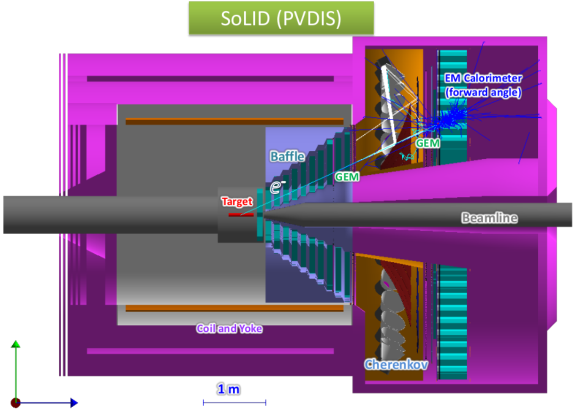
Tracking for SoLID experiment

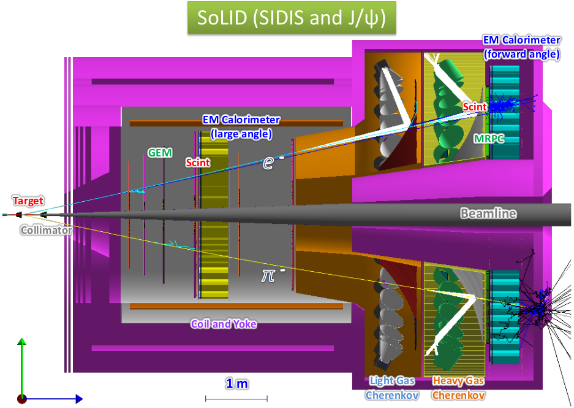
Zhongling Ji
based on the work from
Weizhi Xiong, Chao Peng, and Zhiwen Zhao
For the SoLID Collaboration Meeting 2026

