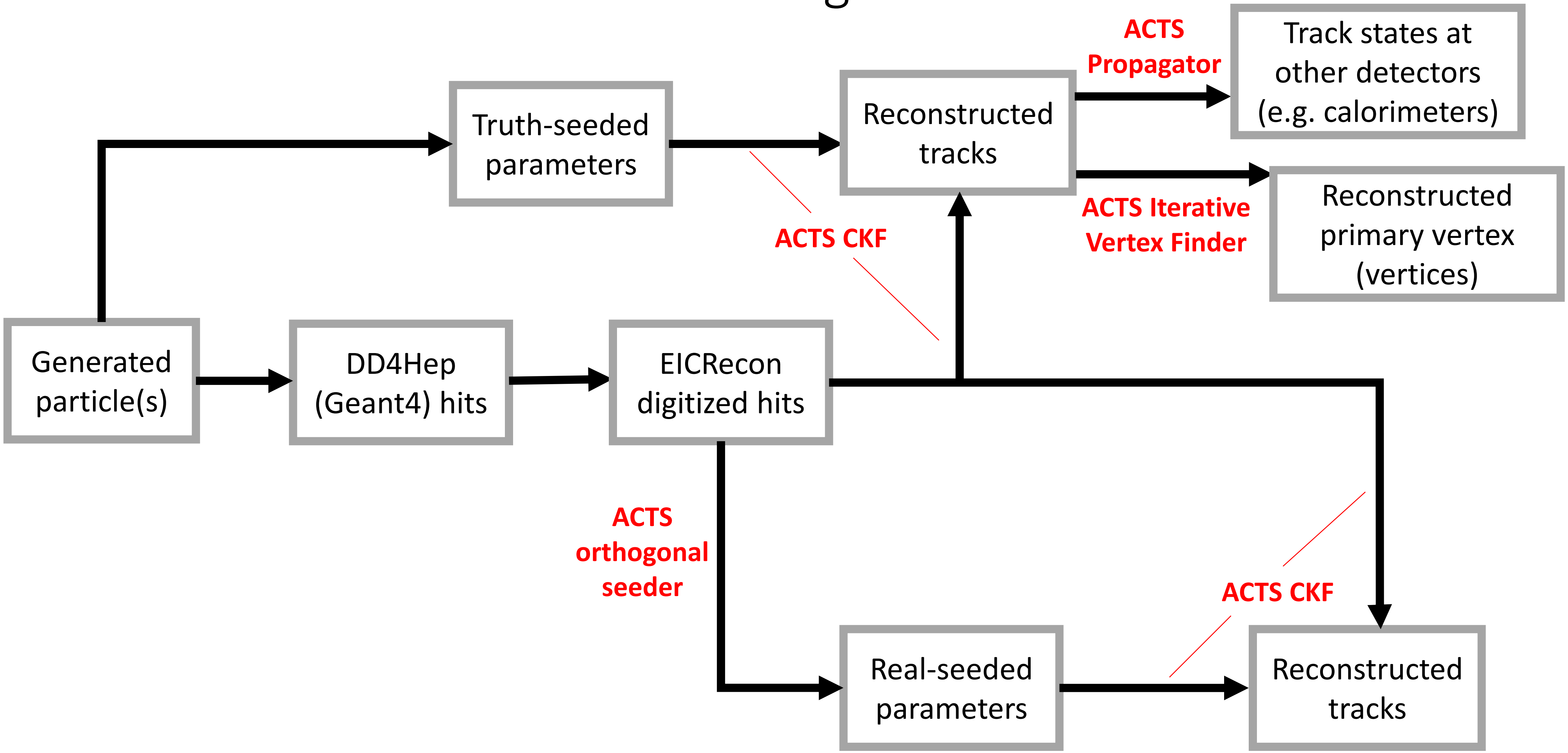
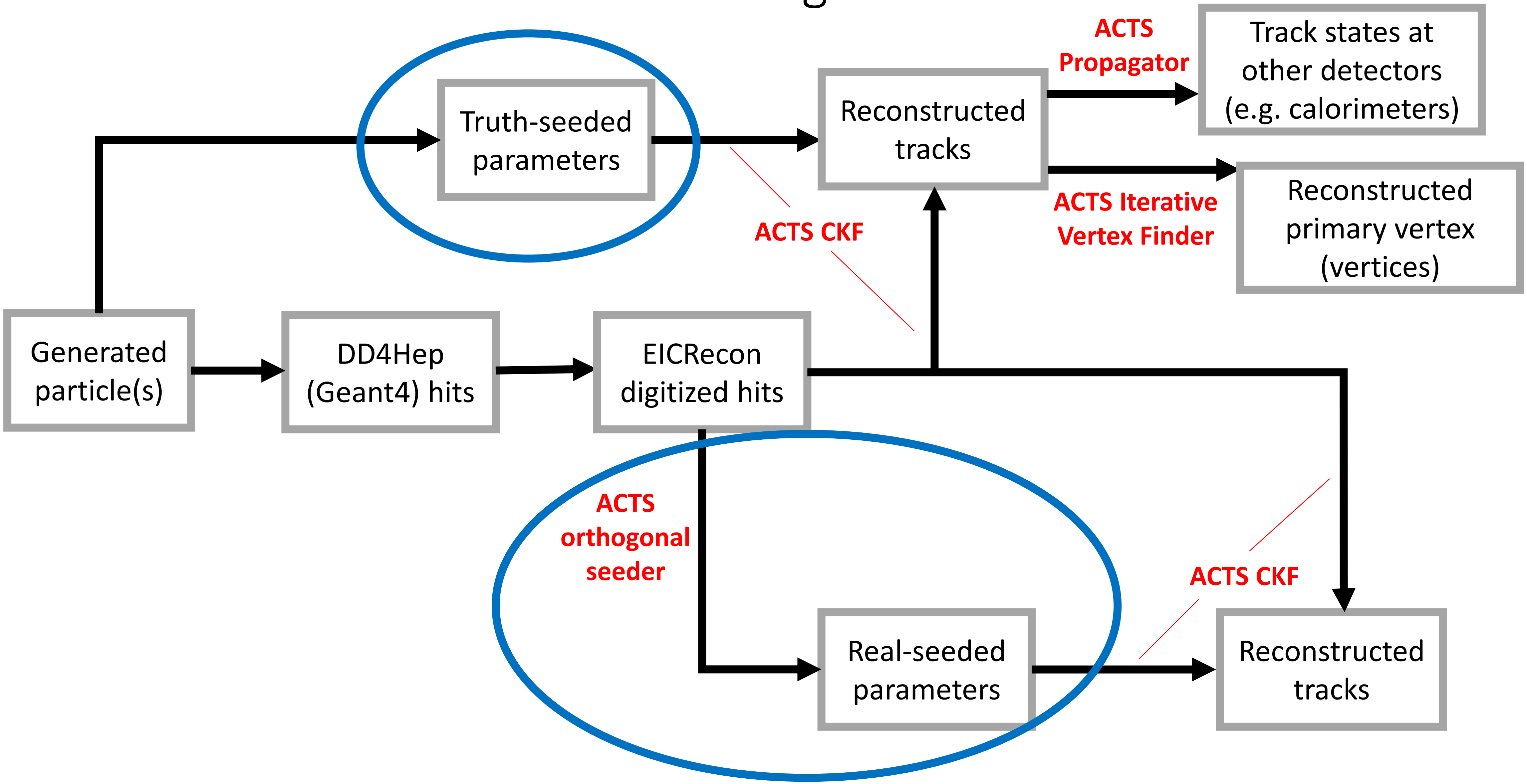
Track seeding for particles generated off the beamline (z axis)

Barak Schmookler

Tracking workflow



Tracking workflow



Seeding and tracking

Seeding implementations

Truth (Ideal) seeding: For every generated, final-state (i.e. status = 1) charged particle, we use the true charge, q/p, theta, phi, and generation vertex to form the seed. Option to smear the initial parameters is included.

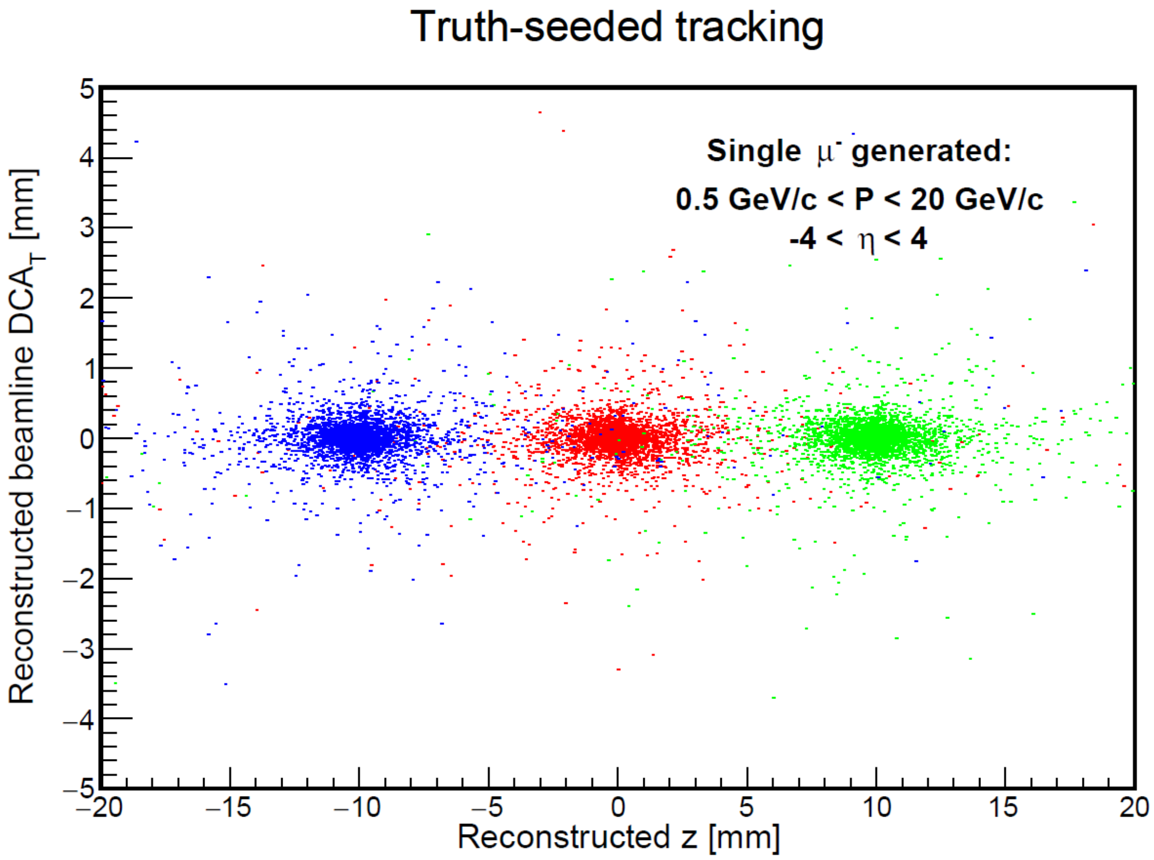
Real seeding: The ACTS orthogonal seeder outputs a set of seeds, with each seed consisting of 3 space points. The seeds need to fulfill certain expectations for a particle moving in a uniform magnetic field. The seed finder and seed filter settings configure the allowed search region and tolerances. We then fit the seed points to determine the charge, q/p, theta, phi, and the initial position coordinates.

