Generative Model Applications 3: Liquid Argon TPCs

SLAC SSI 2023

Taritree Wongjirad (Tufts University)

Tufts University

The NSF Institute for Artificial Intelligence and Fundamental Interactions

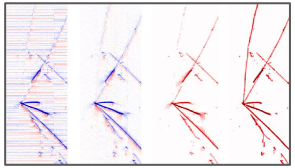
NSF

Department of Energy United States of America

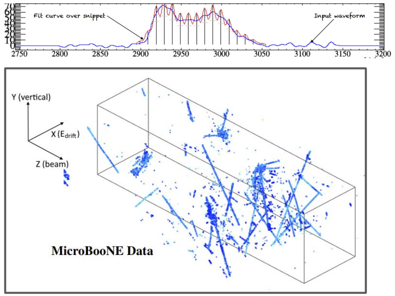
ML Reconstruction

Last week I talked primarily about Reconstruction for LArTPCs where

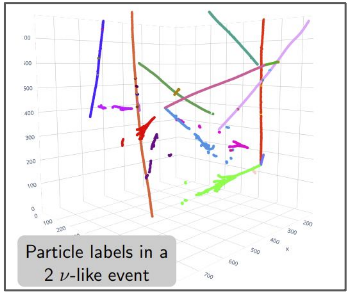
Deep neural networks having success in identifying key physical quantities for studying neutrino interactions



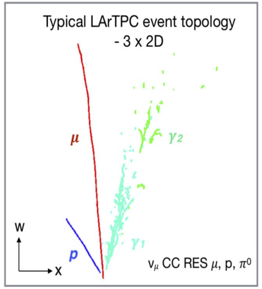
Signal conditioning



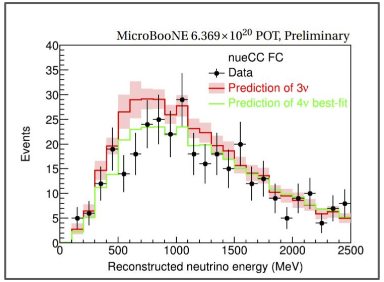
"Hit" finding
(first-stage summary)



Clustering+particle ID



Interaction



Measurement
Observable

Challenges

This talk will touch on early work to eventually address other challenges through generative models

ation that we can
from our data, precisely and accurately?

Simulation/Modeling: can we translate physics to observables faster? Can we better use data-driven methods?

Inference: are we testing our models against data as best as we can while accounting properly for and mitigating our model uncertainties?

Operations: are we saving the right events? Is the experiment running optimally? Can we detect and make decisions faster?

Fast ML on FPGAs to implement special rare process triggers

frai.2022.855184

Challenge: Surrogate Simulations

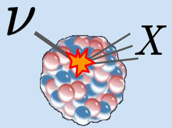
Simulation/Modeling: can we translate physics to observables faster? Can we better use data-driven methods?

Like in other frontiers, generating simulated data is a major bottleneck of analyses

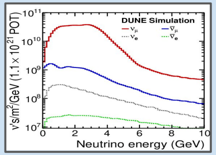
- Example: Individual events can take ***5-10 mins/event+*** for MicroBooNE TPC with simulated cosmics + neutrino interaction
- Data driven methods are used to get better estimate: in MicroBooNE and SBN experiments which are on surface, cosmic background data is collected and used in simulated data by adding neutrino interaction – but cannot save enough of these events due to processing and storage constraints

Simulations for LArTPCs

Neutrino Argon Interaction Model



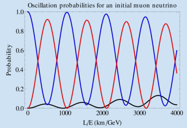
Neutrino Flux Model



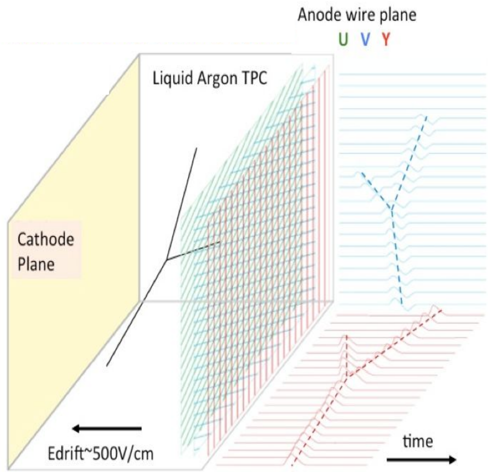
Null Hypothesis

Physics Model

$$P(E_\nu, \theta, \Delta m^2)$$

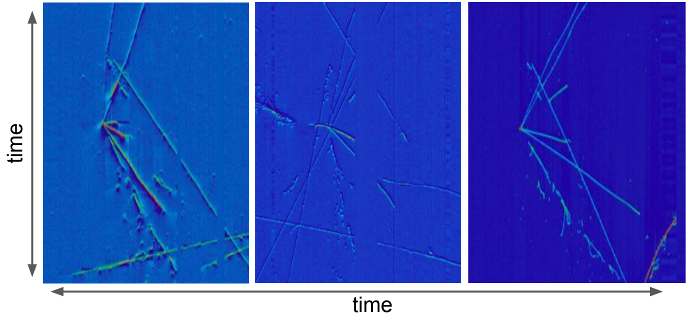


Detector Simulation



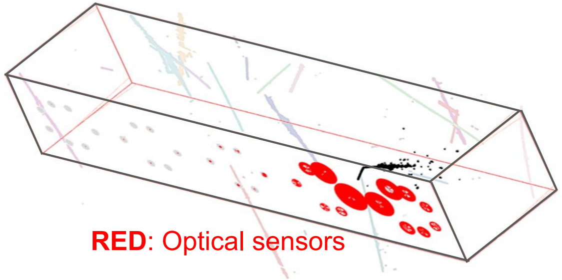
TPC Wireplane Data

(measuring ionization)



Optical Data

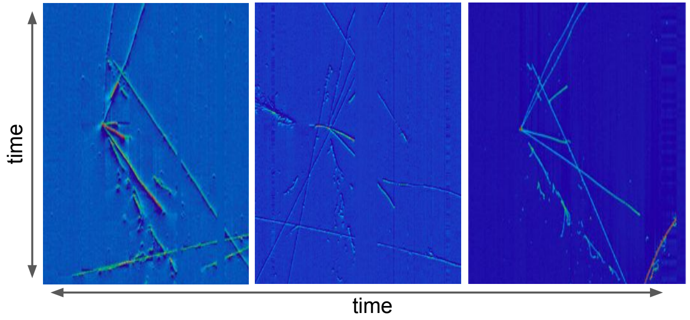
(measuring Scintillation photons)



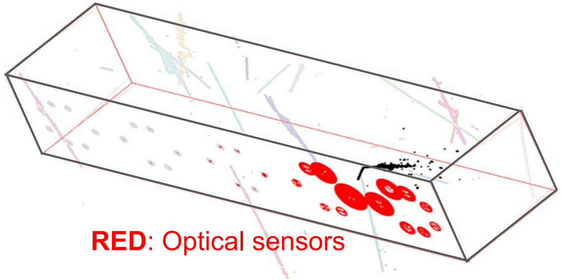
Outline

- Approach to improve **optical data** modeling using machine learning
- Early work to use generative models to reproduce **LArTPC wire plane images**

TPC Wireplane Data
(measuring ionization)



Optical Data
(measuring Scintillation photons)



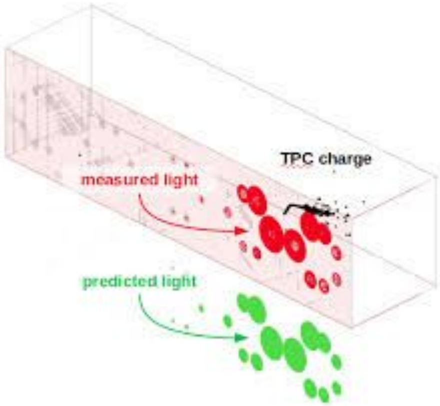
Optical Data

Liquid argon (and other cryogenic noble liquids) are excellent scintillators: 40k scintillation photons produced per MeV of deposited energy

Provides **measurement of time** when particles(s) traverse the detector

Useful for reconstruction: pattern of light helps with **same-time clustering** for ionization 3D spacepoints

There is some Particle ID information as well between **high** (proton, nuclei) vs **low** (electron, muon) density ionizing particles

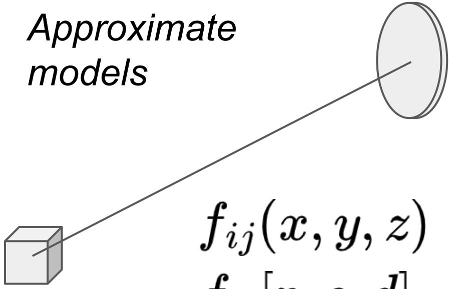


Matching pattern of sensor position and total charge to cluster spacepoints

Optical Simulation Challenges

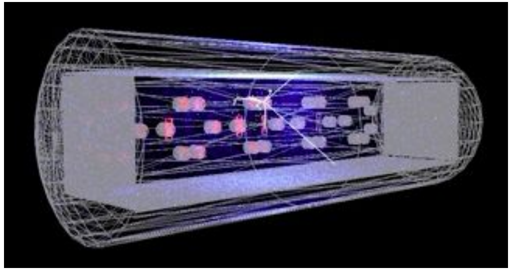
- CPU photon transport simulation for O(GeV) events is prohibitively expensive:
  Requires transporting O(10M) photons!
- Current standard practice is to use an approximate model
  - Generate table of output vs. position for voxelized data
  - Analytic function for mean sensor output vs. position
    [EPJC 81: 349 (2021)]
- Unknown parameters
  - Reflection/Transmission for 128 nm (VUV) scintillation photons not known for many materials/surfaces
  - Still some uncertainties in response of the wavelength shifter (TPB: converts 128 nm to detectable 420 nm light)

Approximate models



$$f_{ij}(x, y, z)$$
$$f_{ij}[r, c, d]$$

One direction: GPU photon sim



Opticks - S. Blythe   S. Siebert - Chroma

Neural Network Estimate

- Neural network learning function to map 3D position and photons emitted at that location to expected signal in the array of optical sensors

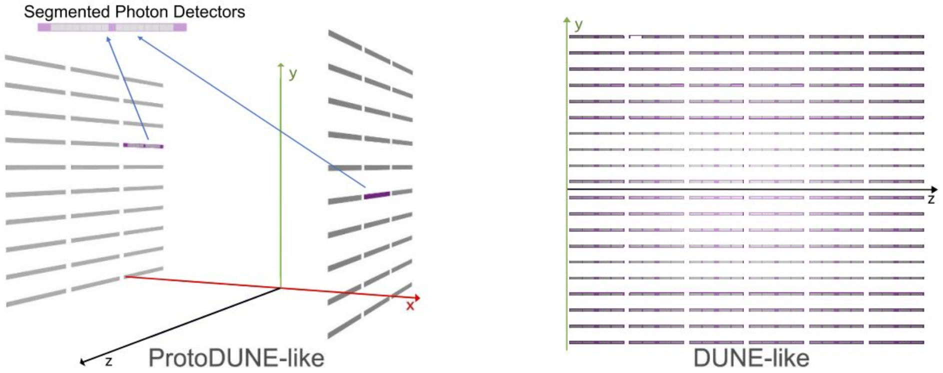
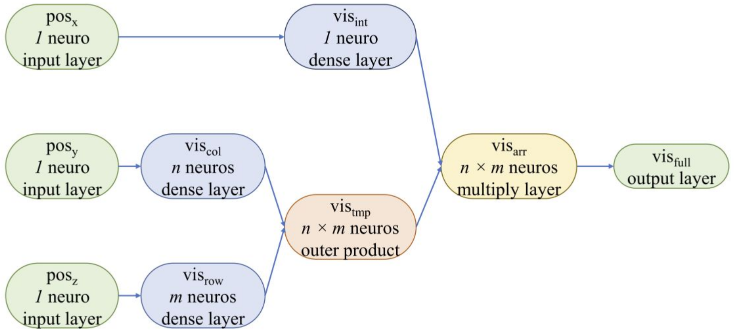


Figure 2. Photon detection systems in ProtoDUNE-like (left) and DUNE-like geometries (right).

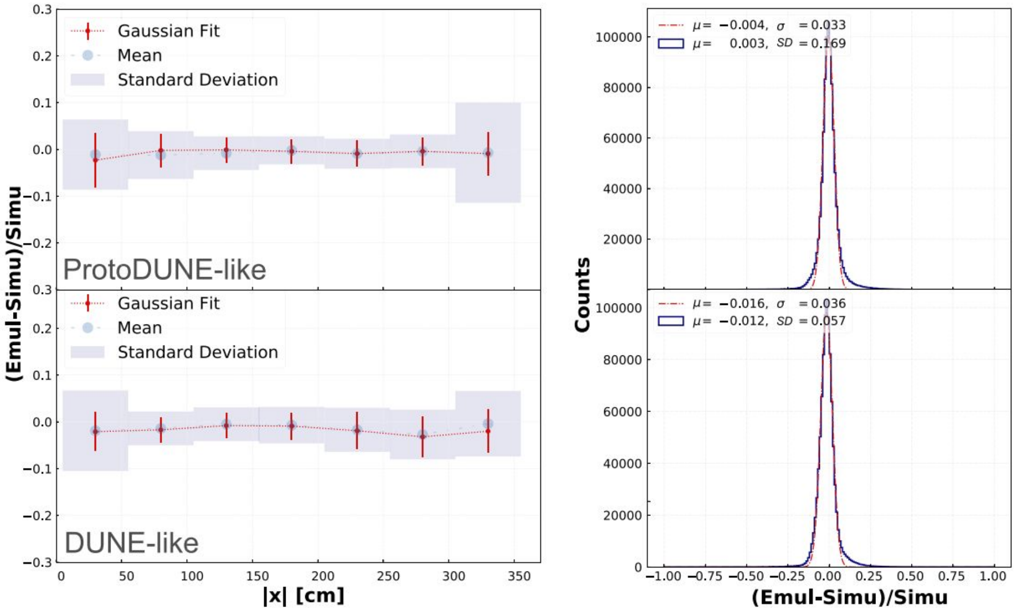


- Trained on point-source full photon transport simulations distributed throughout the detector

[Wei Mu et al 2022 Mach. Learn.: Sci. Technol. 3 015033]

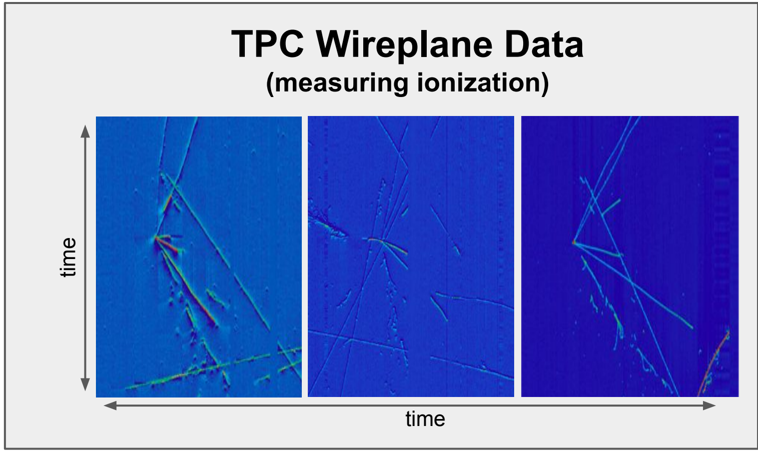
Neural Network Estimate

- Good reproduction of predicted simulation
- Simple Neural network operations produces large speed-up compared to geant4: 20-50x

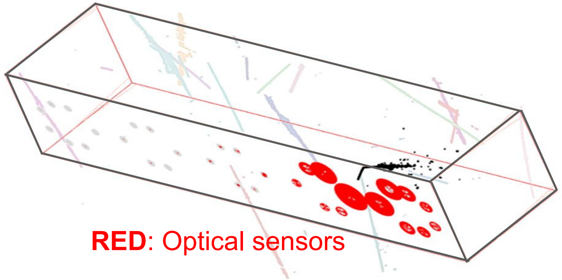


Outline

- Approach to improve **optical data** modeling using machine learning
- Early work to use generative models to reproduce **LArTPC wire plane images**

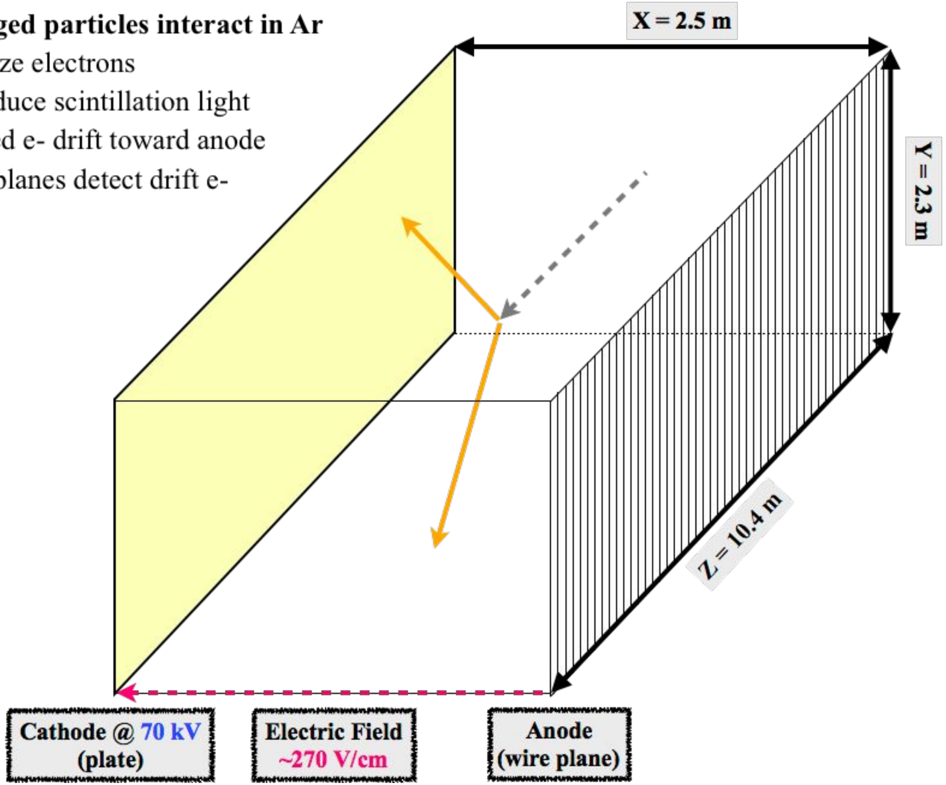


Optical Data
(measuring Scintillation photons)



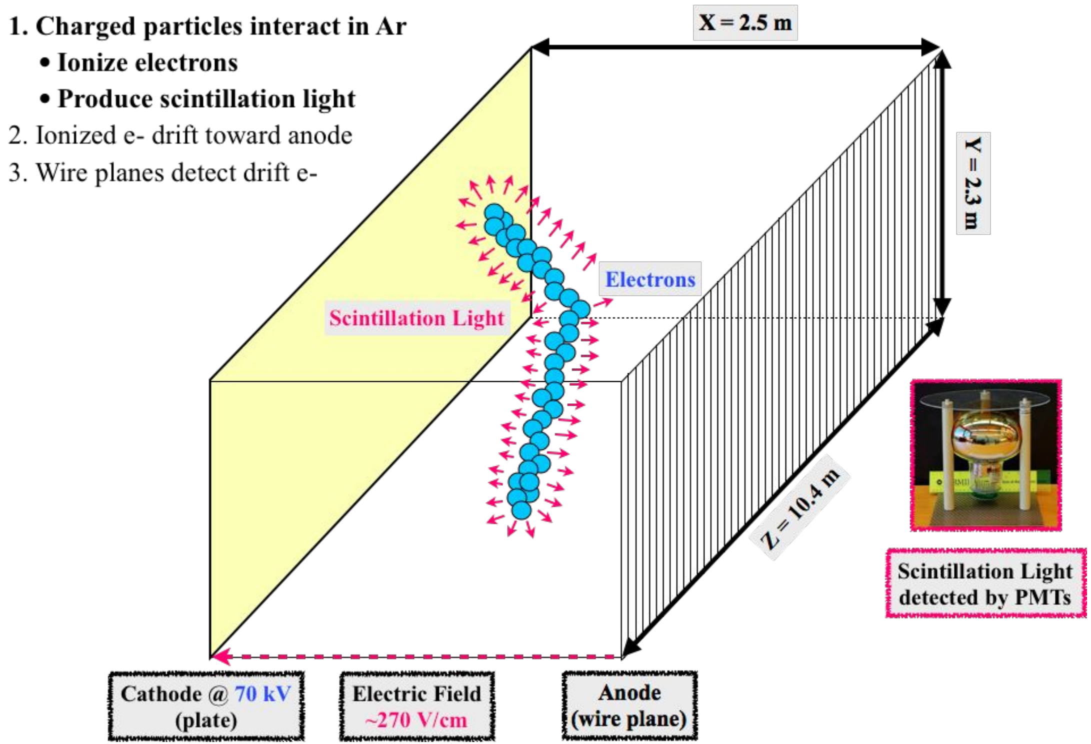
Capturing Neutrino Interaction Images w/ LArTPCs

