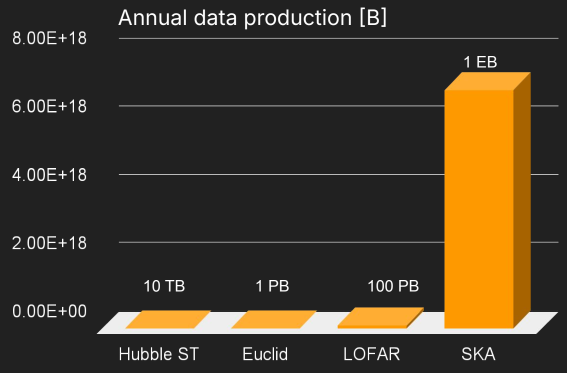
# Radio galaxy image generation with the scattering transform

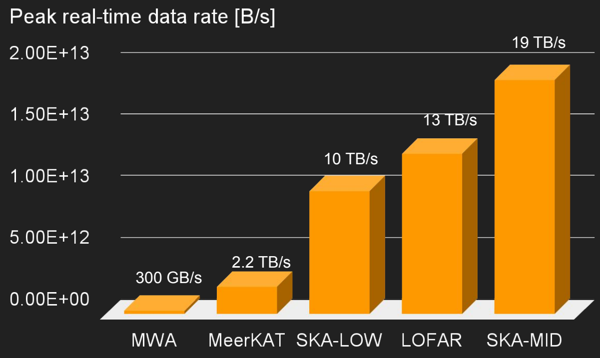
**<u>Markus Bredberg</u>**
markus.bredberg@epfl.ch
PhD student, LASTRO, EPFL,
Supervisor: Emma Tolley
Co-supervisor: Rémi Poitevineau
3rd of September 2024

EPFL

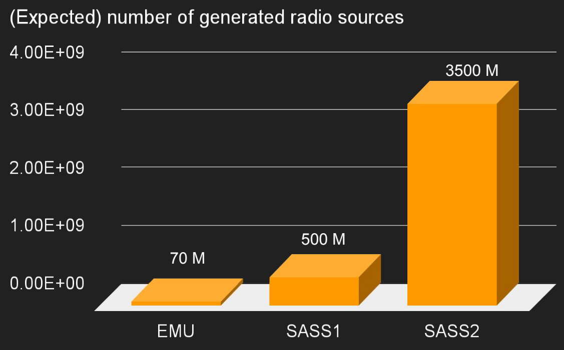
# With great telescopes comes great data challenges




Credit: NASA, ESA, IRA, SKAO




Lonsdale et al., 2009; Boot & Jonas, 2012; Labate et al., 2022; van Haarlem et al., 2013; Swart et al., 2022

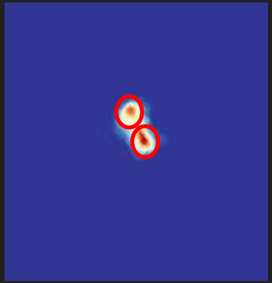



Norris et al., 2011; Norris et al., 2014;

# Machines learn differently from humans

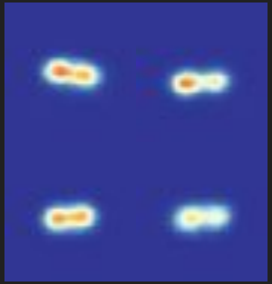
## Interpretation

## Biases



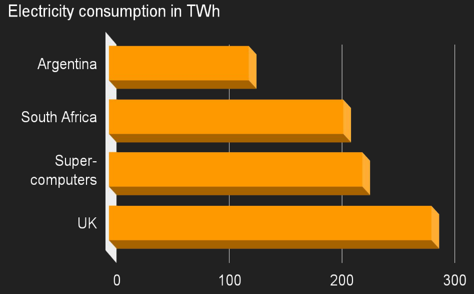
## Generalisation

## Reliability



## Scalability

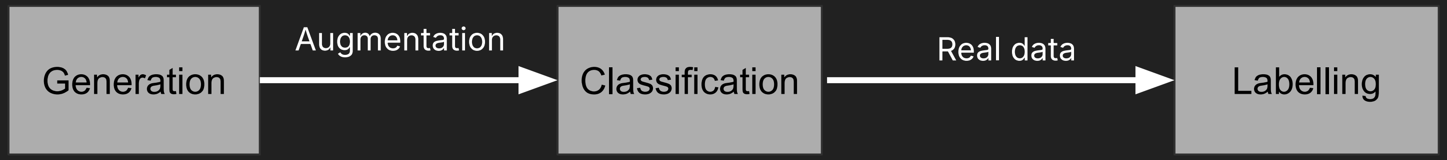
## Energy consumption




Credit: Enerdata, IEA, 2020

# Content

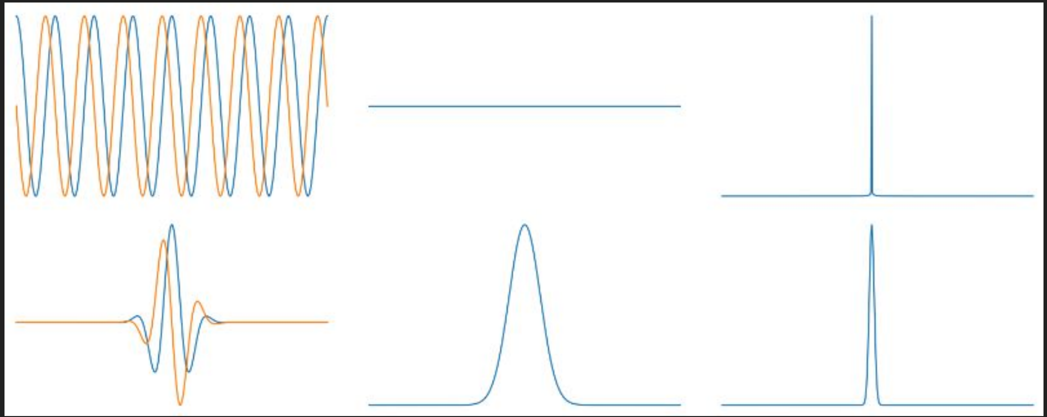
1. **The scattering transform** - A non-trainable CNN
2. **Generation** - Shallow and deep image generation
3. **Conclusions and outlook**



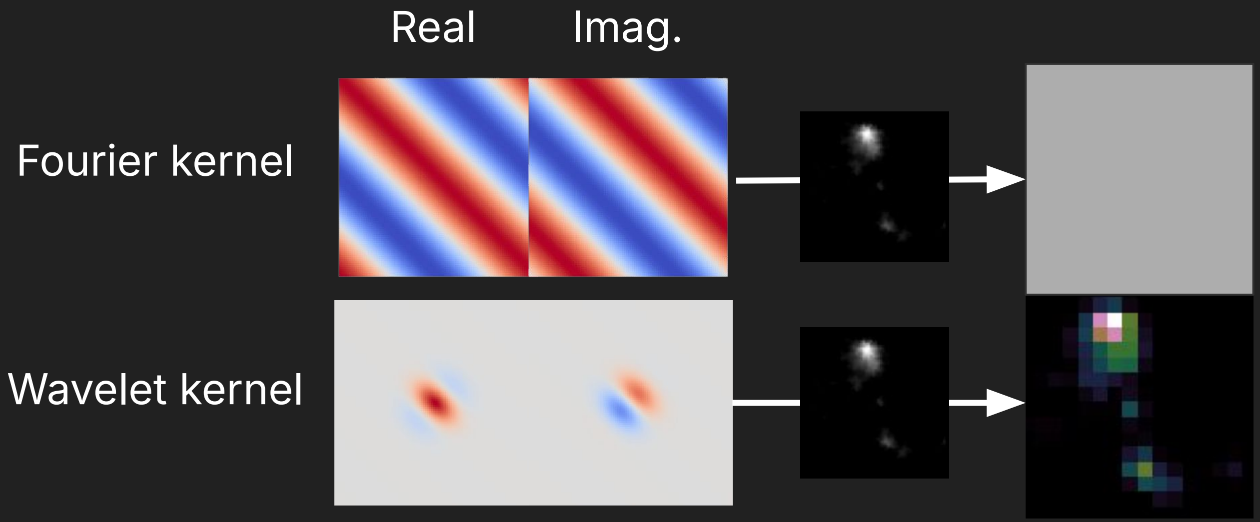
# Scattering transform

A non-trainable convolutional neural network

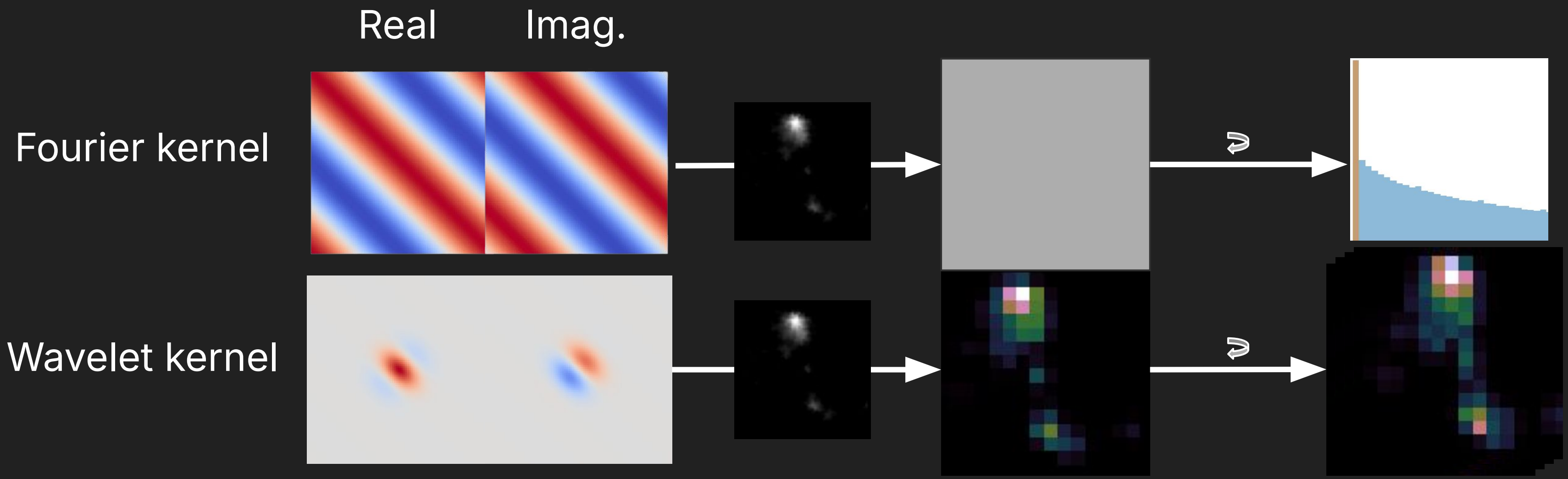
# The scattering transform uses wavelet kernels

| | Real part<br>Imaginary part | Modulus | Fourier domain | Units |
|---|---|---|---|---|
| Fourier kernel |  | | | Frequency |
| Wavelet kernel | | | | Wavelet coefficients |
| Convolutional kernel | Arbitrary form, given size | Arbitrary | Arbitrary | Arbitrary |

