Scintillation Light in LArTPCs: Simulation and Reconstruction

Patrick Green

University of Oxford

Adapted from previous slides written by Andrzej Szelc and Diego Garcia-Gamez

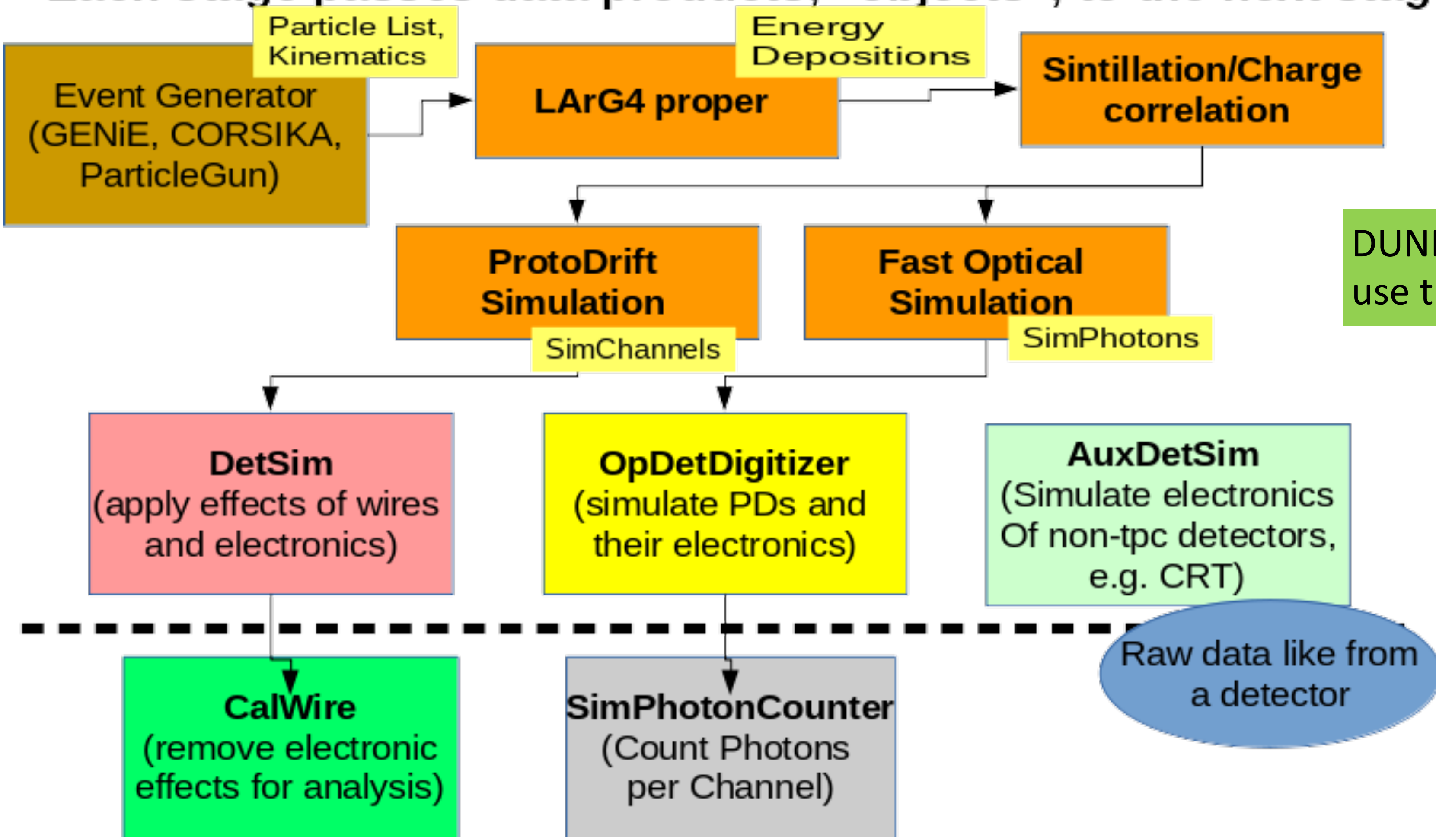
UK-LArSoft workshop, Edinburgh, October 2024

Outline

- This talk will give an overview of how LArSoft deals with simulating light and why it's hard
- I will also talk a bit about reconstruction
- Next, we will go through a few hands-on examples in the tutorial.

Simulation Flowchart

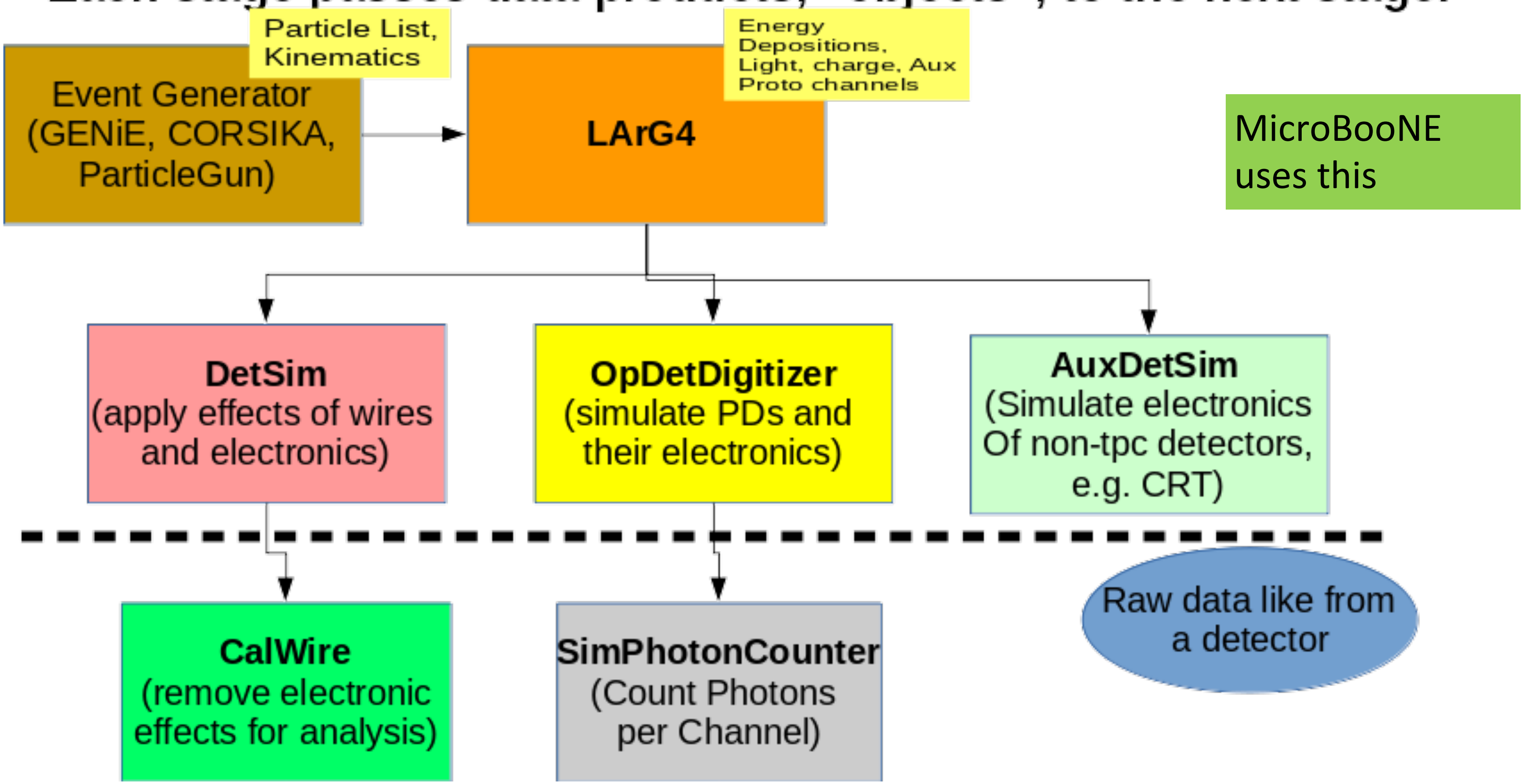
Each stage is a module
Each stage passes data products, “objects”, to the next stage.



DUNE, SBND use this

Simulation Flowchart (Legacy Version)

Each stage is a module
Each stage passes data products, “objects”, to the next stage.



Elements of Light Simulation

- **Light source:**
  - How many photons are generated?
  - What is their time distribution?
  - What is their wavelength?

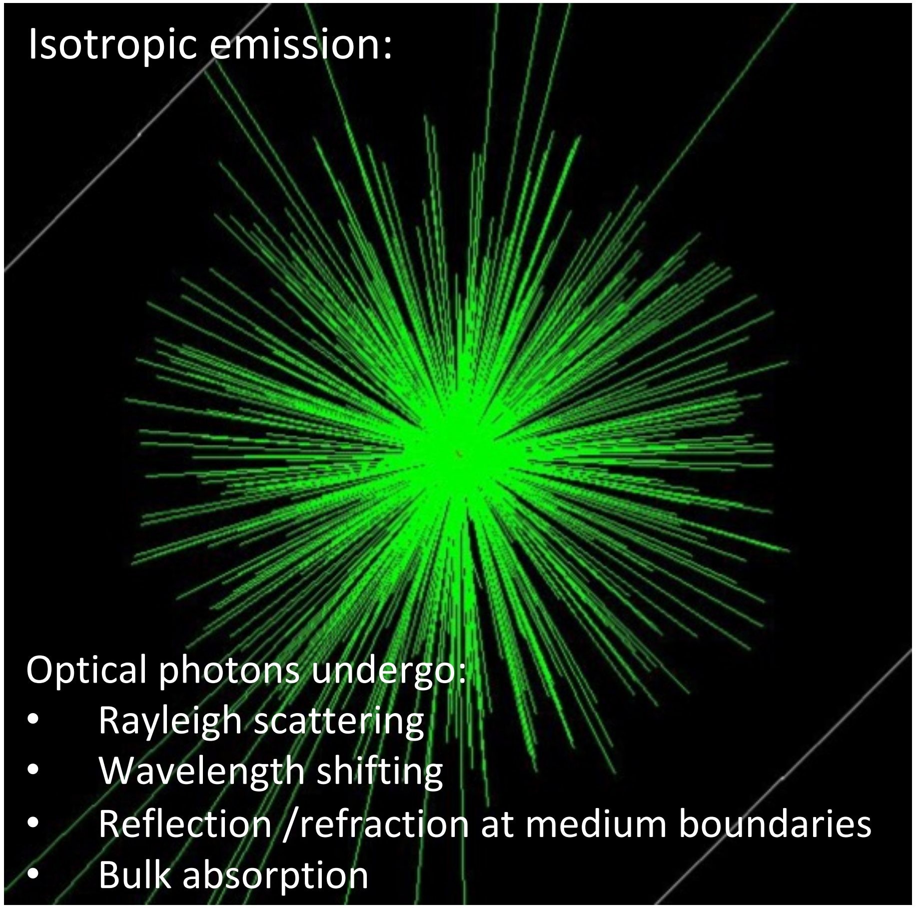
- **Transport:**
  - How many photons make it to the detector?
  - How long does it take them?
  - Do they scatter / get absorbed / reflected etc?

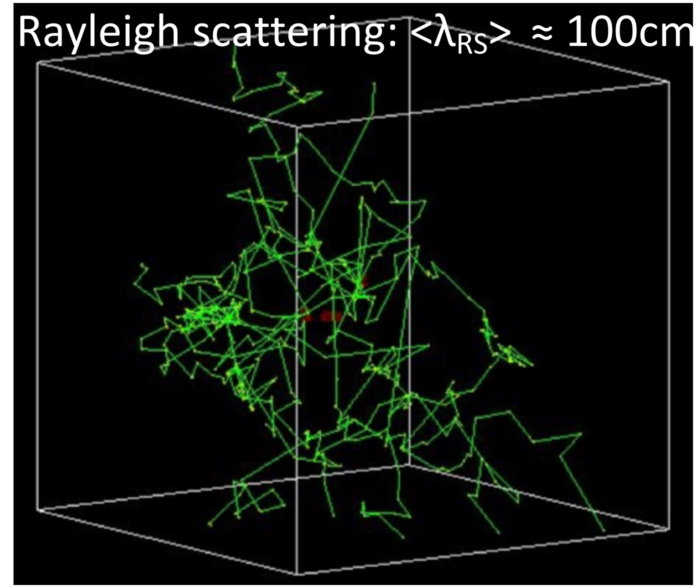
- **Detection:**
  - What is our detection efficiency?
  - Does it depend on position on detector?
  - Are there any extra timing effects?

