Scintillation Light in LArTPCs: Simulation and Reconstruction

Andrzej Szelc

University of Edinburgh

With: Diego Garcia-Gamez and Patrick Green

UK-Latin America School in Lancaster, September 2022

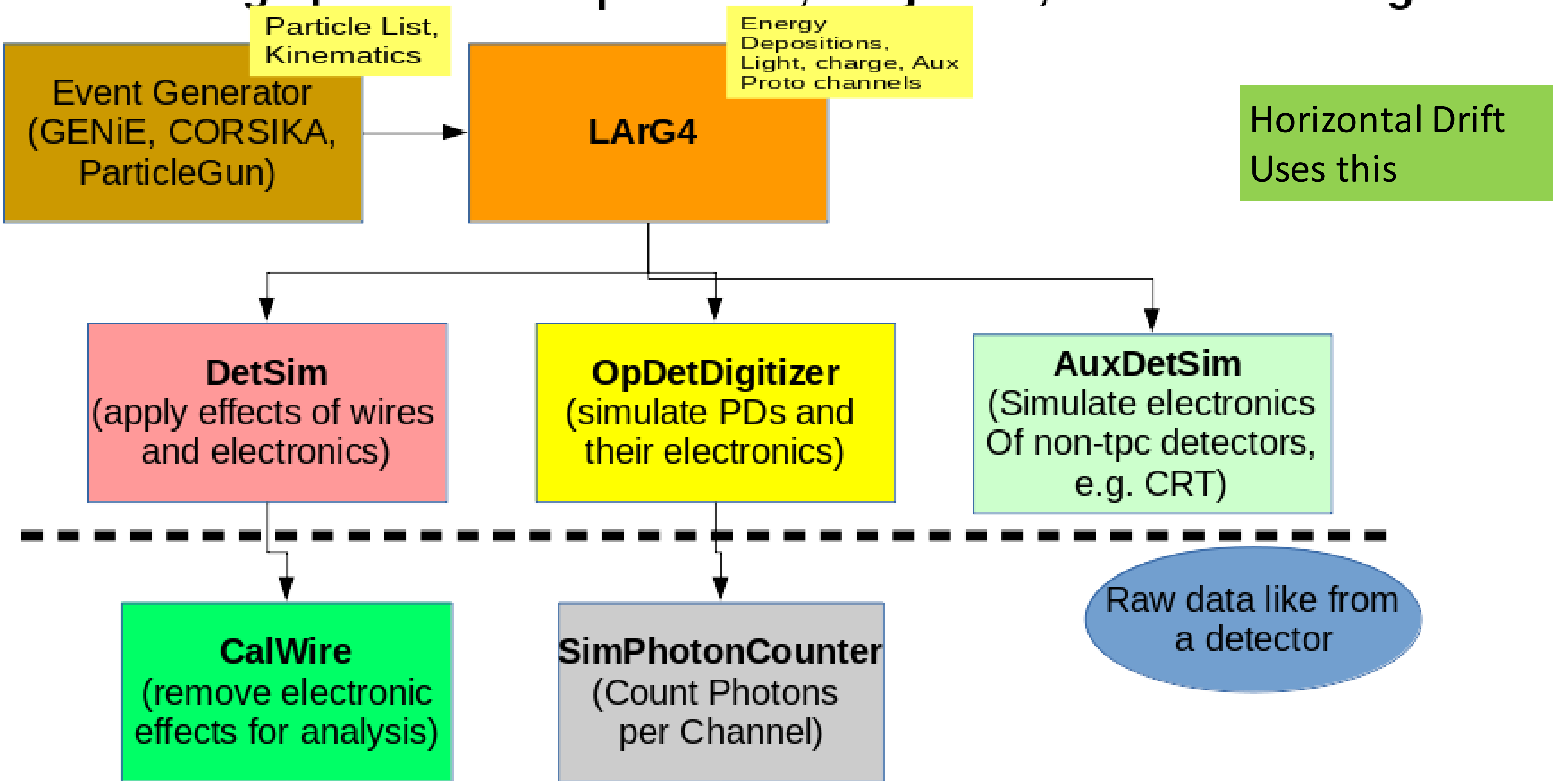
Outline

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

Simulation Flowchart (Legacy Version)

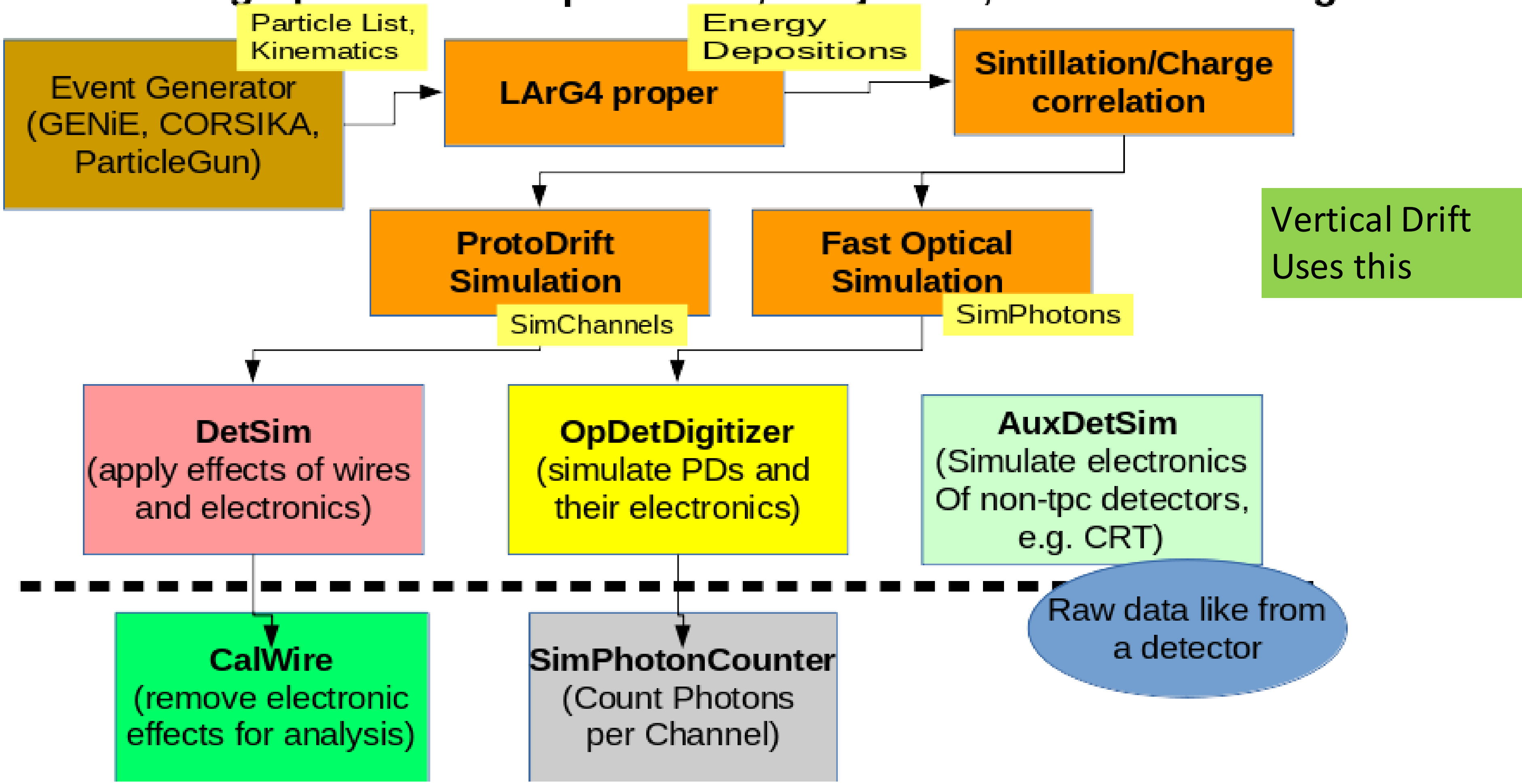
Each stage is a module

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



Simulation Flowchart (New Version)

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



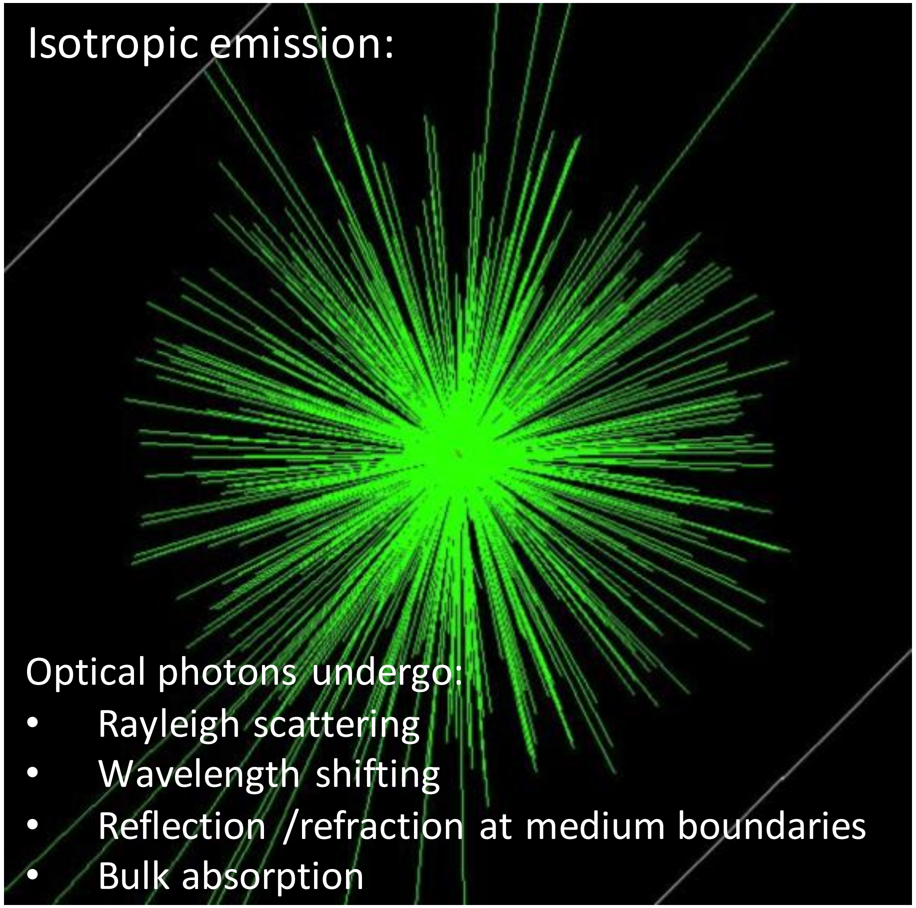
Elements of Light Sim in a Nutshell

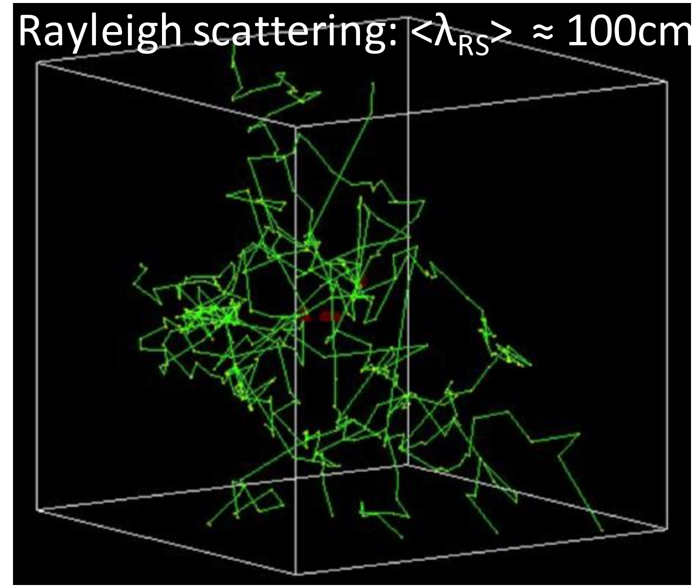
- **Light source:**
  - How many photons are generated?
  - What is their time distribution?
  - What is their wavelength?
- **Transport:**
  - How many photons make it to the detector?
  - How long does it take them?
  - Do they scatter/get absorbed etc?
- **Detection:**
  - What is our detection efficiency?
  - Does it depend on position on detector?
  - Are there any extra timing effects?

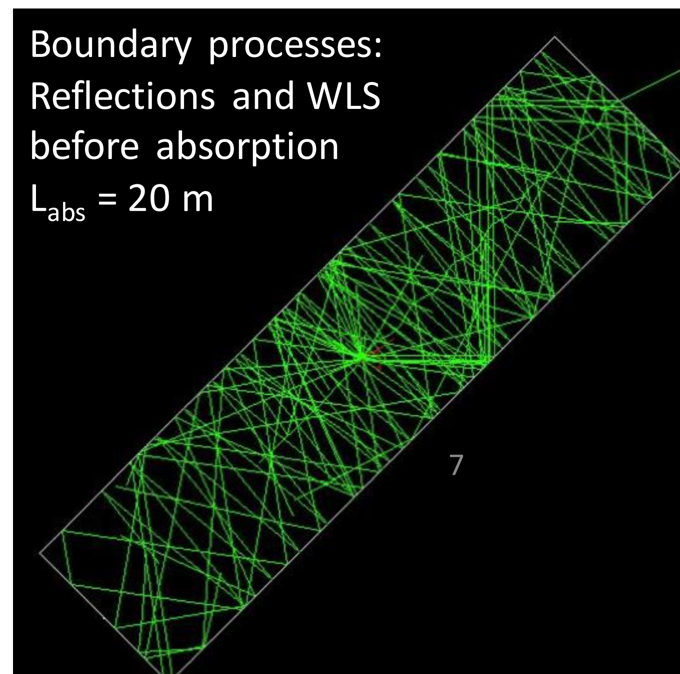
Different modes of simulation

- 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: Semi-analytic, optical library, GANN.

