Highlights of the computing and data handling session at ICHEP2022

Weidong Li

CEPC Day Meeting

29 Aug 2022

Software optimization and modernisation

- Improvements in ROOT to make analysis easier
- Faster fitting in RooFit

Simpler, faster analysis with modern ROOT

Enrico Guiraud, I. Kabadzhov , A. Naumann, V. Padulano, Pawan J., E. Tejedor
for the ROOT team
ICHEP 2022, 8/7/2022

A swiss-army knife for data analysis

Analysis life cycle

skimming, ntuple production | quick exploration, first implementation | systematics, scale out

Platforms

laptop or PC | many-core machine | computing cluster + job submission

Analysis languages

⬇ ~50% C++
⬆ ~50% Python

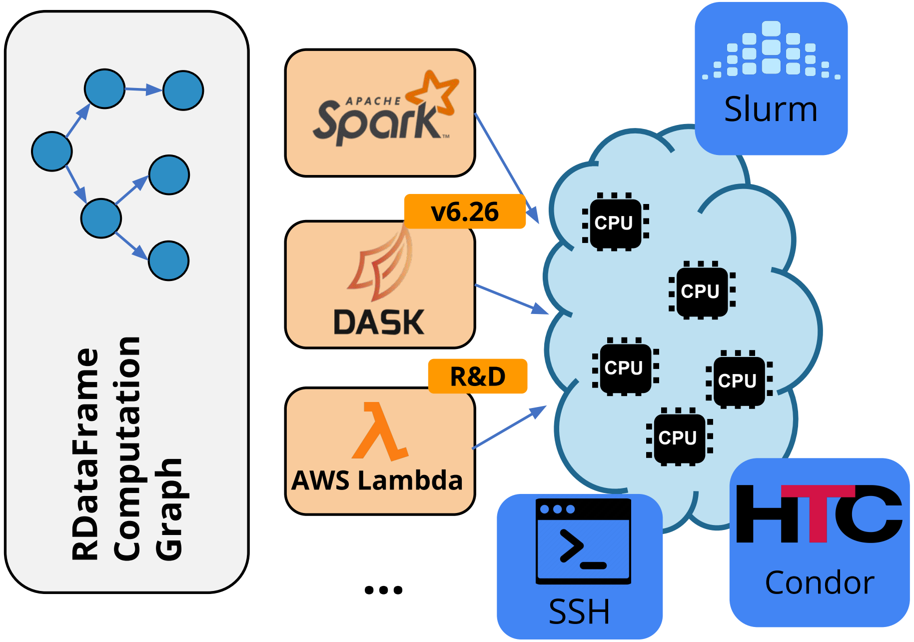
Storage

local disk
fast-access network storage
EOS or other not-so-fast backend

ROOT.RDataFrame is a modern analysis interface that addresses all these use cases with **one high-level programming** model that performs well, scales well and enables **HEP-specific ergonomics**, in C++ and Python.

Distributed execution with RDataFrame

- Enables **interactive large-scale distributed** data analysis
- Python RDF API, C++ event loop
- Full access to ROOT I/O
- Let Spark/Dask/HTCondor/Slurm/SSH.... take care of scheduling and resource management
- Transparently merges results coming from different computing nodes



Wide adoption from analysts

- Dark matter sensitivity study (Pani & Polesello, 2018)
- Distributed analysis with RDataFrame in TOTEM (Avati et al., 2019)
- **ATLAS**: prototype xAOD data source DOI 10.5281/zenodo.1303038
- **ALICE**: Apache Arrow support contributed by G. Eulisse
- **FCC** is developing analysis workflows based on RDF (see also the GitHub project)
- Building block in INFN analysis facility effort
- many users **"in the wild"**: 650+ threads tagged #rdataframe on the ROOT forum, about the same as #tree and #hist

RDF as a framework building block

Some examples of analysis software based on RDataFrame

- bamboo (recent talk)
- KIT's CROWN (recent talk)
- W mass analysis framework
- LoopSUSYFrame ATLAS analysis tool
- ("Latinos" CMS framework planning transition to RDF)
- narf (recent talk)
- ...

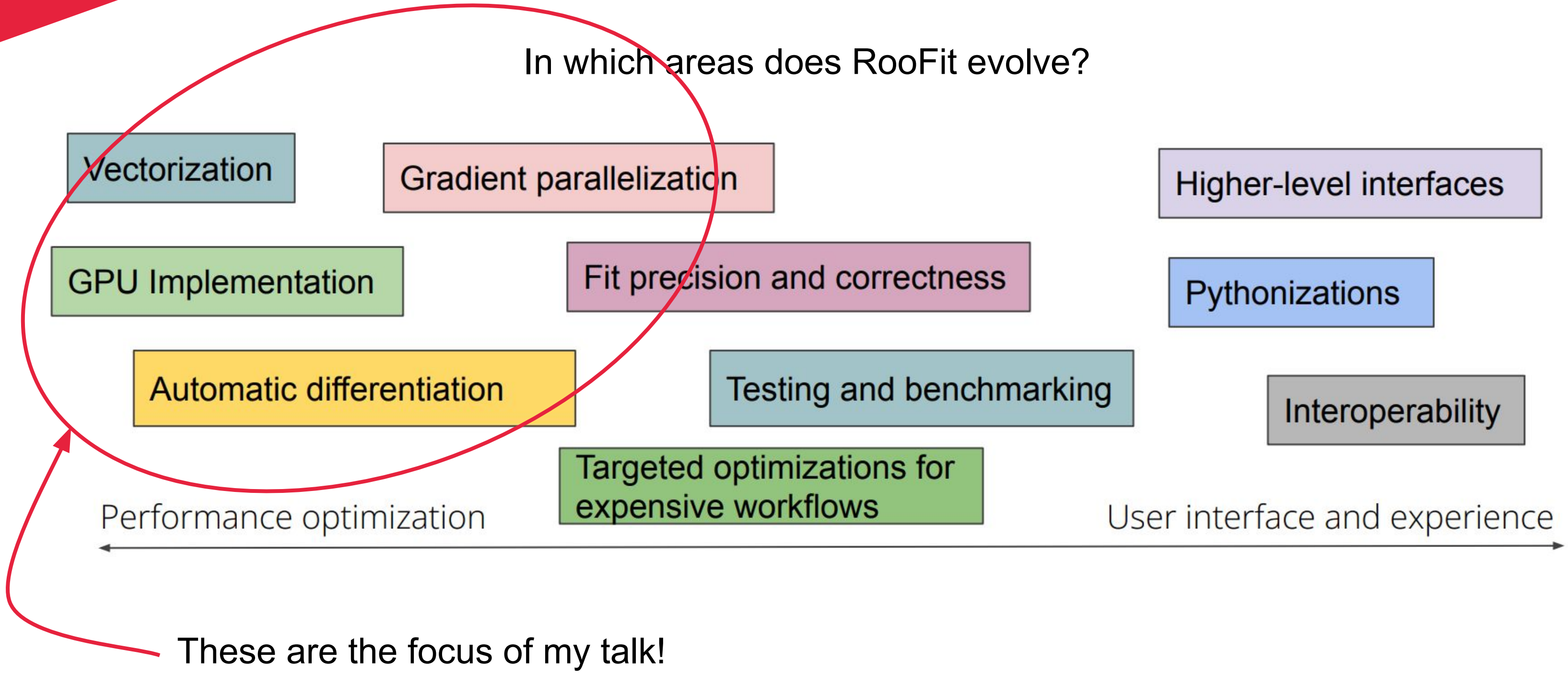
Feedback (and code) from users regularly integrated upstream (*thank you*!)

New RooFit Developments to Speed up your Analysis

Zef Wolffs (Nikhef), Carsten Burgard (DESY), Jonas Rembser (CERN),
Lorenzo Moneta (CERN), Wouter Verkerke (Nikhef),
Patrick Bos (Netherlands eScience Center)

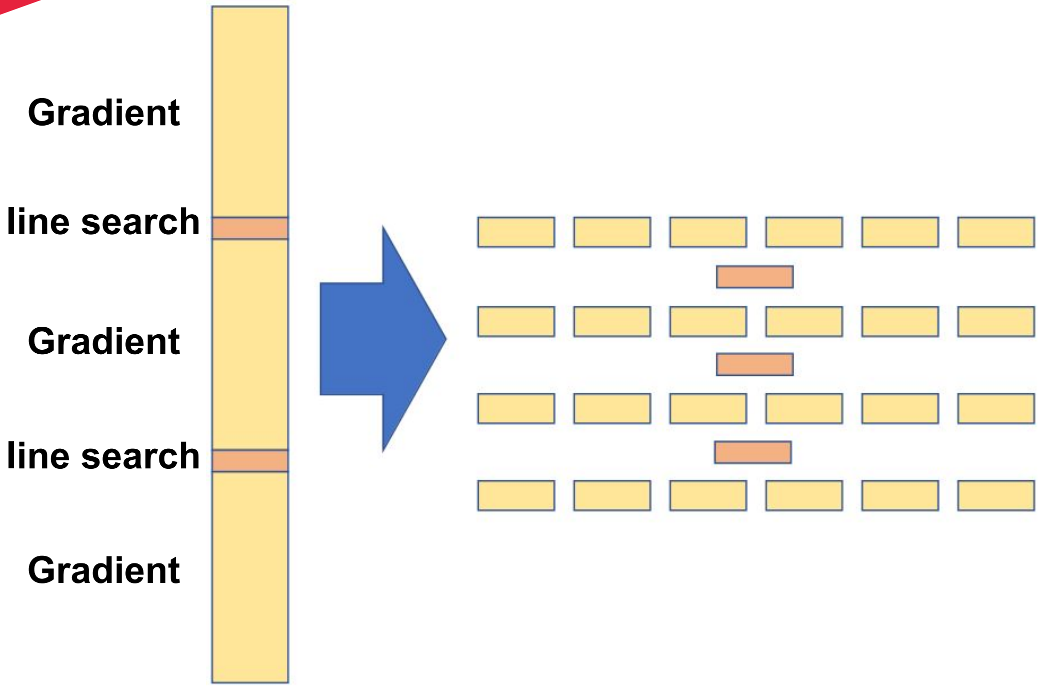
ICHEP 2022, 08-07-2022

Introduction



Gradient Parallelization

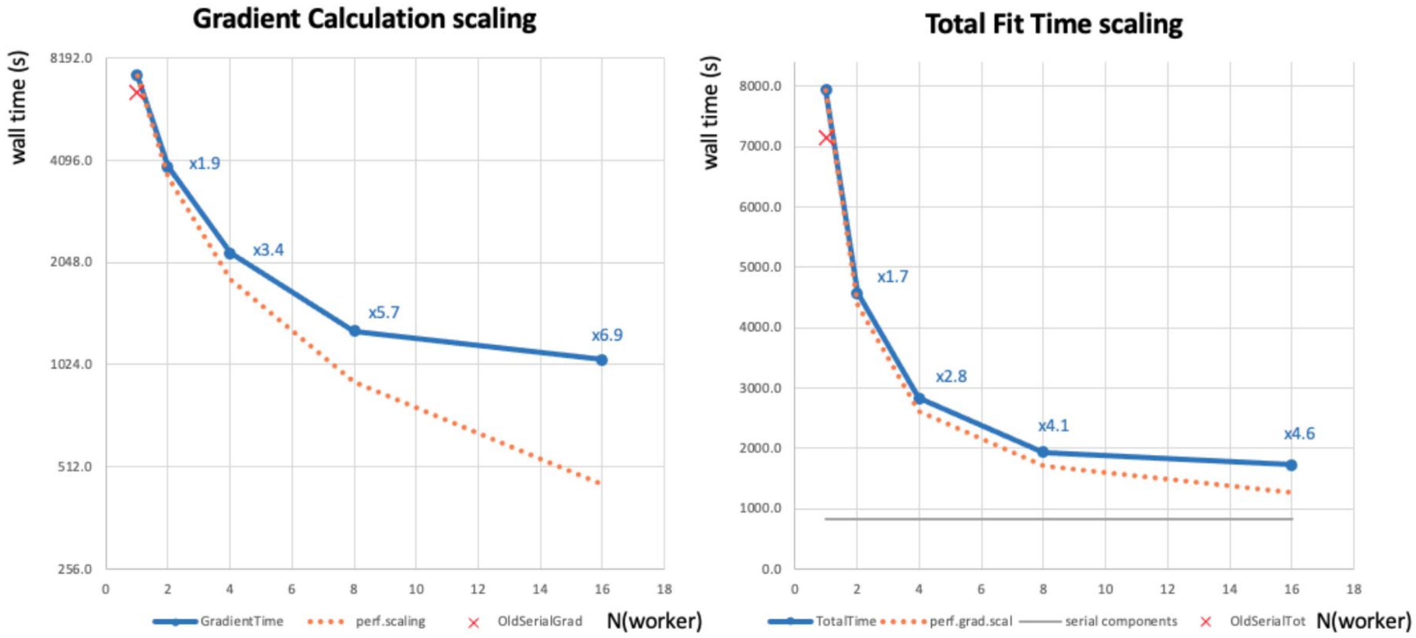
CPU parallelization - parallel MIGRAD



- Parallelize at gradient calculation level
- Dynamic load balancing over workers through random work stealing algorithm
- Designed to have maximum speed impact of complex fits with many parameters
- Line search has limited impact on scaling, but this was investigated further

[3] Bos, E. G. P., Burgard, C. D., Croft, V. A., Hageboeck, S., Moneta, L., Pelupessy, I., ... & Verkerke, W. (2020). Faster RooFitting: Automated parallel calculation of collaborative … (see last slide)

CPU parallelization - Higgs combination



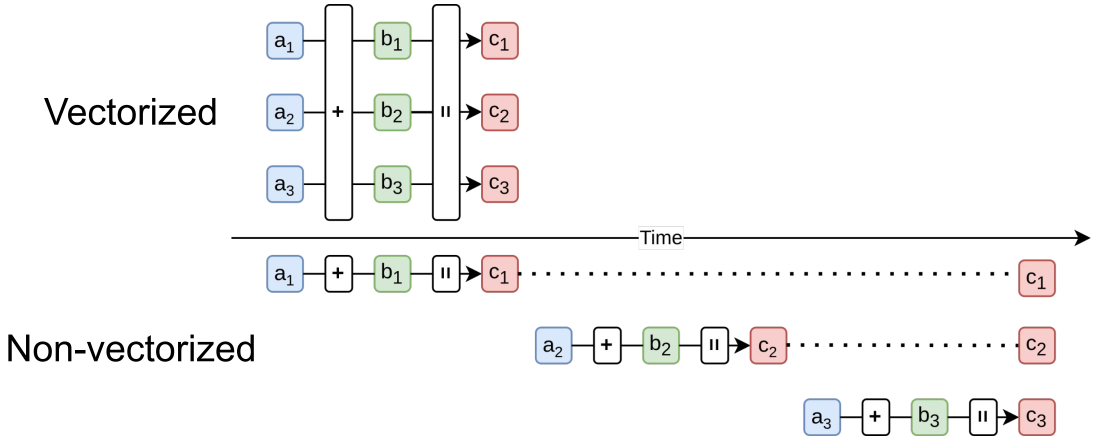
- Walltime decrease in Higgs combination fit from from **2h12m26s → 28m52s**
- Serial time expenditure close to parallel time expenditure with single worker
- Fit validated to conform to serial run, all parameters agree within 1% of estimated uncertainty

Batched Computation

Batched Computation - Motivation

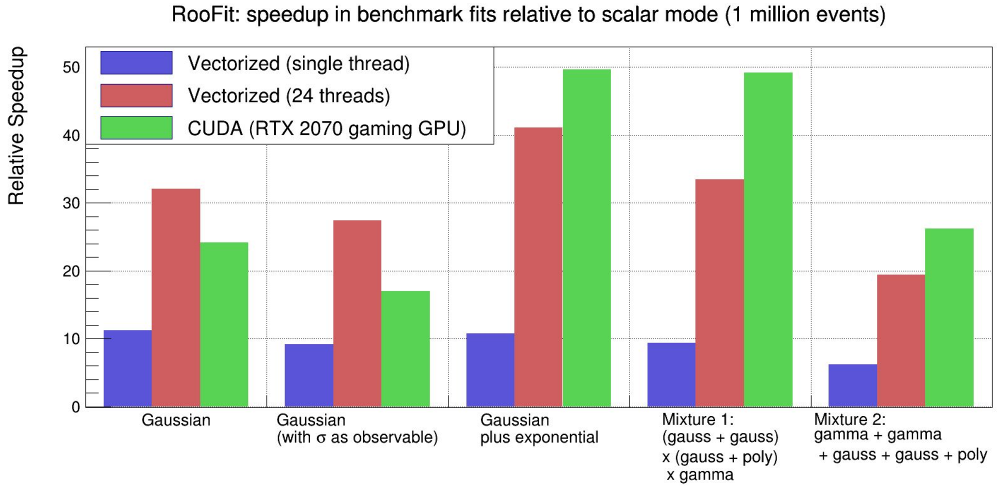
- Batched computation in RooFit refers to the principle of computing batches of events simultaneously, this has the following benefits:
  - GPU parallelization: Simple operations applied on each of the O(1000)+ CUDA cores on the GPU in parallel

  - Vectorization: Most modern processors are equipped with instruction sets which apply the same operation simultaneously to multiple pieces of data, or “batches”



Batched Computation - Results

- For unbinned fits with large numbers of events we see a huge speedup using batched computation
  - Vectorization alone effective, but with also multithreading and GPU parallelization (CUDA) we record even larger speedups
- Speedup grows with the size of the dataset that is being evaluated
  - Especially true for GPU due to relatively large overhead of data transfer to GPU cores



Common Software

- Common HEP event data model (EDM4Hep)
- A turnkey software stack for HEP (Key4Hep)

EDM4hep -
a common event data model for HEP experiments

ICHEP 2022, Bologna (IT)