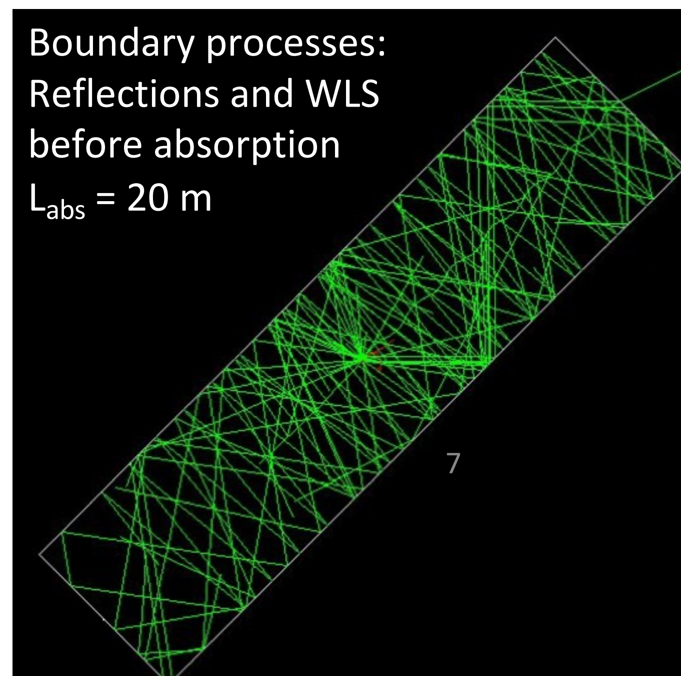
Different modes of simulation

- *Full optical simulation* (extremely slow)
  - requires definition of all optical properties.
- *Fast optical simulation* (faster, but less precise)
  - still need to run full optical at least once
  - majority of optical properties "burned in"
  - three primary methods exist: optical library, semi-analytical, generative neural network

Full optical light simulation







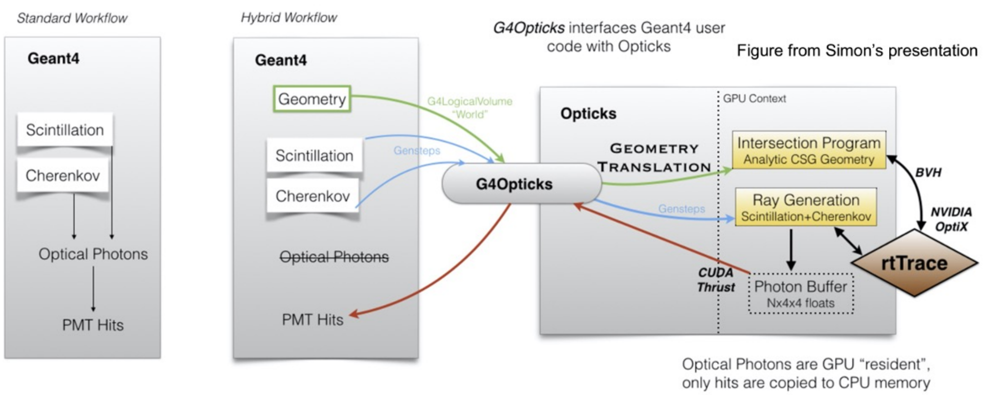
Scintillation yield ~24000 photons/MeV

For GeV-scale interactions in large detectors, the tracking of each individual photon is prohibitively slow -> alternatives needed

(Aside): Using GPUs – NVIDIA OptiX

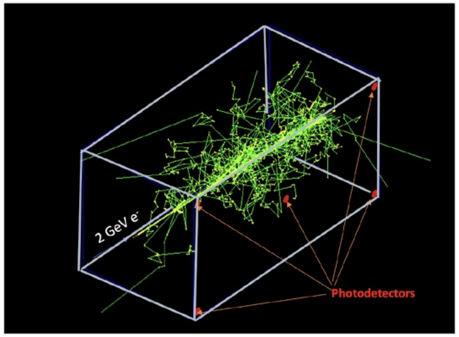
Some progress using GPUs for full optical simulation, but challenges remain
- GPU resources limited + using vendor specific tools (NVIDIA OptiX)

GPU based photon simulation



- simulation of photons within open detectors can be incredibly CPU intensive and limit event by event simulation
- often solved with voxel-ized lookup tables
- Opticks developed by Simon Blyth for JUNO, now part of GEANT4 releases

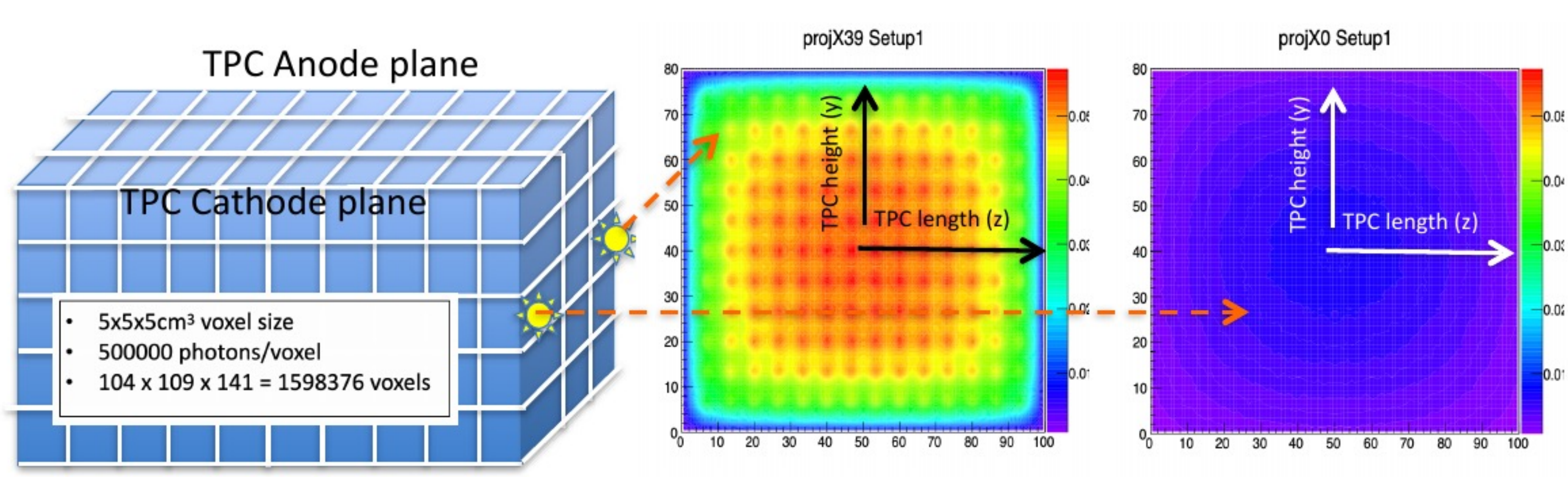
- CaTS: integrate GEANT4+Opticks in LArSoft for simplified LArTPC
- speedup of several 100 times for photon simulation
- also report that 1 core could saturate the GPU



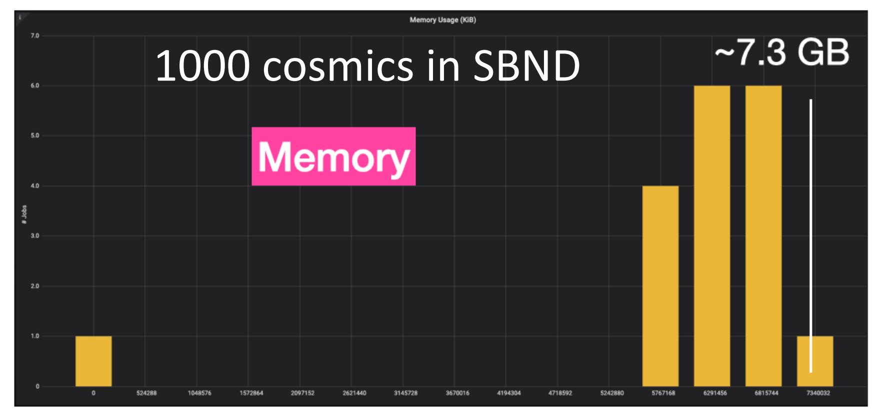
Brookhaven National Laboratory

Slide stolen from Michael Kirby, NuFact 2024

Fast optical model: Optical Library

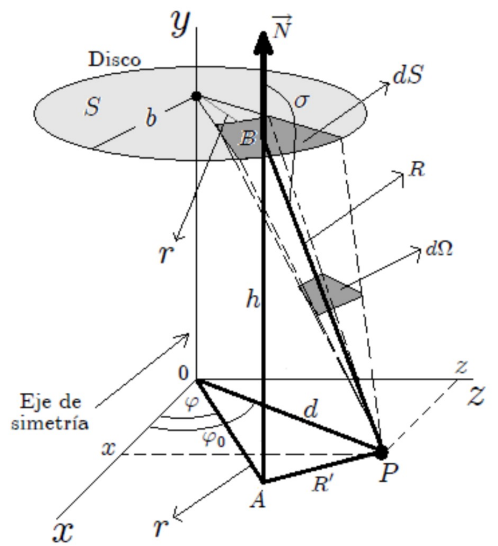


$$\langle N \rangle_{PMT-hits} = \left(\frac{dE}{dx}_{step} \cdot Length_{step} \right) \cdot LY \cdot visibility_{step}^{PMT}$$



- Resolution depends on voxel sizes: granularity effects at short distances
- Optical library size scales with detector size and number of photon detectors
- Prohibitive memory use for large detectors -- difficult to get working in SBND and DUNE, so different approaches currently used.

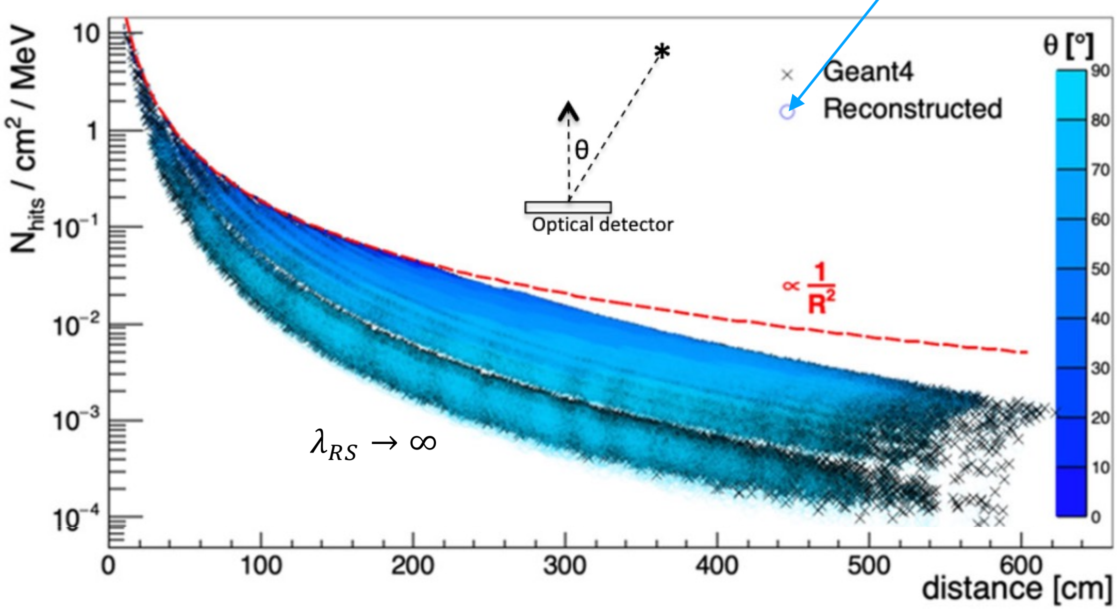
Fast optical model: Semi-Analytical



$$\Omega = h \int_0^{2\pi} \int_0^{b} \frac{r}{[h^2+r^2+d^2-2rd\,\mathrm{Cos}(\varphi_o-\varphi)]^{3/2}} dr d\varphi$$

- Given a dEdx in a point (x, y, z) we want to predict the number of hits in our optical detector (x_i, y_i, z_i)
- Isotropic scintillation emission makes the problem **"almost"** geometric

$$N_\Omega = e^{-\frac{d}{\lambda_{abs}}} \Delta E \cdot S_\gamma(\mathscr{E}) \frac{\Omega}{4\pi}$$



- "Almost" because we have Rayleigh scattering
- Need to correct for it, via parameterization
- Approach used in SBND, and some cases in DUNE

Eur.Phys.J.C 81 (2021) 4, 349

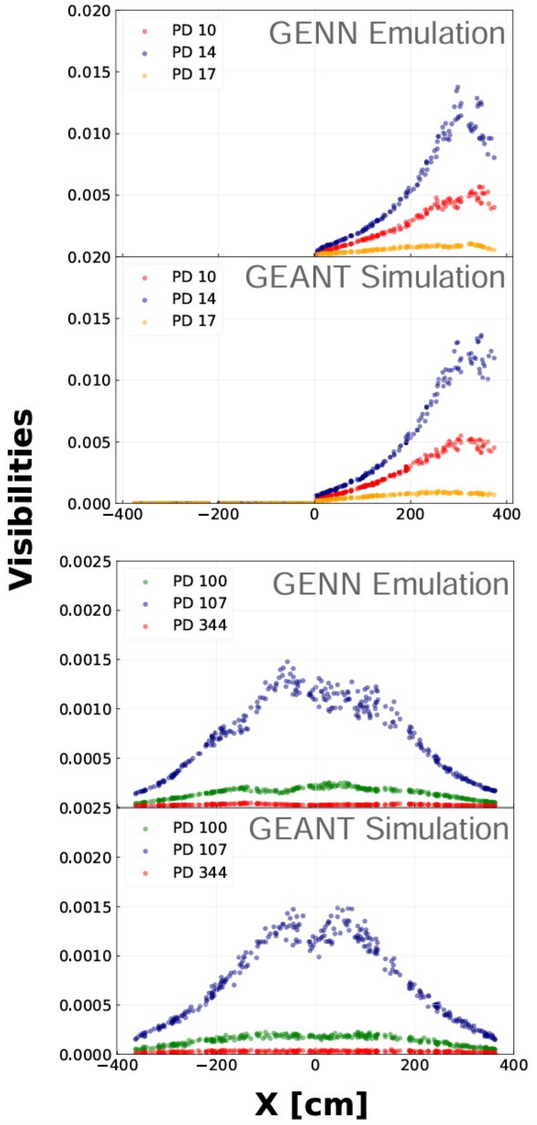
Fast optical model: Neural Networks

Generative neutral network (GENN):

- network trained on Geant4 simulation of voxelized detector, similar to library
- rather than storing visibilities in look-up table, use network to predict at run-time
- avoids memory limitation of library, at some CPU cost; currently used in DUNE

Some ongoing attempts to train similar networks using data rather than simulation:

- no longer have to rely on often poorly known detector properties → could be more accurate / closer to data
- but challenging to get well understood training samples



Mach.Learn.Sci.Tech. 3 (2022) 1, 015033

Full Optical Sim vs FastSim knobs

	Full Optical Sim	Fast Optical
Timing Constants	Tunable	Tunable
Energy Spectrum	Tunable	Tunable (although affects transport)
Ionization/Scintillation Yield	Tunable (handwavy implemented)	Tunable (handwavy implemented)
Rayleigh Scattering	Tunable	“Burned in”
Timing Parametrization	Not needed	“Burned in”/but separate
Material Properties	Tunable	“Burned In”
OnePhoton vs LitePhotons	chooseable	chooseable

- This table is for reference – we'll come back to it later.

