STAR Forward Upgrade Tracking & Software Status

02-28-2020

Daniel Brandenburg

Tracking Code & tutorial: https://github.com/jdbrice/star-fwd-dev

Simulation code & tutorial: https://github.com/jdbrice/star-fwd-sim

Outline

- Tutorial for new Tracking code
- Follow-up Tracking progress
  - Optimizing track finding parameters
  - Track fitting with/without Silicon in low mult
- Fitting without Primary Vertex
  - Estimate Vertex finding ability
  - Estimate for “global” tracks
- Tracking efficiency 10 π / event
- Other software tasks / status / discussion
- Summary + TODO Items

Forward Tracker Tutorial (Official StRoot Version)

1. **Obtain Code:**

   git clone https://github.com/jdbrice/star-fwd-dev.git

   cd star-fwd-dev

   git submodule update --init --recursive

2. **Run** in Docker (make sure to install)

   ./dev.sh --network none

```
[root@405ad708ce6a work]# pwd
/tmp/work
```

3. Build: ./build
4. Run: ./run <n_events> <input_file.fzd>

Produces efficiency & tracking QA plots in <output_file> defined in config

Anatomy of the Forward Tracker

- **fwd-dev** ← Main Project
  - star-sw/
    - StRoot/
      - StgMaker/ ← Code: StgMaker.cxx, StgMaker.h /include/Tracker/ = forward tracking code
      - StgcFastSim/
      - SiFastSim/ ← Not in master branch yet, TODO: write hits into StgMaker instead of StEvent to simplify integration
    - …
  - work/
    - config.xml ← Main configuration, use this to modify track finding / fitting parameters See next slide
    - run.sh
    - build.sh
    - …
  - dev.sh
  - README.md
  - Dockerfile

Forward Tracker Configuration

```xml
<?xml version="1.0" encoding="UTF-8"?>
<config>

    <Output url="results_si_refit.root" />

    <SiRasterizer active="true" r="3" phi="0.004090615" />

    <TrackFinder nIterations="1">
        <Iteration> <!-- Options for first iteration -->
            <SegmentBuilder>
                <Criteria name="Crit2_RZRatio" min="0.9" max="1.11" />
                <Criteria name="Crit2_DeltaPhi" min="0" max="10" />
                <Criteria name="Crit2_DeltaRho" min="-5" max="20"/>
                <Criteria name="Crit2_StraightTrackRatio" min="0.9" max="1.1"/>
            </SegmentBuilder>

            <ThreeHitSegments>
                <Criteria name="Crit3_3DAngle" min="0" max="90" />
                <Criteria name="Crit3_PT" min="0" max="100" />
                <Criteria name="Crit3_ChangeRZRatio" min="0" max="1.11" />
                <Criteria name="Crit3_2DAngle" min="0" max="2" />
            </ThreeHitSegments>
        </Iteration>

        <!-- These defaults are used if not defined inside <Iteration> -->
        <ThreeHitSegments>
            <Criteria name="Crit3_2DAngle" min="0" max="50" />
            <Criteria name="Crit3_PT" min="0.0" max="5" />
        </ThreeHitSegments>

        <Connector distance="1"/>

        <SubsetNN active="true" min-hits-on-track="4" >
            <!-- <InitialTemp>2.1</InitialTemp> -->
            <!-- <InfTemp>0.1</InfTemp> -->
            <Omega>0.99</Omega>
            <StableThreshold>0.001</StableThreshold>
        </SubsetNN>

        <HitRemover active="true">
        </HitRemover>

    </TrackFinder>

    <TrackFitter constB="false" display="false" noMaterialEffects="false" off="false"
  refitSi="true">
        <Vertex sigmaXY="0.05" sigmaZ="5" includeInFit="true" />

        <!-- for MC only -->
        <Hits sigmaXY="0.01" useFCM="true" />
    </TrackFitter>
</config>
```

- All tracking parameters are read from XML configuration file
- Easily study changes without modifying code

https://github.com/jdbrice/star-fwd-dev/blob/master/README.md

In the following I use the syntax `node.node:attribute`, e.g. the `sigmaXY` attribute of the `Vertex` Node inside `TrackFitter` would be `TrackFitter.Vertex:sigmaXY`

Input - not needed for version running in StRoot - it reads data from GEANT tables. In future we will read from hit collections

Output:url - specifies the output file for QA histograms written by tracking code. Once the code is fully integrated we will/should do QA on the tracks in StEvent/StMuDst.

Track Finding

Purpose: control the track finding parameters used by the CA procedure. All of these are implicitly prefixed by `TrackFinder` (i.e. inside the `TrackFinder` Node):

:nIterations - determines the number of iterations to run. An iteration is puctuated by track fitting and then removing hits that belong to accepted tracks.

Iteration[1..N] - Specify parameters for a given iteration, if they are not specified here, then the default will be used as defined in `TrackFinder`
Iteration.SegmentBuilder - parameters for the 2-hit segment builder. This is the first step in the CA track building process
Iteration.ThreeHitSegments - parameters for the 3-hit segment builder

Connector:distance - determines whether or not tracks can skip layers. For now we require all 4 sTGC hits, so no skipping is allowed. Should not be changed for normal studies.

SubsetNN - controls the Hopfield Neural Network used to select the best subset of tracks with highest quality that do not share hits.
SubsetNN:active - turn it on and off
SubsetNN:min-hits-on-track - specify the minimum number of hits required for a track seed.
SubsetNN.Omega - parameter controls weight of quality
SubsetNN:StableThreshold - controls how much noise is considered stable, changing may help speed up convergence

HitRemover:active - controls whether or not hits are removed after each iteration

Minimal Simulation Tutorial

- See: https://github.com/jdbrice/star-fwd-sim

1. Obtain Code:

git clone https://github.com/jdbrice/star-fwd-sim.git

cd star-fwd-sim

2. Run starsim (uses docker or run on RCF)

./run

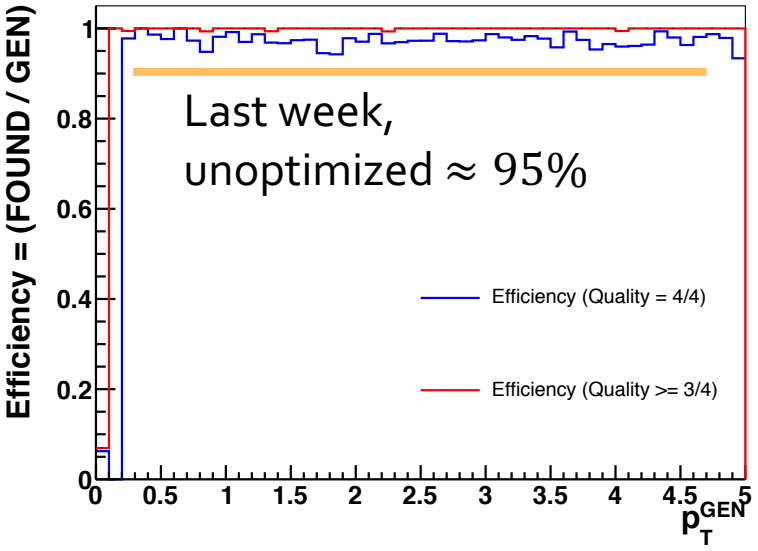
This runs e.g.:

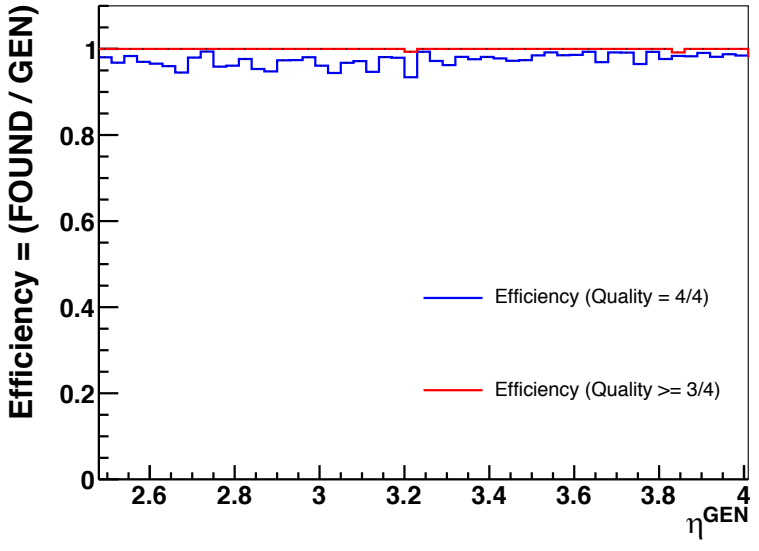
starsim -w 0 -b testg.kumac nevents=1 ntrack=100 etamn=2.4 etamx=4.1 ptmn=4.5 ptmx=5.0

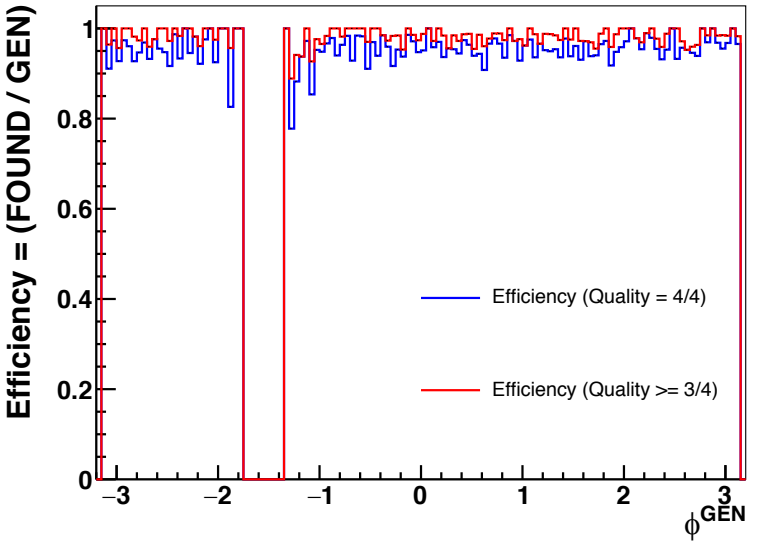
Produces testg.fzd – can be used with tracking code

TODO: Run pythia + HIJING samples for all to use for studies

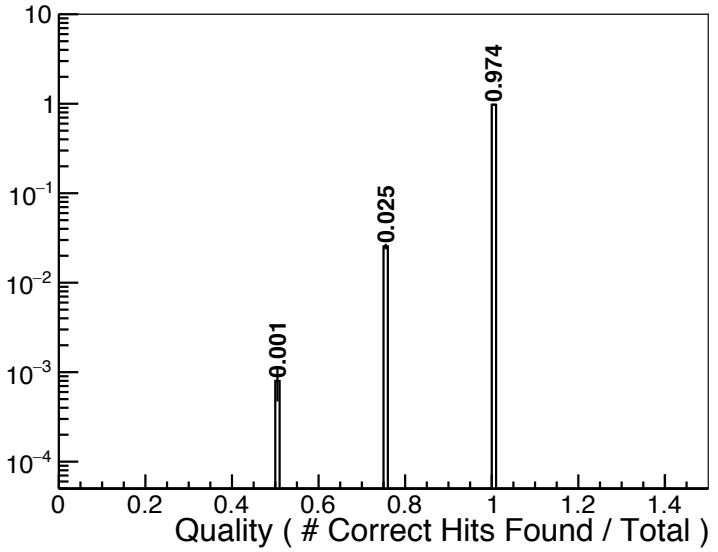
Track Finding Efficiency Low Mult (1 track / event)





