MPGD 2DStrip Readout *via* Multiple Sensitive Volumes

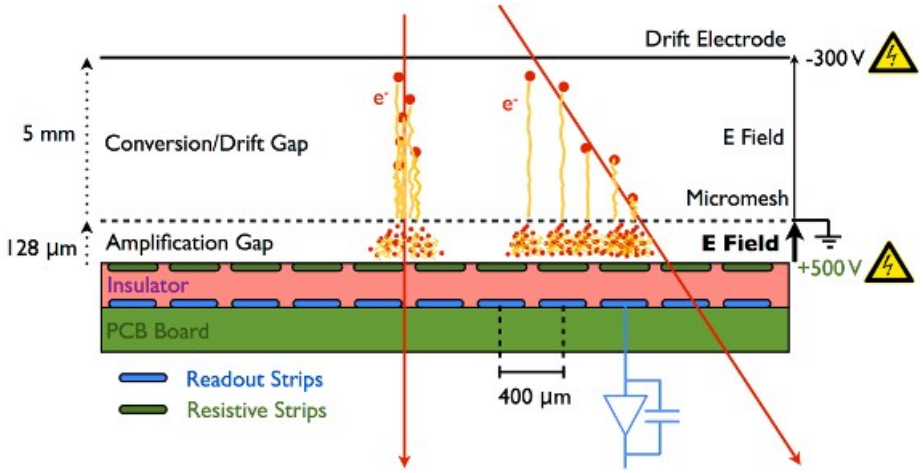
Yann Bedfer

CEA/DPhN Saclay

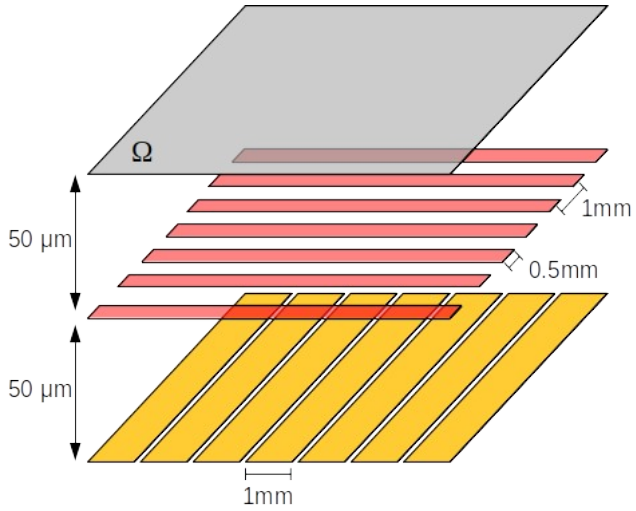
4 December 2025

2DStrip Readout

- Barrel: CyMBaL, μRWELL-BOT. EndCaps: μRWELL-ECT's.



Class12 Micromegas: 1D-strip Readout



Charge Sharing ⇒ 2D-strip Readout

- **One I**onization × **A**mplification $\xrightarrow[\textbf{Sharing}]{\textbf{Charge}}$ **Two proto** Hits, along strips $\boldsymbol{u = p, n}$
- **Charge Spreading** (*enhanced by resistive layer* Ω): **proto** Hit → **cluster** of Hits.

- **Strips** $\boldsymbol{per\ se}$, no big problem: they are **stretched pixels**. . .

  . . . *i.e.* **2D objects**, w/ **bad precision** (= $\mathcal{O}$(size)) **along one of the D's**.

- **Problem** is **2D**Strip, because **One and only one Sensitive Surface** $\boldsymbol{per}$ **volume** in `DD4hep`.

Previous Solution: MultiSegmentation + ReSegmentation

- **Original** scheme: **pixel** SimHit → **pixel** RawHit.

  <segmentation type="CartesianGridUV" **grid_size_u="0.150*mm*sqrt(12)"**
  grid_size_v="0.150*mm*sqrt(12)" **grid_angle**="M_PI*rad/4" />
  <**id**>system:8,layer:4,module:12,sensor:2,u:-16,v:-16</id>

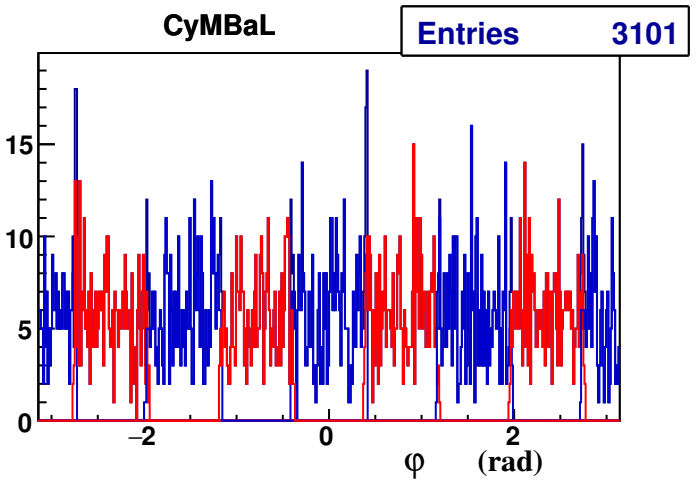
- **2D-strip**: **pixel** SimHit $\xrightarrow[\textbf{ReSegmentation}]{\textbf{MultiSegmentation}}$ **2 strip** RawHits.

