Differentiable surrogate for modeling the physics of optical propagation in a LArTPC

Sam Young
youngsam@stanford.edu

with Carolyn Smith

for the DUNE Collaboration

DUNE ML reco meeting

Mar. 27 2024

SLAC NATIONAL ACCELERATOR LABORATORY

Stanford University | U.S. DEPARTMENT OF ENERGY

Optical photon transport in LArTPCs

- **Photon detectors (PMTs, more recently SIPMs)** detect scintillation photons produced isotropically from Ar

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

ICARUS detector

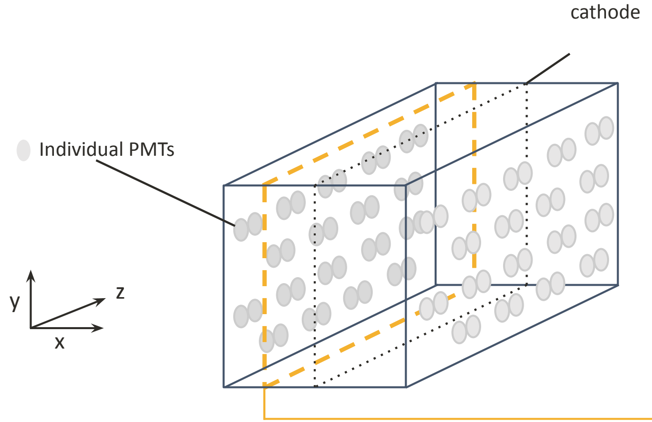
Optical photon transport in LArTPCs

- **Photon detectors (PMTs, more recently SIPMs)** detect scintillation photons produced isotropically from Ar
- 1 meter muon produces approx. **5 million photons**

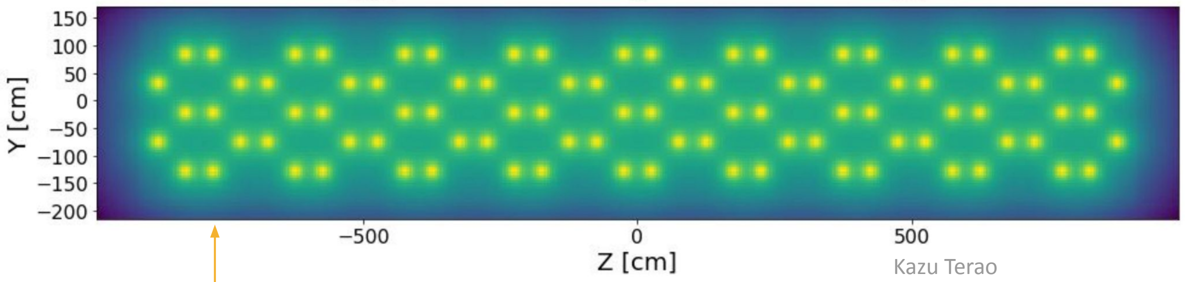
ICARUS detector

Optical transport simulation: **Lookup tables**

- Issue: Simulating each individual photon is too slow
- Conventional method: lookup tables (LUT)
  - Divide the detector volume into voxels of *O*(cm) in size
  - For each voxel, simulate and propagate millions of photons using conventional simulation software (i.e. **`larnd-sim`**)
  - Visibility @ [x,y,z] = # detected photons / # generated photons



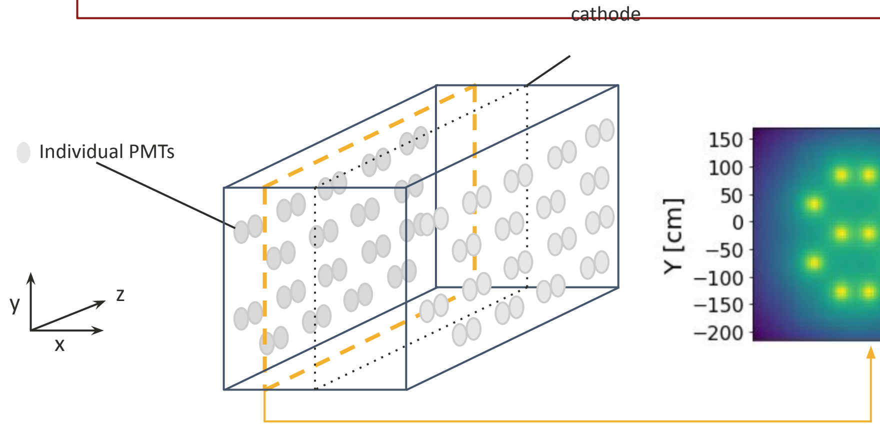
ICARUS detector, 2D slice of 3D visibility map



