GEANT4 COLLABORATION MEETING 2015

J. Apostolakis

With slides by the Geant4 developers on developments, fixes, improvements, validation and plans undertaken by collaborators

TOPICS

- Technical
  - C++11 migration, Multi-threading
- Simulation
  - Physics, Geometry
- Beyond

BACKGROUND

- Geant4 10.0 (Dec 2013) introduced multi-threading (MT)

CMS using 2014/5

- Geant4 10.1 released 5 Dec 2014 - release notes
  - Consolidation of MT features and improvements
  - Slight speedup from EM revisions
  - Physics improvements

ATLAS preparing to use 10.1 end-2015(for MC 16)

- Geant4 10.2 scheduled 6 Dec 2015

C++11 MIGRATION

- Task force, led by Gabriele first goals
  - identify strategy
  - provide documentation & guidelines
  - prepare the ground - study issues, prepare testing

- Created first guidelines , and identified / track issues *Link to slides*
- Current support good from most compilers
  - issues with Visual Studio (awaited VS 15)
  - icc15: e.g. problem with std::vector<big POD> - workaround found

C++11 & G4 MULTITHREADING

- migration from __thread to thread_local done
- migration from pthread to std::thread is being evaluated
  - Big advantage is use for Windows (not critical)

C++11 MIGRATION - PRECIS

- Geant4 code inclusion of C++11 features is well under-way
- Ready for some features to be widely adopted for new developments
- We'll evaluate, case by case, migration of legacy code to c++11 based on effectiveness: e.g. possible speedup with RNG, std::thread
  - expect activities to continue in 2016 and possibly beyond 10.3

Geant4 CMake: 10.2 to 10.3

Slides (with notes)

Ben Morgan

THE UNIVERSITY OF WARWICK

- Reaching limits - seek to tackle in 2016
- Need new **single** structure of libraries
  - neither 'granular' (per directory) nor global (per top level category)
- Propose: custom modular CMake API

Improved Build Product Layout

- When building Geant4, libraries etc are output to same directory structure as they will be installed in:

```
+- YourBuildDir/
   +- <buildscripts>
   +- BuildProducts/
      +- <MODE>/
         +- bin/
         +- lib/
```

Present for Xcode, VS etc

Modularization: G4processes

- **Reached symbol table limit on Windows VS2015**
  - See, e.g., http://cdash.cern.ch/viewBuildError.php?buildid=163810
- **Plan, for discussion, to break into three libs for 10.2:**
  - "G4processes-{hadronic,electromagnetic,general}"

Modularization: After 10.2

- Geant4 == >3500 .hh files over 145 modules
- "Granular" too small, "Global" both too big/small
- How then to organise code into libraries?
  - Let's start discussion this week on how to approach this!
  - There are CMake, Code organization and C++ API design issues here

MULTI THREADING

- Continued refinement
  - option to **seed** worker RNG **once** per run (for 'small' events)
  - setting **affinity** on linux (for performance)
  - refined interface to improve integration with **TBB**

For 10.2:

- some MT-related improvements:
  1. removed explicit initialization of Bertini in run-manager
  2. added possibility to create a non-worker thread (e.g. visualization thread)

Moved to 2016

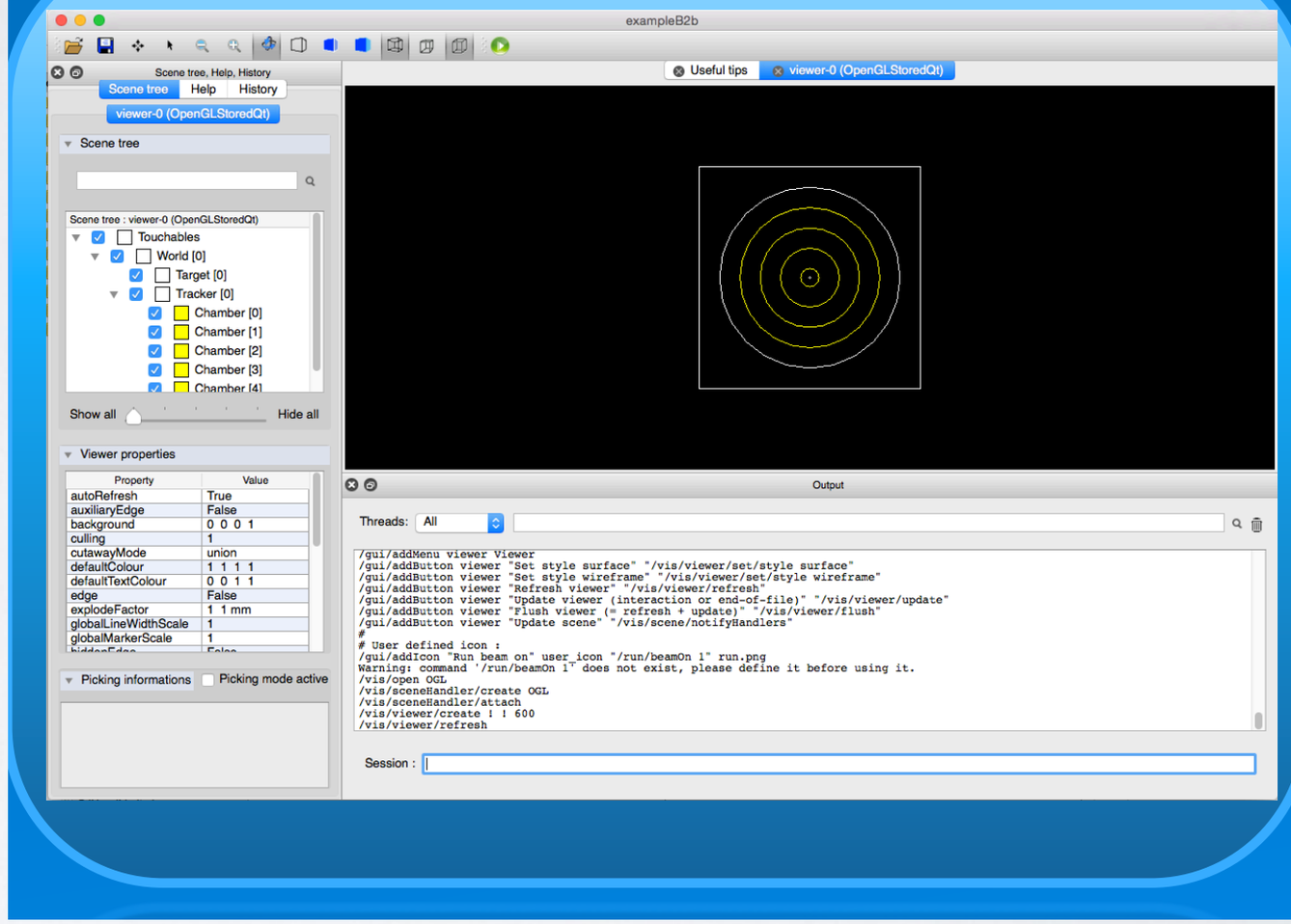
- Use std::thread concurrency from C++11 instead of pthread
  - motivation: Windows support in next version of VS

UI & VISUALIZATION

L. Garnier, J. Allison

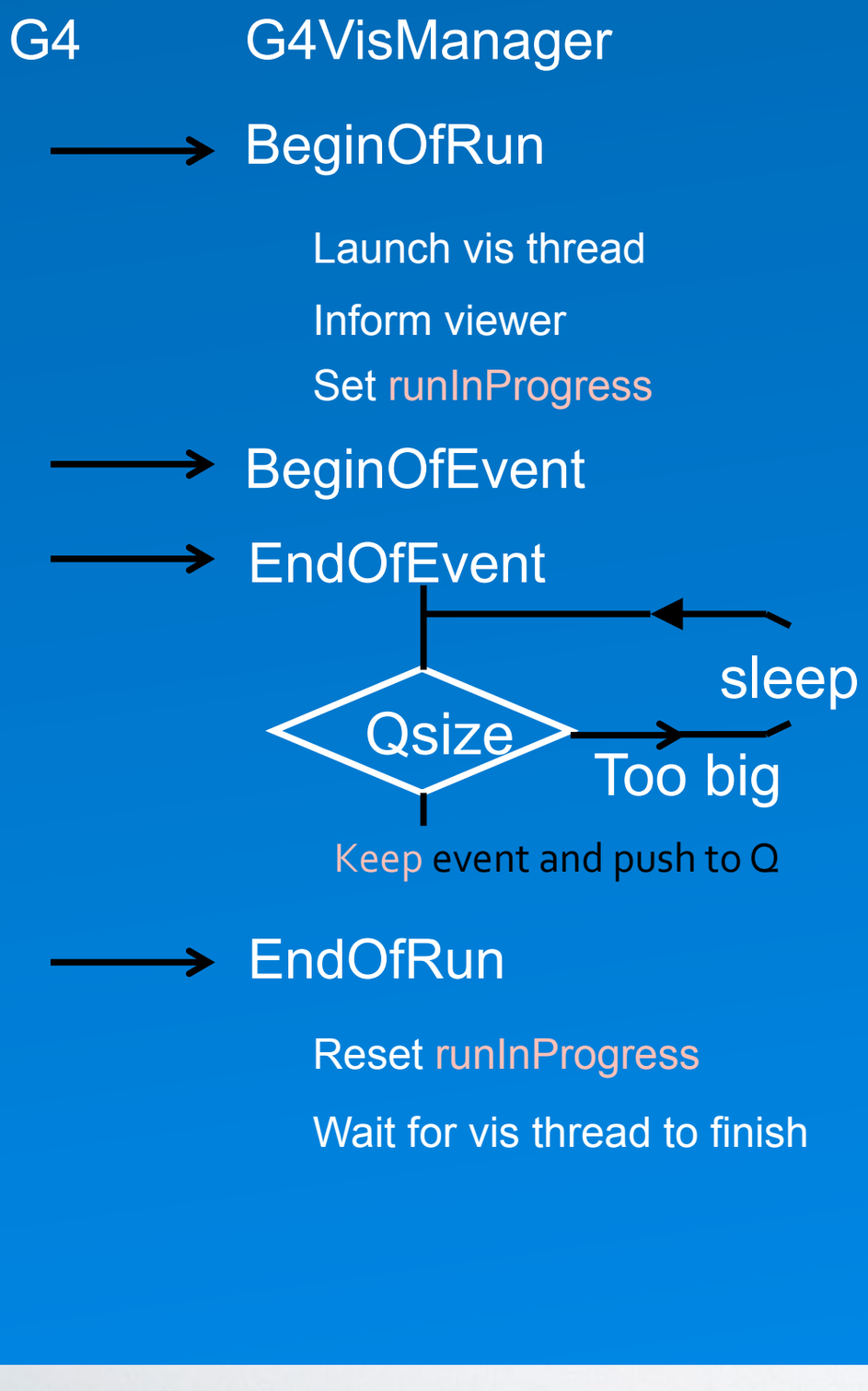
- Make viz in Multi-Threaded ‘transparent’
- user should not see a difference (vs sequential viz)

