SAND Data Analysis Status and Prospective

M. Tenti - INFN Bologna,
DUNE Collaboration Meeting
26/01/2022

INFN
Istituto Nazionale di Fisica Nucleare

Status

- Full simulations w/ official beam: (FLUKA & GENIE + GEANT4)
- Fast Reconstruction
  - Beam Monitoring
  - External Background
- Studies w/ partial info from MC truth, in particular:
  - STT digit == smearing of hit (time, position)
  - Particle ID
- Performances:
  - Preliminary Pattern Reco
  - Momentum resolution w/ circular fit
  - neutrino energy resolution
- Physics Analyses: docdb-13262

Next tasks

- Define Calibration Data:
  - How do we produce it?
  - Which format?
  - How do we input in the reconstruction?

- Digitization:
  - Trigger: which time reference (t0)
  - Digit format (input from DAQ?)

- Integrate «Fast Reconstruction»

Next tasks

- Improve Reconstruction:
  - Tracking & Vertexing:
    - STT Kalman Filter
  - EM Shower:
    - ECAL Clustering
  - Particle ID:
    - e- ID in ECAL
    - μ/π separation:
      need for downstream μ catcher?

- EventSummaryBuilder $\rightarrow$ CAF

Final Goals

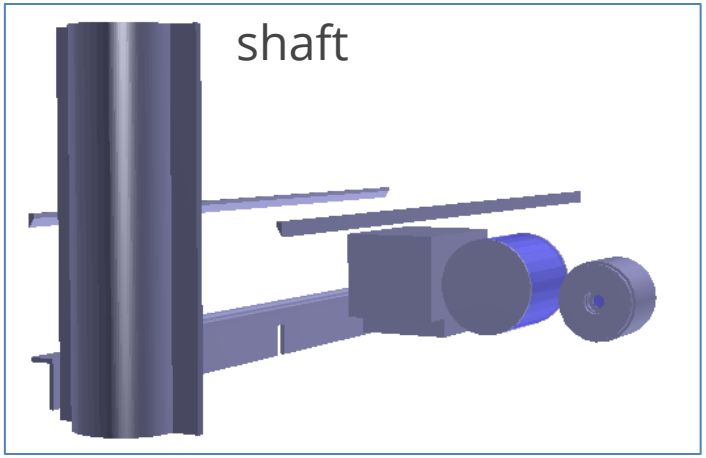
- Full event reconstruction
- Event classification
- Background rejection
- Physics analyses

Requirements

- run on CENTOS7
- complaint with interfaces
- complaint with coding convention
- C++/ROOT (cannot fix a release too early…)

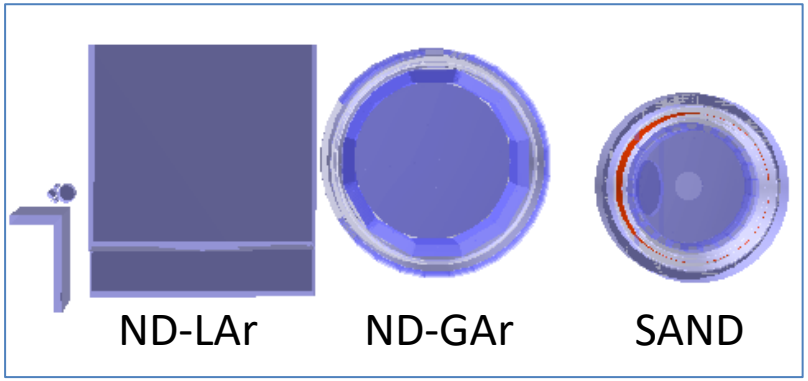
- Questions:
  - High-level requirements from ND-Software?
  - Performance?
  - Functionalities, if any?
    - How to deal with competive algorithms ?
    - How to manage calibrations ?

Repositories



- Baltig group: **dune**

- Geometry successfully merged in
  - **https://github.com/DUNE/dunendggd**



- Digitization/Reconstruction
  - **https://baltig.infn.it/dune/sand-reco**
  - Public read access
  - Exploit continuous integration resources by INFN
  - License?

Repositories

D

dune

Group ID: 631

Subgroups and projects | Shared projects | Archived projects

Search by name | Name

C	**cluster-analysis** The project aims to test and validate the SAND ECAL clustering algorithm.	★ 0	12 seconds ago
G	**grain-physics-case** Studiare la sensitività di SAND a differenti modelli di interazioni di neutrino in LAr con G...	★ 1	3 minutes ago
N	**nuev-generator** development of code, script and macros in order to generate neutrino event in a format...	★ 1	1 minute ago
S	**sand-FLUKA** Codes for FLUKA simulations analysis	★ 0	3 minutes ago
S	**sand-reco** This project aims at developping tools to reconstruct neutrino interaction in the SAND ...	★ 1	4 minutes ago
S	**STTTrackReco**	★ 0	44 seconds ago

Interfaces (Data Model)

INPUT

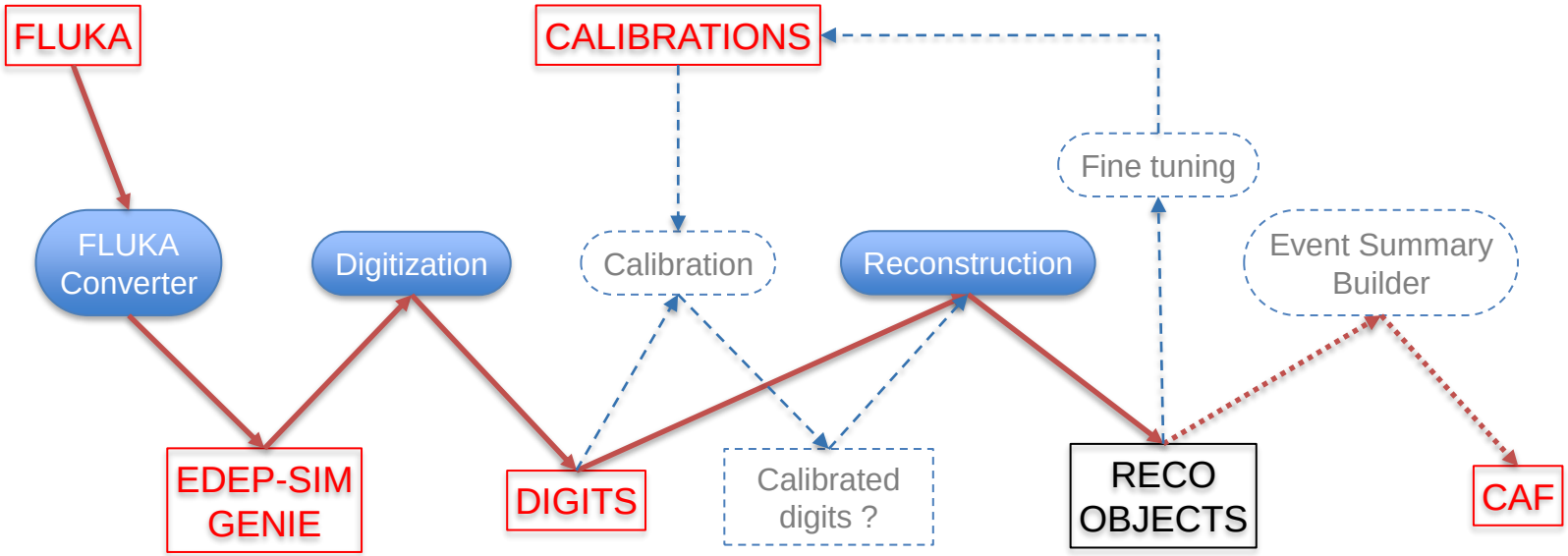
- **[edep-sim]**
  - https://github.com/DUNE/edep-sim#output-tree-format
  - Edep-sim output (cern.ch)
- **[genie] (embedded in the edep-sim data format)**
  - GENIE Physics and User Manual (rl.ac.uk)
- **[FLUKA]**
  - /eos/user/s/salap/DUNE-IT/ntuple.spiega
- **[Digits]**
  - from DAQ
- **[Calibrations]**
  - External interface

The set of external specifications conditioning this SW project.

