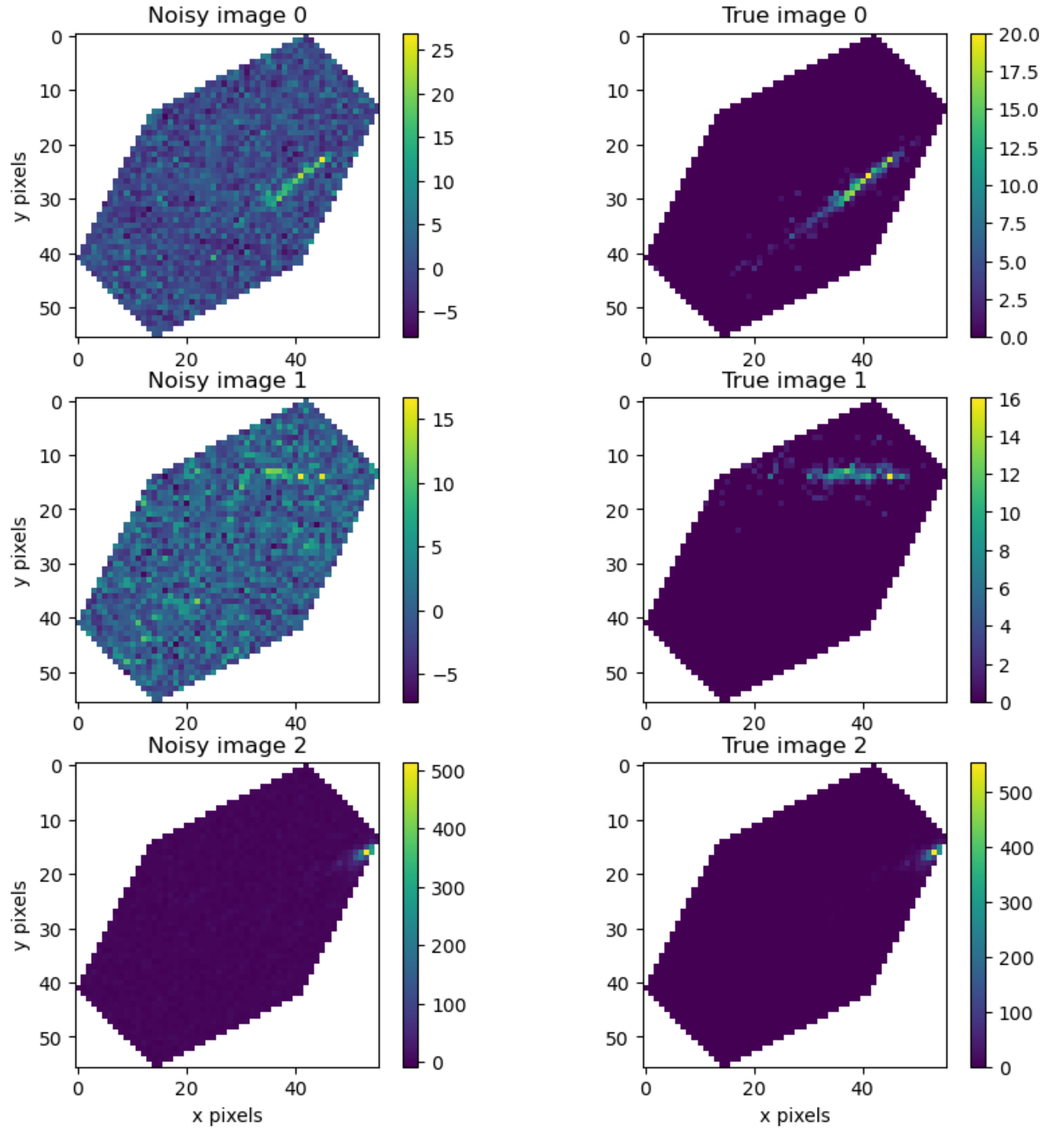
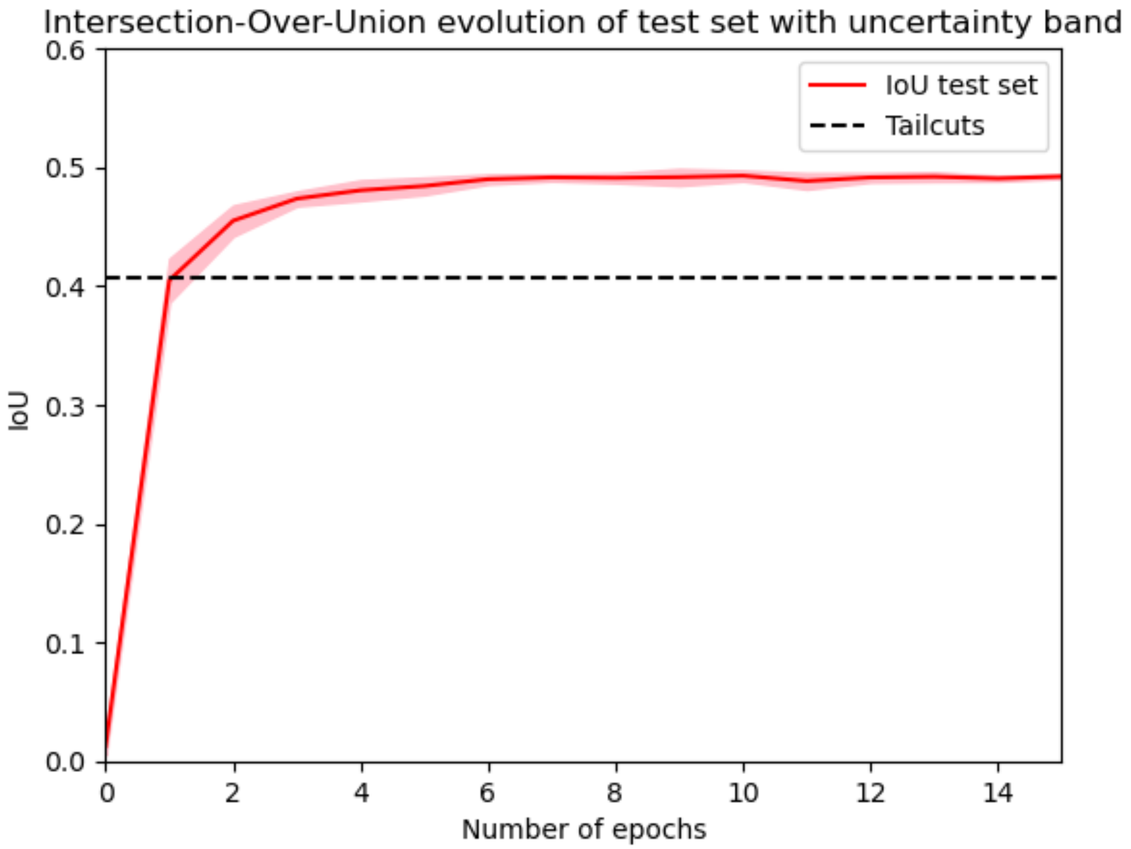
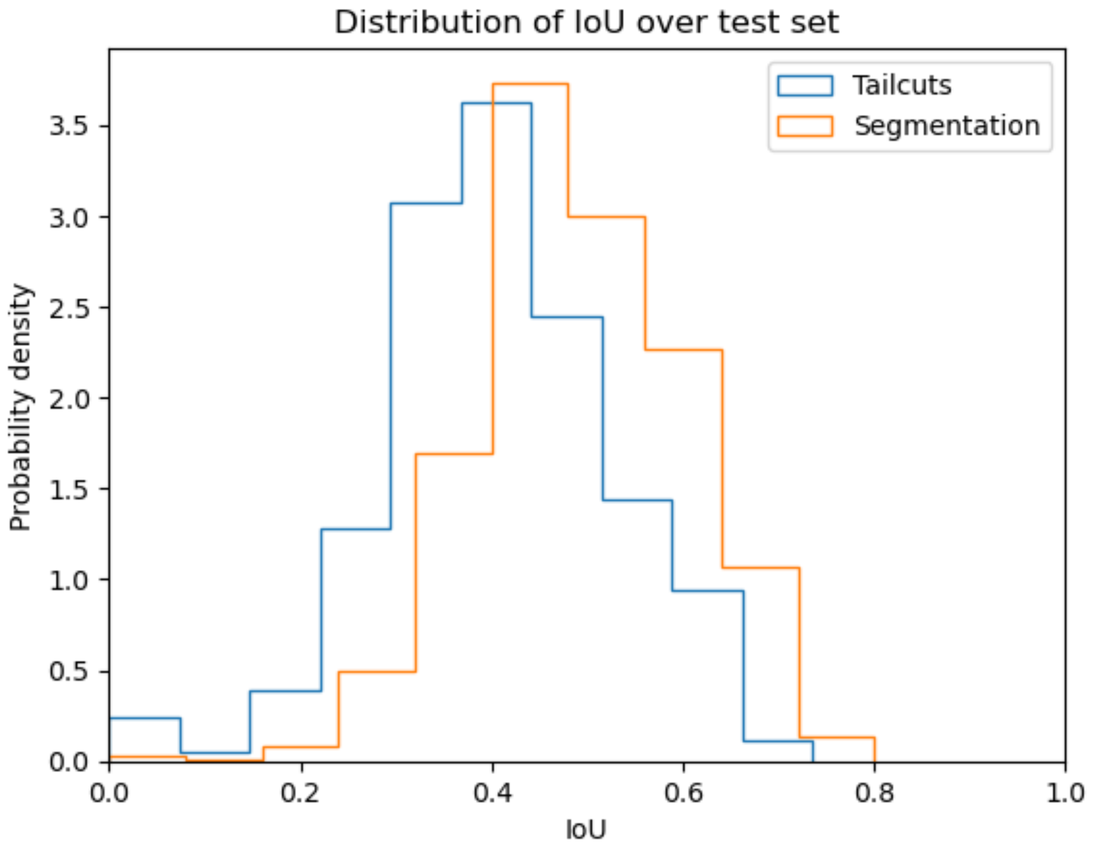
CTA MST Image Cleaning using ML



