Blind Source Separation with Learnlets

Victor Bonjean

with A. Gkogkou, J.-L. Starck and P. Tsakalides

TITAN-ARGOS-TOSCA Meeting - July 8th 2025

Illustration

What are Learnlets?

(Ramzi et al., 2021)

→ Extension of wavelets (sparse)

→ Filters are learned (CNN)

→ Mathematical frame (component separation)

Illustration

Wavelets decomposition:

Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$

Illustration

Wavelets decomposition:

Data: $X \mapsto \left\{ wt_1, wt_2, wt_3, wt_4, wt_5 \right\}$

LP HP

Illustration

Wavelets decomposition:

Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$

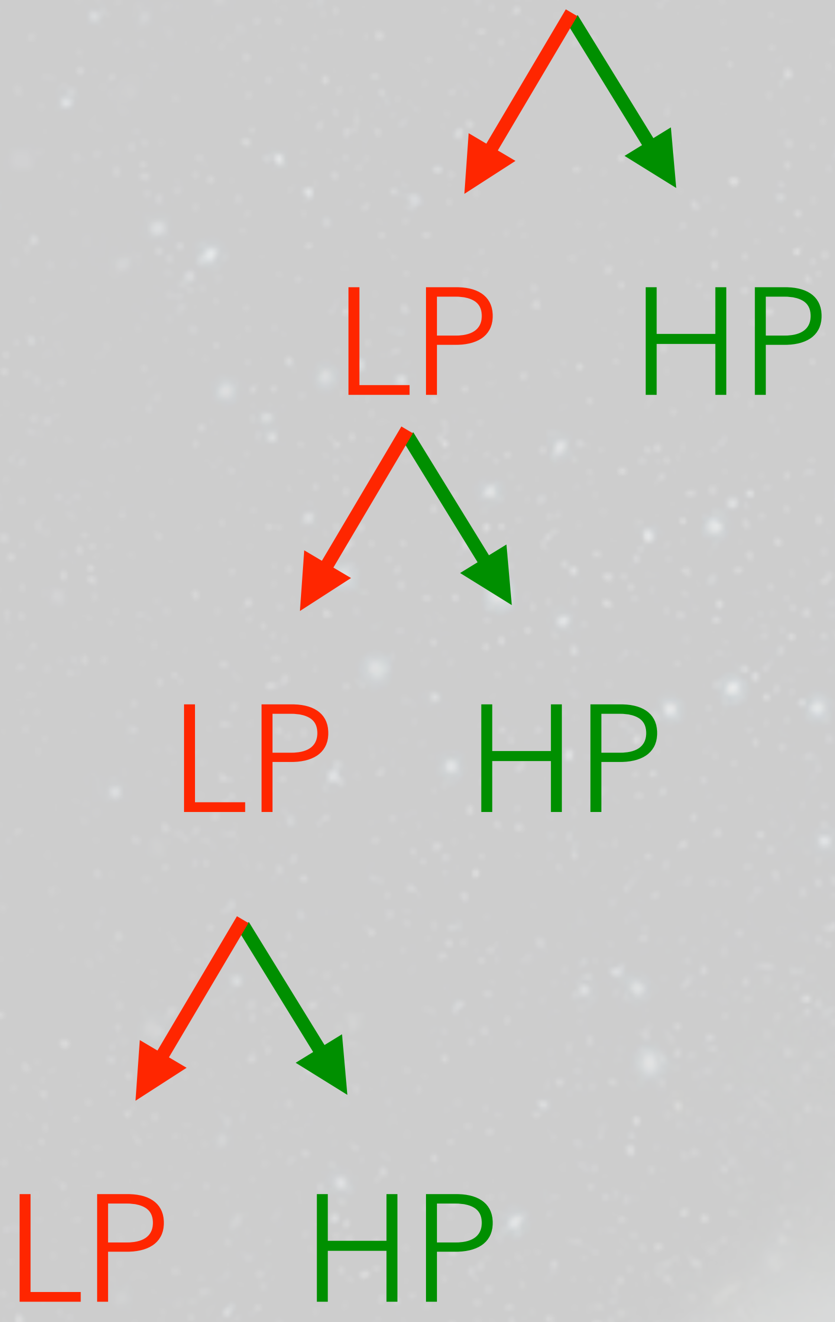
LP HP

LP HP

Illustration

Wavelets decomposition:

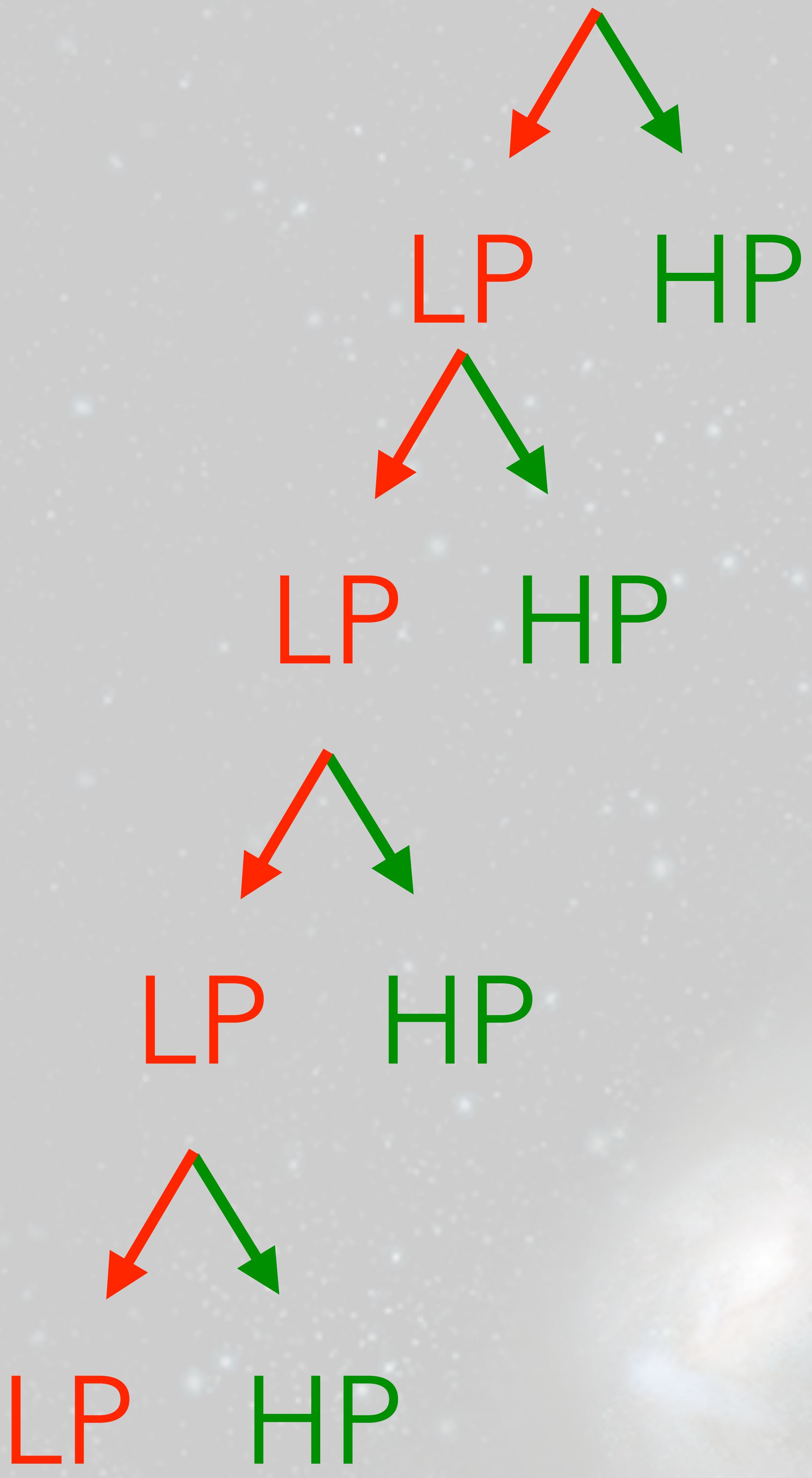
Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$



Illustration

Wavelets decomposition:

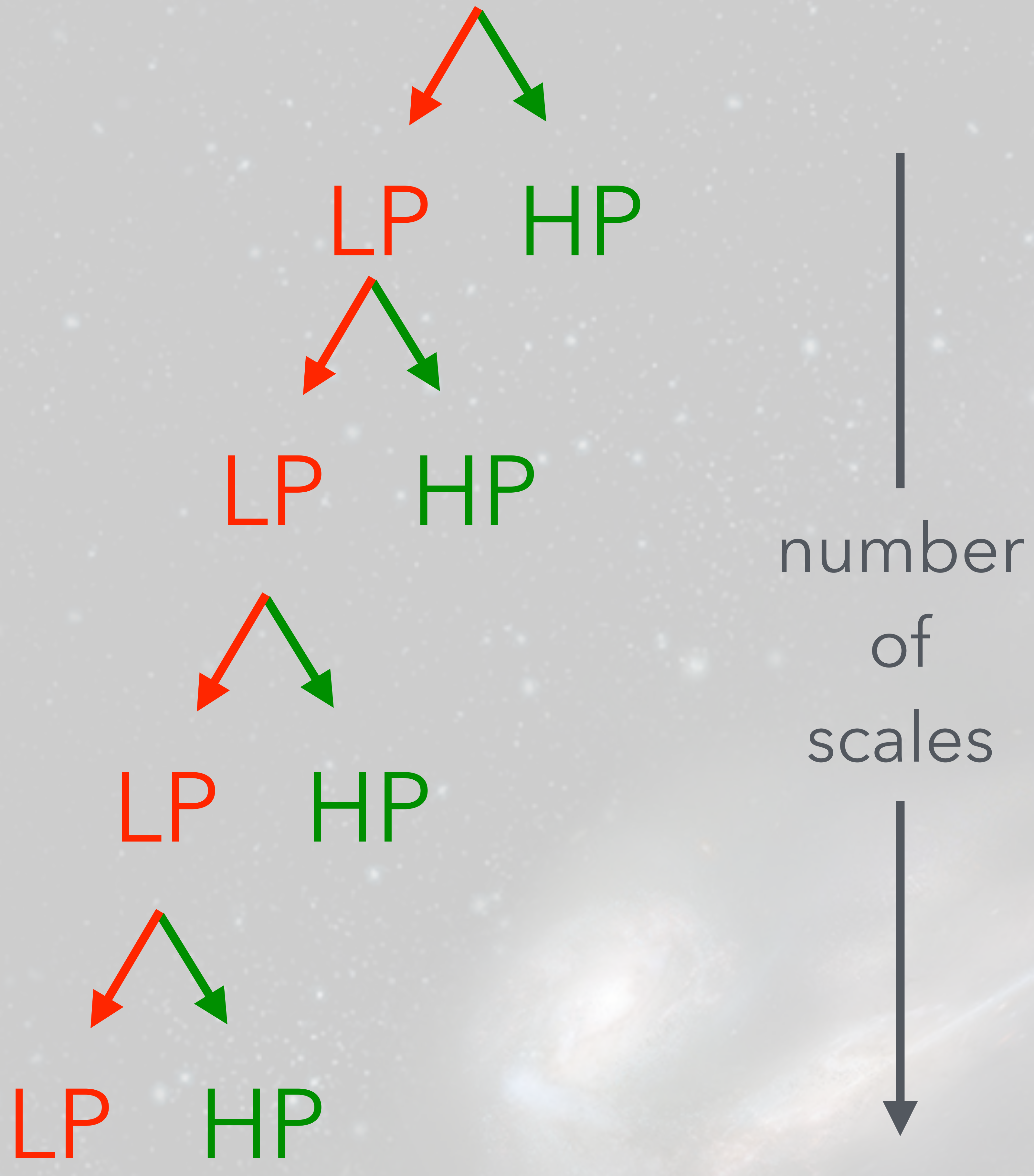
Data: $X \mapsto \left\{wt_1, wt_2, wt_3, wt_4, wt_5\right\}$



Illustration

Wavelets decomposition:

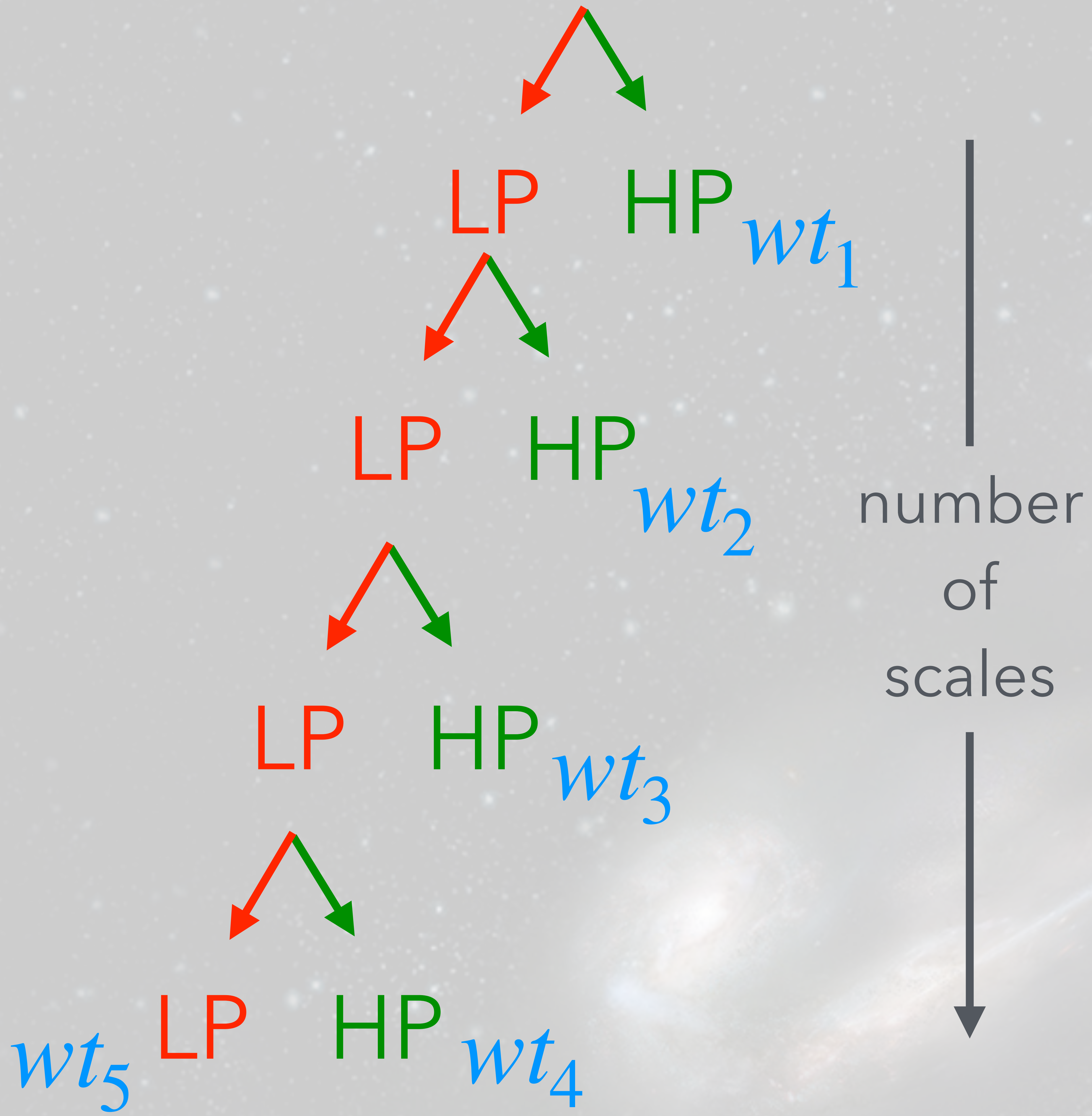
Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$



Illustration

Wavelets decomposition:

Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$



Illustration

Wavelets decomposition:

Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$

$*$

X

LP HP wt_1

LP HP wt_2

LP HP wt_3

wt_5 LP HP wt_4

number of scales

$*$ No humans were harmed during this experiment

Illustration

Wavelets decomposition:

Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$

LP HP wt_1

LP HP wt_2

LP HP wt_3

wt_5 LP HP wt_4

number of scales

*

X

wt_1

* No humans were harmed during this experiment

Illustration

Wavelets decomposition:

Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$

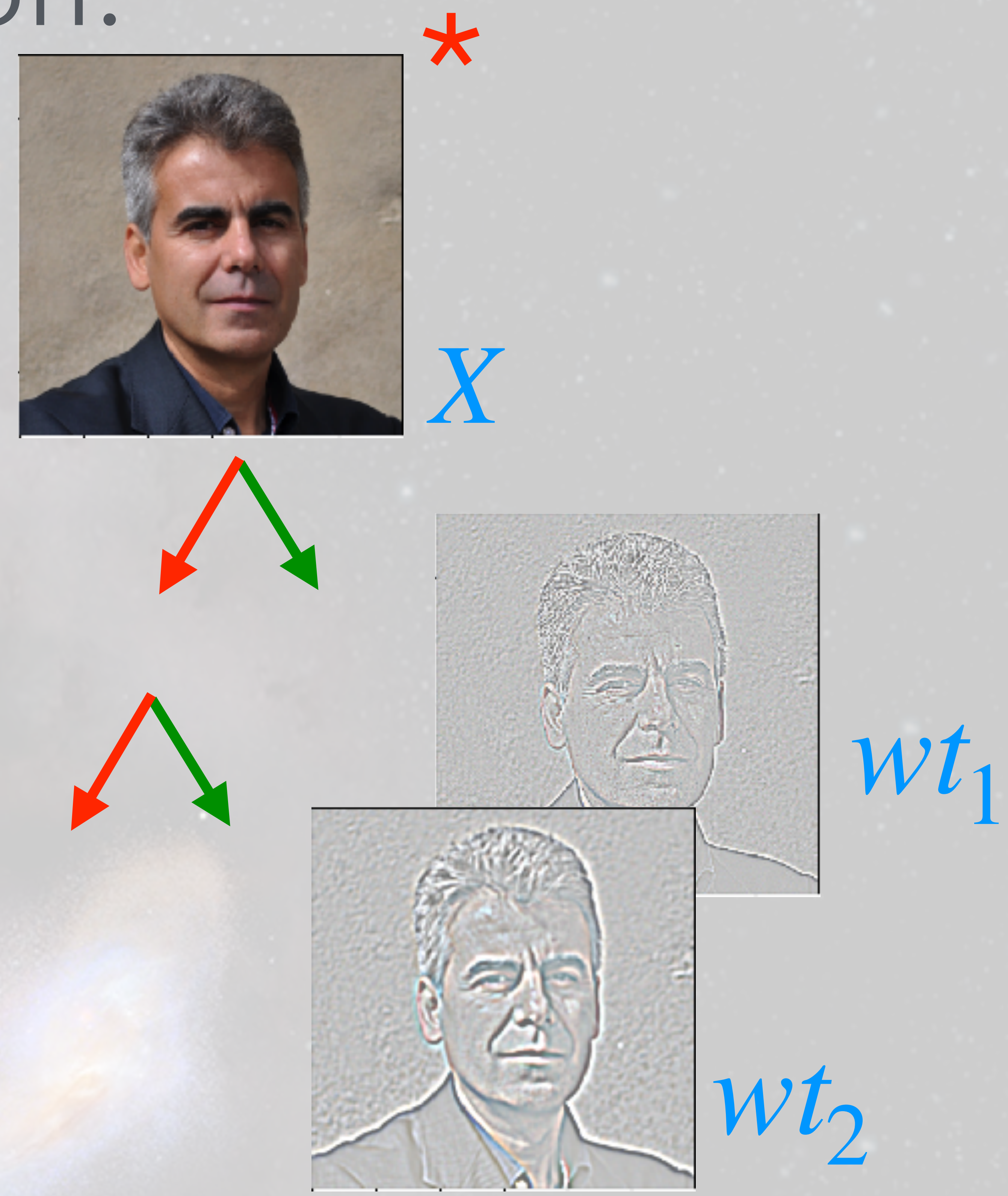
LP HP wt_1

LP HP wt_2

LP HP wt_3

wt_5 LP HP wt_4

number of scales



* No humans were harmed during this experiment

Illustration

Wavelets decomposition:

Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$

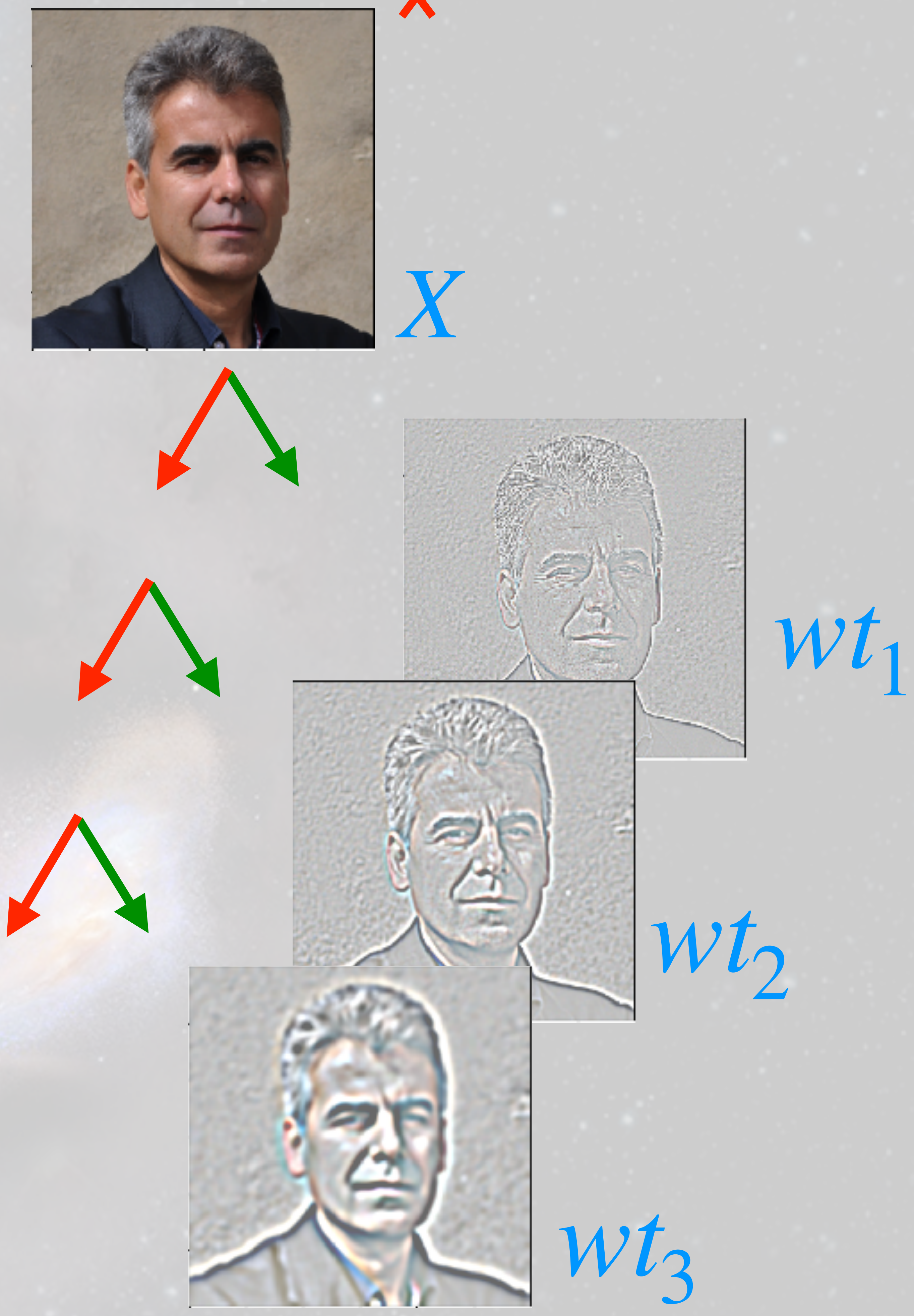
LP HP wt_1

LP HP wt_2

LP HP wt_3

wt_5 LP HP wt_4

number of scales



* No humans were harmed during this experiment

Illustration

Wavelets decomposition: *

Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$

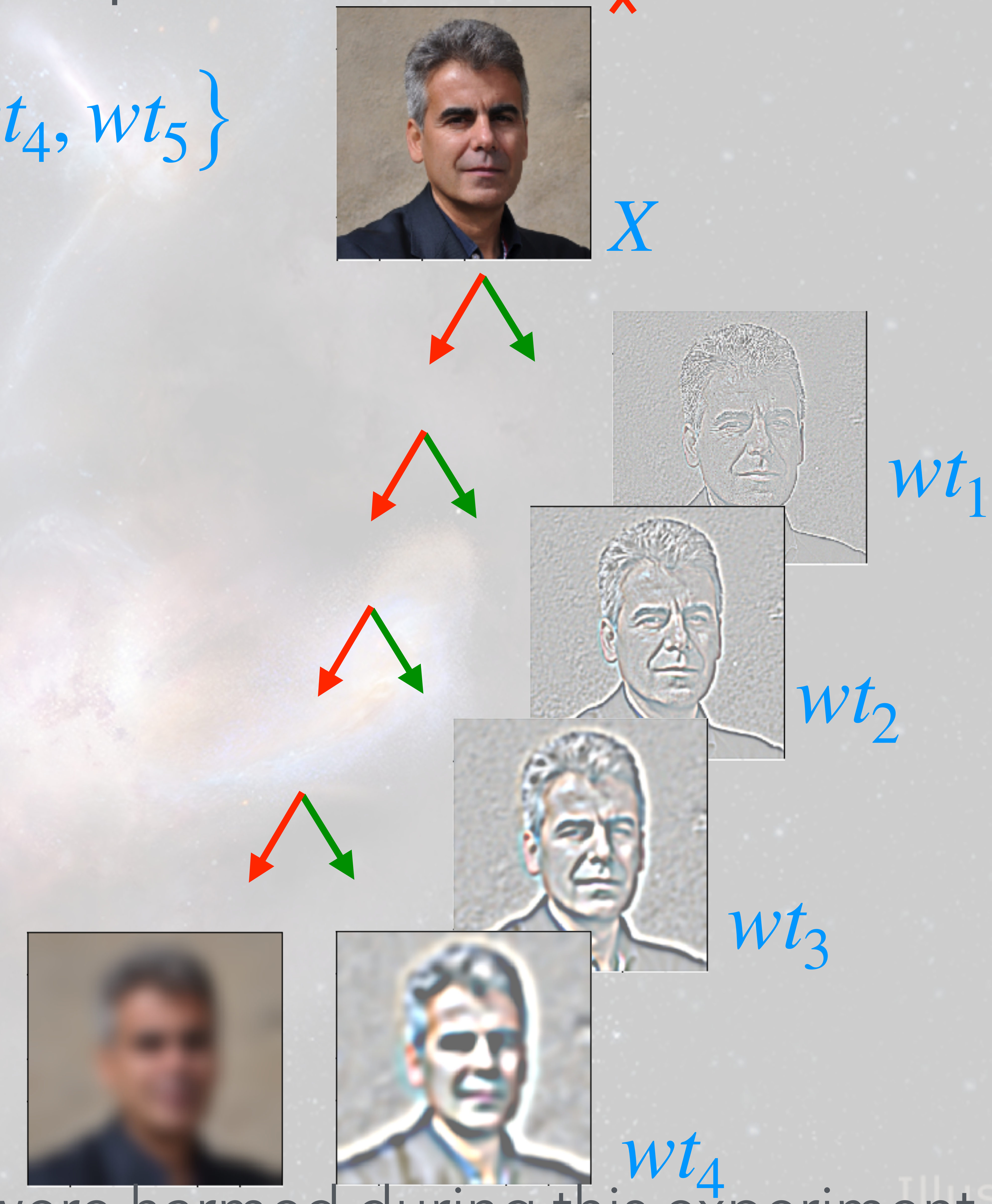
LP HP wt_1

LP HP wt_2

LP HP wt_3

wt_5 LP HP wt_4

number of scales



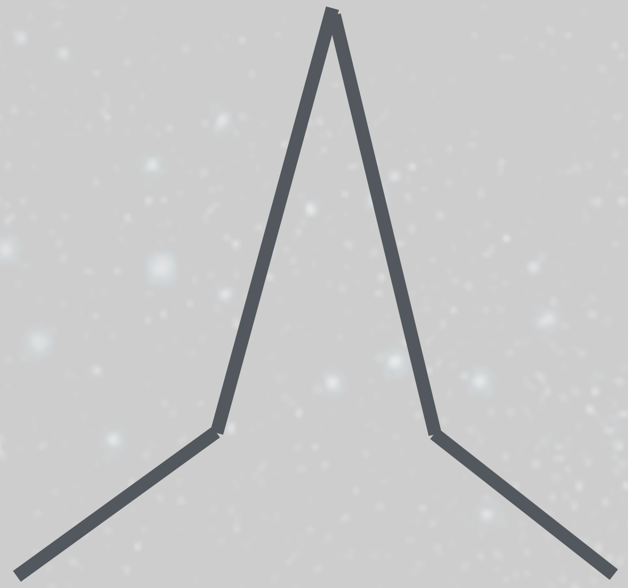
* No humans were harmed during this experiment

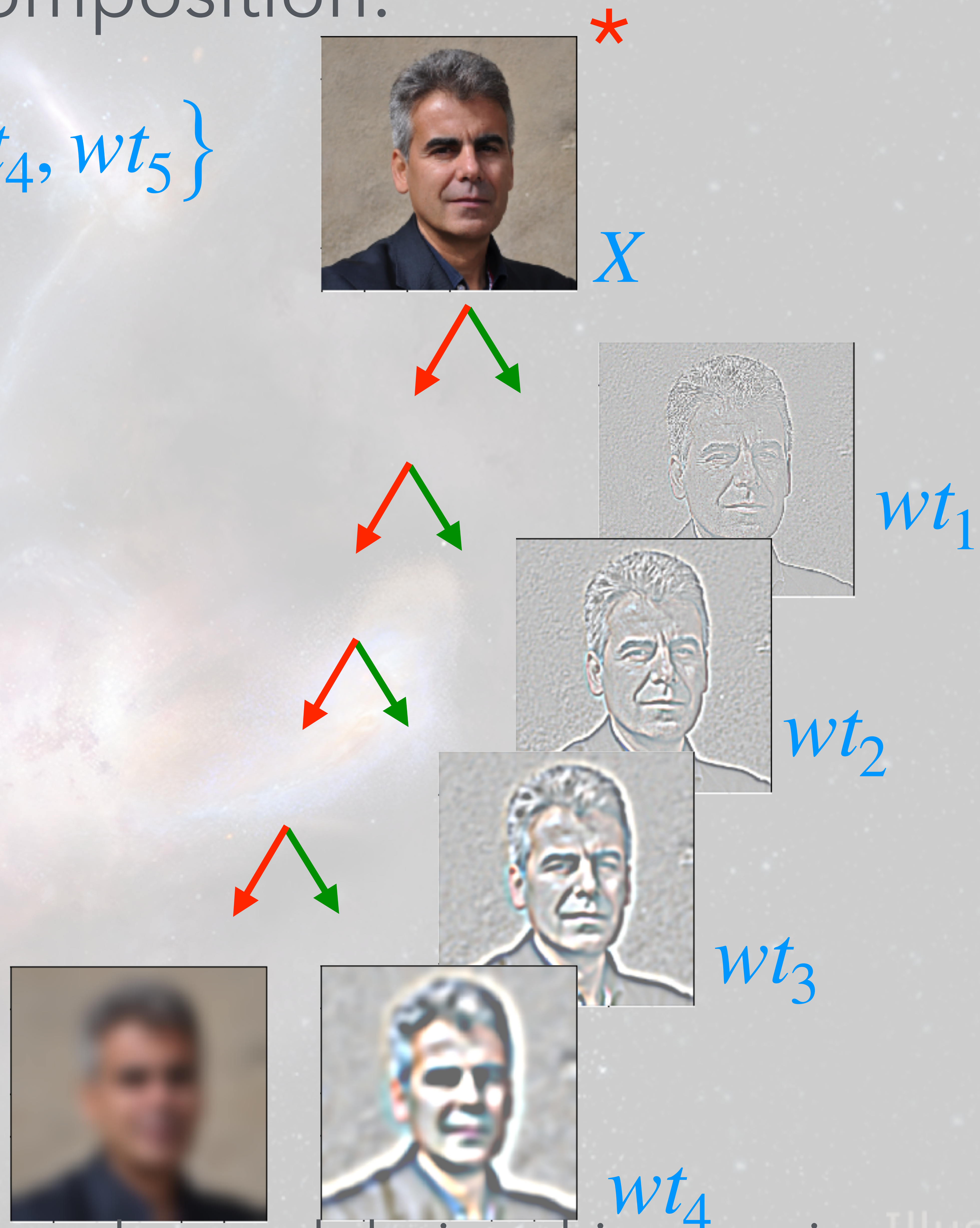
Illustration

Wavelets decomposition:

Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$

Example with starlet:





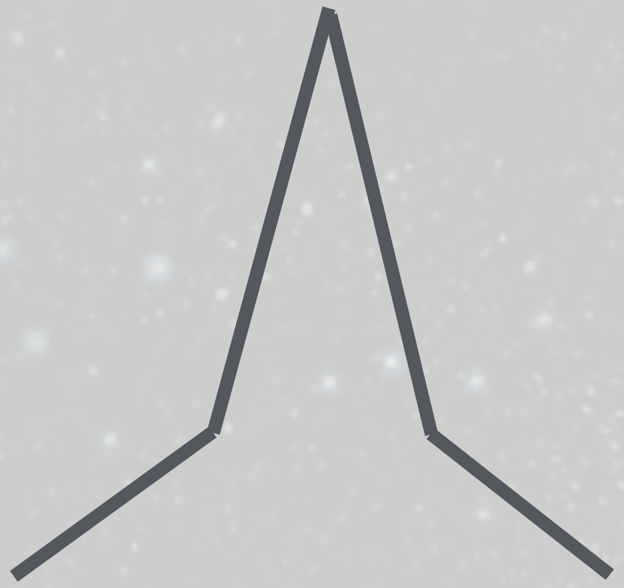
* No humans were harmed during this experiment

Illustration

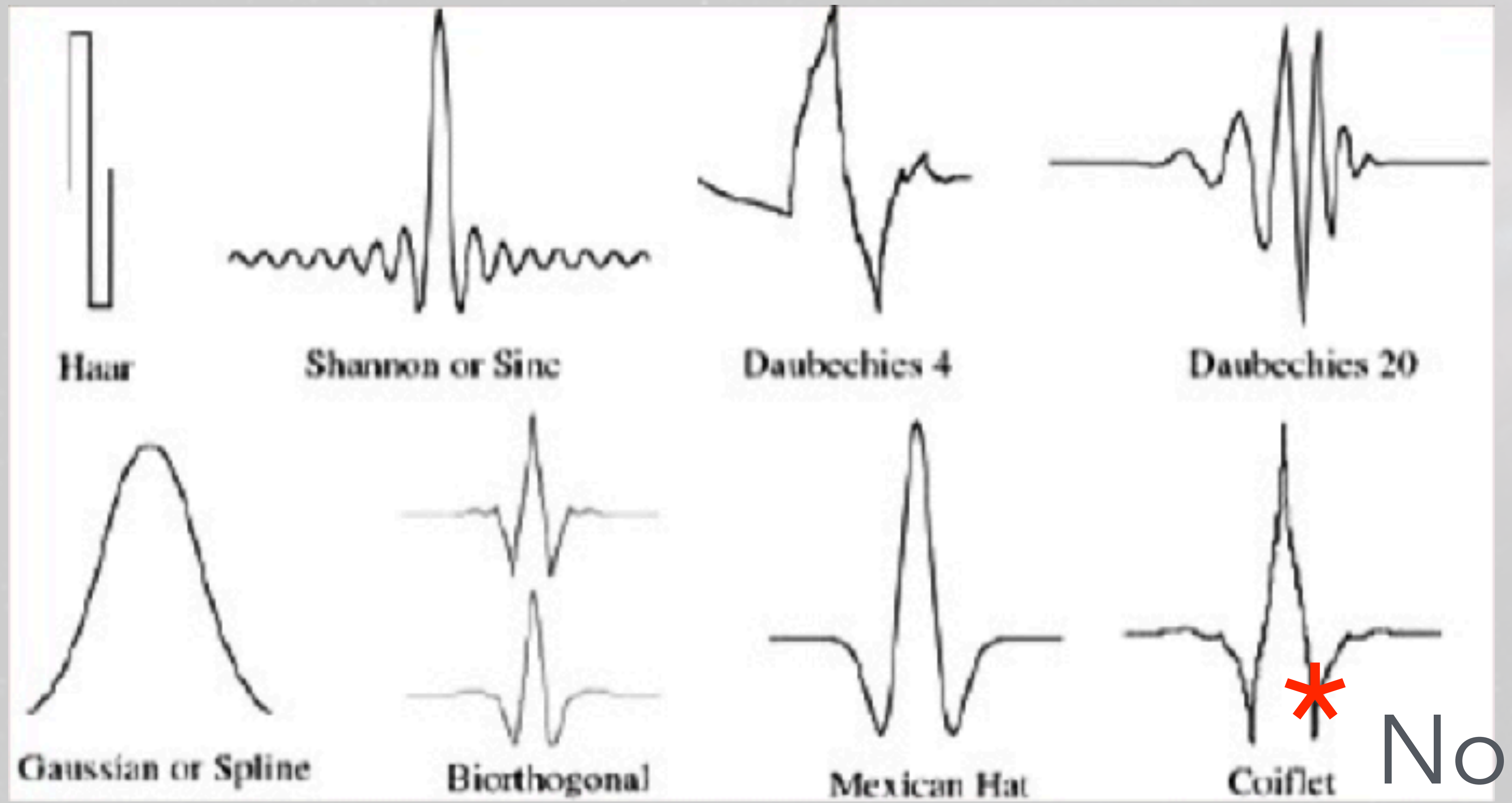
Wavelets decomposition:

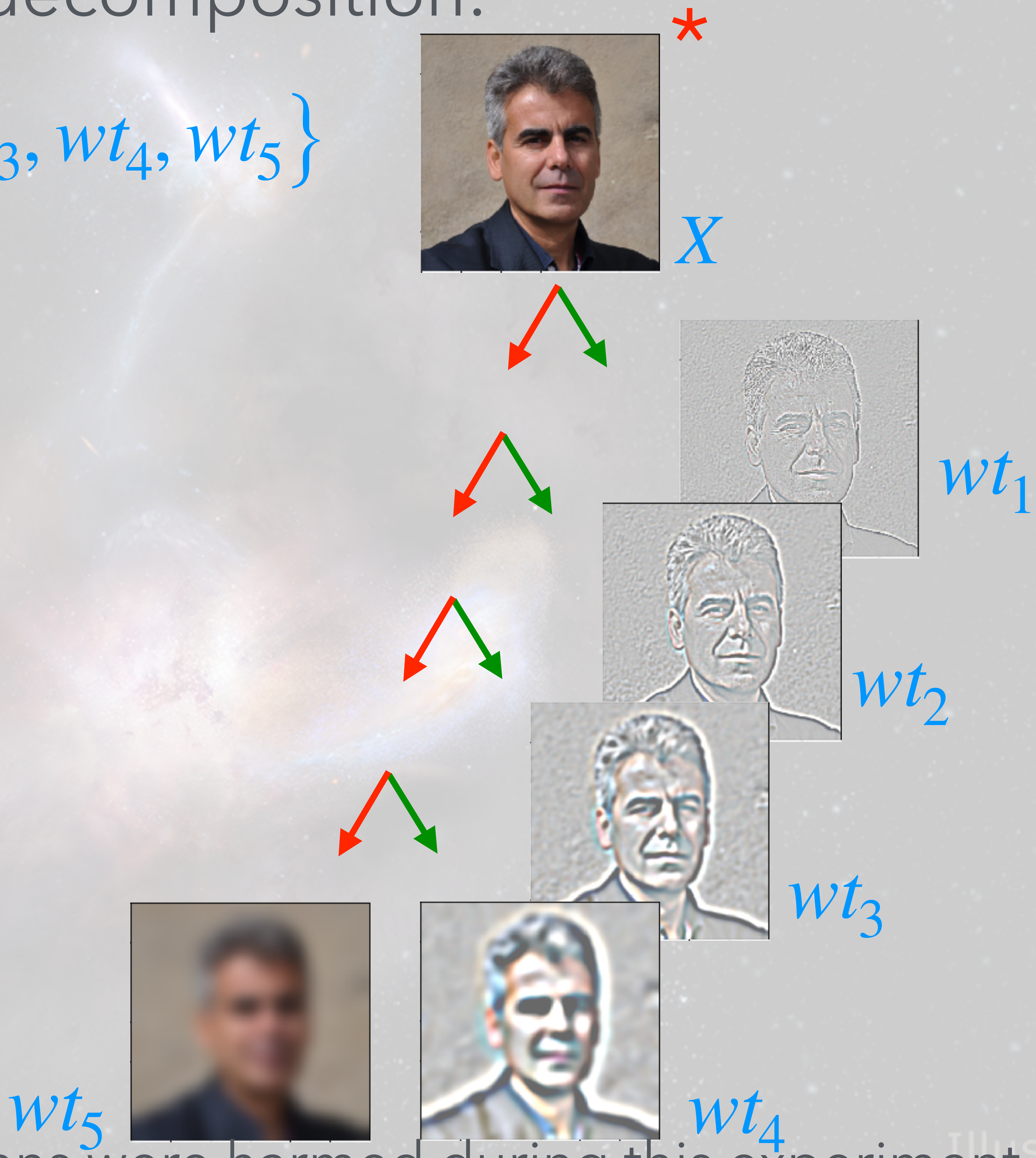
Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$

Example with starlet:



But also:





* No humans were harmed during this experiment

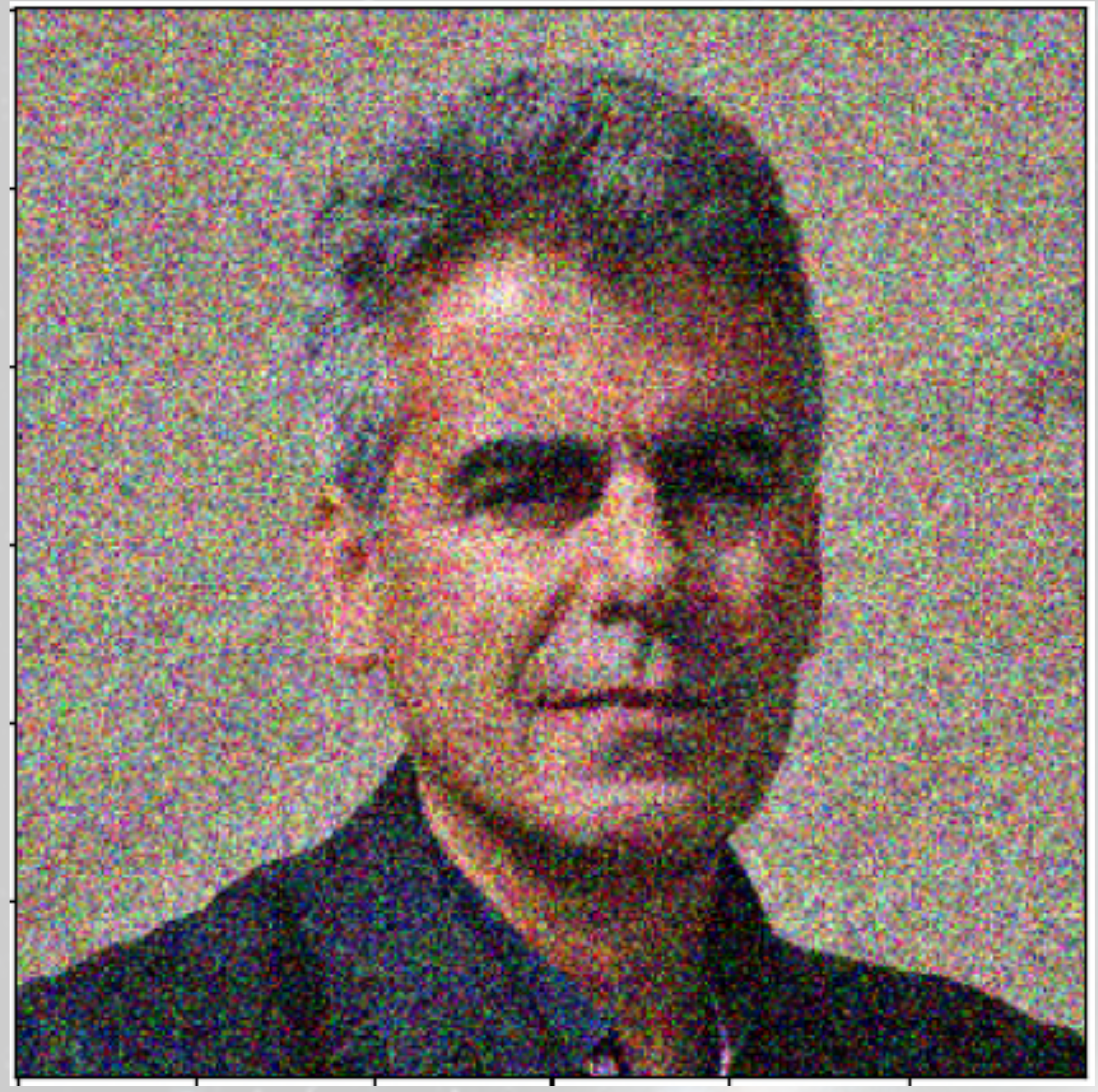
Illustration

Denoising with wavelets:

Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$

$\downarrow$

Denoising: Sparser



Illustration

Denoising with wavelets:

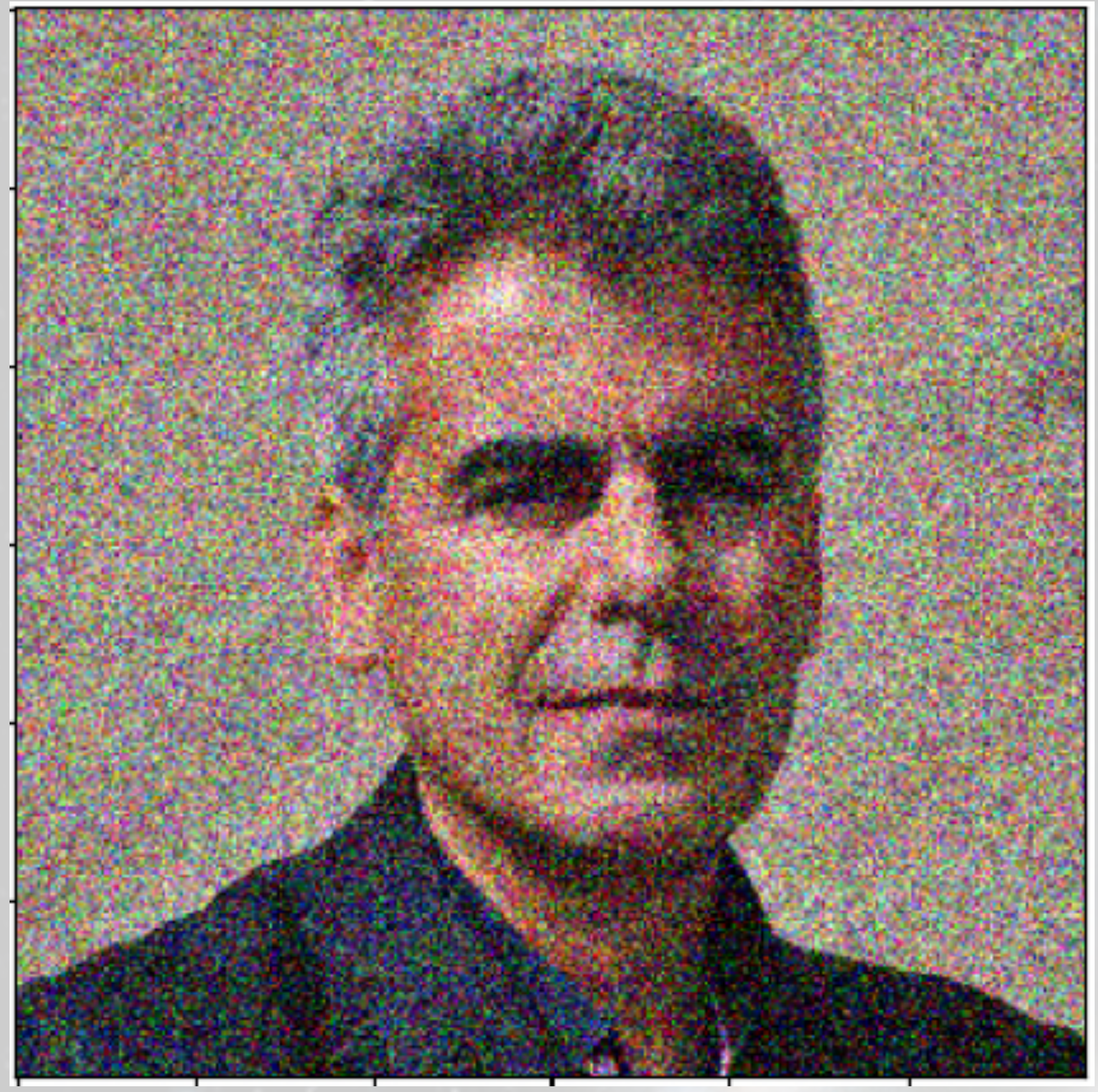
Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$

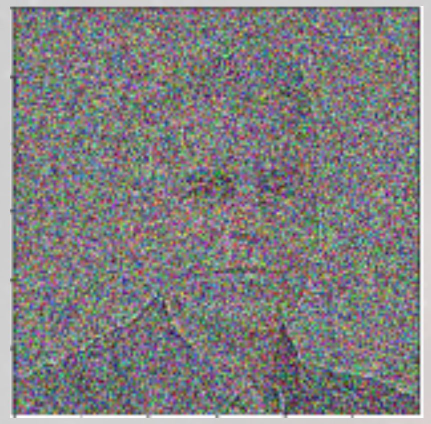
$\downarrow$

Denoising:

Sparser

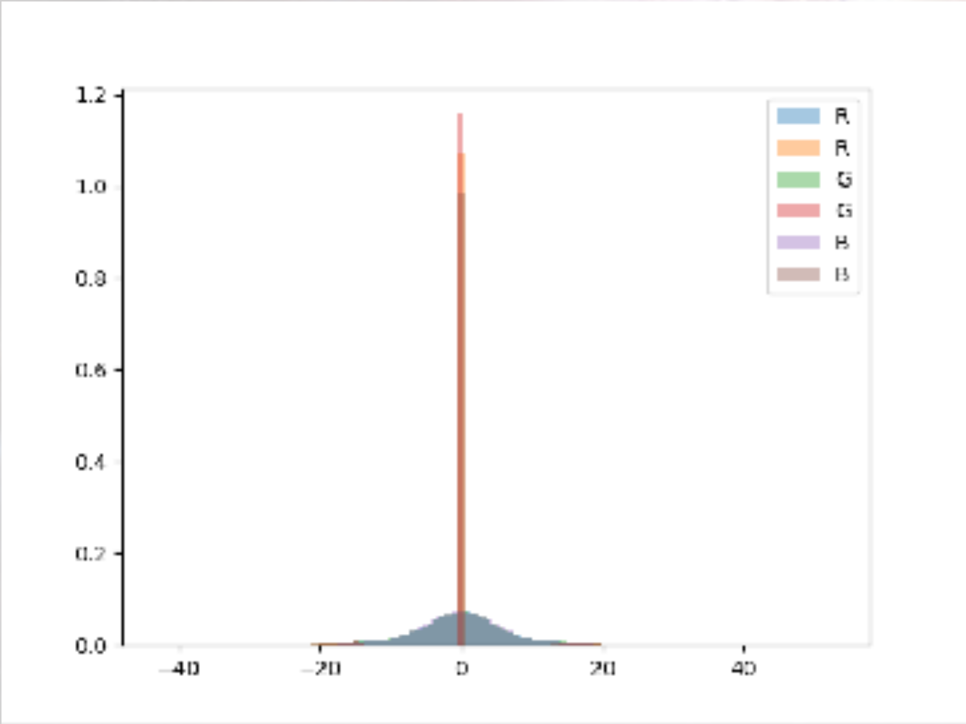
0-mean + stay gaussian





wt_1

$\longrightarrow$



for all wt_i

Illustration

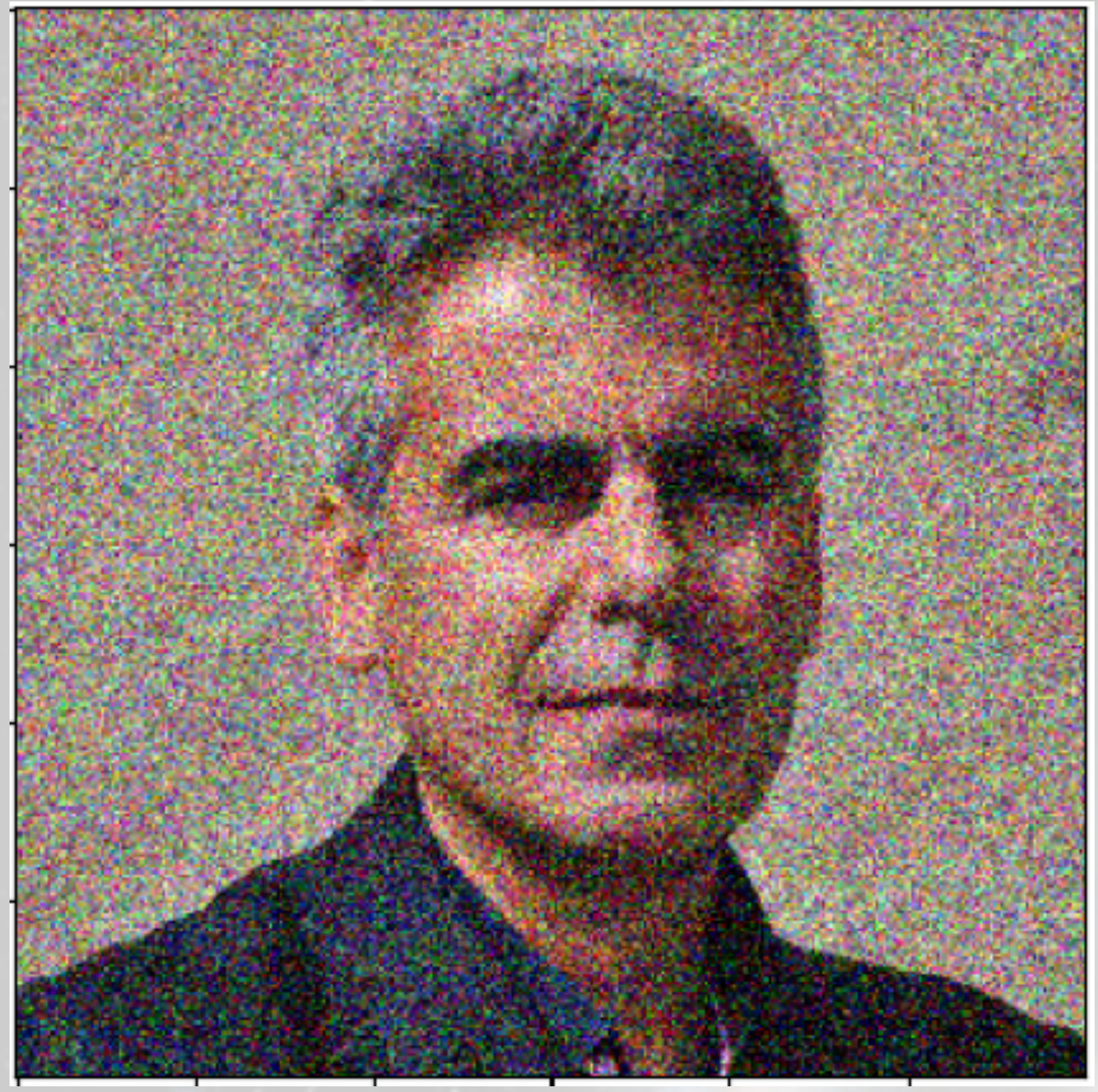
Denoising with wavelets:

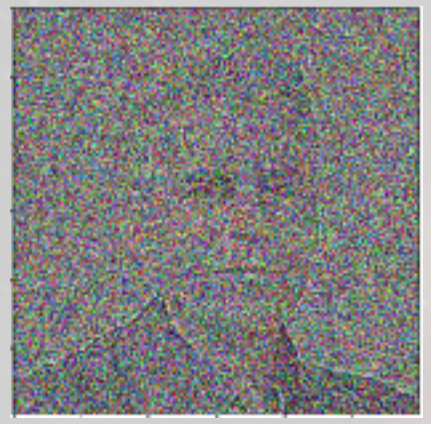
Data: $X \mapsto \{wt_1, wt_2, wt_3, wt_4, wt_5\}$

Sparser

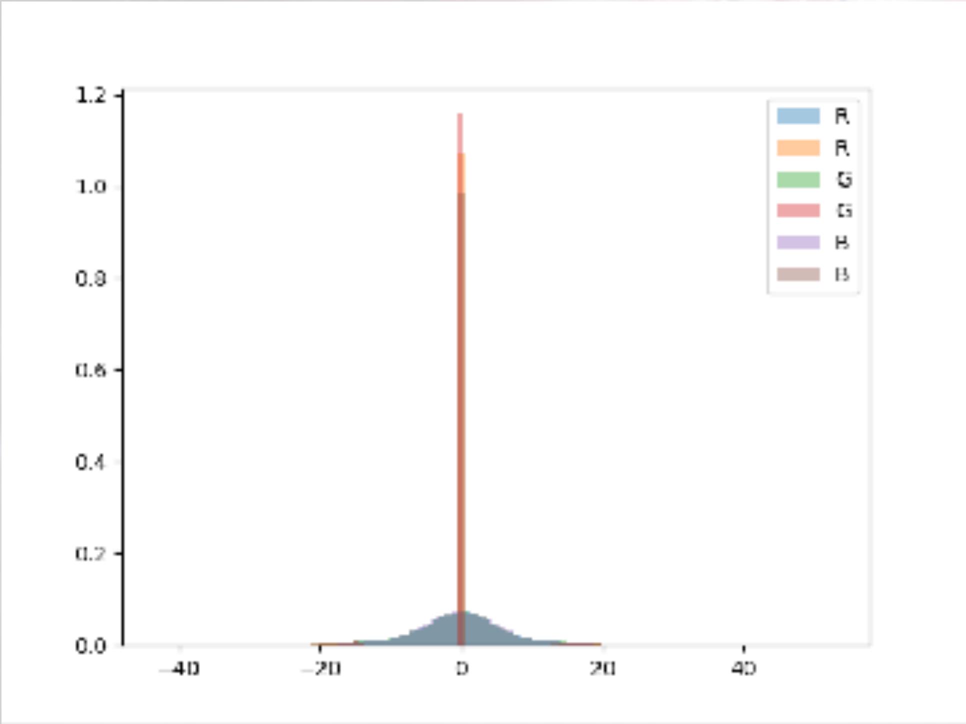
Denoising:

0-mean + stay gaussian





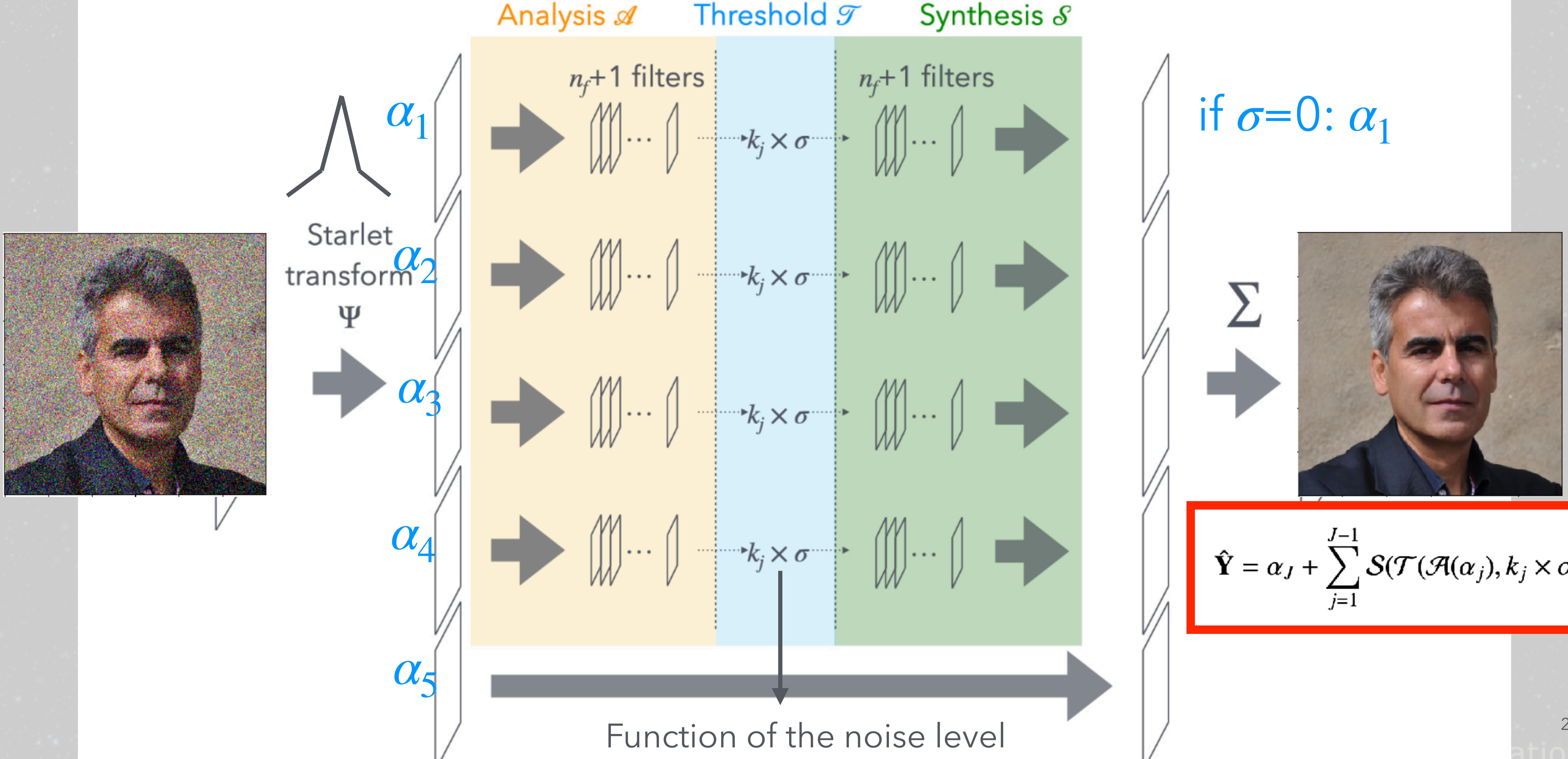
wt_1



for all wt_i

Illustration

Learnlet network architecture



$$\hat{\mathbf{Y}} = \alpha_J + \sum_{j=1}^{J-1} \mathcal{S}(\mathcal{T}(\mathcal{A}(\alpha_j), k_j \times \sigma)).$$

