ElFNet: Ellipticity Finding Network

Lucy Reynolds

with Martin Eriksen, Cristobal Padilla, Marc Manera

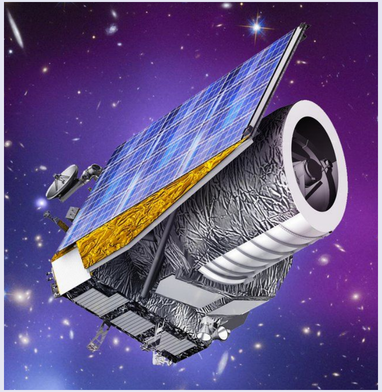
Euclid consortium

IFAE Institut de Física d'Altes Energies

EXCELENCIA SEVERO OCHOA

UAB Universitat Autònoma de Barcelona

Motivation for Euclid



Euclid Consortium

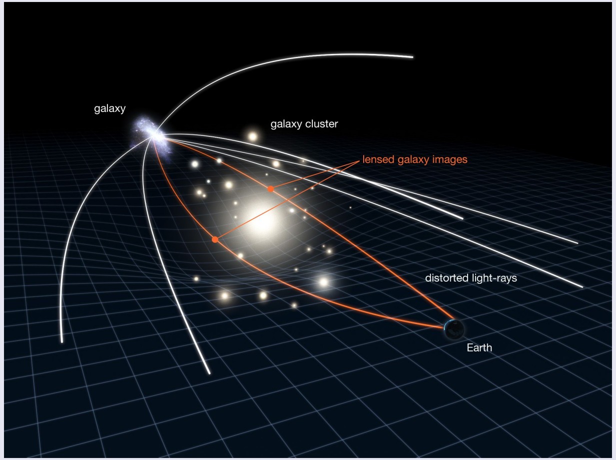
$\sigma_\mu \leq 2 \times 10^{-3}$

$$\gamma_i^{obs} = (1 + \mu_i)\gamma_i^{true} + c_i$$

$\sigma_c \leq 5 \times 10^{-4}$

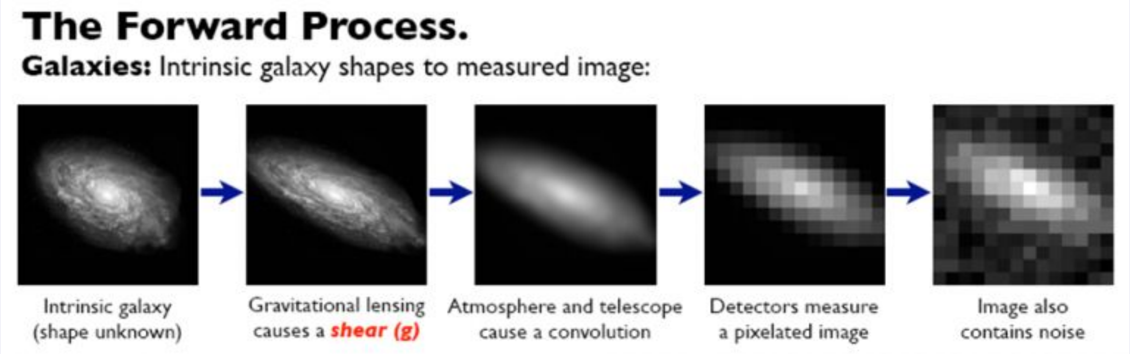
Noise bias is a significant component of of σ_μ → **denoising** images before shear measurement could reduce bias.