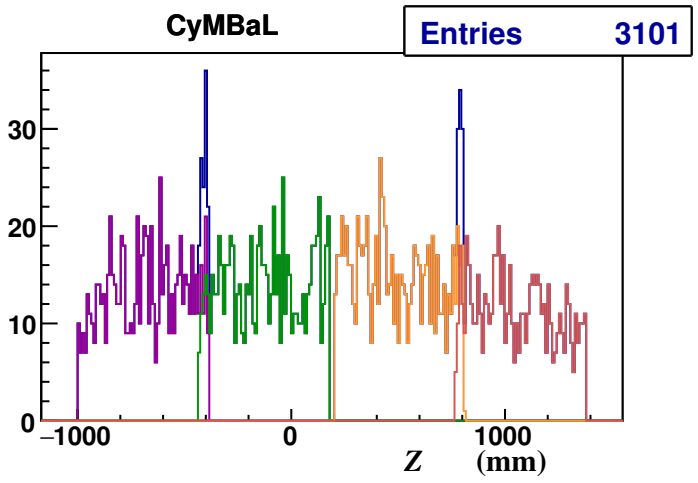
  <segmentation type="**MultiSegmentation**" **key**="**strip**">
  <segmentation name="**Pixels**" type="CartesianGridUV" key_value="**0**"
  grid_size_u="0.150*mm*sqrt(12)" grid_size_v="0.150*mm*sqrt(12)" grid_angle= . . .
  <segmentation name="**UStrips**" type="CartesianGridUV" key_value="**1**"
  grid_size_u="0.150*mm*sqrt(12)" **grid_size_v="μRWellStrip_range"** grid_angle= . . .
  <segmentation name="**VStrips**" type="CartesianGridUV" key_value="**2**"
  grid_size_u="μRWellStrip_range" grid_size_v="0.150*mm*sqrt(12)" grid_angle= . . .
  </segmentation/>
  <**id**>system:8,layer:4,module:12,sensor:2,**strip:30:2**,u:-16,v:-16 </id>

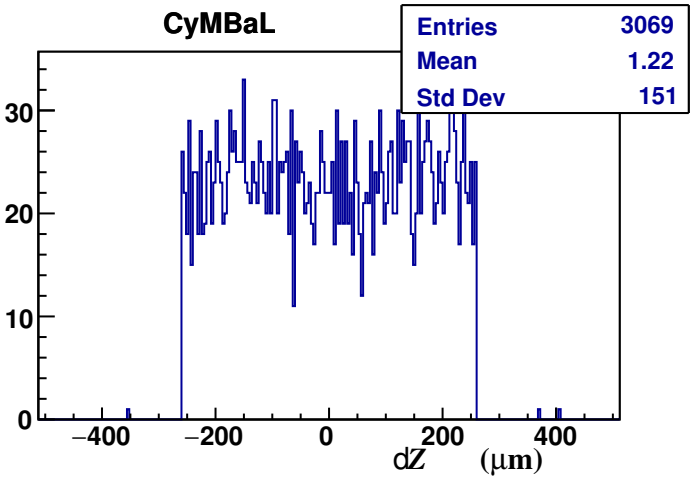
  - **ReSegmentation** done in `EICrecon`'s digitization step *via* `MPGDTrackerDigi`.

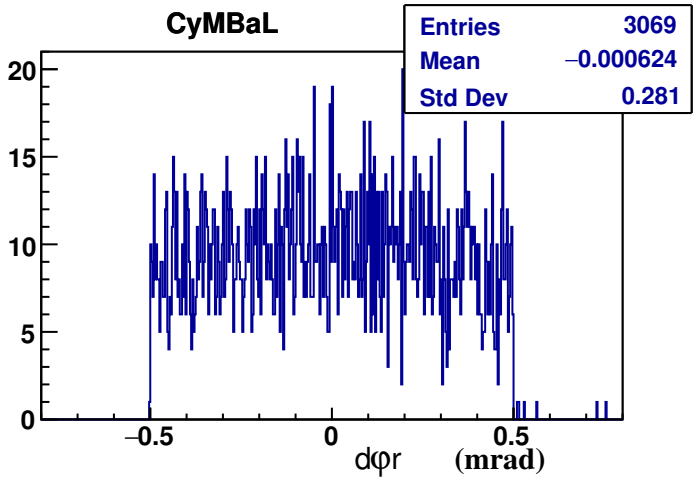
Pixel Residuals: SiliconTrackerDigi

- `ddsim: SIM.gun` μ^- $15 < \theta < 165$ deg. $P = 10$ GeV
- `eicrecon -PMPGD:SiFactoryPattern=0x3`$(=default)$ . . . $\rightarrow$ SimHits, Residuals (`RawHit-SimHit`).





