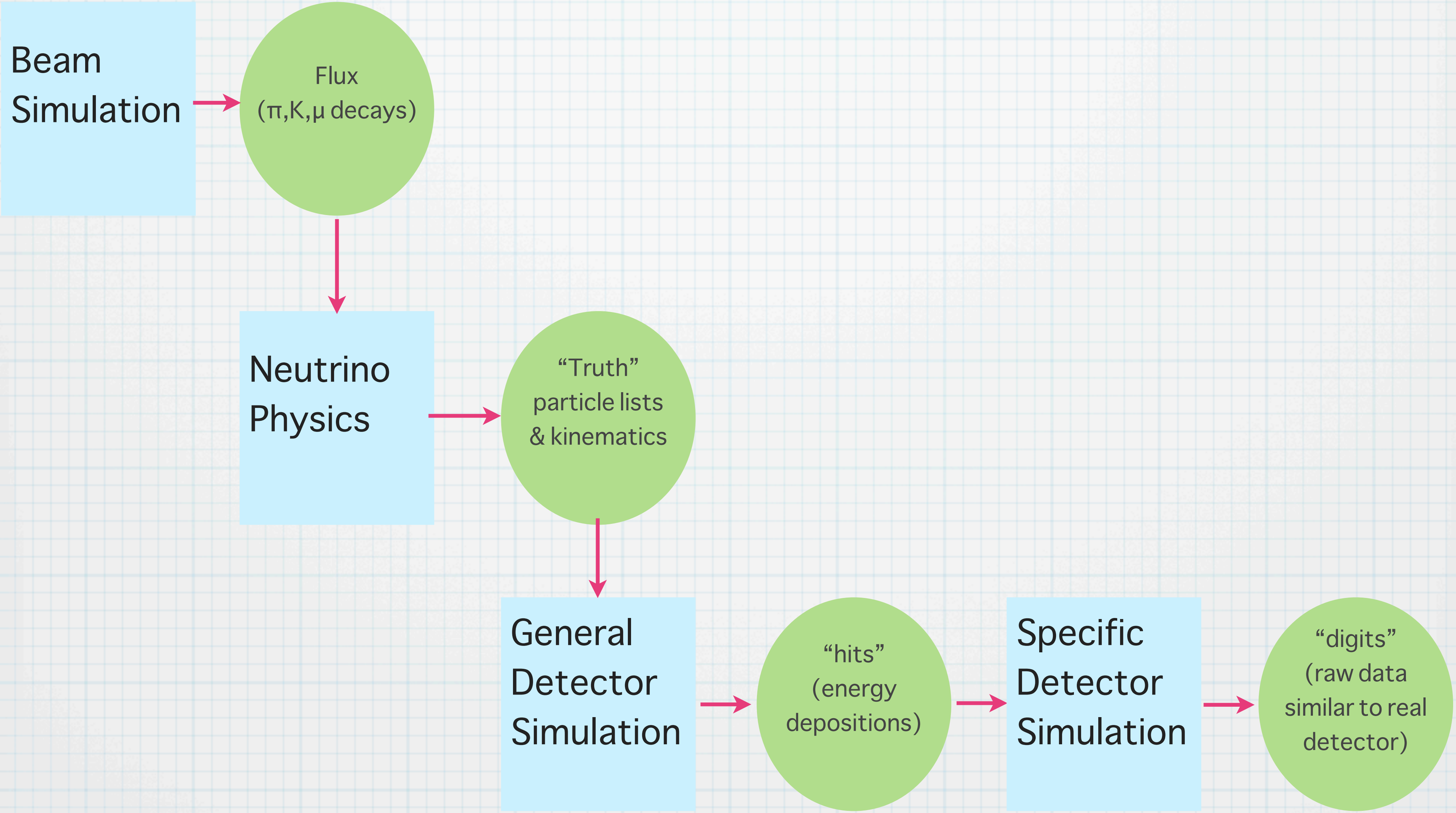
# “Event Generators” (and related elements) in the Neutrino Experiments

Robert Hatcher
Fermilab Computing Division

2013-10-31

# General Workflow & Products



Beam Simulation

Flux
(π,K,μ decays)

Neutrino Physics

"Truth" particle lists & kinematics

General Detector Simulation

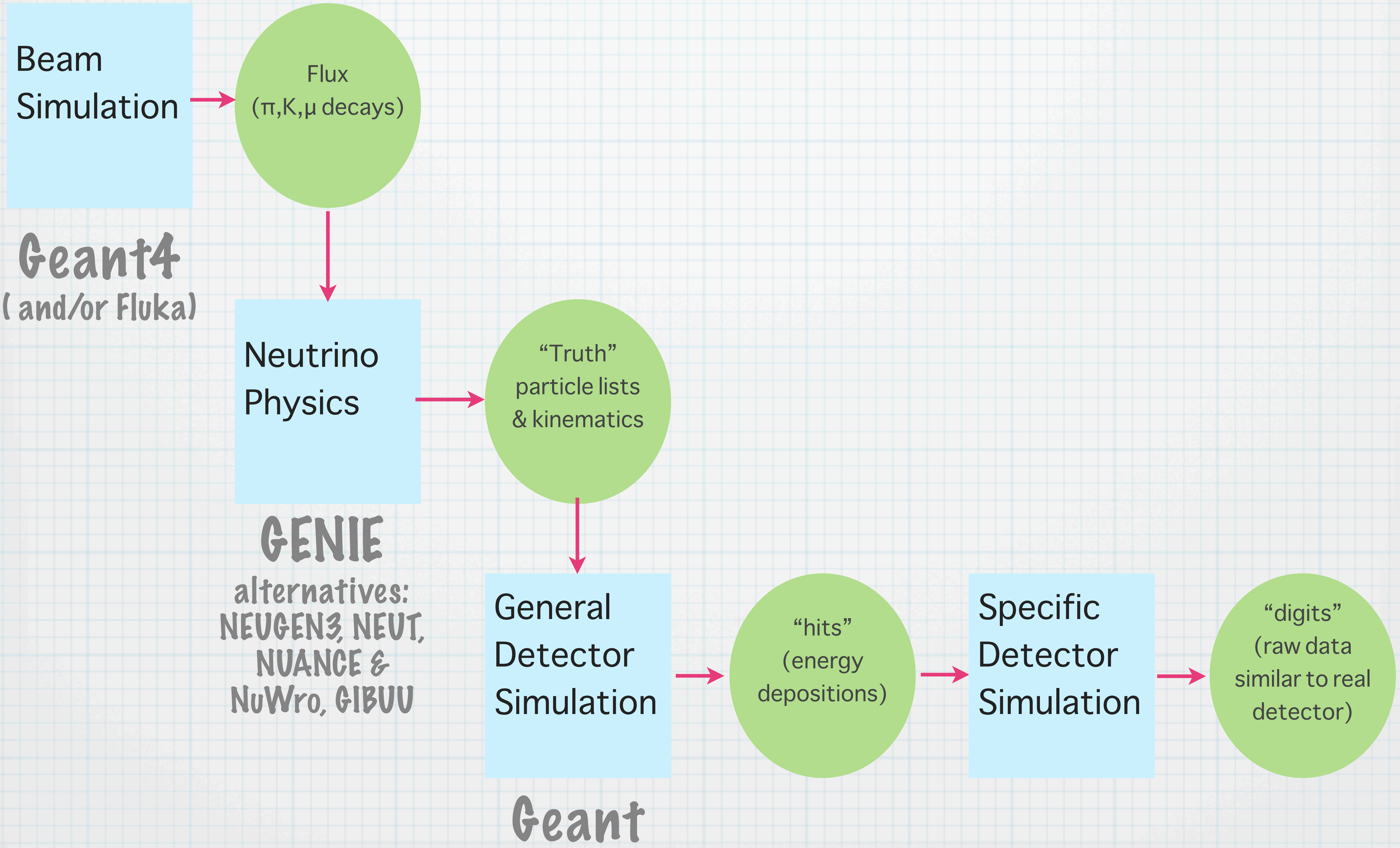
"hits" (energy depositions)

Specific Detector Simulation

"digits" (raw data similar to real detector)

Geant 4

# General Workflow & Products



Beam Simulation

Flux (π,K,μ decays)

Geant4 ( and/or Fluka)

