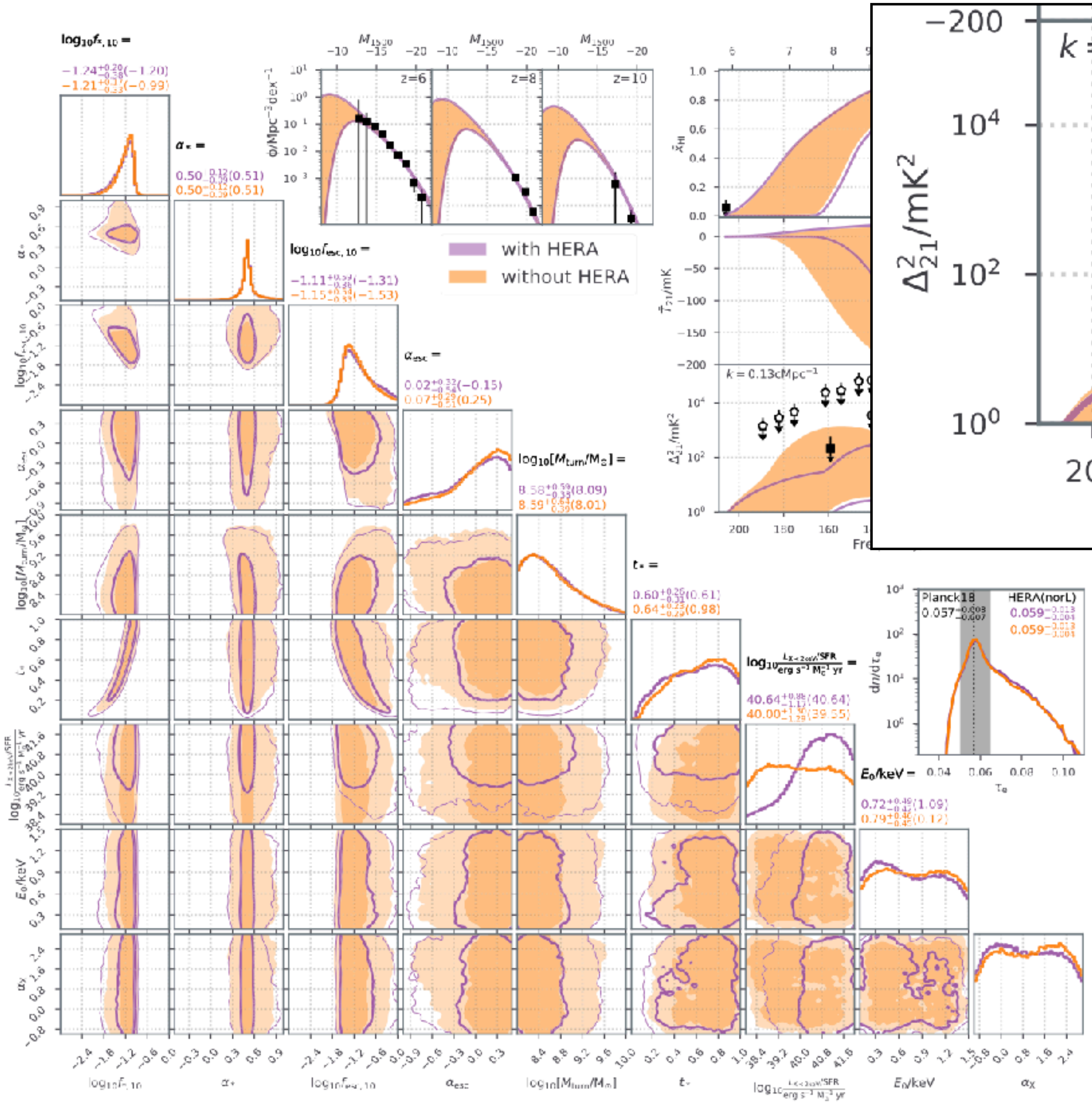
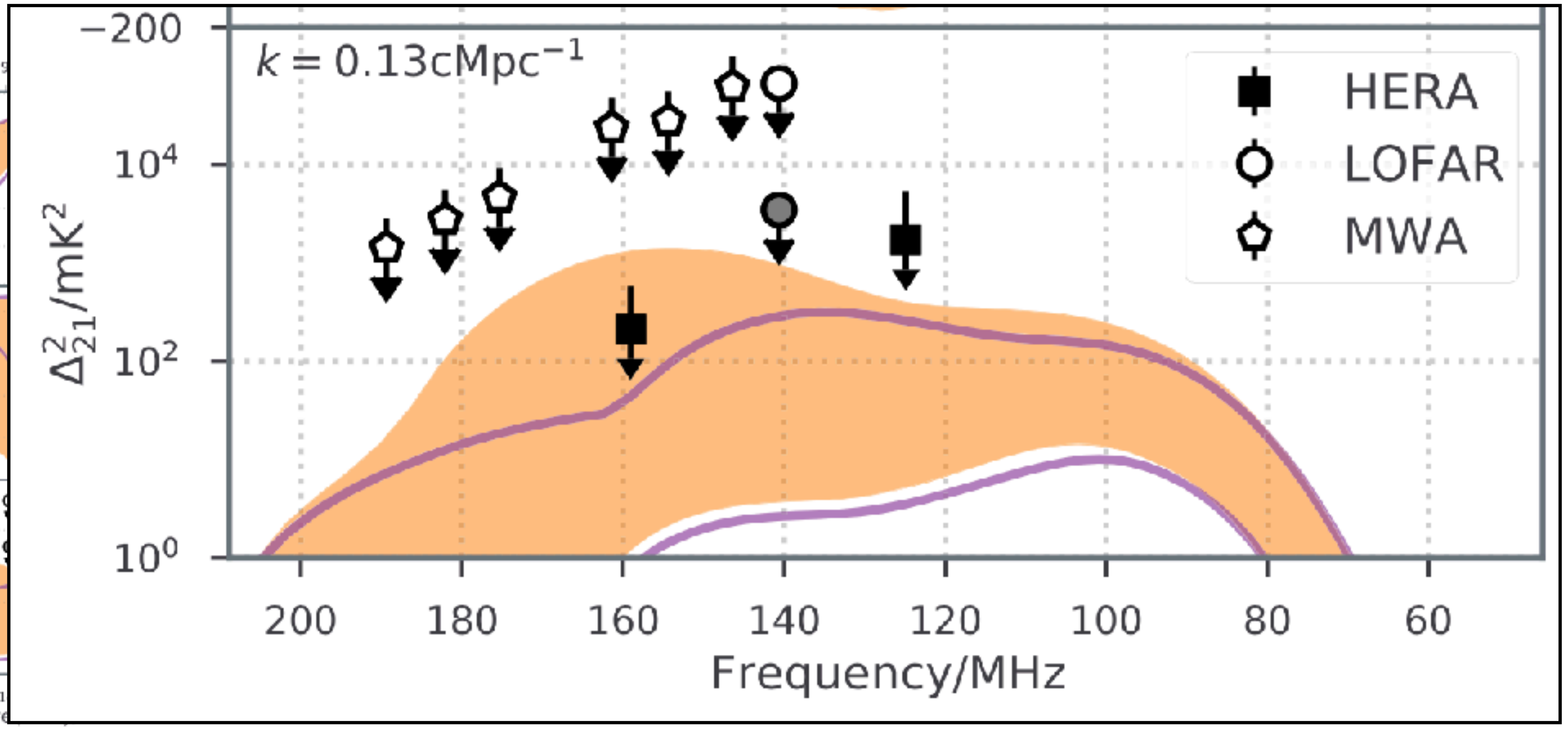
Estimating EoR parameters with deep learning

Kana Moriwaki (The University of Tokyo)

Hongo 21cm workshop
3-4 Oct. 2024

Estimating EoR parameters

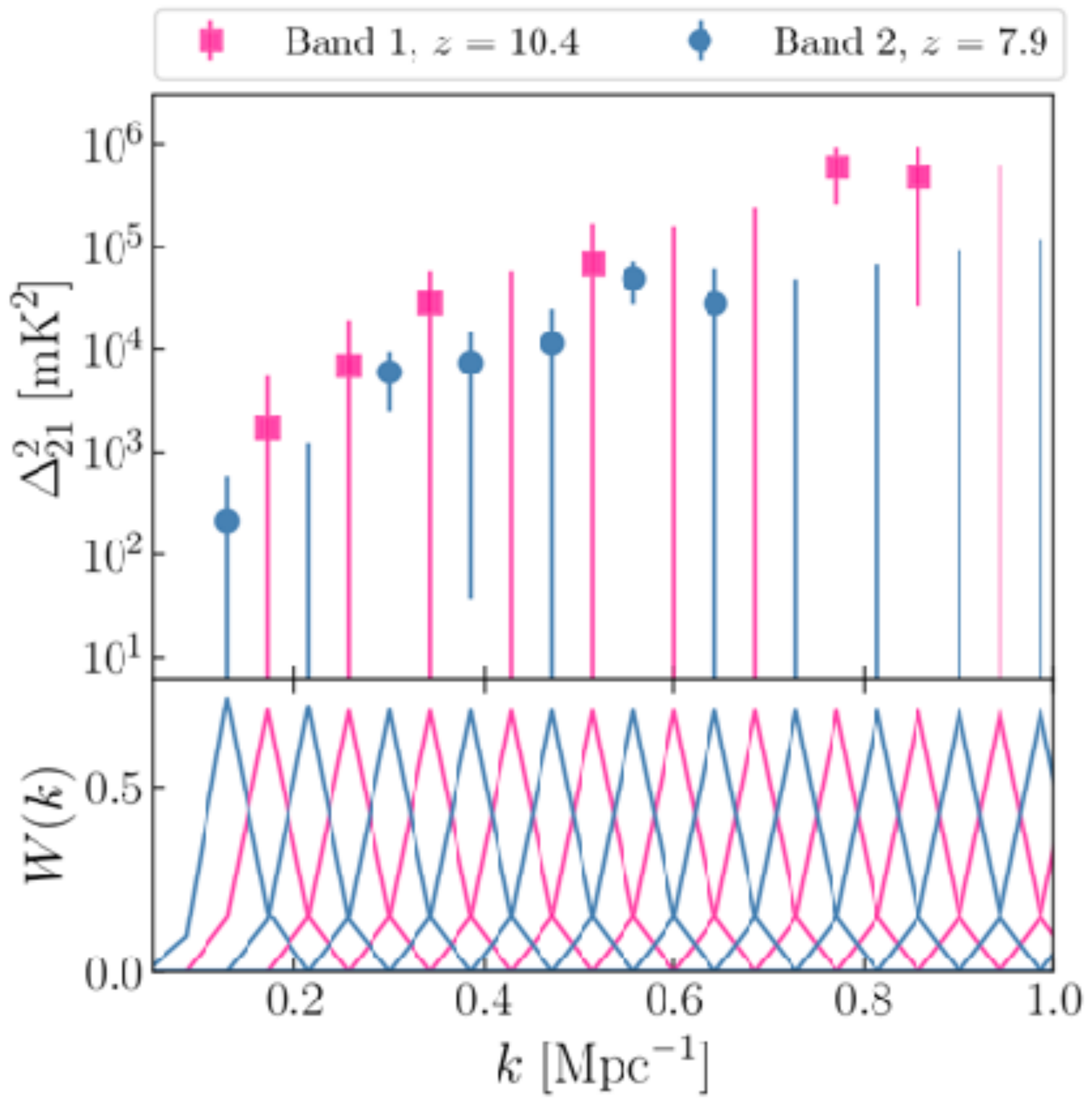




Abdurashidova+2022 (HERA collaboration)

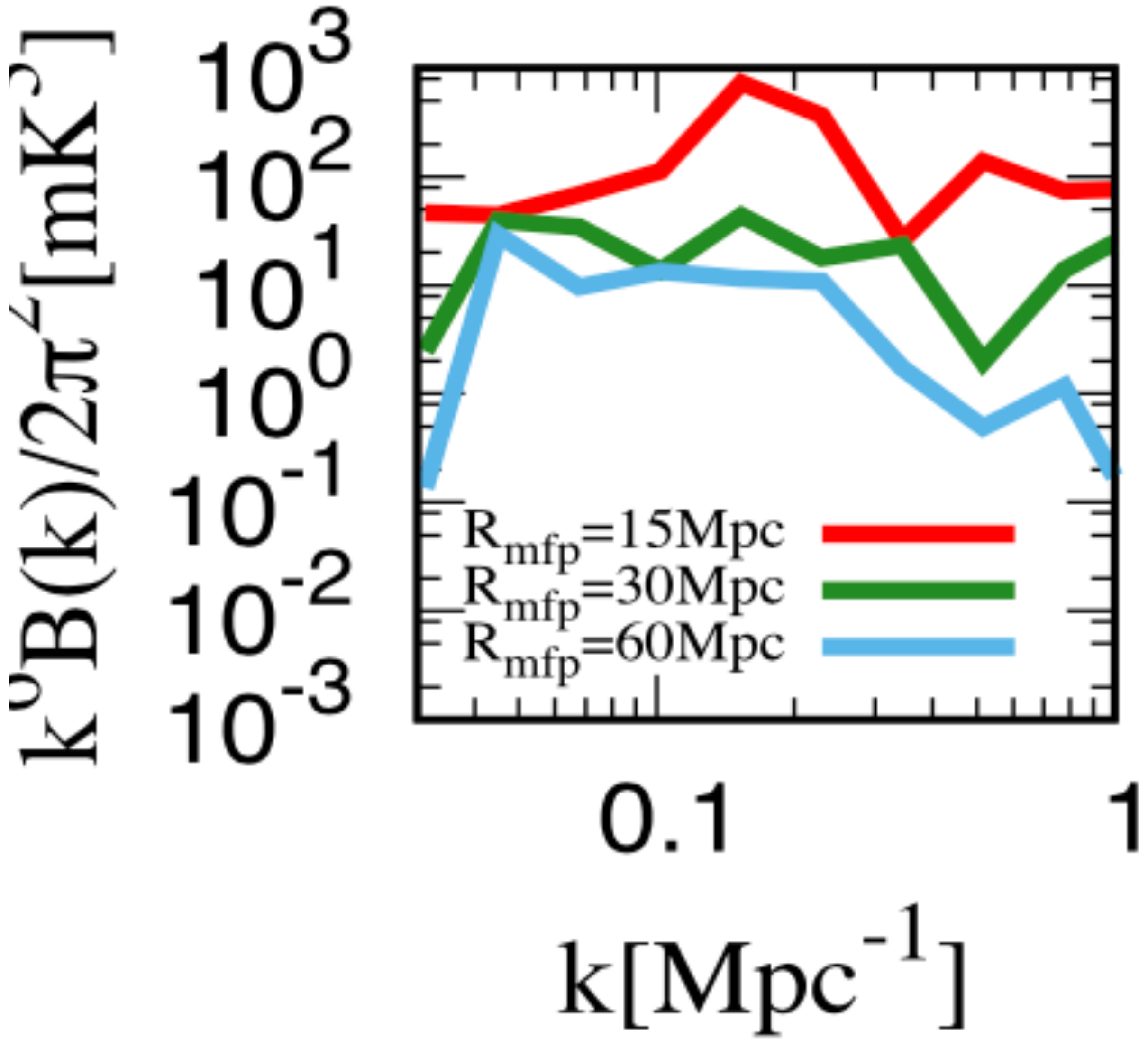
What summary statistics to use?

