Jet Response Prediction Using Jet Images

**Alexx Perloff (TAMU), Andrew Whitbeck (FNAL), Javier Duarte (FNAL),
Jean-Roch Vilimant (CalTech), Maurizio Pierini (CERN),
Raghav Kunnawalkam Elayavalli (WSU), Rohan Bhandari (UCSB)**

12th Dec 2017

Machine Learning for Jet Physics

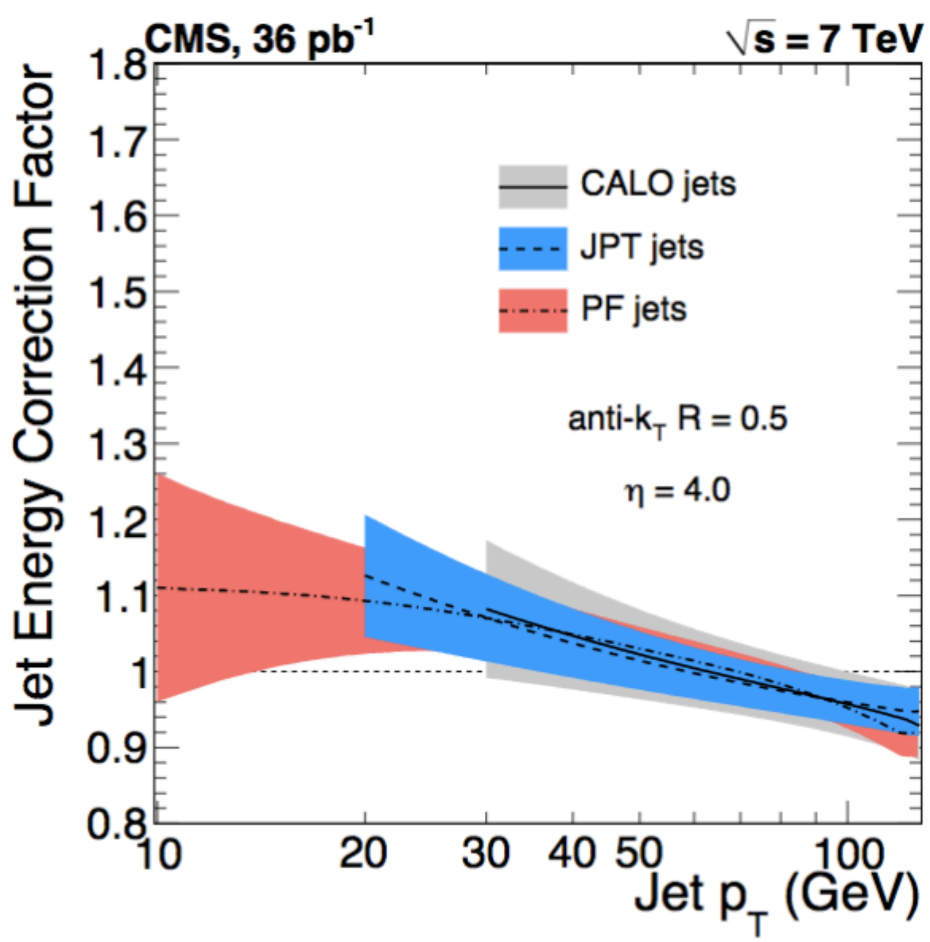
11-13 December 2017
Lawrence Berkeley National Laboratory
US/Pacific timezone

Detector Level Correction

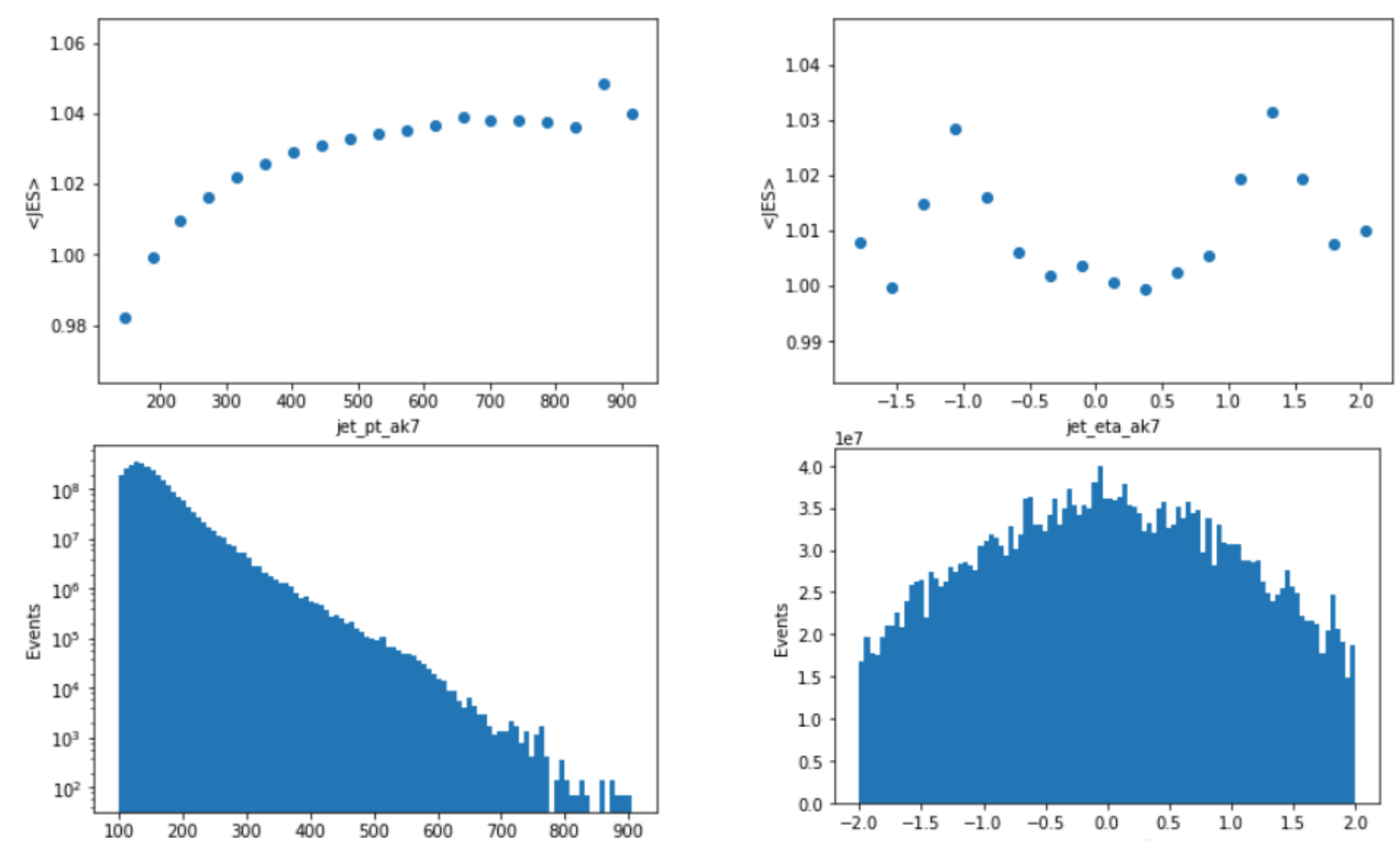
Jet Energy Correction necessary to correct for detector response

