

# Integrating ML algorithms for LHC data compression into the ESCAPE Virtual Research Environment

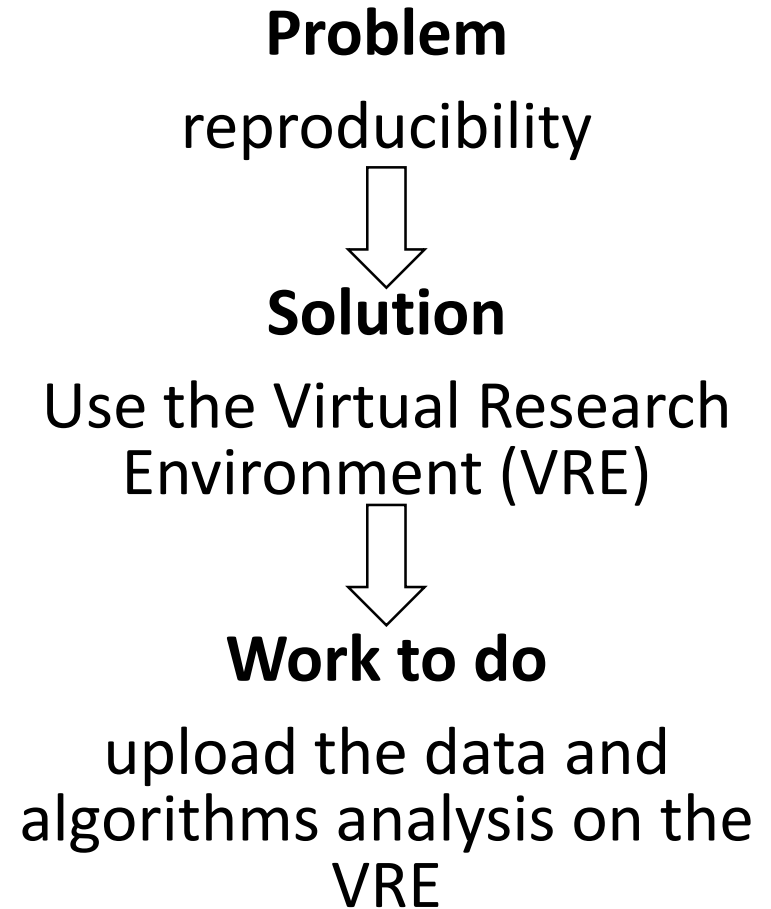
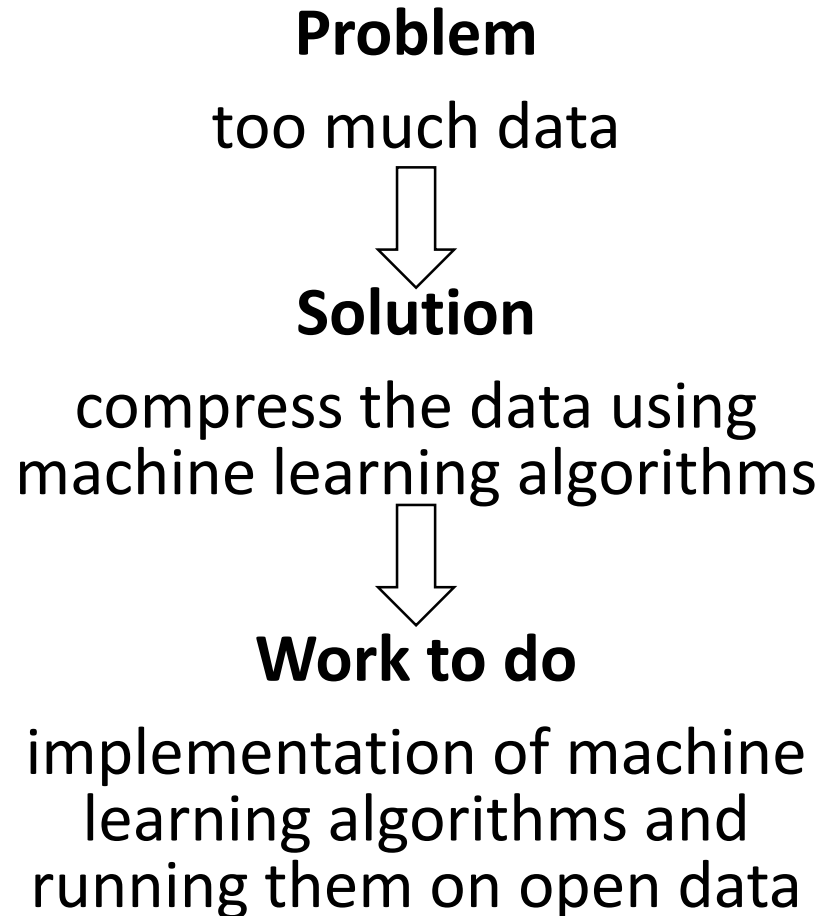
Mentor: Caterina Doglioni

