PyHEP 2020

3rd Workshop on Python in High Energy Physics

KU THE UNIVERSITY OF KANSAS

Machine learning technique for signal-background separation of nuclear interaction vertices in the CMS detector

Phil Baringer and Anna Kropivnitskaya
University of Kansas

Code with Toy data is available at Binder

Introduction

In HEP, very often problem of signal-background (noise) separation appears:

- resonant peaks from decay particle,
- physics objects reconstruction/identification, for example, pions, photons, electrons, b(t)-jets…
- nuclear interactions with material (this analysis),
- photon conversions with material,

which are on top of combinatorial background.

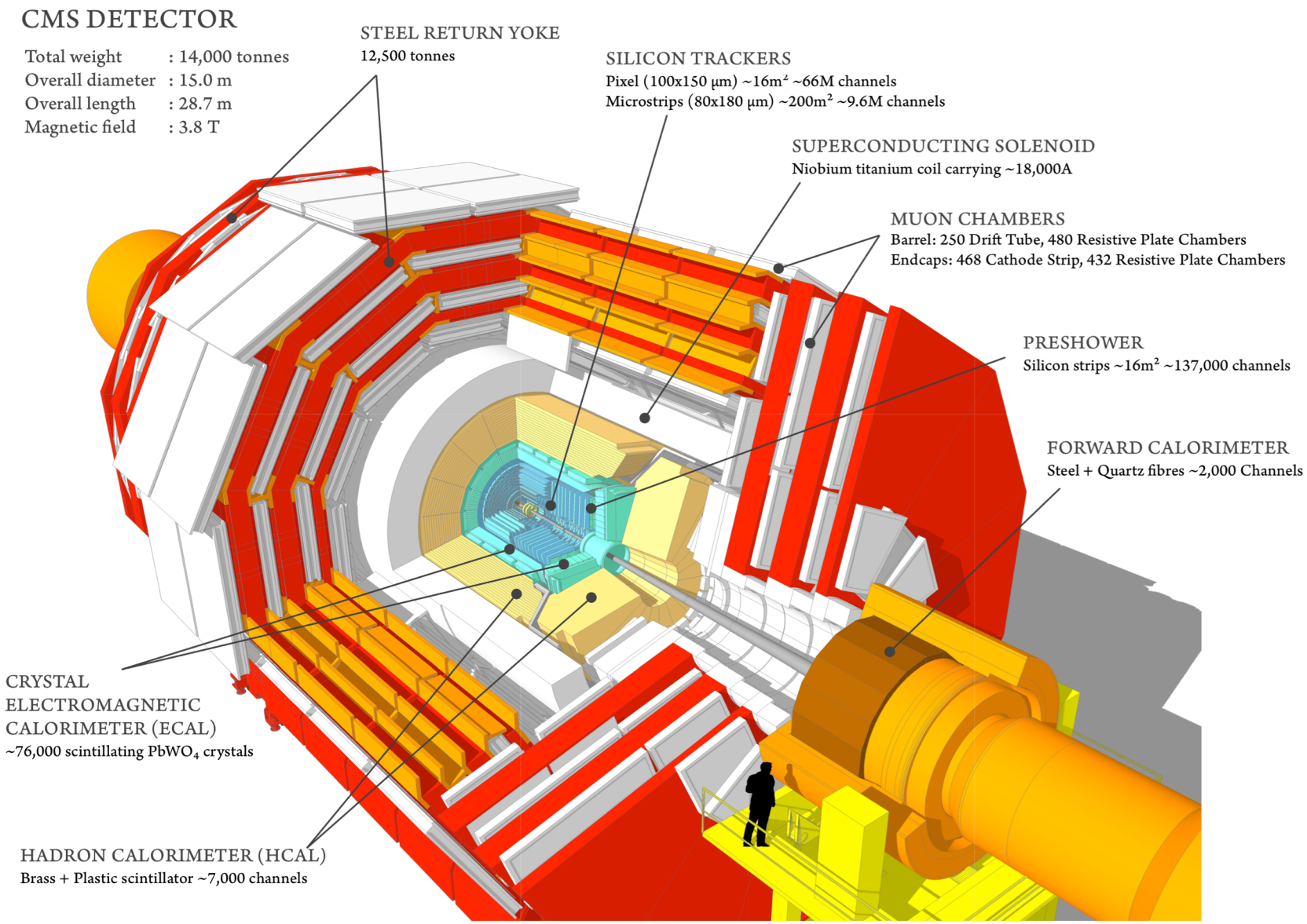
Very often, there is no clean sample of signal to train a neural network (NN), but regions with enhanced contribution of signal could be known (by mass, geometrical, or any phase space cuts).

→ Machine learning technique of NN with floating classification could be helpful:

- input classification is done on mixed samples with different fractions of background to signal,
- output classification is optimized for real signal and combinatorial background separation.

This analysis is performed with Jupyter notebook at SWAN platform:
https://github.com/kropiv/MLforNIatPyHEP and Binder
and is based on the CMS public results, public twiki
There is no public CMS data sample for this notebook
→ Toy sample was generated with NumPy to show the notebook functionality.

CMS Detectors



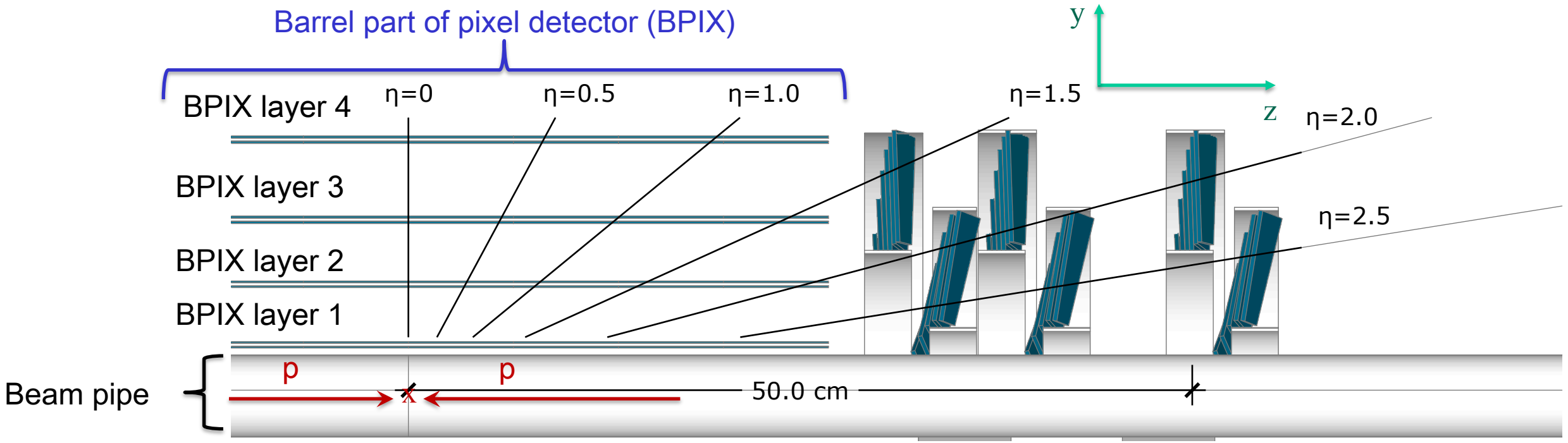
CMS Detectors are designed to study different particle properties, created during p-p collisions, with high performance and resolutions:

- muon system detects and measures muons,
- central tracking system gives accurate momentum measurements, as well as the position of primary and secondary vertices
- an electromagnetic calorimeter detects and measures electrons and photons,
- hadron calorimeter detects and measures hadrons.

In this analysis, the material of the pixel detector (part of the tracking system) is studied with nuclear interactions.

Top-right Part of CMS Phase-1 Pixel Detector: y-z View

Inner tracker system consists of Pixel and Strip detectors,
which reconstruct charged particles in the CMS detector.



Pixel detector consists from Barrel Pixel (BPIX) and Forward Pixel detectors.
Data from barrel region only are analyzed in this analysis.

Data Selection and Reconstruction

- Nuclear interactions (NIs) are interactions of hadrons with the detector material, which create secondary vertices.
- For Physics analysis, the NIs play the role of background (noise) in the data.
- Due to its interaction with material of the detector, the positions of these secondary vertices map the material with a sub-millimeter precision. Accurate knowledge of the positions of active and inactive elements is important for simulating the detector, planning detector upgrades, and reconstructing charged particle tracks.

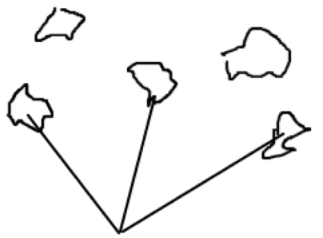
- p-p collisions at 13 TeV from the single muon collection:
  - study with part of 2018 promptly reconstructed data (4.3 fb^{-1}).

- Nuclear interaction (NI) reconstruction technique is presented at:
  - JINST 2018: http://dx.doi.org/10.1088/1748-0221/13/10/P10034
  - CERN-CMS-DP-2019-001: http://cds.cern.ch/record/2664786?ln=en

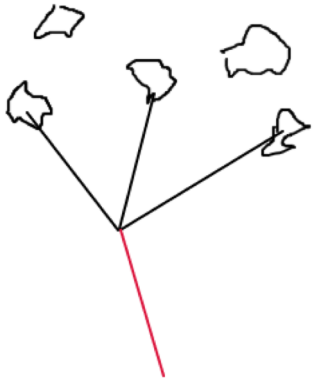
- NI is a displaced vertex with following requirements:
  - at least 3 tracks incoming or outgoing from the vertex;
  - invariant mass of the outgoing system > 1 GeV
    to suppress light mesons, baryons, and photon conversions.

- Only NI vertex candidates (NI candidates) from barrel region of the tracker detector ($|z| < 25$ cm) are analyzed:
  - around 1.5×10^6 NI candidates.

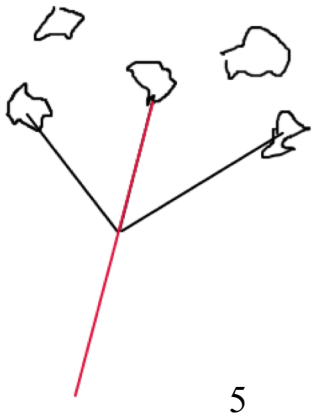
NI without incoming track



NI with incoming track

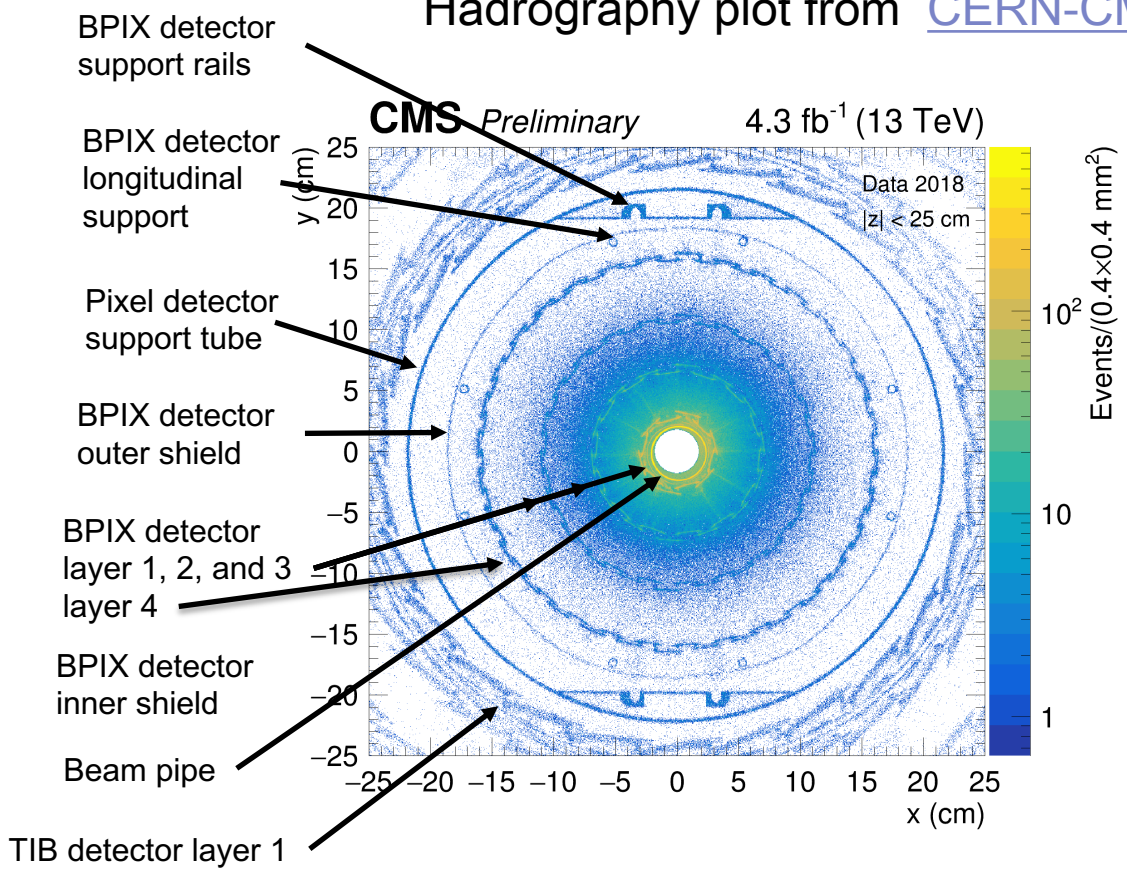


NI with merged track



Inner Tracker Hadrography x-y Plane

Hadrography plot from CERN-CMS-DP-2019-001



NI candidates reproduce inner tracker hadrography, but a big combinatorial background from random vertices is observed near the beam pipe, 1st, 2nd and 3rd layers of the Barrel Pixel (BPIX) detector.