JEC as function of pT and eta available in the CMS QCD open data samples

JINST 12 (2017) P02014



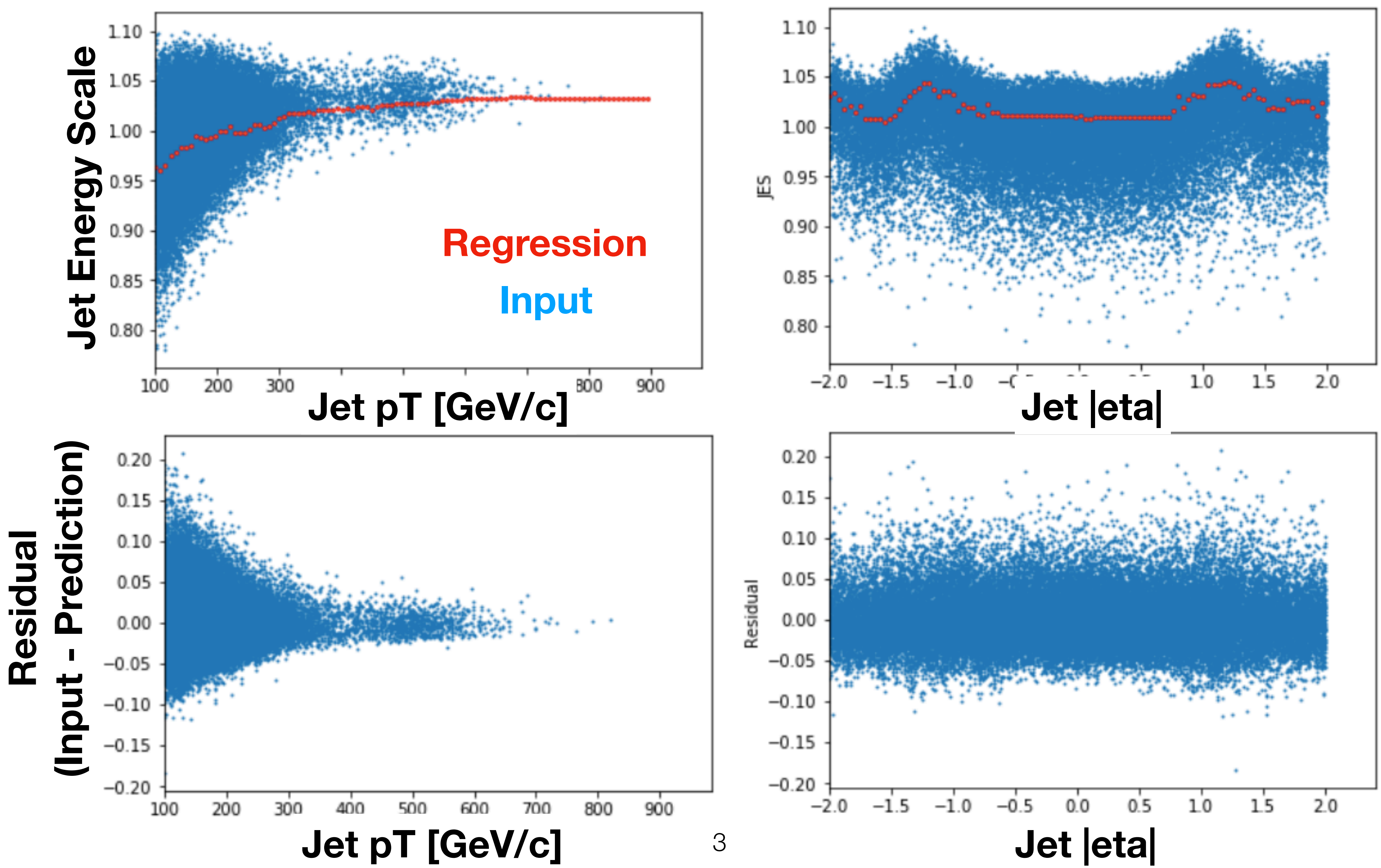
anti-k$_t$ R = 0.7 PF Jets



Jet p$_T$ [GeV/c] **Jet η**

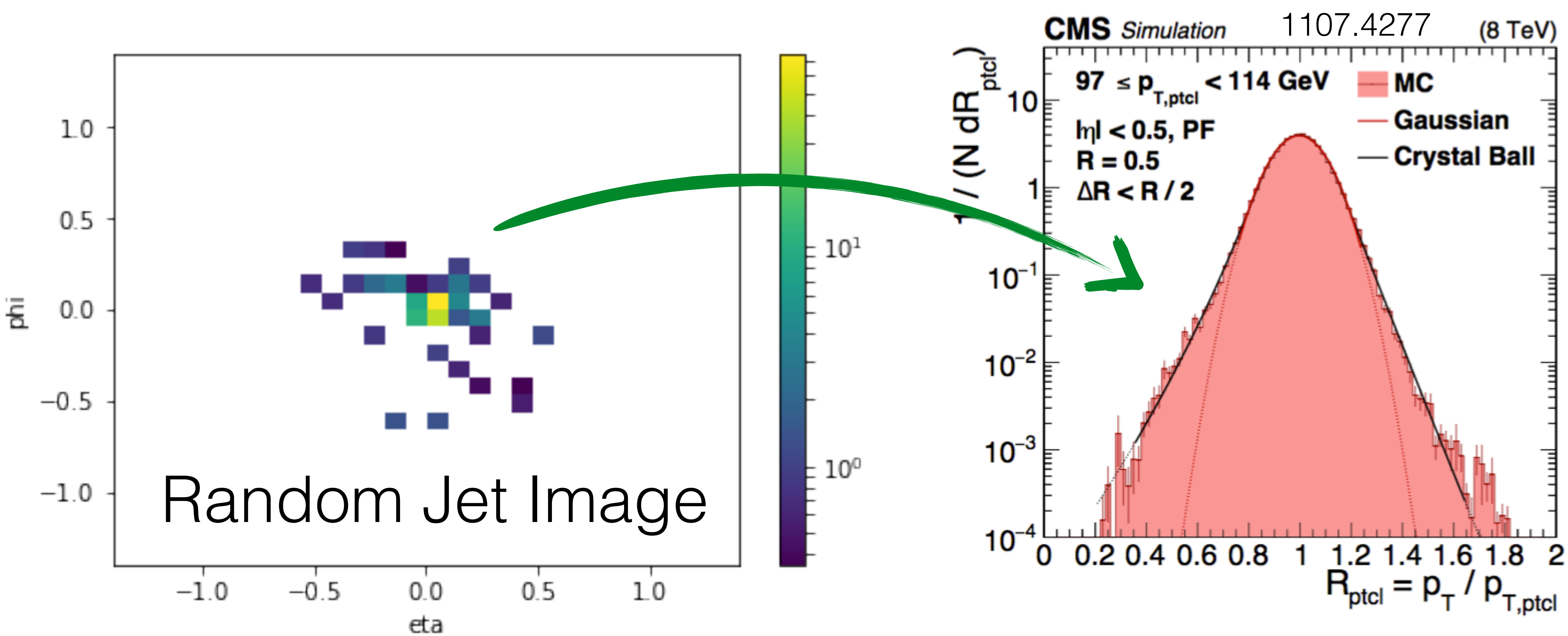
Regression with Scikit-learn

- Very simple random forest
- Minimal optimization on max.depth (currently = 11)