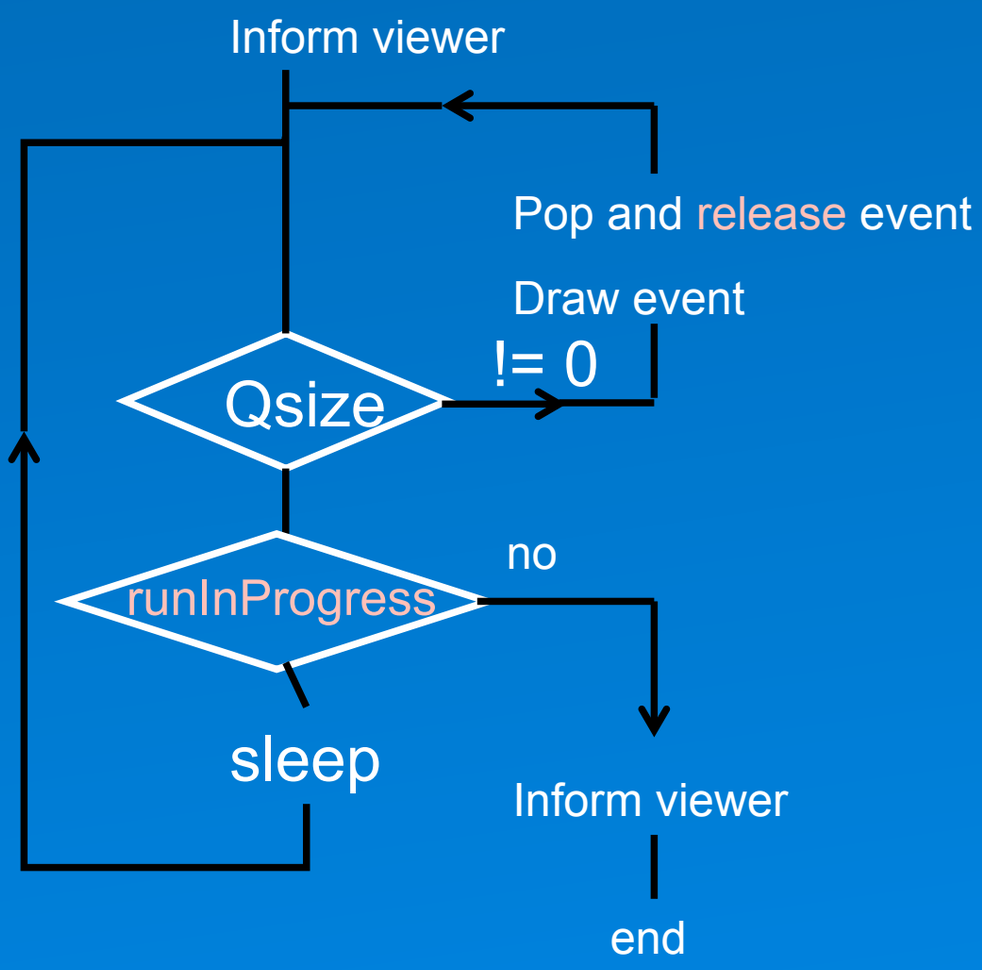
Geant4.10.2 with Qt5.5



- **From 10.2**
  Events are queued and drawn by a separate vis thread during the run
  To the user, multithreaded behaves same as sequential
  As users adopt multithreading they require full vis features

Vis flow diagram - multithreaded mode





Notes:

- Queue is `std::deque<const G4Event*>`
- Appropriate locks are used to set and reset runInProgress
- "Keep" and "release" refer to messages to the run manager (new feature)

ANALYSIS

- Introduced G4Parameter<> like TParameter<> in Root to simplify merging of user data in MT

B1 example

```
class B1Run : public G4Run {
  public:
    ...
    // method from the base class
    virtual void Merge(const G4Run*);
    void AddEdep (G4double edep);
    // ...
  private:
    G4double  fEdep;
    G4double  fEdep2;
};
```

Run class and Merge() method are not needed

```
#include “G4Parameter.hh”
...
class B1RunAction : public G4UserRunAction {
  public:
    ...
    // method from the base class
    void AddEdep (G4double edep);
    // ...
  private:
    G4Parameter<G4double>  fEdep;
    G4Parameter<G4double>  fEdep2;
};
```

VALIDATION TOOL

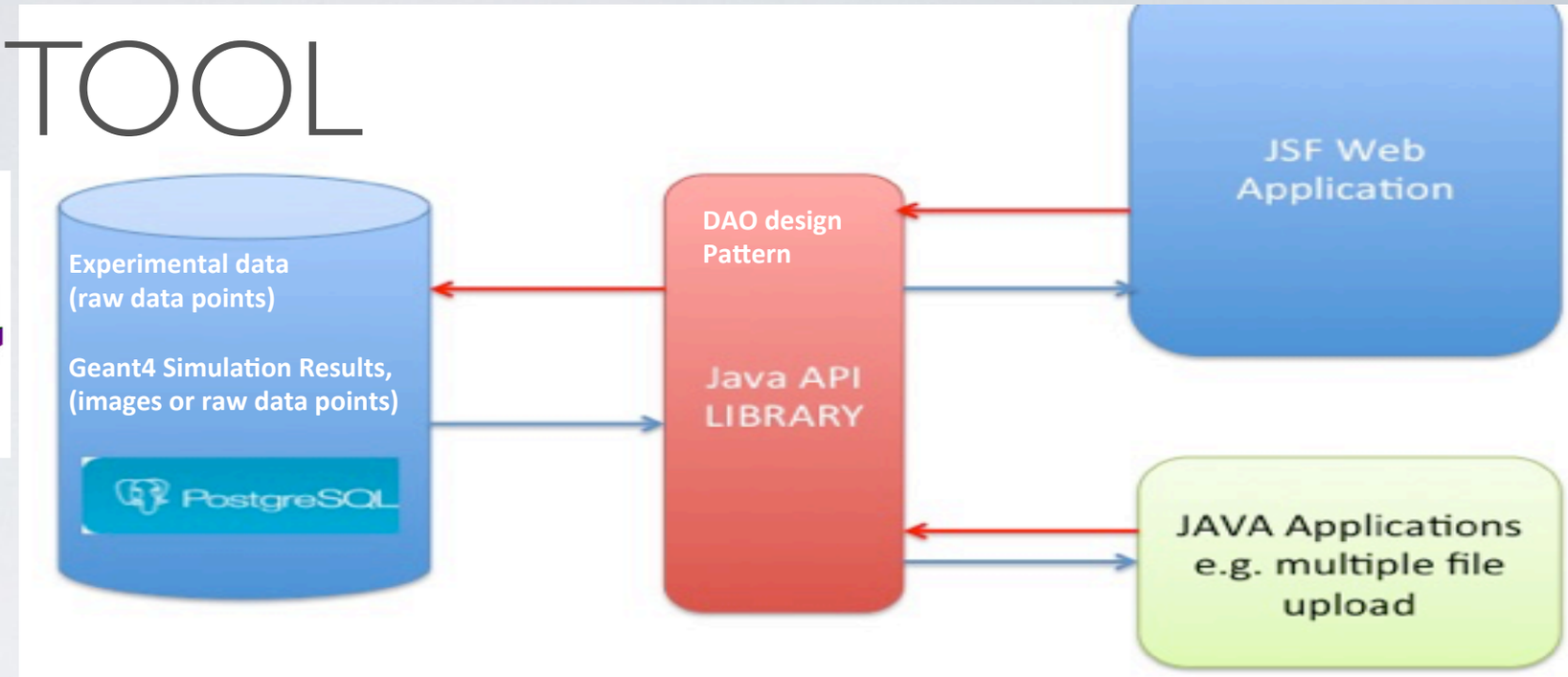
Web Application & Validation

Hans Wenzel

Fermilab

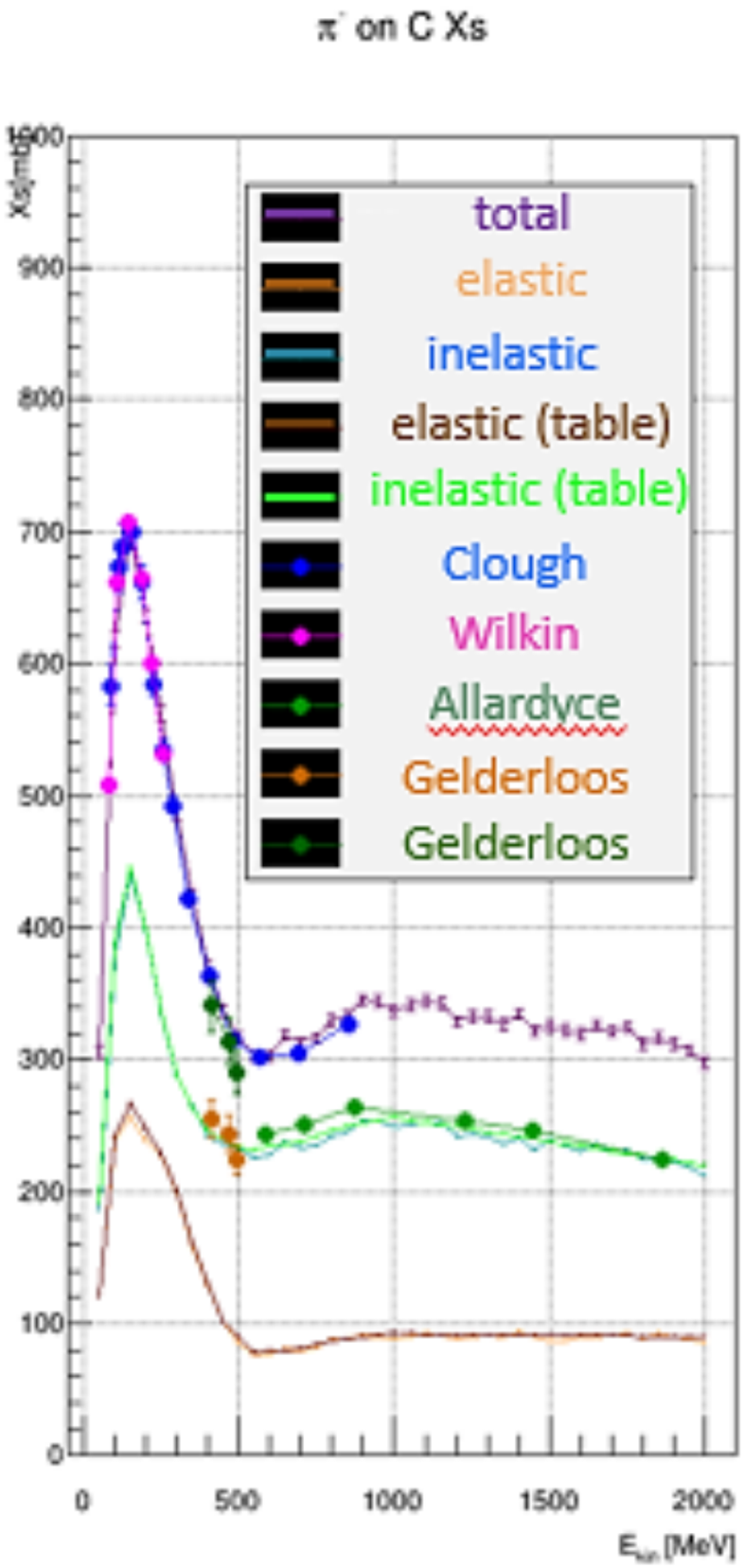
Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

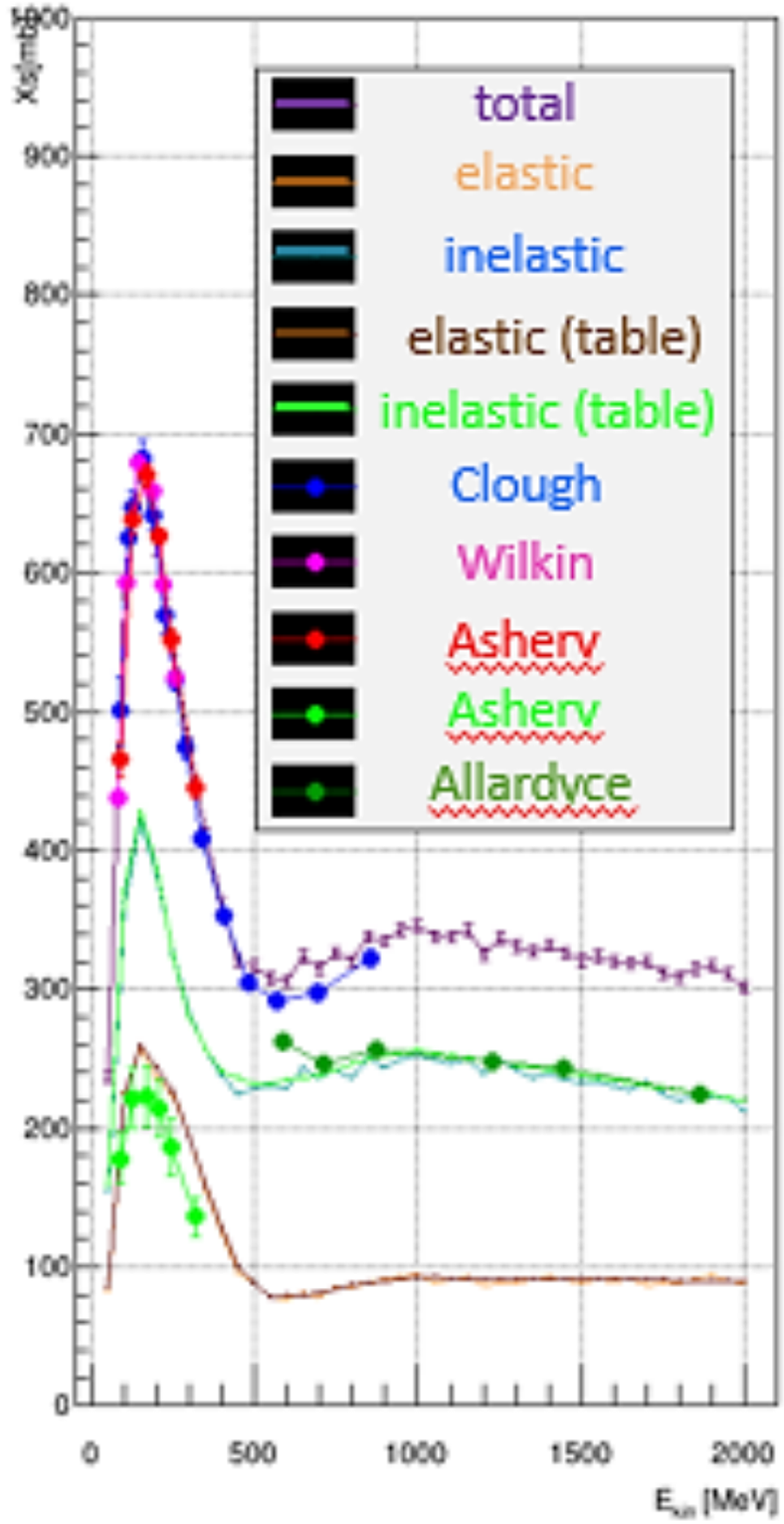
- Display sim vs. data
  - for releases
  - for model developers (future)



