Differentiable Surrogate for Photon Propagation

Application to Module-0 Data

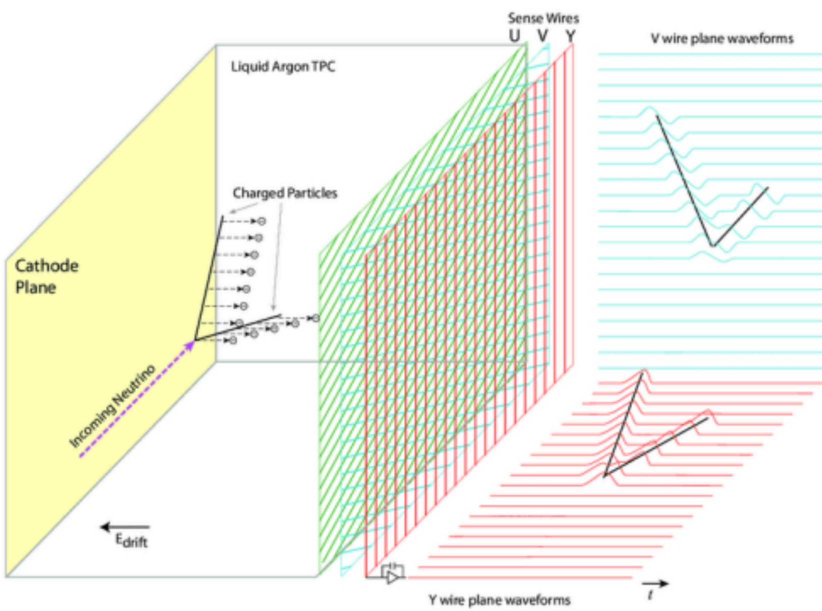
Patrick TSANG (SLAC)
Nov 2, 2022

DUNE-ND ML Reco. Meeting

Liquid Argon Time Projection Chamber (LArTPC)

SLAC

LArTPC w/ wire-based readout



Something happens (e.g. neutrino interaction, cosmic rays) => *charged* particles

=> ionization of Ar

=> ionized e- drifts toward anode

Charge readout options

3 sets of parallel <u>wires</u>

=> 3 projections of 1D+time

—- *OR* —

an array of <u>pixels</u>

=> one 2D+time view

3D Image

Pro: high resolution (~mm) 3D imaging device for charged particles

Challenge: intuitively $x_{drift} = v_{drift} \Delta t$, but what is the reference time (t_0) for Δt ?

LArTPC: Scintillation Light

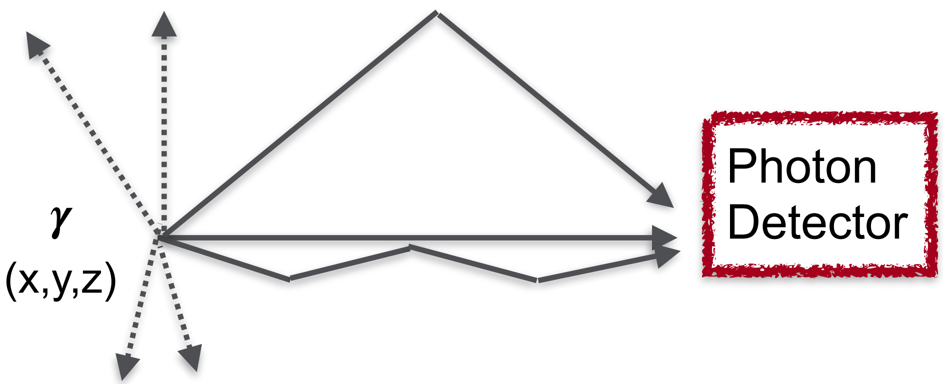
Recombination

$$e^- + {}^{40}\mathrm{Ar}^+ + {}^{40}\mathrm{Ar} \quad \rightarrow \quad {}^{40}\mathrm{Ar}_2^* \quad \rightarrow \quad 2\,{}^{40}\mathrm{Ar} + \gamma.$$

The prompt light signal at O(ns) provides a reference time for Δt

Visibility: Modeling of Photon Propagation

What is the probability to detect those photons produced at a given position (x,y,z)?



- possible optical paths (direct, scattered, reflected) for photon at (x,y,z) to be detected.
- some photons (dotted paths) are not detected.

Visibility: Lookup Table Approach

Traditional Method (Photon Library)

1. divide the detector into O(cm) girds
2. for each grid point (aka voxel), generate O(1M) photons isotropically
3. simulate the photon propagation paths
4. count how many photons are detected
5. estimate the visibility as $N_{detected} / N_{generated}$
6. build a lookup take (LUT)

Cons

- long construction time, O(1) week
- poor scalability
    - ICARUS (the largest LArTPC in operation):
        - ~404M parameters in resolution of 5 cm
    - DUNE-FD (the next generation in construction): ~100x of ICARUS
        - concerns in computation and memory usage
- no well-developed calibration method due to computational complexity

Implicit Neural Representation

Can we parameterize the photon lib. for better computation and memory usage?

No (in the past). The underlying function is usually not analytically traceable. The hand-written formula does not have enough accuracy.

Yes (now). With recent advancement in the *implicit neural representation*.

Implicit Neural Representation

Parameterize signals as *continuous* functions via *neural networks*, which are trained to map the domain the signal (e.g. spatial coordinates) to the target outputs (e.g. signal at those coordinates).

$$f: R^M \rightarrow R^N$$

Example: Image Reconstruction

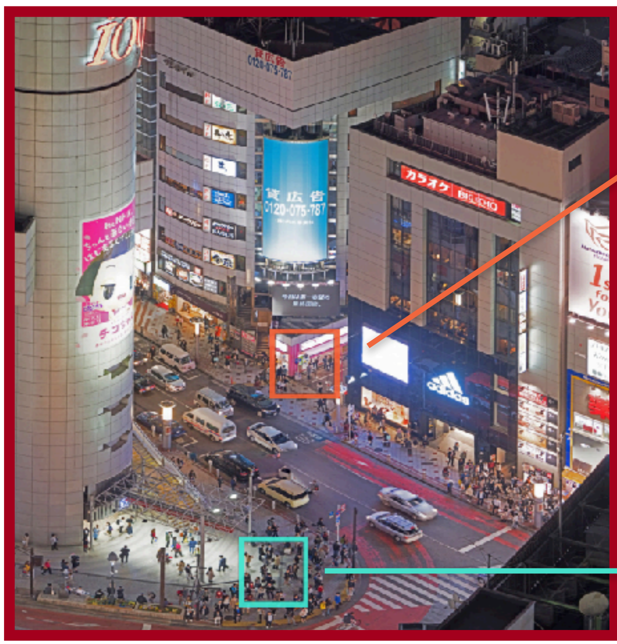
Map pixel coordinates to color pixel: f(x,y) → (R,G,B)

A gigapixel (19,456 × 51,200) panorama of Tokyo is represented by a function of 168M parameters using *ACORN* model

ACORN (arXiv:2105.02788)

Image Reconstruction w/ ACORN

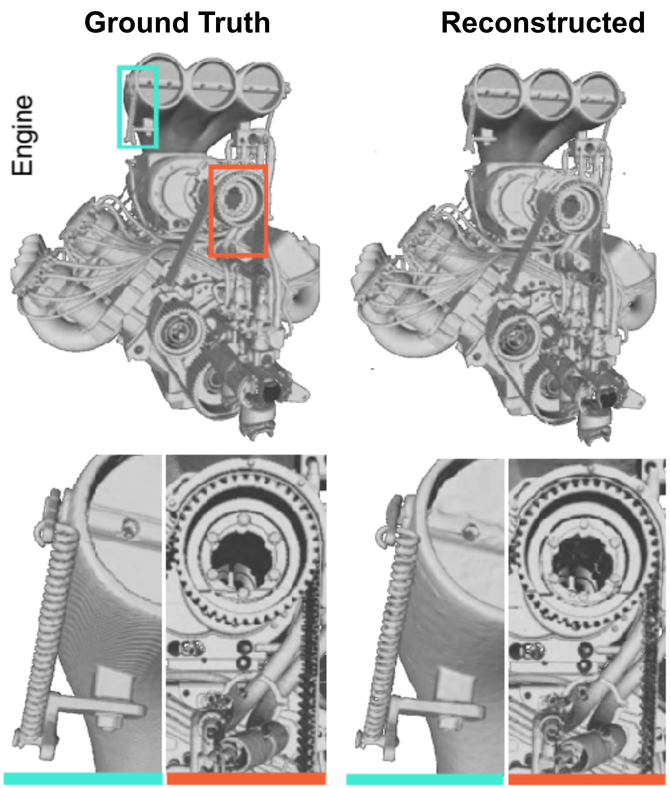
Zoom in



Ground Truth | Reconstructed

High fidelity representation using much less parameters.

