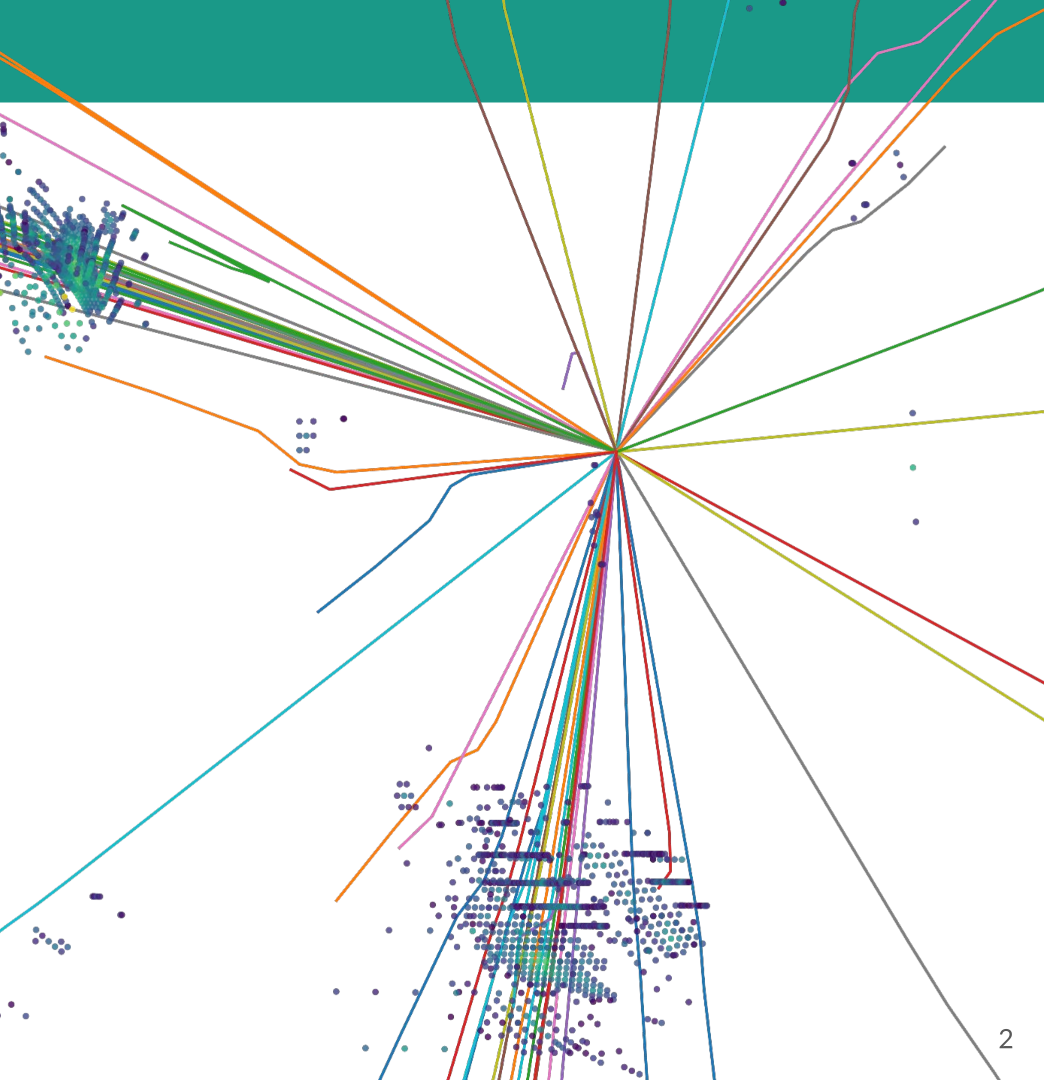
PointNet for pflow

Joshua Himmens,
Dr. Luca Clissa,
Dr. Maximilian Swiatlowski

Outline

- The p-flow problem
  - Current approaches
- Using PointNet for p-flow
  - Why PointNet
- Processing tracks and cells
- Loss, Metrics, and baseline calculations
- Results
  - Several datasets
  - Modal collapse
- Ongoing work
  - Transfer learning
  - Multi-class labeling



Problem

Existing ATLAS particle flow (p-flow) algorithm

- calo & track information
- good energy and angular resolution
- very helpful with pile-up

Main limitations:

- associate track to topo-clusters, not cells directly
- hand-tuned "ordering" algorithm for cell energy subtraction
- fails when tracks are too dense

Goals

Test ML approaches to target cell-to-track association and replace hand-tuned ordering

The main idea is to start from single-particle MC data building upon ATL-PHYS-PUB-2022-040

Specific focus is be on:

- improve association performance
- extend to jet environments
- optimise neighborhood size

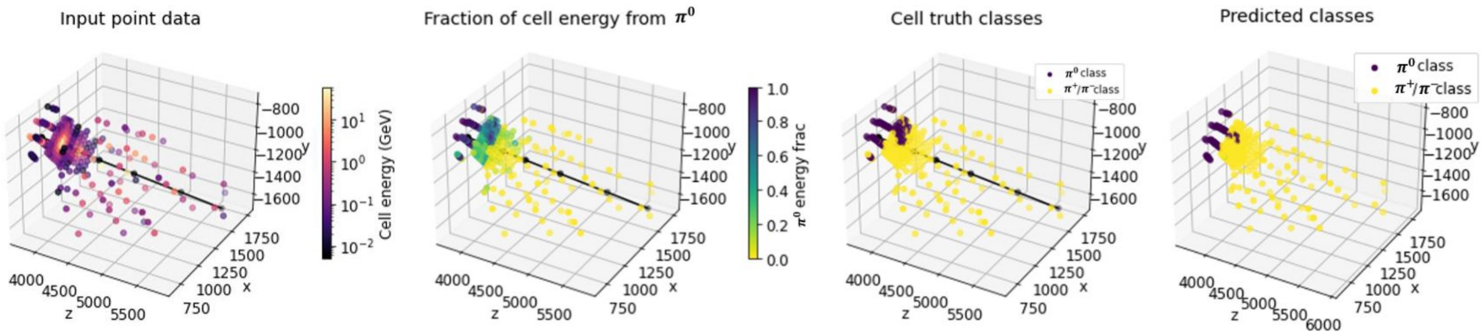
JIRA tracking: https://its.cern.ch/jira/browse/ATLJETMET-1699

Current strategy

Represent data as point clouds

- sparse, efficient representation
- enable joint use of calorimeter and tracker information

Adopt segmentation models, i.e. assign track labels to each point (track & calo)