Segmentation

Best auto-encoder model

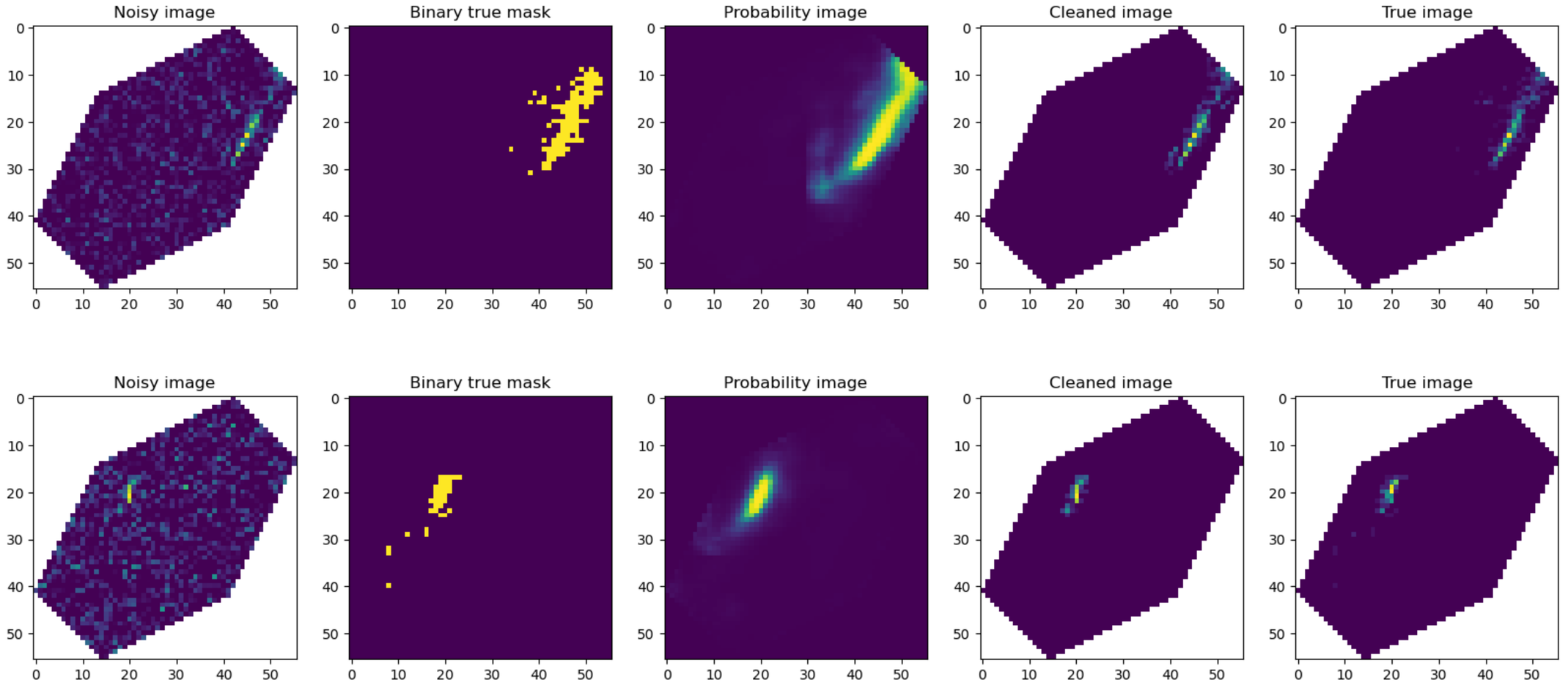




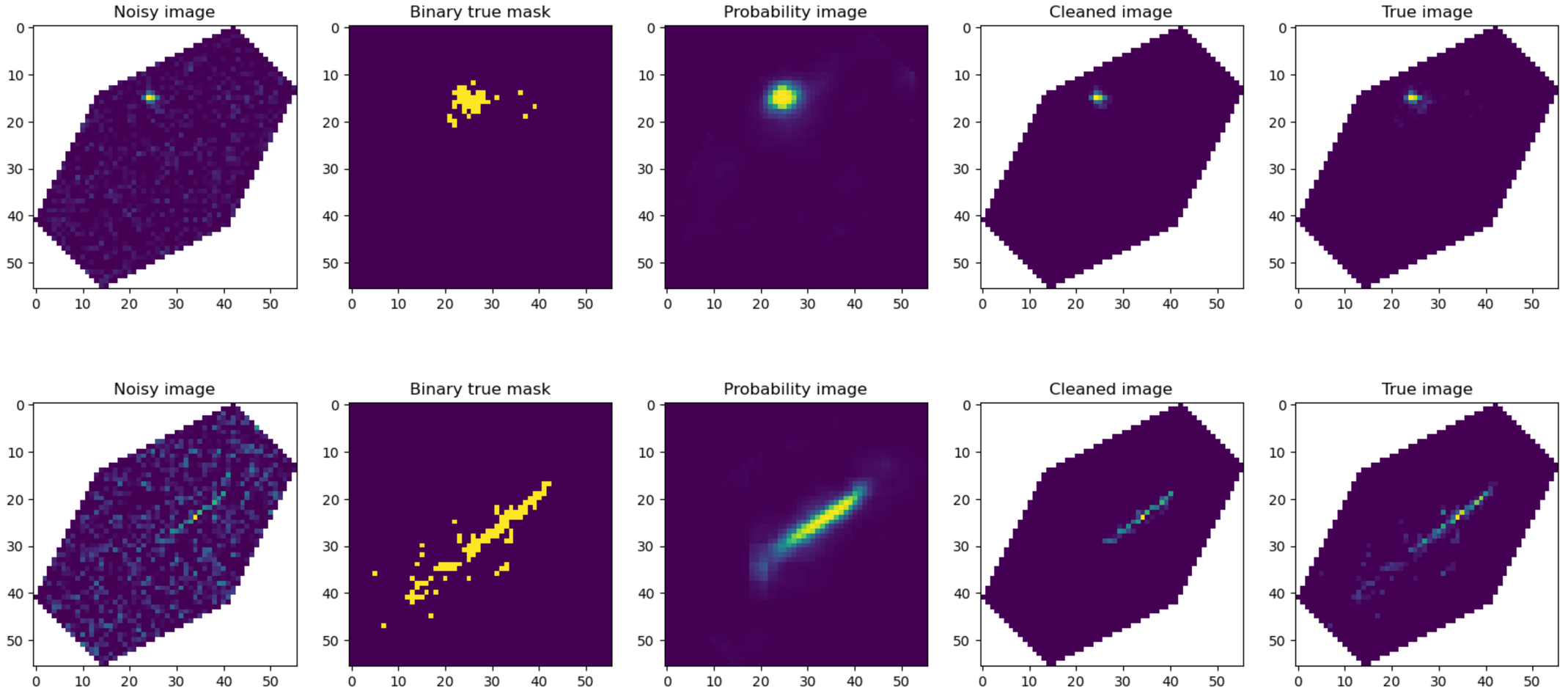
IoU tailcuts = 0.41

IoU segmentation = 0.49

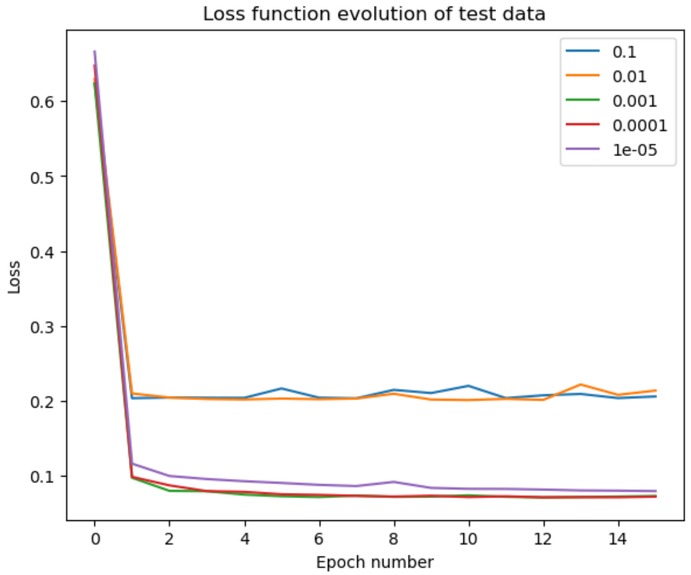
Some cleaning examples (I)

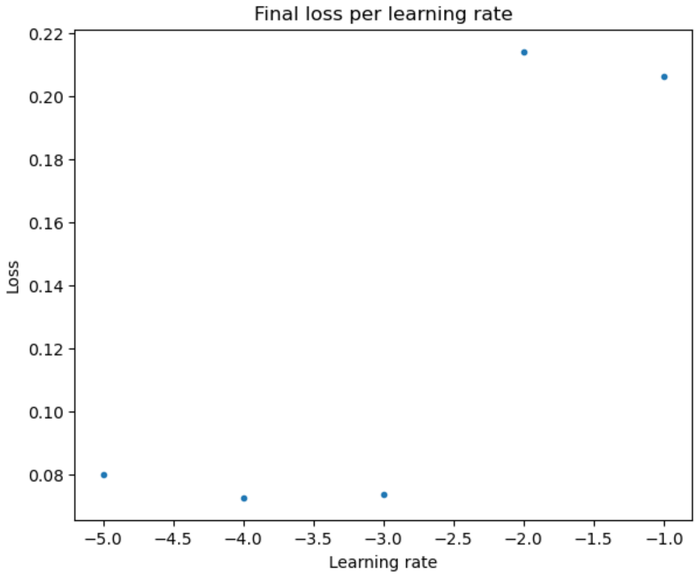


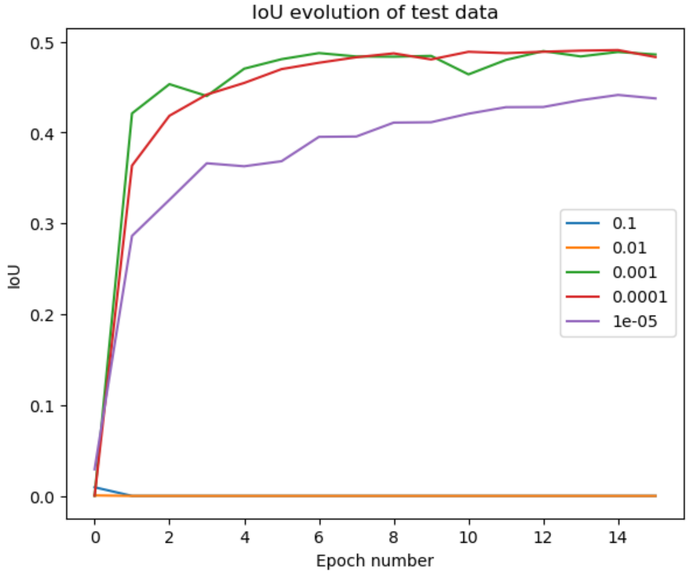
Some cleaning examples (II)

