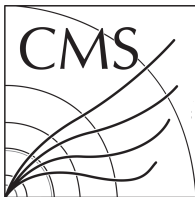


Off-line Electron Seeding Validation - Update

Caleb Fangmeier
Ilya Kravchenko, Greg Snow

University of Nebraska - Lincoln

EGamma Reco/Comm/HLT Meeting — February 16, 2018

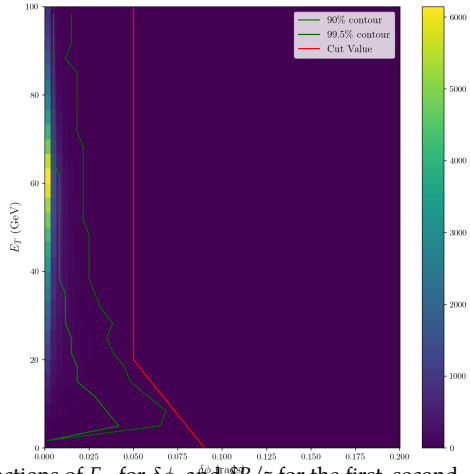


INTRODUCTION

- ▶ 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.
- ▶ This Talk:
 - ▶ Show the effect of linearly scaling matching windows up and down
 - ▶ Show first set of **optimized** windows
 - ▶ Next steps
- ▶ Full set of results is available here
https://eg.fangmeier.tech/seeding_studies_2018_02_15_18/output/

$\delta\phi$ RESIDUALS

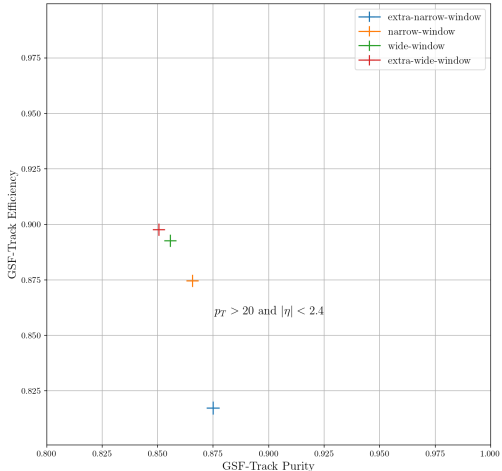
- ▶ Distribution of $\delta\phi$ residuals for first matched hits in truth-matched seeds where the hit was in BPIX-L1
- ▶ Truth-matching requires sufficient (75%) matched hits with a sim-track as well as less than 10% energy discrepancy between super-cluster and sim-track.
- ▶ Differential in E_T of the matched super-cluster
- ▶ Red line shows the default (aka HLT) window.
- ▶ Contour lines are biased by the matching cut necessarily being applied before deriving the contours.



Cut windows are specified as functions of E_T for $\delta\phi$, and $\delta R/z$ for the first, second, and third matched hits.

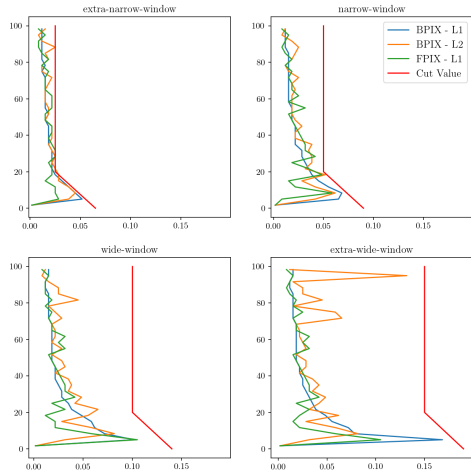
LINEAR SCALING OF WINDOWS

- ▶ Modified windows with uniform scaling
 - ▶ x0.5(extra-narrow)
 - ▶ x1.0(narrow)
 - ▶ x2.0(wide)
 - ▶ x3.0(extra-wide)
- ▶ Uniform scaling draws out a clear curve in Efficiency V. Purity.
- ▶ But can we do better?
Find windows with points above the curve?



