FERMILAB-SLIDES-24-0314-STUDENT

Training NuGraph2 for ICARUS

Eric Novello

2024 Fall SULI Final Presentation

December 4, 2024

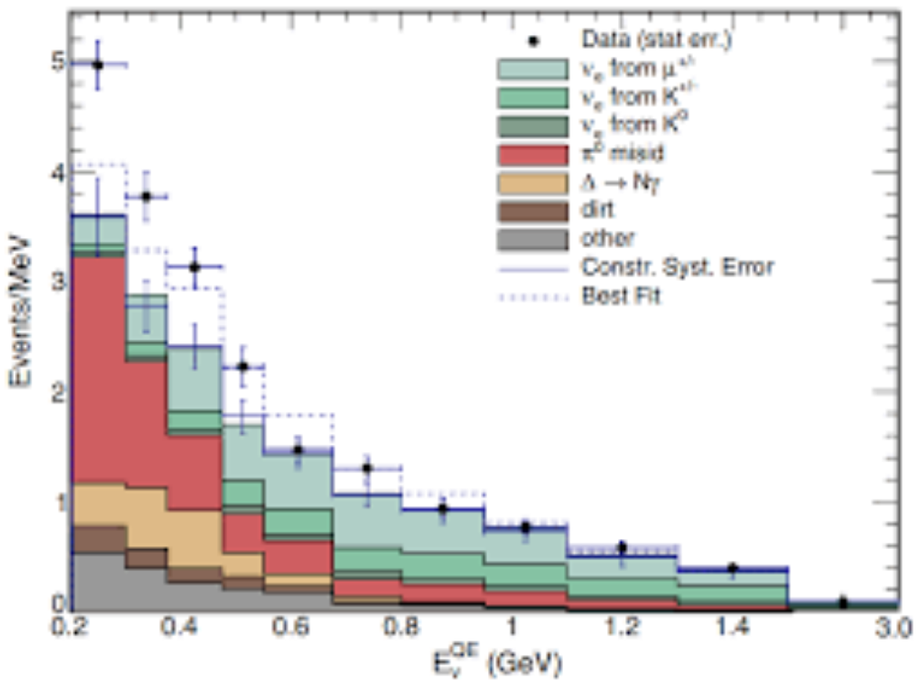
Supervisors:

Dr. Sunny Seo

Dr. Giuseppe Cerati

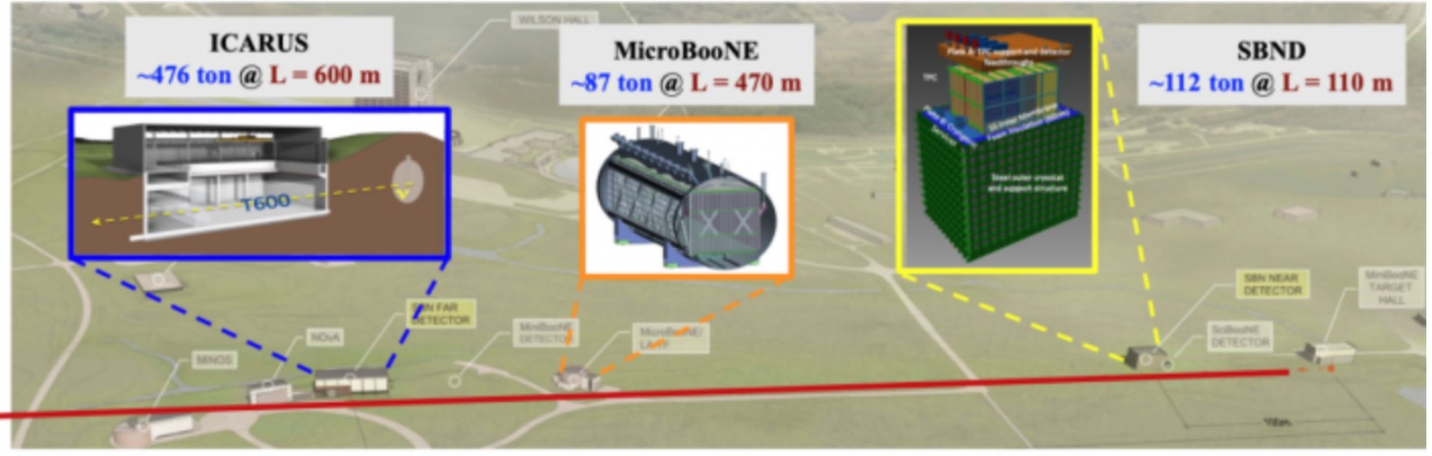
This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.
This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships Program (SULI).