09.07.22

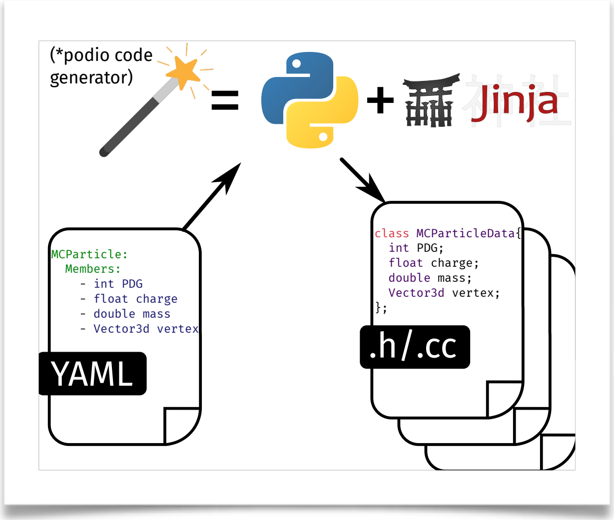
Placido Fernandez Declara (CERN), Frank Gaede (DESY), Gerardo Ganis (CERN), Benedikt Hegner (CERN), Clement Helsens (KIT), Thomas Madlener (DESY), Andre Sailer (CERN), Graeme A Stewart (CERN), Valentin Volkl (CERN)

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

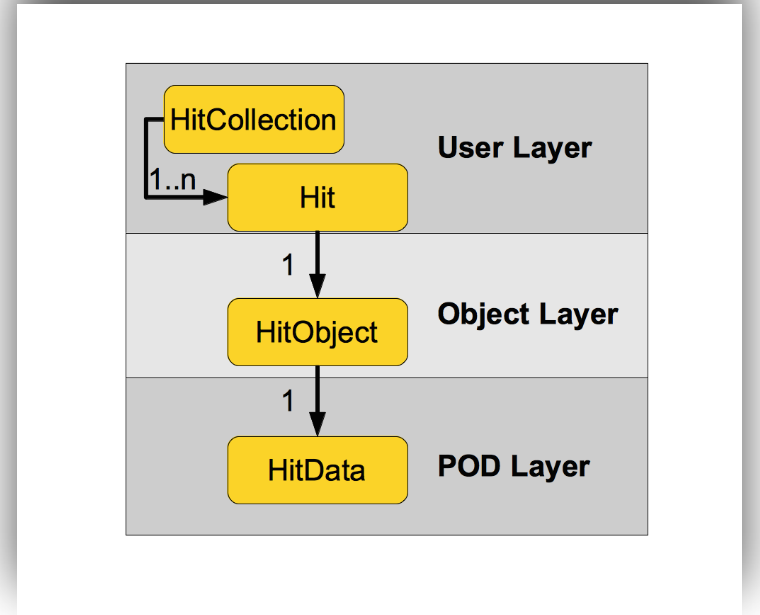
PODIO

the EDM toolkit

- PODIO developed in **AIDA2020** in context of FCC study
    - CHEP2016 - J. Phys.: Conf. Ser. 898 072039
    - CHEP2019 - EPJ Web Conf. 245 (2020) 05024
    - CHEP2021 - EPJ Web Conf. 251 (2021) 03026
- first used in **FCC-edm/pLCIO** and now for **EDM4hep**



- use ***yaml*-files** to define EDM objects then **generate C++** code via **Python/Jinja** scripts
- three layers of classes:
    - **POD** layer - the actual data in array of structs
    - **Object** layer - add relations and vector members
    - **User** layer - thin handles and collections



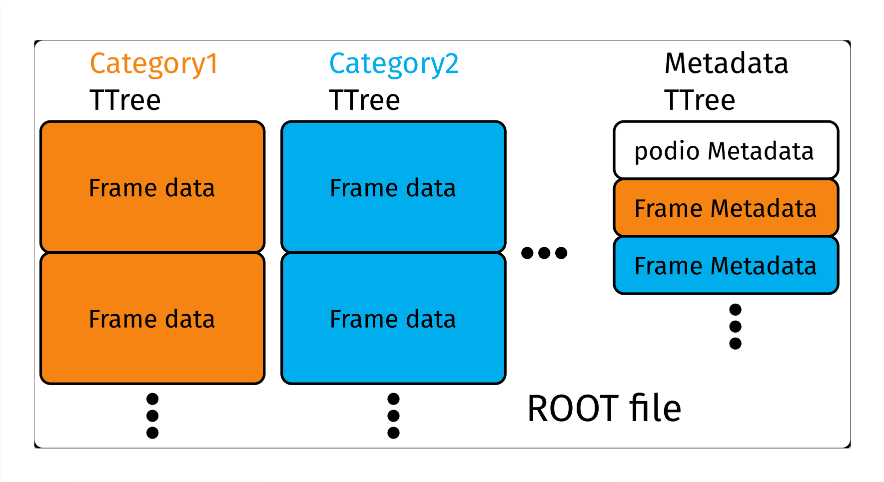
PODIO

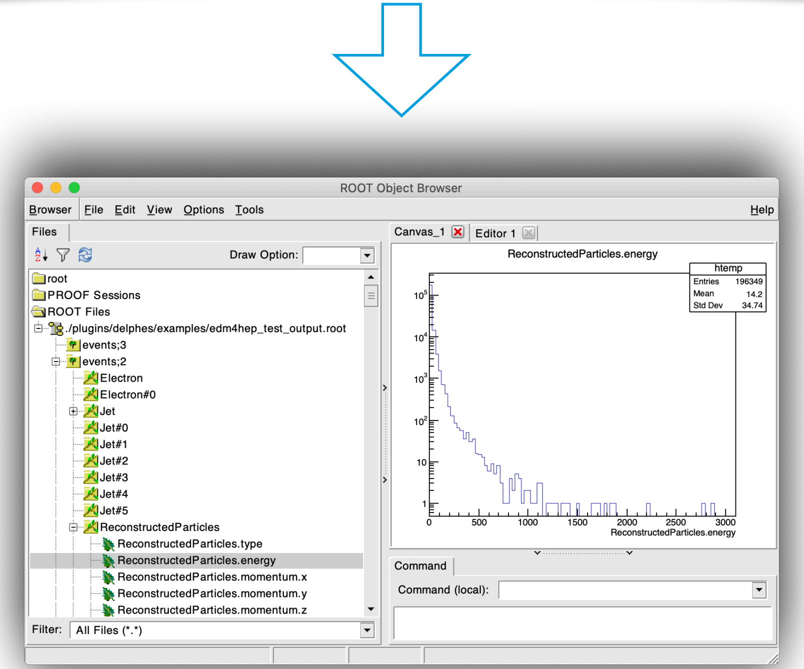
I/O backends and File formats

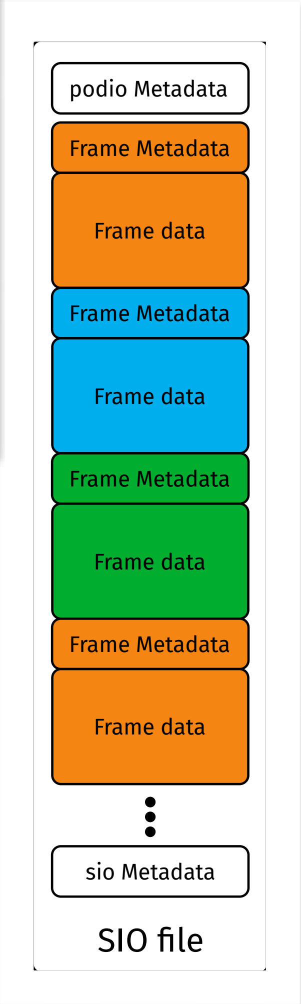
- **default** I/O backend: **ROOT**
  - POD buffers are stored as branches in a *TTree*
- files can also be read without EDM library(!)
  - e.g. RDataFrame, TBrowser, uproot,…

- **alternative** I/O backend: **SIO**
  - persistency library used in LCIO
  - complete events are stored as binary records
  - … faster reading of complete events…

- adding more I/O backends is possible
  - e.g HDF5



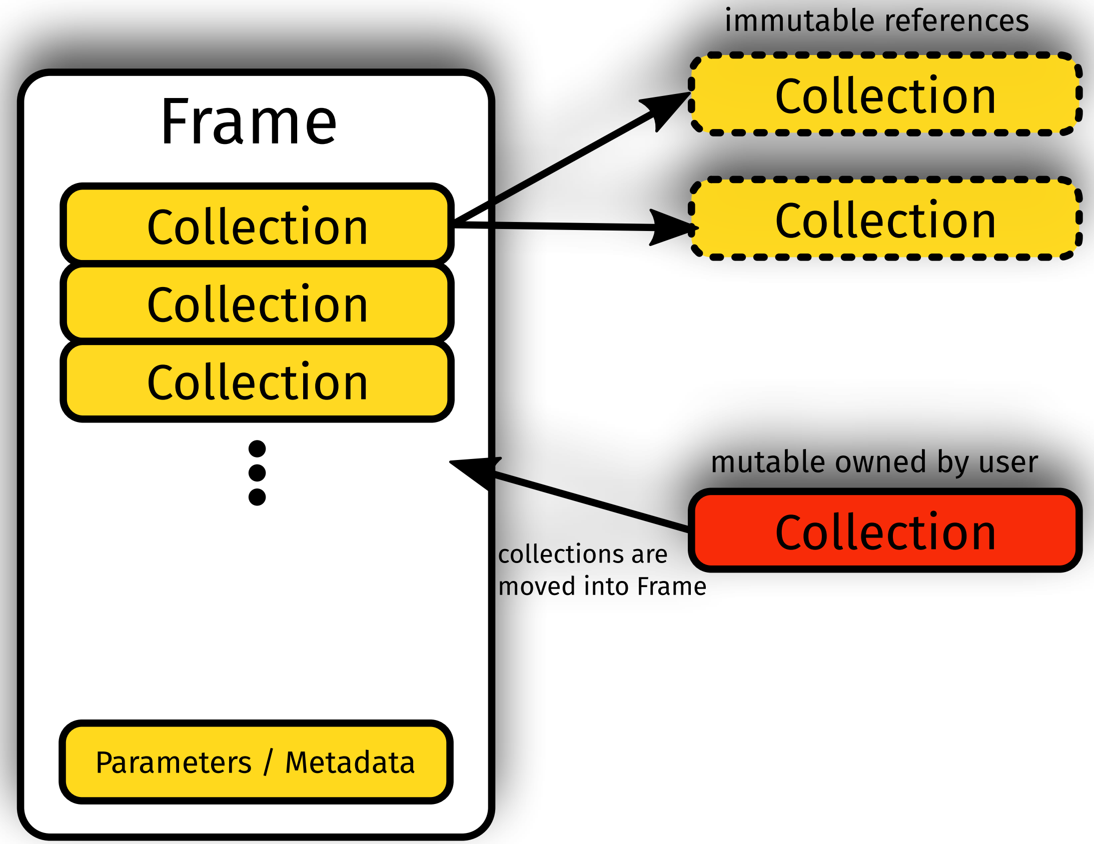




PODIO recent developments

Frames

- replaced initial (and experimental) EventStore in PODIO w/ a **new Frame concept**
- Frames can hold data for any validity range, e.g.
  - events, runs, lumi sections, …
  - previously ‘events only’
- attempt to guarantee ***thread-safety***:
  - after a collection has been put (moved !) into a frame only *immutable* access possible
- implemented for ROOTWriter and SIOWriter
- prepare for additional features:
  - hooks schema evolution
  - potential lazy unpacking
  - …

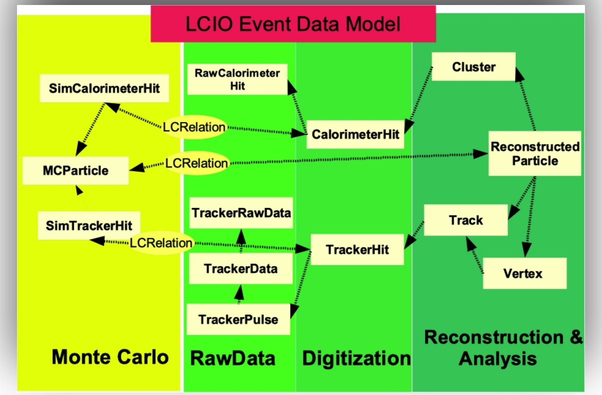


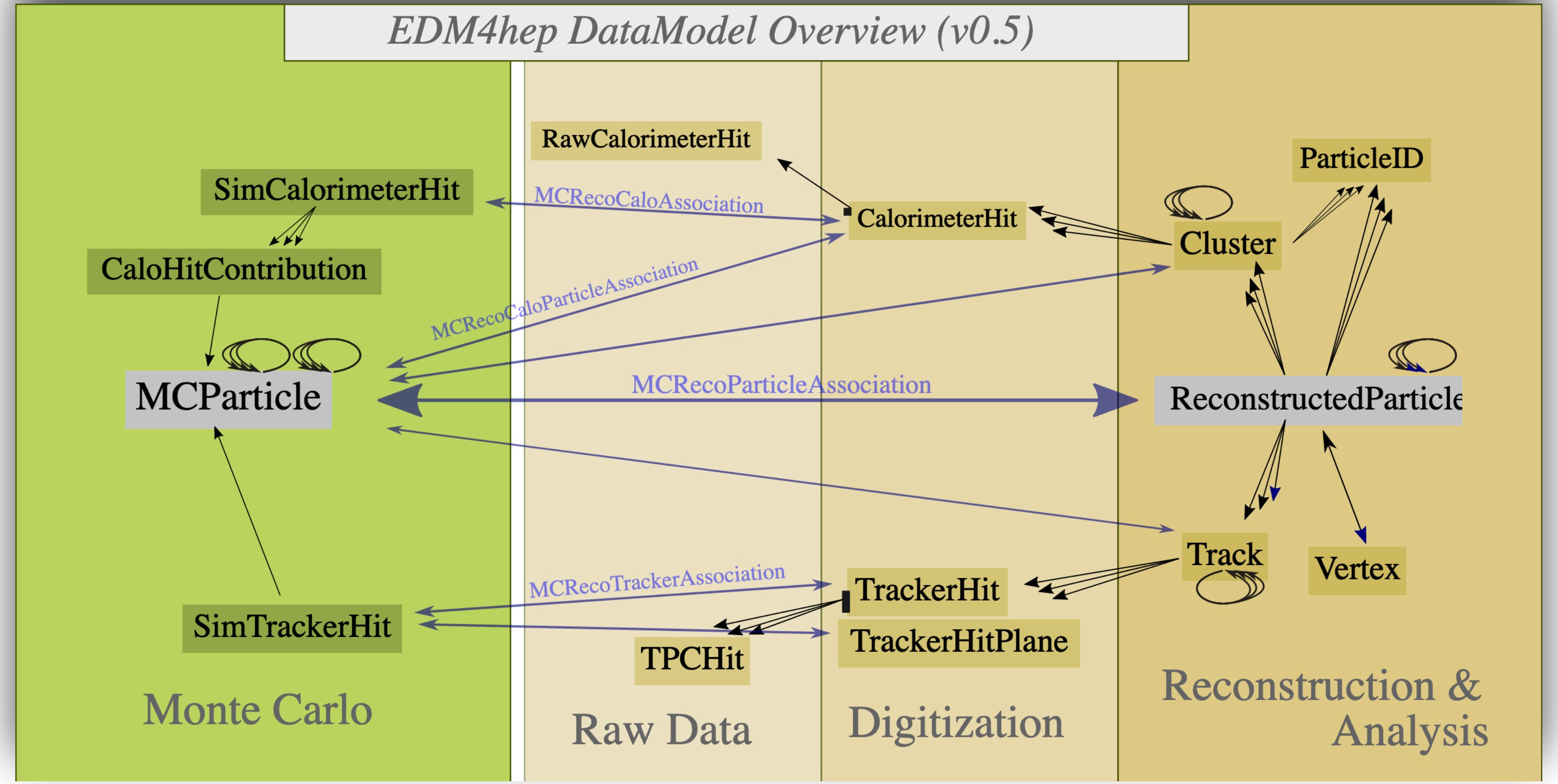
file format changed with introduction of Frames
should be last non-backward compatible change in PODIO

EDM4hep

the actual event data model

for more details: https://edm4hep.web.cern.ch/



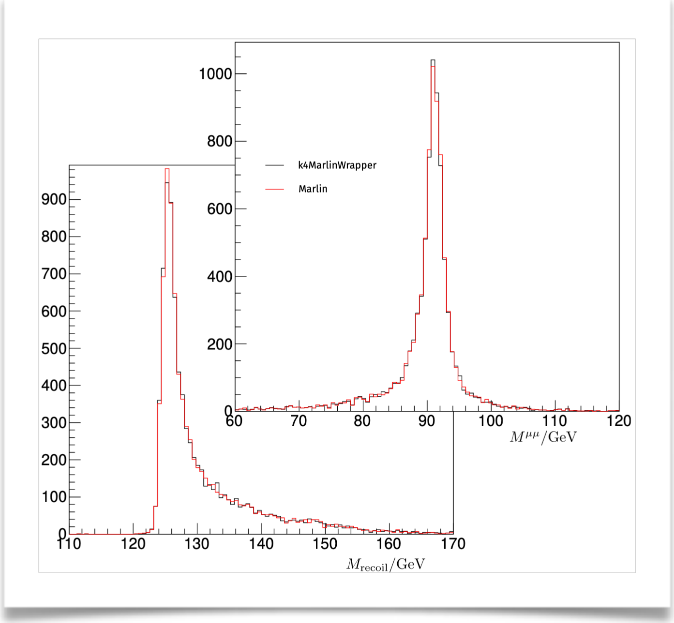


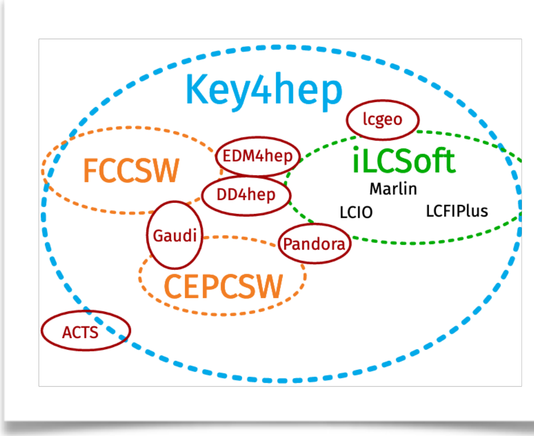
- **hierarchical** EDM from MC-truth (MCParticle) to high level objects (ReconstructedParticles)
  - too a large extent **one-to-one correspondence w/ LCIO**
  - should serve the needs of **all lepton colliders** - potentially some extensions/add-ons for others ?