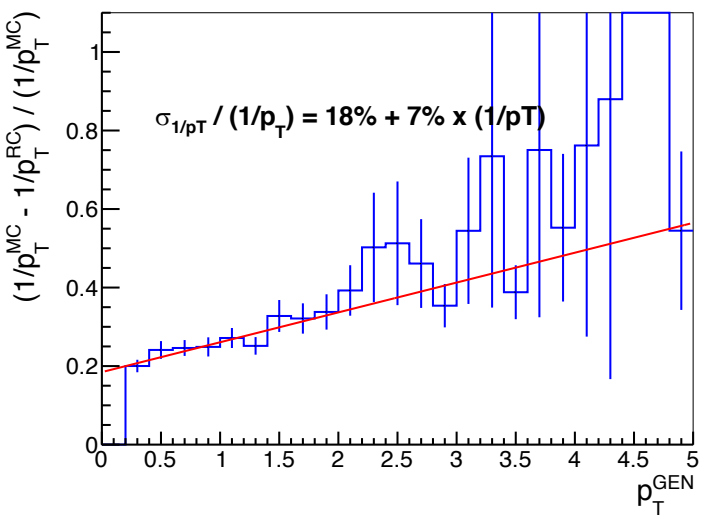


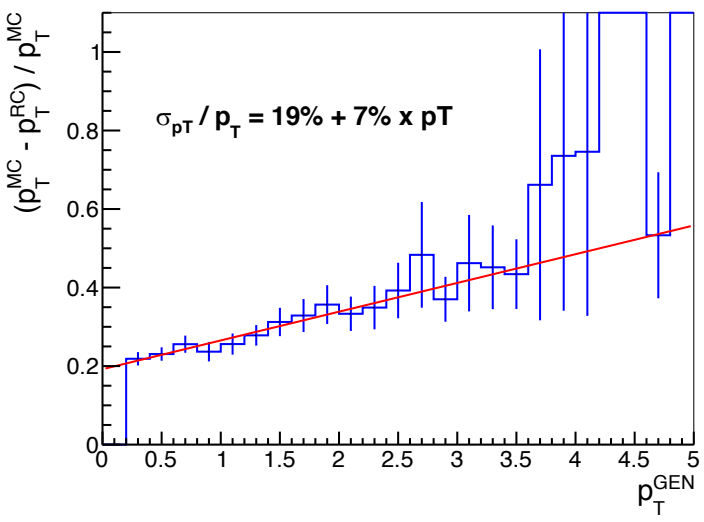
- Performed some optimization since last week
  - Track finding efficiency (perfect 4/4 correct hits) is ≈ 98%
  - Track finding efficiency (3/4 or more correct hits) is ≈ 99.5%
- Full material effects
- Real STAR B-field
- Sharp turn-on curve for the efficiency vs. pT – full efficiency by pT ≈ 200 MeV/c

