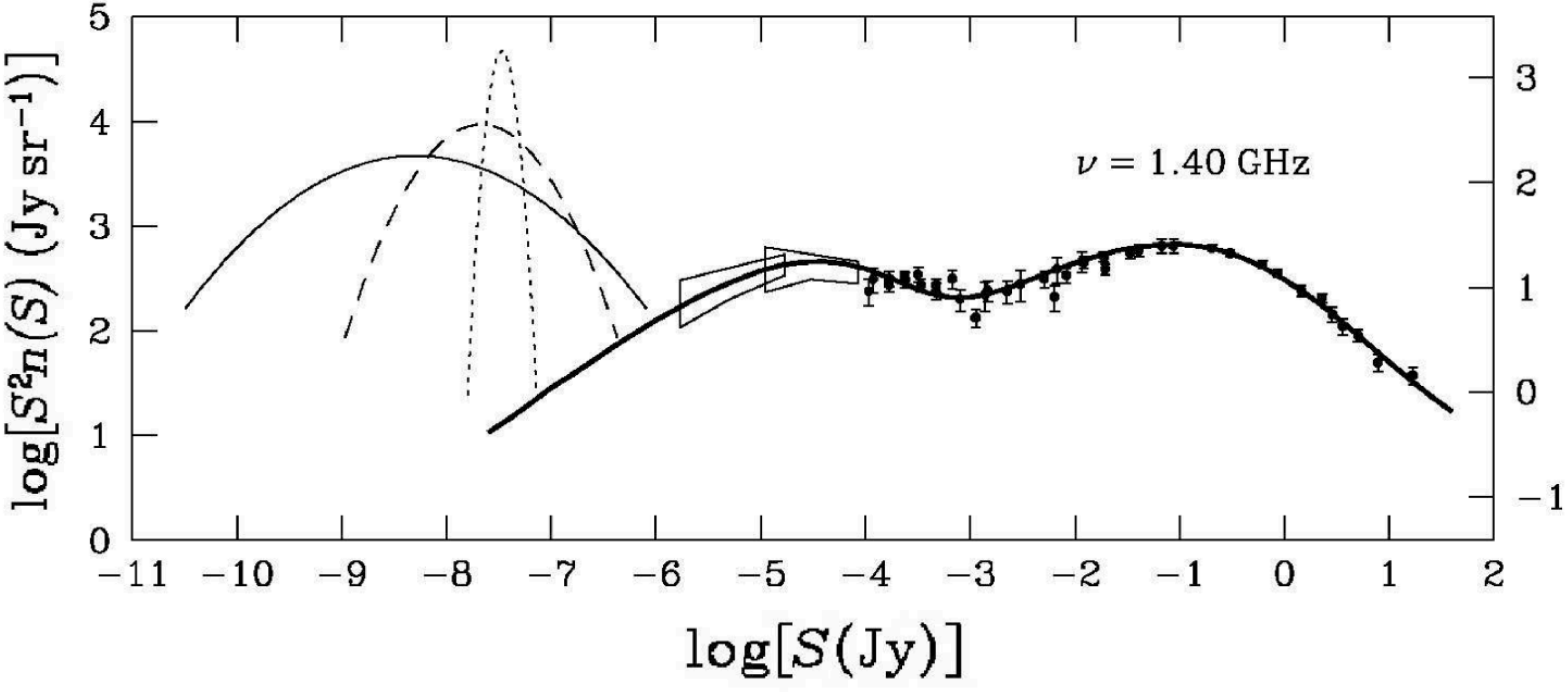
Constraining below-threshold radio source counts with machine learning

M. Regis, A. Scaffidi, M. Taoso, E. Todarello
(University of Turin and INFN, Turin)

hosts
Radio Synchrotron Background

Barolo, 16/06/2022

Source counts



J. Condon et al., 758 (2012) 23

Possible new source population

- dark matter decays and annihilations
- ultracompact halos
- dark stars in the early universe
- supernovae of massive population III stars
- dense nuggets of quarks
- primordial blackholes

Our final goal

Use a Convolutional Neural Network (CNN)

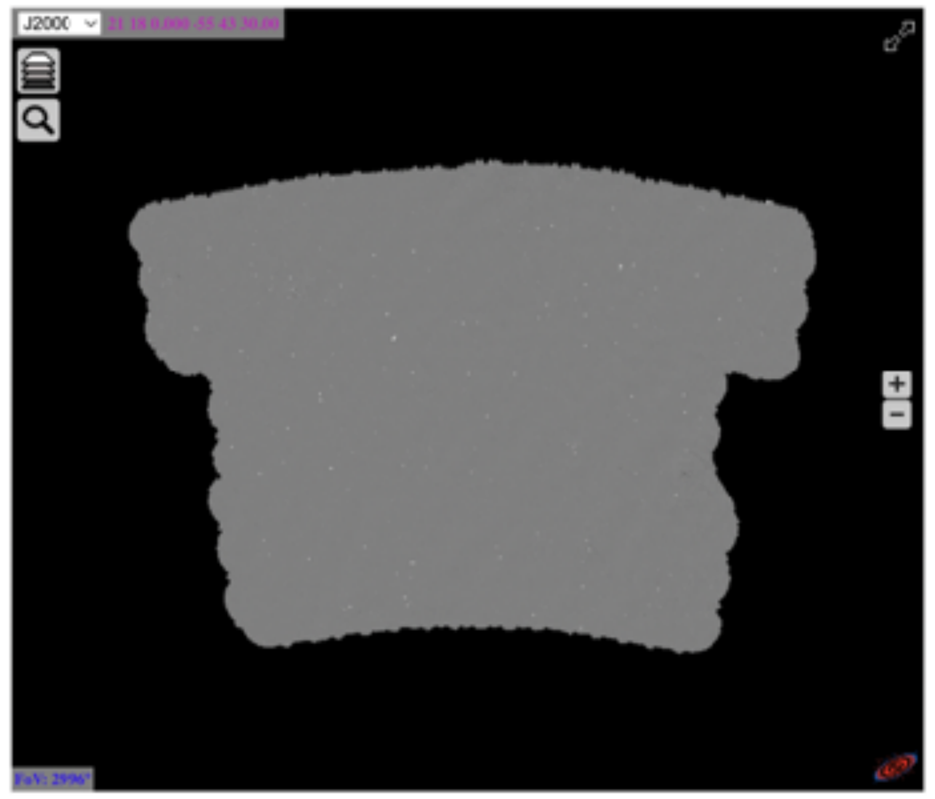
to extract the source counts at low flux (between 10^{-5} and 10^{-7}Jy)

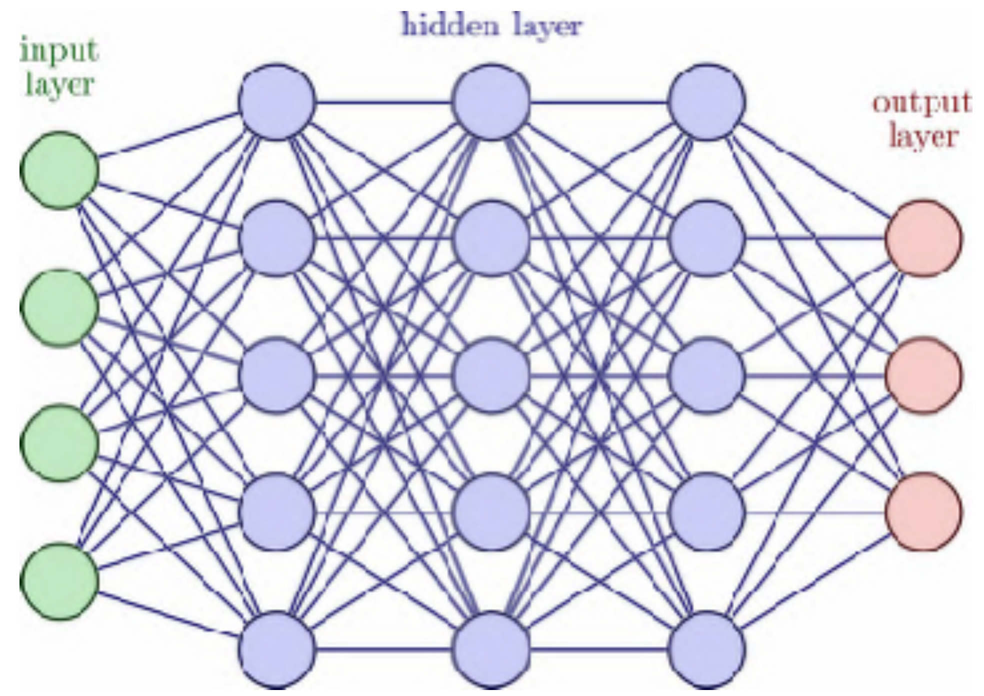
from the Evolutionary Map of the Universe (EMU) radio survey

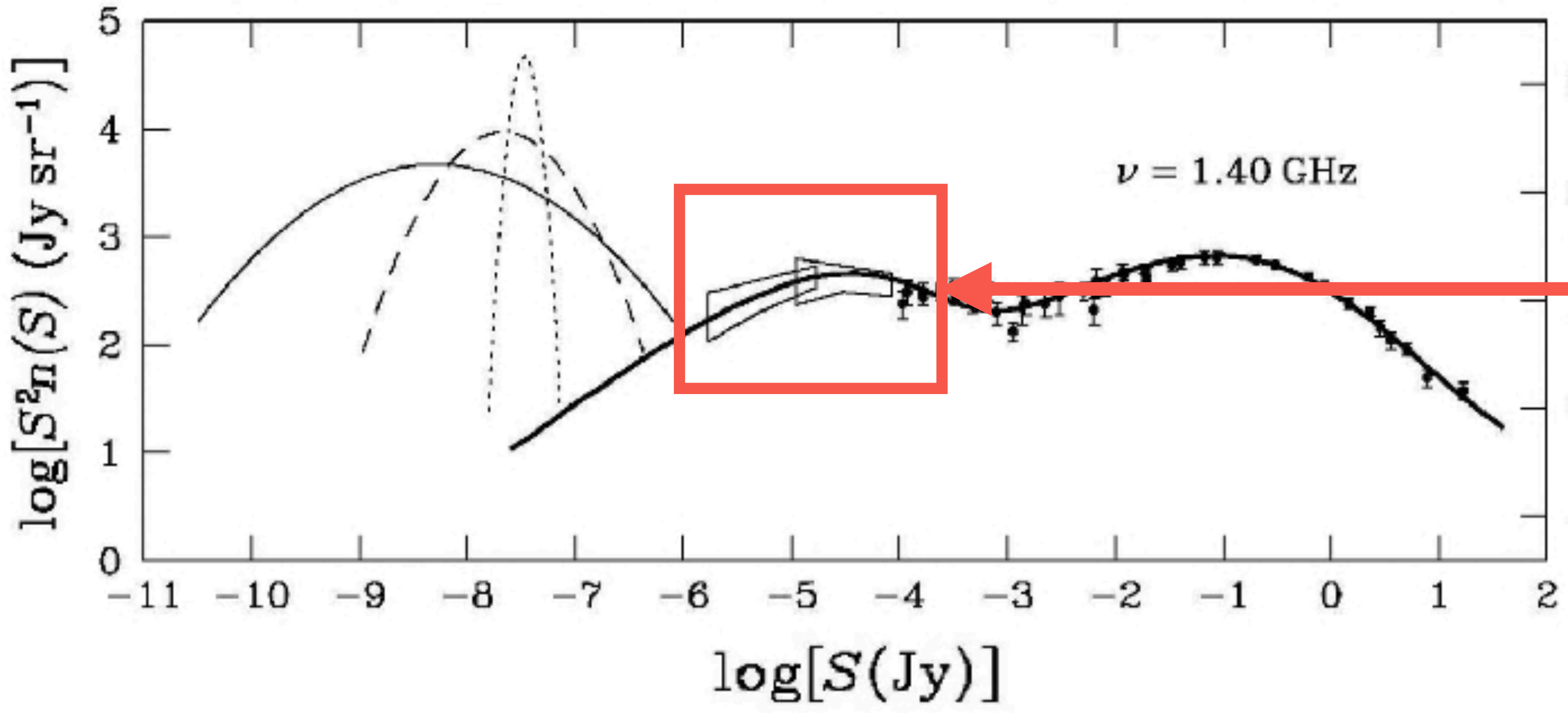
Can CNN "see through" noise and artefacts?

Can CNN correct for incompleteness, Eddington bias, resolution bias?