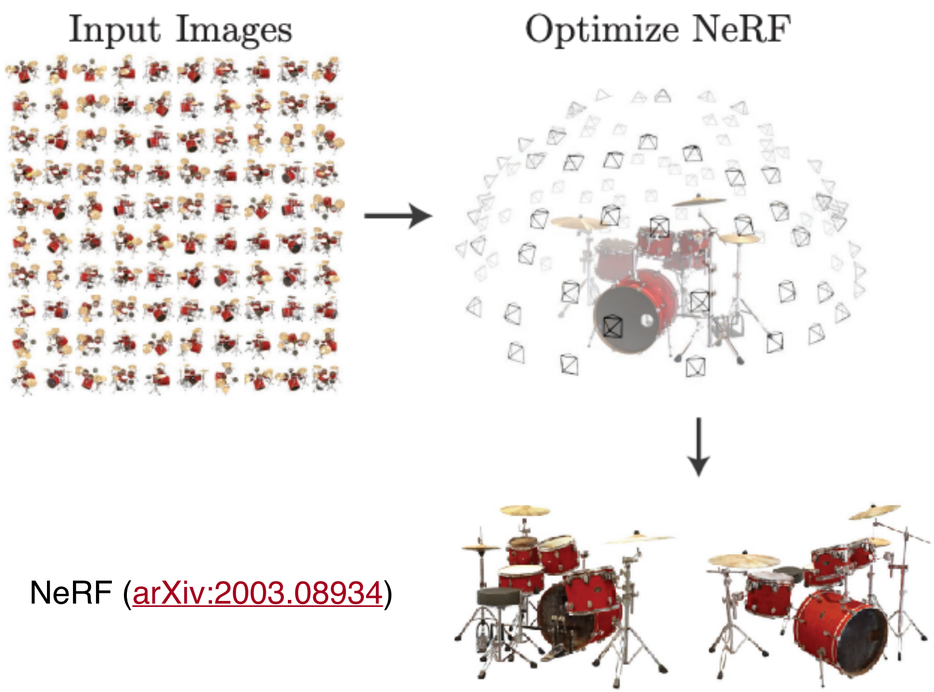
Example: 3D Shape / Scene Rendering



Map 3D coordinates to occupancy

f(x,y,z) → 0 or 1

ACORN (arXiv:2105.02788)



NeRF (arXiv:2003.08934)

Map 3D coord. + camera angles to RGB + volume density

$f(x,y,z,\theta,\varphi) \rightarrow (R,G,B,\sigma)$

Learn the function from a set of 2D images at different views and synthesize a 3D scene at arbitrary angels.

Sinusoidal Representation Network (SIREN)

SLAC

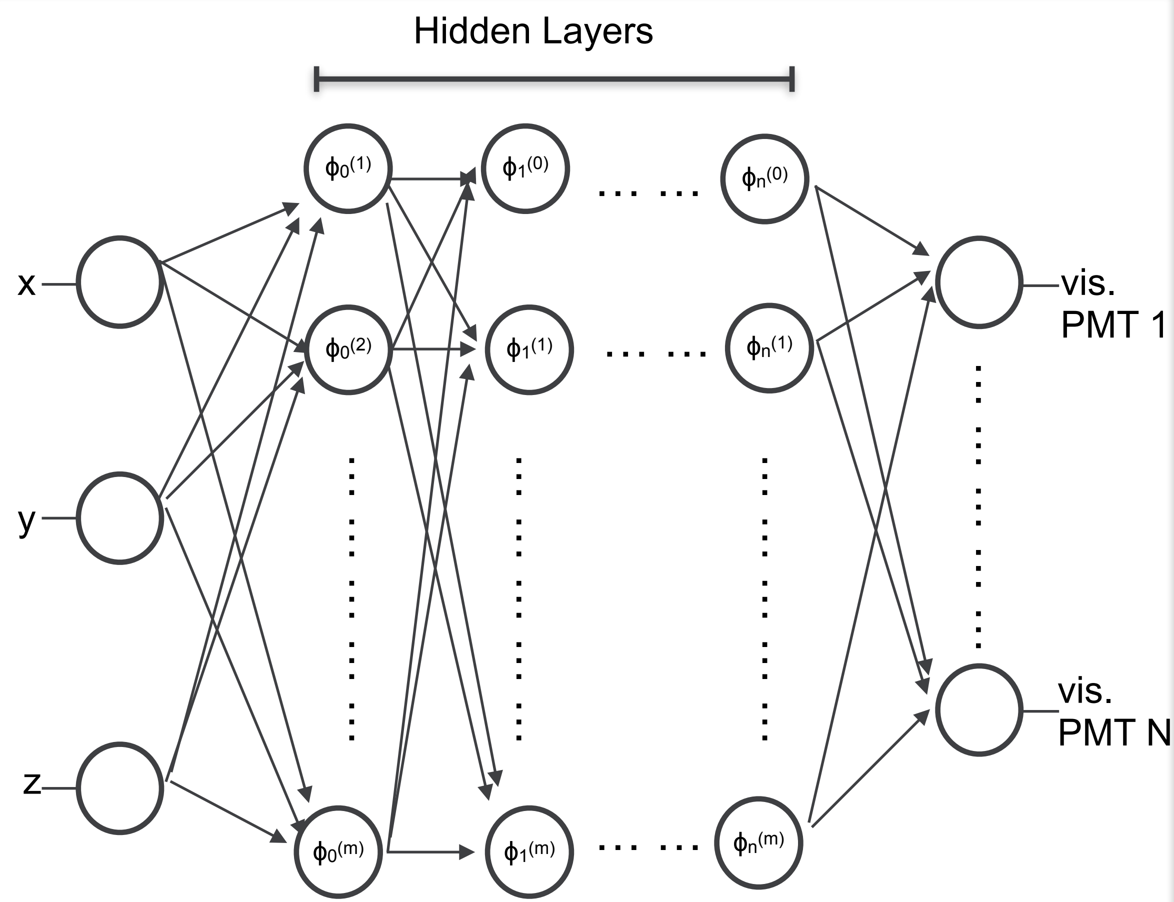
Application to Photon Propagation

map 3D coordinates of the light source to the visibilities of *N* photon detectors (e.g. PMTs)

$\phi(x,y,z) \rightarrow$ visibility x N

SIREN

a simple multilayer perceptron (MLP) network architecture along with periodic *sine* function activations (arXiv:2006.09661)



$$\mathbf{\Phi}(\mathbf{x}) = \mathbf{W}_n(\phi_{n-1} \circ \phi_{n-2} \circ \cdots \circ \phi_0)(\mathbf{x}) + \mathbf{b}_n,$$
$$\phi_i(\mathbf{x}_i) = \sin(\mathbf{W}_i\mathbf{x}_i + \mathbf{b}_i),$$

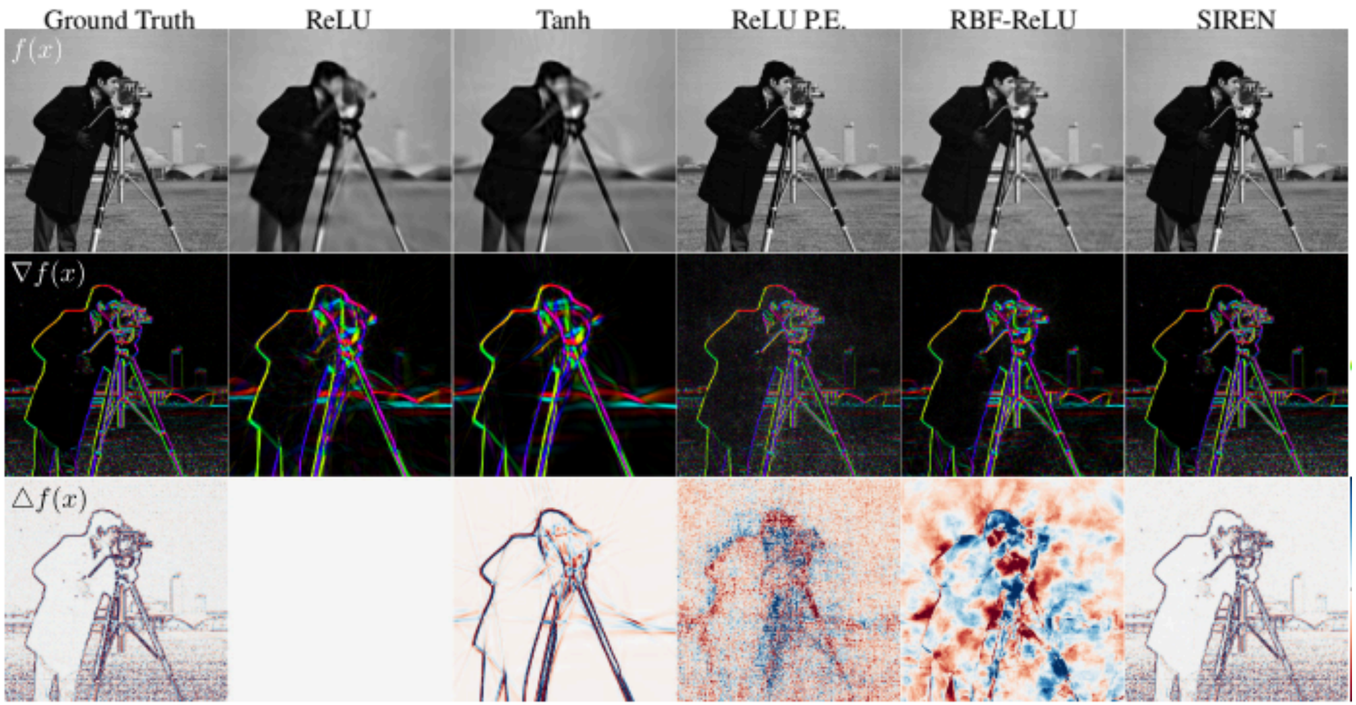
Why SIREN?

