JOINT INSTITUTE
FOR NUCLEAR RESEARCH

SPD NICA

Identificantion of $\boldsymbol{\mu}$ and $\boldsymbol{\pi}$ tracks based on reconstructed parameters using NN

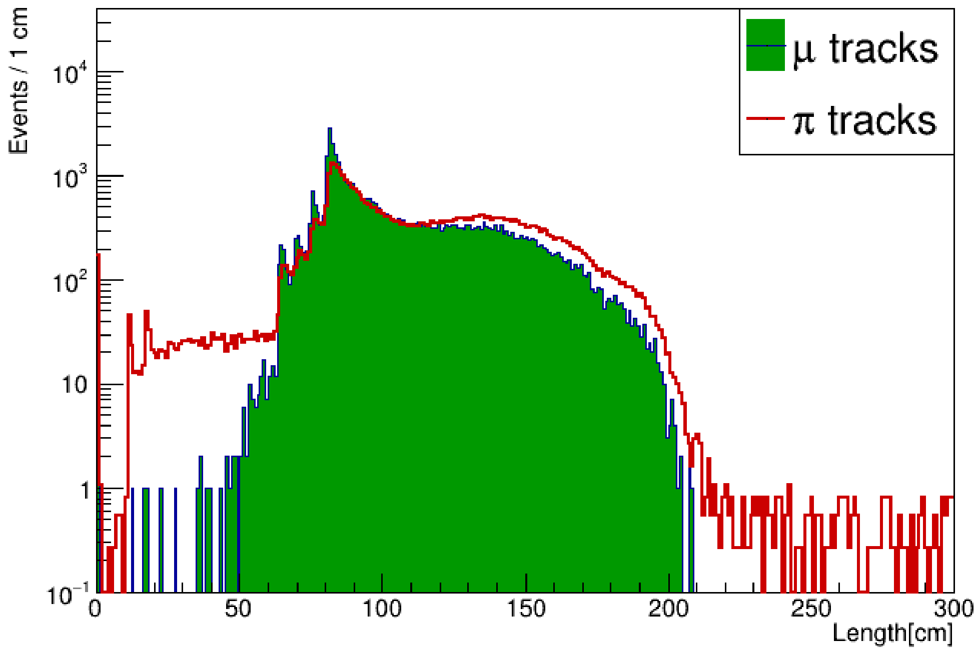
I. Yeletskikh, A.Gridin

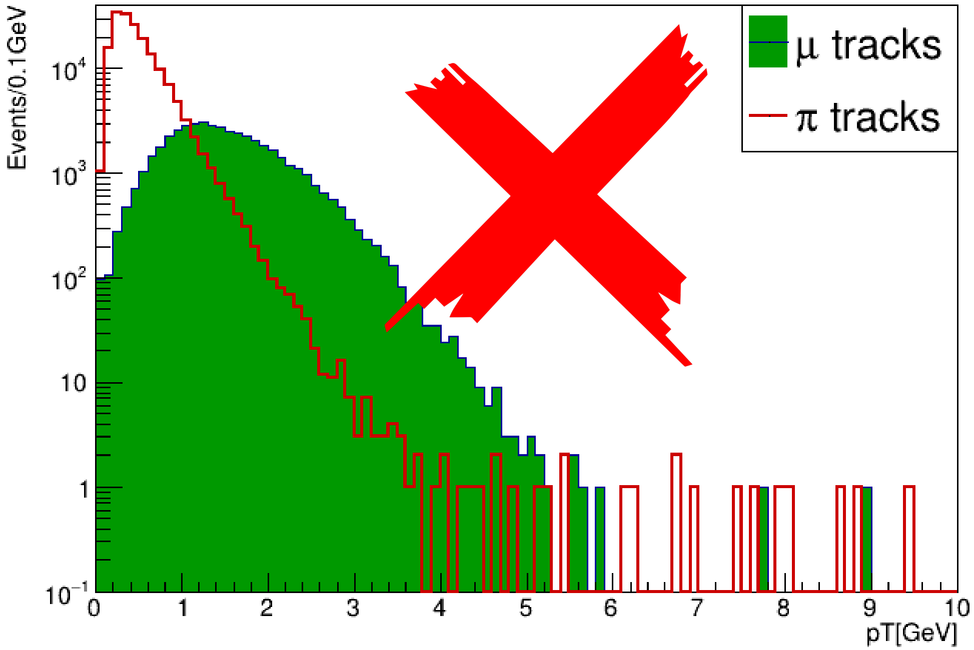
SPD Collaboration meeting, 08.11.2024

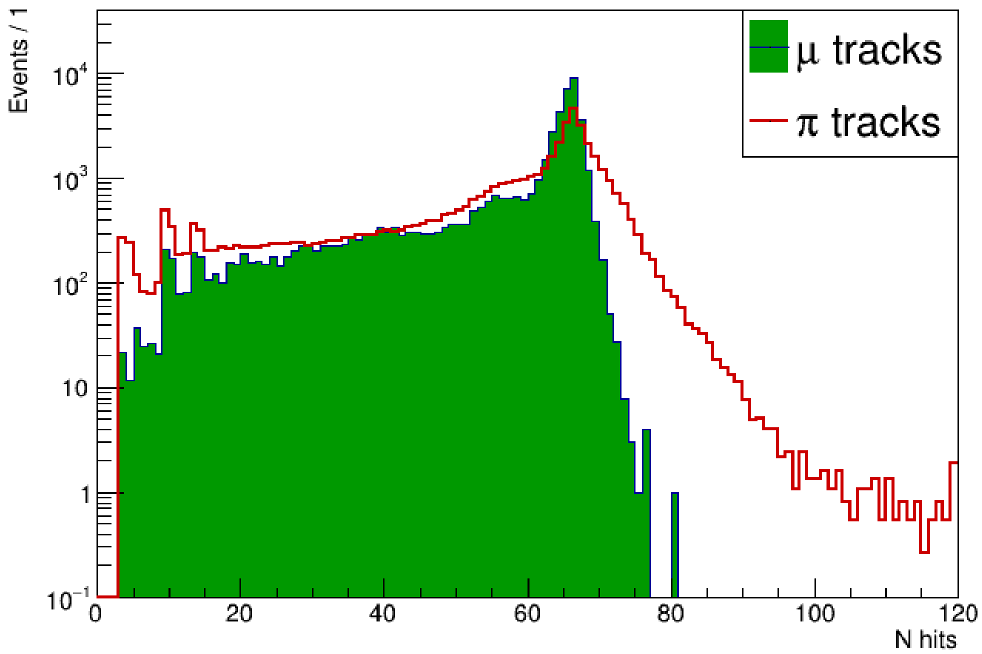
Intro

- Attempt to distinguish pion and muon reconstructed tracks using NN
- ~100K fully reconstructed events (~1.5M tracks) including charmonia production
- Discriminating variables: track length, pt, eta, number of hits in detector subsystems
- ===
- Include dE/dx information? Include vertex information and account for correlations between different tracks in the same event?
- ===
- Tracks are split in bins w.r.t. total momentum
- chi2/ndf is required to be less than 10 to cut marginal track candidates
-
- ===
- Different machine learning approaches are tested.
  - Simple NN based on Keras API is used of structure 140-35-35, Adam optimizer
  - Custom NN API (C++ based) that allow custom evolutionary training algorithm and different options for the optimization (loss) functions and overtraining control.