Full optical light simulation

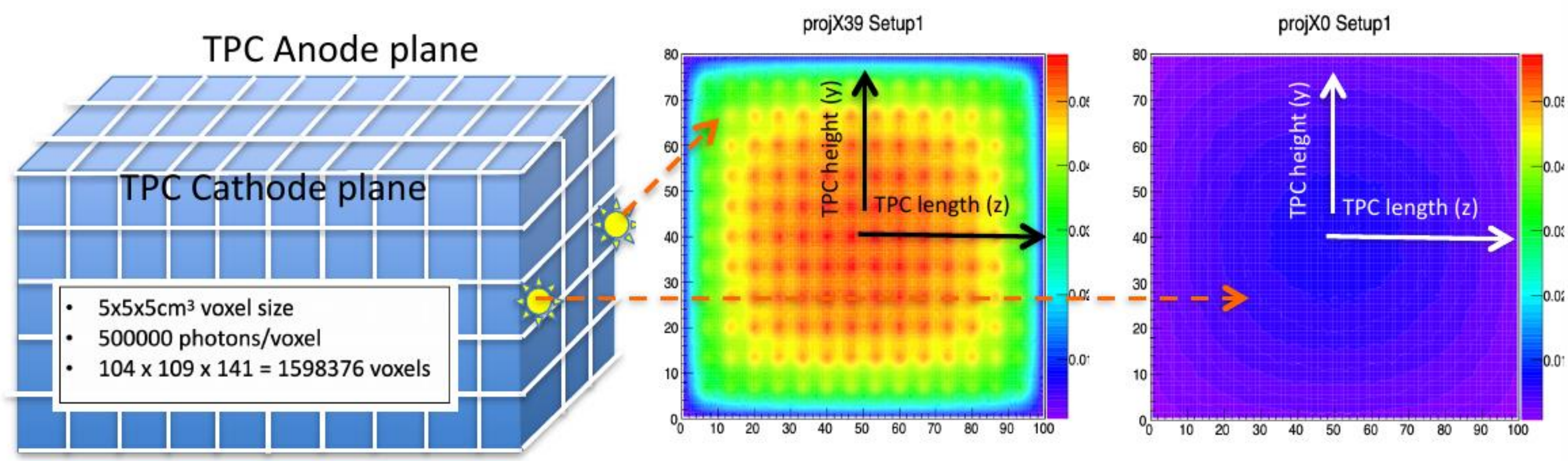




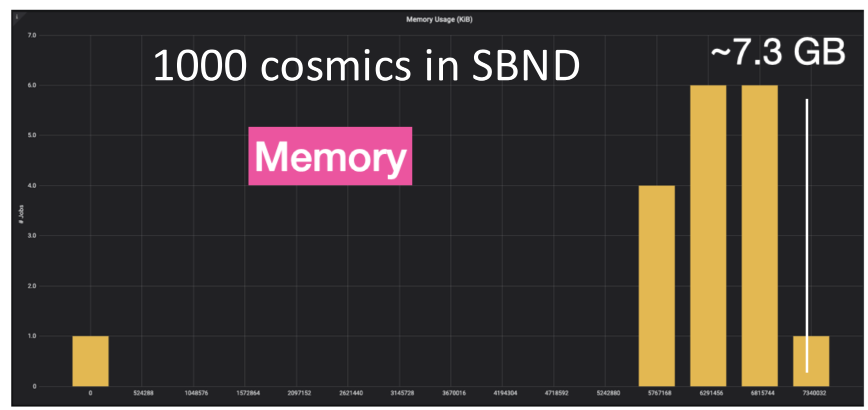


In large detectors, the tracking of each individual photon is prohibitively long:
approaches need to be used →

Fast optical model: Optical Library



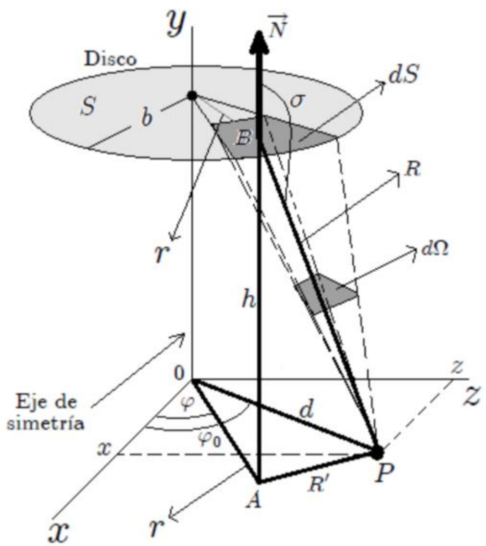
$$\langle N\rangle_{PMT-hits} = \left(\frac{dE}{dx}_{step}\cdot Length_{step}\right)\cdot LY\cdot visibility^{PMT}_{step}$$



- 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 events with large energy depositions (i.e. cosmics)
- Difficult to get working in DUNE, so different approach currently used.

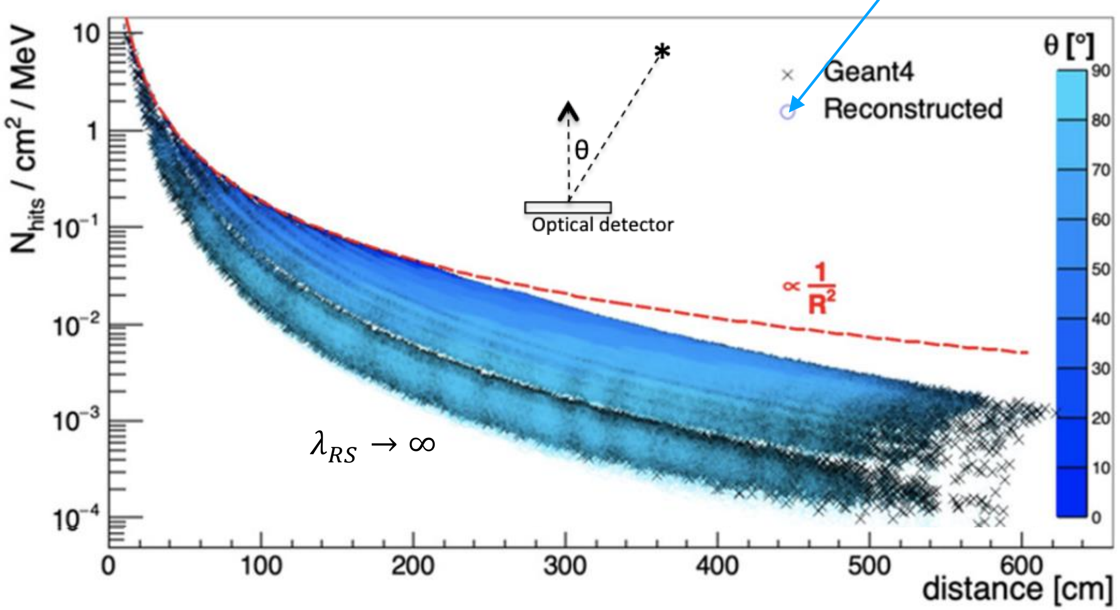
Fast optical model: Semi-Analytic



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

- 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

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



λ_{abs} = *LAr absorption length*

S_γ = *Scintillation Yield*

$\mathscr{E}$ = *Electric Field*

Ω = *Solid angle*

- **"Almost" because we have Rayleigh scattering**

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.

