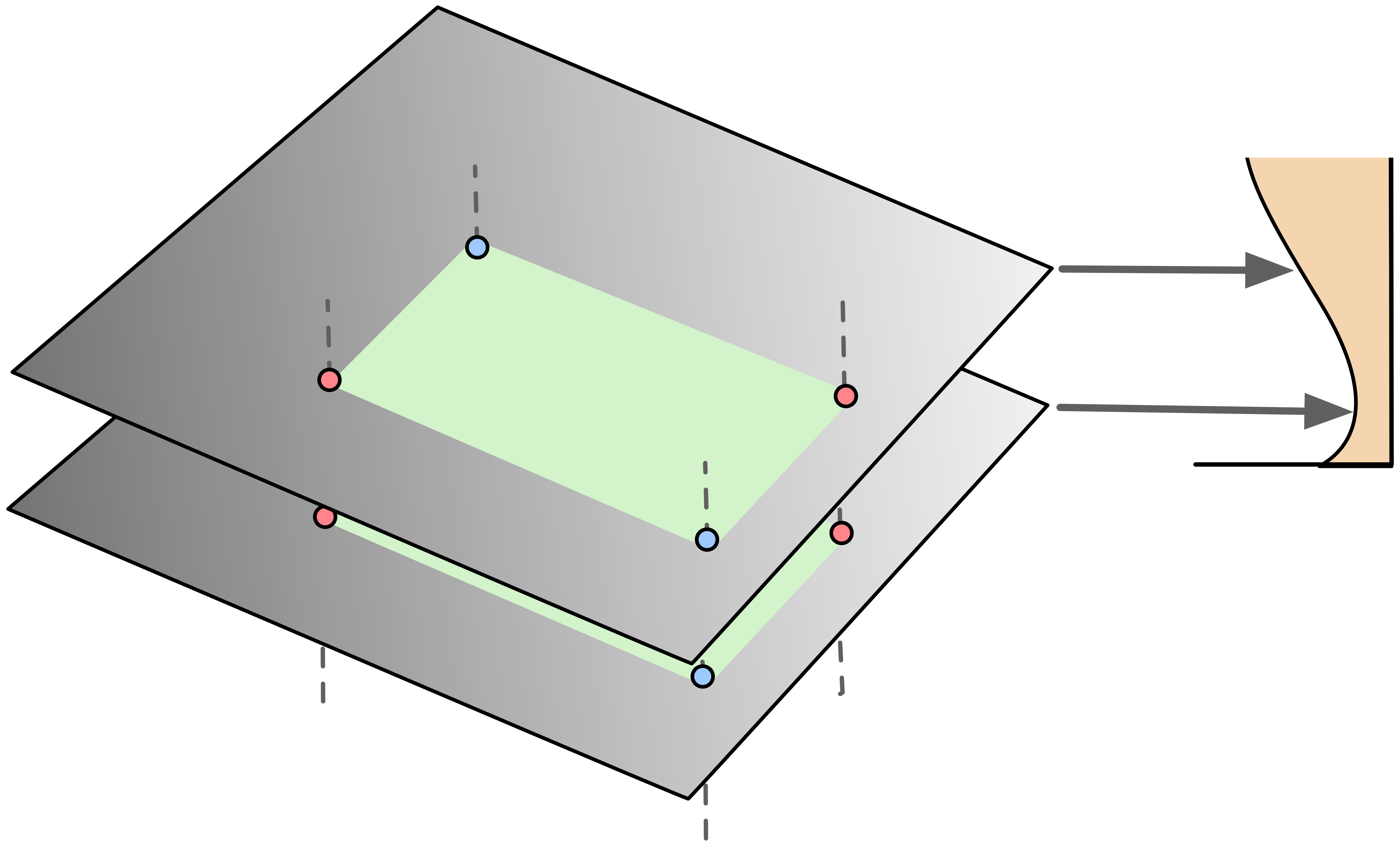
# Summed Area Tables



# Integral Histograms

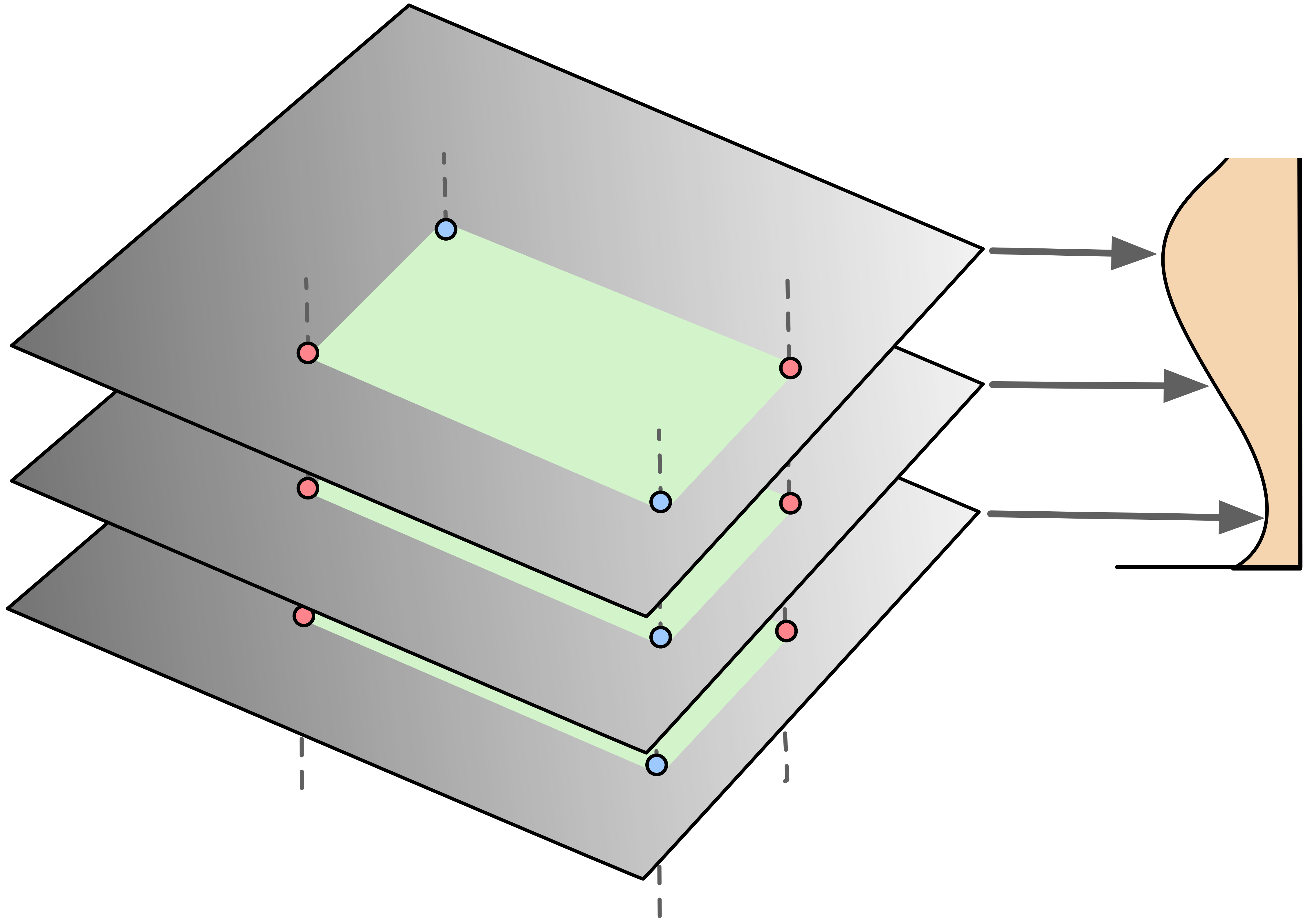


# Integral Histograms

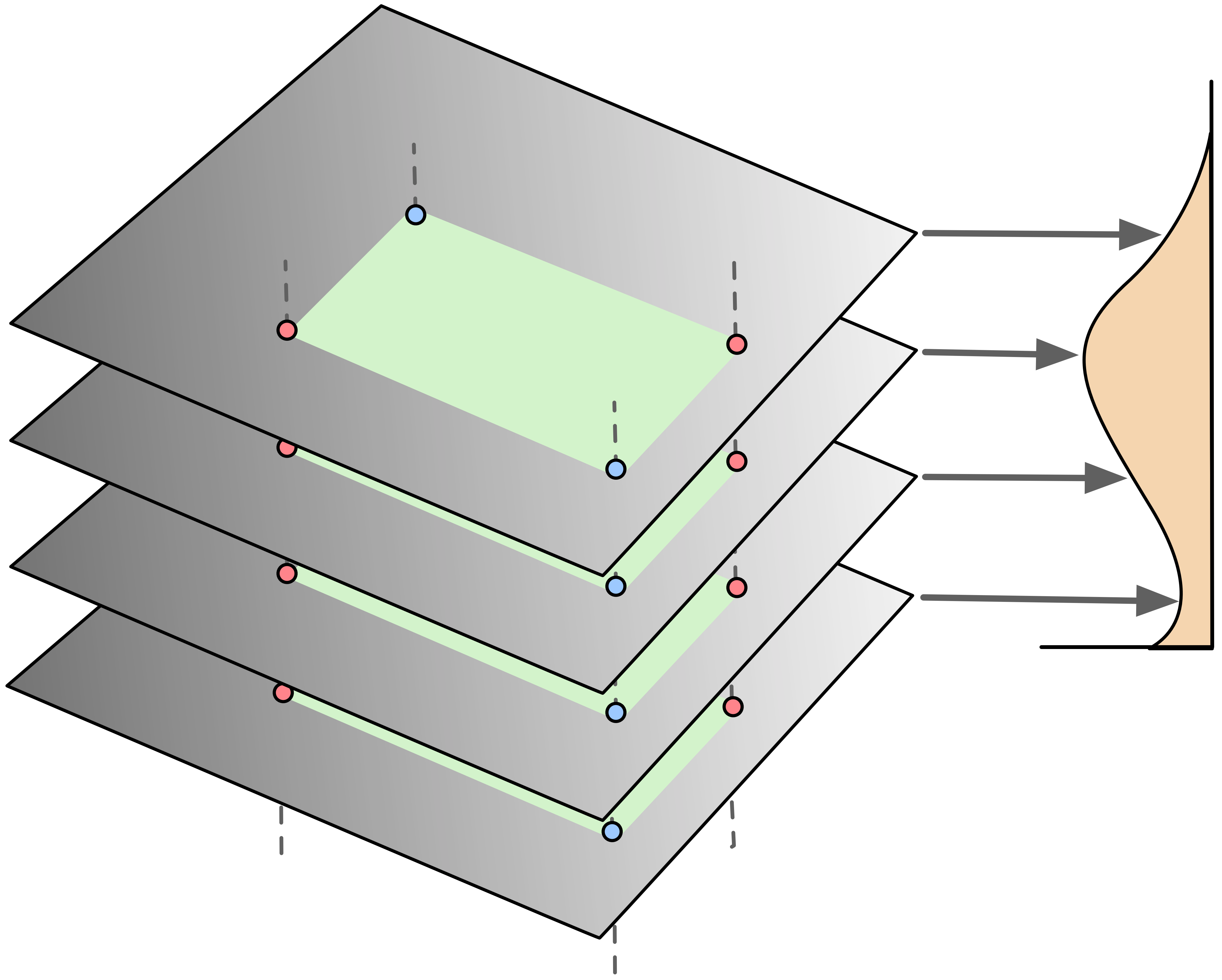


VISUALIZATIONAND
MULTIMEDIALAB

# Integral Histograms

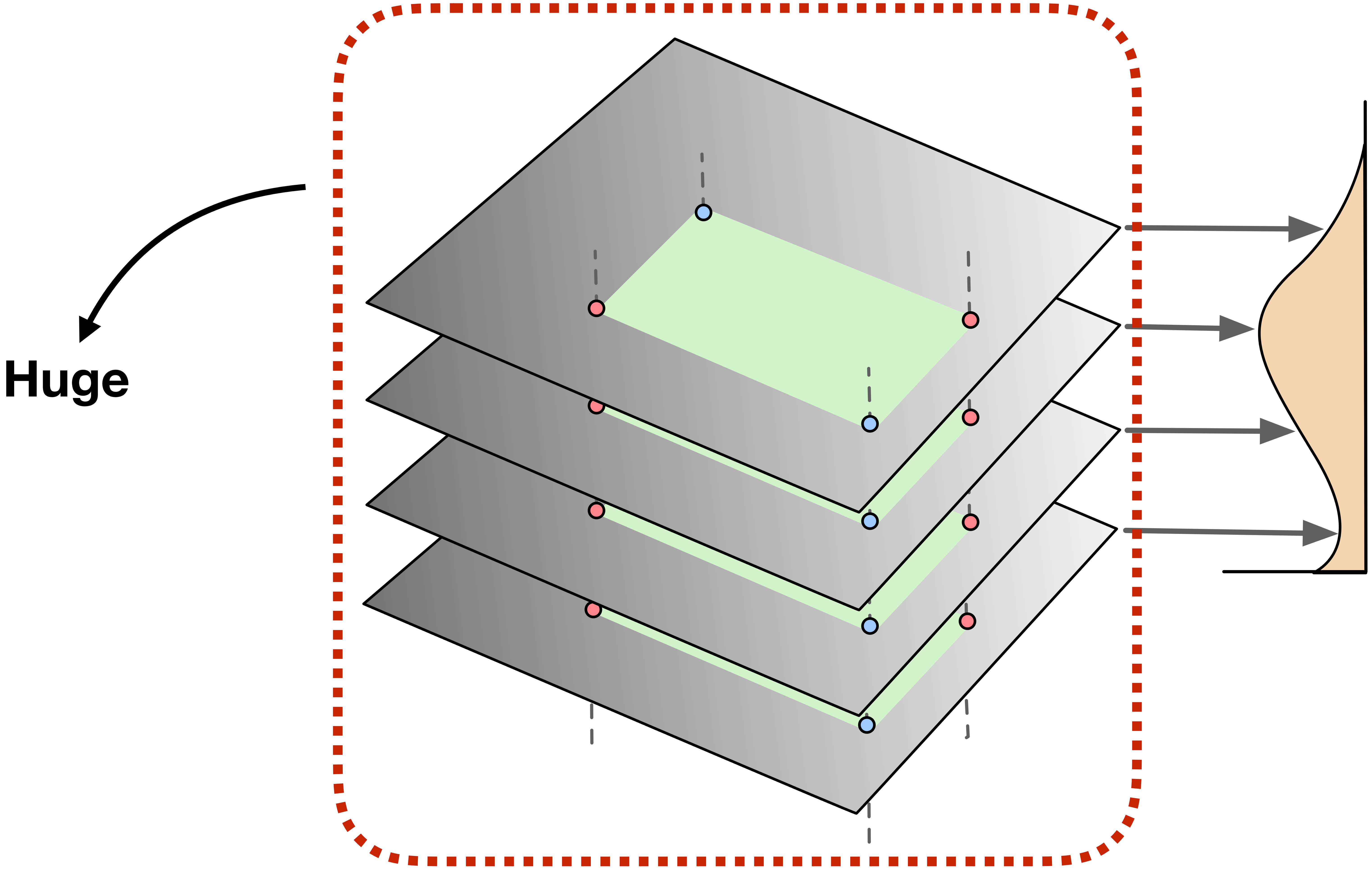


# Integral Histograms



VISUALIZATIONAND
MULTIMEDIALAB

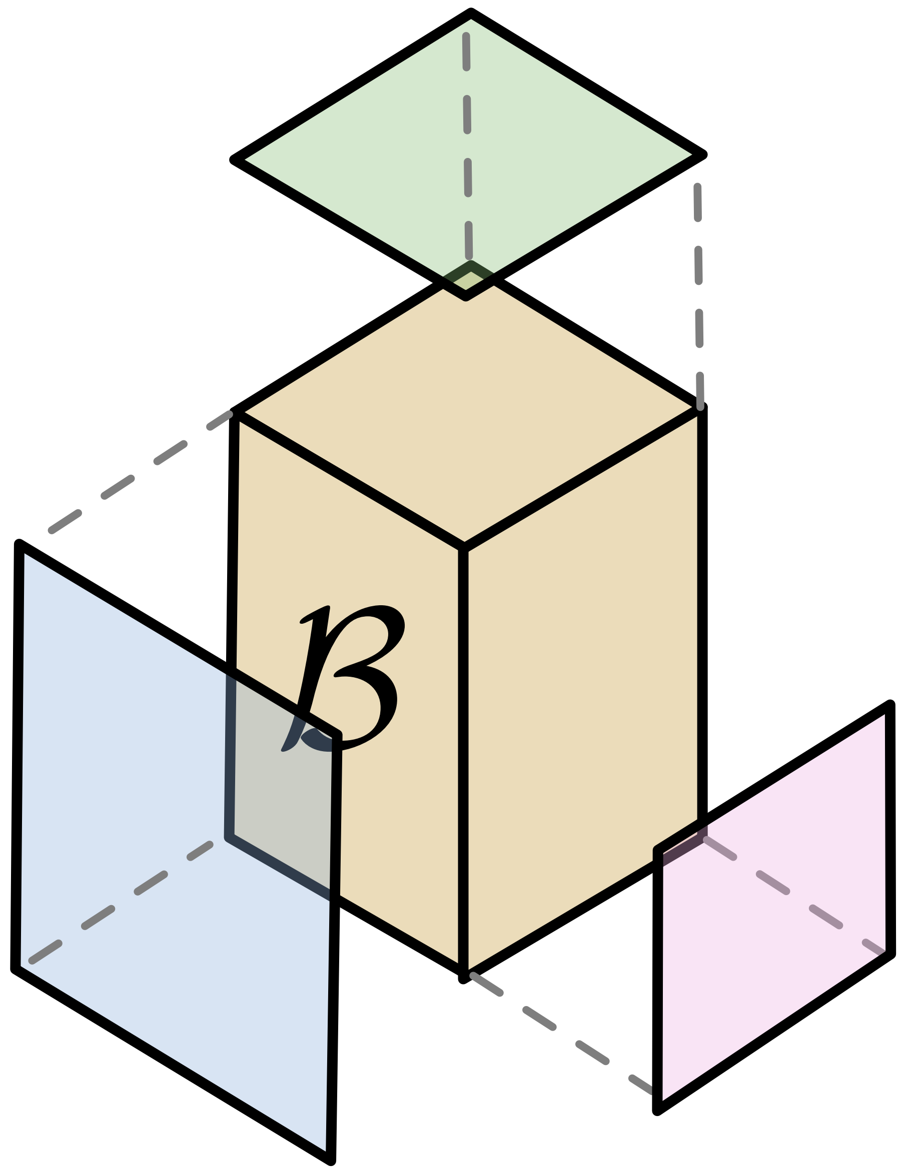
# Integral Histograms




VISUALIZATIONAND
MULTIMEDIALAB

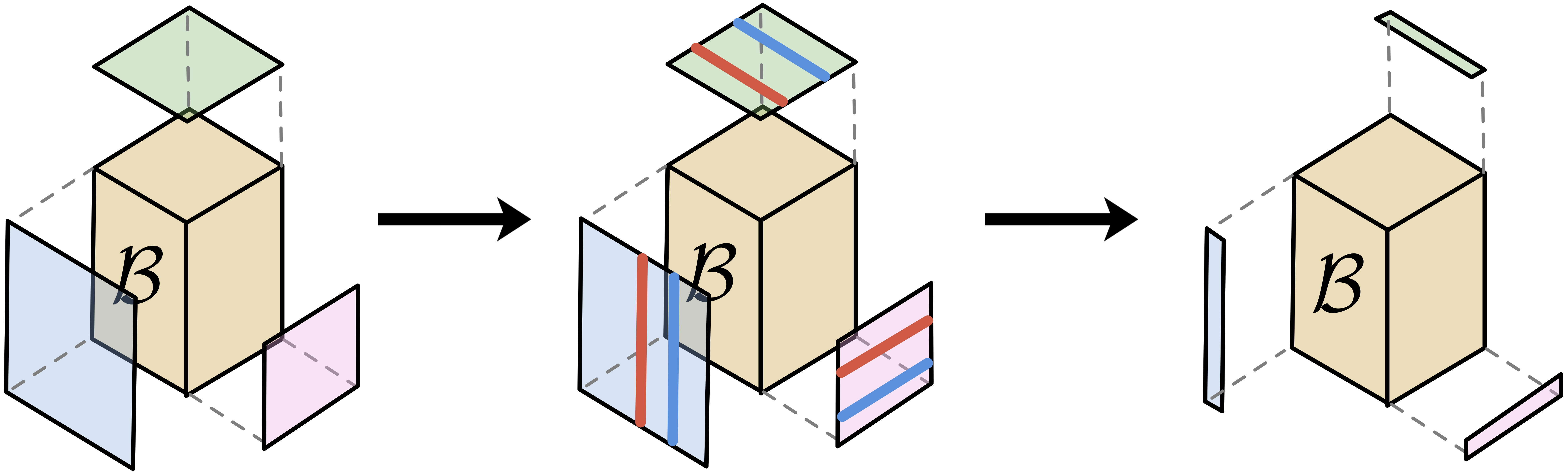
# Proposed Approach

- Compress the integral histogram

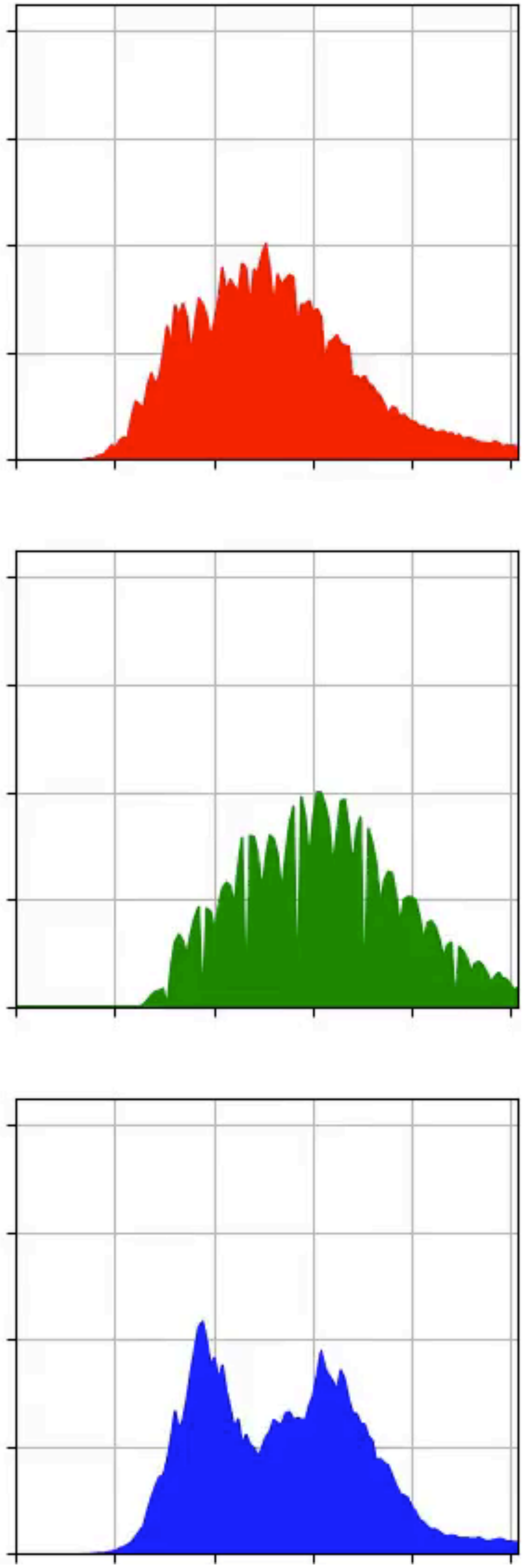


# Proposed Approach

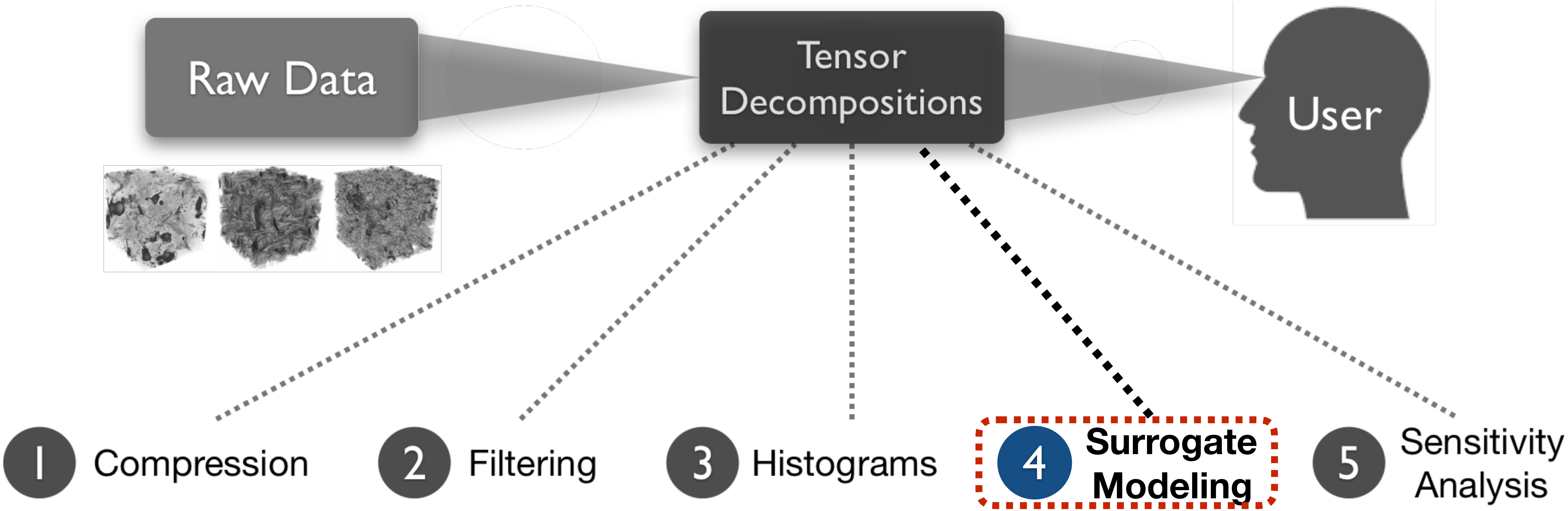
- Compress the integral histogram
- Combine region borders
- Reconstruct