Track candidates: input variables

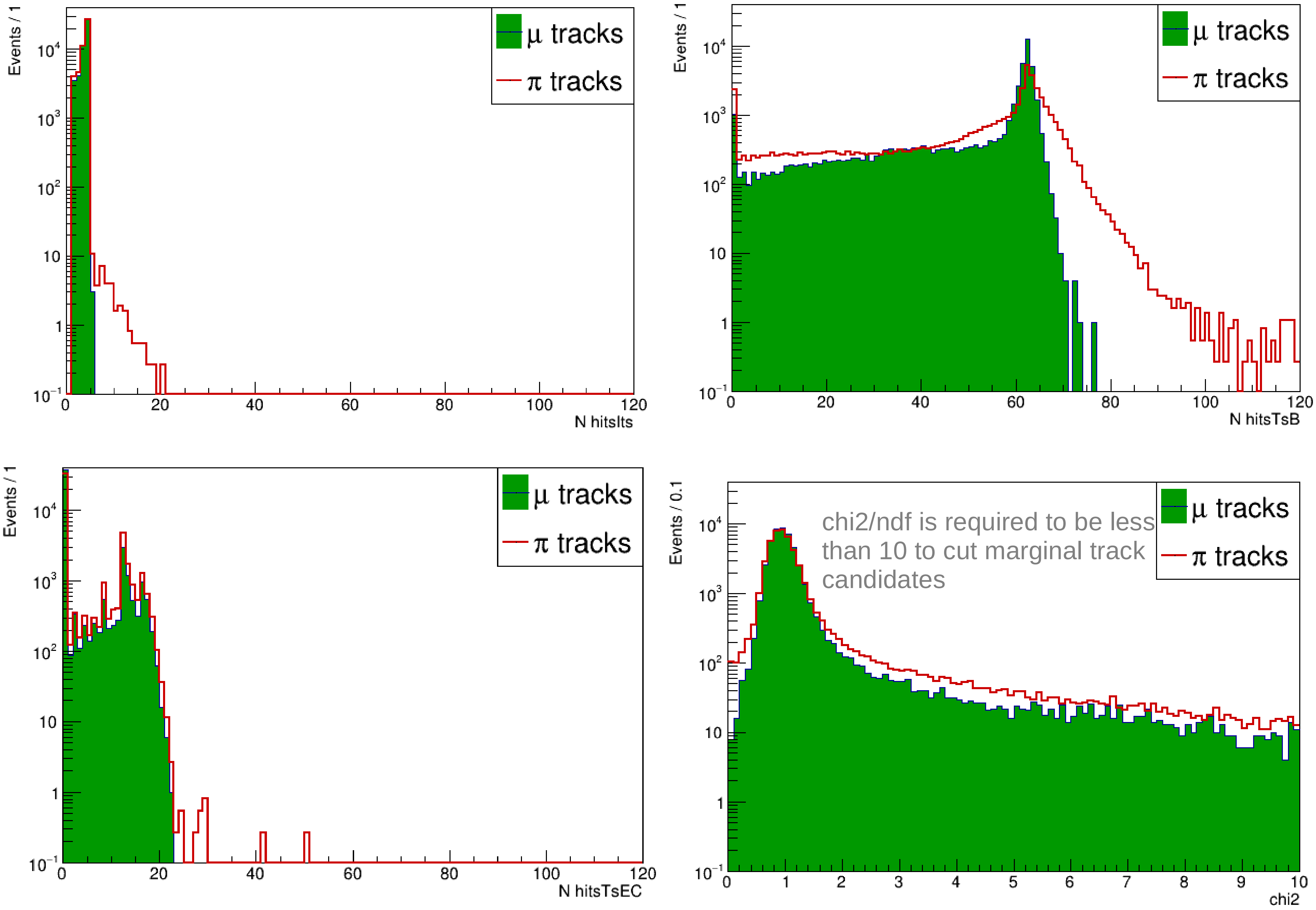






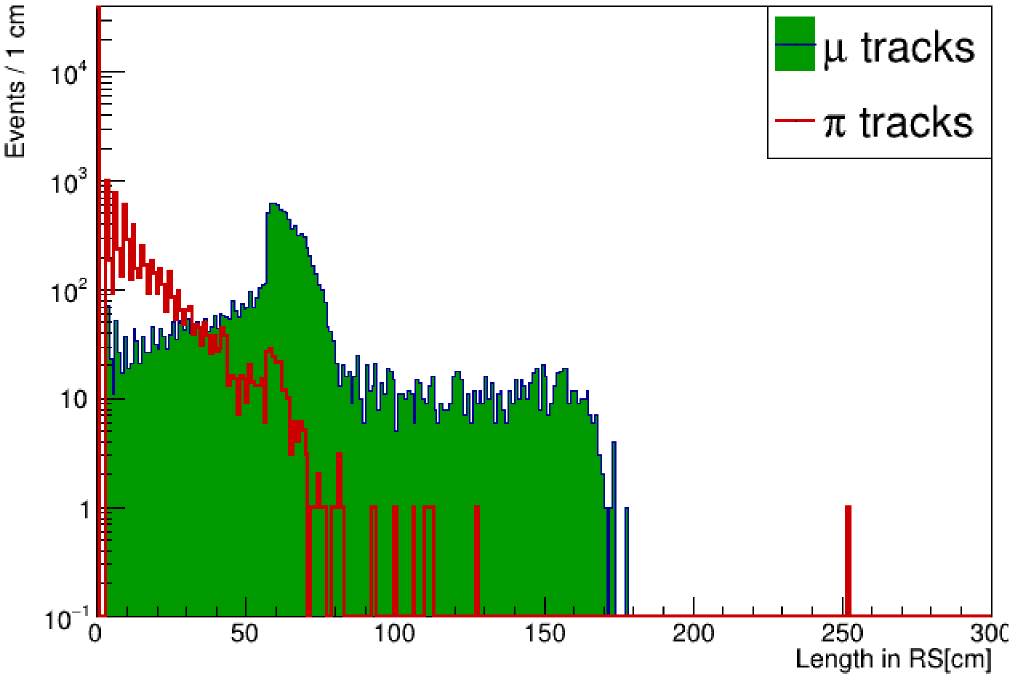
- p_T is not a reliable discrimination variable, since pT spectra strongly depend on the specific physical process contributions.
- p_T can be used on top of the identification result in case muons from specific process (e.g., charmonia decay) are selected