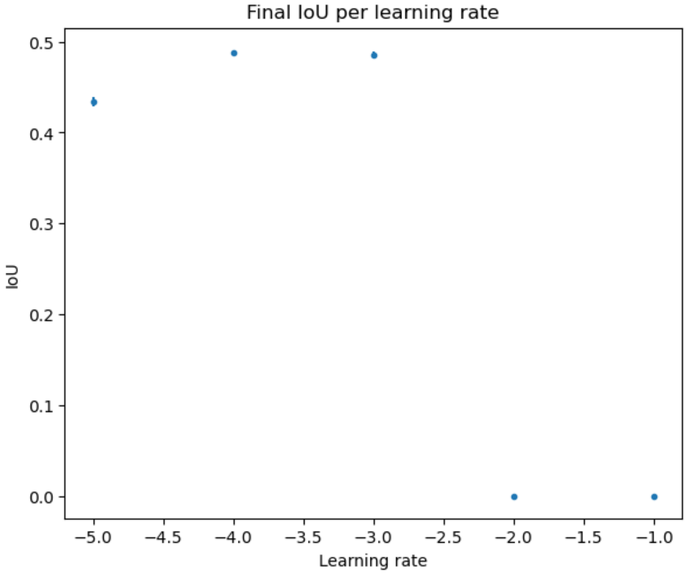


Optimal learning rate



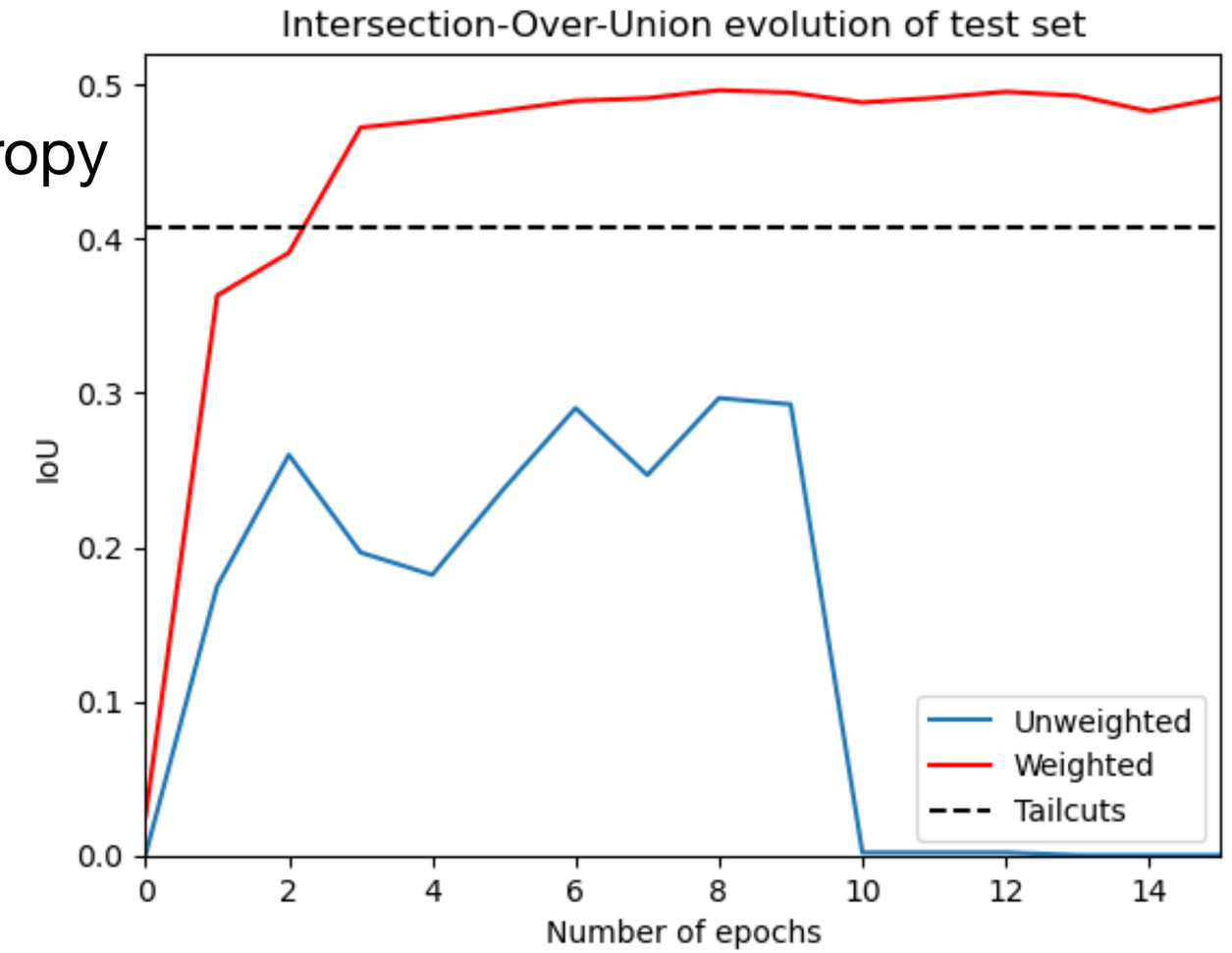






Weighted vs unweighted

- Blue line: binary cross entropy
- Red line: weighted binary cross entropy

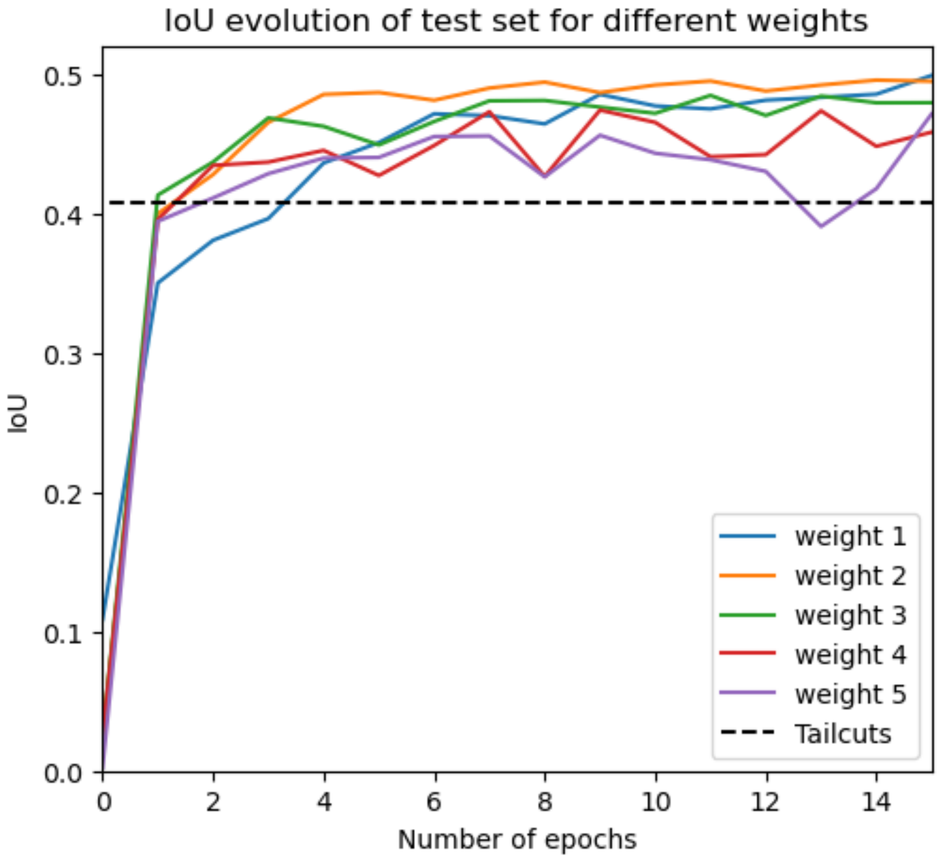


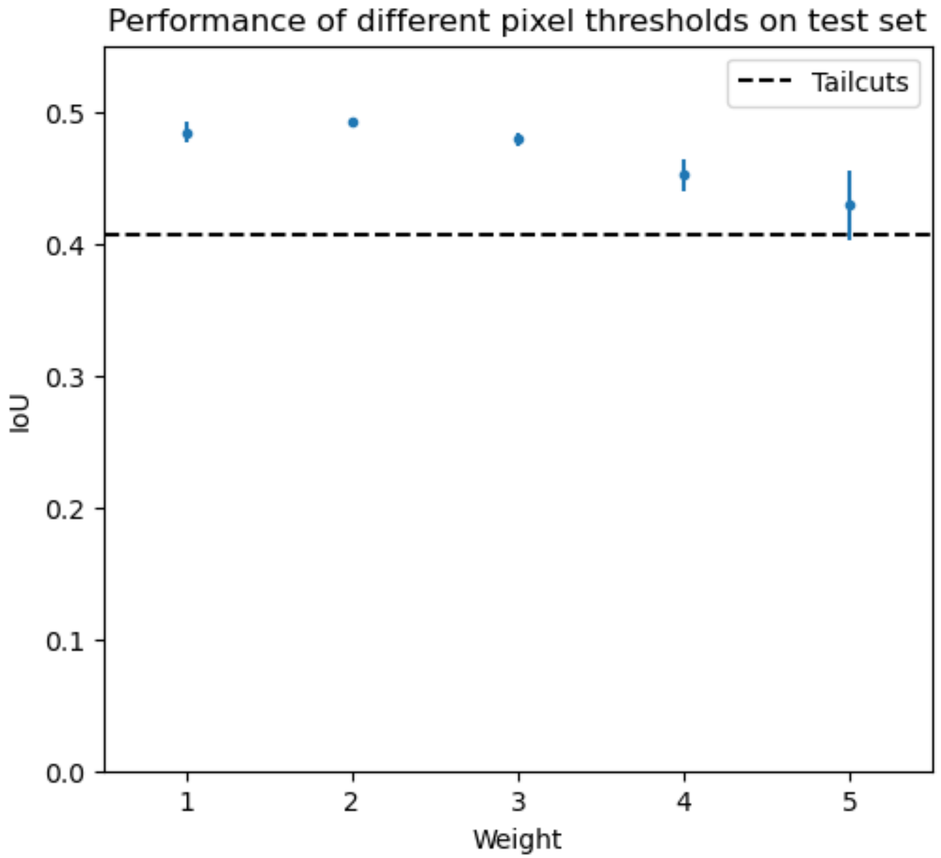
Motivation: Models with normal

Binary cross entropy collapsed for

>2 layers (everything set to 0)

Weighted: different weights

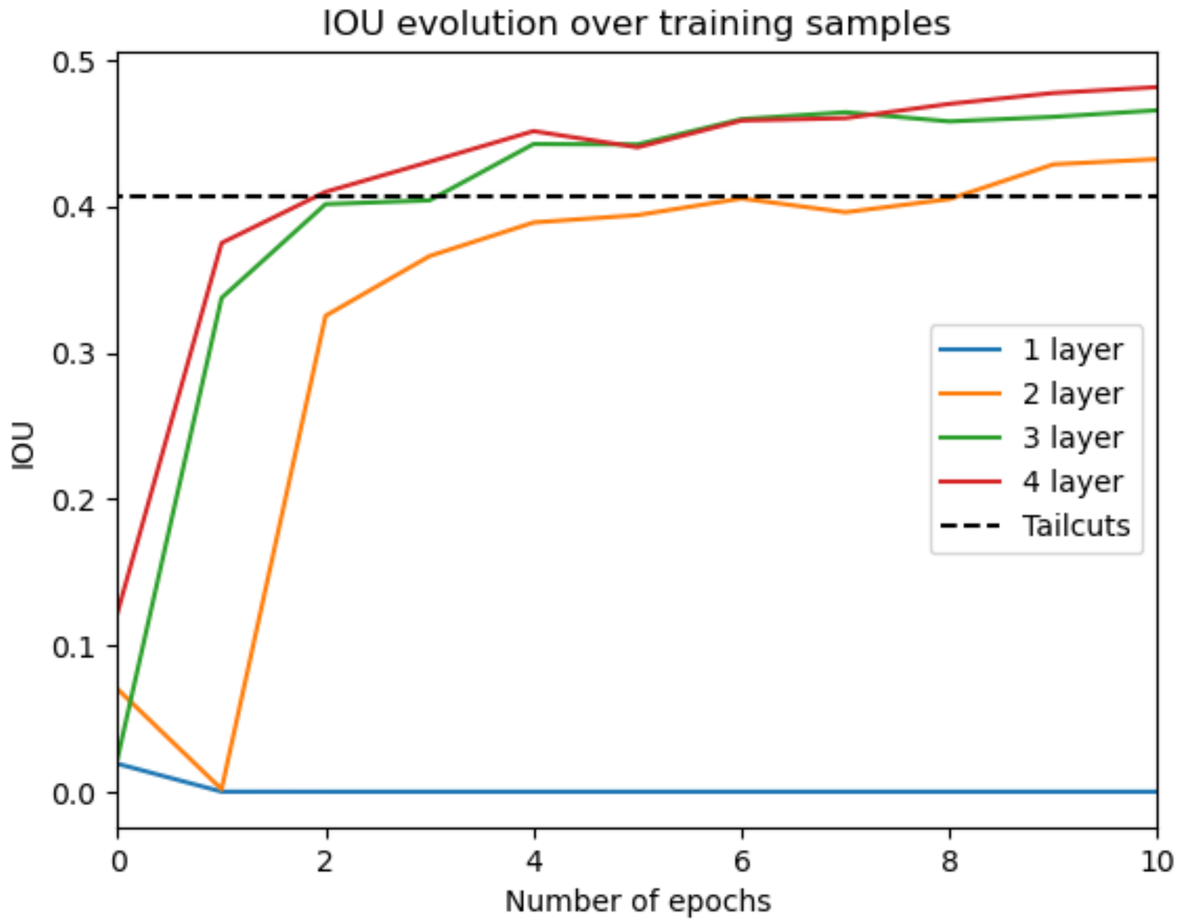




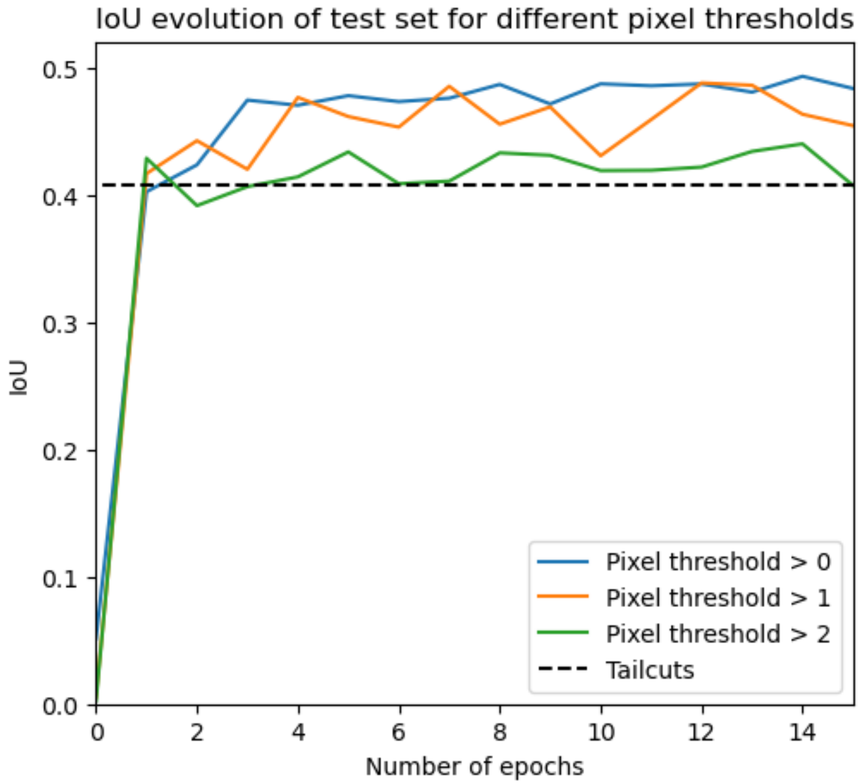
Weird result: weight = 1 performs better than normal binary cross entropy

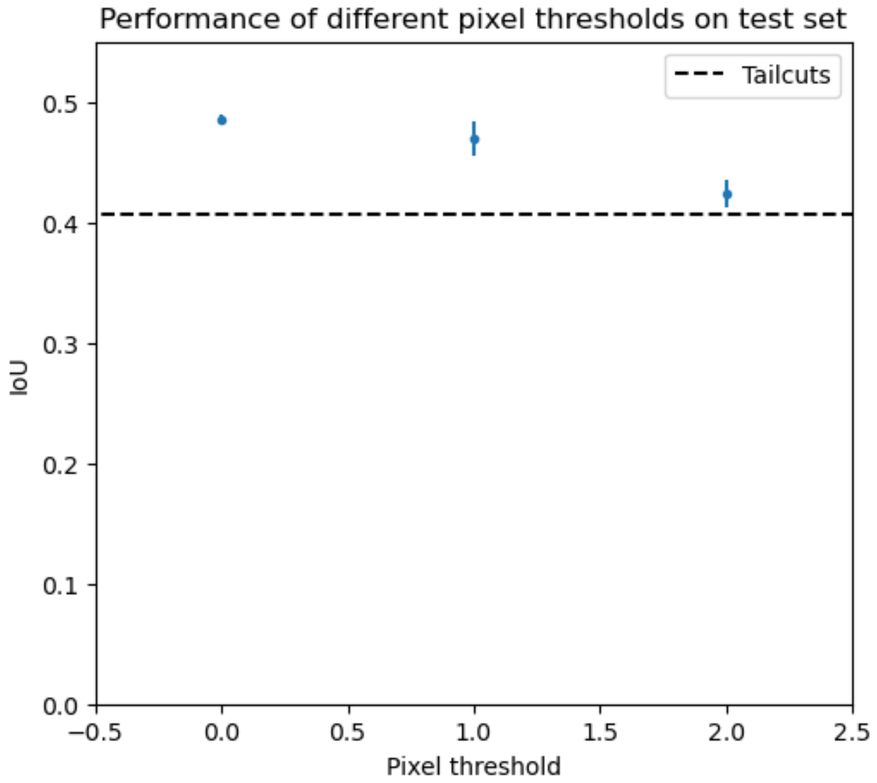
Different number of layers

- 1 layer: ~2k parameters
- 2 layer: ~18k parameters
- 3 layer: ~40k parameters
- 4 layer: ~120k parameters