Neutrino Physics

“Truth” particle lists & kinematics

GENIE

alternatives: NEUGEN3, NEUT, NUANCE & NuWro, GIBUU

General Detector Simulation

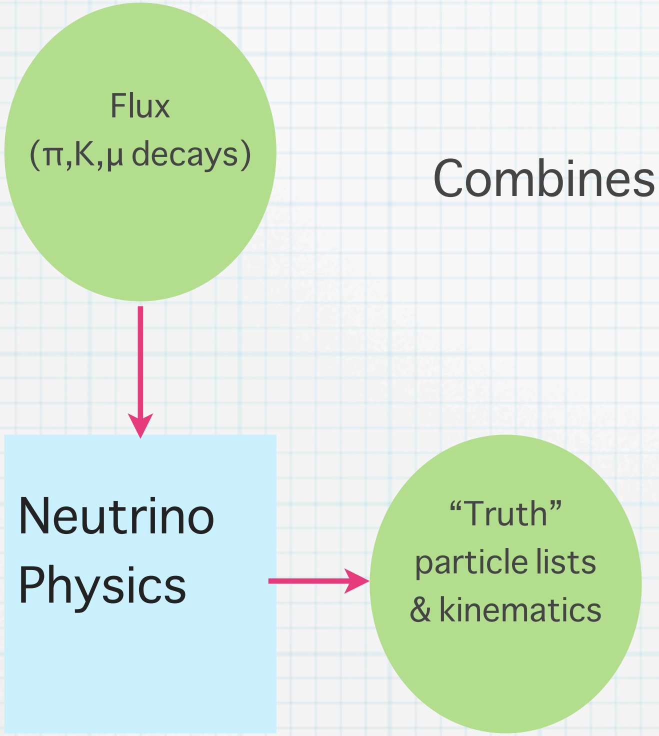
“hits” (energy depositions)

Specific Detector Simulation

“digits” (raw data similar to real detector)

Geant

# Neutrino Physics



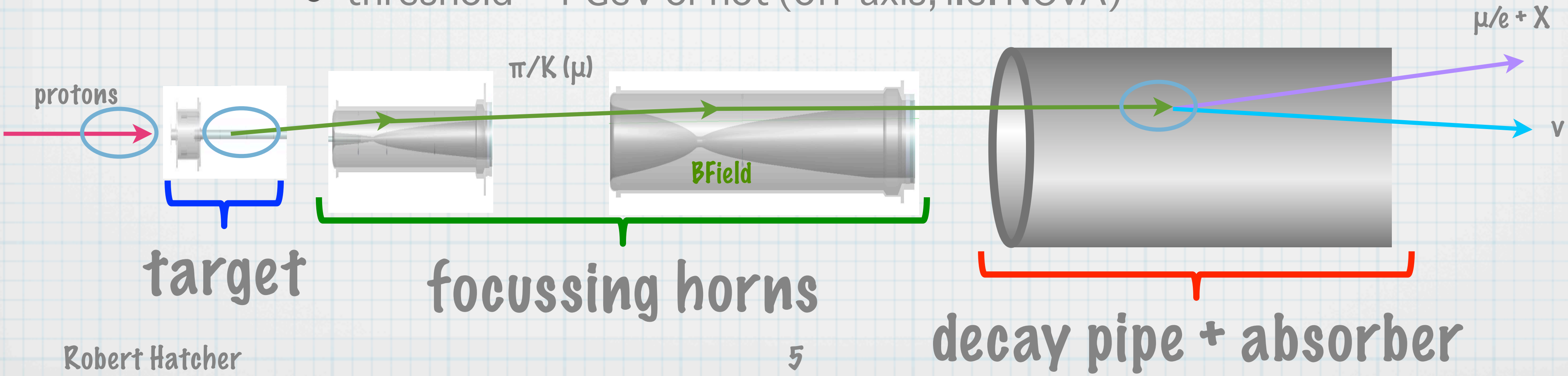
Combines flux, geometry w/ neutrino physics

Currently in use: GENIE and NEUGEN3 (MINOS is deprecating this in favor of GENIE)

GENIE can be used stand-alone (Minerva) or embedded in the offline framework (NOvA/ LArSoft, eventually MINOS)

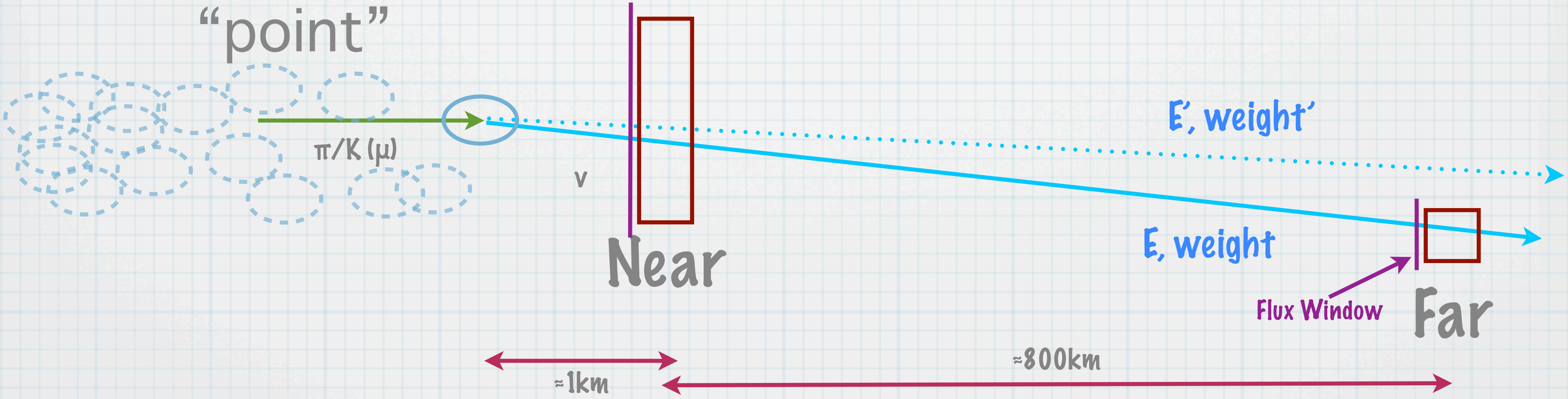
# Beam Simulation (Flugg)

- Is it “fl-ugh” or “floog”?
- Geant4 geometry (C++) + Fluka interactions (fortran)
  - G4 geometry is quite detailed (and good match to as-built)
  - fluka interactions everywhere (not just target)
- Record decay, initial secondary production and initial proton info
  - initial protons have some position and angular distribution
  - 2ndary production models are active areas of study
- Uses importance weights and thresholds
  - $w_{\pi} = \min(\max(30/p_{tot}, 1) * w_{parent}, 100)$
  - threshold = 1 GeV or not (off-axis, i.e. NOvA)



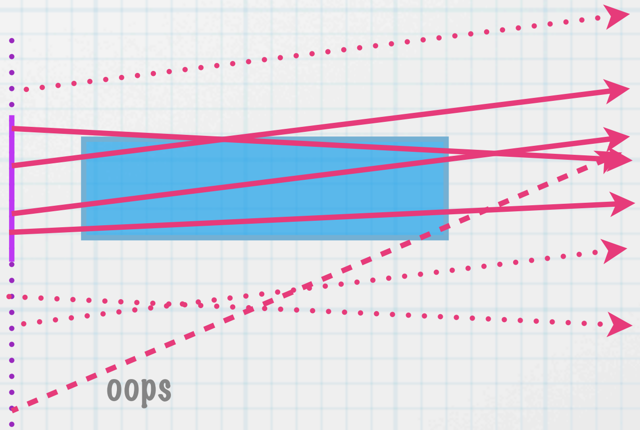
# Decay Reweighting

- The probability that a decay results in a neutrino ray that goes through any point depends on the relativistic boost at the decay point; the ν energy will also depend on position
- Near and Far detectors subtend a different angular size → they see different spectra
- Near also sees a “diffuse line source” vs. far “point”