EMU 940 MHz Pilot Survey



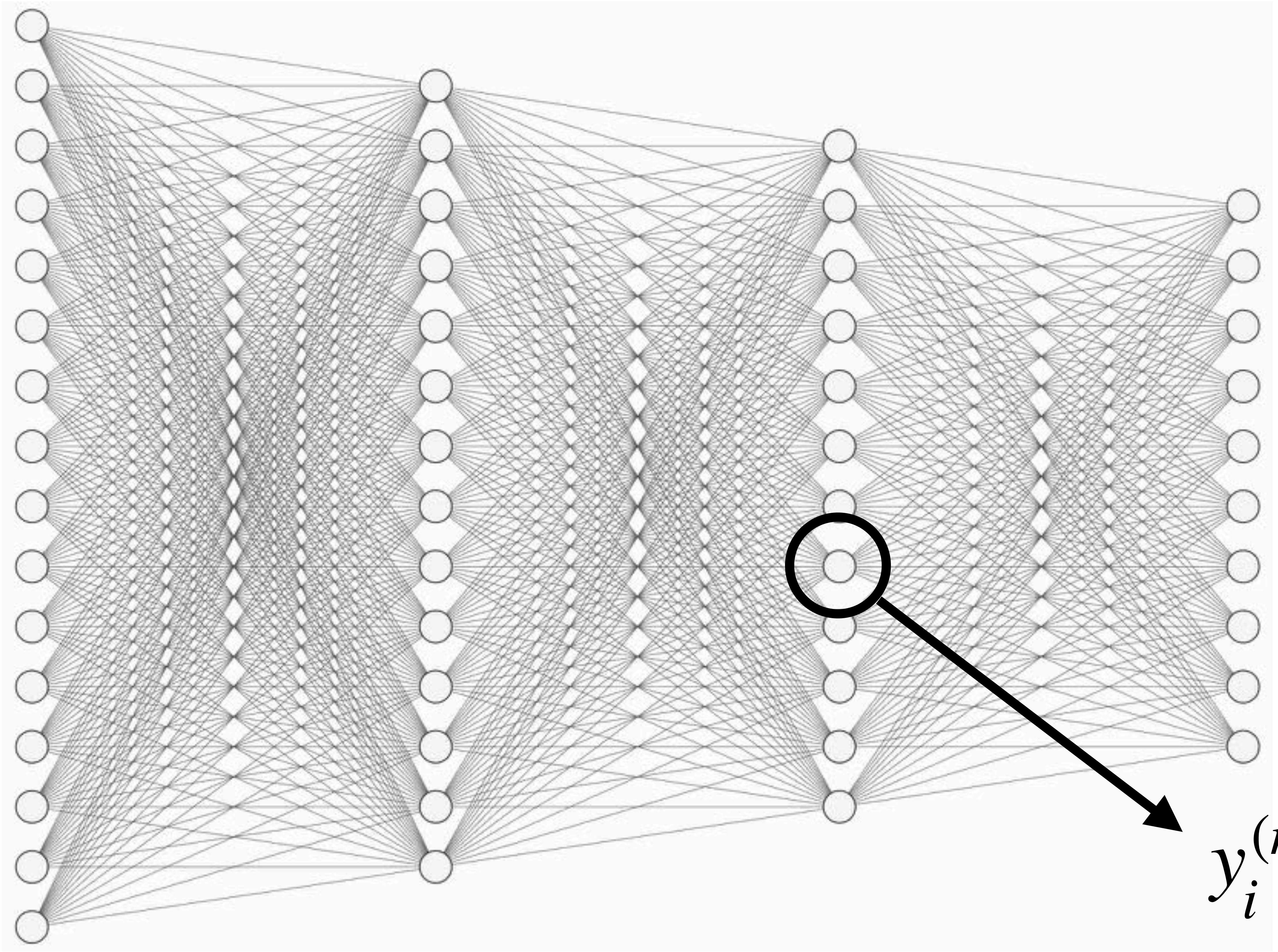




Obtained with P(D) method

Neural Networks

Neural Networks



Number 8

Activation

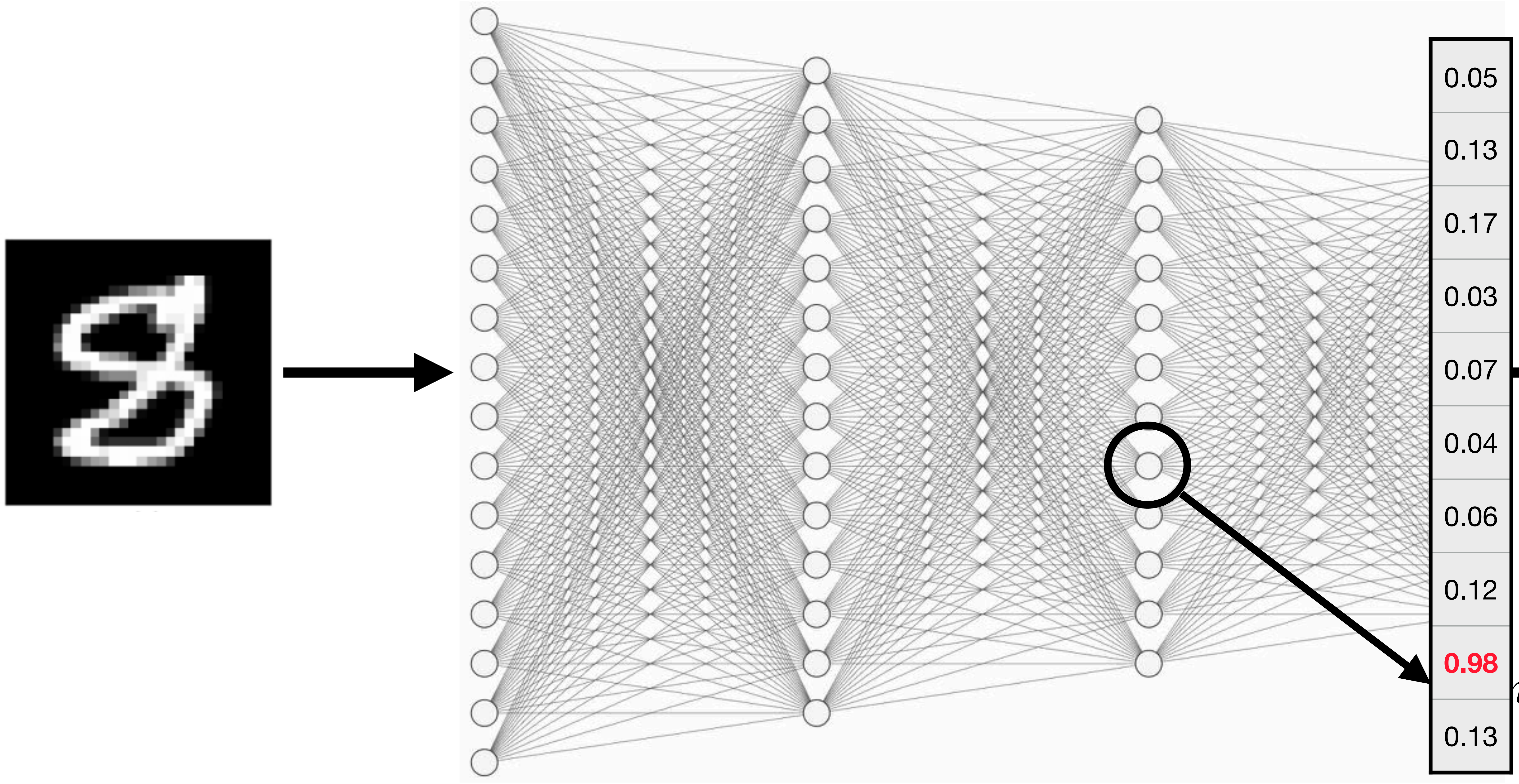
$$y_i^{(n+1)} = \sigma\left(\sum_{j=1}^{N} w_{ij}\, y_j^{(n)} + b_i^{(n)}\right)$$

Input layer X

Hidden layers $f(X)$

Output layer Y

Neural Networks



Number 8

Activation

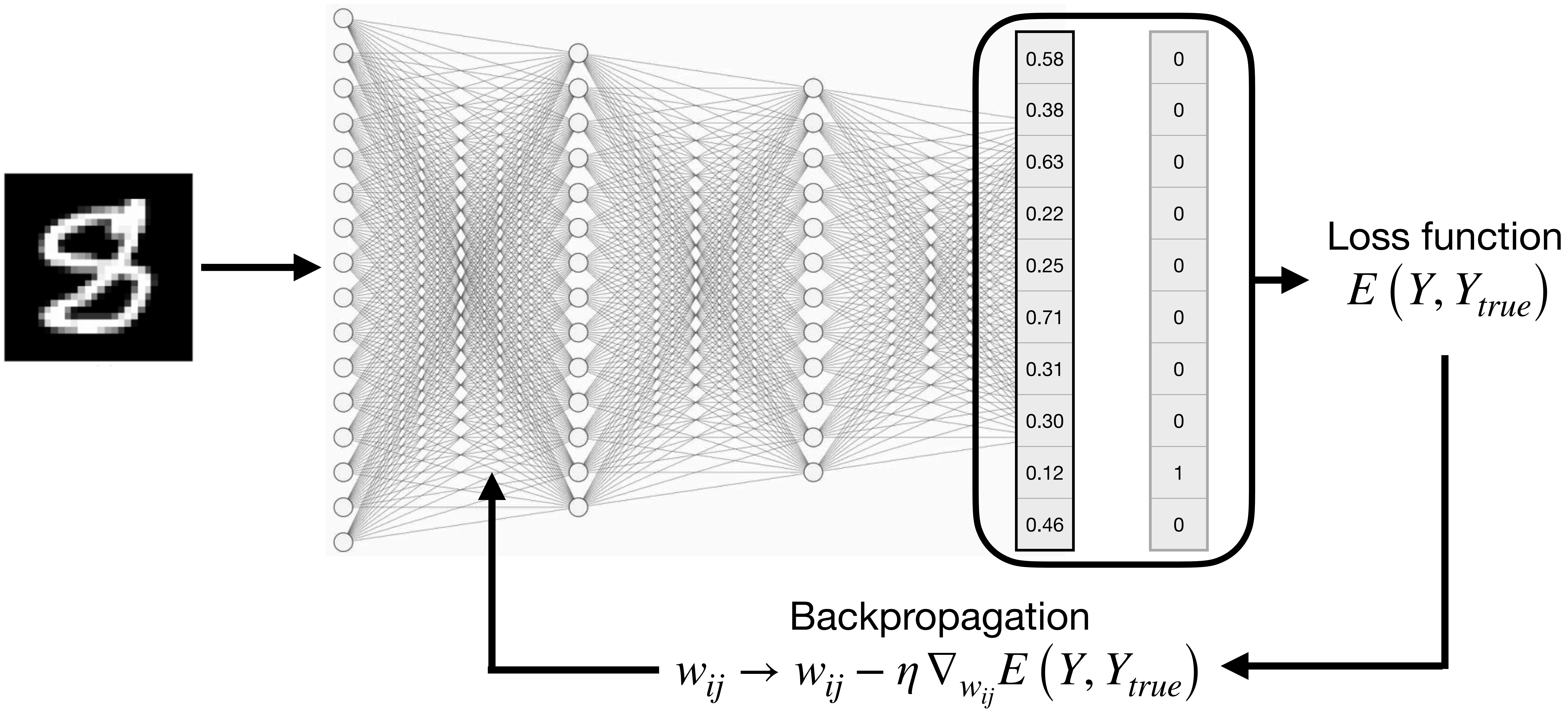
$$\ldots n+1) = \sigma\left(\sum_{j=1}^{N} w_{ij}\, y_j^{(n)} + b_i^{(n)}\right)$$

