Neural network for He event reconstruction quality

Antimatter meeting 21.12.2023

Francesco Rossi, Prof. P. Zuccon

Work strategy:

The goal:

Develop a fully connected neural network capable to identify well reconstructed events with charge 2.

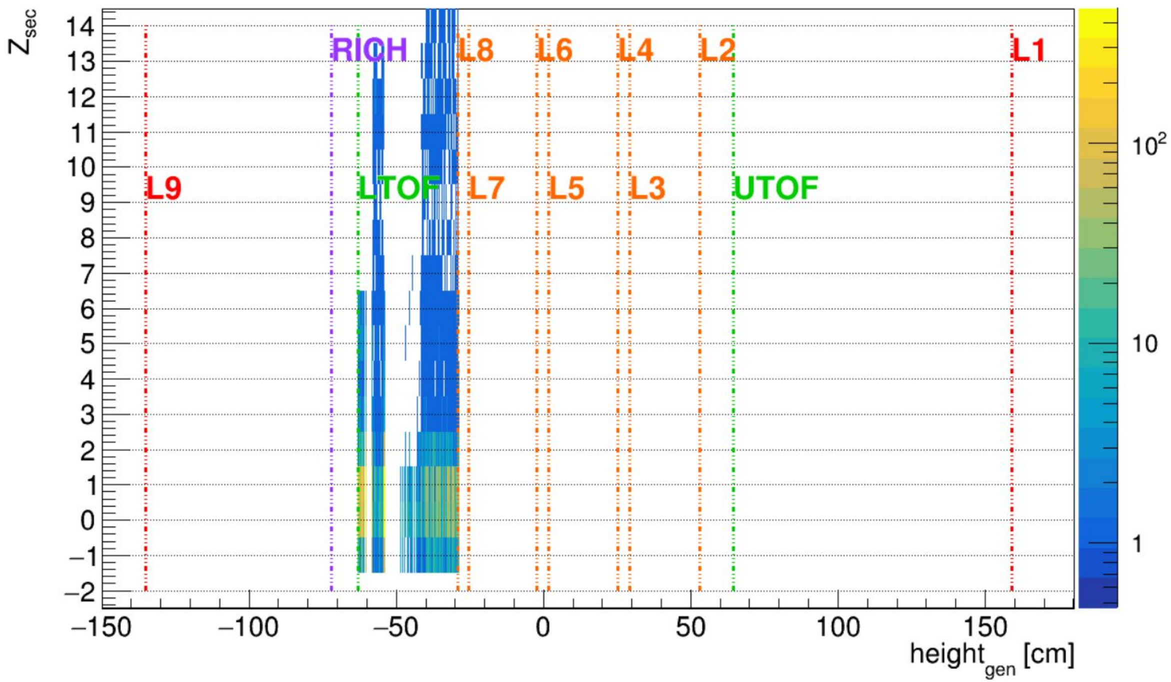
Work strategy updates:

1. Use ^{4}He Monte Carlo truth to define the good (signal) and bad (background) reconstructed events.
2. New dataset with two more background types: “large” variations in parallel and transverse momentum.
3. Correlation between rigidity-related input features.
4. Training the classifier without rigidities absolute values.

Monte Carlo truth

L1-focused MC, **he4.pl1.l1.24000.6_02** with **NAIA v1.1.0 ntuples.**

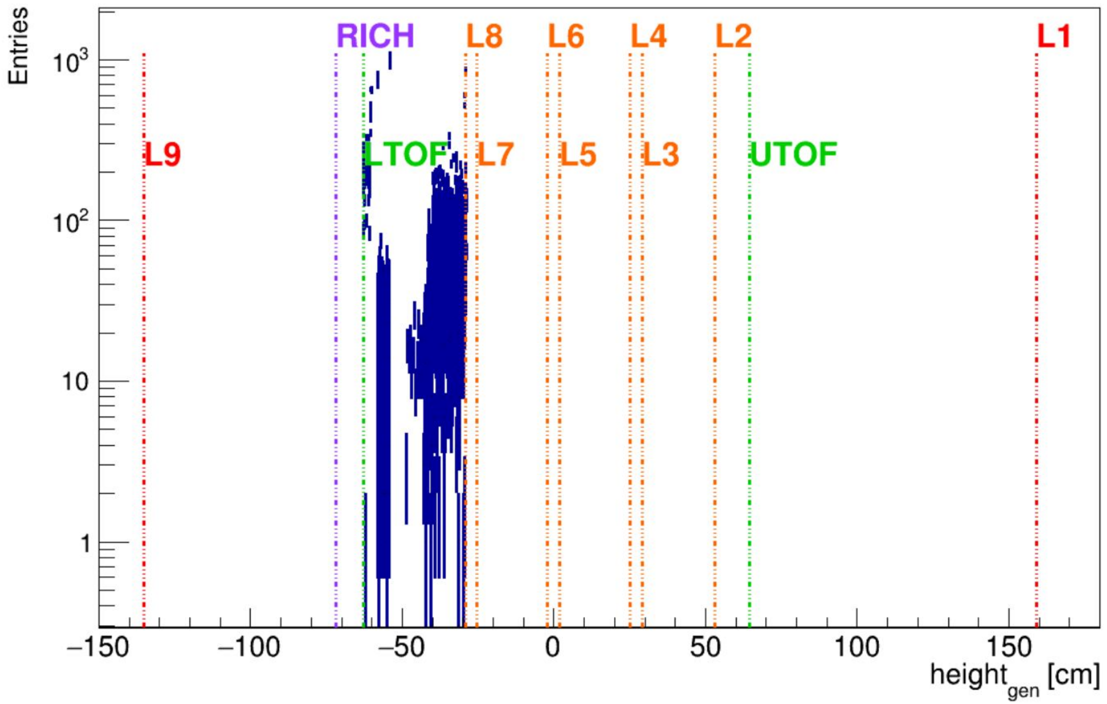
No secondary production above L8.



Monte Carlo truth

L1-focused MC, **he4.pl1.l1.24000.6_02** with **NAIA v1.1.0 ntuples.**

No secondary production above L8.

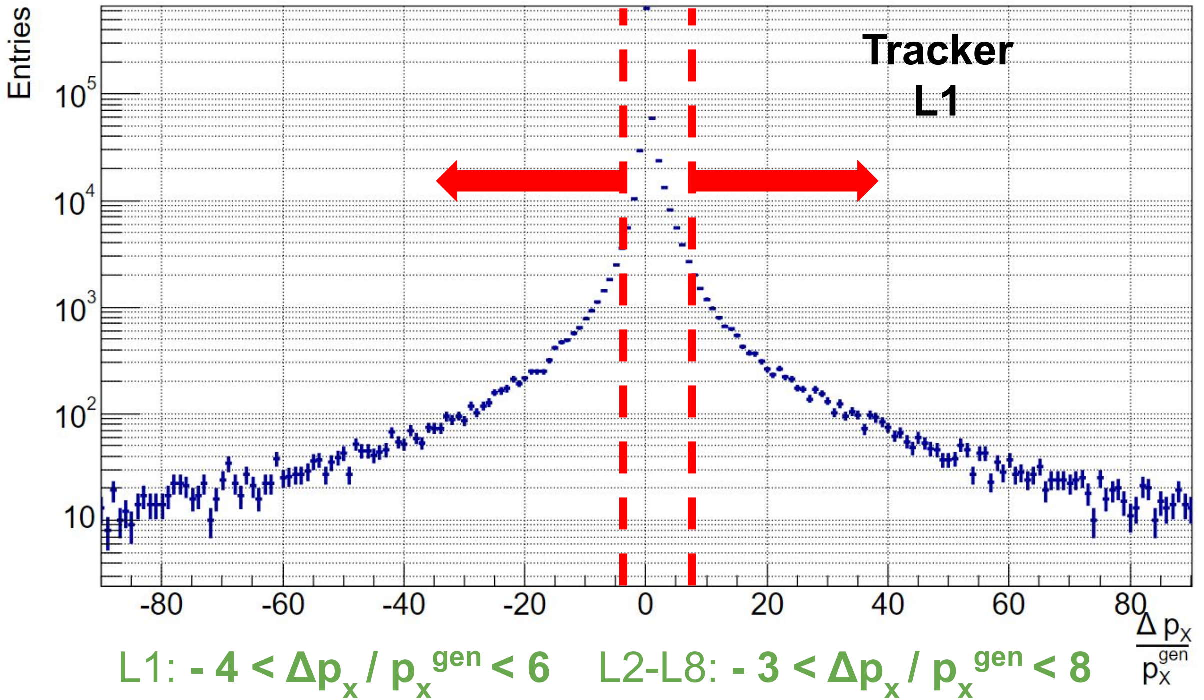


Monte Carlo truth

L1-focused MC, **he4.pl1.l1.24000.6_02** with **NAIA v1.1.0 ntuples.**

No secondary production above lower TOF.

Percentage variation of parallel momentum at each tracker layer.



L1: **- 4 < $\Delta p_x / p_x^{gen}$ < 6** L2-L8: **- 3 < $\Delta p_x / p_x^{gen}$ < 8**

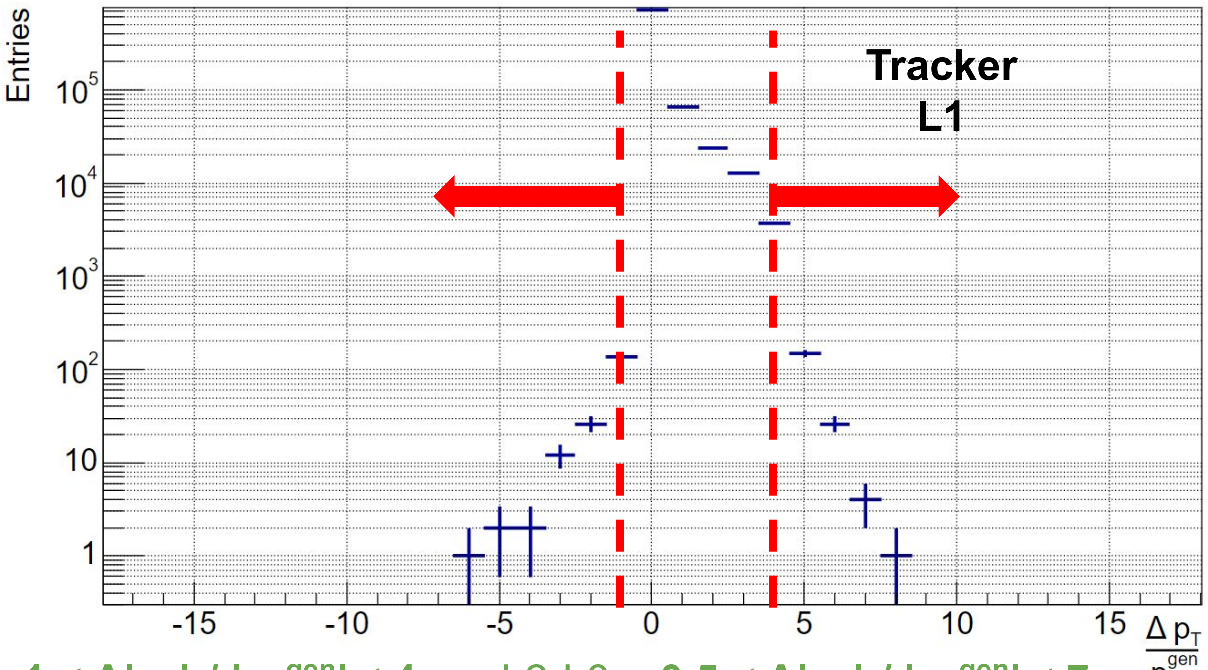
Monte Carlo truth

L1-focused MC, **he4.pl1.l1.24000.6_02** with **NAIA v1.1.0 ntuples.**

No secondary production above lower TOF.

