Graph Neural Network for τ identification

Authors: Kirill Bukin, Leonid Didukh

Supervisors:
Leonid Didukh
Dirk Kruecker
Isabell Melzer-Pellmann

August 28, 2019

Introduction

		τ Decay Mode	Branching Fraction (%)
Leptonic		$\tau^{\pm} \to e^{\pm} + \overline{\nu}_e + \nu_\tau$	17.84 ± 0.04
		$\tau^{\pm} \to \mu^{\pm} + \overline{\nu}_\mu + \nu_\tau$	17.41 ± 0.04
Hadronic	One-prong	$\tau^{\pm} \to \pi^{\pm} + (\geq 0\ \pi^0) + \nu_\tau$	49.46 ± 0.10
		$\tau^{\pm} \to \pi^{\pm} + \nu_\tau$	10.83 ± 0.06
		$\tau^{\pm} \to \rho^{\pm}(\to \pi^{\pm} + \pi^0) + \nu_\tau$	25.52 ± 0.09
		$\tau^{\pm} \to a_1(\to \pi^{\pm} + 2\pi^0) + \nu_\tau$	9.30 ± 0.11
		$\tau^{\pm} \to \pi^{\pm} + 3\pi^0 + \nu_\tau$	1.05 ± 0.07
		$\tau^{\pm} \to h^{\pm} + 4\pi^0 + \nu_\tau$	0.11 ± 0.04
Hadronic	Three-prong	$\tau^{\pm} \to \pi^{\pm} + \pi^{\mp} + \pi^{\pm} + (\geq 0\pi^0) + \nu_\tau$	14.57 ± 0.07
		$\tau^{\pm} \to \pi^{\pm} + \pi^{\mp} + \pi^{\pm} + \nu_\tau$	8.99 ± 0.06
		$\tau^{\pm} \to \pi^{\pm} + \pi^{\mp} + \pi^{\pm} + \pi^0 + \nu_\tau$	2.70 ± 0.08

- Goal of our work: discriminate hadronically decaying tau from quark and gluon jets using graph learning techniques
- Results are compared with Deep Neural Network called Deep Particle Flow (DPF) and BDT-based MVA classifier

Tau Isolation Discriminators

Different discriminators were developed to discriminate τ_h from jets, e and μ

- MVA: boosted decision trees - based discriminator
- DeepPF: DNN-based discriminator with 10 convolutional layers and 4 fully-connected layers
- DeepTau ID: DNN-based discriminator

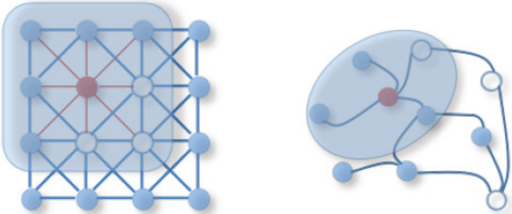
Graph Convolutional Network

https://arxiv.org/abs/1902.08570

- tau are reconstructed from particle flow candidates
- Variables from particles contained in cone centered on the tau are represented as a point cloud (each particle represented as a point)
- Graph is constructed from a particle cloud as k-nearest neighbour graph
- Graph fed into Graph Convolutional Neural Network (referred to as Edge Convolutional Network, or ECN), which consists of graph convolutional layers and FC layers

Graph Convolutional Network

https://arxiv.org/abs/1901.00596



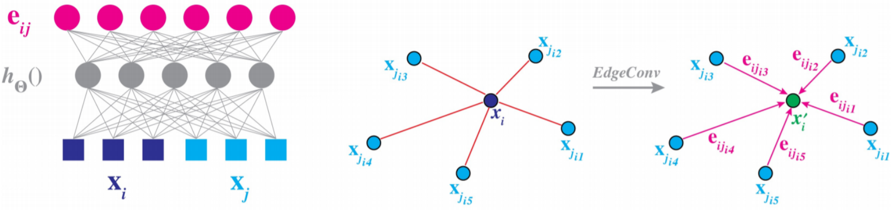
(a) 2D Convolution. Analogous to a graph, each pixel in an image is taken as a node where neighbors are determined by the filter size. The 2D convolution takes a weighted average of pixel values of the red node along with its neighbors. The neighbors of a node are ordered and have a fixed size.

(b) Graph Convolution. To get a hidden representation of the red node, one simple solution of graph convolution operation takes the average value of node features of the red node along with its neighbors. Different from image data, the neighbors of a node are unordered and variable in size.

Fig. 1: 2D Convolution vs. Graph Convolution.

EdgeConv Layer

https://arxiv.org/abs/1801.07829



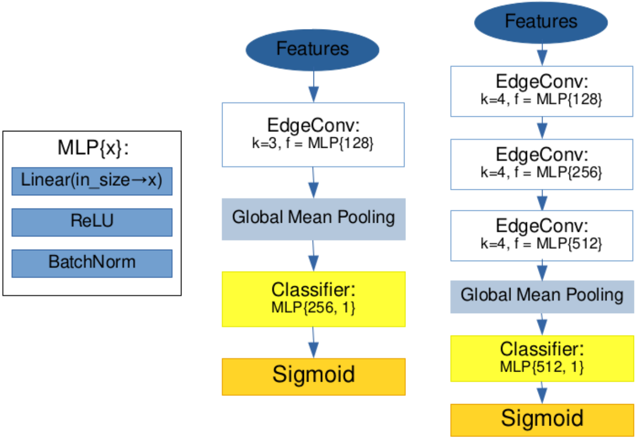
Edge features are defined as:

$$e_{ij} = h_\Theta(x_i, x_j)$$

EdgeConv operation is defined by applying a symmetric aggregation operation $\square$ (e.g., $\sum$ or max):

$$x'_i = \square h_\Theta(x_i, x_j)$$

Edge Convolutional Network

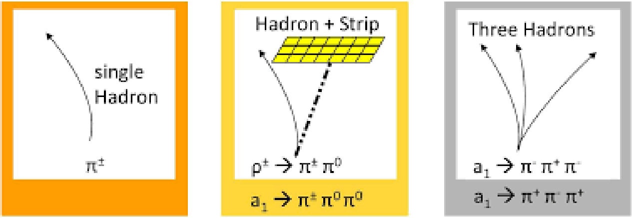


Left: MLP layer; **Center:** ECN with 1 conv layer; **Right:** ECN with 3 conv layers

