Deep Learning-based Deconvolution for Astronomical Surveys

Utsav Akhaury

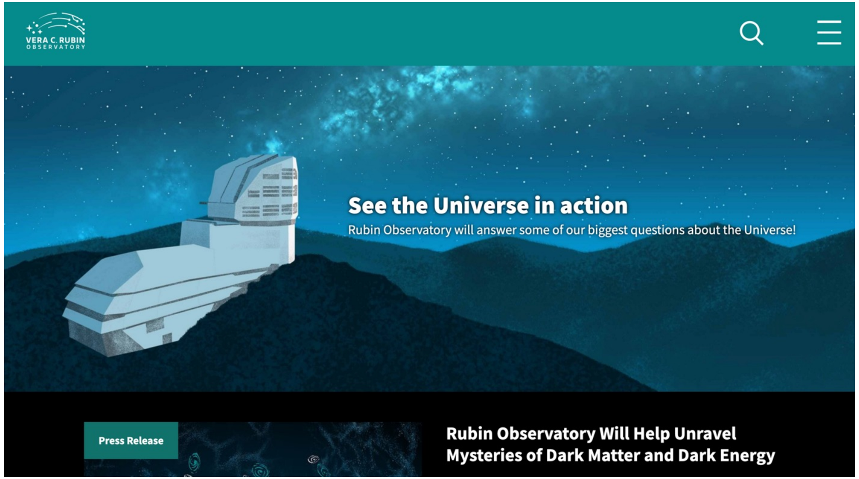
Pascale Jablonka, Jean-Luc Starck, Frédéric Courbin

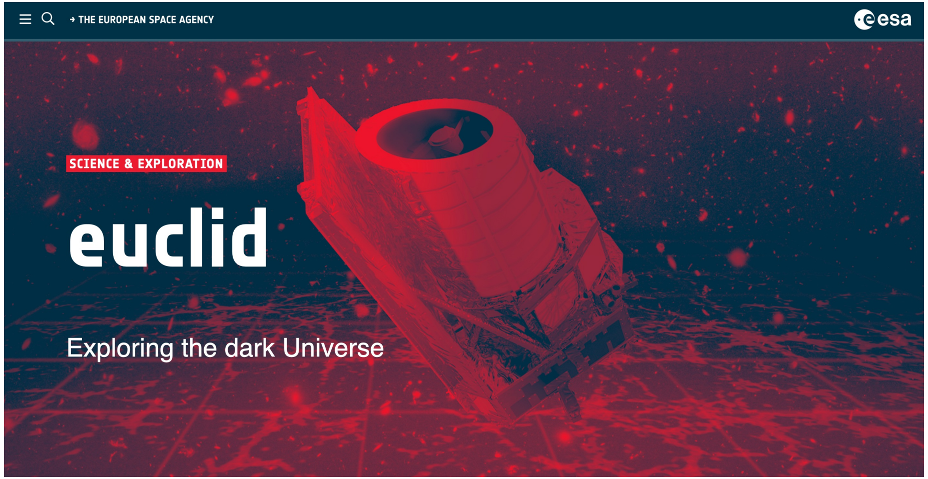
FOUNDATION FOR RESEARCH AND TECHNOLOGY - HELLAS

VERA C. RUBIN
OBSERVATORY

esa
European Space Agency

esa
euclid

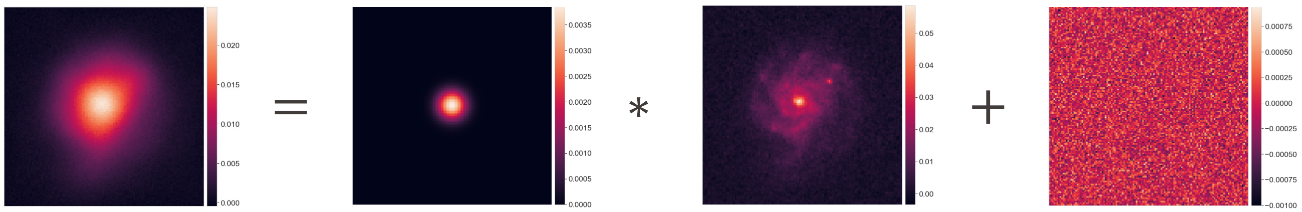




The Single-channel Deconvolution Problem

Model

$$\mathbf{y} = \mathbf{h} * \mathbf{x}_t + \eta$$



Observed Noisy Image | PSF | Ground Truth | Additive Noise

- $\mathbf{y} \in \mathbb{R}^{n\times n}$ — Observed Noisy Image
- $\mathbf{x}_t \in \mathbb{R}^{n\times n}$ — Ground Truth Image
- $\mathbf{h} \in \mathbb{R}^{n\times n}$ — PSF
- $\eta \in \mathbb{R}^{n\times n}$ — Additive Noise

Issues

- The equation is **ill-conditioned** and **ill-posed**
- Solution oscillates while using the least-squares method
- Problem could be handled by regularization

The Deconvolution Step

Loss Function

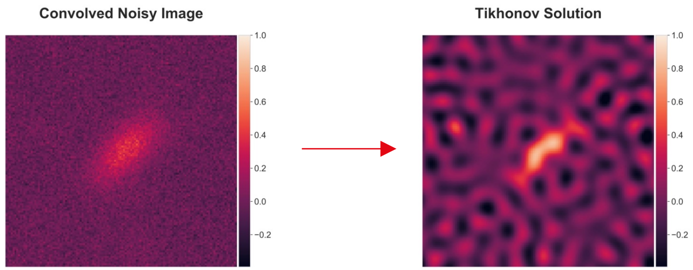
$$L(\mathbf{x}) = \frac{1}{2\sigma^2} \| \mathbf{Hx} - \mathbf{y} \|_2^2 + \lambda \| \mathbf{\Gamma x} \|_2^2$$

Tikhonov Deconvolution

$$\hat{\mathbf{x}} = (\mathbf{H}^\top \mathbf{H} + \lambda\, \mathbf{\Gamma}^\top \mathbf{\Gamma})^{-1} \mathbf{H}^\top \mathbf{y}$$

- $\sigma \in \mathbb{R}$ — Noise standard deviation
- $\mathbf{\Gamma} \in \mathbb{R}^{n^2 \times n^2}$ — Linear Tikhonov filter set to a Laplacian high-pass filter (to penalize high frequencies)
- $\lambda \in \mathbb{R}_+$ — Regularization weight
- $\mathbf{H} \in \mathbb{R}^{n^2 \times n^2}$ — Block circulant matrix associated with the convolution operator $\mathbf{h}$

$\mathbf{y}$ $\hat{\mathbf{x}}$

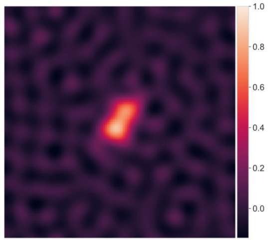


Tikhonov output contains correlated noise

The Denoising Step

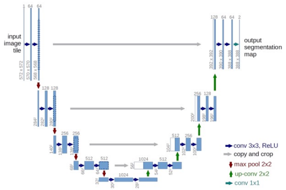
The training is aimed to make the network learn the following mapping while minimizing a suitable loss function:

- Tikhonov output $\hat{\mathbf{x}}$ → ground truth image $\mathbf{x}_t$

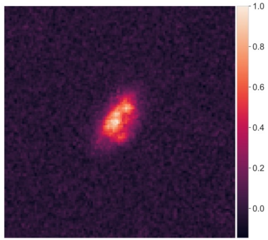


Tikhonov Solution

$\hat{\mathbf{x}}$



- Denoiser

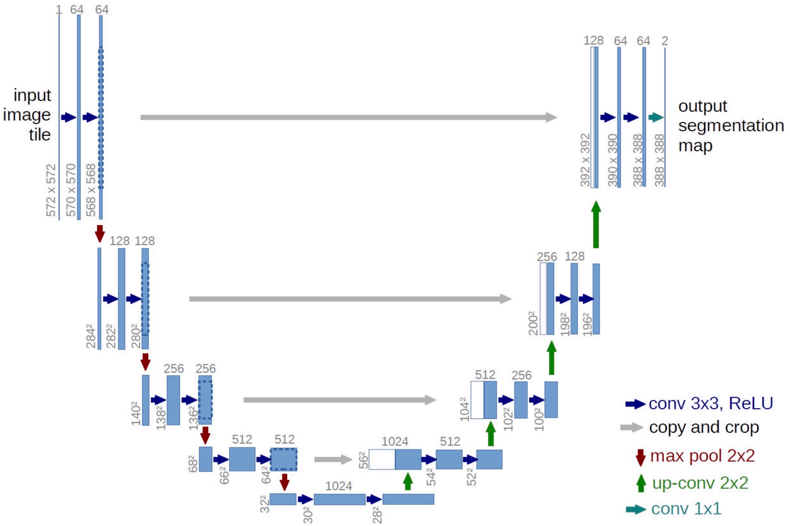


HST Target

$\mathbf{x}_t$

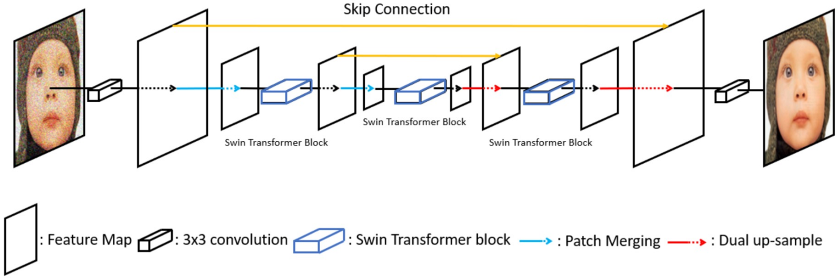
U-net

- Originally developed for biomedical image segmentation
- Relevant to many other imaging problems, like **denoising**
- U-nets consist of a multi-scale approach, allowing the signal to be analyzed at multiple resolutions



