The simulation of the Dual-Readout Calorimeter for future collider experiments

Sanghyun Ko (sanghyun.ko@cern.ch)

Seoul National University

on behalf of the dual-readout calorimeter team in Korea

08 Jun 2021

Dual-readout calorimeter

The dual-readout calorimetry

- The major difficulty of measuring energy of hadronic showers comes from the fluctuation of EM fraction of a shower, f_em
- f_em can be measured by implementing two different channels with different h/e response in a calorimeter

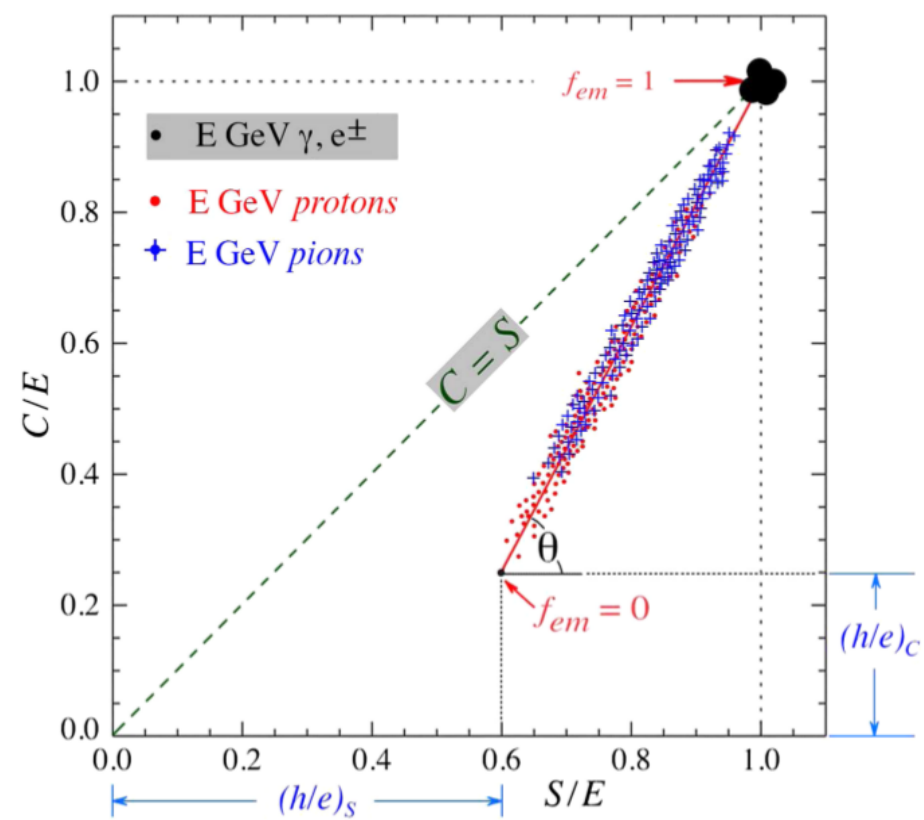
$$S = E[f_{em} + (\frac{h}{e})_S (1 - f_{em})],$$

$$C = E[f_{em} + (\frac{h}{e})_C (1 - f_{em})]$$

$$\cot\theta = \frac{1-(h/e)_S}{1-(h/e)_C} \equiv \chi,$$

$$f_{em} = \frac{(h/e)_C - (C/S)(h/e)_S}{(C/S)[1-(h/e)_S] - [1-(h/e)_C]}$$

$$E = \frac{S - \chi C}{1-\chi}$$



Energy measured from scintillation channel vs Čerenkov channel for EM particle, π & p.

- Dual-readout calorimeter may offer high-quality energy measurement for both EM particles and hadrons
- Excellent energy resolution for hadrons can be achieved by measuring f_em and correcting the measurement event-by-event
- Dual-readout fiber-sampling calorimetry is a key element of the IDEA detector concepts

Dual-readout calorimeter

The dual-readout calorimetry

- The majo
showers c
shower, f_

- f_em can
channels

$S = E[f_{en}$

$C = E[f_{en}$

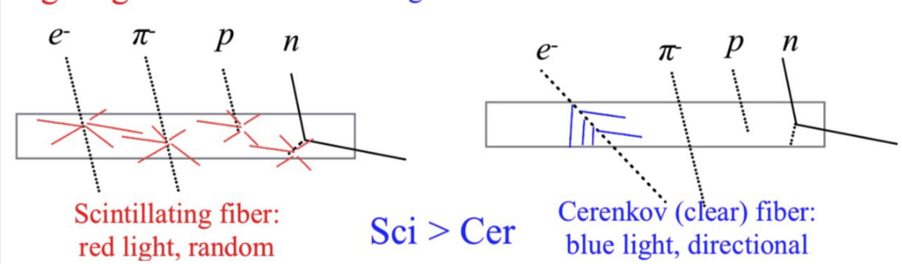
$f_{em} = \frac{}{(C/}$

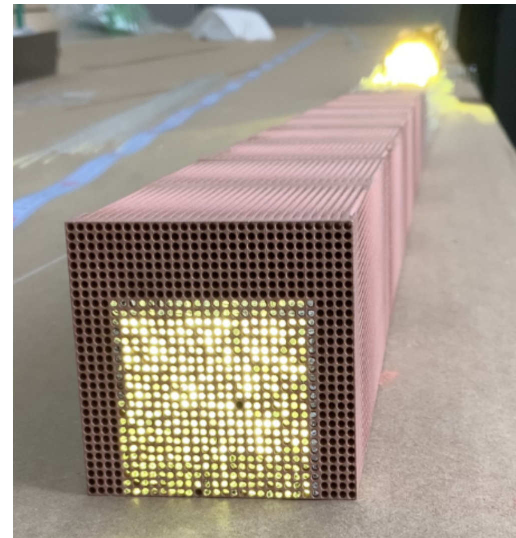
- Dual-reac
measurer

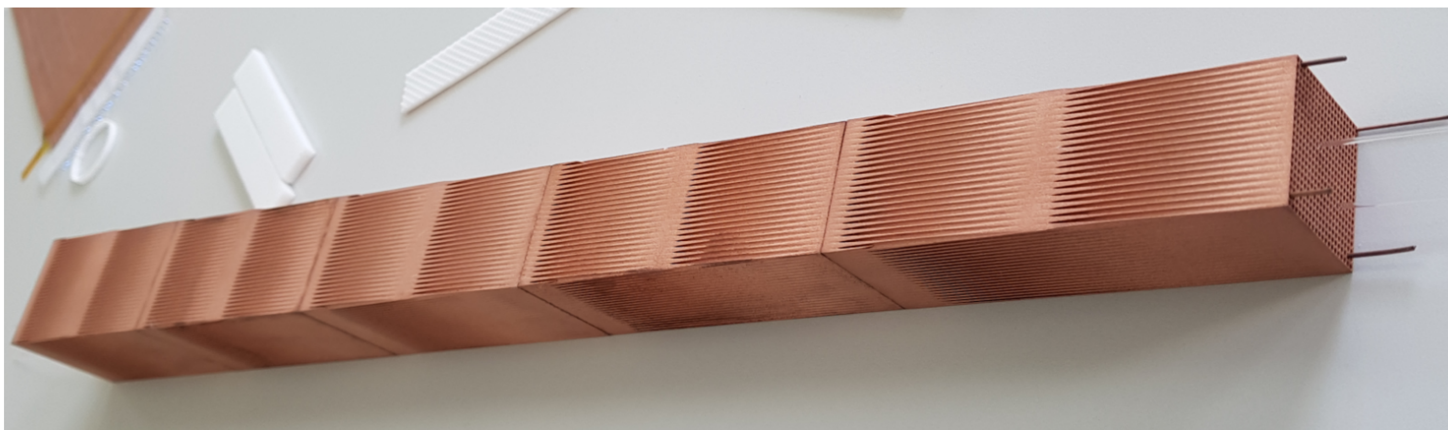
- Excellent
measurin
by-event

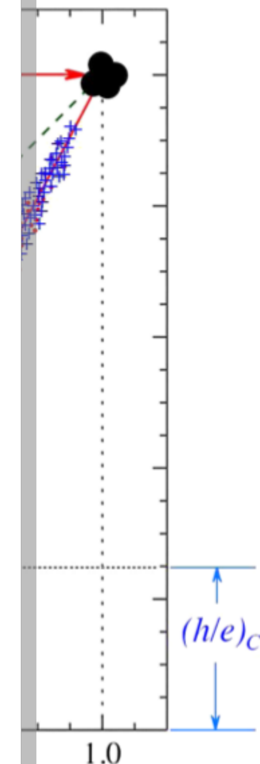
See presentations from Hwidong last week
for organizations & Hardware efforts

→ https://indico.bnl.gov/event/11917/#2-dual-readout-calorimetry









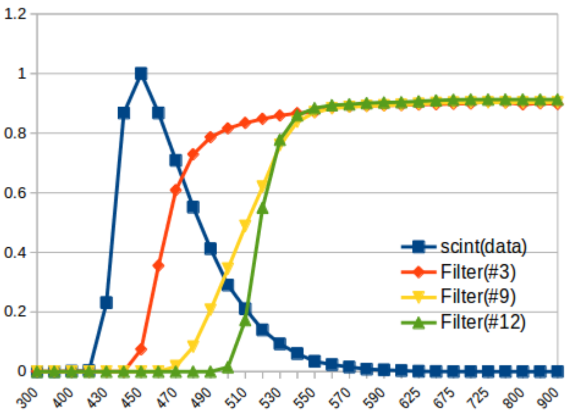
hannel vs
& p.

- Dual-readout fiber-sampling calorimetry is a key element of the IDEA detector concepts

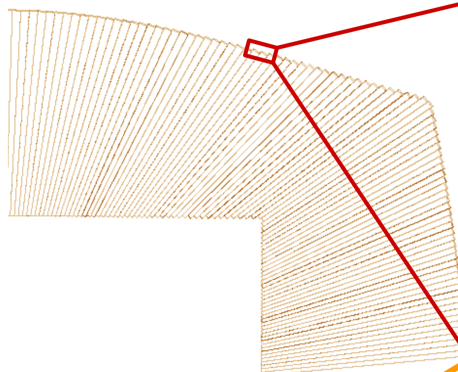
Optical properties of DRC

Properties of optical fibers [github]

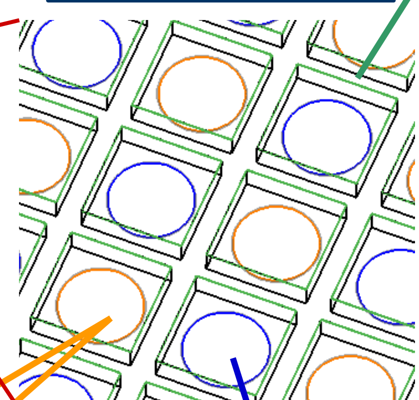
Transmission eff of filters



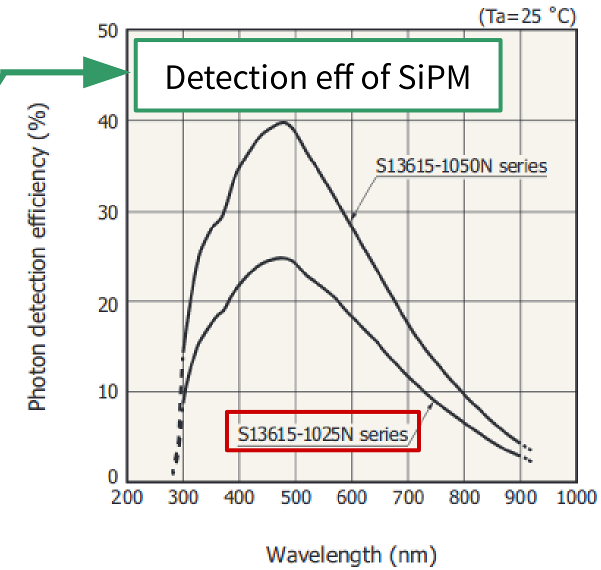
x-z view of DRC



Rear end of a tower

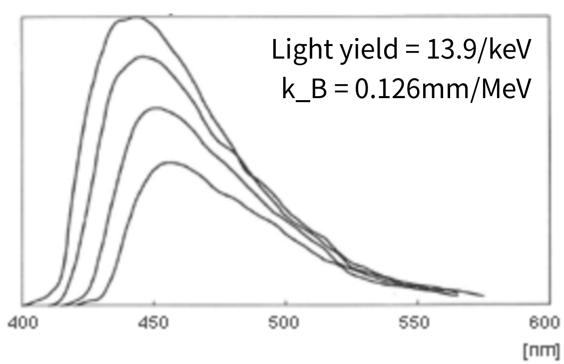


Detection eff of SiPM



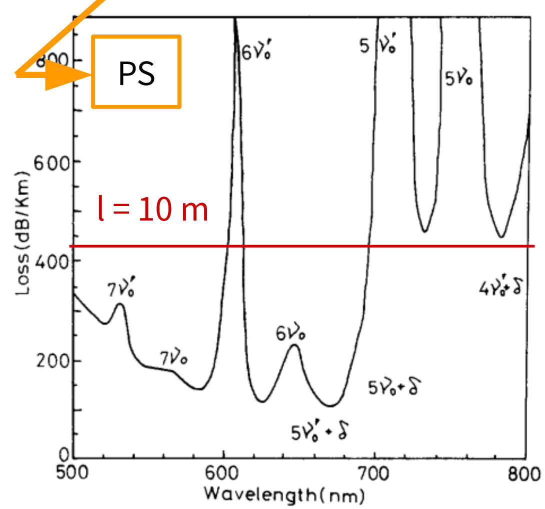
Attenuation loss diverges at 400nm → applied filter to S channel to mitigate it

Scintillation spectra of PS

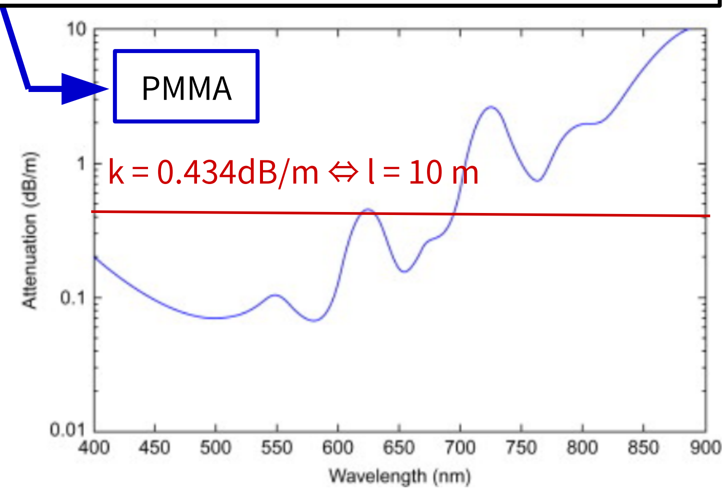


Attenuation loss of Polystyrene (PS) & PMMA

PS

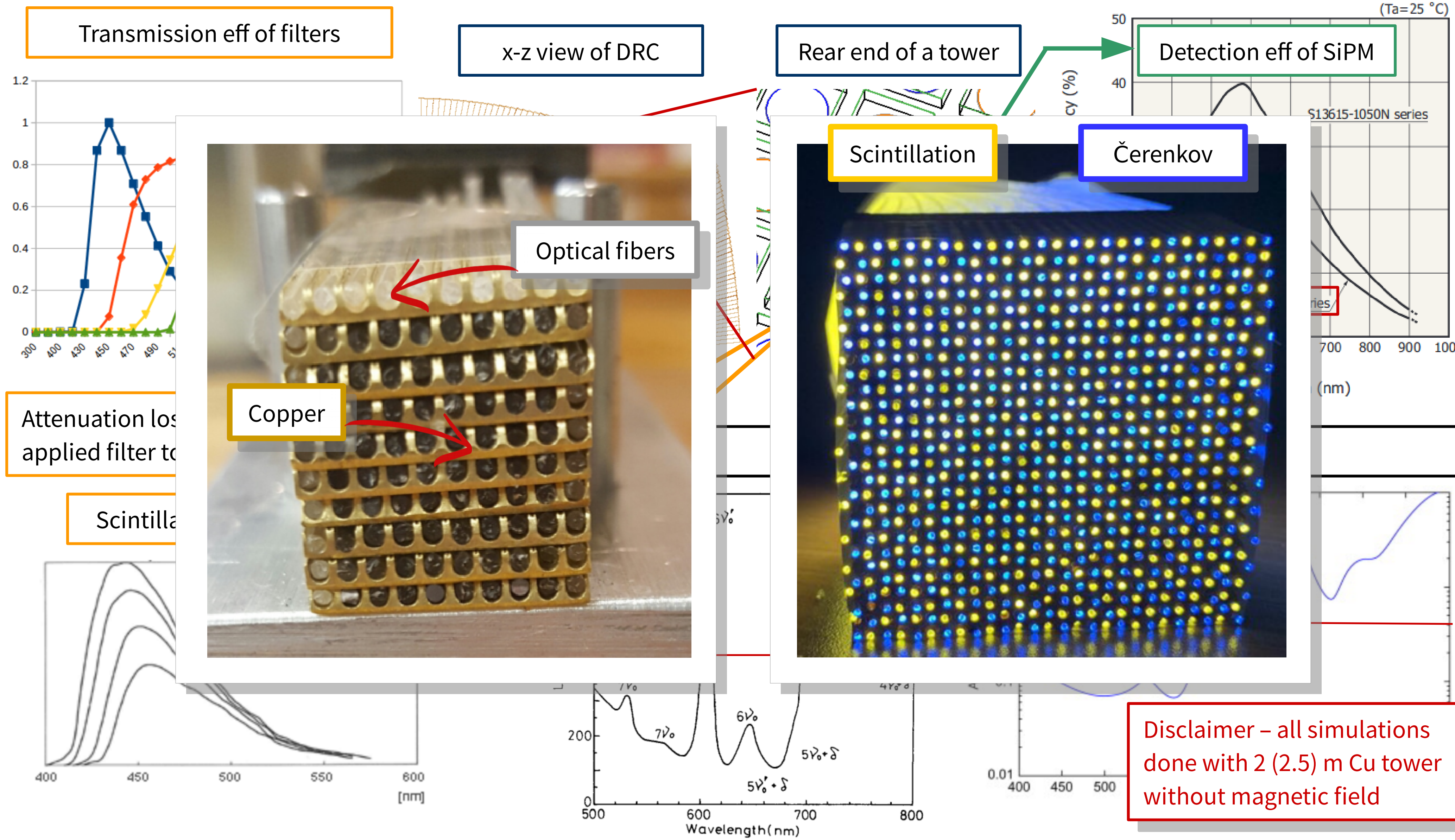


PMMA



Optical properties of DRC

Properties of optical fibers [github]



Disclaimer – all simulations done with 2 (2.5) m Cu tower without magnetic field

Speeding up optical photon tracking

Optical photons tracking in dual-readout calorimeter

- Full simulation of optical photon tracking explodes CPU cost: O(h)/evt

  O(10k) tracks/photon × O(10k) photons/GeV ≃ O(100M) tracks/GeV

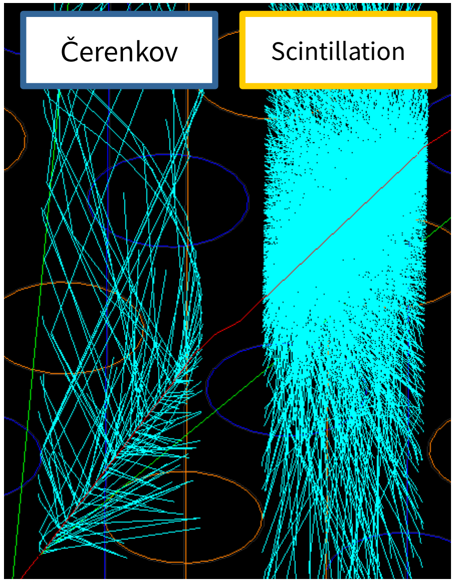
  → Due to total internal reflections → Due to scintillation yield

- Having reasonable CPU cost is important

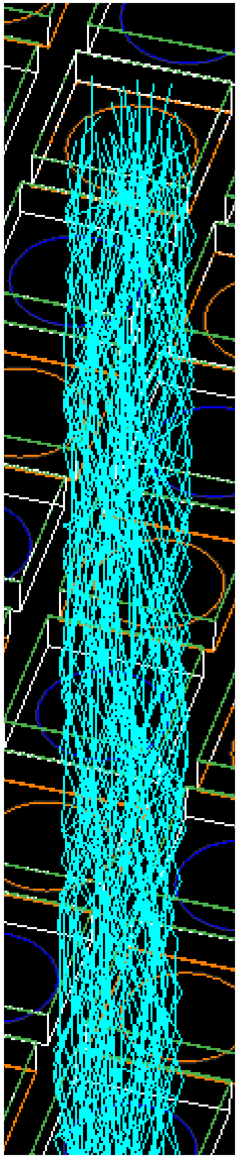
  Conventional solution is killing optical photons after counting

  → not an optimal solution for dual-readout calorimeter

  - Numerical aperture is important for the yield of Čerenkov signal
  - Light attenuation of fibres & timing of optical photons