Percentage variation of parallel momentum at each tracker layer.

Percentage variation of transverse momentum at each tracker layer.



L1: **$-1 < \Delta|p_T| / |p_T^{gen}| < 4$** L2-L8: **$-0.5 < \Delta|p_T| / |p_T^{gen}| < 7$**

New signal and background definition

Sign of the reconstructed rigidity (R),inner-L1 using (GBL).

R < 0 → **Background: He events with R<0.**

R > 0 ↓

Check for “large” variation in transverse momentum in at least one tracker layer.

YES → **Background: He events with R>0 and large parallel momentum variation.**

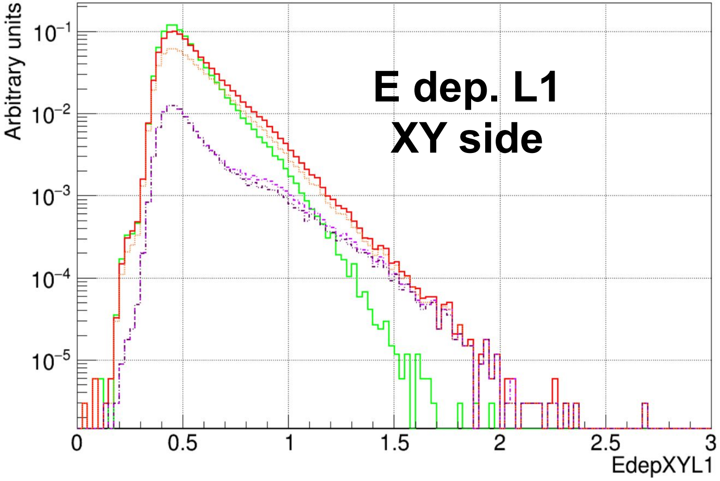
NO ↓

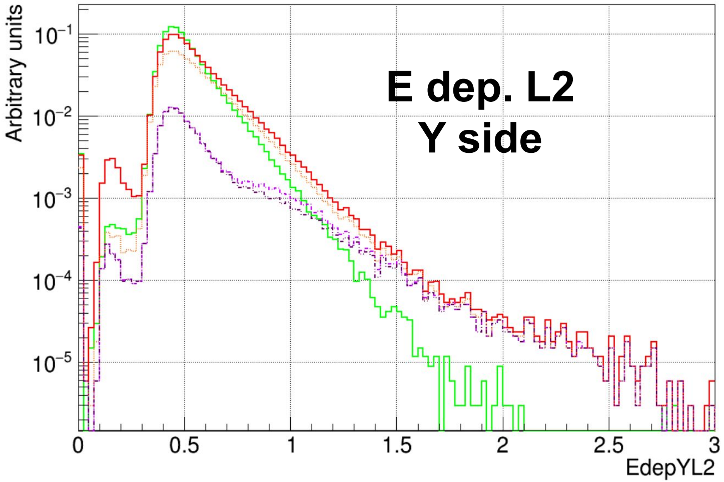
Check for “large” variation in transverse momentum in at least one tracker layer.

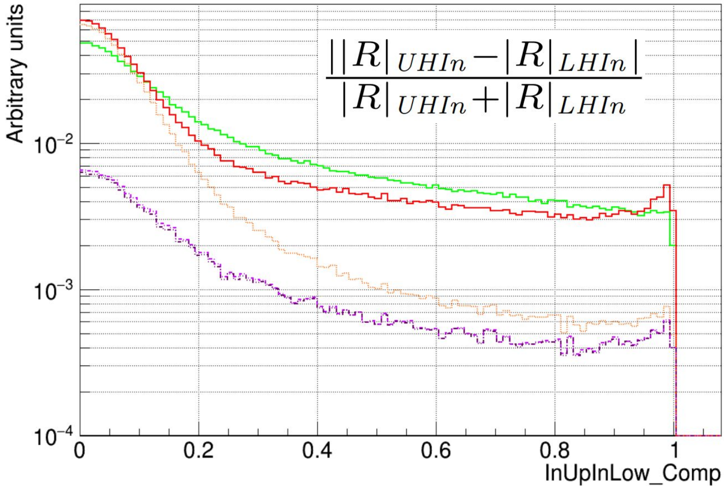
YES → **Background: He events with R>0 and large p_T variation.**

NO ↓

Signal: He events with R>0, “small” parallel momenta and p_T variations







signal

background

P_X bkg

P_T bkg

P_X and P_T bkg

Input features

- Track pattern Y and XY sides.
- Energy deposited in first eight layers of the tracker both Y and XY sides.
- Total number of tracker clusters, X and Y side.

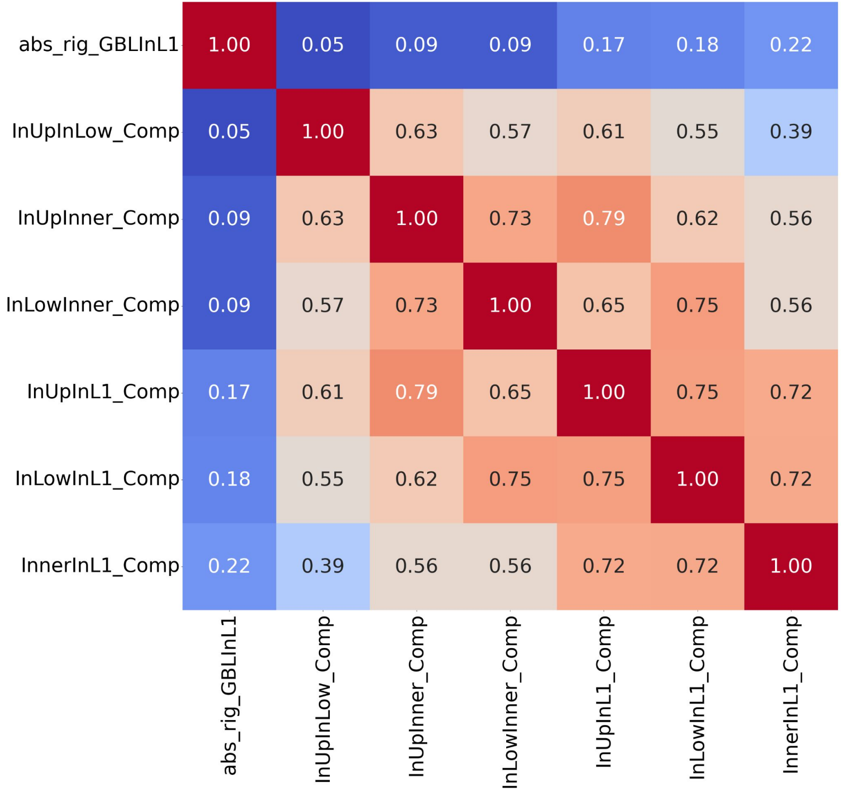
- Track $\chi^2_{X/Y}$ normalised
- Compatibility between different spans: $\frac{||R|_{UHIn} - |R|_{LHIn}|}{|R|_{UHIn} + |R|_{LHIn}}$

- Number of ACC clusters.
- Number of ACC counters.

Total number: 52 input features.

Removed explicit dependencies from rigidity

Correlation between rigidity-related input features



Compatibility between different spans and inner-L1 rigidity.

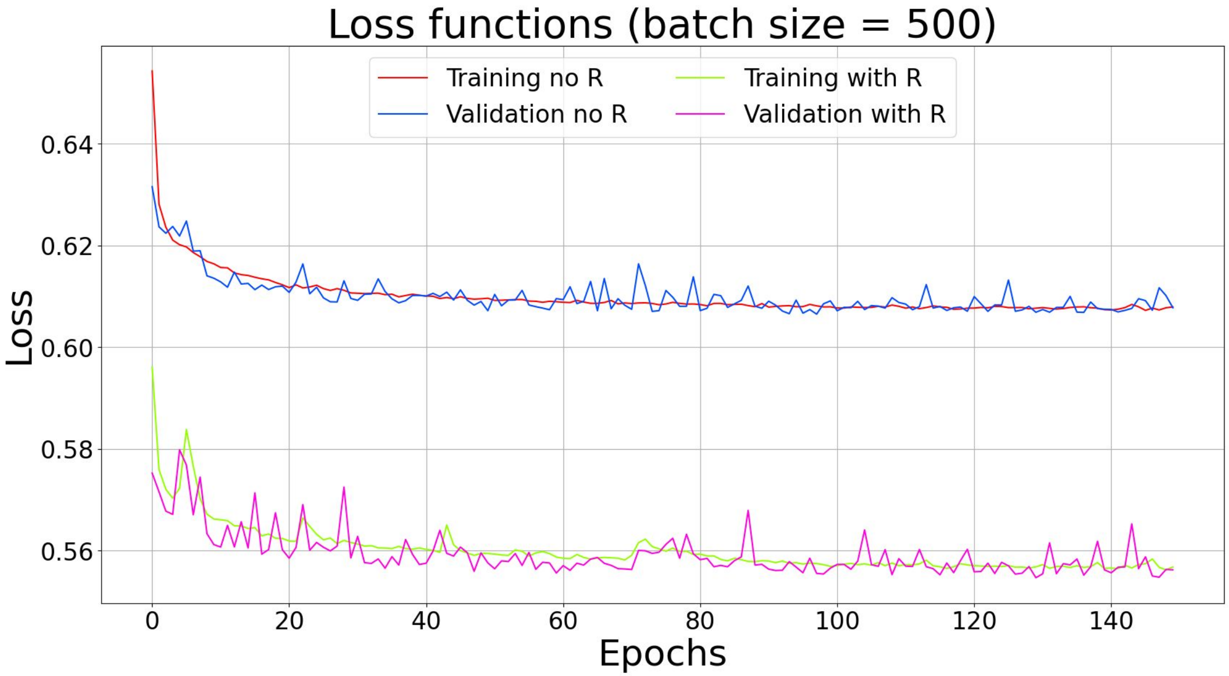
All possible combinations of:

- Inner only
- Inner upper half
- Inner lower half
- Inner-L1

Training and test of the Neural Network

Chosen architecture:

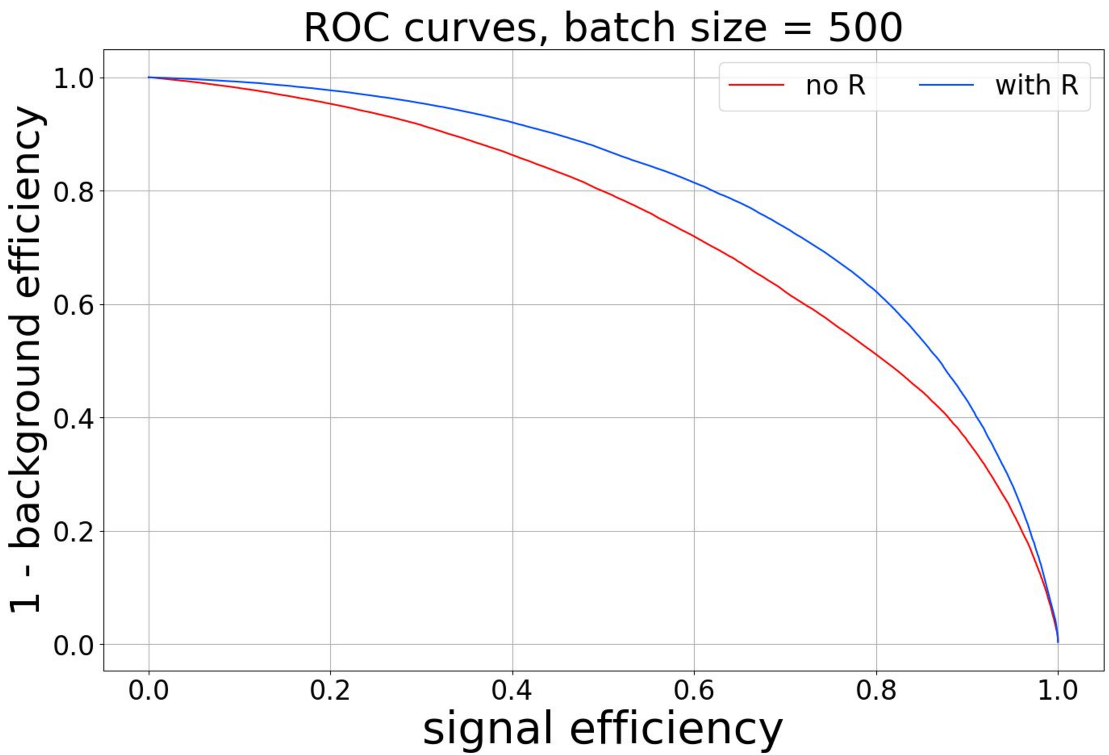
- batch size: 500
- learning rate: 5e-3
- layer multiplicity: 7
- nodes per layer: 64



Training and test of the Neural Network

Chosen architecture:

- batch size: 500
- learning rate: 5e-3
- layer multiplicity: 7
- nodes per layer: 64



Area under the curve = 0.728

Area under the curve = 0.788

Previous results = 0.950

Conclusions

Two new types of background have been added in the dataset.
The Neural Network has been trained avoiding explicit dependencies from rigidity.
Further studies on the dataset before optimize the neural network.

Next steps

- Correlation between all the input features.
- Add variable scaling in data pre-processing.
- Use the MC truth to classify different backgrounds (spill-over).

- MC is generated flat in logarithm, it might be necessary to reproduce the true spectra (upsampling?).
- Study the few background events with a network output ~ 1.

Conclusions

Two new types of background have been added in the dataset.
The Neural Network has been trained avoiding explicit dependencies from rigidity.
Further studies on the dataset before optimize the neural network

A special thanks to
Greta Brianti
&
Francesco Follega
for the help

Next steps

- Correlation between all the input features.
- Add variable scaling in data pre-processing.
- Use the MC truth to classify different backgrounds (spill-over).

- MC is generated flat in logarithm, it might be necessary to reproduce the true spectra (upsampling?).
- Study the few background events with a network output ~ 1.

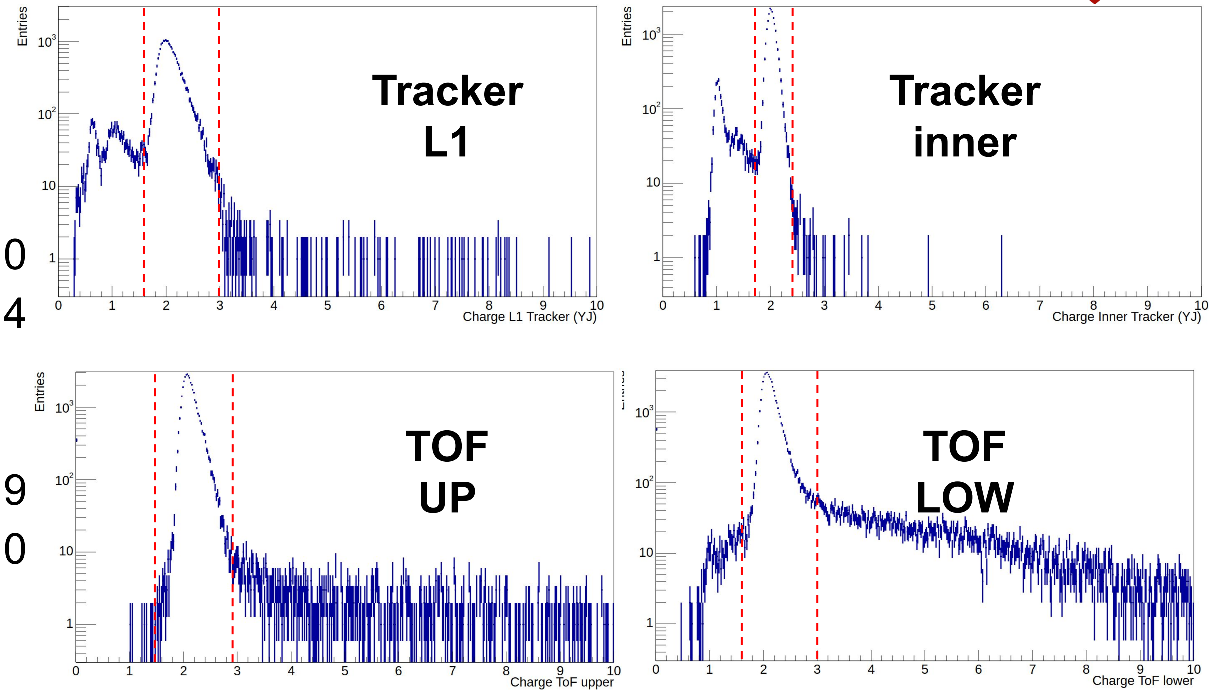
Backup

Monte Carlo samples and data pre-selection

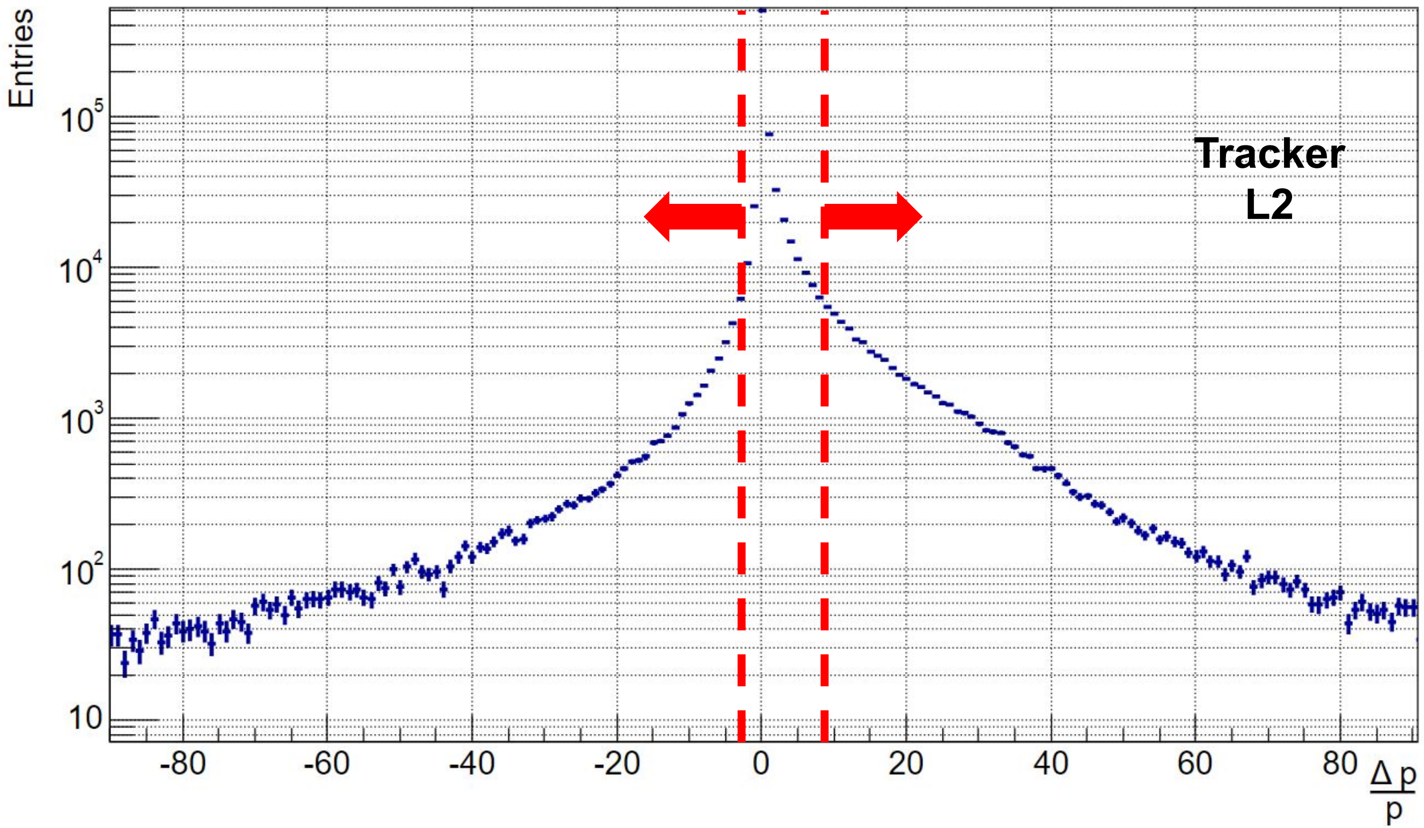
L1-focused MC, **he4.pl1.l1.24000.6_02** with **NAIA v1.1.0 ntuples.**

Skimming selections:

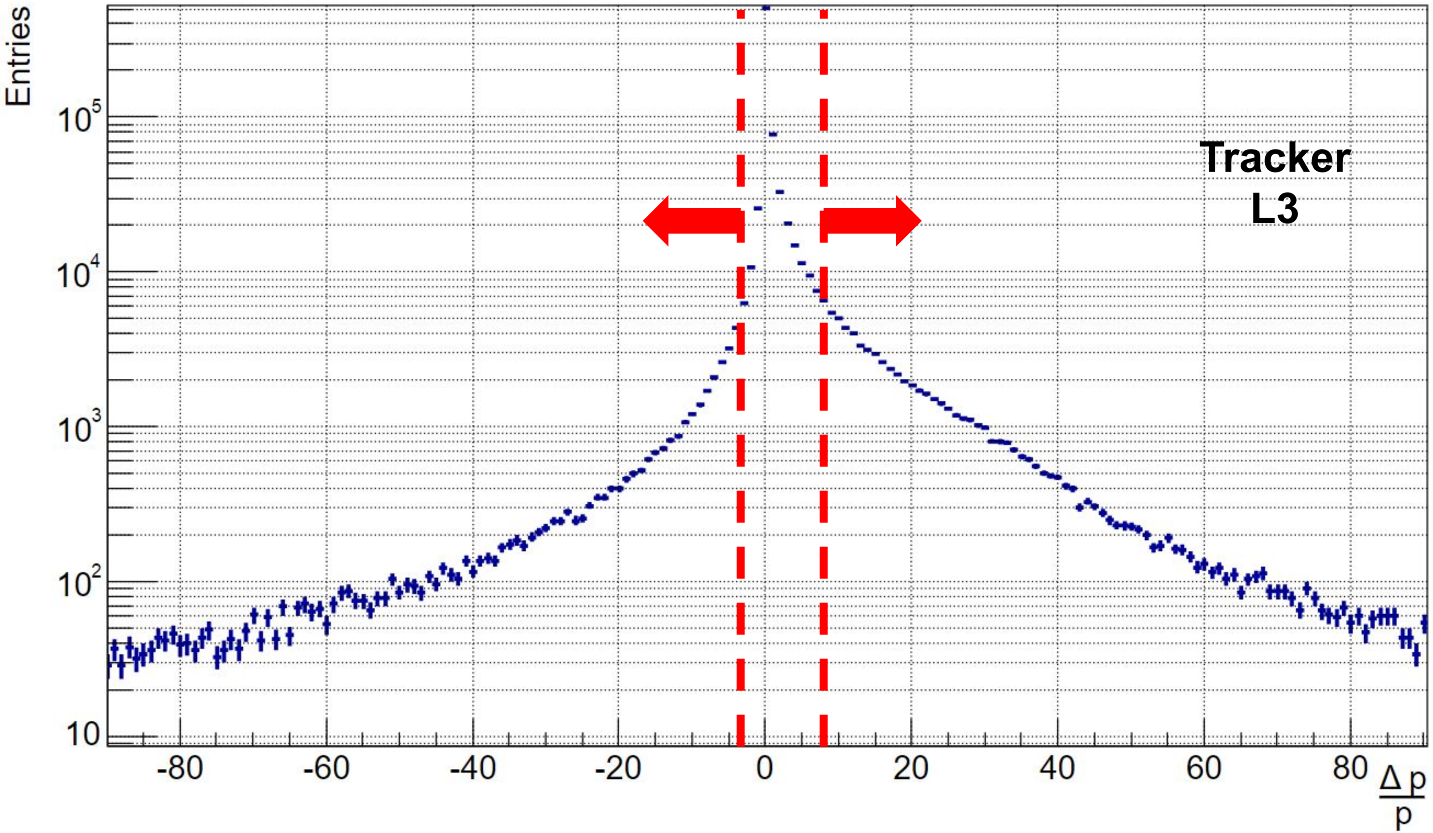
Physics trigger:	true
L1 charge (YJ):	1.6 < |Q| < 3.0
Inner tracker (YJ):	1.7 < |Q| < 2.4
LayerGoodPathl	true
Upper TOF:	1.5 < |Q| < 2.9
Lower TOF:	1.6 < |Q| < 3.0