By construction, SRIEN is a continuous, differentiable signal representations
=> modeling signals with fine detail, AND
=> representing smooth gradient surface (and higher order of derivatives)

f(x)

1st derivative

2nd derivative



SIREN (arXiv:2006.09661)

Allows wide range of applications from gradient-based algorithms, solving differential equation, optimizing on the derivative … etc

NDLAr Prototype: Module-0

Module-0 Detector

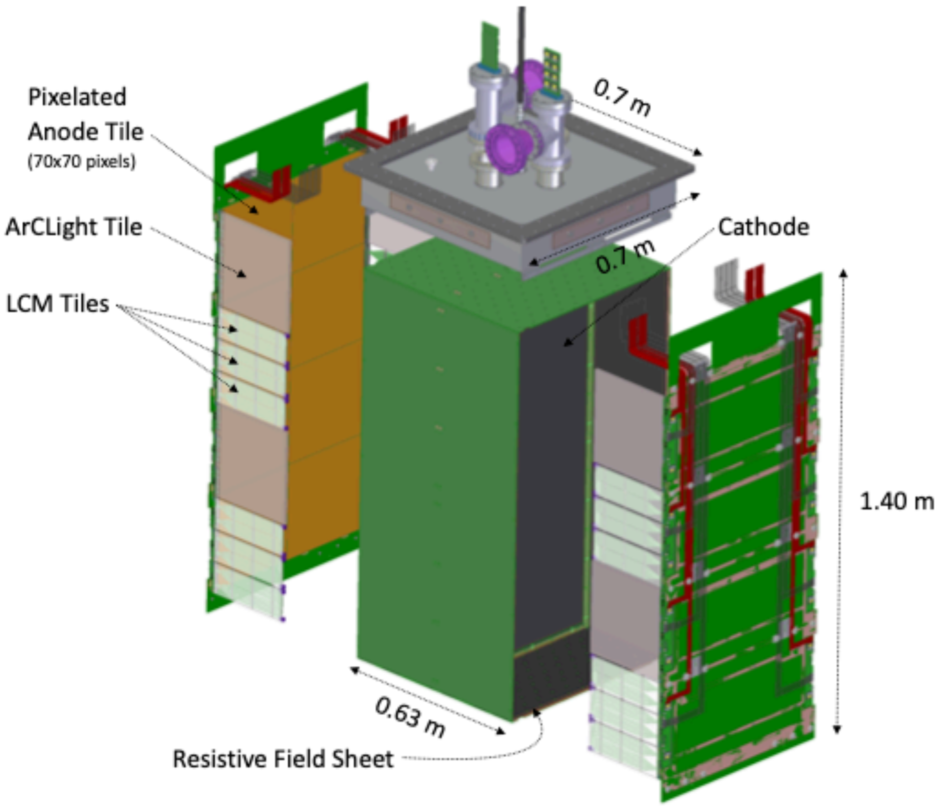


Figure 1. Schematic of the 0.7 m × 0.7 m × 1.4 m Module-0 detector with annotations of the key components.

Short term goal

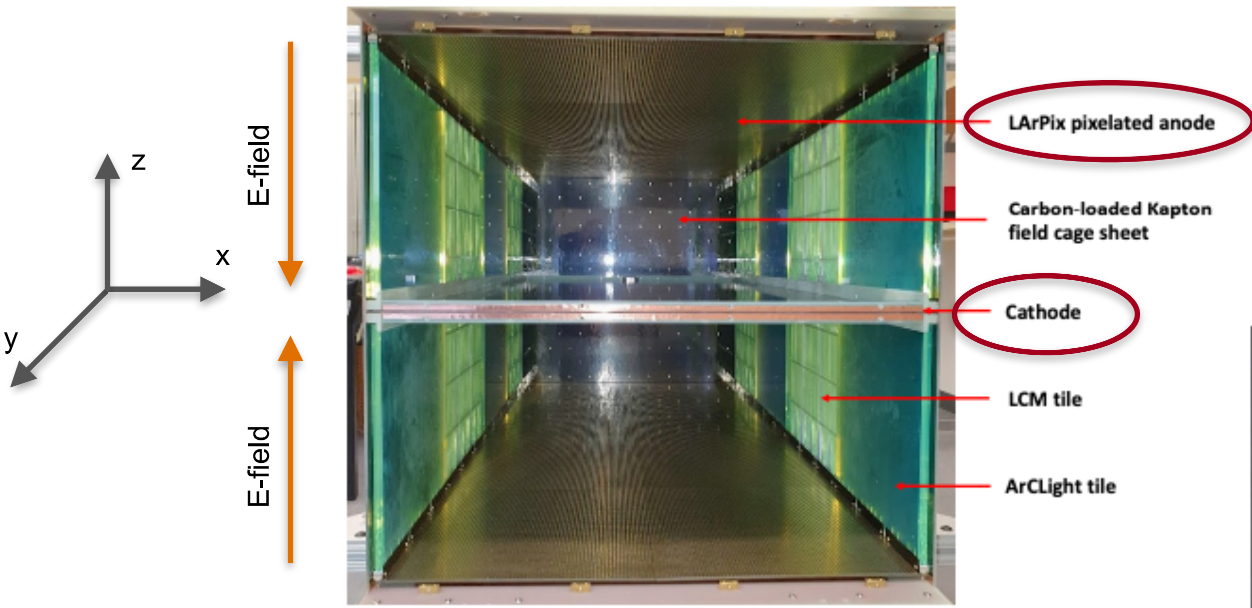
- build a prototype of 2x2 array of detector modules
- test w/ NuMI neutrino beam at Fermilab

Long term goal

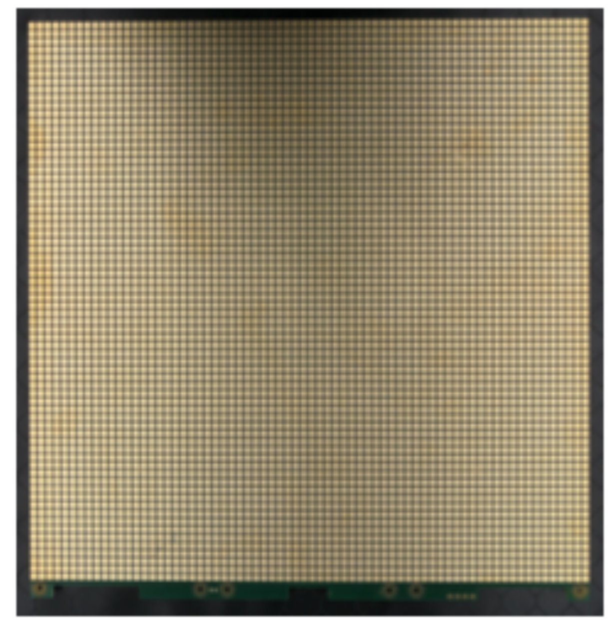
- build a 7x5 array (TBC) for the DUNE Liquid Argon Near Detector

Module-0 Charge Readout System

View from the top of Module-0

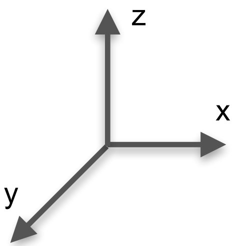


LArPix Tile

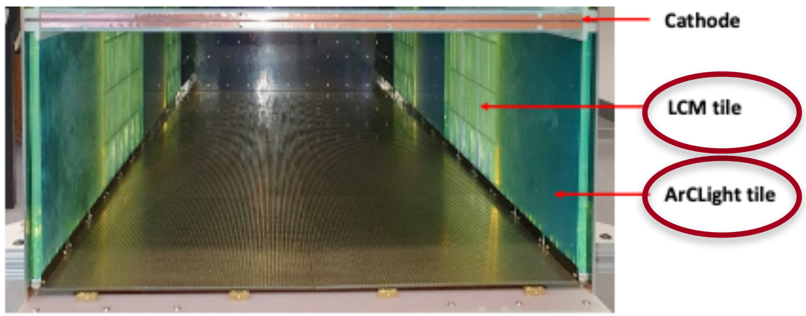


- 2 drift volumes (TPCs)
- separated by a cathode plane
- 4x2 LArPix tiles per anode plane
- 70x70 pixels per tile
- pixel pitch 4.43 mm