VISUALIZATION AND MULTIMEDIA LAB



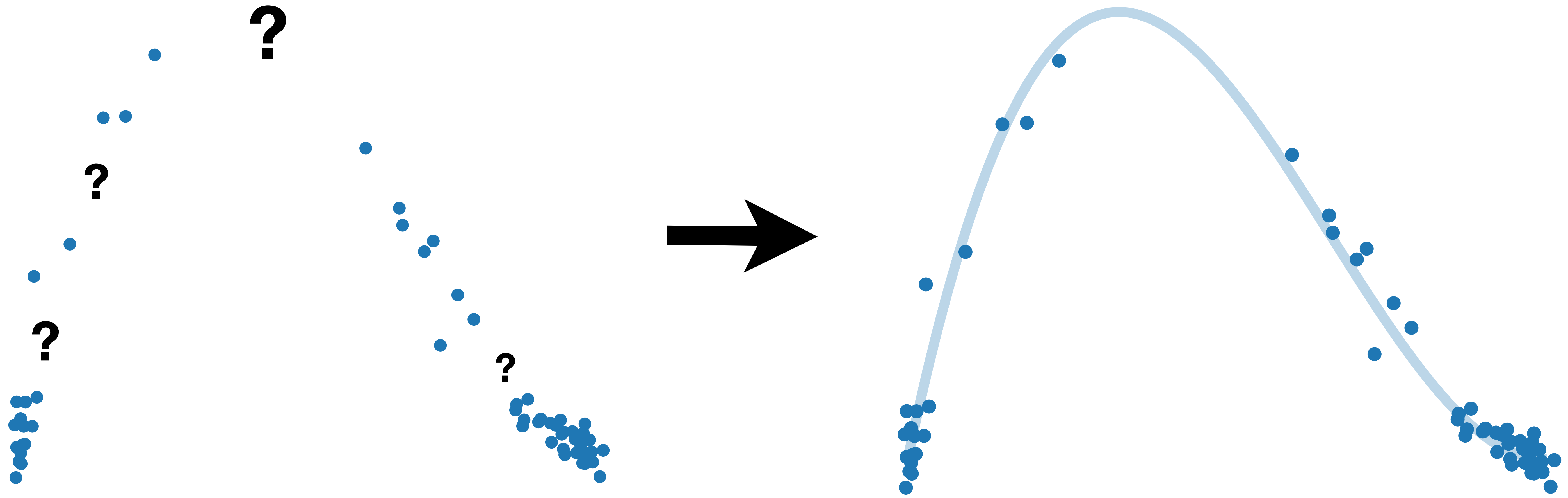
# Surrogate Modeling




*“A Surrogate Visualization Model Using the Tensor Train Format”*

# Surrogate Models

- Often: data set only partly known




Estimate unknown regions

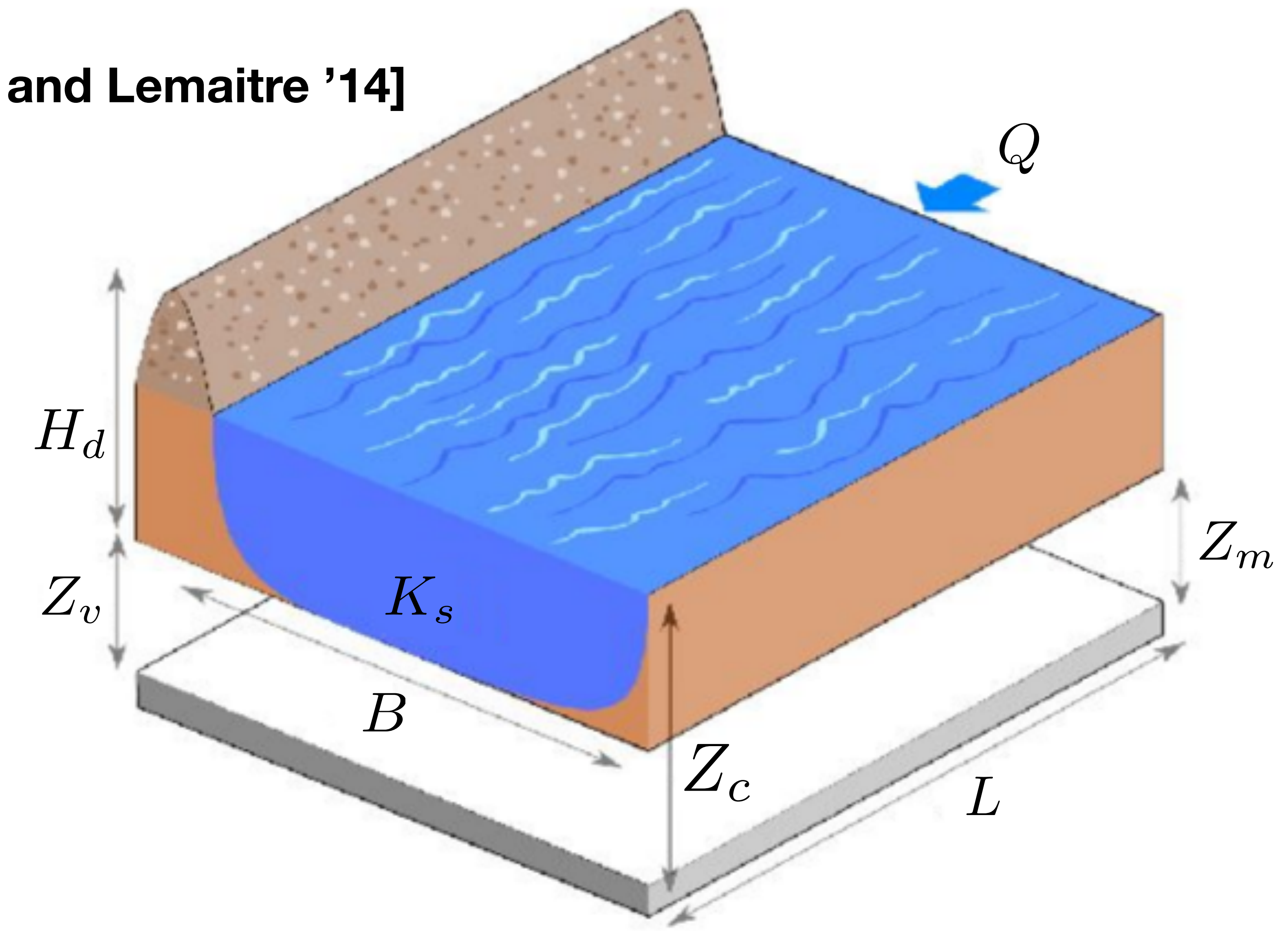
- Great for interactive **analysis** and **visualization**

VISUALIZATIONAND
MULTIMEDIALAB

# Challenges

- In many dimensions:
  - Exponential **complexity**
  - Vast **unknown regions**
  - We want **fast predictions**
    - Maximal/minimal values? Plateaus?

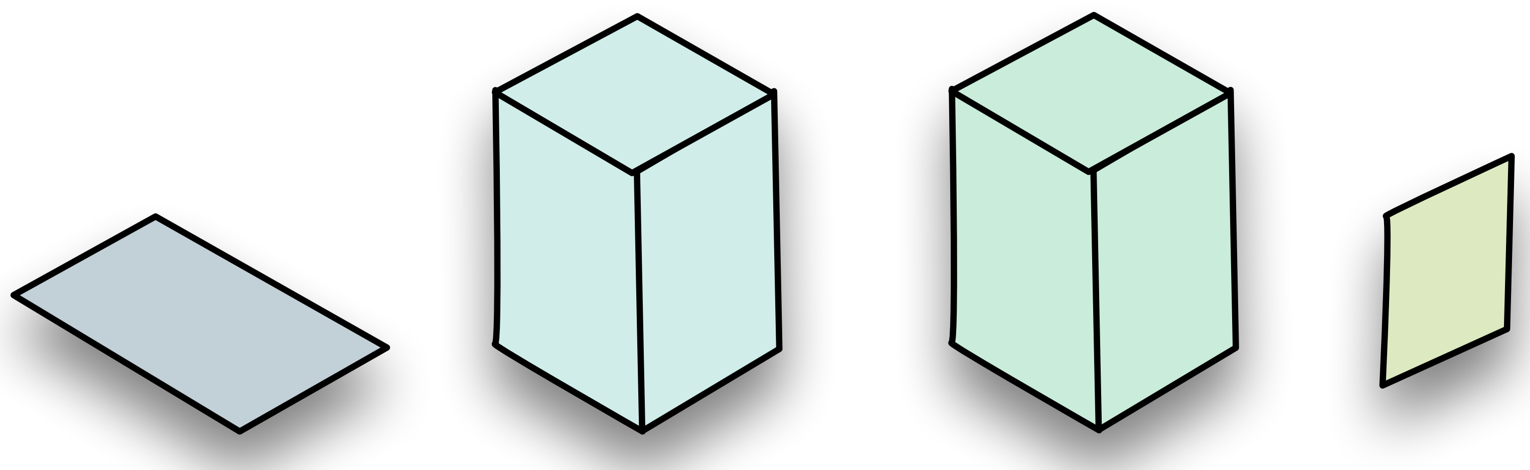
[Iooss and Lemaitre '14]




$$\text{dike cost} = f(Q, Z_m, L, Z_c, B, K_s, Z_v, H_d)$$

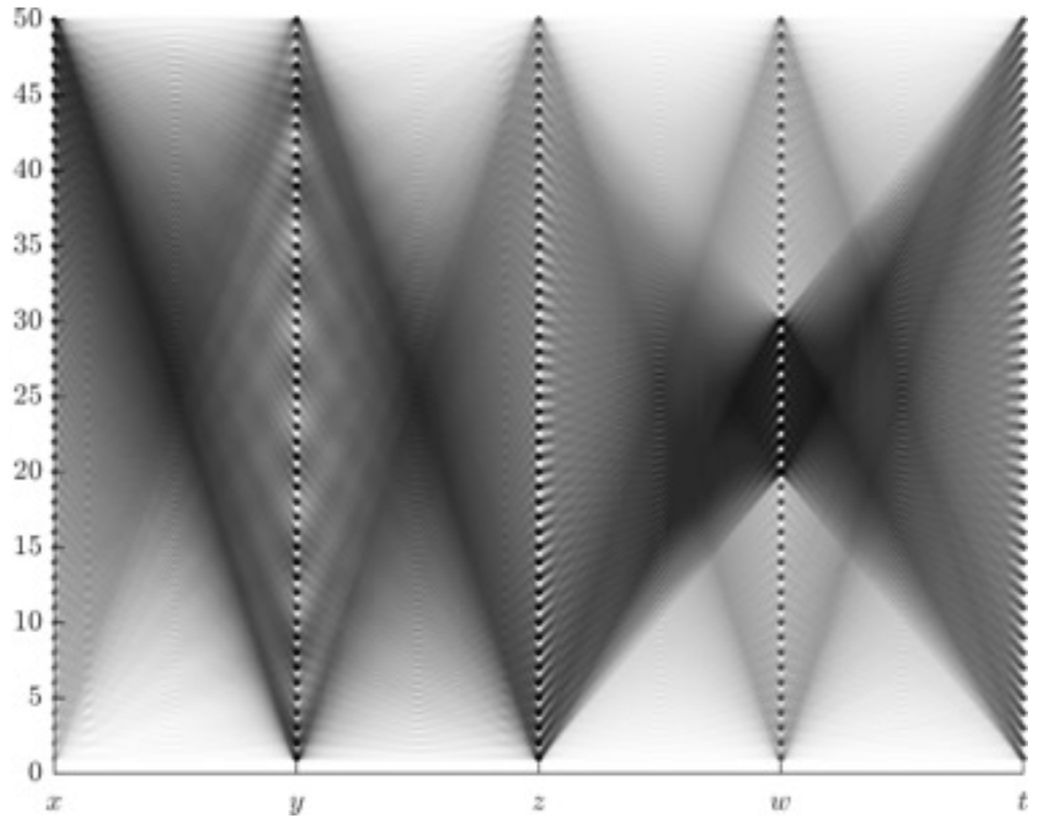
**How to understand** a high-dimensional surrogate?

# Proposed Approach

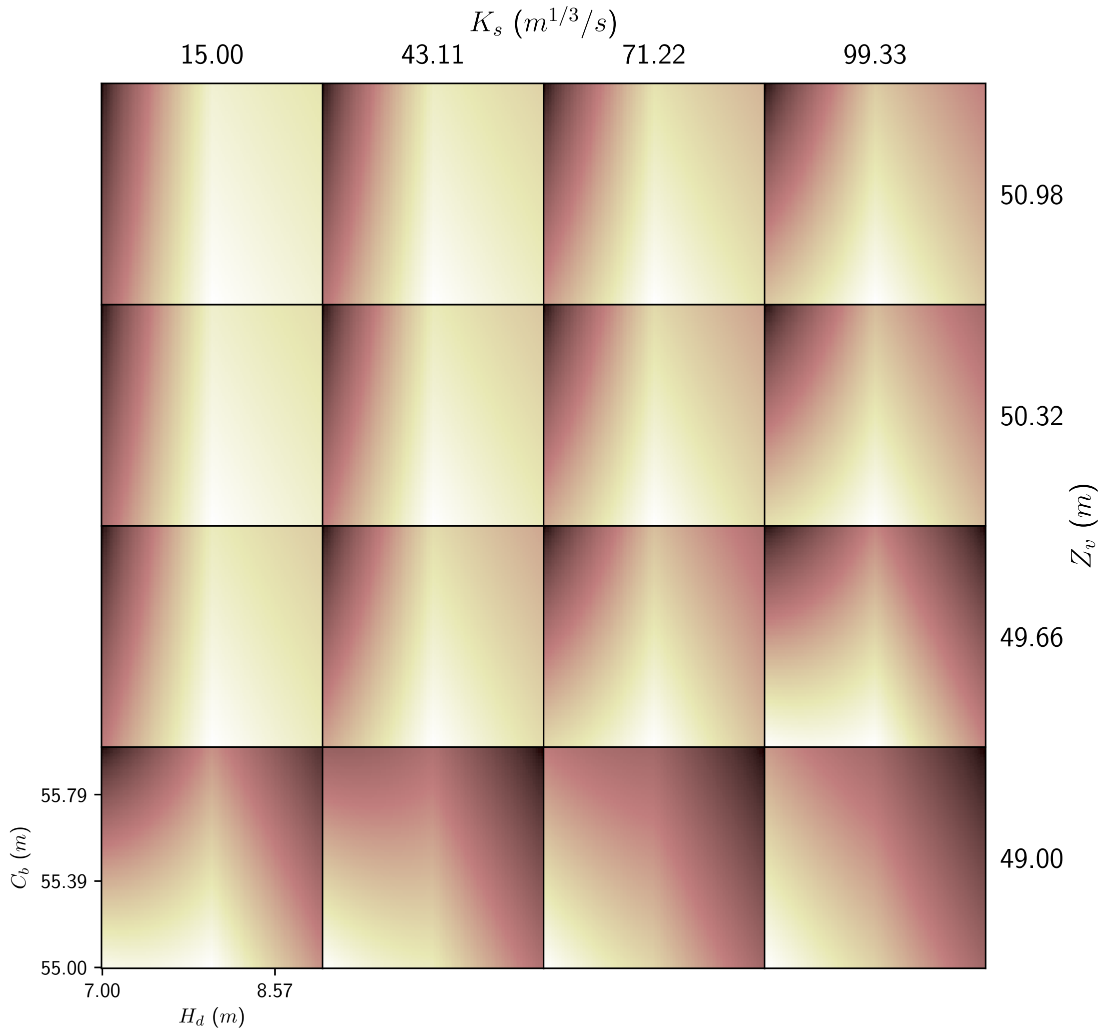


- Create **tensor train surrogates**
  - Samples are fixed ⟶ *tensor completion* **[Kressner et al. '13], [Steinlechner '15]**
  - We can sample at will ⟶ *adaptive sampling* **[Oseledets & Tyrtyshnikov '10], [Savostyanov & Oseledets '11]**
- **Visualize the surrogate's behavior:**
  - Fast reconstruction
  - Statistics and moments in the TT format
  - TT global optimization **[Mikhalev & Oseledets '15]**

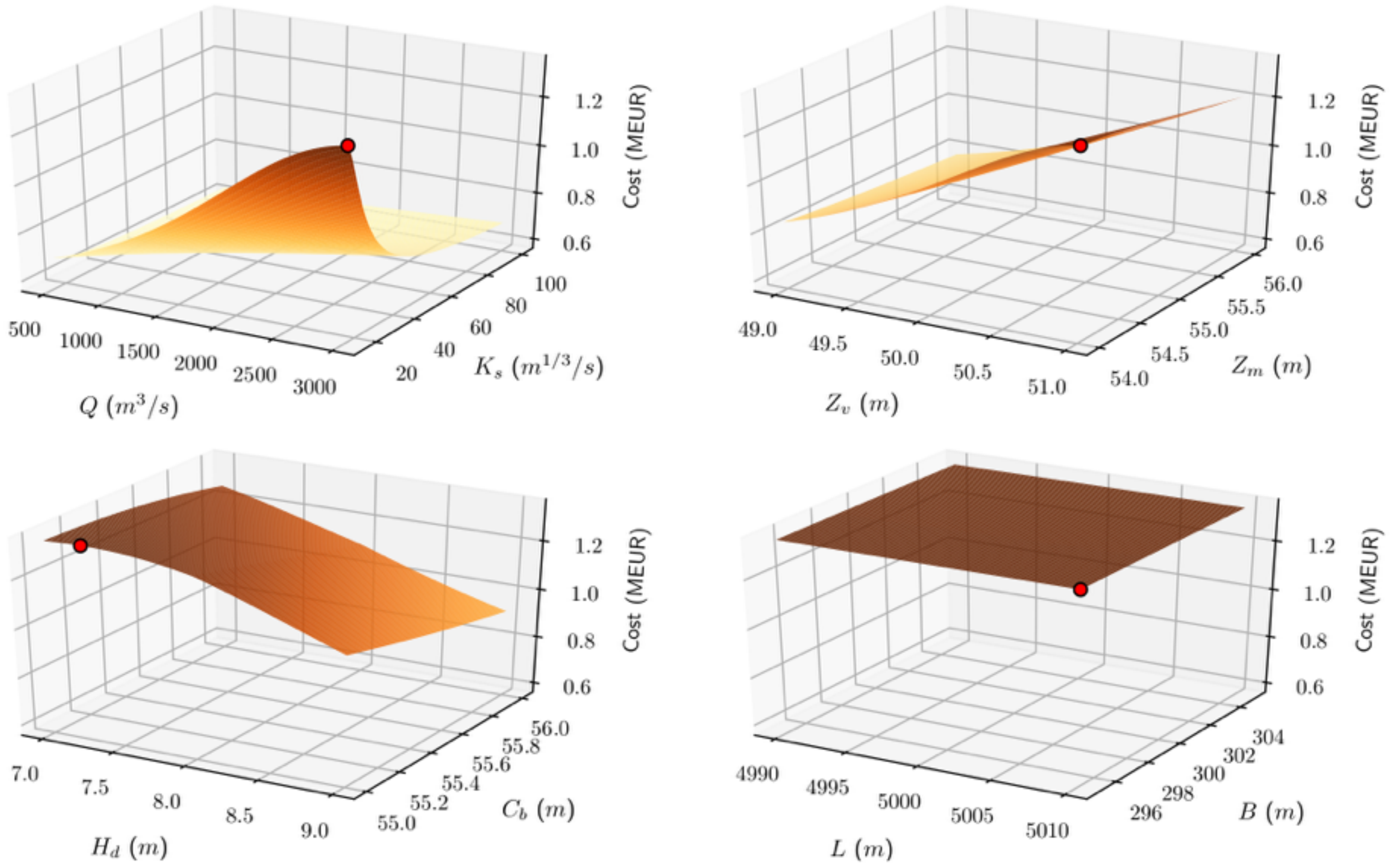
# Results




Parallel coordinates

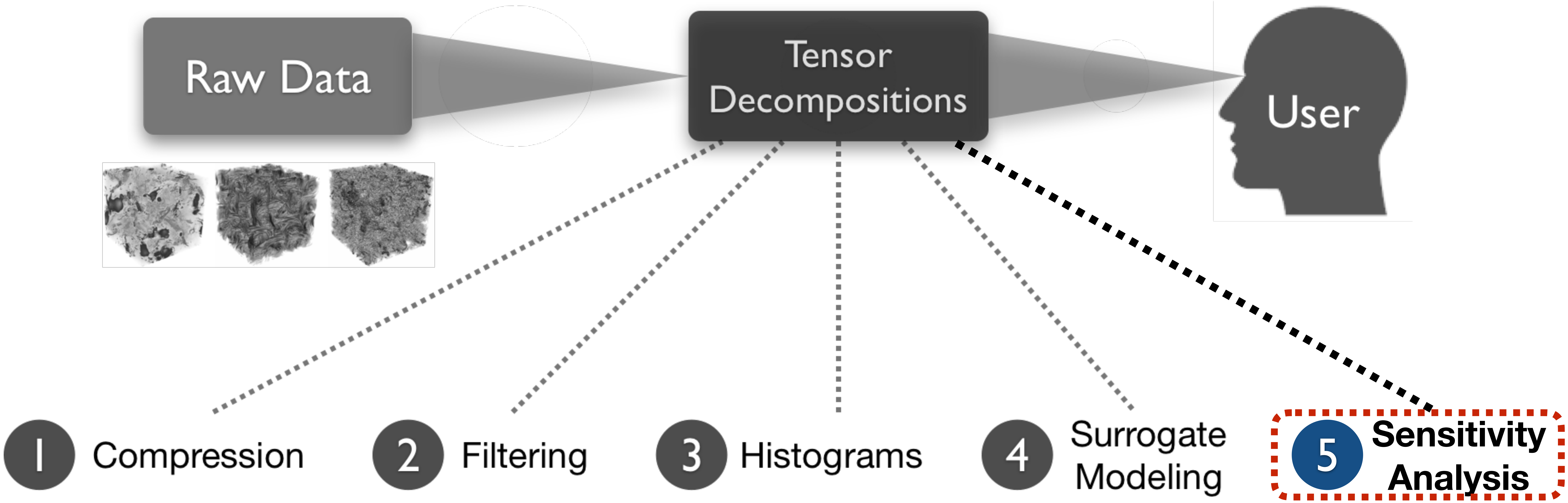



Nested diagrams




Interactive navigation

# Sensitivity Analysis




*"Sobol Tensor Trains for Global Sensitivity Analysis"*

# Sensitivity Analysis

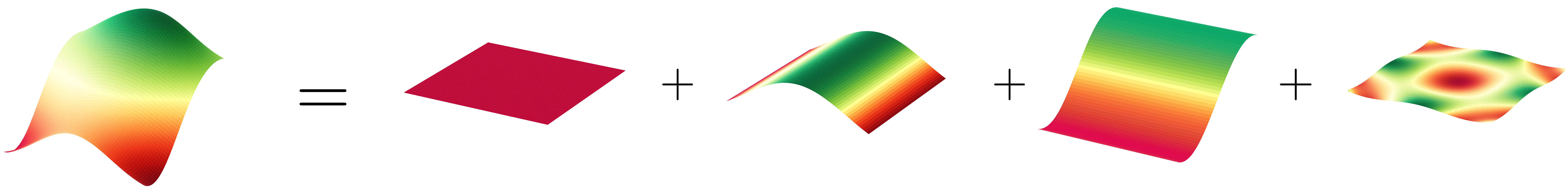
- How does a model **depend on each variable**?
- How about **combinations of variables**?
- Applications:
  - Model interpretation
  - Dimensionality reduction
  - Factor prioritization
  - More informed visualization

# ANOVA Decomposition

- **Partition** a function $f(x_1, \ldots, x_N)$ into subfunctions:
  - A **constant**
  - **One-variable** functions
  - **Two-variable** functions, etc.

