SBND: Light Simulation.

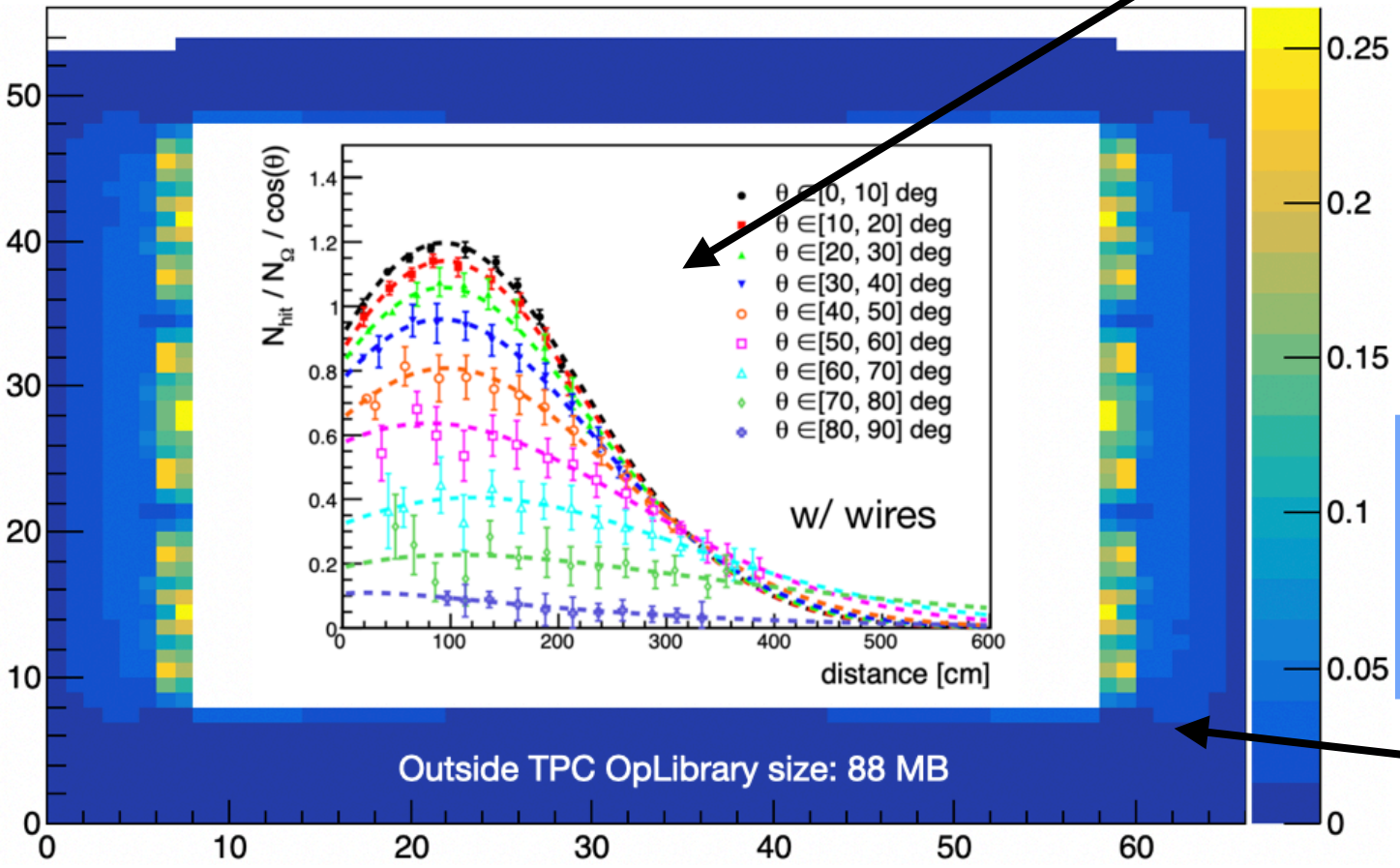
Proposed changes for the Hybrid Model in LArSoft.

Claudia Álvarez García.

Photon Propagation: Hybrid Model motivation.

- Hybrid Model: implementation in the New LArG4 (modular structure).

Semi-Analytic Model: inside Active Volume



SBND case

- Currently used by SBND and DUNE (DUNE-SP and DUNE-VD) but with the Old/Legacy LArG4.
- Photon propagation using corrections to the geometric efficiencies.

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

- By construction valid only inside active volume (based on the solid angle).

Reference: arxiv/2010.00324

Predicting Transport Effects of Scintillation Light Signals in Large-Scale Liquid Argon Detectors

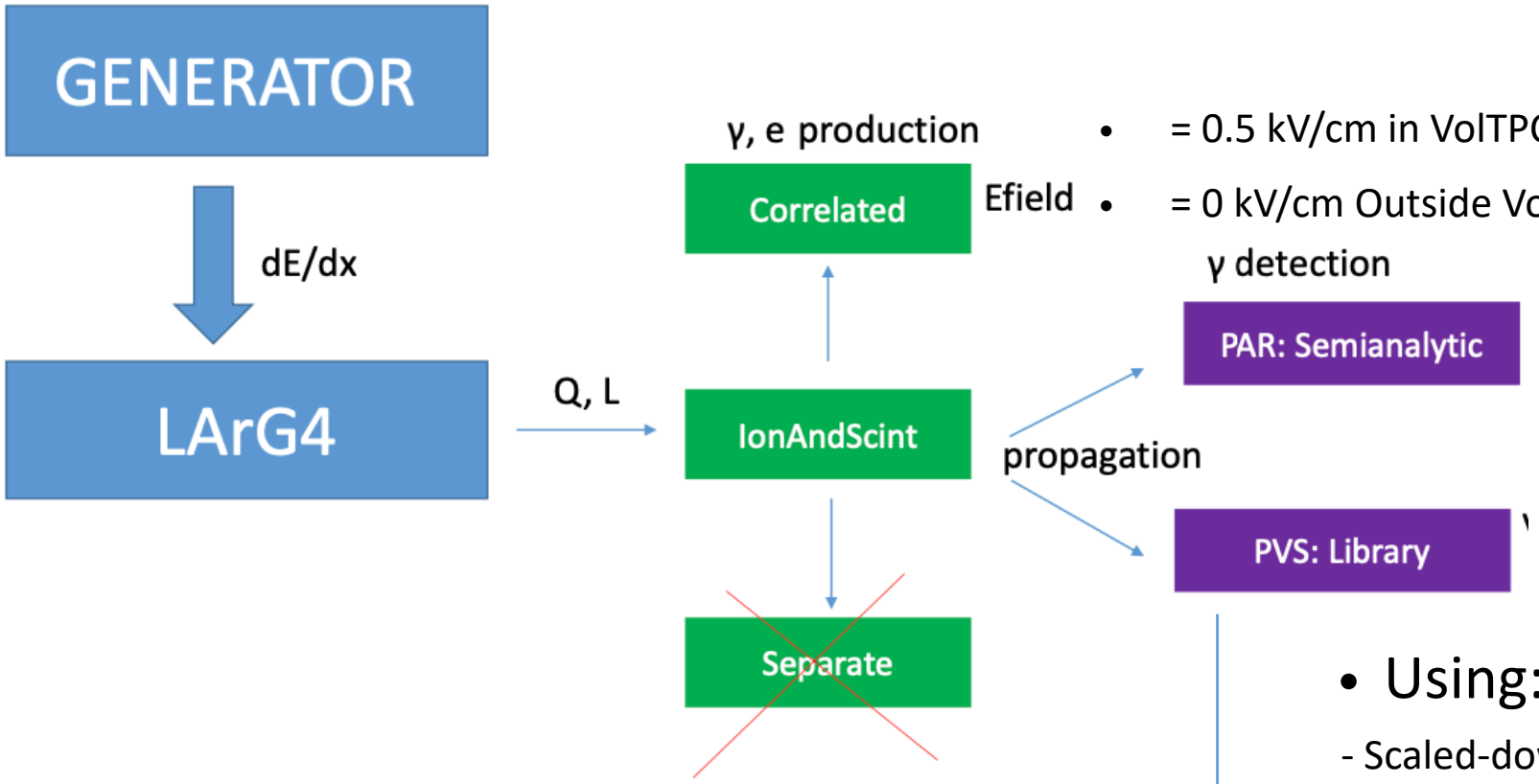
Diego Garcia-Gamez[a,1], Patrick Green[2], Andrzej M. Szelc[b,2]