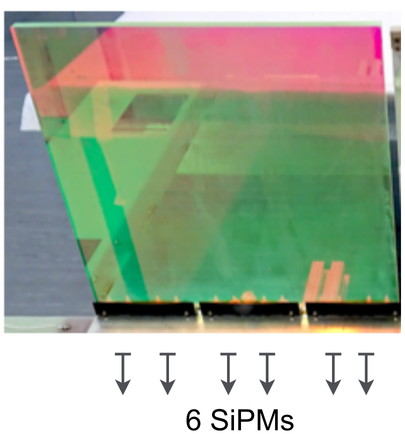
Module-0 Light Readout System



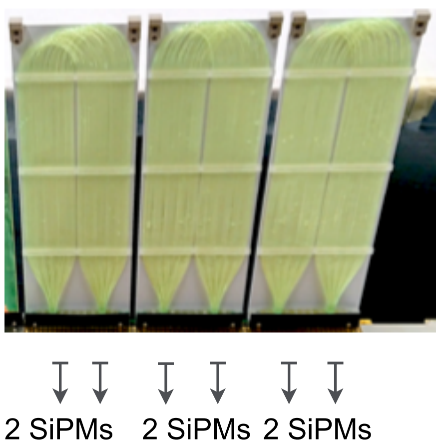
Light Readout System of Module-0



ArCLight tile

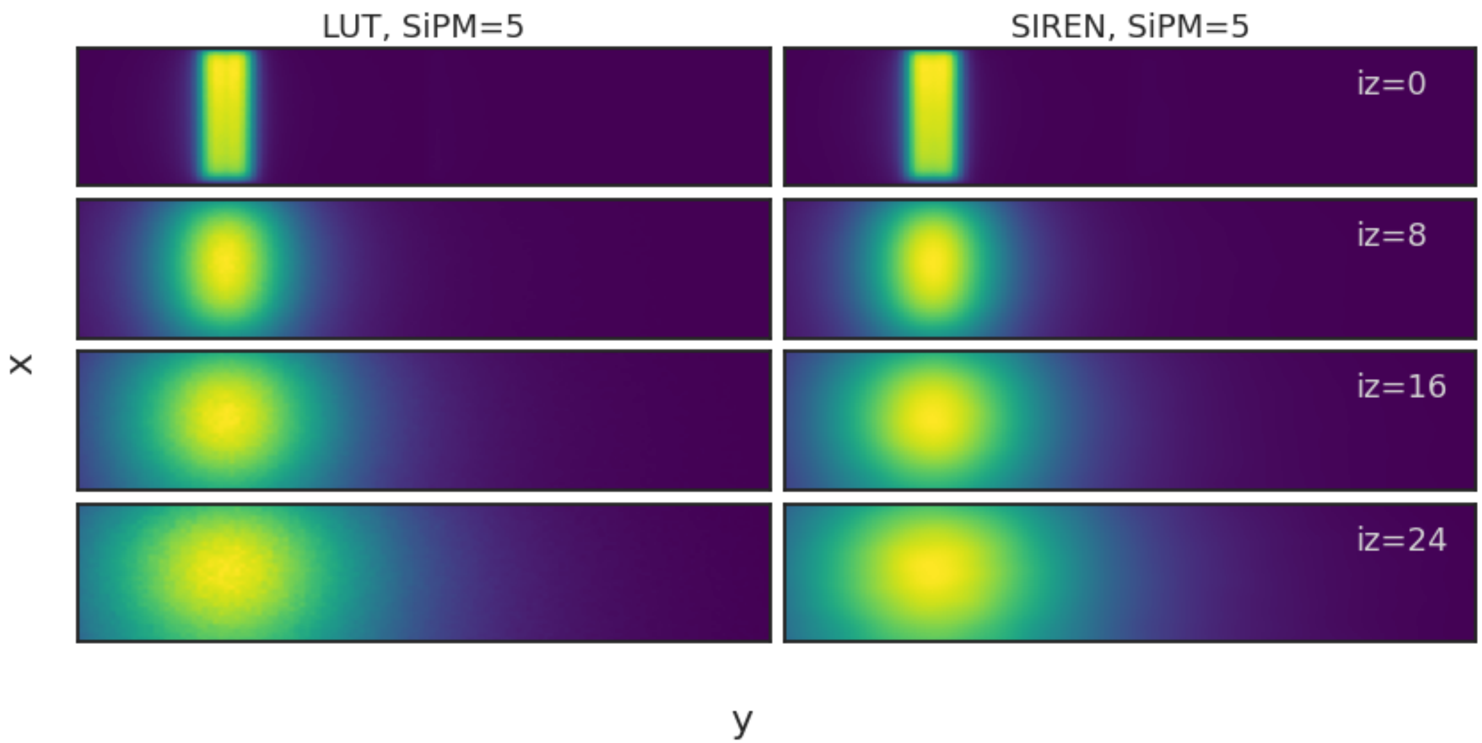


LCM tile



- 4 LCM and 4 ArCLight tiles per TPC
- each tiles ~300 mm x 300 mm x 10 mm
- 6 SiPMs per tile
- total of 48 SiPMs per TPC

Visibility: LUT & SIREN



- LUT is generated from photon simulation of 1 TPC in 64 x 128 x 32 voxels (~12.5M parameters)
- SIREN has 5 hidden layers and 128 hidden features (~89.3k parameters)
- relative bias is at few % level, dominated by the statistical uncertainty of the LUT

From Simulation to Data

Problem

- input data as *point sources* {x, y, z, visibility}
- obtained from optical detector *simulation*
- *not* available in data (no point-like calibration source)

Has to work with physics objects (i.e. *tracks*) from data

Modeling Optical Detector Readouts

$$\lambda_j = \sum_{t \in T} Y_j \times \Delta E_t \times \Psi_j(\mathbf{x}_t)$$

- Expected p.e. for the j-th PMT → λ_j
- combinations of light yield, PMT efficiency etc … → Y_j
- visibility of j-th PMT at the voxel coordinates x_t → $\Psi_j(\mathbf{x}_t)$
- sum over all voxels occupied by track(s) → $\sum_{t \in T}$
- energy deposition in the t-th track voxel → ΔE_t

Poisson Likelihood

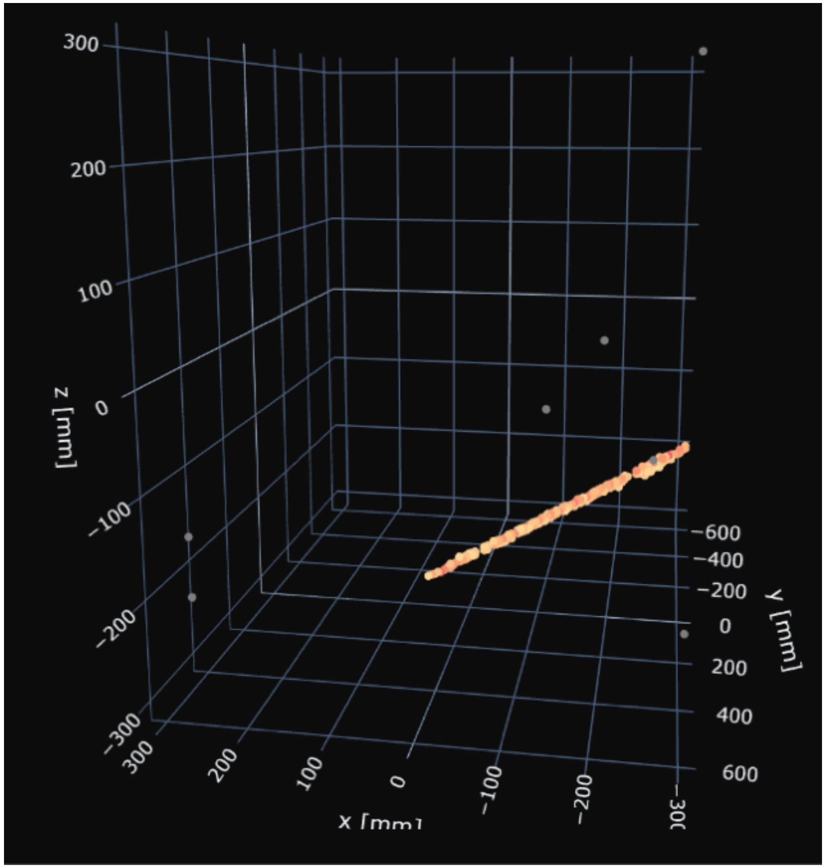
$$\mathcal{L}_{\mathrm{track}} = \prod_{j=1}^{N} \mathrm{Pois}(n_j | \lambda_j)$$