[Hoeffding '83]

[Sobol' 90]

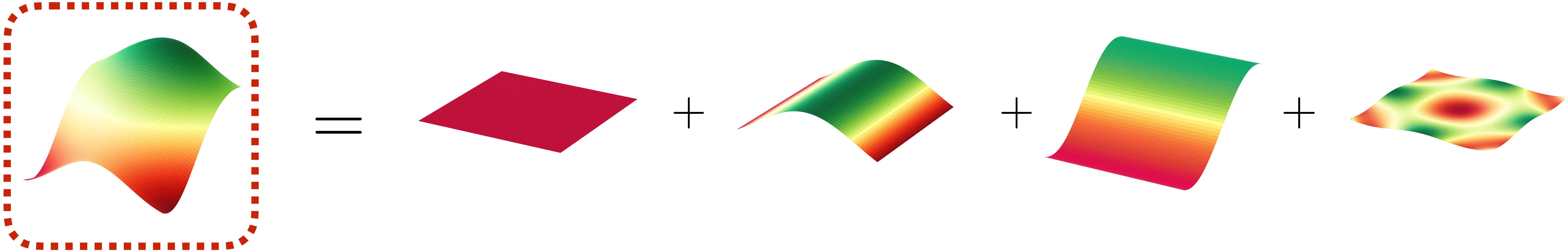


# ANOVA Decomposition

- **Partition** a function $f(x_1, \ldots, x_N)$ into subfunctions:
  - A **constant**
  - **One-variable** functions
  - **Two-variable** functions, etc.

[Hoeffding '83]

[Sobol' 90]

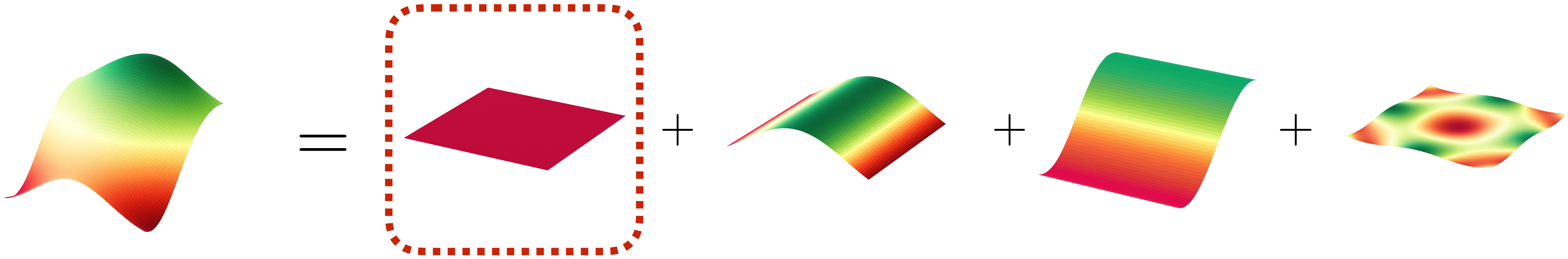


# ANOVA Decomposition

- **Partition** a function $f(x_1, \ldots, x_N)$ into subfunctions:
  - A **constant**
  - **One-variable** functions
  - **Two-variable** functions, etc.

[Hoeffding '83]

[Sobol' 90]

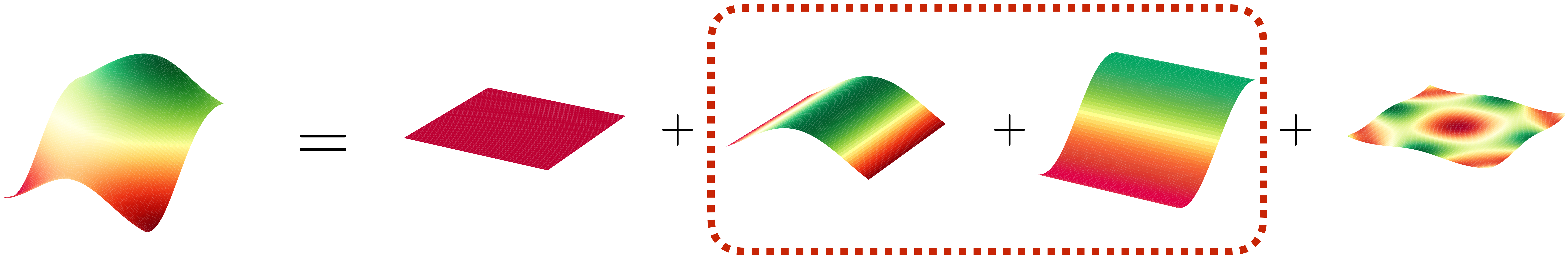


# ANOVA Decomposition

- **Partition** a function $f(x_1, \ldots, x_N)$ into subfunctions:
  - A **constant**
  - **One-variable** functions
  - **Two-variable** functions, etc.

[Hoeffding '83]

[Sobol' 90]

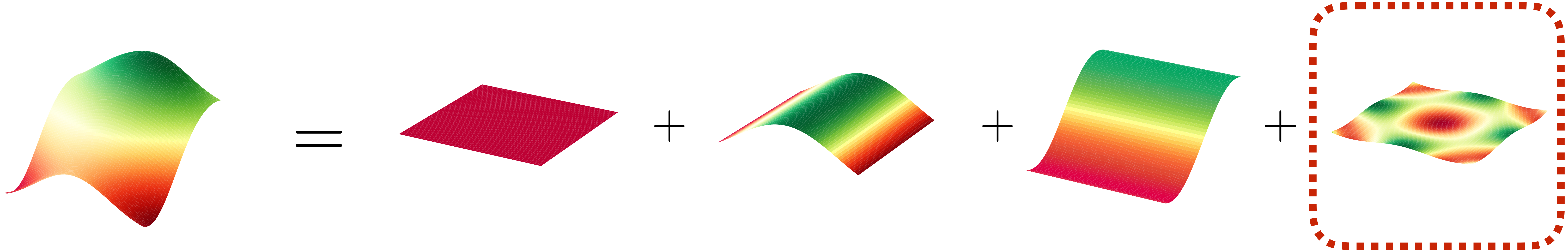


# ANOVA Decomposition

- **Partition** a function $f(x_1, \ldots, x_N)$ into subfunctions:
  - A **constant**
  - **One-variable** functions
  - **Two-variable** functions, etc.

[Hoeffding '83]

[Sobol' 90]

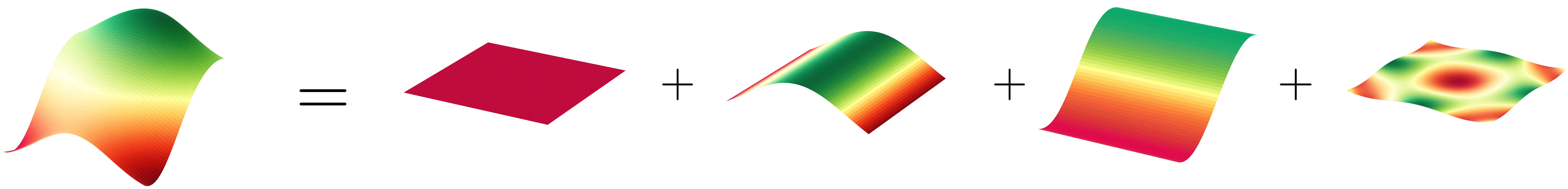


# ANOVA Decomposition

- **Partition** a function $f(x_1, \ldots, x_N)$ into subfunctions:
  - A **constant**
  - **One-variable** functions
  - **Two-variable** functions, etc.

[Hoeffding ‘83]

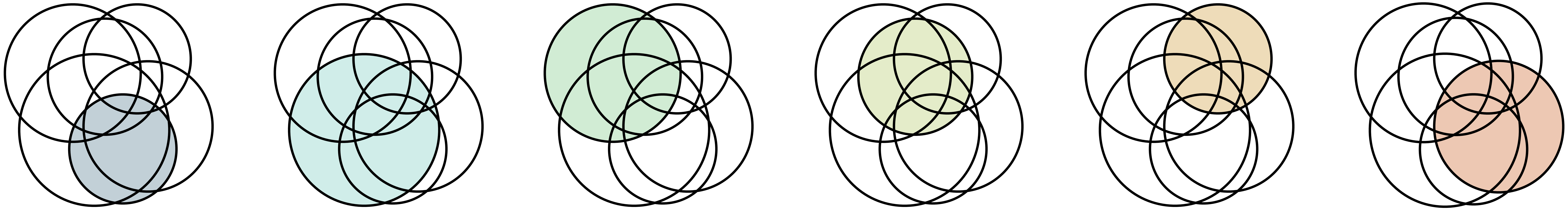
[Sobol’ 90]



- They encode **all possible variable interactions**

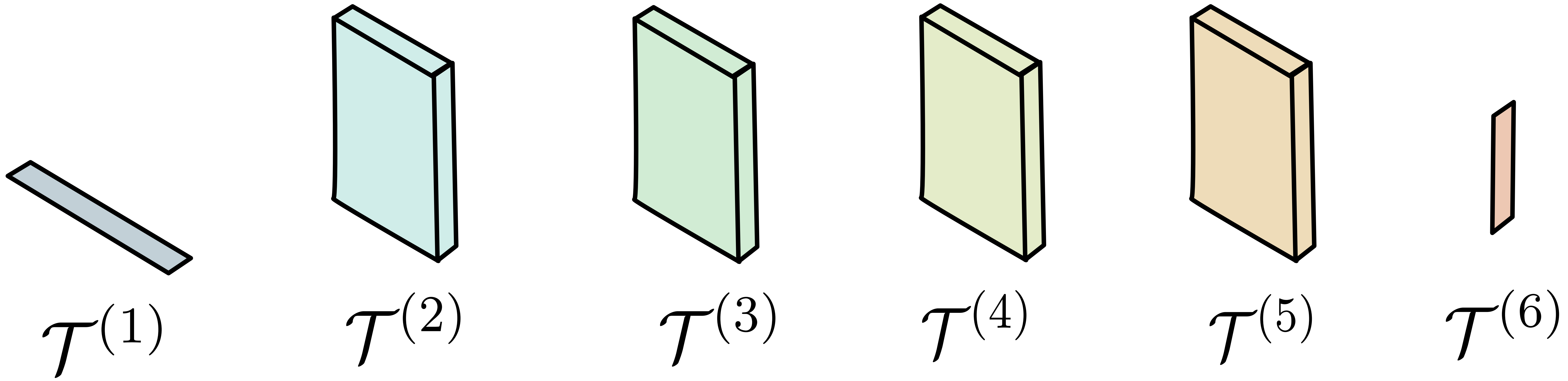
# Sobol Indices

- Variance % induced by each subfunction
- One for **each possible tuple** of variables



- Past approaches: compute **one at a time** [Sobol ’90], [Sudret ’08], [Dolgov ’14], [Song ’16]

# Proposed Method




- First algorithm that gathers **all Sobol indices in one structure**
  - Compressed TT of size $2 \times \cdots \times 2$
- How? We build up a **TT surrogate**
- We extract **all ANOVA** terms **in one go**
- Then compute their **variance**

# Results

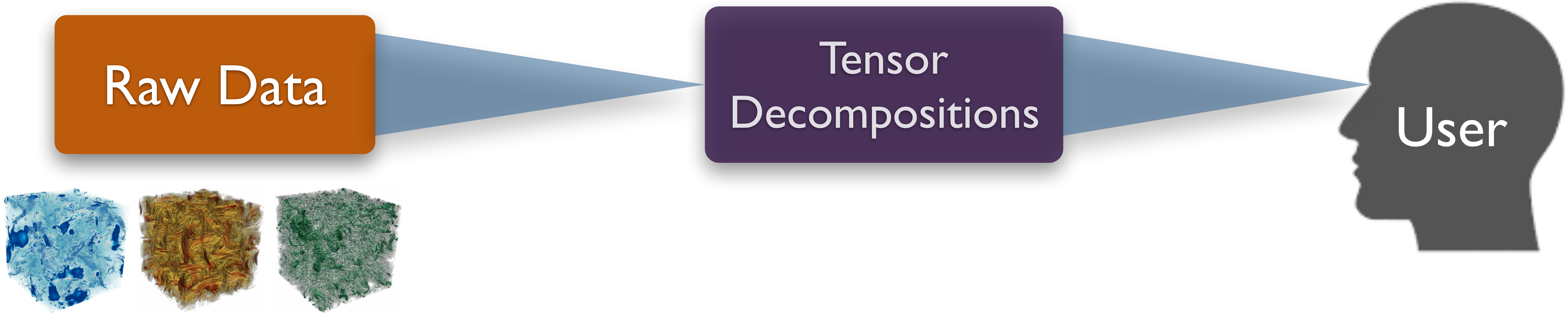
- Handle the **full index set at once**
- Derive **sensitivity metrics**
- **Optimization**: find largest indices
- **Sensitivity queries**:
  - "What 3 variables influence the least?"
  - "Find 8 variables that capture 95% of the variance"
  - "Now do so including certain variables"

Few seconds

# Summary

| | | |
|---|---|---|
| 1 | Compression | *"Lossy Volume Compression Using Tucker Truncation and Thresholding"* |
| 2 | Filtering | *"Multiresolution Volume Filtering in the Tensor Compressed Domain"* |
| 3 | Histograms | *"Tensor Decompositions for Integral Histogram Compression and Look-up"* |
| 4 | Surrogate Modeling | *"A Surrogate Visualization Model Using the Tensor Train Format"* |
| 5 | Sensitivity Analysis | *"Sobol Tensor Trains for Global Sensitivity Analysis"* |

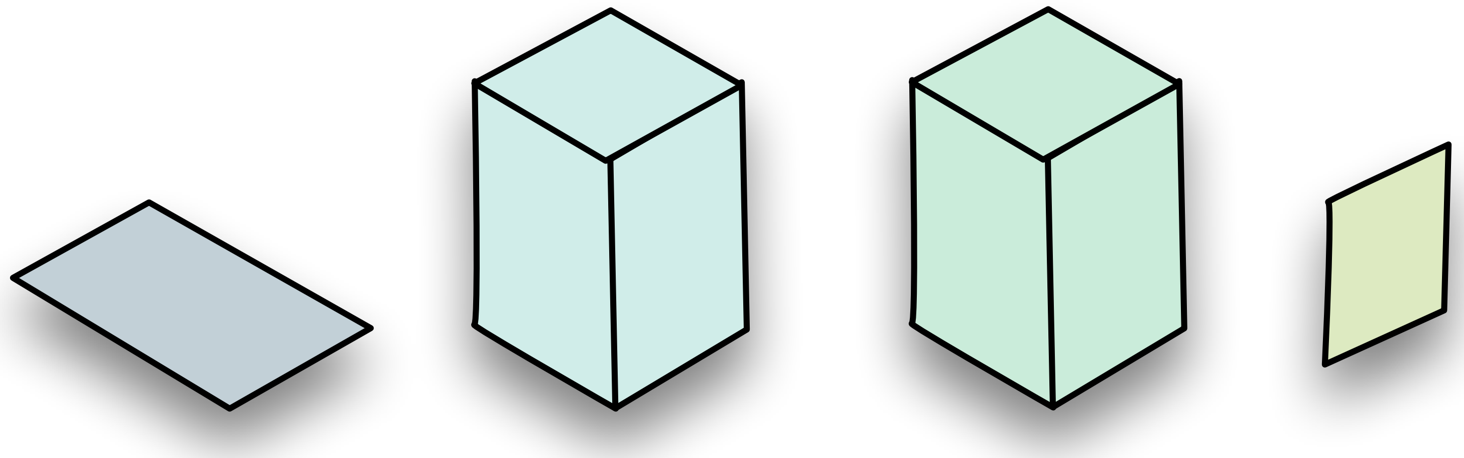
# Conclusions



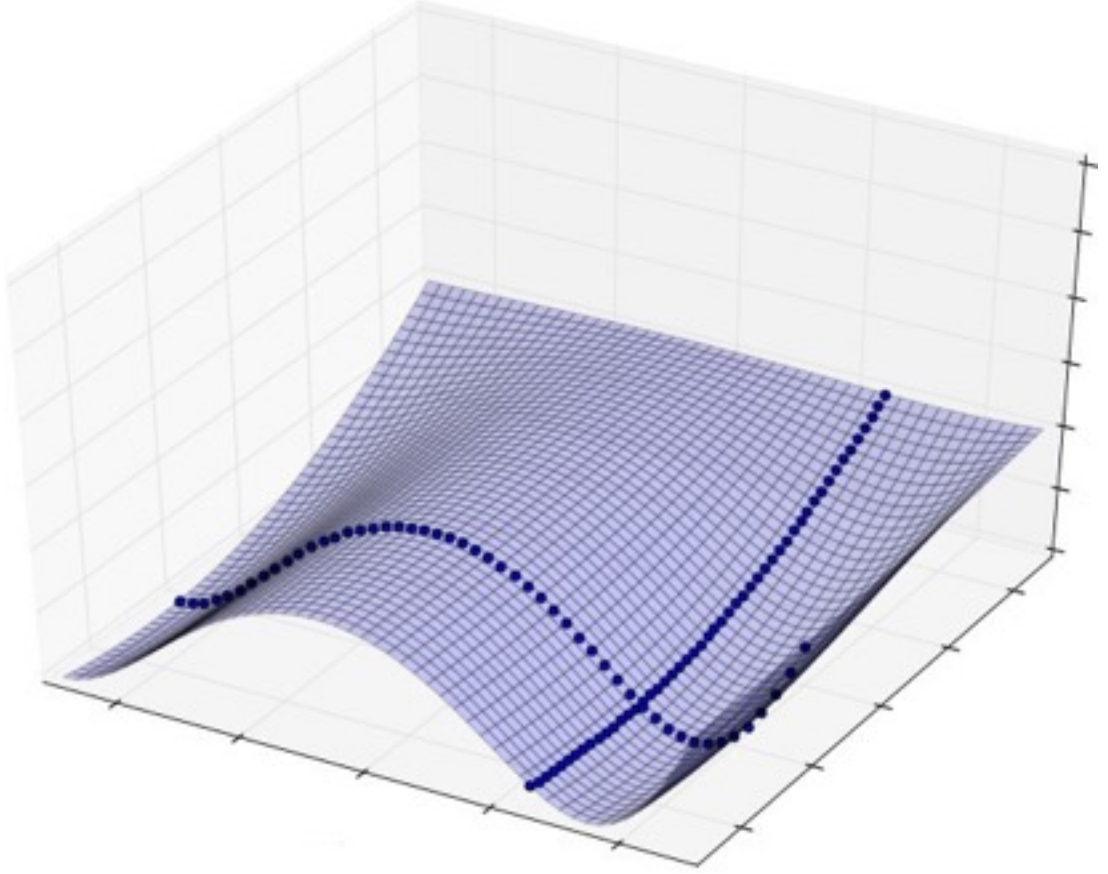

- **Tensor decompositions**: rich toolbox for **analysis and visualization**
- Real-time results
- Any number of dimensions
    - 3D: **Tucker model**
    - General case: **tensor trains** as a *golden standard*