U-Net: Convolutional Networks for Biomedical Image Segmentation, *Ronneberger et al, 2015*

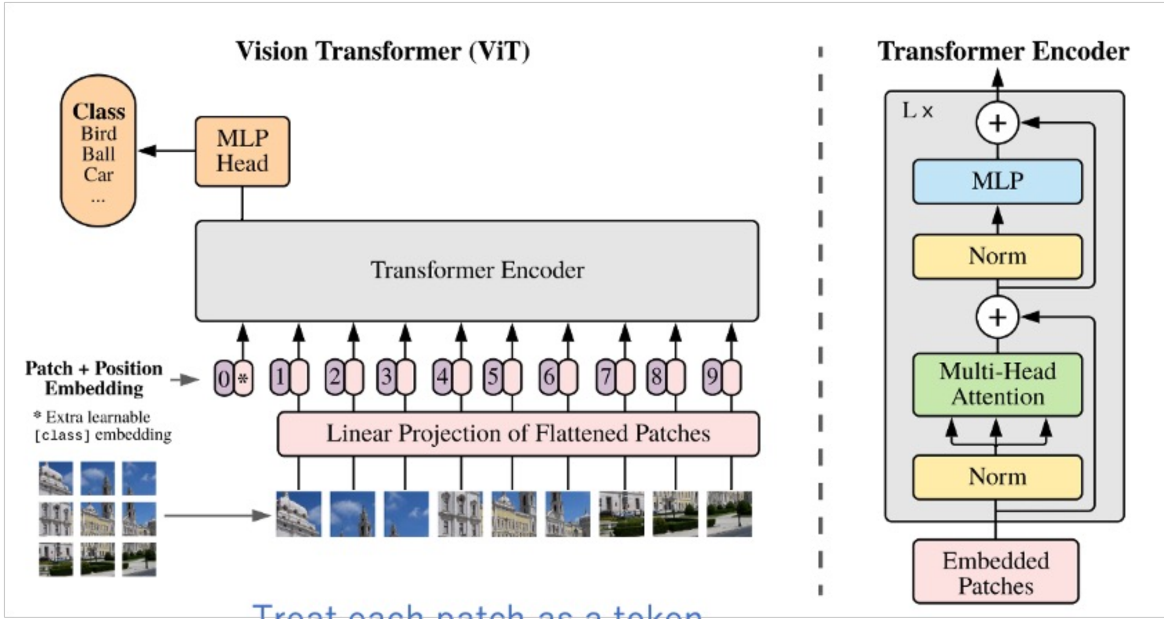
SUNet



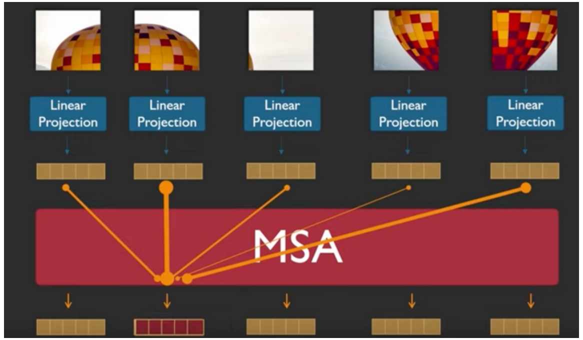
- A Unet with **Swin Transformer** blocks incorporated in the architecture

SUNet: Swin Transformer UNet for Image Denoising, *Fan et al, 2022*

Vision Transformer (ViT)



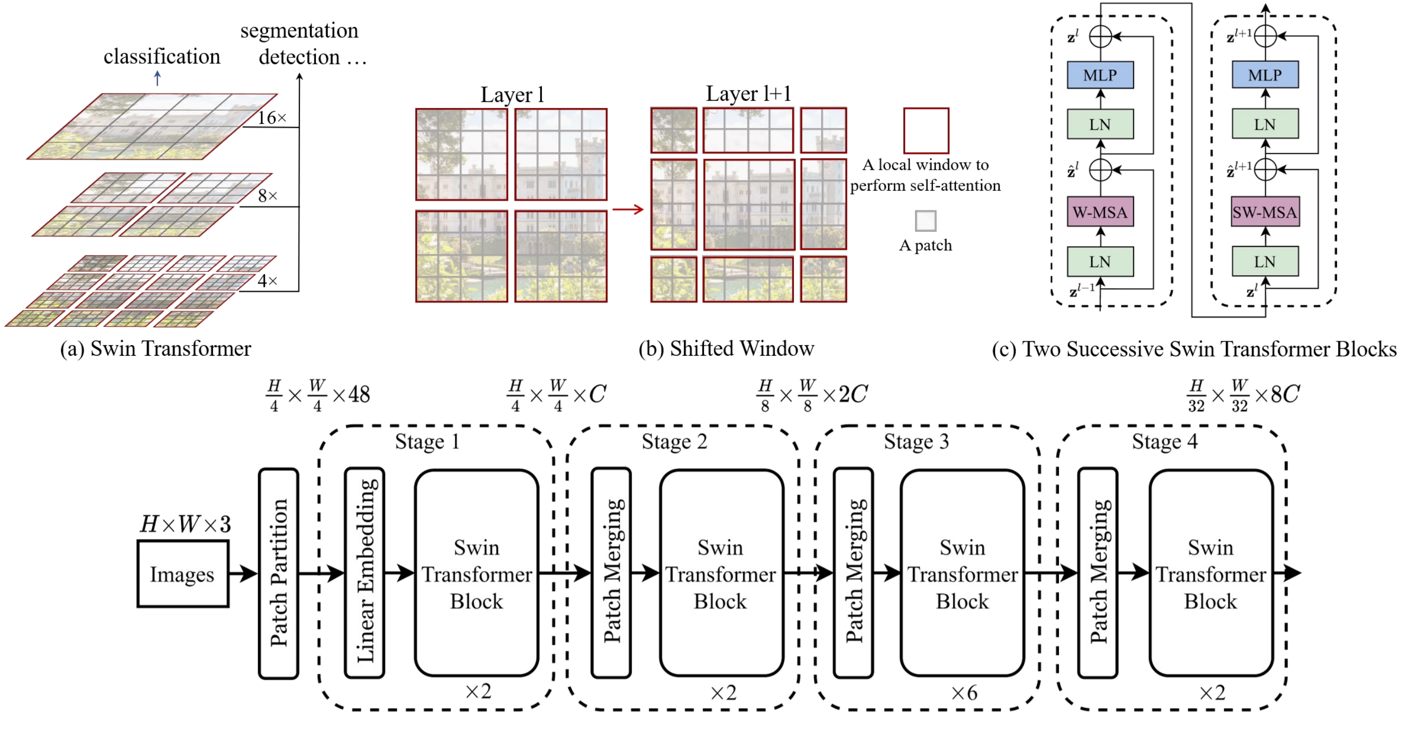
Treat each patch as a token (like a word) in NLP



An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, *Dosovitskiy et al., 2020*