[1]University of Granada & CAFPE, Campus Fuentenueva, 18002 Granada, Spain
[2]Department of Physics and Astronomy, University of Manchester, Oxford Road, Manchester, M13 9PL, UK

Optical Library: outside Active Volume

- Currently used by Protodune (full detector) with New LArG4.
- Valid inside and outside active volume but serious problems with memory.

Hybrid Model concept in the New LarG4.



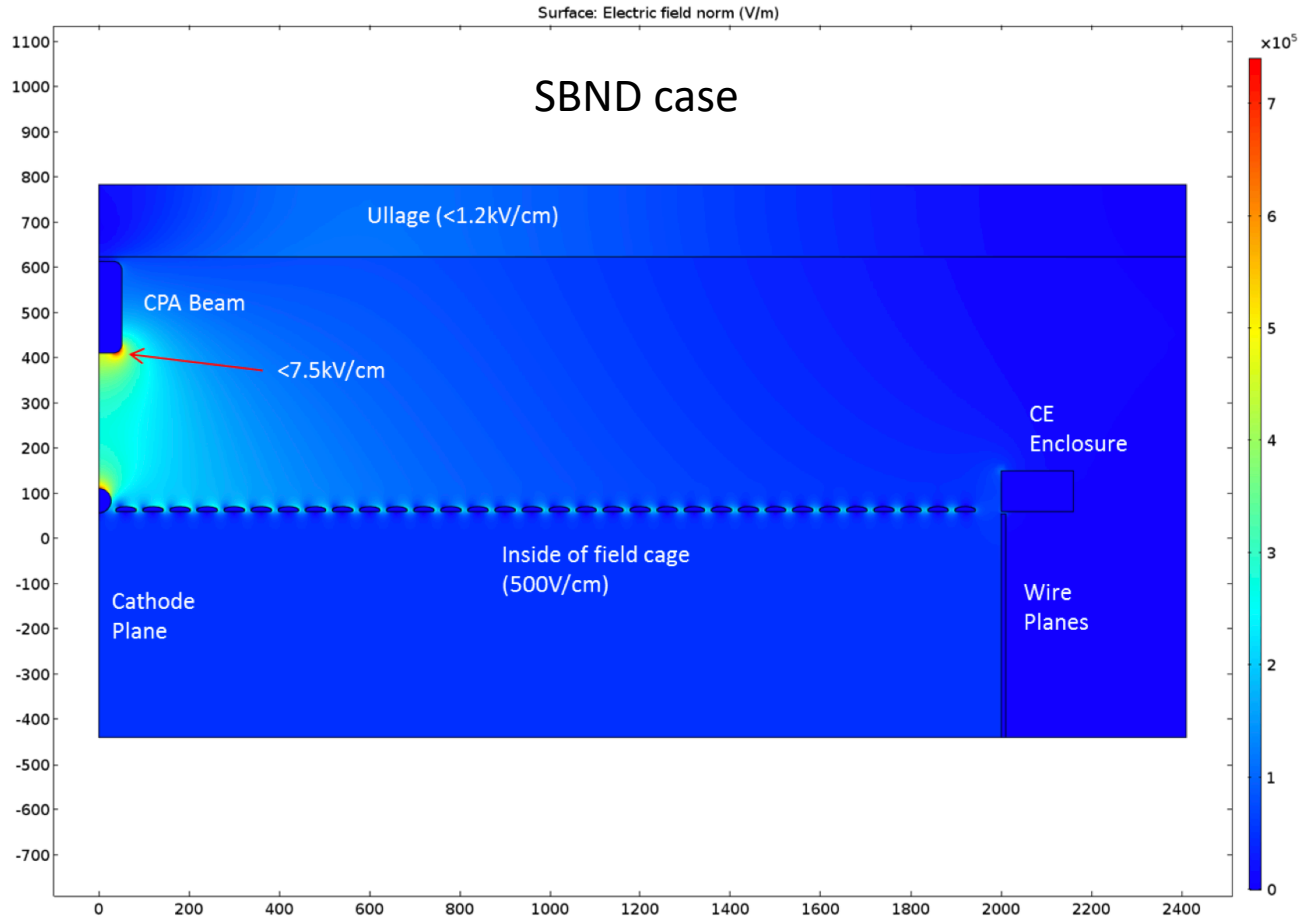
- = 0.5 kV/cm in VolTPCActive **(Already implemented)**
- = 0 kV/cm Outside VolTPCActive **(Minimal LArSoft modifications)**

- VolTPCActive
- Outside VolTPCActive

- Using:

- Scaled-down library (88 MB for SBND): just stores visibilities outside active volume.
- Most updated version by Mu Wei: includes reflected light.

Efield approach behind PMTs.



- Good approach: Efield = 0 kV/cm behind PMTs.

A 2D electric field map at the top of the TPC.

- Inside Active: Efield = 0.5 kV/cm.
- Top side of the Cryostat: Efield ~ 1.2 kV/cm.
- Behind PMTs: Efield ~ 0 kV/cm.

