Beyond the Standard LArSoft

Gianluca Petrillo

SLAC NATIONAL ACCELERATOR LABORATORY

PITT PACC Workshop: Nu Tools for BSM at Neutrino Beam Facilities, December 16, 2022

Outline

Who speaks the speaker

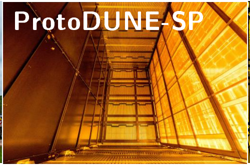
- at SLAC since 2018
  - DUNE, including BSM, but not very much these days
  - SBN/ICARUS
    (trigger, optical sim/reco, software support, ...)
- earlier, 4.5 years at Fermilab
  - development, maintenance, user support of LArSoft

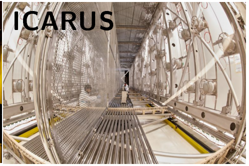
Outline

What is LArSoft, who uses it

LArSoft is a toolkit for simulation, reconstruction and analysis of liquid argon TPC:

- born from Brian Rebel as software for ArgoNeuT
- taken over by Fermilab with the ambition of being the common toolkit for all LArTPC experiments therein (and beyond)
- successfully used by ArgoNeuT, MicroBooNE, LArIAT and ProtoDUNE
- adopted by SBND, ICARUS and DUNE (only Far Detector)
- also, never particularly loved by any of the users...

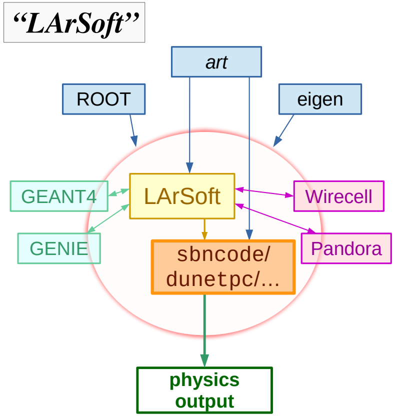




What is LArSoft, and what is not

LArSoft is in the middle of an "ecosystem":

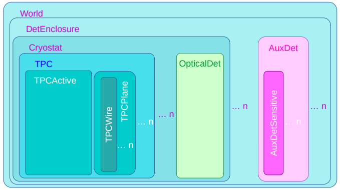
- LArSoft is a modular "toolkit" which knows of LArTPC and physics
- experimentcode contains extensions and algorithms specific to each experiment
- Fermilab *art* framework glues the tools: it provides data I/O (ROOT or, in principle, HDF5), event loop, and the management of workflow with the modules
- then there are all the other utility libraries (e.g. CLHEP)...
- ... and the physics ones (e.g. GENIE, Pandora, ...)



What LArSoft does, what it could do

LArSoft is modular enough to be widely extensible, but in its core it supports:

- LArTPC: volumes of LAr with a wire readout by any number of planes on one side *(simple geometry, 2D (wire-like) readout)*
- argon scintillation ("optical") detection with single readout
- other detectors, most commonly scintillators for cosmic ray detection



LArSoft grows organically:

- ArgoNeuT: single drift volume, two wire planes, no optical detector
- MicroBooNE: third plane, optical detection, cosmic ray taggers
- DUNE-35t: anode shared between drift volumes, and multiple TPC with varying size
- SBND and ICARUS: cathode-in-the-middle configuration
- ICARUS: two cryostats
- ...

Outline

Stages of a LArSoft workflow

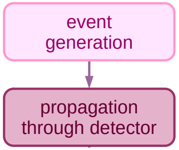
A typical LArSoft workflow proceeds in five stages:

- within LArSoft, interfaced generators (GENIE, CORSIKA), particle lists

→ a list of particles (`simb::MCParticle` objects)

Stages of a LArSoft workflow

A typical LArSoft workflow proceeds in five stages:



- within LArSoft, interfaced generators (GENIE, CORSIKA), particle lists

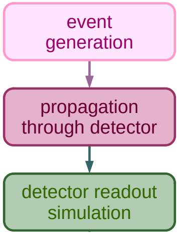
$\rightarrow$ a list of particles (`simb::MCParticle` objects)

- GEANT4, scintillation, ionisation, transportation physics

$\rightarrow$ many more particles, energy depositions, charge and light at the detectors