# Future Research

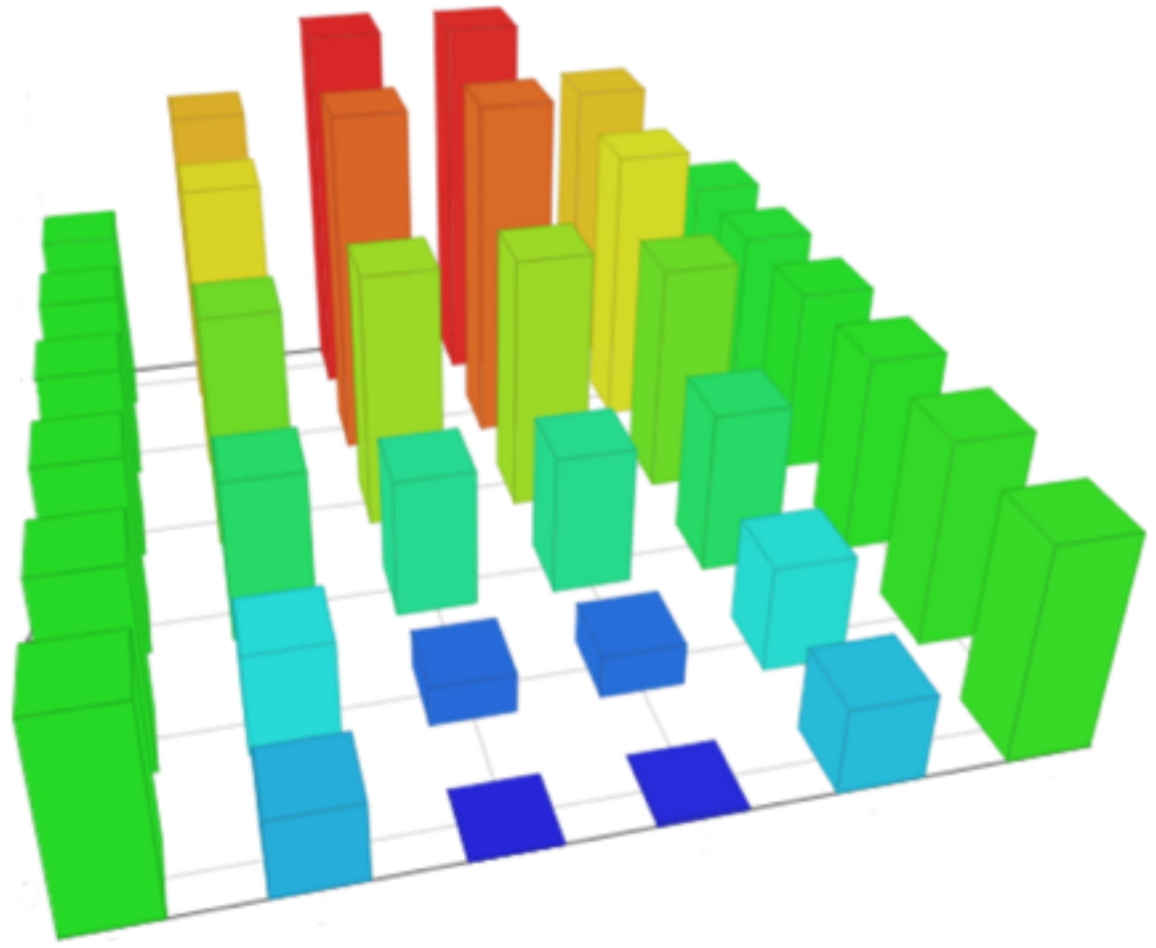
Tensor train compression



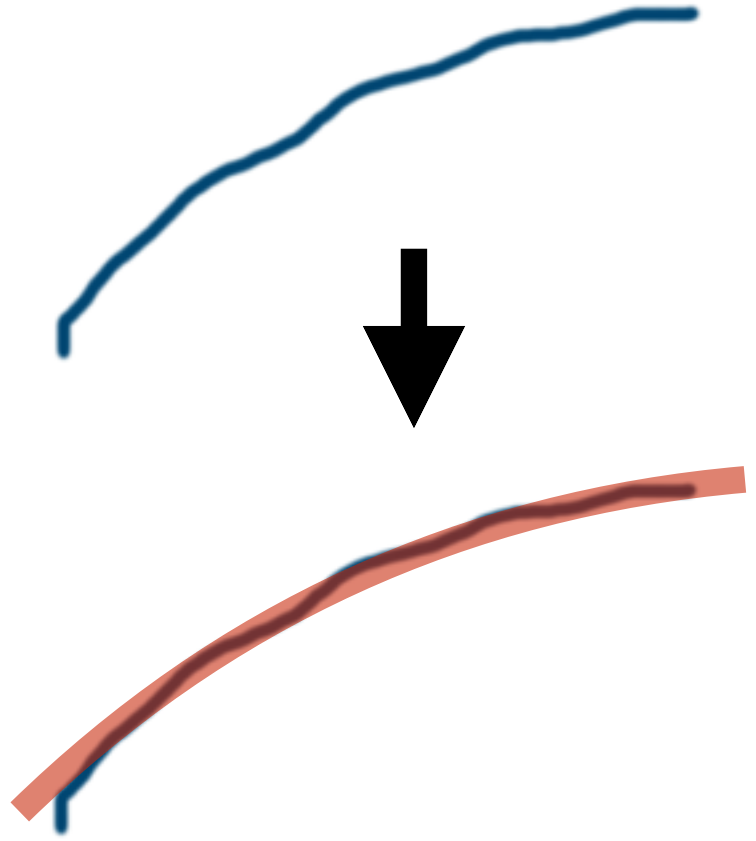
Smooth interpolation



Categorical visualization



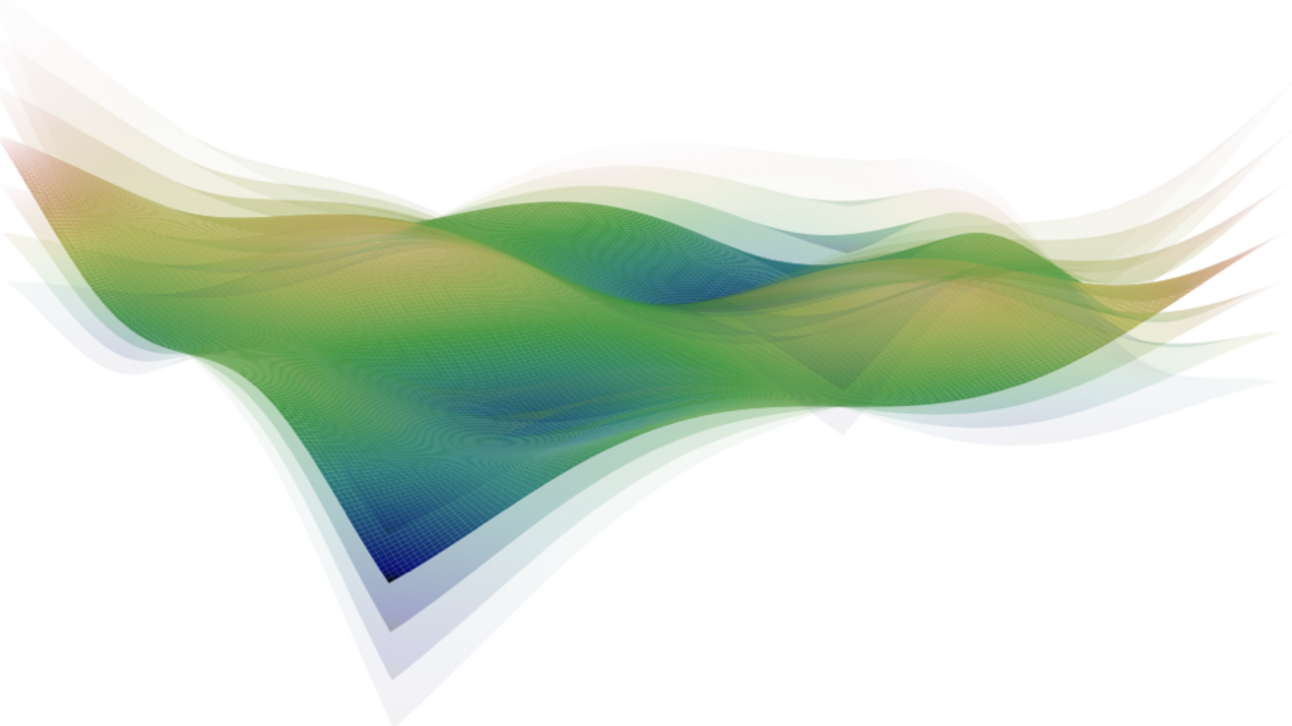
Sketch-based queries

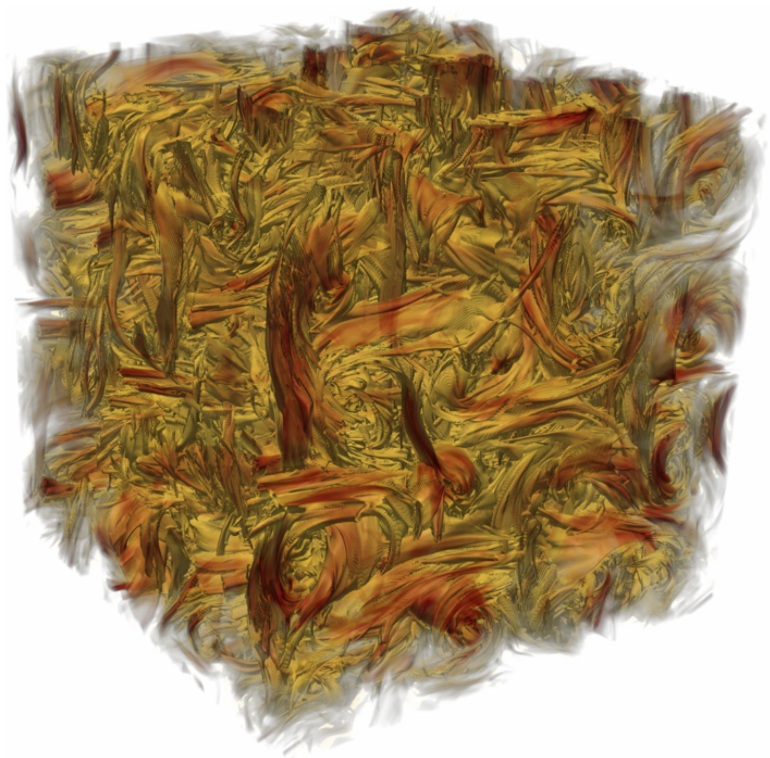


VISUALIZATIONAND MULTIMEDIALAB

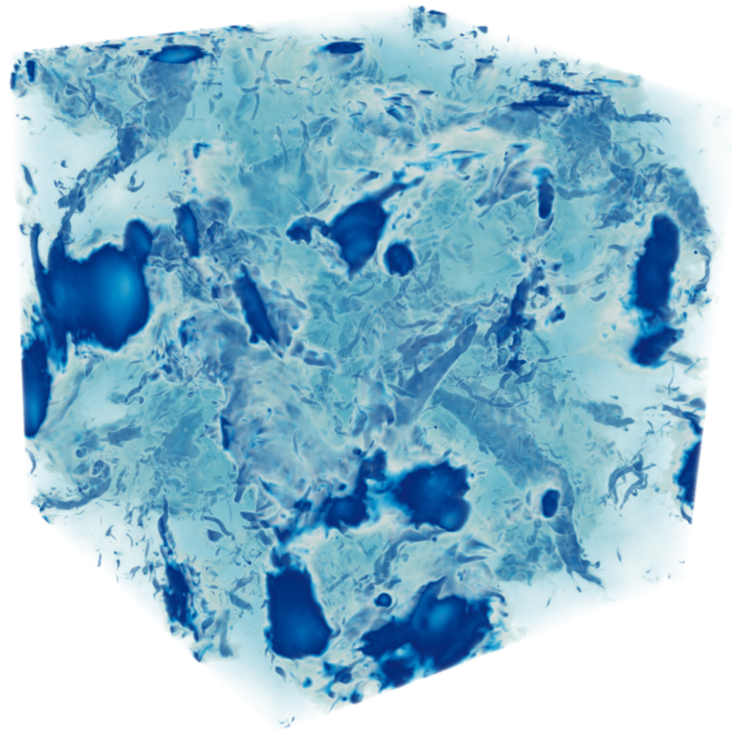
University of Zurich[UZH]

VISUALIZATIONAND MULTIMEDIALAB



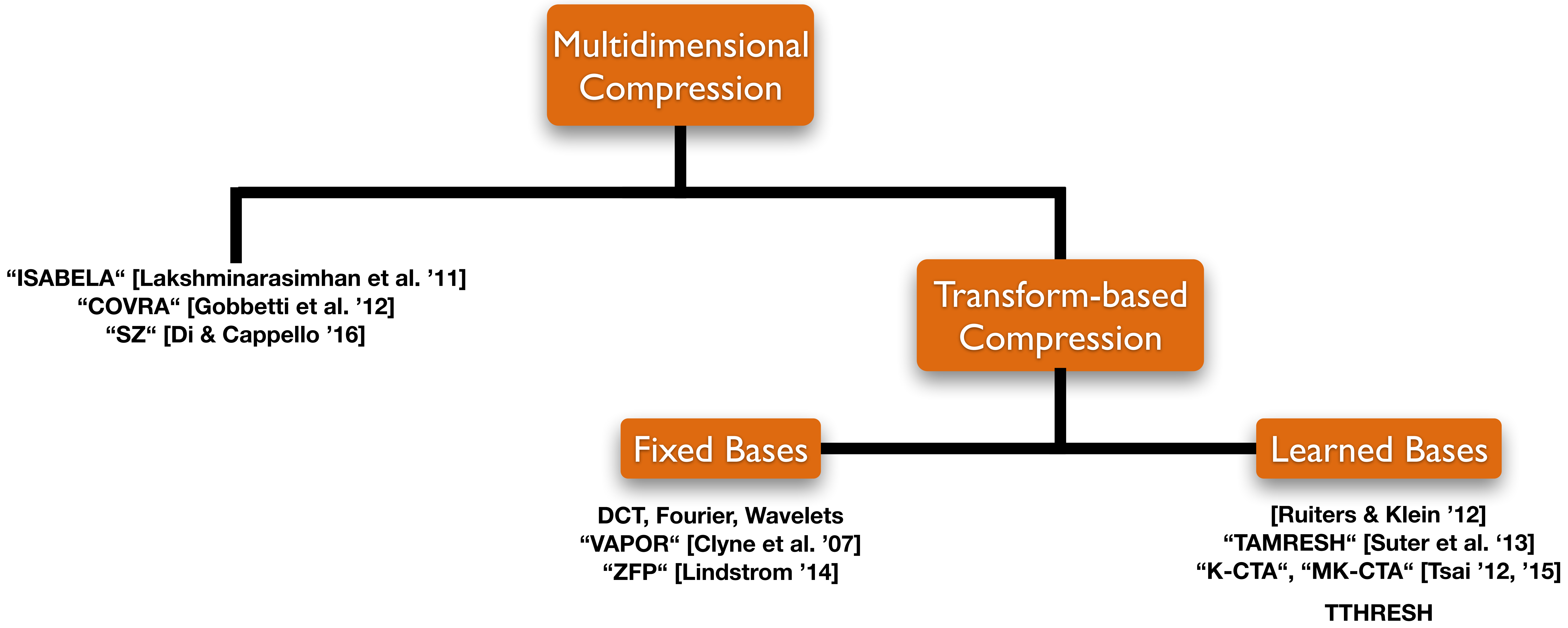


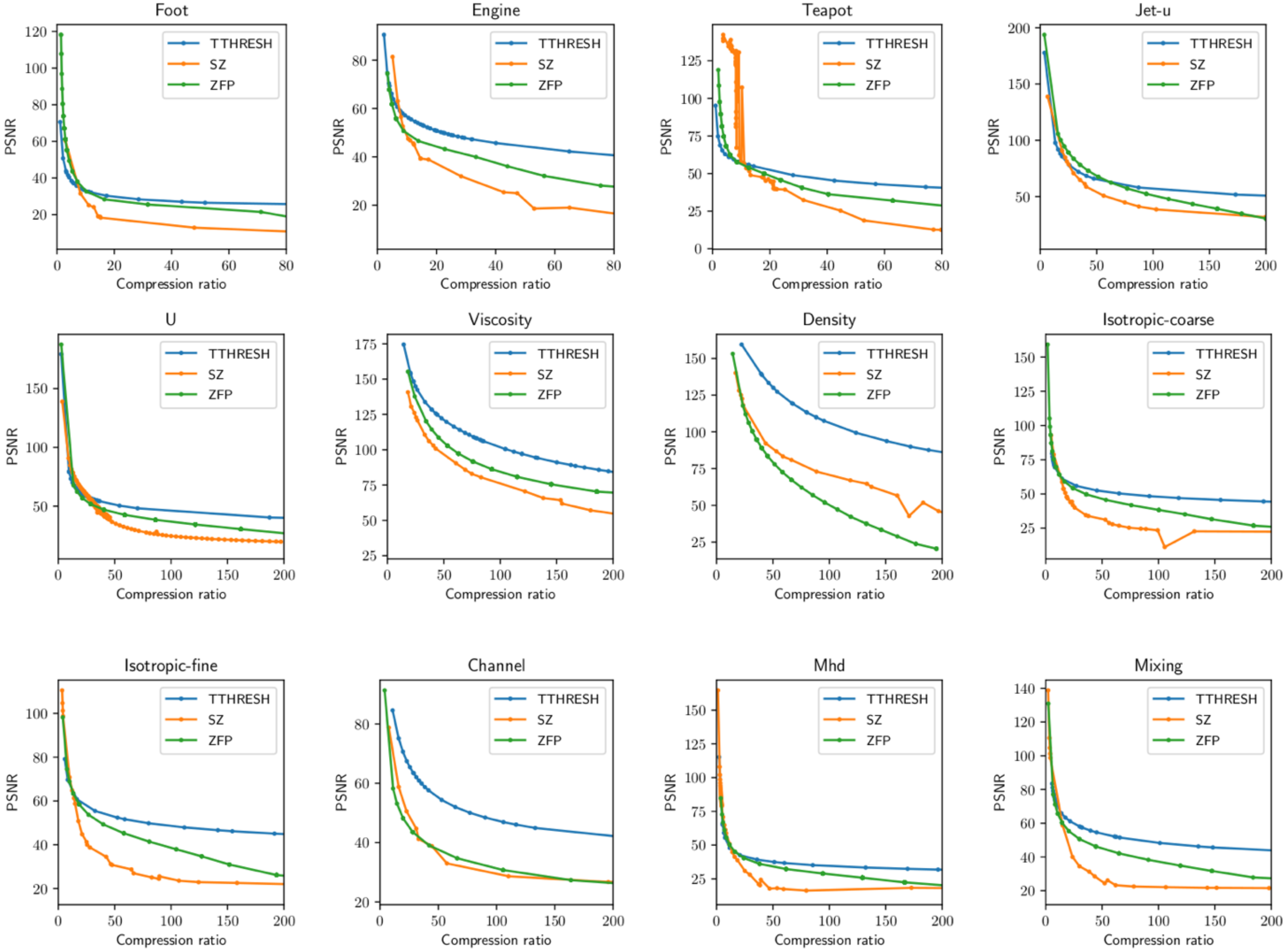
# Thank You!



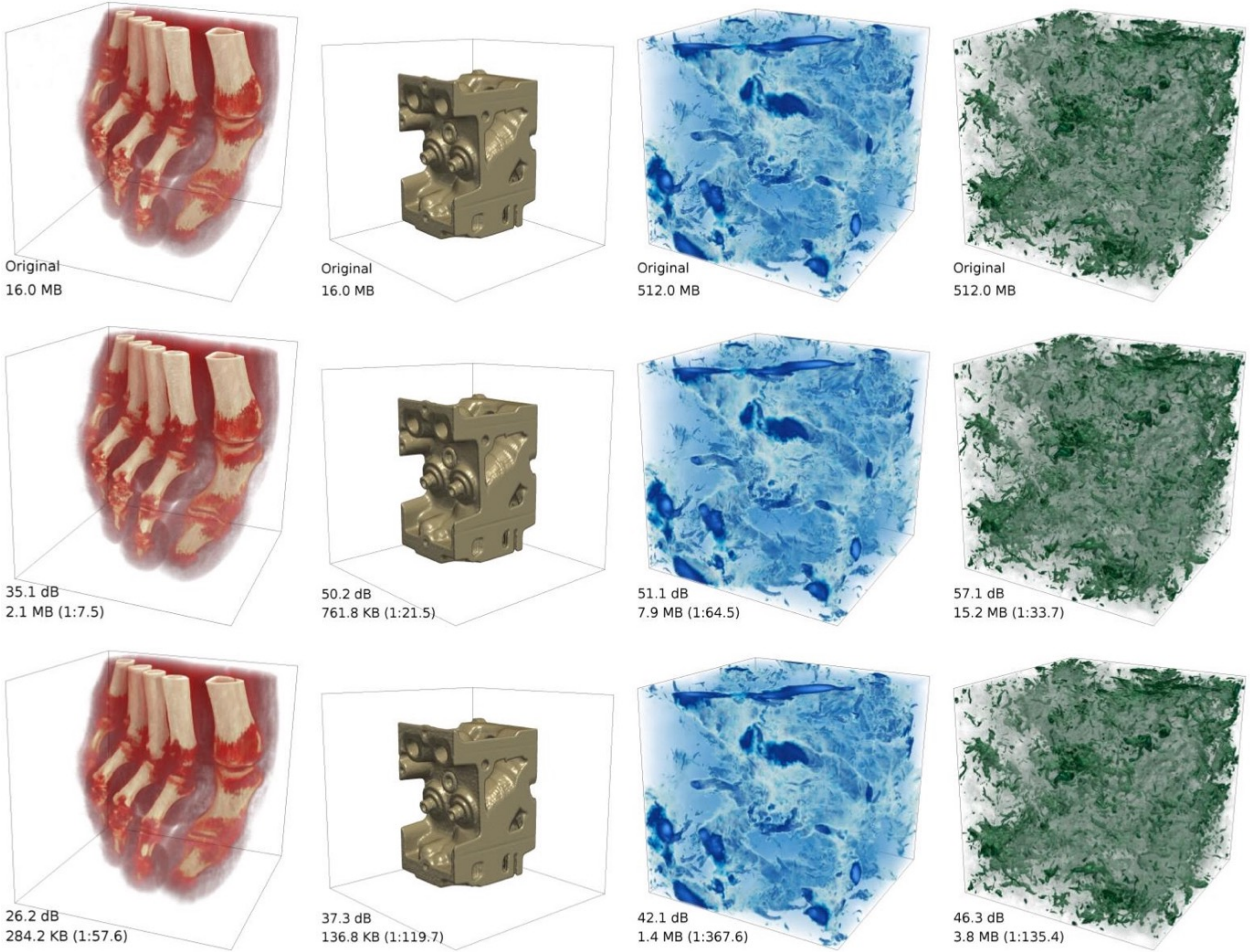
# State-of-the-Art

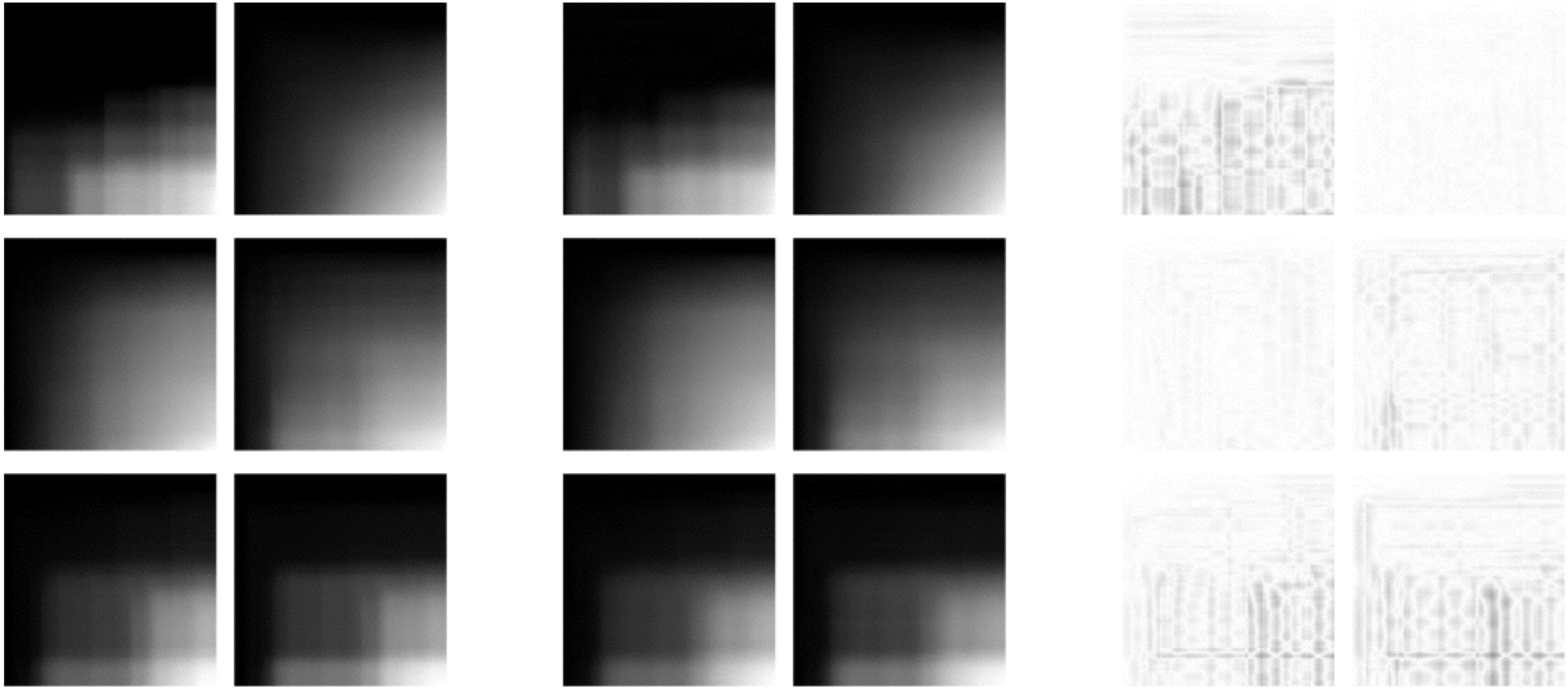
- Many compression algorithms for 2D, fewer for 3D



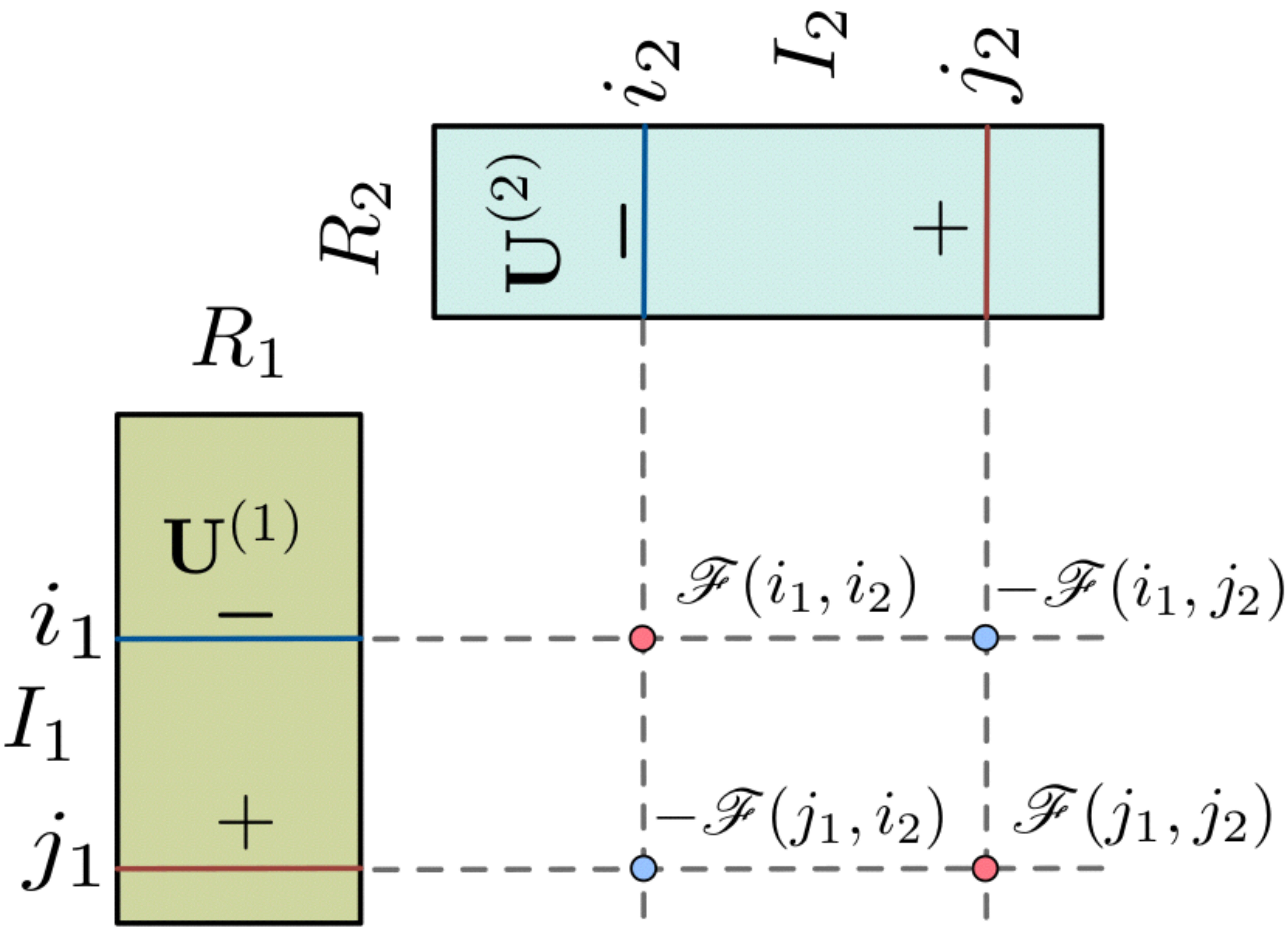


Foot
Engine
Teapot
Jet-u
U
Viscosity
Density
Isotropic-coarse
Isotropic-fine
Channel
Mhd
Mixing
TTHRESH
SZ
ZFP
PSNR
Compression ratio


