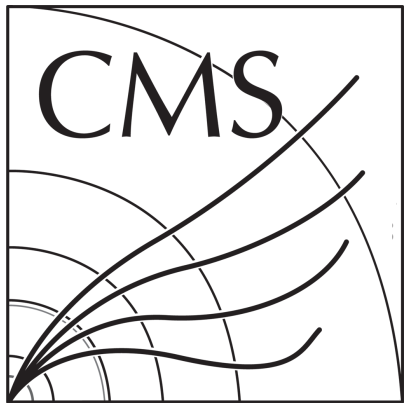
Offline Electron Seeding Validation - Update

Caleb Fangmeier
Ilya Kravchenko, Greg Snow

University of Nebraska - Lincoln

EGM General Meeting — November 19, 2018



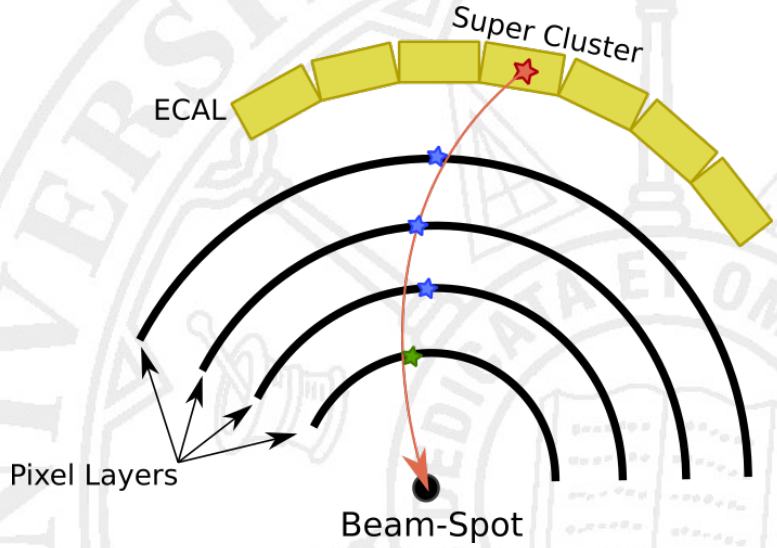
Introduction

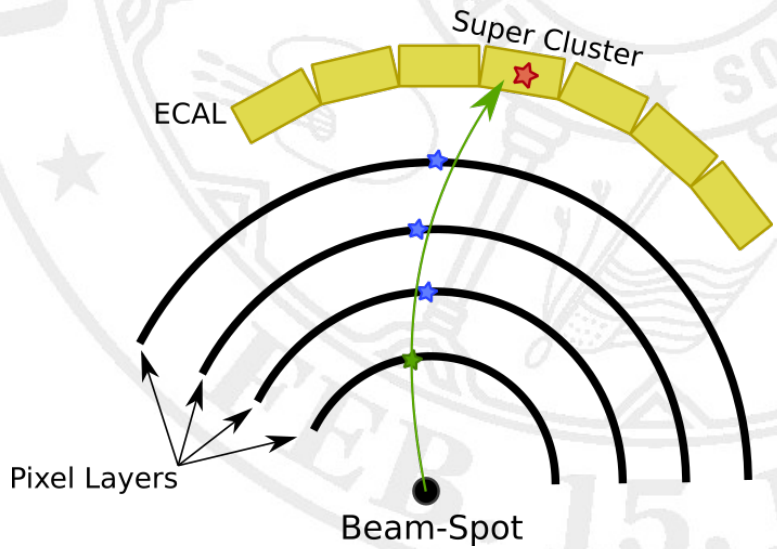
- Our goal is to study **seeding** for the **offline** GSF tracking with the **Phase I pixel detector**.
- Specifically, we want to optimize the new pixel-matching scheme from HLT for use in off-line reconstruction.
- Previous presentation[1] showed efficiency/purity/fake-rate for proposed offline electron seeding working points.
- This Talk:
  - Explain "Hit Skipping" and demonstrate effects on seeding performance
  - Examine effects of adding pileup on seeding performance.