Swin Transformer

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, *Liu et al, 2021*



(a) Swin Transformer

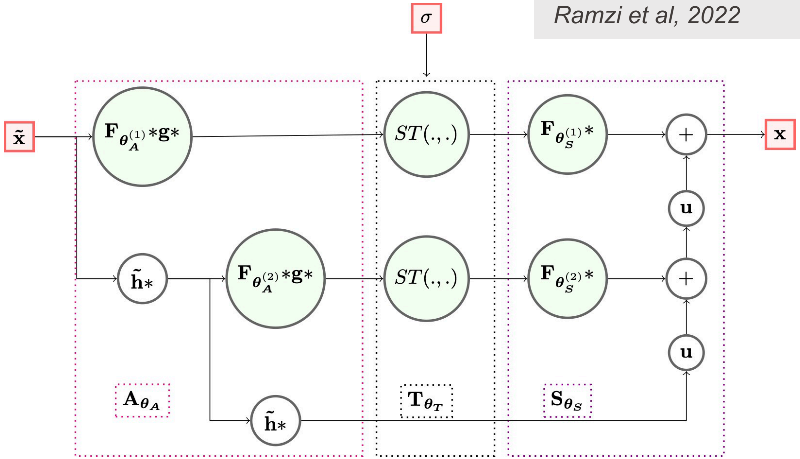
(b) Shifted Window

(c) Two Successive Swin Transformer Blocks

(d) Architecture

Learnlet

Wavelets in the Deep Learning Era
Ramzi et al, 2022



- The Learnlet decomposition (Ramzi et al., 2021) aims at learning a filter bank in a denoising setting with backpropagation and gradient descent
- Learnlets exploit the best of both of deep learning and classical algorithms –
  - uses gradient descent to improve the expressive power of wavelets
  - preserves some interesting wavelet properties like exact reconstruction

$$\mathbf{f}_{\theta}(\tilde{\mathbf{x}}, \sigma) = \mathbf{S}_{\theta_s}\left(\mathbf{T}_{\theta_t}\left(\mathbf{A}_{\theta_a}(\tilde{\mathbf{x}}), \sigma\right)\right) : (\mathbb{R}^{n\times n} \times \Sigma) \rightarrow \mathbb{R}^{n\times n}$$

Σ	set of possible values for the noise standard deviation σ
m	Number of scales
$\theta = (\theta_s, \theta_t, \theta_a) \in \Theta_m$	a given set of parameters

EPFL

Analysis Function

$$A_{\theta_a}(\tilde{x}) = \left(\left(\mathbf{F}_{\theta_a^{(i)}} * \mathbf{g}\left(\tilde{\mathbf{h}}^{i-1}(\tilde{\mathbf{x}})\right)\right)_{i=1}^{m}, \tilde{\mathbf{h}}^{m}(\tilde{\mathbf{x}})\right)$$

equivalent to the wavelet transform with learned filters

- $\mathbf{F}_{\theta_a^{(i)}}$ — filter bank at scale i
- $\theta_a^{(i)}$ — convolutional kernels
- $\tilde{\mathbf{h}}: \ \mathbf{y} \mapsto \mathbf{u}(\mathbf{h} * \mathbf{y})$ — low-pass filtering ($\mathbf{h}$) followed by decimation ($\mathbf{u}$)
- $\mathbf{g}(\mathbf{y}) = \mathbf{y} - \mathbf{u}\left(\tilde{\mathbf{h}}(\mathbf{y})\right)$ — high-pass filter with bicubic interpolation up-sampling ($\mathbf{u}$)

Thresholding Function

$$\mathbf{T}_{\theta_t}\left(\left((\mathbf{d}_i)_{i=1}^{m}, \mathbf{c}\right), \sigma\right) = \left(\left(\left(t_{ij}(d_{ij}, \sigma)\right)_{i=1}^{J_i}\right)_{i=1}^{m}, \mathbf{c}\right)$$

$$t_{ij}(\mathbf{d}, \sigma) = \hat{\sigma}_{ij}\mathrm{ST}\left(\frac{1}{\hat{\sigma}_{ij}} d_{ij}, \theta_T^{(ij)}\sigma\right)$$

$$\mathrm{ST}(\mathbf{d}, \mathbf{s}) = \mathrm{sign}(\mathbf{d})\max(|\mathbf{d}| - \mathbf{s}, 0)$$

- d_{ij} — output of the j^{th} filter of i^{th} scale
- $\hat{\sigma}_{ij}$ — estimated standard deviation of d_{ij} with an input WGN of variance 1
- $\theta_T^{(ij)}$ — thresholding level applied to the j^{th} analysis filter at scale i

Synthesis Function

$$\mathbf{S}_{\theta_s}\left((\mathbf{d}_i)_{i=1}^{m}, \mathbf{c}\right) = \mathbf{S}_{\theta_s}^{(m-1)}\left((\mathbf{d}_i)_{i=1}^{m-1}, \mathbf{u}(\mathbf{c}) + \mathbf{F}_{\theta_s^{(m)}} * \mathbf{d}_m\right)$$

$$\mathbf{S}_{\emptyset}(\emptyset, \mathbf{c}) = \mathbf{c}$$

equivalent to the wavelet reconstruction operator with learned filters

- $\mathbf{F}_{\theta_s^{(i)}}$ — filter bank at scale i
- $\theta_s^{(i)}$ — convolutional kernels
- $\mathbf{u}$ — bicubic interpolation up-sampling

Dataset Generation & Training

Ground Truth Images

- CANDELS – Five different image mosaics (GOODS-N, GOODS-S, EGS, UDS, COSMOS)
- HST cutouts of 128 × 128 pixels from CANDELS in the *F606W* filter (*V*-band) centred at the object centroid

Noisy Simulations

- Convolved ~25,000 ground-truth images with PSFs
- Added white Gaussian noise such that images have varying SNR
- Train-Test split – 0.9 : 0.1

Data Augmentation

- Random rotations in multiples of 90°, translations and flips along horizontal & vertical axes

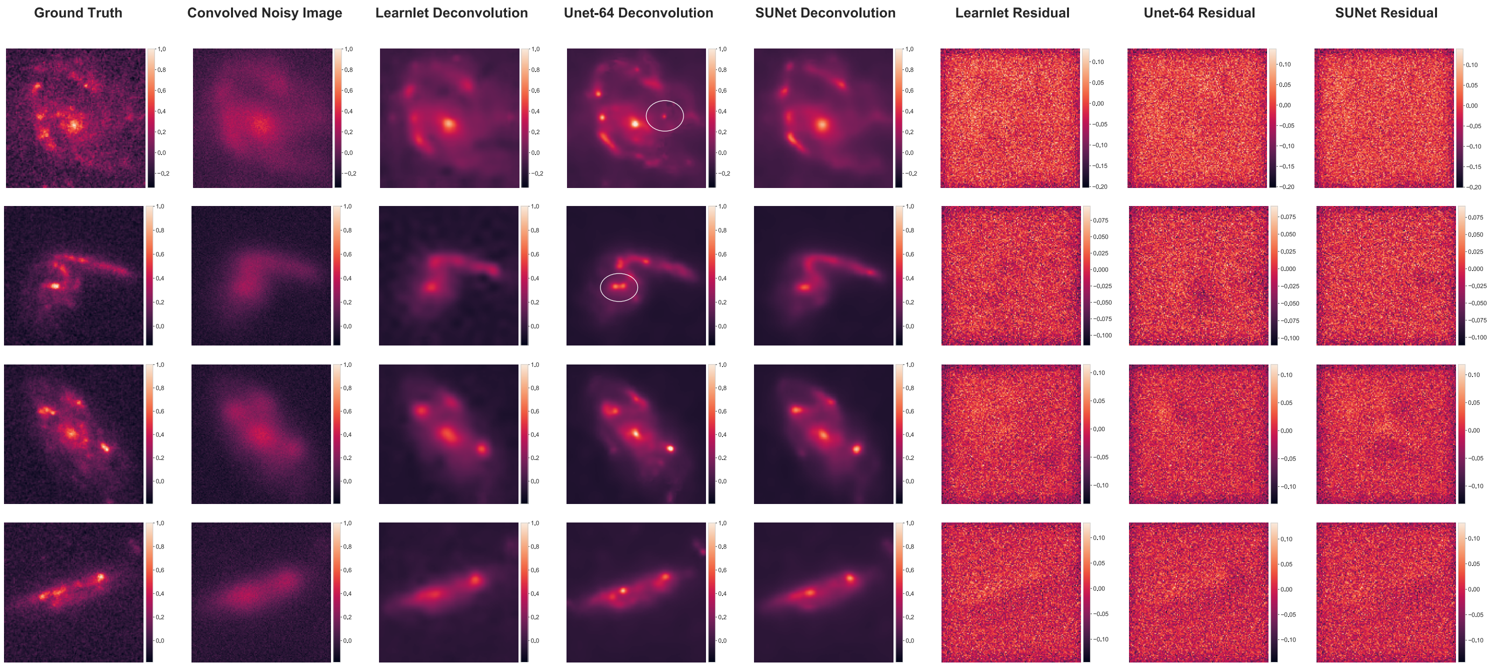
Results

Performance Comparison

Method	No. of parameters	Batch Size	Epochs	Training Time (hrs.)	Runtime per image (ms)
Learnlet	44,840	32	150	5.45	30.8
Unet-64	31,023,940	32	500	14.4	26.3
SUNet	38,365,111	16	250	90	15.2