Different pixel thresholds

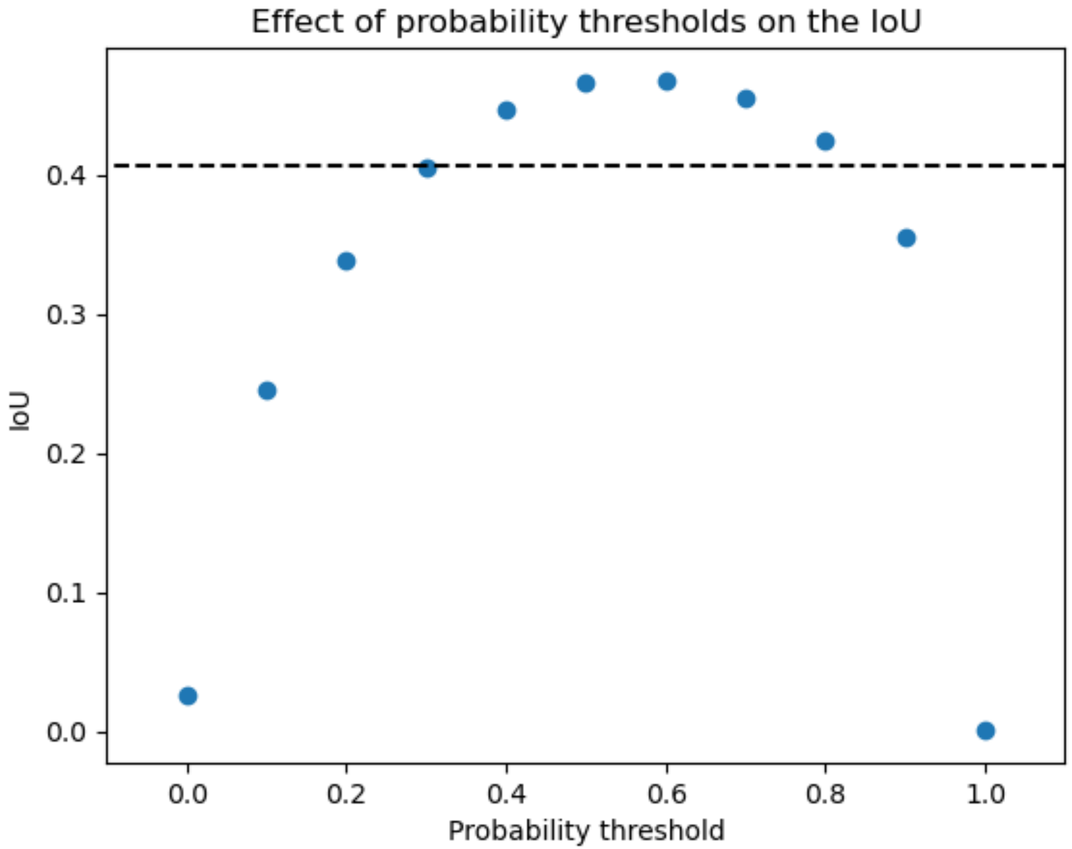




Pixel threshold is the threshold pixels

are included in the binary true image mask

Different probability thresholds

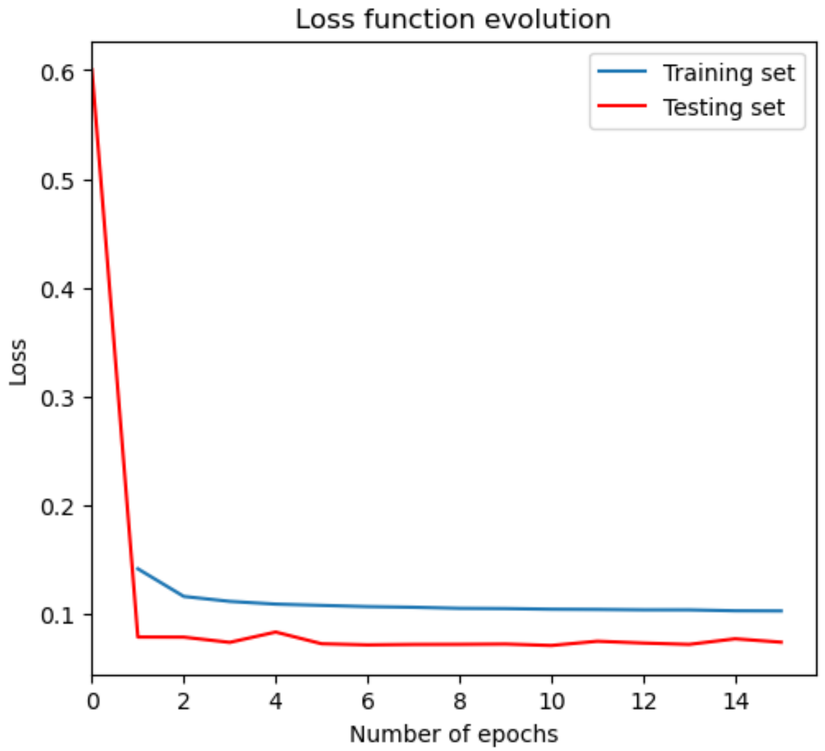


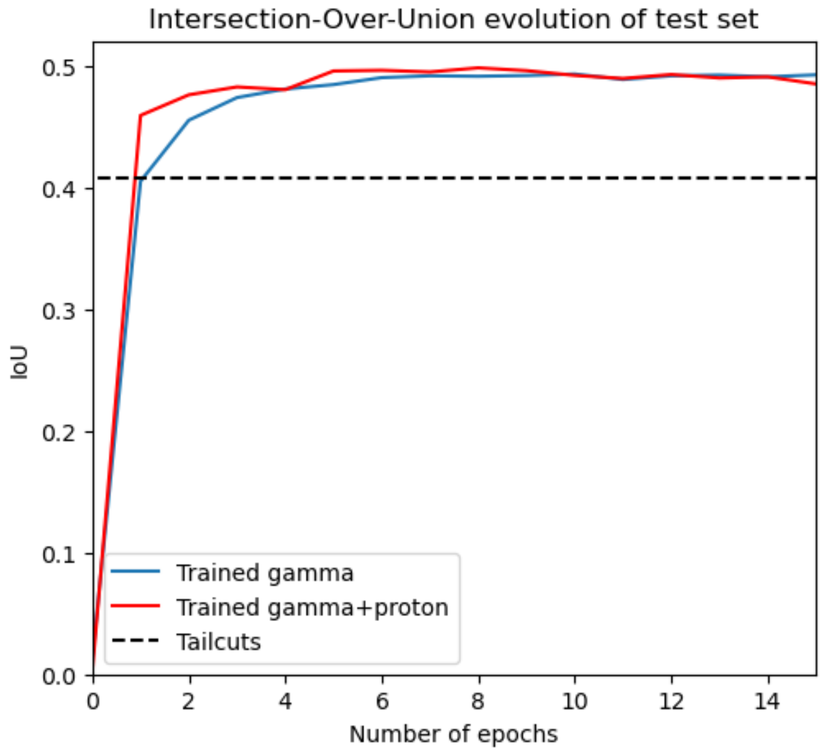
This is the threshold from which probability a pixel is included as signal

(Normally > 0.5)

Training on gamma + protons

	Train	Test
Model 1	γ	γ
Model 2	$\gamma + p$	γ





Adding star noise manually

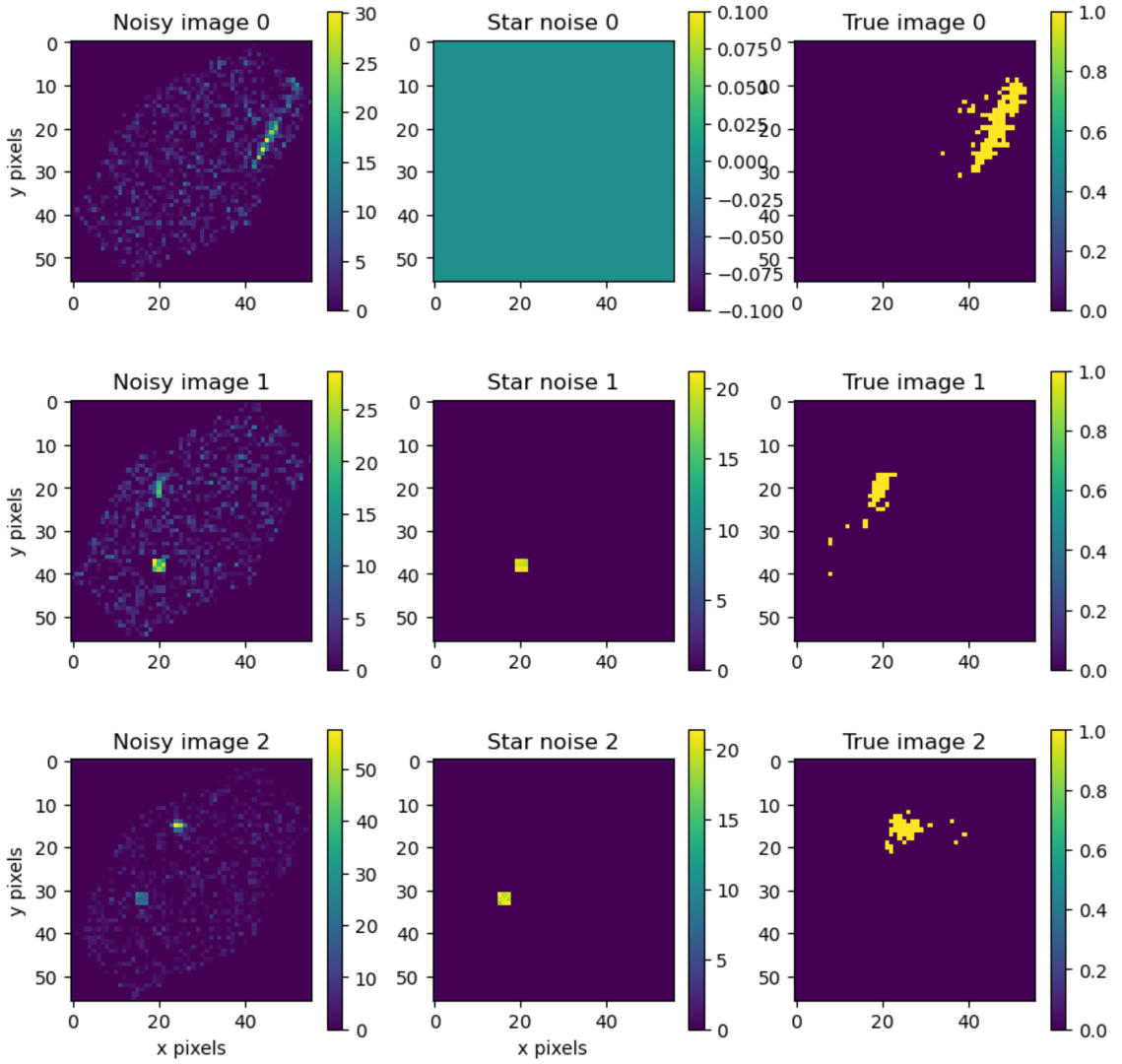
Auto-encoder gets 2 channel input:

- Noisy image
- Star noise image

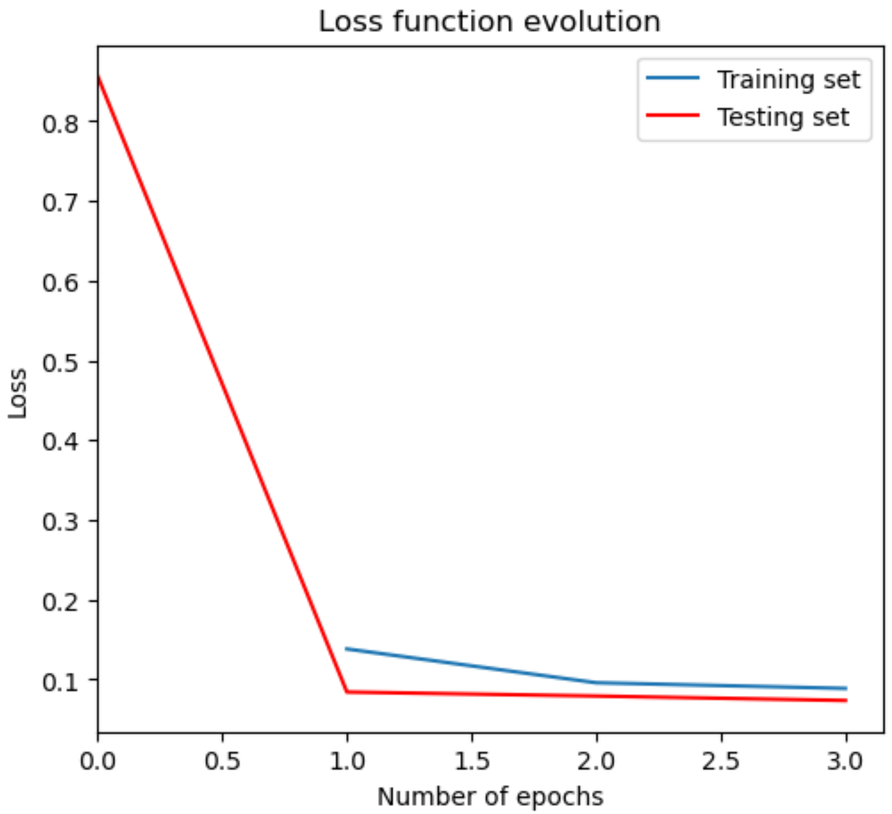
Star is 9 pixels at random location.