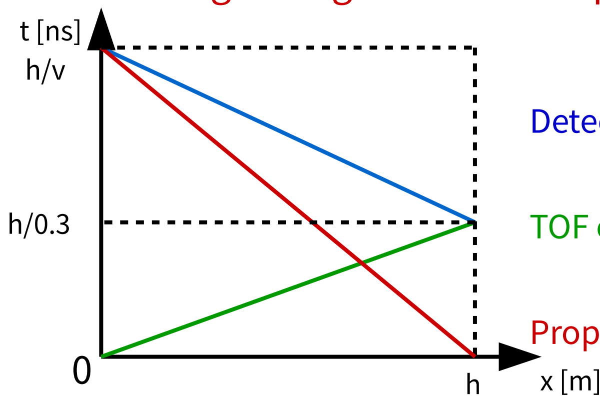
Important for a longitudinally unsegmented calorimeter



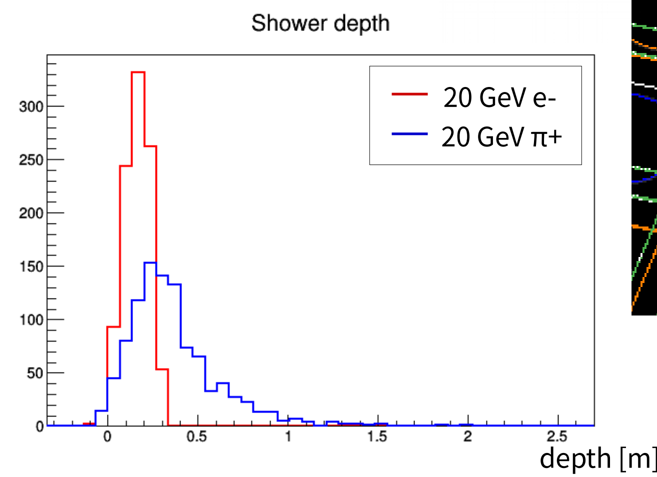
Measuring depth with dual-readout calorimeter

- Depth x can be represented as a function of detection time

  → simulating timing structure of optical photons is essential



Detection time
=
TOF of π+ in vacuum/tower
+
Propagation time of optical photons

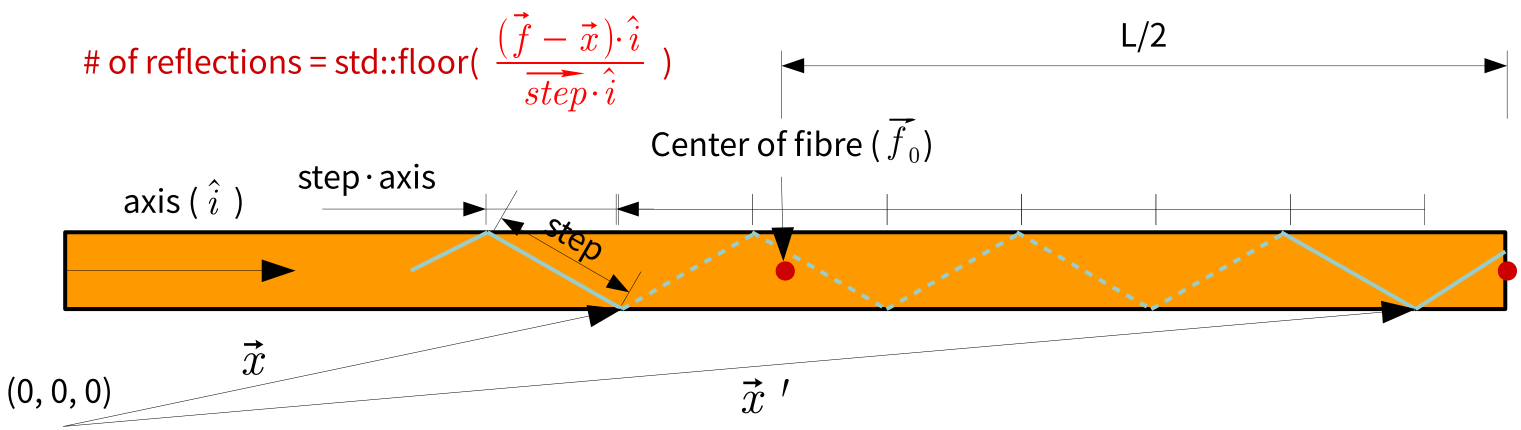


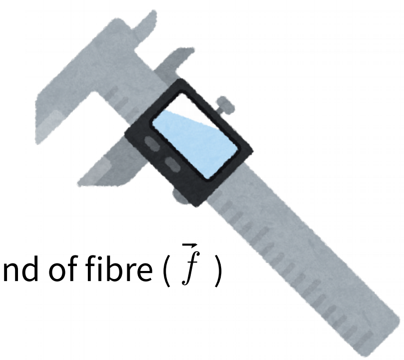
Speeding up optical photon tracking

Developing fast simulation for optical photon tracking

- Simulating photon propagation is necessary, but dominates CPU consumption
- Yet, propagation of optical photons in fibres can be estimated, skipping full tracking

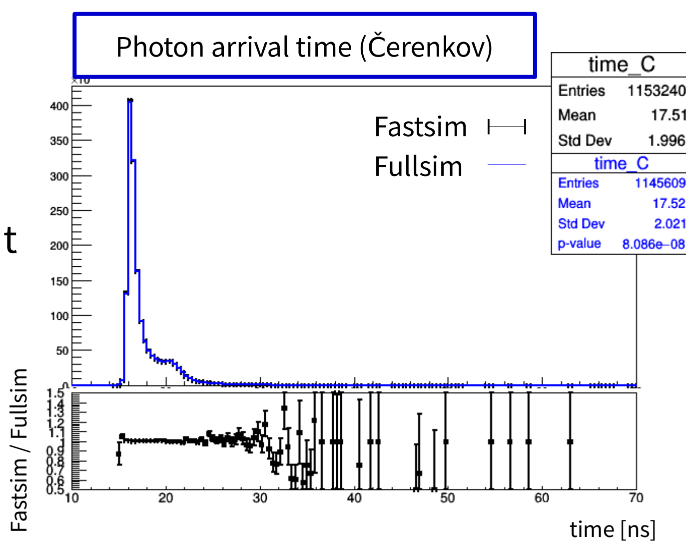
  → Developing a fast simulation module presented at GEANT4 R&D meeting [link]

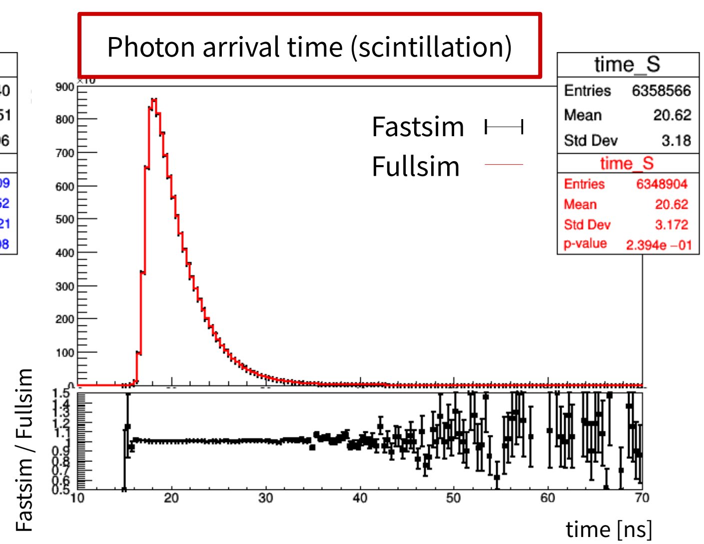




- Preliminary Fastsim model shows excellent agreement with Fullsim
- Takes ~ 4 mins to simulate an event of 20 GeV e-

  → more efforts for further improvement on-going





Calibration

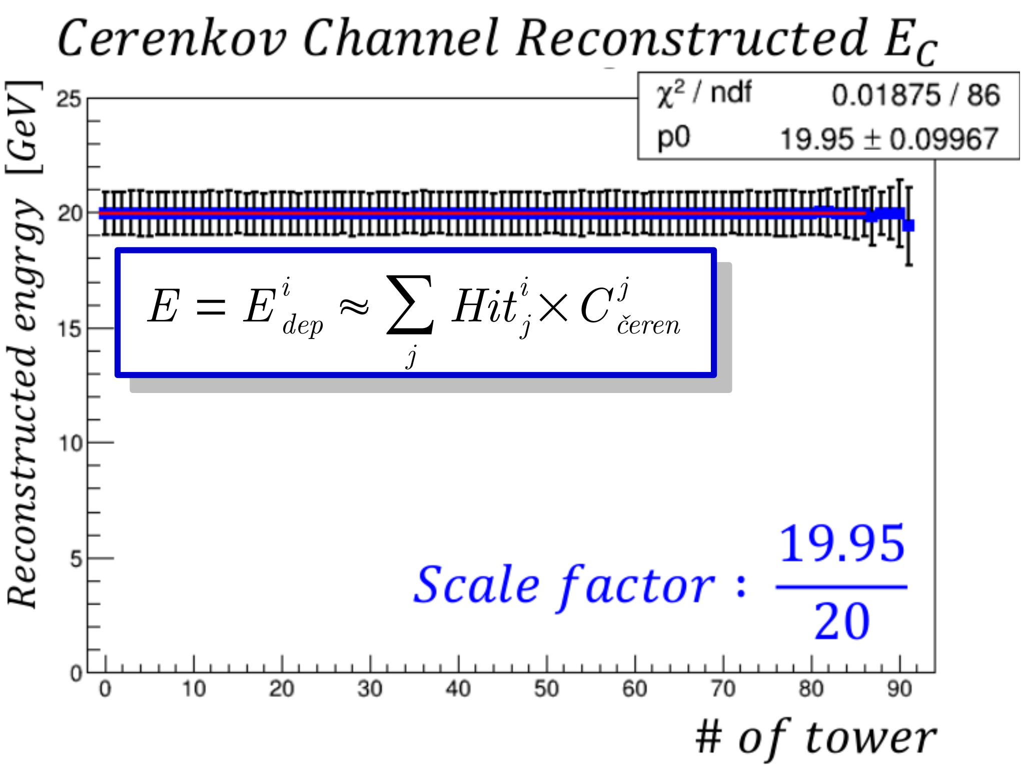
Calibration using 20 GeV e-

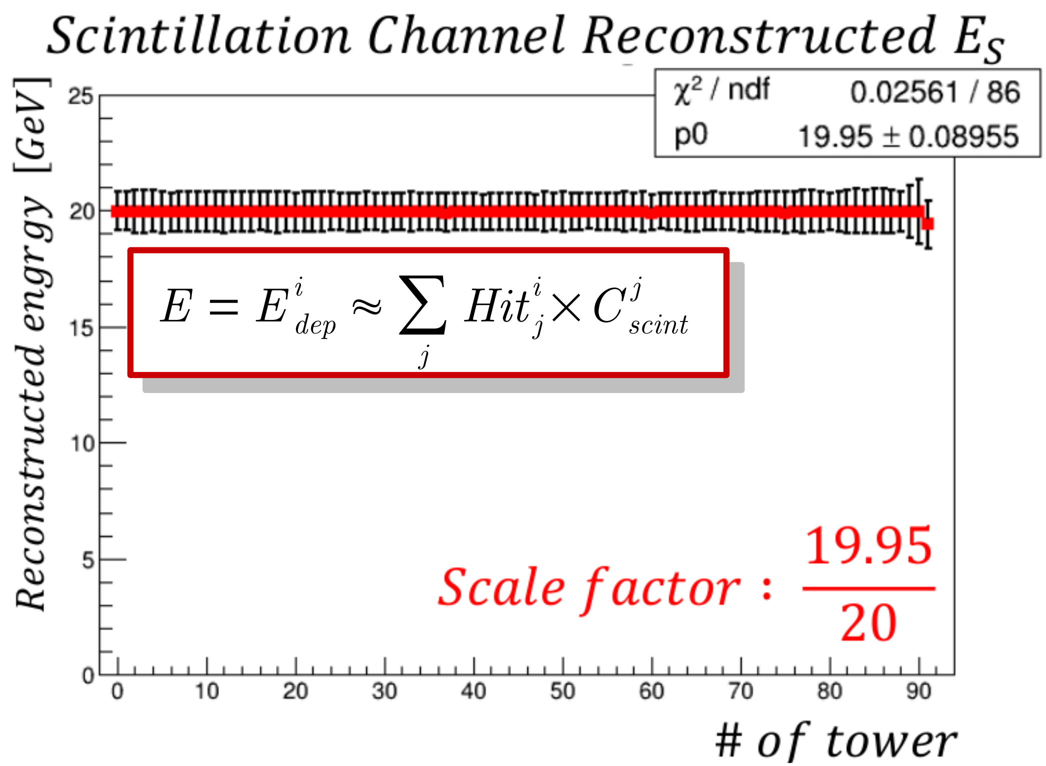
- Measure Energy deposit, scintillation ch. p.e. and Čerenkov ch. p.e. at i-th tower (0^{th} - 91^{st})
- Energy can be expressed as a linear combination with simulations of 92 towers

  → Estimate calibration constants

$$Energy = \sum_{i=0}^{92} Hit_{i^{th}\ tower} \times Calibration\ constant^{i^{th}\ tower}$$

$$\Rightarrow \begin{bmatrix} E_{dep}^{0} \\ E_{dep}^{1} \\ \vdots \\ E_{dep}^{90} \\ E_{dep}^{91} \end{bmatrix} = \begin{bmatrix} Hit_0^0 & Hit_0^0 & \cdots & Hit_{90}^0 & Hit_{91}^0 \\ Hit_0^1 & Hit_1^1 & & Hit_{90}^1 & Hit_{91}^1 \\ \vdots & & \ddots & \vdots & \\ Hit_0^{90} & Hit_1^{90} & \cdots & Hit_{90}^{90} & Hit_{91}^{90} \\ Hit_0^{91} & Hit_1^{91} & & Hit_{90}^{91} & Hit_{91}^{91} \end{bmatrix} \begin{bmatrix} C^0 \\ C^1 \\ \vdots \\ C^{90} \\ C^{91} \end{bmatrix}$$



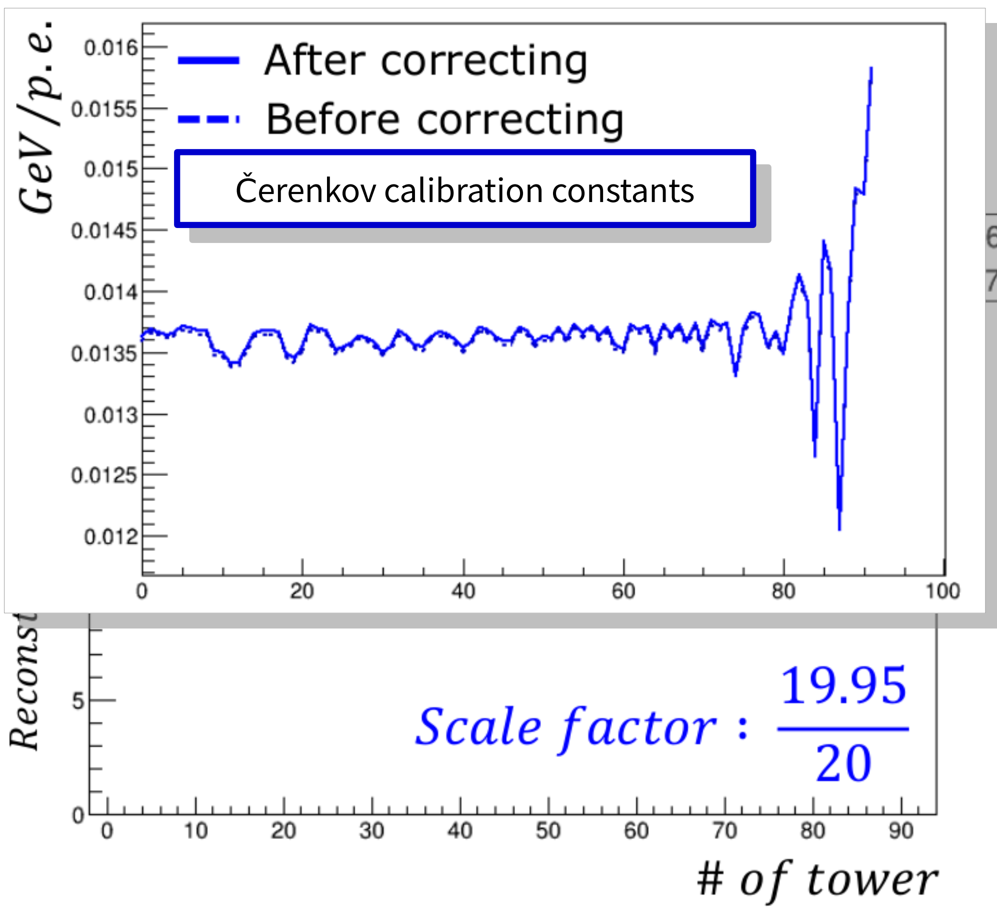


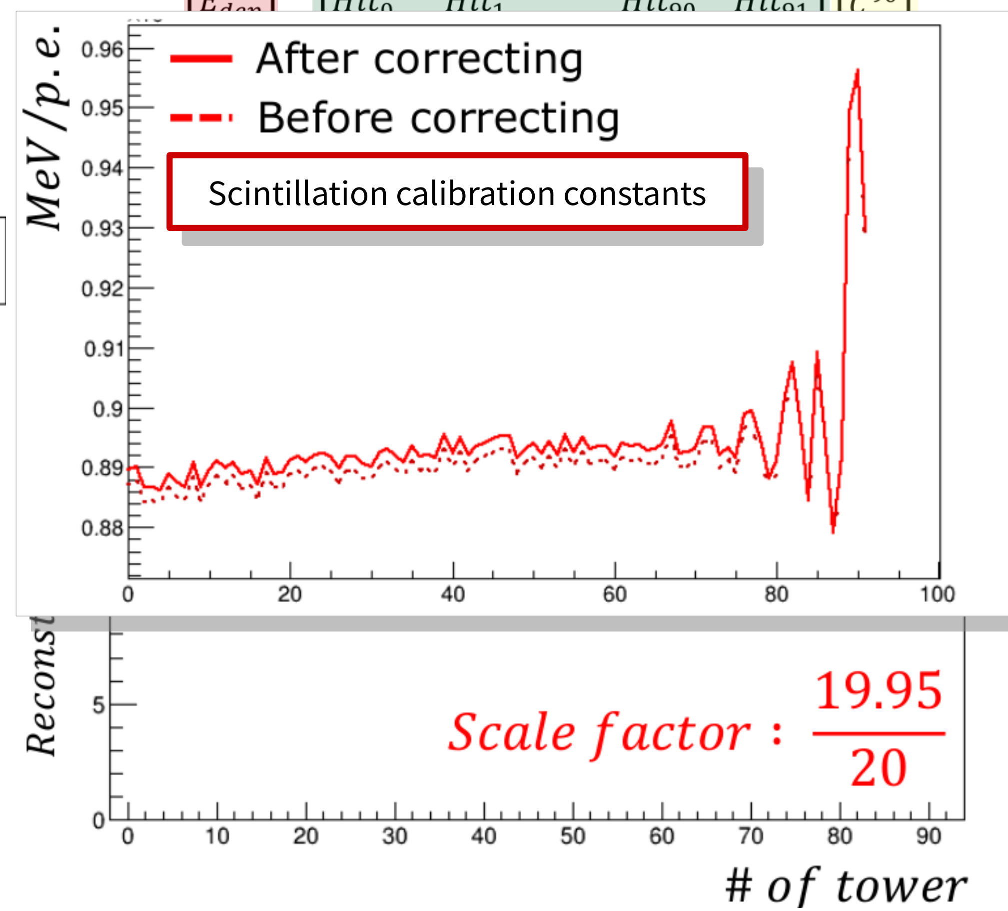
Calibration using 20 GeV e-

- Measure Energy deposit, scintillation ch. p.e. and Čerenkov ch. p.e. at i-th tower (0^{th} - 91^{st})
- Energy can be expressed as a linear combination with simulations of 92 towers

$$Energy = \sum_{i=0}^{92} Hit_{i^{th}\ tower} \times Calibration\ constant^{i^{th}\ tower}$$

$$\Rightarrow \begin{bmatrix} E^0_{dep} \\ E^1_{dep} \\ \vdots \\ E^{90}_{dep} \end{bmatrix} = \begin{bmatrix} Hit^0_0 & Hit^0_0 & \cdots & Hit^0_{90} & Hit^0_{91} \\ Hit^1_0 & Hit^1_1 & & Hit^1_{90} & Hit^1_{91} \\ \vdots & & \ddots & \vdots & \\ Hit^{90}_0 & Hit^{90}_1 & & Hit^{90}_{90} & Hit^{90}_{91} \end{bmatrix} \begin{bmatrix} C^0 \\ C^1 \\ \vdots \\ C^{90} \end{bmatrix}$$