All computations on Titan RTX Turing GPU with 24 GB RAM

Results

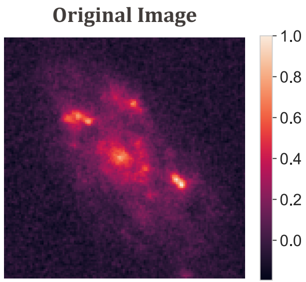


$$\text{Residual} = \mathbf{y} - \mathbf{h} * N_\theta(\hat{\mathbf{x}})$$

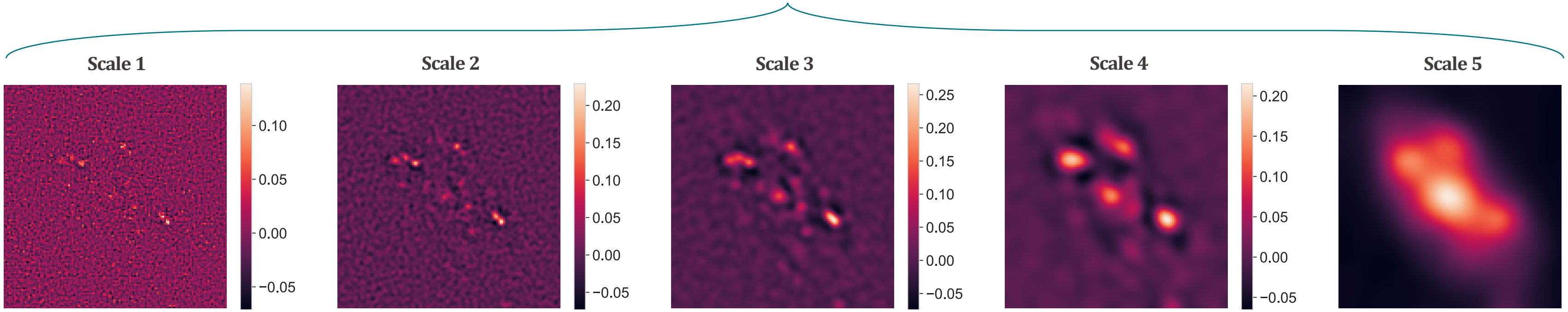
- $\mathbf{y}$ – noisy image
- $\mathbf{h}$ – PSF
- $\hat{\mathbf{x}}$ – noisy tikhonov input
- N_θ – network model

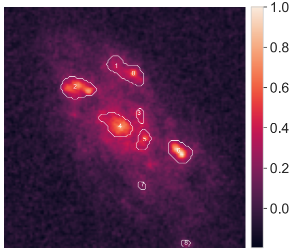
Deep learning-based galaxy image deconvolution
Akhaury et al, 2022