NN Motivation and Strategy

For each NI candidate, monitoring variables are available, which are very similar for real NIs (signal) and combinatorial background (background).

→ Machine learning technique could be helpful for signal-background separation.
→ A NN with 2 hidden layers (with 12 neurons in each hidden layer) is selected.

- For simplicity, only vertices for a NI candidate with exactly 3 tracks are used:
  1.4×10^6 (=3 tracks, majority of the vertices) from 1.5×10^6 ($\geq$ 3 tracks) vertices.

- 23 variables are injected to the input layer of NN (23 input variables for NN):
  - number of primary vertices;
  - number of NI candidates per one event;

  3 tracks for each NI vertex candidate:
  - p_T, η, and ϕ;
  - χ^2, and normalized χ^2;
  - number of valid hits;
  - track reconstruction algorithm.

- The vertex position (x, y, and z) is used for classification of NI candidates.
  N.B.: It is not injected as input variable for NN.

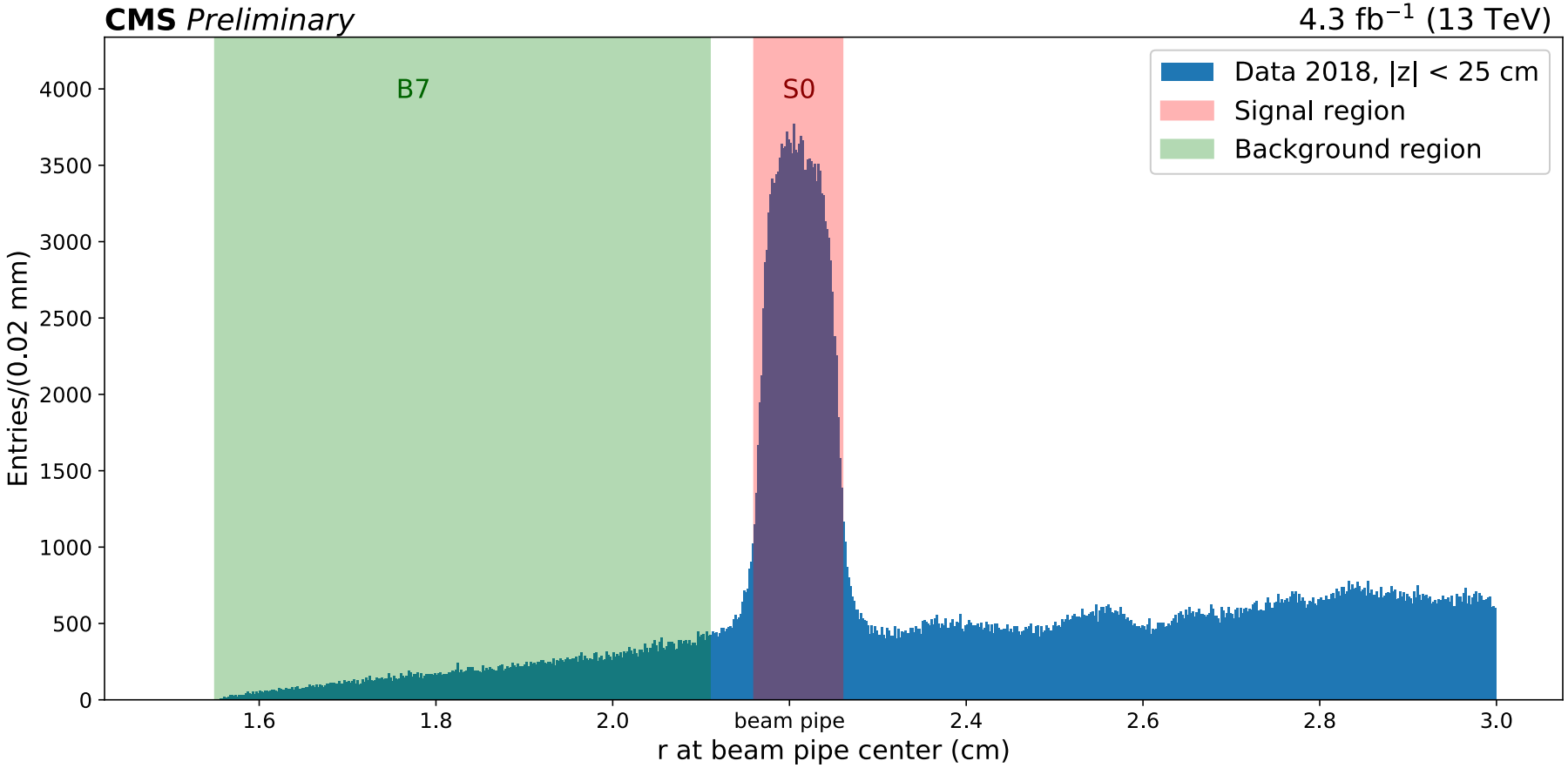
Classification Strategy for NN

- Classification for signal and background regions is done by using the radius of the NI candidate, r, where $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$:
  - (x, y) is the vertex position of the NI candidate,
  - $(x_0,\ y_0)$ is the position, in CMS coordinate, of the material structures, which are installed with millimeter precision, and is taken from CERN-CMS-DP-2019-001.

- Classification is done for 7 signal regions (S0-S6) and for 5 background regions (B7-B11).

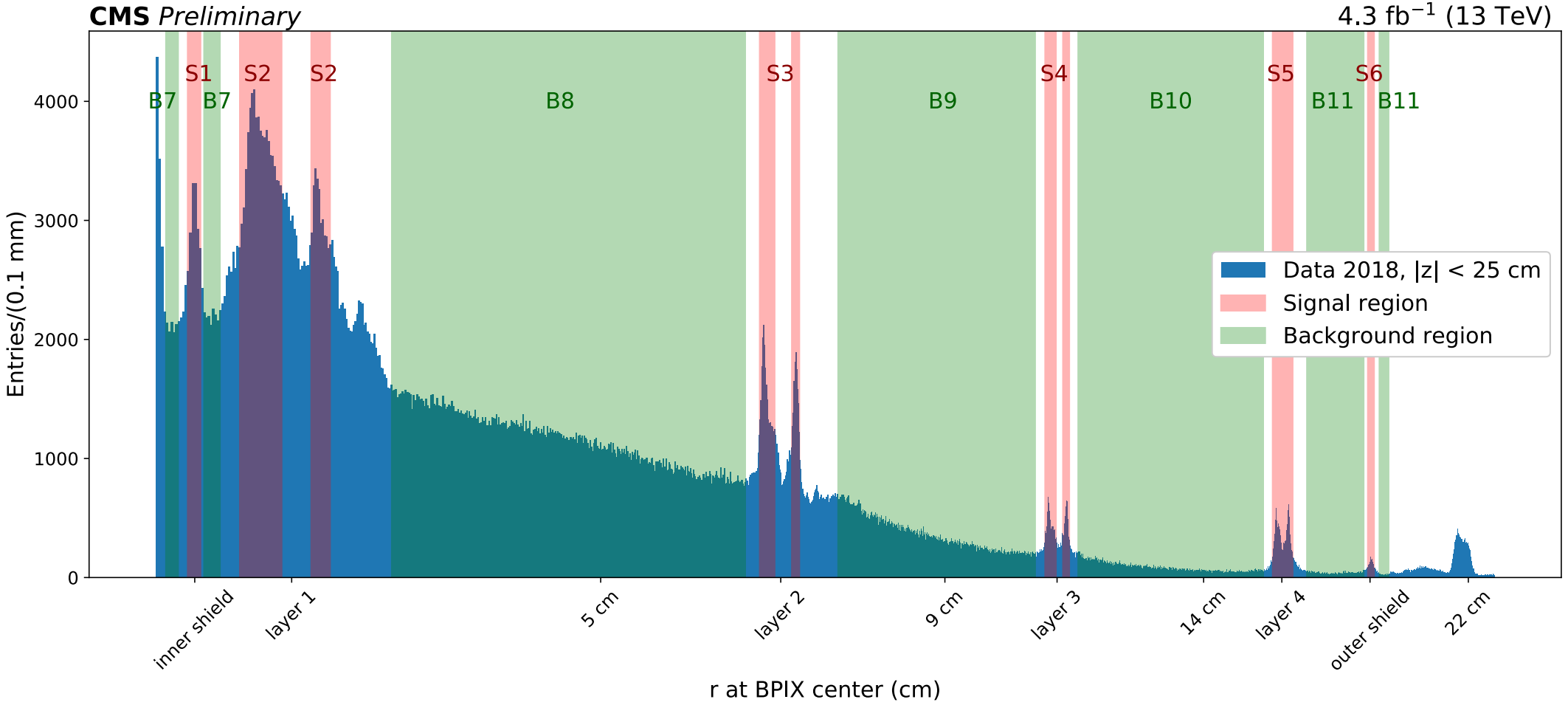
Material structure	**Radius definition**	Material center position $(x_0,\ y_0)$
beam pipe (BP)	r, centered at BP	(1.71, -1.76) mm
BPIX inner/outer shields, layers 1-4	r, centered at BPIX	(0.86, -1.02) mm
BPIX rails and pixel support tube	r, centered at tube	(0.80, 3.18) mm

Classification around Beam Pipe



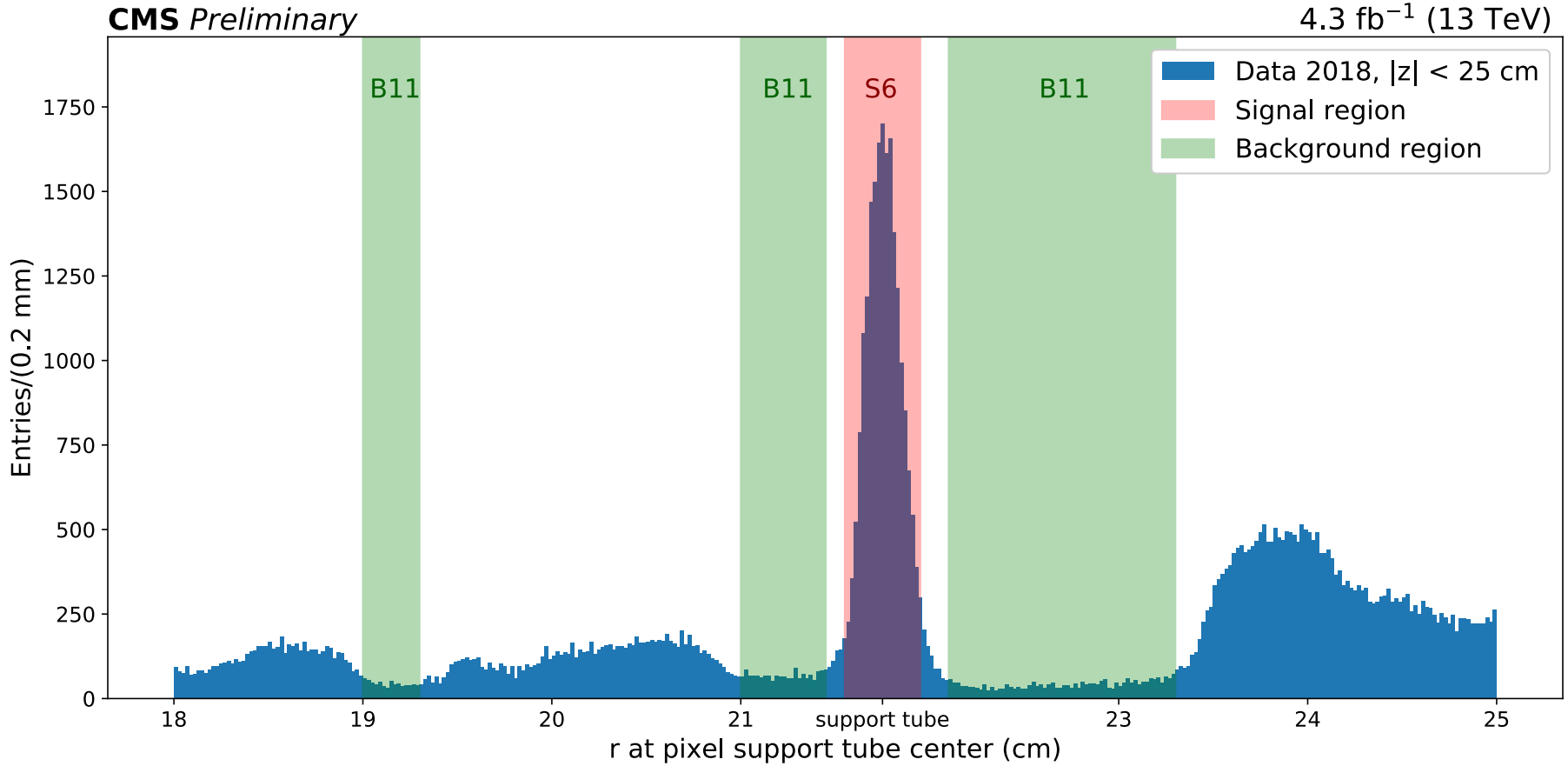
- Classification for signal, S0 (red), and background, B7 (green) regions is done.
- Background downstream of the the beam pipe is not classified, because there is a contribution from smeared BPIX detector inner shield and layer 1.
- Background-signal ratio (B/S) for the beam pipe region is estimated from sidebands to be around 0.16.