Input layer X

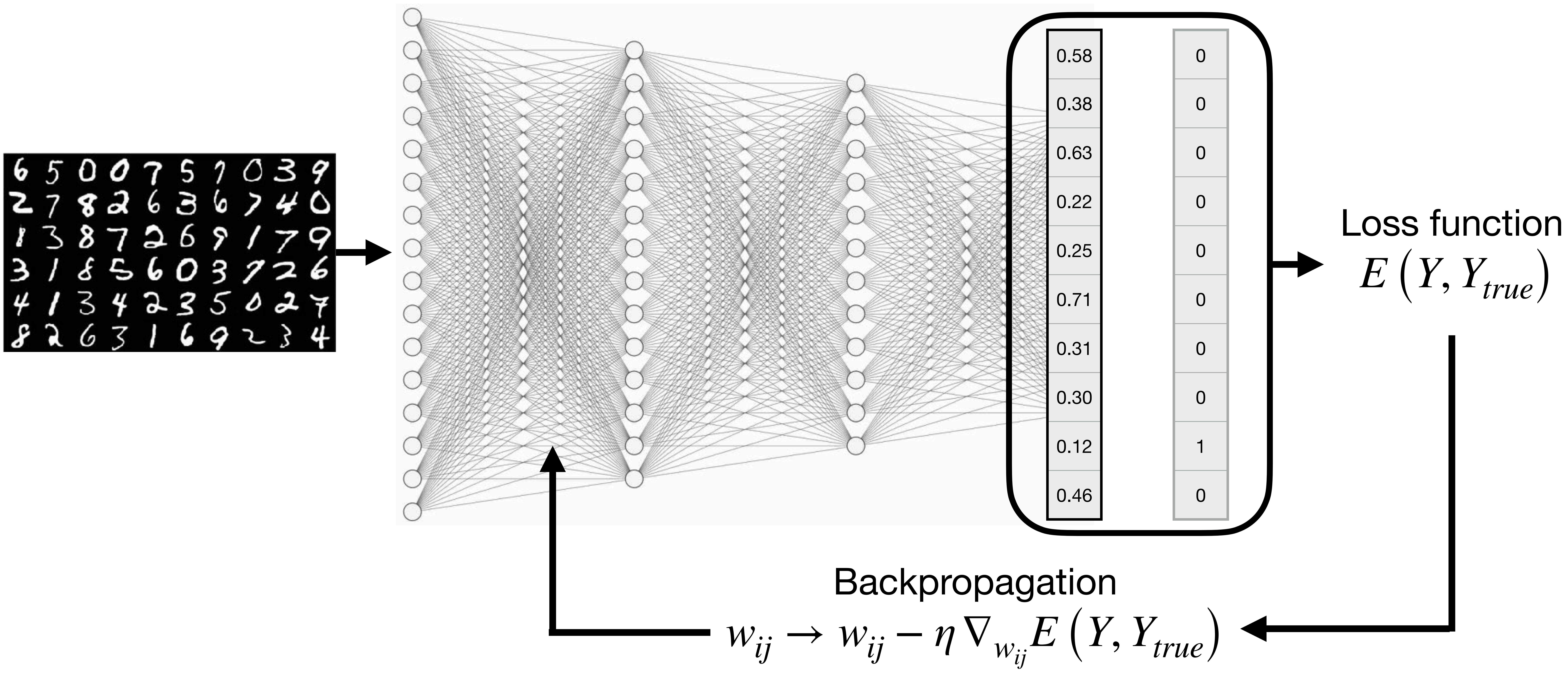
Hidden layers $f(X)$

Output layer Y

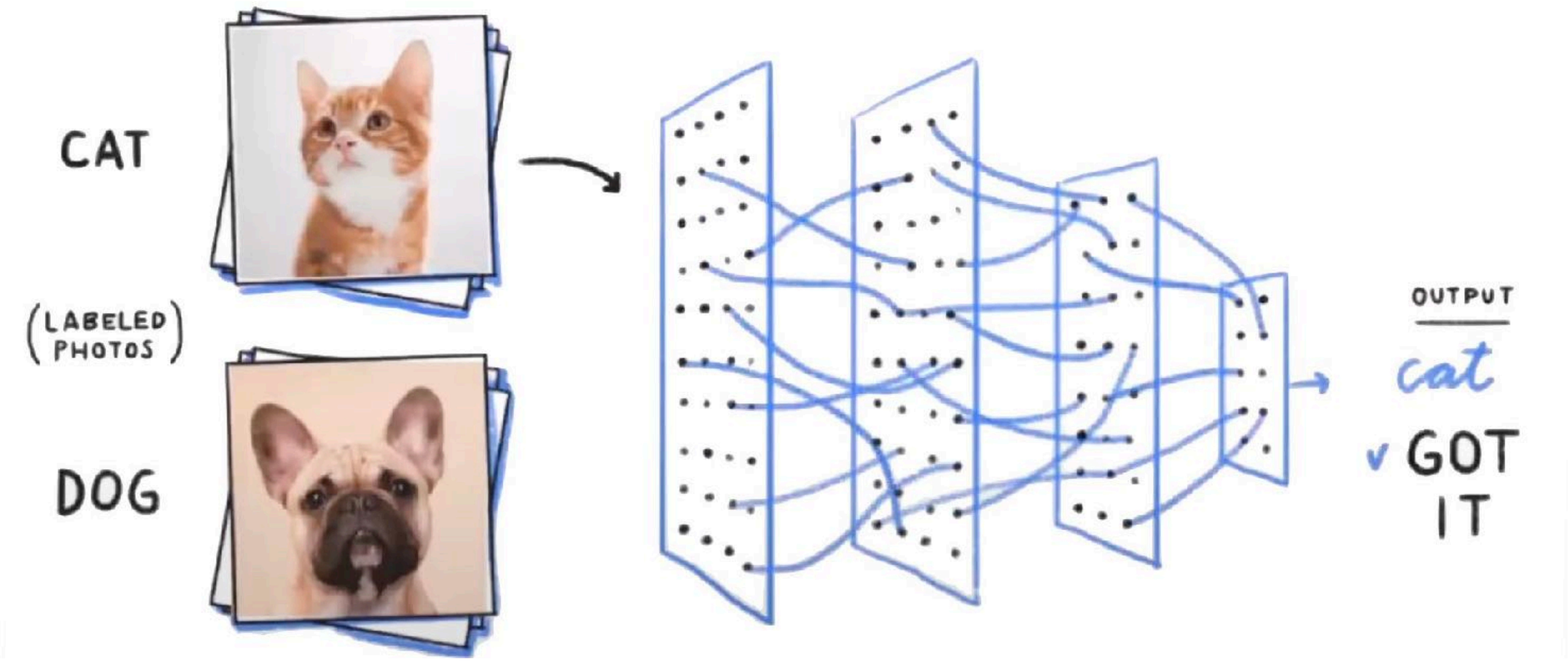
Neural Networks



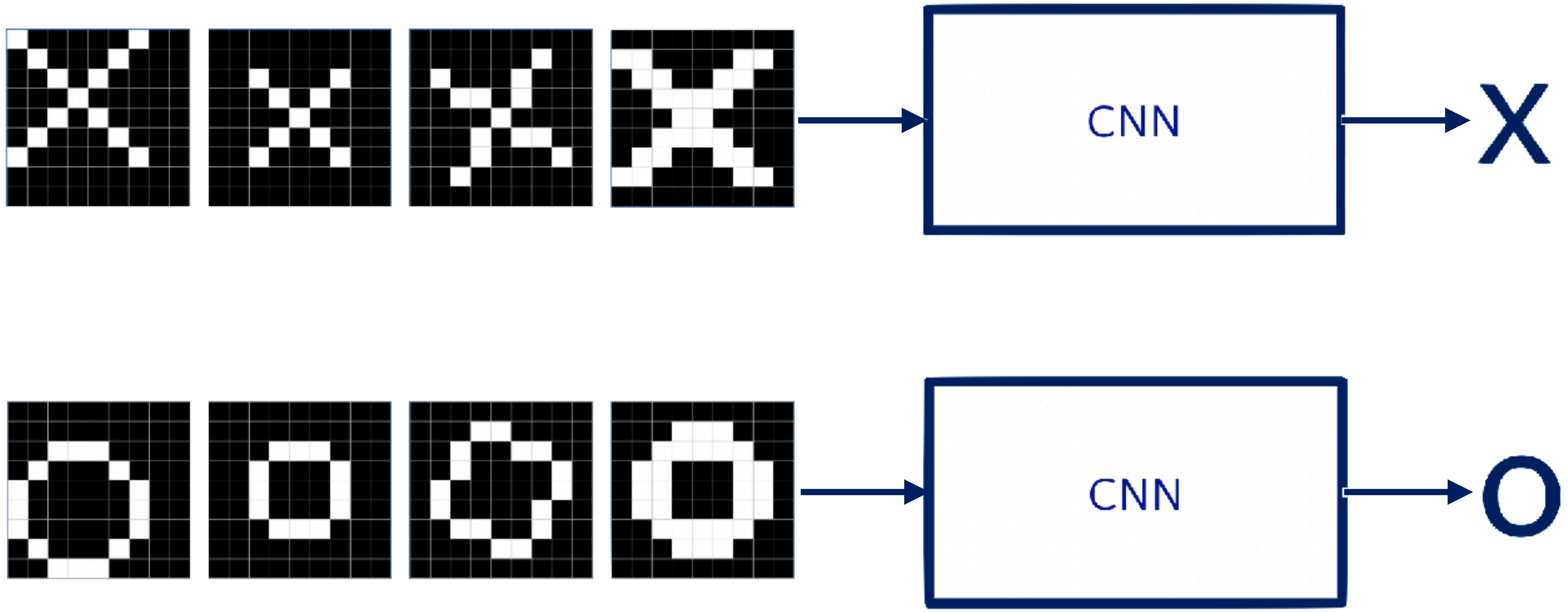
Neural Networks



Convolutional Neural Networks

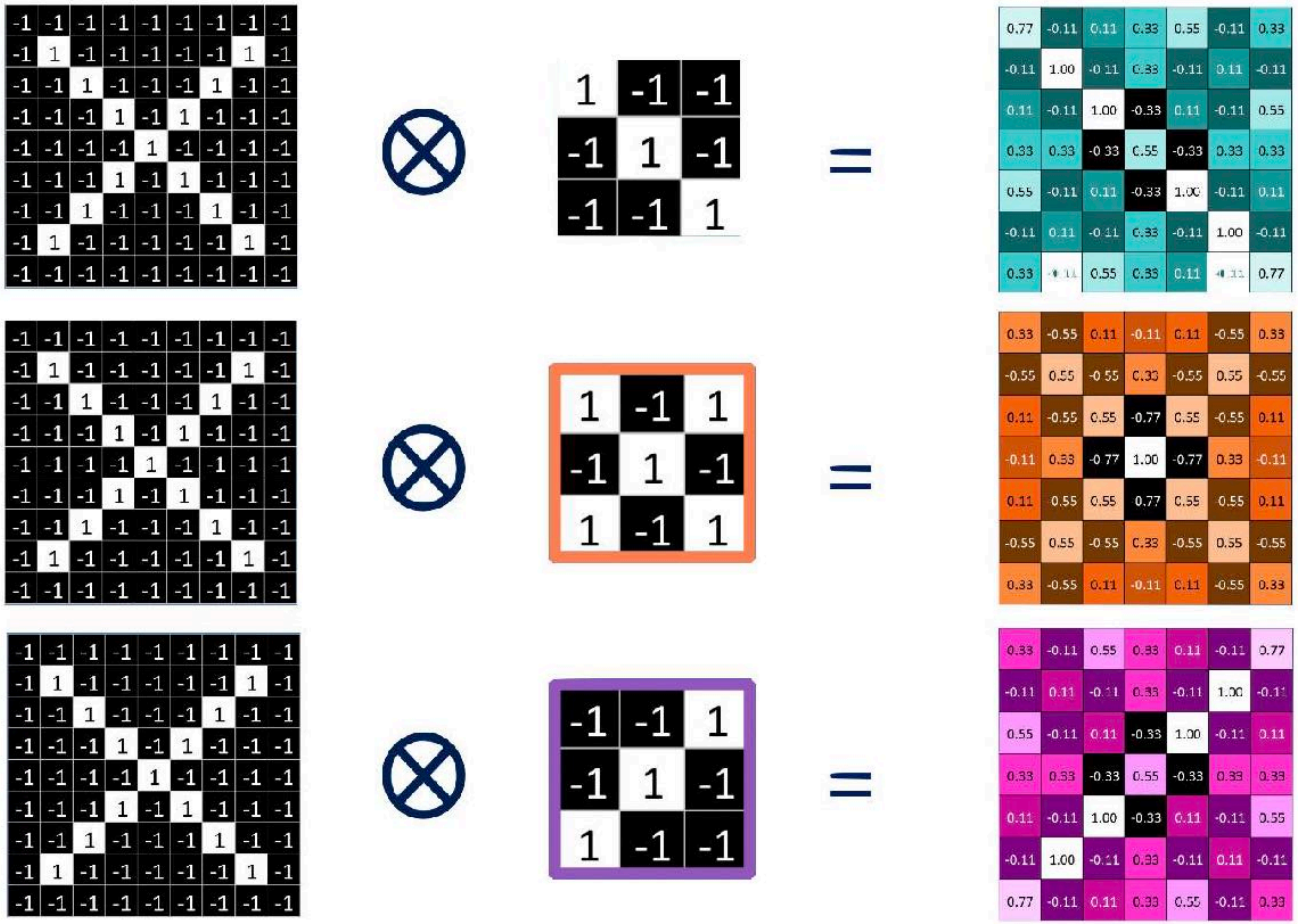


Convolutional Neural Networks



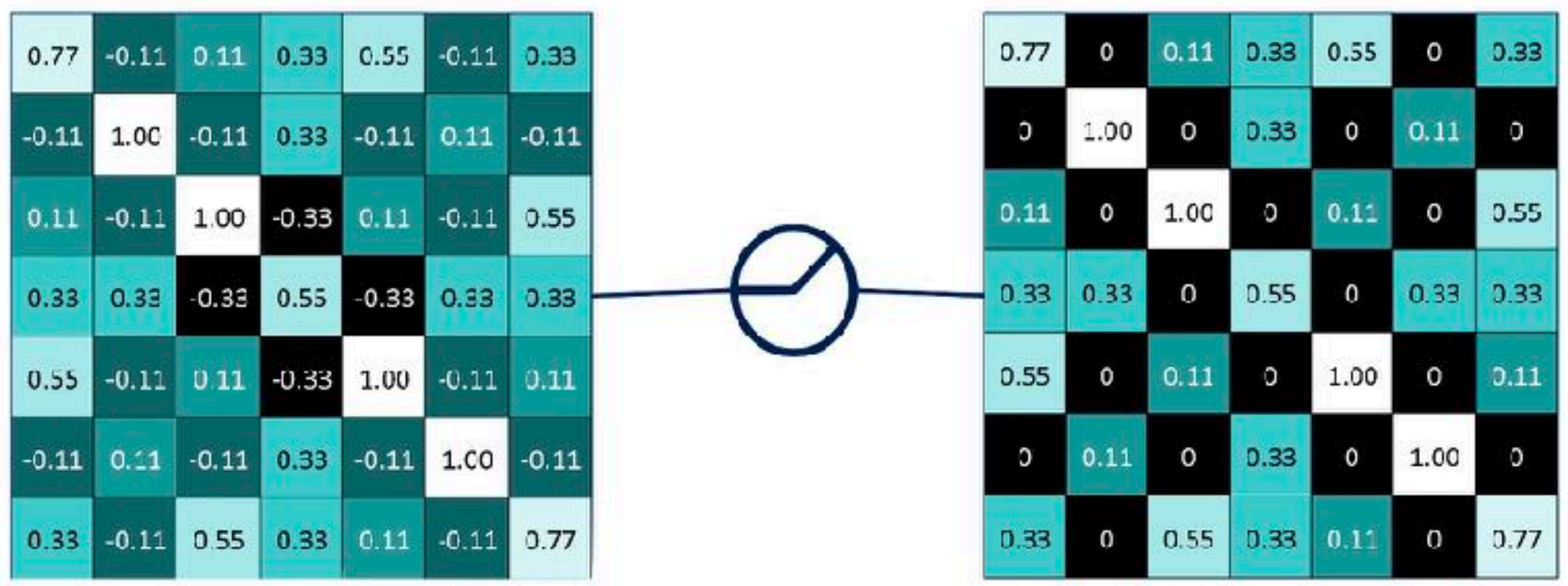
Example taken from https://e2eml.school/blog.html