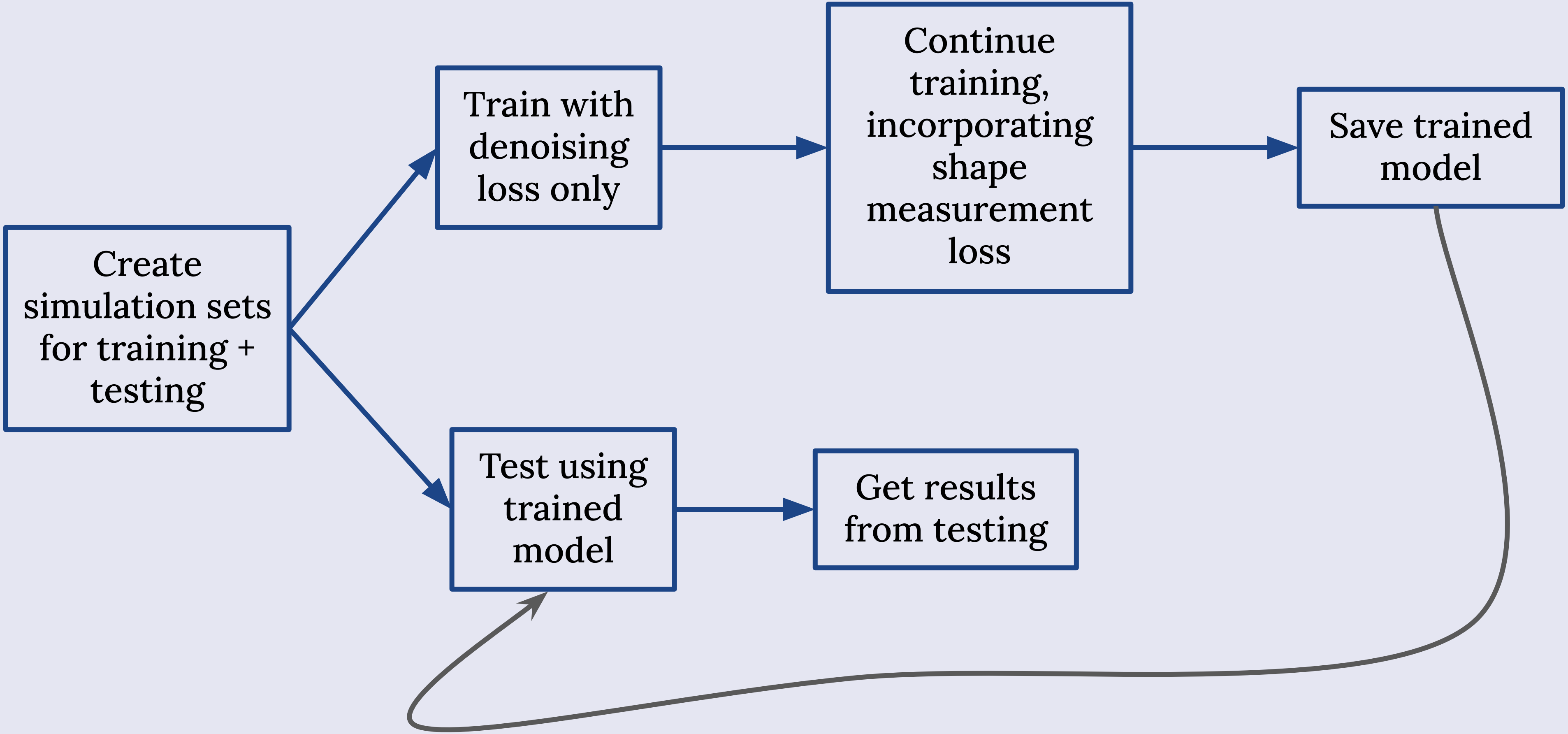
Cosmic Shear Measurement



Cosmic shear is the coherent distortion of galaxies in the sky due to large scale structure. It cannot be observed visually (like strong lensing) so we must average measurements over large areas of the sky.

Cosmic shear traces LSS growth → probe cosmological parameters & dark energy.