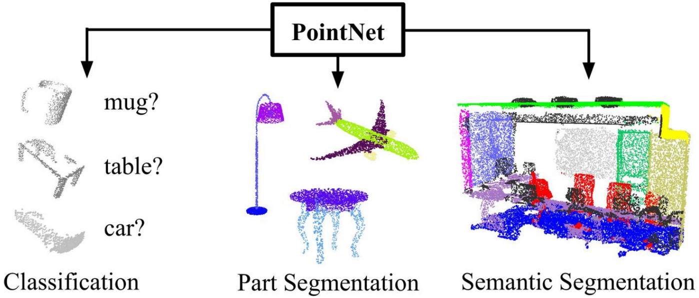
Jessica Bohm et al., Science Week 23

PointNet model [12]

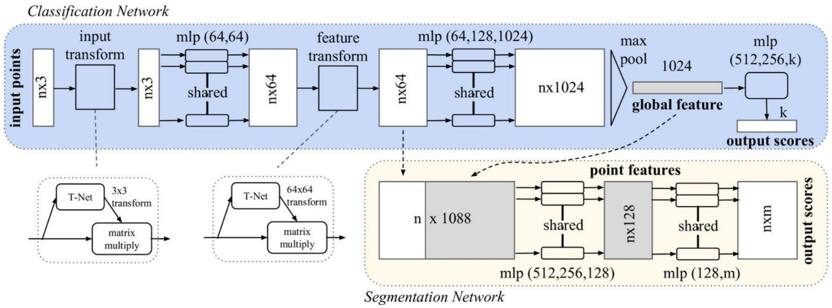
The idea is to adopt NN architecture suitable for point clouds → PointNet

Several learning tasks:

- classification
- part segmentation
- semantic segmentation



- 👍 permutation invariant
- 👍 transformation equivariance
- 👍 both shape classification & segmentation
- 👍 robust to data corruption → critical points
- 👎 no local context → global feature learning
- 👎 generalization to unseen scenes → global features depend on absolute coordinates
- 👎 no rotation/shape equivariance



Datasets

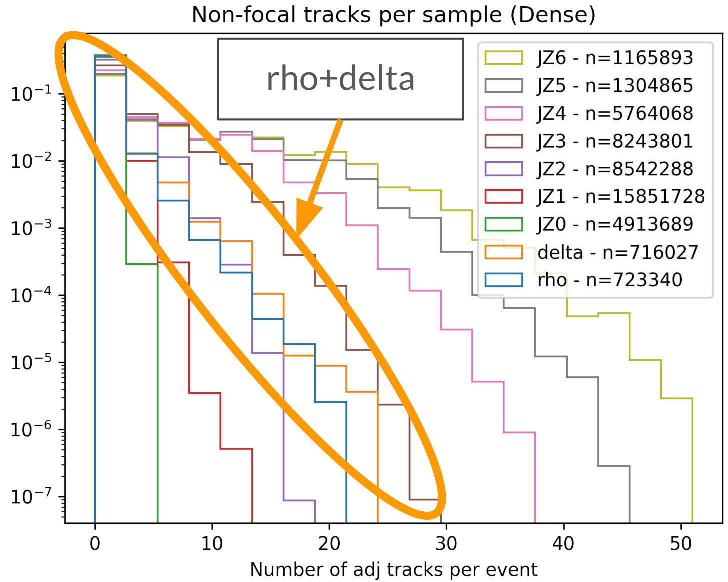
Heterogeneous Data

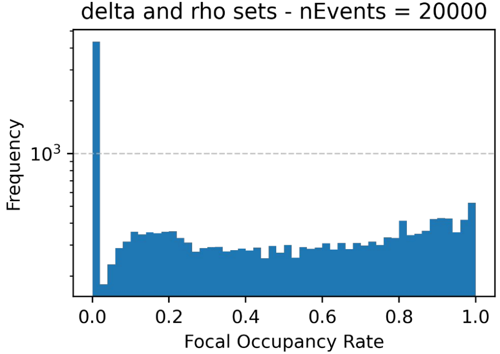
We analysed **several MC simulations** (ML Tree.):

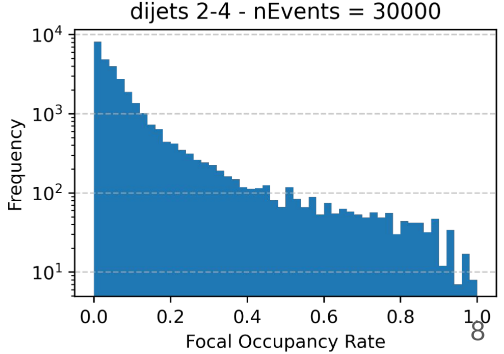
- **rho, delta decays**
  - typically 1 track per event
  - many focal calo hits per sample (~10-200)
- **dijets decays (split into JZ by pt)**
  - 1 to 32 tracks per event
  - few focal calo hits per sample (<50)

For each track, we create a *sample* with all hits in a ΔR of size 0.2

- the samples constitute individual point clouds fed into PointNet
- **focus VS "unfocus" hits**