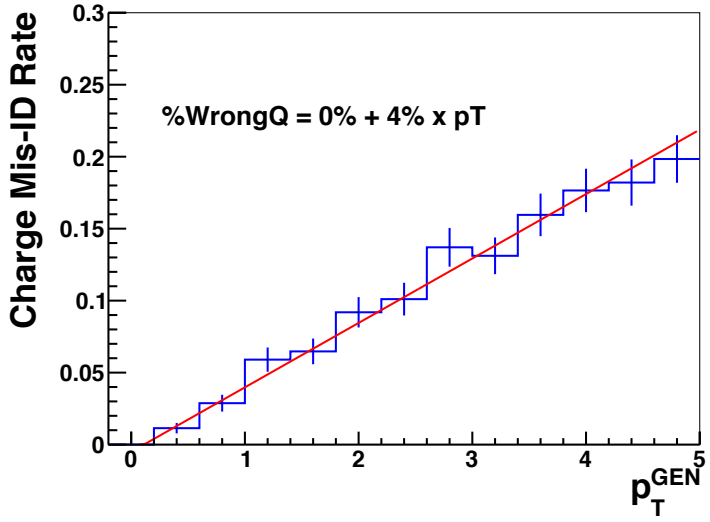


Track Fitting Performance with Primary Vertex $\sigma_{XY} = 500\ \mu m$

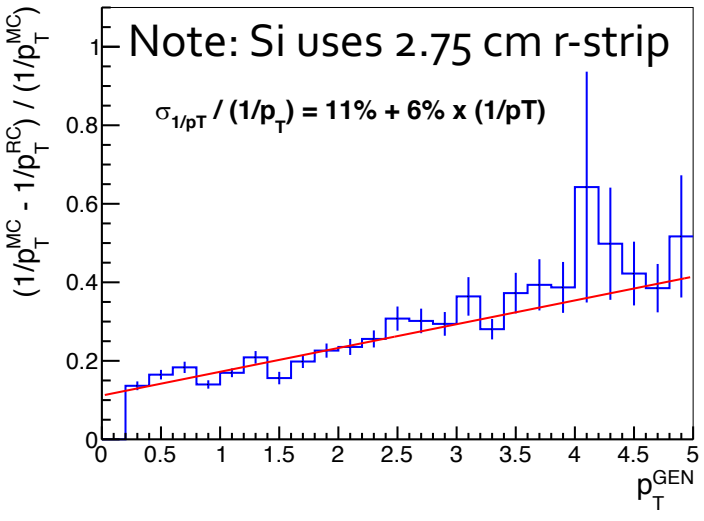
sTGC + PV

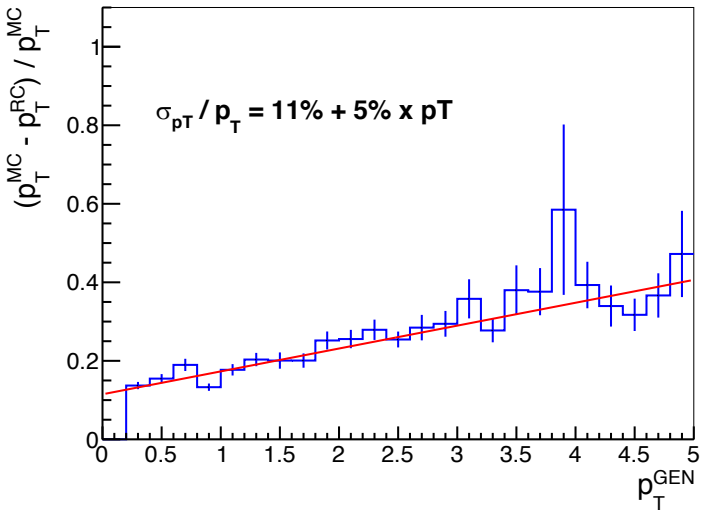


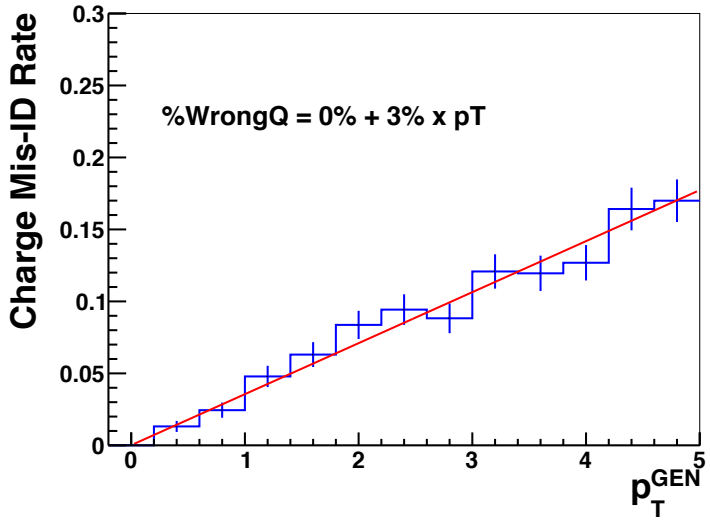




Refit 3 Silicon hits

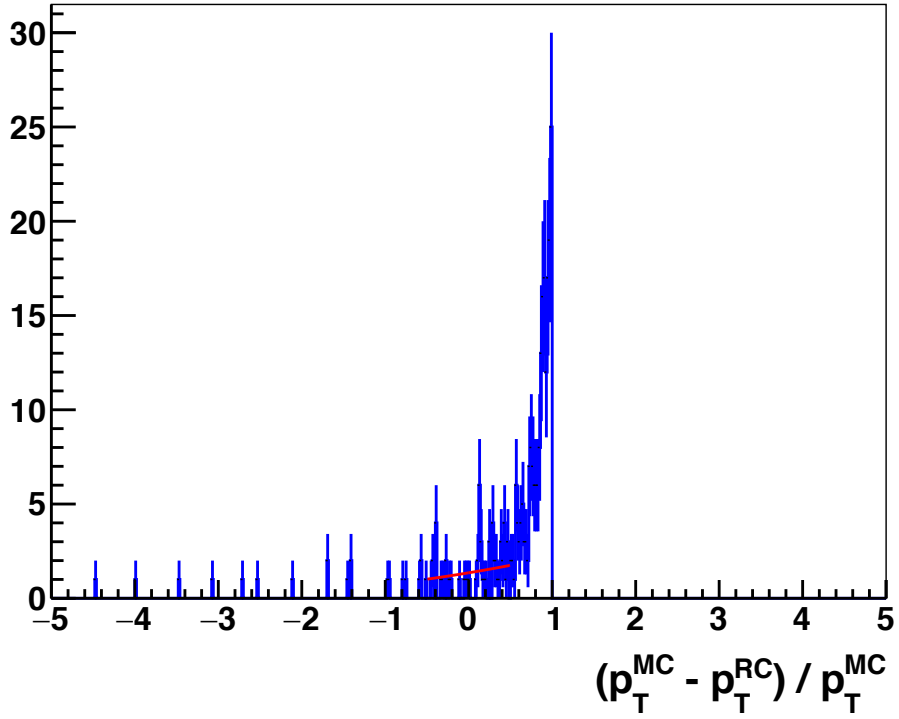


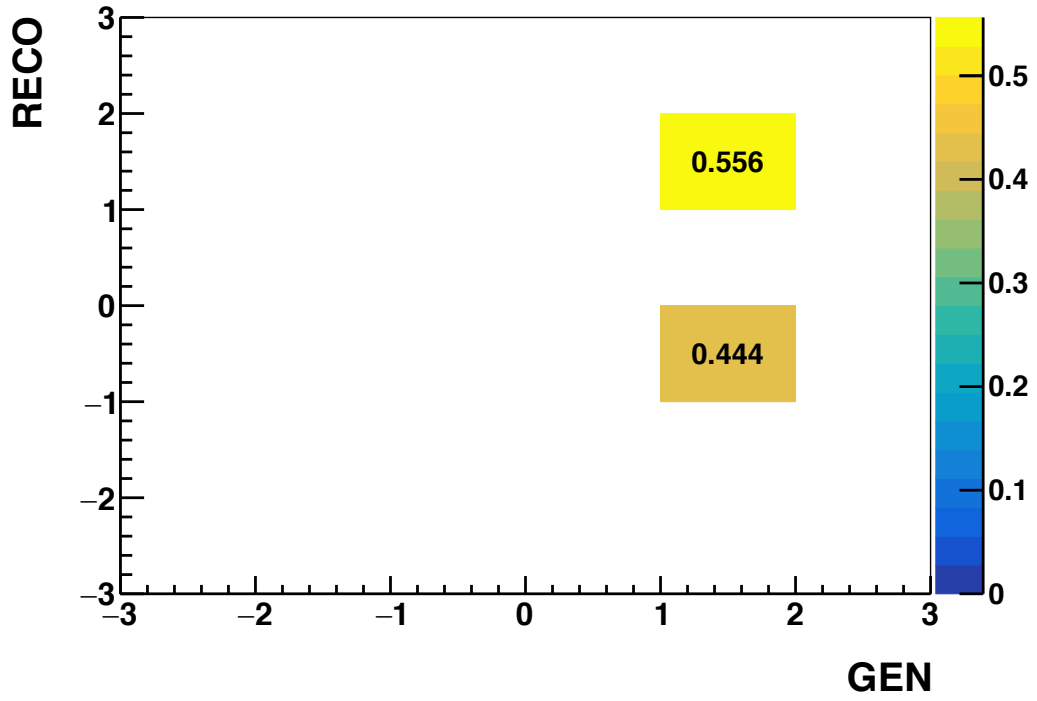




Track Fitting without Primary Vertex

- Very poor constraint on track parameters with only 4 sTGC hits

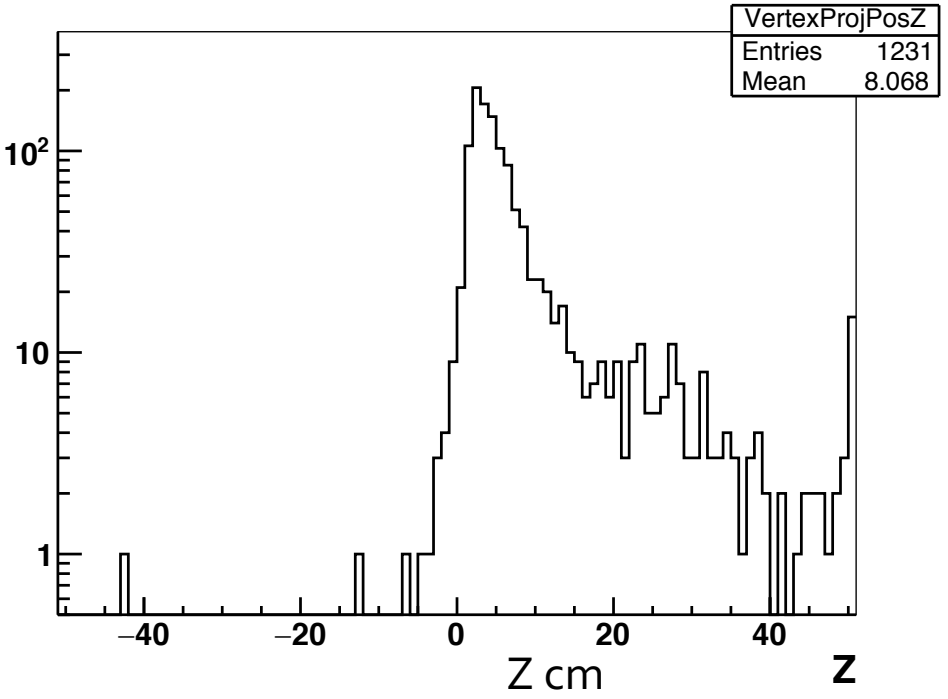


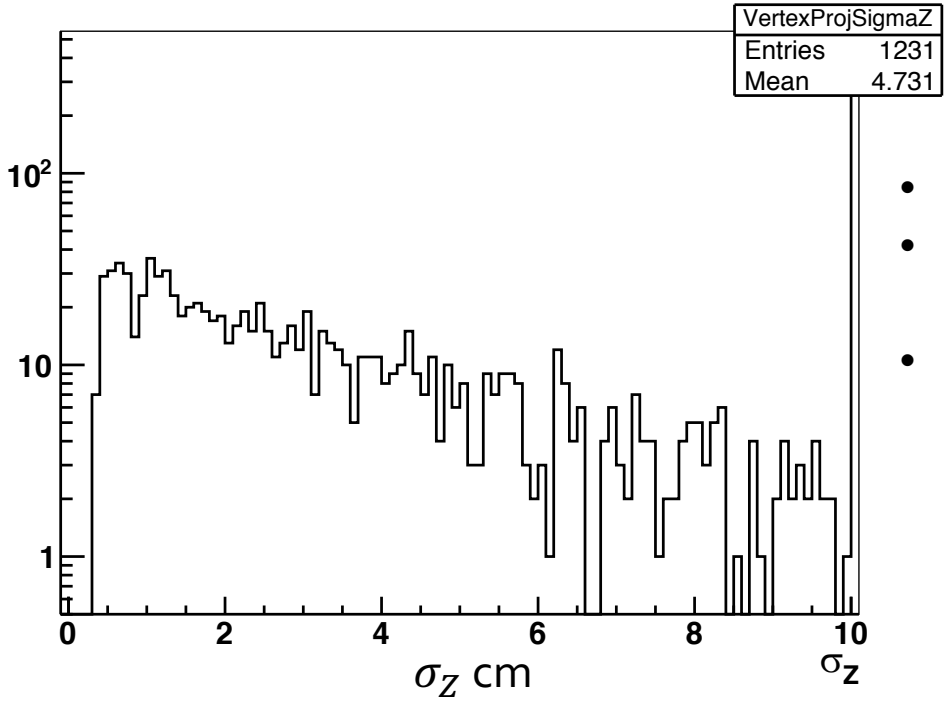


- Even charge discrimination is not good, but can we improve this?
- With such large uncertainty, projection to Si has huge uncertainty

Situation **without** Primary Vertex from TPC

- Rough estimate of how well we can determine primary vertex with Forward detectors only:
- Track fitting without PV → project tracks to beamline $(0, 0, z)$
- Only using sTGC hits for now

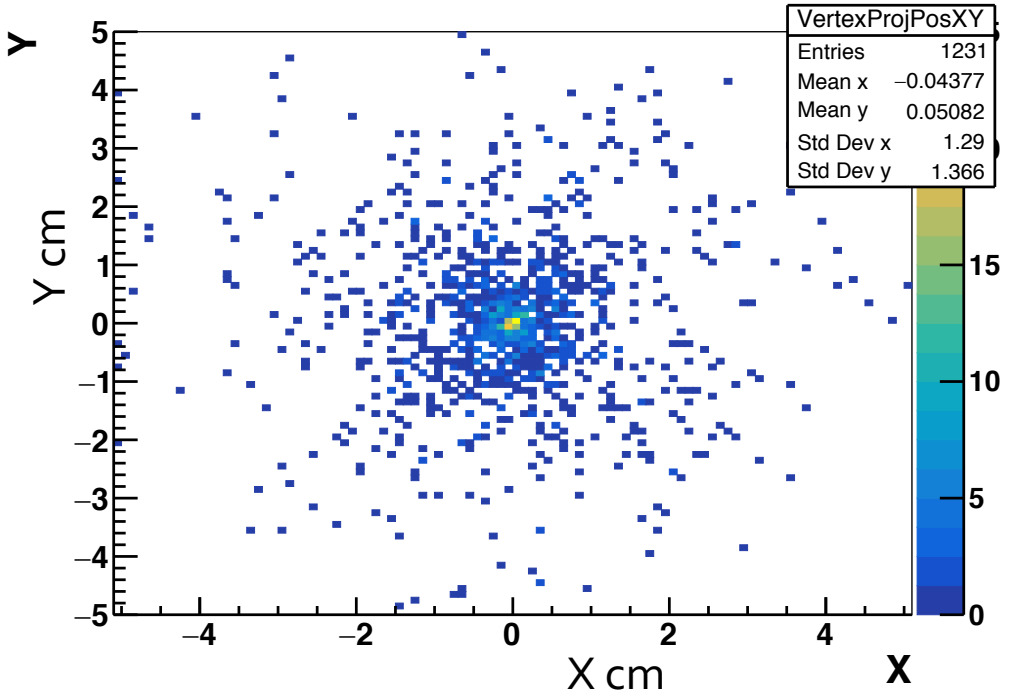


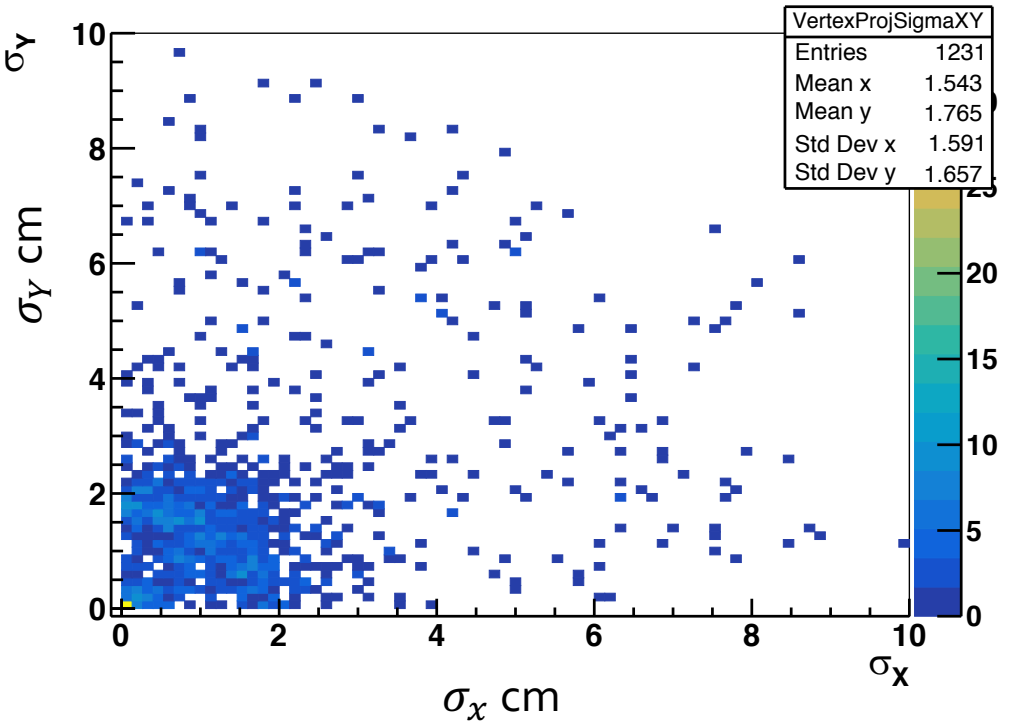


- Bias to positive Z
- Strong η dependence, needs more study
- GenFit includes RAVE – perform full vertex finding/fitting

Situation **without** Primary Vertex from TPC

- Rough estimate of how well we can determine primary vertex with Forward detectors only:
- Track fitting without PV → project tracks to beamline $(0, 0, z)$
- Only using sTGC hits for now