EDM4hep in use

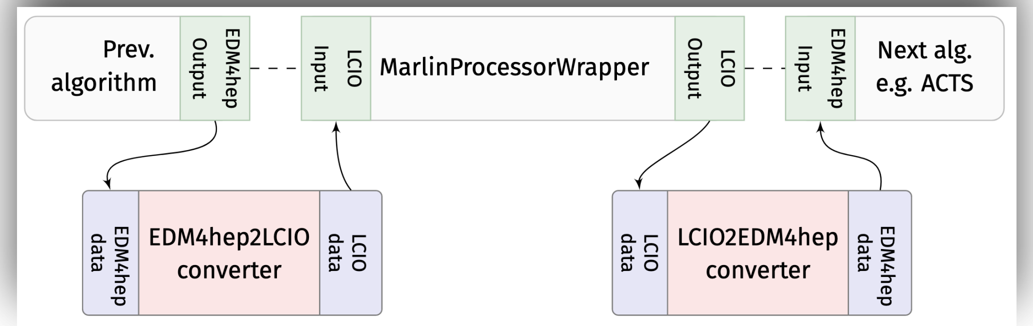
for the linear collider community: ILC, CLIC

- linear collider community is moving to Key4hep and **EDM4hep** in adiabatic way
- need to **preserve** all existing **code base** and algorithms
- possible via MarlinWrapper (see Key4hep talk)





- LCIO <-> EDM4hep conversion available
- can run full simulation and reconstruction chain as before and write **EDM4hep output**
- plan to finalise the physics validation of EDM4hep output this summer

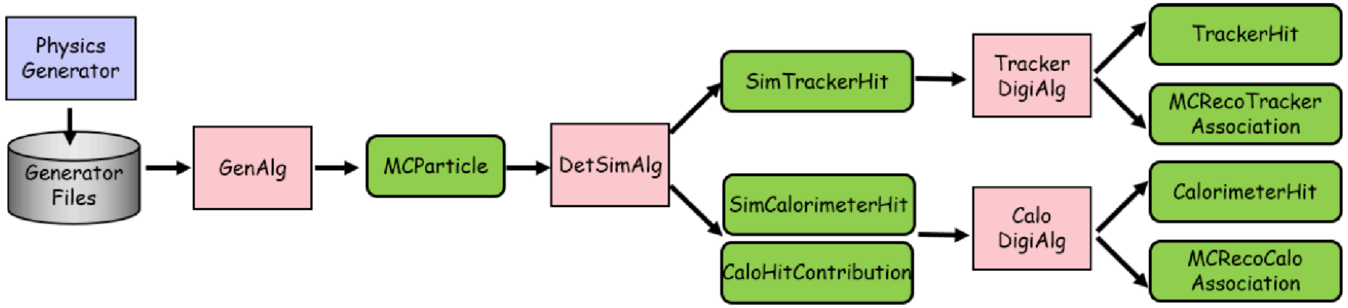


EDM4hep in use

for the circular collider community: FCCee, CEPC

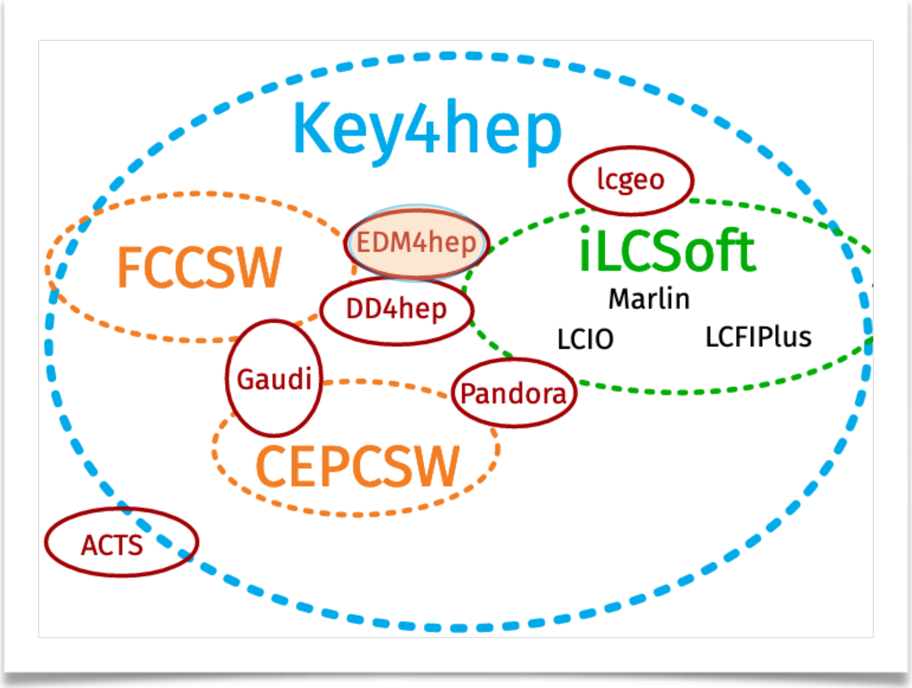
- **EDM4hep** used in **FCCAnalysis**
    - Python scripts using **RDataFrame** on EDM4hep ROOT files directly
    - **fast** yet not using the EDM4hep API
    - the only **production use** of EDM4hep so far

- CEPC community considering to port existing iLCSoft algorithms from Marlin to Gaudi
    - replacing **LCIO** with **EDM4hep**



Summary and Outlook

- **EDM4hep** is a **common event data model** for HEP
  - developed in context of **Key4hep** software eco system
- based the EDM toolkit PODIO
- **PODIO** recently undergone significant new developments:
  - introduction of **Frames** (thread-safety)
  - **schema evolution**
- EDM4hep adopted by **ILC, CLIC, FCC, CEPC**
  - under investigation by **EIC**



aiming to have **first production releases** v1.0 this summer
will guarantee **backward compatibility** from then on

The Key4hep turnkey software stack

ICHEP 2022

July 09th, 2022

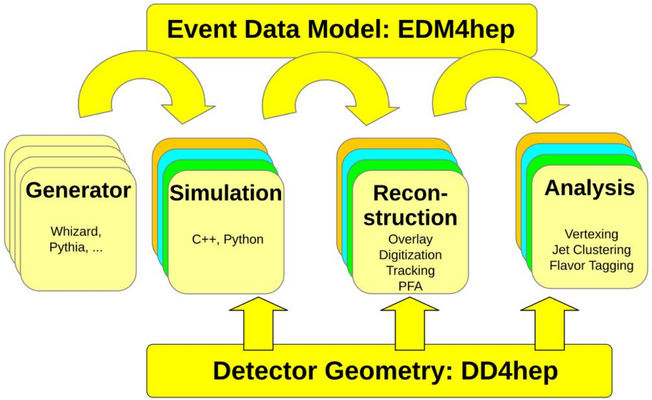
Valentin Volkl, for the Key4hep SW group
CERN

This work benefited from support by the CERN Strategic R&D Programme on Technologies for Future Experiments (https://cds.cern.ch/record/2649646/, CERN-OPEN-2018-006).

Common software tools for future colliders - a.k.a. Key4hep

Software stack that connects and extends packages to provide a complete data processing framework, comprising fast and full simulation, reconstruction, and analysis.

- **Contributions from different Future Collider communities**
  - FCC, CLIC, ILC, CEPC, EIC, ...
- **Consistent choice of technologies for interoperability**
  - EDM4hep: data model
  - Gaudi: framework
  - DD4hep: geometry description
  - Spack: package manager
- **Ease of use for librarians, developers and users**
  - Provide examples, documentation, templates and common practices



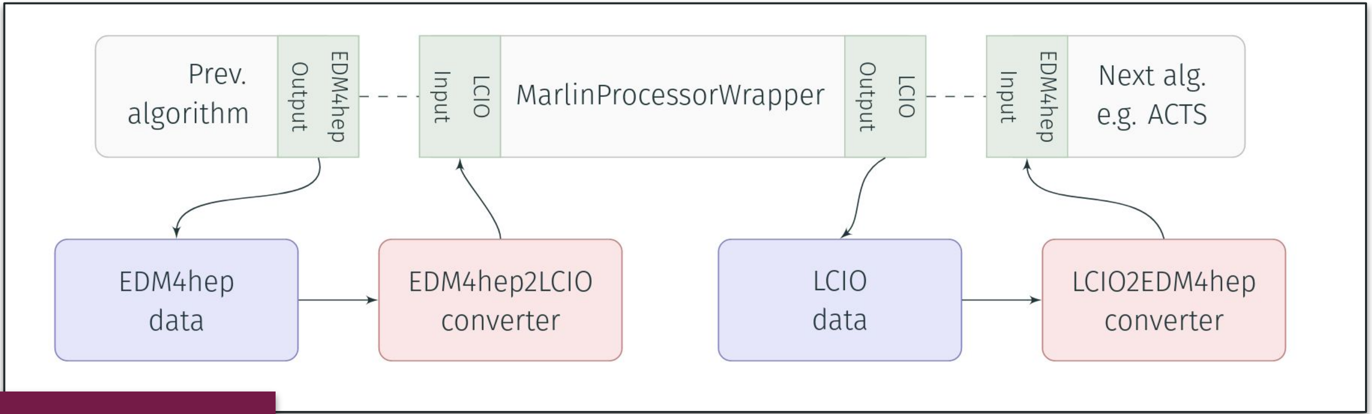
iLCSoft reconstruction chain

- Standard reconstruction for CLD:
  - Background Overlay, Digitisation
  - Track Pattern Recognition (*ConformalTracking*), track fit
  - Particle Flow Reconstruction (*PandoraPFA*)
  - Vertexing and Flavour Tagging (*LCFIplus*)

… and more (FastJet, KinematicFitting, Particle Flow Reconstruction …) available

Full iLCSoft Reconstruction chain in Key4hep available through **k4MarlinWrapper**

- Allows running all existing *Marlin* processors from iLCSoft in the Key4hep Gaudi framework



Andre Sailer, Placido Fernandez

ACTS A Common Tracking Software

Project to preserve and enhance LHC track reconstruction software for future **detectors**
Widely used already: ATLAS, ALICE, Belle II, sPHENIX ...

A flexible, **open source R&D testbed**:

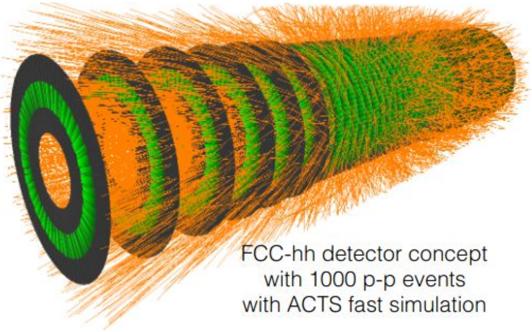
- Facilitate collaboration across experiments and external contributors, e.g. machine learning experts
- Allow for novel algorithms and detector components (e.g. timing, tracklets)

A high-performant toolbox for track reconstruction based on LHC experience

- Modern code and software concepts to allow for concurrent computing
- Support high luminosity and high precision tracking algorithms

Very active ongoing efforts:

- Updating geometry loading for seamless use with FCC detector models
- Include existing EIC framework components



FCC-hh detector concept
with 1000 p-p events
with ACTS fast simulation

k4Acts: Andreas Salzburger, Paul Gessinger, Andreas Steffl

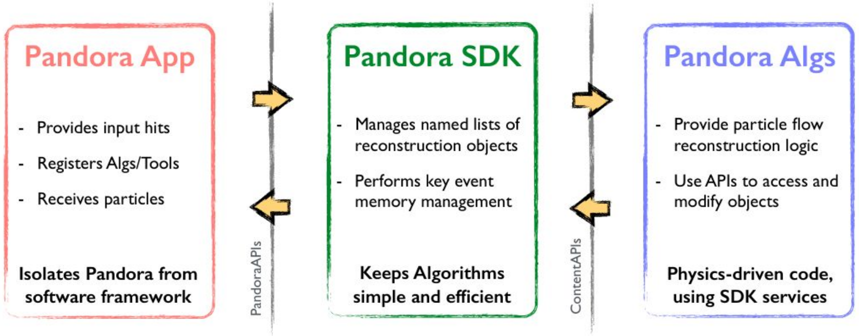
Workshop on GranuLAr noble liquid argon detectors

k4Pandora

First step towards particle flow: use existing components with some conversions

Ongoing development of a **dedicated key4hep package** (k4Pandora, under construction)

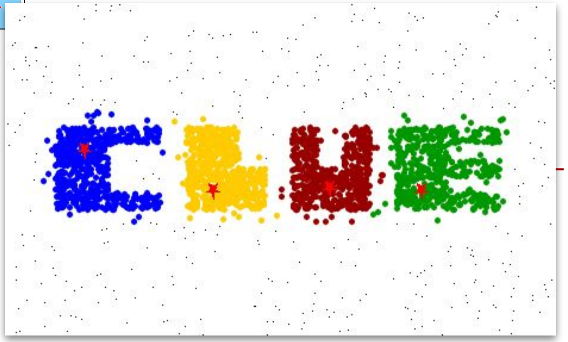
k4MarlinWrapper ↔ DDMarlinPandora ↔ **Pandora**



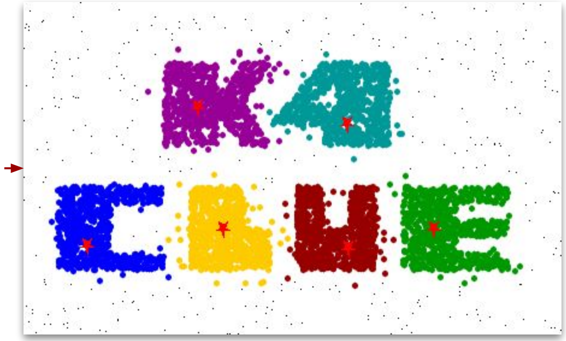

```
from Configurables import MarlinProcessorWrapper

pandora = MarlinProcessorWrapper('DDMarlinPandora')
pandora.OutputLevel = DEBUG
pandora.ProcessorType = 'DDPandoraPFANewProcessor'
pandora.Parameters = {
    'Verbosity': ['WARNING'],
    'PandoraSettingsXmlFile': ['/some/path'],
    'CreateGaps': [False],
    'ECalCaloHitCollections': ['ECalBarrelCells']
}
ApplicationMgr().TopAlg += [pandora]
```

Juraj Smiesko, Eduardo Ploerer

Integration of experiment-independent libraries:



https://gitlab.cern.ch/kalos/clue



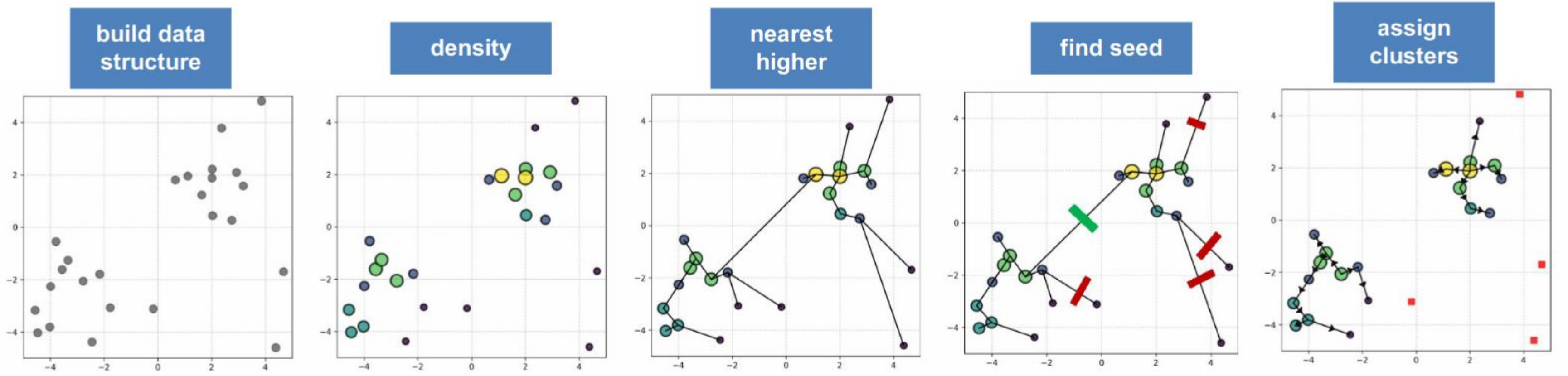
https://github.com/key4hep/k4Clue

Erica Brondolin, Marco Rovere, Felice Pantaleo

Link for more details

2D Clusters with CLUE

- CLUE (**CLUstering by Energy**) is an algorithm inspired by “Clustering by fast search and find of density peaks” [link to Ref.]
- Main characteristic:
  - *Energy density* - rather than individual cell energy - used to define ranking, seeding threshold, etc…
- GPU-friendly, i.e. suitable for the upcoming era of heterogeneous computing in HEP



