HIJING++ High-energy Jet Interaction Generator for the New Generations

Speaker: Gergely Gábor Barnaföldi, Wigner RCP of the H.A.S.

Group: GGB, G. Bíró, Sz.M. Harangozó, W.T. Deng, M. Gyulassy, G.Y. Ma,
P. Lévai, G. Papp, X.N. Wang, B.W. Zhang

HpT4LHC Workshop 2019, Knoxville, TN, USA, 17-22th March 2019

Outline

- Motivation for HIJING++
- Technical details of the HIJING++
  - The structure of the program
  - Simulation framework & new features
- New physics & tests
  - Code validation in proton-proton collisions
  - Fine-tuning
- Outlook...

MOTIVATION

A QUESTION
How long does an ‘event’ takes?

Simulation vs. data taking

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

Simulation vs. data taking

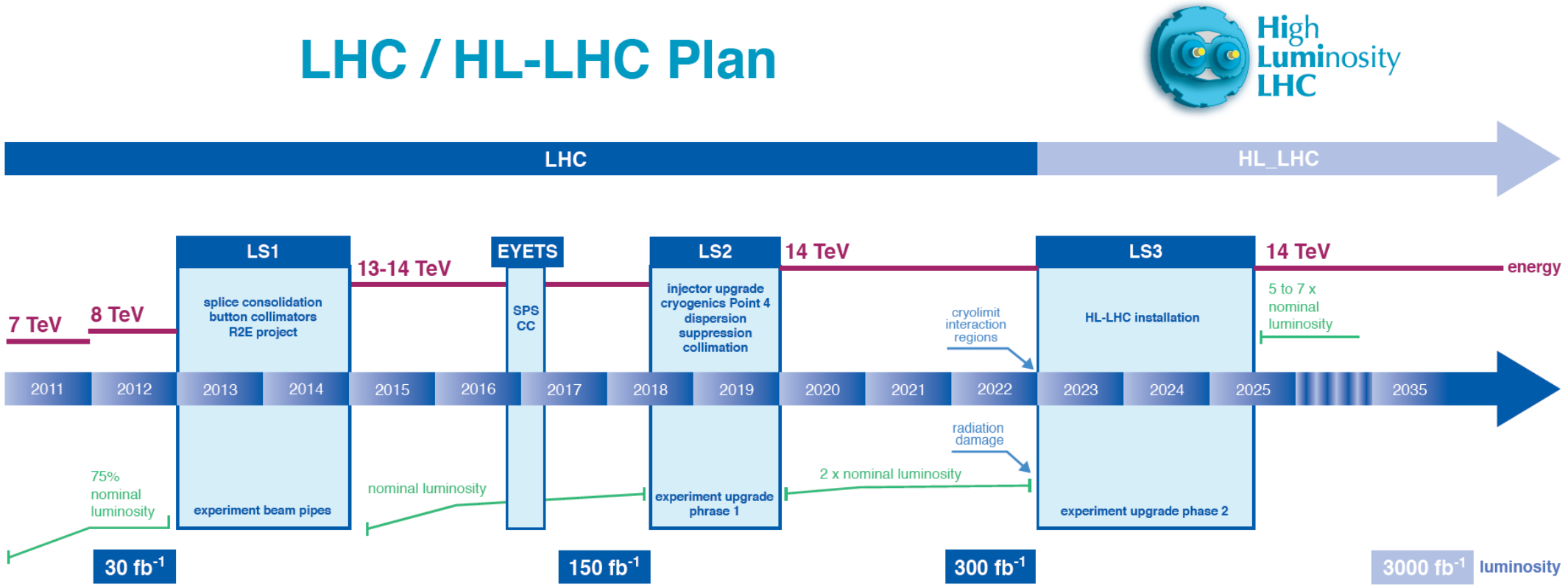
- Ideal: amount of simulated data $\approx$ real data
  - Number of events at LHC: $\mathcal{O}(10^8)$ / s
  - Necessary time for Monte Carlo with ALICE geometry: $3.8\ ms/track$

Simulation vs. data taking

- Ideal: amount of simulated data $\approx$ real data
  - Number of events at LHC: $\mathcal{O}(10^8)$ / s
  - Necessary time for Monte Carlo with ALICE geometry: $3.8\,ms/track$
- Necessary time to simulate 1 s of ALICE data: $\mathcal{O}(days)$

HI data from the Large Hadron Collider

- LHC upgrades & theories required more and faster HI simulations



HIJING++

(C++ based HIJING version 3.1 with parallel opportunities)

The HIJING++

HIJING(Heavy-Ion Jet INteraction Generator)

易經

Bagua (eight simbols)

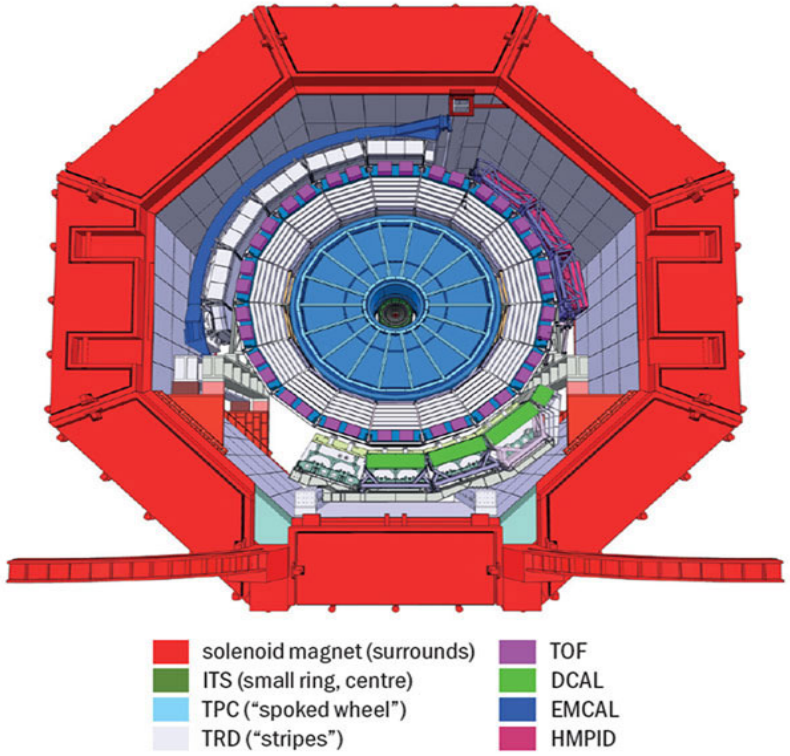
fundamental principles of reality

adjoint representation 8 of *SU(3)*

The HIJING++

HIJING(Heavy-Ion Jet INteraction Generator)

易經



Bagua (eight simbols)

fundamental principles of reality

adjoint representation 8 of *SU(3)*

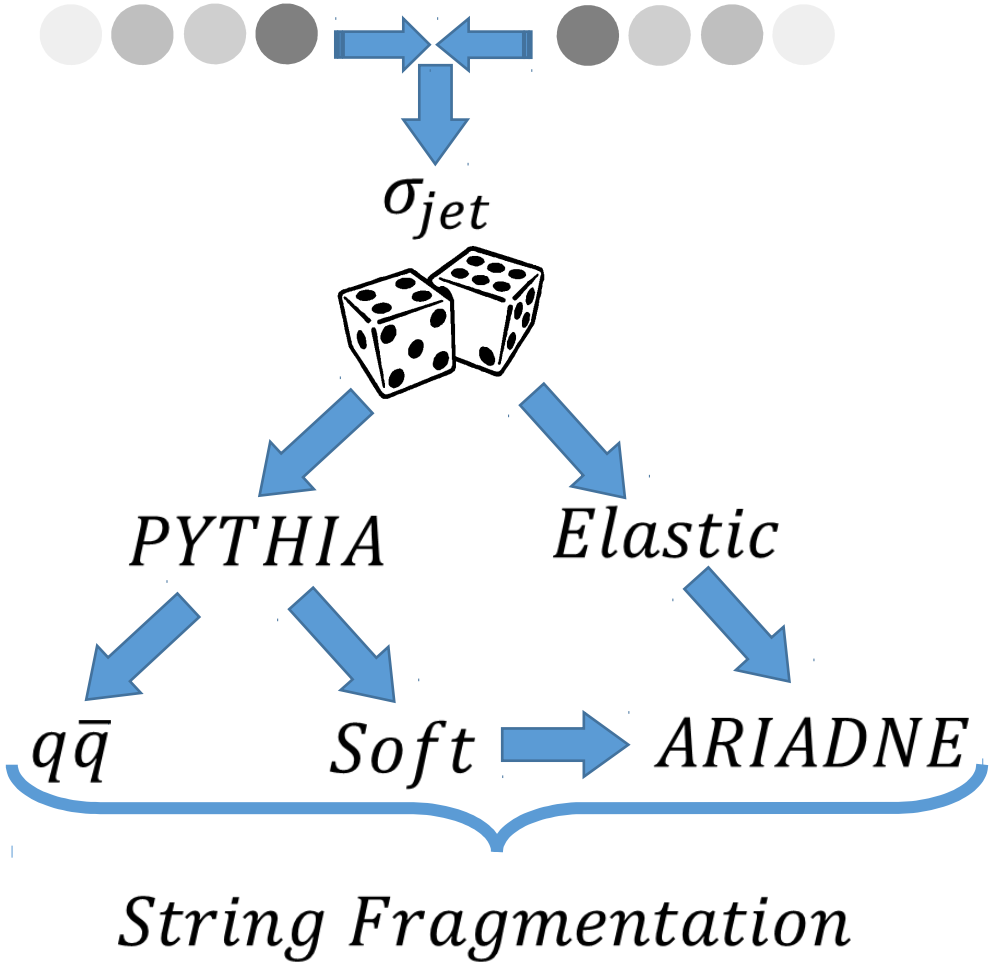
The HIJING++

- is a framework, not a black box.
- ...is not a direct port of the old FORTRAN code.
- ...is a direct port of the old FORTRAN code after all

  (regarding the physics).
- ...is not wrapper for Pythia8.
- ...is not published (yet).

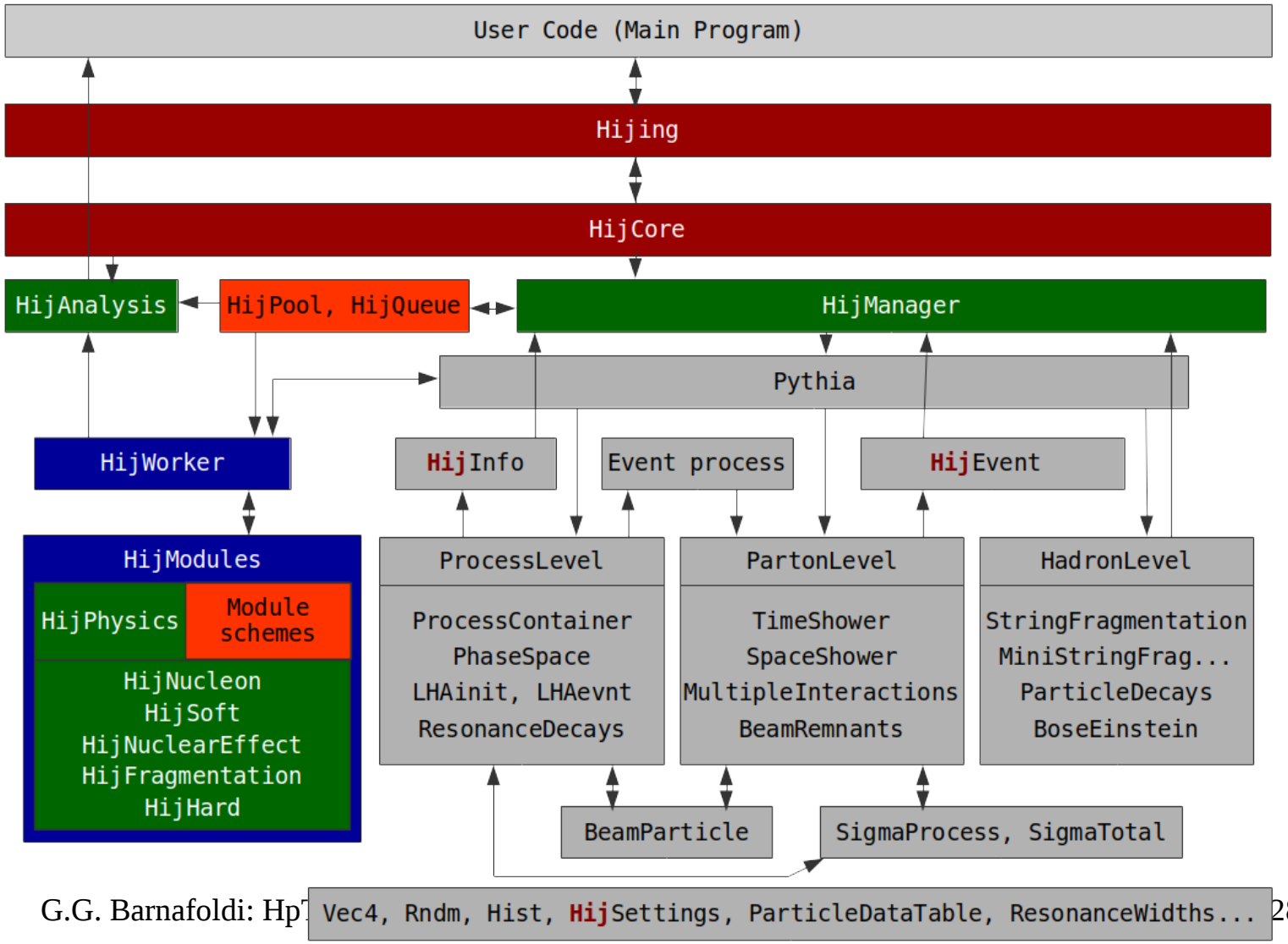
Program Flow – in general

- Pair-by-pair nucleon-nucleon events
- Multiple soft gluon exchanges between valence- and di-quarks
- String hadronization according to Lund fragmentation scheme



Program Structure – HIJING 3.1

- Pythia8 namespace containers
- Structure similarities
- Actual program flow is more complicated
- New: HijManager



Program Structure
Hijing class

```
namespace Pythia8 {

    class Hijing {

    public:

        Info            info;
        Rndm            rndm;
        Settings        settings;
        ...

    private:

        HardCollision   hijhard;

        SoftScatter     hijsoft;

        Fragmentation   fragmentation;

        NucleonLevel    nucleonlevel;

        ...
    }
}
```

- Processes ordered in class hierarchy
- Former common blocks ⎵ class variables
- Processes called through object functions

```
// Class for handling the hard collisions

// Class for handling the soft interactions

// Class for handling the Lund string fragmentation

// Class for  the nuclear effects
```

The 'main' example

Usual form kept for regular users

Form also similar to Pythia 8.x

FORTRAN

```
PROGRAM TEST
...
PARM(1) = 'DEFAULT'
VALUE(1) = 80060
CALL PDFSET(PARM, VALUE)
CALL GetDesc()
...

CALL HIJSET(EFRM, FRAME, PROJ, TARG, IAP, IZP, IAT, IZT)

N_EVENT=1E6
DO 200 IE = 1, N_EVENT
    CALL HIJING(FRAME, BMIN, BMAX)
200 CONTINUE

STOP
END
```

C++

```
#include "Hijing.h"

using namespace Pythia8;

int main() {
    Hijing hijing("../xmldoc", true);
    hijing.readString("PDF:pSet = LHAPDF6:GRV98lo");

    bool okay = hijing.init(200.0, frame,
                "A", "A", 197, 79, 197, 79);
    if (!okay) return 1;

    int MaxEvent = 1e6;
    for (int iEvent = 0; iEvent < MaxEvent; ++iEvent)
        hijing.next(frame, 0.0, 0.0);
}
```

Program Features

- Calculation by improved models
- Pythia like prompt Histogram creation
- CPU level Parallel computing


```
const std::size_t num_threads = std::thread::hardware_concurrency();
for (std::size_t i = 0u; i < num_threads; ++i){
    async_hijing.at(i) = std::unique_ptr<Hijing>(new Hijing);
}
for (std::size_t I = 0; I < num_threads; ++I){
    ...async run...
    okay[I] = async_hijing[I]->init(...);
    for (int iEvent = 0; iEvent < numEvent; ++iEvent)
        async_hijing[I]->next(...);
    for (int i = 0; i < async_hijing[I]->event.size(); ++i)
        if(...) hist[I]->fill(...);
}
```

- MCNet2: RIVET, YODA compatibility

Dependencies & External packages

- Boost

  sudo apt-get install libboost-all-dev

- LHAPDF 6

  ./configure –prefix=$HOME/…/share/LHAPDF

  make all

  insert downloaded PDF library to $HOME/…/share/LHAPDF

  optionally modify **pdfsets.index**, add set if needed

  export LD LIBRARY PATH=<library path>

- Pythia 8

  *./configure --with-lhapdf6-lib=$HOME/…/lib *
  --with-boost-lib=/usr/lib/x86_64-linux-gnu
  make –j4

- GSL (optional)

  HIJING **make** option

HIJING vs. HIJING++

	FORTRAN HIJING	HIJING++:
Precision	single	double
Pythia version	5.3*	8.2+**
PDF	GRV98lo	LHAPDF6.2+
Colour reconnection	✗	✓
Jet quenching	(✓)	(✓)
Multithreading	✗	✓
Analysis interface	✗	✓***
Module management**	✗	✓

*Was modified and hardwired into HIJING

**Default tune for HIJING++ is Monash, for that re-tuning of the parameters is needed

***Includes: simple ascii, ROOT and HepMC2 (Rivet)

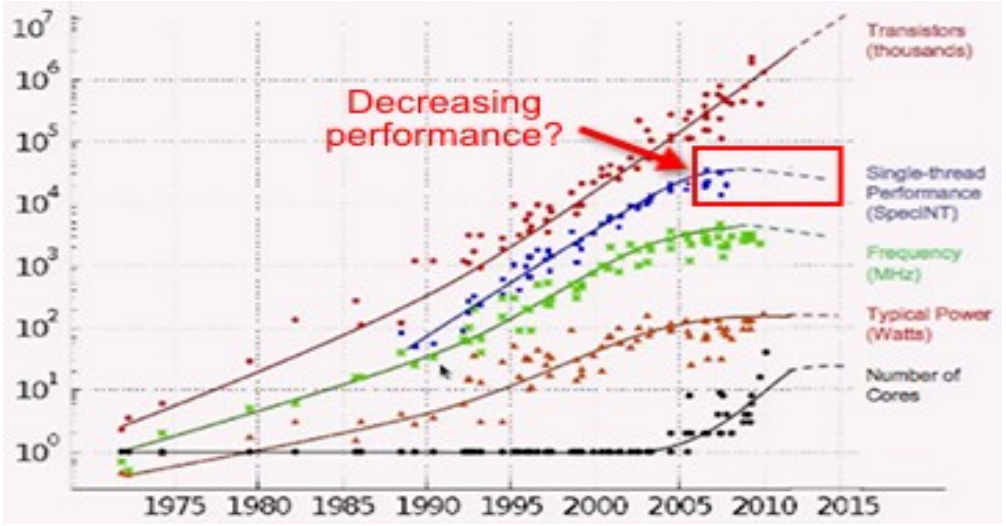
****In Backup

Performance tests with HIJING++

Fast computing = parallel computing

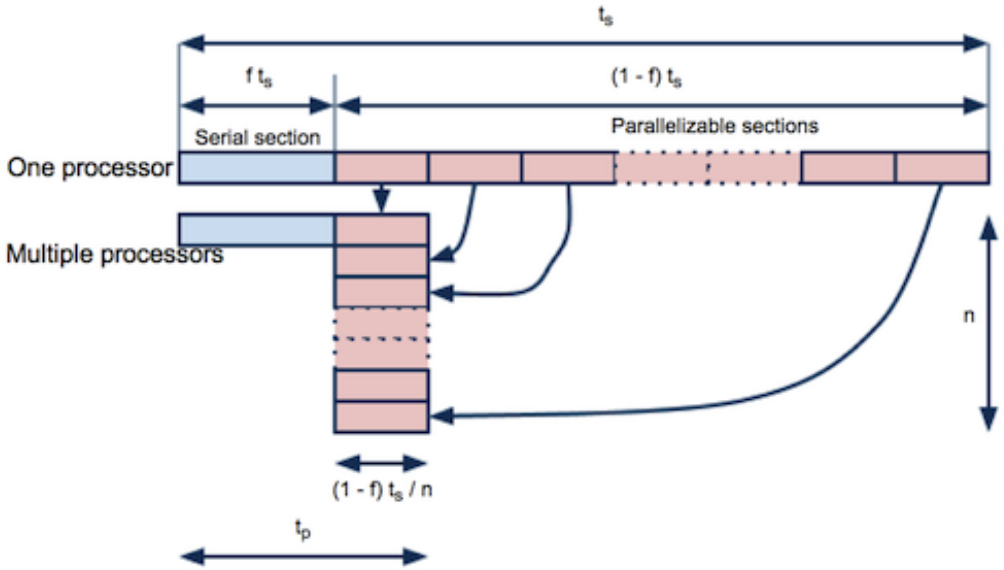
- Moore's law:

Every 2nd year the number of transistors (integrated circuits) are doubled in computing hardwares.