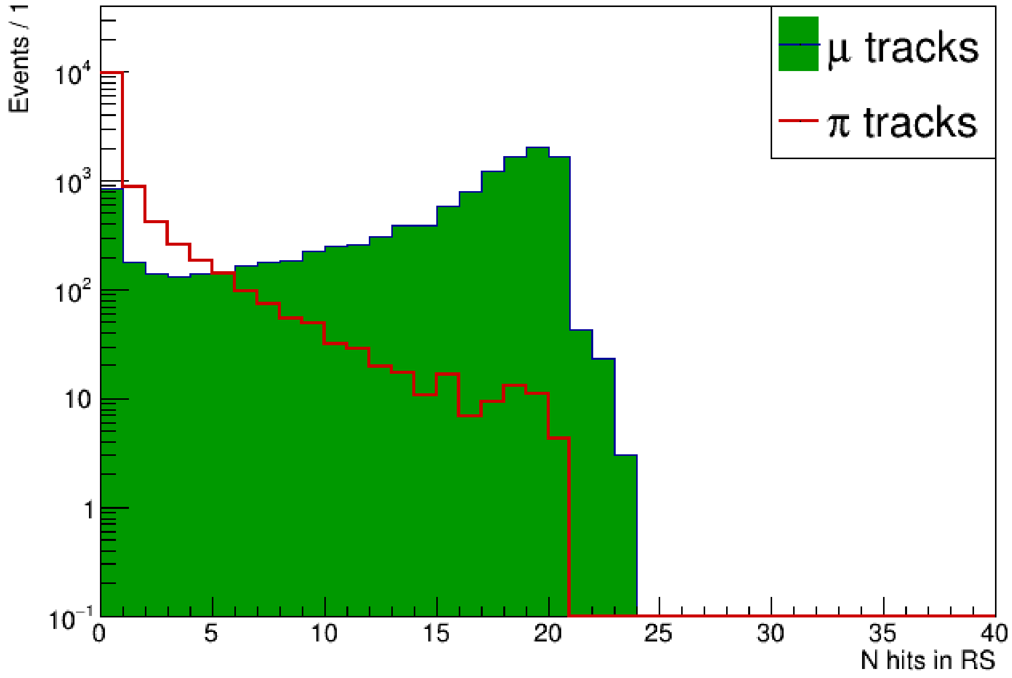
Track candidates: input variables



chi2/ndf is required to be less than 10 to cut marginal track candidates

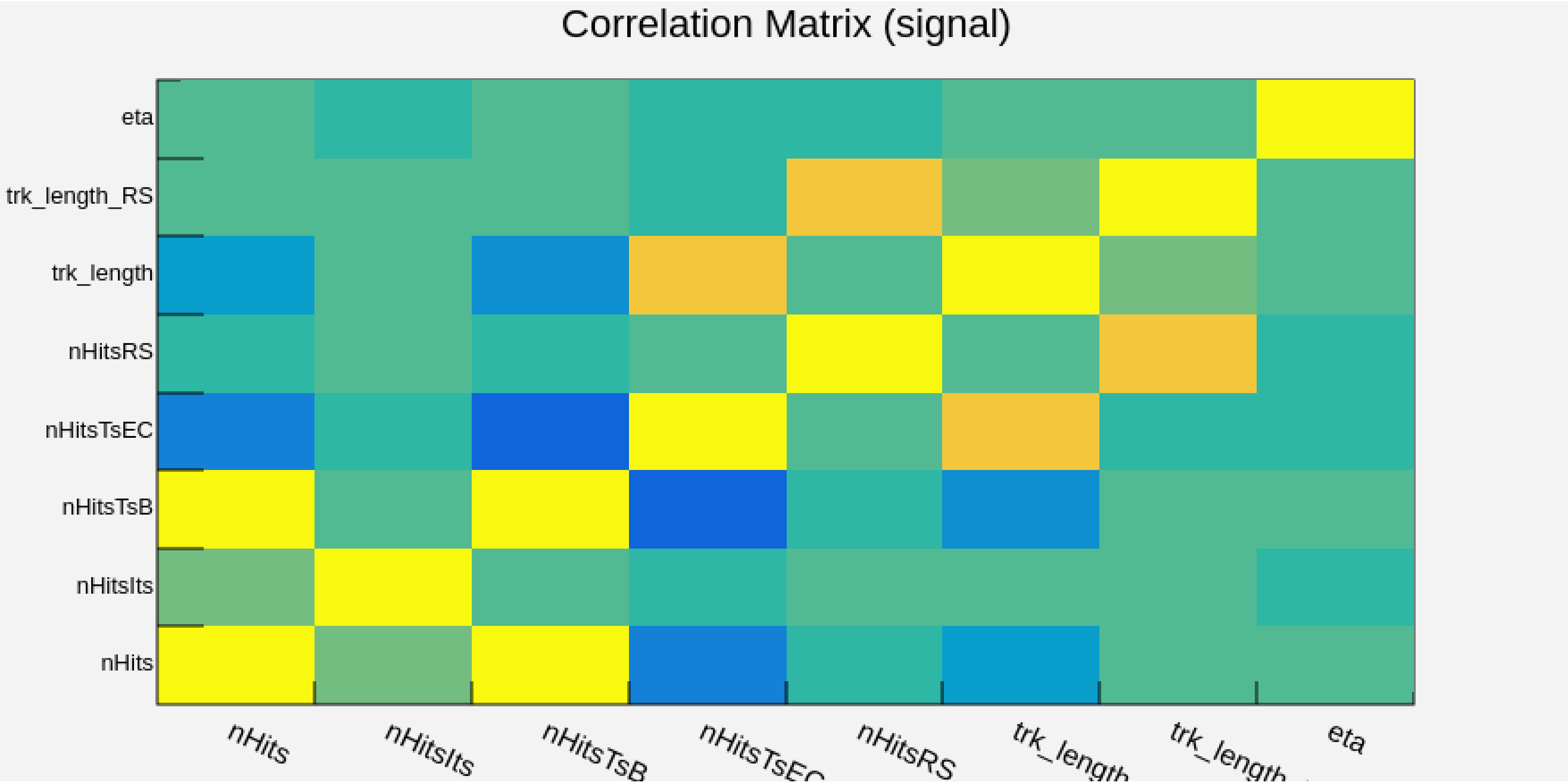
Track candidates: input variables – extrapolation to RS





- Track parameters in RS provide strong discrimination information. Very few pion tracks reach RS and have long extrapolated tracks.
-
- Extrapolation code is slow, so one of the tasks would be to speed it up… work is in progress on this

Track candidates: input variables



- Correlations between 8 new input variables. pT excluded.
- Yellow (blue) color correspond to positive (negative) correlations between variables

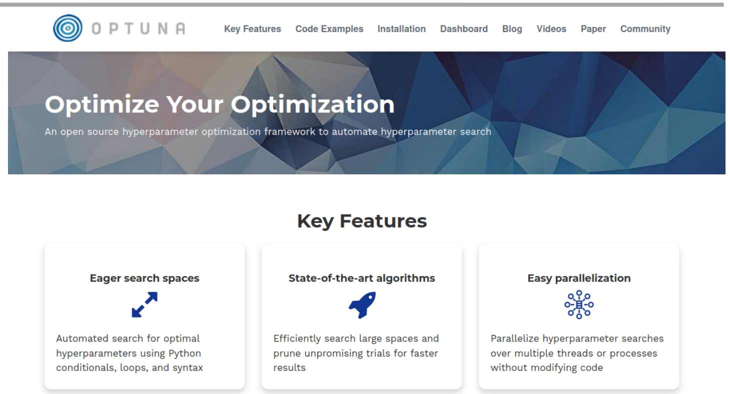
Problems in common NN APIs: hyperparameters optimization

- In addition to explicit parameters (synapse weights, neuron shifts) NN includes a lot of implicit parameters (number of layers, neurons in layers, activation functions, specific set of input variables, options for training algorithm, etc.) that are referred to as hyperparameters.

- Particular choice of hyperparameters can substantially affect the performance of the NN. Intuitive choices are often far from optimal.

- There are applications that allow optimization of hyperparameters (e.g., optuna). In practice they show poor performance, since hyperparameter space is essentially irregular.

- Moreover, memory leaks is a common problem for the hyperperameter optimization applications being applied to python based NN APIs.