Convolution layer

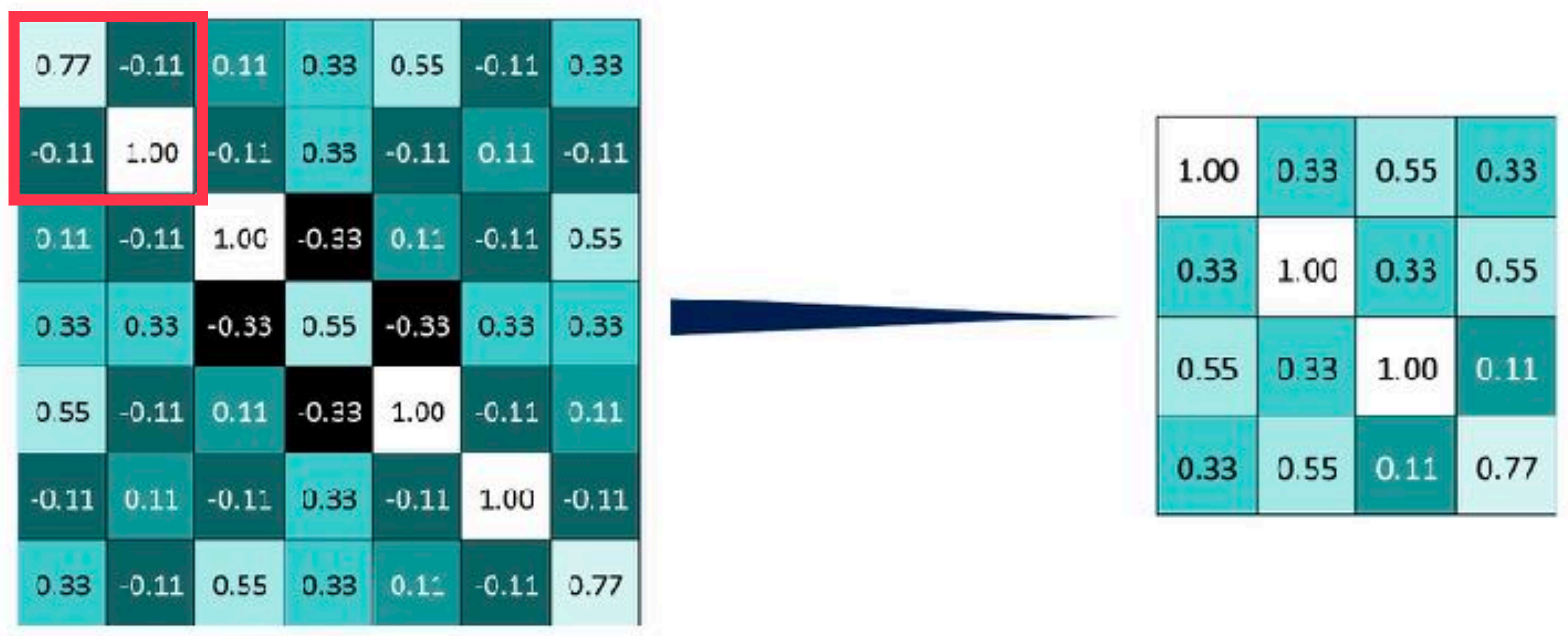


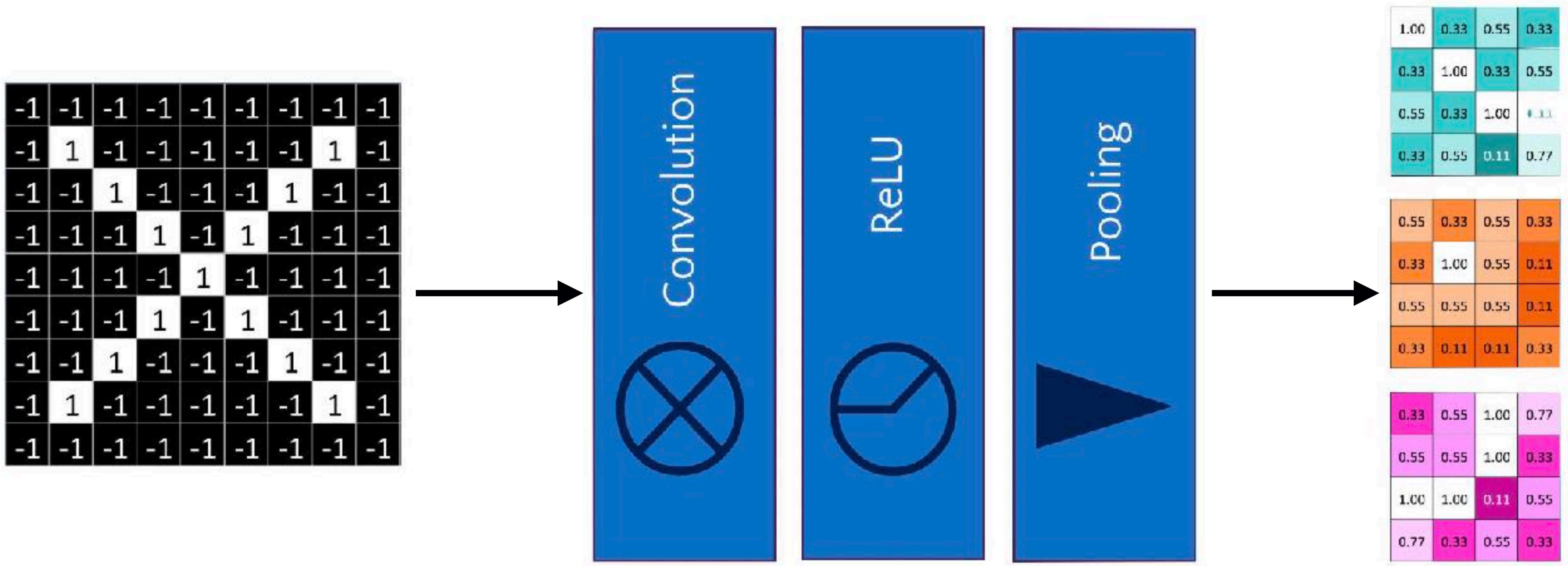
From https://e2eml.school/blog.html

ReLU

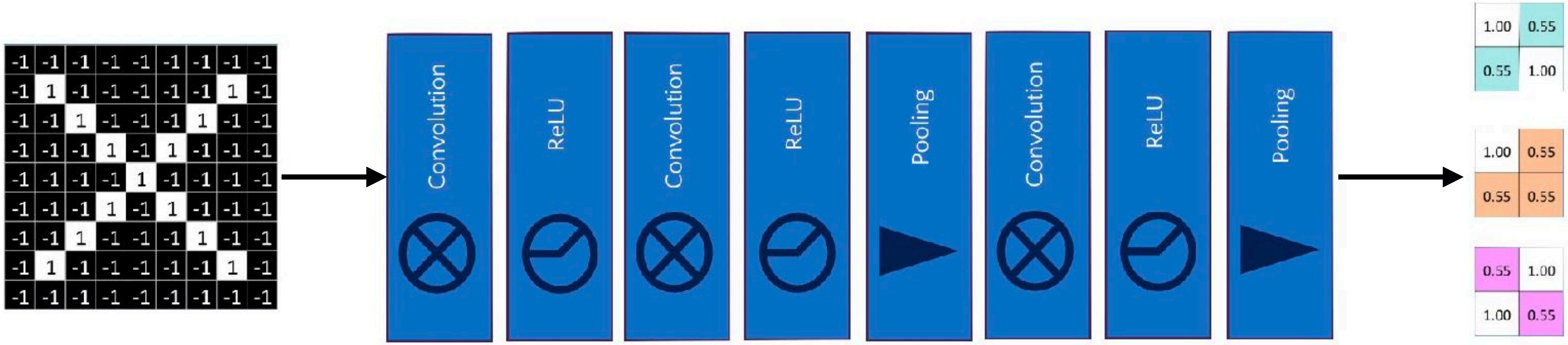


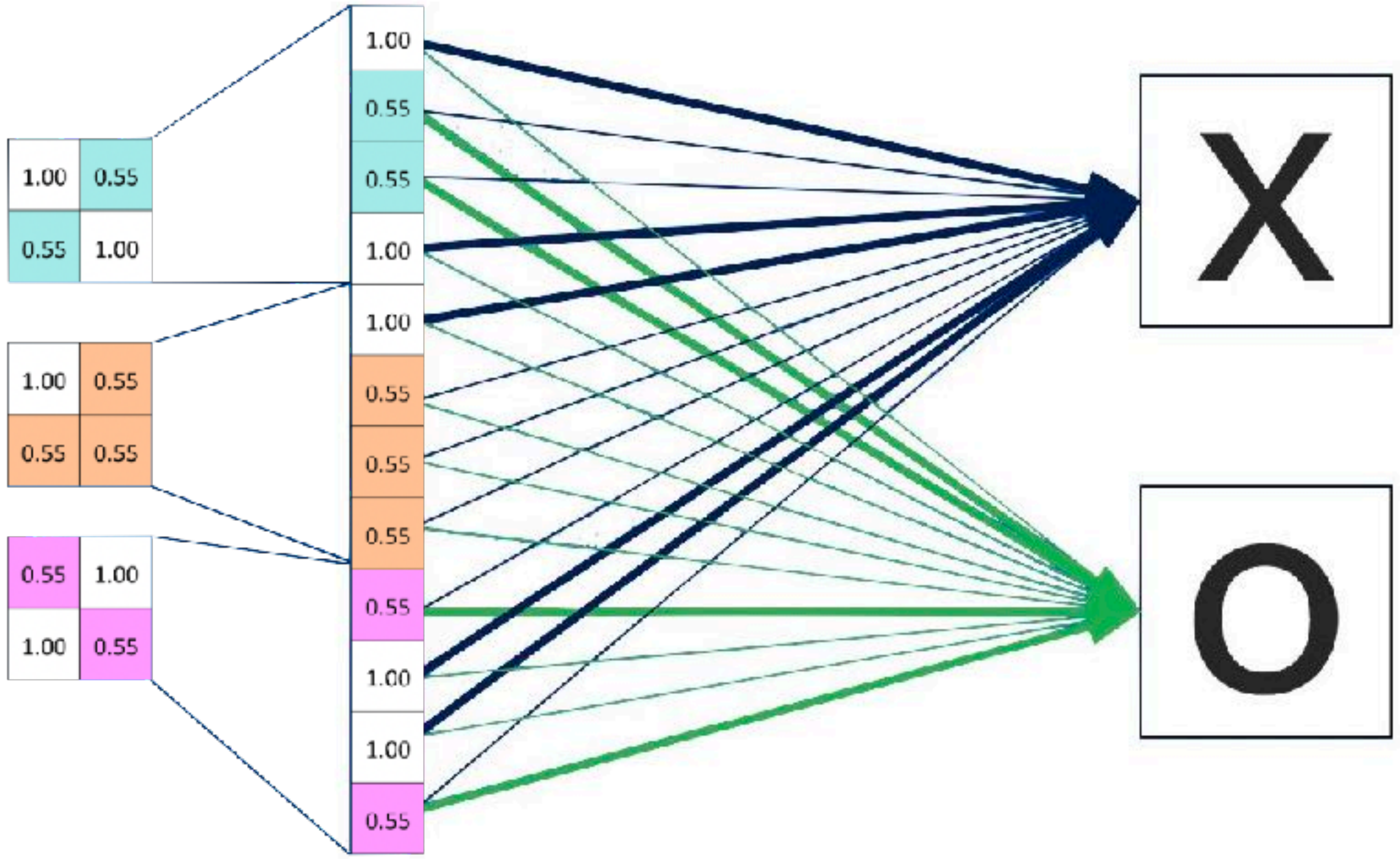
Max pooling





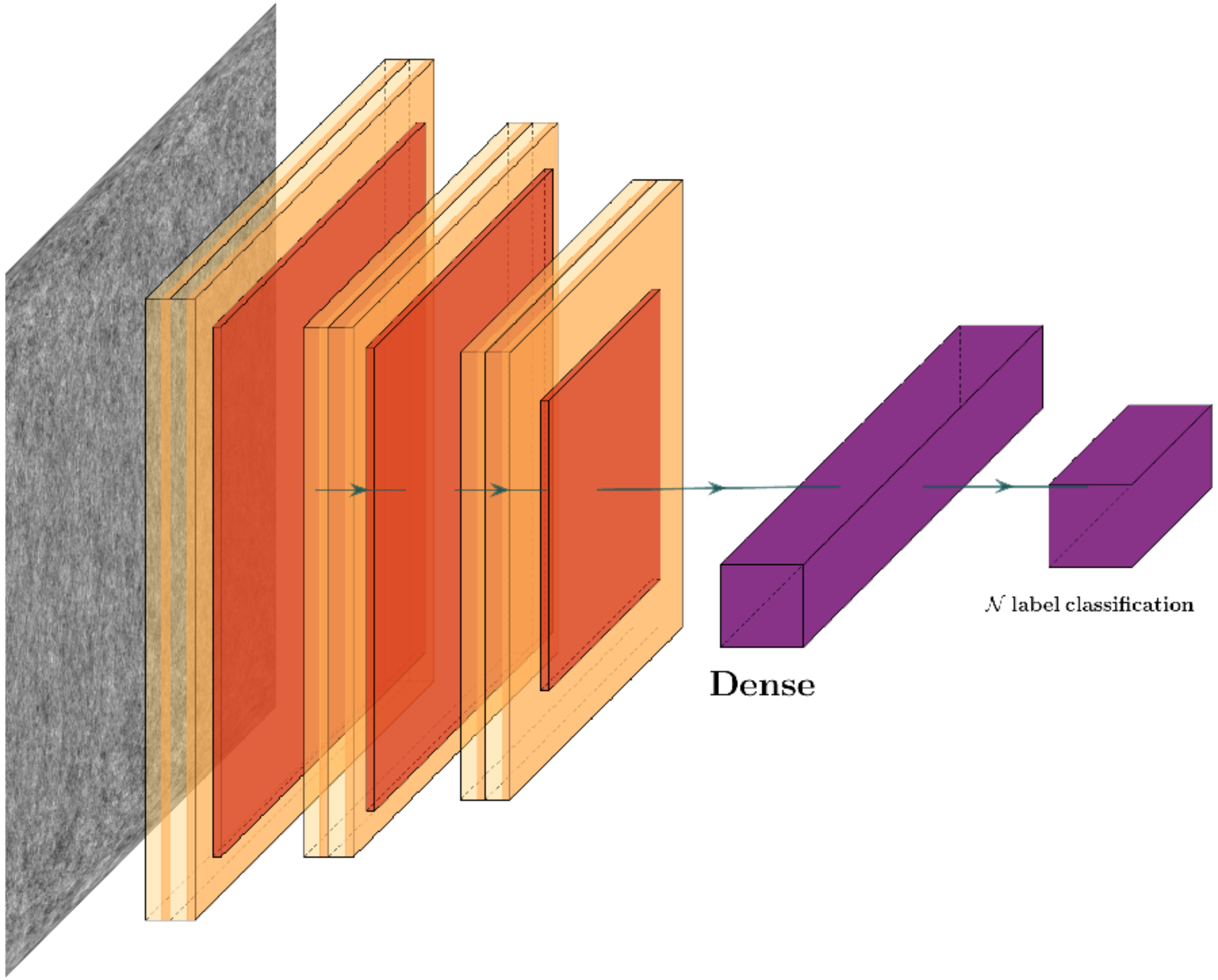
From https://e2eml.school/blog.html



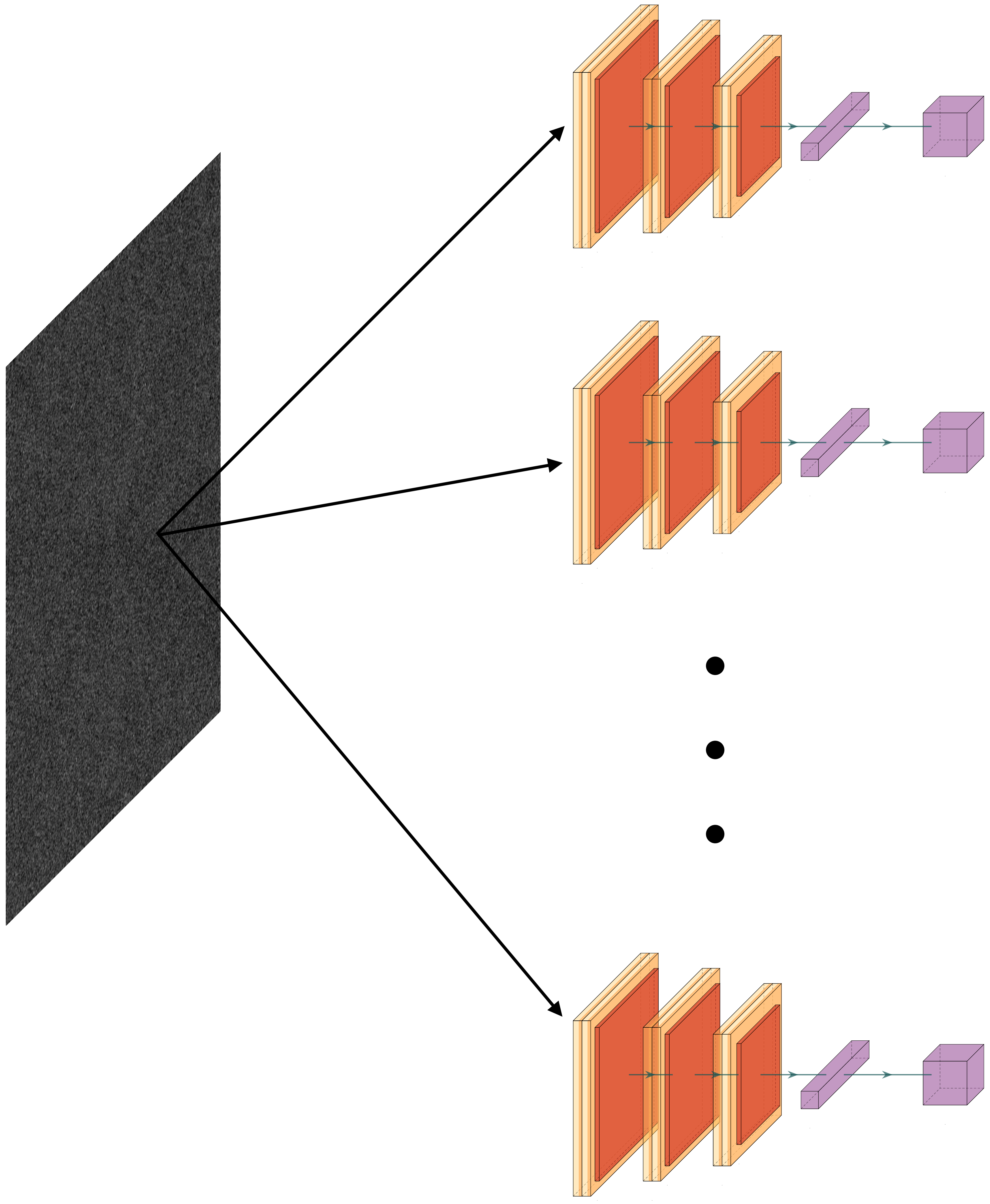


From https://e2eml.school/blog.html