Reference: SBN-doc-1317-v3

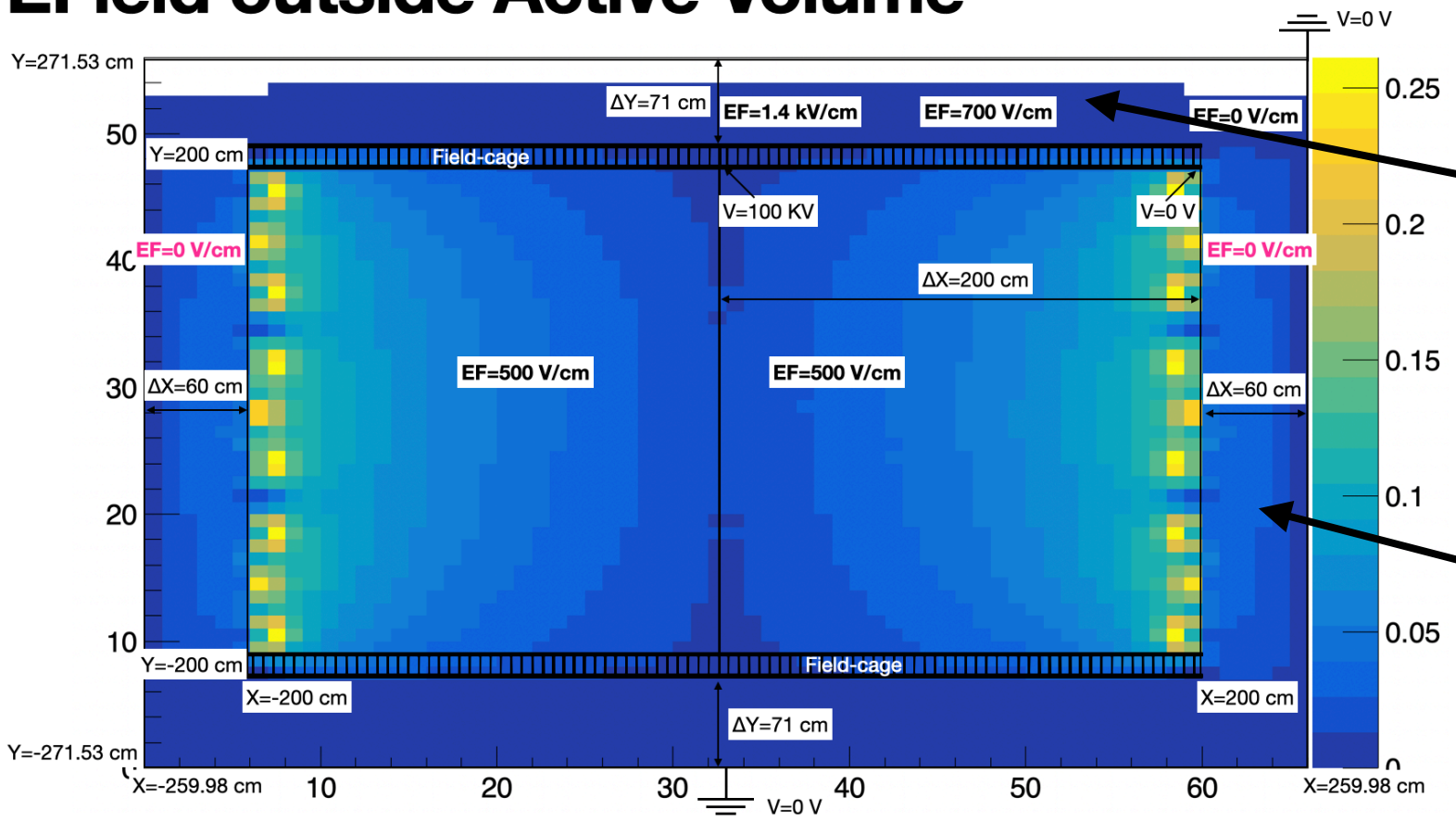
SBND Field Cage Technical Design Report

Bo Yu[1], George Mahler[1], and Serhan Tufanli[2] for the TPC group

[1]Brookhaven National Laboratory
[2]Yale University

October 20, 2016

Efield approach behind PMTs.



Visibilities on the top and bottom side of the cryostat are almost negligible, so including the variable electric field will add little change to the Hybrid Model results.

Behind PMTs visibilities are significant, then we need setting the electric field properly.

- As first approach we can consider Efield = 0 kV/cm in the whole cryostat.

Photon generation (Efield dependence).

$$ph/\mathrm{MeV}(efield) = \left(\frac{1}{19.5 \cdot 10^{-6}\mathrm{MeV}} - \frac{recomb}{23.6 \cdot 10^{-6}\mathrm{MeV}}\right)$$

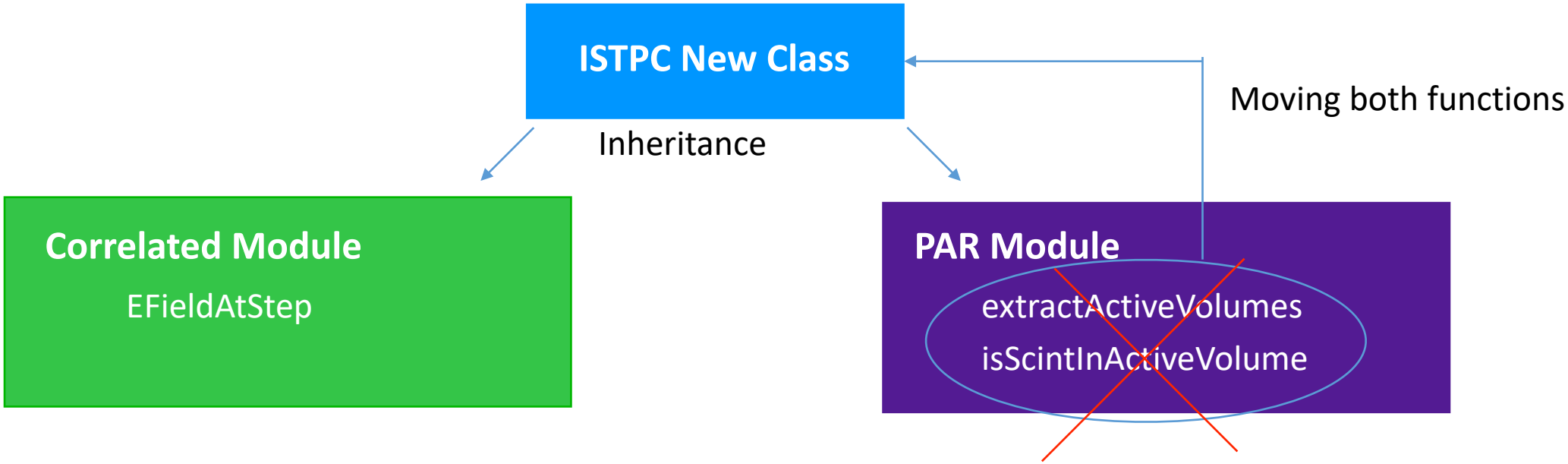
$$\text{recomb} = \frac{\ln(\,0.93 \; + \frac{0.153124\; dE}{0.5\; dx})}{\frac{0.153124\; dE}{0.5\; dx}}$$

- Higher Efield → Less recombination (recomb ↑) → Less photons generated.
- Efield = 0 kV/cm → All electrons recombine (recomb=0) → More photons generated.
  - Using LArQL model: Efield = 0 kV/cm → Escaping electrons do not recombine (recomb ↑)

$$ph/\mathrm{MeV}(efield = 0) = \frac{1}{19.5 \cdot 10^{-6}\mathrm{MeV}} = 51282.05$$

Moving physics concepts into modules.