A new evolutionary algorithm for optimizing the search of a rare Higgs boson production channel

Новый эволюционный алгоритм для оптимизации поиска редкого канала рождения бозона Хиггса

I. Boyko[a], A. Didenko[a,b,1], O. Dolovova[a], N. Huseynov[a,c], A. Tropina[a,d], I. Yeletskikh[a]

И.Р. Бойко[a], Н.А. Гусейнов[a,c], А.Р. Диденко[a,b,1], О.А. Доловова[a], И.В. Елецких[a], А.Д. Тропина[a,d]

[a] Joint Institute for Nuclear Research, Dubna, Russia

[b] Lomonosov Moscow State University, Moscow, Russia

[c] Institute of Physics Ministry of Science and Education Republic of Azerbaijan, Baku, Azerbaijan

[d] Moscow Institute of Physics and Technology, Dolgoprudny, Russia

В данной работе описываются результаты применения эволюционного алгоритма для оптимизации гиперпараметров нейронной сети (НС), решающей задачу разделения редкого процесса рождения бозона Хиггса в ассоциации с одиночным топ-кварком $pp \to tH(H \to bb)$ от основных фоновых процессов $pp \to tt, ttH, tZbq$.

This paper describes the results of applying an evolutionary algorithm to optimize the hyperparameters of a neural network (NN) solving the problem of separating the rare Higgs boson birth process in association with a single top quark $pp \to tH(H \to bb)$ from the main background processes $pp \to tt, ttH, tZbq$.

PACS: 07.05.Mh; 14.65.Ha; 14.80.Cp

Evolutionary algorithm for NN training

- Most of this problems can be (partially) solved. However, a lot of custom code required.
- Some of the problems (like feeding non-differentialble loss to gradient training algorithm, optimization of hyperparameters and memory leaks) are problematic to address.

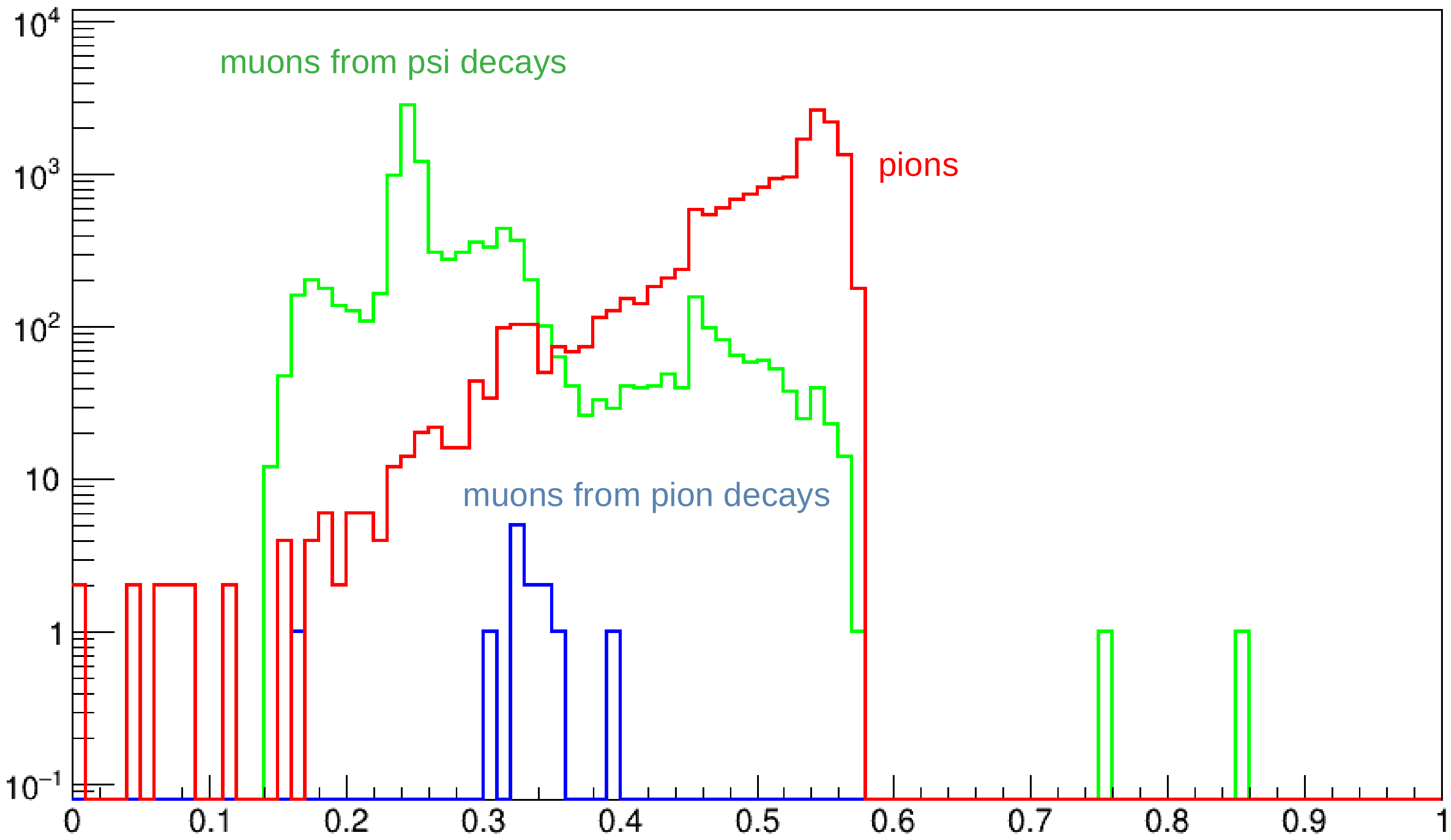
- One of the possible solutions is using custom NN API (c++ based in my case) that uses evolutionary algorithm for training.

- Non-differentiable functions are allowed, since no gradients are computed.
- Uncertainties of the input values can be included and reflected to the uncertainty of the NN input, thus automatically accounting for different ‘importance’ of different input events.
- Input neurons can be ‘switched off’ for those events where some of inputs are not defined.
- Overtraining is controlled by comparing ROC-AUC, significance or NN output distributions between training and testing samples.
- Hyperparameters can be optimized alongside explicit parameters, c++ code allows simple and transparent memory management.

