

# IDENTIFIED HADRON NUCLEAR MODIFICATION FACTORS IN XeXe COLLISIONS AT $\sqrt{s_{NN}} = 5.44 \text{ TeV}$ CALCULATED BY THE NEW **Hijing++** FRAMEWORK

ZIMÁNYI WINTER SCHOOL ON HEAVY ION PHYSICS 2017

---

Gábor Bíró

4-8. december 2017.

Wigner R.C.P. of the H.A.S.  
Eötvös Loránd University

# IDENTIFIED HADRON NUCLEAR MODIFICATION FACTORS IN XeXe COLLISIONS AT $\sqrt{s_{NN}} = 5.44 \text{ TeV}$ CALCULATED BY THE NEW **Hijing++** FRAMEWORK

ZIMÁNYI WINTER SCHOOL ON HEAVY ION PHYSICS 2017

---

Gábor Bíró

4-8. december 2017.

Wigner R.C.P. of the H.A.S.  
Eötvös Loránd University

# COLLABORATORS



EMBERI ERŐFORRÁSOK  
MINISZTÉRIUMA



Gergely Gábor Barnaföldi  
Gábor Papp  
Péter Lévai  
Miklós Gyulassy  
Szilveszter Harangozó  
Xinnian Wang  
Benwei Zhang  
Guoyang Ma

The research is supported by the

- Hungarian-Chinese cooperation grant No TéT 12 CN-1-2012-0016 and No. MOST 2014DFG02050, MTA HU-China Mobility grant
- OTKA grants K120660
- Wigner GPU Laboratory
- ÚNKP-17-3 New National Excellence Program of the Ministry of Human Capacities

## INTRODUCTION

---

# MOTIVATION FOR HIJING++

What is the 'real' HIJING???



It is a BIG mess....

G.G. Barnafoldi: QCD@LHC2017

12

# MOTIVATION FOR HIJING++

We need BIG Wizards to manage it...



Gábor Papp



Gábor Bíró  
Leading by GGB



Miklós Gyulassy

and the need for a magic stick...

# MOTIVATION FOR HIJING++

We need ~~BIG Wizards~~ to manage it...  
*Students*



Gábor Papp



Gábor Bíró  
Leading by GGB



Miklós Gyulassy

and the need for a magic stick...

# MOTIVATION FOR HIJING++

What is the 'real' HIJING???



It is a BIG mess....

Also working  
on this...

G.G. Barnafoldi: QCD@LHC2017

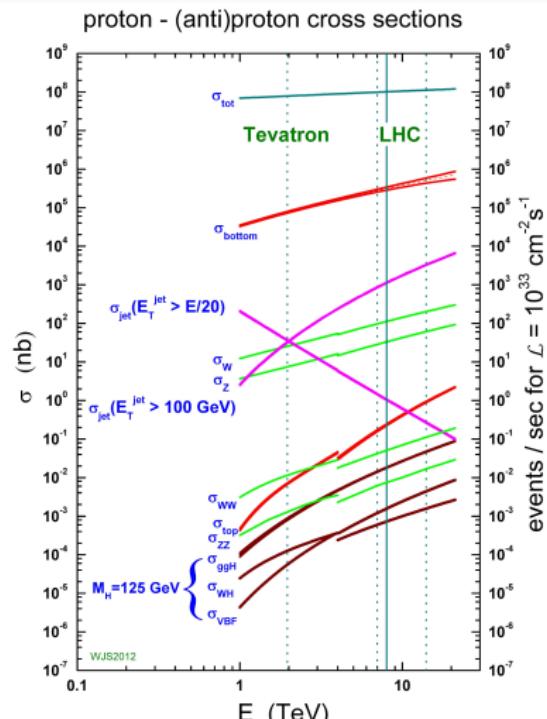
12

# MOTIVATION FOR HIJING++

Real event 

Simulated event 

- Ideal: amount of simulated data  $\approx$  real data

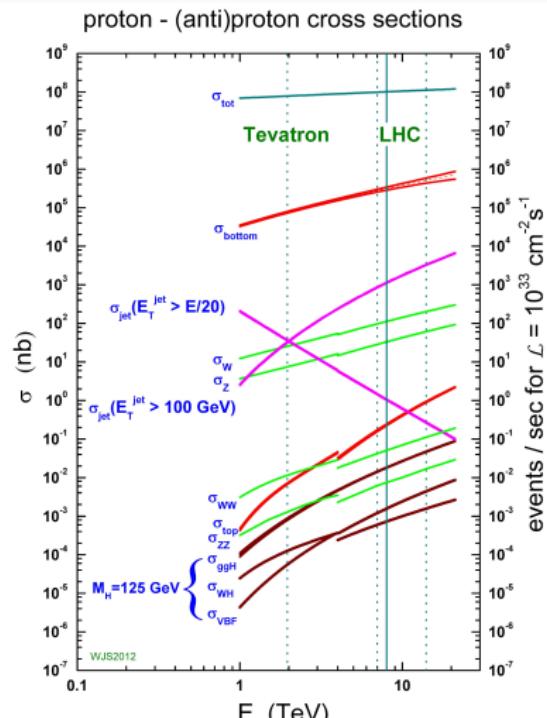


# MOTIVATION FOR HIJING++

Real event 

Simulated event 

- Ideal: amount of simulated data  $\approx$  real data
- Number of events at LHC:  $\mathcal{O}(10^8)/\text{s}$

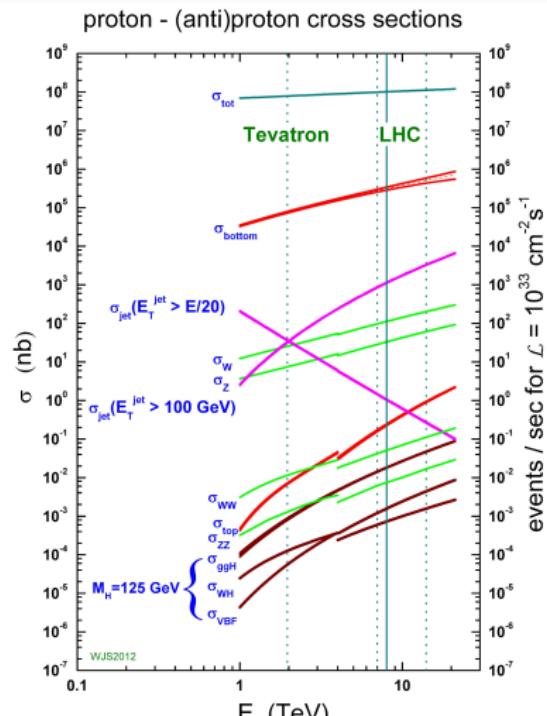


# MOTIVATION FOR HIJING++

Real event 

Simulated event 

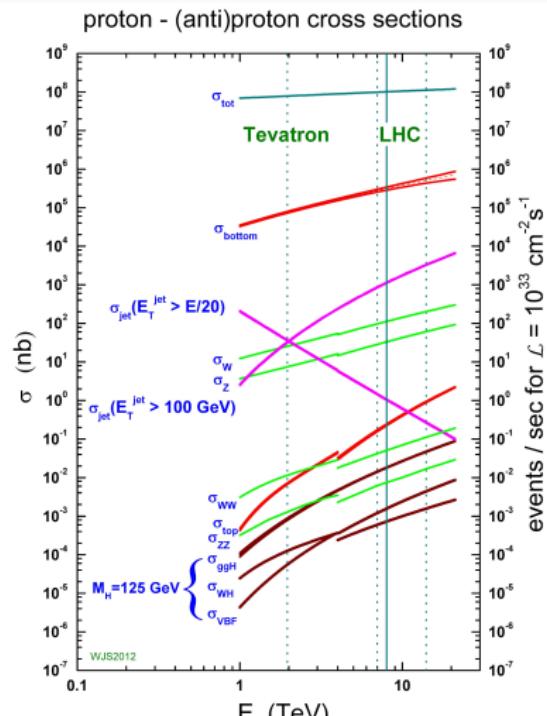
- Ideal: amount of simulated data  $\approx$  real data
- Number of events at LHC:  $\mathcal{O}(10^8)$ / s
- Necessary time to generate 1 s of ALICE data:  $\mathcal{O}(\text{days})$



# MOTIVATION FOR HIJING++



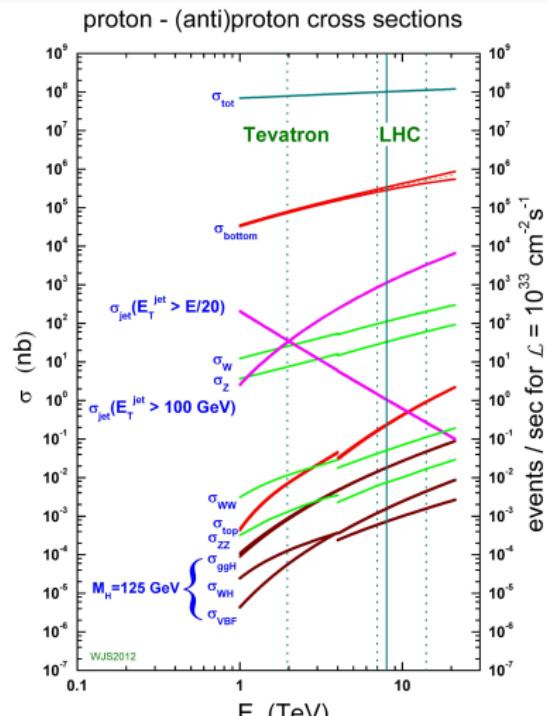
- Ideal: amount of simulated data  $\approx$  real data
- Number of events at LHC:  $\mathcal{O}(10^8)$ / s
- Necessary time to generate 1 s of ALICE data:  $\mathcal{O}(\text{days})$
- Challenge:



