中国科学院高能物理研究所
Institute of High Energy Physics
Chinese Academy of Sciences

Accurate dE/dx simulation and prediction using ML method in the BESIII experiment

Tong Chen[a], Wenxing Fang[a], Xiaobin Ji[a], Weidong Li[a], Xiaoling Li[b], Tao Lin[a], Fang Liu[a], Jinfa Qiu[a], Shengsen Sun[a], Kai Zhu[a]

[a] Institute of High Energy Physics, Beijing 100049, People's Republic of China

[b] Institute of Frontier and Interdisciplinary Science, Shandong University, Qingdao, Shandong, China

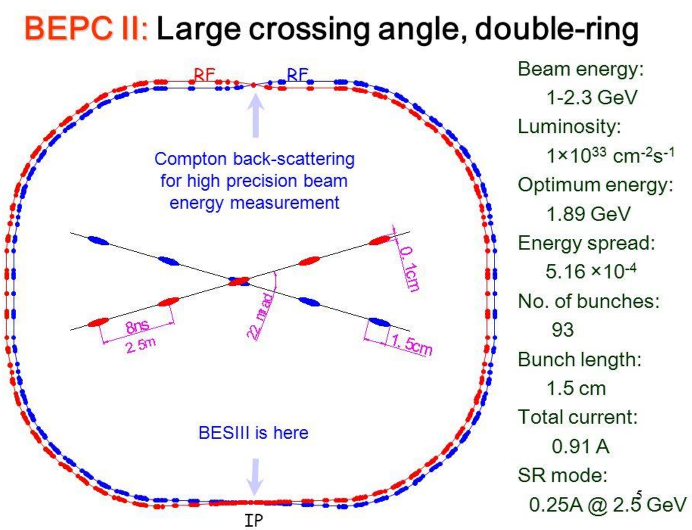
21th International Workshop on Advanced Computing and Analysis Techniques in Physics Research

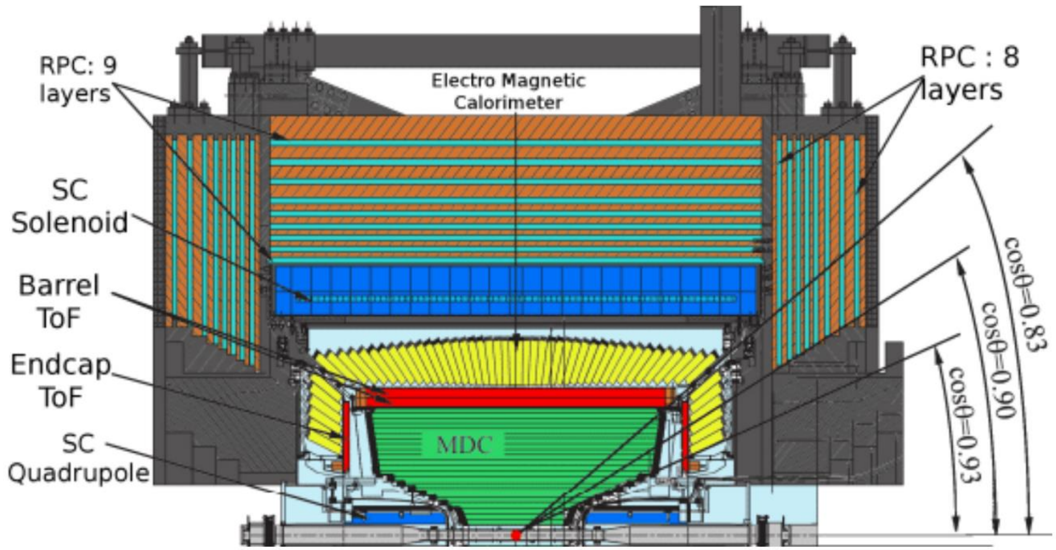
Outline

- Introduction:
  - The BESⅢ experiment
  - The dE/dx simulation, reconstruction, calibration in the BESⅢ
- dE/dx simulation with ML method
- dE/dx prediction with ML method
- Summary

The BEPCⅡ and the BESⅢ

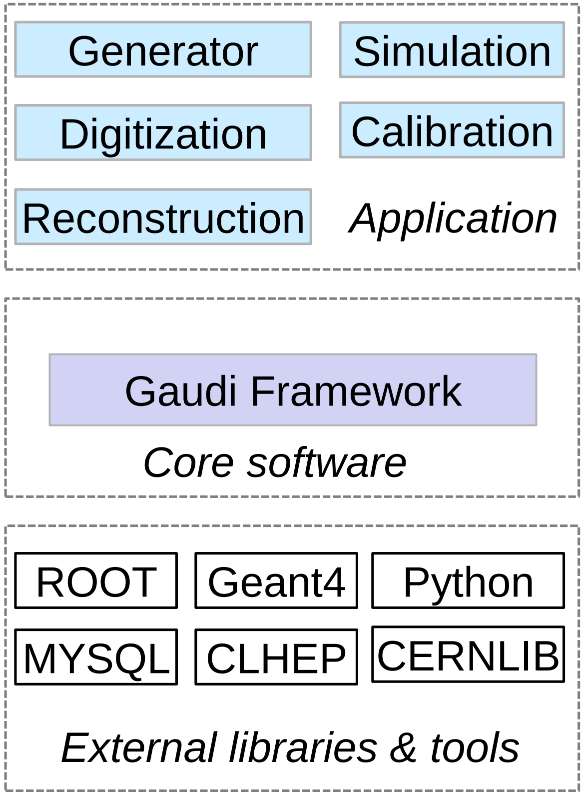
- The Beijing Electron Collider Ⅱ (BEPCⅡ) is a high luminosity e^+e^- collider with center mass-energy from 2 to 4.6 GeV
- The BESⅢ experiment at BEPCⅡ focuses on tau-charm physics. Such as non-perturbative QCD, exotic hadrons, BSM
- The BESⅢ has accumulated an unprecedented amount of dataset in this energy region. For example, 10B J/psi data





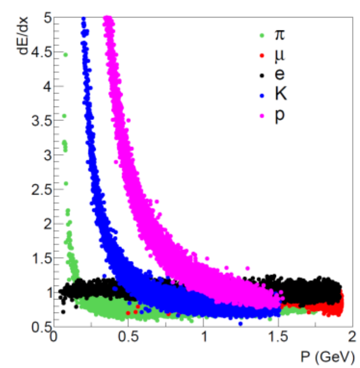
Offline software for the BESⅢ

- BOSS (BESⅢ offline software system) software structure
  - External libraries:
    - Geant4: detector simulation, particle propagation (decay) in the detector, interaction with detector material, …
    - ROOT, Python, …
  - Core software:
    - Gaudi framework: defines interfaces to all software components and controls their execution
  - Applications (BESⅢ-specific software):
    - Generator
    - Gean4 simulation
    - Digitization
    - Calibration
    - Reconstruction



PID in the BESⅢ

- For some analyses, the statistic uncertainty is small enough, and the systematic uncertainties become dominant. One of the most important systematic uncertainty is from particle identification (PID)
- The PID is essential for the BESⅢ experiment. Almost all analyses need it. It used to identify the particle to be one of it: $e, \mu, \pi, K, proton$
- For $\pi, K, proton$, the identification mainly relies on the dE/dx and the Time of flight (TOF)
- dE/dx: $\frac{dE}{dx} = D\frac{z^2}{m_e\beta^2}[\ln(\frac{2m_ec^2T_{max}}{I^2}\beta^2\gamma^2) - 2\beta^2 - \delta]$
- TOF: $v = \frac{L}{tof}, m = p\sqrt{\frac{1}{\beta^2} - 1}$



- This presentation will focus on the dE/dx, similar study can be done for the TOF

dE/dx simulation and reconstruction

Simulation

G4Step

↓

Particle's $\gamma\beta,\theta$,charge

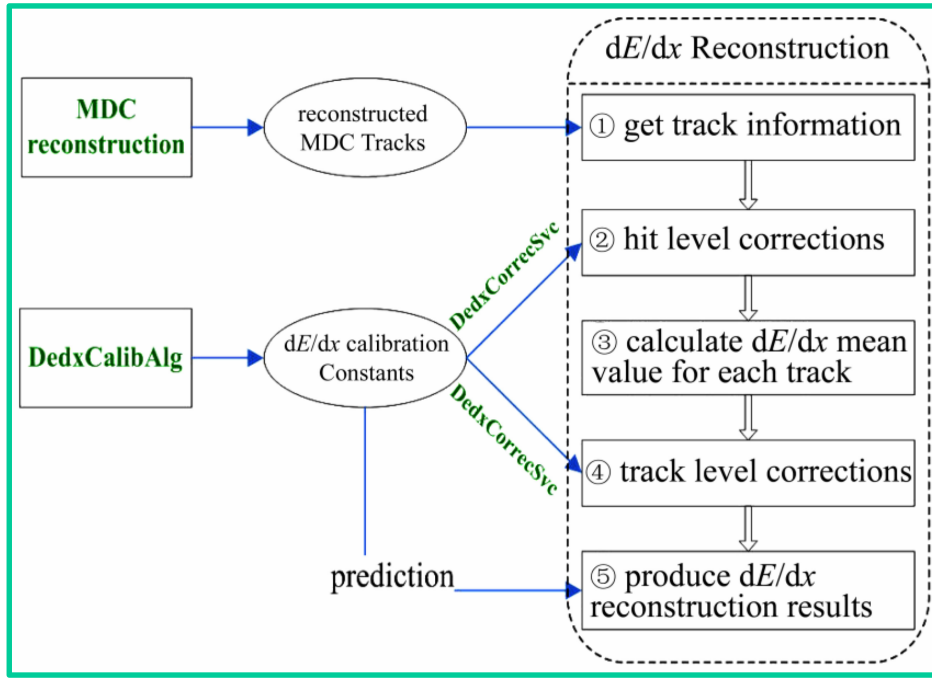
↓

Sampling dE/dx

↓

dE/dx distribution histograms in different bins ($\gamma\beta,\theta$,charge)

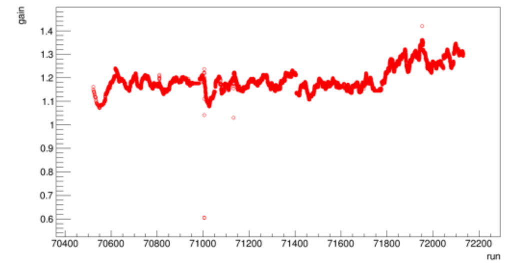
Reconstruction

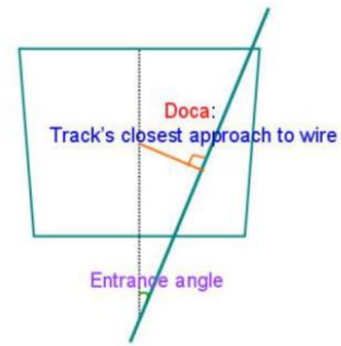


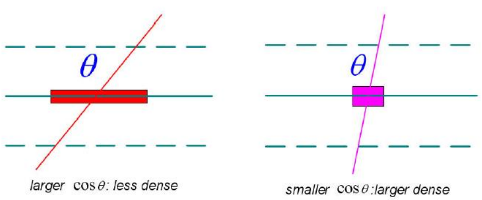
The corrections will be explained in next slide

dE/dx corrections

- To get unbiased dE/dx measurements
- Hit level corrections:
  - Run by run: due to the changes in gas pressure and temperature
  - Wire by wire: different drift chamber cell size, geometry, high voltage of signal wire, the radius of the signal wire
  - Doca and entrance angle: different drift distance of ionized electron to signal wire, non-uniform electromagnetic field
- Track level corrections:
  - Space charge effect depends on cosθ and dE/dx itself. smaller cosθ or larger dE/dx will have a larger space charge effect