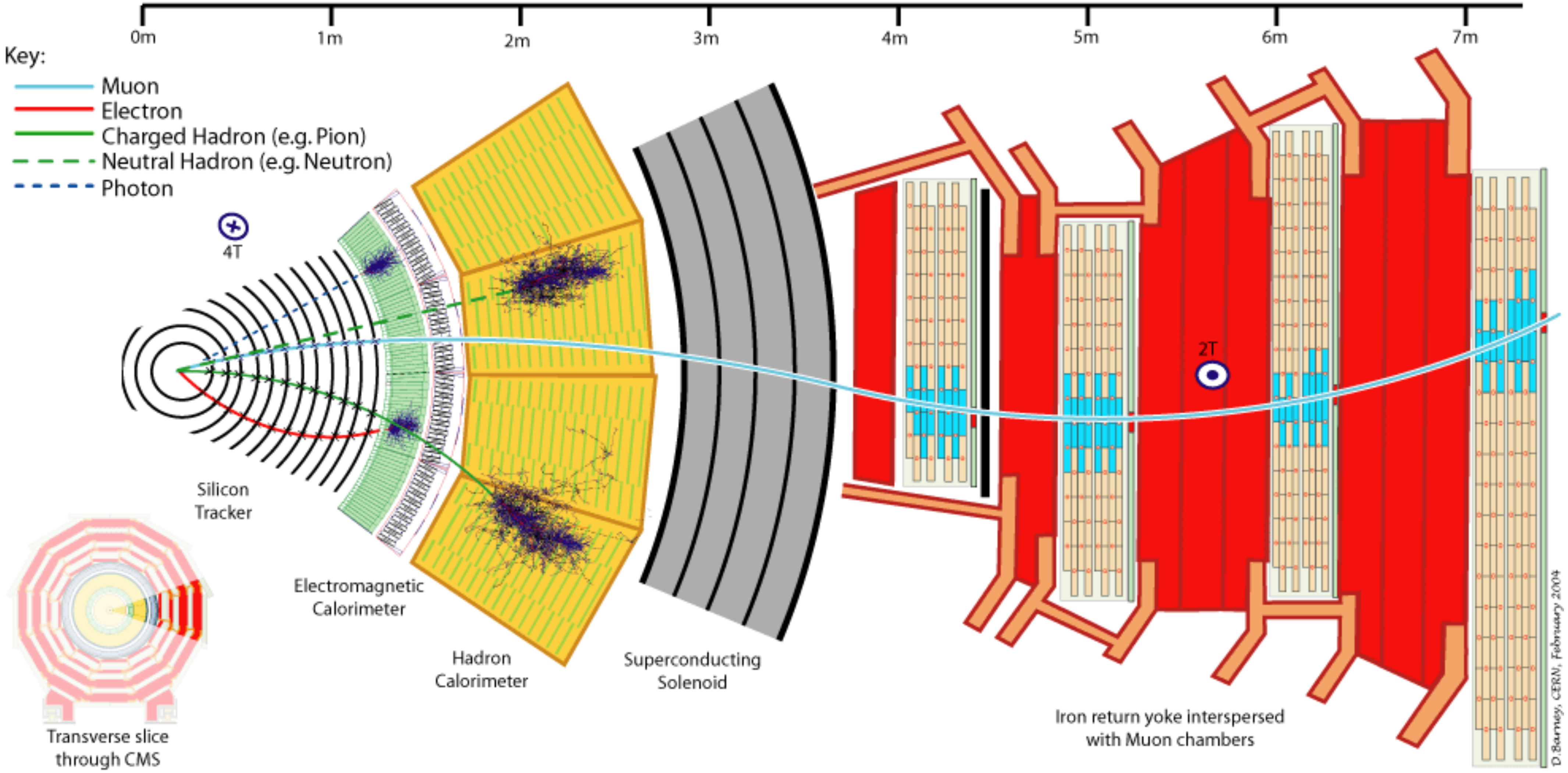


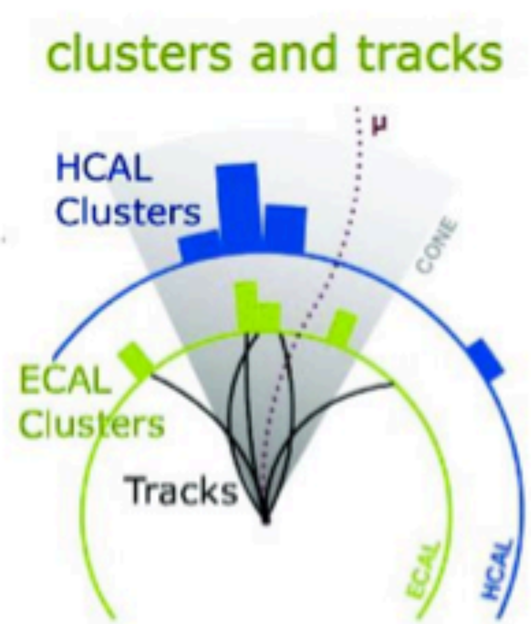
Motivation

- Train a Network —> Jet Energy Response
- Use CMS Open data for training

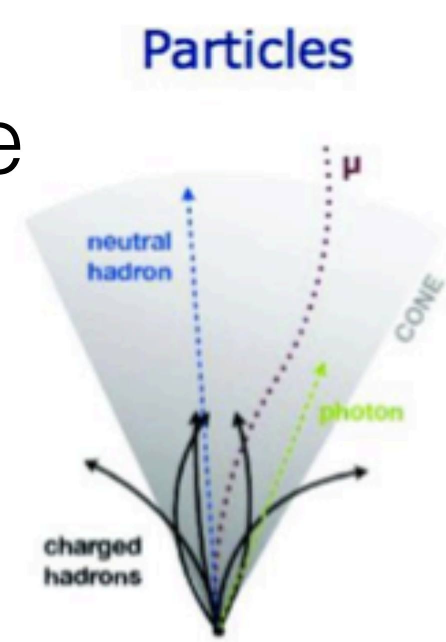


- Regression of a 2D image to a continuous variable



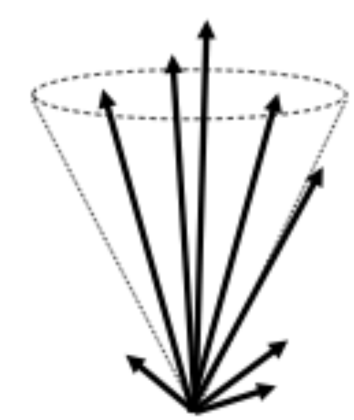


Particle Flow



FastJet

Jet

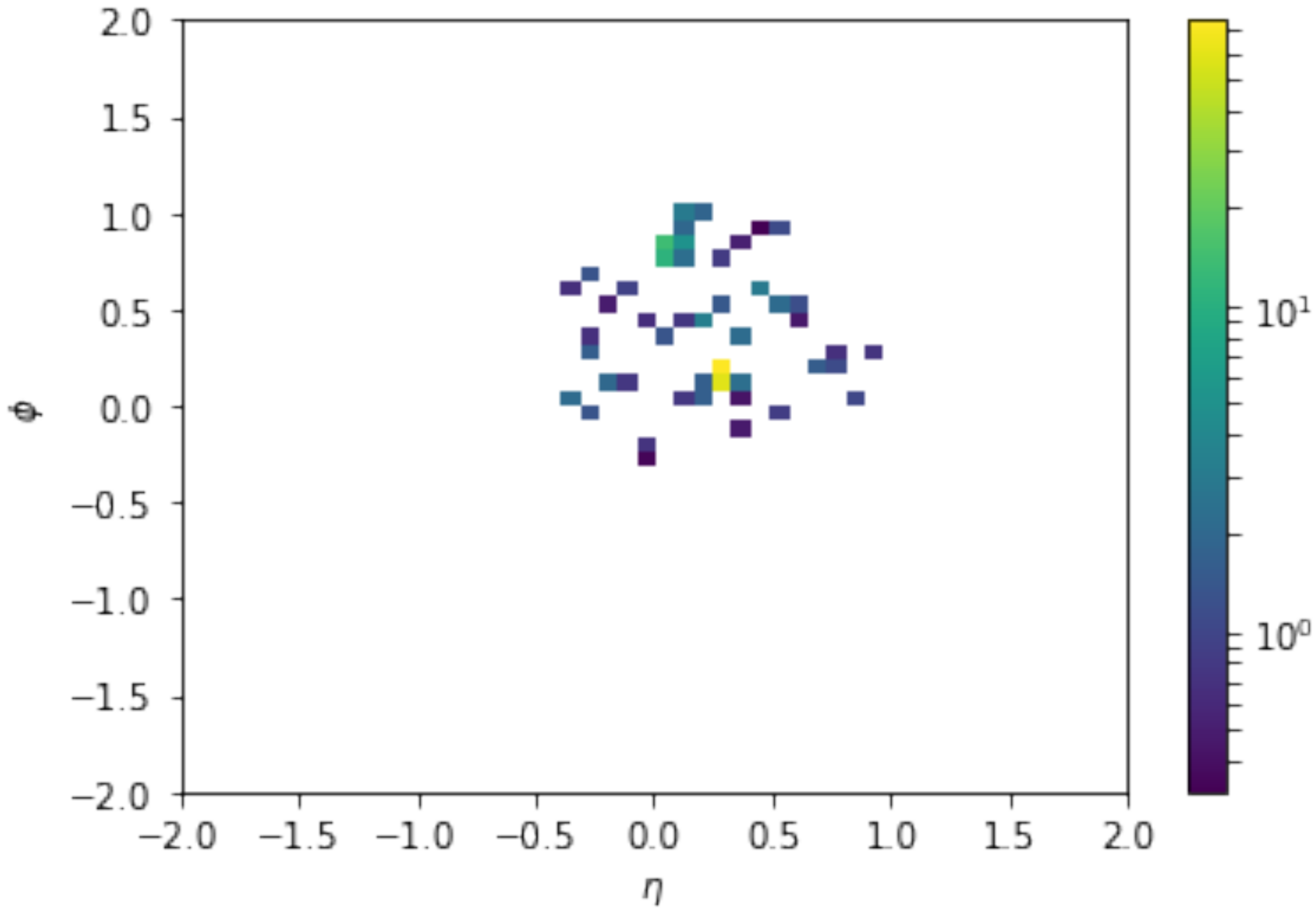


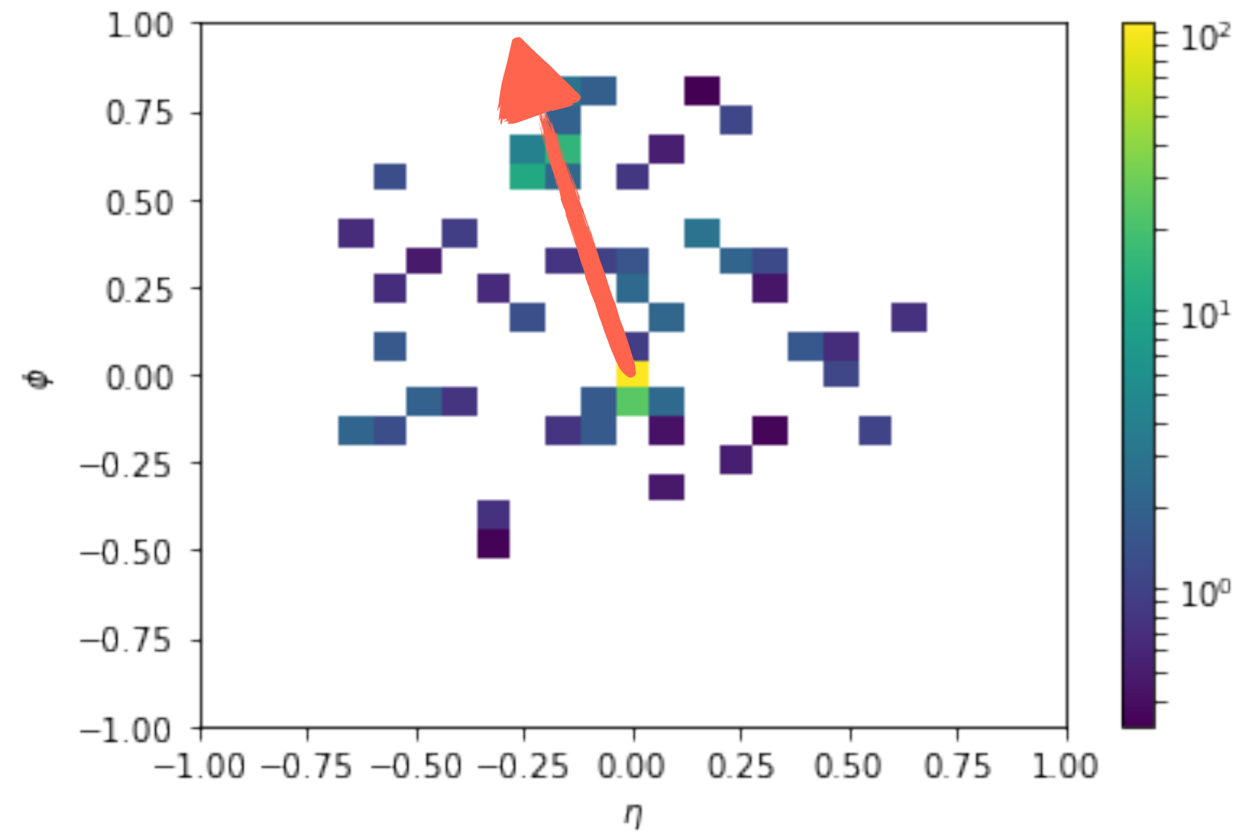
Anti-k_T algorithm is used in most of CMS publications

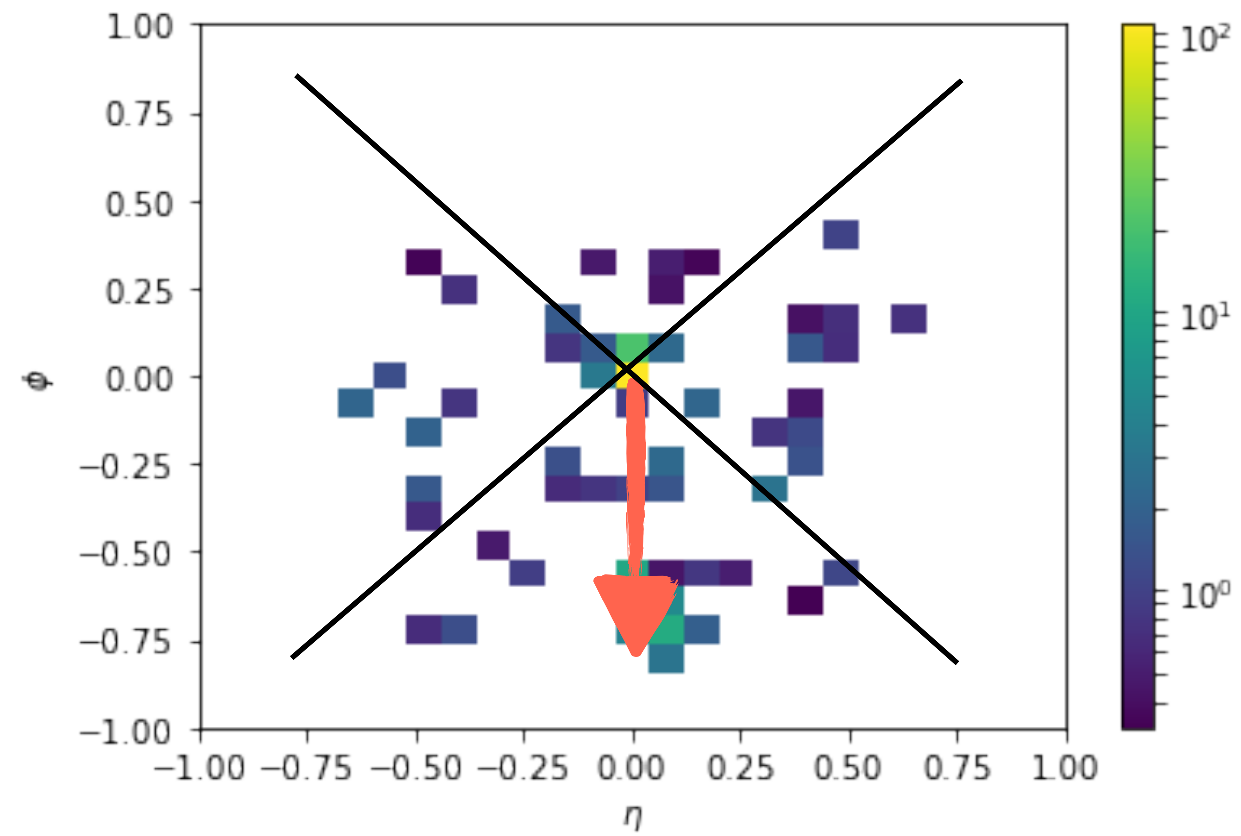
For instance, $\Delta\eta \times \Delta\phi$ 0.076 x 0.076 in barrel