# MOTIVATION FOR HIJING++



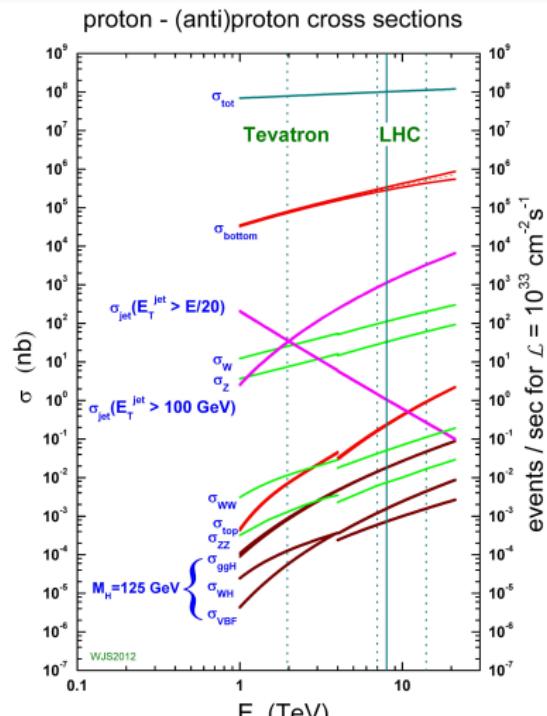
- Ideal: amount of simulated data  $\approx$  real data
- Number of events at LHC:  $\mathcal{O}(10^8)$ / s
- Necessary time to generate 1 s of ALICE data:  $\mathcal{O}(\text{days})$
- Challenge:
  - high statistics



# MOTIVATION FOR HIJING++



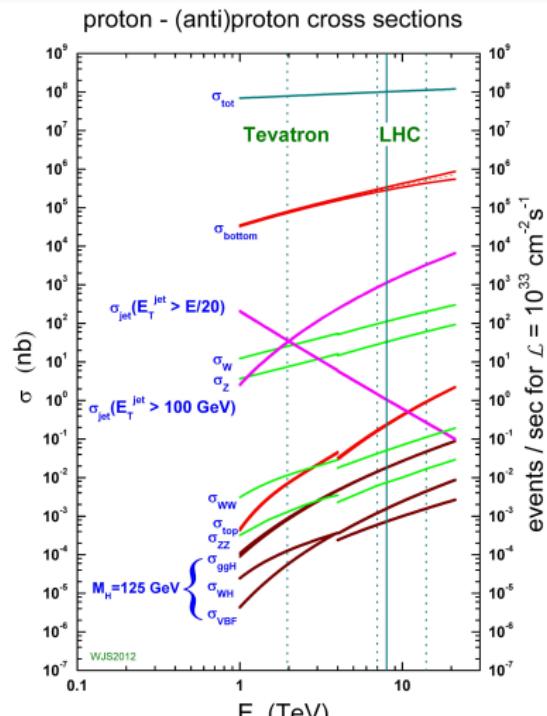
- Ideal: amount of simulated data  $\approx$  real data
- Number of events at LHC:  $\mathcal{O}(10^8)$ / s
- Necessary time to generate 1 s of ALICE data:  $\mathcal{O}(\text{days})$
- Challenge:
  - high statistics
  - event-by-event fluctuations



# MOTIVATION FOR HIJING++



- Ideal: amount of simulated data  $\approx$  real data
- Number of events at LHC:  $\mathcal{O}(10^8)$ / s
- Necessary time to generate 1 s of ALICE data:  $\mathcal{O}(\text{days})$
- Challenge:
  - high statistics
  - event-by-event fluctuations
  - modular and scalable structure



## HIJING++ IN DETAILS

---

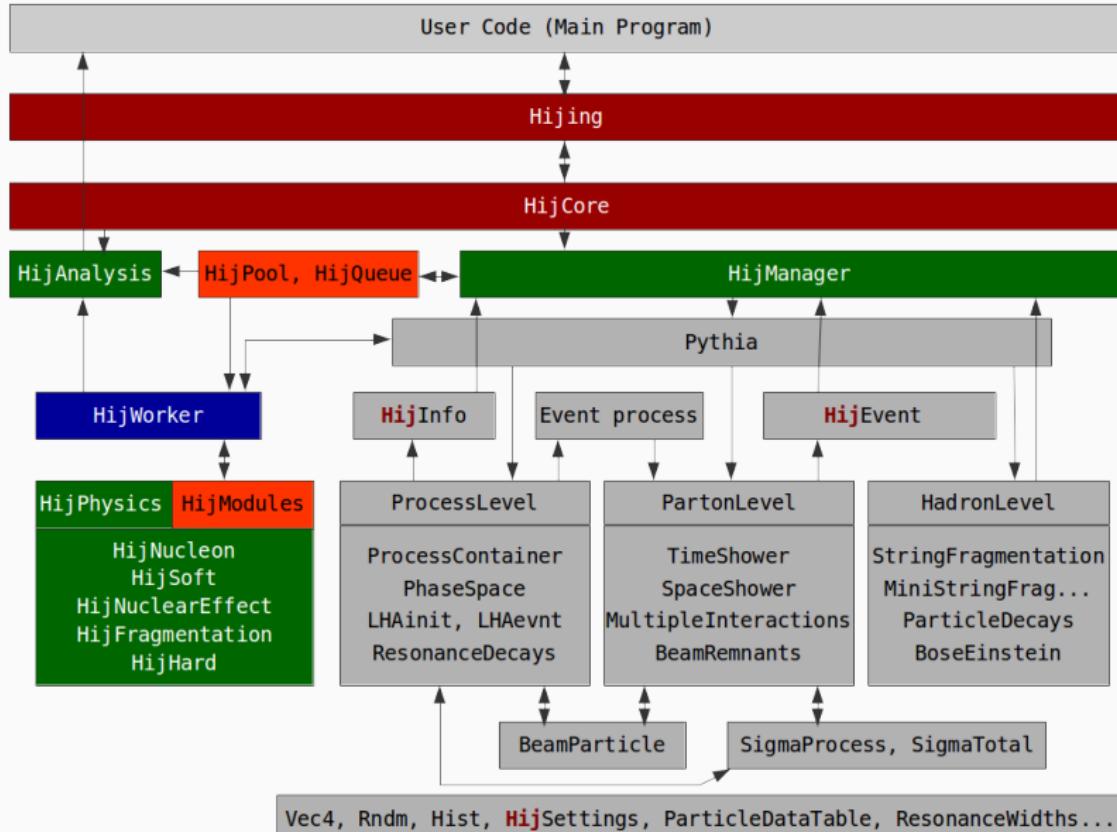
# HISTORY

- HIJING: Heavy-Ion Jet INteraction Generator
- Original FORTRAN version: M. Gyulassy, X-N. Wang, 1991
- HIJING 2.x FORTRAN X-N. Wang, W-T. Deng, 2010
- event generator based on pairwise nucleon-nucleon interactions
  - hard scattering (PYTHIA)
  - elastic collisions
  - soft scattering
    - diffractive processes, string excitations (FRITIOF)
    - soft gluon radiation (ARIADNE; Gunion-Bertch direct radiation)
  - Lund hadronization (PYTHIA)

C++ based HIJING 3.x with parallel features: 2016-18

Modern, modular, parallel C++14 framework

# PROGRAM STRUCTURE



# USAGE - BASICS

## HIJING++

- Easy to use (C++14 compiler, cmake, LHAPDF6, Pythia8)
- Optional addons (ROOT, FastJet, ...)
- Implicitly parallel

```
testSettings.cmnd:  
1 #include "Hijing.hpp"  
2  
3 using namespace Hijing3;  
4  
5 int main(int argc, char* argv[]){  
6  
7     Hijing hijing;  
8     if(argc==1)  
9         hijing.readFile("testSettings.cmnd");  
10    else  
11        hijing.readFile(argv[1]);  
12  
13    hijing.init();  
14    hijing.start();  
15 }  
16  
17 PDF:pSet = GRV98lo  
18 ! PDF:pSet = CT14nnlo  
19  
20 Hijing:threads = 3  
21 Beams:eCM   = 8160  
22 Hijing:NuclearEffects = on  
23  
24 Main:numberOfEvents = 50000  
25 Hijing:idA = P  
26 Hijing:aproj = 1  
27 Hijing:zproj = 1  
28 Hijing:idB = A  
29 Hijing:atarg = 208  
30 Hijing:ztarg = 82  
31 (...)
```

All parameters in **xml** files, full compatibility with Pythia8

# USAGE - HijAnalysis

```
1 hijing.newAnalysis("root", "HardCollision", "pTdata_hard", 50, 0.0001, 20.0);
2
3 hijing.newAnalysis("root", "SoftProcess", "proj_pxpy", 100, -15.0, 15.0,
4                                     100, -15.0, 15.0,
5                                     "p_{x,Proj}", "p_{y,Proj}");
6
7 hijing.newAnalysis("ascii","SoftProcess", "proj_pz_ascii",100, -5000.0, 15000.0);
8
9 hijing.newAnalysis("root", "raw","EventEnd","raw_data");
10
11 ///////////////////////////////////////////////////////////////////
12
13 hijing.analysisProperties("pTdata_hard","pT","ID1","ID2"); // u and d quarks
14
15 hijing.analysisProperties("proj_pz_ascii","final","pz");
16
17
18
```

Define a **new** analysis:

```
newAnalysis(type, location, name, histogram details)
```

Define the **properties** of an **existing** analysis:

```
analysisProperties(name, property1, property2, ...)
```