OUTPUT

- **[CAF]**
  - https://cdcvs.fnal.gov/redmine/projects/dune-neardet-design/wiki/_CAF_ntuple_format_

SW Design (and Data Models)

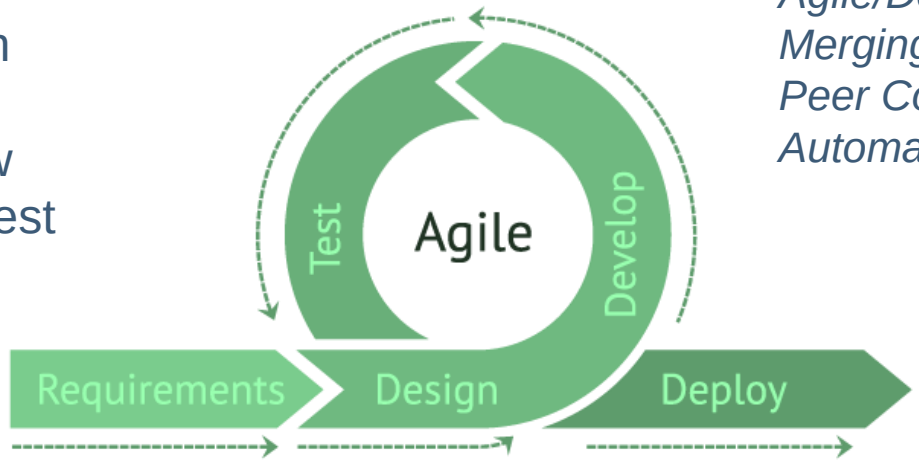


Digits & Reco Data Model: [Data Model · Wiki · dune / kloe-simu · GitLab (infn.it)]

Coding and Development Workflow

- Language: C++11
- Code Format:
  - Based on Google C++ Style Guide
  - Proposal: *clang-format -style="{BasedOnStyle: Google, BreakBeforeBraces: Linux, DerivePointerAlignment: false}"*
- Project layout: pitchfork
- Development Workflow:

1. Create Issue
2. Open Branch
3. Develop
4. Test / Review
5. Merge Request
6. Release

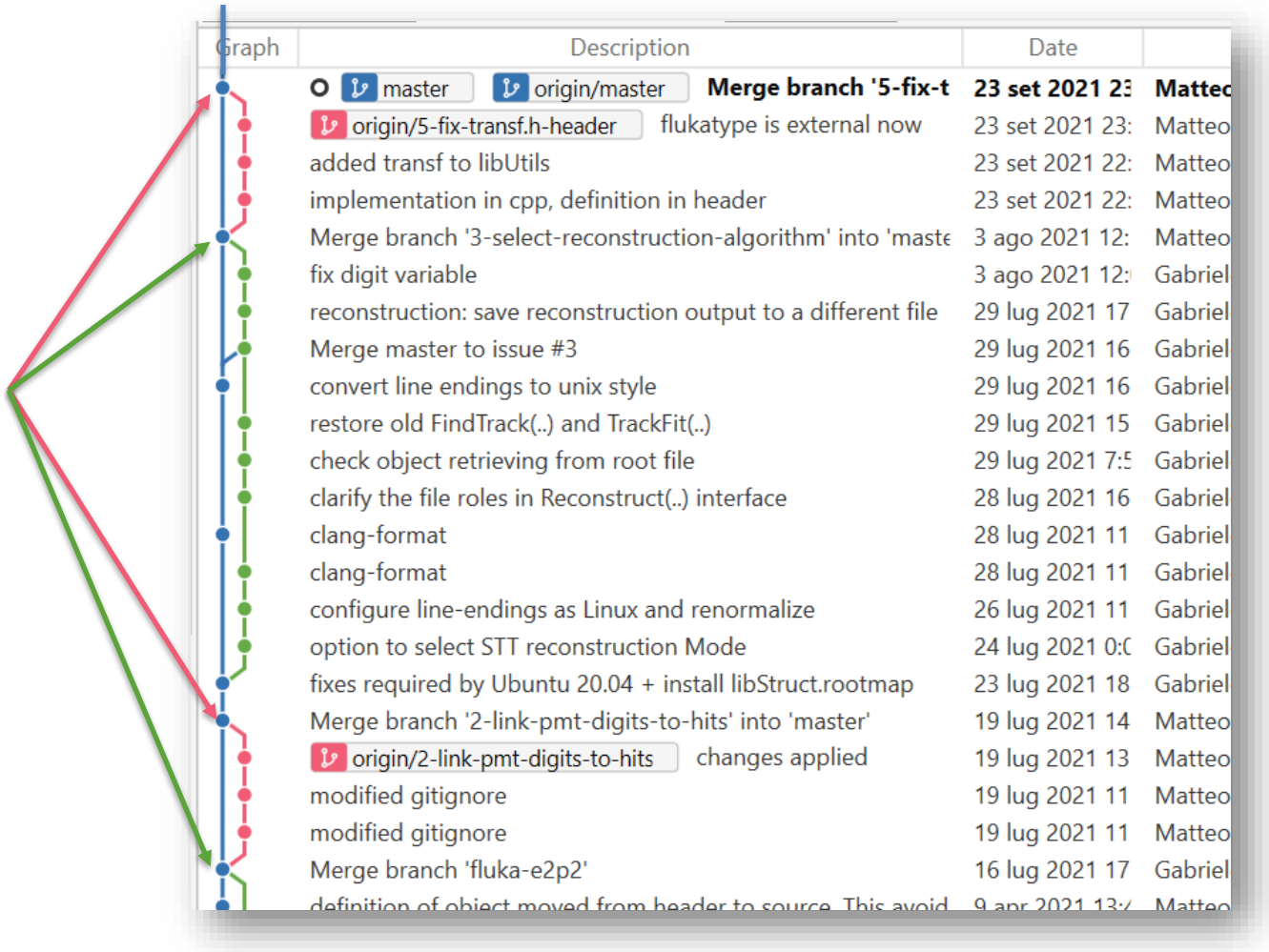


Exploits:
Agile/DevOps,
Merging by Pull Request
Peer Code Review,
Automated Tests,