[1] https://indico.cern.ch/event/697084/#2-update-on-offline-electron-s

N-Hit Electron Seeding

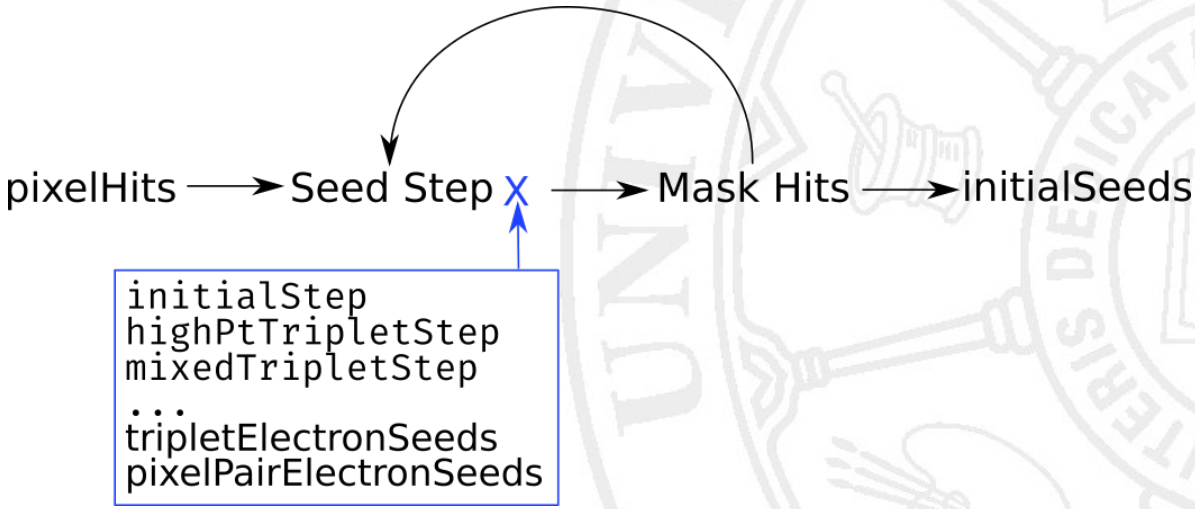
1. Using the beam spot, the SC position, and SC energy, propagate a path through the pixels.
2. Require the first hit to be within a $\delta\phi$ and δz window. ($\delta\phi$ and δR for FPIX)
3. δz window for first hit is huge as SC and beam spot positions give very little information about z.
4. Forget the SC position, and propagate a new track based on the vertex and first hit positions, and the SC energy.
5. Progress one-by-one through the remaining hits in the seed and require each one fit within a specified window around the track.
6. Quit when all hits are matched, or a hit falls outside the window. **No skipping is allowed.**





Hit-Skipping

- Normally, general tracking seeds are made with an iterative procedure
- Each iteration masks hits from use in subsequent steps
- Reduces combinatorics for CKF tracking

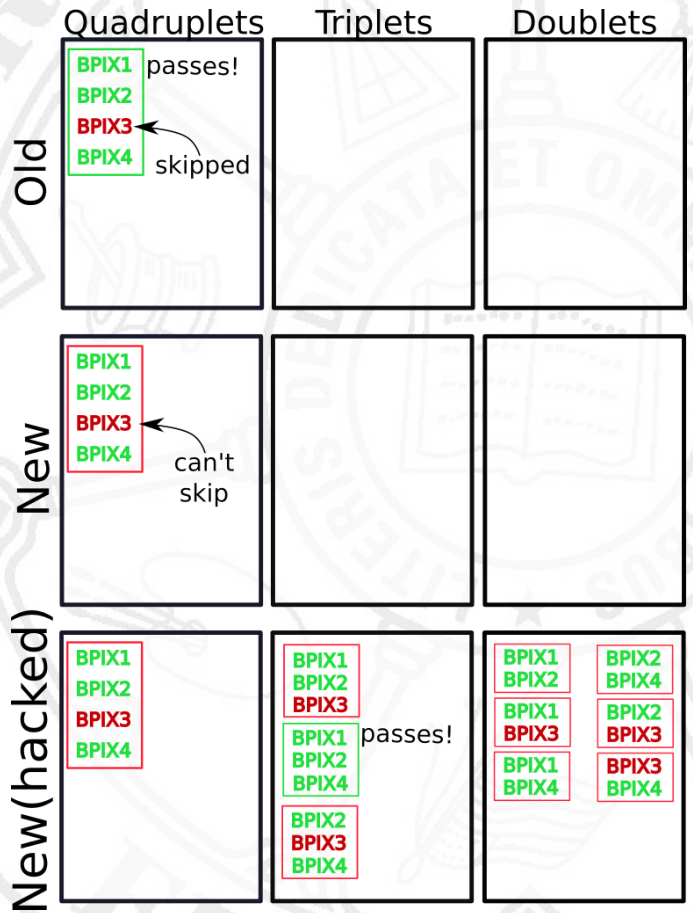


- Each iteration works on a specific set of combinations of layers. eg. `initialStep` uses:

```
BPix1+BPix2+BPix3+BPix4
BPix1+BPix2+BPix3+FPix1_pos
BPix1+BPix2+BPix3+FPix1_neg
BPix1+BPix2+FPix1_pos+FPix2_pos
BPix1+BPix2+FPix1_neg+FPix2_neg
BPix1+FPix1_pos+FPix2_pos+FPix3_pos
BPix1+FPix1_neg+FPix2_neg+FPix3_neg
```