Background subtraction and jet clustering

Pre-Processing



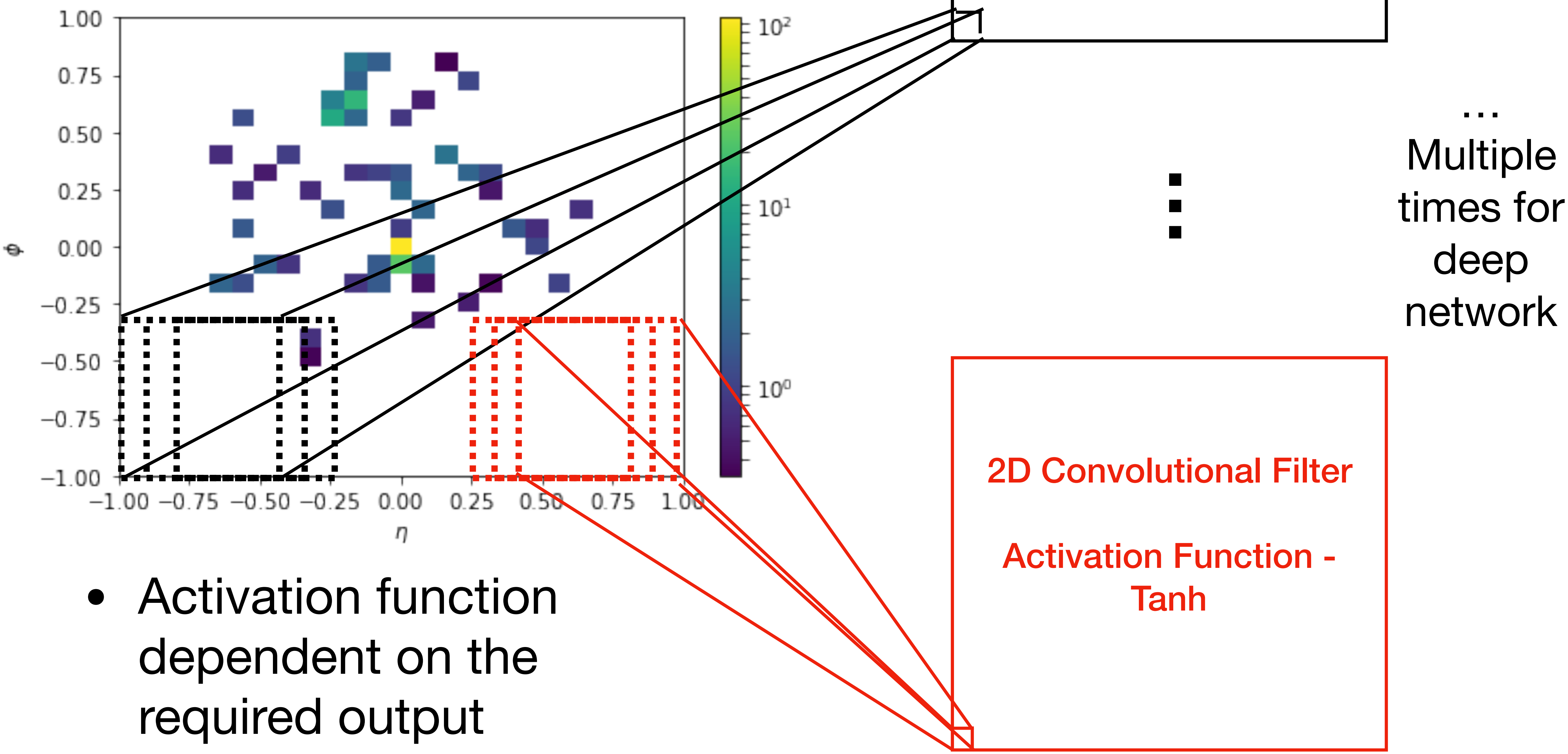




- Common procedure to induce uniformity in training
- Rotation loses significant detector-level information

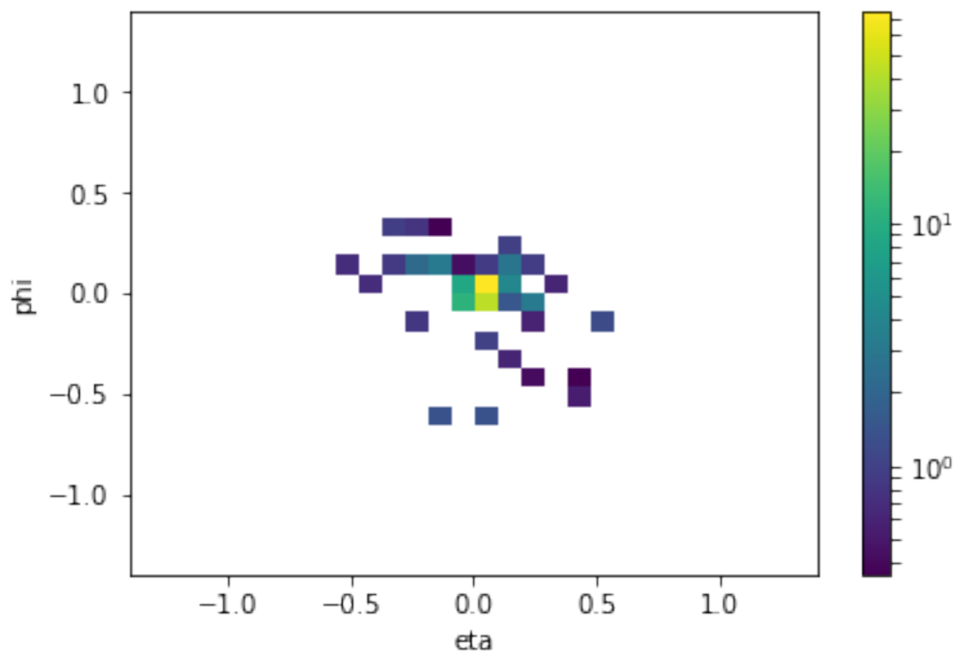
Setting up a DCNN

- Creating Multiple convolutional filters
- The larger the filter the more physics it captures - reduces effect of sparsity