Syracuse University

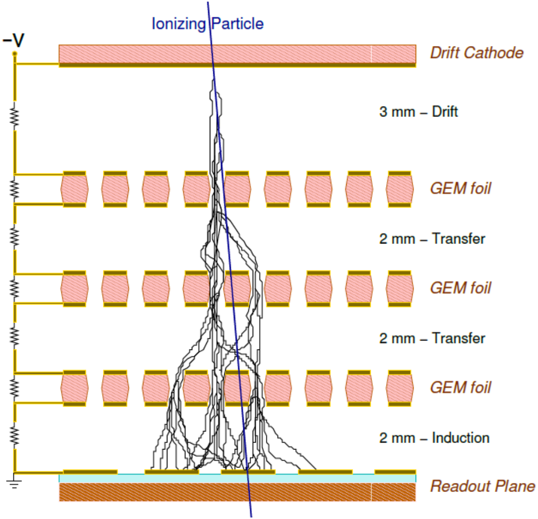
February 19, 2026

SoLID detector for PVDIS and SIDIS



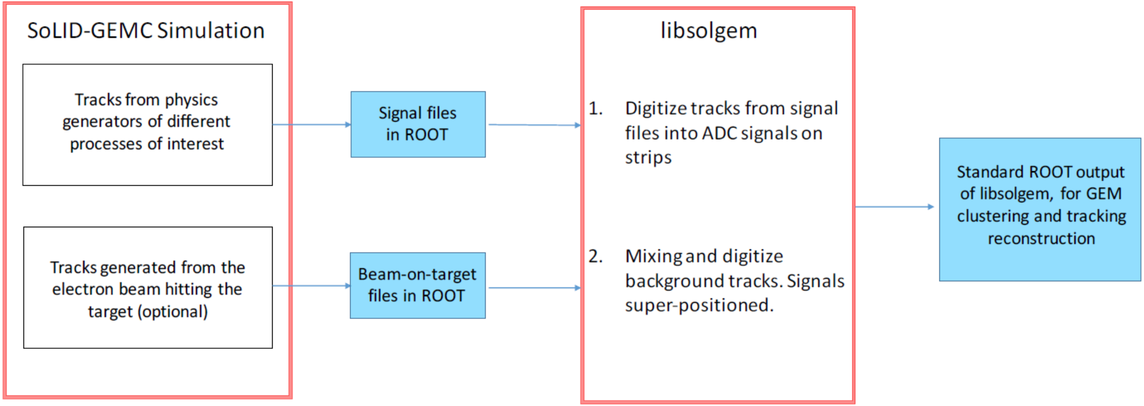


Principle of GEM detector

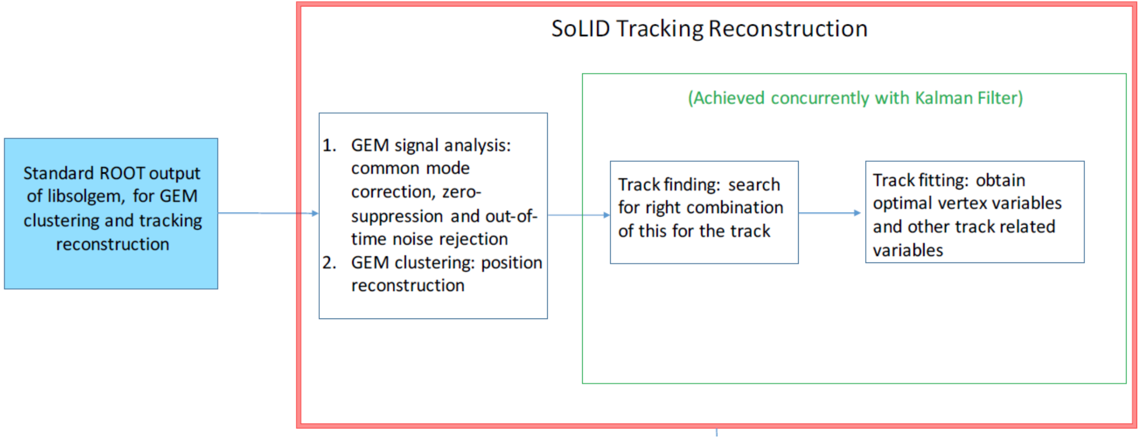


Standalone implementation

Mixing with background

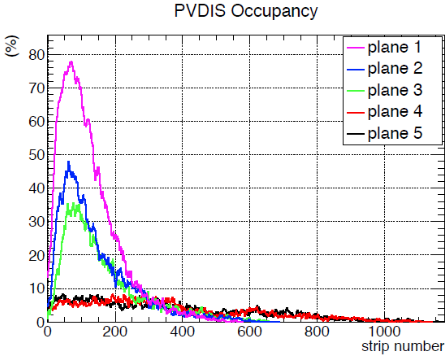


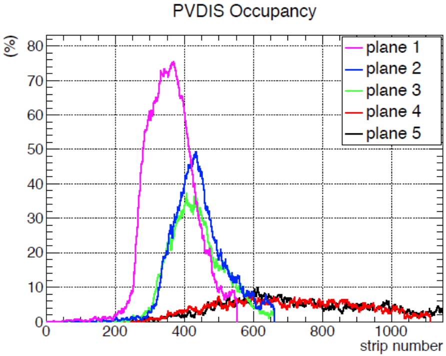
Data flow for tracking reconstruction



GEM occupancy

- 10^{37} cm^{-2}s^{-1} for SIDIS-^{3}He and J/ψ.
- 10^{39} cm^{-2}s^{-1} for PVDIS.
- Occupancy by strips in the PVDIS configuration:

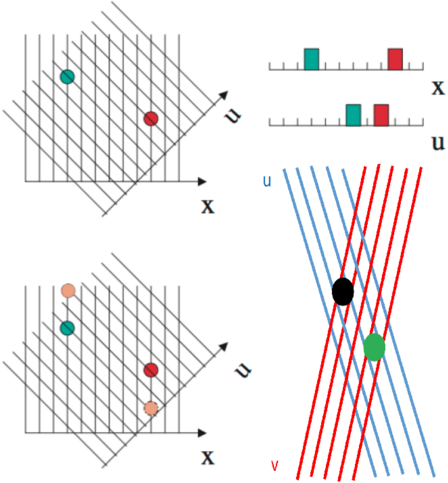




Hit multiplicity and false combinations

	GEM 1	GEM 2	GEM 3	GEM 4	GEM 5	GEM 6
Occupancy	2.5%	9.7%	4.1%	2.6%	2.0%	1.5%
Hit Multi.	420	5048	1860	1136	460	424