Stages of a LArSoft workflow

A typical LArSoft workflow proceeds in five stages:



- within LArSoft, interfaced generators (GENIE, CORSIKA), particle lists

$\rightarrow$ a list of particles (`simb::MCParticle` objects)

- GEANT4, scintillation, ionisation, transportation physics

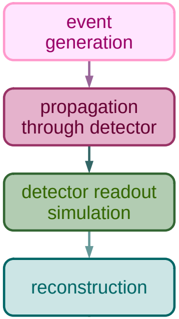
$\rightarrow$ many more particles, energy depositions, charge and light at the detectors

- mostly readout electronics simulation

$\rightarrow$ waveforms (TPC, OpDet) and hits (CRT): like the real detector ($\pm$)

Stages of a LArSoft workflow

A typical LArSoft workflow proceeds in five stages:



- within LArSoft, interfaced generators (GENIE, CORSIKA), particle lists

→ a list of particles (`simb::MCParticle` objects)

- GEANT4, scintillation, ionisation, transportation physics

→ many more particles, energy depositions, charge and light at the detectors

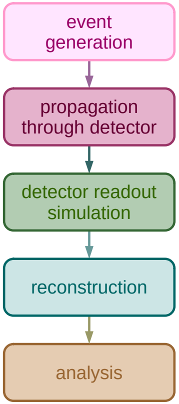
- mostly readout electronics simulation

→ waveforms (TPC, OpDet) and hits (CRT): like the real detector ($\pm$)

→ higher level objects: tracks, e.m. showers, particle identification, energy reconstruction

Stages of a LArSoft workflow

A typical LArSoft workflow proceeds in five stages:



- within LArSoft, interfaced generators (GENIE, CORSIKA), particle lists

→ a list of particles (`simb::MCParticle` objects)

- GEANT4, scintillation, ionisation, transportation physics

→ many more particles, energy depositions, charge and light at the detectors

- mostly readout electronics simulation

→ waveforms (TPC, OpDet) and hits (CRT): like the real detector ($\pm$)

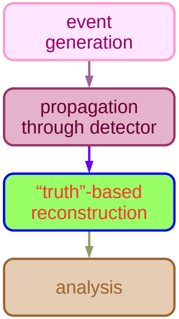
→ higher level objects: tracks, e.m. showers, particle identification, energy reconstruction

→ plots, papers[a], ~~big money, power and~~ glory

[a] For presentation purposes only. Paper generator not included.

Stages of a LArSoft workflow

A typical LArSoft workflow proceeds in five stages:



- within LArSoft, interfaced generators (GENIE, CORSIKA), particle lists

$\rightarrow$ a list of particles (`simb::MCParticle` objects)

- GEANT4, scintillation, ionisation, transportation physics

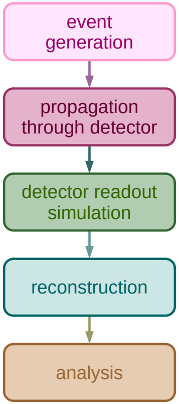
$\rightarrow$ many more particles, energy depositions, charge and light at the detectors

- TPC only: higher level objects (tracks, e.m. showers) and association to generated particles

$\rightarrow$ plots, papers, ~~big money, power and~~ glory

Stages of a LArSoft workflow

A typical LArSoft workflow proceeds in five stages:



- within LArSoft, interfaced generators (GENIE, CORSIKA), particle lists

$\rightarrow$ a list of particles (`simb::MCParticle` objects)

- GEANT4, scintillation, ionisation, transportation physics

$\rightarrow$ many more particles, energy depositions, charge and light at the detectors

- mostly readout electronics simulation

$\rightarrow$ waveforms (TPC, OpDet) and hits (CRT): like the real detector ($\pm$)

$\rightarrow$ higher level objects: tracks, e.m. showers, particle identification, energy reconstruction

$\rightarrow$ plots, papers, ~~big money, power and~~ glory

It's easy to stop at any point, very very hard to reenter.