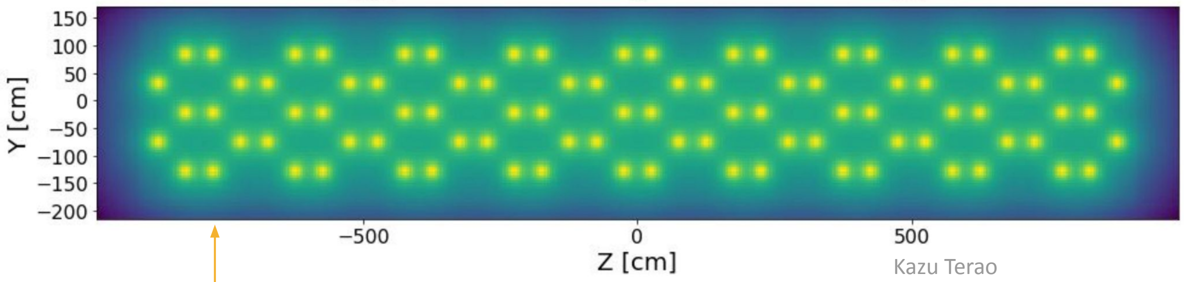
Kazu Terao

Optical transport simulation: **Lookup tables**

- Superseding issue:
    - Not scalable for larger detectors, $N_{voxel} \sim V$
    - Coarse voxel sizes (5 cm)
    - Large statistical error (~30k photons/voxel)
    - Limited by memory usage and time (~2 weeks to produce a single LUT for ICARUS)



ICARUS detector, 2D slice of 3D visibility map



Kazu Terao

Differentiable neural scene representations

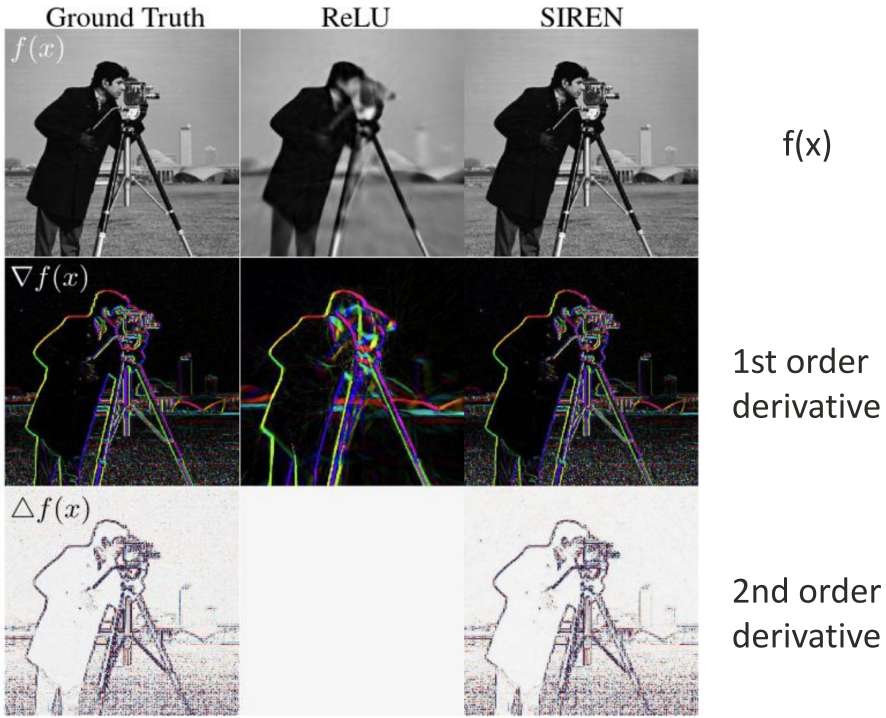
Implicit neural representation:

- Parametrize signals as **continuous functions**
- Neural networks are trained to map the domain signal (e.g., x,y,z) to the target outputs (e.g., signal at x,y,z)

$$f : \mathbb{R}^M \to \mathbb{R}^N$$

SIREN:

- MLP with periodic **sine** activation functions
- Models gradients (and higher order gradients) smoothly
- Continuous and differentiable