Multi-resolution Analysis



- Wavelet decomposition using SCARLET

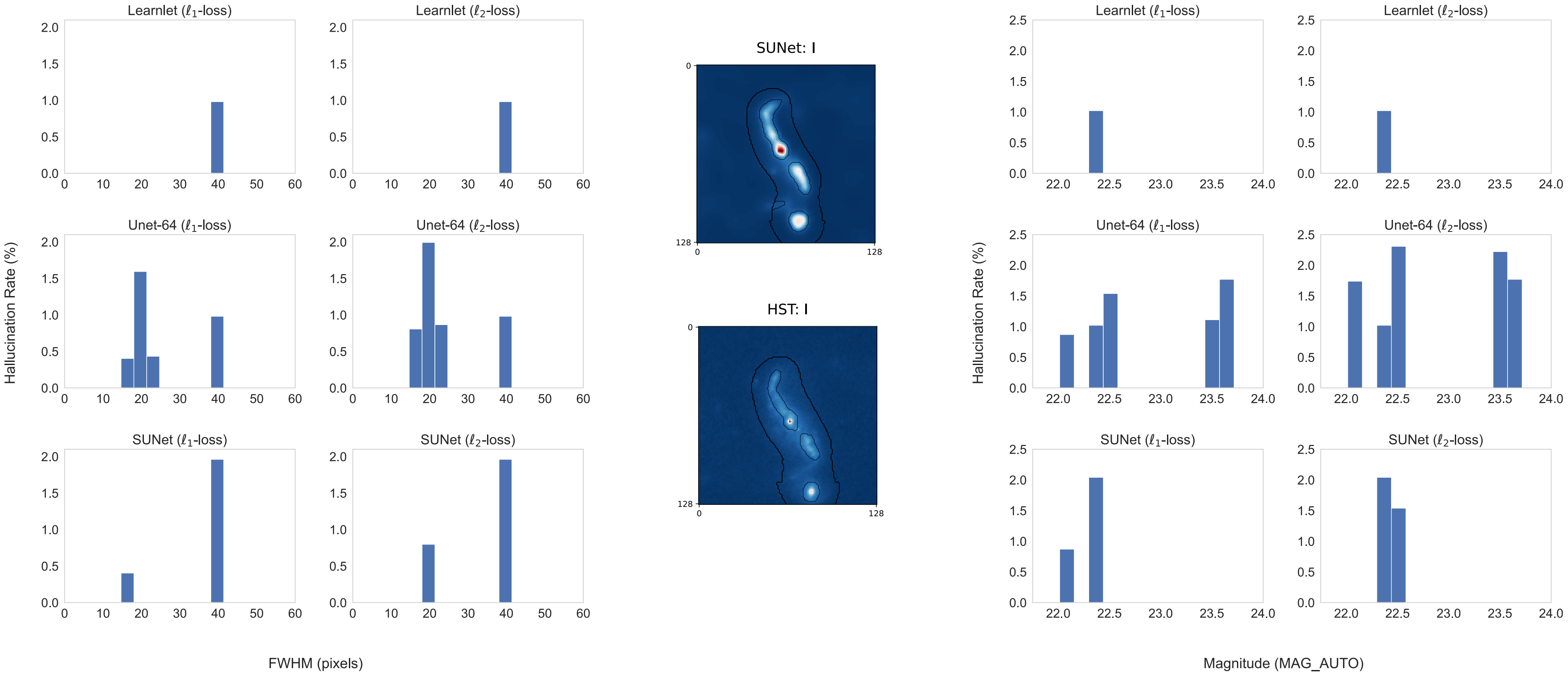




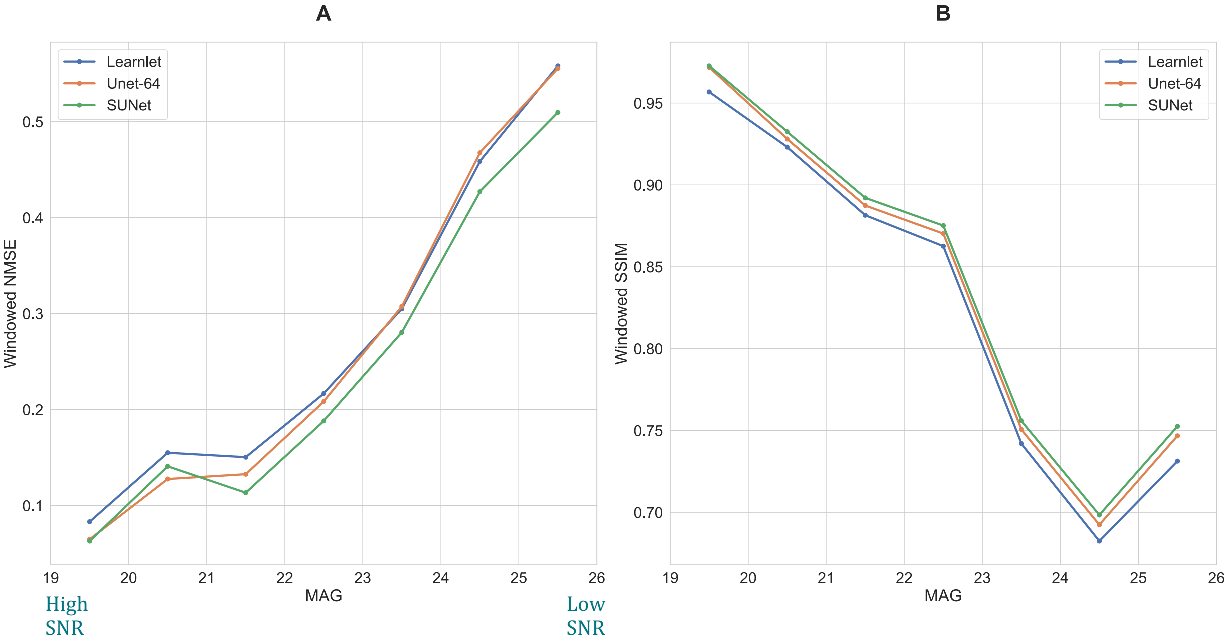
3rd Scale selected to detect clumps

5th Scale selected to detect size

Hallucination Rate



Metrics



NMSE **SSIM**

SSIM = 1: Identical
SSIM = 0: Dissimilar

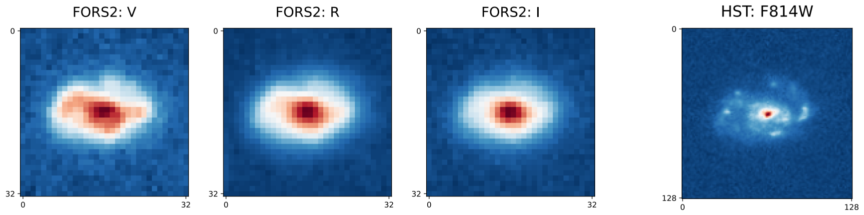
Test on VLT Images

EDisCS – the ESO distant cluster survey★,★★,★★★

Sample definition and optical photometry

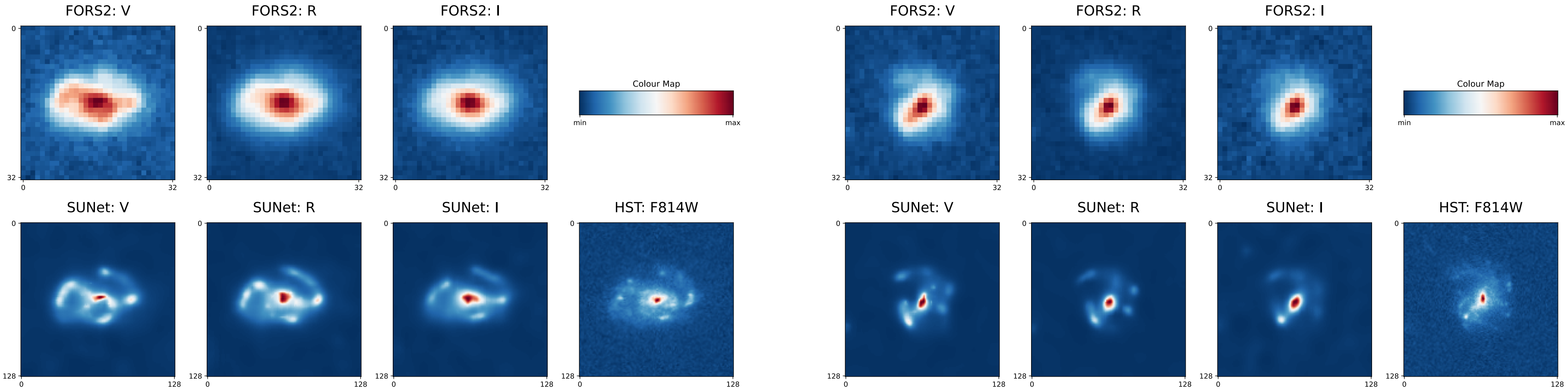
S. D. M. White[1], D. I. Clowe[2], L. Simard[3], G. Rudnick[1], G. De Lucia[1], A. Aragón-Salamanca[4], R. Bender[5], P. Best[6], M. Bremer[7], S. Charlot[1], J. Dalcanton[8], M. Dantel[9], V. Desai[8], B. Fort[10], C. Halliday[11], P. Jablonka[12], G. Kauffmann[1], Y. Mellier[10,9], B. Milvang-Jensen[5], R. Pelló[13], B. Poggianti[14], S. Poirier[12], H. Rottgering[15], R. Saglia[5], P. Schneider[16], and D. Zaritsky[2]

- All cluster members at redshifts: $\mathbf{z \approx 0.58}$, $\mathbf{z \approx 0.7}$, and $\mathbf{z \approx 0.79}$
- **Noisy images:** VLT FORS2 cutouts of 32×32 pixels in V (**555nm**), R (**655nm**), and I (**768nm**) bands with resolution = **0.2″**
- **Ground truth:** HST ACS cutouts of 128×128 pixels in the ***F814W*** filter with resolution = **0.05″**



Outputs

Ground-based Image Deconvolution with Swin Transformer UNet
Akhaury et al, 2024



- Able to resolve small-scale structures and recover morphology
- Achieves a resolution close to HST
- Generalizes well to images with completely different noise properties than the training dataset

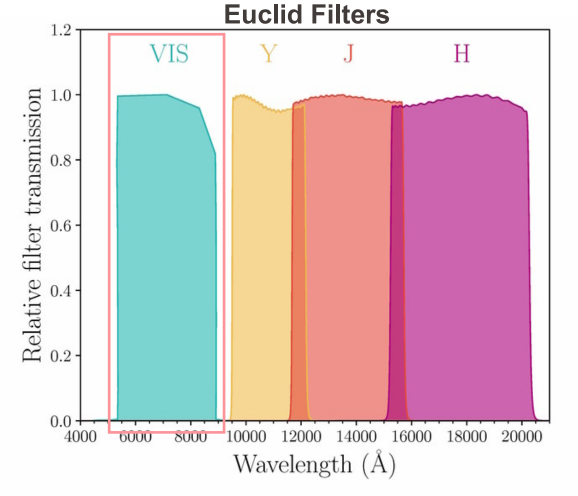
Reproducible Research

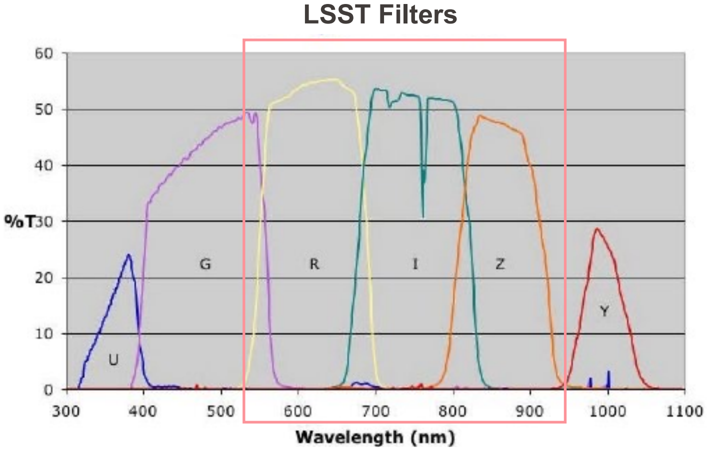
- The ready-to-use version of our SUNet deconvolution method

  https://github.com/utsav-akhaury/SUNet/tree/main/Deconvolution

Multi-channel Deconvolution

Motivation

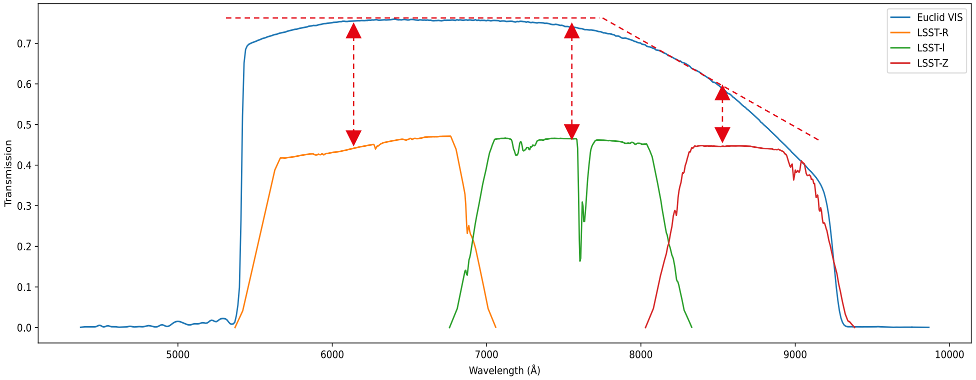




$$\mathbf{x}_{euc} = \alpha_r \mathbf{x}_r + \alpha_i \mathbf{x}_i + \alpha_z \mathbf{x}_z$$