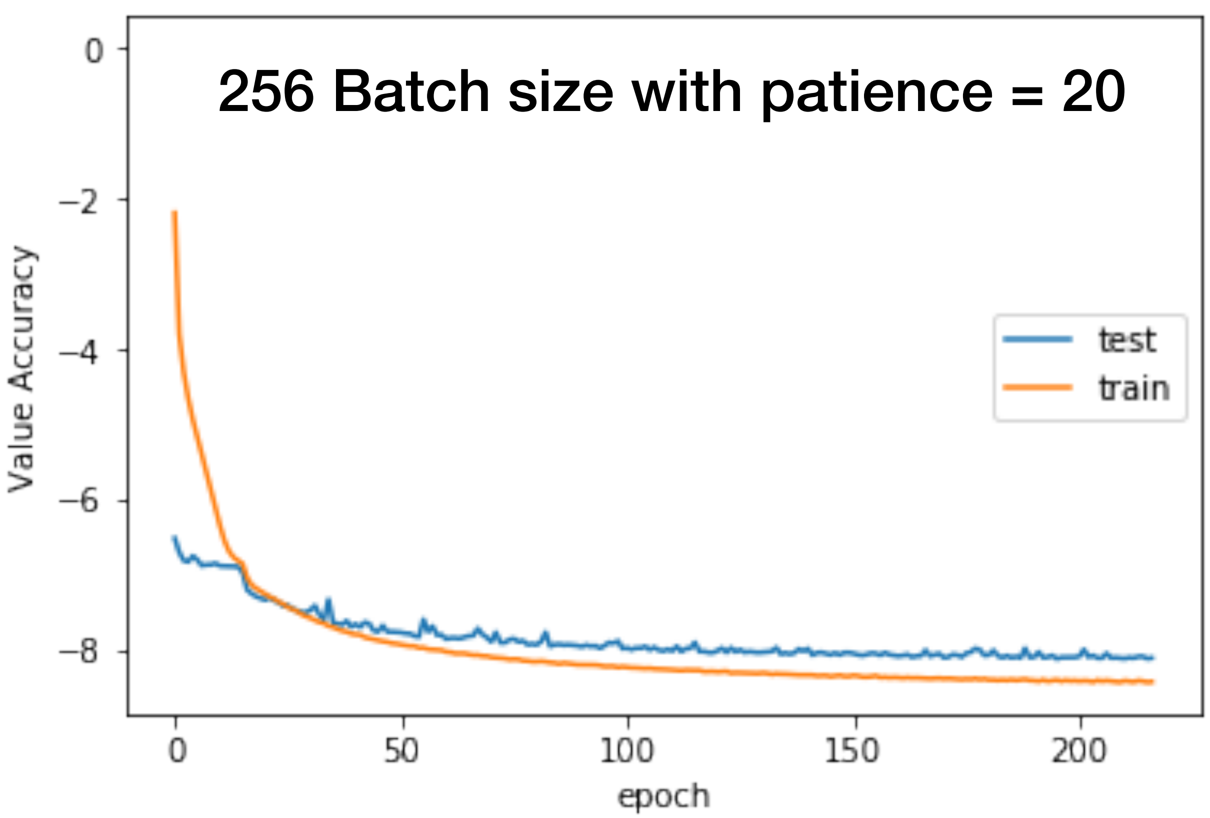


- Activation function dependent on the required output

DCNN Details

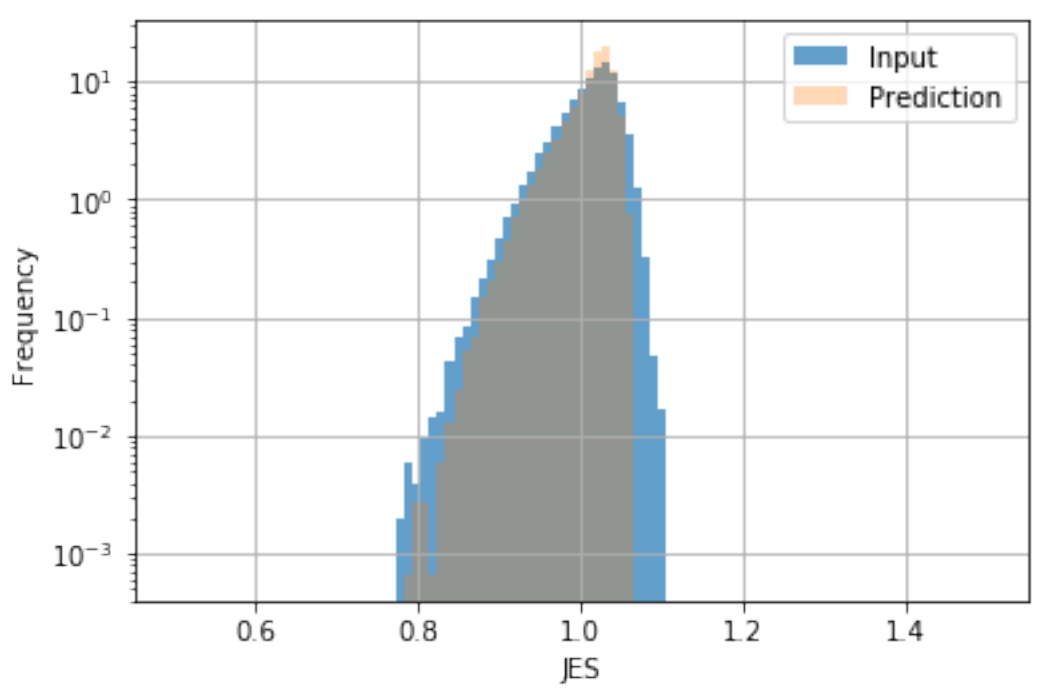


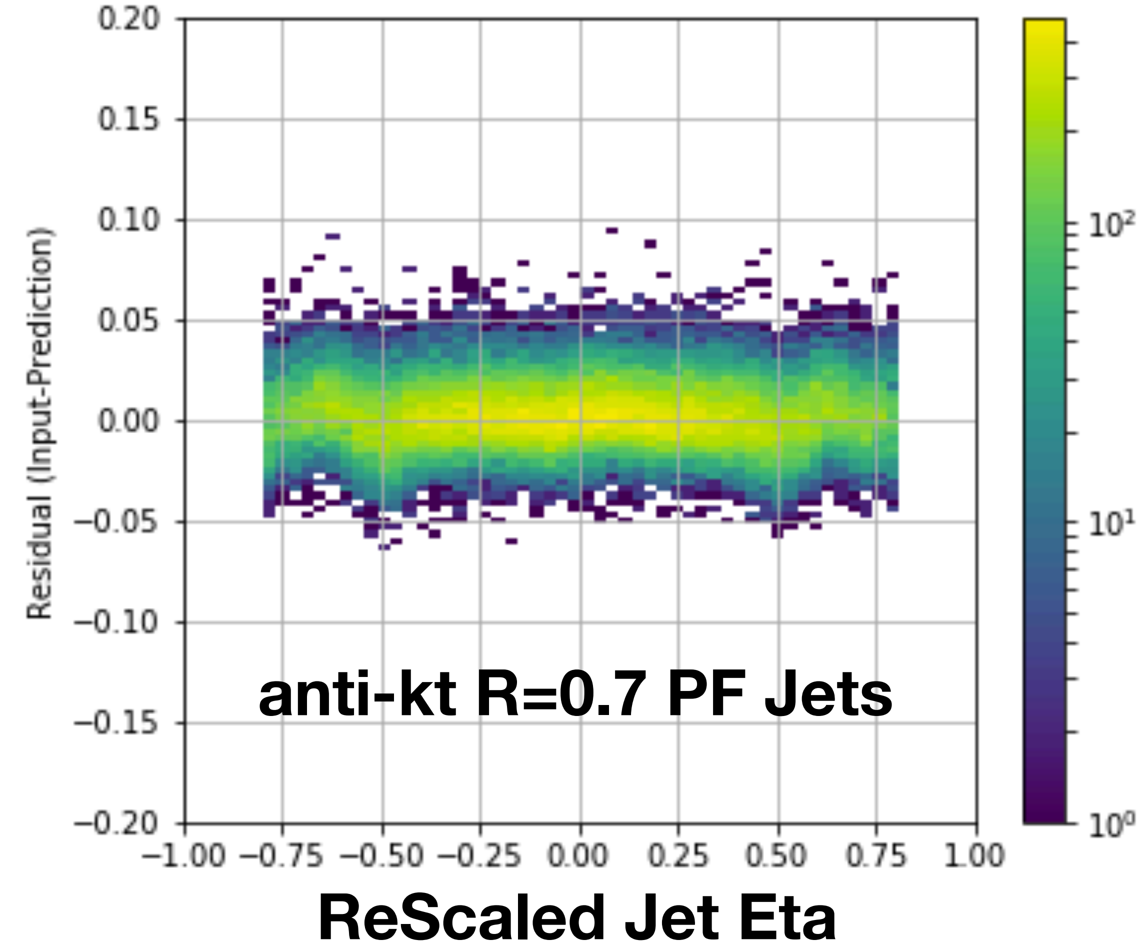
- Convolution2D (20, 11,11)
- MaxPooling (2, 2)
- Convolution2D (10, 7,7)
- MaxPooling (3, 3)
- Convolution2D (8, 5,5)
- Convolution2D (6, 5,5)
- MaxPooling (2, 2)
- Convolution2D (4, 5, 5)
  - Tanh activation for conv2D
- Flatten
- Merge Jet Eta
  - 20 Dense layers w/ sigmoid
  - Dropout 0.08
- Merge Jet pT
  - 20 Dense Layers w/ soft plus
  - Dropout 0.08
- Output layer - Linear activation
- Adam optimizer with mean squared error loss function

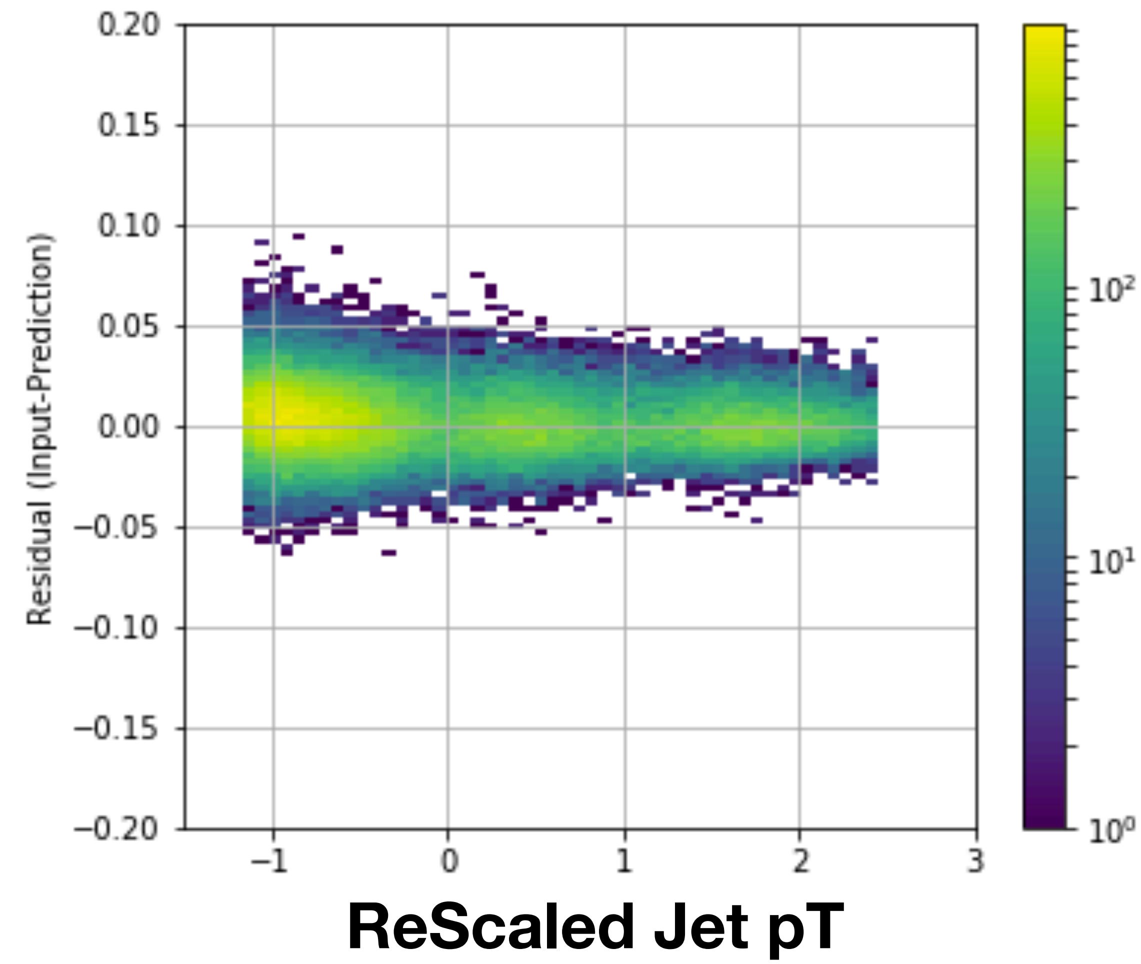


Predicting the Response

- Residuals as a function of the scaled jet eta, pT
- grid size : 30 x 30
- Effectively captures behavior with a smaller width