Classification around BPIX Detector



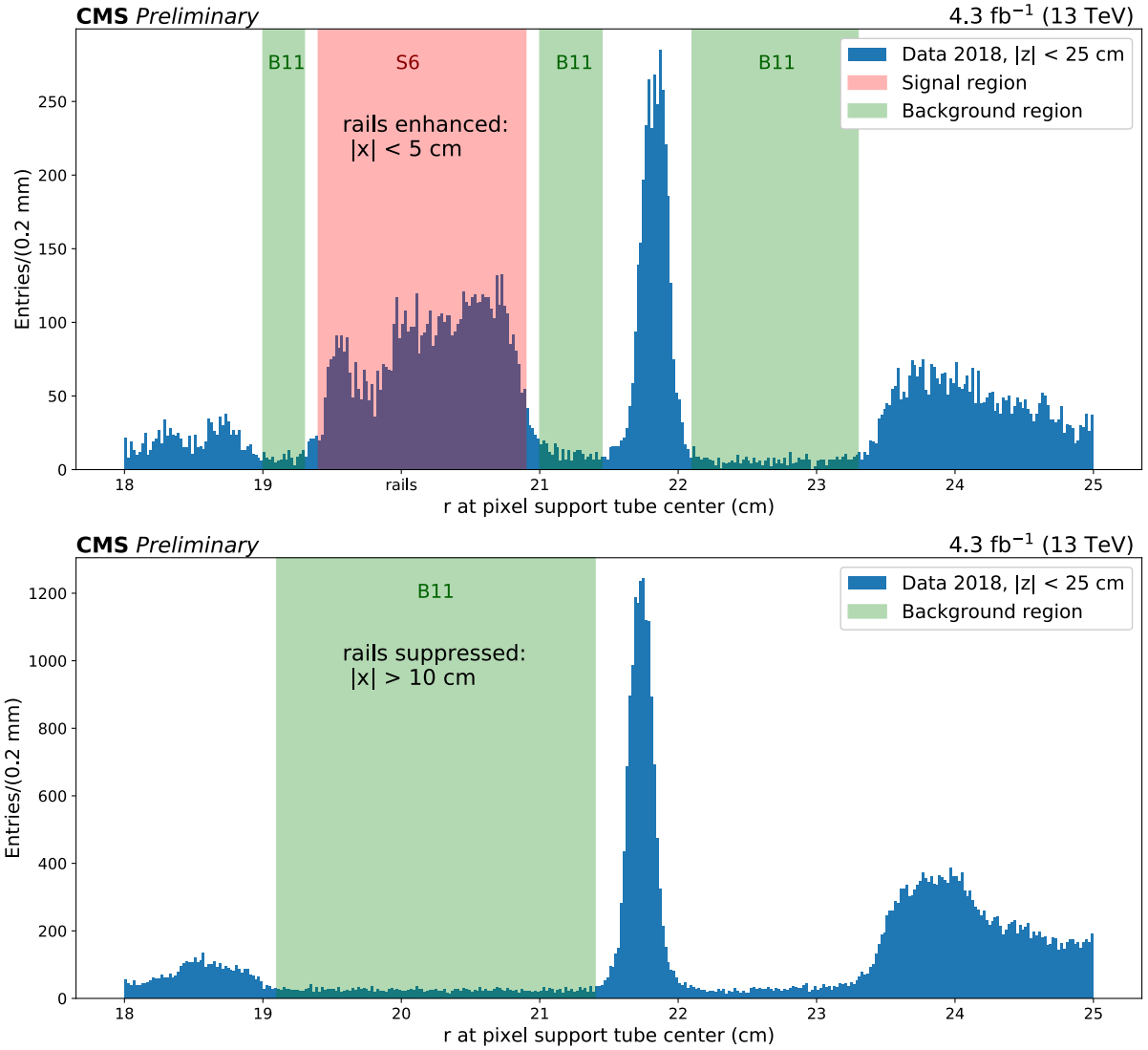
- Classification for signal, S1-S6 (red), and background, B7-B11 (green) regions is done.
- Background-signal ratio (B/S) for all material elements in the BPIX detector is estimated from sidebands to be around 2.6 for inner shield (S1), 1.2 for layer 1 (S2), 0.9 for layer 2 (S3), 0.7 for layer 3 (S4), 0.17 for layer 4 (S5), and 0.4 for outer shield (S6).

Classification around Pixel Detector Support Tube



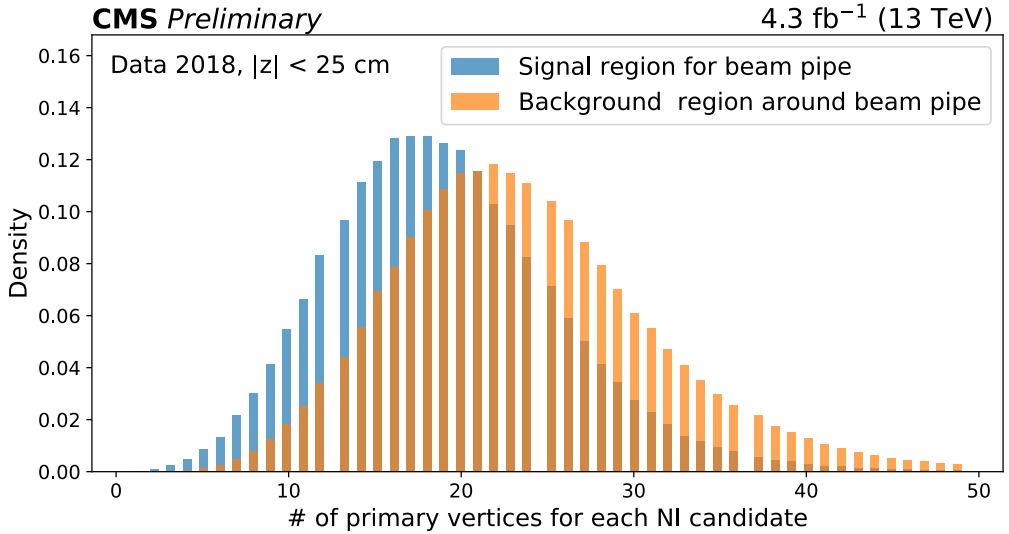
- Classification for signal, S6 (red), and background, B11 (green) regions is done.
- Background-signal ratio (B/S) for the support tube is estimated from sidebands to be around 0.06 (S6).

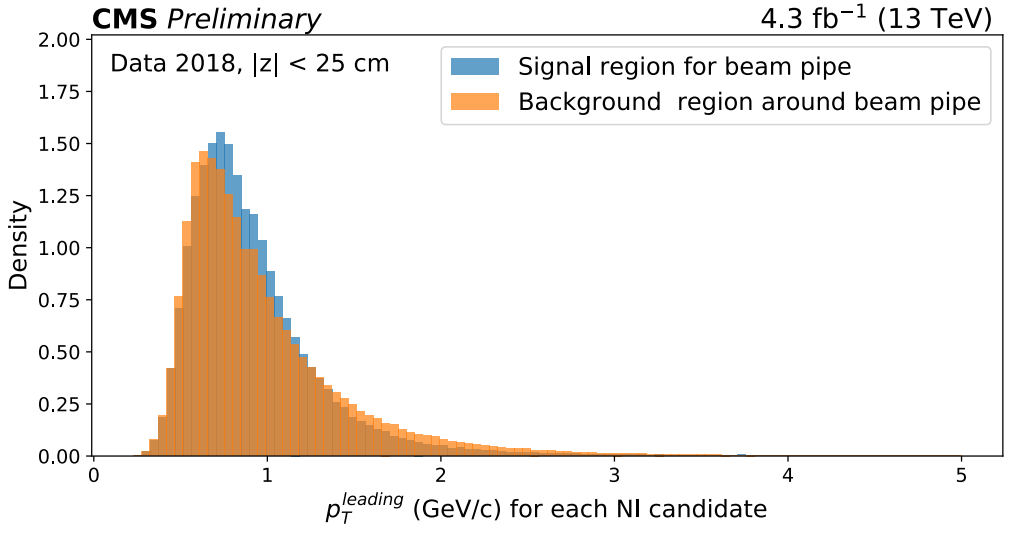
Classification around BPIX Detector Rails

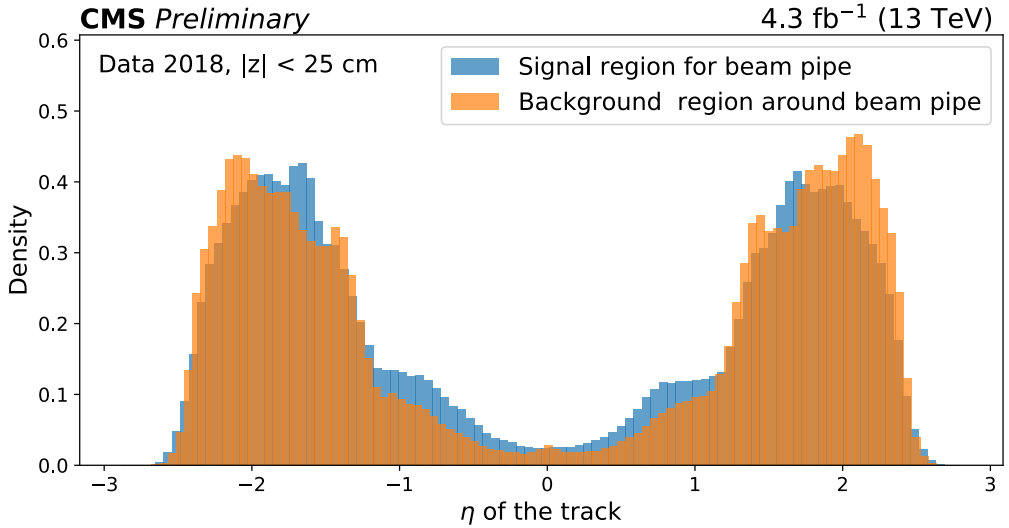


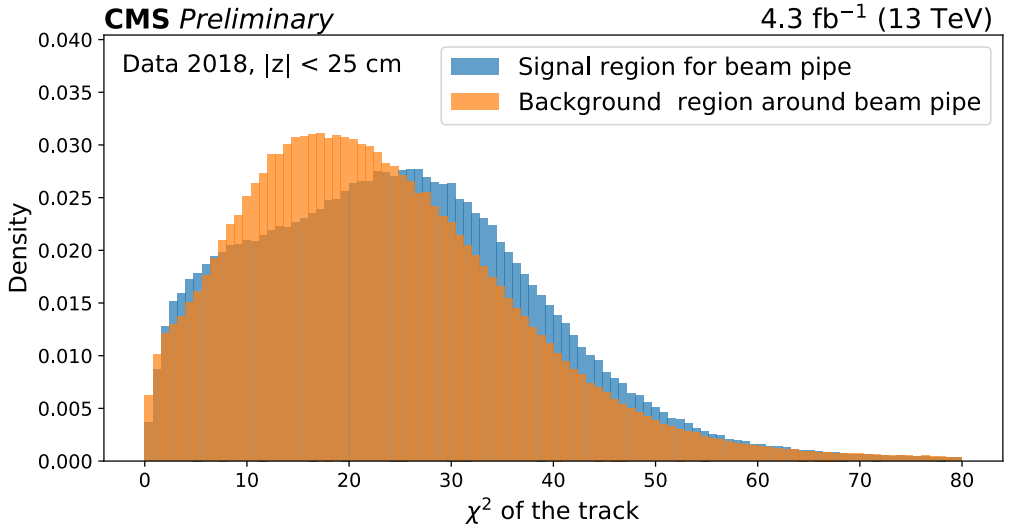
- Top plot corresponds to the enhanced region for the BPIX detector rails ($|x| < 5$ cm, where x is the x coordinate of the vertex), while bottom plot corresponds to the suppressed region for the rails ($|x| > 10$ cm).
- Classification for signal, S6 (red), and background, B11 (green) regions is done. Background-signal ratio (B/S) for the BPIX detector rails is estimated from sidebands to be around 0.13 (S6).