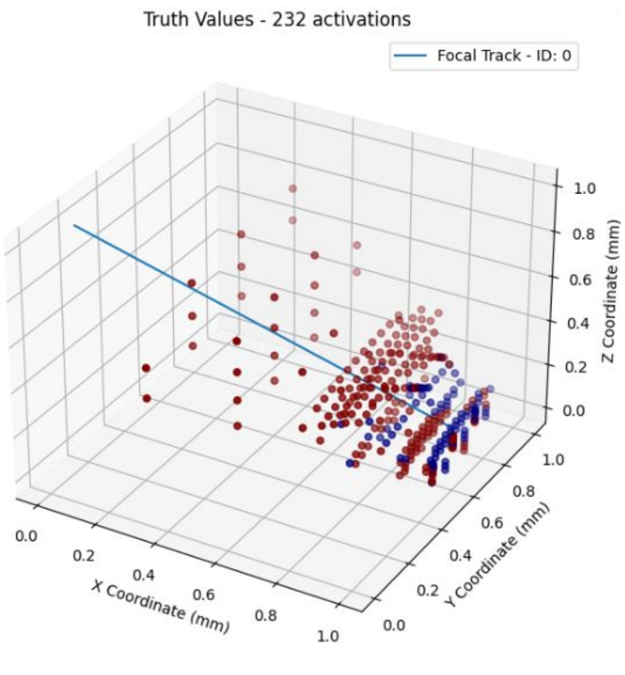




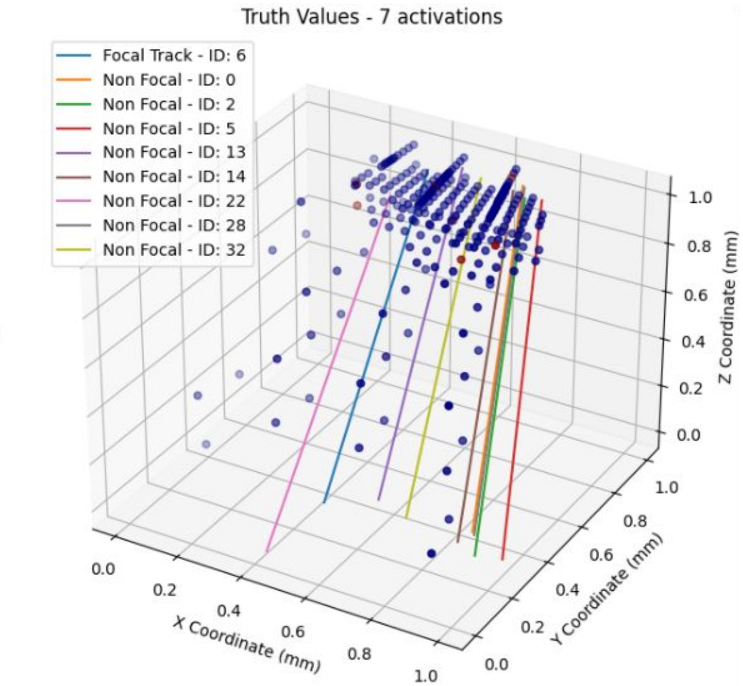


Disparate Training Data - Example

rho sample



dijet (JZ4) sample



Pre-processing

The majority of the complexity is the preprocessing to make the root data suitable for a PointNet.

- Flattening ROOT ntuples and cutting for ΔR
- Split events trackwise $\rightarrow$ samples
- Remove ernouse tracks caused by simulation quirkes
- Embed data into a matrix of points

All this is available as part of JetPointNet on gitlab under nightly.

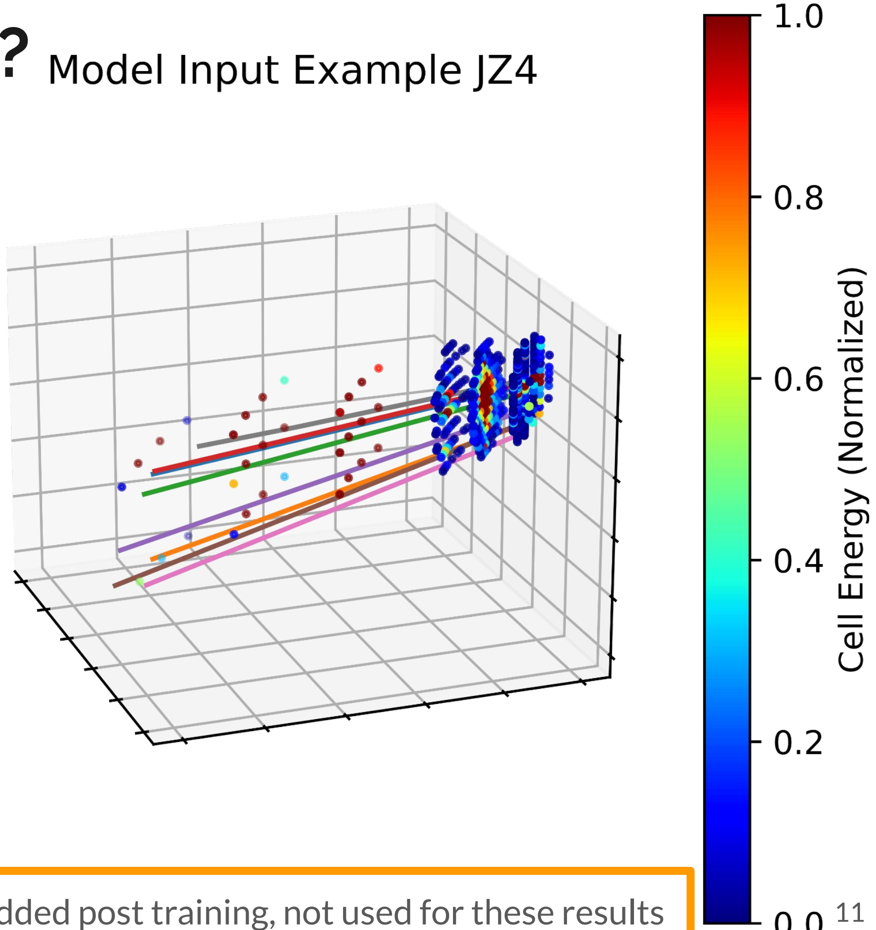
.

What does the model see?

The cell hits and extrapolated track hit point clouds are overlaid.

Tracks are associated to a truth particle.*

Cells are attributed to a single truth particle even if they are a member of multiple clusters or have multiple contributing particles.



*There are cases where tracks have interactions in the tracker and lose their label, these events are cut from the training set.

*Added post training, not used for these results

Data Processing Challenges

Not all fields are relevant to all points.

Some points are semantically related, like track points, which is difficult to represent in a point cloud.

A static length input means that most events will be significantly padded.

*Added post training, not used for these results

Data encoded into each point
Point type (e.g. cell hit, track, padding)
ΔR
Track Identifier (for tracks)
Normalized coordinates
Track □^{2}/dof (for tracks)
Cell E (for cells)
Track pt (for tracks)
Cell Sigma (for cells)

Data encoded into each point

Loss, Metrics and baseline

How do we measure performance?

Given the complexity of the problem, we must carefully what constitutes good performance and how to measure it.

- what do we care about?
  - mostly cell hits
  - more energetic hits
    → masked loss/metrics and energy weights
- how to measure it?
  - several alternative options, with pros and cons
    → no *perfect* metric, consider multiple together
  - data challenges: class imbalance (e.g., rho dataset: 66% class 0)
    → use class-weighted loss/resampling
    → accuracy alone is misleading

Loss

We adopt a standard **energy-weighted binary cross-entropy**:

$$\mathrm{WBCE} = -\mathrm{alfa} * y \log(y^*) - (1-y) \log(1 - y^*)$$

This should ensure that most energetic hits have larger impact on the loss, thus forcing the training to pay more attention to those hits.

Several weighting schemes have been attempted: none, quadratic, absolute value, Cox- transform....

