Machine Learning for pixel selection → *Using Deep learning to improve reconstruction algorithm in low energies region.*

Preliminary Results

Guilherme Lopes with Rafael Nóbrega

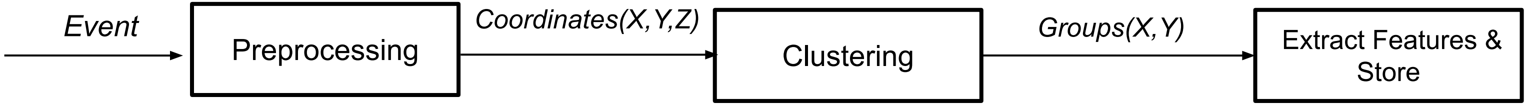
Introduction

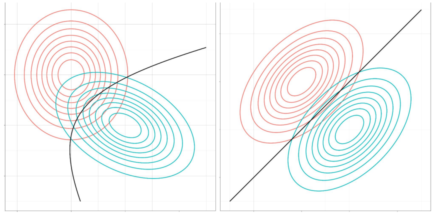
Motivation: Improvements in pixel selection can help reconstruction

Preprocessing stage select the pixel coordinates to be processed in Clustering.

Clustering creates groups based on pixel similarities (positional and intensity)

Important groups features are extracted to be analyzed

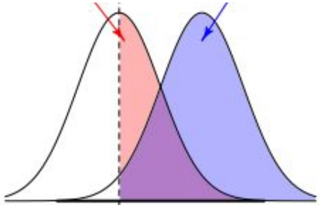




Select pixels is a challenge, mainly for lower energy events.

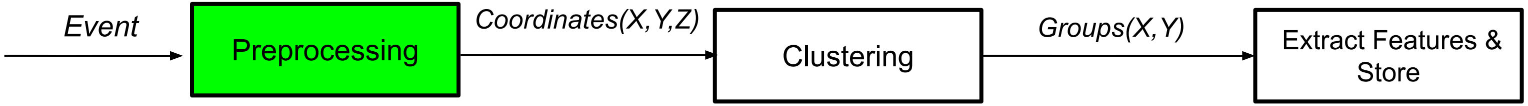
Time spent in DBSCAN increases with number of samples

High efforts are spent with background pixels



False alarm events can be stored
Important information can be loss throughout steps

