

# TMS studies Update

Jeffrey Kleykamp

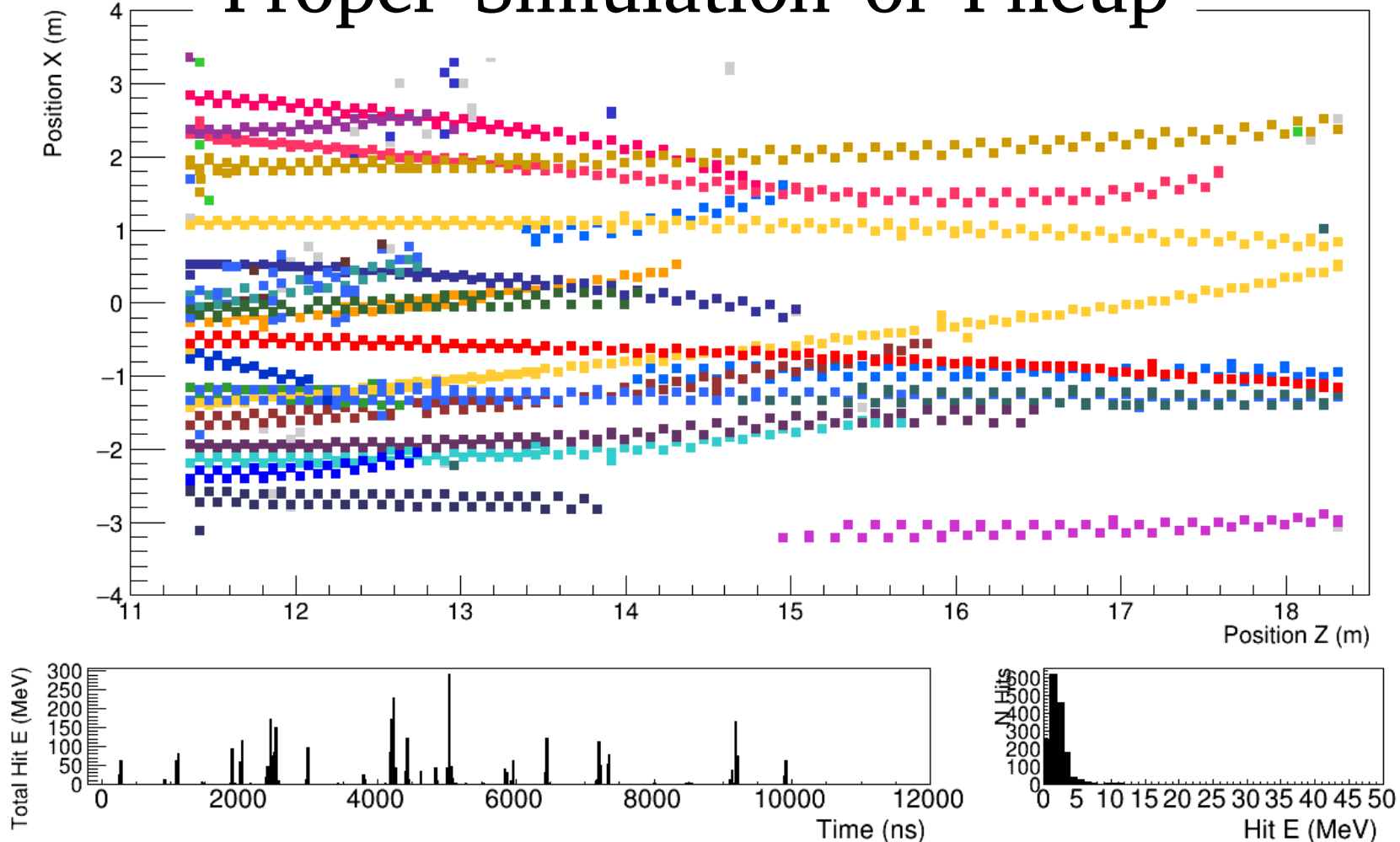
2024-08-09



# Updates

- Kalman Filter PR is on dune-tms
  - <https://github.com/DUNE/dune-tms/pull/122>
- Nersc-style pileup is now properly simulated
  - Lives in [kleykamp\\_pileup](#) branch
- Alex Booth is going to run TMS at nersc today
  - For ND reco/sim workshop
- Needed a release candidate
  - Suggested [kleykamp\\_pileup](#). See [tagged release](#)
    - Would've liked to incorporate kalman but this talk shows issues
    - Running again is really fast and can be done on fermigrid too

# Proper Simulation of Pileup



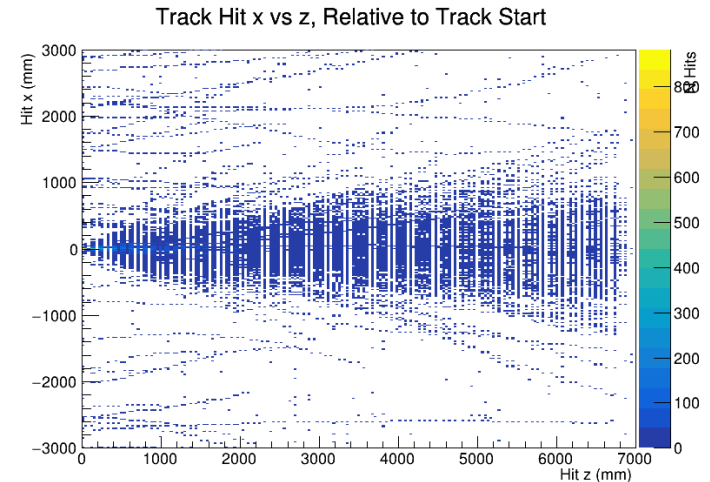
Each color represents a different timeslice  
Gray=slice 0 (ie all the hits not in a slice)

# Testing and Validation Framework

- Unfortunately many changes are not thoroughly tested when they are added
  - And some break downstream code bc of branch name changes