MC samples

2016 MC:

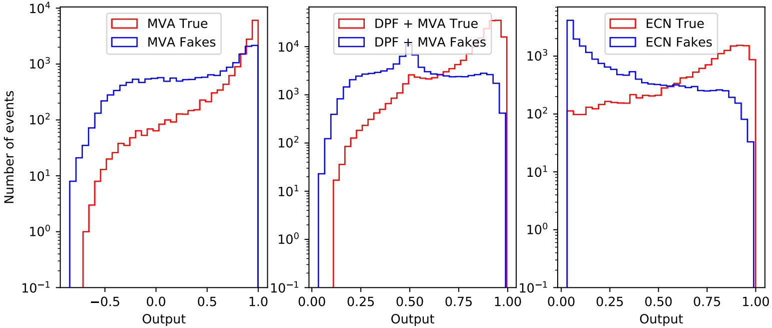
- True τ: Drell-Yan ($Z \to \tau\tau$)
- Fake τ: W+jets (jet misidentified as τ)
- In both cases, only following three τ decay modes are used:



Decay mode 0 Decay mode 1 Decay mode 10

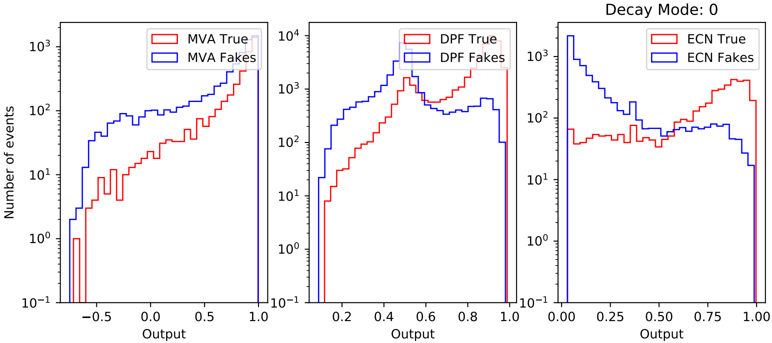
- Reweighting is applied to flatten in tau p_T spectrum

Output distribution



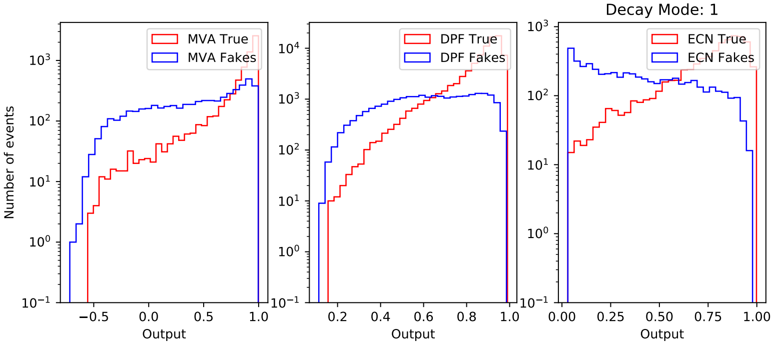
Output distribution for all the decay modes combined

Output distributions for different decay modes



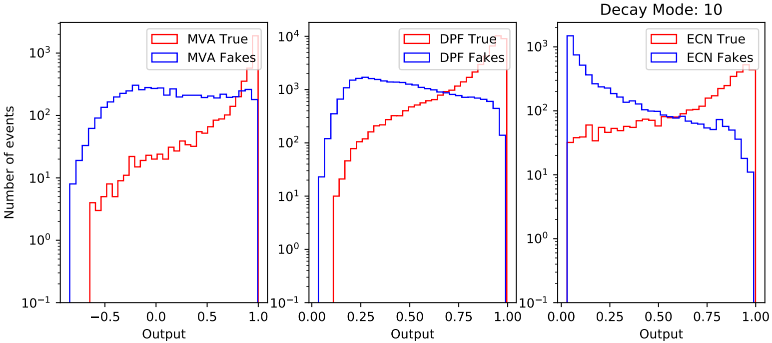
Output distribution for decay mode 0

Output distributions for different decay modes



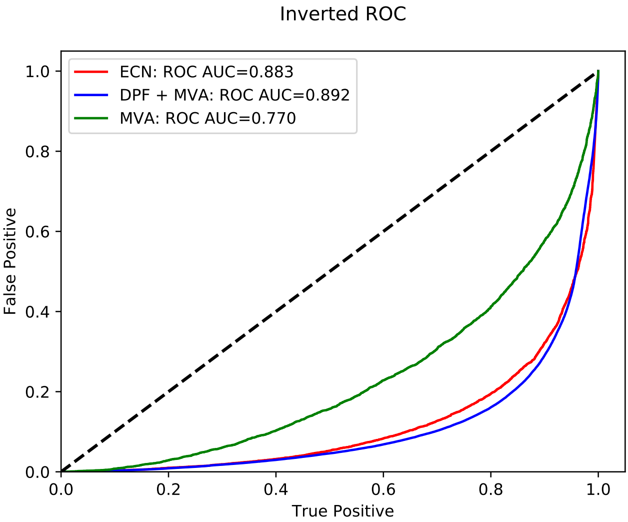
Output distribution for decay mode 1

Output distributions for different decay modes

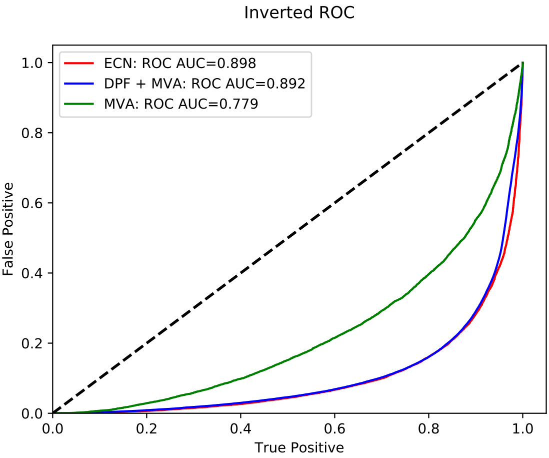


Output distribution for decay mode 10

ECN performance (1 Convolutional layer)



ECN performance (3 Convolutional layers)



ECN has performance close to the ensemble of DPF and MVA while the number of parameters is much smaller