- Amdalh's law:

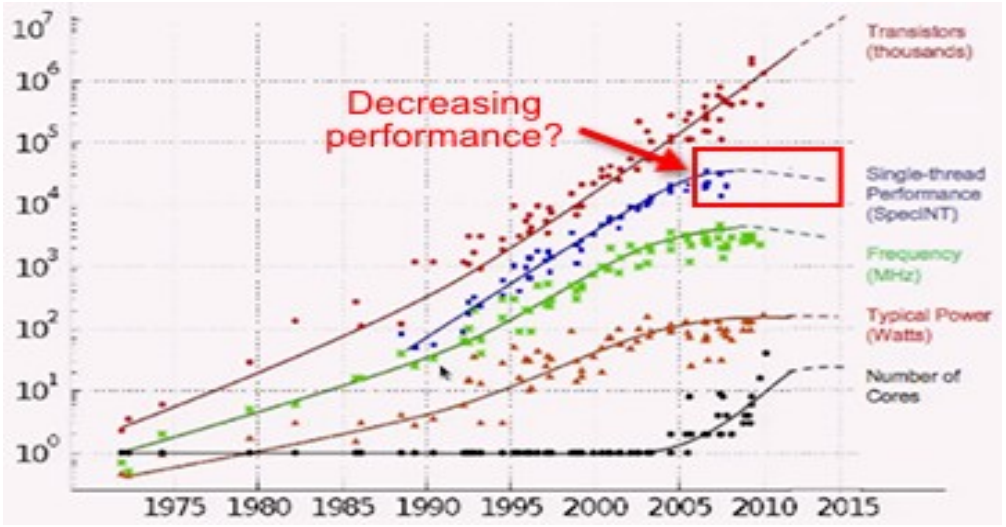
The theoretical speedup is given by the portion of parallelizable program, p, & number of processors, N, is:



Fast computing = parallel computing

- Moore's law:

Every 2nd year the number of transistors (integrated circuits) are doubled in computing hardwares.



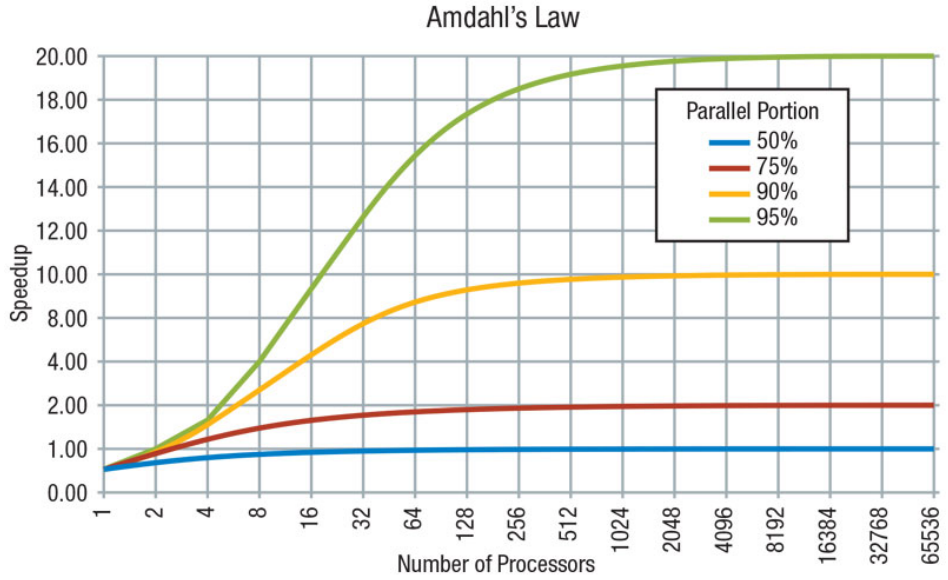
- Amdalh's law:

The theoretical speedup is given by the portion of parallelizable program, p, & number of processors, N, is:

$$\text{Speedup}(N) = \frac{1}{(1-P)+\frac{P}{N}}$$

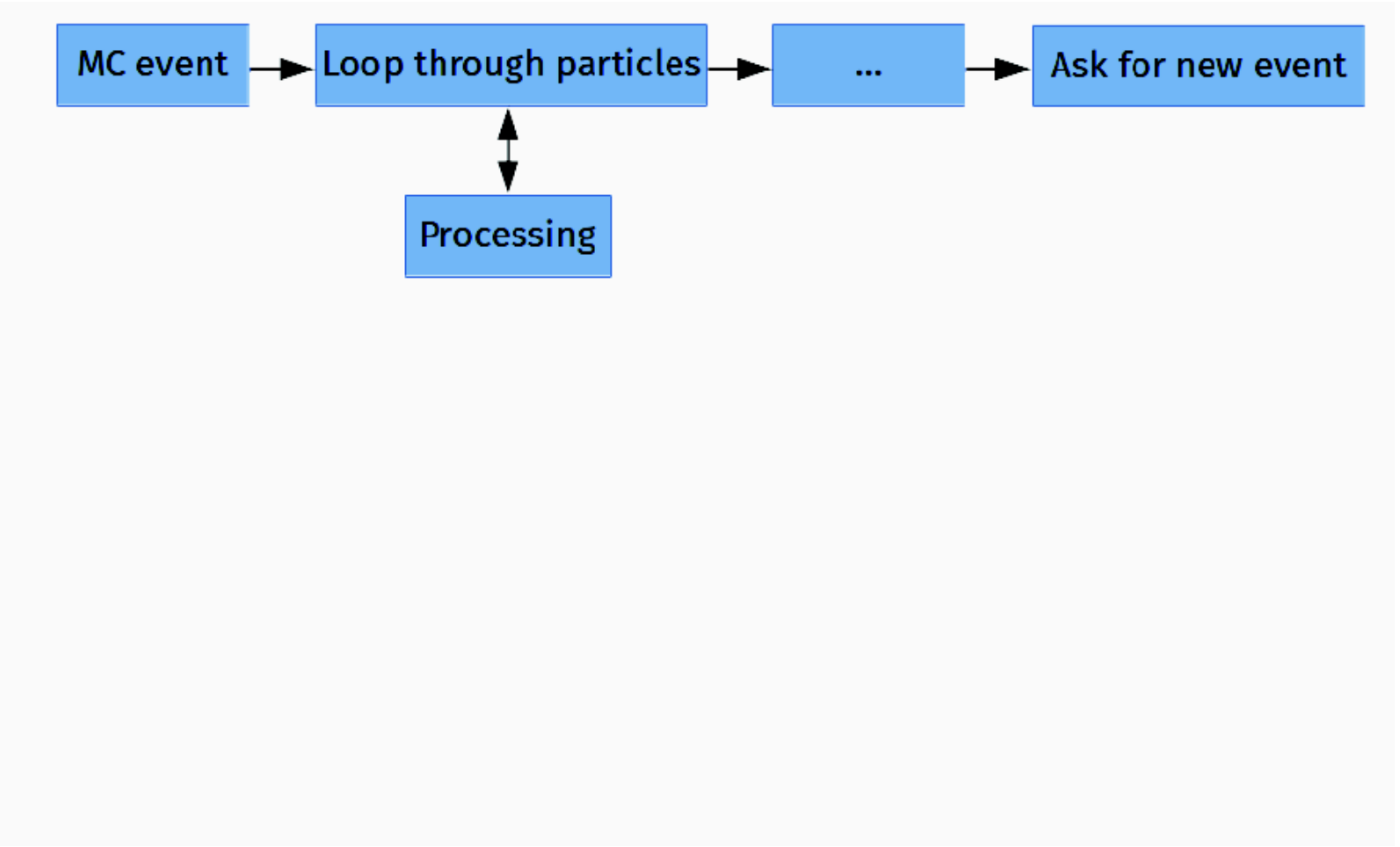
Serial part of job = 1 (100%) - Parallel part

Parallel part is divided up by N workers



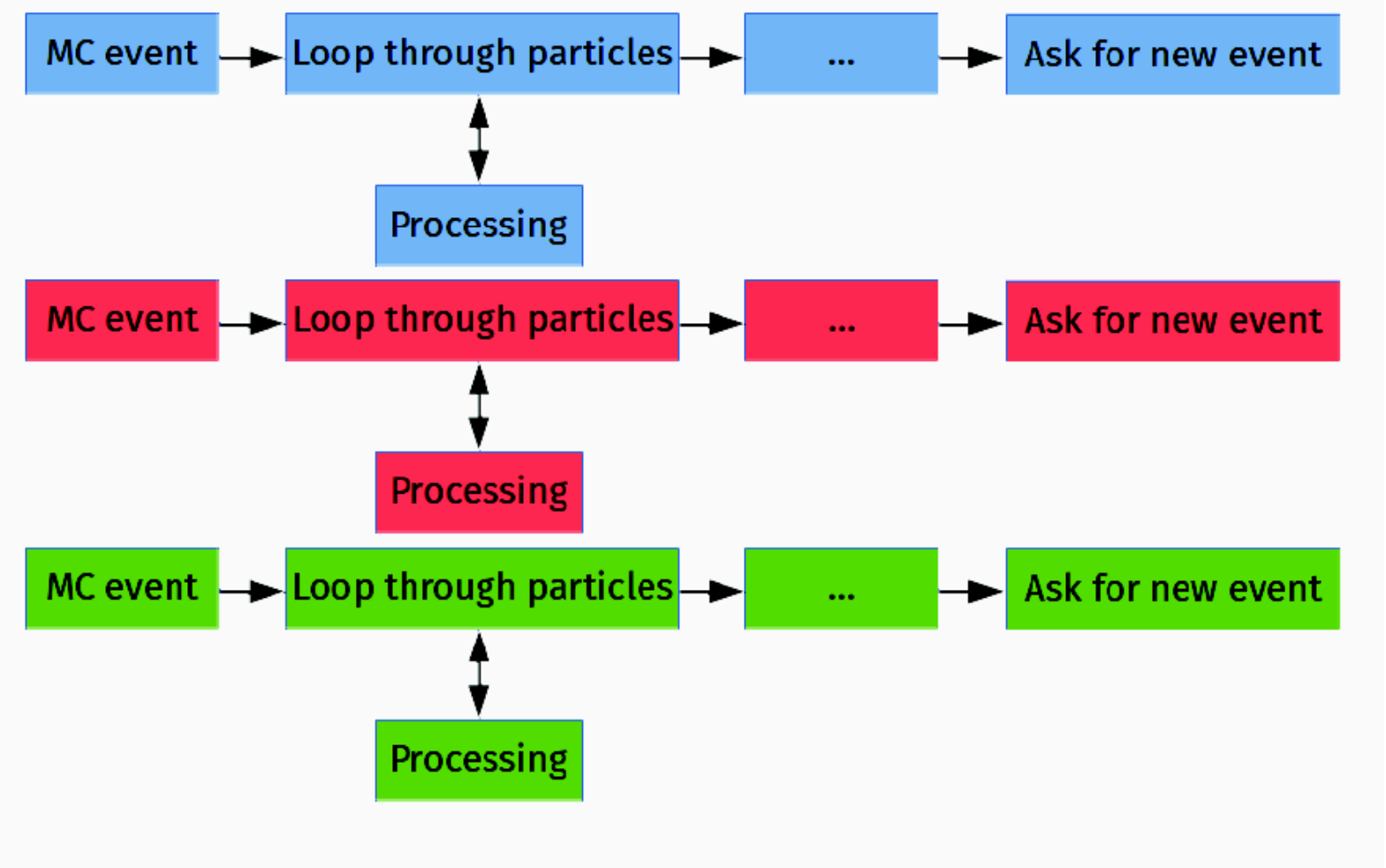
Multi-thread features

What is in the DO LOOP?



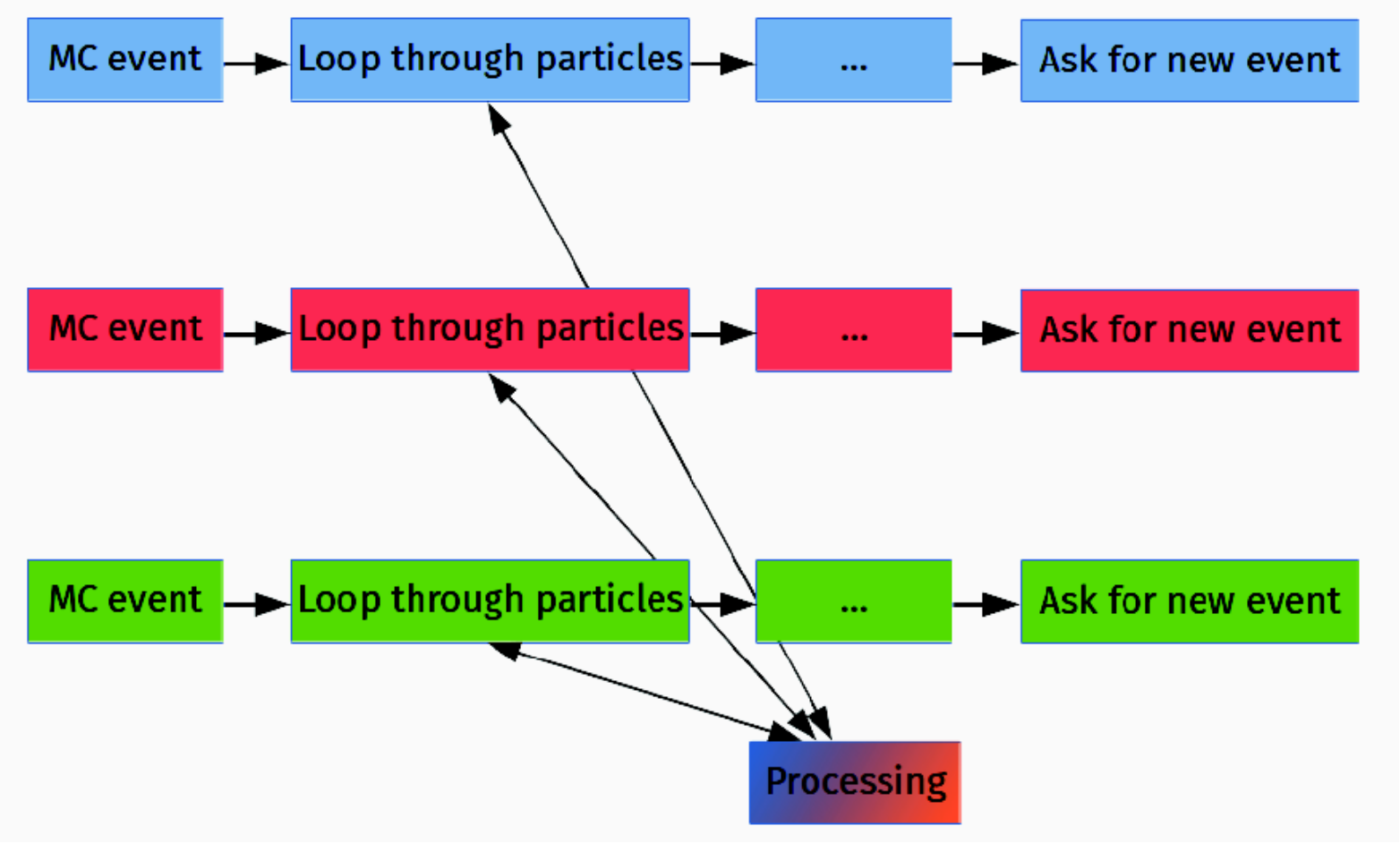
Multi-thread features

What is ongoing in a “mass” production of using MC in data analysis?