Erica Brondolin, Marco Rovere, Felice Pantaleo

Usage of GPU/FPGA in HEP

- GPU/FPGA based tracking for ATLAS Event Filter
- Heterogeneous HLT at CMS
- Development to run LHCb payloads on GPUs

Event Filter Tracking for the Upgrade of the ATLAS Trigger and Data Acquisition System

Viviana Cavaliere for the ATLAS TDAQ Collaboration

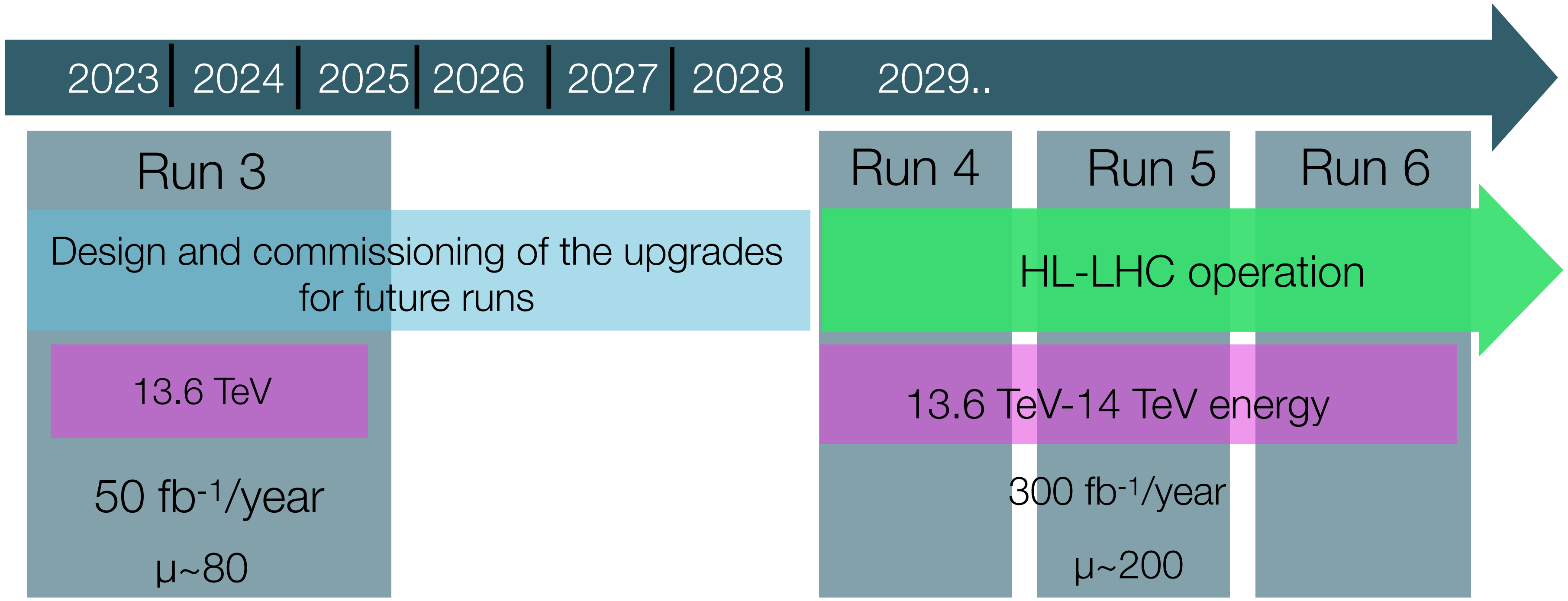
July 8th, 2022

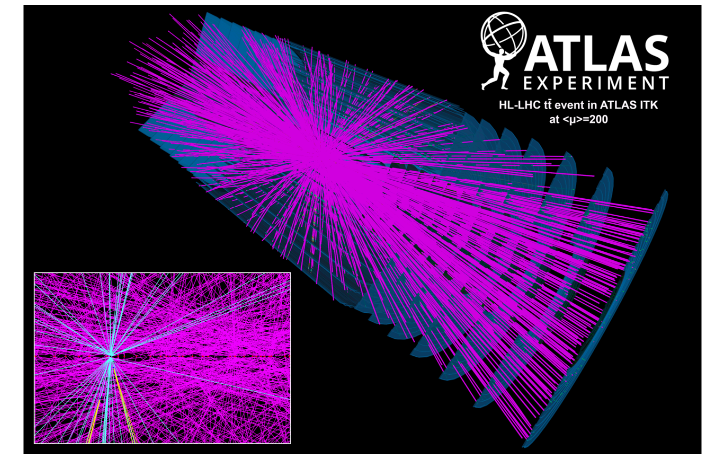
ICHEP 2022

Brookhaven National Laboratory

ATLAS EXPERIMENT

ATLAS Phase II upgrade

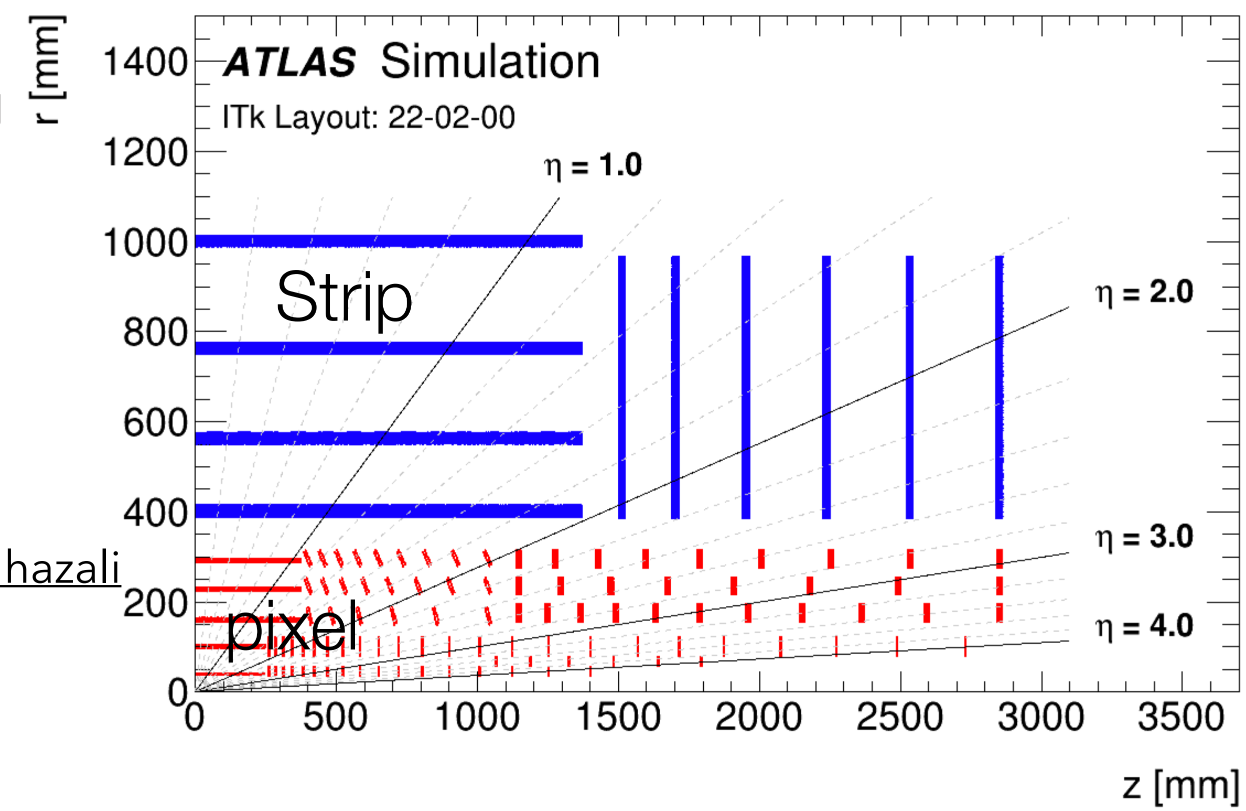




$t\bar{t}$ event at average pile-up of 200 collisions per bunch crossing.

- Conditions at the **HL-LHC,** with an average of 200 simultaneous collisions (pile-up) per bunch crossing expected, will be challenging for experiments:
  - **ATLAS is planning a major, including a new inner tracking detector, a lighter and more granular all-silicon tracking detector to allow high-precision reconstruction of charged particle tracks (ITk)**

    See talk by Yassine El Ghazali
  - **Triggering will become more difficult and time consuming**

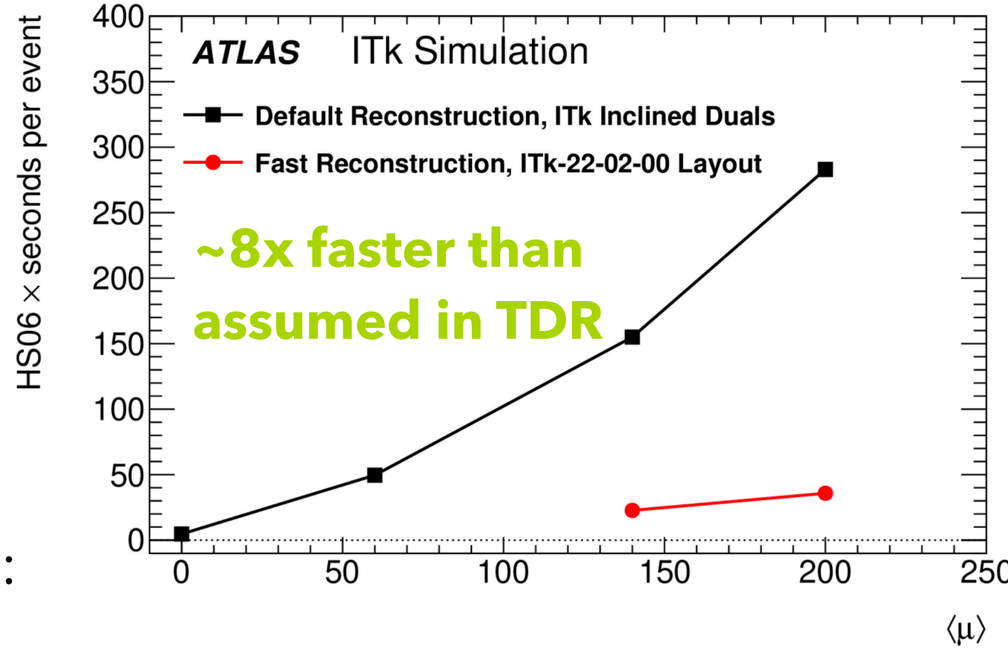


Brookhaven National Laboratory

Tracking at trigger level is essential to control rates while maintaining good efficiency for relevant physics processes

Tracking @ Event Filter

- The Technical Design Report (CERN-LHCC-2017-020) assumed a hardware-based track reconstruction based on associative memory ASICs and FPGAs
  - Plan to possibly process higher input rate (1MHz→4MHz)
    - Decided not to pursue the evolution scenario
  - Software tracking improvements
  - The rise of commercial accelerator cards



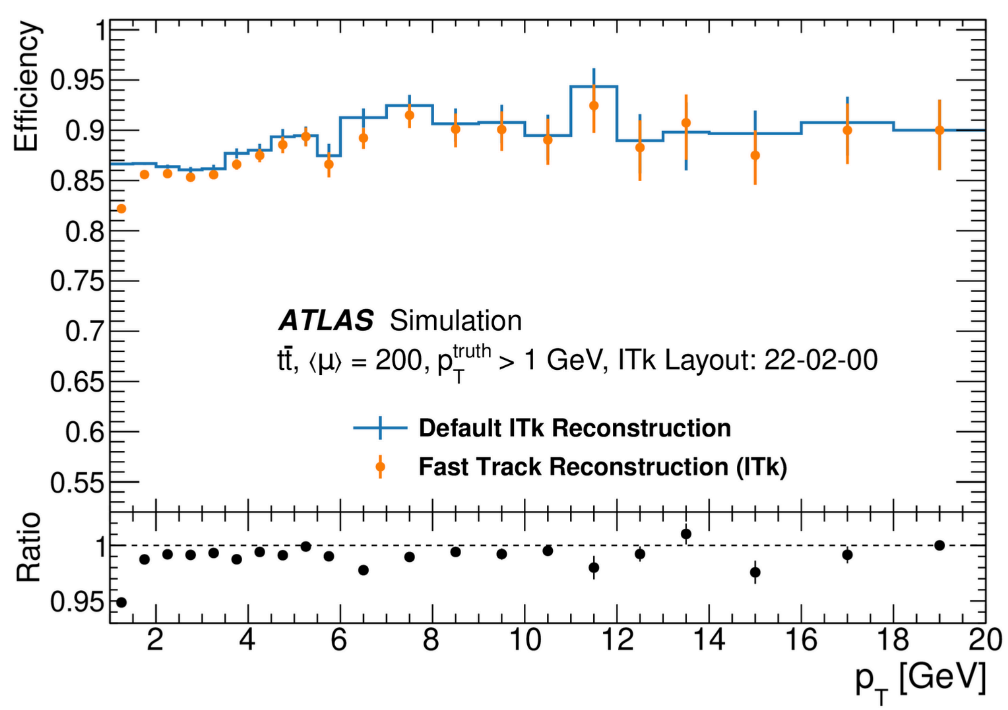
- Revisited the solution for EF Tracking (ATLAS-TDR-029-ADD-1):
- New proposed EF design==> flexible, heterogeneous commercial system consisting of CPU cores and possibly accelerators
- **Develop demonstrators for CPU/GPU/FPGA** with a decision of the technology of the final system in 2025 driven by
  - requirements on the tracking performance (efficiency, resolutions & fake rate)
  - Must come in within budget and satisfying space and power constraints
  - **The final system is unlikely to be one of the exact demonstrators outlined in these slides but working prototypes provide confidence in our ability to build a system that meets all specifications**

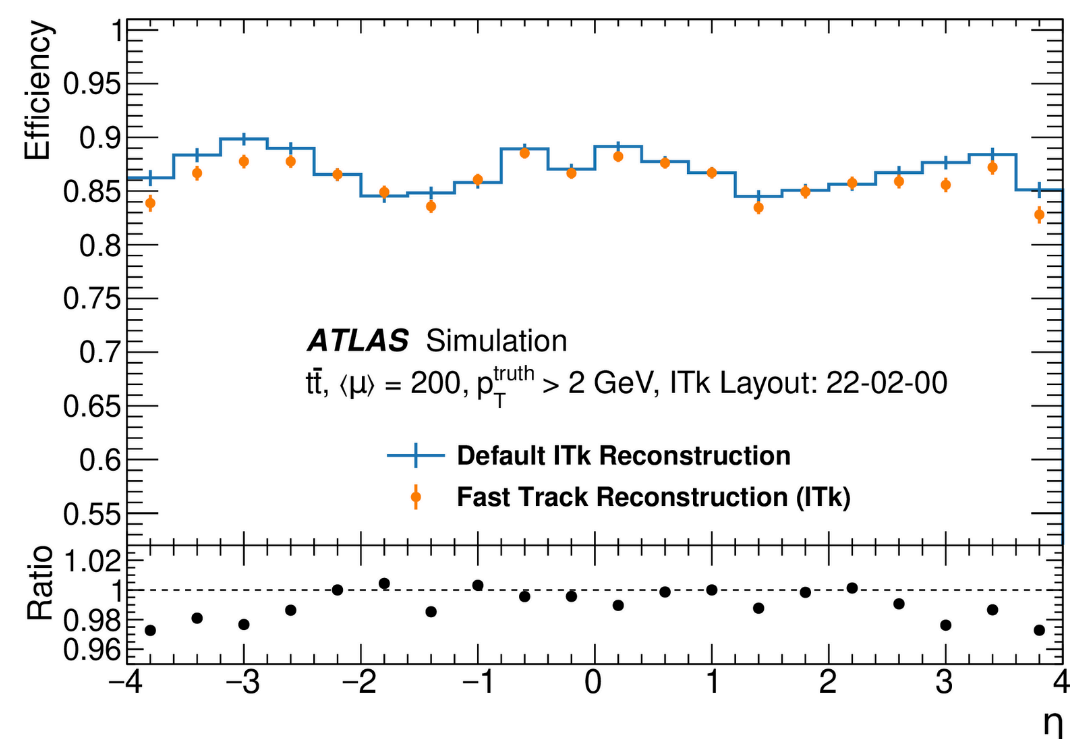
Brookhaven National Laboratory

Fast-tracking prototype for CPU

ATLAS EXPERIMENT

- Based on recent fast ITk reconstruction that makes use of updates to ITk geometry and new performance improvements to **speed up total track reco from earlier estimates by ~x8:**
  - **Remove ambiguity resolution** by implementing tighter track selection upstream, make use of fast Kalman filter (some approximations in material model), seed finding only in pixels (no strips)
  - **Regional tracking for η-φ coverage of 5%**, corresponding to 15% of ITk detector elements
    - Raise p_T threshold to 2 GeV for region tracking (800 MeV in forward) for regional tracking