- In order to set the electric field to zero outside the active volume we need two additional functions so as to distinguish between Active and Non-Active volume.
- These functions were defined in the PAR module (at photon propagation stage).
- For the Hybrid Model we also need them in a previous stage (where photons are generated, in the EfFieldAtStep function of the Correlated module) → Create a new class.



- Both modules are able to inherit the new class.

LArSoft-larsim/IonizationScintillation/ISCalcCorrelated.

- Changes in: IonizationScintillation/ISCalcCorrelated.cxx
- Modifying the function EFieldAtStep.
  - If the scintillation point is not in Active Volume set Efield= 0 kV/cm.
  - Else (if it is in Active Volume) returns Efield value (0.5 kV/cm).

```
double
ISCalcCorrelated::EFieldAtStep(double efield, sim::SimEnergyDeposit const& edep)
{
  //electric field outside active volume set to zero
  if(isScintInActiveVolume(edep.MidPoint())==false) {
    efield = 0.0;
    return efield;
  }
  else { //electric field in active volume
    if (!fSCE->EnableSimEfieldSCE()) return efield;
    auto const eFieldOffsets = fSCE->GetEfieldOffsets(edep.MidPoint());
    return efield * std::hypot(1 + eFieldOffsets.X(), eFieldOffsets.Y(), eFieldOffsets.Z());
  }
}
//
```

Added

Current Version

- If Efield = 0, recombination factor = 0 (+ LArQL correction).
- If Efield ≠ 0, calculate recombination factor.

```
double EFieldStep = EFieldAtStep(detProp.Efield(), edep);
double recomb = 0.;


// calculate recombination survival fraction value inside an EField, otherwise zero

if(EFieldStep > 0) {
// Guard against spurious values of dE/dx. Note: assumes density of LAr
if (dEdx < 1.) dEdx = 1.;

// 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);
}
}//Efield

if(fUseModLarqlRecomb){ //Use corrections from LArQL model
  recomb += EscapingEFraction(dEdx)*FieldCorrection(EFieldStep, dEdx); //Correction for low EF
}
```

Added

LArQL Model

LArSoft-larsim/IonizationScintillation new class definition: ISTPC.

- IonizationScintillation/ISTPC.h

```
#ifndef IS_ISTPC_H
#define IS_ISTPC_H


#include "larcore/Geometry/Geometry.h"
#include "larcorealg/CoreUtils/counter.h"
#include "larcorealg/CoreUtils/enumerate.h"

#include "messagefacility/MessageLogger/MessageLogger.h"

namespace larg4 {
  class ISTPC {
  public:

    ISTPC();
    std::vector<geo::BoxBoundedGeo> fActiveVolumes;

     bool isScintInActiveVolume(geo::Point_t const& ScintPoint);

    static std::vector<geo::BoxBoundedGeo> extractActiveVolumes(geo::GeometryCore const& geom);


  };
}
#endif
```

LArSoft-larsim/IonizationScintillation new class definition: ISTPC.

- IonizationScintillation/ISTPC.cxx

```
#include "larsim/IonizationScintillation/ISTPC.h"
#include "larcore/CoreUtils/ServiceUtil.h"

namespace larg4 {
  //----------------------------------------------------------------------
  ISTPC::ISTPC()
  {
    std::cout << "IonizationAndScintillation/ISTPC Initialize." << std::endl;
    std::cout << "Initializing the geometry of the detector." << std::endl;
    geo::GeometryCore const& geom = *(lar::providerFrom<geo::Geometry>());

    fActiveVolumes = extractActiveVolumes(geom);
    {
      auto log = mf::LogTrace("IonAndScint") << "IonAndScint: active volume boundaries from
                                             << fActiveVolumes.size() << " volumes:";
      for (auto const& [iCryo, box] : util::enumerate(fActiveVolumes)) {
        log << "\n - C:" << iCryo << ": " << box.Min() << " -- " << box.Max() << " cm";
      }
    } // local scope
  }


  //---------------------------------------


 bool
 ISTPC::isScintInActiveVolume(geo::Point_t const& ScintPoint)
 {
   //semi-analytic approach only works in the active volume
   return fActiveVolumes[0].ContainsPosition(ScintPoint);
 }
//----------------------------------------------------------------------
```

```
std::vector<geo::BoxBoundedGeo>
ISTPC::extractActiveVolumes(geo::GeometryCore const& geom)
{
  std::vector<geo::BoxBoundedGeo> activeVolumes;
  activeVolumes.reserve(geom.Ncryostats());

  for (geo::CryostatGeo const& cryo : geom.IterateCryostats()) {

    // can't use it default-constructed since it would always include origin

    geo::BoxBoundedGeo box{cryo.TPC(0).ActiveBoundingBox()};

    for (geo::TPCGeo const& TPC : cryo.IterateTPCs())
      box.ExtendToInclude(TPC.ActiveBoundingBox());

    activeVolumes.push_back(std::move(box));

  } // for cryostats

 return activeVolumes;
}

}
```

LArSoft-larsim/IonizationScintillation/ISCalcCorrelated.

- Minimum change in: IonizationScintillation/ISCalcCorrelated.h

```
#ifndef IS_ISCALCCORRELATED_H
#define IS_ISCALCCORRELATED_H

#include "larsim/IonizationScintillation/ISCalc.h"
#include "larsim/IonizationScintillation/ISTPC.h"

#include "lardata/DetectorInfoServices/DetectorPropertiesService.h"
#include "lardata/DetectorInfoServices/LArPropertiesService.h"
#include "larevt/SpaceChargeServices/SpaceChargeService.h"
#include "larsim/Simulation/LArG4Parameters.h"

#include "larcoreobj/SimpleTypesAndConstants/PhysicalConstants.h"
#include "larcoreobj/SimpleTypesAndConstants/geo_vectors.h"
#include <vector>

#include "CLHEP/Vector/ThreeVector.h"
#include "messagefacility/MessageLogger/MessageLogger.h"

namespace larg4 {
  class ISCalcCorrelated : public ISTPC, public ISCalc{
  public:
    ISCalcCorrelated(detinfo::DetectorPropertiesData const& detProp);
```

LArSoft-larsim/IonizationScintillation/CMakeLists.txt.