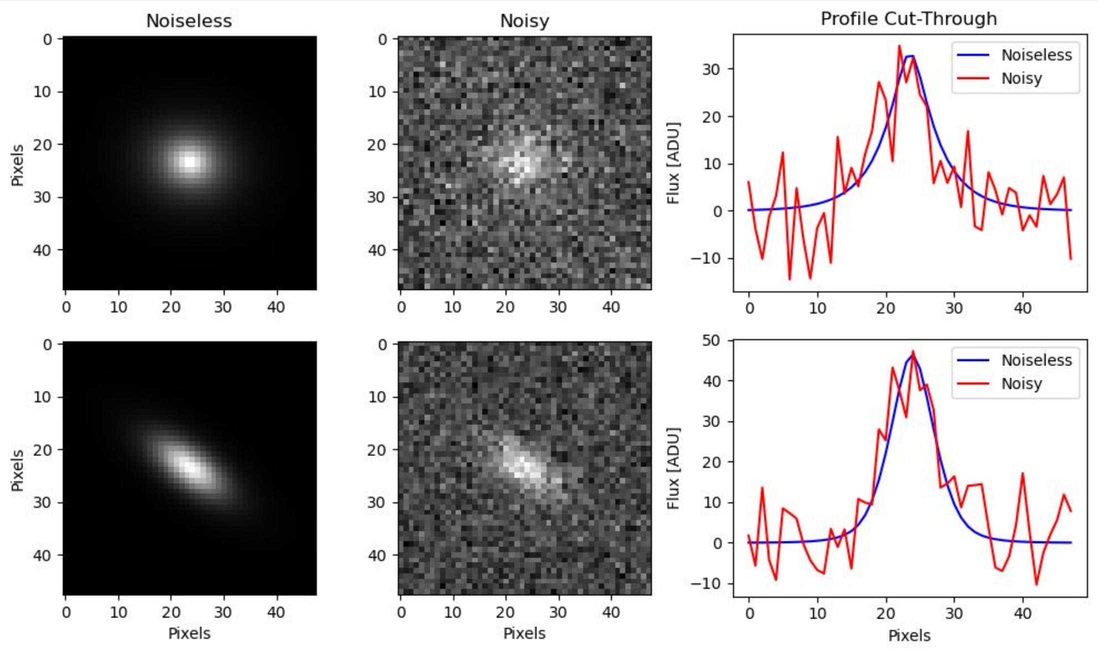




Create simulation sets for training + testing
Train with denoising loss only
Continue training, incorporating shape measurement loss
Save trained model
Test using trained model
Get results from testing

Simulations

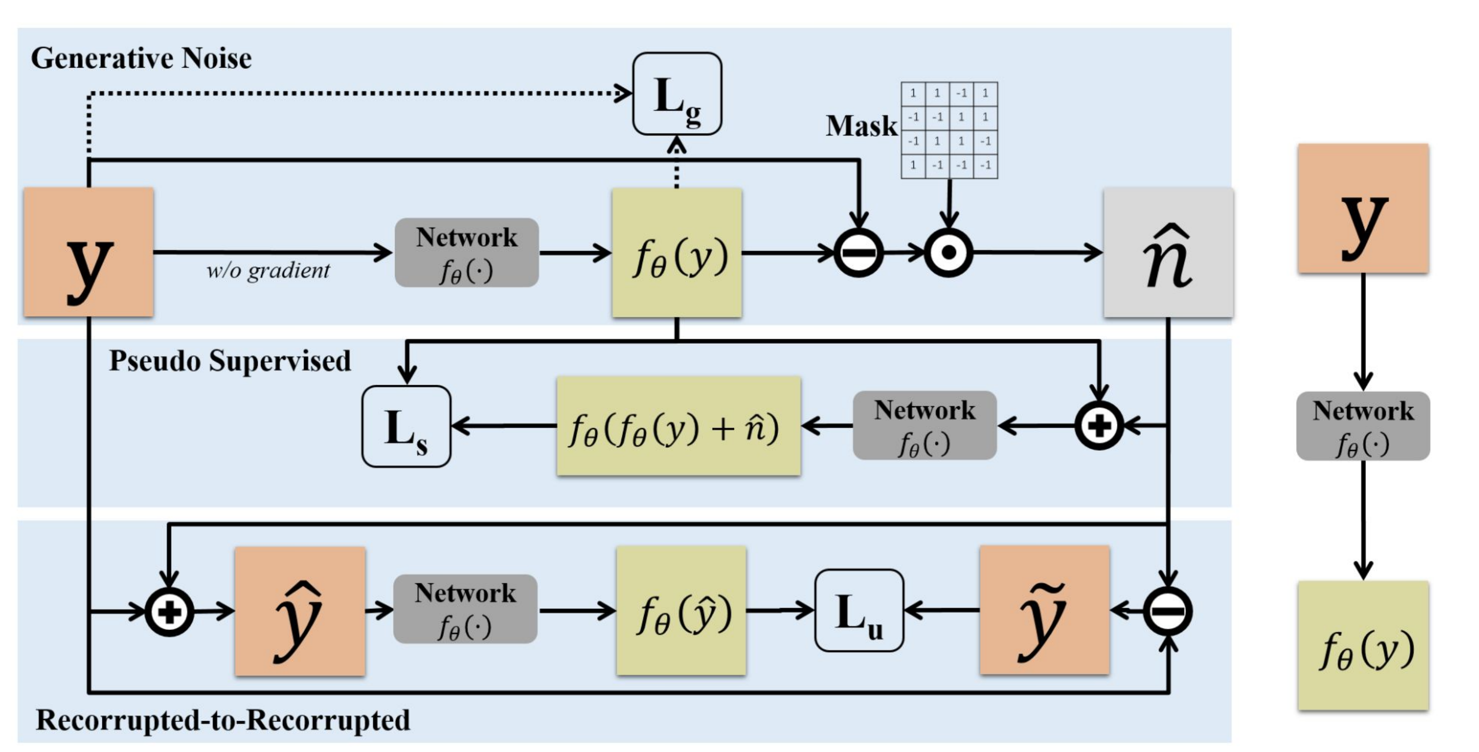
- Generate 48x48 pixels postage stamps with **GalSim** (Rowe et al. 2015)
- 2 component: bulge + disk
- Galaxy properties (e.g. scale-lengths, intrinsic ellipticities, shears, fluxes, etc.) taken from Flagship 2 catalogue (Castander et al. 2024)
- $20 < M_{AB} < 24.5$
- Noise:



$$n^{ADU}_{Poisson} = (R_{bkg} + R_{dark}) \cdot \frac{\tau}{g}$$

$$n^{ADU}_{Gaussian} = \sigma_{read} = 4.5\frac{e^-}{g}$$

Generative Recorrupted to Recorrupted (GR2R) - Pang et al. 2021



(a) Training Phase (b) Inference Phase

Pang et al. 2021

$$y = x + n$$

1. GR2R: Generative Noise

1. Generate residual image (approximates the noise level):

Original noisy image

Denoised image

$$y - f_\theta(y)$$

2. Generate random mask:

$$m = \begin{cases} 1 & \text{with p} = 0.5 \\ -1 & \text{with p} = 0.5 \end{cases}$$

3. Multiply residual & mask (mask reduces image feature information):

$$\hat{n} = (y - f_\theta(y)) \odot m$$

Element-wise multiplication

Generative Noise Loss:

$$L_g = \frac{1}{n} \Sigma_{i=1}^{n} [f_\theta(y_i) - y_i]^2$$

2. GR2R: Pseudo-Supervised

The pseudo-supervised module suppresses any error due to biases in estimating the true noise model in step 1.

Pseudo-Supervised Loss:

Denoised image

$$L_s = \gamma \cdot \frac{1}{n} \Sigma_{i=1}^{n} [f_\theta(f_\theta(y_i) + \hat{n}) - f_\theta(y_i)]^2$$

hyperparameter (controls strength)

Generative noise

With no bias, the two terms should be the same.

3. GR2R: Recorrupted-to-Recorrupted (R2R)

Create 2 corrupted images by adding/subtracting generated noise to input.

$$\hat{y} = y + \hat{n}$$
$$\tilde{y} = y - \hat{n}$$

f_θ: denoising network

R2R Loss:

$$L_u = \frac{1}{n}\Sigma_{i=1}^{n}[f_\theta(\hat{y}_i) - \tilde{y}_i]^2$$

R2R loss can be shown to be equivalent to the supervised denoising loss.

Advantages of GR2R

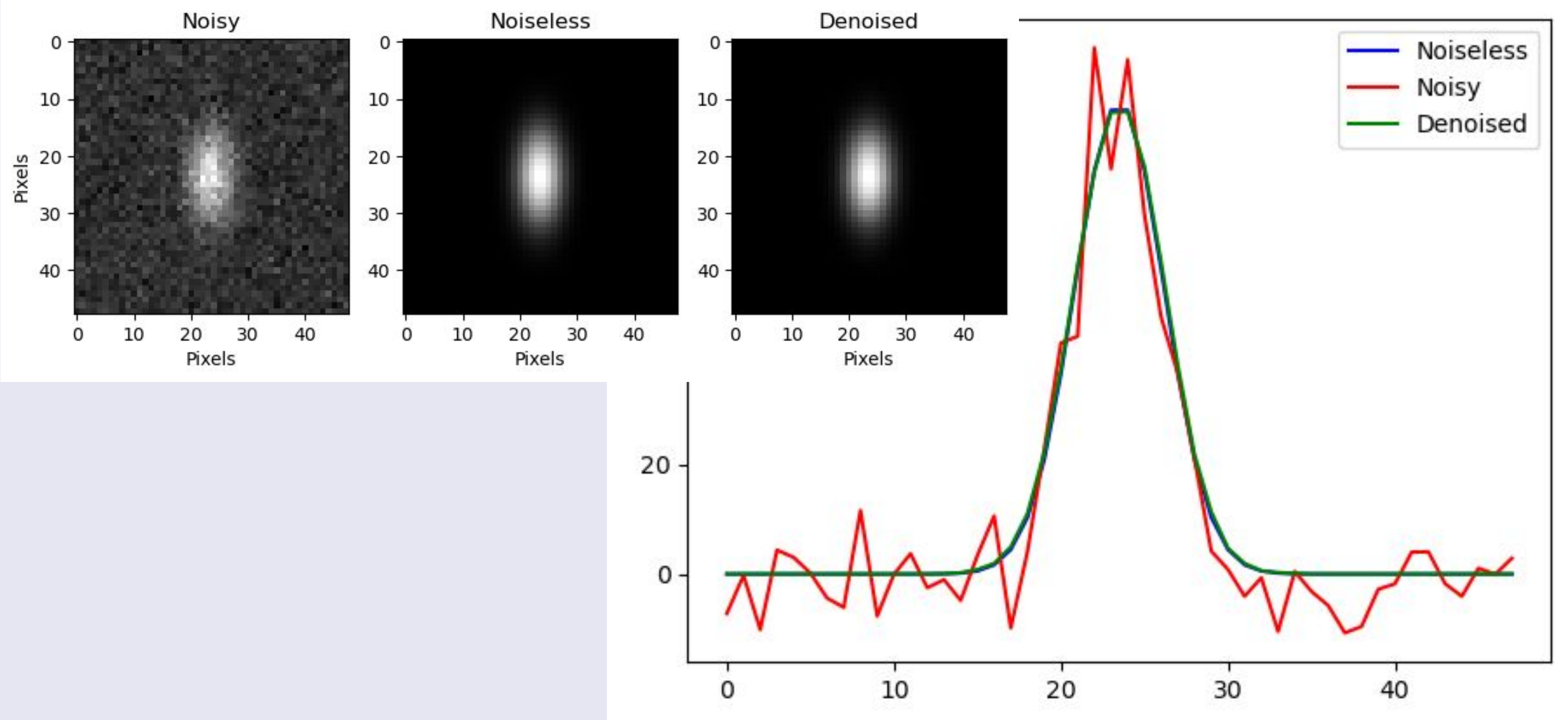
- Only requires original, noisy images (no clean targets/ labels needed)
- Does not assume prior noise model
- R2R loss equivalent to supervised learning loss