# Wavelet kernels are localised and extract features

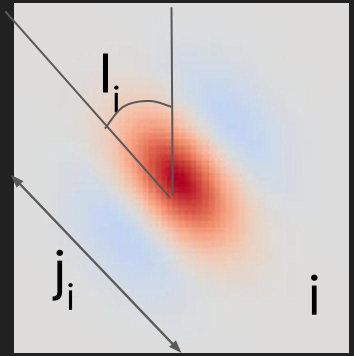


# Wavelet kernels are localised and extract features

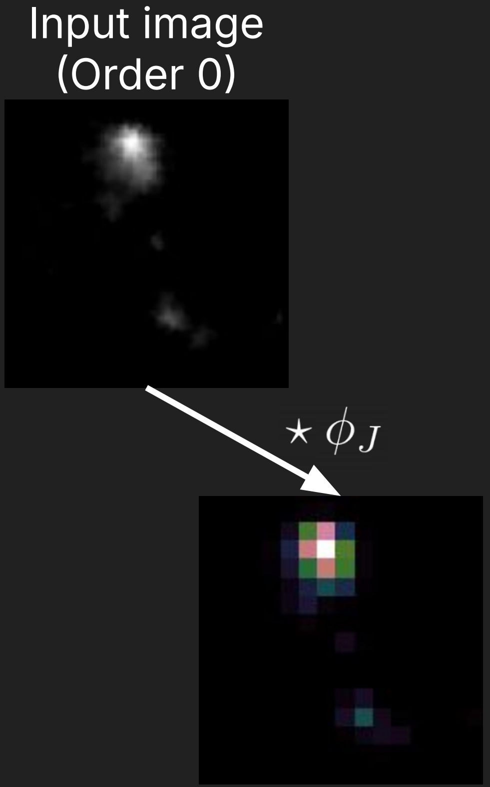


# Morlet wavelets are sinusoids with Gaussian envelopes

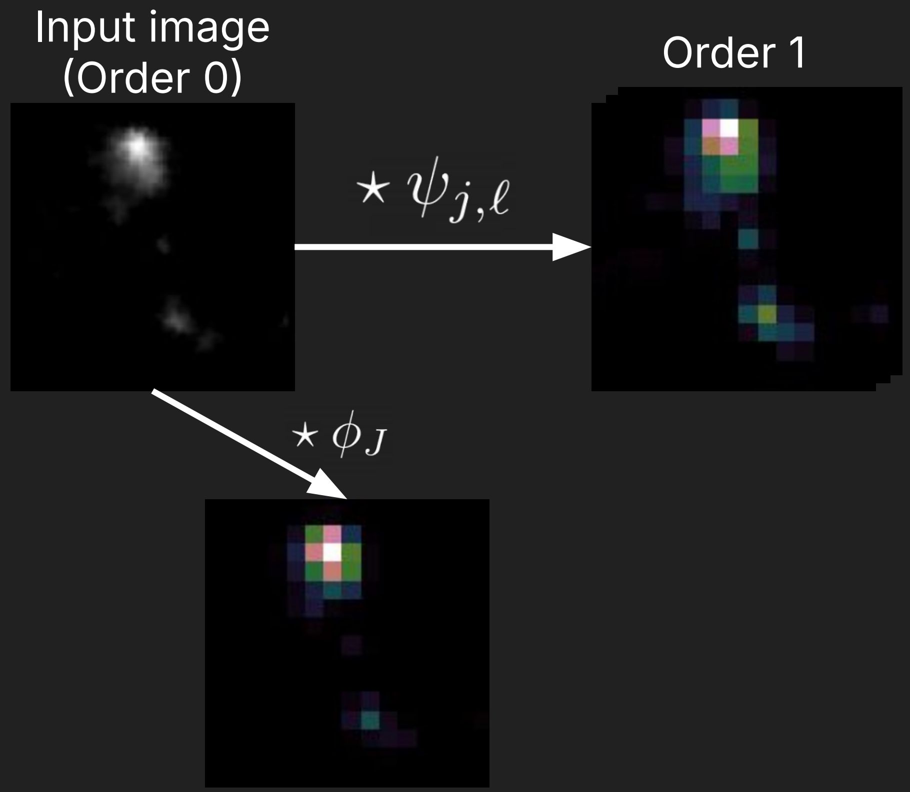
$$\psi_{j,\ell}(u) = 2^{-2j}\psi(2^{-j}r_\theta u) \quad \text{for} \quad 0 \leq \ell < L, \theta = \ell\pi/L$$



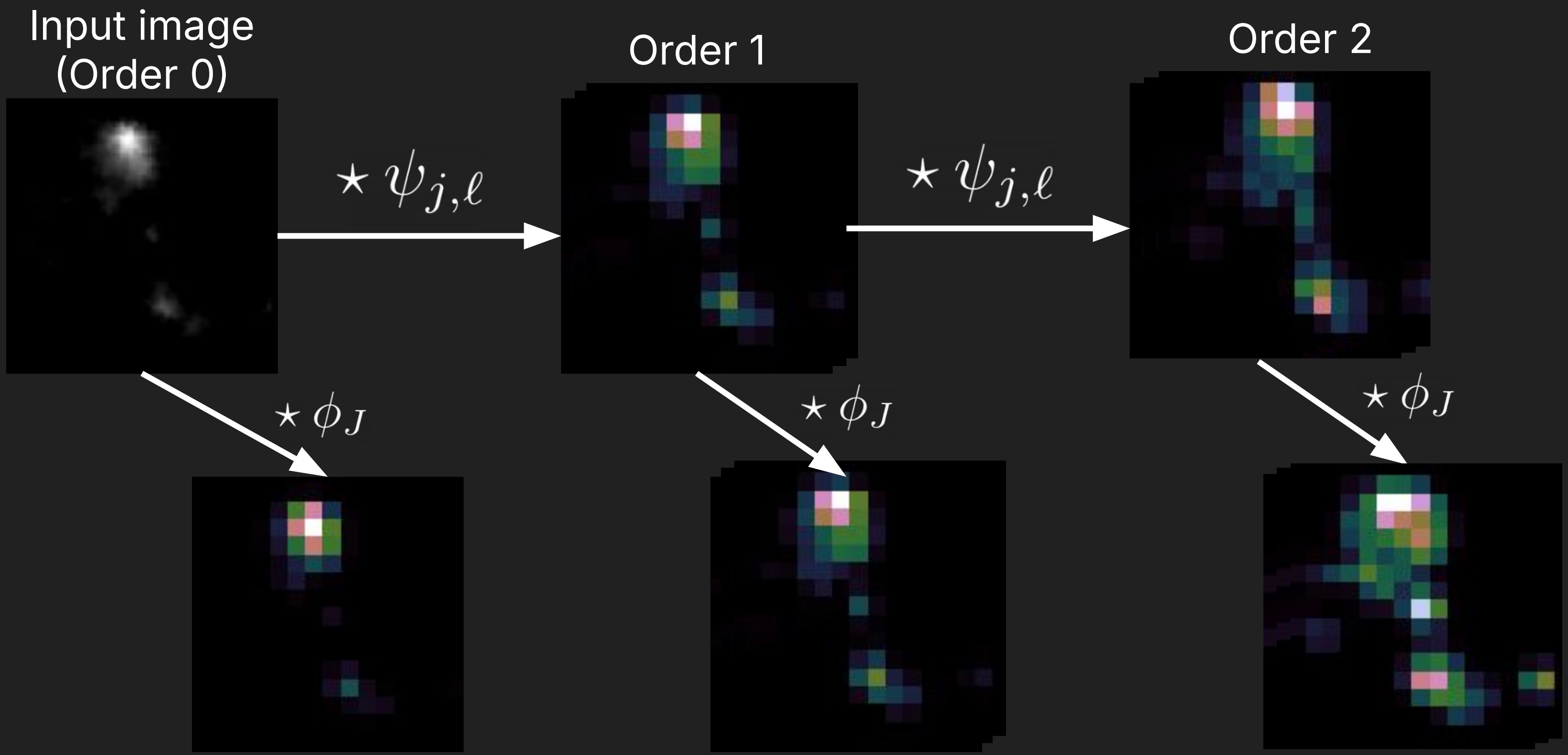
# Scattering transform is a cascade of wavelet transforms



# Scattering transform is a cascade of wavelet transforms



# Scattering transform is a cascade of wavelet transforms



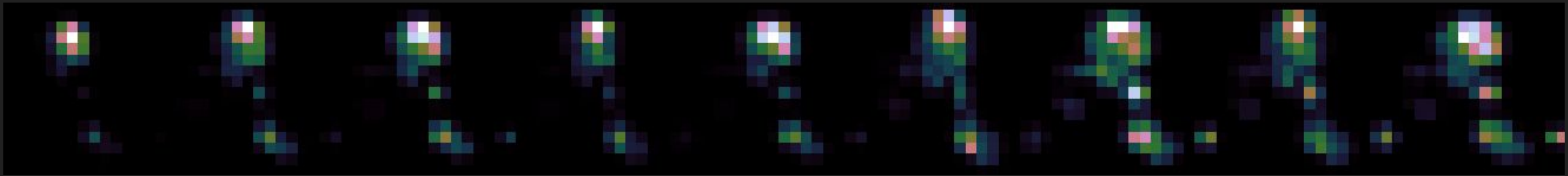
# Scattering transforms are iterative wavelet transforms

$$\Psi_{J,L}(x) = \left[X, |X \star \psi_{j,\ell}|, ||X \star \psi_{j,\ell}| \star \psi_{j',\ell'}|\right]_{1 \le \ell, \ell' \le L,\ 1 \le j < j' \le J}$$

$$\phi_J(u) = 2^{-2J} \phi(2^{-J} u)$$

$$\begin{aligned} S_{J,L}(X) &= \Psi_{J,L}(X) \star \phi_J \\ &= \left[X \star \phi_J, |X \star \psi_{j,\ell}| \star \phi_J, ||X \star \psi_{j,\ell}| \star \psi_{j',\ell'}| \star \phi_J\right]_{1 \le j < j' \le J, 1 \le \ell, \ell' \le L} \end{aligned}$$

$X \star \phi_2$ $|X \star \psi_{0,0}| \star \phi_2$ $|X \star \psi_{0,1}| \star \phi_2$ $|X \star \psi_{1,0}| \star \phi_2$ $|X \star \psi_{1,1}| \star \phi_2$ $||X \star \psi_{0,0}| \star \psi_{1,0}| \star \phi_2$ $||X \star \psi_{0,0}| \star \psi_{1,1}| \star \phi_2$ $||X \star \psi_{0,1}| \star \psi_{1,0}| \star \phi_2$ $||X \star \psi_{0,1}| \star \psi_{1,1}| \star \phi_2$