    - Please don't change branch names anymore!
  - So trying to validate files
- Added information on validating changes to wiki [here](#)
  - See also: [dune-tms issue tracker](#) and [current PRs](#)
- Ran tests of of several
  - Nercs-style, LAr-only, LAr + TMS
  - And we already found issues with infinite loops, etc. Most resolved on git
  - But there are remaining issues with reco that I'm showing today

# What I'm showing today

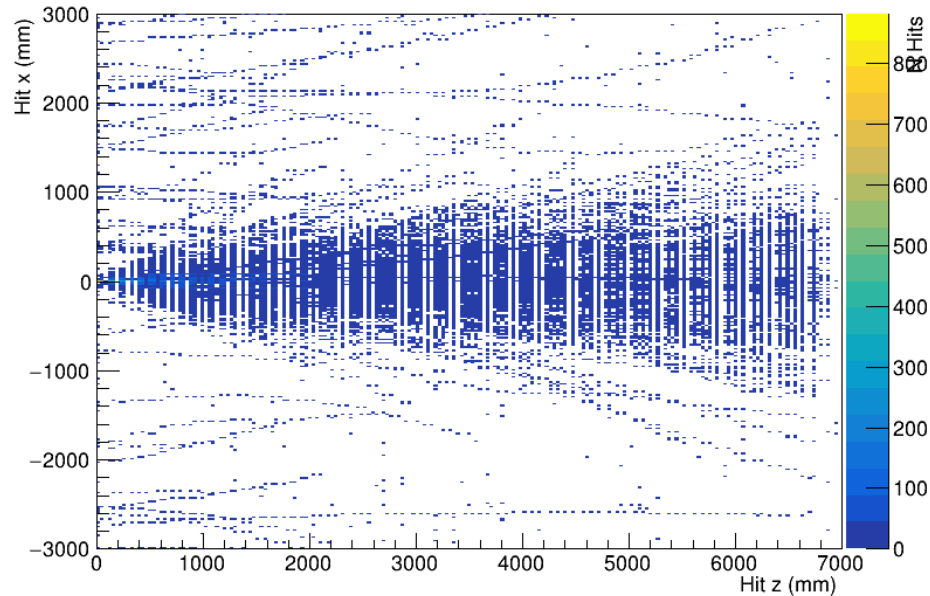
- Some truth comparisons are not so simple
- Sometimes true particle enters from outside, or interacts, so reco track may not be exactly where true particle is
- So showing reco compared to other reco
  - Specifically track hits relative to track starting location
- Comparing kleykamp\_pileup with kalman PR
  - pileup has older reco
  - kalman has kalman filter, but also slightly updated reco



