Self-supervised **deep learning** for **component separation** in the **submillimeter sky**

Victor Bonjean

with Nabila Aghanim, Marian Douspis, Tony Bonnaire, and Hideki Tanimura

ML4ASTRO2 - Catania - 11/07/2024

Funded by the European Union

What is component separation?

What is component separation?

$$X = S_0 + S_1 + S_2$$

What is component separation?

Observation

$$X = S_0 + S_1 + S_2$$

What is component separation?

Observation

$$X = S_0 + S_1 + S_2$$

3 components

What is component separation?

Multiple frequencies

Observation

$$X_i = a_i.S_0 + b_i.S_1 + c_i.S_2$$

3 components

What is component separation?

$$\begin{Bmatrix} X_0 \\ X_1 \\ \ldots \\ X_n \end{Bmatrix} = \begin{Bmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ \ldots & \ldots & \ldots \\ a_n & b_n & c_n \end{Bmatrix} \cdot \; S_0 \, S_1 \, S_2$$

n frequencies

What is component separation?

nx3: « **Mixing matrix** »

$$\begin{Bmatrix} X_0 \\ X_1 \\ \dots \\ X_n \end{Bmatrix} = \begin{Bmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ \dots & \dots & \dots \\ a_n & b_n & c_n \end{Bmatrix} \cdot \quad S_0 \, S_1 \, S_2$$

n frequencies

What is component separation?

$$\mathbf{X} = \mathbf{A}.\mathbf{S}$$

What is component separation?

$$\mathbf{X} = \mathbf{A}.\mathbf{S}$$

1. We know **A**

« No problem »

$$\mathbf{S} = \mathbf{A}^{-1}.\mathbf{X}$$

- Never the case

What is component separation?

$$\mathbf{X} = \boxed{\mathbf{A}} . \mathbf{S} + \mathbf{N}$$

Blind Source Separation (BSS)

1. We know **A**

« No problem »

$\mathbf{S} = \mathbf{A}^{-1} . \mathbf{X}$

- Never the case

2. We don't know **A**

Problem

- FastICA
- GMCA
- Unsupervised Learning

$\longrightarrow$ Prior on S

What is component separation?

$$\mathbf{X} = \mathbf{A}.\mathbf{S} + \mathbf{N}$$

Blind Source Separation (BSS)

1. We know A

« No problem »

$$\mathbf{S} = \mathbf{A}^{-1}.\mathbf{X}$$

- Never the case

1.5. We know a bit of A

Problem

- ILC
- GMCA
- Self-supervised Learning
- Template based fitting

→ Prior on A

2. We don't know A

Problem

- FastICA
- GMCA
- Unsupervised Learning

→ Prior on S

What is component separation?

$$X = A.S + N$$

Blind Source Separation (BSS)

1. We know **A**

« No problem »

$S = A^{-1}.X$

- Never the case

1.5. We know a bit of **A**

Problem

- ILC
- GMCA
- **Self-supervised Learning**
- Template based fitting

→ Prior on A

2. We don't know **A**

Problem

- FastICA
- GMCA
- **Unsupervised Learning**

→ Prior on S

What is component separation?

$$X = A . S + N$$

Blind Source Separation (BSS)

1. We know **A**

« No problem »

$$S = A^{-1} . X$$

- Never the case

1.5. We know a bit of **A**

Problem

- ILC
- GMCA
- **Self-supervised Learning** ✓
- Template based fitting

→ Prior on A

1.75 My work

2. We don't know **A**

Problem

- FastICA
- GMCA

→ Prior on S

- **Unsupervised Learning** ✓

Component separation with deep learning

Component separation with deep learning

Deep Image Prior (DIP) and « Double-DIP »

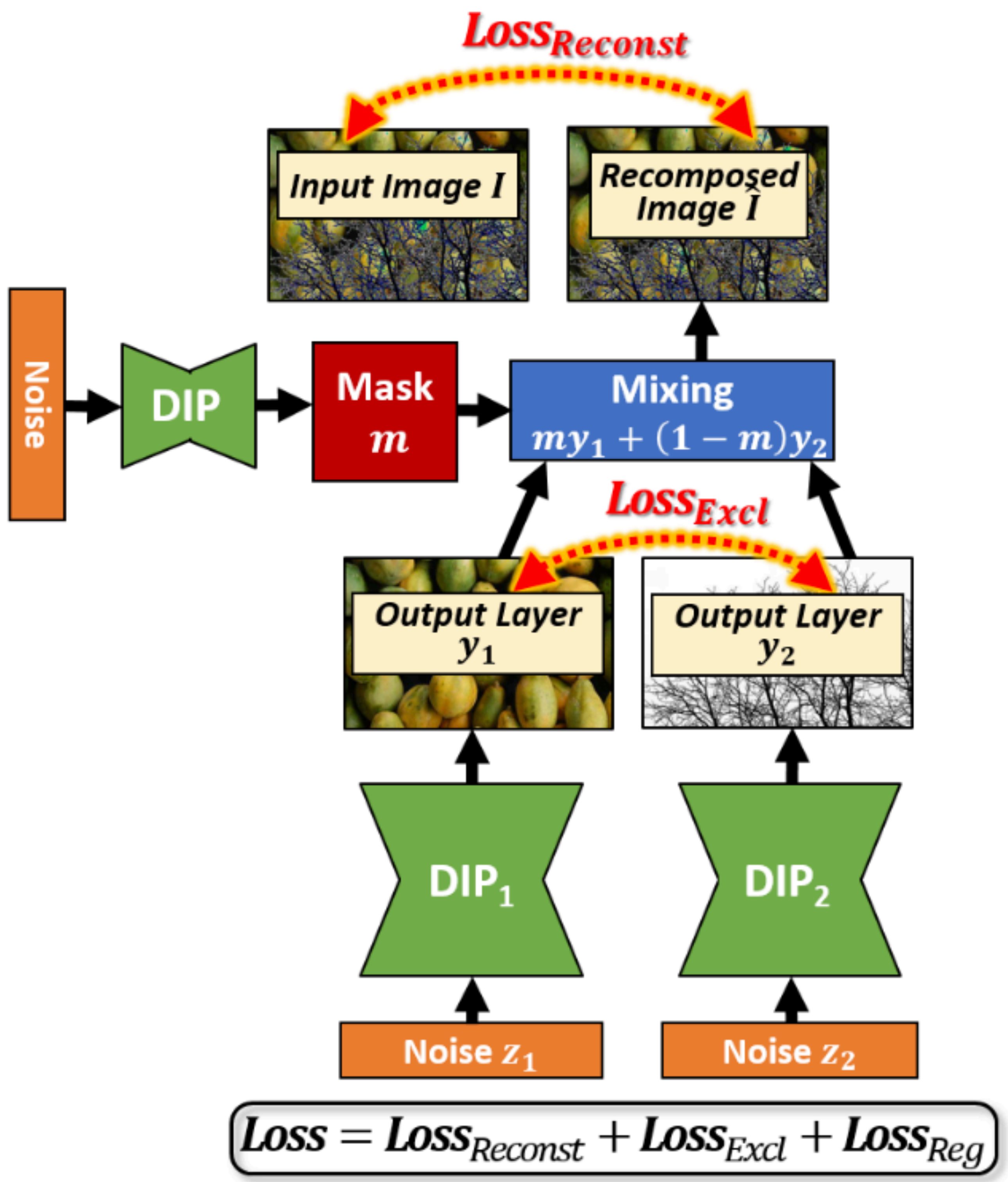


Figure 2: **Double-DIP Framework.** *Two Deep-Image-Prior networks (DIP_1 & DIP_2) jointly decompose an input image I into its layers (y_1 & y_2). Mixing those layers back according to a learned mask m, reconstructs an image $\hat{I} \approx I$.*

Gandelsman et al, 2018

Component separation with deep learning

Deep Image Prior (DIP) and « Double-DIP »

1) Start from noise

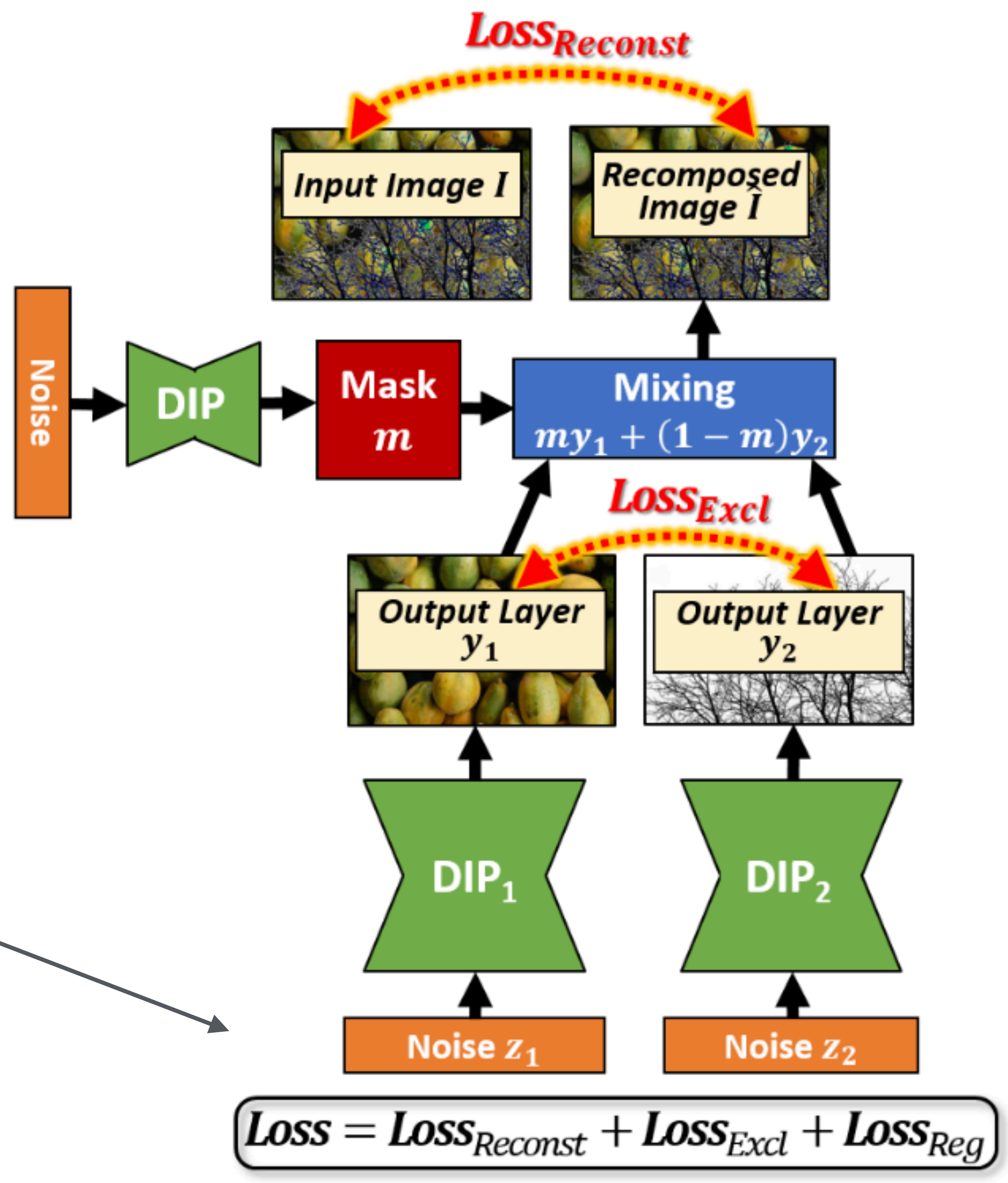


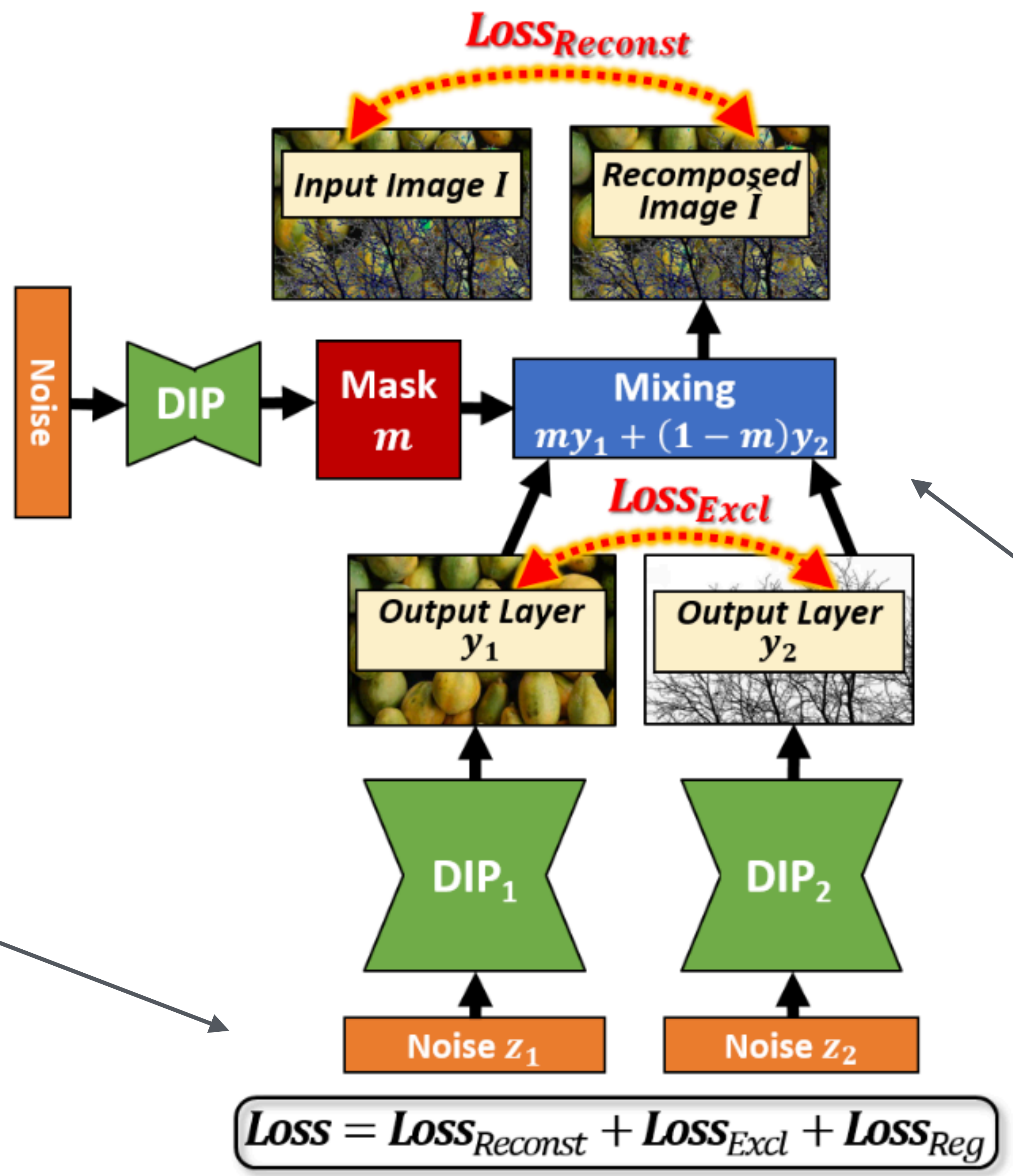
Figure 2: **Double-DIP Framework.** *Two Deep-Image-Prior networks (DIP_1 & DIP_2) jointly decompose an input image I into its layers (y_1 & y_2). Mixing those layers back according to a learned mask m, reconstructs an image $\hat{I} \approx I$.*

Gandelsman et al, 2018

Component separation with deep learning

Deep Image Prior (DIP) and « Double-DIP »

1) Start from noise



2) Apply priors on the mixing matrix

Figure 2: **Double-DIP Framework.** *Two Deep-Image-Prior networks (DIP_1 & DIP_2) jointly decompose an input image I into its layers (y_1 & y_2). Mixing those layers back according to a learned mask m, reconstructs an image $\hat{I} \approx I$.*

Gandelsman et al, 2018

Component separation with deep learning

Deep Image Prior (DIP) and « Double-DIP »

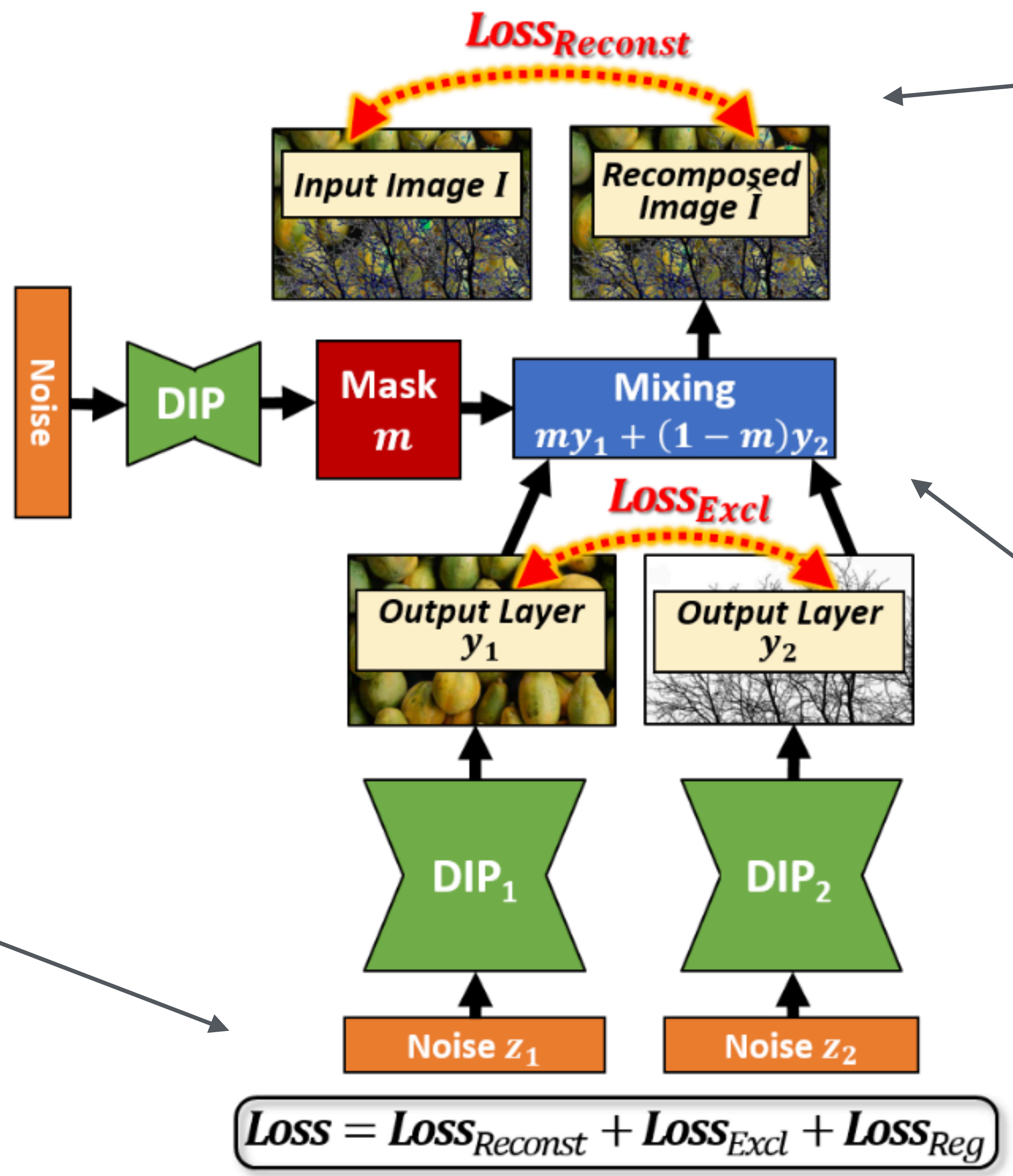


Figure 2: **Double-DIP Framework.** *Two Deep-Image-Prior networks (DIP_1 & DIP_2) jointly decompose an input image I into its layers (y_1 & y_2). Mixing those layers back according to a learned mask m, reconstructs an image $\hat{I} \approx I$.*

