Flux and Geometry drivers Summary

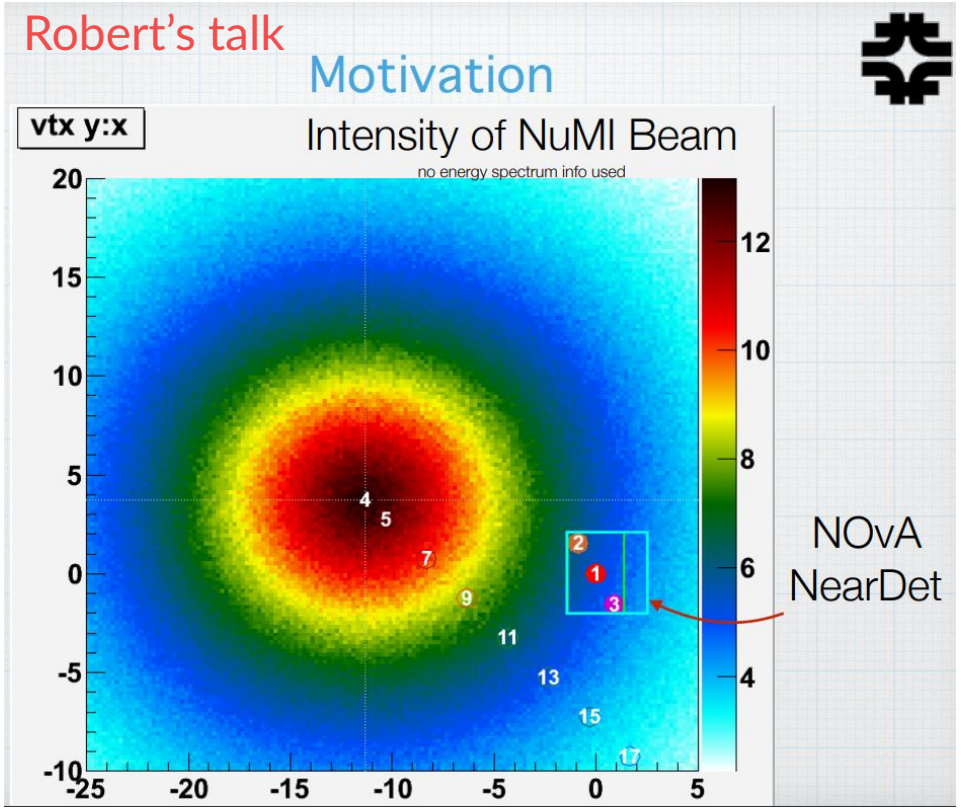
UNIVERSITY of ROCHESTER

Clarence Wret, Laura Fields, Joshua Isaacson
Robert Hatcher, Costas Andreopoulos
Hayato-san, Luke Pickering, ...

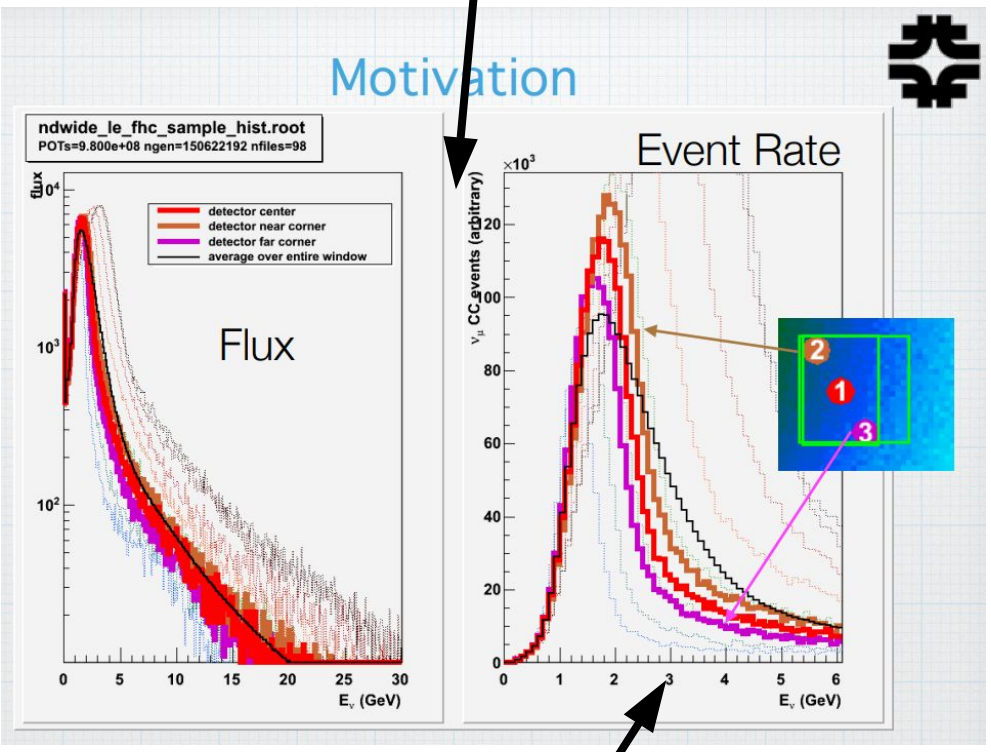
Generator Tools Workshop, FNAL
10 January 2020