Motivation: Improvements in pixel selection can help reconstruction

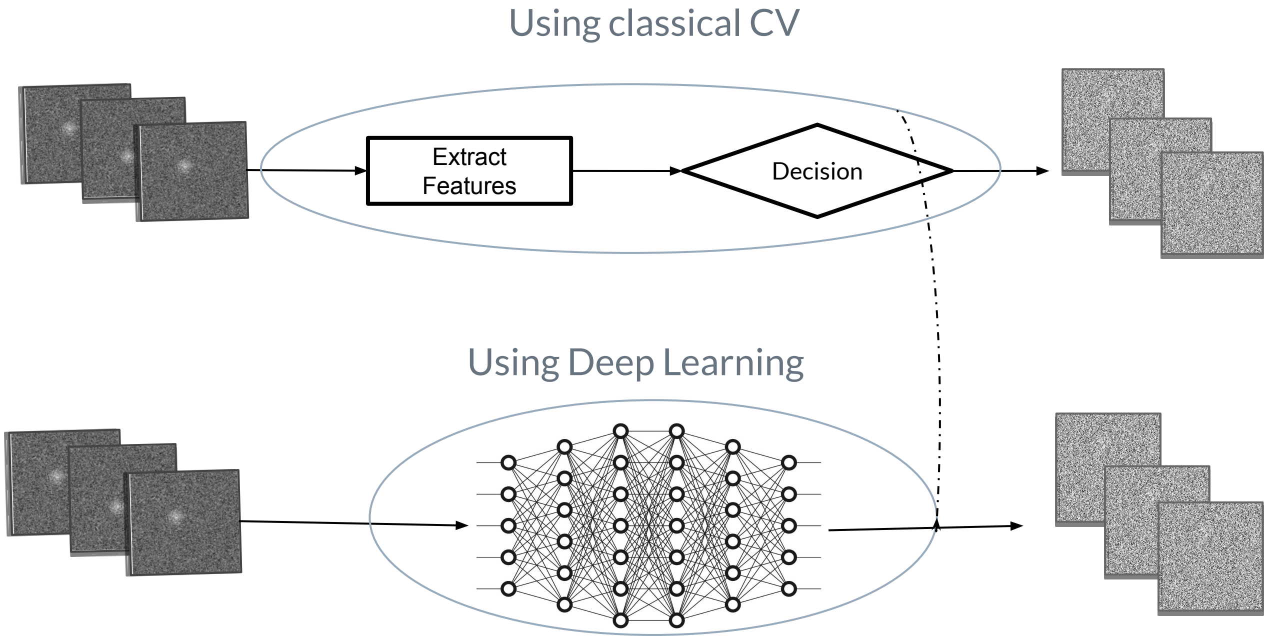


Our work is concerned here

A good pixel selection stage
will improve clustering results
at the sime time to reduce
processing time

And store only relevant
information for post analysis

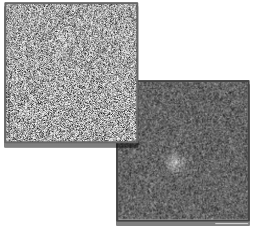
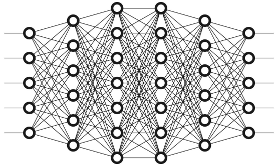
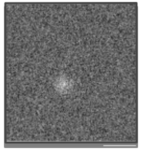
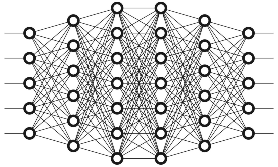
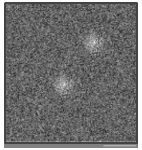
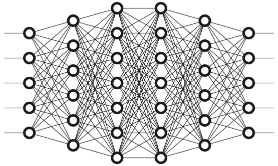
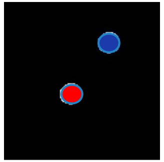
Preprocessing: Challenges on pixel selection task



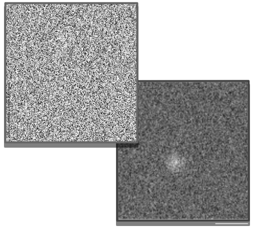
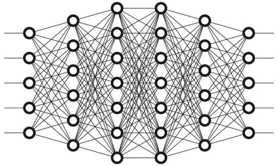
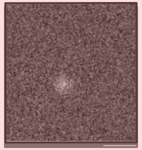
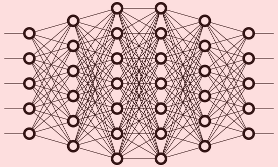
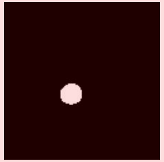
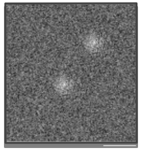
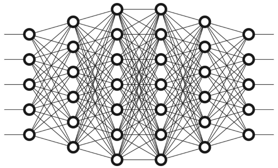
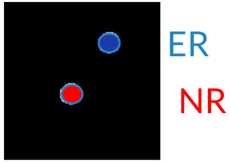
Feature engineering can be complex for low energy events

Feature engineering is replaced by an optimization task

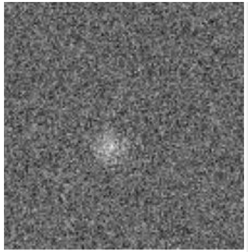
Preprocessing: Performing pixel selection with Deep Learning

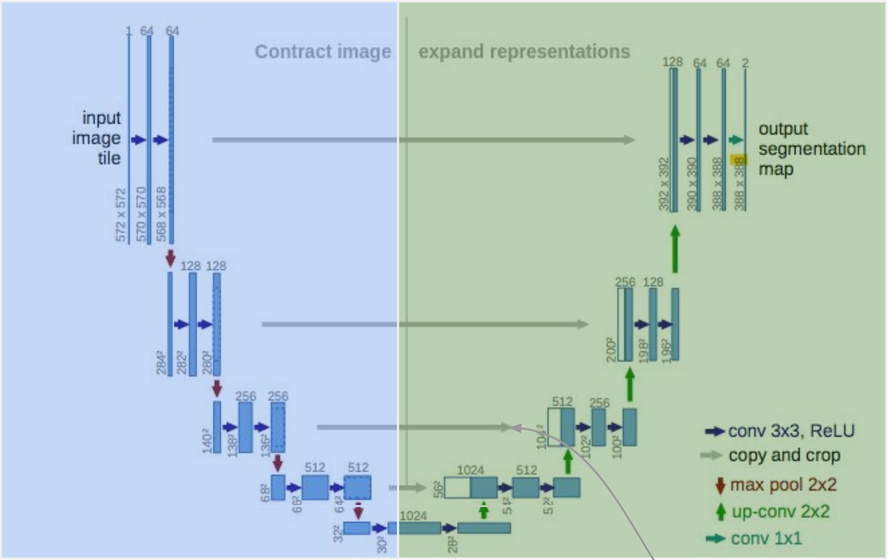
Input	DNN	Output	Task
		Categorical [0,1,..N] **1- has event** **0 - does not have event**	Classification
		Image 	Semantic Segmentation
		Image 	Instance Segmentation