$$L_{denoising} = L_u + L_g + \gamma \cdot L_s$$

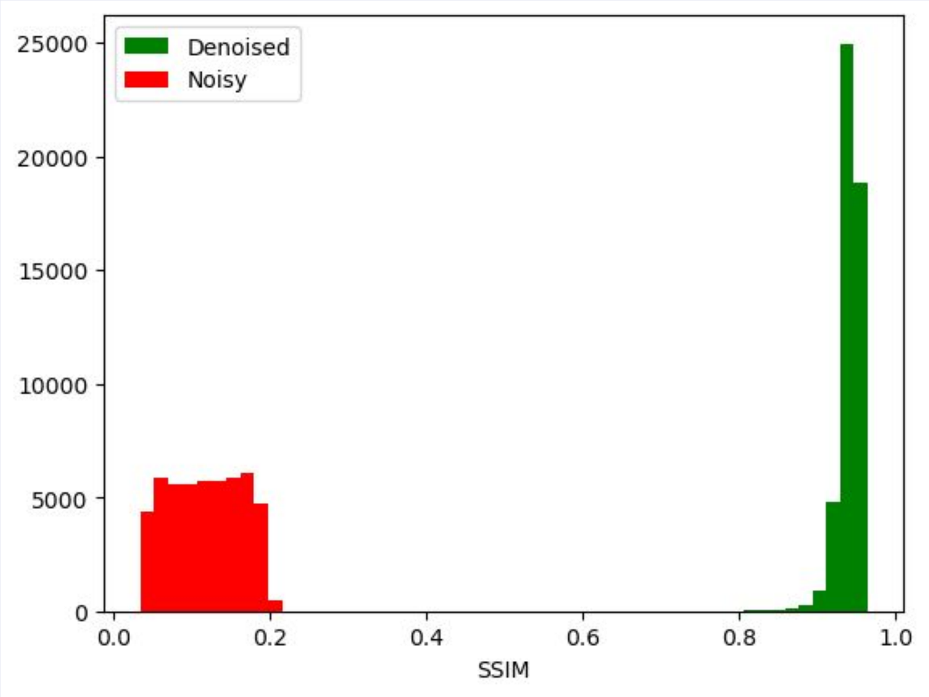
$$= \underbrace{\frac{1}{n}\Sigma_{i=1}^{n}[f_\theta(\hat{y}) - \tilde{y}]^2}_{\text{R2R Loss}} + \underbrace{\frac{1}{n}\Sigma_{i=1}^{n}[f_\theta(y) - y]^2}_{\text{GN Loss}} + \gamma \cdot \underbrace{\frac{1}{n}\Sigma_{i=1}^{n}[f_\theta(f_\theta(y) + \hat{n}) - f_\theta(y)]^2}_{\text{PS Loss}}$$

