# David Morales

Evaluation exercise report for GSoC - ATLAS autoencoders

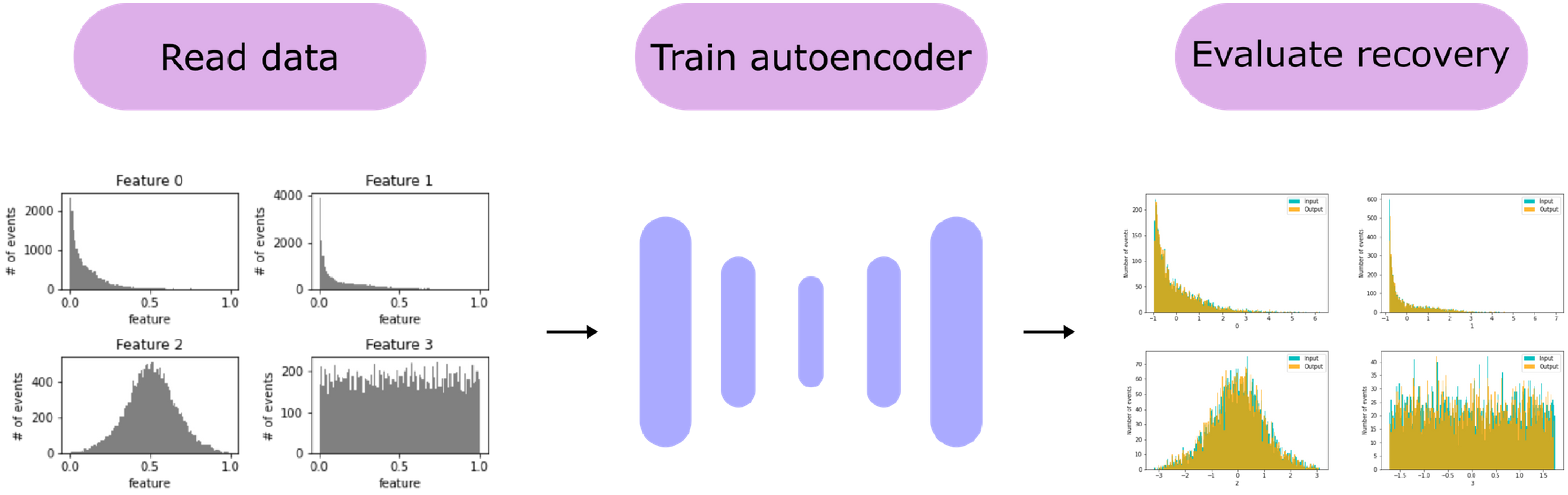
Google
Summer of Code

CERN

HSF
HEP Software Foundation

# Problem statement

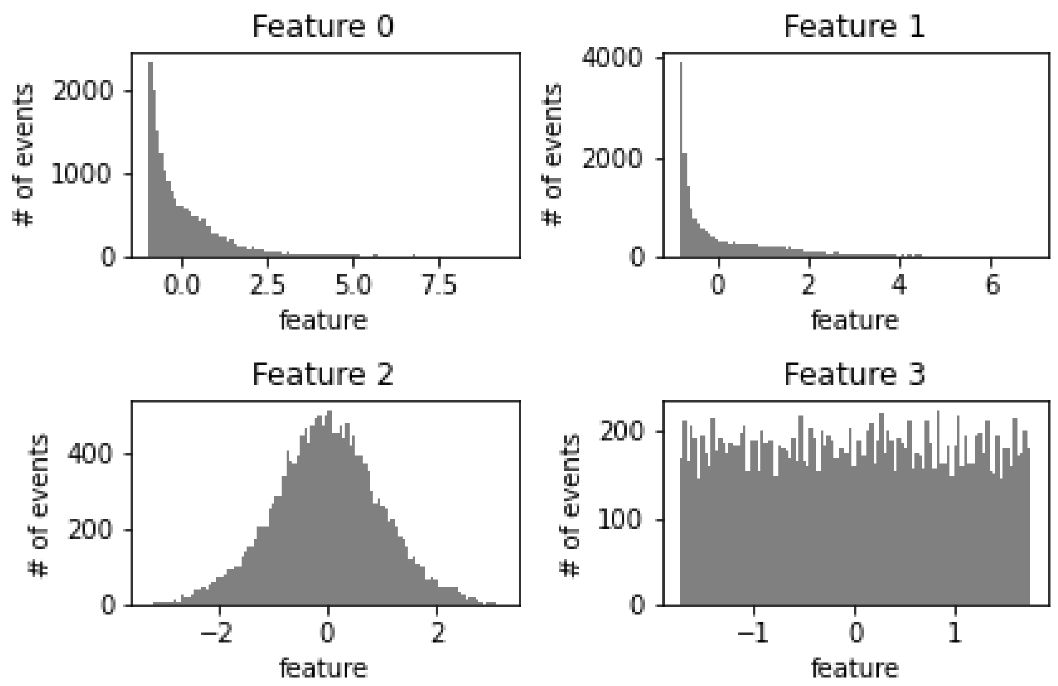
- Read and preprocess data
- Set up the autoencoder model
- Calculate metrics to evaluate the recovery of the compressed data