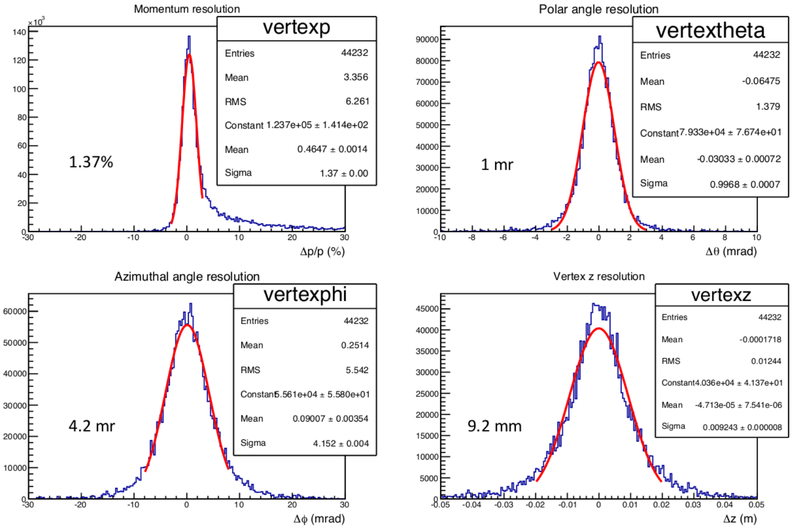
- Hit multiplicity for SIDIS:
  - High multiplicity after threshold cut.
  - Number will go up for 20 GEM sectors (currently 30).
  - Large amount of combinations $\sim 10^3$.
- Reject false combinations:
  - Out of physical intersection.
  - Charge asymmetry: $A = \frac{q_u - q_v}{q_u + q_v}$.



Track finding and fitting

- Track finding:
  - Assume 100% beam-on-target background.
  - Use the calorimeter to start track finding.
  - Assume that there is only one high energy hit on the calorimeter.
  - Use hits on downstream detectors as the hit density there is much lower.
- Track fitting:
  - Use Kalman filter (KF) to look for hits along its way of propagation.
  - Allow only one missing hit.
  - Use a special hit from beam position monitor (BPM).
- Final selection rules:
  - Propagate to the calorimeter to check the agreement.
  - Compare with the energy measurement from the calorimeter.
  - For tracks that share a common hit, only the track with the largest number of hits are kept.

Resolutions in SIDIS at forward angle



ACTS implementation

Purposes and procedures

- Purposes:
  - ✔ Signal level ACTS tracking (single electron with SIDIS configuration) and comparison with the standalone results
  - ✖ Realistic ACTS tracking (signal + background for SIDIS/PVDIS configuration) and comparison with the standalone results
- Procedures:
  - ✔ Verify the geometries for SIDIS tracking
  - ✔ Implement the ACTS tracking geometry in the solid_dd4hep repository and run single electron simulations
    - Use the existing algorithms in EICrecon for digitization (add resolution smearing and efficiency) and track seeding/finding/fitting

DD4hep and JANA2/EICrecon

DD4hep: code structure (mental map)

- **Core idea**: one detector description, accessed by simulation and reconstruction
- Typical layers/components:
  - **Compact description (XML)**: defines constants, materials, regions, readouts/segmentation, detector placements
  - **Detector constructors (C++ plugins)**: implement each sub-detector (GEMs, calorimeters, support) and register to DD4hep
  - **Sensitive detectors + hit collections**: connect volumes to Geant4 stepping and define what is recorded
  - **Conditions/field**: magnetic field map + calibration/conditions interfaces used consistently by downstream code
- How this shows up in practice:
  - A top-level `compact.xml` includes subsystem XML + loads constructor plugins
  - Reconstruction consumes the same geometry via `dd4hep::Detector` (plus converters for ACTS surfaces/material)

JANA2 / EICrecon: code structure (mental map)