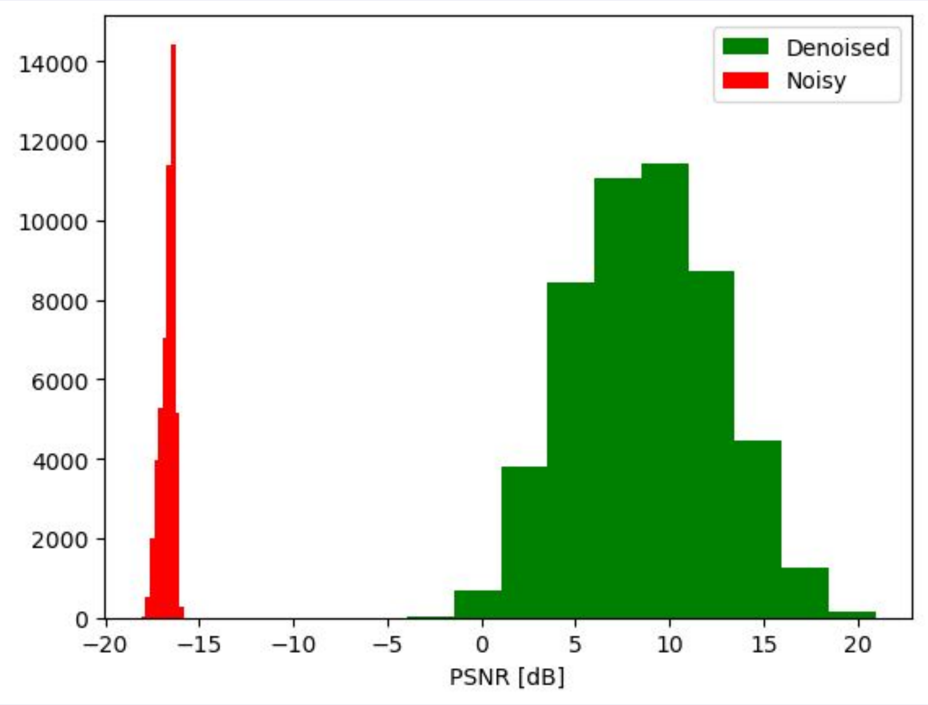
R2R Loss **GN Loss** **PS Loss**

Denoising Results (Gaussian Profiles)