- product of all PMTs → $\prod_{j=1}^{N}$
- Observed p.e. for j-th PMT → n_j

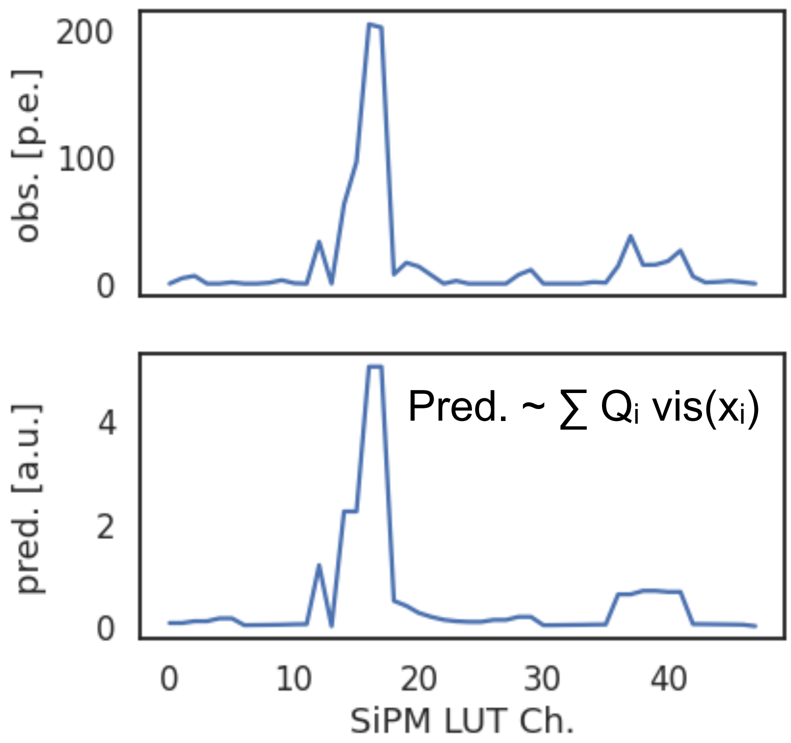
Optimization of *-ln* L_{track} using *track data*.

Charge-to-Light Prediction

An example of charge image from Module-0 data



Observed and Predicted Light Signal



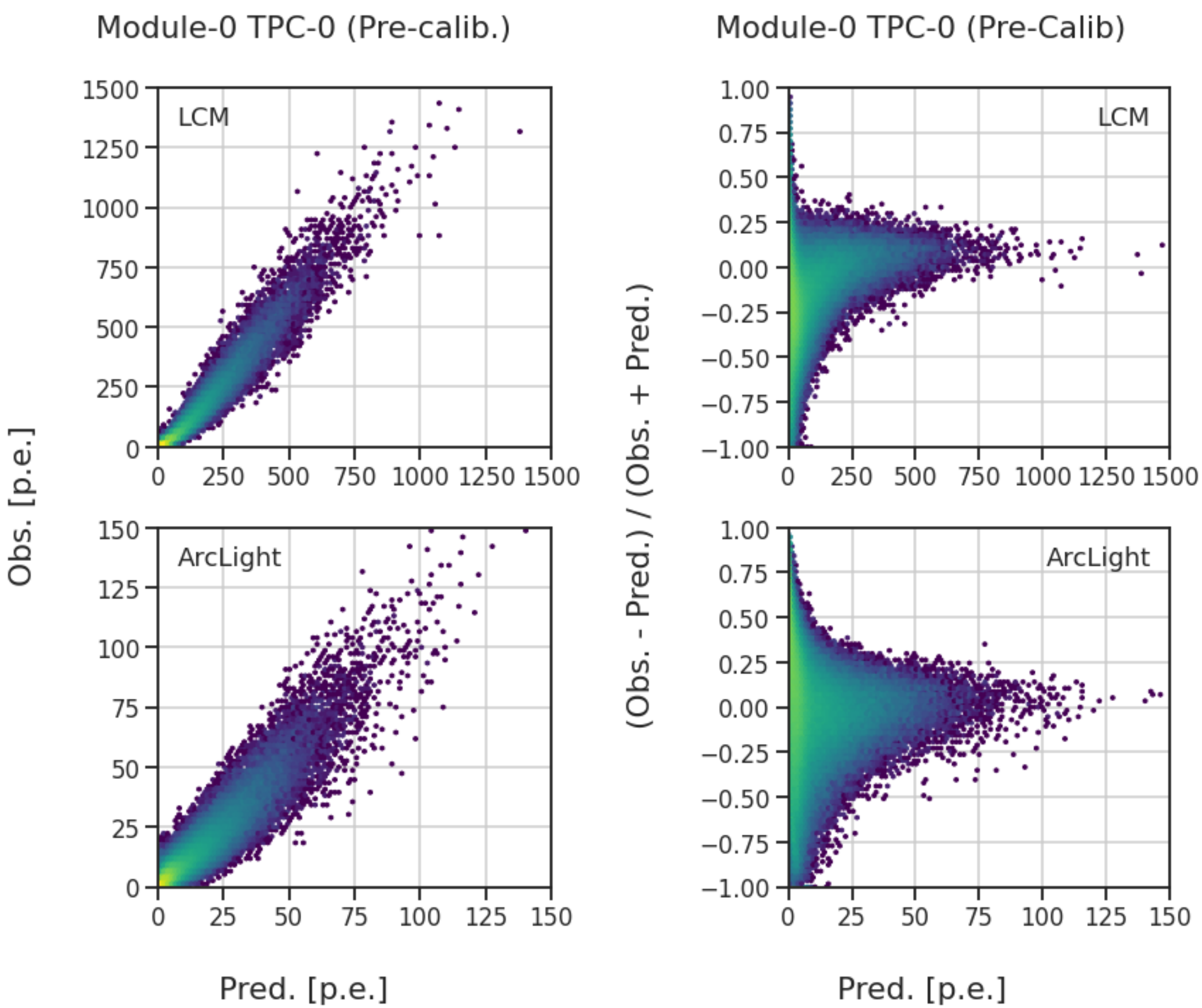
Dataset

- data collected between 4/4/21 - 4/10/21 at Bern
  - “default” settings (0.5 kV/cm, med. threshold ….)
- cathode-anode crossing tracks in TPC-0
  - one clustered object per charge image
- matching charge-light pairs by trigger timestamp

Training sample: ~670k tracks

Testing sample: ~13k tracks

Performance (Before Calibration)



In general, the prediction match to the light observation.

LCM: obs. p.e. is ~20% higher than prediction

ArcLight: better agreement than LCM, almost linear.

Calibration !!!

Calibration of SIREN Model

Calibration => Multi-parameters optimization problem of the SRIEN model

<u>Objective</u>: minimize the difference between observation and prediction

scaling factors for 48 SiPMs (floating)

visibility given by SIREN ~89.3k parameters (floating)

Prediction: $\lambda_j = \sum_{t \in T} Y_j \times \Delta E_t \times \Psi_j(\mathbf{x}_t)$

measured charge of the track (fixed)

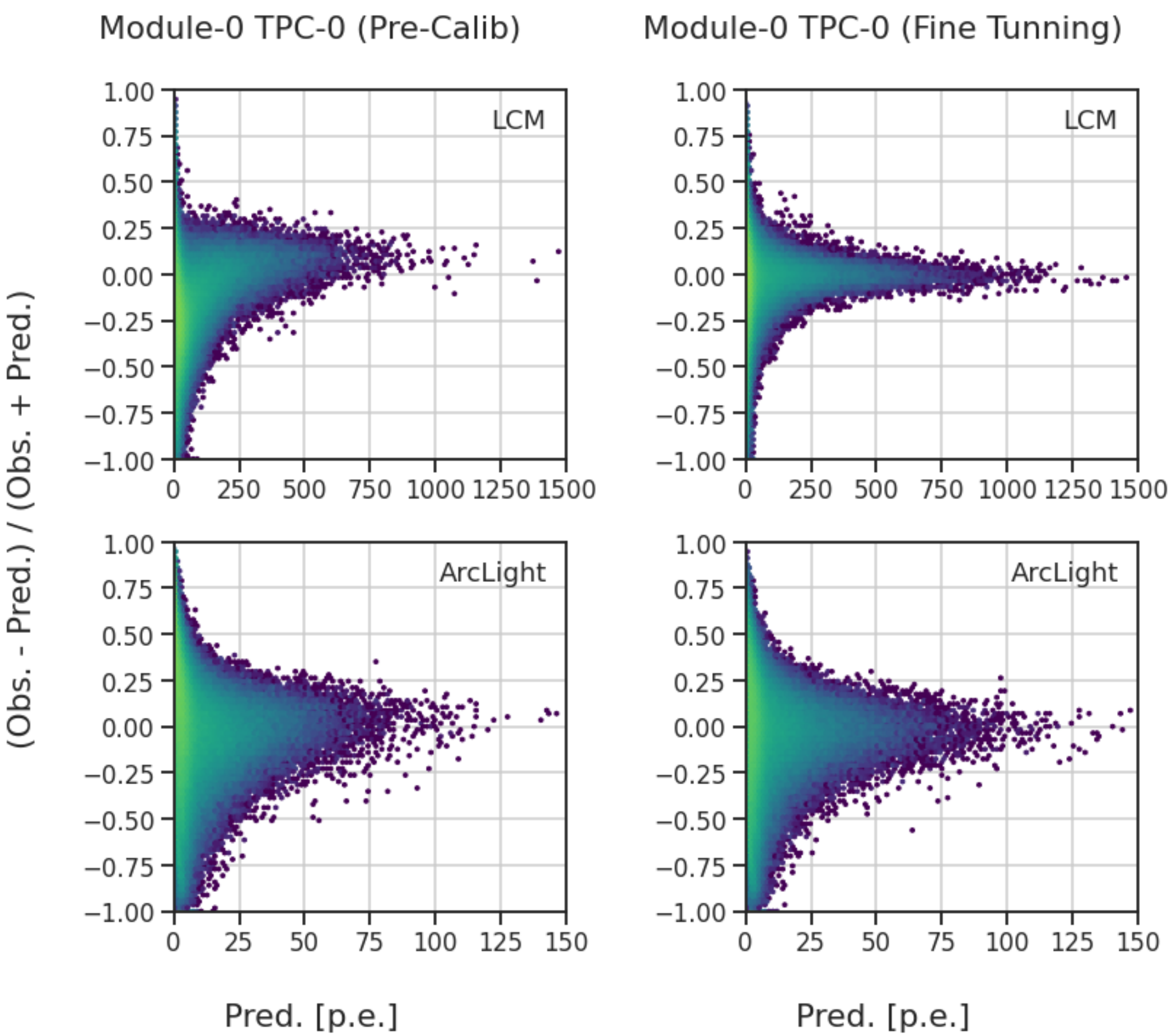
locations of charge deposition (fixed)