# Track Hit Positions Relative to Track Start

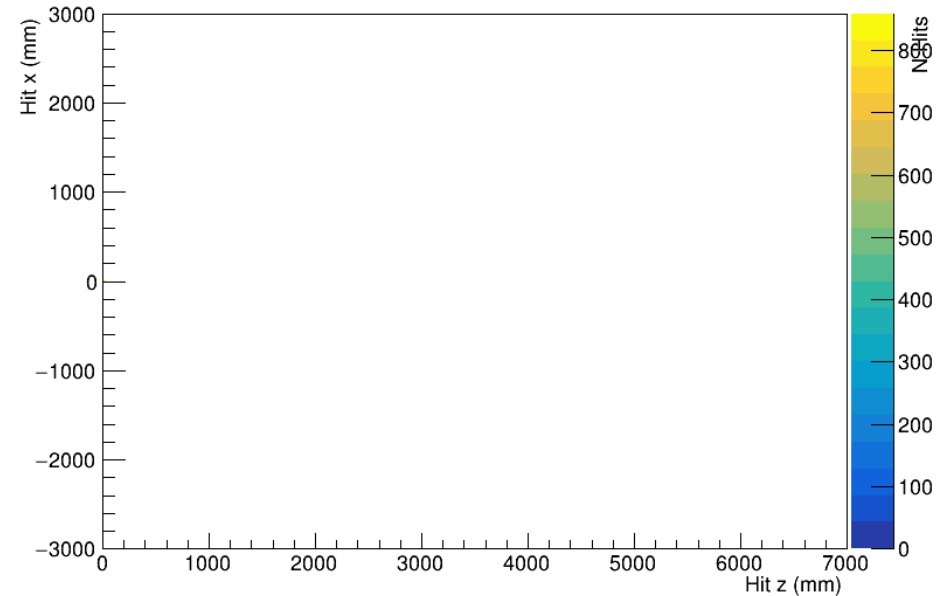
Pileup

Track Hit x vs z, Relative to Track Start



Kalman

Track Hit x vs z, Relative to Track Start

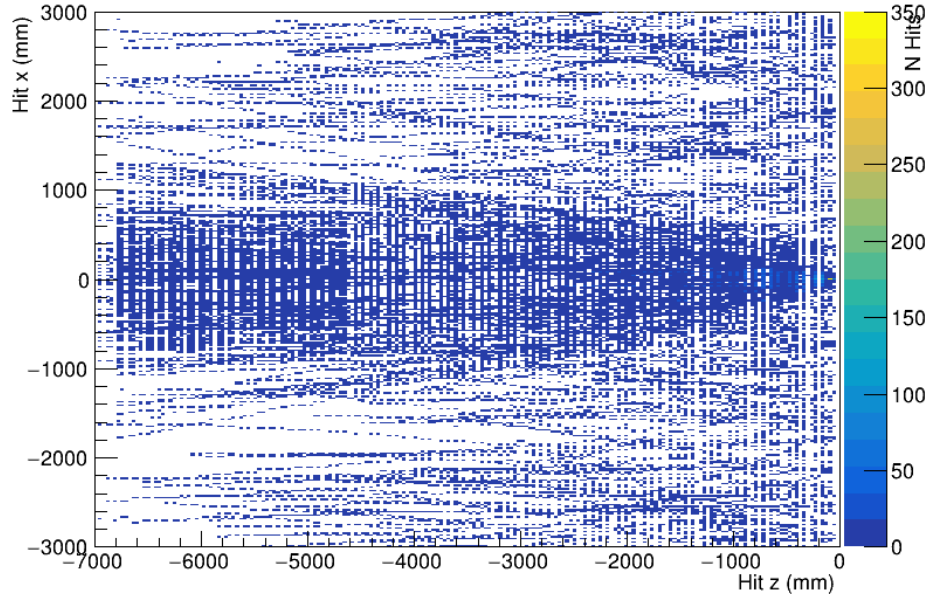


Plotting track hit  $x[i] - x[0]$  vs  $z[i] - x[0]$   
So we expect all hits to start at 0,0 and then move away

