Off-line Electron Seeding Validation - Update

Caleb Fangmeier
Ilya Kravchenko, Greg Snow

University of Nebraska - Lincoln

Joint ECAL/EGM Meeting — December 13, 2017

Introduction

- Our goal is to study **seeding** for the **off-line** GSF tracking with the **new pixel detector**.
- Specifically, we want to optimize the new pixel-matching scheme from HLT for use in off-line reconstruction.
- Since last update[1],
  - Created sets of nTuples to compare/contrast seeding with new/old scheme.
  - Dataset:

    `/ZToEE_NNPDF30_13TeV-powheg_M_120_200/`

    `RunIISummer17DRStdmix-NZSFlatPU28to62_92X_upgrade2017_realistic_v10-v1/GEN-SIM-RAW`
  - Ntuples on Nebraska T2 (happy to share with interested parties!)
- This Talk:
  - Show performance comparisons between new and old seeding schemes
  - Show correlations between performance and detector geometry
  - Next steps

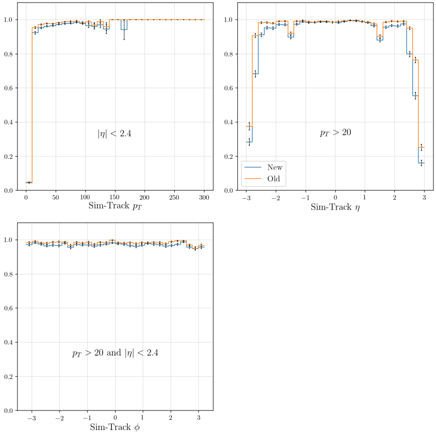
[1] https://indico.cern.ch/event/662751/contributions/2778076/attachments/1562070/2460731/main.pdf

First, some definitions

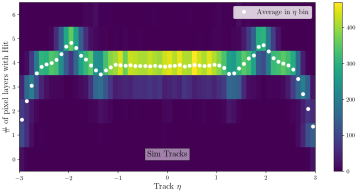
- **Sim-Track -** A track from a simulated electron originating from the luminous region of CMS (beam-spot +- 5σ)
- **ECAL-Driven Seed -** A seed created via a matching procedure between Super-Clusters and General Tracking Seeds (Either from `ElectronSeedProducer` or `ElectronNHitSeedProducer`)
- **GSF Track -** A track from GSF-Tracking resulting from an **ECAL-Driven Seed**
- **Seeding Efficiency -** The fraction of **Sim-Tracks** that have a matching **ECAL-Driven Seed** (based on simhit-rechit linkage)
- **GSF Tracking Efficiency -** The fraction of **Sim-Tracks** that have a matching **GSF Track** (again, based on simhit-rechit linkage)
- **ECAL-Driven Seed Purity -** The fraction of **ECAL-Driven Seeds** that have a matching **Sim-Track**
- **GSF Tracking Purity -** The fraction of **GSF Tracks** that have a matching **Sim-Track**

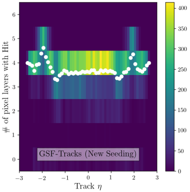
ECAL-Driven Seeding Efficiency

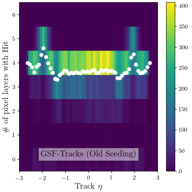
- In general, performance is similar between old and new seeding scheme
- Some early drop-off in efficiency at high eta
- Note the drop in efficiency around $\eta \approx 1.4$. (see next slide)



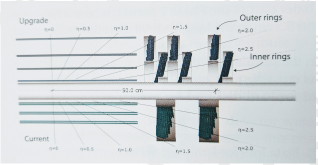
Number of Pixel Layers vs. η





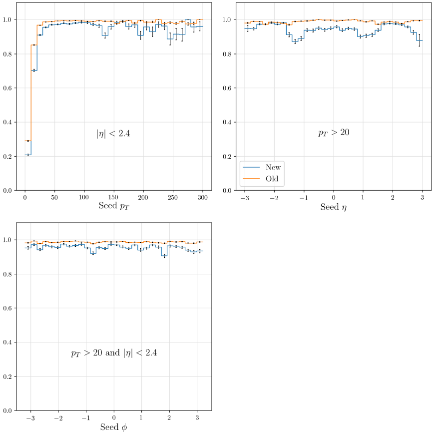


- Expected number of layers with hits is flat just under 4 for $|\eta| < 1.2$, but
- Drops significantly at the boundary between BPIX and FPIX
- However, at $|\eta| = 2$, it peaks since the track could pass through BPIX L1-L2 *and* FPIX L1-L3.