"Loss function": chi2 = ∑ (obs - pred)2 / (pred + ε)

ε = 5 p.e.

Calibration: Before and After

SLAC

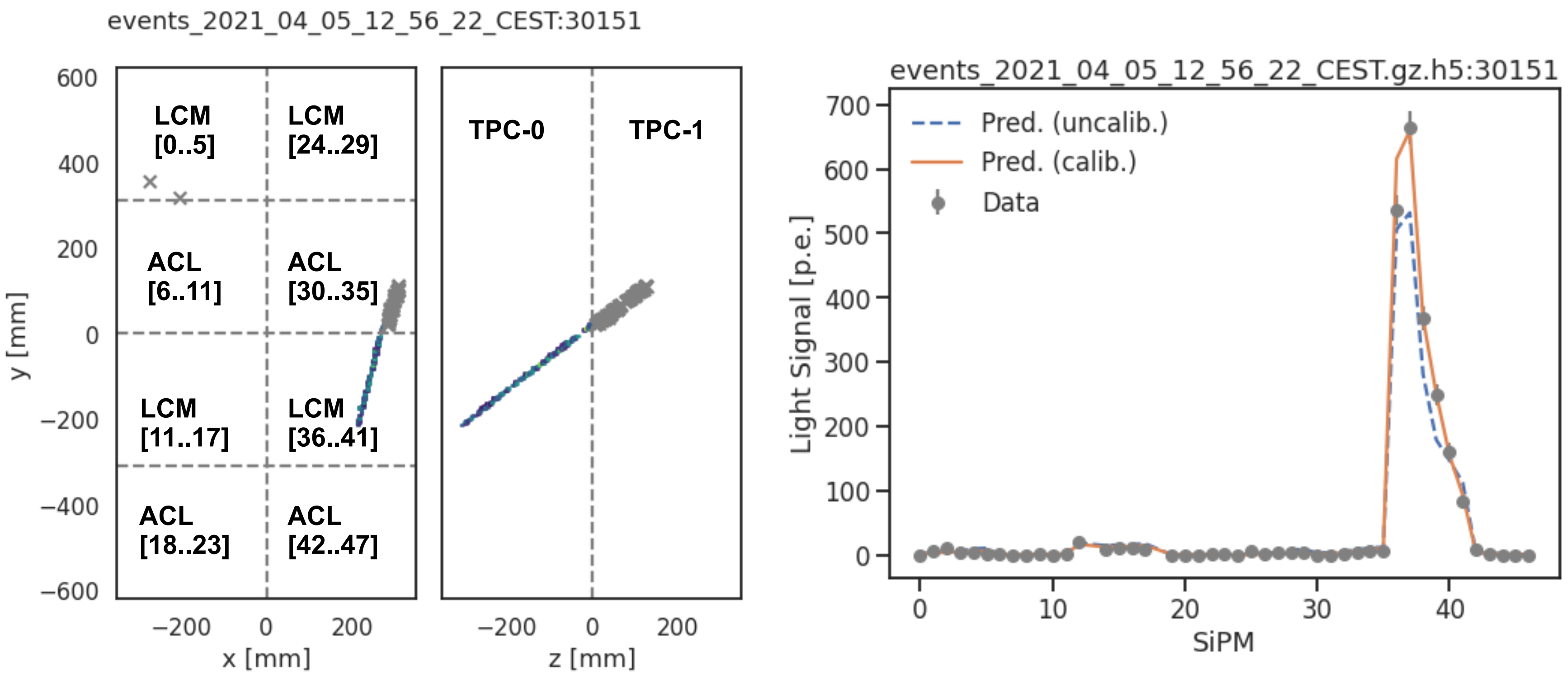


Pre-trained a SIREN model using the LUT.

Calibrate (fine-tune) the SIREN model using track *training* samples.

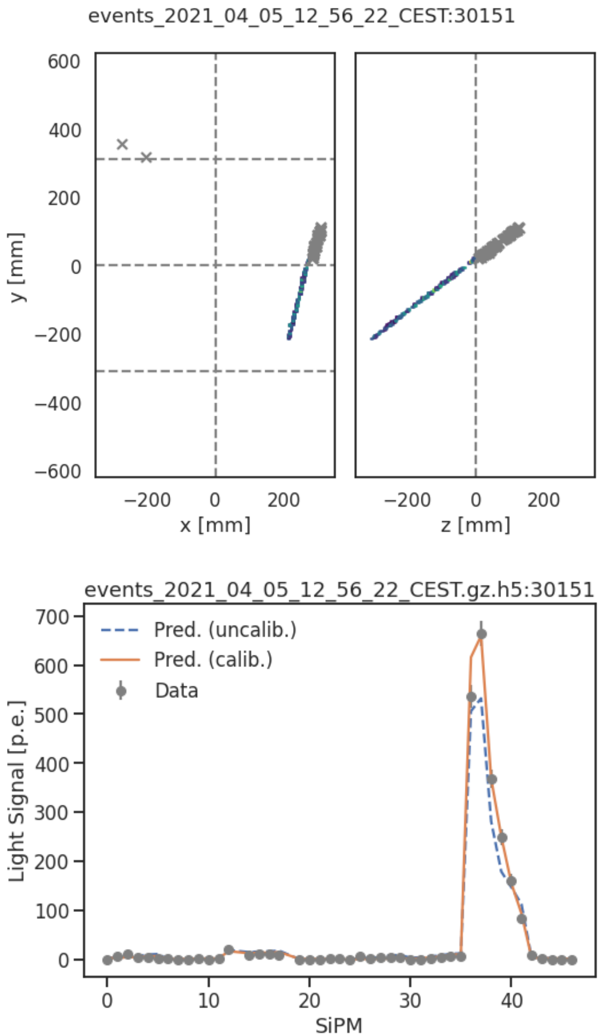
Evaluate the performance using *testing* samples.