Choice of technologies

PostgreSQL	Open source relational data base, hosted by Fermilab data base group.
	Glassfish: Web Application server hosted on fermicloud
PRIMEFACES	Primefaces JSF (Java Server Faces) based framework to create modern looking web pages (provides HTML5 support) and easy to navigate menus.
NetBeans	Integrated Development Environment
	Java programming language, JAVAEE
JFREECHART	Java library used to create the graphs





Sample results - Simulations with G4 ver 10.01.p02

Courtesy of Rasheed Auguste, Geant4 SIST Final Presentation

New round of discussions for FNAL/CERN collaboration on tool for developer & release regression

Physics

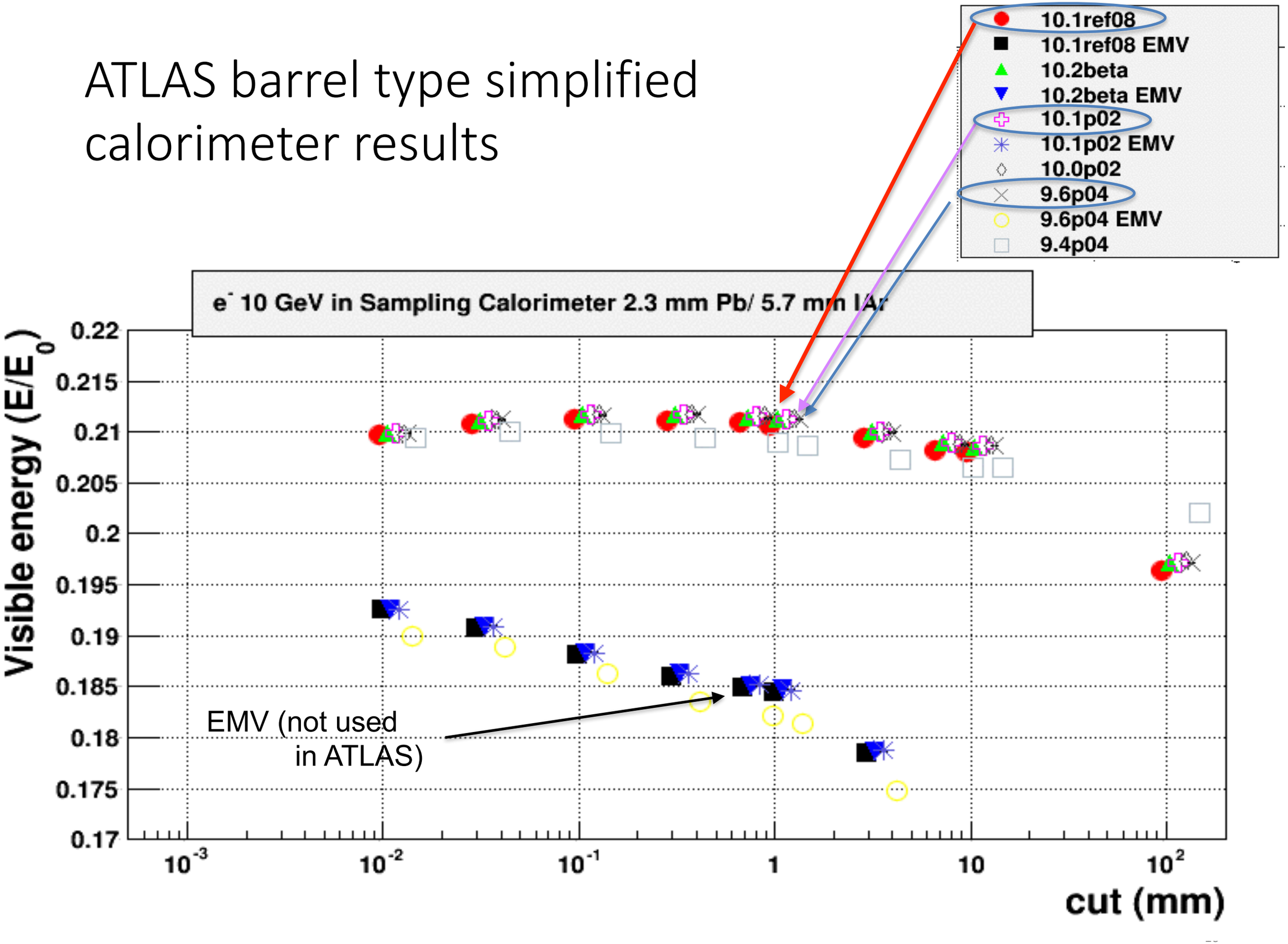
EM validation

- Validation of EM showers shows stable results from 9.6 to 10.0 and 10.1
- Very small variation of EM energy deposition in tested simplified setups - typically less than 0.2%
  - Fe/LAr, Pb/LAr, W/LAr, Fe/Scintillator (see next slides)

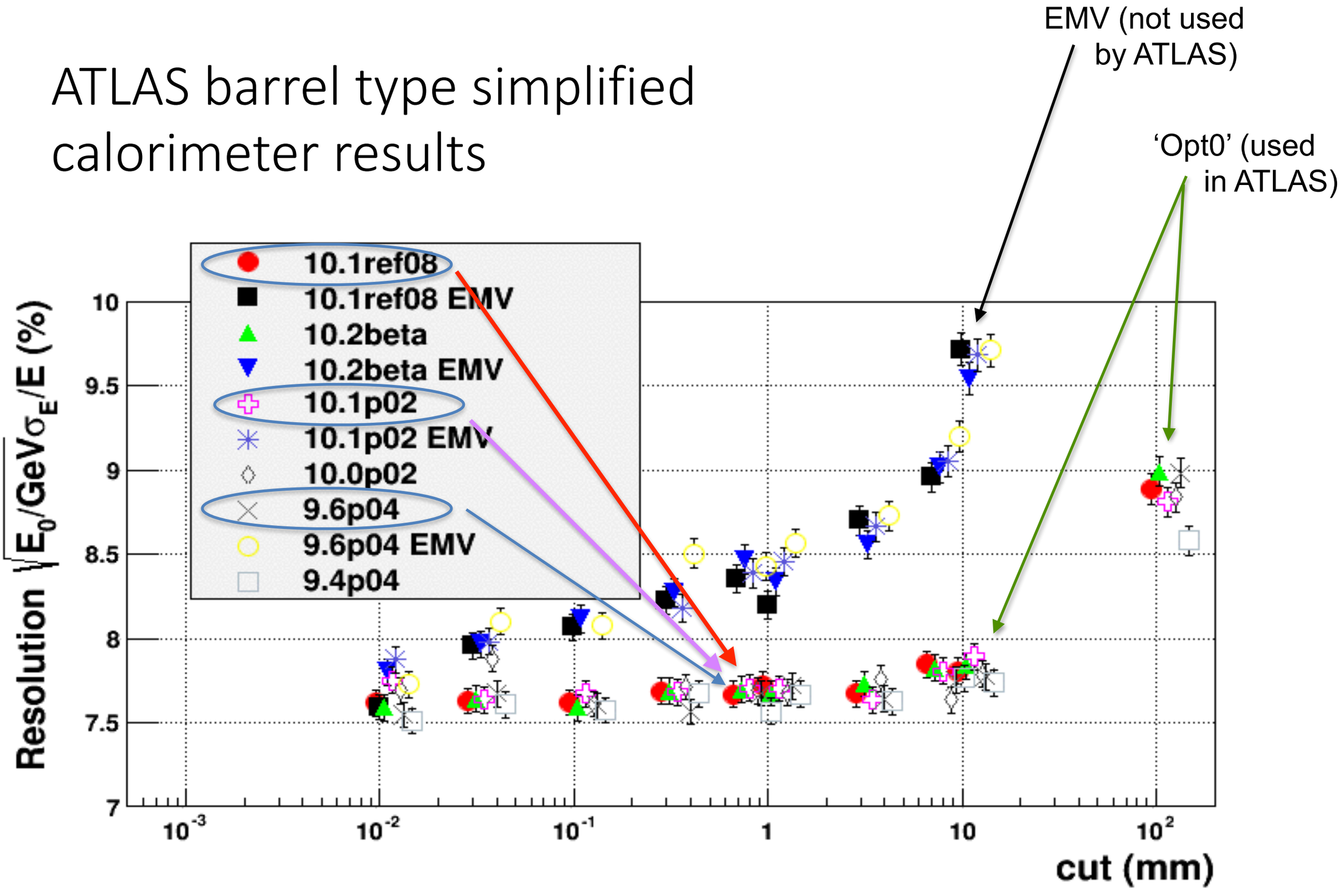
- ASIDE: Contradiction with recent ATLAS reports/observation
  - in EMB (change ~0.5% in some cases) and
  - Tilecal (angular dependence and change 0.2 - 3.0%)

Need to follow-up, using further information about setup

ATLAS barrel type simplified calorimeter results



ATLAS barrel type simplified calorimeter results



EM Developments for 10.2

- From the ground rewrite of Goudsmit-Saunderson Msc model
  - New approach from first principles, avoid parameterisations (M.Novak - developed for GeantV, adapted for G4)
  - Promising early results - better accuracy and speed, *but changes some simplified calorimeter results**
  - Created new EM physics builder G4EmStandardPhysicsGS
- Revised Urban Msc model
  - option of new revised lateral displacement sampling - *better backscattering, but changes some simplified calorimeter results**
  - added positron correction

** clarification(s) added - in italic*

EM Developments for 10.2

- Before Geant4 9.6 low energy protons/muons could survive many steps - caused problem in ATLAS
  - Fix: hard-coded tracking cuts in elastic scattering
  - Problem: difficult to control for studies / other users
- In 10.2 moved to lowest energy threshold/cut in ionisation processes
  - replaced hidden hard-coded tracking cuts in elastic scattering

Now it is made consistent:

- stops all charged particles when energy at end-of-step is below it
- turns the remaining kinetic energy into energy deposition
- allows at rest processes to be applied (e.g. e+ annihilation)