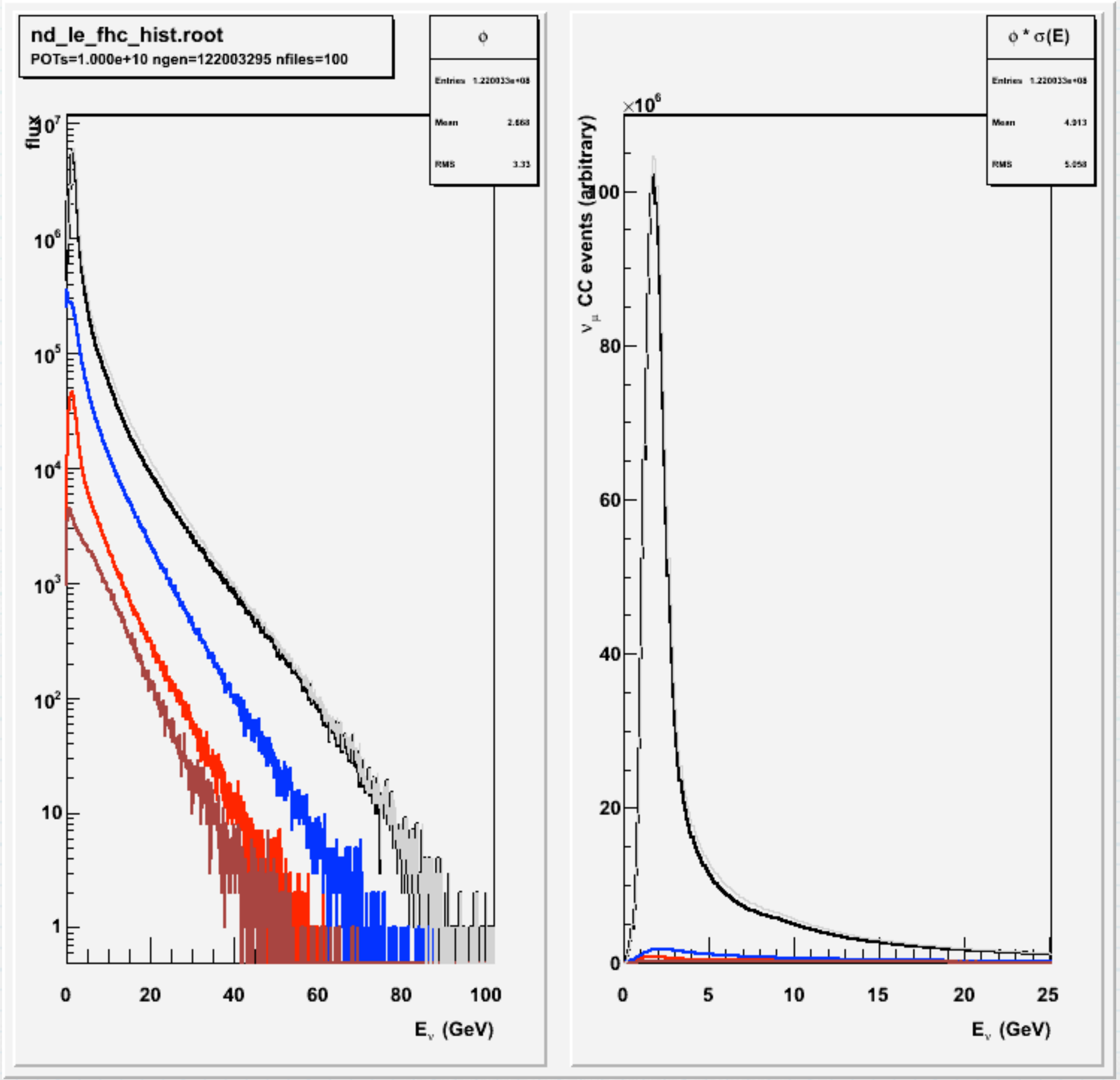
# Flux Window

- A fictitious parallelogram in space from whence neutrino rays emanate
- needs to be sized:
  - large enough that all (to best approximation) relevant rays that might run through the geometry pass through the window
  - small enough to exclude rays that aren't of interest



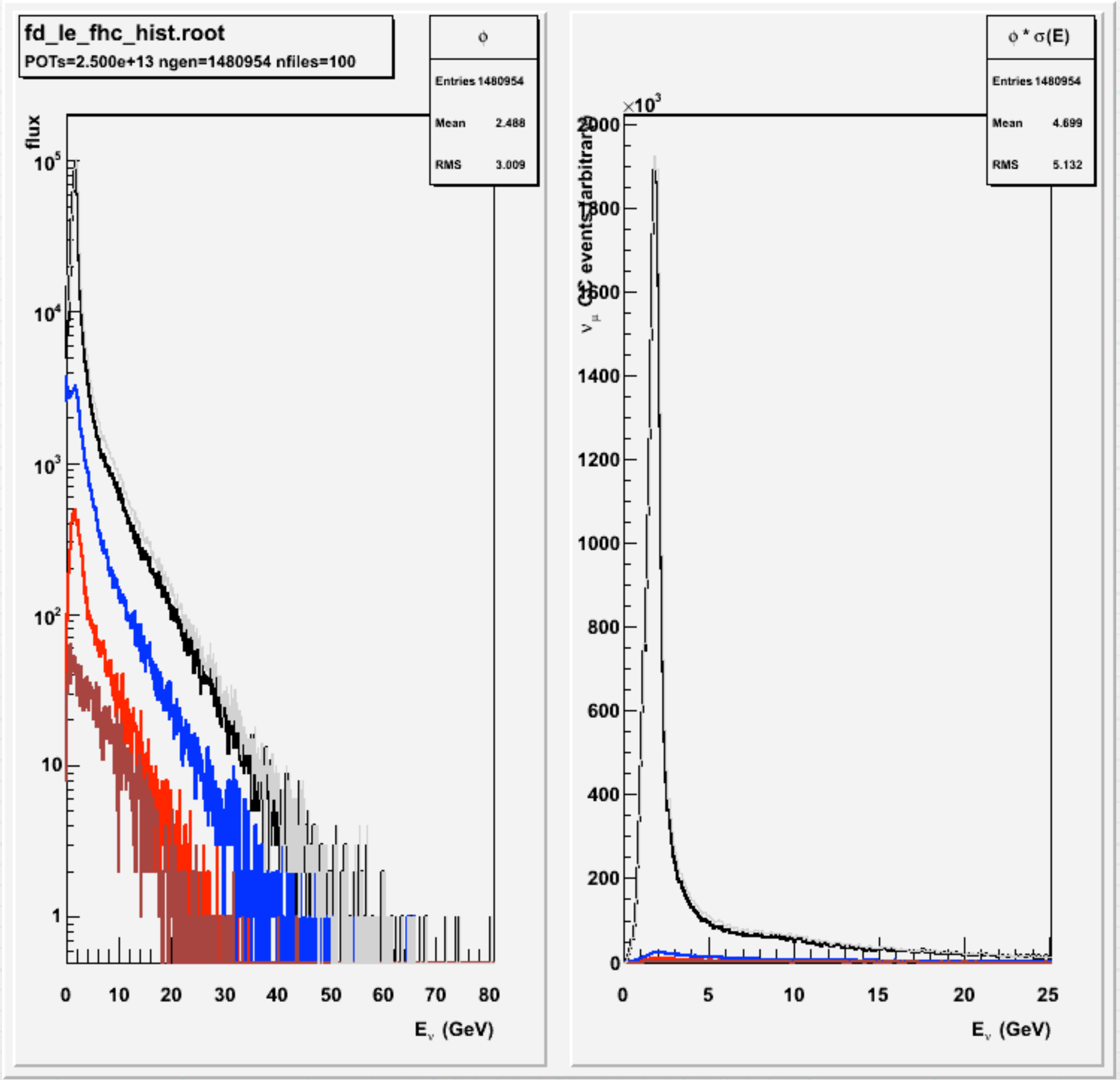
# ν Spectrum (Near)