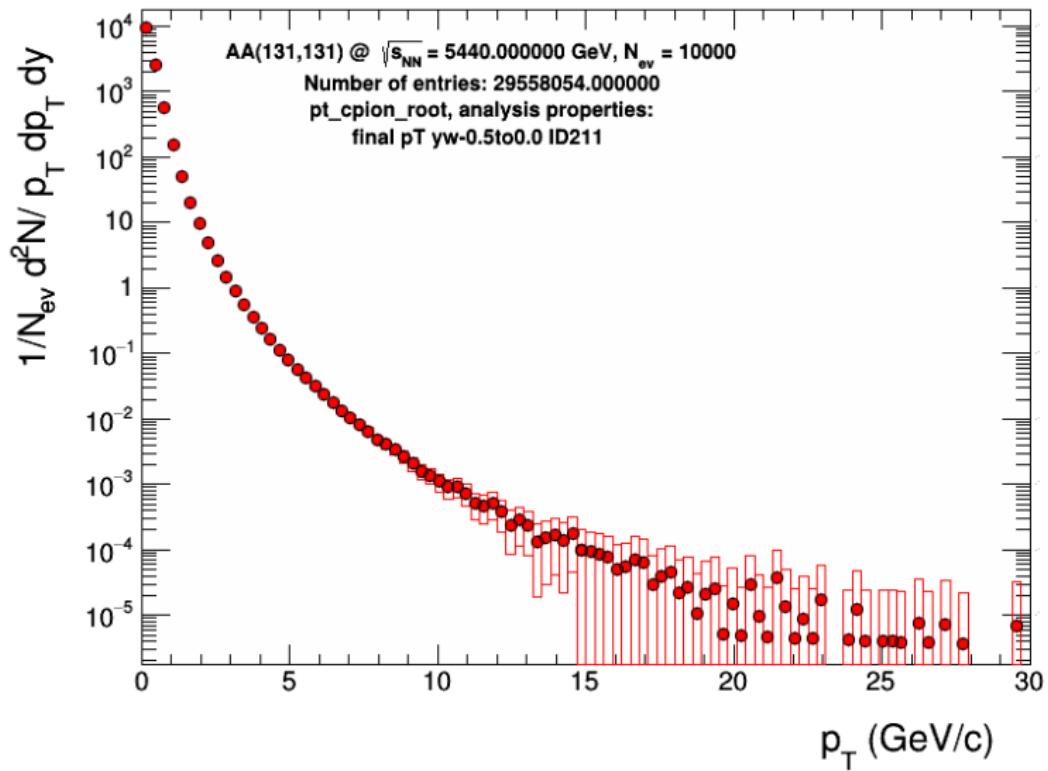
# USAGE - HijAnalysis

Default: simple ascii file

```
1 # Name: ptEnd_charged_ascii
2 # System: PP(1,1) at #sqrt{s_{NN}} = 8160 GeV, N_{ev} = 50000
3 # Entries: 3968
4 # Analysis properties: charged final pT
5 # bincenter    binwidth      value   stat.error
6 0.4           0.1          1.3951    7.5726e-08
7 1.2           0.1          0.071687   3.3406e-07
8 2             0.1          0.015661   7.1472e-07
9 2.8           0.1          0.0021052  1.9494e-06
10 3.6           0.1          0.00072602 3.3195e-06
11 4.4           0.1          0.00014937 7.3183e-06
12 5.2           0.1          9.4148e-05 9.2181e-06
13 6             0.1          1.2517e-05 2.5281e-05
14 6.8           0.1          3.422e-05  1.529e-05
15 7.6           0.1          1.0082e-05 2.8168e-05
16 ...
```

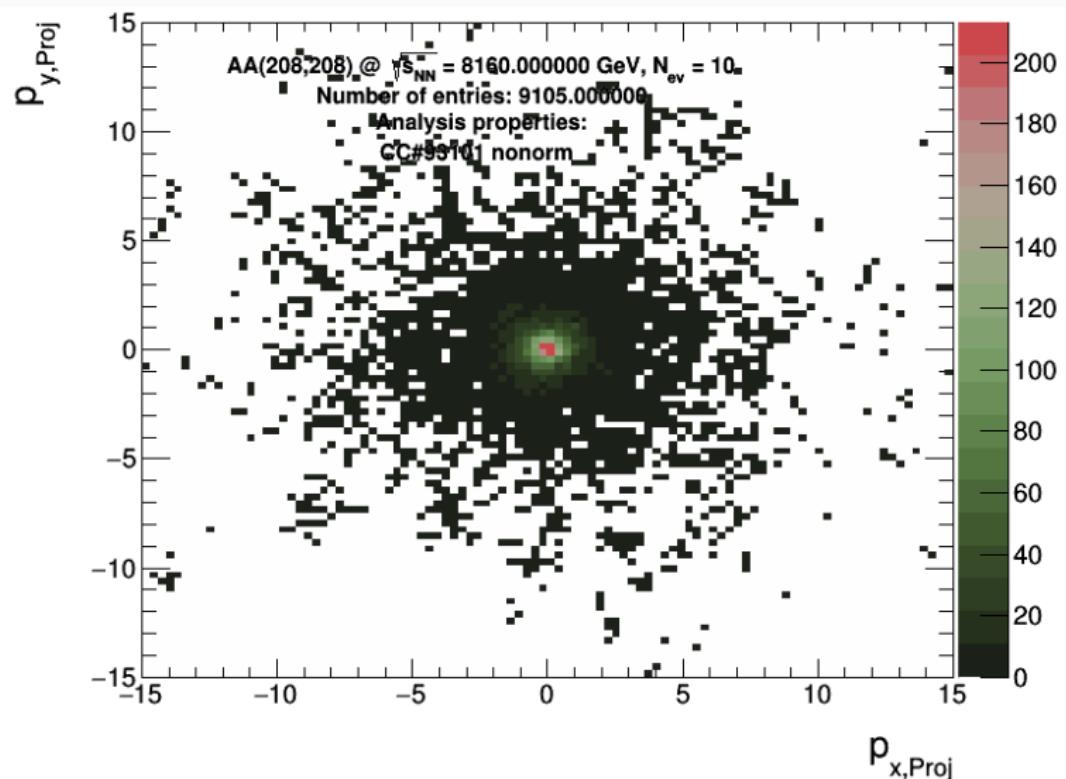
# USAGE - HijAnalysis

With **ROOT** support:



# USAGE - HJANALYSIS

With **ROOT** support:



# USAGE - HijAnalysis

With **ROOT** support:

```
1
2 =====
3 j   id      E      px     py     pz     m      phi    eta    y      theta
4 =====
5 Event 9:
6
7 0   -2112   1.97   0.597   -0.482   1.55   0.94   -0.679   1.45   1.06   0.461
8 1   211     0.184   0.0317   0.0795   -0.083   0.14   1.19   -0.86   -0.487   2.34
9 2   -211    1.57    -1.4     0.626   0.287   0.14   2.72   0.186   0.185   1.39
10 3   -13     1.4     -0.952   0.692   0.749   0.106   2.51   0.6     0.598   1
11 4   14     1.51    -0.563   0.469   1.32    0       2.45   1.35   1.35   0.508
12 5   2112   3.35    -2.17   0.607   2.29    0.94   2.87   0.895   0.839   0.776
13 6   211    0.323   0.0635   -0.172   0.226   0.14   -1.22   1.04   0.868   0.682
14 ...
```

# USAGE - HijANALYSIS

Complete user freedom with user analysis:

```
1 ofstream outfile("myAnalysis.dat");
2 auto myCode = [] (unique_ptr<HijEvent>& hijevent) {
3
4     for (int i=0; i<hijevent->event.size(); i++)
5     {
6         if (hijevent->event[i].isFinal()){
7             outfile << hijevent->event[i].eta();
8             // do something else
9         }
10    }
11};
```

New analysis with **user** type:

```
1 hijing.newAnalysis("user", "EventEnt", "myAnalysis", myCode);
```

Or, attach to an **existing** analysis:

```
1 hijing.analysisCustomCode(93001,myCode);
2 hijing.analysisProperties("myAnalysis", "CC#93001");
```

# PARALLELISM, PERFORMANCE BENCHMARKS

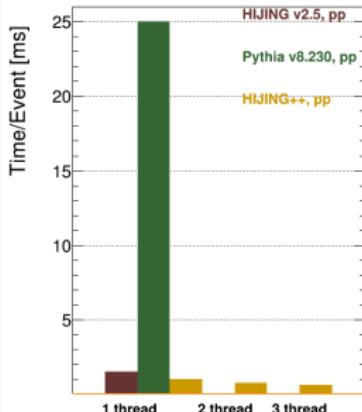
Utilizing more CPU threads:

```
1     hijing.readString("Hijing:threads = 3");
2
```

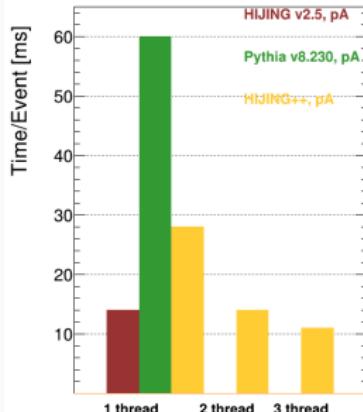
Or in a settings.cmnd:

```
1     Hijing:threads = 3
2     ! 0: get maximum number depending on hardware
3     ! default: 1
```