Development workflow

Master branch

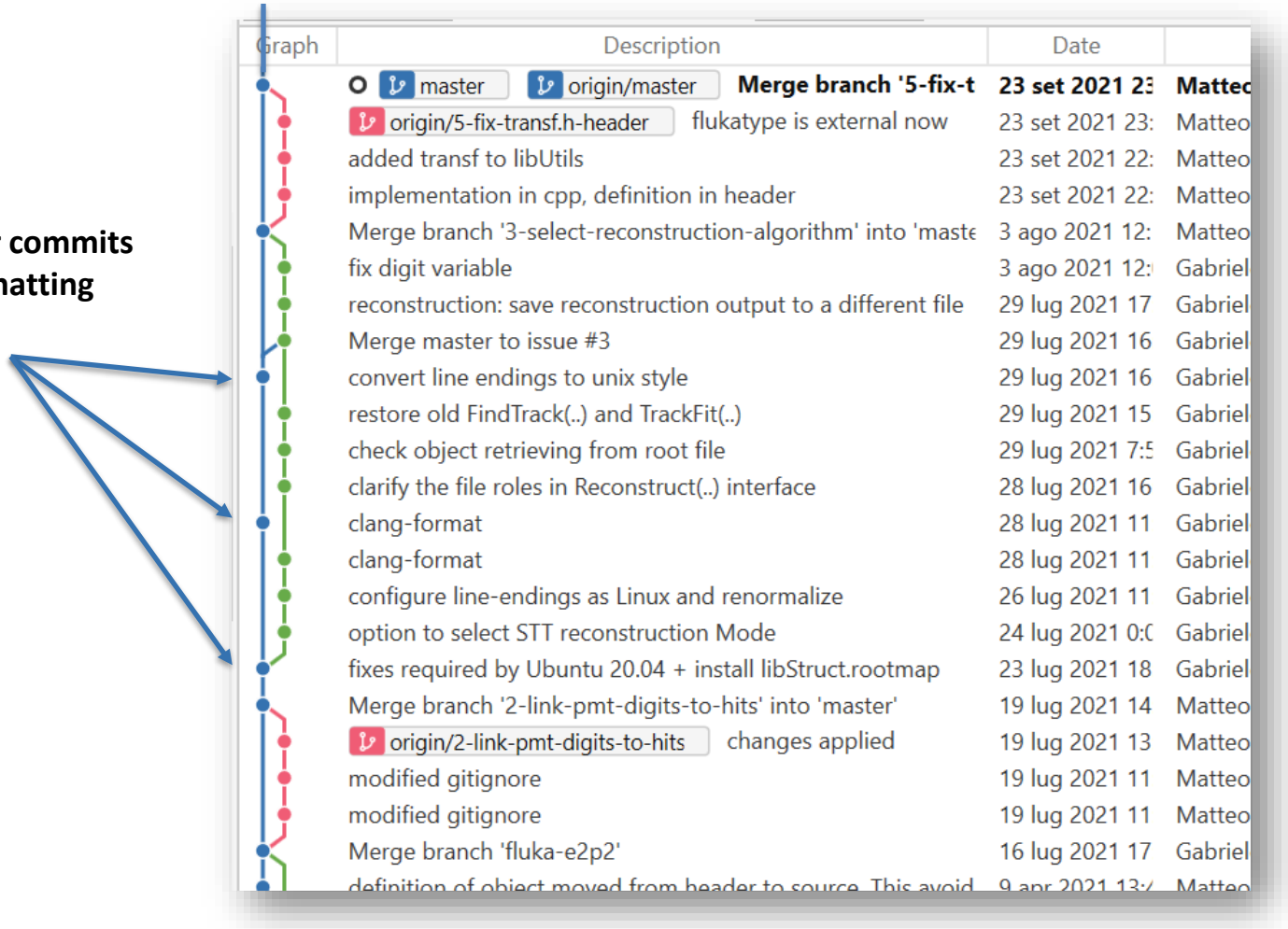
On master commit only when merge



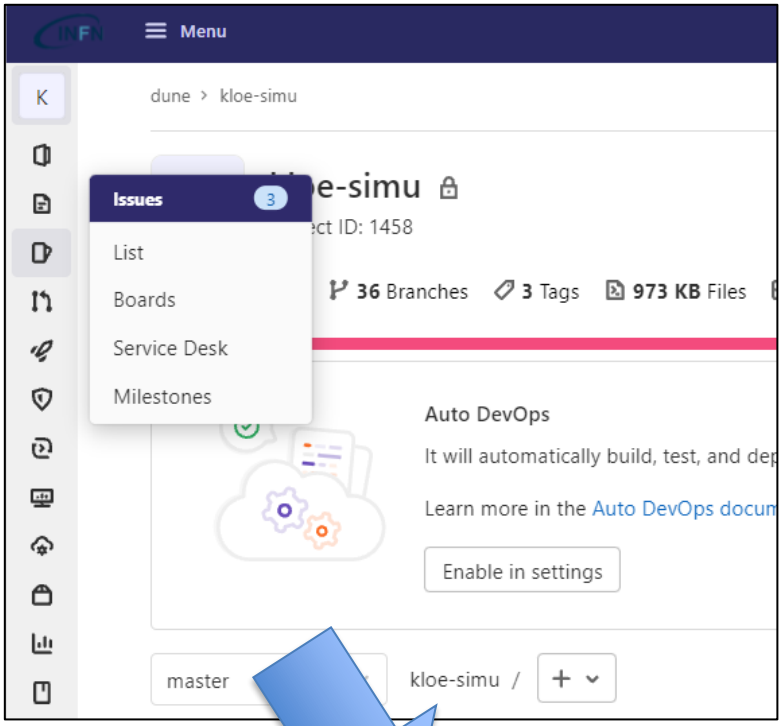
Development workflow

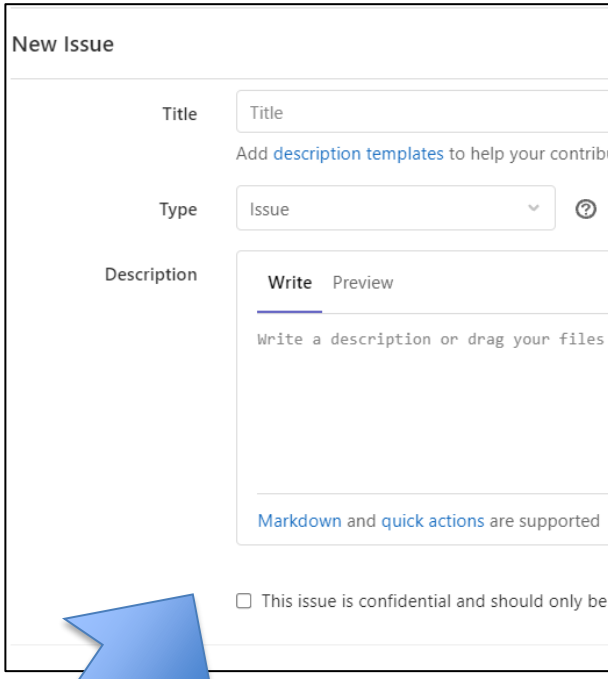
Master branch

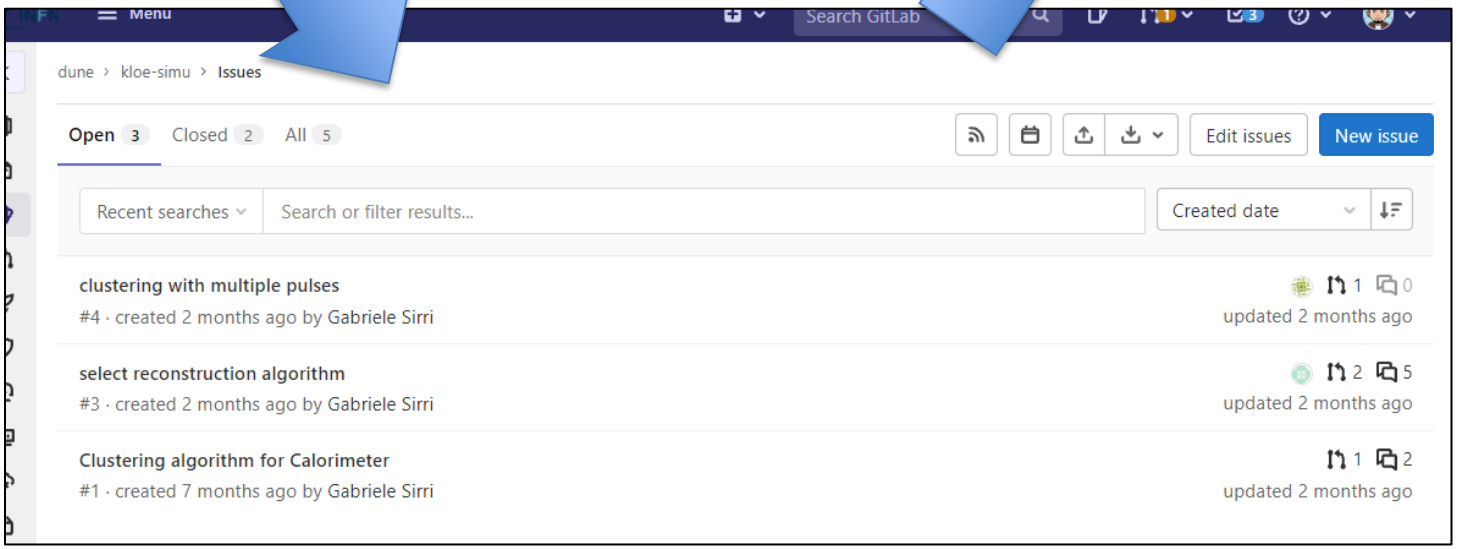
or
very minor commits
- code formatting
- bug fix



1. Create Issue
2. Open Branch
3. Develop
4. Test / Review
5. Merge Request
6. Release





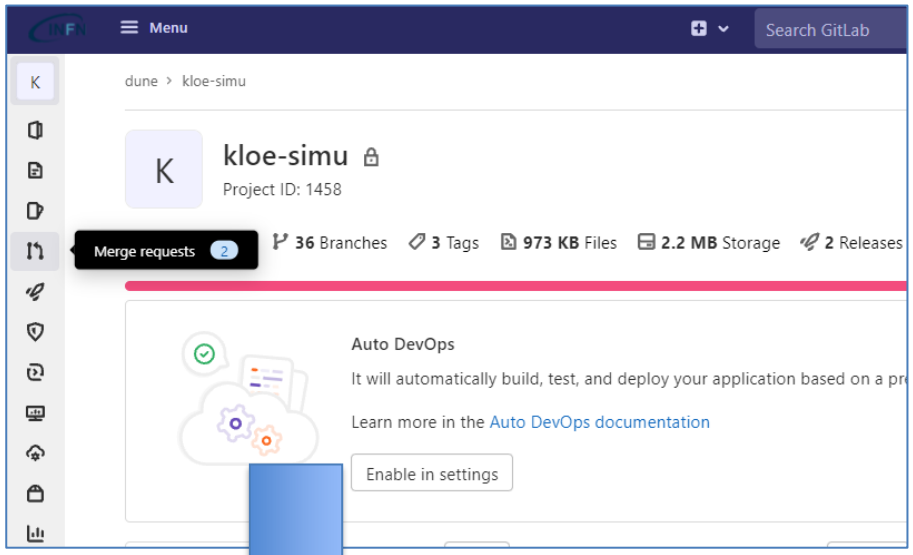


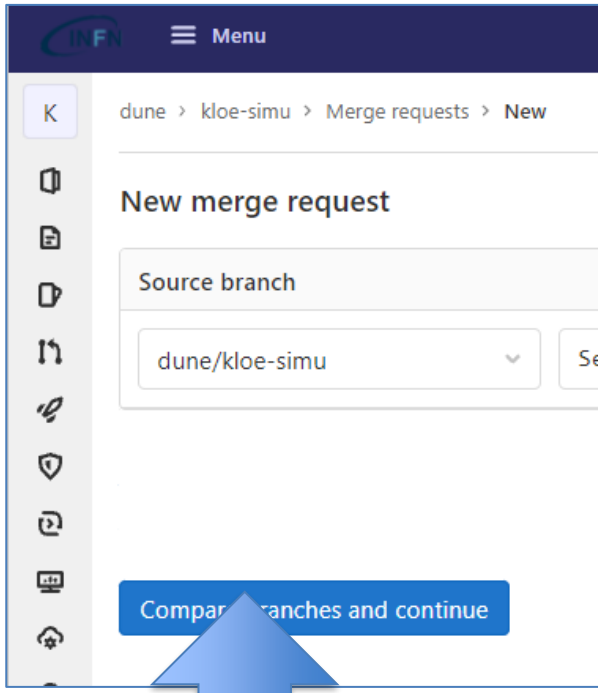
1. Create Issue
2. Open Branch
3. Develop
4. Test / Review
5. Merge Request
6. Release