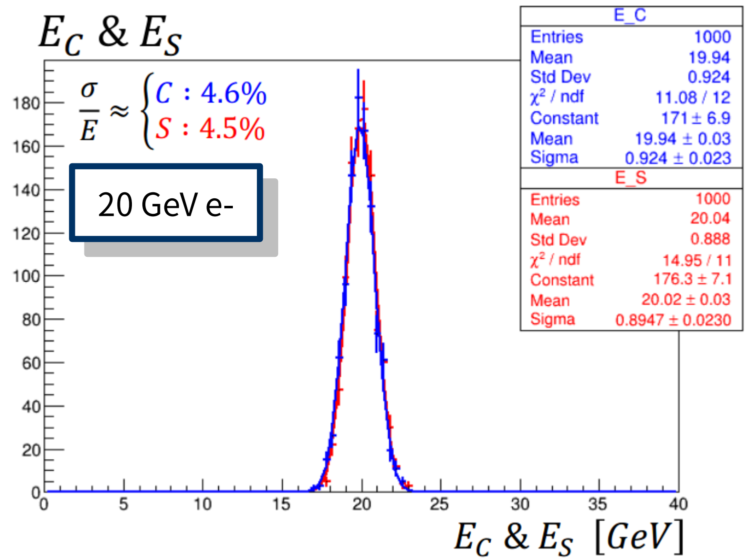


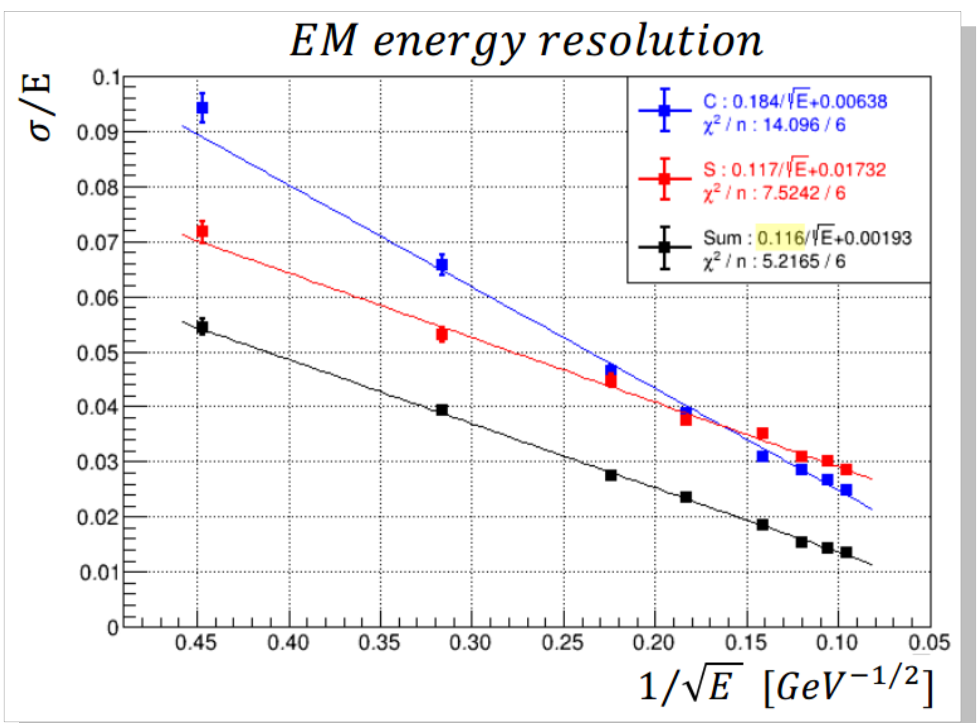


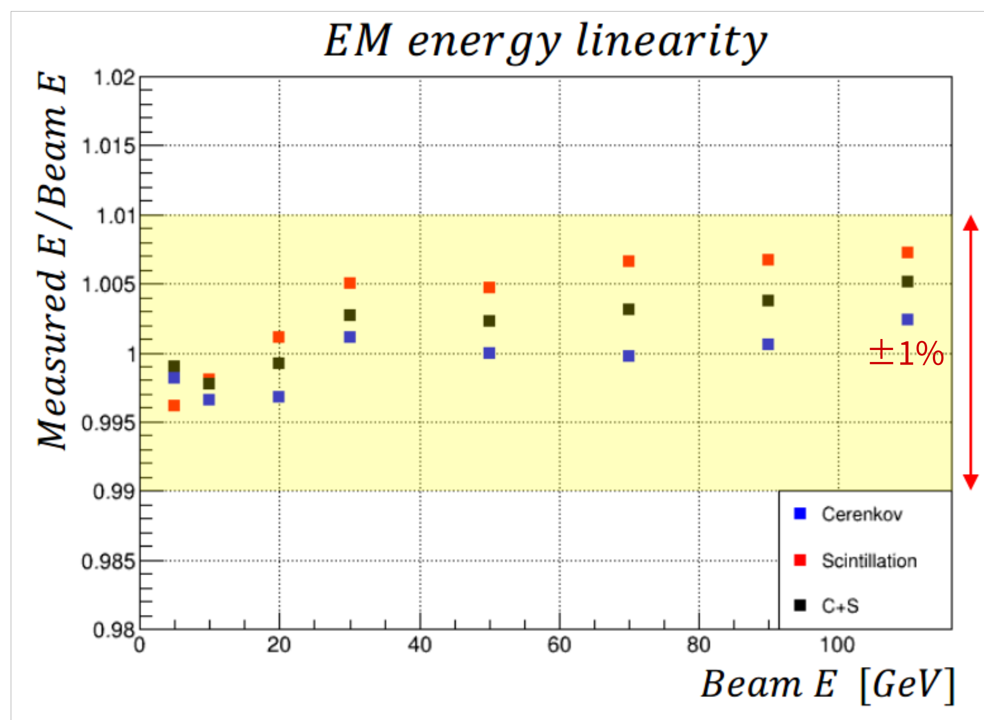
EM energy resolution

- Measured with 8 different energy of e- beams
  → 5, 10, 20, 30, 50, 70, 90, 110 GeV
- Scaled to 1/√E
  → stochastic & constant term estimated by linear fit

- Stochastic term ~ 12% (combined channel)
- Linearity within ±1% level for both and combined channel





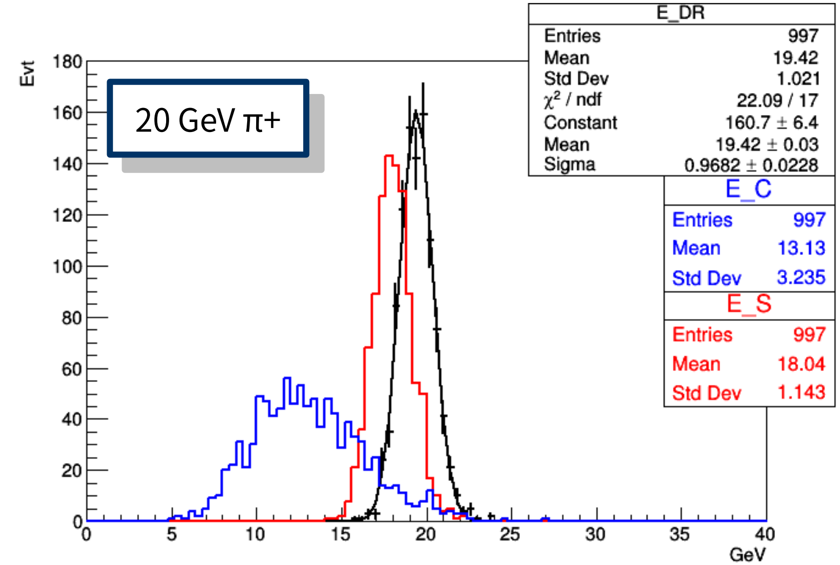


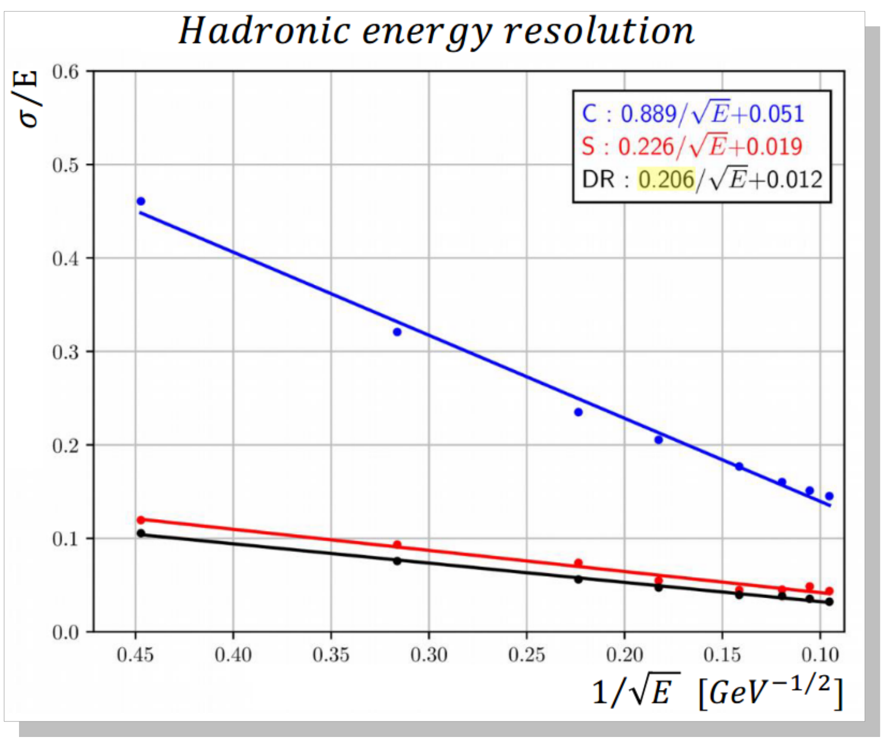
Hadronic energy resolution

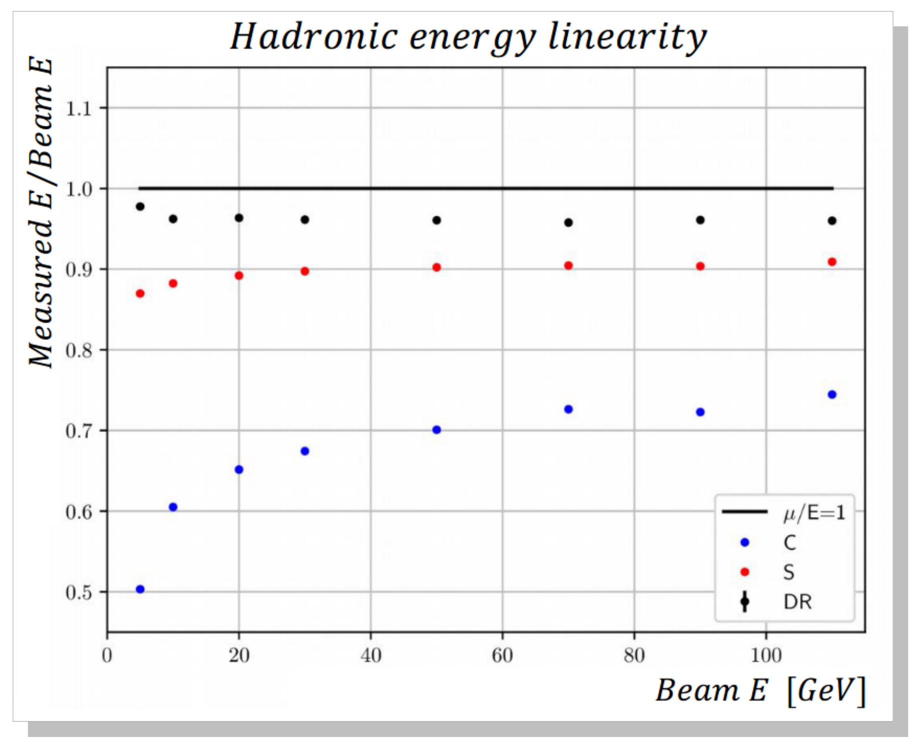
- Measured with 8 different energy of π+ beams

  → 5, 10, 20, 30, 50, 70, 90, 110 GeV
- Light attenuation & Dual-readout correction applied

- Stochastic term ~ 21% (dual-readout)
- Dual-readout correction improves linearity



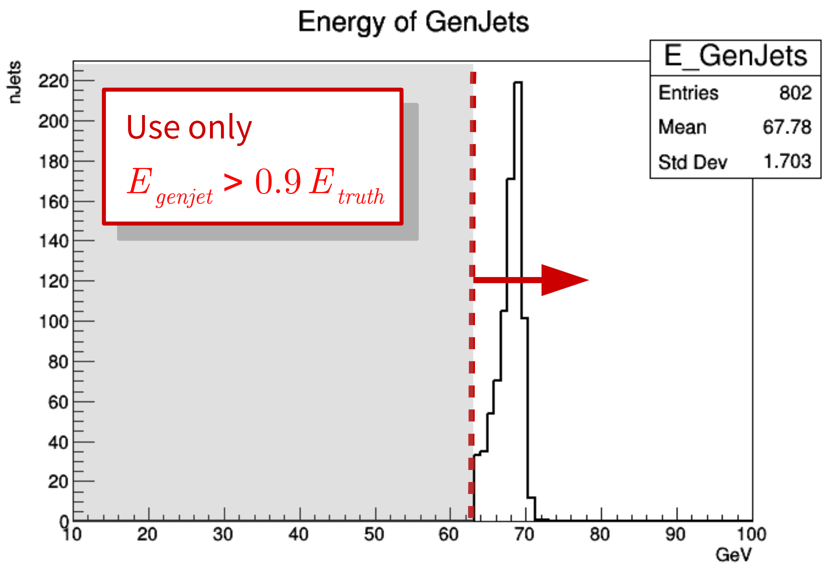




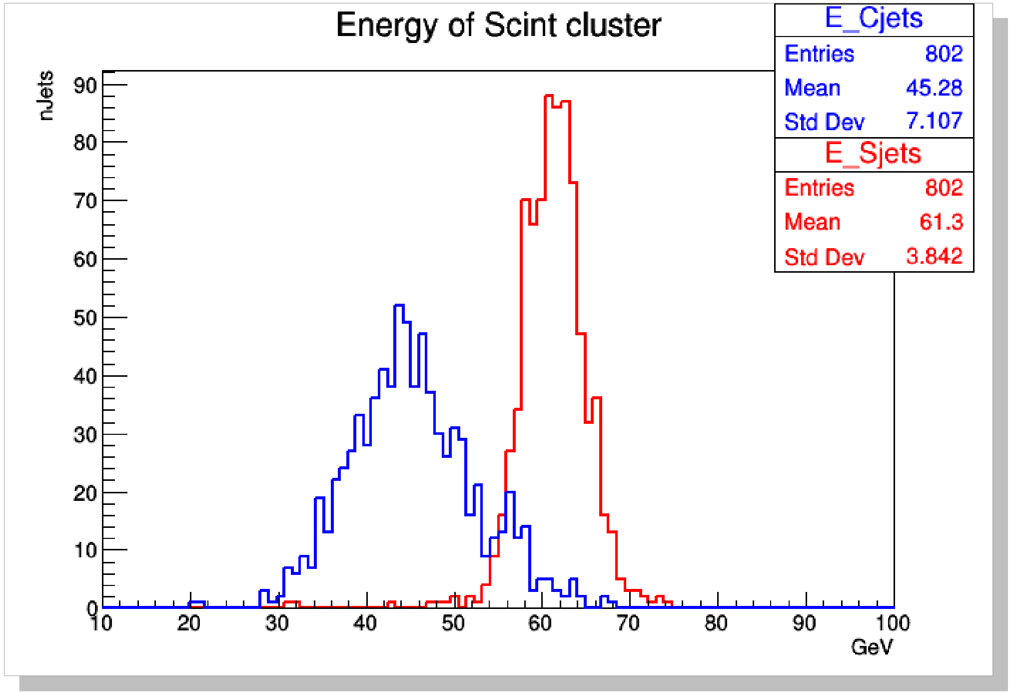
Jet energy resolution

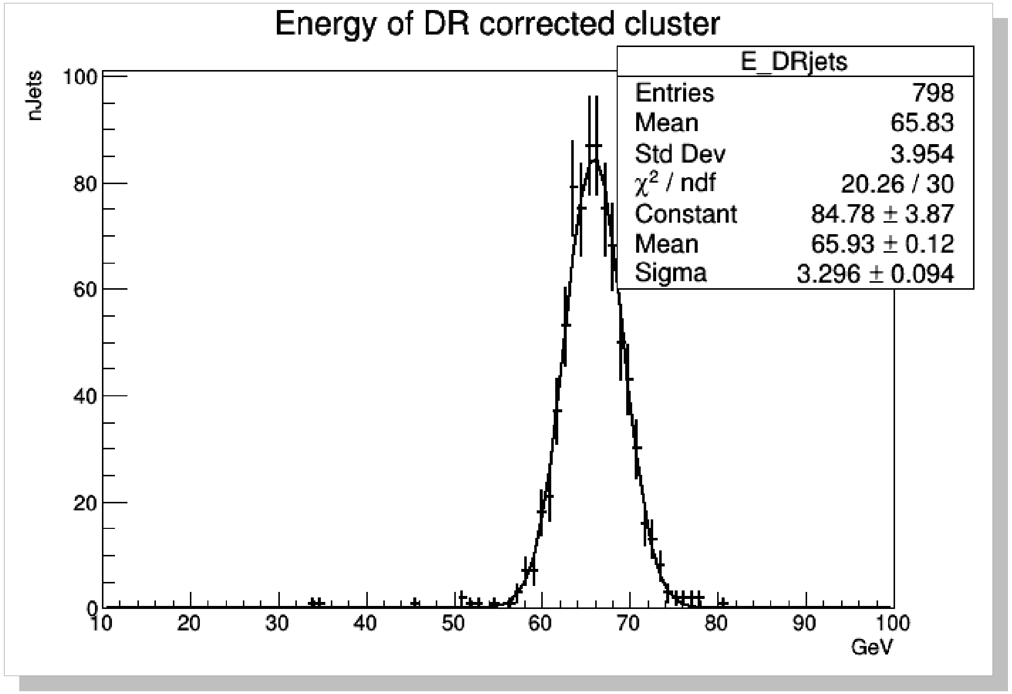
- Measured with 4 different energy of u-quark jets (ee → qq)

  → 50, 70, 90, 110 GeV
- Clustered jets with anti-kT algorithm (R = 0.8)
- To disentangle clustering performance from detector, used only events with GEN jet E > 90% of truth energy
- Dual-readout correction applied



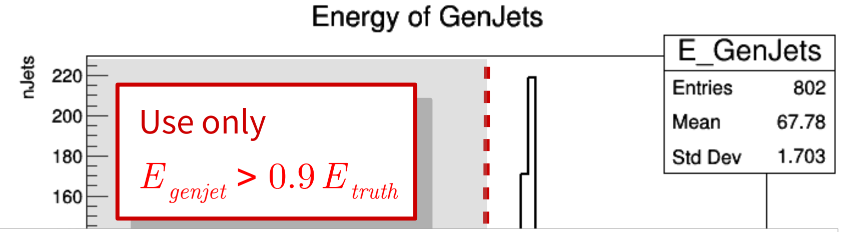
- Stochastic term ~ 26% (dual-readout)