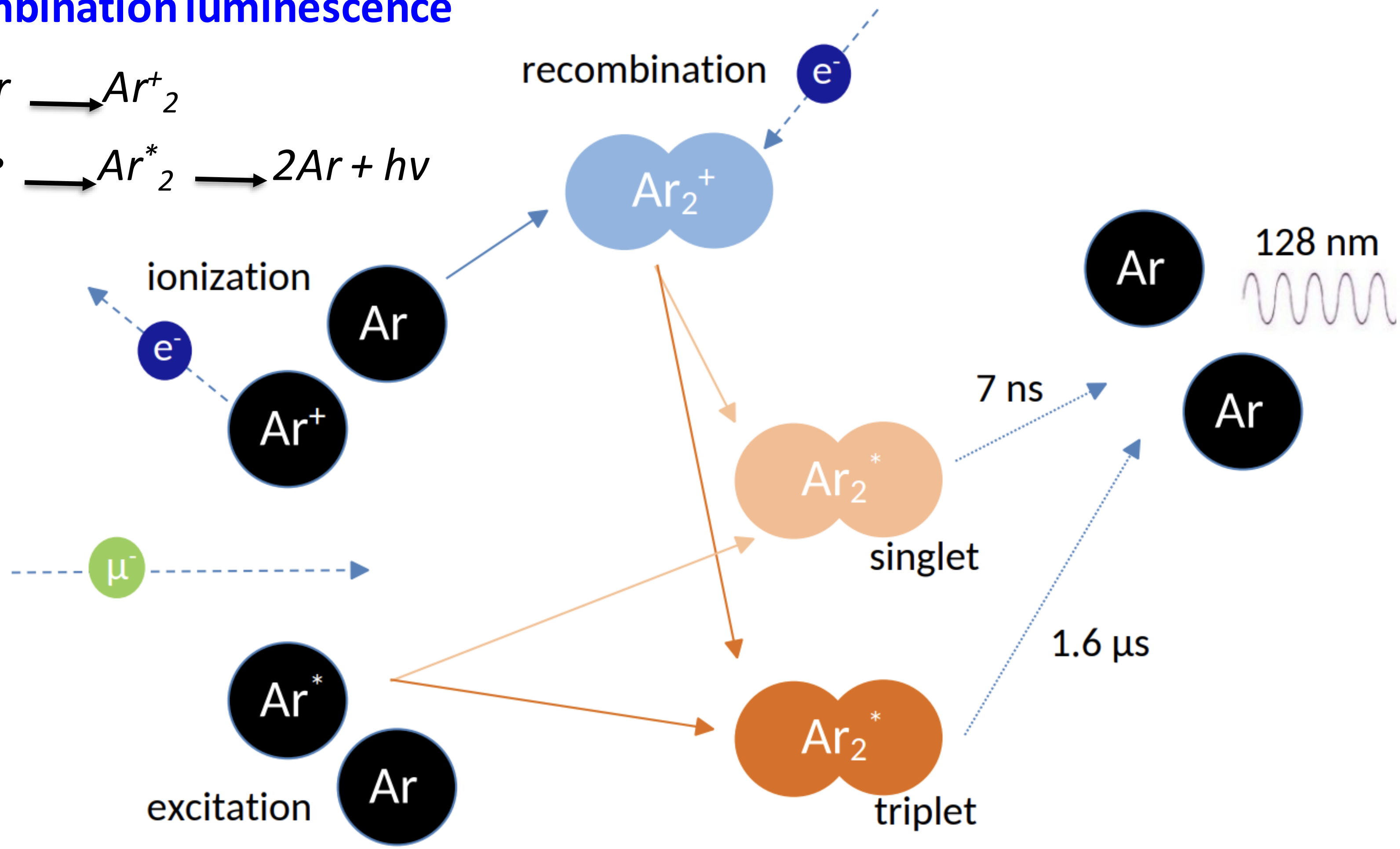
Emission

Scintillation mechanism in LAr

- **Recombination luminescence**

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

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

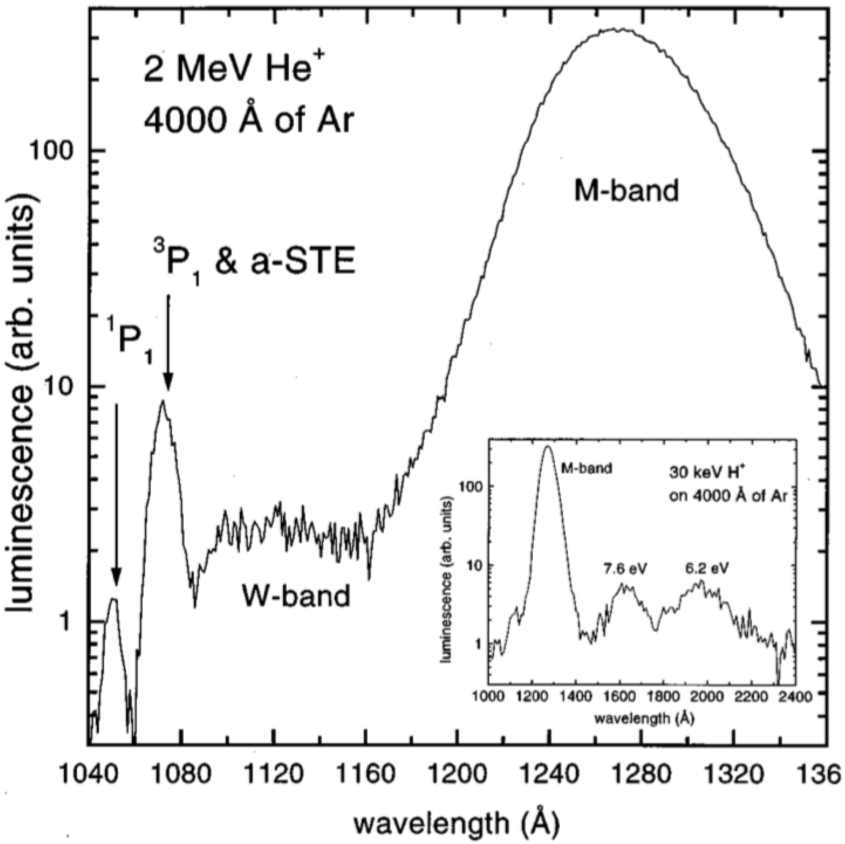


- **Self-trapped excitation luminescence**

$$Ar^{*} + Ar \longrightarrow Ar^{*}_{2} \longrightarrow 2Ar + h\nu$$

Scintillation wavelength in LAr

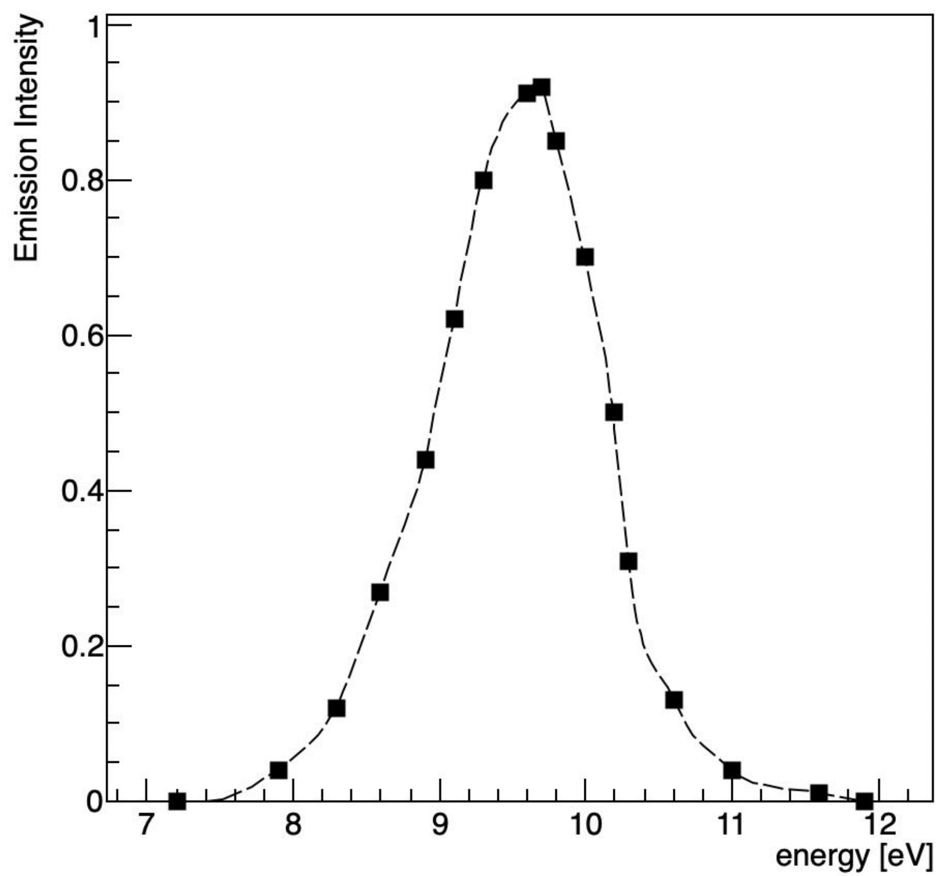
Ph. Rev. B 56 (1997), 6975



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

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,  10.6,  11,  11.6,  11.9]
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,  10.6,  11,  11.6,  11.9]
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, 0.13, 0.04,  0.01, 0.0]
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, 0.13, 0.04,  0.01, 0.0]
```



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 excimer decays (at 90 K) characterized by two very different components: *a fast component*, with a time constant of $\tau_S \approx 6$ns, and by a slow component, with a time constant of $\tau_T \approx 1.3\mu s$

lardataalg / lardataalg / DetectorInfo / larproperties.fcl

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

Warning: In the refactored LArG4 decay times are defined in a different place

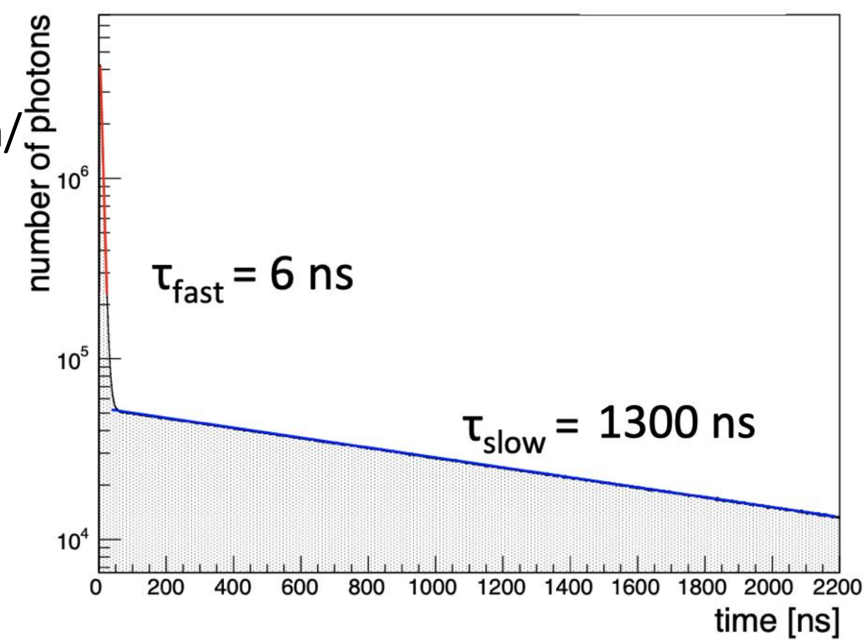
This is where DUNE-specific parametrizations live now:

duneopdet/duneopdet/**PhotonPropagation/**

Note:

(*) A slow time constant value convolved with the WLS-delay, results in a larger value.

That is what we're currently doing in DUNE

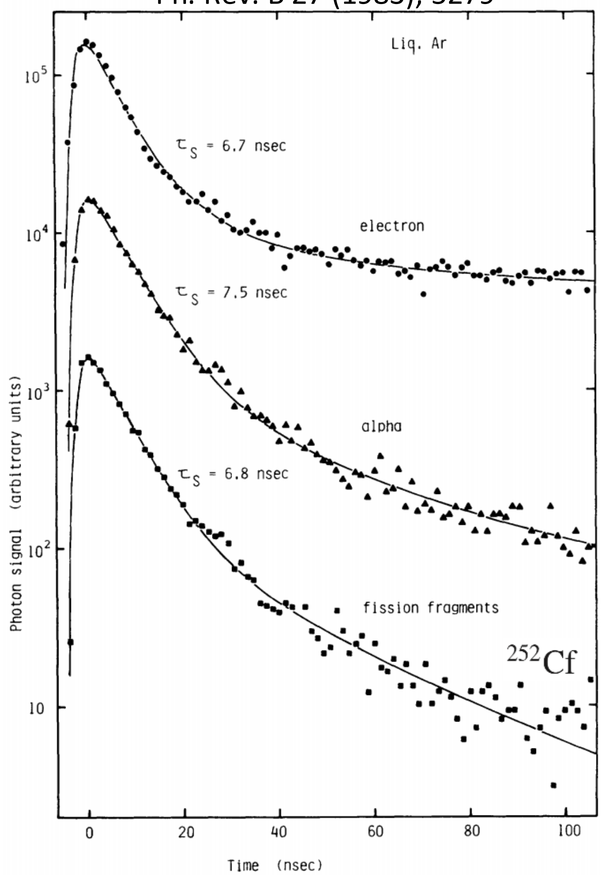


Scintillation yields

- The lifetimes of the fast and slow components agree within experimental uncertainties for different particles
- Light yield and fast/slow ratio depend on LET (the specific energy loss along the path)

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

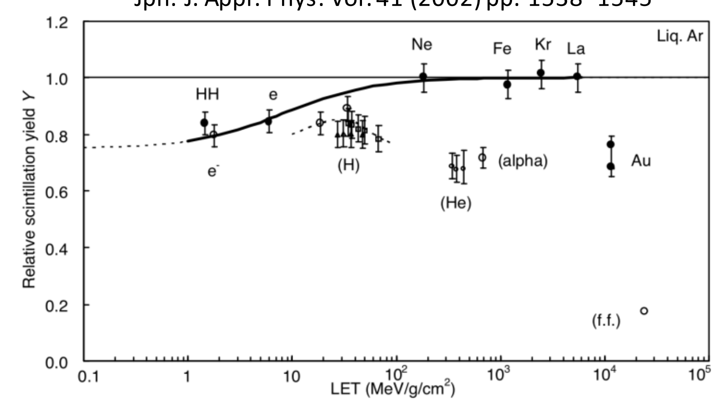
Ph. Rev. B 27 (1983), 5279



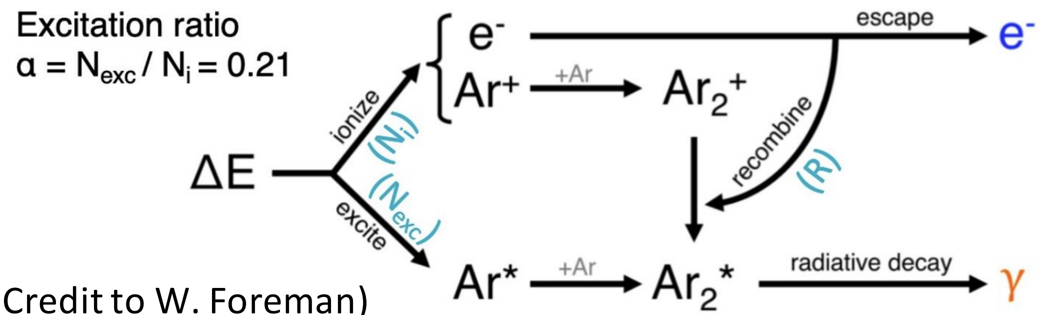
lardataalg / lardataalg / DetectorInfo / larproperties.fcl

```
ScintYield:           24000. # total scintillation yield (ph/Mev)
ScintYieldRatio:      0.3    # fast / slow scint ratio (needs revisitting)
ScintByParticleType:  true   # whether to use different yields and

# Scintillation yields and fast/slow ratios per particle type
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
```

Jpn. J. Appl. Phys. Vol. 41 (2002) pp. 1538–1545



L vs Q and Electric Field



(Credit to W. Foreman)
PHYSICAL REVIEW D **101**, 012010 (2020)

$$Q = N_e = N_i R$$

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

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

(19.5±1.0) eV

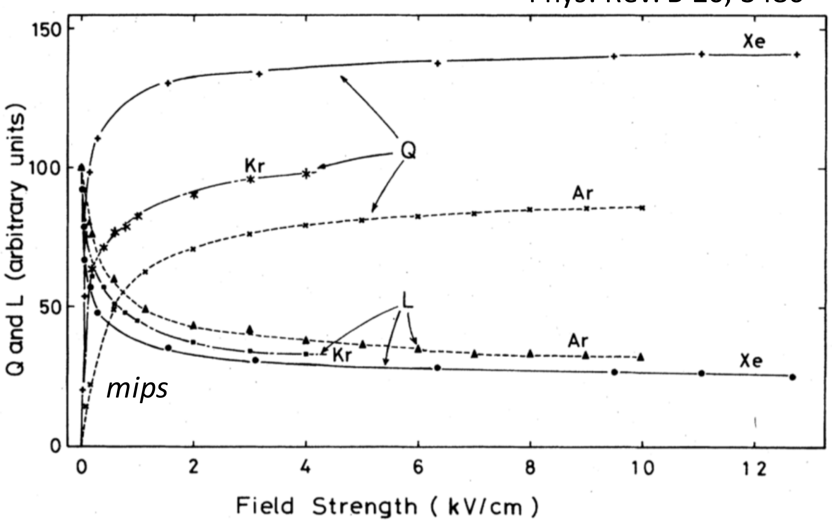
larsim / larsim / IonizationScintillation / **ISCalcCorrelated.cxx**