Flux vs Geometry drivers

- Flux driver decides on which E_ν is generated
  - Experiment provides flux in some format
  - Flux driver returns E_ν to generate event with



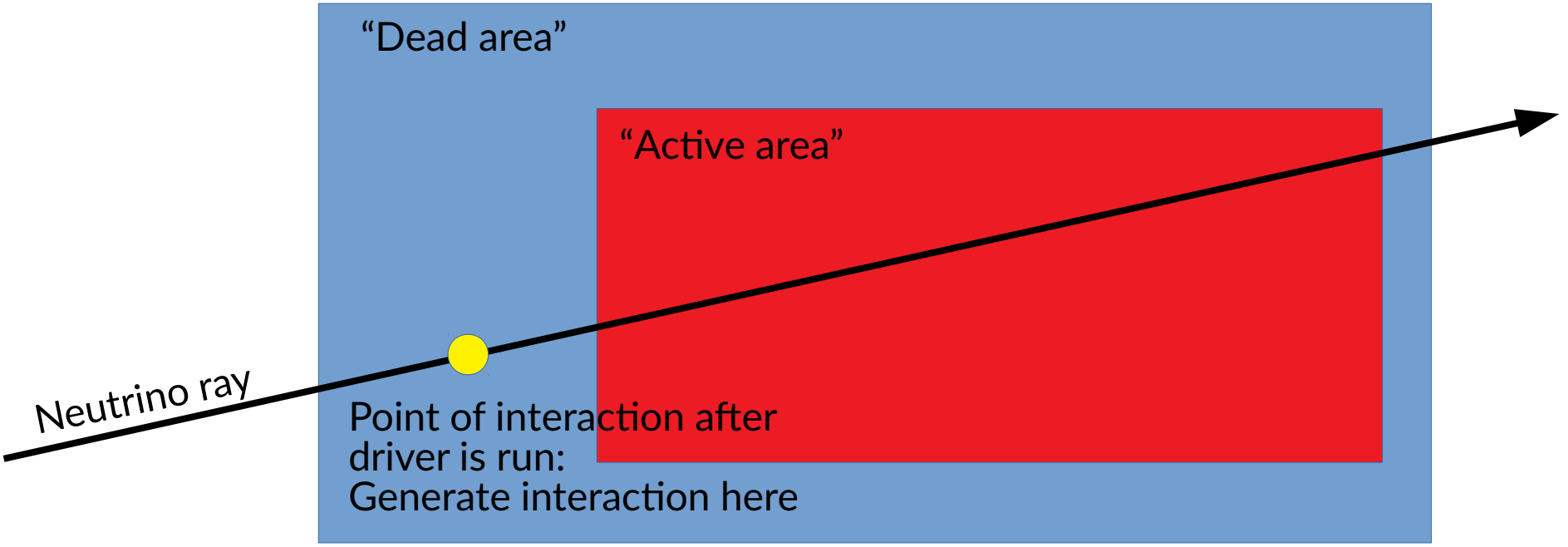
Enter your favourite generator
→ different event rate



Flux driver returns an E_ν

Flux vs Geometry drivers

- Geometry driver decides on where interaction happens
  - Generator/theory provide $\sigma(E_\nu)$ for all targets in volume
  - Experiment provides geometry and targets of detector
  - Geometry driver returns point of interaction