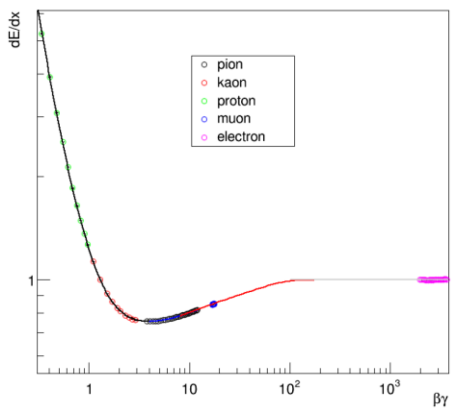


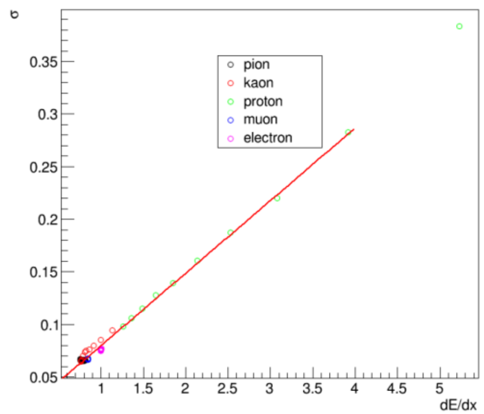


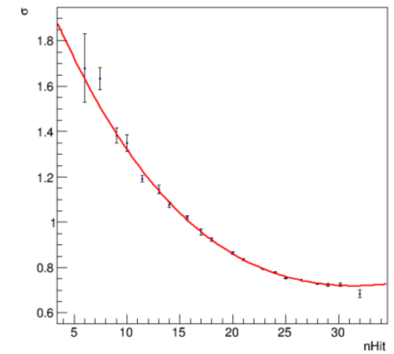


dE/dx calibration

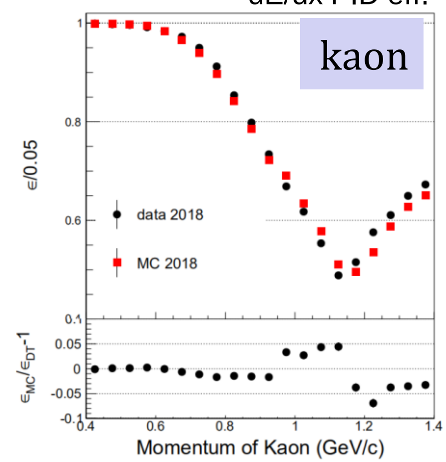
- Using reconstructed dE/dx of different particles, calibrating the expected dE/dx vs βγ, and the σ of dE/dx vs dE/dx (cosθ, nhit)







dE/dx PID eff.



- Using $\chi_{\mathrm{dE/dx}}$ for PID: $\chi_{\mathrm{dE/dx}} = \dfrac{|(\frac{\mathrm{dE}}{\mathrm{dx}})_{\mathrm{obs}} - (\frac{\mathrm{dE}}{\mathrm{dx}})_{\mathrm{exp}}|}{\sigma_{\mathrm{dE/dx}}}$

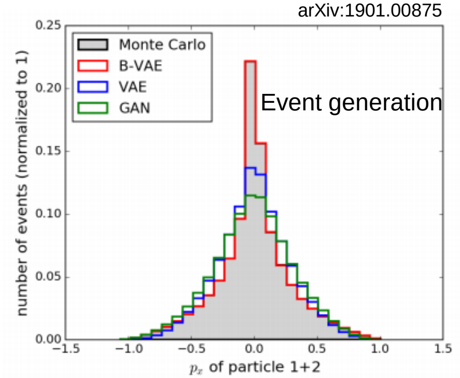
- This method has been smoothly working for many years, while there is still space for improving, especially for hadrons (π, K, proton)

- The key is to improve the simulation of dE/dx

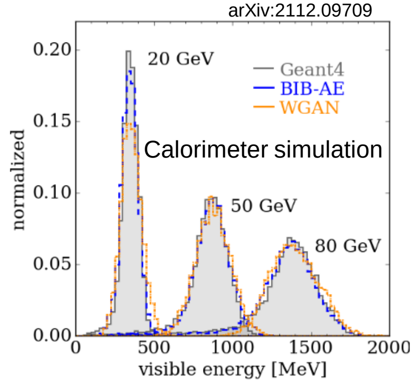
Simulation by machine learning

- Machine learning (ML) technology has the ability to learn the complex relationship between data. It is already widely used in HEP:
  - Jet tagging, particle identification, S/B separation, …
- Doing simulation (or generation) using ML is developing quickly. In HEP many studies are ongoing:

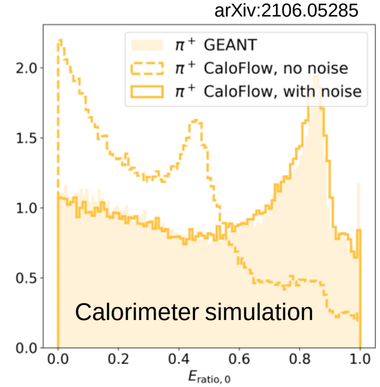
arXiv:1901.00875



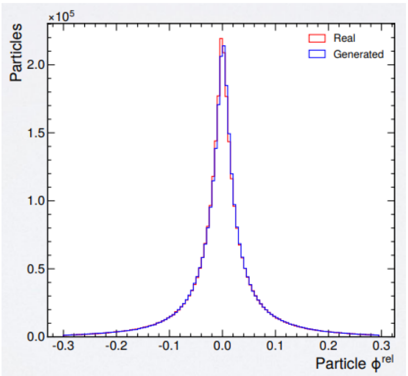
arXiv:2112.09709



arXiv:2106.05285



Jet simulation



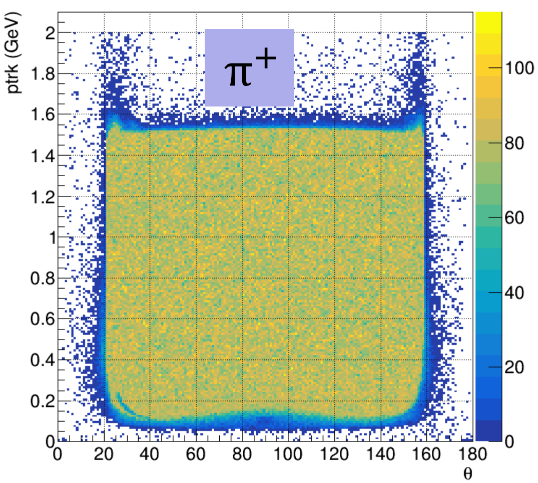
- Currently, BESIII owns massive real data and it is advantaged to utilize ML technology to do data-driven simulation at the BESIII

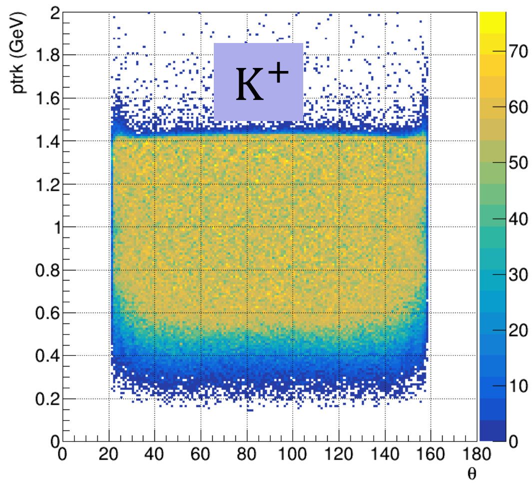
dE/dx simulation using ML

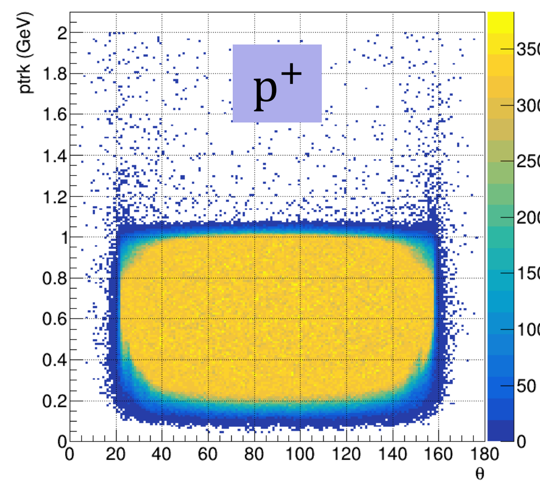
- Here we perform the dE/dx simulation using ML at track level, comparing to hit level simulation, it is easier and still precise enough
- 1, Learning the dE/dx distribution as function of momentum (p_{rec_track}), polar angle (θ_{rec_track}), number of hits ($nHit_{rec\_track}$) of reconstructed track from experiment data
  - Will be done by neutral network
- 2, Check the consistent of dE/dx distribution between data and simulation
- 3, Check the agreement of dE/dx PID efficiency between data and simulation

Dataset

- Dataset, 2018 J/ψ :
  - $\pi^{\pm}$: $J/\psi \longrightarrow \rho\pi \longrightarrow \pi\pi\pi$
  - $K^{\pm}$: $J/\psi \longrightarrow K_S^0 K^{\pm}\pi^{\mp} \longrightarrow K\pi\pi\pi$
  - $p^{\pm}$: $J/\psi \longrightarrow pp\pi\pi$
- The training data is smoothed in momentum and θ dimensions



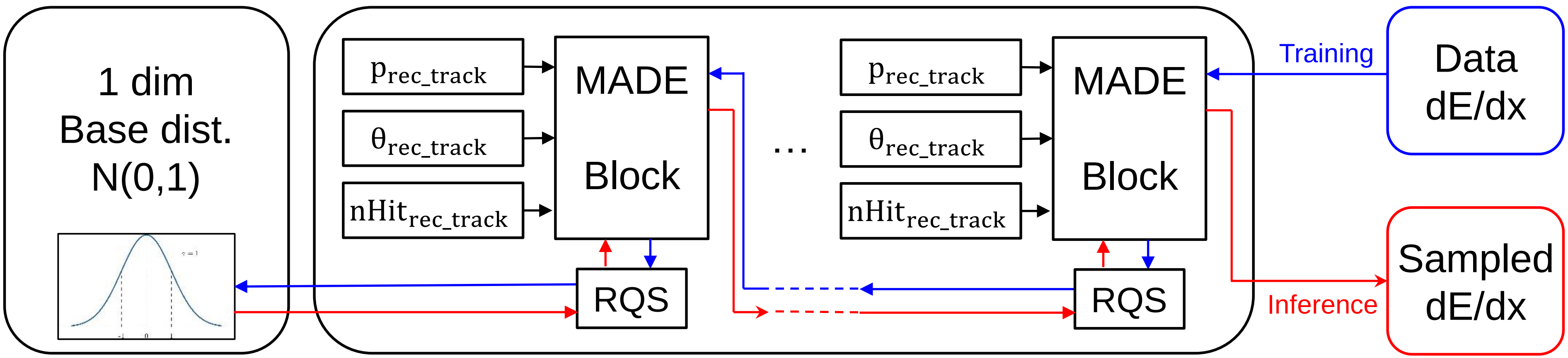




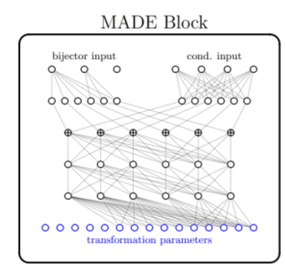
	π^+	π^-	K^+	K^-	p^+	p^-
Training data	1M	1M	0.5M	0.5M	2M	2M
Testing data	0.4M	0.4M	0.2M	0.2M	0.9M	0.9M

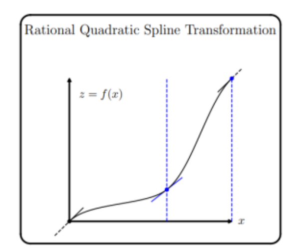
ML for dE/dx simulation

- The Normalizing Flow is adopted:
  - Learning bijective transformation between two distributions(e.g. dE/dx ~ N(0,1))
  - Comparing to GAN, it is much easier to training (stable and convergent)
  - Reference to the CaloFlow, a similar model is used, RQS (for transformation)+MADE block (for the parameters of RQS)