Preprocessing: Performing pixel selection with Deep Learning

Input	DNN	Output	Task
		Categorical [0,1,..N] **1- has event** **0 - does not have event**	Classification
		Image 	Semantic Segmentation
		Image 	Instance Segmentation

Preprocessing: U-Net for semantic segmentation





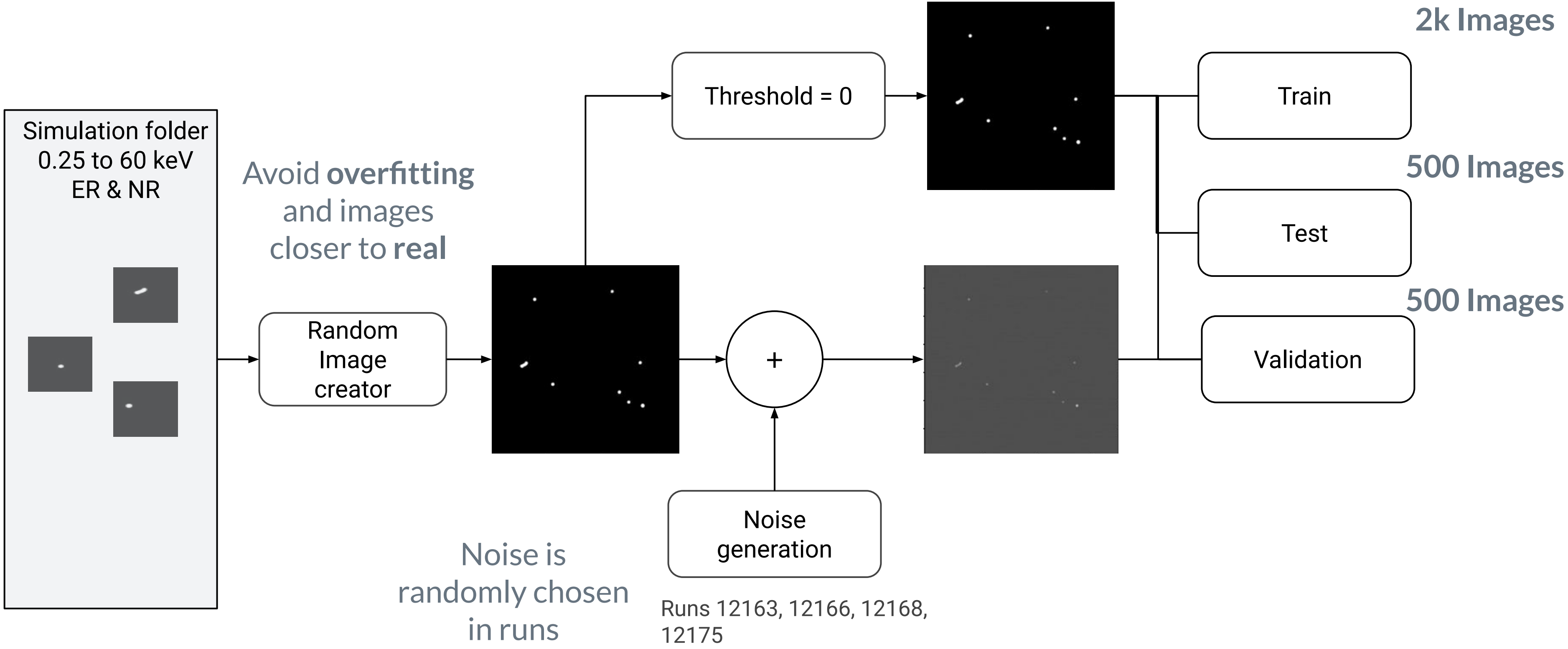
- Expand step will upsample these features,using deconvolution, to allow pixel wise relationship between input and output;

- Contract step is used to build feature maps using convolution;

- High resolution information is fully connected to outputs, to preserve this information.