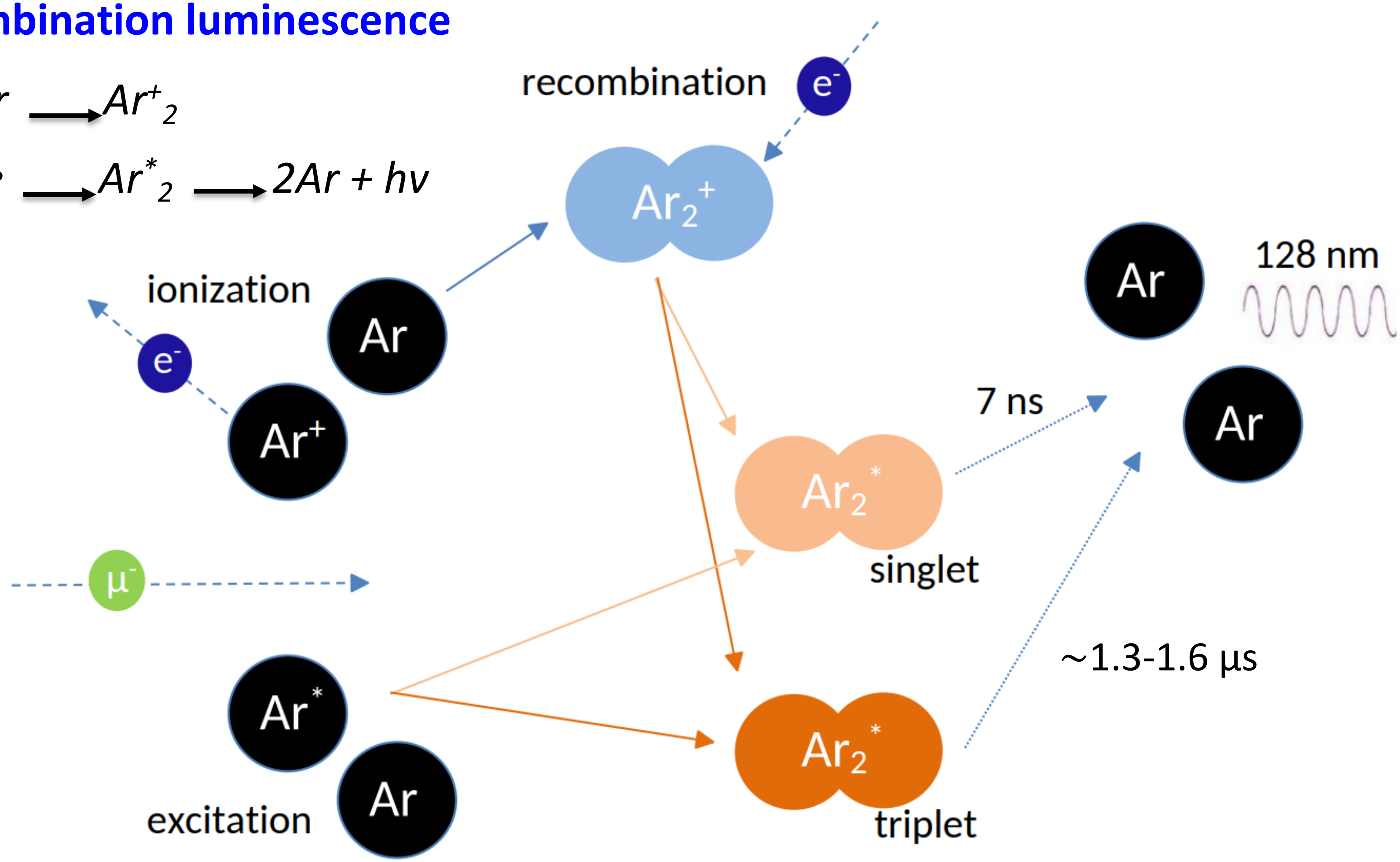
Emission

Scintillation mechanism in LAr

- **Recombination luminescence**

$Ar^{+} + Ar \longrightarrow Ar^{+}_{2}$

$Ar^{+}_{2} + e \longrightarrow Ar^{*}_{2} \longrightarrow 2Ar + hv$



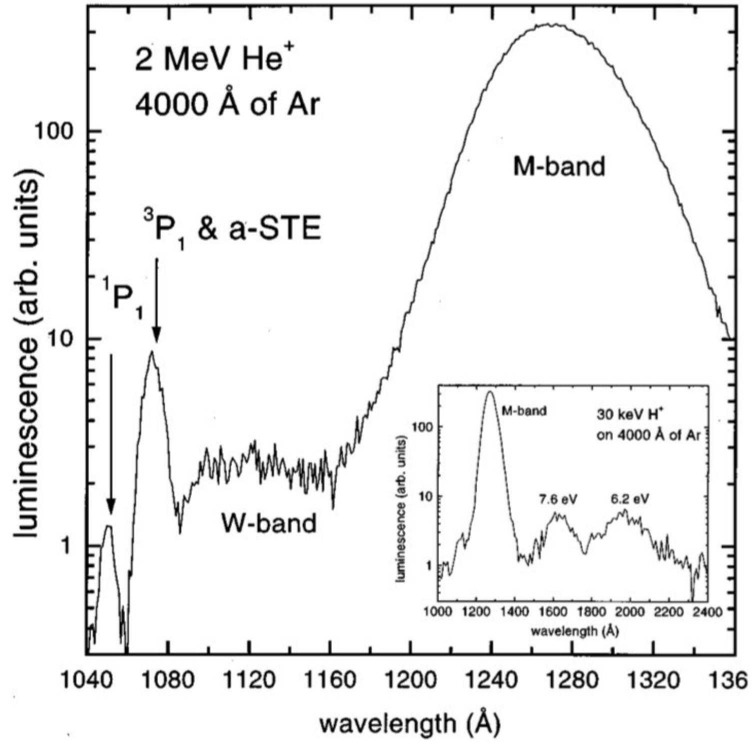
- **Self-trapped excitation luminescence**

$Ar^{*} + Ar \longrightarrow Ar^{*}_{2} \longrightarrow 2Ar + hv$

Scintillation wavelength in LAr

In liquid argon, the overall emission spectrum is well represented by a gaussian shape peaking around:
λ = 128 nm (FWHM $\simeq$ 6 nm)

In LArSoft this is parameterised in larproperties.fcl



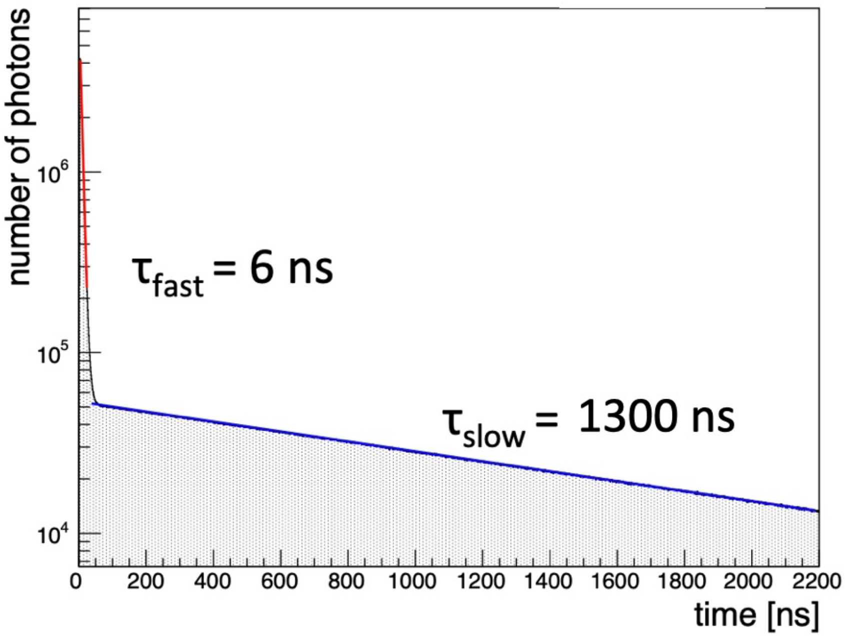
Ph. Rev. B 56 (1997), 6975

lardataalg / lardataalg / DetectorInfo / **larproperties.fcl**

```
# Fast and slow scintillation emission spectra, from [J Chem Phys vol 91 (1989) 1469]
FastScintEnergies:    [ 7.2,  7.9,  8.3,  8.6,  8.9,  9.1,  9.3,  9.6,  9.7,  9.8,  10,  10.2,  10.3,  1
SlowScintEnergies:    [ 7.2,  7.9,  8.3,  8.6,  8.9,  9.1,  9.3,  9.6,  9.7,  9.8,  10,  10.2,  10.3,  1
FastScintSpectrum:    [ 0.0,  0.04, 0.12, 0.27, 0.44, 0.62, 0.80, 0.91, 0.92, 0.85, 0.70, 0.50, 0.31, 0.
SlowScintSpectrum:    [ 0.0,  0.04, 0.12, 0.27, 0.44, 0.62, 0.80, 0.91, 0.92, 0.85, 0.70, 0.50, 0.31, 0.
```

Scintillation signal shape in LAr

- In all measurements the overall scintillation light emission exhibits a double exponential behavior in time
- This is a result of Ar excimer decays characterized by two very different components:
  a *fast component*, $\tau_S \approx$ 6ns,
  and a slow component, $\tau_T \approx$ 1.3μs



- Implementation in LArSoft:

lardataalg / lardataalg / DetectorInfo / larproperties.fcl

```
ScintFastTimeConst:    6.      # fast scintillation time constant (ns)
ScintSlowTimeConst:    1590.*  # slow scintillation time constant (ns)
```

sbndcode/sbndcode/LArSoftConfigurations/opticalproperties_sbnd.fcl

```
# Updating the triplet decay-time (Phys. Rev. C 91, 035503). Note that in our simulations we account
# independently for the TPB-delay time and the emission (fast and slow) decay times.
ScintSlowTimeConst:   1300.  # slow scintillation time constant (ns)
```

*Note: a slow time constant value convolved with the WLS-delay, results in a larger value ≈ 1.5-1.6μs

Scintillation Yields: E-field

Liquid Argon is a prolific scintillator: ~40000 photons/MeV @ zero electric field

Strength of the electric field applied to the LAr impacts the amount of recombination
→ alters amount of charge (Q) and light (L)

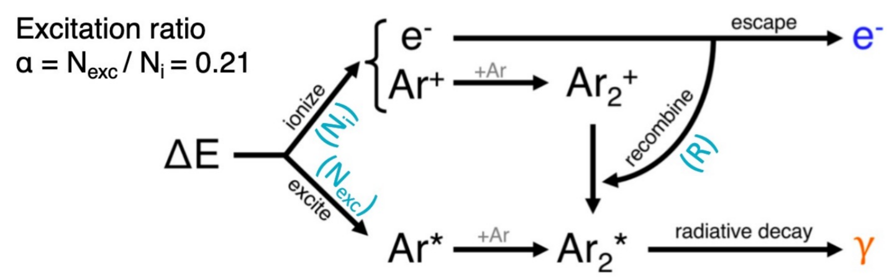
Effect is (anti-)correlated, as electric field increases Q grows, L decreases. At 500 V/cm, energy deposit about equally divided between Q and L

This is modelled in LArSoft ISCalcCorrelated, and enabled in SBND:

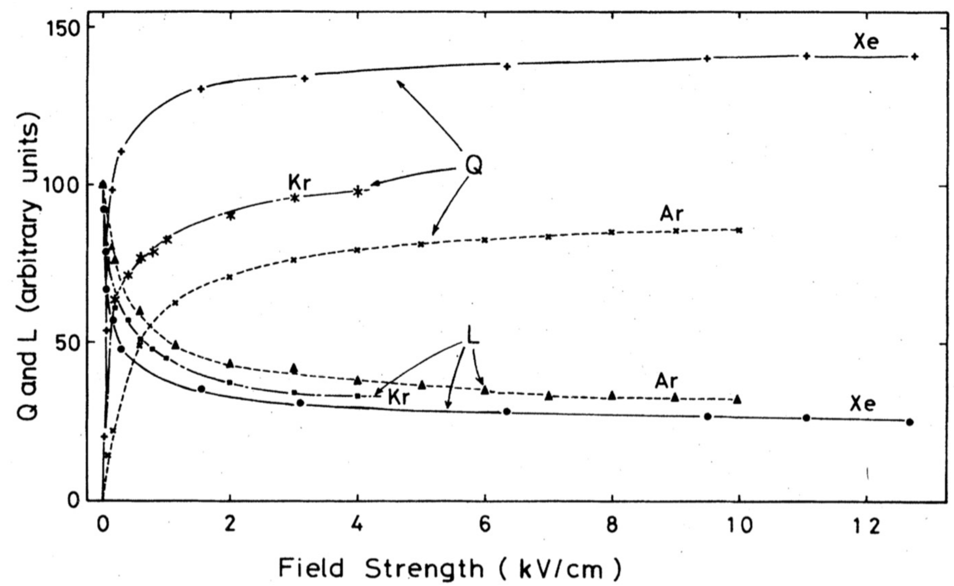
larsim / larsim / IonizationScintillation / **ISCalcCorrelated.cxx**

```
// using this recombination, calculate number of ionization electrons
double const num_electrons = (energy_deposit / fWion) * recomb;

// calculate scintillation photons
double const num_photons = (Nq - num_electrons) * fScintPreScale;
```

Services.LArG4Parameters.IonAndScintCalculator: “Correlated”



PHYSICAL REVIEW D **101,** 012010 (2020)



Phys. Rev. B 20, 3486

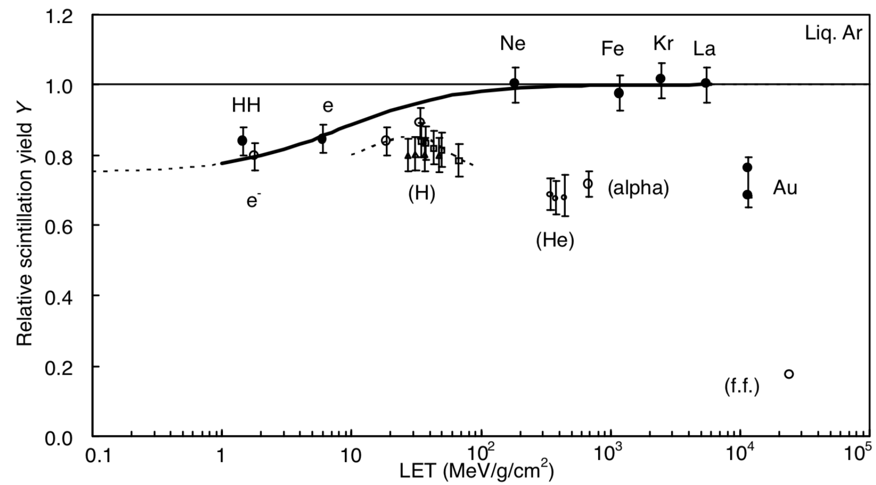
Scintillation Yields: Particle Type

Light yield and fast/slow ratio depend on how ionising the particles are:

- electrons: ~20000 photons/MeV, ~ 30% prompt light
- alphas: ~16800 photons/MeV, ~ 60% prompt light

In LArSoft this is configured in larproperties.fcl, ScintByParticleType:

```
ScintByParticleType: true

# Scintillation yields and fast/slow ratios
MuonScintYield:             24000
MuonScintYieldRatio:        0.23
PionScintYield:             24000
PionScintYieldRatio:        0.23
ElectronScintYield:         20000
ElectronScintYieldRatio:    0.27
KaonScintYield:             24000
KaonScintYieldRatio:        0.23
ProtonScintYield:           19200
ProtonScintYieldRatio:      0.29
AlphaScintYield:            16800
AlphaScintYieldRatio:       0.56
```



Particle	τ_S	τ_T	I_S/I_T	Reference
Electron	6.3 ±0.2	1020±60	0.083	Kubota *et al.*[a]
	(5.0 ±0.2)	(860±30)	(0.045)	(E = 6 kV/cm)[a]
	4.6	1540	0.26	Carvalho and Klein[b]
	4.18±0.2	1000±95		Keto *et al.*[c]
		1110±50		Suemoto and Kanzaki[d]
	6 ±2	1590±100	0.3	This work
α	~5	1200±100		Kubota *et al.*[e]
	4.4	1100	3.3	Carvalho and Klein[b]
	7.1±1.0	1660±100	1.3	This work
F.F.	6.8±1.0	1550±100	3	This work

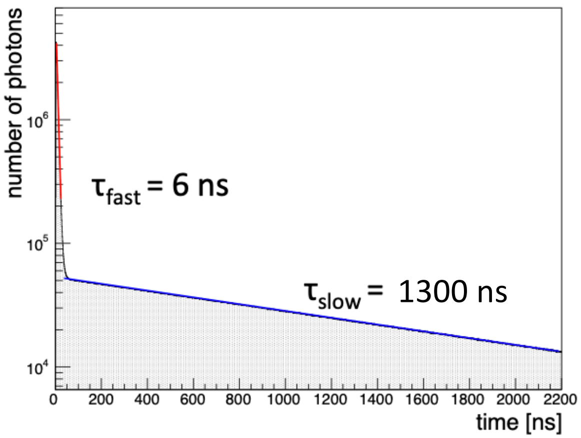
Jpn. J. Appl. Phys. Vol. 41 (2002) pp. 1538–1545;
Ph. Rev. B 27 (1983), 5279

Propagation

Scintillation light propagation

Scintillation (emission):

$0.3 \times \tau_{fast}(6\ ns) + 0.7 \times \tau_{slow}\ (1300\ ns)$



$$Q = N_e = N_i R,$$
$$L = N_\gamma = N_{ex} + N_i(1 - R),$$

$$Q + L = N_{ex} + N_i = \frac{\Delta E}{W_{ph}}$$

We need how to get our number of detected photons and their arrival times ⇒ **Transport effects**

- Scintillation photons have energy lower than the first excited state of the Ar atom, therefore pure LAr is transparent to its own scintillation radiation
- However, during propagation through LAr VUV photons may undergo elastic interactions on Ar atoms ⇒ Rayleigh scattering
- **Rayleigh Scattering affects, in a non negligible way, the light signals in our detectors in comparison with the "pure" emitted scintillation light**
- It is important to understand/model it properly in liquid argon

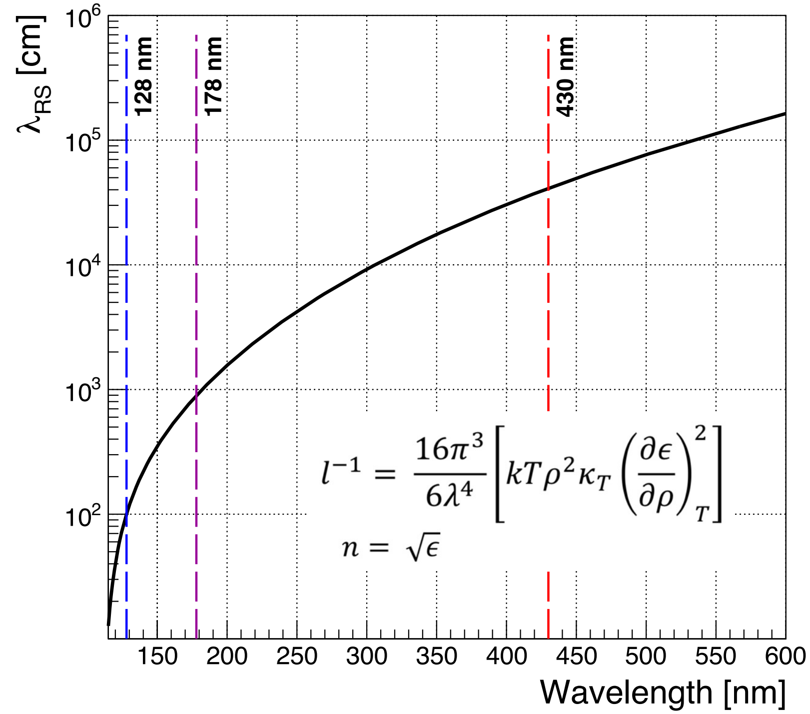
Rayleigh Scattering in Argon

Rayleigh scattering length for VUV photons in Ar measured variously between ~50 – 120 cm

- very hard to measure: small uncertainties in the index of refraction can drastically change the scattering length λ_{RS}
- most recent measurement around 100 cm, adopted by many LArTPC experiments

RS ~100cm < typical size of LArTPC detectors
→ has significant impact on light seen

In LArSoft, parameterized in larproperties.fcl



lardataalg / lardataalg / DetectorInfo / larproperties.fcl

```
# Refractive index as a function of energy (eV) from arXiv:2002.09346
RIndexEnergies: [ 1.18626, 1.68626, 2.18626, 2.68626, 3.18626, 3.68626, 4.18626, 4.68626, 5.18626,
RIndexSpectrum: [ 1.24664, 1.2205, 1.22694, 1.22932, 1.23124, 1.23322, 1.23545, 1.23806, 1.24116, 1

RayleighEnergies: [ 1.18626, 1.68626, 2.18626, 2.68626, 3.18626, 3.68626, 4.18626, 4.68626, 5.18626,
RayleighSpectrum: [ 1200800, 390747, 128633, 54969.1, 27191.8, 14853.7, 8716.9, 5397.42, 3481.37, 23
```

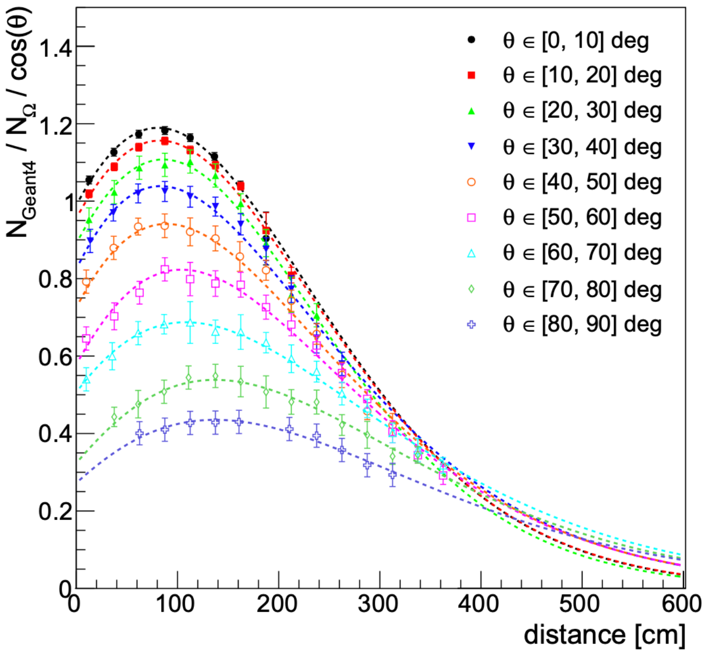
Modelling in fast optical simulation

Optical library / GENN network:

- Encoded in *visibilities* in each voxel directly from full optical simulation

Semi-analytical model:

- Treated as a correction to the geometric prediction
- Parameterised based on difference between geometric prediction and full optical simulation
- Also correct for border effects in analogous way (reflections/absorption)



$$N_{\gamma} = N_{\Omega} \times GH'(d, \theta, d_T)/cos(\theta)$$

Eur. Phys. J. C (2021) 81:349

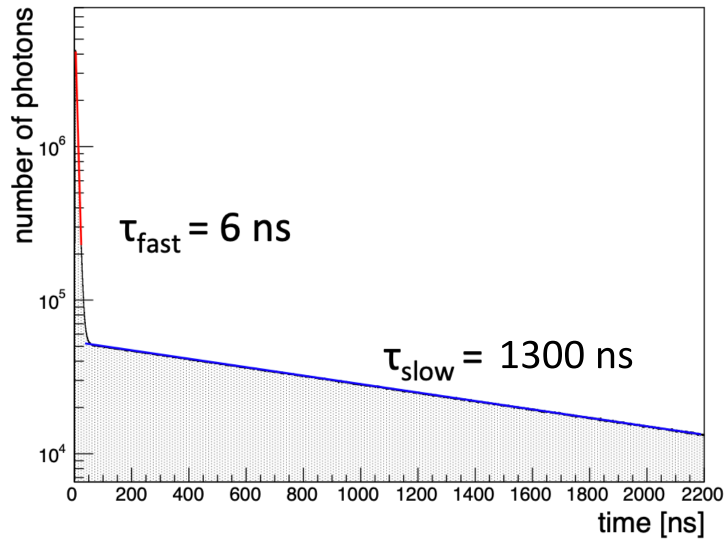
In LArSoft models implemented in larsim:

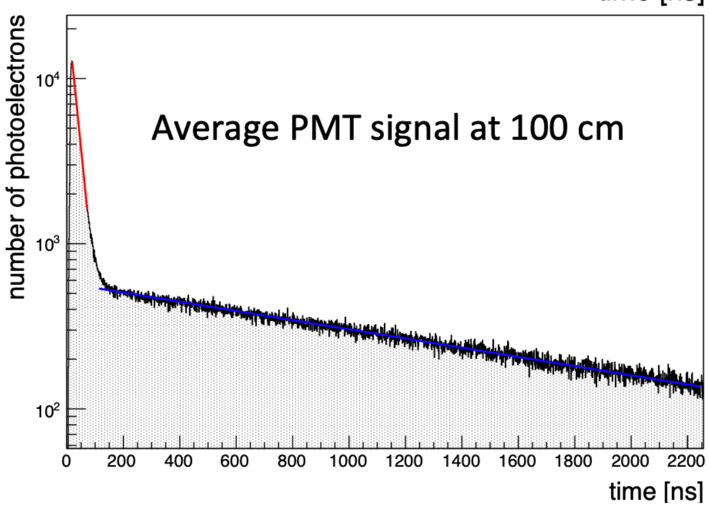
larsim/PhotonPropagation/PDFastSimPVS_module.cc (Library)
larsim/PhotonPropagation/PDFastSimPAR_module.cc (Semi-analytical)

Time structure of detected signals

Scintillation (emission):

$0.3 \times \tau_{fast}(6\ \text{ns}) + 0.7 \times \tau_{slow}(1300\ \text{ns})$

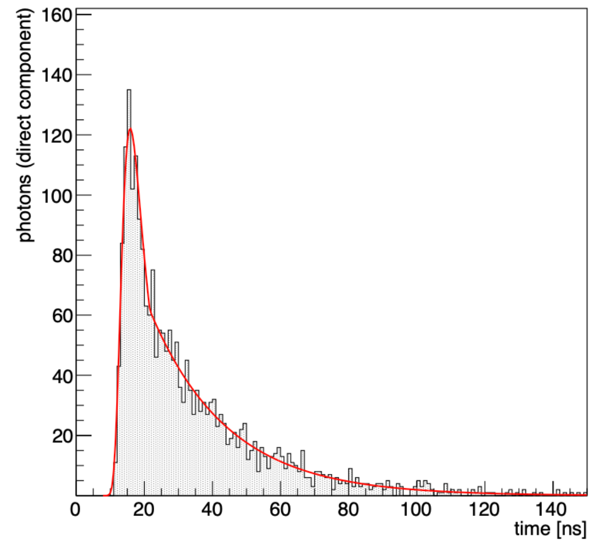




+

Propagation:

Direct transportation + Rayleigh Scattering



=

In “large” detectors transport effects will affect the effective time structure of the detected scintillation light

$$t_\gamma = t_E + t_t(d,\theta) + t_{WLS} + t_{det},$$

$t_E = \text{emission time}$

$t_t = \text{transport time}$

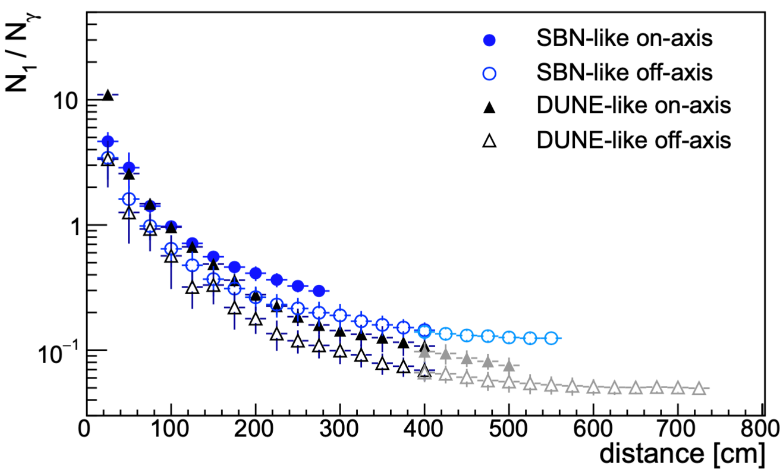
$t_{WLS} = \text{WLS delay time}$

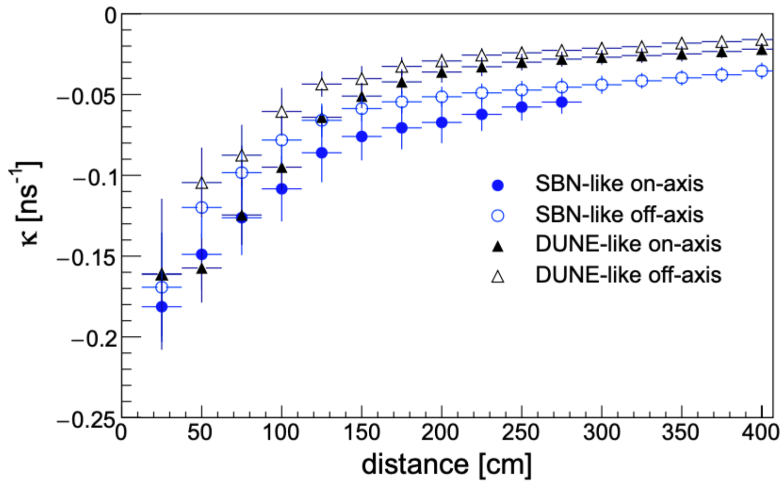
$t_{det} = \text{detector time}$

Time structure of detected signals

In fast optical simulation, modelled using parameterisations of Landau + Exponential fits to distributions from full optical simulation