Flux vs Geometry drivers

- A common flux and geometry driver will ease:
  - Multiple generators in experiments' productions
  - A step towards theory-driven "production"

- The former is achievable on a short time-scale
  - GENIE, NEUT and NuWro all have their own versions of these
  - Unifying would mean much less effort for experiments running multiple productions
  - Large benefit from having GiBUU available to experiments
- The latter requires a lot more work, e.g. theory event generation, interface to FSI routines: discussed in other summary talks

Flux driver discussions

- Fairly straightforward, Robert's talk
- Costas' talk brought forward a few options, two of which have been highlighted

- Option "2b)" Use GENIE's driver and a thin wrapper interfacing to a different generator or theory:
  - GENIE returns E_ν via its current flux driver
  - Minimal amount of work required (?)
  - Some discussion about separating from GENIE
    - GENIE dependencies not needed, e.g. PYTHIA, LHAPDF
    - GENIE folks needed for the surgery for barely any GENIE benefit; may not have people for this
  - MOU needed for licensing?

Flux driver discussions

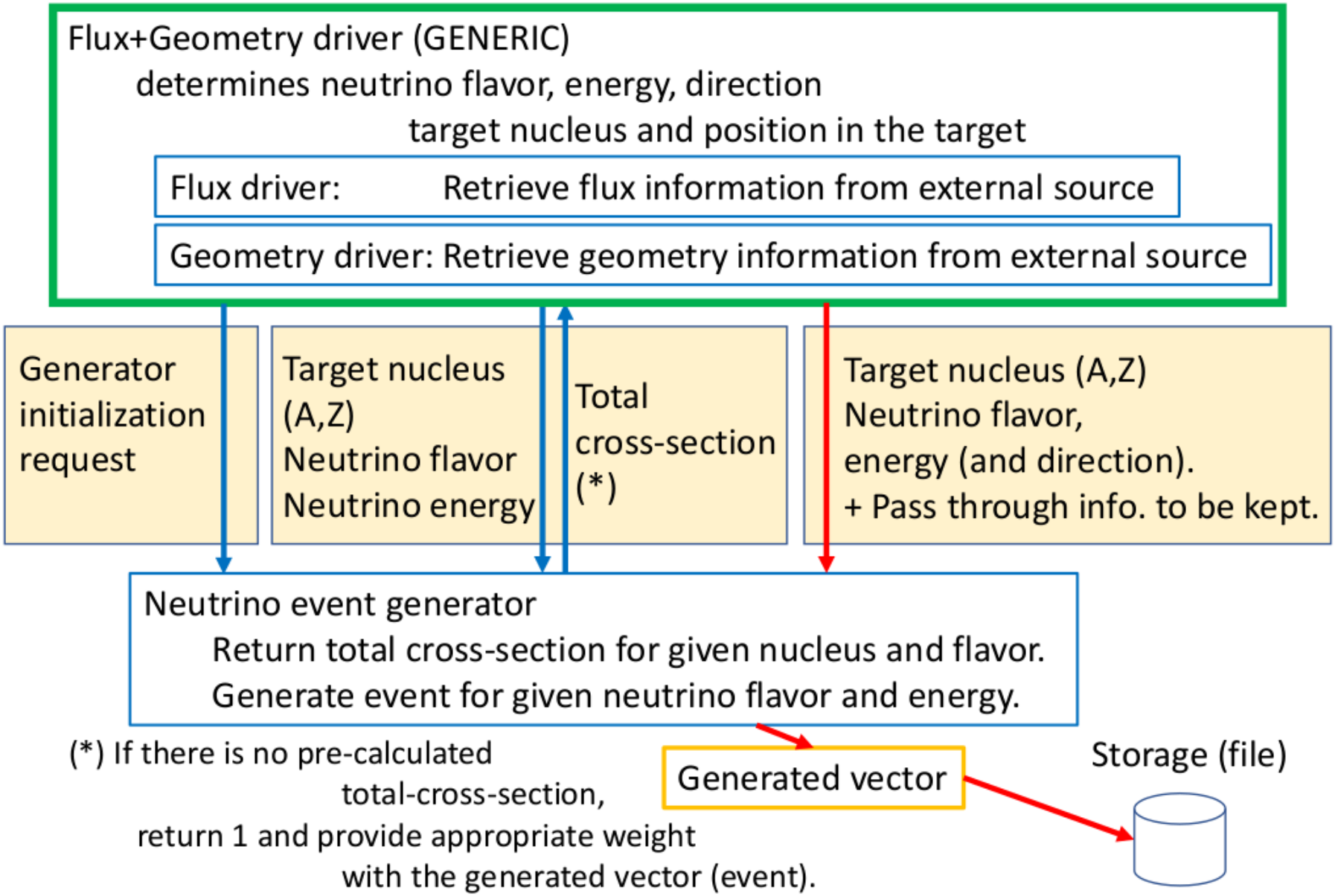
- "Community based":
  - Write a purpose-built flux driver for community
  - Larger workload, but avoids "issues" of GENIE's "2b)"
  - First support accelerator and atmospheric fluxes, then see if community wants more?
  - Likely to benefit experiments wanting other generator productions: is this where responsibility falls?
    - e.g. NEUT production at NOvA, GiBUU at T2K