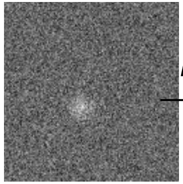
Methodology

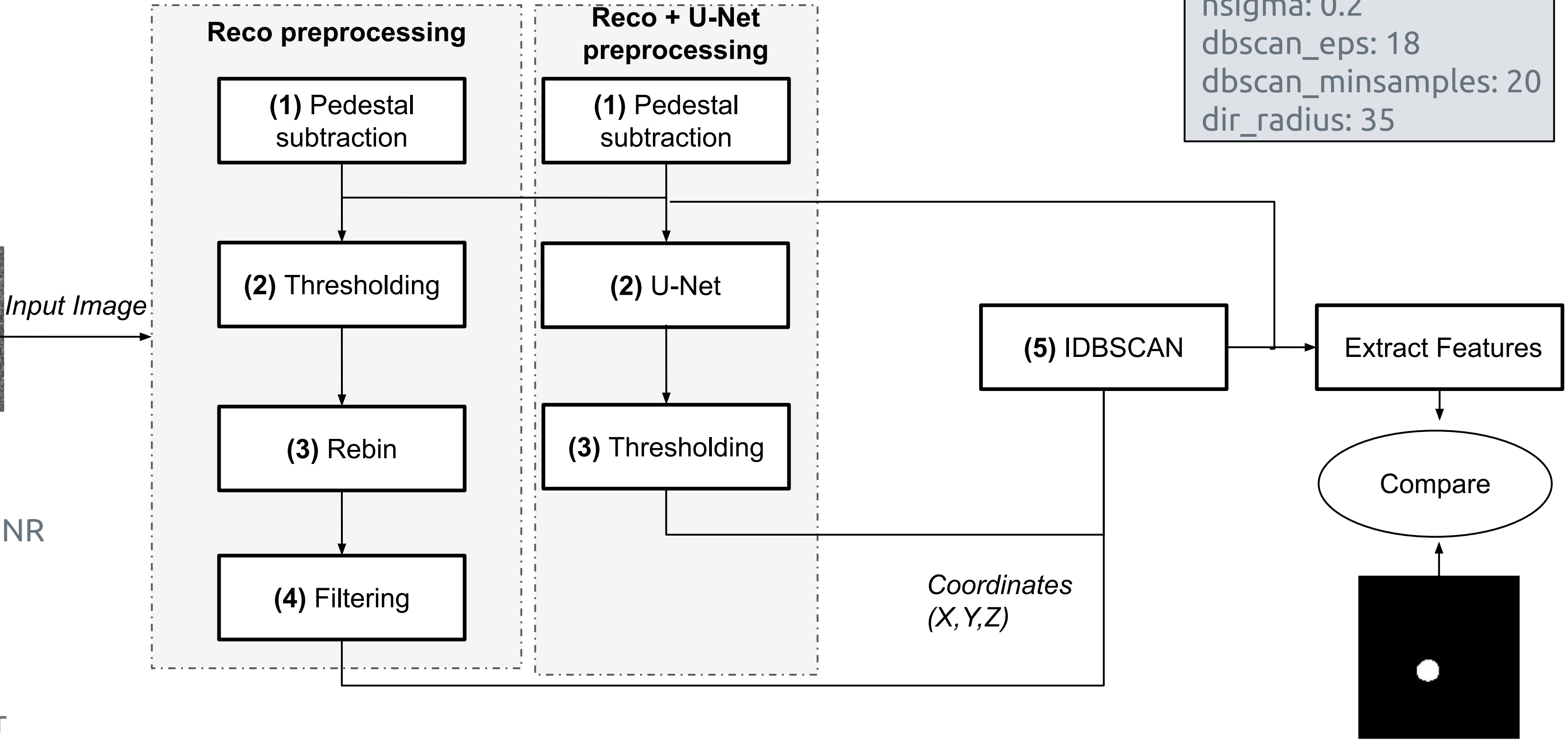
Methodology: Training U-Net from scratch