Metrics

For a thorough assessment, we jointly look at several metrics:

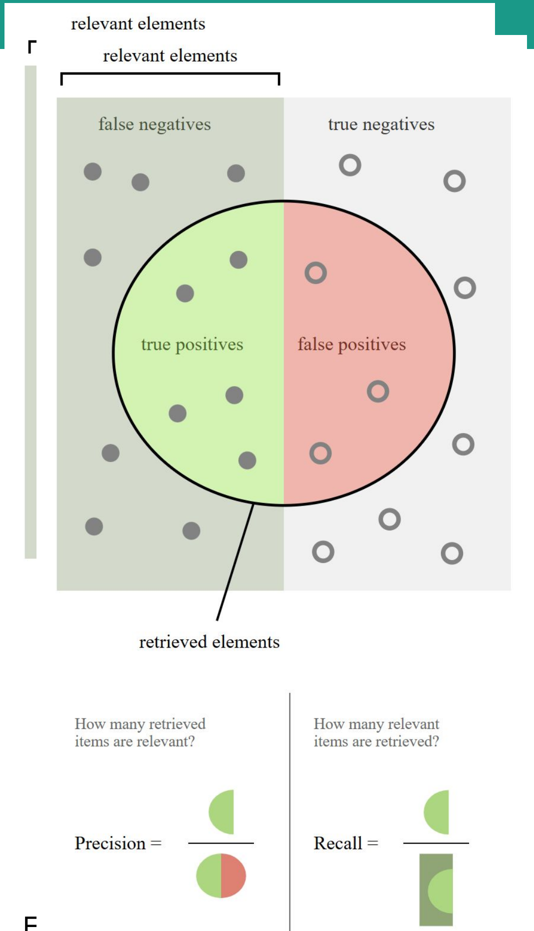
- **Accuracy**:

  (TP+TN) / (TP+TN+FP+FN)
- **Precision** (purity):

  TP / (TP + FP)
- **Recall** (signal efficiency):

  TP / (TP + FN)
- **F1-Score**:

  2 * P * R / (P + R)



F

Baseline

How do we define “good” performance?

Baselines:

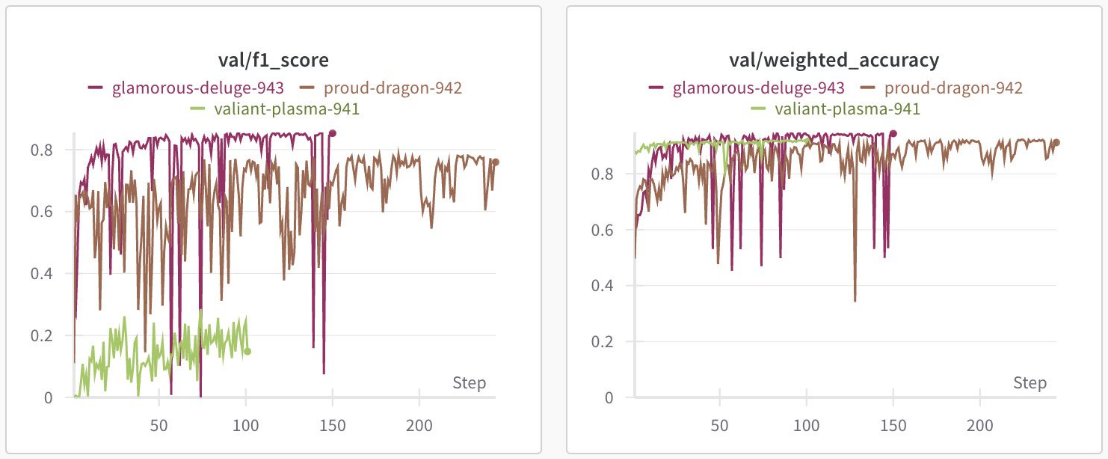
- “dumb” model: always predict majority class
- random model: random prediction (50/50 chance)
- “smart” random model: random prediction with probability linked to class proportions
    - ***rho + delta***: ***~64% of unfocus points*** (class 0)
    - ***JZ2 + JZ3 + JZ4:~94%of unfocus points*** (class 0)

Model training

Training Infrastructure

We train using NVIDIA RTX A6000's and manage training with Weights and Biases.

To avoid long epochs, we randomly sample ~204,800 events (~14 minutes of compute). Model convergence requires 100 epochs (~23h). Our training data contains 246M trainable samples (1.1TB)



Example Weights and Biases Plots

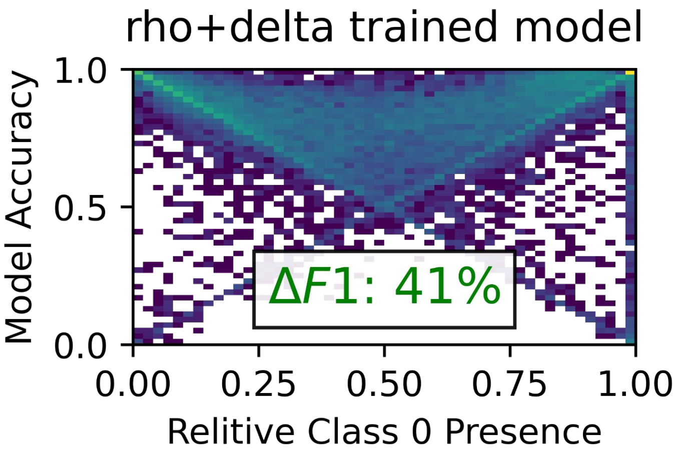
Results

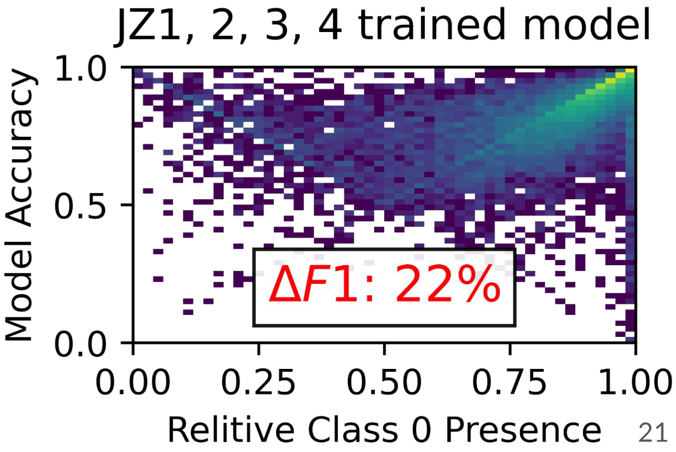
Results

Training on the rho+delta sets is successful with an F1 score of ~0.8 and a significantly improved energy weighted accuracy >0.9.

