Deep-compression for HEP data

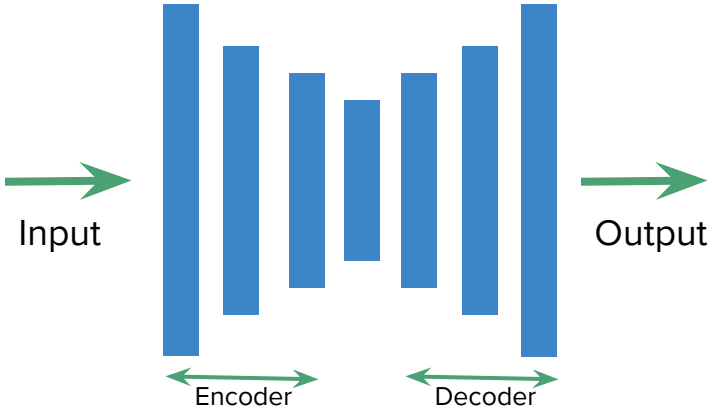
Honey Gupta (hn.gpt1@gmail.com)
Google Summer of Code 2020 & CERN - HSF

Motivation

- There are approximately 1.7 billion events occurring inside the ATLAS detector, each second.
- Storage of these events is limited by the event size and a reduction of the event size will allow for searches that were not previously possible.

Deep-compression

- Deep compression refers to usage of autoencoders for performing data compression.
- Learn the data distribution by projecting it to a lower-dimension and then reprojecting.
- The idea is to use deep compression for High Energy Physics (HEP) data and check their efficacy.



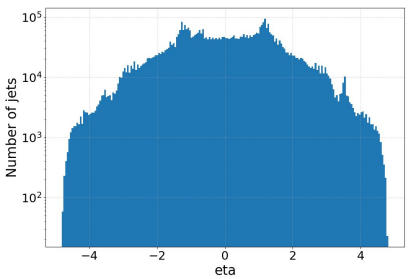
A typical autoencoder (encoder+decoder) network

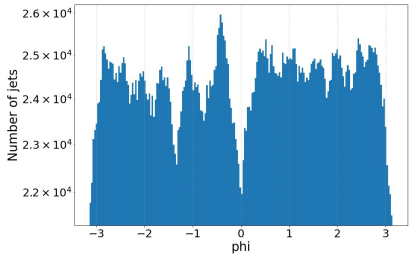
Phase 1: Validation of existing network

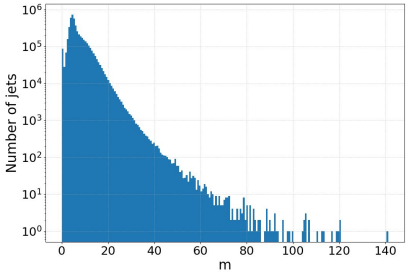
- **Analyse the available data**
  - plot the distribution of each variable
  - compare the plots with the plots mentioned in prior experiments (Eric Wullf's thesis, a Masters student who worked on the project earlier)
- **Test the available pre-trained model**
  - create plots from the pre-trained model
  - compare and validate the published results
- **Train the model on the available data**
  - create response and correlation plots
  - analyse the performance

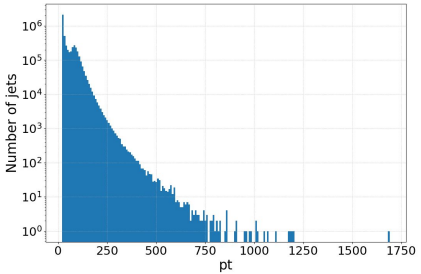
1. Data distribution - 27D

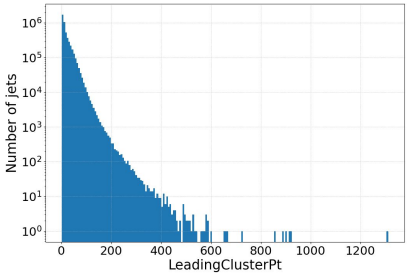
We visualize the data distribution for few variables in the training set