Methodology: Where trained U-Net can be placed

Parameters
branch: winter23
nsigma: 0.2
dbscan_eps: 18
dbscan_minsamples: 20
dir_radius: 35





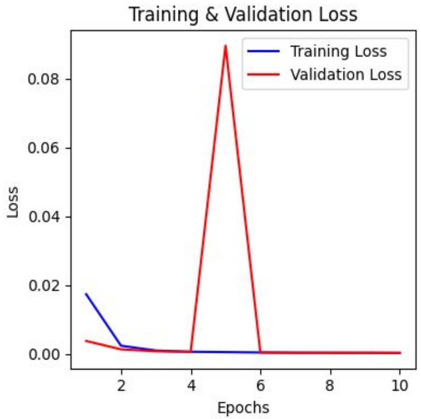
Events:
1 event of NR or ER per image.

Energies:
0.25, 0.5 or 1keV

Results

U-Net: Training results

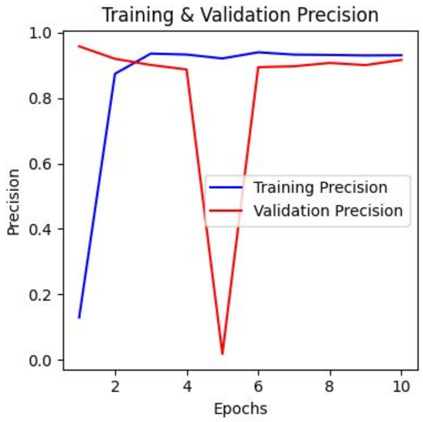
Loss

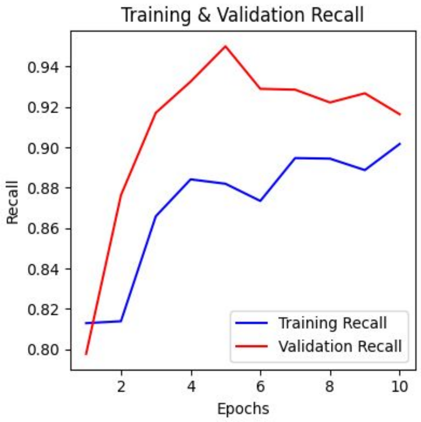


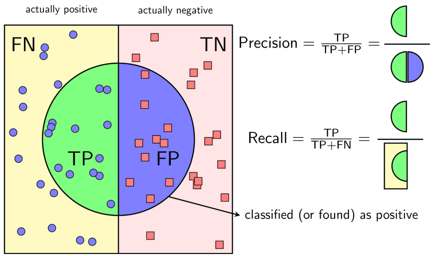
$$\mathrm{FL}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t).$$

Focal Loss was used as loss function. It's a generalization of Binary Cross Entropy. We can use alpha parameter to control trade-off between precision and recall. In our case we use an alpha = 4 to priorize Recall.