Input Variables to the NN near Beam Pipe









Small shape differences between signal and background distributions are observed.
→ A NN technique could be helpful for signal-background separation.

Data Preparation and Classification for the NN

- Data (NI candidates) are presented in NumPy matrix X:
  - each column represents: 23 input variables for NN + 3 variables for vertex position (classification);
  - each row represents a new NI candidate:
    - ➢ if we have *n* NI candidates per event → we will have *n* rows.

- Classification for NN, Y_{true} set includes 2 classes, calculated by using signal and background region definitions on slides 9-12:
  - Class 0 (signal class): Beam pipe, layers 2-4, rails, and tube signal regions (S0, S3-S6).
    - N.B.: inner shield and layer 1 signal regions (S1 and S2), due to high background contamination, are not classified as signal class for NN (not used during training):
      - ➢ input variable distributions in the beam pipe region should be similar to the inner shield and layer 1;
      - ➢ great cross-check for NN signal-background separation power in non-trained signal regions.
  - Class 1 (background class): all classified background (B7-B11).
  - Some NI candidates, including inner shield and layer 1 signal regions, stay in non-classified state, they are not used during training.

NN Model with Floating Classification

- Classification of the input to NN, Y_{true}, is done on the mixed samples:
  - Signal class (class 0) includes real NIs (S) and combinatorial background (B). For some structures background-signal ratio (B/S) reaches almost 1.
  - Background class (class 1) includes mostly combinatorial background (>99%).

- NN model is trained on the mixed sample and calculates the probability distribution, p_{Signal}, for the injected vertex to be a signal.

- The predicted classification, $Y_{predict}$, is estimated by optimizing cut on the p_{Signal} for real NIs and combinatorial background separation (see slides 18 and 19).

This is an example of an NN Model with floating classification:

- Input classification is done on mixed samples with different fraction of B/S.

- Output classification is optimized for real NIs and combinatorial background separation.

- This technique is similar to CWoLa method, described in https://arxiv.org/abs/1708.02949.

Data Splitting for Train and Test Sets

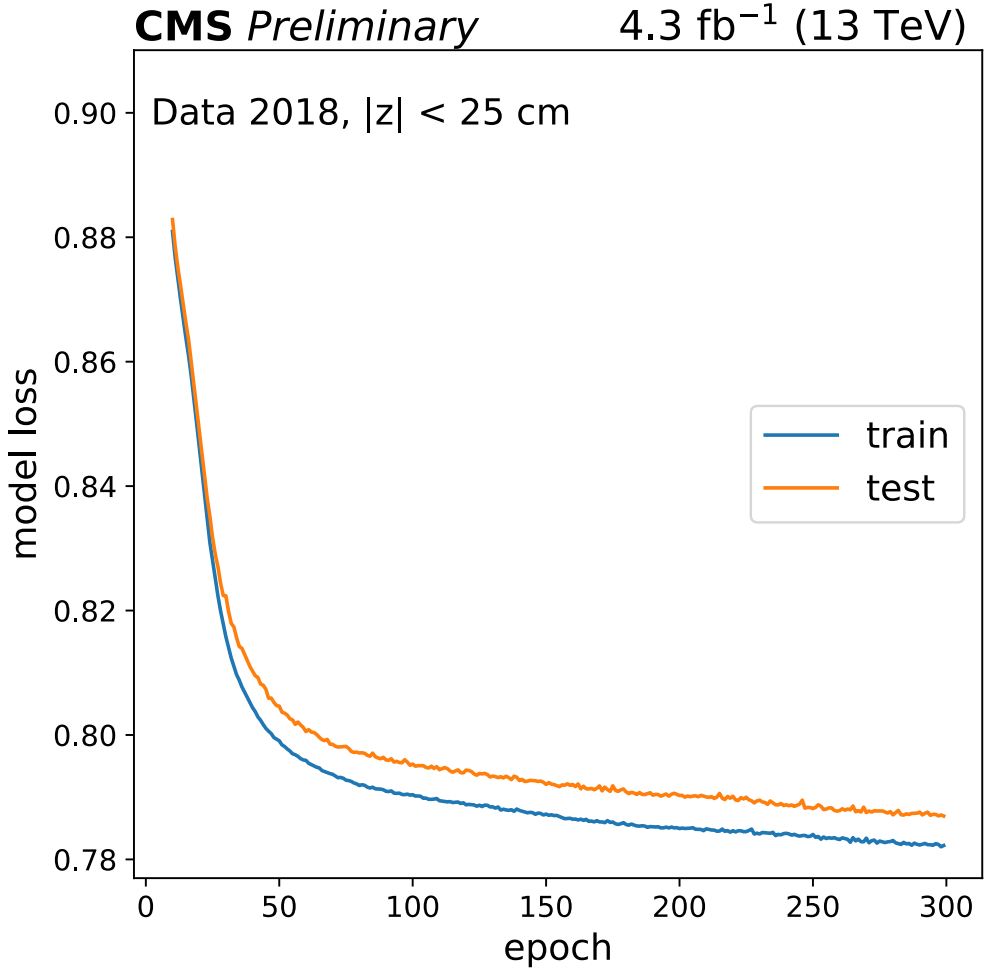
- Shuffle data X and Y_{true} (classification) to avoid run (time)-dependent correlations.

- Divide X and Y_{true} into Train and Test data sets ~50:50% (not much data):
  - Test set includes ~0.7×10^6 NI candidates;
  - Train set includes ~0.7×10^6 NI candidates.

- Normalize each input variable in the X Train set on its mean and standard deviation for better performance.
- Perform Principal Component Analysis (PCA) on input variables in the X Train set:
  - PCA is defined as an orthogonal linear transformation that transforms the Data to a new coordinate system: 1st PCA variable has greatest variance, 2nd PCA variable has 2nd greatest variance, and so on…
  - PCA shows that all 23 variables after PCA transformation have significant variance, that is why all PCA variables were kept for the injection to the NN.
- Test set uses the same normalization and PCA as the Train set.

- For model training and validation, only classified events (class 0 and 1) were used: ~0.4×10^6 from ~0.7×10^6 NI candidates.
- For the model prediction, all events (classified and non-classified) are used.

History of the NN Model during Training (Classified Candidates only)



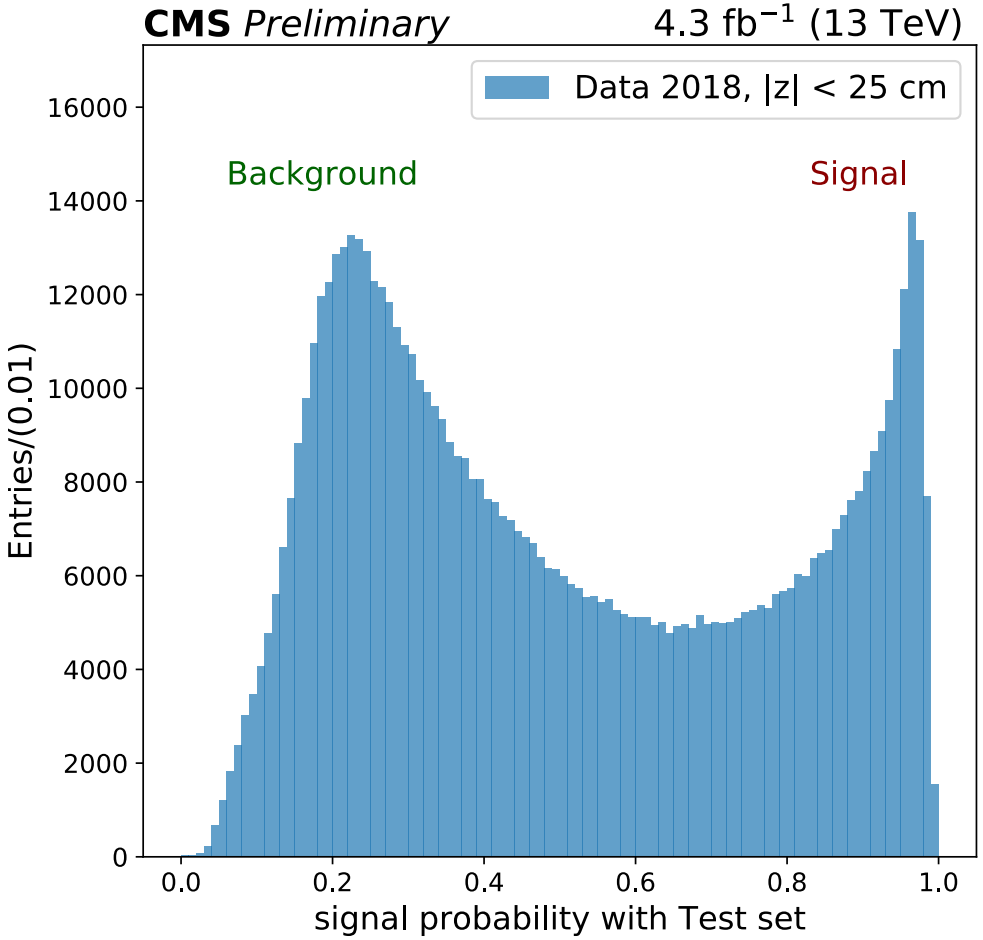
model loss – categorical cross entropy is used for the optimization function for the NN

$$-\frac{1}{N}\sum_{i=1}^{N} class_weight \times \log\left(Y_{predict}\right)\left[Y_{true}^{i} \in Class^{i}\right].$$

epochs – number of passes of the Train Set to the NN.

Test and Train loss distributions are very close and are decreasing during training, which shows that model is not over/under trained.