CNN architecture (1st attempt)



CNN architecture (2nd attempt)



Training and testing

Our pipeline

1. Create training set

    A. Start from T-RECS simulated catalogue of AGN and SFG (Bonaldi et al., 2019, MNRAS, 482, 2)

    B. Modify number of sources as a function of flux N(s)

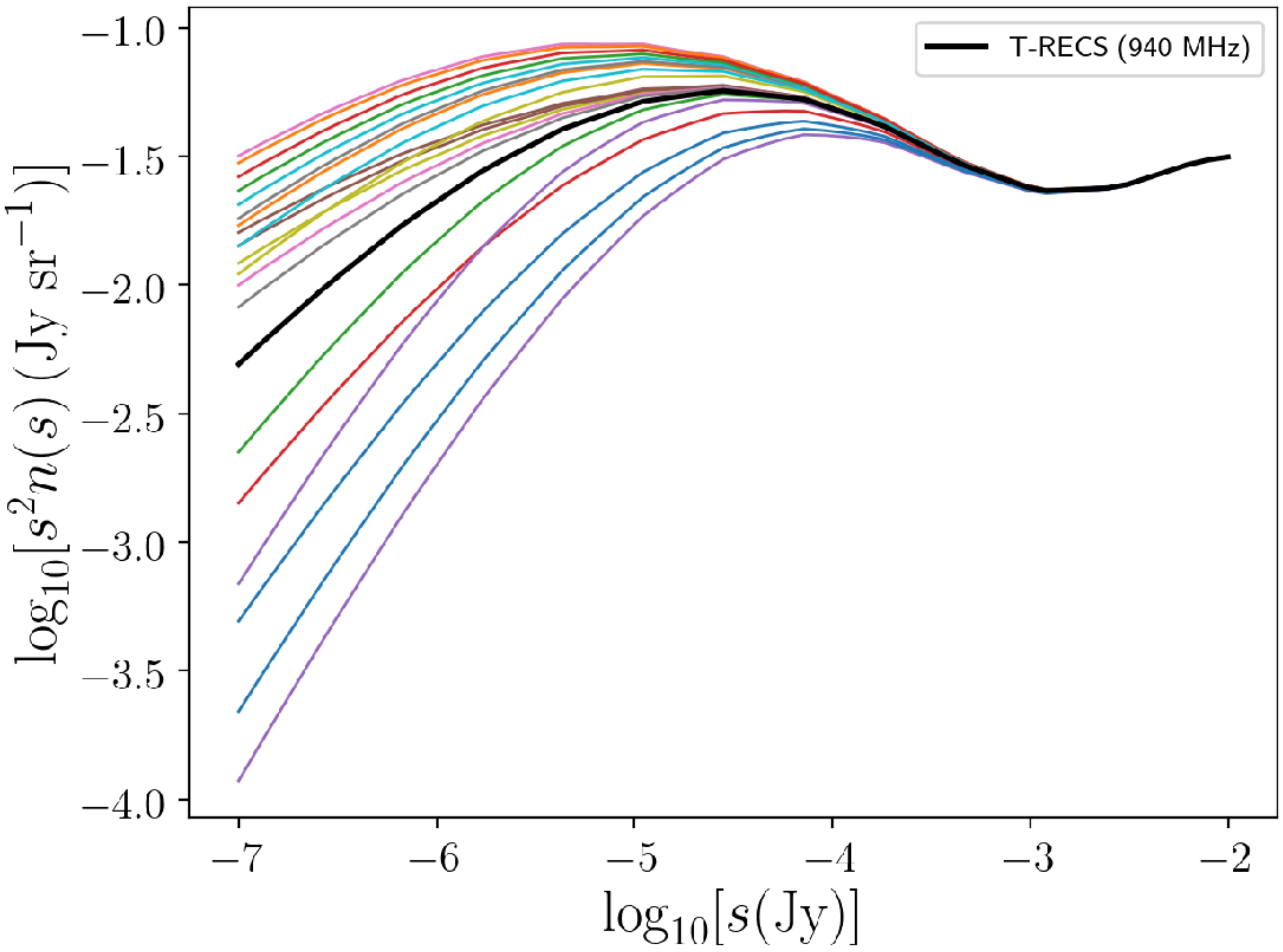
    C. Create images of the sky with ASKAPsoft (Yandasoft) (Guzman J., et al., 2019, ascl:1912.003)