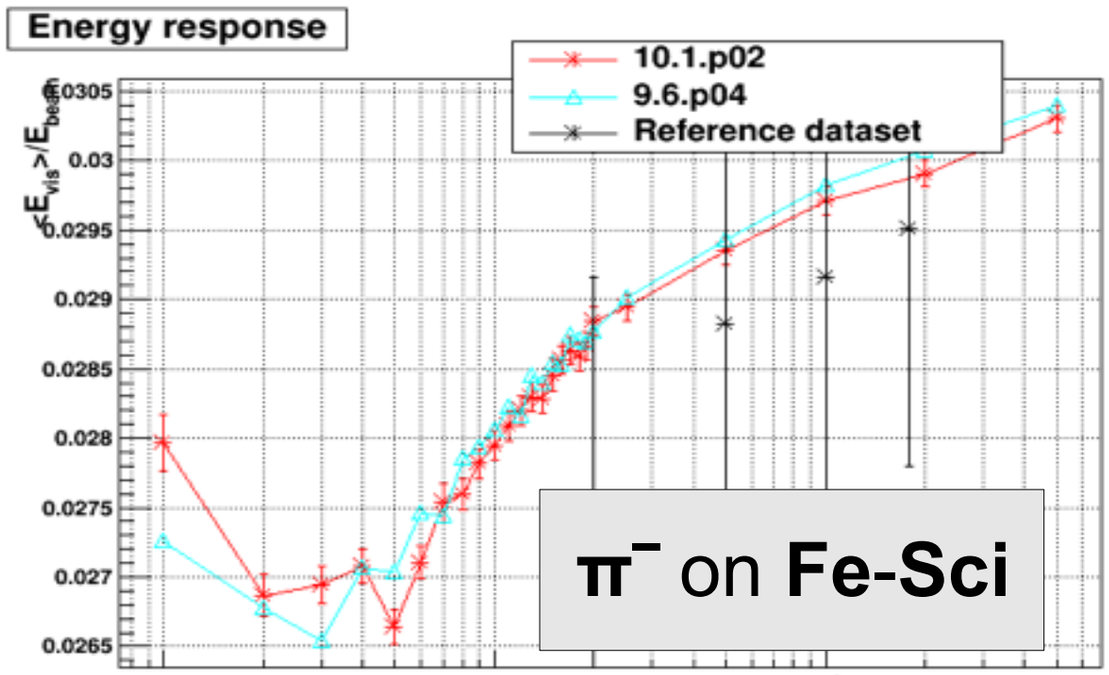
Threshold can be controlled by user (via method or UI-command)

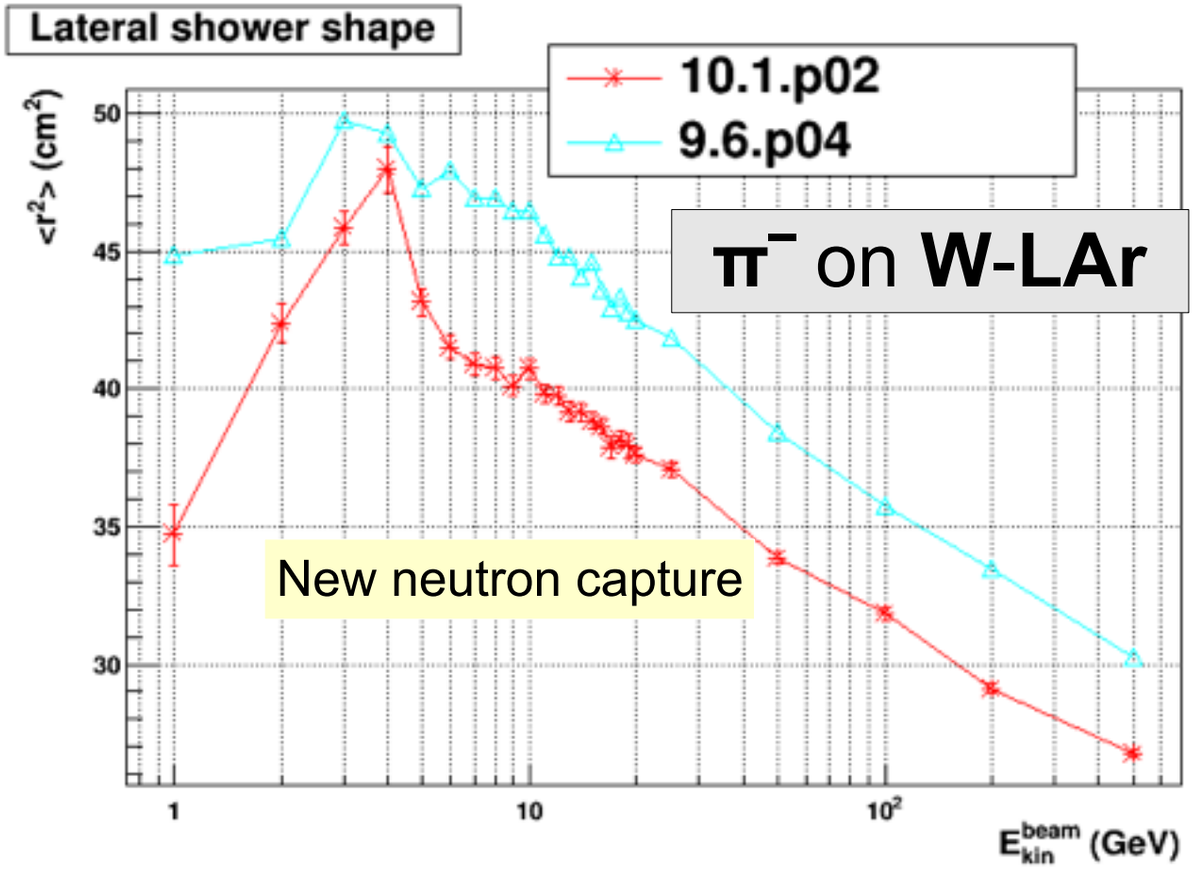
***NOTE**: this is separate from the production threshold for protons/ions**

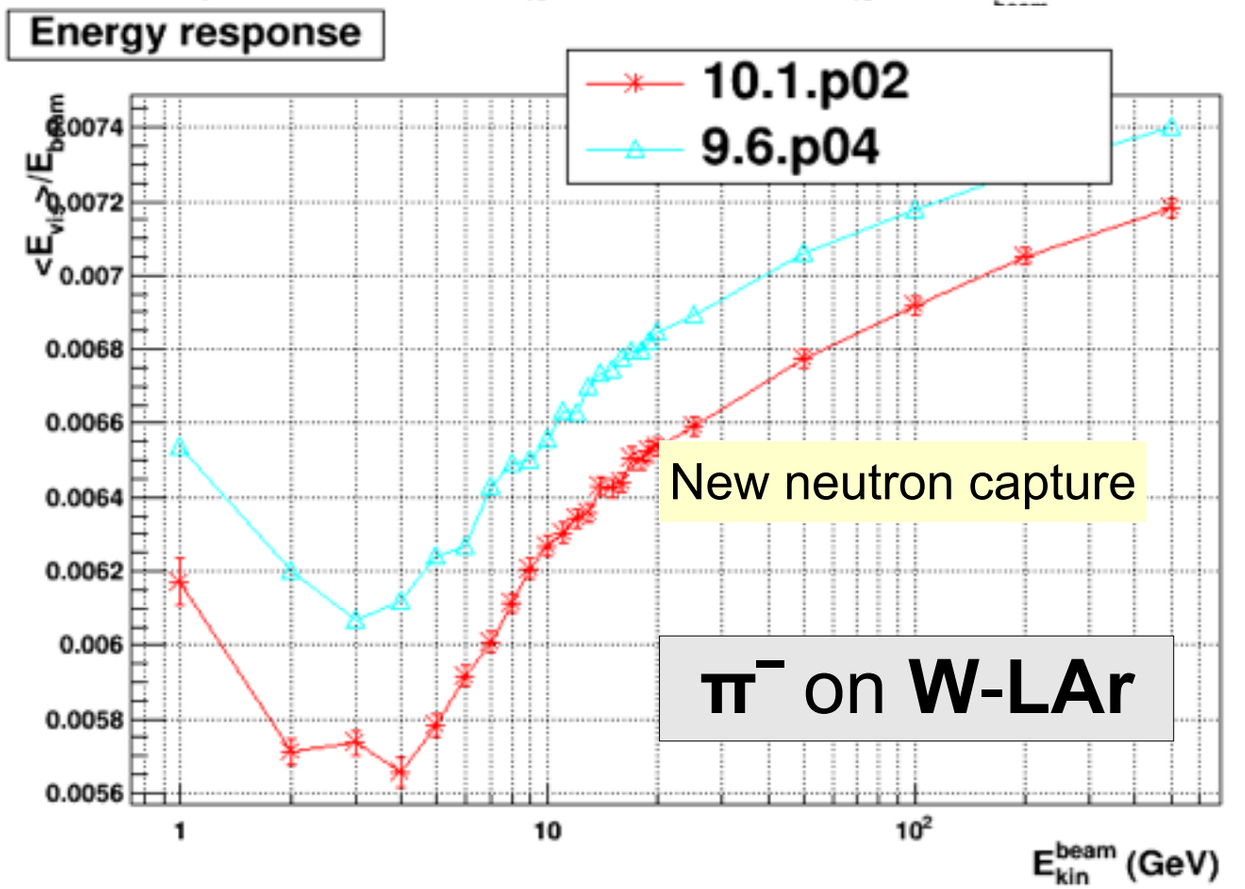
Hadronic showers (G4 10.0 & 10.1)

- **FTFP_BERT** remains our recommended HEP physics list
- Stable hadronic showers

  Tungsten calorimeters are the **exception**
  - Lower response and narrower lateral shape - due to the improved neutron capture (xsec & final state)
  - Now closer to FTFP_BERT_HP







Hadronics: FTF in G4 10.2

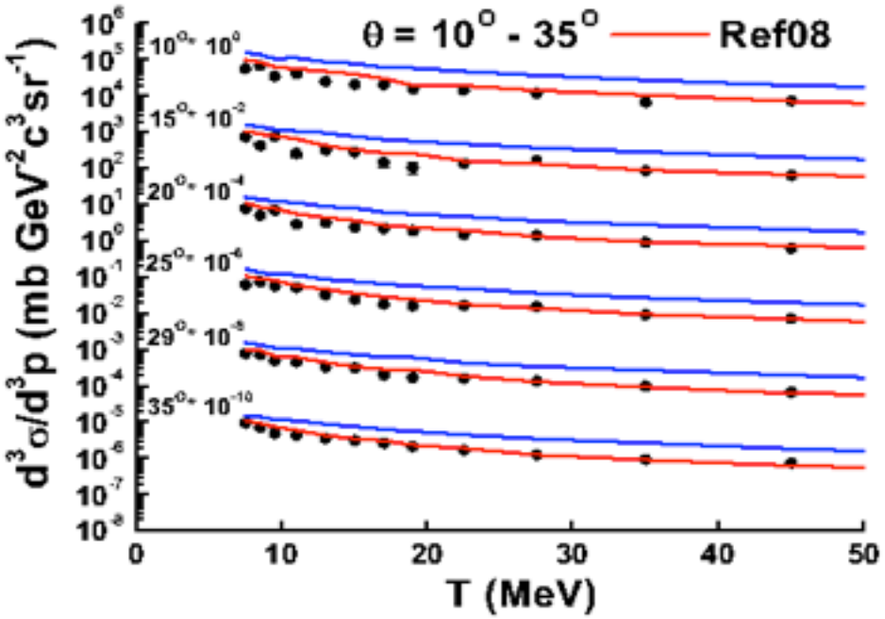
Revised preparation of the nuclear remnant by FTF model which it hands over to Precompound/evaporation

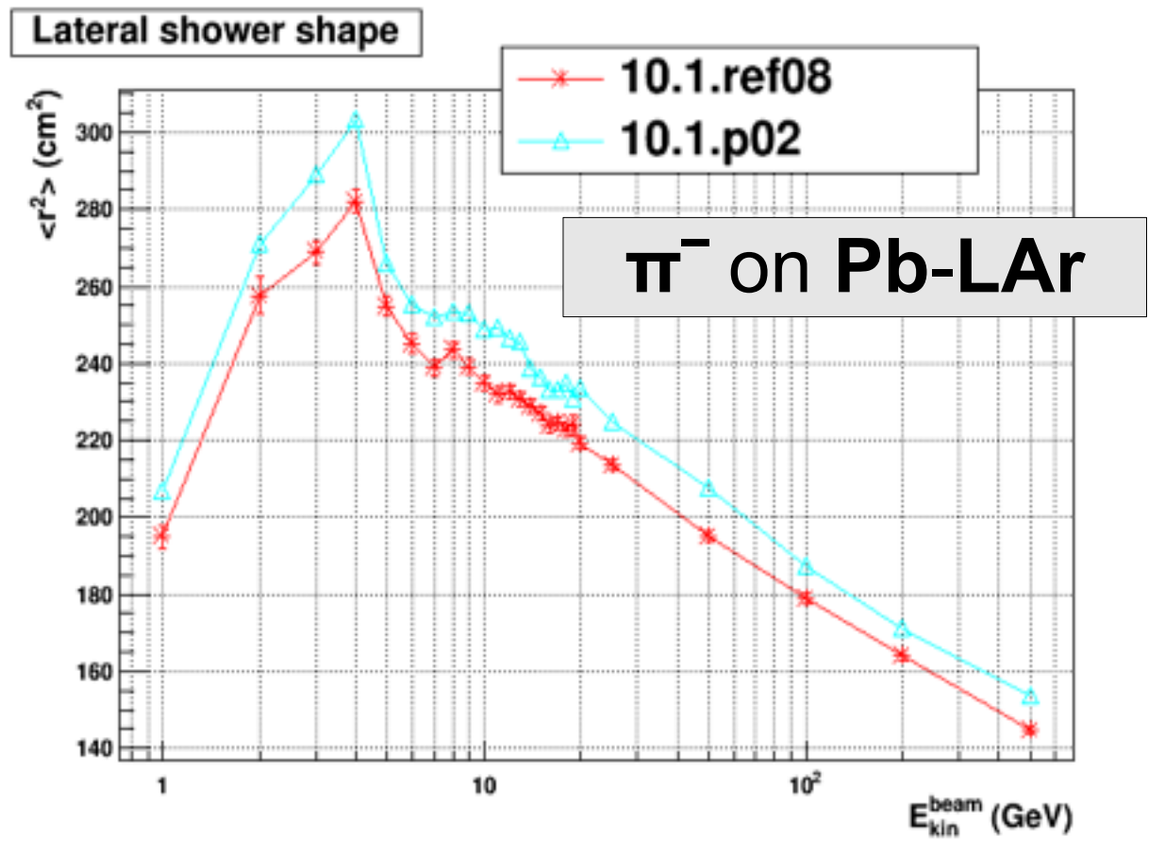
- Improved slow neutron production vs. thin-target ITEP data
- Little effect on the other thin-target observables

Significant (few %) impact on FTFP_BERT hadronic showers

- Higher energy response in Fe and Cu
- More optimistic energy resolution
- Narrower shower shapes

both likely in unwanted direction !