Power spectrum



HERA 2022

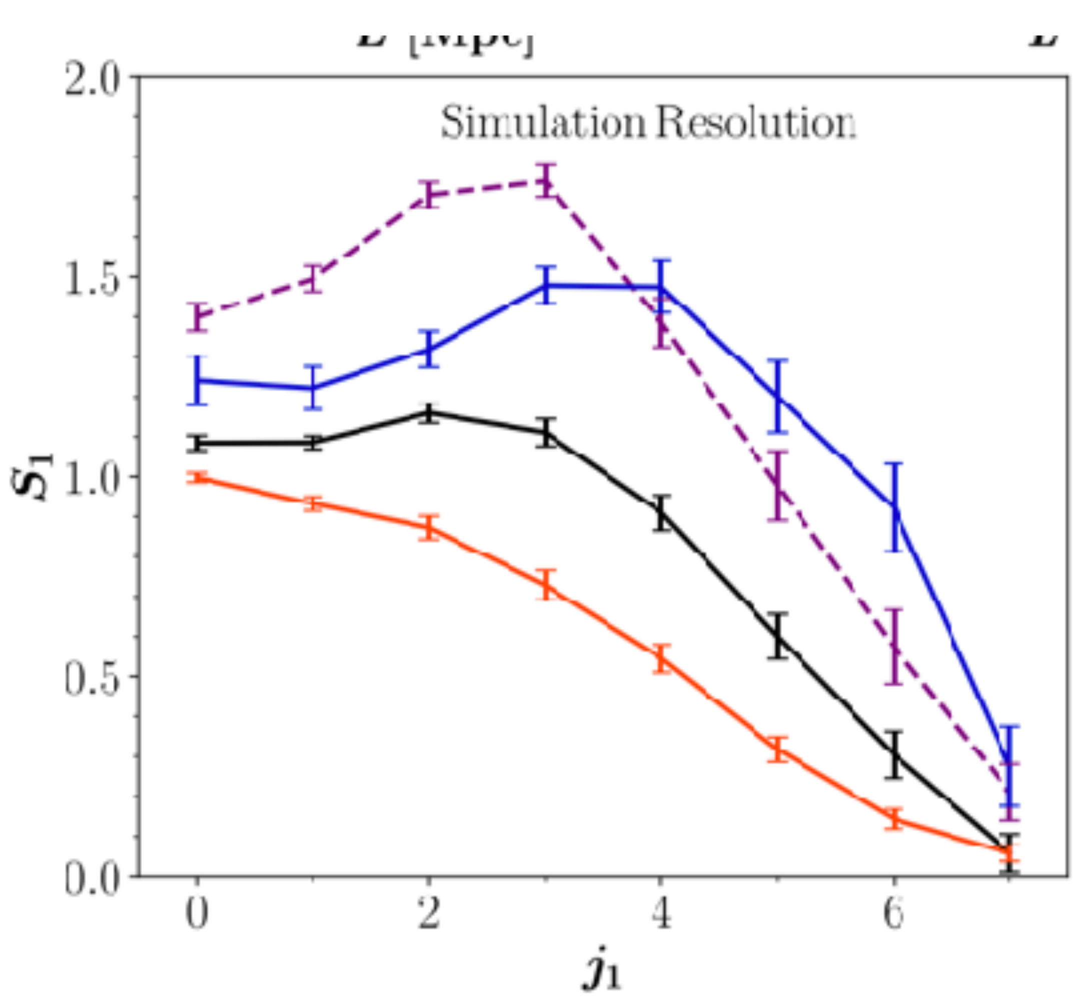
Bispectrum



Shimabukuro+2017

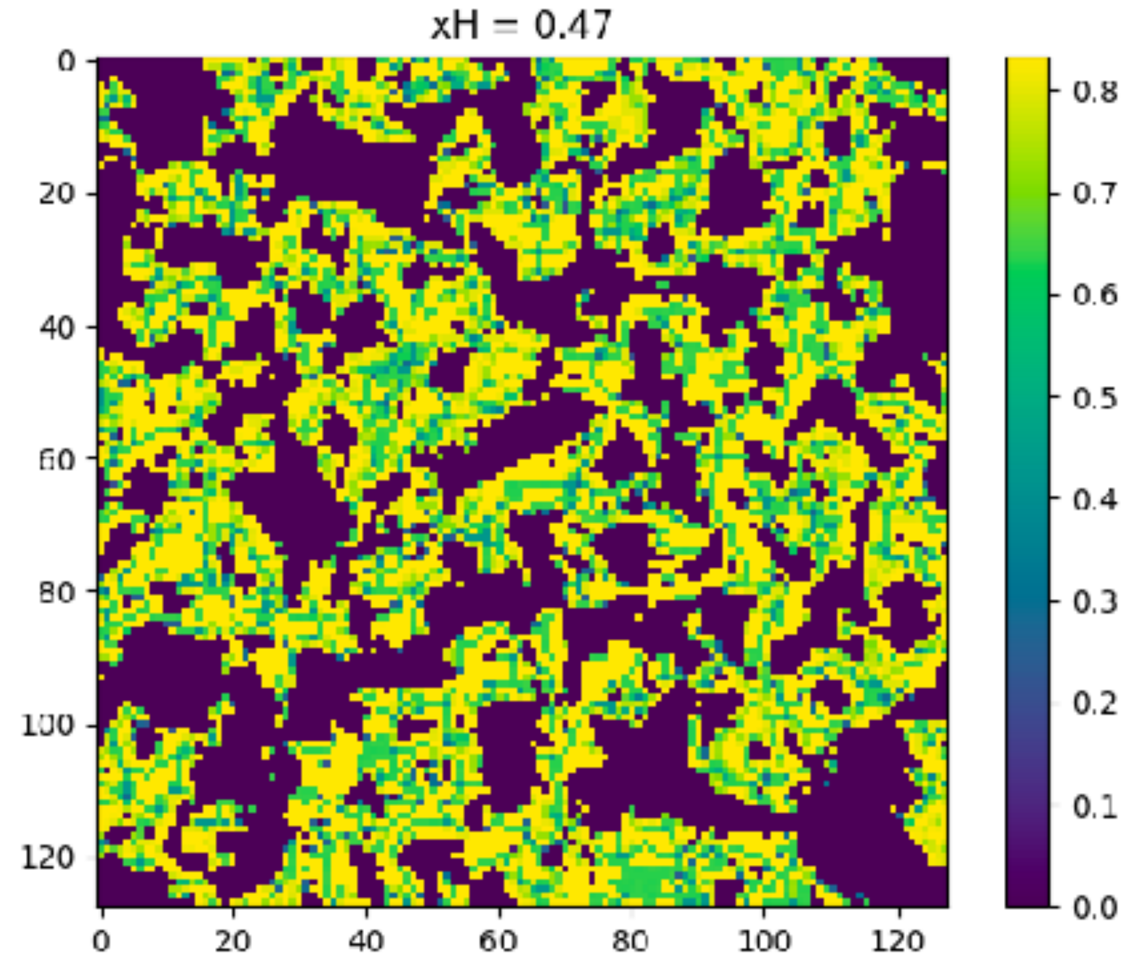
WST

wavelet scattering transform



Greig+2023

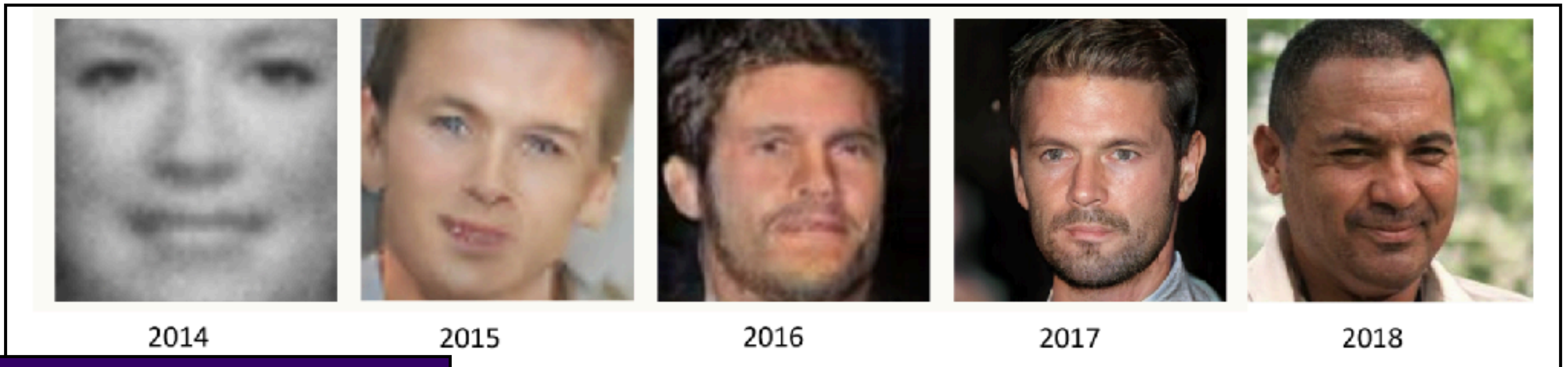
Image?



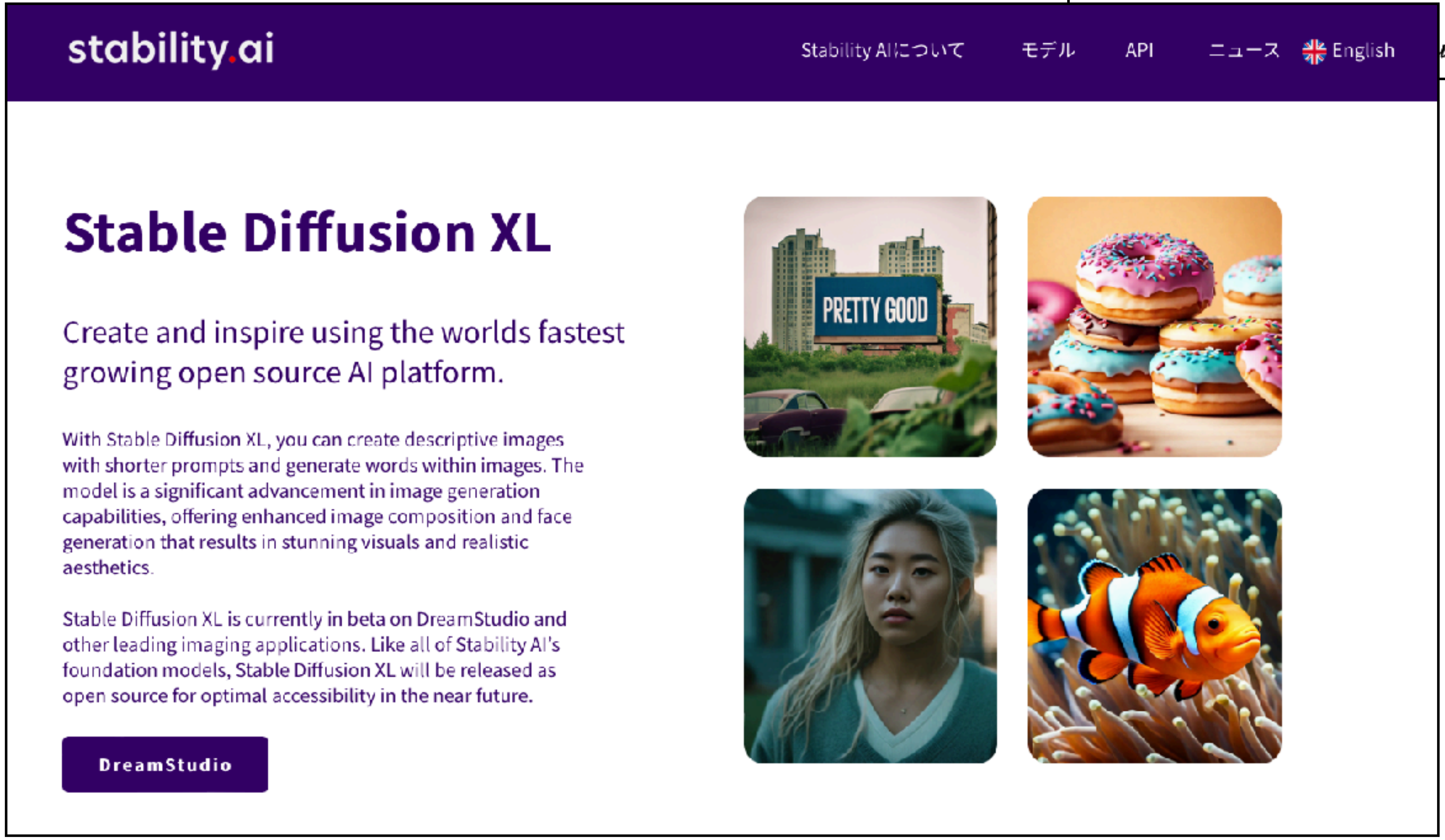
Can we directly deal with the image cubes?

Imaging is expected be possible at the late stage of SKA1 or in SKA2

Generative models

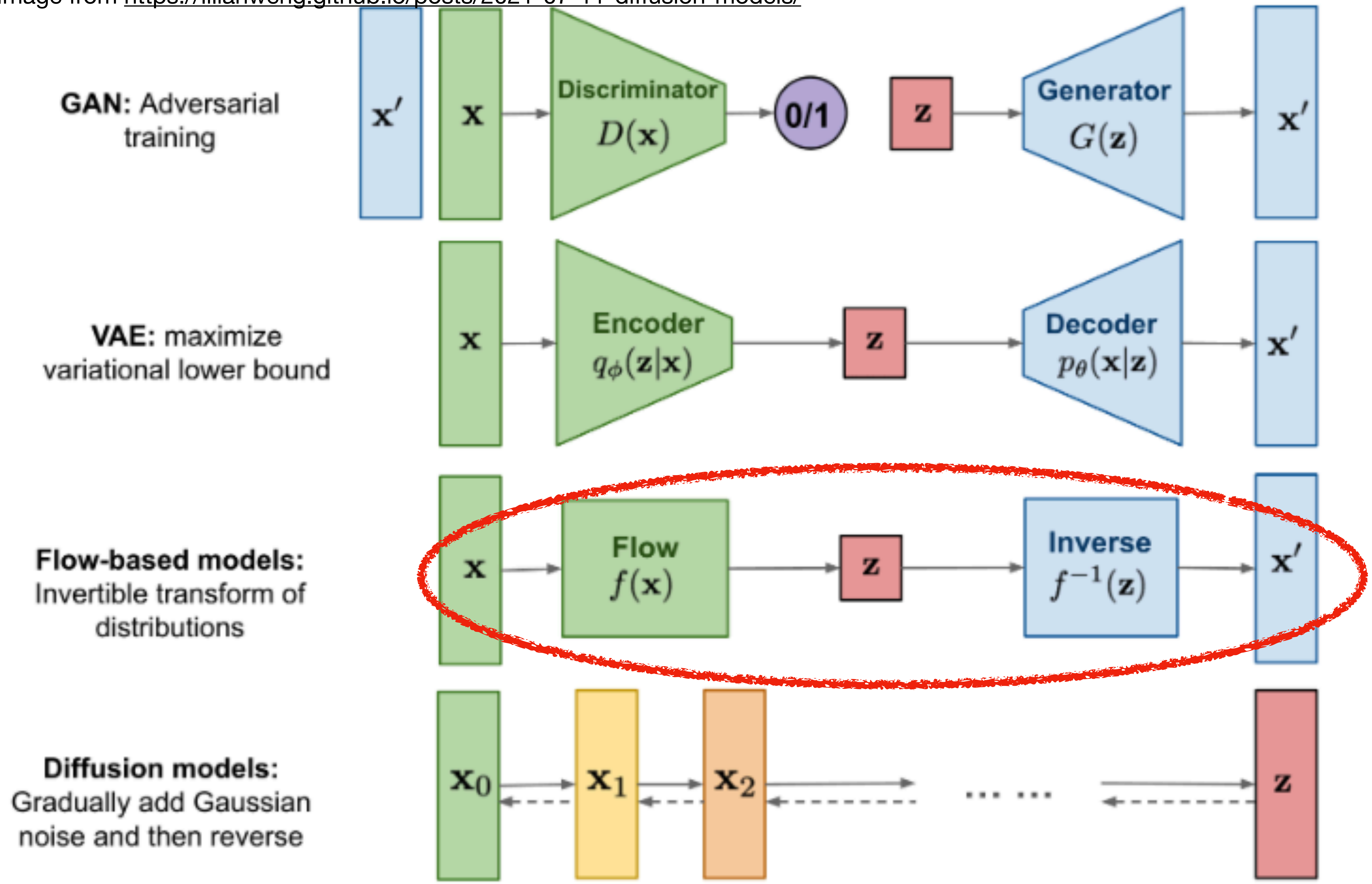


...sing generative modeling has improved significantly in the last four years[4]



There are several different generative models

Image from https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

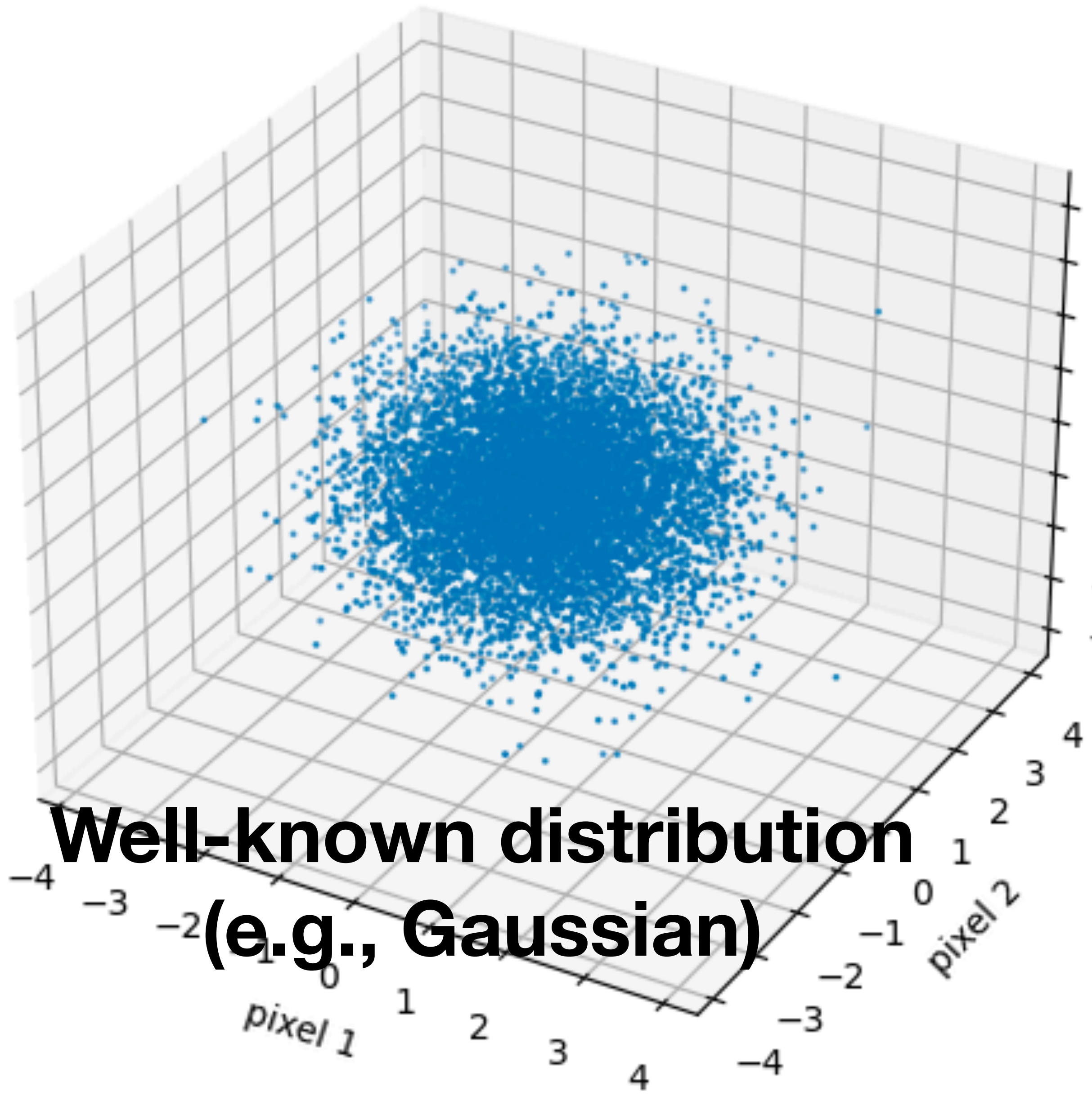


Might not be the SOTA in terms of generation quality… But NF tells us about the probability density

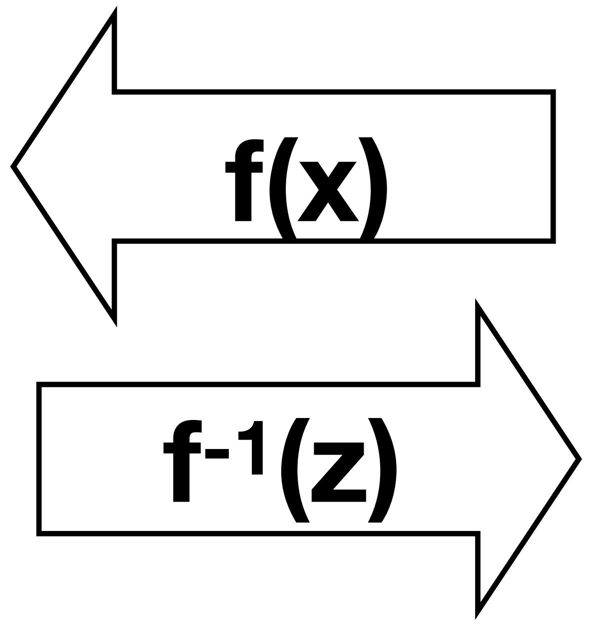
Data space ⇔ Latent space

Flow-based models transfer data into latent space with invertible functions.