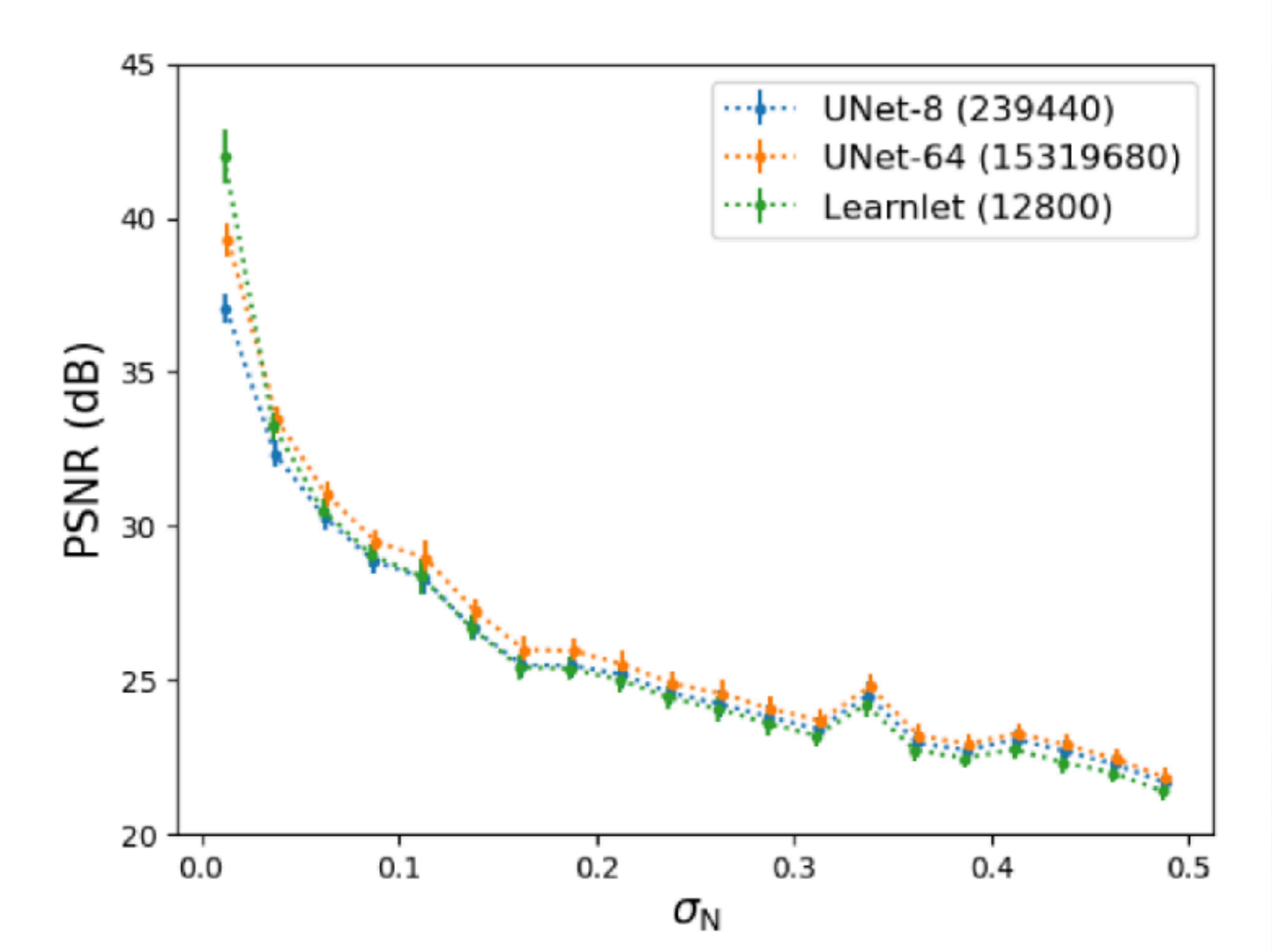
Training learnlets

10.000 images from ImageNet:
8.000 training
1.000 validation
1.000 test

Illustration

Training learnlets

10.000 images from ImageNet:
8.000 training
1.000 validation
1.000 test



Performance of networks on the test set

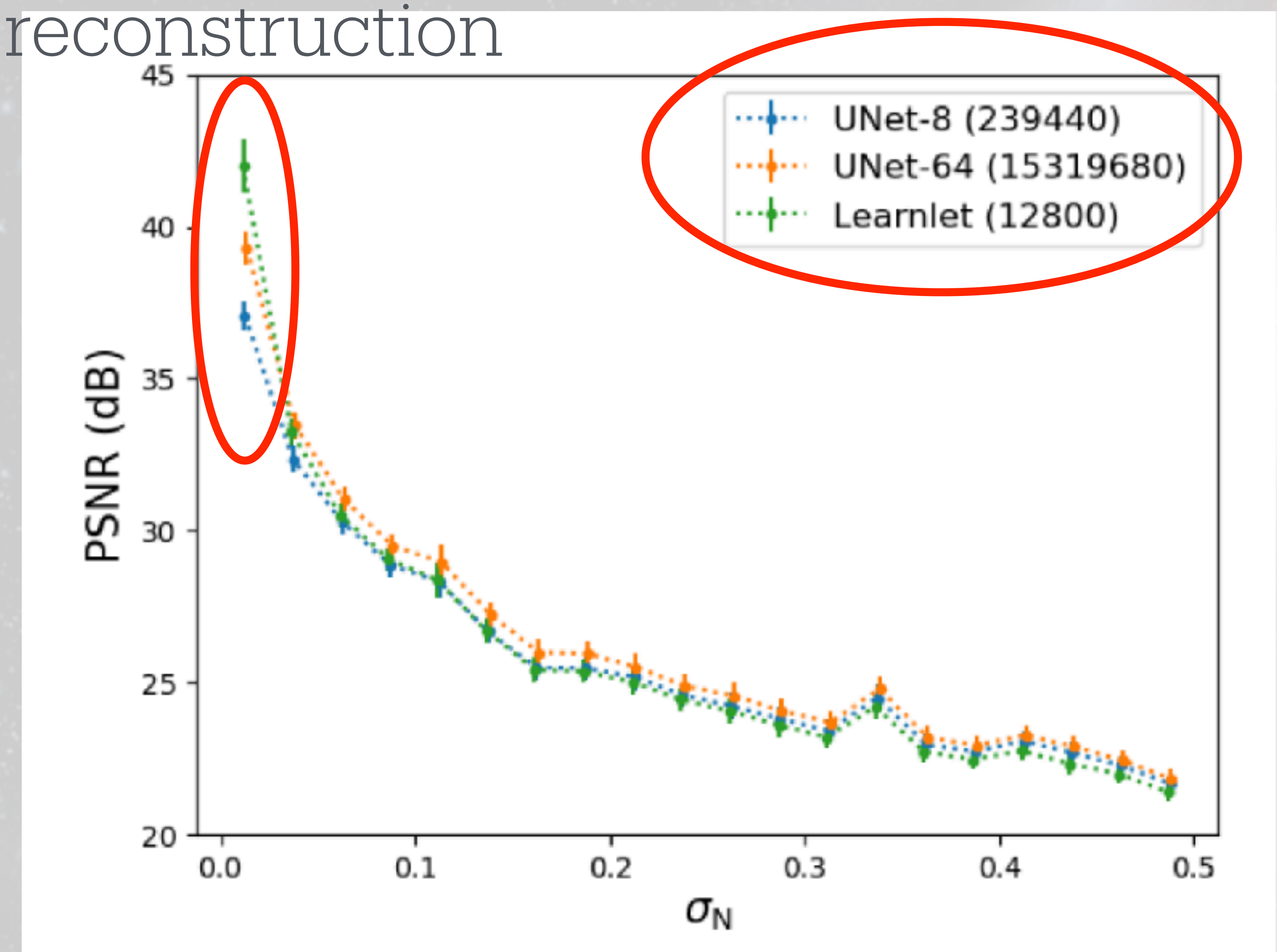
Illustration

Training learnlets

10.000 images from ImageNet:
8.000 training
1.000 validation
1.000 test

Significant lower number of free parameters

Exact reconstruction



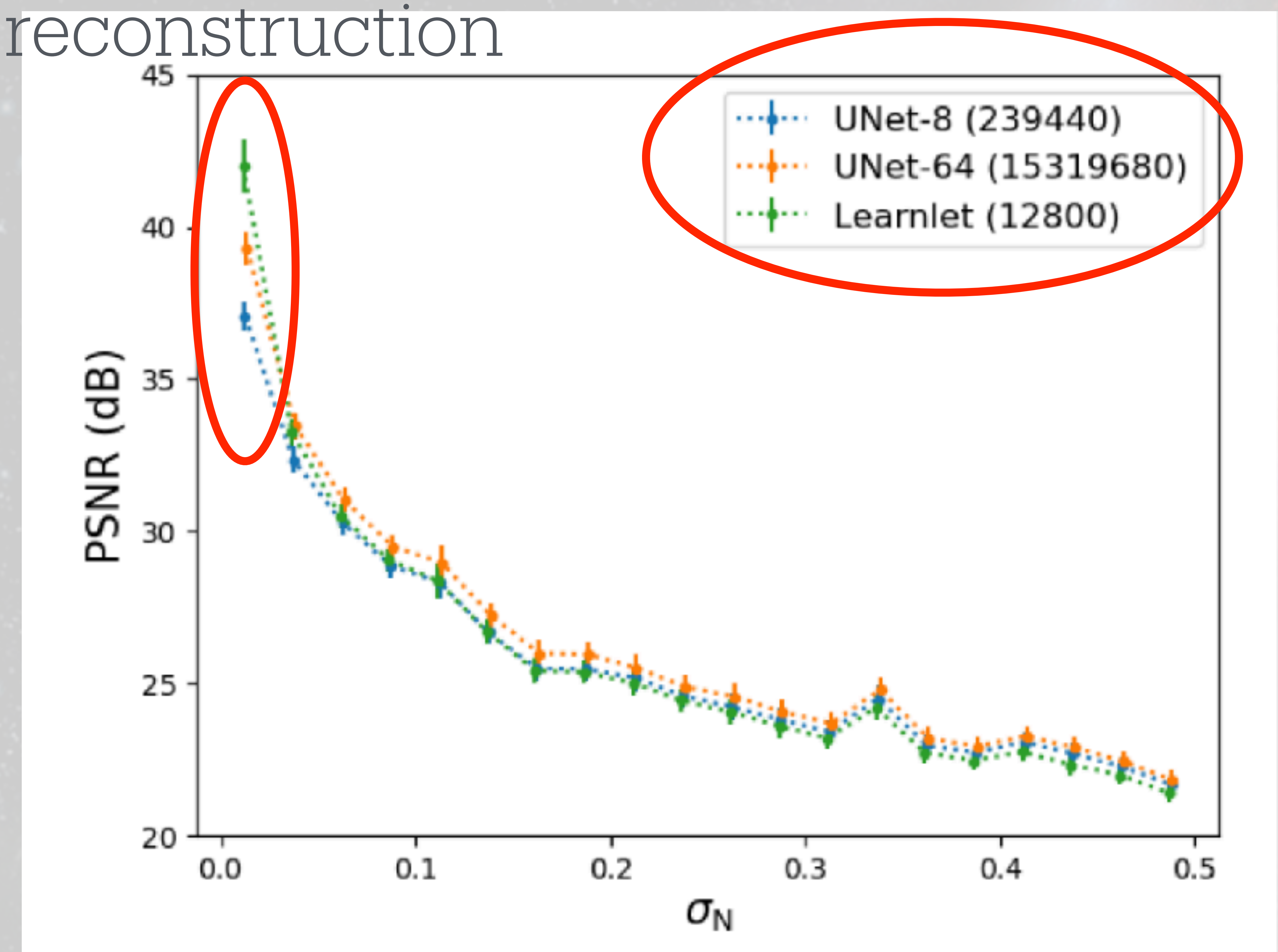
Performance of networks on the test set

Illustration

Training learnlets

10.000 images from ImageNet:
8.000 training
1.000 validation
1.000 test

Exact reconstruction



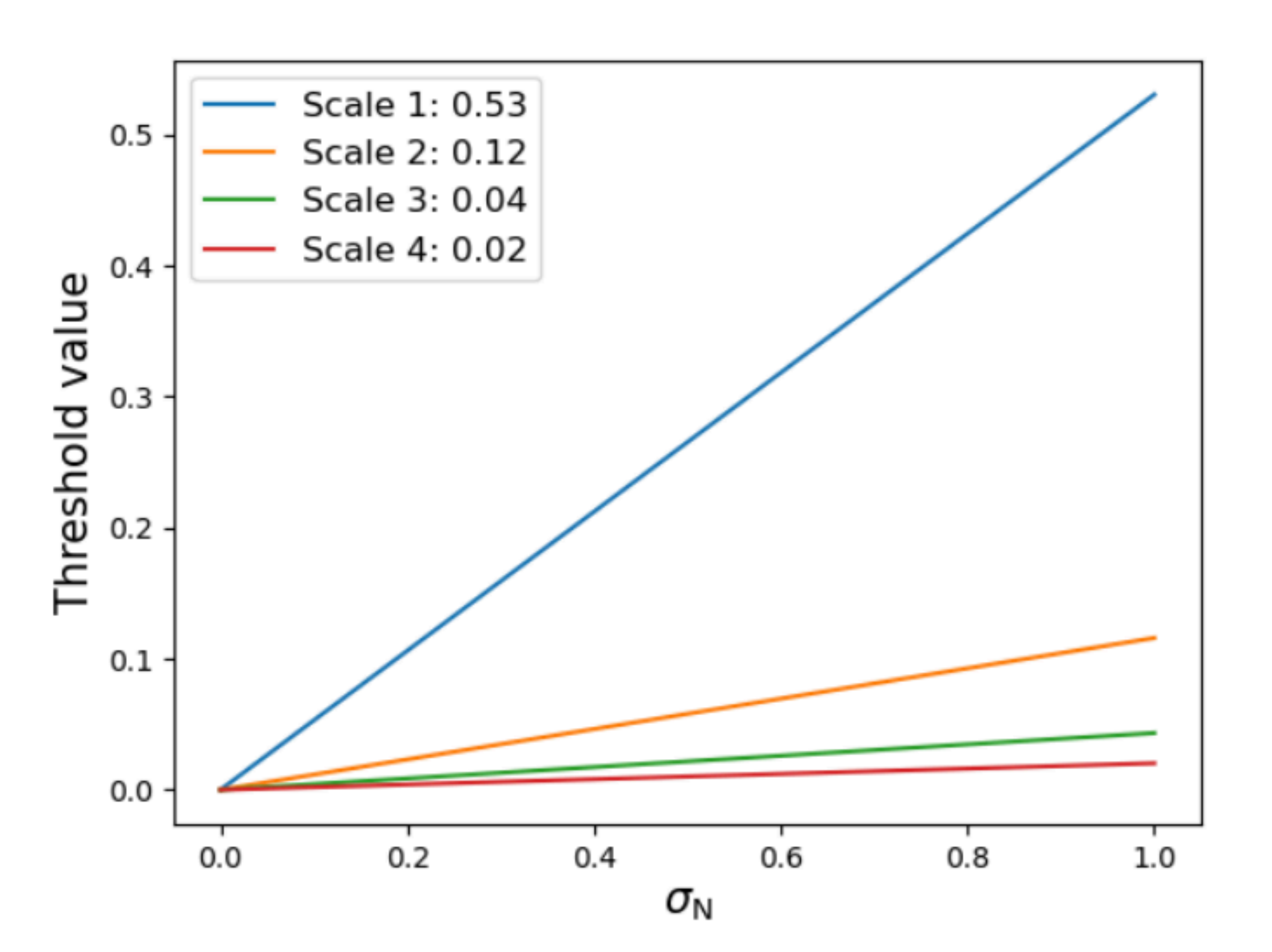
Significant lower number of free parameters

Performance of networks on the test set

GitHub:
https://github.com/vicbonj/learnlet.git
(PyTorch, pre-trained loaded weights)