# Data preprocessing

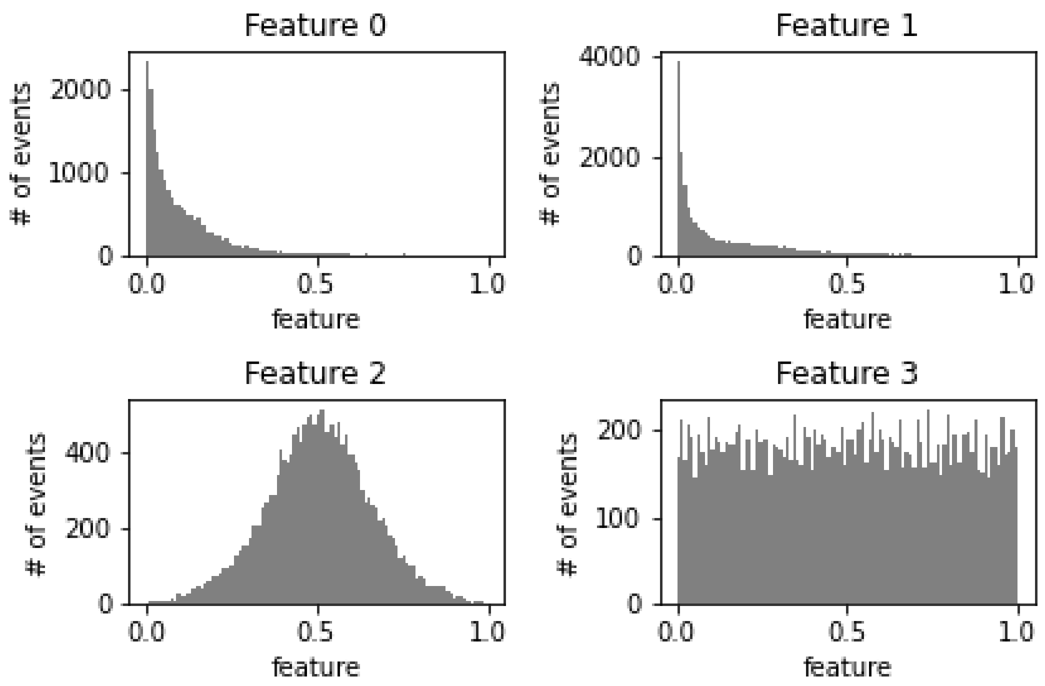
Standard normalization

$$\mathbf{z} = \frac{\mathbf{x} - \mu}{\sigma}$$




MinMax normalization

$$\mathbf{z} = \frac{\mathbf{x} - min(\mathbf{x})}{max(\mathbf{x}) - min(\mathbf{x})}$$