Flux

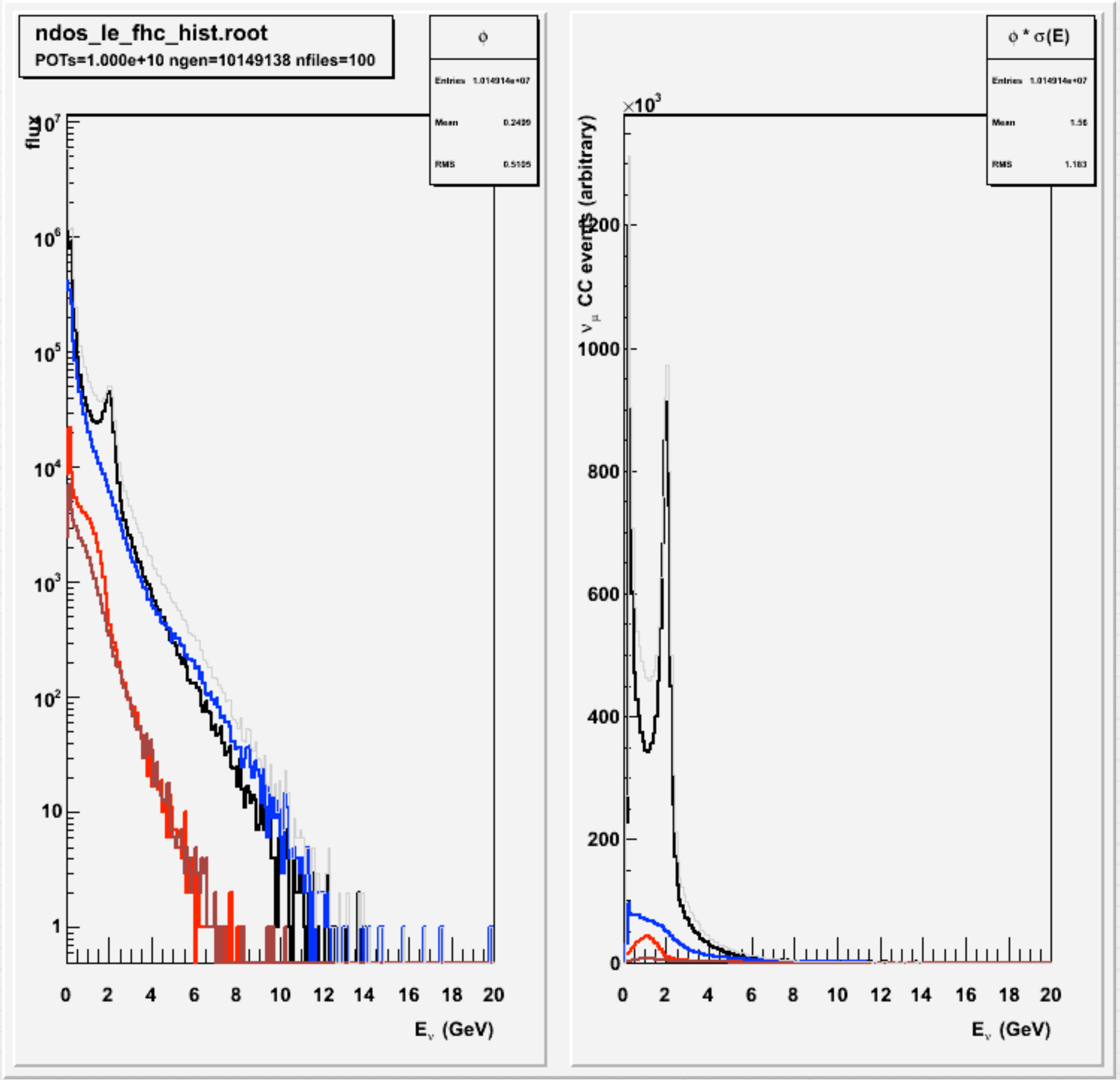


Flux × σ

# ν Spectrum (Far)



# ν Spectrum (NDOS)



Geant 4

# Beamline Simulation

| simulation step | software | work product |
|---|---|---|
| Beamline | flugg{ fluka<br>Geant4 } | flugg.root |
| ν Interactions<br>or other "generator" | | ux><br>article> |
| Energy Depositions | | |
| Light Propagation | | l> |
| DAQ Electronics | ReadoutSim | vector<RawDigit><br>+<br>vector<RawTrigger> |
| Reco/Analysis | UserCode | Clusters, Prongs, Tracks<br>+ histograms |

Flugg files are primarily (weighted) ntuple entries representing decays of hadrons (pions,kaons) and muons that result in a neutrino. For simple plotting they contain info for energies and weights of nu rays running through a fictitious detector center; for all but the far detector the energy and weight spectrum varies across the detector.

Geant 4

# Beamline Simulation

| simulation step | software | work product |
|---|---|---|
| Beamline | flugg{ fluka<br>Geant4 } | flugg.root |
| ν Interactions<br>or other "generator" | | |
| Energy Depositions | | |
| Light Propagation | | |
| DAQ Electronics | ReadoutSim | vector<RawTrigger> |
| Reco/Analysis | UserCode | Clusters, Prongs, Tracks<br>+ histograms |

The information in the file must be re-evaluated to sample the "flux window". Think of neutrino rays passing through the window and intersecting the geometry. The window placement and orientation relative to the beam system is specific for each detector. The window must be of sufficient size to encompass all rays that might intersect the detector.

Geant 4

# Beamline Simulation

| simulation step | software | work product |
|---|---|---|
| Beamline | flugg{ fluka<br>Geant4 } | flugg.root |
| ν Interactions<br>or other "generator" | | |
| Energy Depositions | | |
| Light Propagation | | |
| DAQ Electronics | | |
| Reco/Analysis | UserCode | Clusters, Prongs, Tracks<br>+ histograms |

Turn decays into (neutrino) rays.

Additional information is carried from the flux ntuple downstream in the chain to allow flux re-weighting of the original hadron production physics (cf. SKZP).

# Beamline Simulation

| simulation step | software | work product |
|---|---|---|
| Beamline | flugg { fluka<br>Geant4 } | flugg.root |
| | GENIE<br>FluxDriver | gsimple.root |
| | nova (ART framework) | art::Event (art::Run) |
| ν Interactions<br>or other “generator” | GENIE<br>or CRY/SingleParticle | + v<MCTruth>,v< MCFlux><br>(RunData, POTSum)<br>MCTruth hold a vector<MCParticle> |
| Energy Depositions | Geant4 | + vector<FLSHitList><br>vector<ParticleList> |
| Light Propagation | Simple<br>Photon ^ Transport | + vector<PhotonSignal> |
| DAQ Electronics | ReadoutSim | + vector<RawDigit><br>vector<RawTrigger> |
| Reco/Analysis | UserCode | Clusters, Prongs, Tracks<br>+ histograms |

# The “Flux” File Problem

- Each beamline simulation used a different (incompatible) variant of ntuple
    - g3numi, flugg, g4numi, g4numi (MINERvA variant), g4lbne
- Solution: design new common variant and an associated GENIE flux driver (dk2nu)
    - carry 2ndary production info for reweighting
        - expt will attempt to use HARP, SPY, MIPP, NA49, NA61/SHINE to adjust production models
        - none quite matches $E$ , $p_t$ , $p_z$ of Main Injector/NuMI

# GENIE Flux Drivers

- GNuMIFlux & Dk2NuFlux
  - read an entry from “flux” (decay) file
  - pick random point on flux window (x,y,z)
  - calculate x-y weight, energy, p4
  - accept/reject based on weights ( $wgt_{x\text{-}y}$ * $wgt_{importance}$ )
  - (possibly) push backwards along ray to (x’,y’,z0)
  - GENIE’s GeometryDriver uses ray ⇒ create PathSegmentList to determine potential targets (position and makeup)
  - cycling back to same entry won’t give same ray
    - different window point ⇒ different weight, energy, trajectory
- Problem: computationally expensive
  - repeatedly generating new rays w/ low probability of interaction
  - off-axis it becomes inefficient

# GSimpleNtpFlux

- GENIE GSimpleNtpFlux
  - filled from using GNuMIFlux or GDk2NuFlux
    - just an intermediate step
  - simple ntuple format of flavor, position, direction, weight
    - provision for carrying extra info to allow hadron reweighting
  - some file level meta data (window position, total protons,...)