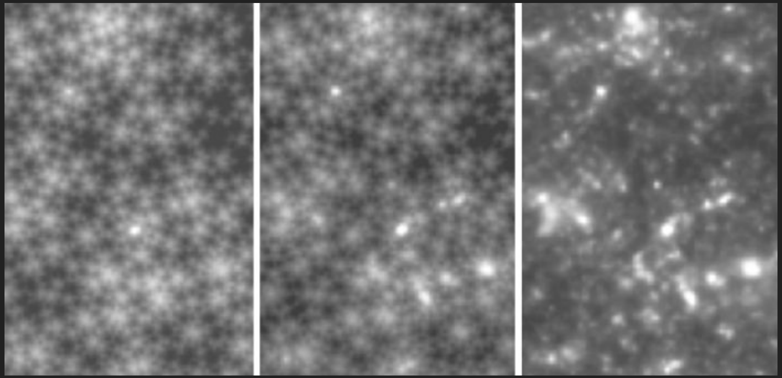


# The scattering transform is a simplified CNN or extended power spectrum

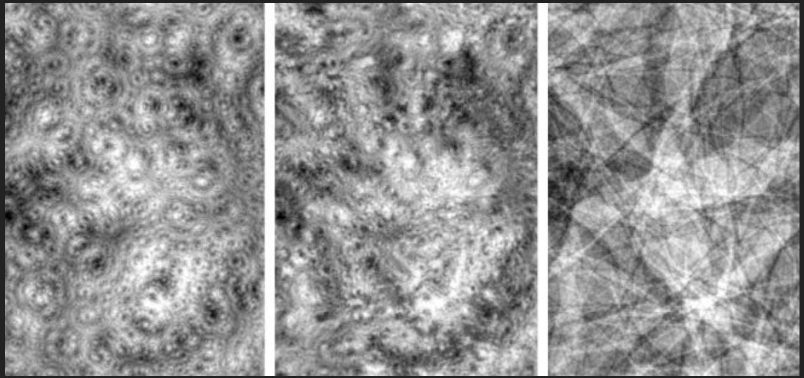
| | CNN | Power Spectrum | Scattering Transform |
|---|---|---|---|
| Convolutions | Feature convolutions | Filters | Localised kernels |
| Non-linear function | Activation function | Modulus squared | Modulus |
| Average | Pooling | Global average | Global average |
| Repetition | Multi-layer | Single layer | Iterations |

# The scattering transform extracts features

Feature sparsity $\langle S_2/S_1\rangle_{l_1,l_2}$



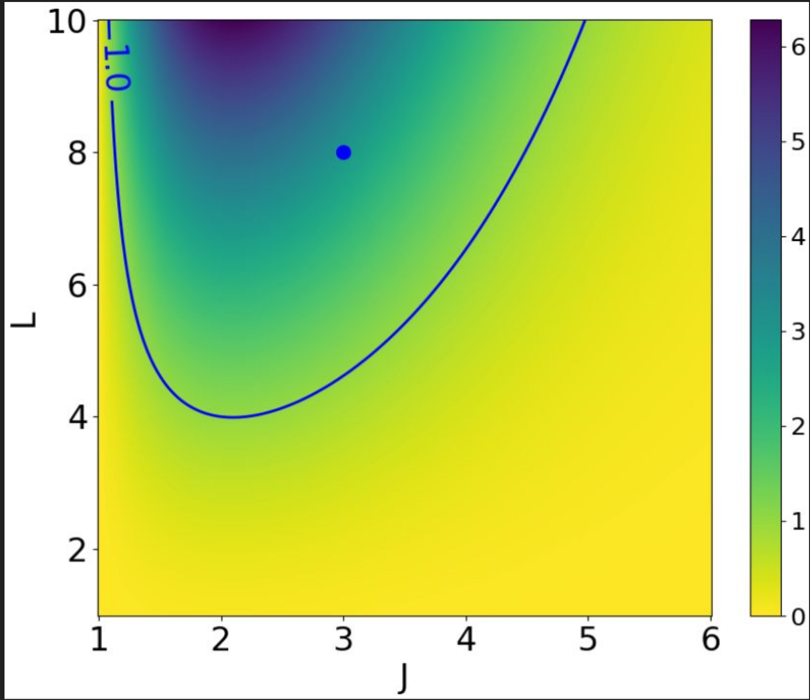
Wrinkliness $\langle S_2^{l_1=l_2}/S_2^{l_1\perp l_2}\rangle_{l_1}$



Cheng & Ménard, 2021, arXiv:2112.01288v1

# The scattering transform could be invertible

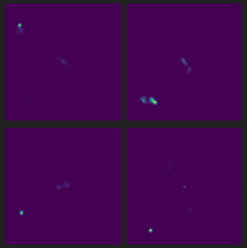
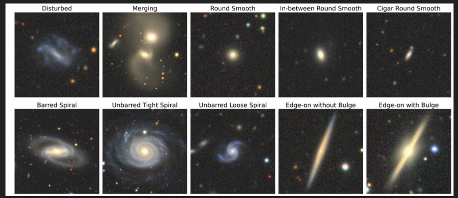
$$\frac{n_{\mathrm{ST}}}{n_{\mathrm{pixels}}} = 2^{-2J}\left(1 + LJ + L^2\frac{J(J-1)}{2}\right)$$



# Generation

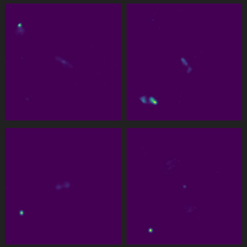
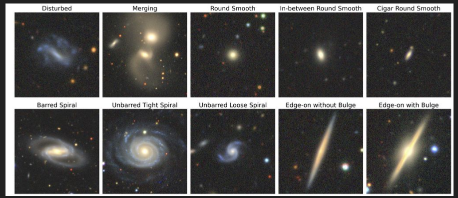
Shallow and deep image generation

# Datasets used for this work

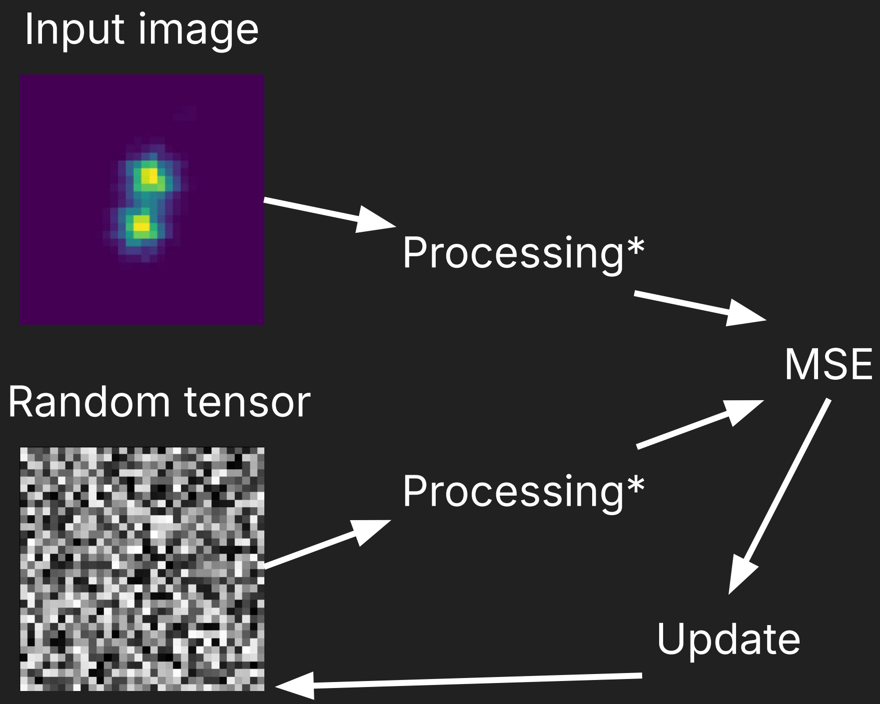
| **Input data** | MNIST | Mirabest | FIRST | Galaxy10 |
|---|---|---|---|---|
| **Image size** | 28x28 | 150x150 | 300x300 | 3x256x256 |
| **Number of samples per class** | 6000 | ~500 | ~500 | ~1000 |
| **Example images** |  |  |  |   |
| **# of classes** | 10 | 2 | 4 | 10 |
| **Source** | Deng 2012, doi.org/10.1109/MSP.2012.2211477 | Porter & Scaife 2023, arXiv:2305.11108v1 | Griese et al. 2023, doi.org/10.1016/j.dib.2023.108974 | Lintott et al. 2011, arXiv:1007.3265 |

# Datasets used for this work

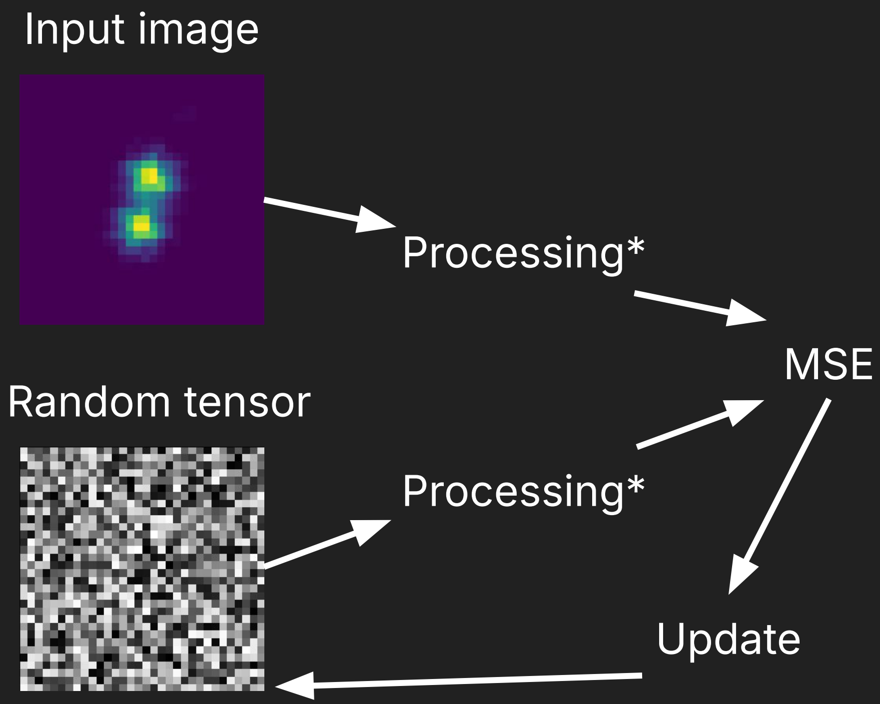
For generation:

| Input data | MNIST | Mirabest | FIRST | Galaxy10 |
|---|---|---|---|---|
| **Image size** | 28x28 | 150x150 | 300x300 | 3x256x256 |
| **Number of samples per class** | 6000 | ~500 | ~500 | ~1000 |
| **Example images** |  |  |  |  |
| **# of classes** | 10 | 2 | 4 | 10 |
| **Source** | Deng 2012, doi.org/10.1109/MSP.2012.2211477 | Porter & Scaife 2023, arXiv:2305.11108v1 | Griese et al. 2023, doi.org/10.1016/j.dib.2023.108974 | Lintott et al. 2011, arXiv:1007.3265 |