Results

Architecture	ROC AUC score	Number of parameters	Time [ms]
MVA	0.792	-	-
DPF	0.863	8838945	166
DPF + MVA	0.892	-	-
ECN (1 layer)	0.883	**49283**	**0.6**
ECN (3 layers)	**0.898**	223939	2.6

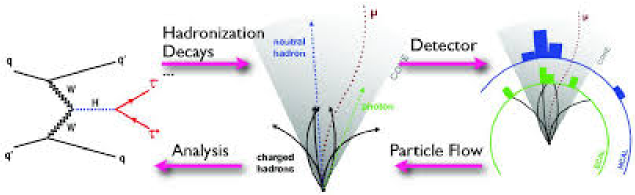
- Time was evaluated on CPU Intel Xeon E5-2650 with 1 thread and batch size of 2048
- Our network has smaller number of parameters and time consumption by the factors of 40 and 64 correspondingly

Conclusions and plans

- Neural Network with Graph Convolutions has performance close to the one of DPF while having much smaller number of parameters
- Possible next steps:
  - Optimize the hyperparameters:
    - Depth of the network
    - Batch size
    - Initial learning rate
    - Number k for knn-graph
  - Train multiclass classificator

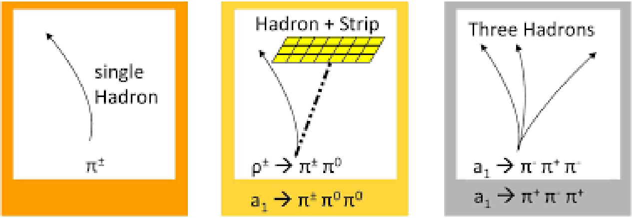
Backup

PF algorithm



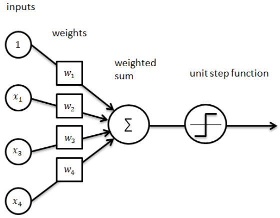
The Particle Flow event reconstruction aims at reconstructing all stable particles arising from collisions, using combined response from all the CMS sub-detectors. This list of individual particles is then used to build jets, to determine the missing transverse energy, to reconstruct and identify taus from their decay products, etc.
https://cds.cern.ch/record/2029414/

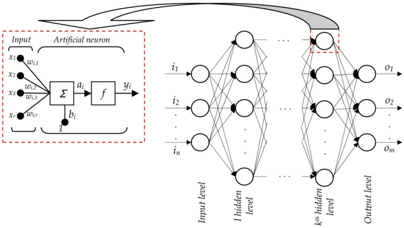
Hadron Plus Strip



The HPS algorithm uses PF candidates to check whether the combined topology is compatible with one of the hadronic decay modes

Artificial Neural Networks



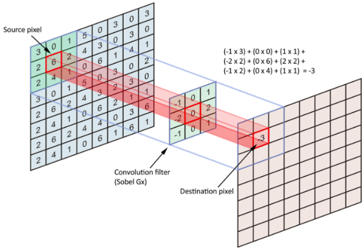


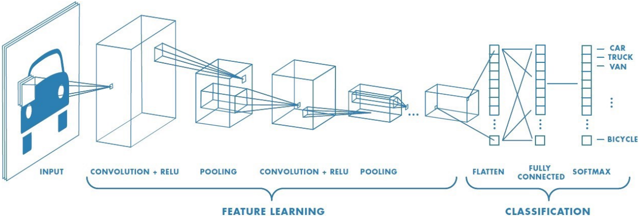
- Linear model is able to approximate simple functions
- Several linear models with nonlinear activations between them can approximate arbitrary functions (Universal approximation theorem)
- Loss function is introduced to describe the value of model's prediction error:

$$BCE = \frac{1}{N}\sum_{i=0}^{N} y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)$$

- Model training is the minimization of the loss function, usually with different modifications of gradient descent

Convolutional Neural Networks

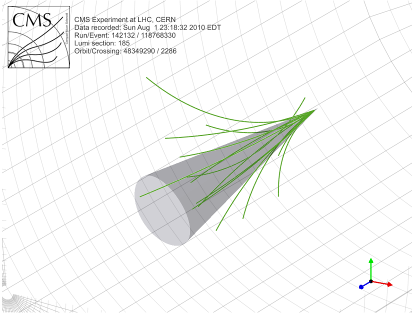




This convolutions are used in the DPF network

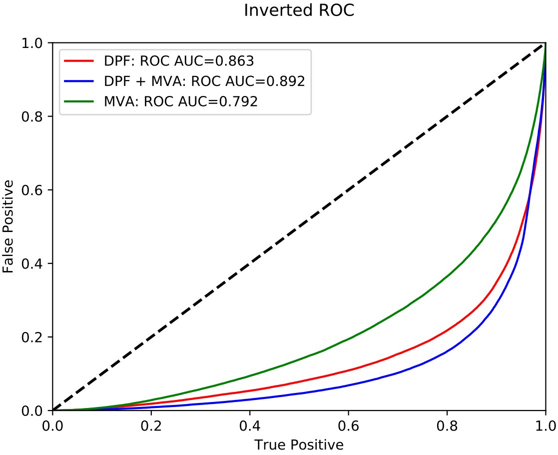
DPF network (Author: Owen Colegrove)

- tau are reconstructed from particle flow particles
- Variables from particles contained in cone centered on the tau are formatted into a 2-D table and fed into a Deep Neural Network
- Each event are represented as matrix 60*47(number of particles - 60; number of features - 47)
- If number of particles in cone is less then 60, the last rows of table are filled with zeroes

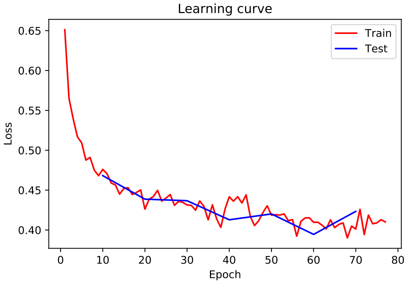


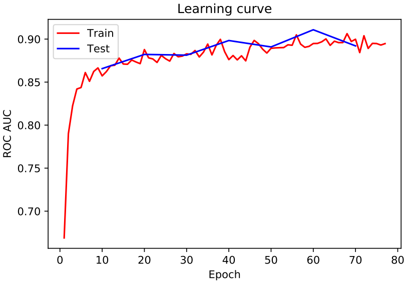
Particles	feature1	...	featureN
p1	f1,p1		fN,p1
...			
pm	f1,pm		fN,pm