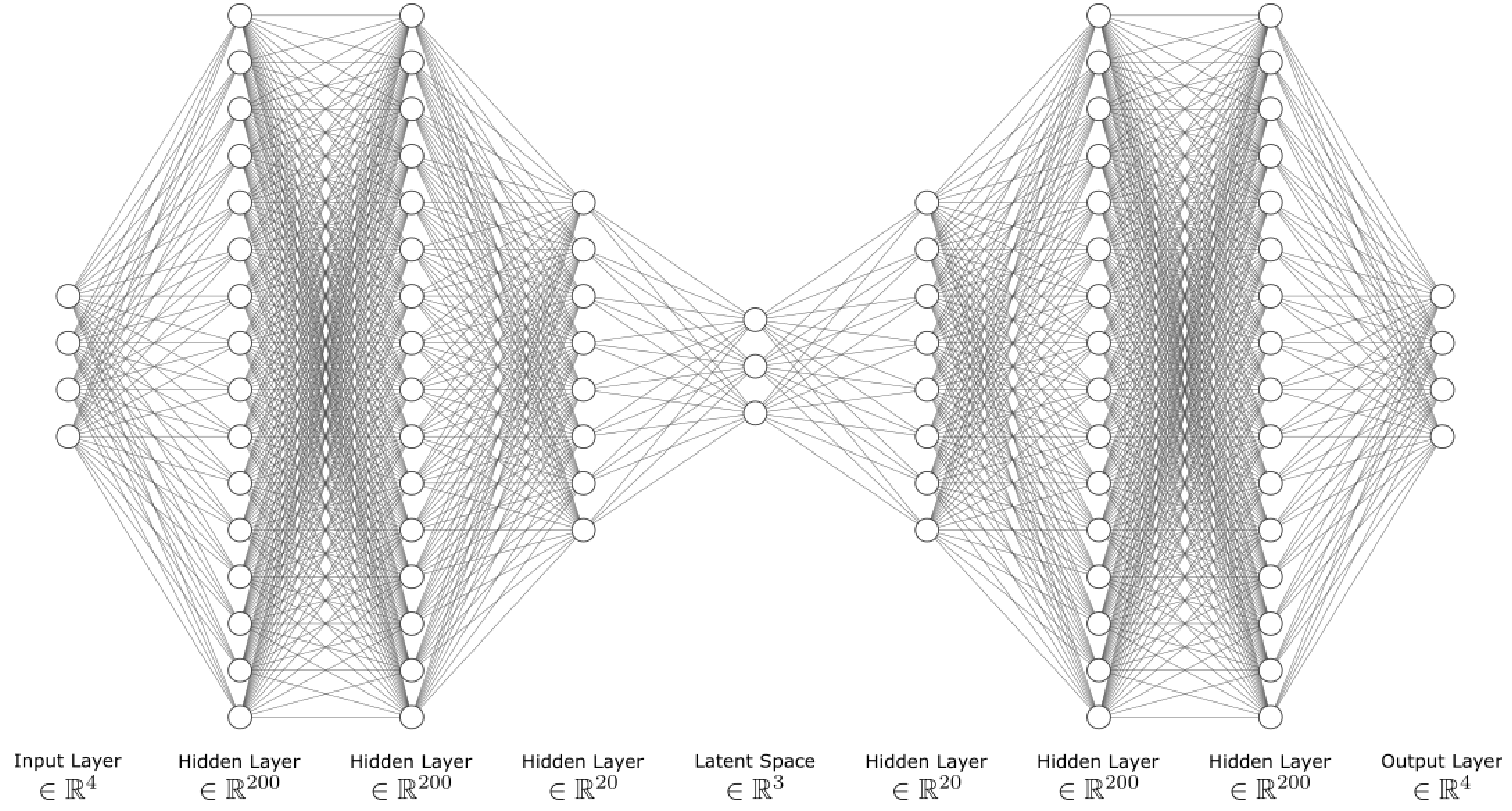



A normalization step is important to overcome the value range difference in the features and avoid the gradient explode problem in normalized data

I experimented with two types of normalization.