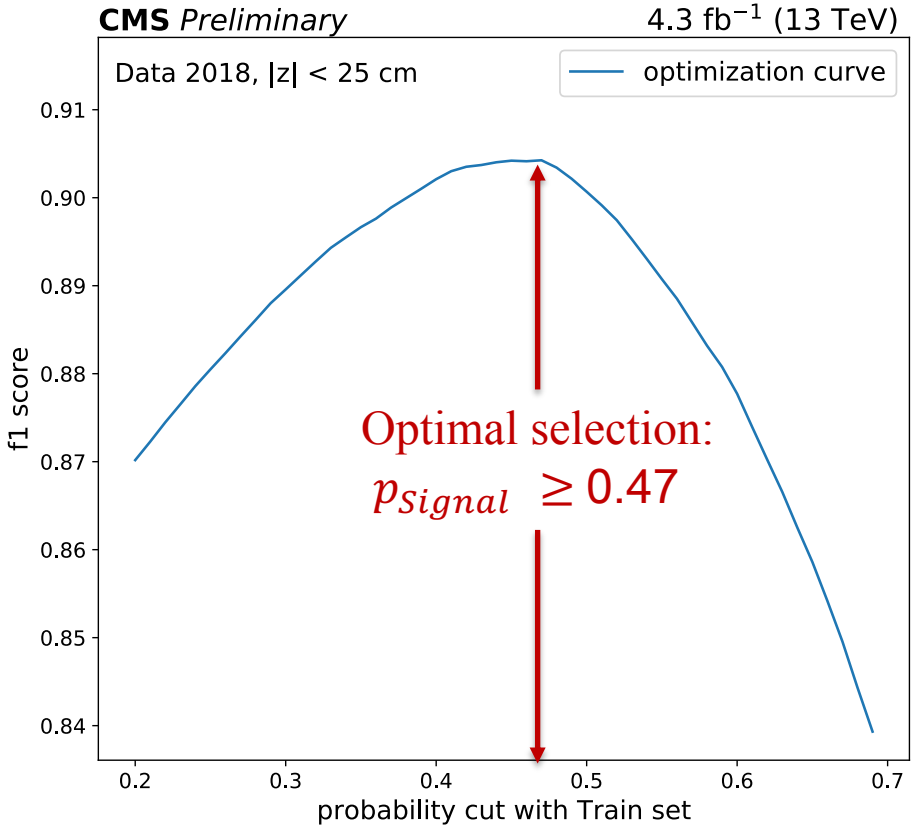
Classification Probability

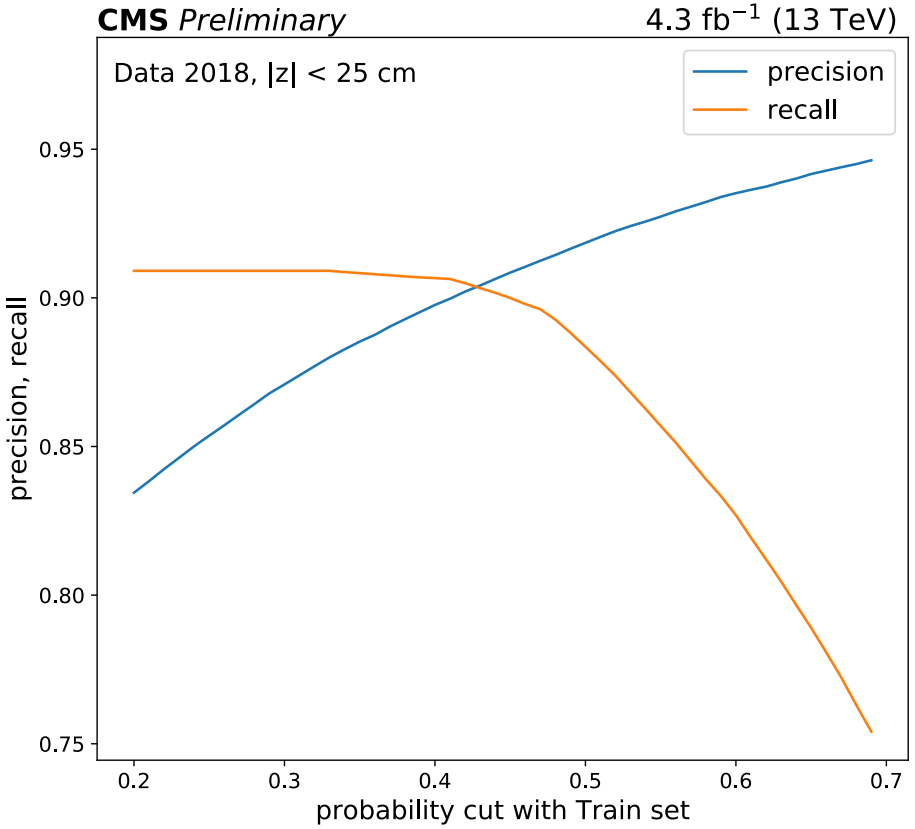


Probability distribution after NN model training of the NI candidate to be a signal, p_{Signal}. It is used to predict classes of the NN output, $Y_{predict}$.

- Output of the NN is a probability for the NI candidate to be classified as a signal : p_{Signal}.
- Class prediction, $Y_{predict}$, is defined by p_{Cut} :
  - class 0 (signal): $p_{Signal} \geq p_{Cut}$;
  - class 1 (background): $p_{Signal} < p_{Cut}$.
- For signal-background classification p_{Cut} is optimized to maximize $f1\ score = 2\times\frac{precision \times recall}{precision+recall}$, calculated on real NIs and combinatorial background, which requires high $precision(purity) = \frac{\sum Y_{true}\times Y_{predict}}{\sum Y_{predict}}$ and high $recall(efficiency) = \frac{\sum Y_{true}\times Y_{predict}}{\sum Y_{true}}$ at the same time.
  - N.B.: $f1\ score$ is calculated on real NIs and combinatorial background, estimated from sidebands, but not on the Signal (mixed sample, which includes combinatorial background) and Background classes prediction ($Y_{predict}$).

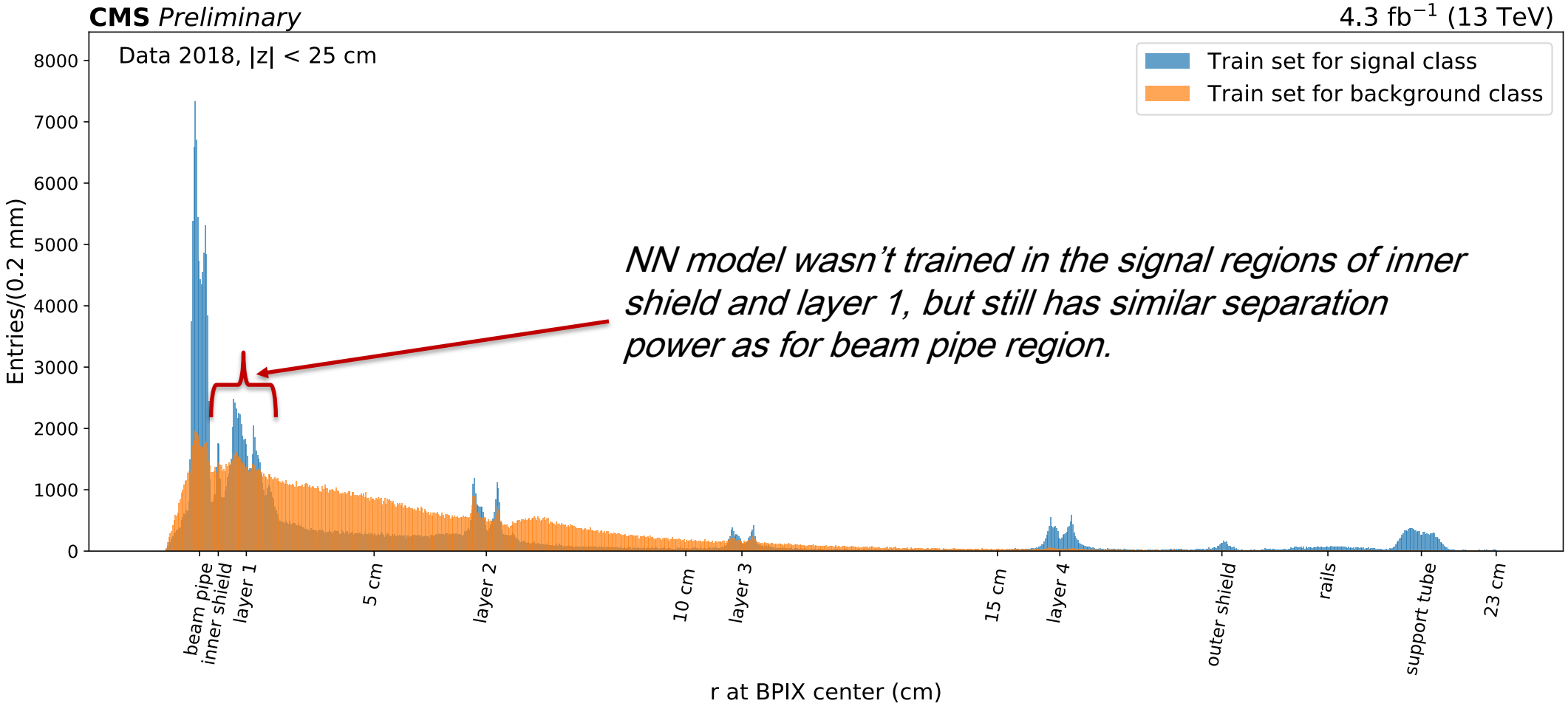
NN Model Optimization with Test Set





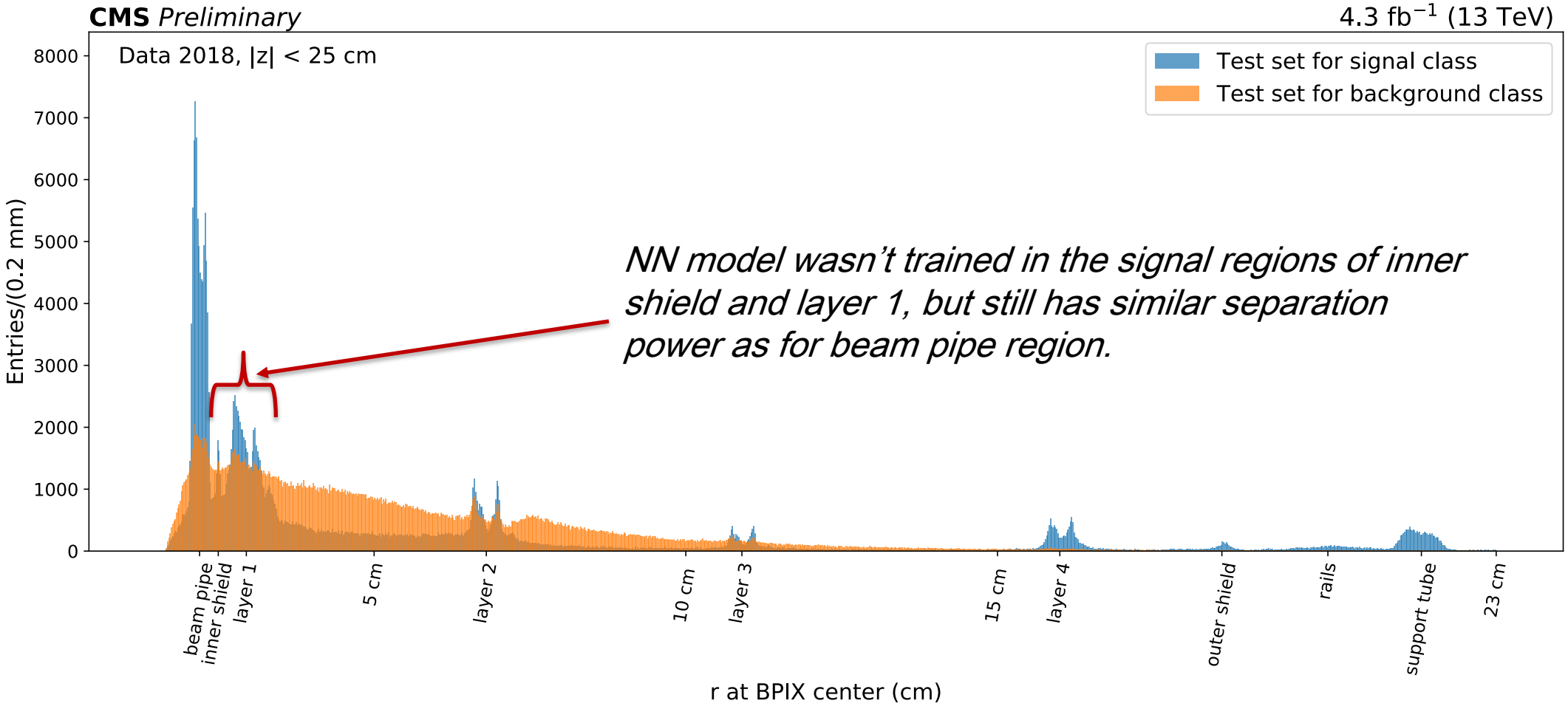
- Optimal value for p_{Signal} is 0.47 to separate signal from background. $f1\ score$ is calculated with the assumption that false-negative prediction (inefficiency) could be reliably estimated only until 10% from true-positive signal.
- $Y_{predict}$ is calculated based on this optimal cut. $Y_{predict}$ optimization should be done on the Validation set, but due to lack of data, Train set is used.

Model Results on the Train Set (Cross Check)



NN model suppresses background by a factor of 4-5 for Train set, and keeps around 90% of the signal (real NIs).

Model Results on the Test Set (Main Results)



NN model suppresses background by a factor of 4-5 for Test set, and keeps around 90% of the signal (real NIs).

Background to Signal (B/S) Ratios in Signal Regions (S0-S6)

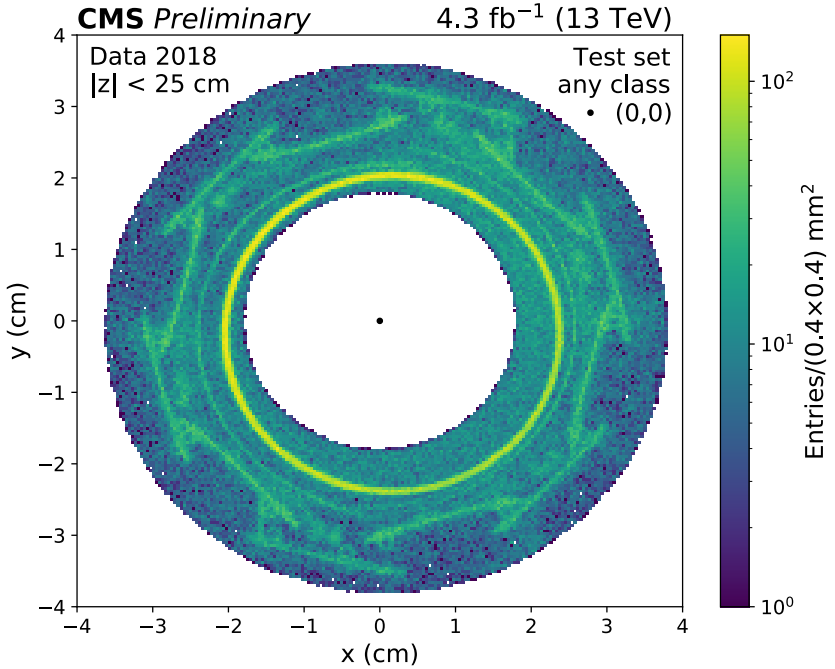
B/S ratio	Beam pipe (S0)	Inner shield (S1)	Layer 1 (S2)	Layer 2 (S3)	Layer 3 (S4)	Layer 4 (S5)	Outer shield (S6)	Tube (S6)	Rails (S6)
Data set without classification	0.16	2.6	1.2	0.9	0.7	0.17	0.4	0.06	0.13
Train set Signal class	0.07	1.1	0.5	0.3	0.3	0.10	0.3	0.04	0.09
Test set Signal class	0.07	1.1	0.5	0.3	0.3	0.10	0.3	0.05	0.09

B/S ratios in Signal Regions (S0-S6) is estimated from sidebands and is presented for:

- whole Data set without classification,
- Train set for Signal class of NN prediction,
- Test set for Signal class of NN prediction.