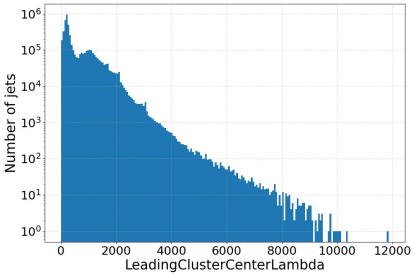


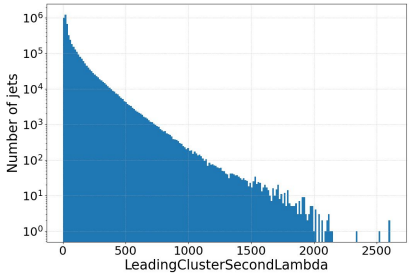


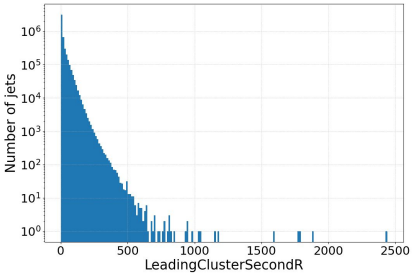












2: Comparison with existing results

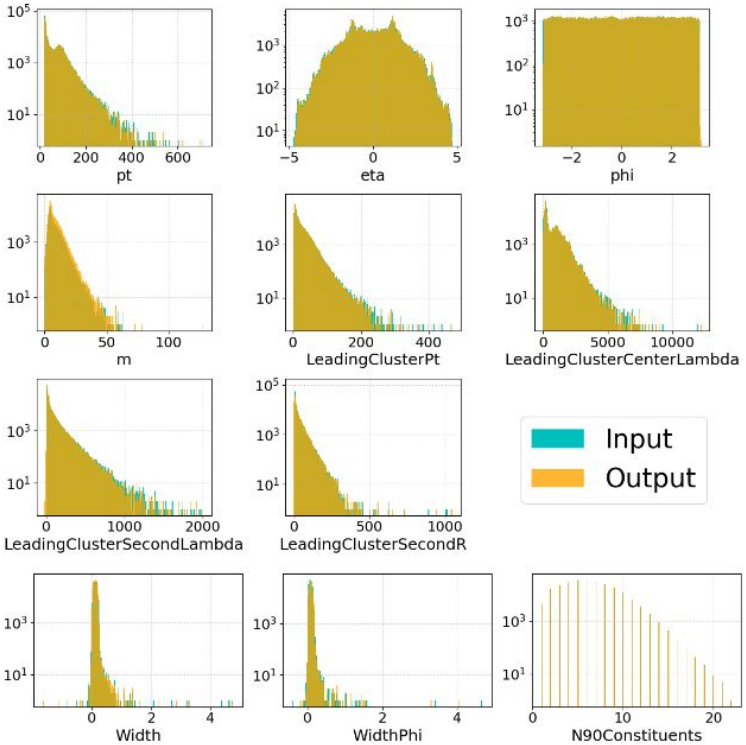
Model details for the available results

- LeakyReLU, BN
- Custom-norm, 27D data
- Latent space = 14
- Model
  - 27-200-200-200-14-200-200-200-27