Dijet training is promising, however the model is unable to overcome modal collapse, where the entire event is attributed to non-focal tracks.

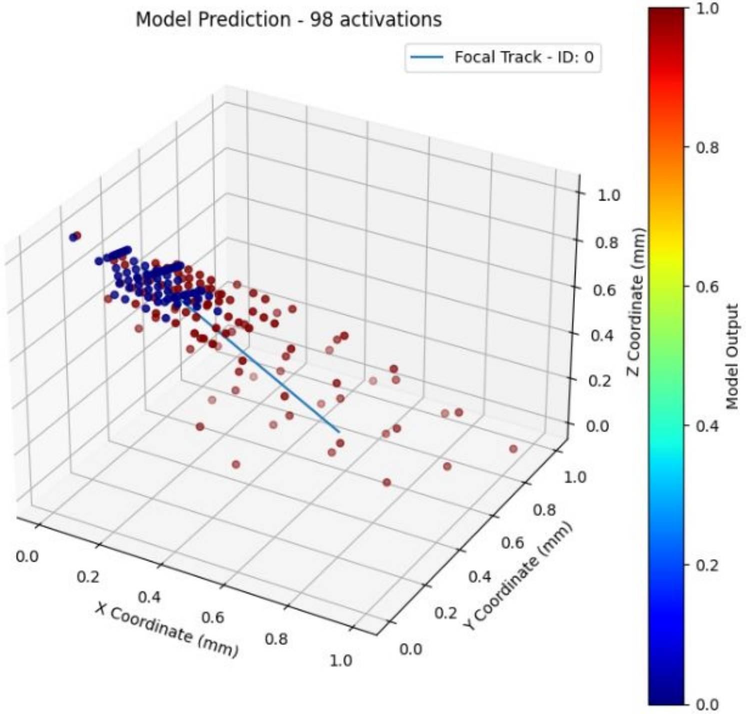
Dataset	Model	Accuracy	F1-score
rho + delta	*PointNet*	0.83	0.78
	smart_random	0.63	0.37
JZ2 + JZ3 + JZ4	*PointNet*	0.94	0.28
	smart_random	0.94	0.06

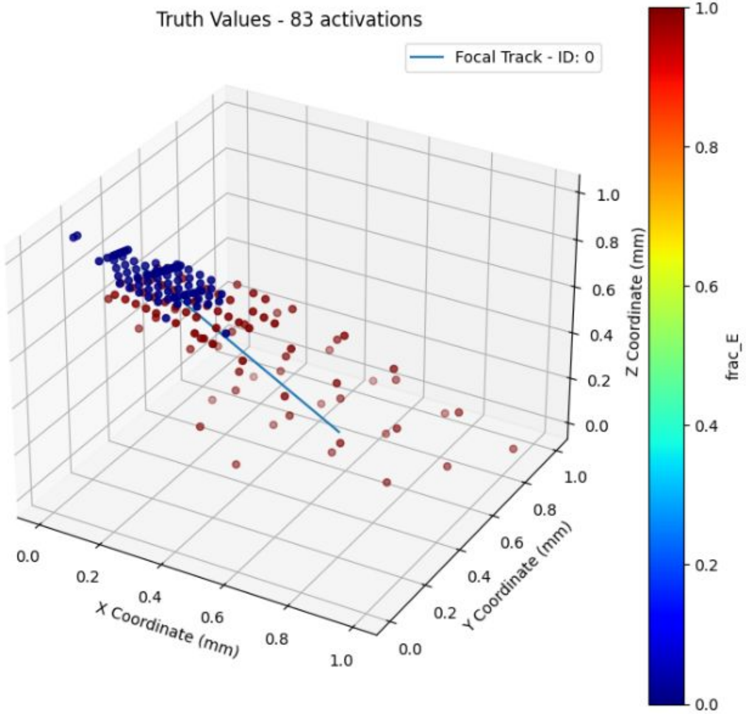




Example of a successful segmentation - rho

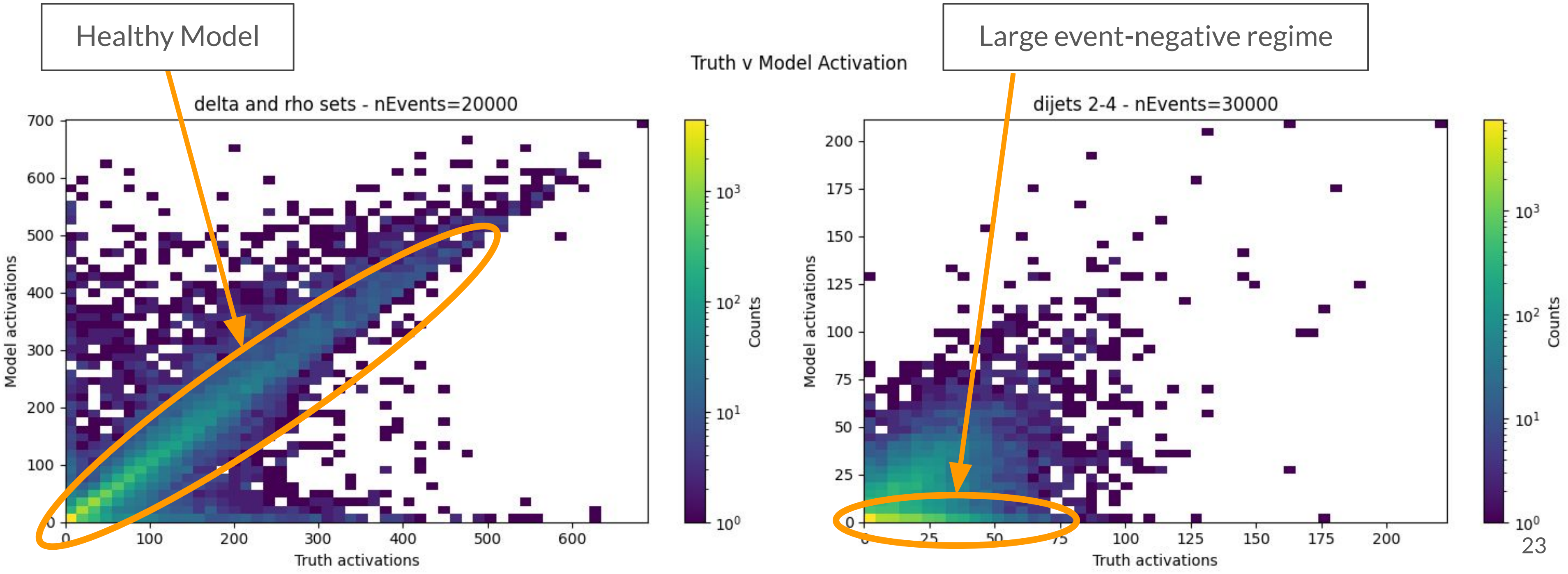
Model activated energy = 1179 GeV ≈ Ideal Activation = 1159 GeV





Modal Collapse

Models tend to become event-wise rather than cell wise for more complex sets.