Significant background suppression is observed in Train/Test sets after NN signal classification.

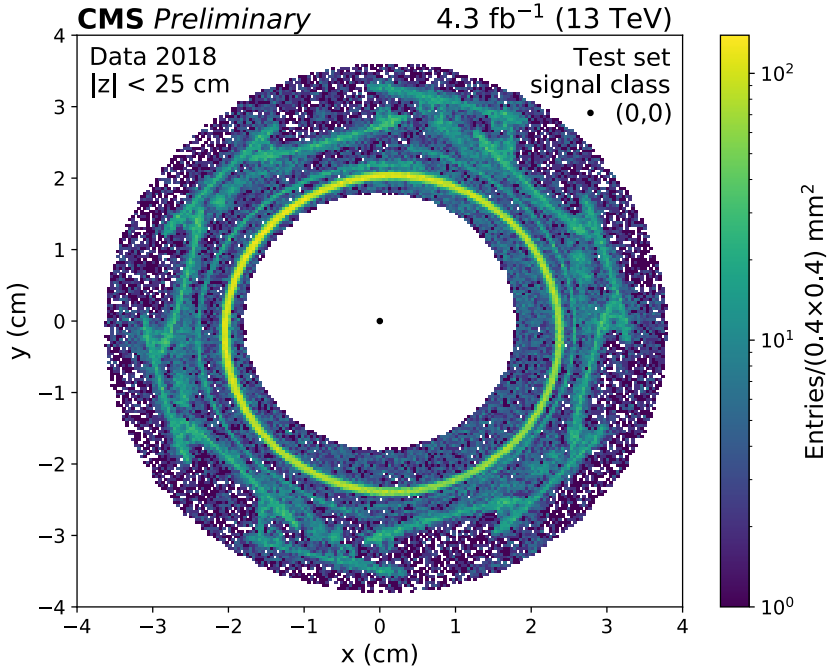
Inner Tracker Hadrography in x-y Plane

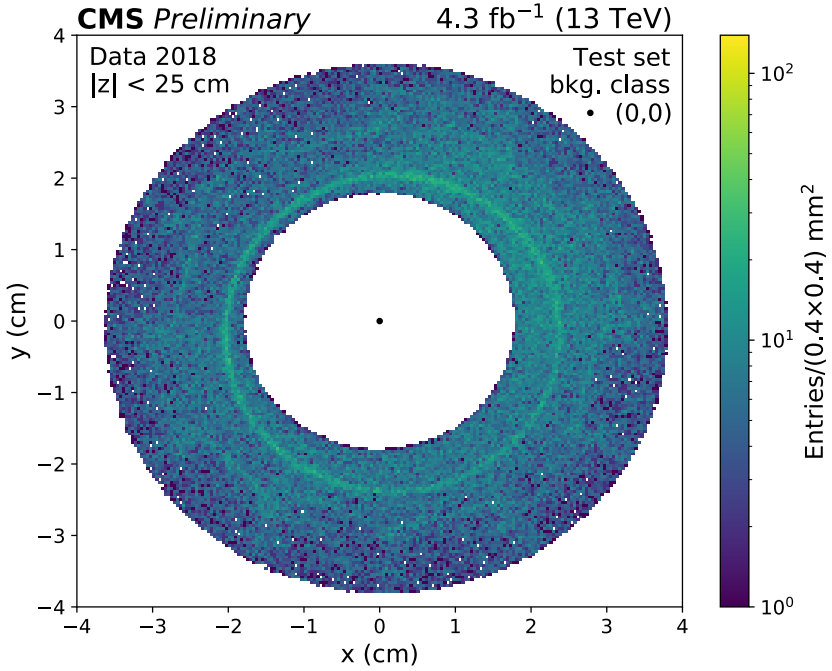


The signatures of

- the beam pipe with radius around 2.2 cm,
- BPIX detector inner shield with radius around 2.5 cm,
- layer 1 with radius around 3 cm

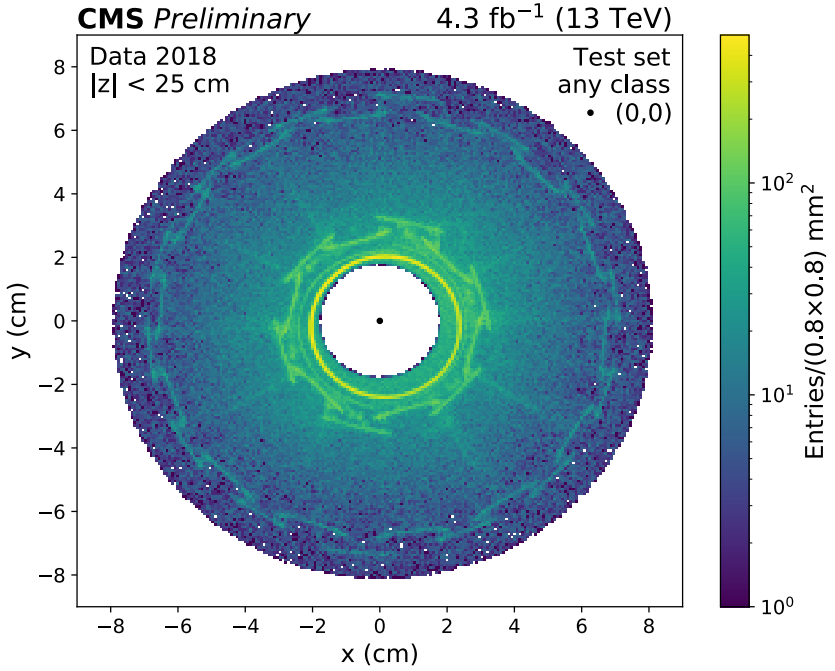
can be observed above the combinatorial background.





NN model has impressive signal-background separation power.

Inner Tracker Hadrography in x-y Plane

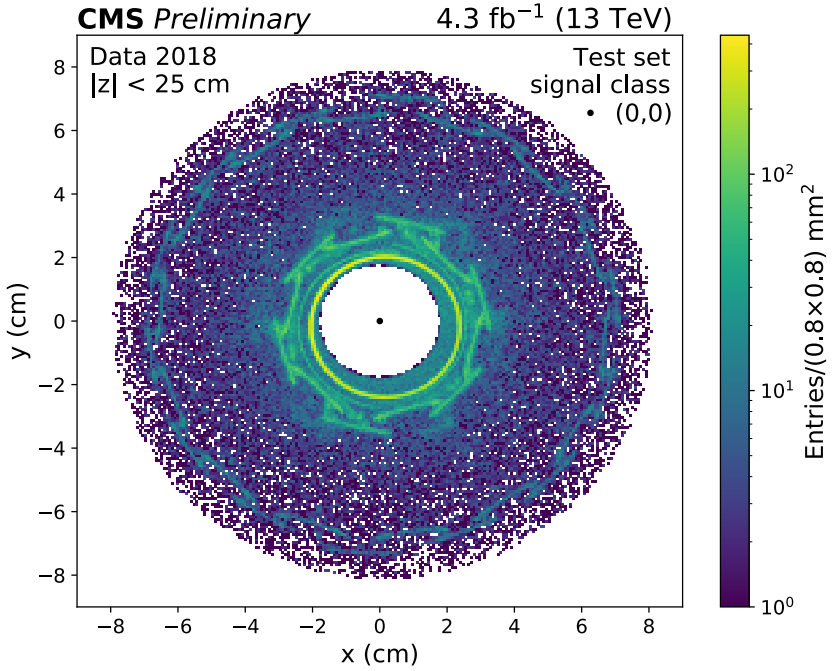


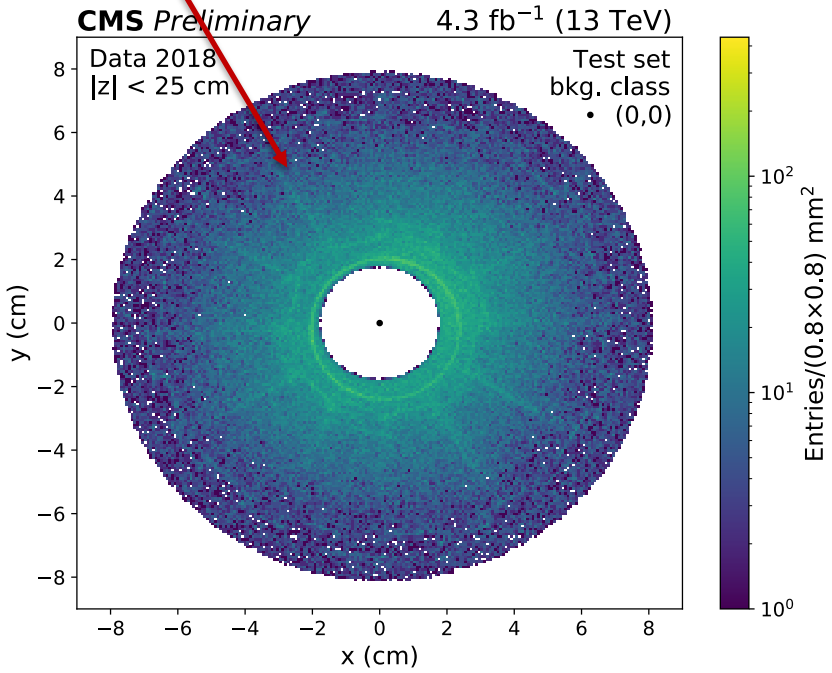
The signatures of

- the beam pipe with radius around 2.2 cm,
- BPIX detector inner shield with radius around 2.5 cm,
- layer 1 with radius around 3 cm,
- layer 2 with radius around 6.8 cm

can be observed above the combinatorial background.

Radial lines between layers 1 and 2 are classified as background as expected.





NN model has impressive signal-background separation power.

Conclusion

- Machine learning method for reducing combinatorial background in data without Monte Carlo is presented:
  - NN model with floating classification on the mixed samples is constructed.
  - Even though the variables used for the training show only small differences between background and signal, the NN has impressive separation power.
  - NN model wasn't trained in the signal regions of inner shield and layer 1, but still has similar separation power as for beam pipe region.
  - Combinatorial background suppression could decrease systematic uncertainties for material position and material density measurements.
  - This technique could be used for any analysis, where you could perform classification on mixed samples with different ratios of background to signal.

Thanks to NSF support

BACK UP

Abstract

The CMS inner tracking system is a fully silicon-based high precision detector. Accurate knowledge of the positions of active and inactive elements is important for simulating the detector, planning detector upgrades, and reconstructing charged particle tracks. Nuclear interactions of hadrons with the detector material create secondary vertices whose positions map the material with a sub-millimeter precision in situ, while the detector is collecting data from LHC collisions.

A neural network (NN) with two hidden layers was used to separate secondary vertices due to combinatorial background from those arising from nuclear interactions with material. The NN was trained and tested on data from proton-proton collisions at a center-of-mass energy of 13 TeV, recorded in 2018 at the LHC.

NN training is performed using Keras and Matplotlib in a Jupyter notebook. Secondary vertices in the training data are classified as signal or background, based on their geometrical position. Even though the variables used in training show only small differences between background and signal, the NN has impressive separation power. Hadrographies of the CMS inner tracker detector before and after background cleaning are presented.