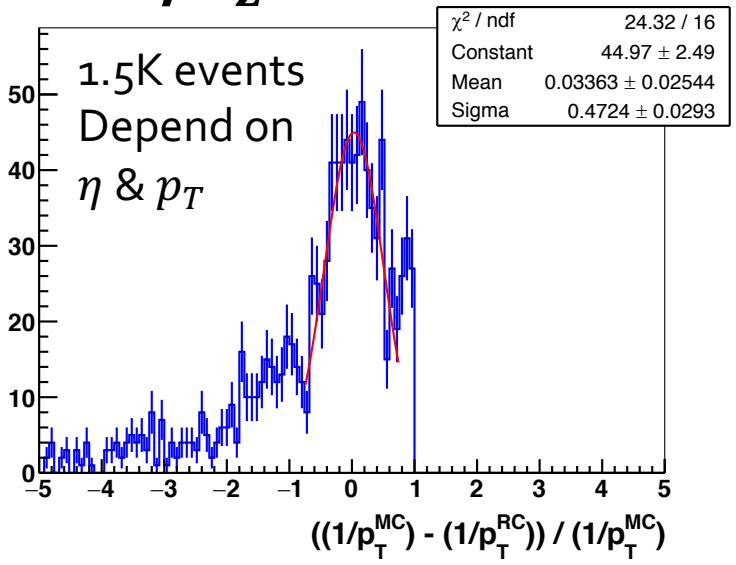


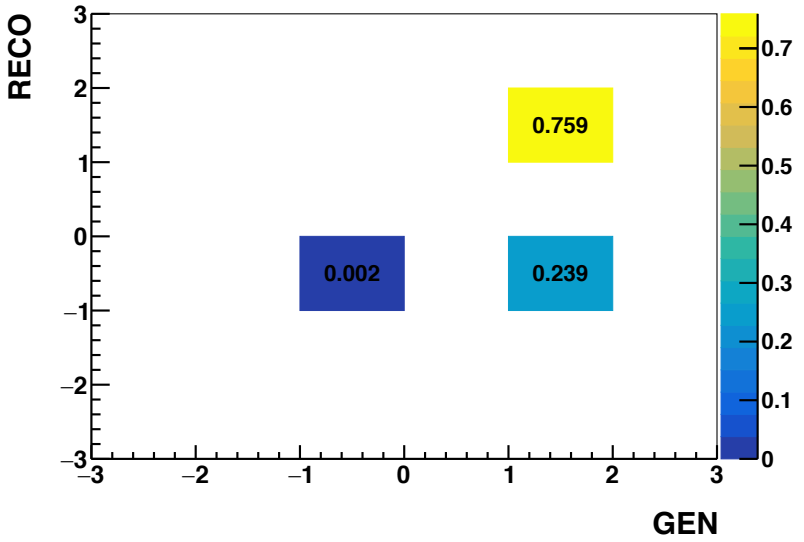
- No visible bias
- Strong p_T dependence, needs more study
- GenFit includes RAVE – perform full vertex finding/fitting

Track Fitting Performance **without** Primary Vertex

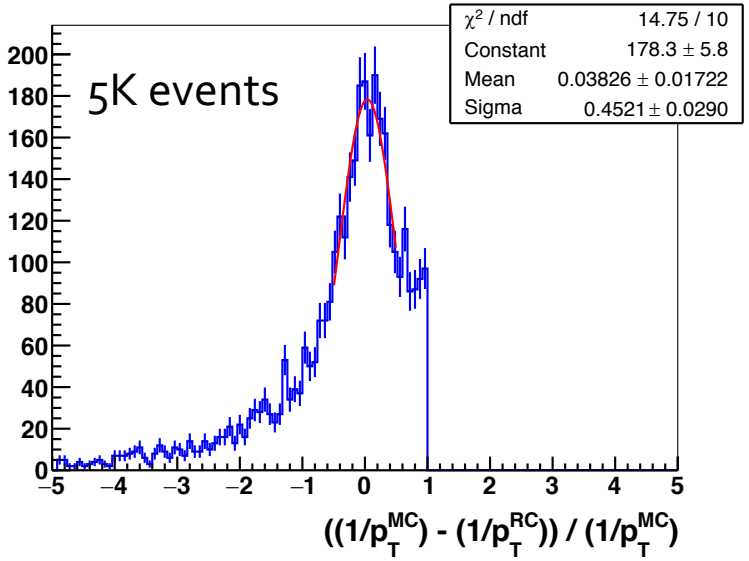
$\sigma_{XY} = 2\text{mm}, \sigma_Z = 100\text{ cm}$