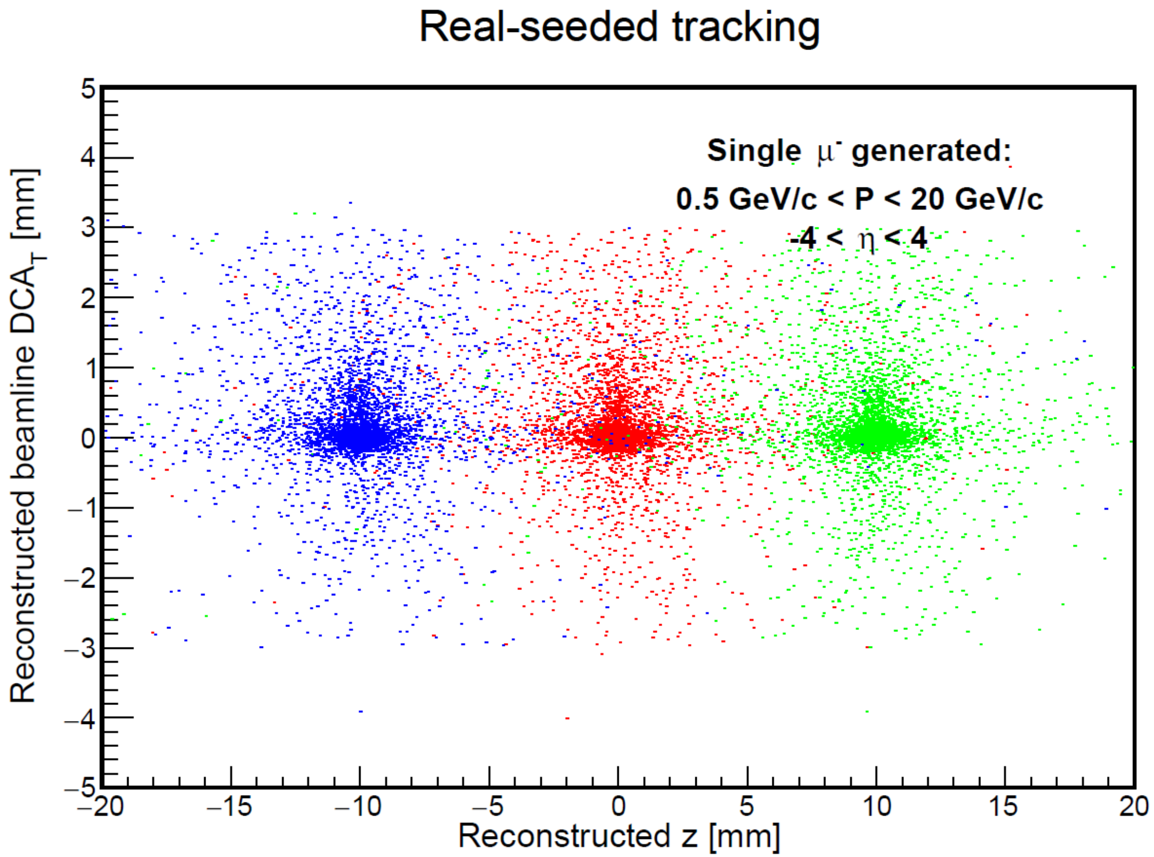
A given seed is then passed into the ACTS CKF for track finding and fitting. At the acceptance edges, the truth-seeded tracks can sometimes have fewer than 3 hits. For real seeding, we can have seed duplicates.

We don't currently have an implementation where we separate track finding and fitting. For example, we don't use Geant information to send the true hits for a given particle to a KF.

https://acts.readthedocs.io/en/latest/tracking.html
https://acts.readthedocs.io/en/latest/core/seeding.html#seeding-core

Single-particle reconstruction on beamline



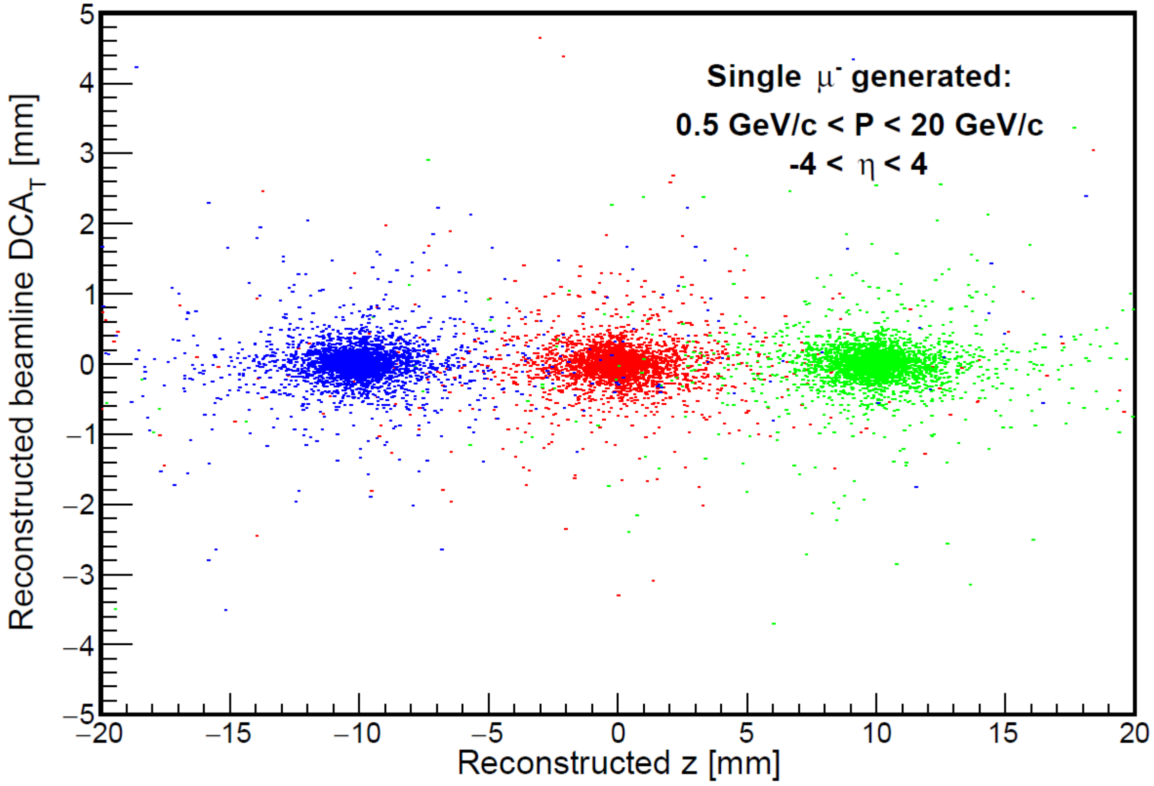


Reconstructed z: longitudinal impact parameter with respect to (0,0,0)
Reconstructed transverse DCA: (signed) transverse impact parameter with respect to (0,0,0)

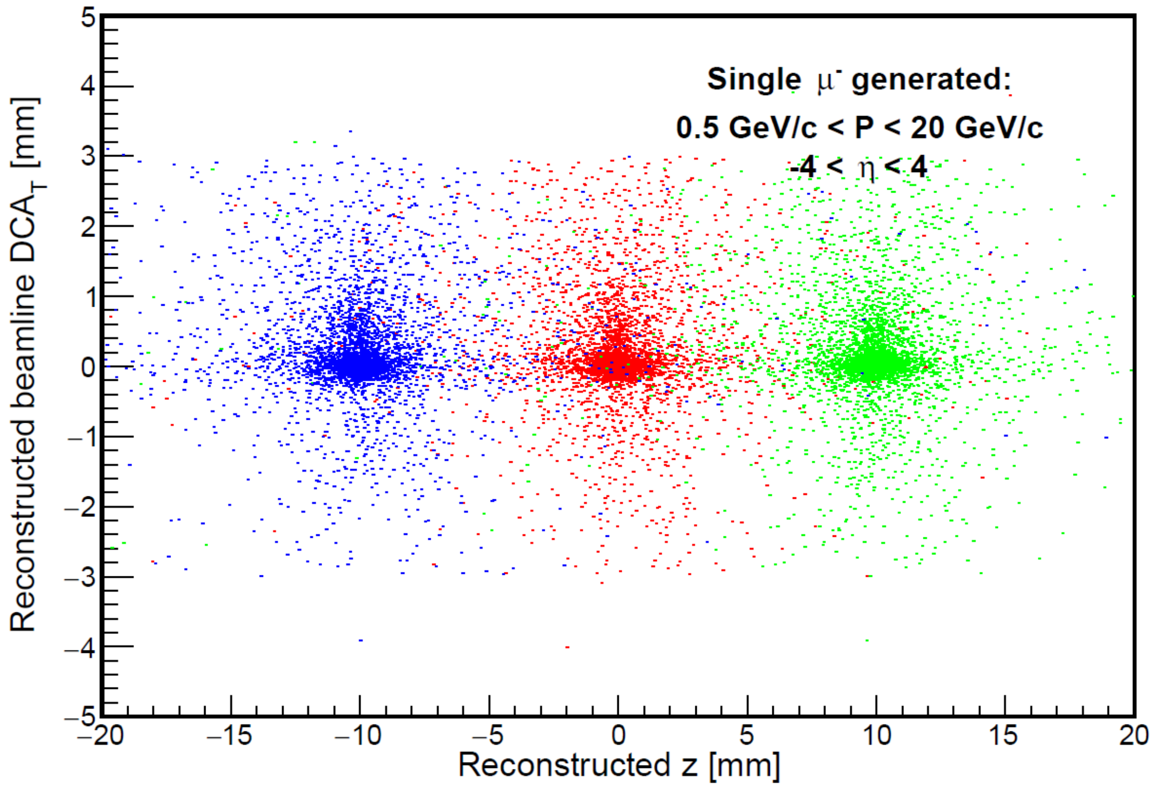
(vx,vy,vz) = (0,0,0) mm
(vx,vy,vz) = (0,0,+10) mm
(vx,vy,vz) = (0,0,-10) mm