1. **Charged particles interact in Ar**
   - Ionize electrons
   - Produce scintillation light
2. Ionized e- drift toward anode
3. Wire planes detect drift e-

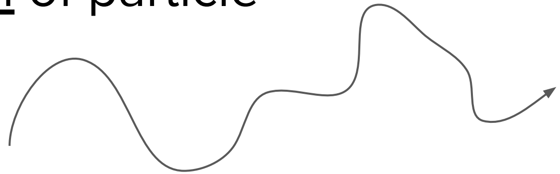


MicroBooNE TPC

Capturing Neutrino Interaction Images w/ LArTPCs



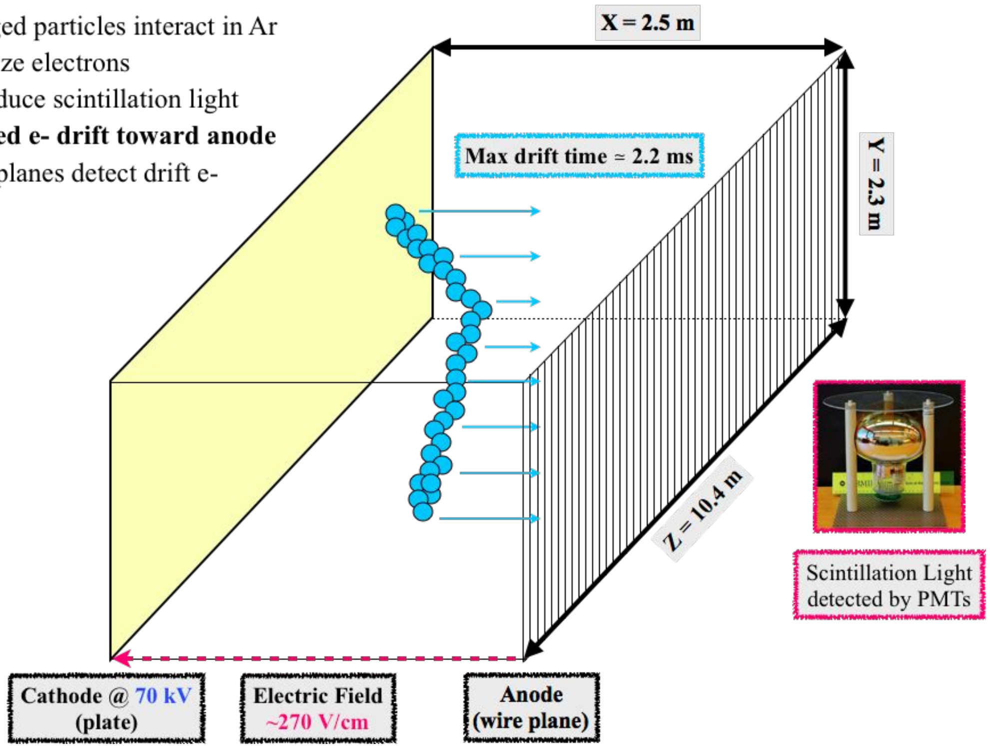
Charged particles produce ionization electrons: tell us path of particle



Also produces *light*: tells us time of the event

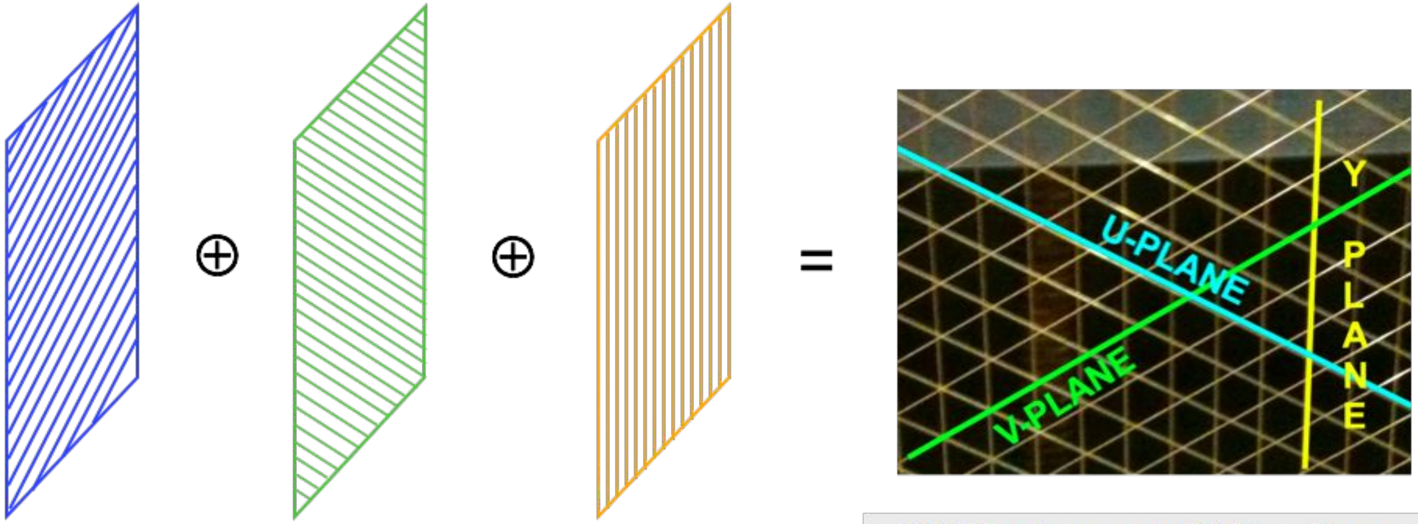
Capturing Neutrino Interaction Images w/ LArTPCs

1. Charged particles interact in Ar
   - Ionize electrons
   - Produce scintillation light
2. **Ionized e- drift toward anode**
3. Wire planes detect drift e-



Ionization electrons drift towards wireplanes

Capturing Neutrino Interaction Images w/ LArTPCs



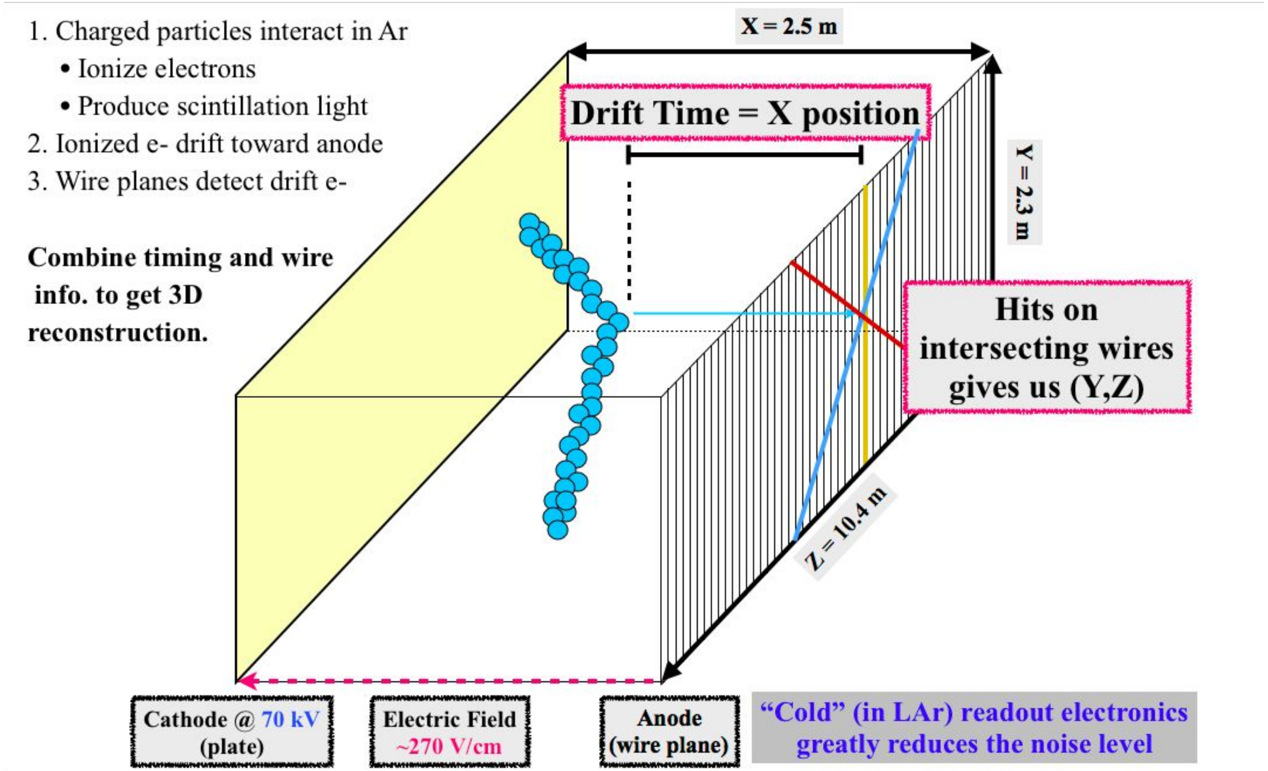
U plane (induction)

V plane (induction)

Y plane (collection)

8256 wires w/ pitch = 3mm
(Y, Z) = coincidence on wire

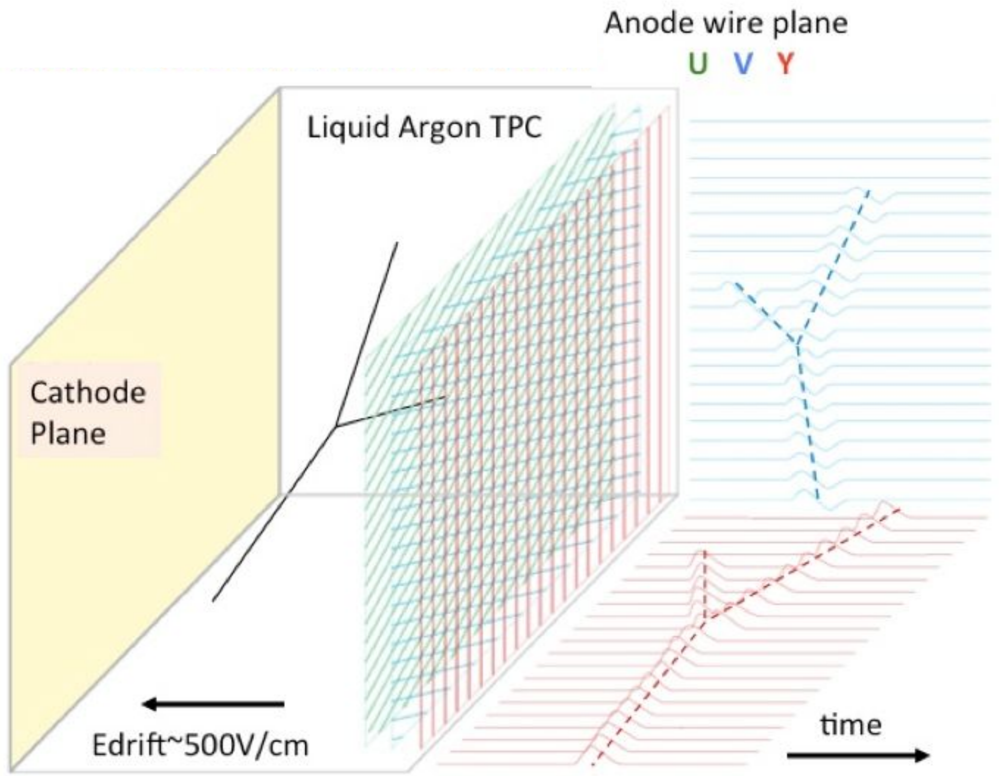
Capturing Neutrino Interaction Images w/ LArTPCs



Regions of ionization past induction wires and collect on last collection wires.

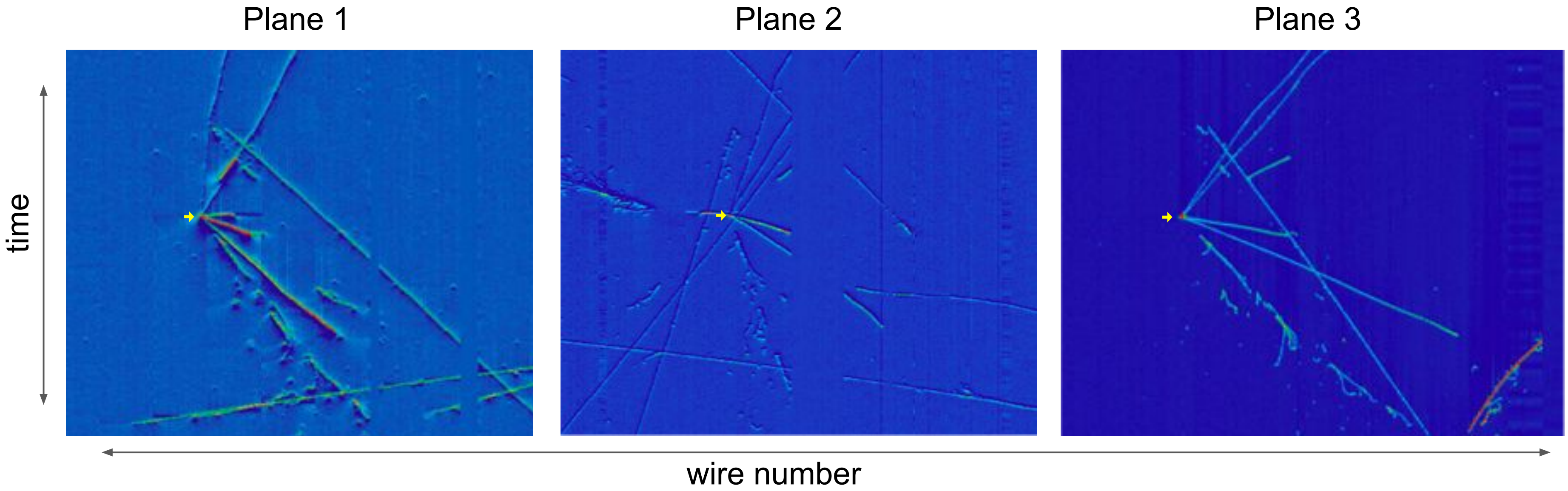
Induces current on wires intersecting at (y,z)

Capturing Neutrino Interaction Images w/ LArTPCs



Recording wire signals over time, detector produces image-like data

Capturing Neutrino Interaction Images w/ LArTPCs

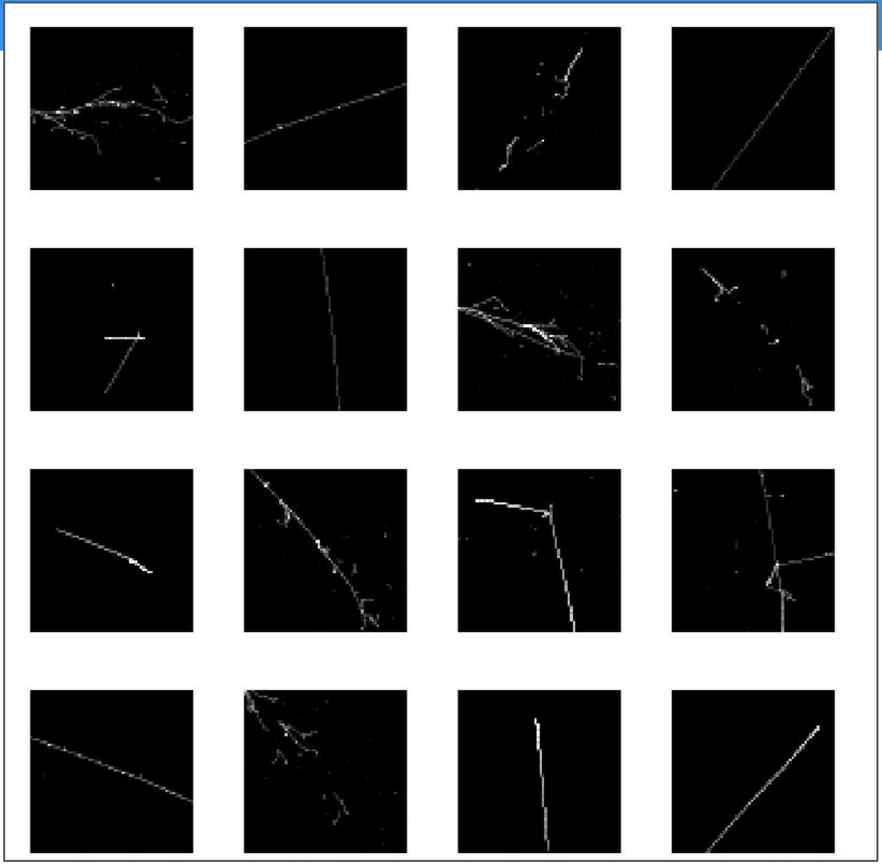


Example of data event in MicroBooNE. View of same event for each projection.

Color scale indicates amount of ionization electrons seen on wire at given time

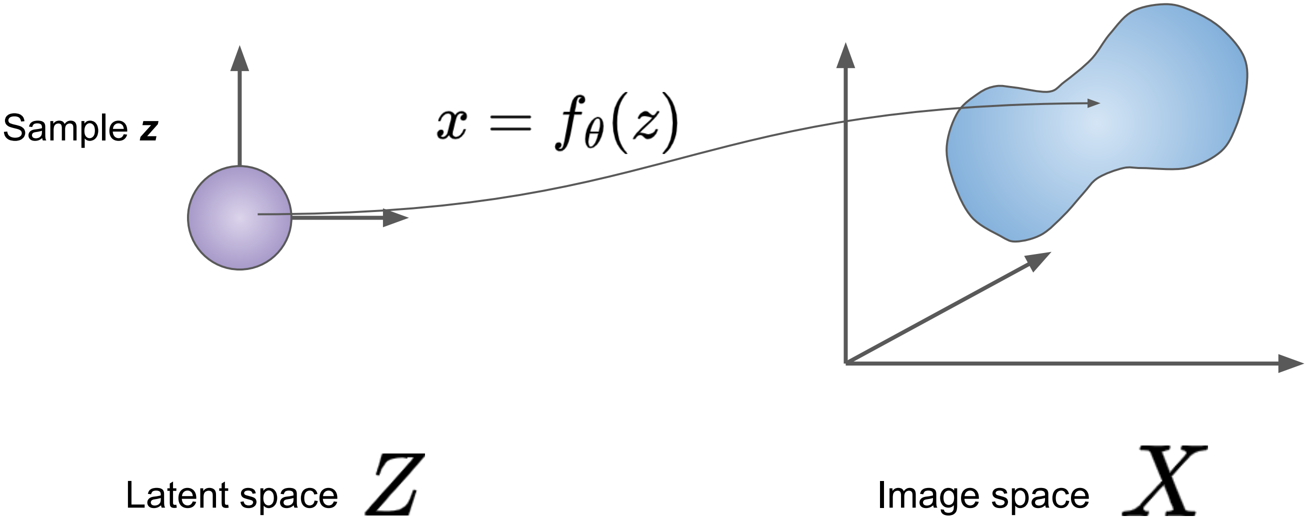
LArTPC data

- Patches of LArTPC images
- Globally Sparse: most pixels are empty
  - Non-zero pixels ~3-5%
- Locally Dense: patches have specific, thin shapes
  - Tracks: lines
  - Showers: branching lines
  - The pixel values along the lines contain info on particle type