Mean of 20 and std of 1

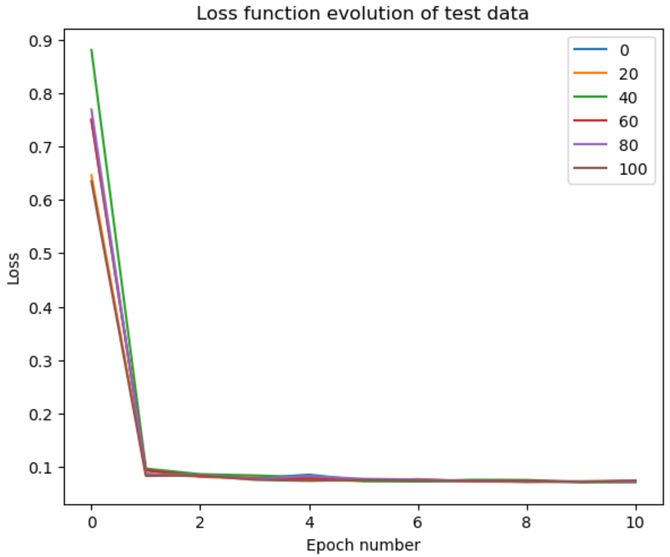
~70% of the pictures have a star

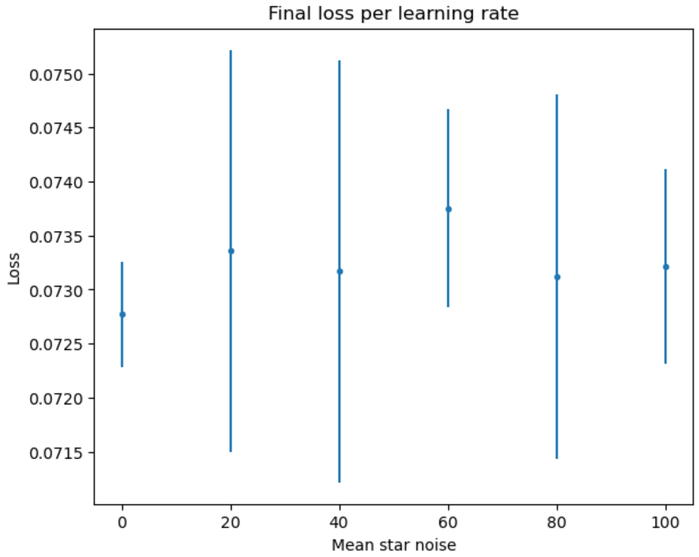


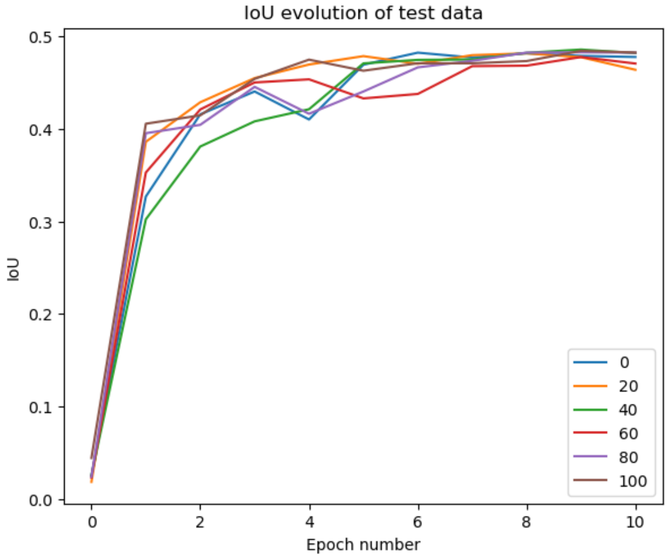
Star noise: IoU

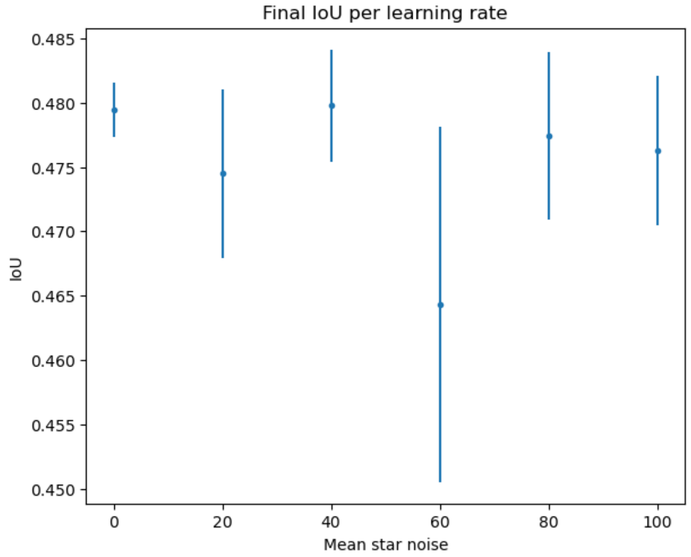


Star noise: Different star means







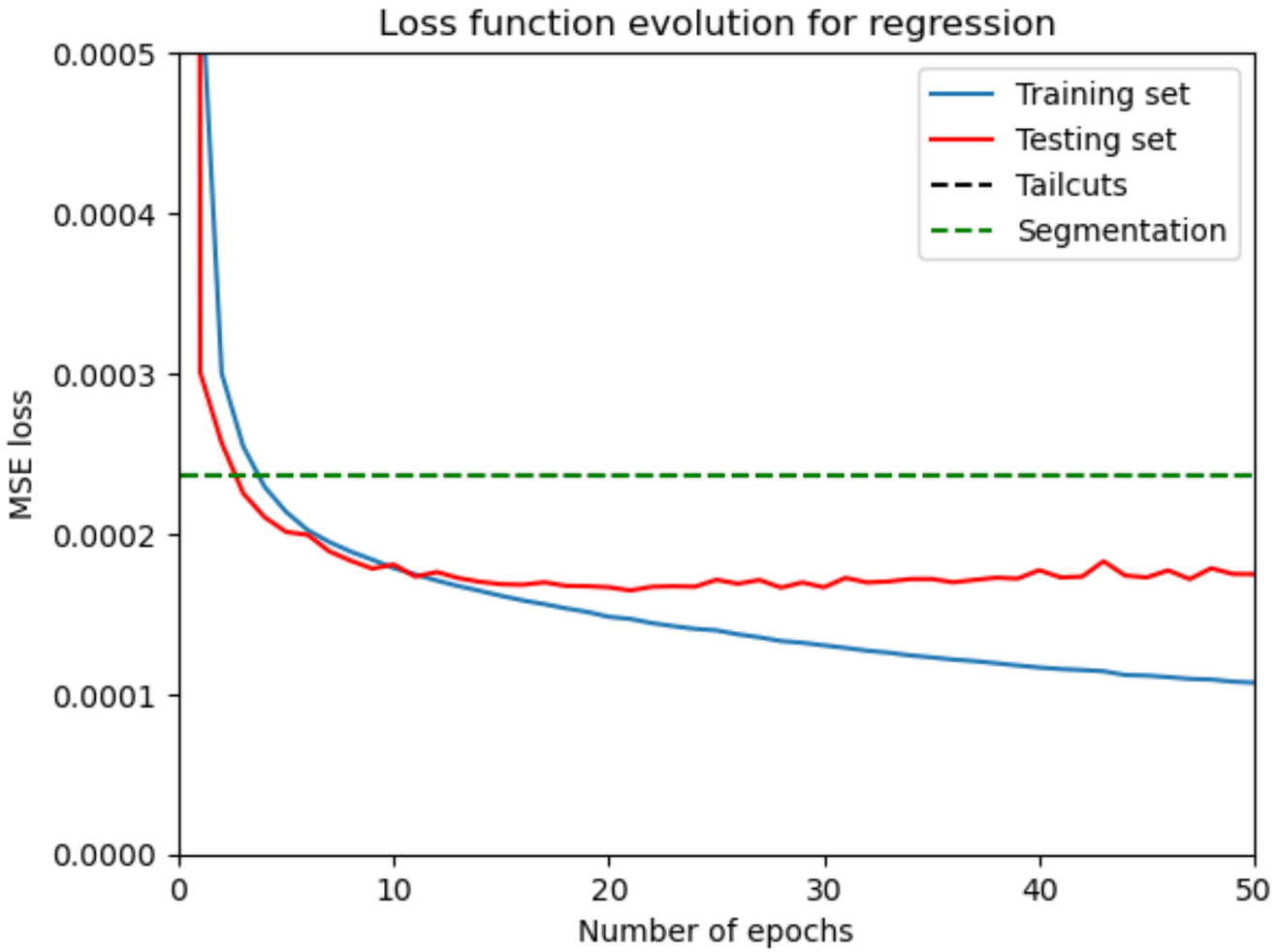


Regression

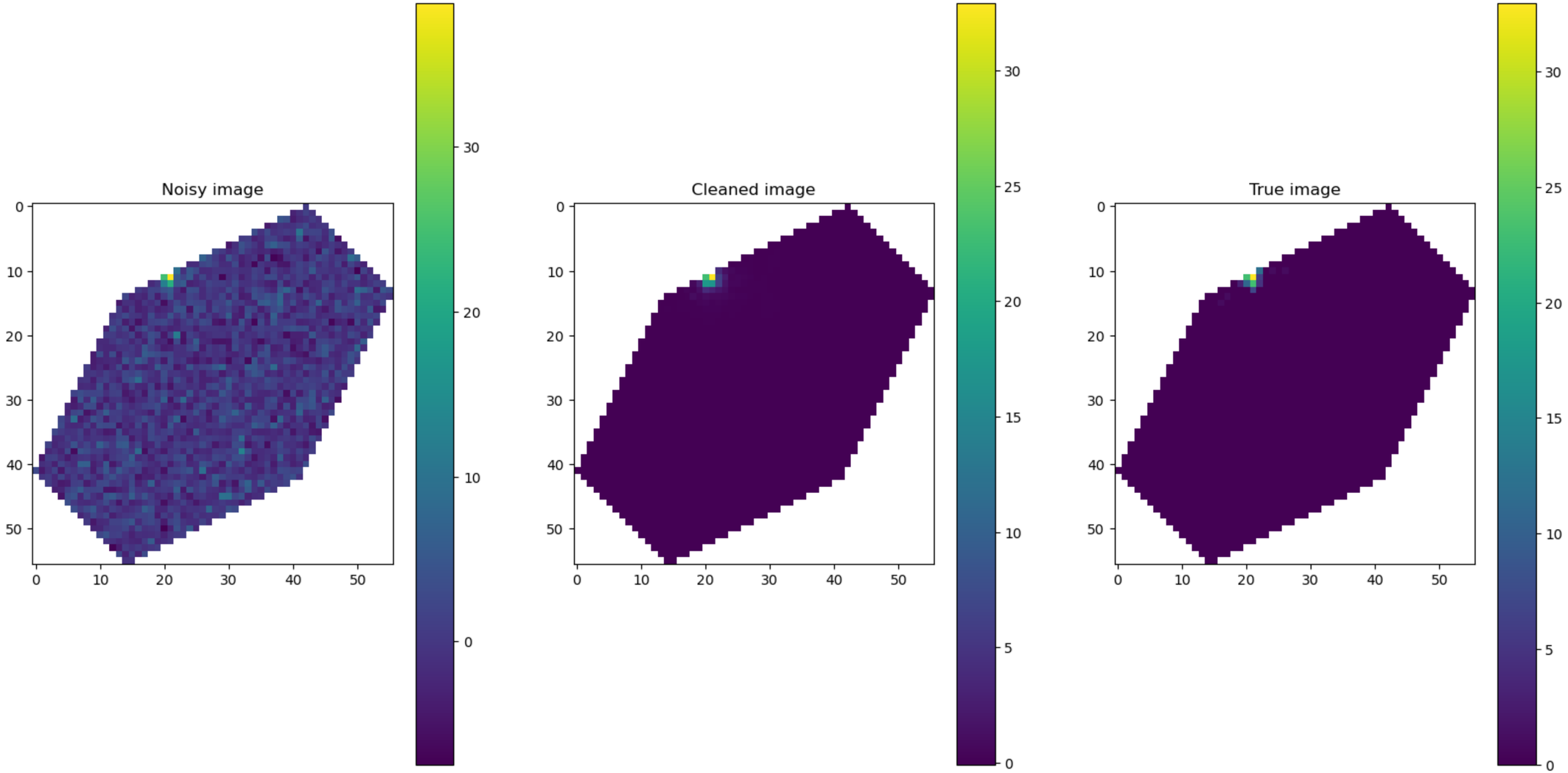
Best auto-encoder model