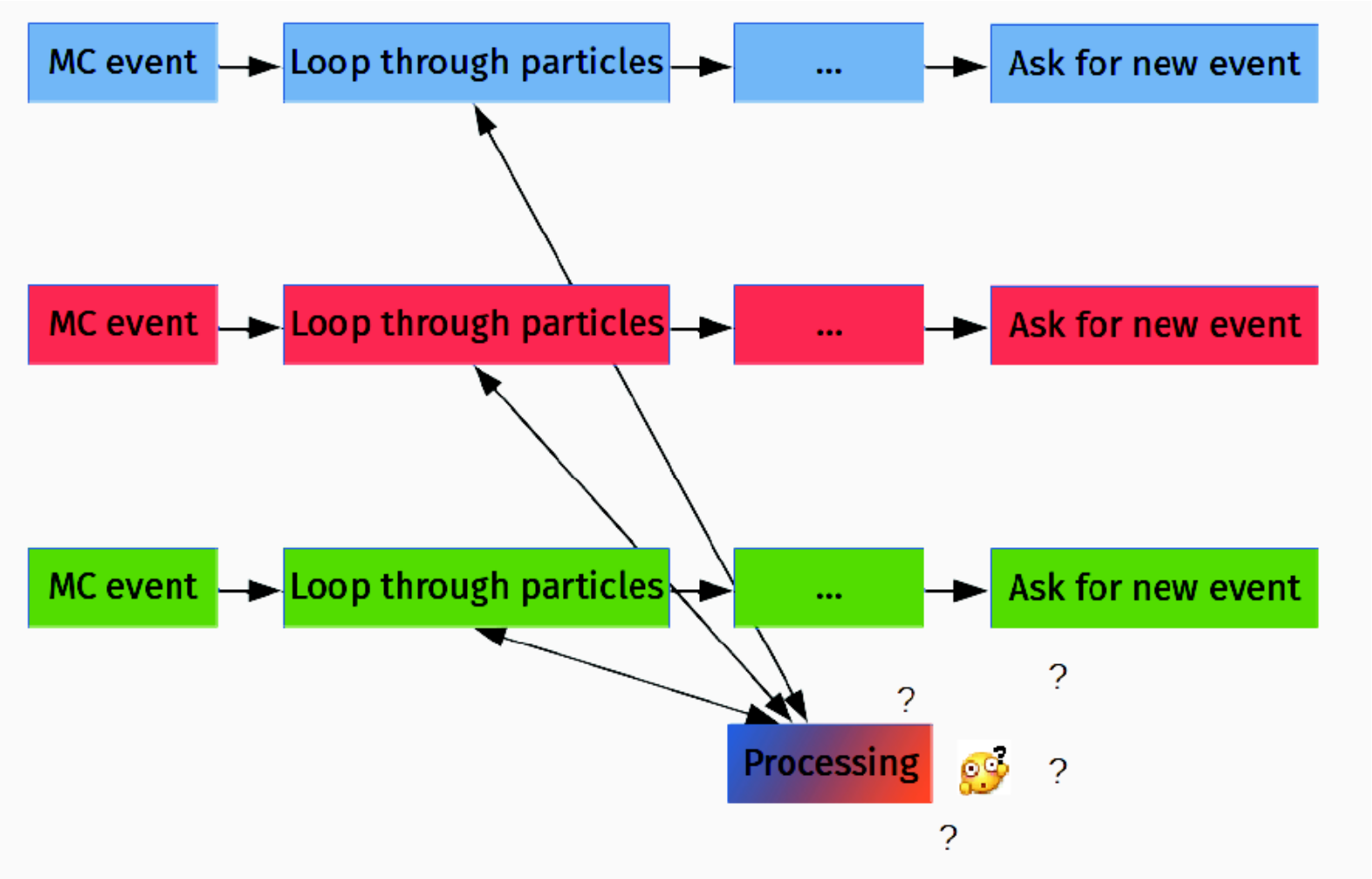
Multi-thread features

What is ongoing in a “mass” production of using MC in data analysis?



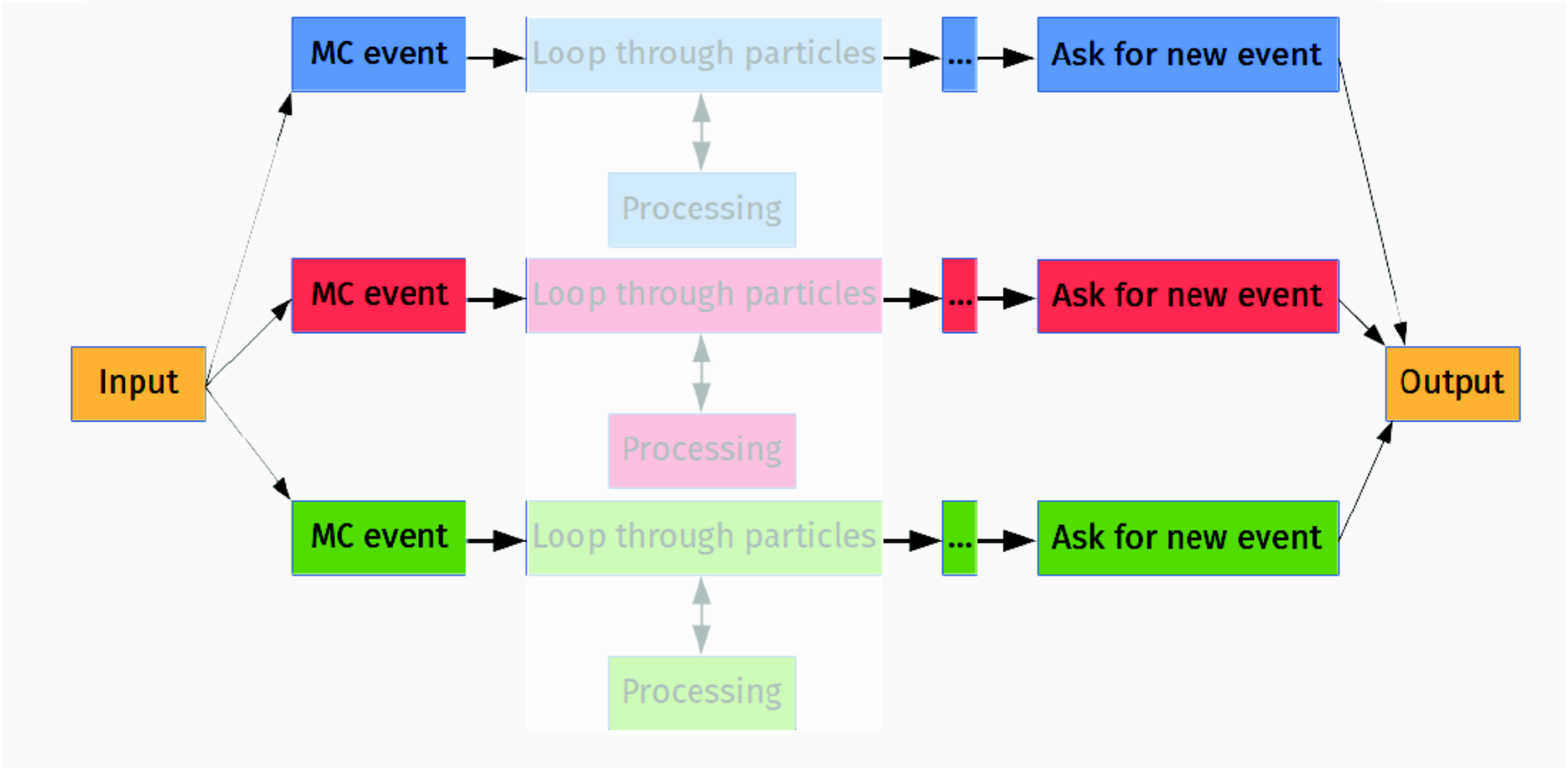
Multi-thread features

What is ongoing in a “mass” production of using MC in data analysis?



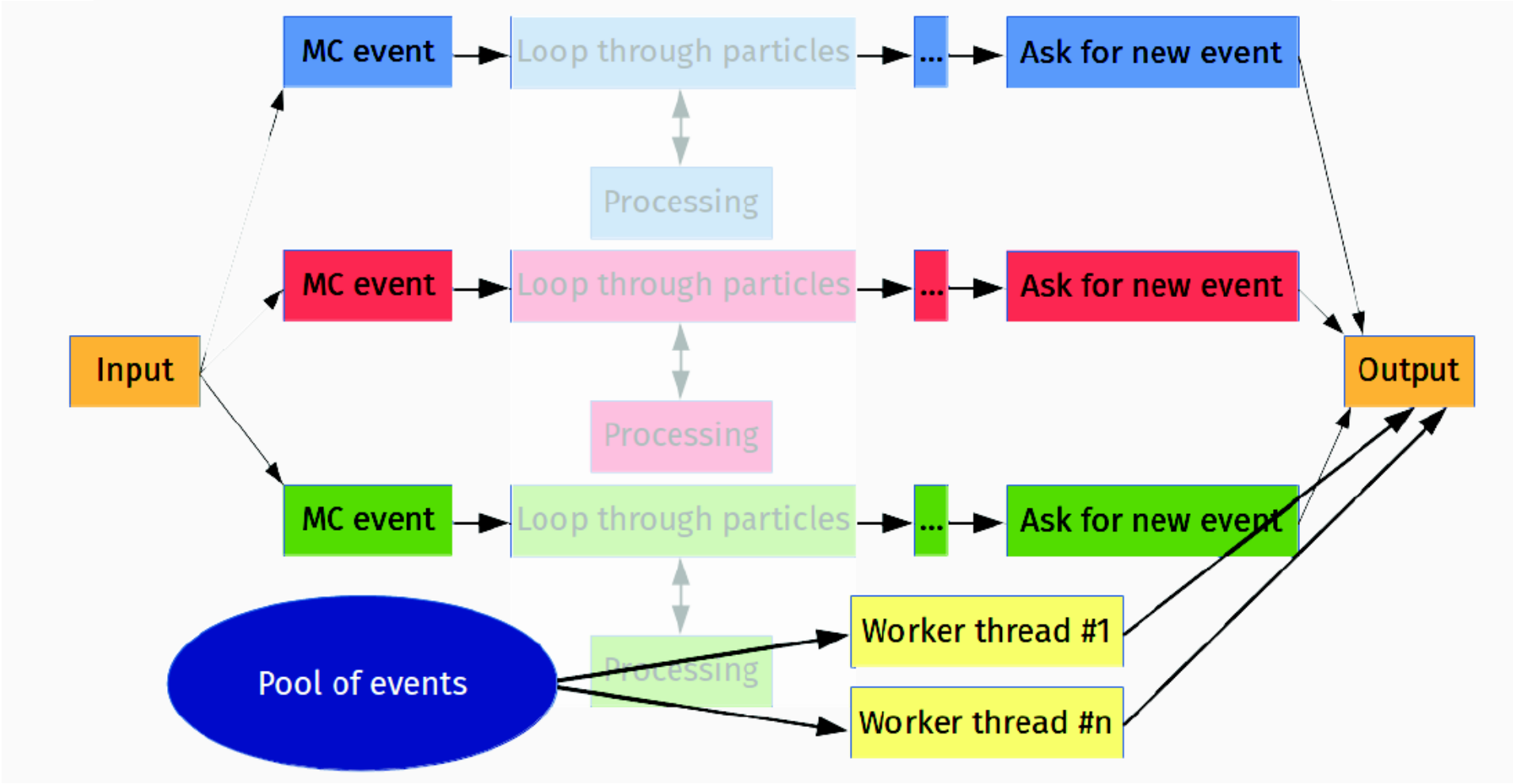
Multi-thread features

Multi-threading is not just running the same code multiple….



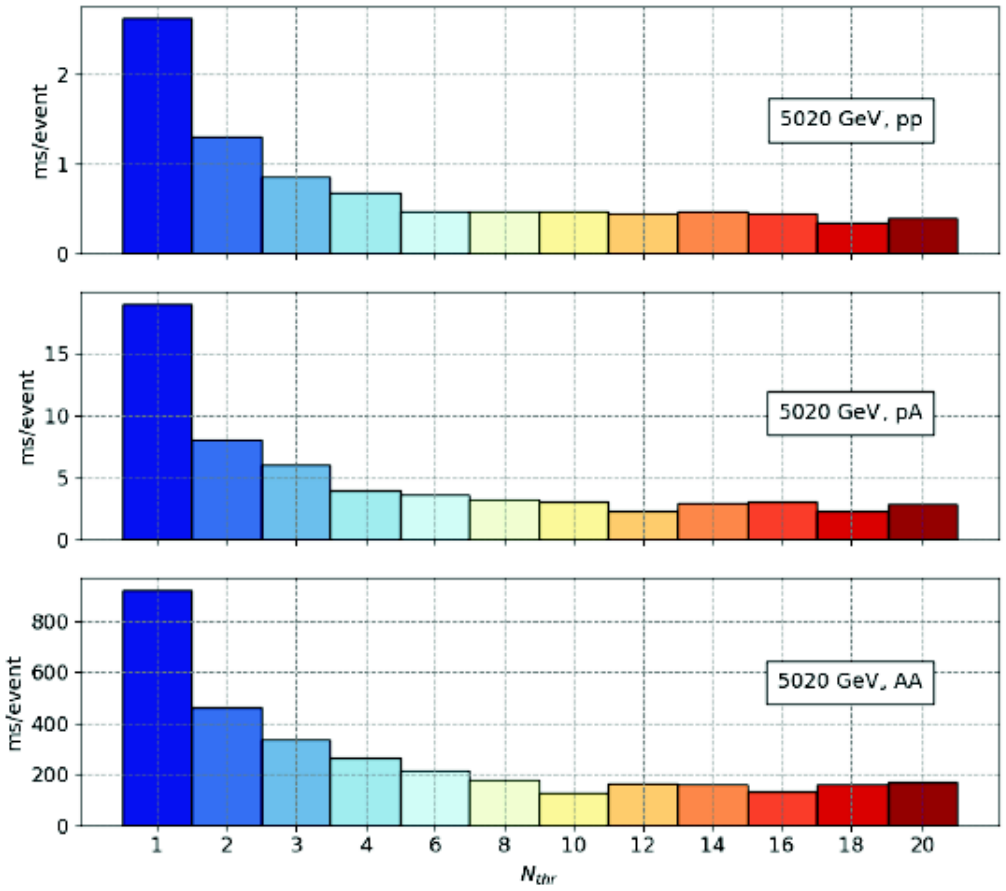
Multi-thread features

What is ongoing in a “mass” production of using MC in data analys



Multi-thread features

How much does a pp/pA/AA collision event cost in time?



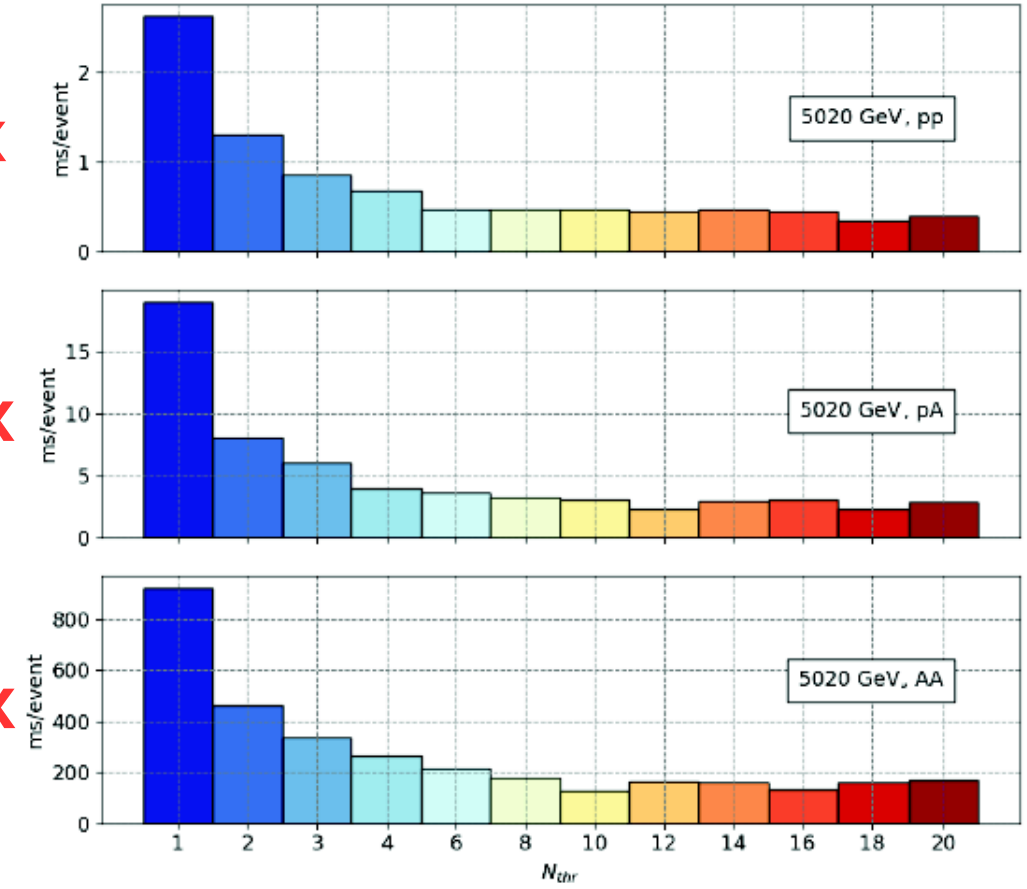
Multi-thread features

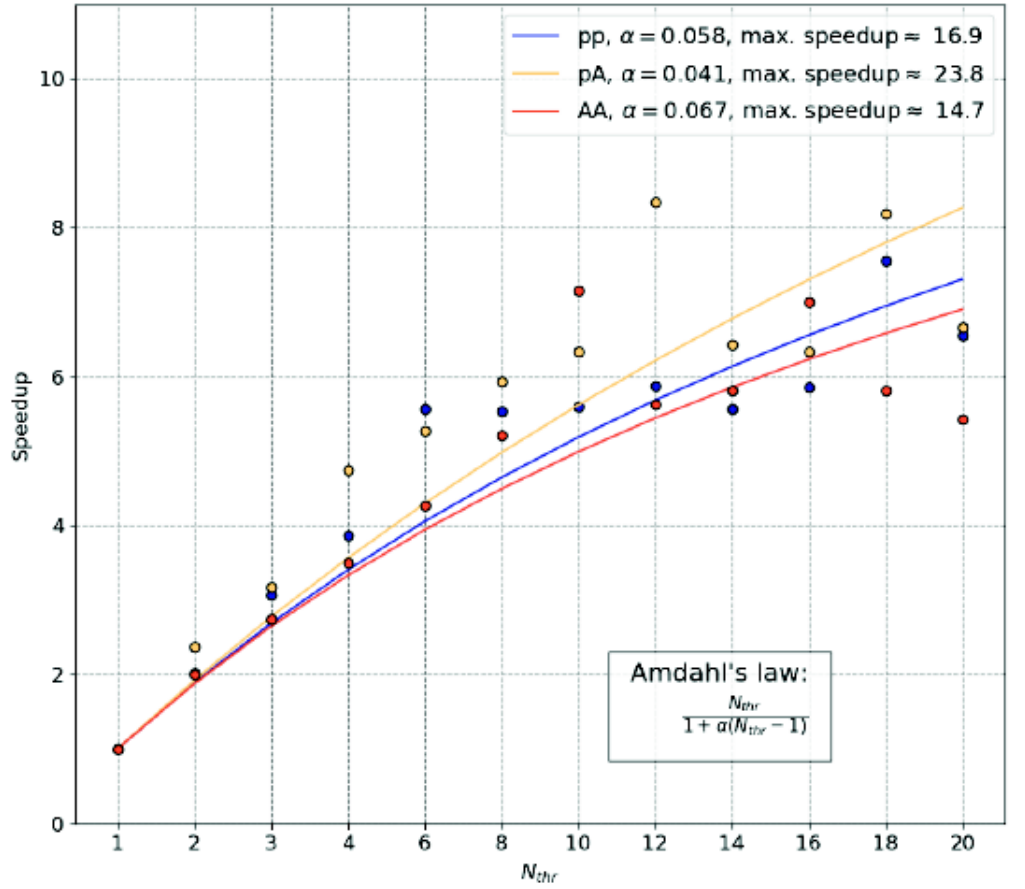
How much does a pp/pA/AA collision event cost in time?