- Standard normalization
- MinMax normalization

# Autoencoder Model



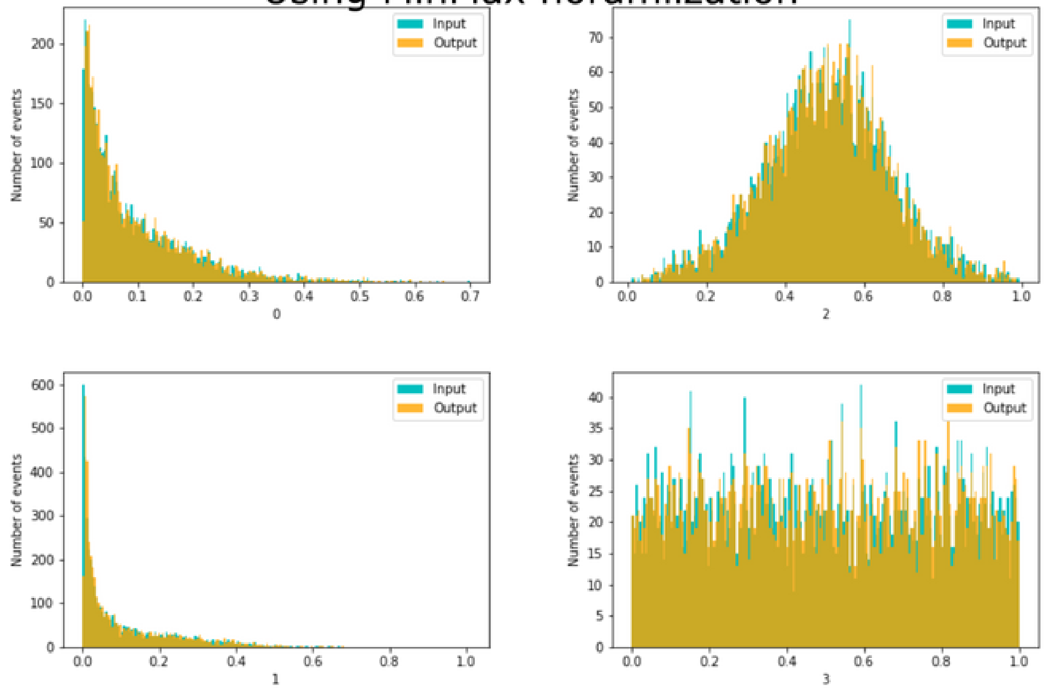

- The figure corresponds with the autoencoder model used in the training.
- A major architecture change is the use of Leaky Relu as the activation function on each layer instead of tanh.
- To evaluate performance, the MSE and PSNR metrics were calculated.

Loss function: $MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{x}_i - x_i)^2$

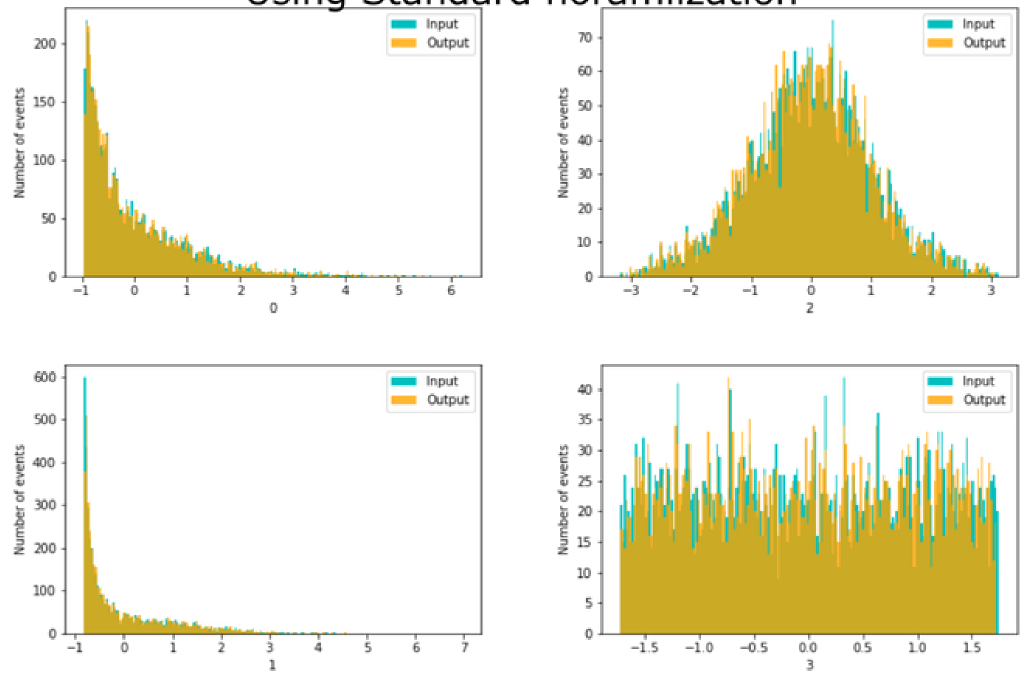
Aditional metric: $PSNR = 20log_{10}\left(\frac{MAX\_RANGE}{\sqrt{MSE}}\right)$