Training learnlets

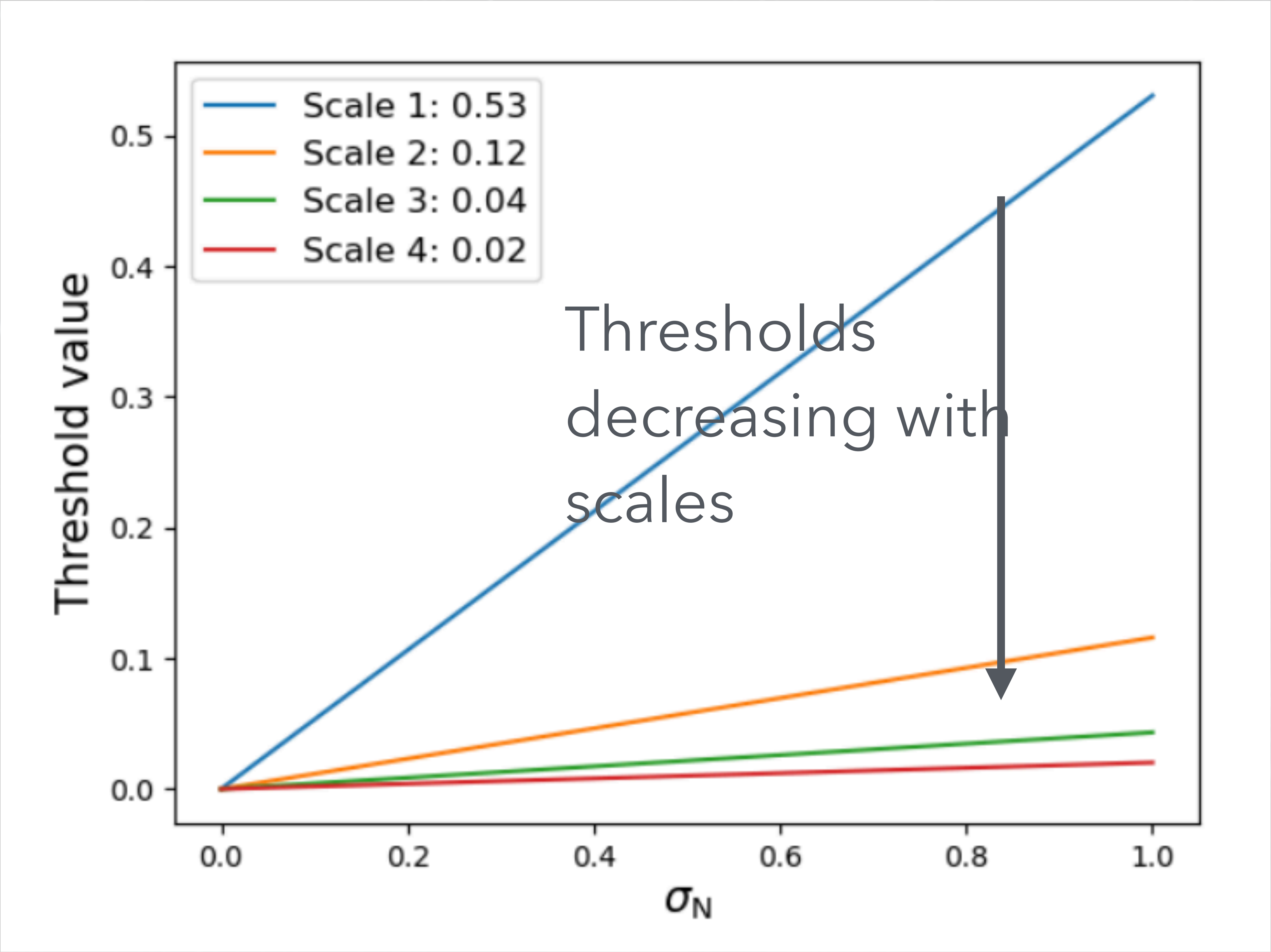
Thresholds k_j learned



Illustration

Training learnlets

Thresholds k_j learned

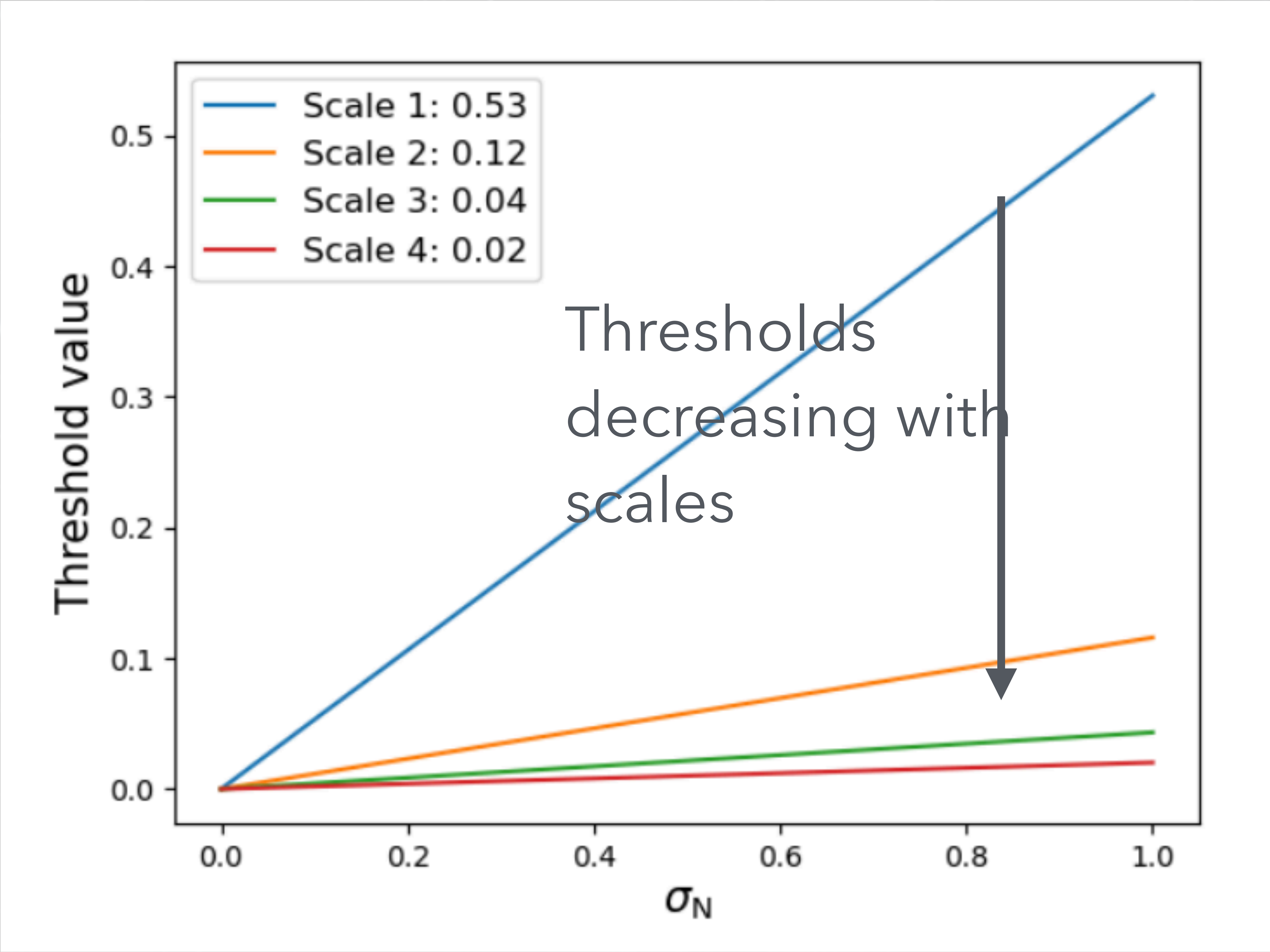


Thresholds decreasing with scales

Illustration

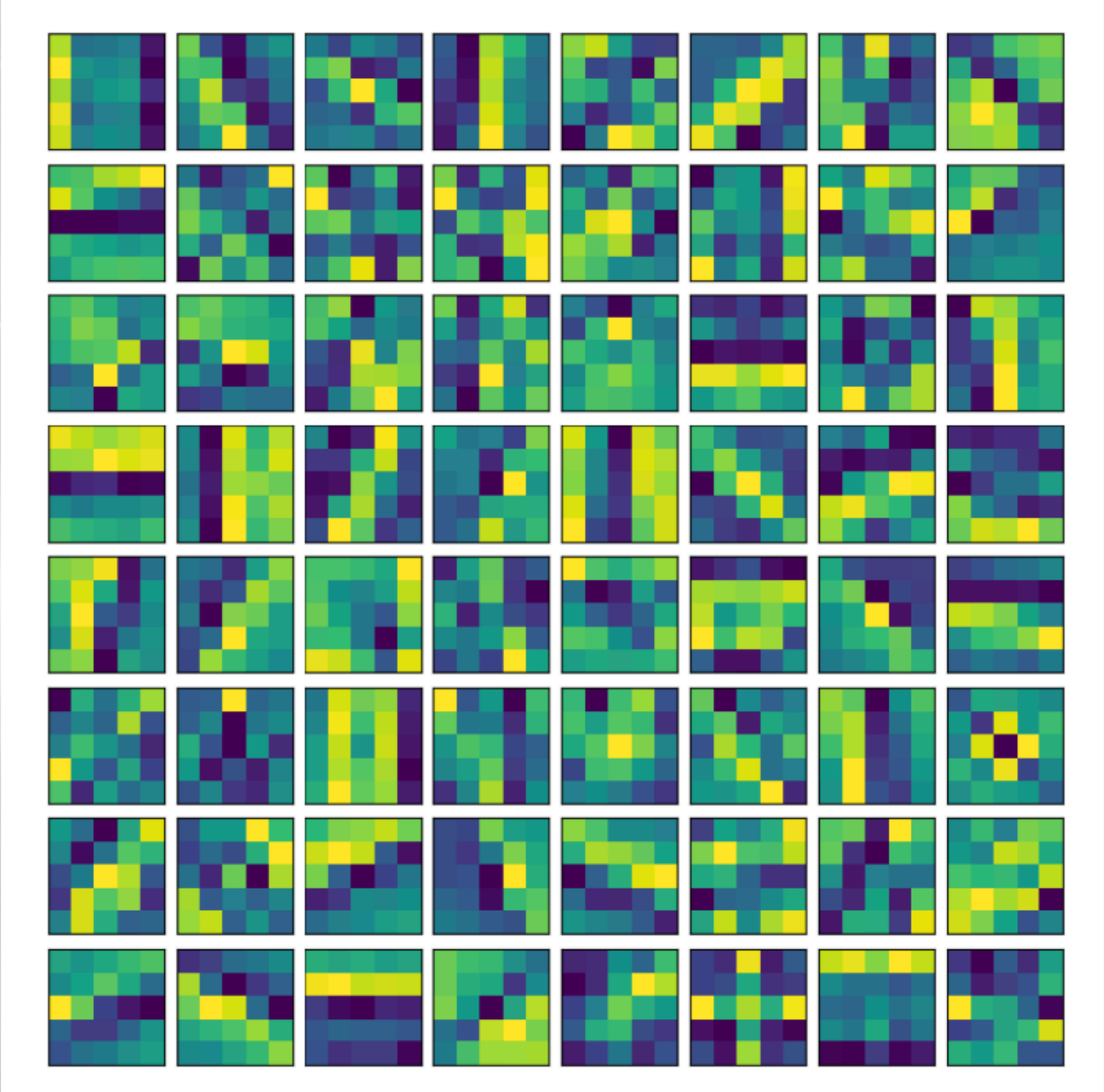
Training learnlets

Thresholds k_j learned



Thresholds decreasing with scales

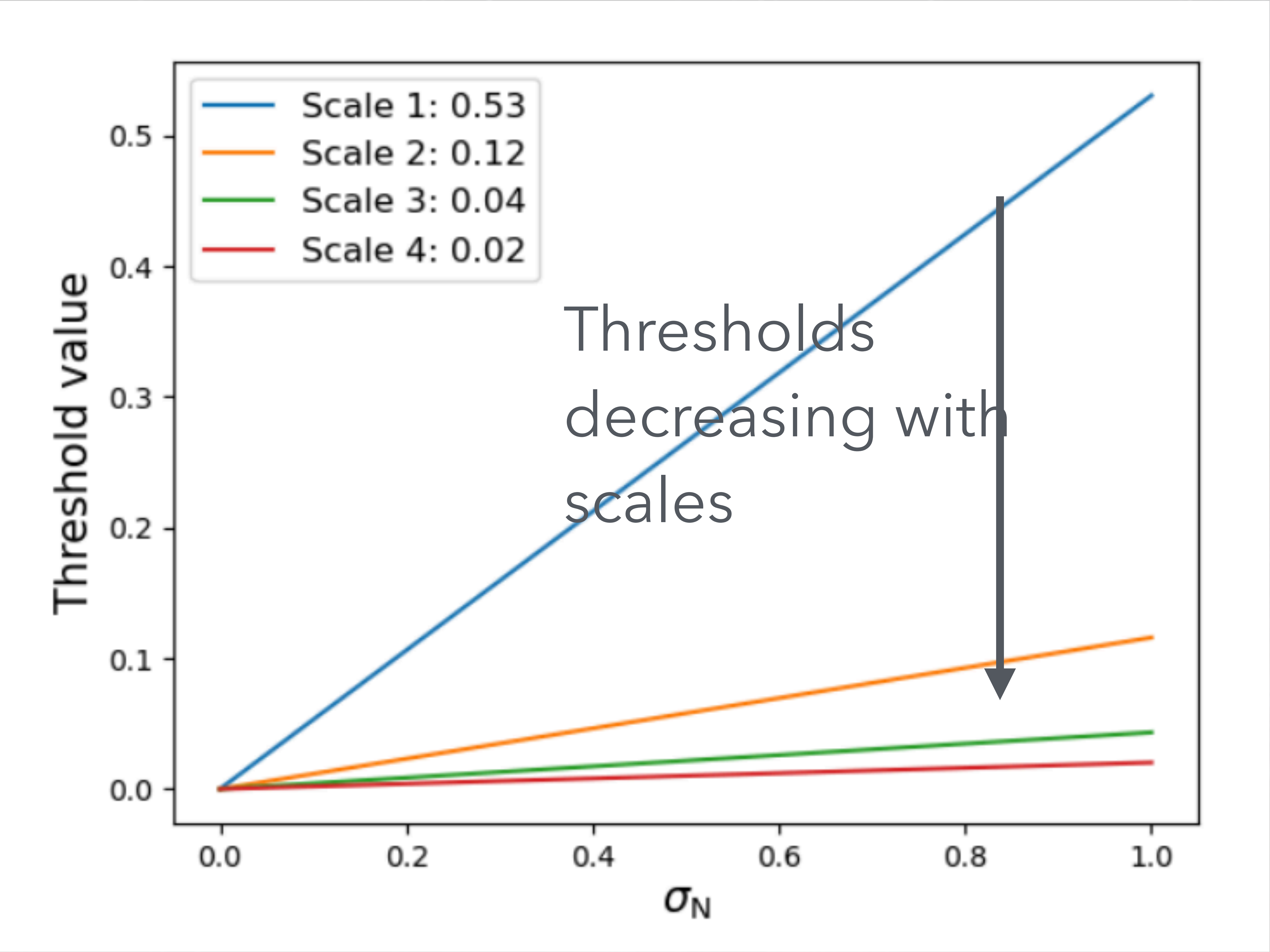
1st scale filters learned



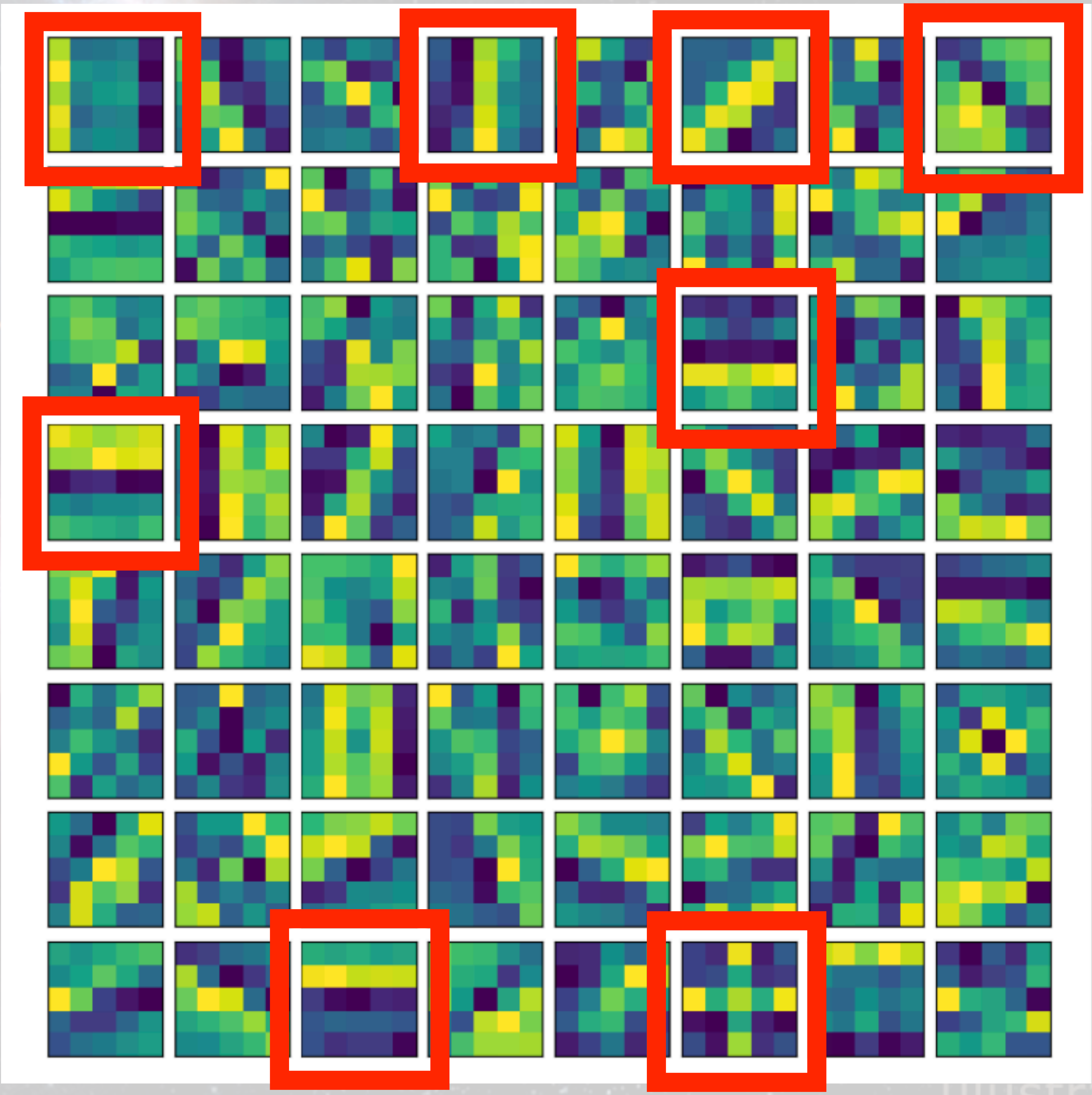
Illustration

Training learnlets

Thresholds k_j learned



1st scale filters learned



Illustration

Learnlet Component Separator (LCS)

BSS: $\mathbf{Y} = \mathbf{A} \cdot \mathbf{S} + \mathbf{N}.$

Illustration

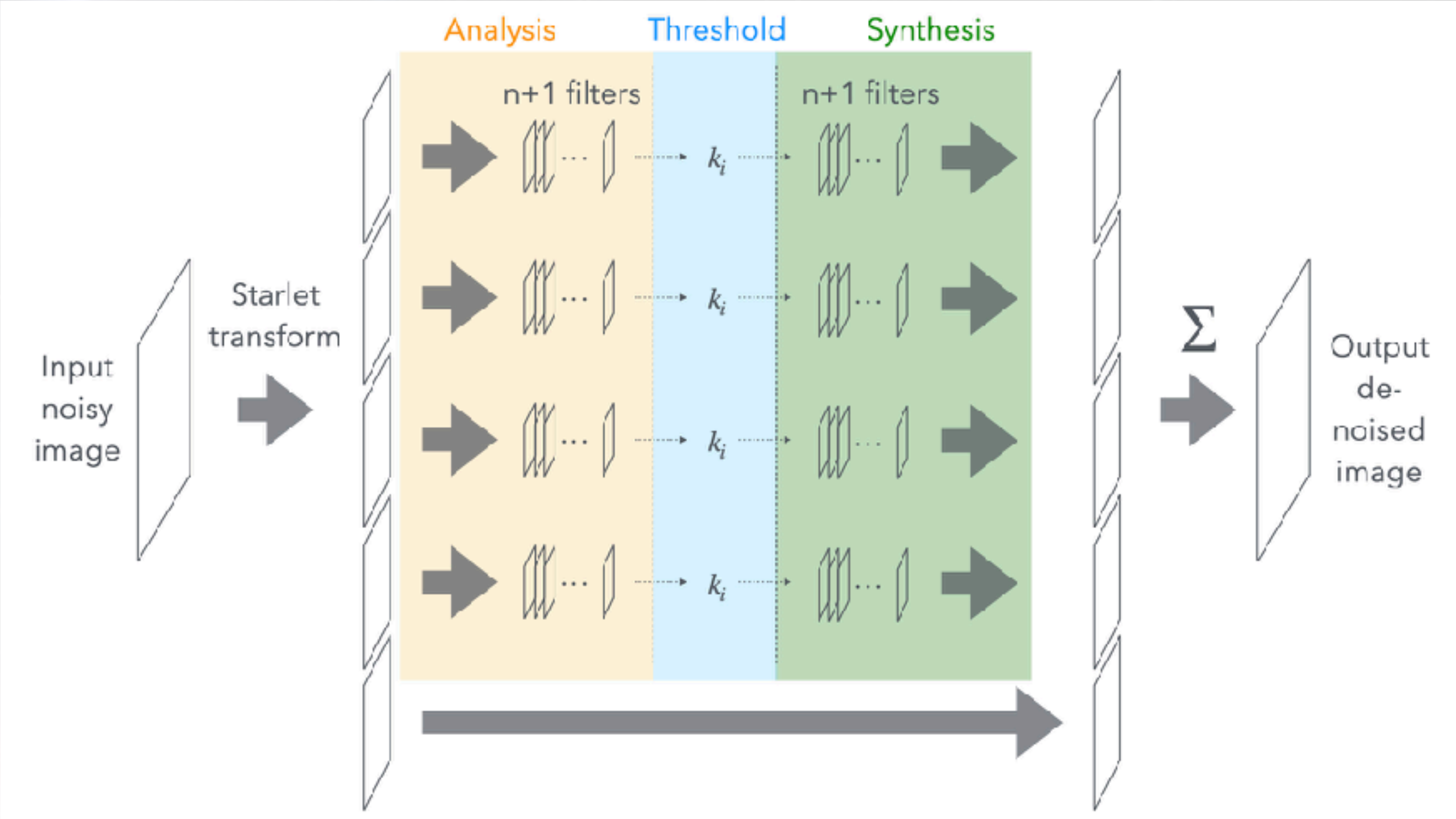
Learnlet Component Separator (LCS)

BSS: $\mathbf{Y} = \mathbf{A} \cdot \mathbf{S} + \mathbf{N}.$

Two steps iterating algorithm (inspired by GMCA):

1) $\min_{\mathbf{S}} \|\mathbf{Y} - \hat{\mathbf{A}} \cdot \mathbf{S}\|_2^2 + \boxed{\lambda \|\mathcal{L}_{\mathbf{S}}\|_{\mathrm{p}}}$

2) $\min_{\mathbf{A}} \|\mathbf{Y}^{\mathrm{no-coarse}} - \mathbf{A} \cdot \boxed{\mathcal{L}_{\hat{\mathbf{S}}}{}^{\mathrm{no-coarse}}}\|_2^2$



$\mathscr{L}_{\mathbf{S}_i}$: Learnlets trained priorly on each components i

Illustration

Learnlet Component Separator (LCS)

BSS: $\mathbf{Y} = \mathbf{A} \cdot \mathbf{S} + \mathbf{N}.$

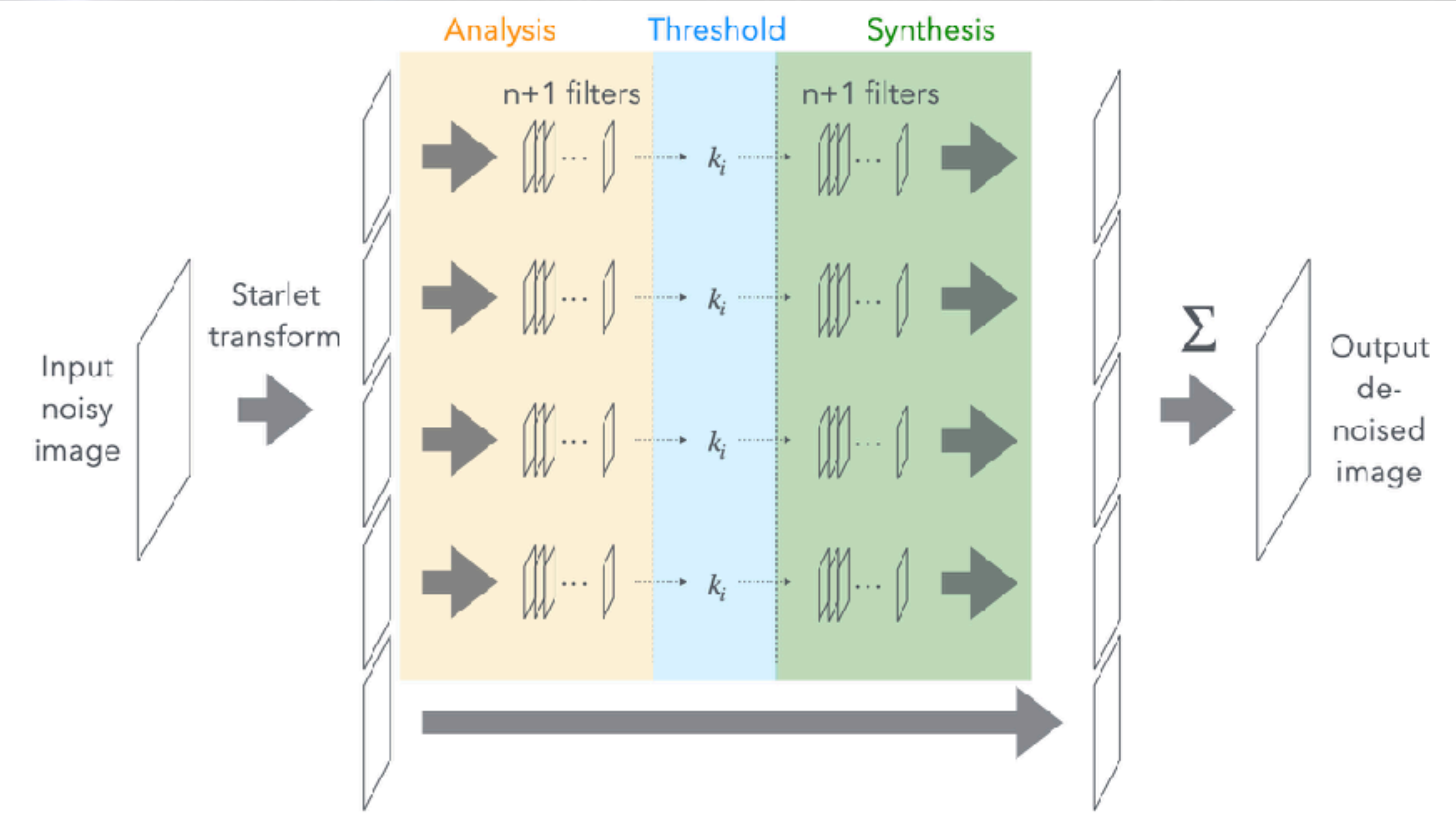
Two steps iterating algorithm (inspired by GMCA):

1) $\min_{\mathbf{S}} \|\mathbf{Y} - \hat{\mathbf{A}} \cdot \mathbf{S}\|_2^2 + \boxed{\lambda \|\mathcal{L}_{\mathbf{S}}\|_p}$

2) $\min_{\mathbf{A}} \|\mathbf{Y}^{\text{no-coarse}} - \mathbf{A} \cdot \boxed{\mathcal{L}_{\hat{\mathbf{S}}}^{\text{no-coarse}}} \|_2^2$

+ last step: Learnlet final denoising of S with threshold computed from the known noise on Y (expected to be known):

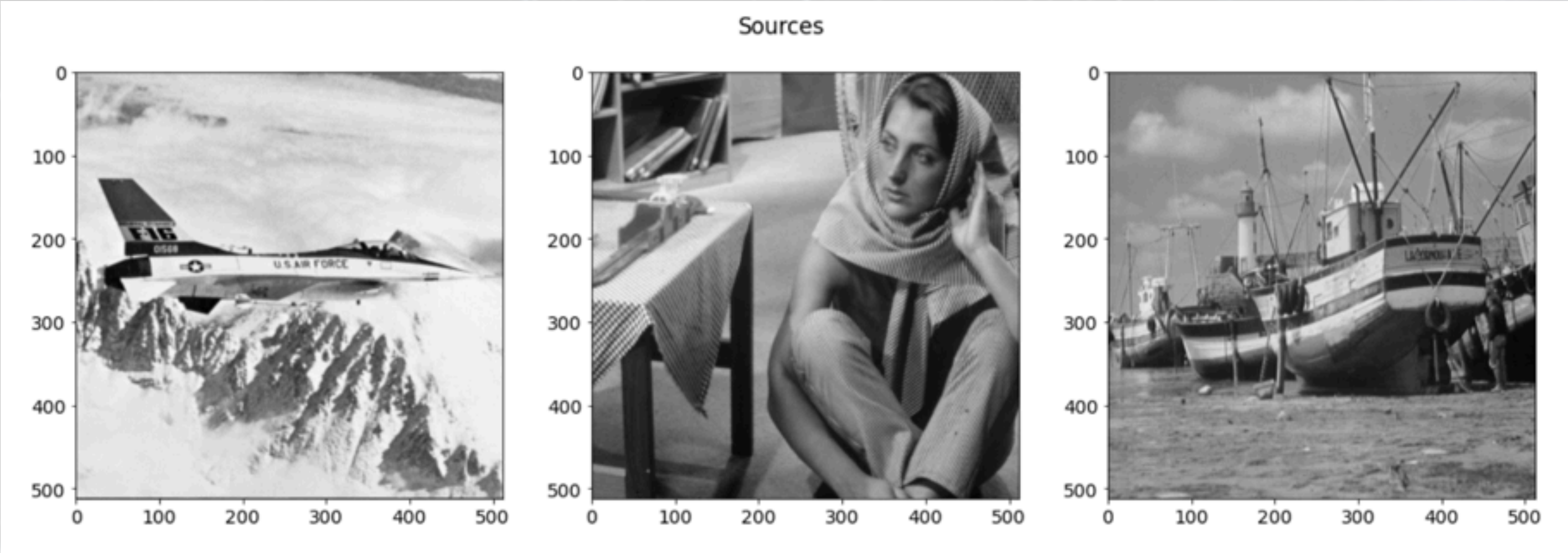
$\Sigma_{\hat{\mathbf{S}}} = \hat{\mathbf{A}}^{+} \Sigma_{\mathbf{N}} (\hat{\mathbf{A}}^{+})^{T}$