- Personally opinion: the only issue here is maintainability and defining who's takes responsibility of the product
  - Lab, university, collaboration, NuSTEC?
  - Form a working group?

- (I think) Luke has played with this already

Community based drivers

- Hayato-san and Luke have been bouncing ideas over email



Flux driver discussions

- GENIE does not enforce a flux format: reads in types as they become available to collaboration

Branch: master ▾ **Generator** / src / Tools / **Flux** /

mroda88 Sync with master

..

GNuMINtuple	put GNuMIFlux.xml into config area
GAstroFlux.cxx	Bump copyright year to 2019
GAstroFlux.h	Bump copyright year to 2019
GAtmoFlux.cxx	Bump copyright year to 2019
GAtmoFlux.h	Bump copyright year to 2019
GBGLRSAtmoFlux.cxx	Bump copyright year to 2019
GBGLRSAtmoFlux.h	Bump copyright year to 2019
GCylindTH1Flux.cxx	Sync with master
GCylindTH1Flux.h	Bump copyright year to 2019
GFLUKAAtmoFlux.cxx	Bump copyright year to 2019
GFLUKAAtmoFlux.h	Bump copyright year to 2019

GFlavorMap.cxx	Now Flux should compile
GFlavorMap.h	Flux compiles
GFlavorMixerFactory.cxx	Now Flux should compile
GFlavorMixerFactory.h	Flux compiles
GFlavorMixerI.cxx	Flux compiles
GFlavorMixerI.h	Start filling Tools subdir
GFluxBlender.cxx	Now Flux should compile
GFluxBlender.h	Flux compiles
GFluxDriverFactory.cxx	Now Flux should compile
GFluxDriverFactory.h	Flux compiles
GFluxExposureI.cxx	Flux compiles
GFluxExposureI.h	Start filling Tools subdir
GFluxFileConfigI.cxx	Flux compiles
GFluxFileConfigI.h	Flux compiles

GHAKKMAtmoFlux.cxx	Bump copyright year to 2019
GHAKKMAtmoFlux.h	Bump copyright year to 2019
GJPARCNuFlux.cxx	Bump copyright year to 2019
GJPARCNuFlux.h	Bump copyright year to 2019
GMonoEnergeticFlux.cxx	Bump copyright year to 2019
GMonoEnergeticFlux.h	Bump copyright year to 2019
GNuMIFlux.cxx	Bump copyright year to 2019
GNuMIFlux.h	Bump copyright year to 2019
GSimpleNtpFlux.cxx	Bump copyright year to 2019
GSimpleNtpFlux.h	Bump copyright year to 2019