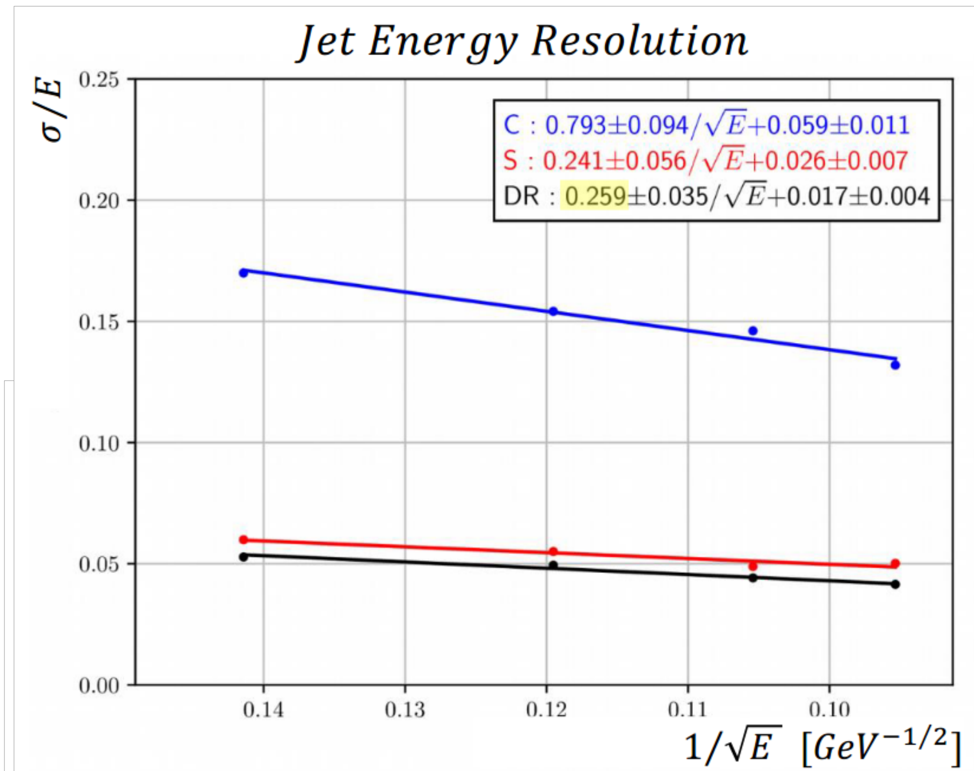




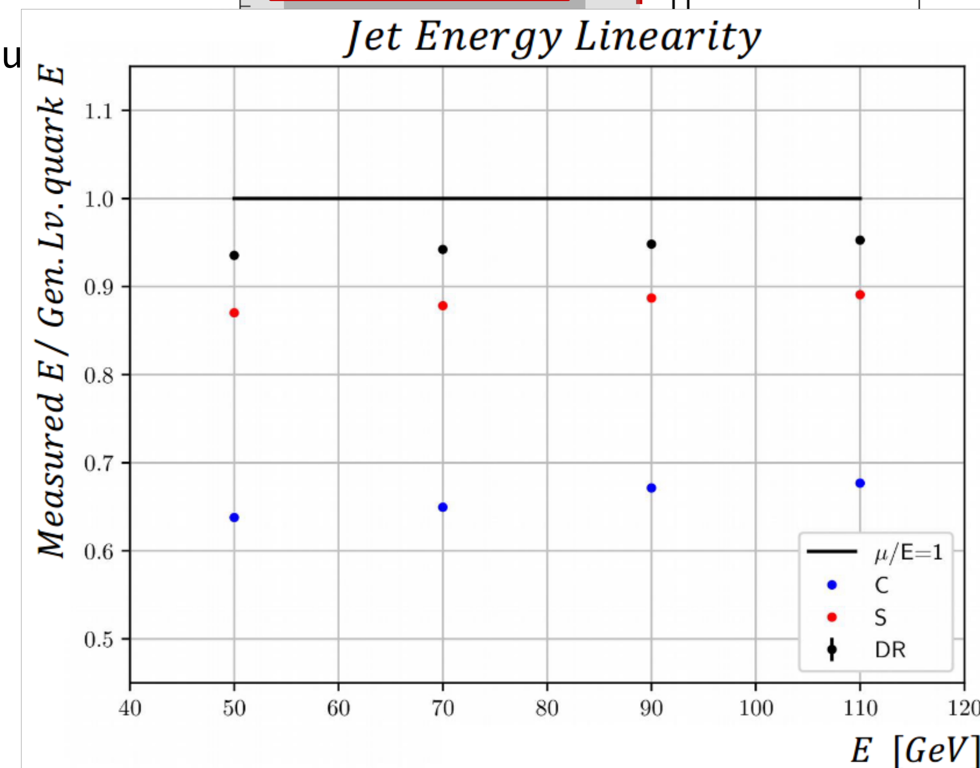
Jet energy resolution

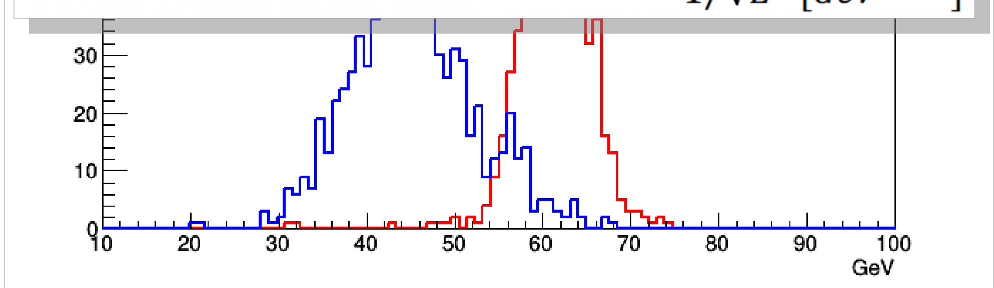
- Measured with 4 different energy of u-quark jets (ee → qq)
  → 50, 70, 90, 110 GeV
- Clustered jets with anti-kT algorithm (R = 0.8)

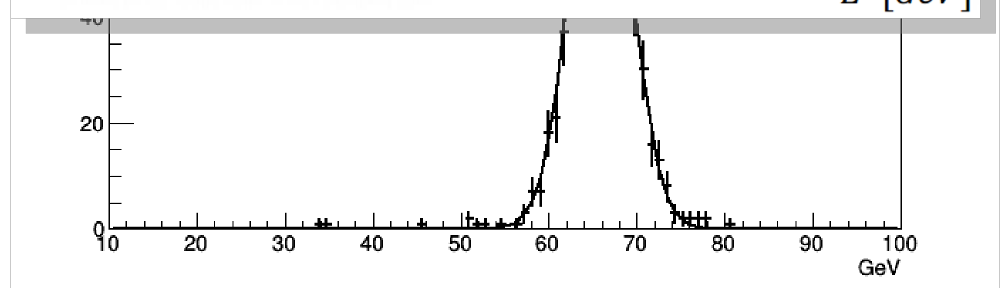




or, u





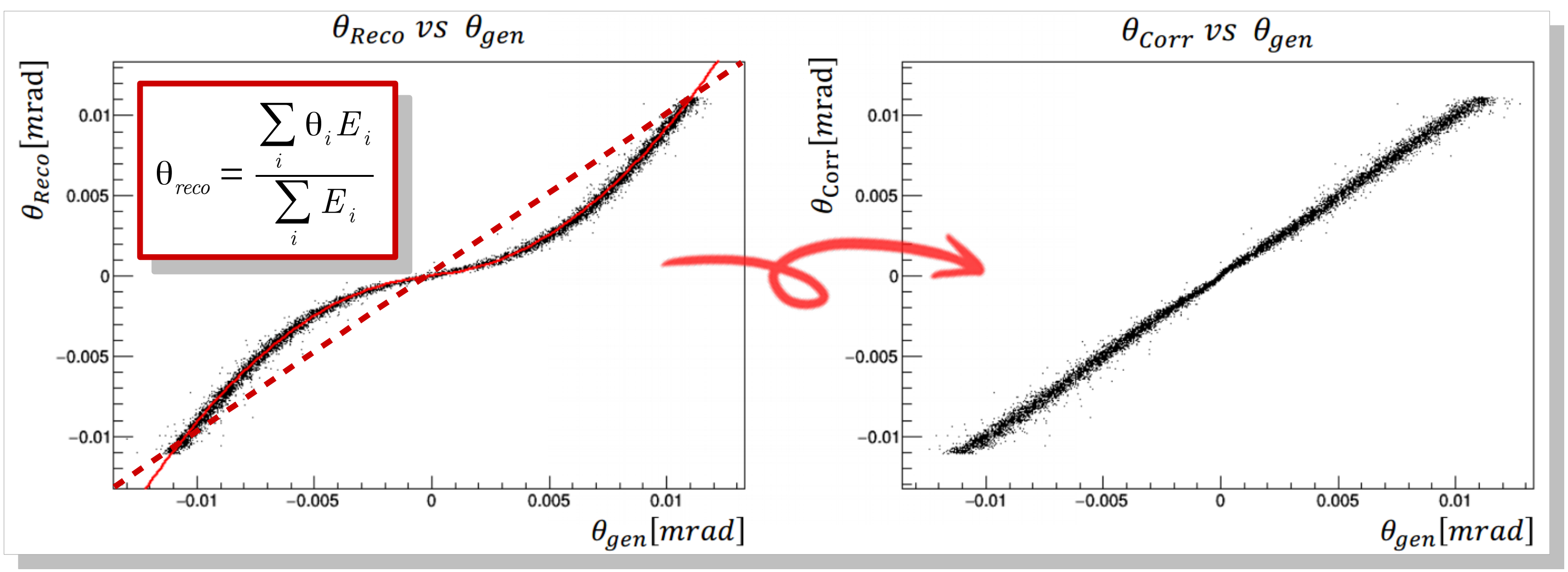


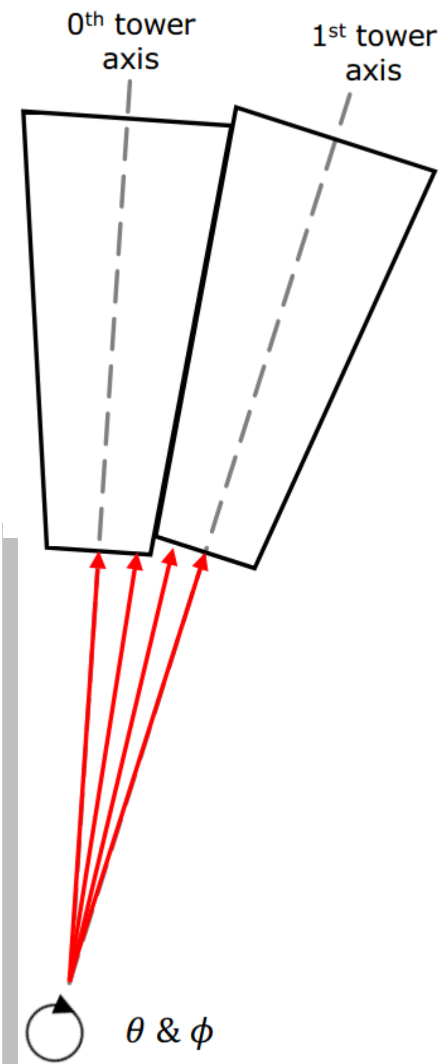
Angular resolution

- Measured with 6 different energy of e- beams

  → 10, 20, 40, 60, 80, 100 GeV
- Uniform-randomly generated events between 0th and 1st towers' center
- Center-of-gravity method used for reconstructing θ & ϕ at fiber-lv granularity
- Correct (map) reconstructed angle to that of incident (generated) particle

- Stochastic term ~ 2.1 mrad*√GeV (scintillation ch.)



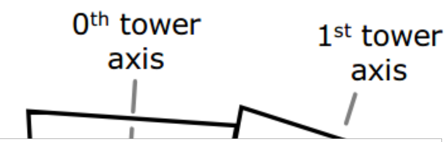


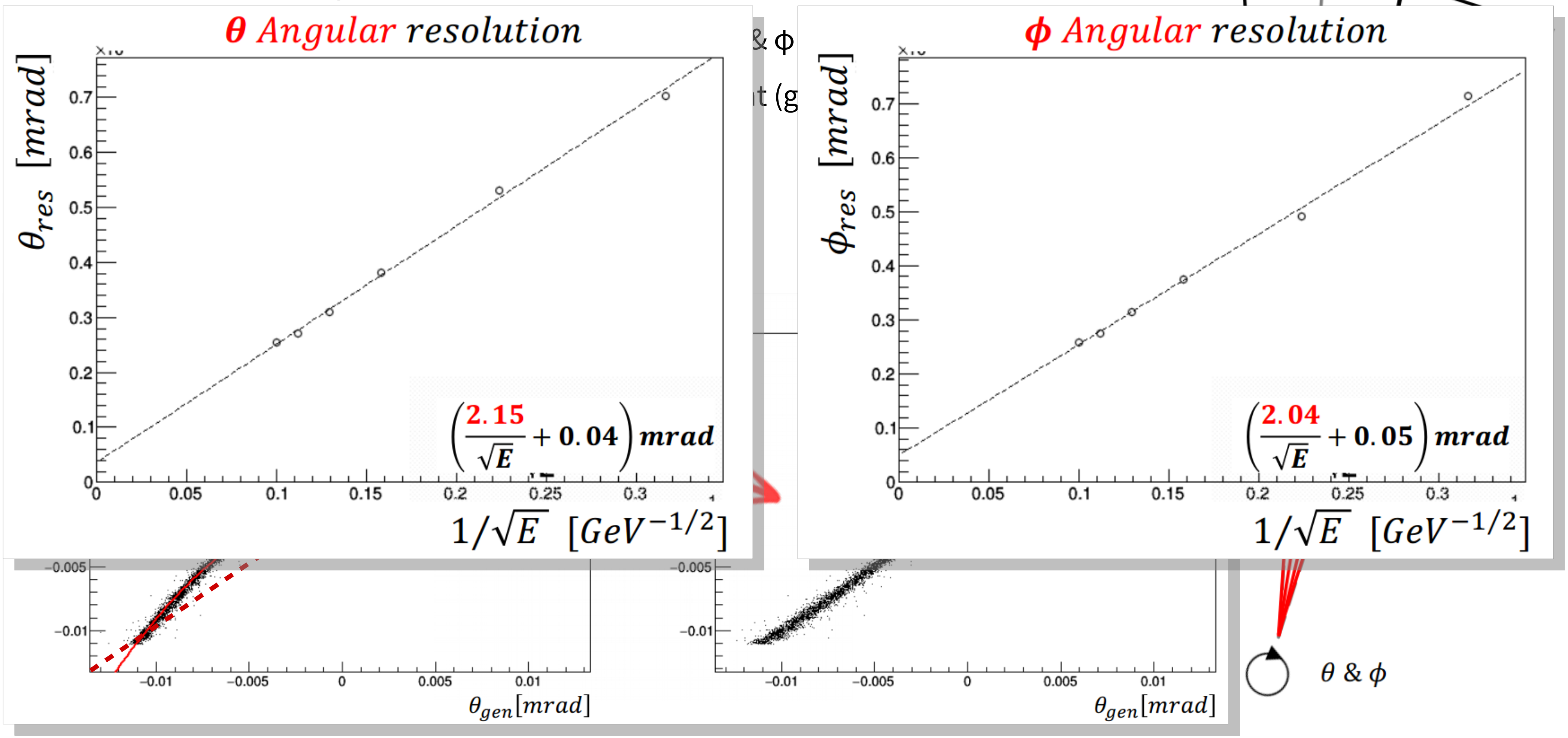
Angular resolution

- Measured with 6 different energy of e- beams

  → 10, 20, 40, 60, 80, 100 GeV

- Uniform-randomly generated events between 0th and 1st towers' center





Deep-Learning application

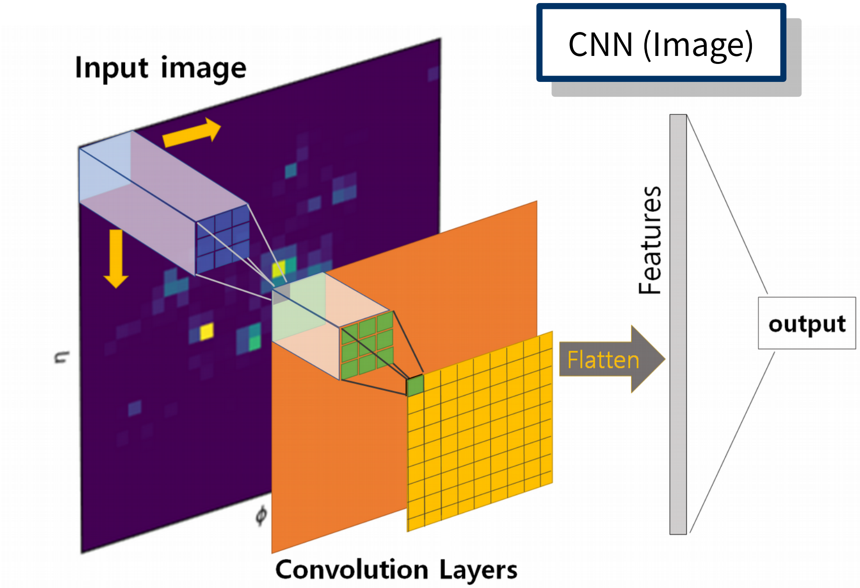
Goals of applications

Areas	Targets	On-going works
Particle ID	Shower ID, quark flavor tagging	Studying DL methods for q/g jet discrimination, e, γ, π shower ID
Jet Reco	Mass & Energy reco	Applying regression for q/g jet discriminant variables
Fastsim (GAN)	GAN based shower generator	Applying GAN for e- shower generation

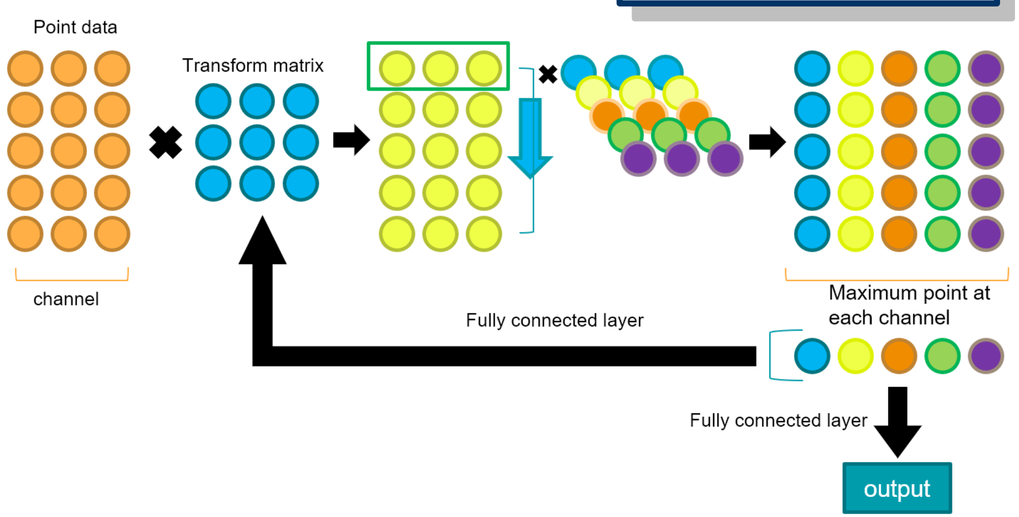
Input data

- Image: Reconstructed energy deposit images (η, ϕ) from S & Č ch.
- Point cloud: Reconstructed energy deposit positions (η, ϕ, depth) from S & Č ch.

CNN (Image)



PointNet (Point cloud)



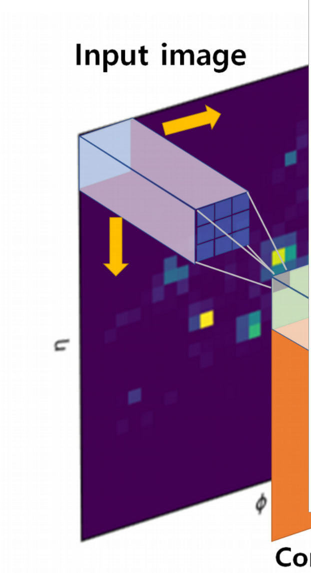
Deep-Learning application

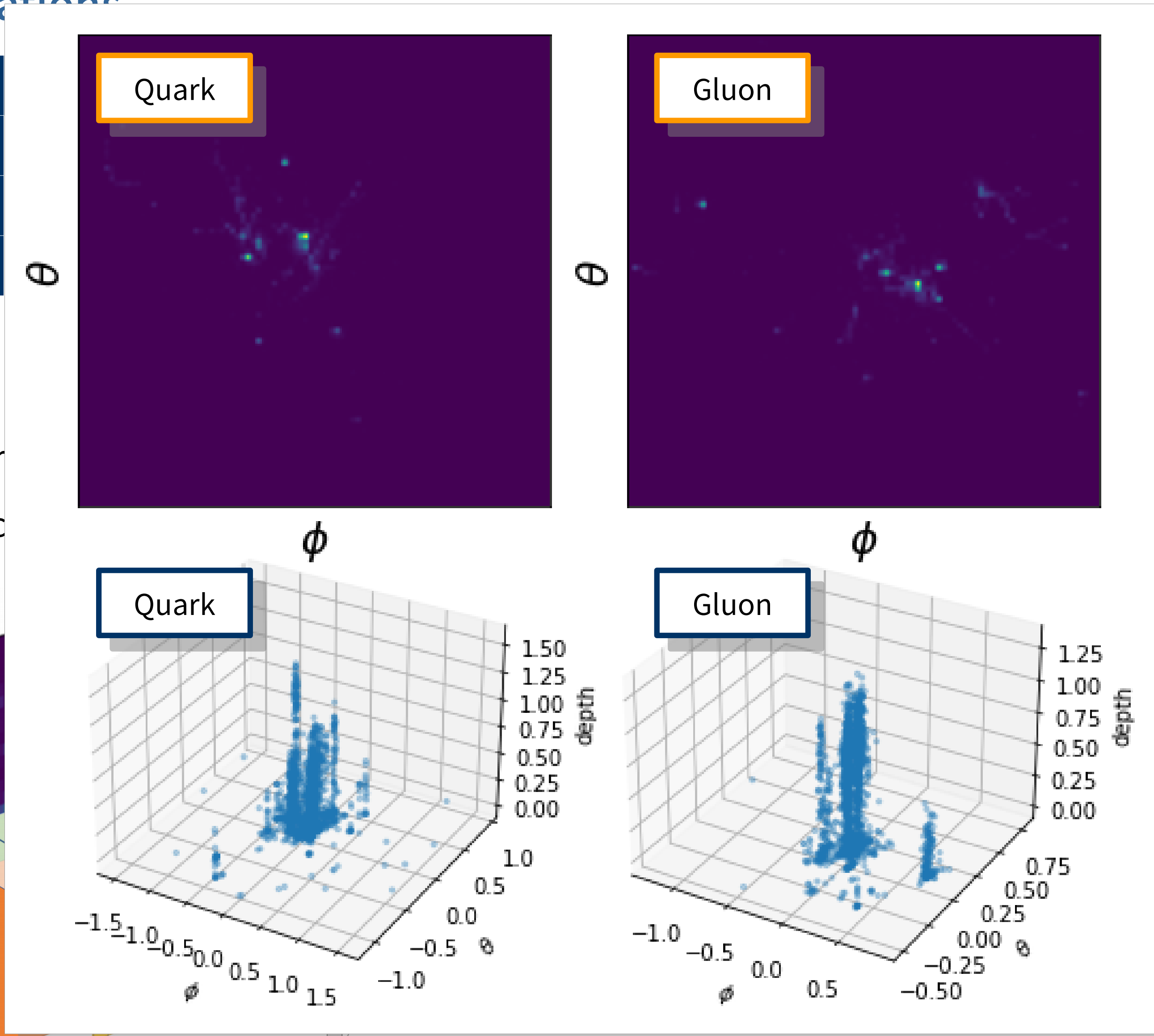
Goals of applications

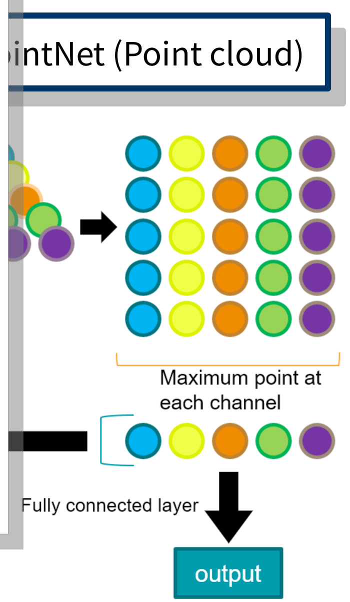
Areas	
Particle ID	, γ, π shower ID
Jet Reco	les
Fastsim (GAN)	

