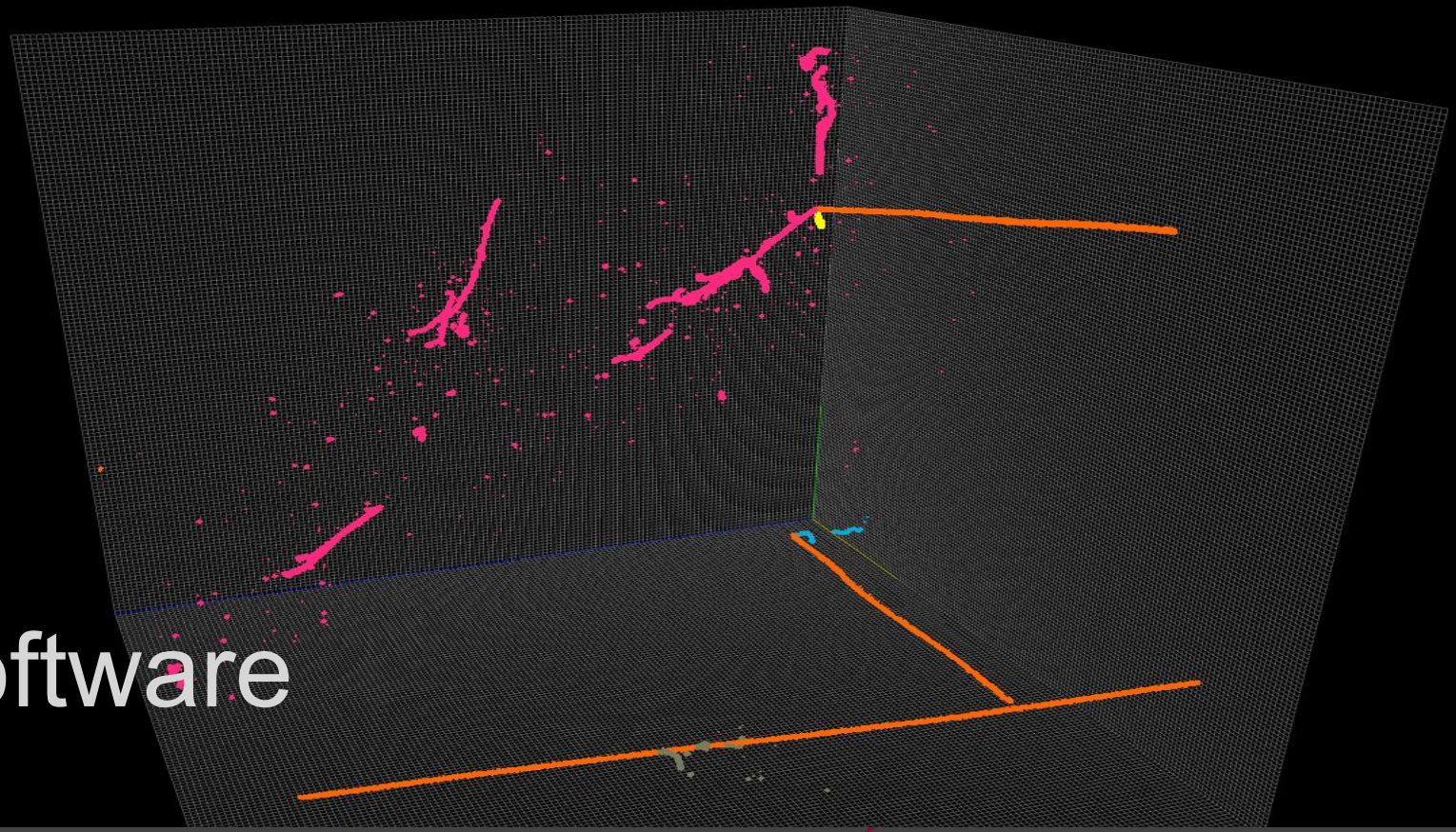


LAr Software

Kazuhiro Terao on behalf of many others
SLAC National Accelerator Lab.
DUNE ND Workshop @ DESY (20-23 October 2019)

DUNE Near Detector
Plan for LArSoft Integration (Simulation)

SLAC

Simulation … what I showed last May …

1. Debug the pixel geometry implementation
   - **Goal**: run largeant for wire & pixel geometry
2. Generate photon library
   - **Goal**: photon library within TPC active volume
3. Implement E-field response into LArSoft
   - **Goal**: run drift simulation for wire & pixel geometry
4. Implement pixel readout response into LArSoft
   - **Goal**: run the whole readout chain for pixel (no wire)
5. Keep working till we are happy

DUNE Near Detector
Plan for LArSoft Integration (Simulation)

SLAC

Simulation … what I showed last May …

1. Debug the pixel geometry implementation
   - **Goal**: run largeant for wire & pixel geometry

2. Generate photon library — **We are still at this :(**
   - **Goal**: photon library within TPC active volume

3. Implement E-field response into LArSoft
   - **Goal**: run drift simulation for wire & pixel geometry

4. Implement pixel readout response into LArSoft
   - **Goal**: run the whole readout chain for pixel (no wire)

5. Keep working till we are happy

DUNE Near Detector
Plan for LArSoft Integration (Simulation)

SLAC

Simulation … what I showed last May …

1. Debug the pixel geometry implementation
   - **Goal**: run largeant for wire & pixel geometry
2. Generate photon library
   - **Goal**: photon library within TPC active volume
3. Implement E-field response into LArSoft
   - **Goal**: run drift simulation for wire & pixel geometry
4. Implement pixel readout response into LArSoft
   - **Goal**: run the whole readout chain for pixel (no wire)
5. Keep working till we are happy

… which is needed for this …

Dan Dwyer and Dan Douglas will contribute

DUNE Near Detector
Plan for LArSoft Integration (Simulation)

SLAC

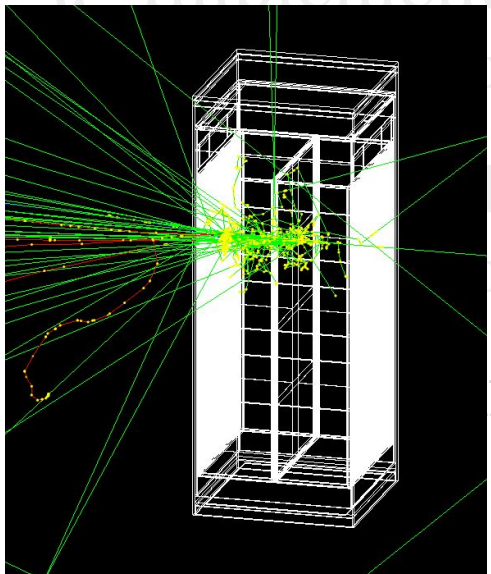
Simulation … what I showed last May …

1. Debug the pixel geometry implementation
   - **Goal**: run largeant for wire & pixel geometry

2. Generate photon library
   - **Goal**: photon library within TPC active volume

3. Implement E-field response into LArSoft
   - drift simulation for wire & pixel geometry
   - pixel signal response into LArSoft
   - the whole readout chain for pixel (no wire)

ing till we are happy



Patrick K. (Bern) started photon sim. for 2x2 outside LArSoft (great!). Natural to continue on 2x2 first…

DUNE Near Detector
Required fundamental changes in LArSoft

SLAC

Geometry in LArSoft: as data representation

- the geometry is a representation of the detector through LArSoft *objects*
- hierarchical: cryostats containing TPC's which contain planes which contain sensitive elements; DUNE near detector:
  - one object per cryostat (... ok, that's 1)
  - one object per TPC (that's 35… nothing new to the far detector)
  - one object per sensitive plane (that's 35 more… are we actually cheaper than wire-based?)
  - one object per sensitive element (about 110,000/m^2… that's 3.9 millions… ouch)
- this approach does not scale well
  - how many square meters in a pixel far detector? we get to CMS levels...
- a different approach is required

Note: here we adopt a novel neutral terminology where a readout plane, based on wires, pixels or mushrooms, is just called a *sensitive plane*, and the wire, pixel or mushroom is a *sensitive element*.

From Gianluca P.