2. Train CNN to extract N(s) from images

3. Test CNN

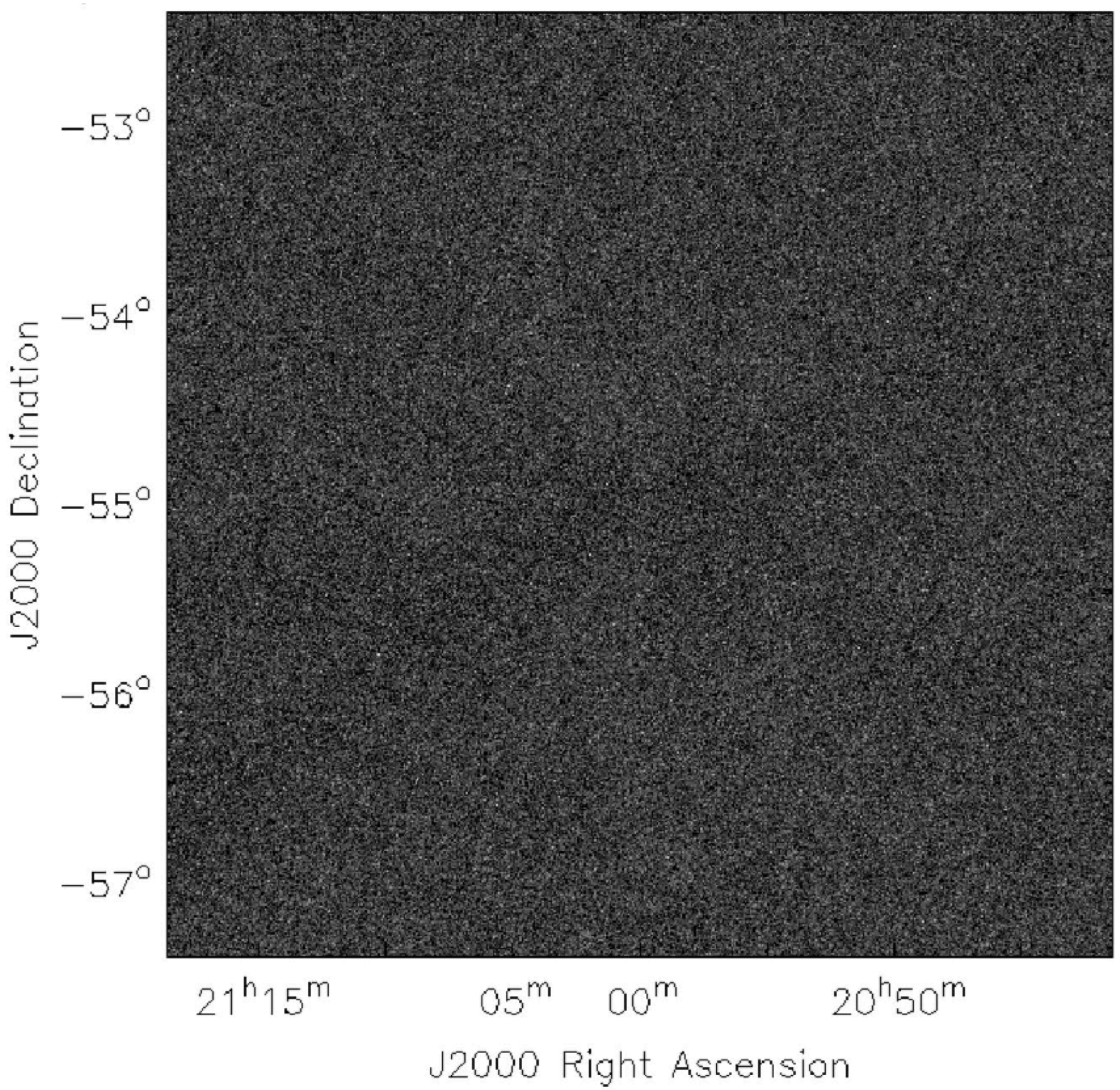
Training set

$$n(s) = \left(1 + \frac{s_0}{s}\right)^{\alpha} n_{T-RECS}(s)$$

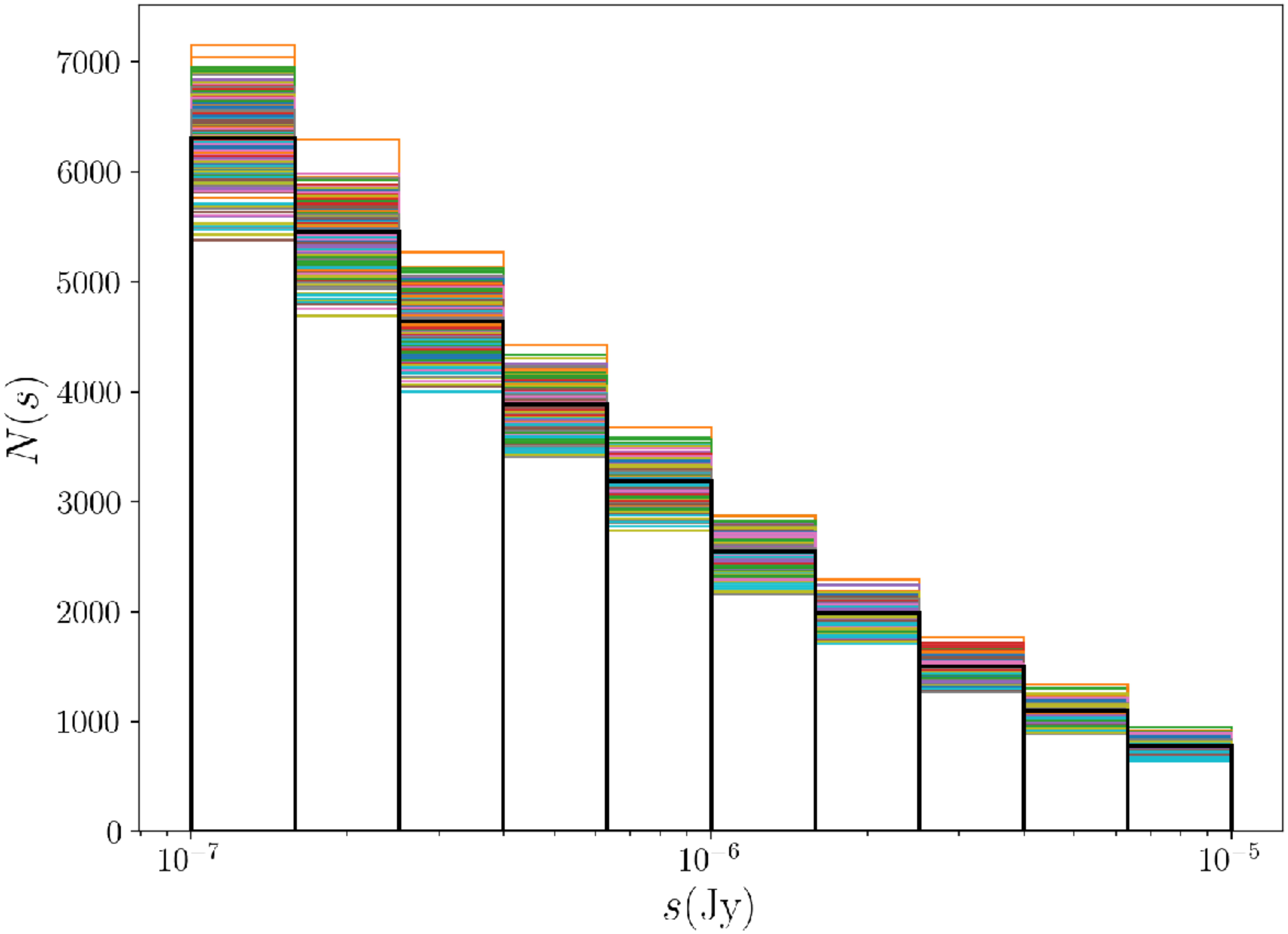


Training set

- 21 images
  - 25 deg^2, 2560x2560 pixels
- 400 sub-images for each image
  - 128x128 pixels
- Total 8400 sub-images
  - 6048 training
  - 1512 validation
  - 840 testing



One image

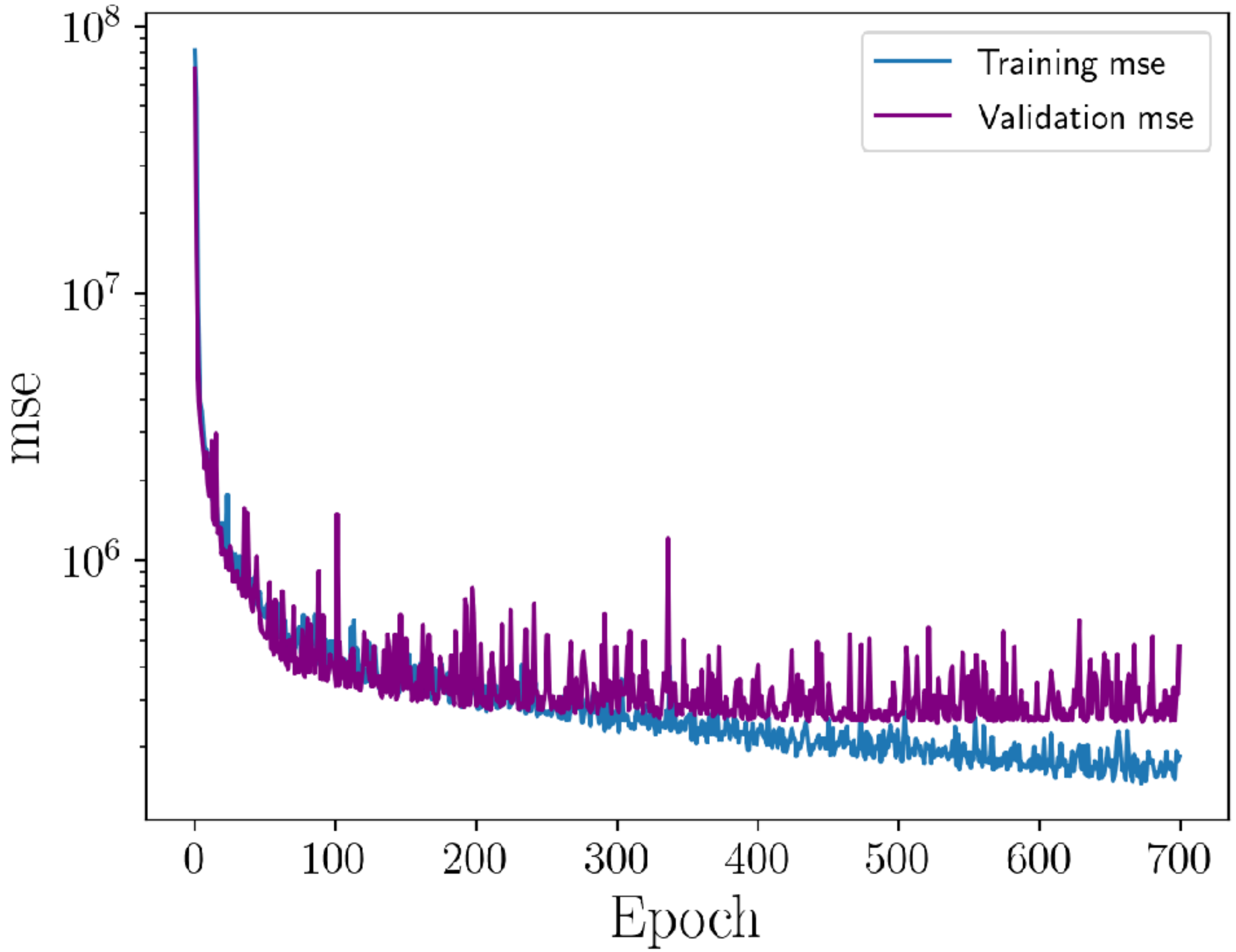


Train with sky model images

- No telescope effects

- Loss function

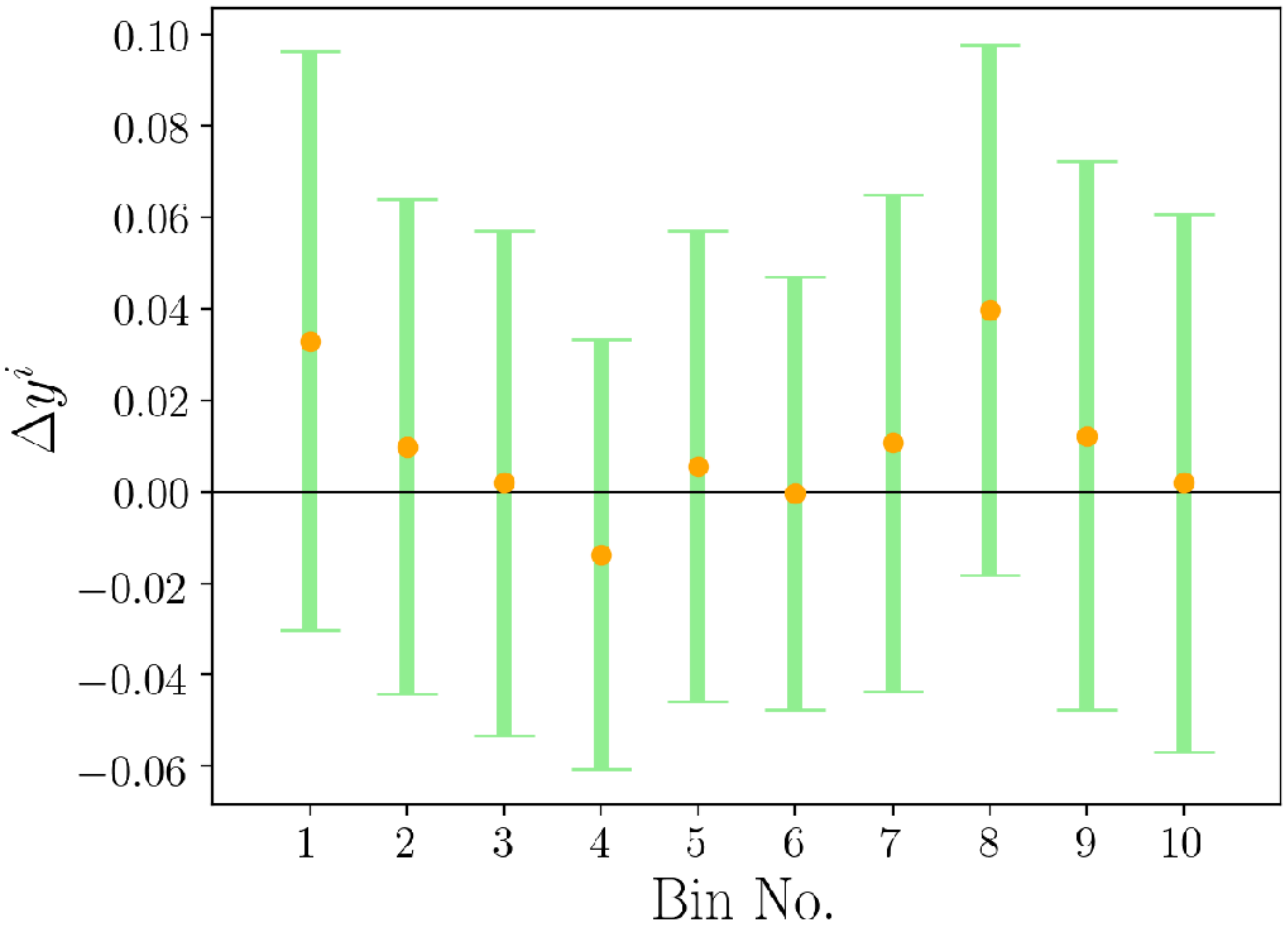
$$mse = \frac{1}{N_{labels}} \sum_{i=1}^{N_{labels}} \left(y^i_{true} - y^i_{prediction} \right)^2$$