Original
16.0 MB
Original
16.0 MB
Original
512.0 MB
Original
512.0 MB
35.1 dB
2.1 MB (1:7.5)
50.2 dB
761.8 KB (1:21.5)
51.1 dB
7.9 MB (1:64.5)
57.1 dB
15.2 MB (1:33.7)
26.2 dB
284.2 KB (1:57.6)
37.3 dB
136.8 KB (1:119.7)
42.1 dB
1.4 MB (1:367.6)
46.3 dB
3.8 MB (1:135.4)

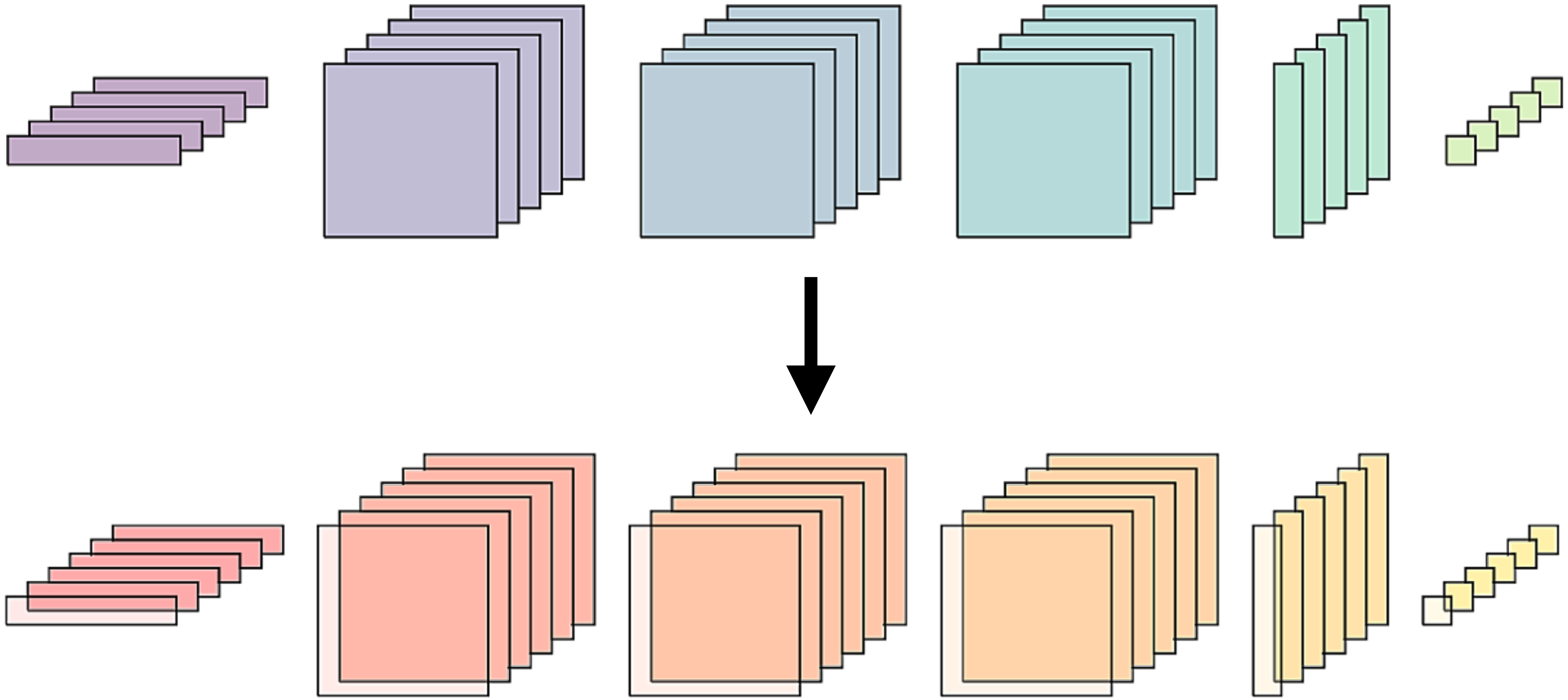


(a) Original (b) After compression (c) Absolute error (x10 amplified)


$i_2$
$I_2$
$j_2$
$R_2$
$\mathbf{U}^{(2)}$
$-$
$+$
$R_1$
$\mathbf{U}^{(1)}$
$i_1$
$-$
$I_1$
$+$
$j_1$
$\mathscr{F}(i_1, i_2)$
$-\mathscr{F}(i_1, j_2)$
$-\mathscr{F}(j_1, i_2)$
$\mathscr{F}(j_1, j_2)$

# ANOVA Tensors

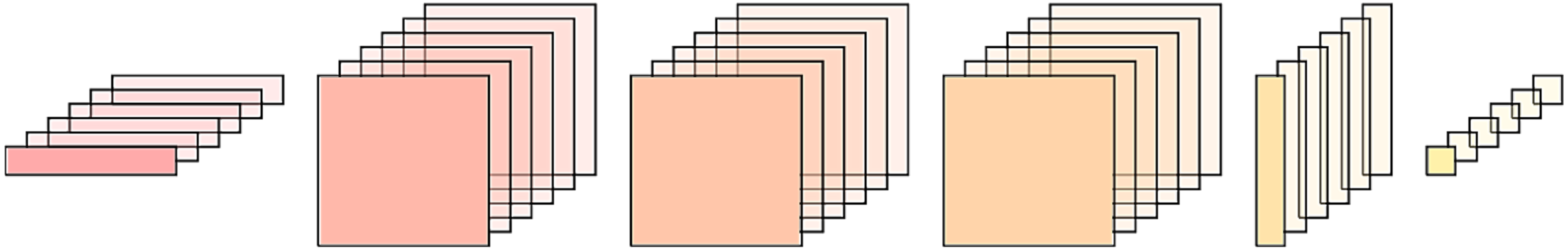
- We extract the ANOVA decomposition of any TT in negligible time
- We create an extra slice ⟶ allows us to encode all ANOVA terms



VISUALIZATIONANDMULTIMEDIALAB

# Indexing

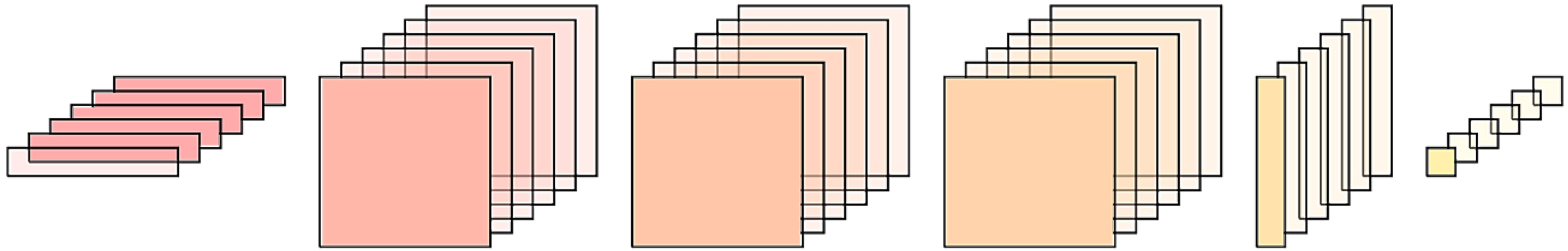
- Exploit the full expressive power of the TT format
- New slice encodes who is in the subscript



$$= f_{\emptyset}$$

# Indexing

- Exploit the full expressive power of the TT format
- New slice encodes who is in the subscript

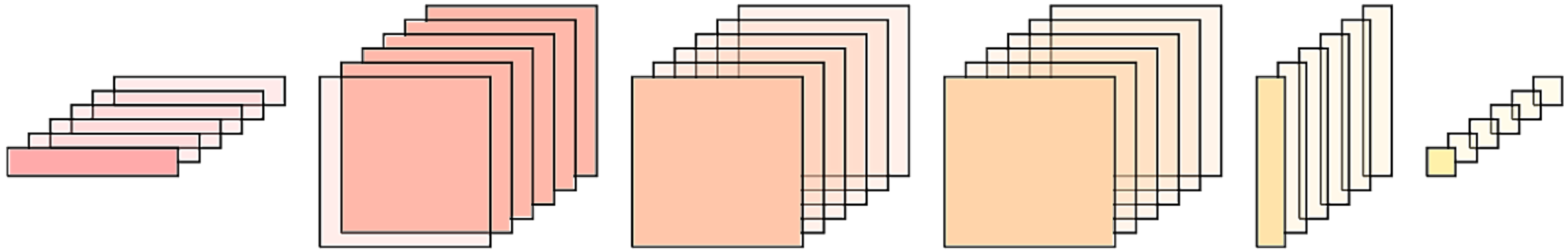


$$= f_1(x_1)$$

VISUALIZATIONANDMULTIMEDIALAB

# Indexing

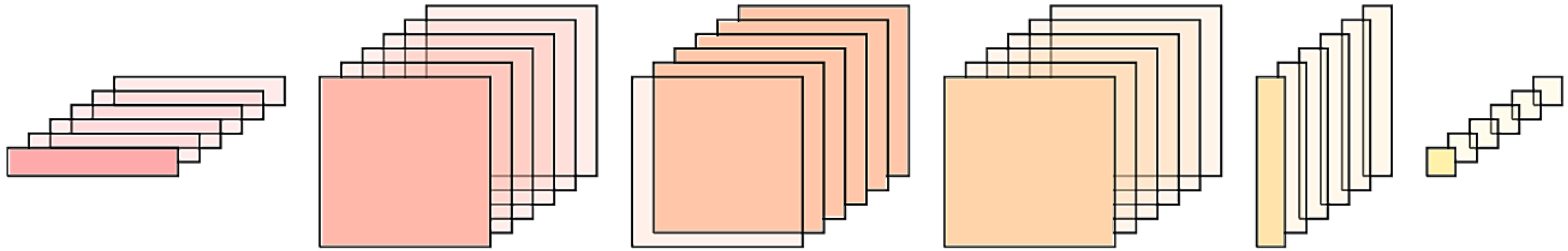
- Exploit the full expressive power of the TT format
- New slice encodes who is in the subscript



$$= f_2(x_2)$$

# Indexing

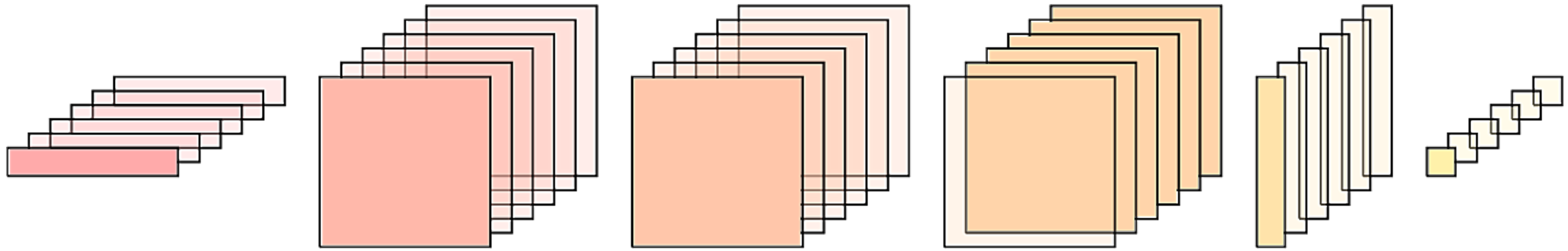
- Exploit the full expressive power of the TT format
- New slice encodes who is in the subscript



$$= f_3(x_3)$$

# Indexing

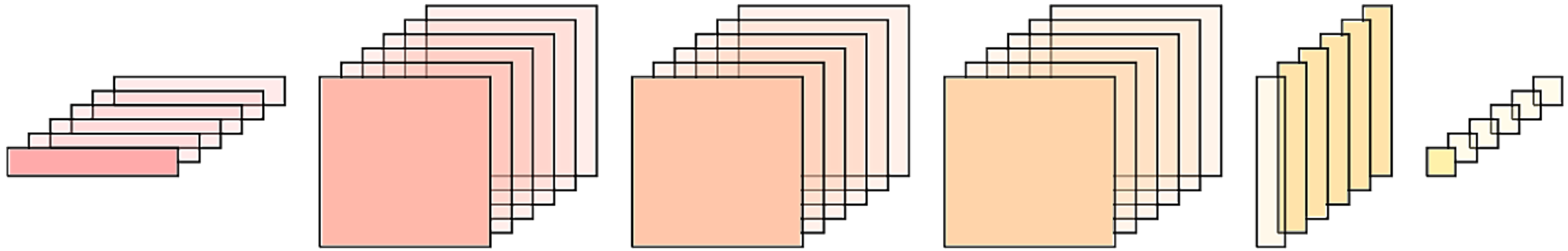
- Exploit the full expressive power of the TT format
- New slice encodes who is in the subscript



$$= f_4(x_4)$$

# Indexing

- Exploit the full expressive power of the TT format
- New slice encodes who is in the subscript

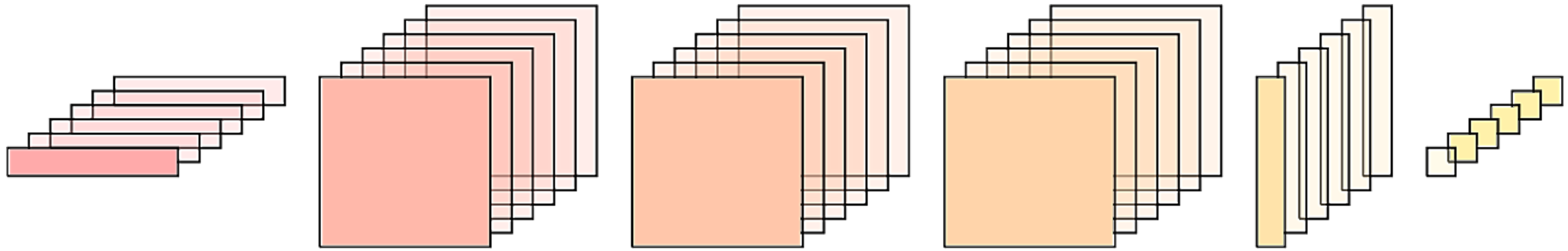


$$= f_5(x_5)$$

VISUALIZATIONAND MULTIMEDIALAB

# Indexing

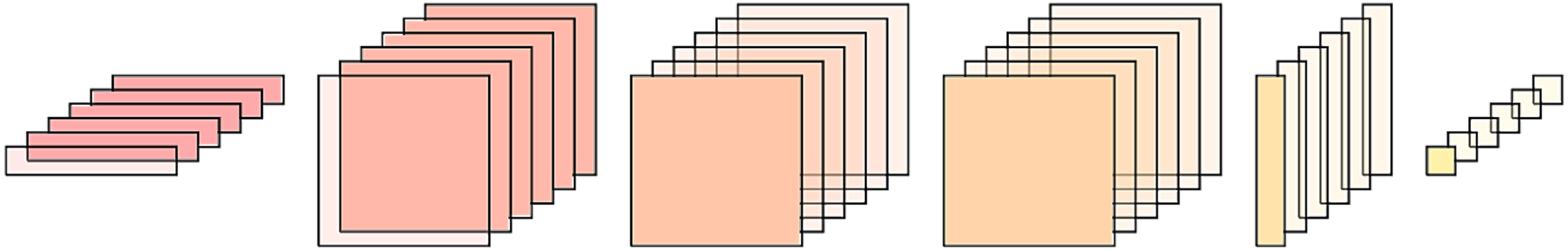
- Exploit the full expressive power of the TT format
- New slice encodes who is in the subscript



$$= f_6(x_6)$$

# Indexing

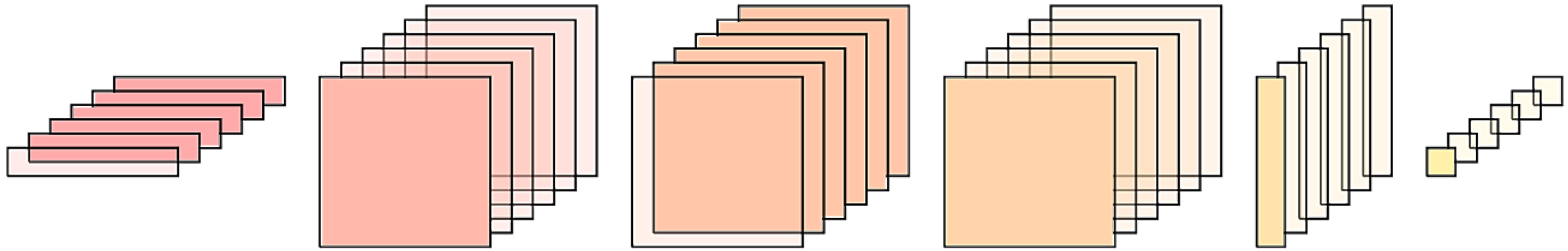
- Exploit the full expressive power of the TT format
- New slice encodes who is in the subscript



$$= f_{1,2}(x_{1,2})$$

# Indexing

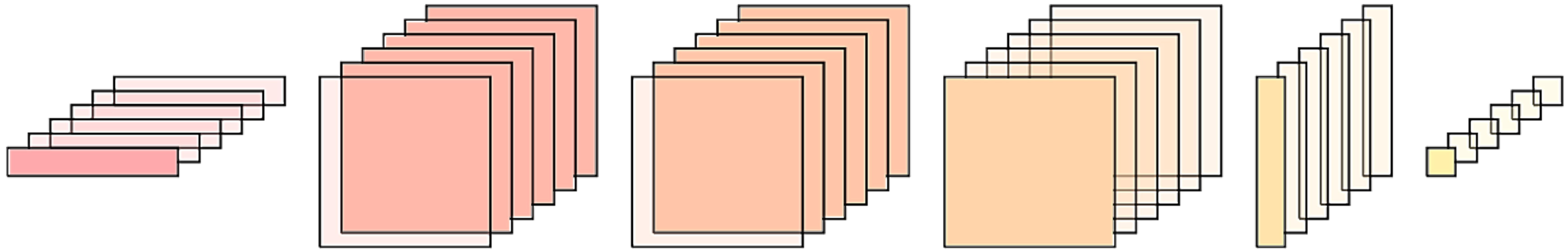
- Exploit the full expressive power of the TT format
- New slice encodes who is in the subscript



$$= f_{1,3}(x_{1,3})$$

# Indexing

- Exploit the full expressive power of the TT format
- New slice encodes who is in the subscript