Single-particle reconstruction on beamline

Truth-seeded tracking



Real-seeded tracking

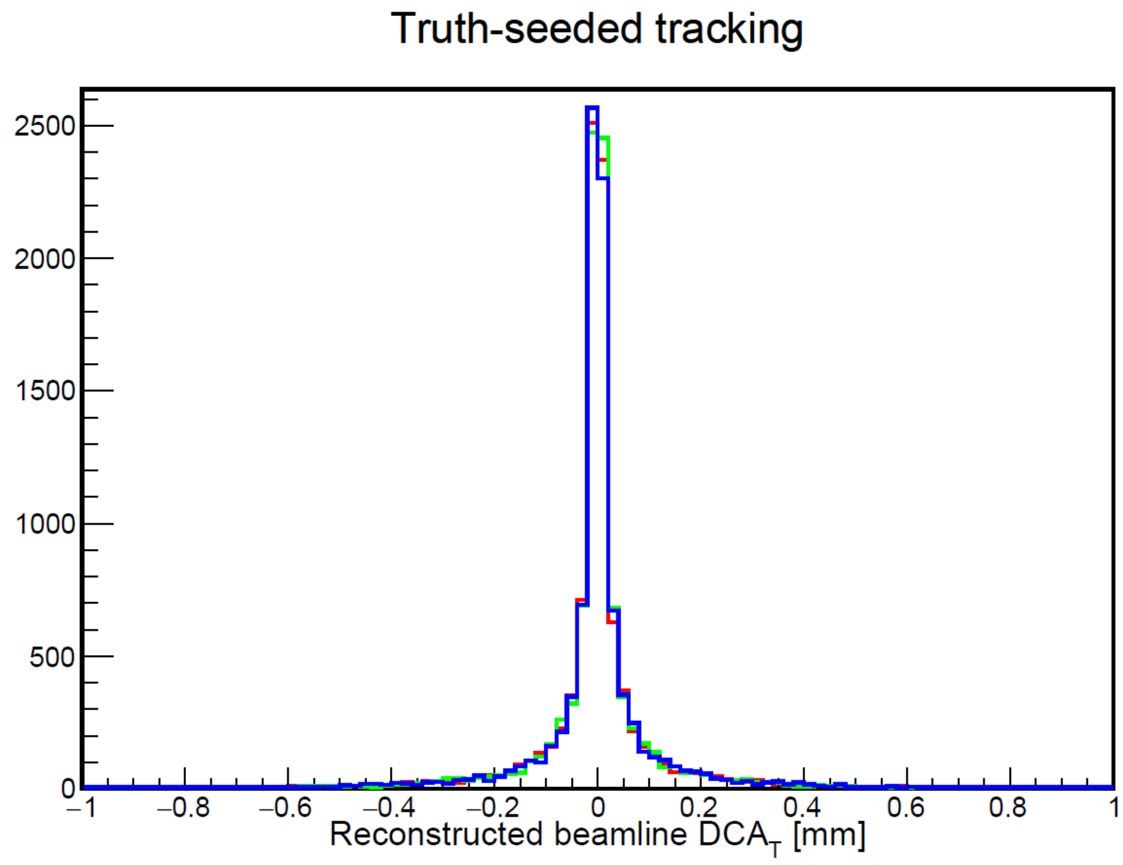


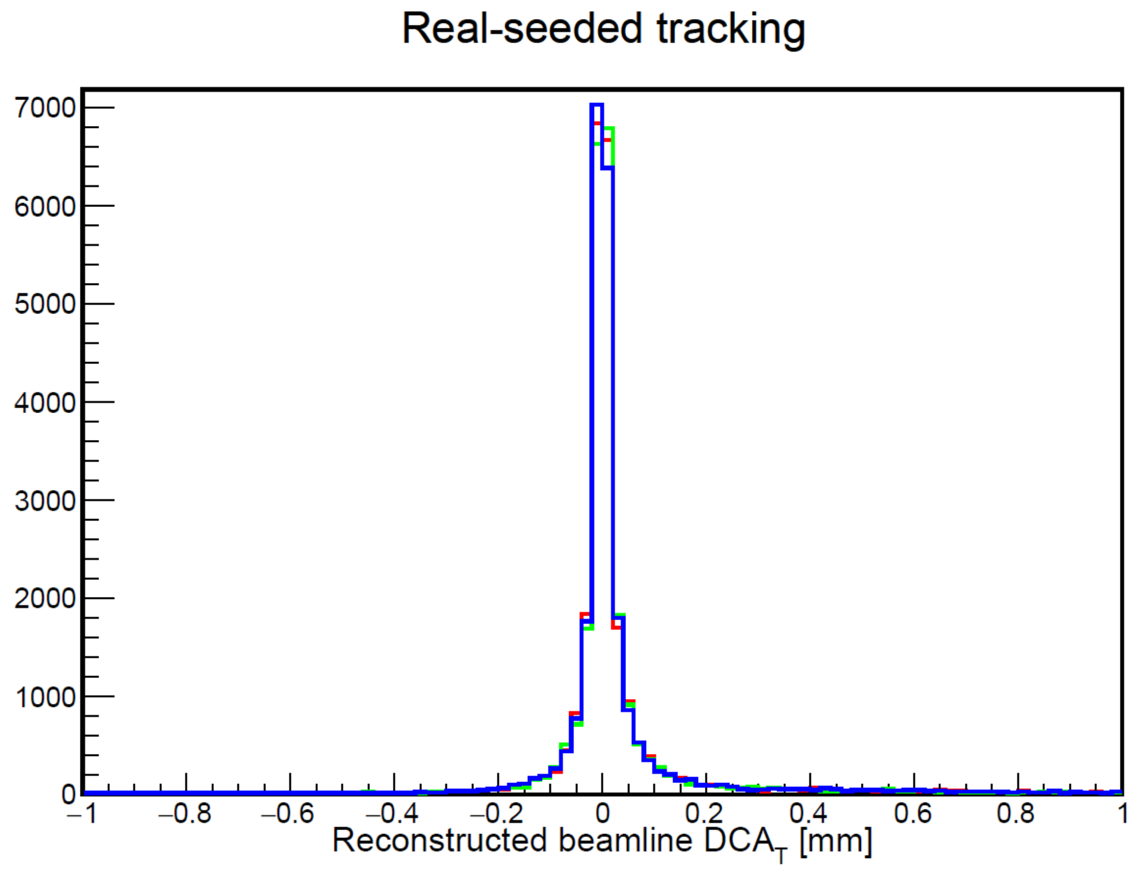
ACTS Loc-b → Reconstructed z: longitudinal impact parameter with respect to (0,0,0)

ACTS Loc-a → Reconstructed transverse DCA: (signed) transverse impact parameter with respect to (0,0,0)

(vx,vy,vz) = (0,0,0) mm
(vx,vy,vz) = (0,0,+10) mm
(vx,vy,vz) = (0,0,-10) mm

Single-particle reconstruction on beamline





Reconstructed z: longitudinal impact parameter with respect to (0,0,0)
Reconstructed transverse DCA: (signed) transverse impact parameter with respect to (0,0,0)

Some issues when particles are generated off the z axis

Truth seeding

1. Particle's generation point is currently used to set ACTS Loc-a and Loc-b. However, the particle will rarely be tangential to the defined line surface passing through (0,0,0) along the z axis.

Real seeding

1. Seed charge is reconstructed incorrectly for a large number of seeds. A proposed fix can be found here: https://github.com/eic/EICrecon/tree/seed_charge. More details in Jeetendra's presentation.
2. In our current approach, the seed phi is wrong by 180 degrees when the seed charge is flipped.
3. Error matrix on seed parameters may be too small.

Some issues when particles are generated off the z axis

Truth seeding

1. Particle's generation point is currently used to set ACTS Loc-a and Loc-b. However, the particle will rarely be tangential to the defined line surface passing through (0,0,0) along the z axis.

Real seeding

1. Seed charge is reconstructed incorrectly for a large number of seeds. A proposed fix can be found here: https://github.com/eic/EICrecon/tree/seed_charge. More details in Jeetendra's presentation.
2. In our current approach, the seed phi is wrong by 180 degrees when the seed charge is flipped.
3. Error matrix on seed parameters may be too small.

Truth seeding: Initial position parameters

- We set the initial track parameters based on the generated particle’s momentum vector, charge, and creation point. This information is then fed into the CKF in addition to a line surface (perigee surface) along the z axis through (0,0,0).

https://github.com/eic/EICrecon/blob/seed_charge/src/algorithms/tracking/TrackParamTruthInit.cc

```
// Insert into edm4eic::TrackParameters, which uses numerical values in its specified units
auto track_parameter = track_parameters->create();
track_parameter.setType(-1); // type --> seed(-1)
track_parameter.setLoc({static_cast<float>(std::hypot(v.x, v.y)), static_cast<float>(v.z)})
track_parameter.setLocError({1.0, 1.0}); // sqrt(variance) of location [mm]
track_parameter.setTheta(theta); //theta [rad]
track_parameter.setPhi(phi); // phi [rad]
track_parameter.setQOverP(charge / pinit); // Q/p [e/GeV]
track_parameter.setMomentumError({0.01, 0.05, 0.1}); // sqrt(variance) on theta, phi, q/p [
track_parameter.setTime(mcparticle.getTime()); // time [ns]
track_parameter.setTimeError(10e9); // error on time [ns]
track_parameter.setCharge(charge); // charge
```

Truth seeding: Initial position parameters

- ➢ We set the initial track parameters based on the generated particle's momentum vector, charge, and creation point. This information is then fed into the CKF in addition to a line surface (perigee surface) along the z axis through (0,0,0).
- ➢ The created particle's momentum vector may not be tangential to the cylinder surrounding the line surface at its creation point.
- ➢ In addition, ACTS Loc-a can be positive or negative depending on whether the particle transverses the line surface clockwise or counterclockwise.

https://github.com/eic/EICrecon/blob/seed_charge/src/algorithms/tracking/TrackParamTruthInit.cc

```
// Insert into edm4eic::TrackParameters, which uses numerical values in its specified units
auto track_parameter = track_parameters->create();
track_parameter.setType(-1); // type --> seed(-1)
track_parameter.setLoc({static_cast<float>(std::hypot(v.x, v.y)), static_cast<float>(v.z)})
track_parameter.setLocError({1.0, 1.0}); // sqrt(variance) of location [mm]
track_parameter.setTheta(theta); //theta [rad]
track_parameter.setPhi(phi); // phi [rad]
track_parameter.setQOverP(charge / pinit); // Q/p [e/GeV]
track_parameter.setMomentumError({0.01, 0.05, 0.1}); // sqrt(variance) on theta, phi, q/p [
track_parameter.setTime(mcparticle.getTime()); // time [ns]
track_parameter.setTimeError(10e9); // error on time [ns]
track_parameter.setCharge(charge); // charge
```

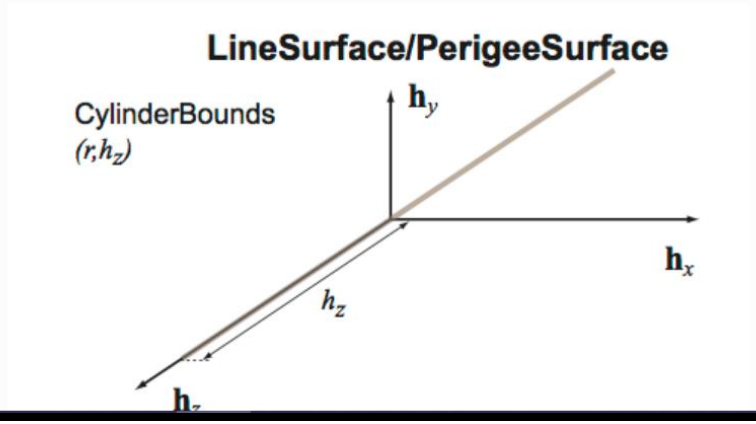
Truth seeding: Initial position parameters

- We set the initial track parameters based on the generated particle's momentum vector, charge, and creation point. This information is then fed into the CKF in addition to a line surface (perigee surface) along the z axis through (0,0,0).
- The created particle's momentum vector may not be tangential to the cylinder surrounding the line surface at its creation point.
- In addition, ACTS Loc-a can be positive or negative depending on whether the particle transverses the line surface clockwise or counterclockwise.

`Acts::LineSurface` is a special kind of surface that depends on a reference direction, typically the unit momentum direction $\vec{d}$ of a particle. A point in space is considered *on surface* if and only if it coincides with the point of closest approach between the direction vector $\vec{d}$ and the line direction vector $\vec{z}$. As such, the function `Acts::LineSurface::globalToLocal()` can fail, if the argument position and direction do not fulfill this criterion. It is pure-virtual, meaning that it can not be instantiated on its own.

class `LineSurface` : *public* Acts::Surface

Base class for a linear surfaces in the TrackingGeometry to describe dirft tube, straw like detectors or the Perigee It inherits from Surface.



https://acts.readthedocs.io/en/latest/core/geometry/surfaces.html#line-surface

Truth seeding: Initial position parameters

- To test, generate a single particle from **(x,y,z) = (10,0,0) mm**, with a momentum direction of **(px,py,pz) = {cos(10 degrees), sin(10 degrees), 0}.**
- In the EICRecon CKF class, use the **LocaltoGlobal** function on the initial track parameters. I assume that the CKF uses this same function internally when doing the particle propagation.
- We see that the CFK will think that the particle's parameters were given at a different position than the creation point – i.e. the tangential point around the line surface which is at the same radius. But this point is not usually a point on the particle's trajectory.