# Results

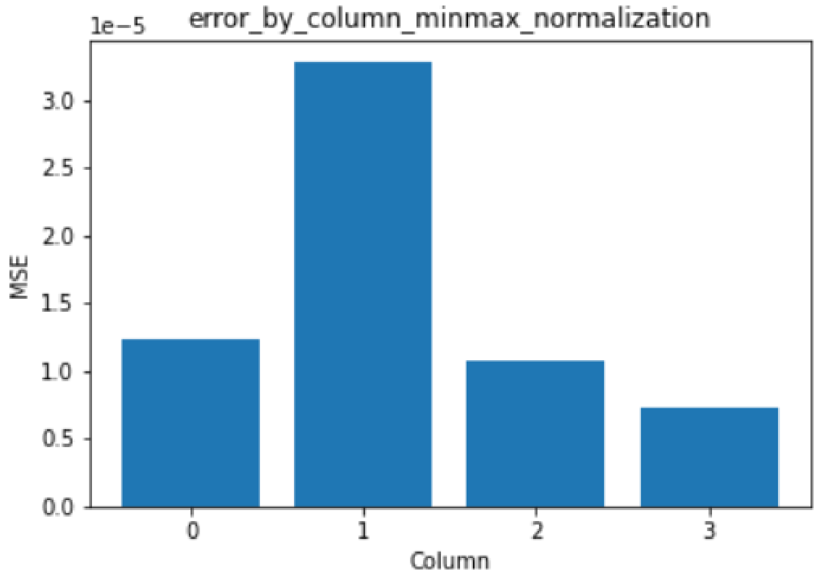
Histogram of each element in the 4D data
Using MinMax noramlization