Evolutionary algorithm for NN training

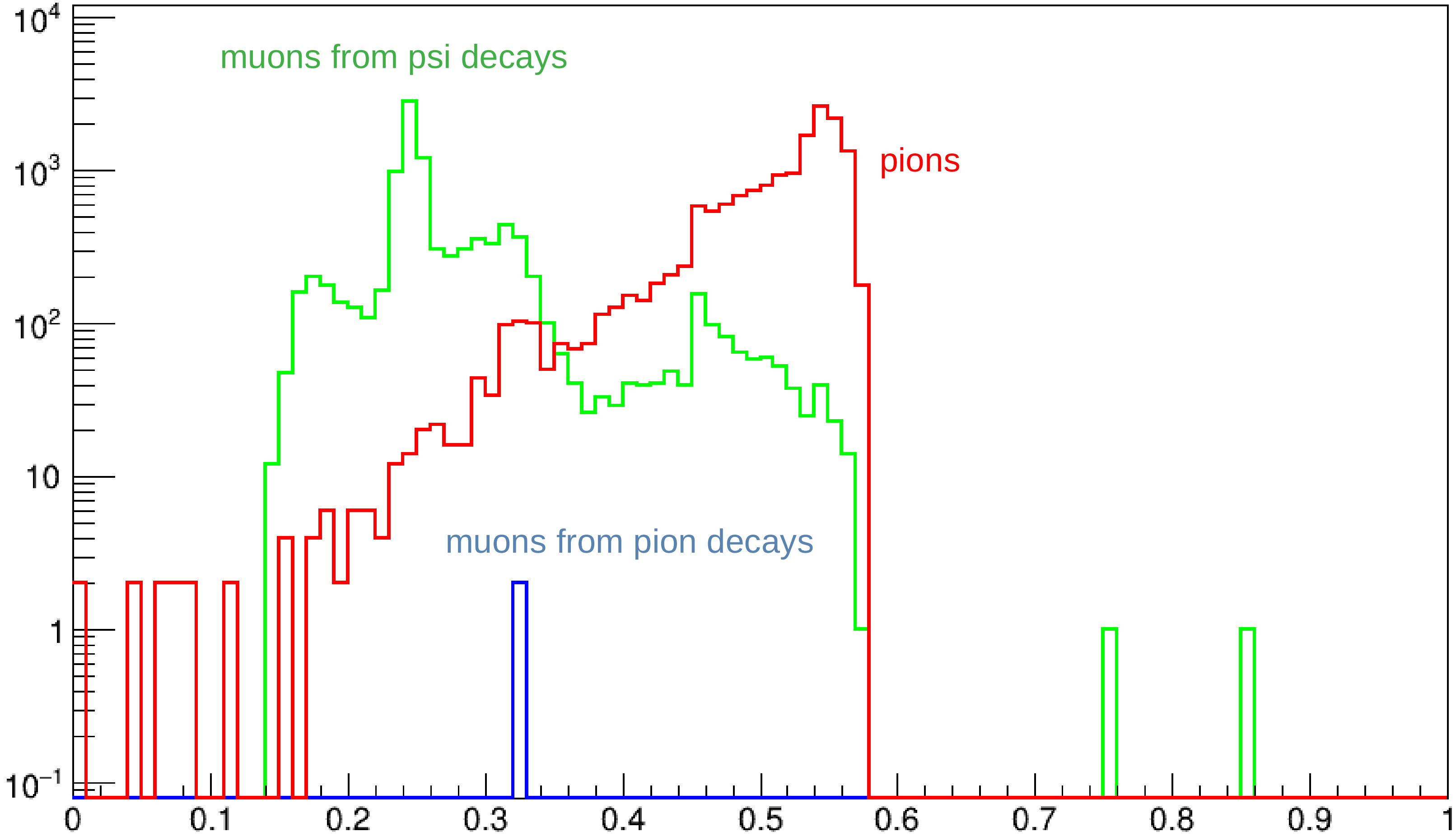
- This custom NN API is applied to the pion-muon identification task.
- Just 3 input variables are used for the test purpose (track length, track length in RS, number of hits in RS)
- NN implementation is simple, involving classes for neuron layers and synapse connection layers.
- Deep NN with 2 hidden layers (15, 9 neurons) is constructed, containing 189 synapse connections (~380 explicit parameters)
- Population of 50 neural networks is created
- At each training step (generation) the one or few best performing NNs give rise to their children with random mutations of the parameters applied
- Overtraining is controlled by the difference between ROC-AUC for training sample and testing sample.

Track candidates: NN response to signal and background



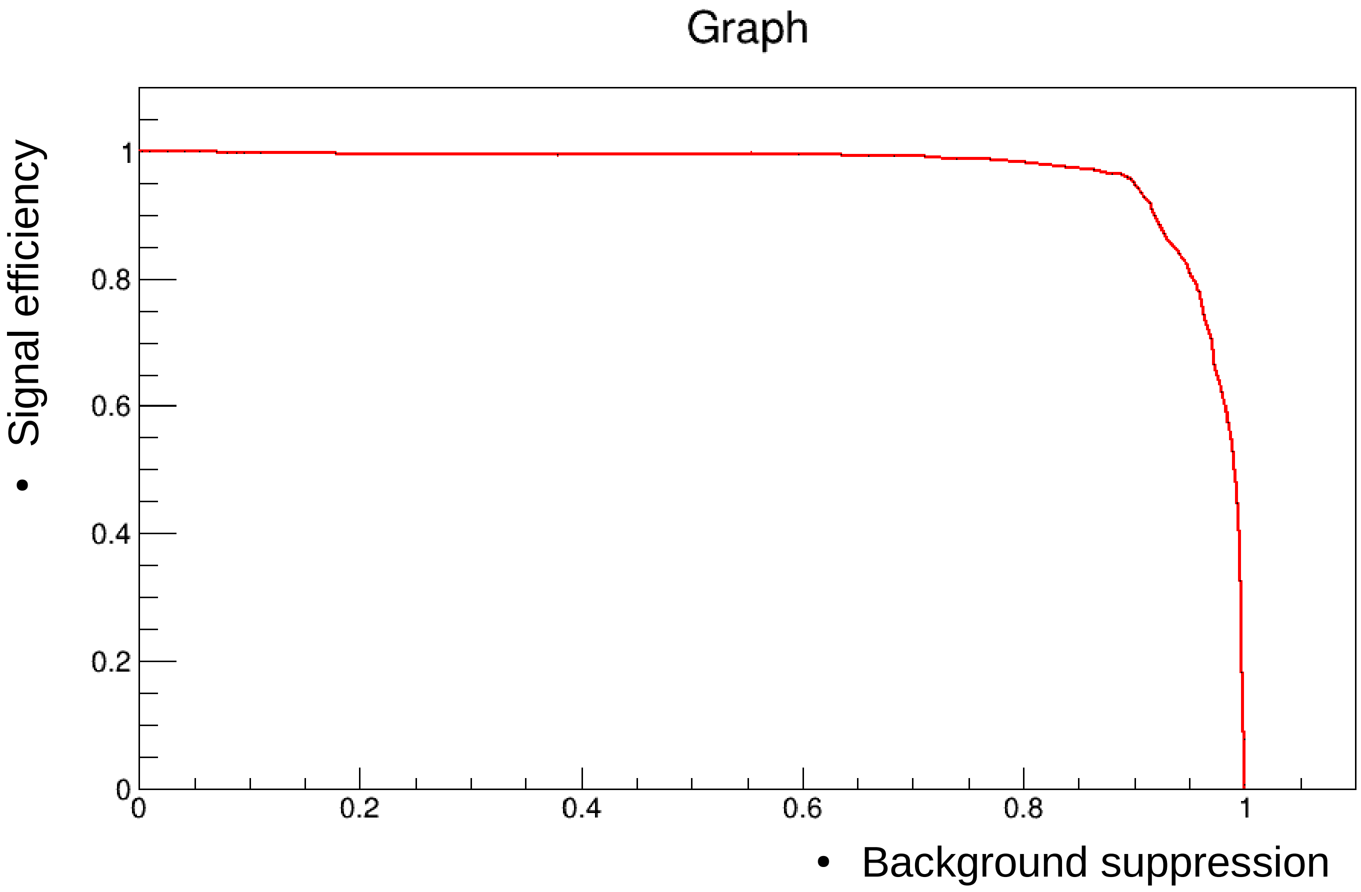
A small contribution from non-prompt muons coming from pion decays.
These muons are softer compared to prompt muons – most of them reside below 1.5GeV

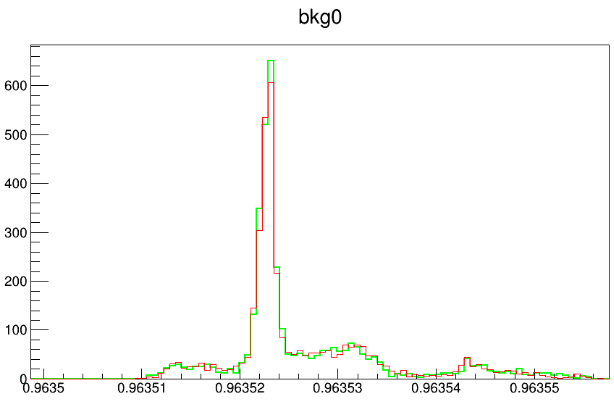
Track candidates: NN response to signal and background



Non-prompt muons coming from pion decays can be supressed using vertex information.
A transverse distance cut of 0.5 mm is applied for the track vertices.
Pion decays in RS were not properly reconstructed in MC. One may anticipate they also have poor consistency with PV candidates.