# Background



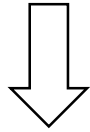
Image: CERN

# Motivation

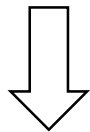


# LHC data compression

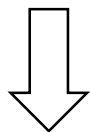
Open data from CMS experiment



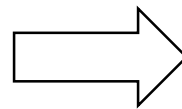
Preprocessing the data



Normalizing the data



Compressing the data



Measuring the performance

# Autoencoders

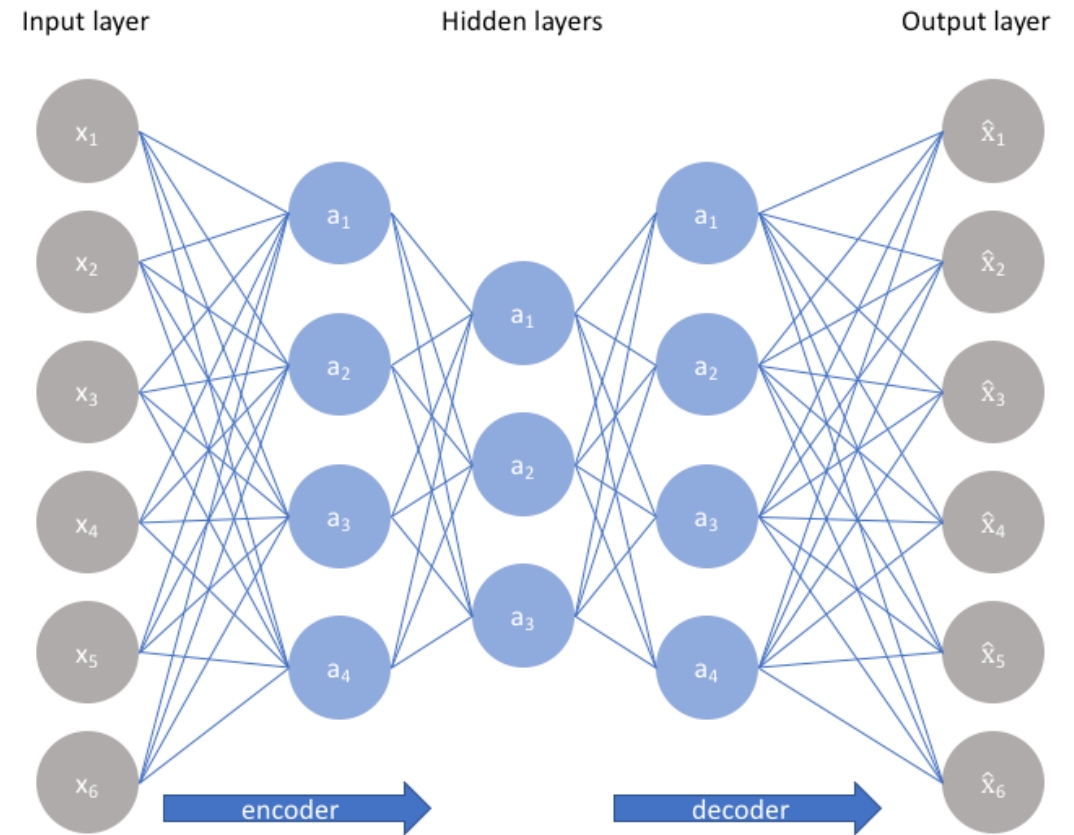
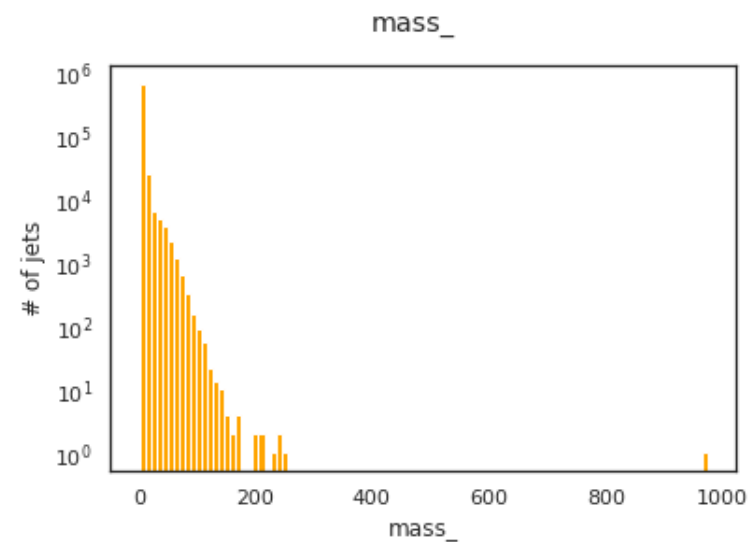
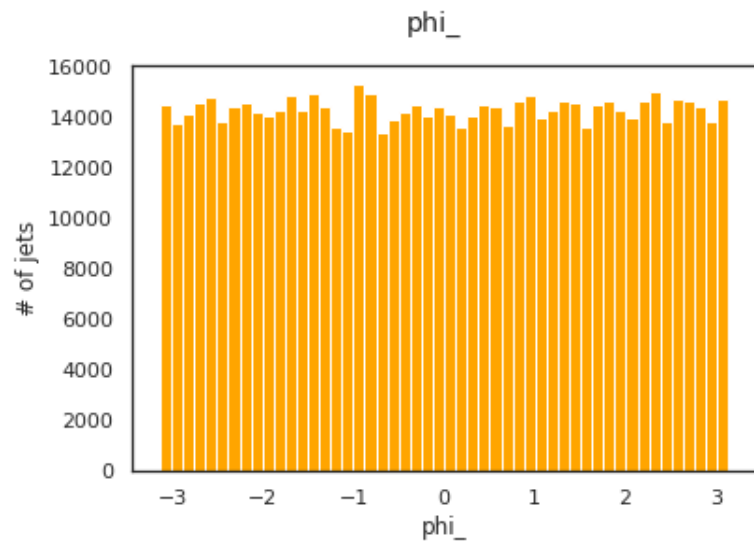
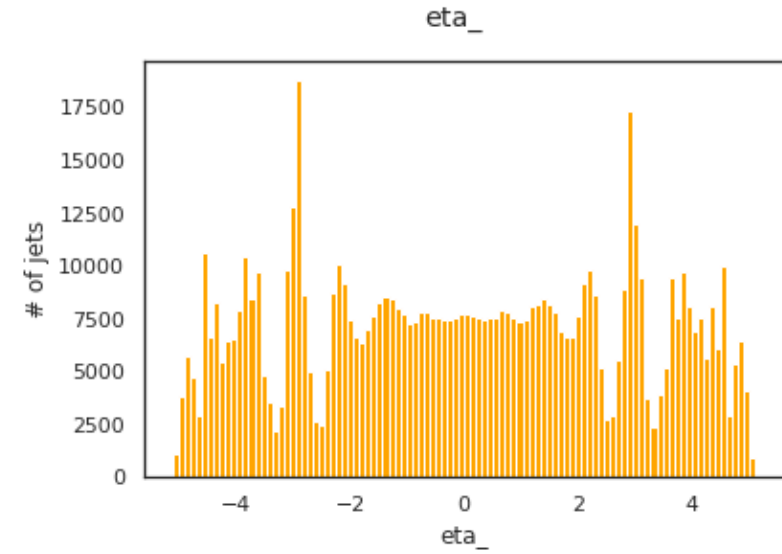
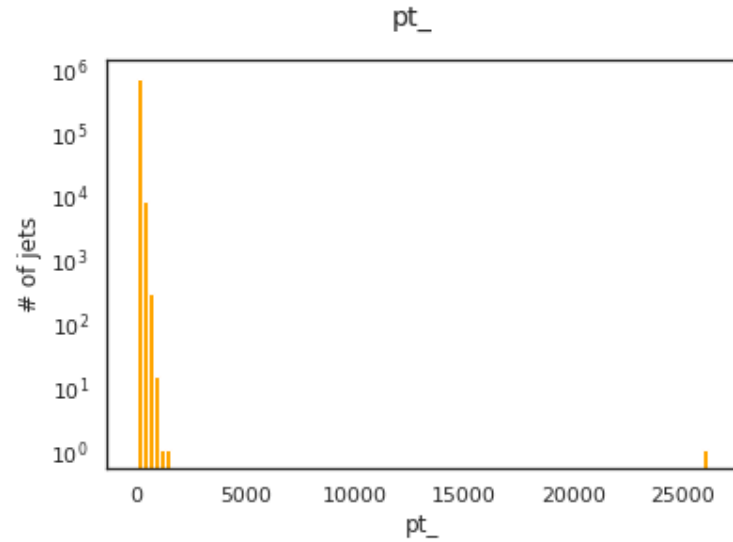