1) Start from noise

2) Apply priors on the mixing matrix

3) Add a reconstruction loss (MSE) between the recomposed image and the input image

Gandelsman et al, 2018

Component separation with deep learning

« Double-DIP »

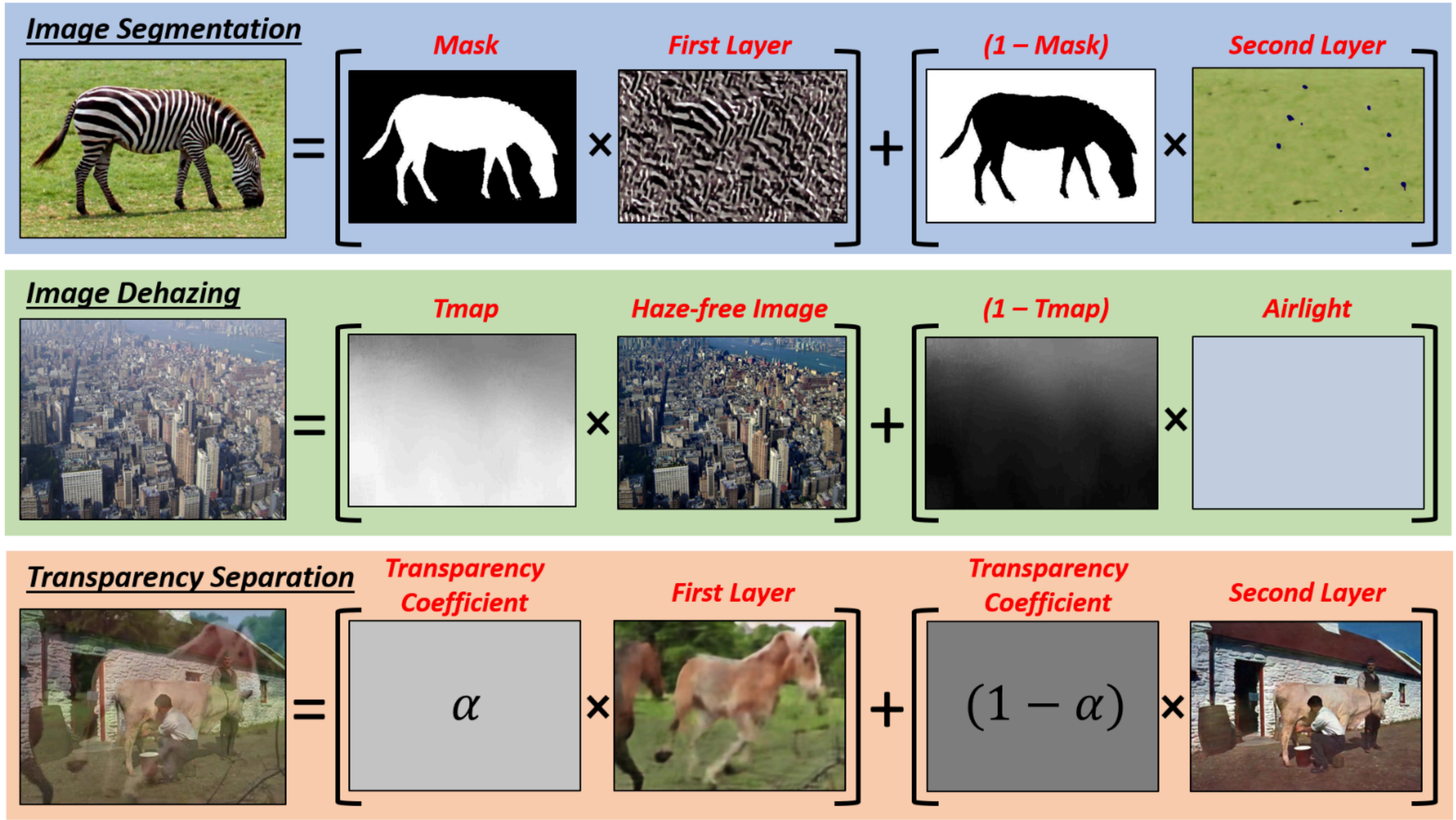
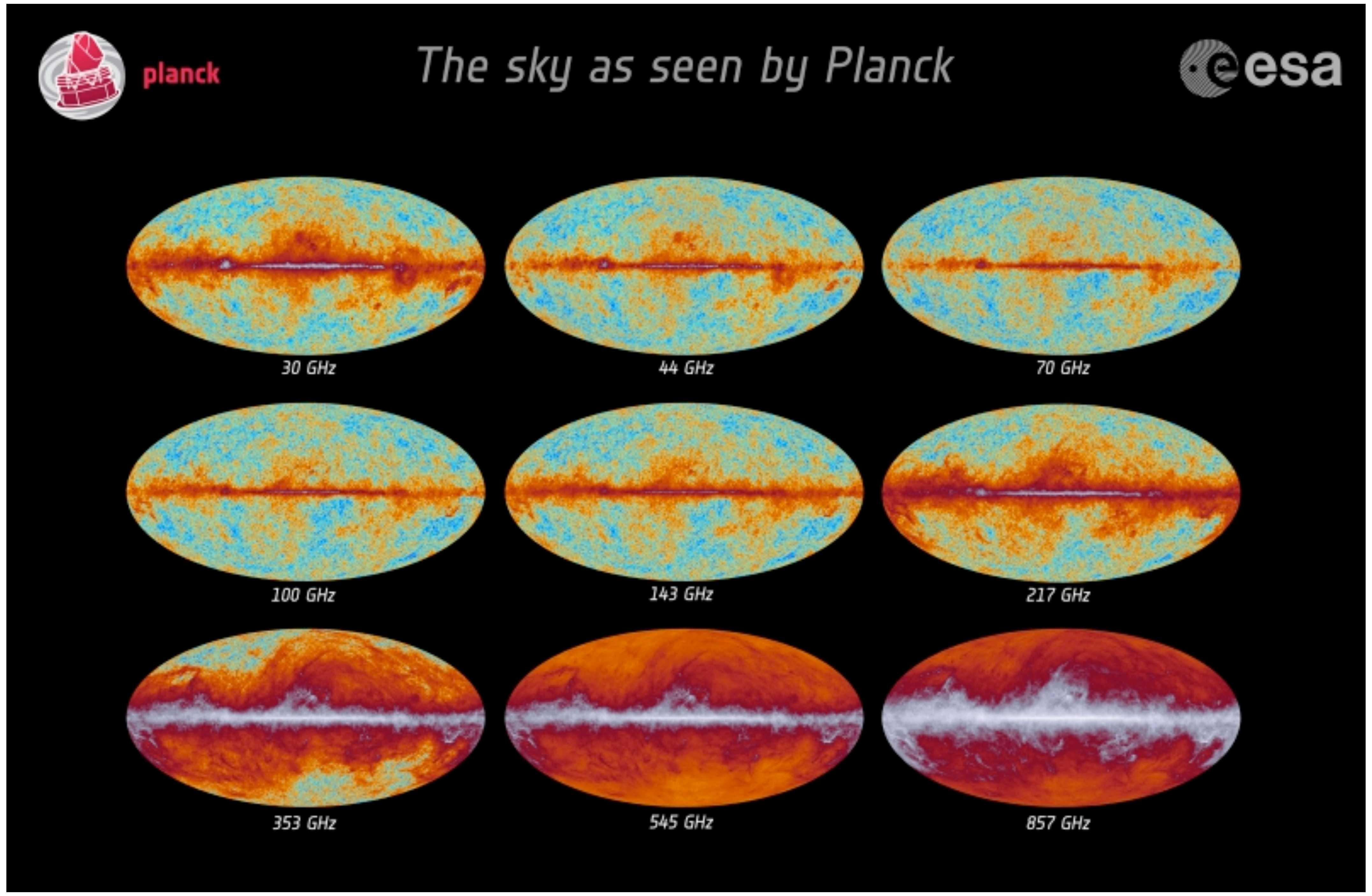


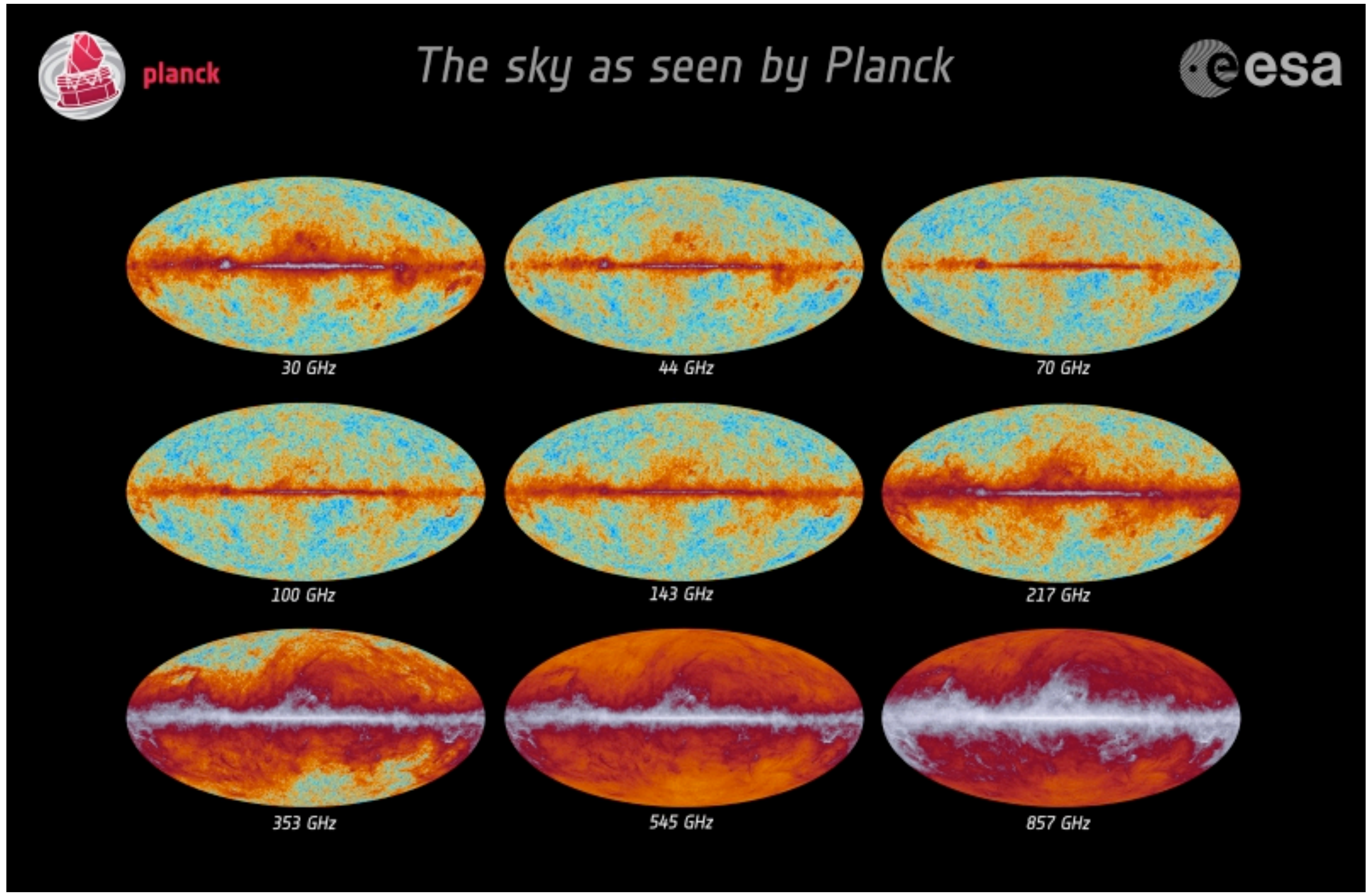
Figure 1: **A unified framework for image decomposition.** An image can be viewed as a mixture of "simpler" layers. Decomposing an image into such layers provides a unified framework for many seemingly unrelated vision tasks (e.g., segmentation, dehazing, transparency separation). Such a decomposition can be achieved using "Double-DIP".

Gandelsman et al, 2018

An application to CMB data - CIB removal - SZ extraction



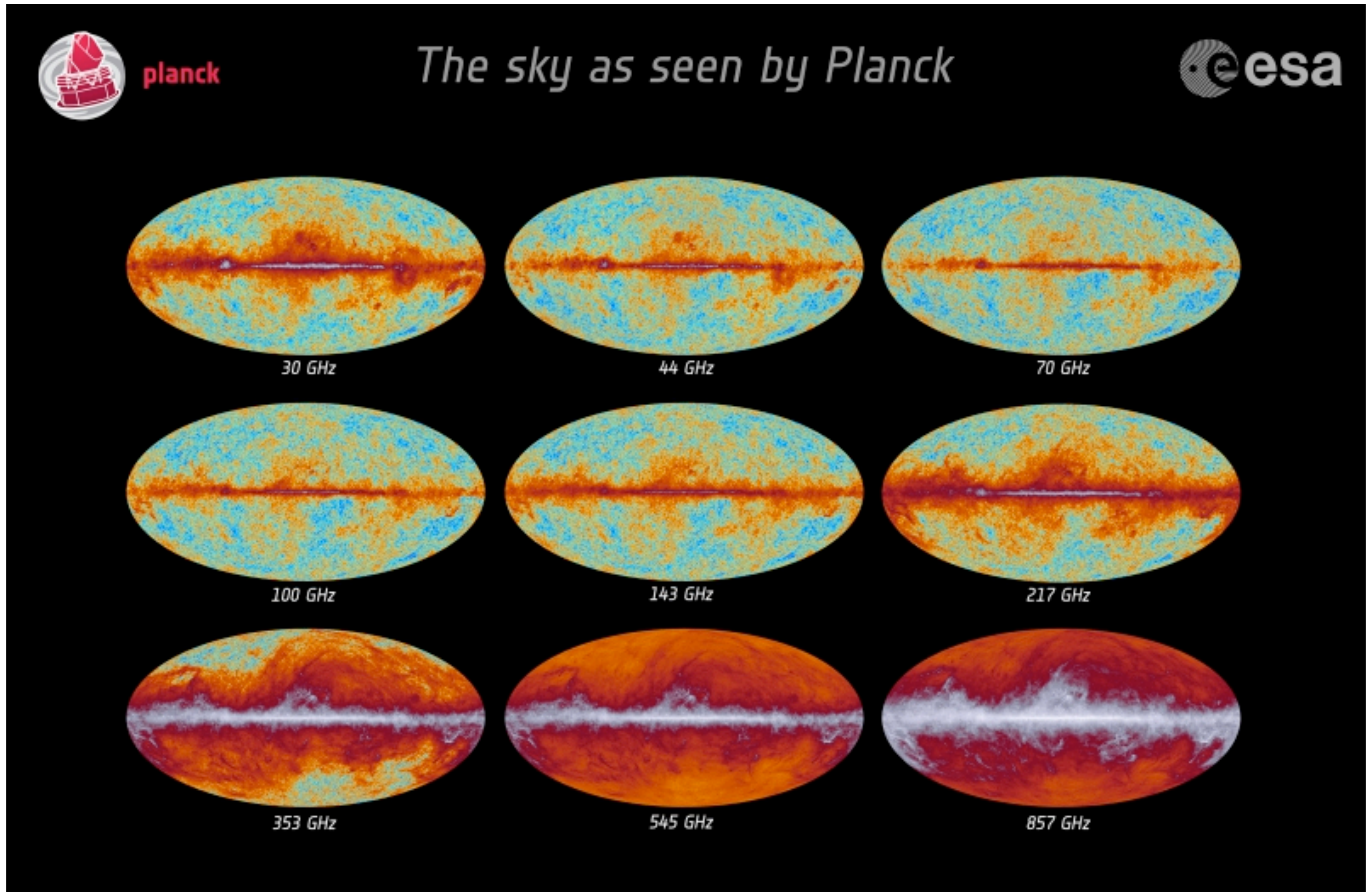
An application to CMB data - CIB removal - SZ extraction