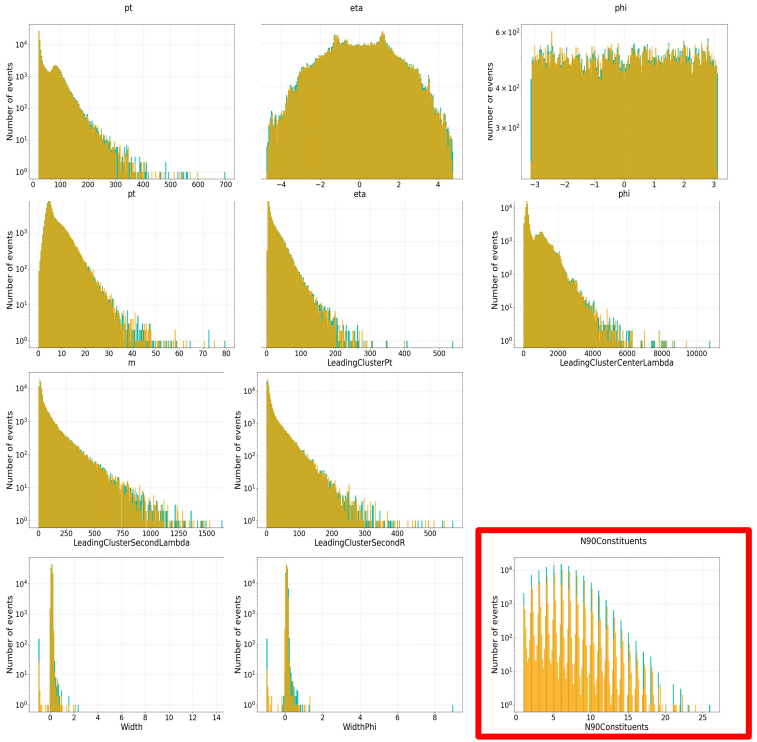
Model details for the available pre-trained model

- LeakyReLU, BN
- Custom-norm, 27D data
- Latent space = 20
- Model
  - 27-200-200-200-20-200-200-200-27

Plots from the available results



Plots for the results from the pre-trained model



Observations:

- Performance of the available pretrained seem to be very similar to the existing results

3. Re-training the network on 27D data

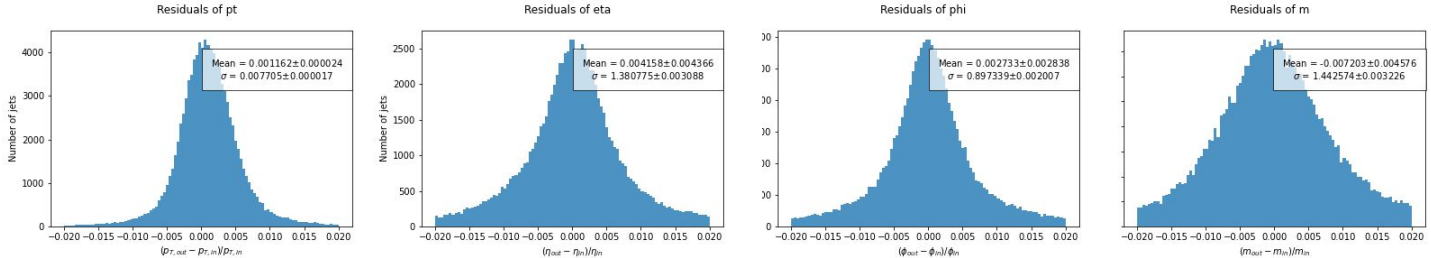
Model details:

- LeakyReLU, BN
- Custom-norm
- Latent space = 20
- Model - [27, 400, 400, 200, 20, 200, 400, 400, 27]

`

MSE on test-set = 7.844e-06

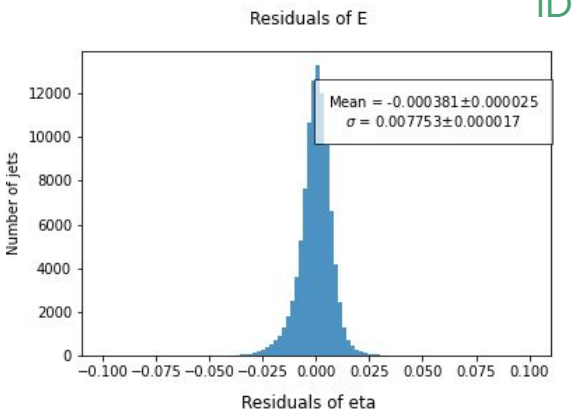
1D response plots for the retrained model