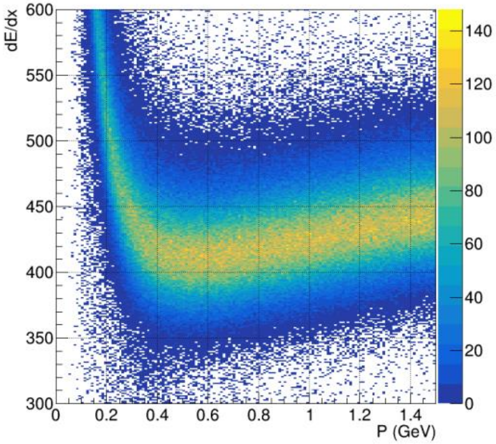


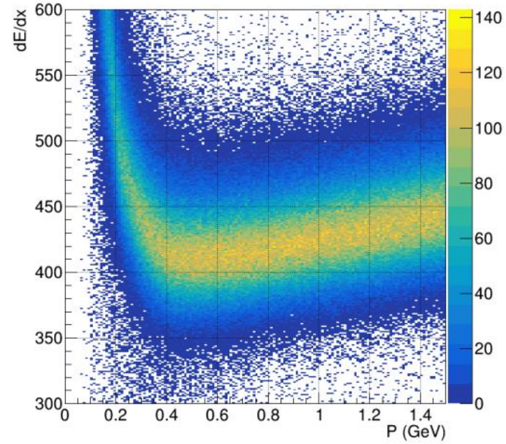
Base distribution	Number of MADE blocks	Layer sizes			Number of RQS bins
		input	hidden	output	
1-dim Standard Normal	6	64	3×64	23	8





Simulation performance (π^+)

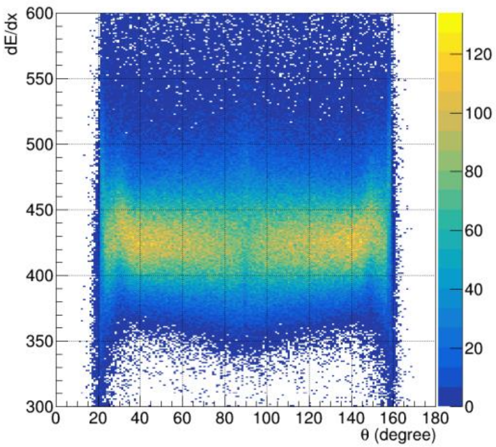


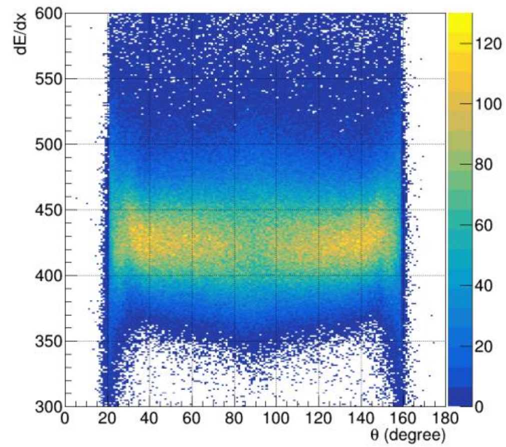


Data

NN

dE/dx vs P

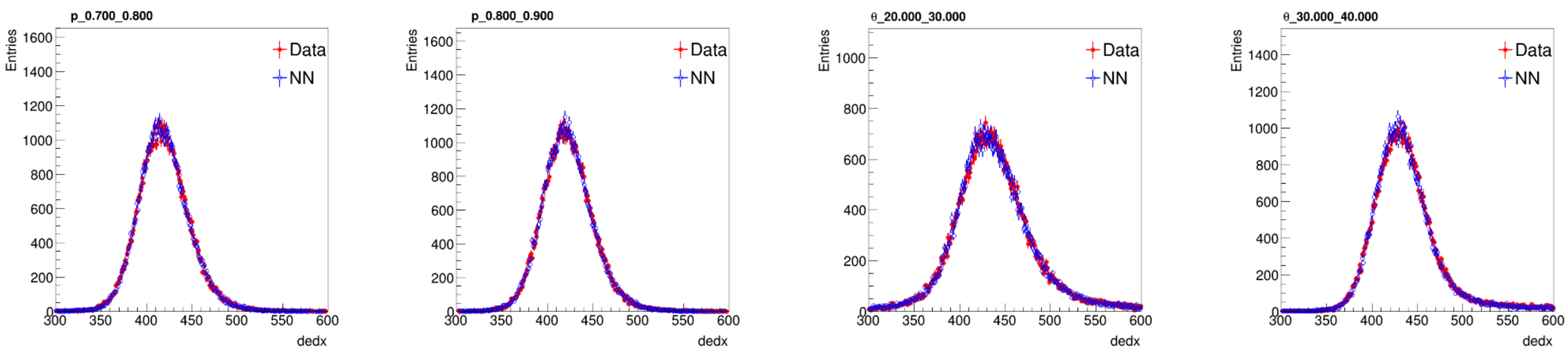




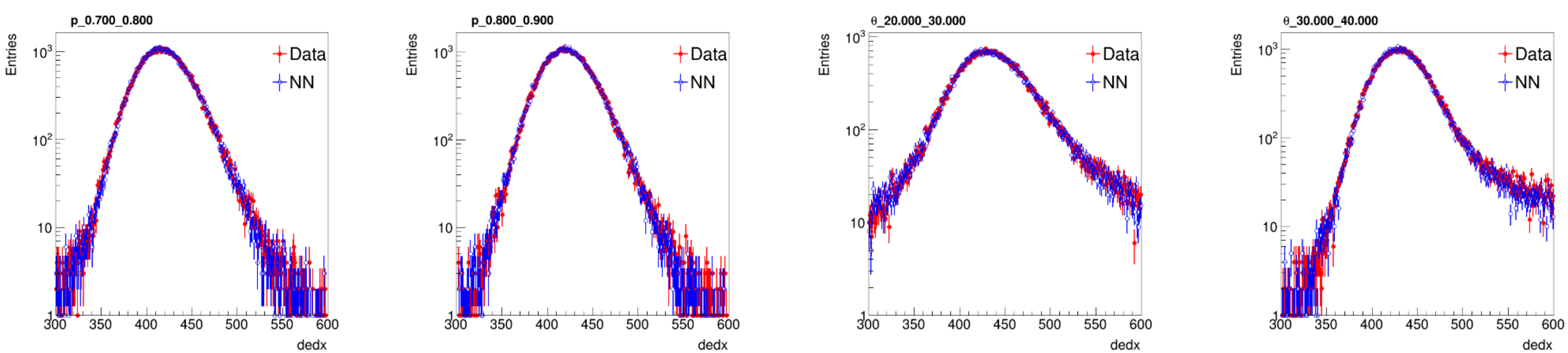
dE/dx vs θ

- Simulated dE/dx distribution is very similar to the data
- π^- in backup

Simulation performance (π^+)

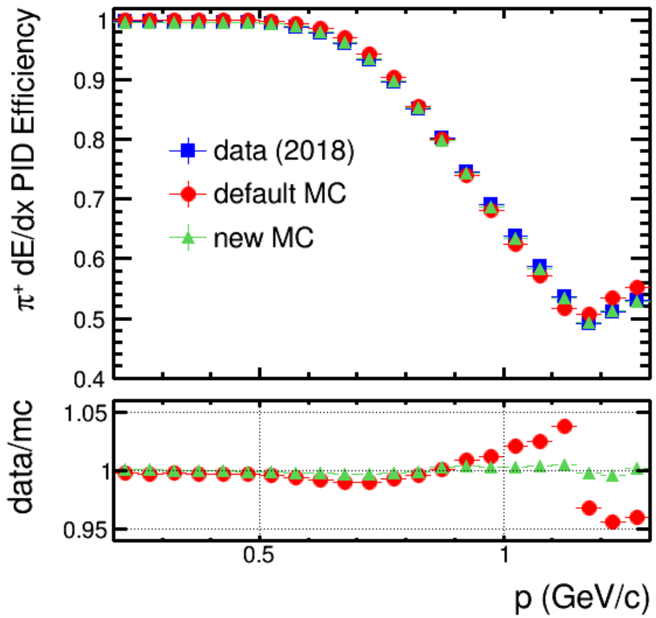


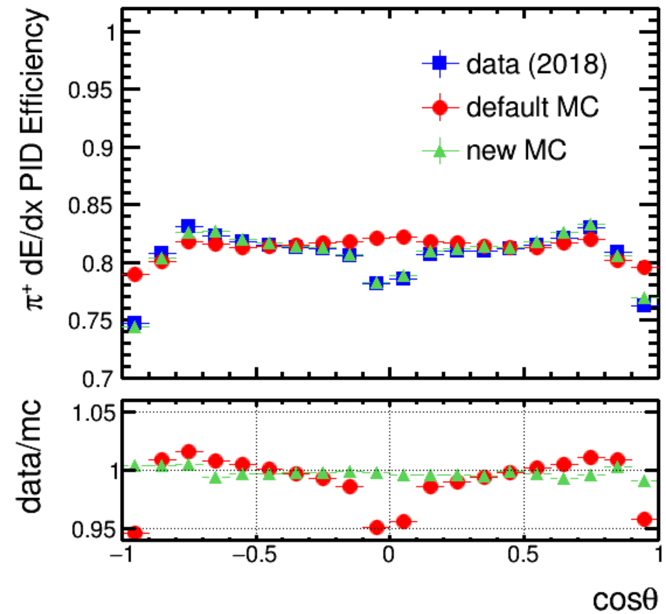
P:0.7-0.8 GeV | P:0.8-0.9 GeV | $\theta{:}20^{\circ}-30^{\circ}$ | $\theta{:}30^{\circ}-40^{\circ}$

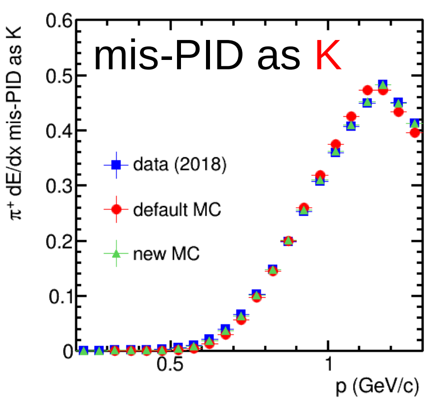


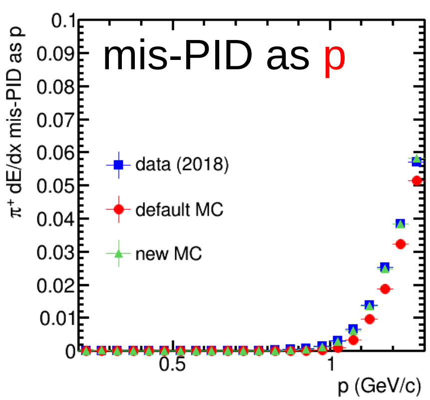
- Simulated dE/dx distribution is very similar to the data
- π^- in backup

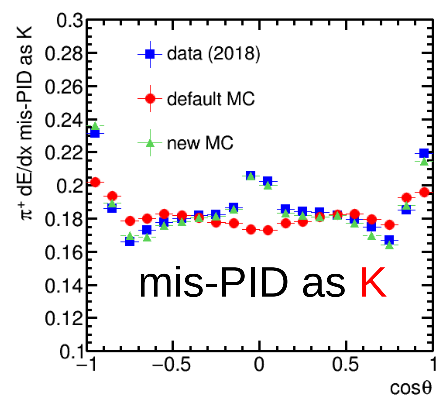
dE/dx PID performance (π^+)