FINDING MORE OPTIMAL WINDOWS (Ex. 1)

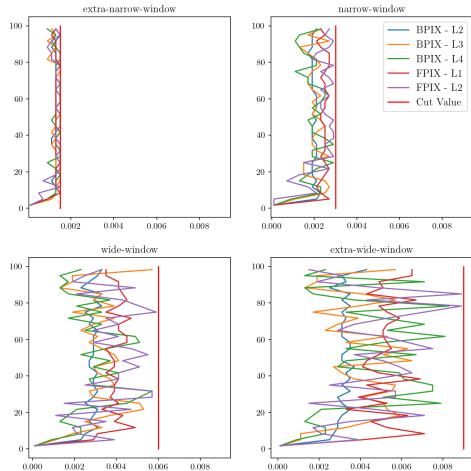
- Figure: first-hit $\delta\phi$ 99% contours for all relevant¹ pixel regions.
- Procedure: Select a cut that tends to reasonably follow the 99% contours in the extra-wide windows.
- Repeat this for each of the five windows.
- In this case, the narrow window seemed appropriate so this particular window was unchanged.



¹meaning the sub-detectors that have a substantial portion of first hits

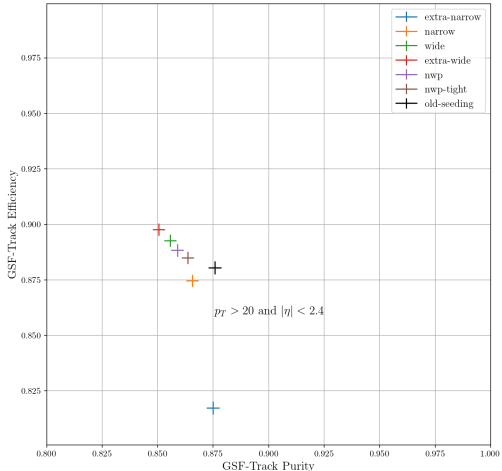
FINDING MORE OPTIMAL WINDOWS (Ex. 2)

- ▶ Figure: second-hit $\delta\phi$ 99% contours for all relevant pixel regions.
- ▶ Quite low statistics in some regions + looking at tails of distribution results in high variability
- ▶ Despite this, estimate an appropriate cut to be 0.005



PROPOSED NEW WORKING POINT PERFORMANCE

- ▶ New Working Point lies basically on the linear-scaling curve
- ▶ However, NWP with extra-narrow first $\delta\phi$ window sets slightly above the curve
- ▶ Hints that better performance is achievable, but it's not obvious how to achieve
- ▶ Many ways to vary parameters...



OUTLOOK

- ▶ Next steps
 - ▶ Testing with an complementary dataset (currently looking at $Z \rightarrow ee$ only)
 - ▶ Possibly breaking down windows sizes in η (code supports this, but is currently unused).
- ▶ Other Thoughts
 - ▶ What is an appropriate working point, and what performance can be deemed adequate?
 - ▶ Are there different figures-of-merit that must be balanced (CPU performance, specific background rejections.)?

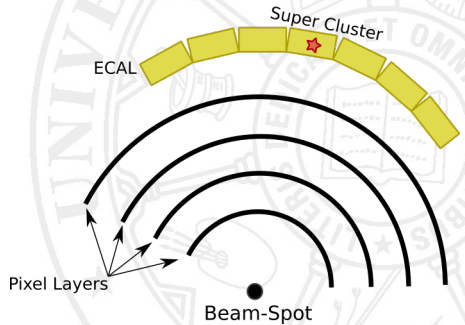
BACKUP



- ▶ **Sim-Track** - A track from a simulated electron originating from the luminous region of CMS (beam-spot $\pm 5\sigma$)
- ▶ **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**