Using “normalized” MSE

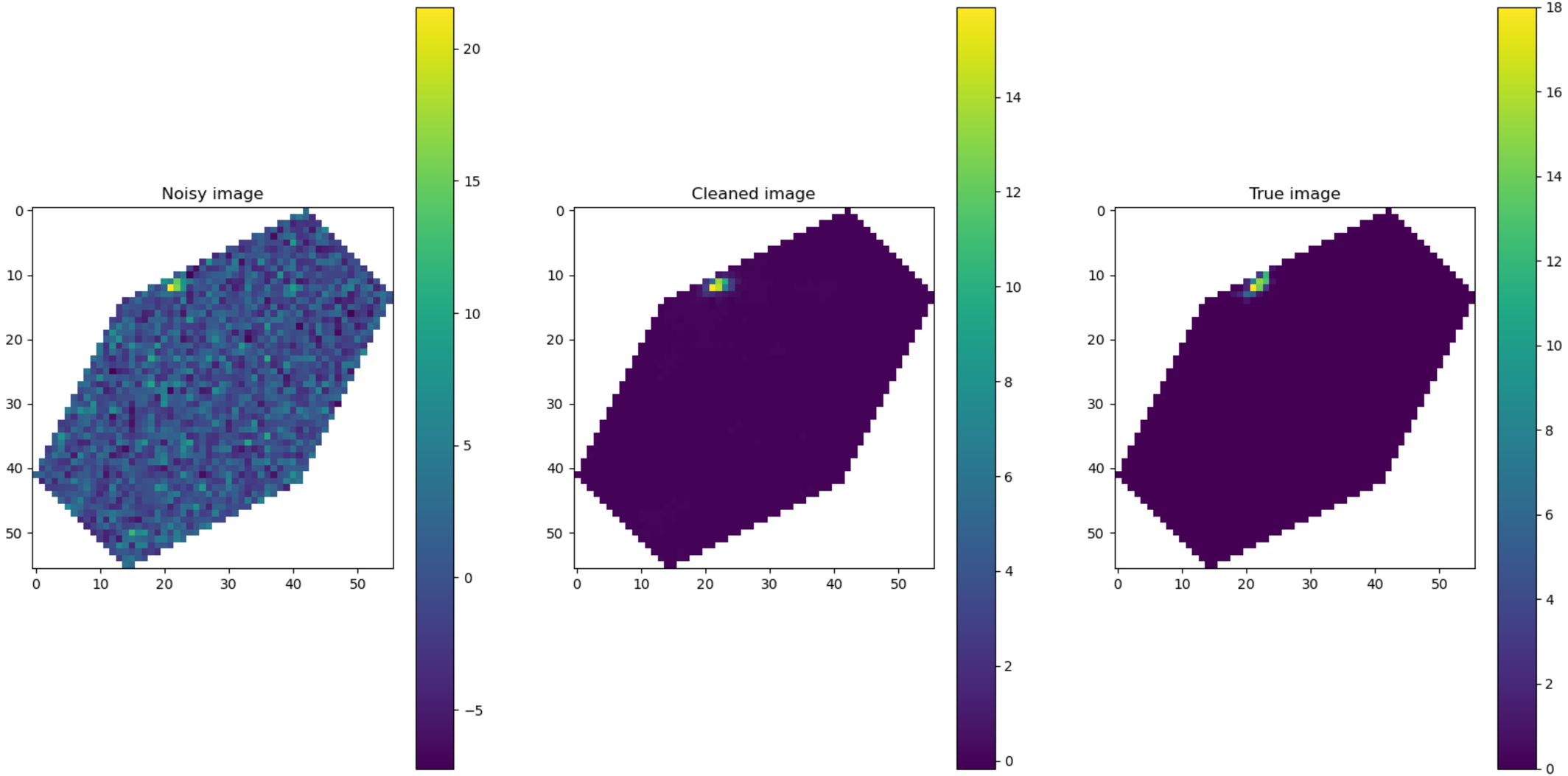
All noisy data transformed to range 0 to 1. True data transformed the same. Ensured that normalized 0 corresponds to pixel value 0.



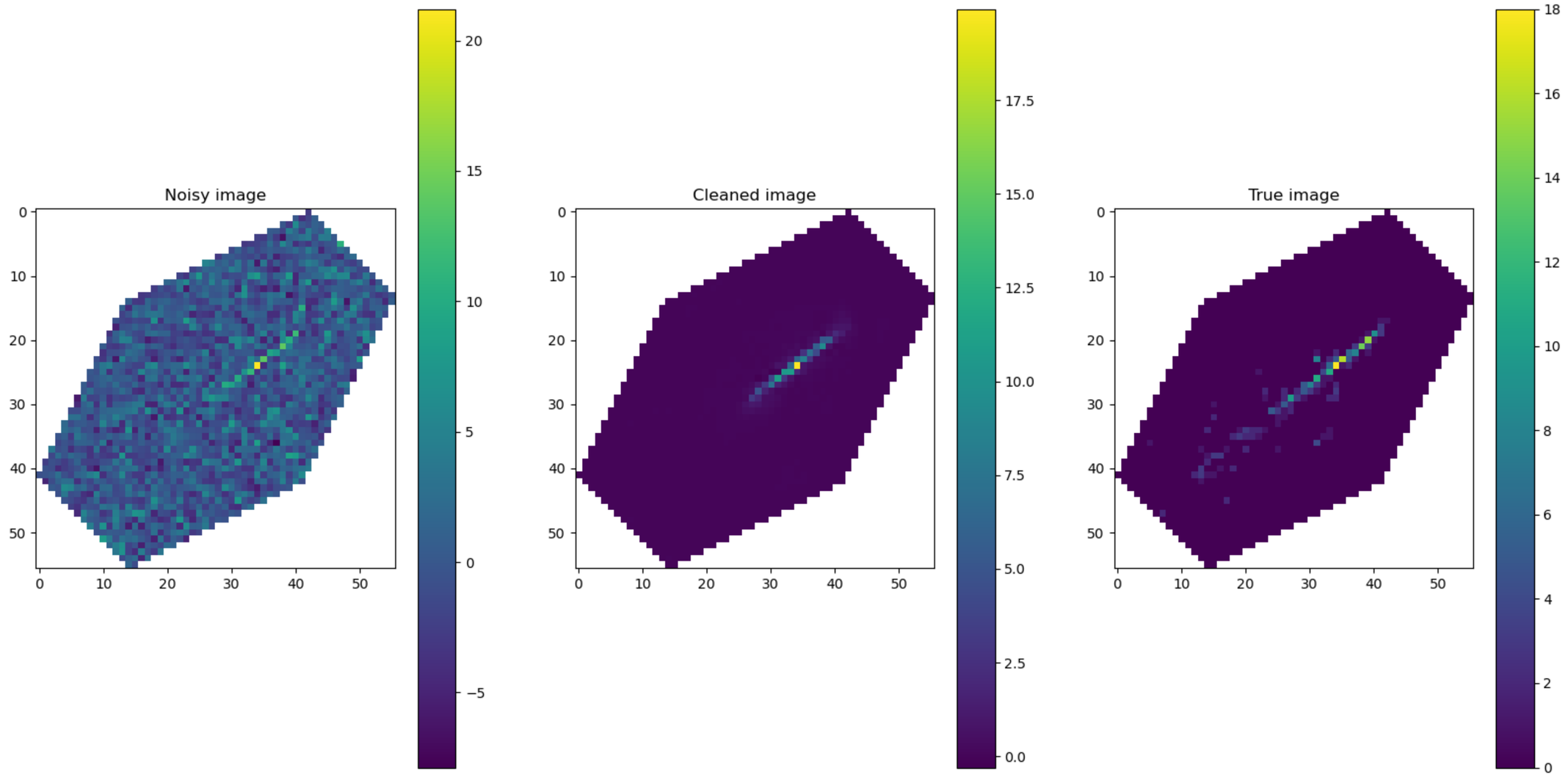
Some cleaning examples (I)



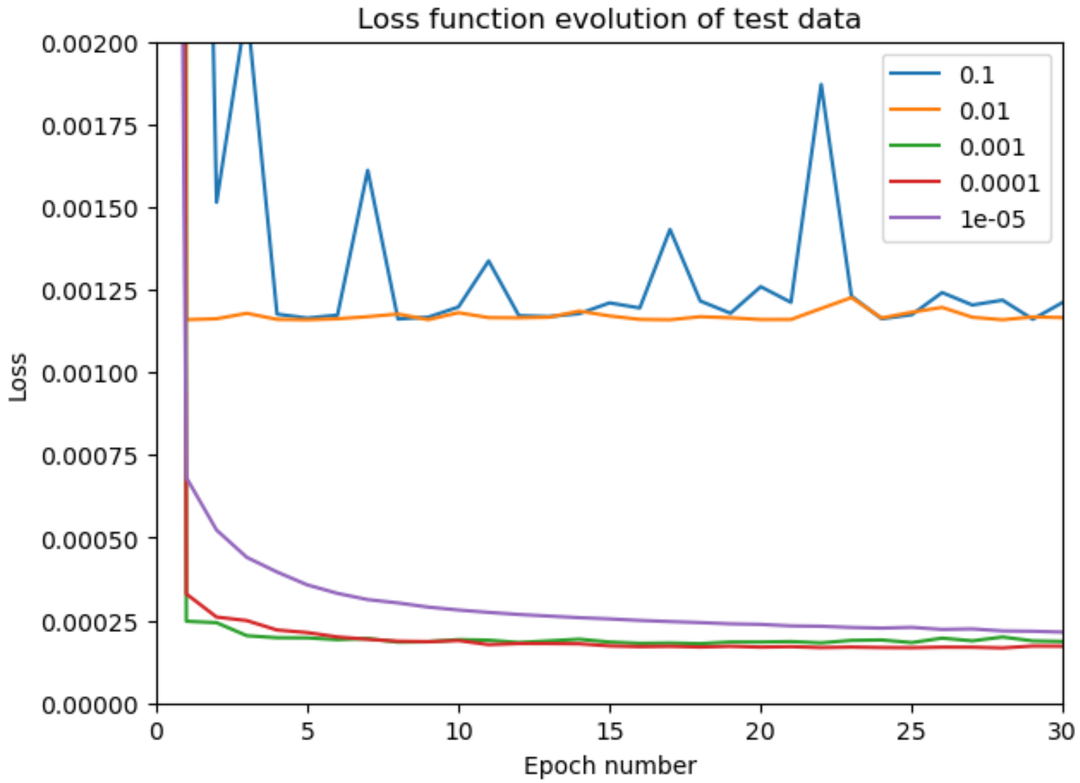
Some cleaning examples (II)

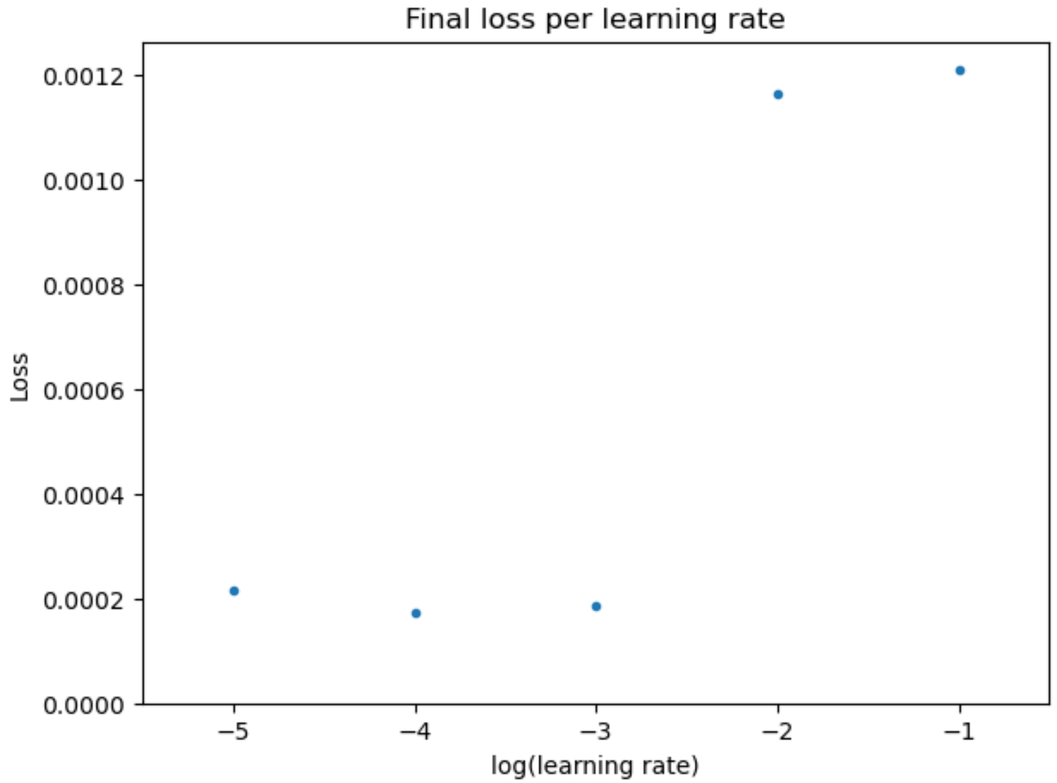


Some cleaning examples (III)