Hadronic Cross Sections (10.2)

Updated K+ and K- inelastic cross sections

- Motivated by LHCb requests on kaon cross section asymmetry

Recent fixes in NeutronXS cross sections

Brings FTFP_BERT showers closer to FTFP_BERT_HP ones for heavy absorbers (Pb, W)

Status of Precompound/de-excitation (10.2)

Unified evaporation cross sections, Preco & de-excitation

New structure of gamma levels shared by all de-excitation

- FermiBreakUp and GEM use same PhotonEvaporation

New gamma de-excitation model

- Planned to replace the current one
- Option to switch on gamma correlation (default=off)

Geometry in release 10.2

- Removed internal USolids module
  - Enabled use of external versioned VecGeom/USolids library
  - First version of shapes with vectorised versions from VecGeom
    - Generated some discussion on last day - questions about maintenance of G4Solids and retrofitting performance improvement to G4Solids
- Enable further volumes in MT mode
  - Divisions, and parameterisation by solids type
- Fix in G4MultiLevelLocator for keep consistent candidate intersection

BEYOND

- Several new young researcher collaborators
  - working on areas for needs of their community
  - radioactive decay - rare event experiments
- Many discussions - in parallel sessions, small groups, ..
  - figured out next steps,

Upcoming Schedule

Minor release: geant4-10-02

- Final date: **4 December 2015**
- October 23rd - GROUP-1 categories deadline
  - configuration, analysis, global, materials, graphics_reps, intercoms, geometry/management, particles, track
- October 30th - GROUP-2 categories deadline
  - rest of geometry, processes (management, transportation, electromagnetic, scoring), digits_hits, tracking
- November 6th - GROUP-3 categories deadline

10.2 Testing and Documentation

- **November 27th**
  - End of Integration/System Testing
  - End of User Documentation update
- November 13th – November 27th - Release phase
  - Scheduled global migrations
  - QA and validation/GRID
  - User documentation packaging
  - Libraries preparation and distribution

THE END

CMAKE

Modularization: Sketch of CMake API

```
# sources.cmake
include(Geant4CMakeAPI)

geant4_add_module(G4MyModule
  PUBLIC_HEADERS G4MyModule.hh
  SOURCES        G4MyModule.cc
  )

geant4_module_link_libraries(G4MyModule
  PUBLIC  G4globman G4csg
  PRIVATE G4intercoms ZLIB::ZLIB
  )

geant4_module_compile_definitions(G4MyModule
  PUBLIC $<$<CONFIG:Debug>:DEBUGMYMODULE>
  )
```

Managing Data Libraries

- We need to look at how data libraries are read/located
- Environment variable based setup awkward with too many potential points of failure
- Access via lots of small files has/is causing issues for MT performance?
- Let's use this week to discuss ideas
  - Slide notes include some of mine

Summary

- CMake system/documentation in good shape for 10.2
- Please come to Parallel 6b and Plenary 7 for C++11!
- Two major Software Management tasks for 10.3:
  - *Library modularisation*
  - *Data library location and format*
- Let's start the discussion on these during this week!

Physics lists

Geant4 10.1

Switched on muon-nuclear by default in all physics lists

In all physics lists using neutron-HP, the production threshold range of proton is set to zero

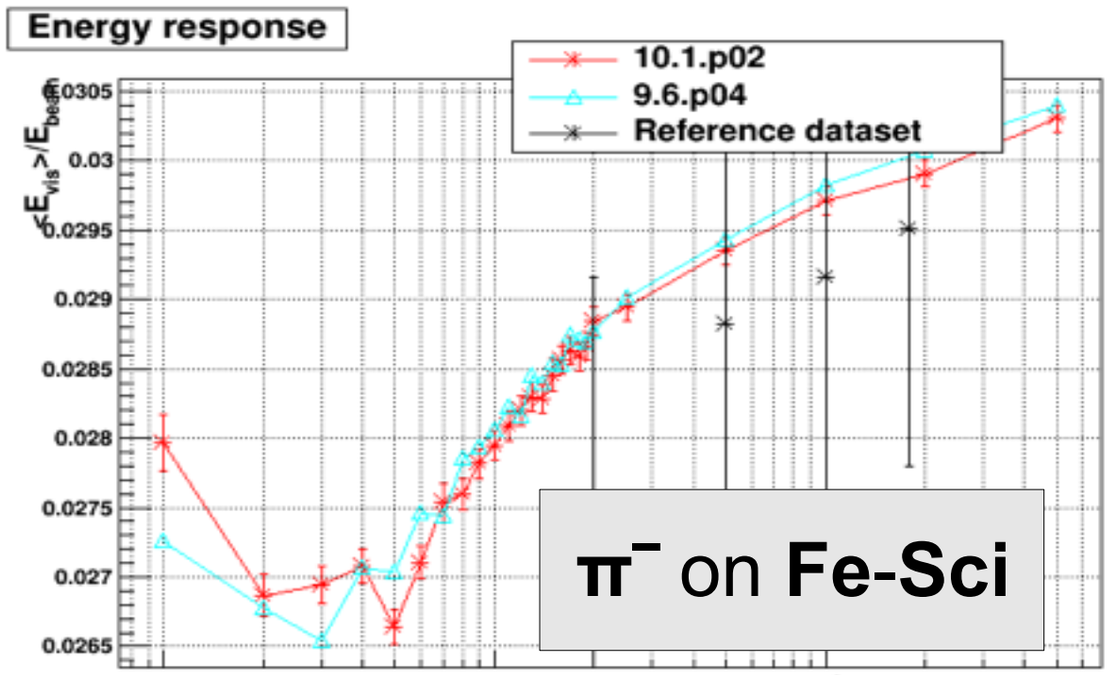
to simulate all recoils from elastic scattering

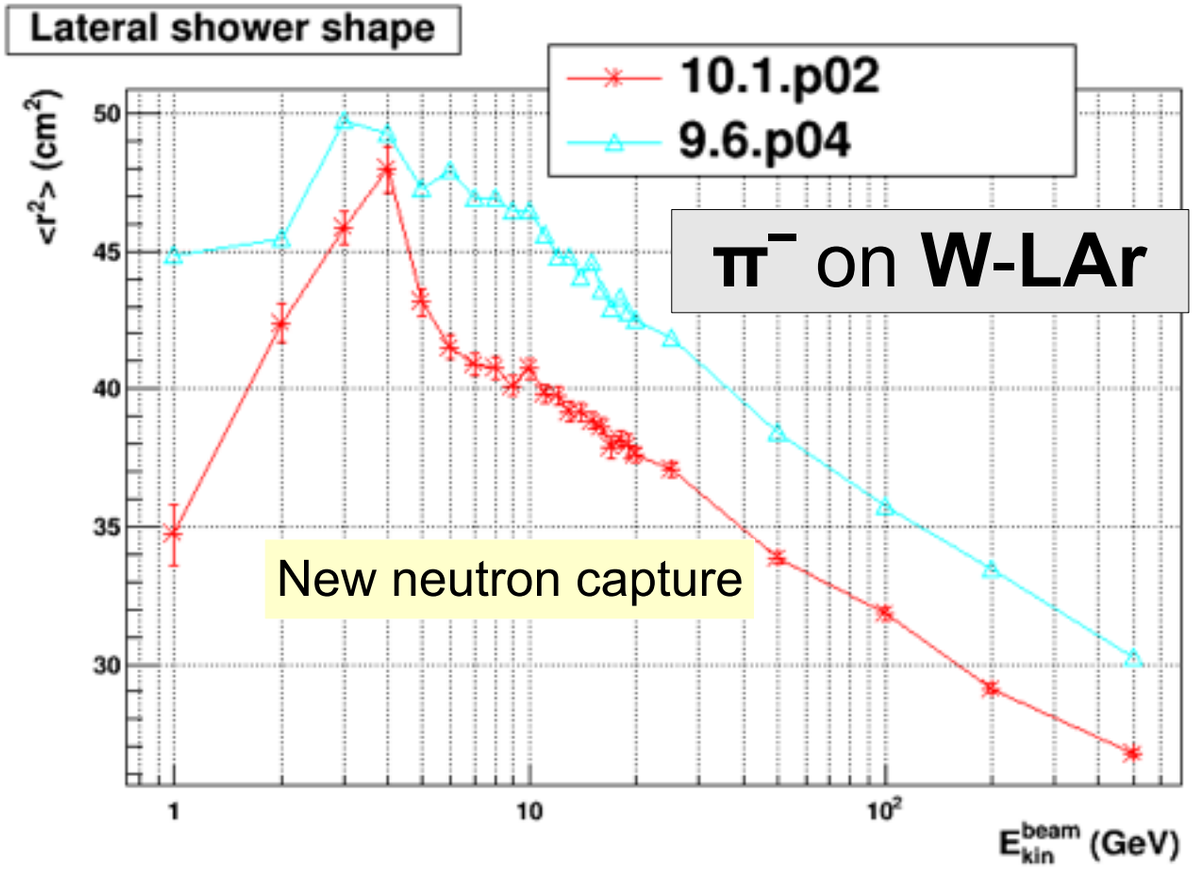
The physics lists based on INCLXX use it for pions, protons and neutrons up to 20 GeV

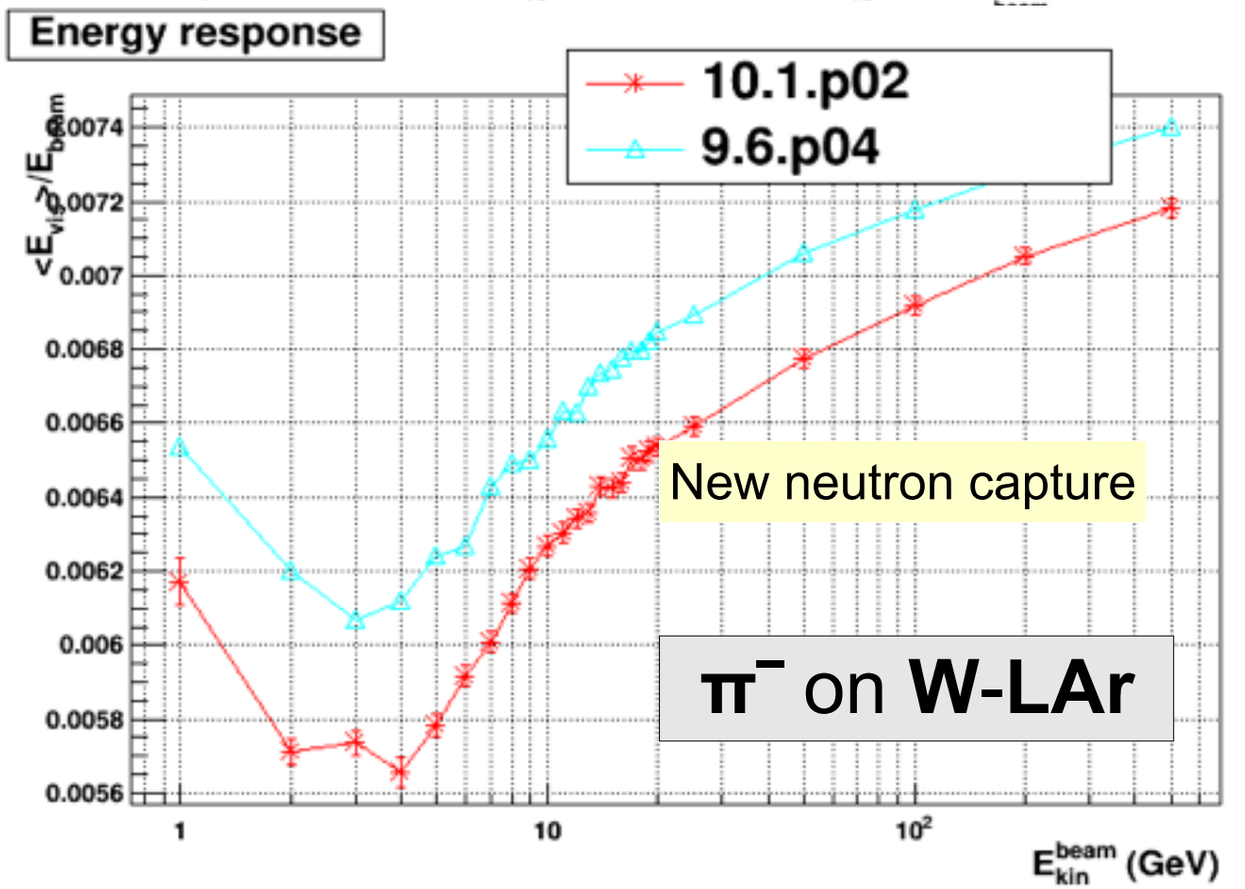
Hadronic showers (G4 10.0 & 10.1)