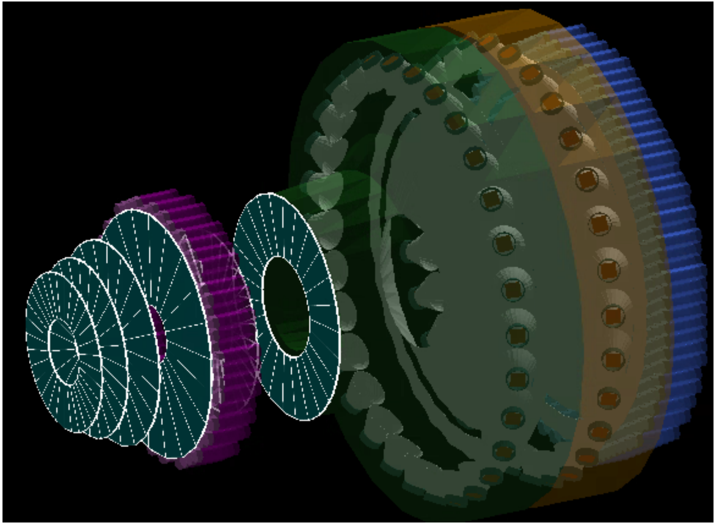
- **JANA2 is the framework**: event loop, scheduling, multi-threading, configuration, plugin loading
- Main building blocks in a JANA2 app:
  - **Plugins**: shared libraries that register factories, processors, services, and configuration
  - **Factories** (e.g. `JFactoryT<T>`): produce typed data objects per event (hits → clusters → tracks)
  - **Services**: long-lived shared resources (geometry/field access, calibration, logging, ROOT output, ACTS contexts)
  - **Event source**: reads input (simulation output, raw data) and injects objects into the event
- **EICrecon is a curated set of plugins** on top of JANA2:
  - Standardized data model (EDM4eic / PODIO-based objects) passed between factories
  - Detector-agnostic algorithms (incl. ACTS tracking) wired into a configurable reconstruction chain
  - For SoLID: swap in SoLID DD4hep geometry + SoLID-specific digitization/selection while keeping the same JANA2 flow

GEM geometry

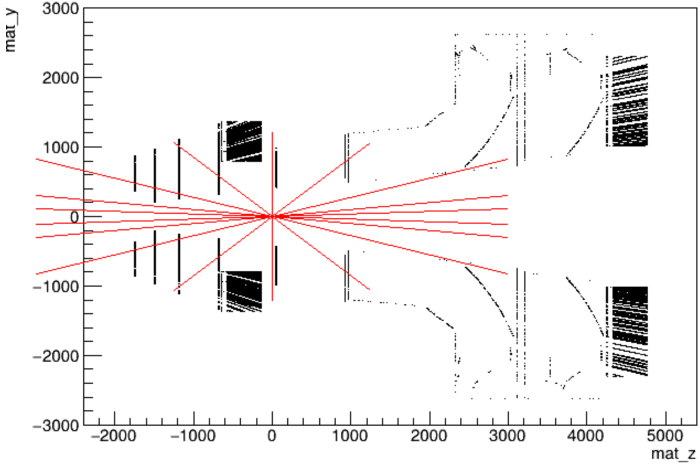
- Use trapezoidal structure to create the GEM disks.
- Build the material map for each tracking layer.

Location	Z (cm)	R_{min}^{needed} (cm)	R_{max}^{needed} (cm)
1	-175	36	87
2	-150	21	98
3	-119	25	112
4	-68	32	135
5	5	42	100
6	92	55	123

DD4hep implementation in SIDIS



Material map in SIDIS



Seed Finding: Overview

- Form seeds using the 3 most downstream GEM planes (lower occupancy / hit multiplicity).
- Use a “2-out-of-3” requirement to tolerate GEM inefficiency ($\sim$95%): any compatible hit pair among the three planes can initiate a seed.
- Two hits can seed because tracks originate from the target region (extra constraint), even though a full helix typically needs three points.

Double-Plane Seeding (`FindDoubleSeed`)

- Apply simulation-tuned pairwise compatibility cuts between two-plane hits:
  - ΔR cut between the two hits
  - $\Delta\phi$ cut between the two hits
- Estimate initial track parameters from the two hits via `CalInitParForPair`:
  - Hard-coded polynomial parameterizations derived from simulation
  - Use strong correlations, e.g. $\Delta\phi_{\mathrm{GEM}} \to (p, \theta, \phi)$

Fast Propagation and Validation

- Propagate the initial estimate quickly (coarse step size; accuracy not critical here).
- Use ACTS `Stepper` and `Propagator`.
- Cheap consistency checks:
  - Back-propagate to target: require rough compatibility with originating in target region.
  - Forward-propagate to ECal: require rough match to an ECal hit.