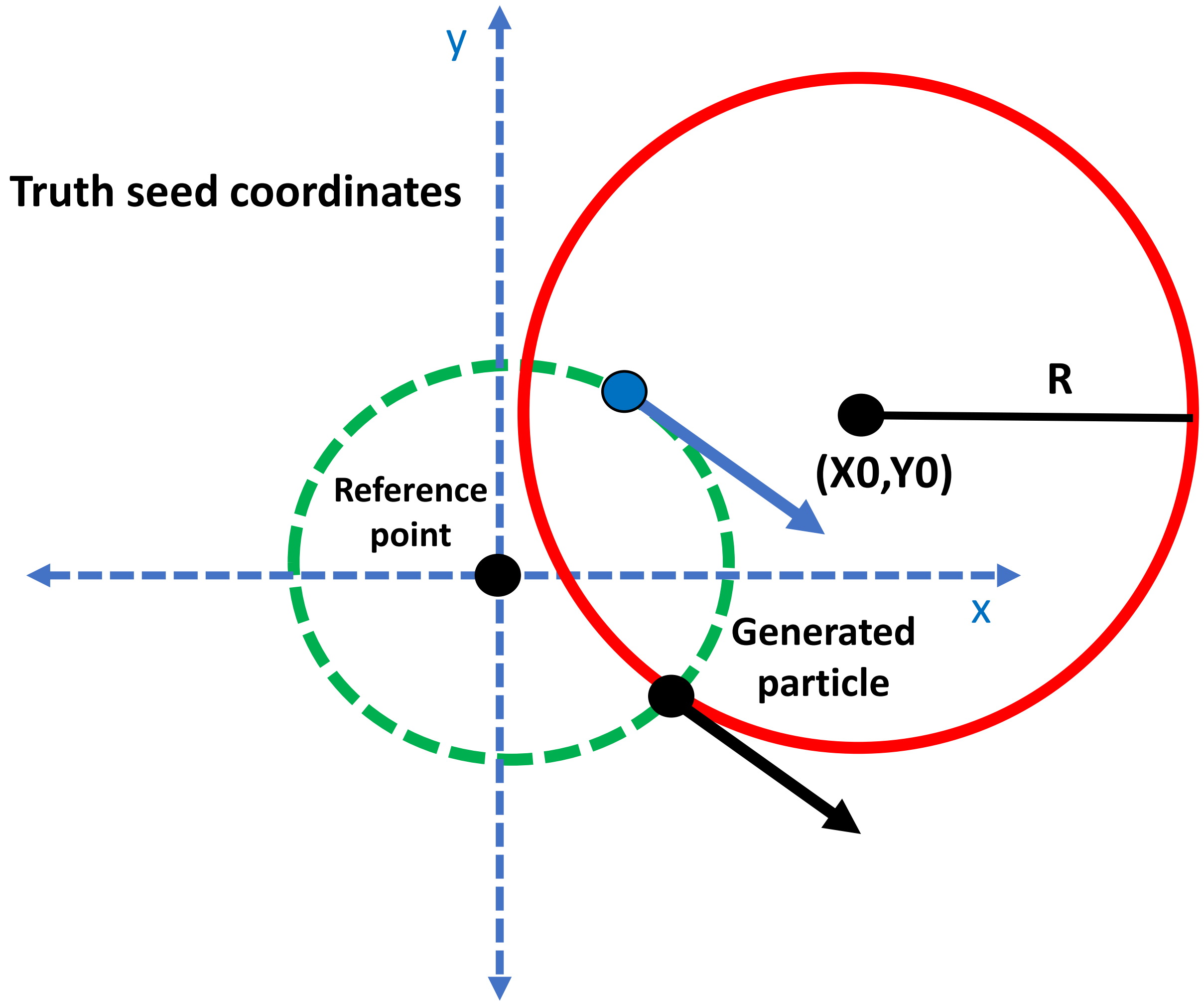
```
Acts::Vector3 mydirection(sin(track_parameter.getTheta())*cos(track_parameter.getPhi()),
                                        sin(track_parameter.getTheta())*sin(track_parameter.getPhi()),
                                        cos(track_parameter.getTheta()));
auto myglobal = pSurface->localToGlobal(m_geoSvc->getActsGeometryContext(),
                                         {params[Acts::eBoundLoc0],params[Acts::eBoundLoc1]},
                                         mydirection);

auto myglobal_r = sqrt(myglobal.x()*myglobal.x()+myglobal.y()*myglobal.y());

std::cout<<"Global x, y, z, r:"<<std::endl;
std::printf("%10.2f, %10.2f, %10.2f, %10.2f\n",myglobal.x(),myglobal.y(),myglobal.z(),myglobal_r);
```

```
Global x, y, z, r:
     -1.74,       9.85,       0.00,      10.00
```

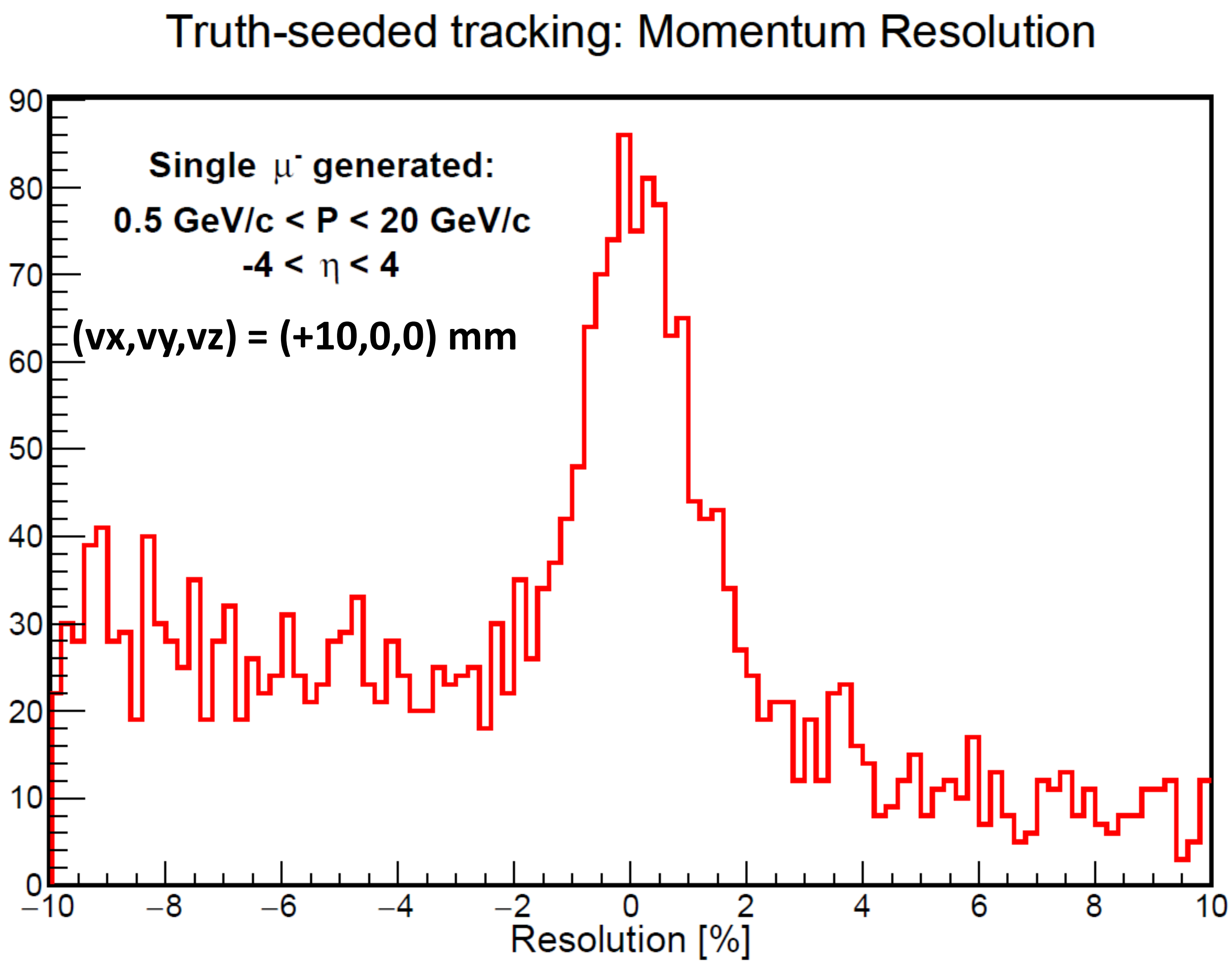
Truth seeding: Initial position parameters



Black arrow: Generated particle at its creation point

Blue arrow: Where the CKF will think the particle comes from in the current truth seeding implementation.

Truth seeding: Initial position parameters

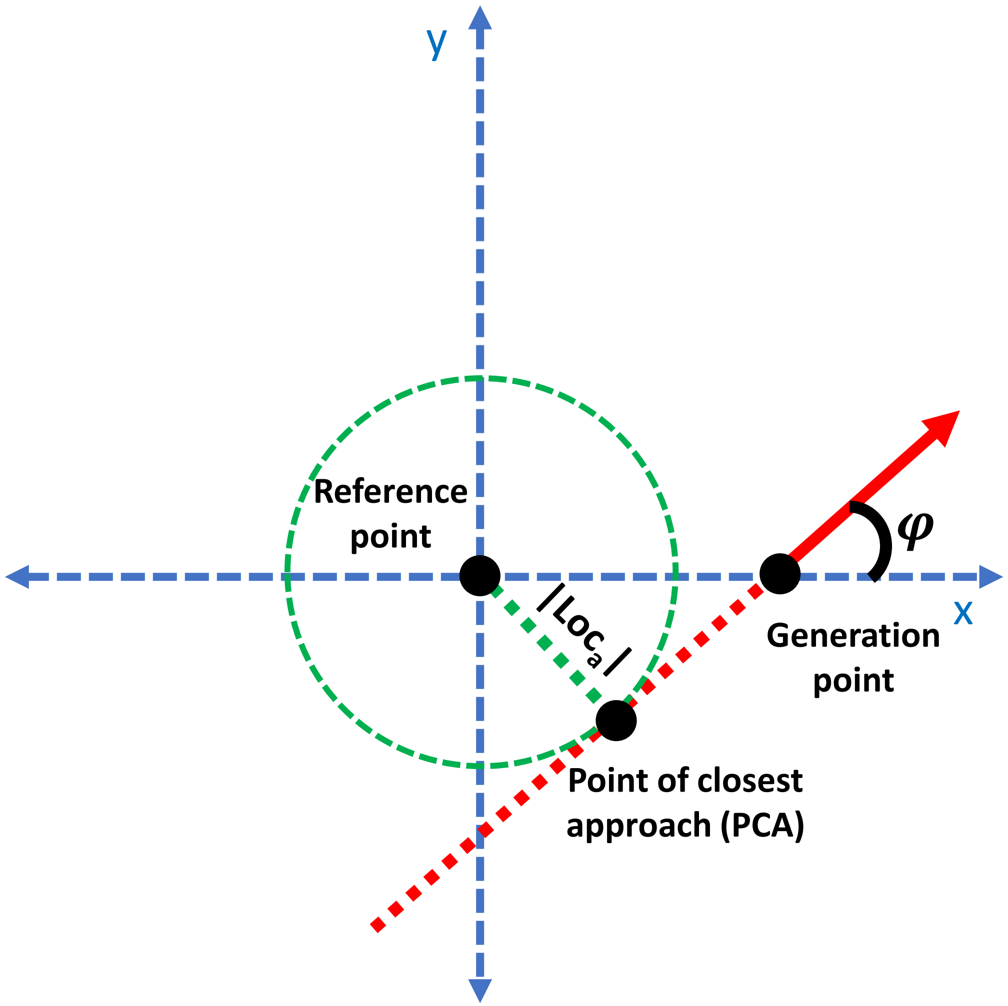


We see poor truth-seeded momentum reconstruction when a particle is generated 10 mm off from the beamline (z axis).

Possible fixes for truth seeding

1. For a given generated particle, shift the line surface so that the reference point is the creation point of the particle. (The line surface would still be parallel to the z axis.) This will require some modification to the data model.
2. Using the particle's truth information, track the particle back to the DCA point with respect to the current line surface. This is similar to the approach used for real seeding.