Geometry in LArSoft: as a service

- geometry is a service with an interface and a promise: software calls
  `geom->NearestChannel({ 1.0, 3.0, 2.0 }, { 0, 2, 1 })`
  and expects a channel number in the first plane of second TPC etc.
- examples of questions to answer: *[geo::GeometryCore, geo::PlaneGeo, geo::WireGeo]*
  - which direction the plane is measuring *(note: "direction" is here singular)*
  - which is the channel of a given *sensitive plane* closest to a point in a TPC
  - where do these two *sensitive elements* intersect
- adapting the geometry to host a pixel detector means
  - to answer these predefined calls in the more fitting possible way
  - to produce a sensible behaviour when these calls can't be answered
  - to rethink the new features of pixel readout in view of the concepts already built in
  - to add the concepts still not fitting in a non-intrusive way

From Gianluca P.

DUNE Near Detector
Required fundamental changes in LArSoft

SLAC

How to get there

- the geometry is able to answer most of the questions *in only one way*
  - ➢ **flexibility needs to be added**
  - ➢ that will be a large bite to swallow for LArSoft: big change, many things may go wrong
- the best we can do this, less work will be required downstream
- we are prototyping a solution envisioning:
  - ➢ replacement of wire permanent objects with *temporary* sensitive element ones
  - ➢ a sensitive plane interface with implementations specific to wire and pixel planes
  - ➢ as much as a backward-compatible interface as possible

From Gianluca P.

DUNE Near Detector
Required fundamental changes in LArSoft

SLAC

Status of our prototype implementation

- architectural changes:
  - ✓ introduction of sensitive plane interface
  - ✓ introduction of sensitive element interface
  - ✓ removal of the necessity of permanent representation of sensitive wire/element in memory
- backward compatibility restoration:
  - ✓ implementation of existing wire plane and wire objects under that interface
- pixel detector specific implementation:
  - extension of the interface to accommodate pixel features: *in progress*
  - implementation of pixel plane under that interface: *in progress*
  - implementation of a specific channel mapping (associating a readout channel to each sensitive element): *to do*

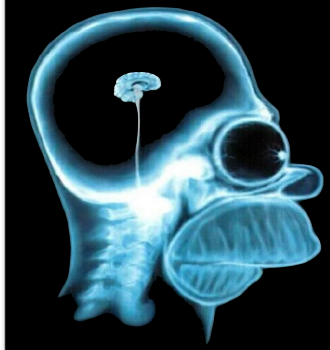
The prototype code is pushed as `feature/gp_wirelessGeometry` in `larcorealg`.
This is the first time this architecture is presented in public. Discussion may follow.

From Gianluca P.

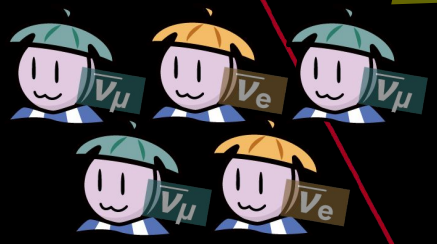
DUNE Near Detector
Moving forward

SLAC

- Big to-do list remains the same
  - **Biggest problem**: my (Kazu's) fault for not chasing and pushing the status/progress constantly. **Intrinsic issue** is increased involvement in ProtoDUNE and ICARUS for both Kazu and Gianluca
  - We try to get this done for the 1st priority for 2x2 now, and more help can be used once the fundamental overhaul in the technical (LArSoft Geometry) is over…
- Expected contributions (human power)
  - Dan Douglas on E-field response, fraction of myself, Gianluca, and a postdoc (unspecified) @ SLAC, anyone else willing to contribute…



Reconstruction For DUNE ND



U.S. DEPARTMENT OF ENERGY
Office of Science

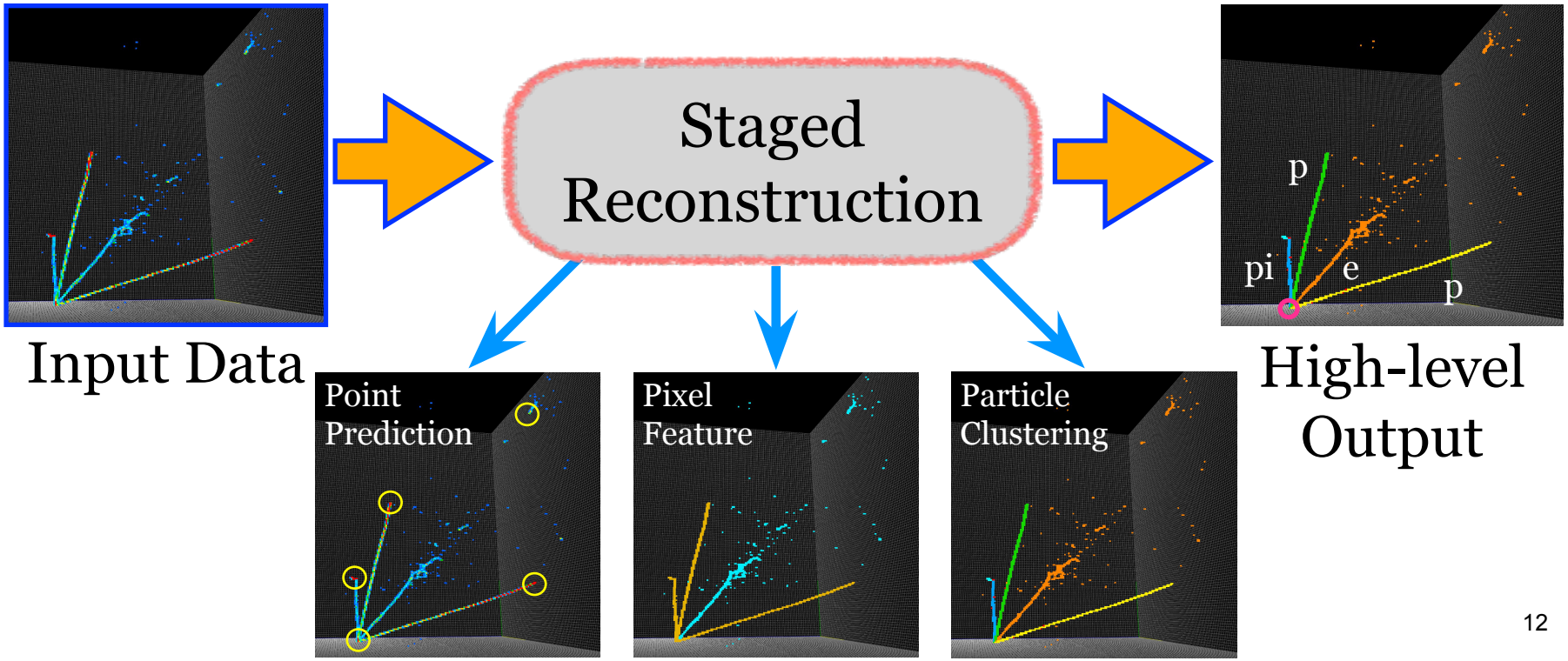
SLAC NATIONAL ACCELERATOR LABORATORY

DUNE Near Detector
(ML-based) Data Reconstruction R&D

SLAC

3D-Native Data Reconstruction

- Some scattered development of algorithms (e.g. Cluster3D)
- ML-based data reconstruction chain (SLAC focus)
  - "Staged" reconstruction approach, which allows us to combine with non-ML techniques where appropriate