TRIPLET ELECTRON SEEDING - SETUP

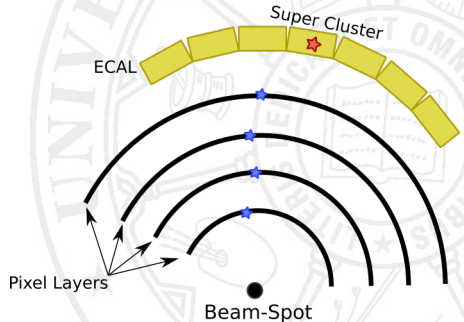
- Begin with ECAL super cluster and beam spot



TRIPLET ELECTRON SEEDING - INTRODUCE SEED

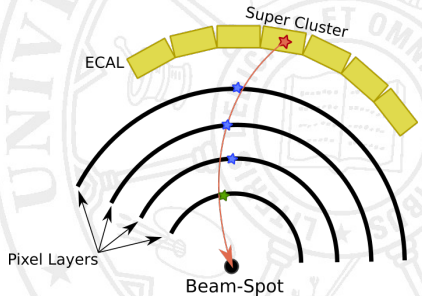
- ▶ Now, examine, one-by-one seeds from general tracking*
- ▶ Note that we do not look at all hits in an event, but rather rely on general tracking to identify seeds.

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



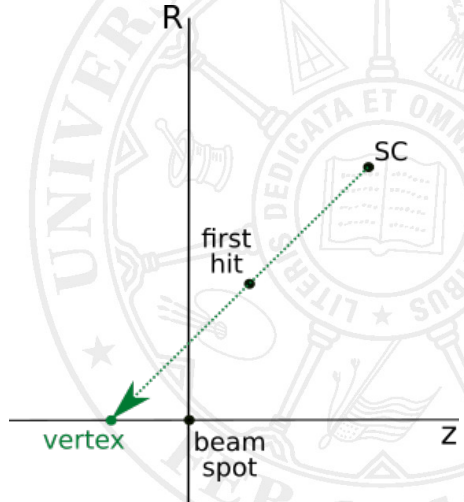
TRIPLET ELECTRON SEEDING - MATCH FIRST HIT

- ▶ 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 .



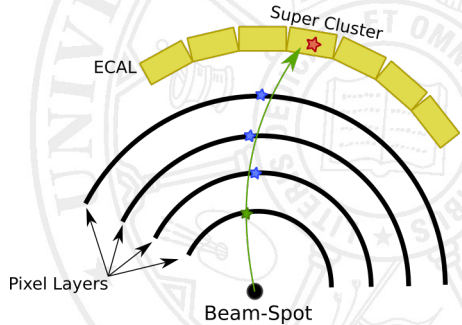
TRIPLET ELECTRON SEEDING - EXTRAPOLATE VERTEX

- ▶ 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.



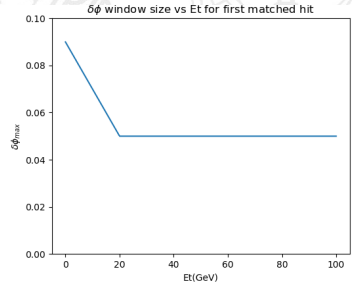
TRIPLET ELECTRON SEEDING - MATCH OTHER HITS

- ▶ 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
`highEt + min(0, Et - highEtThreshold) * lowEtGradient`.
- ▶ For the first hit, these parameters for the $\delta\phi$ window are,
 - ▶ `highEt = 0.05`
 - ▶ `highEtThreshold = 20`
 - ▶ `lowEtGradient = -0.002`

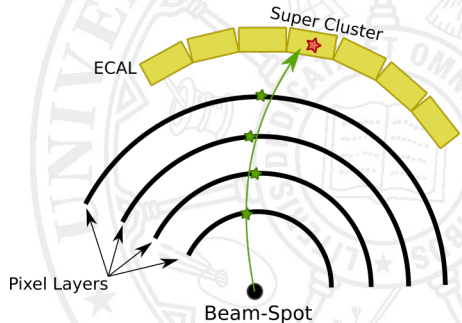


These parameters can be specified for each successive hit, and in bins of η , so optimization is a challenge!

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 exact¹ 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



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