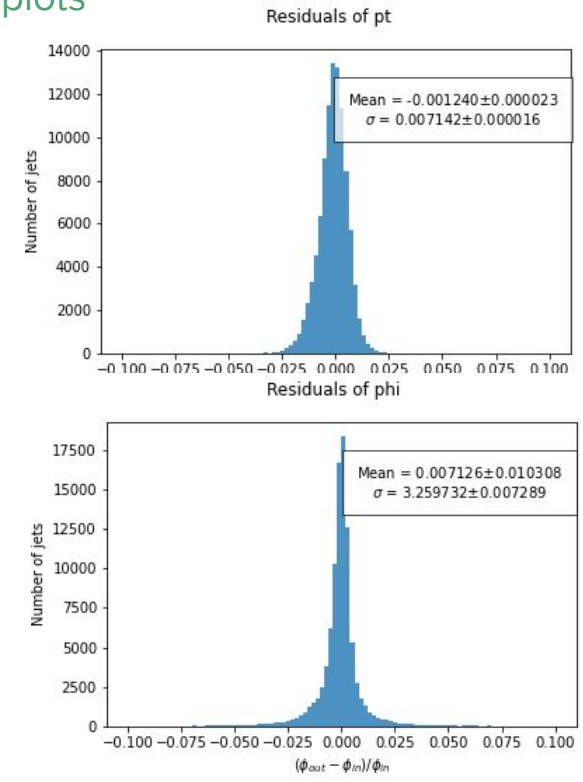


Phase 2: Expansion to event-level data

Phase 2a: Training on processes having jet particles in majority: from PhenoML dataset

1D response plots

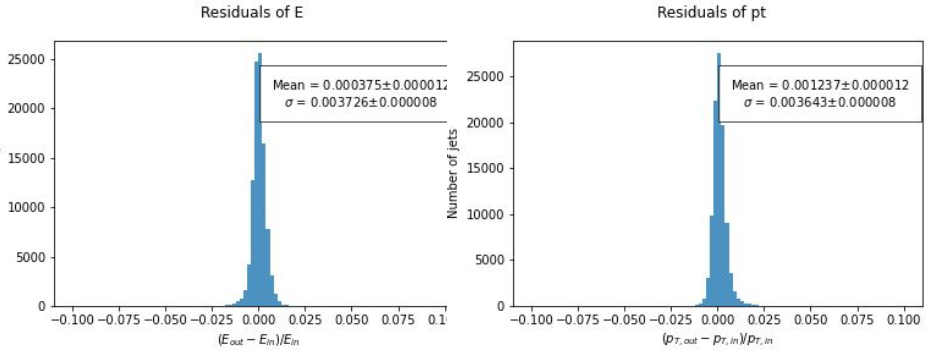




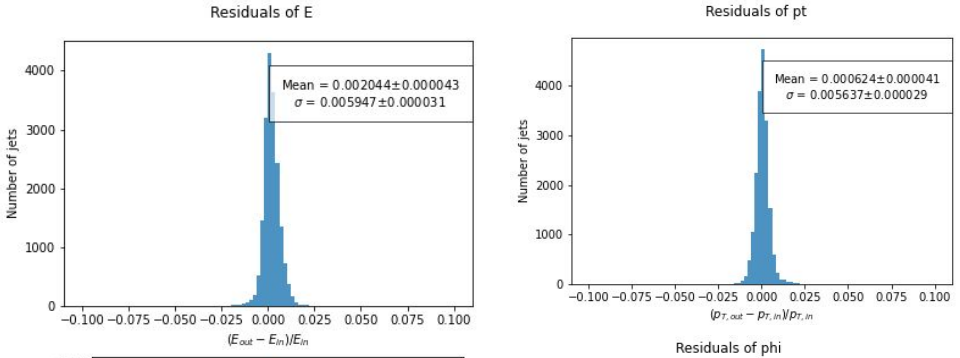
2b: Testing a jets-trained-model on ‘other’ particles - combined