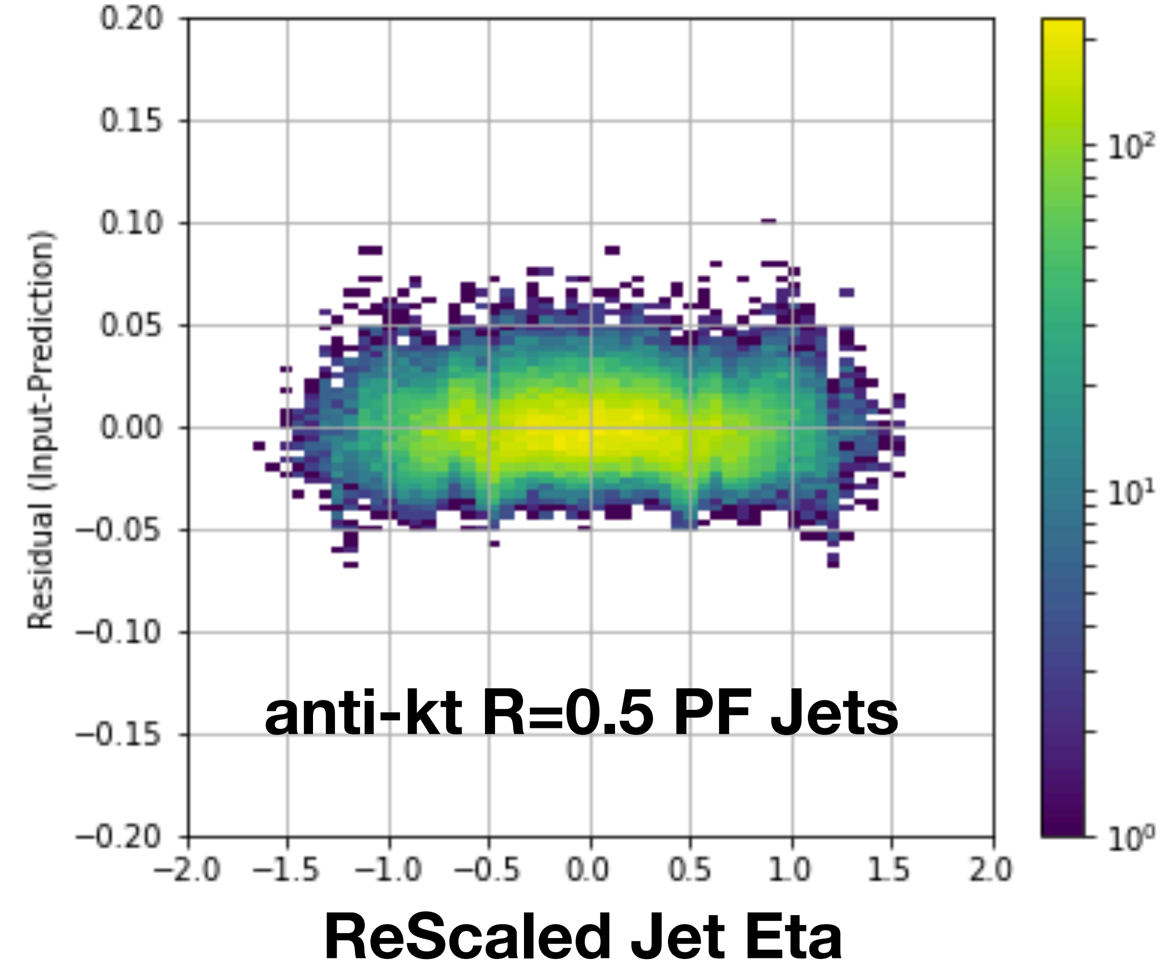


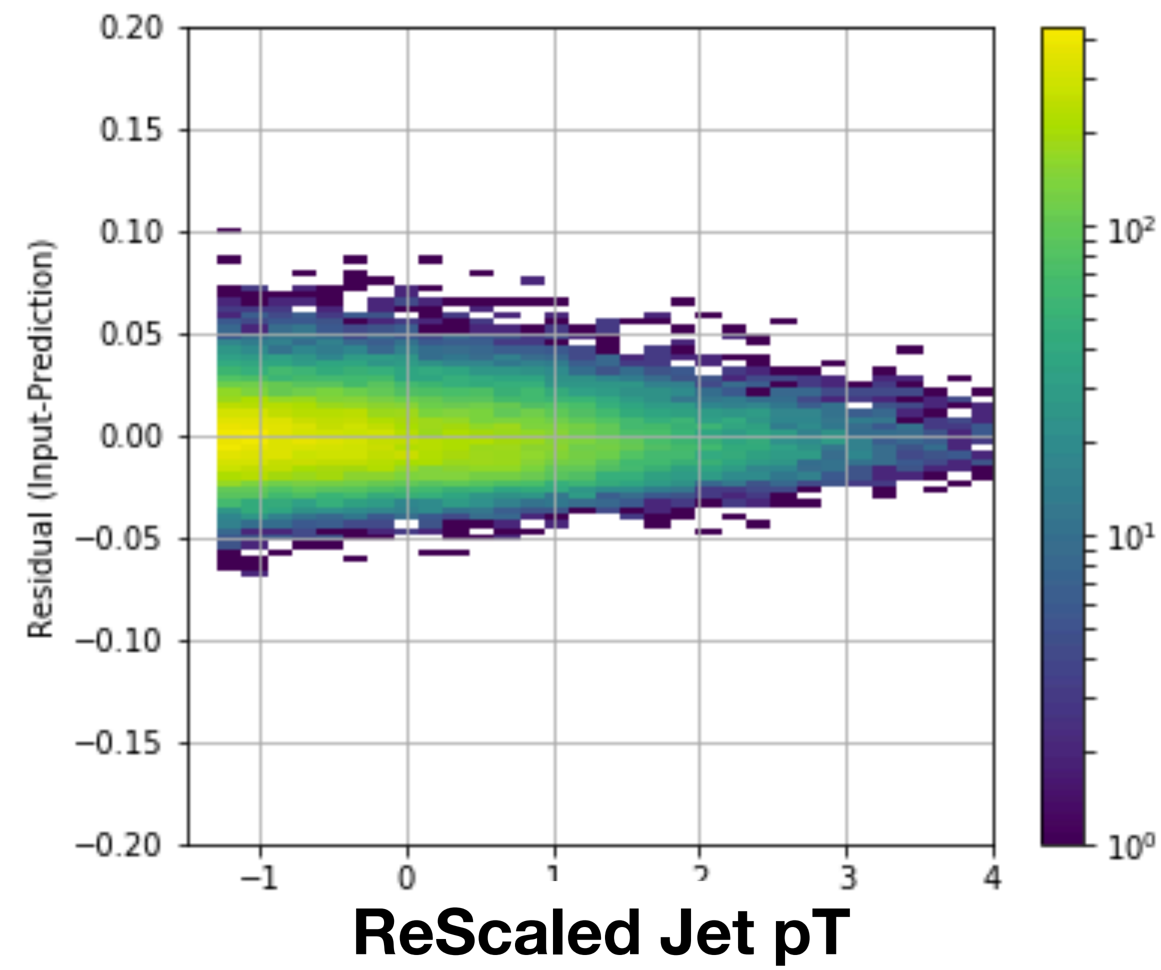




Can we go to smaller radii

- R = 0.5 jets require larger correction factor
- Model capable of reproducing similar levels of performance for R =0.7
- Grid size is reduced to 25x25





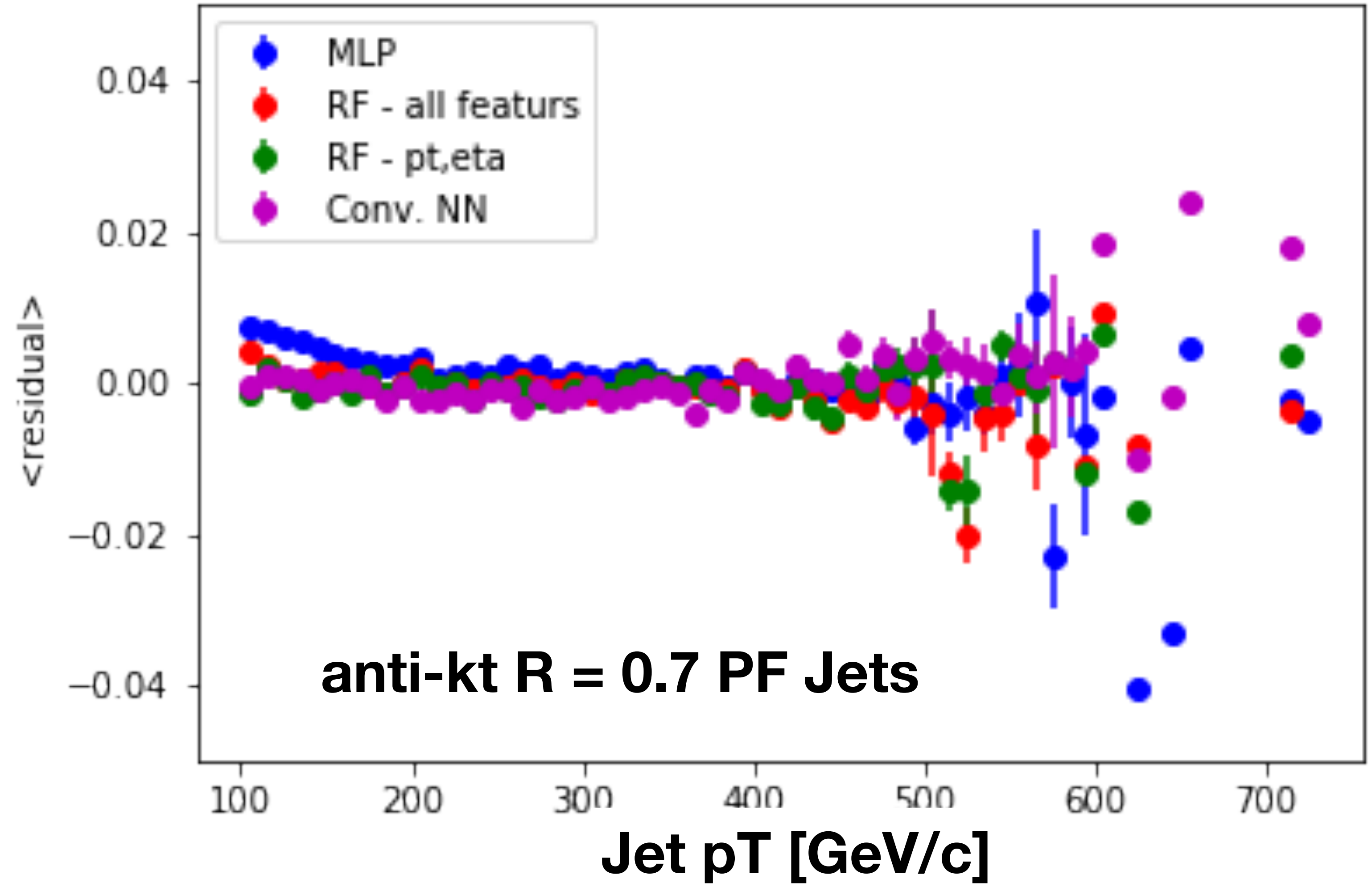
Model Comparisons

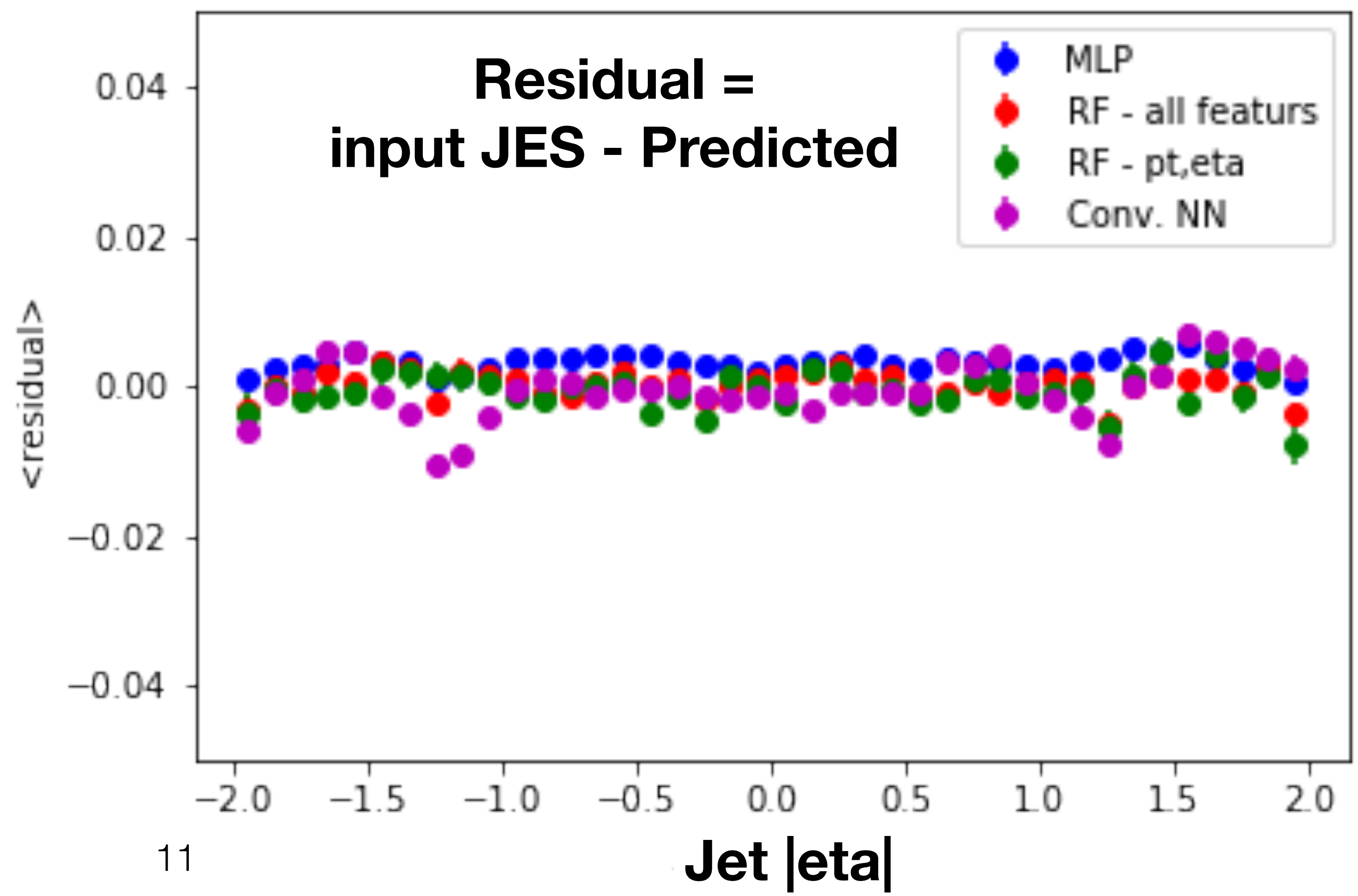
- R = 0.7 Jets
- DCNN - Jet Image w/ pT/eta (Mean Residual)
- RF - Random Forests

```
models=random_forest_regression(fact
ors=factorNames,regressor='jtjec')
models.max_depth=20
models.n_trees=10
models.fit(new_df_train,True)
models.test(new_df_test,True)
```