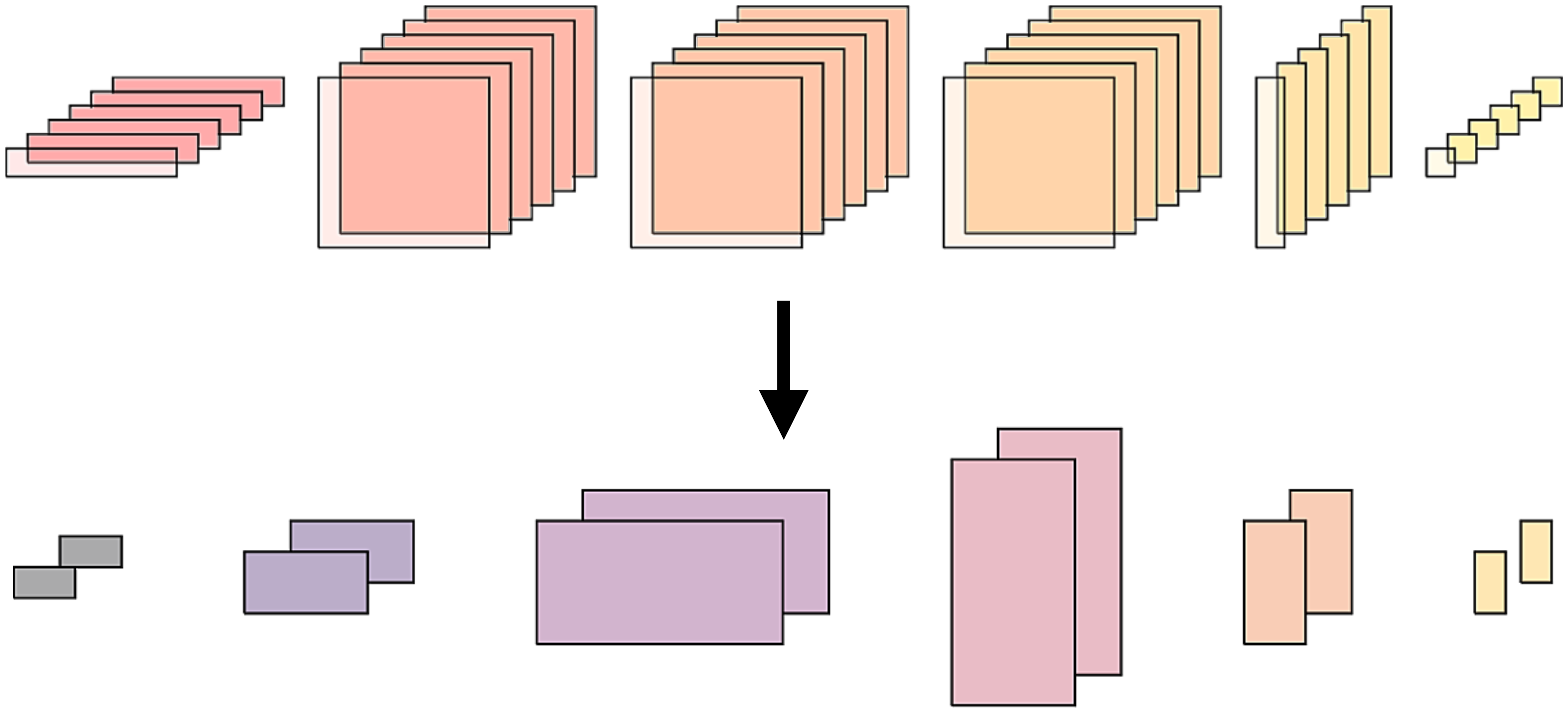


$$= f_{2,3}(x_{2,3})$$

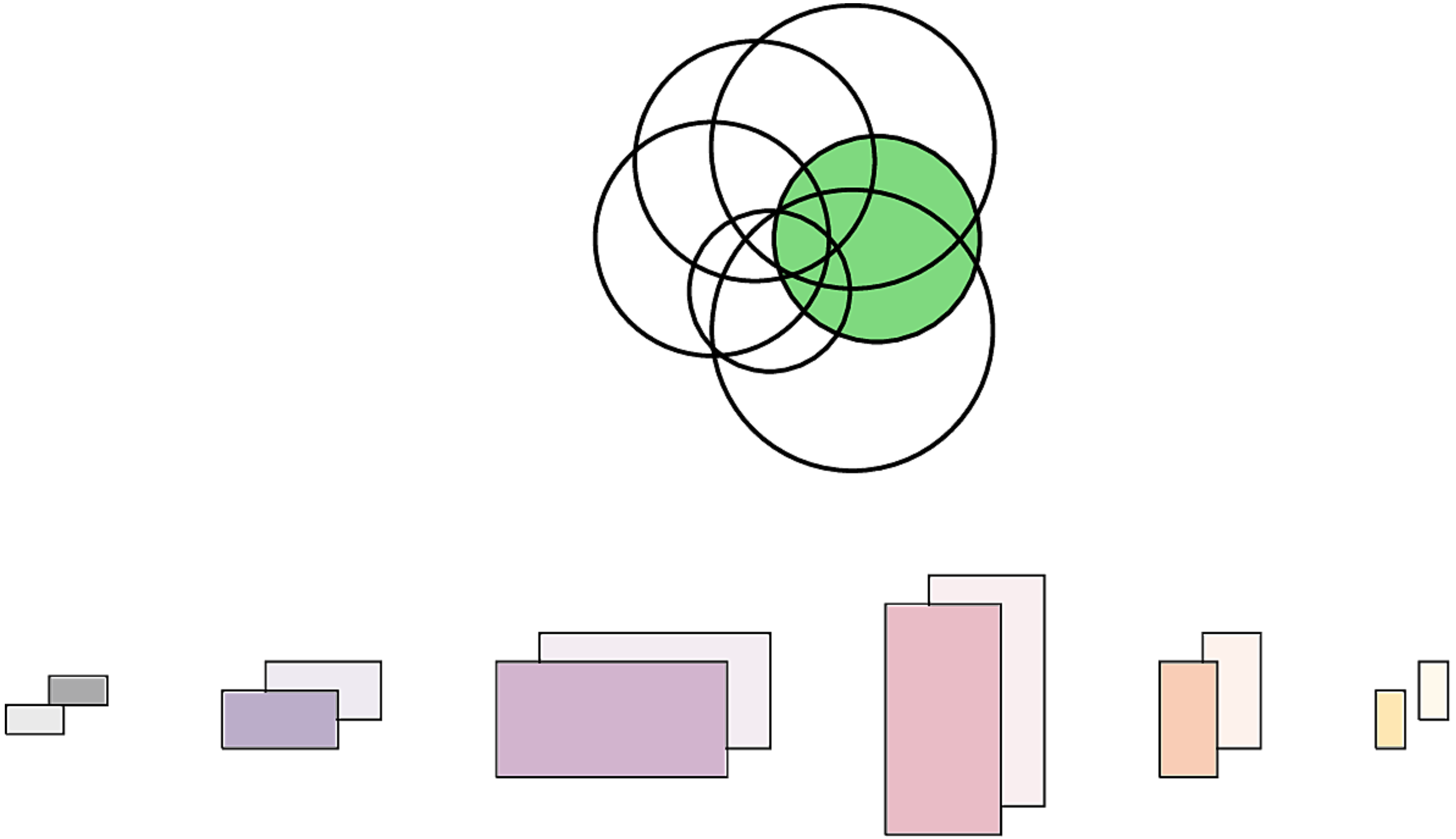
VISUALIZATIONAND MULTIMEDIALAB

# Sobol Tensor Trains

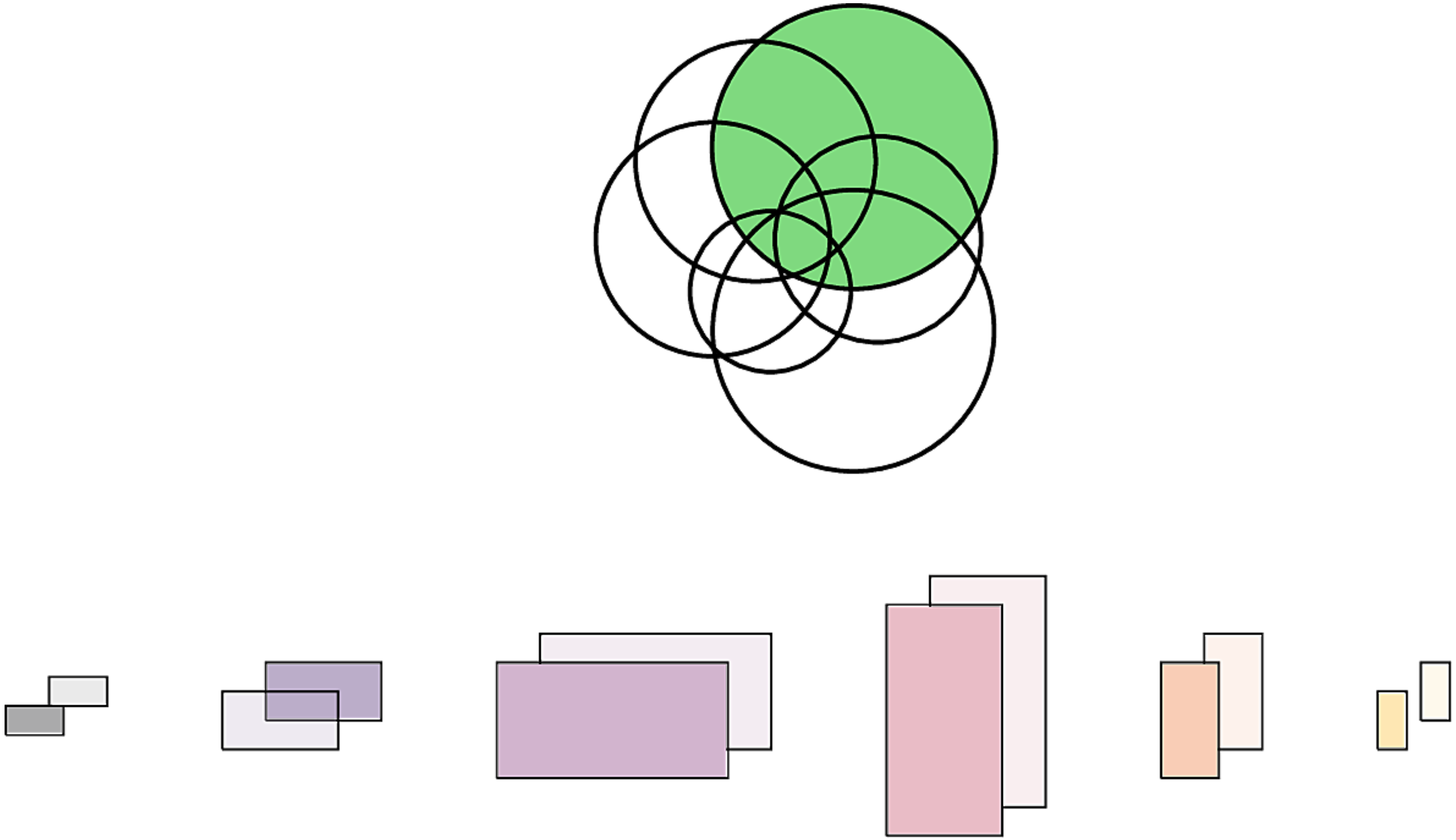
- Next, we calculate the variance of the whole tensor **at once**
  - Result: all Sobol indices in one tensor!