DUNE Near Detector
ML-based Data Reconstruction Development

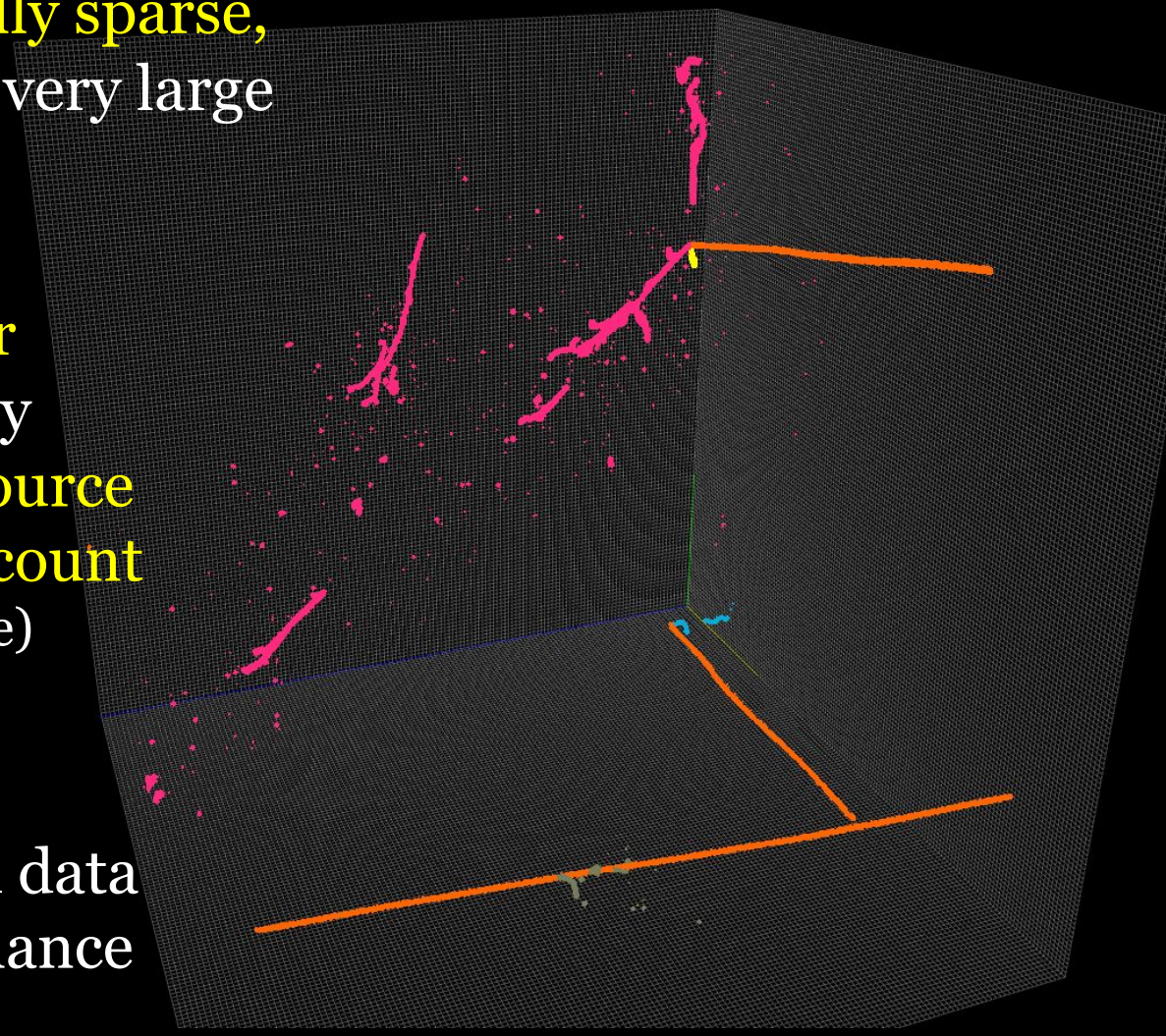
SLAC

Data feature: generally sparse, locally dense image, and very large volume (1 E10-15 pixels)

Solution: sparse linear algebra that operates only on non-zero pixels = resource scales by non-zero pixel count (i.e. DUNE-FD is piece of cake)

Fitness to LArTPC

Perfect: densely sampled data improve physics performance with no down-sampling

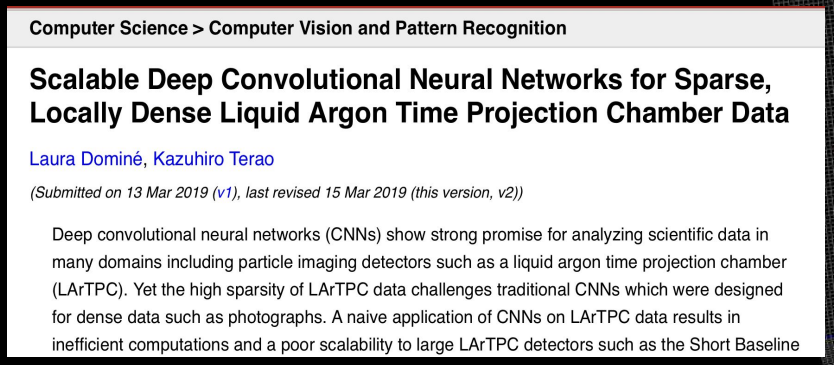


DUNE Near Detector
ML-based Data Reconstruction Development

SLAC

Five-Type Segmentation Accuracy

Type	HIP	MIP	Shower	Delta	Michel
Acc.	0.99	0.98	0.99	0.97	0.96



Computer Science > Computer Vision and Pattern Recognition

Scalable Deep Convolutional Neural Networks for Sparse, Locally Dense Liquid Argon Time Projection Chamber Data

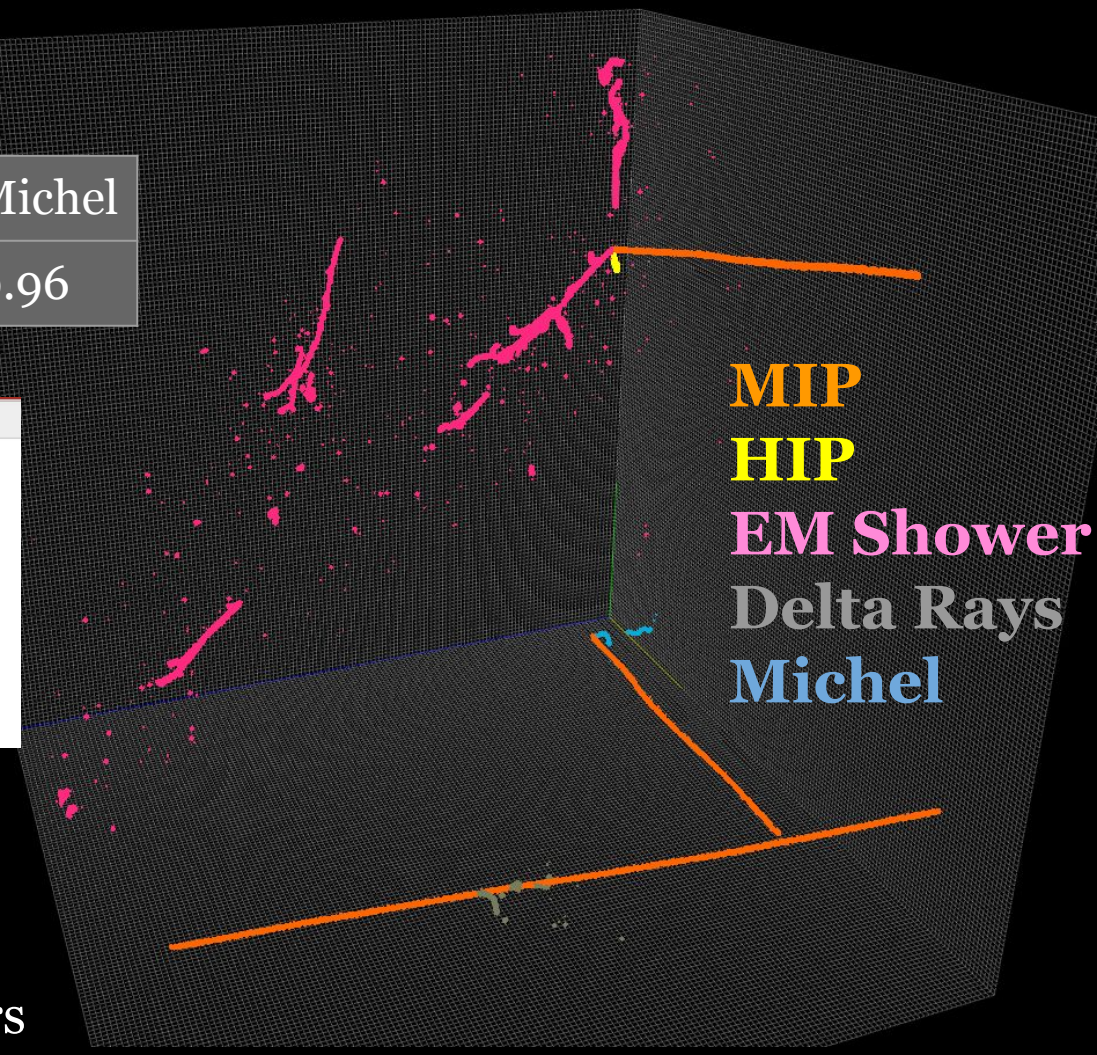
Laura Dominé, Kazuhiro Terao

(Submitted on 13 Mar 2019 (v1), last revised 15 Mar 2019 (this version, v2))

Deep convolutional neural networks (CNNs) show strong promise for analyzing scientific data in many domains including particle imaging detectors such as a liquid argon time projection chamber (LArTPC). Yet the high sparsity of LArTPC data challenges traditional CNNs which were designed for dense data such as photographs. A naive application of CNNs on LArTPC data results in inefficient computations and a poor scalability to large LArTPC detectors such as the Short Baseline

arXiv:1903.05663 presented @ ACAT 2019

- Memory reduction ~ 1/360
- Compute time ~ 1/30
- Viable to process large detectors (ICARUS, DUNE-FD, etc.)