- MLP - MultiLayer Perceptron

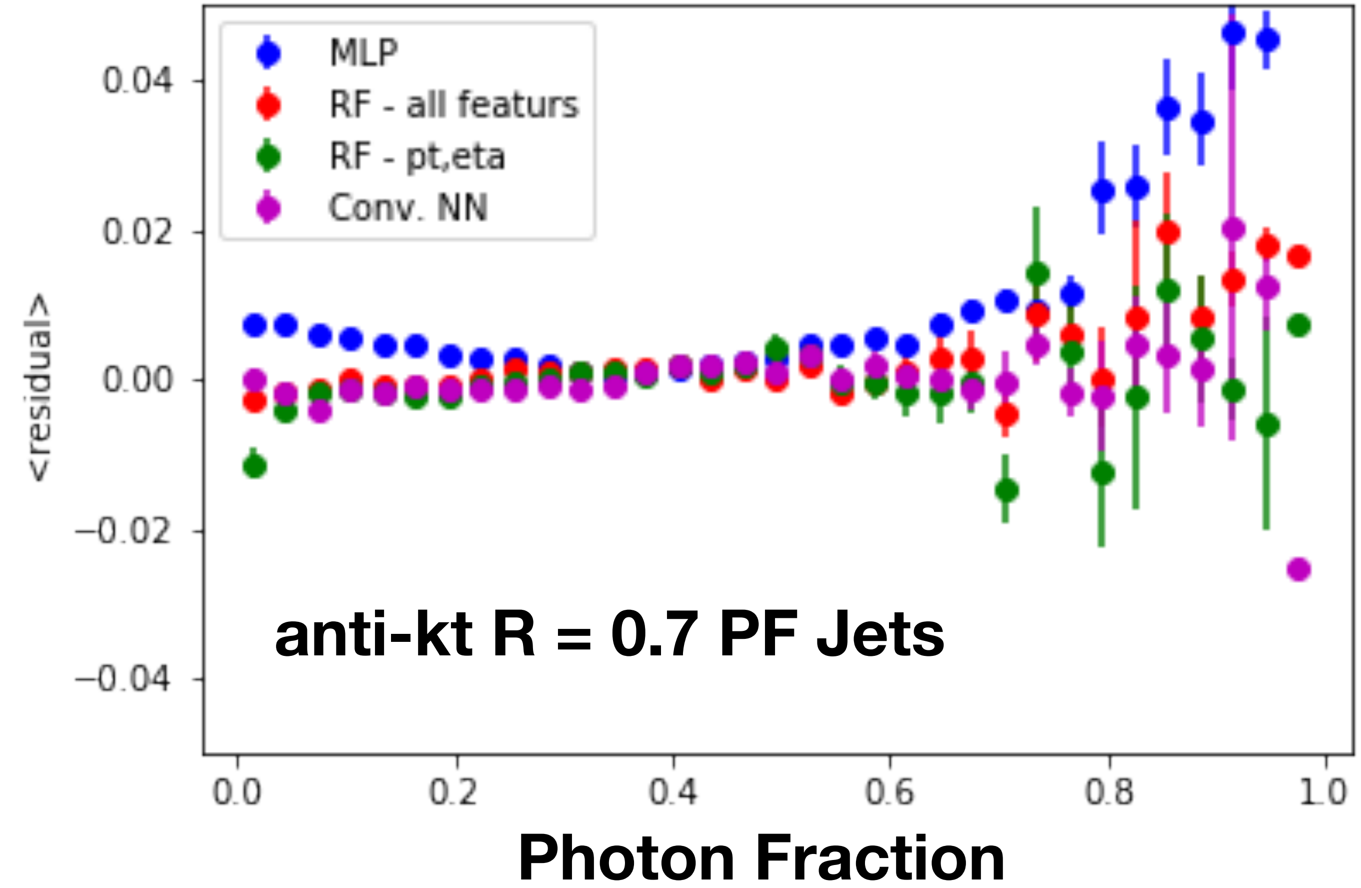
```
model =
MLPRegressor(hidden_layer_sizes=[200
,200,200,200],activation='relu',rand
om_state=12345)
```

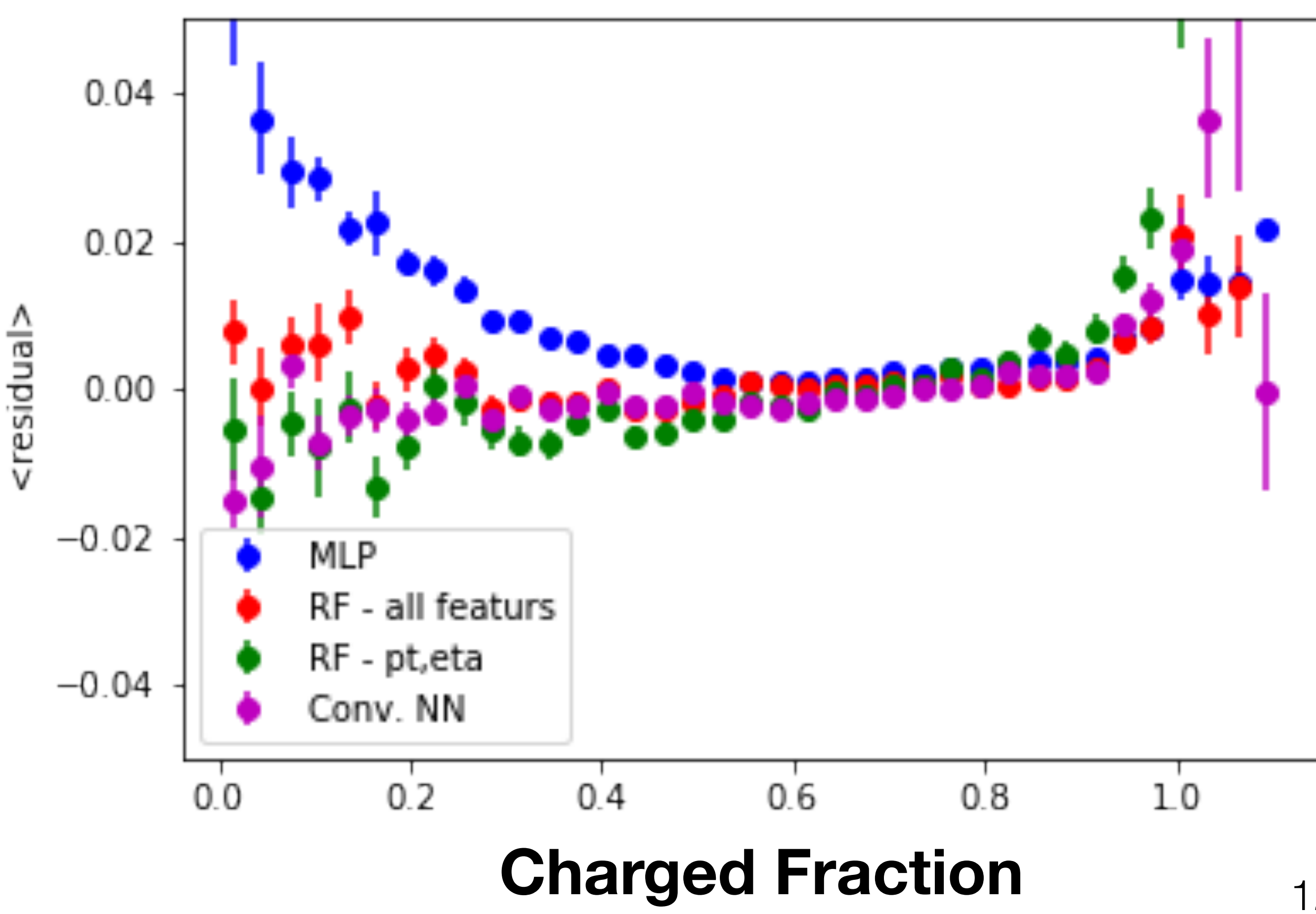


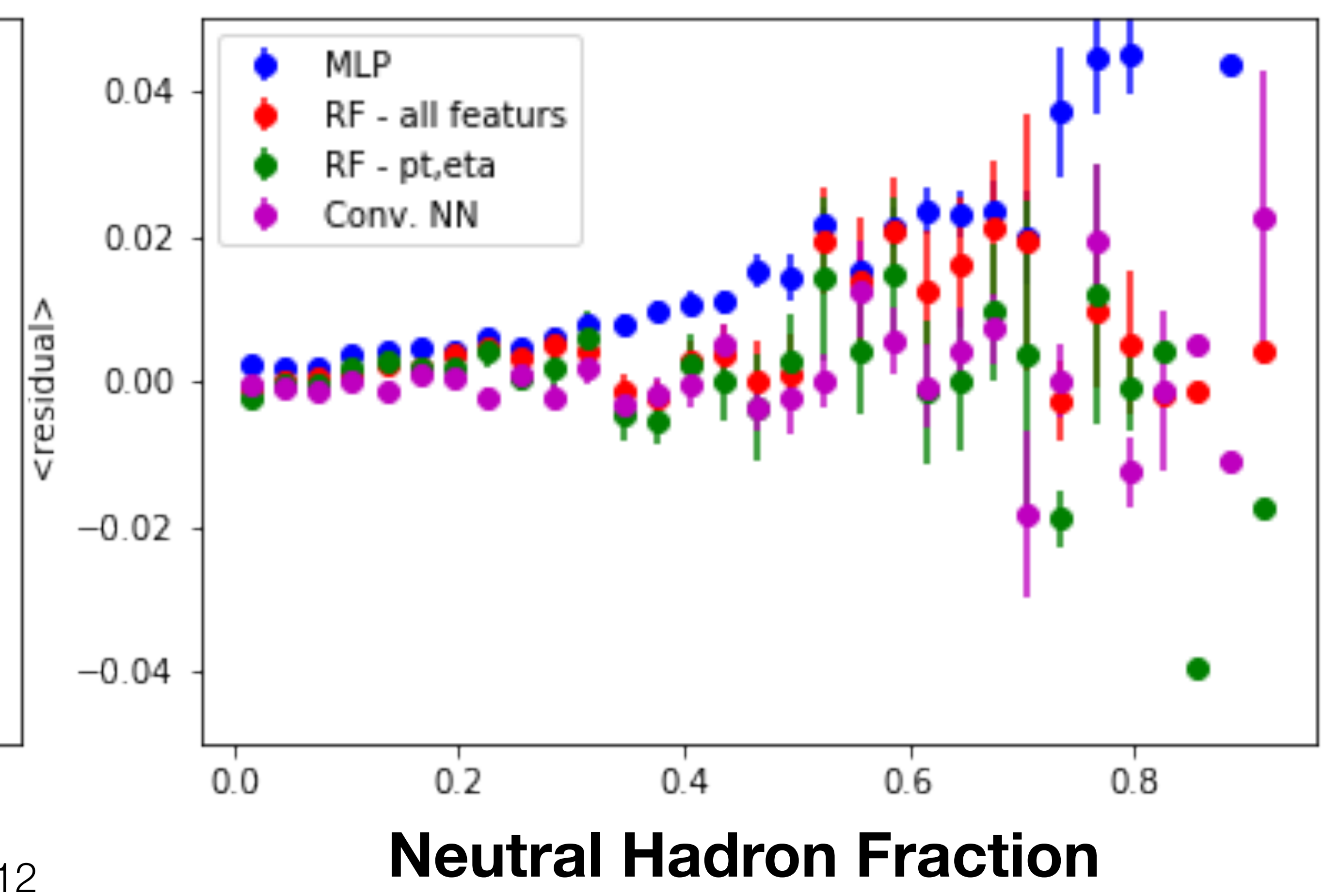


What is the model learning? - I

- Residuals as function of energy carried by various jet constituents (charged, neutral hadron, photons)
- RF (all features) and DCNN showcase very good performance