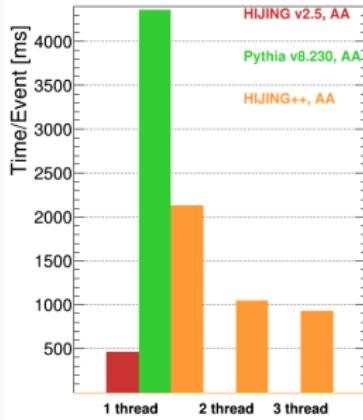
pp



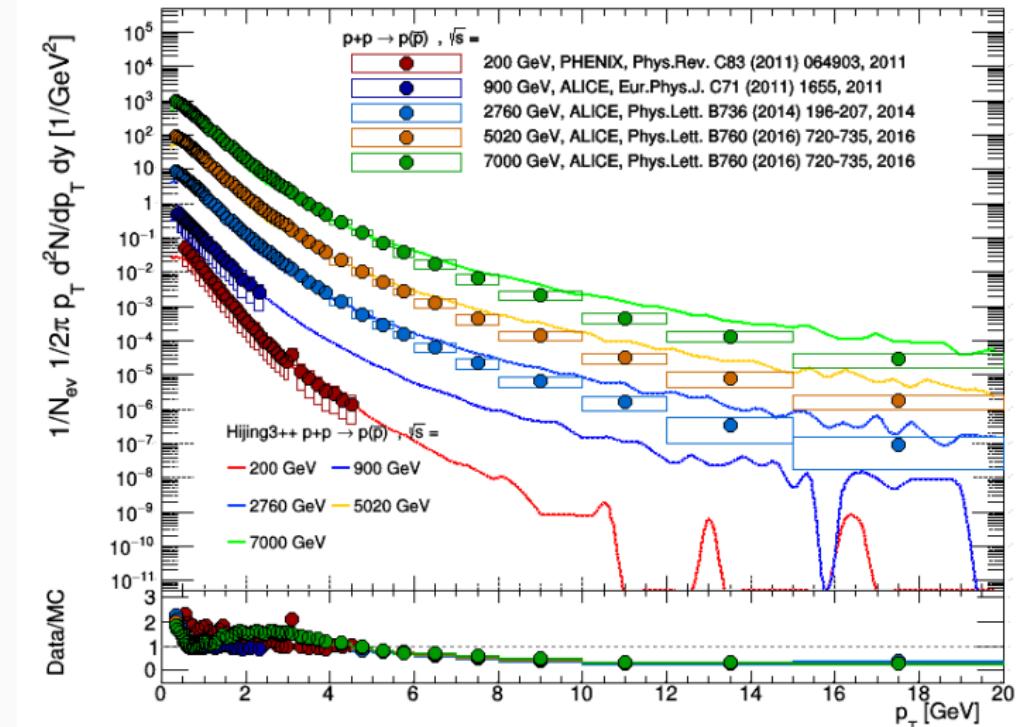
pA



AA

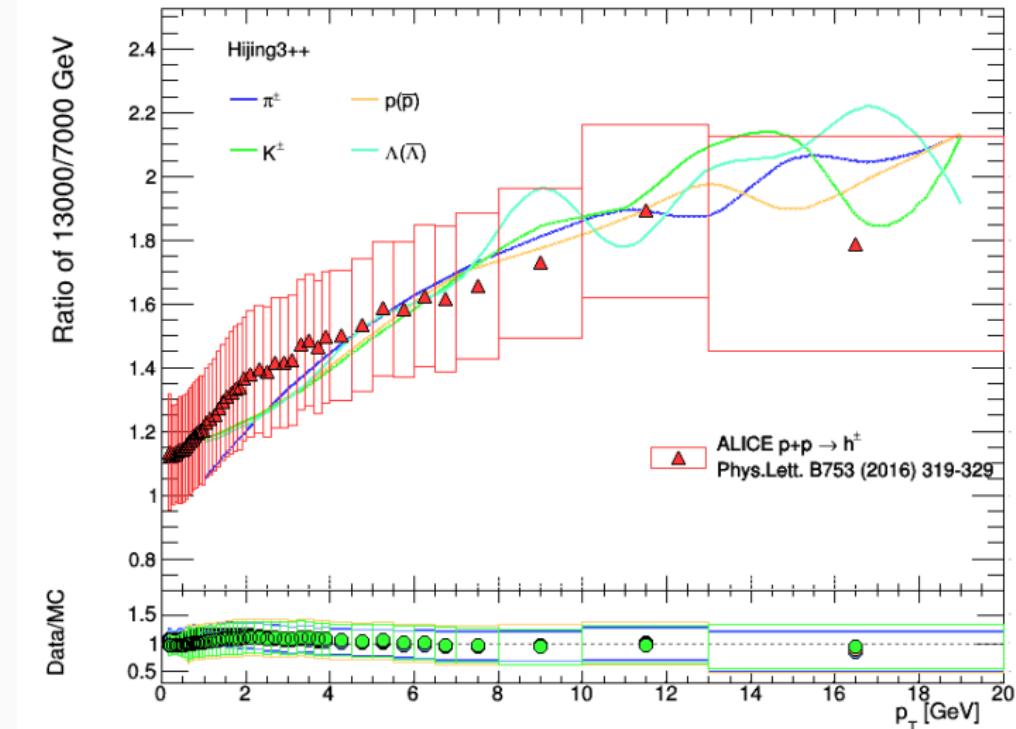


# PHYSICS BENCHMARKS

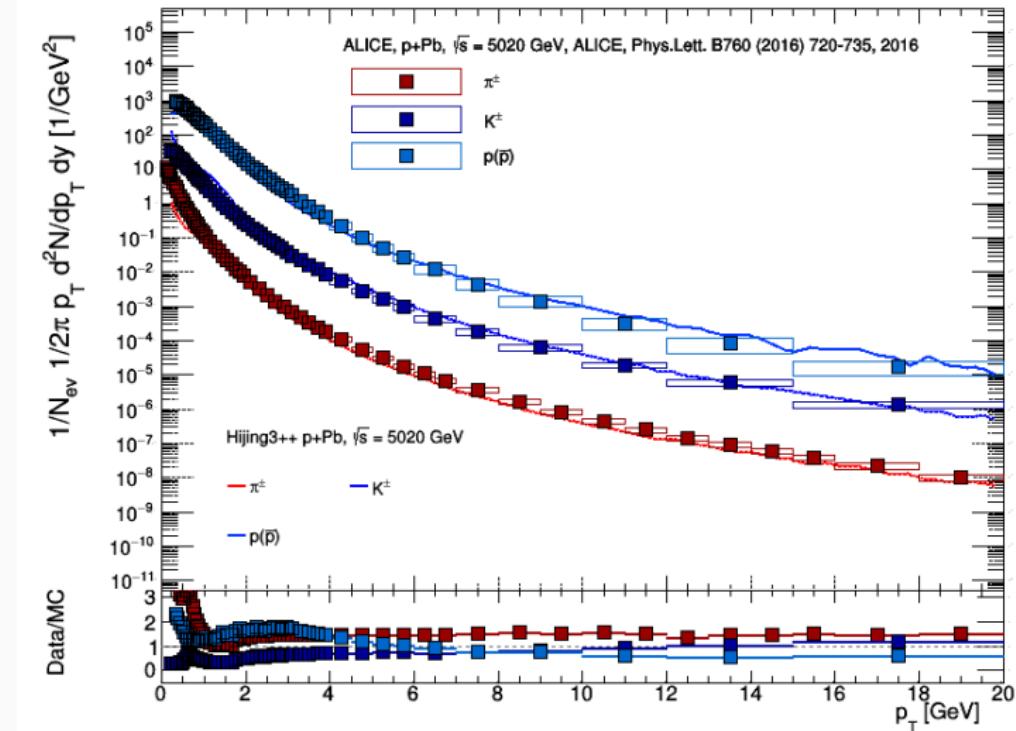


Identified hadron spectra measured in  $p + p$  collisions (presented in SQM17)

# PHYSICS BENCHMARKS

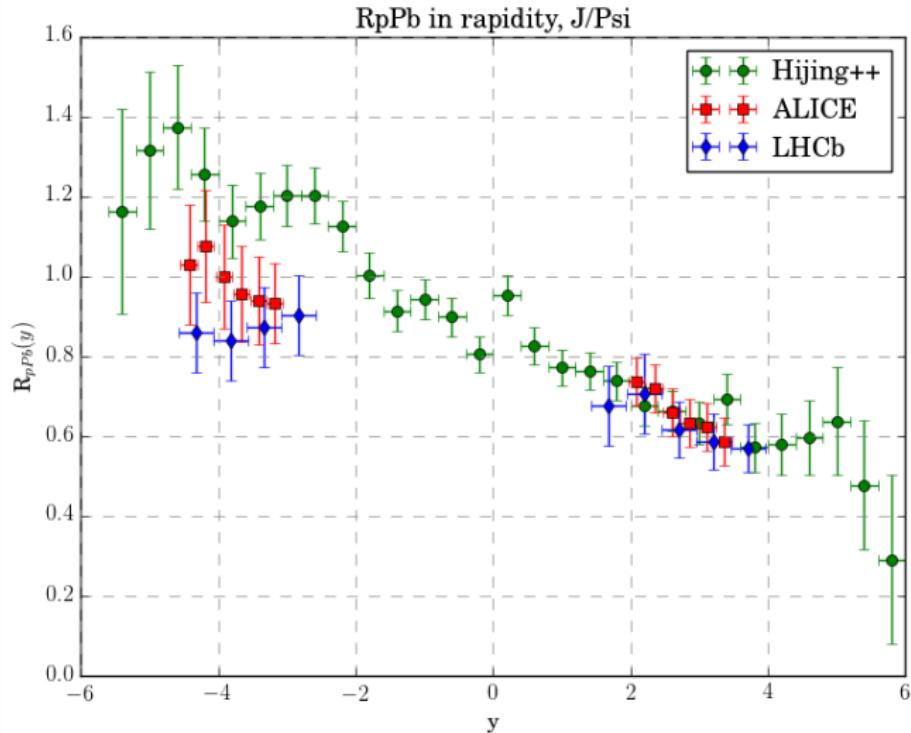


# PHYSICS BENCHMARKS



Identified hadron spectra measured in  $p + Pb$  collisions, with old Hijing shadowing  
(presented in SQM17)