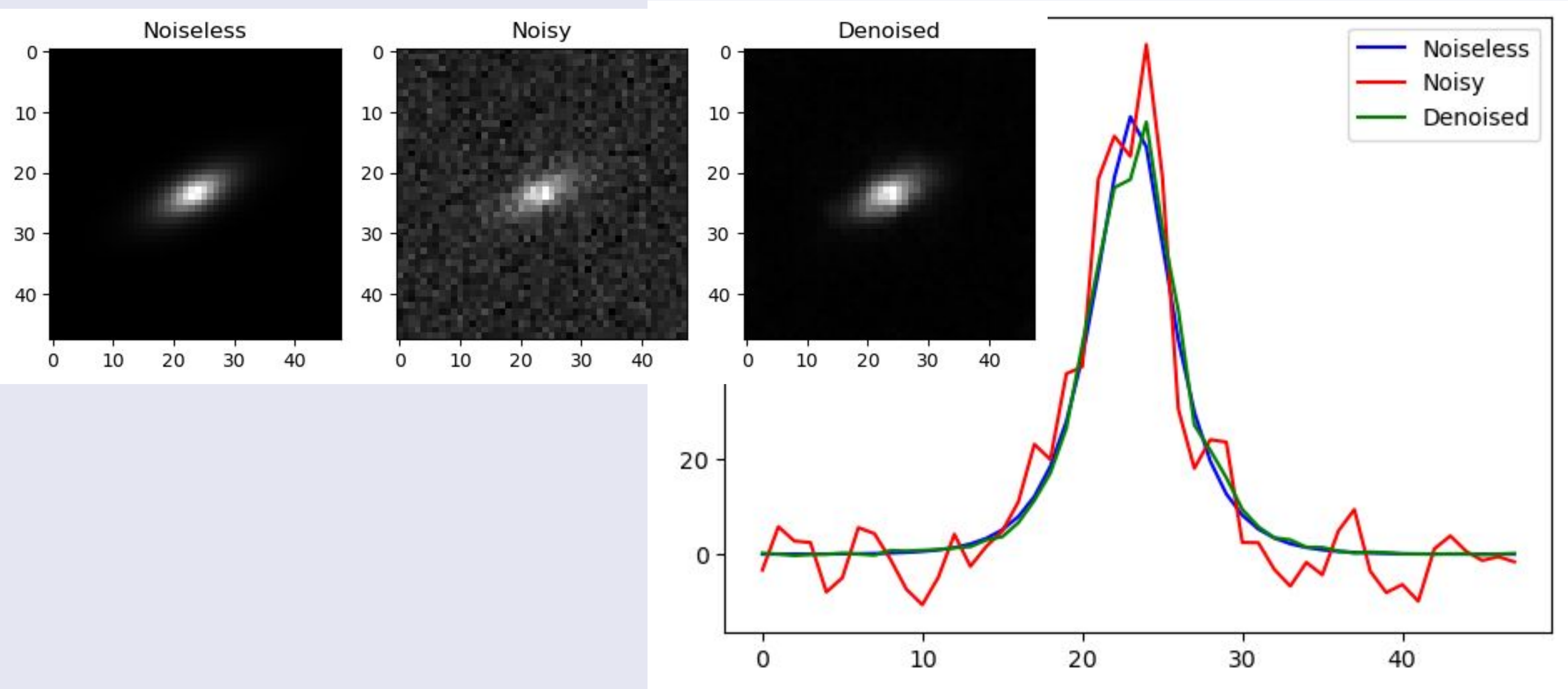


Denoising Metrics (Gaussian Profiles)