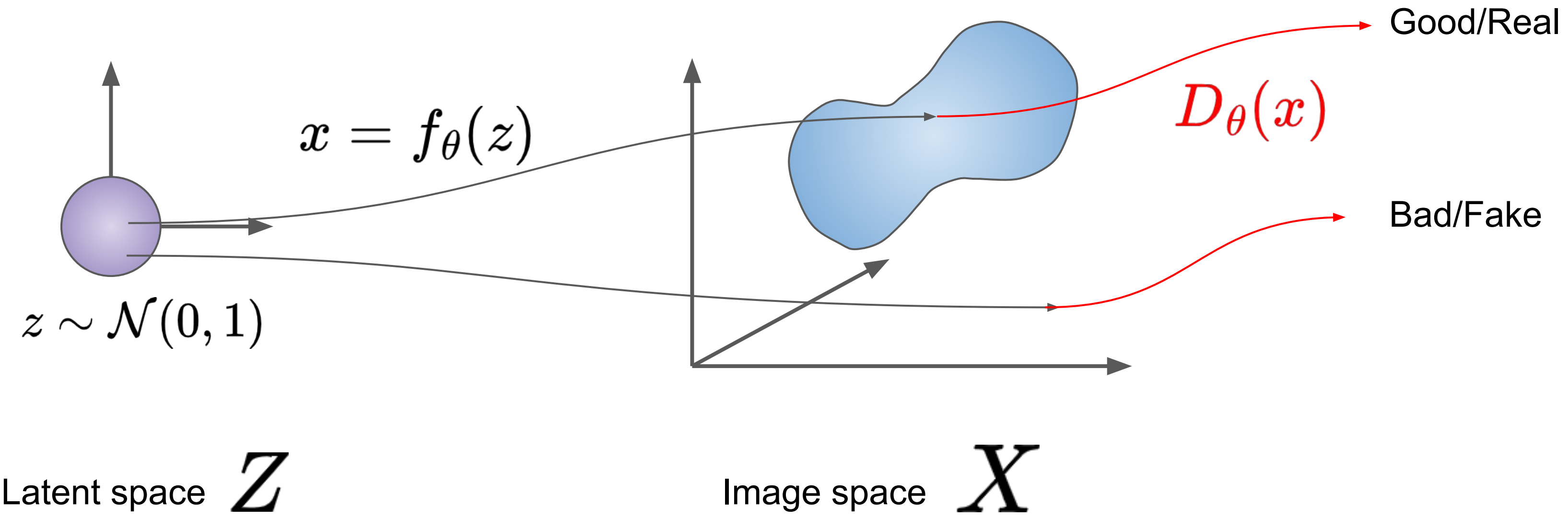
Generative Model

- Define latent vector, z, whose distribution is one we know how to sample, e.g. $z \sim \mathcal{N}(0,1)$
- Find a map from z to x, $f_\theta(z)$, so that $p(x) = p(f_\theta(z))$



How to Determine *f*

- Use a neural net for $f_\theta(z)$ and train on data to map into data region
- Need something to tell you if good or bad spot: hard since region where images located not defined
- Learn a function to tell you if good or bad: Discriminator

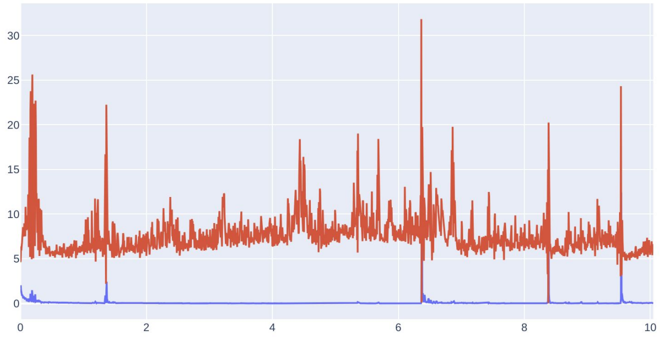


DCGANs

Not gotten very good results with GANS

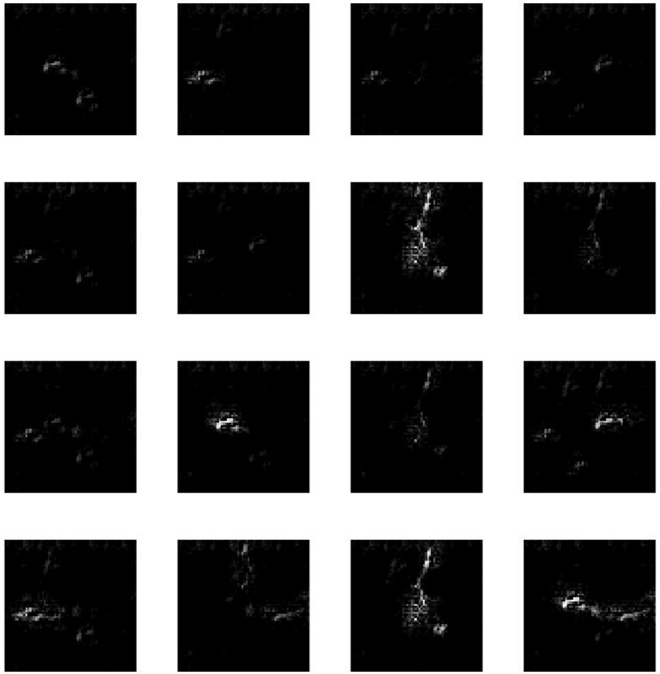
(Curious to know about any projects with GANs using IF sample – don't hesitate to let me know your experience: good or bad!)

Typical result training collapses: discriminator, too good, too early. Generator produces unconving images



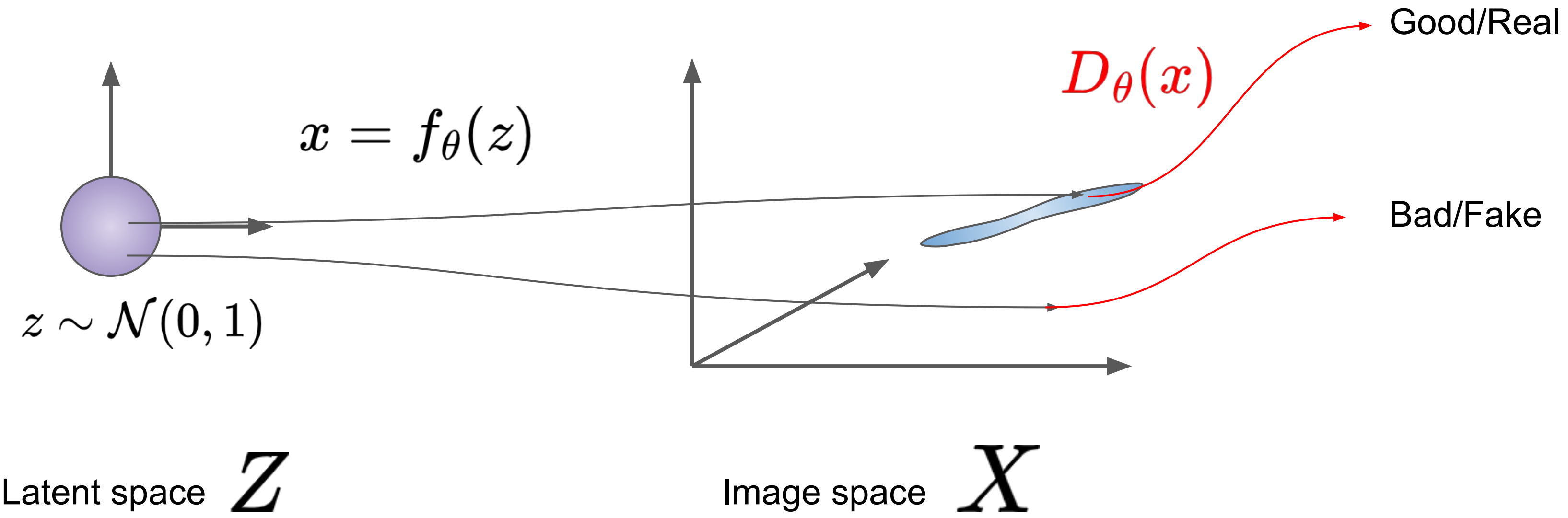
[DCGAN Tutorial]

Generated Samples using DCGAN[arxiv:1511.06434]



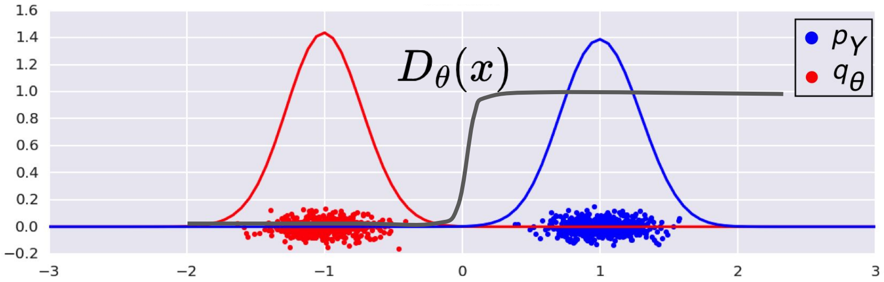
For LArTPC Images

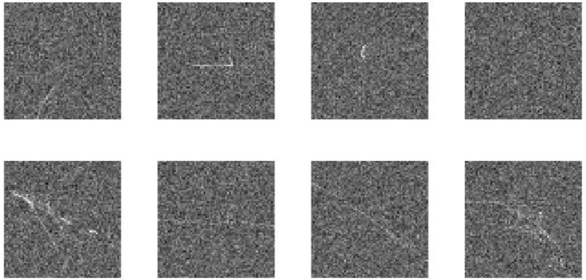
- Is sparse nature of LArTPCs an issue?
- Smaller space to find?

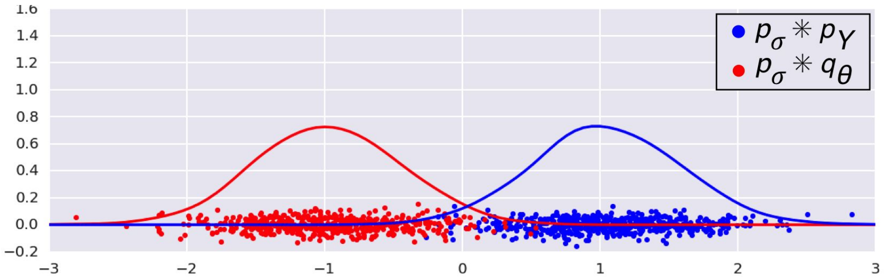


Adding Noise

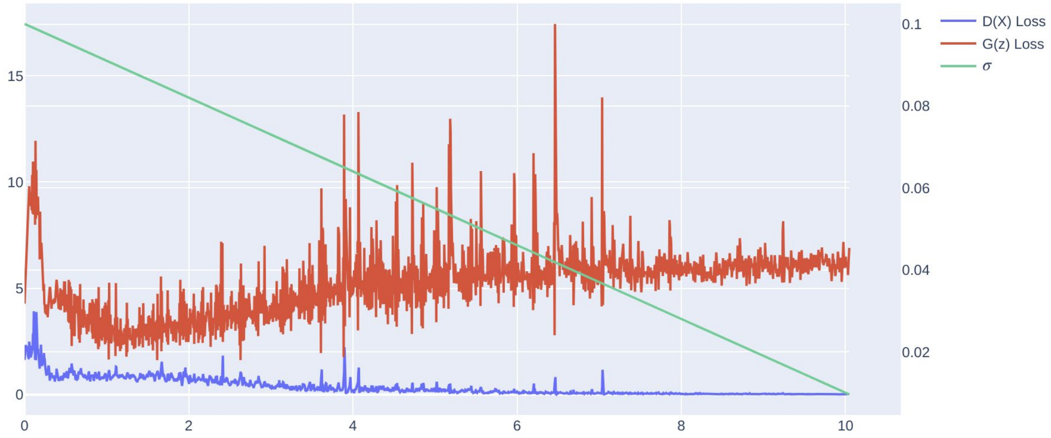
- Training collapses when generated and real images very different
- Discriminator becomes too good
- Add noise to real images, so initial overlap with generated sample

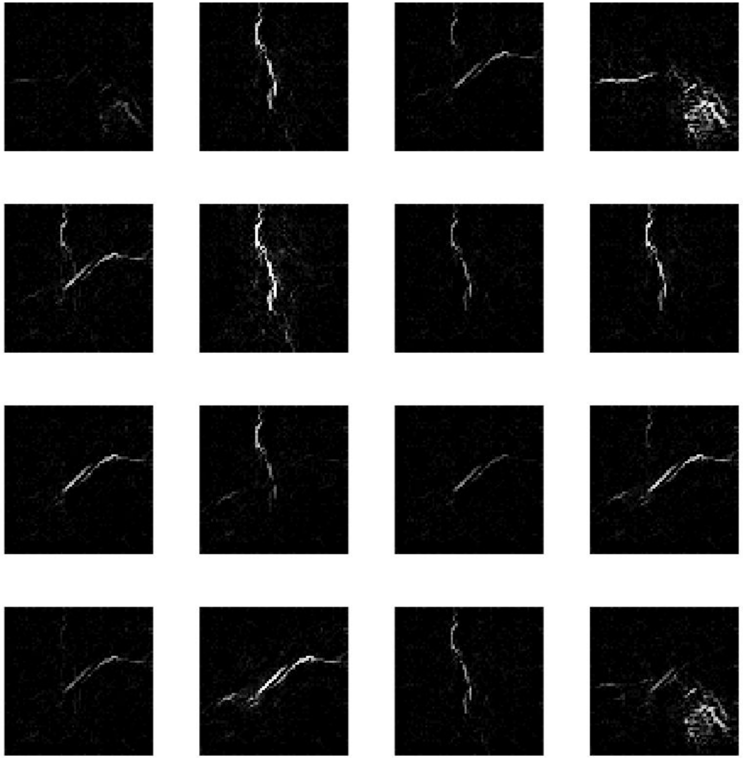






Adding noise helps a little – but not very good





Pixel-CNN

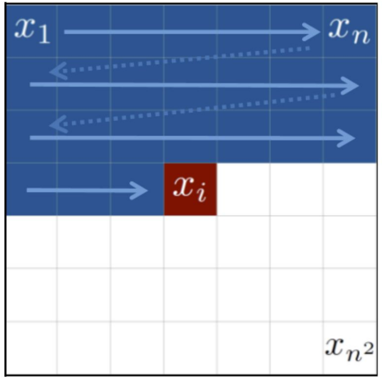
Auto-regressive approach train to predict sequence in smaller space

$$p(\mathbf{x}) = p(x_1, x_2, ..., x_{n^2})$$

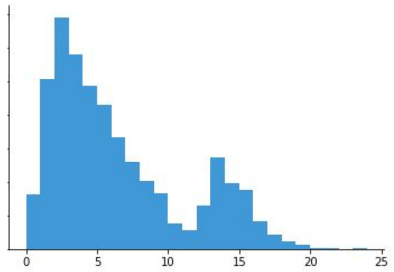
Bayes Theorem:

$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, ..., x_{i-1})$$

A sequential model!



$$p(i|x_1, x_2, \ldots, x_{i-1})$$



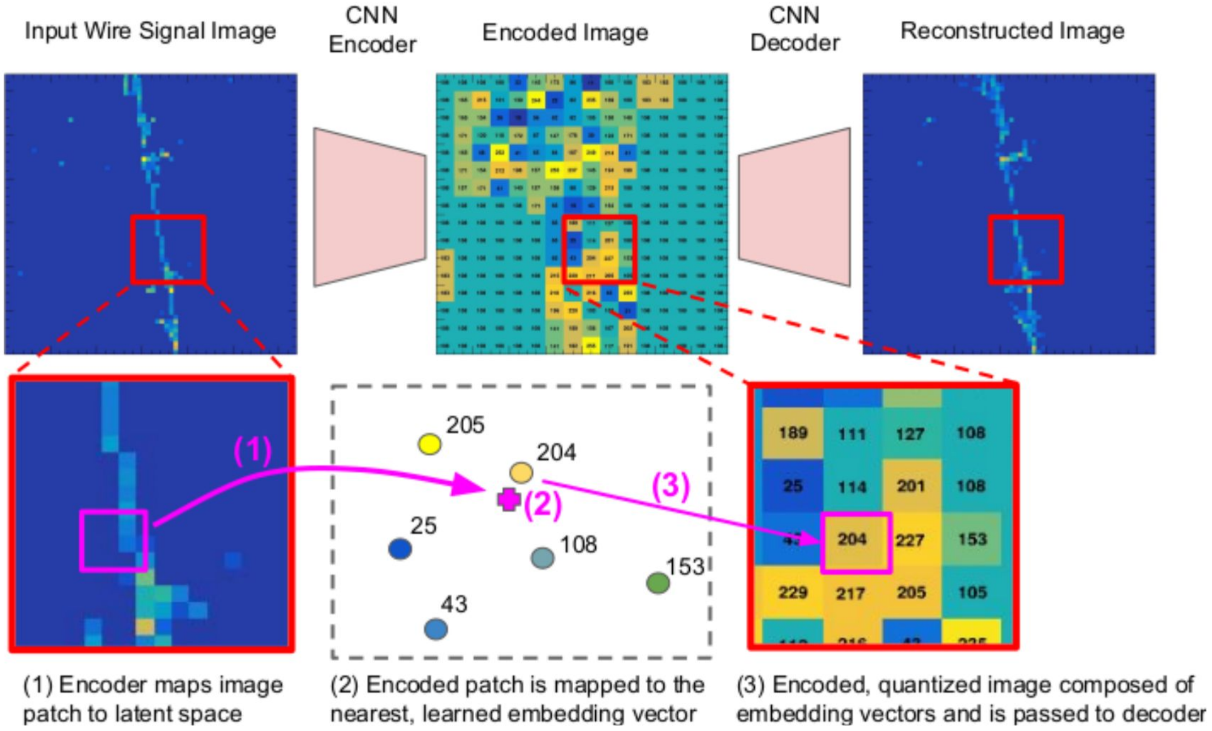
code index

Codes?

Use a CNN (with masking) to learn these distributions

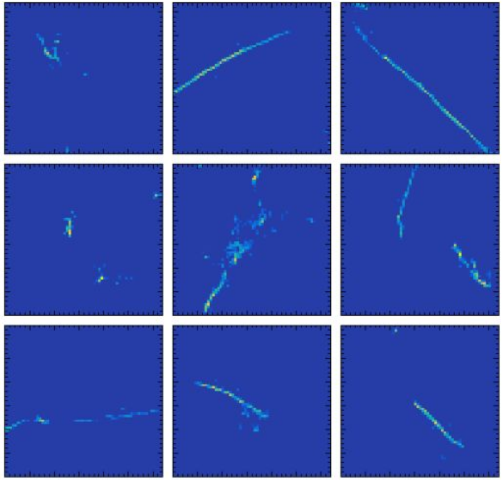
VQ-VAE for LArTPCs

- Use Vector-Quantized VAE to describe image as sequence of codes
- Patches in image mapped to 1 of 256 embedding vectors
- Embedding vectors learned during Autoencoder training
- Decoder learned as part of AE training
- Does not require that we train a discriminator

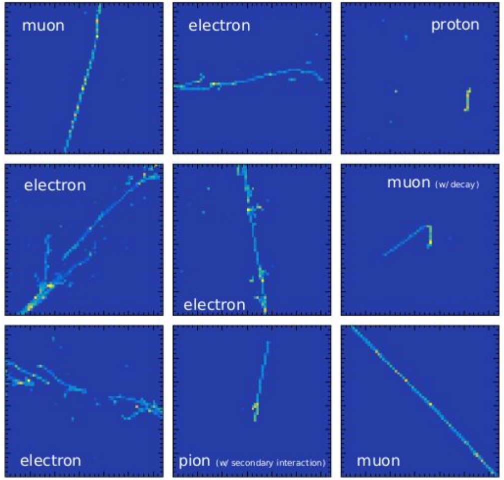


Pixel-CNN Results

- Image quality better than GAN
- Seeing wider variety of patterns
- Not seeing as many long lines or showers
- More quantitative measures latter