# PHYSICS BENCHMARKS

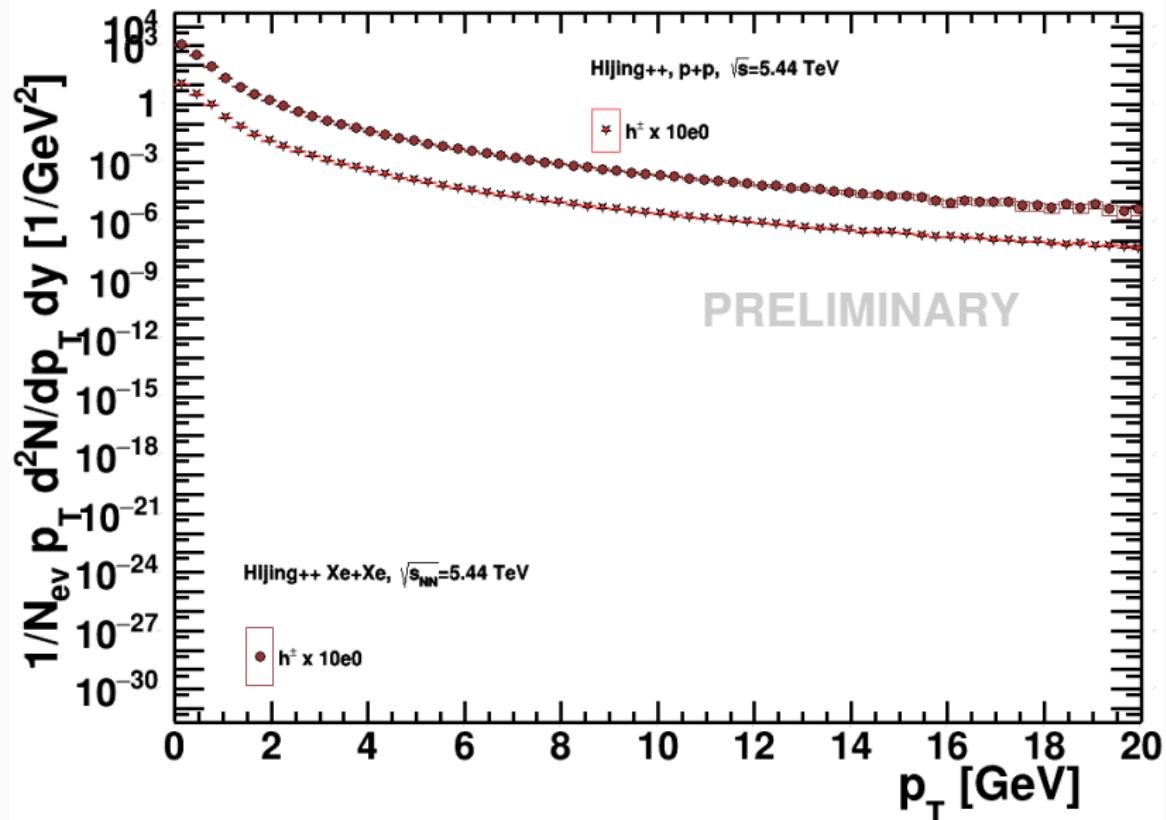


Predictions for p+Pb Collisions at  $\sqrt{s_{NN}} = 8.16$  TeV; arXiv:1707.09973 (L. Albacete *et al.*)

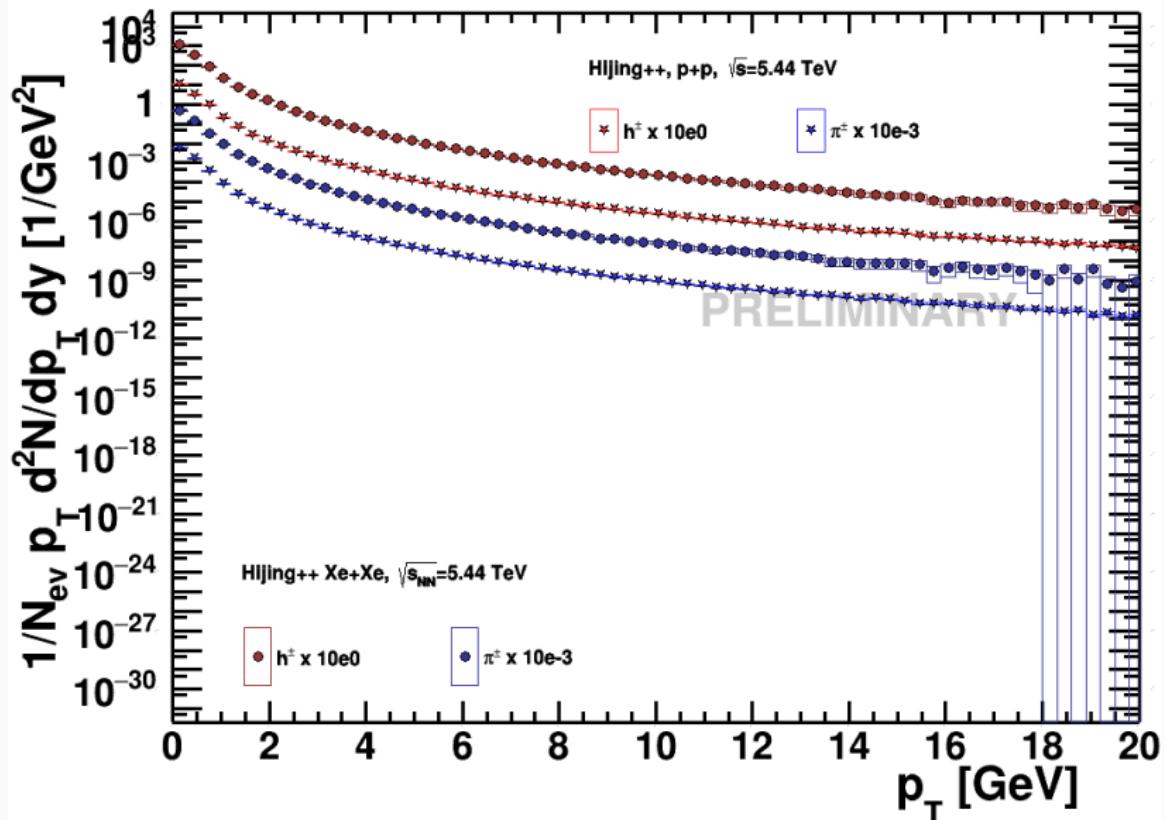
## PREDICTIONS FOR THE XE<sub>E</sub> RUN