DPF performance



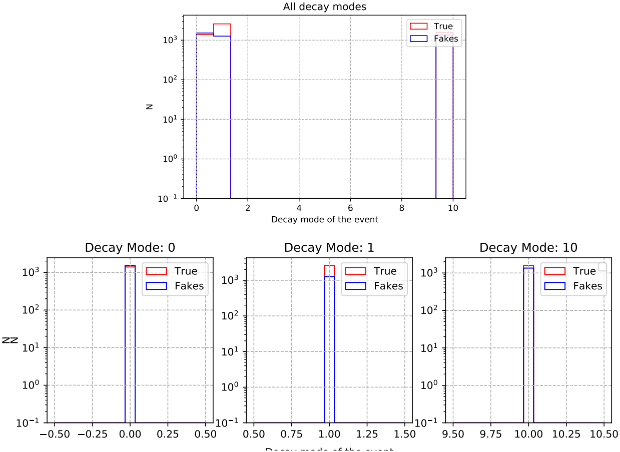
Learning curves for ECN with 3 layers



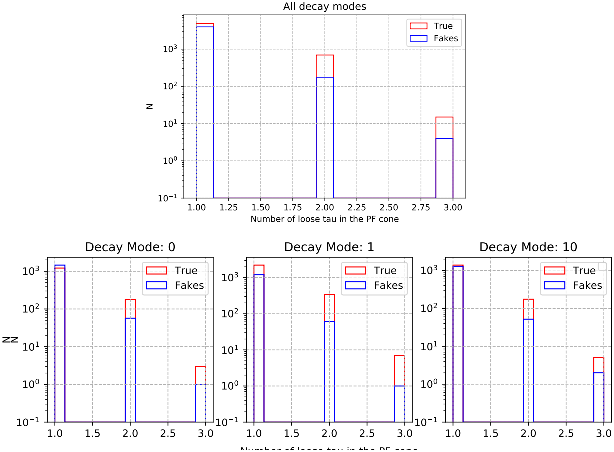


There is no significant difference between results on the train and test sets, i.e. there are no overfitting

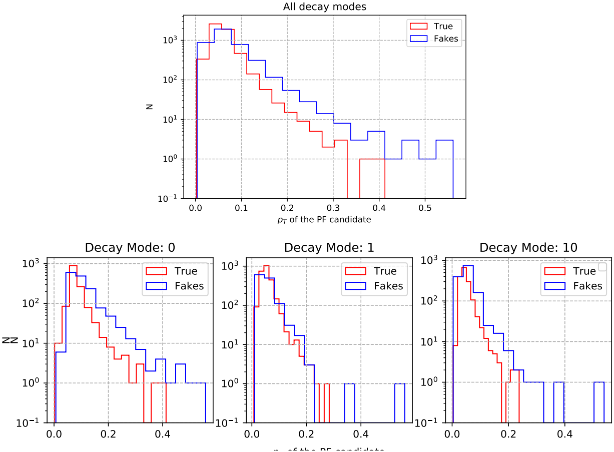
Feature: Decay mode of the event



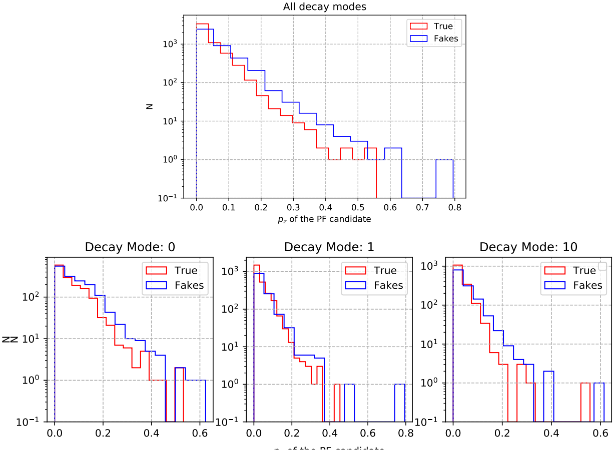
Feature: Number of loose tau in the PF cone



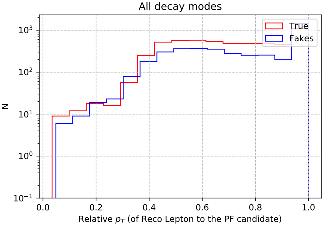
Feature: p_T of the PF candidate

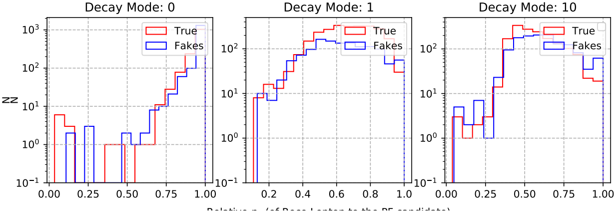


Feature: p_z of the PF candidate

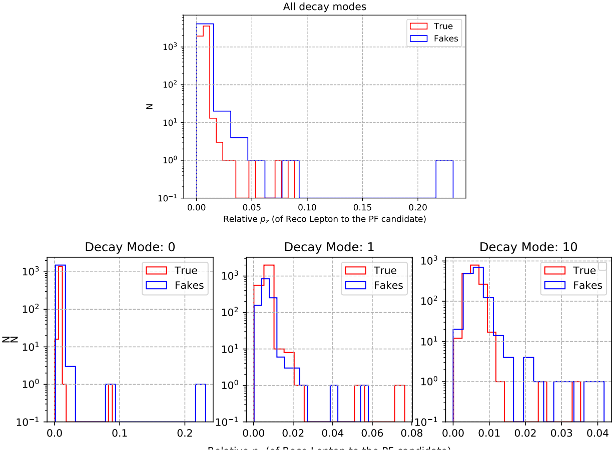


Feature: Relative p_T (of Reco Lepton to the PF candidate)