(a) Generated images

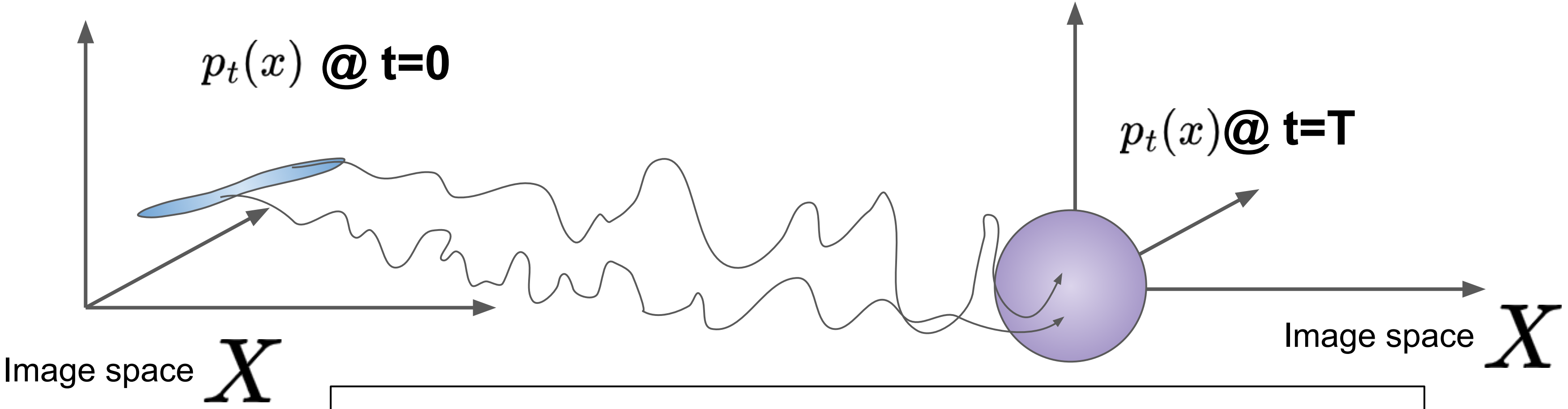


(b) Training images

arXiv:2204.02496

Diffusion Models

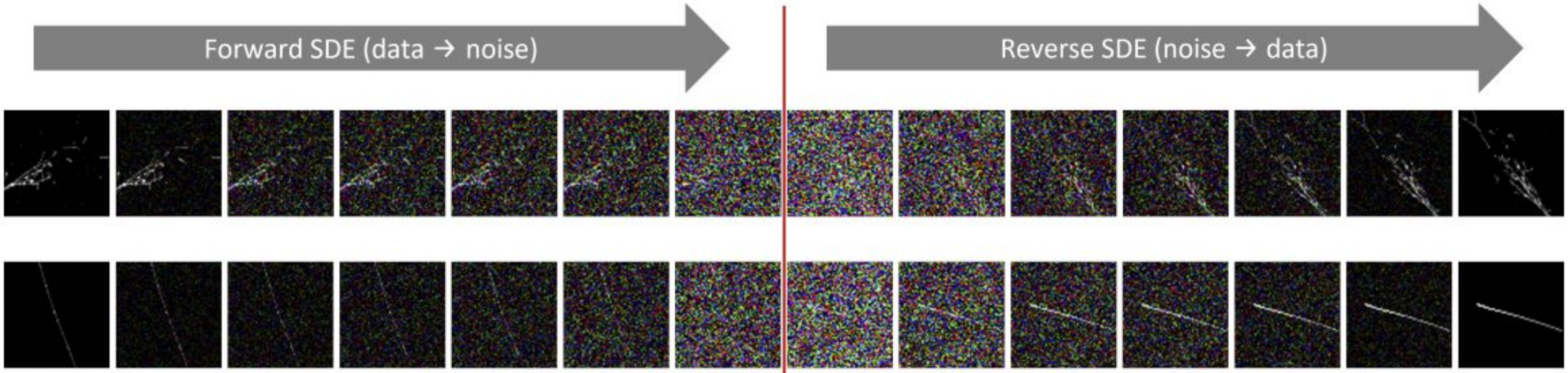
- Change our generative picture a little
- Choose our latent space to be same size as ***X***
- Associate map between ***X*** and ***Z*** as some process that runs for time, t=0 to t=T so that $p_T(x) = \mathcal{N}(0, I)$



For LArTPCs, instead of trying to find (very thin) manifold in ***X*** (as we did with GAN), move samples into Normal Distribution

SDEs

- The process can be modeled as a stochastic differential equation (SDE)
- Diffusion process defined by drift and brownian motion
- Can define a reverse process back to data. Requires estimating the <u>score function</u>: related to gradient of $p(x)$. In practice, this is a neural net.



$$\mathrm{d}\vec{\boldsymbol{X}}_t = \mathbf{f}(\vec{\boldsymbol{X}}_t, t)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{W}_t$$

score function

$$\mathrm{d}\overleftarrow{\boldsymbol{X}}_s = [-\mathbf{f}(\overleftarrow{\boldsymbol{X}}_s, s) + g^2(s)\boxed{\nabla \log p(\overleftarrow{\boldsymbol{X}}_s)}]\mathrm{d}s + g(s)\mathrm{d}\boldsymbol{W}_s$$

SDE

- Diffusion Process SDE
- Choice: Variance Preserving - SDE

[Y. Song et al. arXiv:2011.13456 (2020)]

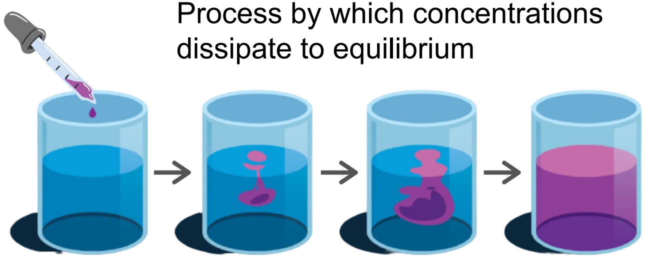
Change in pixel values at t

$$\mathrm{d}\vec{\boldsymbol{X}}_t = -\frac{1}{2}\beta(t)\vec{\boldsymbol{X}}_t\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\vec{\boldsymbol{W}}_t.$$

Drift

Brownian Motion
Provides change in random directions

Process by which concentrations dissipate to equilibrium



Note: For us, equilibrium not uniform, but the Normal distribution

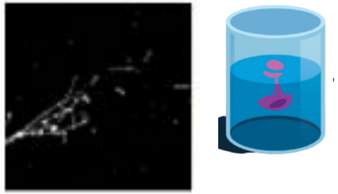
SDE: Forward Process

- The distribution after some time, ***t**,* using SDE

$$p(\vec{\boldsymbol{X}}_t|\vec{\boldsymbol{X}}_0) = \mathcal{N}(\vec{\boldsymbol{X}}_0 e^{-\frac{1}{2}\int_0^t \beta(s)ds}, \boldsymbol{I} - \boldsymbol{I}e^{-\int_0^t \beta(s)ds})$$

Starting with original image at **t=0**, $\vec{X}_0$, gives the distribution of pixels at later time, ***t*.**

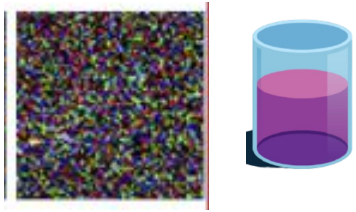
$\vec{X}_0$



At time ***t=T***, configure parameters such that

$$\vec{X}_T \sim \mathcal{N}(0, I)$$

$\vec{X}_T$



SDE: Reverse Process

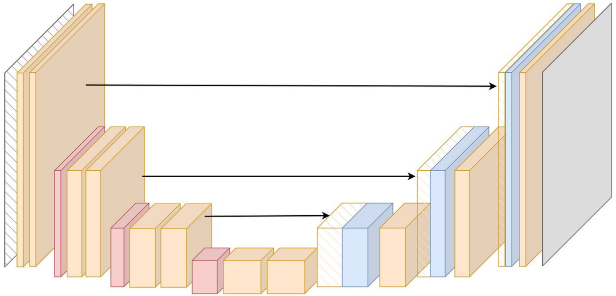
Reverse process SDE, with time parameter ***s goes from 0 to T*** and ***s=T-t***, is

score function

$$d\overleftarrow{\boldsymbol{X}}_s = [-\mathbf{f}(\overleftarrow{\boldsymbol{X}}_s, s) + g^2(s)\boxed{\nabla \log p(\overleftarrow{\boldsymbol{X}}_s)}]ds + g(s)d\boldsymbol{W}_s$$

Estimate with neural net: $\boldsymbol{s}_\theta(\vec{X}_t, t)$

Output is change for every pixel value: **U-Net Architecture**



B. Anderson. *Stoch. Proc. and their App.*, 12 (3):313–326 (1982)

SDE: Reverse Process

Empirical loss follows from a previous slide's: $p(\vec{\boldsymbol{X}}_t|\vec{\boldsymbol{X}}_0) = \mathcal{N}(\vec{\boldsymbol{X}}_0 e^{-\frac{1}{2}\int_0^t \beta(s)ds}, \boldsymbol{I} - \boldsymbol{I}e^{-\int_0^t \beta(s)ds})$

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_i^N ||\boldsymbol{s}_{\boldsymbol{\theta}}(\vec{\boldsymbol{X}}^i_{t^i}, t^i) - \frac{-(\vec{\boldsymbol{X}}^i_{t^i} - \vec{\boldsymbol{\mu}}^i_{t^i})}{\vec{\boldsymbol{\sigma}}^{\boldsymbol{2}\,i}_{t^i}}||_2^2$$

Where we can sample a batch with *N* images $\vec{X}_t$

Each image is associated with a different time, *t*, sampled uniformly from (0,T]

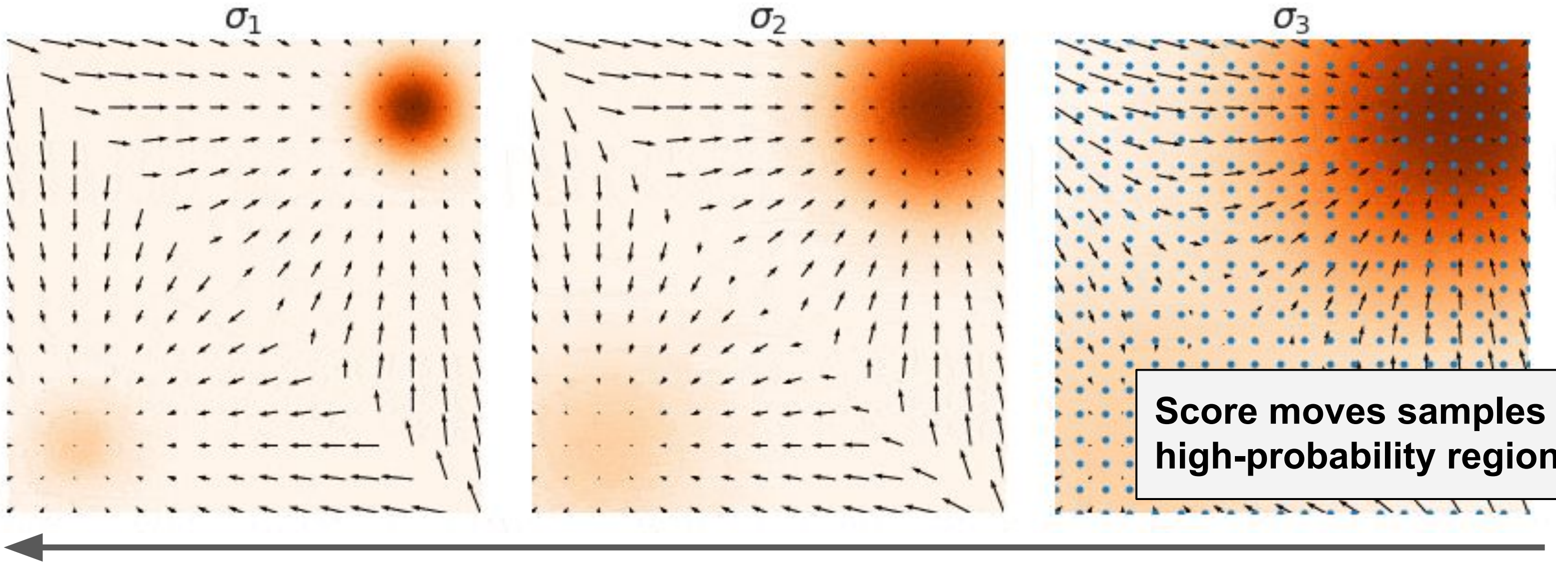
After training ***s***, we can solve Reverse SDE to sample $\vec{X}_T$

B. Anderson. *Stoch. Proc. and their App.*, 12 (3):313–326 (1982)

Sampling

Recall: variance changes with time. Largest at time of sampling, near zero when at end of reverse process. Thus we are using Annealed Langevin Dynamics.

Large variance at beginning helps produce images from different modes



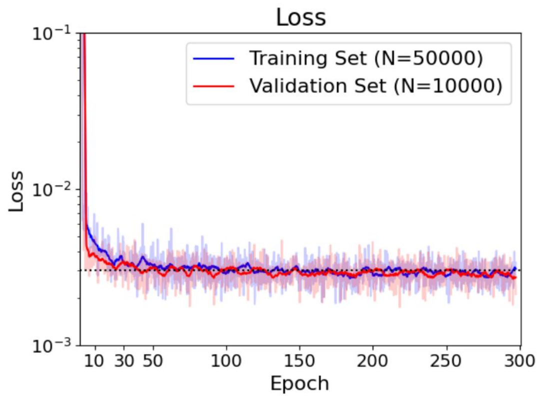
Score moves samples towards high-probability regions

Sampling (reverse process) time direction

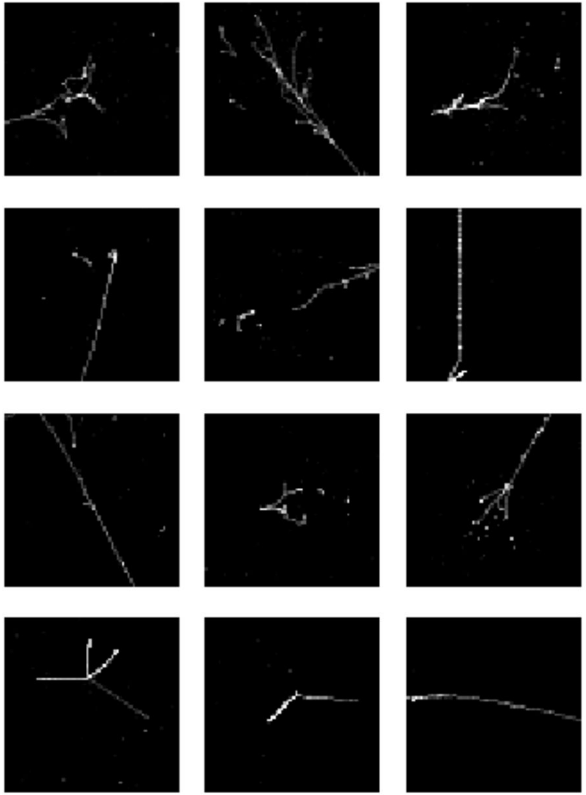
Score-matching Diffusion Model for LArTPCs

Trained a VPSDE using the Y. Song repo

Dataset consisted of 64x64 pixel crops of images from the PiLArNet open dataset
Note: not full particle trajectories.



Plateau reached by 50 epochs



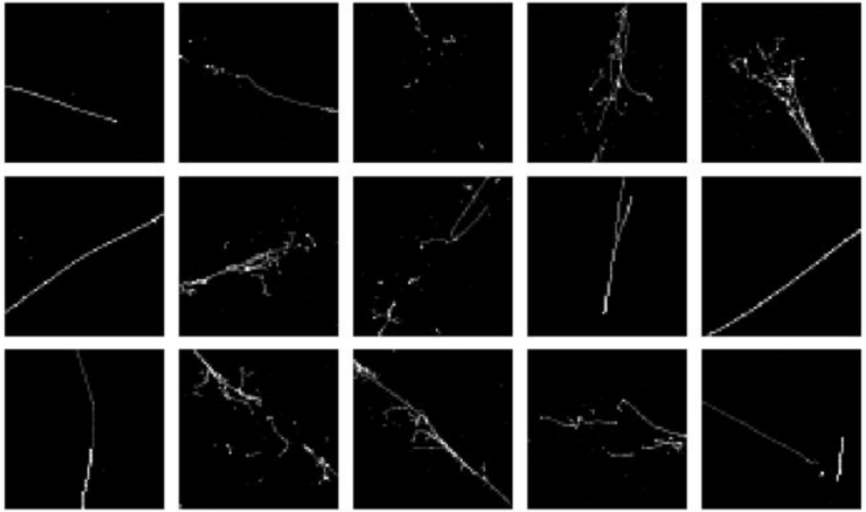
Z. Imani
Tufts Physics

S. Aeron
Tufts ECE

Supported through NSF IAIFI

Score-matching Diffusion Model for LArTPCs

Which set, left or right, is training images – which are generated?



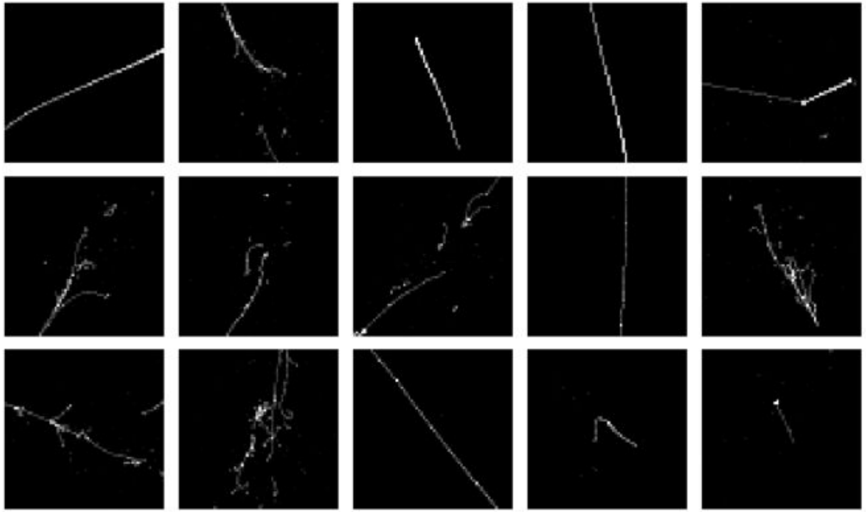
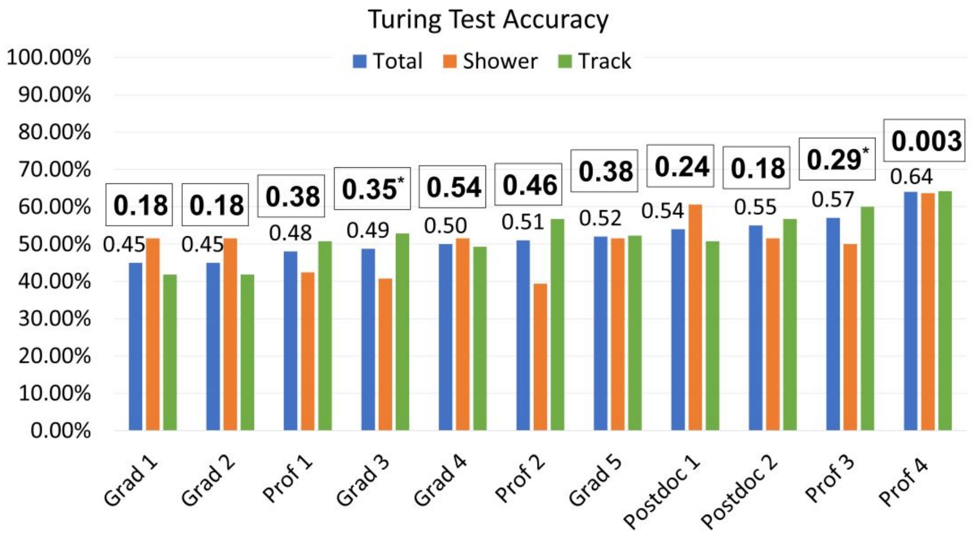


Image fidelity enough to be indistinguishable?

Real versus Fake

With exception of one person, accuracy for small sample of scanners statistically consistent with random guessing



arXiv:2307.13687 (2023)

Want to try yourself?

100 examples to judge can be found here.

(Don't forget to do the practice quiz to help train yourself.)