Track candidates: NN application. Muons and pions with 1.5GeV < pT < 2.5GeV



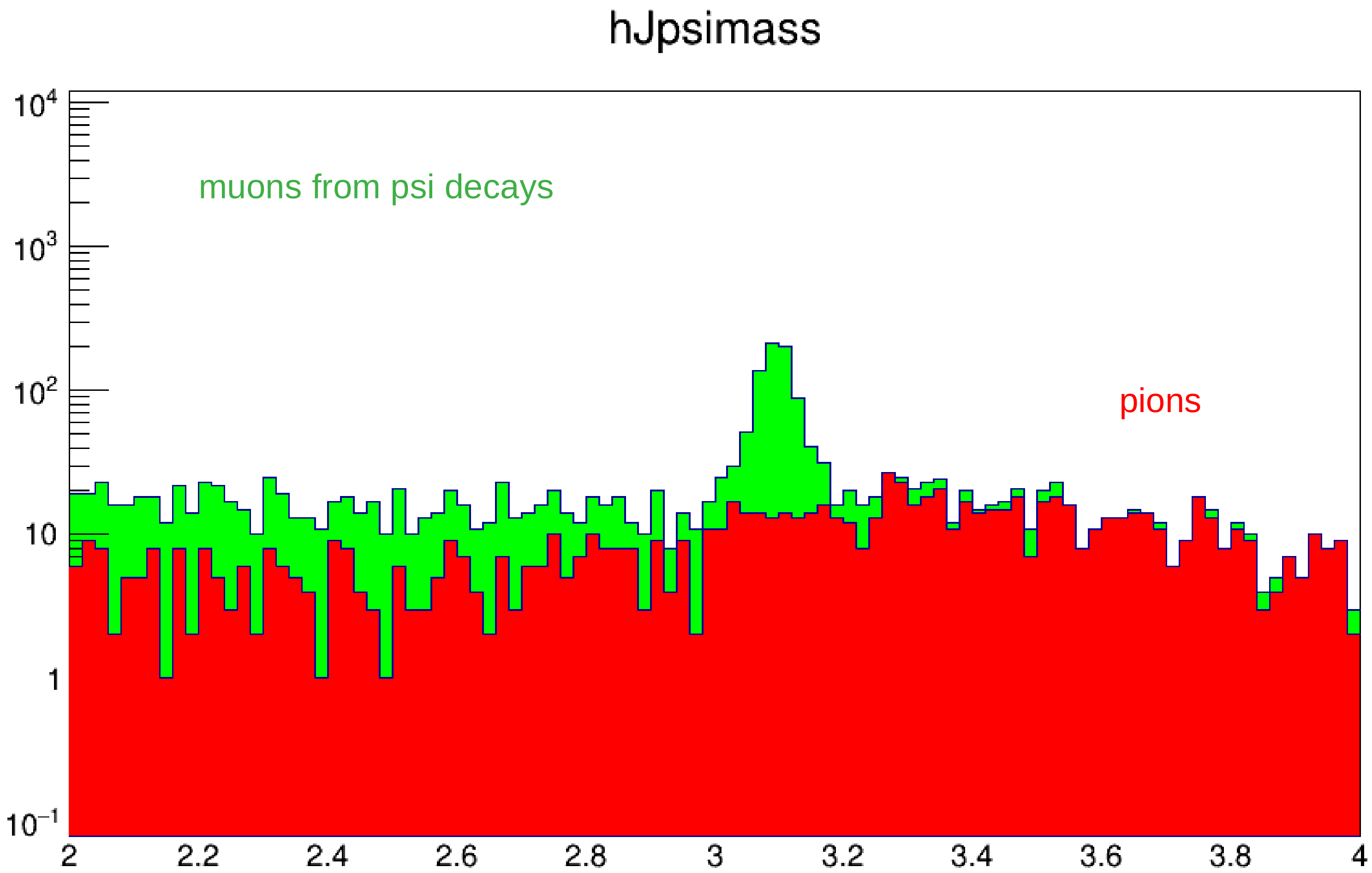


- Overtraining is controlled by comparing ROC-AUC for testing and training samples
- The observed difference is <0.4 permille
- Comparison of NN response to training and testing samples for signal and background are shown on the plots. No systematic deviations are seen.

Possible WP:

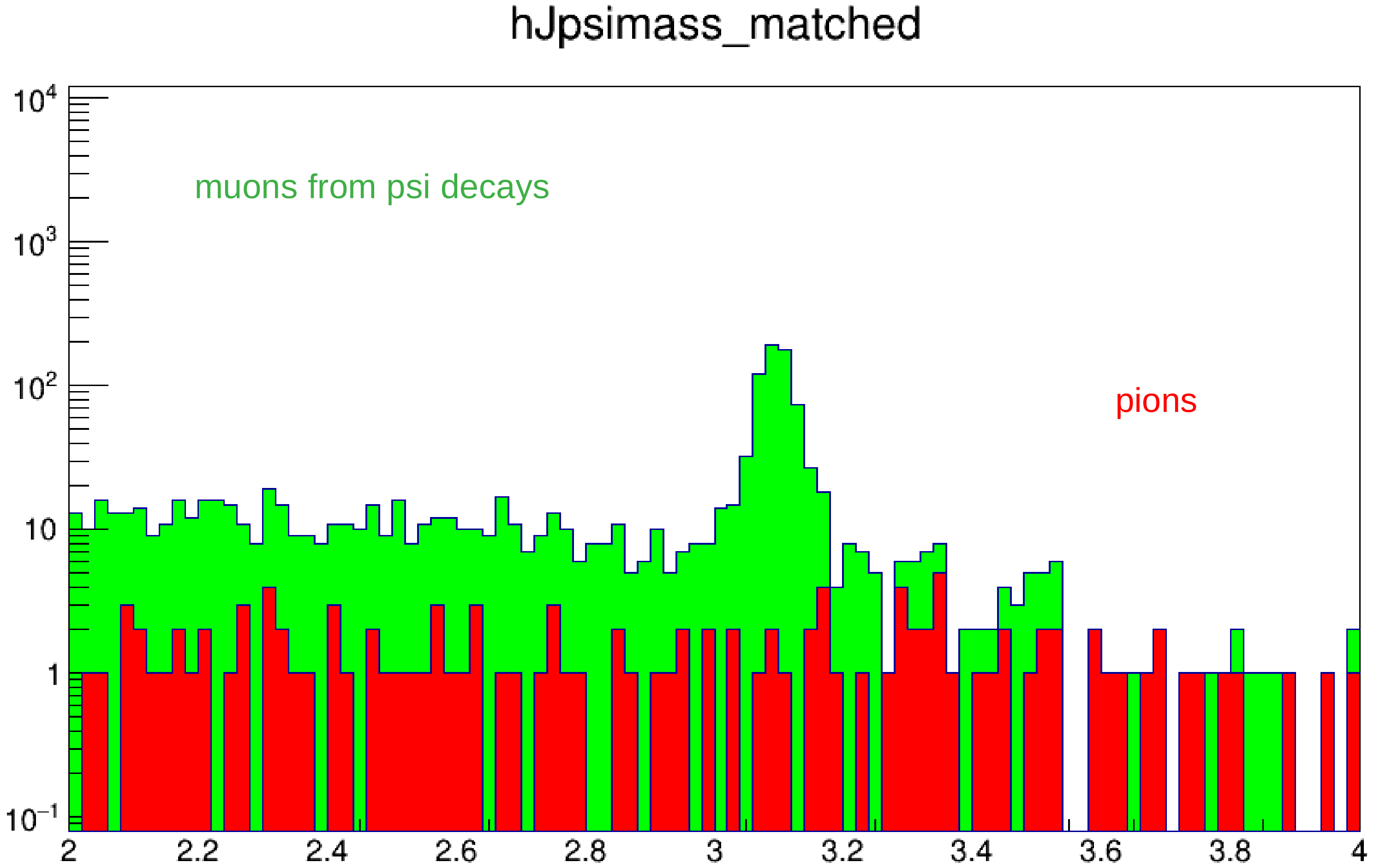
- Signal eff 9.902e-01 BG rejection 5.451e-01
- Signal eff 9.813e-01 BG rejection 6.407e-01
- Signal eff 6.938e-01 BG rejection 9.901e-01
- Signal eff 7.972e-01 BG rejection 9.802e-01