```
// 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;
```

```
// calculate recombination survival fraction
if (fUseModBoxRecomb) {
  if (ds > 0) {
    double Xi = fModBoxB * dEdx / EFieldStep;
    recomb = log(fModBoxA + Xi) / Xi;
  }
  else {
    recomb = 0;
  }
}
else {
  recomb = fRecombA / (1. + dEdx * fRecombk / EFieldStep);
}
```

- Electric Fields applied to the LAr medium also affect the intensity weights of the decay components by the recombination (R)

Phys. Rev. B 20, 3486



dunesim/LArG4/IonAndScint_dune.fcl

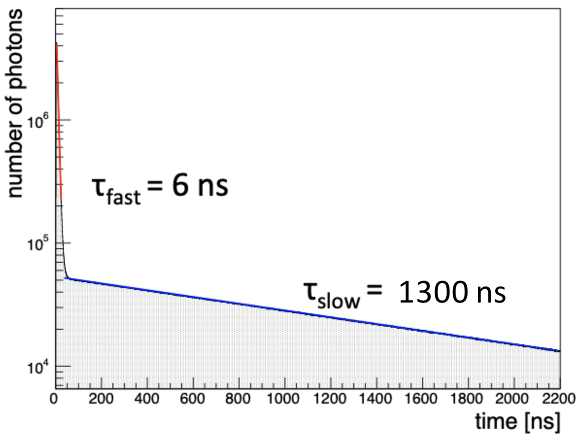
Services.LArG4Parameters.IonAndScintCalculator: "Correlated" (available from v09_09_03)

Propagation

Scintillation light propagation

<u>Scintillation</u> (emission):

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



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

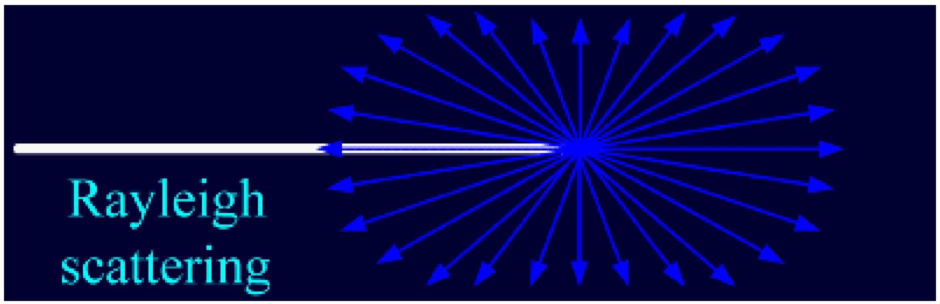
$$Q + L = N_{ex} + N_i = \frac{\Delta E}{W_{ph}}$$

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

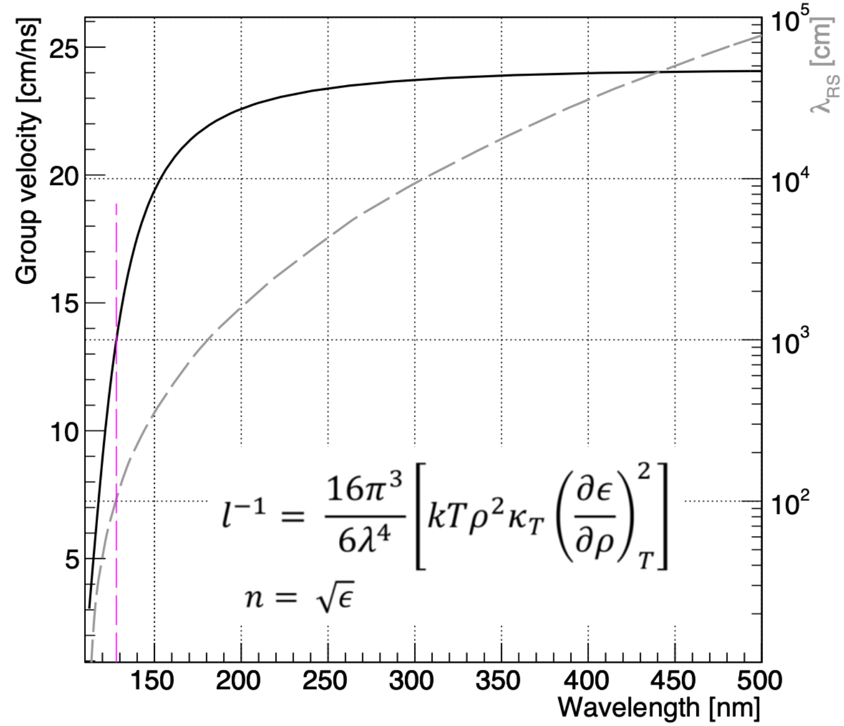
- Scintillation photons have energy lower than the first excited state of the Ar atom, therefore pure LAr is transparent to its own scintillation radiation
- However, during propagation through LAr VUV photons may undergo elastic interactions on Ar atoms ⇒ Rayleigh scattering
- **Rayleigh Scattering 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 LArSoft

- Elastic scattering of photon with medium of particle ~1/10 size of the wavelength (change angle/direction)



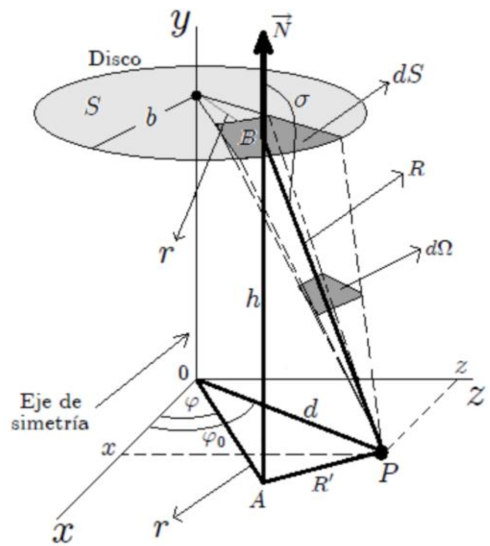
- Small uncertainties in the index of refraction can drastically change the scattering length λ_{RS}



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
```

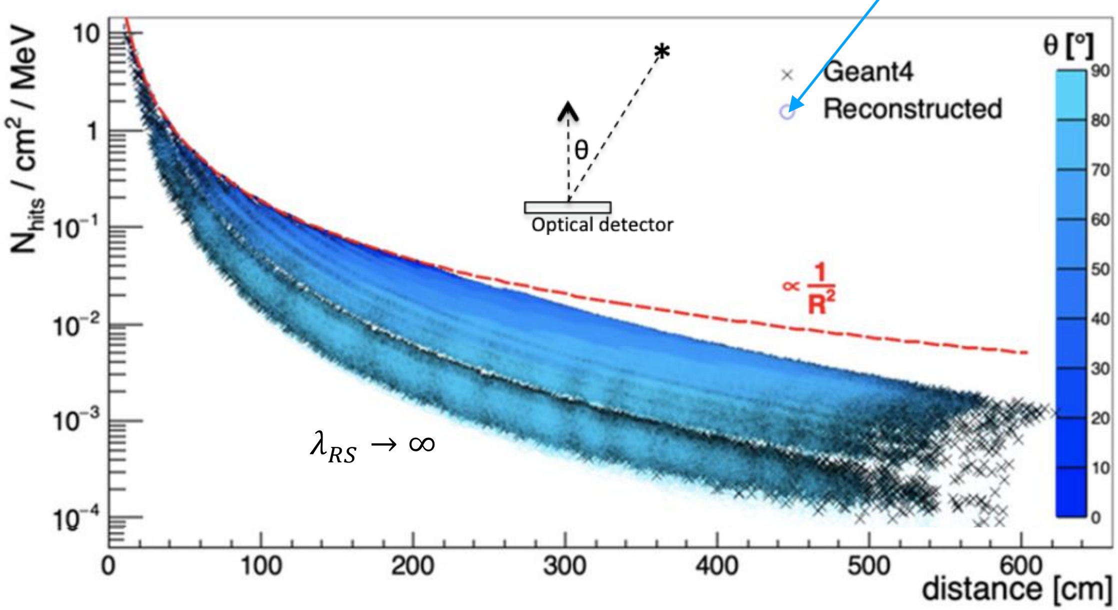
Fast optical model: Semi-Analytic



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

- 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

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



$\lambda_{abs} = LAr\ absorption\ length$

$S_\gamma = Scintillation\ Yield$

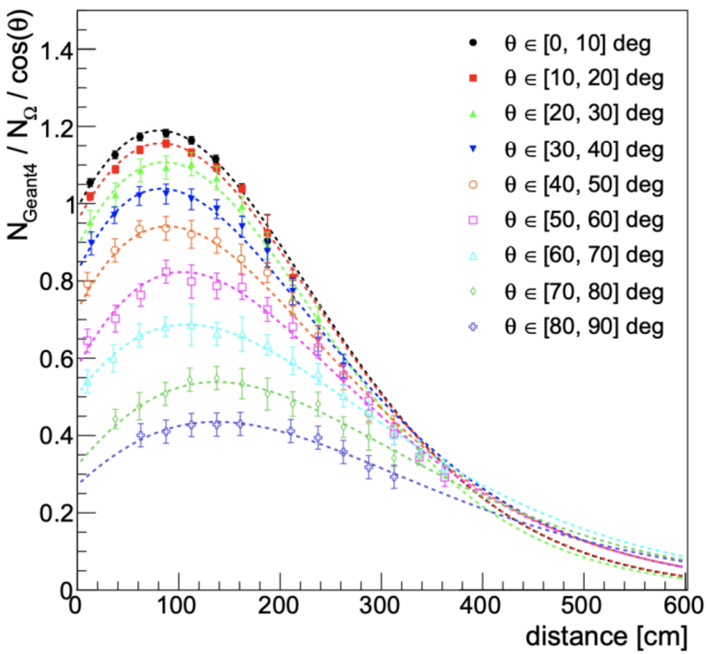
$\mathscr{E} = Electric\ Field$

$\Omega = Solid\ angle$

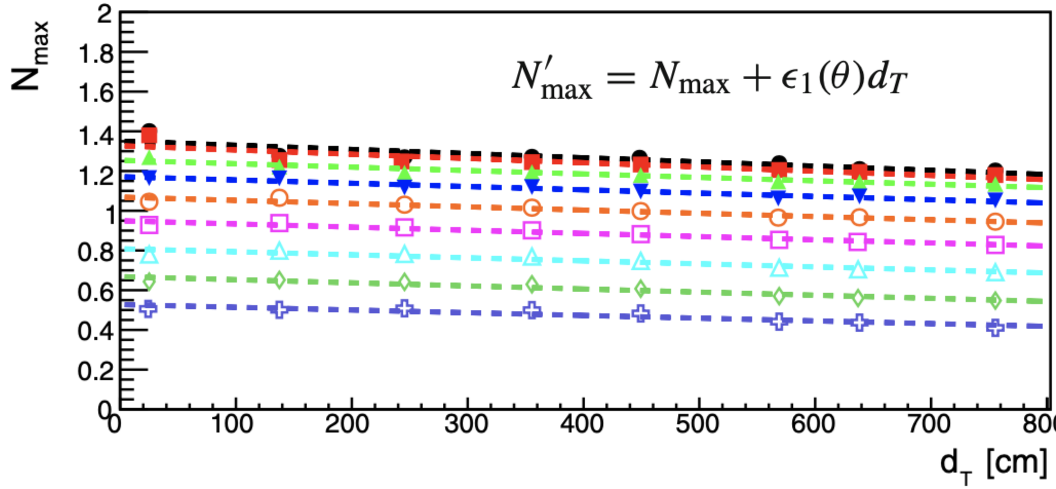
- **"Almost" because we have Rayleigh scattering**

Transport corrections to light signals

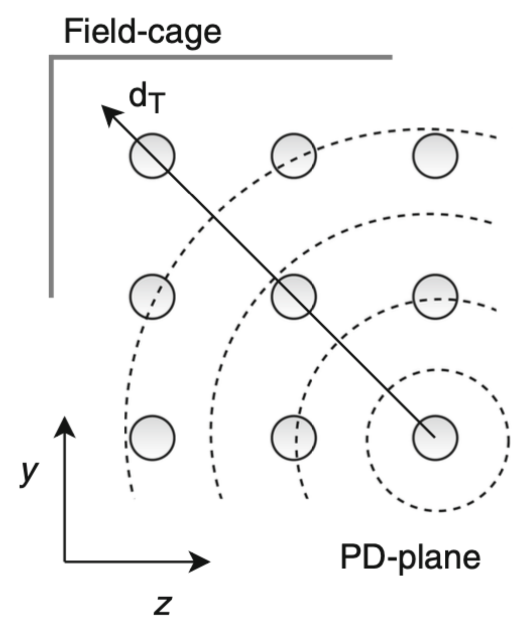
Eur. Phys. J. C (2021) 81:349



Border effects:



$$GH(d) = N_{max}\left(\frac{d - d_0}{d_{max} - d_0}\right)^{\frac{d_{max} - d_0}{\Lambda}} e^{\frac{d_{max} - d}{\Lambda}}$$



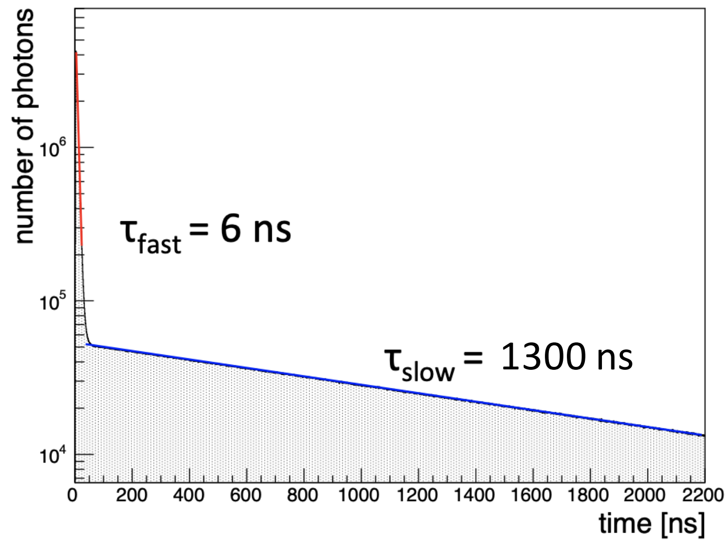
$$N_\gamma = N_\Omega \times GH'(d, \theta, d_T)/cos(\theta)$$

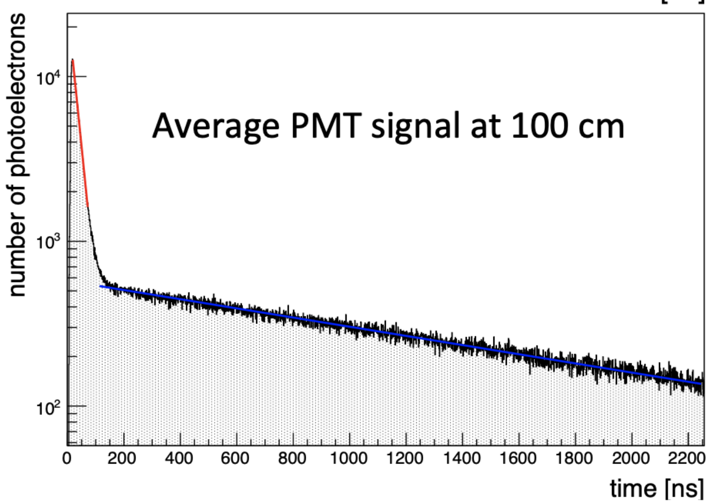
Geometric estimation Transport correction

Time structure of detected signals

Scintillation (emission):

0.3 x τ_{fast}(6 ns) + 0.7 x τ_{slow}(1300 ns)

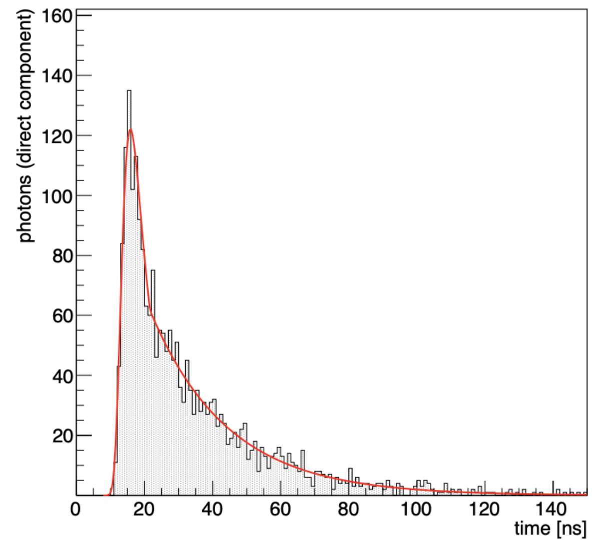




+

Propagation:

Direct transportation + Rayleigh Scattering



=

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

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

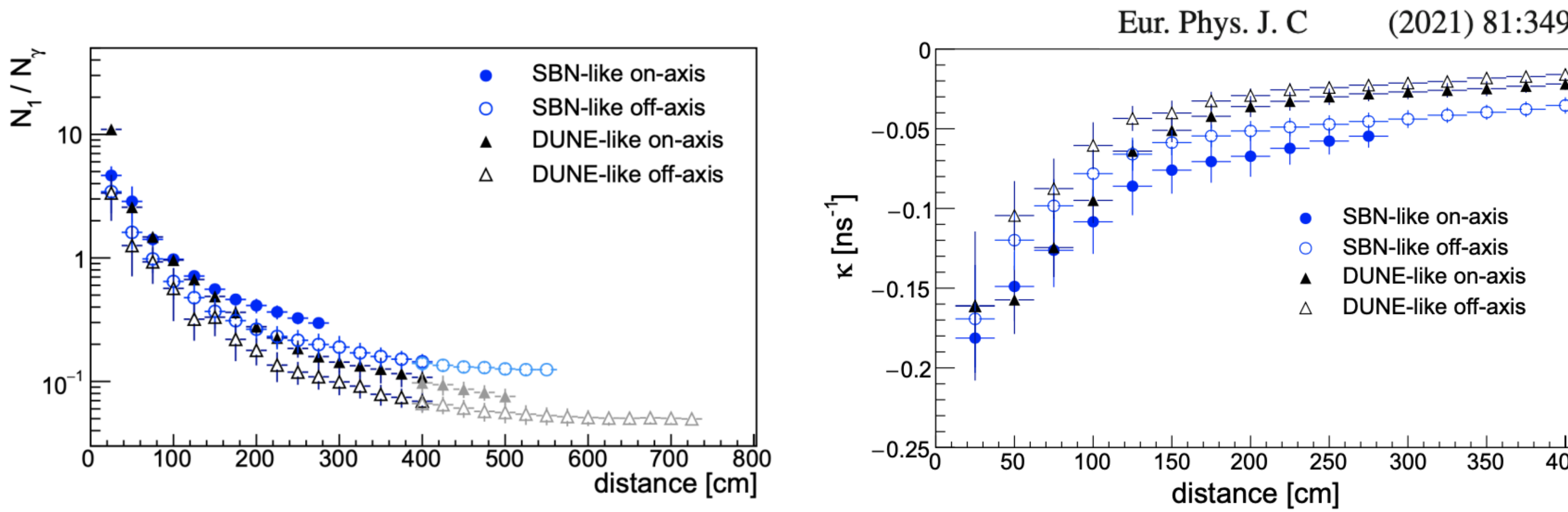
t_E = *emission time*

t_t = *transport time*

t_{WLS} = *WLS delay time*

t_{det} = *detector time*

Time structure of detected signals



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

larsim / larsim / PhotonPropagation / opticalsimparameterisations.fcl

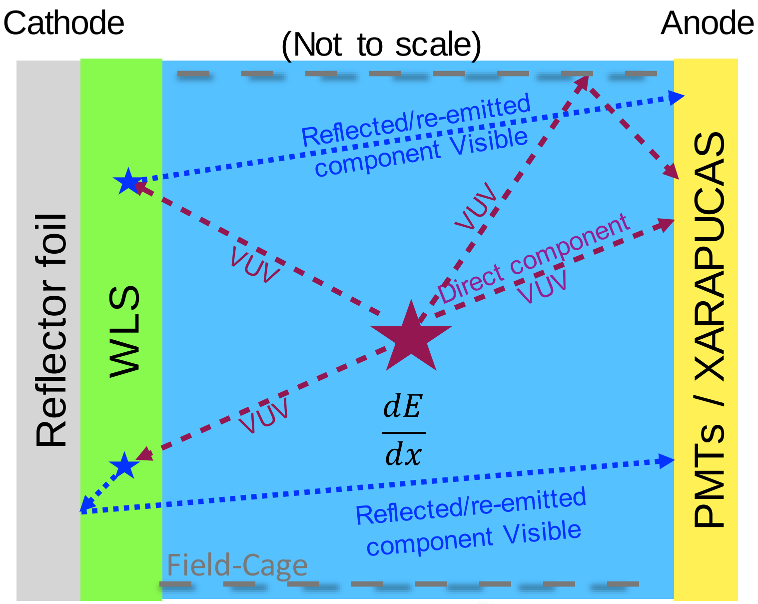
```
# VUV/DIRECT LIGHT: TIMING PARAMETERISATION
# Parameters of the Landau + Exponential (<= 350 cm) and Landau (> 350 cm) models
# Landau parameters
Distances_landau_generic: [0, 25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300, 325, 350, 375, 40
Norm_over_entries_generic: [ [4.64837, 4.64837, 2.86581, 1.4143, 0.974871, 0.71311, 0.55772, 0.461078, 0
                0.297132, 0.297132, 0.297132, 0.297132, 0.297132],
                [3.43562, 3.43562, 1.61042, 0.981127, 0.64465, 0.476552, 0.369063, 0.310461, 0.264819, 0
Mpv_generic: [  [2.73373, 2.73373, 3.599, 5.80141, 7.57883, 9.56959, 11.6047, 13.6676, 15.6126, 17.5389,
                21.3254, 21.3254],
                [2.19076, 2.19076, 4.0163, 5.86531, 8.09466, 10.4547, 12.9261, 15.2731, 17.7939, 20.6664
```

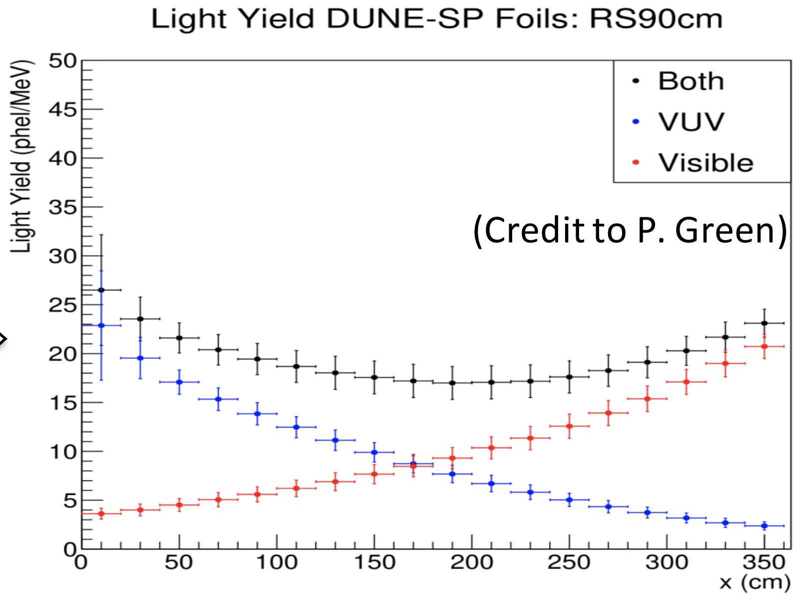
larsim / larsim / PhotonPropagation / PDFastSimPAR.fcl

```
IncludePropTime:        true
```

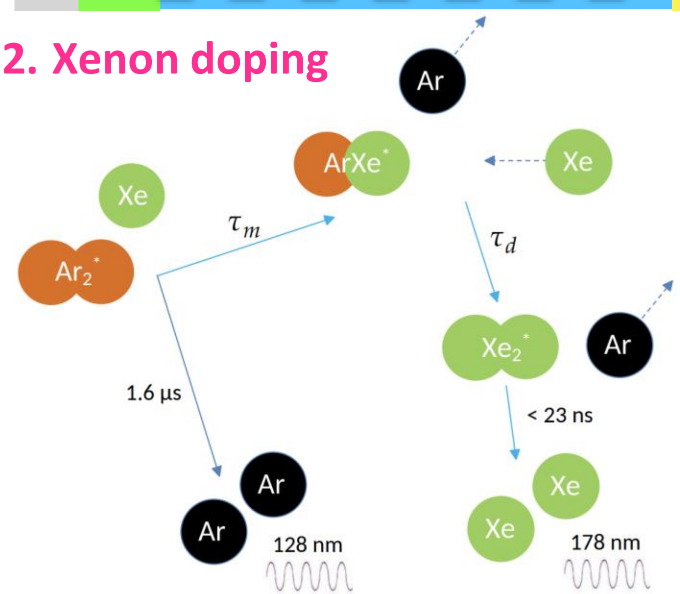
(Digression): Enhancing the Light Yield in LArTPCs

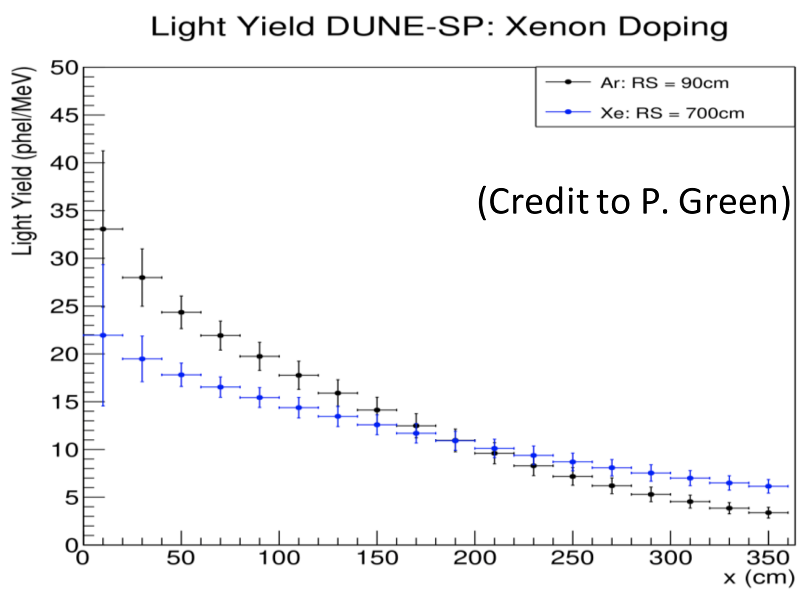
1. WLS-Coated reflector foils





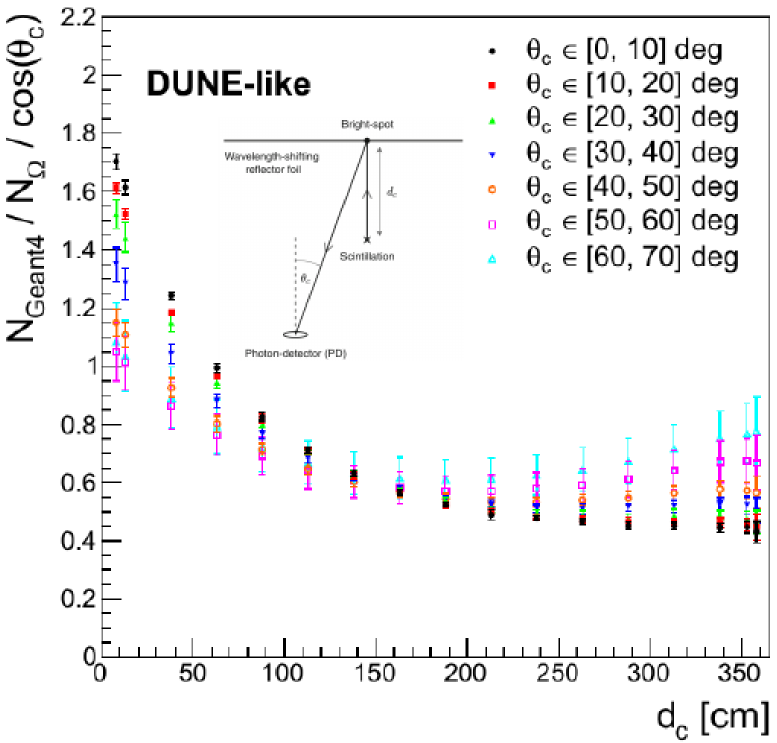
2. Xenon doping

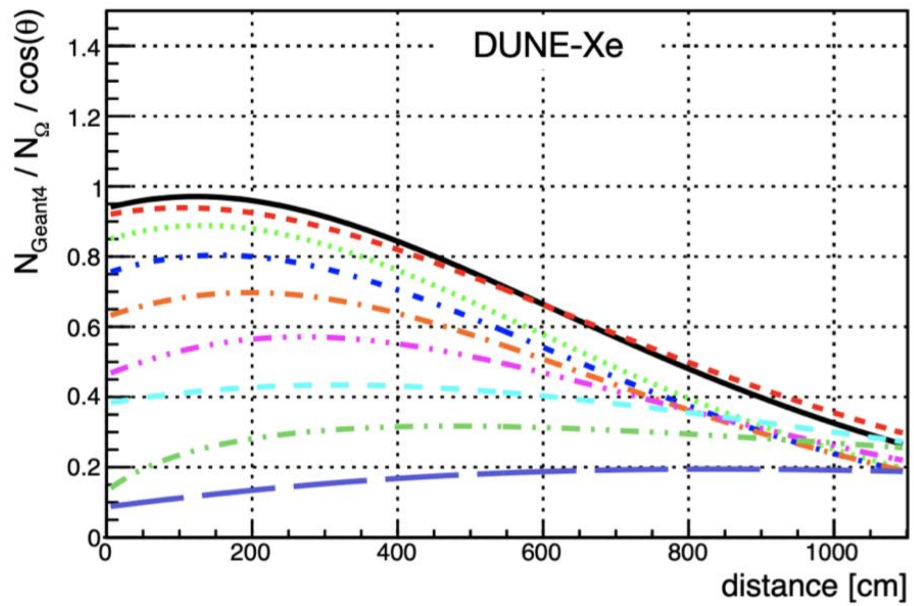




Semi-Analytic model extensions (available)