Spectral Energy Distributions (SED)

$$\alpha_r,\ \alpha_i,\ \alpha_z \in \mathbb{R}^n$$



The Multi-channel Deconvolution Problem

Model

$$\mathbf{y}_r = \mathbf{h}_r * \mathbf{x}_r^t + \eta_r$$

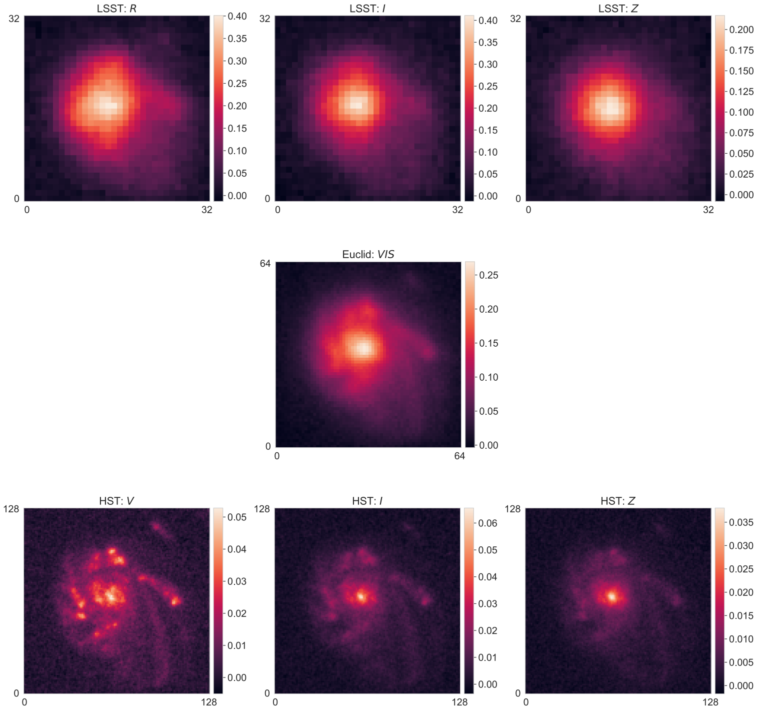
$$\mathbf{y}_i = \mathbf{h}_i * \mathbf{x}_i^t + \eta_i$$

$$\mathbf{y}_z = \mathbf{h}_z * \mathbf{x}_z^t + \eta_z$$

$$\mathbf{x}_{euc}^t = \alpha_r \mathbf{x}_r^t + \alpha_i \mathbf{x}_i^t + \alpha_z \mathbf{x}_z^t$$

$$\mathbf{y}_{euc} = \mathbf{h}_{euc} * \mathbf{x}_{euc}^t + \eta_{euc}$$

- Observed Noisy Images $\mathbf{y}_r, \mathbf{y}_i, \mathbf{y}_z \in \mathbb{R}^{n\times n}$
- PSFs $\mathbf{h}_r, \mathbf{h}_i, \mathbf{h}_z \in \mathbb{R}^{n\times n}$
- Ground Truth Images $\mathbf{x}_r^t, \mathbf{x}_i^t, \mathbf{x}_z^t \in \mathbb{R}^{n\times n}$
- Additive Noise $\eta_r, \eta_i, \eta_z \in \mathbb{R}^{n\times n}$
- Spectral Energy Distributions (SED) $\alpha_r, \alpha_i, \alpha_z \in \mathbb{R}^n$



LSST 0.2”

Euclid 0.1”

HST 0.05”

The Loss Functions

$$L_r(\mathbf{x}_r) = \frac{1}{2}\left\|\frac{\mathbf{h}_r * \mathbf{x}_r - \mathbf{y}_r}{\sigma_r}\right\|_F^2 + \lambda_{constr}\left\|\frac{\mathbf{h}_{euc} * \sum_c \alpha_c \mathbf{x}_c - \mathbf{y}_{euc}}{\sigma_{euc}}\right\|_F^2$$

$$L_i(\mathbf{x}_i) = \frac{1}{2}\left\|\frac{\mathbf{h}_i * \mathbf{x}_i - \mathbf{y}_i}{\sigma_i}\right\|_F^2 + \lambda_{constr}\left\|\frac{\mathbf{h}_{euc} * \sum_c \alpha_c \mathbf{x}_c - \mathbf{y}_{euc}}{\sigma_{euc}}\right\|_F^2$$

$$L_z(\mathbf{x}_z) = \frac{1}{2}\left\|\frac{\mathbf{h}_z * \mathbf{x}_z - \mathbf{y}_z}{\sigma_z}\right\|_F^2 + \lambda_{constr}\left\|\frac{\mathbf{h}_{euc} * \sum_c \alpha_c \mathbf{x}_c - \mathbf{y}_{euc}}{\sigma_{euc}}\right\|_F^2$$

where

$c \in \{r, i, z\}$

$\lambda_{constr} \in \mathbb{R}_+$

- Spectral Energy Distributions (SED) $\alpha_r, \alpha_i, \alpha_z \in \mathbb{R}^n$
- Noisemaps $\sigma_r, \sigma_i, \sigma_z \in \mathbb{R}^{n \times n}$

Optimization

$$\hat{\mathbf{x}}_{\{r,i,z\}} = \underset{\mathbf{x}_{\{r,i,z\}}}{\operatorname{argmin}} L_{\{r,i,z\}}\left(\mathbf{x}_{\{r,i,z\}}\right)$$

Loss Functions iteratively minimized using Gradient Descent

$$\mathbf{x}_{\{r,i,z\}}^{[k+1]} = \mathbf{x}_{\{r,i,z\}}^{[k]} - \beta_{\{r,i,z\}} \nabla L_{\{r,i,z\}}\left(\mathbf{x}_{\{r,i,z\}}^{[k]}\right)$$

Step Sizes

$$\beta_r, \beta_i, \beta_z \in \mathbb{R}^n$$

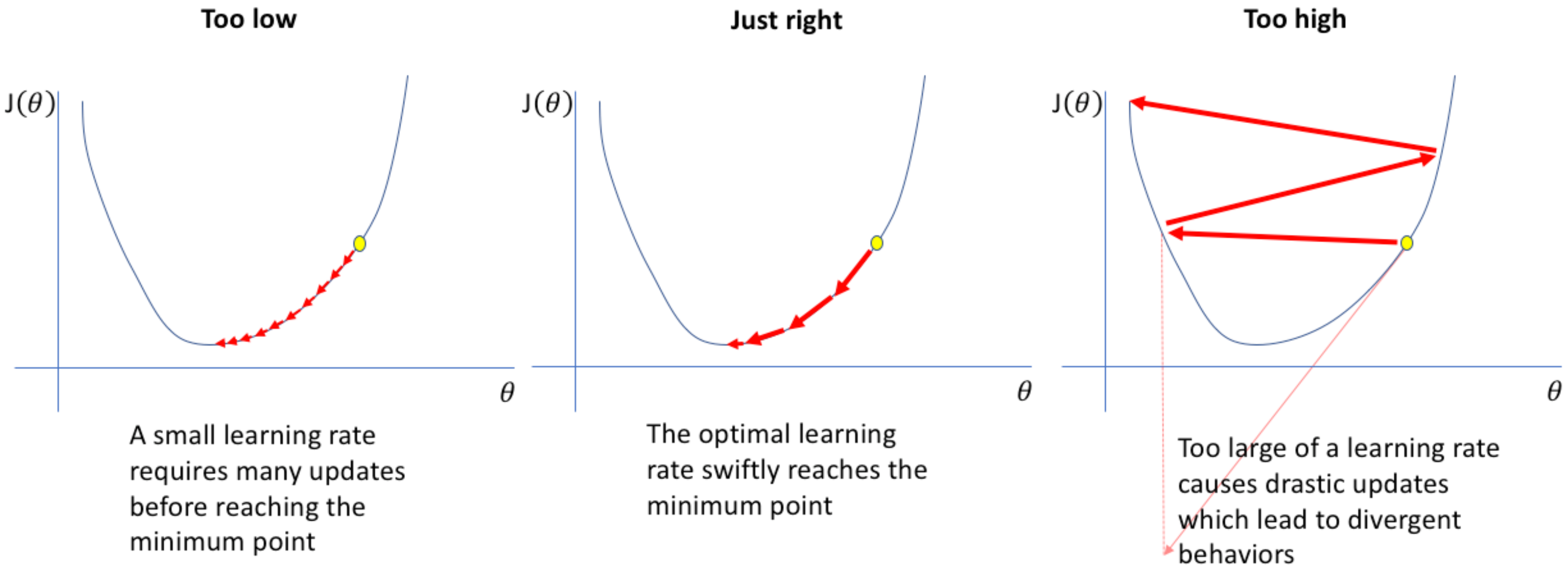
Gradients of the Loss Functions

$$\nabla L_r(\mathbf{x}_r) = \frac{\mathbf{h}_r^\top * (\mathbf{h}_r * \mathbf{x}_r - \mathbf{y}_r)}{\|\sigma_r\|_F^2} + 2\lambda_{constr}\alpha_r \mathbf{h}_{euc}^\top * \left[\frac{\mathbf{h}_{euc} * \sum_c \alpha_c \mathbf{x}_c - \mathbf{y}_{euc}}{\|\sigma_{euc}\|_F^2}\right]$$

$$\nabla L_i(\mathbf{x}_i) = \frac{\mathbf{h}_i^\top * (\mathbf{h}_i * \mathbf{x}_i - \mathbf{y}_i)}{\|\sigma_i\|_F^2} + 2\lambda_{constr}\alpha_i \mathbf{h}_{euc}^\top * \left[\frac{\mathbf{h}_{euc} * \sum_c \alpha_c \mathbf{x}_c - \mathbf{y}_{euc}}{\|\sigma_{euc}\|_F^2}\right]$$

$$\nabla L_z(\mathbf{x}_z) = \frac{\mathbf{h}_z^\top * (\mathbf{h}_z * \mathbf{x}_z - \mathbf{y}_z)}{\|\sigma_z\|_F^2} + 2\lambda_{constr}\alpha_z \mathbf{h}_{euc}^\top * \left[\frac{\mathbf{h}_{euc} * \sum_c \alpha_c \mathbf{x}_c - \mathbf{y}_{euc}}{\|\sigma_{euc}\|_F^2}\right]$$