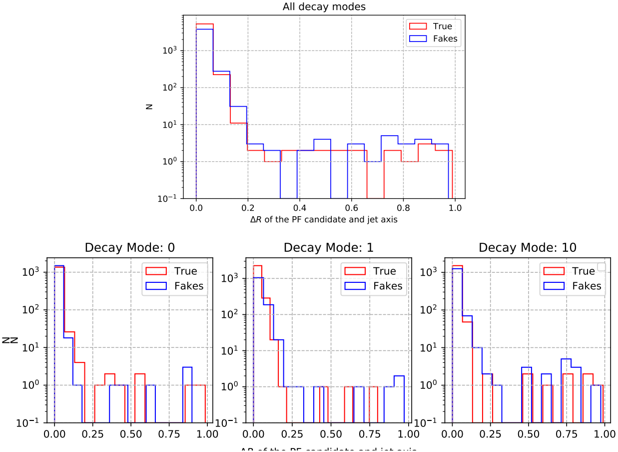




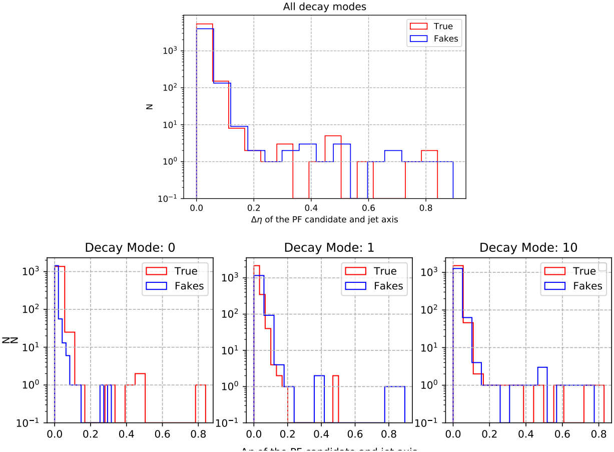
Feature: Relative p_z (of Reco Lepton to the PF candidate)



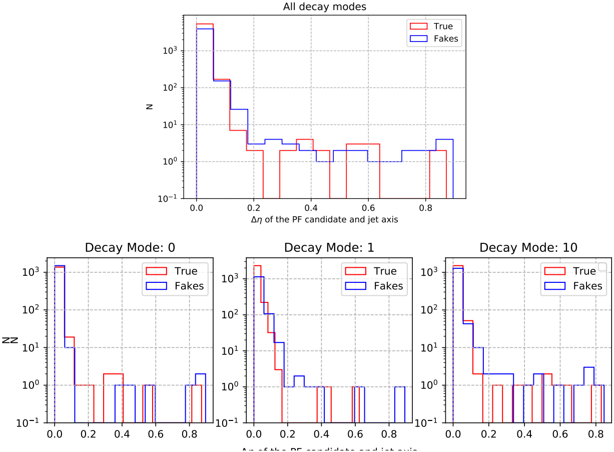
Feature: ΔR of the PF candidate and jet axis



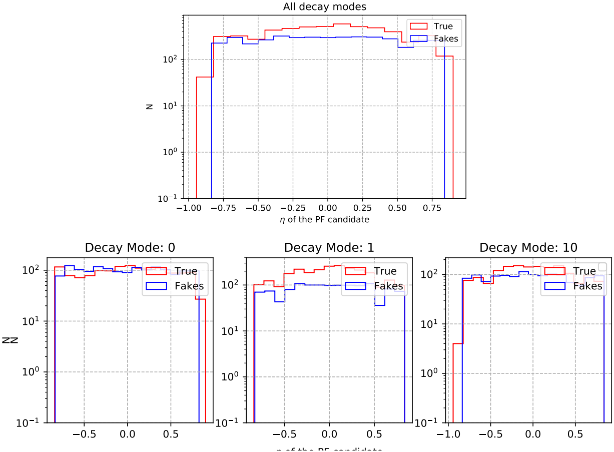
Feature: $\Delta\eta$ of the PF candidate and jet axis



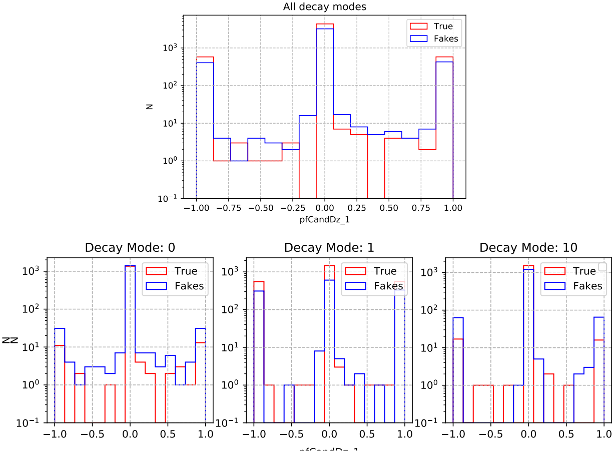
Feature: $\Delta\phi$ of the PF candidate and jet axis