- **FTFP_BERT** remains our recommended HEP physics list
- Stable hadronic showers

  Tungsten calorimeters are the **exception**
  - Lower response and narrower lateral shape - due to the improved neutron capture (xsec & final state)
  - Now closer to FTFP_BERT_HP

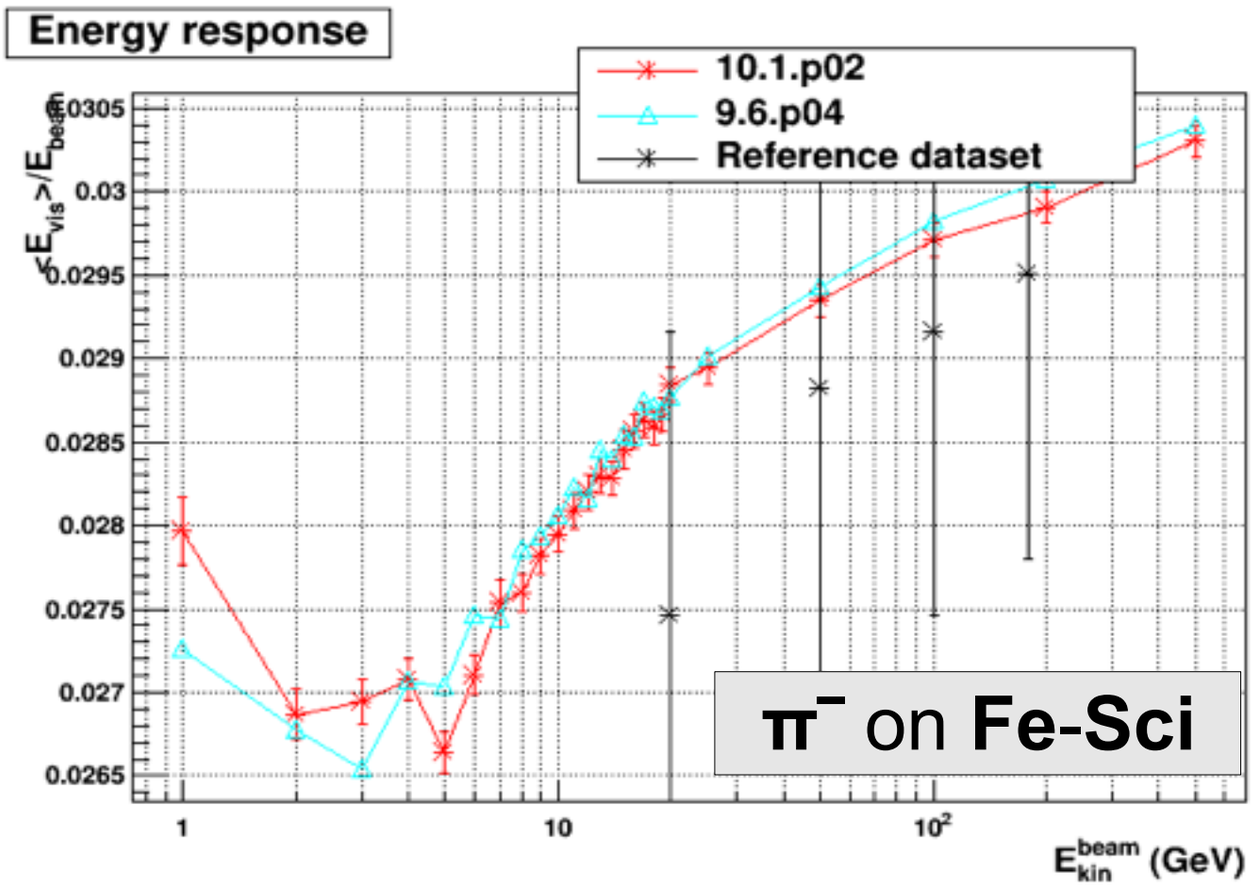


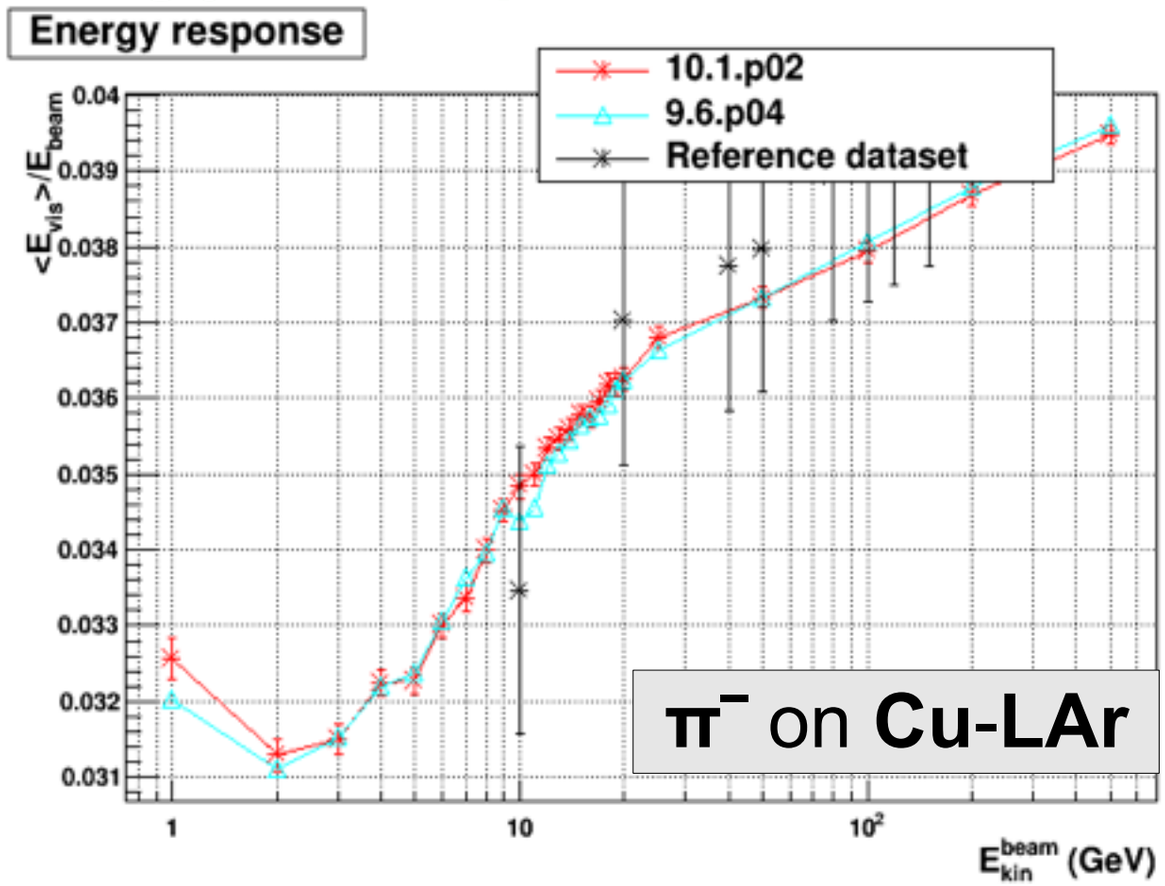


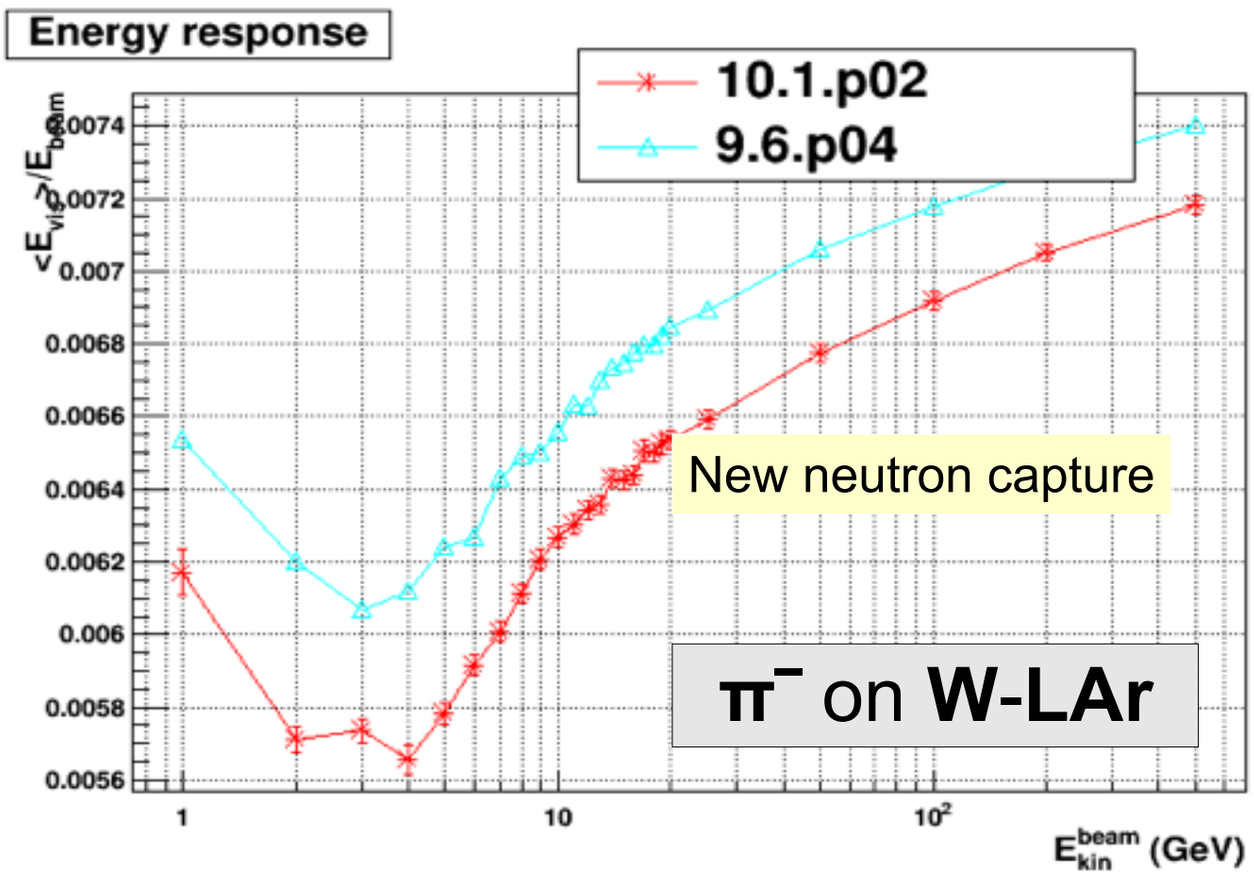


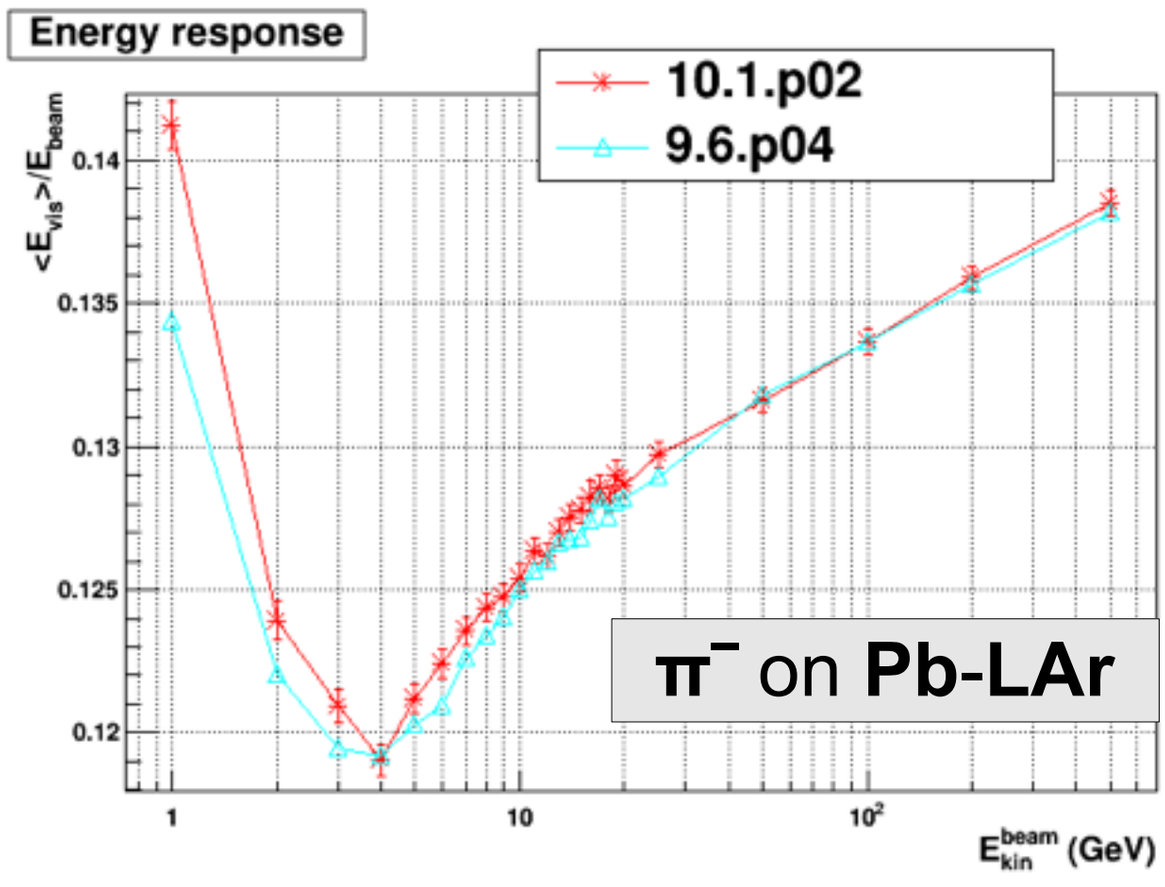
Shower variation G4 9.6.p04 to 10.1p02

FTFP_BERT : Energy Response

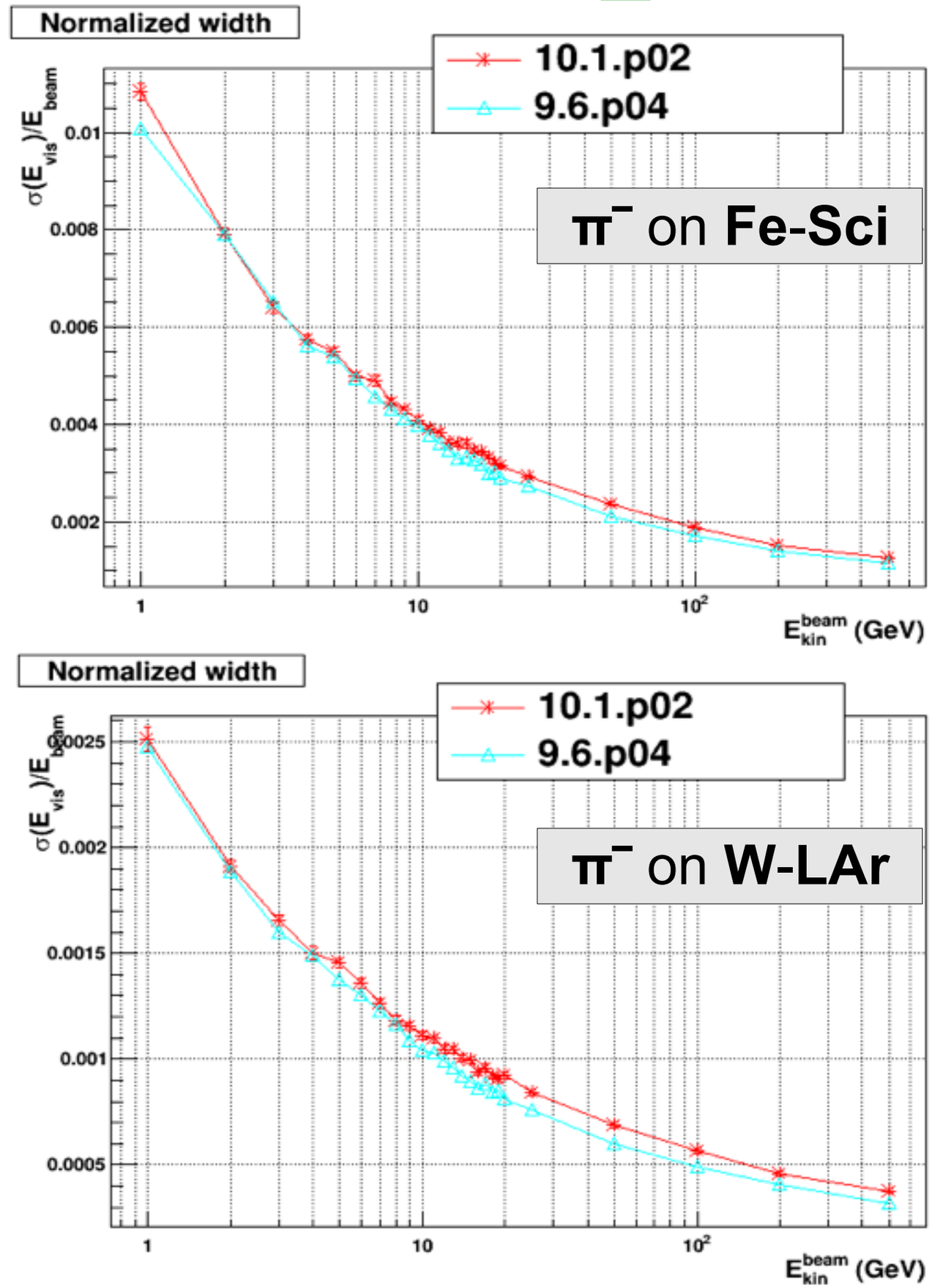