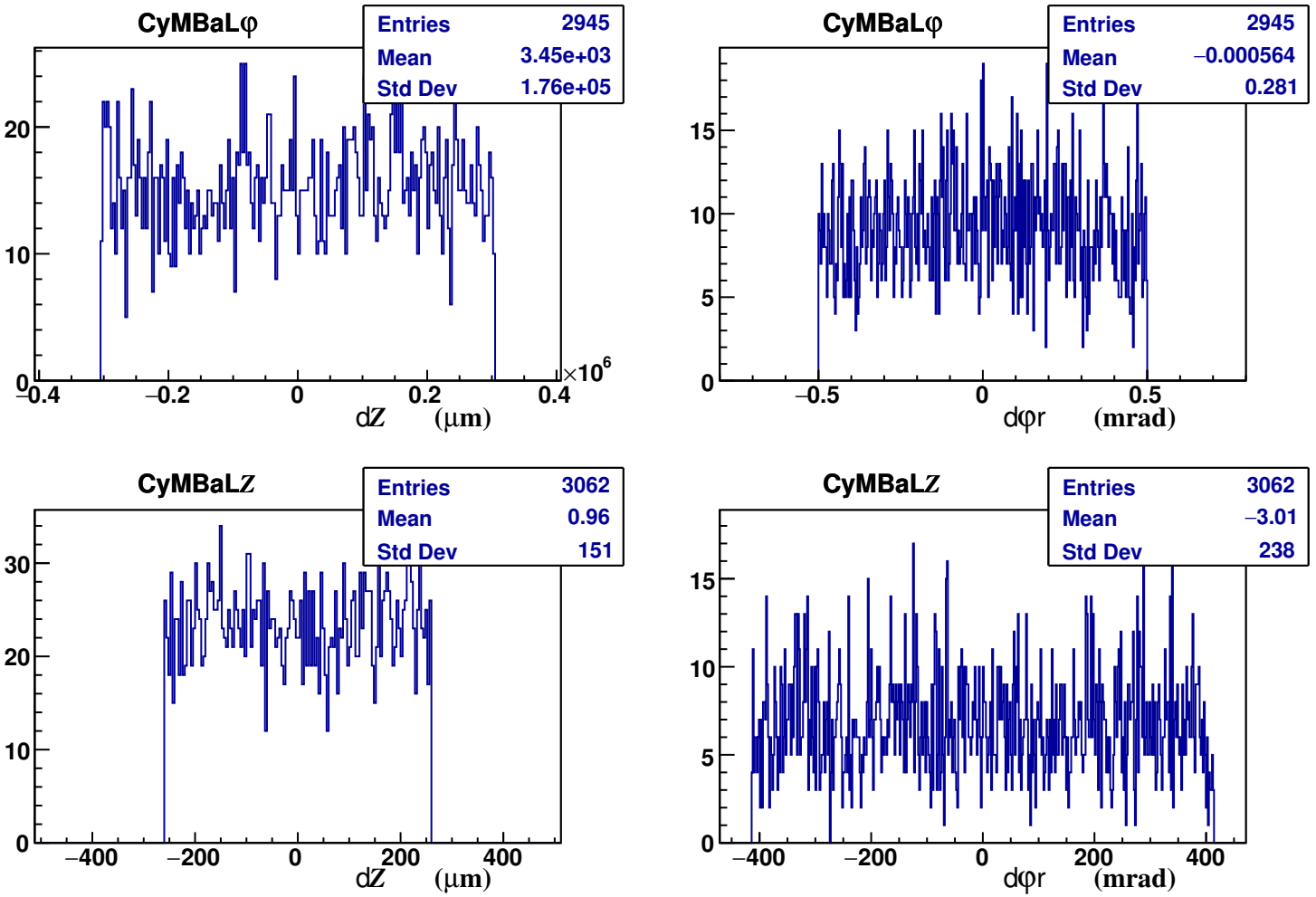




- $\boldsymbol{Z}$: 150 **μm**, $\boldsymbol{\varphi}$: $\sim$1 mrd$/\sqrt{12}$ ($\simeq$ **150 μm**$/R$).

Strip Residuals: MPGDTrackerDigi

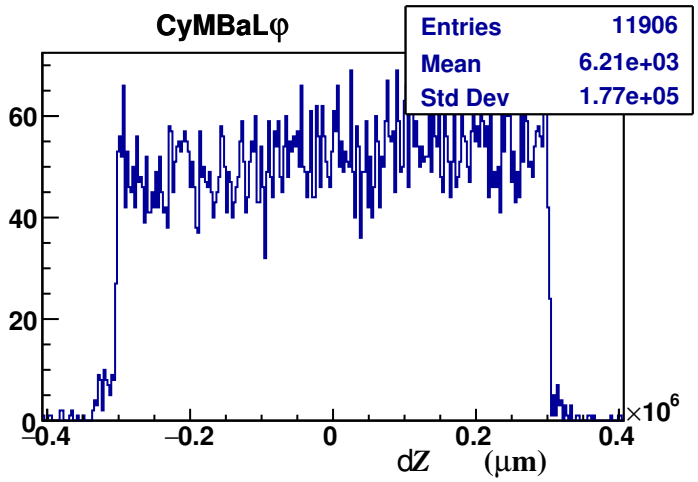
- Same simulation as before.
- `eicrecon -PMPGD:SiFactoryPattern=0x0 ...` → Residuals (RawHit-SimHit).