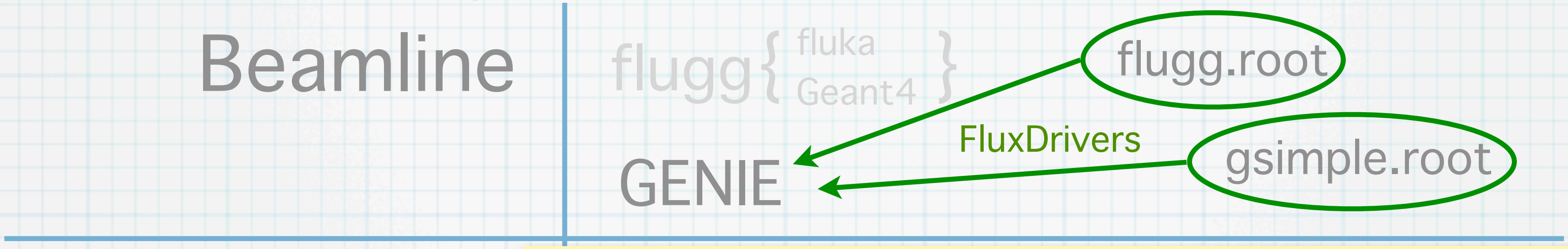
- Outstanding Issue:
  - still need infrastructure to map from GSimpleNtpFlux info back to original files for 2ndary ancestor chain when necessary
    - most likely not at generation time, but analysis
  - (also existing files are ambiguous due to failure of beam simulation generators filling important distinguishing info)

# Flux Validation

- Hard ... nothing directly measures flux
- Ongoing effort to compare secondaries w/ experiment as proxy for flux
- Comparison of alternatives
  - flugg vs. geant4 vs. geant3
- Possibilities for future test suites:
  - validation of coordinate transform + boost
  - ...

# Beamline Simulation

simulation step | software | work product



Beamline

flugg { fluka Geant4 }

flugg.root

gsimple.root

FluxDrivers

GENIE

GENIE's GNuMIFlux class interprets flugg (and similar) files, performing the sampling over a window for a given detector location relative to the beam. This can be computationally expensive, so it makes sense to do it separately and save the results.

ν Interactions

or other "generator"

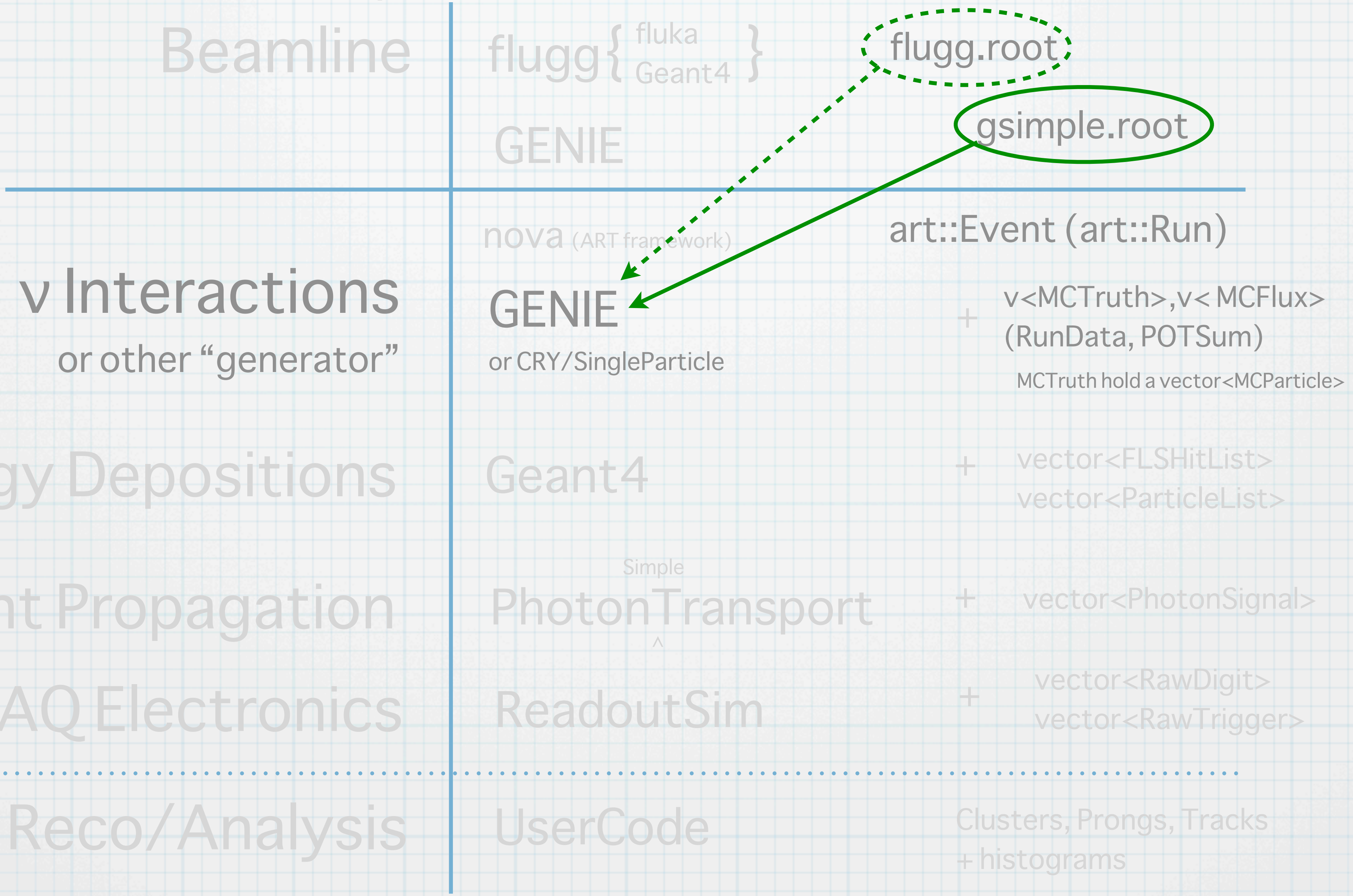
Energy Depositions

Light Propagation

DAQ Electronics

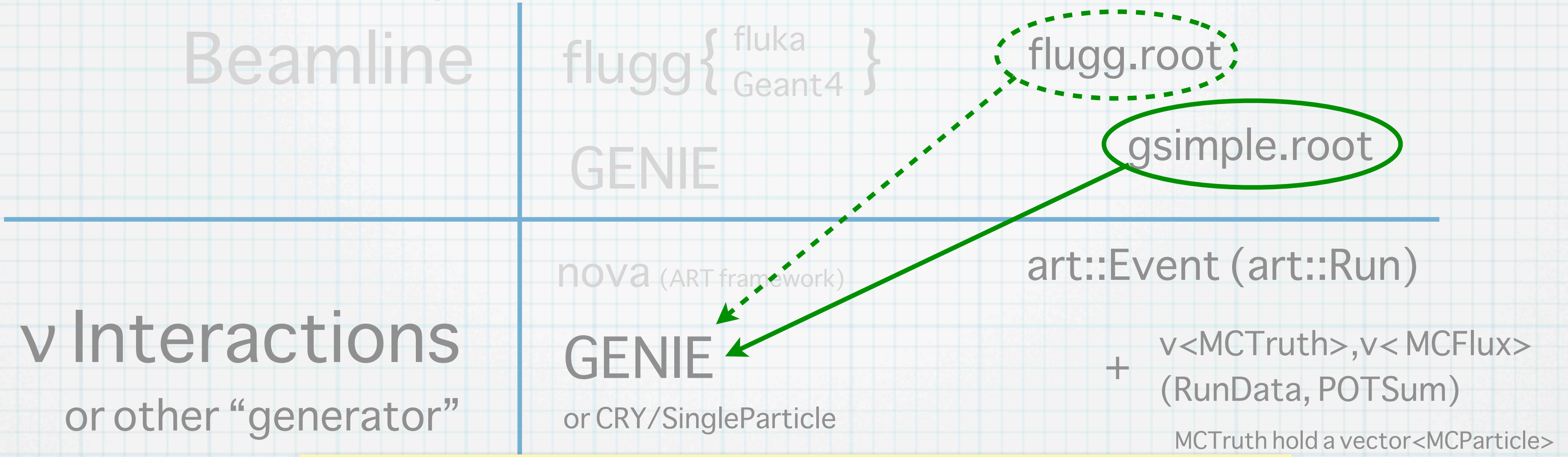
Reco/Analysis | UserCode | Clusters, Prongs, Tracks + histograms

# Neutrino Physics



| simulation step | software | work product |
|---|---|---|
| Beamline | flugg{ fluka<br>Geant4 } | flugg.root |
| | GENIE | gsimple.root |
| | nova (ART framework) | art::Event (art::Run) |
| ν Interactions<br>or other "generator" | GENIE<br>or CRY/SingleParticle | + v<MCTruth>,v< MCFlux><br>(RunData, POTSum)<br>MCTruth hold a vector<MCParticle> |
| Energy Depositions | Geant4 | + vector<FLSHitList><br>vector<ParticleList> |
| Light Propagation | Simple<br>PhotonTransport<br>^ | + vector<PhotonSignal> |
| DAQ Electronics | ReadoutSim | + vector<RawDigit><br>vector<RawTrigger> |
| Reco/Analysis | UserCode | Clusters, Prongs, Tracks<br>+ histograms |