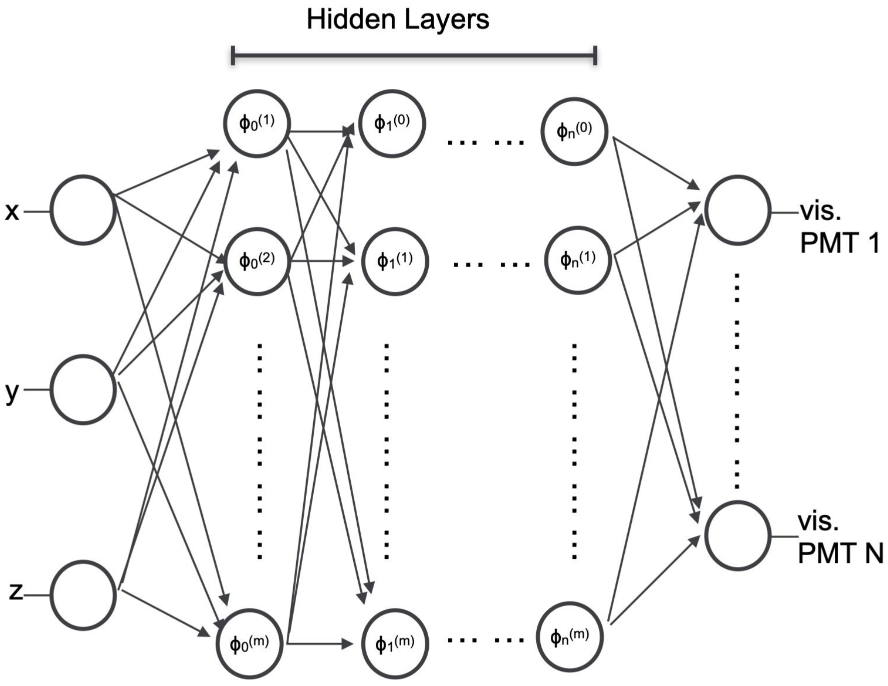


Sitzmann et al., arXiv:2006.09661

Optical transport simulation: **SIREN**

- Models visibility per PMT as a **spatially dependent distribution**.
  - Models higher order gradients smoothly (suitable for visibility, E-field, …)
  - Trade off between an analytical function and a table
- **Continuous and differentiable**
  - Can optimize directly against data vs. simulation discrepancy
- Smaller in size:

LUT (2x2)	SIREN
32 x 128 x 64 voxels	5 hidden layers
48 channels	512 hidden features
404 million params	**1.5 million params**