# Reconstruction from scattering coefficients without DL



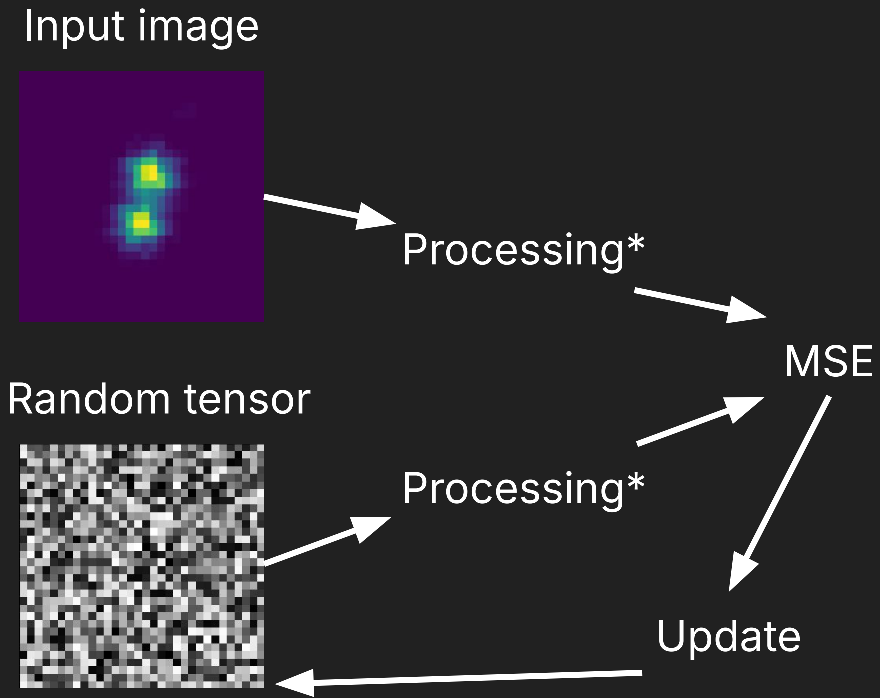
# Reconstruction from scattering coefficients without DL




* Processing:

1. Pixels: no processing
2. Full: scattering transform (ST)
3. lavg: ST + average across direction
4. ldiff: ST + subtract adjacent directions
5. javg: ST + average across scale

# Reconstruction from scattering coefficients without DL



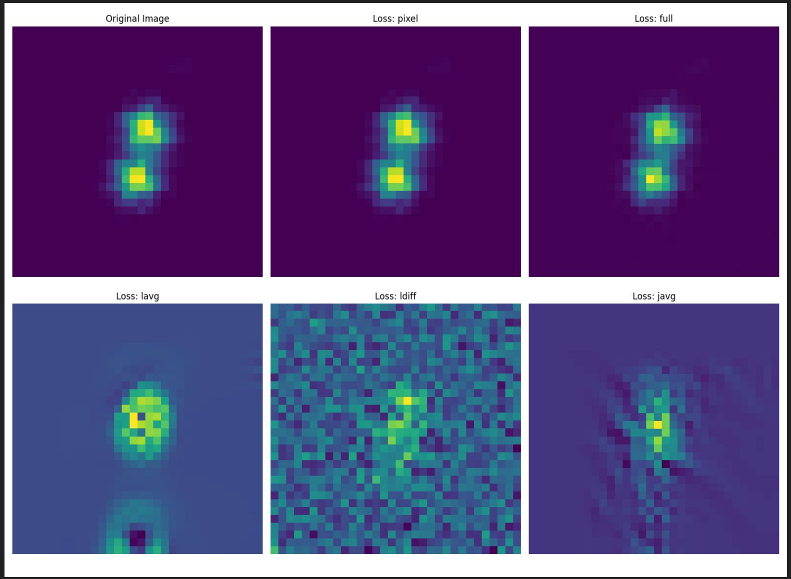

* Processing:

1. Pixels: no processing
2. Full: scattering transform (ST)
3. lavg: ST + average across direction
4. ldiff: ST + subtract adjacent directions
5. javg: ST + average across scale

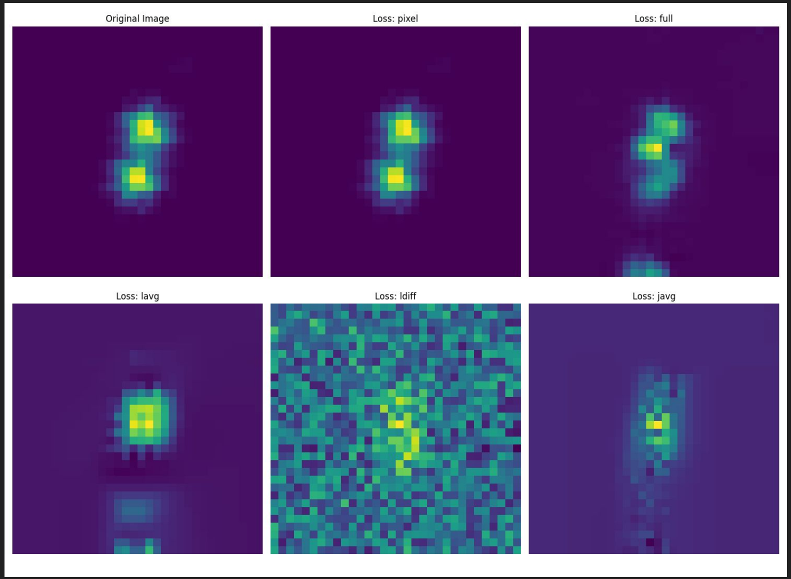

- PCA
- Average across space
- Average across space + direction
- Average across adjacent directions
- ...