- Required change in: IonizationScintillation/CMakeLists.txt

```
art_make(LIB_LIBRARIES
           larsim_Simulation
           lardataobj_Simulation
           lardataalg_DetectorInfo
           larevt_SpaceCharge
           nusimdata_SimulationBase
           larcorealg_Geometry
           larcore_Geometry_Geometry_service
           ${CLHEP}
           ${FHICLCPP}
           ${MF_MESSAGELOGGER}
         MODULE_LIBRARIES
           larsim_IonizationScintillation
           larsim_Simulation
           lardataobj_Simulation
           nurandom_RandomUtils_NuRandomService_service
           larcorealg_Geometry
           larcore_Geometry_Geometry_service
           ${ART_FRAMEWORK_SERVICES_REGISTRY}
           ${ART_ROOT_IO_TFILE_SUPPORT}
           ${ART_ROOT_IO_TFILESERVICE_SERVICE}
           ${CLHEP}
           ROOT::Core
           ROOT::Tree
           ${MF_MESSAGELOGGER})

install_headers()
install_fhicl()
install_source()
```

LArSoft-larsim/Photonpropagation/ PDFastSimPAR_module.

- Reading the functions from ISTPC new class.
  - Inherit the new class.

```
// LArSoft libraries
//#include "larcore/Geometry/Geometry.h"
//#include "larcorealg/CoreUtils/counter.h"
//#include "larcorealg/CoreUtils/enumerate.h"

// New include

#include "larsim/IonizationScintillation/ISTPC.h"

namespace phot {
  class PDFastSimPAR : public larg4::ISTPC,  public art::EDProducer {
  public:
```

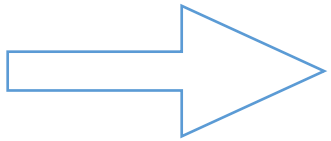
  - Remove private members that are now inherited.

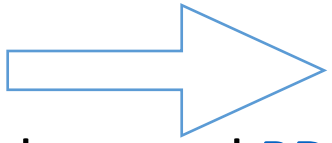
```
// std::vector<geo::BoxBoundedGeo> fActiveVolumes;

//bool isScintInActiveVolume(geo::Point_t const& ScintPoint);

//static std::vector<geo::BoxBoundedGeo> extractActiveVolumes(geo::GeometryCore const& geom);
```

  - Remove: isScintInActiveVolume and extractActiveVolumes functions.
  - PDFastSimPAR::Initialization() `//fActiveVolumes =extractActiveVolumes(geom);`
  - PDFastSimPAR::produce(art::Event& event): Cross-check.

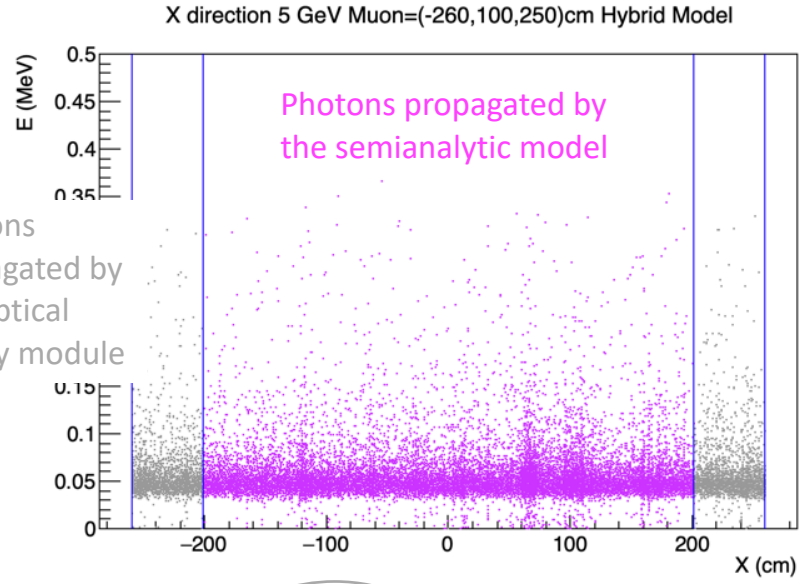
```
//Double cross-check for the hybrid model
if (fOnlyActiveVolume && !isScintInActiveVolume(ScintPoint)) continue;
```

Summary: Changes in sbndcode and larsim.

- sbndcode:
  - PDFastSim_sbnd.fcl
  - standard_g4_refactored_sbnd.fcl.  New-LArG4 Modular structure.
  - New Geometry (v02_00)
  - New Optical Library file

- larsim:
  - ISTPC new class  Proposal.
  - Minimal change in ISCalcCorrelated.cxx and PDFastSimPAR_module.cc

Testing the Hybrid Model (LArSoft v09_22_02). Energy depositions for a single muon crossing the 3 dimensions.



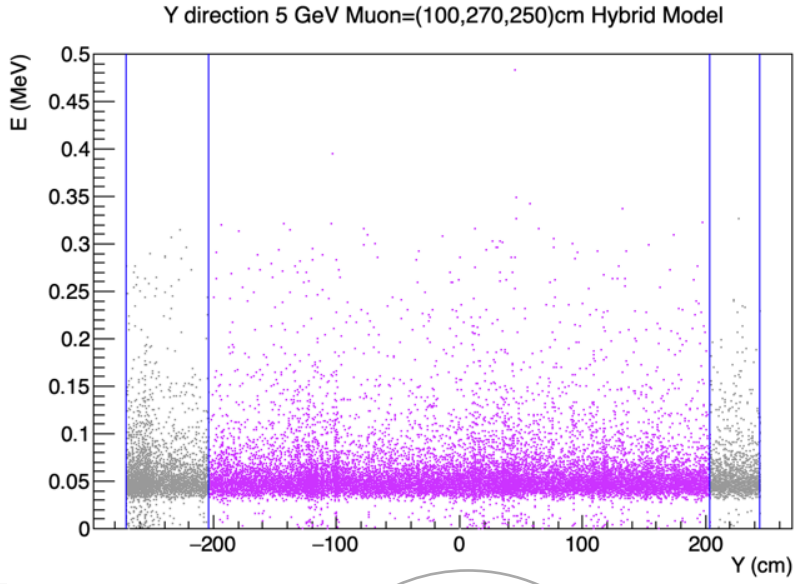
Photons propagated by the optical library module