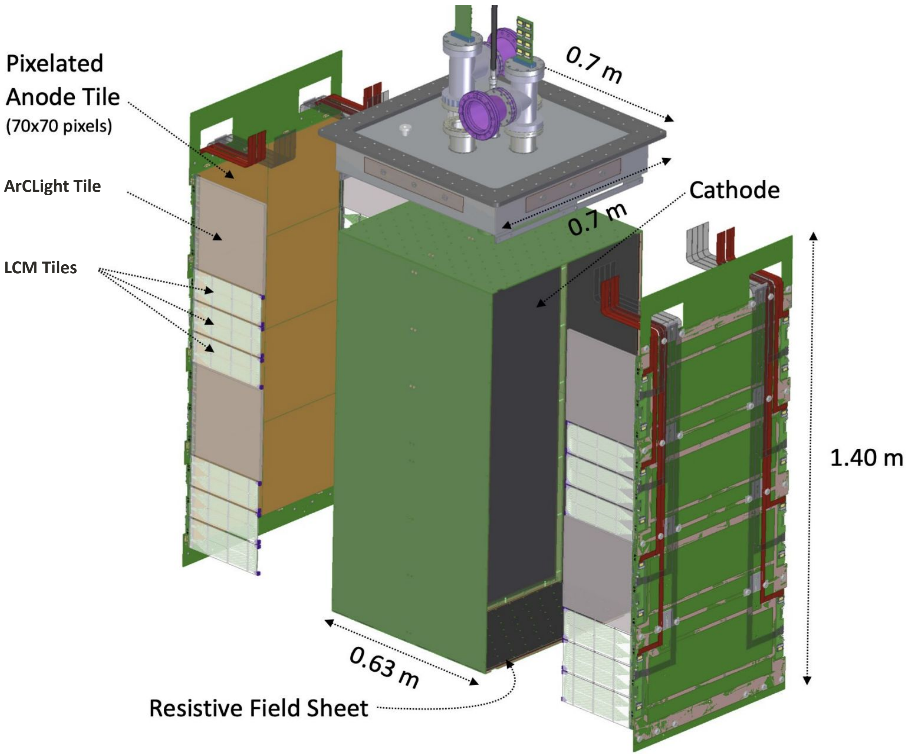


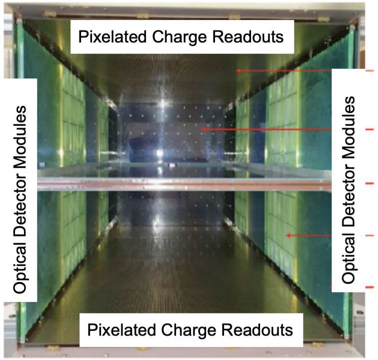
$$\mathbf{\Phi}(\mathbf{x}) = \mathbf{W}_n(\phi_{n-1} \circ \phi_{n-2} \circ \cdots \circ \phi_0)(\mathbf{x}) + \mathbf{b}_n,$$
$$\phi_i(\mathbf{x}_i) = \sin(\mathbf{W}_i\mathbf{x}_i + \mathbf{b}_i),$$

SIREN for ND-LAr 2x2

- 2x2 demonstrator:
  - 4 modules
  - Each module divided into two TPCs
  - ~0.7 x 0.7 x 1.4 m
  - 2 different optical detector prototypes: LCM and ArcLight
  - Modules 1-3 have reflective coating increasing light yield, while module 0 does not.

A single module





SIREN vs. LUT

- We train a single SIREN across similarly configured TPCs.
- Two SIRENs, one for low visibility TPCs (module 0), one for high visibility TPCs (module 1-3)
- **As a proof of concept, we train on individual voxels of the LUT**
- LUT (Left): (32,128,64) voxels, 48 channels per module
- SIREN (Right): 3 layers, 512 hidden features
  - Trained for 1m epochs
- Voxel-wise loss:

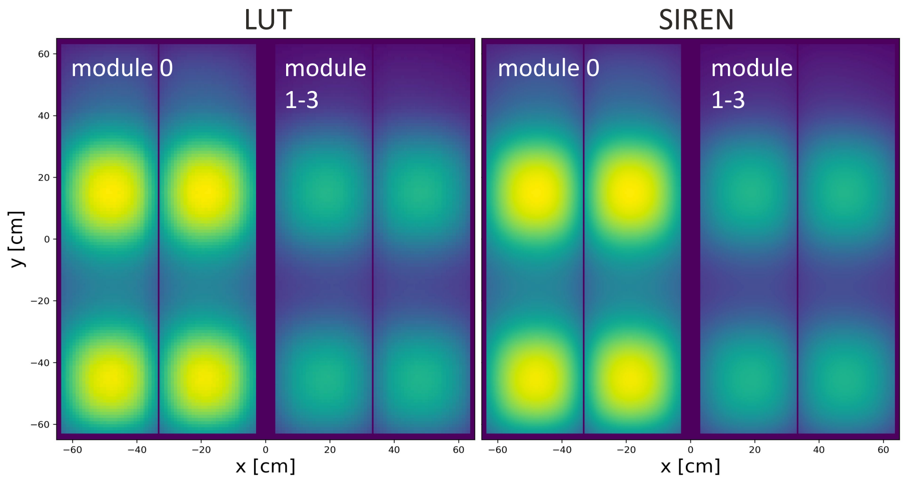
$$\mathcal{L}_2 = \frac{1}{N}\sum_{\mathbf{x}_i}\sum_{j=1}^{N_{\mathrm{pmt}}} w_{ij}[\tilde{\Psi}_j(\mathbf{x}_i) - \tilde{v}_{ij}]^2$$

w_{ij}: Precomputed weight

$\tilde{\Psi}_j(\mathbf{x}_i)$: Log-transformed **predicted visibility** of ith voxel position on jth PMT

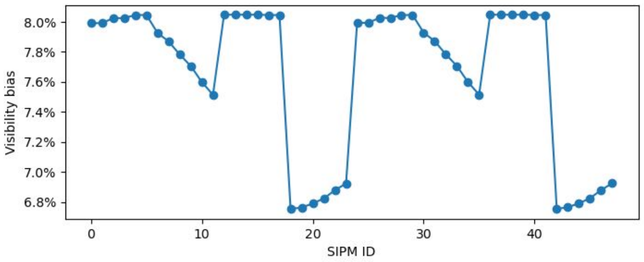
$\tilde{v}_{ij}$: Log-transformed **observed visibility** of ith voxel on jth PMT