Introduction to ICARUS

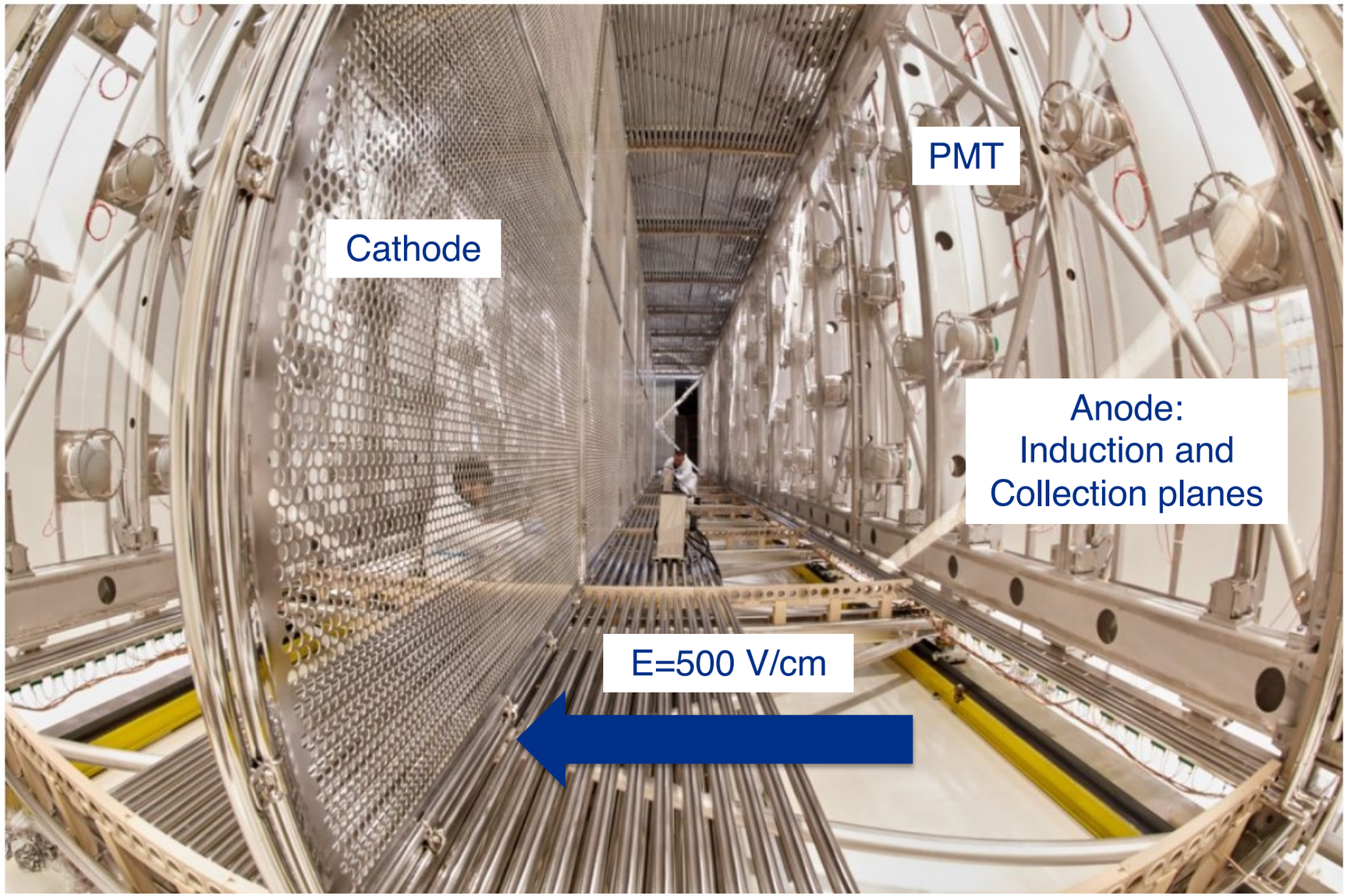


MiniBooNE results that show the excess of energy that could be explained by a fourth generation of neutrino oscillation

- ICARUS aims to confirm or refute the existence of a fourth generation of neutrino oscillation as indicated from other neutrino experiments such as MiniBooNE.
  - ICARUS is one of three experiments in the Short Baseline Neutrino program.
  - The Short Baseline Neutrino Program mainly utilizes the **Booster Neutrino Beam**, provides it with **neutrinos** produced by **8 GeV protons on target**.



ICARUS Detector



Fermilab

ICARUS Geometry

- ICARUS is comprised of **four** Liquid Argon Time Projection Chambers (**LArTPC**) in **two identical cryostats** filled with total **760 tons liquid Argon**
  - LArTPCs in each cryostat separated by cathode
  - Edges of cryostats have anode (2 induction and one collection wire planes).
    - Electric field between cathode and anode **accelerates ionized electrons** to **anode**
    - LArTPCs in each cryostat need their **data to be "stitched" together** for use in NuGraph2

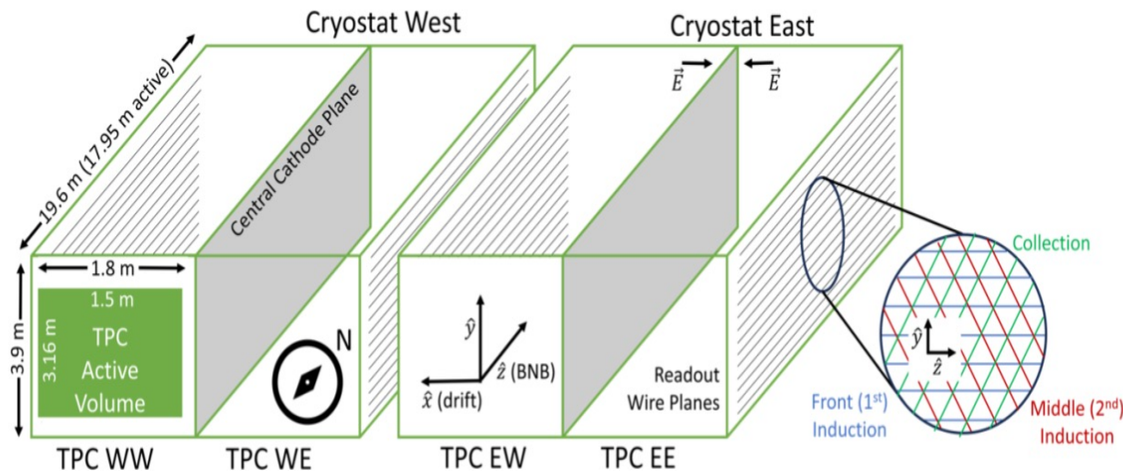
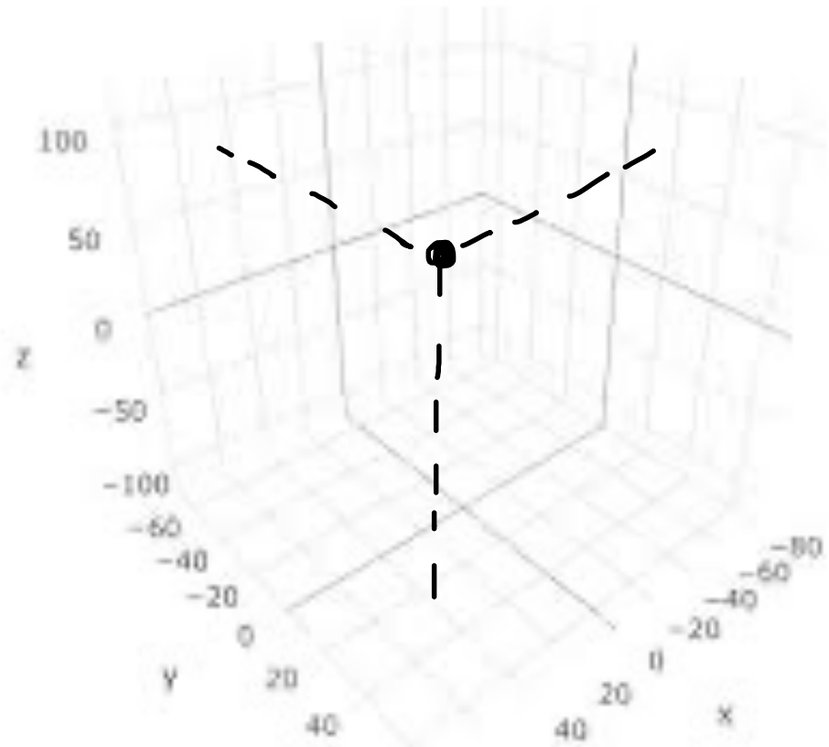