Convergence Guarantee & Optimal step size



Credits: Jeremy Jordan

Convergence Guarantee & Optimal step size

A function's gradient is Lipschitz continuous if

$$\|\nabla f(\mathbf{x}') - \nabla f(\mathbf{x})\| \leq C\|\mathbf{x}' - \mathbf{x}\|$$

where C is the **Lipschitz constant**

In our case

$$\left\|\nabla L_{\{r,i,z\}}\left(\mathbf{x}'_{\{r,i,z\}}\right) - \nabla L_{\{r,i,z\}}\left(\mathbf{x}_{\{r,i,z\}}\right)\right\| \leq C_{\{r,i,z\}}\left\|\mathbf{x}'_{\{r,i,z\}} - \mathbf{x}_{\{r,i,z\}}\right\|$$

Substituting the individual loss functions, we get

$$C_{\{r,i,z\}} \geq \frac{\mathbf{h}_{\{r,i,z\}}^{\top} * \mathbf{h}_{\{r,i,z\}}}{\left\|\sigma_{\{r,i,z\}}\right\|_F^2} + \frac{2\lambda_{constr}\alpha_{\{r,i,z\}}^2 \mathbf{h}_{euc}^{\top} * \mathbf{h}_{euc}}{\|\sigma_{euc}\|_F^2}$$

The Optimal Condition for Convergence

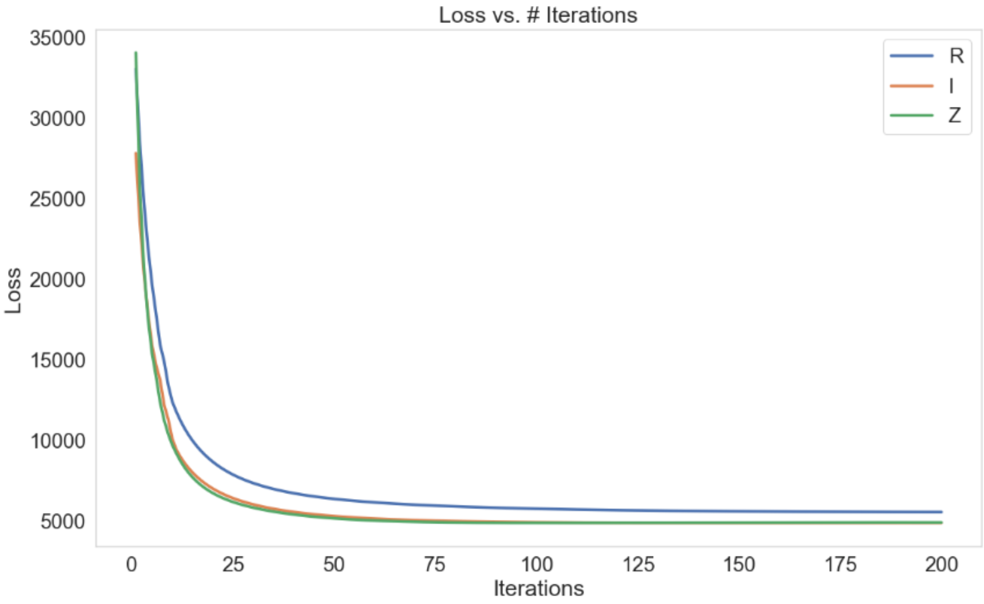
$$\beta_{\{r,i,z\}} \leq \frac{1}{C_{\{r,i,z\}}}$$

Hence, we choose

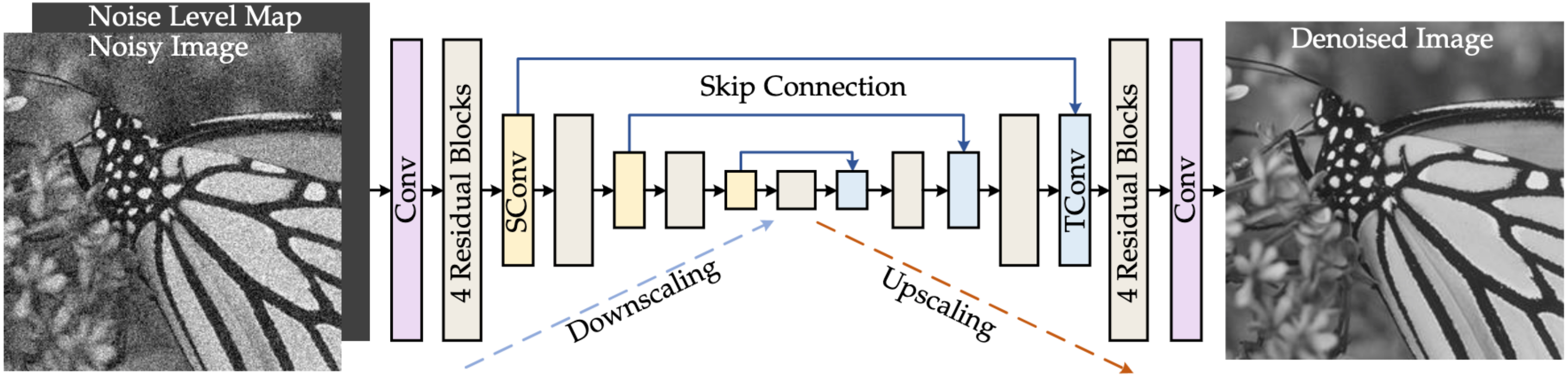
$$\beta_{\{r,i,z\}} = \frac{1}{(1+10^{-5})C_{\{r,i,z\}}}$$