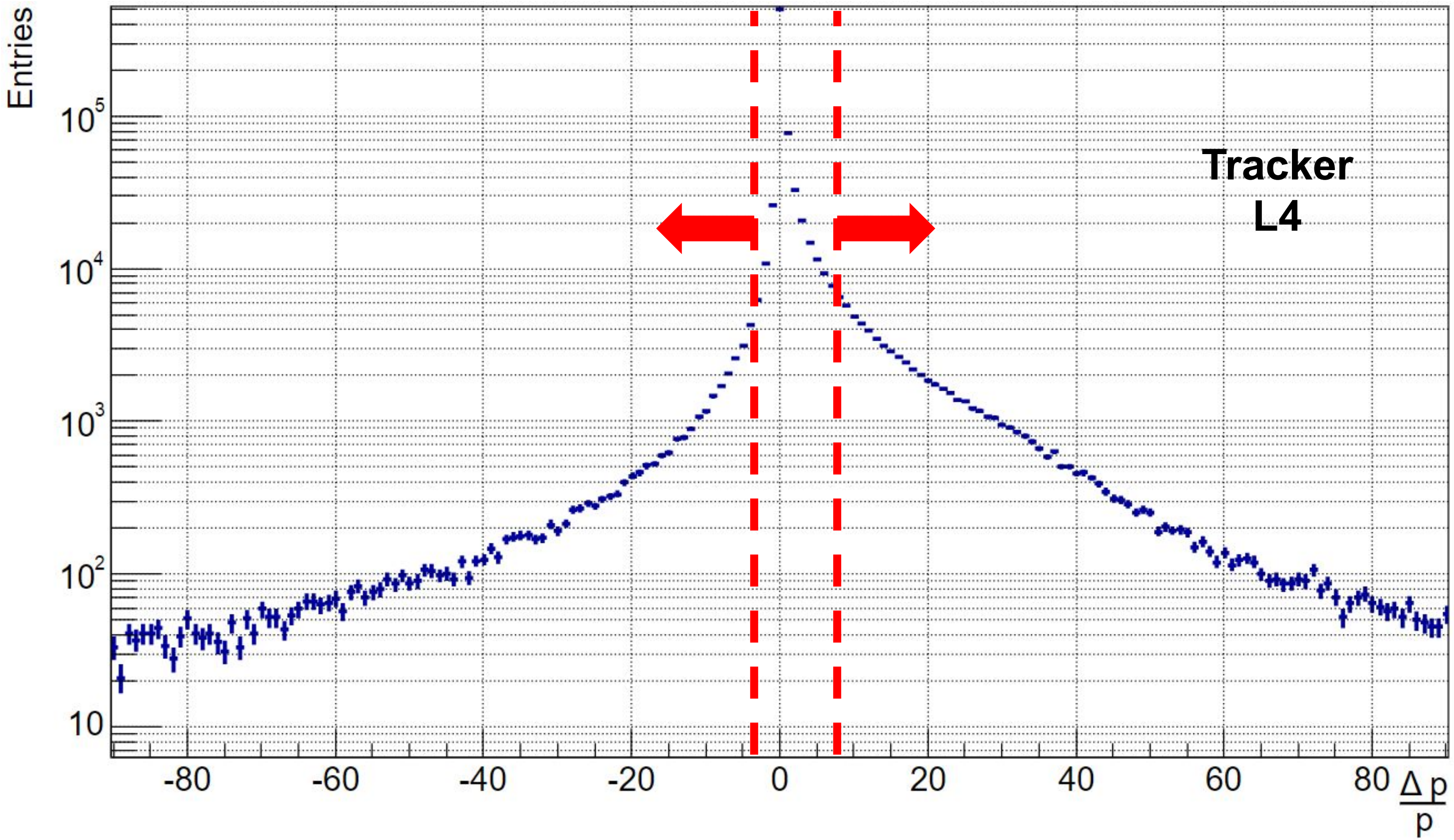
Monte Carlo truth parallel momentum



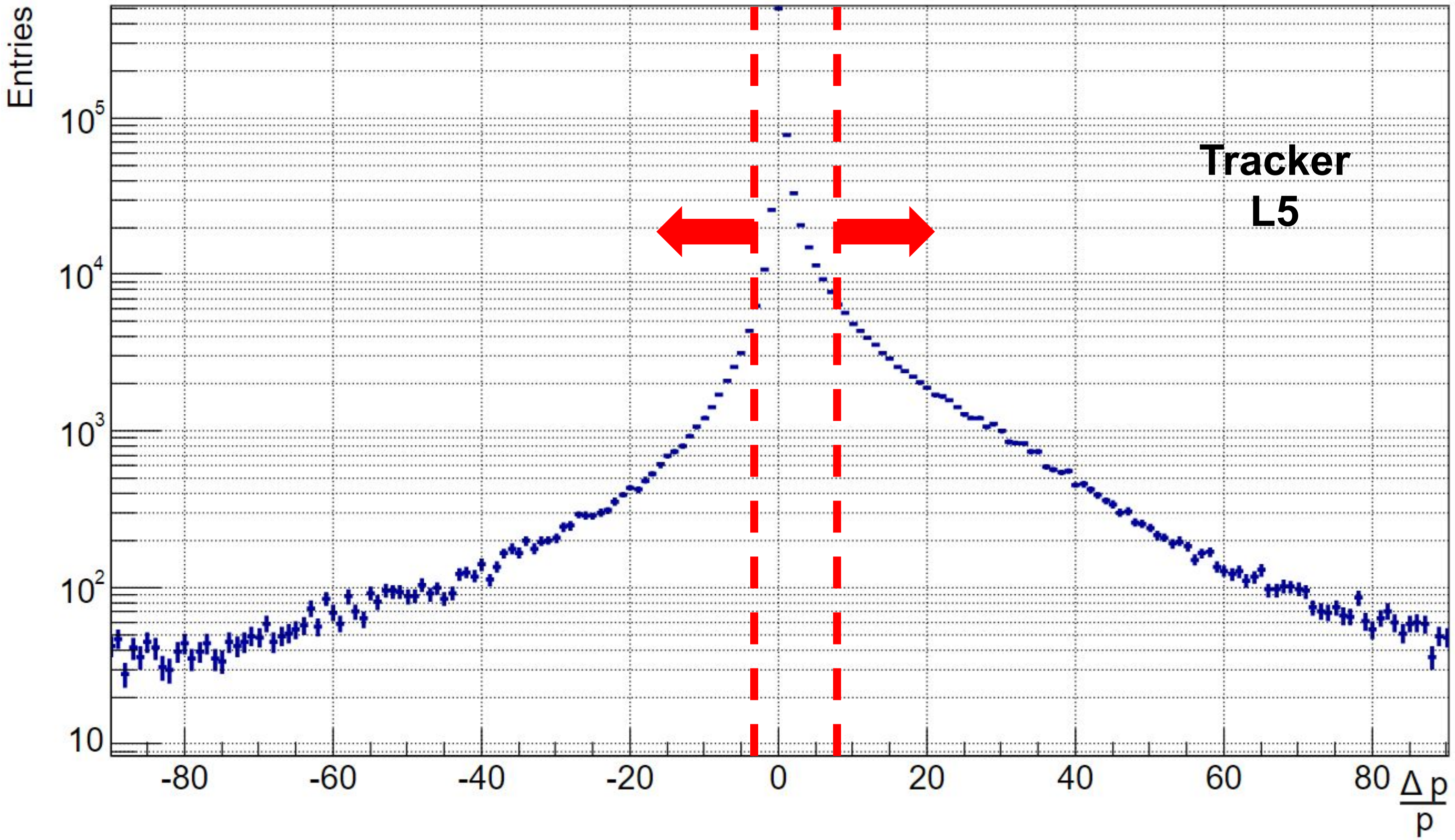
Monte Carlo truth parallel momentum



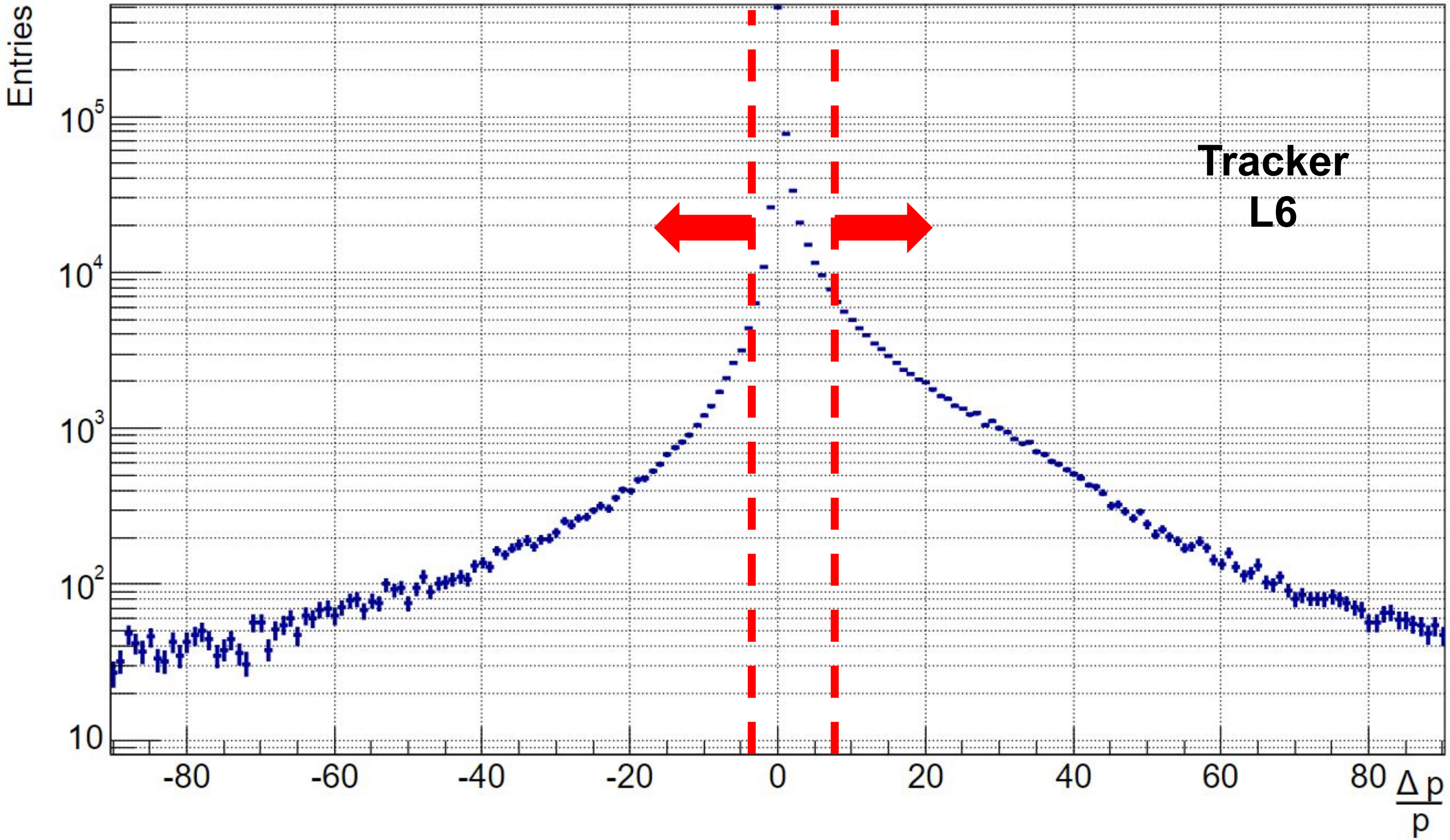
Monte Carlo truth parallel momentum



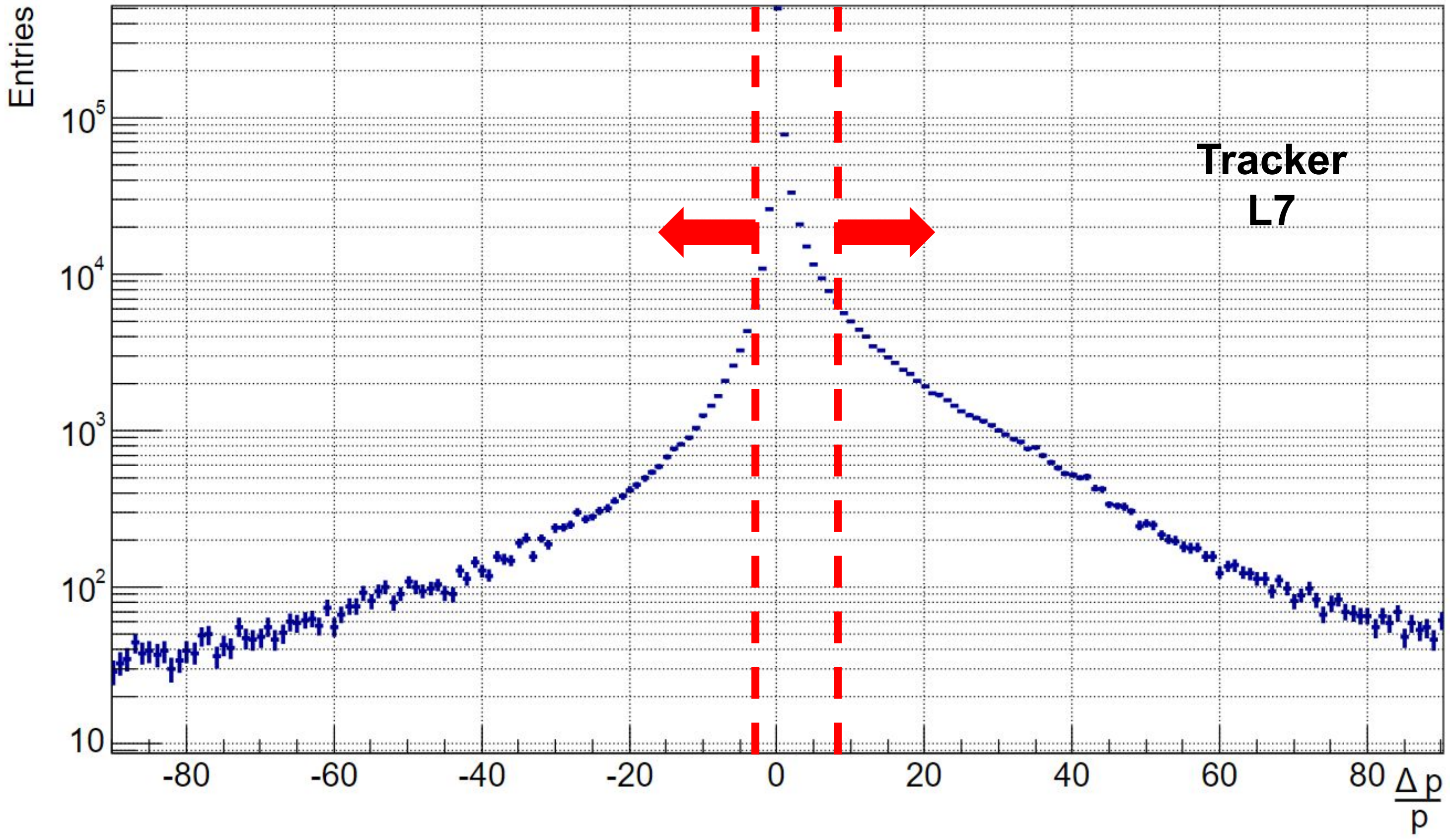
Monte Carlo truth parallel momentum



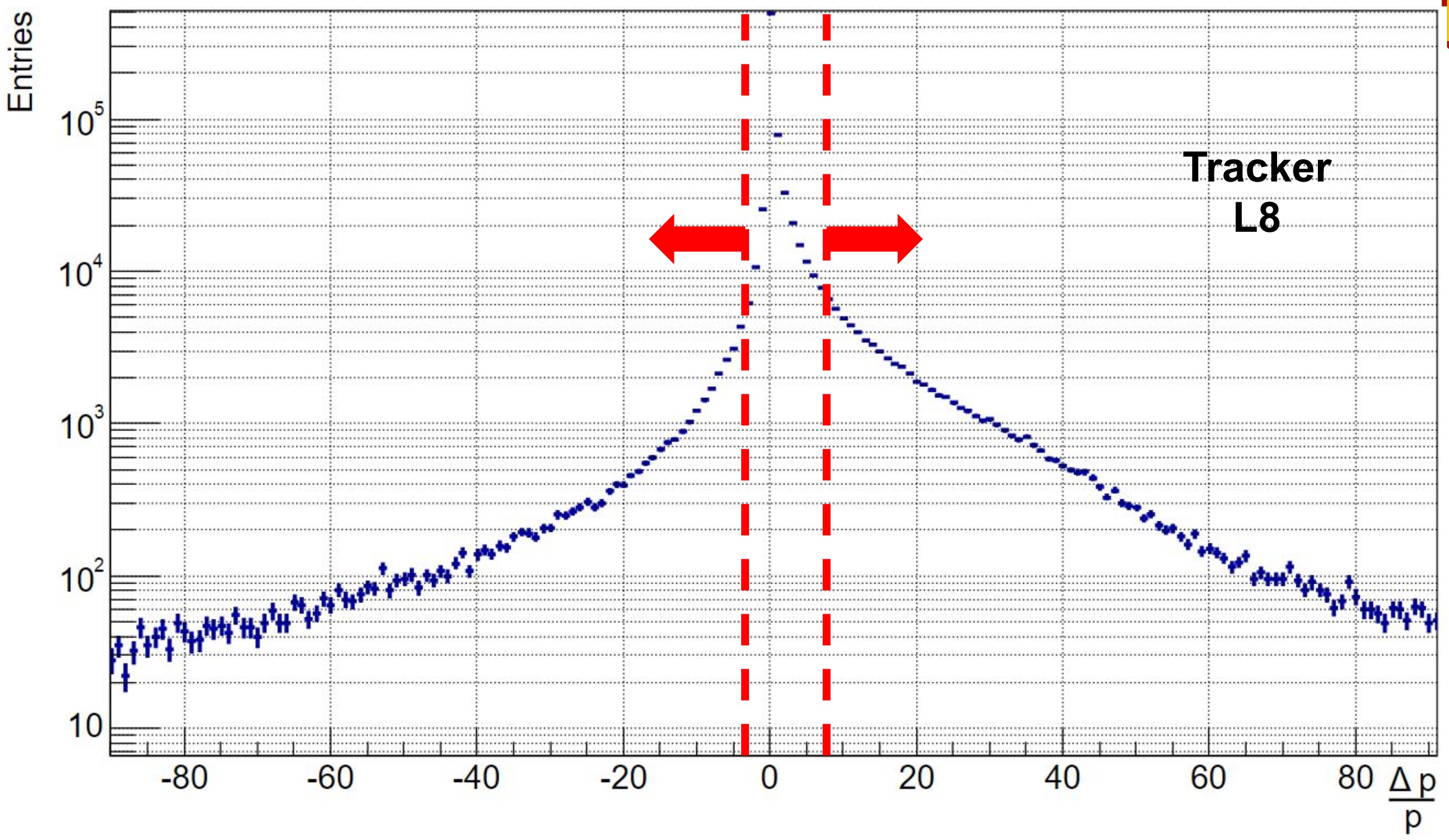
Monte Carlo truth parallel momentum



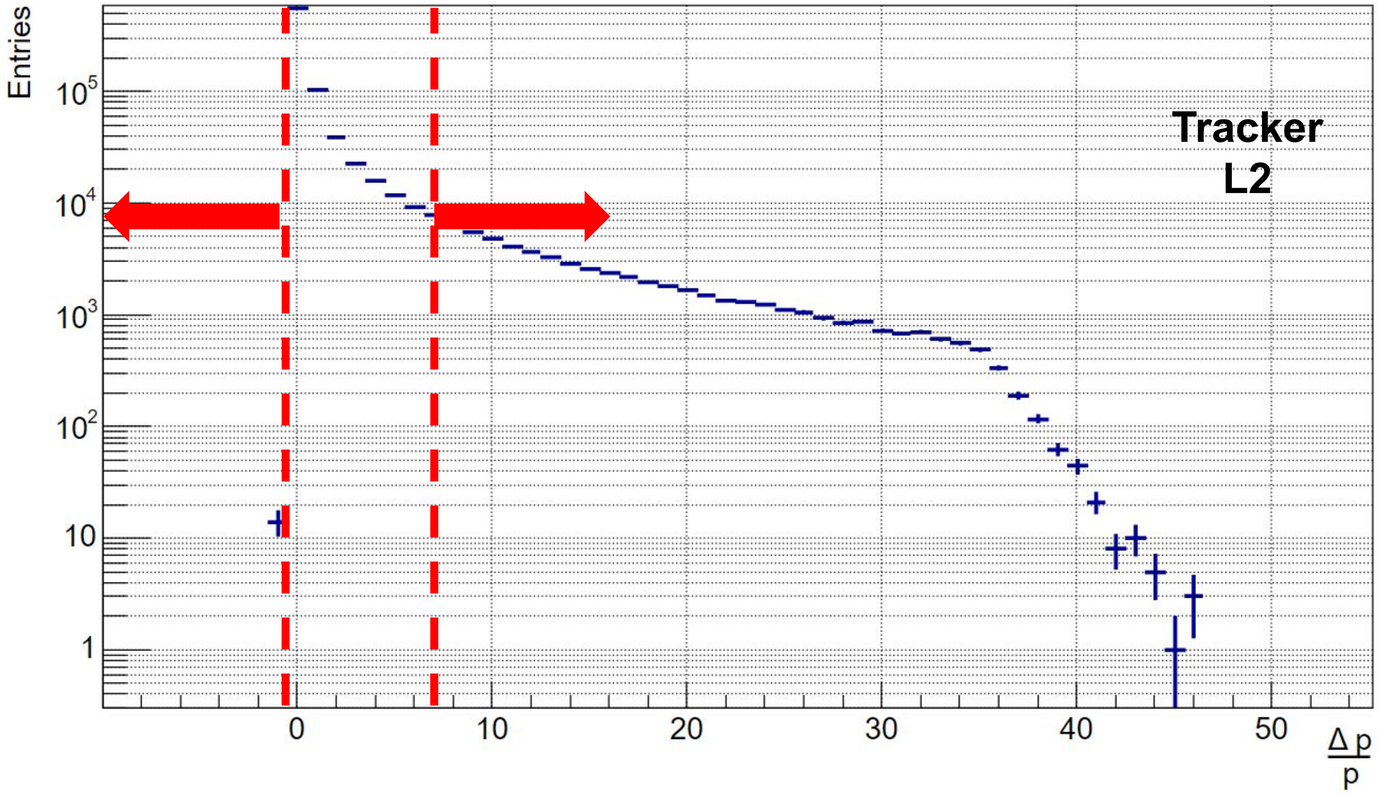
Monte Carlo truth parallel momentum



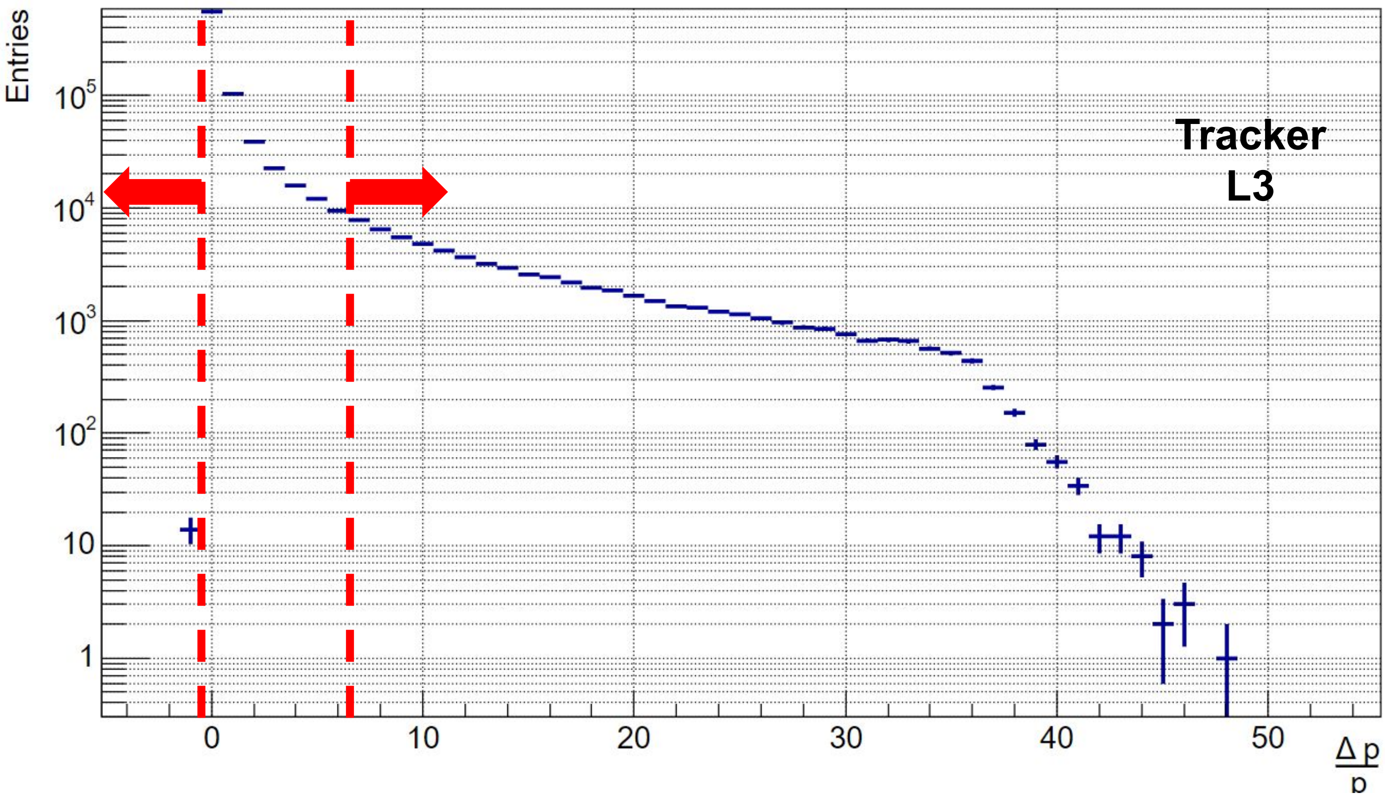
Monte Carlo truth parallel momentum



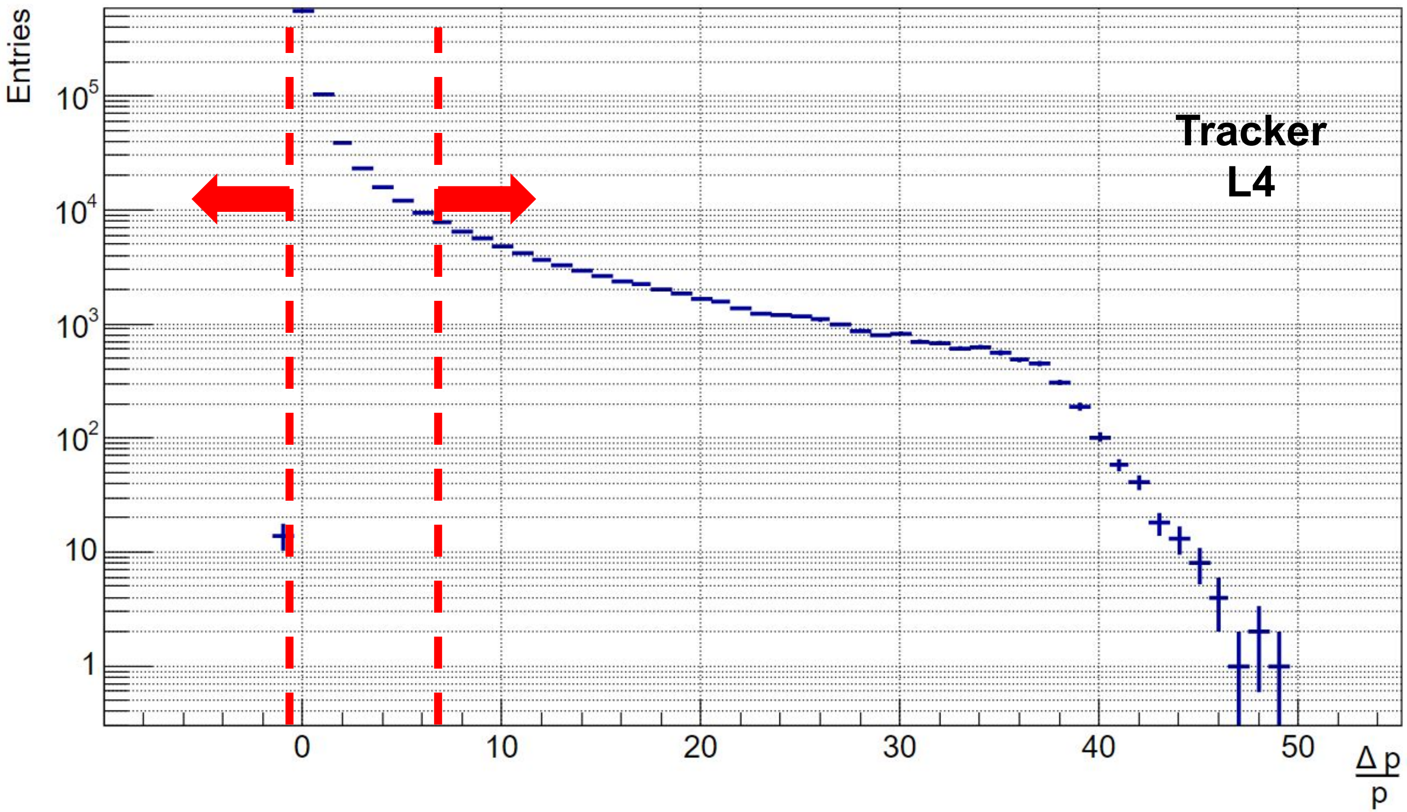
Monte Carlo truth transverse momentum



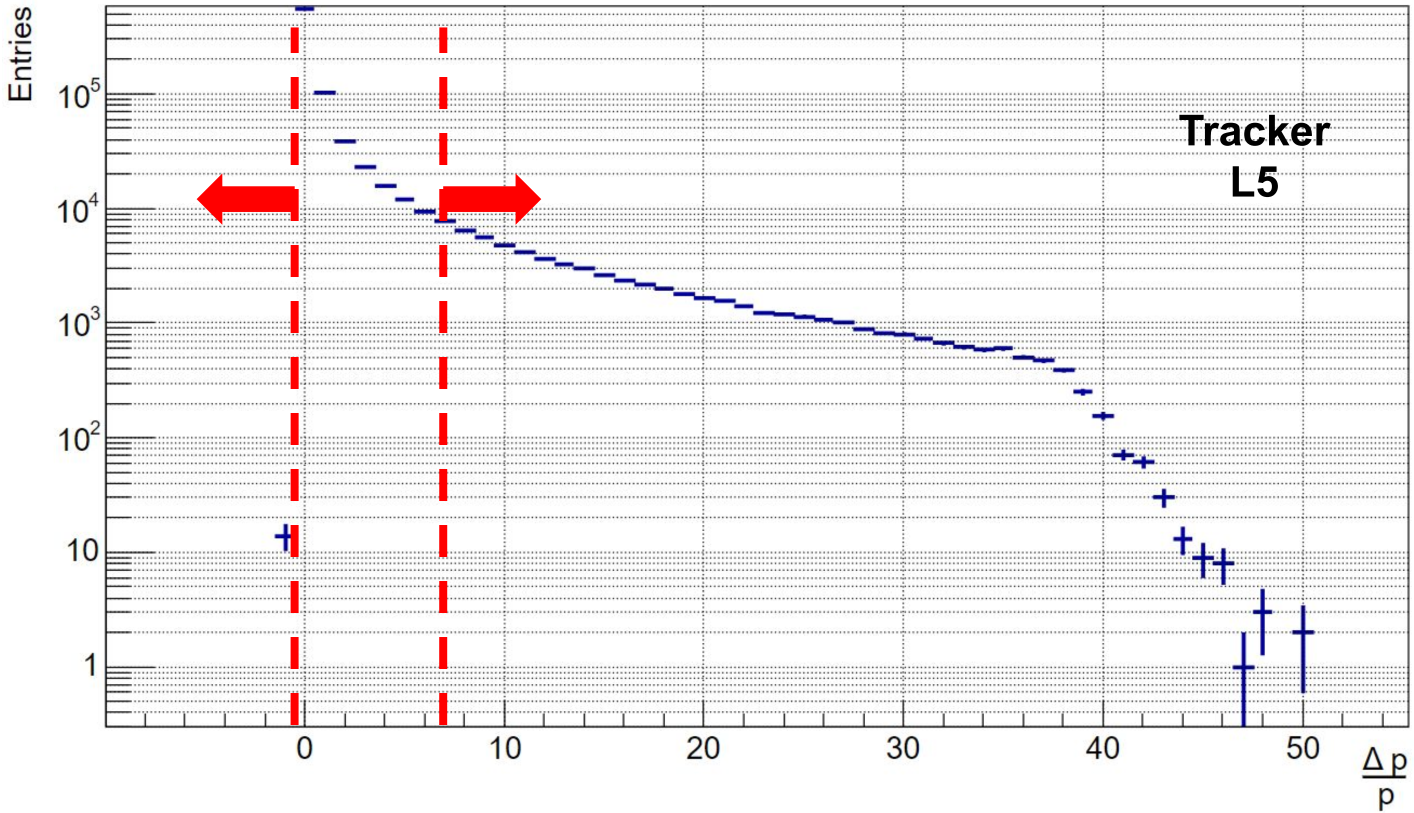