↓

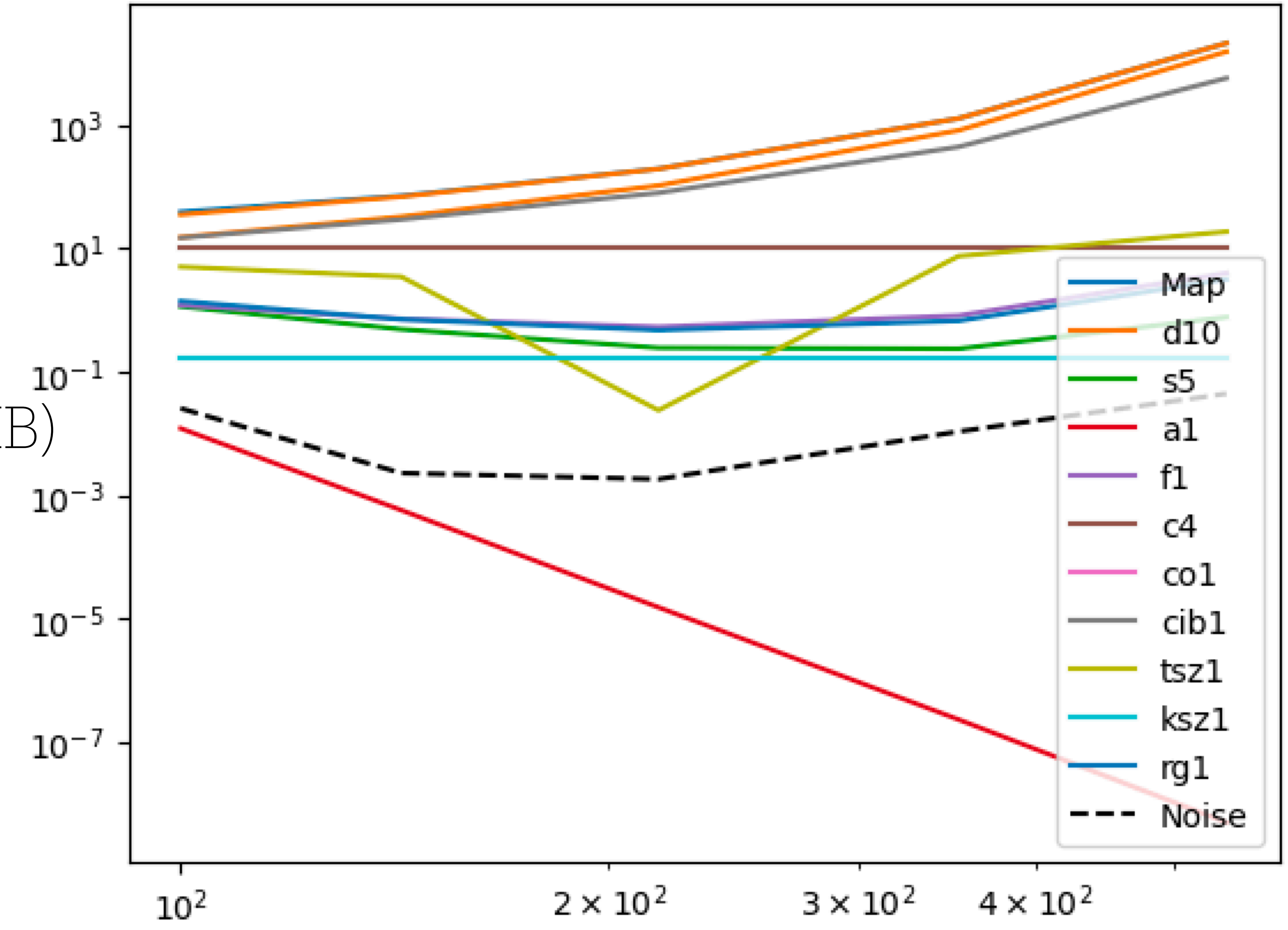
CMB, tSZ, kSZ, CIB, Radio sources, CO,
Free-free, Synchrotron and Galactic dust

An application to CMB data - CIB removal - SZ extraction



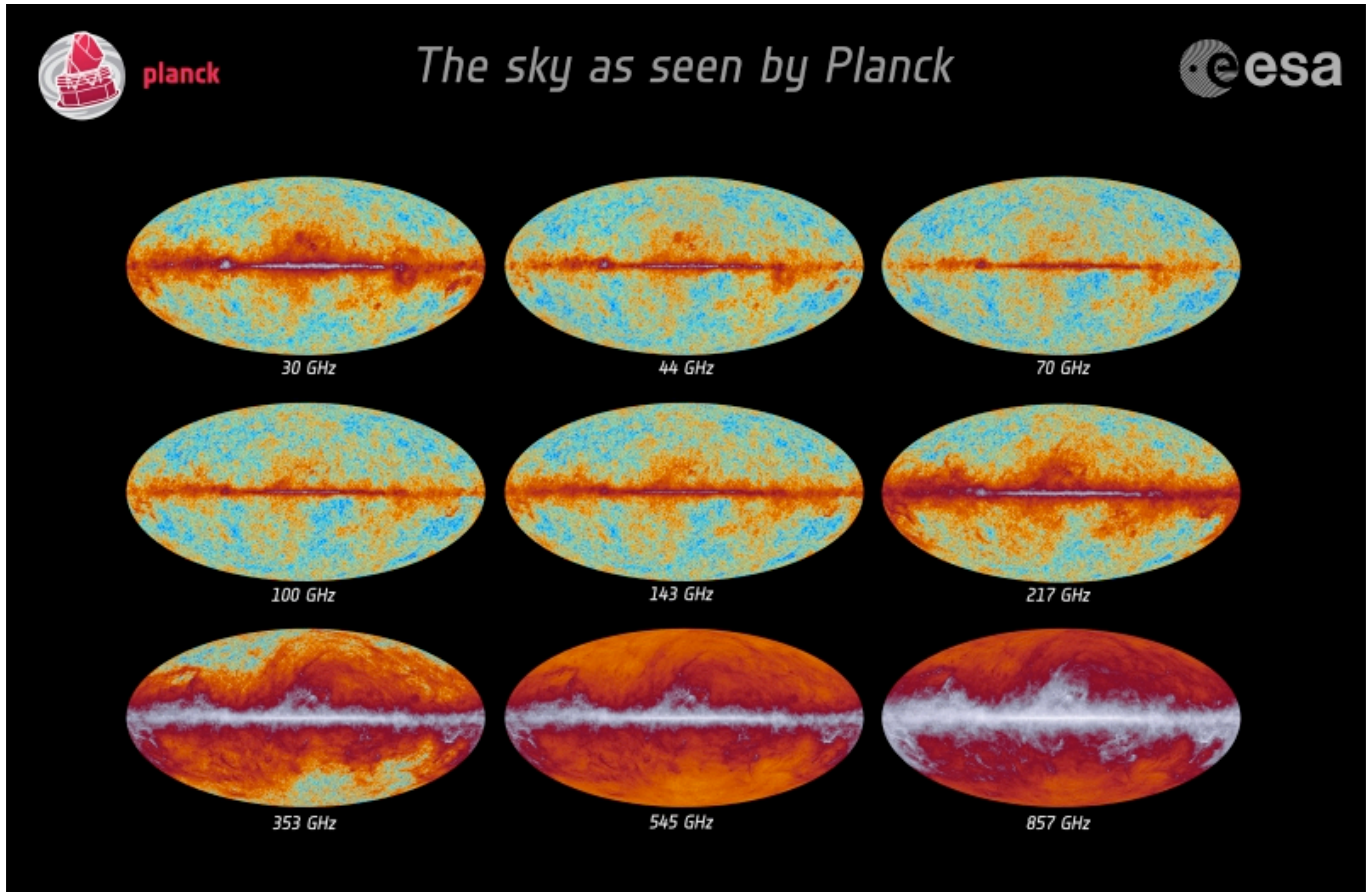
CMB, tSZ, kSZ, CIB, Radio sources, CO, Free-free, Synchrotron and Galactic dust

Means (uK_CMB)

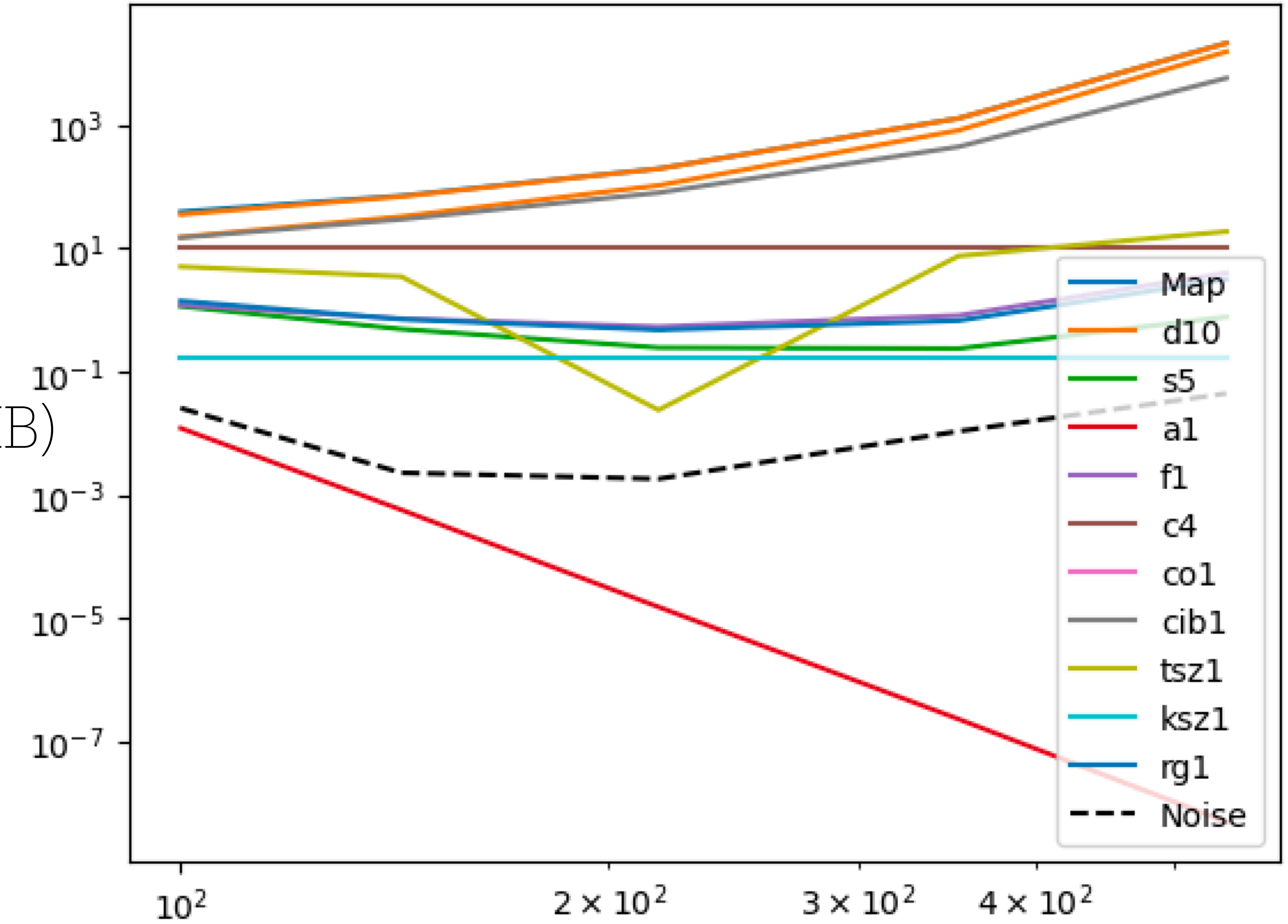


Frequencies

An application to CMB data - CIB removal - SZ extraction



Means (uK_CMB) →



Frequencies

↓

CMB, tSZ, kSZ, CIB, Radio sources, CO, Free-free, Synchrotron and Galactic dust

Non linear

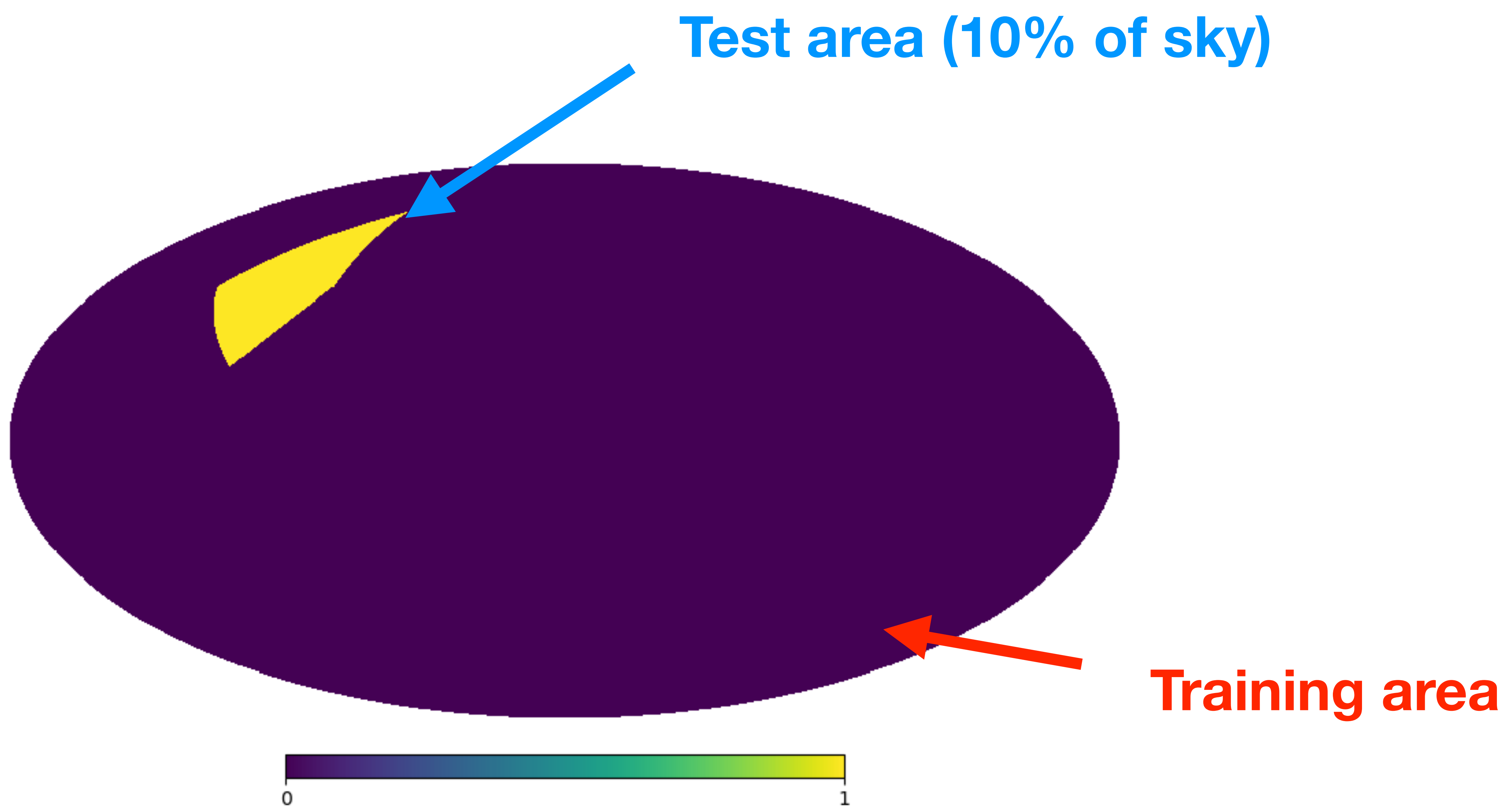
→ $\{C_i\}$ = Bi o (1*(CMB+kSZ) + fi*SZ + CIBi) + Ni

Non independent

Data

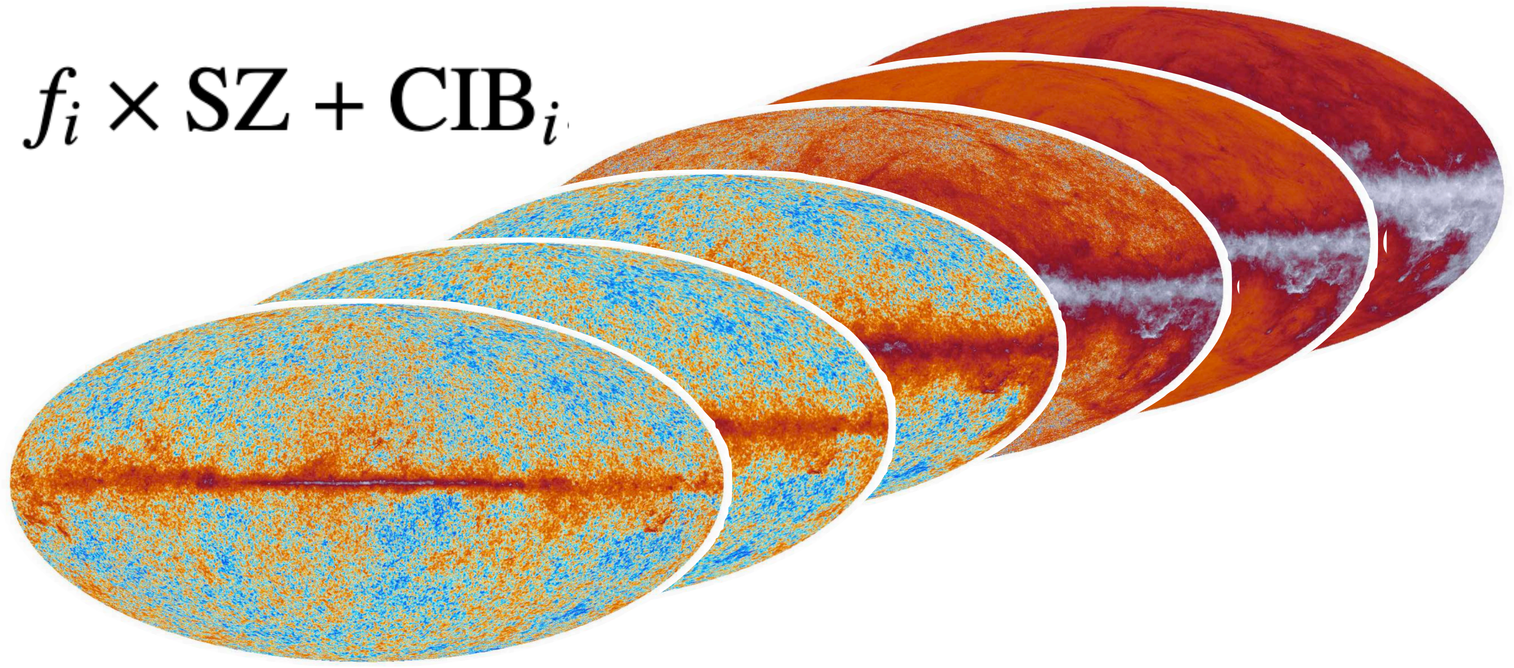
First study on **WebSky** numerical simulations (Stein et al, 2020):

- 90, 100, 143, 145, 217, 225, 280, 353, 545 GHz (Planck and SO)
- Healpix nside=4096
- CMB, CIB, SZ
- No noise, no beams