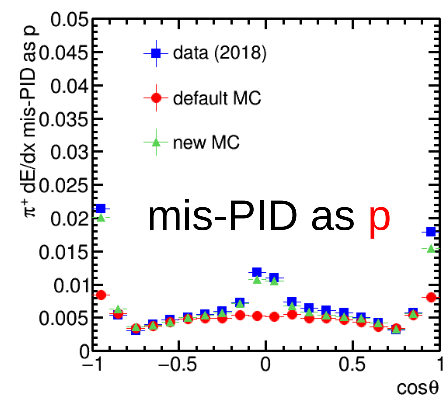






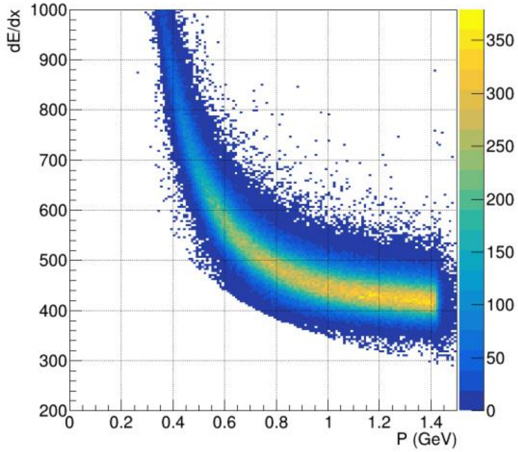


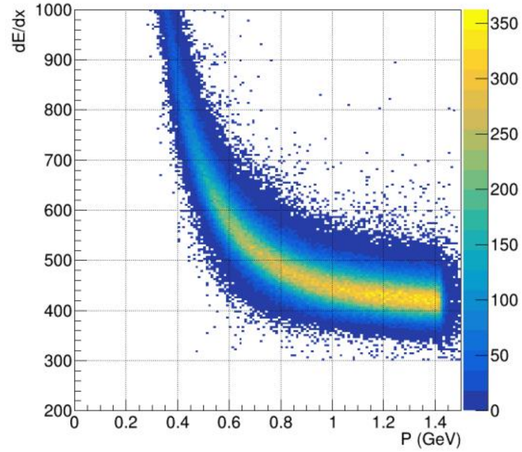




- new MC (from NN simulation) has better agreement with data

Simulation performance (K^+)



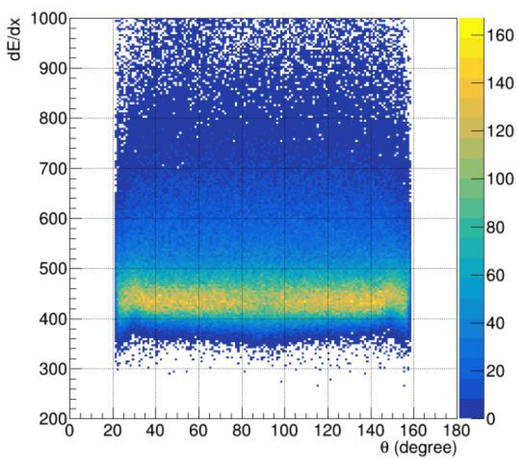


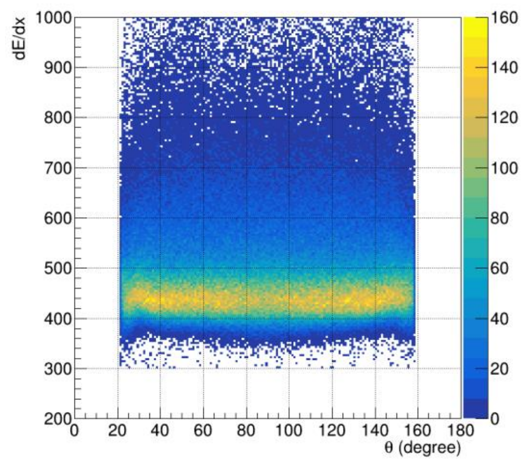
dE/dx vs P

Data

NN

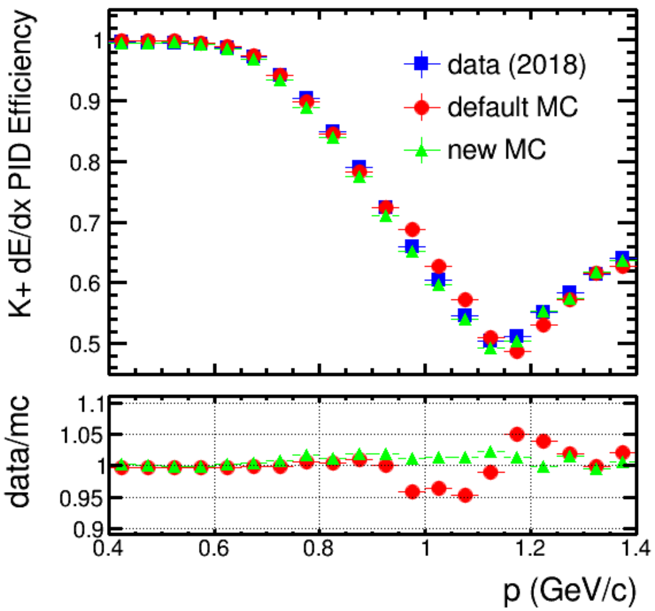
- Simulated dE/dx distribution is very similar to the data
- 1 dim. plots in backup
- K^- in backup

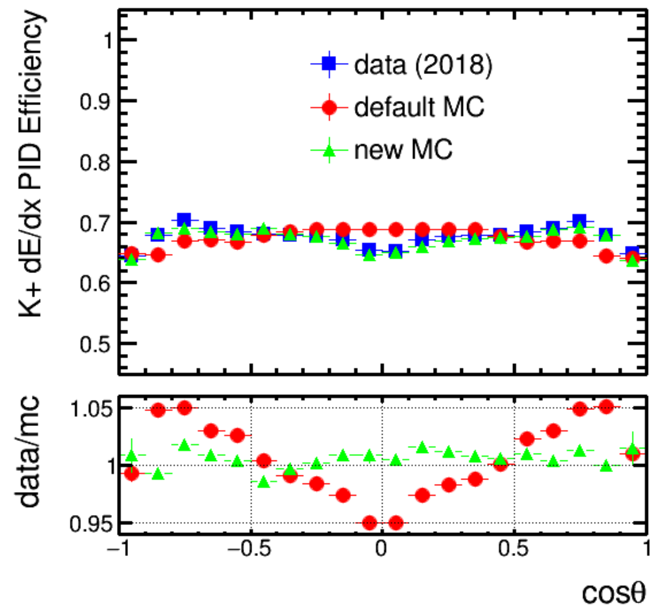




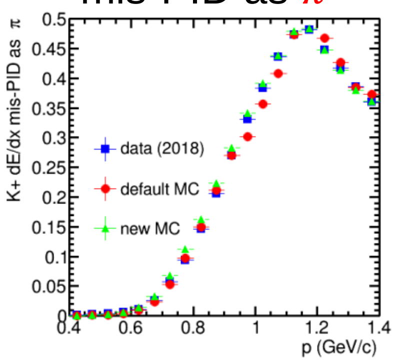
dE/dx vs θ

dE/dx PID performance (K^+)

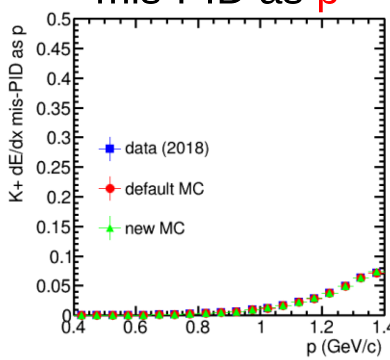




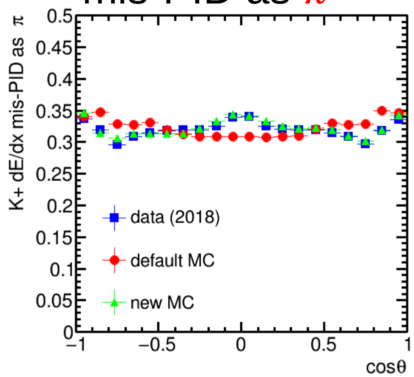
mis-PID as π



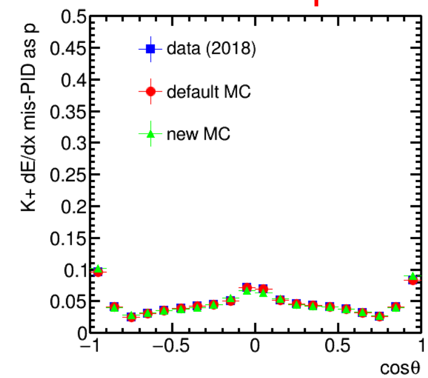
mis-PID as p



mis-PID as π

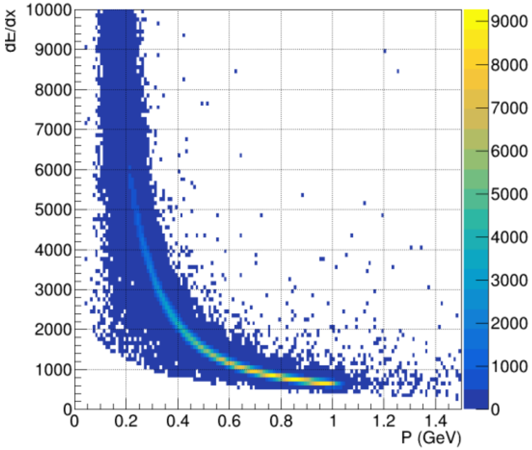


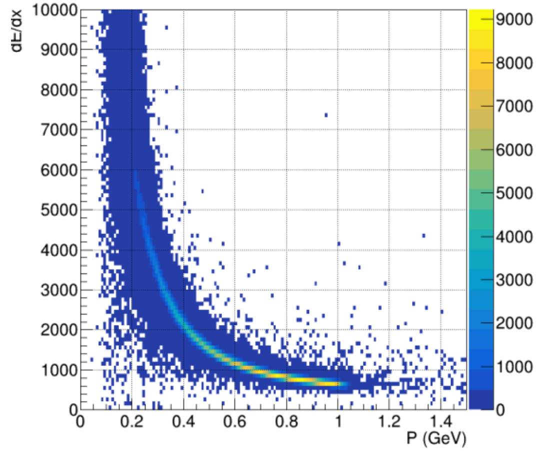
mis-PID as p



- new MC (from NN simulation) has better agreement with data

Simulation performance (p^+)