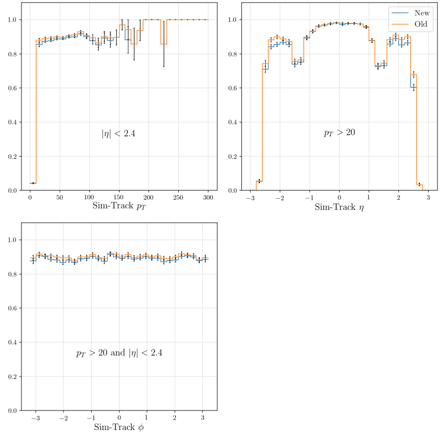
ECAL-Driven Seeding Purity

- Similar performance in forward region, but new seeding suffers from low purity in the barrel, and especially in the transition region
- Kinematic quantities here are from the seeds (based on some basic fitting), so likely worse resolution than from the GSF Tracks.



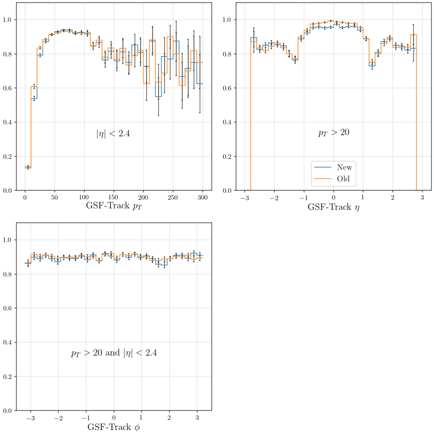
GSF Tracking Efficiency

- Again, similar performance between seeding strategies, although new is slightly worse
- Note that both strategies share a performance dip in the BPIX-FPIX transition region



GSF Tracking Purity

- Similar performance, *but*
- Strangely, it seems that the purity of the GSF-Tracks is worse than the ECAL-Driven Seeds that produced them!
- Which doesn't seem right... Needs further investigation.



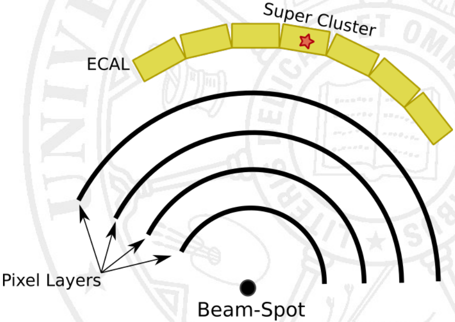
Outlook

- Targets for immediate investigation
  - Sources of impurity in ECAL-Driven Seeds and GSF-Tracks (Pile-up? Conversions? Will be relatively straight-forward w/ truth info)
  - Reasons for GSF-Tracks being less pure than their associated ECAL-Driven Hits
  - Ensure that the simhit-rechit matching procedure isn't biasing these results based on the number of available hits
- After that
  - Determine method to optimize window sizing, trying to improve, ideally, both tracking efficiency and purity (Not so easy. Many knobs to adjust!)
  - Suggestions?

BACKUP

Triplet Electron Seeding - Setup

- Begin with ECAL super cluster and beam spot

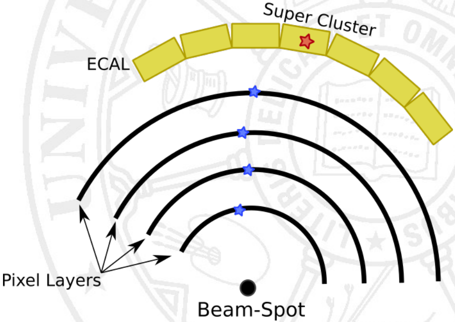


Triplet Electron Seeding - Introduce Seed

- Now, examine, one-by-one seeds from general tracking*
- Note that we do not look at all hits in an event, but rather rely on general tracking to identify seeds.