$\mathscr{L}_{\mathbf{S}_i}$: Learnlets trained priorly on each components i

Illustration

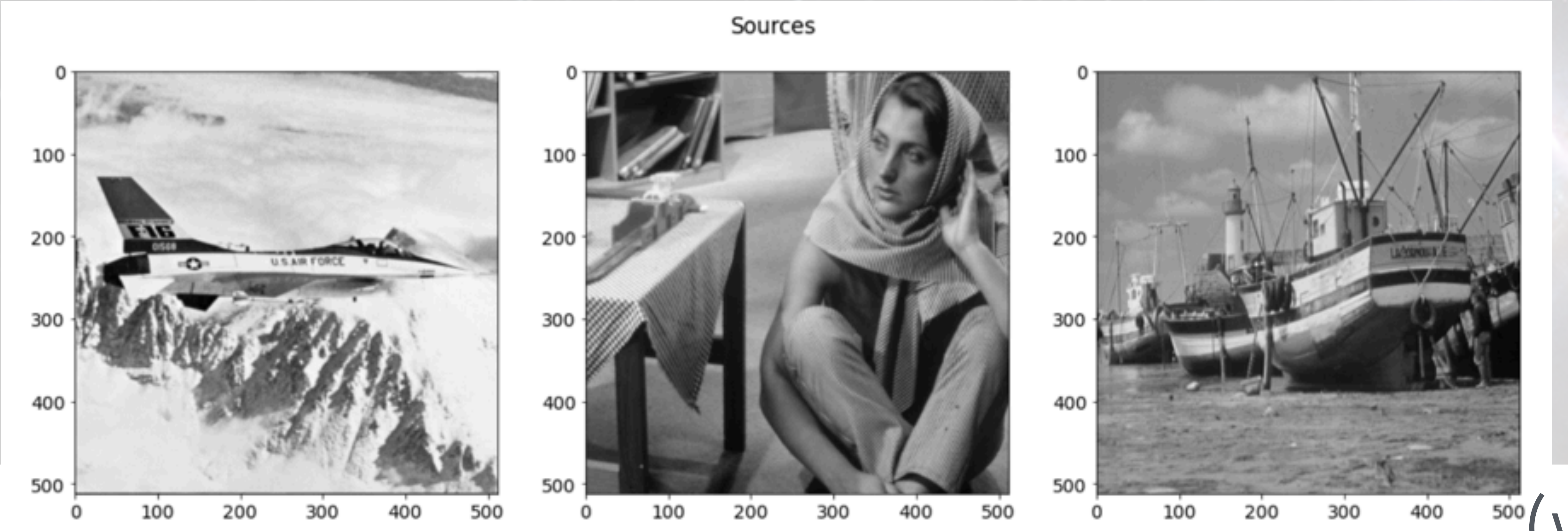
Results of LCS



Toy model with multiple realizations of A (6 channels) and N (gaussian white) with evolving σ

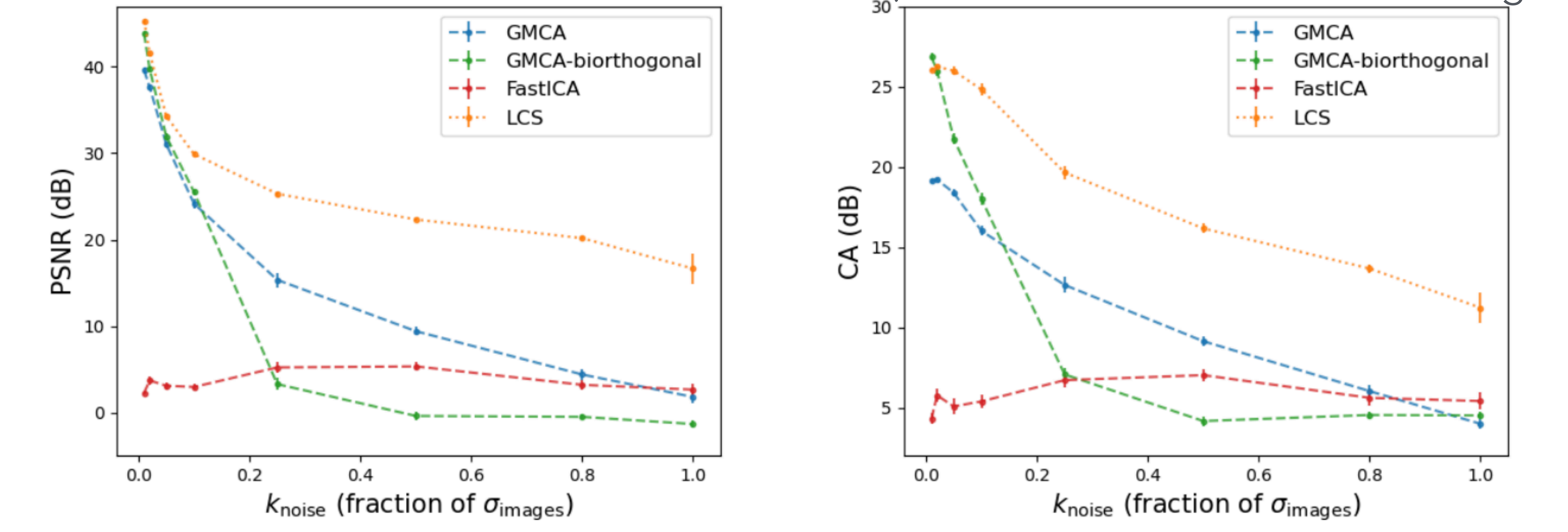
Illustration

Results of LCS

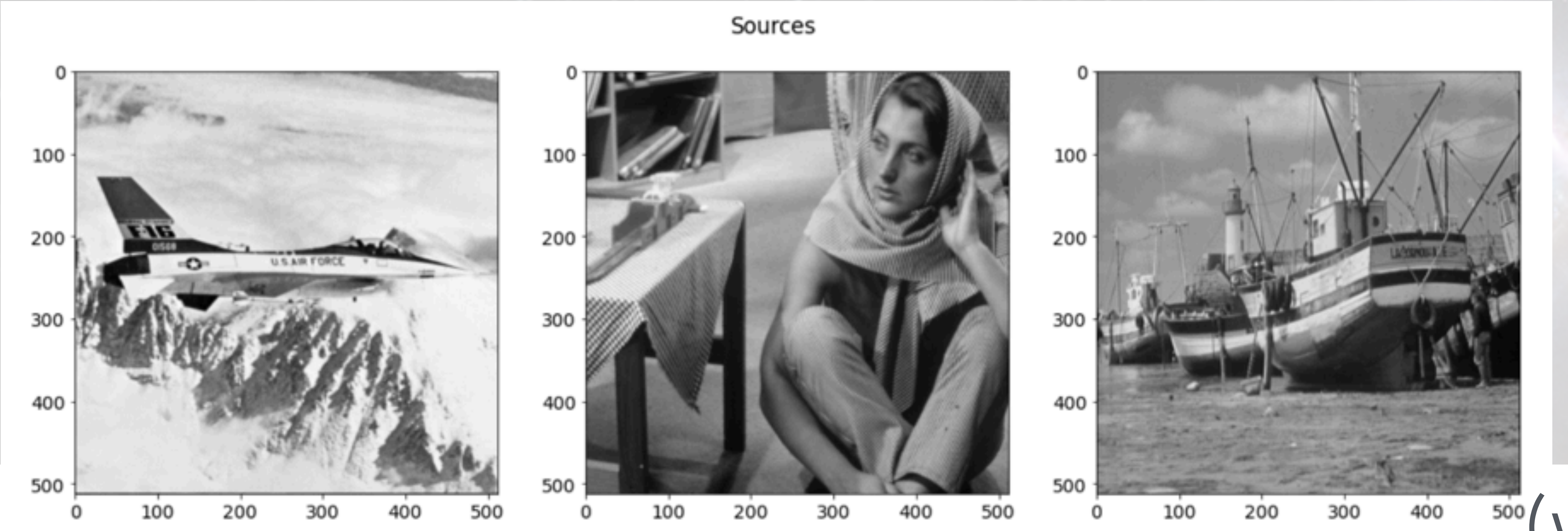


Toy model with multiple realizations of A (6 channels) and N (gaussian white) with evolving σ

(with common learnlets from ImageNet)

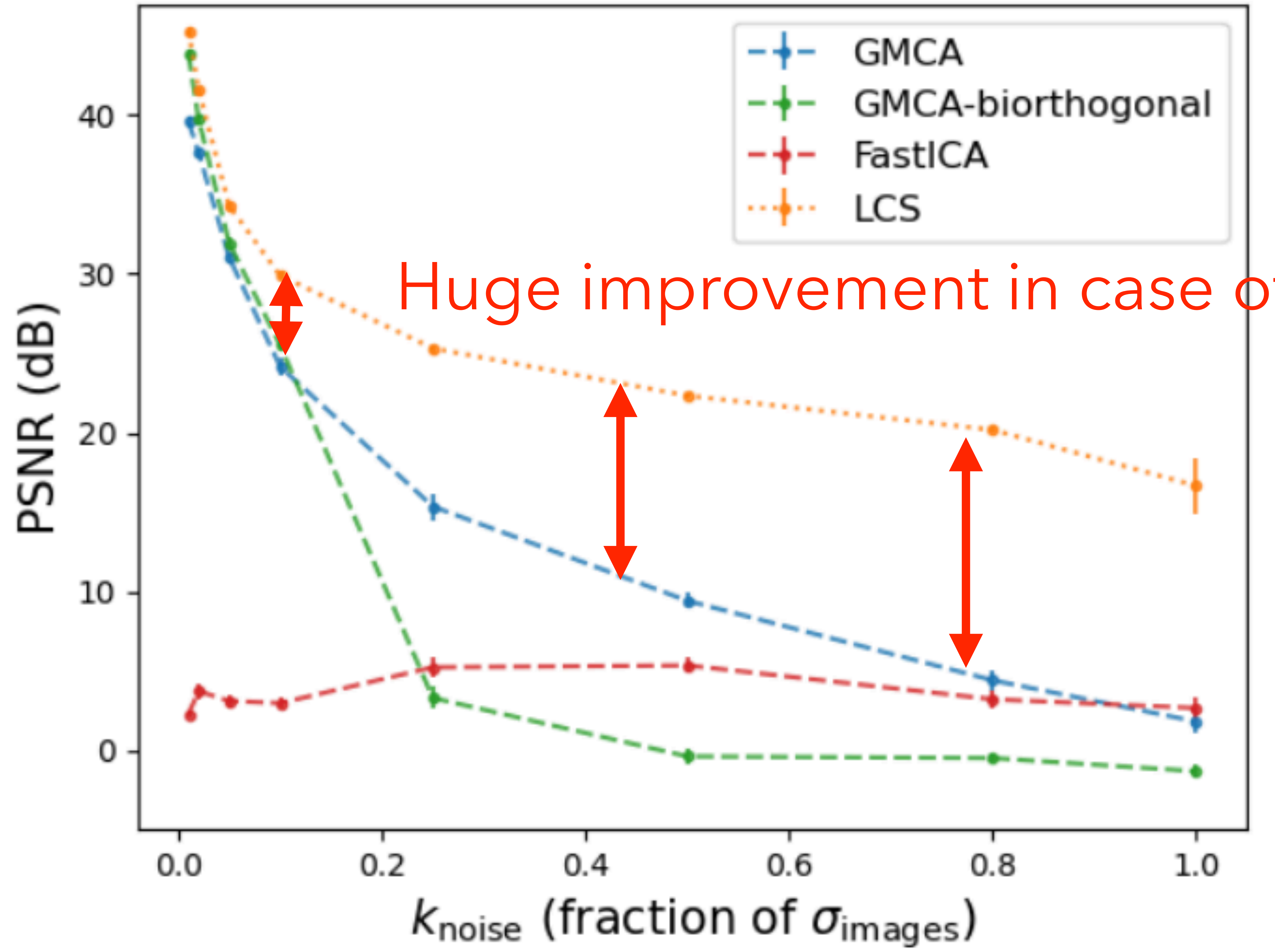


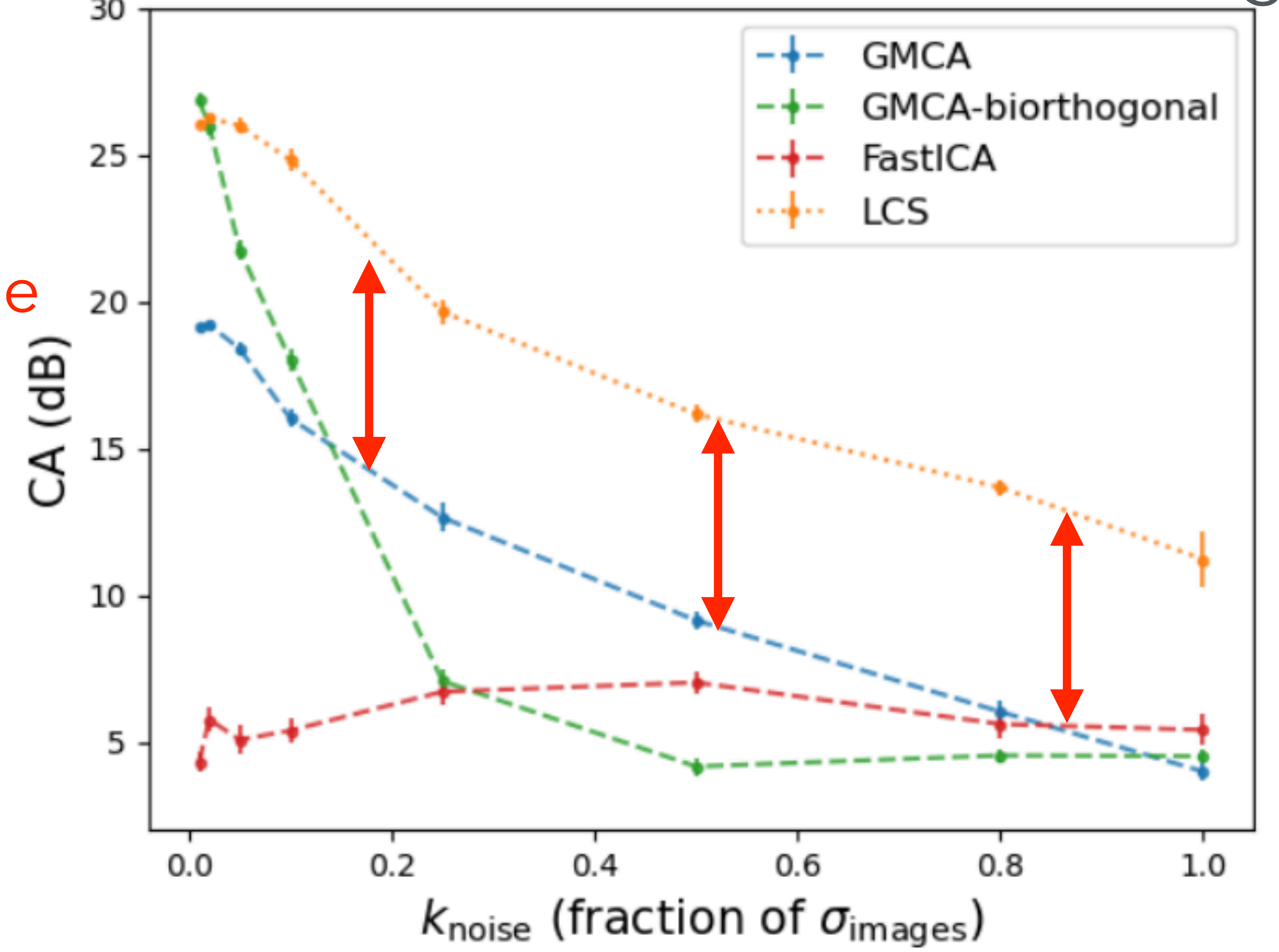
Results of LCS



Toy model with multiple realizations of A (6 channels) and N (gaussian white) with evolving σ

(with common learnlets from ImageNet)





Results of LCS

Learnlets trained for each components i:

Illustration

Results of LCS

Learnlets trained for each components i:

DTD texture dataset (Cimpoi et al, 2014), 120 images per 47 classes, here focused on 2 classes:

banded

dotted

Illustration

Results of LCS

Learnlets trained for each components i:

DTD texture dataset (Cimpoi et al, 2014), 120 images per 47 classes, here focused on 2 classes:

banded

dotted

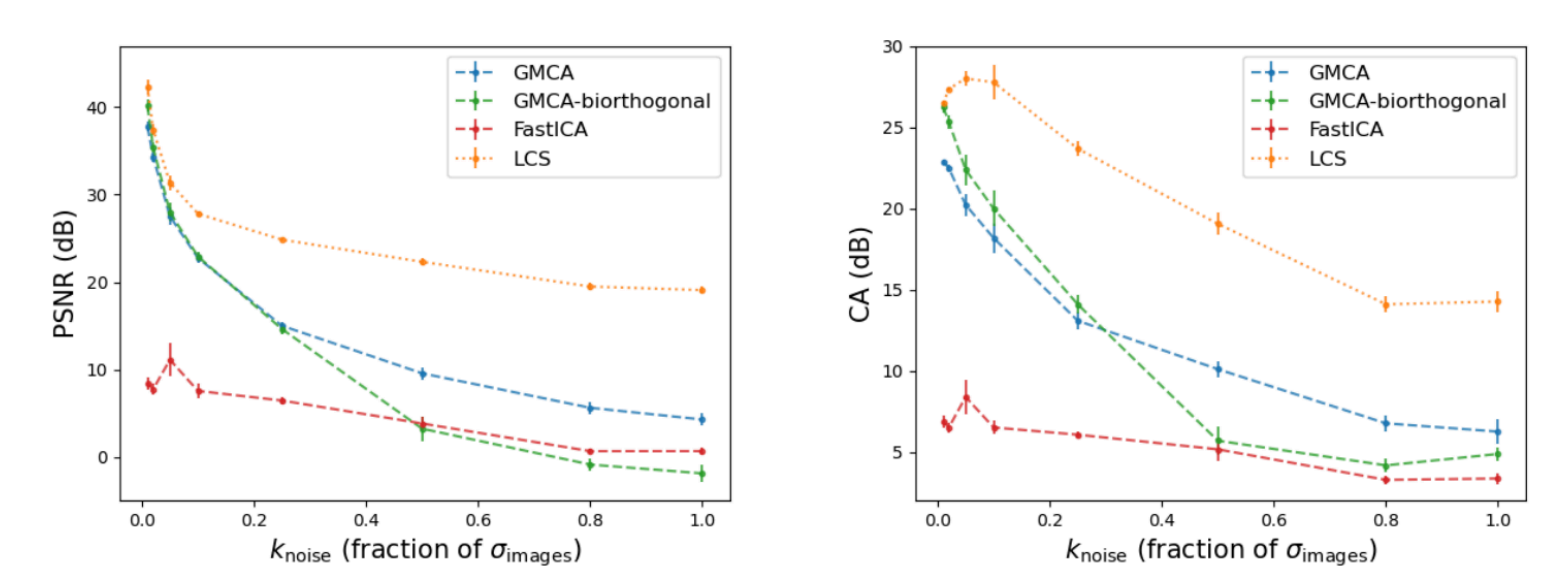
→ One training per class: $\mathscr{L}_{\text{banded}}$ and $\mathscr{L}_{\text{dotted}}$ with 119 images (transfer learning from ImageNet)