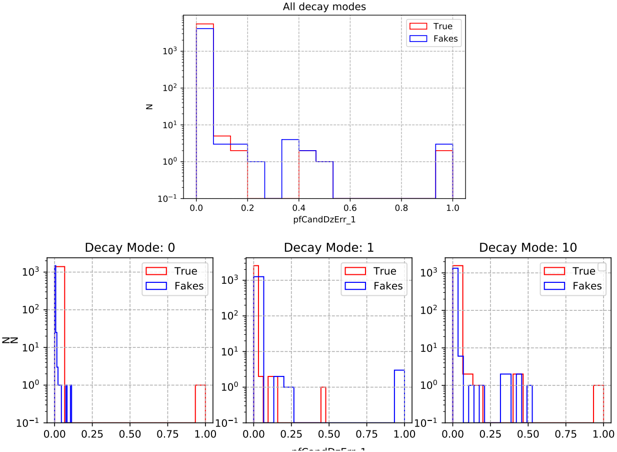
Feature: η of the PF candidate



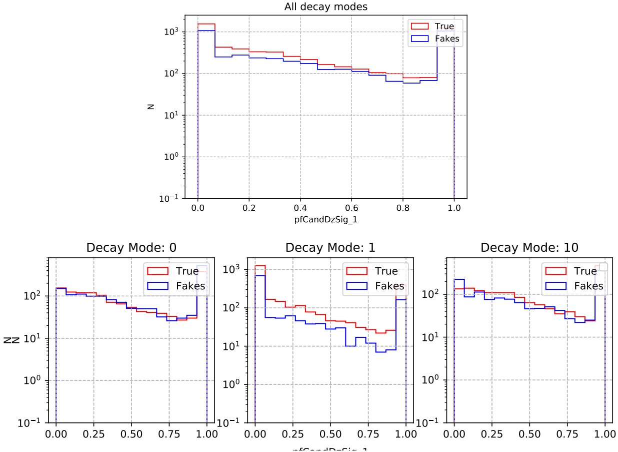
Feature: pfCandDz



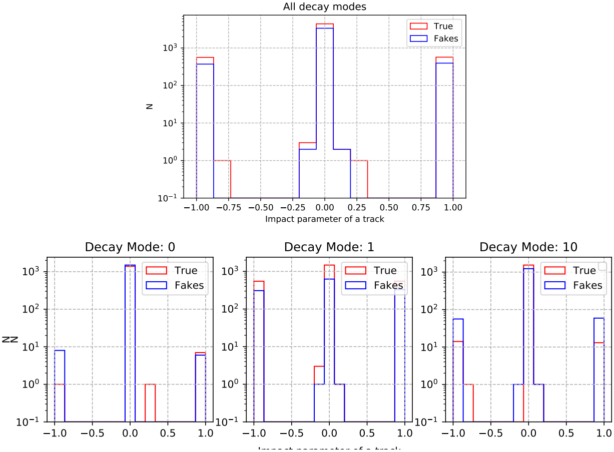
Feature: pfCandDzErr



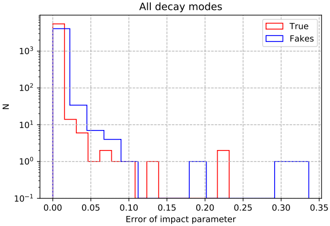
Feature: pfCandDzSig

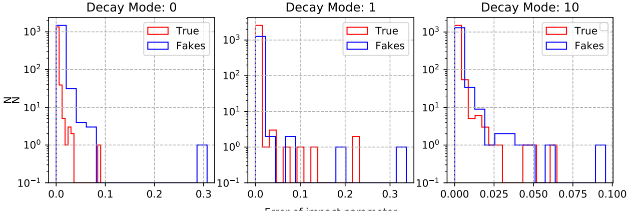


Feature: Impact parameter of a track

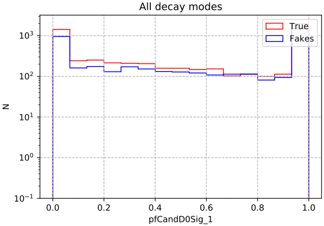


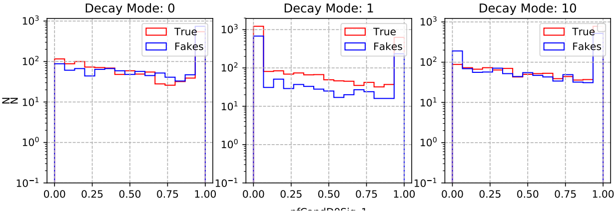
Feature: Error of impact parameter



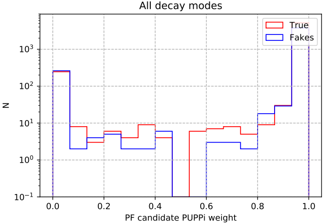


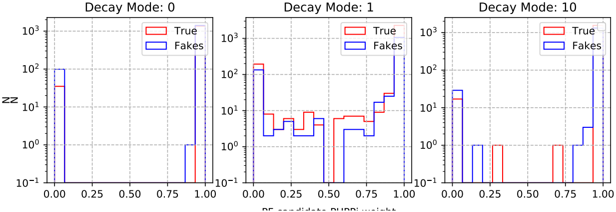
Feature: pfCandD0Sig



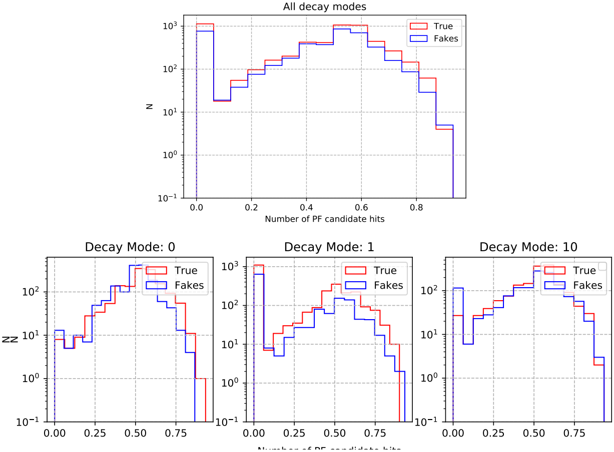


Feature: PF candidate PUPPi weight

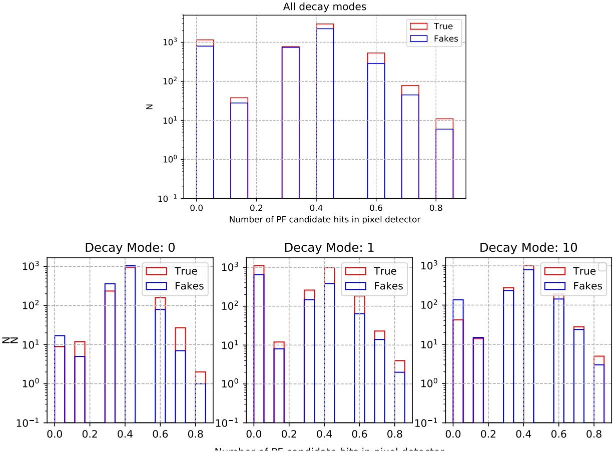




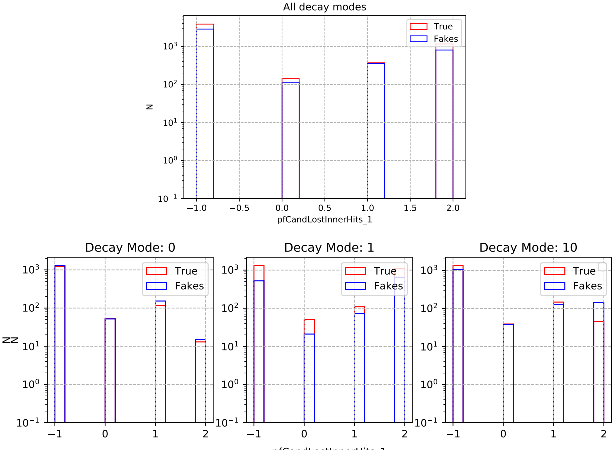
Feature: Number of PF candidate hits



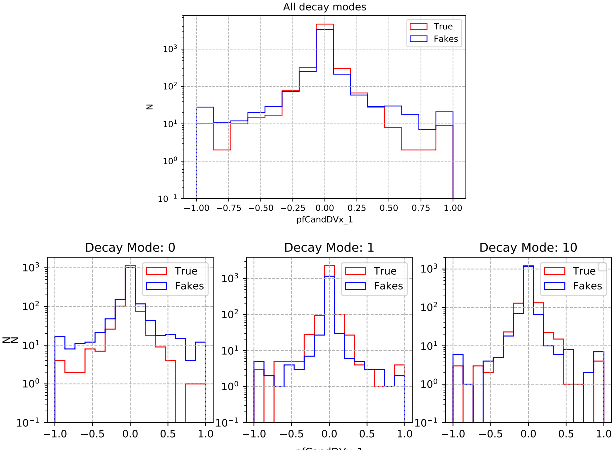
Feature: Number of PF candidate hits in pixel detector