Visibility slice of 2x2 prototype at Z=10 cm



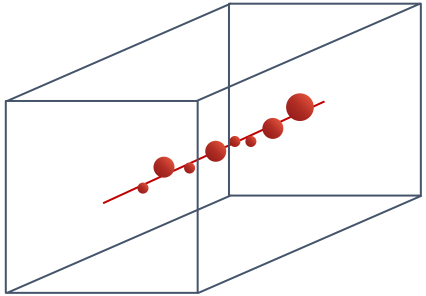
Average relative visibility bias across SIPMs

$$\frac{1}{N}\sum_{\mathbf{x}_i}\frac{|\Psi_j(\mathbf{x}_i) - v_{ij}|}{(\Psi_j(\mathbf{x}_i) + v_{ij})/2}$$



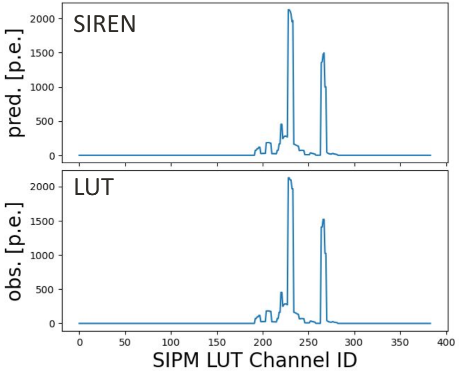
Train via charge-to-light prediction

- Point-like source (i.e. visibility at coordinates in the detector) isn't accessible in data
- We can infer the light signal from physics objects (e.g., tracks) that **are** accessible.



Detected track $(x,y,z,Q)_i$

Photoelectrons (PE) detected
$\sim\sum Q_i \cdot \text{visibility}(x_i,y_i,z_i)$



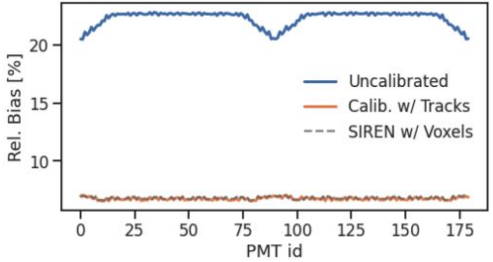
Poisson Likelihood

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

product over all PMTs

obs. PE for jth PMT

pred. PE on jth PMT



See also:
Carolyn Smith (next talk) on reconstructing tracks using this method

Conclusion

Github: CIDeR-ML

- SIREN can reproduce **both the values and gradients** of a visibility LUT with *O*(>100x) less parameters.
- It can be **directly trained from data** using tracks from simulation or real data.
- Unlike the lookup table, **it is scalable** for massive detector volumes like the upcoming DUNE experiment.
- This technique can be used in general to model some spatially dependent process, especially in a homogeneous region (e.g., LAr, LXe, WCh, LS, …).
- **See Carolyn Smith's talk (next talk!) on how to use SIREN to reconstruct track positions in the 2x2 prototype and match tracks to their respective optical signals.**
- Check out the original paper for SIREN in LArTPCs: arXiv:2211.01505