Image: <https://www.jeremyjordan.me/autoencoders/>

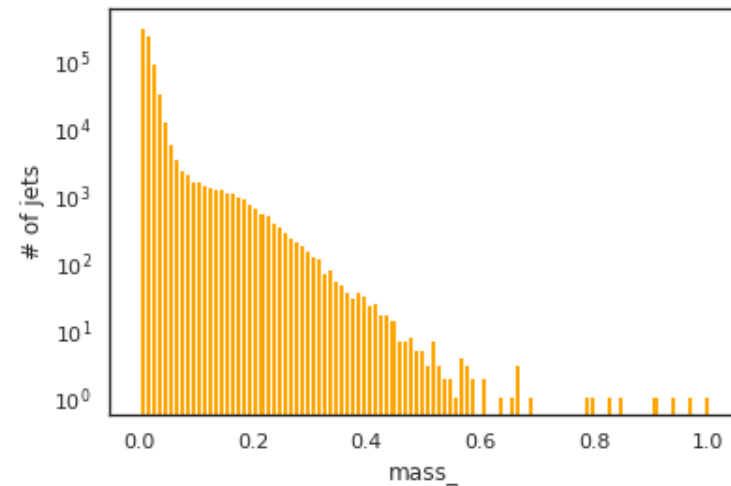
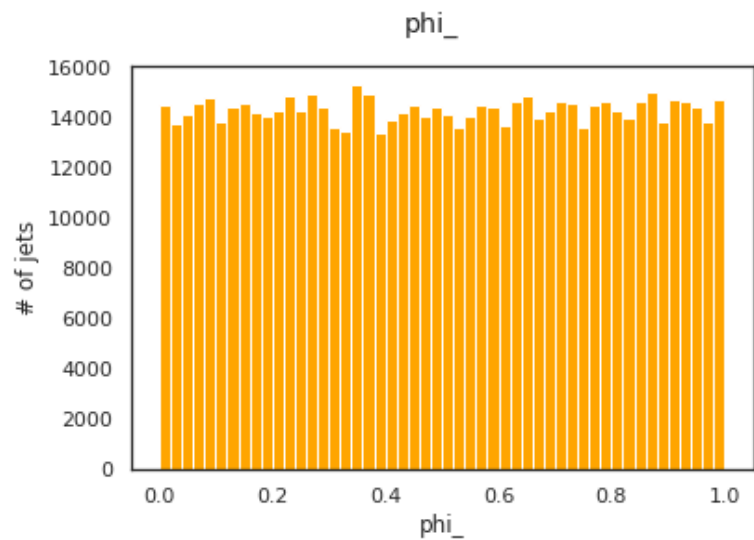
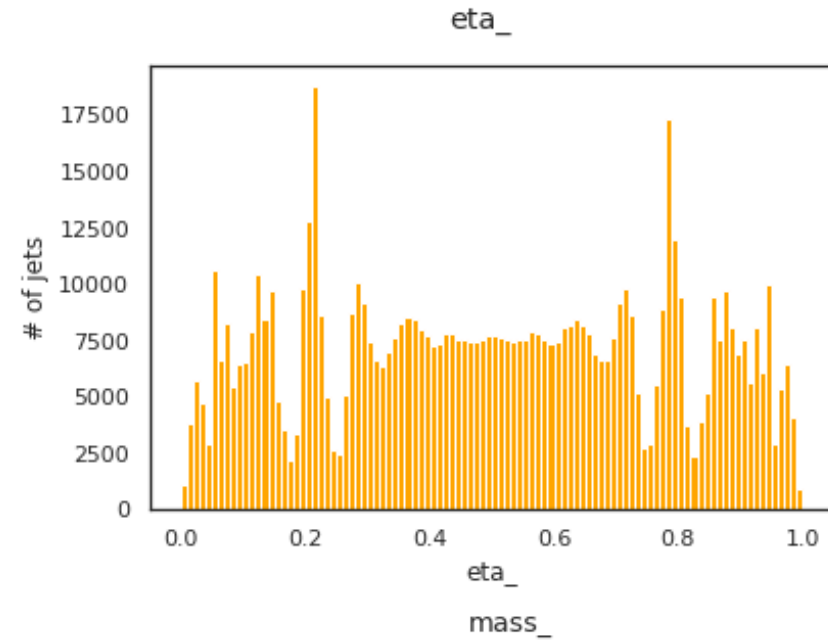
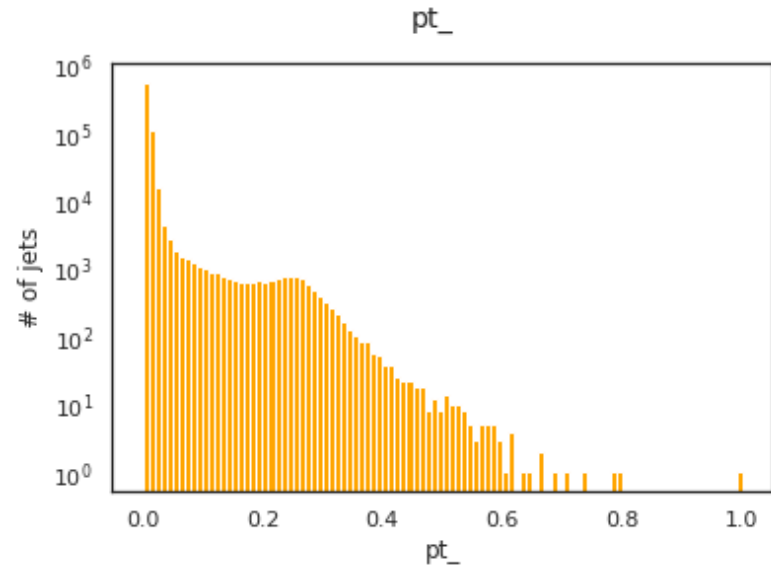