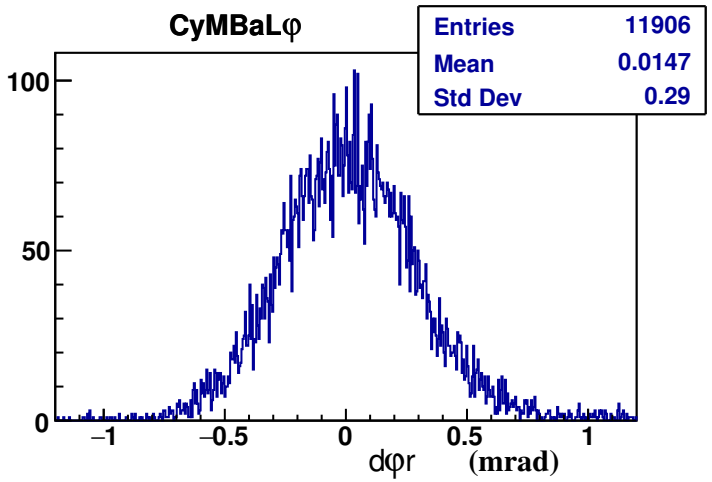


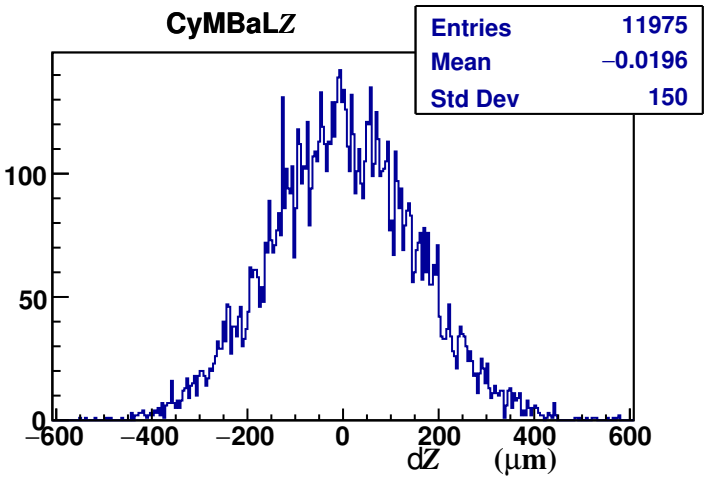
- Strip precision as expected: **150 μm** × $\mathcal{O}$**(size)**.

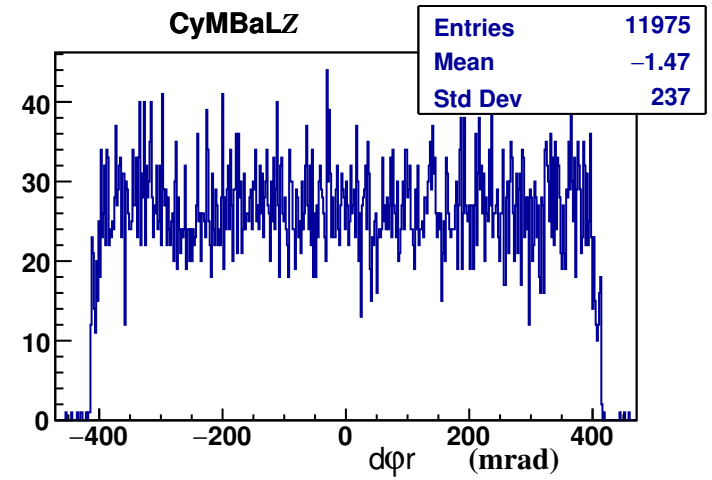
Cluster Residuals `MPGDTrackerDigi, MPGDHitReconstruction`

- Same simulation as before.
- `eicrecon -PMPGD:seed=$seed ...` → **Residuals** (**RecHit**-**SimHit**).