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

$Q_{WLS} \times Q_{foil}$

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 suits Semi-Analytic model simulation incorporating all of the extensions:
  - LAr and LXe wavelengths (doping)
  - Direct and Reflected light (foils)

Fast optical model: Semi-Analytic

duneopdet/duneopdet/PhotonPropagation/opticalsimparameterisations_dune.fcl
Refactored:
duneopdet/PhotonPropagation/**PDFastSim_dune.fcl**

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


# VUV/DIRECT LIGHT: TIMING PARAMETERISATION
```

Detector specific

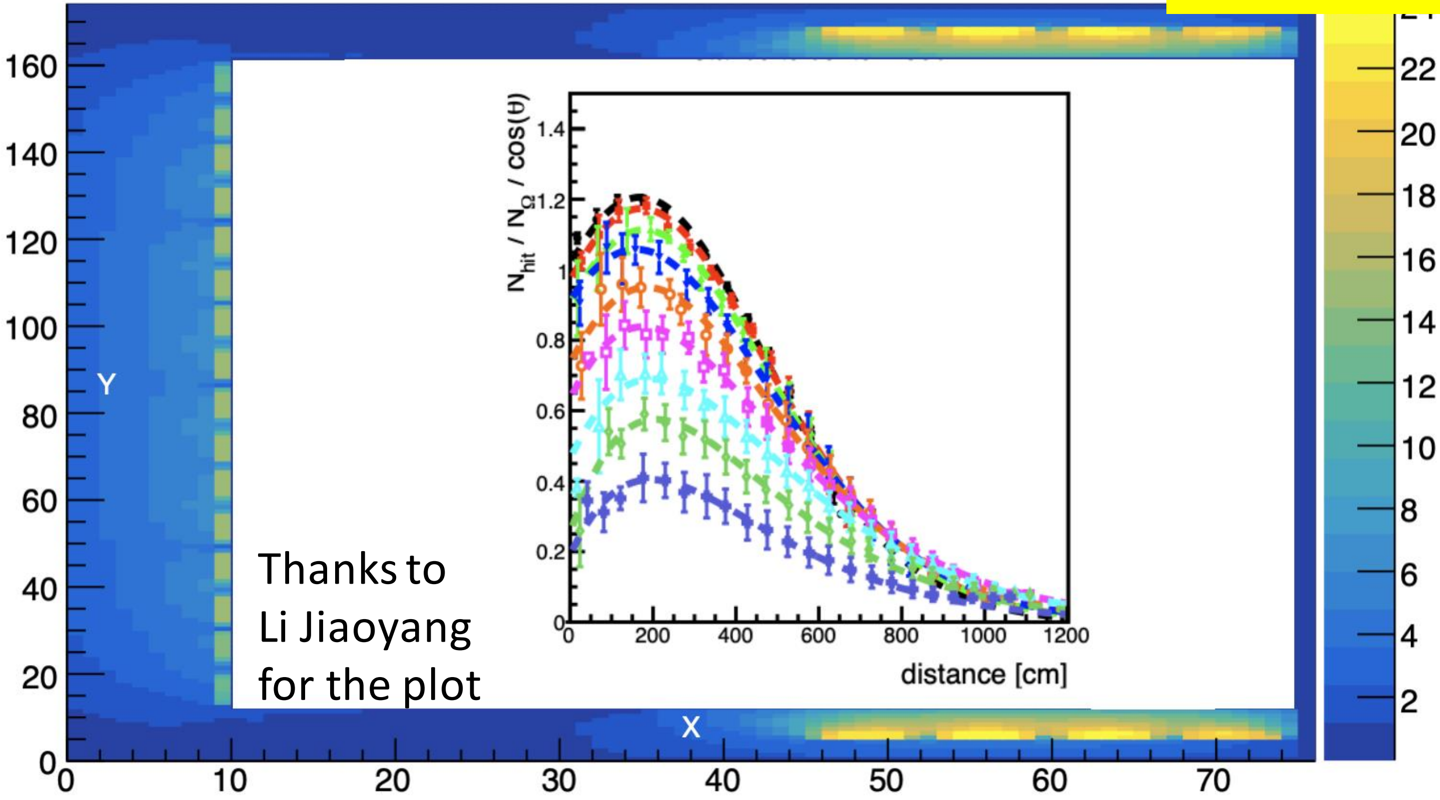
```
###########
# DUNE FD #
###########

# Hits & Timing parameterization for DUNE FD, Ar scintillation
dunefd_pdfastsim_par_ar:                        @local::standard_pdfastsim_par_ar
dunefd_pdfastsim_par_ar.VUVTiming:              @local::dune_vuv_timing_parameterization
dunefd_pdfastsim_par_ar.VUVHits:                @local::dune_vuv_RS100cm_hits_parameterization

# As above, with cathode reflections included
dunefd_pdfastsim_par_ar_refl:                   @local::dunefd_pdfastsim_par_ar
dunefd_pdfastsim_par_ar_refl.DoReflectedLight: true
dunefd_pdfastsim_par_ar_refl.VISTiming:         @local::dune_vis_timing_parameterization
dunefd_pdfastsim_par_ar_refl.VISHits:           @local::dune_vis_RS100cm_hits_parameterization

# As above, but fast-only scintillation for high Xe concentration
dunefd_pdfastsim_par_ar_fastonly:               @local::dunefd_pdfastsim_par_ar
dunefd_pdfastsim_par_ar_fastonly.DoSlowComponent: false
```

More configurations below.

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

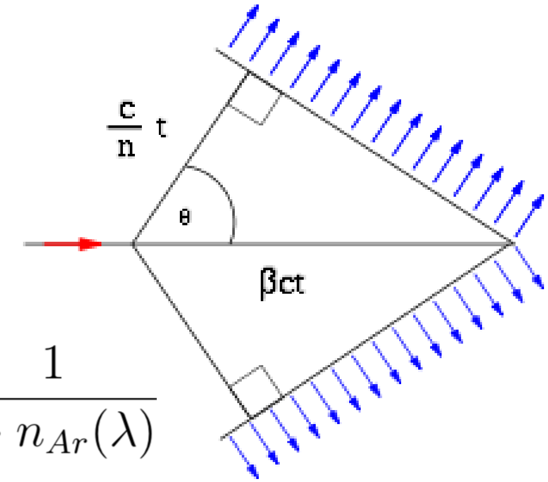
Currently implemented in DUNE -VD



Thanks to Li Jiaoyang for the plot

(Digression): 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

$$\cos\theta_{\check{C}} = \frac{1}{\beta \cdot n_{Ar}(\lambda)}$$

NIM A 516 (2004) 348–363

$$\frac{d^2N}{d\nu dx} = \frac{2\pi\alpha}{c}\sin^2\theta_{\check{C}}$$

$$\Rightarrow \int_{109nm\,\textit{(LAr absorbed)}}^{600nm\,\textit{(hard to detect)}}$$

$$\Rightarrow \quad R_{\check{C}} = \frac{dN_{\check{C}}/dx}{dN_{scint}/dx + dN_{\check{C}}/dx} = 2.4\%$$

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

duneopdet/duneopdet/PhotonPropagation/photolibbuild_services_dune.fcl

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

Detection

Wavelength shifter in LArSoft

Processes handled by `G4OpWLS`:

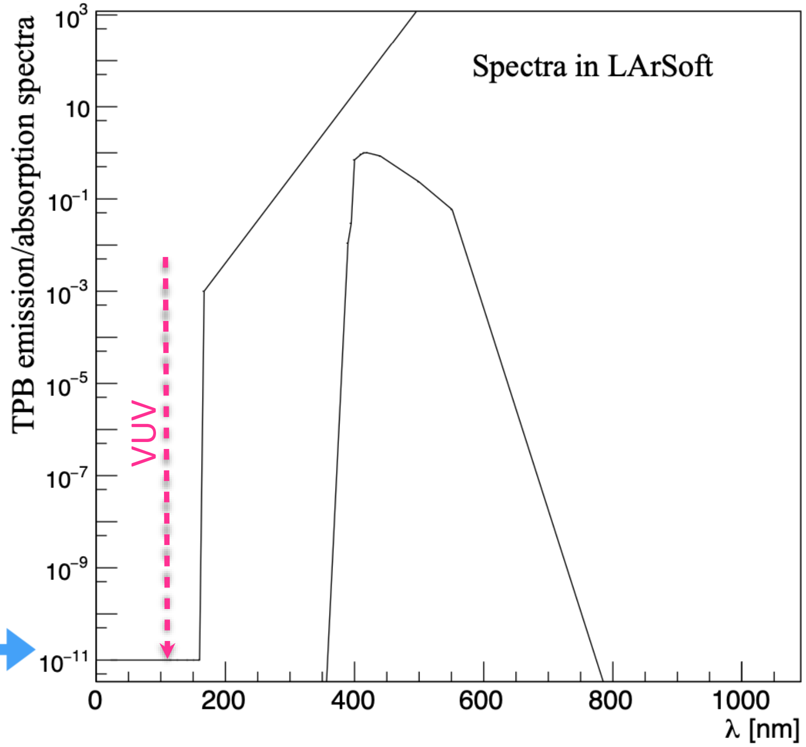
- Initial photon killed and a new one created with different wavelength
- User must provide:

• 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

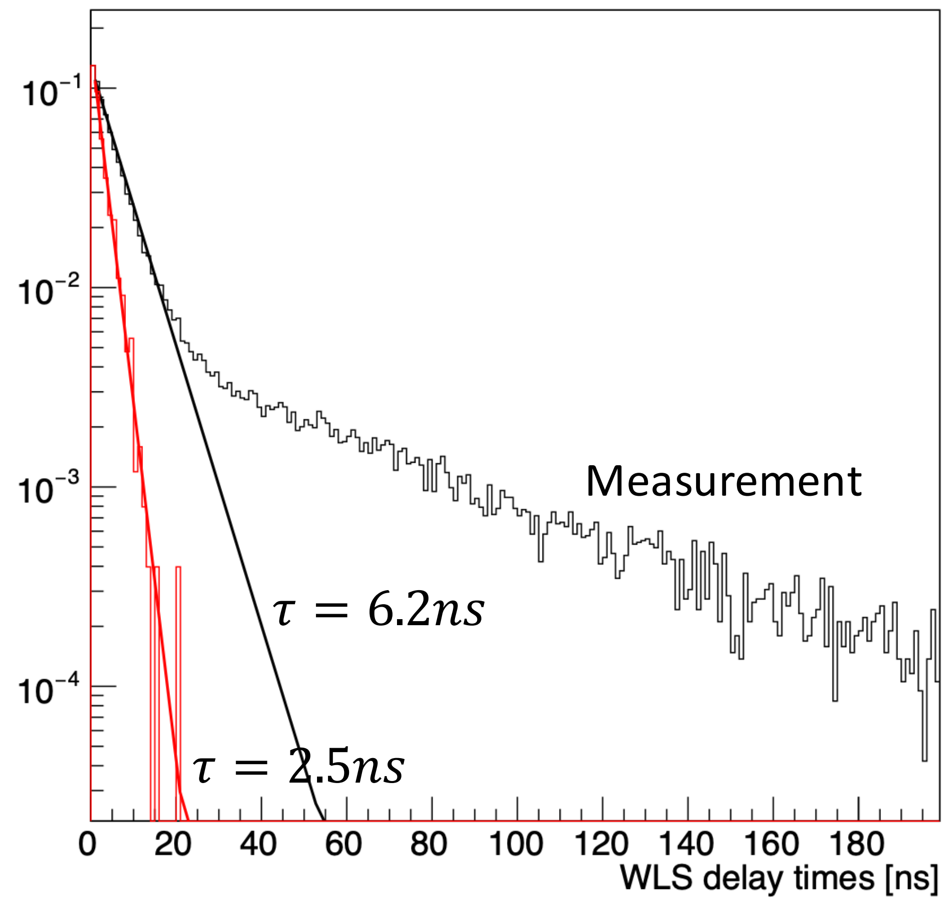
Geant4 (G4OpWLS class) only simulates Delta or Exponential model (none is the case for TPB)

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

TABLE I. Decay times and relative abundances of the components found in the decomposition into exponentials of the response function of TPB to 127 nm photons. Only statistical errors from the fit are quoted.

	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

- If we want to use the Geant4 class then we would have to approach it by a single exponential (~6.2 ns):
  - We know this is not what we measure
  - It would also require adding a line in OpticalPhysics (model switching not possible via .fcl).
- In DUNE we don't use the Geant4 WLS time simulation.



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
- while 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

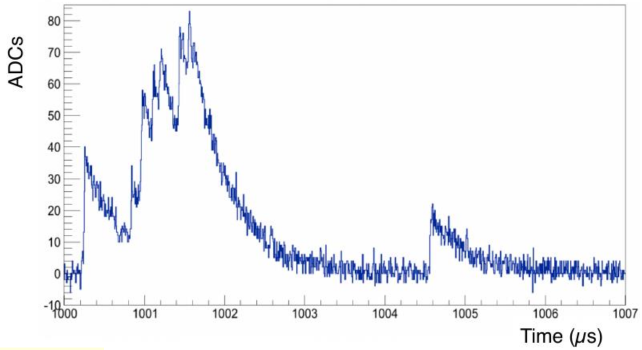
- Hopefully should make more sense now.

Detector effects

Signal digitisation

There are several different prototypes and electronics available in the LArSoft directory. My impression is that config files for many of them are not up-to date (**good project opportunity**)

duneopdet/duneopdet/**OpticalDetector**/



Main digitizer module is called **OpDetDigitizer,**

One of the configurations is: dunefd_opdigi_unganged

```
# Assume 25 V bias with Sensl C-series SiPMs
# Gain at this voltage is 4e6 -- that this corresponds to
# the MaxAmplitude and VoltageToADC below has not been confirmed.