# Preprocessing the data

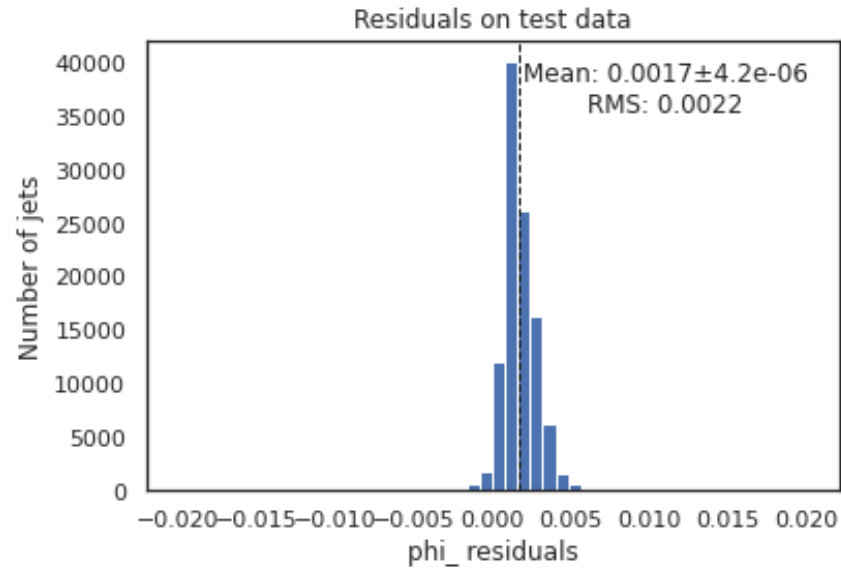


# Normalizing the data

Min-max normalization: 
$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

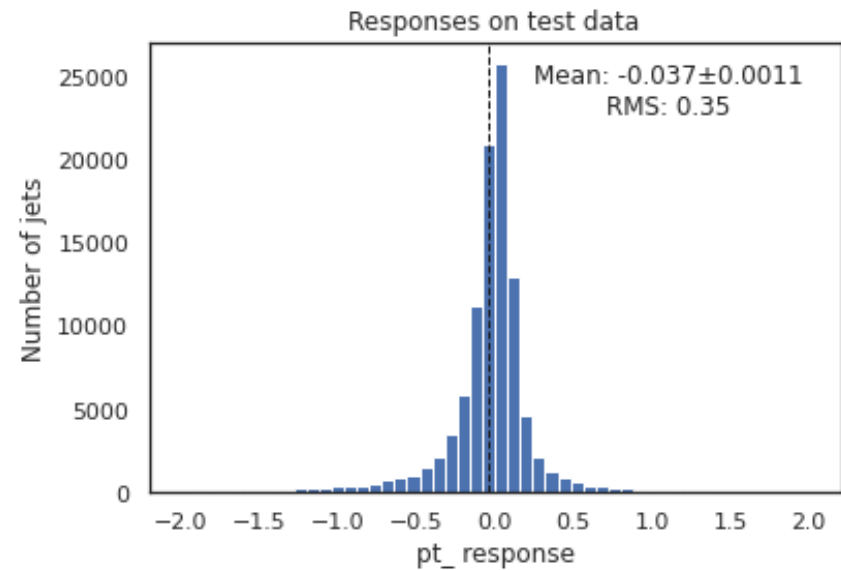
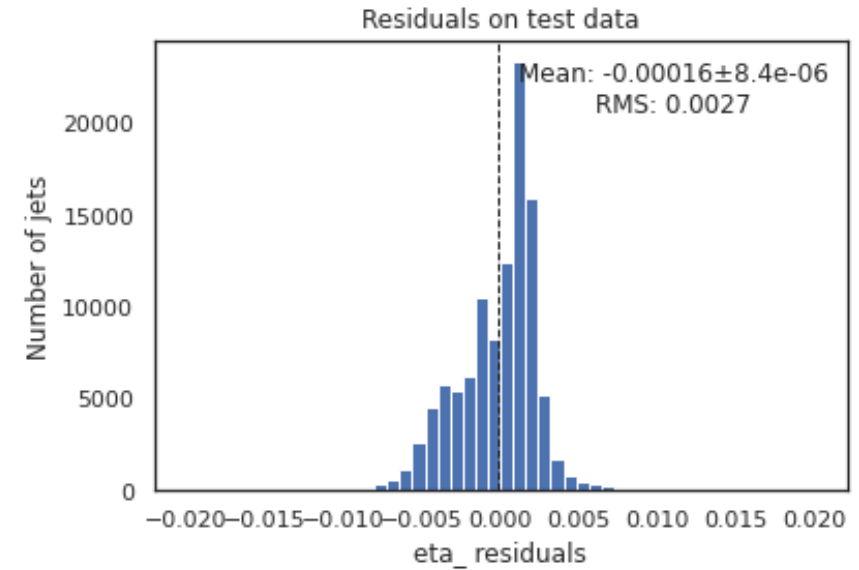


# Evaluation metrics



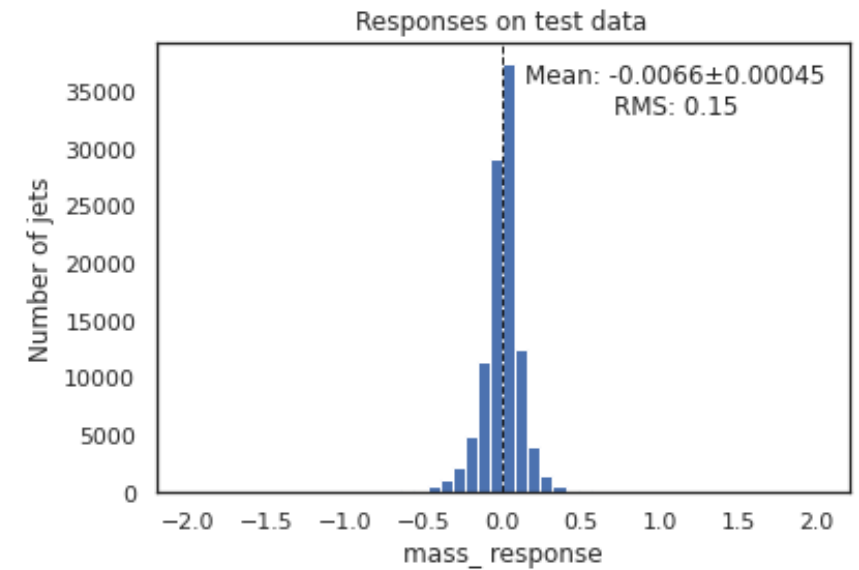
Residuals:

$$X_{in} - X_{out}$$



Responses:

$$\frac{X_{in} - X_{out}}{X_{in}}$$



# Compressing the data using ML algorithms

Previous work: Autoencoders implementation - George Dialektakis's studies

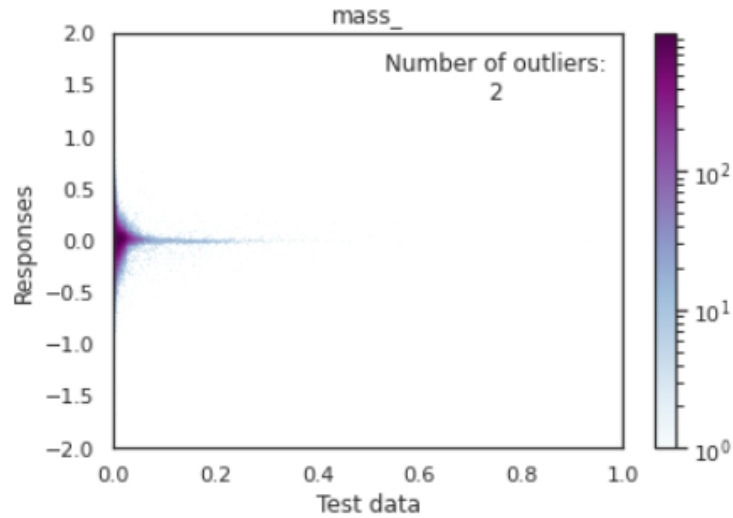
Original work: PCA implementation and comparison to Autoencoders, creation of a standalone and documented Jupyter notebook and uploading it on Virtual Research Environment (VRE)