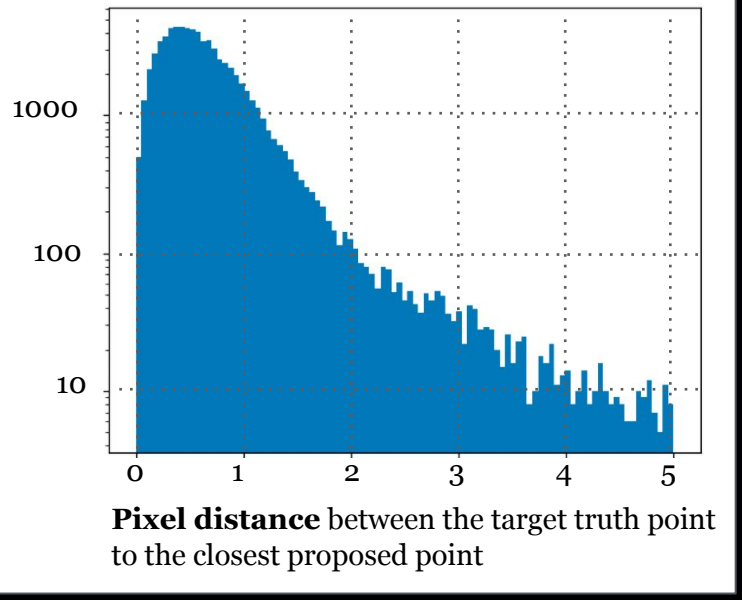
DUNE Near Detector
ML-based Data Reconstruction Development

SLAC

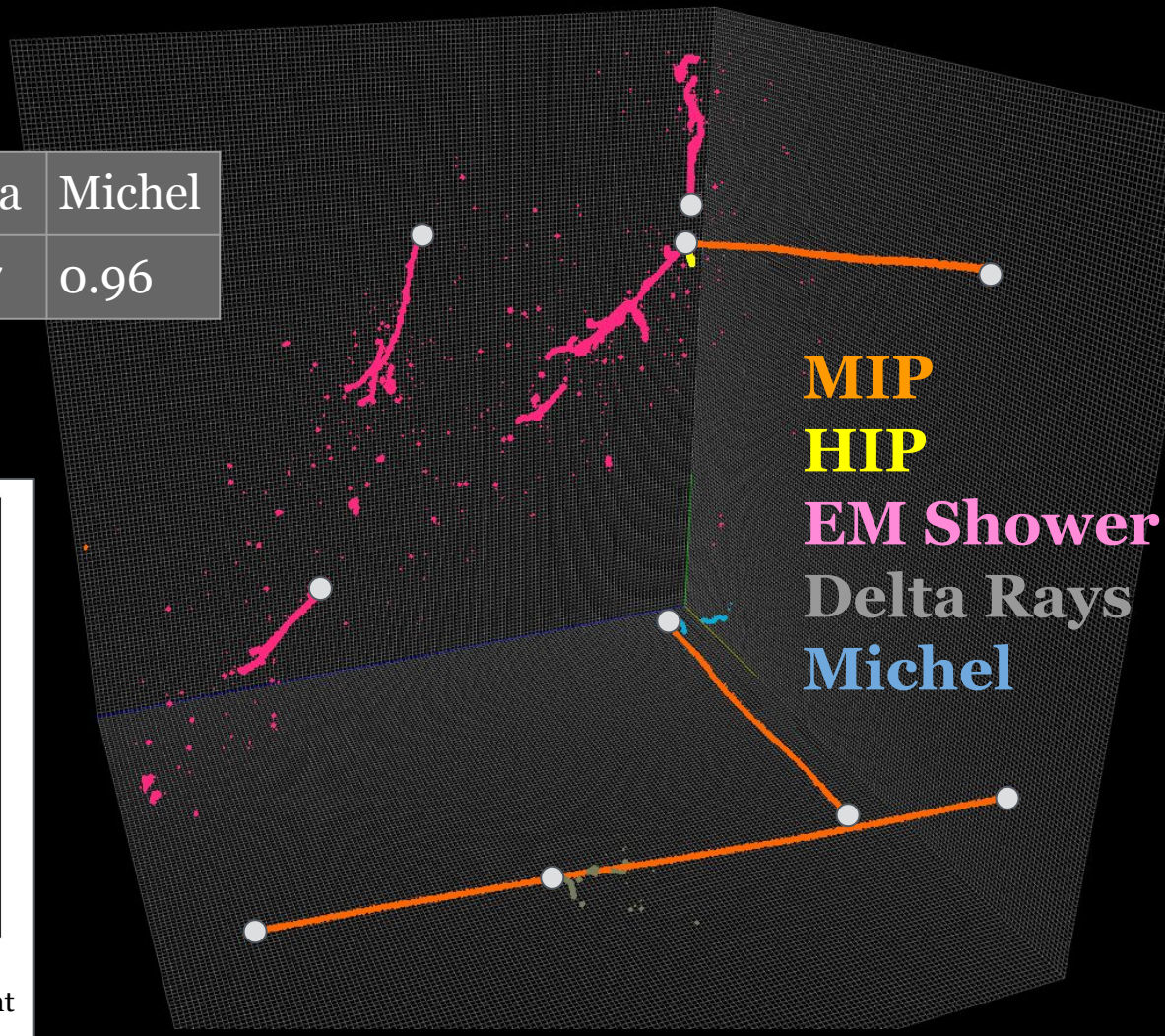
Five-Type Segmentation Accuracy

Type	HIP	MIP	Shower	Delta	Michel
Acc.	0.99	0.98	0.99	0.97	0.96

Point Prediction



Pixel distance between the target truth point to the closest proposed point



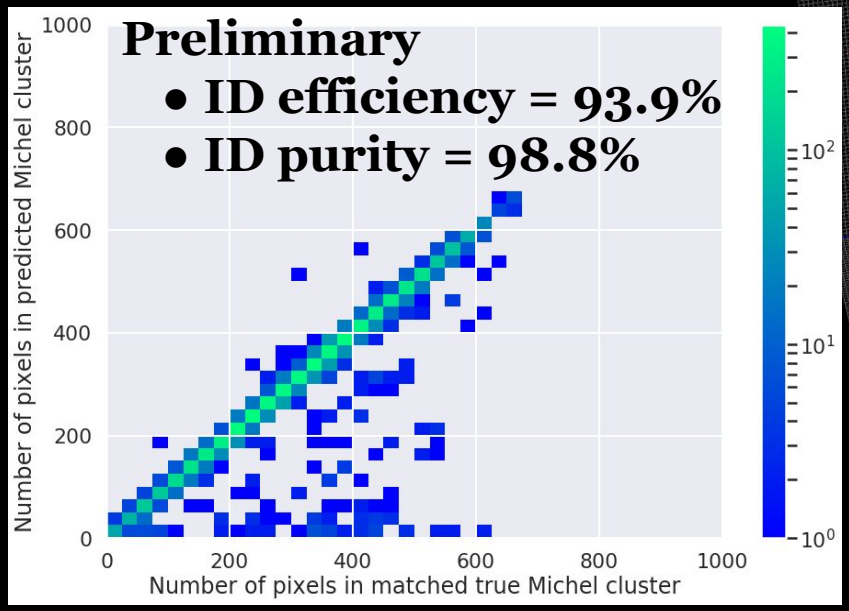
DUNE Near Detector
ML-based Data Reconstruction Development