Diagram of ICARUS detector geometry

Fermilab

Hits and Spacepoints



Simplified Diagram of How Spacepoints are made from hits (A.U.)

- Gaussian Pulses on TPC wires form hits which are input to graph nodes
- **Hits** are found by **Gaussian hit finder module.**
  - Wire numbers and times from hits across planes are converted to cartesian coordinates, **spacepoints,** produced by **Cluster3D module**

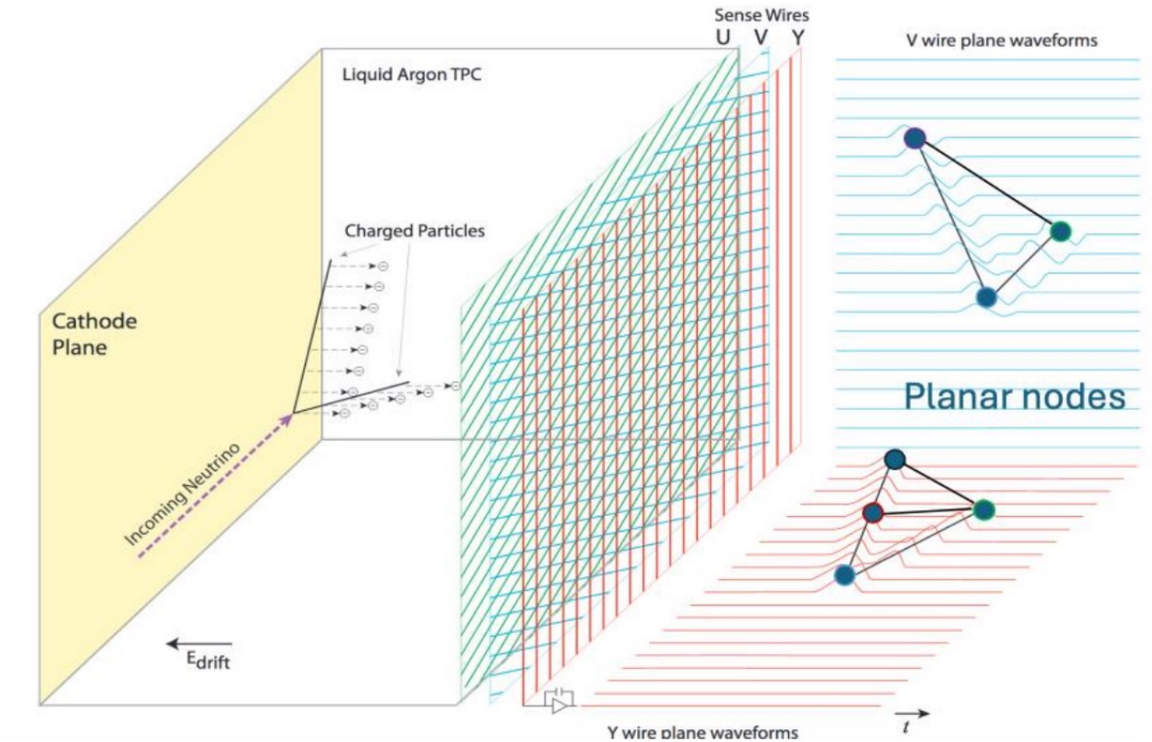
Fermilab

NuGraph2

- Graph Neural Network (GNN) produces **hit labels** for **particle identification**, **event reconstruction**

- While CNNs require grid-like inputs and outputs format, GNNs operate on more flexible data formats and are naturally sparse

- NuGraph2 has data in form of **planar** (hits) and **nexus** (spacepoints) **graph nodes** connected with **edges.**
- Planar nodes connected to each other, **nexus** nodes only **connected** to **planar** nodes
- Already implemented in MicroBooNE
- Want to implement NuGraph2 in ICARUS
- Nugraph2 uses **same inputs** as **other reconstruction algorithms** such as Pandora