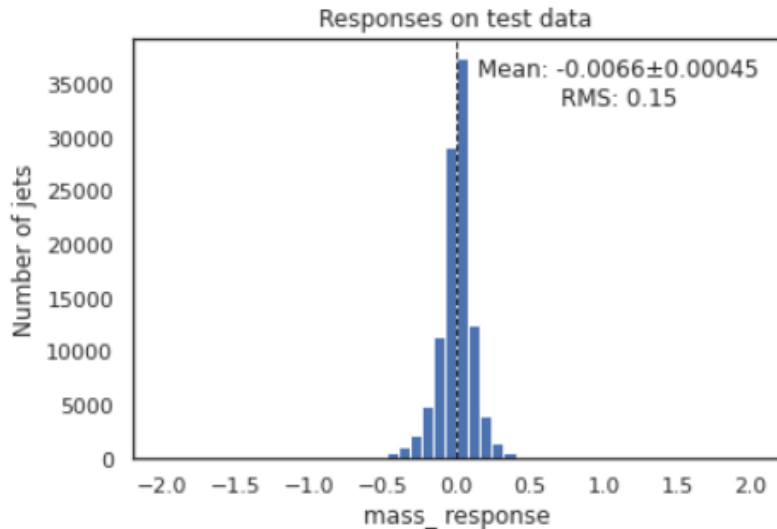
# PCA/Autoencoders comparison – 24D

The 2D histogram:

SAE 24D



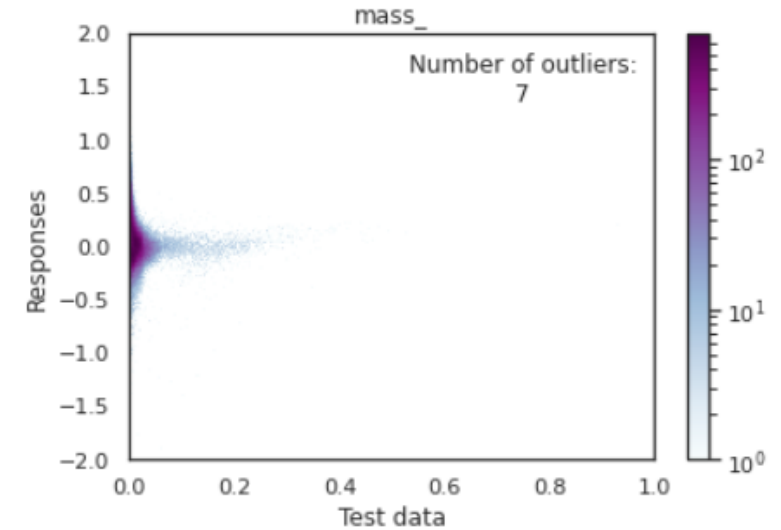
The histogram:



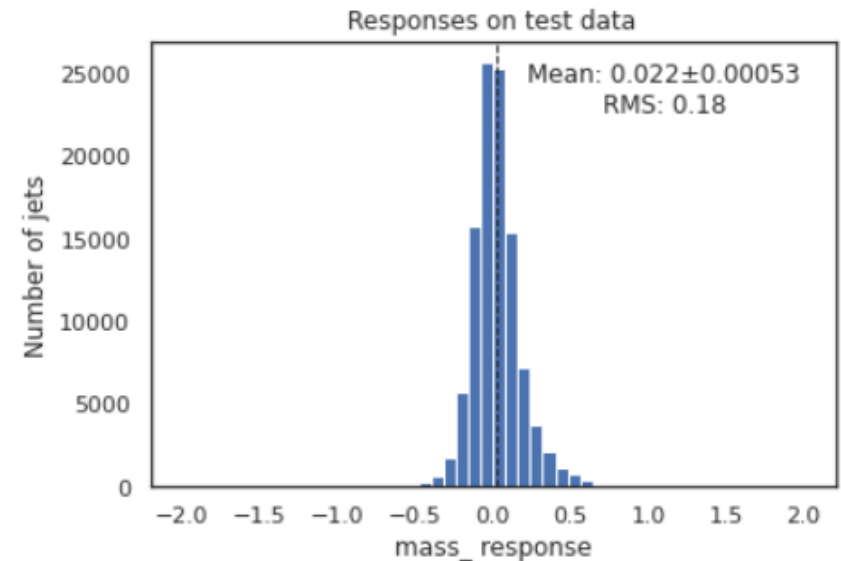
mass\_response

The 2D histogram:

PCA 24D



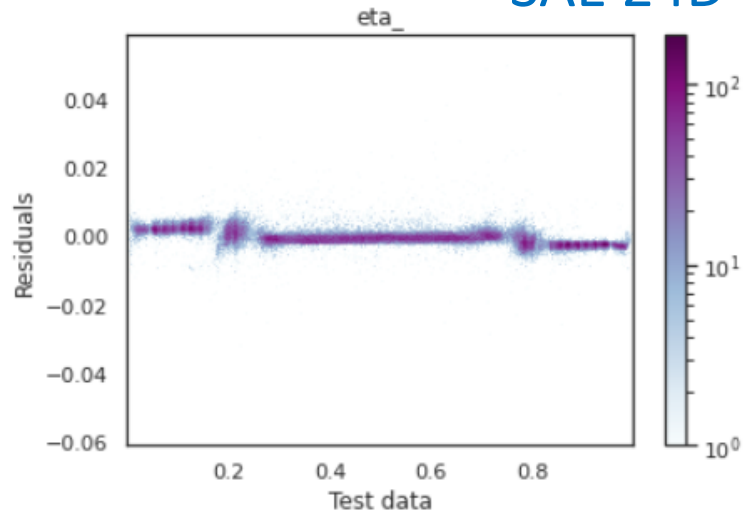
The histogram:



# PCA/Autoencoders comparison – 24D

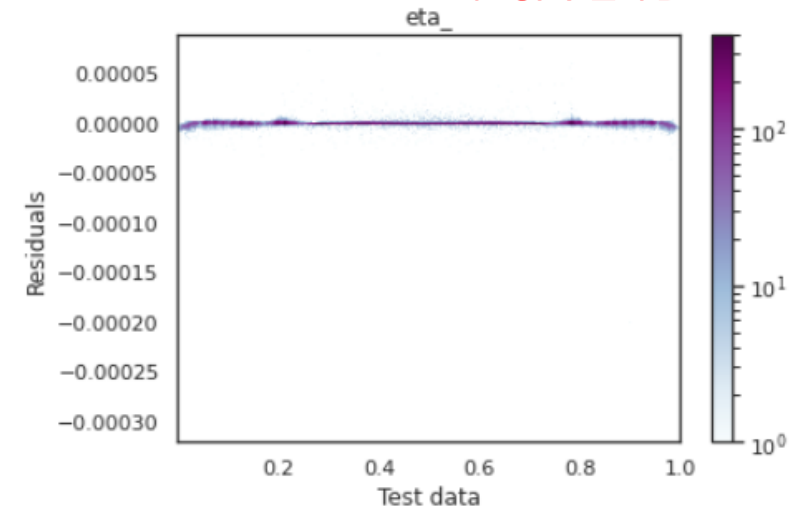
The 2D histogram:

SAE 24D

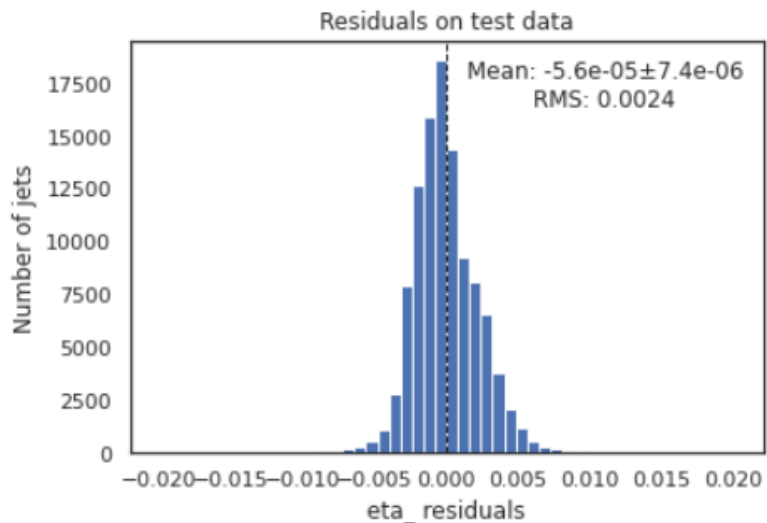


The 2D histogram:

PCA 24D



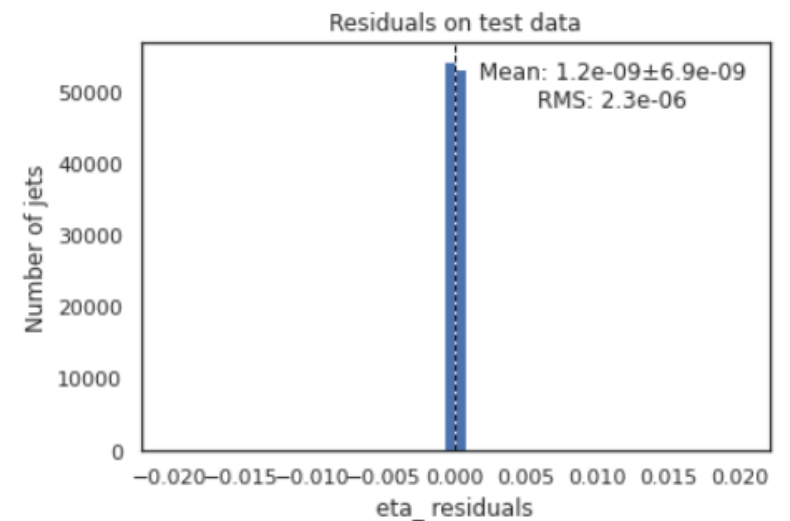
The histogram:



eta\_residuals

Residuals mean:  $-5.6e-05 \pm 7.4e-06$   
Residuals RMS: 0.0024

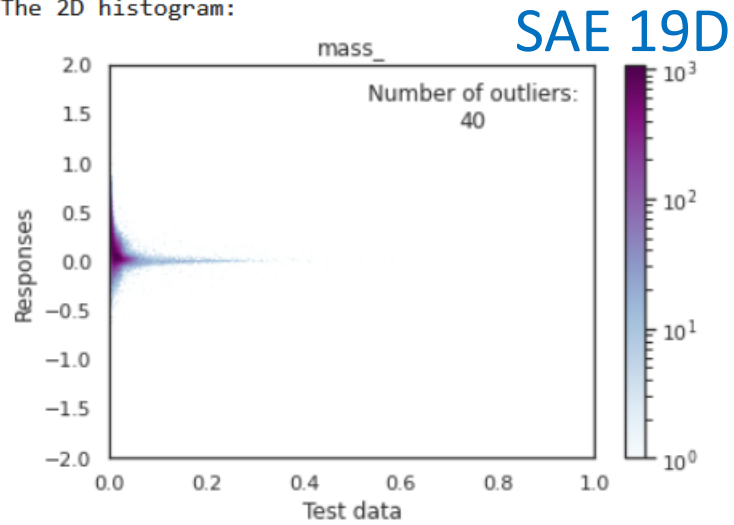
The histogram:



Residuals mean:  $1.2e-09 \pm 6.9e-09$   
Residuals RMS:  $2.3e-06$

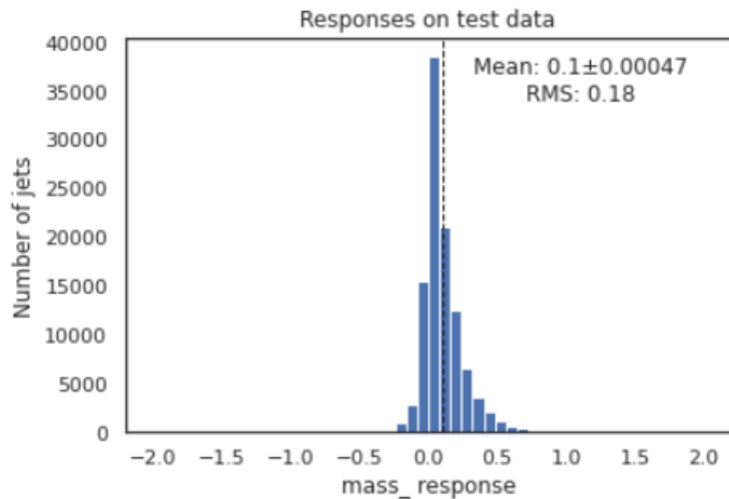
# PCA/Autoencoders comparison – 19D

The 2D histogram:



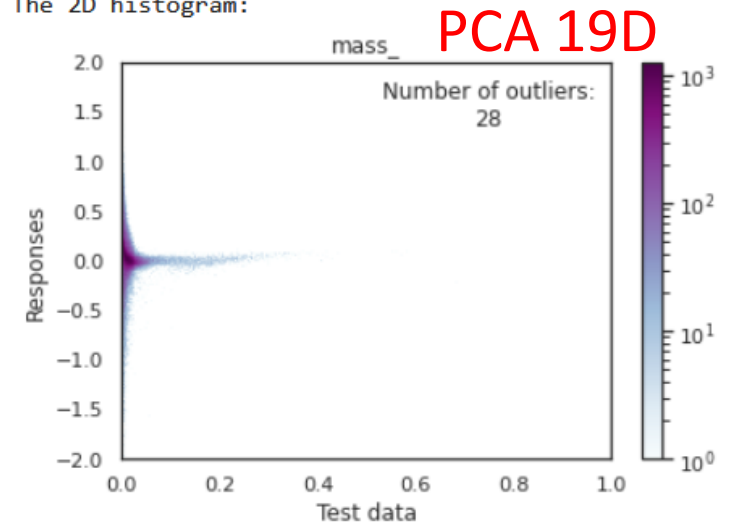
Number of responses: 107467

The histogram:



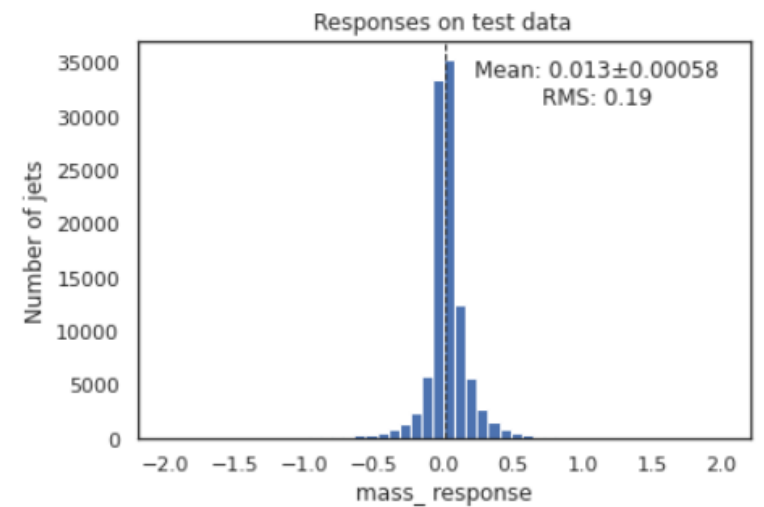
Response mean:  $0.1 \pm 0.00047$   
Response RMS: 0.18

The 2D histogram:



Number of responses: 107467

The histogram:



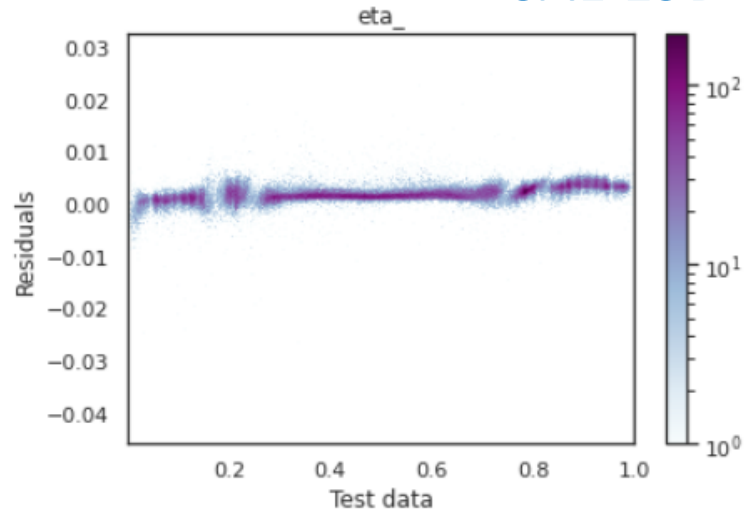
Response mean:  $0.013 \pm 0.00058$   
Response RMS: 0.19

mass\_response

# PCA/Autoencoders comparison – 19D

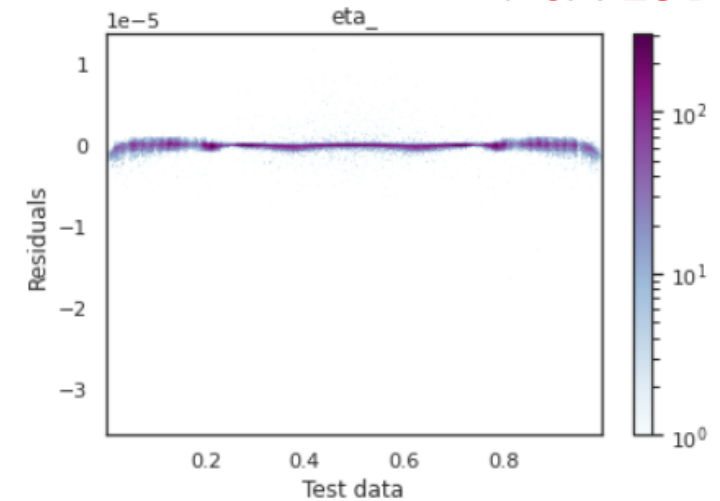
The 2D histogram:

SAE 19D

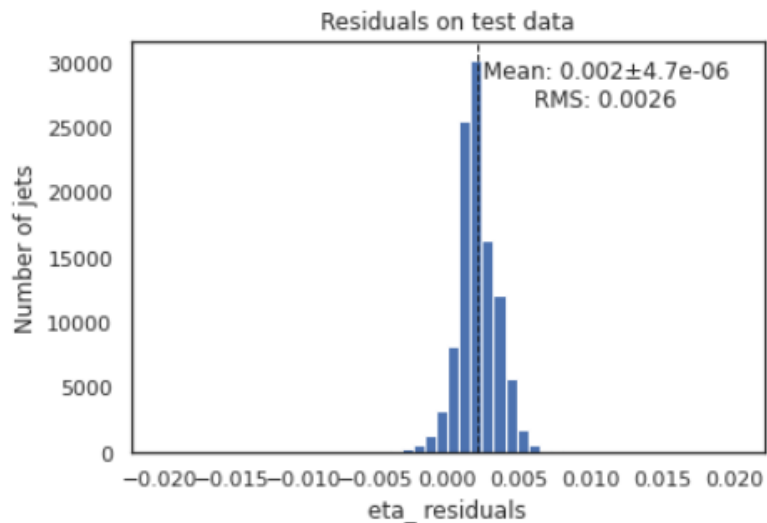


The 2D histogram:

PCA 19D



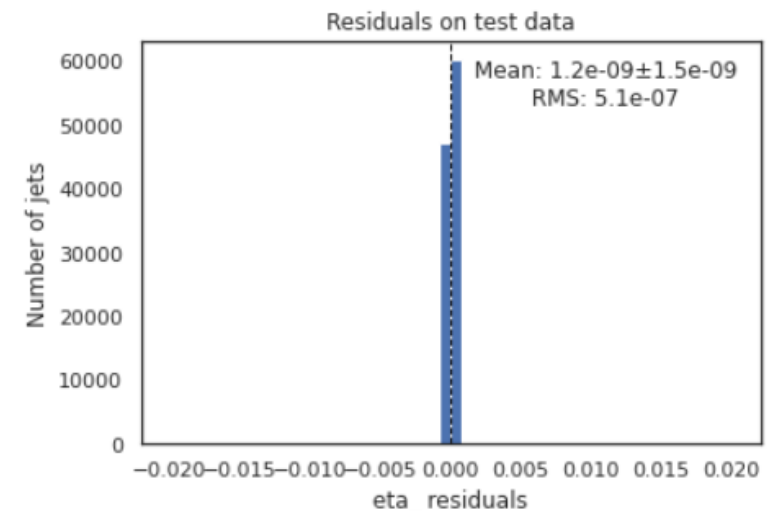
The histogram:



eta\_residuals

Residuals mean:  $0.002 \pm 4.7e-06$   
Residuals RMS: 0.0026

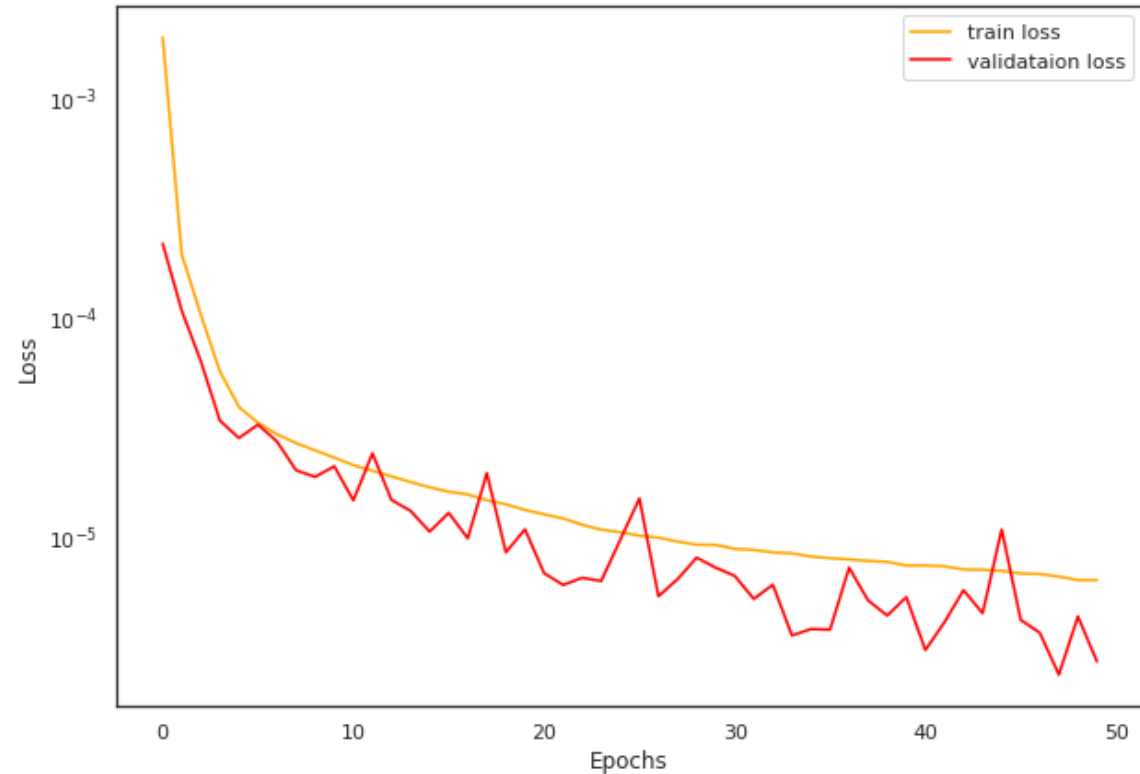
The histogram:



Residuals mean:  $1.2e-09 \pm 1.5e-09$   
Residuals RMS:  $5.1e-07$

# Further work

- Fit the response with a Gaussian to have a more precise idea of the performance
- Train the AEs for longer (That may lead for AEs to outperform PCA on 19D)
- Use the GPU distribution



# Rucio Data Management Service - VRE

← → ↻ https://escape-notebook.cern.ch/user/maxym/lab/tree/CMS-AE-TEST\_LAPP\_SP/Autoencoders\_and\_PCA\_v1.ipynb ☆

File Edit View Run Kernel Tabs Settings Help

**RUCIO**

EXPLORE NOTEBOOK

4S-AE-TEST\_LAPP\_SP:open\_cms\_dk

Search Everything

SEARCH RESULTS

- CMS-AE-TEST\_LAPP\_SP:open\_cms\_dk... 3.04GiB
- Available [Add to Notebook](#)

Autoencoders\_and\_PCA\_v1.ipynb Python 3

```
#Printing Mean and RMS on the histogram
text = "Mean: " + str(round_it(np.mean(response_norm),2)) + "±" + str(ME) + "\nRMS: " + str(round_it(RMS,2))
ax.text(0.75, 0.89, text, horizontalalignment='center',
        verticalalignment='center',
        transform = ax.transAxes)

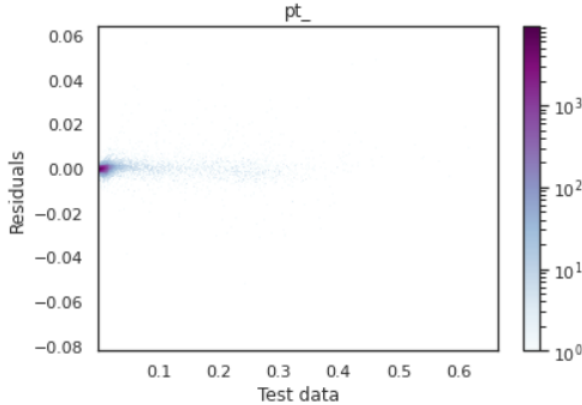
plt.show()

#Printing the evaluation metrics
print("Response mean:", round_it(np.mean(response_norm),2),"±",ME)
#print("Standard error of the mean:", ME)
print("Response RMS:", round_it(RMS,2), "\n")
```

To plot the residulas:

```
[98]: for kk in range(0, 4):
      plot_residuais(test_data[:,kk], reconstructed_data[:,kk], kk)
```

The 2D histogram:



Mode: Command Ln 1, Col 1 Autoencoders\_and\_PCA\_v1.ipynb



# Conclusions

- PCA was implemented and compared to Autoencoders – AEs show themselves better on 24D and PCA – on 19D and residuals (for now)
- Standalone documented Jupyter notebook was created and uploaded on Virtual Research Environment (VRE) – results of the project can now be easily reproduced by other researches

Thank you for attention!

Student: Maxym Naumchyk