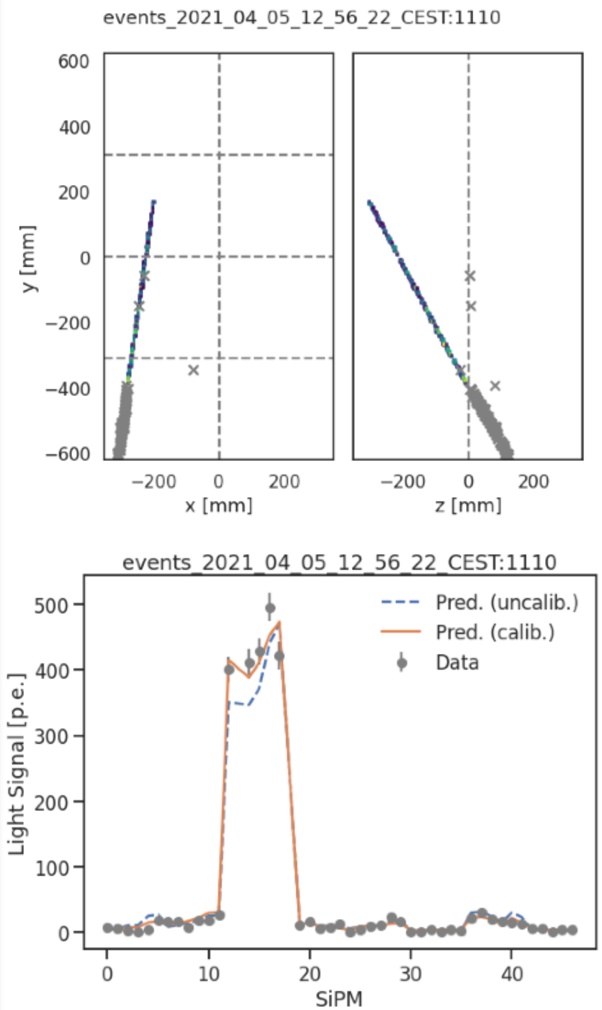
An Example Event

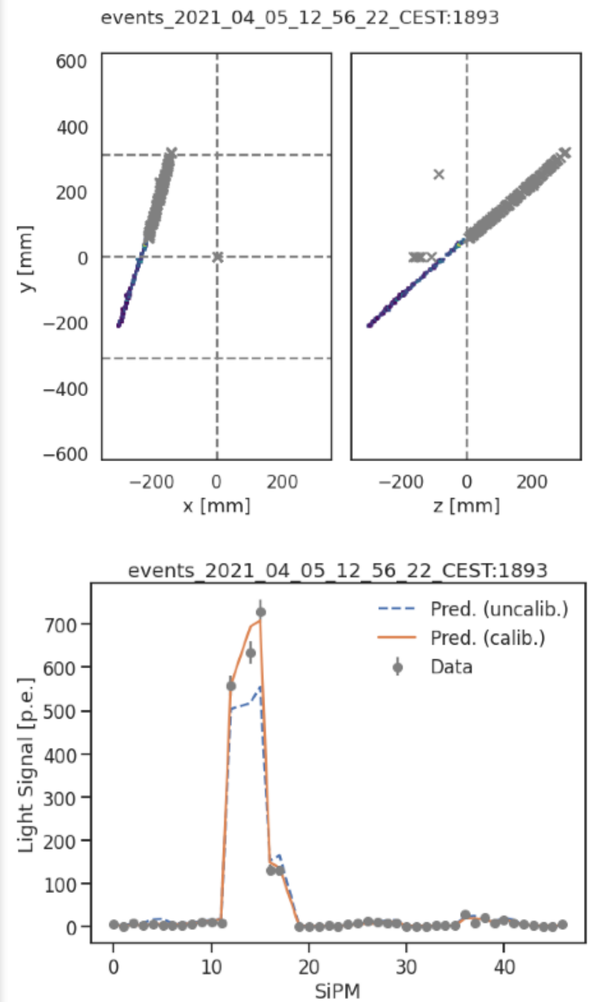


** Grayed out points are
excluded from the prediction.

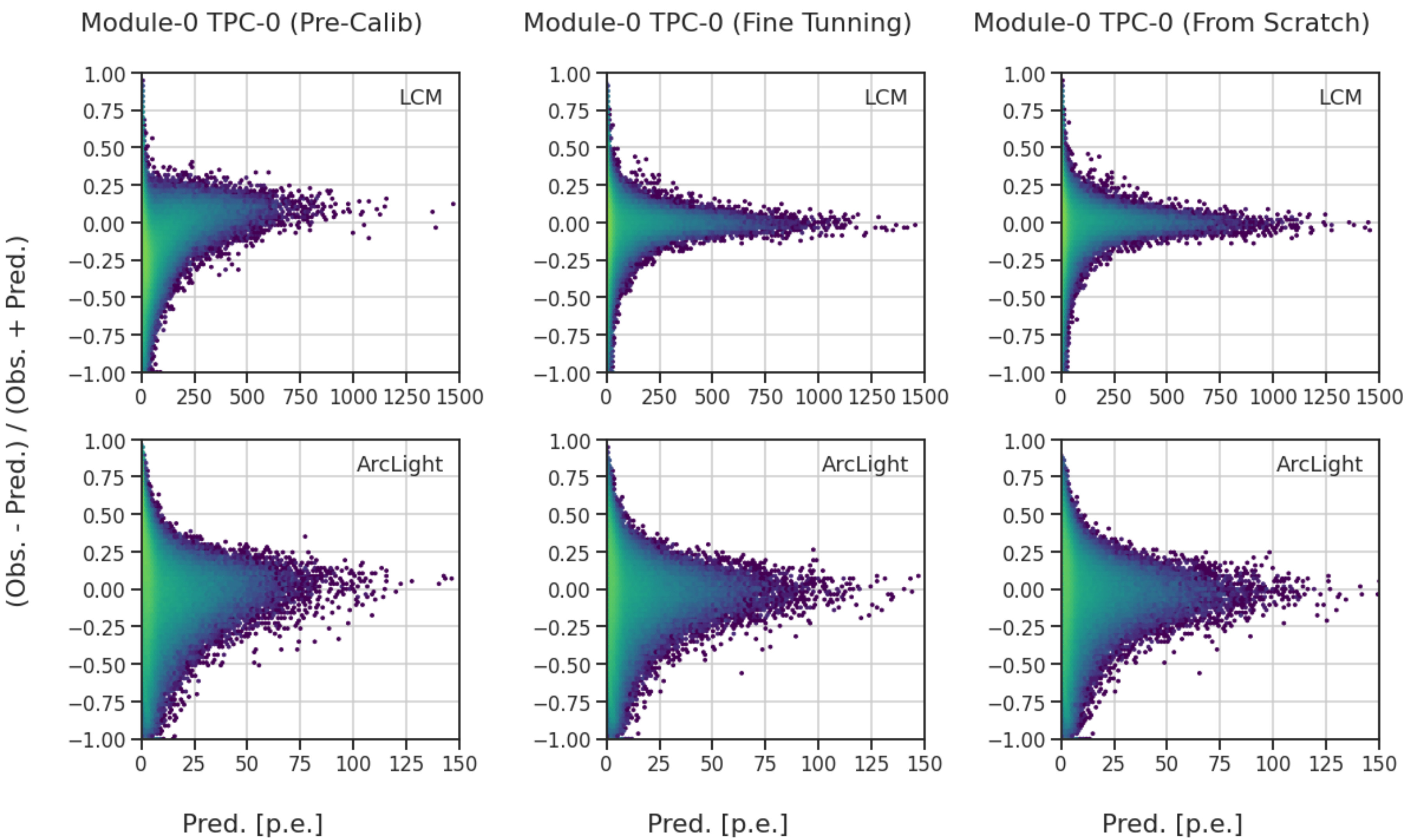
More Examples



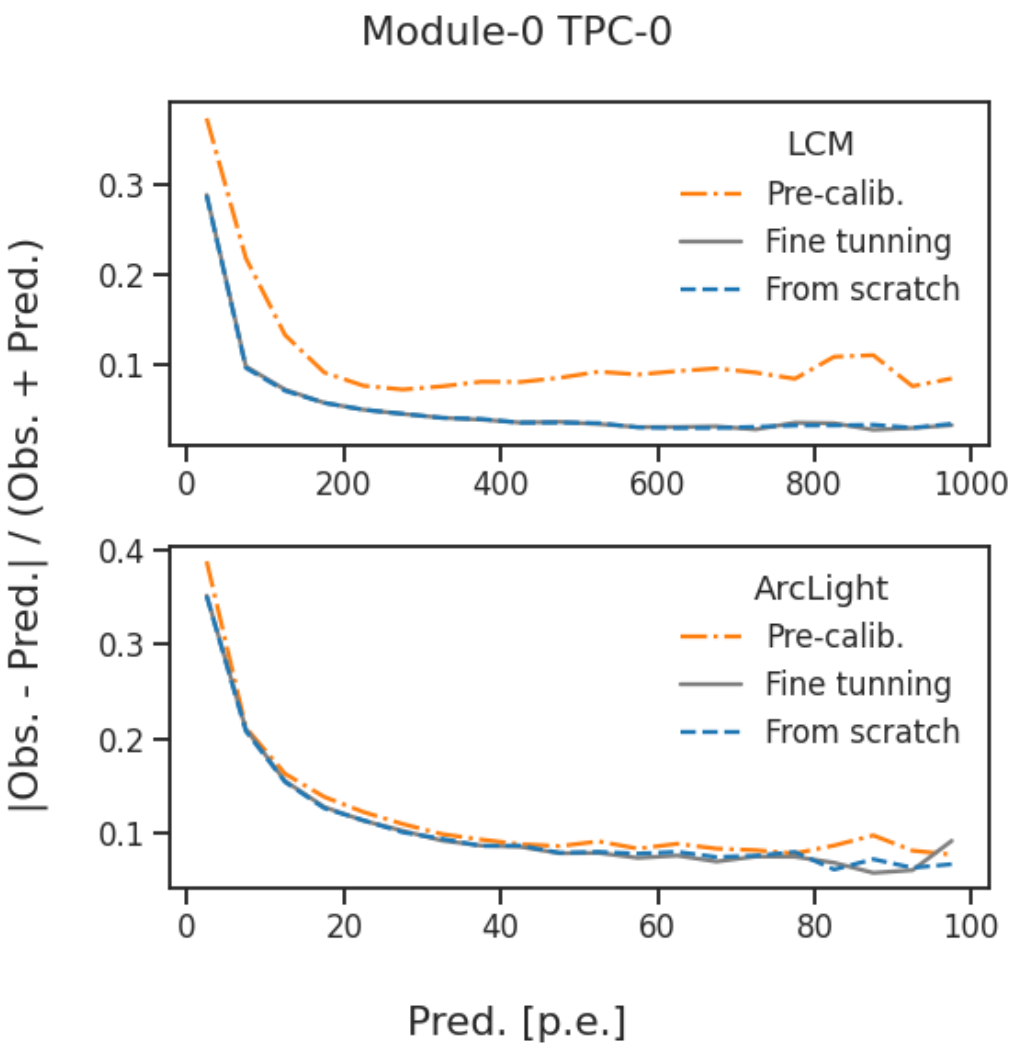




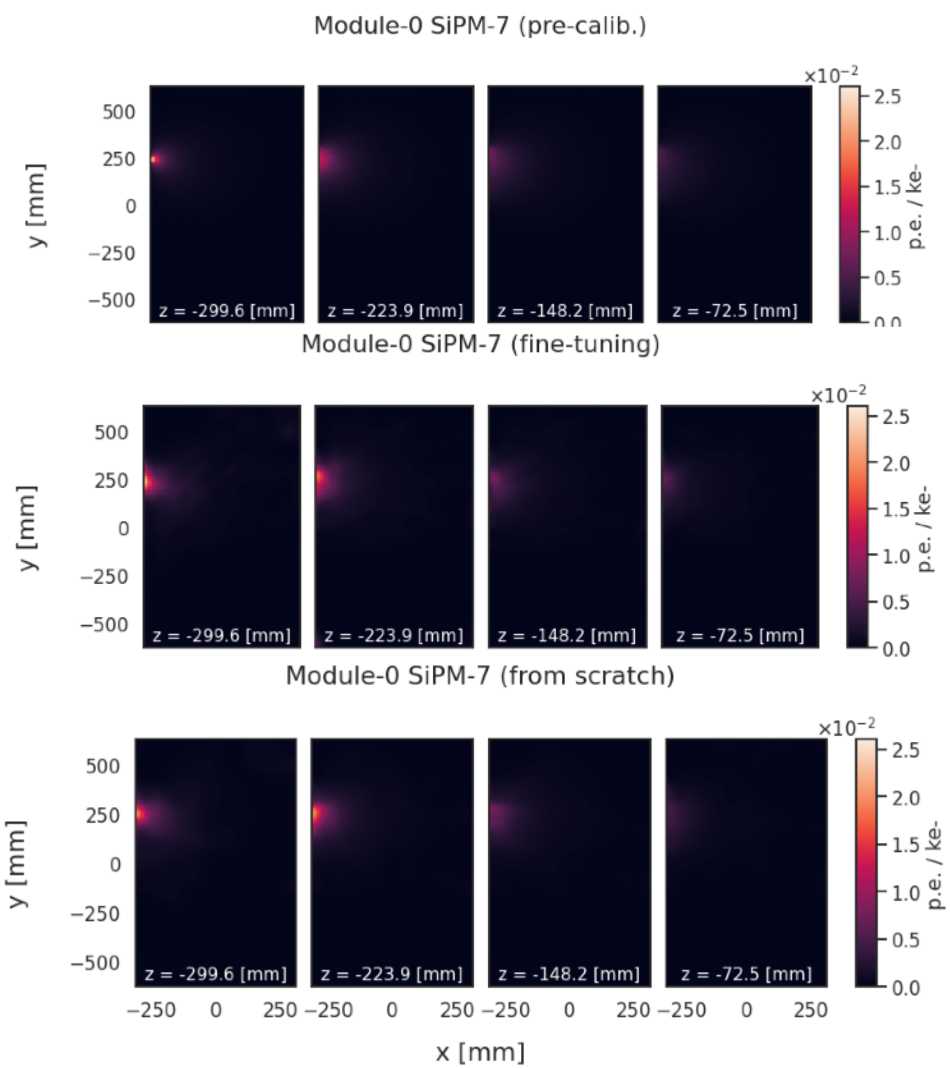
Construct a SIREN Model from Scratch Using Track Sample

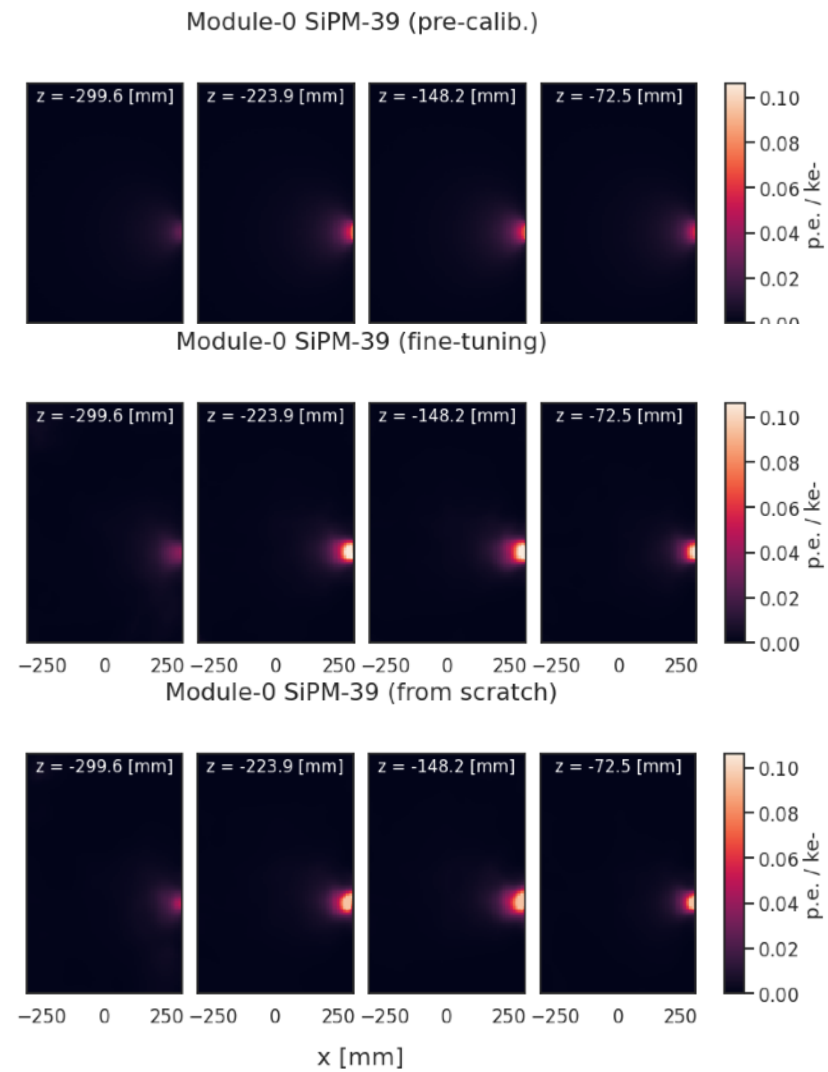


SIREN from Scratch



Inspection of Visibility Map





Publication in Progress

SLAC

Implicit Neural Representation as a Differentiable Surrogate for Photon Propagation in a Monolithic Neutrino Detector

Minjie Lei,[2,*] Ka Vang Tsang,[1,†] Sean Gasiorowski,[1] Chuan Li,[3] Youssef Nashed,[1]
Gianluca Petrillo,[1] Olivia Piazza,[4] Daniel Ratner,[1] and Kazuhiro Terao[1]
(on behalf of the DeepLearnPhysics Collaboration)
[1]*SLAC National Accelerator Laboratory, Menlo Park, CA, 94025, USA*
[2]*Stanford University, Stanford, CA, 94305, USA*
[3]*Lambdalab Inc., San Francisco, CA, 94107, USA*
[4]*University of California, Berkeley, CA, 94720, USA*

Optical photons are used as signal in a wide variety of particle detectors. Many modern neutrino detectors employ hundreds to tens of thousands of photon detectors to observe signal from millions to billions of scintillation photons produced from energy deposition of charged particles. These neutrino detectors are typically large, containing $\mathcal{O}(10^2 \approx 10^5)$ tonnes of target volume, and may be consist of many materials with different optical properties. These natures make modeling of individual photon propagation challenging as it would need a prohibitive amount of computational resource. As an alternative to tracking individual photons, the experimental community has traditionally used a *look-up table*, which contains a mean probability of observing a photon per photon detector at each

A proof-of-concept study with ICARUS simulation [arXiv:2211.01505]

Summary

- demonstration of SIREN with module-0 data
- on-going studies
  - how many tracks are needed to construct a SIREN model?
  - application of gradient surface (e.g. t0-finding)
  - uncertainty qualification
- aiming a dedicated paper for module-0 data