# Track Hit Positions Relative to Track End

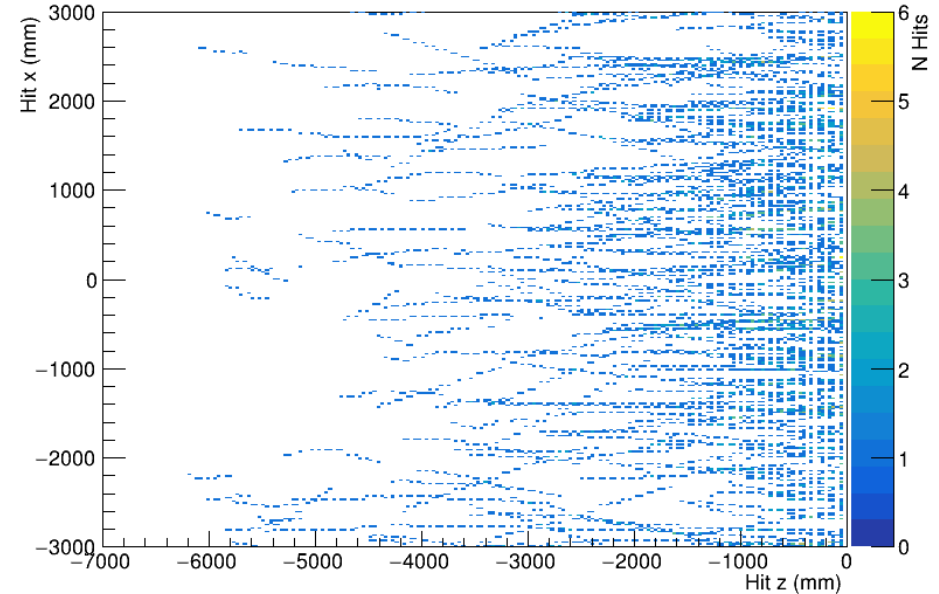
Pileup

Track Hit x vs z, Relative to Track End



Kalman

Track Hit x vs z, Relative to Track End

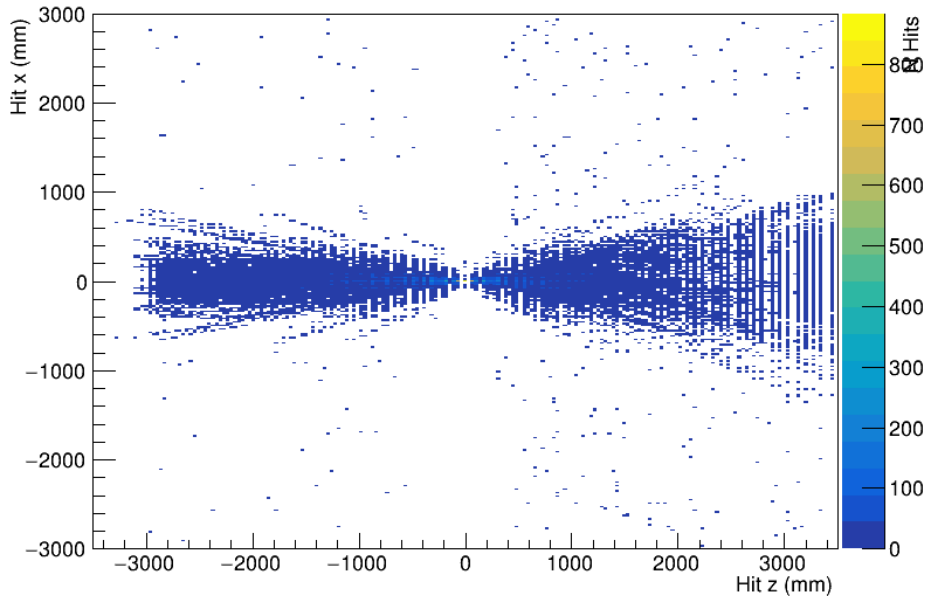


Plotting track hit  $x[i] - x[n-1]$  vs  $z[i] - z[n-1]$   
So we expect all hits to end at 0,0 and then move away