latent space

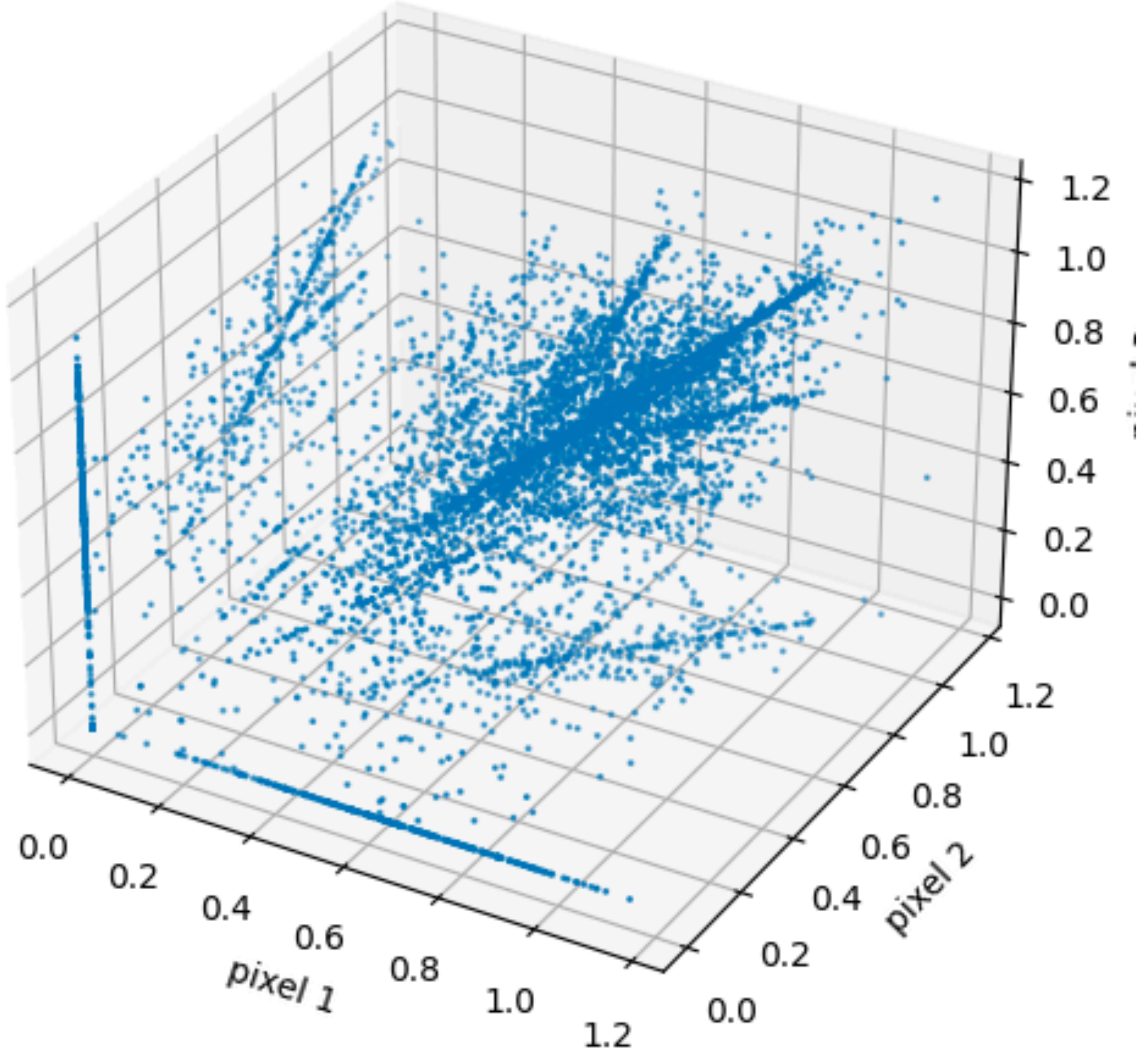


Well-known distribution (e.g., Gaussian)

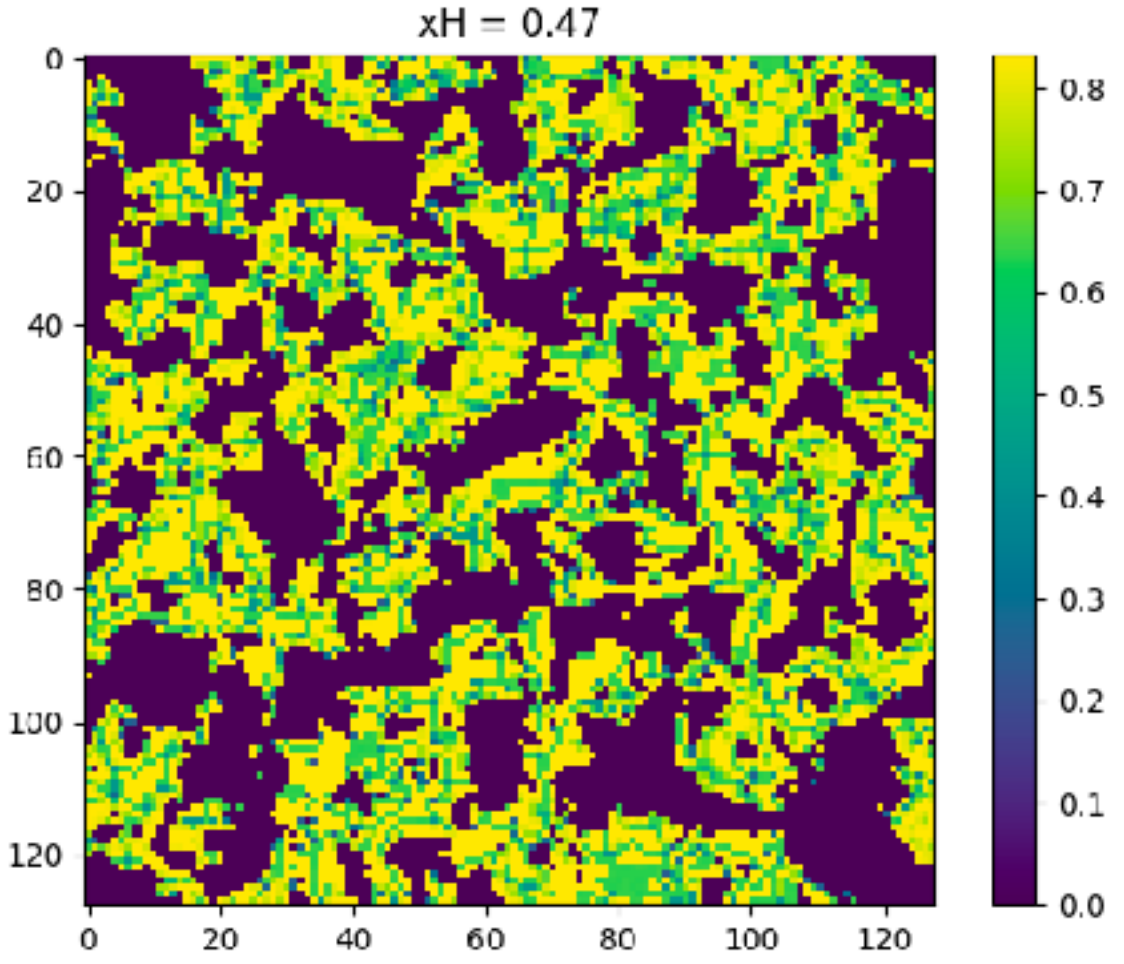


f: invertible

data space: 256×256 dimension



← distribution of pixel values of 21cm map



- **When generating a new data, one can sample a random point in the latent space and apply "f^{-1}"**
- **Invertible functions can be used to infer the probability density of a given data**

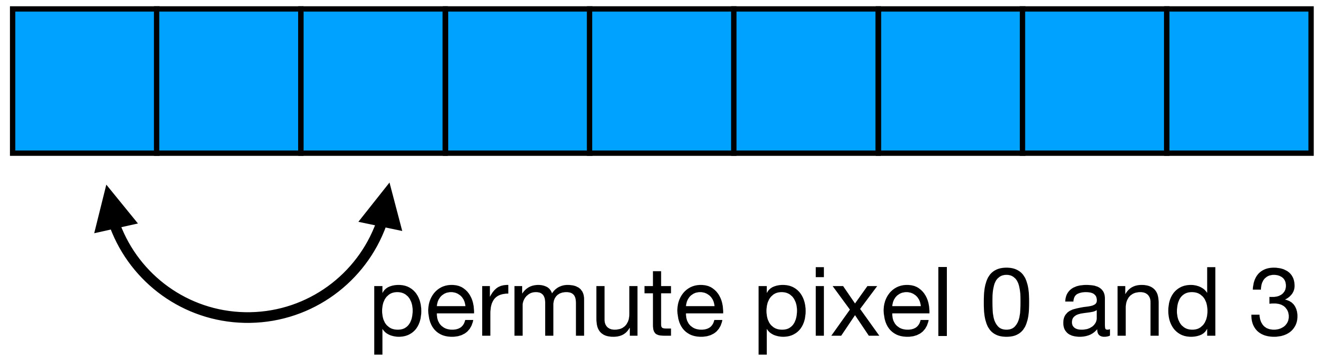
What transformation to use?

Transformations are invertible and their determinants can be (easily) computed

Examples:

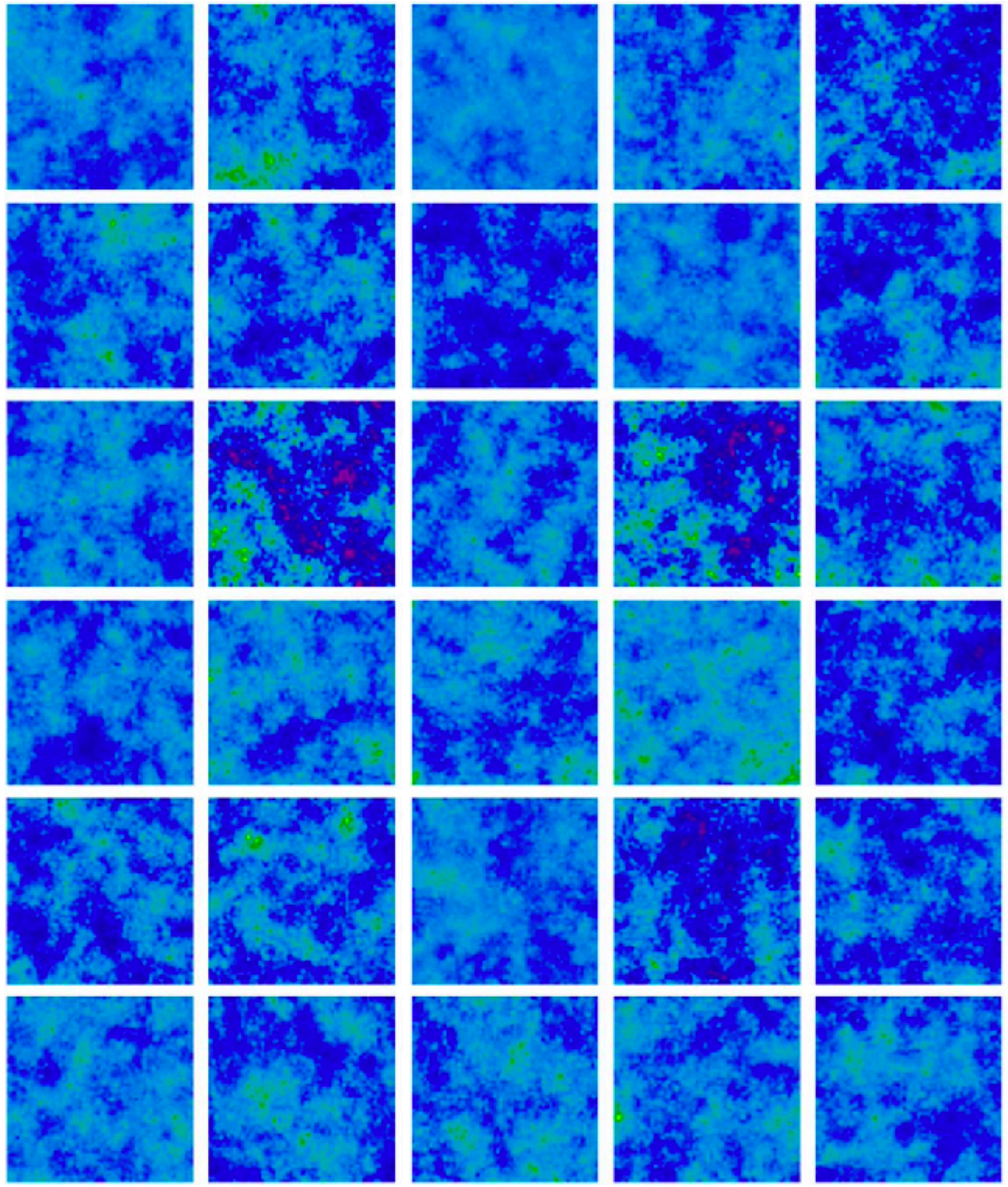
$$f(x) = \begin{pmatrix} 3 & 2 \\ 0 & 4 \end{pmatrix} x \longrightarrow f^{-1}(x) = \begin{pmatrix} 1/3 & -1/6 \\ 0 & 1/4 \end{pmatrix} x, \quad \det\left(\frac{\partial f(x)}{\partial x}\right) = 3 \times 4$$

$f:$ permutation $\longrightarrow$ f^{-1} : inverse permutation, $\det\left(\frac{\partial f(x)}{\partial x}\right) = 1$