Ongoing Work

Transfer Learning - Architectural

Attempting to transfer the positive results of the rho set to the dijet sets by freezing parameters to freeze model knowledge.

Strong performance in the rho set

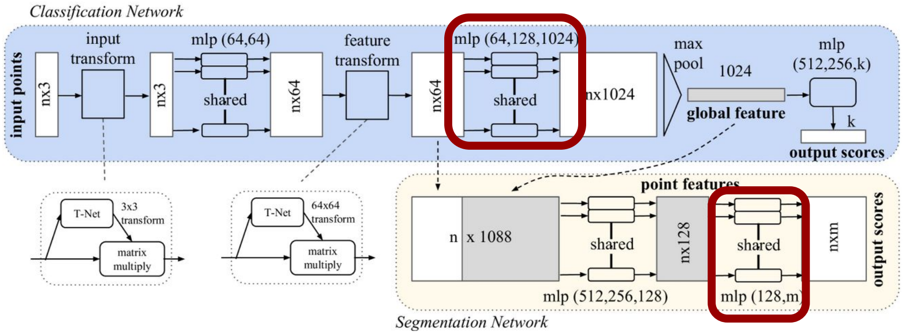
Train to convergence with rho dataset → **Freeze parameters of TNet and base layers** → **Train to convergence with dijet dataset**

Weaker performance in both sets (so far!)

Classification Net: ~4M Params -> (200-500)k Params

Segmentation Net: ~1M Params -> (25-100)k Params

More hyper-parameter sweeping is needed.

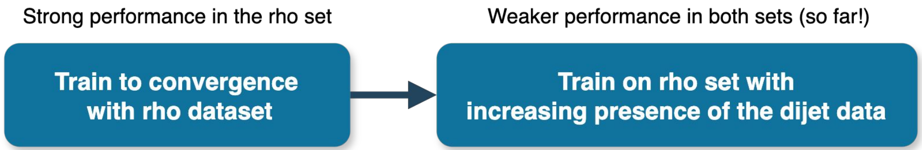


Tuned Parameters

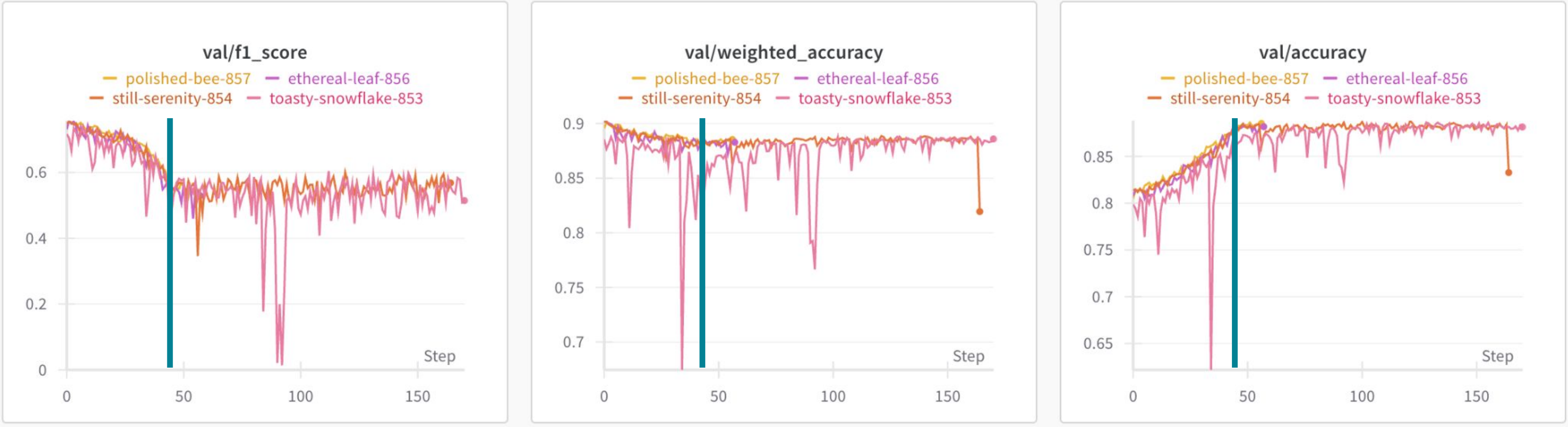
Transfer Learning - Replay

Attempting to transfer the positive results of the rho set to the dijet sets by tuning with both the source and target sets.

Model appears to be a linear combination of a directly trained rho+delta and dijet model. More hyper parameter sweeping needed.