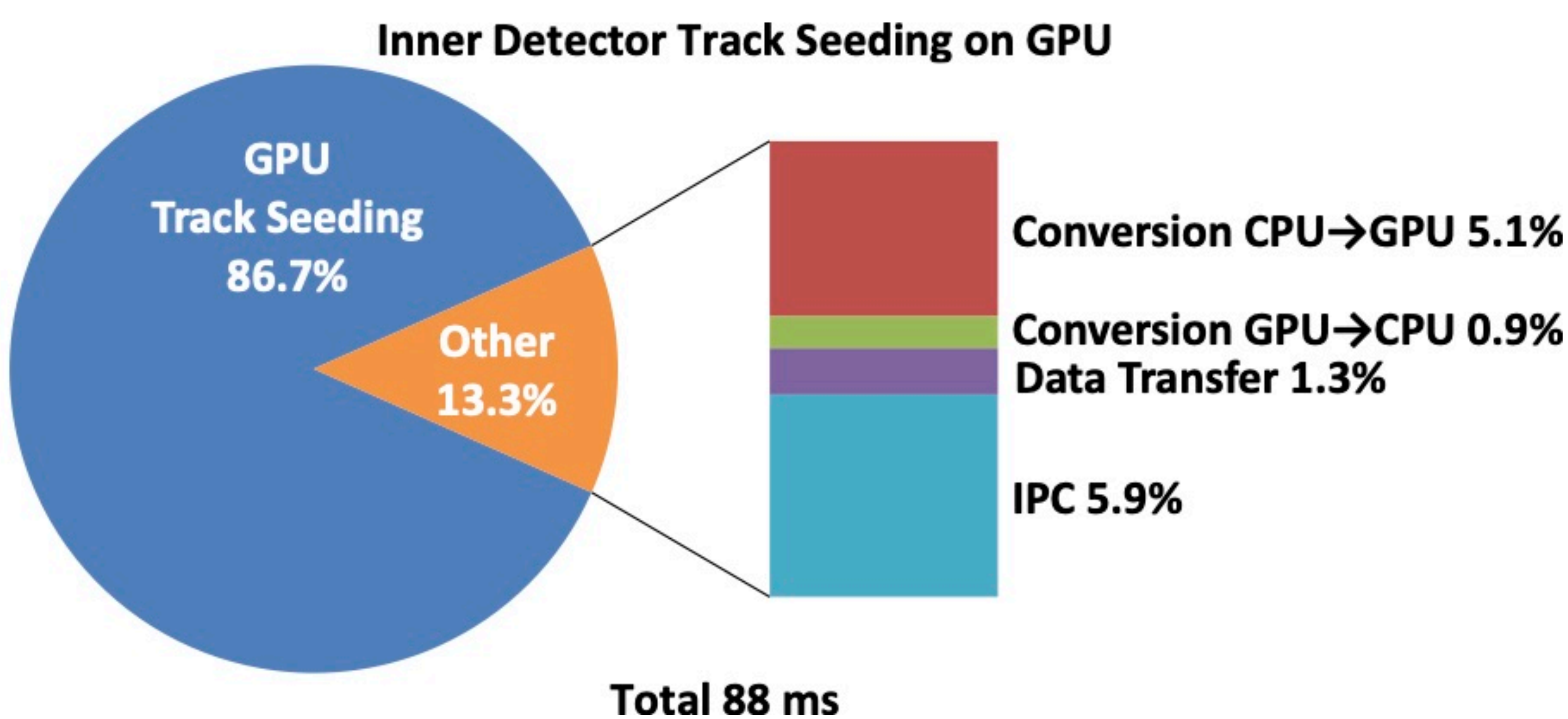


Brookhaven National Laboratory

GPU-based demonstrator

- Provided a **summary of GPU results** in TDAQ TDR that demonstrate the potential of GPUs
  - Uses current ID system and μ=46 samples
- The most computationally intensive data preparation and track-seeding stages
- Overheads for data conversion, communication between processes and not having every stage moved to GPU limits potential gains.
- TDR found that using GPUs would provide the same cost/benefit as adding more CPUs, but this is already a **demonstration of feasibility**

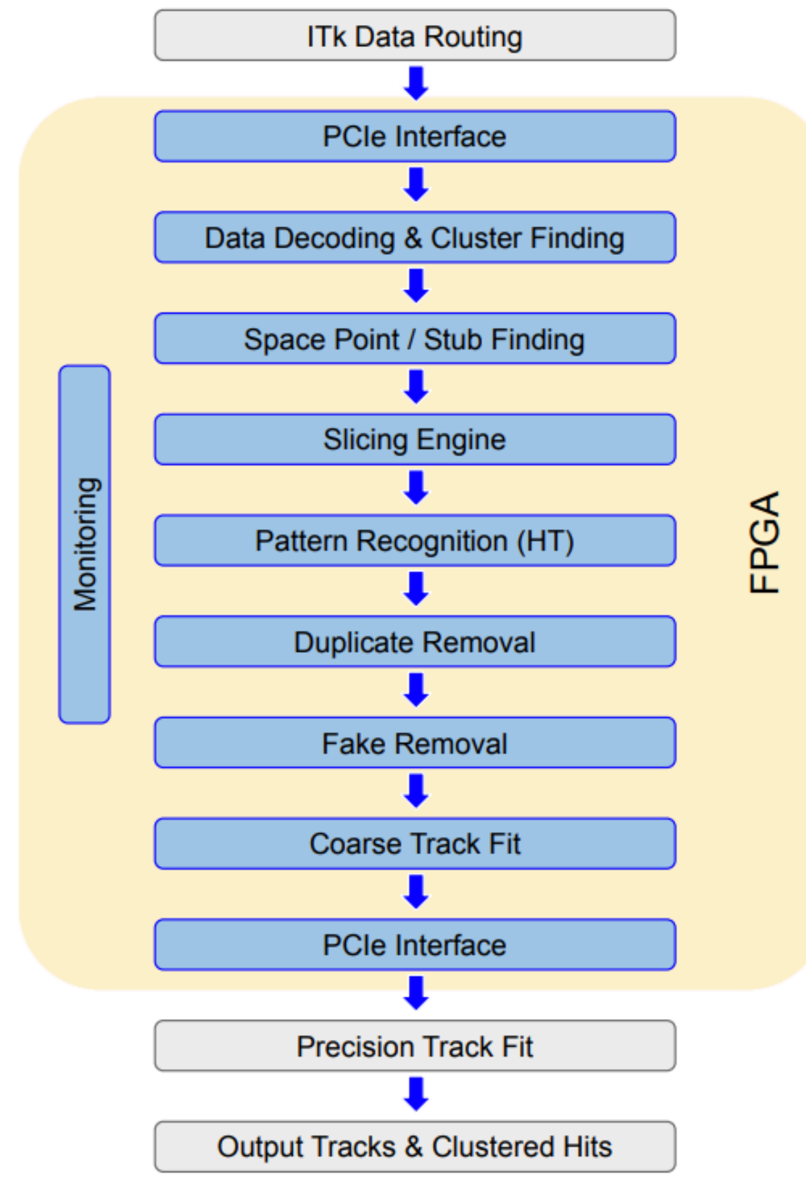
Minimizing data format conversions critical

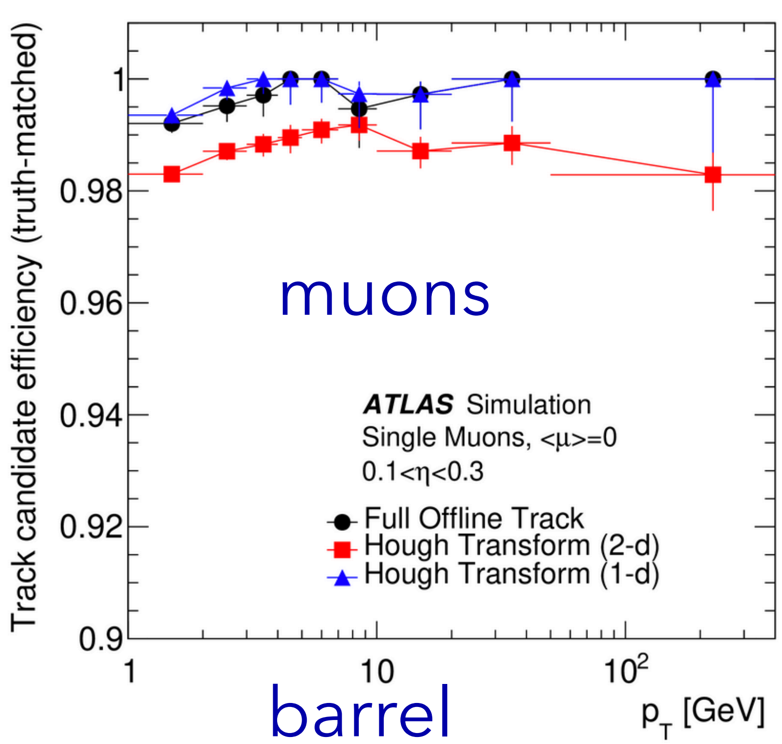


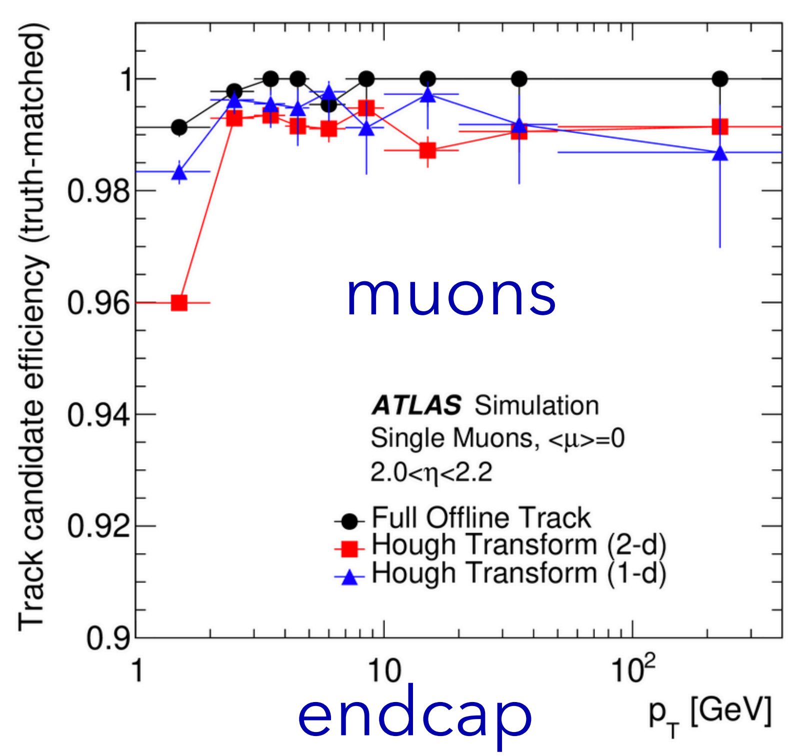
FPGA-based demonstrator

- Results focused on Hough Transforms inside FPGAs (as one example)
- Full track extrapolation to all layers inside FPGA, final CPU-based precision fit for final rejection (but one of many options)
- Target FPGA wherever possible Xilinx Alveo U250
- Neural Network to reject fake+duplicate tracks from Hough Transform

Talk by Kazuki Todome on Hough Transform







Brookhaven National Laboratory

Summary

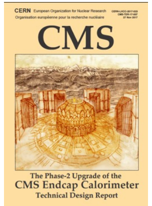
- Ongoing redesign of the EF Tracking for Phase-II upgrade of ATLAS
- Commodity systems based on CPUs/GPUs/FPGAs identified as viable solutions in the amendment of the TDAQ technical design report
- Fast tracking on CPUs demonstrated as a possible solution matching the requirements at the time of the TDR amendment
  - A speedup by factor of 8 achieved in the prototype
- Initial demonstrators using accelerators exist
  - Ongoing work on demonstrators of these (heterogeneous) systems will lead to decision on technology of the final system in 2025
  - Demonstrate delivery of the performance requirements in terms of track reconstruction performance, cost and power consumption

The High-Level Trigger for the CMS Phase-2 Upgrade

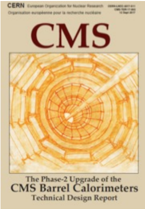
Thiago R. F. P. Tomei

SPRACE-Unesp

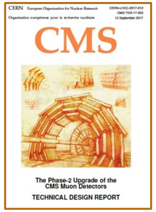
CMS Phase-2 Upgrade Overview



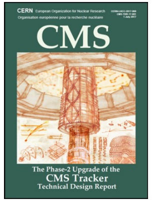
Endcap Calorimeter
- 3D showers + precise timing
- Si, Scint+SiPM in Pb/W-SS



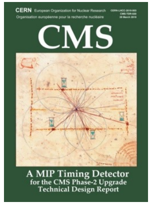
Barrel Calorimeters
- ECAL readout at 40 MHz w/ precise timing at 30 GeV
- ECAL/HCAL new back-end boards



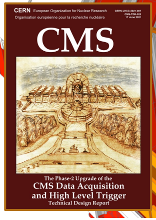
Muon Systems
- DT/CSC new FE/BE readout
- RPC back-end electronics
- New GEM/RPC $1.6 < \eta < 2.4$
- Extended to $\eta \simeq 3$

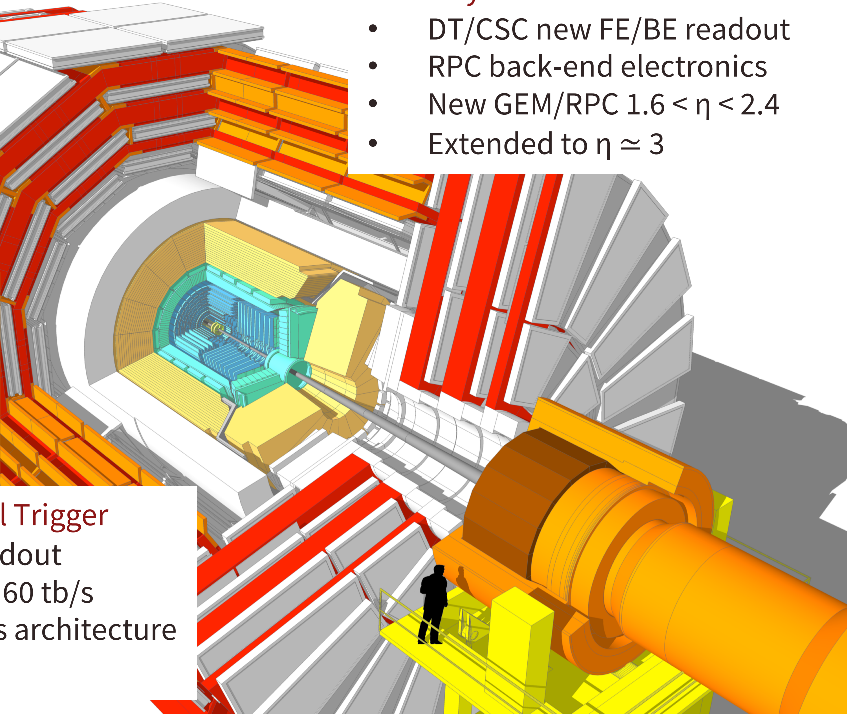


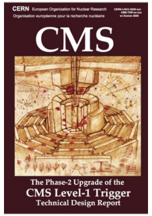
Tracker
- Si-Strip/Pixels increased granularity
- Tracking in L1-Trigger
- Extended coverage to $\eta \simeq 3.8$



MIP Timing Detector
- Precision timing with:
- - Barrel layer: Crystals + SiPMs
- - Endcap layer: Low Gain Avalanche Diodes







L1-Trigger
- Tracks in L1-Trigger at 40 MHz
- PFlow selection
- 750 kHz L1

DAQ and High-Level Trigger
- Full optical readout
- Event network 60 tb/s
- Heterogeneous architecture
- 7.5 kHz HLT

Heterogeneous Computing

Ubiquitous solution for CMS computing needs by 2027

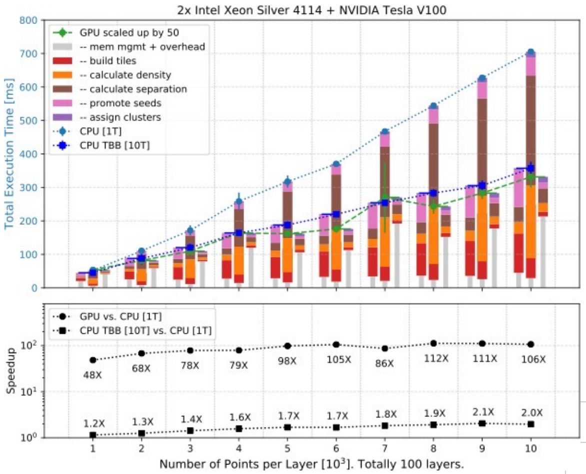
- Heterogeneous HLT farm already starting from Run-3.

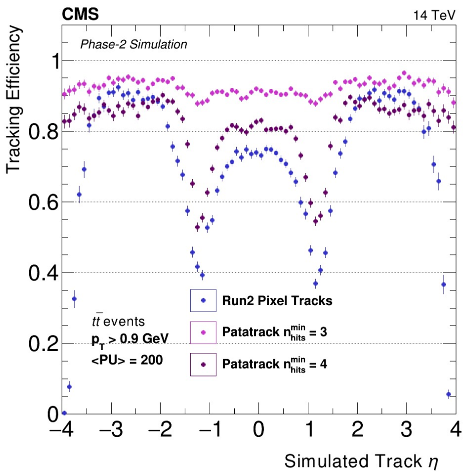
Phase-2 heterogeneous HLT

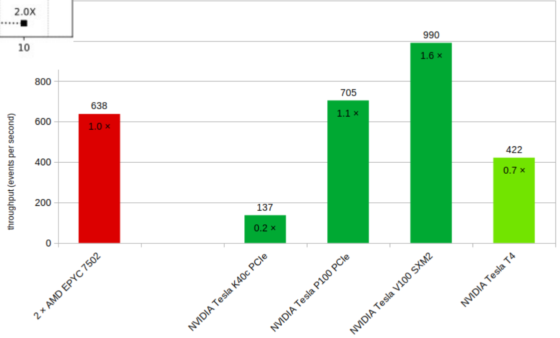
- Under development: HGCAL local reconstruction, Patatrack pixel reconstruction.