How to fix using approach 2



- Example: generation point at (x,y,z) = (+1,0,0) mm. $|Loc_a| = |\sin\varphi|$ under 'straight line' approximation.
- The sign of Loc_a will be positive if the particle at the PCA transits clockwise around the line surface (reference point) through (x,y) = (0,0). It will be negative if it transits counterclockwise. See here: https://github.com/acts-project/acts/blob/main/Core/src/Surfaces/LineSurface.cpp#L80-L123.
- We need to think about how to set the corresponding z value (ACTS Loc_b) correctly.

Some issues when particles are generated off the z axis

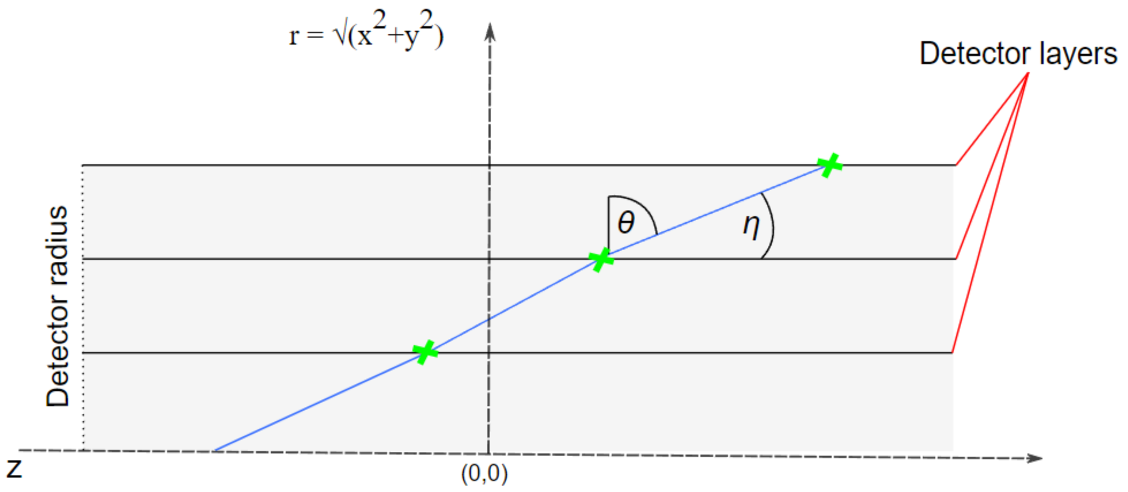
Truth seeding

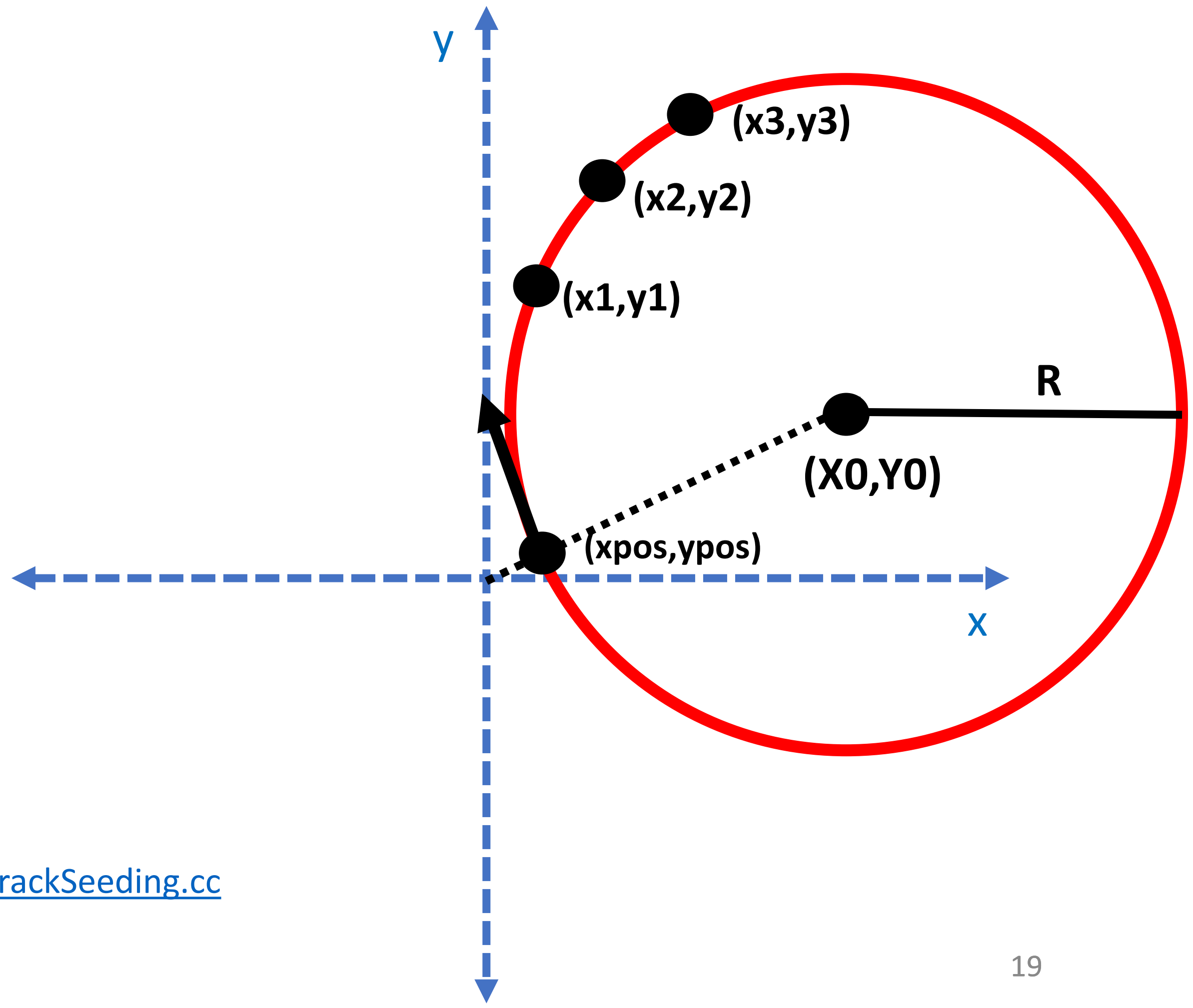
1. Particle's generation point is currently used to set ACTS Loc-a and Loc-b. However, the particle will rarely be tangential to the defined line surface passing through (0,0,0) along the z axis.

Real seeding

1. Seed charge is reconstructed incorrectly for a large number of seeds. A proposed fix can be found here: https://github.com/eic/EICrecon/tree/seed_charge. More details in Jeetendra's presentation.
2. In our current approach, the seed phi is wrong by 180 degrees when the seed charge is flipped.
3. Error matrix on seed parameters may be too small.

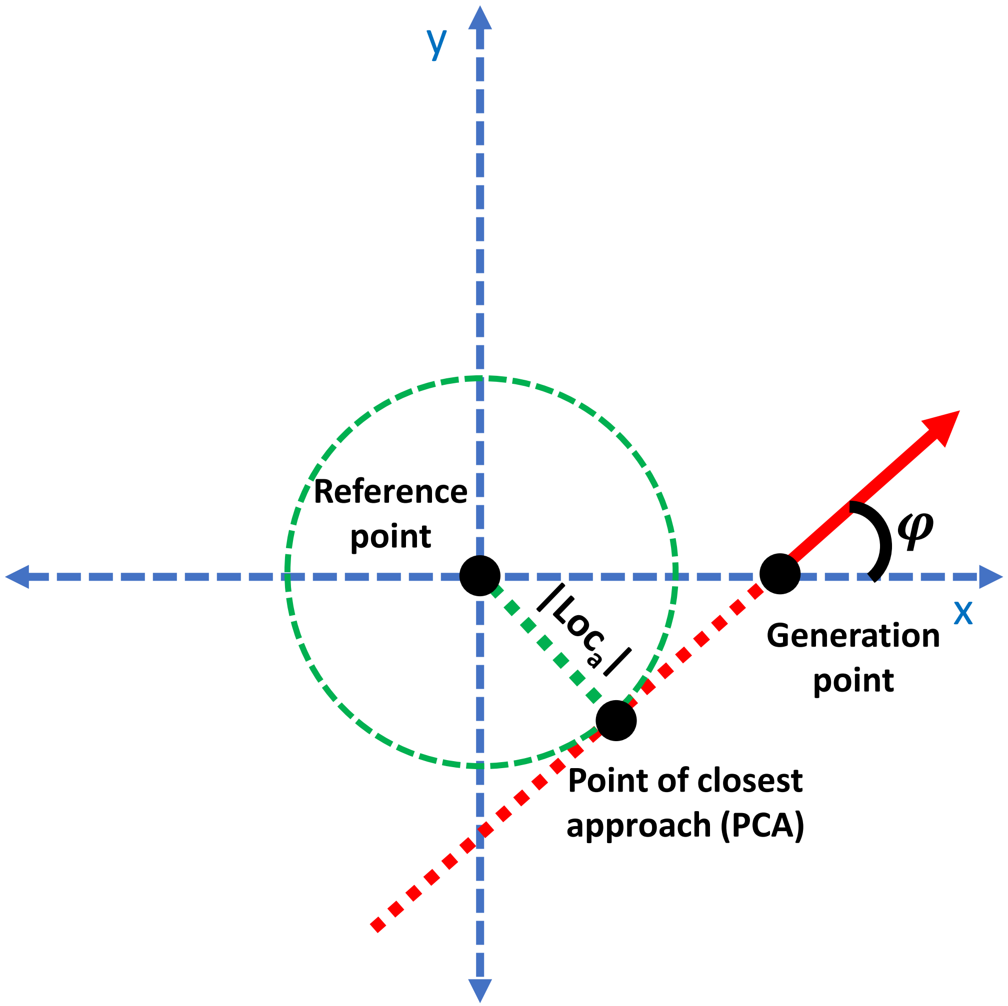
Reconstruction of seed parameters for real seeding