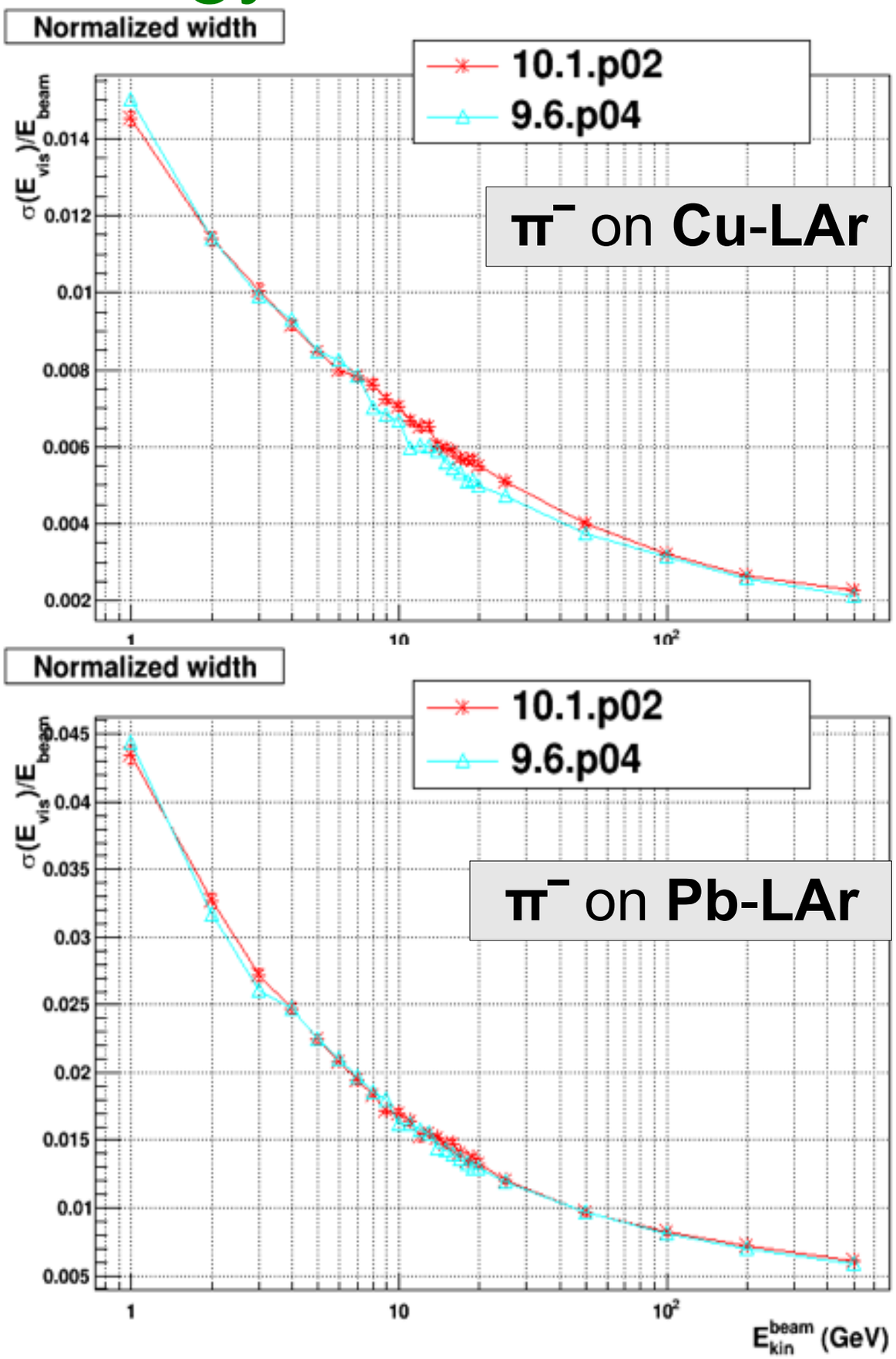




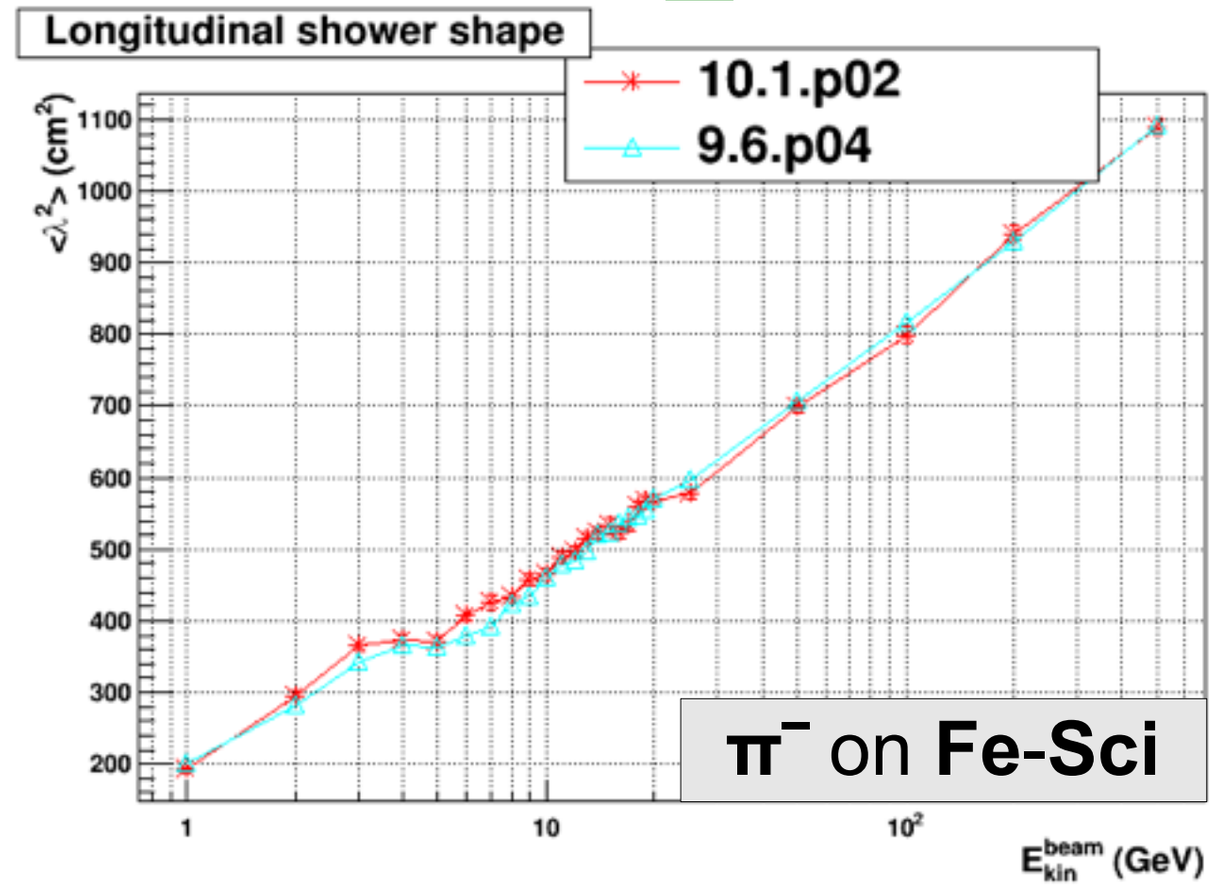


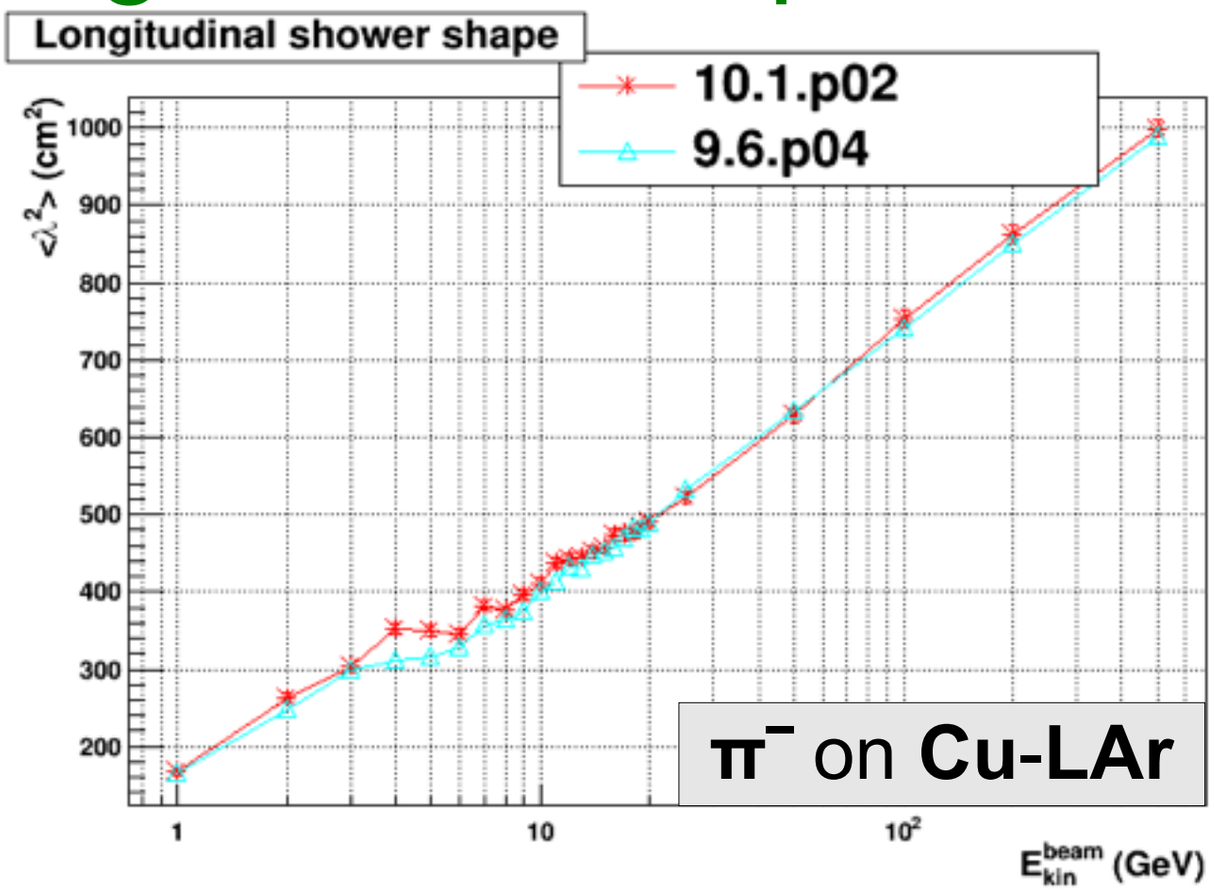
FTFP_BERT : Energy Width

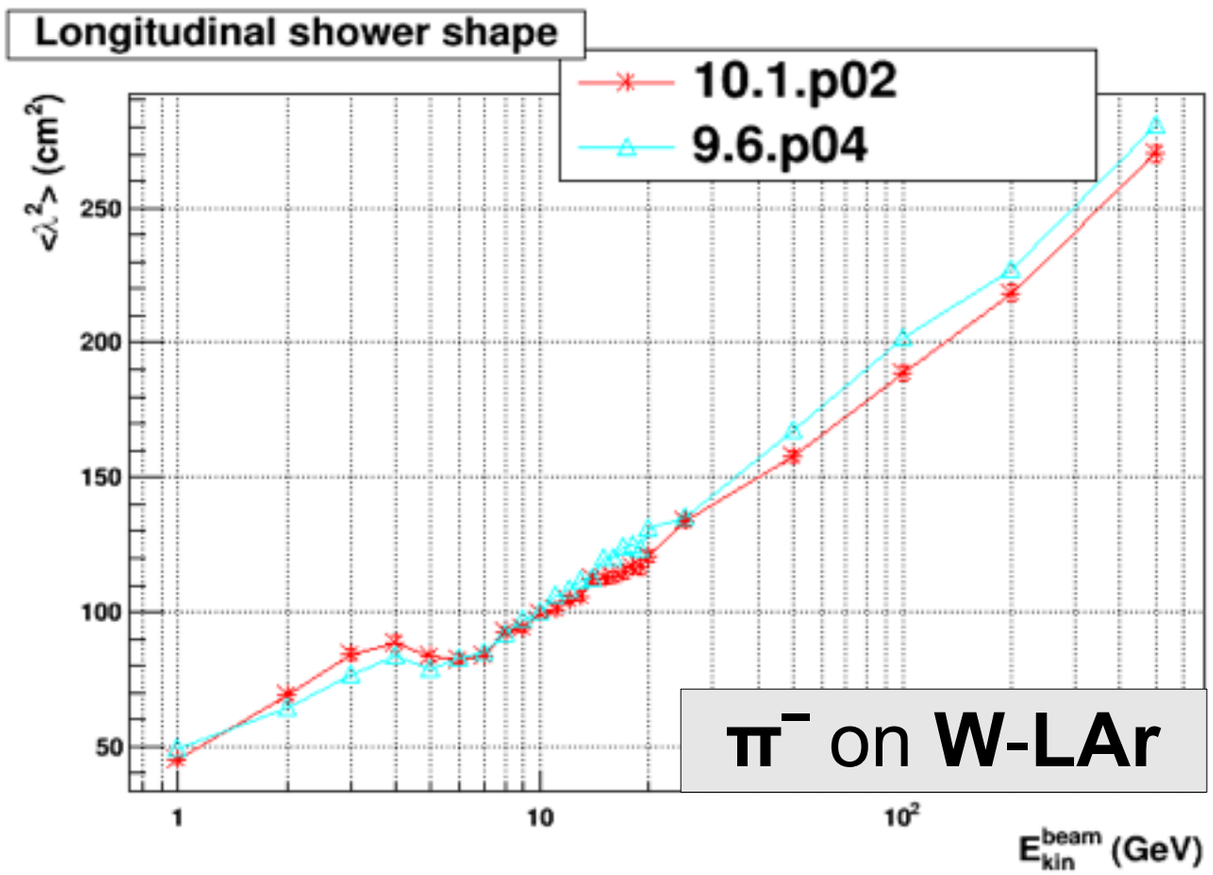


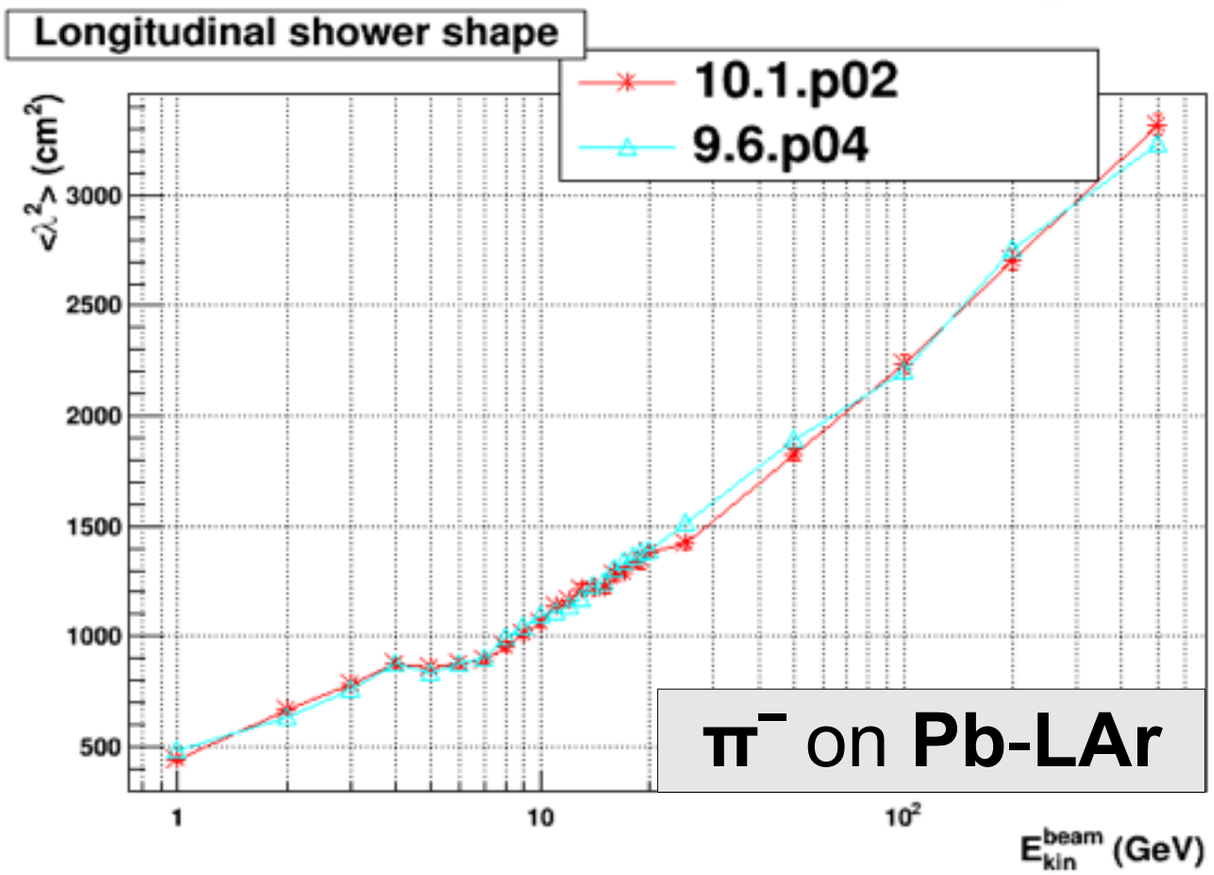


FTFP_BERT : Longitudinal Shape









FTFP_BERT : Lateral Shape