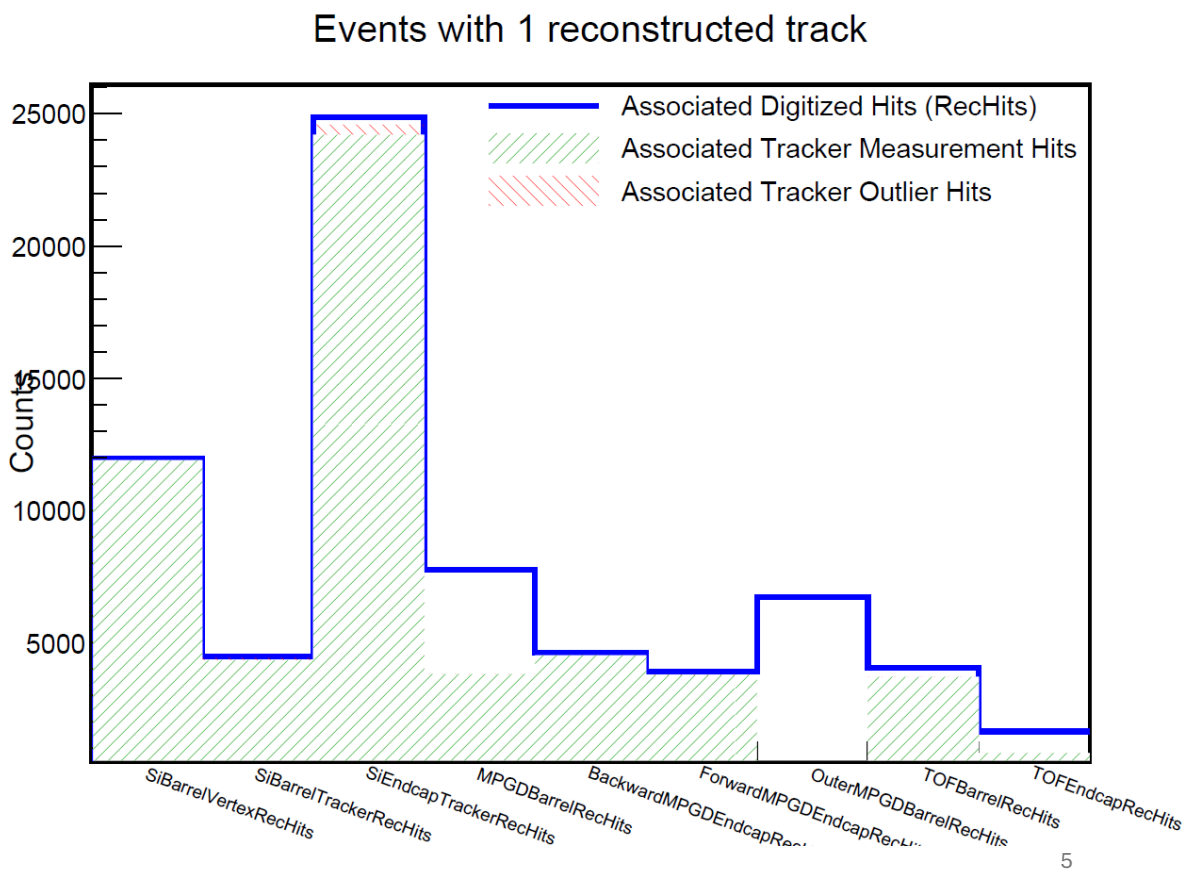




⇒ **150 μm** × $\mathcal{O}$**(size)** in Z, ∼1 mrd/$\sqrt{12}$ (≃**150 μm**/R) × $\mathcal{O}$**(size)** in φ .

Problem: 2DStrip Hit-to-Track Association

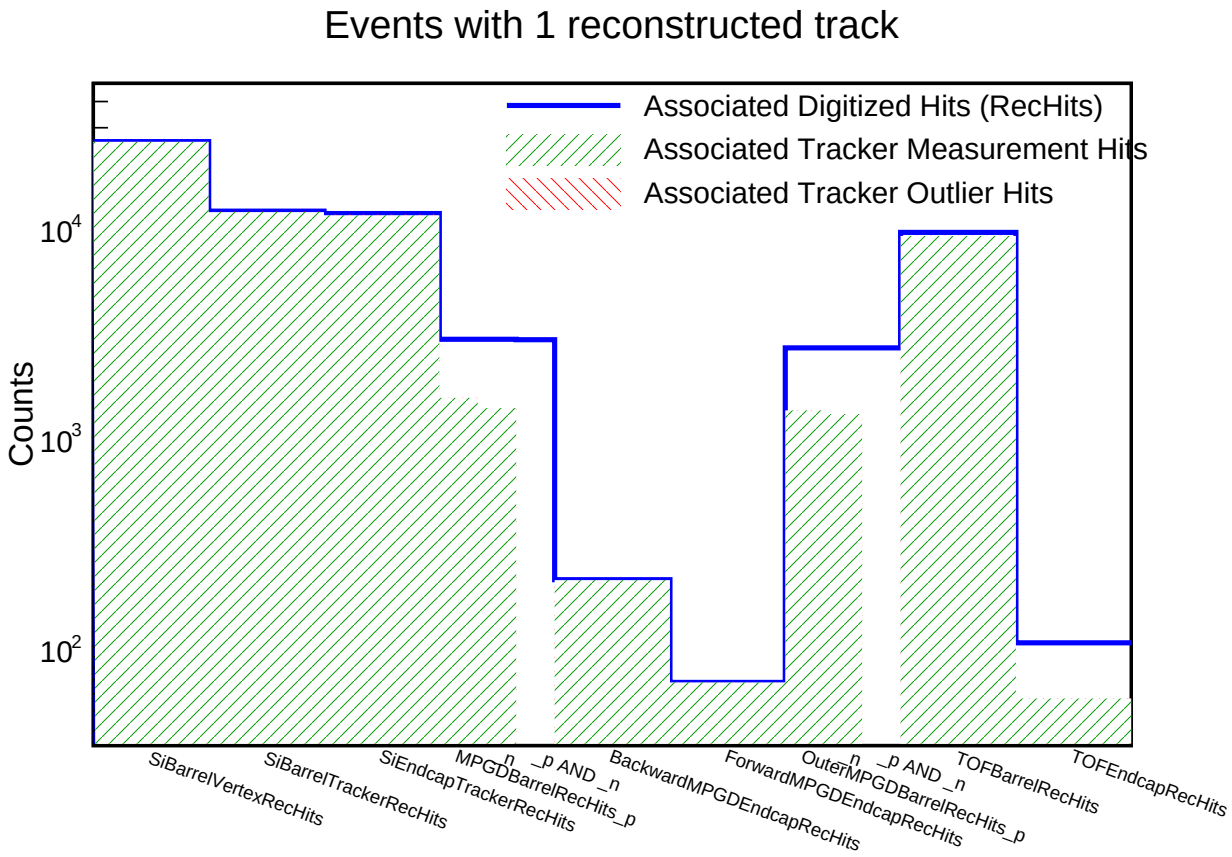
- Reported by Barak Schmookler at Tracking Meeting of April, 3rd:



5

⇒ **Hit-to-Track Association Efficiency**: CyMBaL = **50%**, Outer = **0%**.

Problem: 2DStrip Hit-to-Track Association (*cont'd*)



- Outer: Fix (*Cov.Matrix rotated* $\pi/4$, *not committed*). $\Rightarrow$ Now behaves as CyMBaL.
- All 2DStrip's: Either $\boldsymbol{p}$ or $\boldsymbol{n}$associated. Never both.