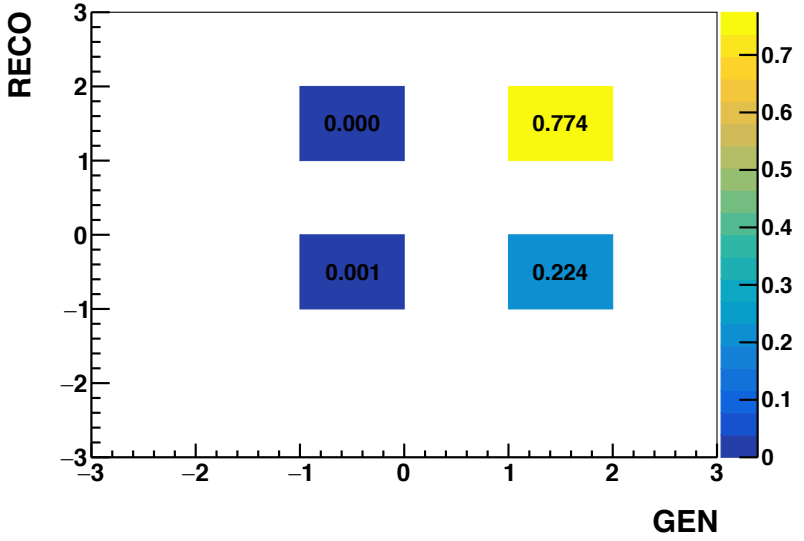
sTGC + PV





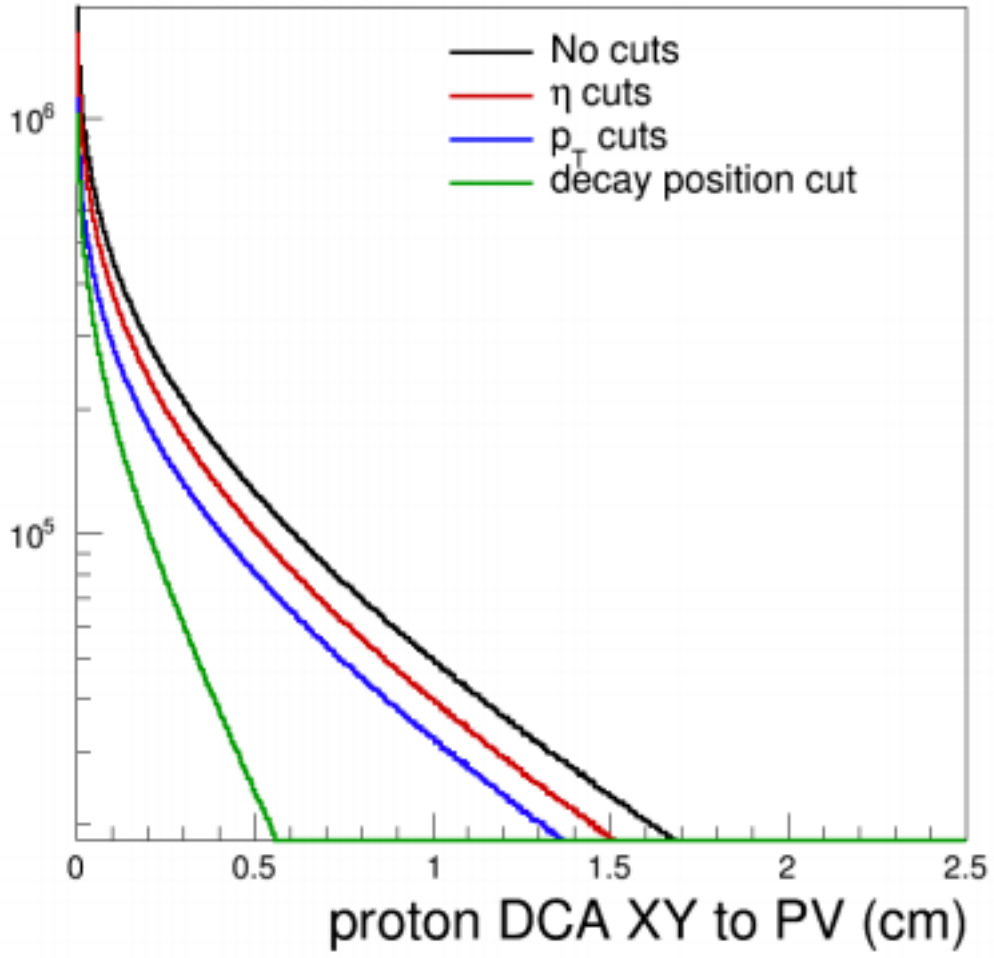
Refit 3 Silicon hits

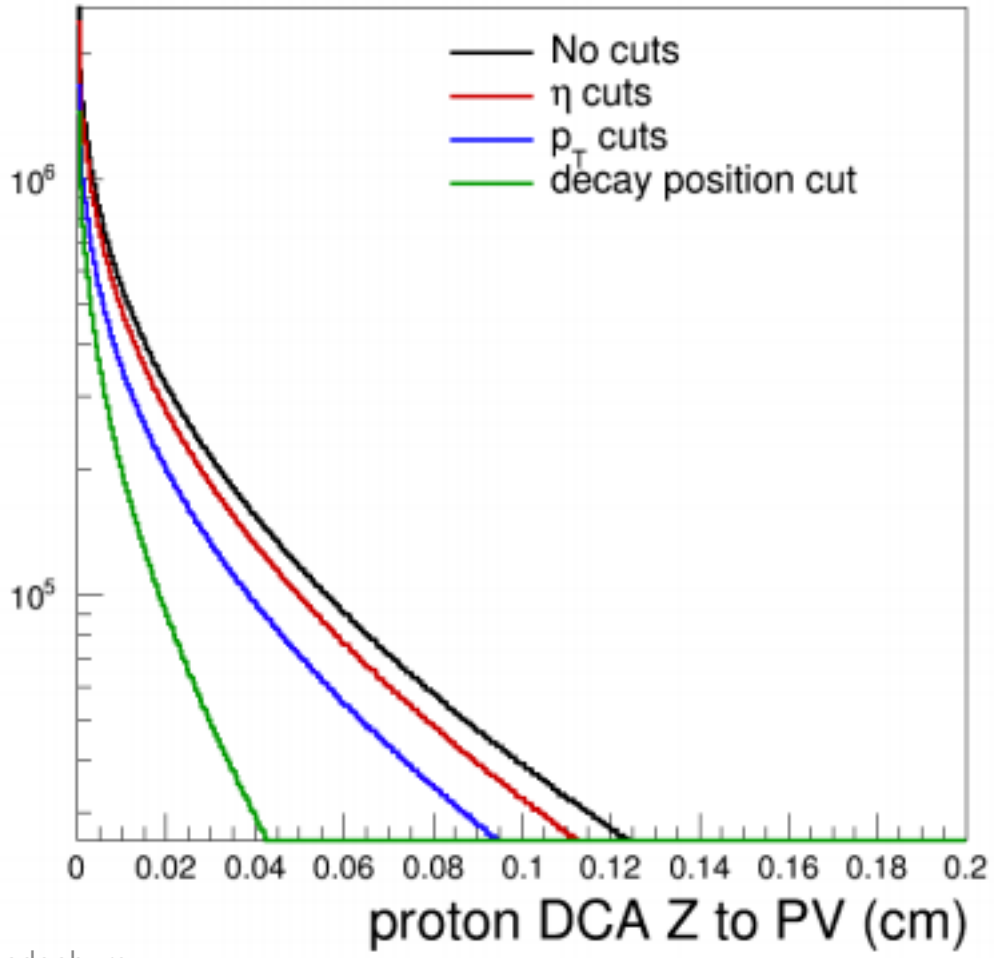




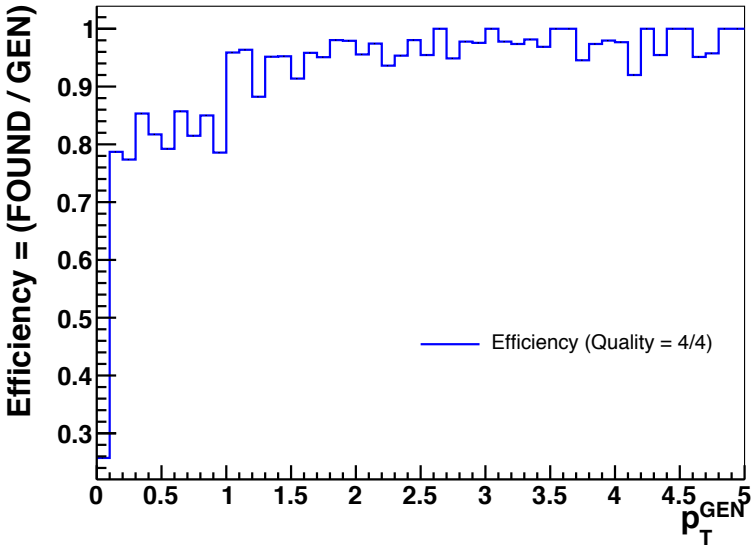
Tracking for Secondary (global tracks)?

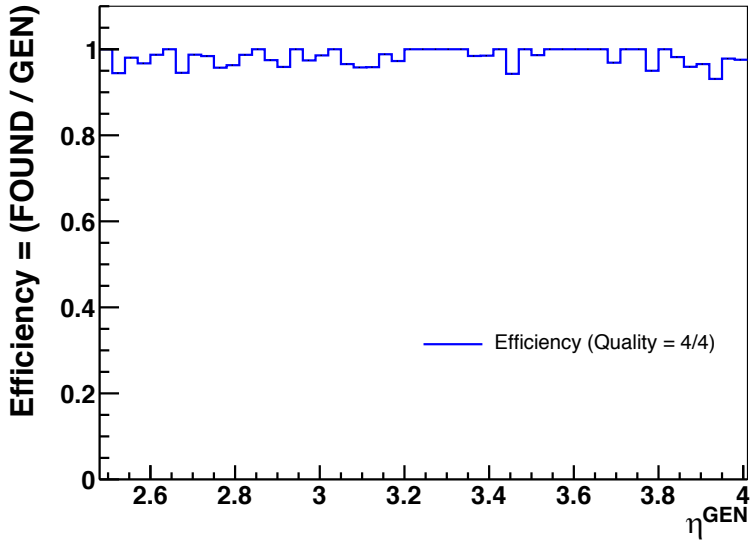
- Can we improve track fitting for secondary (global tracks) by imposing a primary vertex with large uncertainty? Can be studied with existing version

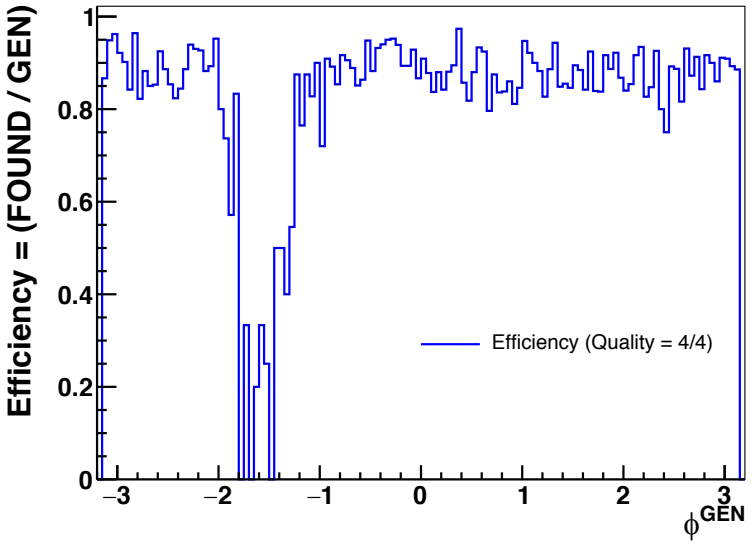




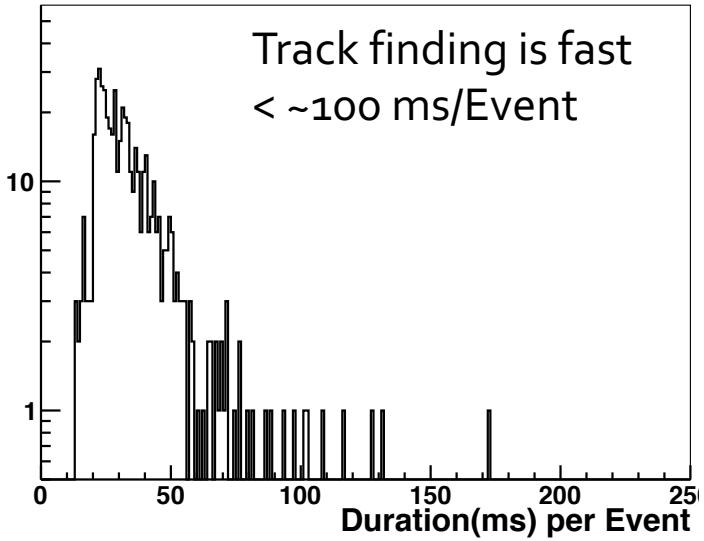
Track Finding Efficiency Low Mult (10 π^+ / event)

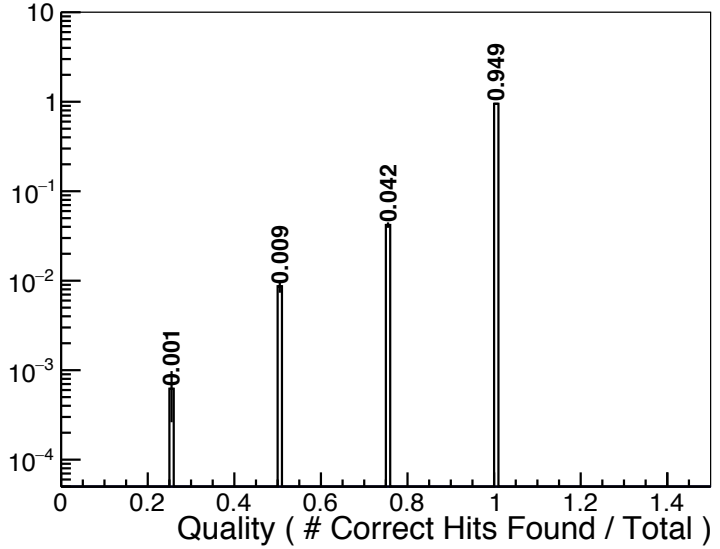






- No optimization, just default parameters
- Plateau efficiency ~100%, slight issue for pT < 1 GeV/c – can be optimized + improved
- Full material effects
- Real STAR B-field

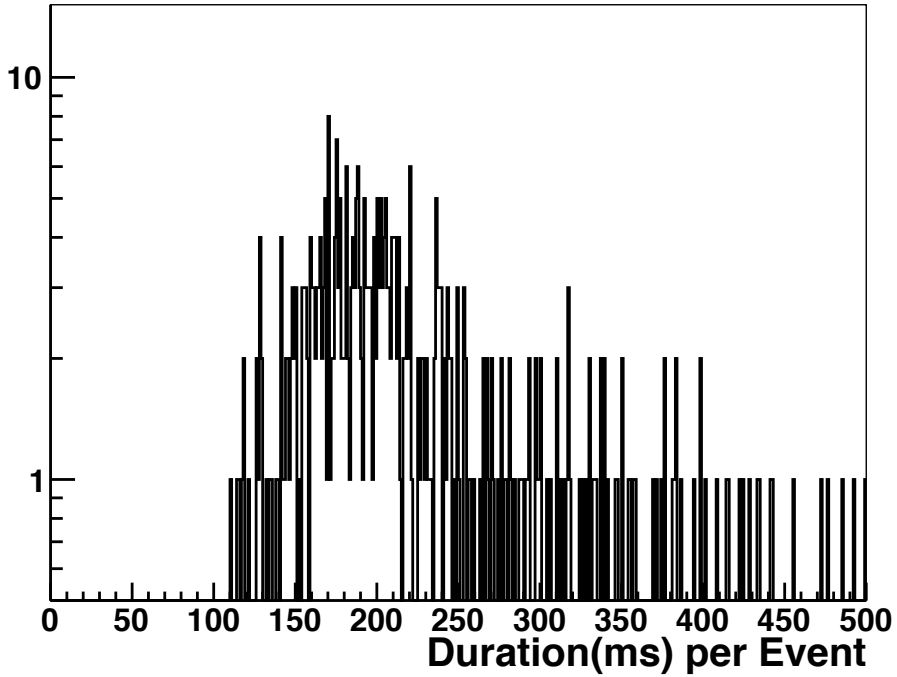




Performance (10 π^+/event) with Primary Vertex $\sigma_{XY} = 500\ \mu m$

Work in progress (Tasks I am working on):

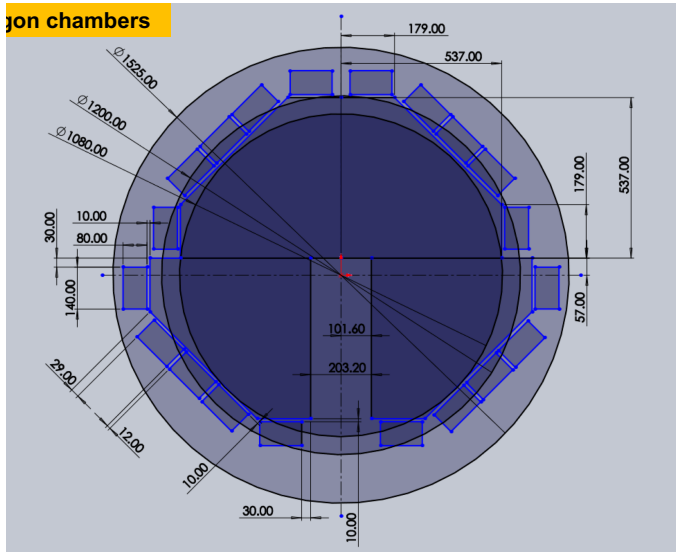
- Incrementally adding Si hits (add nearest disk first, then next disk)
- Add all nearby hits, let GenFit determine "outliers" – maybe better than providing only nearest hit
- Most events take ~ 200ms / event (including fitting) but some (1%) get "stuck" for 1 second or more
  - Add additional stopping conditions to catch these.
  - Note: these are for debug builds with debug histogram
  - Work on tuning performance (task for student)