---

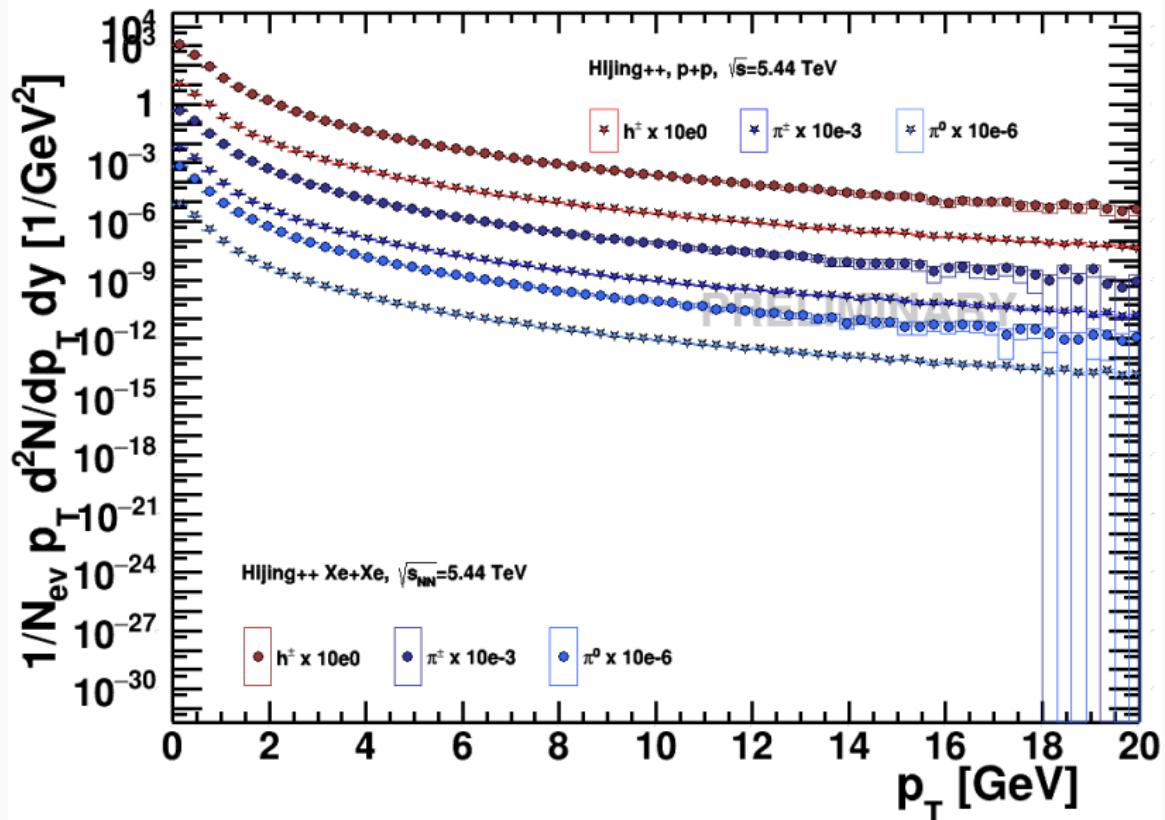
# IDENTIFIED HADRON SPECTRA



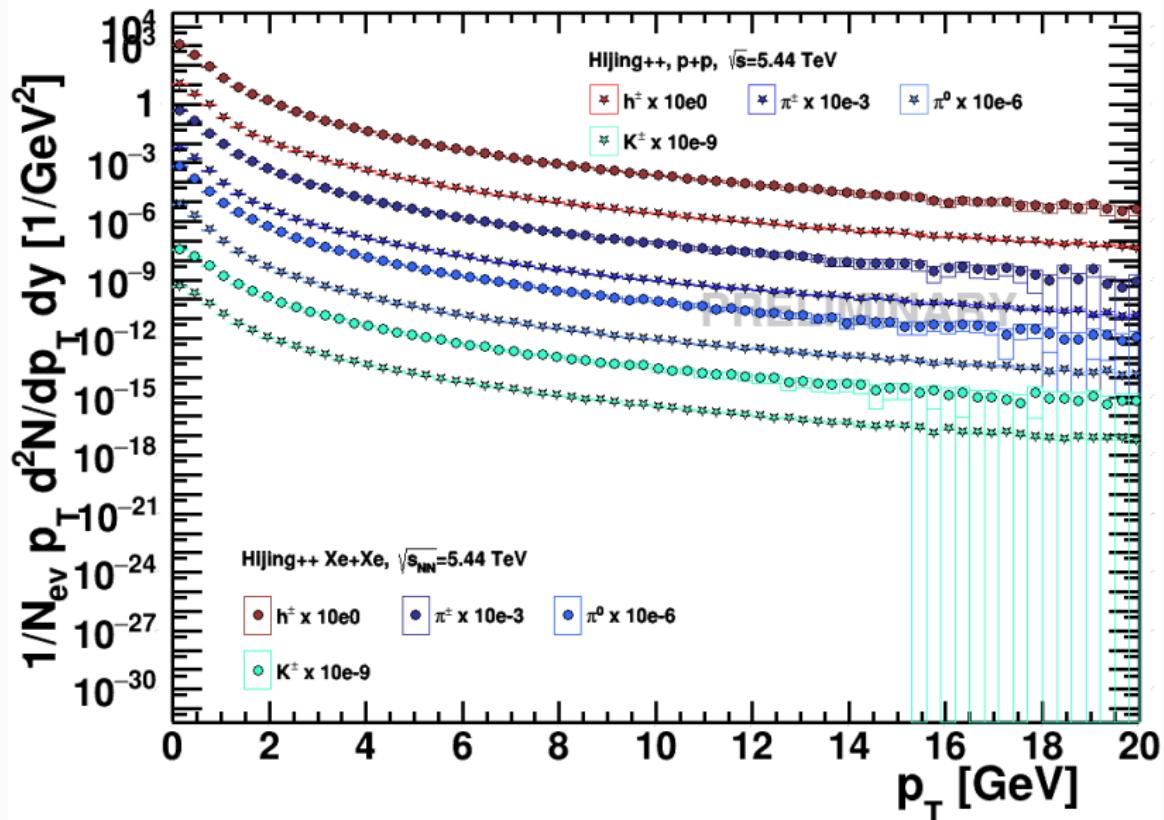
# IDENTIFIED HADRON SPECTRA



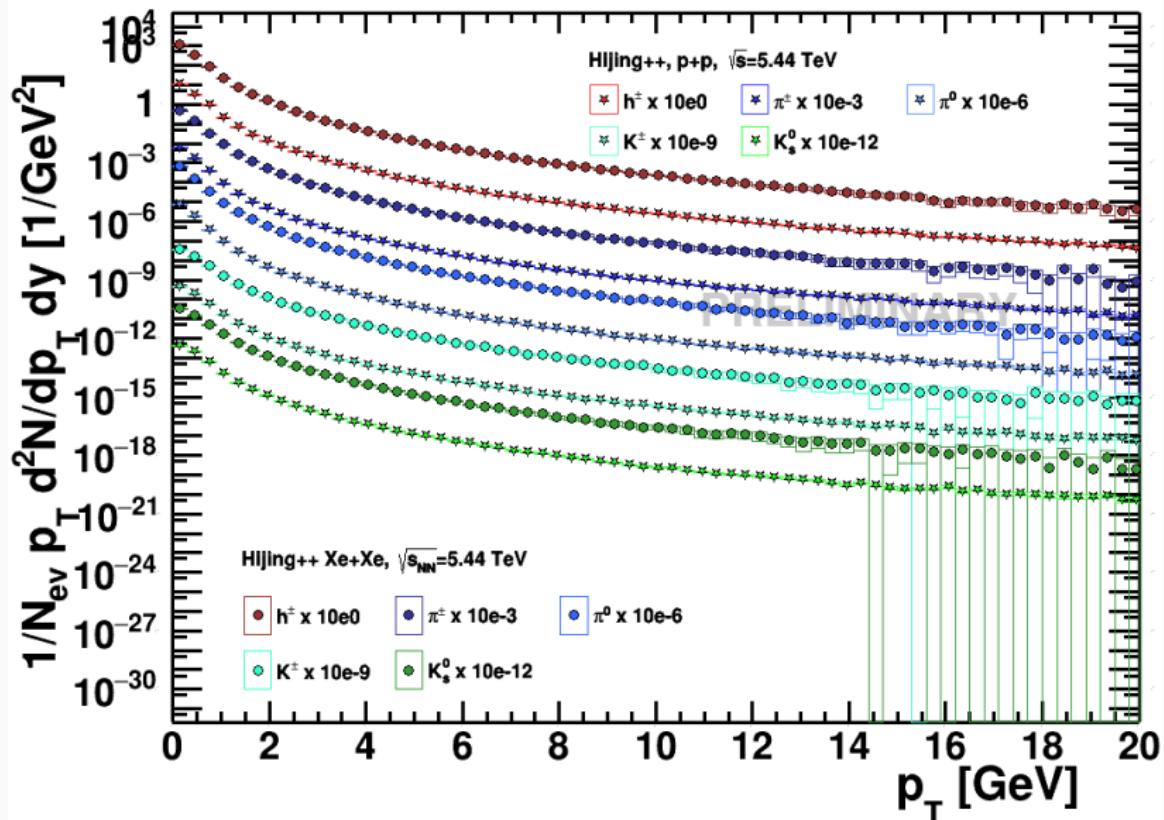
# IDENTIFIED HADRON SPECTRA



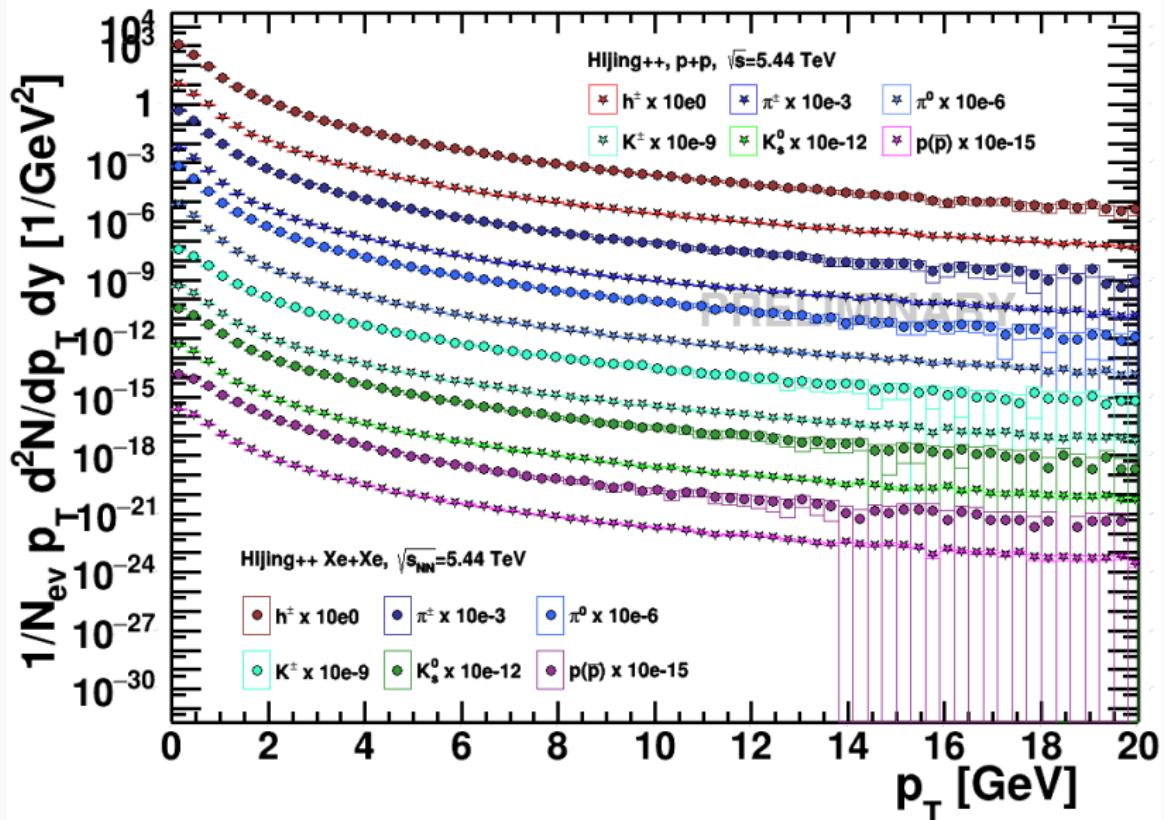
# IDENTIFIED HADRON SPECTRA



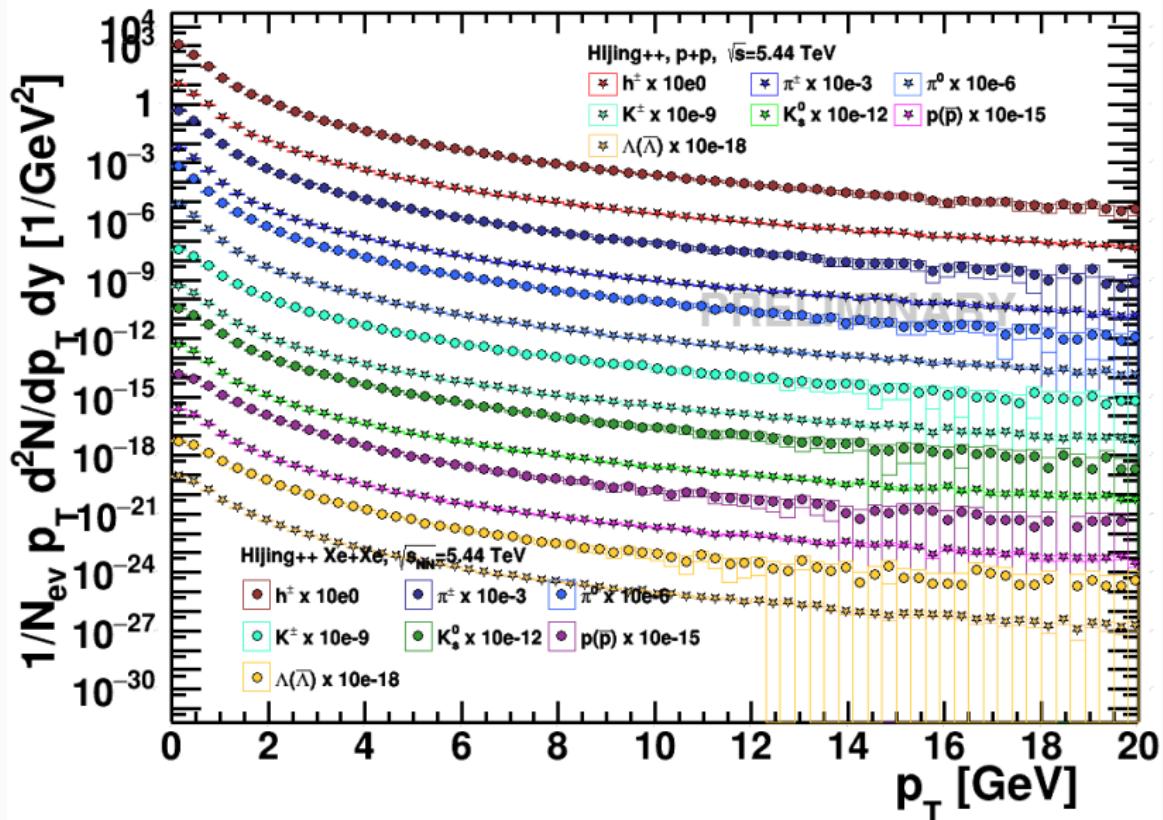