  One and only one hit associated $\boldsymbol{per}$ **Sensitive Surface** (*the least* χ^2 *increment*).

2DStrip Hit-to-Track Association: Solutions?

- Verbatim of Mattermost following Track&Vertex of 2025/07/24:

```
Ways to create 2d strip:

 1. Divide the geometry in multiple layer volume In dd4hep: this will change the
  energy deposit pattern which is a major concern.

 2. Follow the ACTS example to split the sensitive surface to two \{ preferable
  method for now.

 3. [...] write a custom DD4hep sensitive detector action that is associated with
  the gas volume, and creates sim hits on one (or more) surfaces from the Geant4
  hit inside the gas volume. see DDG4/plugins/Geant4SDActions.cpp
```

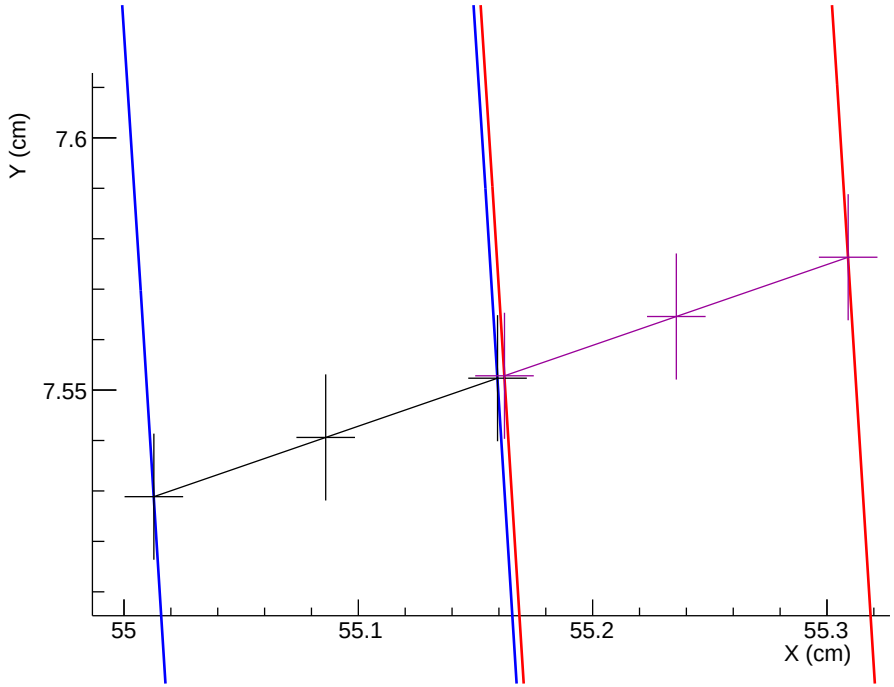
- I haven't been able to implement anything like (2) or (3).

- I have implemented a solution based on (1),

  fixing the energy deposit pattern issue.

Solution implemented: Multiple Sensitive Volumes

- See PR #2177 on `EICrecon` github.
- Principle = **Coalescence** + **Extension** + **ReSegmentation**
  - **Subdivision** into Sensitive **SubVolumes**: each of $\boldsymbol{p}$ and $\boldsymbol{n}$ gets its **own Sensitive Surface**.
  - **Coalescence** of the **SubHits**: needed to preserve $\boldsymbol{p}$ *vs.* $\boldsymbol{n}$ **Correlation**.
  - **Only one SubVolume** gets EDep: **Extend SubHit** to fill the gap.



Five **SubVolumes** in all:

Two **Radiator SubVolumes** (*red and blue*)
to collect energy deposit,

\+ Two **Readout SubVolumes** (*in central gap*)
to which RawHits are eventually assigned.

\+ A **Reference SubVolume** (*at exact centre*)
to rationalize the source code.

(*Markers @ SubHit position* $\pm$ *path-length*)

- **Guideline**: **Reproduce** as exactly as possible **Single Volume** situation, *whether or not it best simulates MPGD inner workings, e.g.: do not coalesce δ-ray w/ primary SubHit.*

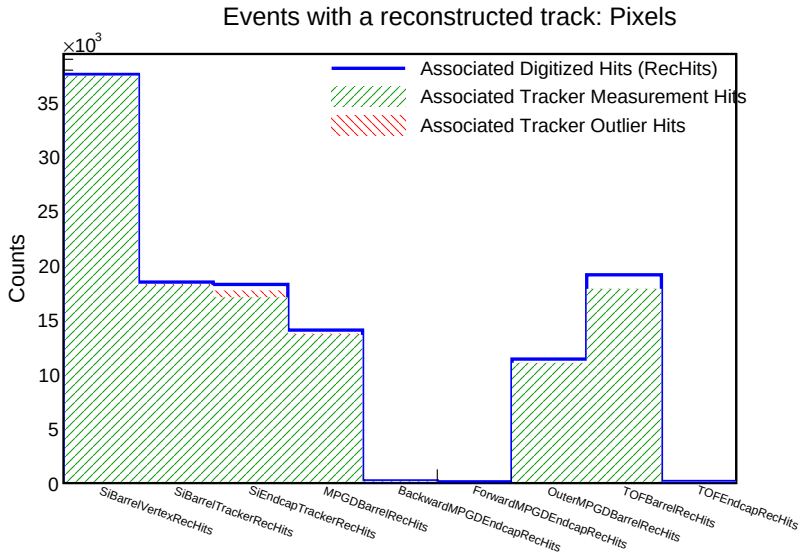
  $\Rightarrow$ **Extensive checks** conducted before **coalescing**.