pp: 17x

pA: 24x

AA: 15x



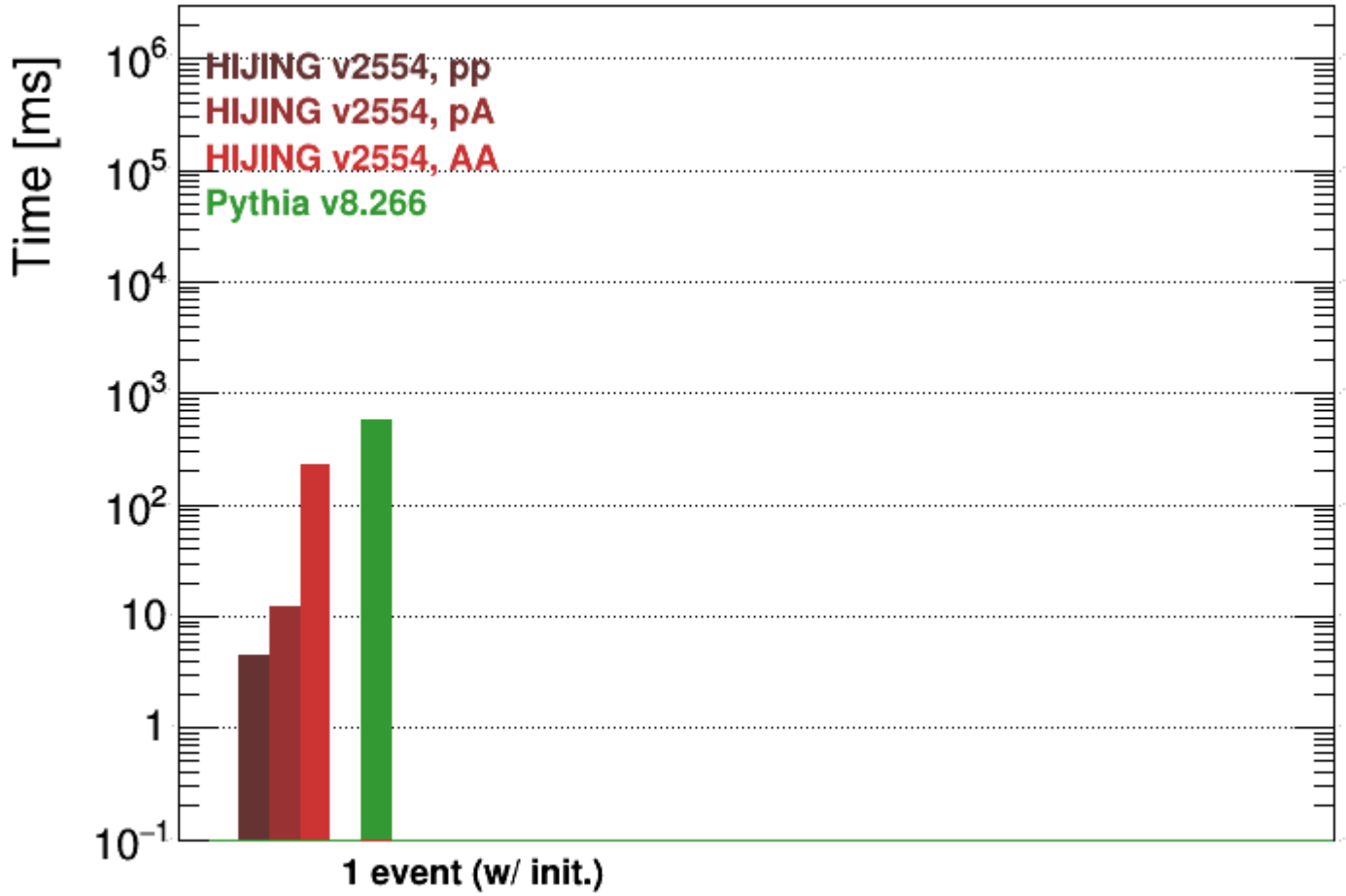


Performance tests: runtime

- Runtime new vs. old

  Single core run & 1 event:

  - Old HIJING pp is faster, than PYTHIA8, but less physics

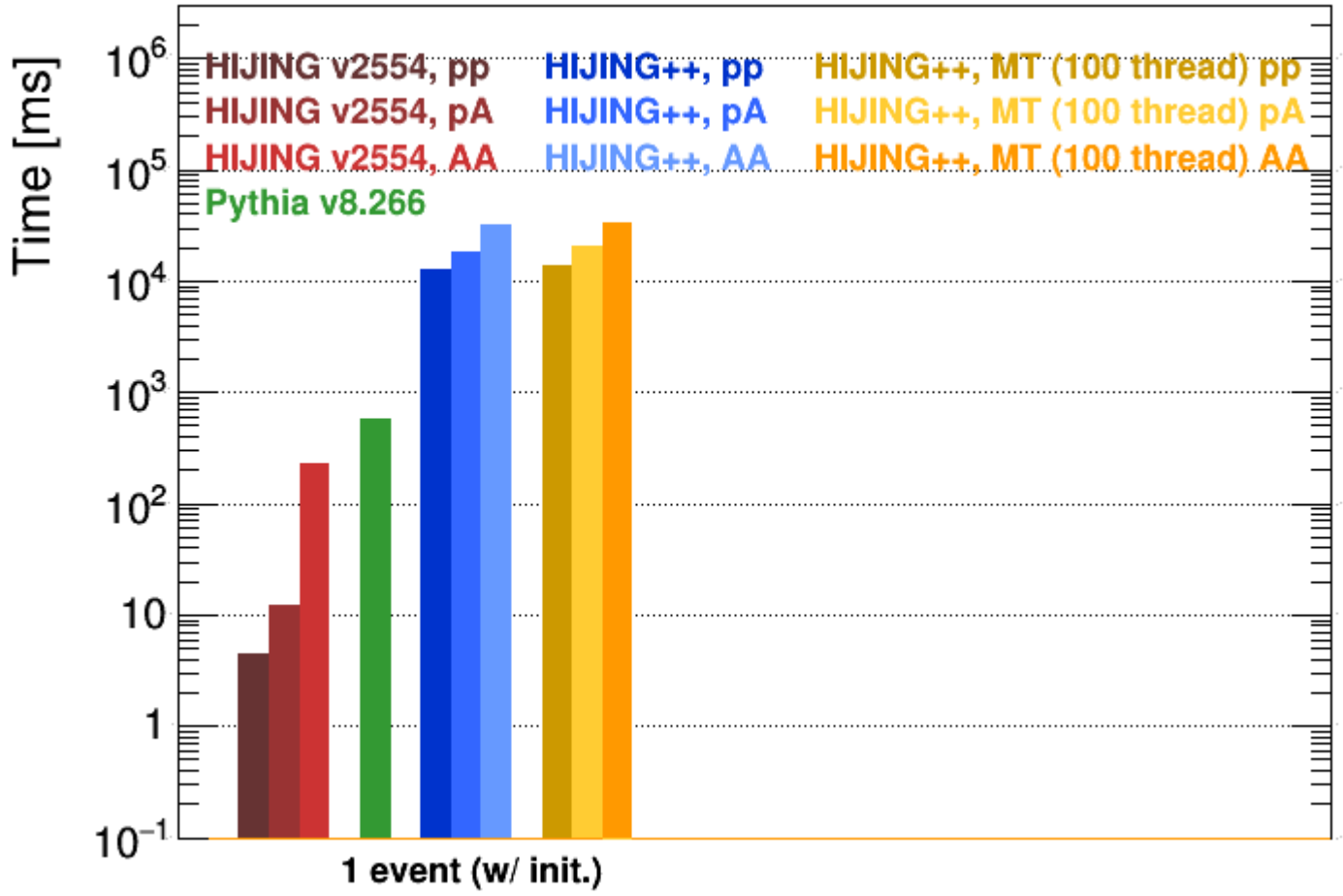


Performance tests: runtime

- Runtime new vs. old

  Single core run & 1 event:

  - Old HIJING pp is faster, than PYTHIA8, but less physics
  - HIJING++ pp is slower, than PYTHIA8: this is the effect of minijets + nuclear effects
  - Init: is longer for HIJING++



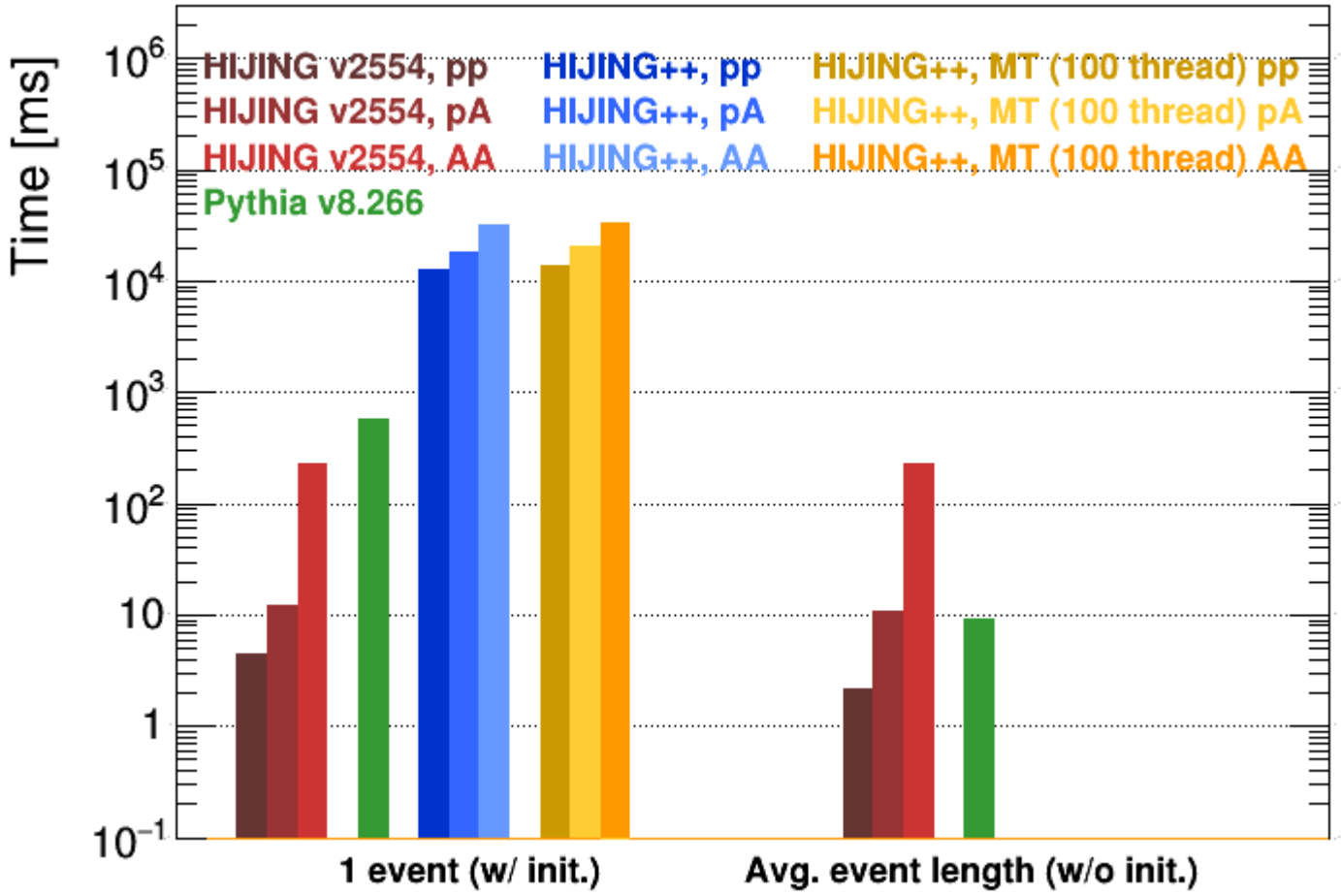
Performance tests: runtime

- Runtime new vs. old

  Single core run & 1 event:

  - Old HIJING pp is faster, than PYTHIA8, but less physics
  - HIJING++ pp is slower, than PYTHIA8: this is the effect of minijets + nuclear effects
  - Init: is longer for HIJING++

  Multi- event & multi-core run:



Performance tests: runtime

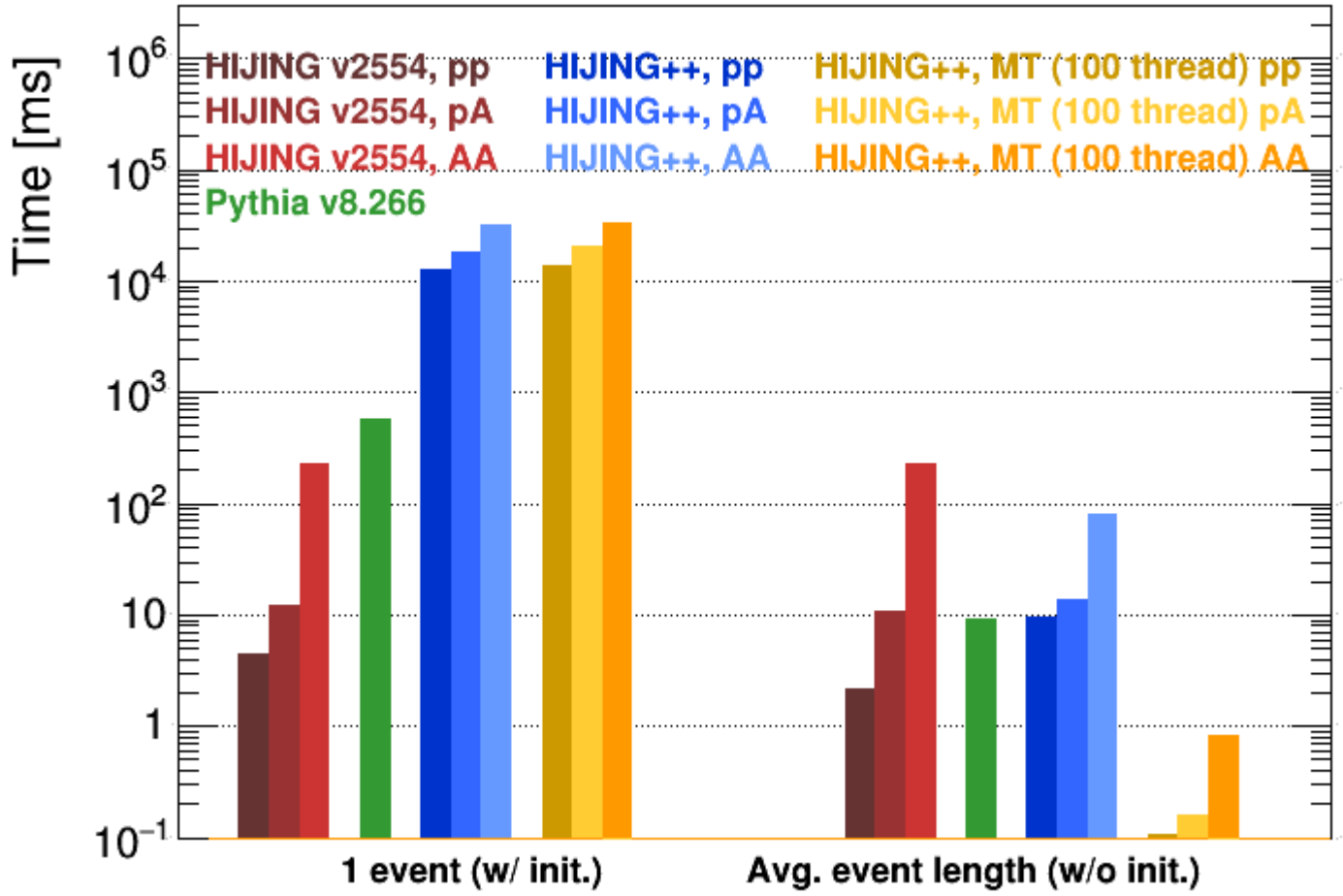
- Runtime new vs. old

  Single core run & 1 event:

  - Old HIJING pp is faster, than PYTHIA8, but less physics
  - HIJING++ pp is slower, than PYTHIA8: this is the effect of minijets + nuclear effects
  - Init: is longer for HIJING++

  Multi- event & multi-core run:

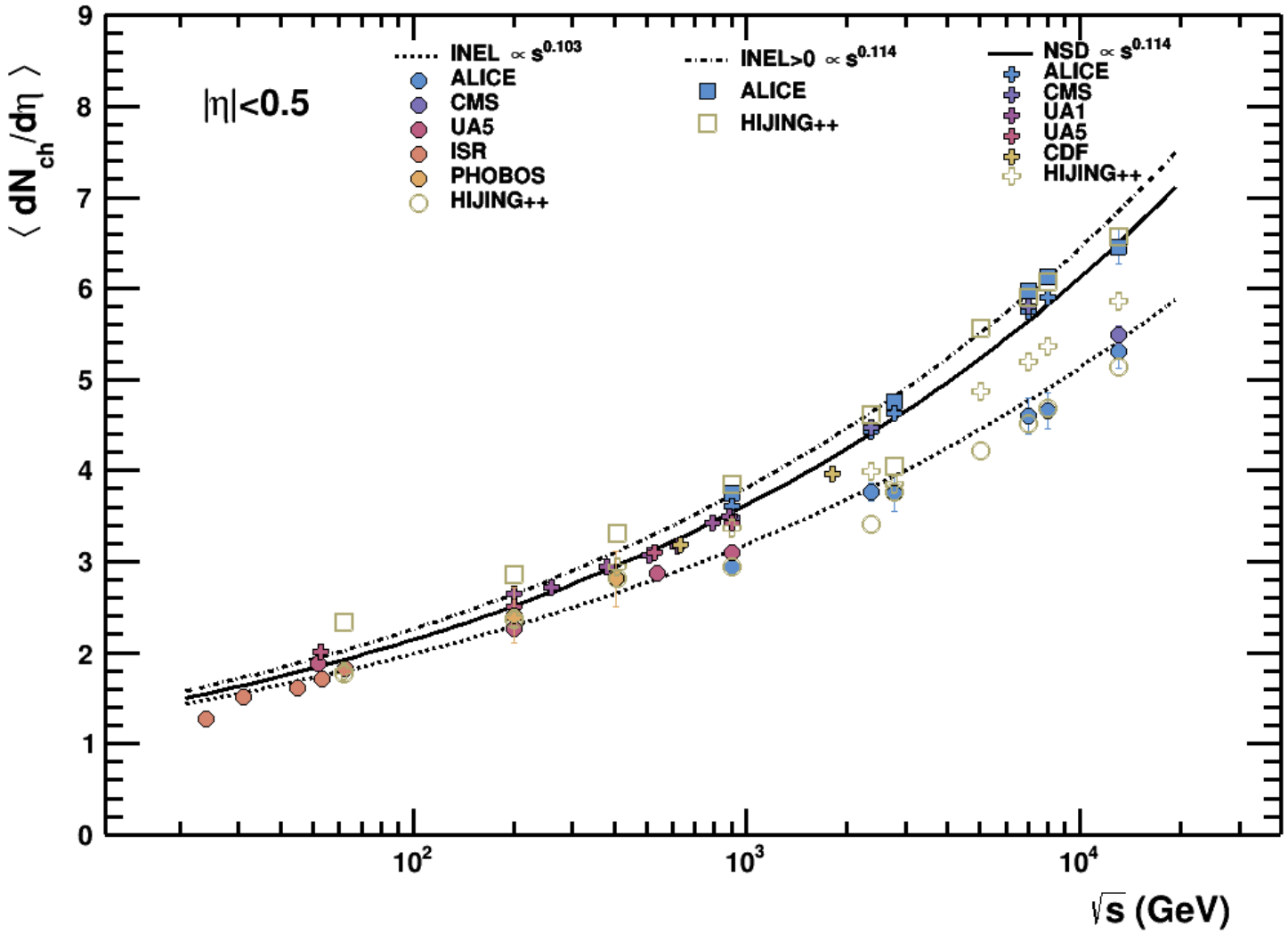
  - Due to the MPI support several times faster
  - Better performance in HIC than in small systems (100 evts)



Physics tests with HIJING++

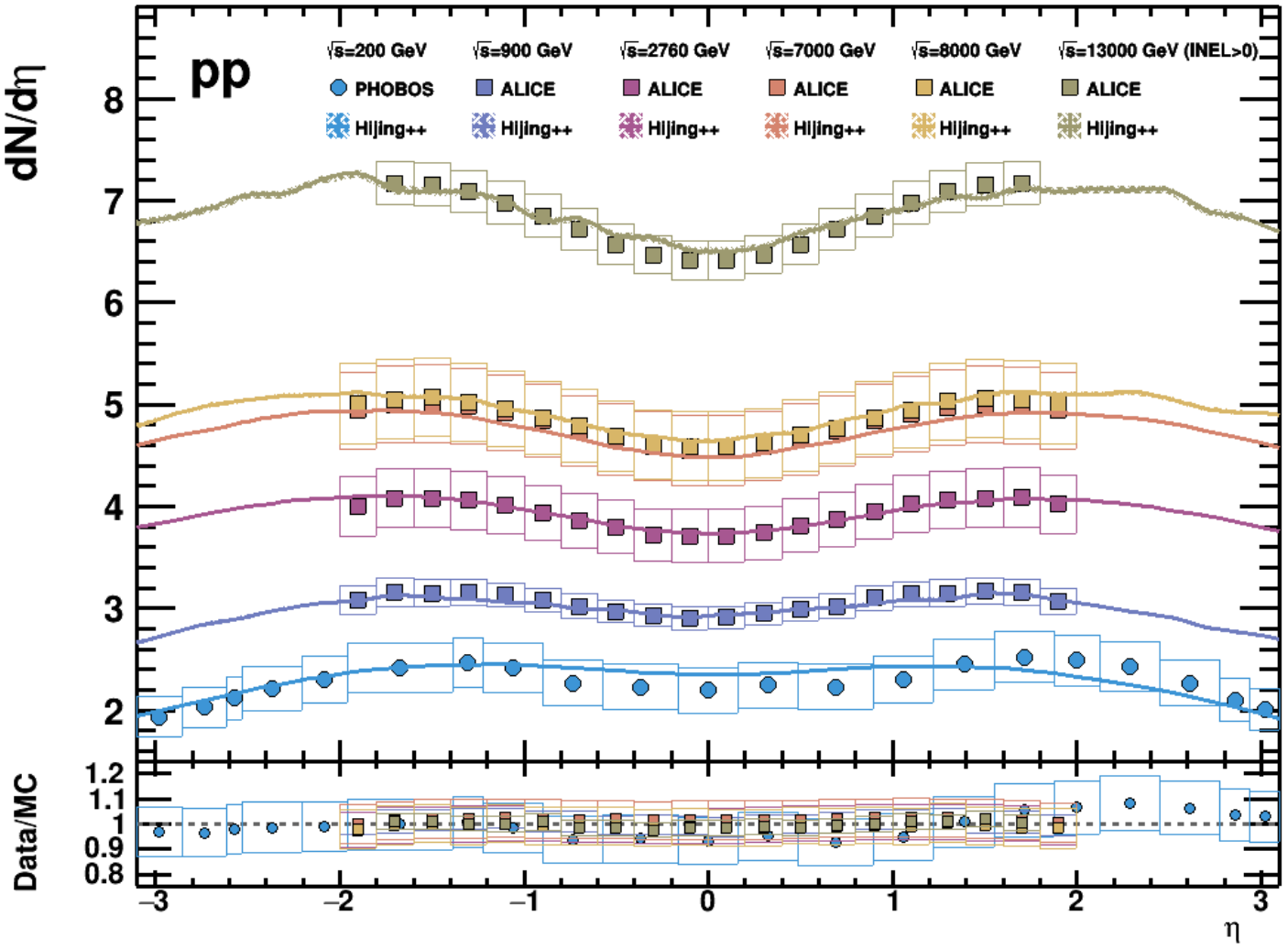
Physics tests: global observables in pp

- Total ch. multiplicity
  - All pp data in a wide center of mass energy range 10 GeV to 13 TeV
  - HIJING++ ch. multiplicity trend is similar than the data



Physics tests: global observables in pp

- (Pseudo) rapidity
  - pp data 200 GeV - 13 TeV
  - In this set PHOBOS & ALICE
  - Perfect agreement up to 5-10% in wide pseudo-rapidity range.

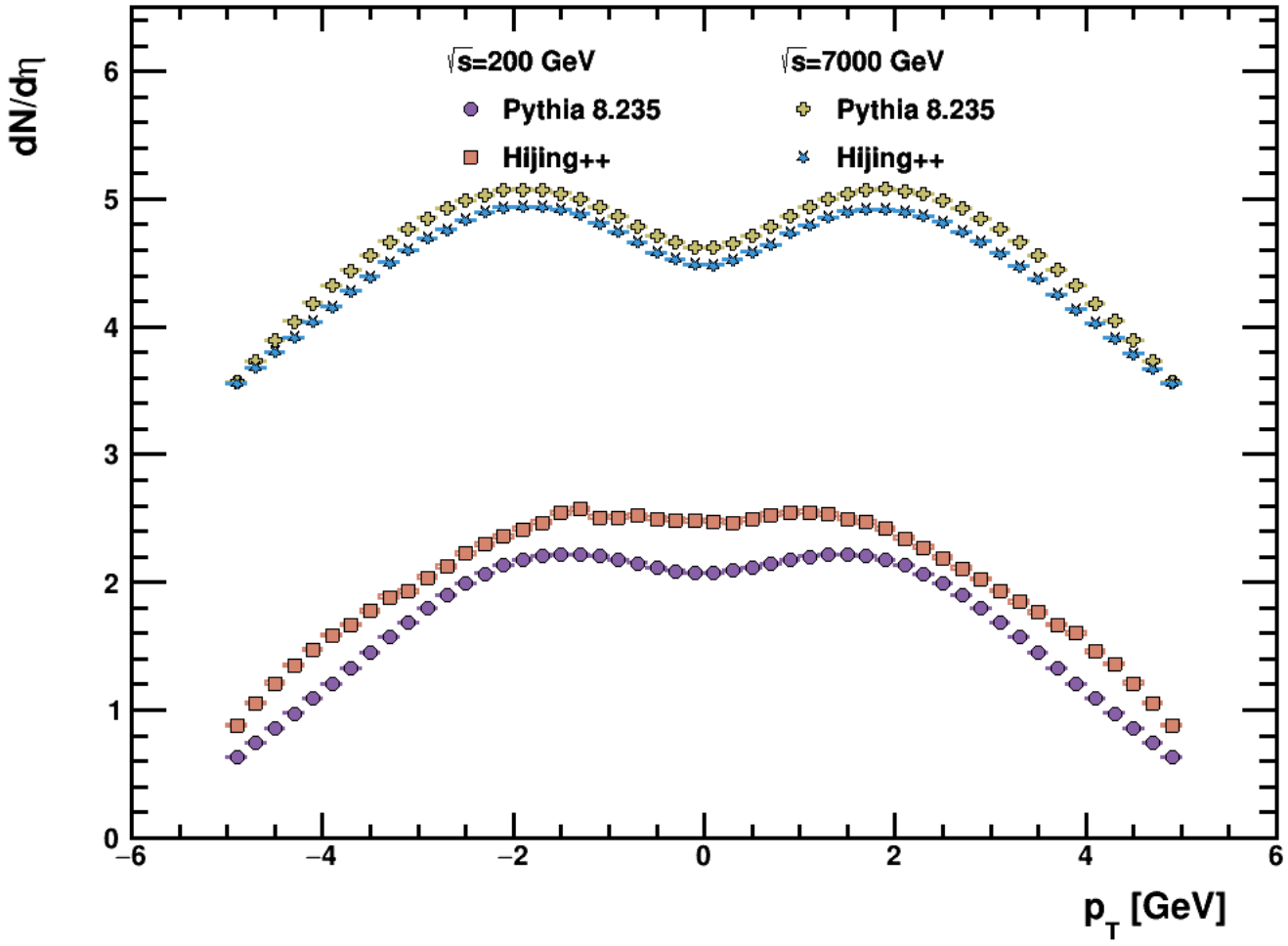


Physics tests: global observables in pp

- (Pseudo) rapidity
  - pp data 200 GeV vs. 7 TeV
  - PYTHIA 8.235 (Monash) vs. HIJING++
  - Change in the trends

    @ 200GeV HIJING++ > PYTHIA

    @ 7TeV PYTHIA > HIJING++
  - At 200 GeV curves are less paralle especially around mid-rapidity.

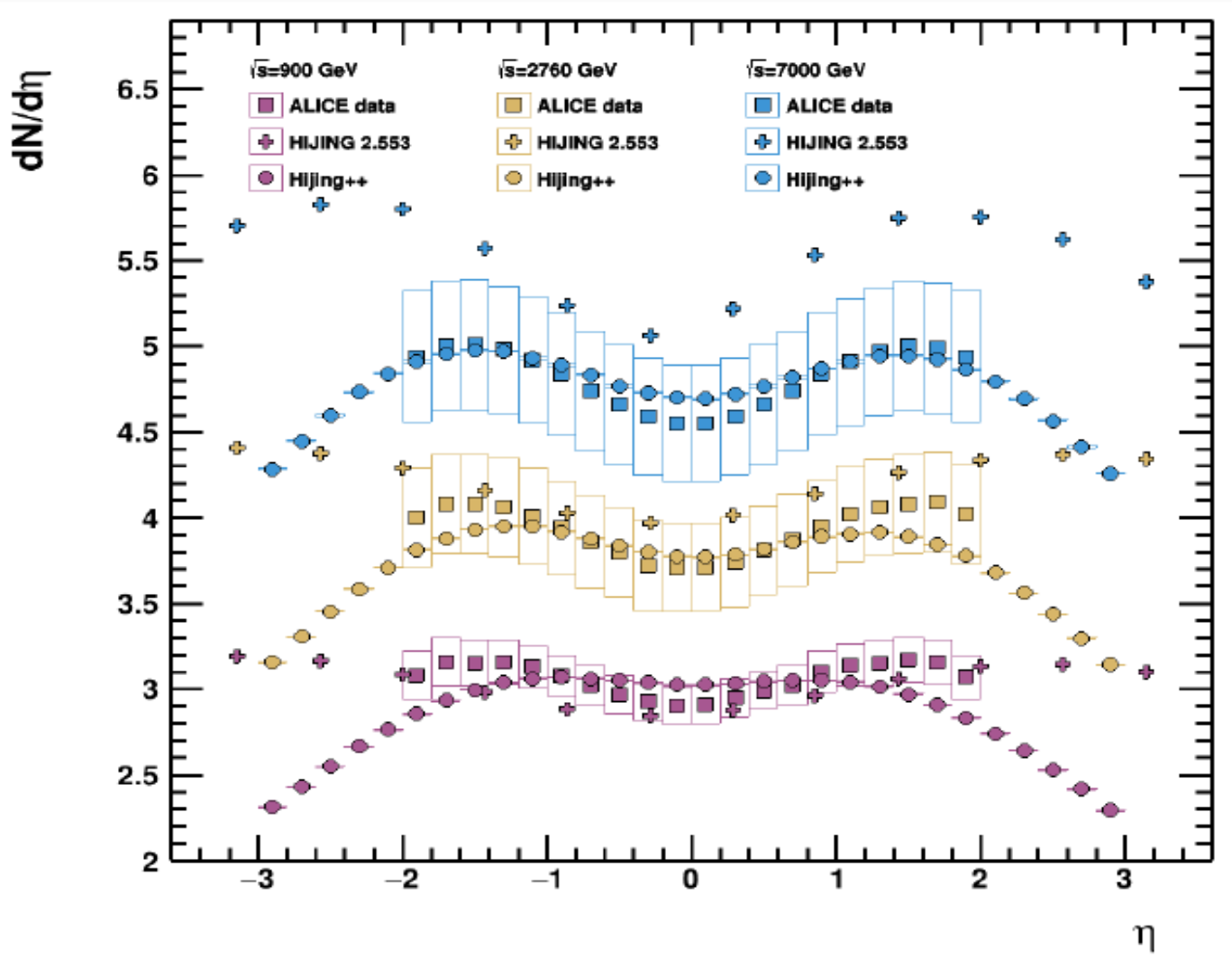


Physics tests: global observables in pp

- (Pseudo) rapidity
  - pp data 900 GeV vs. 7 TeV
  - HIJING 2.553 vs. HIJING++
  - Change in the trends

    @ 900GeV HIJING++ = HIJING

    @ 7TeV HIJING > HIJING-
  - Differences are stronger at high energies and higher pseudorapi