# Neutrino Physics

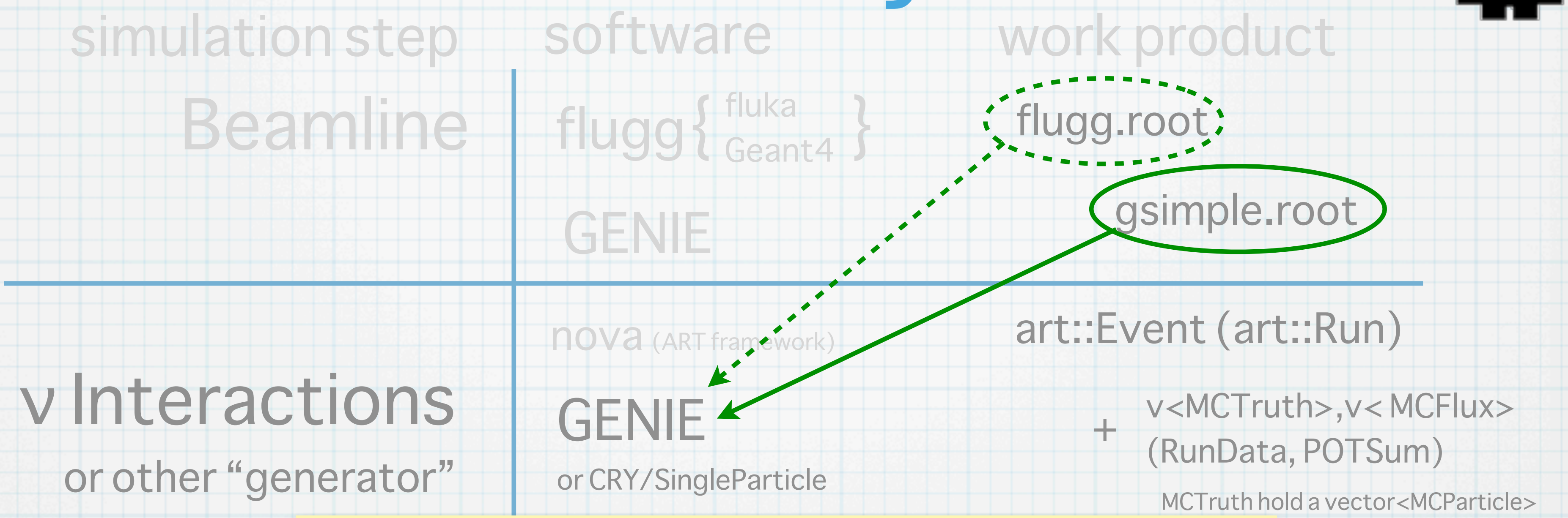


| simulation step | software | work product |
|---|---|---|
| Beamline | flugg { fluka<br>Geant4 }<br>GENIE | flugg.root<br>gsimple.root |
| ν Interactions<br>or other "generator" | nova (ART framework)<br>GENIE<br>or CRY/SingleParticle | art::Event (art::Run)<br>+ v<MCTruth>,v< MCFlux><br>(RunData, POTSum)<br>MCTruth hold a vector<MCParticle> |
| Energy Deposi | | FLSHitList><br>ParticleList> |
| Light Propaga | | <PhotonSignal> |
| DAQ Electr | | <RawDigit><br><RawTrigger> |
| Reco/An | | ngs, Tracks |

**GENIE brings together an external flux, a user supplied ROOT geometry and its knowledge about neutrino cross sections to determine if individual neutrino rays pulled from the flux interact in the geometry. If they do, then it chooses a vertex location and the kinematics of the interaction (QE/resonance/DIS/etc,x,y...).**

# Neutrino Physics

Geant 4



| simulation step | software | work product |
|---|---|---|
| Beamline | flugg { fluka<br>Geant4 } | flugg.root |
| | GENIE | gsimple.root |
| | nova (ART framework) | art::Event (art::Run) |
| ν Interactions<br>or other “generator” | GENIE<br>or CRY/SingleParticle | + v<MCTruth>,v< MCFlux><br>(RunData, POTSum)<br>MCTruth hold a vector<MCParticle> |

Energy Deposi

SHitList>

ticleList>

Light Propaga

otonSignal>

DAQ Electr

awDigit>

awTrigger>

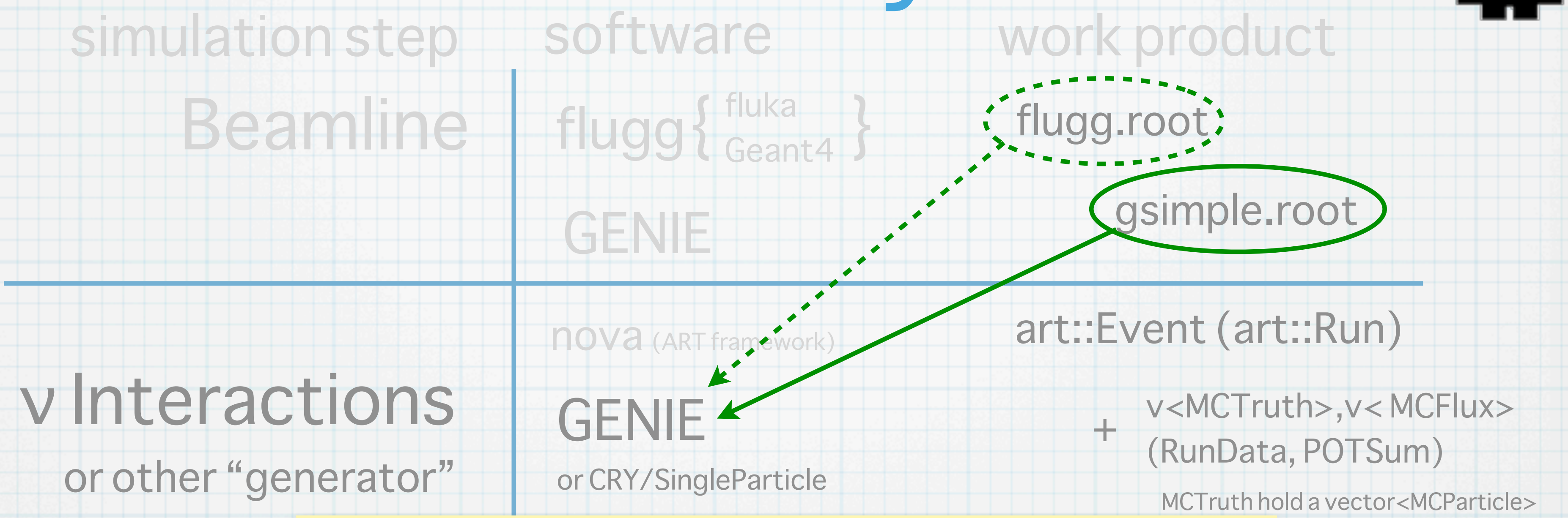
Reco/An

, Tracks

Given an initial state in the nuclear environment, GENIE also propagates particles out of that nucleus (undergoing internuclear scattering and absorption). This results in a list of target / probe (nu) particles, intermediate states and final state particles.

Robert Hatcher

# Neutrino Physics



Energy Deposi

SHitList>

leList>

Light Propaga

nSignal>

DAQ Electr

Digit>

Trigger>

Reco/An

racks

GENIEHelper (NOvA,LAr) handles doing overlays (multiple interactions w/ offset times) and filling the ART record. It also copies the flux info to the ART record.