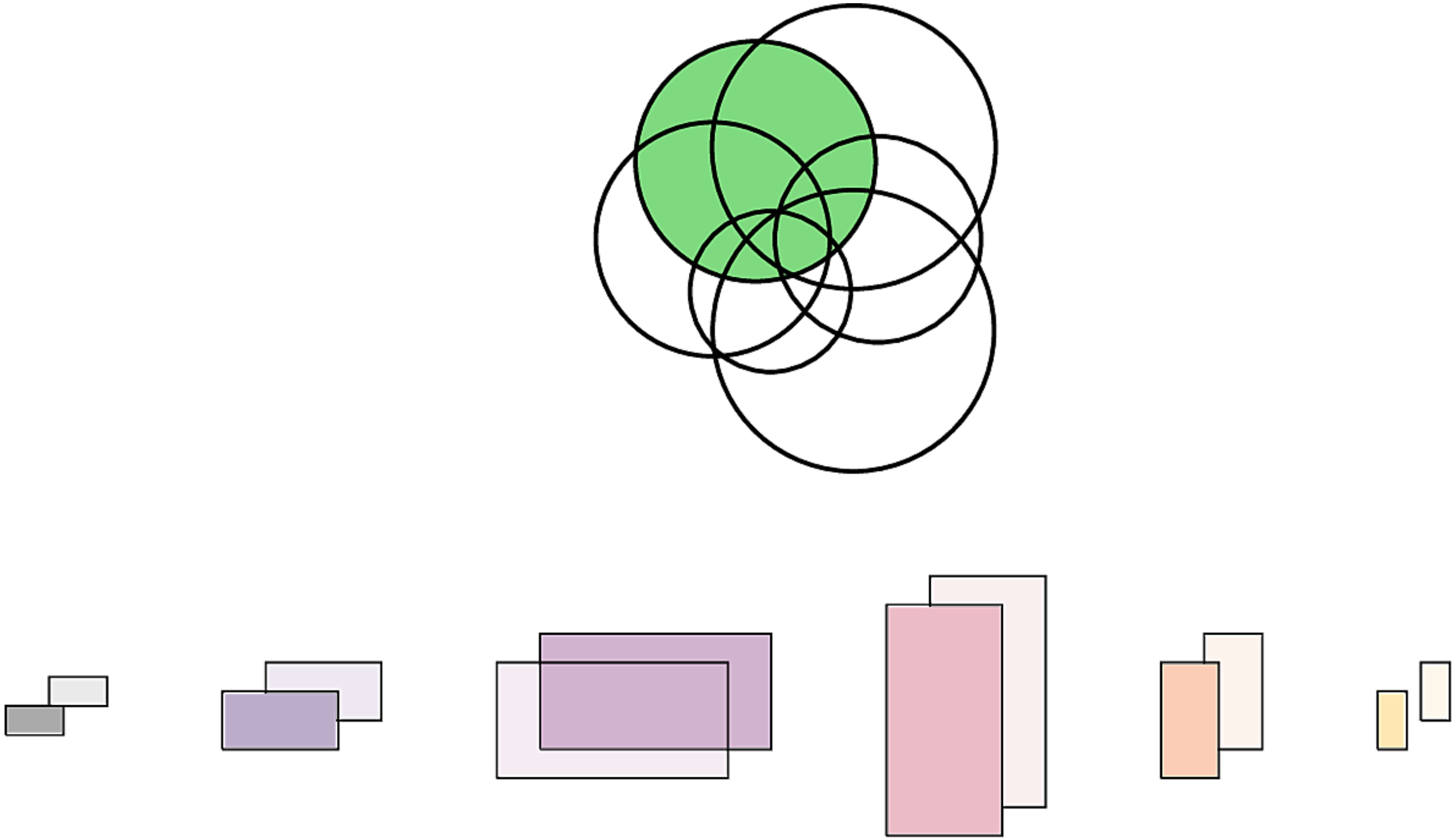
# Sobol Tensor Trains



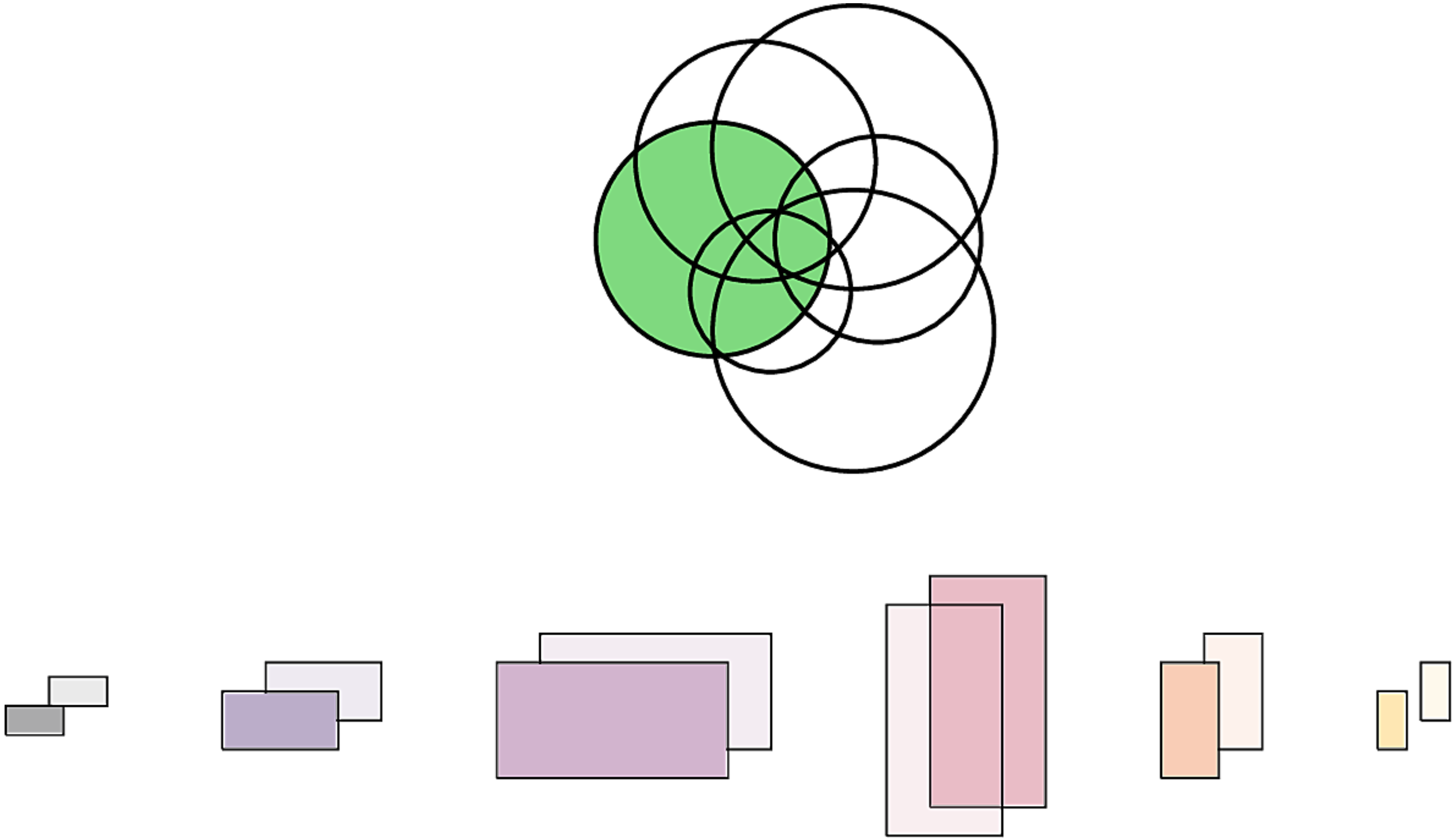
# Sobol Tensor Trains



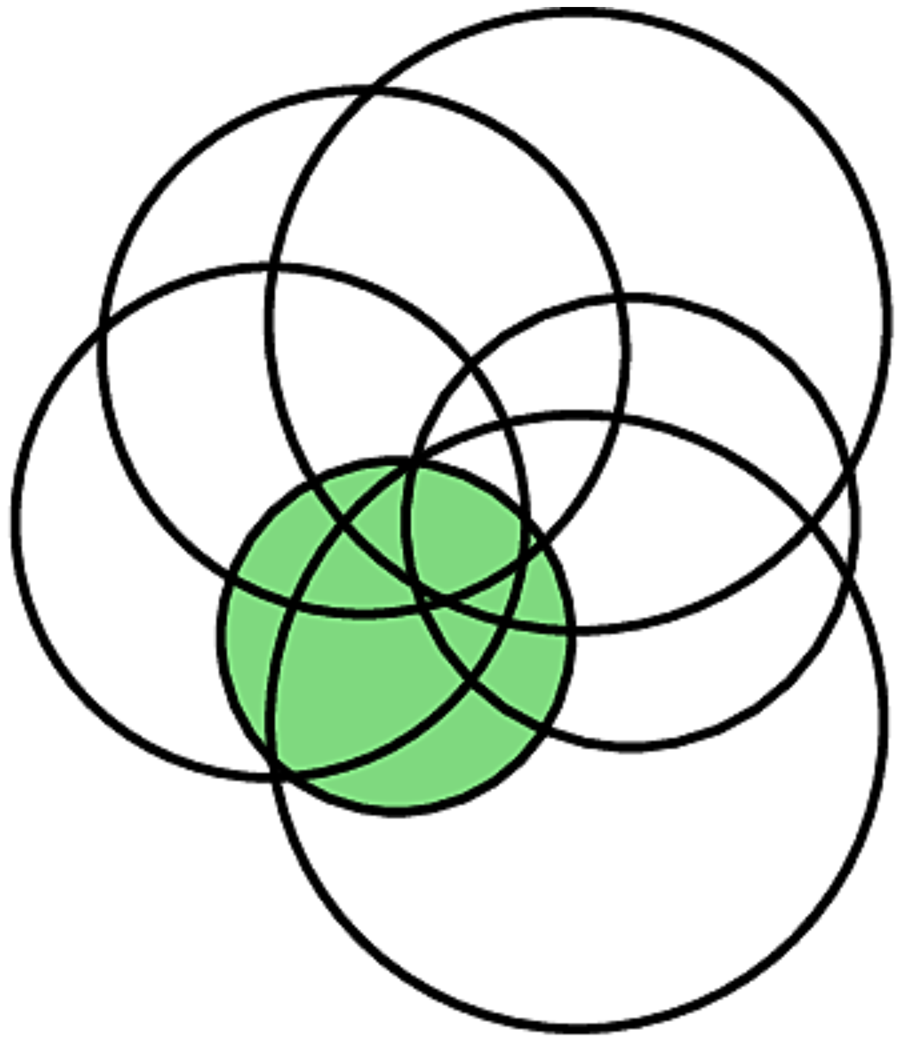
# Sobol Tensor Trains

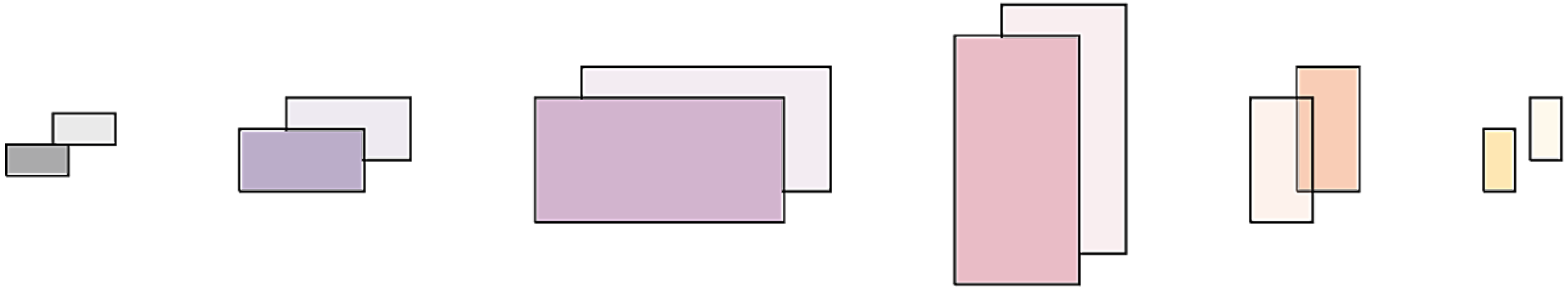


# Sobol Tensor Trains



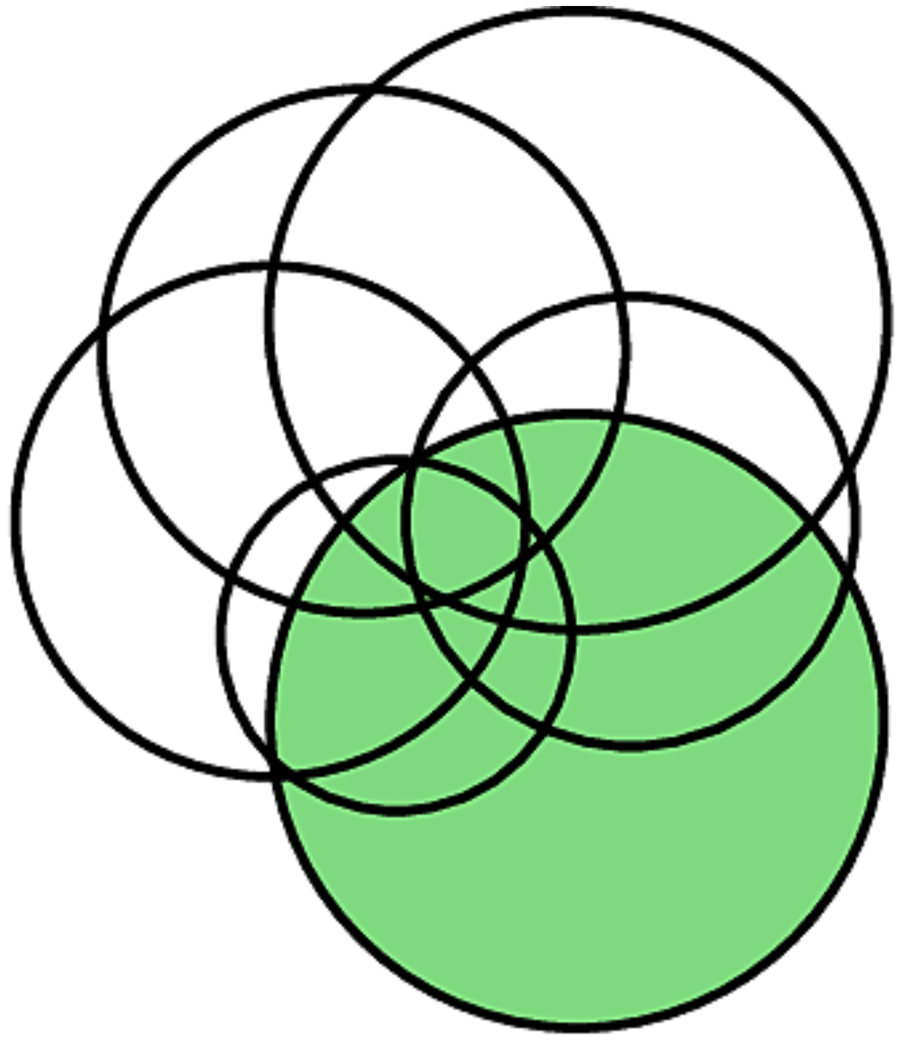
# Sobol Tensor Trains

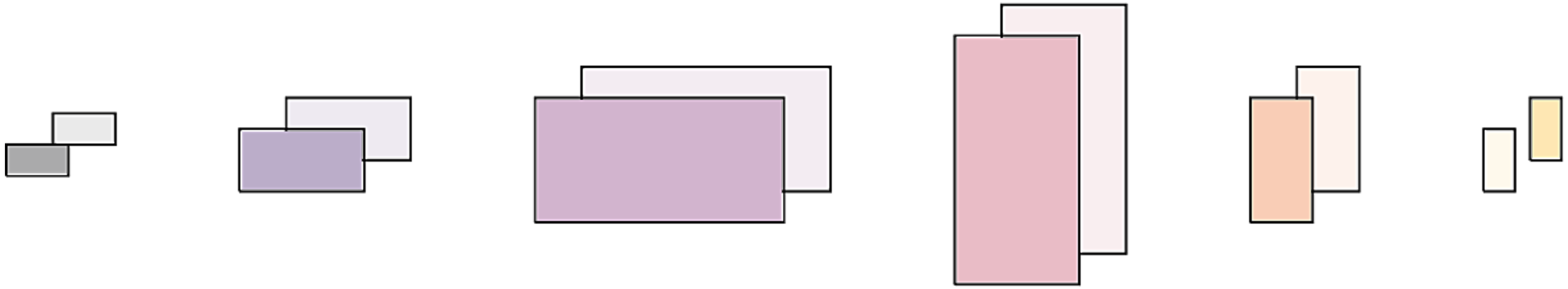




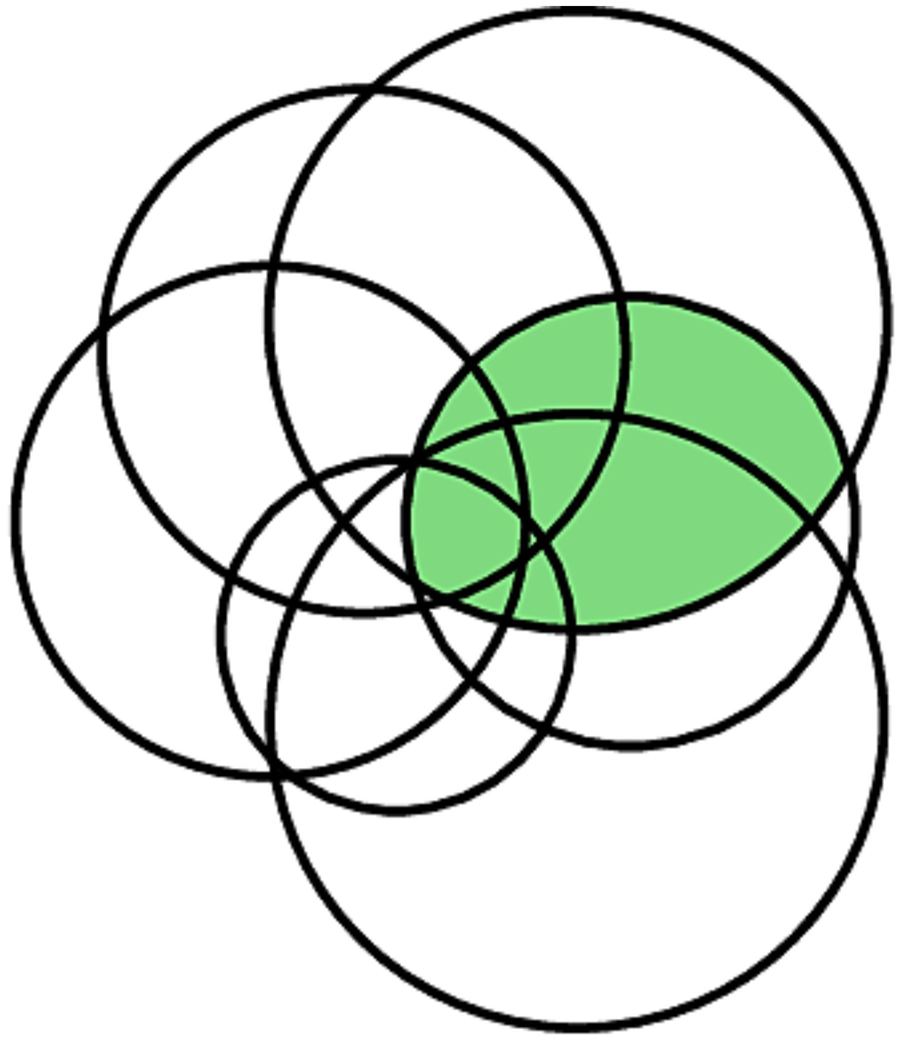
VISUALIZATIONAND
MULTIMEDIALAB

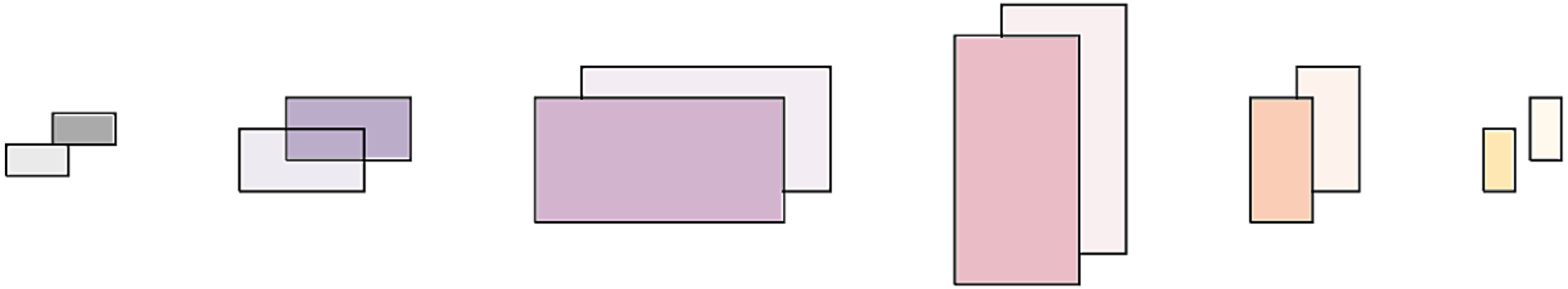
# Sobol Tensor Trains



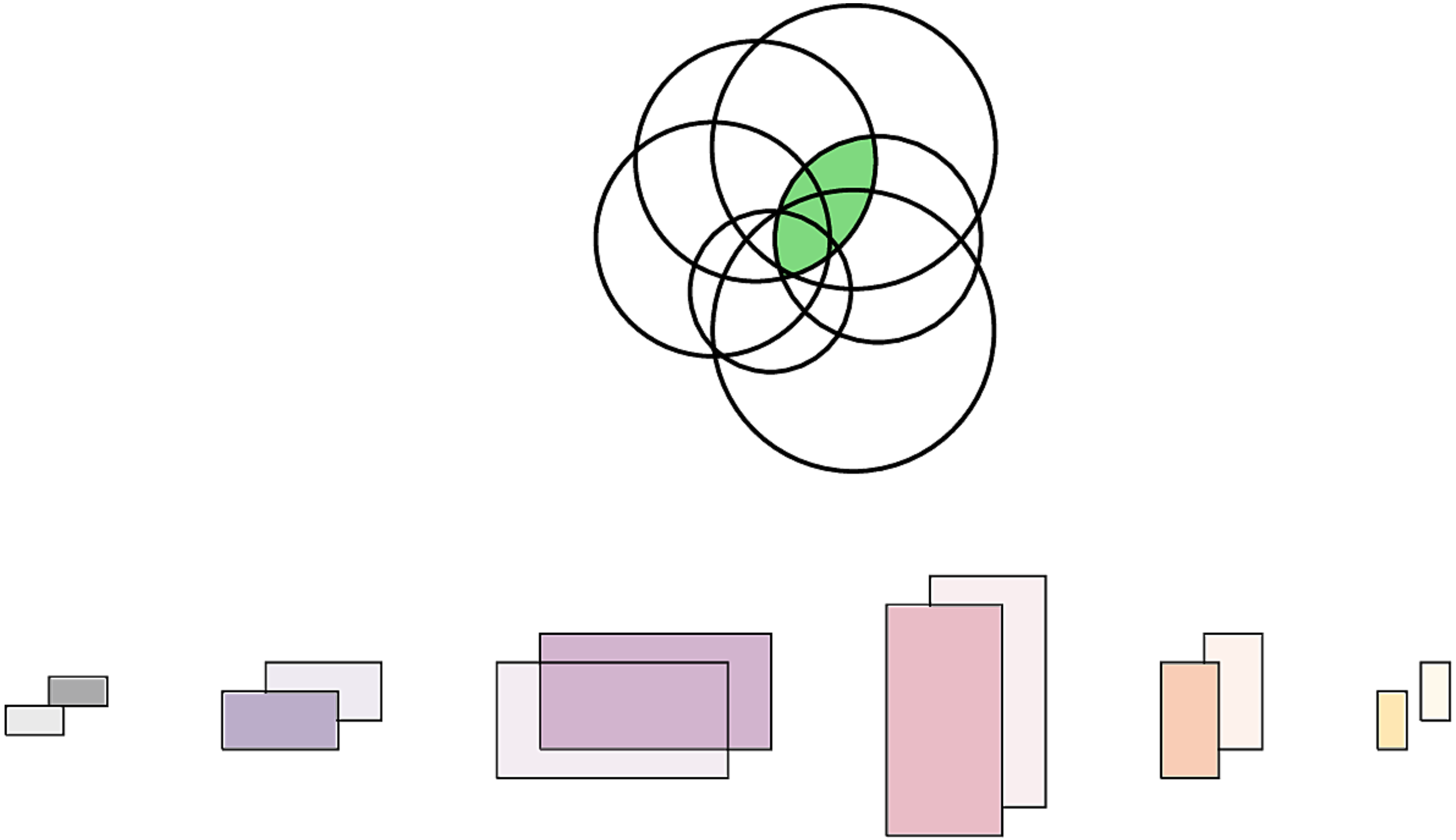


# Sobol Tensor Trains

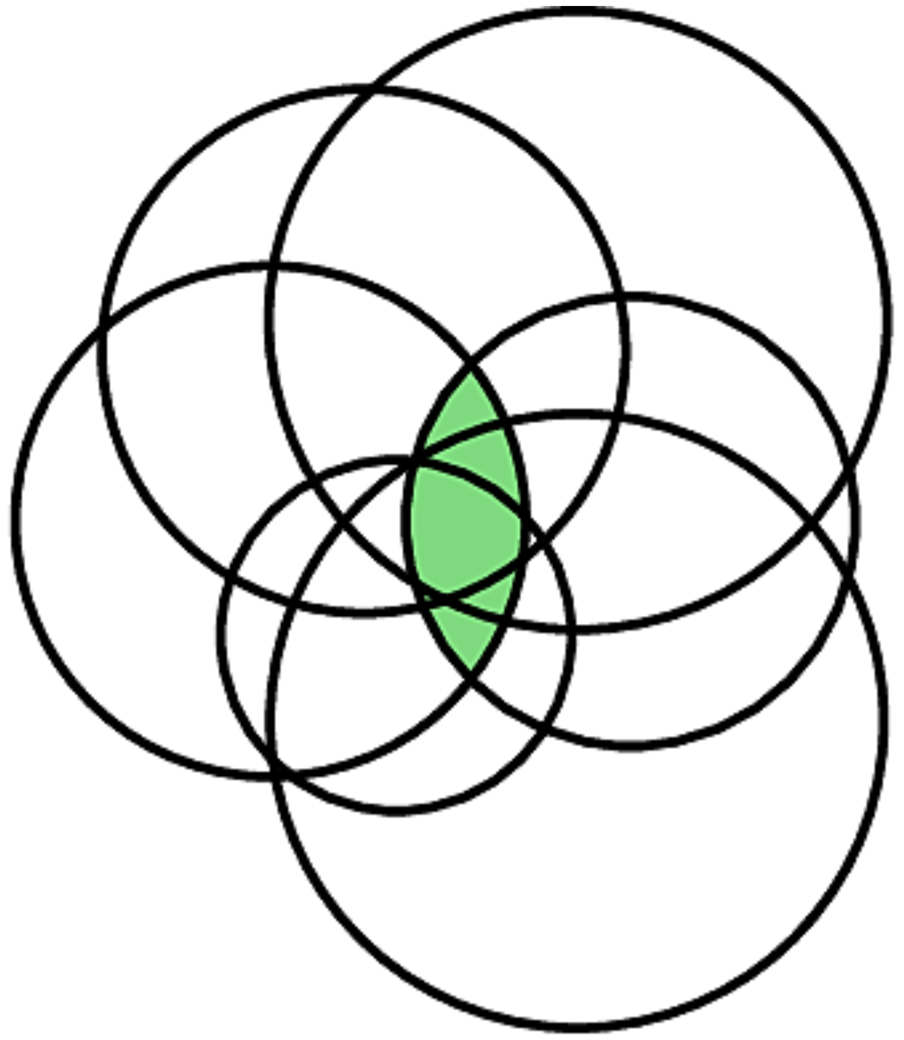


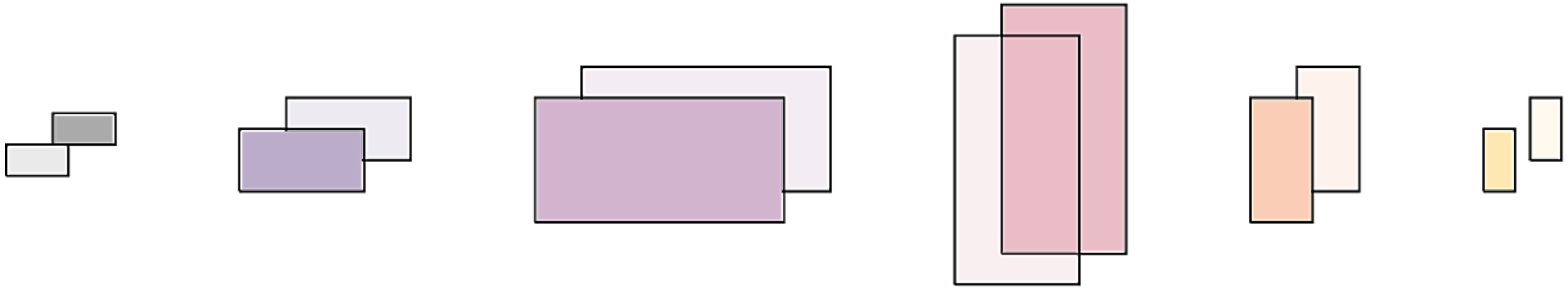


# Sobol Tensor Trains

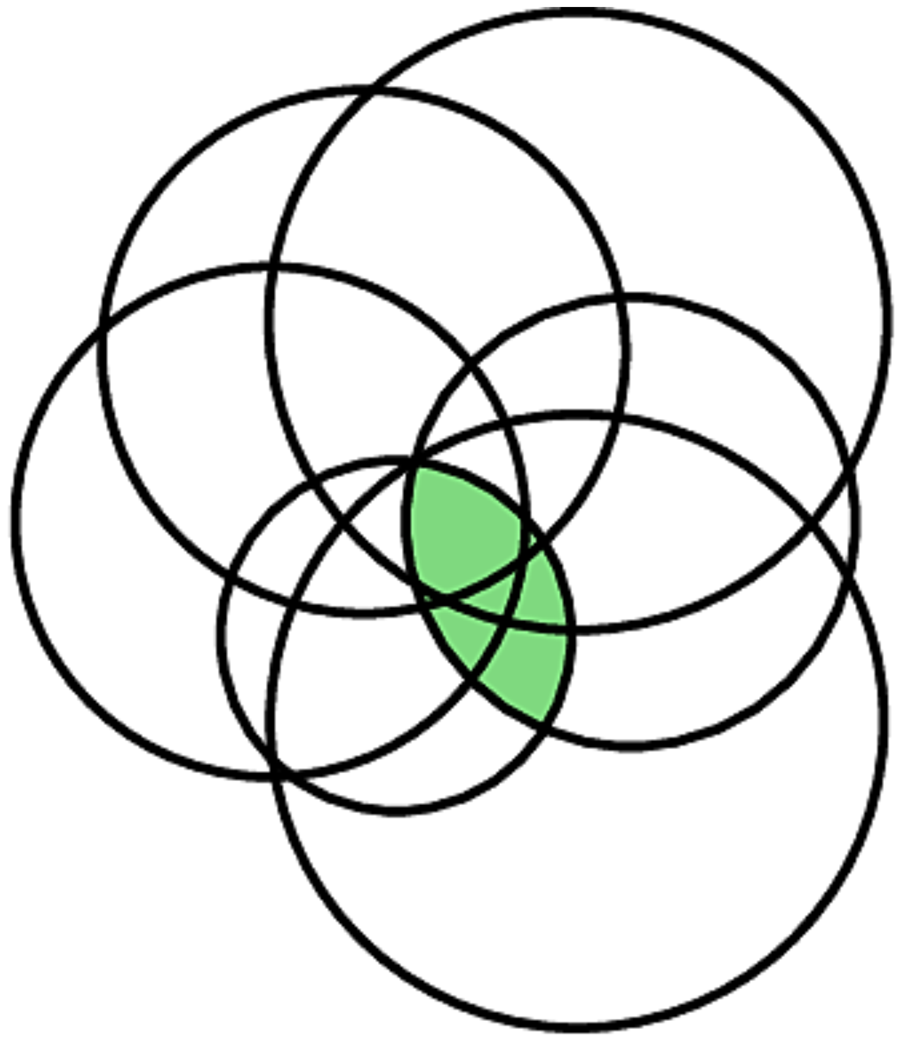


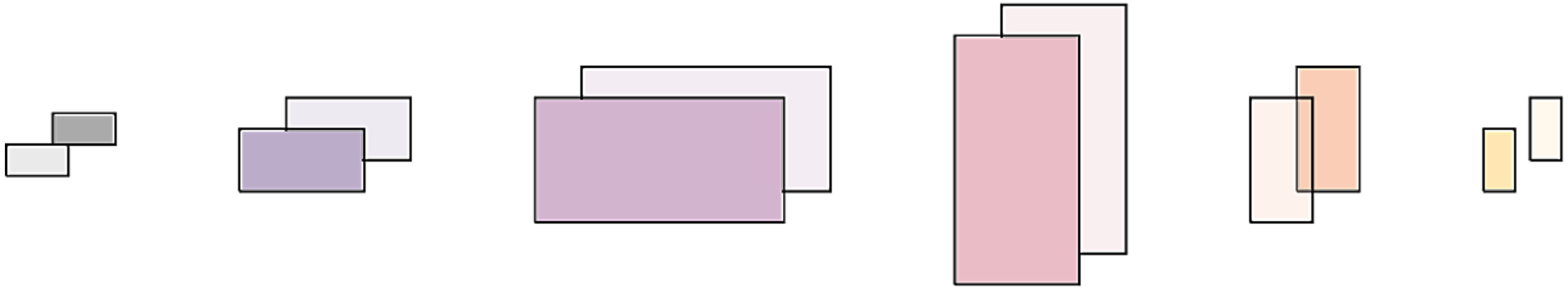
# Sobol Tensor Trains



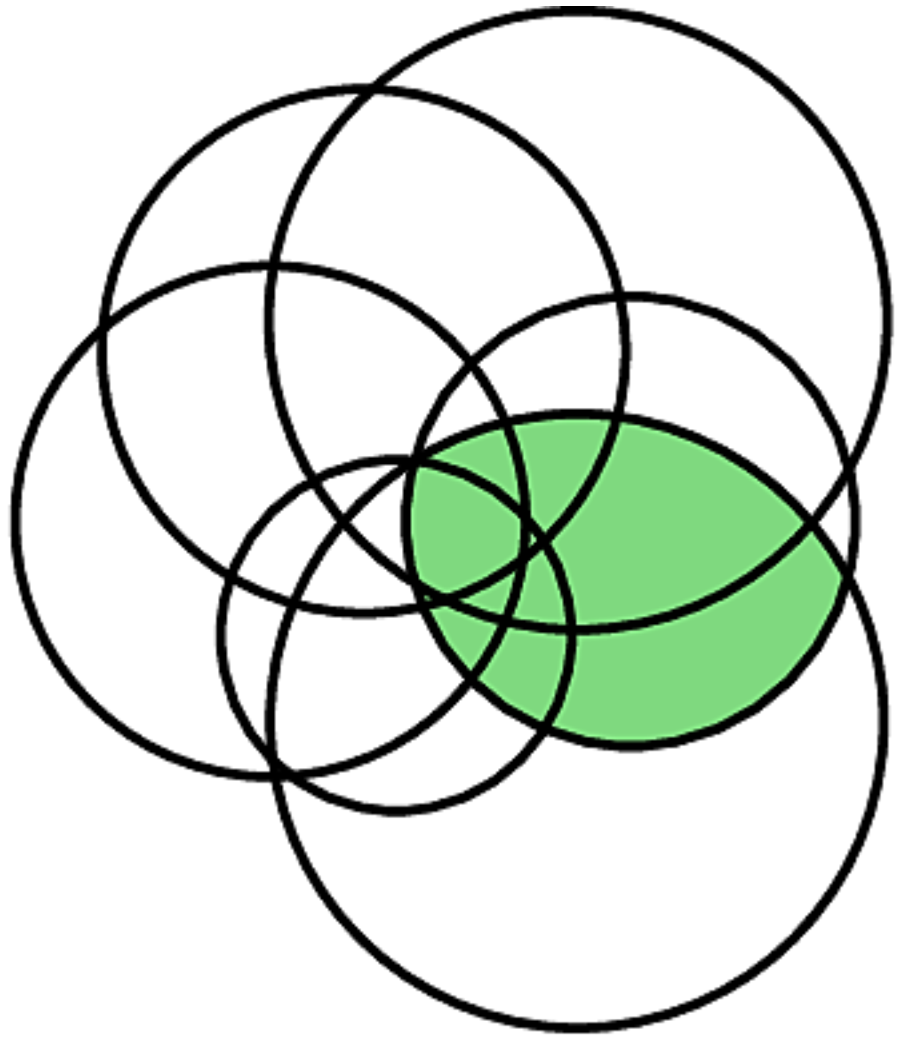