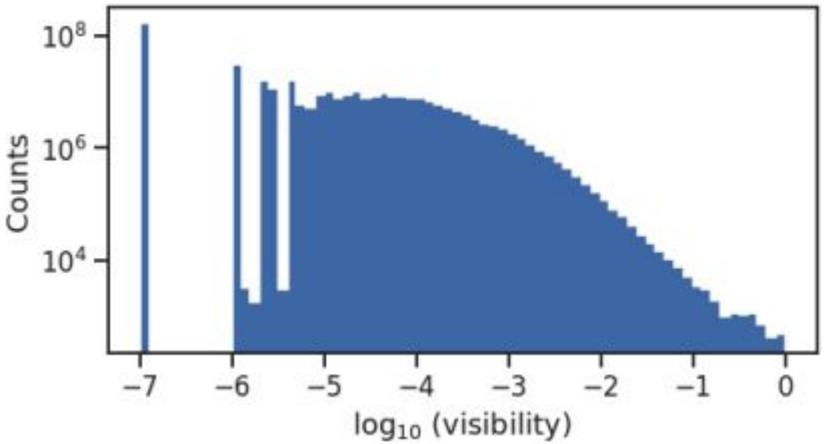
Backup

Log transforming visibilities

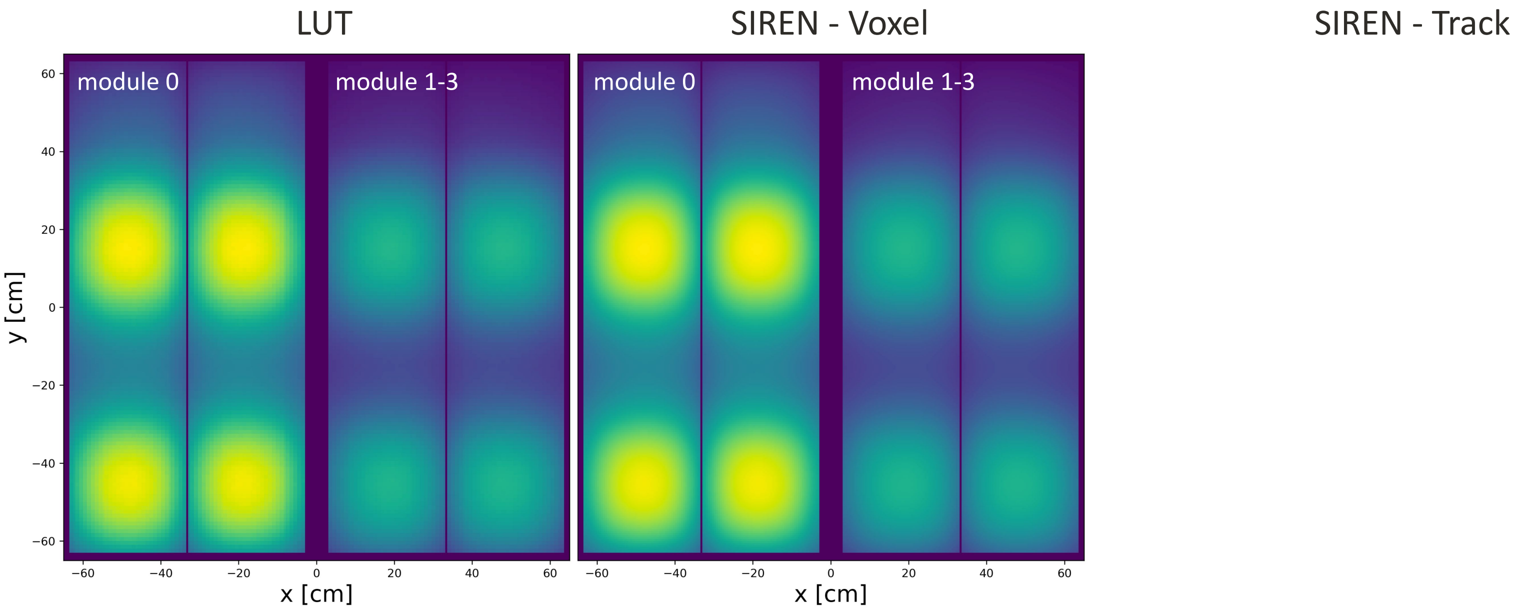
- Visibilities span several orders of magnitude