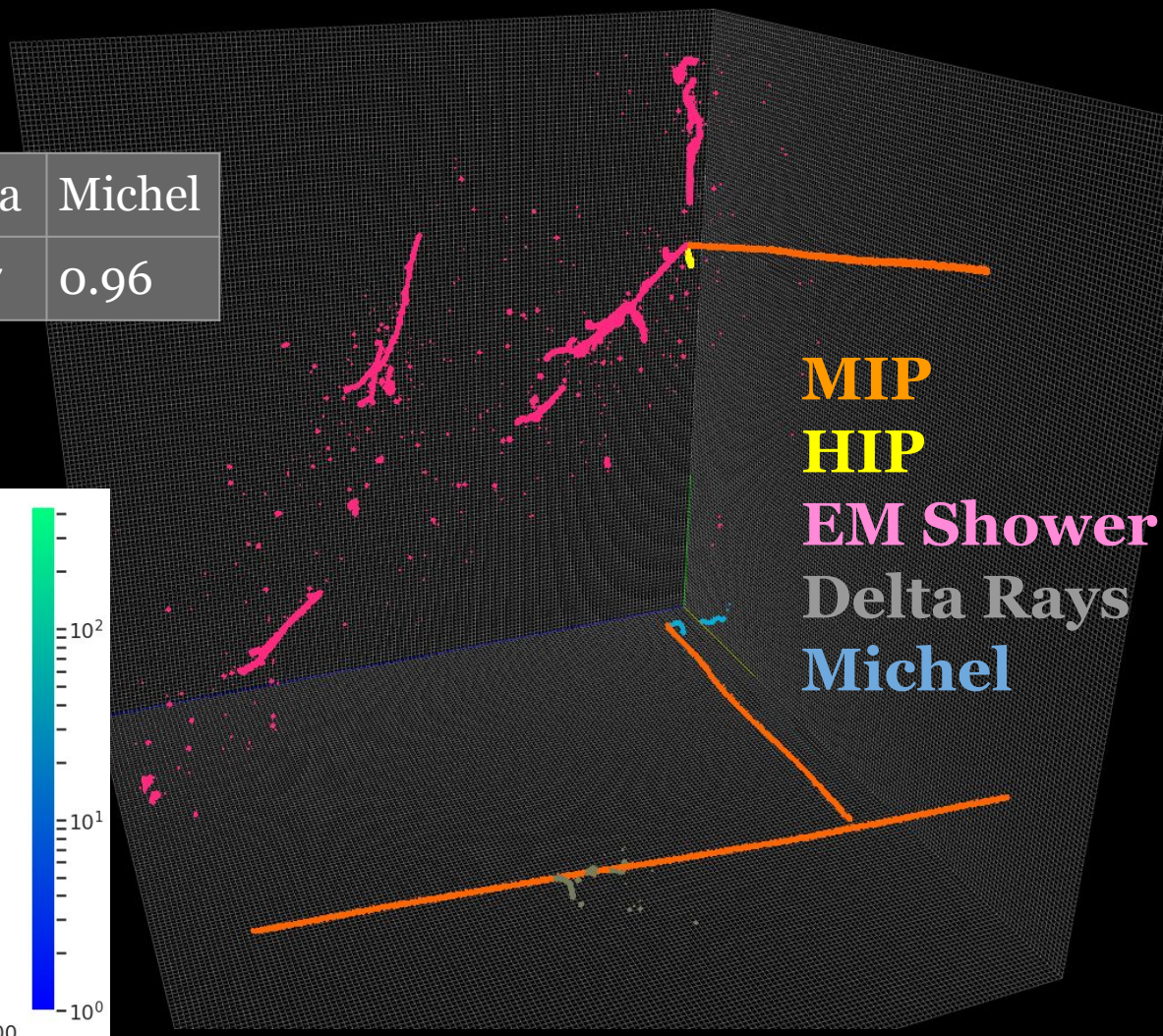
SLAC

Five-Type Segmentation Accuracy

Type	HIP	MIP	Shower	Delta	Michel
Acc.	0.99	0.98	0.99	0.97	0.96

Michel Electron Demo



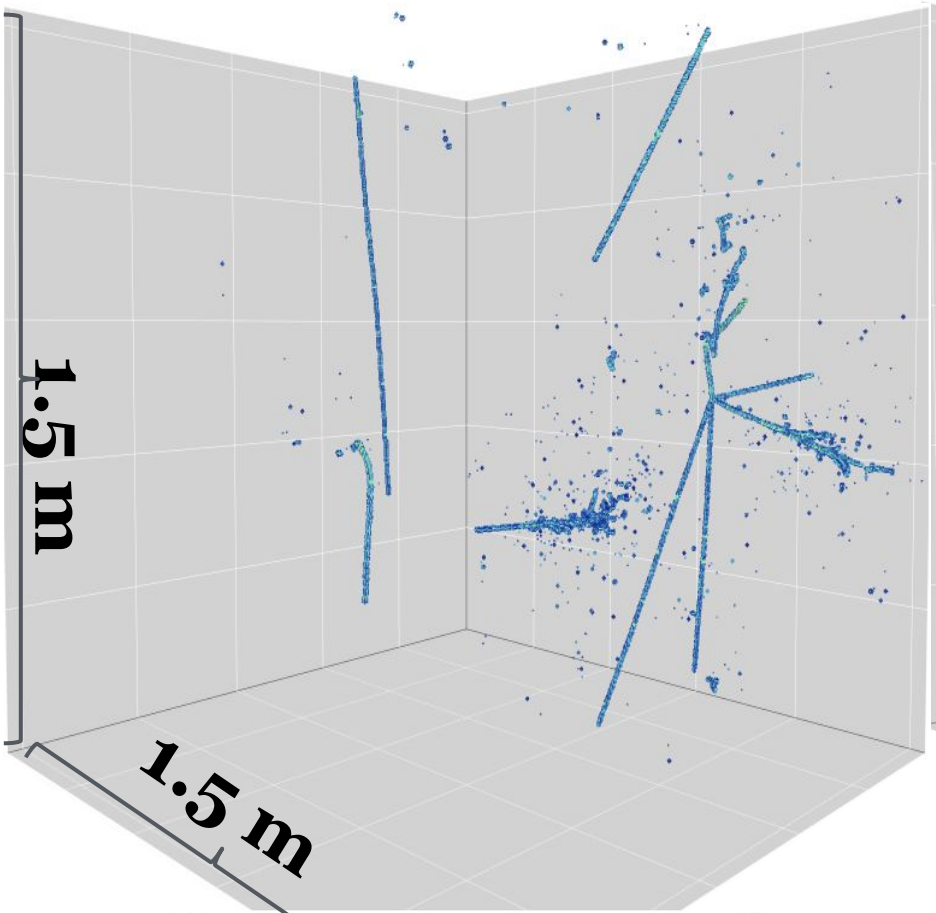


DUNE Near Detector
ML-based Data Reconstruction Development

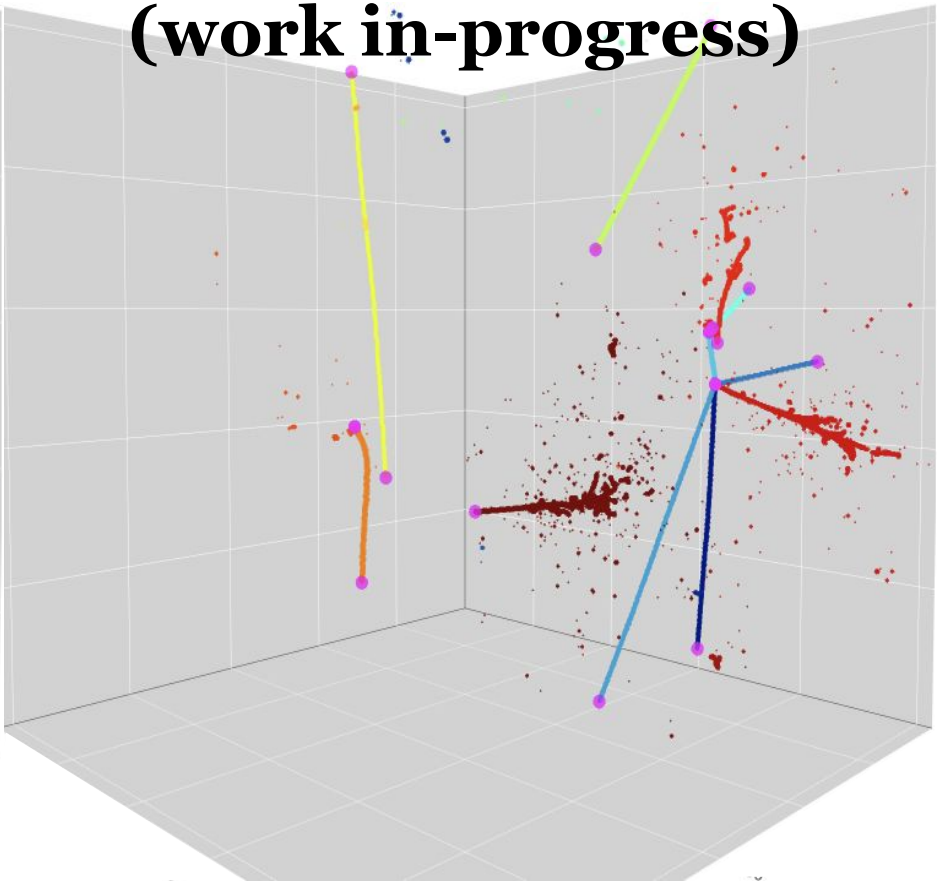
SLAC

Current focus: particle-instance clustering (CNN/GNN)

Input



Output
(work in-progress)