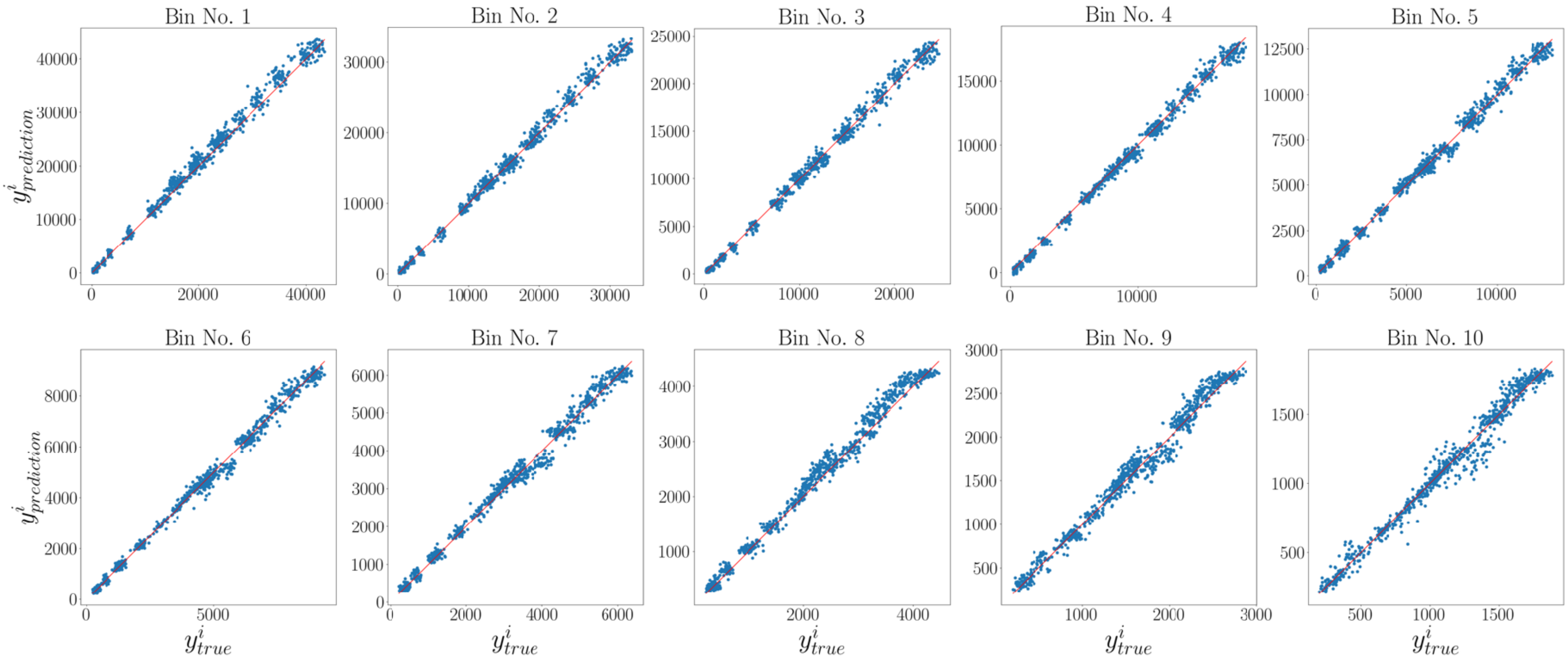


Train with sky model images

- Reconstruction residual

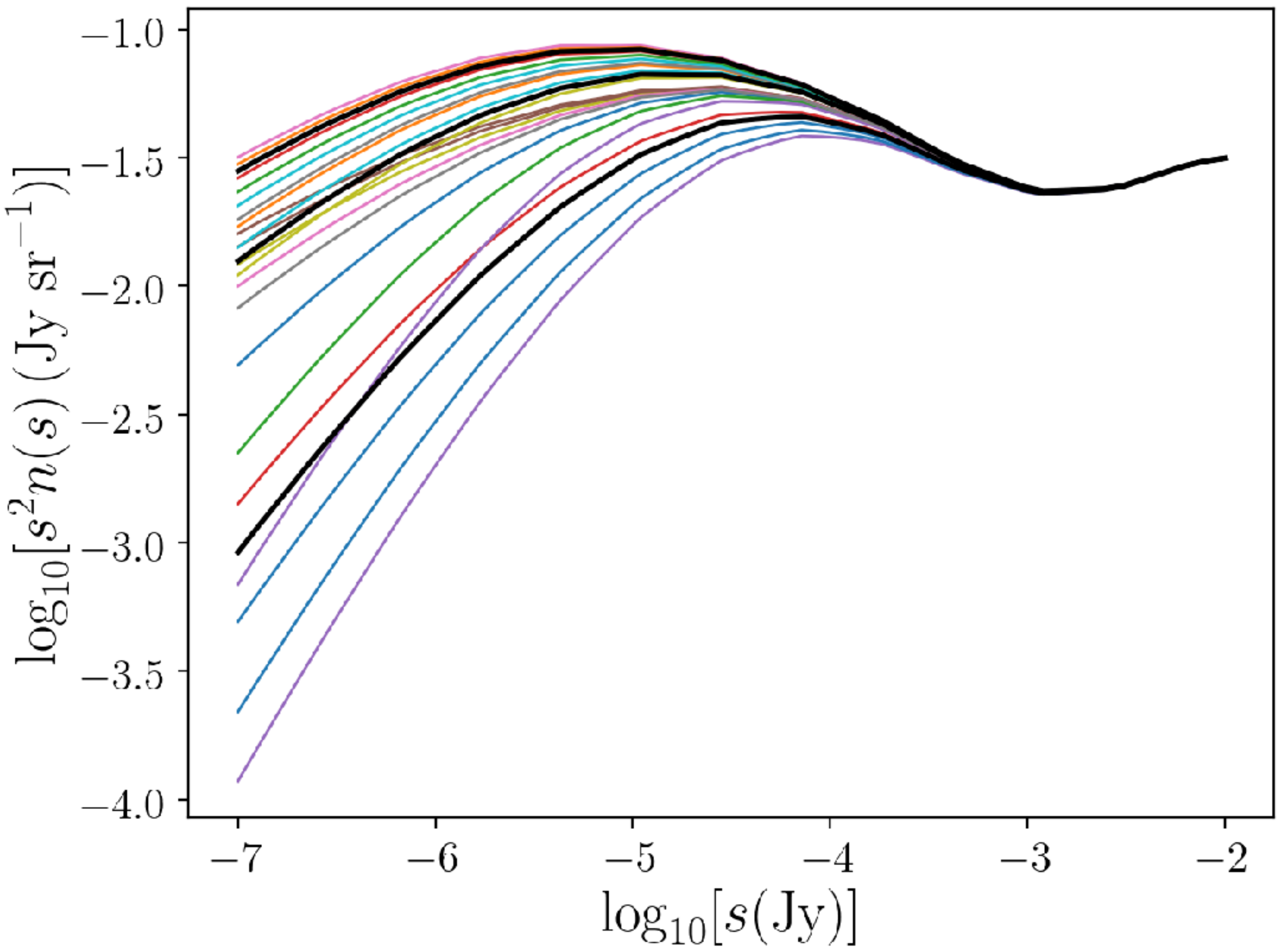
$$\Delta y^i = \frac{y^i_{prediction} - y^i_{true}}{\frac{1}{N_{labels}} \sum_{i=1}^{N_{labels}} y^i_{true}}$$

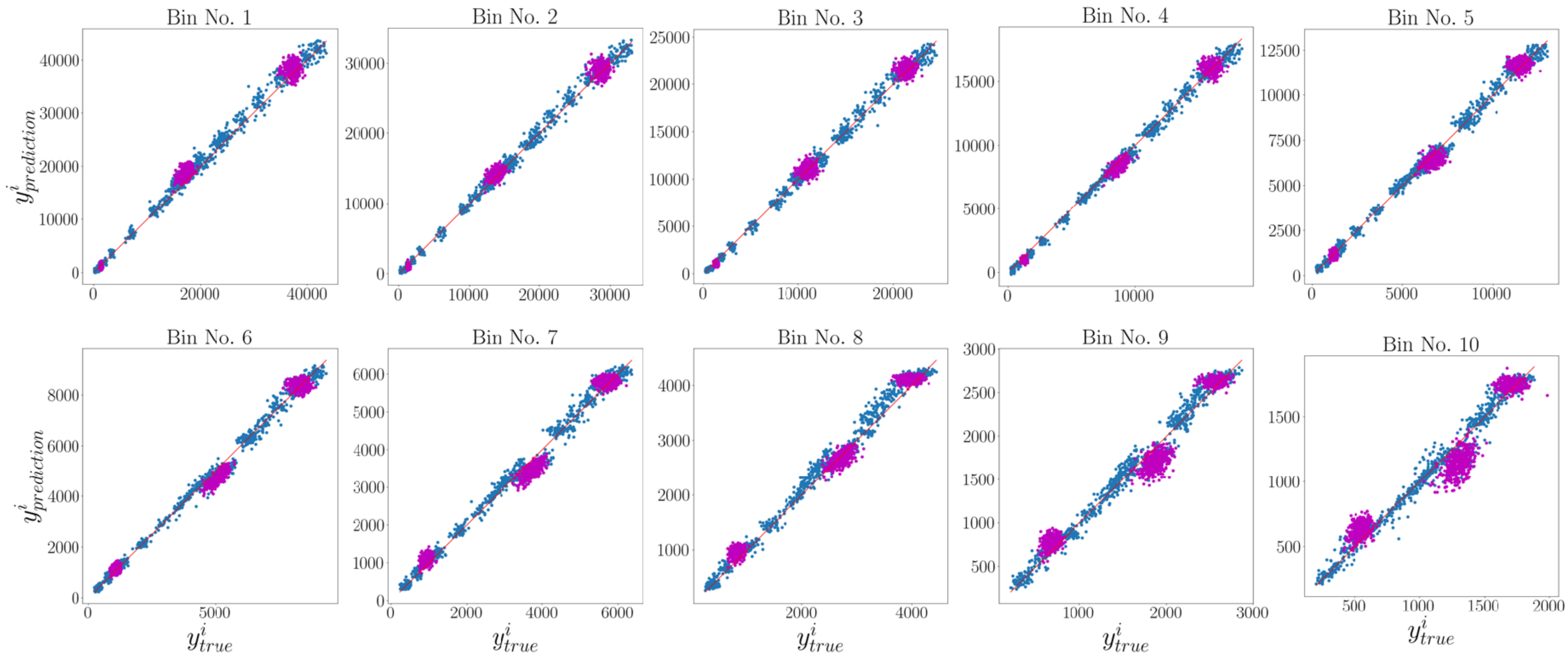




Bin No. 1
Bin No. 2
Bin No. 3
Bin No. 4
Bin No. 5
Bin No. 6
Bin No. 7
Bin No. 8
Bin No. 9
Bin No. 10
$y^i_{prediction}$
y^i_{true}

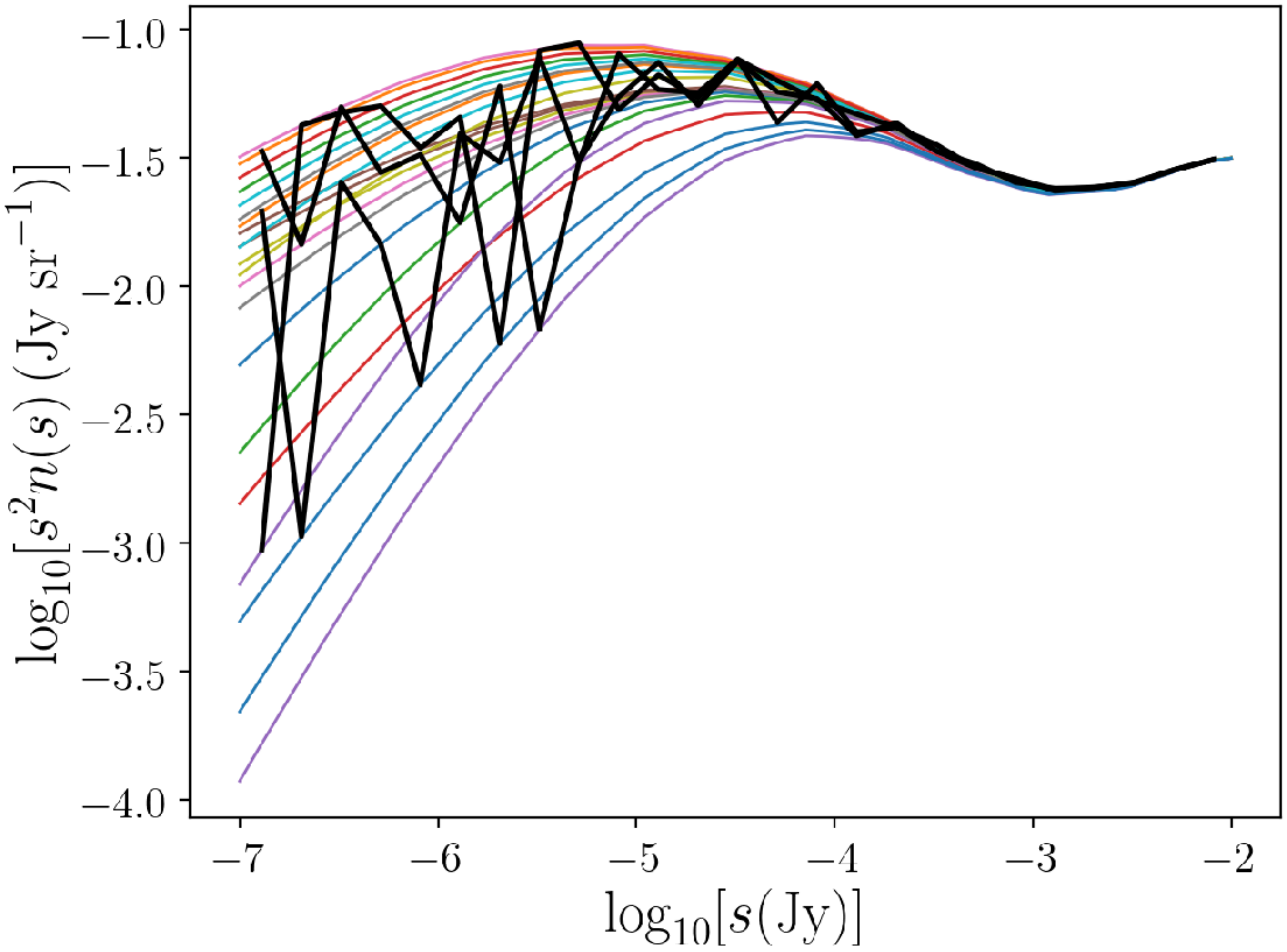
Test: same shape, different parameter values