Metrics





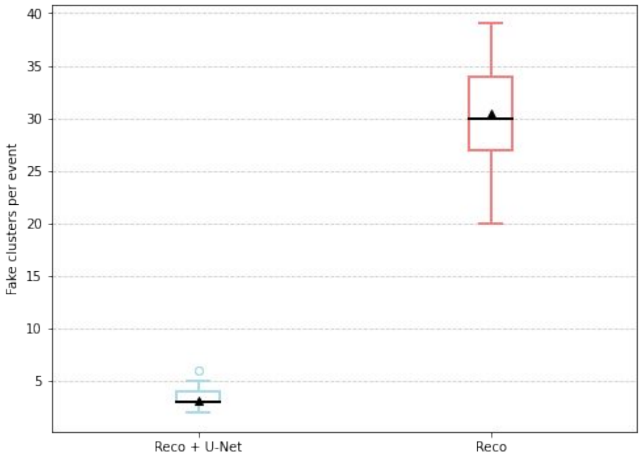


The rate of truth pixels on classified pixels as true

The rate of truth pixels found

U-Net: Results of U-Net used in reconstruction

Fake events

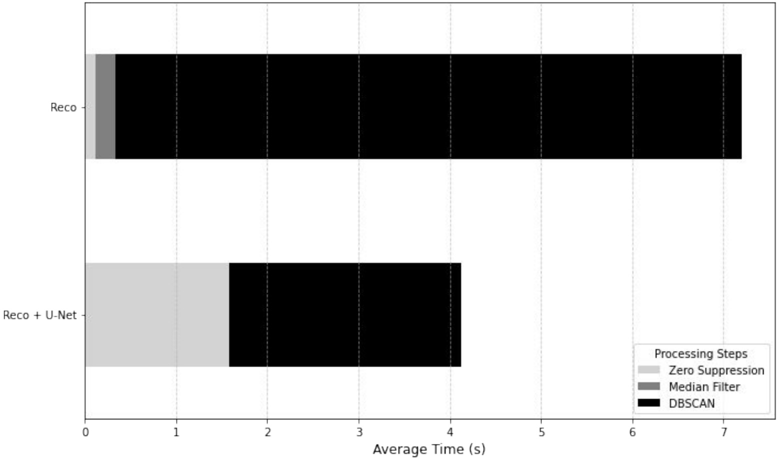


Each image has just **one** event

99% of truth events were detected

Reconstruction statistically detects less fake clusters when uses U-Net

Processing time



Gray region is the preprocessing step

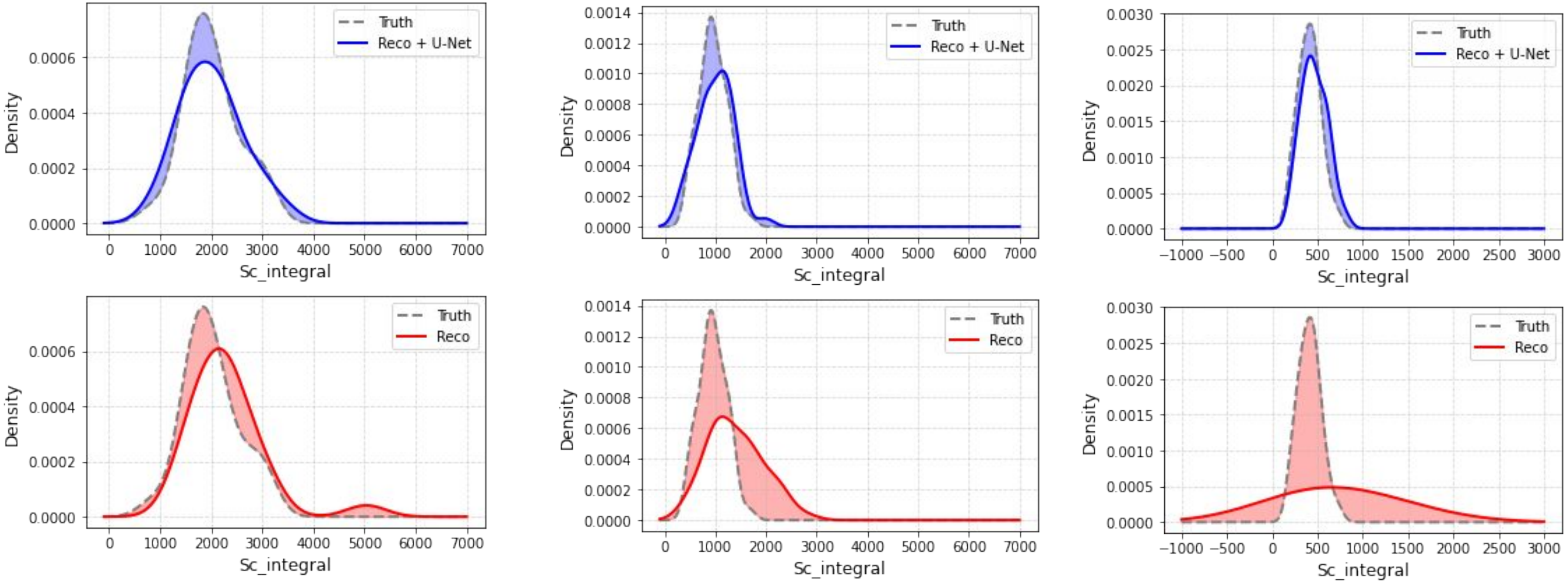
Even using images in **high resolution,** U-Net can send less pixels to DBSCAN and spent less time.