# IDENTIFIED HADRON SPECTRA



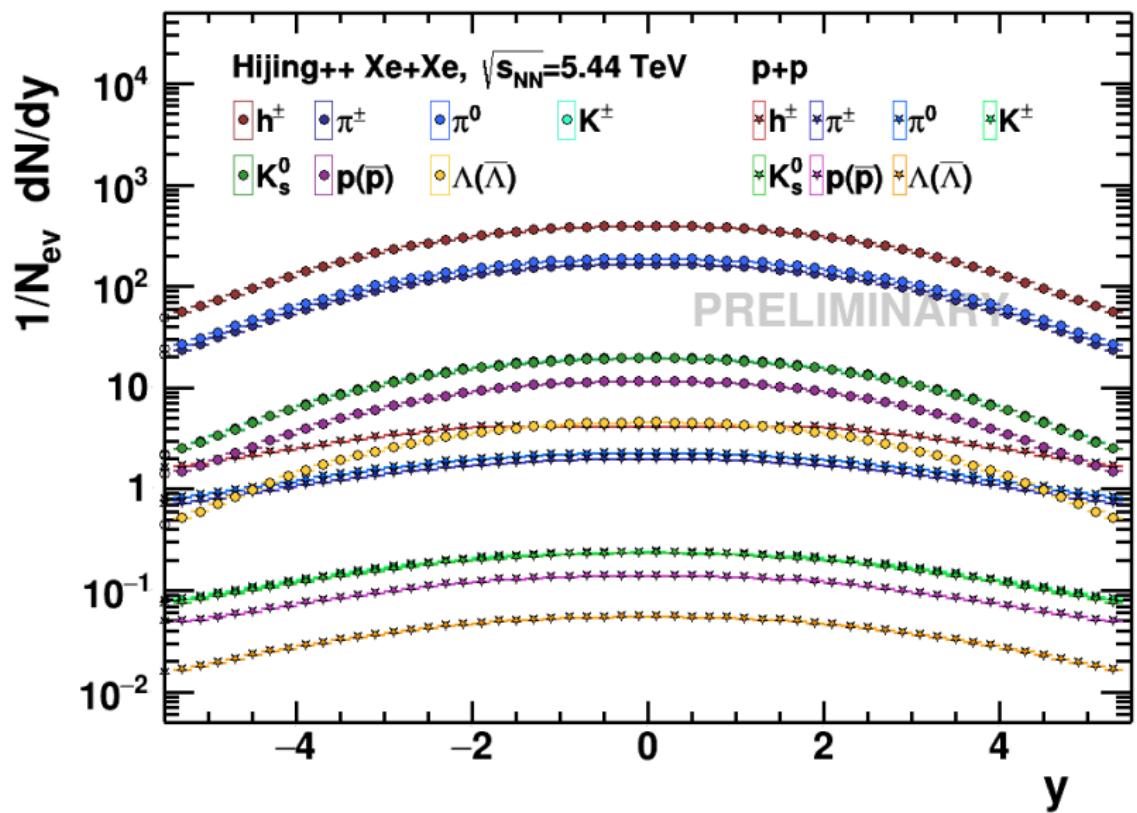
# IDENTIFIED HADRON SPECTRA



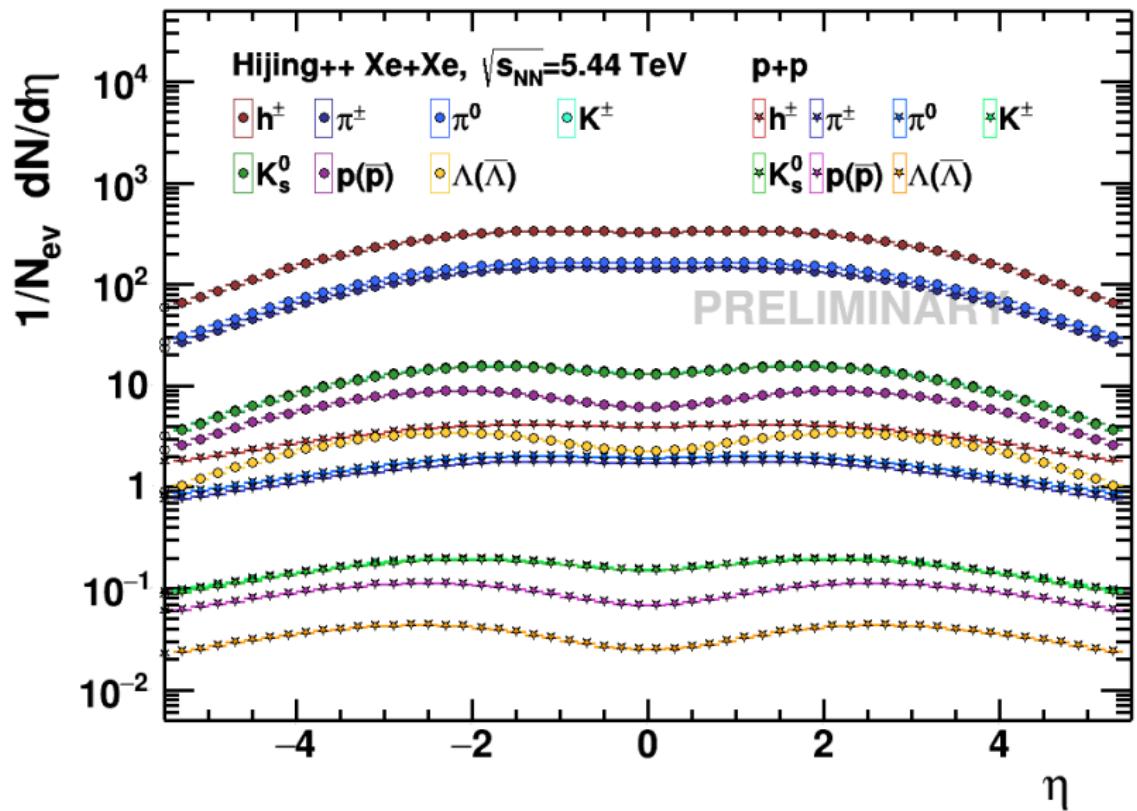
# IDENTIFIED HADRON SPECTRA



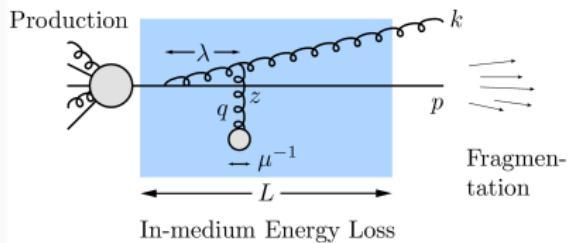
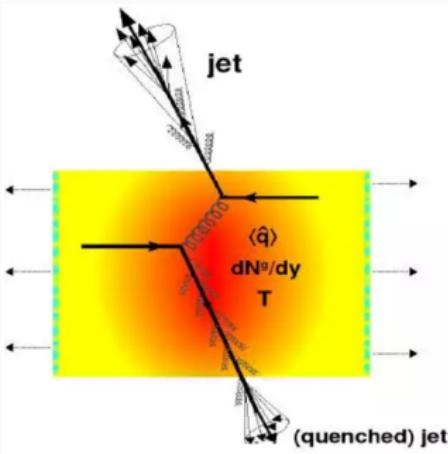
# RAPIDITY DISTRIBUTION