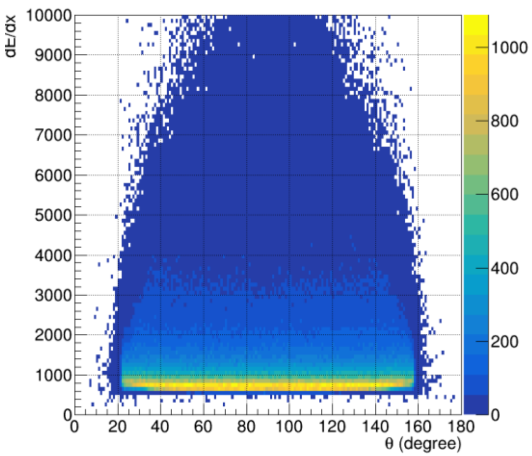


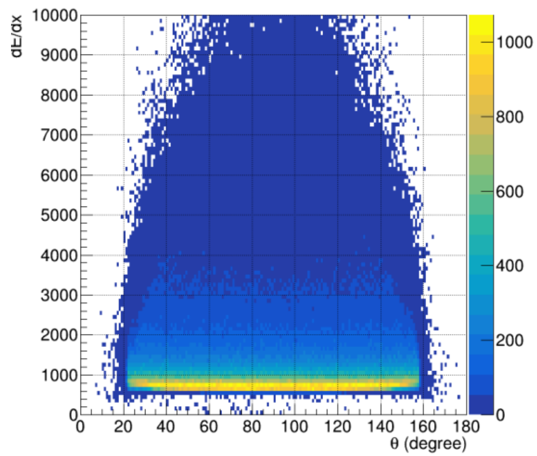


Data

NN

dE/dx vs P

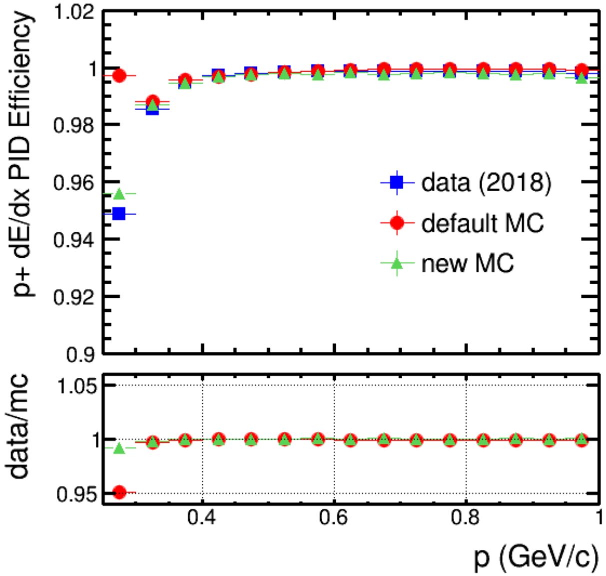


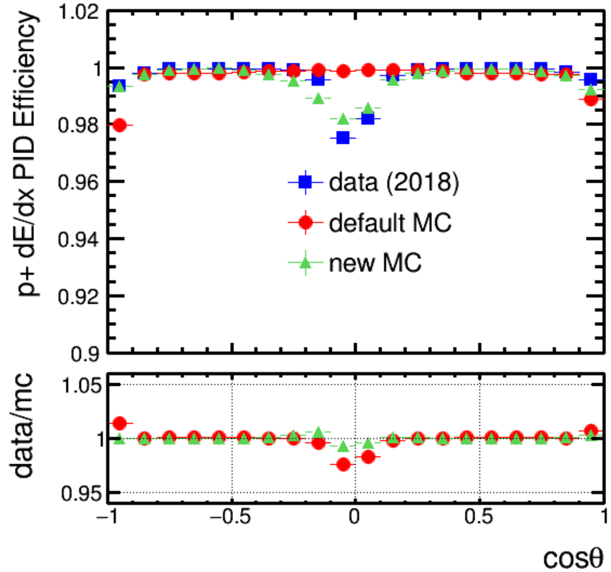


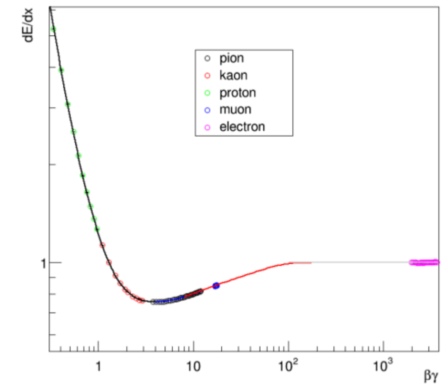
dE/dx vs θ

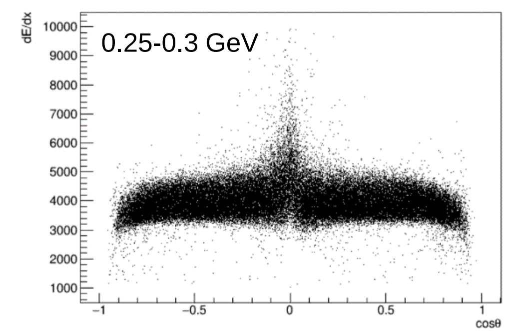
- Simulated dE/dx distribution is very similar to the data
- 1 dim. plots in backup
- p^- in backup

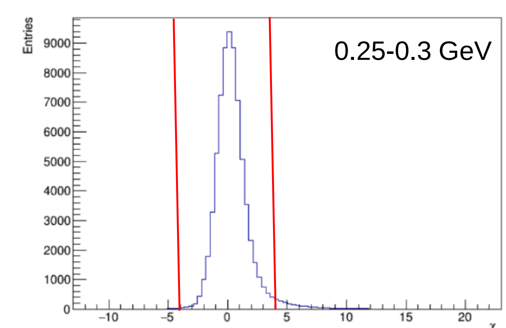
dE/dx PID performance (p^+)











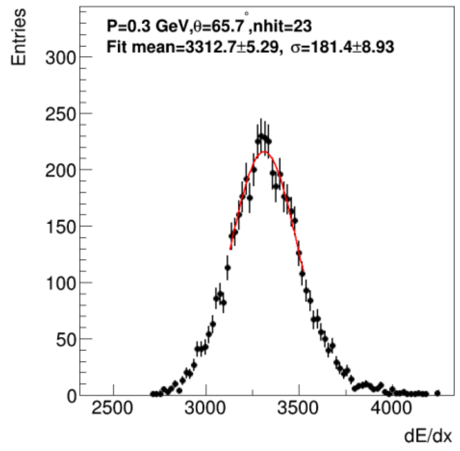
- For low momentum regions, the dE/dx changes sharply, and the space charge corrections do not work very well (specially for |cosθ| close to 0), making part of samples have large | $\chi_{dE/dx}$ | (>4), which causes the loss of dE/dx PID efficiency

dE/dx prediction

- To solve the problem in low momentum region, one should give the expected dE/dx and $\sigma_{dE/dx}$ according to the $\beta\gamma$, θ_{rec_track}, and $nHit_{rec\_track}$ of reconstructed track
- Traditional fitting method is not easy to simultaneously fit dE/dx (and $\sigma_{dE/dx}$) with $\beta\gamma$, θ_{rec_track}, and $nHit_{rec\_track}$
- For deep learning, it is typically a regression problem:
  - Predicting expected dE/dx and $\sigma_{dE/dx}$ according to $\beta\gamma$, θ_{rec_track}, and $nHit_{rec\_track}$

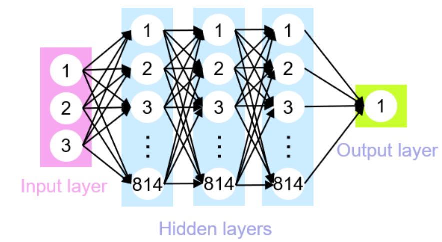
ML for dE/dx prediction

- Dataset:
  - By using the previous trained dE/dx simulation flow, the dE/dx distribution can be generated for different p_{rec_track}, θ_{rec_track}, and $nHit_{rec\_track}$
  - Then fitting the generated dE/dx distribution with Gaussian to obtain the expected dE/dx and $\sigma_{dE/dx}$



- Model: fully connected neural network

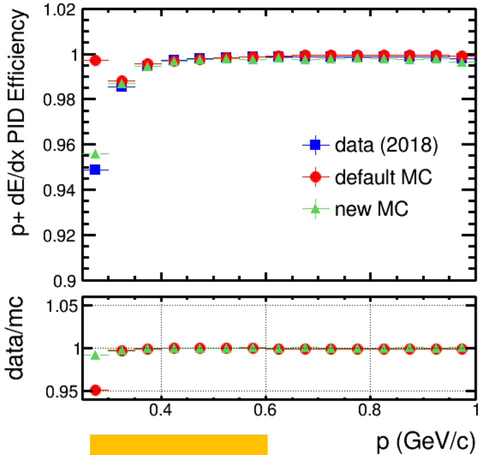
Model	Input data	Layer sizes			Output data
		input	hidden	output	
Predict expected dE/dx	p_{rec_track} θ_{rec_track} $nHit_{rec\_track}$	3	3× 814	1	Expected dE/dx
Predict $\sigma_{dE/dx}$	p_{rec_track} θ_{rec_track} $nHit_{rec\_track}$	3	3× 814	1	$\sigma_{dE/dx}$



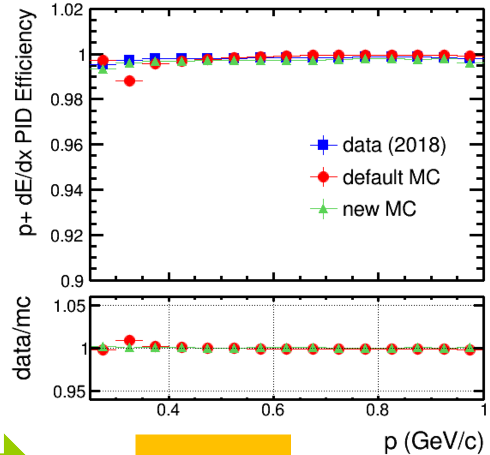
optimizer: Adam

Loss: L1Loss

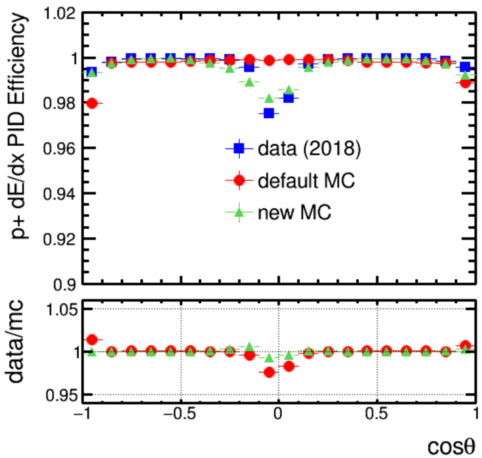
dE/dx PID performance for p^+

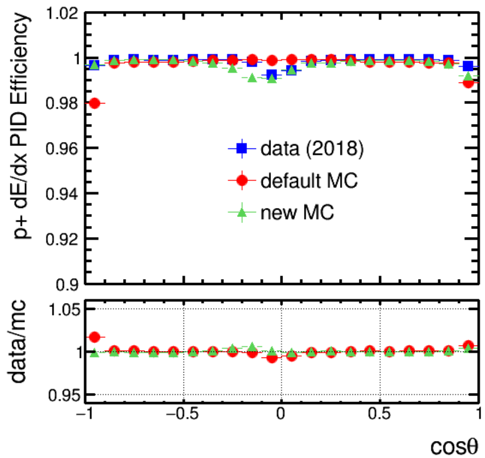


before



after





dE/dx PID eff. vs P

After using the expected dE/dx and $\sigma_{\mathrm{dE/dx}}$ from NN, the PID efficiency in data and new MC is recover to >99% in low momentum region

dE/dx PID eff. vs θ

Summary

- This talk presented the dE/dx simulation based on Normalizing Flow by using experiment data sample
- The simulated dE/dx has very high fidelity, and the final systematic uncertainty of dE/dx PID is reduced to ~1% in overall
- The prediction of expected dE/dx and $\sigma_{\mathrm{dE/dx}}$ using NN is developed which aims to solve the lost of dE/dx PID efficiency for proton(anti) at low momentum region
- Using the expected dE/dx and $\sigma_{\mathrm{dE/dx}}$ from NN, the dE/dx PID efficiency for proton(anti) at low momentum region can be recovered to ~100%

Thanks for your attention !

Back up

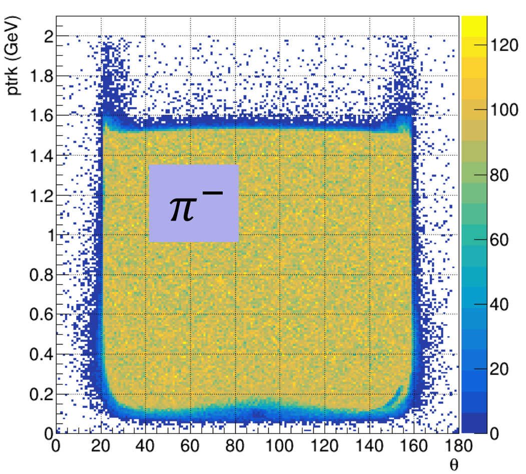
Summary and plan

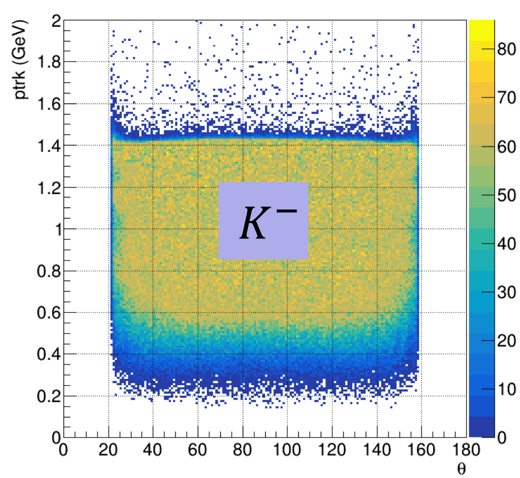
- The study of dE/dx simulation using deep learning technique is presented
- It is shown that the dE/dx distribution of real data can be learnt by neutral network
- The dE/dx PID efficiency agreements between data and simulation for $\pi^{\pm}$, $K^{\pm}$ and $p^{\pm}$ are improved by this method

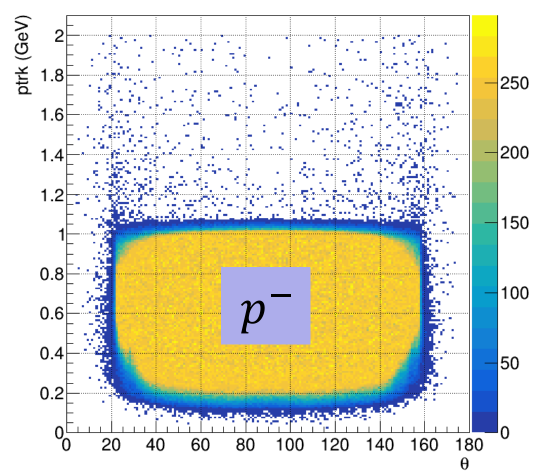
- Future plan:
  - Mis-indentificated efficiency (sample impure effect)
  - GNN, low Pt region, different physics channels, PID efficiency code public
  - Mean truncate -> ML method ?
  - Tuning the network to get better performance
  - Check the performance of this method for leptons (e and muon)
  - Perform more detailed dE/dx simulation which is at hit level