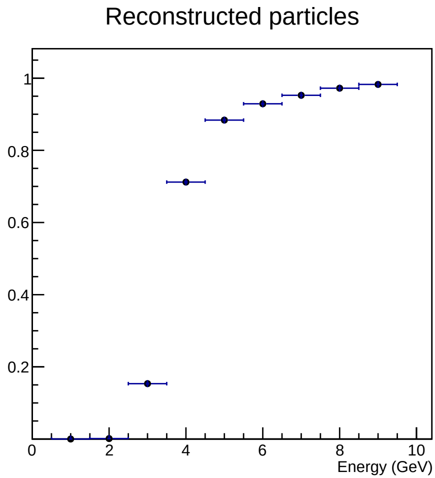
Seed Merging

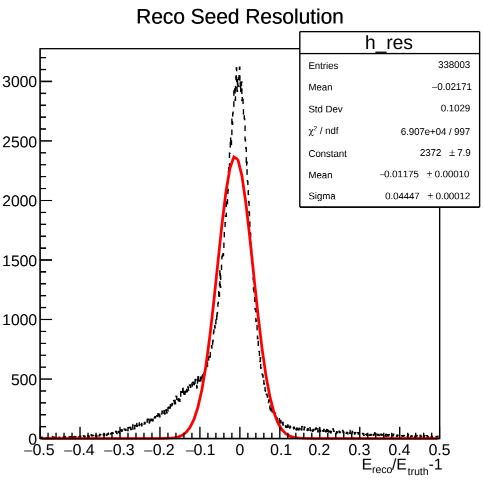
- After passing cuts, merge seeds likely from the same track using shared-hit logic.
- Example: a (GEM3,GEM4) seed and a (GEM4,GEM5) seed are merged if they share the same middle-plane hit (GEM4).

Tracking performance

- Use Weizhi's fit for $\Delta\phi_{\mathrm{GEM}} \to (p, \theta, \phi)$.
- Use ACTS `Propagator` instead of Runge-Kutta propagation.
- Only use the GEM detector, not use ECal.
- Updated from a uniform to a non-uniform magnetic field in seeding.

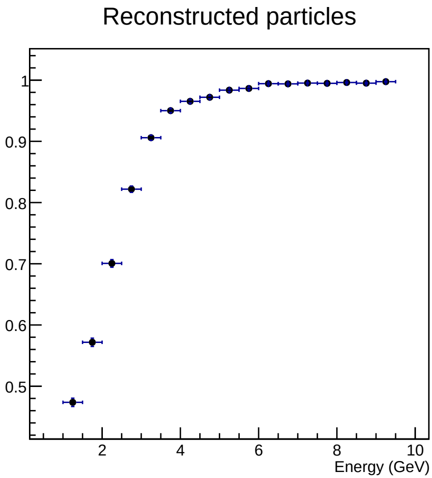
Tracking performance: previous reconstructed seed

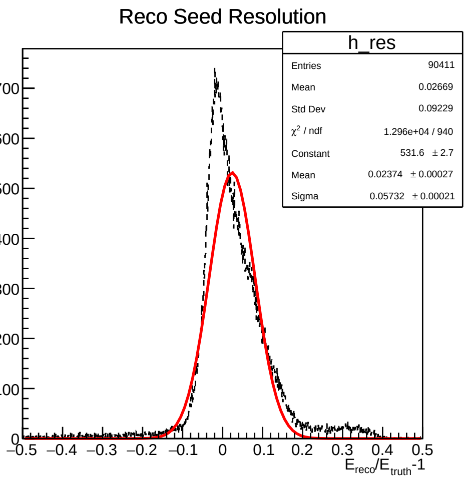




Previous reco seeding does not perform well.

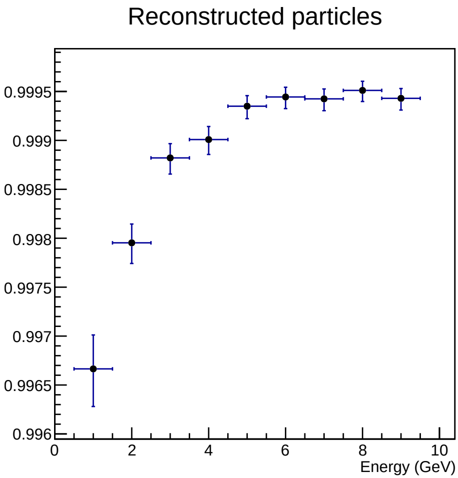
Tracking performance: updated reconstructed seed

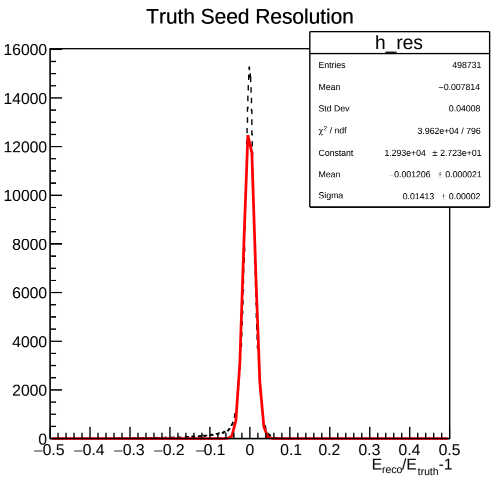




Using Weizhi's reco seeding performs better.