Effective farm cost

- 0.70 CHF/HS06 in 2028 – 50% code ported
- 0.22 CHF/HS06 in 2032 – 80% code ported







Figs 10.5, 12.11, 12.12

Offline data processing and analysis at LHCb in the 2020s

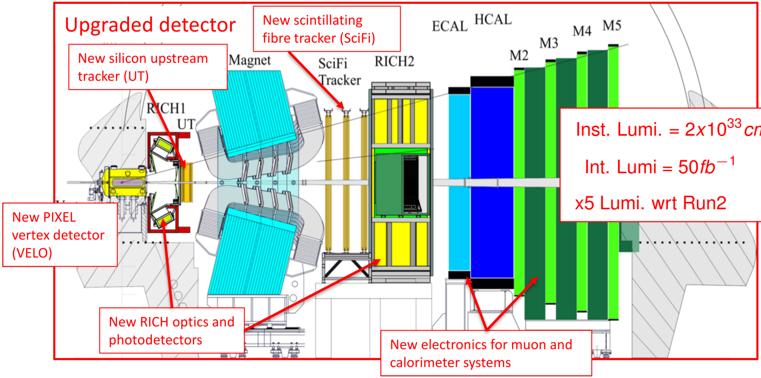
International Conference of High Energy Physics 2022

Davide Fazzini
on behalf of the LHCb Collaboration

July 6-13 2022, Bologna, Italy

The upgraded LHCb experiment

- LHCb is a single-arm forward spectrometer redesigned for Run3:
  - about 95% of the sub-detectors is completely new
  - high-precision tracking and & vertexing
  - excellent PID & hadron separation
- Physics programme in practice far more general
  - Electroweak, Exotica, LLPs, Fixed-target, heavy ions
- **Think of it as a more general purpose detector!**



Inst. Lumi. = $2x10^{33} cm^{-2} s^{-1}$

Int. Lumi = $50 fb^{-1}$

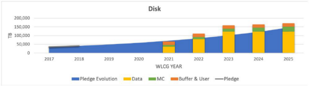
x5 Lumi. wrt Run2

WP4: R&D Innovative analysis techniques

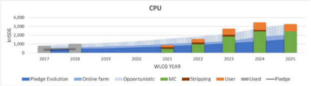
- Focus on exploitation of new analysis facilities with heterogeneous computing resources (GPU/CPU/FPGA)

- Worldwide LHC Computing Grid consists of $\sim$ 1M CPU cores over 170 sites
  - main LHCb activities based on CPUs
  - supporting High Performance Computing centers providing large GPU resources
  - potential to utilise LHCb's HLT1 GPU farm during detector downtime
- Development to run LHCb payloads on GPUs
  - use advanced algorithms, as Generative Adversarial Networks (GANs), to train models describing LHCb sub-detector $\Longrightarrow$ GPUs speed up GAN training, *Ultra-fast-simulation*
  - users using GPUs for complex amplitude analysis models with large statistics

- In Run3 LHCb will produce $\sim$ 15PB of data on disk per year



- Simulation will require 90% of total offline CPU resources



LHCb-TDR-18

- First investigation into use of Quantum Machine Learning for jet tagging (Davide Zuliani's talk "Quantum Machine Learning for b-jet identification")

Machine learning

- New fast simulation method
- Novel graph-based tracking algorithm

Generative Models for Fast Simulation of Electromagnetic and Hadronic Showers in Highly Granular Calorimeters

Sebastian Bieringer, Erik Buhmann, Sascha Diefenbacher, Engin Eren, Frank Gaede, Gregor Kasieczka, William Korcari, Katja Krüger, **Peter McKeown**[1], Lennart Rustige, Imahn Shekhzadeh

[1] **Deutsches Elektronen-Synchrotron**

08.07.2022

peter.mckeown@desy.de

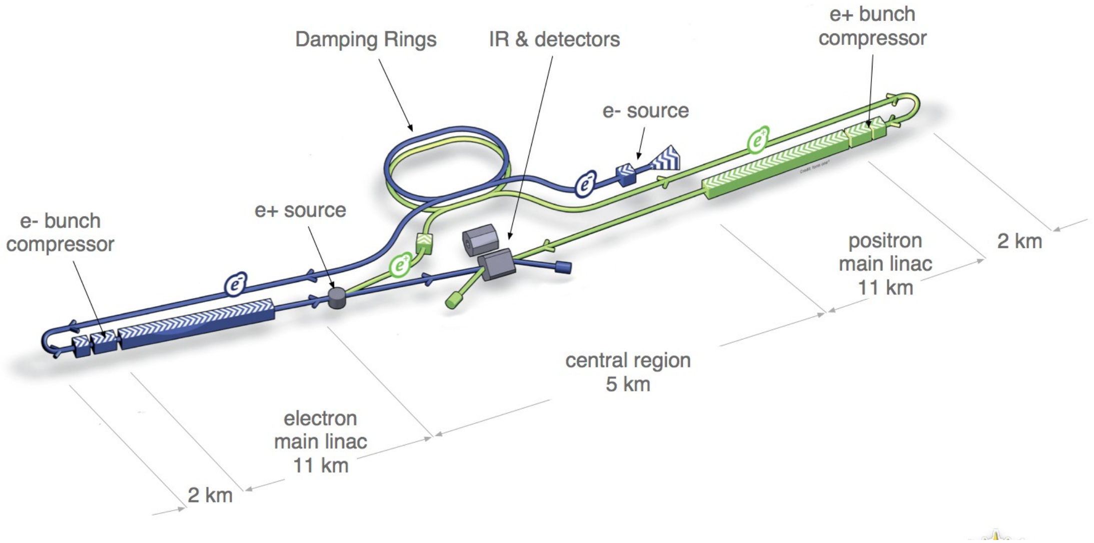
The ILD Concept

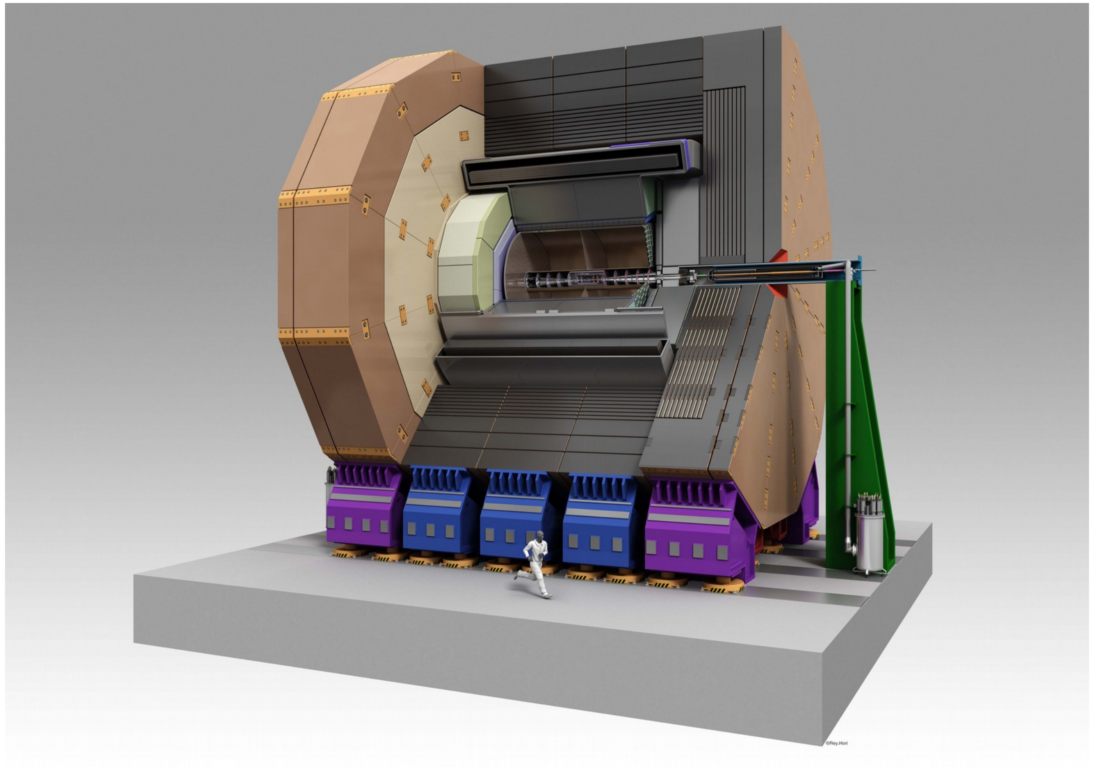
- Context: Future Higgs Factories

- Case Study: International Large Detector (**ILD**) concept for the International Linear Collider (ILC)

- Optimized for Particle Flow
  - Reconstruct each individual particle in subdetector
  - Obtain optimal detector resolution

- High granularity calorimeters:
  - Sampling calorimeters
  - **SiW Ecal**: 30 layers, 5x5 mm^2, 2 sampling fractions
  - **FeSci Hcal**: 48 layers, 3x3 cm^2





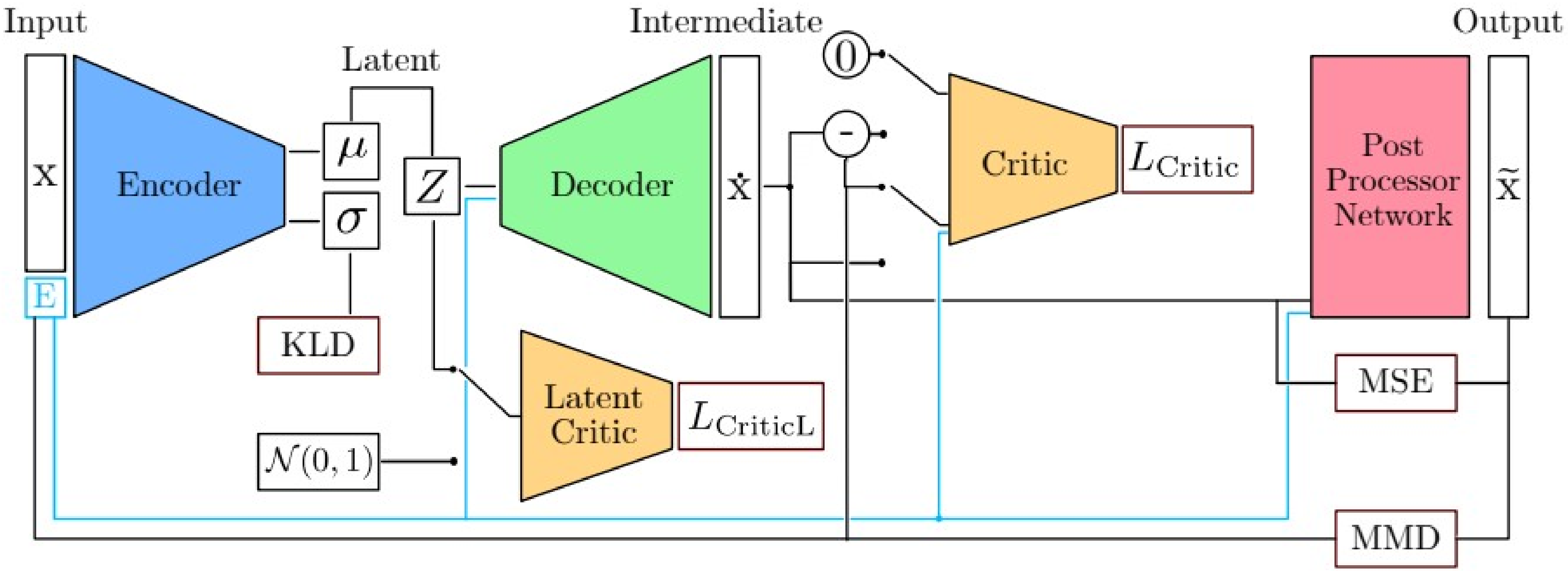
Architectures: BIB-AE

Bounded-Information Bottleneck Autoencoder (BIB-AE)

Voloshynovskiy et. al: Information bottleneck through variational glasses, arXiv:1912.00830 (2019)

Buhmann et. al: **Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed**, CSBS 5, 13 (2021)

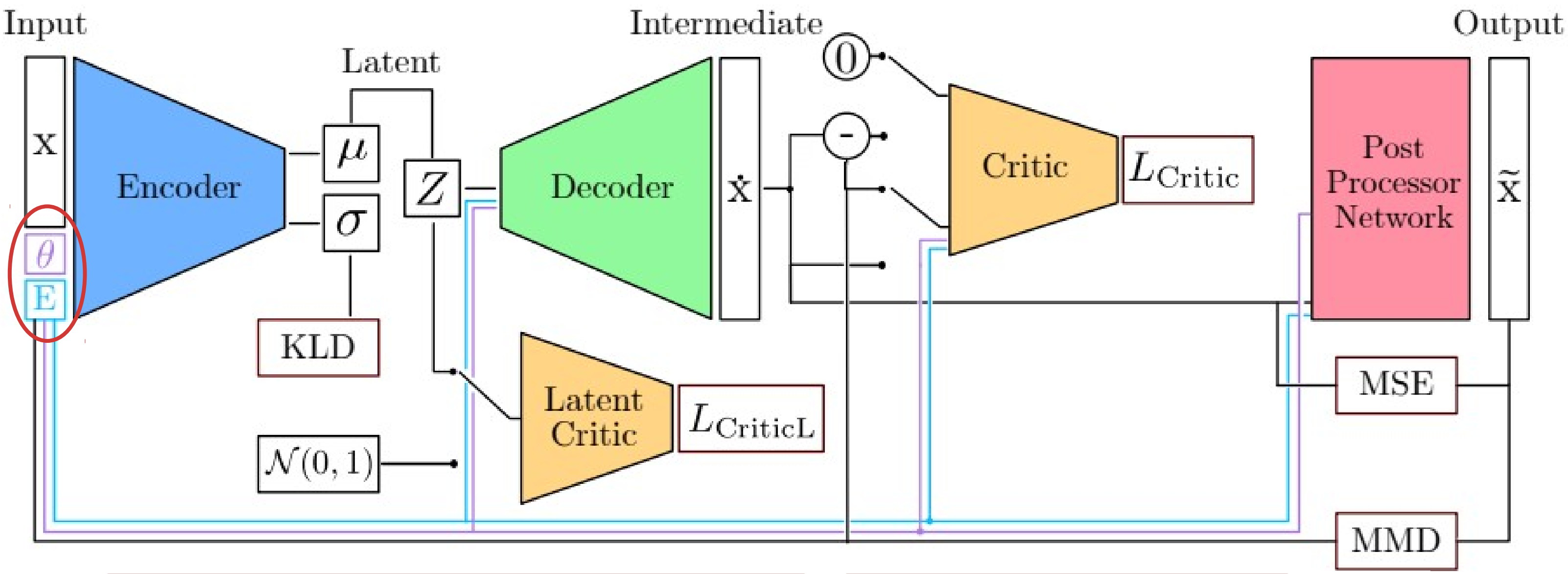
- **Unifies** features of both **GANs** and **VAEs**
- **Post-Processor** network: Improve per-pixel energies; second training
- Multi-dimensional KDE sampling: better modeling of latent space



$L_{BIB-AE} = KLD + L_{CriticL} + L_{Critic}$ $\quad$ $L_{Post} = MMD + MSE$

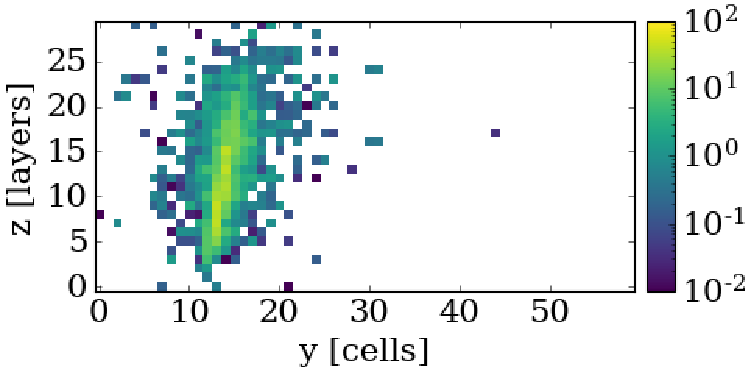
Angular and Energy conditioning- Training data

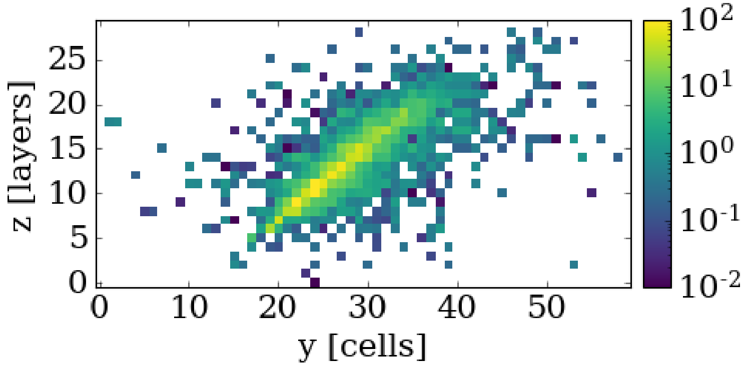
- 500,000 **photons** with fixed incident point
- Vary **energy**: **10-100 GeV**
- Vary **polar angle** in one direction: **90°-30°**
- Project to regular grid
  - Shape (30,60,30) (x,y,z)