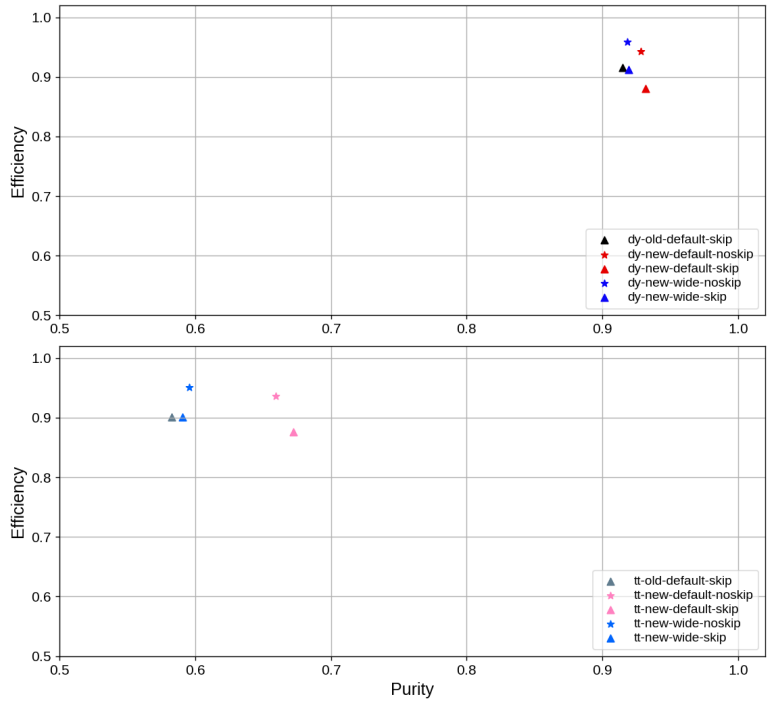
Hit-Skipping

- When `NHitElectronSeedProducer` was implemented for HLT, hit skipping was not added.
- Consider an example configuration where we are generating first quadruplet, then triplet, and then finally doublet seeds, masking hits along the way.
- If we require at least 3 matched hits, the old method *with* hit skipping would create a seed of hits `BPIX1, BPIX2, BPIX4`.
- But new method *without* hit skipping wouldn't make any seed from these hits.
- The "hack" is to create seeds using only steps `tripletElectronSeeds`, and `pixelPairElectronSeeds` with **no masking**.
- Adding skipping and removing the hack would reduce cpu time from redundant seeds.



Hit-Skipping - Performance

- Enabling hit skipping and removing hack reduces number of seeds by 36% to 51%.
- 3-5x fewer seeds with respect to old seeding
- Efficiency reduced by between 4% and 6% to align more with old seeding performance.
- Purity improved by about 1%.
- (table in backup)



Process	Seeding Method	$< N_{seeds} >$(no-skip)	$< N_{seeds} >$(with-skip)	Percent Reduction
$t\bar{t}$	Old - default settings	-	12.69	-
	New - HLT settings	4.40	2.56	41%
	New - "wide" settings	7.28	4.65	36%
Drell-Yan	Old - default settings	-	11.40	-
	New - HLT settings	4.70	2.32	51%
	New - "wide" settings	5.38	2.65	51%