Input data

- Image: Reconstr
- Point cloud: Rec







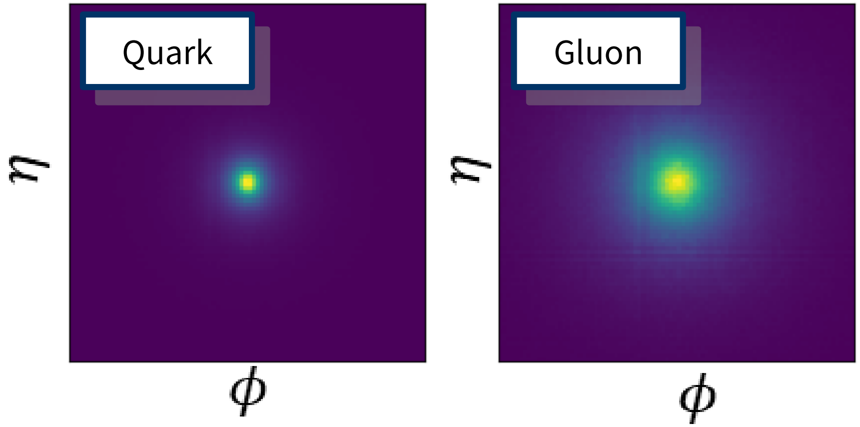
Classification of quark and gluon jets

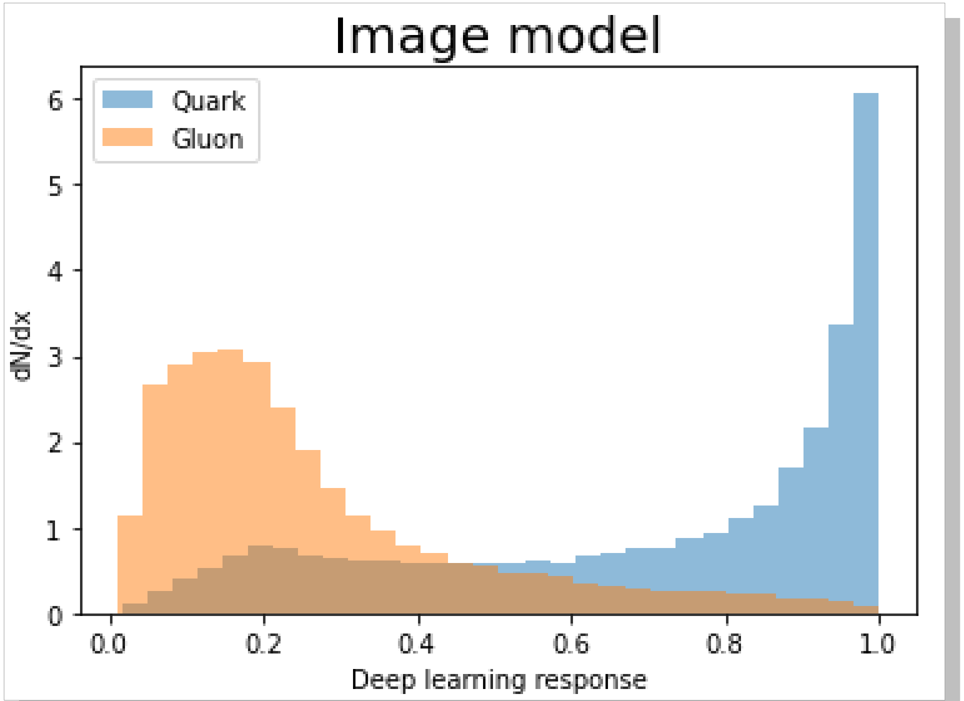
- Performed with 50 GeV u q & g jets
- Tested with both CNN and PointNet model

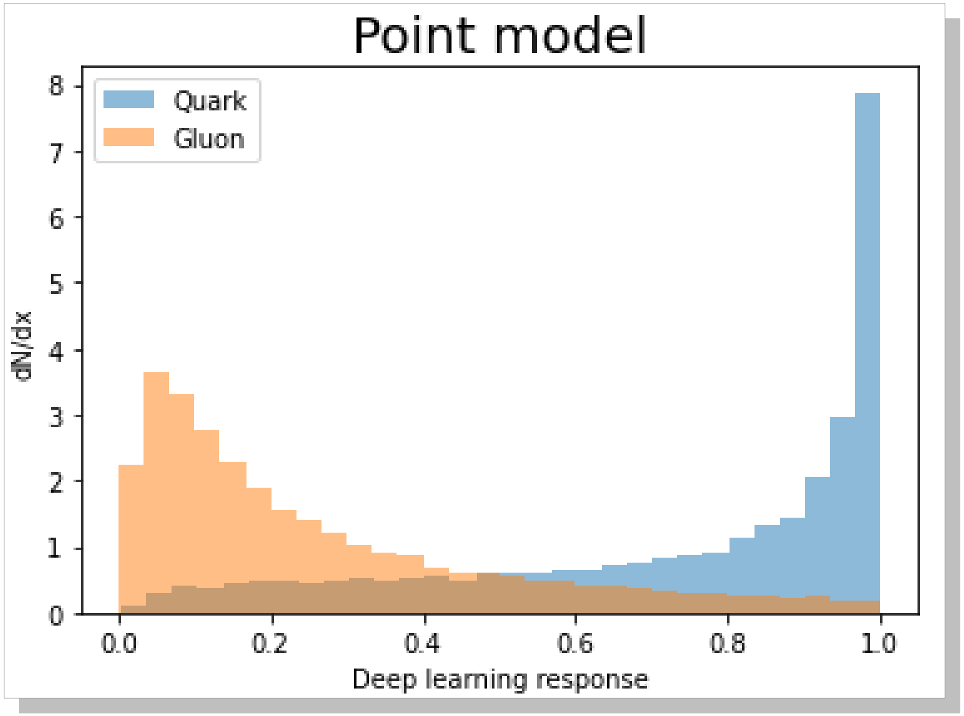
  → larger multiplicity & dispersion for g compared to q

- Jet flavor tagging in pipeline

Averged scintillation energy images





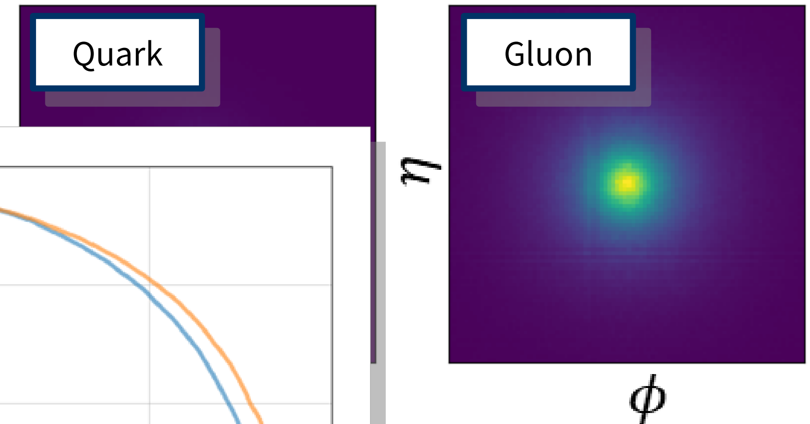


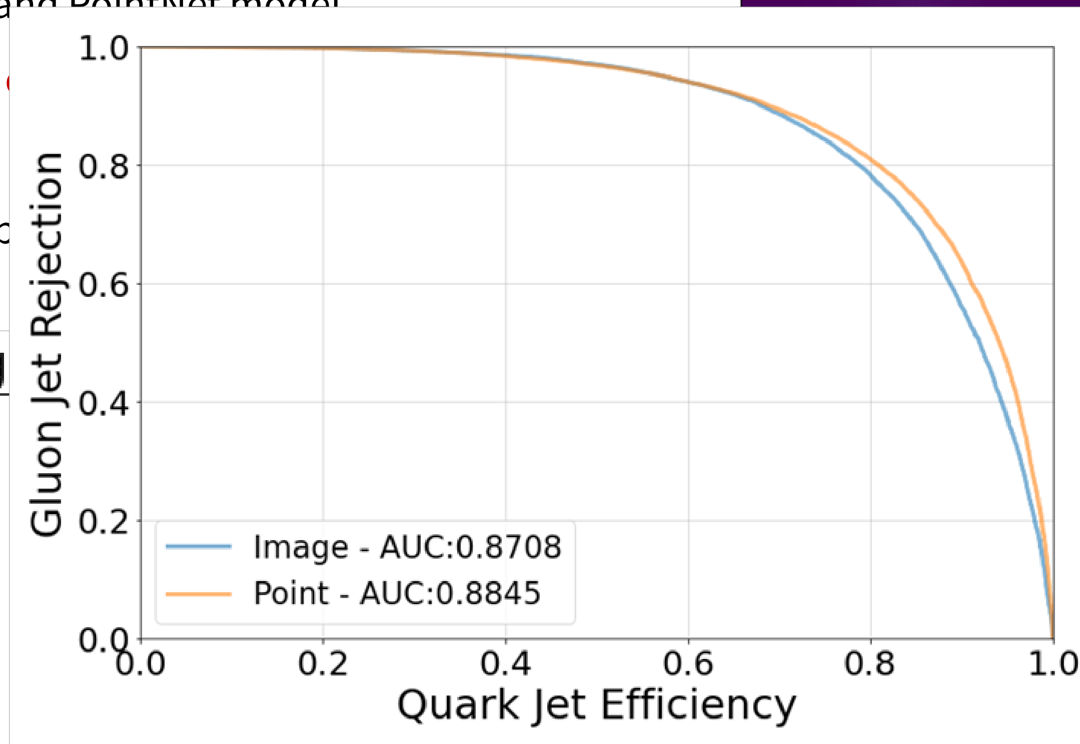
Jet identification

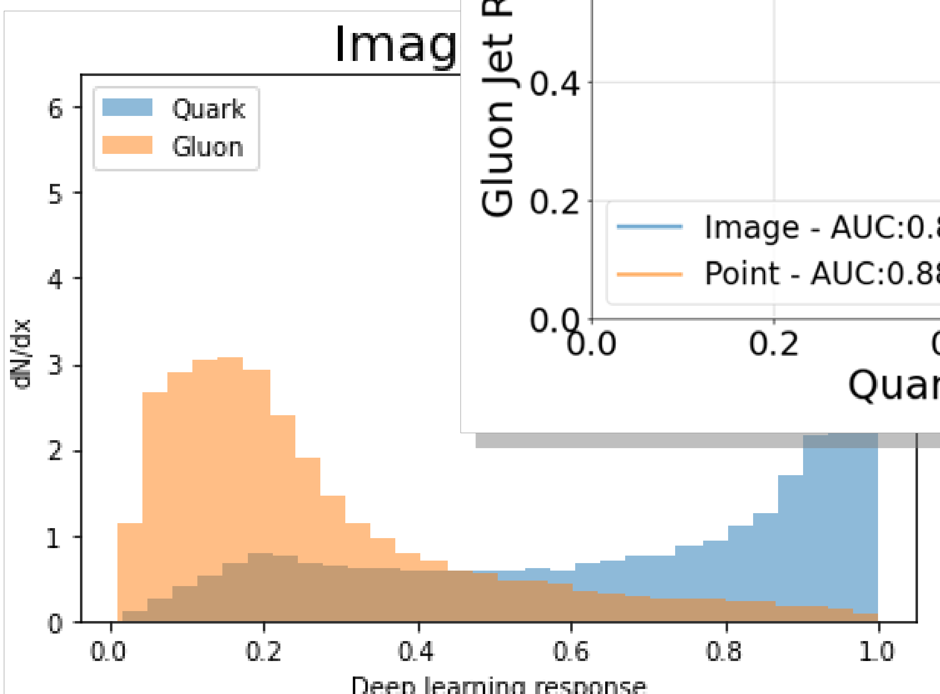
Classification of quark and gluon jets

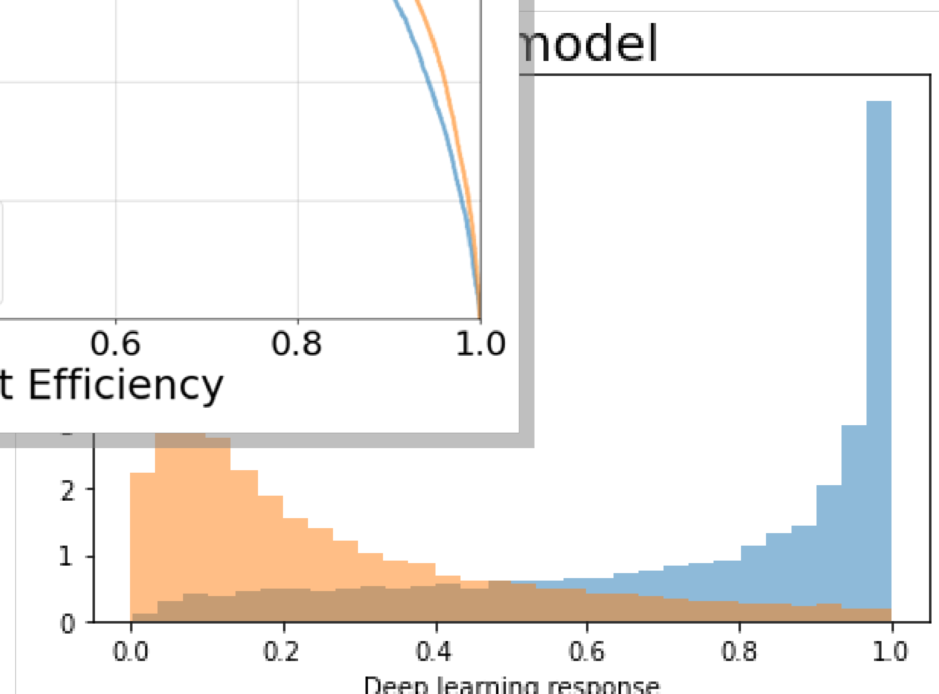
- Performed with 50 GeV u q & g jets
- Tested with both CNN and PointNet model
  → larger multiplicity & (
- Jet flavor tagging in pip

Averged scintillation energy images



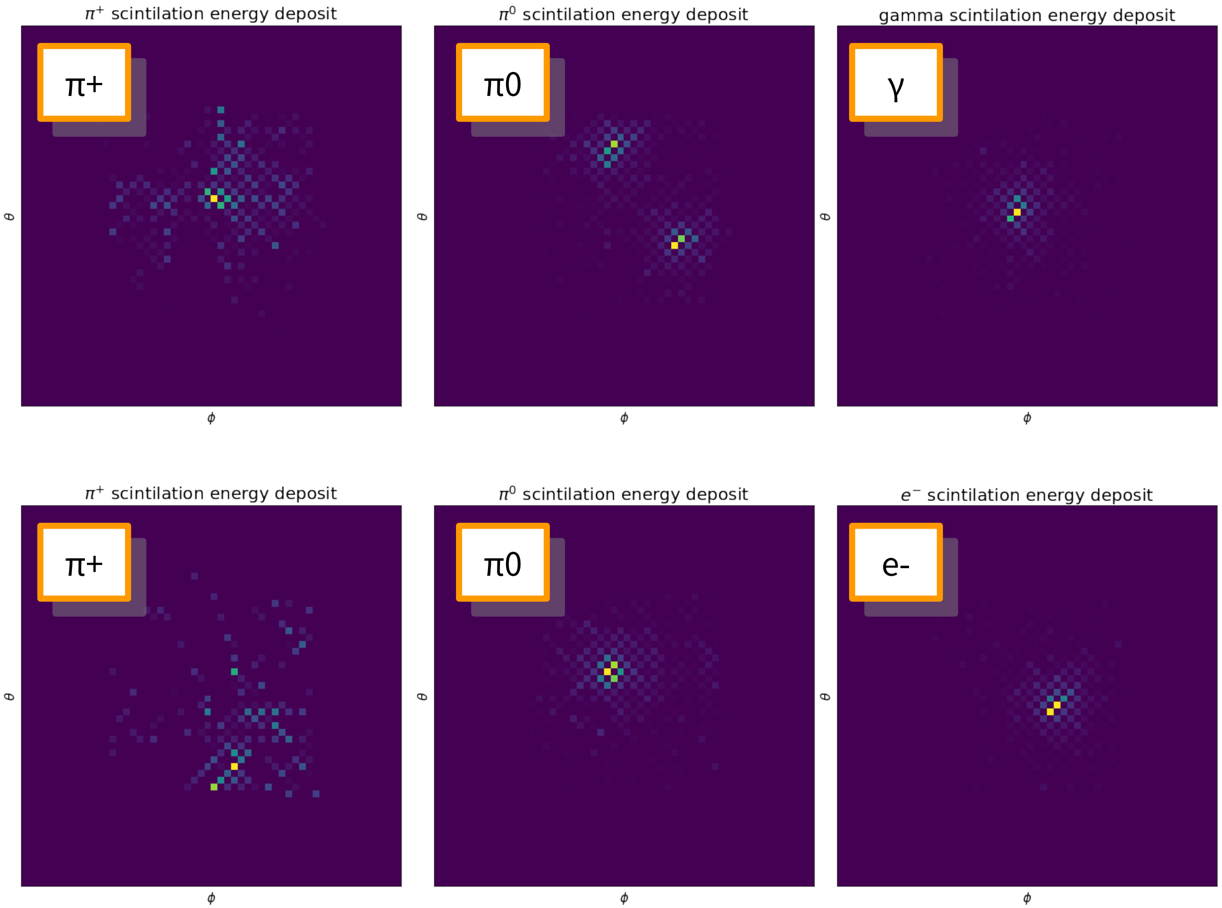


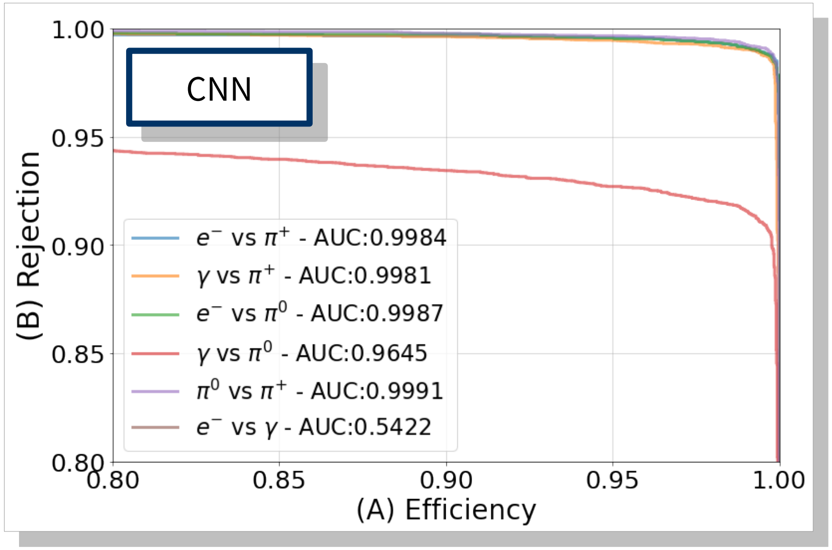


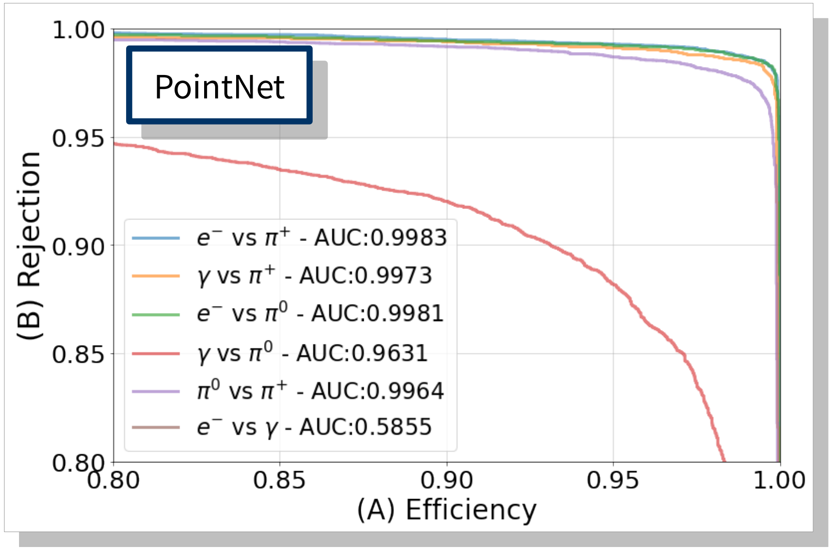


Particle identification

- Discrimination of various particles – e-, π+, γ, π0 (20 GeV)
- Excellent e- vs π+ performance
- Great γ vs π0 separation
- Extending energy scope and types of particles (p, n) in pipeline