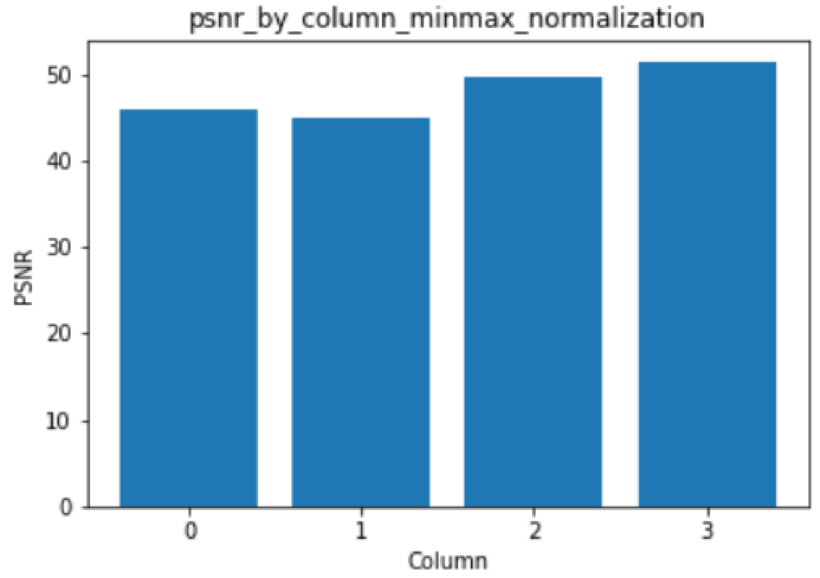



Histogram of each element in the 4D data
Using Standard noramlization

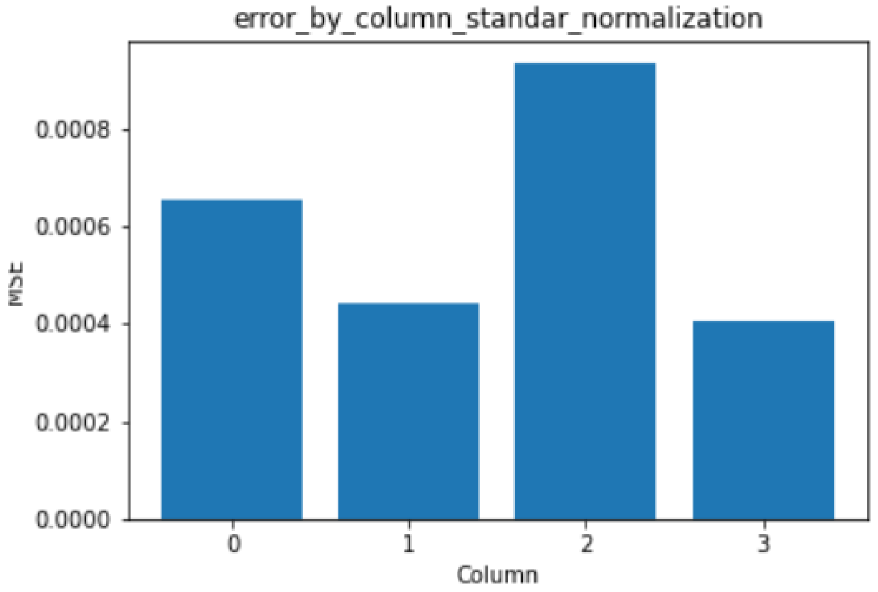


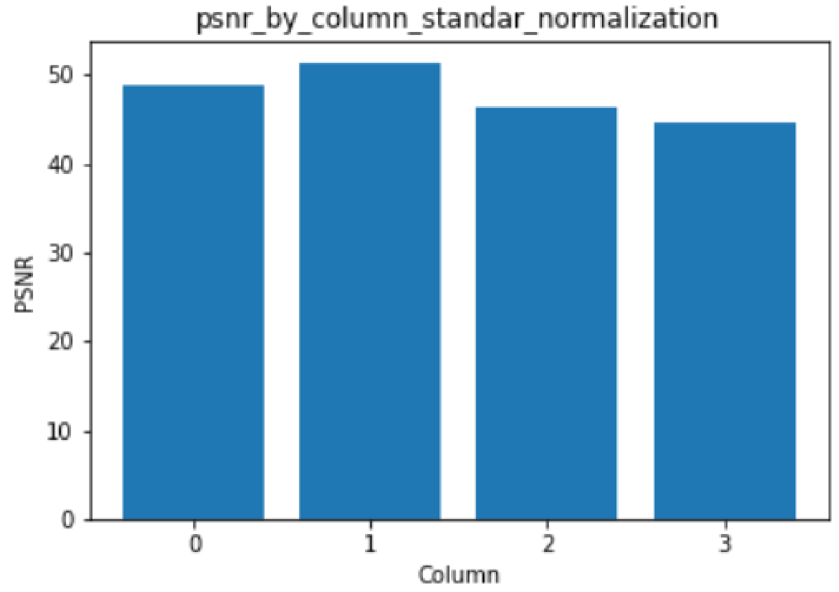

## Calculated MSE and PSNR over the test set with MinMax noramalization




## Calculated MSE and PSNR over the test set with standard noramalization



# Summary

- An autoencoder to compress the four-momentum of a sample of simulated particles from 4 to 3 variables was successfully implemented and evaluated using PSNR and MSE.
- Two different methods to normalized data were evaluated achieving outstanding results, therefore, any of these normalizations methods could be used to train the autoencoder.

Scan this QR code to access the repo