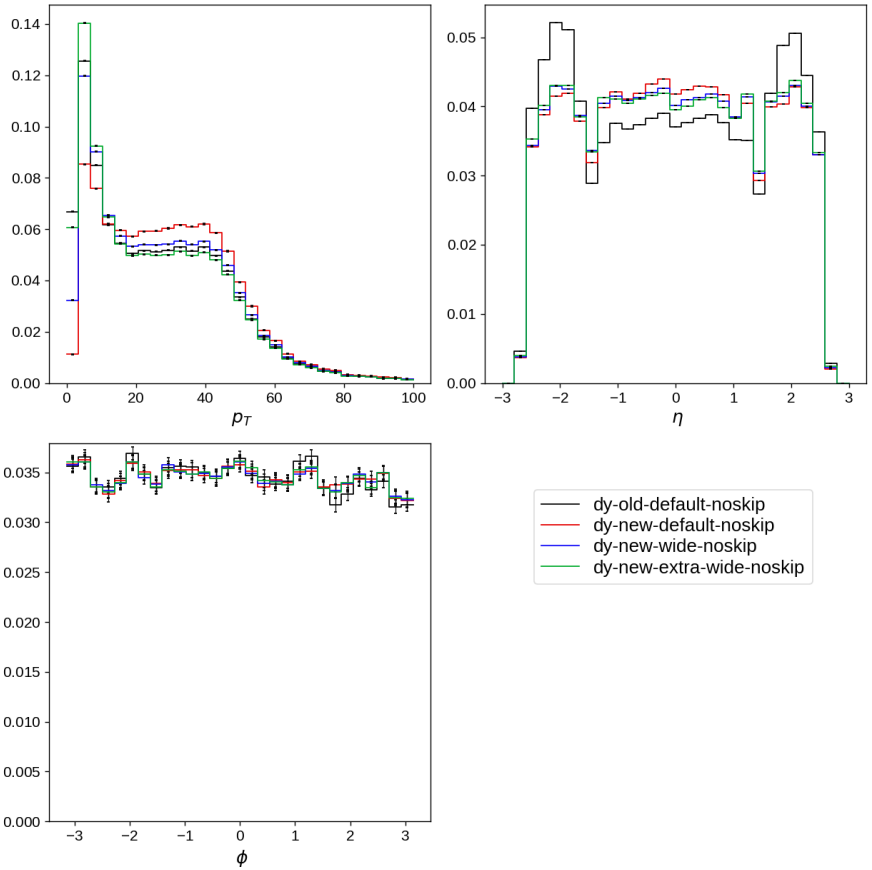
Adding Pileup

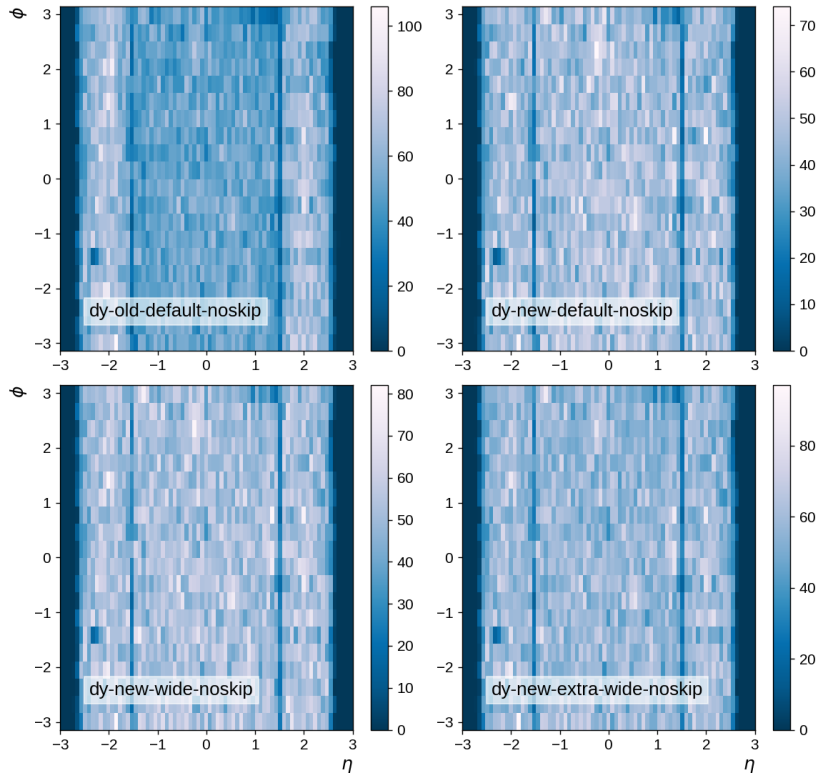
- The simhit-rechit linkage that was previously used in efficiency/purity measurements is not saved in `GEN-SIM-RAW`.
- Therefore, the `DIGI` step was re-run, but *only for the signal event* because `GEN-SIM-RAW` does not contain `SIM` information for pileup events.
- However, running this instead of the `RAW2DIGI` step discarded the previously mixed pileup in the `RAW`.
- So even though there is a `PileupInfo` collection with reasonable values, there's no actual pileup hits being used for tracking (caused quite some confusion for me).
- In the end, abandon simhit-rechit linkage and just use ΔR matching for efficiency/purity.
- Some technical details in backup.

Adding Pileup - Issues

- Creating kinematic distributions raises some apparent issues with how the new seeding handles pileup.
- Next three slides show pt/eta/phi distributions of GSF tracks resulting from ECAL-Driven seeds

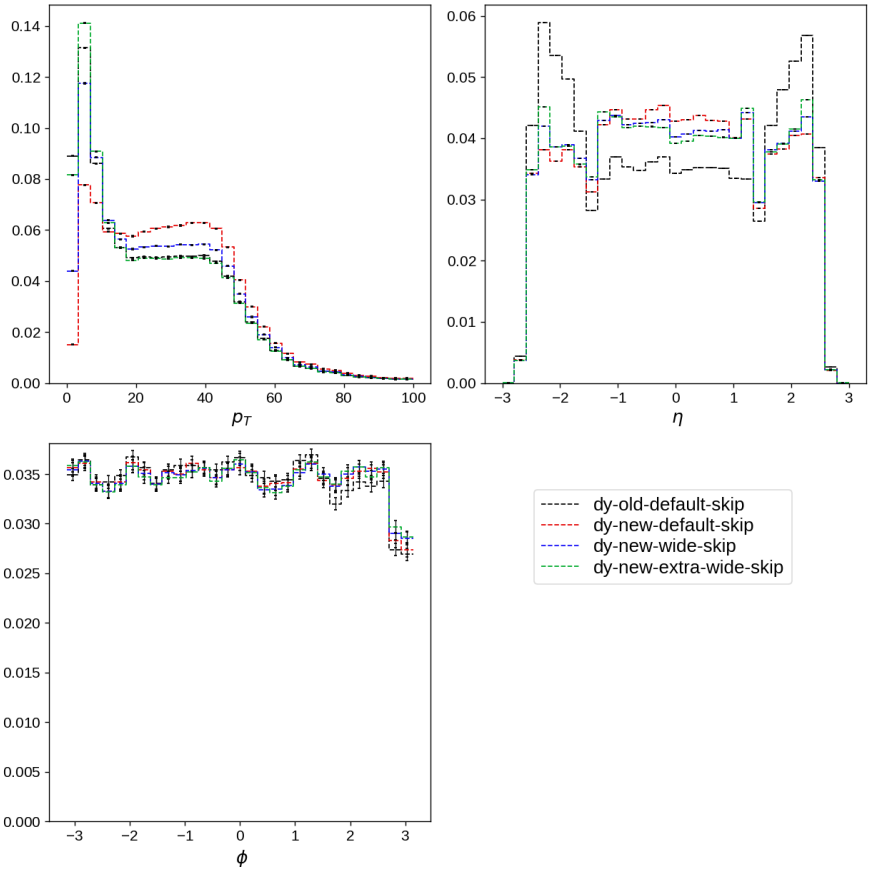
Adding Pileup - Issues - No Skipping, No Pileup