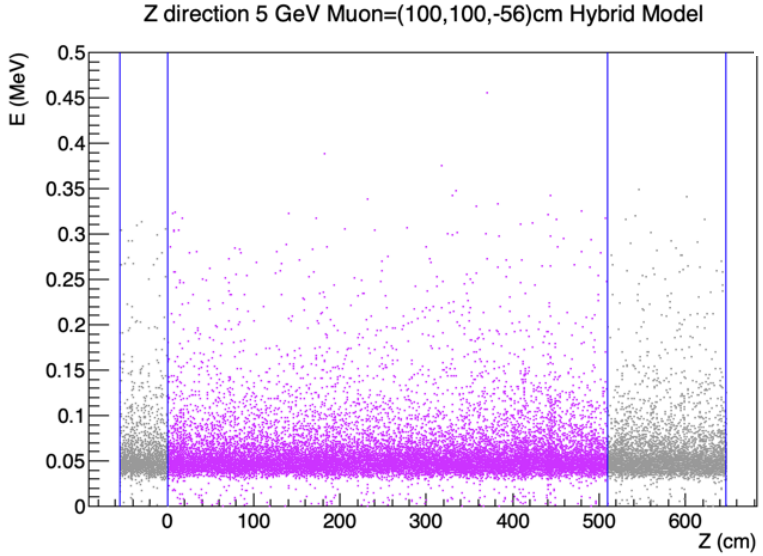
Cryostat Boundaries.

X [-259.98 , 259.98] cm

Y [-271.53, 244.38] cm

Z [-55.4 , 647.3] cm





Adding the new class doesn't affect simulation time or memory → Backup

Expected result!
We are able to cover all the cryostat volume with energy depositions.

Conclusions.

- 1- We have developed the Hybrid model for the Light Simulation/Propagation in the New-LArG4 (tested for SBND case).
    - Semianalytic in the Active Volume.
    - New scaled-down Optical Library outside the Active Volume.
- 2- Minimal modifications needed in larsim package: our proposal is to include in EFieldAtStep() function the possibility to distinguish between Active (EField=0.5 kV/cm) and Non-Active (EField=0 kV/cm as a first approach) volume through extractActiveVolumes and isScintInActiveVolume functions.
- 3- Functions defined in a new class: ISTPC, inherited by ISCalcCorrelated and PDFastSimPAR_module.
- 4- Pull Request: It is not advisable to use multiple inheritance because it can cause art problems.

2nd Proposal. Avoiding multiple inheritance: ISTPC as a Singleton class.

- IonizationScintillation/ISTPC.h

```
#ifndef IS_ISTPC_H
#define IS_ISTPC_H


#include "larcore/Geometry/Geometry.h"
#include "larcorealg/CoreUtils/counter.h"
#include "larcorealg/CoreUtils/enumerate.h"

#include "messagefacility/MessageLogger/MessageLogger.h"

//Define A Singleton Class

  class ISTPC {
  public:

    //Delete copy constructor
    ISTPC(const ISTPC&)=delete;

    static ISTPC& Get()
    {
      static ISTPC instance;
      return instance;
    }


     static std::vector<geo::BoxBoundedGeo> extractActiveVolumes(geo::GeometryCore const& geom){ return Get().IextractActiveVolumes(geom); }
     static bool isScintInActiveVolume(geo::Point_t const& ScintPoint){ return Get().IisScintInActiveVolume(ScintPoint); }


    private:

    std::vector<geo::BoxBoundedGeo> fActiveVolumes;

    //Constructor
    ISTPC(){

      std::cout << "IonizationAndScintillation/Initializing ISTPC." << std::endl;
      std::cout << "Initializing the geometry of the detector." << std::endl;
      geo::GeometryCore const& geom = *(lar::providerFrom<geo::Geometry>());

      fActiveVolumes = IextractActiveVolumes(geom);


      {
        auto log = mf::LogTrace("IonAndScint") << "IonAndScint: active volume boundaries from "
                                               << fActiveVolumes.size() << " volumes:";
        for (auto const& [iCryo, box] : util::enumerate(fActiveVolumes)) {
          log << "\n - C:" << iCryo << ": " << box.Min() << " -- " << box.Max() << " cm";
        }
      } // local scope
    }

    //Internal functions
    bool IisScintInActiveVolume(geo::Point_t const& ScintPoint);
    std::vector<geo::BoxBoundedGeo> IextractActiveVolumes(geo::GeometryCore const& geom);
  };

#endif
```

IonizationScintillation/ISTPC.cxx

```
#include "larsim/IonizationScintillation/ISTPC.h"
#include "larcore/CoreUtils/ServiceUtil.h"


  //----------------------------------------------------------------------------
  //Define internal functions
 bool
 ISTPC::IisScintInActiveVolume(geo::Point_t const& ScintPoint)
 {
   //semi-analytic approach only works in the active volume
   return fActiveVolumes[0].ContainsPosition(ScintPoint);
 }
//----------------------------------------------------------------------------

std::vector<geo::BoxBoundedGeo>
ISTPC::IextractActiveVolumes(geo::GeometryCore const& geom)
{
  std::vector<geo::BoxBoundedGeo> activeVolumes;
  activeVolumes.reserve(geom.Ncryostats());

  for (geo::CryostatGeo const& cryo : geom.IterateCryostats()) {

    // can't use it default-constructed since it would always include origin

    geo::BoxBoundedGeo box{cryo.TPC(0).ActiveBoundingBox()};

    for (geo::TPCGeo const& TPC : cryo.IterateTPCs())
      box.ExtendToInclude(TPC.ActiveBoundingBox());

    activeVolumes.push_back(std::move(box));

  } // for cryostats

 return activeVolumes;
}
```

ISCalcCorrelated and PDFastSimPAR_module.

- ISCalcCorrelated.h: Delete ISTPC inheritance.
- ISCalcCorrelated.cxx:

```
double
ISCalcCorrelated::EFieldAtStep(double efield, sim::SimEnergyDeposit const& edep)
{
  //electric field outside active volume set to zero

  if(ISTPC::isScintInActiveVolume(edep.MidPoint())==false) {
    efield = 0.0;
    return efield;
  }
  else{ //electric field in active volume

    if (!fSCE->EnableSimEfieldSCE()) return efield;
    auto const eFieldOffsets = fSCE->GetEfieldOffsets(edep.MidPoint());
    return efield * std::hypot(1 + eFieldOffsets.X(), eFieldOffsets.Y(), eFieldOffsets.Z());
  }
}
```

```
#ifndef IS_ISCALCCORRELATED_H
#define IS_ISCALCCORRELATED_H

#include "larsim/IonizationScintillation/ISCalc.h"
#include "larsim/IonizationScintillation/ISTPC.h"

#include "lardata/DetectorInfoServices/DetectorPropertiesService.h"
#include "lardata/DetectorInfoServices/LArPropertiesService.h"
#include "larevt/SpaceChargeServices/SpaceChargeService.h"
#include "larsim/Simulation/LArG4Parameters.h"

#include "larcoreobj/SimpleTypesAndConstants/PhysicalConstants.h"
#include "larcoreobj/SimpleTypesAndConstants/geo_vectors.h"
#include <vector>

#include "CLHEP/Vector/ThreeVector.h"
#include "messagefacility/MessageLogger/MessageLogger.h"

namespace larg4 {
  class ISCalcCorrelated : public ISTPC, public ISCalc{
  public:
    ISCalcCorrelated(detinfo::DetectorPropertiesData const& detProp);
```

- PDFastSimPAR_module.cc: Delete ISTPC inheritance.

```
std::vector<geo::BoxBoundedGeo> fActiveVolumes;

fActiveVolumes = ISTPC::extractActiveVolumes(geom);
```

```
if (fOnlyActiveVolume && !ISTPC::isScintInActiveVolume(ScintPoint)) continue;
```

3rd Proposal. Avoiding multiple inheritance: ISTPC as a Singleton class.