Fermilab

NuGraph2 Training Workflow

1. Spacepoints production
2. TPC stitching

Credit Ricardo Campos, summer intern

3. Spacepoint filtering

My Work

   - Number of associated hits, χ^2, neutrino slices

4. Reassign run numbers, Concatenate run files into single file
5. Preprocessing

   - Removing bad data, forming graph nodes and edges

6. Training
7. Parameter optimization based on NuGraph2 training performance

Next Steps

Stitching TPCs

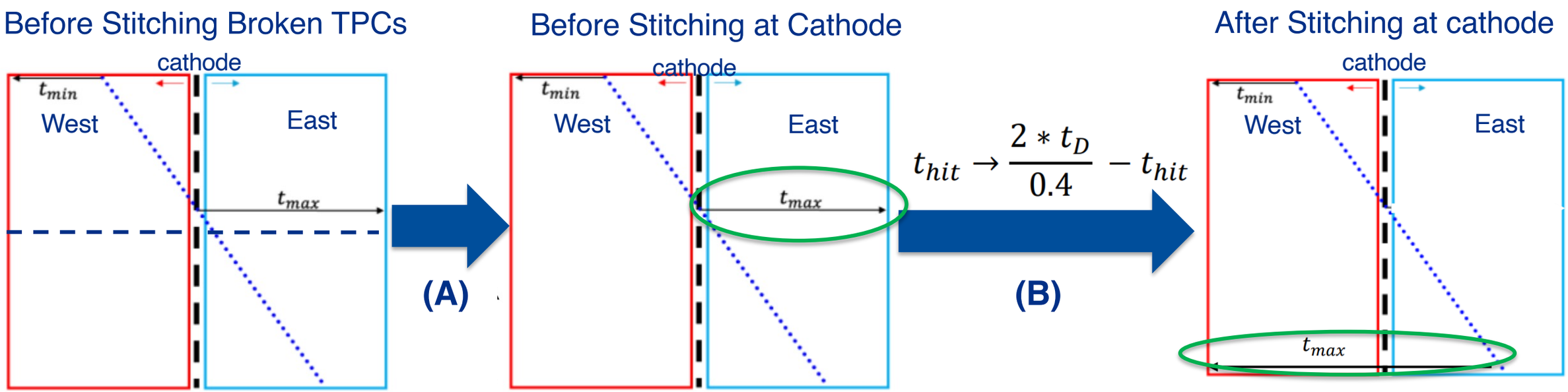
Credit Ricardo Campos, summer intern

Each Cryostat split in four logical TPCs, NuGraph2 wants each cryostat as a single TPC. There are two different stitches (A) and (B):

(A) Each TPC is broken into two at middle of induction plane 1, stitched by offset of wire IDs

(B) TPCs in cryostat are stitched across the cathode

- TPCs east of the cathode have global time equal to local time subtracted from constant



Fermilab

Filtering Spacepoints

Initial data full of noise and background spacepoints, which need to be reduced

- Two filtering techniques (A) and (B) investigated separately then combined (C)

(A) Filtering spacepoints based off χ^2 value

- Setting χ^2 cutoff too low can lead to loss of signal data
- Histogram used to find reasonable values of χ^2 to filter out without loss of track data
  - Spacepoints above χ^2 cutoff **filtered out**

(B) Filtering spacepoints based off number of associated hits in the three wire planes

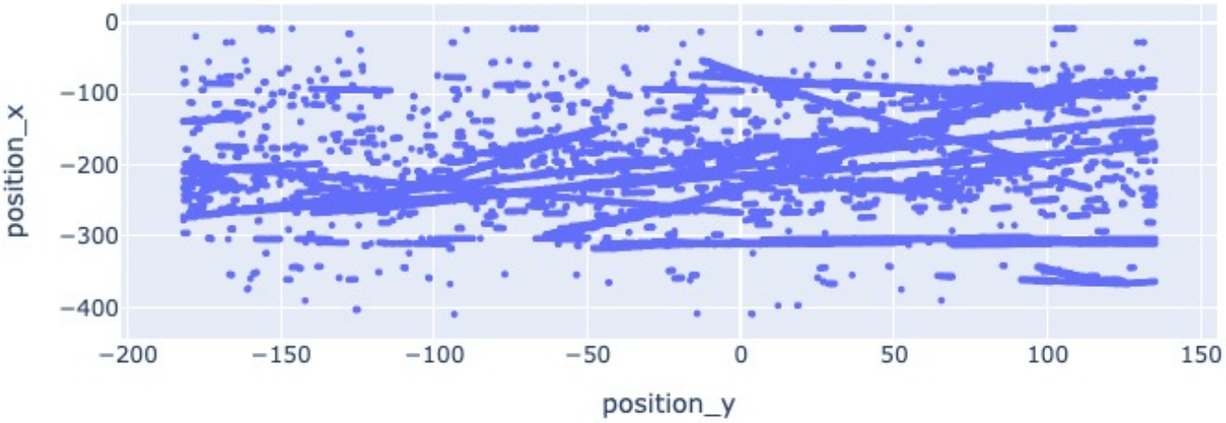
- Many times, hits from only 2 planes were used to make a spacepoint.
  - Spacepoints **constructed** from **less than 3 hits**, one per wire plane, **filtered out**

(C) Filtering spacepoints based off χ^2 value and number of associated hits

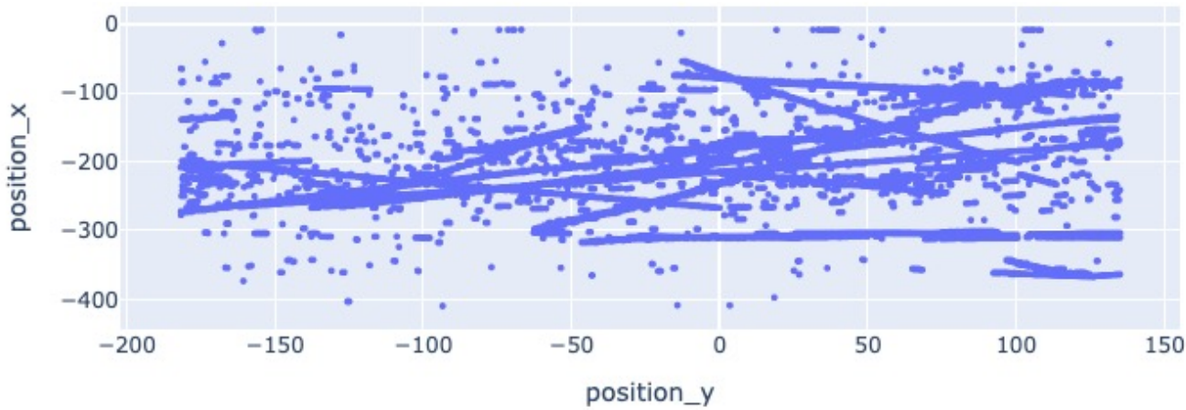
- Combination of filtering methods (A) and (B)