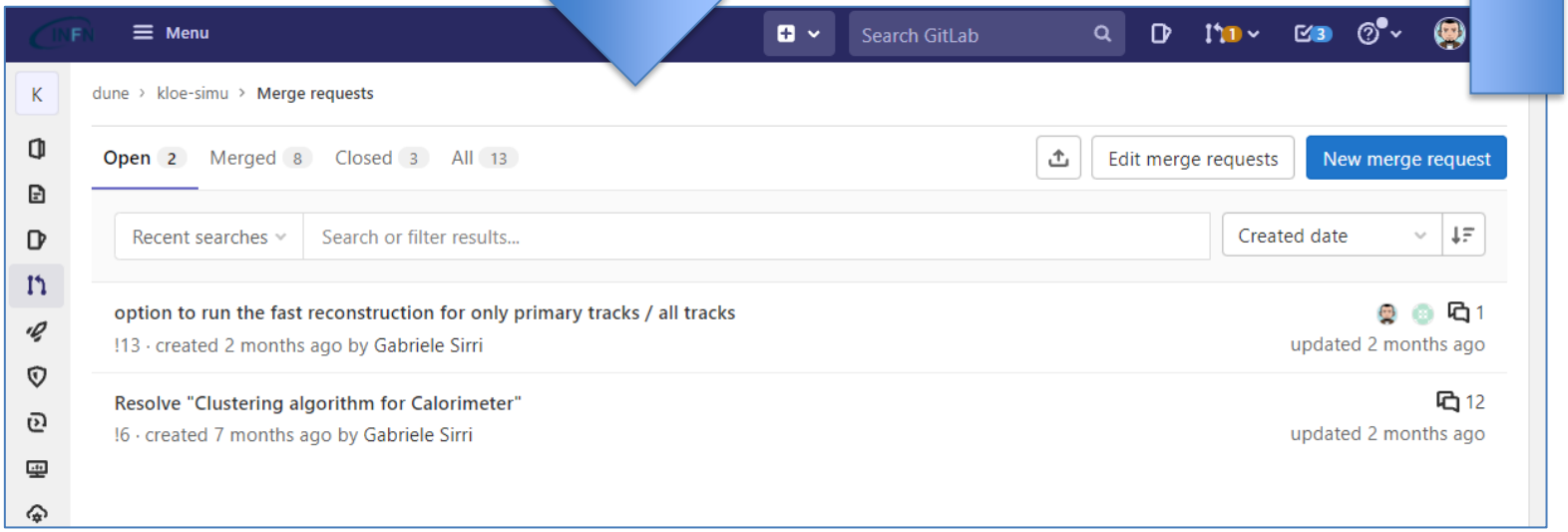
Testing and code review may be permorfed at each PUSH by other developers or continous integration

```
tentineutrino@neutrino-01:analysis (redesign-STTCluster)$ git checkout -b new_branch
Switched to a new branch 'new_branch'
tentineutrino@neutrino-01:analysis (new_branch)$ echo new_file > new_file
tentineutrino@neutrino-01:analysis (new_branch)*$ git add new_file
tentineutrino@neutrino-01:analysis (new_branch)*$ git commit -m "created new_file"
[new_branch 8950a83] created new_file
 1 file changed, 1 insertion(+)
 create mode 100644 new_file
tentineutrino@neutrino-01:analysis (new_branch)$ git push --set-upstream origin new_branch
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 281 bytes | 281.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for new_branch, visit:
remote:   https://baltig.infn.it/tenti/STTTrackReco/-/merge_requests/new?merge_request%5Bsource_branch%5D=new_branch
remote:
To baltig.infn.it:tenti/STTTrackReco.git
 * [new branch]      new_branch -> new_branch
Branch 'new_branch' set up to track remote branch 'new_branch' from 'origin'.
tentineutrino@neutrino-01:analysis (new_branch)$
```

1. Create Issue
2. Open Branch
3. Develop
4. Test / Review
5. Merge Request
6. Release





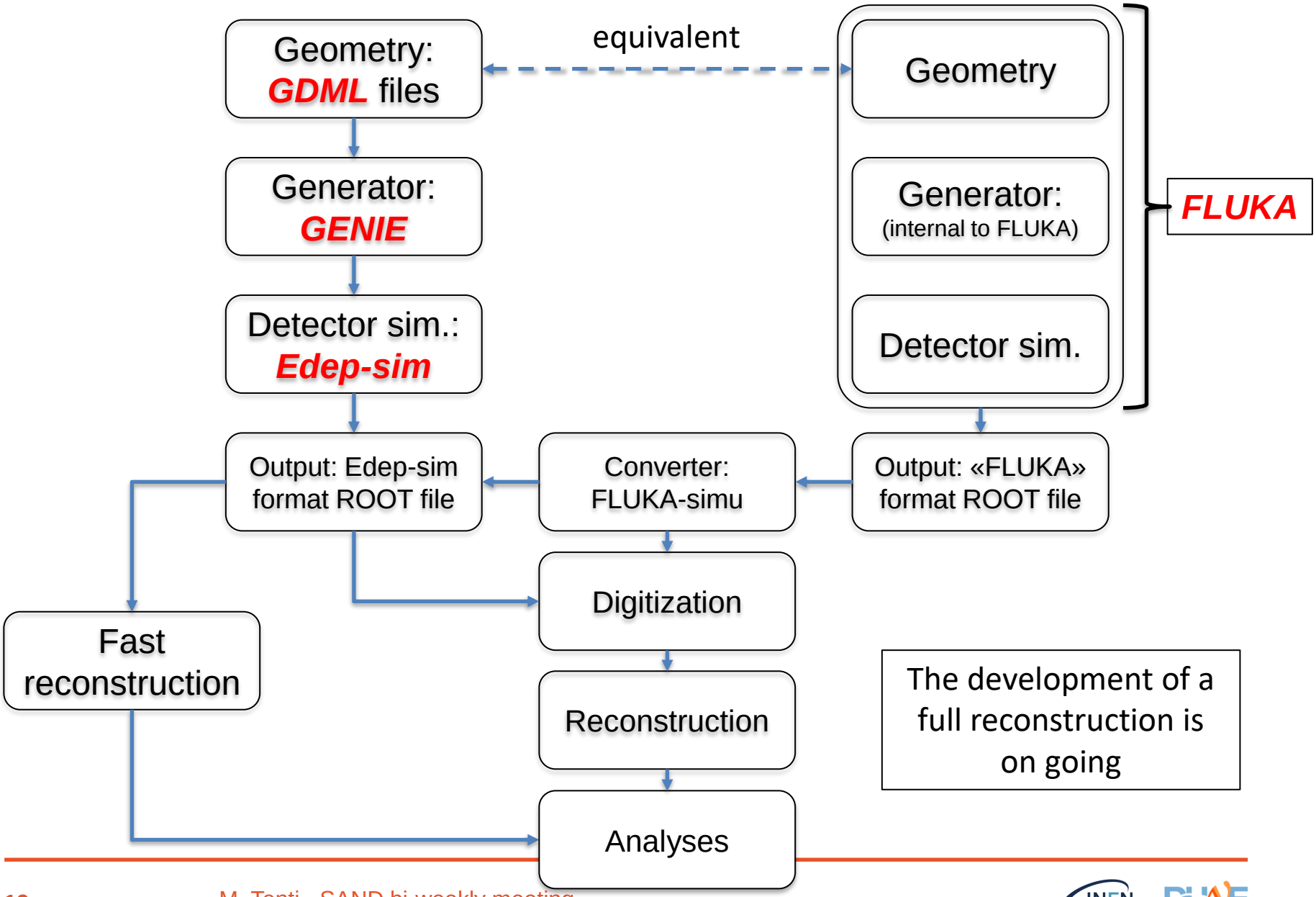


Conclusions

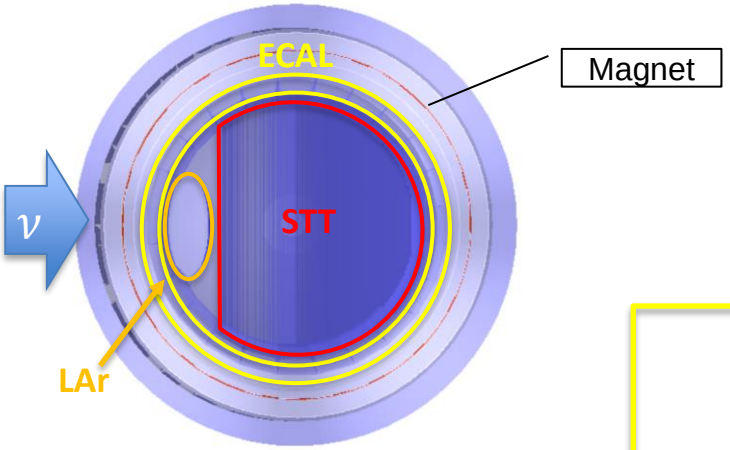
- A lot of work has been done … and has to be done
- We successfully merged SAND geometry in the official ND sim. and reco repository
- We defined SW design and interfaces
- We setup repositories for the analysis tools and rules to interact with it