Tracking performance: truth seed





Truth seeding gives good performance. Track fitting works well.

Future plans

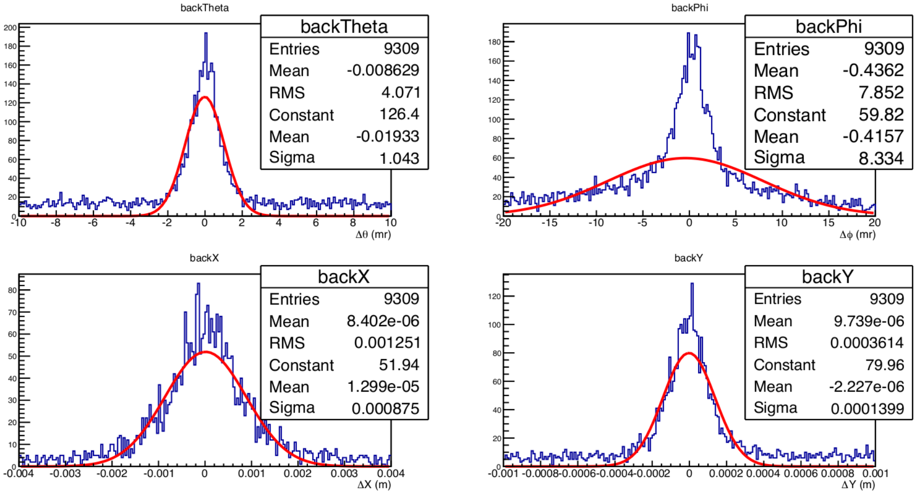
- Refit for $\Delta\phi_{\mathrm{GEM}} \to (p, \theta, \phi)$.
- Use Runge-Kutta propagation.
- Use ECal for seeding.
- Add background particles.
- Add PVDIS configuration in DD4hep and solid_recon.
- Use machine learning to improve efficiency.

Summary

- High occupancy in GEM tracker brings difficulty in track reconstructions.
- Large multiplicity leads a lot of false combinations in UV hits.
- How to use ACTS to help on
  - Denoising at high background?
  - Finding missing planes?
  - Implementing AI/ML for tracking finding and fitting?

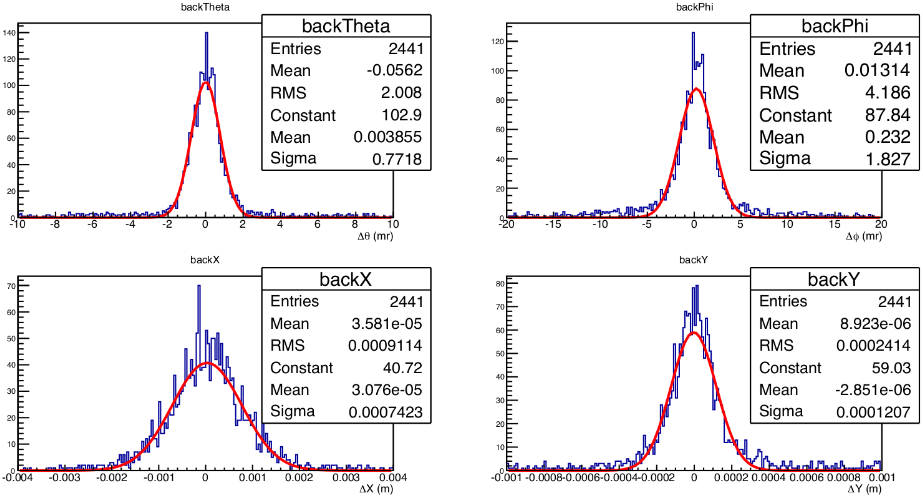
Backup

Resolutions of kinematic variables in PVDIS



Differences between reco and truth in PVDIS. No cut.