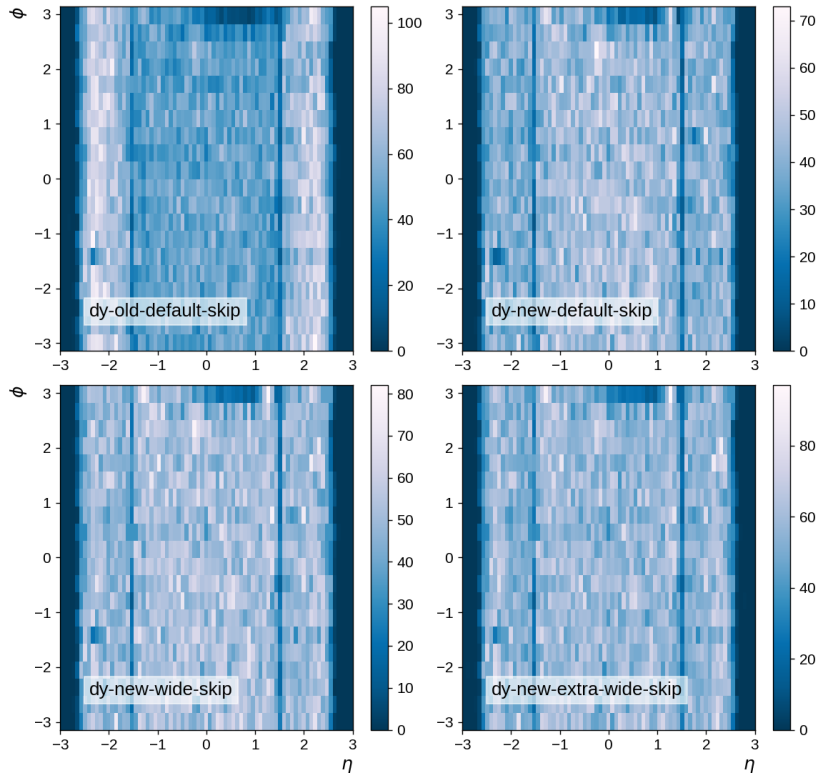




- Looks basically ok, use this as a baseline for comparison.

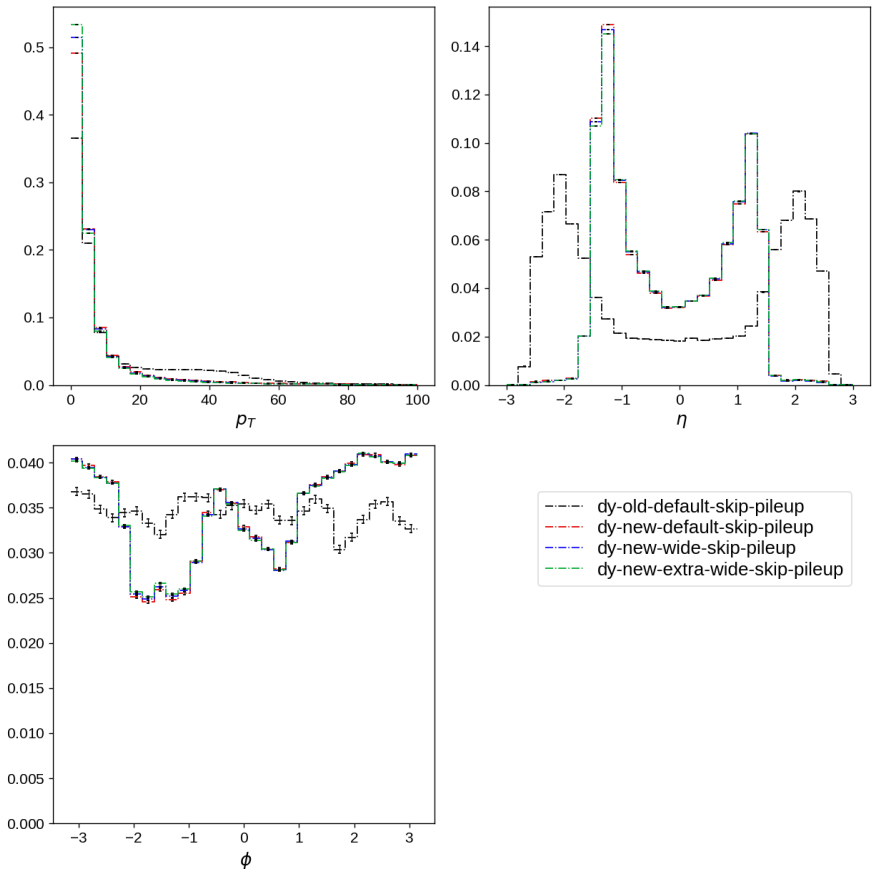
Adding Pileup - Issues - With Skipping, No Pileup