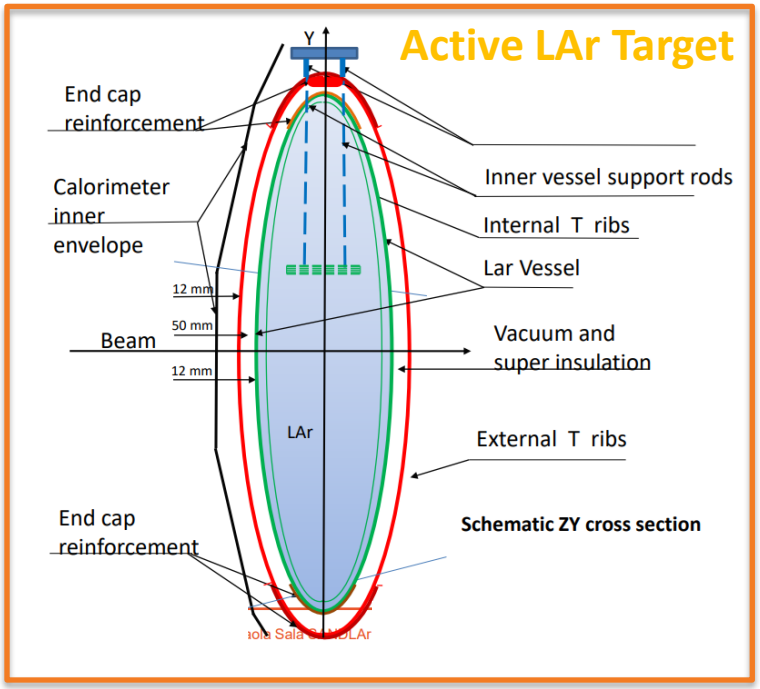
Backup

Status of the software in a nutshell



GDML geometry: ECAL+STT+LAr





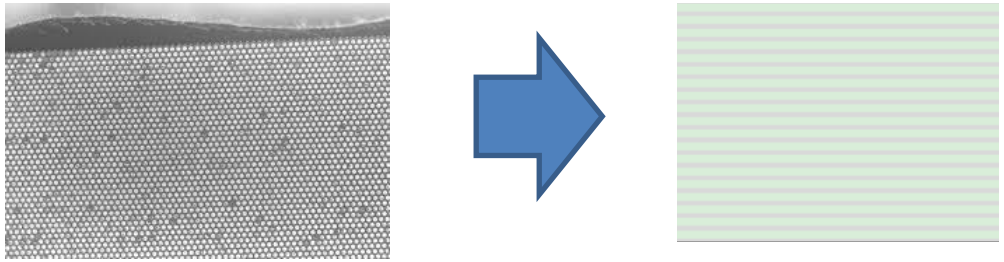
~ 90 STT modules:

- target (CH2 or C)
- radiator (plastic foils)
- XX straw tube plane
- YY straw tube plane

ECAL

24 barrel modules + 2 endcaps

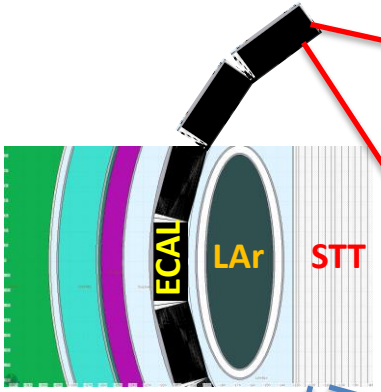
Spaghetti calorimeter approximated as 209 scintillation layers alternated with 209 lead layers



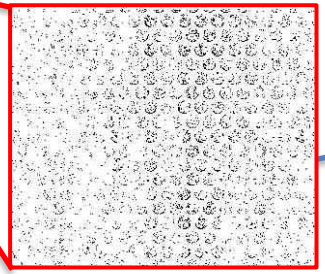
0.7 mm scintillation layer (green)

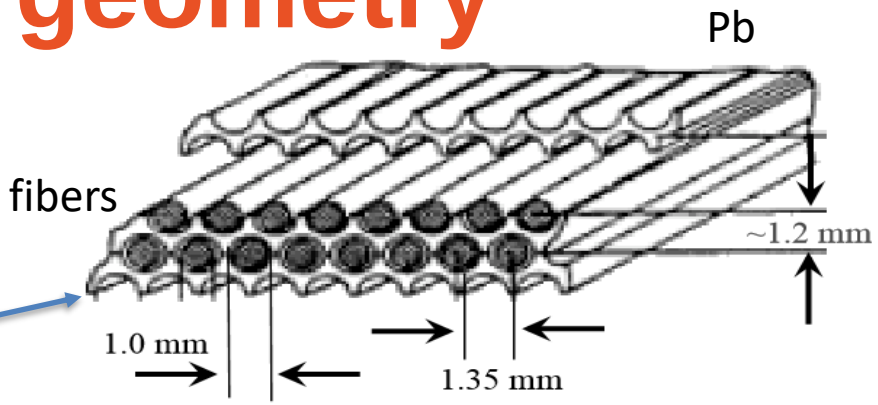
0.4 mm lead layer (gray)

Equivalent FLUKA geometry

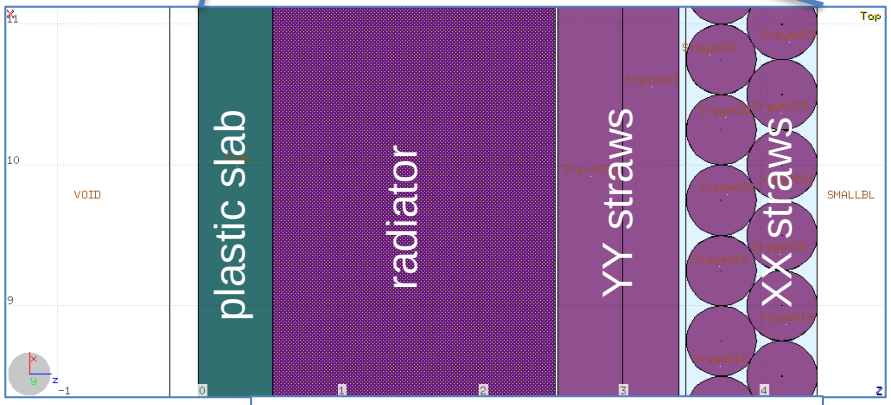


Detail of the scintillating fibers

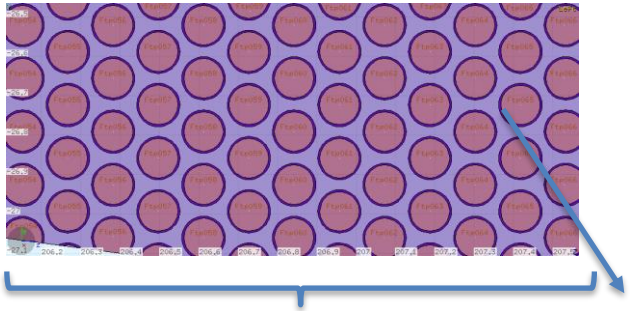




Kloecal fine structure



Detail of a STT module



Detail of a ECAL module implemented in FLUKA: fibers, glue, lead

GENIE

- We developed a dedicated code based on GENIE 3 (GENIE 3.00.06) to generate neutrino events in the detector. The code:
  - accepts geometry files in gdml format
  - accepts DUNE beam spectrum files in root format
  - takes into account the beam direction (theta=0.101 rad). Beam size set to 3 m.
  - generates any neutrino flavors in the beam and all possible interactions but it is possible to change:
    - the neutrino flavor list
    - the GENIE tune (default: G18_02a_00_000)
    - the GENIE event generator list (default: Default)

Digitization (for FLUKA files)

- A proper and distinct from GEANT4 process is necessary
  - since the geometry information are stored in different way (from SttTree)
  - since for ECAL the simulation details are different

→the parameters for p.e./MeV and the p.e. time distribution were tuned accordingly to the measured values for MIP particle crossing the middle of the barrel module (as done for geant4)

THE OUTPUT from digitization is the same as from edepsim!
→ the reconstruction will be the same for fluka and genie+edepsim chains

Digitization: ECAL

NIM A 482 (2002) 364-386

- Detailed digitization of the ECAL response takes into account:
  - Number of photons per deposited energy; scintillation time; attenuation and propagation time along the fibers; response of PMT
- Reproduction of measured performances:

Time resolution

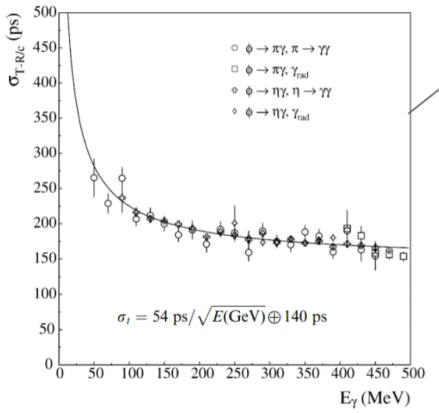
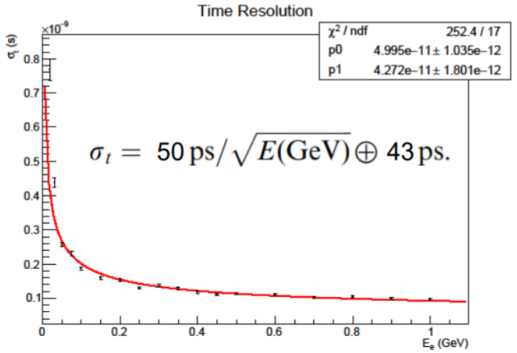


Fig. 32. Time resolution as a function of E_γ for ϕ radiative decays.