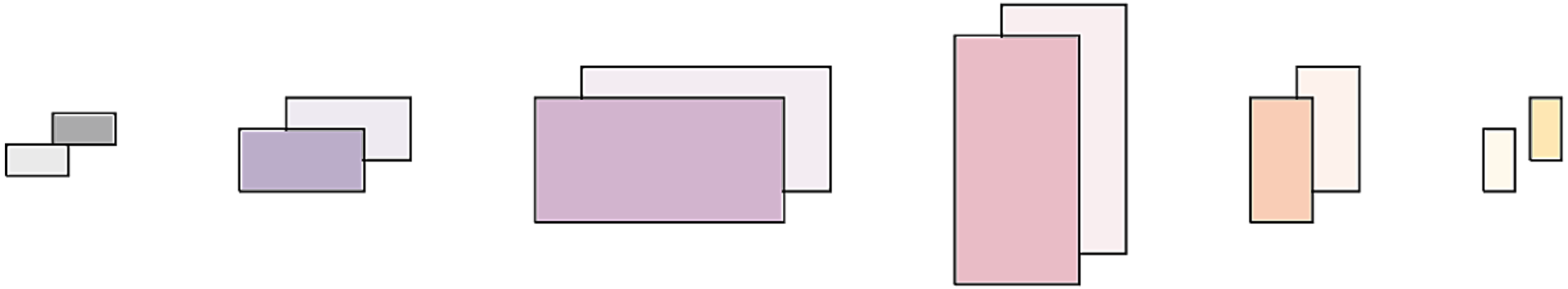
# Sobol Tensor Trains



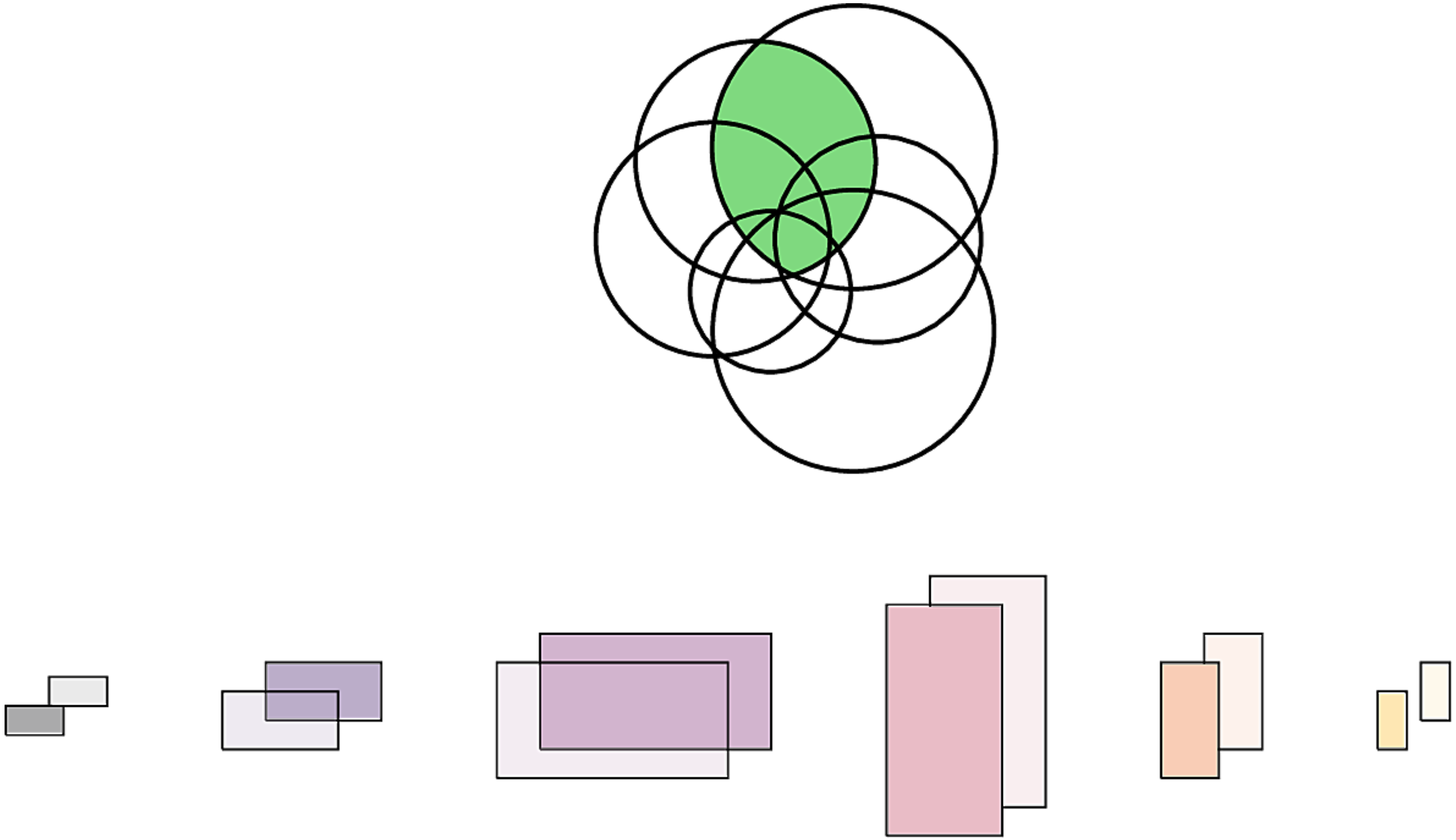


# Sobol Tensor Trains

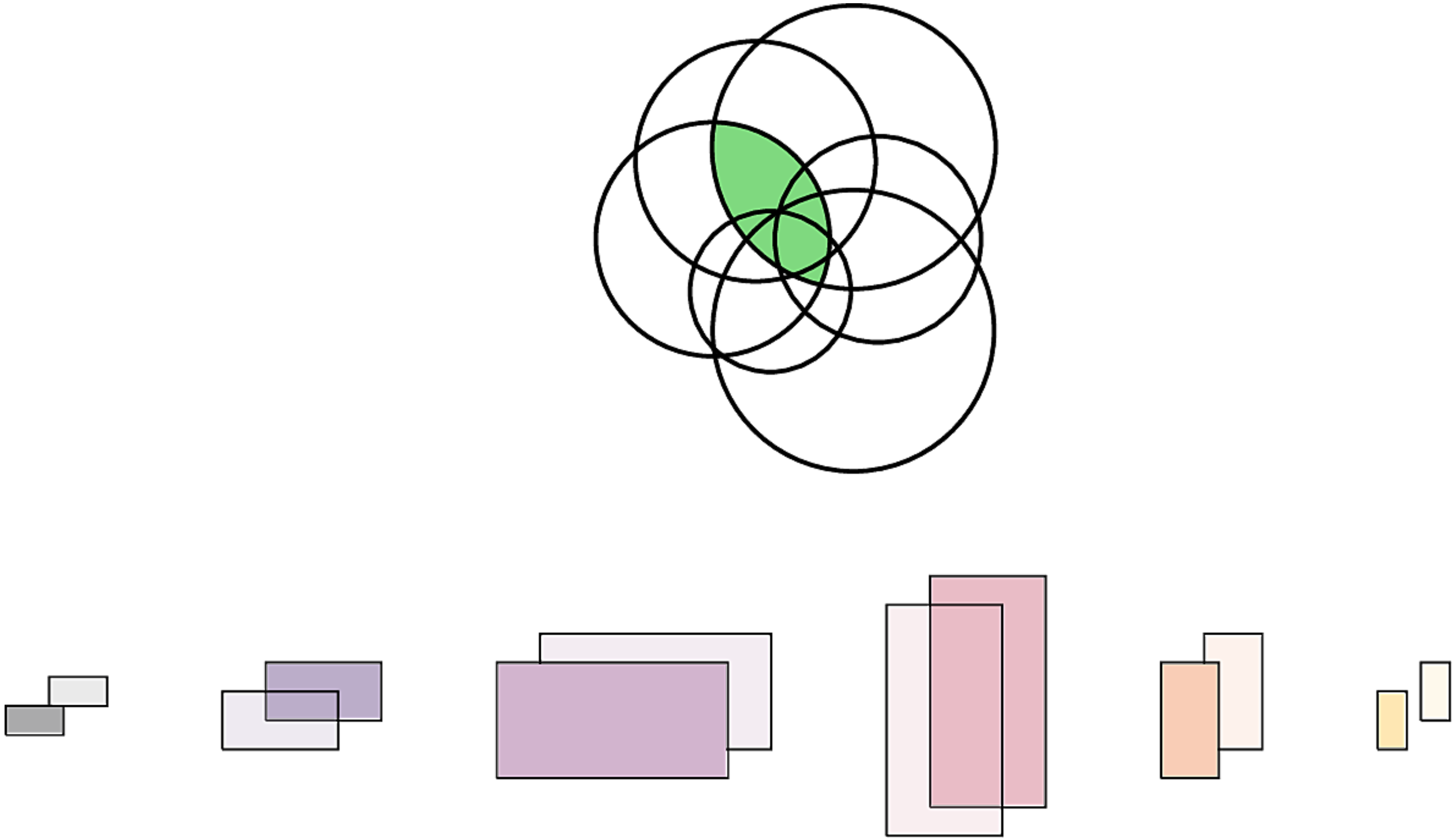




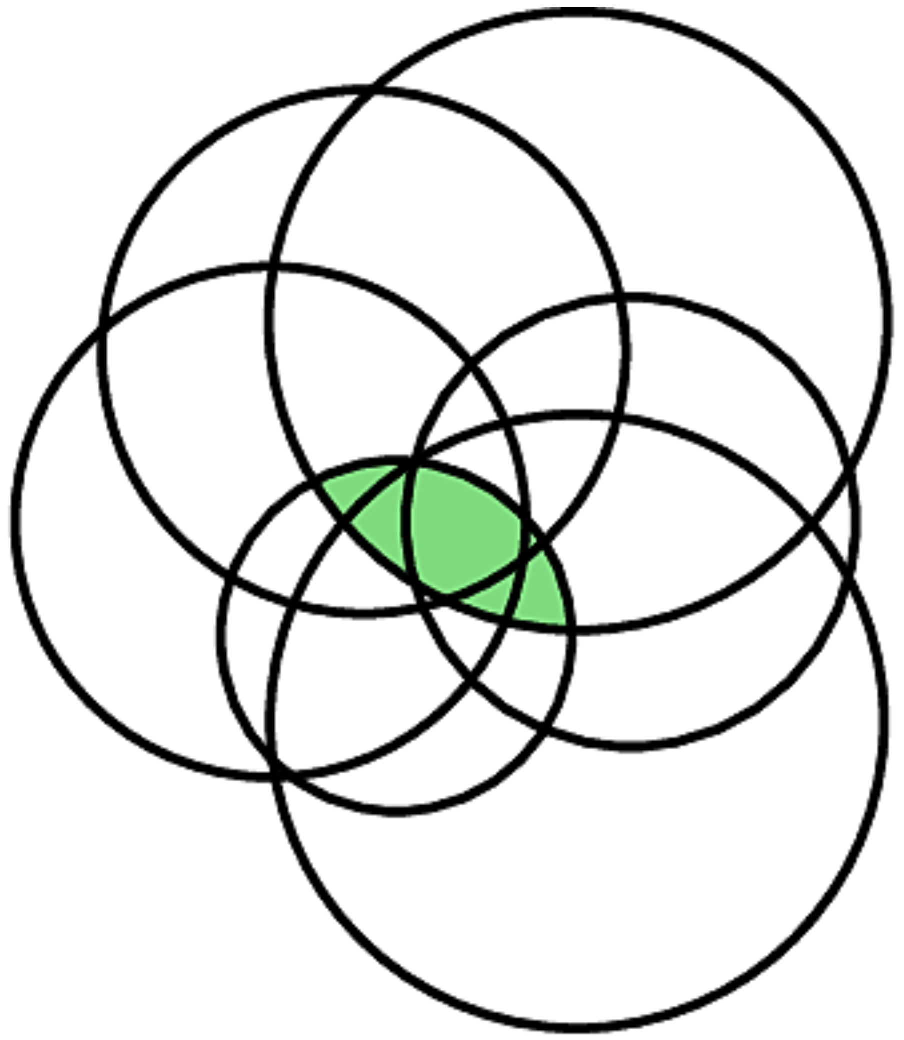
# Sobol Tensor Trains

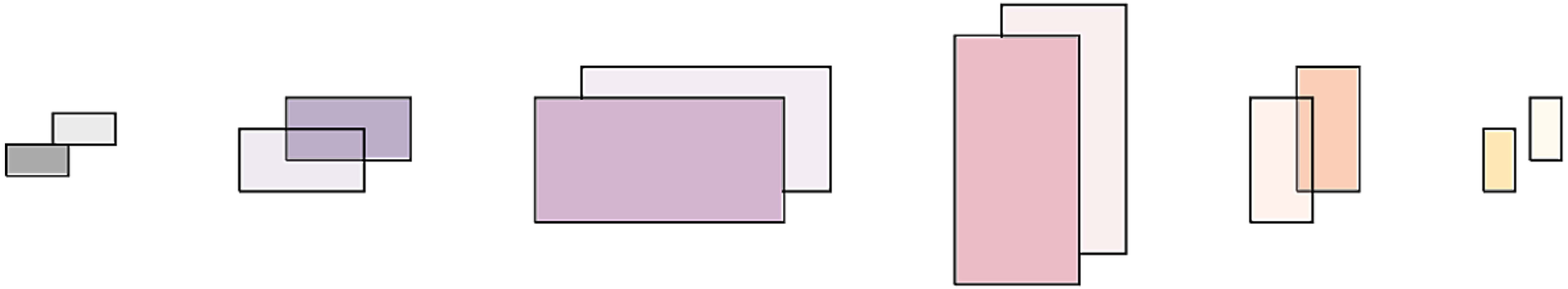


# Sobol Tensor Trains



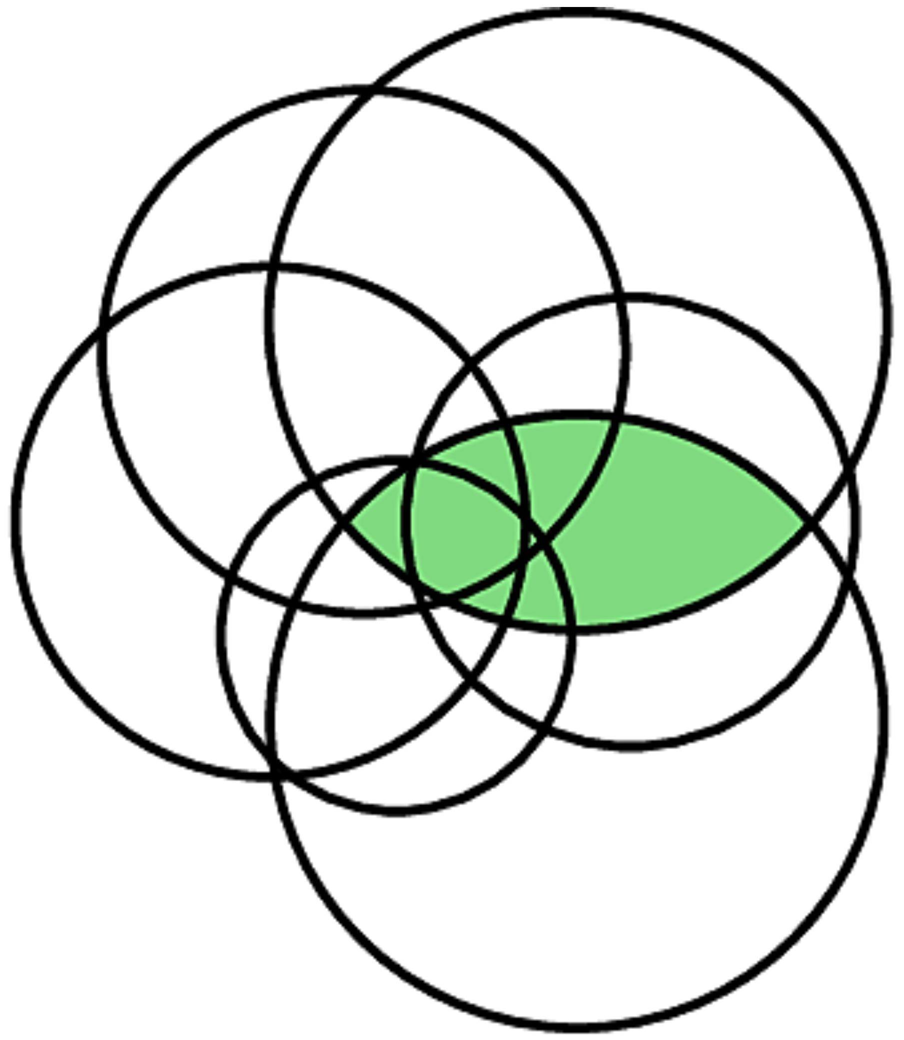
# Sobol Tensor Trains

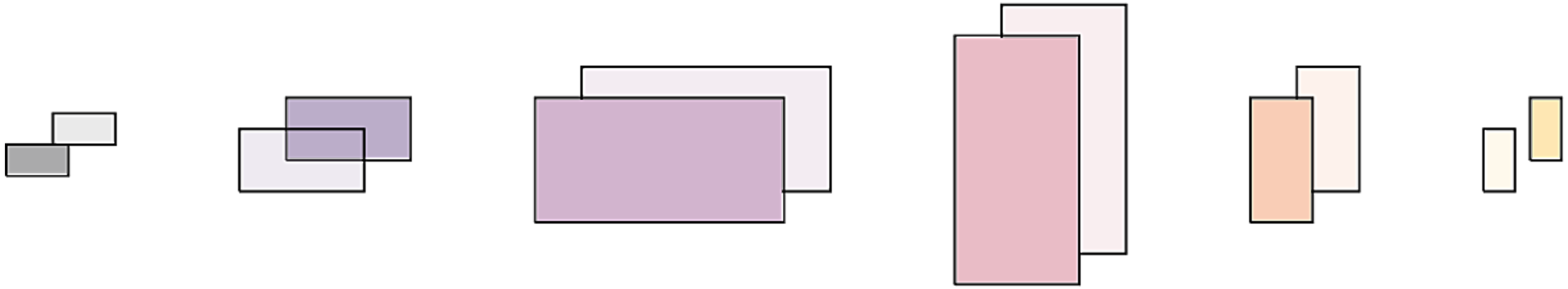




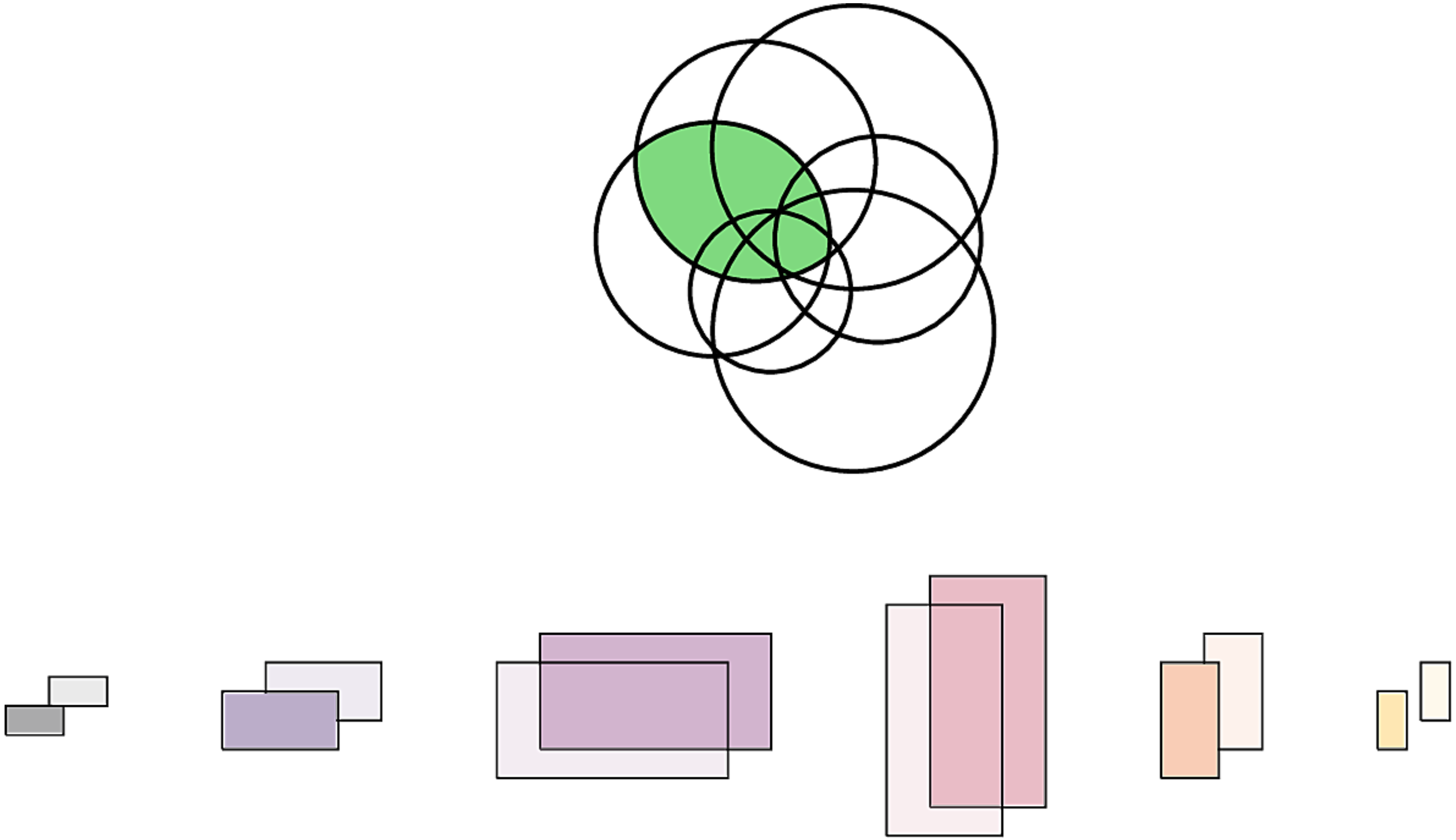
VISUALIZATIONAND MULTIMEDIALAB

# Sobol Tensor Trains

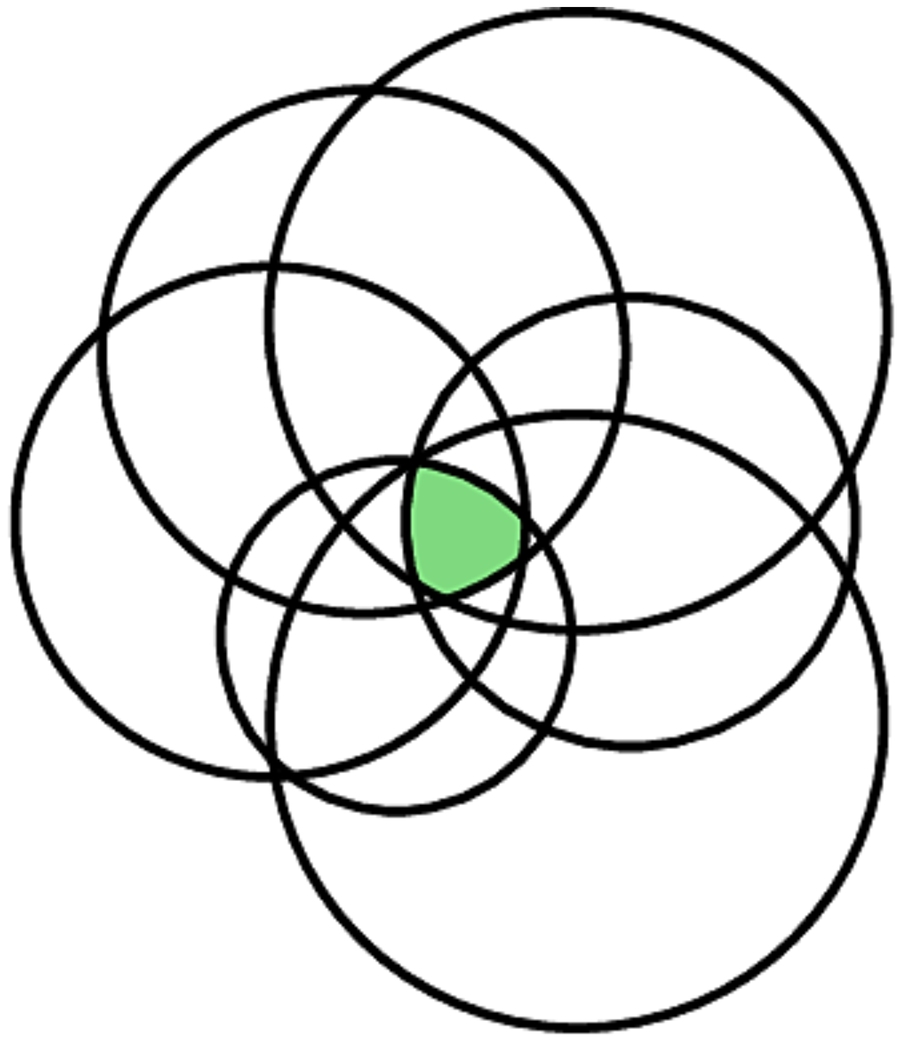


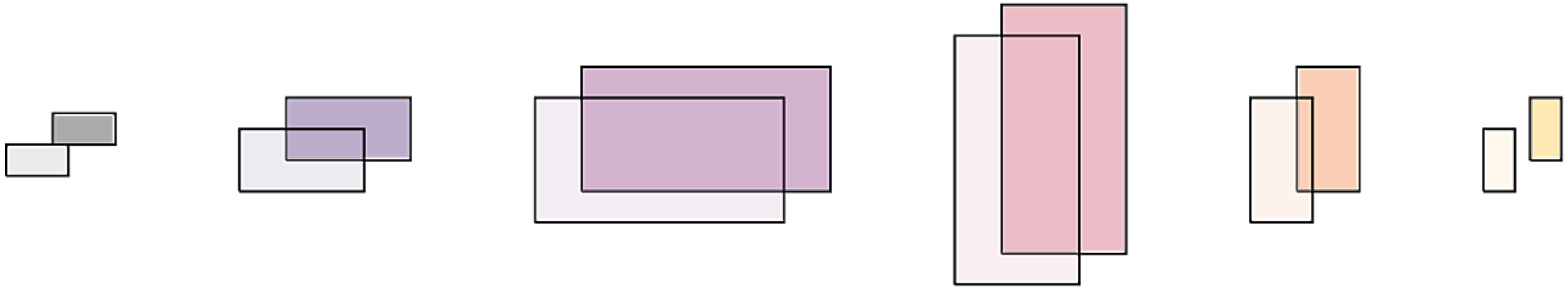


# Sobol Tensor Trains

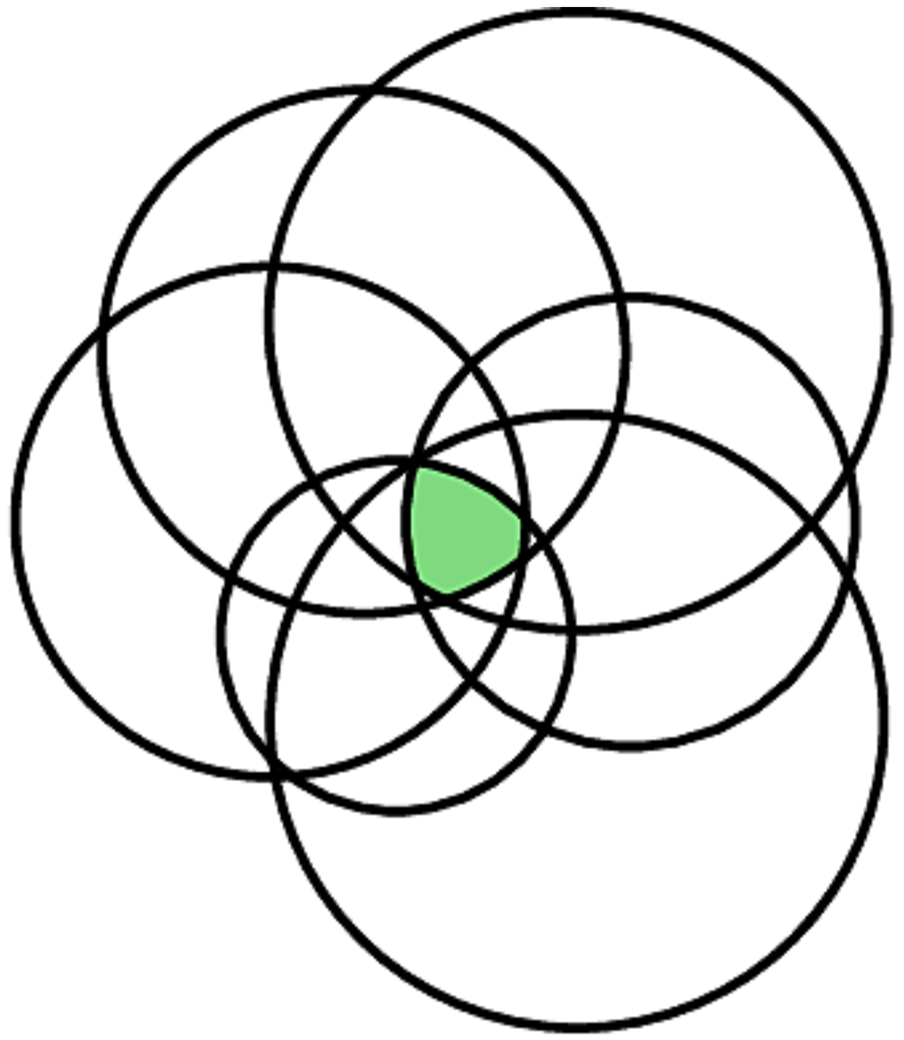


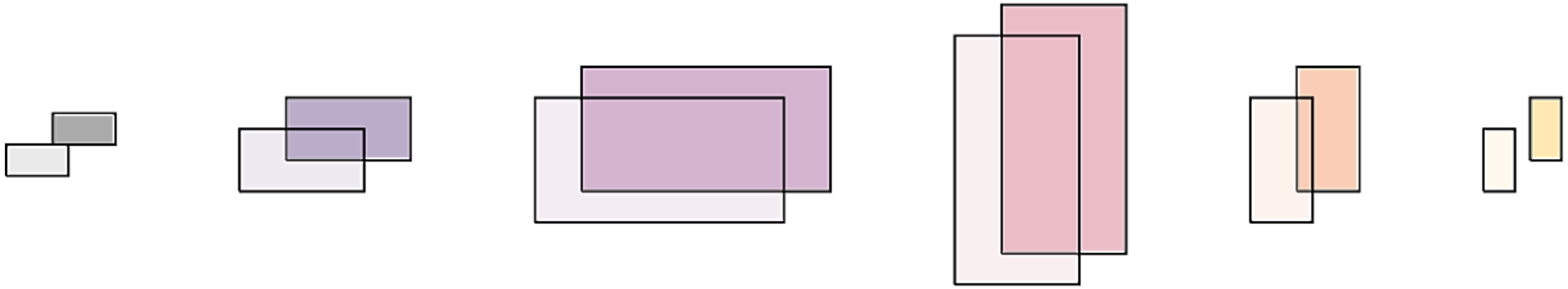
# Sobol Tensor Trains





# Sobol Tensor Trains





VISUALIZATIONAND
MULTIMEDIALAB