# Numerical reconstruction is poor

J=3, L=8, 15,000 iterations, T=6h

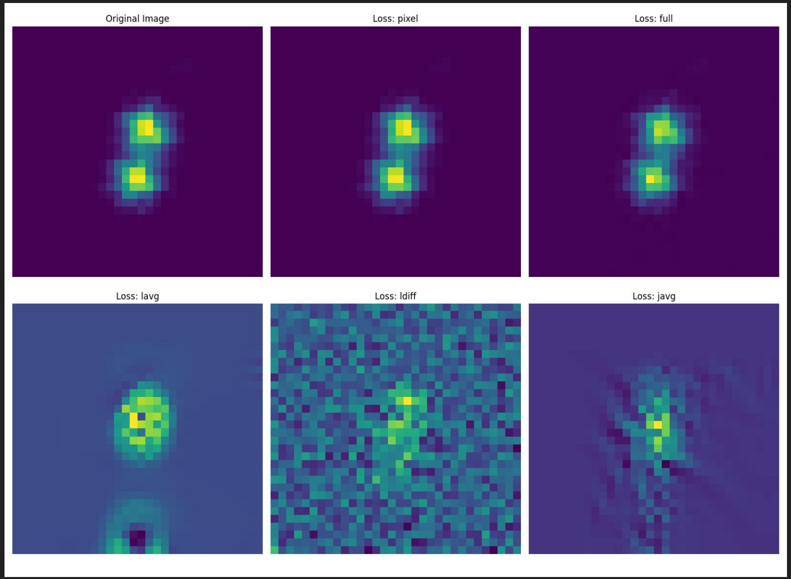



J=3, L=2, 15,000 iterations, T=0.5h

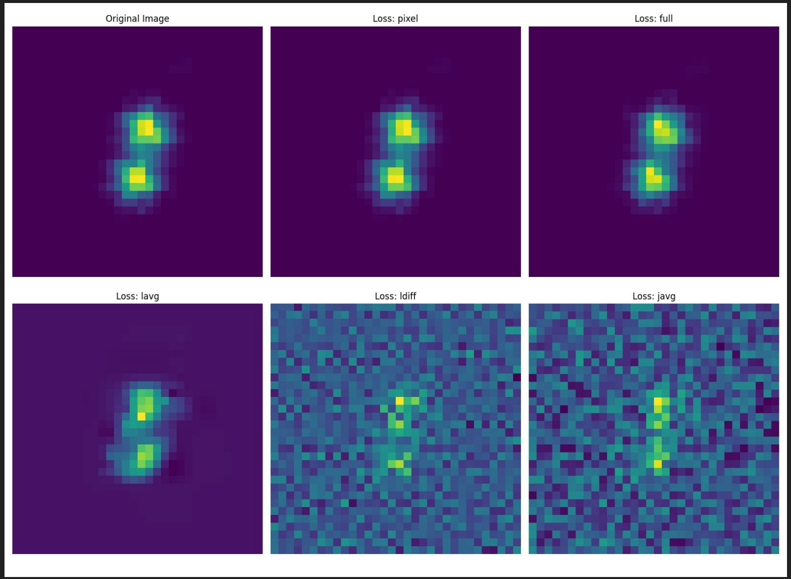


# Numerical reconstruction is poor

J=3, L=8, 15,000 iterations, T=6h

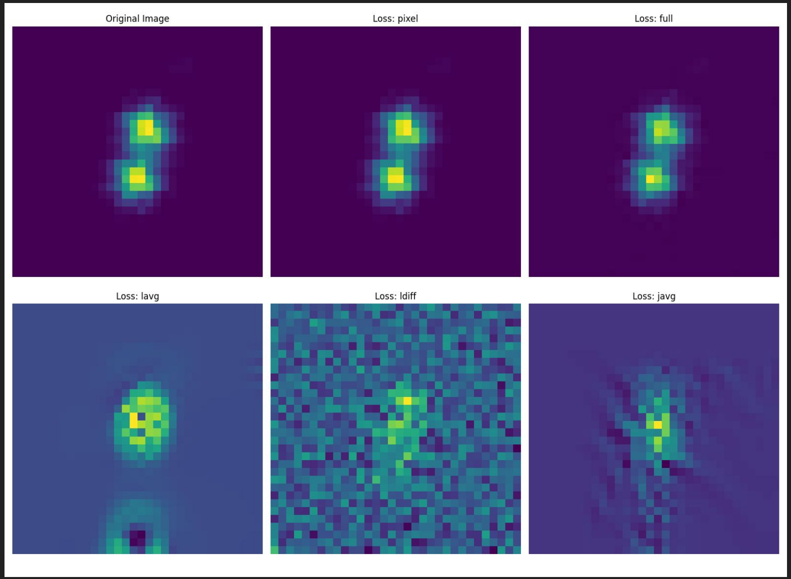



J=3, L=2, 50,000 iterations, T=2h

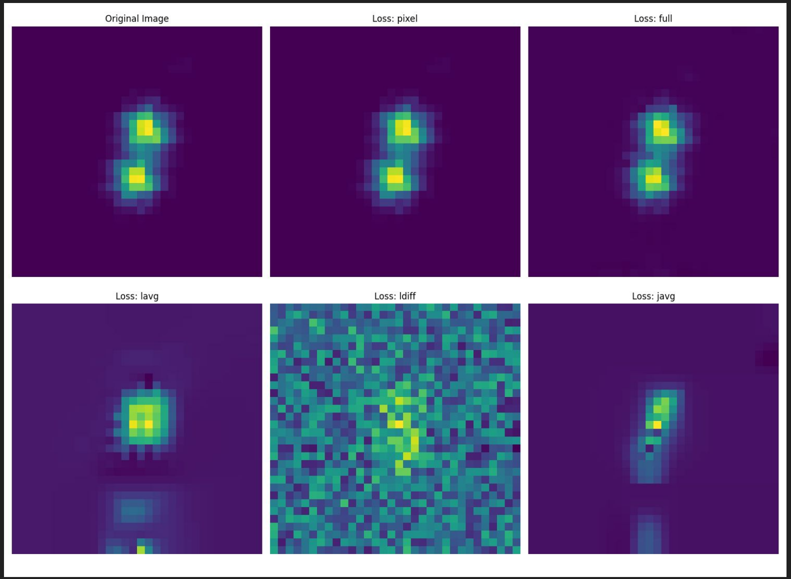


# Numerical reconstruction is poor

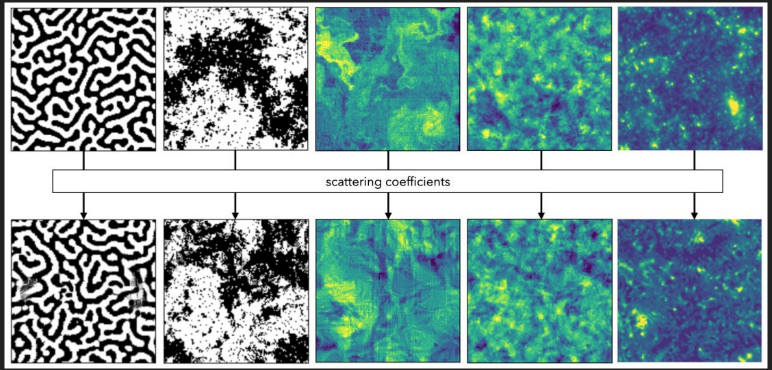
J=3, L=8, 15,000 iterations, T=6h

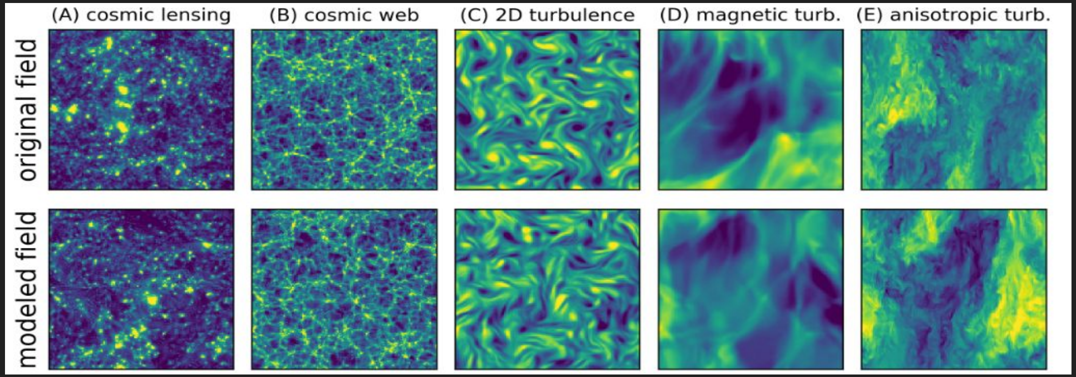



J=3, L=2, 100,000 iterations, T=4h



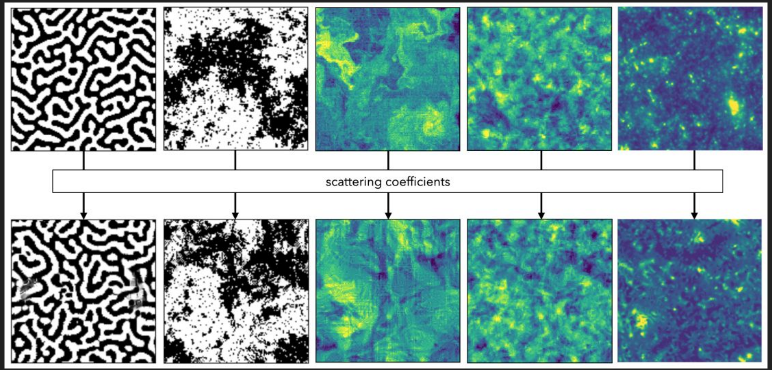
# The scattering transform reproduces fields and textures

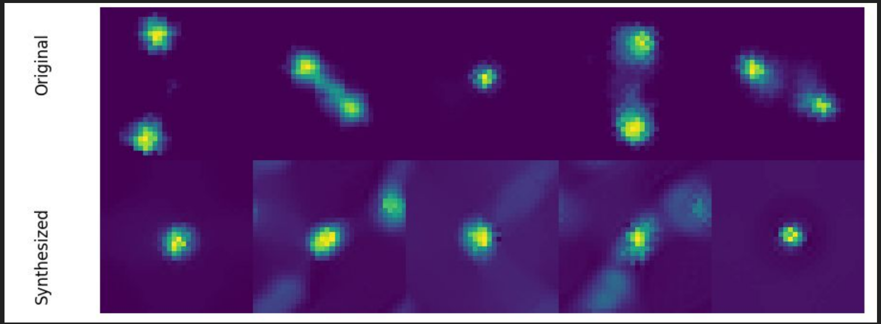


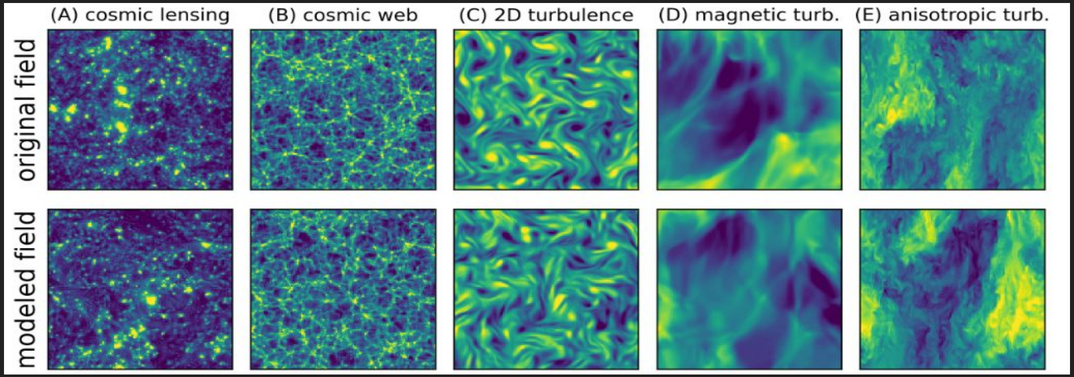




Cheng & Ménard, 2021,
arXiv:2112.01288v1

# The scattering transform reproduces fields and textures but not sparse sources



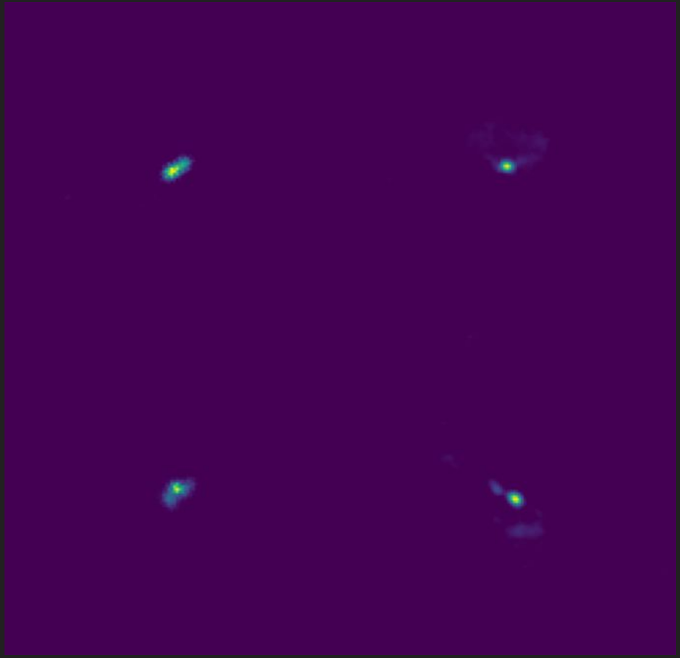



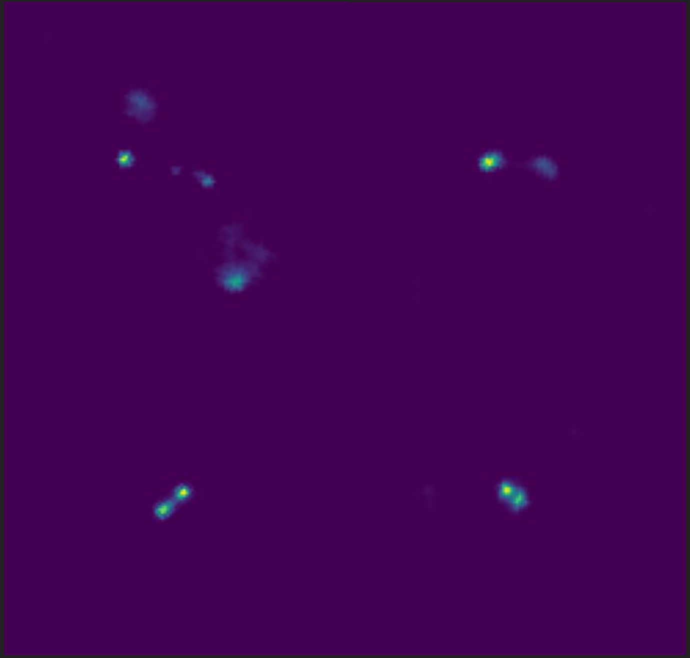

Cheng & Ménard, 2021,
arXiv:2112.01288v1

# Dimensional correlations demand variable mixing

FRI

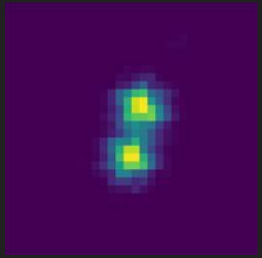


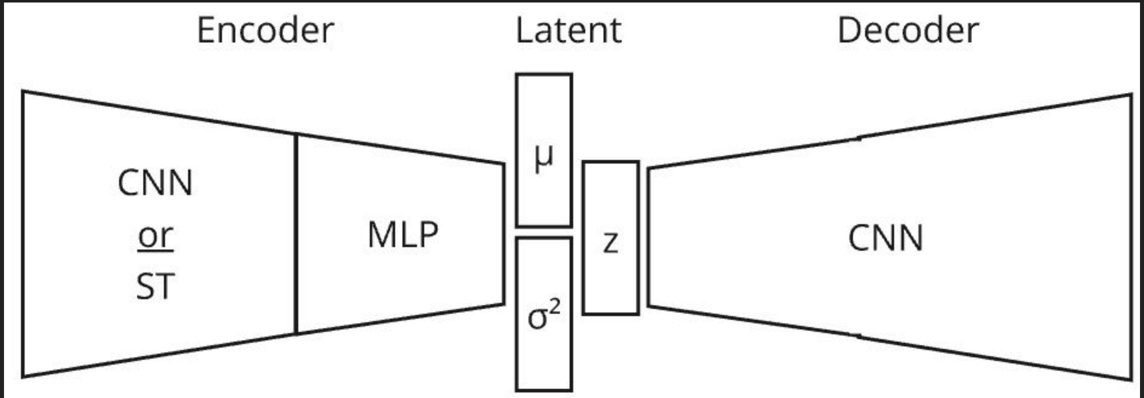
FRII



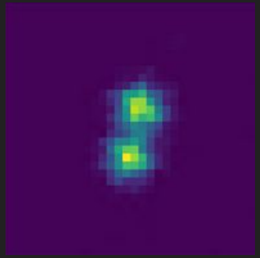
# Variational autoencoders sample features and construct images