Dataset

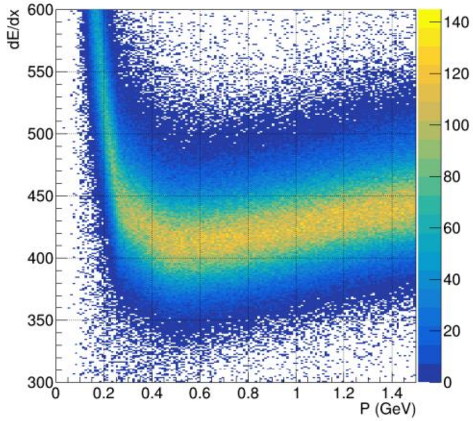
- Dataset, 2018 J/ψ :
  - $\pi^{\pm}$: $J/\psi \longrightarrow \rho\pi \longrightarrow \pi\pi\pi$
  - $K^{\pm}$: $J/\psi \longrightarrow K_S^0 K^{\pm}\pi^{\mp} \longrightarrow K\pi\pi\pi$
  - $p^{\pm}$: $J/\psi \longrightarrow pp\pi\pi$
- The training data is smoothed in momentum and θ dimensions

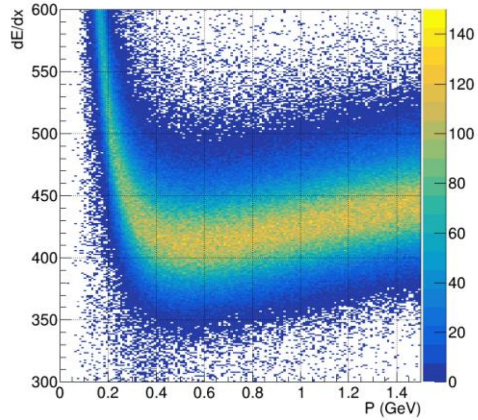






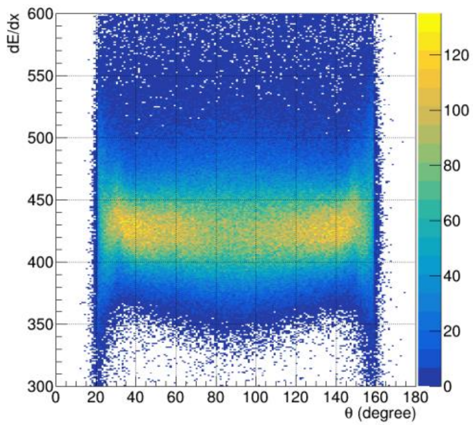
Simulation performance (pi-)

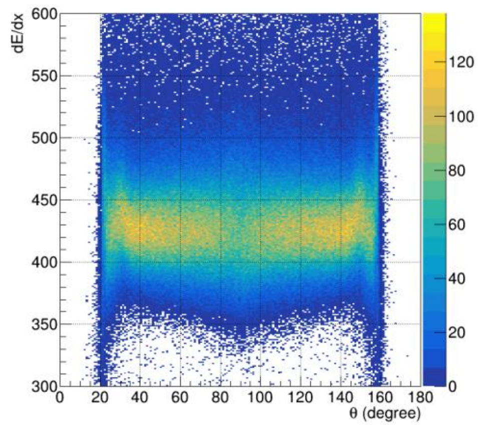




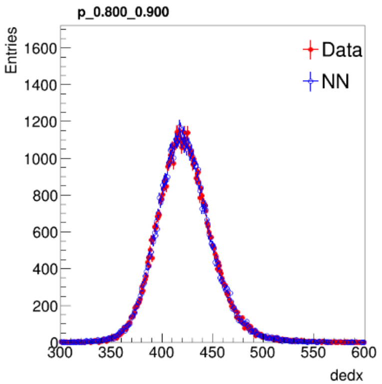
Pi- NN

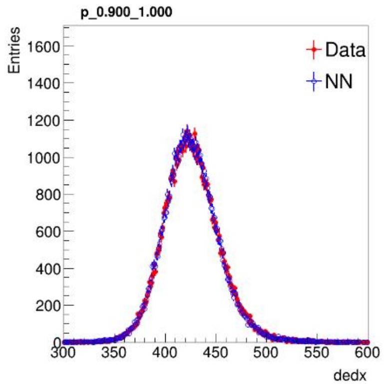
Pi- data

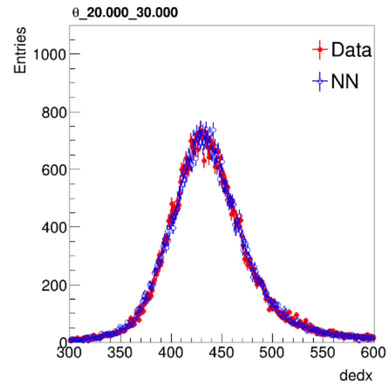


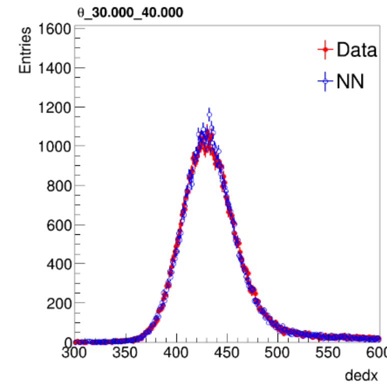


Simulation performance (pi-)







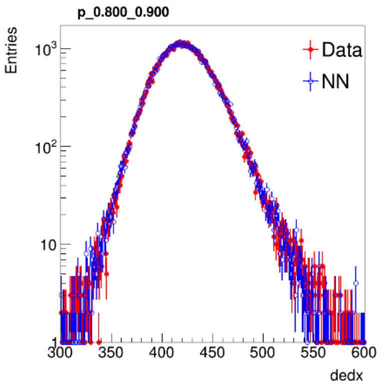


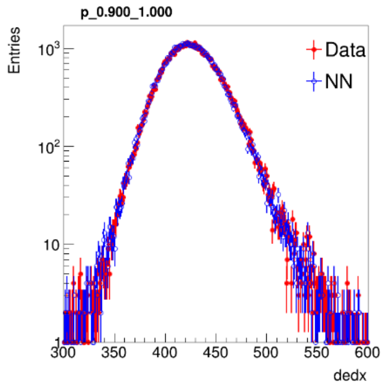
P:0.8-0.9 GeV

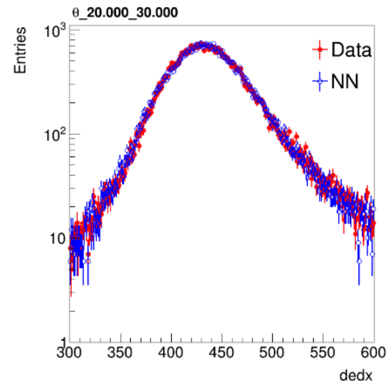
P:0.9-1.0 GeV

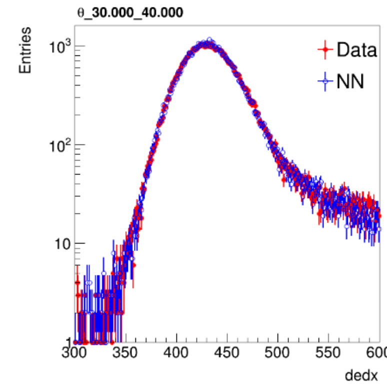
$\theta{:}20^{\circ} - 30^{\circ}$

$\theta{:}30^{\circ} - 40^{\circ}$

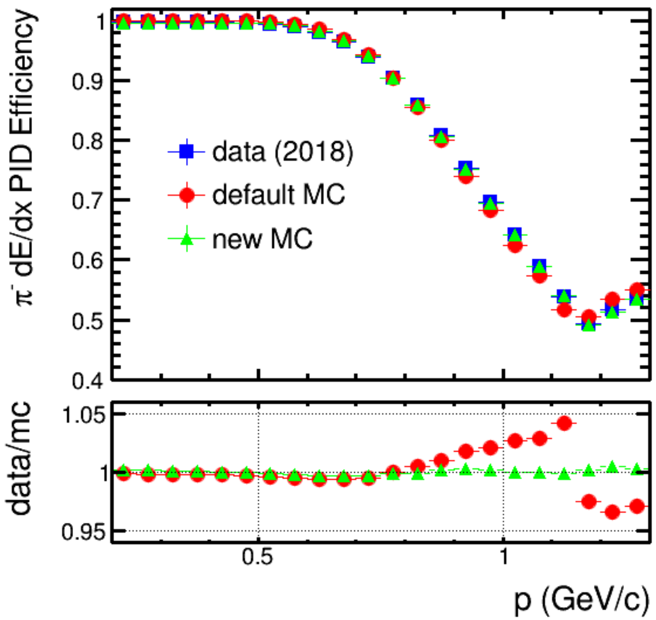


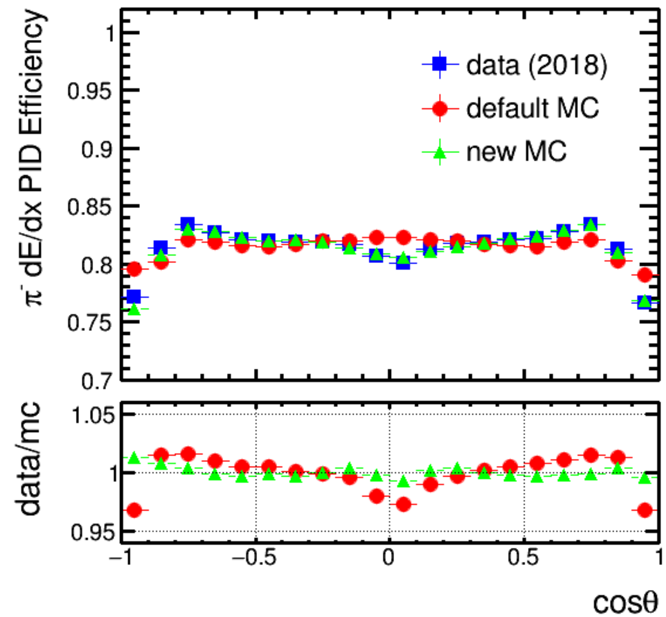


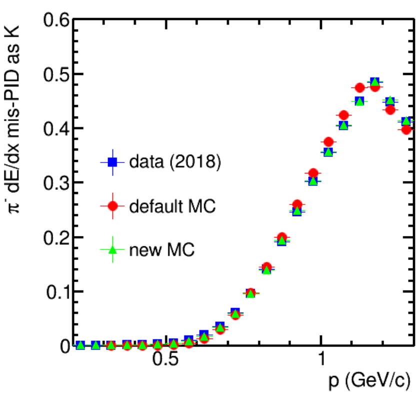


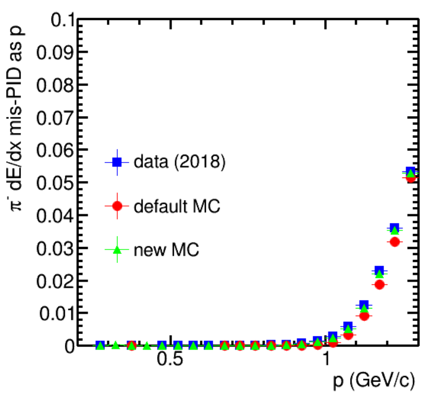


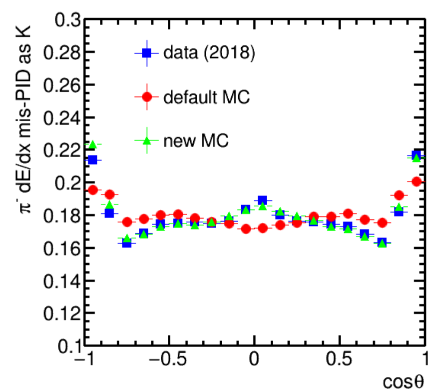
Simulation performance (pi-)