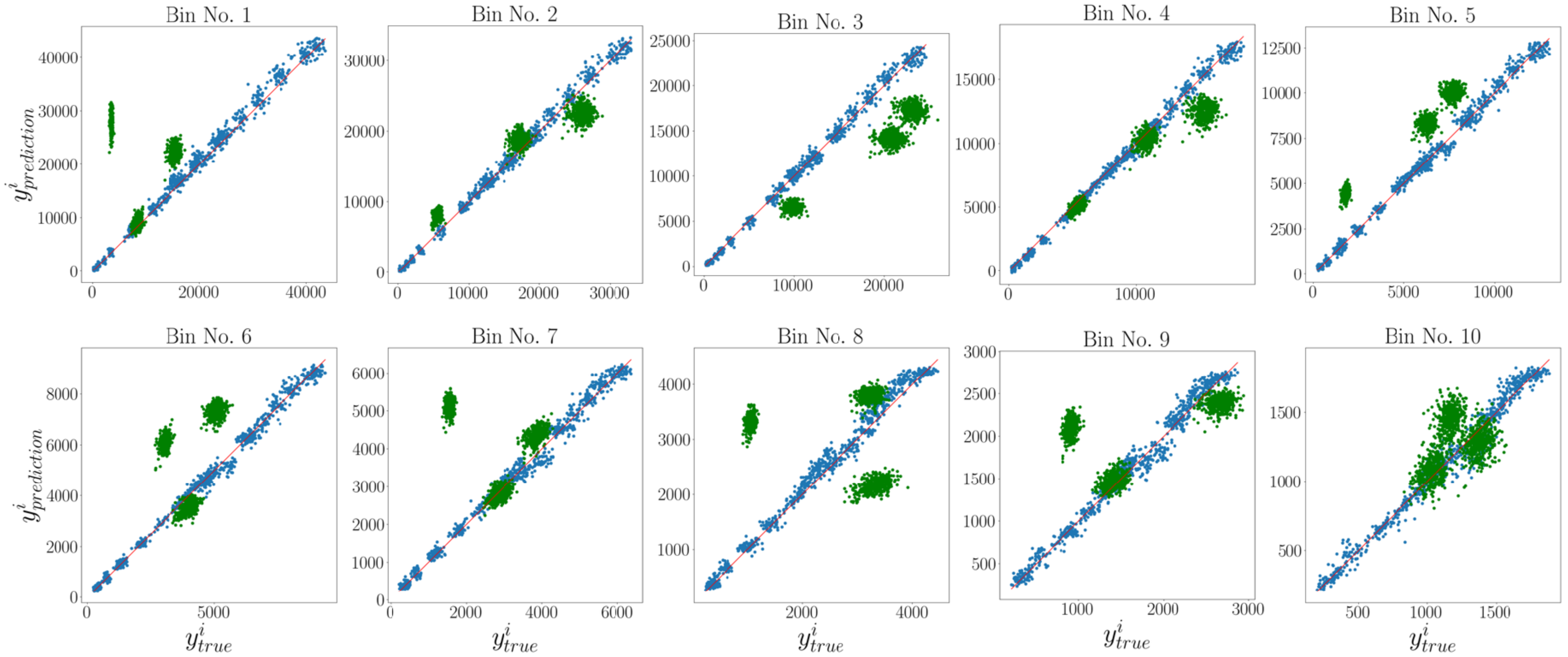




Bin No. 1
Bin No. 2
Bin No. 3
Bin No. 4
Bin No. 5
Bin No. 6
Bin No. 7
Bin No. 8
Bin No. 9
Bin No. 10
$y^i_{prediction}$
y^i_{true}

Test: same shape, different parameters





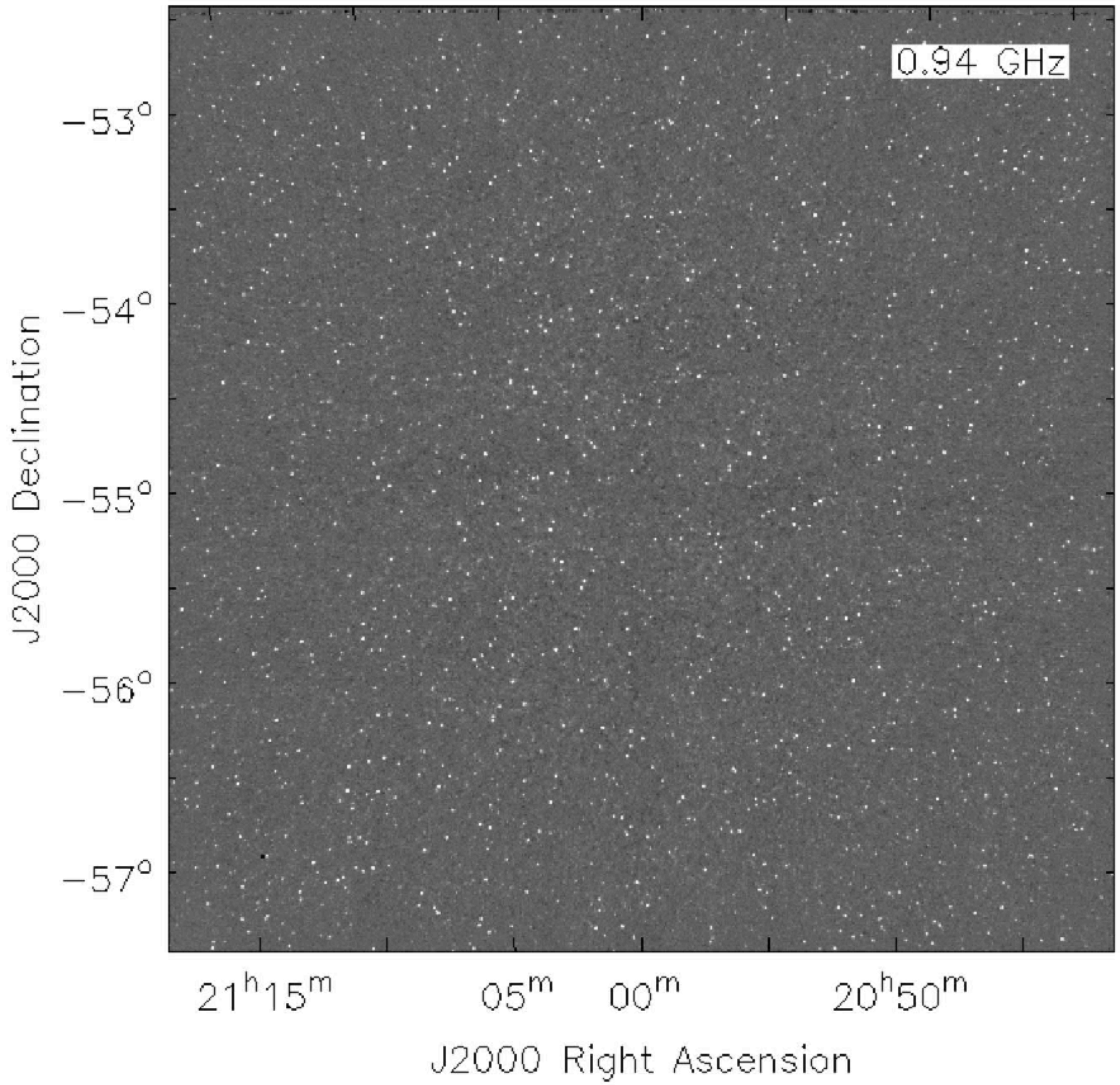
Bin No. 1
Bin No. 2
Bin No. 3
Bin No. 4
Bin No. 5
Bin No. 6
Bin No. 7
Bin No. 8
Bin No. 9
Bin No. 10
$y^i_{prediction}$
y^i_{true}

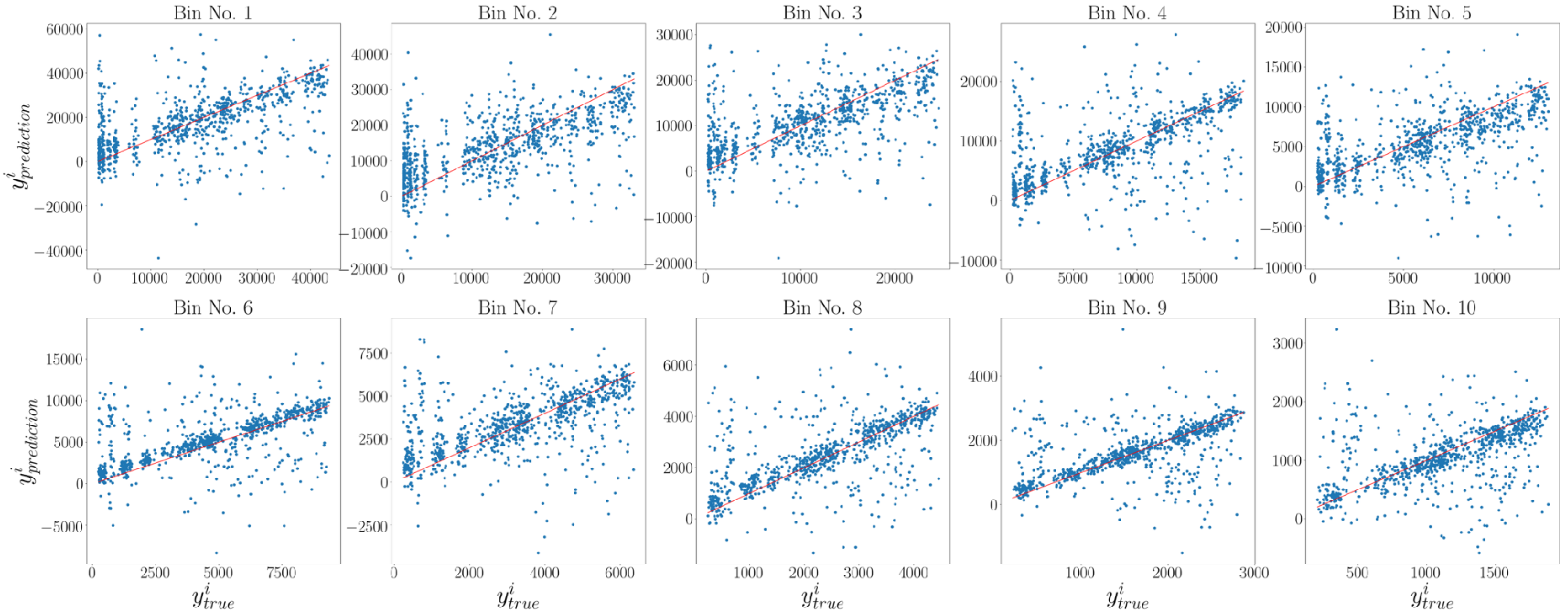
How to decorrelate the bins?

- Network learns the functional shape of N(s)
- Even training one separate network for each bin does not solve the problem
- What we'll try next:
  - Train with different functional shapes
  - Denser training data in each bin

Train with restored images

- Simulation of visibilities starting from sky model with ASKAPsoft
- CLEAN image
- Images include
  - noise
  - convolution with PSF gives side lobes





Bin No. 1
Bin No. 2
Bin No. 3
Bin No. 4
Bin No. 5
Bin No. 6
Bin No. 7
Bin No. 8
Bin No. 9
Bin No. 10
$y^i_{prediction}$
y^i_{true}

Avenues to improvement

- Find better ASKAPsoft parameters to improve cleaning
- Data pre-processing
  - Normalization and stretching
  - Data augmentation
- Give network auxiliary information (eg. total brightness)
- CNN hyperparameters
- CNN architecture

Thank you!