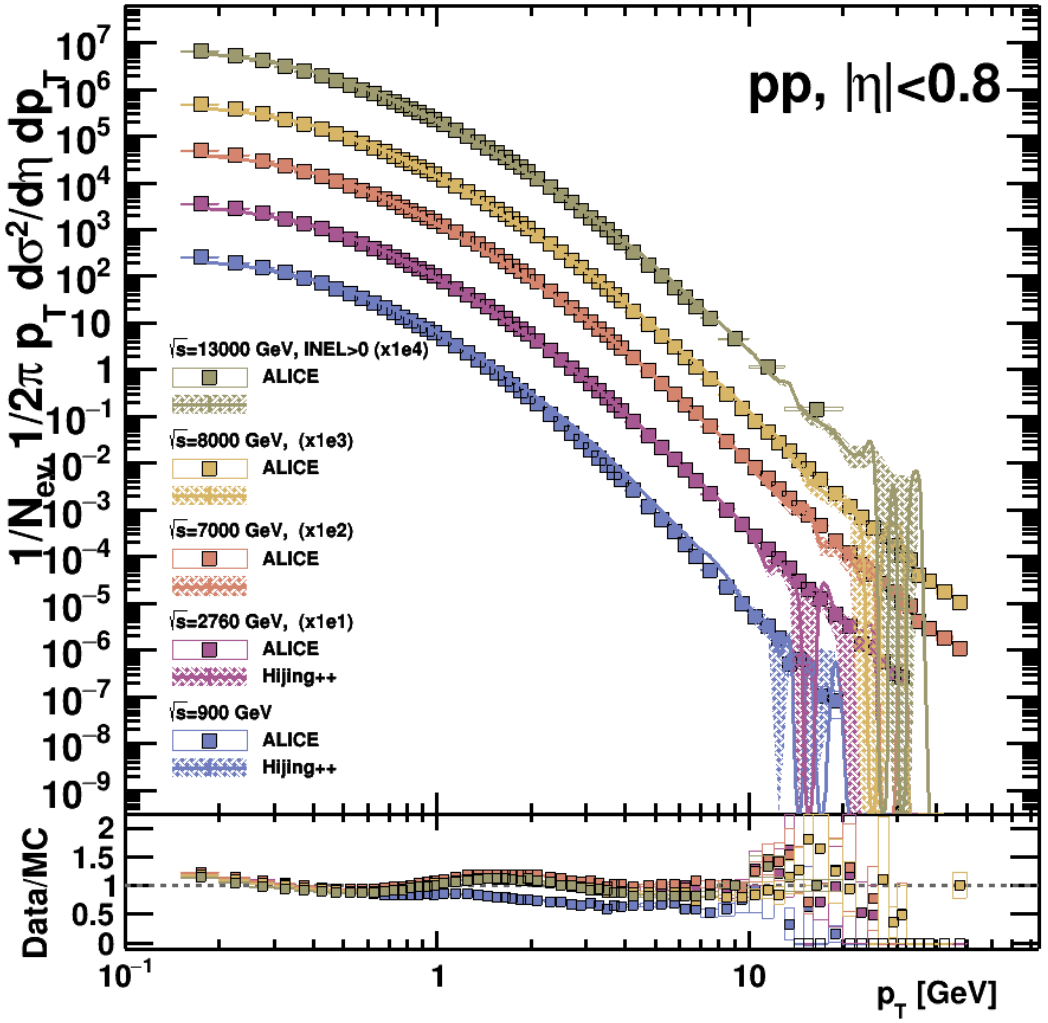


Physics tests: global observables in pp

- Charged hadron spectra
  - pp data 900 GeV - 13 TeV

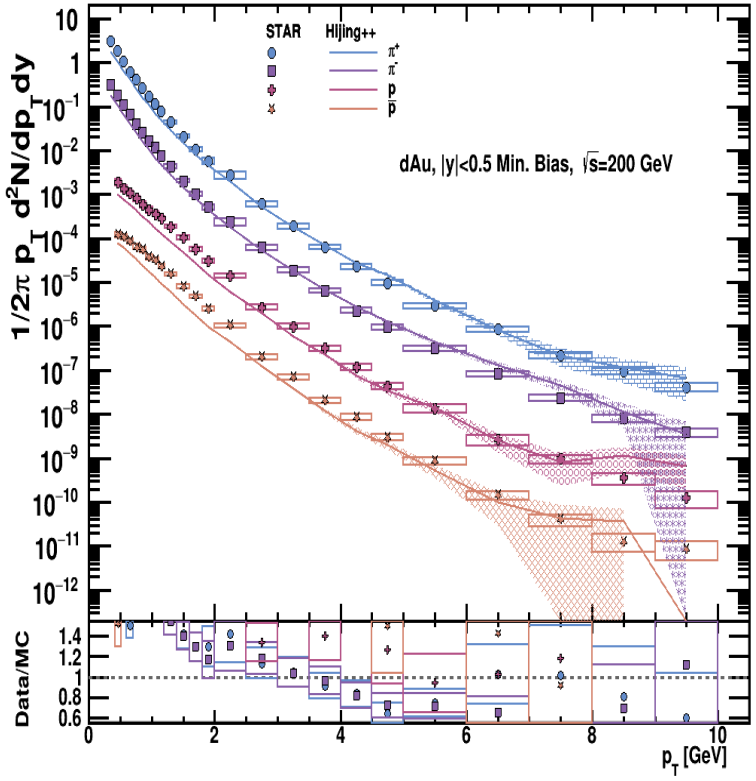
    In this set ALICE data

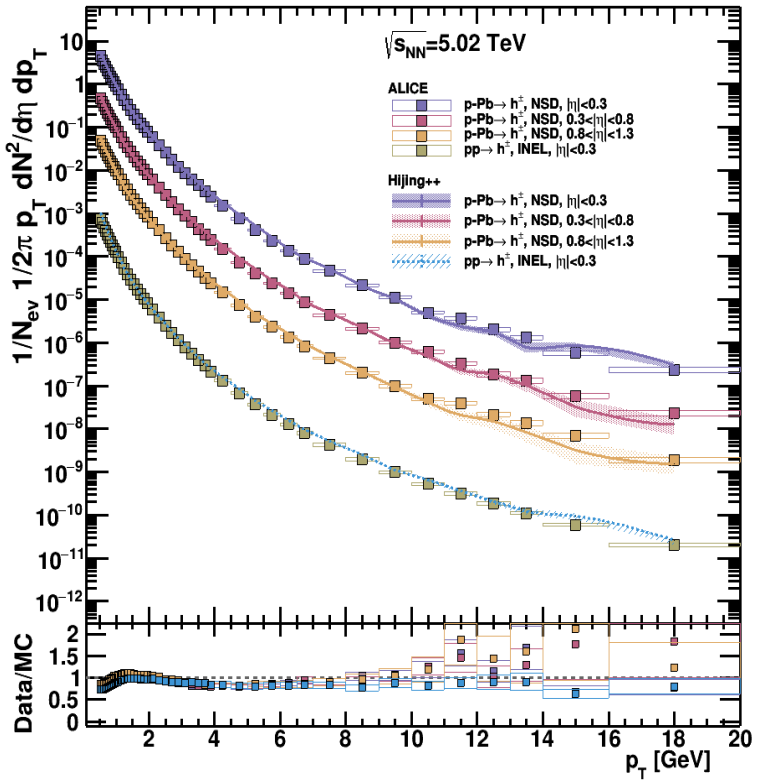
  - Perfect agreement up to 50% in wide transverse momentum and center of mass energy range.



Physics tests: global observables in pA

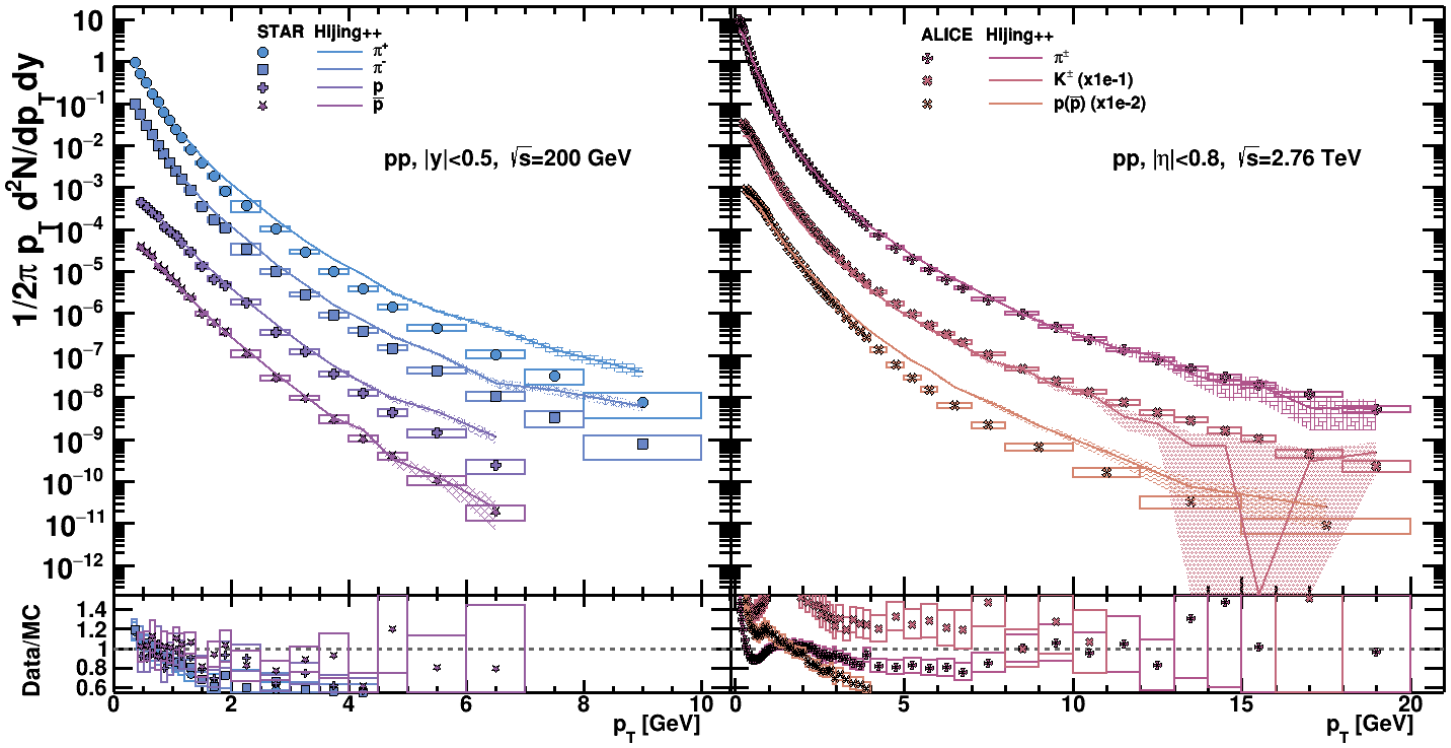
- Charged hadron spectra
  - pp & pA data dAu at 200 GeV and pPb 5.02 TeV
  - In this set STAR & ALICE data
  - Perfect agreement up to 50% in wide transverse momentum and center of mass energy range.





Physics tests: global observables in pA

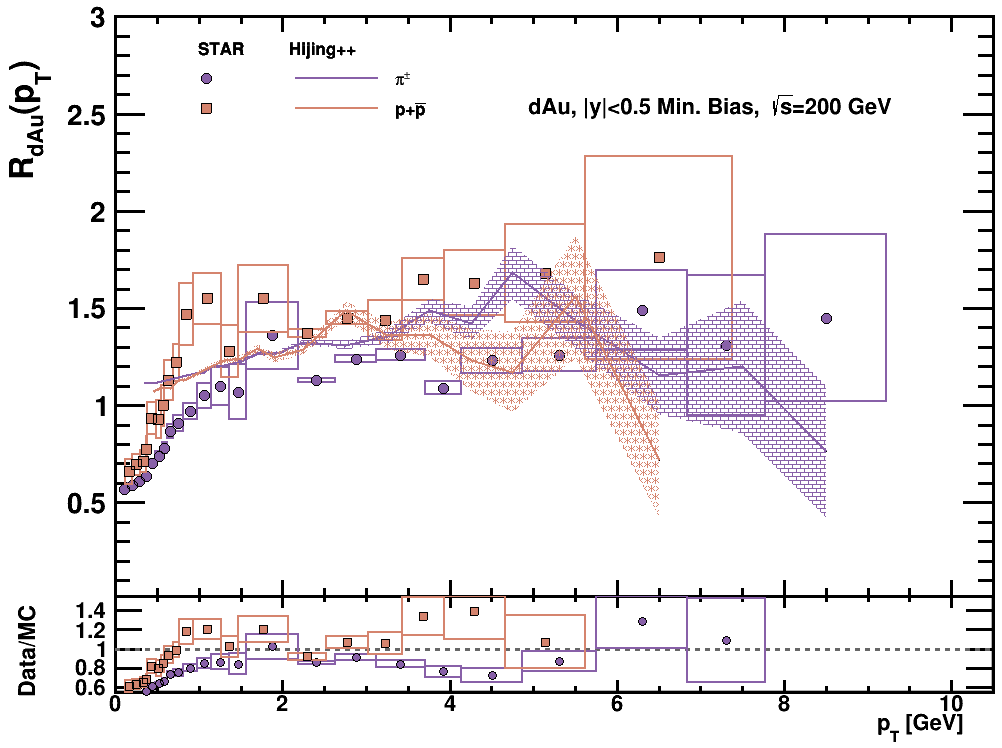
- Identified hadron spectra
  - pp data at 200 GeV and 2.76 TeV
  - In this set STAR & ALICE data
  - Perfect agreement up to 50% in wide transverse momentum and center of mass energy range.
  - High-pT proton production has to be improved.

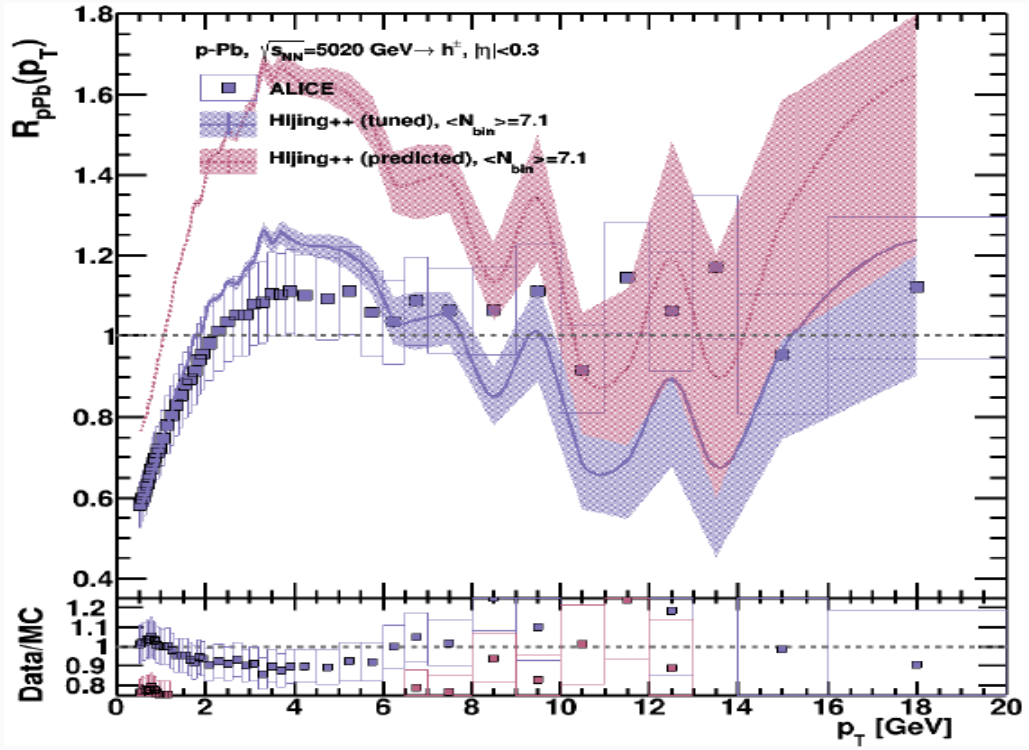


Physics tests: global observables in pA

- Nuclear Modification

$$R_{pPb} = \frac{d^2N_{pPb}/d\eta dp_T}{\langle N_{bin}\rangle d^2N_{pp}/d\eta dp_T}$$

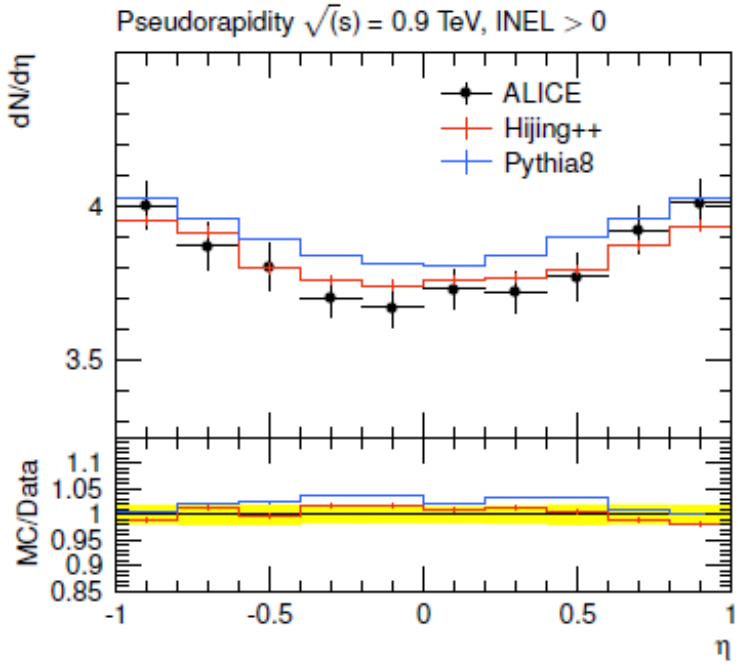


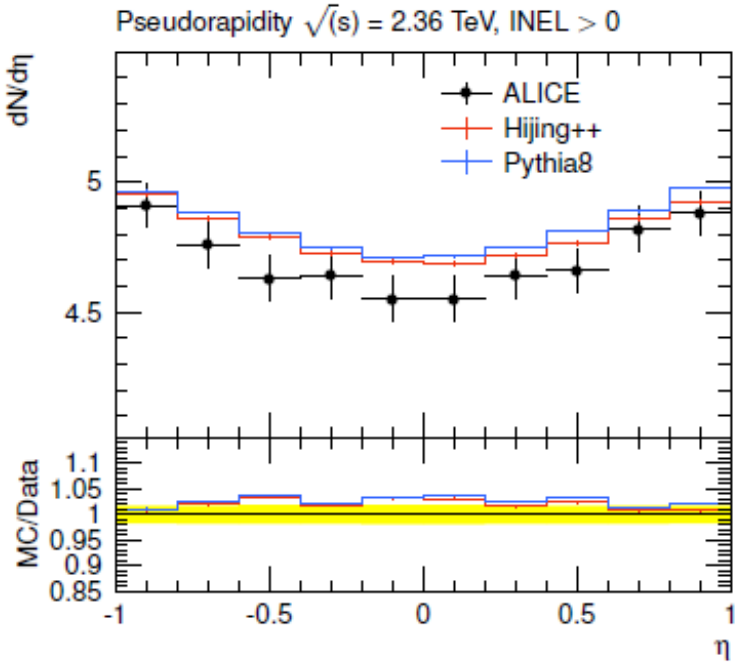


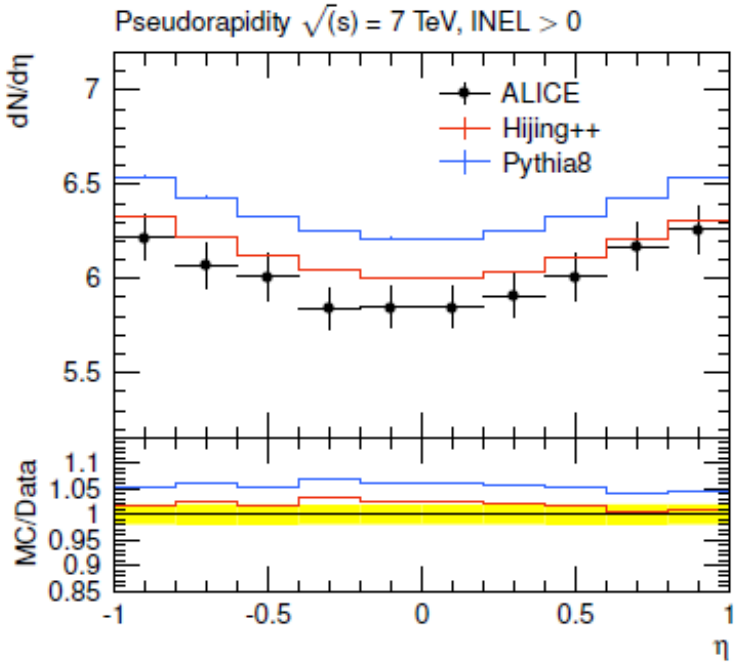
What is the next ???