Filtering Spacepoints

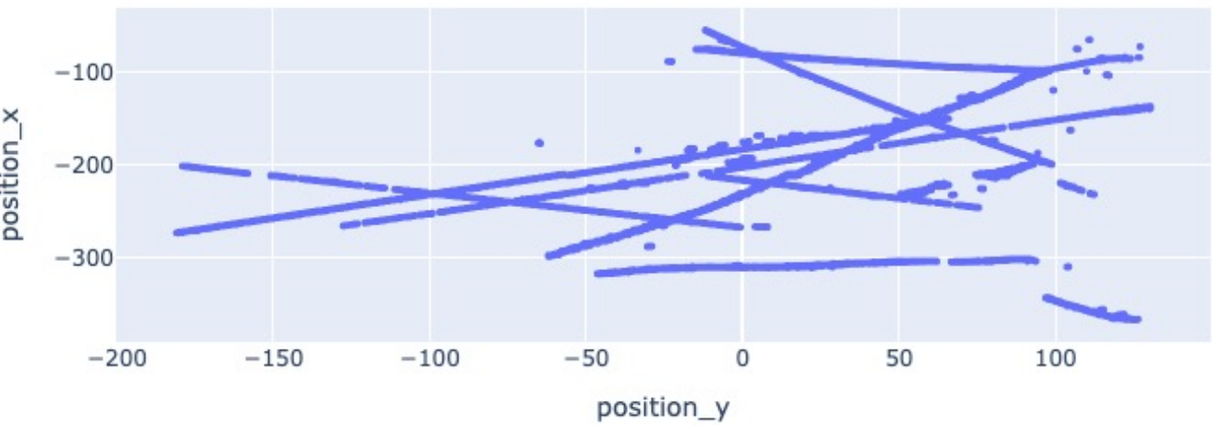
XY projection of spacepoints **without filtering**



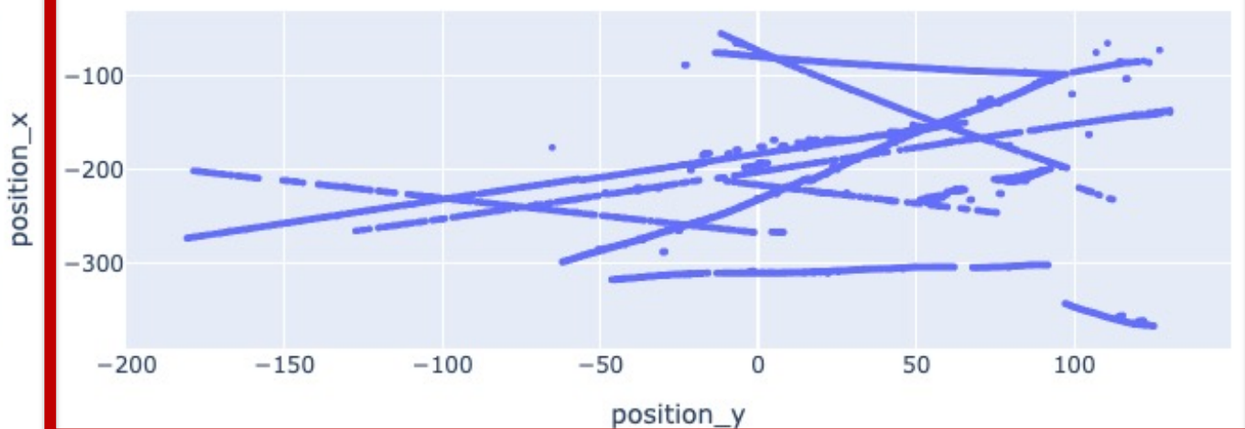
Spacepoints **Removing χ^2>0.5** (A)



Spacepoints **removing** points from **< 3 hits** (B)



Spacepoints **removing** points from **< 3 hits** and **χ^2>0.5** (C)



Fermilab

Neutrino Slices

Credit Ricardo Campos, summer intern

- Each ICARUS event can contain **multiple** candidate **neutrino slices,** Nugraph2 designed for only **one slice** per event
  - Best slice is chosen by the minimum distance between charge and flash barycenters.
  - This further reduces background from cosmics and electric noise

Simulated Event Samples

- MPVMPR sample
- 3 samples:
  - Small(~3,500 events): MPVMPR_MC_v09_89_01_01
  - Medium(~31,000 events): MPVMPR_MC_v09_89_01_01
  - Large(~272,000 events): MPVMPR_MC_FRFIX_v09_89_01_01 (Bug fix)

- Combine run files for each sample to a single hdf5 file to train (ph5concat)
  - Samples made from many run files, each with same run number
  - Run numbers made unique before runs combined

Preprocessing

- Applies χ^2 cutoff of 0.5 and removes rare empty events
  - Could be done during sample generation as well

- Makes graph **edges** between **planar** and **nexus** nodes
- Assigns semantic labels:
  - MIP, HIP, Michel, shower, diffuse