Summary

Simulation of dual-readout calorimeter

- Performed Geant4 simulation with detailed optical properties
- Developed Fastsim module for optical photon propagation

Performances of dual-readout calorimeter

- Measured EM/Hadronic/Jet energy resolution & angular resolution

	EM	Had	Jet	Angular
Stochastic	12 %	21 %	26 %	2.1 mrad
Constant	0.2 %	1.2 %	1.7 %	0.05 mrad

Deep-Learning applications

- Studies on various DL application are on-going
  - Quark-gluon jet discrimination (AUC ~ 0.88)
  - Particle identification

AUC (CNN)

	e-	γ	π0	π+
e-		0.542	0.9987	0.9984
γ			0.9645	0.9981
π0				0.9991
π+				

Future plans

- Probe performances with (potential) ECCE configuration
  - 1.25 m tower length
  - Cu/W/Fe absorbers …

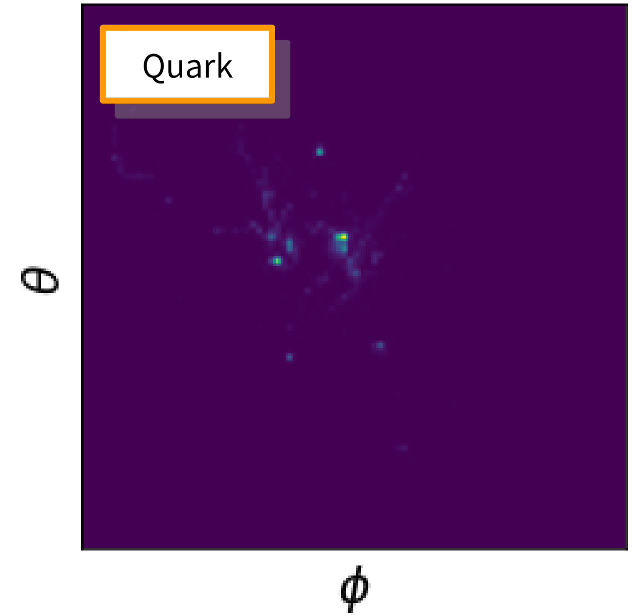
Backups

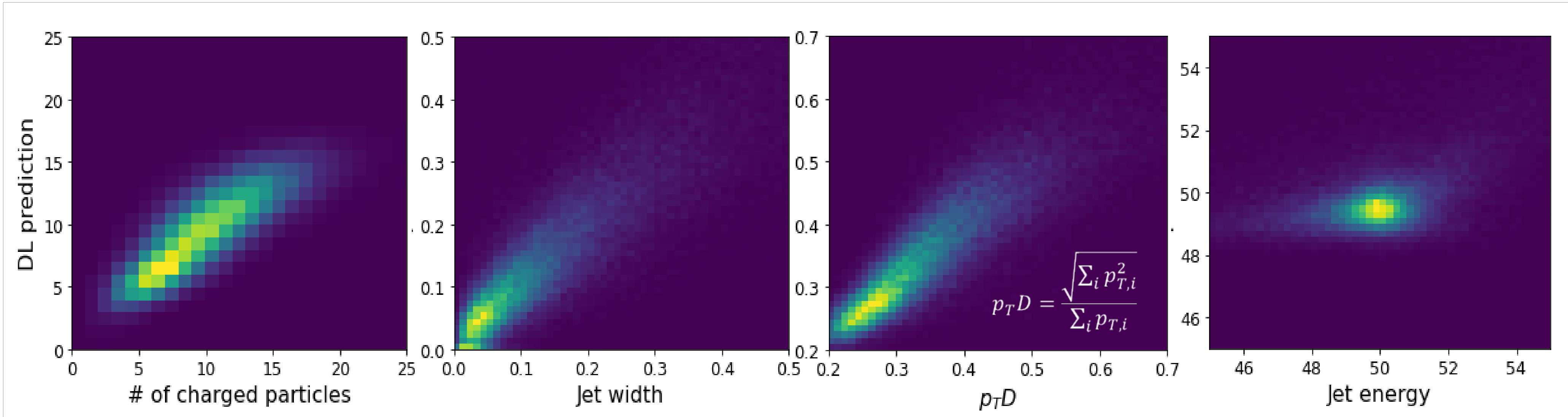
Jet variable regression

Predicting jet reconstruction variables

- Trained with 50 GeV u-quarks and gluon jets
- Performed regression with
  - Particle multiplicity
  - Jet width
  - PtD (jet fragment variable)
  - Jet energy

→ Predictions follow trend of generated variables





Performance

Current computing performance of dual-readout fullsim

By KY Hwang

	10GeV	20GeV	30GeV	50GeV	70GeV
# of threads			2		
# of event in 1 file			2		
Average memory			~5GB		
Average time taken per 1 event	~ 10min	~ 12 min	~ 16min	~ 22 min	~ 28 min
Average time taken per 1000 evet (w/ 100 cpus)	~ 1h 40min	~ 2h	~ 2h 40min	~ 3h 40 min	~ 4h 40min

- Computing performance is tested after optimization with electron gun of energy from 10 GeV to 70 GeV.
  - At local institution server with Intel Xeon CPU E5-2680 v2 2.8GHz
  - Fast simulation for optical photon is applied
  - Time taken for geometry construction ~ 7 min
  - Time taken for 10 GeV simulation w/o geometry construction ~ 3 min
- Although calorimeter is the most heavy consumer of fullsim routine, there is still more room to be improved.
  - Exploring multi-threading
  - Improving fast simulation for optical photon tracking

Introduction to Key4hep

Infrastructure of FCCSW

- The FCC software stack has been assembled using as much as possible existing components
- Adopting a common software infrastructure for future experiments – Key4hep, EDM4hep

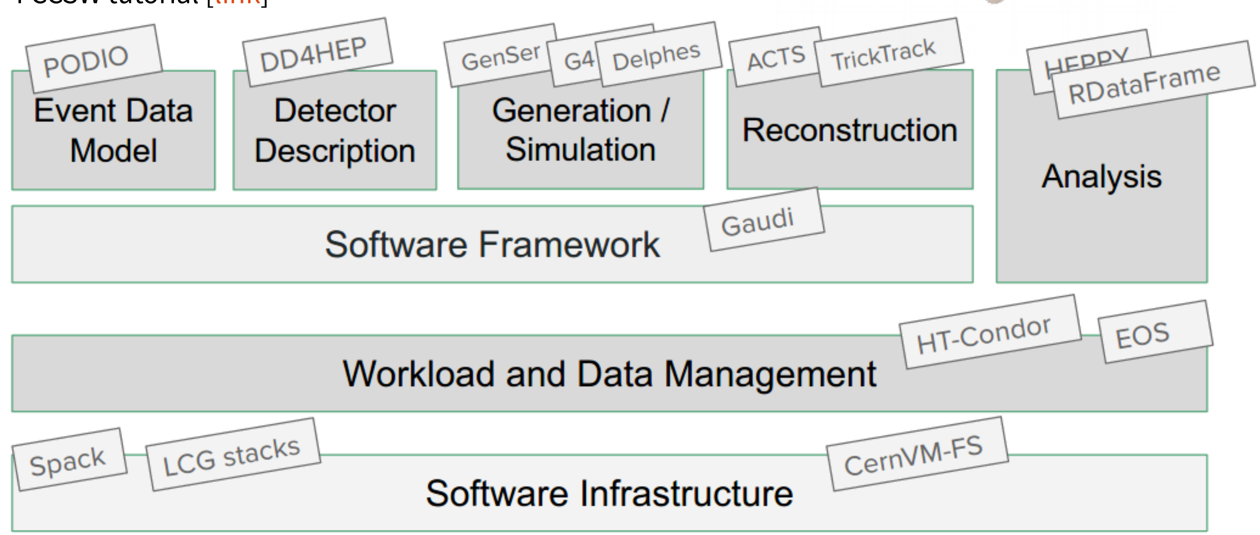
Brief history of Key4hep

- Common software framework for all future HEP experiments proposed at the Future Collider Software Workshop at Bologna June 2019 [link], followed by mini-workshop at Hong Kong Jan 2020 [link]
- Consensus among future collider communities including ILC, CLIC, CEPC and FCC
- Encompass typical needs of HEP experiments, provide common turnkey stack covering different domains.

- Identified as CERN EP strategic R&D program for future experiments

  → Essential to follow up recent strategy as an application developer

FCCSW tutorial [link]



Migration to Key4hep elements

Migration of detector description (DD4hep) & event data model (EDM4hep)

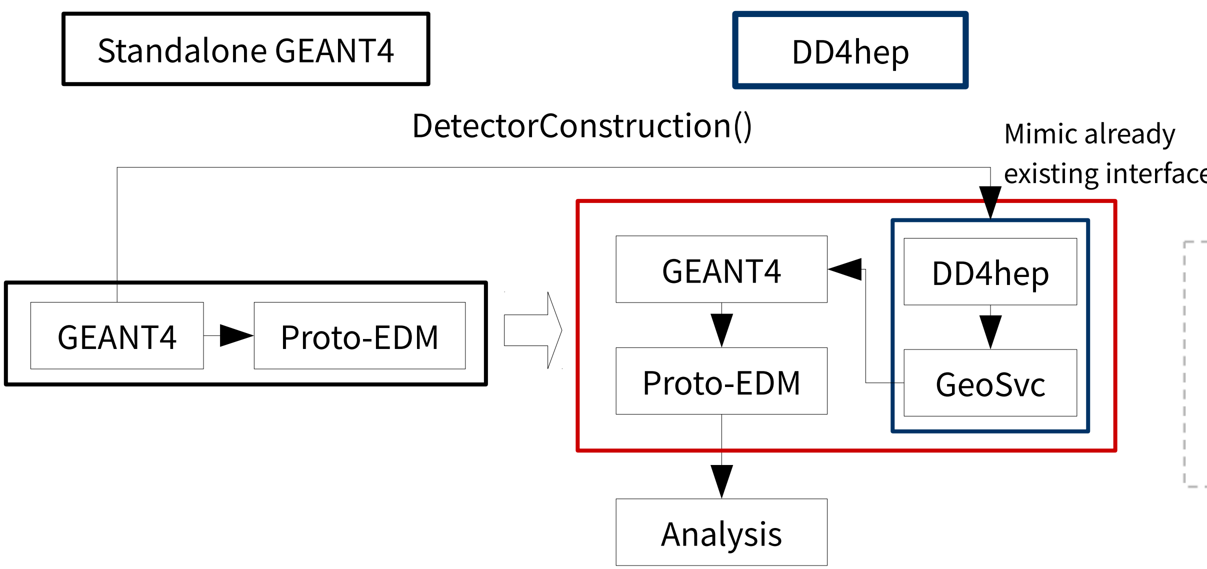
- All future collider communities (ILC, CLIC, FCC, CEPC) + current experiments (CMS, LHCb) have migrated or are migrating to DD4hep detector description tool
- A good opportunity to take advantage of efforts from wider communities

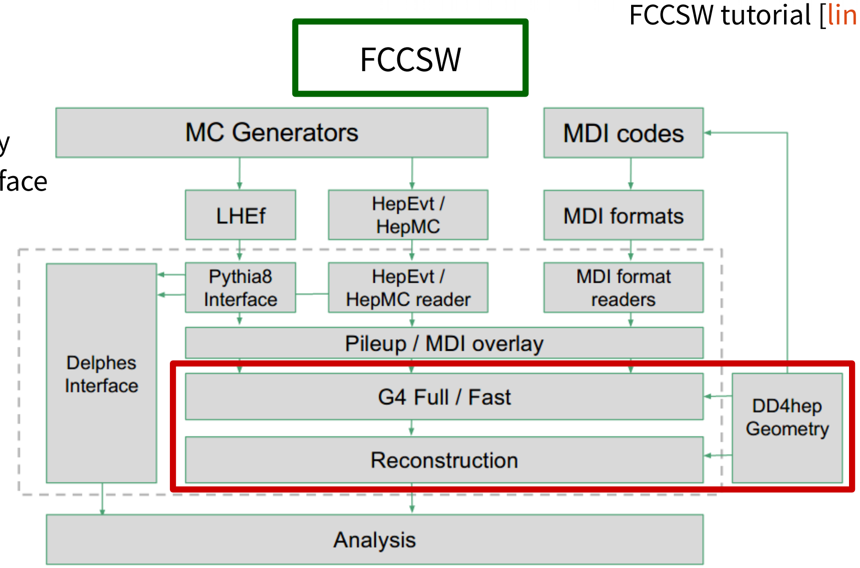
  ex) combined detector performances, e.g. tracker+calorimeter, MIP timing detector, etc
- Also, future collider communities are migrating to EDM4hep to share a common EDM

FCCSW tutorial [link]

→ Start by migrating to necessary subset of Key4HEP elements for dual-readout calorimeter

- DD4hep – completed [FCCSW meeting] [DD4hep developers meeting]
- EDM4hep – on-going



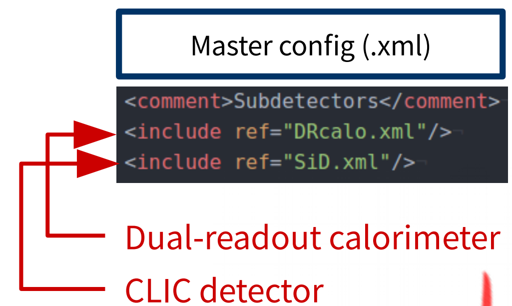


Advantages of DD4hep

Interacting with other sub-detector components using DD4hep

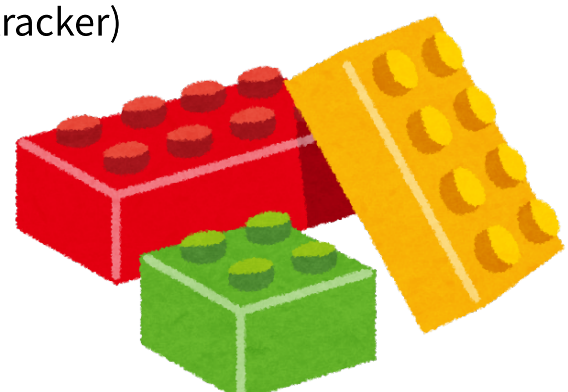
- DD4hep [github] – SW framework providing a complete solution for full detector description (DD)
- DD of dual-readout calorimeter can be easily extended by configuring the compact file (.xml)

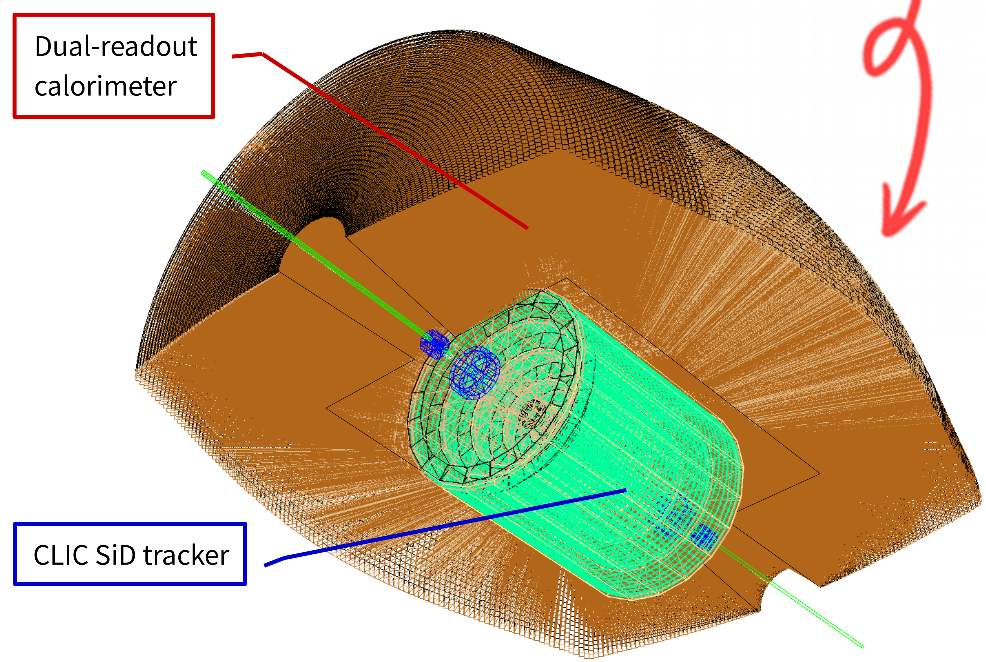
  e.g. dual-readout calorimeter + tracker
- Interfaced to GEANT4 (i.e. simulation is done by GEANT4)
- Making available to study performance of multiple sub-detector with relatively easier access



Study with other detectors

- Migrated to DD4hep implementation, planning to study performance with other detectors
- Migration of other IDEA detector components are on-going (e.g. tracker)





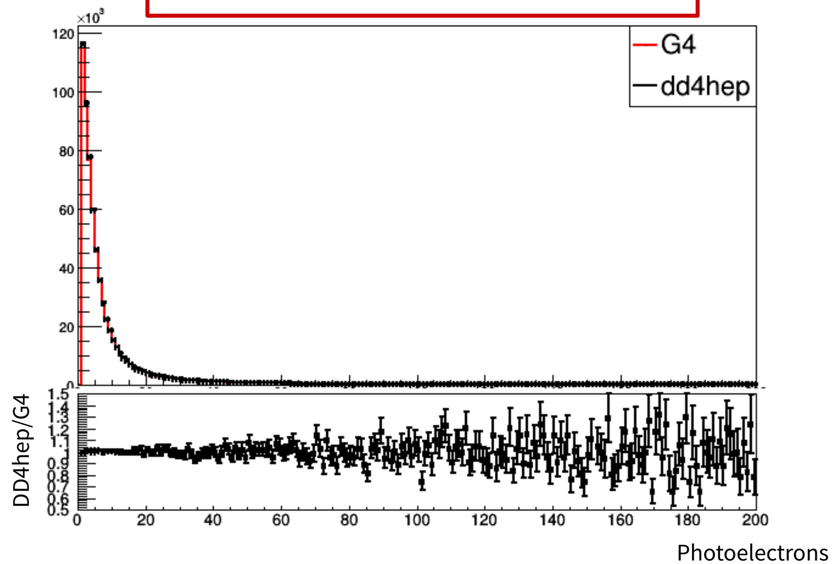
DD4hep vs GEANT4

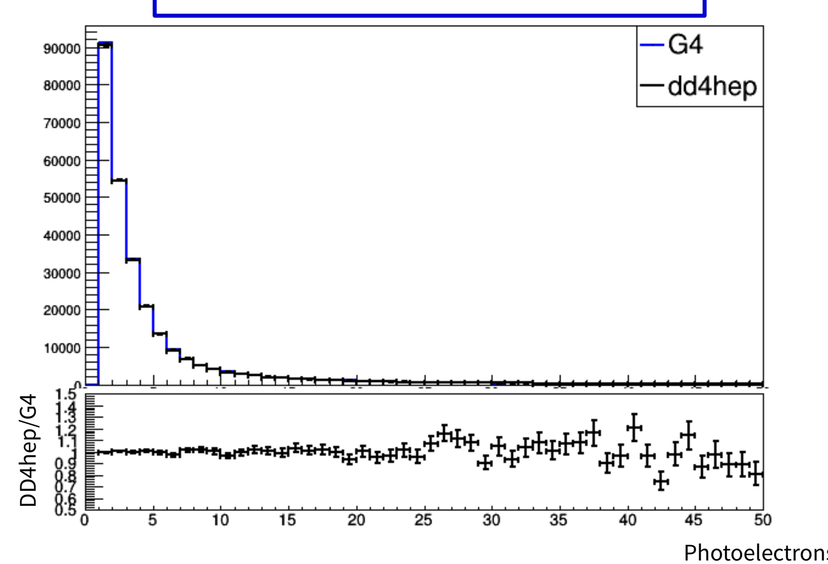
Comparison between DD4hep & standalone GEANT4