# Track Hit Positions Relative to Track End

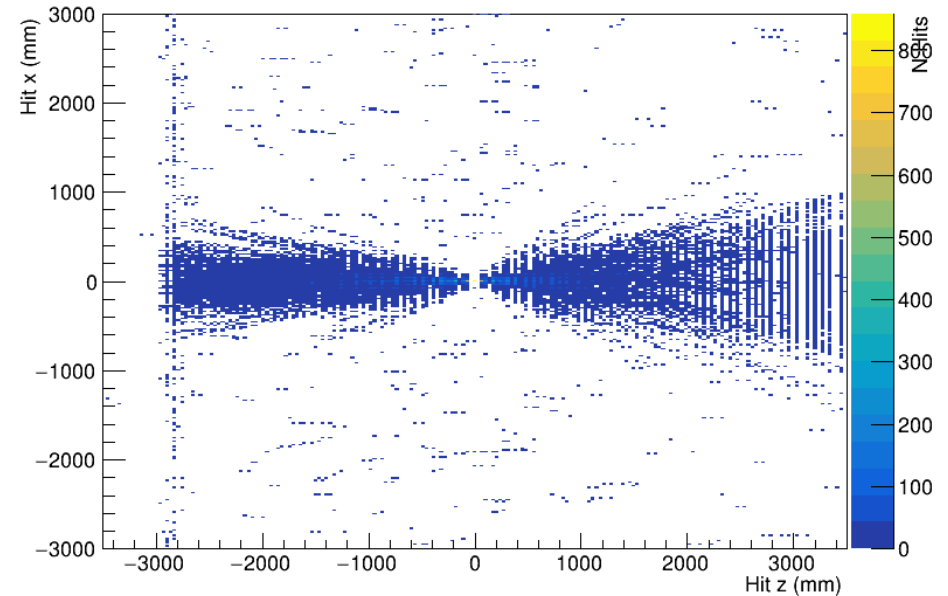
Pileup

Track Hit x vs z, Relative to Track Center



Kalman

Track Hit x vs z, Relative to Track Center



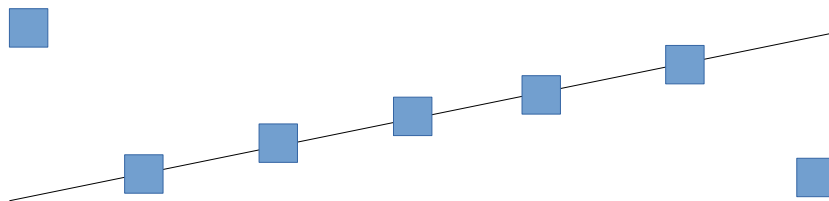
Plotting track hit  $x[i] - x[n//2]$  vs  $z[i] - x[n//2]$