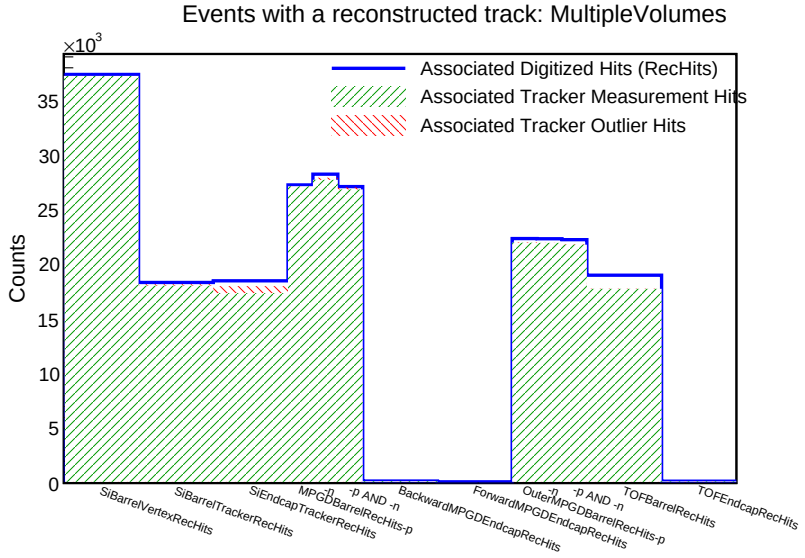
Multiple Sensitive Volumes:ReSegmentation

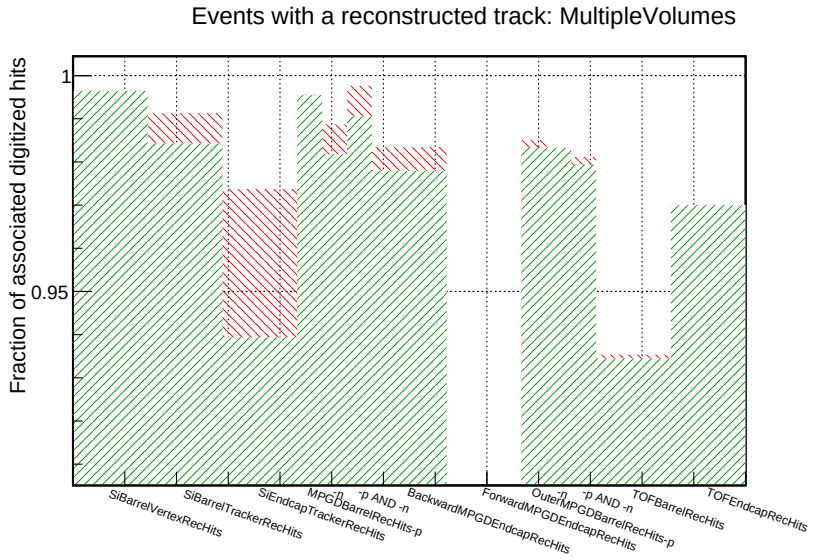
- <segmentation type="**MultiSegmentation**" **key**="**strip**">
  <segmentation name="**Pixels**" type="CartesianGridUV" key_value="**0**"
  grid_size_u="0.150*mm*sqrt(12)" grid_size_v="0.150*mm*sqrt(12)" grid_angle= . . .
  <segmentation name="**UStrips**" type="CartesianGridUV" key_value="**1**"
  grid_size_u="0.150*mm*sqrt(12)" **grid_size_v**="**µRWellStrip_range**" grid_angle= . . .
  <segmentation name="**VStrips**" type="CartesianGridUV" key_value="**2**"
  grid_size_u="**µRWellStrip_range**" grid_size_v="0.150*mm*sqrt(12)" grid_angle= . . .
  <segmentation name="**BotRad**" type="CartesianGridUV" key_value="**3**"
  grid_size_u="0.150*mm*sqrt(12)" grid_size_v="0.150*mm*sqrt(12)" grid_angle= . . .
  <segmentation name="**TopRad**" type="CartesianGridUV" key_value="**4**"
  grid_size_u="0.150*mm*sqrt(12)" grid_size_v="0.150*mm*sqrt(12)" grid_angle= . . .
  </segmentation/>
  <**id**>system:8,layer:4,module:12,sensor:2,**strip:28:2**,u:-16,v:-16 </id>
  - **ReSegmentation** done in `EICrecon`, based on **Coalesced/Extended SimHit position**.

Multiple Sensitive Volumes: Hit-to-Track Association

- Association **Efficiency recovered**:







- Twice (*approx.*) as many SimHits.
- **RawHits** are **associated** *collection-wise* to **all coalesced SubHits**.

Note: CyMBaL non ***pn****-symmetric*

Multiple Sensitive Volumes: Pros and Cons

- Cons:
  - It's **complicated**: $\tilde{1}500$ lines of code.
  - Coalesced/Extended SimHits not available in `podio` output.

- Pros:
  - **It's working!**
  - It's debugged: **Many thanks to Dmitri Kalinkin**!
  - It's available. 3 PRs:

    `MPGDTrackerDigi` + *little fix to* `TrackerMeasurementFromHits`

    Almost no impact on `EICrecon` behavior.

    `epic`: Actual commit of Multiple Sensitive Volumes.