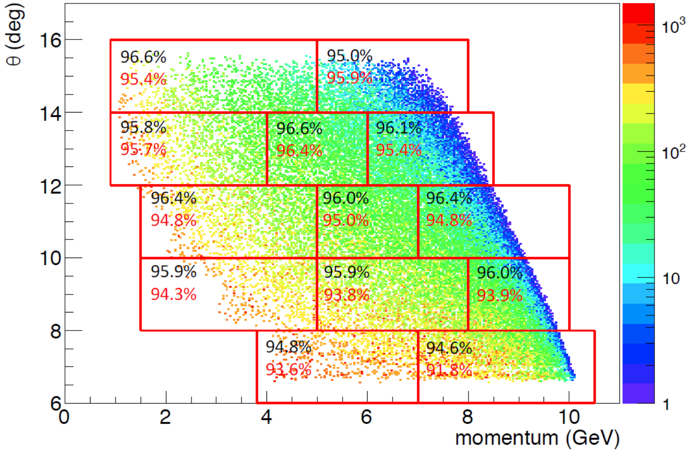
Resolutions of kinematic variables in PVDIS



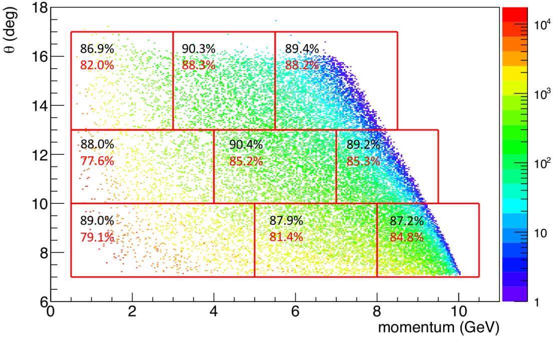
Require hit on every GEM tracker and $\chi^2/\text{ndf} < 8$.

Tracking efficiency and accuracy in SIDIS-^{3}He

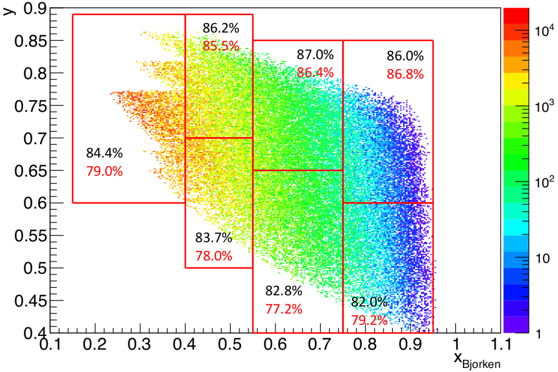
- Hits of an accurate track are all from hits of the MC particle.
- Track finding efficiency (black numbers) and accuracy (red numbers) for SIDIS-^{3}He.