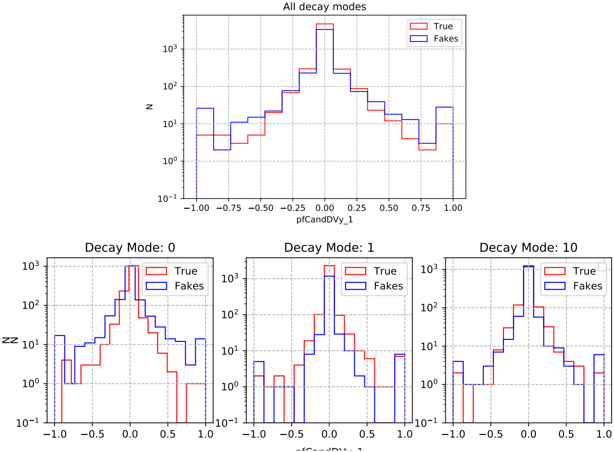
Feature: pfCandLostInnerHits



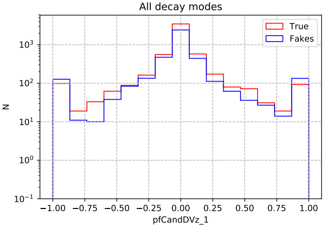
Feature: pfCandDVx

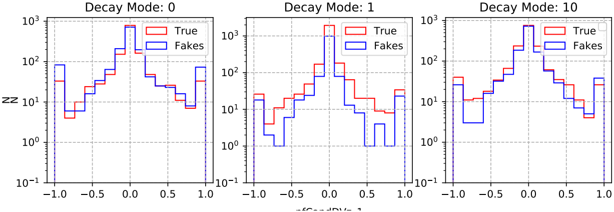


Feature: pfCandDVy

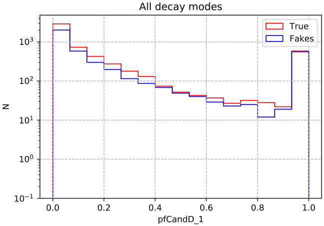


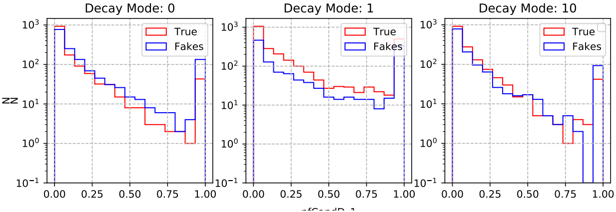
Feature: pfCandDVz



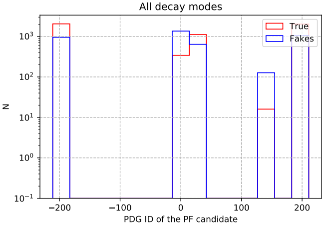


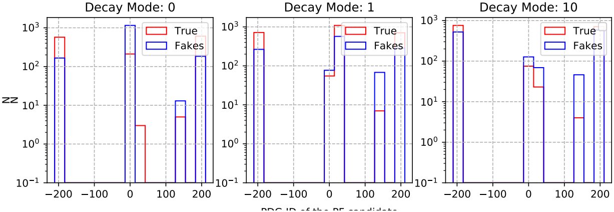
Feature: pfCandD



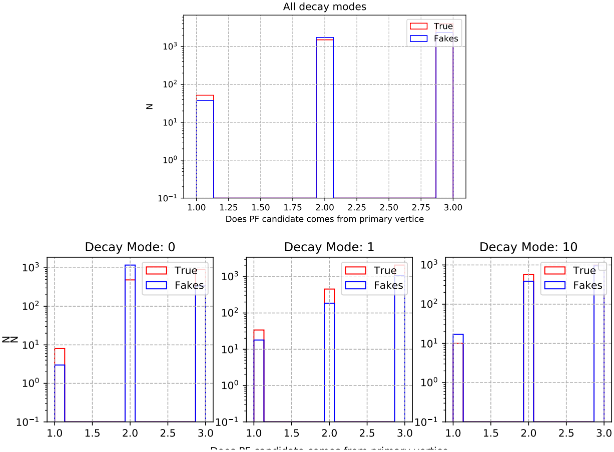


Feature: PDG ID of the PF candidate

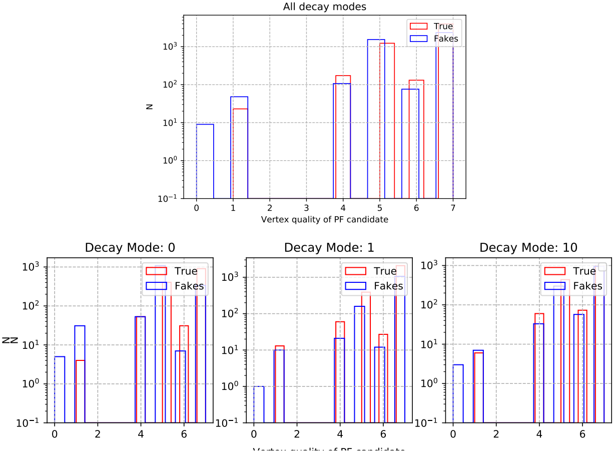




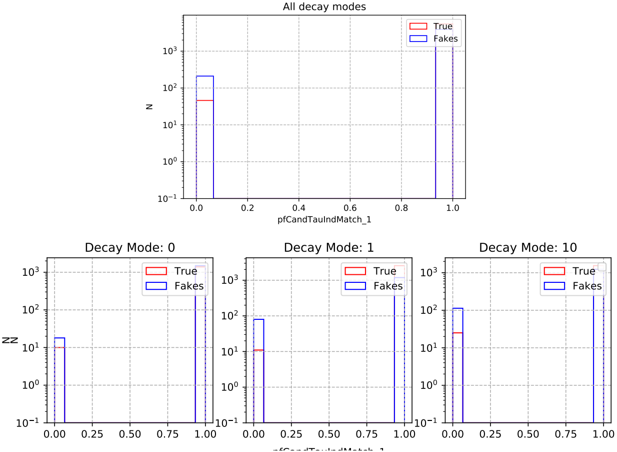
Feature: Does PF candidate comes from primary vertice