Evaluating Quality

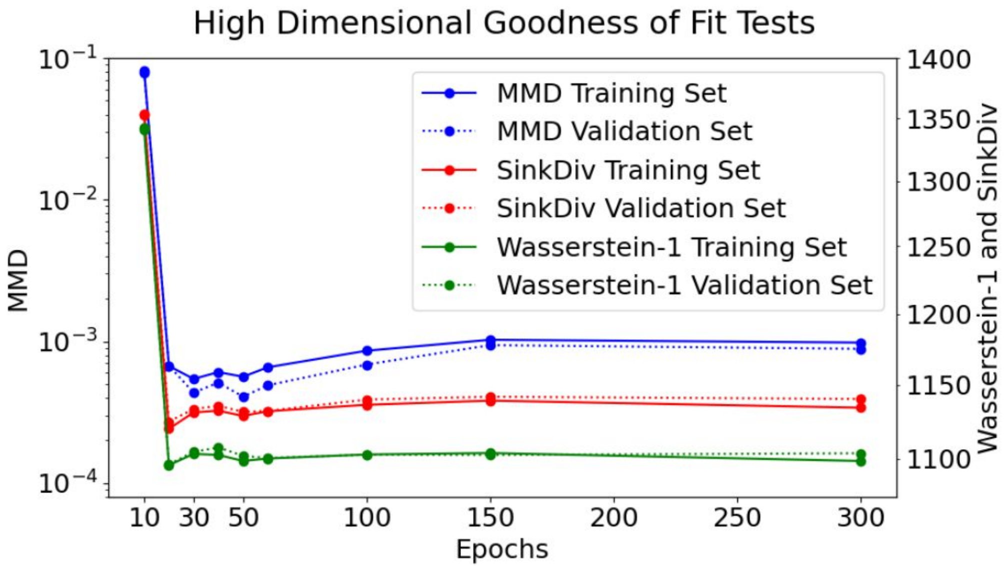
Need additional measures to quantify beyond visual inspection:

- High dimensional comparisons between images
- Comparison of other neural network outputs
- Physics-analysis motivated quantities

arXiv:2307.13687 (2023)

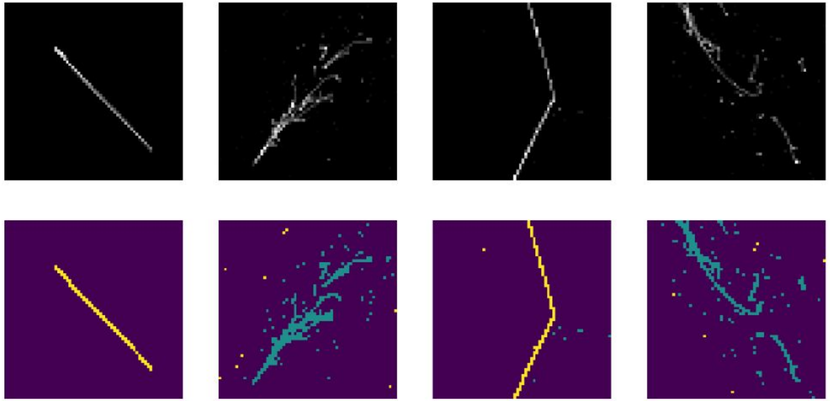
Evaluation: High-Dim. Metrics Directly on Images

- Each comparison is done between the training and generated images and the validation and generated images



Evaluation: Comparing SSNet Behavior

- Another approach: compare outputs of a neural network
- Given prominent role of track/shower pixel labeling in neutrino reconstruction, we use SSNet [Phys. Rev. D 99, 092001]

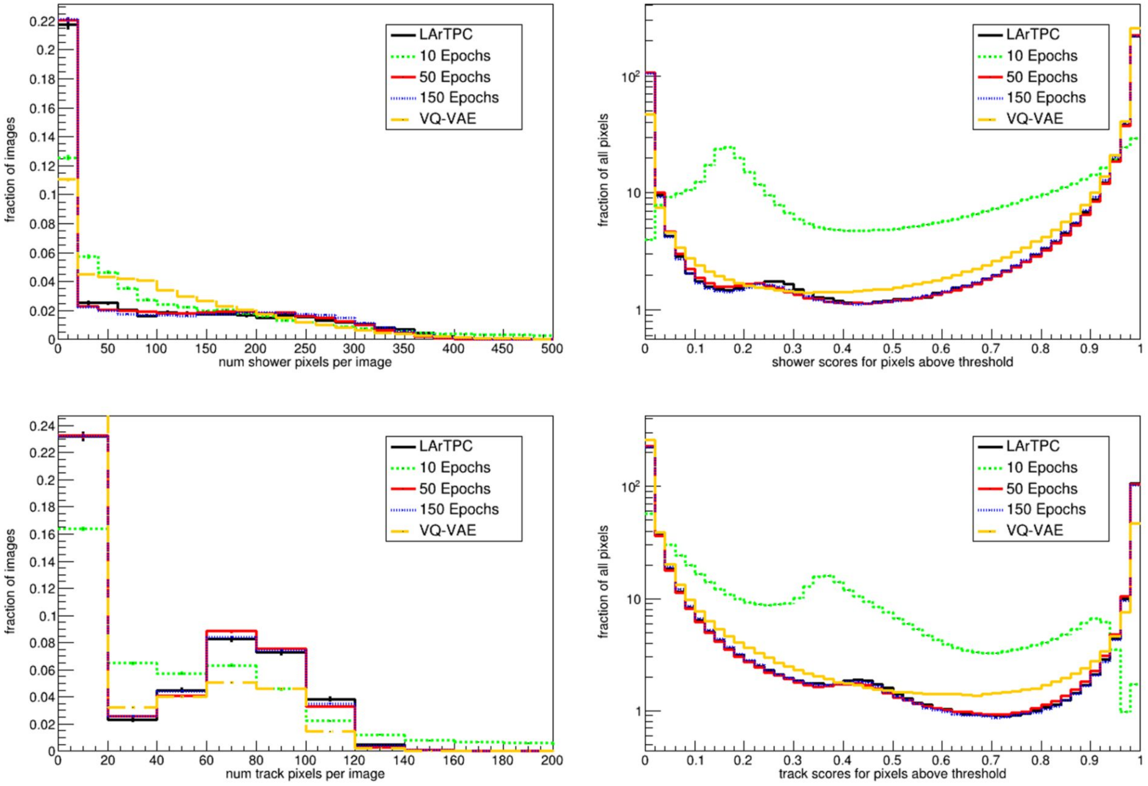


Version of SSNet trained on the training data.

SSNet Weights for those who want to compare: zenodo

Evaluation: Comparing SSNet Behavior

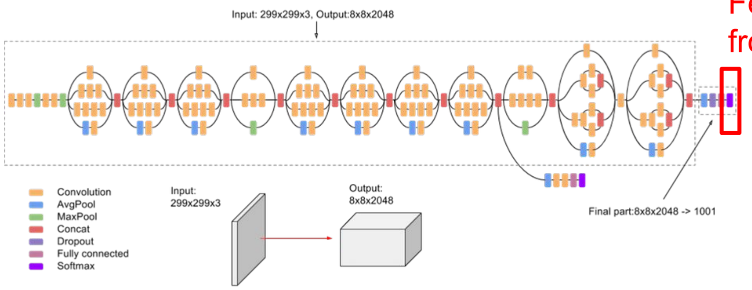
- labeling frequencies and output scores
- For SBDM, matches well after 50 epochs of training
- VQ-VAE generated images also compared – SBDM clear improvement, matching visual judgement



Evaluation: SSNet-FID

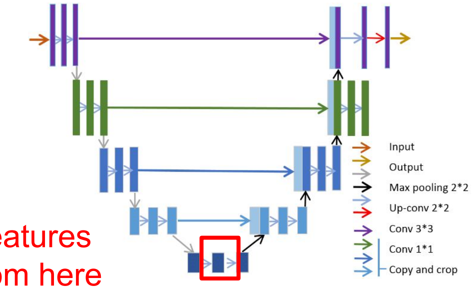
- For natural images, a standard metric is the FID or Frechet Inception Distance
- It compares the feature map of the last layer of the ImageNet-trained Inception v3 classifier
- SSNet being a U-Net, using that layer was not tractable, so we looked at layers near the bottleneck

Inception v3 layers



Features from here

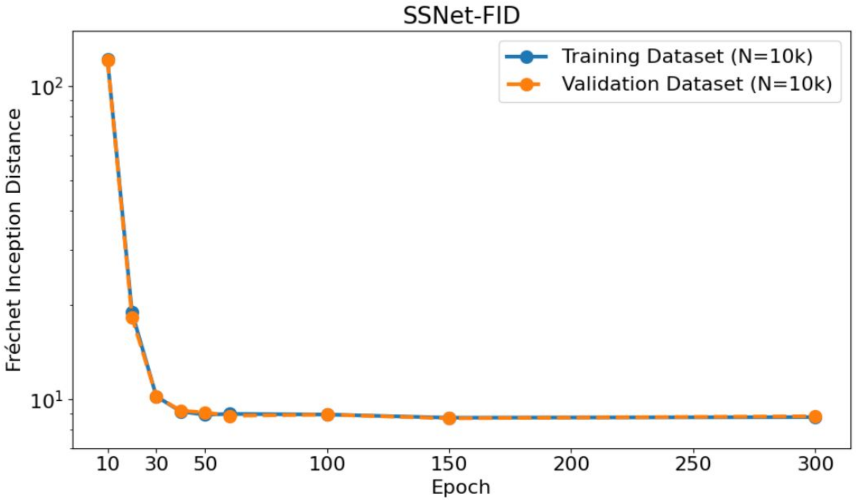
U-Net Layers



Features from here

Evaluation: SSNet-FID

- SSNet-FID comparing train and test sets each to the generated image set
- Generated images produced for several epochs
- Plateau around 50 epochs, in agreement with metrics

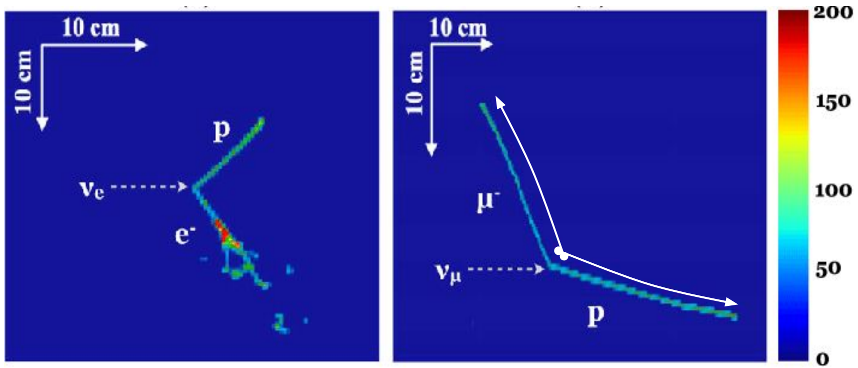


Epochs	10	20	30	40	50	60	100	150	300
Training	121.63	18.96	10.2	9.15	8.94	8.99	8.95	8.75	8.79
Validation	121.38	18.31	10.2	9.21	9.09	8.87	8.94	8.71	8.84

SSNet-$\text{FID}_{N=10K}$ for future comparisons

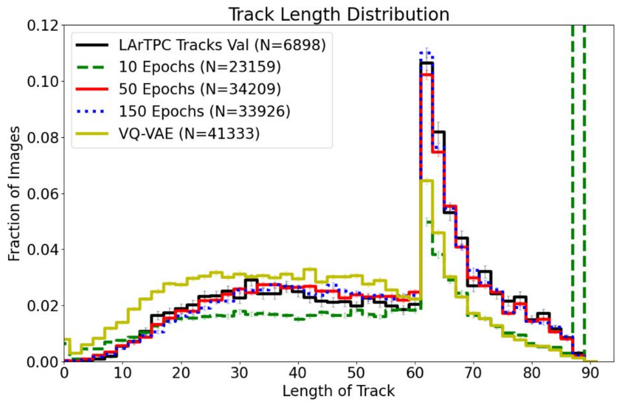
Evaluation: Reco-inspired Quantities

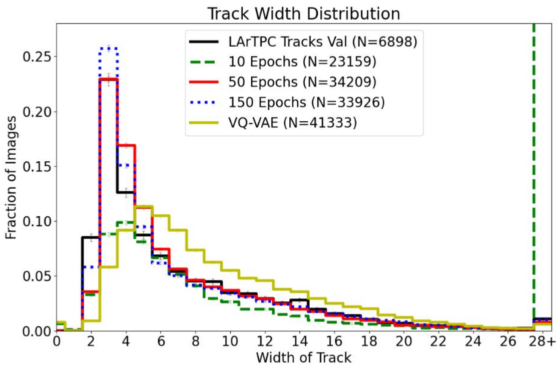
- Ultimately, threshold for “good enough” means no bias introduced into reconstruction quantities and physical observables
- Our dataset are only crops: so full reconstruction of momenta not possible yet
- Use algorithms related to energy calculation
  - Track length of track-like pixels (judged by SSNet) - proxy for range
  - Number of shower pixels in an image - proxy for calorimetry

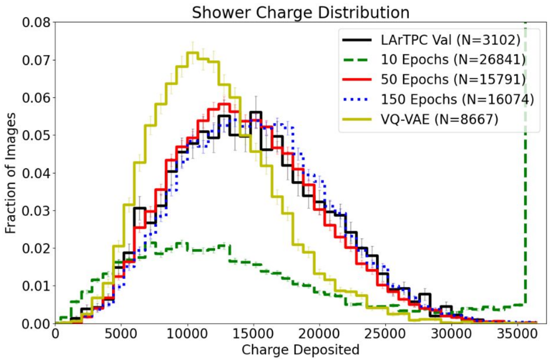


Evaluation: Reco-inspired Quantities

- Track length and width derived from 1st and 2nd PCA axes
- Similar – but statistically different - not perfect, close enough?
- Track width least similar: generated tracks somewhat too wide (Jitter? Scattering? Delta-rays?)

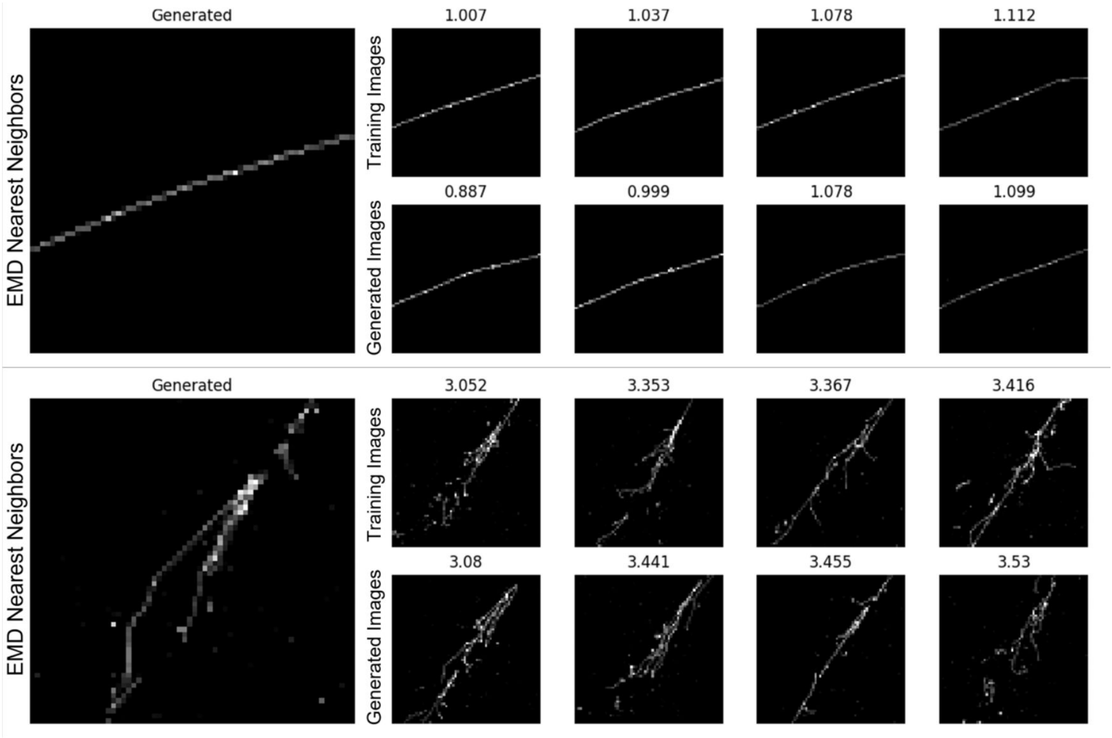






Checking for Mode Collapse

- Do not want generated images to be small variations on training set examples
- Used Earth Mover's distance (Wasserstein-1 distance) to find closest neighbors
- Match trajectory direction
- Shower patterns varied
- Tracks have variation in high-energy deposits (expected from Bethe-Block distribution)
- Similar EMD among generated and training images



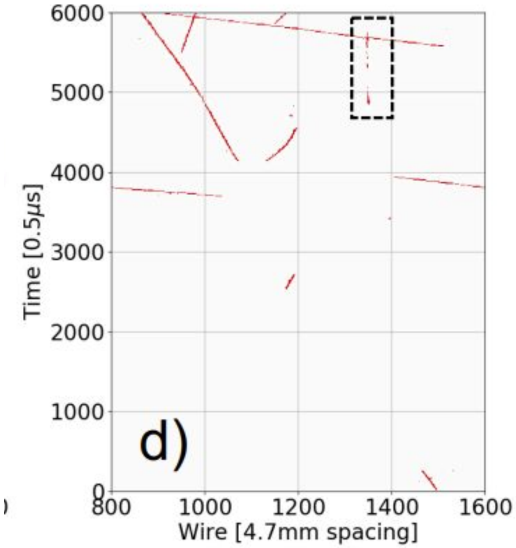
SBDM Next Steps

- Proof-of-principle that despite LArTPC sparse structure, generative models seem possible if using Diffusion Models
- Establish metrics to compare future work by others – nascent effort so lot's of easy things to try
    - E.g. How do Flow networks perform?
- All methods and image sets used will be posted on zenodo (coming soon: watch for updates to arXiv:2307.13687 (2023)
- All metrics tell a consistent story: converge around 50 epochs and plateau, matching (subjective) visual quality trends

SBDM Next Steps

- Next biggest step is generation of individual particles conditioned on momentum
- Require bigger image sizes: 128x128 or 512x512
- Neutrino interaction can be composed with individual particles

Uses: Image repair/Gap Filling

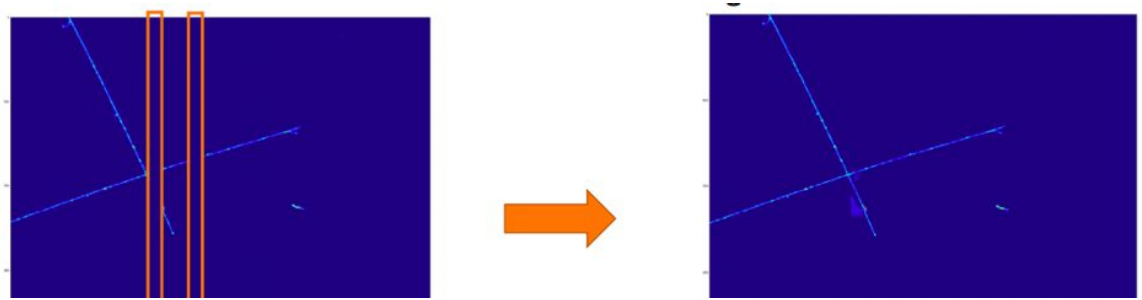


Missing sensor-signals either from induction dynamics or non-responsive wires is a major cause of downstream reconstruction errors

Limited use (e.g. no generation of particles entirely in a gap) can help a lot to:

- Reduce broken tracks
- Reduce False 3D spacepoints
- Connect tracks in uninstrumented regions between detector modules

Past efforts identity right regions, but do not produce realistic looking track segments

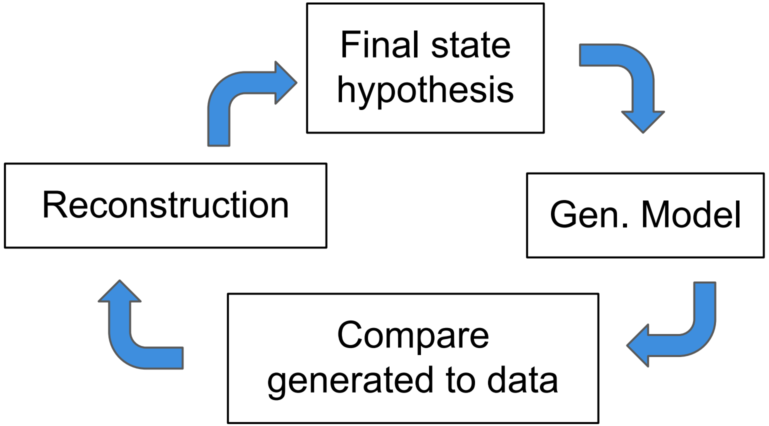


Uses: Cosmic Background Augmentation

- In order to estimate backgrounds and mis-reconstruction rates due to cosmic background, surface LArTPCs use data taken when beam is off
- Simulated neutrino are overlaid into image to make simulation samples
- Not nearly enough samples can be recorded due to storage
- Can we generate large enough background crop (512x512 or 1024x1024)?
- ***Must generate 3-plane consistent image****: potential challenge*

Uses: Iterative final state comparisons (speculative)

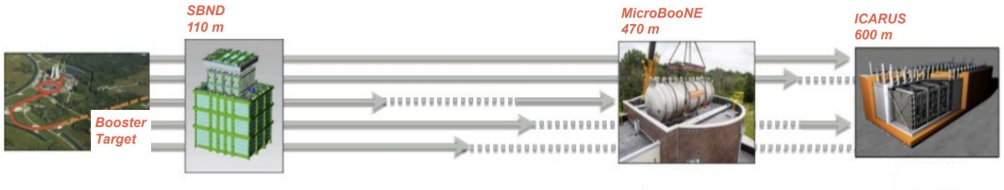
- Reconstruction makes one-pass and extracts final state particles from image of neutrino interaction
- Use generative models to generate expected image of propose final states and use comparison to inform errors and ***action***:
  - E.g. adjust reco vertex, recluster particle, retrieve portion of raw image for review, instigate image repair
  - Moving from the ML into **AI** for Neutrino Physics
- Near-term useful comparisons possible
  - Compare photon-induced shower (BG) versus small proton+electron shower (signal) using matching images
  - Probing classifier robustness by modifications of particles in images



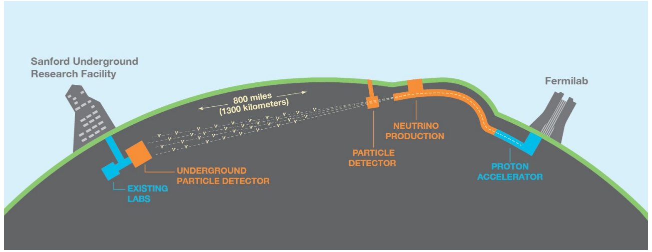
Other Data-Driven Possibilities

- Elements of Neutrino Interaction Generators do not have good theoretical models yet, e.g. hadronic output
- Neutrino oscillations have near and far detector data to constrain models of flux and cross section and to extrapolate far detector neutrino flux
  - Remove reliance of cross section or flux models through data representations learned by training generative models

SBN Program



DUNE



Conclusions

- Early days of generative modeling for Neutrino Physics
- Despite sparse LArTPC images, at least diffusion models produce quality images both visually and in the reproduction of various metrics
- Opens doors to approach several challenges in new ways

Backups

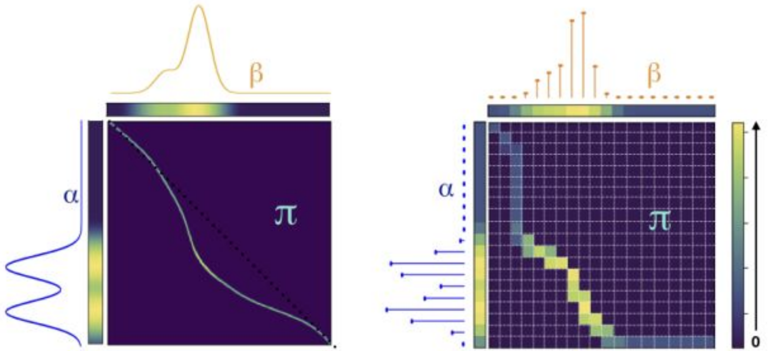
Definitions

The Wasserstein Distance

Let $\alpha \in \mathcal{M}_+^1(\mathcal{X})$ and $\beta \in \mathcal{M}_+^1(\mathcal{Y})$,

$$W_c(\alpha, \beta) = \min_{\pi \in \Pi(\alpha,\beta)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y)\mathrm{d}\pi(x,y) \qquad (\mathcal{P})$$

For $c(x,y) = \|x-y\|_2^p$, $W_c(\alpha,\beta)^{1/p}$ is the Wasserstein distance.