$$\tilde{v}_{ij} = \log_{10}(v_{ij} + \epsilon)$$

Distribution of visibility of ICARUS LUT

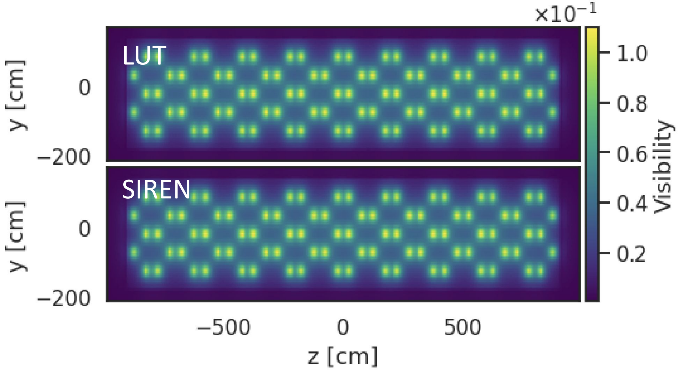


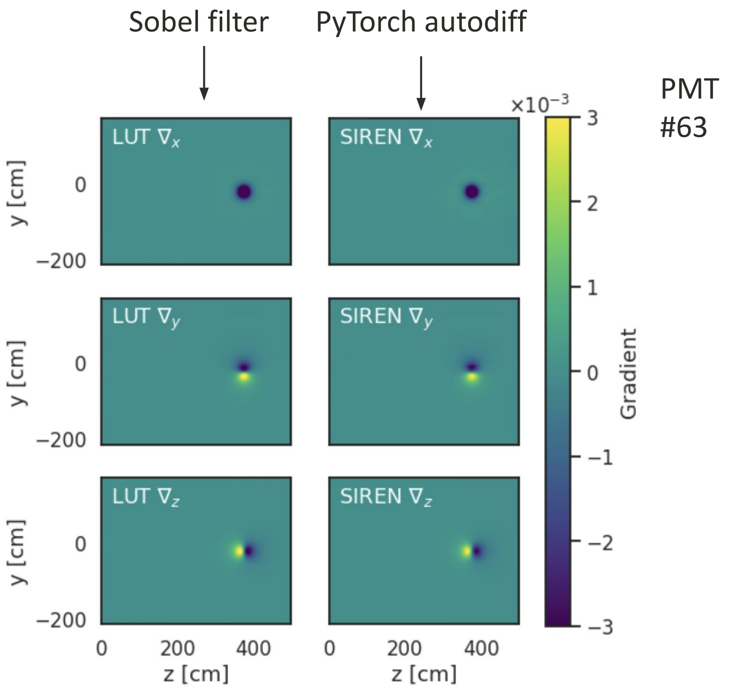
Voxel-trained v. track-trained SIREN



Gradients

- Example SIREN trained on ICARUS LUT, slice at x=-362.5 cm

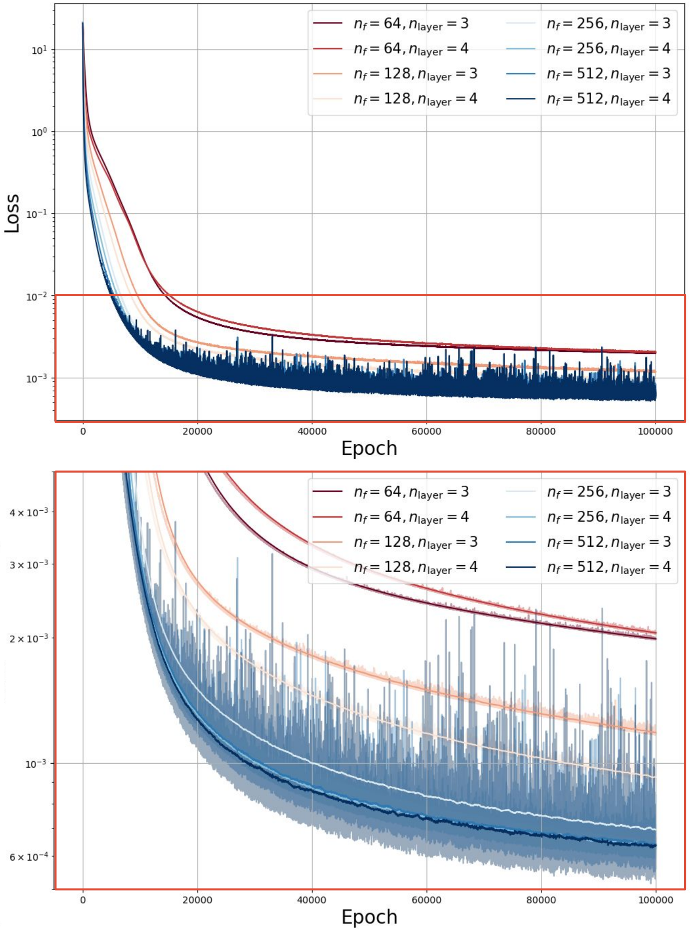




SIREN-LArTPC Hyperparameters

- Voxel-wise weighted MSE loss:
  - $MSE(pred,obs,w) = 1/N \sum_i w_i(pred_i-obs_i)^2$
  - Weight higher visibilities (more important) higher than lower visibilities.
- Based on hyperparameter search, we find 512 hidden features with 3 hidden layers is reasonable
- Use LR of 2e-06, Adam optimizer

n_f, n_{layer} search using module 0 loss



LR search