- developed in conjunction with semi-analytical model, but can be used in combination with any approach to get number of photons (library, etc.)





$$t_t(x) = \underbrace{N_1 \frac{1}{\xi} \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} e^{\lambda s + s \log s}\, ds}_{Landau} + \underbrace{N_2\, e^{\kappa\, x}}_{Exponential} ,$$

Eur. Phys. J. C (2021) 81:349

larsim / larsim / PhotonPropagation / PropagationTimeModel.h

larsim / larsim / PhotonPropagation / PDFastSimPAR.fcl

```
IncludePropTime:        true
```

Implementation in SBND

Detector specific

Hybrid approach used in SBND:

- semi-analytical model (hits + timing) inside TPC
- slimmed-down optical library outside TPC

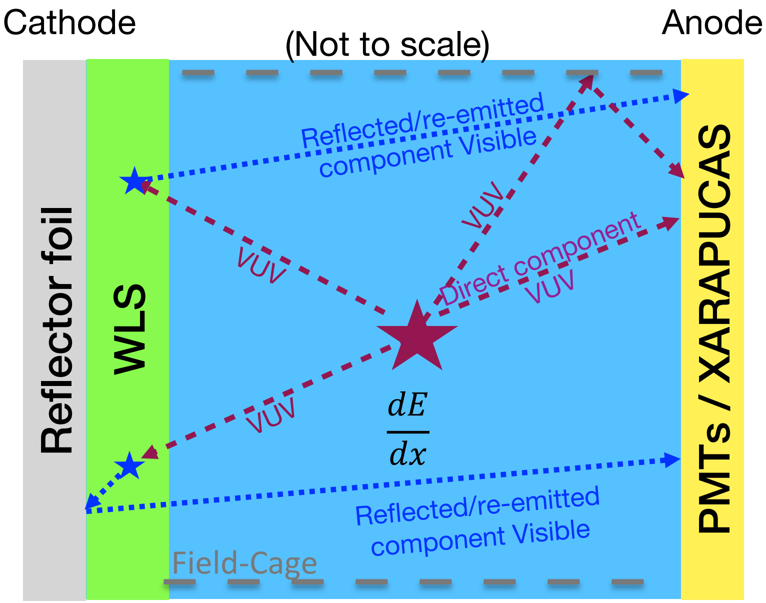
Configuration in SBND:

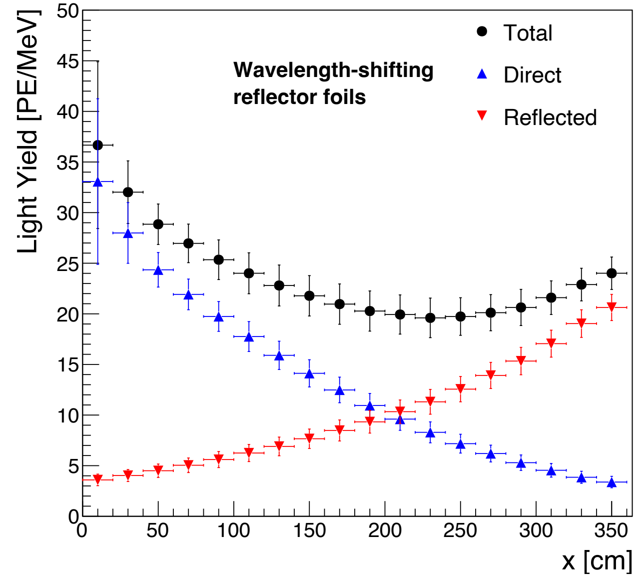
sbndcode/sbndcode/LarSoftConfigurations/opticalsimparameterisations_sbnd.fcl

```
# *********************************************************************************
#          PARAMETERS SETS FOR SEMI-ANALYTIC SIMULATION ARE DEFINED HERE
# *********************************************************************************

# VUV/DIRECT LIGHT: TIMING PARAMETERISATION

# VUV/DIRECT LIGHT: NUMBER OF HITS CORRECTIONS

# *********************************************************************************
#  VIS semi-analytic model, specific to SBND
# *********************************************************************************

# VISIBLE/REFLECTED LIGHT: TIMING PARAMETERISATION

# VISIBLE/REFLECTED LIGHT: NUMBER OF HITS CORRECTIONS
```

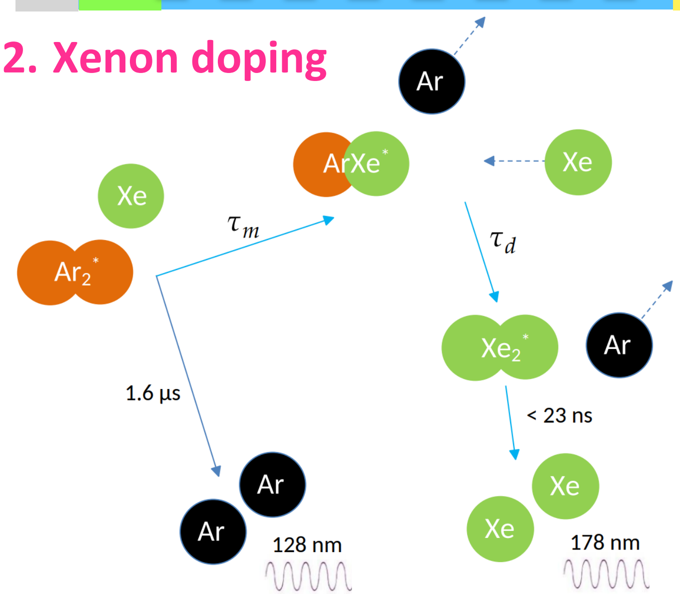
(Aside): Enhancing the Light Yield in LArTPCs

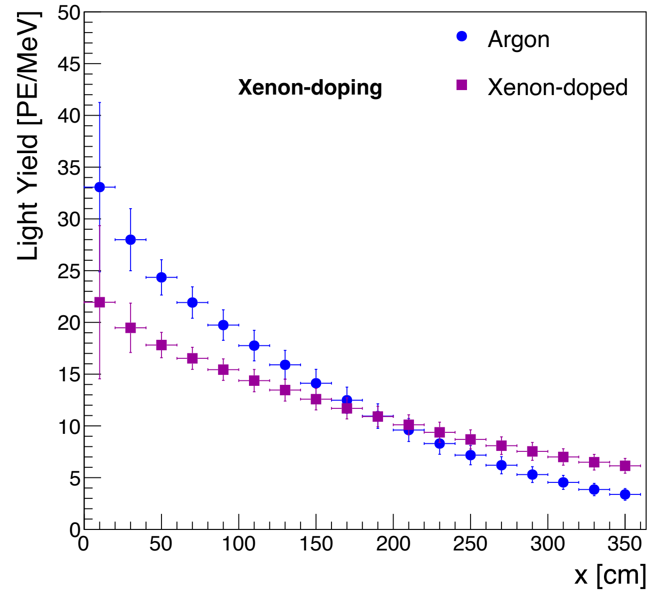
1. WLS-Coated reflector foils





2. Xenon doping

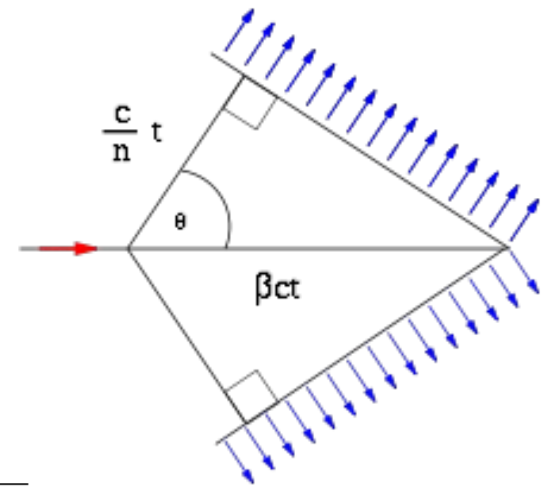




(Aside): Cherenkov radiation in LAr

- A particle propagating in a medium with velocity greater than that of light in the medium produces an electromagnetic shock-wave with conic wavefront

- Photons are emitted with a precise angle with respect to particle direction



NIM A 516 (2004) 348–363

$$\frac{d^2N}{d\nu dx} = \frac{2\pi\alpha}{c}\sin^2\theta_{\check{C}} \qquad \cos\theta_{\check{C}} = \frac{1}{\beta \cdot n_{Ar}(\lambda)}$$

$$\Rightarrow \int_{109nm\,(LAr\ absorbed)}^{600nm\,(hard\ to\ detect)} \qquad R_{\check{C}} = \frac{dN_{\check{C}}/dx}{dN_{scint}/dx + dN_{\check{C}}/dx} = 2.4\%$$

$\Rightarrow$

→ Can be considered a second order effect with respect to scintillation light emission

sbndcode/sbndcode/LArSoftConfigurations/opticalproperties_sbnd.fcl

```
EnableCerenkovLight: false # Cerenkov light OFF by default
```

Beware enabling: no fast optical simulation exists, will use very slow full simulation!

Detection

Detecting light in LArTPCs

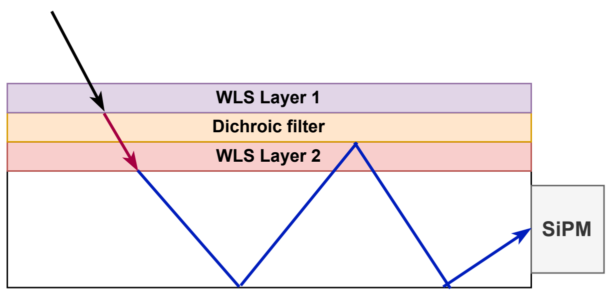
VUV LAr scintillation light is hard to detect directly, absorbed by most materials

- need to make use of wavelength shifters, most commonly TPB

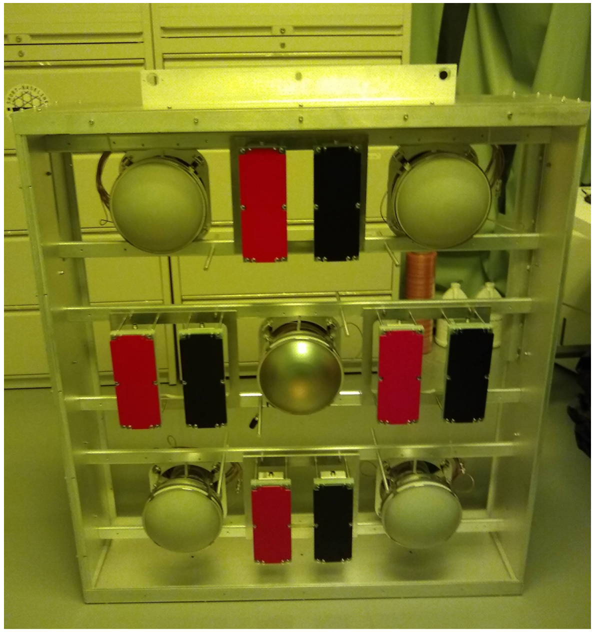
Photon detectors used:

- PMTs coated with WLS (SBND) or with WLS-coated plates in-front of them (MicroBooNE)
- Arapuca/XArapuca wavelength-shifting light traps using SiPMs (DUNE, SBND)

Wavelength shifters emit ~isotropically, lose 50% of light emitted away from photon-detectors



Arapuca operational principle

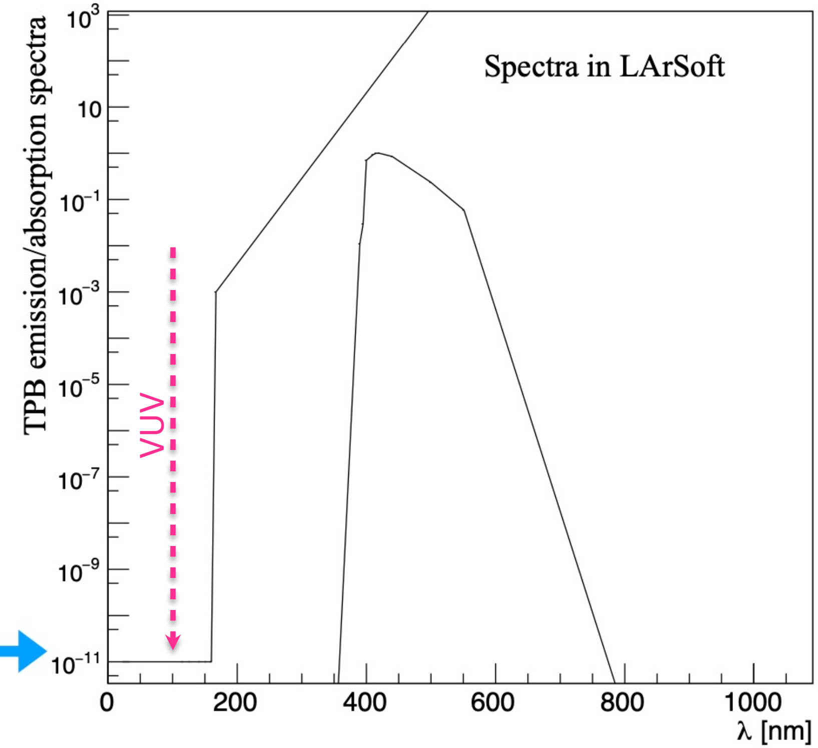


Photon detection system module in SBND, mixture of PMTs and XArapucas

Wavelength shifter in LArSoft

Processes handled by `G4OpWLS`:

- Initial photon killed and a new one created with different wavelength
- User must provide:
  - Absorption length as function of photon energy
  - Emission spectrum as function of photon energy
  - Time delay between absorption and re-emission



The **WLSABSLENGTH** defines the absorption length which is the average distance travelled by a photon before it is absorbed by the TPB.

lardataalg / lardataalg / DetectorInfo / **larproperties.fcl**

```
# WLS - TPB properties original tpb [0.0, 0.0, 0.0, 0.0588,0.235, 0.853, 1.0,1.0,0.9259,0.704
TpbEmmisionEnergies: [0.05,1.0,1.5, 2.25, 2.481, 2.819, 2.952,2.988,3.024, 3.1, 3.14,3.1807,
TpbEmmisionSpectrum: [0.0, 0.0, 0.0, 0.0588,0.235, 0.853, 1.0,1.0,0.9259,0.704,0.0296,0.011,
TpbAbsorptionEnergies: [0.05,1.77,2.0675, 7.42, 7.75, 8.16, 8.73, 9.78,10.69, 50.39]
TpbAbsorptionSpectrum: [100000.0,100000.0, 100000.0,0.001,0.00000000001,0.00000000001, 0.000
```

Wavelength shifter time delay

TPB has complex time structure:

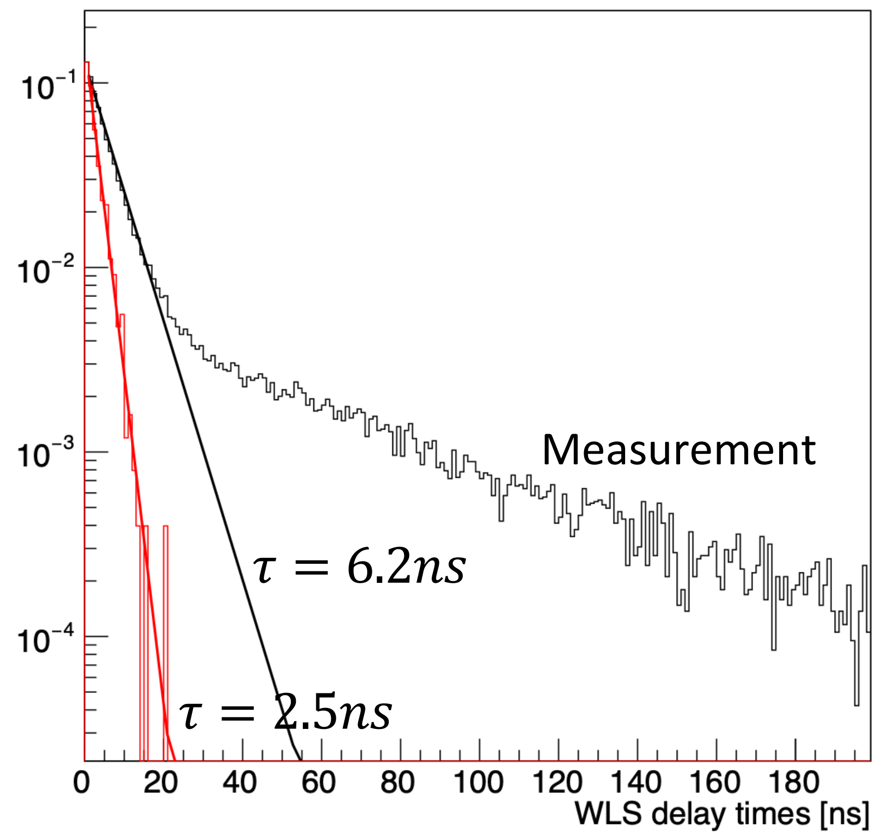
- bulk of light emitted promptly, but non-negligible longer components
- would non-negligibly alter time distribution

Geant4 (G4OpWLS class) only simulates Delta or Exponential model (neither is the case)

Instead, we simulate this separately in LArSoft:

- in SBND this is done in the at the optical detector digitizer stage (slightly hacky)
- sbndcode/sbndcode/OpDetSim/DigiPMTSBNDAlg.cc

	Decay time (ns)	Abundance (%)
Instantaneous component	1–10	60 ± 1
Intermediate component	49 ± 1	30 ± 1
Long component	3550 ± 500	8 ± 1
Spurious component	309 ± 10	2 ± 1



PHYSICAL REVIEW C **91**, 035503 (2015)

Photon simulation output objects

lardataobj / lardataobj / Simulation / SimPhotons.h

```
// This structure contains all the information per photon
 // which entered the sensitive OpDet volume.

 class OnePhoton
 {
 public:
   OnePhoton();

   bool          SetInSD;
   TVector3      InitialPosition;
   TVector3      FinalLocalPosition; // in cm
   float         Time;
   float         Energy;
   int           MotherTrackID;
 };
```

```
class SimPhotons : public std::vector<OnePhoton>
```

```
class SimPhotonsLite
{
  public:
    SimPhotonsLite();
    SimPhotonsLite(int chan)
      : OpChannel(chan)
    {}

    int   OpChannel;
    std::map<int, int> DetectedPhotons;

    SimPhotonsLite& operator+=(const SimPhotonsLite &rhs);
    const SimPhotonsLite operator+(const SimPhotonsLite &rhs) const;

    bool operator==(const SimPhotonsLite &other) const;
};

// Define a OpDet Hit as a list of OpDet photons which were
// recorded in the OpDet volume.

class SimPhotons : public std::vector<OnePhoton>
```

- `SimPhotons` objects (collections of OnePhoton) save detailed information about each detected photon

- `SimPhotonsLite` objects reduce memory and size at the price of keeping only the number of photons at a time-slot.

- The kind of object you want to save in your simulation is specified in the configuration file by the line:

```
services.LArG4Parameters.UseLitePhotons: true  # false to save SimPhotons
```

Full Optical Sim vs FastSim knobs

	Full Optical Sim	Fast Optical
Timing Constants	Tunable	Tunable
Energy Spectrum	Tunable	Tunable (although affects transport)
Ionization/Scintillation Yield	Tunable (handwavy implemented)	Tunable (handwavy implemented)
Rayleigh Scattering	Tunable	“Burned in”
Timing Parametrization	Not needed	“Burned in”/but separate
Material Properties	Tunable	“Burned In”
OnePhoton vs LitePhotons	chooseable	chooseable

- Hopefully should make more sense now.

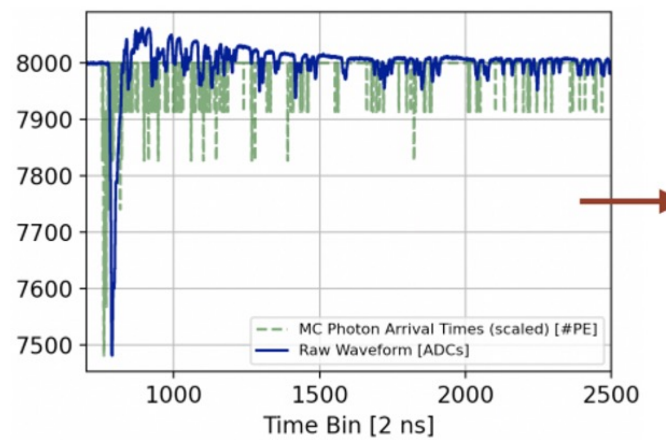
PMT digitisation (SBND example)

Finally, we simulate the resulting waveforms we'd see on the photo-detectors:

- each PE gets swapped for an electronics response, either constructed from parameters (idealised) or from a measured response
- the expected noise is also added to the waveforms

sbndcode / sbndcode / OpDetSim / digi_pmt_sbnd.fcl

```
# Parameters for ideal SER simulation
PMTRiseTime:            3.8        #ns
PMTFallTime:            13.7       #ns
PMTMeanAmplitude:       0.9        #in pC
TransitTime:            55.1       #ns
PMTChargeToADC:         -25.97     #charge to adc factor

# Parameters for test bench SER simulation
PMTSinglePEmodel:       true      #false for ideal PMT respons
PMTDataFile:            "OpDetSim/digi_pmt_sbnd_v2int0.root"
```



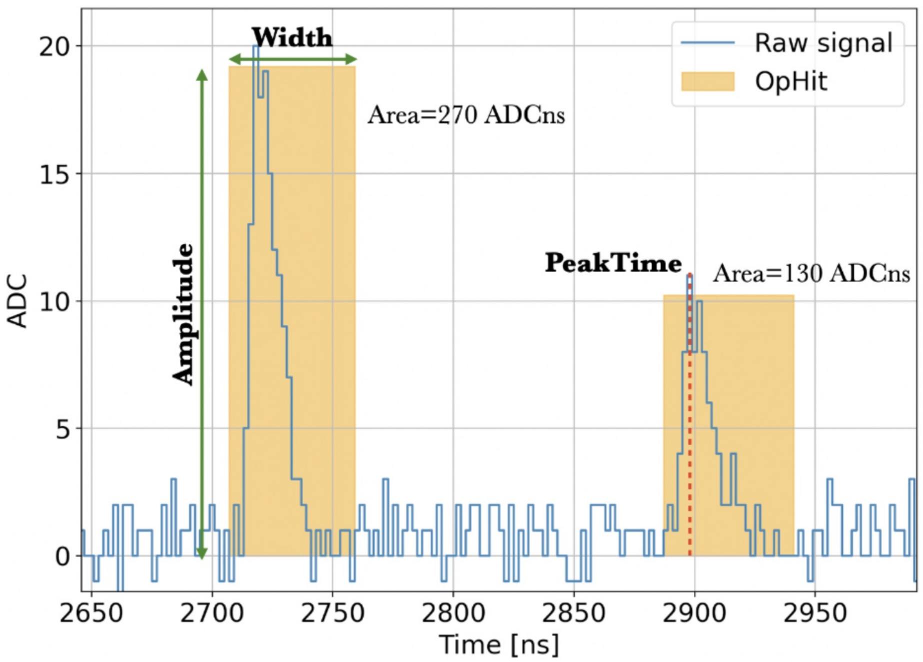
Example PMT waveform in SBND compared with MC photon arrival times:

- undershoot due to AC coupling → bipolar single electron response

Reconstruction

Optical Hits

- First, we look for pulses raw (or deconvolved) waveforms
- The light pulses in LArSoft are stored in objects called OpHits
- OpHits are found when the waveform goes above a certain threshold and are held while it continues to be so
- The OpHit Time is decided by the first arriving photon
- This can lead to the merging of visible separate optical signals, especially in the case of SiPMs (in the Arapucas)



lardataobj / lardataobj / RecoBase / OpHit.h

```
int             fOpChannel;
unsigned short  fFrame;
double          fPeakTime;
double          fPeakTimeAbs;
double          fWidth;
double          fArea;
double          fAmplitude;
double          fPE;
double          fFastToTotal;
```

Optical Flashes

Optical hits from different photon detectors that are time-coincident are combined into Optical Flashes:

- these are analogous to clusters in the charge reconstruction, but matched in time rather than space

Having a flash allows us to reconstruct the position of the particles that generated the light (roughly)

This can then be used to match the light signals to the reconstructed TPC tracks -- Flash Matching

lardataobj / lardataobj / RecoBase / OpFlash.h

```
private:

    double                fTime { 0.0 }; ///< Time on @ref DetectorClocksHardware
    double                fTimeWidth;    ///< Width of the flash in time [us]
    double                fAbsTime;      ///< Time by PMT readout clock
    unsigned int          fFrame;        ///< Frame number
    std::vector< double > fPEperOpDet;   ///< Number of PE on each PMT
    std::vector< double > fWireCenters;  ///< Geometric center in each view
    std::vector< double > fWireWidths;   ///< Geometric width in each view
    double                fXCenter { NoCenter }; ///< Estimated center in x [cm]
    double                fXWidth { NoCenter }; ///< Estimated width in x [cm]
    double                fYCenter;      ///< Geometric center in y [cm]
    double                fYWidth;       ///< Geometric width in y [cm]
    double                fZCenter;      ///< Geometric center in z [cm]
    double                fZWidth;       ///< Geometric width in z [cm]
    double                fFastToTotal;  ///< Fast to total light ratio
    bool                  fInBeamFrame;  ///< Is this in the beam frame?
    int                   fOnBeamTime;   ///< Is this in time with beam?
```

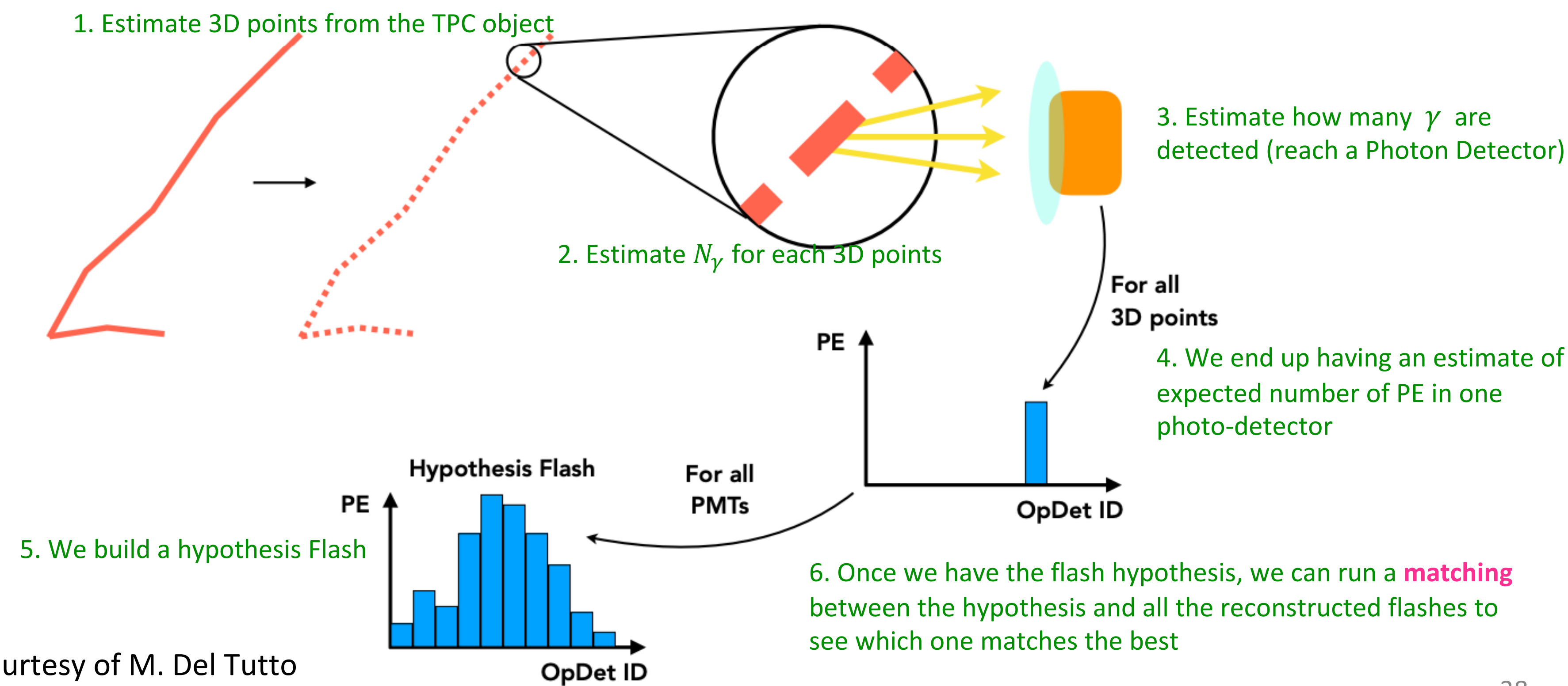
Flash Matching: `OpT0Finder` example

Flash matching goals:

- Distinguish a neutrino interaction from cosmic backgrounds
- Provide T_0 for each TPC interaction

The two ingredients for flash matching:

- Reconstructed Flashes
- TPC Objects (reconstructed objects in the TPC, i.e. Pandora's `recob::Slice`) ⇒ The flash matching code should match a TPC Object with its flash



Courtesy of M. Del Tutto

Summary

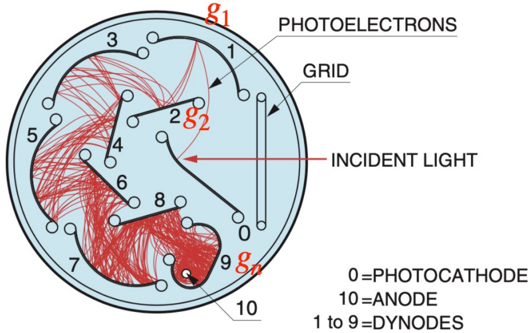
- Optical simulation is tricky, need to cut some corners to get it working in LArSoft with reasonable resource usage (size, number of photons -> Memory, CPU).
- Corners are cut, so there is always room for improvement.
- Applications of scintillation light in LArTPCs are not fully developed – always lots of opportunities to do new things.

Backups

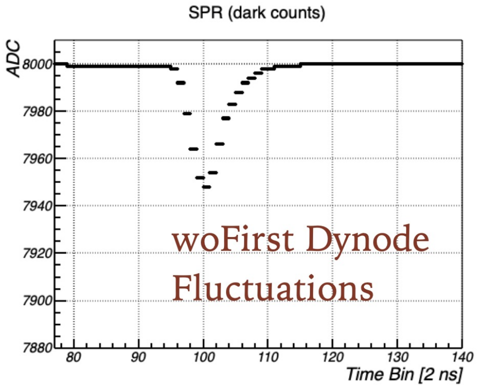
PMT Gain fluctuation

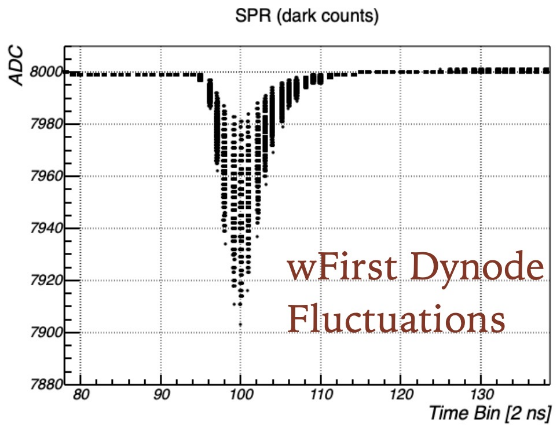
(Slide from F.J. Nicolás)

- Number of secondary electrons generated at each dynode: random variable
- Toy example:
  - Consider 1e hits one of the dynode (with gain g_i)
  - On average $< m >= g_i$ with a standard deviation $\sigma = \sqrt{g_i}$
  - This leads to fluctuations in the SER



- Approximations (approach directly taken from icaruscode) ⇒ Only takes into account fluctuations at first dynode: $\longrightarrow \frac{\sigma_N^2}{< N >^2} = \frac{1}{g_1} + \frac{1}{g_1 g_2} + \ldots \frac{1}{g_1 g_2 \cdots g_n}$
  - <N>: average number of electrons at the end of the multiplication chain (anode)
  - σ_N^2: fluctuations in the total number of electrons at the anode





sbndcode / sbndcode / OpDetSim / PMTAlg / pmtgainfluctuations_config.fcl

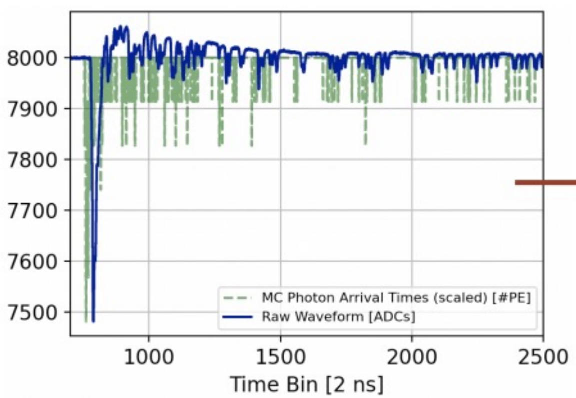
Light Signal Deconvolution

- In SBND we have PMT (and XARAPUCA) readout with **AC coupling**: bipolar SER ⇒ This makes accurate light reconstruction **a challenge**

(by F.J. Nicolás)

➤ OpDeconvolution module (in brief): **sbndcode / sbndcode / OpDetReco / OpDeconvolution /**

Starts with the **raw::OpDetWaveform** objects (from DetSim stage)

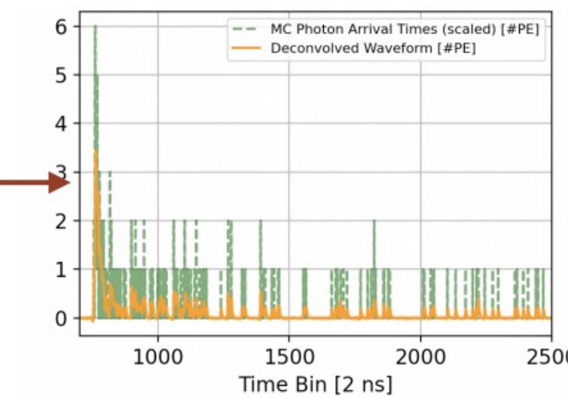


OpDeconvolution module

➤ Methods to:

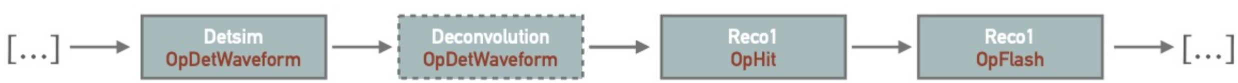
- ➤ Perform deconvolution (using FFT)
- ➤ Baseline estimator
- ➤ Reduce noise (waveforms smoothing and filtering in the frequency domain)

Produces deconvolved signals (also **raw::OpDetWaveform** objects) to be fed to downstream reco algorithms



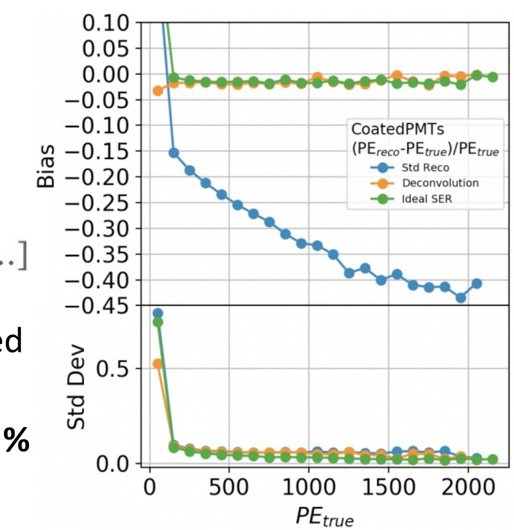
Downstream reconstruction chain:

- Use standard OpHit and OpFlash finder algorithms to recover pulses ⇒ #PE, t0… using the deconvolved signals



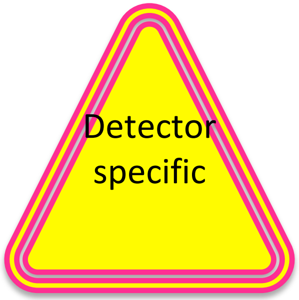
- OpHit and OpFlash configuration file with refined parameters for deconvolved waveforms

⇒ **Performance: resolution better than ~5% and unbiased at the level of few %**

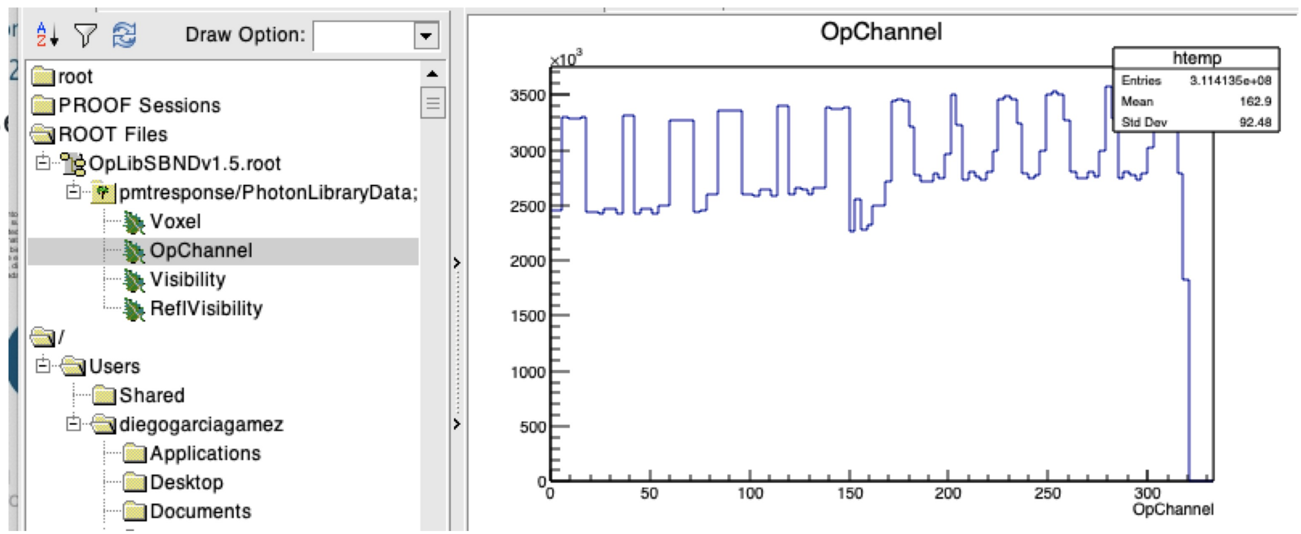


Optical Library parameters: voxelization scheme

sbndcode / sbndcode / LArSoftConfigurations / photpropservices_sbnd.fcl

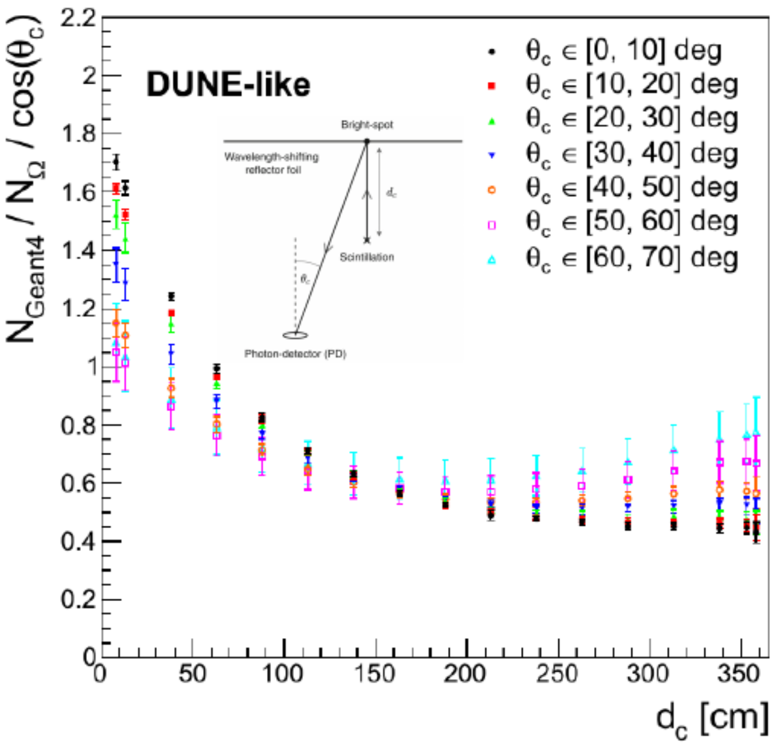


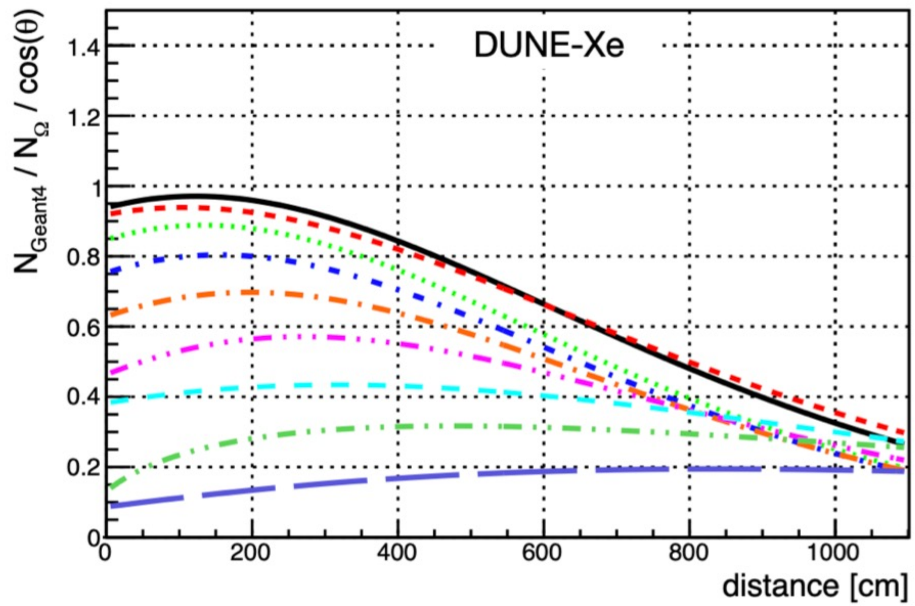

```
# (Re)Defining the Optical Library information/files for the PD-fast HYBRID optical mode
sbnd_library_for_hybrid_mode_photonvisibilityservice:
{
   @table::sbnd_library_vuv_vis_prop_timing_photonvisibilityservice
   LibraryFile: "OpticalLibrary/SBND_OpLibOUT_v2.00.root"
   NX: 66
   NY: 56
   NZ: 71
   UseCryoBoundary: false
   # IF UseCryoBoundary is s
   XMin:  -264
   XMax:   264
   YMin:  -280
   YMax:   280
   ZMin:  -60
   ZMax:   650
}
```