Monte Carlo truth transverse momentum



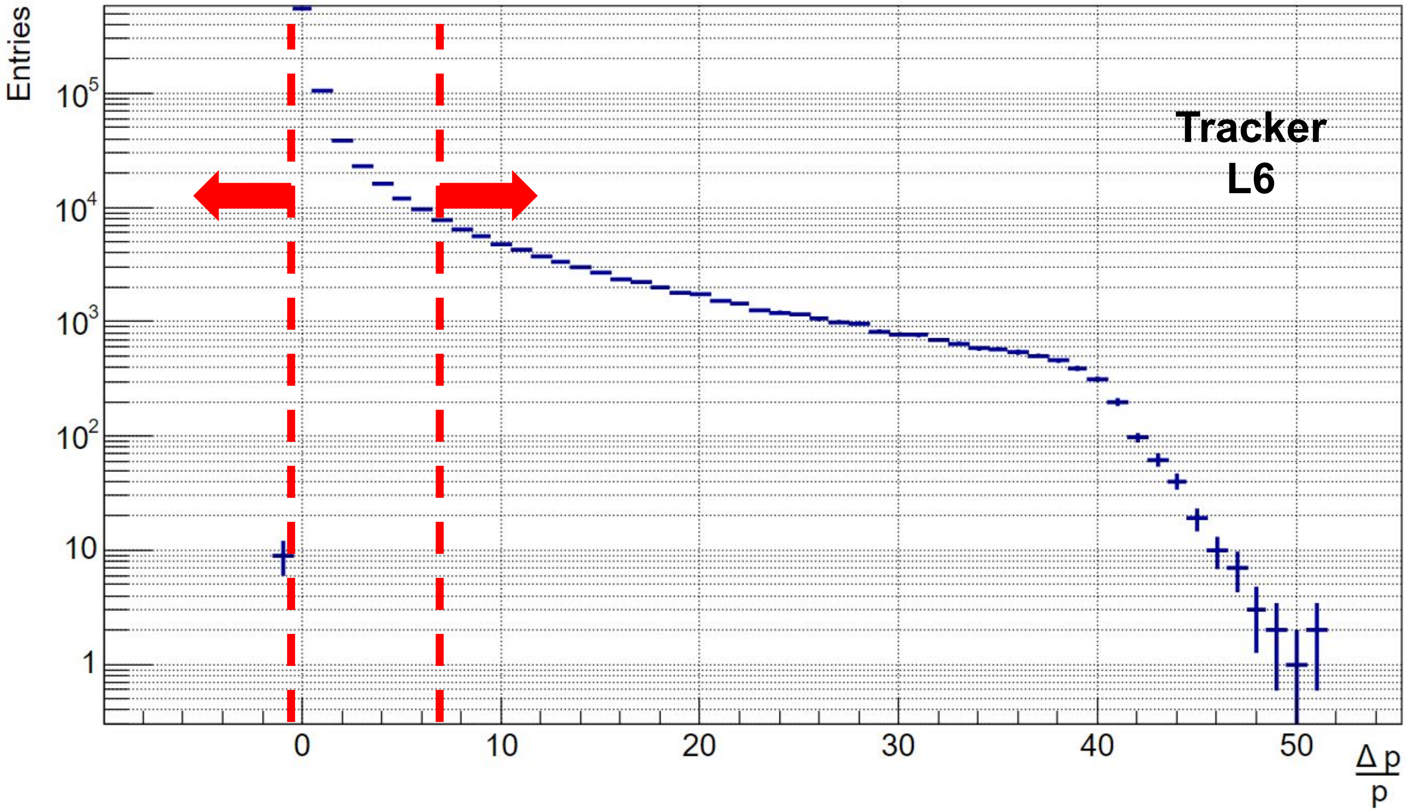
Monte Carlo truth transverse momentum



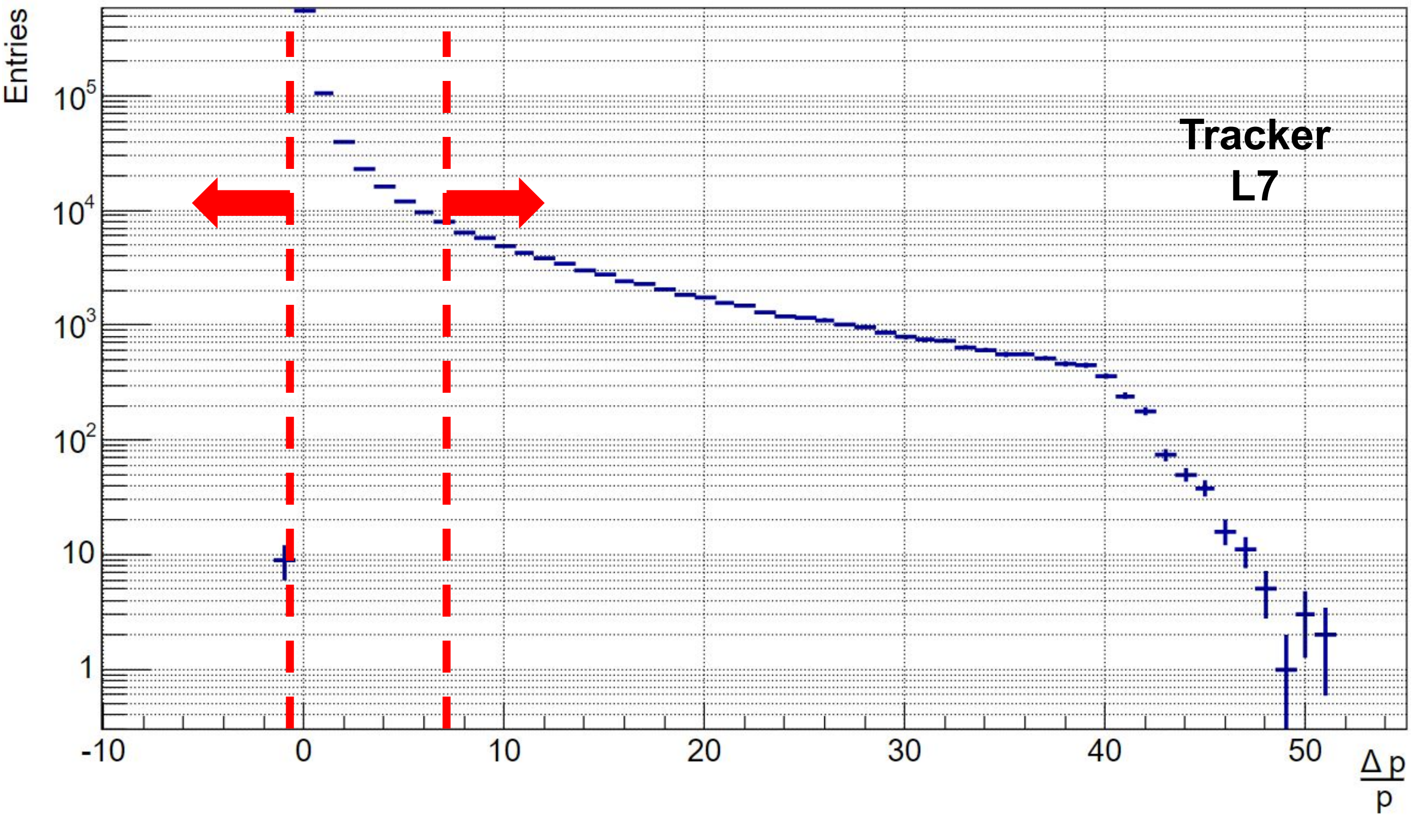
Monte Carlo truth transverse momentum



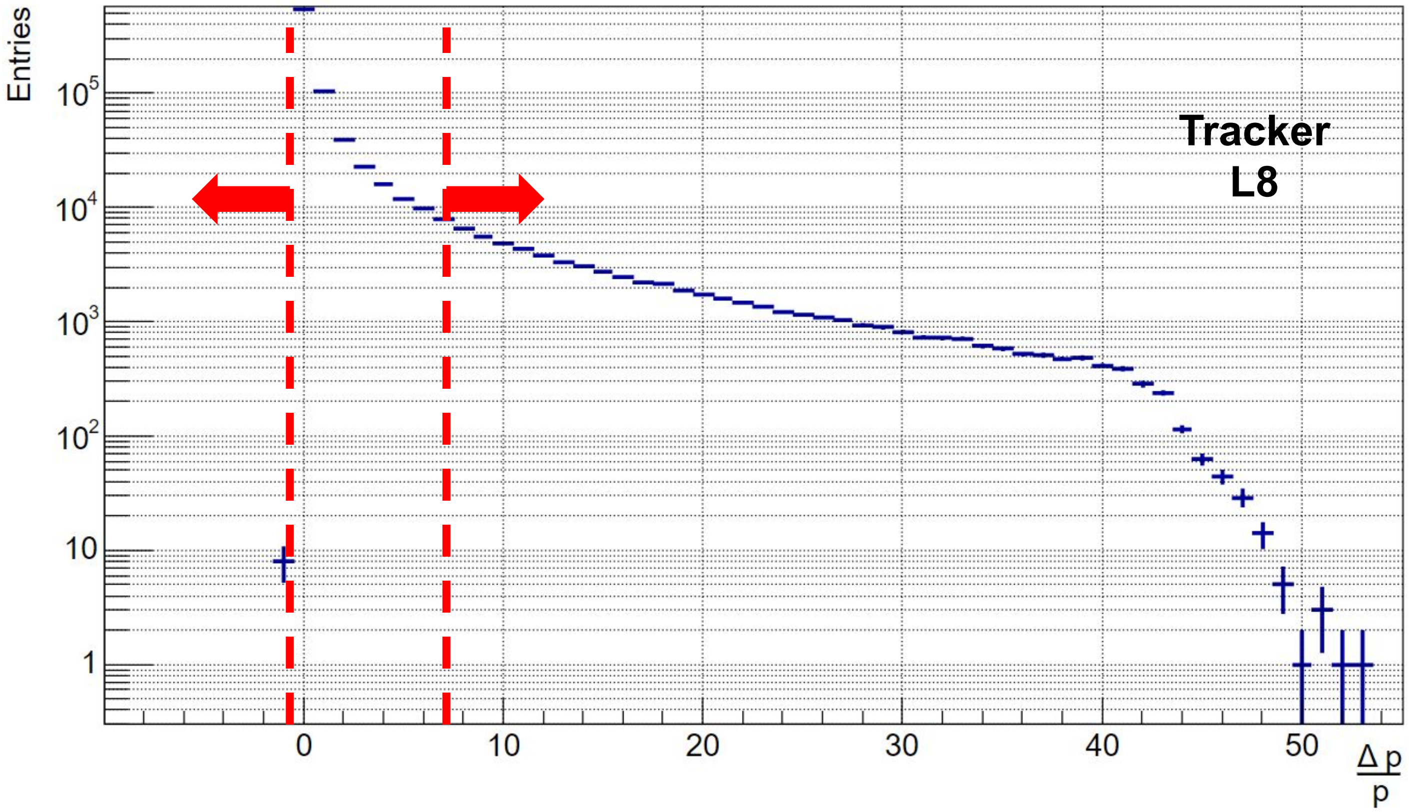
Monte Carlo truth transverse momentum



Monte Carlo truth transverse momentum



Monte Carlo truth transverse momentum



Creation of the dataset

Further requirements:

- Track pattern requirement: one hit (Y or XY side) on 7/8 layers, L1 and L2 must be present.
- The rigidity fit (GBL) must exist for: inner-only, inner-L1, upper and lower inner half.
- $|R|_{In}$ and $|R|_{InL1}$ > 2 GV
- $|R|_{UHIn}$ and $|R|_{LHIn}$ > 1 GV