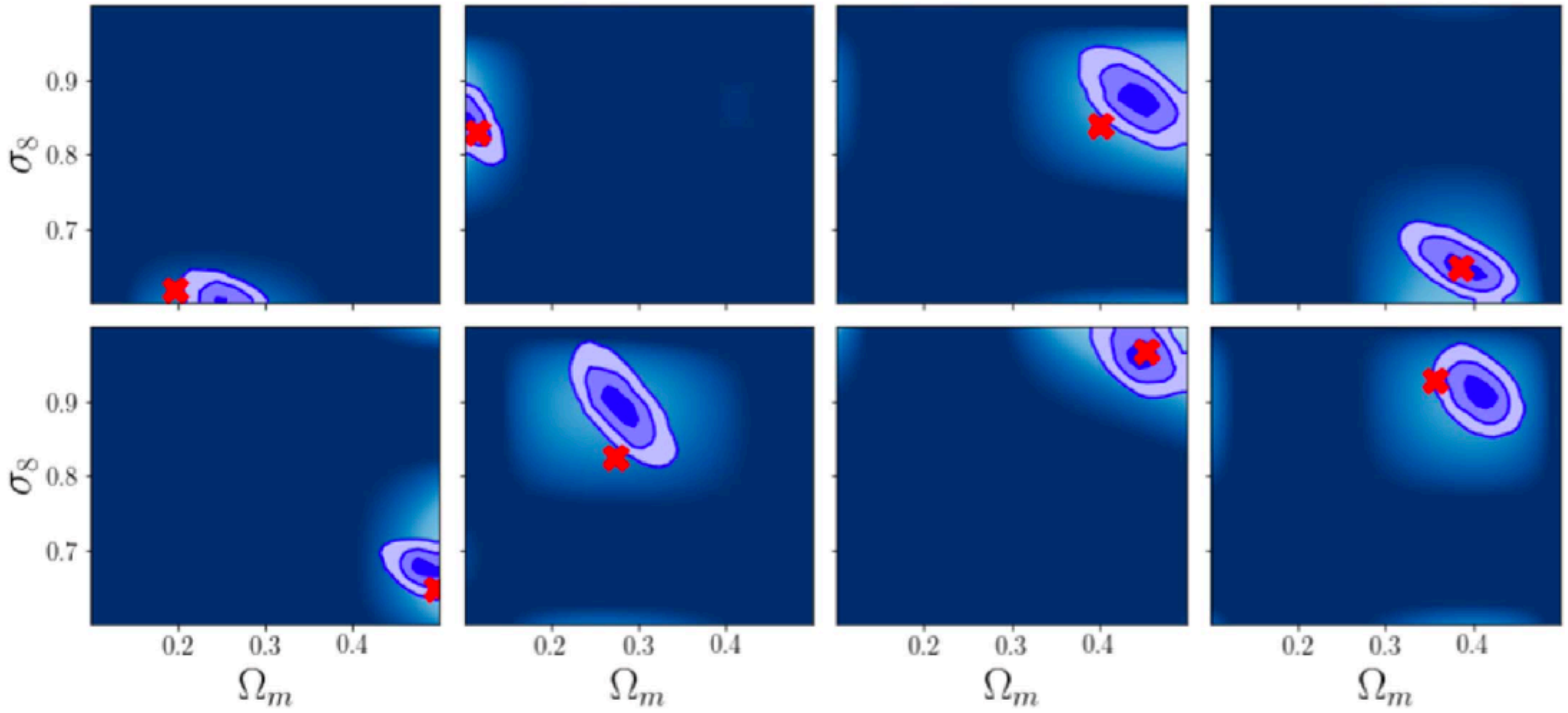


Hassan+ 2022: application of NF to post-reionization 21cm maps

Generated LSS images

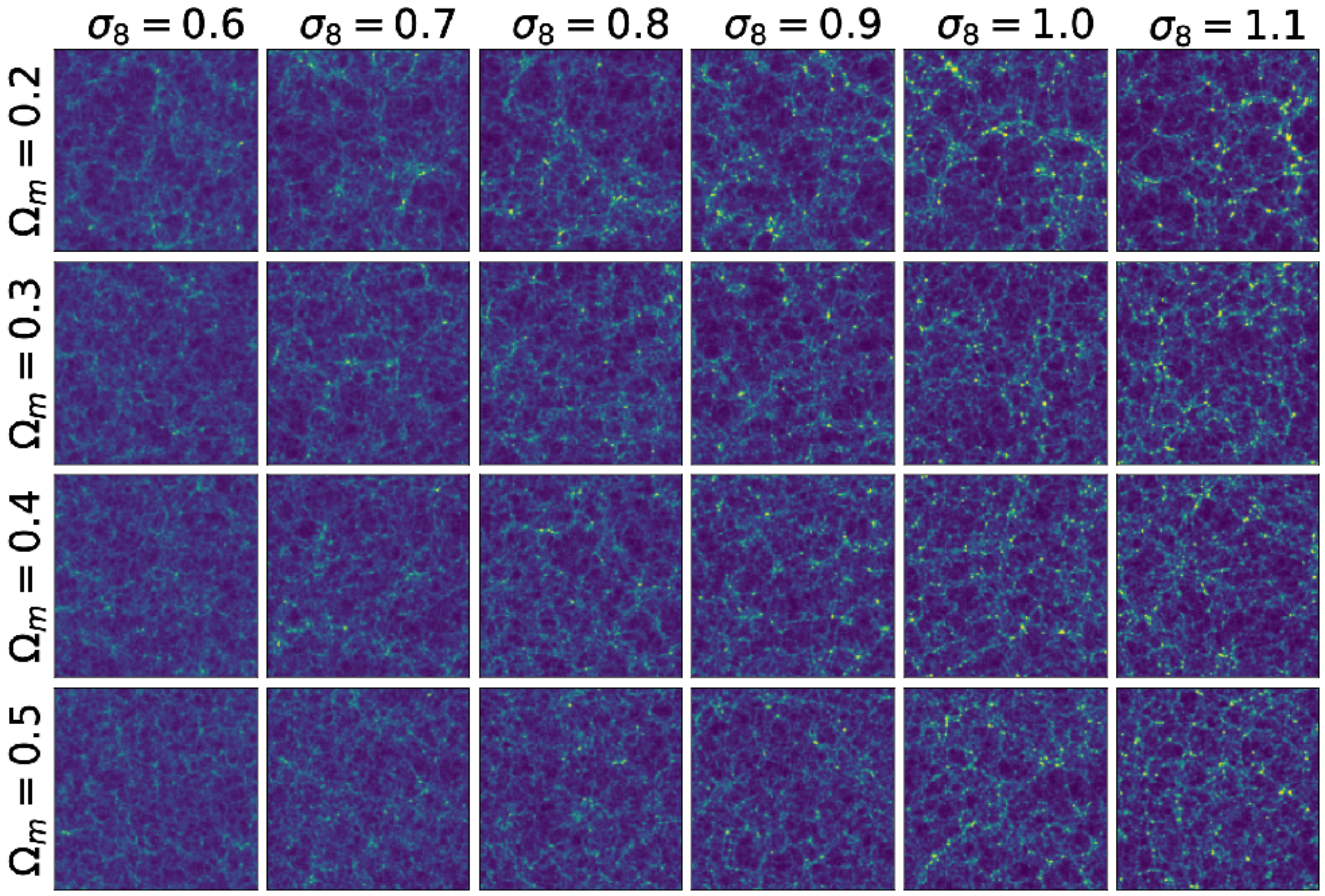


Cosmological parameter inference

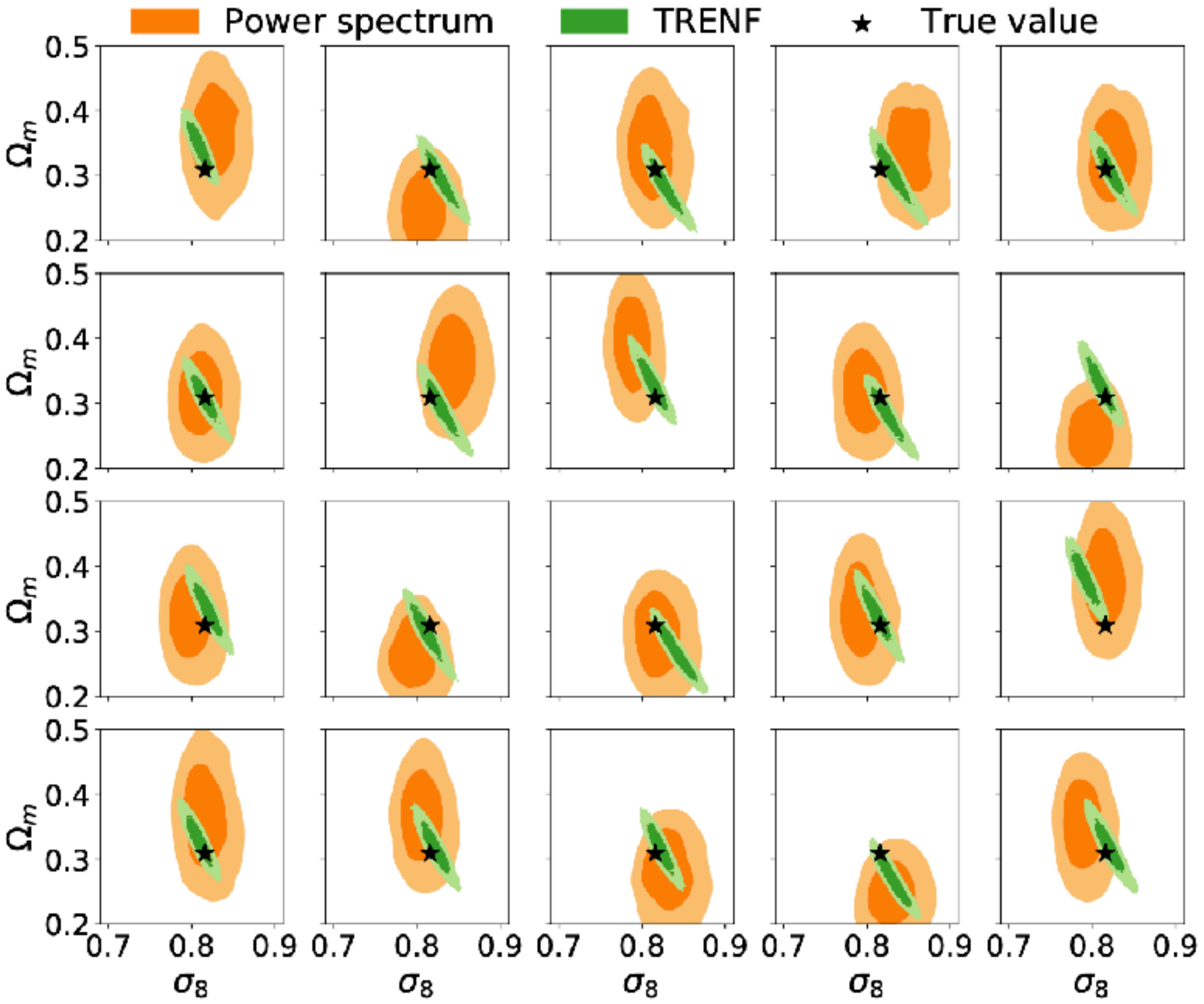


Dai+2022: Translation and Rotation Equivariant Normalizing Flow

Generated LSS images



Cosmological parameter inference



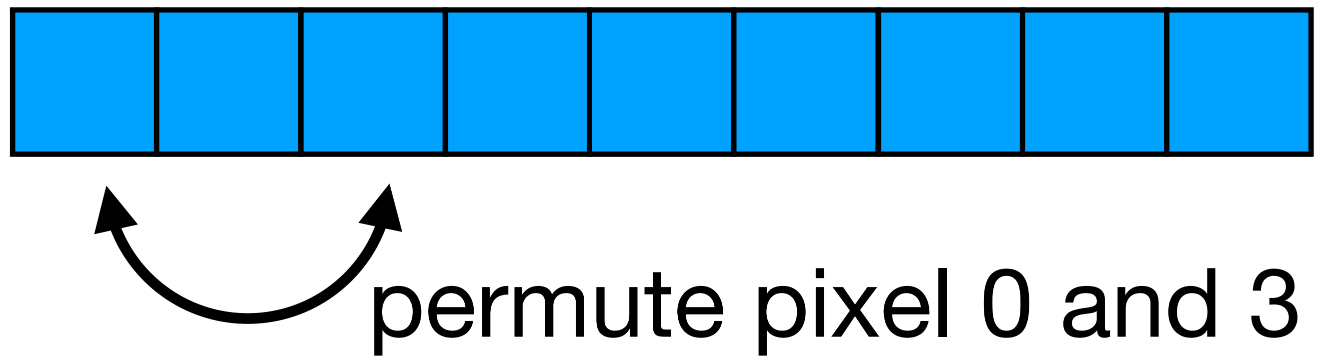
What transformation to use?

Transformations are invertible and their determinants can be (easily) computed

Examples:

$$f(x) = \begin{pmatrix} 3 & 2 \\ 0 & 4 \end{pmatrix} x \longrightarrow f^{-1}(x) = \begin{pmatrix} 1/3 & -1/6 \\ 0 & 1/4 \end{pmatrix} x, \quad \det\left(\frac{\partial f(x)}{\partial x}\right) = 3 \times 4$$

$f:$ permutation $\longrightarrow$ f^{-1} : inverse permutation, $\det\left(\frac{\partial f(x)}{\partial x}\right) = 1$



Translation and Rotation Equivariant Normalizing Flow (TRENF; Dai+2022)

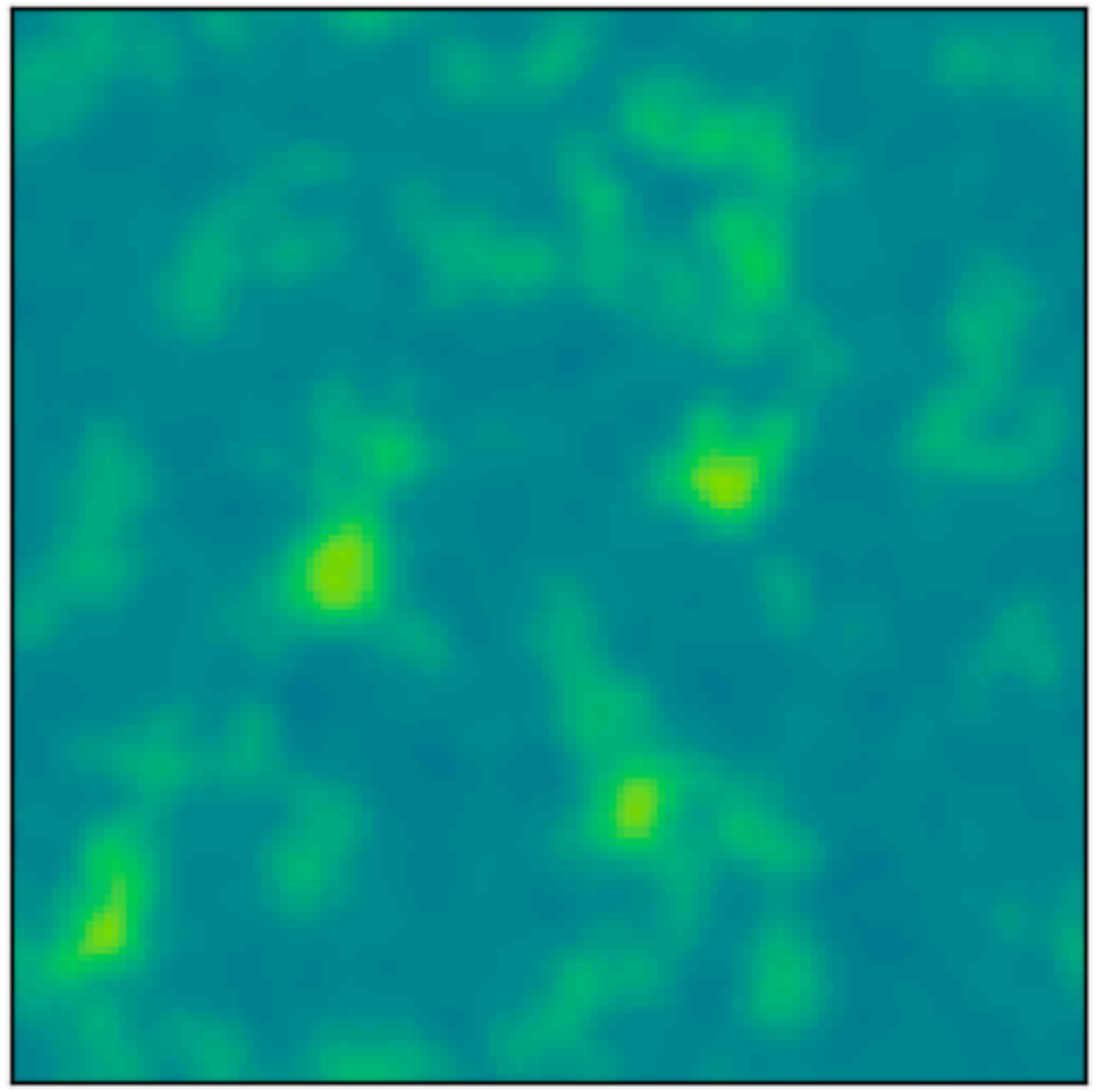
$$f = \Psi\left(\hat{F}^{-1}\tilde{T}(k)\hat{F}x\right)$$

Ψ **: Monotonic nonlinear function**

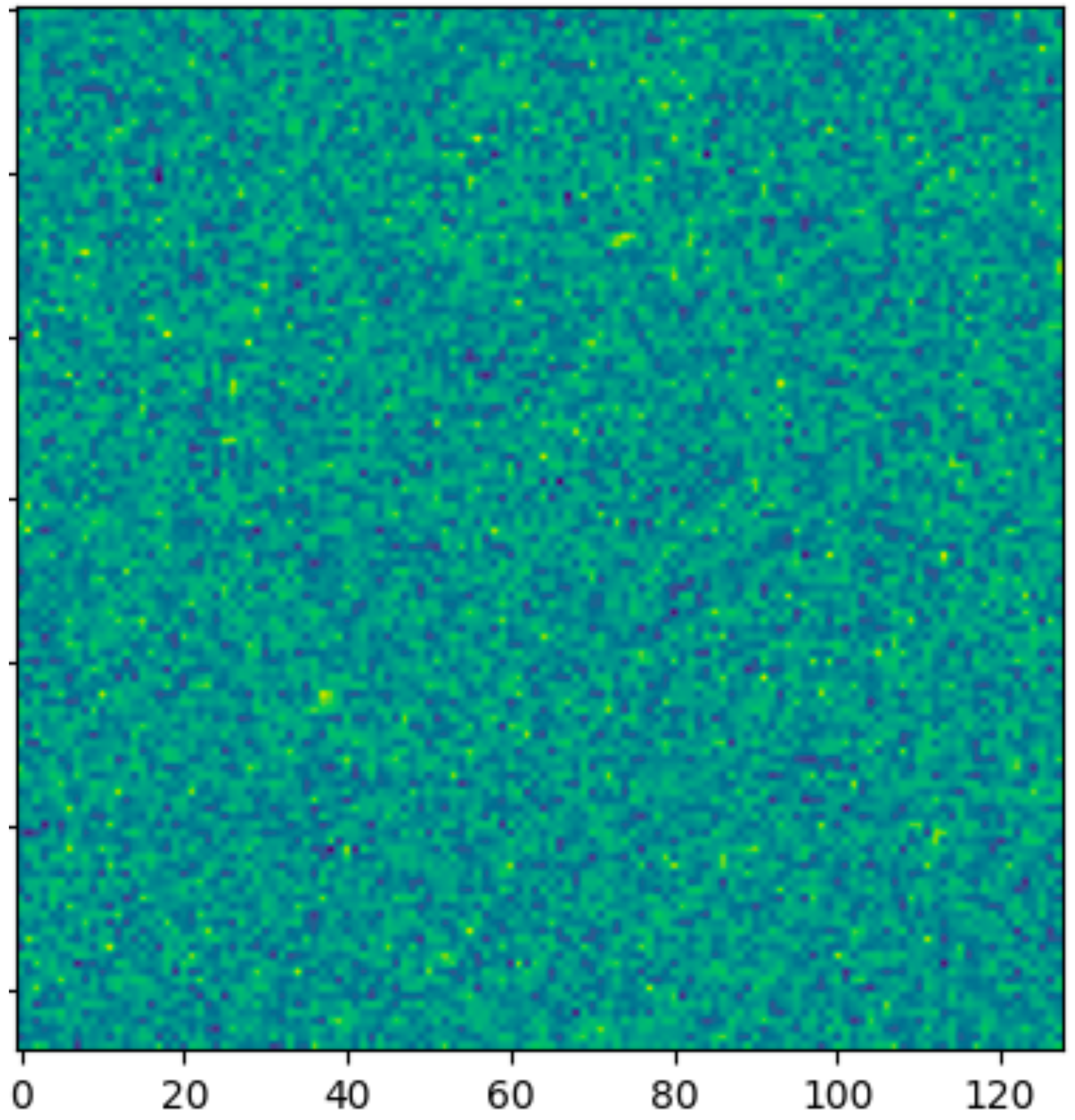
$\tilde{T}(k)$**: non-zero function**

Let us use NF for EoR 21cm map!

21cm map + noise + smoothing



Latent variable



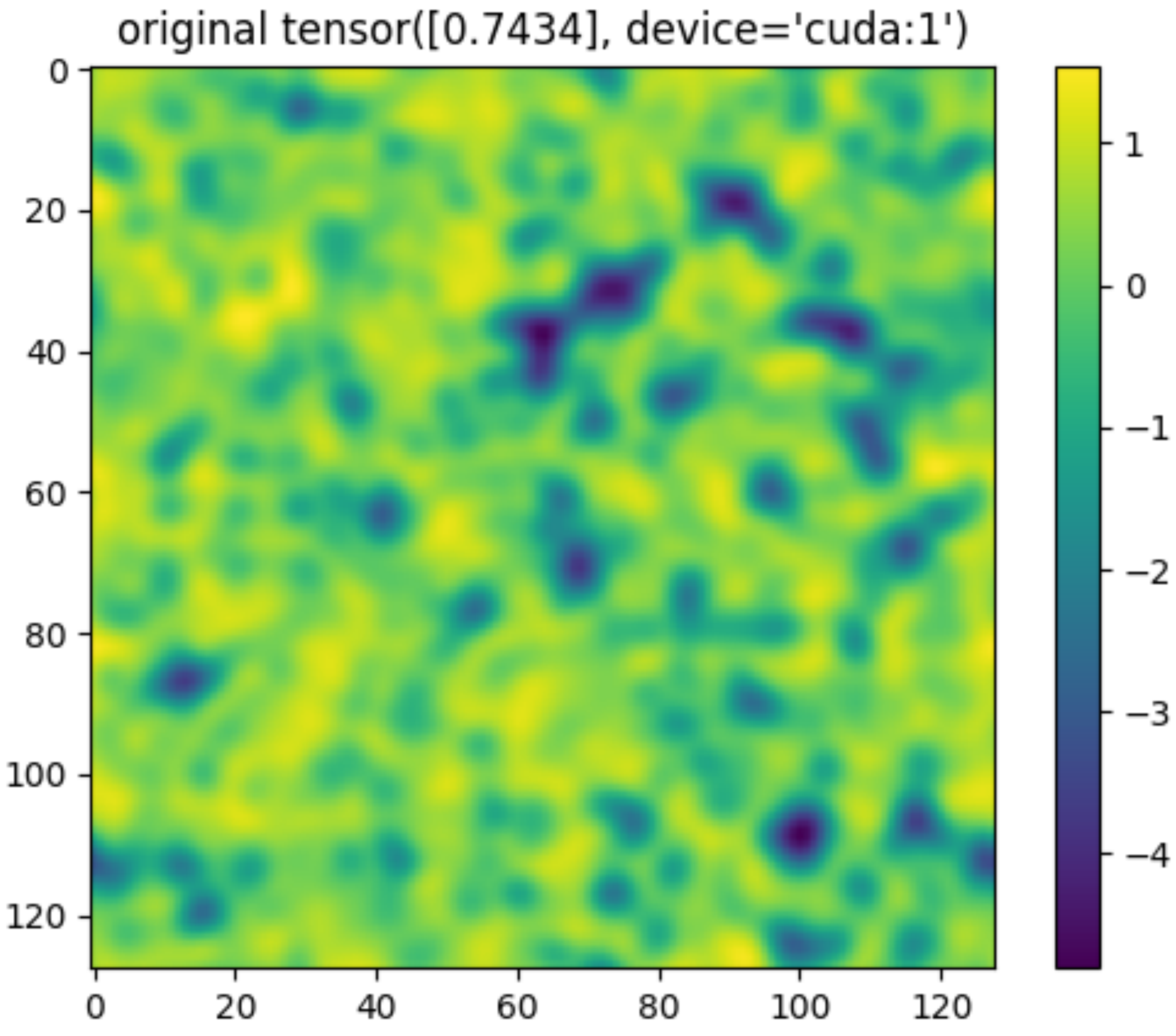
Transformation parameters $T_1, T_2, \ldots, T_N$