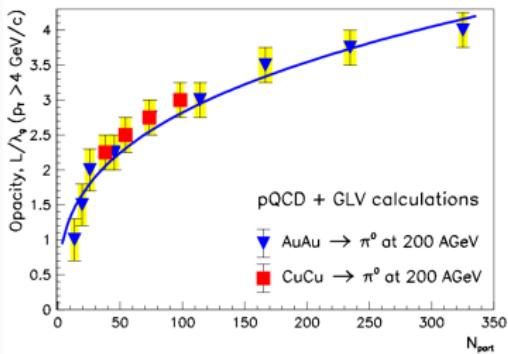
# PSEUDORAPIDITY DISTRIBUTION



# JET QUENCHING BASED ON GLV OPACITY APPROACH



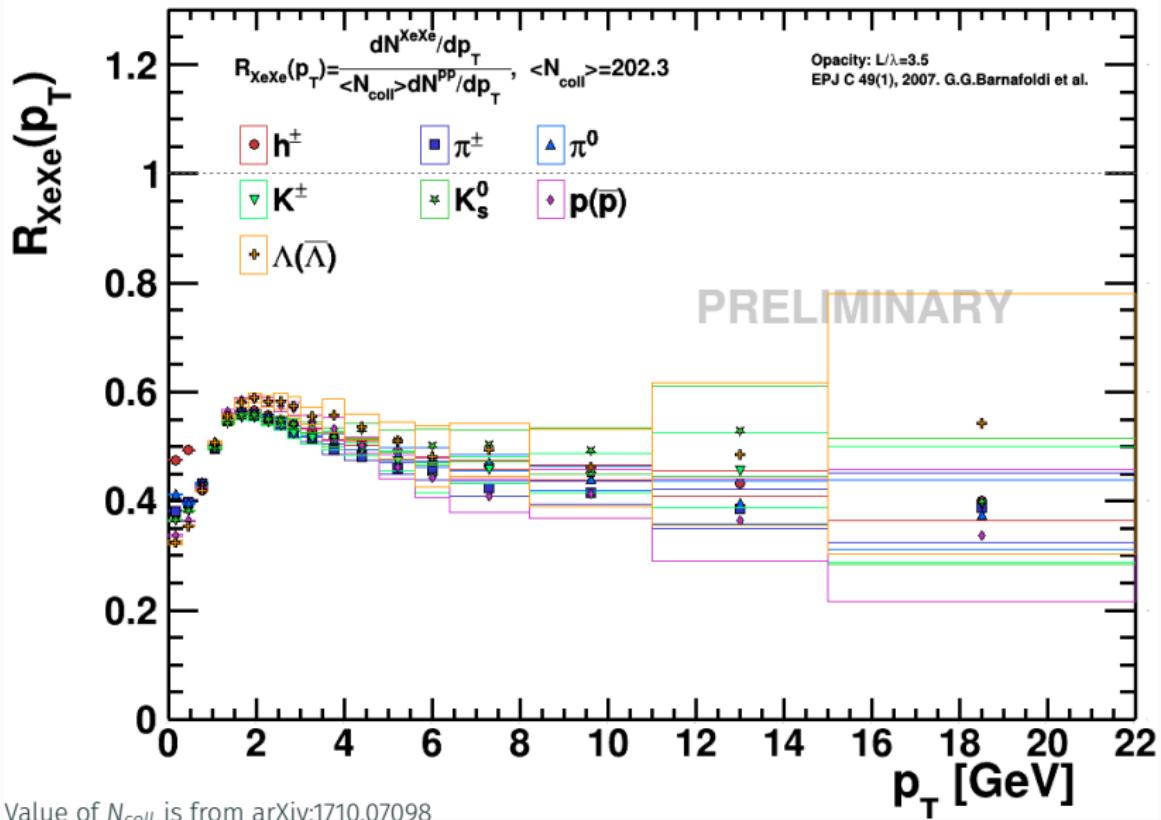
$$\text{Opacity: } L/\lambda = (0.62 \pm 0.09) \cdot \langle N_{part} \rangle^{0.33 \pm 0.03}$$



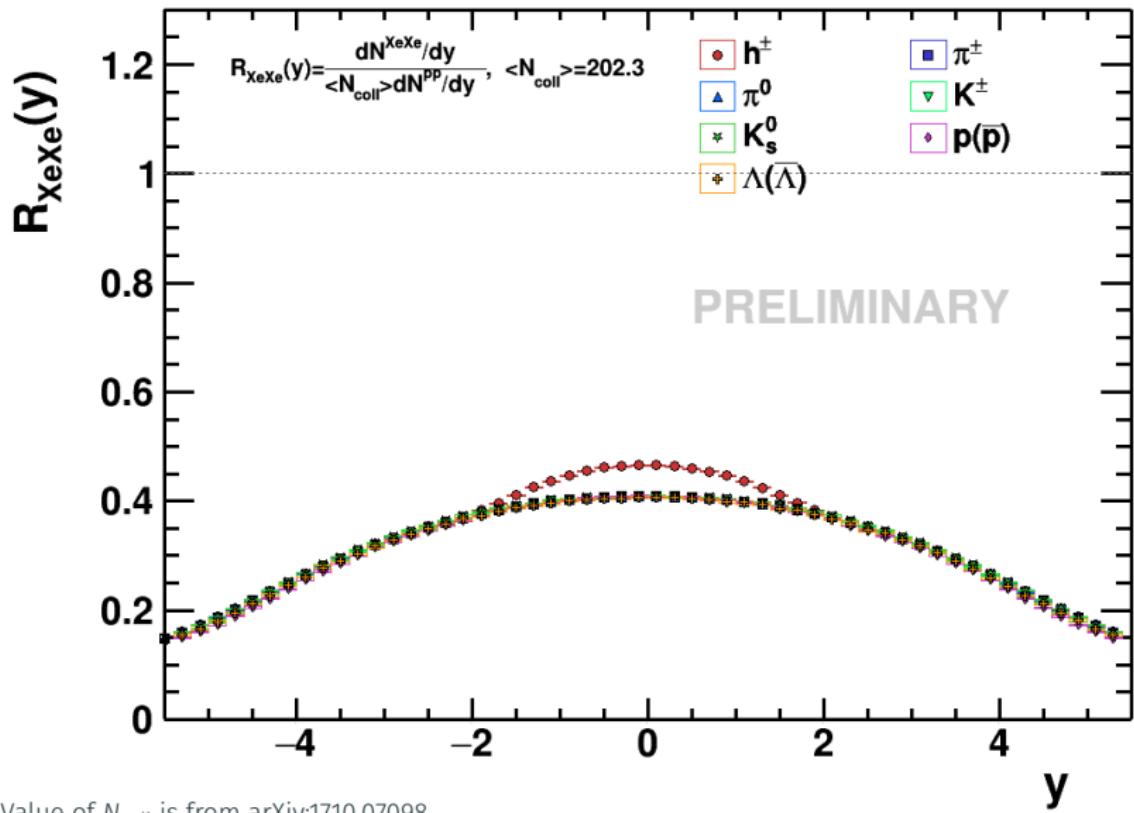
Figures are from EPJ C 49(1), 2007 (G.G.Barnaföldi *et al.*) and

Phys.Rev. C81(024909) 2010 (W.A. Horowitz *et al.*)

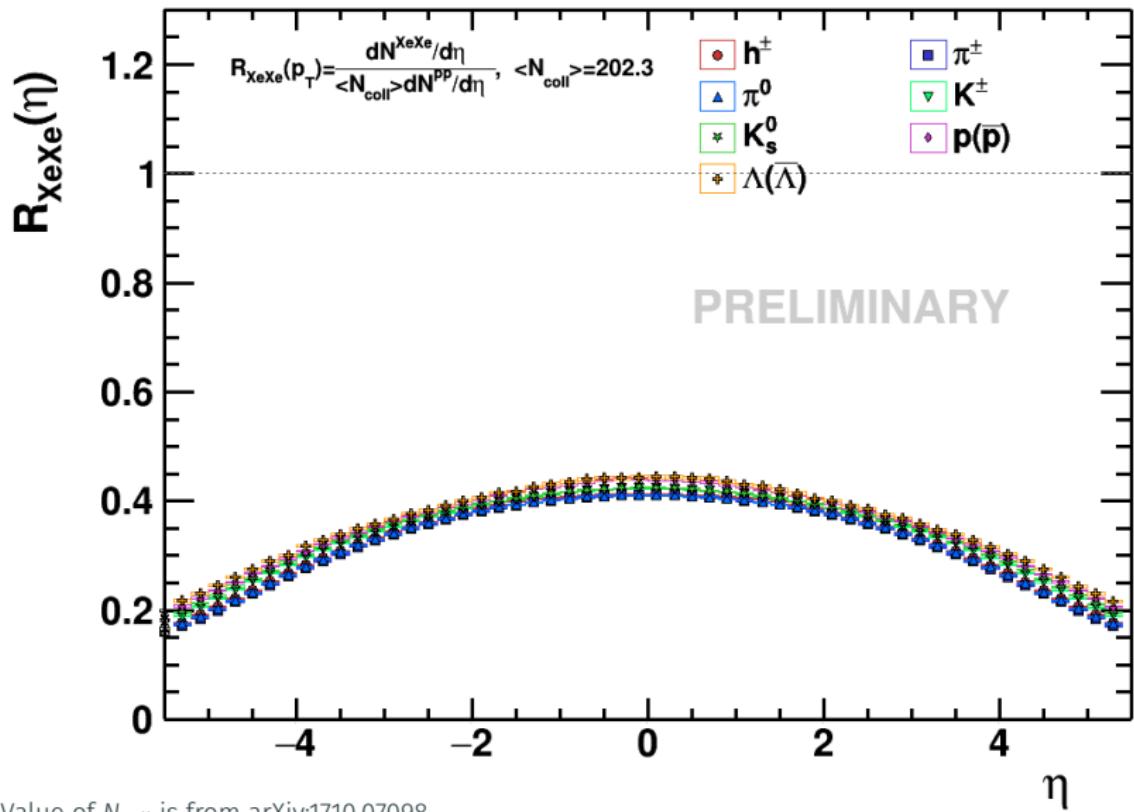
# NUCLEAR MODIFICATION FACTOR IN $p_T$



# NUCLEAR MODIFICATION FACTOR IN $y$



# NUCLEAR MODIFICATION FACTOR IN $\eta$



Value of  $N_{coll}$  is from arXiv:1710.07098

## SUMMARY

---

## SUMMARY, OUTLOOK

Work in progress: new features not covered in this talk

- New Q-dependent version of the HIJING shadowing parametrization ↗
- Gunion-Berch direct soft radiation ↗
- Jet Quenching: several models ↗

Summary

- Porting from FORTRAN to C++: ✓
- GRV98 PDF was included to LHAPDF6 for backward compatibility ✓
- Testing physics: ✓ ↗
- Testing/optimizing CPU parallel performance: ↗
- OpenCL (GPU) support: ✘
- Tsallis-motivated FF (see Á.T.'s talk on Wednesday): ↗

Thank you for your attention!