Intrinsic resolution:
$\sigma_t = 54\ \mathrm{ps}/\sqrt{E(\mathrm{GeV})} \oplus 50\ \mathrm{ps}$.



Energy resolution

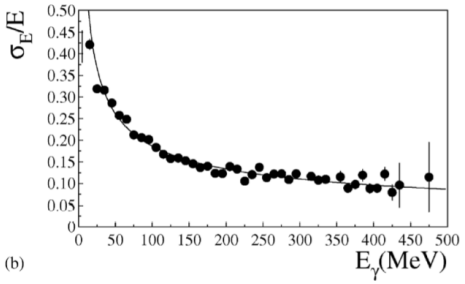
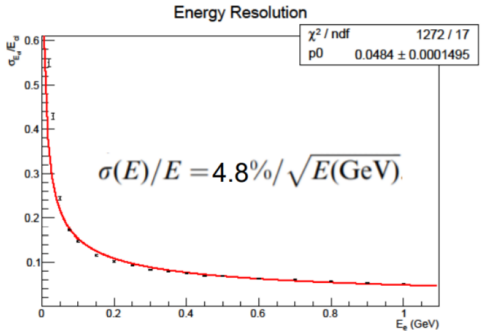


Fig. 20. $(E_{cl} - E_\gamma)/E_\gamma$ (a) and resolution (b) vs. E_γ for $e^+e^-\gamma$ events. The fit gives $\sigma(E)/E = 5.7\%/\sqrt{E(\mathrm{GeV})}$.



Digitization: STT

- STT space-resolution (0.2 mm for X and Y coordinates, 0.1 mm for Z coordinate) simulated by means of Gaussians
- Energy threshold for STT-hits: 0.1 keV
- For any charged particle in MC-tracks, hits for each STT plane are grouped to get the “STT-digits” in X-Z and Y-Z views
  - Digit coordinates from the average of hit coordinates
  - Time-resolution on STT digits: 1 ns (Gaussian smearing)

Reconstruction strategy (without MC truth)

- **Step 0** - Vertex reconstruction based on STT-hit topology
- Track finding (global transform method)
- Linear or circle fits of the tracks
- **Step 1** - Vertex reconstruction from crossing of 2 most rigid tracks
- Possible procedure iteration

- Track matching ⇒ 3D track

in both views (Y-Z and X-Z)

- $p_\perp$ from Larmor radius
- dip-angle λ from x-vs-ρ fit

momentum estimate $p = p_\perp / \cos\lambda$

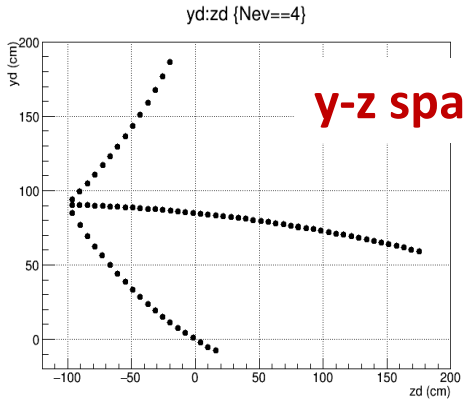
- ECal hits compatible with tracks ⇒ ToF measurement

⇒ β estimate for each track ⇒ charged particle Id …

From Vertex to Track reconstruction

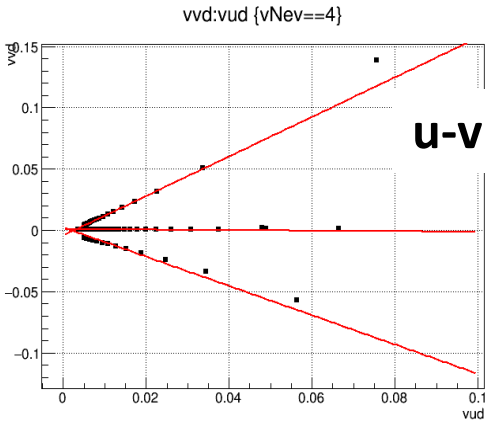
Coordinate transformation by using reco-Vertex (z_V,y_V):

$$u = +(z-z_V) / [(z-z_V)^2 + (y-y_V)^2]$$
$$v = -(y-y_V) / [(z-z_V)^2 + (y-y_V)^2]$$



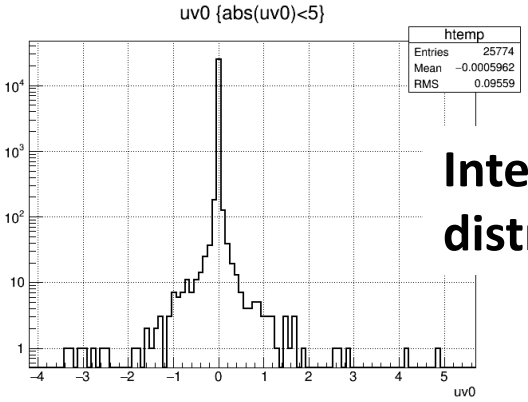
y-z space

Curved trajectories become straight lines



u-v space

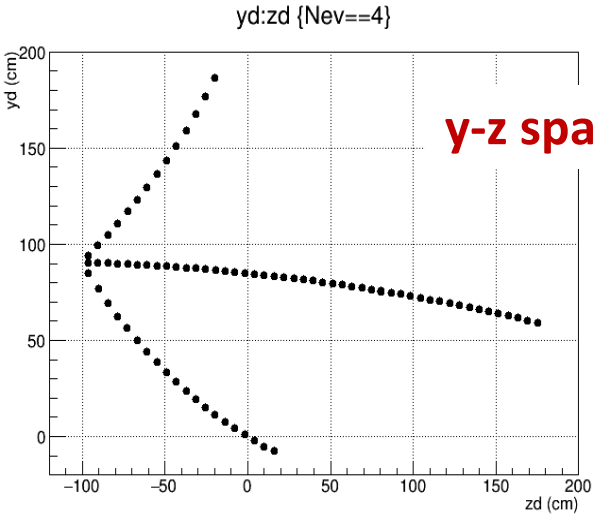
Parameters of tracks in u-v space



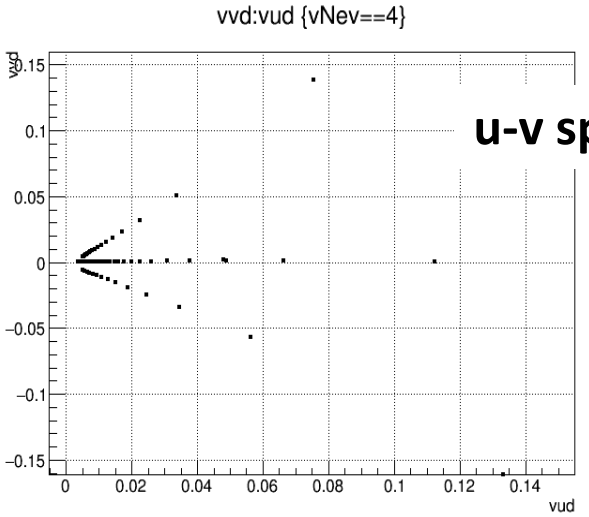
Intercept distribution

Intercepts more peaked at 0 if Vertices are well reconstructed

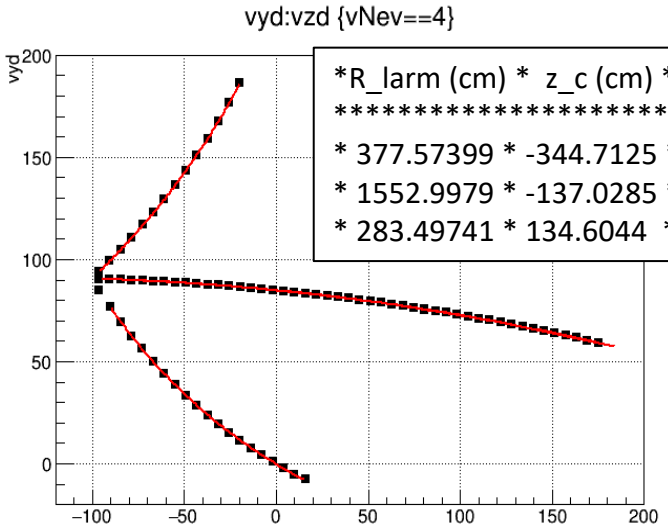
Track reconstruction by fits



y-z space

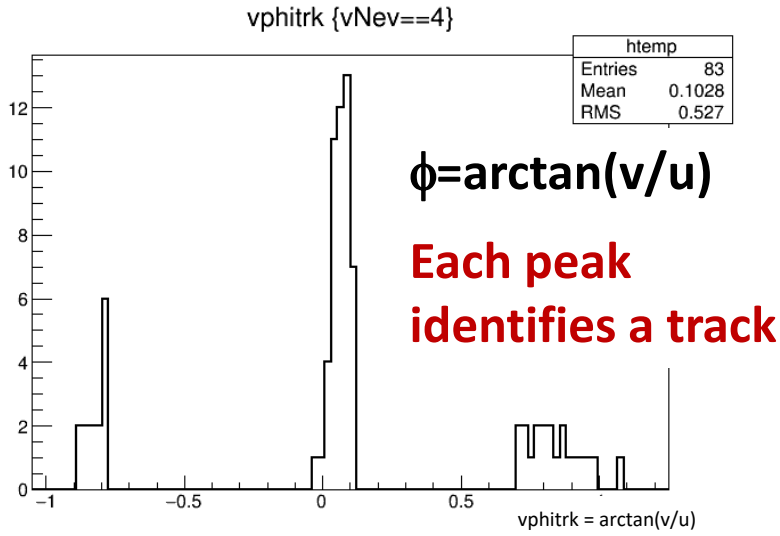


u-v space



*R_larm (cm) * z_c (cm) * y_c (cm) *
****
* 377.57399 * -344.7125 * 378.457
* 1552.9979 * -137.0285 * -1462.071
* 283.49741 * 134.6044 * 249.4256