From the universal to the particular

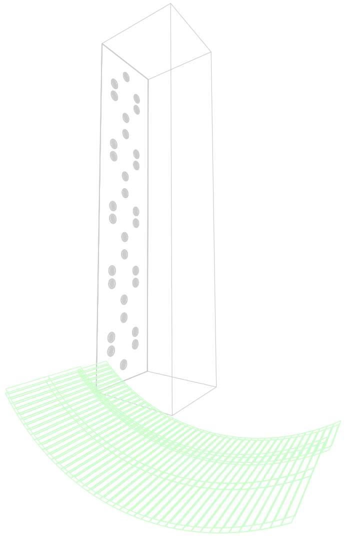
Some aspects of the workflow are ✓“universal”, other ✗require customisation:

- ✗ detector geometry is, surprise surprise, *detector-specific*
- ✓ charge simulation is mostly universal (except field distortions)
- ✗ light propagation is geometry-dependent (you can create your light map… it just takes CPU time)
- ✗ readout electronic response (TPC, OpDet, CRT) are stubbornly non-universal
- ✗ trigger response is also wildly dependent on the experiment
- ✓ reconstruction is a mixed bag, with the most sophisticated techniques being detector-dependent
- ✗ no reconstruction of cosmic ray taggers is included

A generic LArTPC detector

An aside: effort has been initiated (Fernanda Psihas, Gray Putnam) to provide a “generic” workable simulated LArTPC detector with no royalties

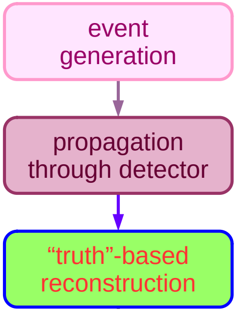
- goal: prototype physics analyses and studies with LArTPC...
- ... that can then be more easily ported to real Experiments (or across Experiment boundaries)
- detector geometry inspired by MicroBooNE’s
- intended to be fully included in the plain LArSoft



Drawing the GDML geometry with ROOT took me an embarrassing amount of time; the result is even more embarrassing.

Ours are exiting times! (I)

Maybe you don't want to undergo all that pain, but just some. Where to exit?

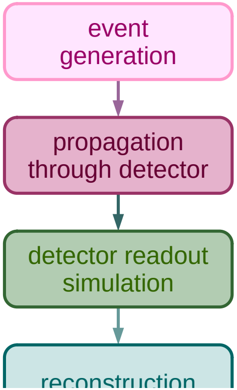


Stop with cheating reconstruction of TPC (no light):

- ... then proceed to the analysis
- requires: a detector geometry (mock up *might* be ok)
- provides a credible view of what can be detected
- grossly overestimates reconstruction efficiency!
  $\rightarrow$ will likely need amendment with smearing and thresholds

Ours are exiting times! (II)

Maybe you don't want to undergo all that pain, but just some. Where to exit?



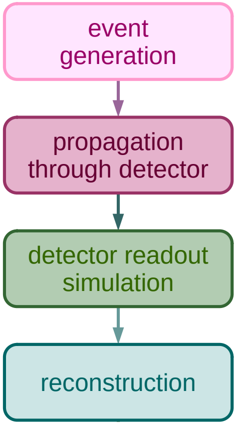
Stop with early reconstruction:

- requires a detector geometry, TPC response, and electronics simulation
- low level reconstruction: "hits", or 3D charge deposits...
- ... then you can build your simple (pseudo)reconstruction and proceed

SLAC Machine Learning reconstruction follows this route, branching after matching hits into 3D points.

Ours are exiting times! (III)

Maybe you don't want to undergo all that pain, but just some. Where to exit?



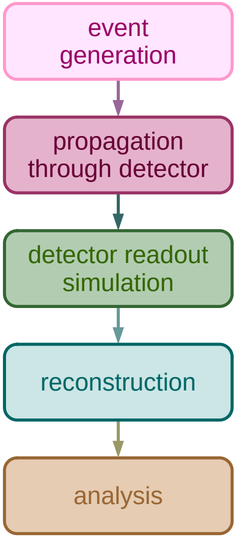
Stop like a pro, after the full reconstruction:

- ... then proceed to the analysis
- requires: a detector geometry, TPC response, and electronics simulation
- using generic, untuned reconstruction algorithms may be ineffective
- but keep an eye on the generic LArTPC detector project: they may have some "tuned" performance

Most of users I know go this way *(backed by an official experiment software)*:

- dump the reconstructed information in a ROOT tree/CSV file/...
- proceed with analysis in ROOT/Pandas/R/...
- many SBN users have adopted a Common Analysis Format (ROOT) designed for oscillation analyses

Ours are exiting times! (IV)



Maybe you don't want to undergo all that pain, but just some.
Where to exit?
More pro than a pro:

- analysis can be written in a LArSoft (*art*) module;
  does anybody do that?
- or use a lightweight library ("*gallery*") to write analysis:
  - as C++ stand alone or ROOT macro, or as Python script
  - uses and depends on most of LArSoft dependencies (UPS...)
  - from the same authors of *art*, but more limited features
  - TITUS event display is based on *gallery*

Outline

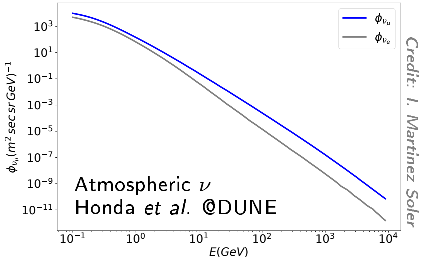
Event generators

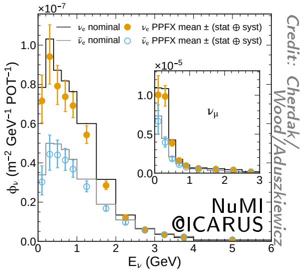
- LArSoft's most advanced interface is with GENIE neutrino generator:
  - users configure a LArSoft module, don't even talk to GENIE
  - the interface takes care of picking a neutrino from a flux…
  - … picking a nucleus type the neutrino passes by…
  - … and finally placing the interaction in an appropriate place (and, once fixed, time)
- other interfaced generators: MARLEY, CORSIKA (indirectly), CRY
- LArSoft also accepts pre-generated events in some form of HEPEVT
- for everything else, one has to write code equivalent to the above
  - not too hard, but not for the casual coder either
    (a lot of C++ to chew… but every experiment has a few "guru" to ask help to)
  - ROOT can help with navigation of the geometry (ray tracing)
  - fluxes from beams may be more complicate to implement
  - for nucleus final state interactions, can we plug into GENIE?
- LArSoft weight tracking is close to non-existent
- a coordinated effort for a more generic interface may go far

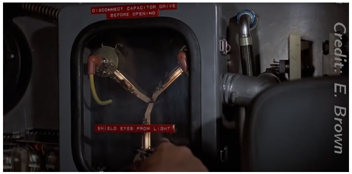
Beam flux

To generate BSM events from accelerator sources, we need a flux:

- SBN BNB neutrinos: pregenerated, then just pick one
- SBN NuMI neutrinos: fluxes are more complicate
    - off-axis source: particles hit from many directions
    - full position, momentum and hadronic "history" are stored
- the hadron information in a flux like NuMI above can be used to generate BSM
- also note that Fermilab experiments tune that flux with data







Beam timing and triggering

Usually neutrino interactions are background to BSM signature. To mitigate their impact:

- BSM particles are likely heavier and slower than neutrinos
- a time delay can be exploited by looking for events:

→ in between the neutrino "buckets" making the spill
  - e.g. BNB has a FWHM ≈ 2 ns bucket every 19 ns
  - requires an extremely precise timing (hard but possible)
  - requires detailed knowledge of optical detector
  - SBN detectors might reach resolution of $\mathcal{O}(1\,\mathrm{ns})$

→ after the end of the neutrino spill
  - you need a readout that extends beyond the beam…
  - … and the trigger too (or you piggyback on other in-time triggers)

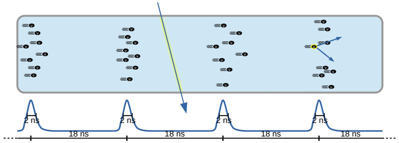
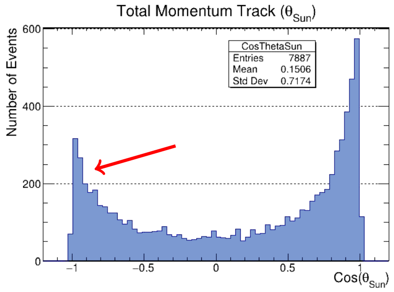


Illustration of BNB neutrinos in the ICARUS detector. With a careful reconstruction of its position and time, an interaction can be tagged as not from a beam neutrino.