So we expect all track centers to cross at 0,0 and then move away

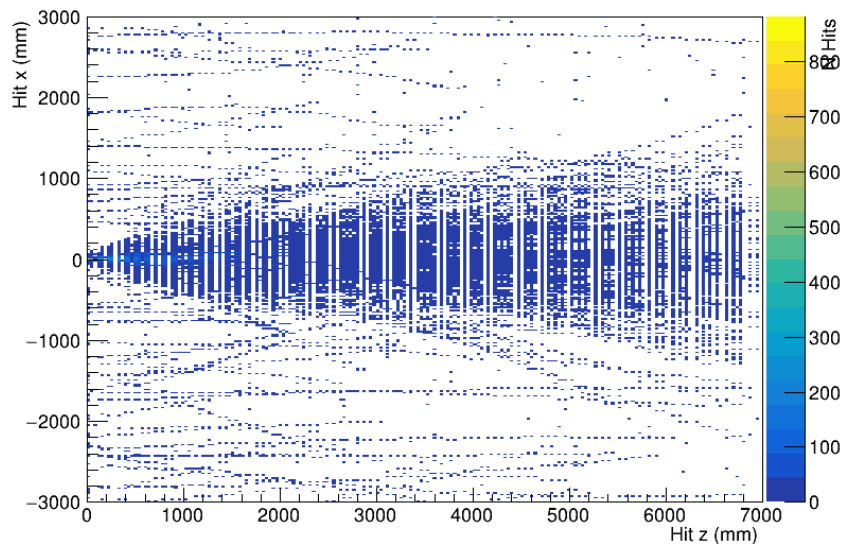


# What is Happening?

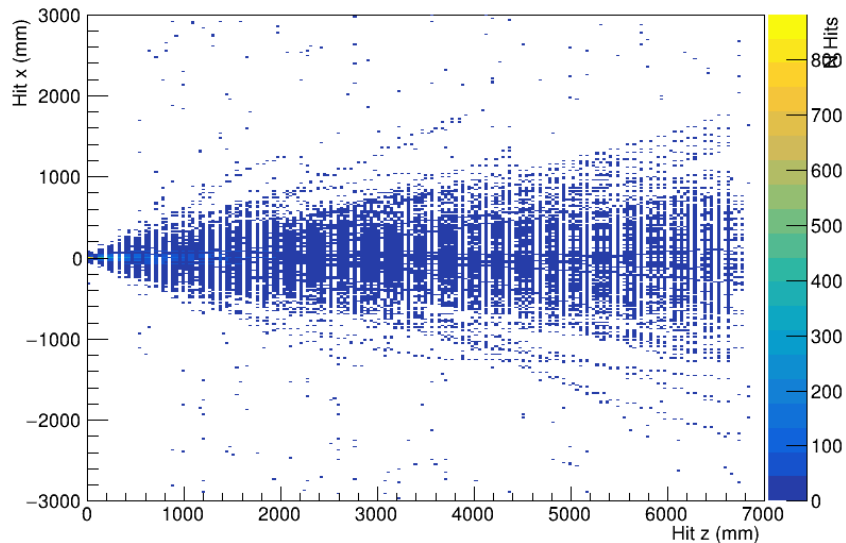
- Track is roughly linear, but then reco jumps at start/end
  - Likely hurts charge ID
  - Worse in kalman PR
- This issue mostly resolves itself after about 1-3 hits



Track Hit x vs z, Relative to Track Start + 1



Track Hit x vs z, Relative to Track Start + 2



# Conclusion

- Tracking reco issues
  - See [comment on PR](#) and [issue](#)
- We need champion to go in deep and validate
  - [Validation scripts](#) also useful starting point for c++ scripting (way faster than python!)
  - See [wiki](#) for a primer on the layout of the dune-tms code
- This is partly why I recommended the pileup version for nersc. Also it can handle the pileup

Godzilla Nebula