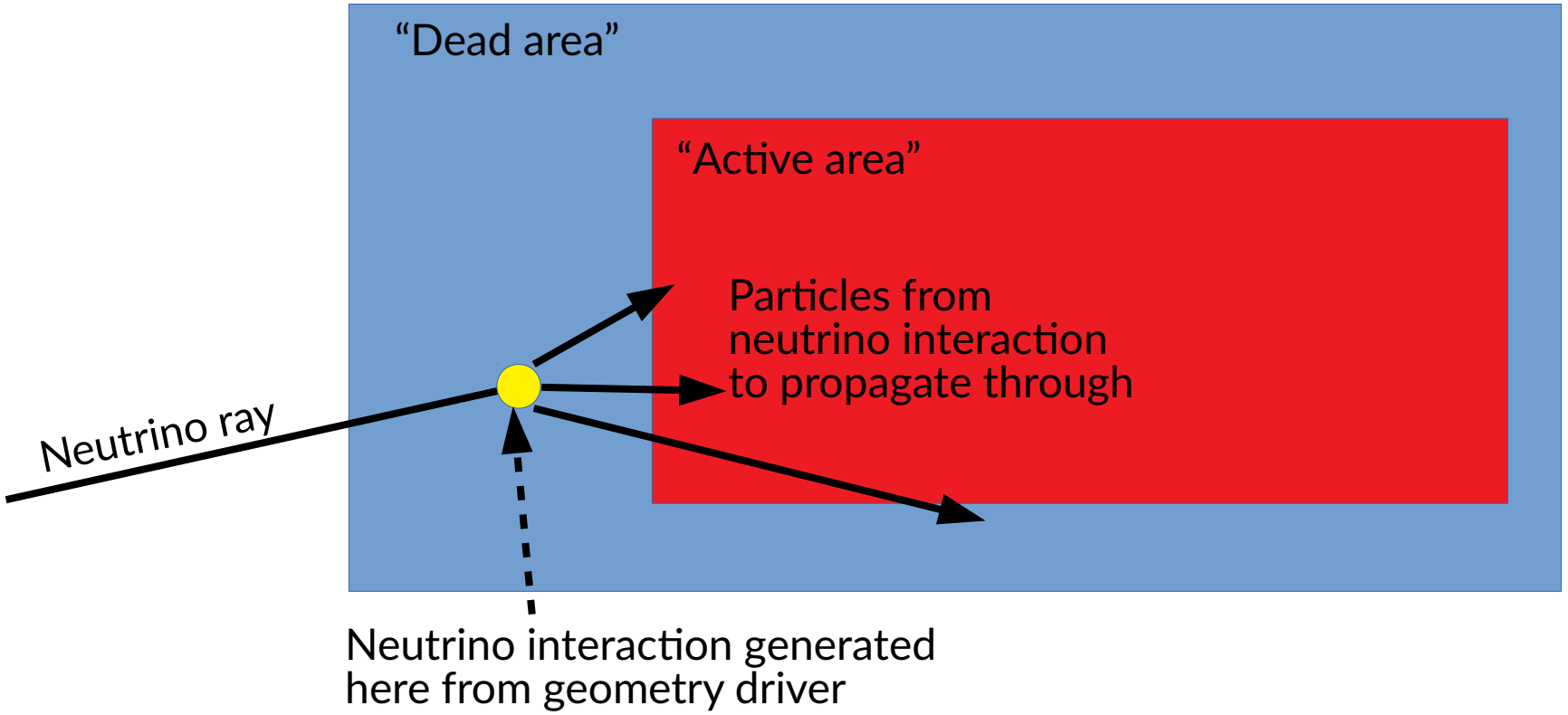
- Briefly discussed unifying flux output from accelerator experiments
  - e.g. J-PARC providing a dk2nu format
  - Seems easier (for now) to code up for different flux formats instead
  - May be something to pursue in the future

Geometry driver discussions

- GENIE, NuWro, NEUT already has some form of geometry implemented
- All use GDML and/or TGeoManager in ROOT
- The tracing algorithm through the volume is not unified
  - (Appears) all generators do things slightly different
  - Different levels of computational optimisation
  - But (probably?) with the same outcome
- Path forward essentially follows same as flux driver
  - Keep flux+geometry driver one package?
- Use GENIE alongside thin wrapper for other generators providing $\sigma(E_\nu, \text{target})$, or write from scratch

Flux vs Geometry drivers

- Maybe missing in our discussion:
  - Our defined “geometry driver” doesn’t include procedure of handing over neutrino simulation products to Geant4 for tracing through detector



  - Needed to provide fully generator agnostic package for experiments
  - Requires common event format (or translators) for outgoing particles

Conclusions

- Very productive talks and discussions on common flux and geometry driver
- For the time being, will pursue both “GENIE option 2b” and “community driver”
  - Kevin’s notes mentions Robert, Luke and Clarence working on prototyping and working on these
  - Clear interest from Hayato-san too
- Seems likely to develop further, but needs adequate support for work to begin
- May be fruitful to commence a small working group?
  - Some gray areas of the future implementation

Thanks, let's discuss!