Reconstruction biases

- "standard" reconstruction algorithm tuning focusses on beam neutrinos
- *most* beam-generated BSM activity will be fine…
  but off-axis sources (like NuMI for SBN/ICARUS) may take a hit, cosmic sources a big one
- typical example: Pandora reconstruction prefers to assign a beam-compatible verse to cosmic particles $\rightarrow$
- we may get away "cheating" with the assumption that future improvements will get rid of such biasses
- using a correct orientation and geometry is also important, e.g. the DUNE far detector is 60 m of argon in the beam direction, much thinner for signals from above



Reconstructed angle of particles from interaction of Sun-born Boosted Dark Matter. The generated value is $\cos\theta = 1$, but wrong verse reconstruction creates an artificial enhancement of opposite direction.

Work by L. Peres, UFRJ, for DUNE.

Computing resources

Anecdotal generation experience:

- DUNE atmospheric neutrinos
  - there we chose to use a *reduced* DUNE geometry: roughly 1/10 of a single far detector module
  - reasonable strategy for "small" events
- recent ICARUS BNB neutrino + cosmic ray background
  - standard MC production chain (geometry, corrections...)
  - especially data size changes often (all the rest too)

Figures are from small test batch (uncertainty: $\pm 50\%$?):

sample	"G4"	digit.	reco.	size
DUNE (≈ 1 kton) atmo. ν	20"	3"	12"	
ICARUS: ν + cosmic rays	2'	1'	3.5'	250 MB

Turns out, anyway, that the largest problem is *memory*.

Data preservation and reprocessing

For the experiment: preserve to investigate/rerun/refine an analysis in the future

- need "perpetual" access to data (and to opportunistic computing)
- need the code and configuration
  - + all LArTPC experiments I know keep their "LArSoft code" in GIT with tags
  - + *art* data file carry the full configuration history
  - – execution and build environment: a "container" is great… but LArSoft does not support any
  - – … is that calibration/run conditions/… database still online?
  - – tracking discipline is quite strong for official productions… for analyses usually not so much

For the rest of the community: understand/reinterpret data used for other analyses

- high level data (raw data is unusable — because of license, means and know how)
- need plenty of *meta*data:
  - – details of the algorithms
  - – generation, simulation, reconstruction parameters, configurations, etc.
  - – understanding of the limits of the reconstruction also helps
    (e.g. why is my 10 GeV cosmogenic particle not well reconstructed?)

Outlook

- LArSoft is firmly adopted by SBN; DUNE is farther, not all LArTPC... the end is not foregone
- I have talked a lot about LArSoft, but many topics are relevant also with other toolkits
- I did not offer solutions, but mostly discussion seeds
- also remember that LArSoft is basically developed by people in the using experiments (so, chances are, also by *you*)

Thank you for your attention!

The build environment

- a long-standing effort aims to standardise building LArSoft and avoid Fermilab-specific infrastructure
- still some way to go, not clear to me where we'll land
- it is possible to run LArSoft locally, but the first setup may be discouraging
  - a Scientific Linux 6 container/virtual machine makes it easier
- also, data files may become big

Event-based data records and continuous data taking

A staple of *art* framework and LArSoft is the concept of *event*:

- usually, experiment readout too: beam is pulsed, one pulse $\rightarrow$ one event
- if there are multiple independent interactions, LArSoft is of little help
  - Pandora reconstruction offers interaction-based "slices"
- if the activity extends over long time, it becomes complicate
  - associating neutron activity far away from the interaction in time and space is challenging
  - correlating hits of a scattering millicharge particle, too
  - on the other end, LArTPC is slow: readout may be active long enough for your $\tau = 100$ µs decay
- processing data from continuous readout is also a challenge