U-Net: Results of U-Net used in reconstruction

Cluster integral

Even for **lower** energies, keep closer to truth

1 keV | **0.5 keV** | **0.25 keV**



Next Steps

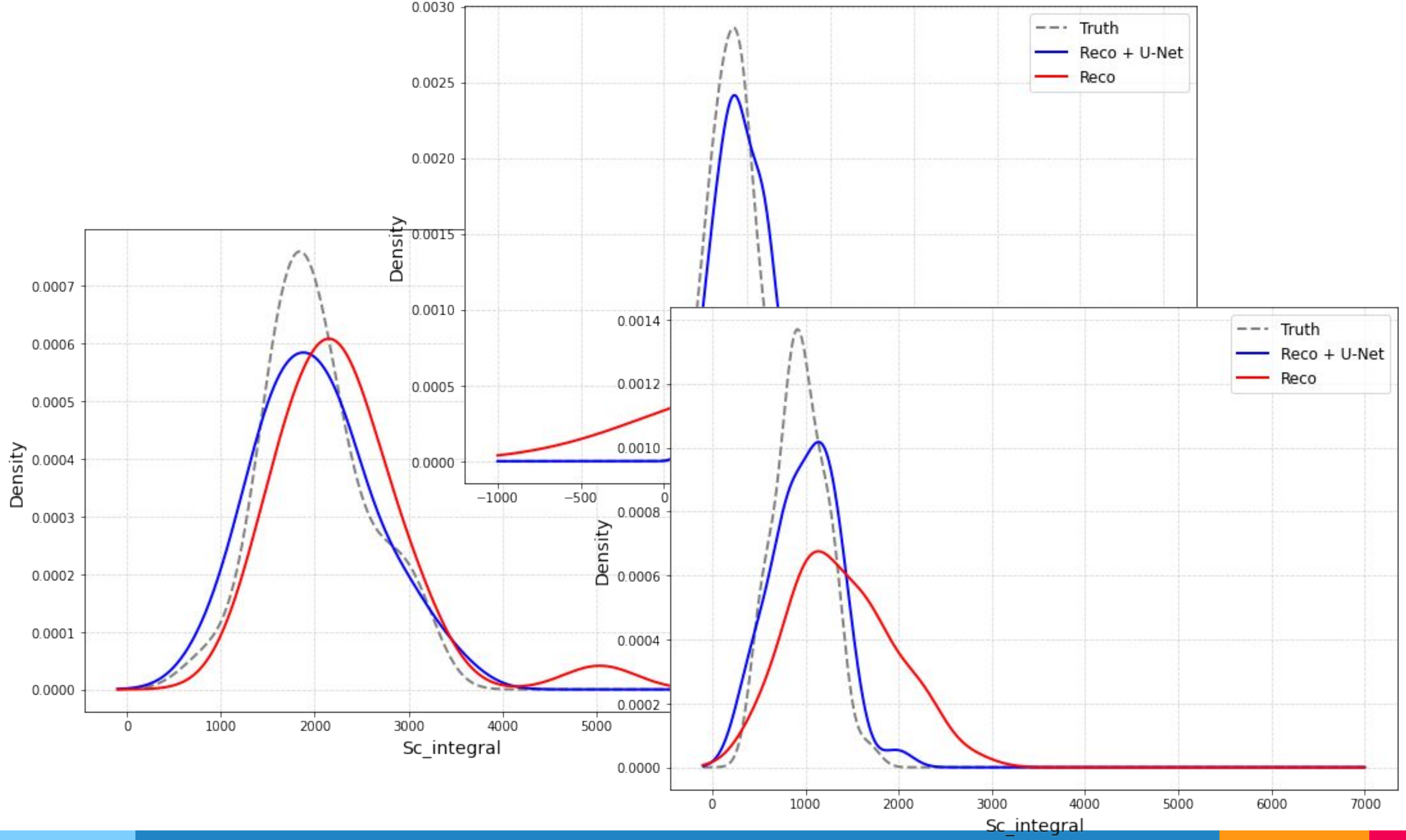
Next Steps

Check U-Net outputs when noise is changed

Evaluate U-Net + Reconstruction proposal in real data

Use U-Net features in different applications

Questions


Truth
Reco + U-Net
Reco
Density
Sc_integral
Truth
Reco + U-Net
Reco
Density
Density
Sc_integral