VoltageToADC:        151.5     # Converting mV to ADC counts (counts in 1 mV)
LineNoiseRMS:          2.6     # Pedestal RMS in ADC counts, likely an underestimate
DarkNoiseRate:        10.0     # In Hz, Ranges 2-50 depending on Vbias
CrossTalk:             0.20    # Probability of producing 2 PE for 1 incident photon
# Afterpulsing:        0.006   # Afterpulsing is not yet simulated
Pedestal:           1500       # in ADC counts
DefaultSimWindow:   true       # Use -1*drift window as the start time and
                               # the TPC readout window end time as the end time
FullWaveformOutput: false      # Output full waveform. Be careful with this option:
                               # setting it to "true" can result in large output files
TimeBegin:             0       # In us (not used if DefaultSimWindow is set to true)
TimeEnd:            1600       # In us (not used if DefaultSimWindow is set to true)
PreTrigger:          100       # In ticks
ReadoutWindow:      1000       # In ticks
algo_threshold: @local::standard_algo_sspleadingedge
```

Each PE gets swapped For an electronics response (here Constructed from Parameters) Noise then added to waveform

```
dunefd_opdigi_threegang:                     @local::dunefd_opdigi_unganged
dunefd_opdigi_threegang.PulseLength:         0.876
dunefd_opdigi_threegang.PeakTime:            0.028
dunefd_opdigi_threegang.MaxAmplitude:        0.0594
dunefd_opdigi_threegang.FrontTime:           0.013
dunefd_opdigi_threegang.BackTime:            0.386
```

Added effects like cross-talk, etc..

Reconstruction

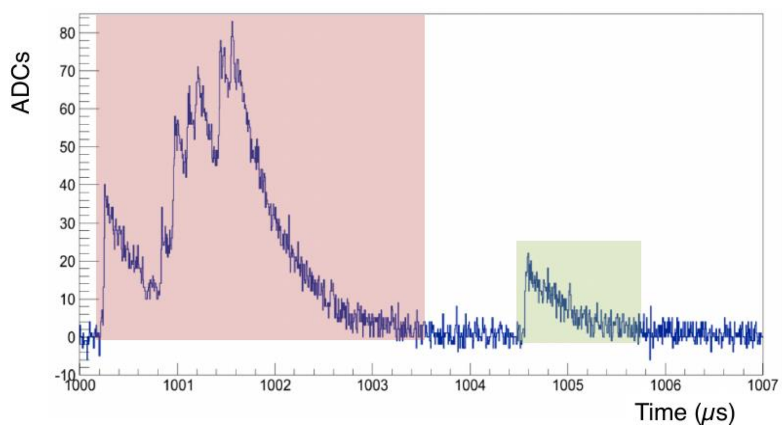
Optical signal reconstruction: OpHits

- The first stage of the optical reconstructions looks for pulses in the raw waveforms.
- The light pulses in LArSoft are stored in objects called OpHits.

- OpHits are found when the waveform is above certain threshold and held until continues to be so.
- Especially for SiPM signals this can lead to merging of visibly separate optical signals.
- OpHit Time is decided on the first arriving photon.

lardataobj / lardataobj / RecoBase / OpHit.h

```
    OpHit(); // Default constructor

private:

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



Optical signal reconstruction: OpFlash

- OpHits from different photon detectors are combined into Flashes. These are analogous to clusters in the charge reconstruction, but matched in time rather than space
- Having a flash allows us to try to reconstruct the position of the particle 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?
```

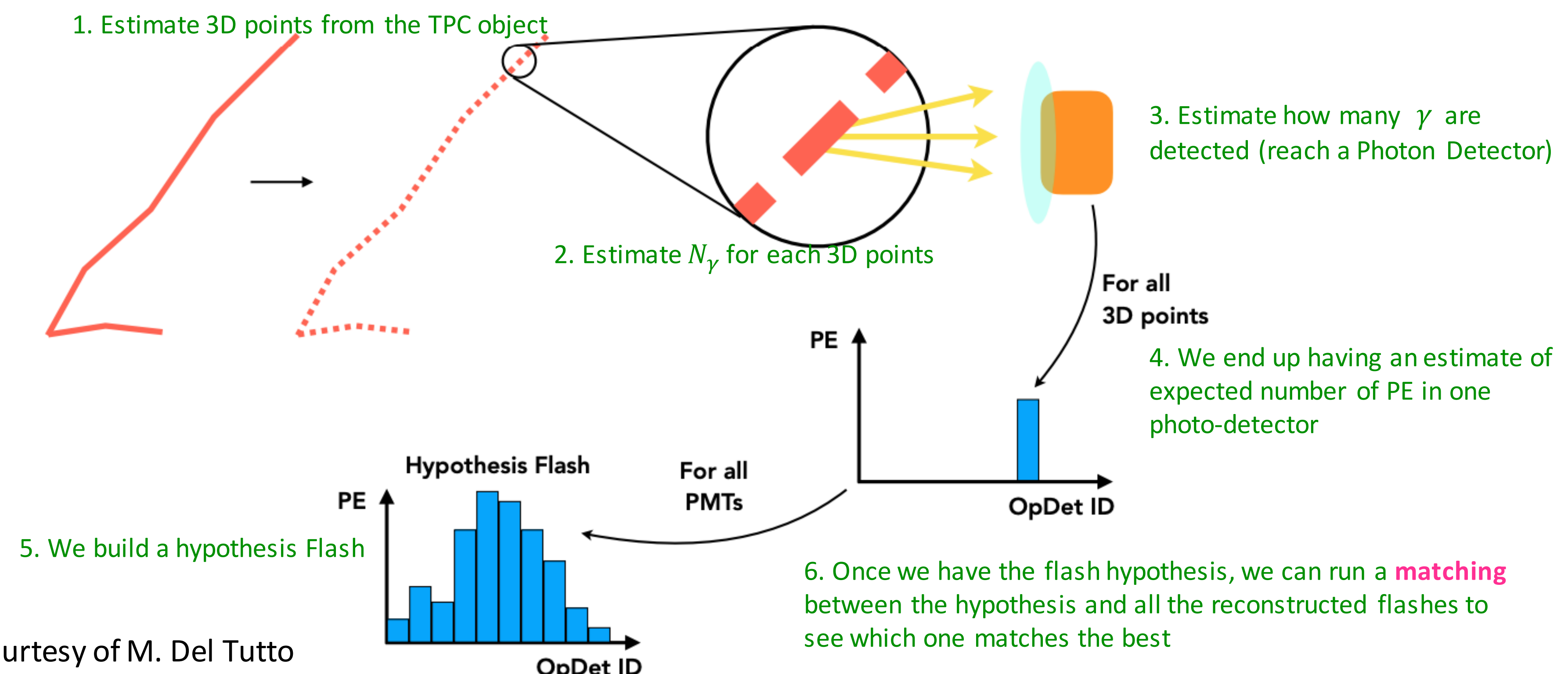
Flash Matching: `OpT0Finder` example

Flash matching goals:

- Identify 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

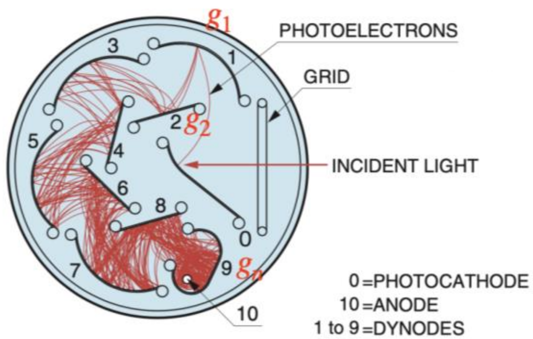
- Optical simulation is tricky, need to cut some corners to get it working in DUNE (size, number of photons -> Memory, CPU).
- The fast simulation now works for both HD and VD Far Detectors.
- There are many projects to look at: optical Reconstruction for VD is not battle tested/optimised and many other places.
- People to contact:
  - PDC Phys conveners: Laura Paulucci, Andrzej Szelc, Michel Sorel
  - Experts at large: Alex Himmel, Diego Garcia Gamez, Patrick Green

Have fun!

PMT Gain fluctuation

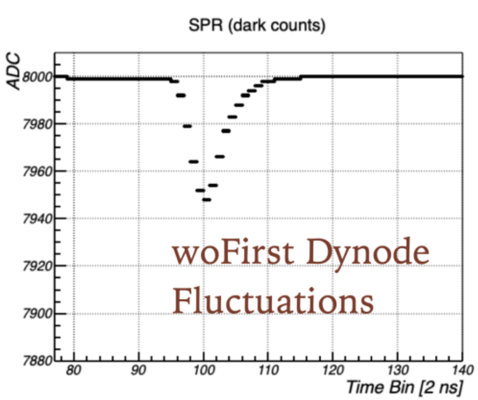
(Slide from F.J. Nicolás)

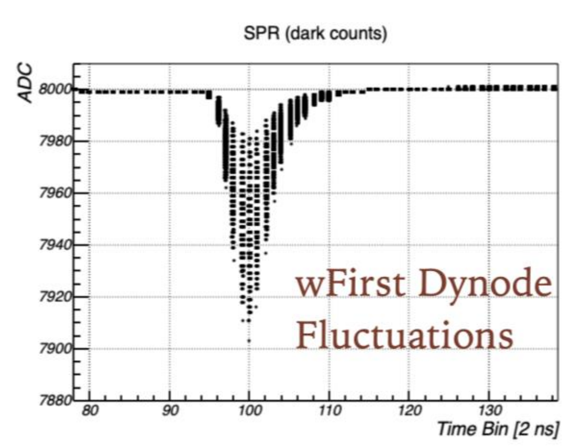
- Number of secondary electrons generated at each dynode: random variable



- 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

(Parenthesis): Light Signal Deconvolution

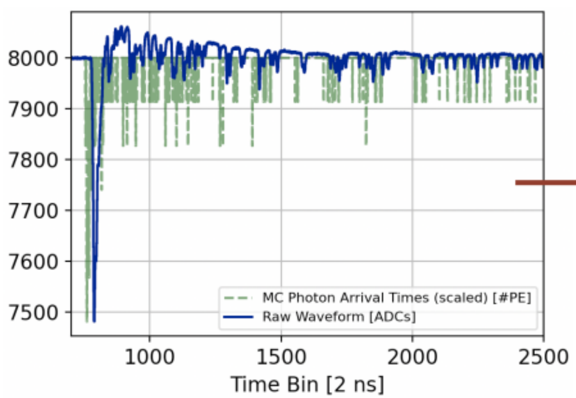
- In SBND we have PMT (and XARAPUCA) readout with **AC coupling**: bipolar SER ⇒ This makes accurate light reconstruction **a challenge** (by F.J. Nicolás)

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

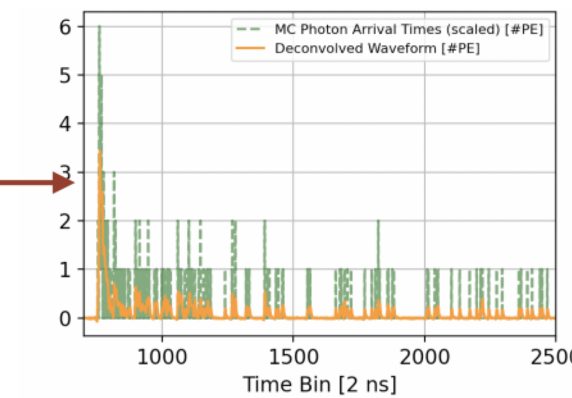
Starts with the **raw::OpDetWaveform** objects (from DetSim stage)

OpDeconvolution module

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



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



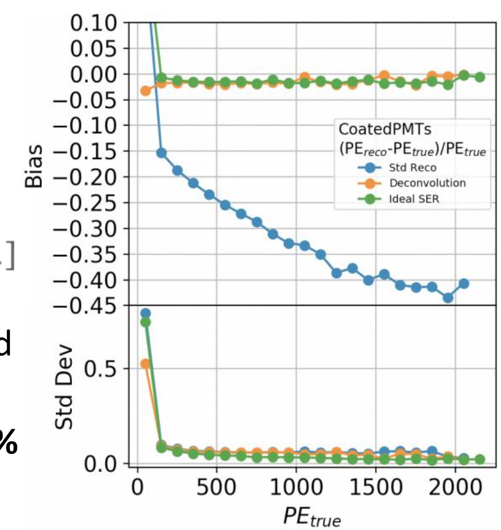
Downstream reconstruction chain:

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

[…] → Detsim OpDetWaveform → Deconvolution OpDetWaveform → Reco1 OpHit → Reco1 OpFlash → […]

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

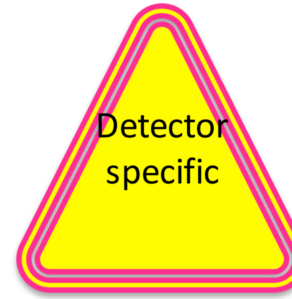
⇒ **Performance: resolution better than ~5% and unbiased at the level of few %**

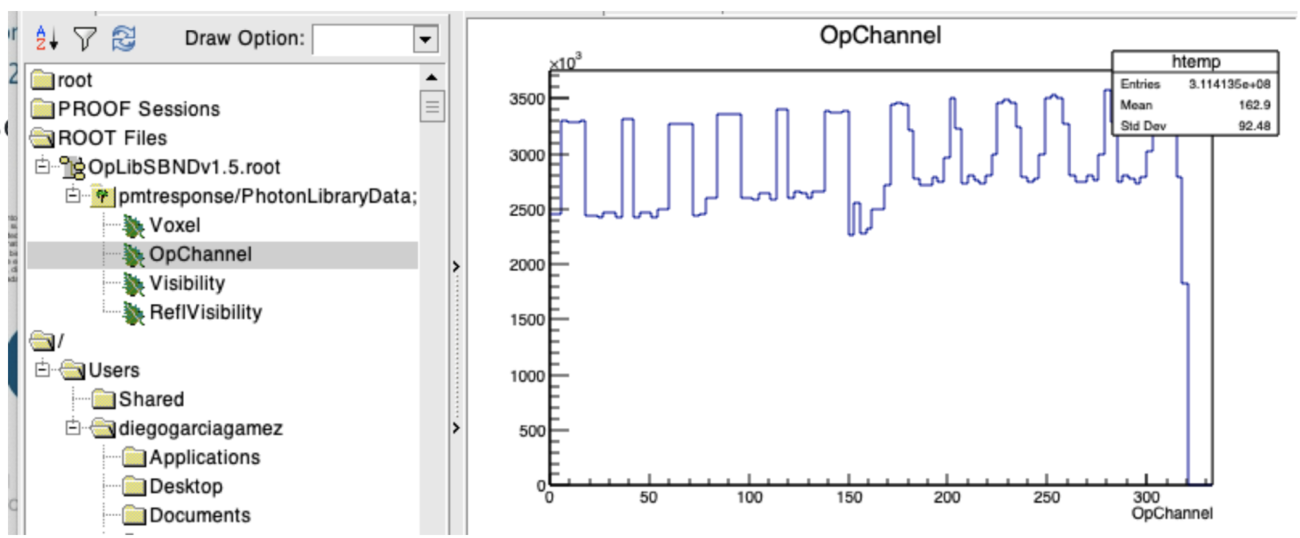


Optical Library parameters: voxelization scheme

sbndcode / sbndcode / LArSoftConfigurations / photpropservices_sbnd.fcl

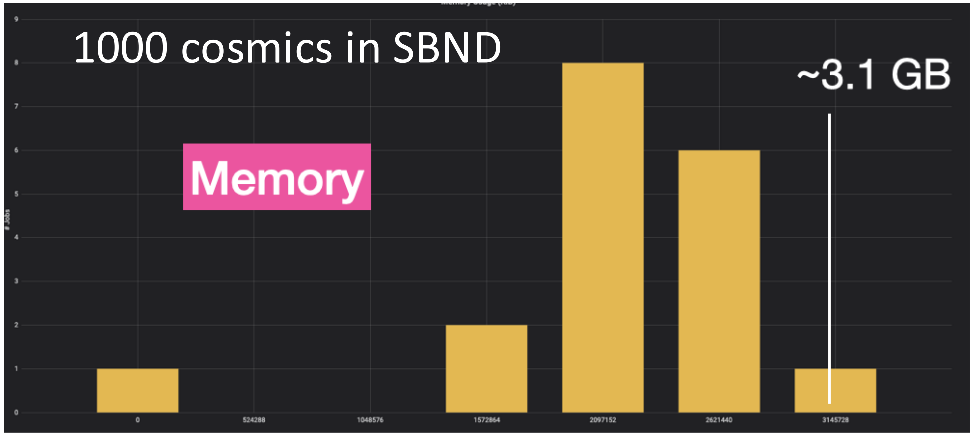
```
# (Re)Defining the Optical Library information/files for the PD-fast HYBRID optical mode
sbnd_library_for_hybrid_mode_photonvisibilityservice:
{
   @table::sbnd_library_vuv_vis_prop_timing_photonvisibilityservice
   LibraryFile: "OpticalLibrary/SBND_OpLibOUT_v2.00.root"
   NX: 66
   NY: 56
   NZ: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