- IonizationScintillation/ISTPC.h

```
#ifndef IS_ISTPC_H
#define IS_ISTPC_H


#include "larcore/Geometry/Geometry.h"
#include "larcorealg/CoreUtils/counter.h"
#include "larcorealg/CoreUtils/enumerate.h"

#include "messagefacility/MessageLogger/MessageLogger.h"

//Define A Singleton Class

  class ISTPC {
  public:

    //Delete copy constructor
    ISTPC(const ISTPC&)=delete;

    static ISTPC& Get()
    {
      static ISTPC instance;
      return instance;
    }

     static std::vector<geo::BoxBoundedGeo> fActiveVolumesfunc(){return Get().IfActiveVolumesfunc();}
     // static std::vector<geo::BoxBoundedGeo> extractActiveVolumes(geo::GeometryCore const& geom){ return Get().IextractActiveVolumes(geom); }
     static bool isScintInActiveVolume(geo::Point_t const& ScintPoint){ return Get().IisScintInActiveVolume(ScintPoint); }


    private:

     std::vector<geo::BoxBoundedGeo> IfActiveVolumes;

    //Constructor
    ISTPC(){

      std::cout << "IonizationAndScintillation/Initializing ISTPC." << std::endl;
      std::cout << "Initializing the geometry of the detector." << std::endl;
      geo::GeometryCore const& geom = *(lar::providerFrom<geo::Geometry>());

      IfActiveVolumes = IextractActiveVolumes(geom);


       {
        auto log = mf::LogTrace("IonAndScint") << "IonAndScint: active volume boundaries from "
                                               << IfActiveVolumes.size() << " volumes:";
        for (auto const& [iCryo, box] : util::enumerate(IfActiveVolumes)) {
        log << "\n - C:" << iCryo << ": " << box.Min() << " -- " << box.Max() << " cm";
        }
       } // local scope

       }

    //Internal functions
    std::vector<geo::BoxBoundedGeo> IfActiveVolumesfunc(){return IfActiveVolumes;}
    bool IisScintInActiveVolume(geo::Point_t const& ScintPoint);
    std::vector<geo::BoxBoundedGeo> IextractActiveVolumes(geo::GeometryCore const& geom);
  };

#endif
```

IonizationScintillation/ISTPC.cxx

```
#include "larsim/IonizationScintillation/ISTPC.h"
#include "larcore/CoreUtils/ServiceUtil.h"


  //----------------------------------------------------------------------------
  //Define internal functions
 bool
 ISTPC::IisScintInActiveVolume(geo::Point_t const& ScintPoint)
 {
   //semi-analytic approach only works in the active volume
   return IfActiveVolumes[0].ContainsPosition(ScintPoint);
 }
//----------------------------------------------------------------------------

std::vector<geo::BoxBoundedGeo>
ISTPC::IextractActiveVolumes(geo::GeometryCore const& geom)
{
  std::vector<geo::BoxBoundedGeo> activeVolumes;
  activeVolumes.reserve(geom.Ncryostats());

  for (geo::CryostatGeo const& cryo : geom.IterateCryostats()) {

    // can't use it default-constructed since it would always include origin

    geo::BoxBoundedGeo box{cryo.TPC(0).ActiveBoundingBox()};

    for (geo::TPCGeo const& TPC : cryo.IterateTPCs())
      box.ExtendToInclude(TPC.ActiveBoundingBox());

    activeVolumes.push_back(std::move(box));

  } // for cryostats

 return activeVolumes;
}
```

ISCalcCorrelated and PDFastSimPAR_module

- ISCalcCorrelated.h: Delete ISTPC inheritance.
- ISCalcCorrelated.cxx:

```
double
ISCalcCorrelated::EFieldAtStep(double efield, sim::SimEnergyDeposit const& edep)
{
  //electric field outside active volume set to zero

  if(ISTPC::isScintInActiveVolume(edep.MidPoint())==false) {
    efield = 0.0;
    return efield;
  }
  else{ //electric field in active volume

    if (!fSCE->EnableSimEfieldSCE()) return efield;
    auto const eFieldOffsets = fSCE->GetEfieldOffsets(edep.MidPoint());
    return efield * std::hypot(1 + eFieldOffsets.X(), eFieldOffsets.Y(), eFieldOffsets.Z());
  }
}
```

```
#ifndef IS_ISCALCCORRELATED_H
#define IS_ISCALCCORRELATED_H

#include "larsim/IonizationScintillation/ISCalc.h"
#include "larsim/IonizationScintillation/ISTPC.h"

#include "lardata/DetectorInfoServices/DetectorPropertiesService.h"
#include "lardata/DetectorInfoServices/LArPropertiesService.h"
#include "larevt/SpaceChargeServices/SpaceChargeService.h"
#include "larsim/Simulation/LArG4Parameters.h"

#include "larcoreobj/SimpleTypesAndConstants/PhysicalConstants.h"
#include "larcoreobj/SimpleTypesAndConstants/geo_vectors.h"
#include <vector>

#include "CLHEP/Vector/ThreeVector.h"
#include "messagefacility/MessageLogger/MessageLogger.h"

namespace larg4 {
  class ISCalcCorrelated : public ISTPC, public ISCalc{
  public:
    ISCalcCorrelated(detinfo::DetectorPropertiesData const& detProp);
```

- PDFastSimPAR_module.cc: Delete ISTPC inheritance.

```
std::vector<geo::BoxBoundedGeo> fActiveVolumes;

fActiveVolumes = ISTPC::fActiveVolumesfunc();
```

```
if (fOnlyActiveVolume && !ISTPC::isScintInActiveVolume(ScintPoint)) continue;
```

4th Proposal. Avoiding multiple inheritance: ISTPC object in ISCalcCorrelated and PDFastSimPAR_module.