To avoid a bias, the dataset has the same amount of signal and background events. Only 1 out of 7 signal events is stored, until reaching the same amount of background events.

Total events number ~ 680 K

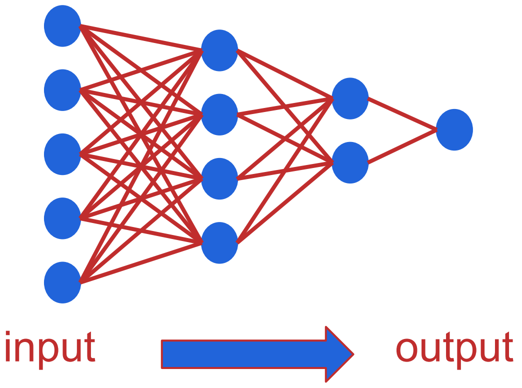
Fully connected neural network architecture

Fully connected neural network (FCNN) based on PyTorch.

The network has a customizable layers number and nodes per layer number.

The output of the network is a number ϵ [0,1].

The activation function is a Rectified Linear Unit (ReLU), and an *“Adam optimizer”* is used. The loss function is a Binary Cross Entropy.



70% of the dataset is used for training, 30% for validation.
The area under the Receiver Operating Characteristic (ROC) curve is used to quantify the performance of the network.

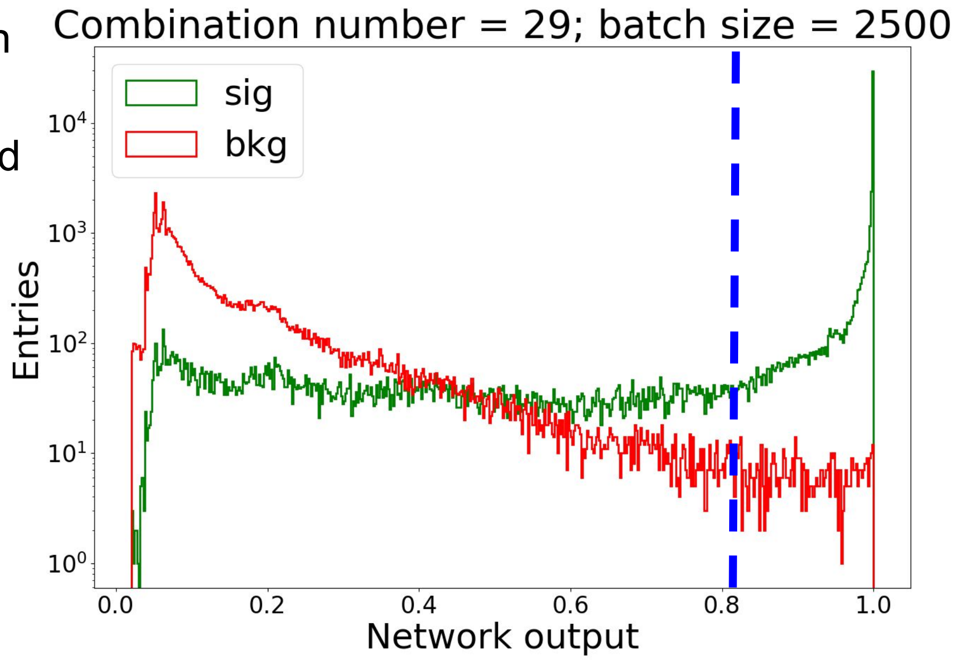
ROC curve calculation strategy

Network output divided by true label indexes.

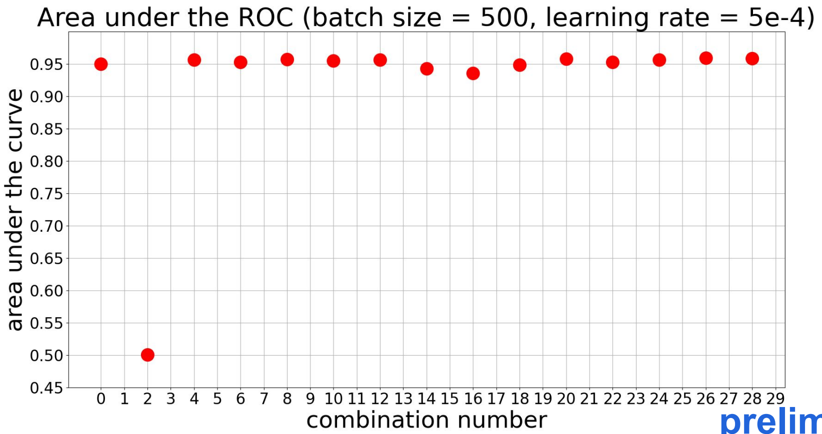
Signal efficiency: normalized signal integration above threshold.
Background efficiency: normalized background integration above threshold.
Background rejection: 1 - bkg efficiency.

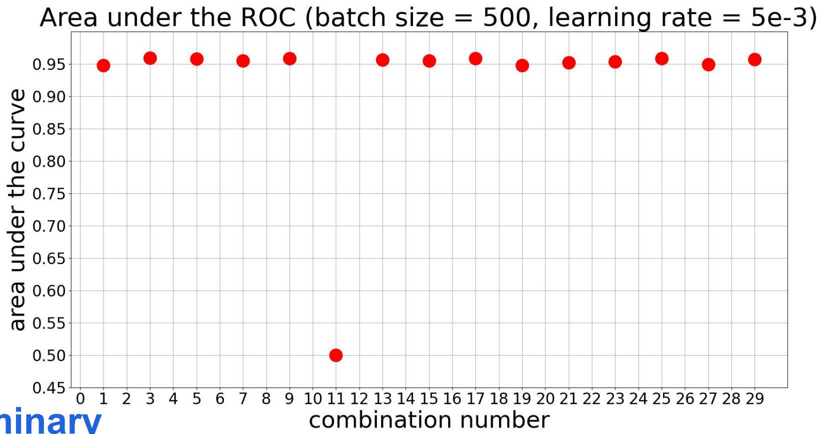
Network architecture

batch size = 2500
learning rate = 5e-3
Layers number = 8
Nodes per layer = 128, 128, 128, 128, 128, 128, 128



Area under the ROC curve





preliminary