X

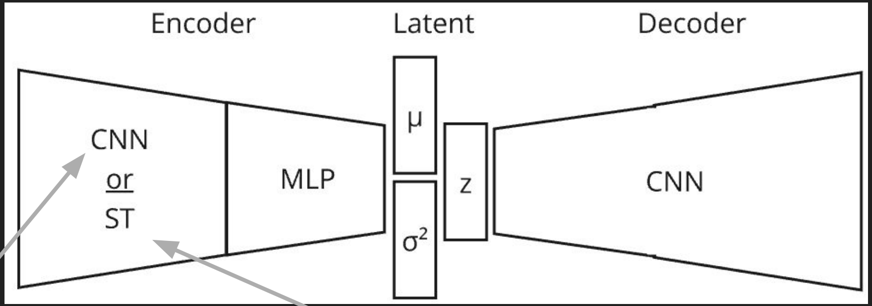





$\hat{X}$



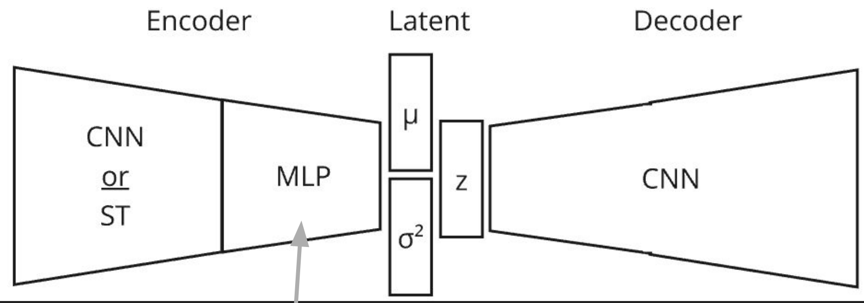
# The encoder learns spatial features, by first extracting shapes




$$\left[X \star \phi_J, |X \star \psi_{j,\ell}| \star \phi_J, ||X \star \psi_{j,\ell}| \star \psi_{j',\ell'}| \star \phi_J\right]_{1\le j<j'\le J, 1\le \ell,\ell'\le L}$$

| Layer | Component | Depth | Activation | Regulariser | Pooling | Parameters |
|---|---|---|---|---|---|---|
| 1-3 | 5x5 Convolution | 32 | Leaky ReLU | Batch norm. | 2x2 Max pooling | 19,520 |
| 4-6 | 5x5 Convolution | 64 | Leaky ReLU | Batch norm. | 2x2 Max pooling | 125,504 |
| 7-9 | 5x5 Convolution | 128 | Leaky ReLU | Batch norm. | 2x2 Max pooling | 353,244 |
| 10-12 | 5x5 Convolution | 256 | Leaky ReLU | Batch norm. | 2x2 Max pooling | 2,001,152 |
| 13-15 | 5x5 Convolution | 256 | Leaky ReLU | Batch norm. | 2x2 Max pooling | 2,820,352 |
| | | | | | | 5,319,772 |

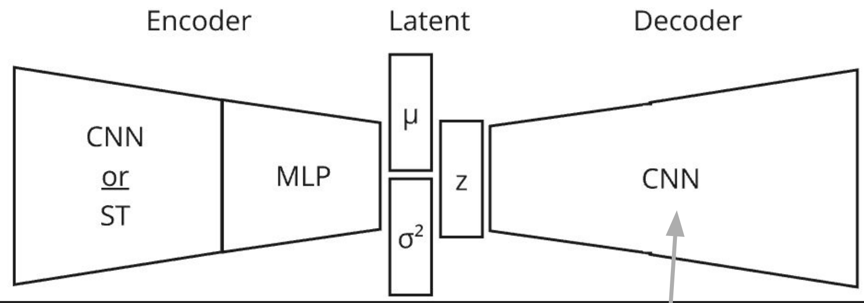
# The encoder learns spatial features, by first extracting shapes, then learning their dependencies




| Layer type | Kernel size | Stride | Padding | Depth | Activation | Regulariser | Pooling | Parameters |
|---|---|---|---|---|---|---|---|---|
| Convolutional | 3x3 | 2 | 1 | 256 | Leaky ReLU | Batch norm. | 2x2 Max pooling | 590,592 |
| Convolutional | 3x3 | 2 | 1 | 384 | Leaky ReLU | Batch norm. | 2x2 Max pooling | 885,888 |
| Convolutional | 2x2 | - | 0 | 384 | - | - | - | 590,208 |
| | | | | | | | | 2,066,688 |

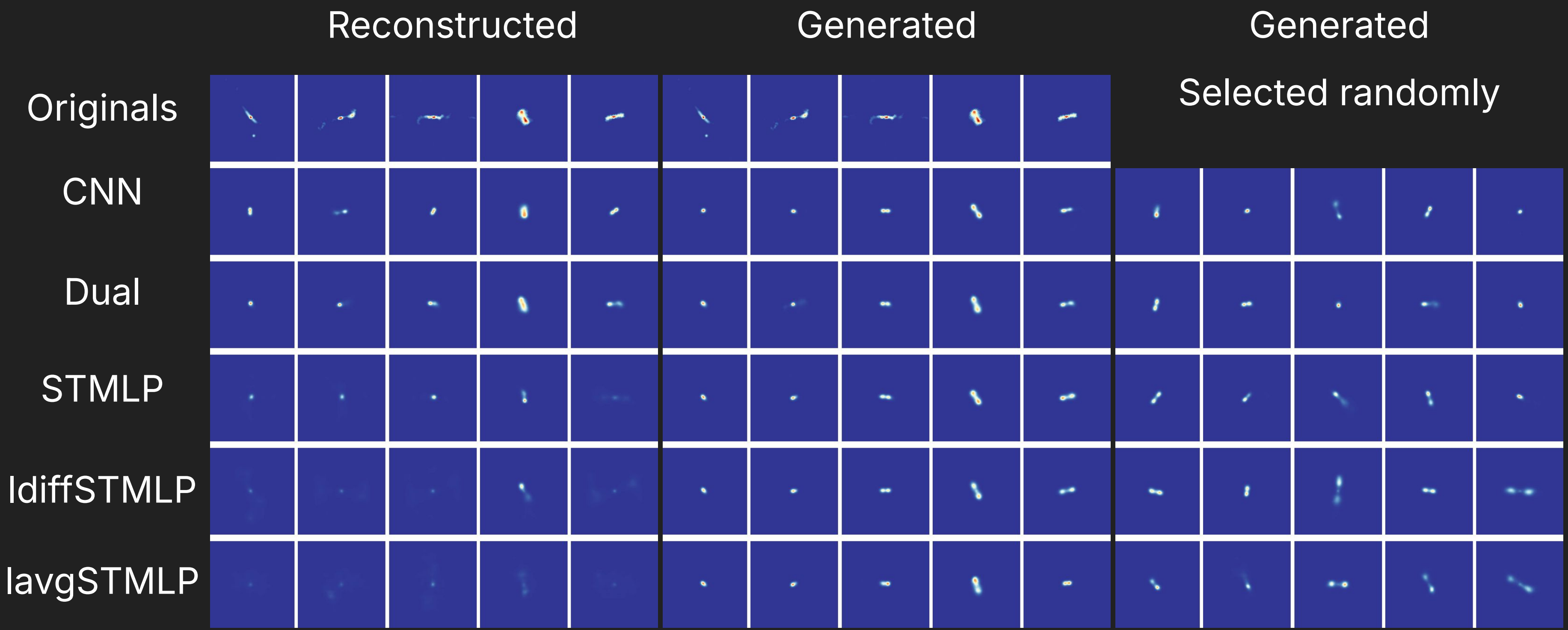
| Layer type | Depth | Activation | Regulariser | Parameters |
|---|---|---|---|---|
| Linear | 384 | Leaky ReLU | Layer norm. | 148,608 |
| Linear | 256 | Leaky ReLU | Layer norm. | 99,072 |
| Linear | 2*32 | - | - | 25,700 |
| | | | | 273,380 |

# The encoder learns spatial features, by first extracting shapes, then learning their dependencies, and then upscaling

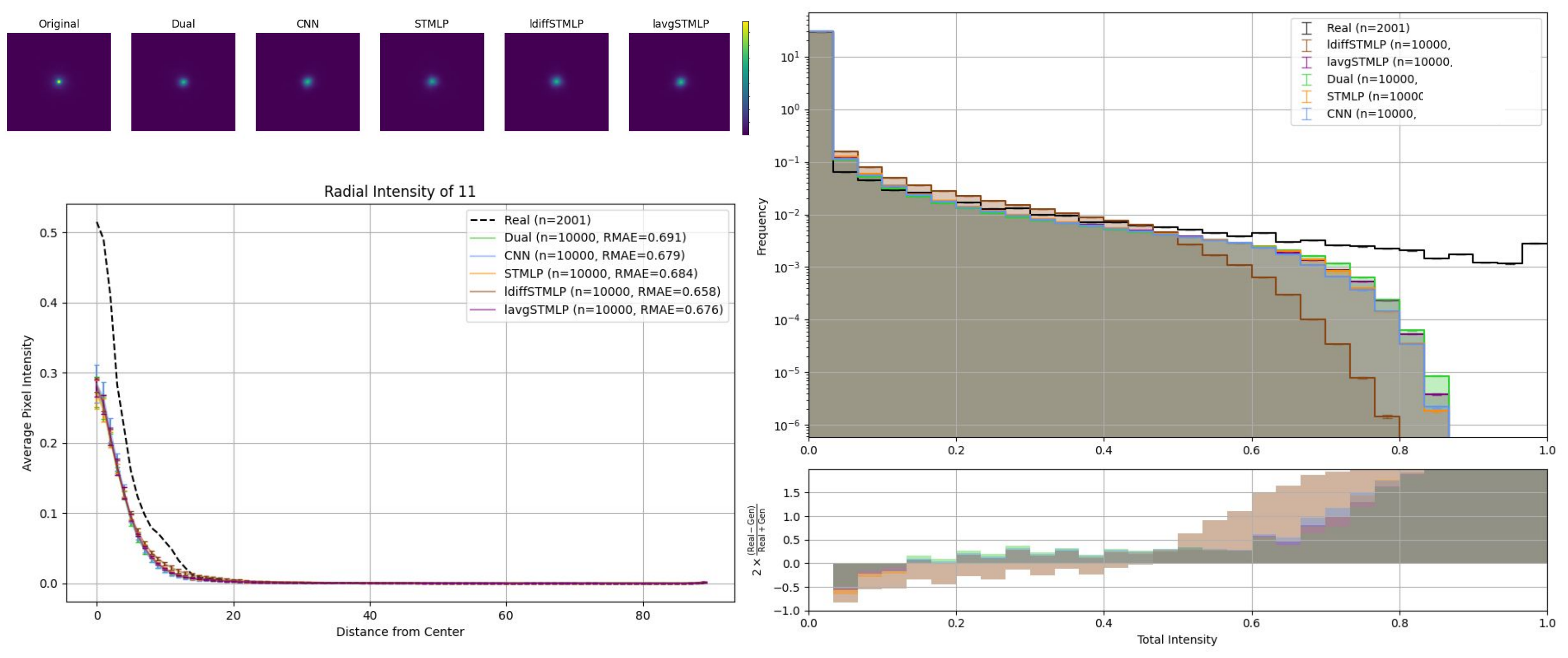



| Layer | Component | Depth | Activation | Regulariser | Parameters |
|---|---|---|---|---|---|
| 1 | Linear | 2048 | Leaky ReLU | - | 67,584 |
| 2-4 | 4x4 Convolutional transpose | 64 | Leaky ReLU | Batch norm. | 205,376 |
| 5-7 | 4x4 Convolutional transpose | 128 | Leaky ReLU | Batch norm. | 427,136 |
| 8-10 | 4x4 Convolutional transpose | 64 | Leaky ReLU | Batch norm. | 205,376 |
| 11-13 | 4x4 Convolutional transpose | 32 | Leaky ReLU | Batch norm. | 51,488 |
| 14 | 4x4 Convolutional transpose | 1 | Sigmoid | - | 513 |
| | | | | | 957,473 |