J/psi selection using NN response (1.5GeV < pT < 2.5GeV)



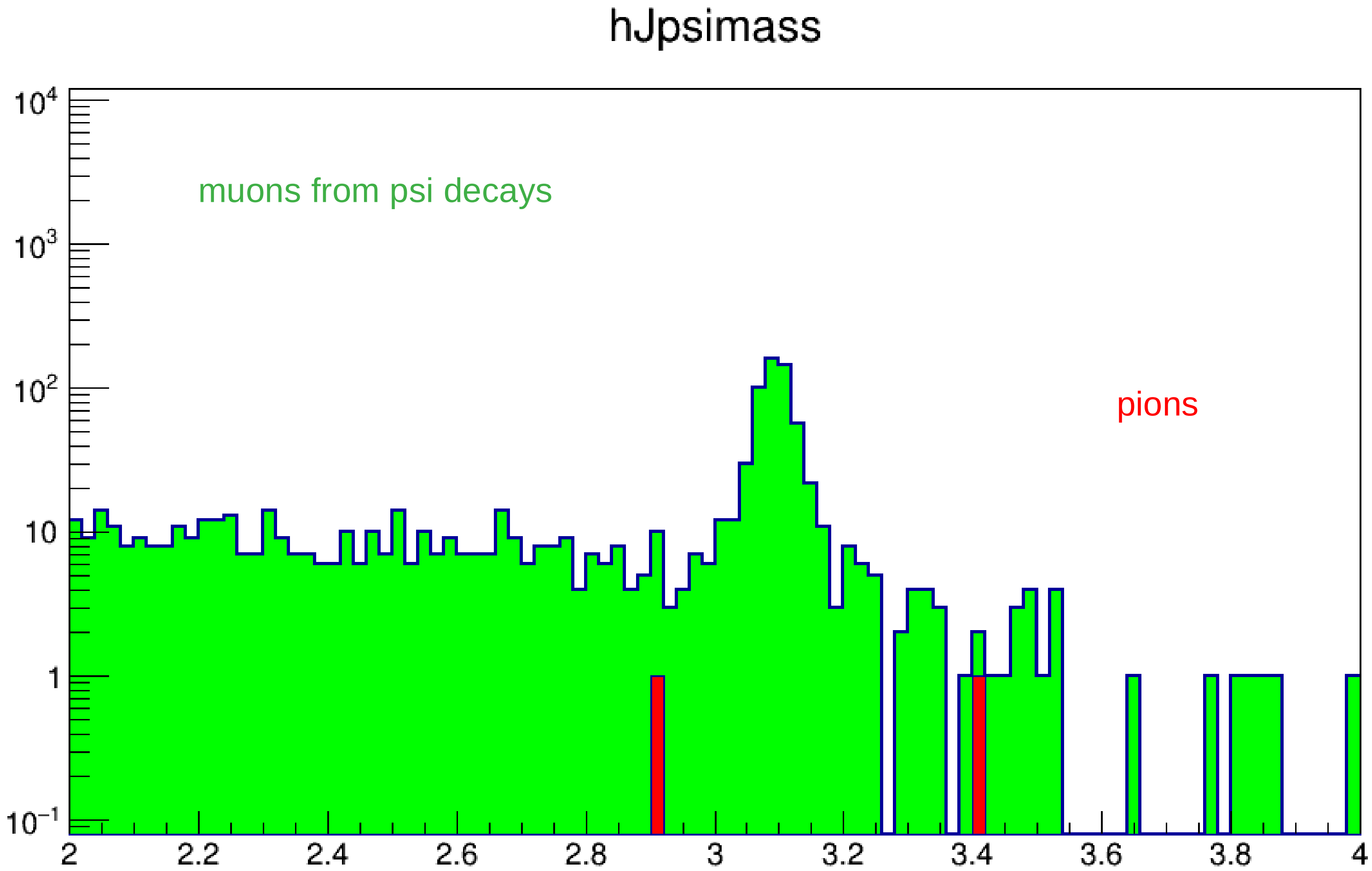
All muons in MC sample come from charmonia decays.
In absence of identification, pions present ~10% background under J/psi signal. To be higher in real data, in part., due to kaon and proton contribution and higher multiplicity in general.

J/psi selection using NN response (1.5GeV < pT < 2.5GeV)



All muons in MC sample come from charmonia decays.
After soft cut on the NN response (NN score < 0.5) that preserve >99% of muons the level of background is much lower

J/psi selection using NN response (1.5GeV < pT < 2.5GeV)



All muons in MC sample come from charmonia decays.
After tighter cut on the NN response (NN score < 0.35) that preserve ~98% of muons the background is extinct

Conclusions and Plans

- Custom NN API has been developed with its performance and available options are designed for detector data analysis

- Hits in subsistems, track length and simplest track kinematics allow some limited discrimination between muon and pion tracks.