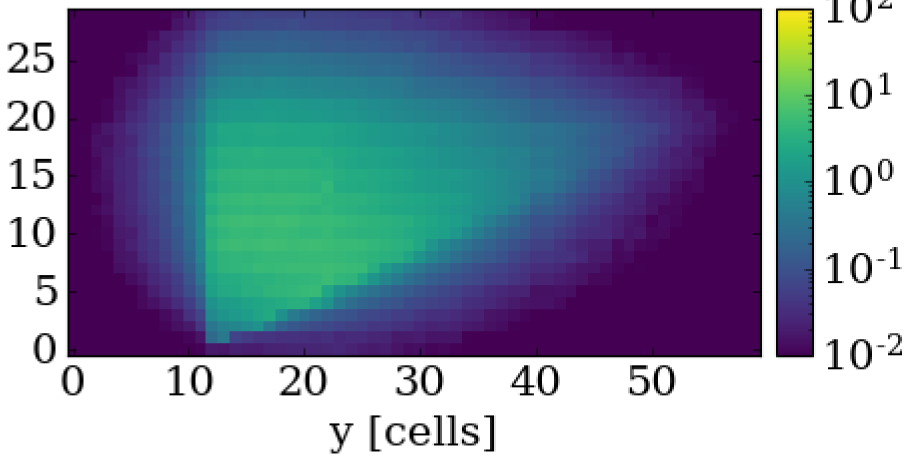


$$L_{\mathrm{BIB-AE}} = \mathrm{KLD} + L_{\mathrm{CriticL}} + L_{\mathrm{Critic}}$$

$$L_{\mathrm{Post}} = \mathrm{MMD} + \mathrm{MSE}$$



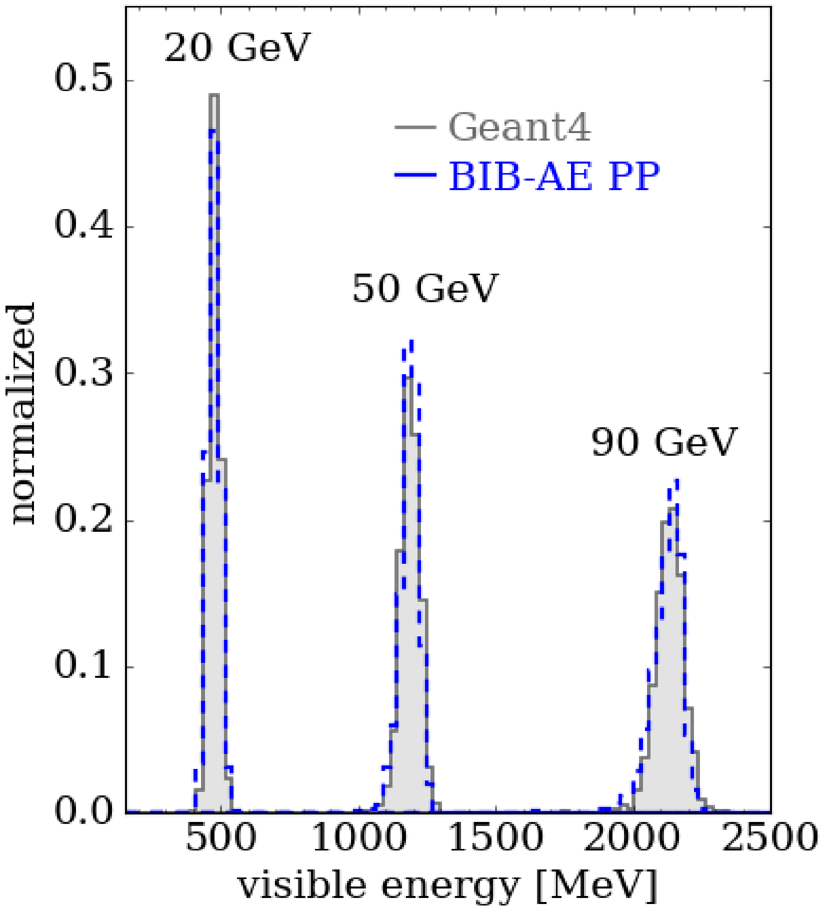




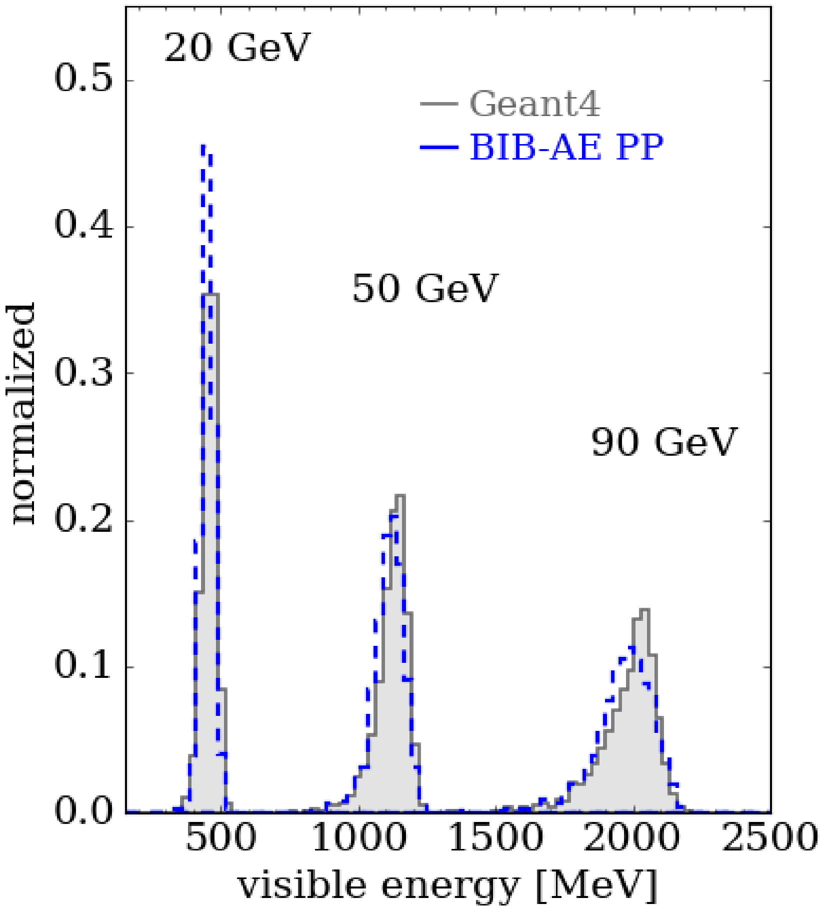
Results: Visible Energy Sum

- Visible energy is nicely described for different incident angles and energies

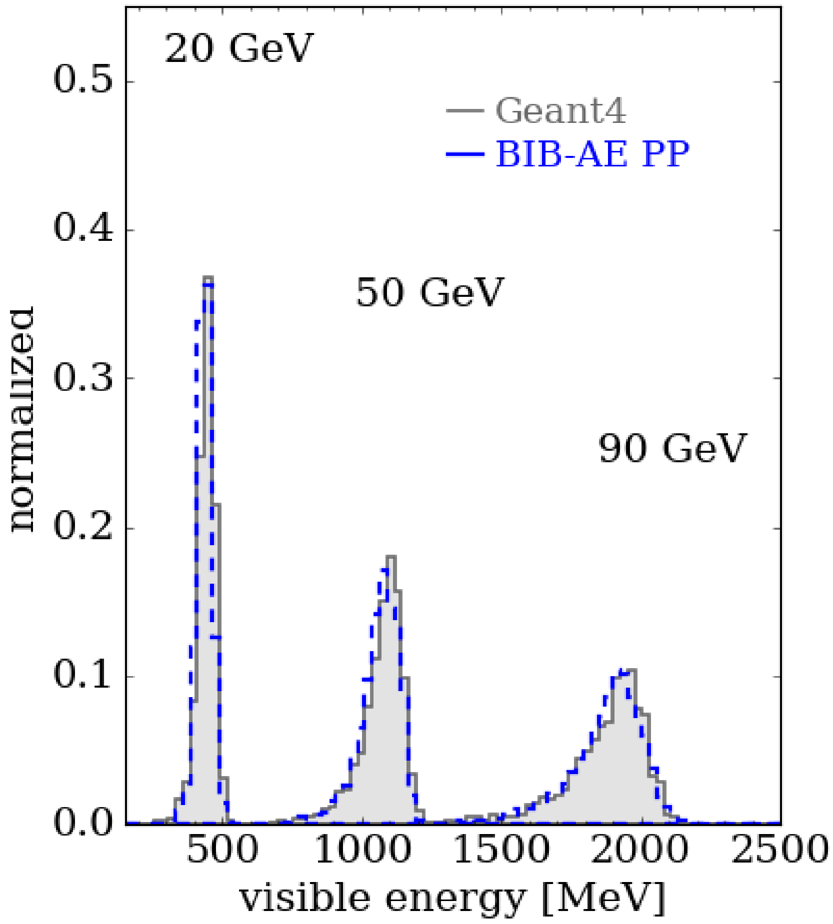
40 degree Photons



60 degree Photons

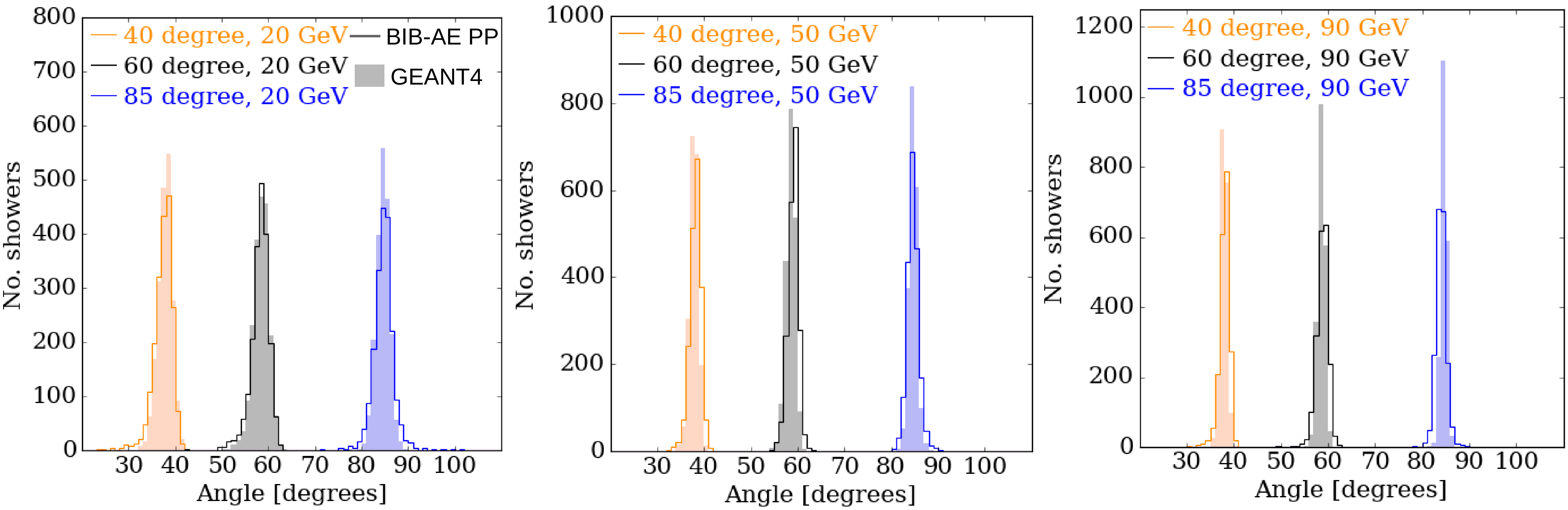


85 degree Photons



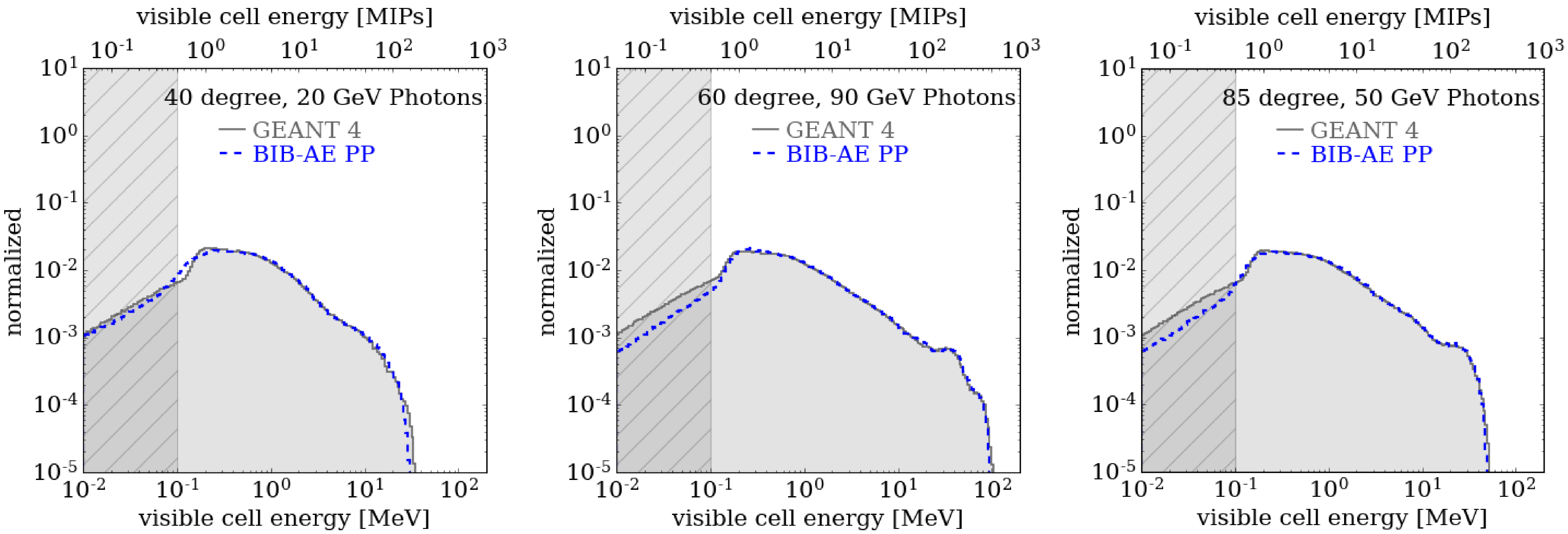
Results: Angular Reconstruction Distributions

- Angular distributions agree well for given incident energies after reconstruction with a PCA



Results: Cell Energy Spectrum

- Post Processor Network retains its ability to correctly describe the cell energy distribution



Conclusion

Achieved

- Generative models hold promise for **fast** simulation of calorimeter showers with **high fidelity**
- Demonstrated high fidelity simulation of **hadronic** showers with generative models
- Demonstrated high fidelity simulation of **photon** showers with **angular and energy conditioning**
- Initial investigation into generative model performance after **reconstruction**

Next Steps

Hadron Shower Simulation

- Simulation of hadronic showers combining **ECAL and HCAL**

Photon Shower Simulation

- Benchmark performance after **reconstruction** and **timing**
- Develop strategy for dealing with **arbitrary incident positions**

ICHEP 2022

Applying the Exa.TrkX pipeline to the OpenDataDetector with ACTS

08.07.2022

Benjamin Huth
Universität Regensburg

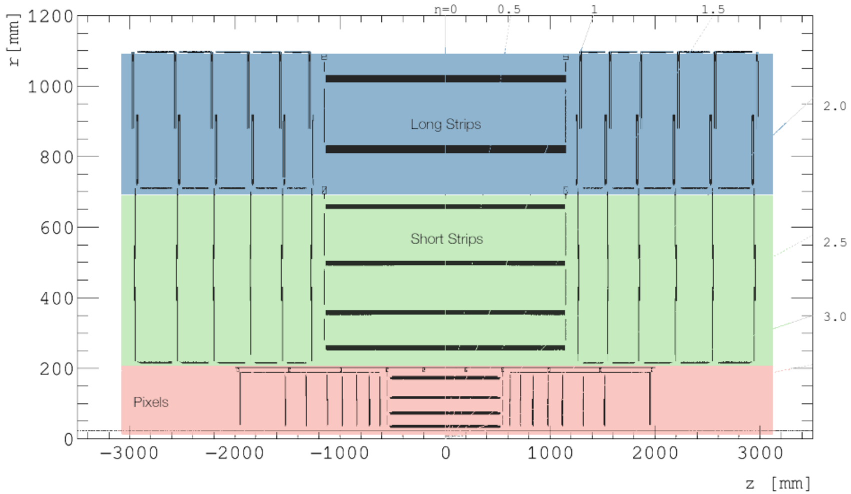
Andreas Salzburger (CERN),
Lukas Heinrich (TU München),
Tilo Wettig (Universität Regensburg),
Exa.TrkX authors: Alina Lazar*, Xiangyang Ju*,
Daniel Murnane*, Paolo Calafiura*
**Lawrence Berkeley National*

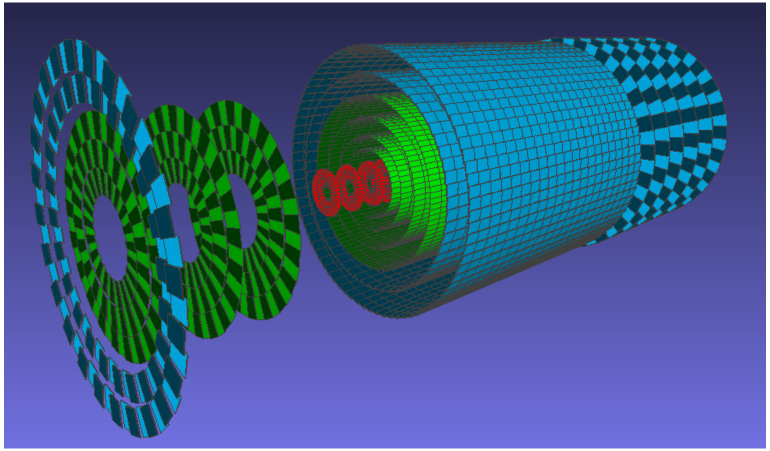
ACTS

- ACTS is an experiment-independent toolkit in modern C++ for charged particle tracking.
  - State-of-the-art implementations of standard algorithms and R&D testbed
  - Used in several experiments: ATLAS, sPHENIX, FASER
  - For more infos see here and on github.com/acts-project/acts