NN

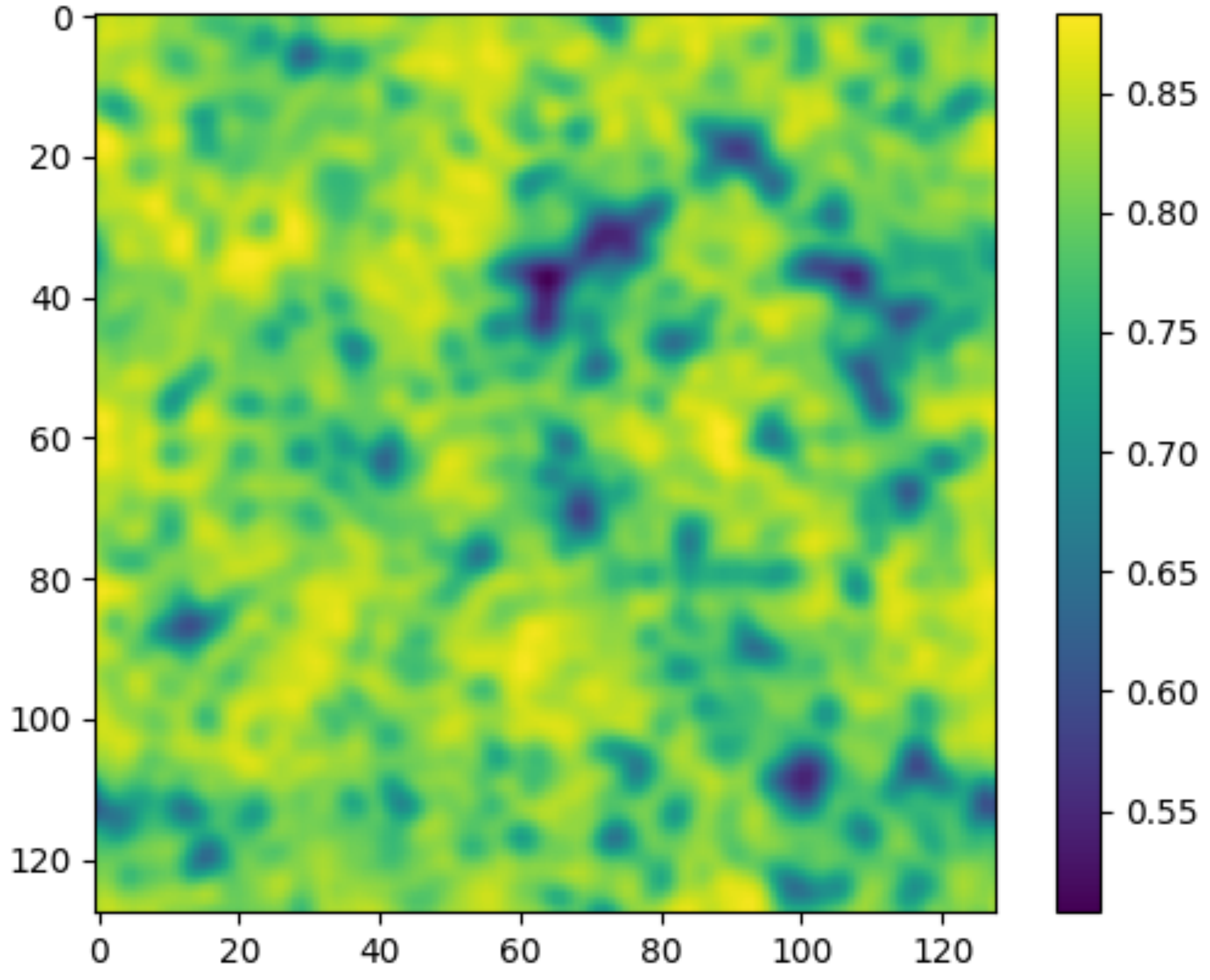
neutral fraction

Training data:
21cmFAST with different parameters of f^*. α^*, f_{esc}, α_{esc}, t^*, M_{turn}, L_X, E_0, α_X