DD4hep v01-12-01
GEANT4.10.06.p02

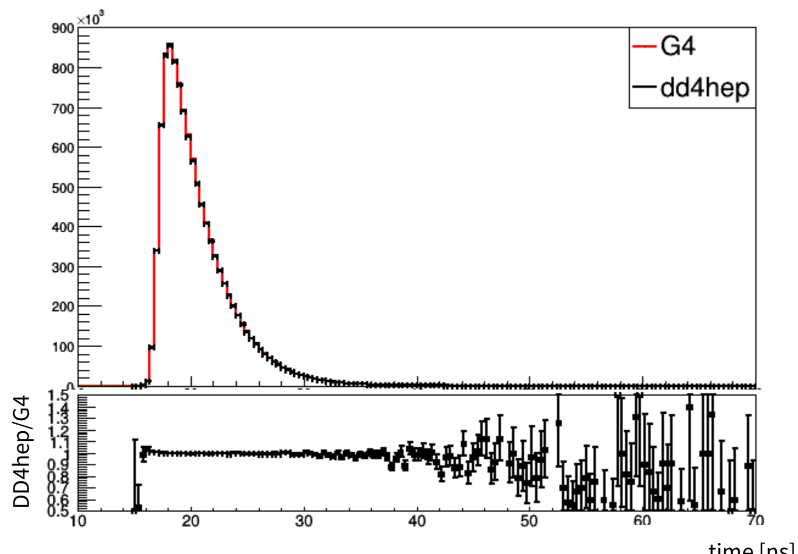
	Scintillation	Čerenkov

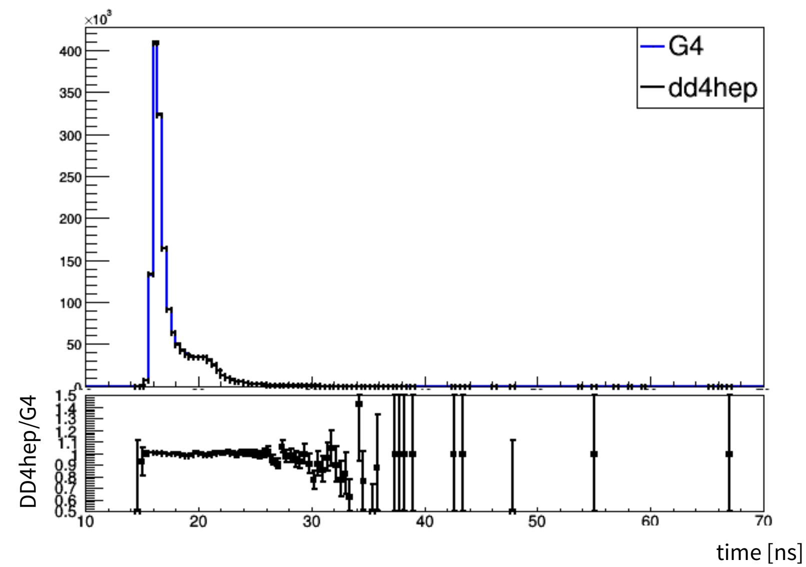
\# of photoelectrons





Photon arrival time



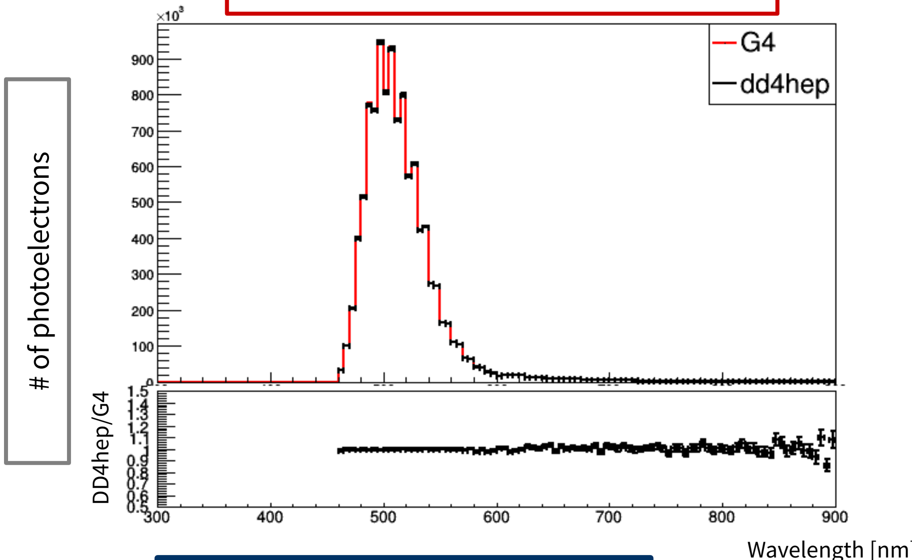


DD4hep vs GEANT4

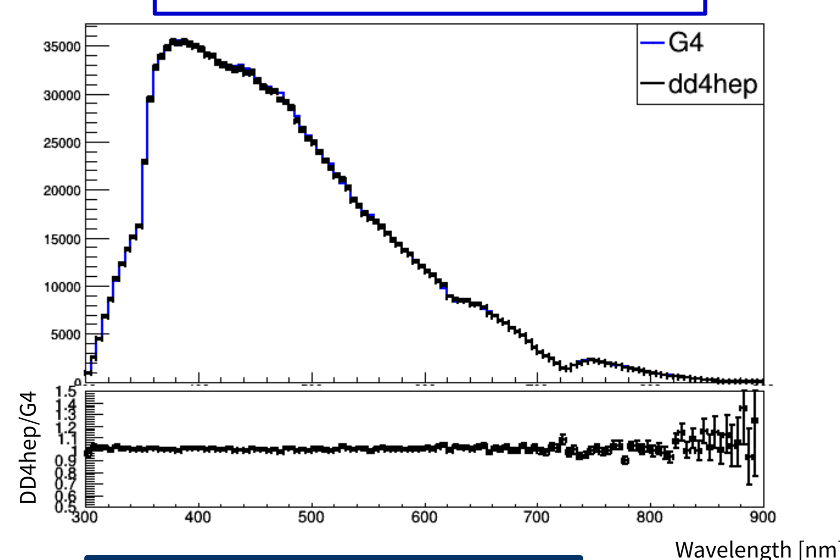
Comparison between DD4hep & standalone GEANT4

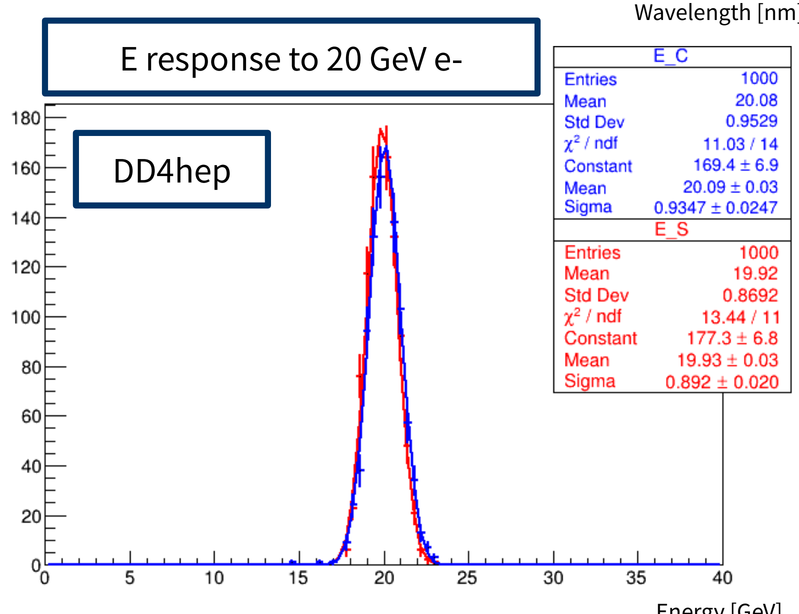
DD4hep v01-12-01
GEANT4.10.06.p02

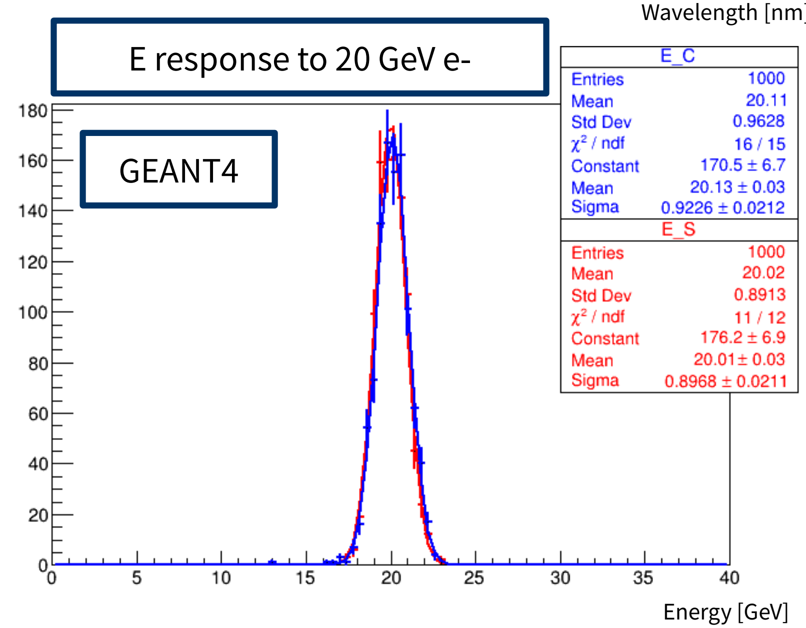
Scintillation



Čerenkov



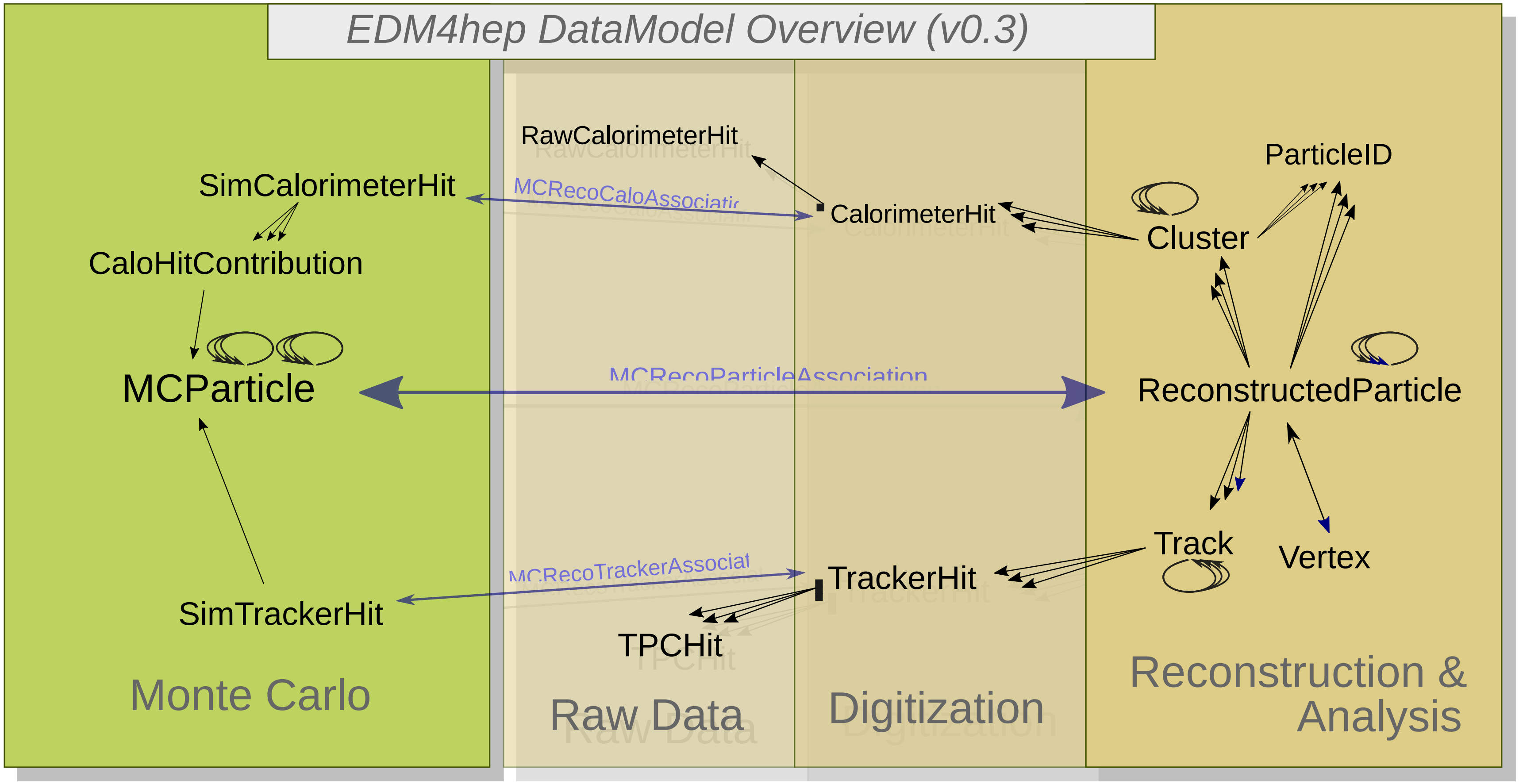




EDM4hep & Podio

Sharing common event data model

- EDM4hep [github] is the common EDM which can be shared by multiple future collider communities
- Support various use-cases motivated from different experiments
- Data model based on the Podio [github], a code-generator which supports the creation and handling data models



Podio – Generator for EDM4hep

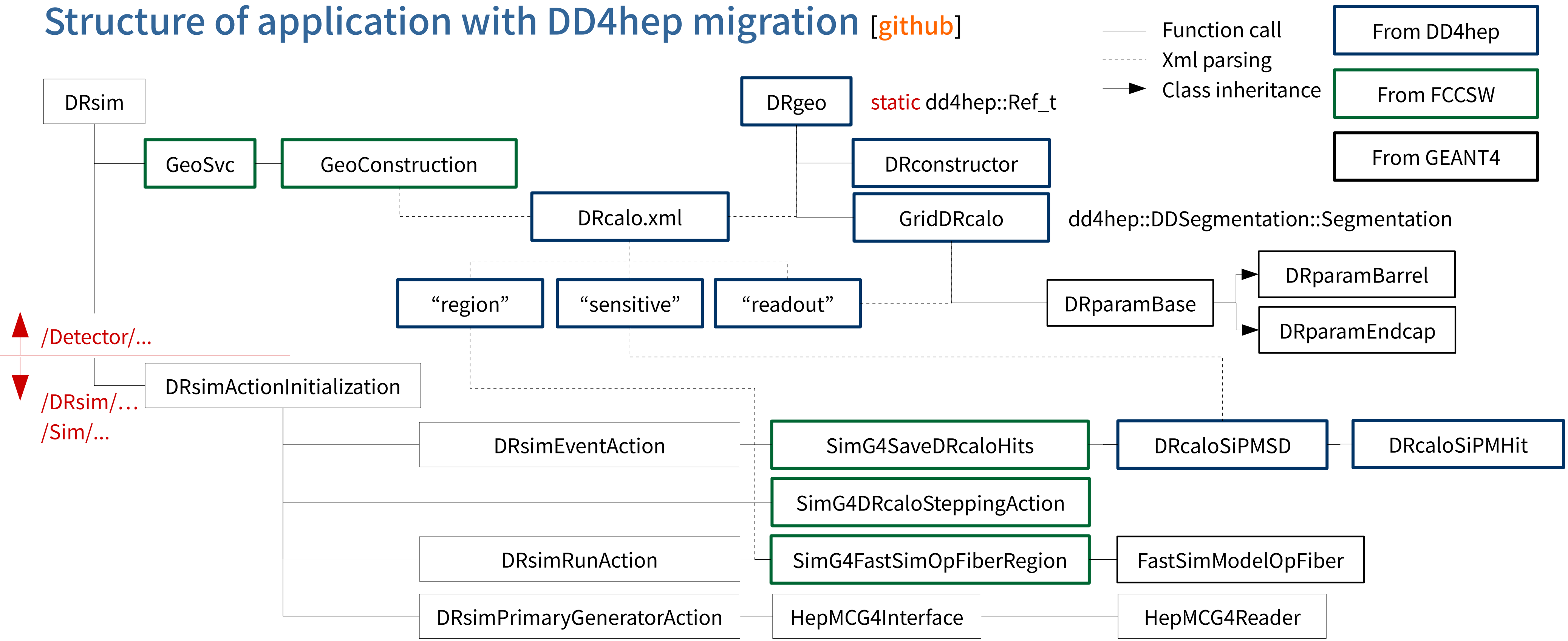
- Generate thread-safe code starting from a high-level description written in yaml format
- Produces c++ code at CMake step via Python
- Also can be extended or adapted by the needs of the dual-readout calorimeter

```yaml
#------------- SimCalorimeterHit
edm4hep::SimCalorimeterHit:
  Description: "Simulated calorimeter hit"
  Author : "F.Gaede, DESY"
  Members:
    - unsigned long long cellID       //ID of the sensor that created this hit
    - float energy                    //energy of the hit in [GeV].
    - edm4hep::Vector3f position      //position of the hit in world coordinates.
  OneToManyRelations:
    - edm4hep::CaloHitContribution contributions  //Monte Carlo step contribution
```

```cpp
class SimCalorimeterHit {

  friend SimCalorimeterHitCollection;
  friend SimCalorimeterHitCollectionIterator;
  friend ConstSimCalorimeterHit;

public:

  /// default constructor
  SimCalorimeterHit();
  SimCalorimeterHit(unsigned long long cellID, float energy, edm4hep:

  /// constructor from existing SimCalorimeterHitObj
  SimCalorimeterHit(SimCalorimeterHitObj* obj);

  /// copy constructor
  SimCalorimeterHit(const SimCalorimeterHit& other);

  /// copy-assignment operator
  SimCalorimeterHit& operator=(const SimCalorimeterHit& other);
public:

  /// Access the ID of the sensor that created this hit
  const unsigned long long& getCellID() const;

  /// Access the energy of the hit in [GeV].
  const float& getEnergy() const;

  /// Access the position of the hit in world coordinates.
  const edm4hep::Vector3f& getPosition() const;



  /// Set the ID of the sensor that created this hit
  void setCellID(unsigned long long value);

  /// Set the energy of the hit in [GeV].
  void setEnergy(float value);

  /// Set the position of the hit in world coordinates.
  void setPosition(edm4hep::Vector3f value);
  /// Get reference to position of the hit in world coordinates.
  edm4hep::Vector3f& position();
```

Framework structure

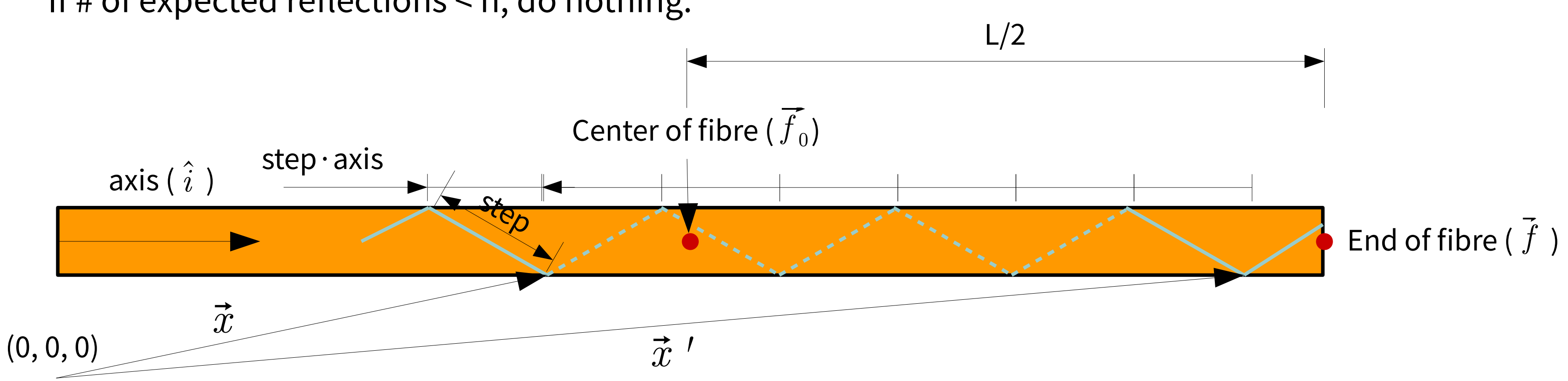
Structure of application with DD4hep migration [github]



- Detector geometry parameters & materials are controlled by configuring xml file.
- Interface between GEANT4-DD4HEP, Fullsim/Fastsim actions followed FCCSW-way of implementation.
  - Designed to require little efforts for integration to central SW.

Estimating the target point of translation for fast optical photon transportation

- Based on the postulate that the step length of individual track remains same throughout whole transportation, the point of translation can be estimated easily.
- $\vec{f} = \vec{f}_0 + L/2\,\hat{i}$
- $\vec{f}_0$ & $\hat{i}$ can be obtained by G4TouchableHandle (touchable→GetHistory()→GetTopTransform().Inverse().TransformPoint/Axis(x, y, z))
- # of expected reflections = std::floor($\frac{(\vec{f} - \vec{x})\cdot\hat{i}}{\overrightarrow{step}\cdot\hat{i}}$)
- $\vec{x}\,' = \vec{x} + (\overrightarrow{step}\cdot\hat{i})\hat{i} \times$ # of expected reflections
- $t\,' = t$ + step/velocity $\times$ # of expected reflections
- User can require n times more total internal reflections by using (# of expected reflections – n).
  - n = 2 is sufficient to make sure everything works.
- If # of expected reflections < n, do nothing.