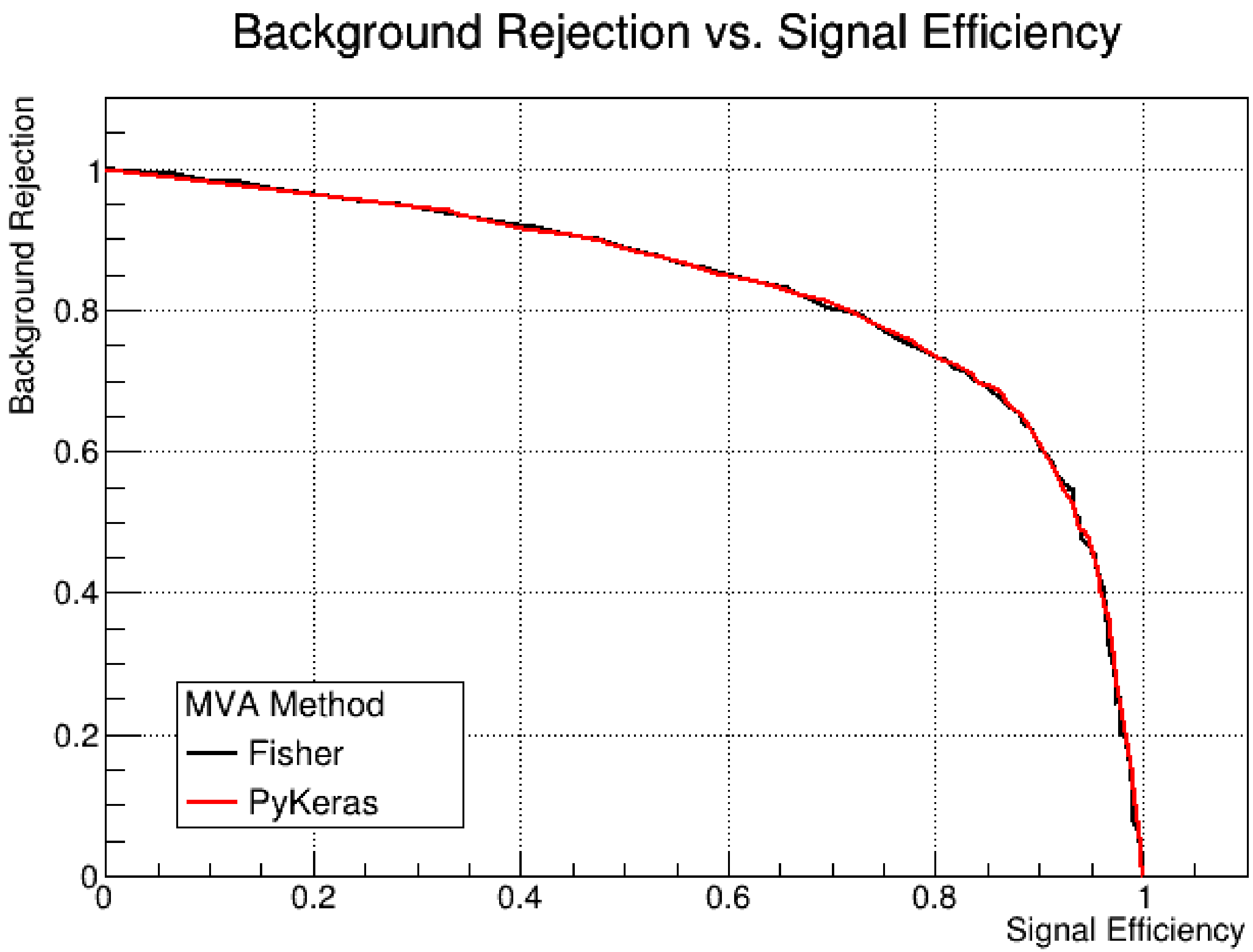
- Range System extrapolation allows much stronger discrimination.
- Extrapolation to RS is slow – work is ongoing on its optimization.
- Try to consider tracks separately in RS barrel and RS end-cap.
- Include proton and kaon tracks as background
-
- Contribution to muon sample from pion decays is small and can be further supressed by using vertex information. These muons contribute to the soft part of momentum spectrum.
- Dedicated MC is generated with artificially increased pion decay probability. NN to be trained to distinguish muons coming from these decays.
-
- It would be important to account for track parameters uncertainties. This is possible with our custom NN.
-
- NN results are tested on Jpsi sample and allow complete suppresion of background
- Vertex information and accounting for decay kinematics allow further discrimination between signal and background.

-

THANKS FOR ATTENTION

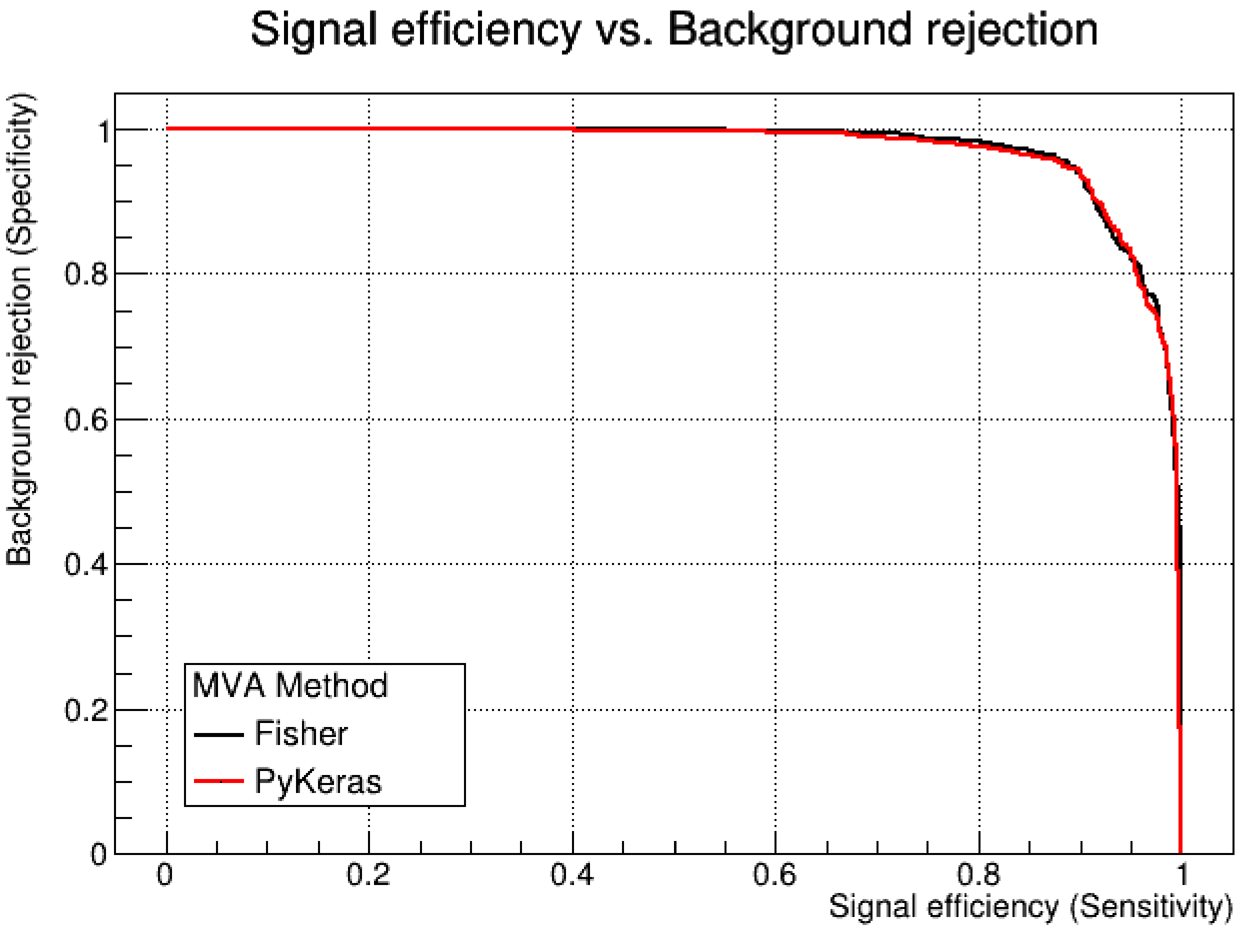
Backup slides

OLD results: Track candidates: NN application



- Total track momentum between 1.5 and 1.7 GeV
- AUC = 0.83
- Muon eff 0.9 correspond to pion rejection of 0.65

New result: Track candidates: NN application



- ~3K muon tracks and ~3K pion tracks are used for NN training+testing;
- Momentum range 1.5-2.1GeV is used;
- More statistics to be included after more MC events processed.

- Total track momentum between 1.5 and 2.1 GeV
- AUC = 0.97
- Muon eff 0.99 corresponds to pion rejection of ~0.6
- Pion rejection of 0.99 corresponds to muon efficiency of ~0.7