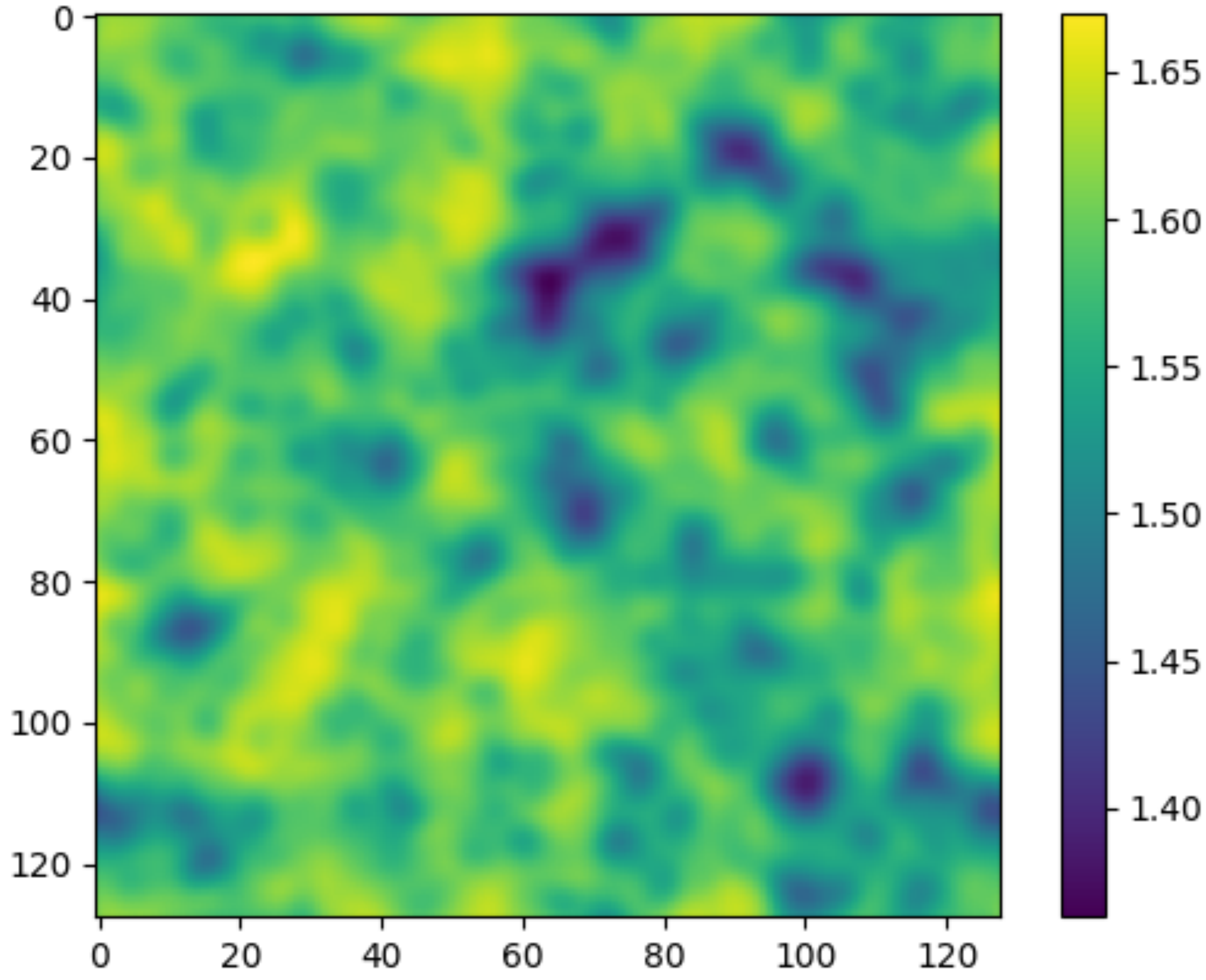
Result: transform to noise



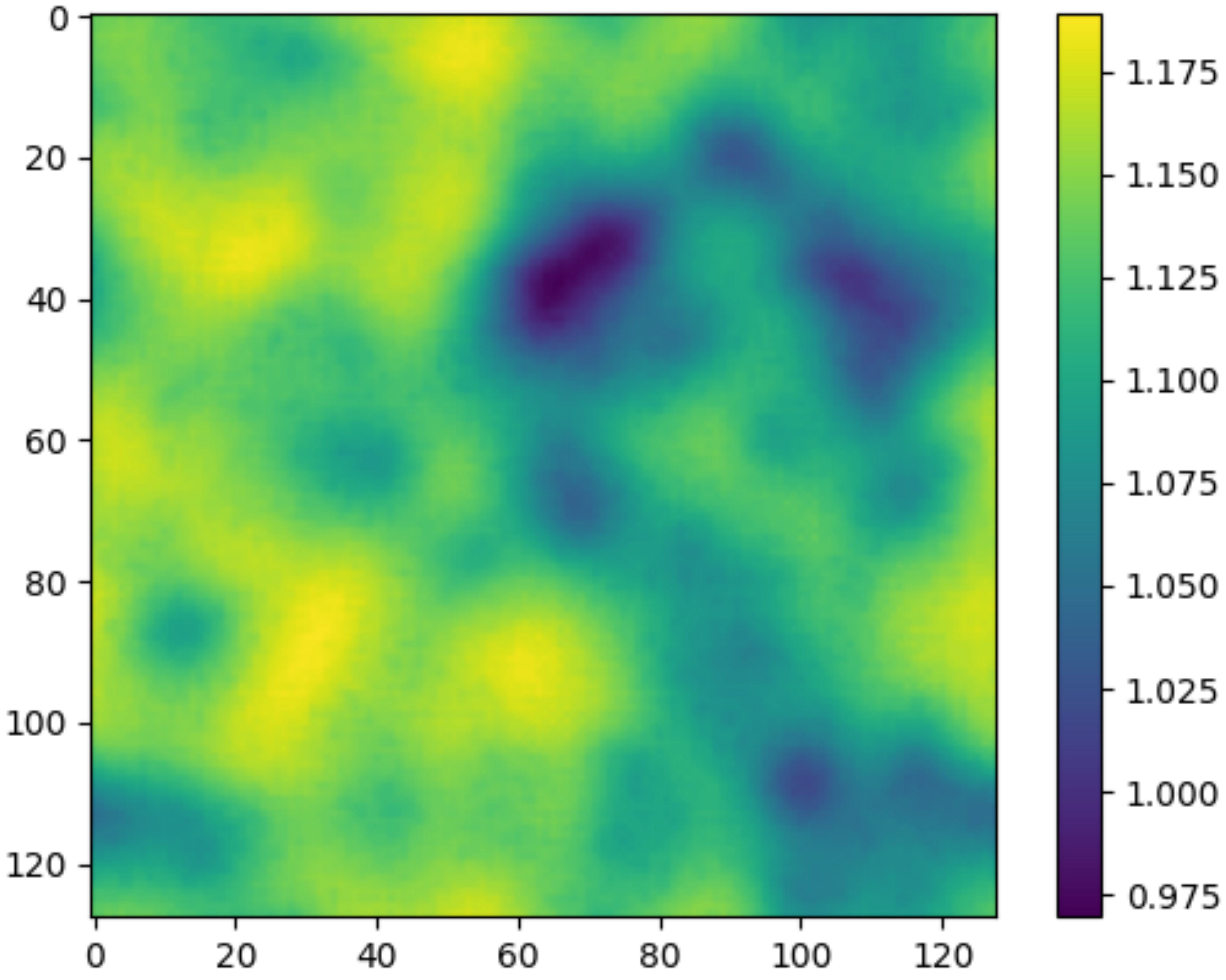
Result: transform to noise



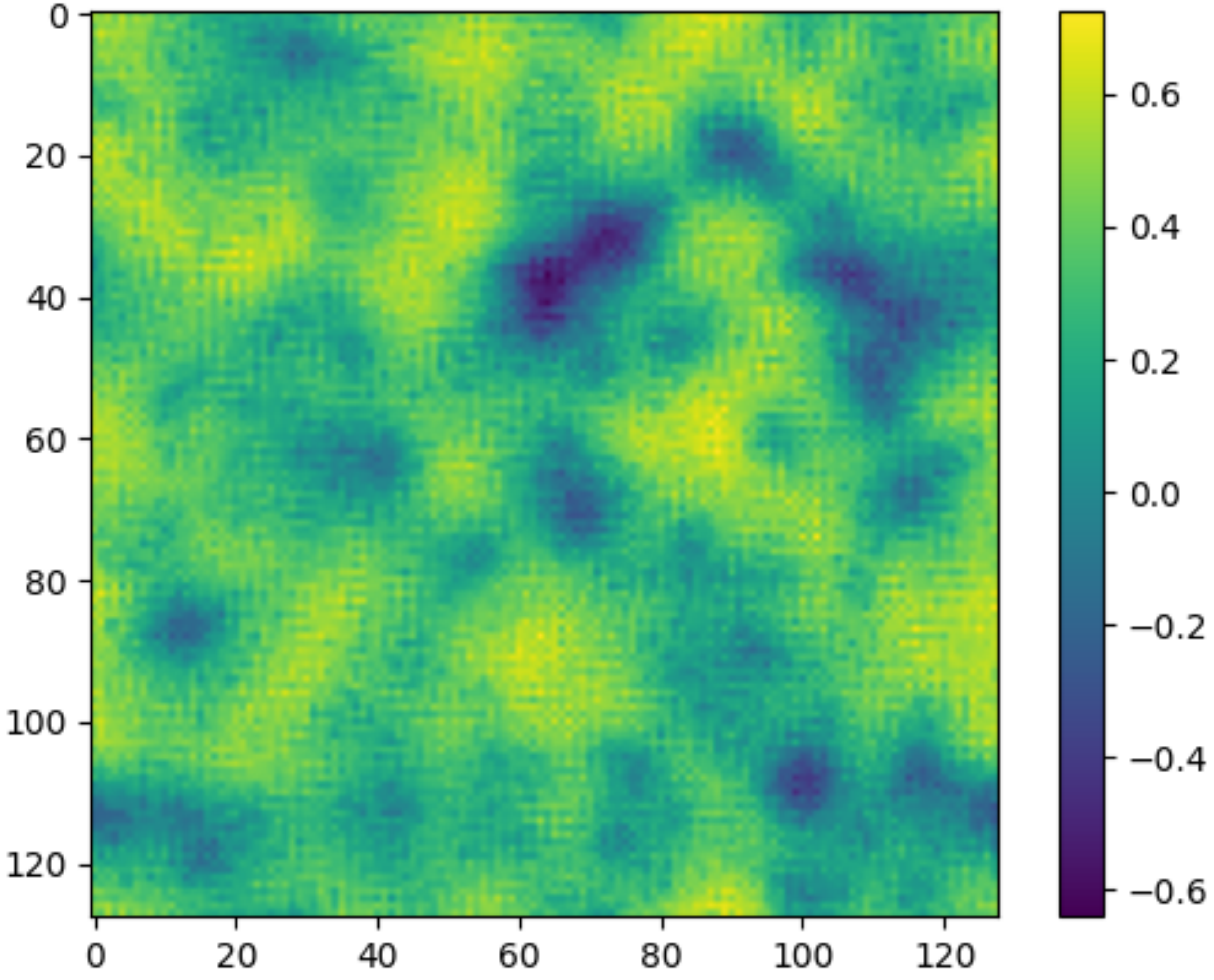
Result: transform to noise



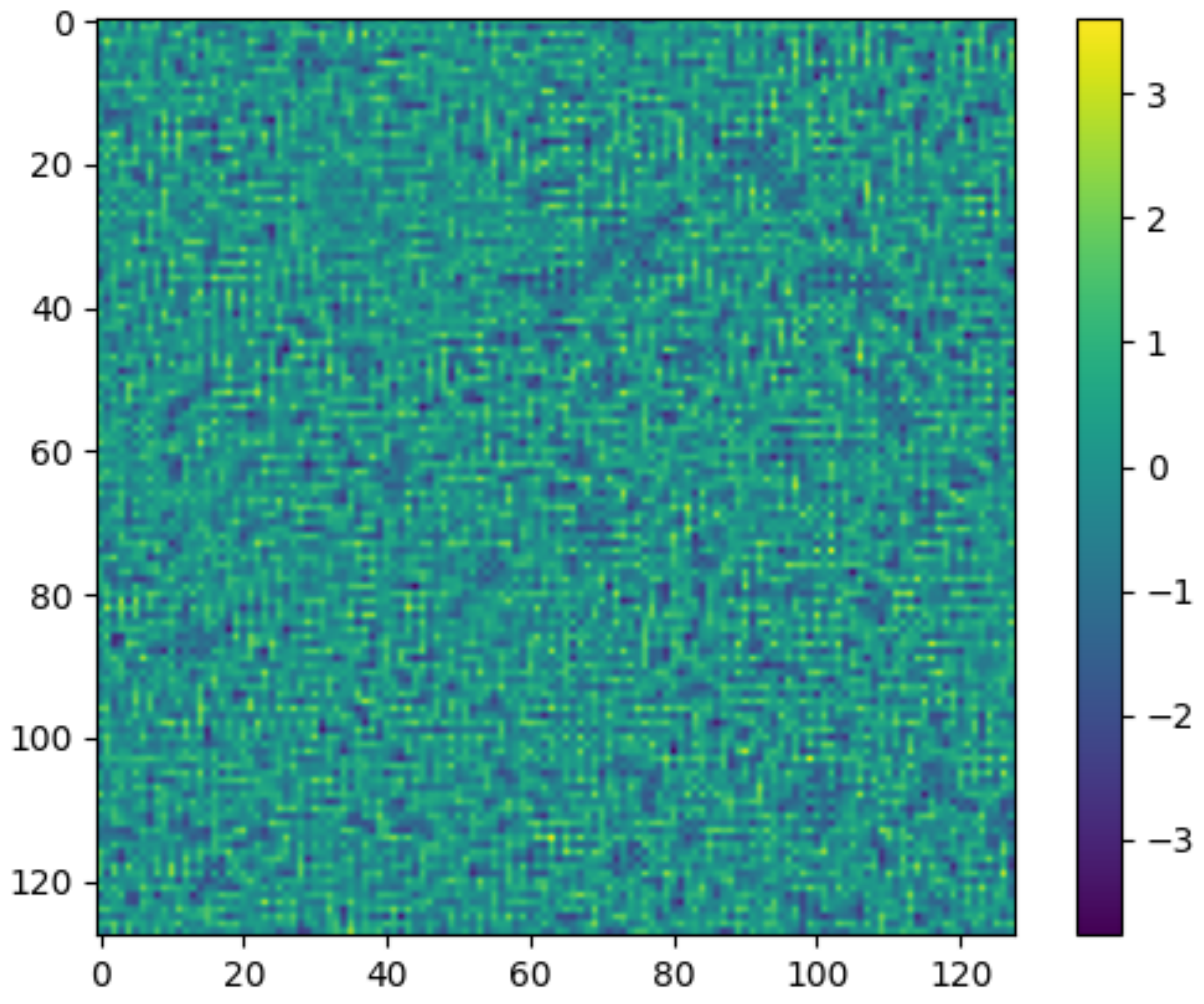
Result: transform to noise



Result: transform to noise

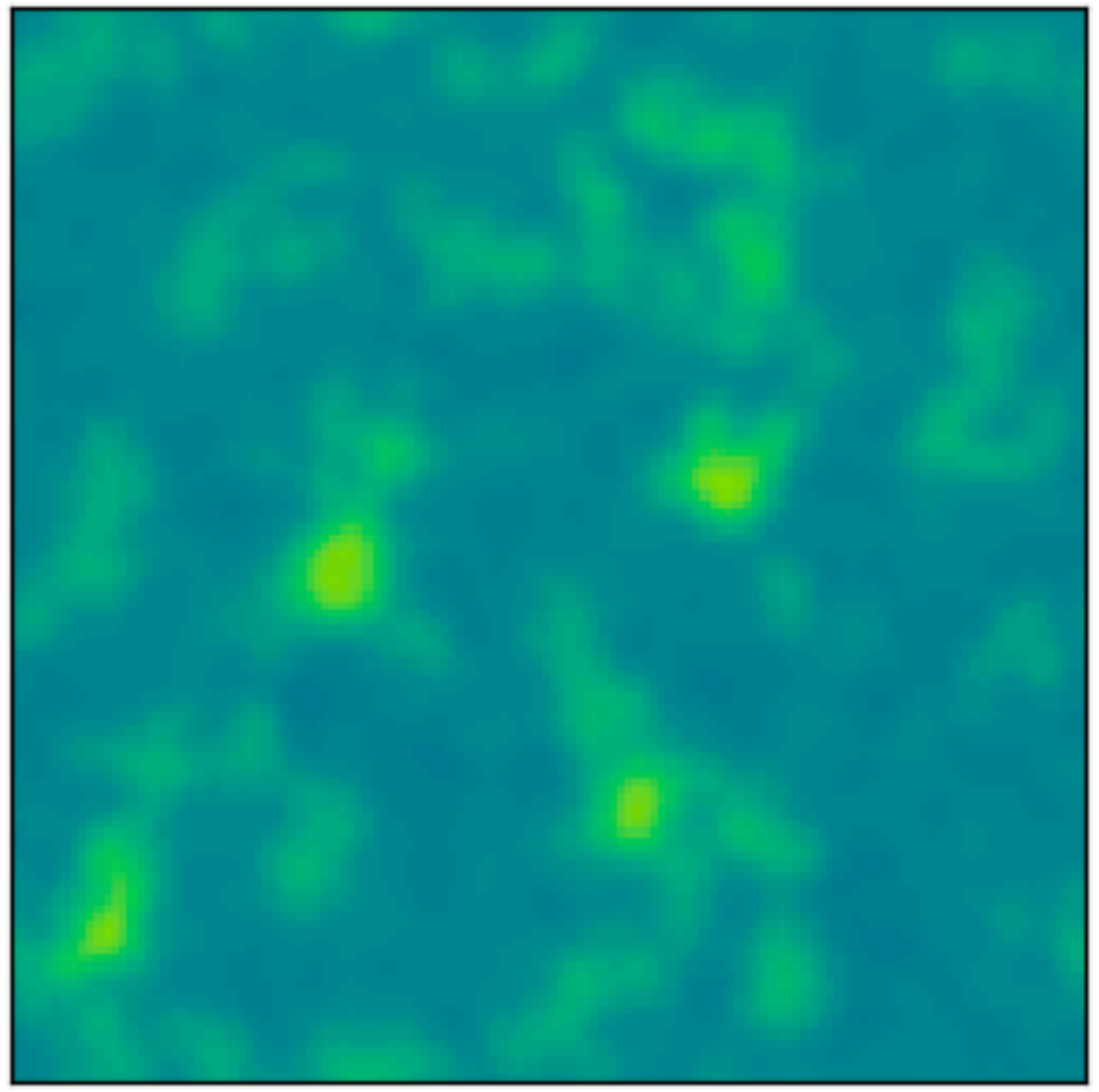


Result: transform to noise

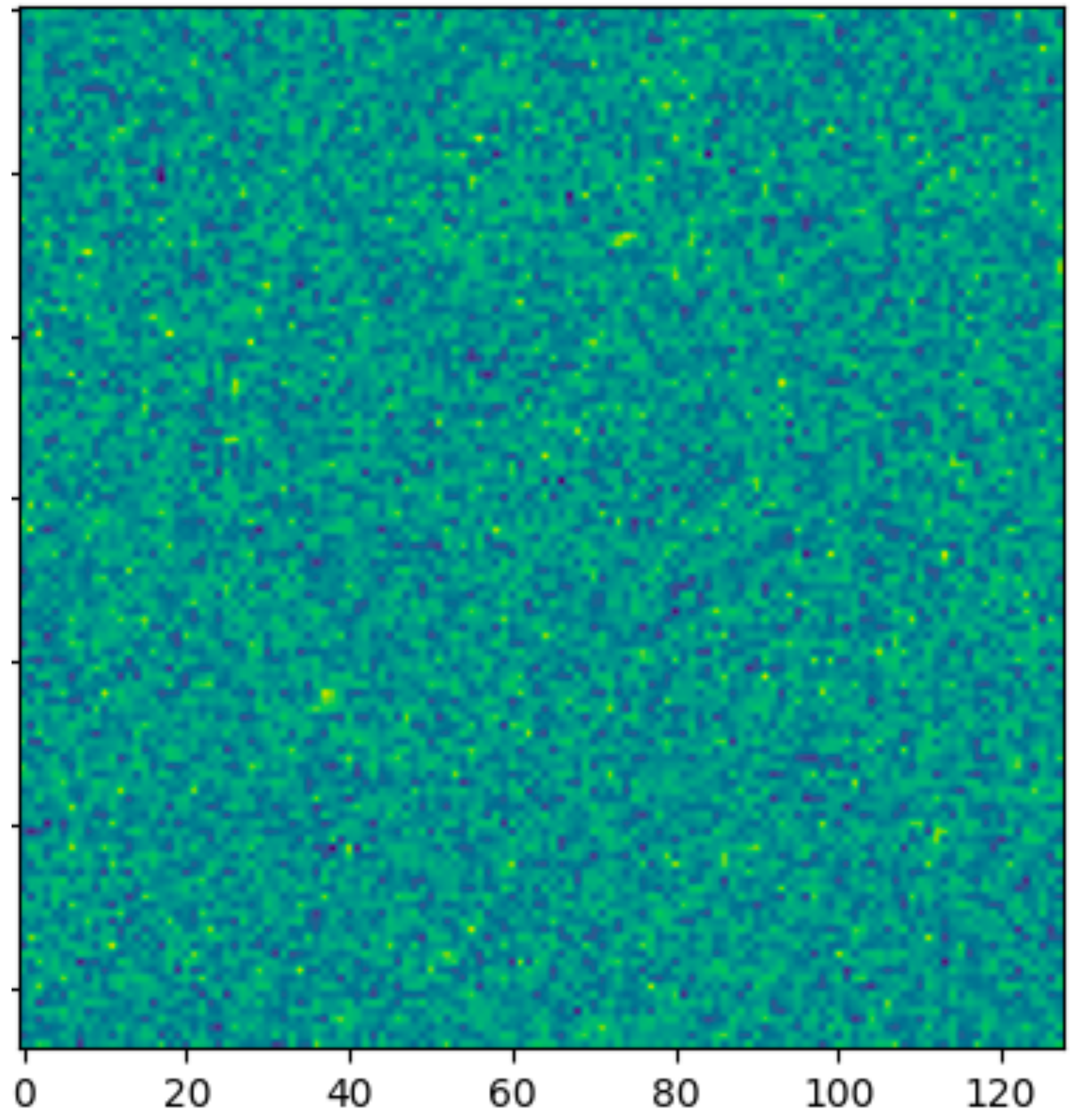


Generation of new images

21cm map + noise + smoothing



Latent variable



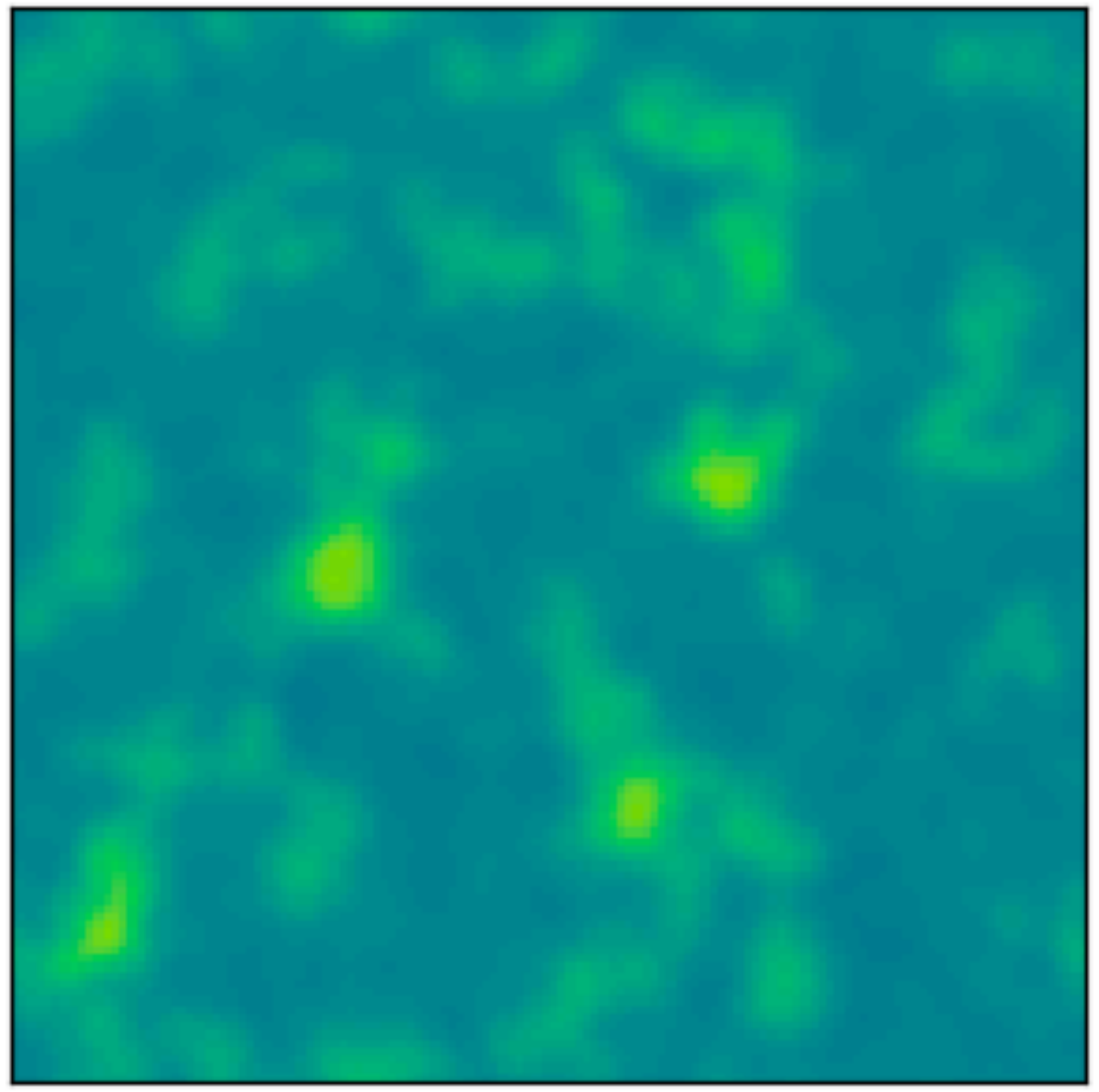
Transformation parameters
$T_1, T_2, \ldots, T_N$