Fast simulation

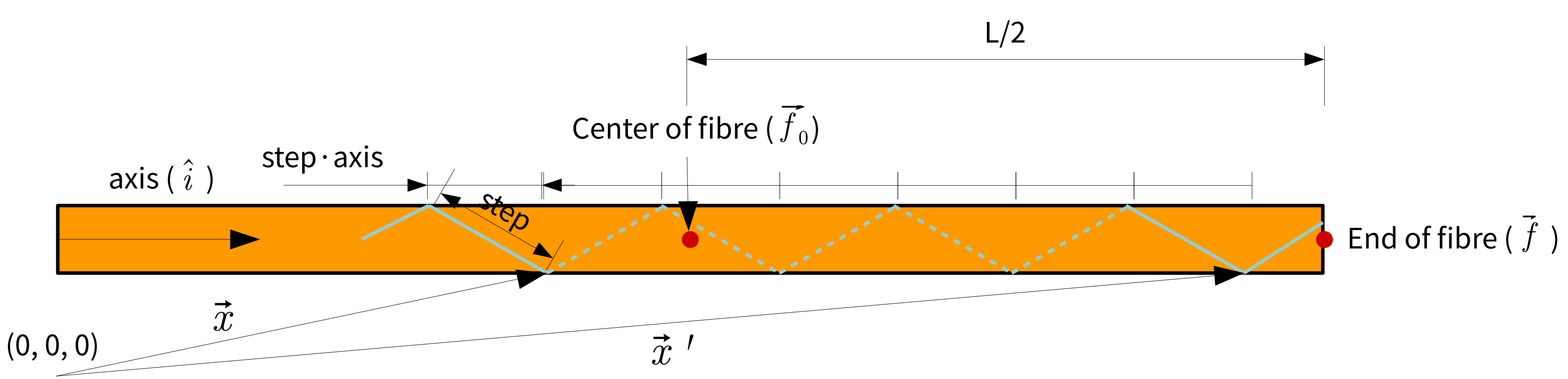
Checking absorption probability of an optical photon

- Skipping intermediate tracking of optical photon forces to check absorption probability by the model.
- In GEANT4, interaction probability with a matter of a particle is given as a ‘lifetime’ as a unit of interaction length.

  i.e., # of interaction length left = -std::log(G4UniformRand())
- The particle is killed when the travel length exceeds # of interaction length left.

- For a fast transported optical photon, absorption can be checked via
  - # of expected reflections × steplength / attenuation length > # of interaction length left
- Attenuation length of a material can be accessed using G4MaterialPropertyTable.

  matPropTable→GetProperty(kABSLENGTH)→Value(momentum)



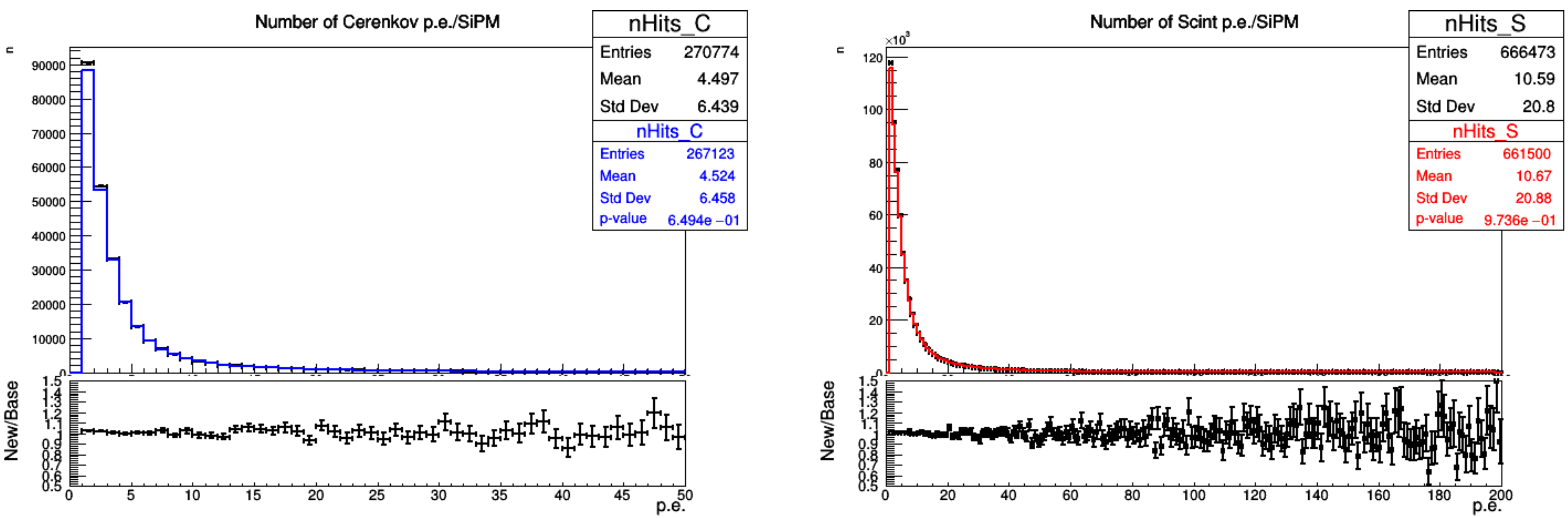
Speeding up optical photons tracking

Comparison to full tracking & conventional fast simulation (GFlash)

- Role of the developed fastsim module is limited to managing optical photon transportation inside optical fibres.
- EM/hadronic physics is same to that of full simulation of G4, i.e. does NOT utilize any shower parameterization.

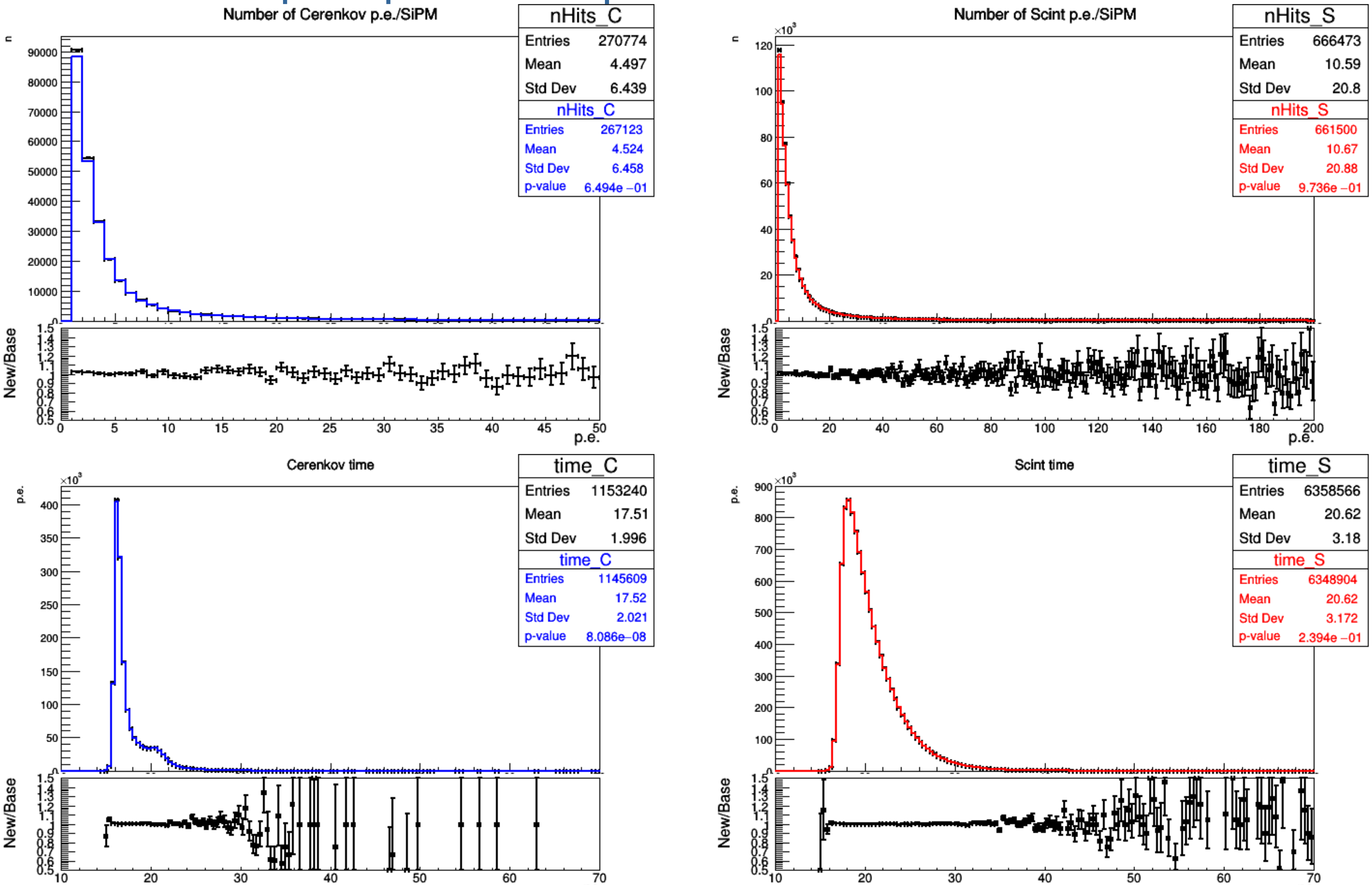
	GEANT4 fullsim	Fast Op transport	GFlash
Shower physics	Full tracking		Parameterization
Relative differences to GEANT4 fullsim	N/A	Optical photons inside optical fibers	EM/hadronic particles in the region of interest

- Comparison of Fast Op transport to full tracking shows good agreement.



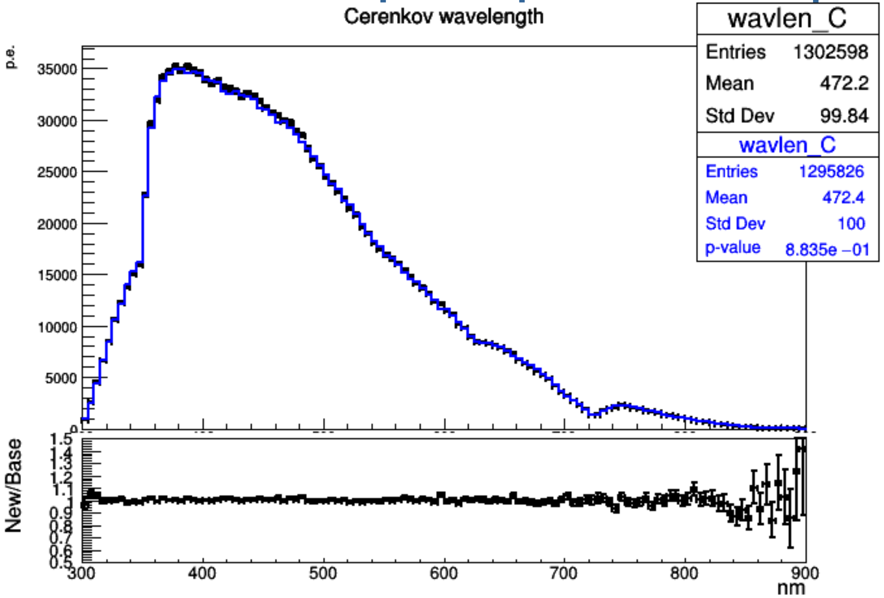
Fast simulation

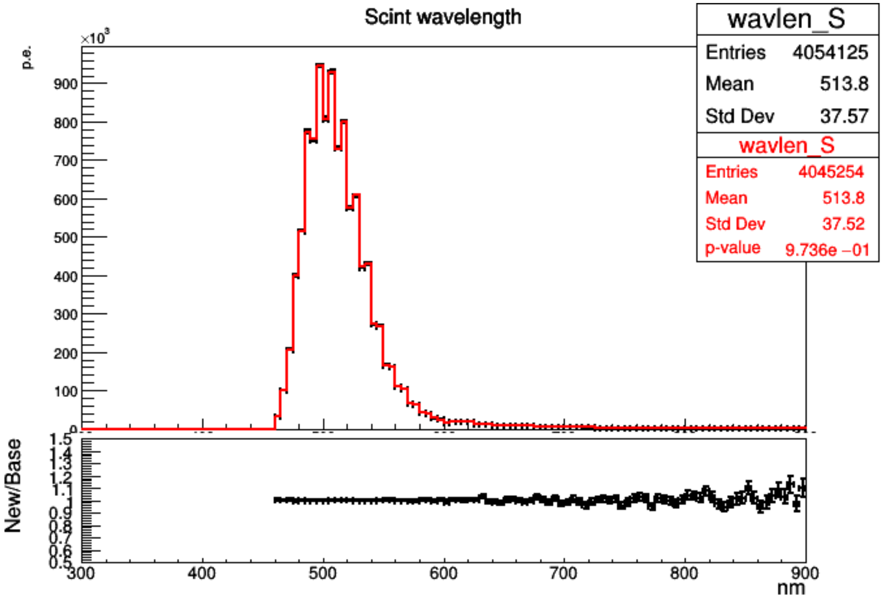
Validation of fast optical photon transportation



Fast simulation

Validation of fast optical photon transportation





Improvement in CPU consumption using fast optical photon transportation

- It takes 4.62 ± 1.17 min in average to produce an event (tested with 1000 of 20 GeV electron events).
- While it was 304 ± 88 min when using full tracking with the same server.
- Almost ~ 70 times faster than full tracking!
- Initial proposal of the idea was presented at GEANT4 R&D meeting [link].

Light attenuation correction (1)

Light attenuation correction

- π+ can go deep inside tower compared to e-.
- Although filters are applied to S channel to mitigate the light attenuation, energy measured from S channel should be corrected to take into account of attenuation properly.
- Can be corrected by measuring the shower depth event-by-event, using time structure of the scintillation signal.

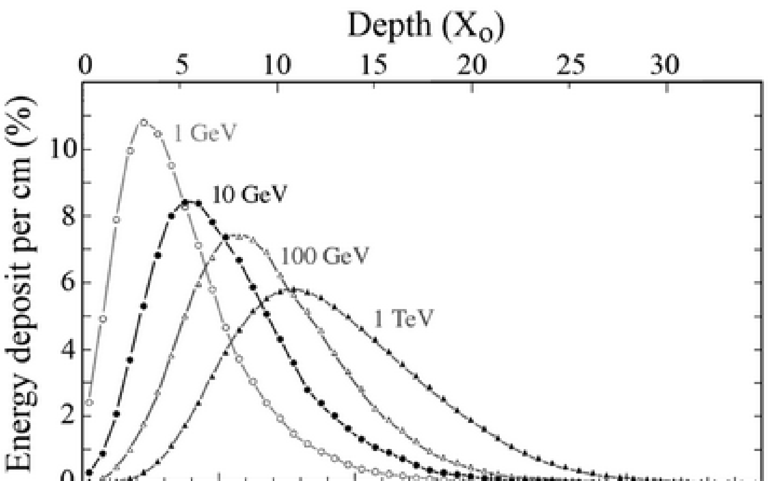
Shower depth as a function of time

- Shower depth x can be represented as a function of detection time

$$t_c = \frac{1}{0.3\,m/ns}x + \frac{1.8\,m}{0.3\,m/ns}$$ TOF of π+ in vacuum/tower

$$t_v = \frac{2.5\,m - x}{v}$$ Propagation time of optical photons

$$t_{max} = t_v + t_c = \frac{2.5 - x}{v} + \frac{x}{0.3\,m/ns} + \frac{1.8\,m}{0.3\,m/ns}$$ Detection time



Longitudinal profile of EM shower (EGS4)

Estimation of average optical photon velocity

- The average velocity of optical photons (v) can be estimated by calculating effective radiation length of the tower & exploiting well-known longitudinal profile of EM showers.

	Cu	PS	PMMA
Volume (%)	65.1	17.45	17.45
X0 (cm)	1.436	41.31	34.07
X0_eff (cm)		2.1613	

Light attenuation correction (2)

Light attenuation correction

- Estimated avg velocity of optical photons using 20GeV e- evts.

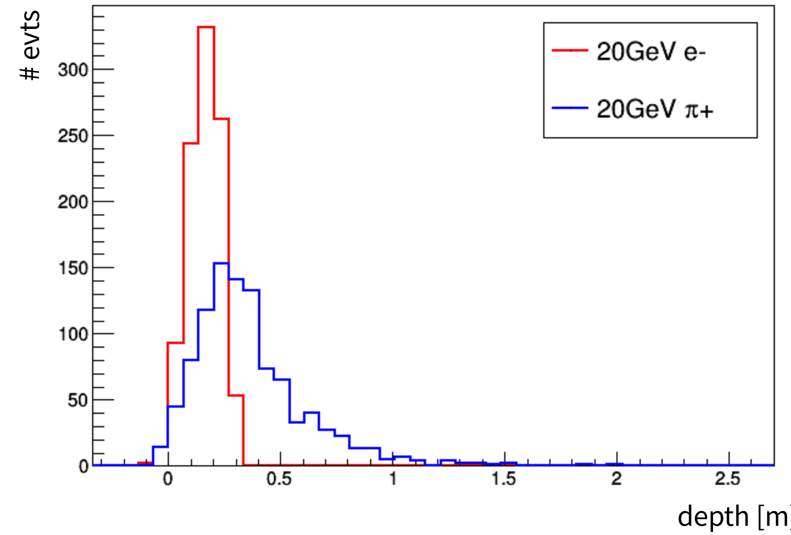
$$v = \frac{2.5\,m - 0.1368\,m}{t_{max} - \frac{0.1368\,m}{0.3\,m/ns} - \frac{1.8\,m}{0.3\,m/ns}}$$

- Shower depth can be estimated event-by-event.
- Average measured energy shows exponential dependency on the depth of a shower.

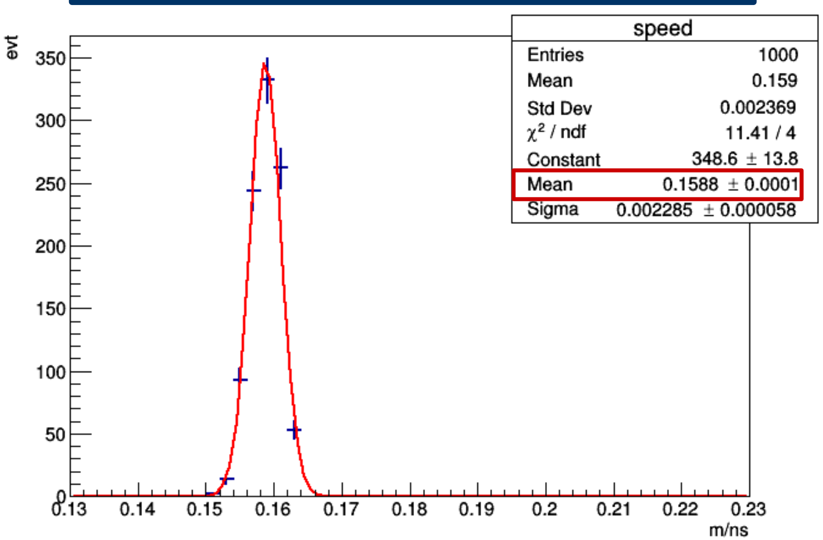
$$E = E_{6.33\,X_0} \exp \frac{x - 6.33\,X_0}{\lambda_{eff}}$$

- Removing the exponential term corrects the attenuation loss.

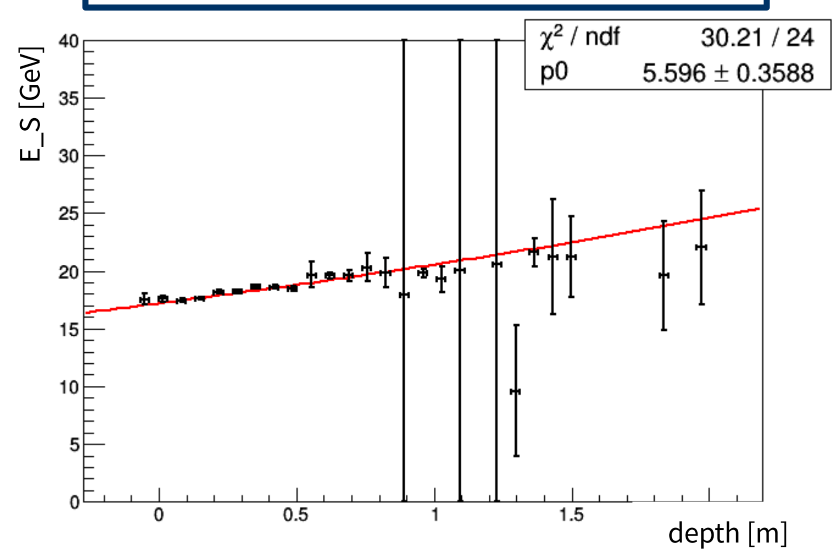
Shower depth of 20 GeV e- & π+



Velocity of optical photon v within fibers



Avg E_S vs shower depth for 20GeV π+



Title

Text

$$formula$$