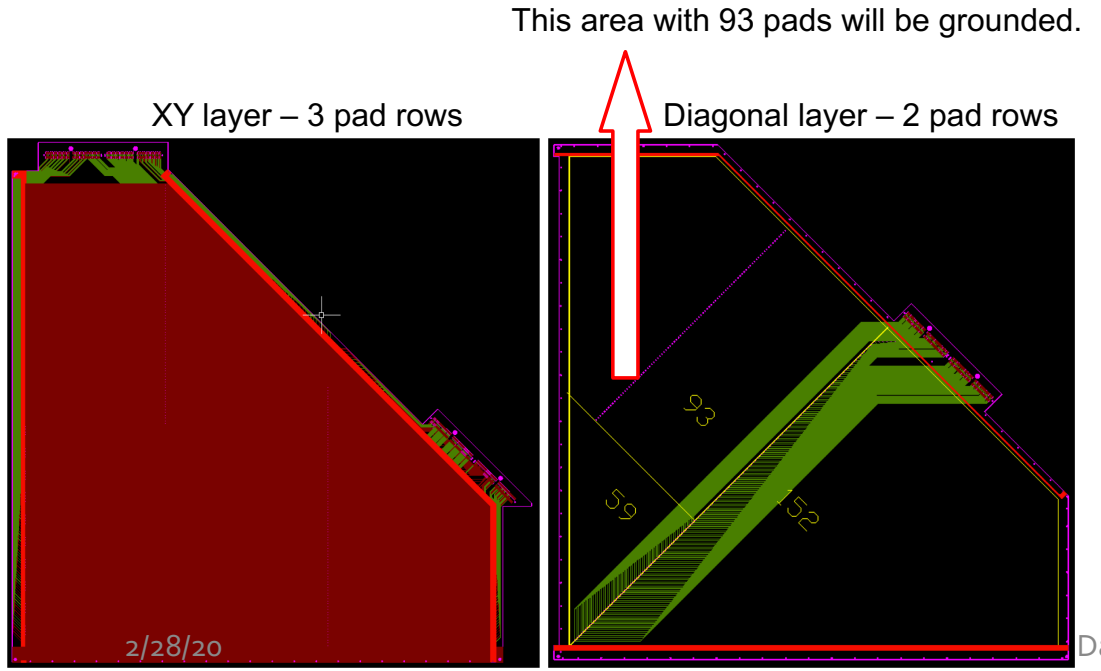


Software Status : Tracking

- Forward Tracking : good progress, v1 available
  - Some details: update Si hits to use z-position from inner/outer etc. → small improvement in fit results
  - Track refitting with Silicon : continue improving + adding missing cases
  - Full Vertex Finding/Fitting with forward tracks only
  - Finalize integration in StRoot (tracks into StEvent)
- Several tasks that others can work on:
  - Machine Learning studies (CA optimization , ghost hit rejection, …)
  - Optimize parameters for higher mult events (CA parameters, # iterations, phi-slices, …)
  - Physics studies (Lambda, Drell-Yan, etc.) + Tracking performance for Pythia & HIJING Events
  - Investigate code slow-downs + optimized builds

sTGC Geometry + Fast Simulator

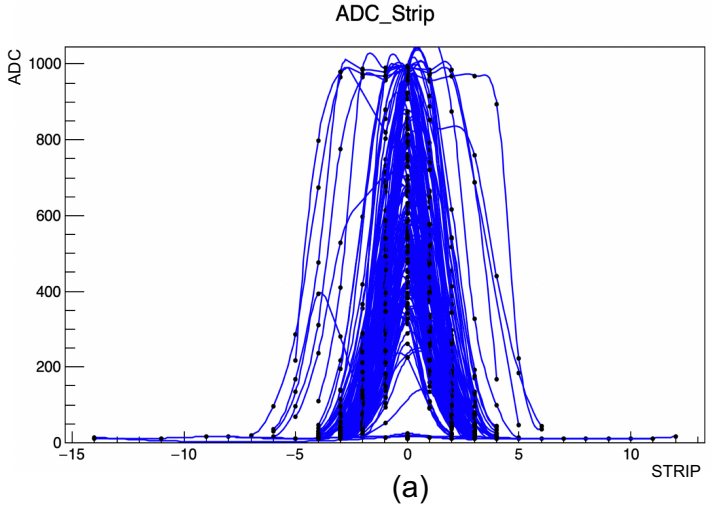




- Symmetric pentagon chambers
  - Design is final? (Raul mentioned slight movement in y…)
  - Update sTGC geometry
- Fast simulator
  - Update digitization based on new geometry
  - I need to know detailed layout of strips
  - Add diagonal strips + logic to combine XY + U/V
- Database
  - Simulation, geom / electronics maps etc.

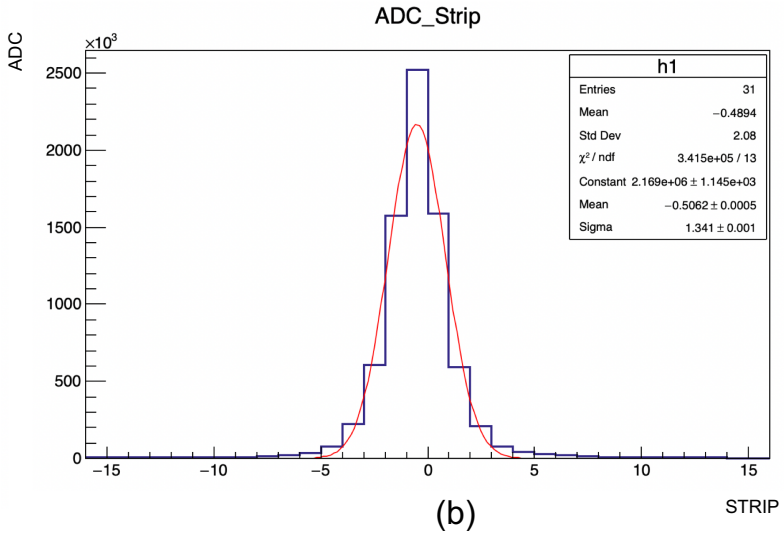
Cluster Finder for sTGC

- Lijuan's action items:
  - Software simulation: need to develop clustering algorithm with high efficiency (prototype run in 2020)



(a)

1. For each event, get the TB of MaxADC.
2. At same TB, the ADC distribution varies with the strip.



(b)

The sumADC distribution of each event at the same the strip.

STRIP number >3sigma ➔ STRIP>=4

From Chi's presentation yesterday

One (two) ways to work on this now:

1. Add sTGC "Slow" simulator that produces clusters from each hit
   - Use Chi's results shown yesterday – prototype
   - Write cluster finder + gravity center method to determine hit position
2. Revisit data from last year?
   - Is it even useful since it was a different gas / low efficiency?

- Akio has working cluster finder to copy as first version, we could also use adapted fastjet anti-kt like algorithm? What is atlas doing?

Data format and StEvent + other integration tasks

- StEvent : stores raw data formats from detectors
  - Calorimeters → Akio (ready to go)
  - Silicon : do we already have electronics data format ? If not then when can we expect this?
  - sTGC: already setup for TPC-like format
    - VMM electronics will be different
      - Timebins?
    - When can we finalize the data format for sTGC
- @Some point: determine framework for CAL matching
- Reminder that "Makers" will need S&C review
  - E.g. sTGC Cluster maker, Silicon HitMaker etc.

  - If we can get skeletons of these in place (i.e. with simulation) we can do S&C review and update later with improvements

```
class StFtsStgcHit : public StObject {
  ...
 protected:
        UShort_t mRdo=0;
        UShort_t mSec=0;
        UShort_t mAltro=0;
        UShort_t mFEE=0;
        UShort_t mCH=0;
        UShort_t mNTimebins=0;
        TArrayS* mAdcs=0;
        TArrayS* mTimebins=0;

        ClassDef(StFtsStgcHit,0)
};
```

Tracking + Simulation Tasks for Silicon

- Newest geometry
  - Update from Te-Chuan last week.
  - Only one possible issue - new geometry has overlaps again (not sure if it is in Si part or elsewhere) – Jason is checking
- In addition to general tracking performance studies, what additional studies are needed for Silicon at this stage?
- Fast Sim :
  - ✓updates from Te-Chuan
  - ○ Update to fill collection in StgMaker instead of StEvent (small task)
- Slow simulator
  - Discussion
  - What priority is this –
  - are there parts we can do easily/quickly that is better than existing Fast sim
  - Preference is to get skeleton in place then improve as it is needed

Online Plots & Real-time monitoring + HLT

- At some point we want Jevp online plots for all systems
  - sTGC : we should add these for the in-beam test (if we agree on that)
  - Silicon – not yet
  - Calorimeters: Feedback from Akio + others?
  - …
- HLT
  - Can we benefit from using HLT for in-beam test
  - Long term use? We can start developing HLT algorithms

Summary & Task List

Summary:

- Tracking V1 : Real Geometry + Mag field, PV + sTGC + Si
- Tutorials available
- Progress on other software integration tasks, still lots to do (below)

Tasks (lets discuss and produce prioritized list + manpower):

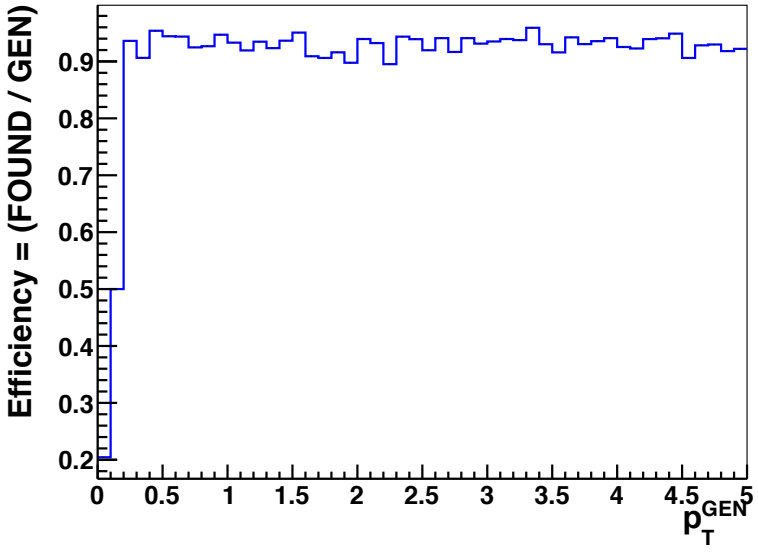
- Finish Tracking with Silicon refitting + misc tracking improvements (Daniel)
- Vertex finding/fitting with forward tracks only + EPD/BBC (Daniel)
- 10x Tracking studies (others)
- sTGC : update geometry / Fast Sim, study ghost hits
- sTGC : sTGC slow-sim (Cluster simulation), cluster finder
- Si-specific studies? Slow simulator, databases, mis-alignment etc.
- Electronics data formats (sTGC, Si, CALs) into StEvent
- Tracking with + Matching to Calorimeter hits
- …