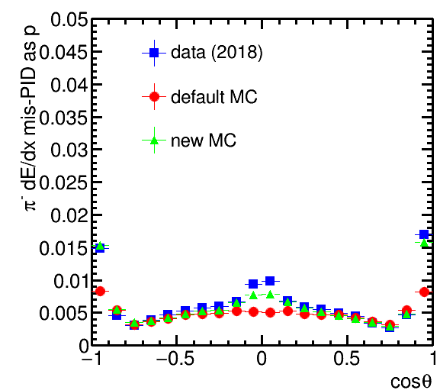




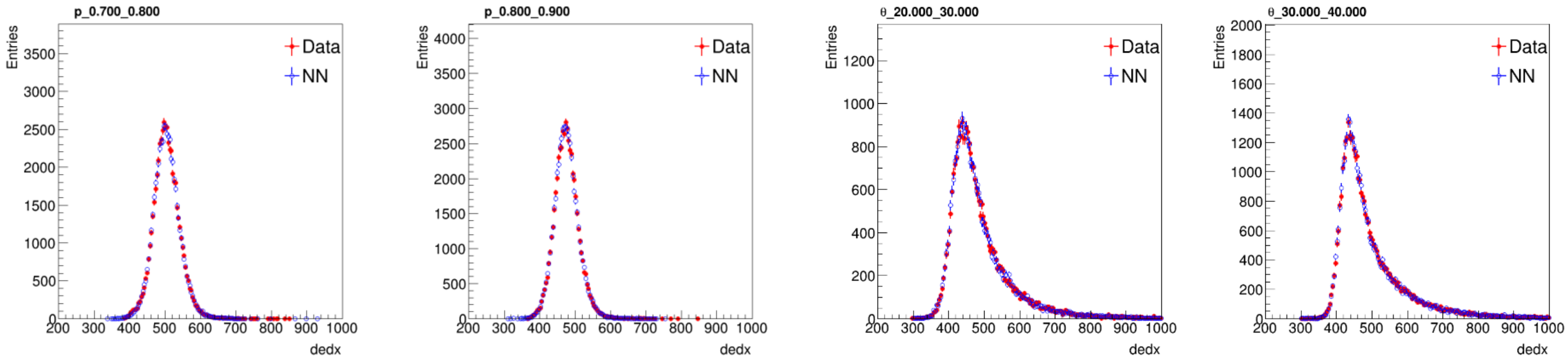




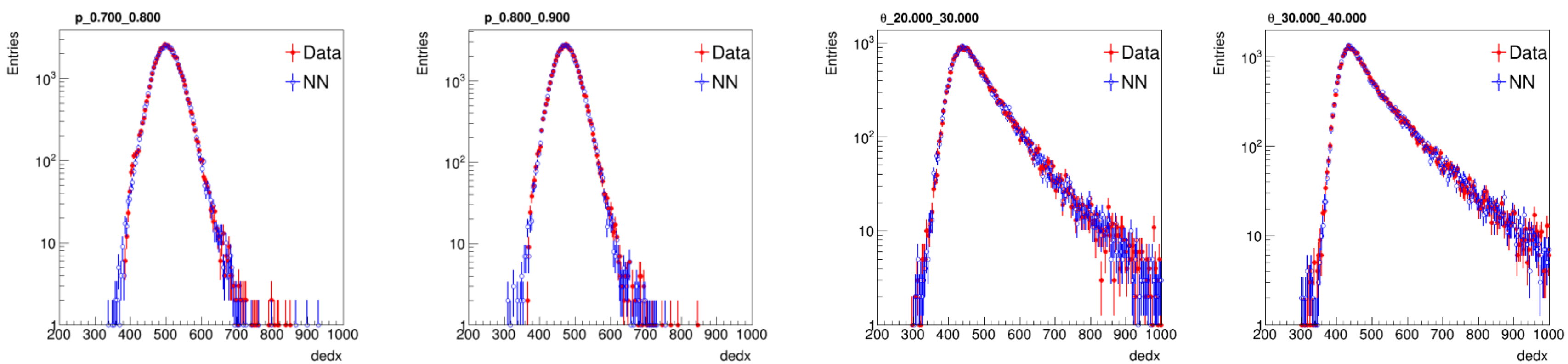




Simulation performance (K^+)

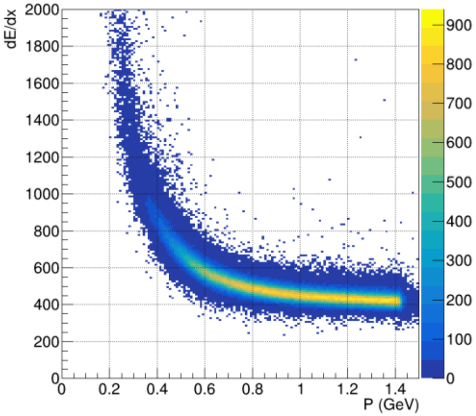


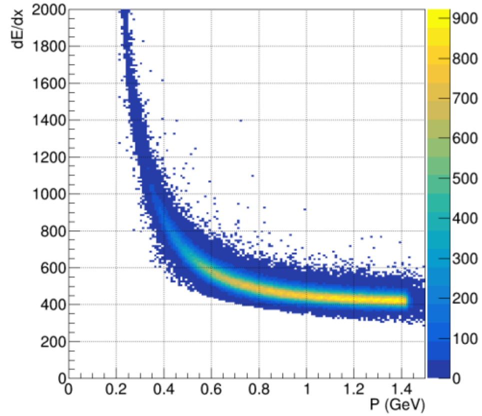
P:0.7-0.8 GeV | P:0.8-0.9 GeV | θ:20° – 30° | θ:30° – 40°



- Simulated dE/dx distribution is very similar to the data
- K^- in backup

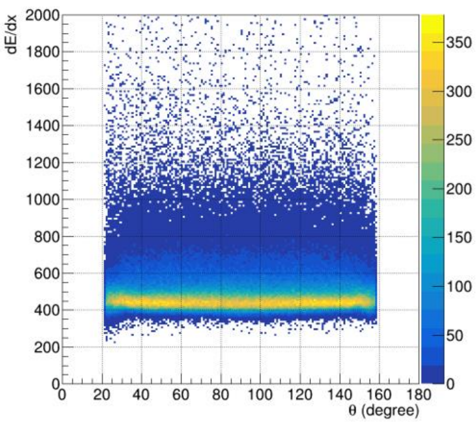
Simulation performance (K-)

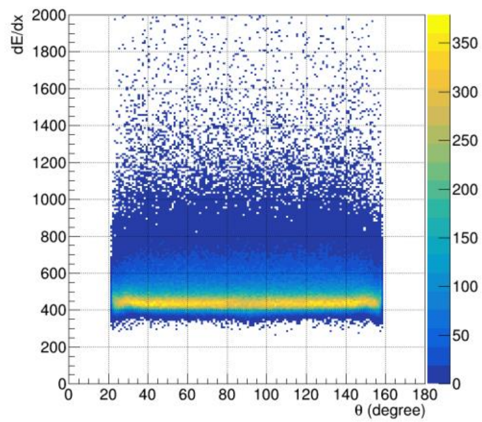




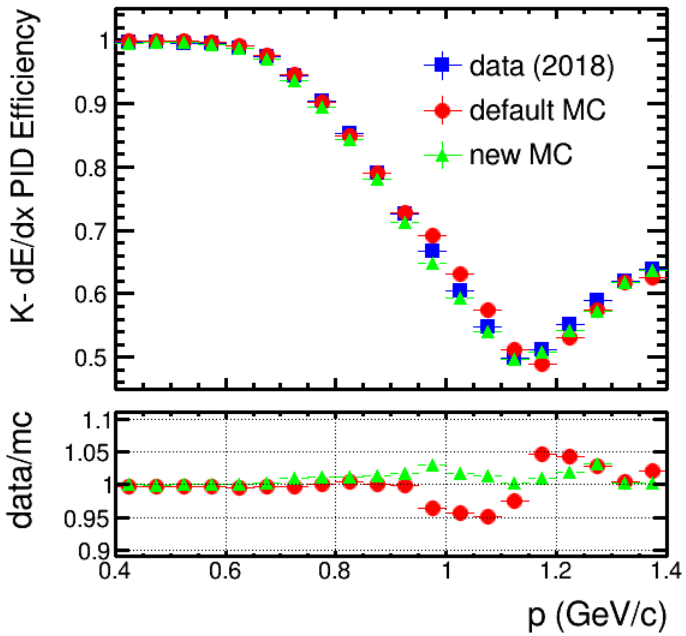
K- NN

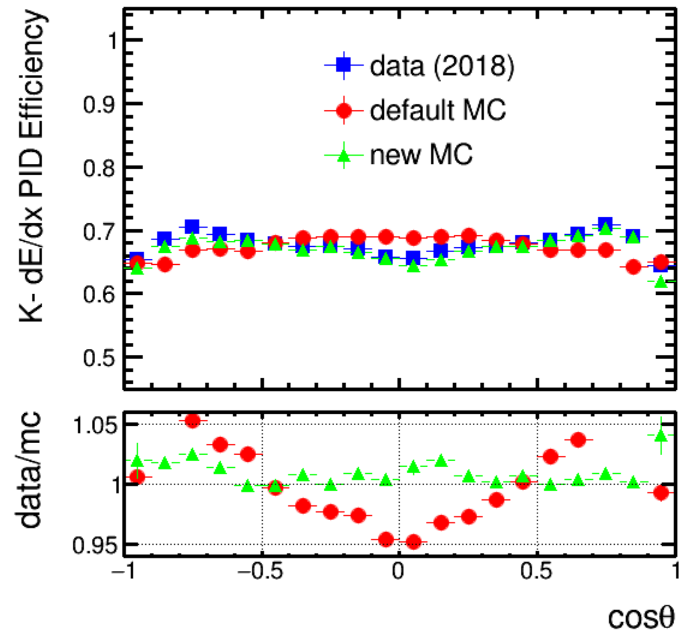
K- data

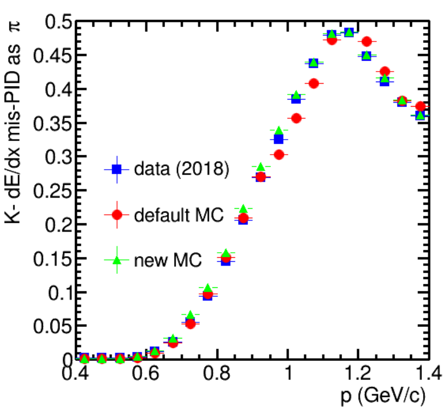


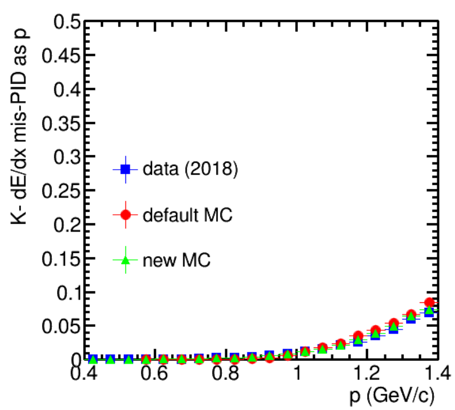


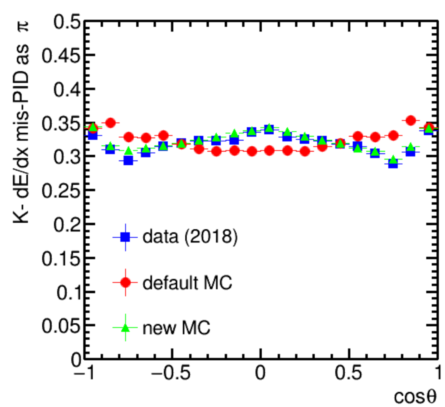
Simulation performance (K-)