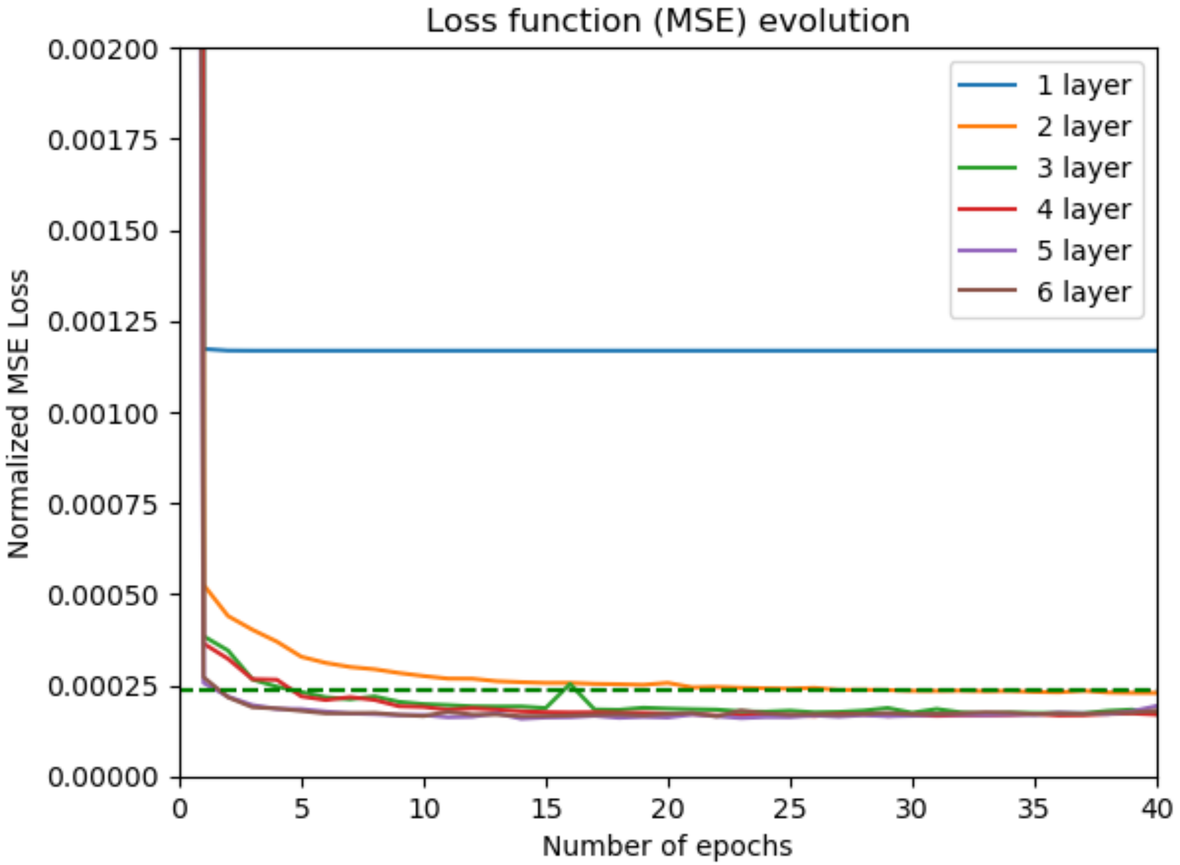


Optimal learning rate

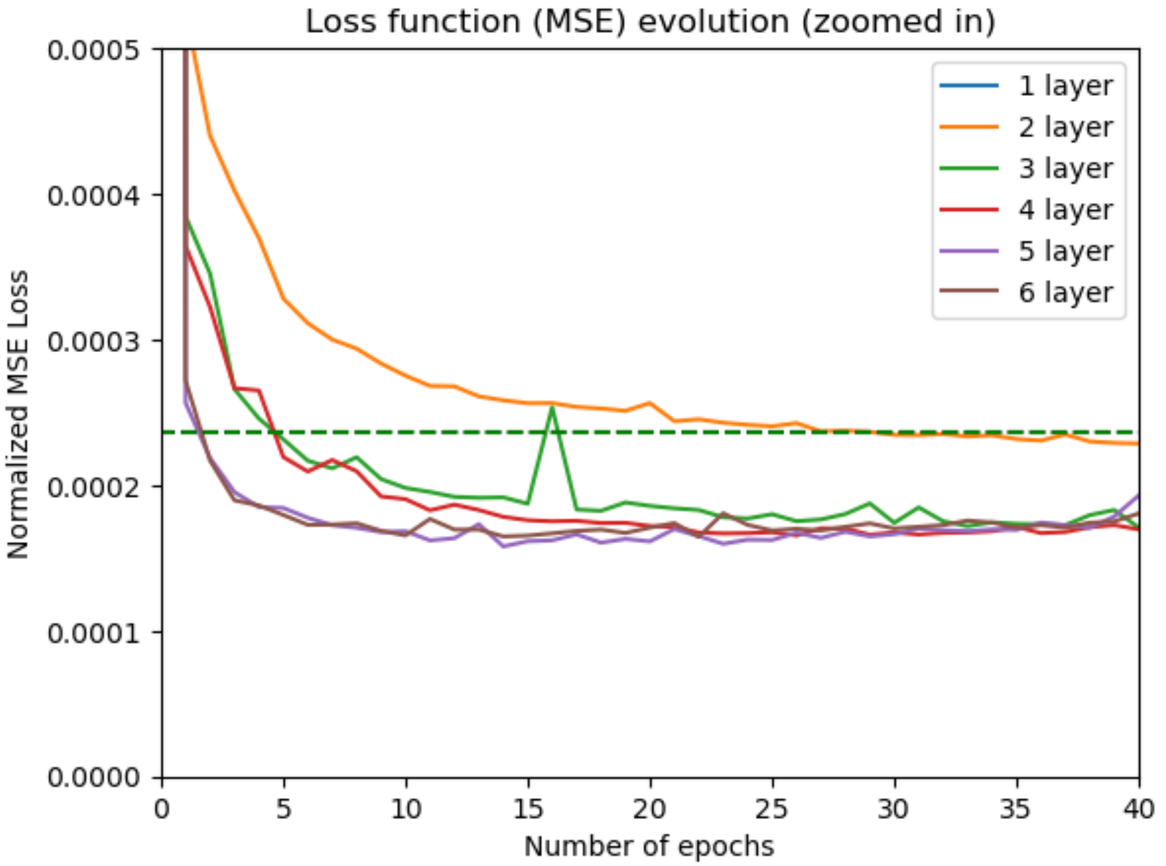




Different number of layers



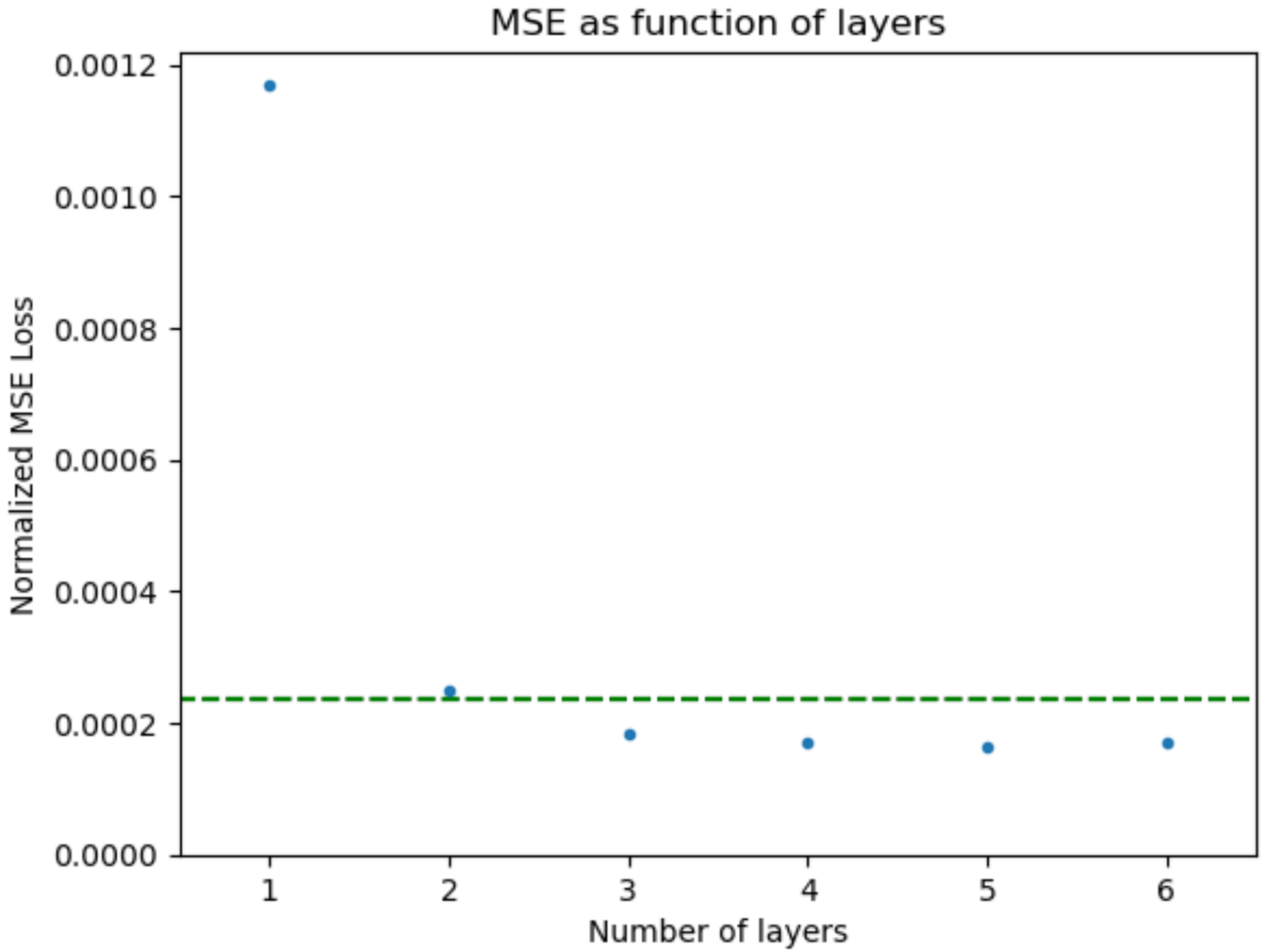
All models



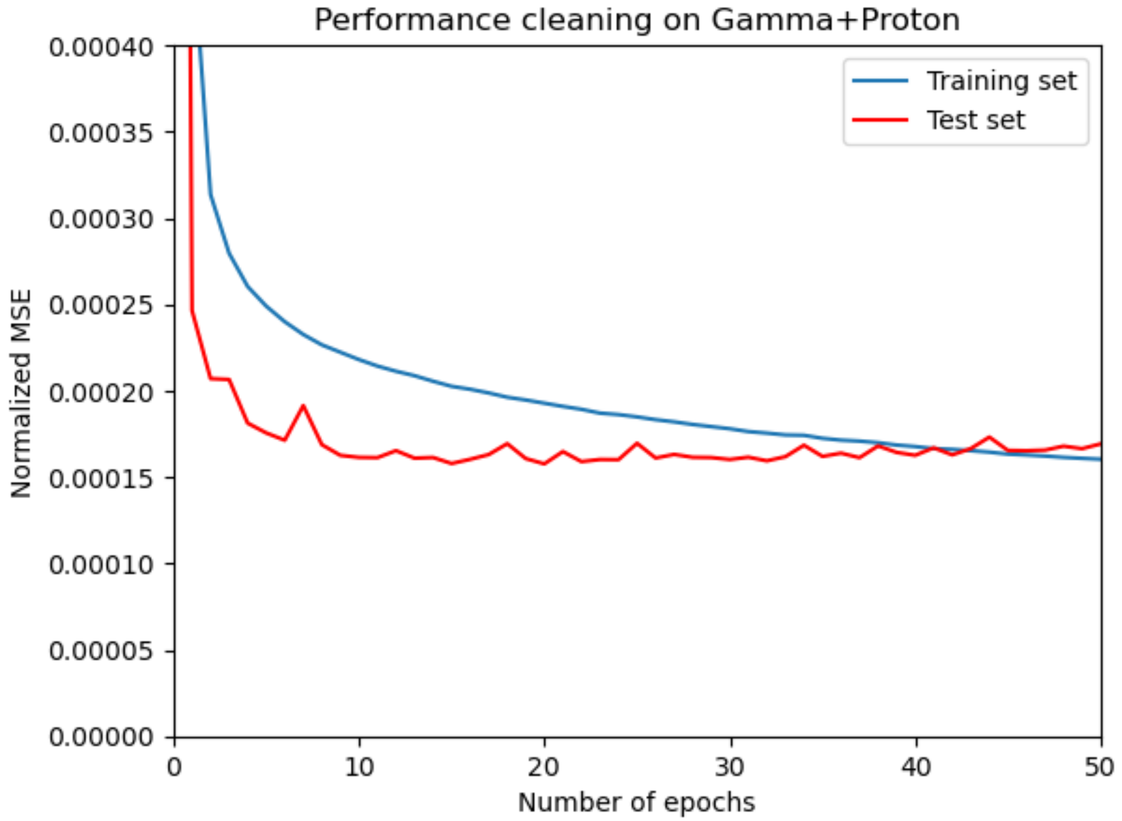
Zoomed in

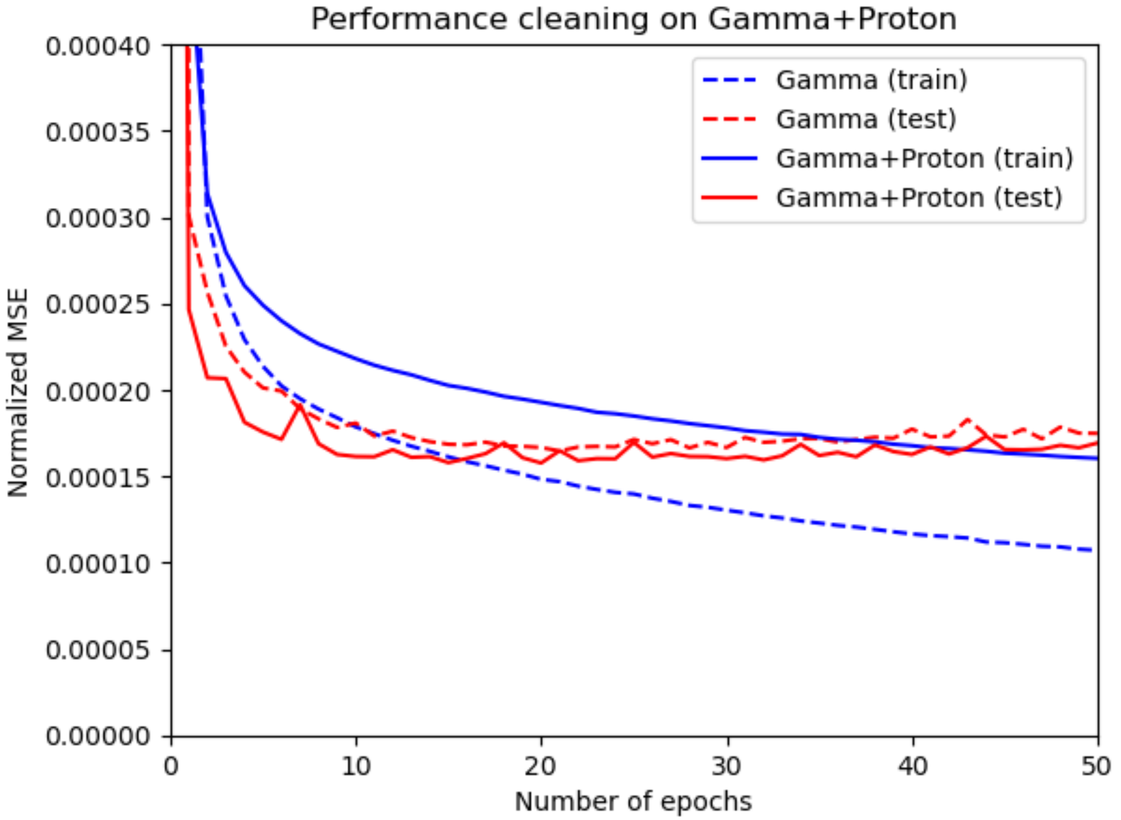
Different number of layers

- 1 layer: ~2k parameters
- 2 layer: ~18k parameters
- 3 layer: ~40k parameters