NN

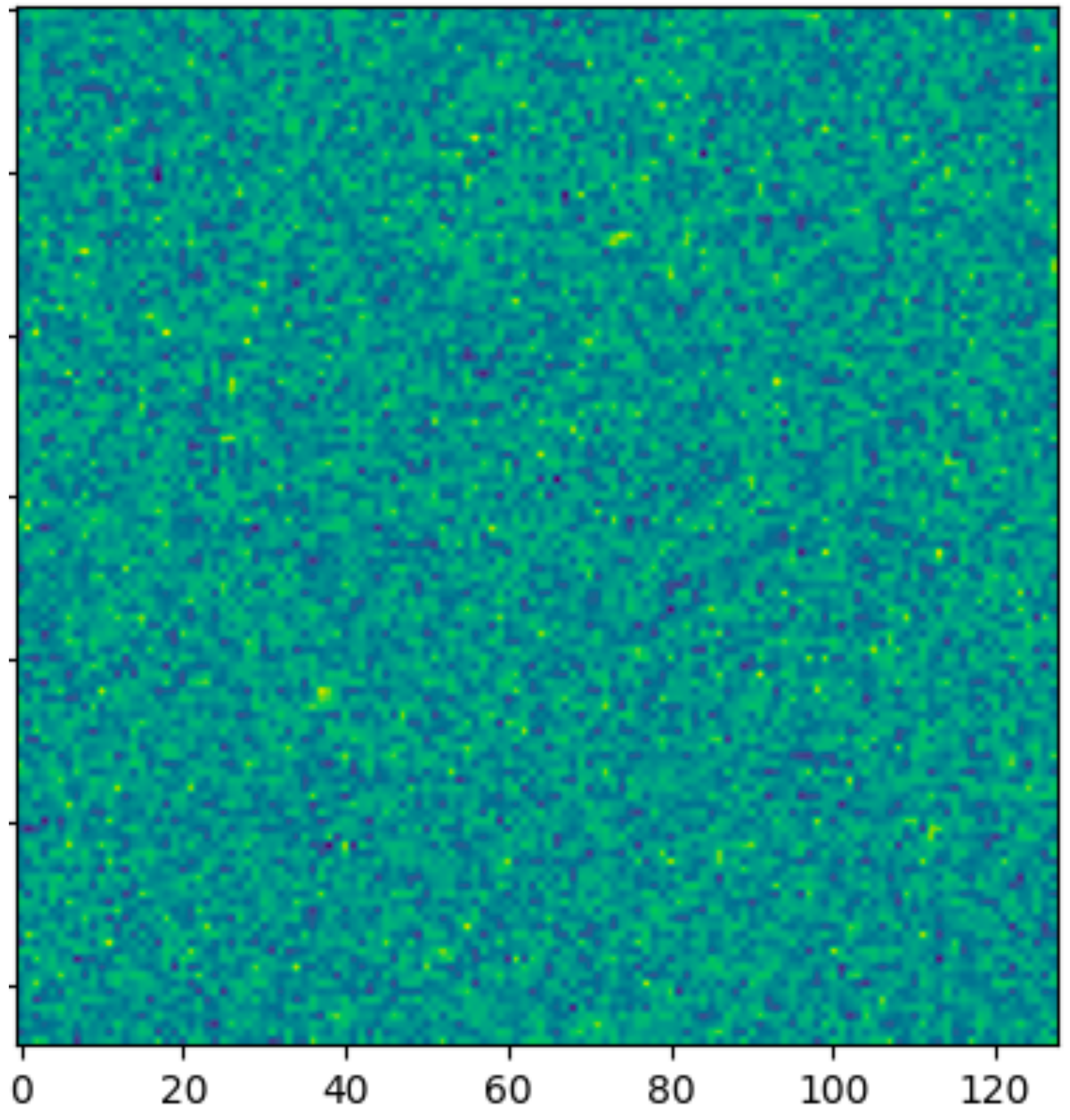
neutral fraction

Generation of new images

21cm map + noise + smoothing



Latent variable

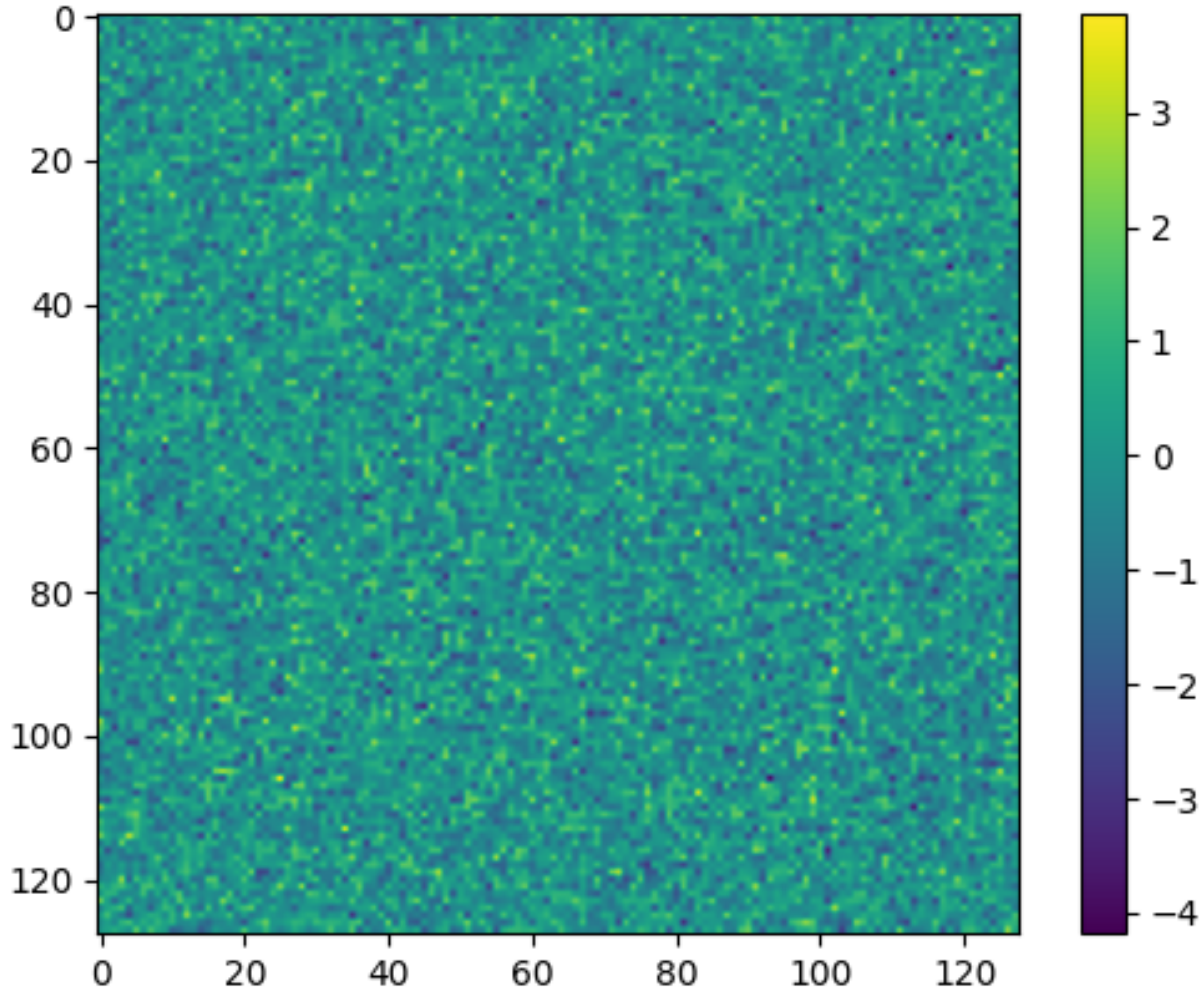


Transformation parameters
$T_1, T_2, \ldots, T_N$

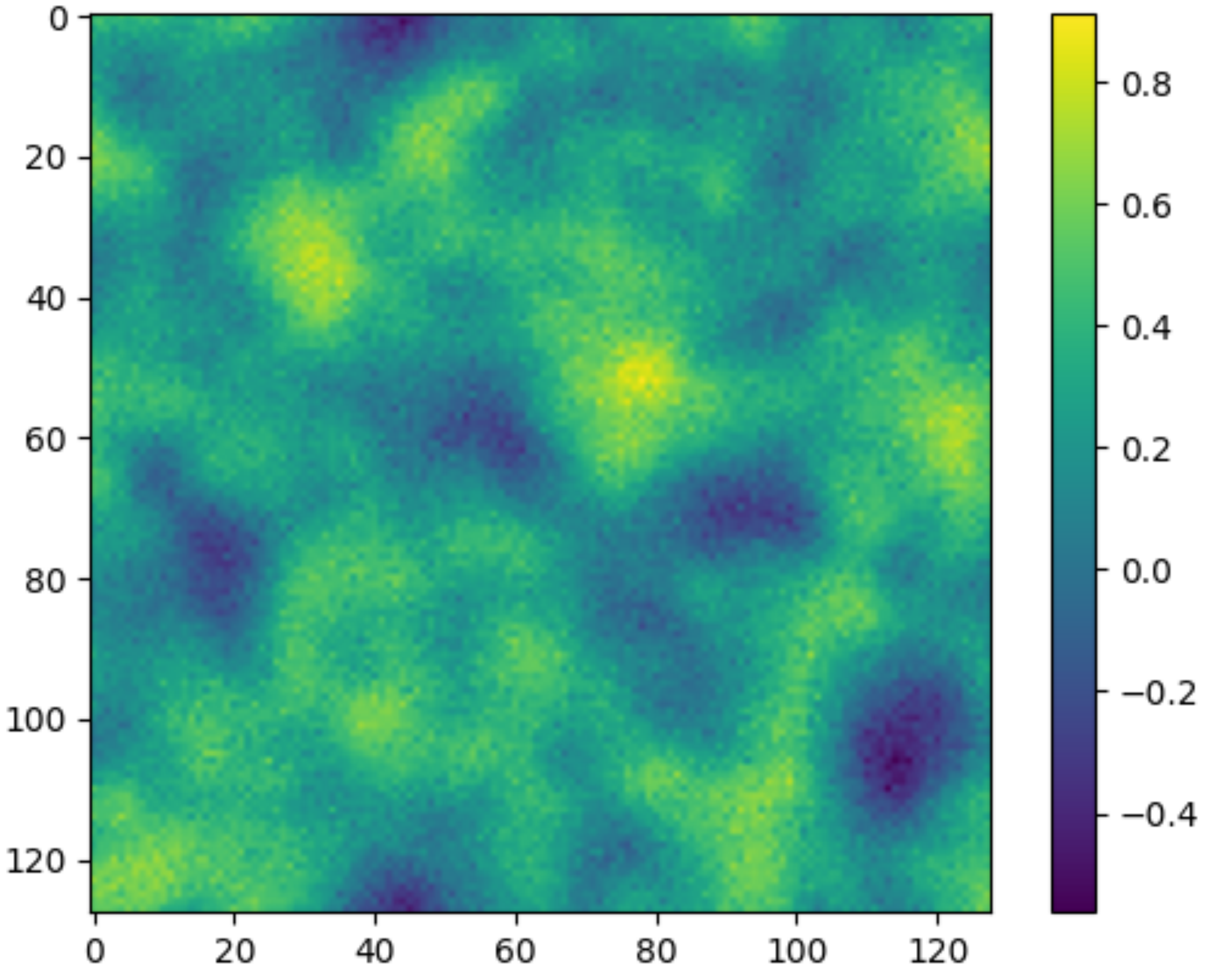
NN

neutral fraction

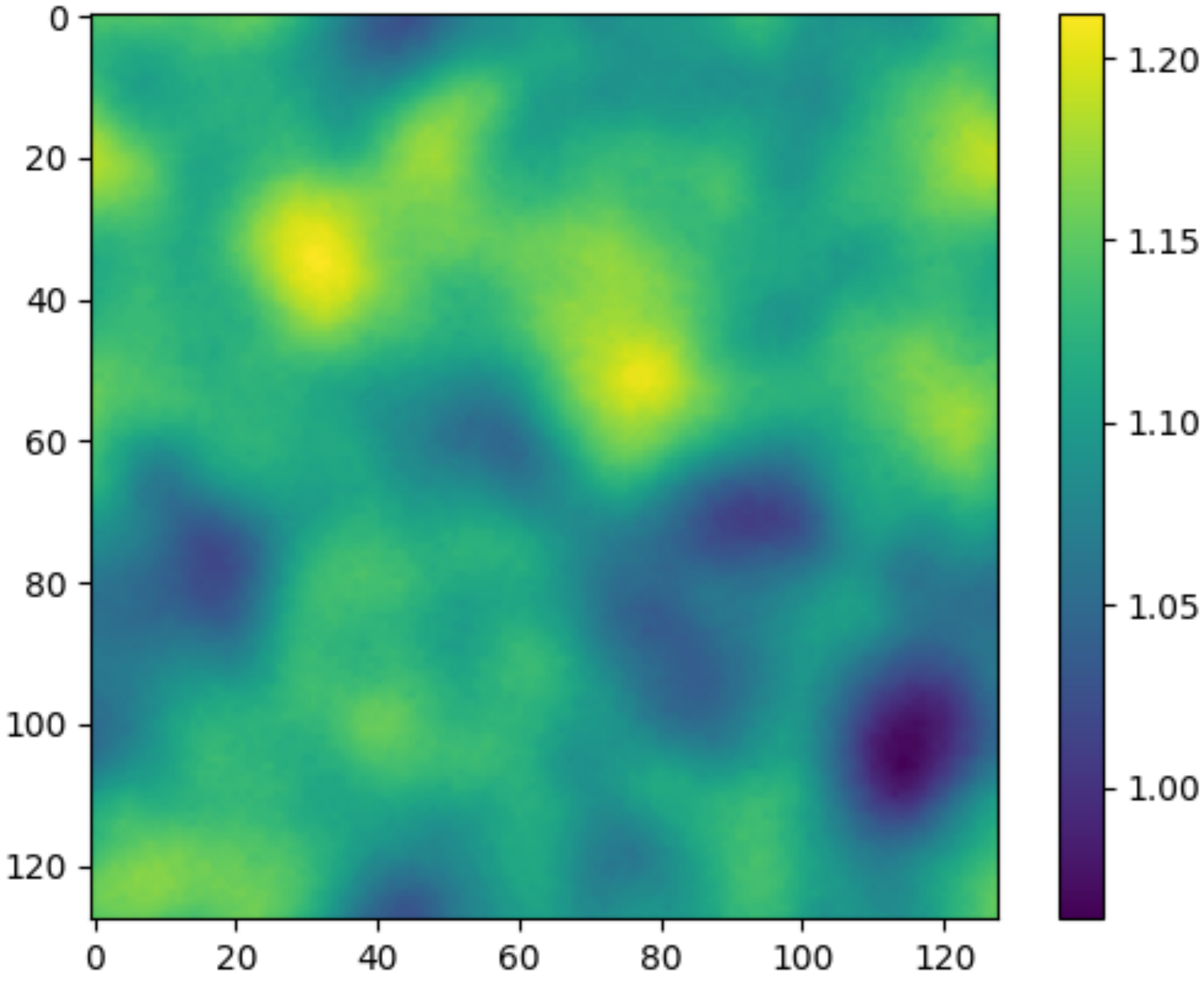
Result: transform from random noise



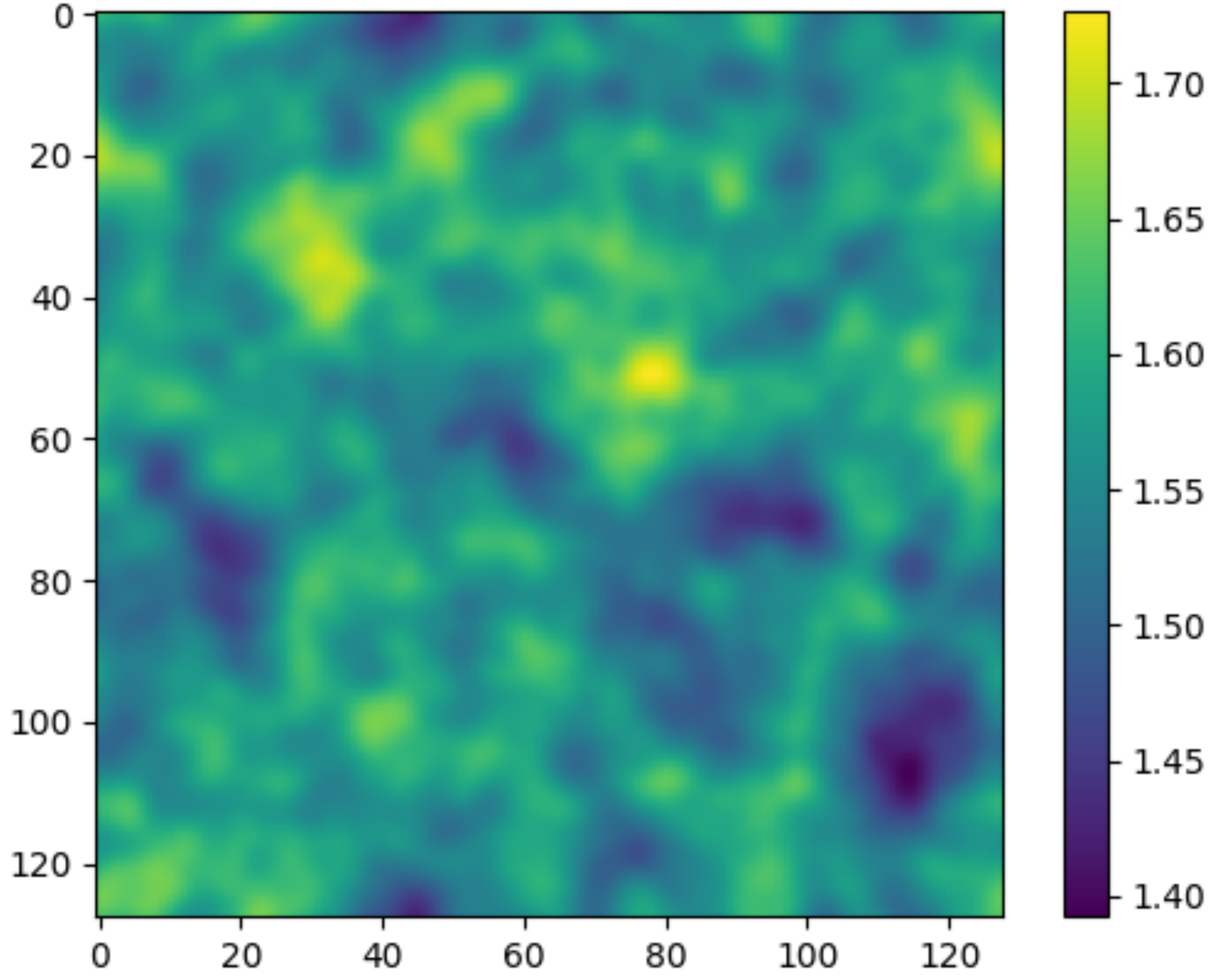
Result: transform from random noise



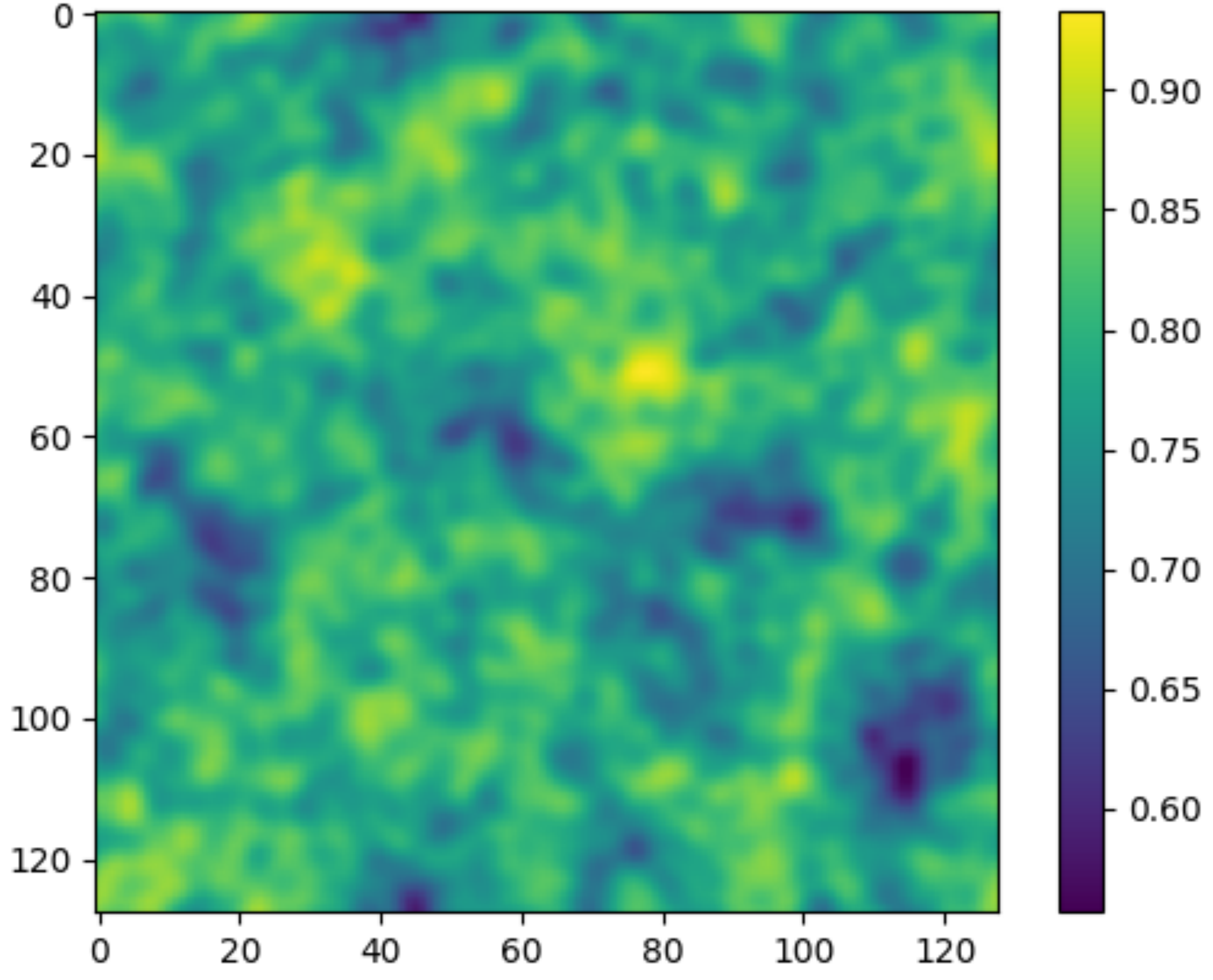
Result: transform from random noise



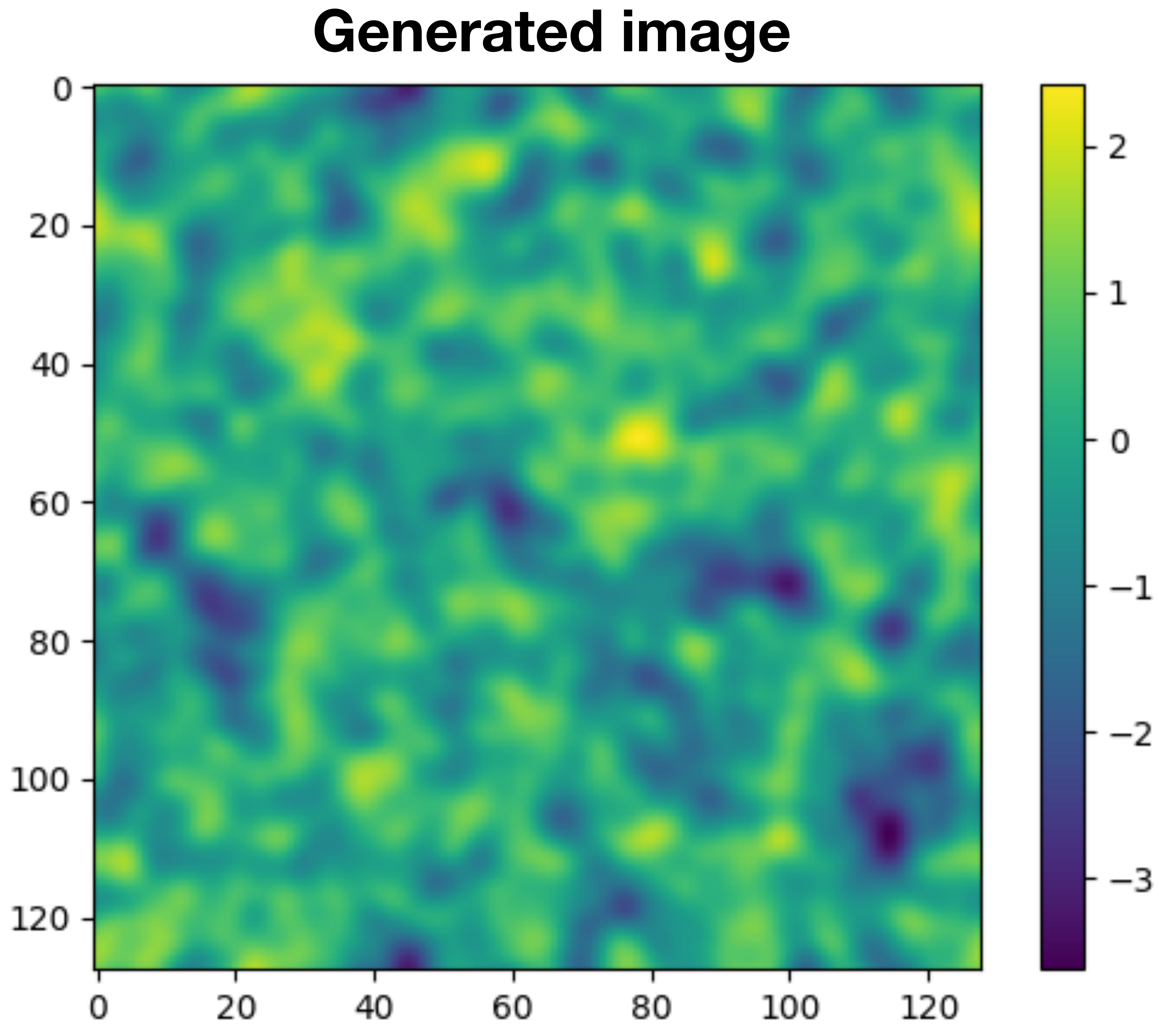
Result: transform from random noise



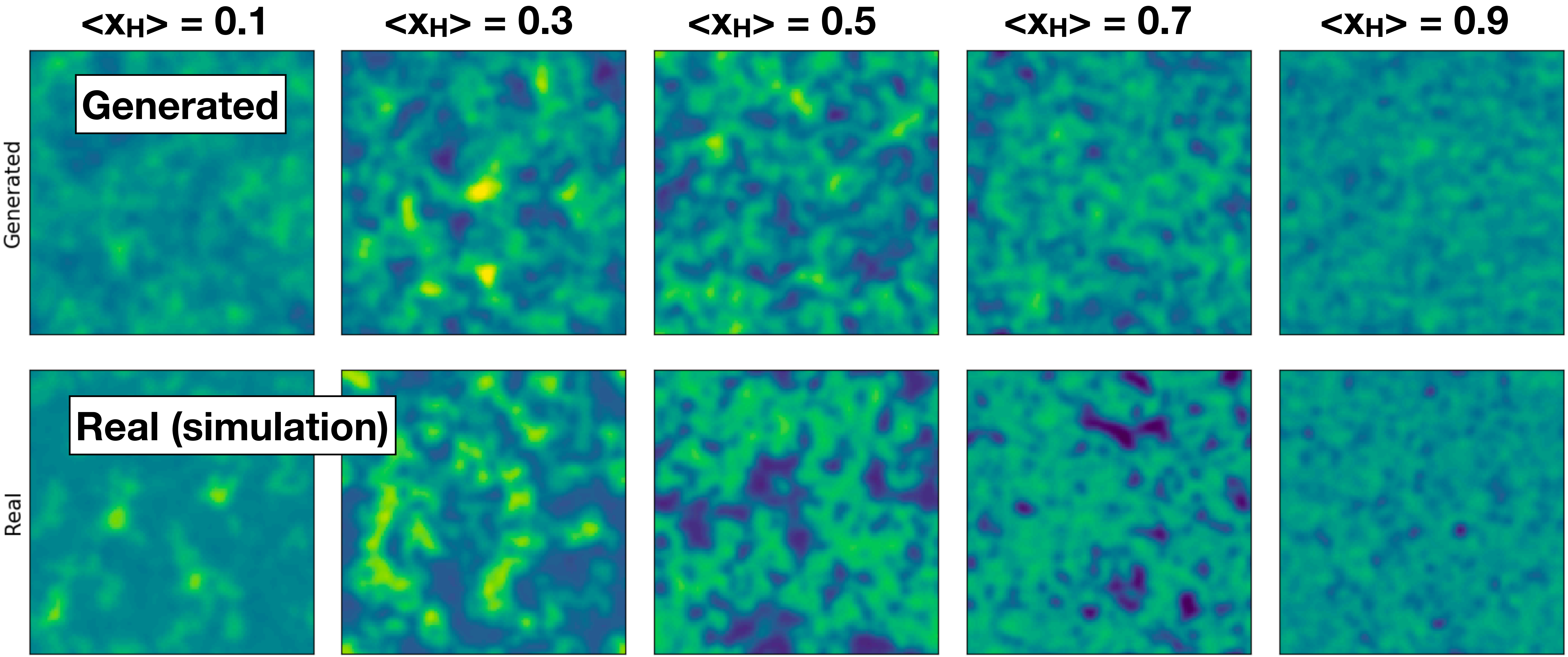
Result: transform from random noise



Result: transform from random noise



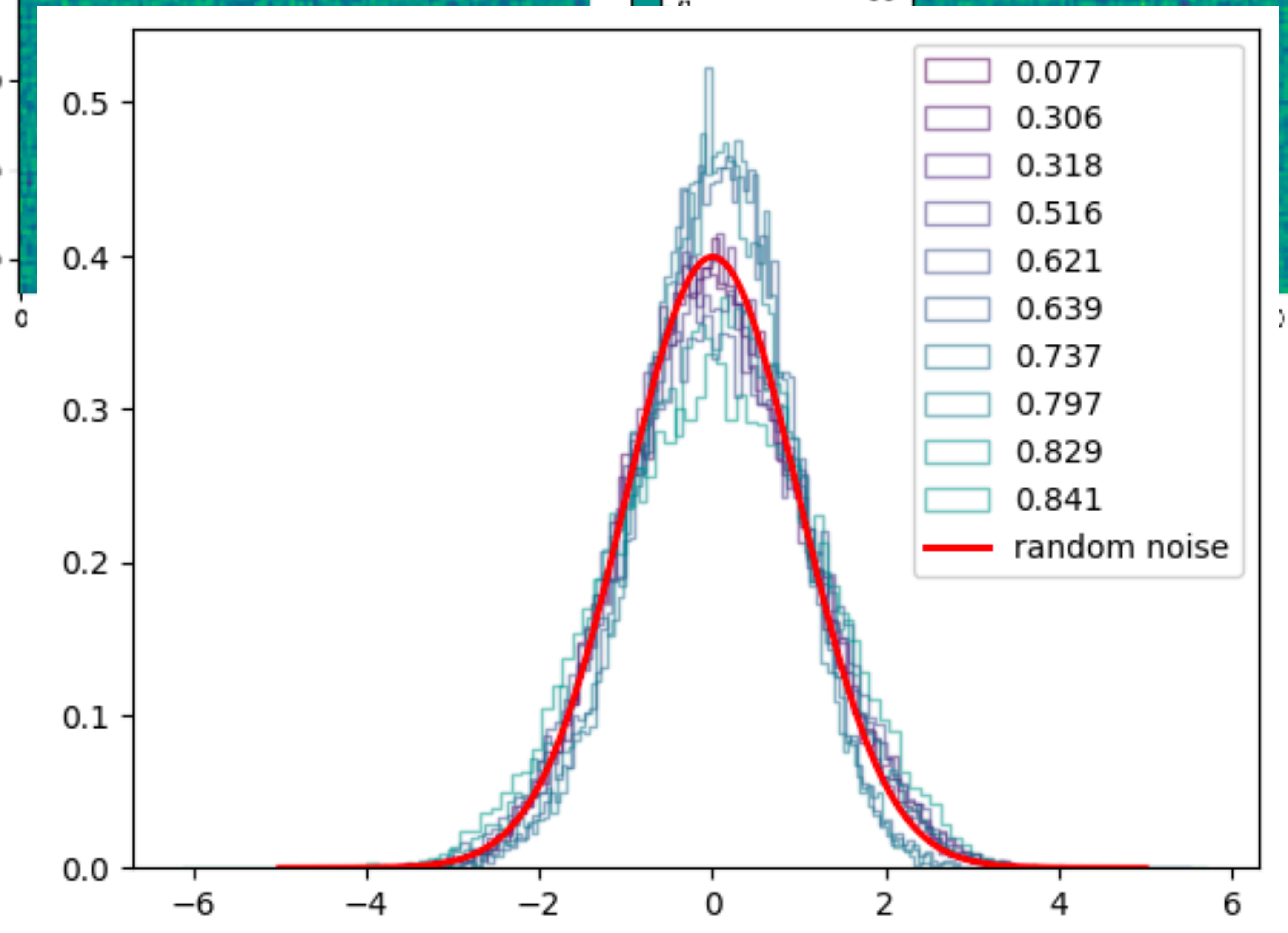
Result: generated images



Result: transformation into latent space

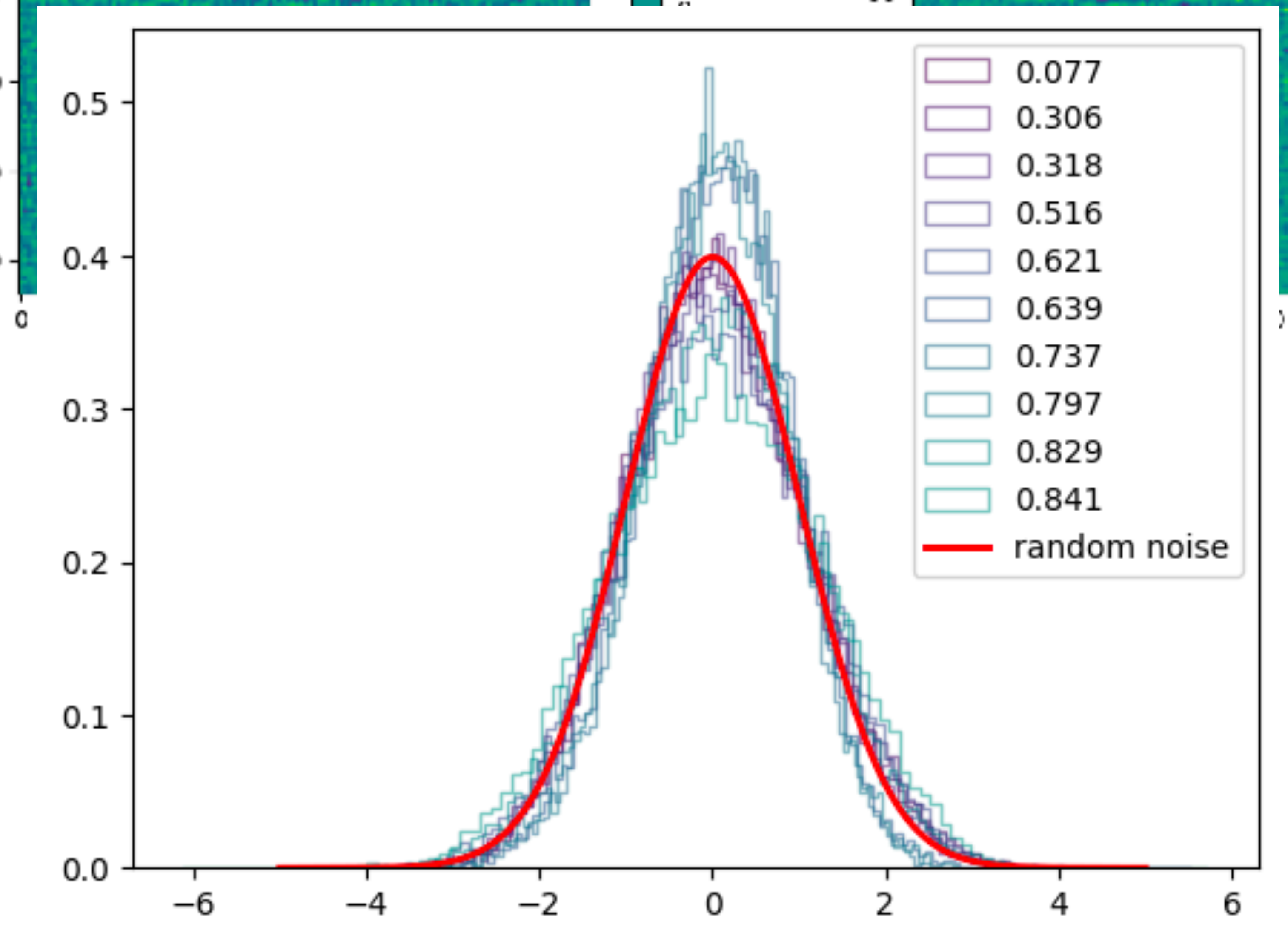
Randomly sampled simulation data (test data)

Result: transformation into latent space



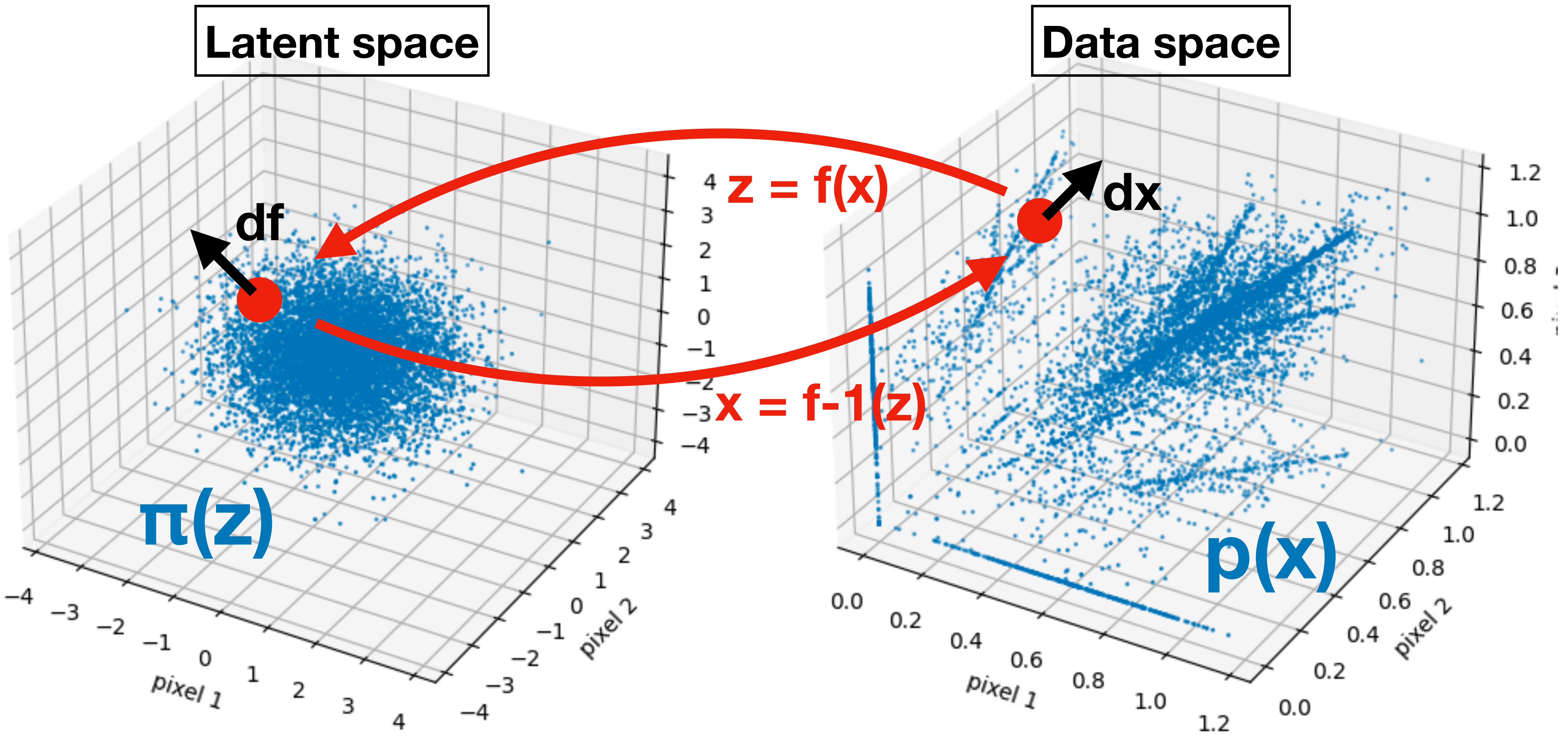
Transformation with correct parameters

Result: transformation into latent space



Transformation with incorrect parameters (x_{HI} = 0.8 for all cases)

Computation of the probability density



Assumption: latent space distribution π(z) is known