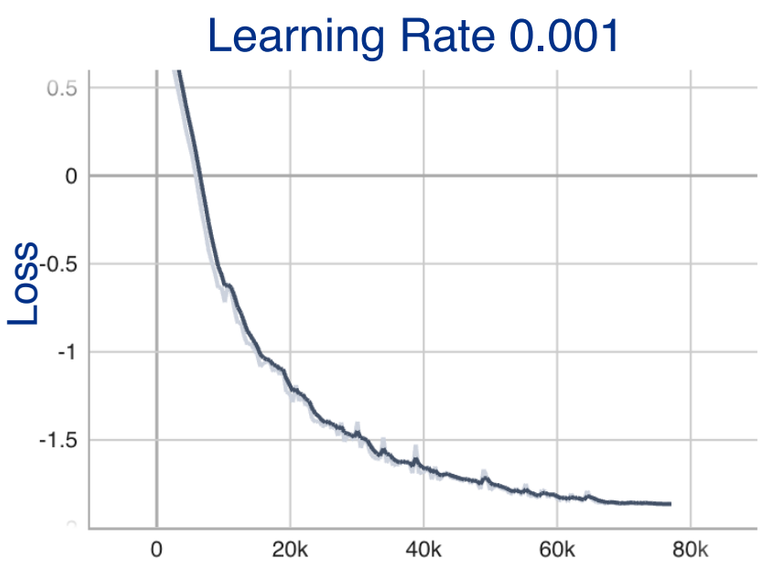
- **Normalizes** each **feature** by Gaussian assumption
- Splits events into **training** (90%), **validation** (5%), and **test** (5%) categories

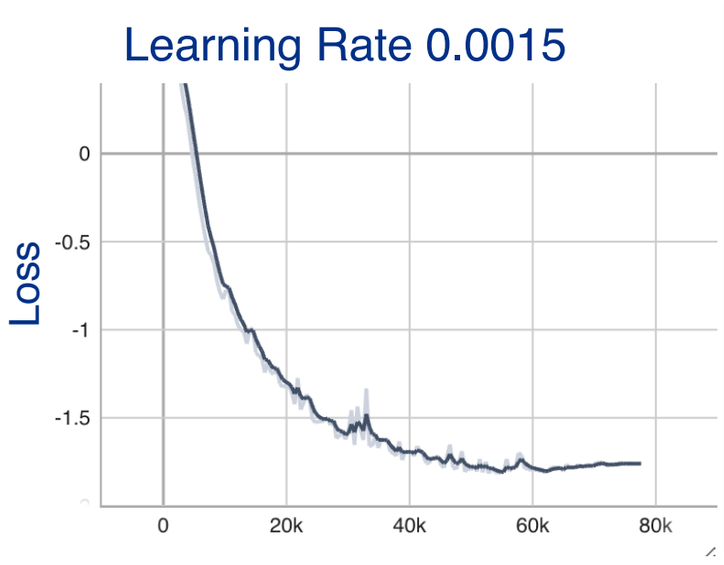
Training

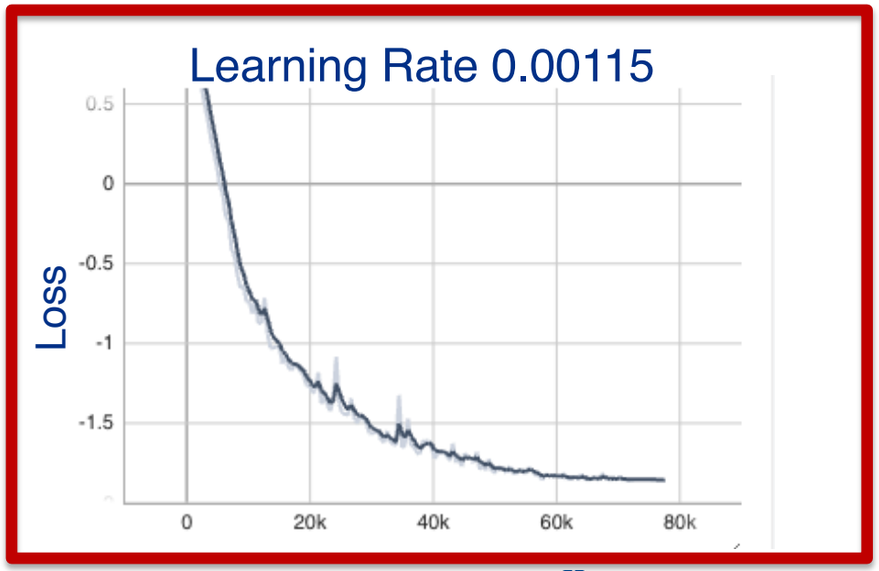
- Events shuffled into equal sized **batches**, training operates on batches in random order
- Compares ML output to their truth, adjusts ML parameters and operates on next batch
  - Repeats for all batches
  - Batch-size: **64 events**
- Number of **epochs** (iterations): 80 (default)
  - Batch order changes each epoch
  - 160 epochs tried with small and medium samples
    - increased training duration (~+50%) with marginal improvements
- Trainings run on 40 GB GPU **EAF server**

Learning-Rate Optimization

- Several learning-rates tried on **medium sample** with **160 epochs** to find optimal rate
  - Default learning-rate **0.001** showed **under-training** in loss plots
  - Learning-rate 0.002, 0.0015, and 0.00125 showed **overtraining**
  - Learning-rate **0.00115** chosen as **optimal rate** for medium sample for now