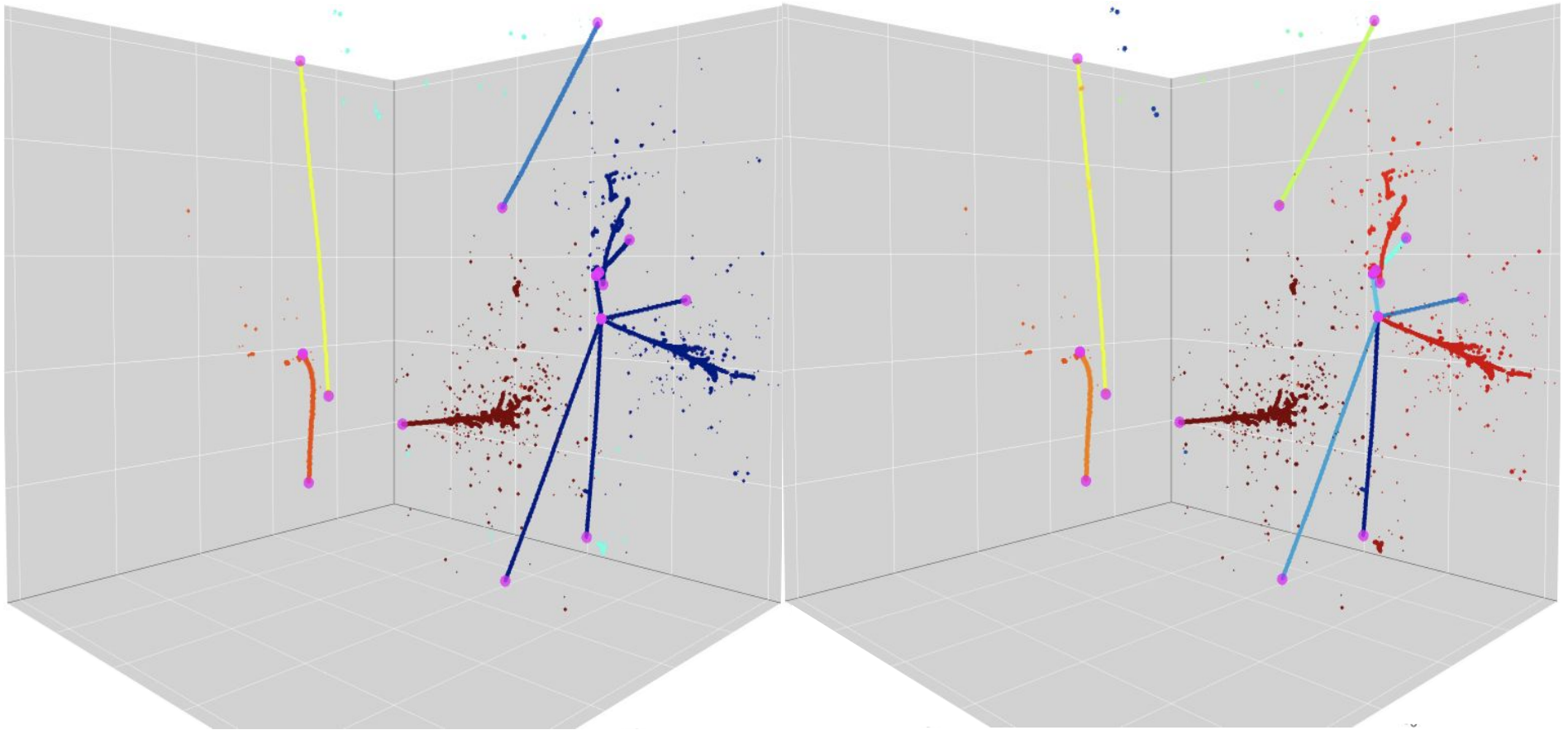


DUNE Near Detector
ML-based Data Reconstruction Development

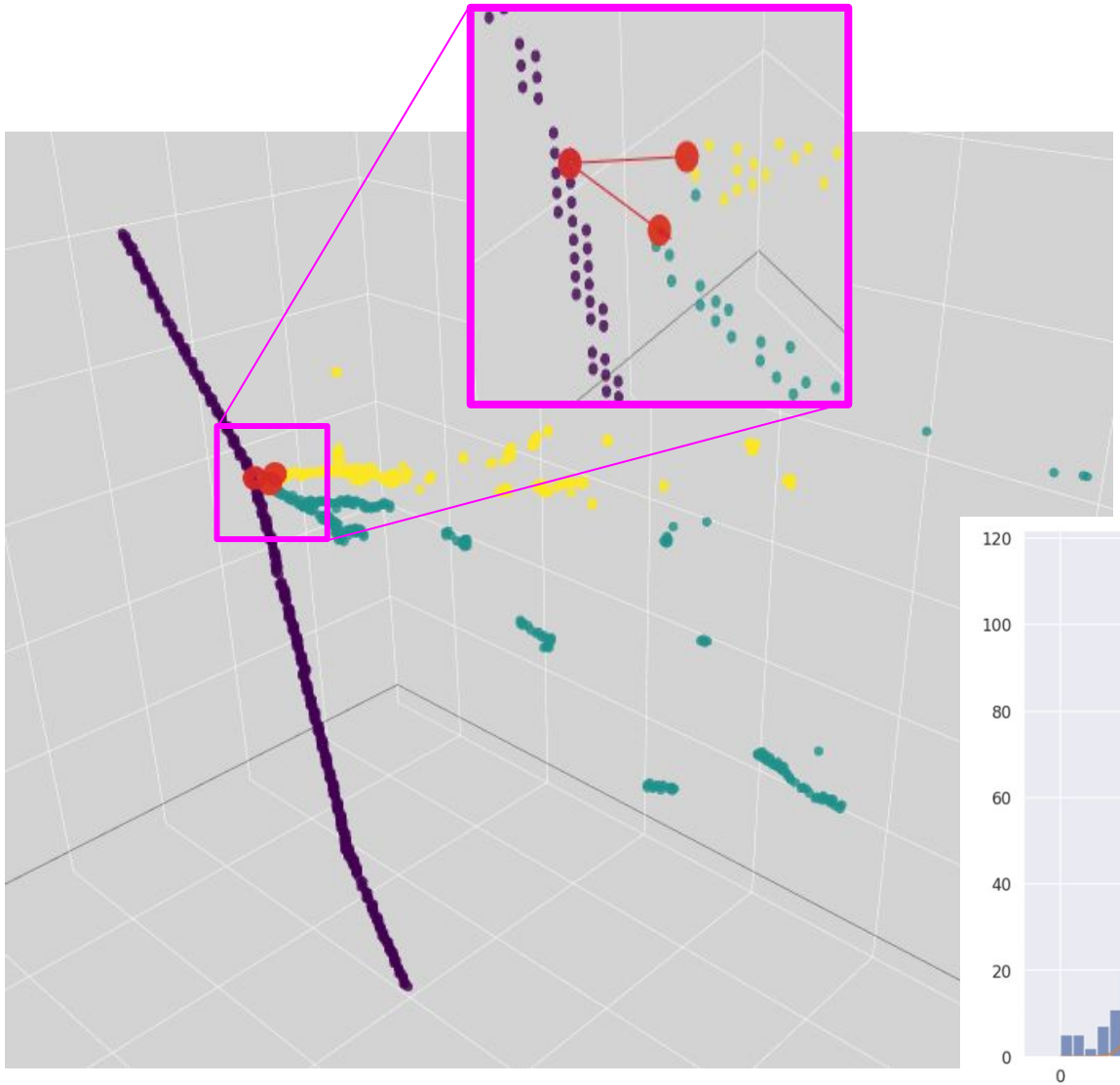
SLAC

Next focus: interaction clustering (GNN)

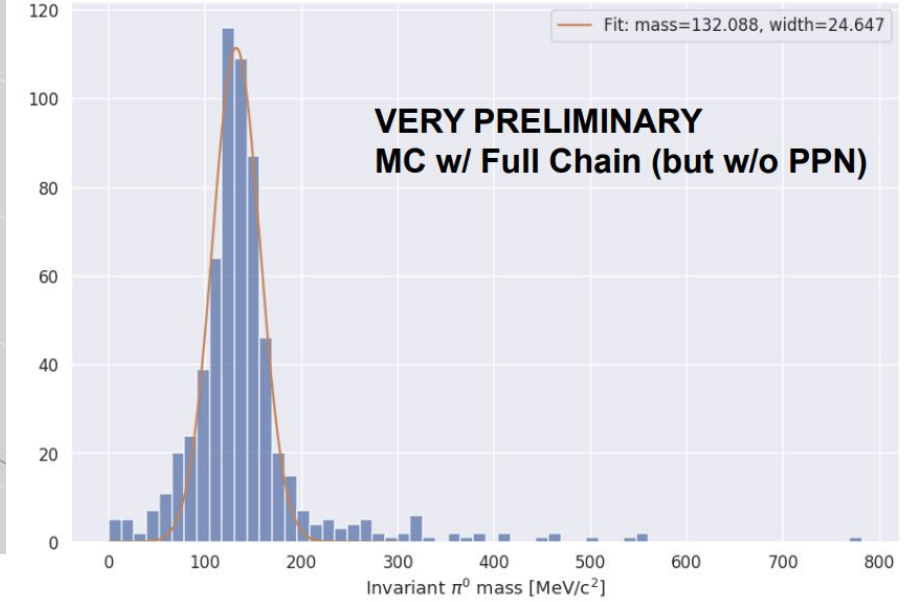
Interaction | **Particle**



DUNE Near Detector
Cross-experimental Development



- **ProtoDUNE-FD π^0** reco (LBNL/CSU/SLAC), interest in joining also 2x2 with Roman @ Bern
- **SBN** in multiple fronts ($\nu\mu$, νe, michel, π^0, etc.)



Software Infrastructure

- Base: pytorch/scipy + custom CUDA/C++ kernels
  - Container-based solution established in UBoone, running on FermiGrid (sparse-CNN allows on-CPU running)
  - HPC-compatible software for massively-parallel data distribution (6TB/s dense-equivalent thru-put, 8000 ranks using MPI w/ infiniband) developed in a small collaboration (SLAC/ATLAS/NEXT), applying for HPC hours next year