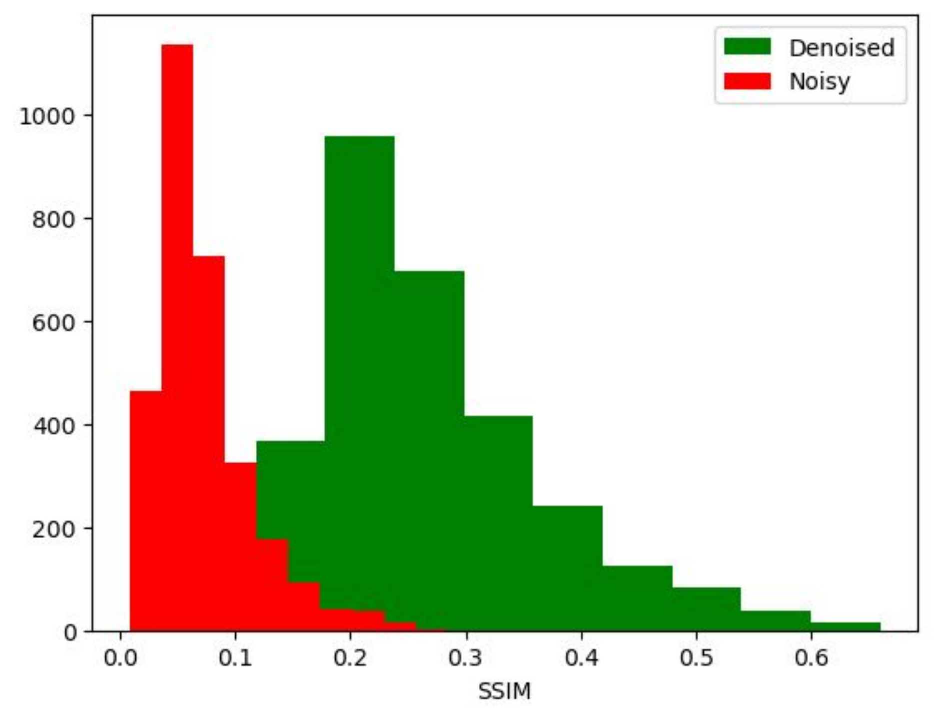


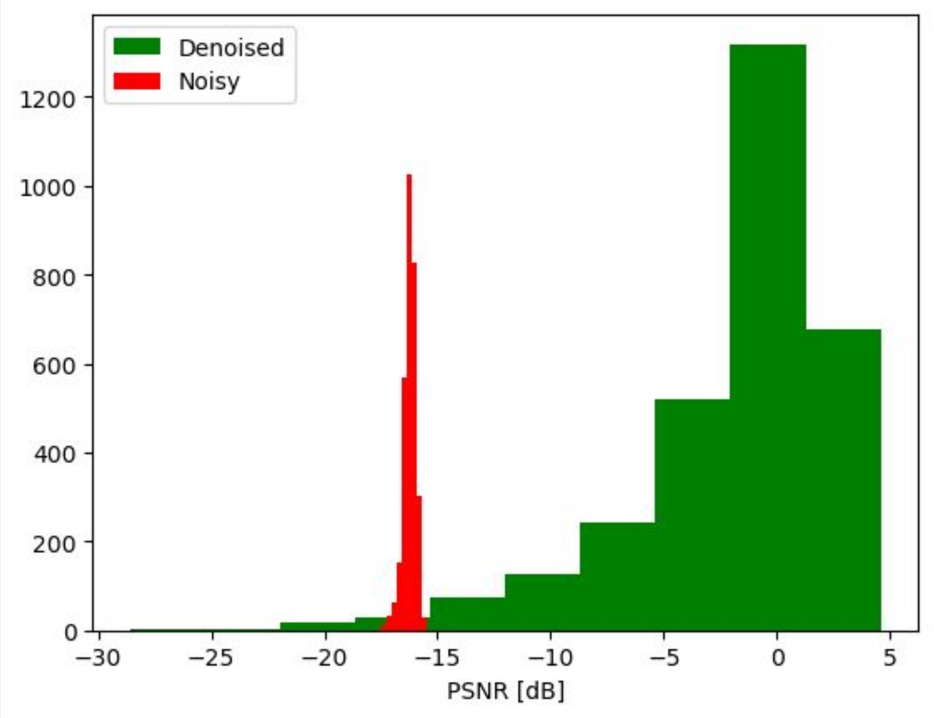


(Preliminary) Denoising Results (bulge+disk)



Denoising Metrics (bulge+disk)





Simultaneous Denoising & Shape Measurement Loss

Denoising is not made to optimise shear specifically

➔ Introduce shear measurement into the loss function (differentiable Metacalibration, using a loss function inspired by GR2R framework)

$$L_{shear} = \frac{1}{n}\Sigma_{i=1}^{n}[S(f_\theta(\hat{y}_i)) - S(\tilde{y}_i)]^2$$

$$\hat{y} = y + \hat{n}$$
$$\tilde{y} = y - \hat{n}$$

Metacalibration

1. Create 5 counterfactual images per galaxy (+g1,-g1,0,+g2,-g2):

Deconvolution of PSF → Shear image (incl. oversampling) → Reconvolve Image with Dilated PSF

2. Measure shape & calculate derivatives of response:

$$e_1 = \frac{M_{00} - M_{11}}{M_{00} + M_{11}} \quad e_2 = \frac{2M_{01}}{M_{00} + M_{11}} \longrightarrow R = \frac{\partial \mathbf{e}}{\partial \mathbf{g}} = \frac{\mathbf{e}^{+} - \mathbf{e}^{-}}{2\Delta \mathbf{g}}$$

3. Correct shear:

$$\gamma = \langle R \rangle^{-1} e$$

Counterfactual Image Creation (in PyTorch)

1. Wiener deconvolution in Fourier Space (reduces noise at high frequencies):

$$\tilde{G}(\mathbf{k}) = \frac{\tilde{I}(\mathbf{k})\tilde{P}(\mathbf{k})^*}{\tilde{P}(\mathbf{k})^*\tilde{P}(\mathbf{k})+n}$$

2. Shearing (+ oversampling) using Lanczos (L):

$$S = \begin{pmatrix} 1+g_1 & g^2 \\ g^2 & 1-g_1 \end{pmatrix}$$

$$S = S_x S_y = \begin{pmatrix} S_{xa} & S_{xb} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ S_{ya} & S_{yb} \end{pmatrix}$$

3. Reconvolve with dilated PSF:

$$\Gamma(x) = P((1+2|\gamma|)^{-1}x)$$

4. Overall Procedure:

$$I'(x|g) = \Gamma * [L(P^{-1} * I)]$$

Work so far:

- The denoising training framework setup
- 2 component Sersic galaxy simulations
- The bulk of Metacalibration framework complete in PyTorch (could add some other corrections)
- PSF correction code to the shape measurement is written (just not implemented-yet)

Work in the Future:

- **Get training working** optimally for denoising + shape measurement together
- Decide and implement binning scheme for getting shear
- Train & implement PSF correction within shape measurement
- Possibly increase simulation complexity

✨✨Thank you for listening !

Applying Shear on Deconvolved Images

With Lanczos interpolation for all steps:

1. Oversampling
2. Shearing (apply shear separately in x- and y-dimensions

$$S = \begin{pmatrix} 1 + g_1 & g^2 \\ g^2 & 1 - g_1 \end{pmatrix}$$

$$S = S_x S_y = \begin{pmatrix} S_{xa} & S_{xb} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ S_{ya} & S_{yb} \end{pmatrix}$$

$$S_{ya} = g_2\,,$$
$$S_{yb} = 1 - g1\,,$$
$$S_{xb} = \frac{g_2}{1 - g_1}\,,$$
$$S_{xa} = 1 + g_1 - \frac{g_2^2}{1 - g_1}\,.$$

3. Downsampling

Denoising Metrics

$$PSNR = 10 \cdot \log_{10} \frac{MAX_x^2}{MSE}$$

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$