Learning-Rate Optimization

lr 0.001					
prec_filter/train	0.9745	prec_filter/val	0.973	prec_filter/test	0.9724
recal_filter/train	0.9763	recall_filter/val	0.9748	recal_filter/test	0.9742
prec_sem/train	0.878	prec_sem/val	0.8755	prec_sem/test	0.8679
recall_sem/train	0.8882	recall_sem/val	0.8755	recal_sem/test	0.8679

lr 0.00115					
prec_filter/train	0.9697	prec_filter/val	0.9731	prec_filter/test	0.9724
recal_filter/train	0.9753	recall_filter/val	0.9749	recal_filter/test	0.9743
prec_sem/train	0.8819	prec_sem/val	0.8771	prec_sem/test	0.8698
recall_sem/train	0.8819	recall_sem/val	0.8771	recal_sem/test	0.8698

lr 0.00125					
prec_filter/train	0.972	prec_filter/val	0.9724	prec_filter/test	0.9727
recal_filter/train	0.9766	recall_filter/val	0.9748	recal_filter/test	0.9754
prec_sem/train	0.921	prec_sem/val	0.878	prec_sem/test	0.8774
recall_sem/train	0.921	recall_sem/val	0.878	recal_sem/test	0.8774

lr 0.0015					
prec_filter/train	0.9745	prec_filter/val	0.9731	prec_filter/test	0.9725
recal_filter/train	0.9761	recall_filter/val	0.9747	recal_filter/test	0.9744
prec_sem/train	0.8925	prec_sem/val	0.8754	prec_sem/test	0.8694
recall_sem/train	0.8925	recall_sem/val	0.8754	recal_sem/test	0.8694

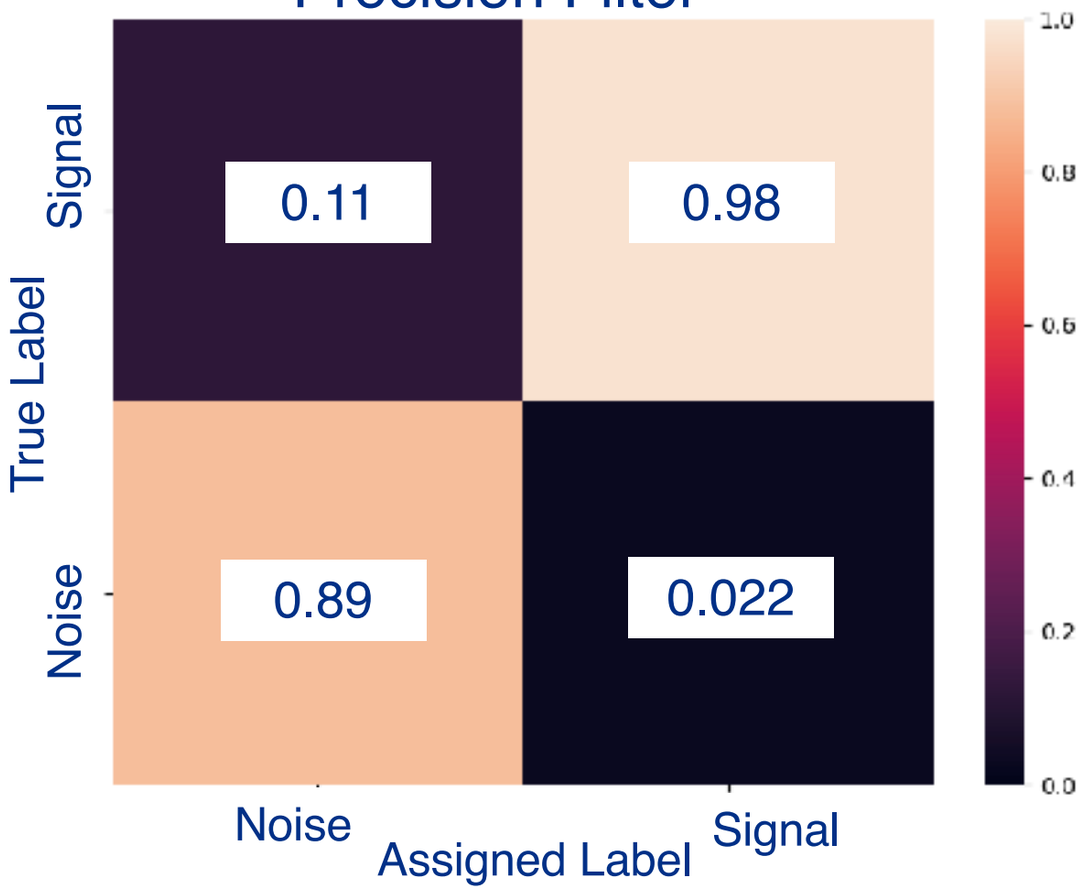
- Learning-rate 0.00125 shows slight overtraining
- Learning-rates 0.001 and 0.0015 show possible bias
- Learning-rate 0.00115 shows overall best results
- Further investigation needed