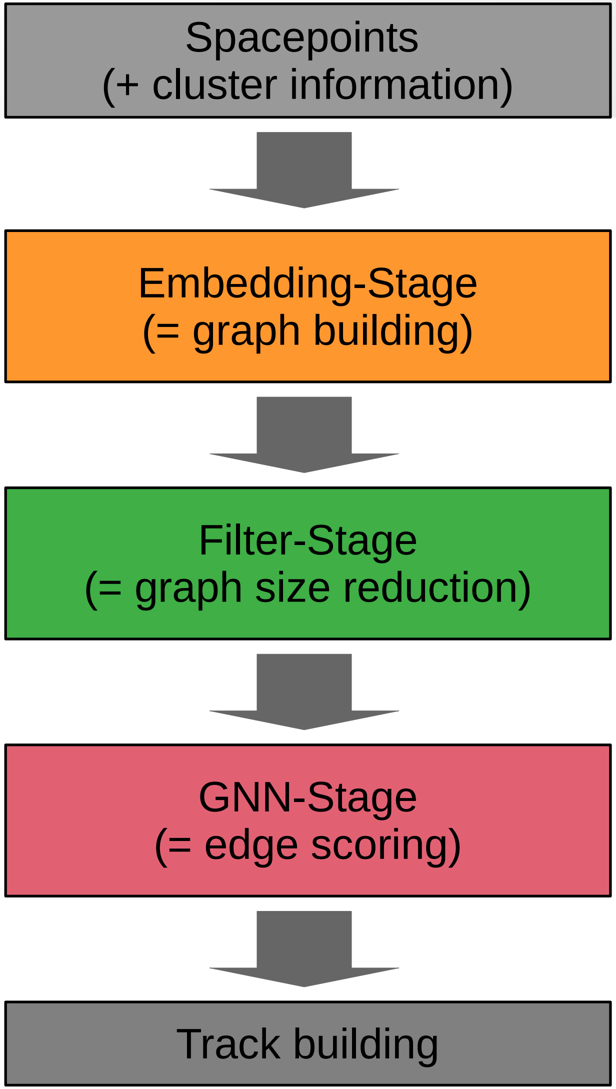
The OpenDataDetector

- The OpenDataDetector is a purely virtual, but realistic detector for testing and R&D purposes
  - Shipped with and interfaces to ACTS
- Changes wrt. the TrackML detector:
  - more precise material description
  - capability of full-simulation
  - based on DD4hep
- For more details see [here](#) (Paul Gessinger, ACAT 2021)



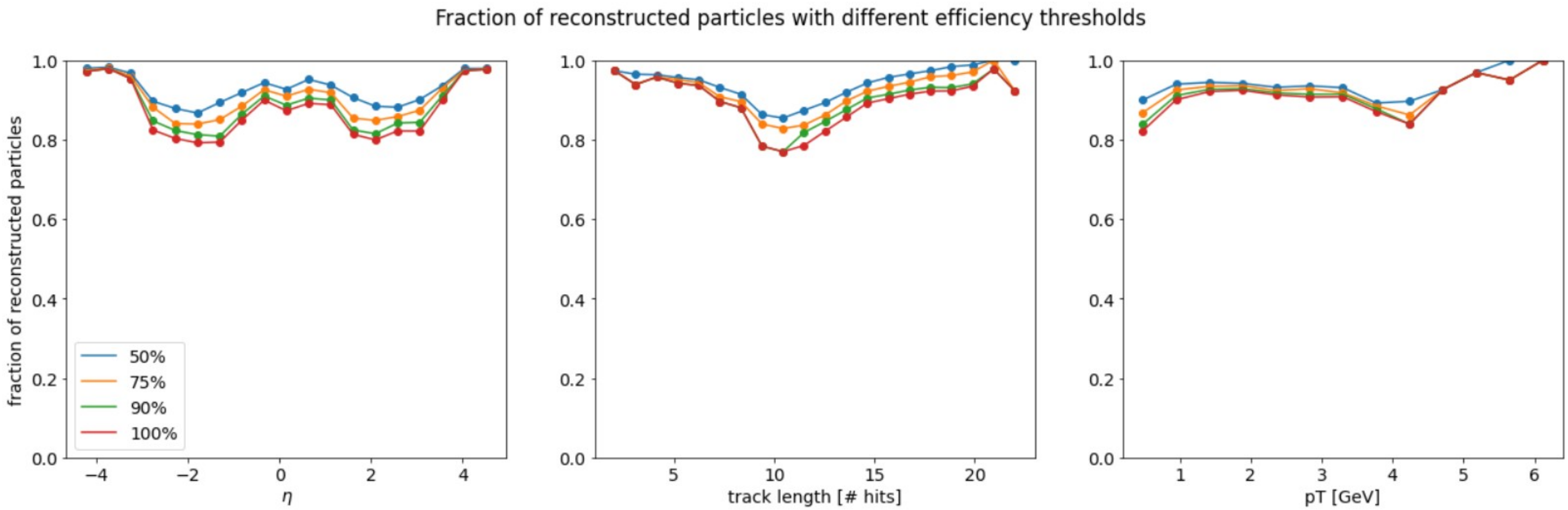


The Exa.TrkX pipeline



- Multi-stage machine-learning pipeline for **track finding**
  - Event as a graph (nodes = hits, edges = potential track segments)
  - Use ML (especially GNNs) to find edges that correspond to track segments
- Track fitting is performed by ACTS
- See e.g. here (Alina Lazar, ACAT 2021)

Inference results in ACTS



- Could be improved by:
  - Adding cluster-information to training data
  - More events / epochs for training
- Results shown for truth hit input

Particle-based **efficiency**:
eff = #hits-in-best-track / #hits-of-particle

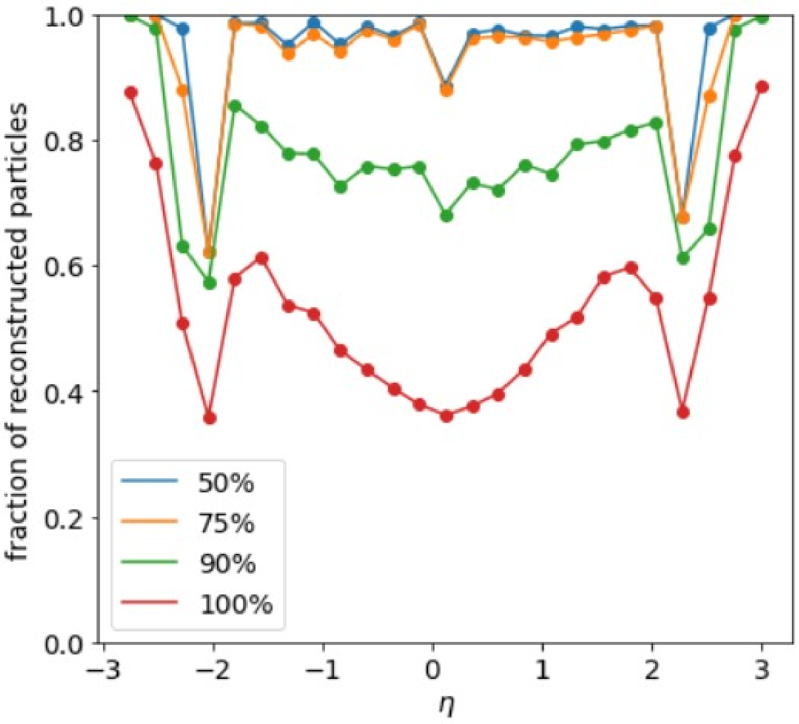
Comparison with CKF*

- CKF relies on
  - correct error and material description
  - tuned parameters for initialization & combinatorial search
- For this work: Auto-Tuning of 7 parameter with *Optuna***
- Applied particle pre-selection for both CKF and Exa.TrkX
  - Cut **pT < 500 MeV**
  - Cut **|eta| < 3**
  - Cut **r < 2 mm** (distance from origin in transverse plane)

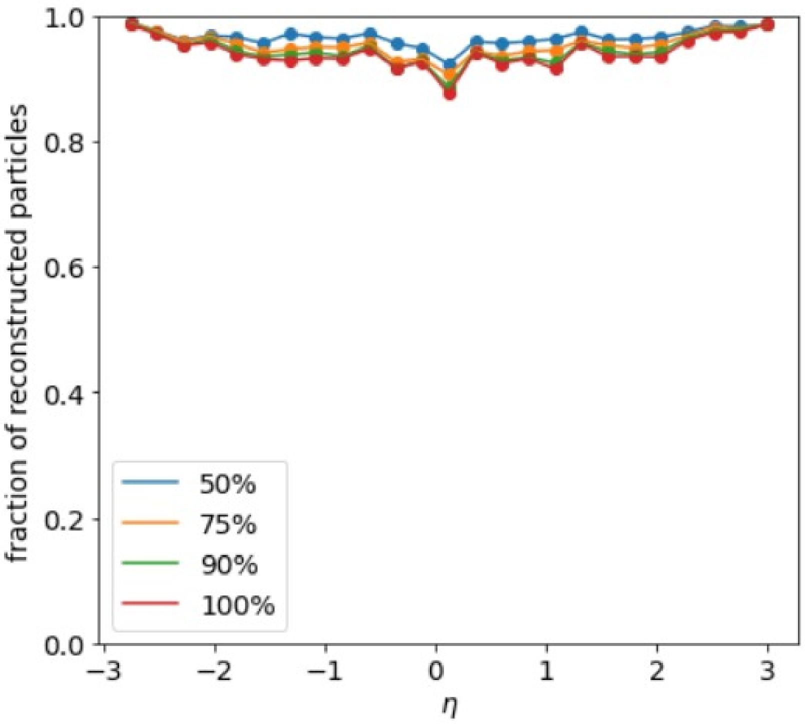
* Combinatorial Kalman Filter
** See work on automatic optimization for the CKF by Rocky Bala et al here

Reconstruction: true hit positions

CKF



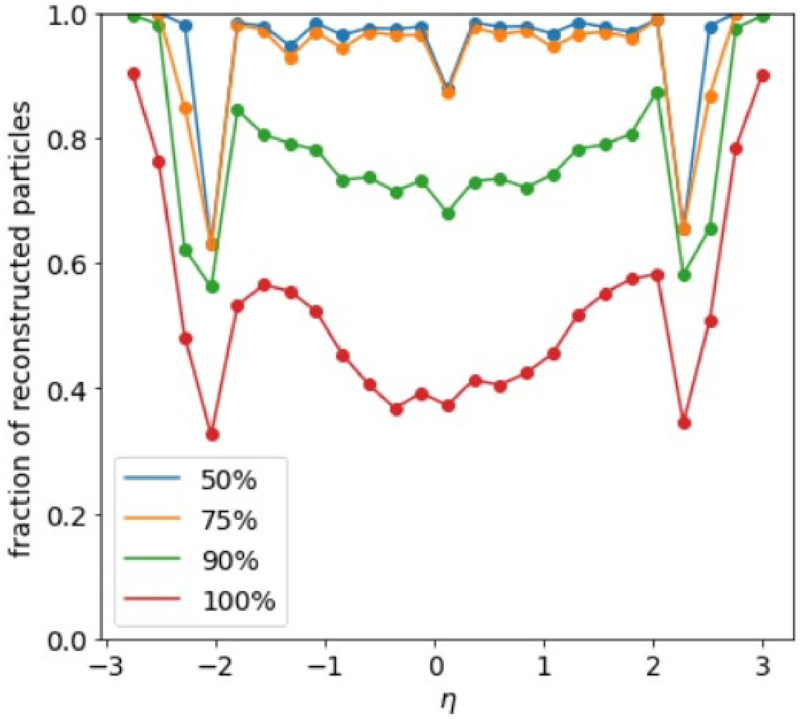
Exa.TrkX



- Error description and parametrization not optimized
- Exa.TrkX profits from applied cuts

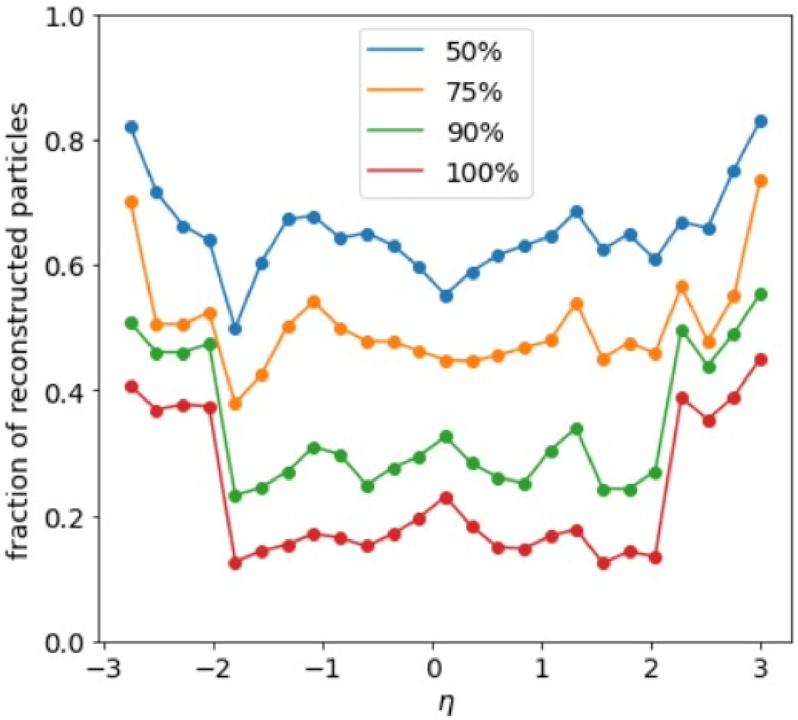
Reconstruction: smeared hits

CKF



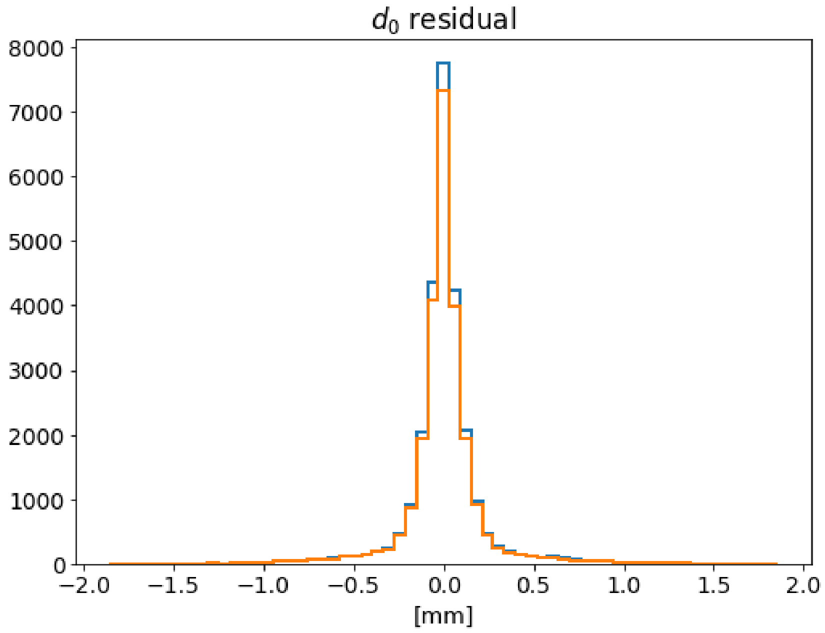
- Almost no sensitivity to hit position changes

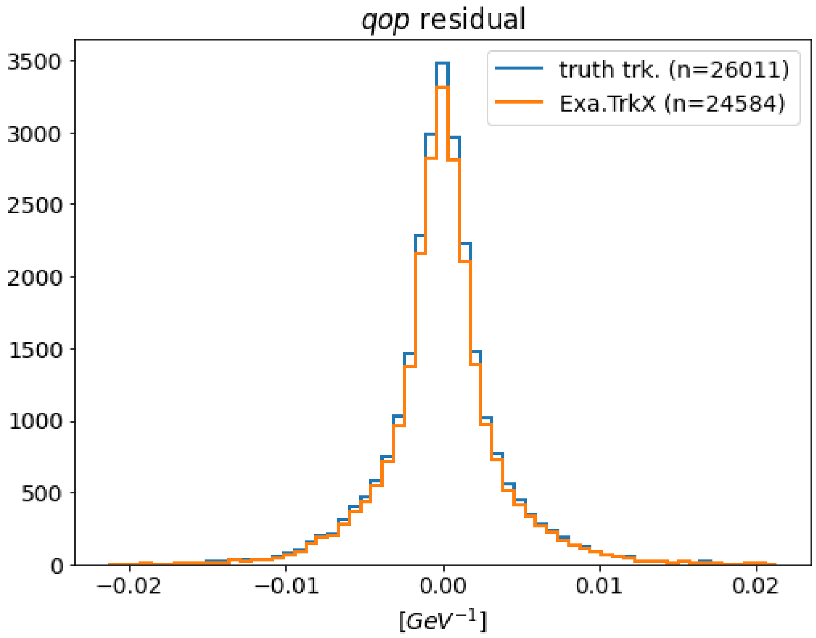
Exa.TrkX



- Influence of training with truth positions
  - very sensitive to hit positions

Parameter Fit





- New in this project: Combination of **Exa.TrkX** and **Kalman-Fitter (KF)**
  - True hit positions as input for Exa.TrkX inference
  - Track parameter resolution of Exa.TrkX pipeline are reaching achievable limit
- CKF not shown here (no ambiguity solution available in ACTS so far)

Conclusion by Heather M. Gray

- Software plays a key role in essentially **every component** of modern HEP experiments
- Within HEP, software been going through a period of **rapid evolution** due to more demanding **experimental requirements** and changing **hardware environment**
- Key features include
  - **Optimization** and **modernization**
  - Movement towards **common software**
  - Increasing diversity of **hardware architectures**
  - Impact of **machine learning**
- This rapid development will need to continue in preparation for **future upgrades** such as the HL-LHC
- For further details, I encourage you to consult the excellent talks from the **parallel sessions**