- Need to start thinking how to incorporate data from other detectors…

DUNE Near Detector
LAr Software Summary

SLAC

- **Simulation software** development has been on a halt, really need get done…
  - New lead recommendation welcome if any… Regardless we will contribute/finish LAr chain!
  - See slide 2 on the big list of projects

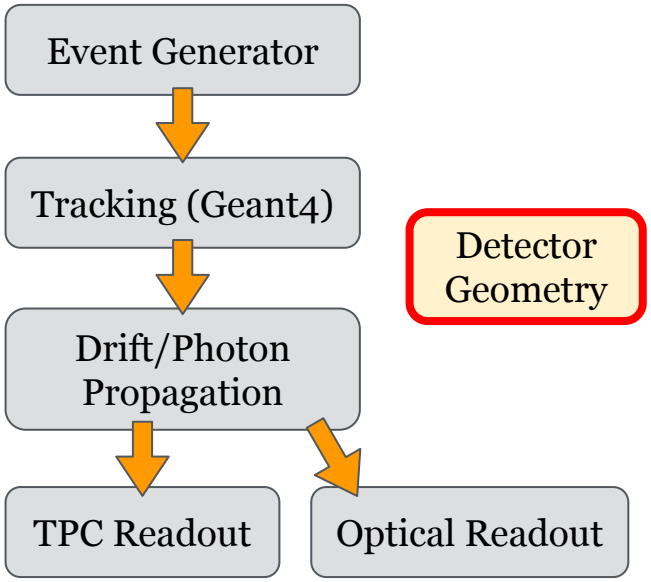
- Reconstruction software development is “active”
  - … but not using a “proper” pixel detector sim.!
  - An inter-experimental effort, lots of interest in applying for 2x2 data.

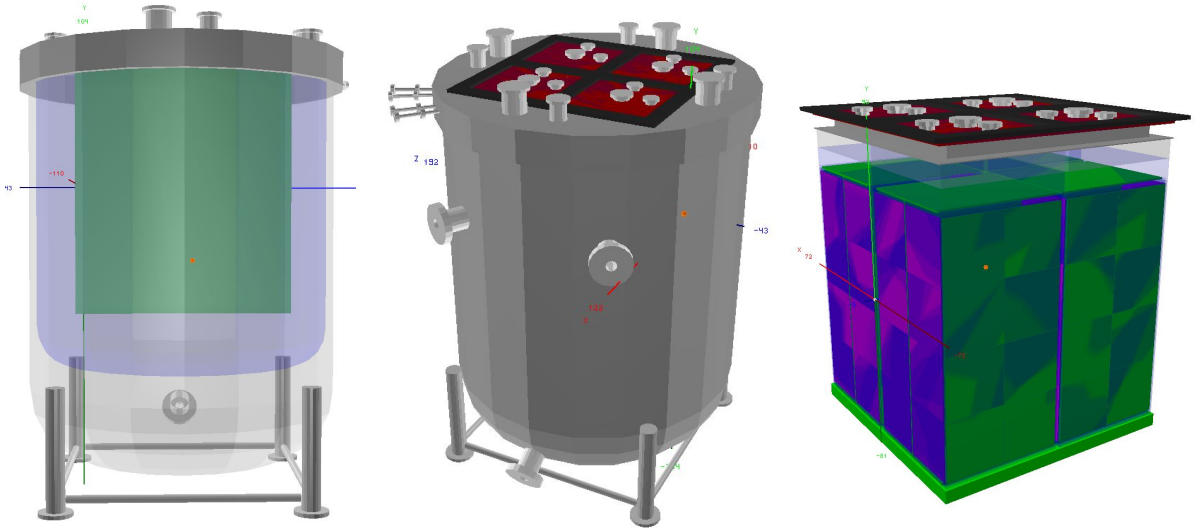
Back ups

DUNE Near Detector
Simulation Software Development

SLAC

Simulation Chain





Images courtesy of Hunter Sullivan @ UTA

Geometry (GDML)