Test with atop_10fb data

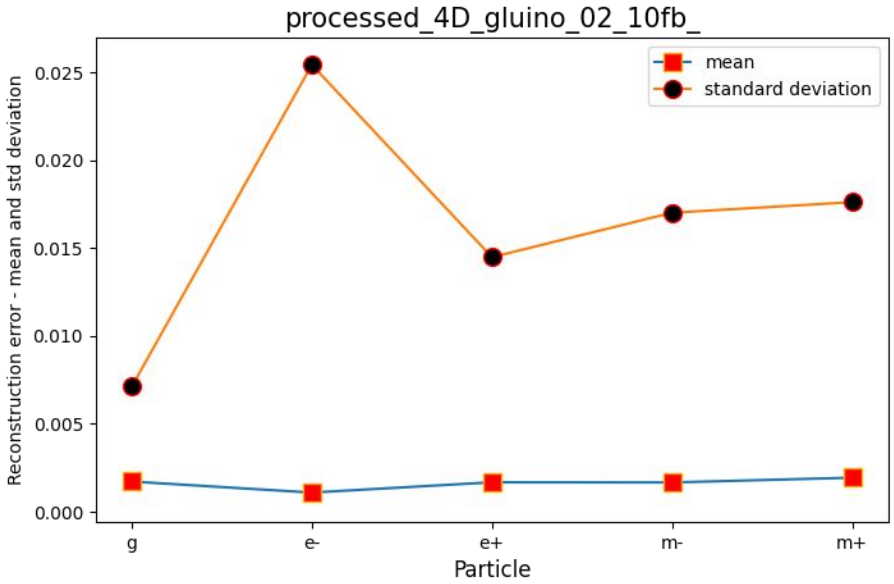
Jets data with custom norm

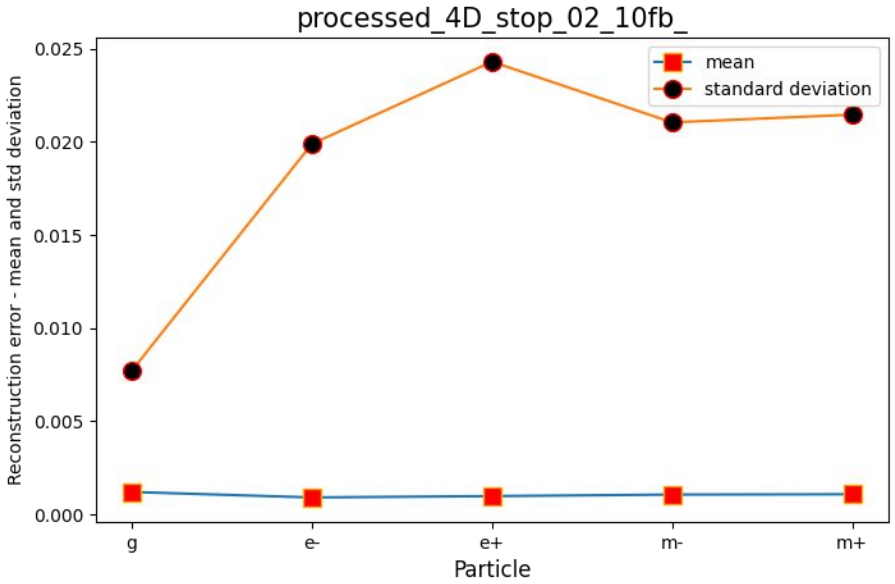


atop data with custom norm



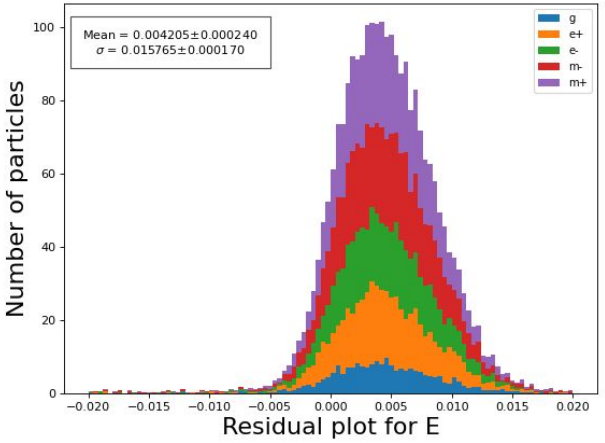
2c: Testing a jets-trained-model on ‘other’ particles individually



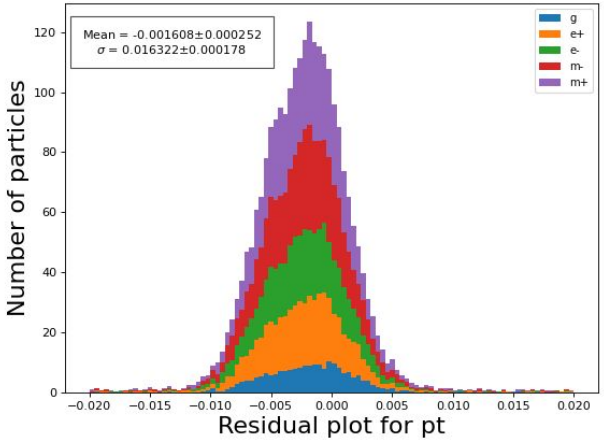


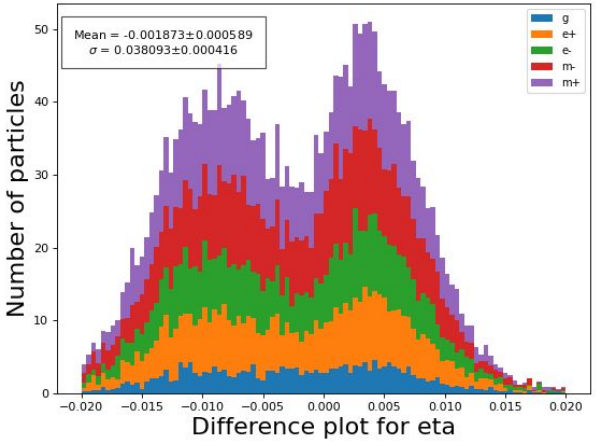
Mean and std-dev for the residuals of pt

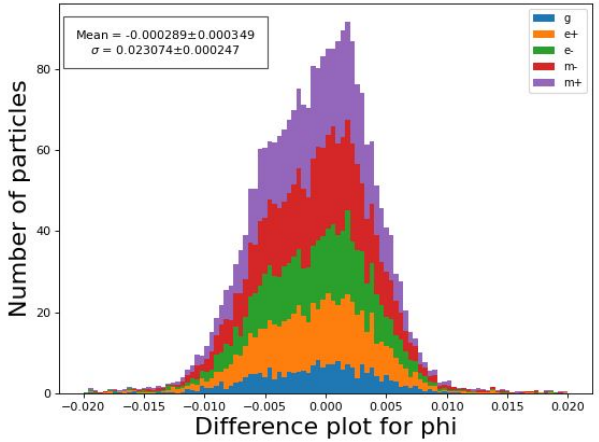
2c: Testing a jets-trained-model on ‘other’ particles individually



gluino_02







Conclusion

- Autoencoder model for compression, trained on jets, works well on other particles
- Highlights the feasibility of using a deep autoencoder for compression of processes containing a mix of particles.

Projects Artifacts:

- Code - GitHub
- Report
- Detailed slides
- Documentation and worklog - Zenodo

For queries, contact: hn.gpt1@gmail.com

Thank you!