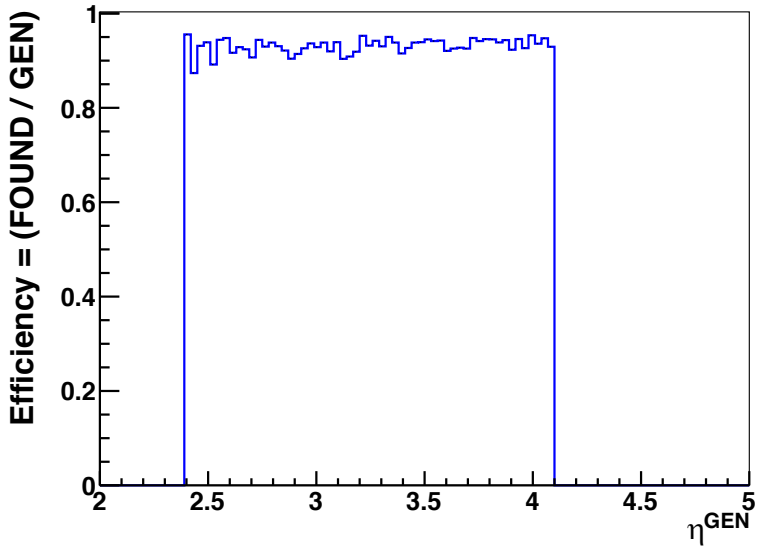
Monte Carlo parameters

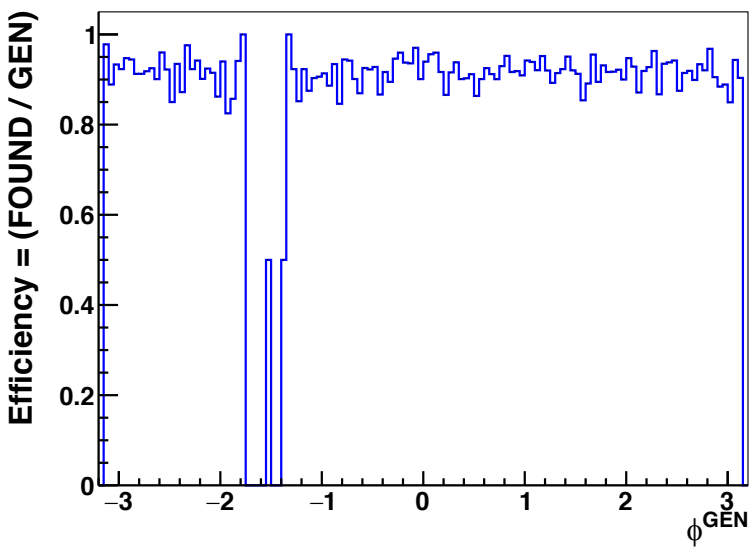
- GENERATOR ("GEN" or "MC" hits):
  - $1\ \mu^+$ / Event
  - $2.45 < |\eta| < 4.05$
  - $0.2 < p_T < 5\ GeV/c$
  - B Field : **REAL** (StarMagField)
  - Primary Vertex distribution $\mu = (0, 0, 0)$, $\sigma = (0.05, 0.05, 5)$ cm.
- ("REAL" hits):
  - Si : Uses 3 cm r-strips with 128*12 divisions in phi
  - sTGC : 15 cm strips, $\sigma_X = \sigma_Y = 100$ microns.
    - No diagonal strips (no studies of ghosts in this presentation)

Meant to be a best case scenario for realistic detector resolutions.

Track Finding Efficiency (Default Parameters)

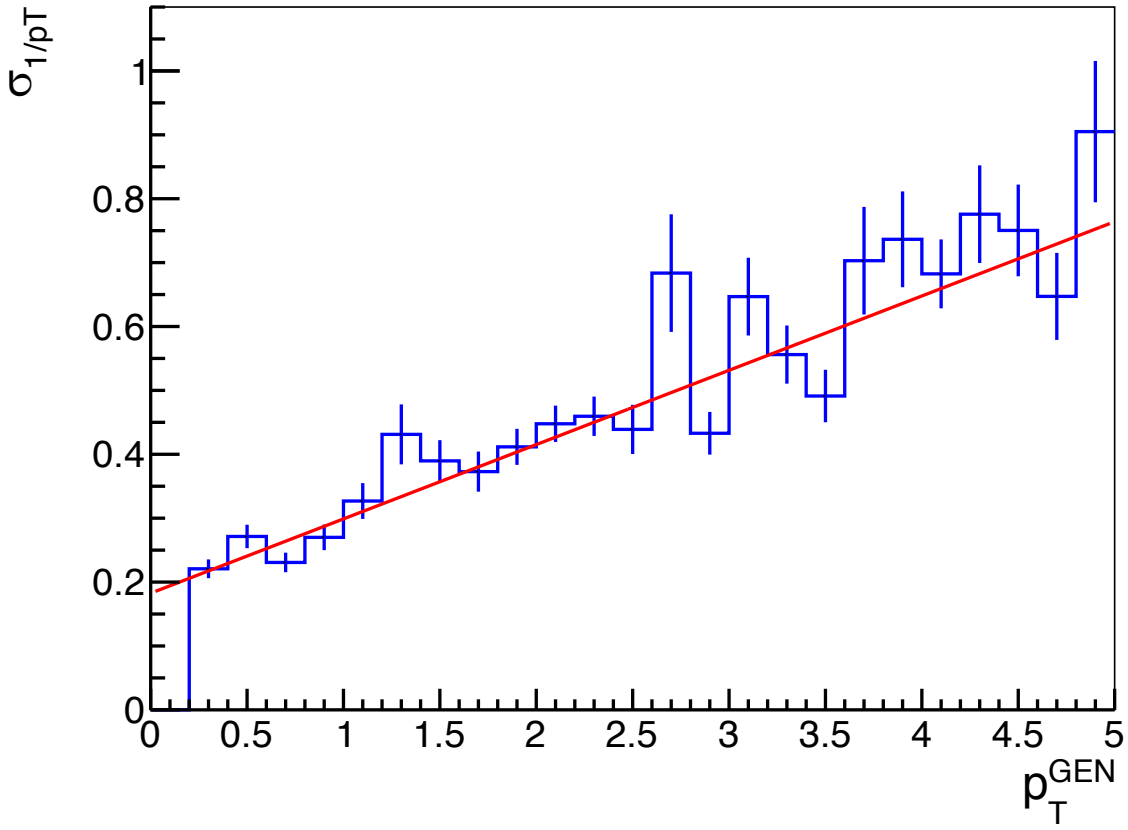


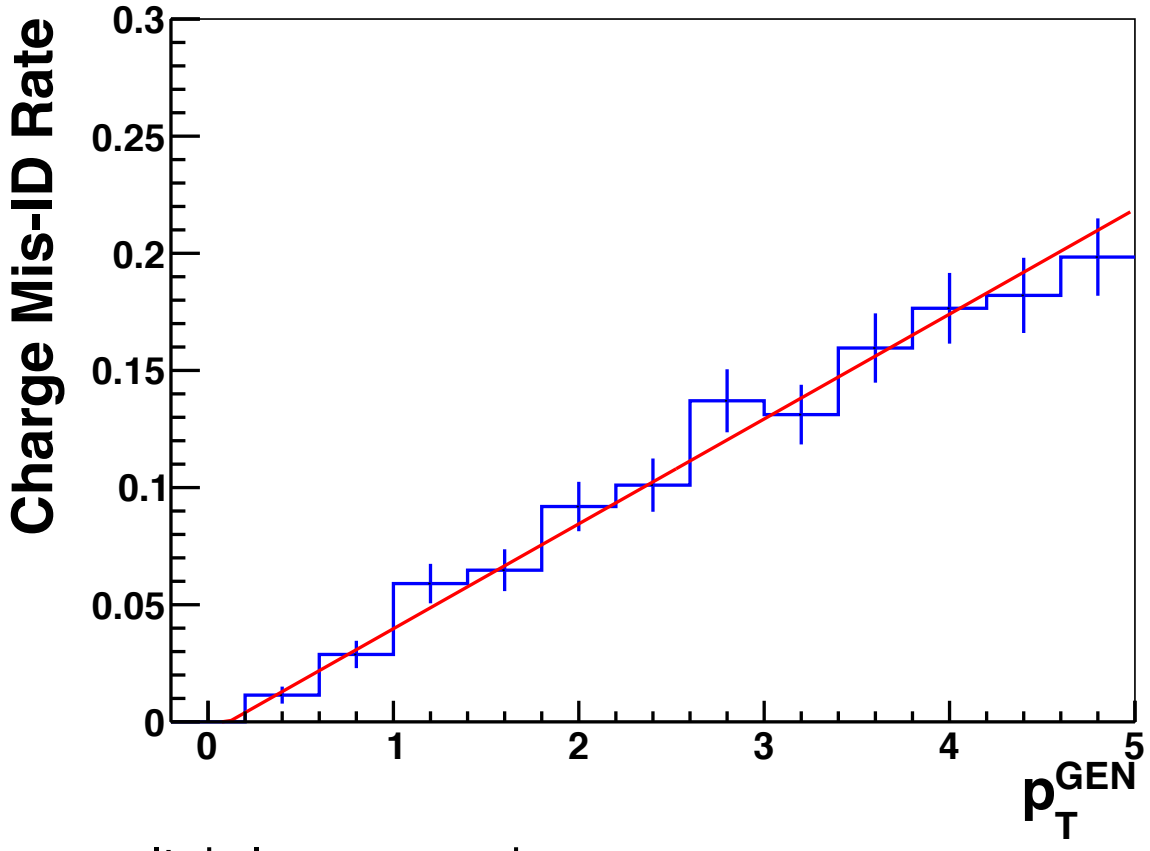




- Track finding efficiency is ≈ 95%
- Full material effects
- Real STAR B-field (still using old config optimized for const. B-field) - still some optimization possible, maybe recover 5% missing now
- Sharp turn-on curve for the efficiency vs. pT

STGC Only with Primary Vertex ($\sigma_{XY}^{PV} = 500\mu$m)





- Real (changing) magnetic field reduces performance slightly compared to const B-field
- Note: with sTGC only but without PV provides very poor constraint on track parameters.