NN Model Structure

NN Structure with 2 Hidden Layers:

Layer (type)	Output Shape	Param #
Input Layer (0)	(None, 23)	0
Dense Layer (1)	(None, 12)	276
Batch Normalization (1)	(None, 12)	48
Activation: ReLu (1)	(None, 12)	0
Dense Layer (2)	(None, 12)	144
Batch Normalization (2)	(None, 12)	48
Activation: ReLu (2)	(None, 12)	0
Dense (2)	(None, 2)	26
Activation: Softmax (2)	(None, 2)	0

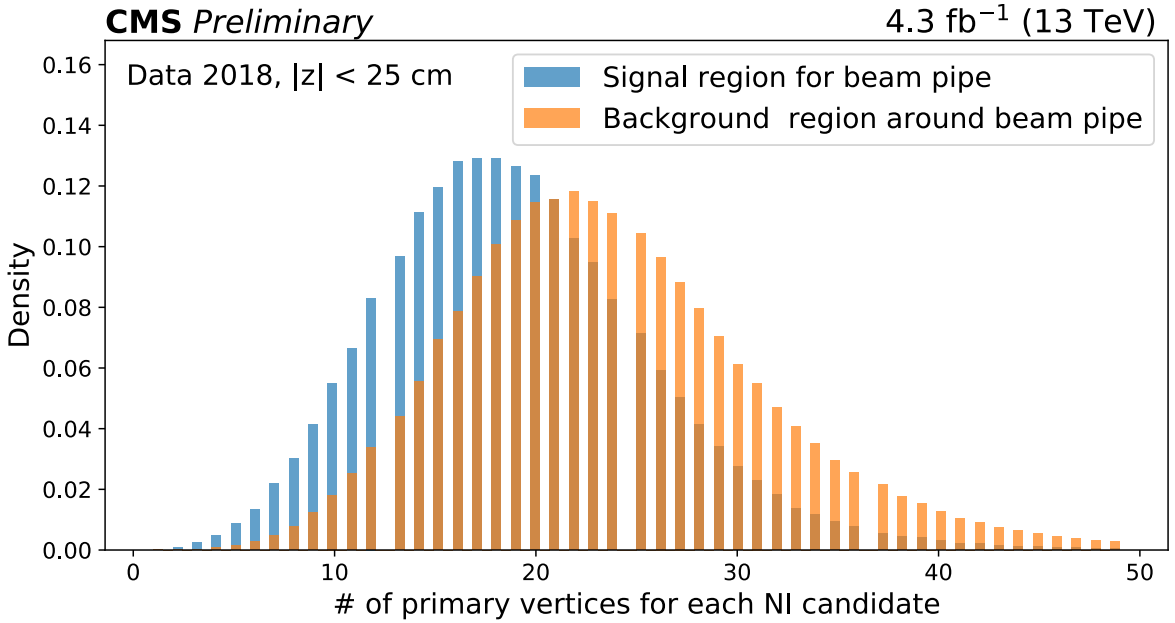
Total parameters: 542
Trainable parameters: 494 (!)
Non-trainable parameters: 48

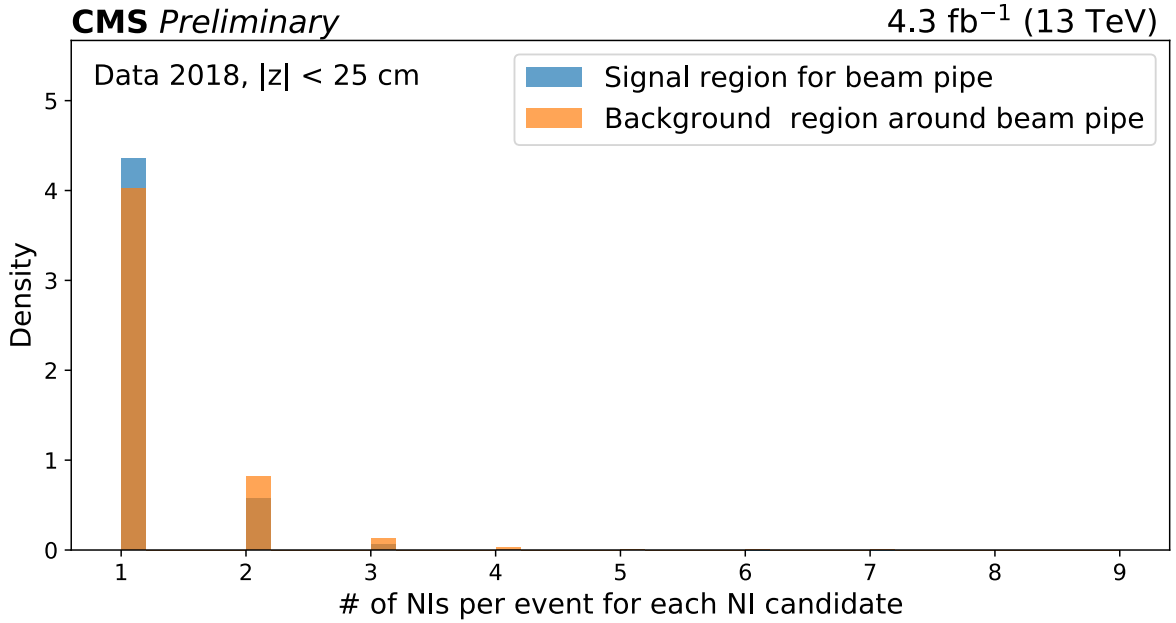
Model Training

Model is trained in Keras with:

- 300 epochs – number of passes of the Train Set to the NN
- batch size = 2^{13} = 8192 – number of events with which you update trained parameters once
- class weight = {0: 3.0, 1: 1.0}, because we have 3 times less real NIs compared to combinatorial background, injected to NN for training
- loss = 'categorical crossentropy' – $-\frac{1}{N}\sum_{i=1}^{N} class_weight \times \log\left(Y_{predict}\right)\left[Y_{true}^{i} \in Class^{i}\right]$
- optimizer = 'adam' with learning rate = 0.005 (5 times more then default one)

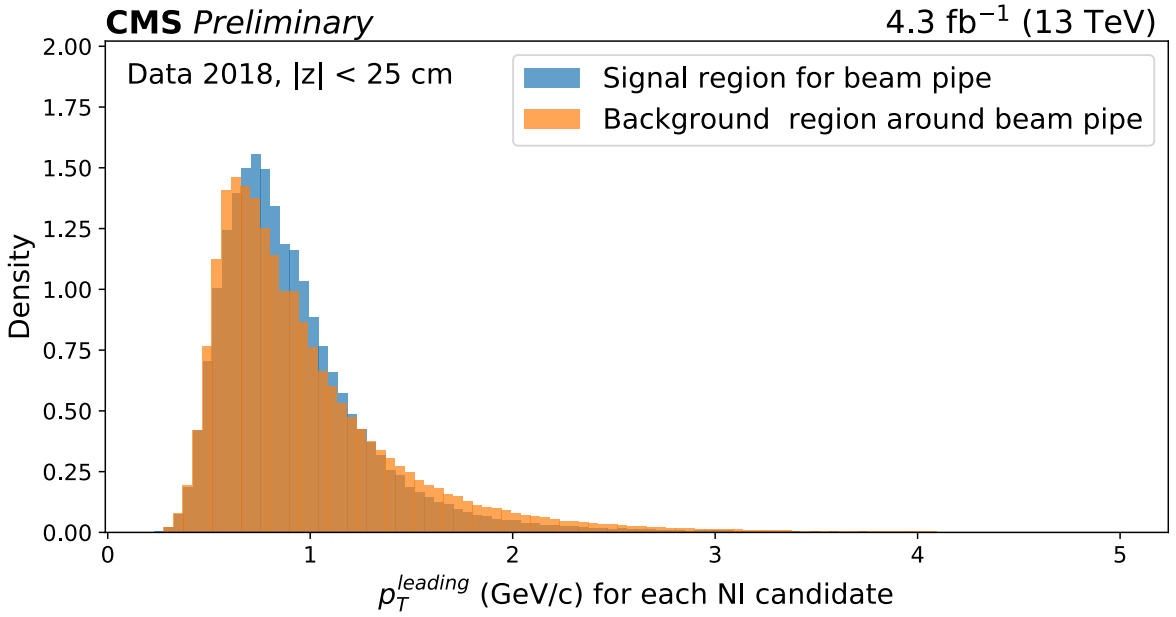
Input Variables to the NN near Beam Pipe (1)

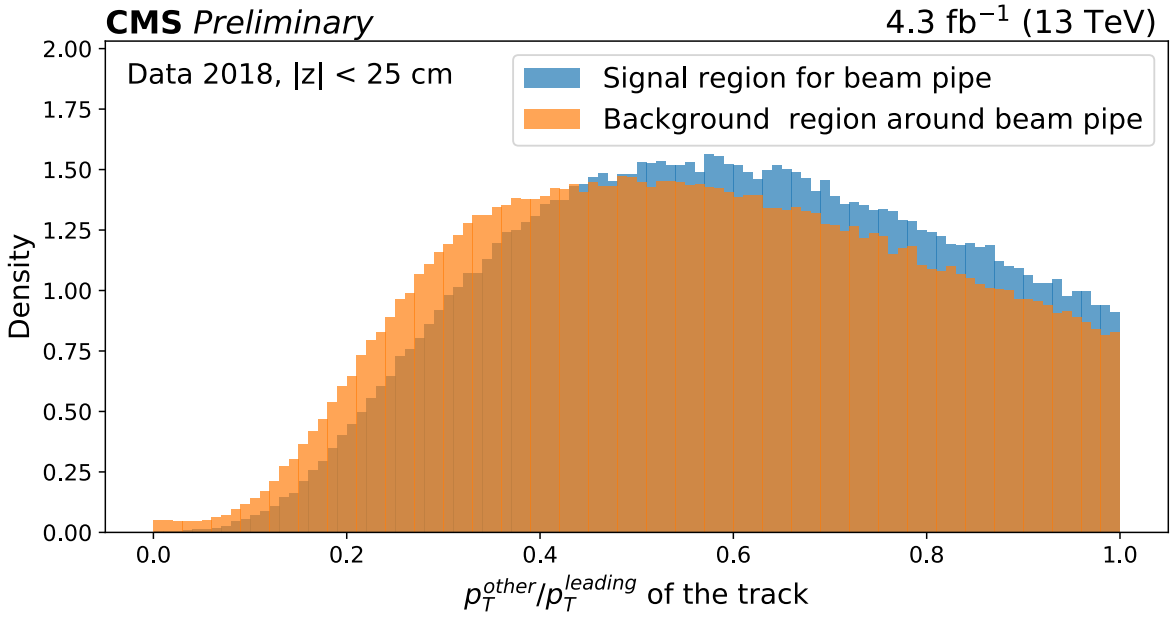




Density of primary vertices (left) and number of NIs (right), reconstructed in the event, for each NI candidate in barrel region ($|z| < 25$ cm) near beam pipe. The blue and orange curves correspond to the signal and background regions, respectively. Shape differences between signal and background distributions are observed.

Input Variables to the NN near Beam Pipe (2)