    `MPGDTrackerDigi` and `MPGDHitReconstruction`: Clusters.
  - Backward compatibility:

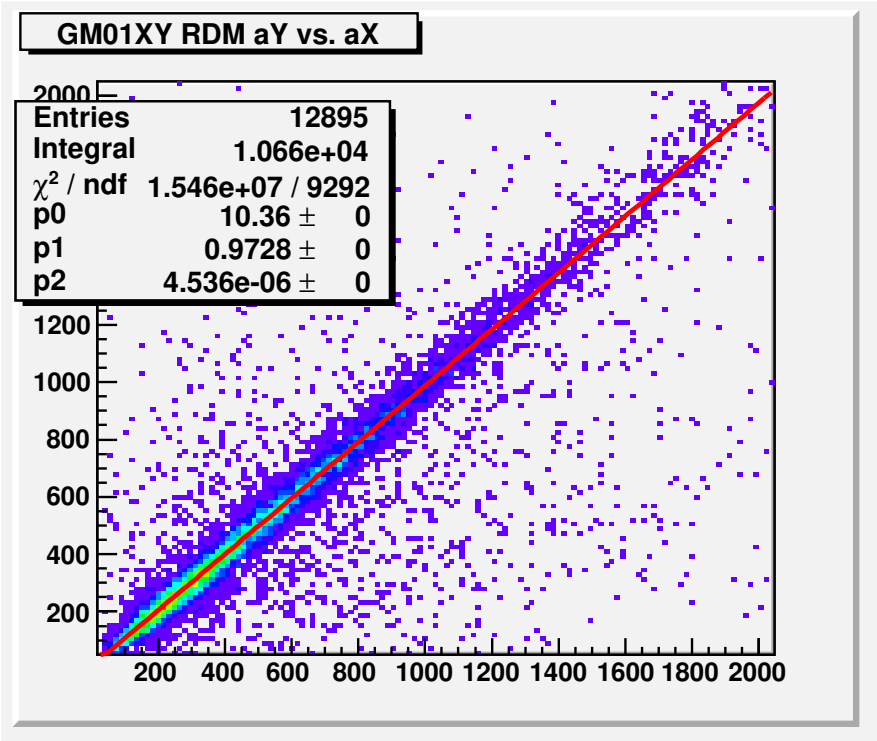
    Each MPGD XML comes in two flavors: `_pixel` and `_2Strip`,

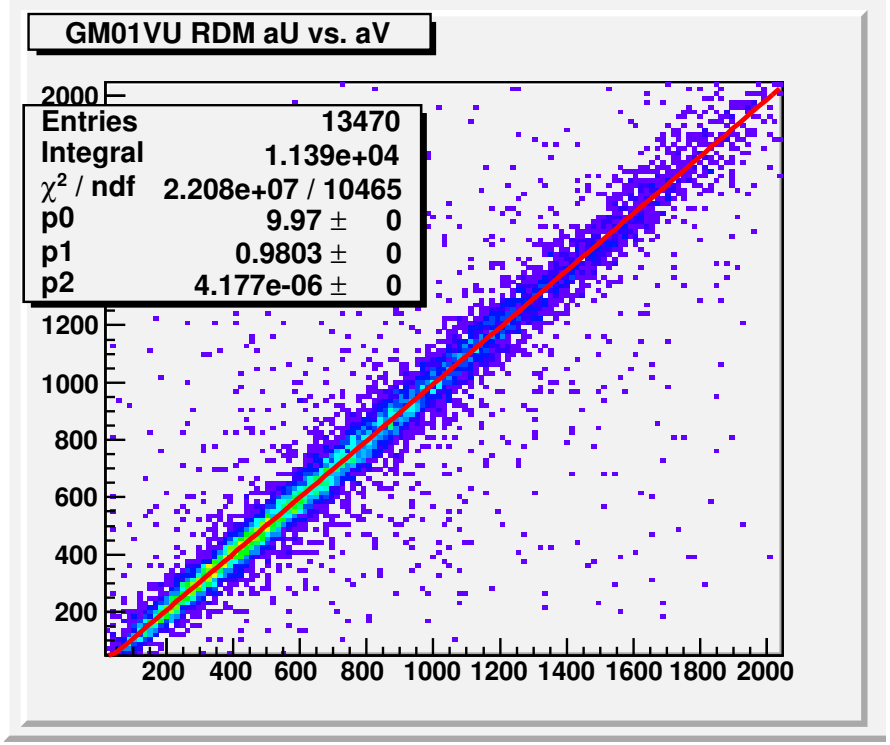
    A `cmake` command line option allows to install one or the other.
  - It's modular: Can easily be changed, **piece-wise**, any time.

  $\Rightarrow$ Could now be used to **evaluate accurate rates** and **MPGD impact on tracking**.

Why so complicated? Correlation

- **Amplitude/Time Correlation**
  - Can be **exploited in Tracking**.
  - As an **example**: **Amplitude** correlation obtained in **COMPASS GEMs** (*after some calibration*):





- **Correlated Smearing** could be applied on **SubHits, bypassing Coalescence**. . .
  . . . But **which SubHits?** is precisely the complicated piece.

- Over-complicated? All is done to be on the safe side.

  E.g. Care to not coalesce SubHits from looping particles, exiting and re-entering.

SubHits Distribution

- Rr = Reduced radius = local to Reference SubVolume.
- EDep = Sum of SimHit[0] + SimHit[1].