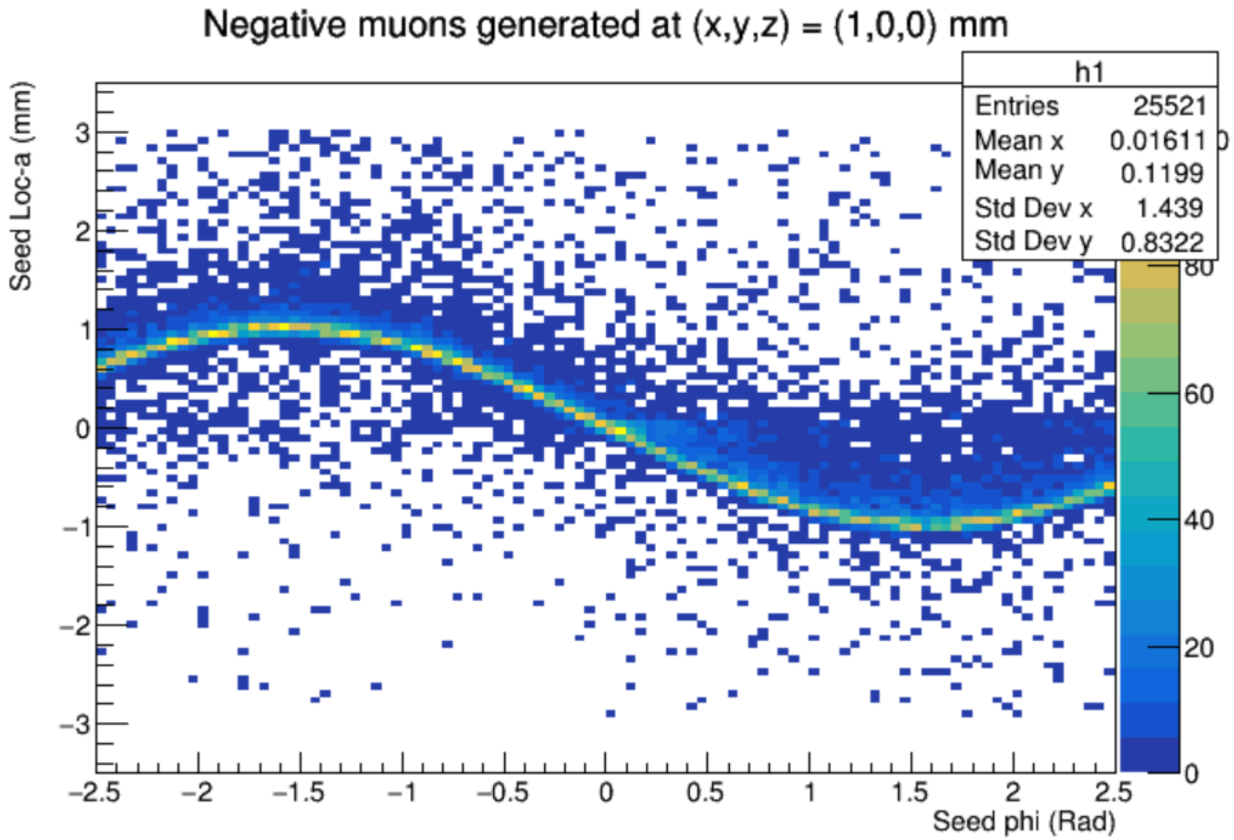




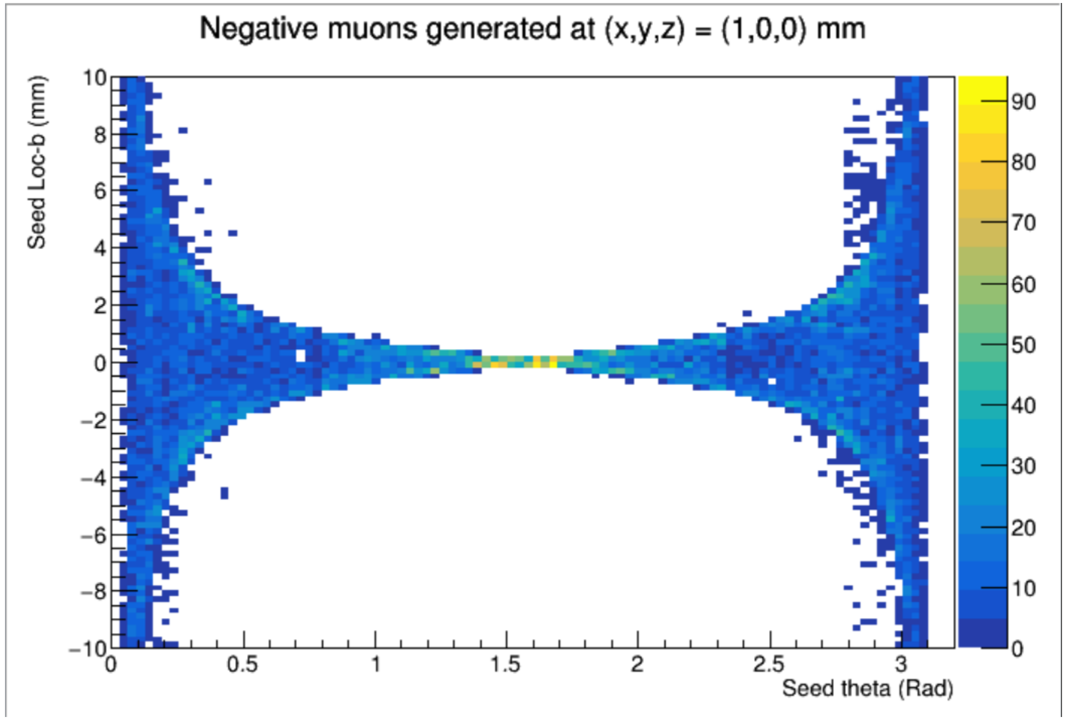
https://github.com/eic/EICrecon/blob/main/src/algorithms/tracking/TrackSeeding.cc

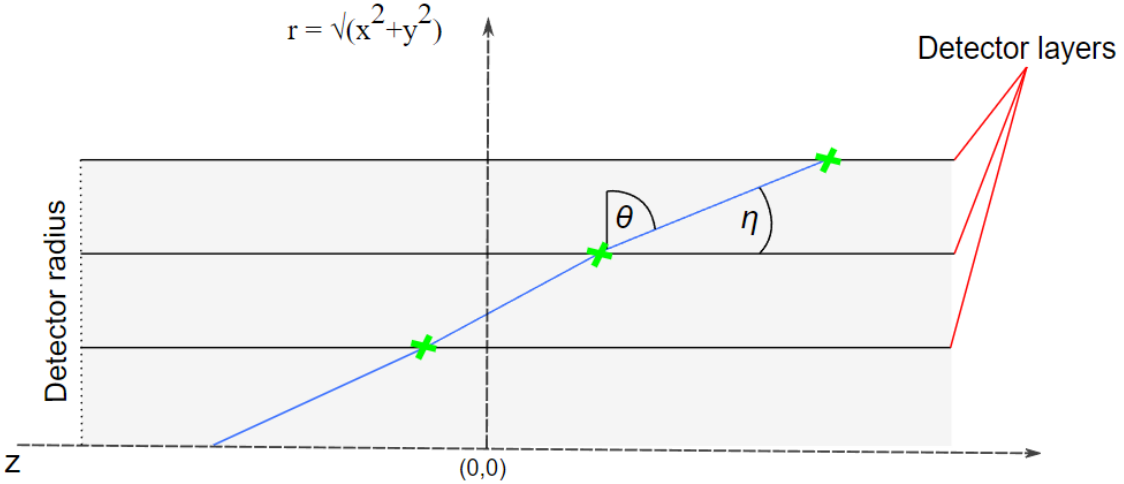
Real seeding reconstructs position parameters well



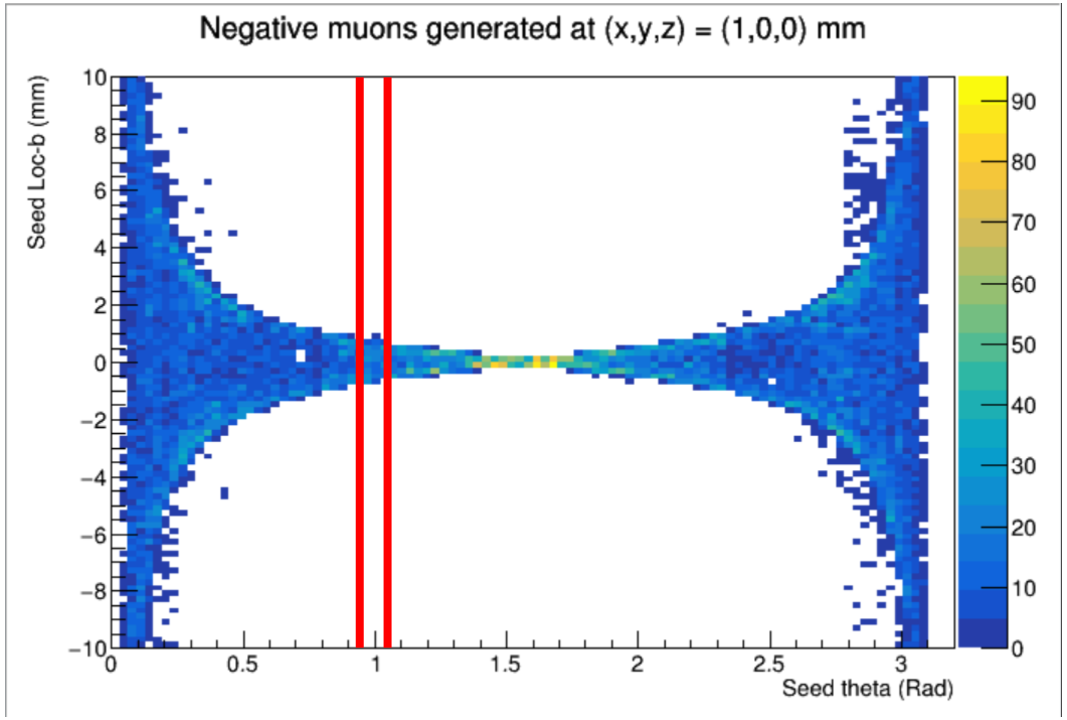


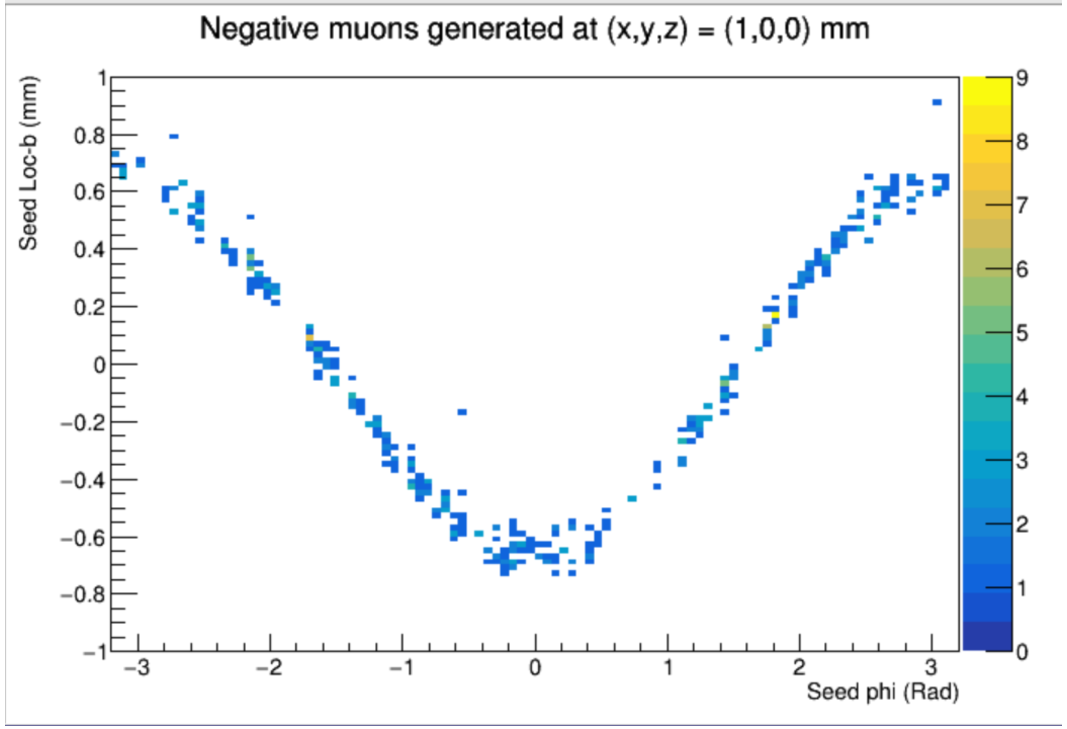
Real seeding reconstructs position parameters well