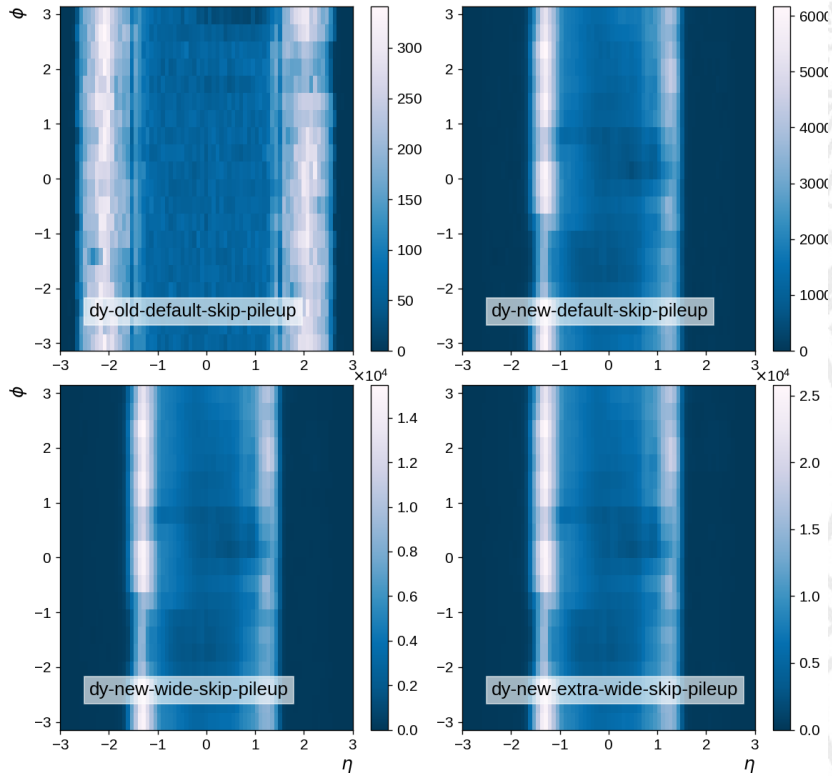




- Concerning dip around $\phi = 3$ coming from $\eta \in (0.5, 1.5)$.

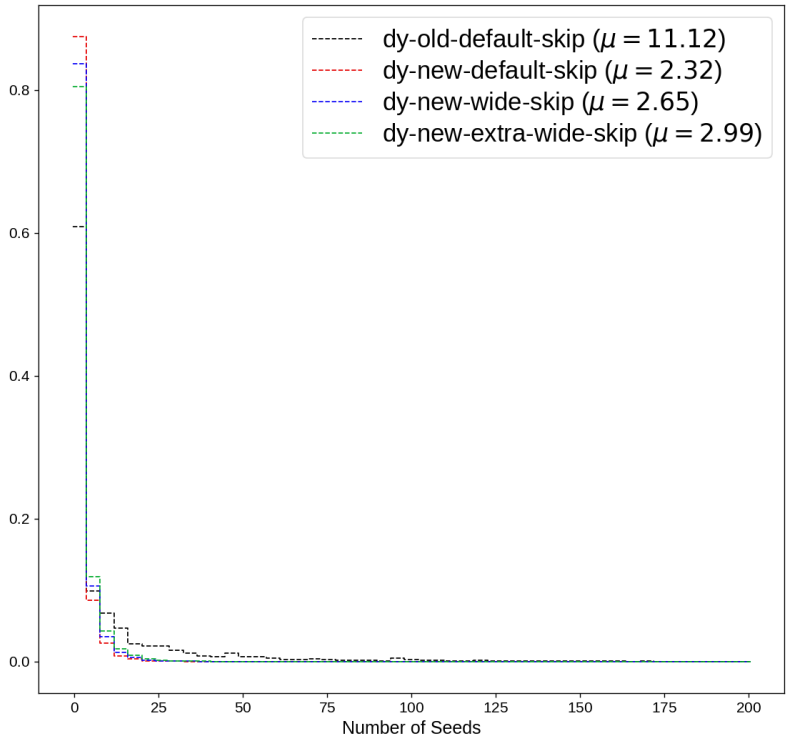
Adding Pileup - Issues - With Skipping, With Pileup