Tracking efficiency and accuracy in J/ψ and PVDIS

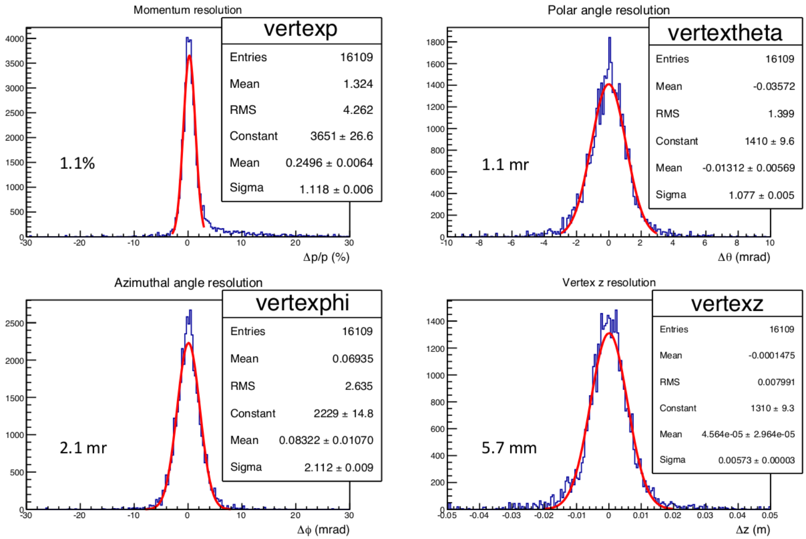


SIDIS-J/ψ

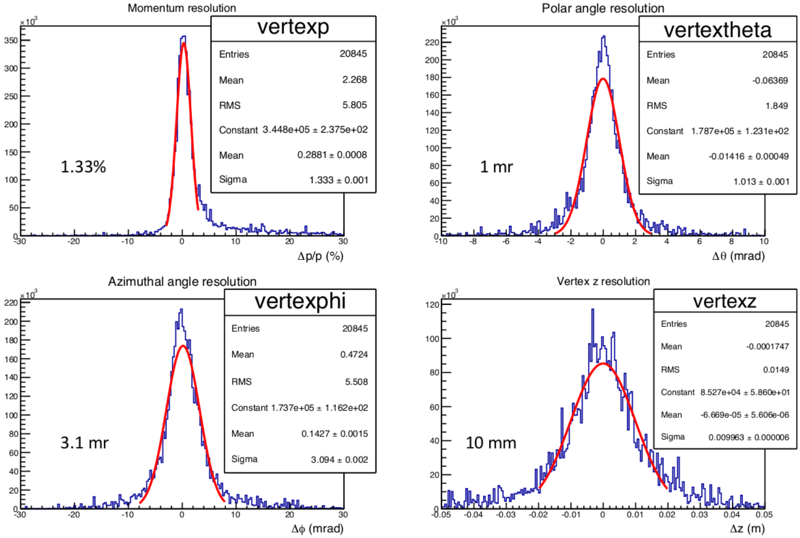


PVDIS

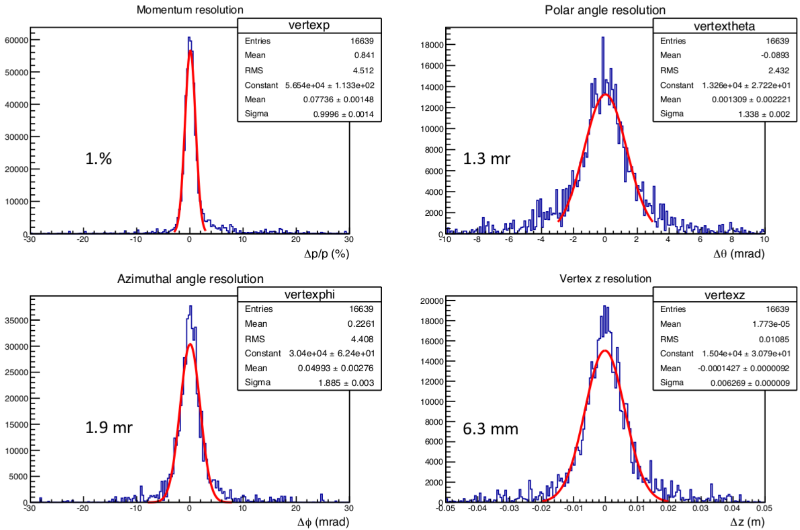
Resolutions in SIDIS at large angle



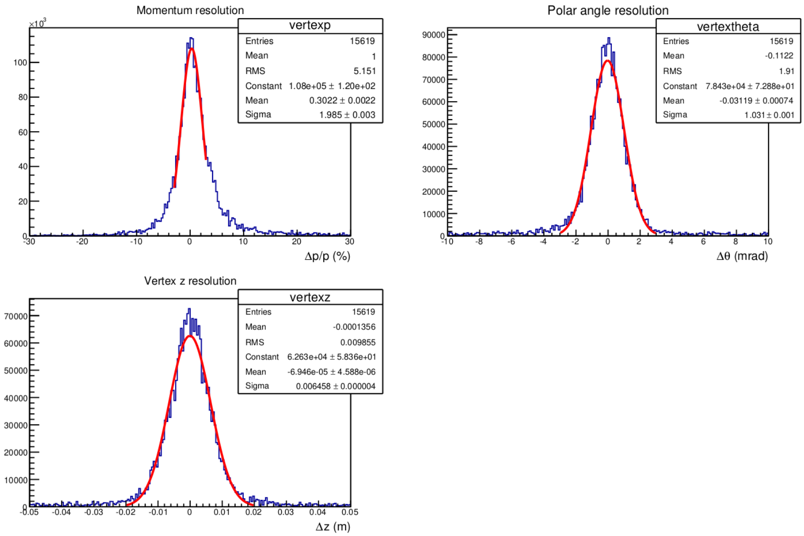
Resolutions in J/ψ at forward angle



Resolutions in J/ψ at large angle



Resolutions in PVDIS



Benchmark for KF propagation in PVDIS

- x position between prediction and measurement in KF propagation.
- Blue arrows indicate the direction of track finding.
- Large propagation distance leads to large error in Δx_2.