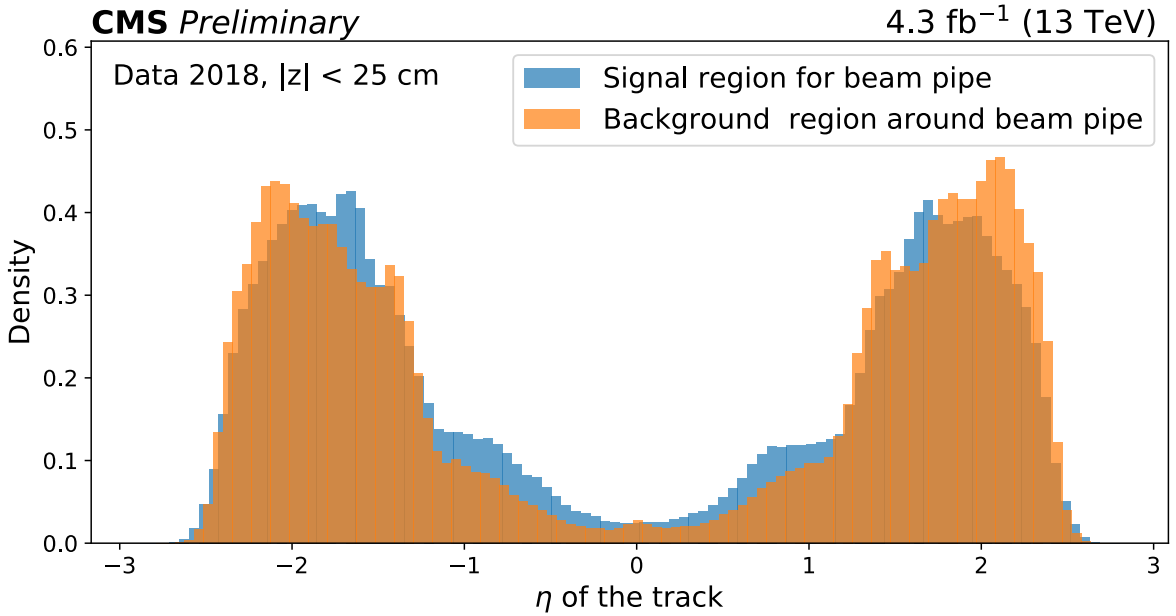


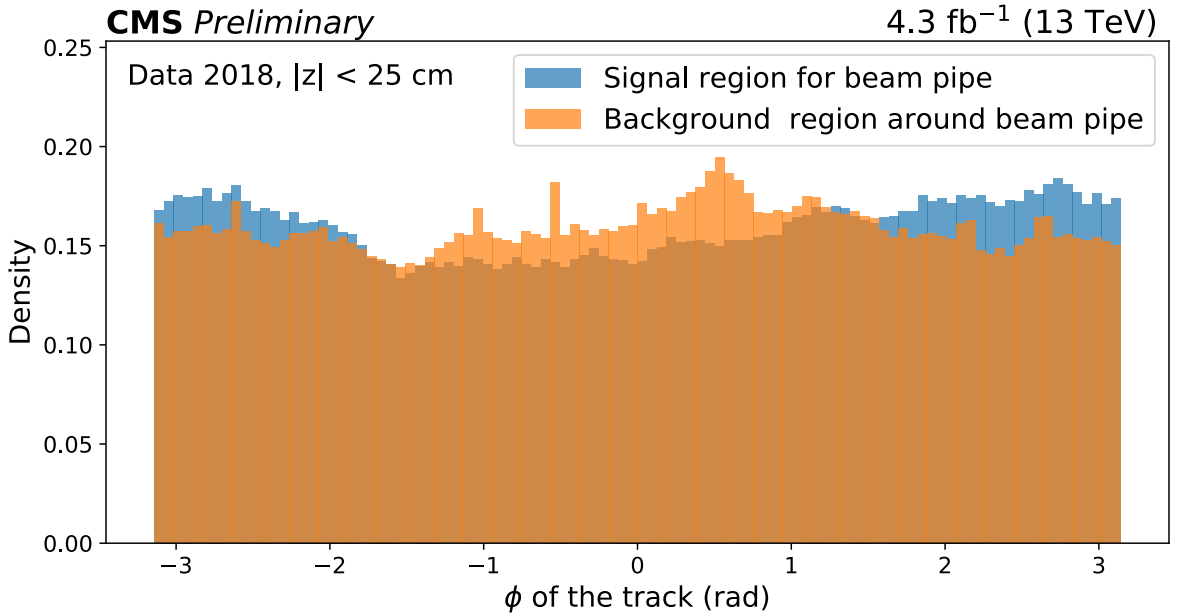


Leading track of NI candidate is the track with maximum value of transverse momentum, p_T, from all tracks belonging to the same vertex.

Density of p_T for leading track, $p_T^{leading}$ (left) and normalized ratio of the p_T for two other tracks to the leading track, $p_T^{other}/p_T^{leading}$, (right) for each NI candidate in barrel region ($|z| < 25$ cm) near beam pipe. The blue and orange curves correspond to the signal and background regions, respectively. Shape differences between signal and background distributions are observed.

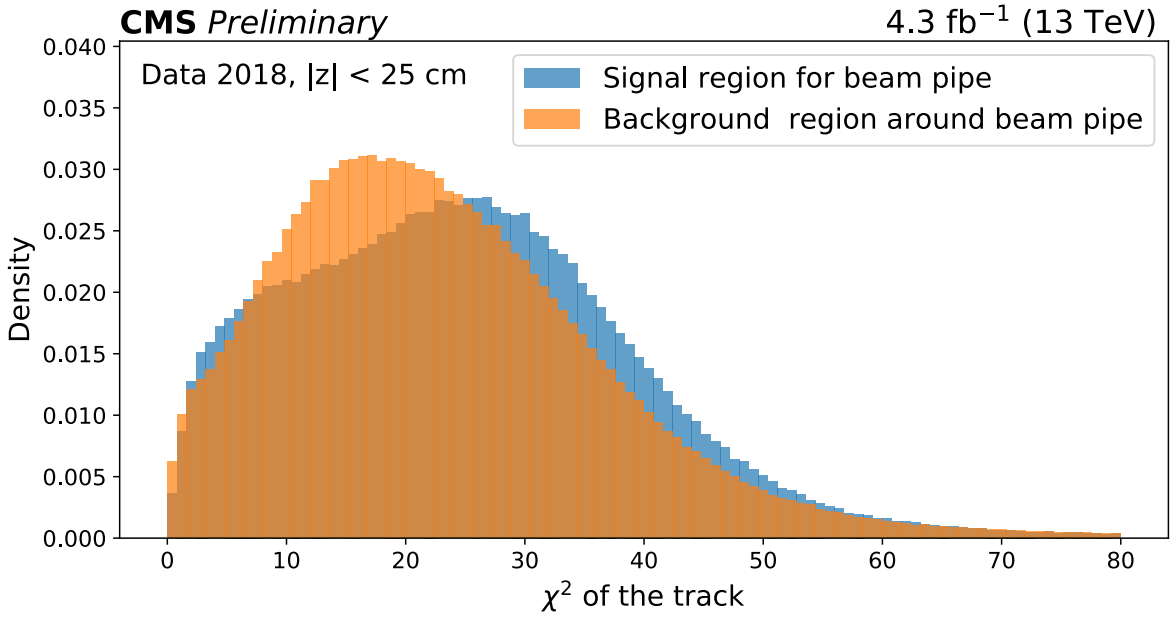
Input Variables to the NN near Beam Pipe (3)

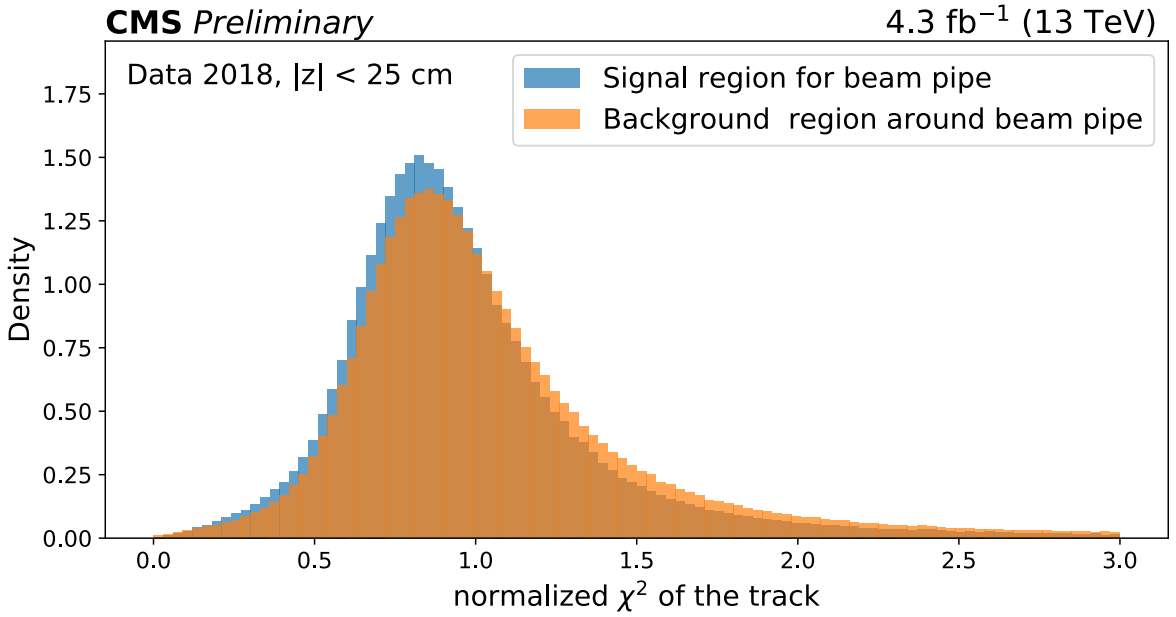




Density of pseudorapidity, η (left) and polar angle, ϕ (right) for the tracks from NI candidates in barrel region ($|z| < 25$ cm) near beam pipe. The blue and orange curves correspond to the signal and background regions, respectively. Shape differences between signal and background distributions are observed.

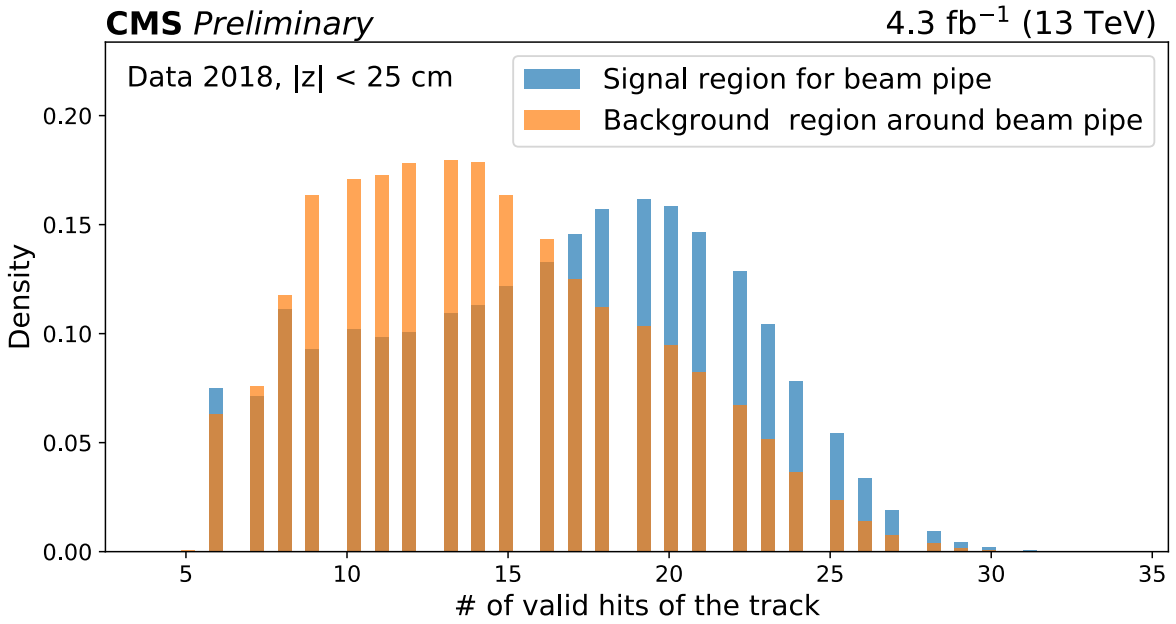
Input Variables to the NN near Beam Pipe (4)

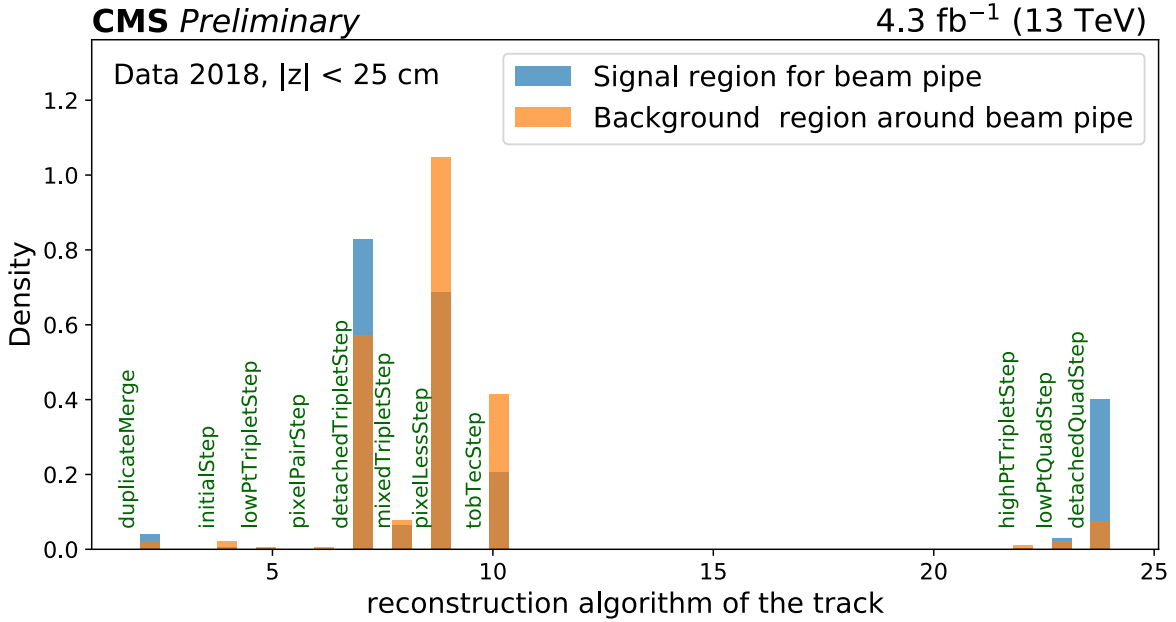




Density of fit quality, χ^2 (left) and normalized χ^2 (right) for the tracks from NI candidates in barrel region ($|z| < 25$ cm) near beam pipe. The blue and orange curves correspond to the signal and background regions, respectively. Shape differences between signal and background distributions are observed.

Input Variables to the NN near Beam Pipe (5)





Density of number of valid hits (left) and track reconstruction algorithm (right) for the tracks from NI candidates in barrel region ($|z| < 25$ cm) near beam pipe. The blue and orange curves correspond to the signal and background regions, respectively. Shape differences between signal and background distributions are observed. Track reconstruction algorithms are described in CMS-DP-2017-015.