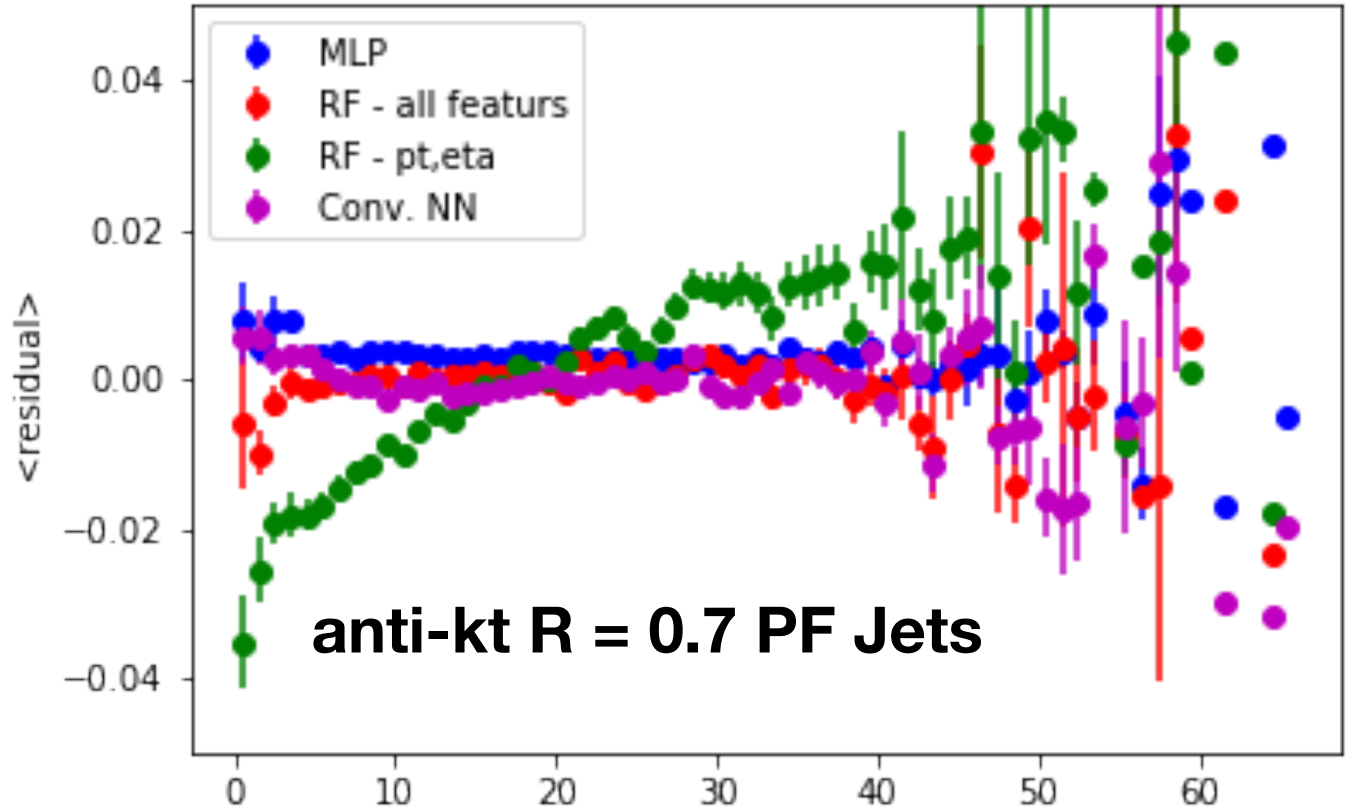




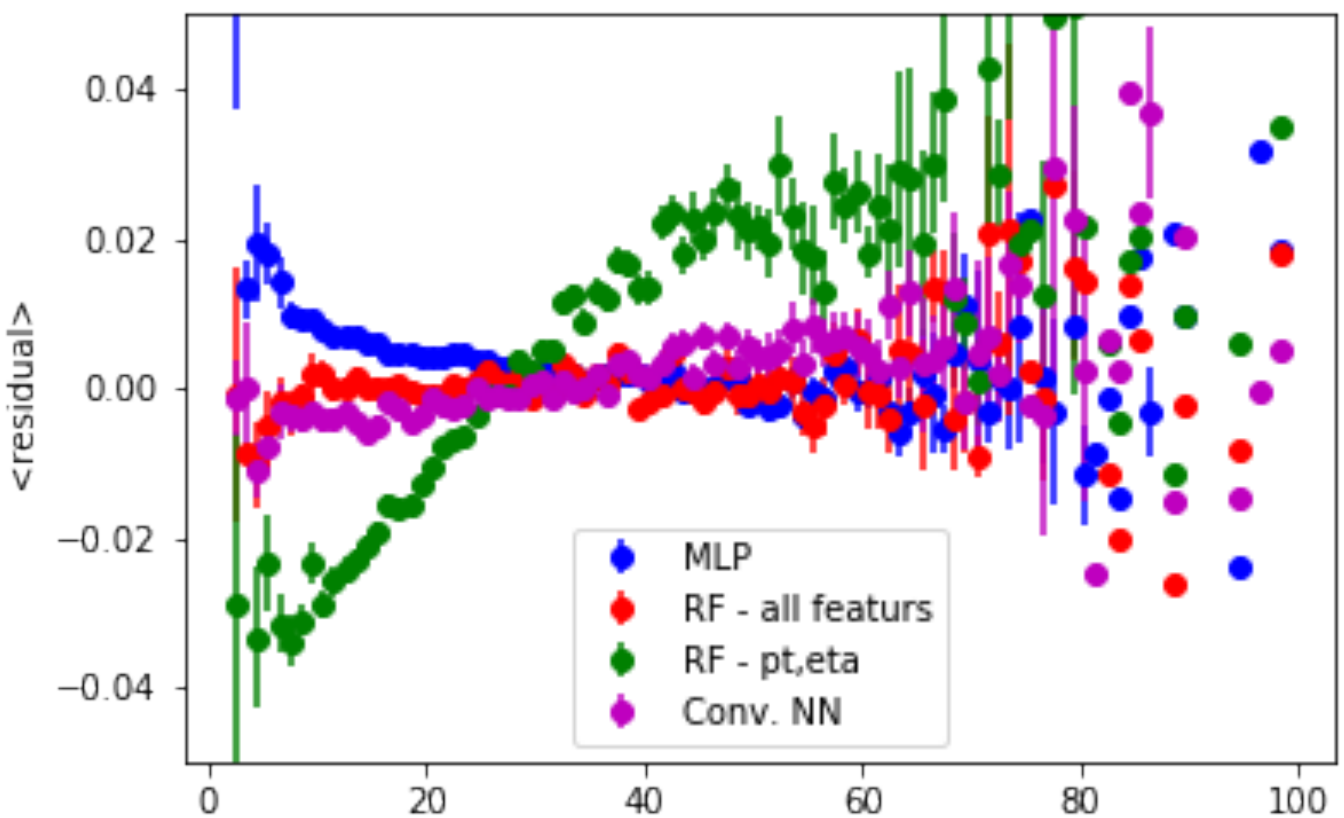


What is the model learning? - II

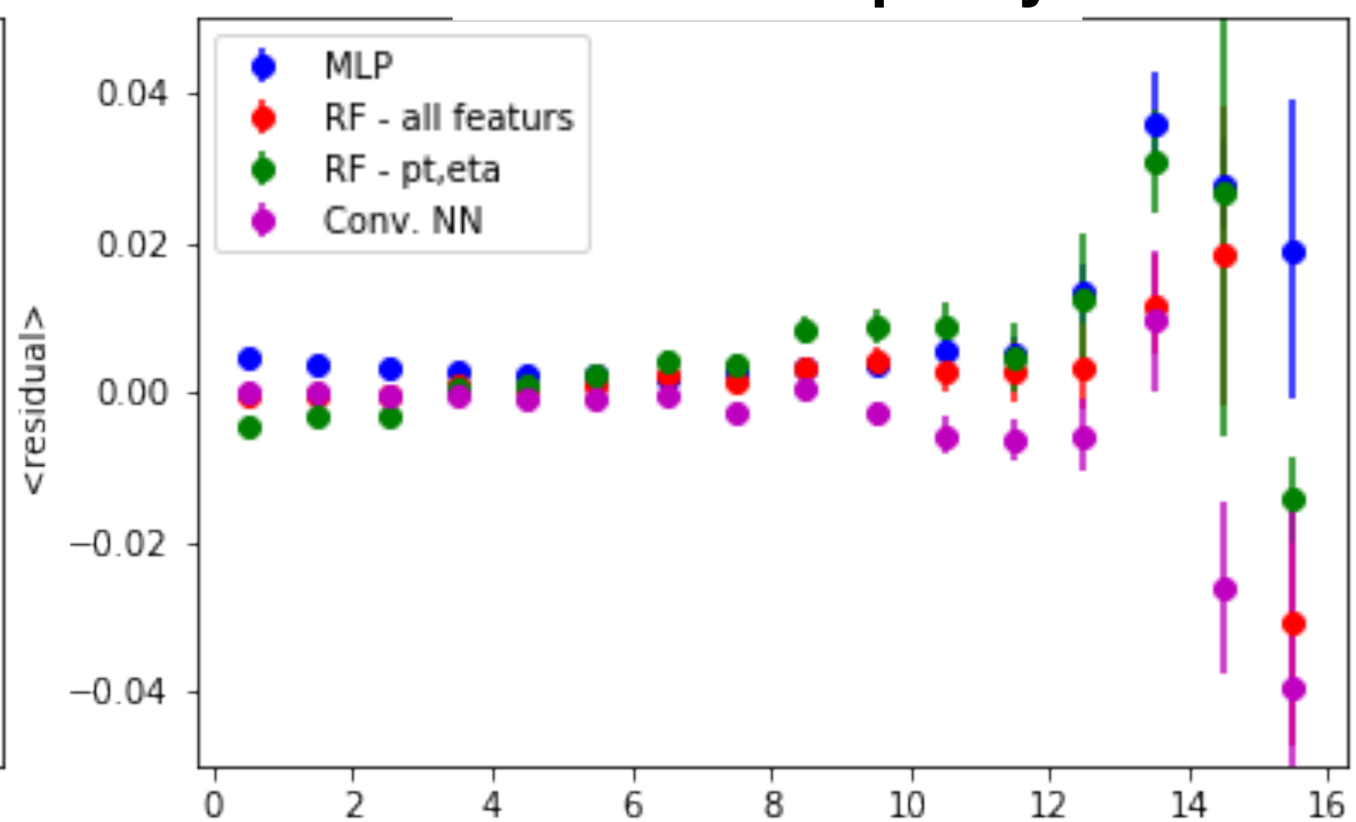
- Residuals as function of # or multiplicity of identified jet constituents (charged, neutral hadron, photons)
- Multiplicity dependence captured in DCNN - from the jet image!



Photon Multiplicity



Charged Multiplicity



Neutral Hadron Multiplicity

Conclusions

- DCNN trained on jet images is effective in learning the detector response

- Fragmentation dependent response encoded in jet images are extracted

- With trained on Open data, one can build up such a resolution-unsmearing model for any experiment

- Currently training on the jet response, next steps is to train with generator level jet information

- Longer term goal would be to look at single jet JER uncertainty

Backup