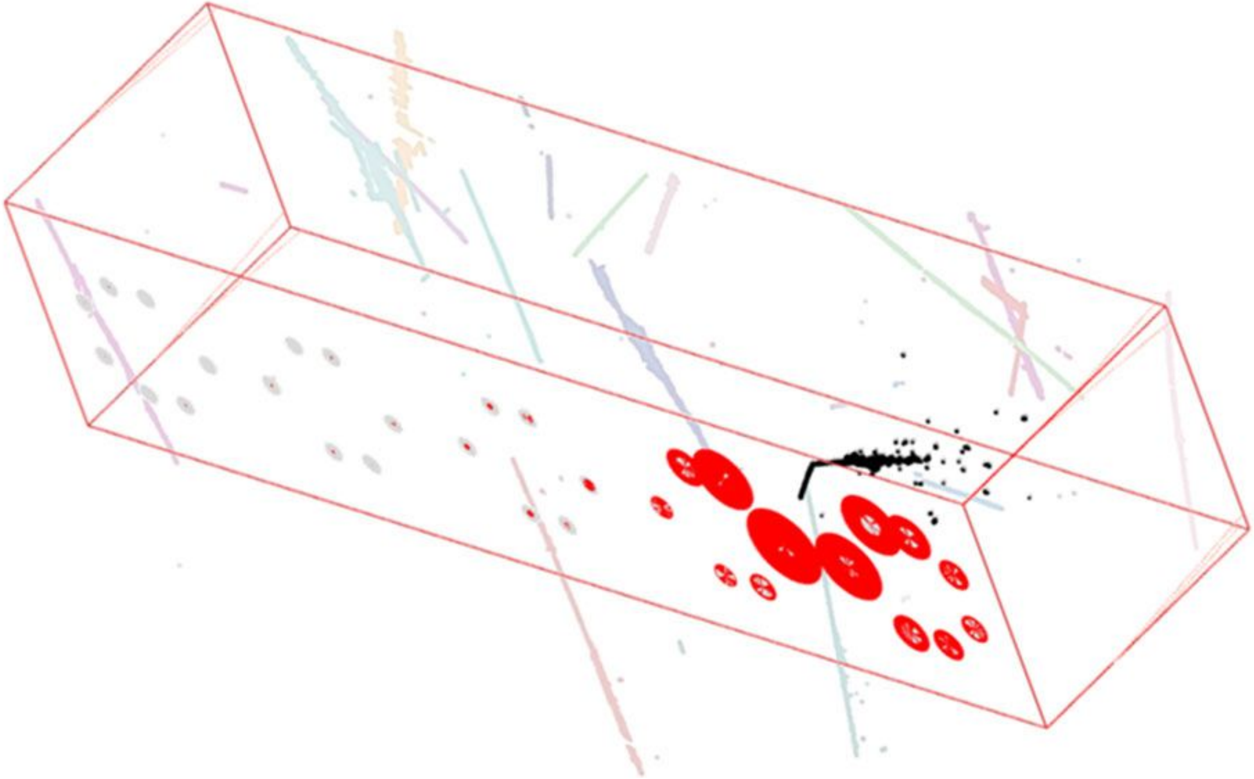
https://audeg.github.io/talks/talkAIP.pdf

Definitions: Frechet Inception Distance

$$d_F(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\mu',\Sigma'))^2 = \|\mu-\mu'\|_2^2 + \operatorname{tr}\left(\Sigma+\Sigma'-2\left(\Sigma^{\frac{1}{2}}\cdot\Sigma'\cdot\Sigma^{\frac{1}{2}}\right)^{\frac{1}{2}}\right)$$

From Wikipedia

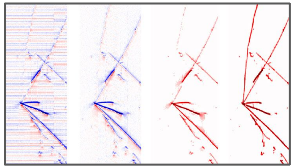
Flash Matched Cluster



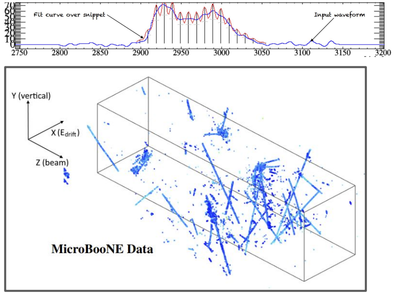
ML Reconstruction

Last week I talked primarily about Reconstruction for LArTPCs

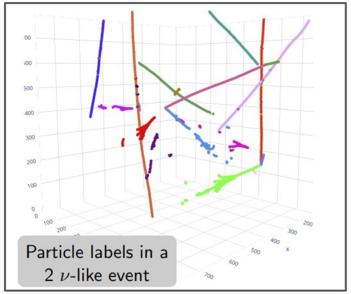
Deep neural networks having success in identifying key physical quantities for studying neutrino interactions



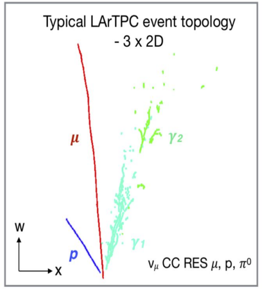
Signal conditioning



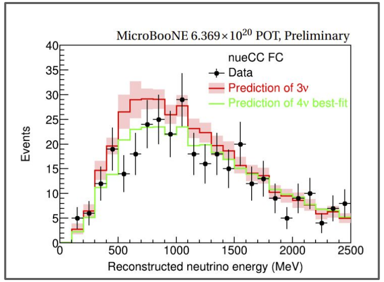
"Hit" finding (first-stage summary)



Clustering+particle ID



Interaction



Measurement Observable

Challenges

This talk will touch on early work to attack the other challenges

Reconstruction: are we getting all the information that we can from our data, precisely and accurately?

Simulation/Modeling: can we translate physics to observables faster? Can we better use data-driven methods?

Inference: are we testing our models against data as best as we can while accounting properly for and mitigating our model uncertainties?

Operations: are we saving the right events? Is the experiment running optimally? Can we detect and make decisions faster?

Fast ML on FPGAs to implement special rare process triggers

frai.2022.855184

Challenges

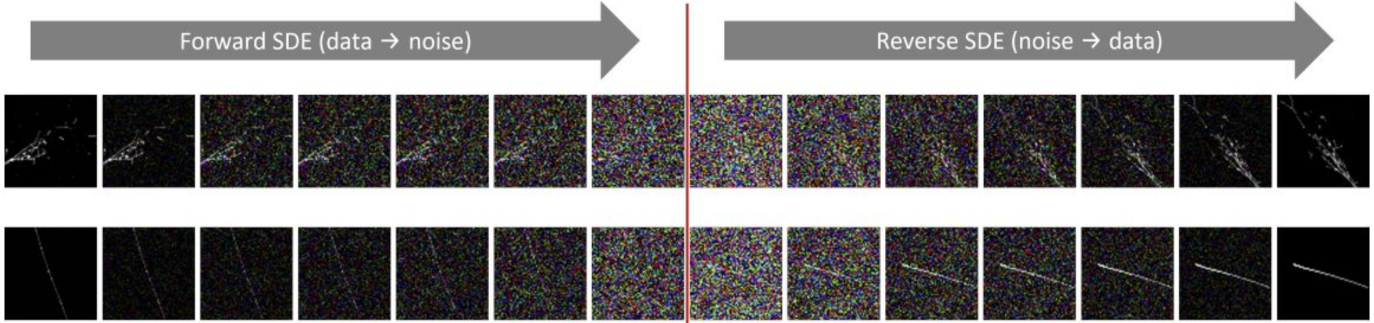
Simulation/Modeling: can we translate physics to observables faster? Can we better use data-driven methods?

Like in other frontiers, generating simulated data is a major bottleneck of analyses

- Example: Individual events can take ***5-10 mins/event+*** for MicroBooNE TPC with simulated cosmics + neutrino interaction
- Data driven methods are used to get better estimate: in MicroBooNE and SBN experiments which are on surface, cosmic background data is collected and used in simulated data by adding neutrino interaction – but cannot save enough of these events due to processing and storage constraints

Simulations - Generative Models

Score-based Generative Modeling shows promise for generating LArTPC-like images



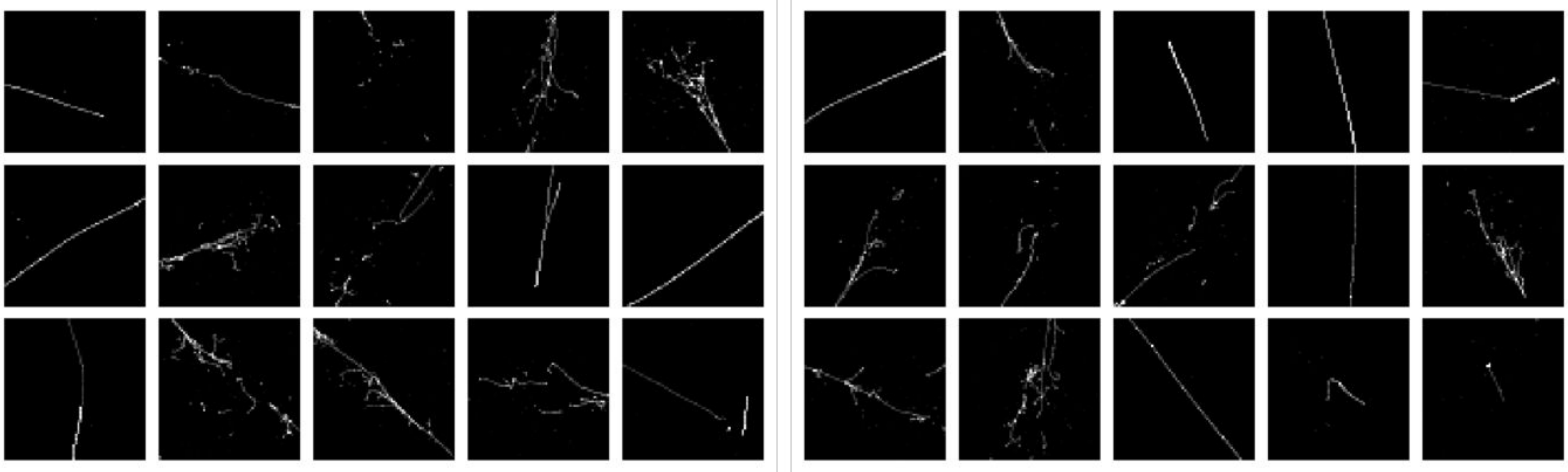
$$\mathrm{d}\vec{\boldsymbol{X}}_t = \mathbf{f}(\vec{\boldsymbol{X}}_t, t)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{W}_t$$

$$\mathrm{d}\overleftarrow{\boldsymbol{X}}_s = [-\mathbf{f}(\overleftarrow{\boldsymbol{X}}_s, s) + g^2(s)\overset{\text{score function}}{\boxed{\nabla \log p(\overleftarrow{\boldsymbol{X}}_s)}}]\mathrm{d}s + g(s)\mathrm{d}\boldsymbol{W}_s$$

Model generation as the reverse of a diffusion process bringing data images to noise

Simulations - Generative Models

Which set, left or right, is training images – which are generated?



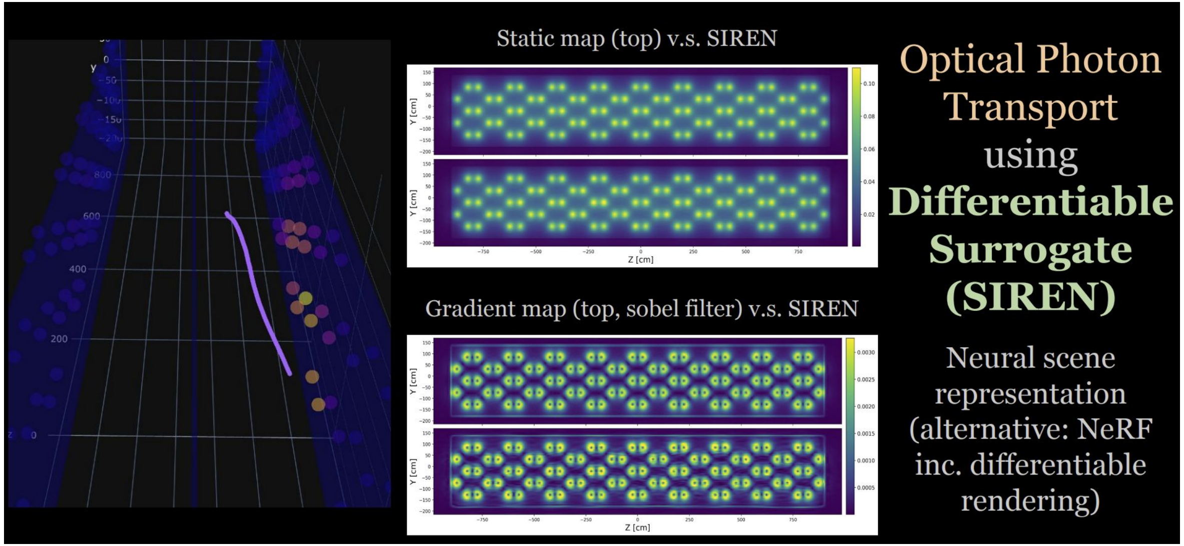
More details in next Wed. talk: "Generative Model Applications 3"

Simulations - Data Driven/Differentiable

Differentiable (surrogate) simulations save time by allowing a way to possible reweight MC events rather than generating additional samples with variation in detector physics parameters

Also enables simulation-based inference for a number of exciting applictions

https://arxiv.org/abs/2211.01505



One more Challenge

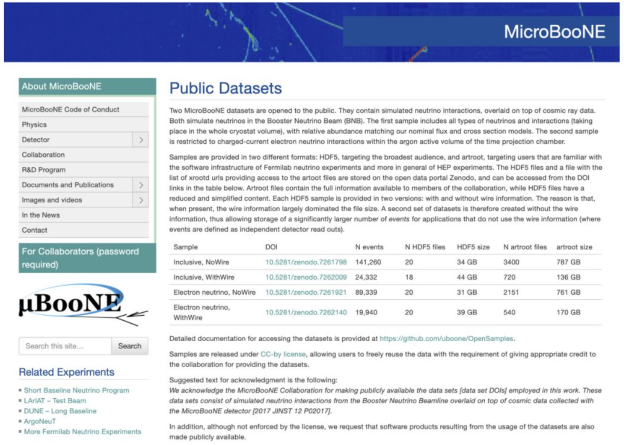
Accelerating Development: Are we providing enough tools to the community to enable new ideas and new contributors?

LArTPC Neutrino Interaction (Simulation) Dataset

MicroBooNE has released some LArTPC simulations: cosmic data overlaid with simulated neutrino interaction

Access point

- Entry point is the MicroBooNE website:
  - https://microboone.fnal.gov/documents-publications/public-datasets/



MicroBooNE

About MicroBooNE

- MicroBooNE Code of Conduct
- Physics
- Detector
- Collaboration
- R&D Program
- Documents and Publications
- Images and videos
- In the News
- Contact

For Collaborators (password required)

Search this site... Search

Related Experiments

- Short Baseline Neutrino Program
- LArIAT – Test Beam
- DUNE – Long Baseline
- ArgoNeuT
- More Fermilab Neutrino Experiments

Public Datasets

Two MicroBooNE datasets are opened to the public. They contain simulated neutrino interactions, overlaid on top of cosmic ray data. Both simulate neutrinos in the Booster Neutrino Beam (BNB). The first sample includes all types of neutrinos and interactions (taking place in the whole cryostat volume), with relative abundance matching our nominal flux and cross section models. The second sample is restricted to charged-current electron neutrino interactions within the argon active volume of the time projection chamber.

Samples are provided in two different formats: HDF5, targeting the broadest audience, and artroot, targeting users that are familiar with the software infrastructure of Fermilab neutrino experiments and more in general of HEP experiments. The HDF5 files and a file with the list of xrootd urls providing access to the artoot files are stored on the open data portal Zenodo, and can be accessed from the DOI links in the table below. Artroot files contain the full information available to members of the collaboration, while HDF5 files have a reduced and simplified content. Each HDF5 sample is provided in two versions: with and without wire information. The reason is that, when present, the wire information largely dominated the file size. A second set of datasets is therefore created without the wire information, thus allowing storage of a significantly larger number of *events* for applications that do not use the wire information (where events are defined as independent detector read outs).

Sample	DOI	N events	N HDF5 files	HDF5 size	N artroot files	artroot size
Inclusive, NoWire	10.5281/zenodo.7261798	141,260	20	34 GB	3400	787 GB
Inclusive, WithWire	10.5281/zenodo.7262009	24,332	18	44 GB	720	136 GB
Electron neutrino, NoWire	10.5281/zenodo.7261921	89,339	20	31 GB	2151	761 GB
Electron neutrino, WithWire	10.5281/zenodo.7262140	19,940	20	39 GB	540	170 GB

Detailed documentation for accessing the datasets is provided at https://github.com/uboone/OpenSamples.

Samples are released under CC-by license, allowing users to freely reuse the data with the requirement of giving appropriate credit to the collaboration for providing the datasets.

Suggested text for acknowledgment is the following:
We acknowledge the MicroBooNE Collaboration for making publicly available the data sets [data set DOIs] employed in this work. These data sets consist of simulated neutrino interactions from the Booster Neutrino Beamline overlaid on top of cosmic data collected with the MicroBooNE detector [2017 JINST 12 P02017].

In addition, although not enforced by the license, we request that software products resulting from the usage of the datasets are also made publicly available.