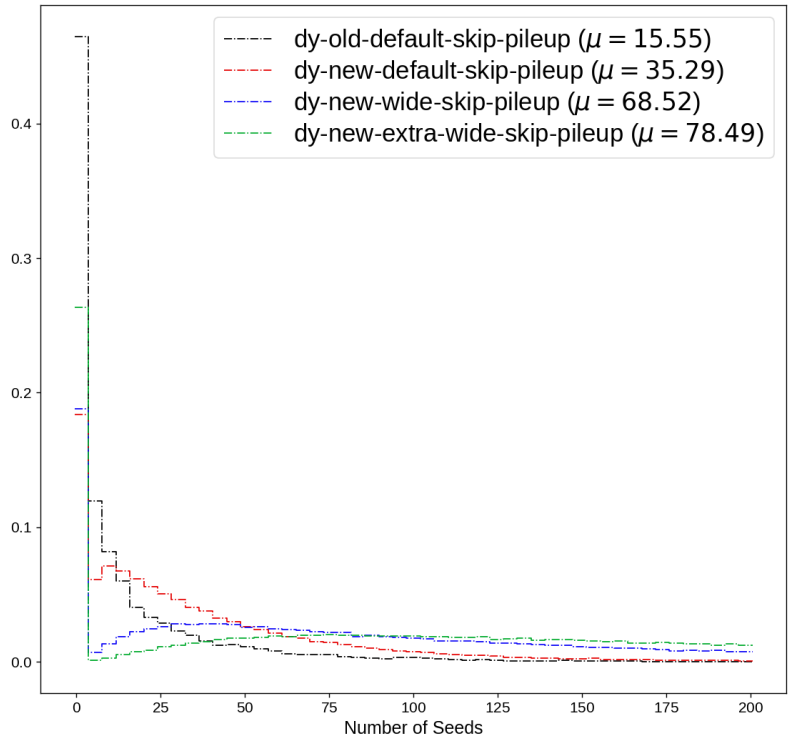




- Strangely non-flat ϕ distribution, manifests differently in both old and new seeding, but rather more pronounced in new.
- Features seem to be somewhat localized in $\phi - \eta$, possibly some kind of detector effect?

Adding Pileup - More Issues





- Relative reduction of number of seeds gone. In fact, with pileup there is a 2.3 to 5 times increase in the number of seeds relative to old seeding method.

Conclusions & Outlook

- Reintroducing hit skipping in new seeding is implemented and has expected results.
- Analyzing MC with pileup mixed highlights potential issues with using new seeding for offline reconstruction.
- Further investigation is necessary to determine the source of the issues.
- Any ideas for checks or fixes are welcome!

Analysis and ploting code is available at https://git.fangmeier.tech/caleb/EGamma_ElectronTrackingValidation

Additional plots are available at http://t3.unl.edu/~cfangmeier/eg/seeding_studies_2018_11_18_19/hists.html

BACKUP

Definitions

- **Sim-Track -** A track from a simulated electron both originating from the luminous region of CMS (beam-spot +- 5σ) and having $|\eta| < 3.0$.
- **ECAL-Driven Seed -** A seed created via a matching procedure between Super-Clusters and General Tracking Seeds (Either from `ElectronSeedProducer` or `ElectronNHitSeedProducer`). Must have $HOE < 0.15$.
- **GSF Track -** A track from GSF-Tracking resulting from an **ECAL-Driven Seed**
- **GSF Tracking Efficiency -** The fraction of **Sim-Tracks** that have a matching **GSF Track** (based on ΔR matching)
- **GSF Tracking Purity -** The fraction of **GSF Tracks** that have a matching **Sim-Track**
- **GSF Tracking Fake Rate -** The fraction of nontruth-matched Super-Clusters which result in at least one **GSF Track**.

Seeding Performance

Sample	Working Point	Config	Efficiency	Purity	Fake Rate
Drell-Yan	old-default	skip	$91.60 \pm 0.51\%$	$91.50 \pm 0.52\%$	$57.62 \pm 1.23\%$
	old-default	skip-pileup	$90.94 \pm 0.51\%$	$79.75 \pm 0.49\%$	$59.19 \pm 1.26\%$
	new-default	noskip	$94.32 \pm 0.51\%$	$92.82 \pm 0.52\%$	$39.46 \pm 1.23\%$
	new-default	skip	$88.01 \pm 0.51\%$	$93.18 \pm 0.54\%$	$35.63 \pm 1.23\%$
	new-default	skip-pileup	$88.15 \pm 0.51\%$	$18.35 \pm 0.24\%$	$92.48 \pm 1.26\%$
	new-wide	noskip	$95.90 \pm 0.51\%$	$91.82 \pm 0.51\%$	$49.83 \pm 1.23\%$
	new-wide	skip	$91.23 \pm 0.51\%$	$91.94 \pm 0.52\%$	$47.24 \pm 1.23\%$
	new-wide	skip-pileup	$92.81 \pm 0.51\%$	$8.97 \pm 0.16\%$	$99.12 \pm 1.26\%$
	new-extra-wide	noskip	$96.22 \pm 0.51\%$	$91.27 \pm 0.51\%$	$57.14 \pm 1.23\%$
	new-extra-wide	skip	$91.89 \pm 0.51\%$	$91.21 \pm 0.52\%$	$54.39 \pm 1.23\%$
	new-extra-wide	skip-pileup	$94.39 \pm 0.51\%$	$5.92 \pm 0.13\%$	$99.74 \pm 1.26\%$
$t\bar{t}$	old-default	skip	$90.13 \pm 0.77\%$	$58.25 \pm 0.62\%$	$53.85 \pm 0.54\%$
	old-default	skip-pileup	$89.08 \pm 0.77\%$	$45.61 \pm 0.55\%$	$56.34 \pm 0.59\%$
	new-default	noskip	$93.59 \pm 0.77\%$	$65.92 \pm 0.65\%$	$29.13 \pm 0.54\%$
	new-default	skip	$87.56 \pm 0.77\%$	$67.21 \pm 0.68\%$	$24.82 \pm 0.54\%$
	new-default	skip-pileup	$87.78 \pm 0.77\%$	$6.85 \pm 0.21\%$	$89.38 \pm 0.59\%$
	new-wide	noskip	$95.10 \pm 0.77\%$	$59.54 \pm 0.61\%$	$48.03 \pm 0.54\%$
	new-wide	skip	$90.12 \pm 0.77\%$	$59.04 \pm 0.62\%$	$44.73 \pm 0.54\%$
	new-wide	skip-pileup	$91.47 \pm 0.77\%$	$3.53 \pm 0.15\%$	$98.30 \pm 0.60\%$
	new-extra-wide	noskip	$95.41 \pm 0.77\%$	$56.63 \pm 0.59\%$	$60.23 \pm 0.54\%$
	new-extra-wide	skip	$90.66 \pm 0.77\%$	$55.02 \pm 0.60\%$	$58.61 \pm 0.54\%$
	new-extra-wide	skip-pileup	$92.66 \pm 0.78\%$	$2.52 \pm 0.12\%$	$99.35 \pm 0.60\%$