Convergence

- Algorithm run for 200 iterations
- Converegence within 50-100 iterations

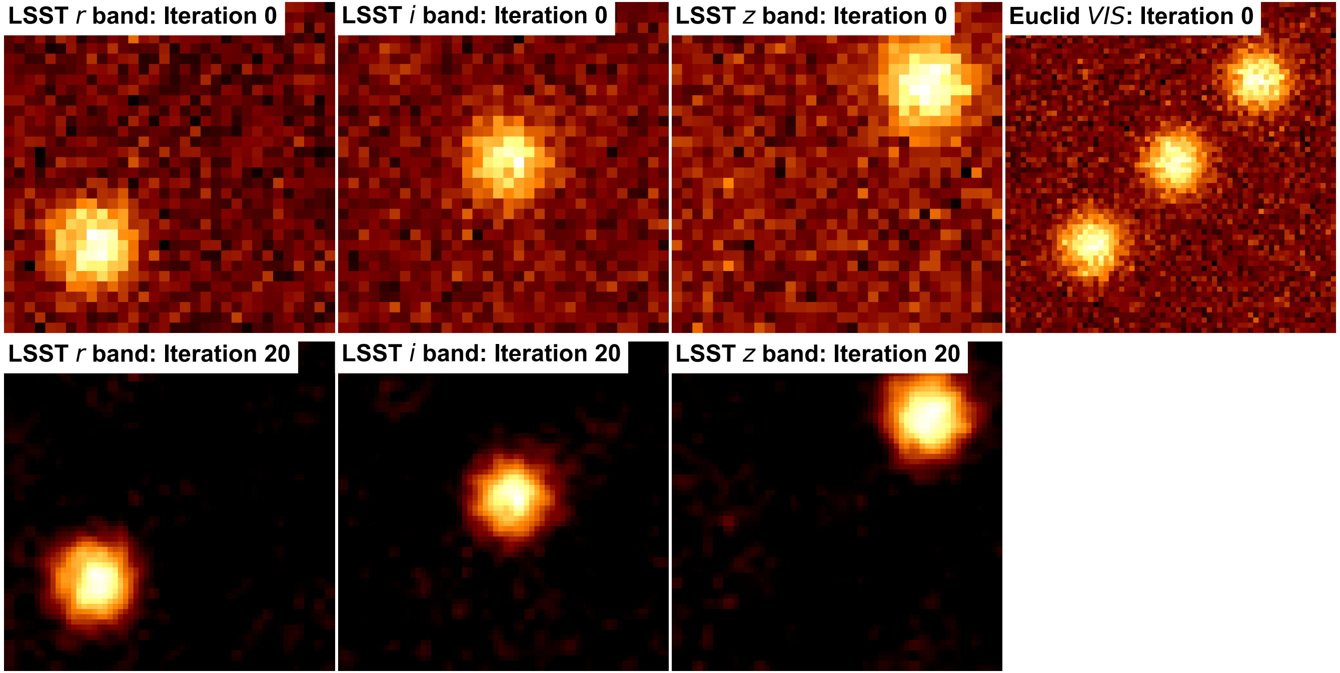


DRUNet Denoising



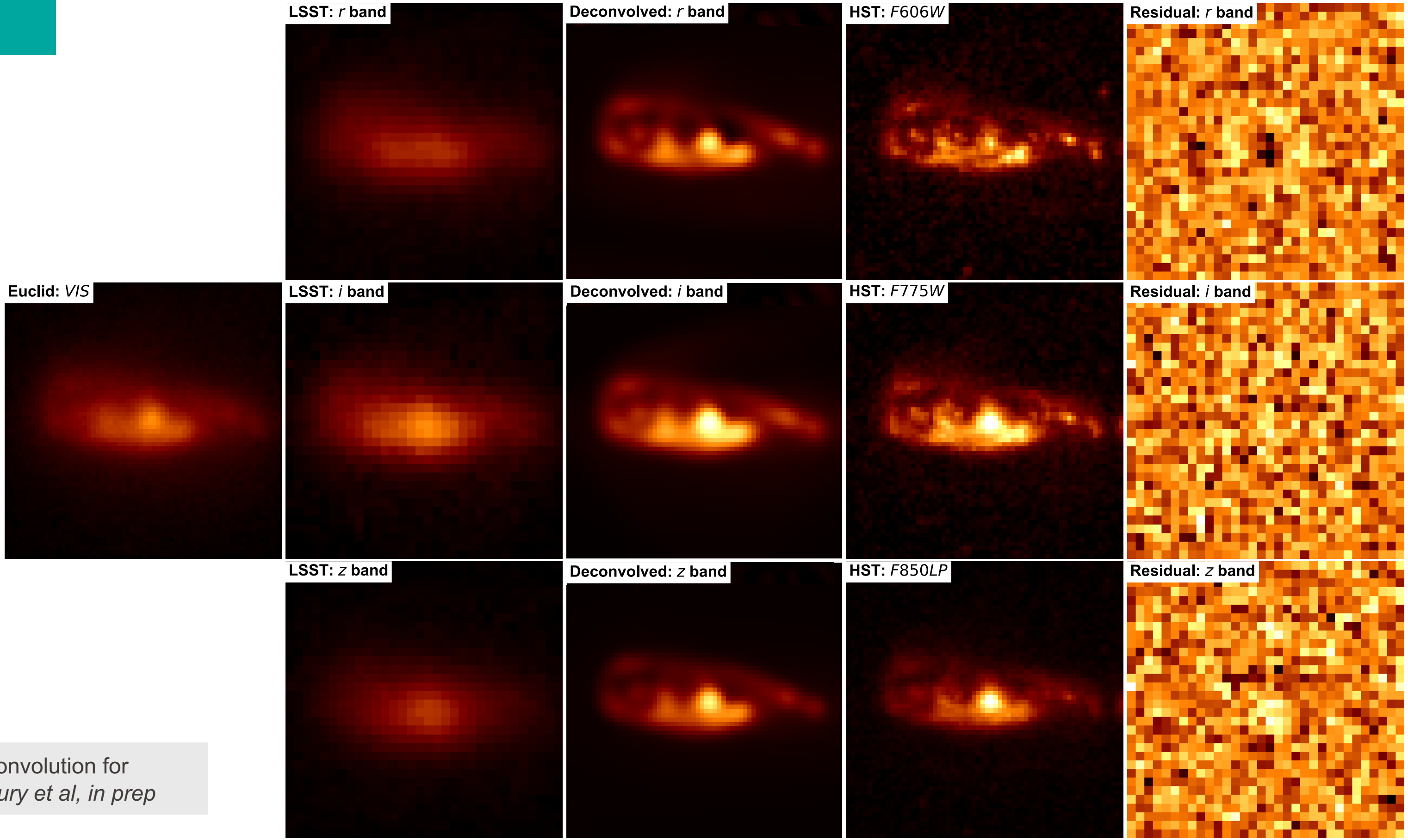
Plug-and-Play Image Restoration with Deep Denoiser Prior, *Zhang et al., 2021*

Flux Leakage Test



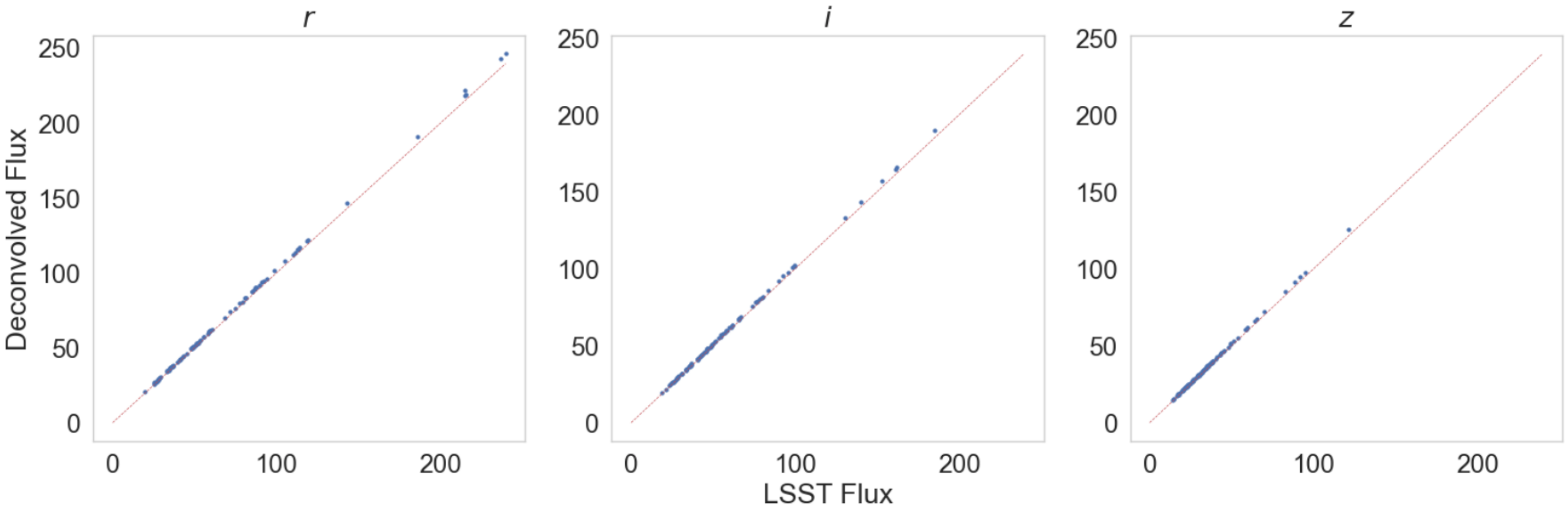
- Assume 3 separately placed Gaussians in each channel (corresponding to LSST channels)
- The joint image (Euclid) is a linear sum of these channels
- No Flux Leakage from one channel to another

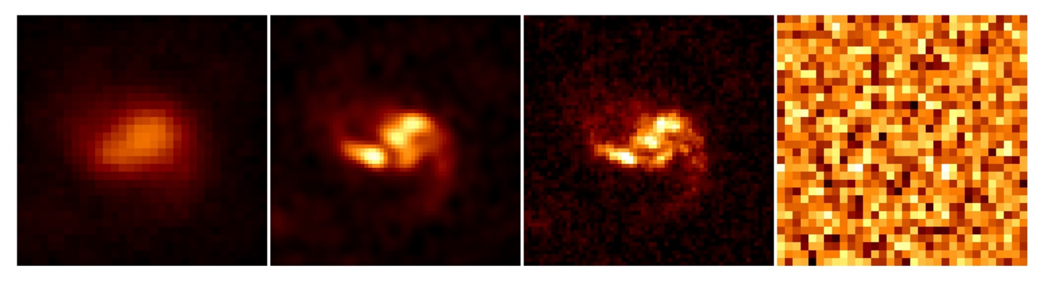
Results



Joint multi-channel deconvolution for Euclid and LSST, *Akhaury et al, in prep*

Flux Recovery





Questions?

utsav.akhaury@epfl.ch