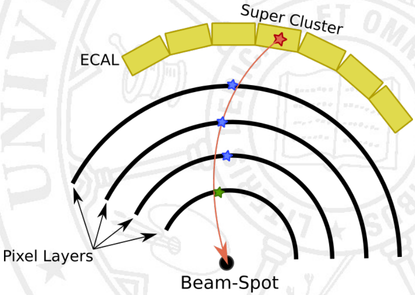
---

*initialStepSeeds, highPtTripletStepSeeds, mixedTripletStepSeeds, pixelLessStepSeeds, tripletElectronSeeds, pixelPairElectronSeeds, stripPairElectronSeeds



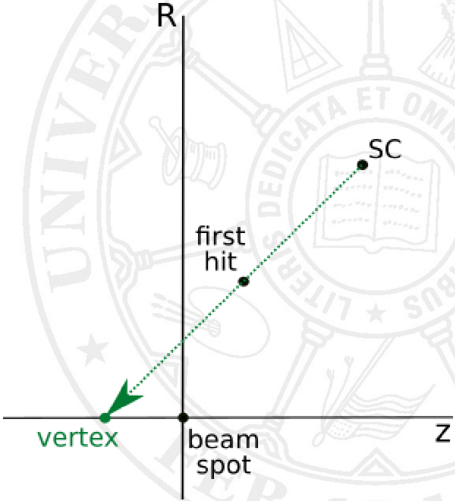
Triplet Electron Seeding - Match First Hit

- Using the beam spot, the SC position, and SC energy, propagate a path through the pixels.
- Next, require the first hit to be within a $\delta\phi$ and δz window. ($\delta\phi$ and δR for FPIX)
- δz window for first hit is huge as SC and beam spot positions give very little information about z.



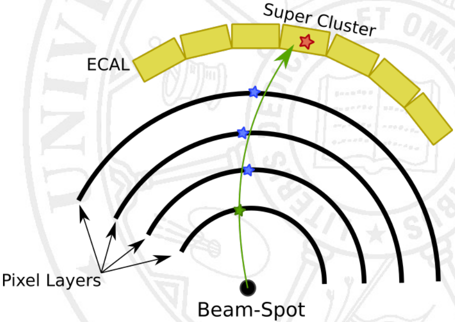
Triplet Electron Seeding - Extrapolate Vertex

- Once we have a matched hit, use it with the SC position, to find the vertex z.
- Vertex x and y are still the beam spot's.
- Just a simple linear extrapolation.



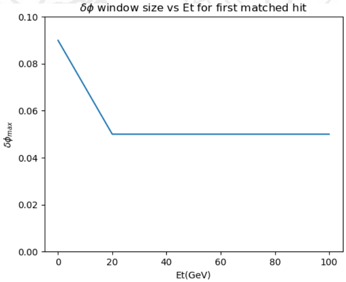
Triplet Electron Seeding - Match Other Hits

- Now forget the SC position, and propagate a new track based on the vertex and first hit positions, and the SC energy.
- Progress one-by-one through the remaining hits in the seed and require each one fit within a specified window around the track.
- Quit when all hits are matched, or a hit falls outside the window. No skipping is allowed.
- However, *layer* skipping is not ruled out if the original seed is missing a hit in a layer



Triplet Electron Seeding - Window Sizes

- The $\delta\phi$ and $\delta R/z$ windows for each hit are defined using three parameters.
  - `highEt`
  - `highEtThreshold`
  - `lowEtGradient`
- From these, the window size is calculated as $\texttt{highEt} + \min(0, \texttt{Et} - \texttt{highEtThreshold}) * \texttt{lowEtGradient}$.
- For the first hit, these parameters for the $\delta\phi$ window are,
  - $\texttt{highEt} = 0.05$
  - $\texttt{highEtThreshold} = 20$
  - $\texttt{lowEtGradient} = -0.002$
- For the first hit, these parameters for the $\delta\phi$ window are,

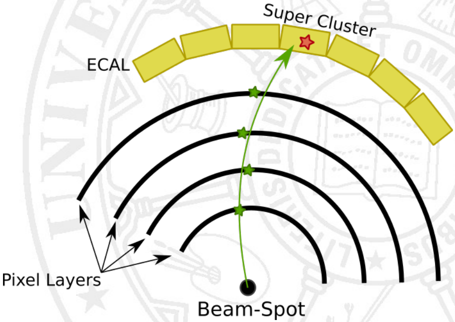


These parameters can be specified for each successive hit, and in bins of η, so optimization is a challenge!

Triplet Electron Seeding - Handle Missing Hits

- Finally, calculate the expected number of hits based on the number of working pixel modules the track passes through.
- Require exact[1] number of matched hits depending on the expected number of hits.
  - If $N_{\text{exp}} = 4$, require $N_{\text{match}} = 3$
  - If $N_{\text{exp}} < 4$, require $N_{\text{match}} = 2$
- If the seed passes all requirements, all information, including
  - Super cluster
  - Original Seed
  - Residuals (For both charge hypotheses)

  are wrapped up and sent downstream to GSF tracking



[1] Exact, rather than minimum to deal with duplicate seeds in input collection. Could switch to minimum with offline cross-cleaned seeds.