$$p(x) = \pi(f(x))\left|\det\left(\frac{\partial f(x)}{\partial x}\right)\right| = \pi(f(x))\Pi_{l=1}^{n}\left|\det\left(\frac{\partial f_l(x)}{\partial x}\right)\right|$$

When the model is conditional:

$$p(x\,|\,y) = \pi(f_y(x))\left|\det\left(\frac{\partial f_y(x)}{\partial x}\right)\right| \rightarrow p(y\,|\,x) \propto p(x\,|\,y)p(y)$$

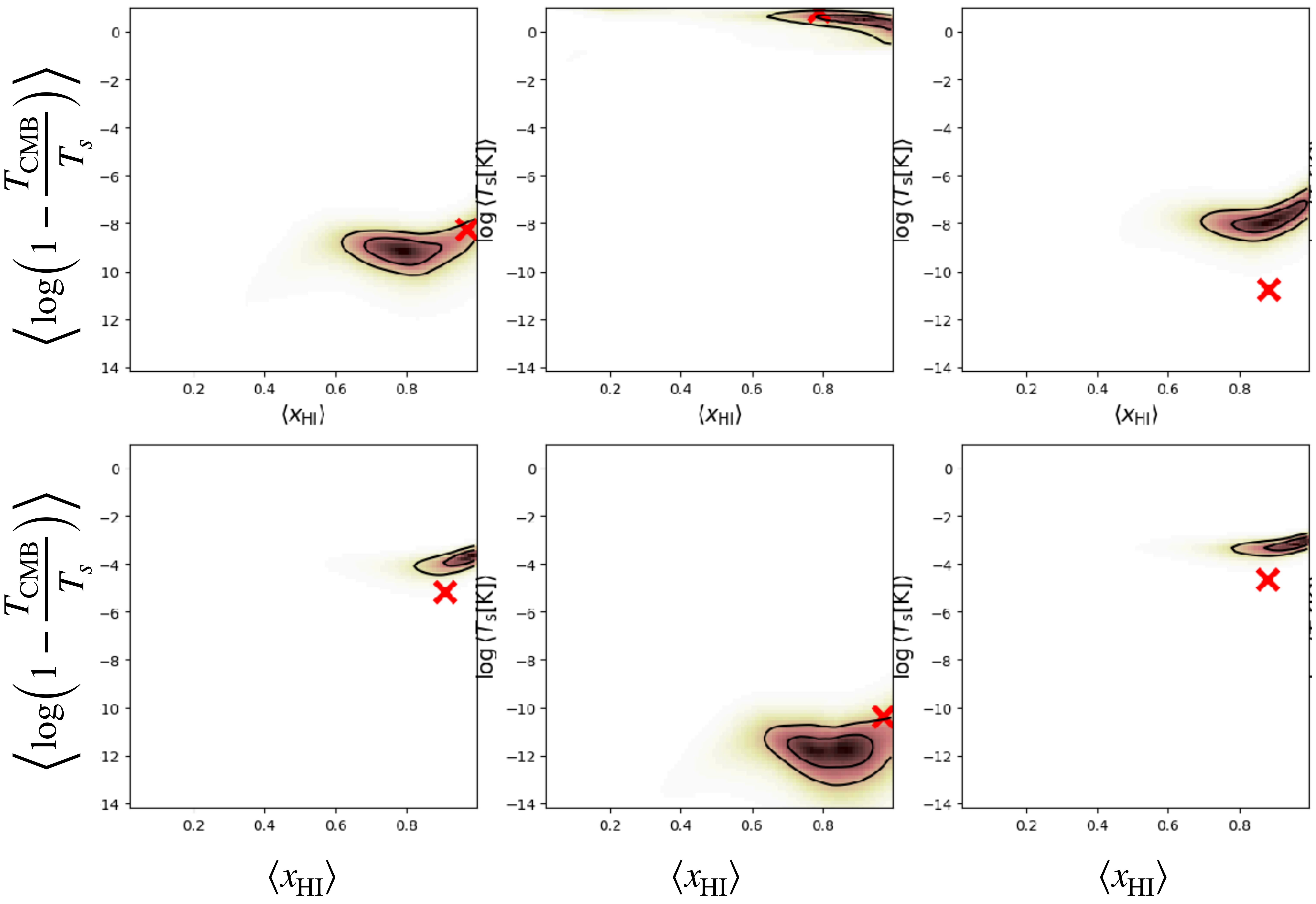
Result: parameter inference

$x_H = 0.1$ $x_H = 0.3$ $x_H = 0.5$ $x_H = 0.7$ $x_H = 0.9$

Is it learning something beyond power spectrum?

Test with three samples with similar power

Inferring both ionization and heating state



21cm signal

$$\delta T_{21} = T_0 x_{\rm HI}(1+\delta_\rho)\left(1-\frac{T_{\rm CMB}}{T_{\rm s}}\right)\frac{H(z)}{H(z)+\frac{dv_{||}}{dr_{||}}}$$

Summary

- Normalizing flow could be a good tool for fully extracting information from EoR 21 cm maps

- Issues
  - Noise and smoothing are too simple
  - Foregrounds are not included
  - Tested only one specific simulation
  - Is it better to directly use uv-plane than images?

  - Could deep-learning methods extract more information than using summary statistics?
  - Is NF better than the other CNN-based models? In what aspect?

Questions, comments, and suggestions are welcome!