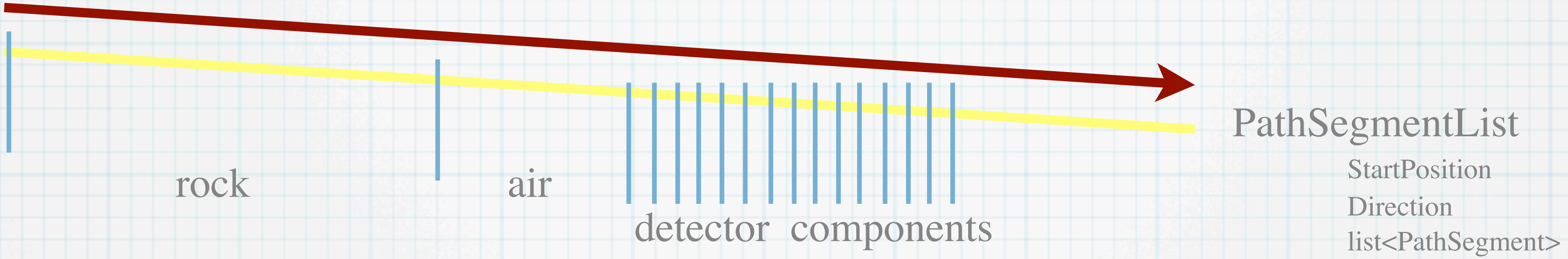
GENIE can also “mix” neutrino flavors from the flux generator.

Geant 4

# GENIE Geometry Analyzer

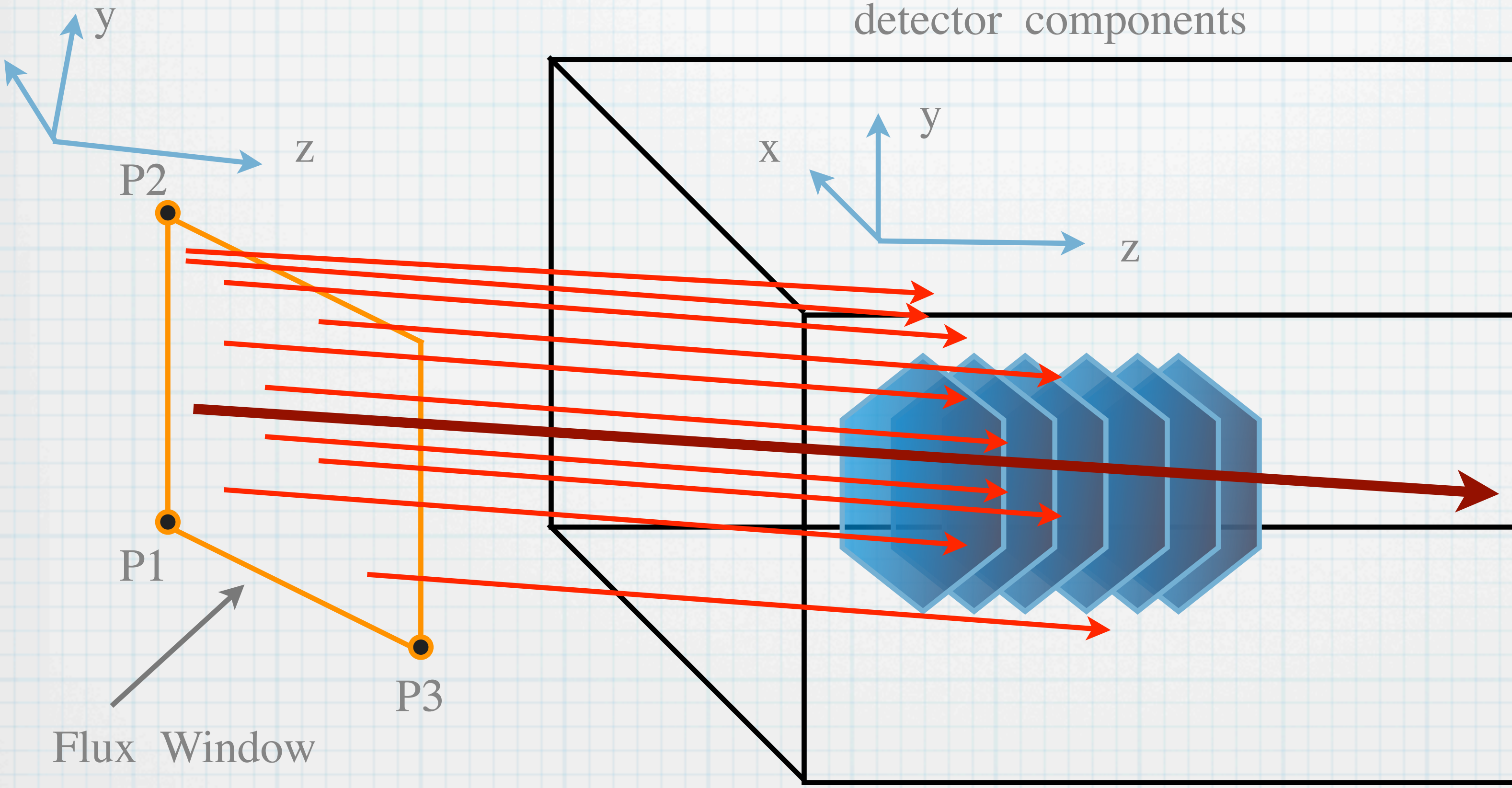
- Uses ROOT geometry
  - that’s what experiments generally use in their offline reconstruction (G4 linkage is looked on askance)
  - has the necessary straight-line “transportation”
    - all that is needed is volume enter/exit point along ν ray
- NOvA: some discussion about validating
  - cross check ROOT vs. G4 vs. simple by-hand cases
  - partly to validate GDML → ROOT / G4 representation

# ν Rays and Geometry



PathSegmentList

- StartPosition
- Direction
- list<PathSegment>



# PathSegments

- components: dist from ray start; step length; volume; medium; material; global position for entrance and exit
- Used to determine probability of interaction in segments and then choice of vertex position

# GENIE x-sections

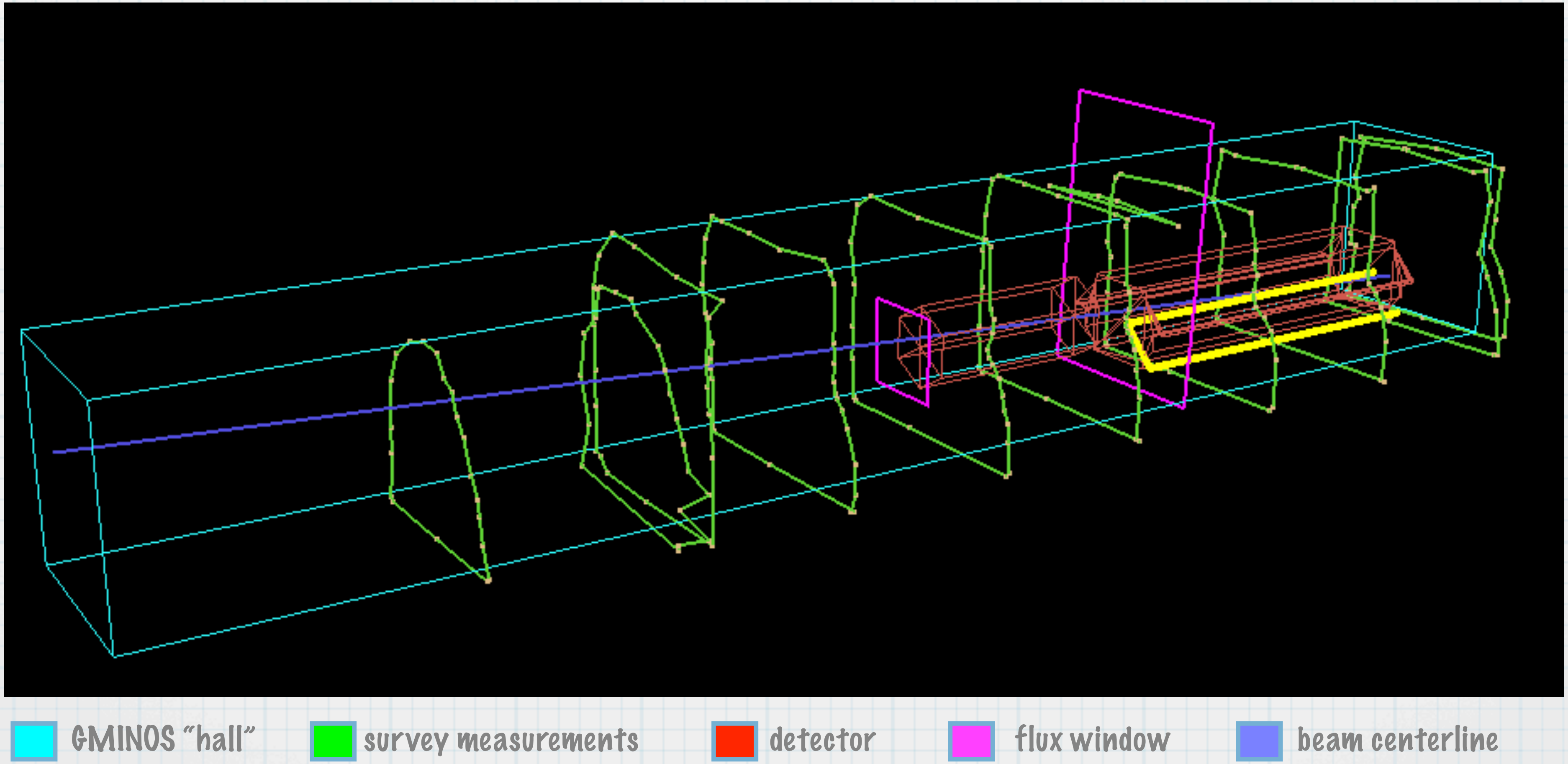
- GENIE cross-sections: ~440 MB (15574 splines)
  - 271 for proton, 301 for neutron, 577 for each of 26 nuclei
    - currently need only ~14 of 26: $C^{12}$, $N^{14}$, $O^{16}$, $Na^{23}$, $Al^{27}$, $Si^{28}$, $S^{32}$, $Cl^{35}$, $Ar^{40}$, $K^{39}$, $Ca^{40}$, $Ti^{48}$, $Fe^{56}$, $Ba^{137}$
    - not: $Mg^{24}$, $P^{31}$, $V^{51}$, $Mn^{55}$, $Fe^{54}$, $Fe^{57}$, $Fe^{58}$, $Ni^{59}$, $Cu^{64}$, $Sn^{119}$, $Pb^{207}$
  - 500 knots, [0.01:400] GeV spaced logarithmically
    - 12% of points > 120 needed for NuMI - but for cosmic neutrinos
      - NOvA ND/FD MN flux has entries up to 104 GeV
  - Files currently stored as XML text
    - loading from ROOT file (vs XML) speeds up loading, but doesn't appear to significantly change peak memory usage
- Study: # knots in spline vs. accuracy