Real seeding reconstructs position parameters well



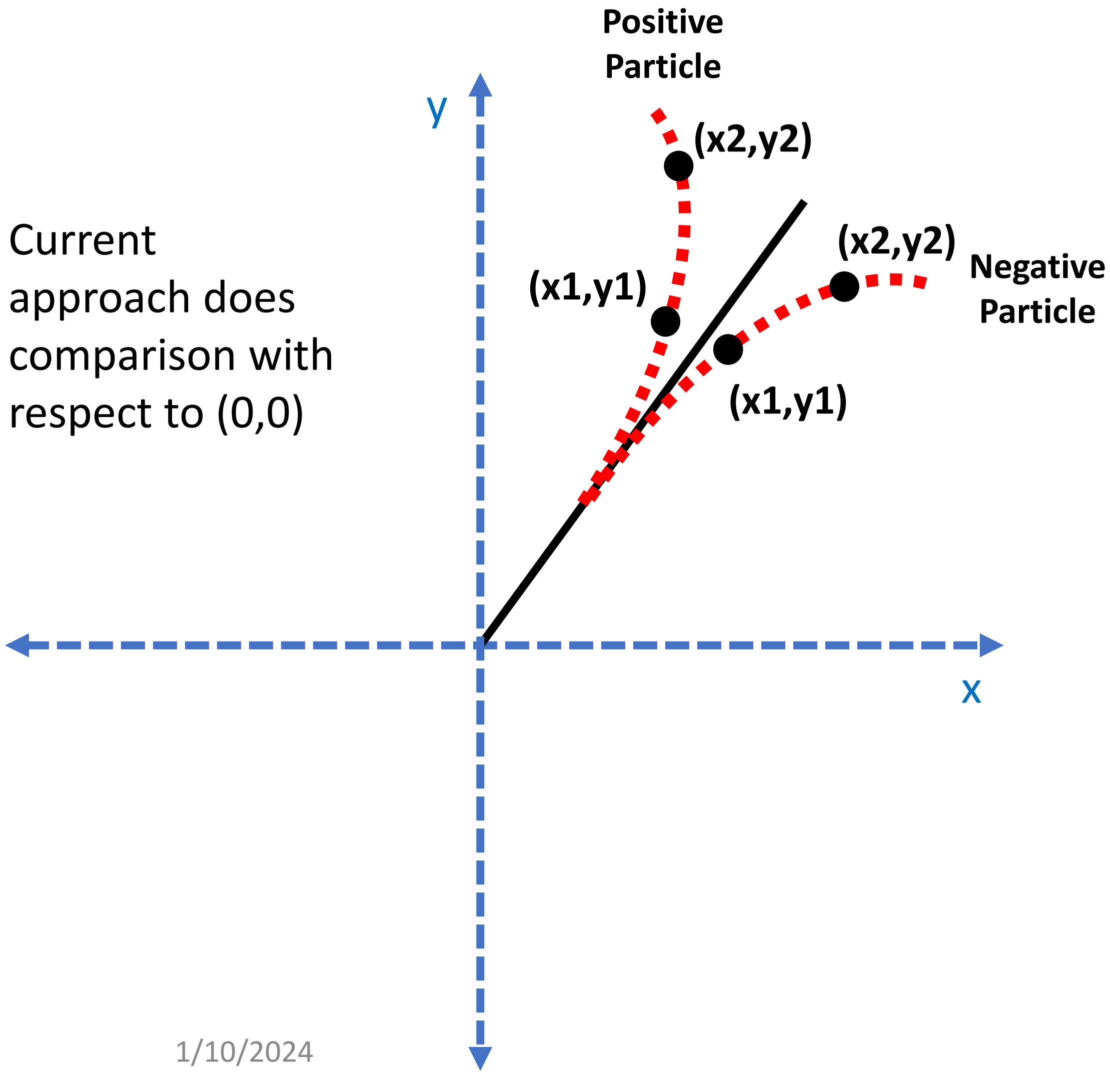


Reconstructed Z (ACTS Loc-b) that corresponds to transverse PCA depends on theta, as well as on phi (projection onto displacement plane).

Seed charge determination

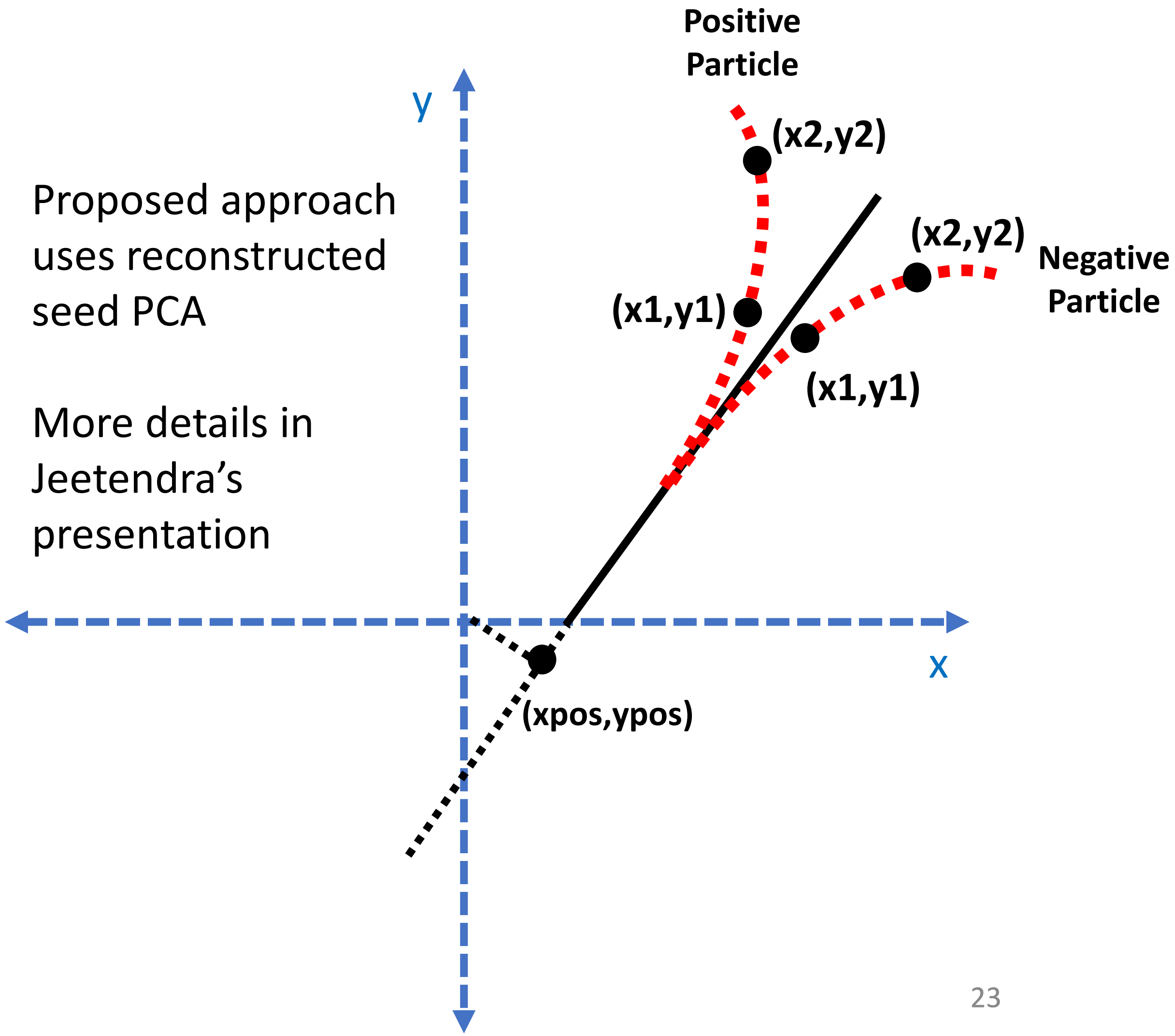
https://github.com/eic/EICrecon/tree/seed_charge

Current approach does comparison with respect to (0,0)



Proposed approach uses reconstructed seed PCA

More details in Jeetendra's presentation



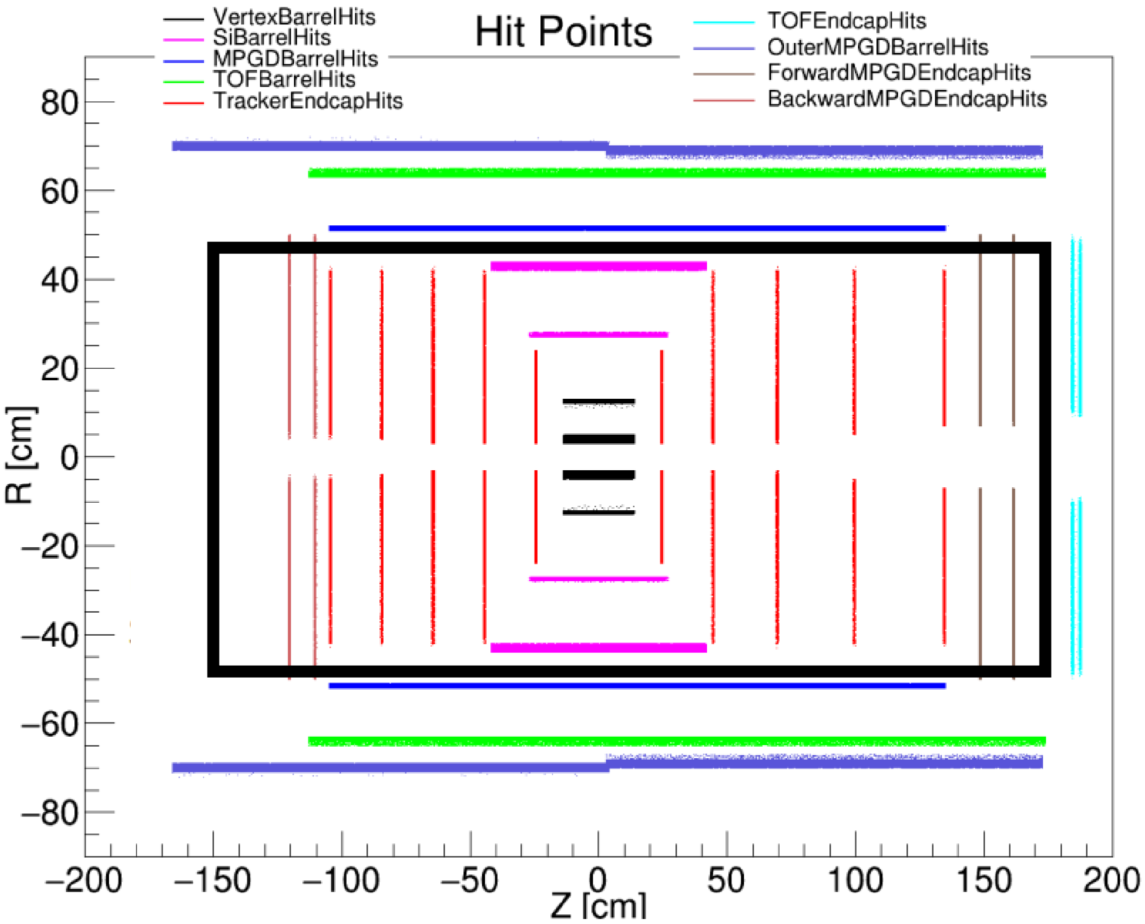
Summary

- Both truth-seeded and real-seeded tracking work well for particles generated along the z axis.
- Single particle simulations when the particle is moved away from the z axis show some problems for both seeding implementations. Some of these issues have been identified and we should work to resolve them.
- These fixes should probably be made before working on secondary vertexing.