Test & tunes within RIVET framework

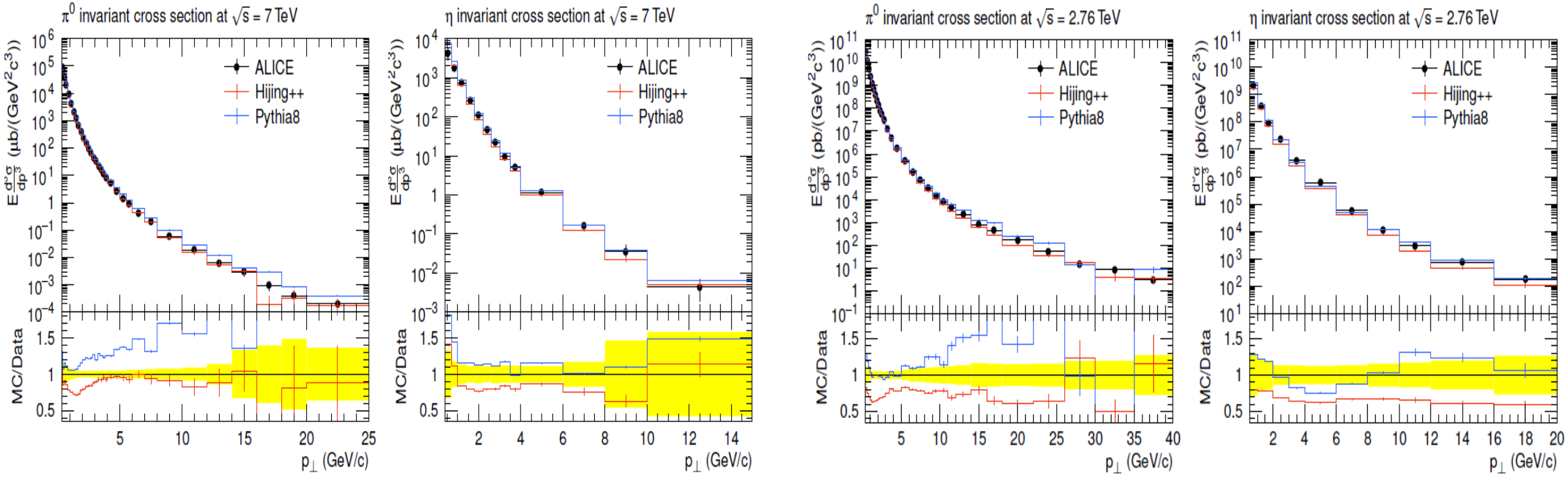
Predictions for ALICE pp collisions at LHC energies



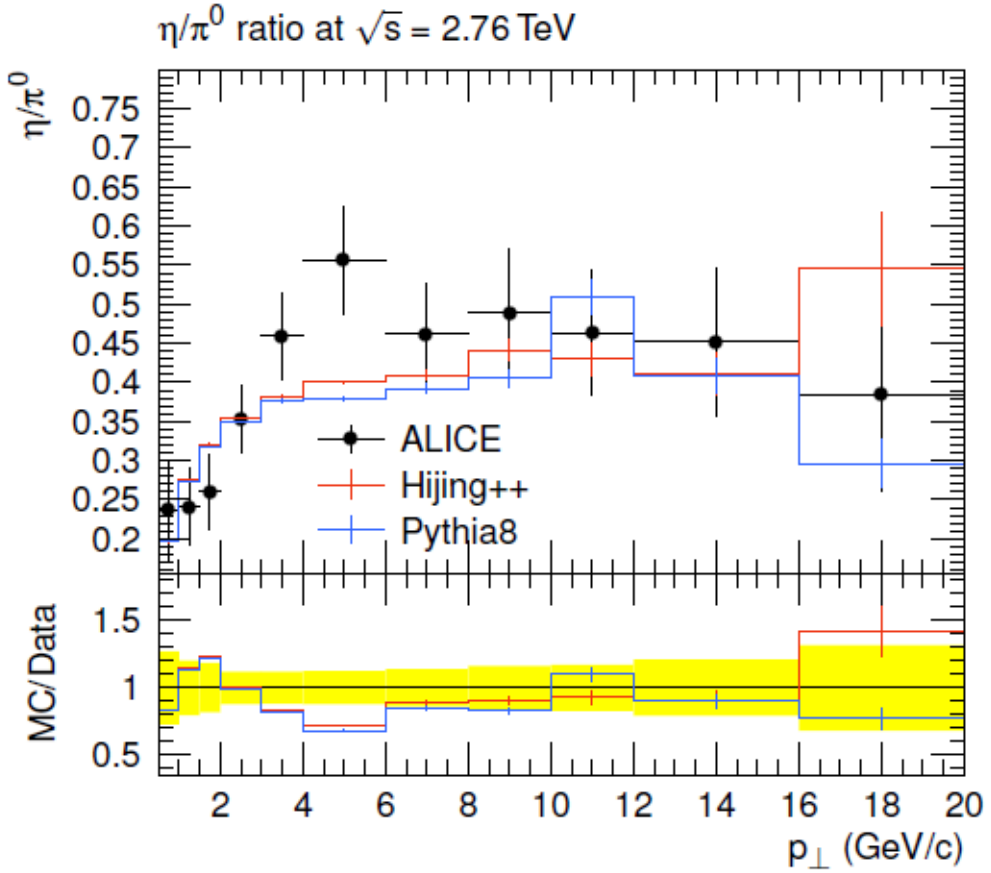


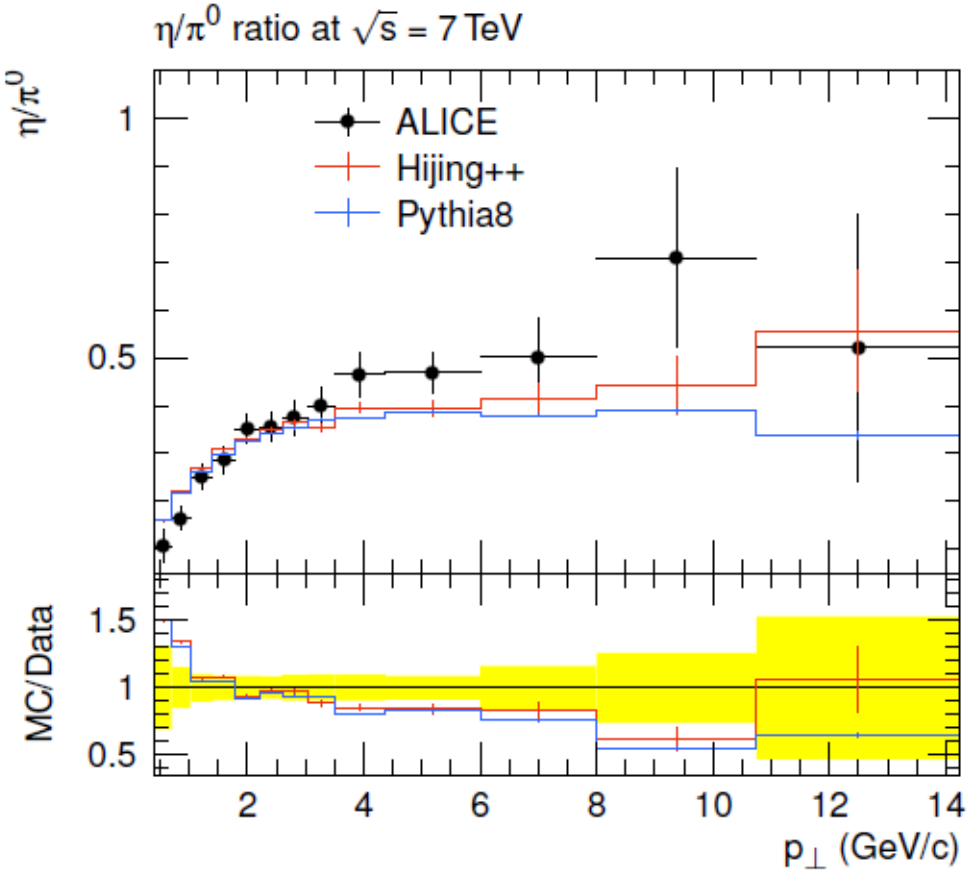


Test & tunes within RIVET framework



Test & tunes within RIVET framework

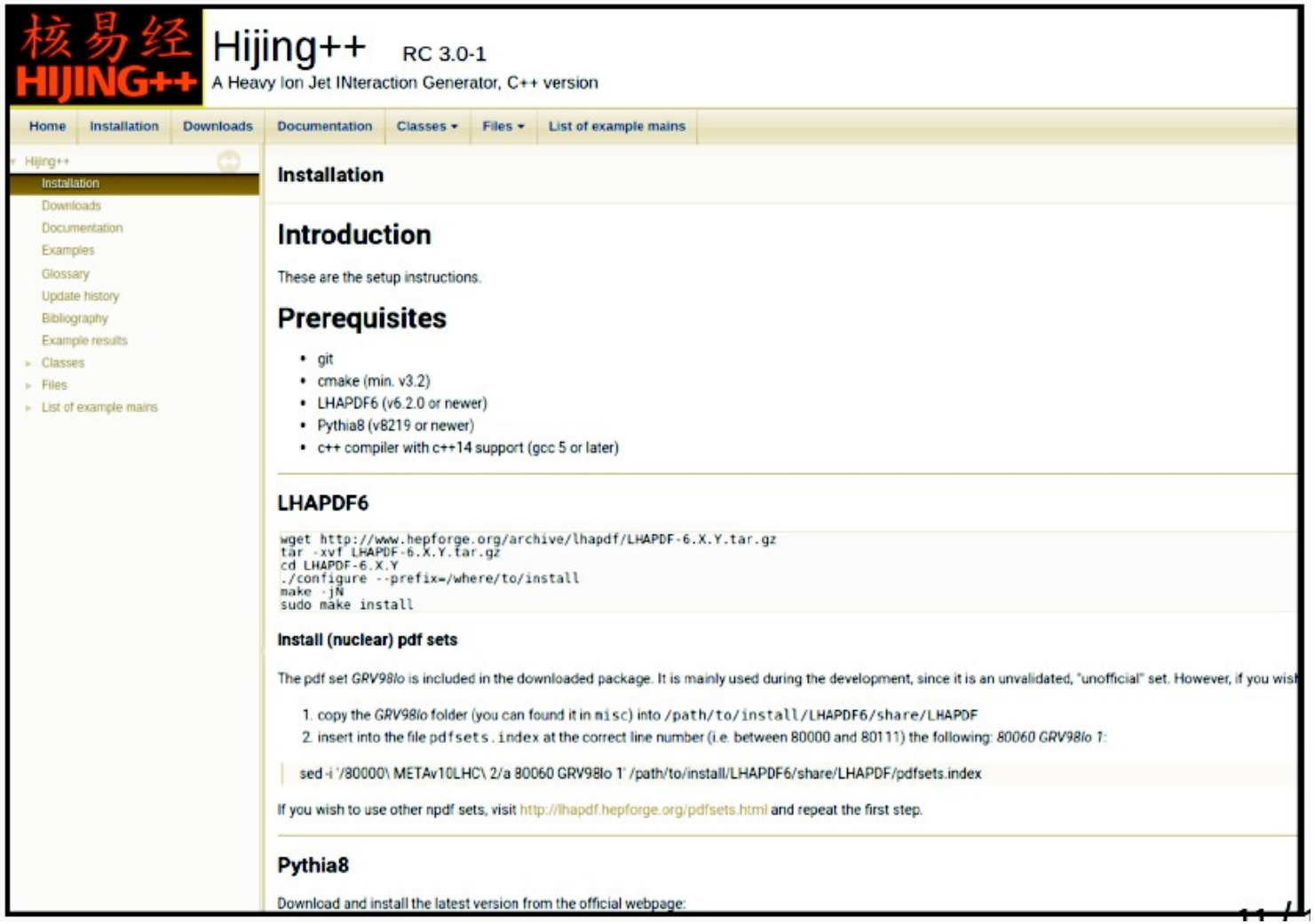




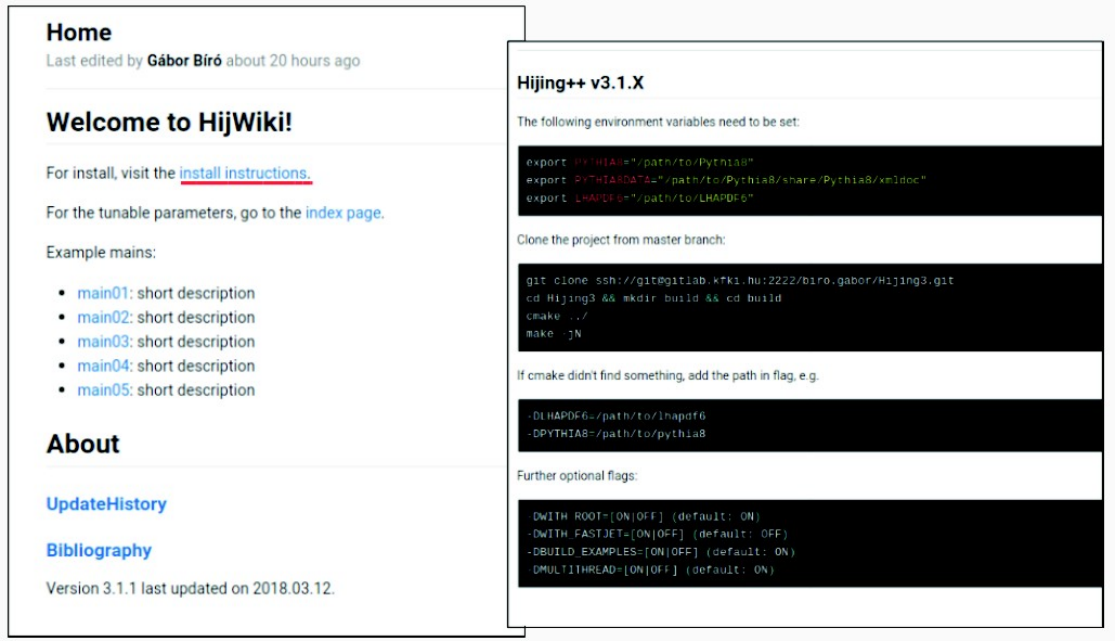
Summary

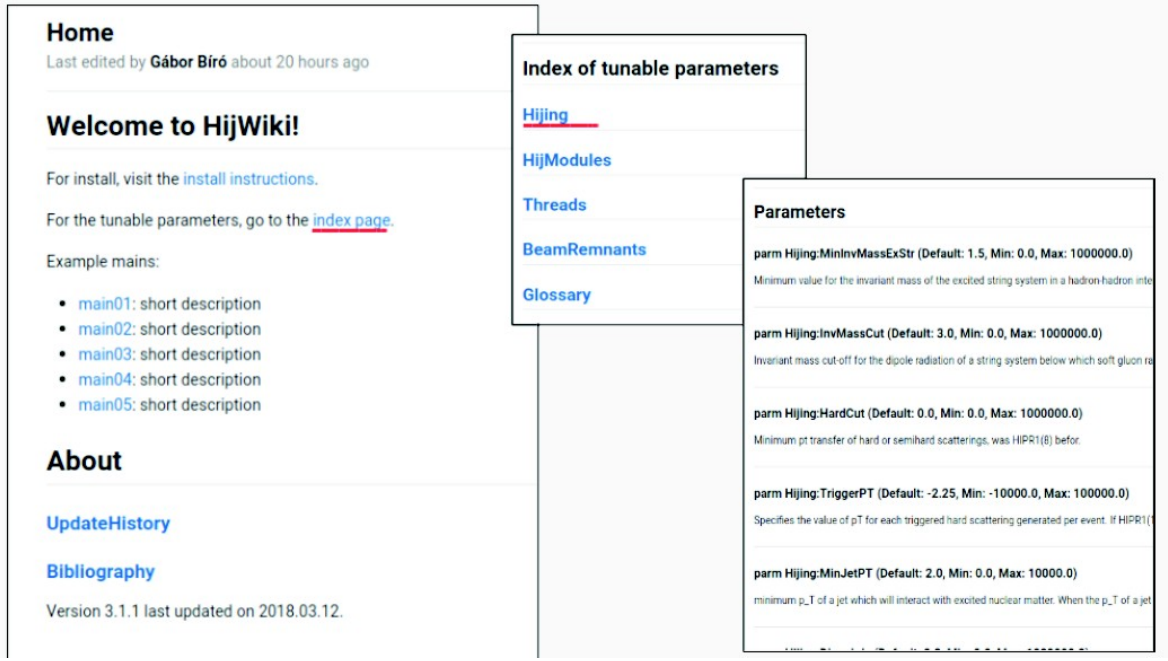
- HIJING++
  - Coding from FORTRAN → C++ has been done
  - One more step HijCore & HijManager were introduced
  - Performance (parallel) tests are ongoing and promising
- First PHYSICS
  - Physics tests has been started
  - Comparison to data is ongoing: RIVET & YODA support is available
  - Tunes are running using PROFESSOR
  - Documentation, documentation, documentation….
- Next
  - Step-by-step reconsidering of nuclear effect (shadowing with Q^2, jet quenching)

Stay tuned... (web page is ready



Documentation is ongoing...