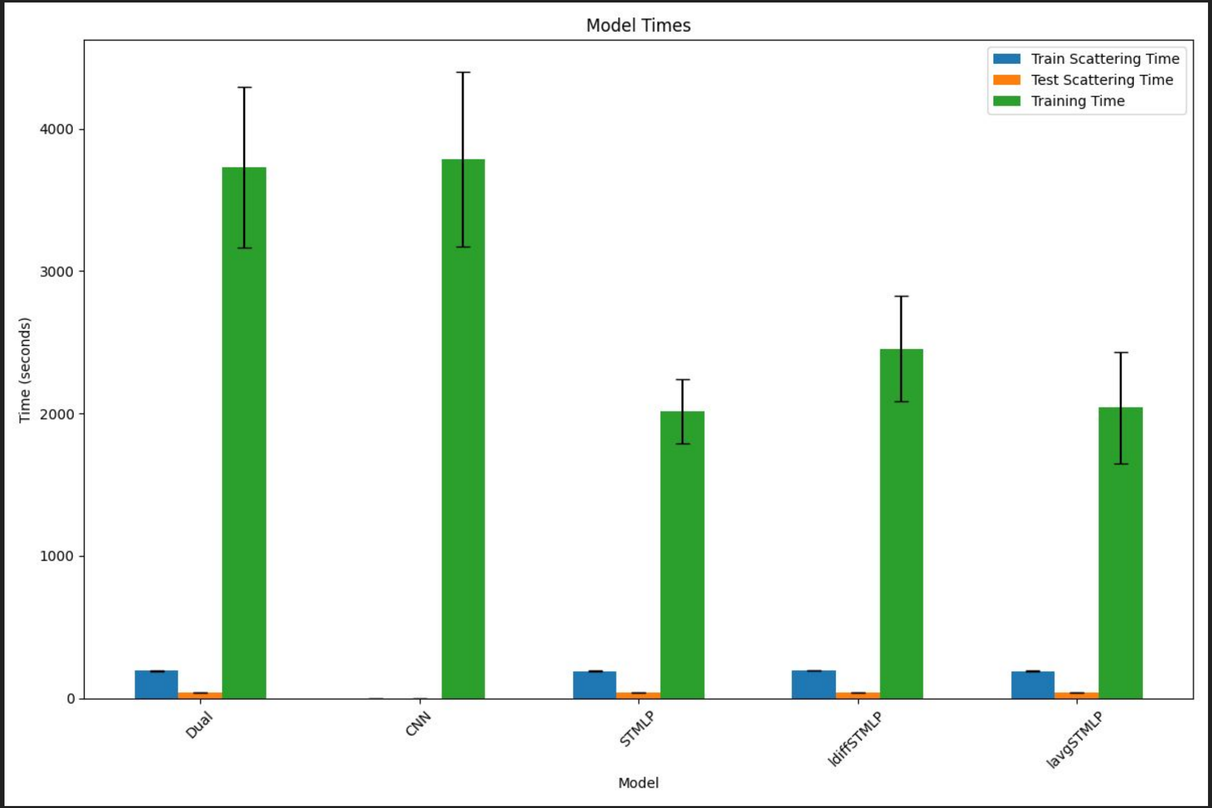
# Model reproduces smoother sources than the originals



# Model reproduces fainter sources than the originals



# But the scattering transform is faster



# Conclusions and outlook

## Conclusions

1. The scattering transform (ST) is computationally efficient and interpretable
2. Scattering-based generative modelling of radio galaxies require multivariate learning
3. My variational autoencoders produces overly smooth and faith images

## Future plans

1. More complex generative models
   a. Generative Adversarial Networks (GANs)
   b. Diffusion models
   c. Normalising flows
2. Apply on diffuse cluster radio emission

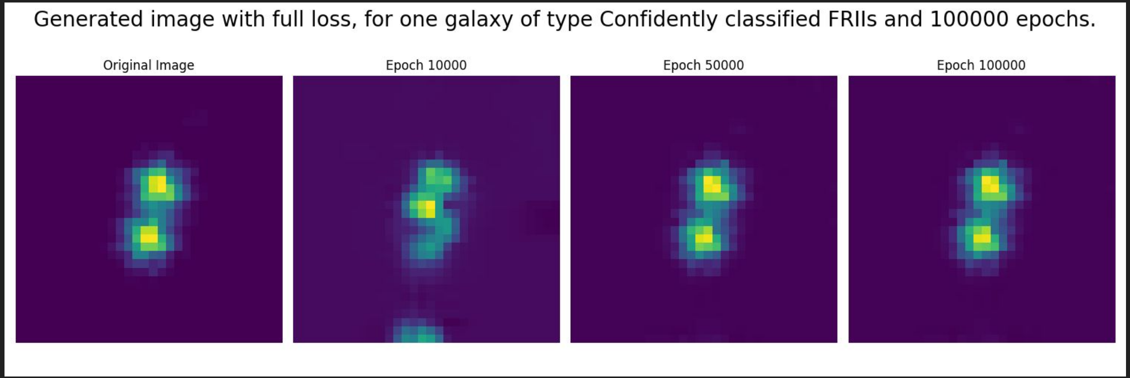
**Markus Bredberg**
markus.bredberg@epfl.ch
EPFL

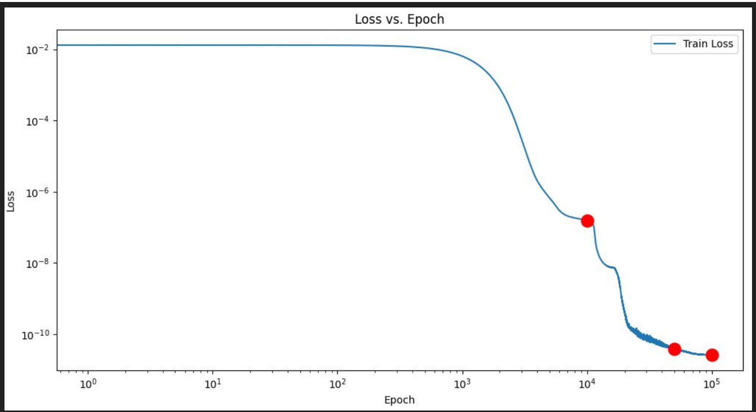
# Bonus slides

Clarifications

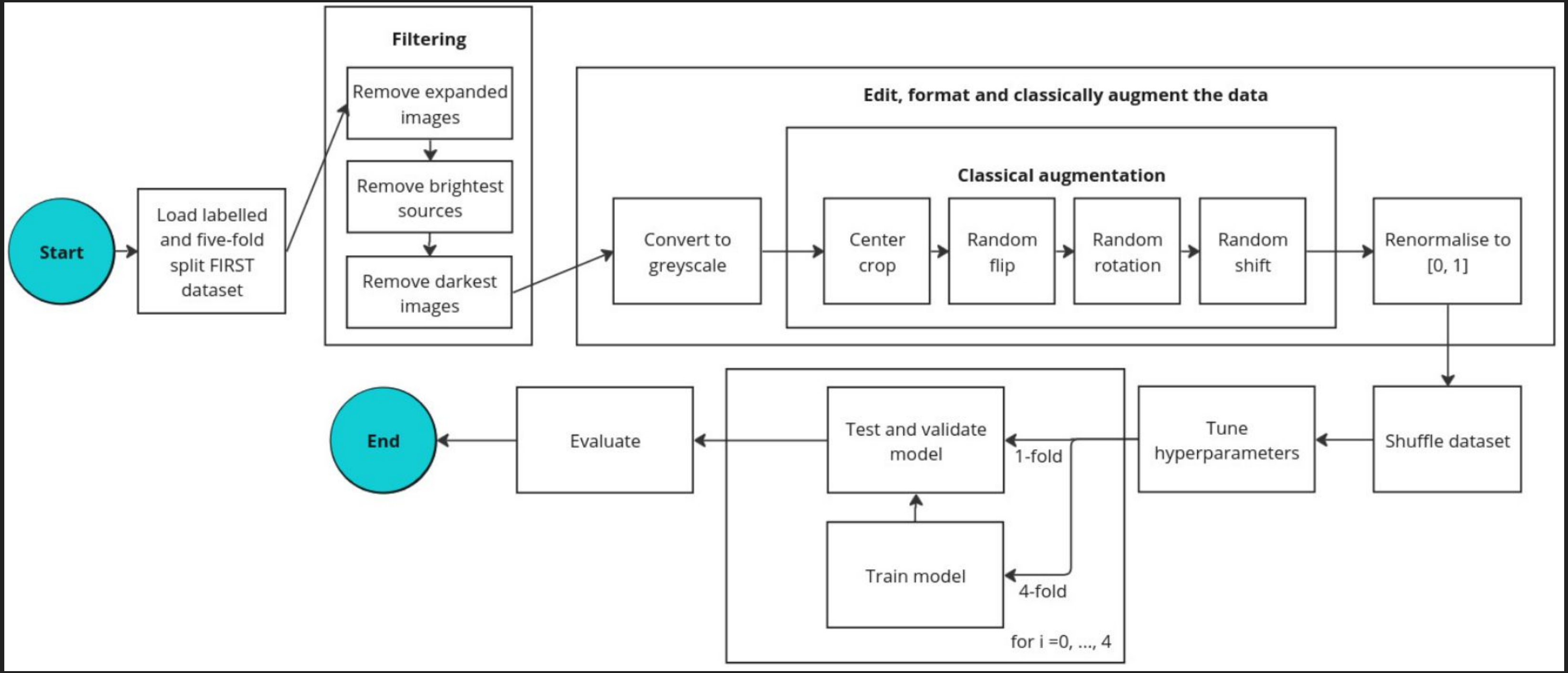
Classification for labelling and evaluation