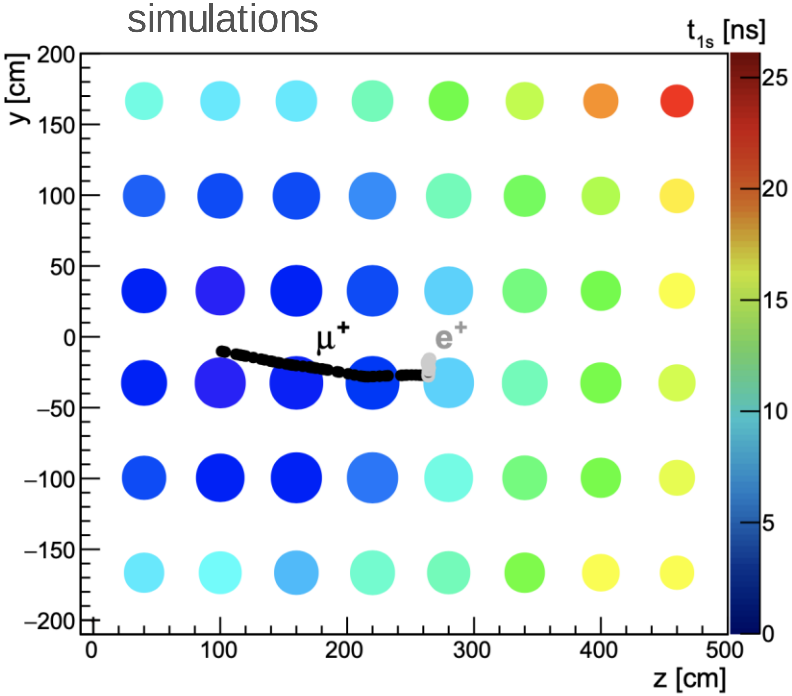


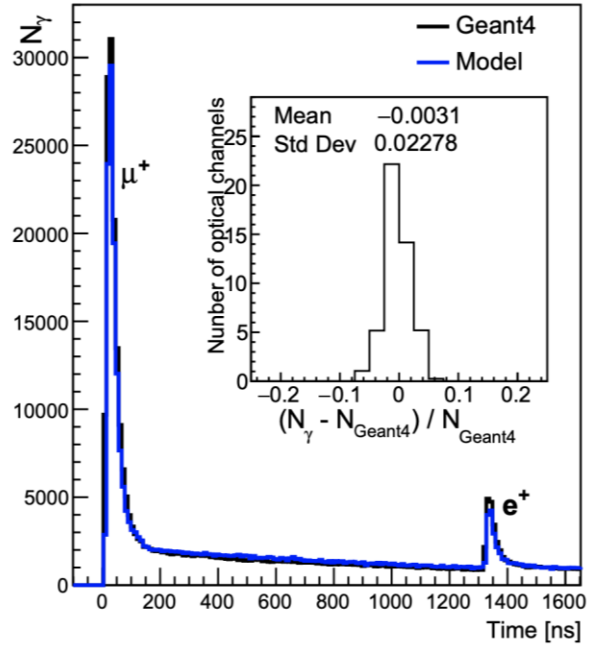


Semi-Analytic model performance

- **Solves the problems of other approaches**
- Photon propagation with no impact on memory (RAM) or simulation (CPU) time
- It models both (N_γ, time)
  - used in SBND and DUNE-SP simulations



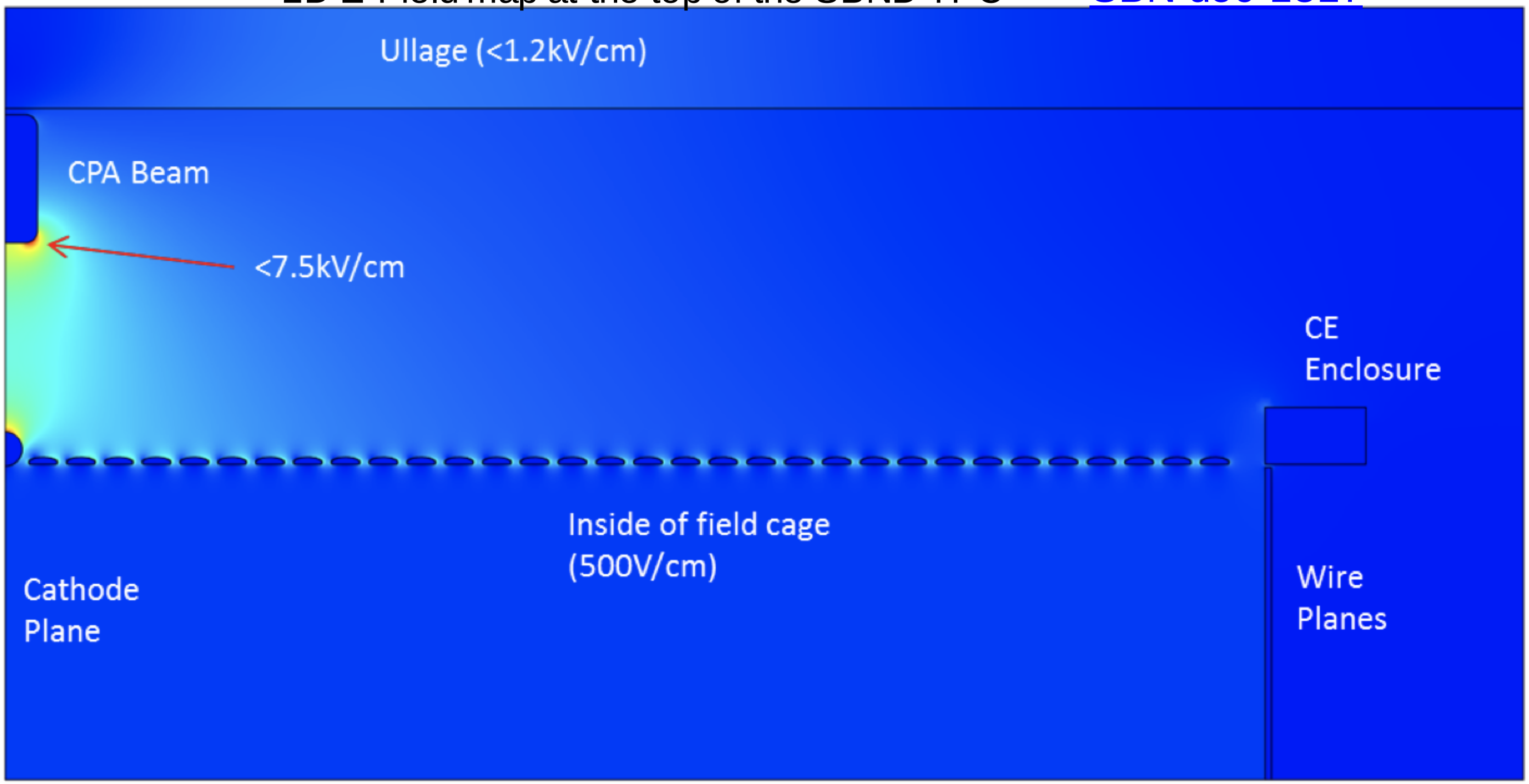




E-Field map in a TPC (SBND case example)

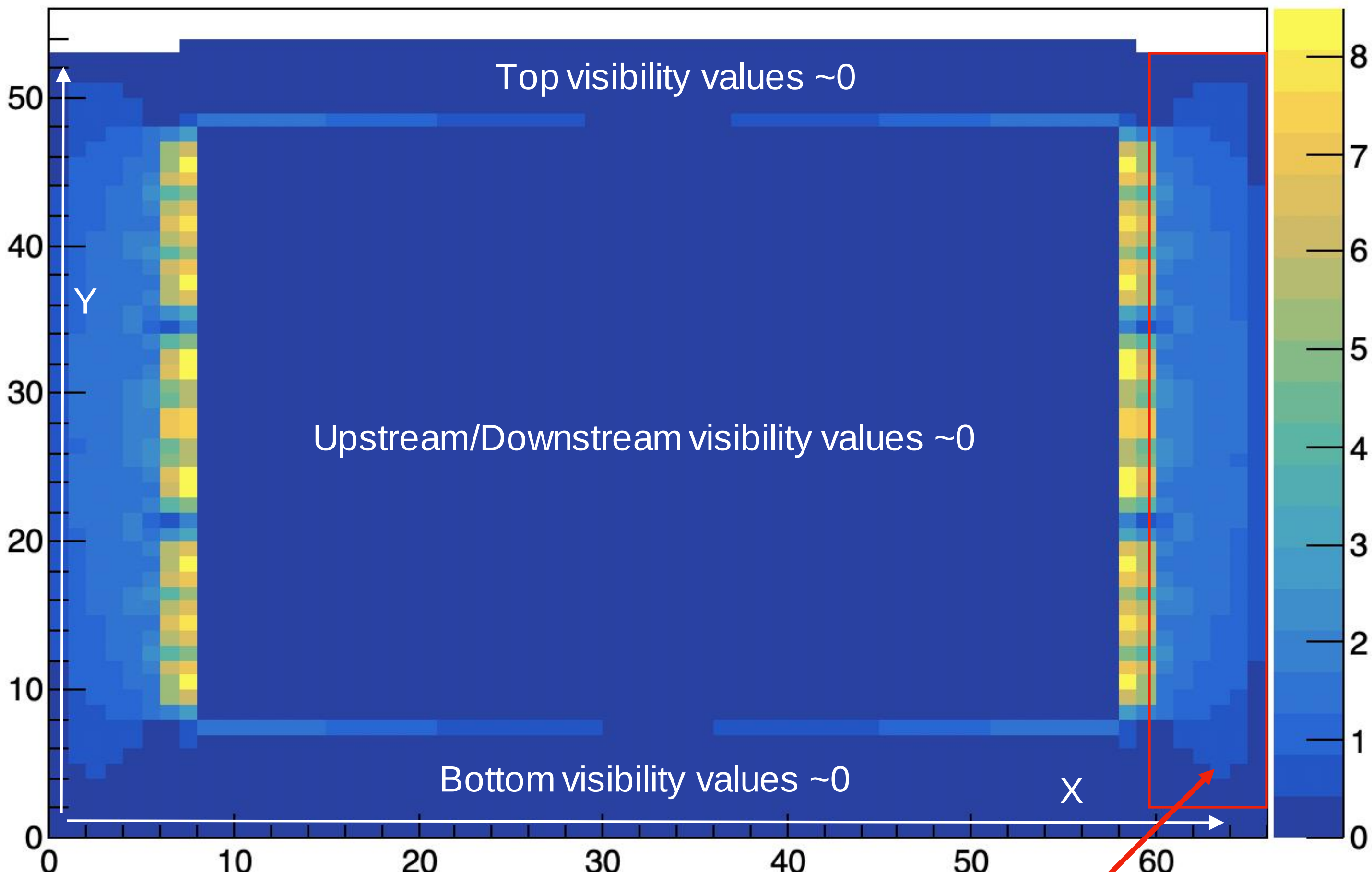
Warning: Light yield strongly depends on the Electric Filed value

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**