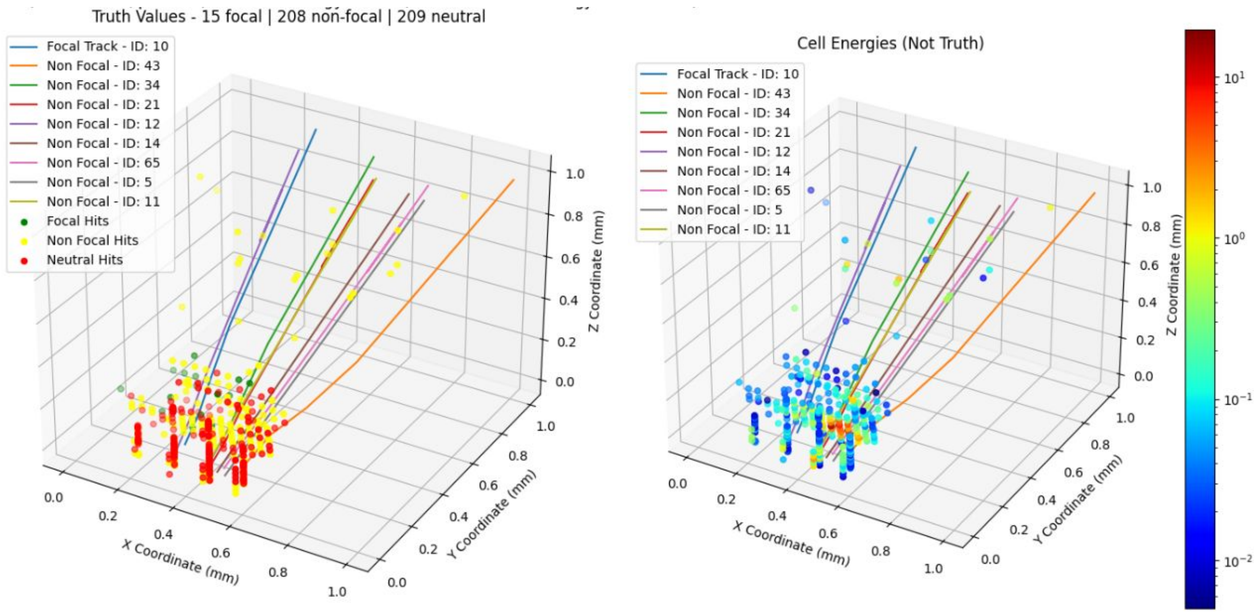


End of data injection phase



Multi-Class Segmentation to Reduce Confusion

Instead of binary class labels, use categorical information for focal, non focal, and neutral. We are hoping to differentiate the negative class to reduce modal collapse.

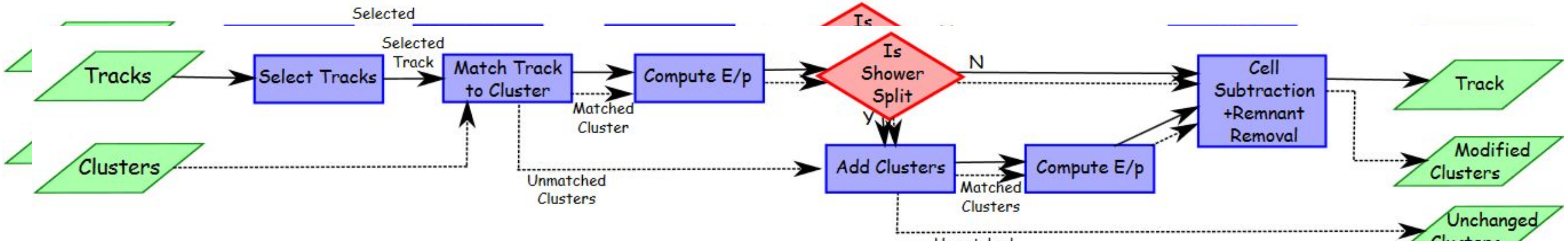


Backup

ATLAS p-flow algorithm [ATL-PERF-2015-09]

For track in descending pT:

- associate closest topo-cluster based on angular distance ΔR’
- compute expected energy deposit based on the topo-cluster position and track momentum
- if expected and measured energies differ significantly, associate more topo-clusters
- subtract the expected energy by calo cells
- if remaining energy lies within expected fluctuations, remove the remnants



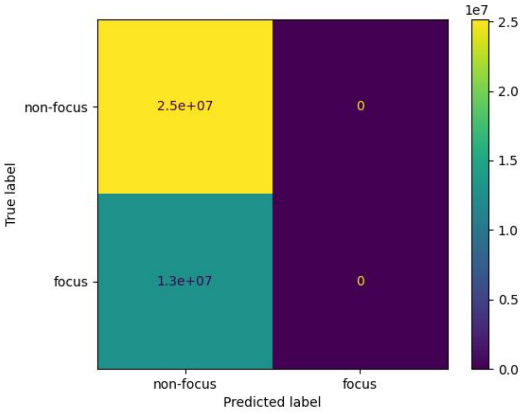
Baseline

Whole datasets results considering all hits except padding

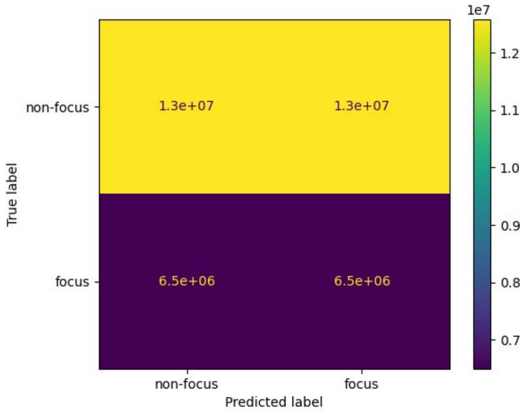
Dataset	Model	Accuracy	Precision	Recall	F1-score
rho + delta	*dumb*	0.6488	0.0000	0.0000	0.0000
	random	0.5001	0.3512	0.4999	0.4126
	smart_random	0.5443	0.3512	0.3511	0.3512
JZ2 + JZ3 + JZ4	*dumb*	0.9414	0.0000	0.0000	0.0000
	random	0.5000	0.0586	0.5000	0.1049
	smart_random	0.8897	0.0586	0.0586	0.0586