# Shallow generation in epochs

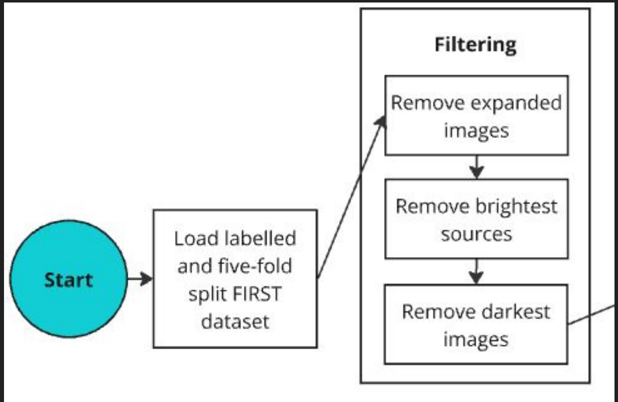


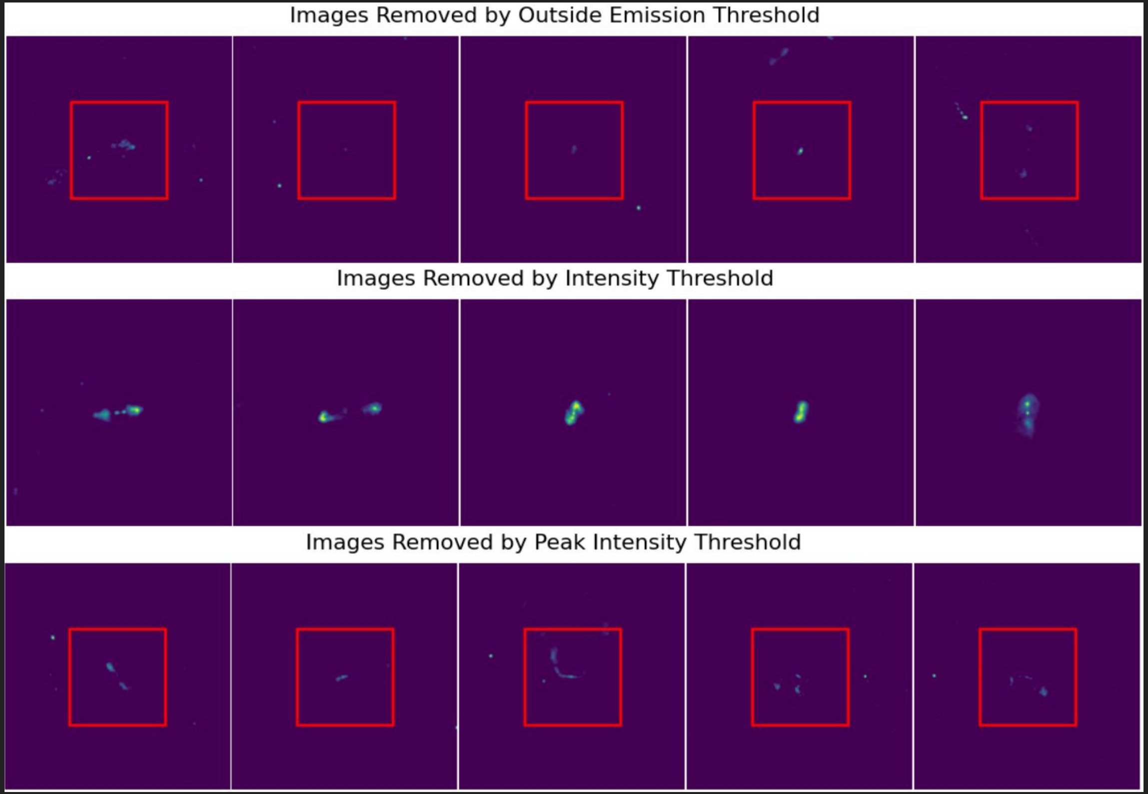



# Running the generative script: the algorithm schematically

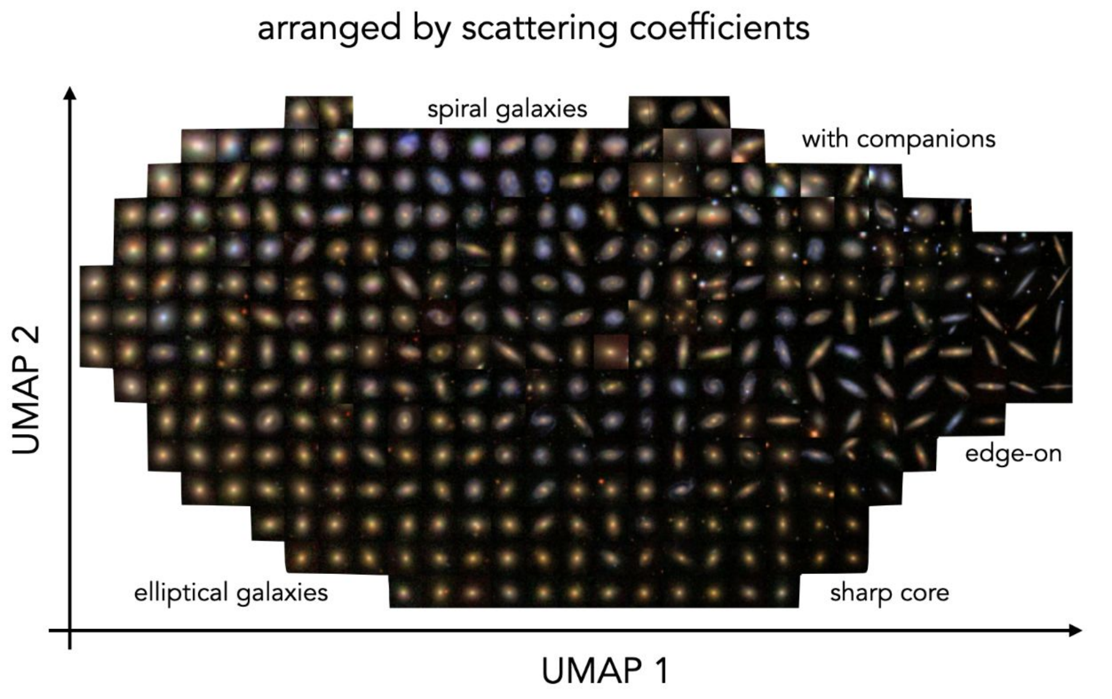


# Running the generative script: the filtering



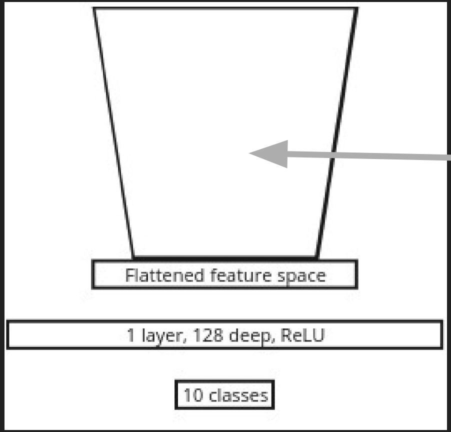



# The scattering transform can categorise




Cheng & Ménard, 2021, arXiv:2112.01288v1

# Classifying with the scattering transform




**CNN**

| Layer type | Depth | Kernel size | Activation | Pooling type | Pooling size |
|---|---|---|---|---|---|
| Conv2D | 32 | 5x5 | ReLU | MaxPool2D | 2x2 |
| Conv2D | 64 | 5x5 | ReLU | MaxPool2D | 2x2 |

or

**Scattering Transform**

$$S_{3,8}(X) = \Psi_{3,8}(X) \star \phi_3$$

# Scattering transform classification speeds up

## CNN: 99.33%

True label / Predicted label

| | 0 | | | | | | | | | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 99.80% | 0.00% | 0.00% | 0.10% | 0.00% | 0.00% | 0.00% | 0.00% | 0.10% | 0.00% |
| | 0.00% | 99.82% | 0.00% | 0.00% | 0.00% | 0.00% | 0.09% | 0.00% | 0.09% | 0.00% |
| | 0.19% | 0.10% | 99.22% | 0.00% | 0.00% | 0.00% | 0.00% | 0.39% | 0.10% | 0.00% |
| | 0.00% | 0.00% | 0.10% | 99.60% | 0.00% | 0.20% | 0.00% | 0.00% | 0.10% | 0.00% |
| | 0.00% | 0.00% | 0.00% | 0.00% | 99.49% | 0.00% | 0.10% | 0.00% | 0.00% | 0.41% |
| | 0.11% | 0.00% | 0.00% | 0.56% | 0.00% | 99.10% | 0.11% | 0.00% | 0.00% | 0.11% |
| | 0.10% | 0.21% | 0.00% | 0.00% | 0.10% | 0.31% | 99.06% | 0.00% | 0.21% | 0.00% |
| | 0.00% | 0.10% | 0.58% | 0.00% | 0.00% | 0.00% | 0.00% | 99.12% | 0.10% | 0.10% |
| | 0.21% | 0.00% | 0.21% | 0.10% | 0.00% | 0.00% | 0.00% | 0.10% | 99.18% | 0.21% |
| 9 | 0.00% | 0.00% | 0.10% | 0.00% | 0.40% | 0.40% | 0.00% | 0.30% | 0.00% | 98.81% |

## Scattering Transform Network: 99:25%

True label / Predicted label

| | 0 | | | | | | | | | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 99.39% | 0.00% | 0.10% | 0.00% | 0.00% | 0.00% | 0.20% | 0.10% | 0.20% | 0.00% |
| | 0.00% | 99.12% | 0.18% | 0.09% | 0.09% | 0.00% | 0.18% | 0.09% | 0.26% | 0.00% |
| | 0.10% | 0.00% | 99.42% | 0.10% | 0.00% | 0.00% | 0.00% | 0.29% | 0.10% | 0.00% |
| | 0.00% | 0.00% | 0.10% | 99.60% | 0.00% | 0.10% | 0.00% | 0.00% | 0.20% | 0.00% |
| | 0.00% | 0.00% | 0.00% | 0.00% | 99.49% | 0.00% | 0.10% | 0.00% | 0.20% | 0.20% |
| | 0.22% | 0.00% | 0.00% | 0.90% | 0.00% | 98.43% | 0.11% | 0.11% | 0.11% | 0.11% |
| | 0.21% | 0.00% | 0.10% | 0.00% | 0.10% | 0.10% | 99.16% | 0.00% | 0.31% | 0.00% |
| | 0.00% | 0.10% | 0.39% | 0.10% | 0.00% | 0.00% | 0.00% | 98.93% | 0.19% | 0.29% |
| | 0.00% | 0.00% | 0.10% | 0.00% | 0.00% | 0.10% | 0.00% | 0.00% | 99.69% | 0.10% |
| 9 | 0.00% | 0.00% | 0.10% | 0.00% | 0.30% | 0.00% | 0.10% | 0.10% | 0.20% | 99.21% |

Markus Bredberg, Jiaxin Guo, Han Zhang (2024)

# The VAEs do not learn the feature representation