Matching Window Parameters

		narrow	default (HLT)	wide	extra-wide
Hit 1	dPhiMaxHighEt	**0.025**	**0.05**	**0.1**	**0.15**
	dPhiMaxHighEtThres	20.0	20.0	20.0	20.0
	dPhiMaxLowEtGrad	-0.002	-0.002	-0.002	-0.002
	dRzMaxHighEt	9999.0	9999.0	9999.0	9999.0
	dRzMaxHighEtThres	0.0	0.0	0.0	0.0
	dRzMaxLowEtGrad	0.0	0.0	0.0	0.0
Hit 2	dPhiMaxHighEt	**0.0015**	**0.003**	**0.006**	**0.009**
	dPhiMaxHighEtThres	0.0	0.0	0.0	0.0
	dPhiMaxLowEtGrad	0.0	0.0	0.0	0.0
	dRzMaxHighEt	**0.025**	**0.05**	**0.1**	**0.15**
	dRzMaxHighEtThres	30.0	30.0	30.0	30.0
	dRzMaxLowEtGrad	-0.002	-0.002	-0.002	-0.002
Hit 3+	dPhiMaxHighEt	**0.0015**	**0.003**	**0.006**	**0.009**
	dPhiMaxHighEtThres	0.0	0.0	0.0	0.0
	dPhiMaxLowEtGrad	0.0	0.0	0.0	0.0
	dRzMaxHighEt	**0.025**	**0.05**	**0.1**	**0.15**
	dRzMaxHighEtThres	30.0	30.0	30.0	30.0
	dRzMaxLowEtGrad	-0.002	-0.002	-0.002	-0.002

`NHit` Seeding window parameters. Bold designates modified values.

Samples

- /ZToEE_NNPDF30_13TeV-powheg_M_120_200/RunIISummer17DRStdmix-NZSFlatPU28to62_92X_upgrade2017_realistic_v10-v1
- /TT_TuneCUETP8M2T4_13TeV-powheg-pythia8/RunIISummer17DRStdmix-NZSFlatPU28to62_92X_upgrade2017_realistic_v10-v2

cmsDriver invocations

RAW→Step2 (old)

```
cmsDriver.py RAW2Step2 \
    --mc \
    --conditions 92X_upgrade2017_realistic_v10 \
    --era Run2_2017 \
    --eventcontent FEVTDEBUG \
    --datatier GEN-SIM-DIGI-RAW \
    --step DIGI:pdigi_valid,L1,DIGI2RAW \
    --geometry DB:Extended \
    --filein file:input.root \
    --fileout file:step2.root \
    --runUnscheduled
```

Step2→TrackingNtuple (old)

```
cmsDriver.py Step2ToTrackingNtuple \
    --mc \
    --conditions 92X_upgrade2017_realistic_v10 \
    --era Run2_2017 \
    --eventcontent RECOSIM,MINIAODSIM,DQM \
    --datatier GEN-SIM-RECO,MINIAODSIM,DQMIO \
    --step RAW2DIGI,L1Reco,RECO,EI,PAT,VALIDATION:@standardValidation+@miniAODValidation \
    --filein file:step2.root \
    --fileout file:trackingNtuple.root \
    --customise Validation/RecoTrack/customiseTrackingNtuple.customiseTrackingNtuple \
    --runUnscheduled
```

RAW→TrackingNtuple (new)

```
cmsDriver.py RAW2TrackingNtuple \
    --mc \
    --conditions 92X_upgrade2017_realistic_v10 \
    --era Run2_2017  \
    --eventcontent FEVTDEBUG \
    --datatier GEN-SIM-RECO \
    --step RAW2DIGI,RECO,EI,PAT,VALIDATION \
    --customise Validation/RecoTrack/customiseTracking
.customiseTrackingNtuple \
    --filein file:input.root \
    --fileout file:trackingNtuple.root \
    --runUnscheduled
```

Additionally, hacks to remove hit truth dependencies from `TrackingNtuple`