Description

Links to Zenodo

Link to documentation

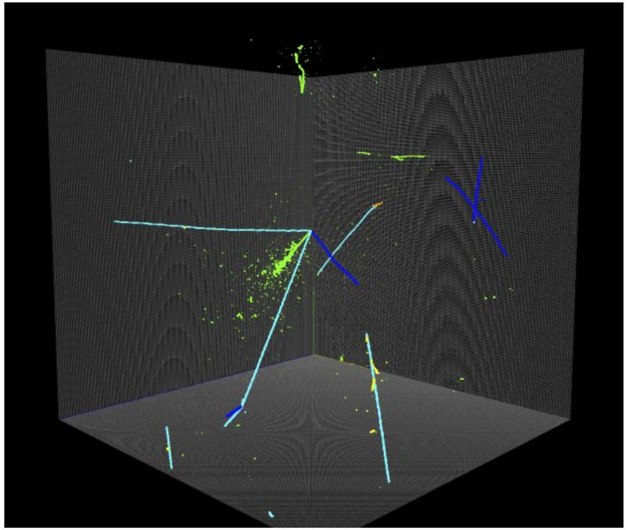
Info about license and citation

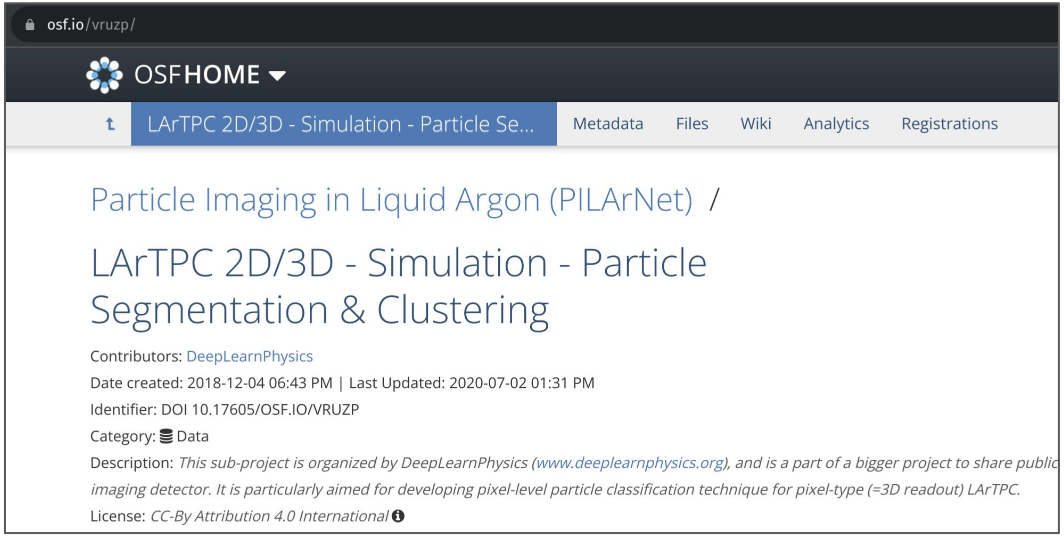
Fermilab

10 2023/05/09 G. Cerati (FNAL)

LArTPC Neutrino Interaction (Simulation) Dataset

Another public dataset of 3D voxels along with simple 2D projections





https://arxiv.org/abs/2006.01993

https://osf.io/vruzp

Conclusions

- ML techniques have already impacted the physics program of several neutrino experiments
- Many develops across experiments should provide further impact
- Developments have been somewhat focused on reconstruction
- But there are still other research challenges that ML might help to advance
- Cross pollination between experiments and frontiers will surely accelerate progress

BACK UP SLIDES

Sampling with Reverse Process

With the choices for:

β(t)

$$t_i = T + (i - M)\Delta t$$

$$\Delta t = -(T-\epsilon)/(M-1)$$

$$\epsilon = 1.0\times 10^{-3}$$

Point is not to get too into the weeds, but can integrate

$$\vec{\boldsymbol{\mu}}_{t_{i-1}} = \vec{\boldsymbol{X}}_{t_i} + [-\frac{1}{2}\beta(t_i)\vec{\boldsymbol{X}}_{t_i} - \beta(t_i)\boldsymbol{s_\theta}(\vec{\boldsymbol{X}}_{t_i}, t_i)]\Delta t$$

$$\vec{\boldsymbol{X}}_{t_{i-1}} = \vec{\boldsymbol{\mu}}_{t_{i-1}} + z\sqrt{-\beta(t_i)\Delta t},$$

$$z = \mathcal{N}(0, I)$$

Sampled Brownian motion direction

Sampling with Reverse Process

With the choices for:

β(t)

$$t_i = T + (i - M)\Delta t$$

$$\Delta t = -(T-\epsilon)/(M-1)$$

$$\epsilon = 1.0\times 10^{-3}$$

$$\vec{\boldsymbol{\mu}}_{t_{i-1}} = \vec{\boldsymbol{X}}_{t_i} + [-\frac{1}{2}\beta(t_i)\vec{\boldsymbol{X}}_{t_i} - \beta(t_i)\boldsymbol{s}_{\boldsymbol{\theta}}(\vec{\boldsymbol{X}}_{t_i}, t_i)]\Delta t$$

$$\vec{\boldsymbol{X}}_{t_{i-1}} = \vec{\boldsymbol{\mu}}_{t_{i-1}} + z\sqrt{-\beta(t_i)\Delta t},$$

Transport Problem

- Comparison of two sampled distributions

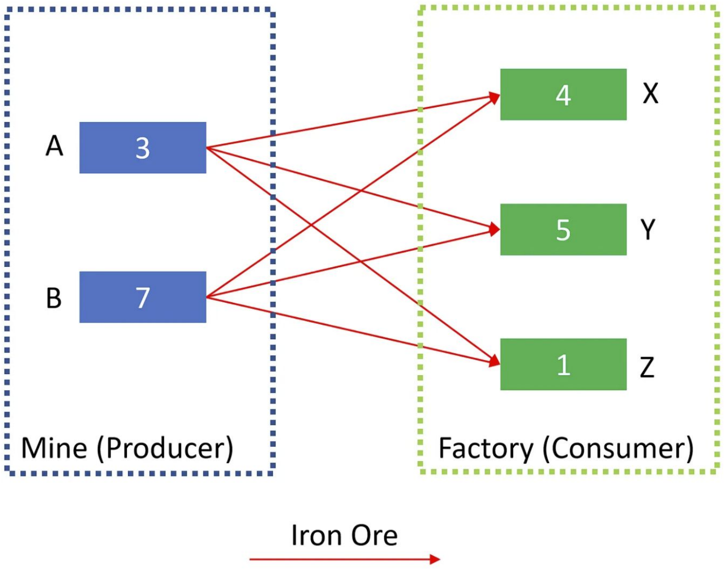


Fig. 1. Illustration of the setup. We need to transport the iron ore from the mines to the factories, which is the transportation problem.

Find P_{ij} that minimizes total cost

$$d = \min \sum_{i,j} P_{i,j} C_{i,j}$$

$$i \in \{A, B\}$$

$$j \in \{X, Y, Z\}$$

$$P_{i,j} \geq 0$$

$$\sum_j P_{i,j} = r_i$$

$$\sum_i P_{i,j} = c_j$$

Wasserstein vs. Sinkhorn Cost Optimization

- Sinkhorn distance relaxes constrain on P_{ij} and imposes term that

$$\boxed{d = \min \sum_{i,j} P_{i,j} C_{i,j}} \qquad \boxed{d^* = \min \sum_{i,j} P_{i,j} C_{i,j}}$$

$$\boxed{\begin{array}{c} P_{i,j} \geq 0 \\ \sum_j P_{i,j} = r_i \\ \sum_i P_{i,j} = c_j \end{array}} \Rightarrow \boxed{\begin{array}{c} \cancel{P_{i,j} \geq 0} \\ \sum_j P_{i,j} = r_i \\ \sum_i P_{i,j} = c_j \end{array}}$$

$$\boxed{KL(\mathrm{P}|\mathrm{rc}^T) \leq \alpha}$$

AE-OT

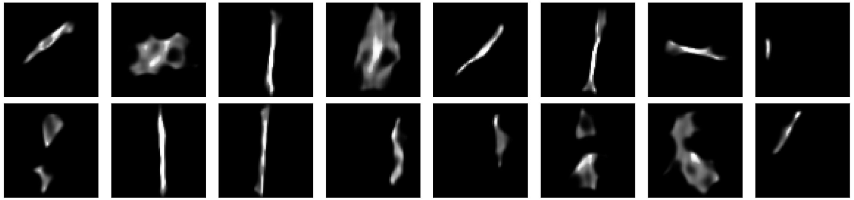
To get around how “thin” the image space might be, we trained an autoencoder to learn embedding for images into a lower dimensional space

Train Generator to map from ***Z*** to embedding space ***H***

Used Optimal Transport to measure how close or far generated examples are from real ones. [AE-OT]

Training examples

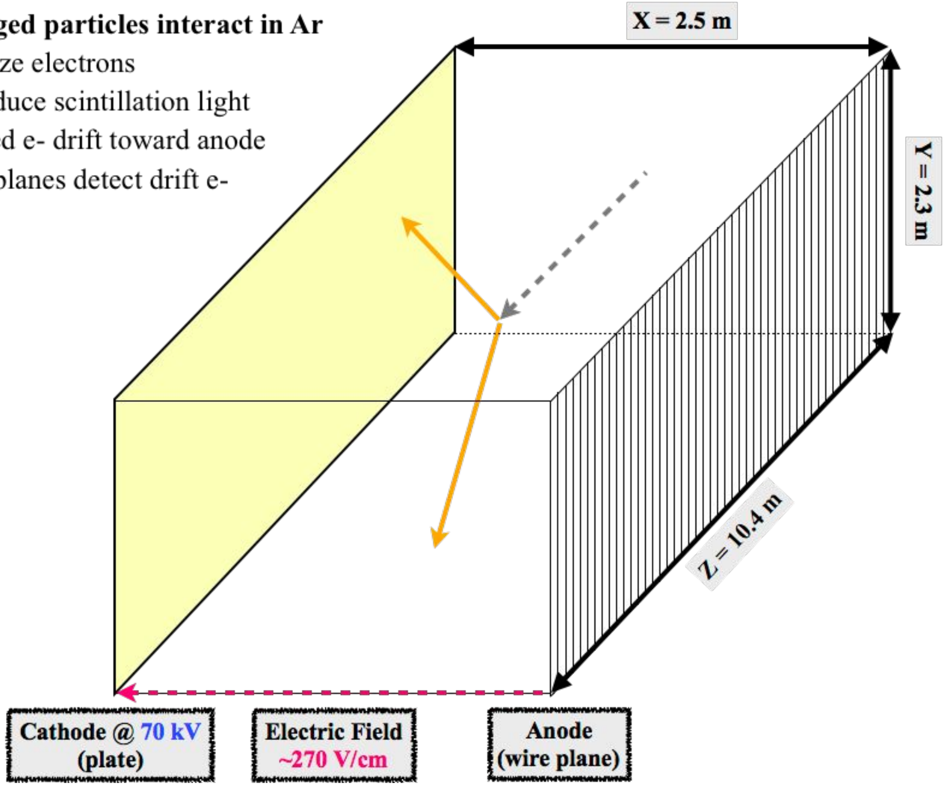
Generated examples



LArTPC Primer

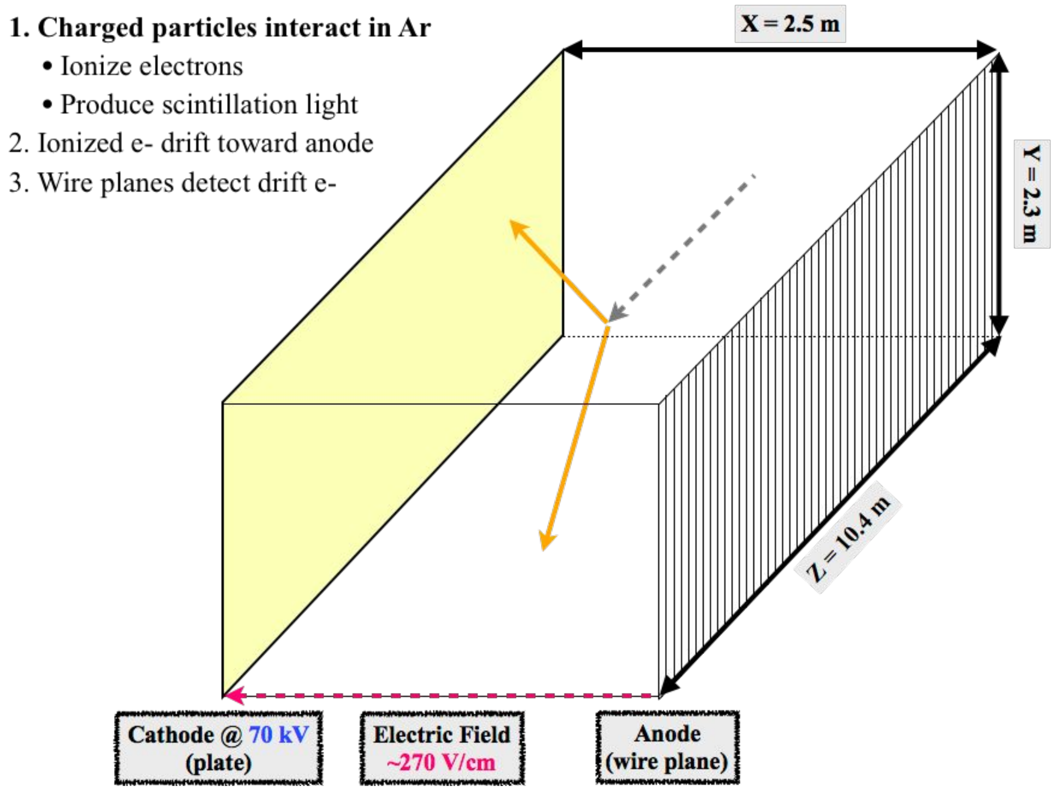
Capturing Neutrino Interaction Images w/ LArTPCs

1. **Charged particles interact in Ar**
    - Ionize electrons
    - Produce scintillation light
2. Ionized e- drift toward anode
3. Wire planes detect drift e-



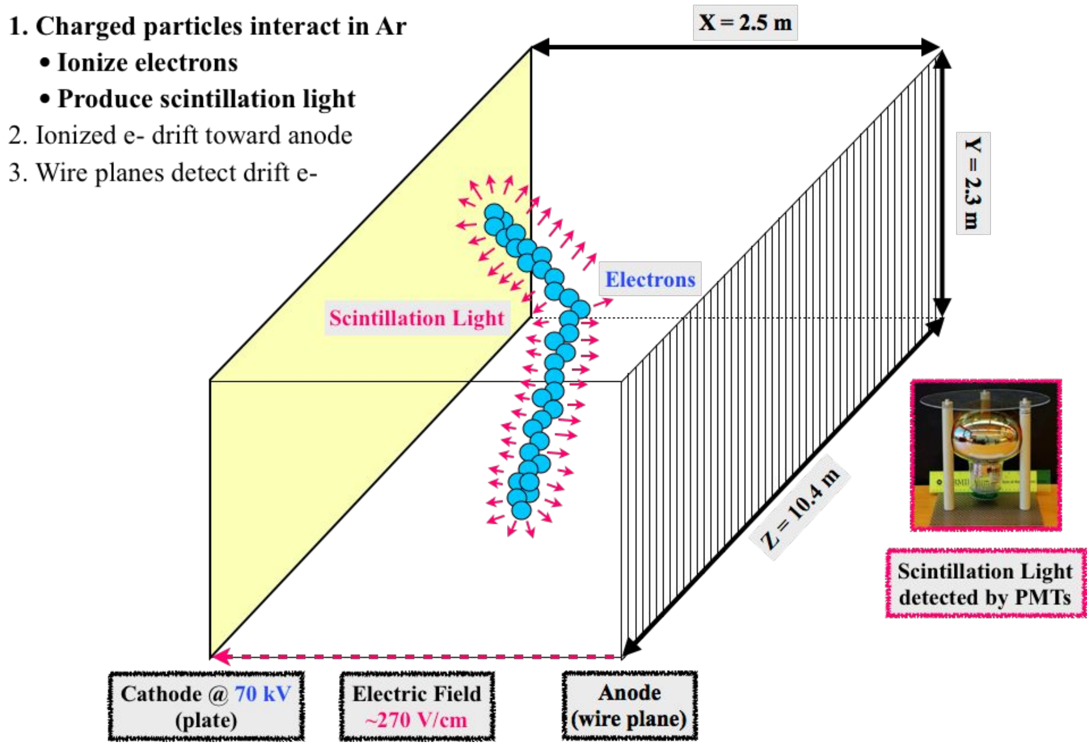
MicroBooNE TPC

Capturing Neutrino Interaction Images w/ LArTPCs

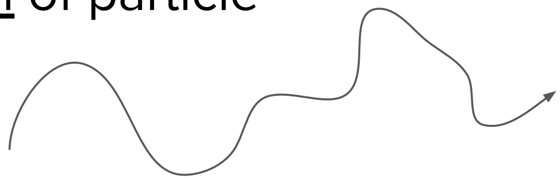


A neutrino (dashed grey) passes into the detector and interacts producing charged particles (solid yellow)

Capturing Neutrino Interaction Images w/ LArTPCs

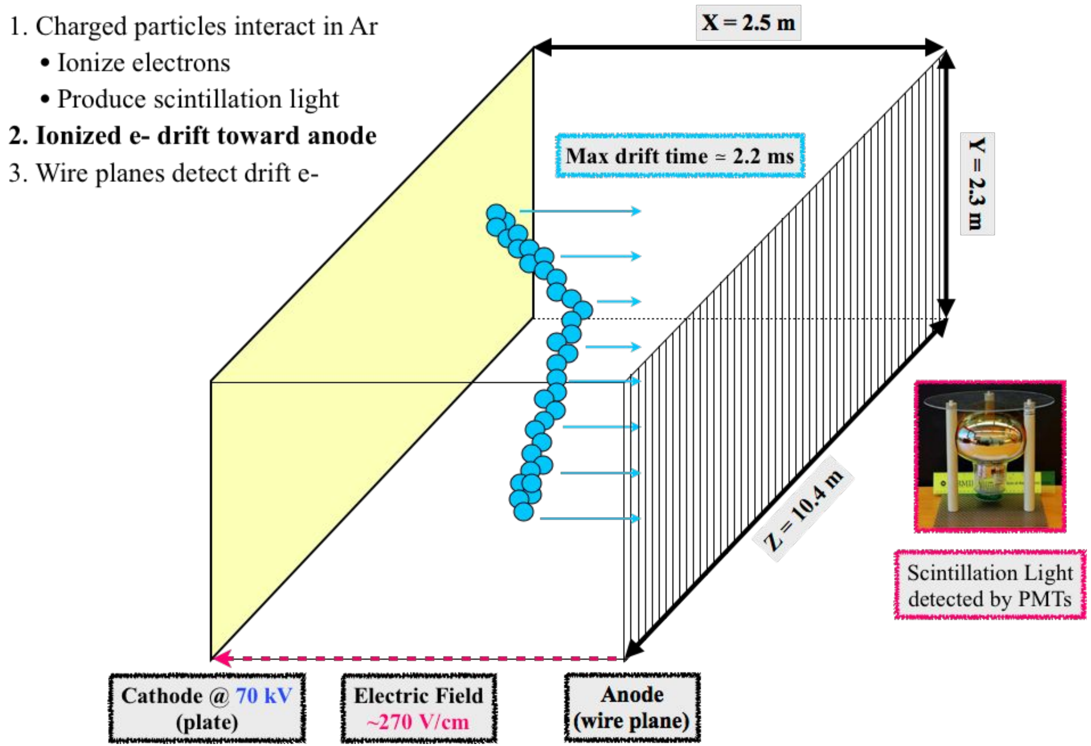


Charged particles produce ionization electrons: tell us <u>path</u> of particle



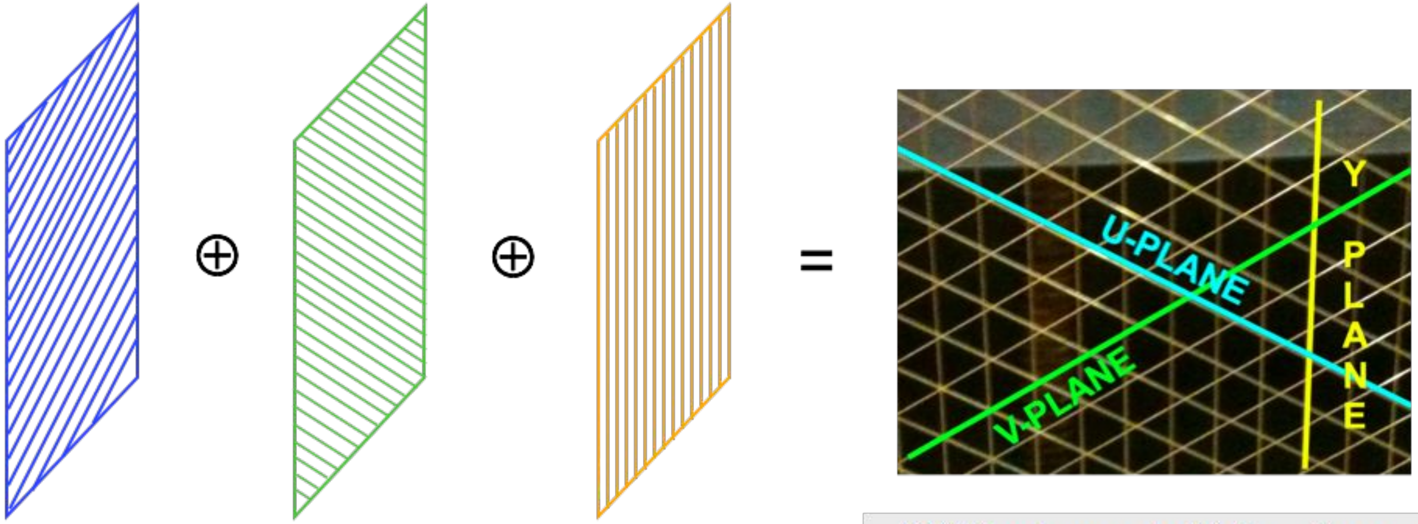
Also produces *light*: tells us time of the event