Track finding and fits



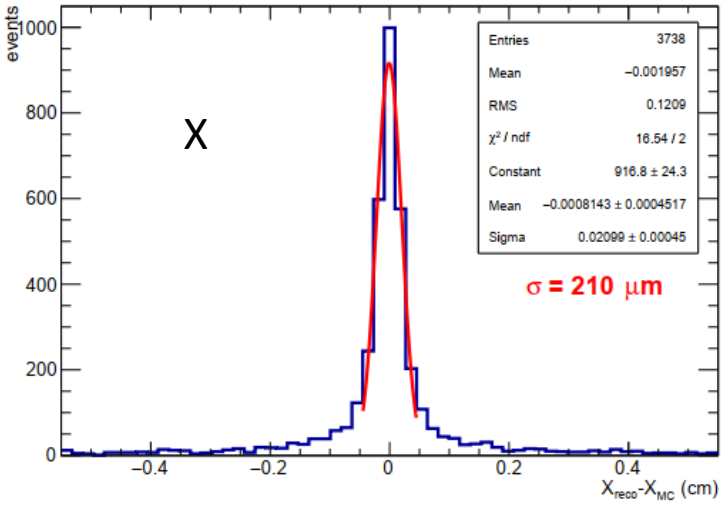
ϕ=arctan(v/u)

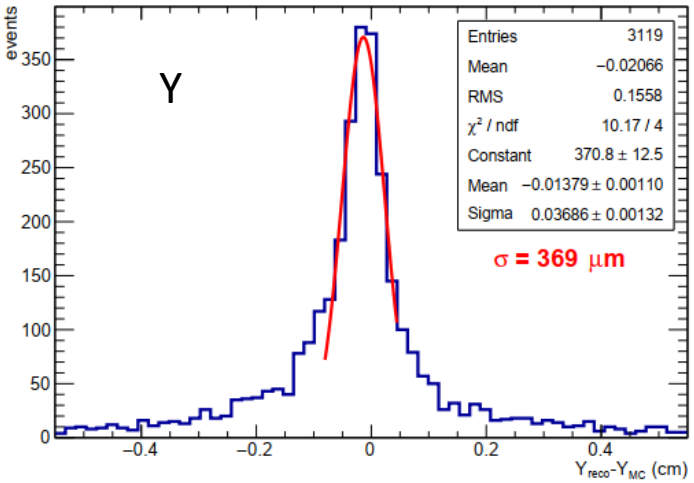
Each peak identifies a track

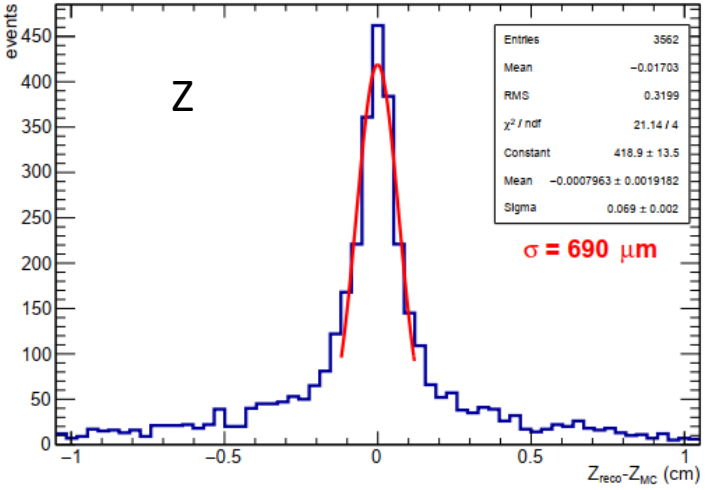
Each trajectory is fully reconstructed!

Vertex position resolution in STT

- After step 1 (see slide 9)
- Reconstruct vertex position using two most rigid tracks







Identification of charged tracks

From 3D-track: evaluation of Track-Length (L) and Time of Flight (ToF)

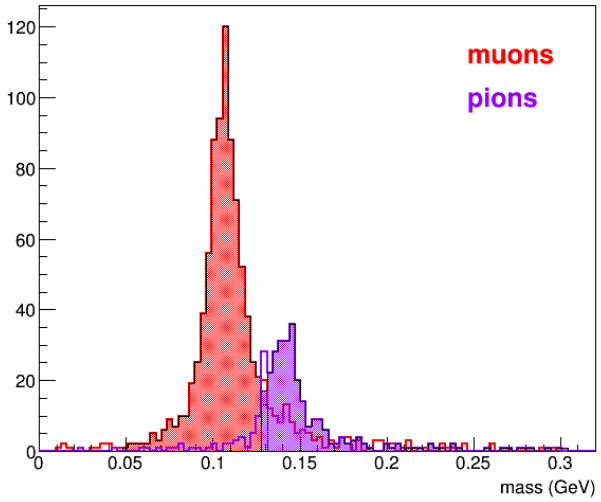
⇒ velocity estimate: $\beta = L / ToF$

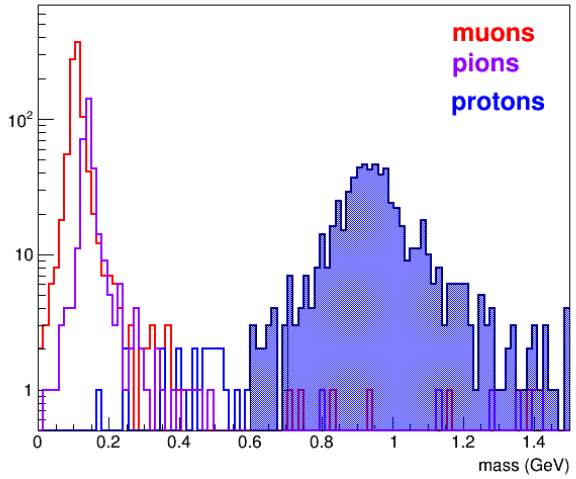
✓ L from sum of distances between STT-digits along the 3D-trajectory

✓ ToF from MC-times of STT-digits … → time resolution NOT included here!

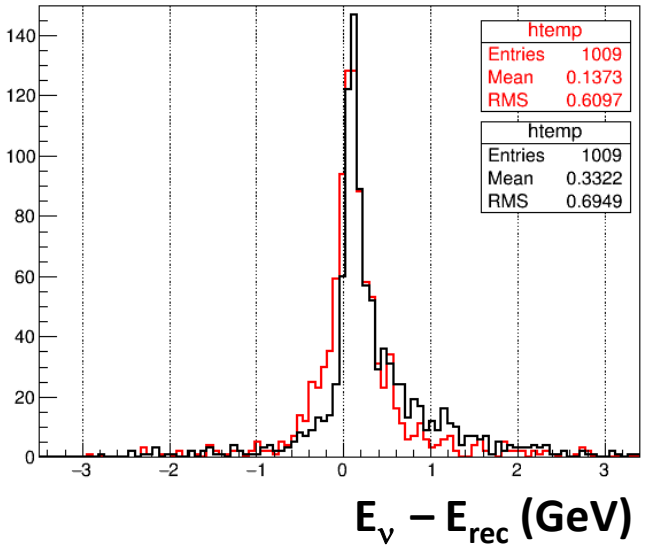
Particle identification: $m = p / \beta \cdot \gamma = p \cdot \sqrt{(1/\beta^2 - 1)}$

Events with no more than 3 matched tracks in the two views:



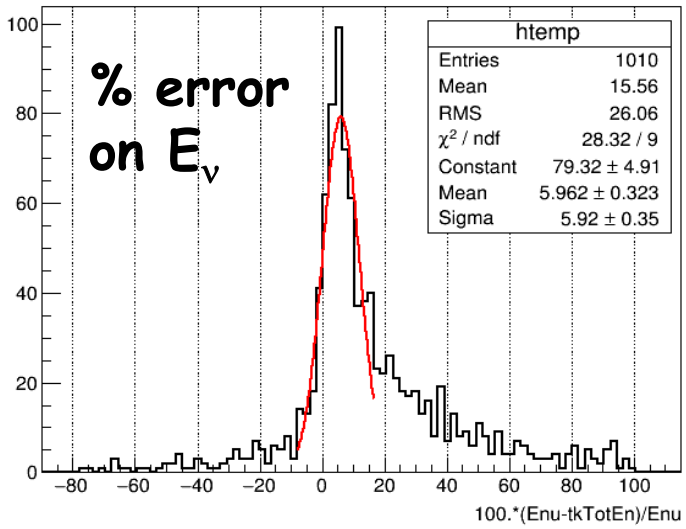


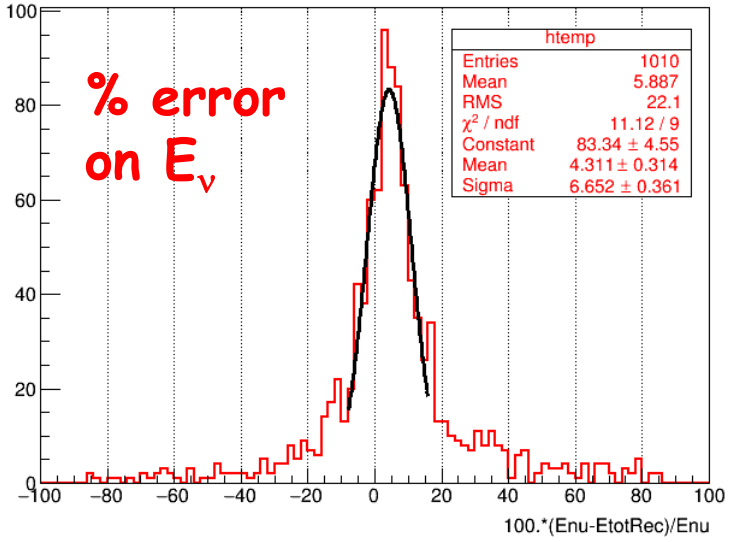
ν energy reconstruction (preliminary)



'All-tracks' energy only

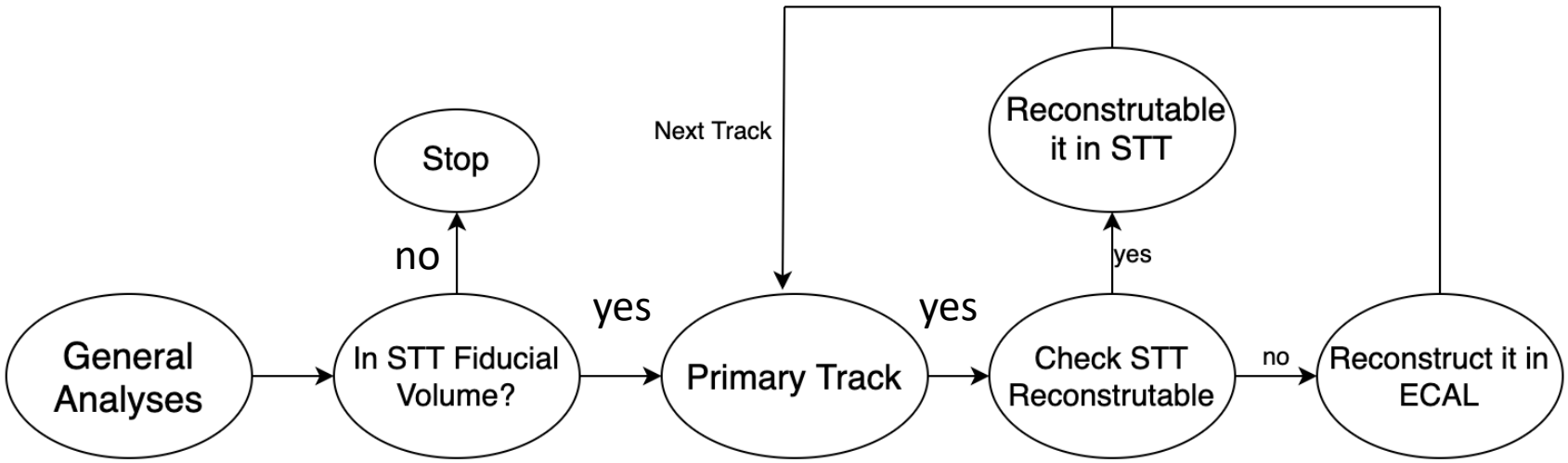
'All-tracks' energy + Off-track ECal energy





Fast Reconstruction

- Based on full detector simulation edep-sim (GEANT4)
- Treat events originating in different detector region differently
- Algorithm depends on specific analysis considered
- Single-particle smearing based on dedicated analysis/reconstruction



Fast Reconstruction in STT

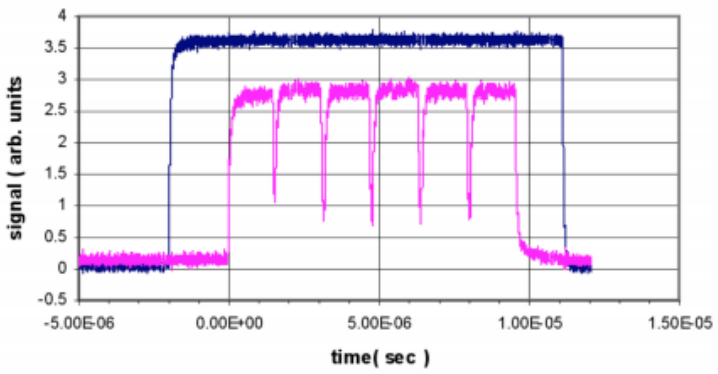
- Charged particles: Check number of Y hit in STT:
  - N(Y) < 4 (6) Stop. No smearing.
  - N(Y) >= 4 (6) Smear it.
- Charged particles: Momentum and angle smearing:
  - Gluckstern formula:
    - Based on track length, N(y), B, X0, single hit resolution.
  - Circular fitting and linear fitting
    - Need smeared position of every hit
- Neutral particles: Check its decay products:
  - Charged
  - Neutral

Neutral Particles Reconstruction

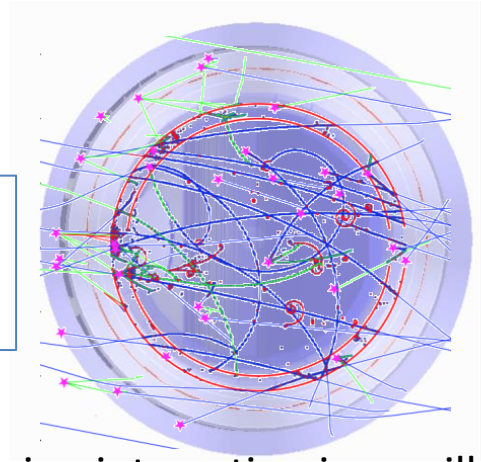
- $\pi^0 \rightarrow 2\gamma$ or $\pi^0 \rightarrow \gamma + e^- e^+$
  - Reconstruct each daughter particle's momentum separately then summing up.
- γ: $e^- e^+$ pair in STT or e.m. shower in ECAL.
  - Convert in STT: Reconstruct $e^- e^+$ track in STT
  - Convert in ECAL: find calibrated energy deposition of the e.m. shower
    - Smear earliest hit position by its resolution, connecting with vertex gives momentum direction
- Neutron: hits/cells detached from primary vertex.
  - Interaction in STT: connecting first hit (smeared) to vertex (or first hit for single track) gives direction, reconstructing the daughter tracks gives momentum.
  - Interaction in ECAL: detached cells are used to define neutral clusters, calibrated energy deposition in the cluster is summed up, connecting earliest cell to the vertex (or first hit for single track) gives momentum direction.
  - Neutron energy in CC: time-of-flight from smeared timing at primary vertex (or first hit) and earliest hit of detected neutron candidate and reconstructed direction.
  - Neutron energy in CC on Hydrogen: calculated analytically from energy-momentum conservation.

Spill simulation

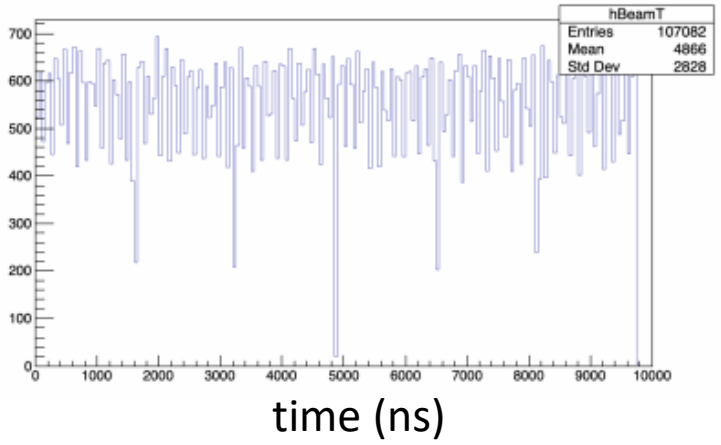
- 9.6 μs per spill
- 6 batches, 84 bunches/batch
- 2 empty bunches
- 1 bunch: Gaus(σ = 1.5 ns)
- Δt bunches = 19 ns



Star: vertex,
blue: muon track,
green: non-muon tracks



Neutrino interaction in a spill with
$N_{CaloCell}(pe > 2.5) > 0$



time (ns)