# X-Section Validation

- GENIE collaboration has collected together extensive set of measurements
  - can compare to other neutrino and electron scattering experiments
  - total σ related to $F_1$ and $xF_2$, ultimately to parton distribution functions
  - some exclusive processes are theoretically better understood and thus can be a standard candle with single adjustable parameter ($M_A$ in QE)
- MINOS/NOvA use 2 detectors to minimize effects of flux and x-section uncertainty

# Detector Environment

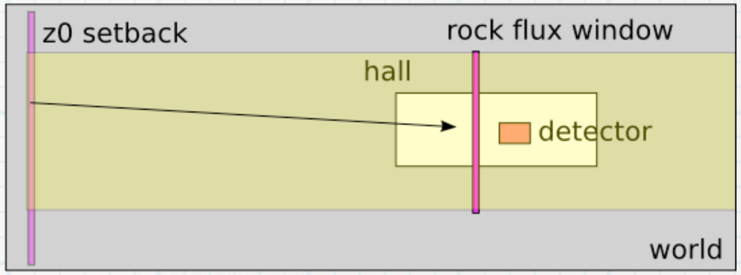
- Detectors aren’t isolated in space
  - surrounding rock acts as targets too!
    - (aside) MINOS Near Hall model is not very accurate
      - ...just a rectangular box
        - (also side note: Minerva flux window is too small -- see 3D representation)



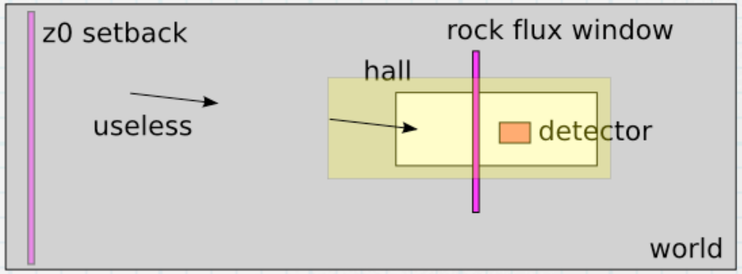
# Choosing a Vertex “Outside the Box”

- When a “topvol” isn’t set, GENIE considers the entire geometry
- GeomSelectorRockBox trims the volume to the hall + minimum safety + a size proportional to the neutrino energy
- Integrated into GENIEHelper
  - but not fully tested
  - needs adjustment for asymmetrical box

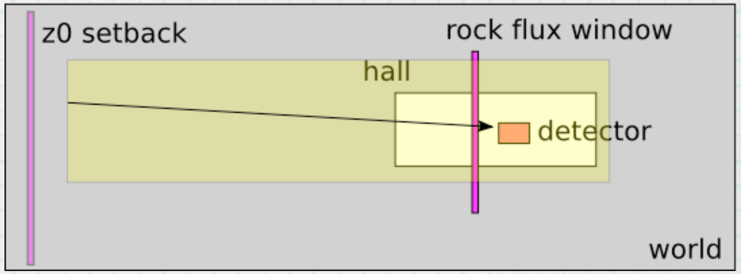
No GeomSelector



RockBox: 2 GeV

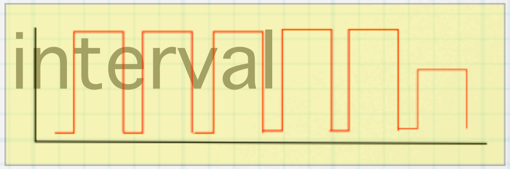


RockBox: 80 GeV



# How to overlay

- All the expt NearDet see flux intensity high enough for multiple interactions per spill
- Collect events
  - MINOS: pull from input sample files
    - Poisson distribution - nDetPerSpill + nRockPerSnarl for a given intensity
    - single use of detector events, randomize pulling from rock files (reuse, except once)
  - NOvA: generate events until used X POTs
- Distribute events in time over spill interval according to intensity profile



  - offset truth info times (StdHep/HepMC)
  - also offset corresponding hit times, if already propagated in GEANT
  - if combined particle list, adjust parentage indices
  - add any event kinematics/flux records to list for spill
    - good to have mechanism tying kin/flux to particle list

Geant 4

# Shower & Track Physics

| simulation step | software | work product |
|---|---|---|
| Beamline | flugg { fluka<br>Geant4 } | flugg.root |
| | GENIE | gsimple.root |
| | nova (ART framework) | art::Event (art::Run) |
| ν Interactions<br>or other "generator" | GENIE<br>or CRY/SingleParticle | + v<MCTruth>,v< MCFlux><br>(RunData, POTSum)<br>MCTruth hold a vector<MCParticle> |
| **Energy Depositions** | **Geant4** | + vector<FLSHitList><br>vector<ParticleList> |
| Light Propagation | Simple<br>PhotonTransport<br>^ | + vector<PhotonSignal> |
| DAQ Electronics | ReadoutSim | + vector<RawDigit><br>vector<RawTrigger> |
| Reco/Analysis | UserCode | Clusters, Prongs, Tracks<br>+ histograms |

*final state only*

Geant 4

# End of Simulation

| simulation step | software | work product |
|---|---|---|
| Beamline | flugg { fluka<br>Geant4 } | flugg.root |
| | GENIE | gsimple.root |
| | | art::Event (art::Run) |
| | | + v<MCTruth>,v< MCFlux><br>(RunData, POTSum)<br>MCTruth hold a vector<MCParticle> |
| Energ | | + vector<FLSHitList><br>vector<ParticleList> |
| Ligh | | + vector<PhotonSignal> |
| DAQ Electronics | ReadoutSim | + vector<RawDigit><br>vector<RawTrigger> |
| Reco/Analysis | UserCode | Clusters, Prongs, Tracks<br>+ histograms |

The goal of a simulation is to create "fake data" that mimics what the real detector would give, but includes truth information.

# Final Notes

- Many steps are computationally expensive
  - combinatorics are a bear (detectors, beams, mixing…)
  - wise to save and share common files
    - sometimes even (sensible) intermediate stages
    - if one is only interested in an early stage and files aren't available, then do only the necessary steps
      - if you're validating GENIE truth info and not using the hits or beyond, then only run through the generator stage.
- Make use of random seeds to "mix it up" without having to re-run earlier stages
    - especially the first GENIE ν interaction; reuse flux files
- As far as possible try to validate individual steps, not the whole chain at once