Confusion matrix

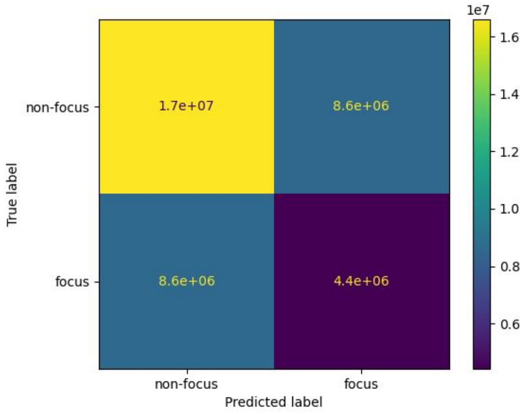
dumb model



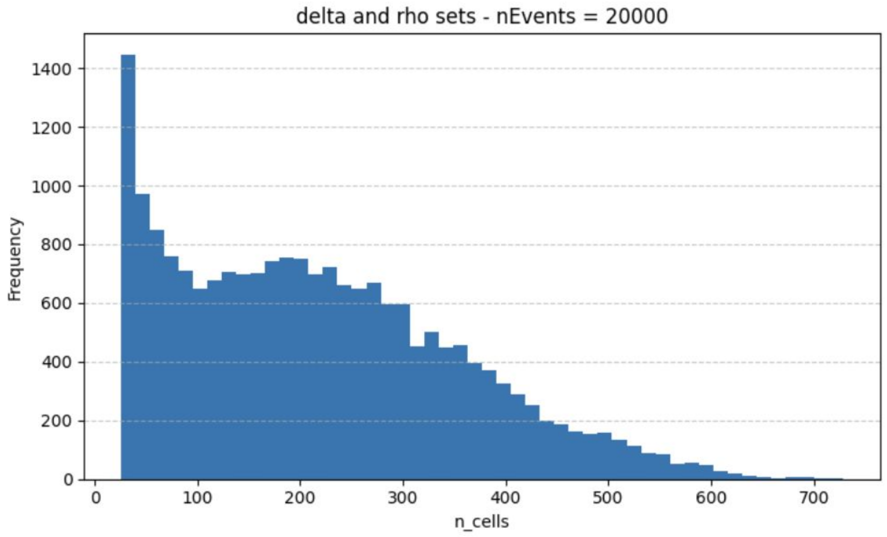
random model

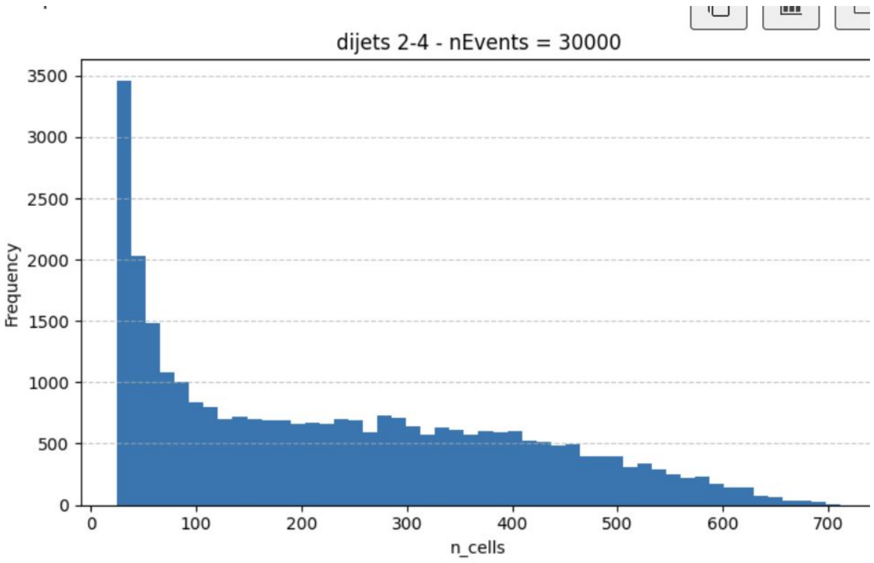


smart random model

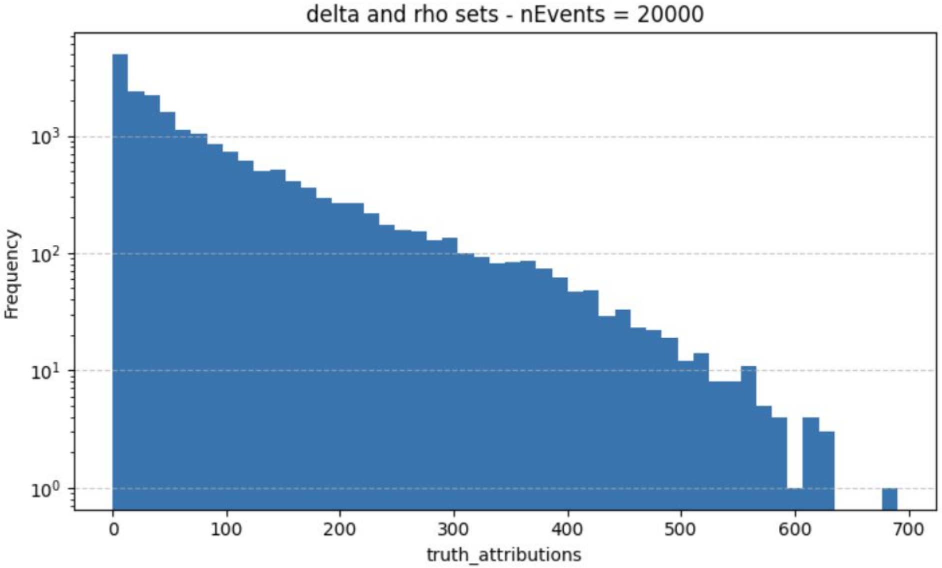


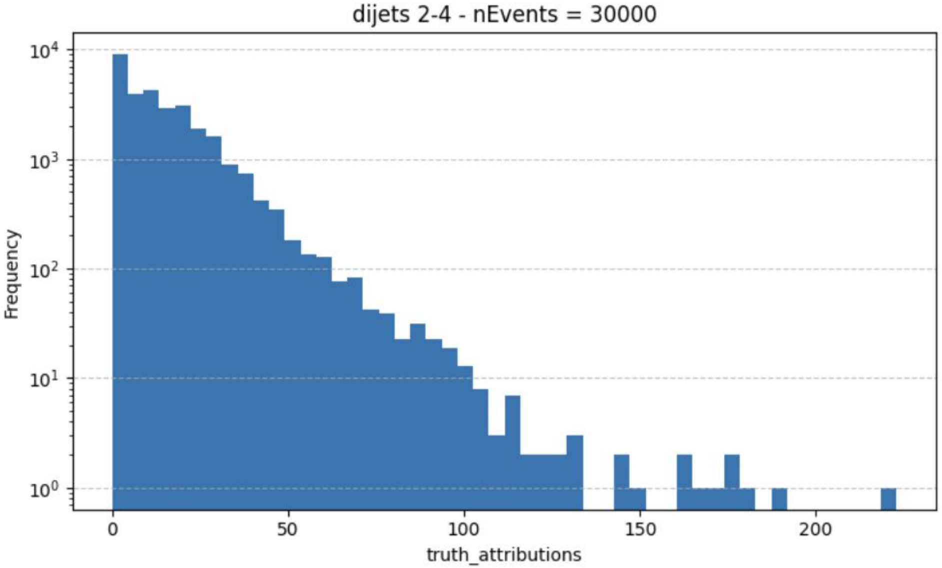
n_cells


delta and rho sets - nEvents = 20000
Frequency
n_cells


dijets 2-4 - nEvents = 30000
Frequency
n_cells

truth_attributions





Dataset	Model	Accuracy	Statistical Certainty (F1)
Simplified Data	*PointNet*	0.83	0.78
	Random Estimator	0.63	0.37
Realistic Data	*PointNet*	0.94	0.28
	Random Estimator	0.94	0.06