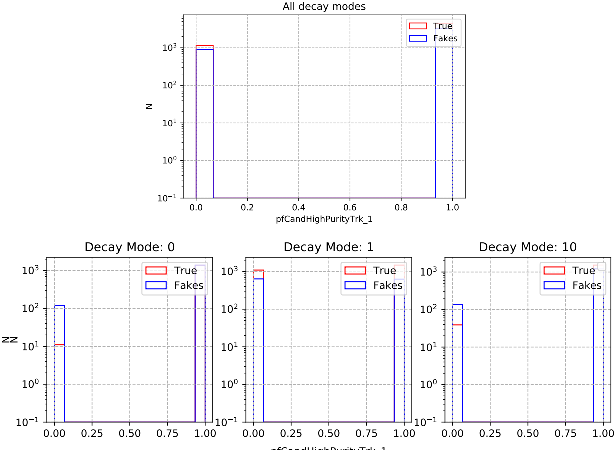
Feature: Vertex quality of PF candidate



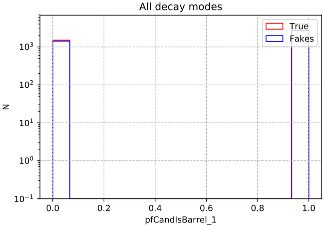
Feature: pfCandTauIndMatch

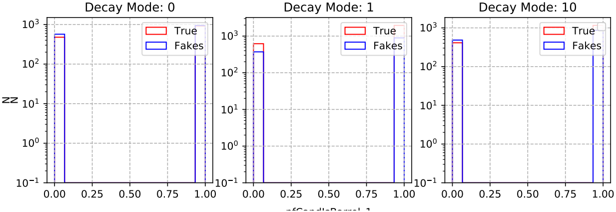


Feature: pfCandHighPurityTrk

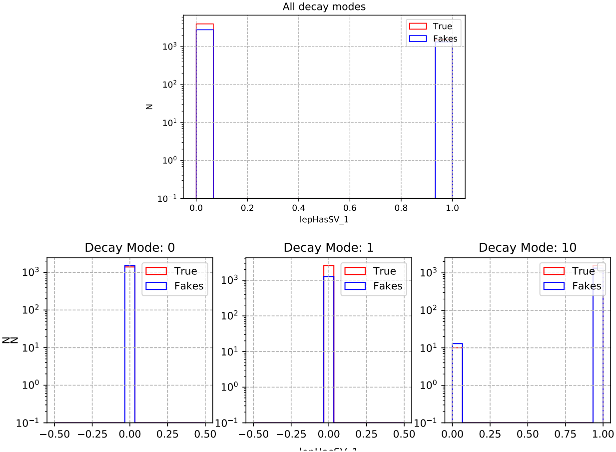


Feature: pfCandIsBarrel

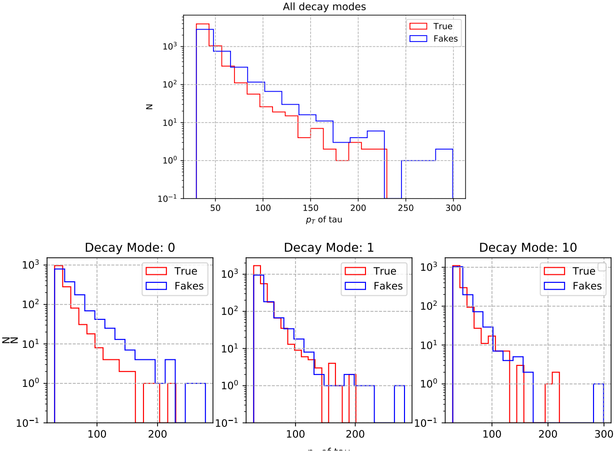




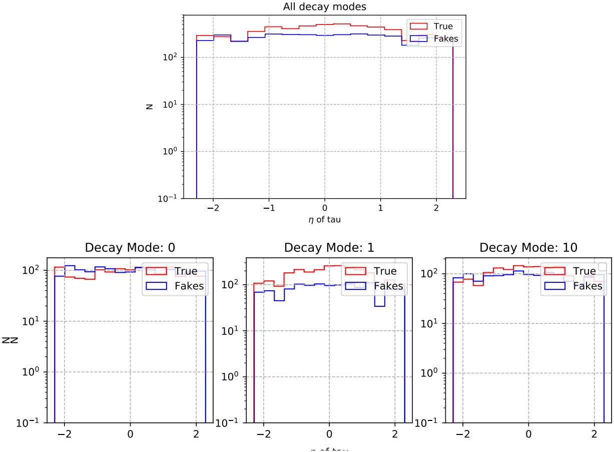
Feature: lepHasSV



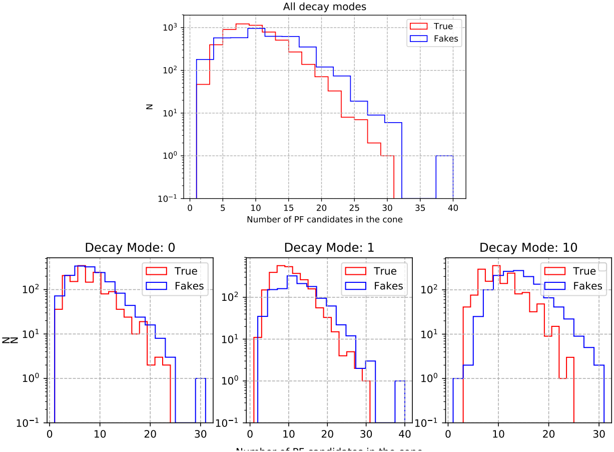
Feature: p_T of tau



Feature: η of tau



Feature: Number of PF candidates in the cone



Feature: Charge of the PF candidate