Method

$$C_i = 1 \times \mathrm{CMB} + f_i \times \mathrm{SZ} + \mathrm{CIB}_i$$

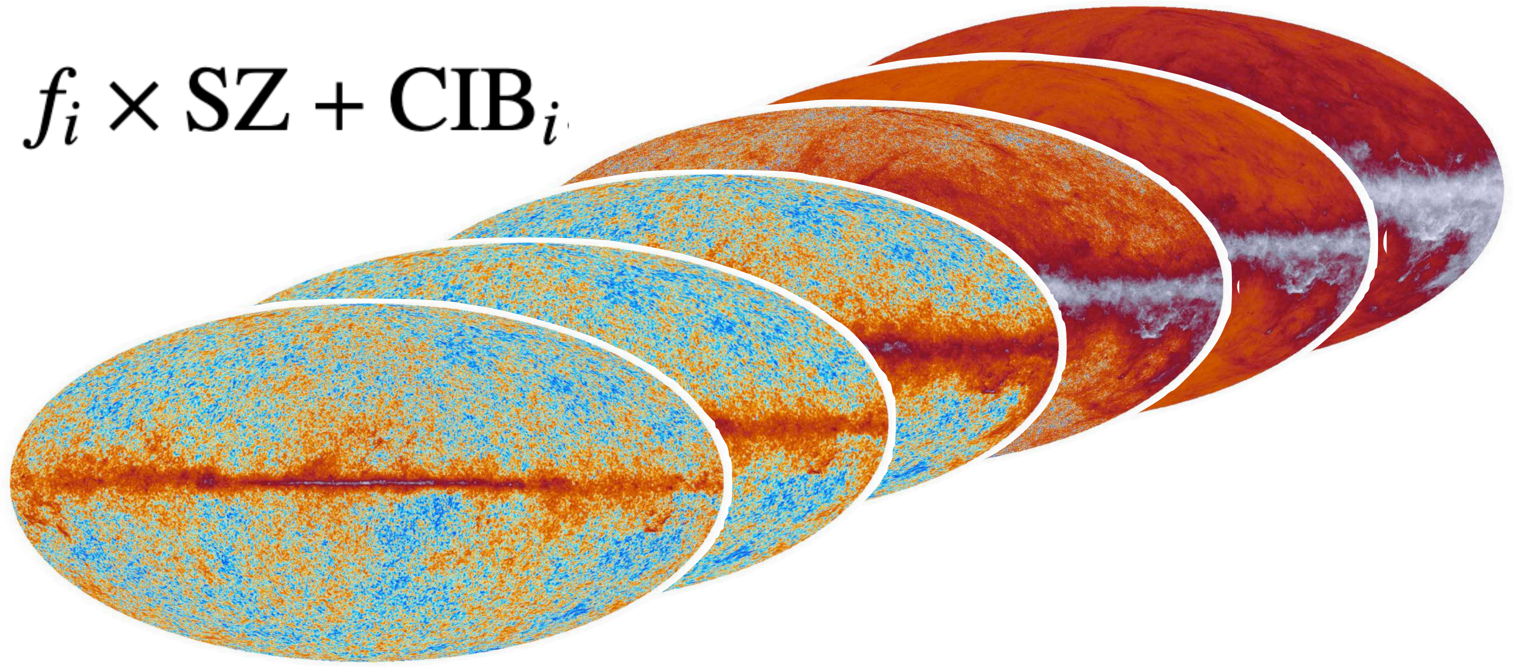


1) Project HEALPIX
nside=4096 in patches

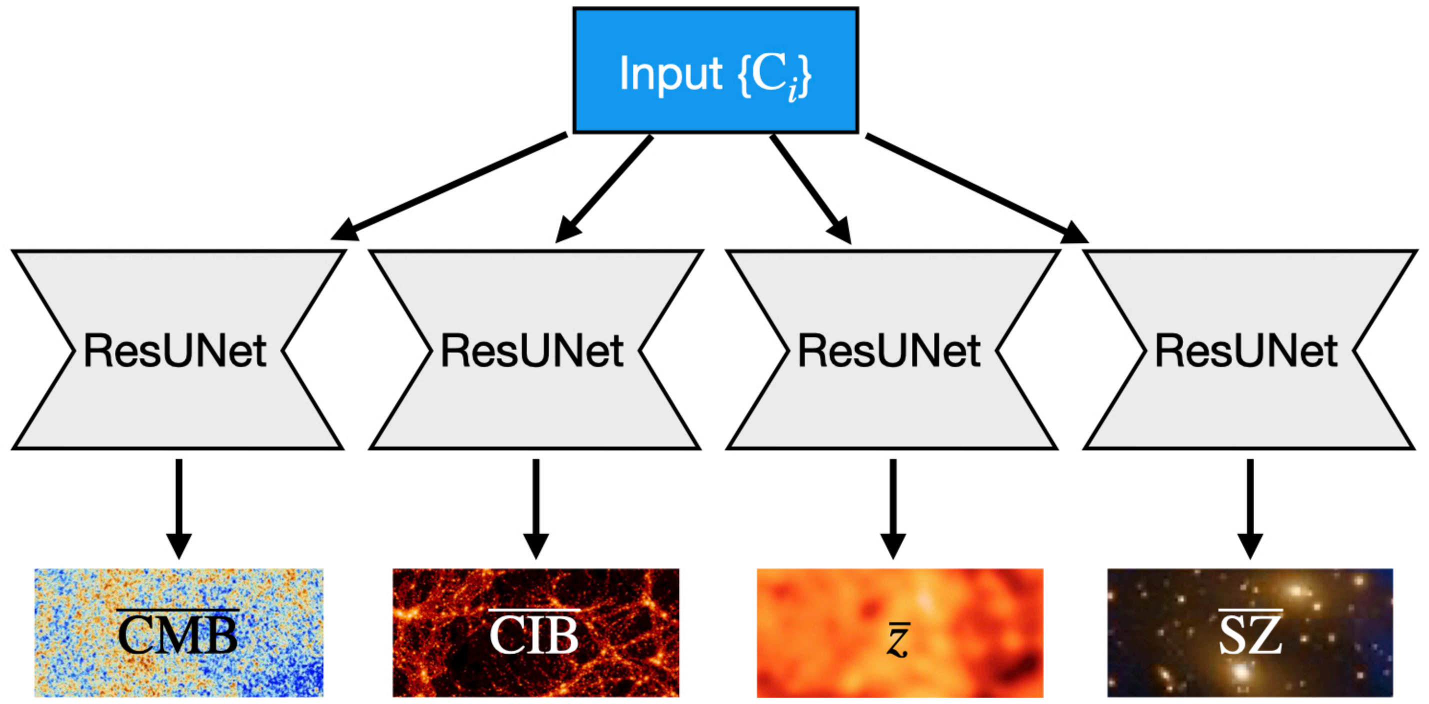
Bonjean et al, 2024

Method

$$C_i = 1 \times \mathrm{CMB} + f_i \times \mathrm{SZ} + \mathrm{CIB}_i$$



1) Project HEALPIX nside=4096 in patches

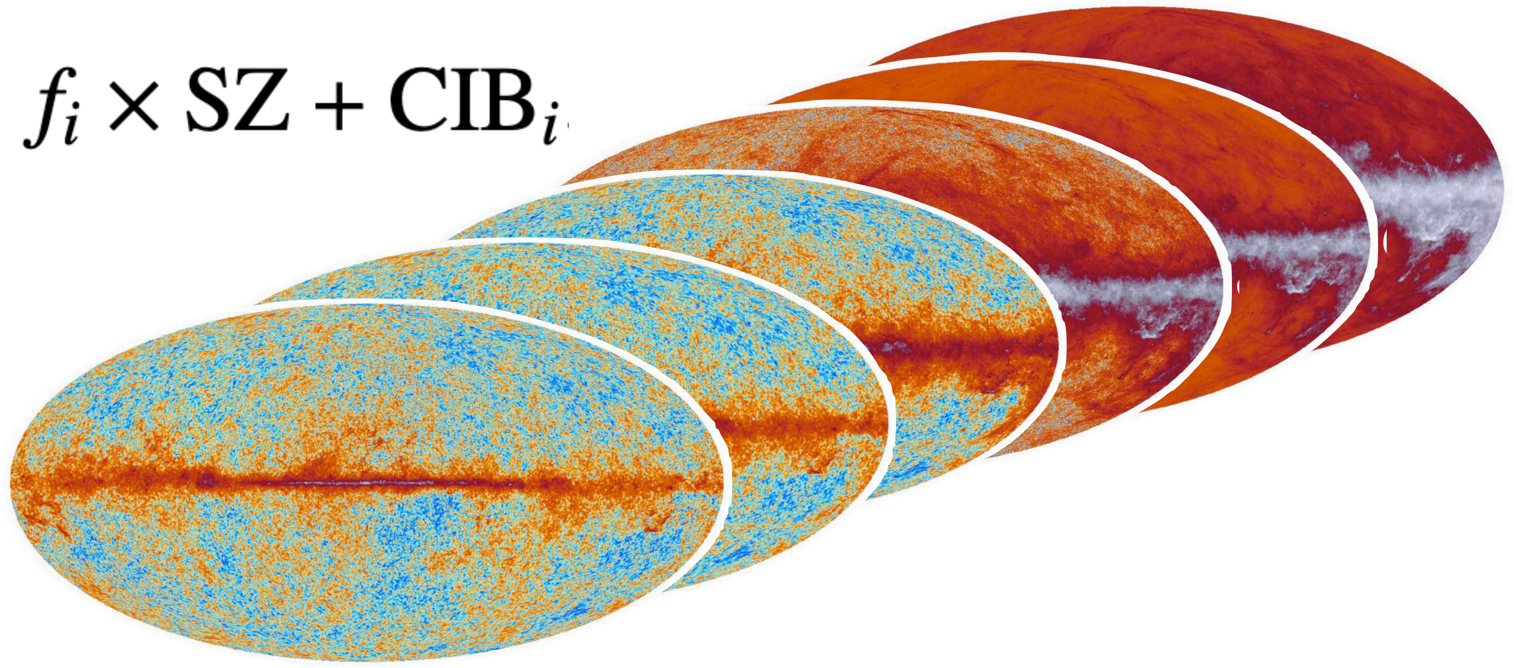


2) Network with n DIP for n components

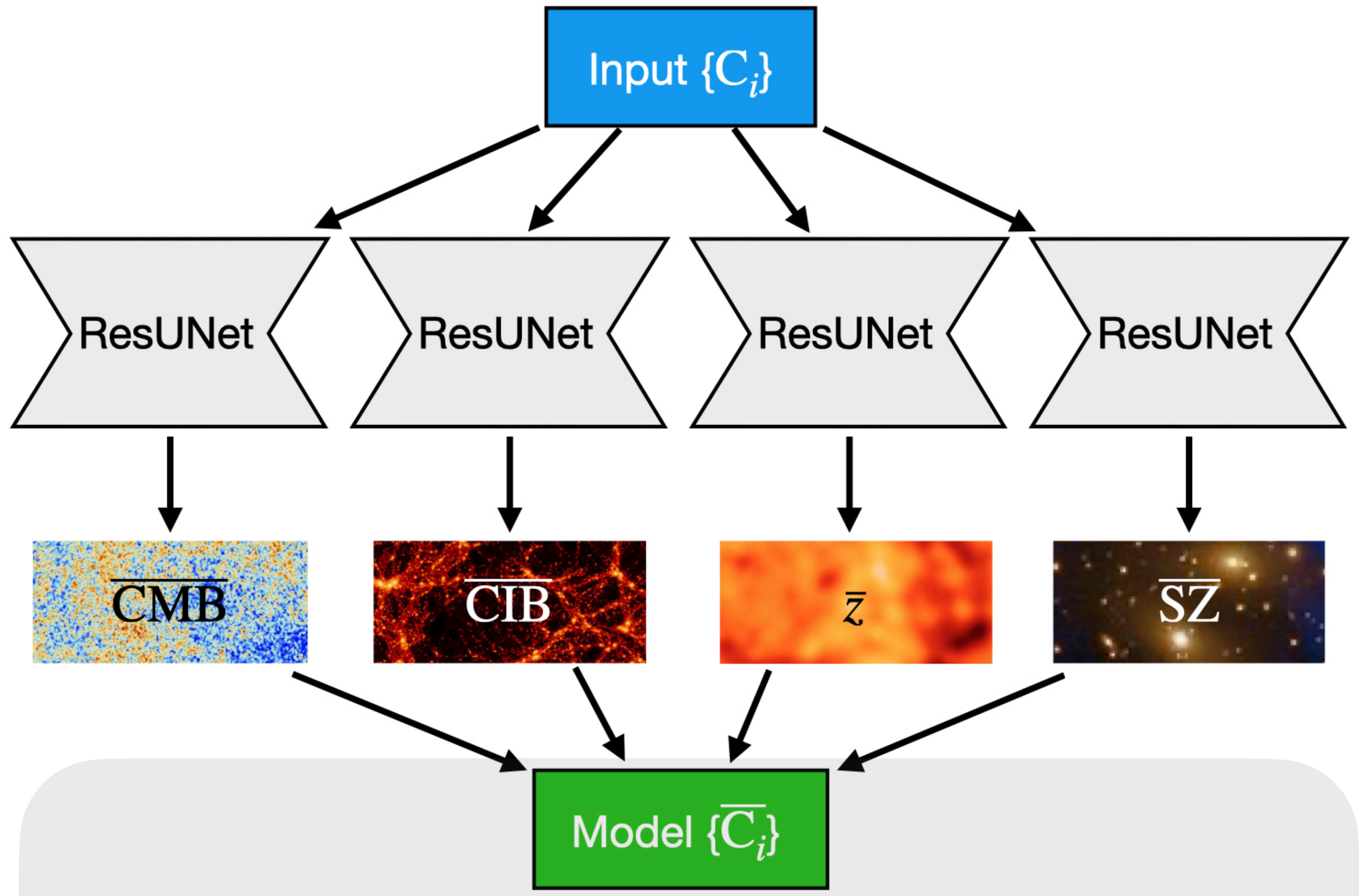
Bonjean et al, 2024

Method

$$C_i = 1 \times \mathrm{CMB} + f_i \times \mathrm{SZ} + \mathrm{CIB}_i$$



1) Project HEALPIX nside=4096 in patches



2) Network with n DIP for n components

3) Reconstruct the maps with the approximated mixing matrix

$$\overline{C}_i = 1 \times \overline{\mathrm{CMB}} + f_i \times \overline{\mathrm{SZ}} + \psi_i(\bar{z})\,\overline{\mathrm{CIB}}$$

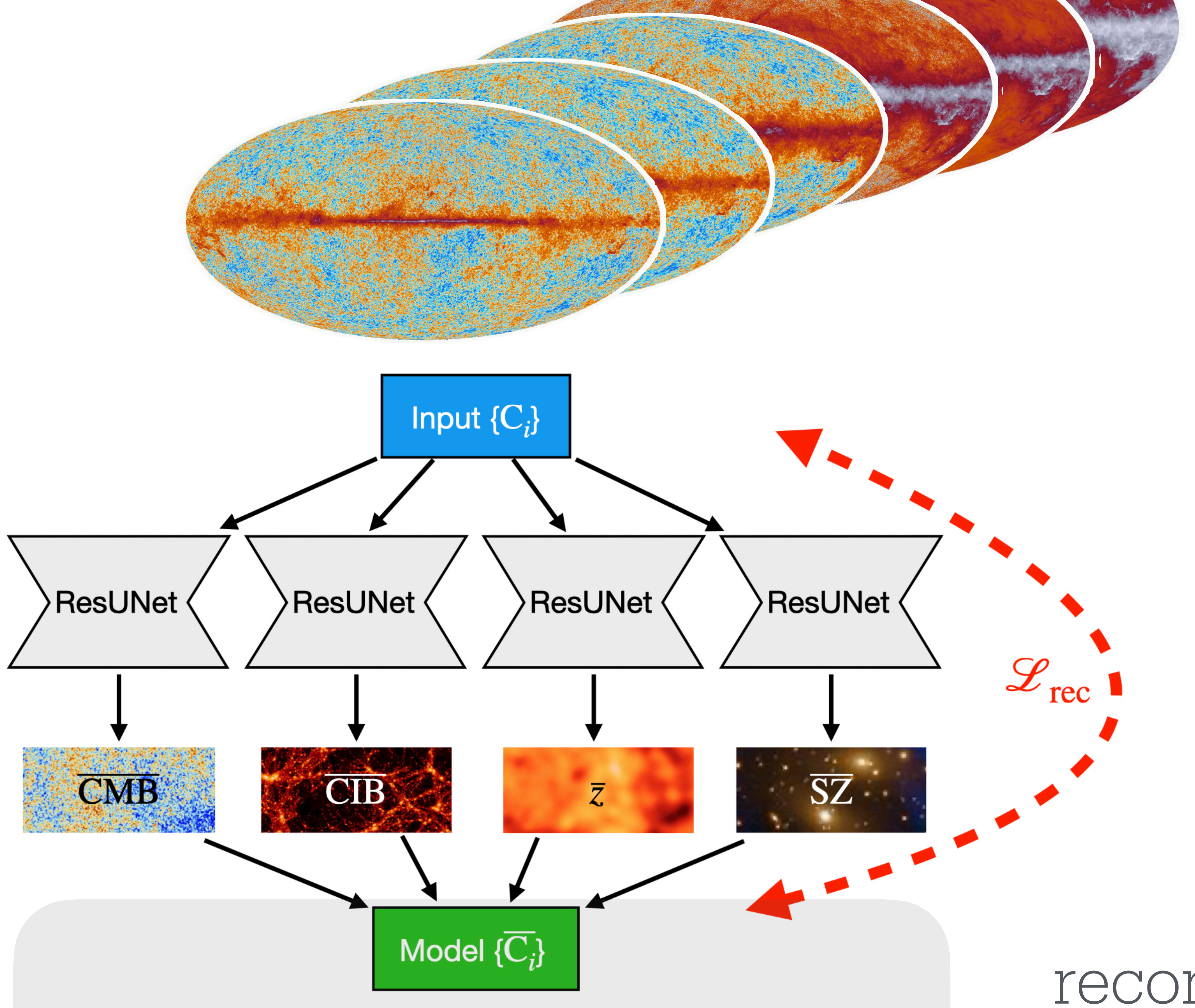
$$\psi_i(\bar{z}) = \left(\frac{i}{545}\right)^{\beta+3} \times \frac{\exp\left(\frac{h \times 545 \times 10^9 (1+\bar{z})}{k_B T_0 (1+\bar{z})^{\alpha}}\right) - 1}{\exp\left(\frac{h \times i \times 10^9 (1+\bar{z})}{k_B T_0 (1+\bar{z})^{\alpha}}\right) - 1}$$

Bonjean et al, 2024

Method

$$C_i = 1 \times \mathrm{CMB} + f_i \times \mathrm{SZ} + \mathrm{CIB}_i$$

1) Project HEALPIX nside=4096 in patches



2) Network with n DIP for n components

3) Reconstruct the maps with the approximated mixing matrix

$$\overline{C}_i = 1 \times \overline{\mathrm{CMB}} + f_i \times \overline{\mathrm{SZ}} + \psi_i(\bar{z})\, \overline{\mathrm{CIB}}$$

$$\psi_i(\bar{z}) = \left(\frac{i}{545}\right)^{\beta+3} \times \frac{\exp\left(\frac{h \times 545 \times 10^9 (1+\bar{z})}{k_B T_0 (1+\bar{z})^{\alpha}}\right) - 1}{\exp\left(\frac{h \times i \times 10^9 (1+\bar{z})}{k_B T_0 (1+\bar{z})^{\alpha}}\right) - 1}$$