- IonizationScintillation/ISTPC.h

```
class ISTPC {
public:
  explicit ISTPC(geo::GeometryCore const& geom);
  bool isScintInActiveVolume(geo::Point_t const& ScintPoint) const;

private:
  std::vector<geo::BoxBoundedGeo> fActiveVolumes;
  static std::vector<geo::BoxBoundedGeo> extractActiveVolumes(...);
};
```

IonizationScintillation/ISTPC.cxx

```
+ #include "larcorealg/CoreUtils/counter.h"
+ #include "larcorealg/CoreUtils/enumerate.h"
+
+ #include "messagefacility/MessageLogger/MessageLogger.h"
```

```
ISTPC::ISTPC(geo:GeometryCore const& geom)
  : fActiveVolumes{extractActiveVolumes(geom)}
{ ... }
```

- Private data member of ISTPC class in ISCalcCorrelated and PDFastSimPAR_module

```
class PDFastSimPAR : public art::EDProducer {
public:
  ...
private:
  larg4::ISTPC fISTPC;
  ...
};
```

Create this object in both modules

fISTPC.isScintInActiveVolume

fISTPC.extractActiveVolumes

```
fActiveVolumes = extractActiveVolumes(geom);

fcathode_centre = {geom.TPC(0, 0).GetCathodeCenter().X(),
                   fActiveVolumes[0].CenterY(),
                   fActiveVolumes[0].CenterZ()};
```

Backup slides

SBND Hybrid Light Simulation

Comparing Hybrid Model: Simulation time and memory.

- 1- Preliminary Hybrid Model: Functions defined in PDFastSimPAR_module.cc and redefined in ISCalcCorrelated.cxx.

- 2- First new class version: Functions defined in PDFastSimPAR_module.cc. ISCalcCorrelated.cxx inheriting the functions from ISTPC class.

- 3- Final new class version: PDFastSimPAR_module.cc and ISCalcCorrelated.cxx both inheriting the functions from ISTPC class.
- 4- PR: Avoid multiple inheritance:
  - 4.1- Singleton class version.
  - 4.2- Creating ISTPC object.

Comparing Hybrid Model: Simulation time and memory.

- 10 crossing muon sample: E=5 GeV (100 cm, 270 cm, 250 cm), not using LArQL Model.

	1- Preliminary Version	2- Mid Version	3- Final Version
TimeReport CPU	138.452446 s	143.629327 s	137.939001 s
TimeReport Real	139.441646 s	144.619775 s	141.595928 s
MemReport VmPeak [base-10 MB]	2476.13 MB	2476.13 MB	2476.09 MB
VmHWM	1437.11 MB	1450.43 MB	1440.55 MB

- Reasonable values.

Tools for Hybrid Model.

- Adding SBND New Geometry
  - sbnd_v02_00_nowires.gdml.
  - sbnd_v02_00.gdml.

  - (Thanks to Gustavo Valdiviesso).

- Semianalytic: volTPCActive

- Library: volCryostat, volFieldCage, volTPCPlaneVert, volPMT, volModuleXArapuca, volPDSstructure.

```
<structure>
        <volume name="volCryostat">
                <materialref ref="LAr"/>
                <solidref ref="boxCryostat"/>
                <auxiliary auxtype="SensDet" auxvalue="SimEnergyDeposit"/>
                <auxiliary auxtype="StepLimit" auxvalue="0.01"/>
                <auxiliary auxtype="Efield" auxvalue="0."/>
```

- PAR: Most updated version.

- PVS: Most updated version. Using New Library (scaled-down library SBND_OpLibOUT_v2.00.root: 88 MB).

G4 fhicl file refactored.

- sbndcode/LArSoftConfigurations/ PDFastSim_sbnd.fcl:

```
#include "PDFastSimPAR.fcl"
#include "PDFastSimPVS.fcl"

#include "opticalsimparameterisations_sbnd.fcl"

BEGIN_PROLOG

############
# SBND #
############

# Semi-analytic model hits & timing parameterization for SBND, Ar scintillation:

# standard configuration
sbnd_pdfastsim_par:                     @local::standard_pdfastsim_par_ar

sbnd_pdfastsim_par.SimulationLabel:     "IonAndScintIN"
# Direct (VUV)
sbnd_pdfastsim_par.VUVTiming:           @local::sbnd_vuv_timing_parameterization
sbnd_pdfastsim_par.VUVHits:             @local::sbnd_vuv_RS100cm_hits_parameterization

# Reflected (Visible)
sbnd_pdfastsim_par.DoReflectedLight: true
sbnd_pdfastsim_par.VISTiming:           @local::sbnd_vis_timing_parameterization
sbnd_pdfastsim_par.VISHits:             @local::sbnd_vis_RS100cm_hits_parameterization




# Optical-Library mode

# standard configuration
sbnd_pdfastsim_pvs:                     @local::standard_pdfastsim_pvs

sbnd_pdfastsim_pvs.SimulationLabel:     "IonAndScintOUT"
sbnd_pdfastsim_pvs.IncludePropTime: true
sbnd_pdfastsim_pvs.StoreReflected: true



END_PROLOG
```

G4 fhicl file refactored.

- standard_g4_refactored_sbnd.fcl:
- Add: #include “PDFastSim_sbnd.fcl”
  - Services:

```
LArG4Detector :
{
   category      : "world"
   # List of volumes where Ionization/scintillation are simulated (i.e. material = "LAr")
   volumeNames   : ["volTPCActive", "volCryostat", "volTPCPlaneVert","volXArapuca","volPMT","volFieldCage",\
"volPDSstructure"]
   # Corresponding stepLimits in mm for the volumes in the volumeNames list
   stepLimits    : [0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3]
   gdmlFileName_ : "sbnd_v02_00_nowires.gdml"
}
```

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

G4 fhicl file refactored.

- producers:

```
physics:
{
    producers:
    {
        largeant:
        {
            module_type:         "larg4Main"
            enableVisualization:  false
        }
        rns:
        {
            module_type:         "RandomNumberSaver"
        }

        # HYBRID Optical mode:
        # Semi-Analytic model INSIDE the Active Volume
        IonAndScintIN:
        {
            module_type:         "IonAndScint"
            Instances:           "LArG4DetectorServicevolTPCActive" # separated by semicolon
            ISCalcAlg:           "Correlated"
            SavePriorSCE:        true
        }
        PAR:  @local::sbnd_pdfastsim_par

        # Optical-Library model OUTSIDE the Active Volume
        IonAndScintOUT:
        {
            module_type:         "IonAndScint"
            Instances:           "LArG4DetectorServicevolCryostat;LArG4DetectorServicevolTPCPlaneVert;LArG\
4DetectorServicevolXArapuca;LArG4DetectorServicevolPMT;LArG4DetectorServicevolFieldCage;LArG4DetectorServi\
cevolPDSstructure" # separated by semicolon
            ISCalcAlg:           "Correlated"
        }
        LIB:  @local::sbnd_pdfastsim_pvs
    }

    analyzers:
    {
    }

    simulate: [ rns, largeant, IonAndScintIN, PAR, IonAndScintOUT, LIB ]
```