Illustration

Results of LCS

Learnlets trained for each components i:

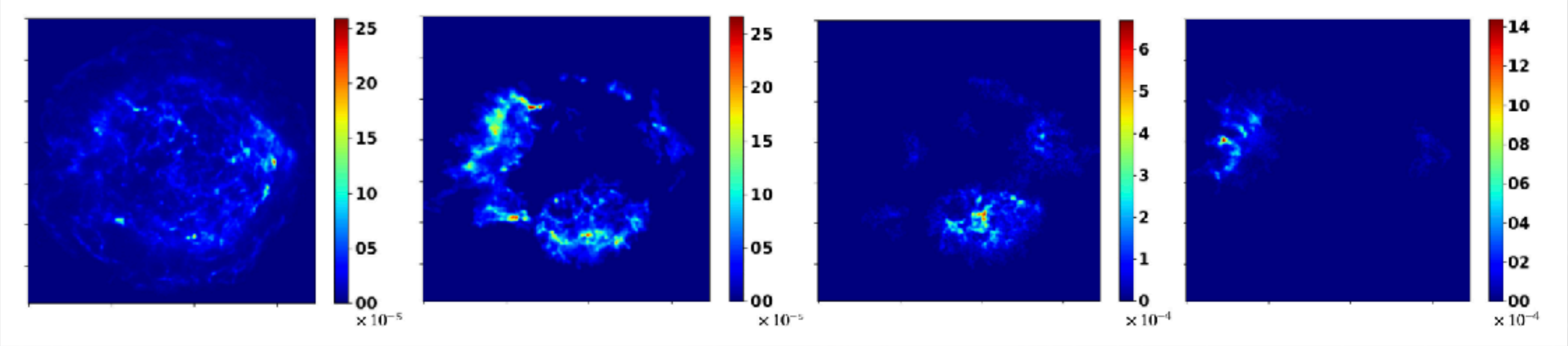
Different realizations of A and N for different σ with the 120th images of the classes



Illustration

Results of LCS in astrophysics:

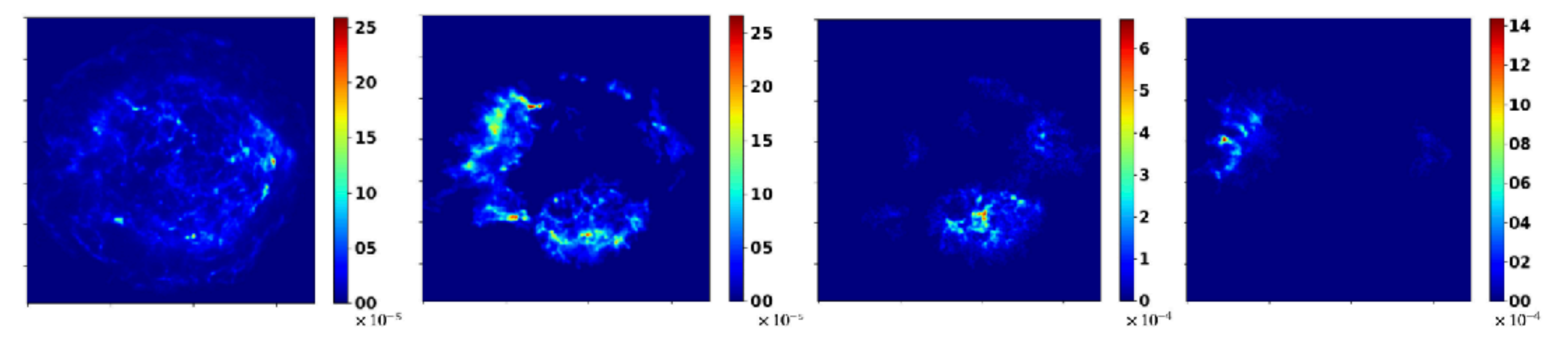
Application to supernovae remnant in X-ray images and comparison with LPALM (Fahes et al, 2022: Unrolling PALM for semi-blind source separation - supposing imperfect knowledge of A)



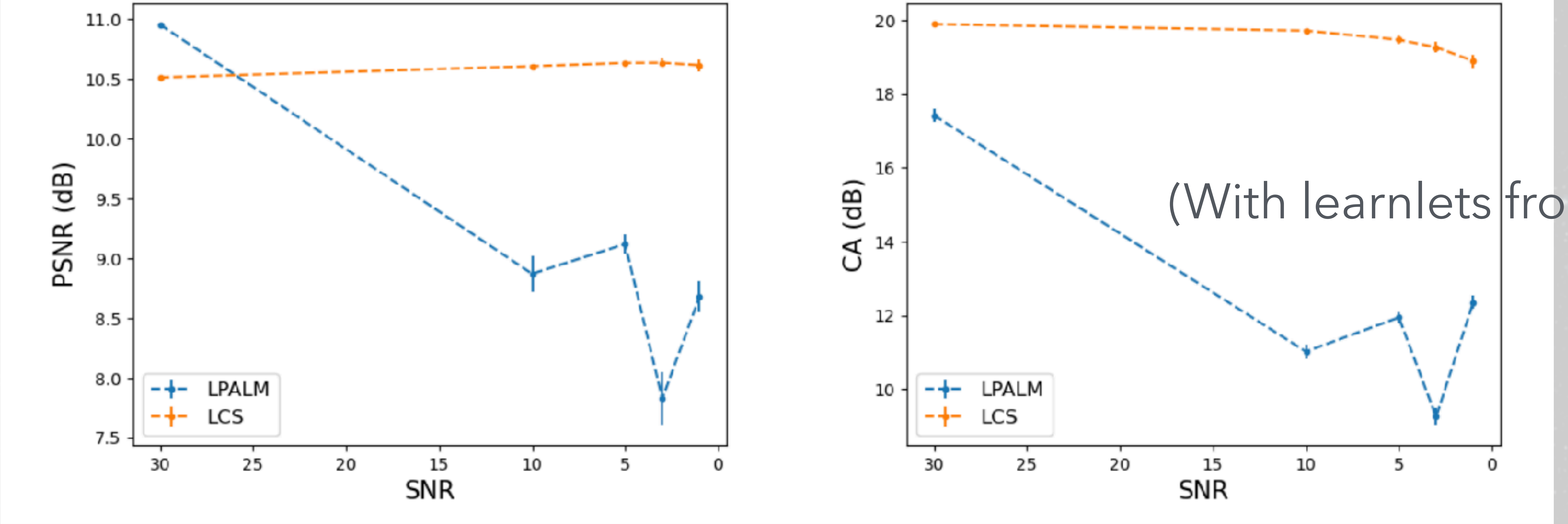
Illustration

Results of LCS in astrophysics:

Application to supernovae remnant in X-ray images and comparison with LPALM (Fahes et al, 2022: Unrolling PALM for semi-blind source separation - supposing imperfect knowledge of A)



Results on the test set of LPALM for different SNR:

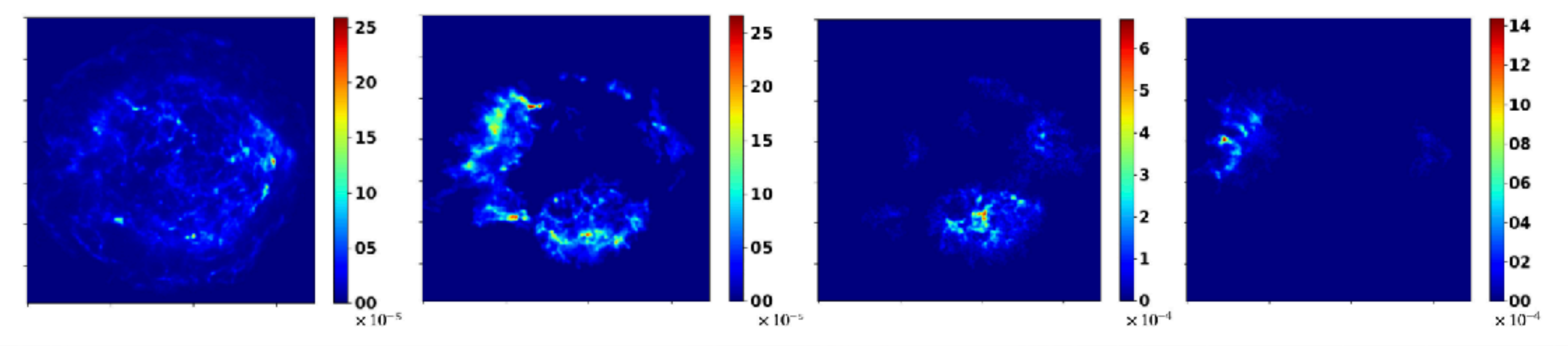


(With learnlets from ImageNet)

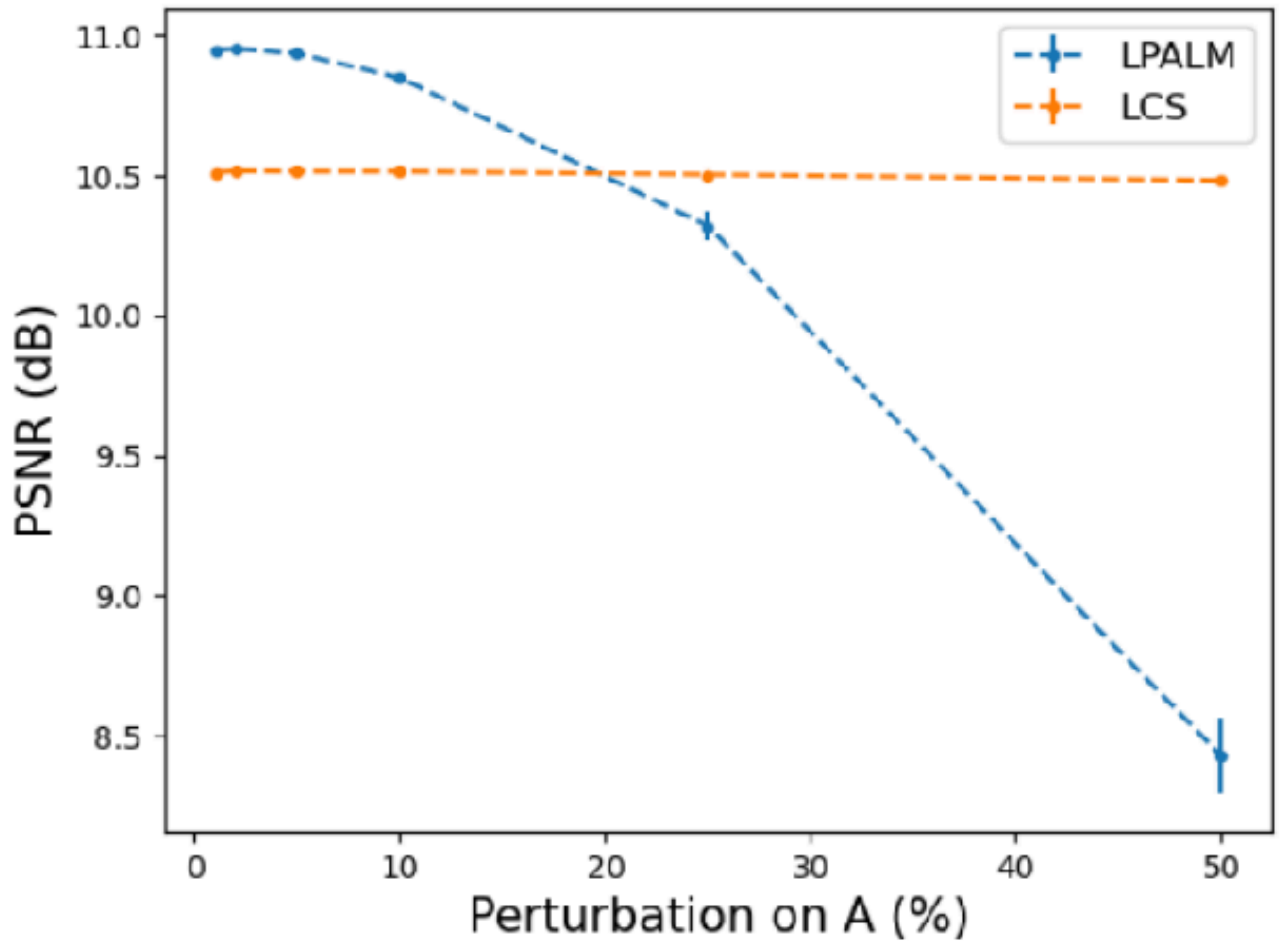
Illustration

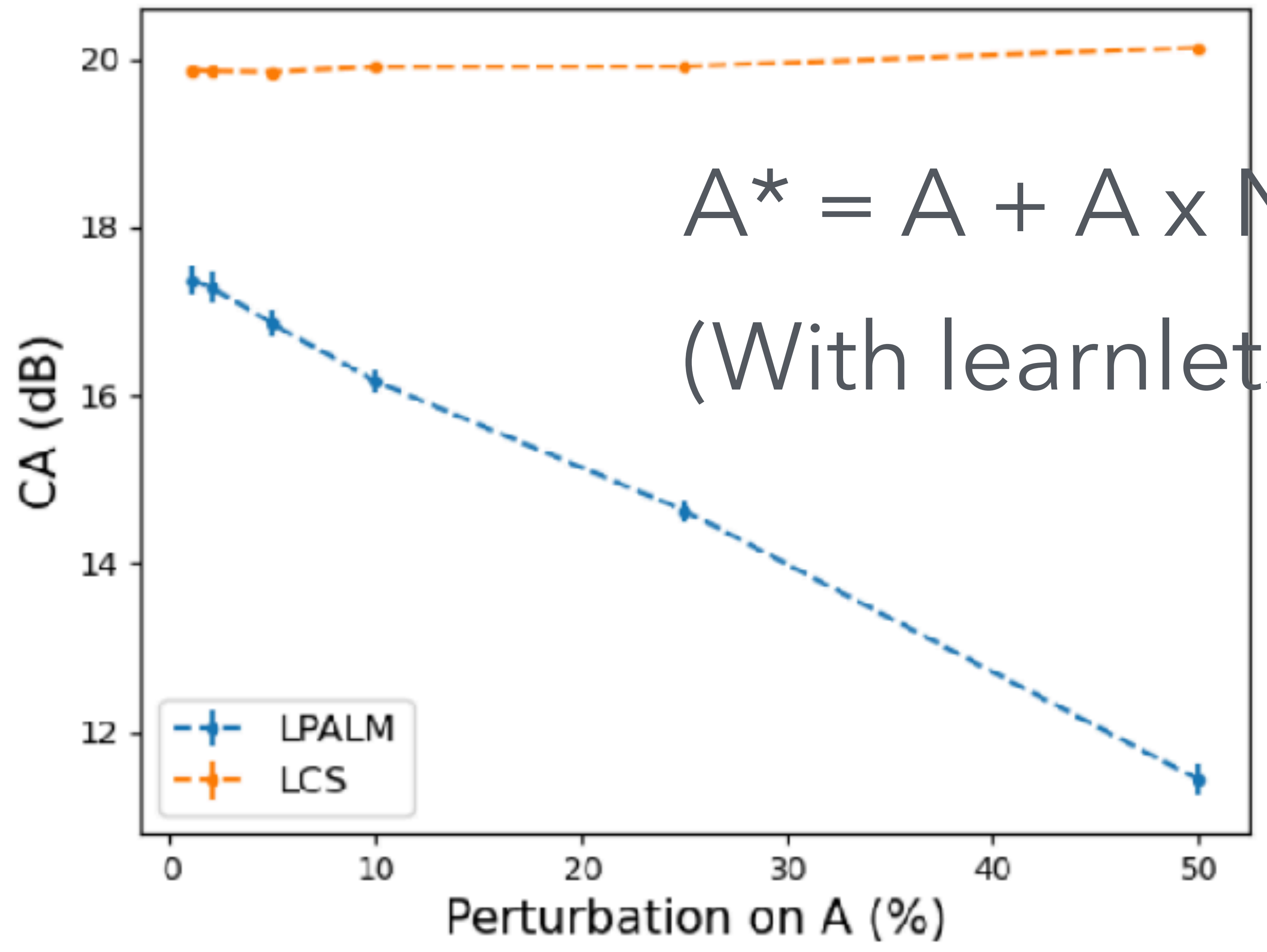
Results of LCS in astrophysics:

Application to supernovae remnant in X-ray images and comparison with LPALM (Fahes et al, 2022: Unrolling PALM for semi-blind source separation - supposing imperfect knowledge of A)



Results with fixed SNR=30 for different perturbation level on A:





A* = A + A x N(0,1) x eps

(With learnlets from ImageNet)

Illustration

Results of LCS in astrophysics:

Application to CMB & Sunyaev-Zel'dovich effect extraction (3 components CMB, SZ, CIB, no beam, noise from Planck):

→ One training per class: $\mathscr{L}_{\mathrm{CMB}}$, $\mathscr{L}_{\mathrm{SZ}}$ and $\mathscr{L}_{\mathrm{CIB}}$ (transfer learning from ImageNet) on patches from healpix numerical simulations WebSky (Stein et al., 2020)

Illustration

Results of LCS in astrophysics:

Application to CMB & Sunyaev-Zel'dovich effect extraction (3 components CMB, SZ, CIB, no beam, noise from Planck):

→ One training per class: $\mathscr{L}_{\mathrm{CMB}}$, $\mathscr{L}_{\mathrm{SZ}}$ and $\mathscr{L}_{\mathrm{CIB}}$ (transfer learning from ImageNet) on patches from healpix numerical simulations WebSky (Stein et al., 2020)

Comparison with ILC and GMCA:

Method	CMB	SZ
ILC	32.88	27.90
GMCA	26.28	25.56
LCS	**34.65**	**42.94**

For this application, column of CMB and SZ have been fixed to the theoretical weights, letting CIB being free

Illustration

Summary

- A new Learnlet implementation (PyTorch)
- BSS algorithm based on Learnlets: LCS (combining expressivity of deep learning and mathematical properties of wavelets)
- Outperforms even SBSS (LPALM) algorithms when noise
- Promising for SKA (HI extraction) and SO, Litebird (CMB, SZ, dust)

Next steps

- Combine with deconvolution (Sia's work)
- Error estimation (Hubert?)
- To the sphere / Healpix

Illustration