Semi-Analytic model extensions (available)

Eur. Phys. J. C (2021) 81:349





$$N_{\Omega,reflected} = N_{\gamma,direct}(\Omega_c, d_c, \theta_c, d_T) \times Q_r \times \frac{\Omega_{PD}}{2\pi}$$

Number of photons incident on the cathode

Q_{WLS} x Q_{foil}

PD aperture as viewed by the bright spot

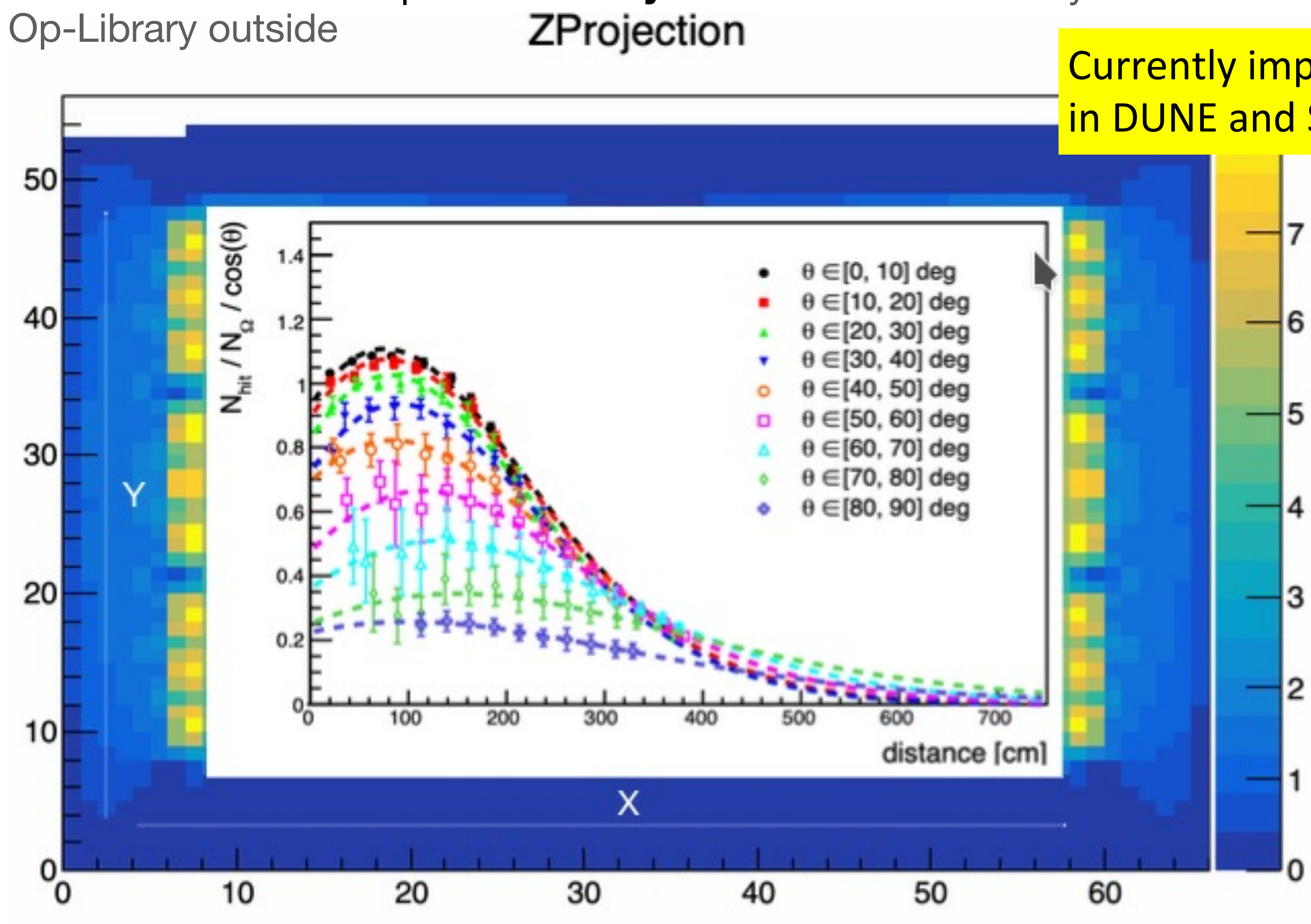
$$N_{\gamma,reflected} = N_{\Omega,reflected} \times A(d_c, \theta_c, d_T)/cos(\theta_c)$$

PD-location + border correction

- LArSoft semi-analytic model simulation can also model these the extensions:
  - LAr and LXe wavelengths (doping)
  - Direct and Reflected light (foils)

Hybrid model for the photon propagation

- Semi-Analytic model has a limitation: only applicable inside the active volume (geometric approach)
- Simple idea to overcome the problem ⇒ **Hybrid model**: Semi-Analytic model inside the TPC + Op-Library outside

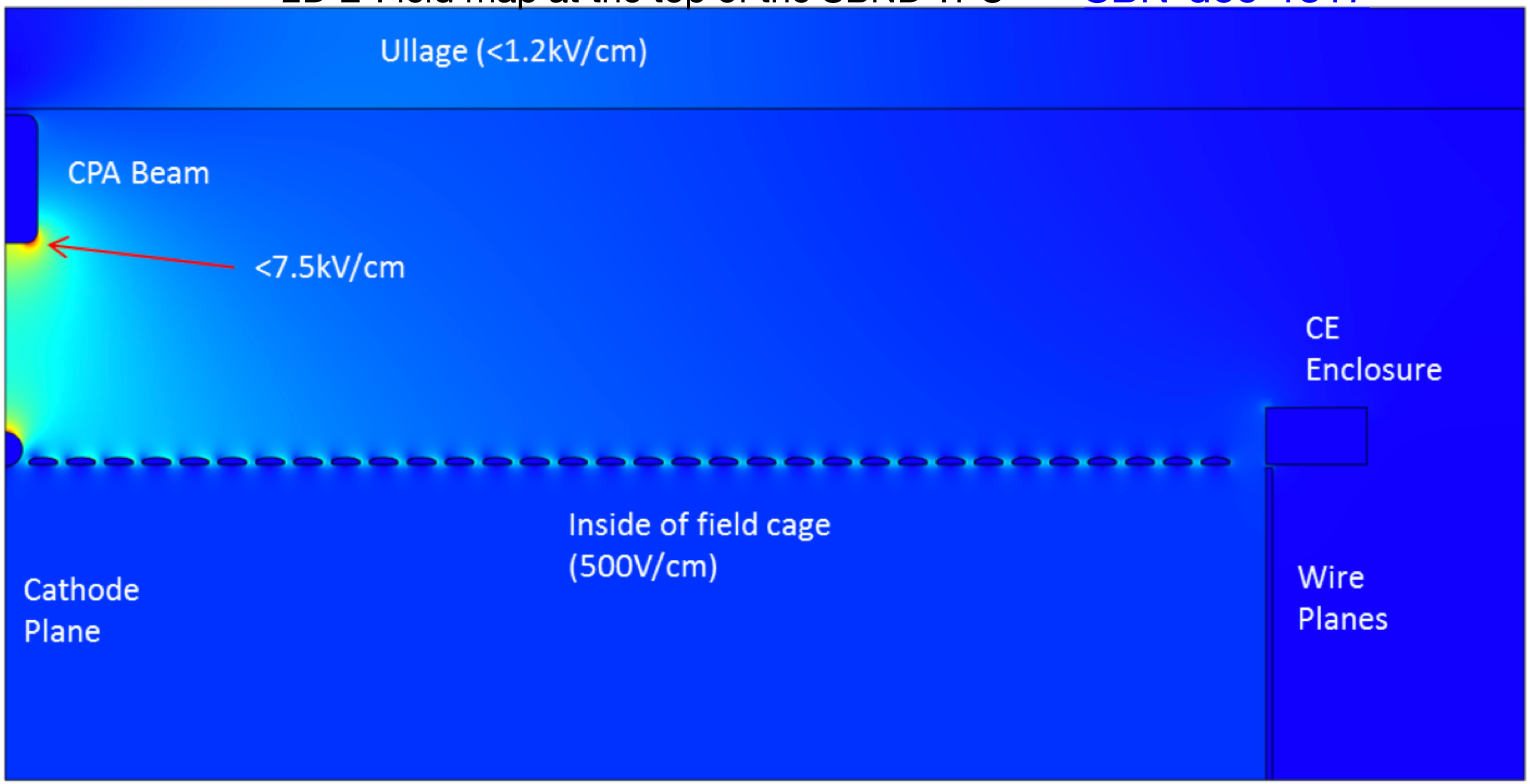


Currently implemented in DUNE and SBND

E-Field map in a TPC (SBND case example)

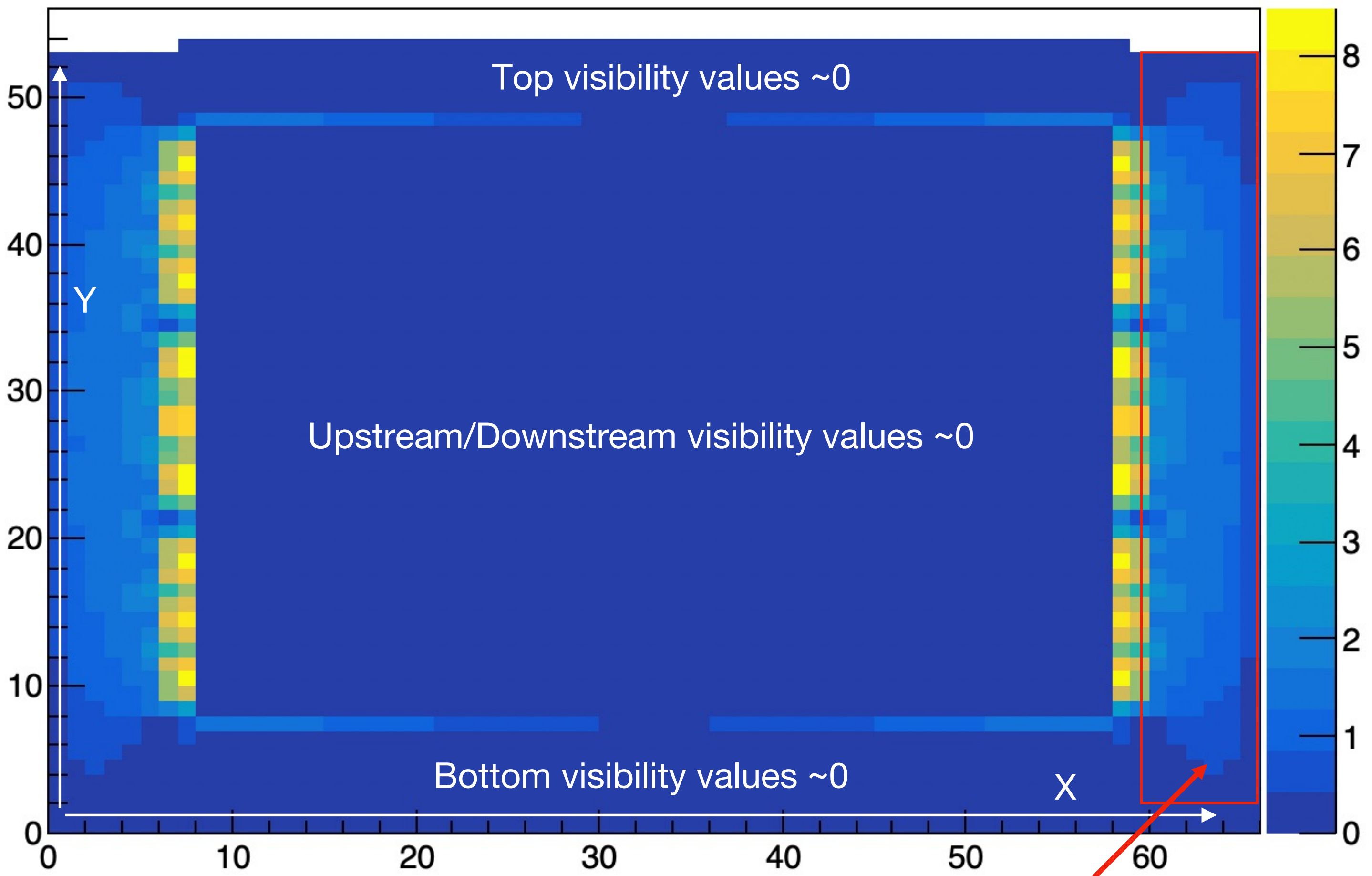
Warning: Light yield strongly depends on the Electric Field value

2D E-Field map at the top of the SBND TPC SBN-doc-1317



- Inside the active volume EF is constant @ 0.5 kV/cm (nominal)
- In the top of the TPC EF values range from few kV/cm at the CPA location decreasing to ~0 at the APA.
- Behind APA (PD-plane) EF = 0 is a good approximation (almost constant)

E-Field x Visibility map in a TPC (SBND case example)



- Only behind APA visibilities are significant
- Current EF model in the hybrid approach: **500V/cm inside the TPC & 0V/cm anywhere else**