Capturing Neutrino Interaction Images w/ LArTPCs



Ionization electrons drift towards wireplanes

Capturing Neutrino Interaction Images w/ LArTPCs



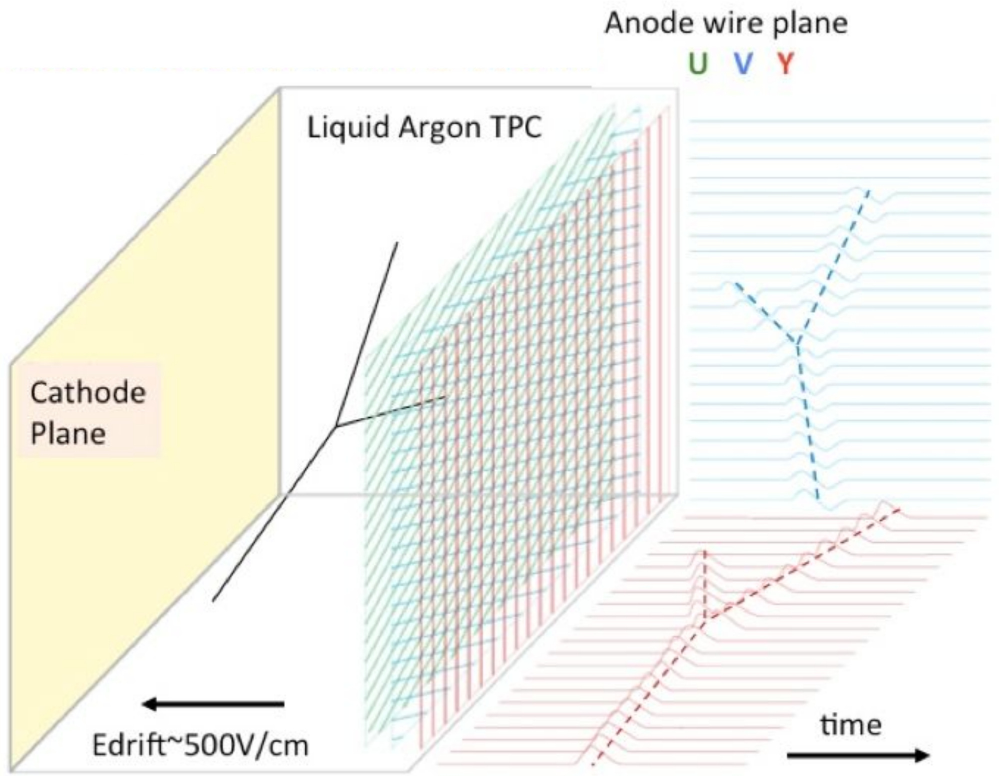
U plane (induction)

V plane (induction)

Y plane (collection)

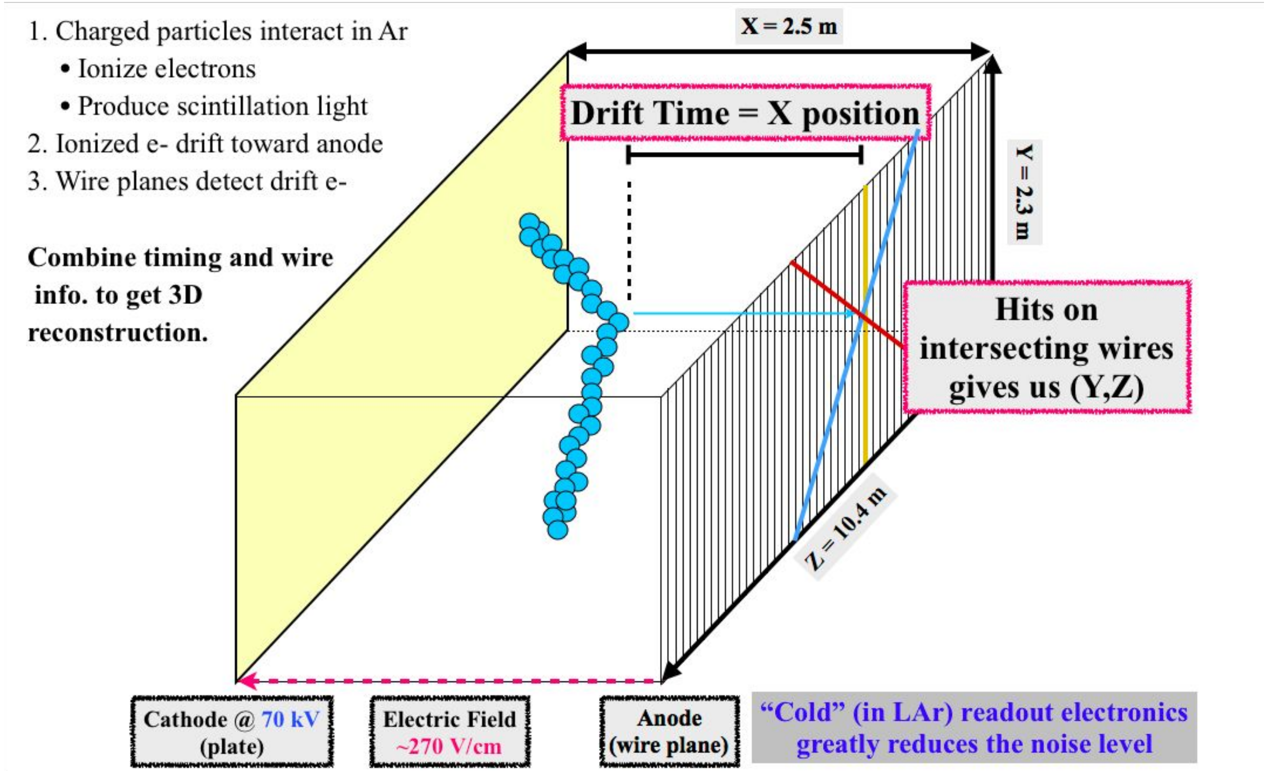
8256 wires w/ pitch = 3mm
(Y, Z) = coincidence on wire

Capturing Neutrino Interaction Images w/ LArTPCs



Recording wire signals over time, detector produces image-like data

Capturing Neutrino Interaction Images w/ LArTPCs



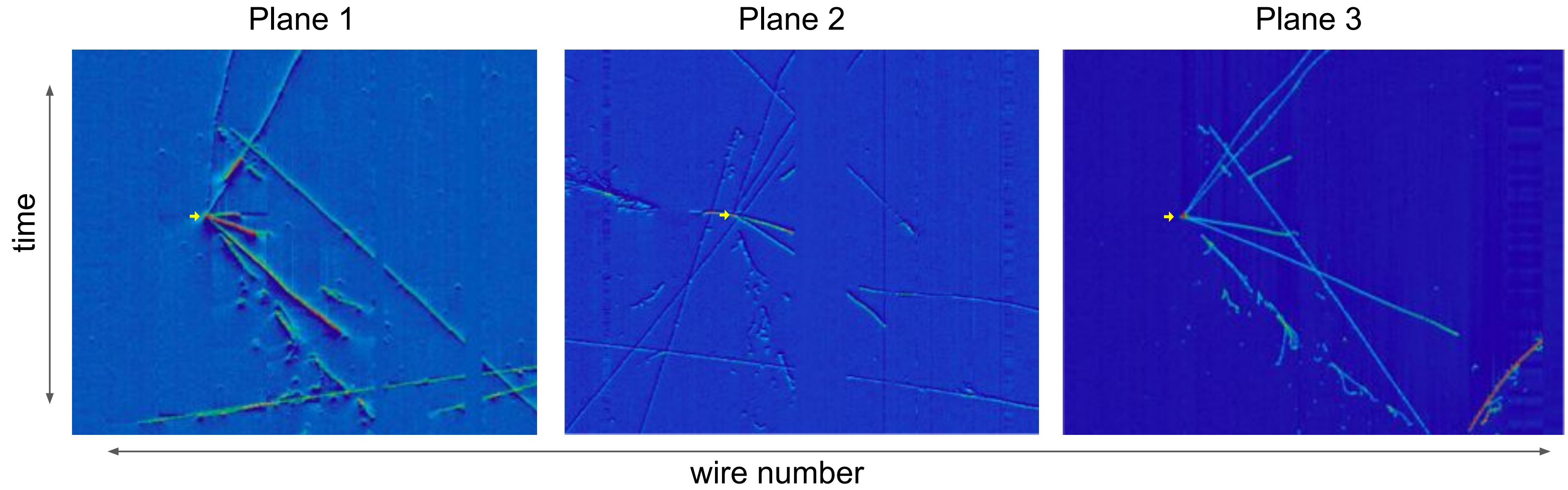
Capture 3 projection images with wire planes

Can solver inverse problem to recover 3D energy deposits

Ionization signals on wires coincident in time provide info for (Y,Z) position

X position given by time delay from light signal

Capturing Neutrino Interaction Images w/ LArTPCs



Example of data event in MicroBooNE. View of same event for each projection.

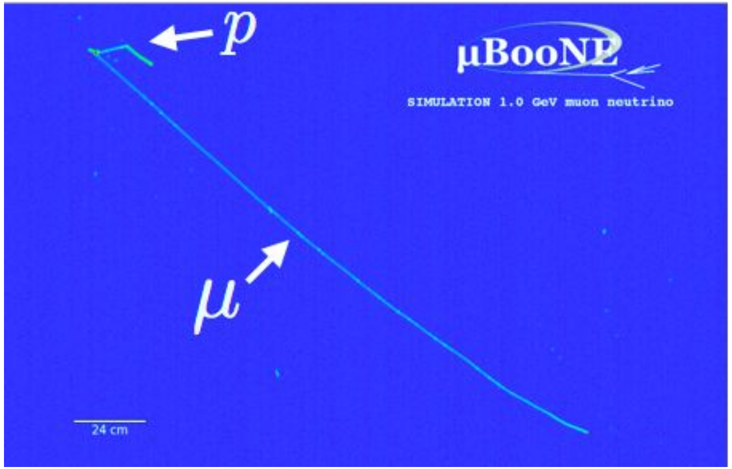
Color scale indicates amount of ionization electrons seen on wire at given time

Capturing Neutrino Interaction Images w/ LArTPCs

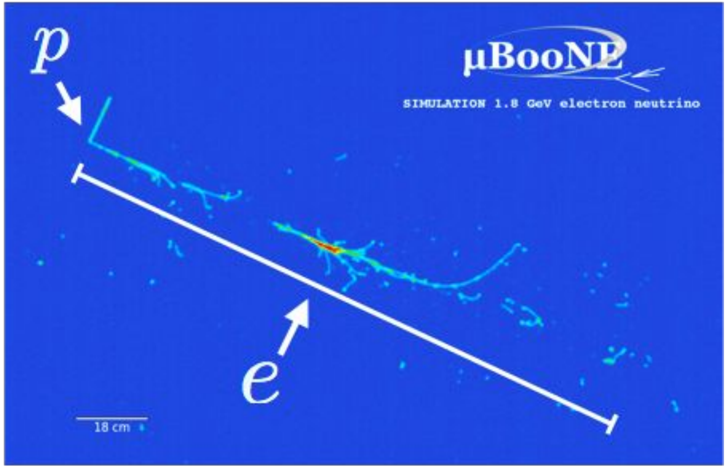
Flavor determined from finding partner lepton (muon,electron) produced in interaction

Neutrino energy inferred from momenta of resulting particles

$$\nu_\mu + n \rightarrow \mu + p$$



$$\nu_e + n \rightarrow e + p$$



Neutrino Oscillation Analysis

Δm^2
θ

+

ν Flux model

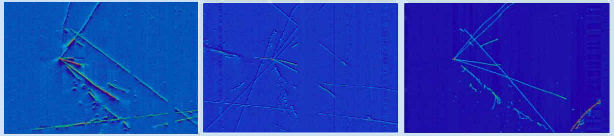
ν-nucleus interaction model

Detector Sim

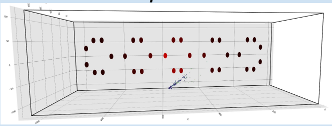
(Truth) final-state particles
$$\{p_1, p_2, p_3, ... p_n\}$$

Detector Outputs

Ionization



Optical

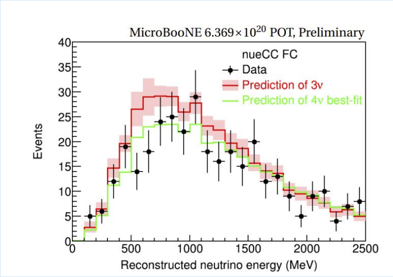


Reconstruction

(reco) final-state particles
$$\{\tilde{p}_1, \tilde{p}_2, \tilde{p}_3, ... \tilde{p}_m\}$$

Calculate observables

Observables



Neutrino Oscillations

Neutrino oscillations occur because

- neutrinos have mass
- and flavor states are mixture of mass states

Neutrinos <u>created/interact</u> in flavor states
They <u>propagate</u> in their mass states

Don't have to line up!

Neutrino Oscillation: 2-flavor example

Neutrino oscillations occur because

- neutrinos have mass
- and flavor states are mixture of mass states

Flavor States | **Mixing Matrix (*U*)** | Mass States

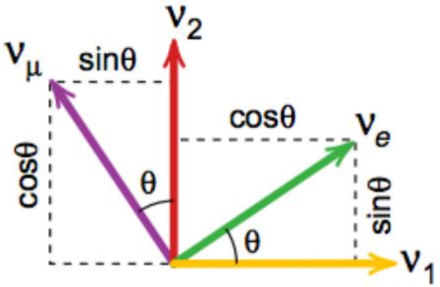
Example 2 ν model

$$\begin{pmatrix} \nu_a \\ \nu_b \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix} \begin{matrix} \text{w/mass } m_1 \\ \text{w/mass } m_2 \end{matrix}$$

In 2D, simple rotation

$$|\nu_e\rangle = \cos\theta|\nu_1\rangle + \sin\theta|\nu_2\rangle$$

$$|\nu_\mu\rangle = -\sin\theta|\nu_1\rangle + \cos\theta|\nu_2\rangle$$



Neutrino Oscillation: 2-flavor example

Let's start with a neutrino created in flavor state $|\nu_\mu\rangle$

Neutrino Oscillations

Want to know state at time, *t*, so we need to apply the propagator, *U(t)*, on the $|\nu_\mu\rangle$ in the mass basis

$$U(t)U_{PMNS}^{-1}|\nu_\mu\rangle = -\sin\theta e^{-iE_1 t}|\nu_1\rangle + \cos\theta e^{-iE_2 t}|\nu_2\rangle$$

$$E = \sqrt{p^2 + m^2}$$

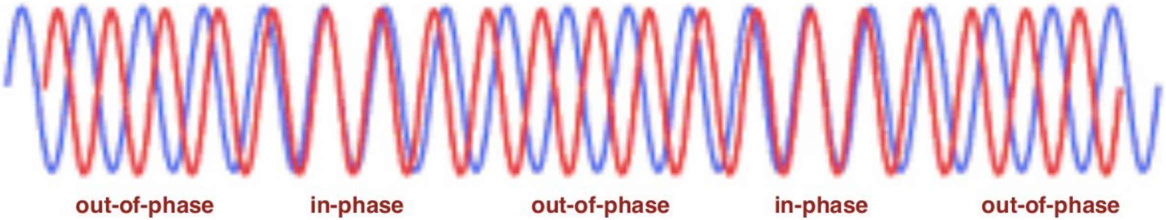
Neutrino Oscillations

Want to know state at time, *t,* so we need to apply the propagator, *U(t)*, on the $|\nu_\mu\rangle$ in the mass basis

$$U(t)U_{PMNS}^{-1}|\nu_\mu\rangle = -\sin\theta e^{-iE_1 t}|\nu_1\rangle + \cos\theta e^{-iE_2 t}|\nu_2\rangle$$

$$E = \sqrt{p^2 + m^2}$$

what we have are two states oscillating at slightly two frequencies due to the slightly different masses



with a slight shift in frequency, you get periods where two waves in phase for some time, and out of phase for others

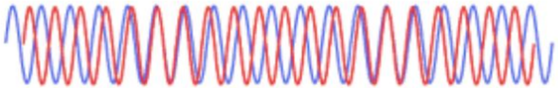
Neutrino Oscillations

Want to know state at time, *t,* so we need to apply the propagator, *U(t)*, on the $|\nu_\mu\rangle$ in the mass basis

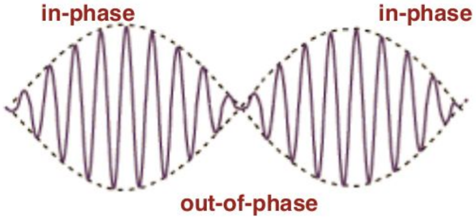
$$U(t)U_{PMNS}^{-1}|\nu_\mu\rangle = -\sin\theta e^{-iE_1 t}|\nu_1\rangle + \cos\theta e^{-iE_2 t}|\nu_2\rangle$$

$$E = \sqrt{p^2 + m^2}$$

what we have are two states oscillating at slightly two frequencies due to the different masses



Classical analogue is the beat-frequency phenomenon



Neutrino Oscillation: 2-flavor example

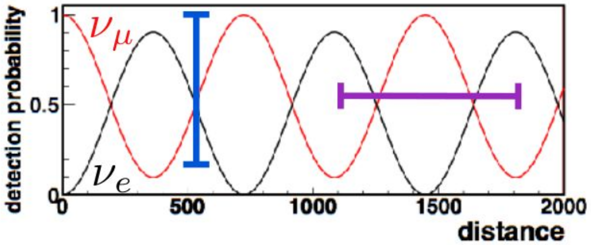
Probability of transition from flavor ν_μ to ν_e :

$$P(\nu_\mu \to \nu_e) = |\langle \nu_e | U_{PNMS} U(t) U^{-1}_{PNMS} | \nu_\mu \rangle|^2$$

$$= \sin^2 2\theta \sin^2 \left(([1.27\ \mathrm{GeV\ km}^{-1}] \Delta m^2 \boxed{\frac{L}{E}} \right)$$

Key signature is oscillatory prob function of L/E

Result is sinusoidal probability, function of L/E



mixing angle, θ, governs amplitude

mass splitting, Δm^2, governs frequency

Neutrino Oscillation: 2-flavor example

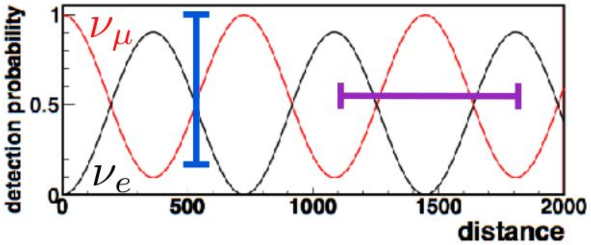
Probability of transition from flavor ν_μ to ν_e :

$$P(\nu_\mu \rightarrow \nu_e) = |\langle \nu_e | U_{PNMS} U(t) U^{-1}_{PNMS} | \nu_\mu \rangle|^2$$

$$= \sin^2 2\theta \sin^2 \left(([1.27 \text{ GeV km}^{-1}] \Delta m^2 \boxed{\frac{L}{E}} \right)$$

Key signature is oscillatory prob function of L/E

Result is sinusoidal probability, function of L/E



mixing angle, θ, governs amplitude

mass splitting, Δm^2, governs frequency

Neutrino nucleon Interactions

Never scattering on a free quark.

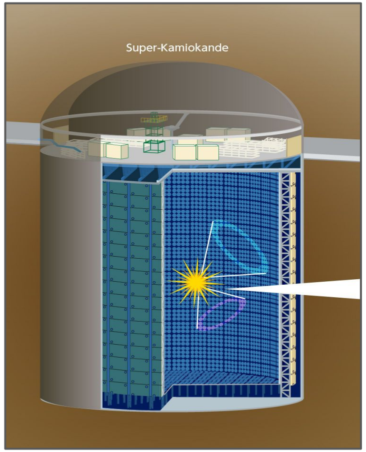
Dominant interactions at ν_μ at $E_\nu \sim 1\,\mathrm{GeV}$
(typical flavor and energy for accelerator ν)

Cherenkov

Other experiments analyze the pattern of Cherenkov Radiation to infer particle momenta and type.

Spatial arrange of optical sensors not grid-like

Example: Super-Kamiokande, T2K, Hyper-K

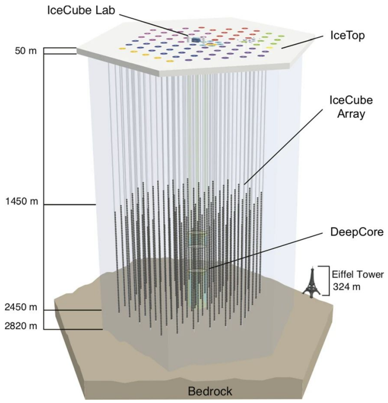


Muons
(sharp ring)

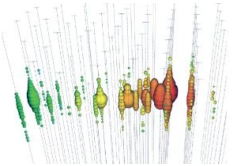
EM showers
(fuzzy ring)

Example: IceCube

Observing atmospheric + astrophysical neutrinos



Muon
(from $\nu_\mu CC$)



Electron or Tau
(from $\nu_e CC$ or $\nu_\tau CC$)