Backup

Seed finding using the ACTS orthogonal seeder

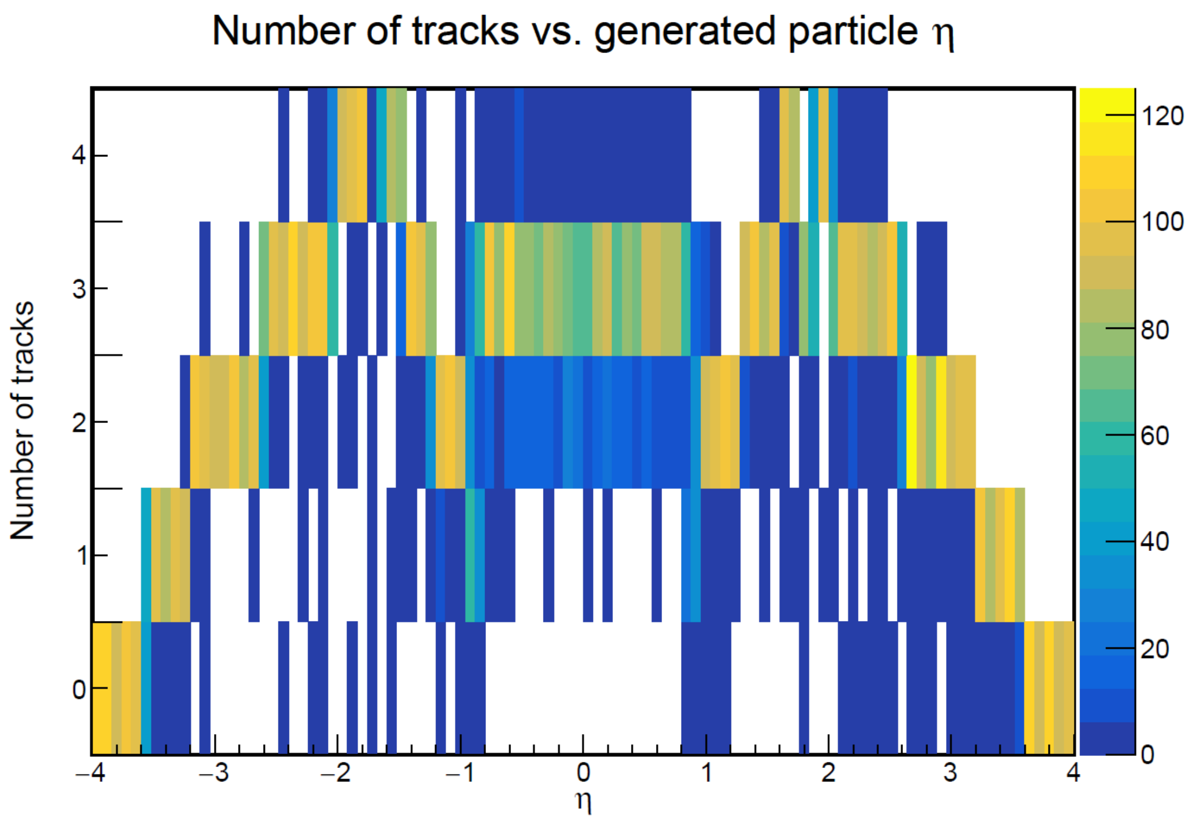
We search for seeds in part of our tracking volume – that is in our MAPS silicon pixel detectors.

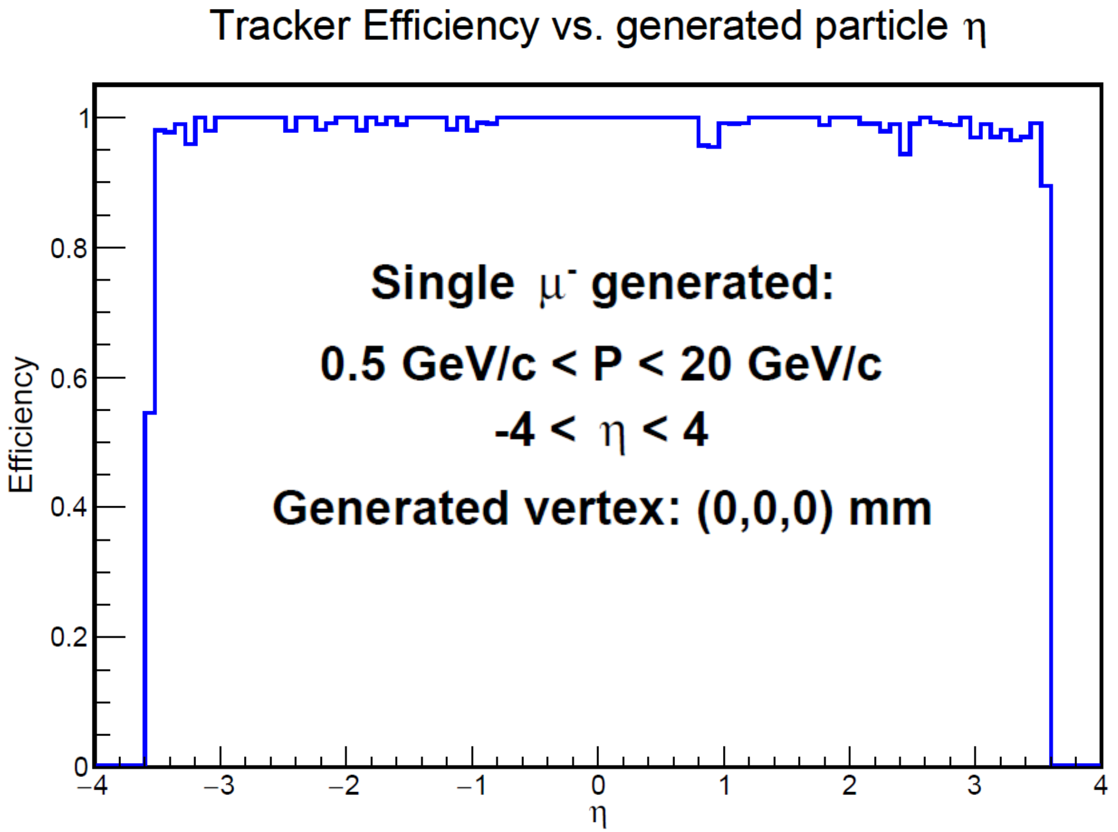


ACTS seed finder and filter parameters

Parameter	Description	Value
bFieldInZ	z component of magnetic field	1.7 T
rMax	Maximum r value to look for seeds	440 mm
rMin	Minimum r value to look for seeds	33 mm
zMin	Minimum z value to look for seeds	-1500 mm
zMax	Maximum z value to look for seeds	1700 mm
beamPosX	Beam offset in x	0
beamPosY	Beam offset in y	0
deltaRMinTopSP	Min distance in r between middle and top SP in one seed	**10 mm**
deltaRMinBottomSP	Min distance in r between middle and bottom SP in one seed	**10 mm**
deltaRMaxTopSP	Max distance in r between middle and top SP in one seed	**200 mm**
deltaRMaxBottomSP	Max distance in r between middle and top SP in one seed	**200 mm**
collisionRegionMin	Min z for primary vertex	-250 mm
collisionRegionMax	Max z for primary vertex	250 mm
cotThetaMax	Cotangent of max theta angle	**27.29**
minPt	Min transverse momentum	100 MeV/cotThetaMax
maxSeedsPerSpM	Max number of seeds a single middle space point can belong to - 1	0
sigmaScattering	How many standard devs of scattering angles to consider	5
radLengthPerSeed	Average radiation lengths of material on the length of a seed	0.1
impactMax	Max transverse PCA allowed	3 mm
rMinMiddle	Min R for middle space point	20 mm
rMaxMiddle	Max R for middle space point	400 mm
bFieldMin	min B field	0.1

Single-particle tracking efficiency/multiplicity





An efficient event is defined as one where at least one track is found. We (almost) always get a single track per seed in our setup right now. This is because we only save the trajectory with the longest branch (i.e. the *trackTips.front()*).

Seed duplicates – particles have multiple seeds

If we have a particle at mid-rapidity which hits layers L0, L1, L2, L3, and L4, then we can make the following combinations:

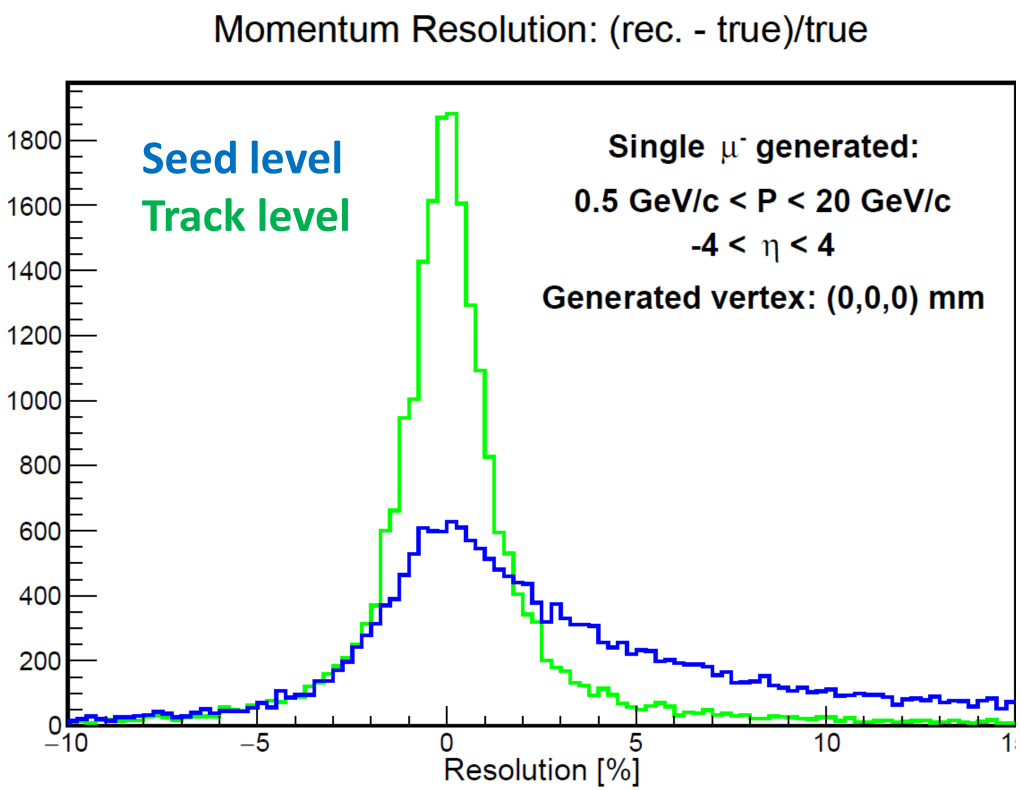
1. L0,L1,L2
2. L0,L2,L3
3. L0,L3,L4
4. ✖ L0,L1,L3
5. ✖ L0,L1,L4
6. ✖ L0,L2,L4
7. ✖ L1,L2,L3
8. ✖ L1,L2,L4
9. ✖ L1,L3,L4
10. ✖ L2,L3,L4

ACTS seed finder and filter parameters

Parameter	Description	My New Default
bFieldInZ	z component of magnetic field	1.7 T
rMax	Maximum r value to look for seeds	440 mm
rMin	Minimum r value to look for seeds	33 mm
zMin	Minimum z value to look for seeds	-1500 mm
zMax	Maximum z value to look for seeds	1700 mm
beamPosX	Beam offset in x	0
beamPosY	Beam offset in y	0
deltaRMinTopSP	Min distance in r between middle and top SP in one seed	**10 mm**
deltaRMinBottomSP	Min distance in r between middle and bottom SP in one seed	**10 mm**
deltaRMaxTopSP	Max distance in r between middle and top SP in one seed	**200 mm**
deltaRMaxBottomSP	Max distance in r between middle and top SP in one seed	**200 mm**
collisionRegionMin	Min z for primary vertex	-250 mm
collisionRegionMax	Max z for primary vertex	250 mm
cotThetaMax	Cotangent of max theta angle	**27.29**
minPt	Min transverse momentum	100 MeV/cotThetaMax
maxSeedsPerSpM	Max number of seeds a single middle space point can belong to - 1	0
sigmaScattering	How many standard devs of scattering angles to consider	5
radLengthPerSeed	Average radiation lengths of material on the length of a seed	0.1
impactMax	Max transverse PCA allowed	3 mm
rMinMiddle	Min R for middle space point	20 mm
rMaxMiddle	Max R for middle space point	400 mm
bFieldMin	min B field	0.1

Tracking performance

Real-seeded tracking



Comparison of truth-seeded and real-seeded tracking