Refitting with Si hits

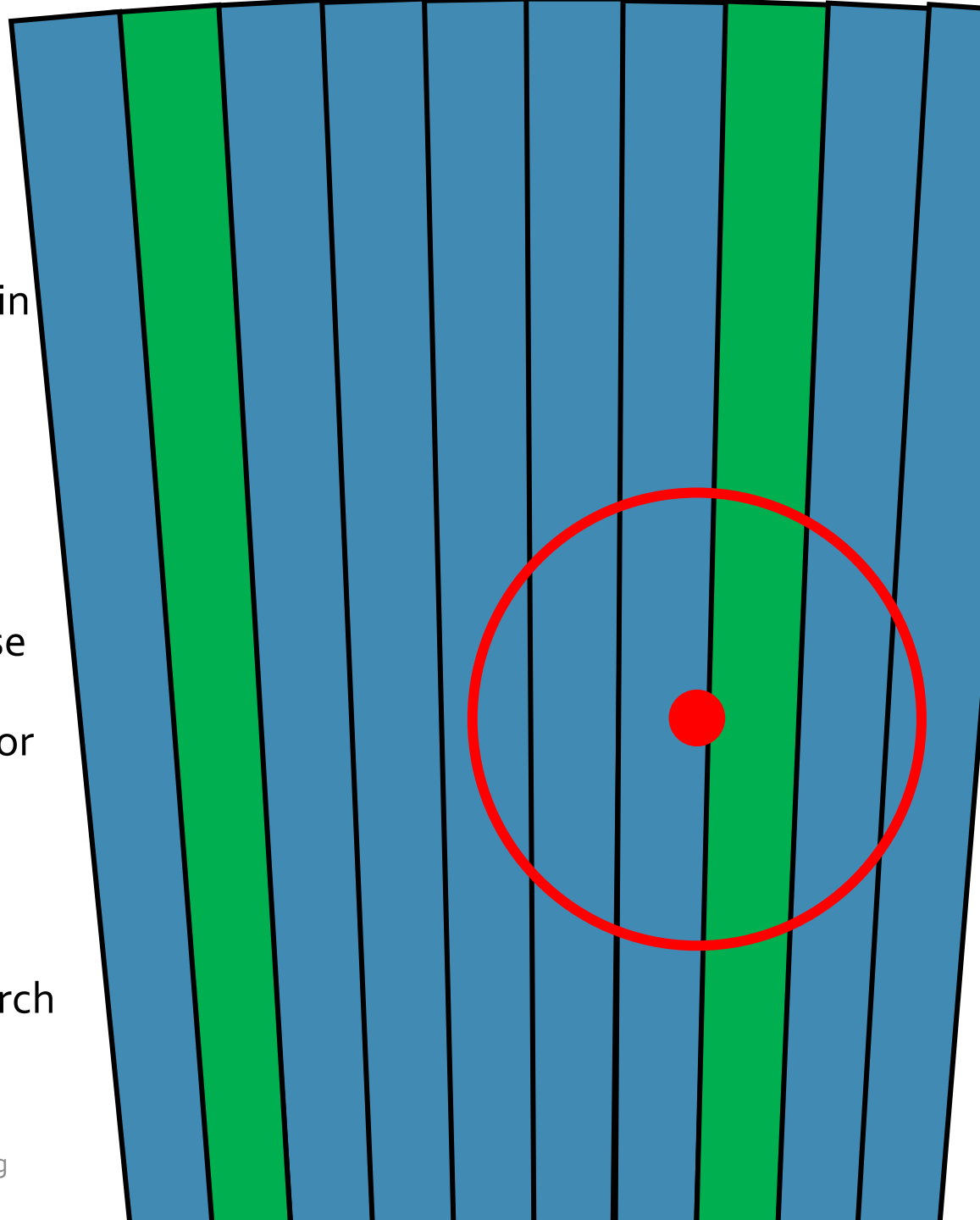
1. Track Finding (cellular automata)
2. Track fitting (PV + sTGC hits)
3. Project tracks to Si disks, search for and add Si hits within range
4. Refit tracks with added Si hits

Caveats while "full" version is being completed:

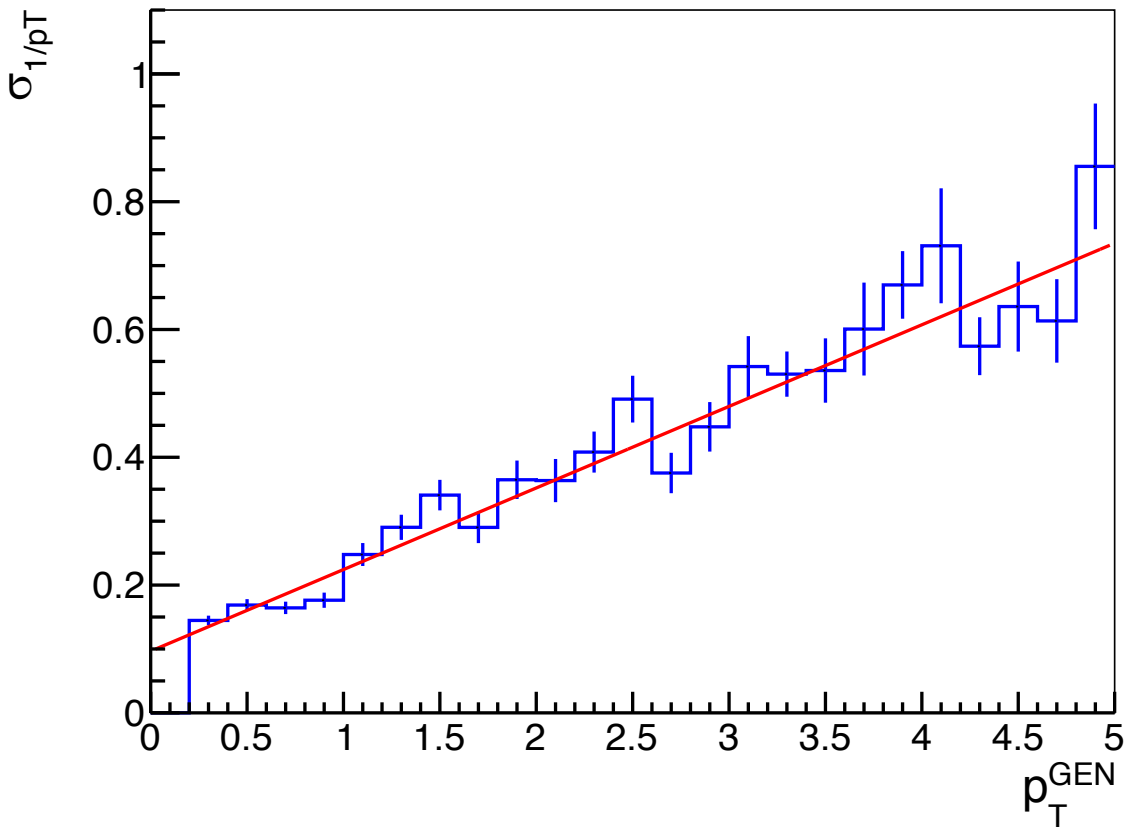
- Only refit tracks that find all 3 Si hits.
- No incremental improvement in projection (all 3 Si disks use the same track projection)
- Only searching inside 2-sigma window, for low occupancy or cases with no ambiguity, should search larger area to gain higher efficiency

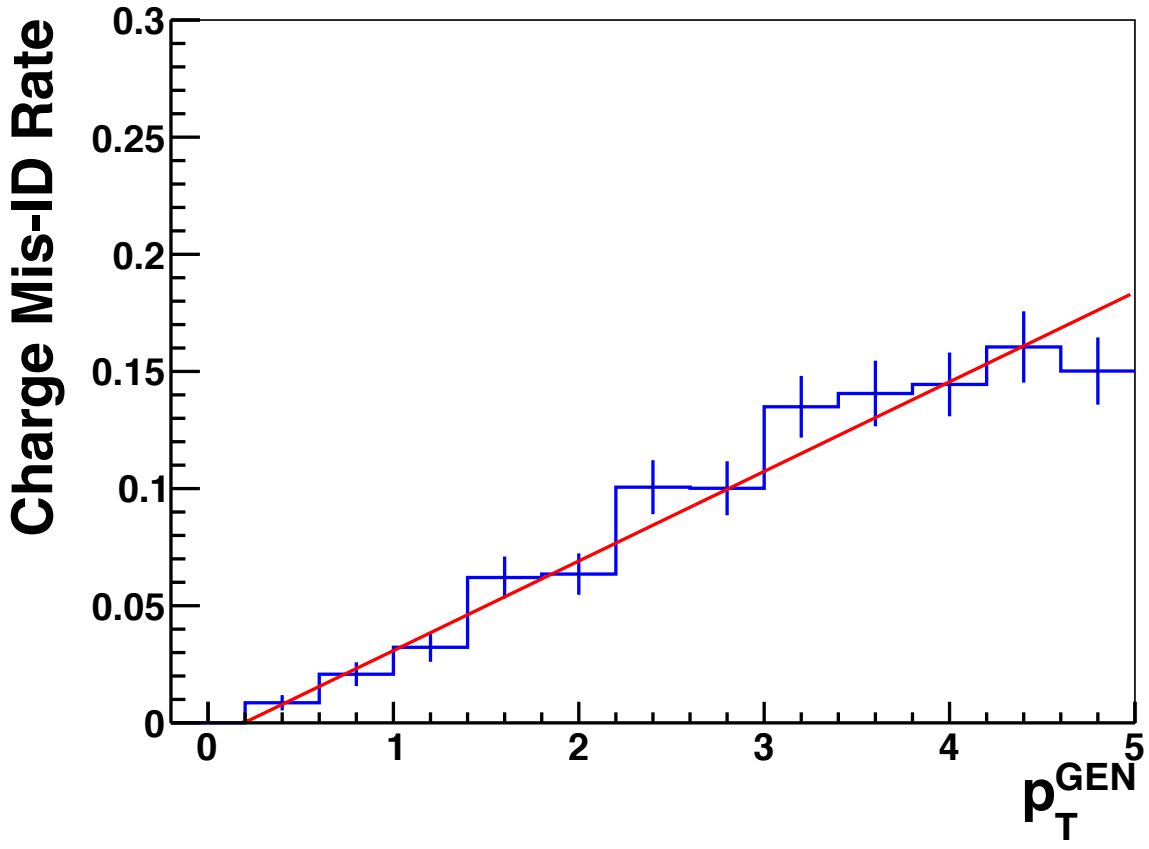
Real Limiting Factor

- Uncertainty on sTGC track projections = uncertainty in search area



STGC + PV($\sigma_{XY}^{PV} = 500\mu$m) + Refit with Si hits





- This uses Si hits with 3cm r-strips and 128*12 phi strips
- Techniques on previous slide – about 90% of tracks find correct three Si hits
- Improved momentum resolution and charge ID rate over sTGC only

Current and Future Tasks

- What I am working on
  - Finish integration (write tracks into MuDst, add back option to read from Fast Simulators, etc.)
  - Continue implementing procedures for finding Si hits – improve on current simple case and handle edge cases.
  - Work on Si refit – investigate charge change scenario
- What others can work on:
  - Study performance vs. parameter X (anything in config can be changed and studied)
  - Study performance in higher multiplicities (investigate benefit of multiple integrations with looser criteria)
  - Re-implement Fast Simulators (study sTGC ghost hits with new diagonal strips)
  - Study performance with other (or no) Primary Vertex assumptions

Summary

- Integration into StRoot on track + complete enough to use now
- Forward tracking software is ready for others to use for performance, design, and physics studies
- Performance with full material + real B-field is ~consistent with past presentations, only a little worse than with const. B-field, as expected
- Some changes/improvements still in progress