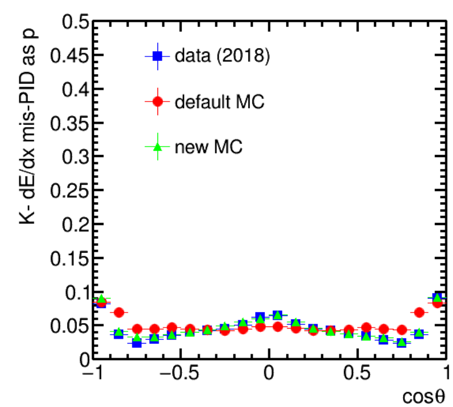




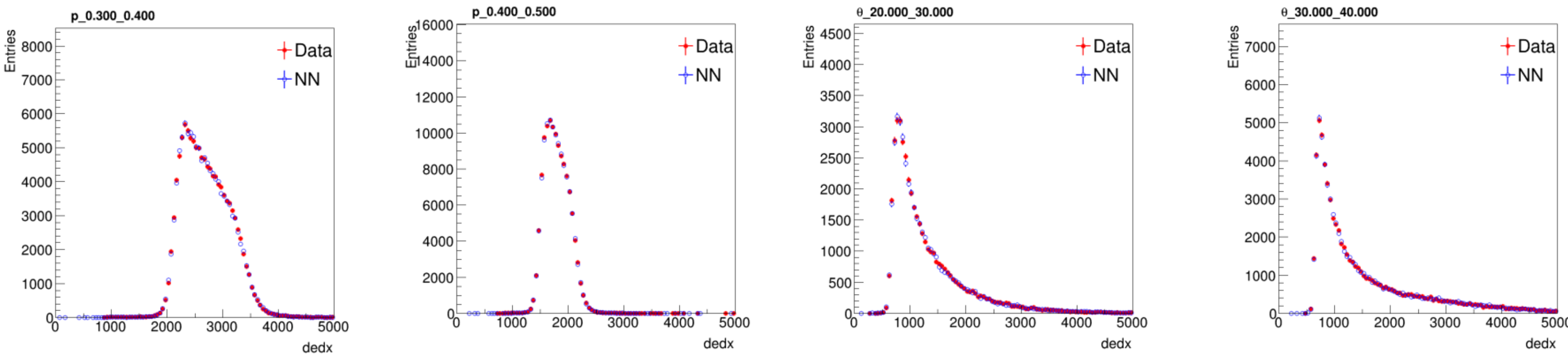




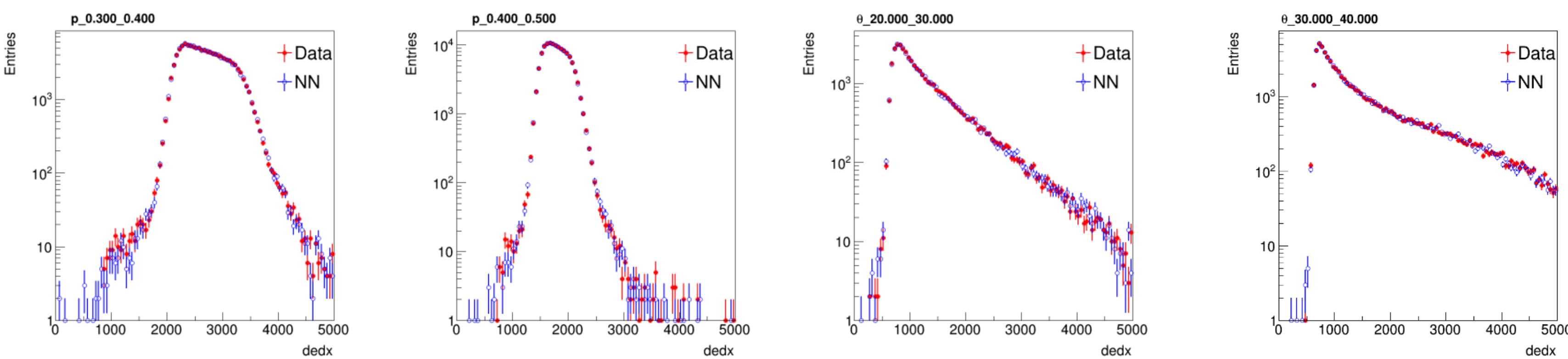




Simulation performance (p^{+})

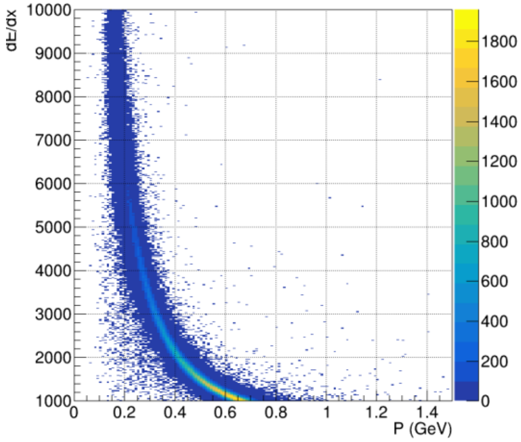


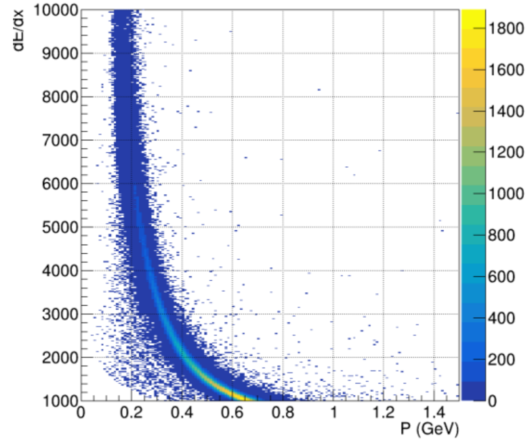
P:0.3-0.4 GeV | P:0.4-0.5 GeV | θ:$20^{\circ}-30^{\circ}$ | θ:$30^{\circ}-40^{\circ}$



- Simulated dE/dx distribution is very similar to the data
- p^{-} in backup

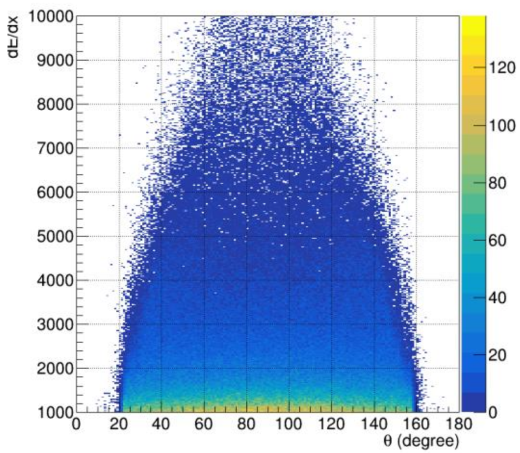
Simulation performance (p-)

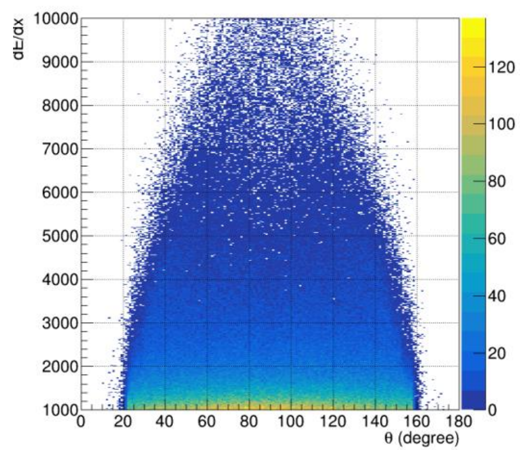




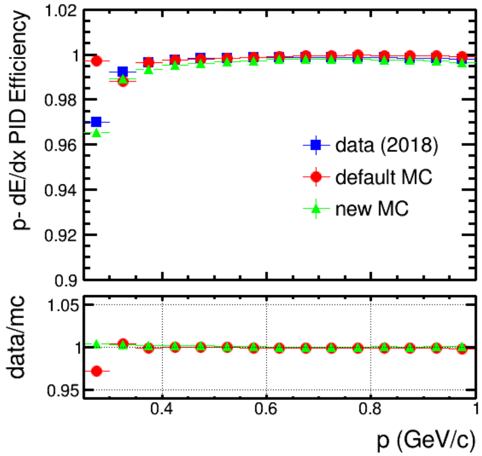
p- NN

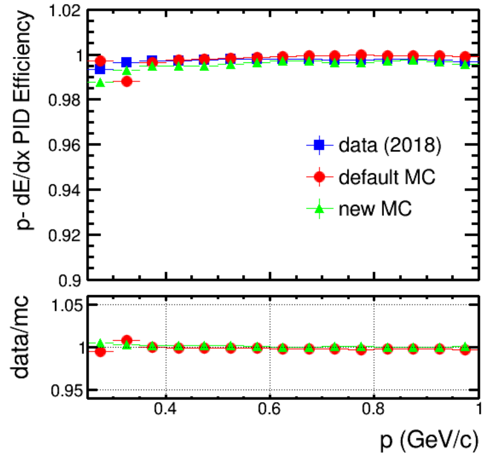
p- data

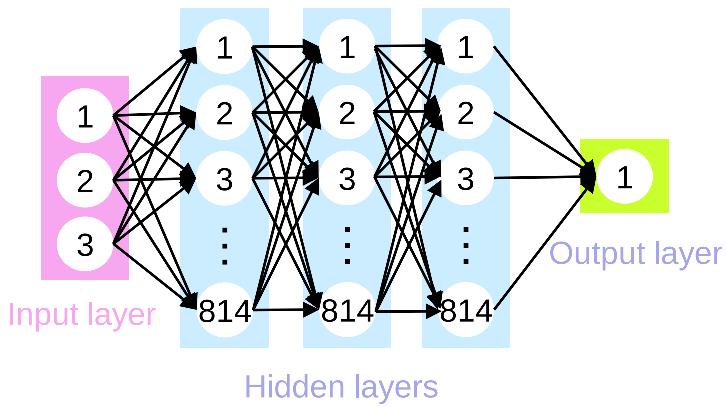




Simulation performance (p-)

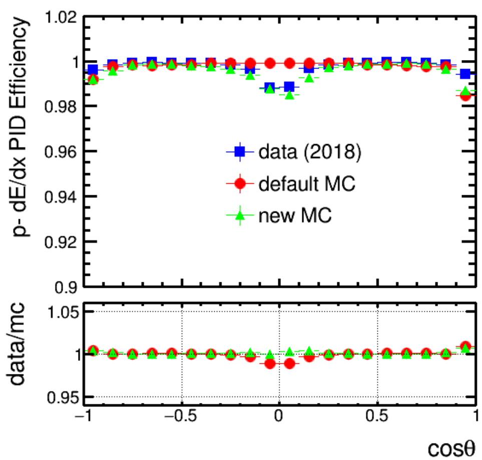


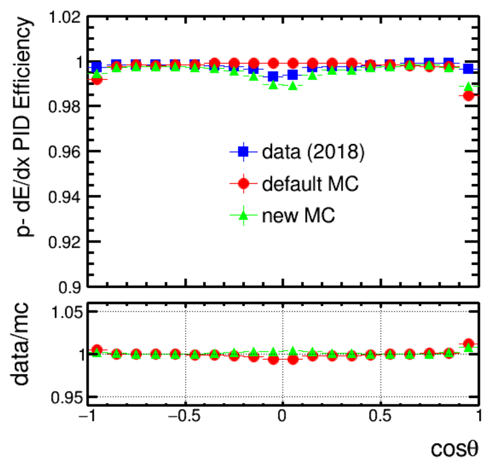


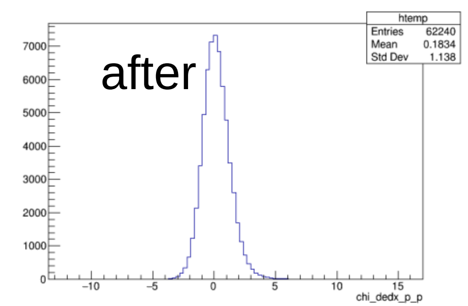


before

after







Acknowledgement

- The work has been performed in collaboration with AIDAinnova (funded by the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No 101004761)
- CAS Center for Excellence in Particle Physics
- Ministry of Science and Technology of the People's Republic of China

AIDA innova

中国科学院
CHINESE ACADEMY OF SCIENCES

中华人民共和国科学技术部
Ministry of Science and Technology of the People's Republic of China

Thanks for your attention !