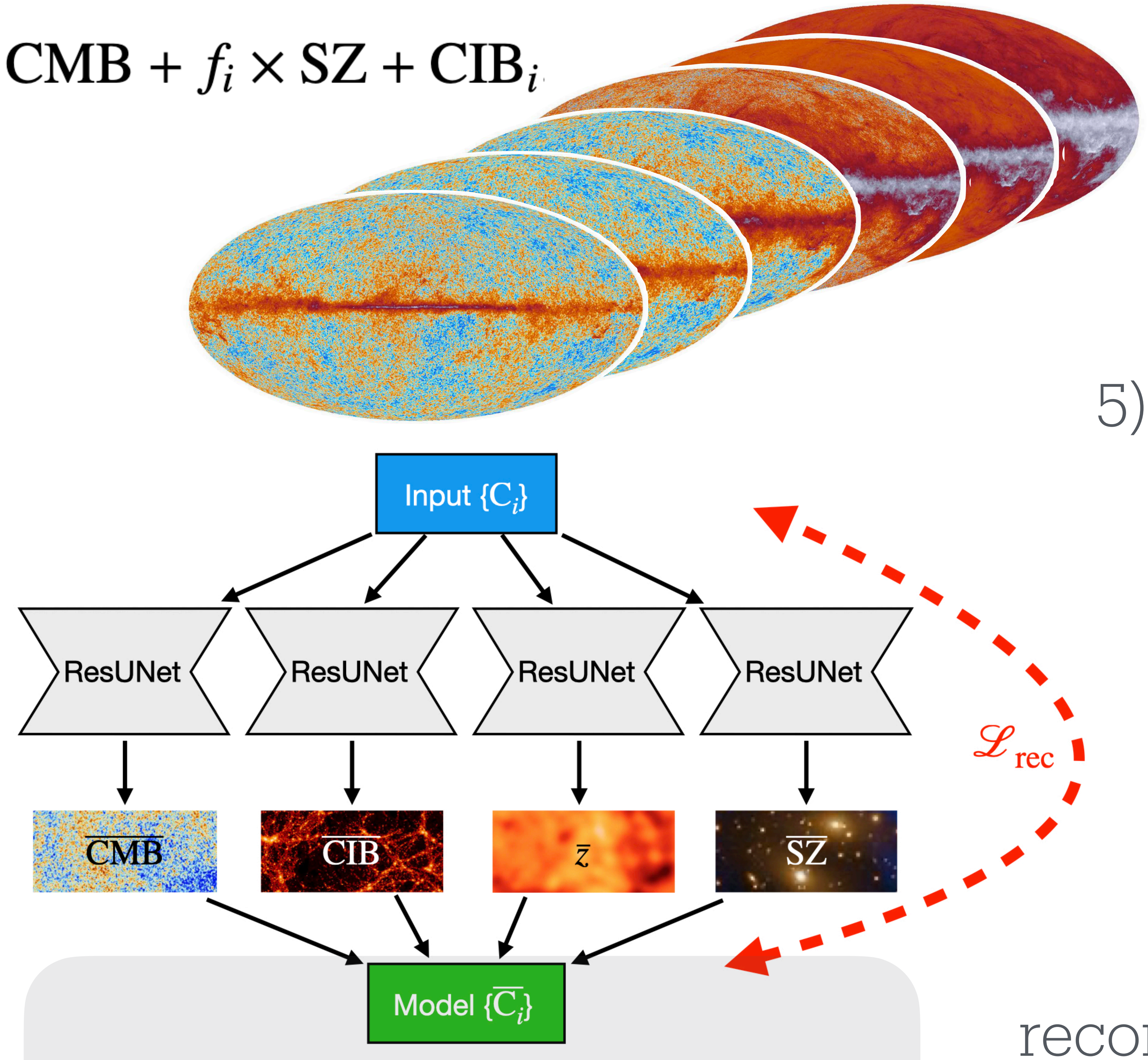
4) Reconstruction loss between reconstructed maps and input maps

Bonjean et al, 2024

Method

$$C_i = 1 \times \text{CMB} + f_i \times \text{SZ} + \text{CIB}_i$$

1) Project HEALPIX nside=4096 in patches



5) Reconstruct HEALPIX maps with the extracted components

2) Network with n DIP for n components

4) Reconstruction loss between reconstructed maps and input maps

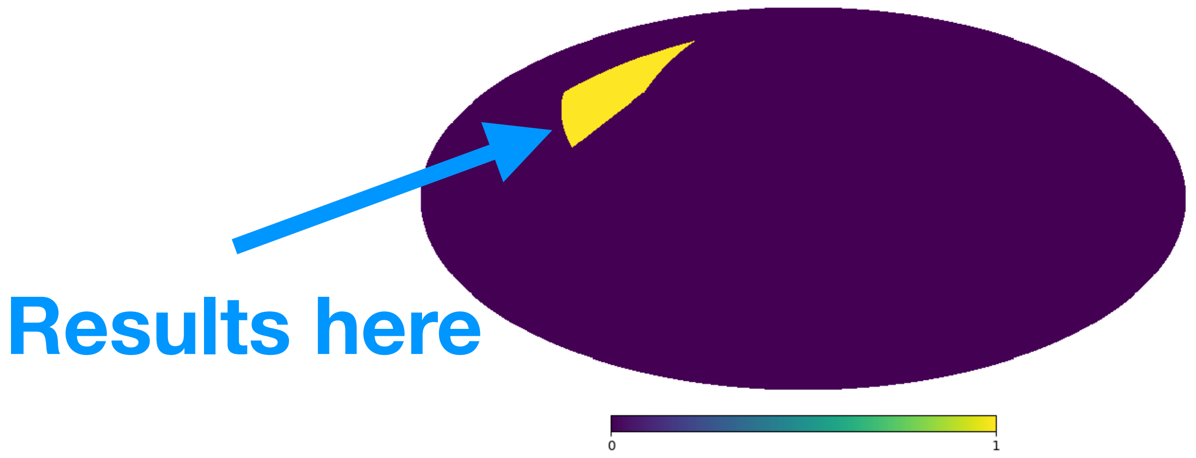
3) Reconstruct the maps with the approximated mixing matrix

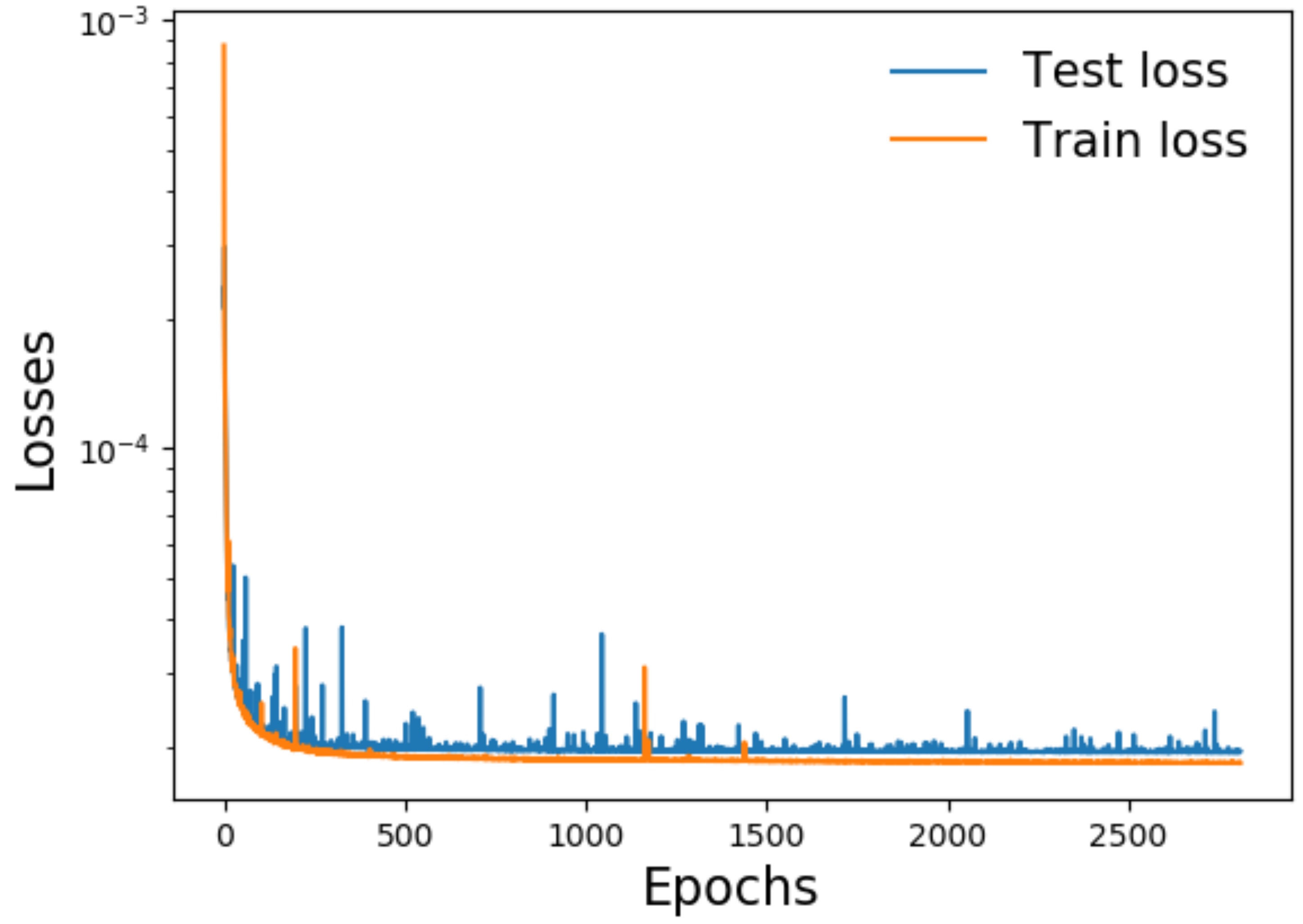
$$\overline{C}_i = 1 \times \overline{\text{CMB}} + f_i \times \overline{\text{SZ}} + \psi_i\,(\bar{z})\,\overline{\text{CIB}}$$

$$\psi_i\,(\bar{z}) = \left(\frac{i}{545}\right)^{\beta+3} \times \frac{\exp\left(\frac{h \times 545 \times 10^9 (1+\bar{z})}{k_B \mathrm{T}_0 (1+\bar{z})^{\alpha}}\right) - 1}{\exp\left(\frac{h \times i \times 10^9 (1+\bar{z})}{k_B \mathrm{T}_0 (1+\bar{z})^{\alpha}}\right) - 1}$$

Bonjean et al, 2024

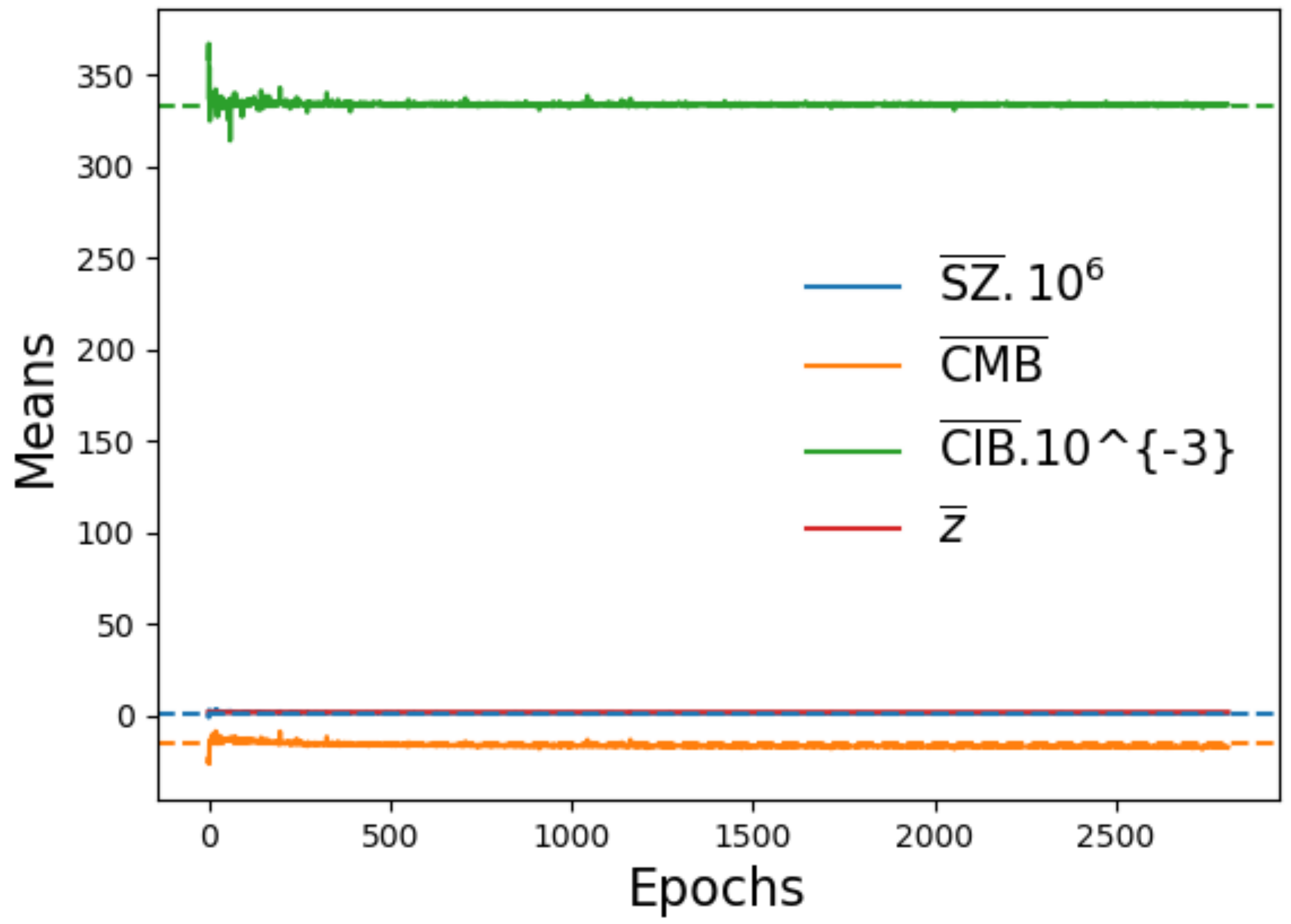
Optimization process



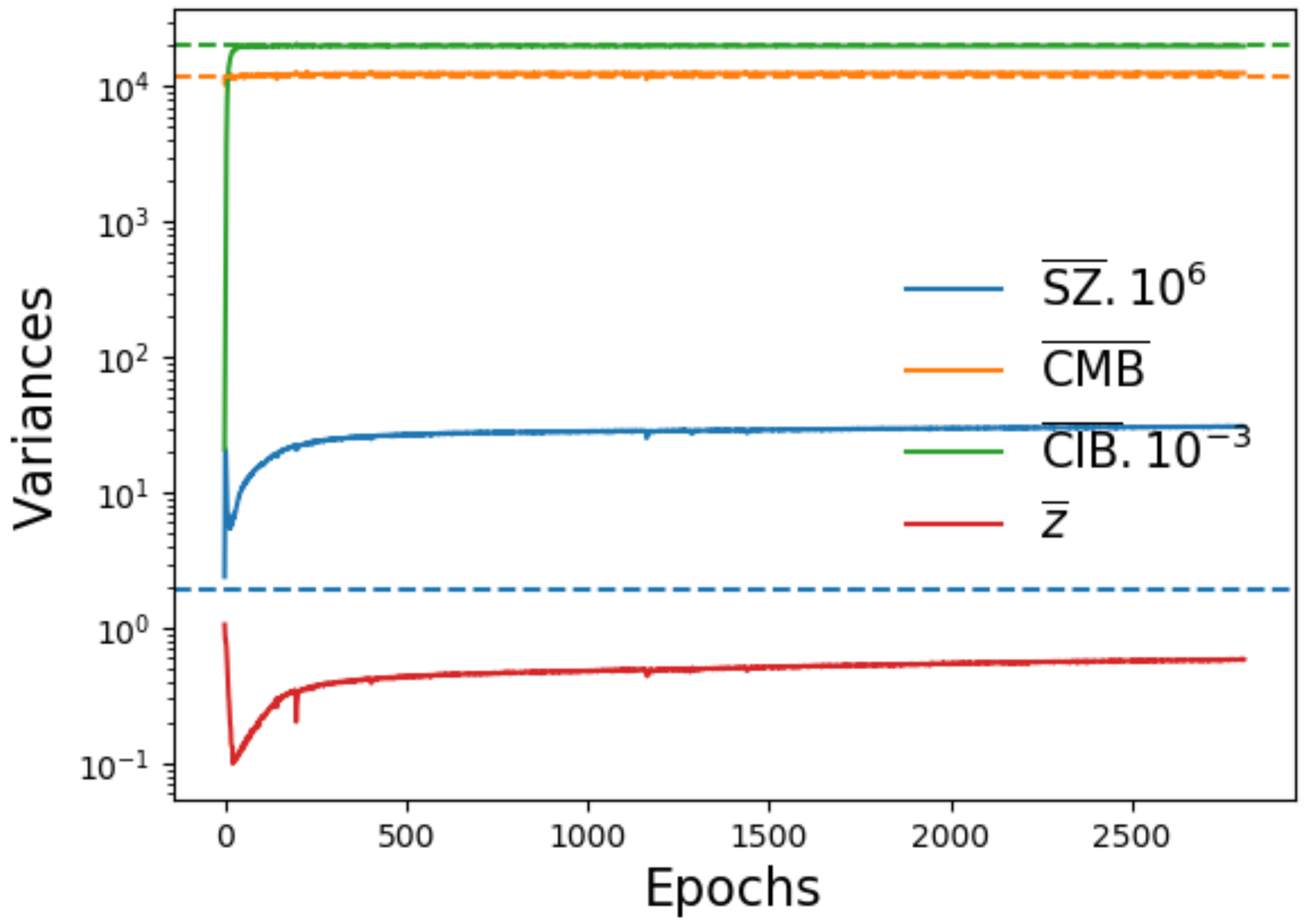


Components are converging into the expected solution during the optimization

Means:



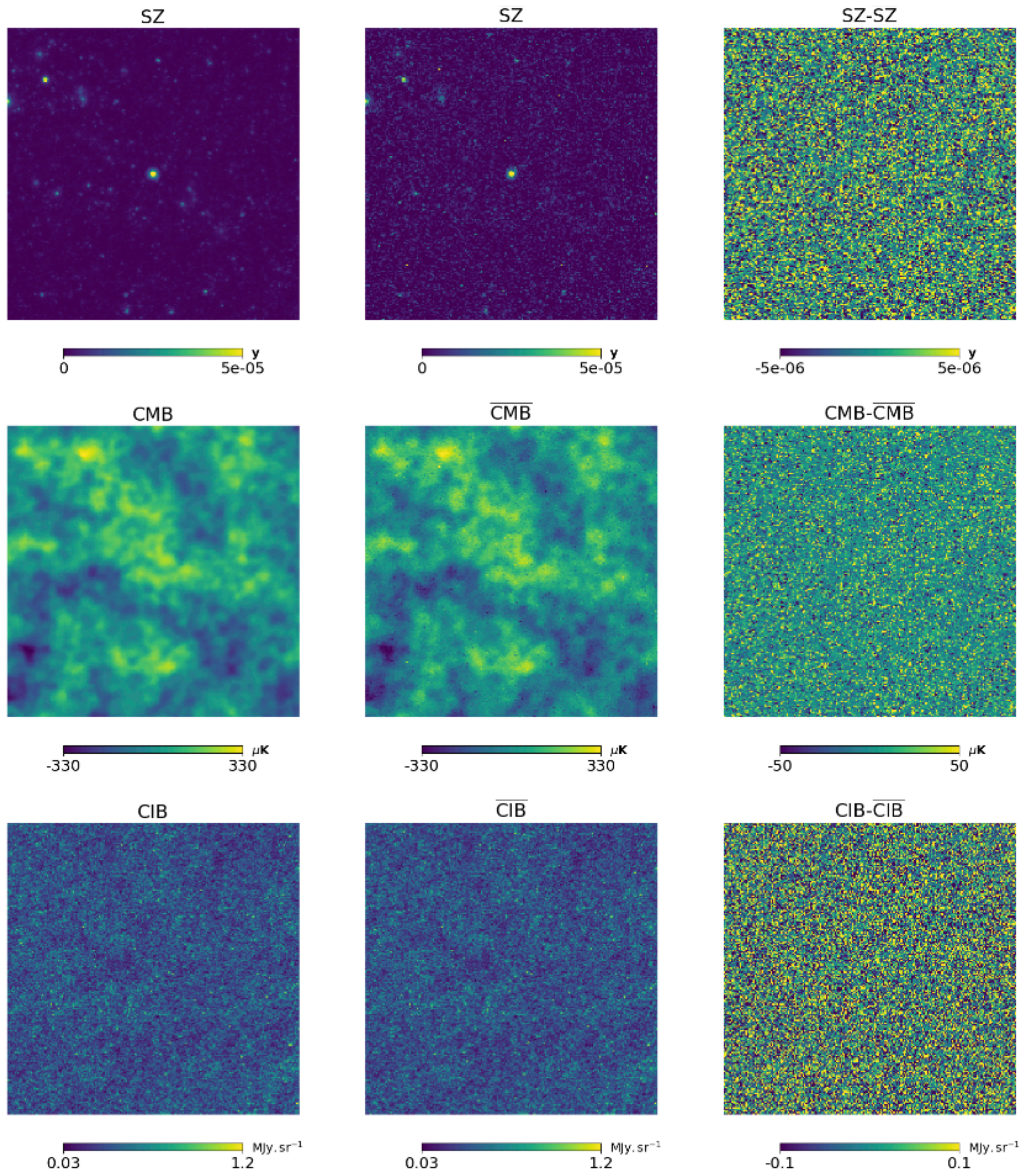
Variances:



Results

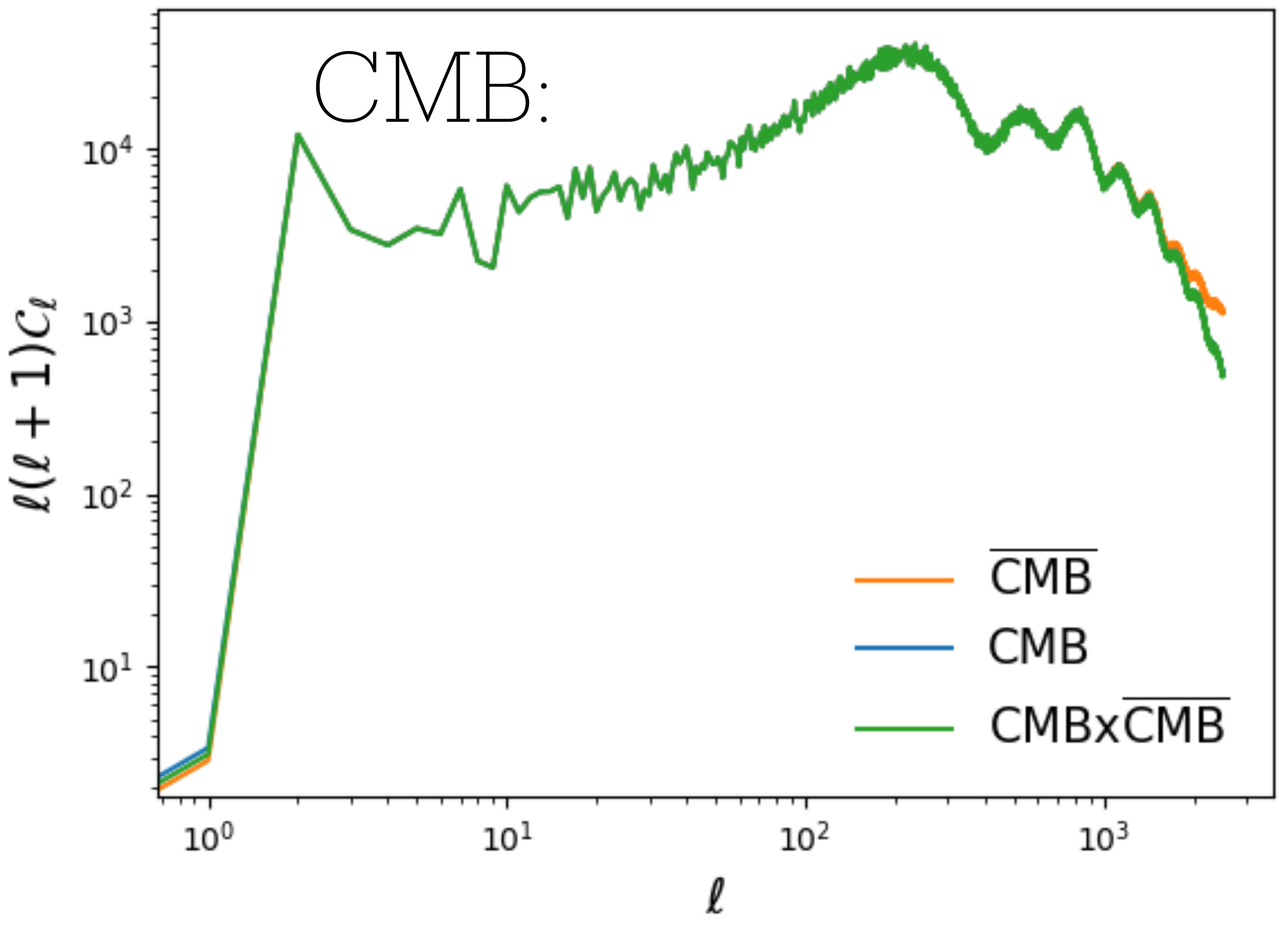
Results

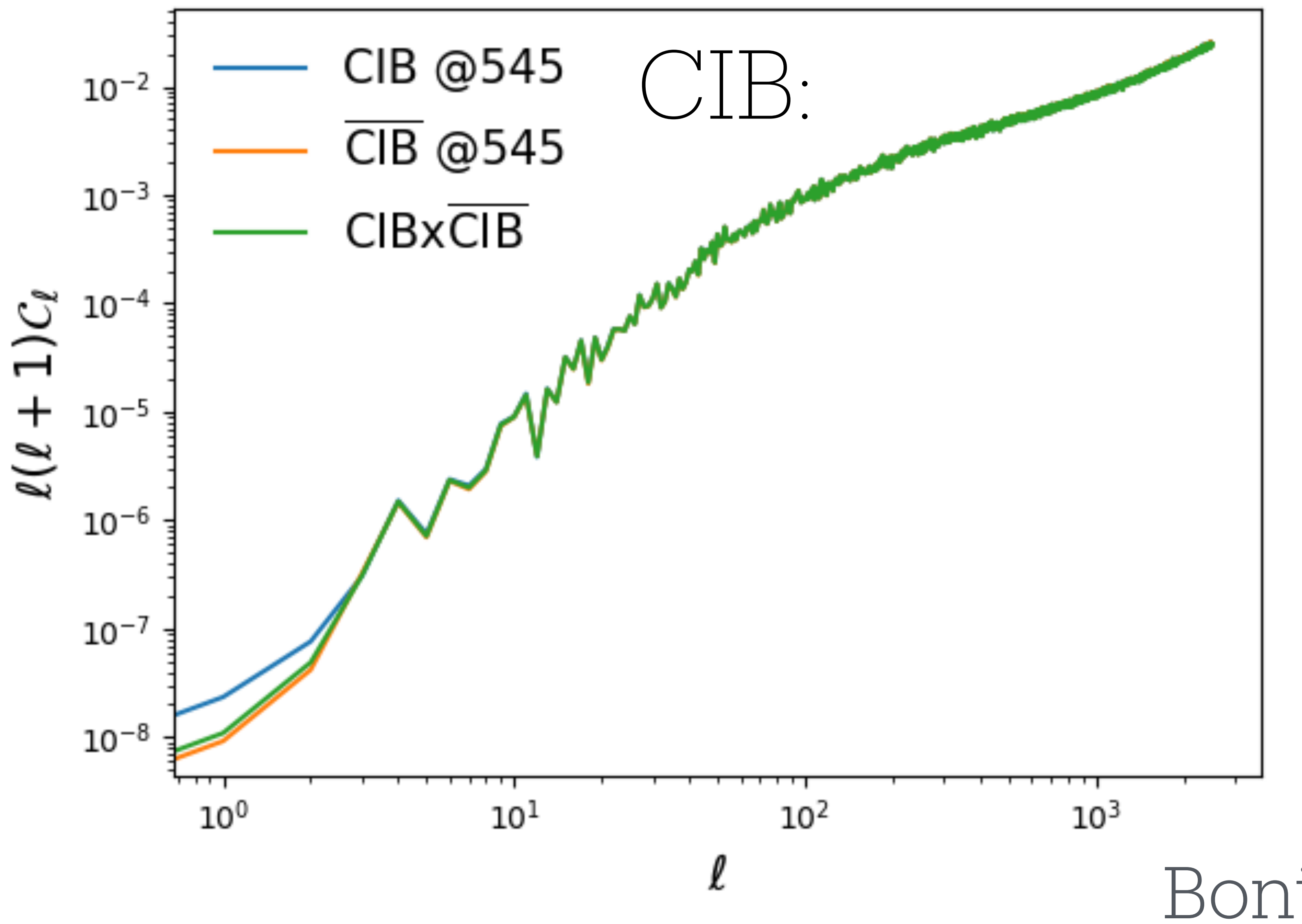
Quick look:



Results

Power spectra:

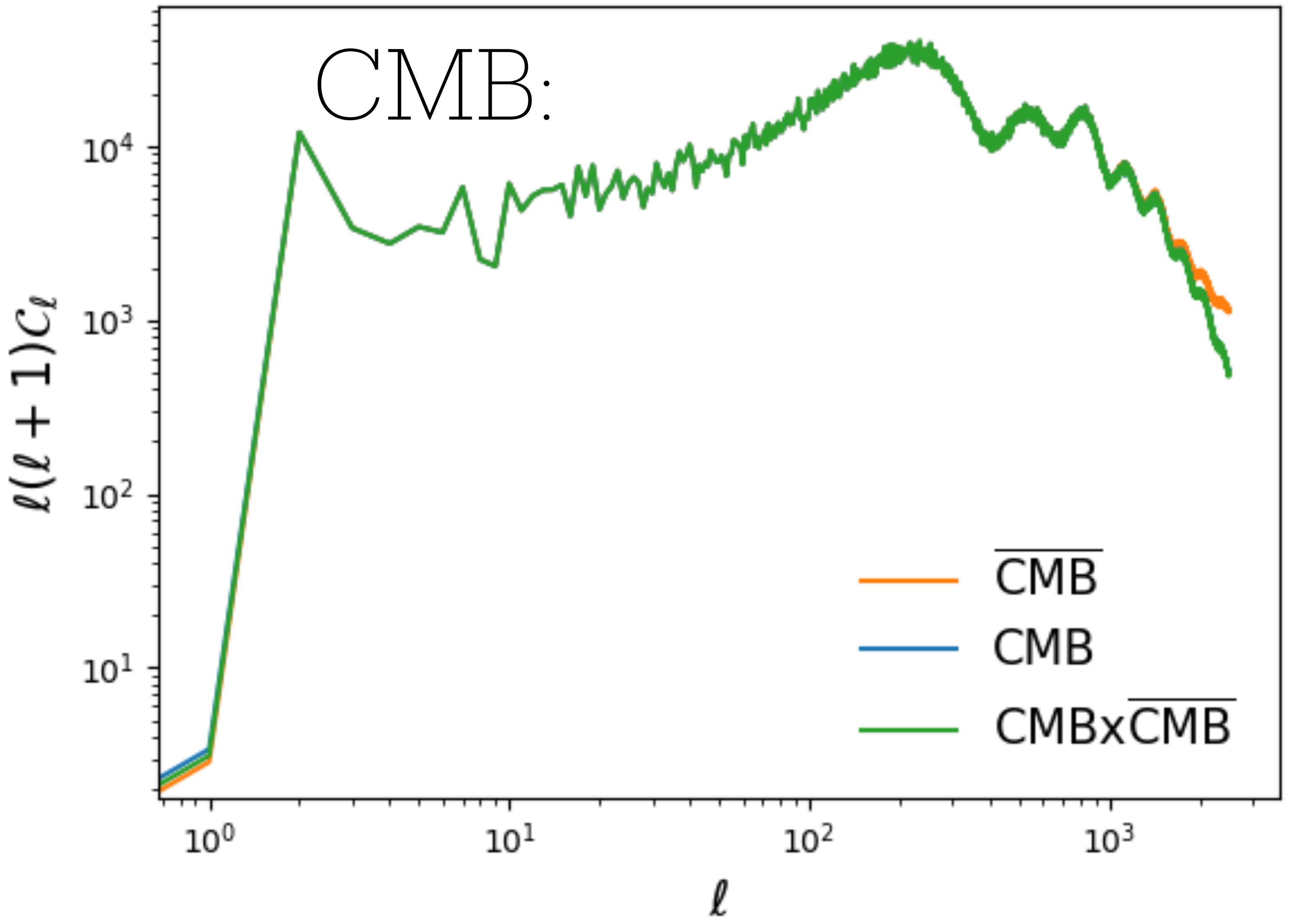


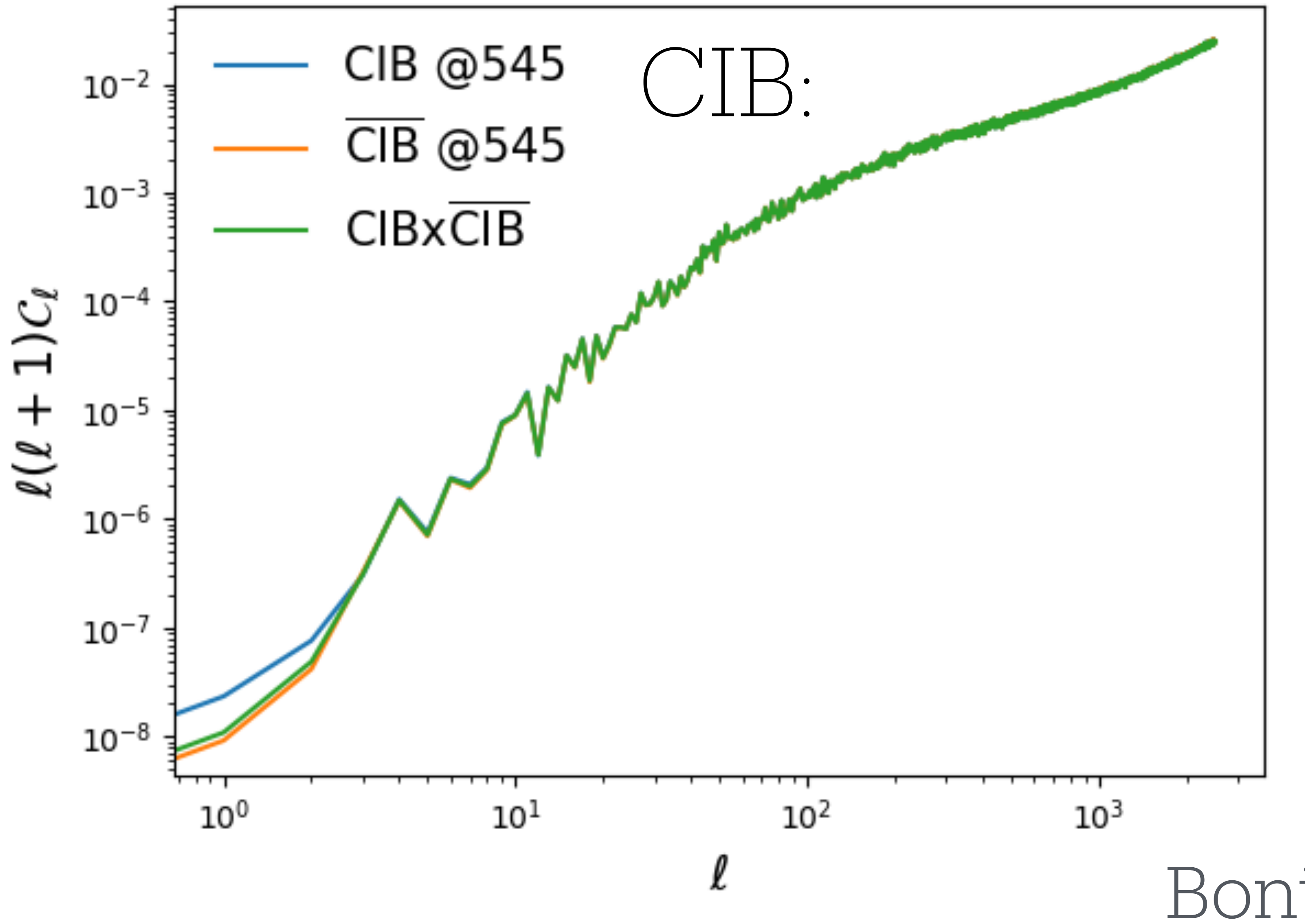


Bonjean et al, 2024

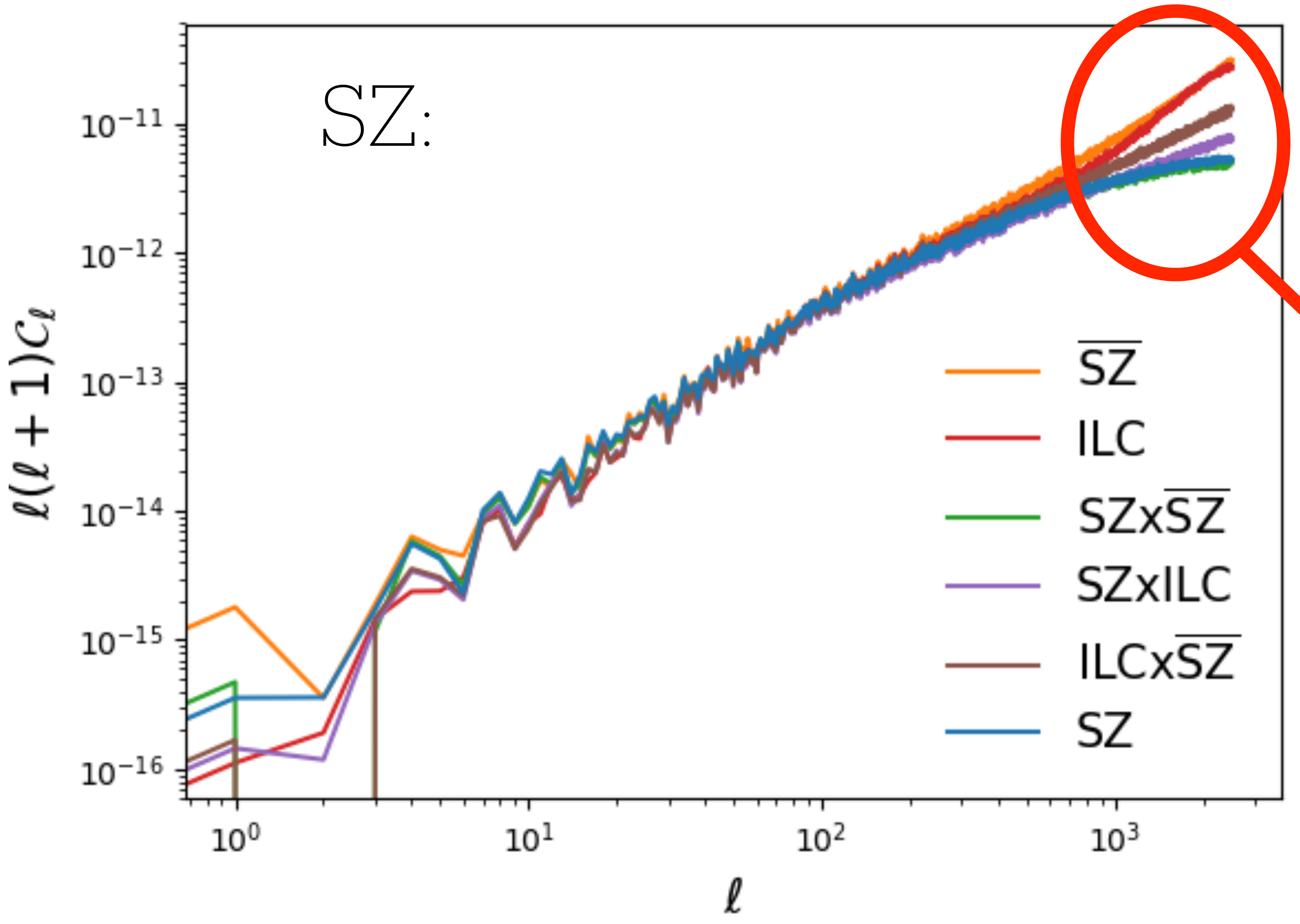
Results

Power spectra:





Bonjean et al, 2024



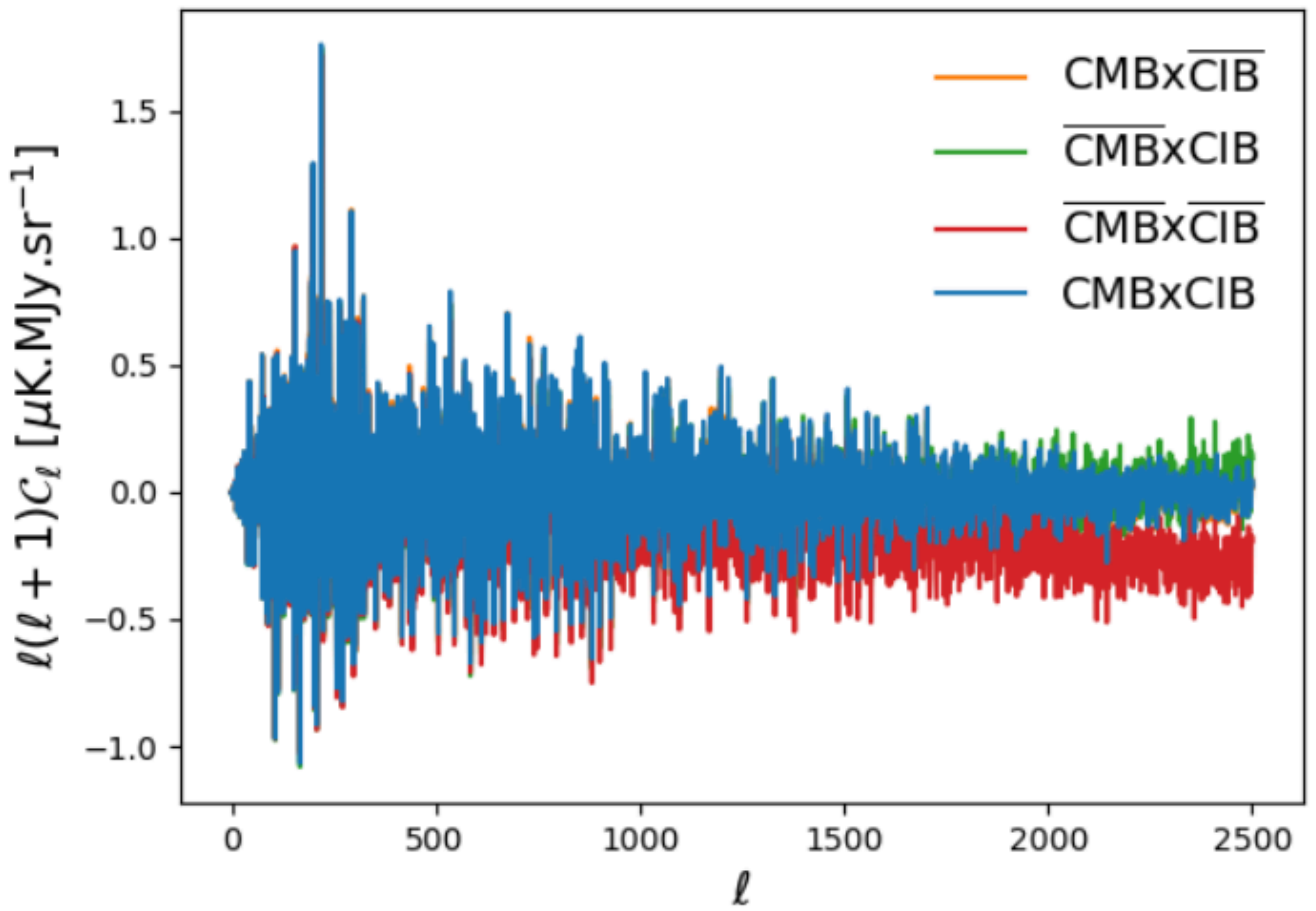
Comparison with MILCA (Planck)

Excess of signal in SZxMILCA

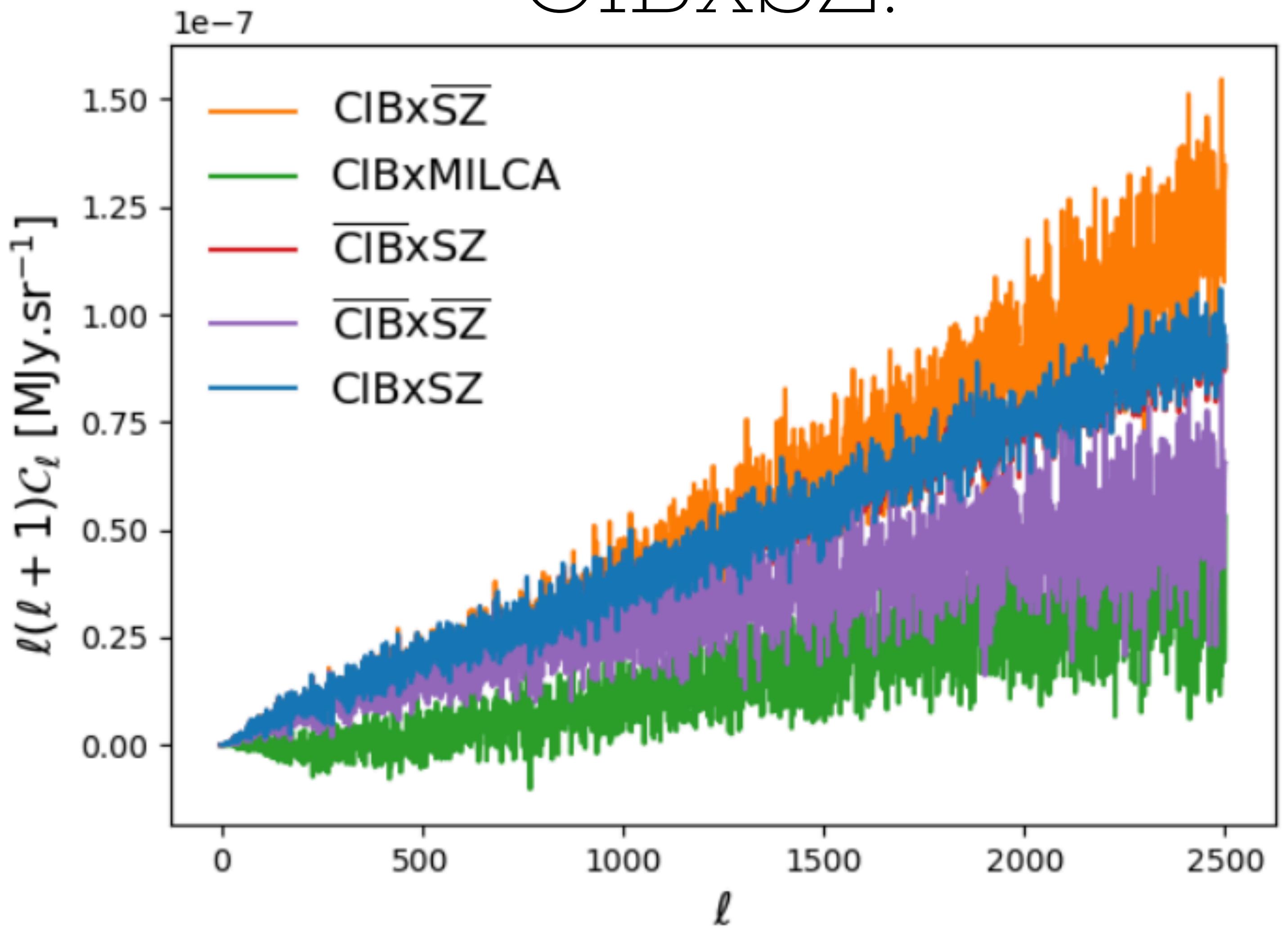
Results

Contamination:

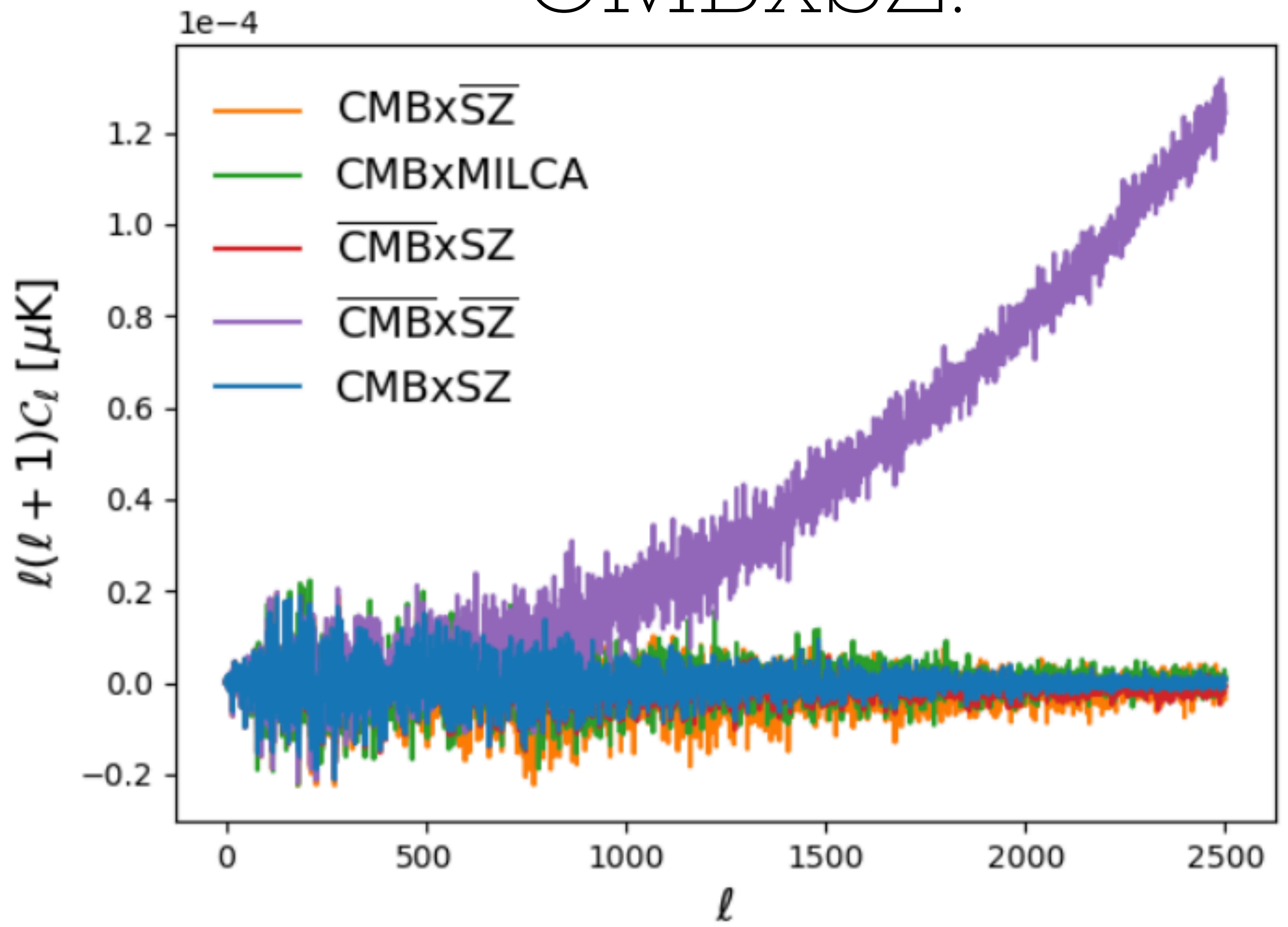
CMBxCIB:



CIBxSZ:



CMBxSZ:

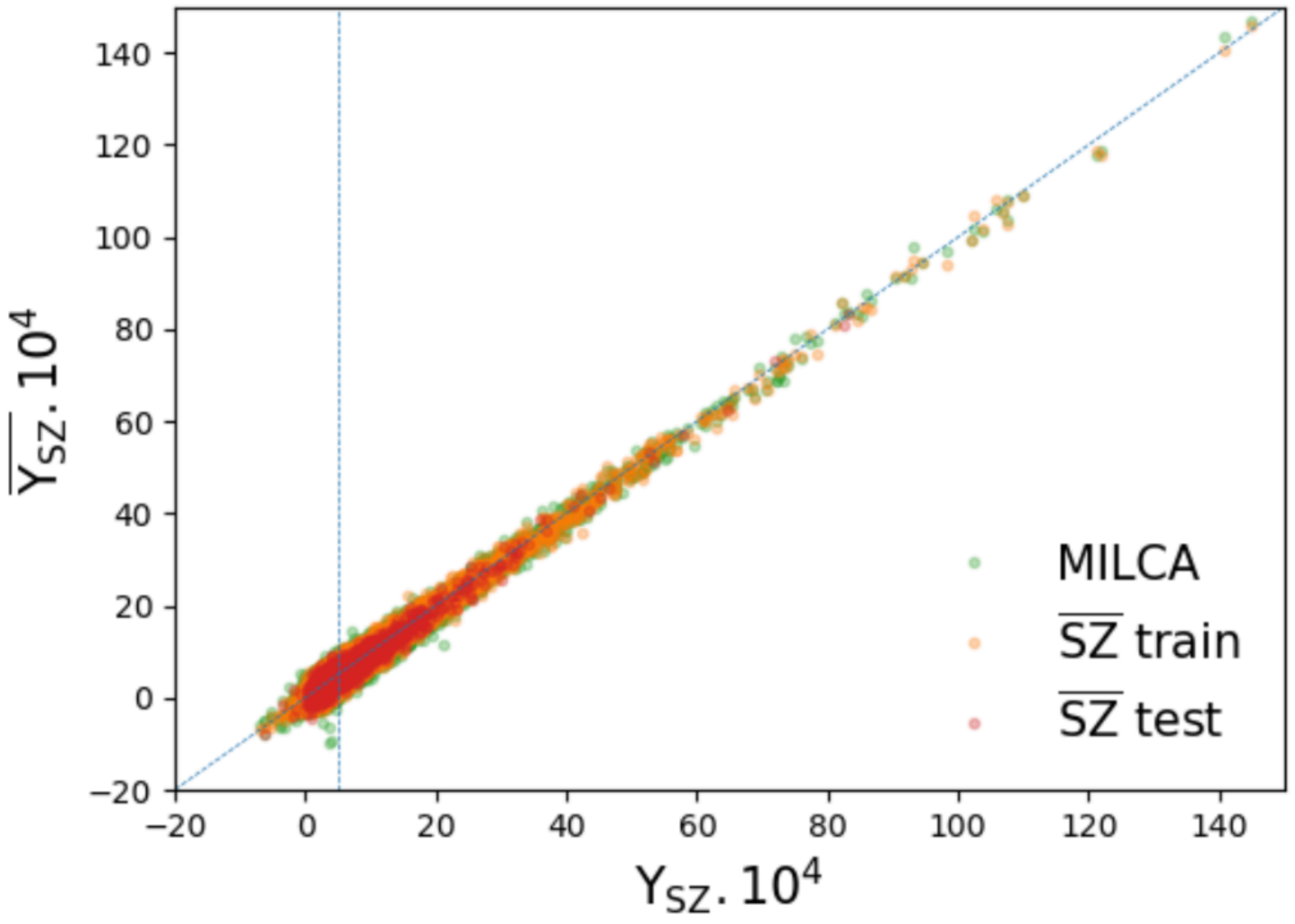


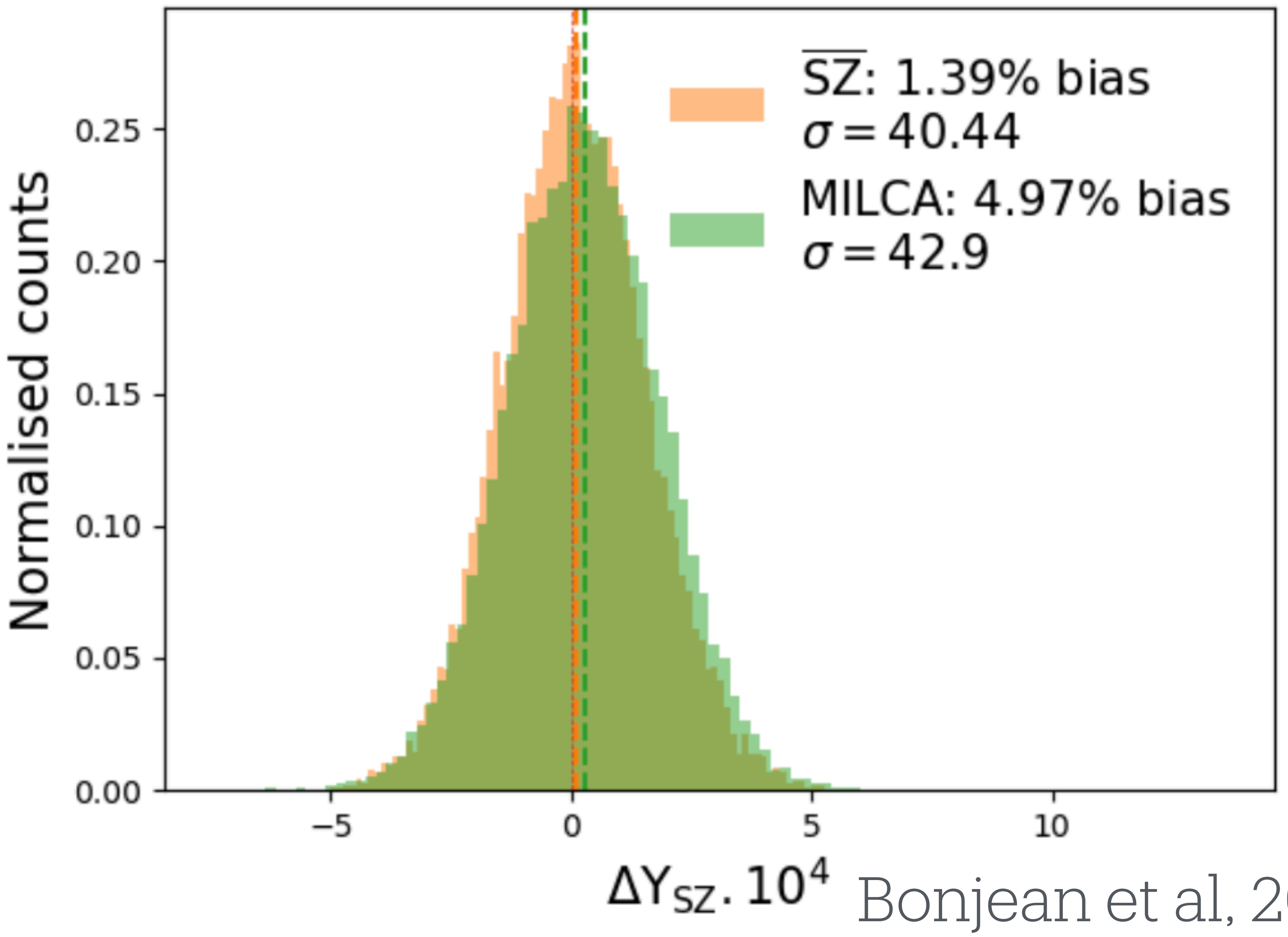
Bonjean et al, 2024

SZ map less contaminated by other components

Results

SZ fluxes around clusters:





Bonjean et al, 2024

Competitive with MILCA

Results including foregrounds, beams, and noise

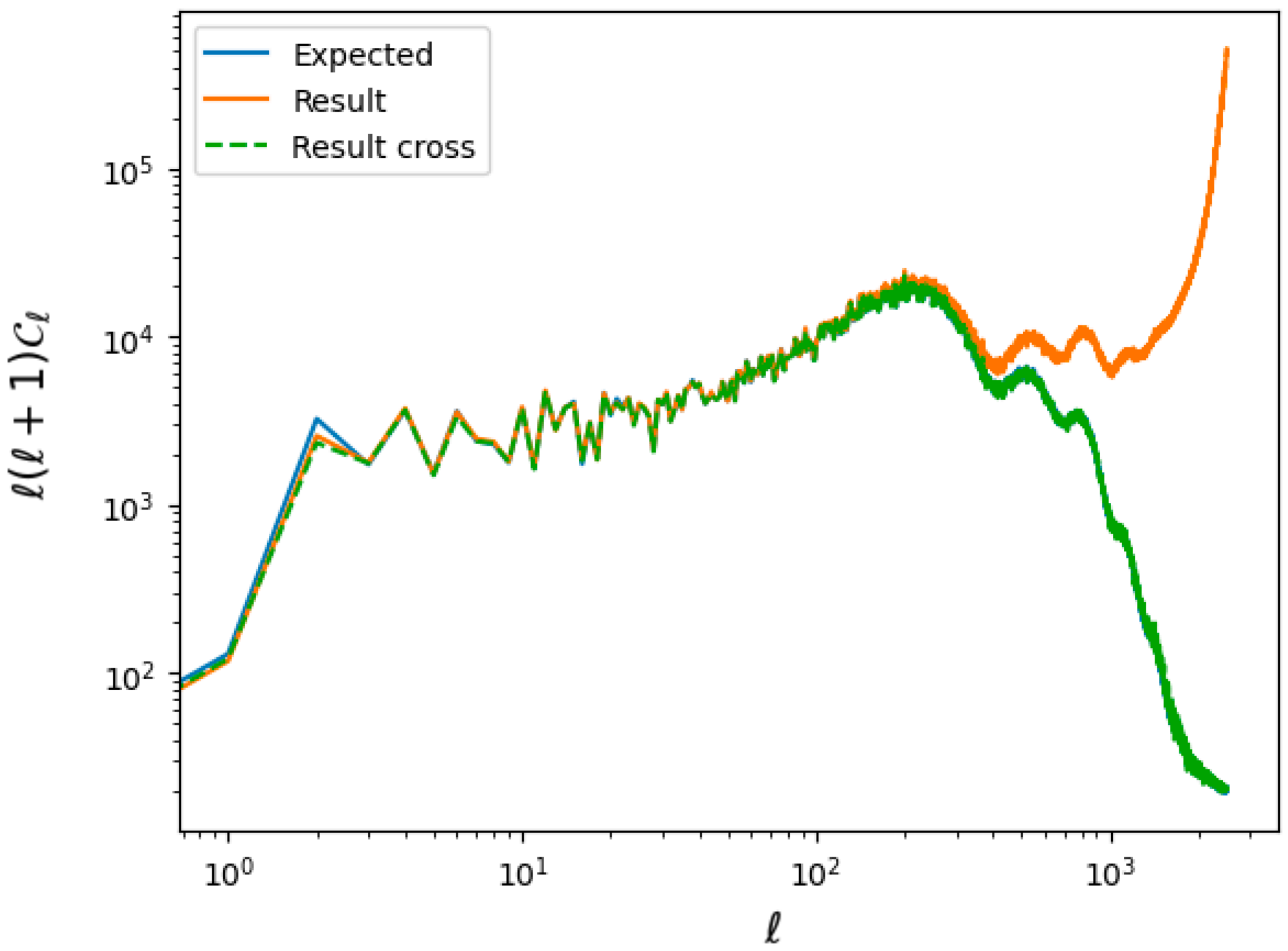
Results **including foregrounds, beams, and noise**

Second study on **WebSky** numerical simulations (Stein et al, 2020) and Planck Sky Model (pySM3, Thorne et al, 2016):

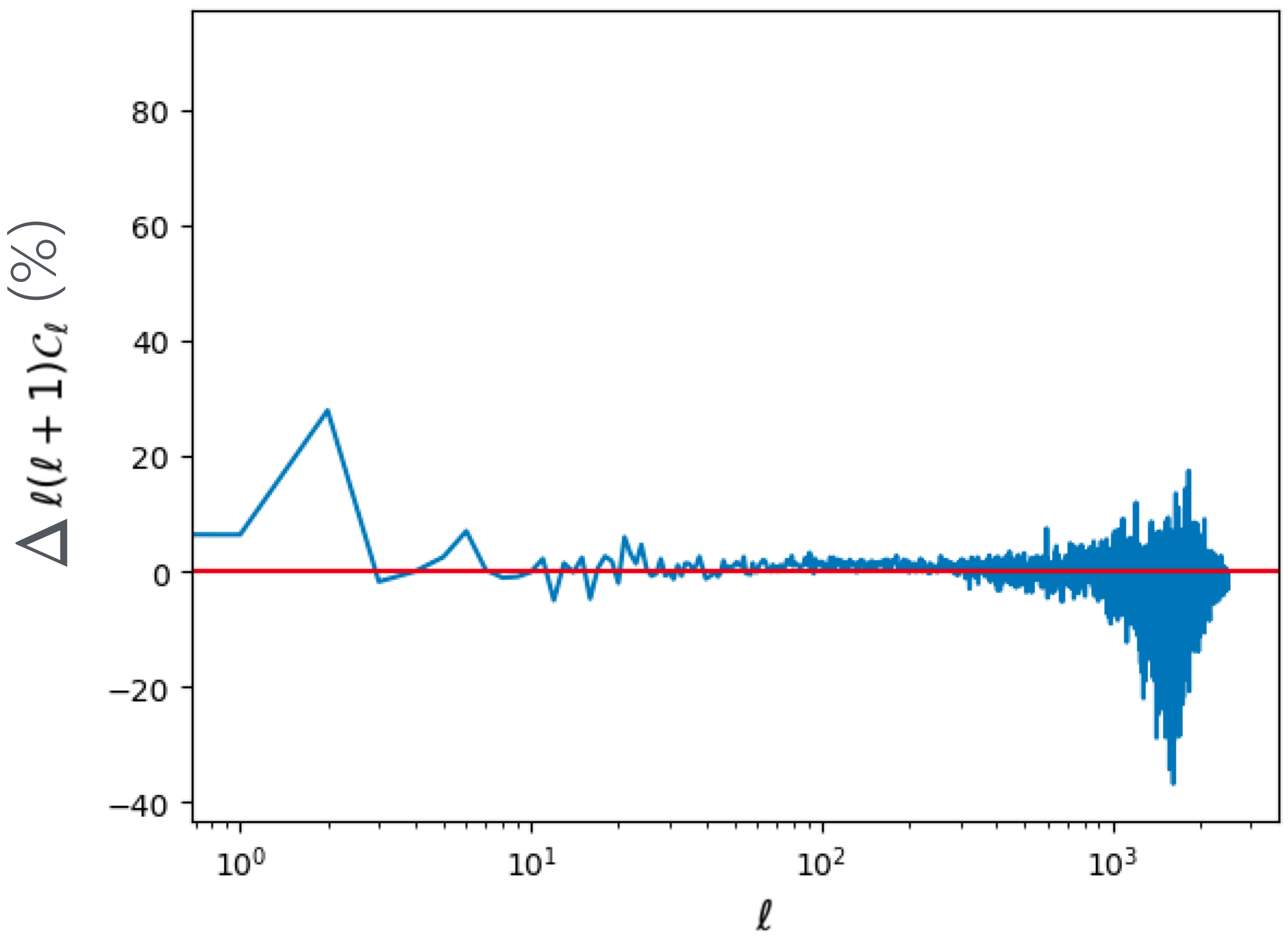
- 100, 143, 217, 353, 545 GHz (Planck)
- Healpix nside=2048
- CMB, CIB, tSZ, kSZ, radio sources, free-free, synchrotron, CO, AME, dust
- Noise & beams of Planck (no degraded resolution)

Results **including foregrounds, beams, and noise**

Cross-correlation on HM1 and HM2



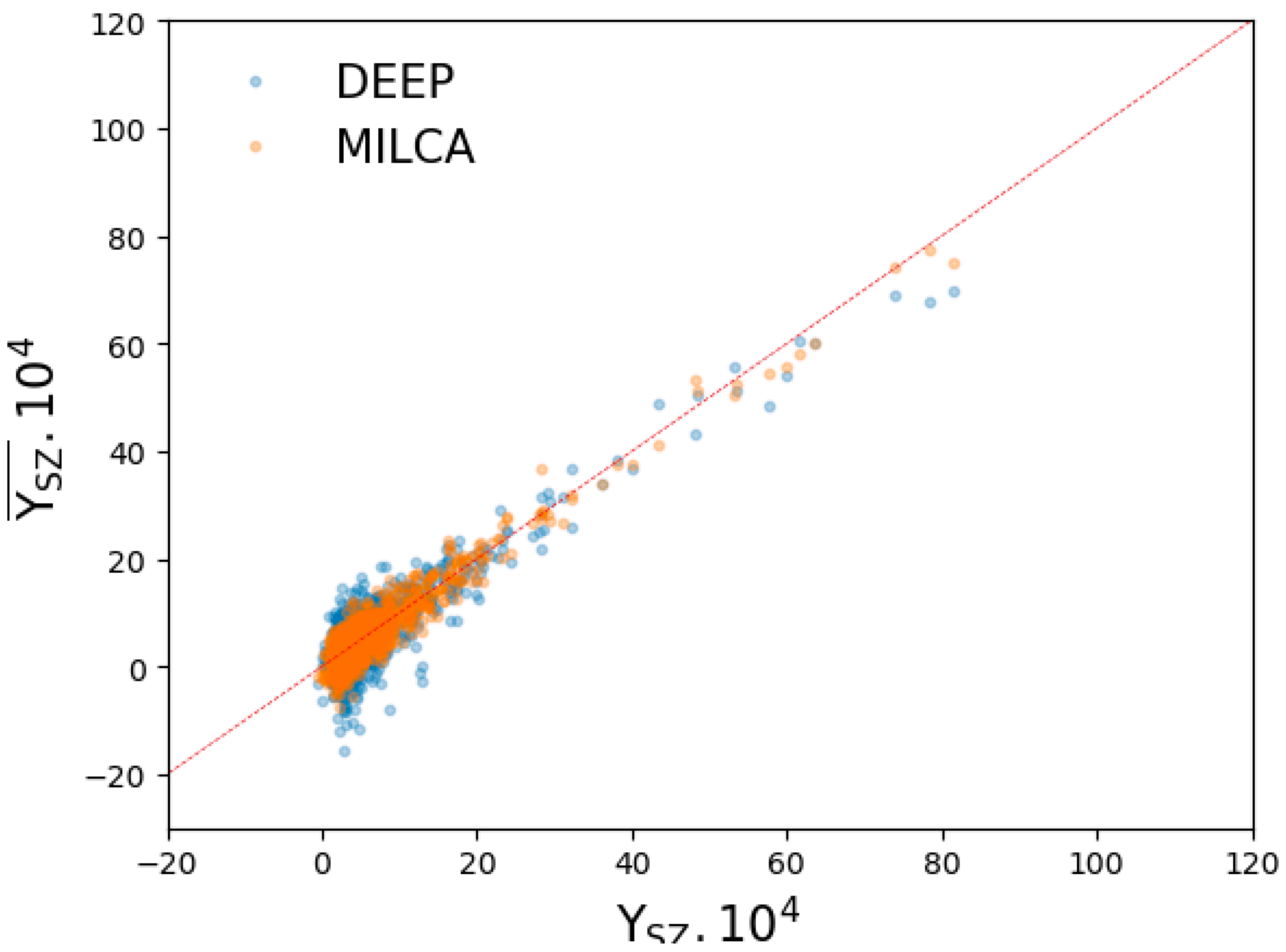
CMB:



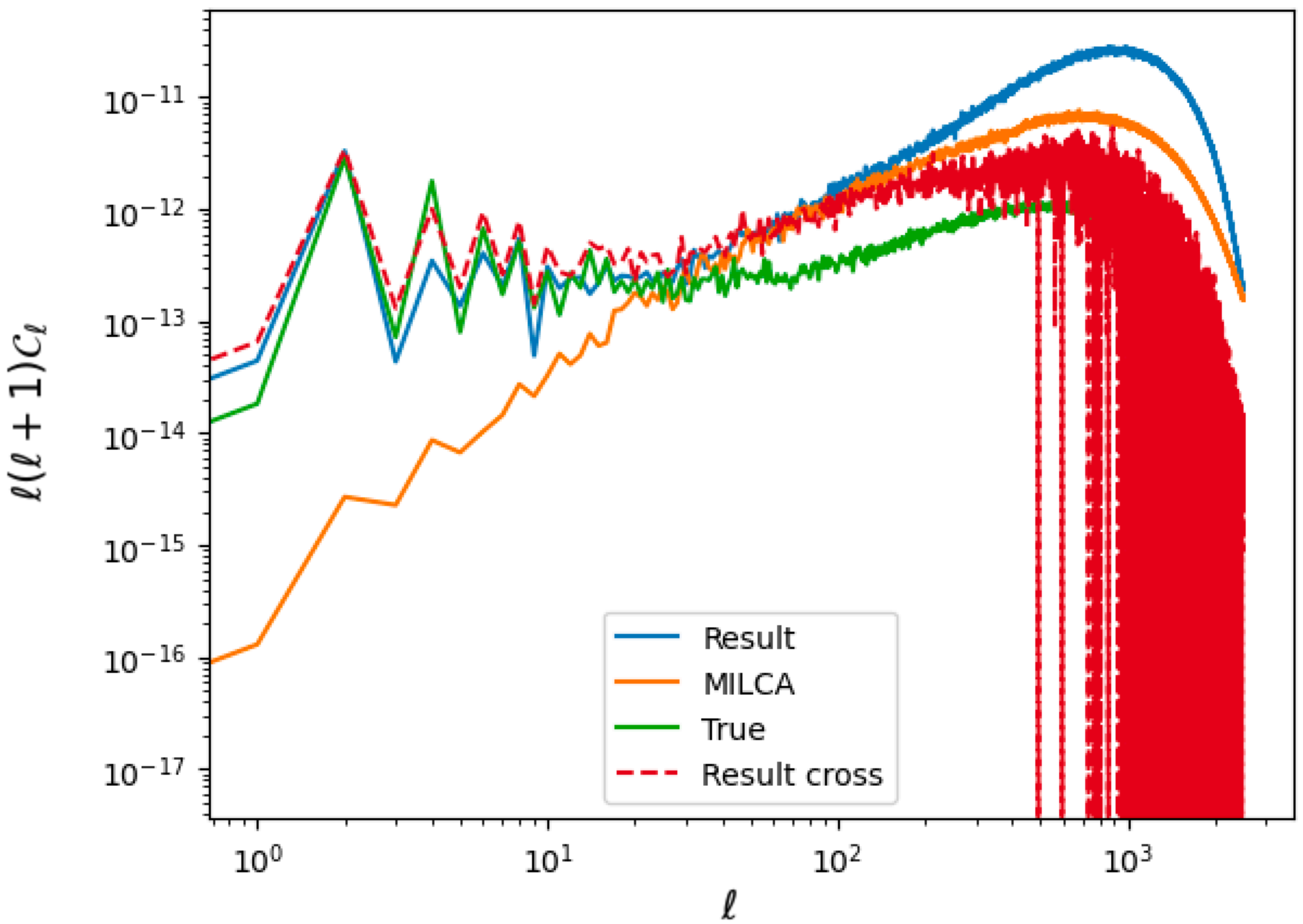
Bonjean et al, in prep.

Results **including foregrounds, beams, and noise**

SZ:



Cross-correlation on HM1 and HM2



Bonjean et al, in prep.

Summary

- Functional deep learning network for blind multi-component separation of CMB, SZ, and CIB
- **Competitive** with state-of-the-art methods (MILCA)
- **Less contaminated** by other components thanks to an adaptative CIB model
- **Use the full information** of the maps (no need to degrade resolution)

Check it out on arxiv!

arXiv

Next steps

- Include **polarization** maps
- Forecast for **SO** & **LiteBIRD**
- Application on foreground removal for **SKA** (M. Spinelli, J-L Starck)

TITAN ARTIFICIAL INTELLIGENCE IN ASTROPHYSICS · FORTH INSTITUTE OF COMPUTER SCIENCE · Signal Processing Laboratory · CosmoStat · cea

Funded by the European Union