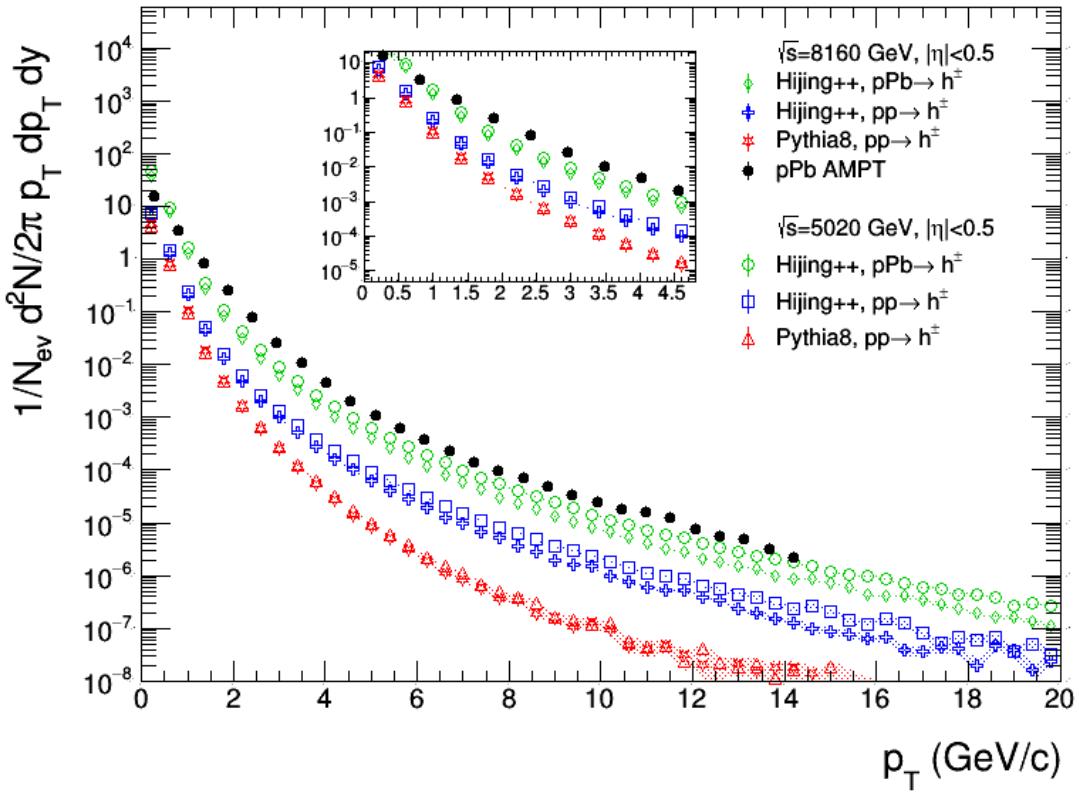
BACKUP

First calculations: pp & pPb

HIJING++ pPb comparison (y=0)

- Test: hadron spectra az 5.02 & 8 TeV

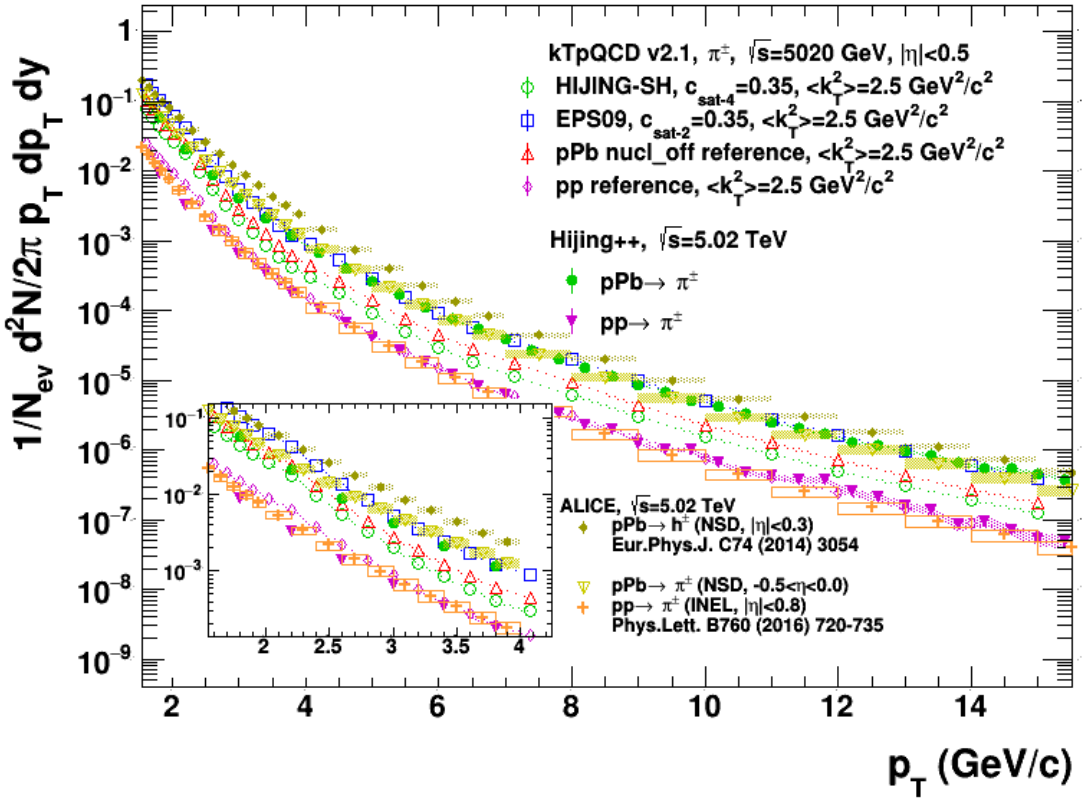
- HIJING++ to Theory (kTpQCD, AMPT)
  - PYTHIA8 on pp
  - AMPT pPb



First calculations: pp & pPb

HIJING++ pPb comparison (y=0)

- Test: hadron spectra az 5.02 & 8 TeV
- HIJING++ to Theory (kTpQCD, AMPT)
  - PYTHIA8 on pp
  - AMPT pPb
  - kTpQCD_v21 with HIJING & EPS09
- HIJING++ to LHC data:
  - ALICE data @ 5.02 TeV pp & pPb

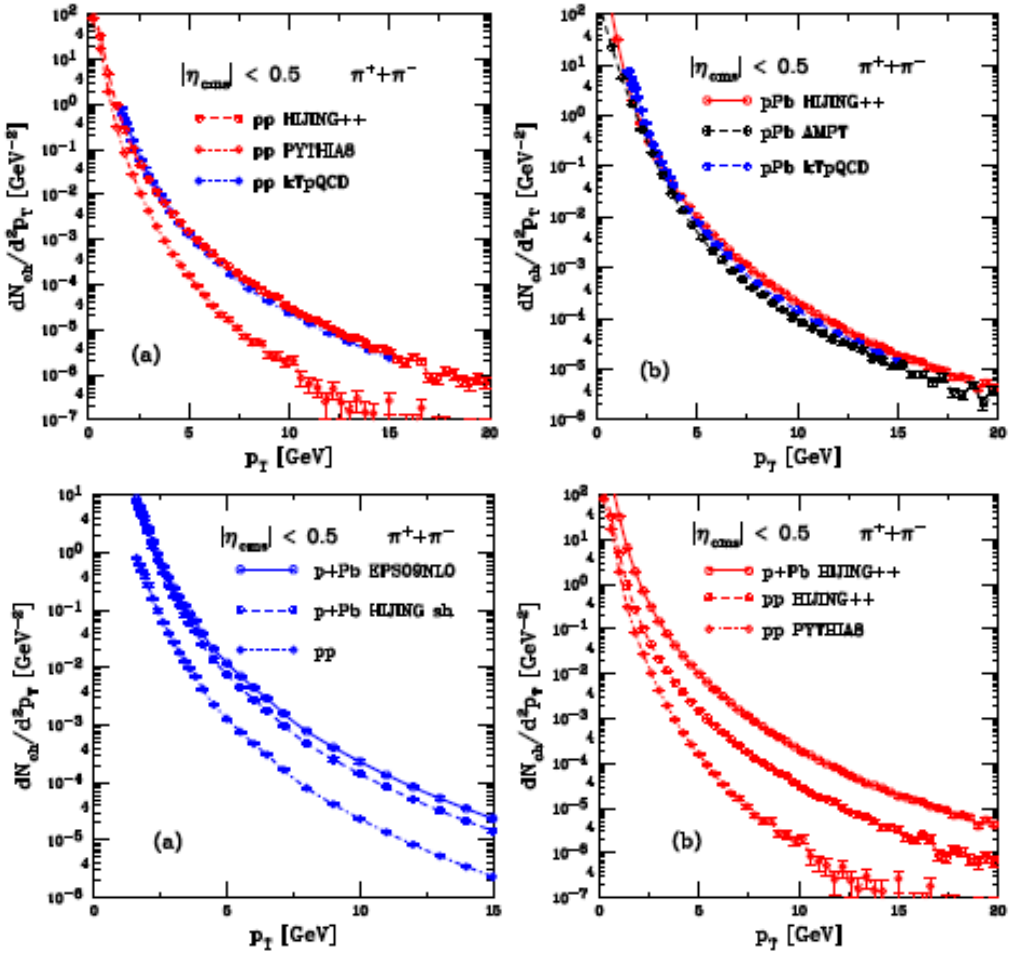


First predictions: pp & pPb

HIJING++ pp & pPb comparison

by R. Vogt: NPA 972 (2018) 18

- Prediction: hadron spectra 8 TeV
- HIJING++ to Theory at 8 TeV
  - PYTHIA8 on pp
  - EPS09NLO
  - AMPT on pPb
  - kTpQCD_v21 on pp & pPb
- Results:
  - Differences at pp level
  - Similar spectra in pPb

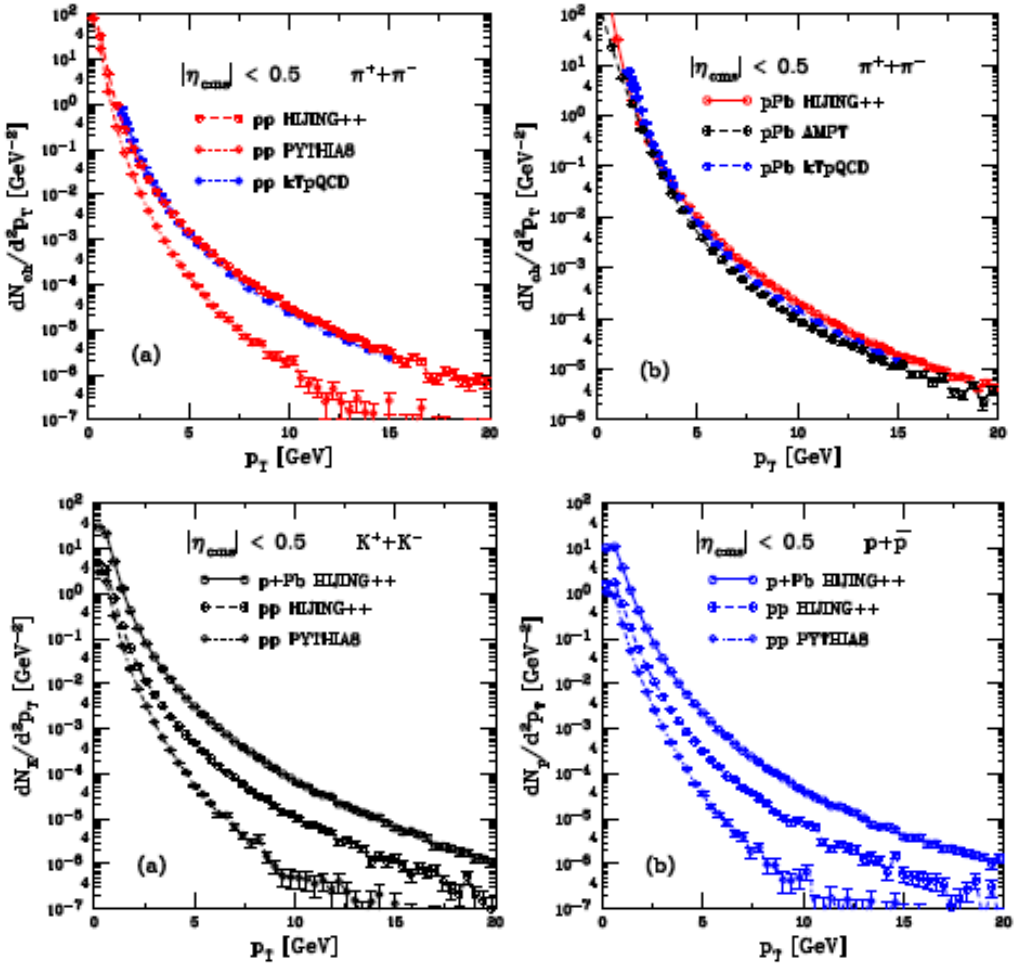


First predictions: pp & pPb

HIJING++ pp & pPb comparison

by R. Vogt NPA 972 (2018) 18

- Prediction: hadron spectra 8 TeV
- HIJING++ to Theory at 8 TeV
  - PYTHIA8 on pp
  - EPS09NLO
  - AMPT on pPb
  - kTpQCD_v21 on pp & pPb
- Results:
  - Major differences for K & p

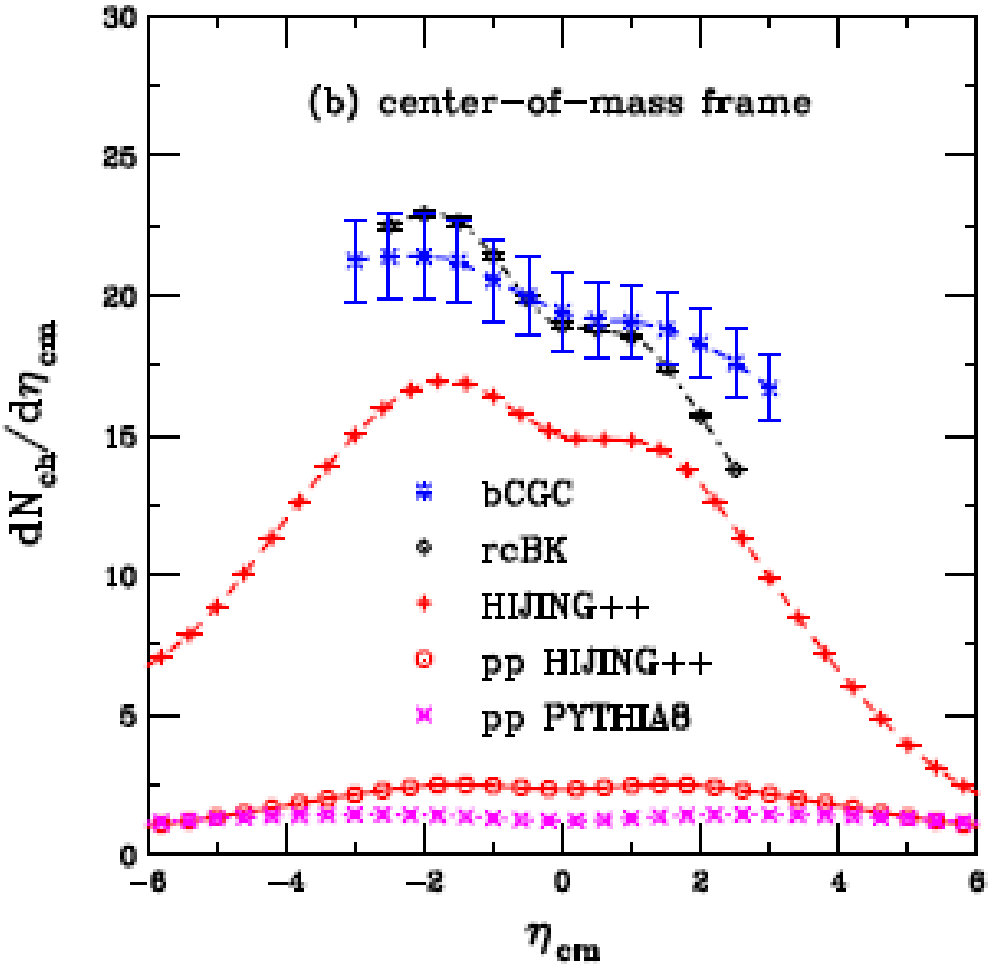


First predictions: pp & pPb

HIJING++ pp & pPb comparison

by R. Vogt NPA 972 (2018) 18

- Prediction: rapidity distribution 8 TeV
- HIJING++ to Theory at 8 TeV
  - PYTHIA8 on pp
  - rcBK
  - bCGC
- Results:
  - Major deviance for PYTHIA8 at midrapidity is coming from minijets



First predictions: pp & pPb

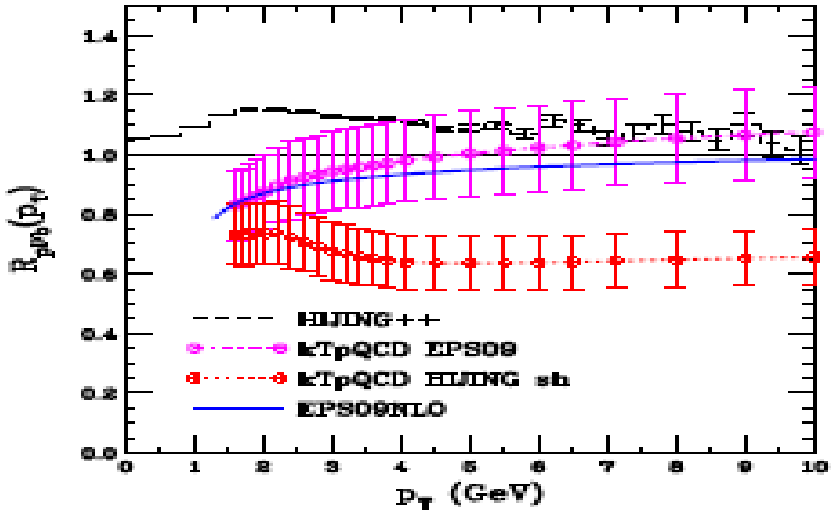
HIJING++ pp & pPb comparison

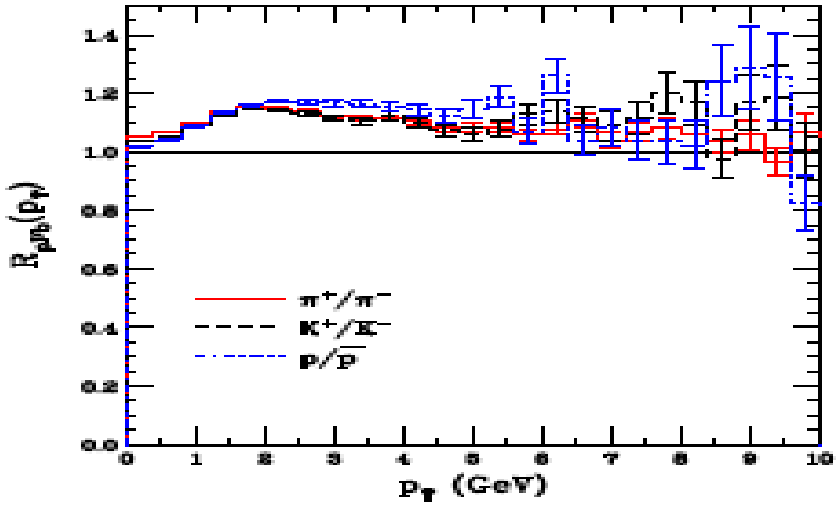
by R. Vogt NPA 972 (2018) 18

- Prediction:

$$R_{pA}(p_T) = \frac{dN_{pA}/dyd^2p_T}{\langle N_{\rm bin}\rangle dN_{pp}/dyd^2p_T}$$

- HIJING++ to Theory at 8 TeV
  - kTpQCD_v21 with EPS09 & HIJING
  - EPS09NLO
- Results:
  - Better agreement with EPS09
  - No relevant difference between π, K, p





First predictions: pPb → heavy hadrons

HIJING++ pPb rapidity dependence

- Prediction at various rapidity:

$$R_{pA}(p_T) = \frac{dN_{pA}/dyd^2p_T}{\langle N_{\rm bin}\rangle dN_{pp}/dyd^2p_T}$$

- Results:
  - To the y>0 similar trends
  - On the y<0 yields increase with mass