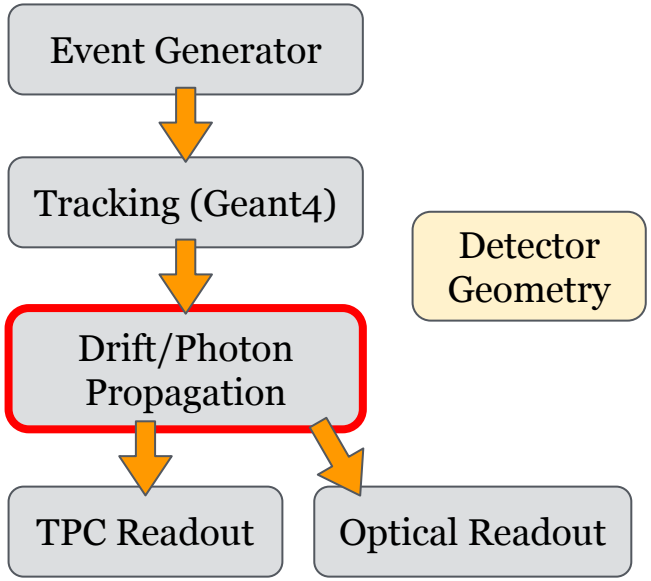
Use GeGeDe-ND toolkit for develop & maintenance of GDML files

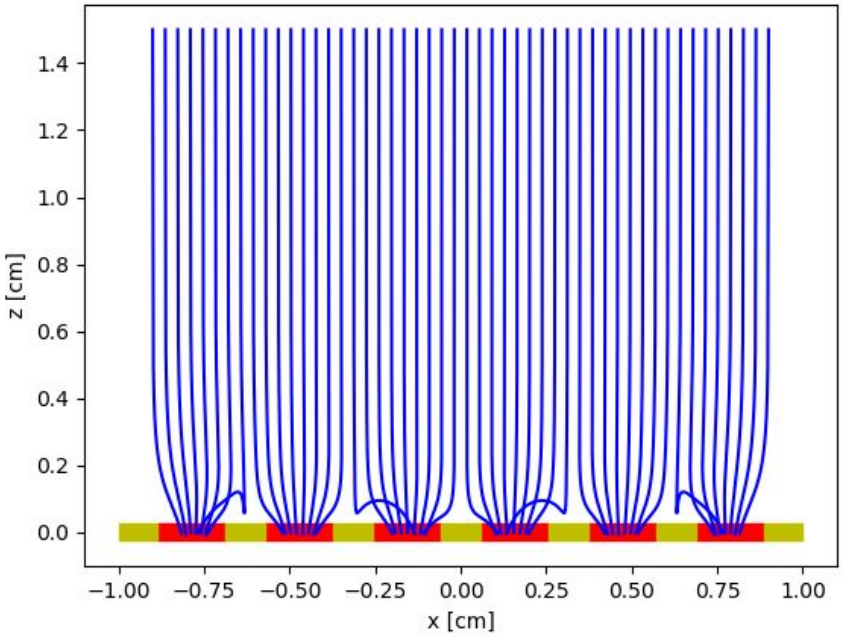
- DUNE-ND preliminary GDML file to be made work in LArSoft
- New GDML file for ArgonCUBE 2x2 module for Fermilab run
  - Implementation by **Patrick** (Bern) & **Hunter** (UTA) including the cryostat, TPC module, and optical modules.

DUNE Near Detector
Simulation Software Development

SLAC

Simulation Chain





Electric field response

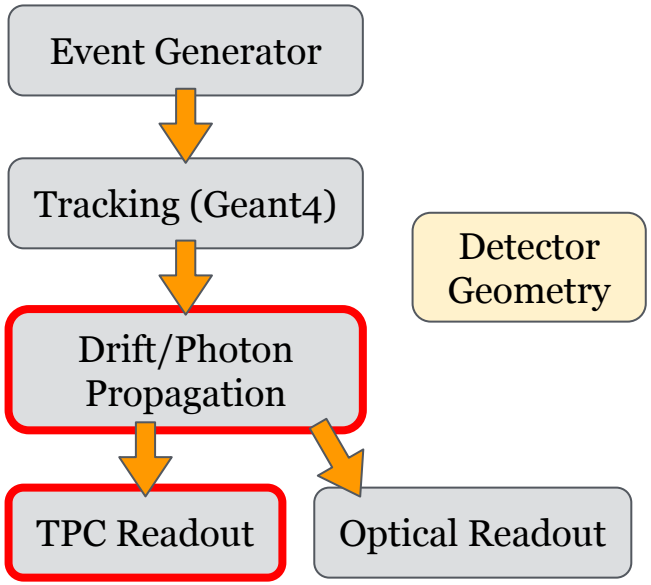
Evolving closely with the detector design updates (kapton wall, pixel geometry, etc.), lead by **Dan Douglas** (MSU)

- Using now 2.5 x 2.5mm pixel + 3.88 mm pitch (from **Knut** @ SLAC)
- Non-uniformity of Kapton surface resistivity
- Dielectric materials

DUNE Near Detector
Simulation Software Development

SLAC

Simulation Chain



TPC Readout

Digitized waveform for electronics response

- "Vectorized code" written by **Dan Dwyer** (LBNL), ready(?) to be integrated with drift/E-field simulation.
- Seek for ways to integrate vectorization scheme made available(?) in LArSoft

Photon Propagation

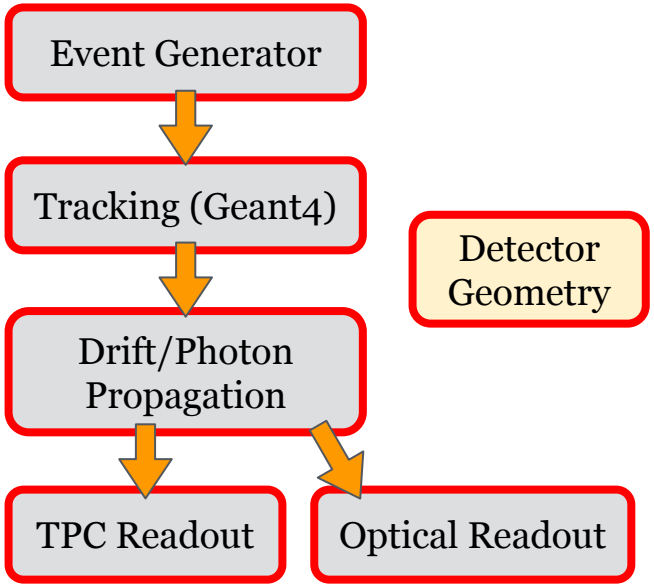
Individual photon ray tracing is time consuming. Study a solution using a photon library for ArgonCUBE 2x2 prototype.

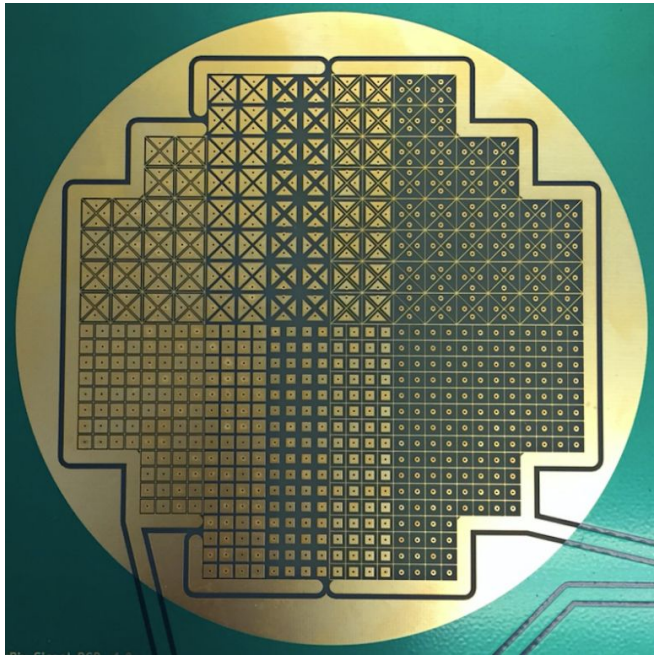
- LArSoft: a "photon library" (look-up table) for photon collection efficiency and timing estimates at different detector locations.
- Need to run a full simulation to build the library

DUNE Near Detector
Simulation Software Development

SLAC

Simulation Chain





Pixel Geometry in LArSoft

LArSoft assumes "wire" in the core of Geometry design & APIs to query geometry information (PlaneGeo/WireGeo). Needs to be changed.

- Designed a generic "charge-sensitive element" to replace the current implementation in non-distruptive manner (by **Gianluca** @ SLAC)
- Now in testing stage: goal is to run largeant for wire & pixel geometry