- 4 layer: ~120k parameters
- 5 layer: ~300k parameters
- 6 layer: ~1 million parameters

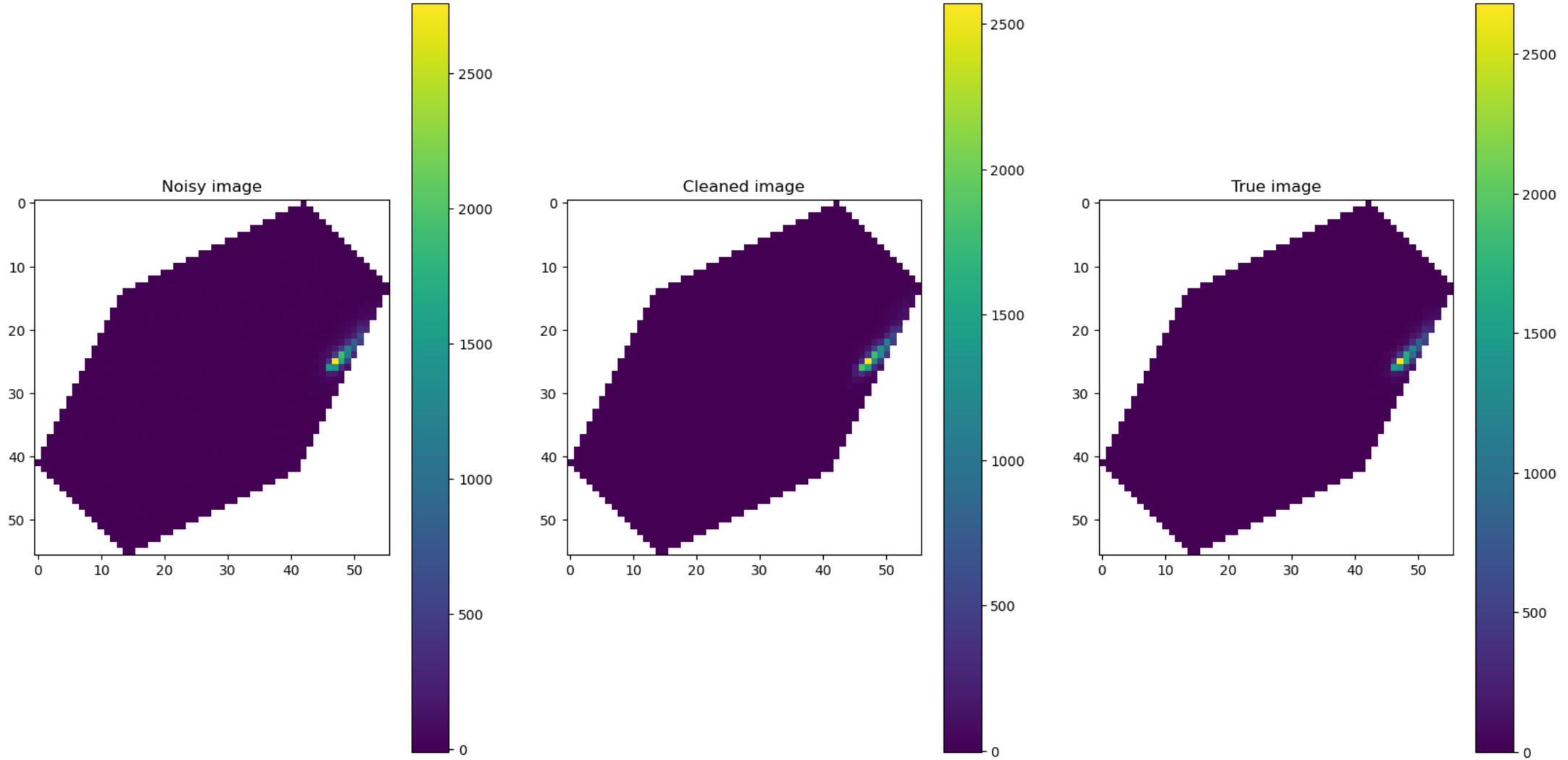


Training on gamma + protons

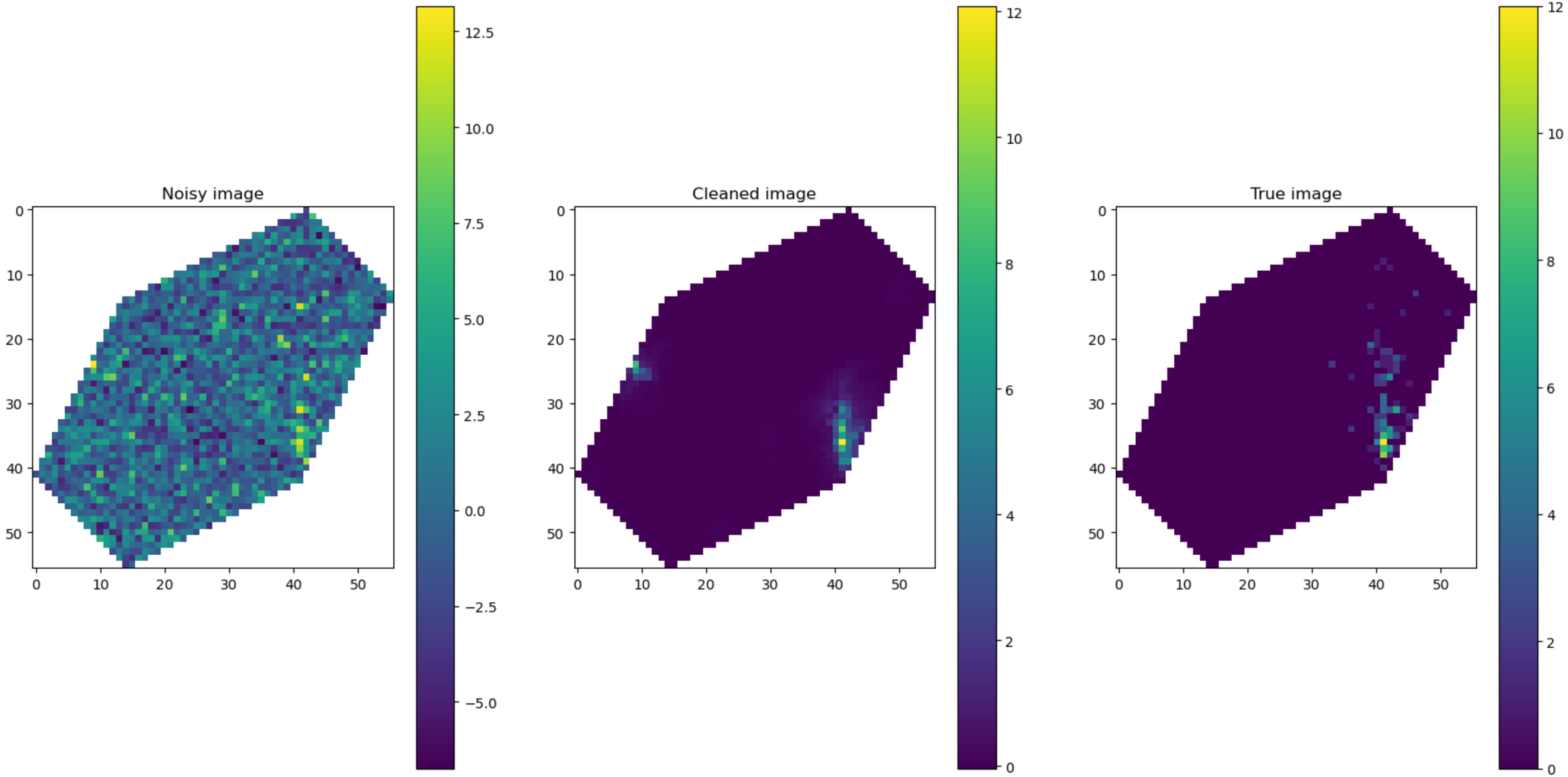




Some cleaning examples (I)



Some cleaning examples (II)



Comparison all methods

	IoU	MSE (10^-3)
Tailcuts	0.41	2.37
Segmentation	0.49	2.36
Regression	0.46	1.64