Results

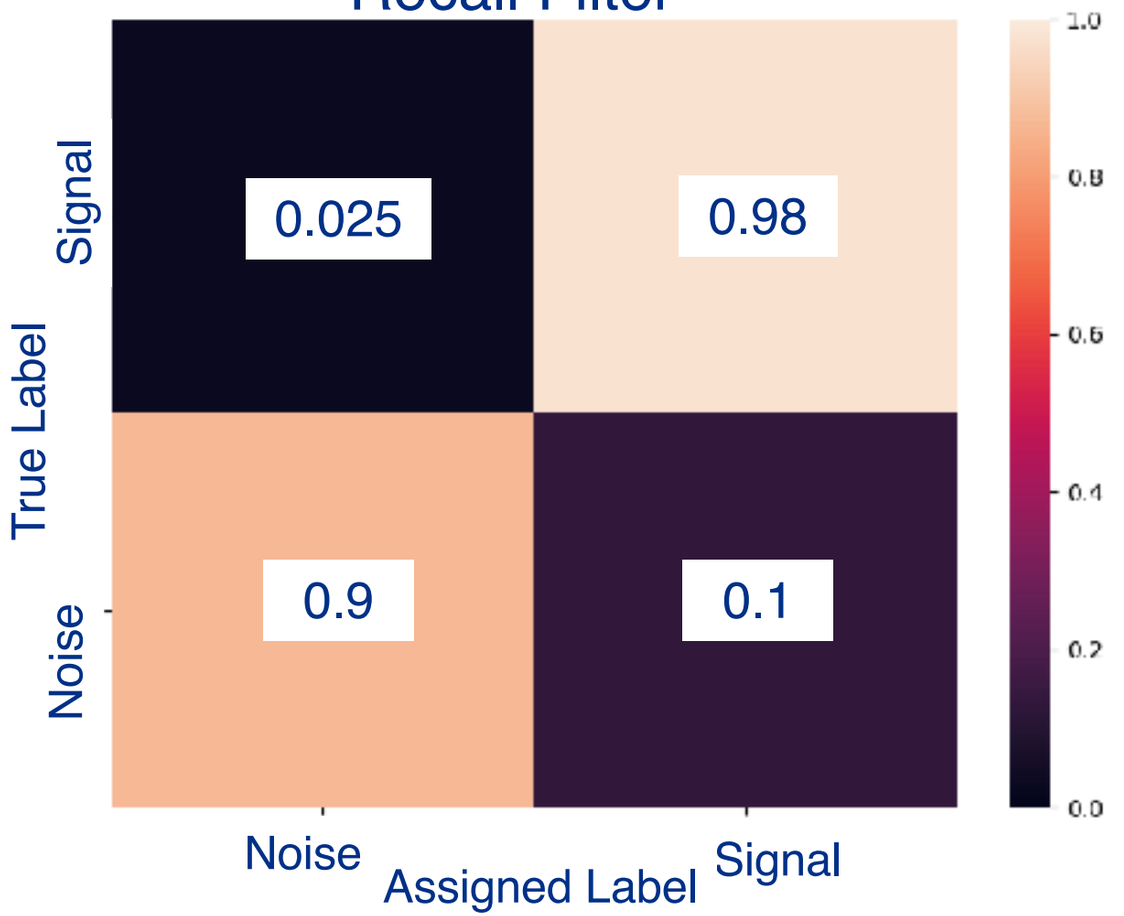
$$\text{Precision(Purity)} = \frac{TP}{TP+FP}$$

$$\text{Recall(Efficiency)} = \frac{TP}{TP+FN}$$

Precision Filter



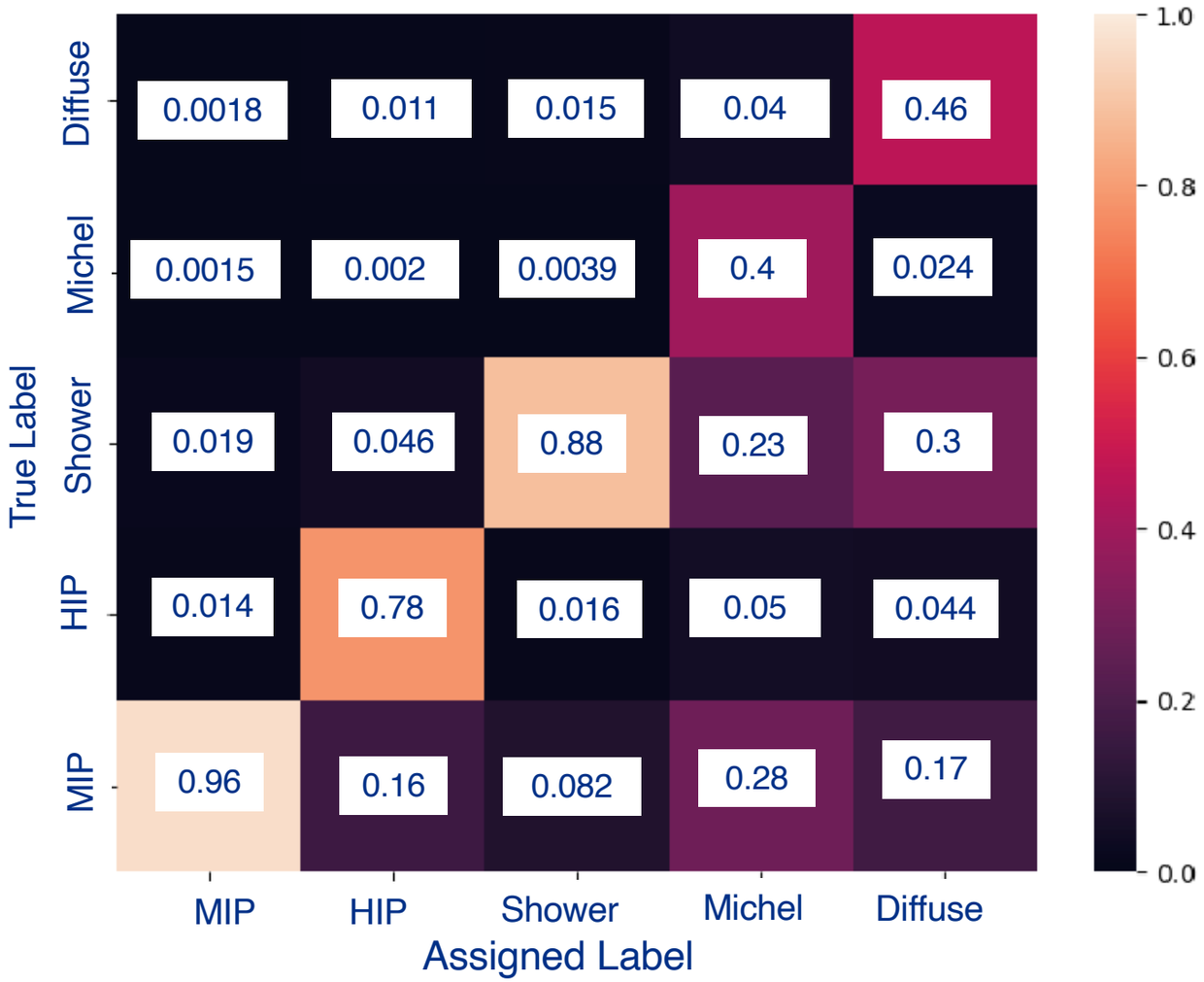
Recall Filter



Fermilab

Results

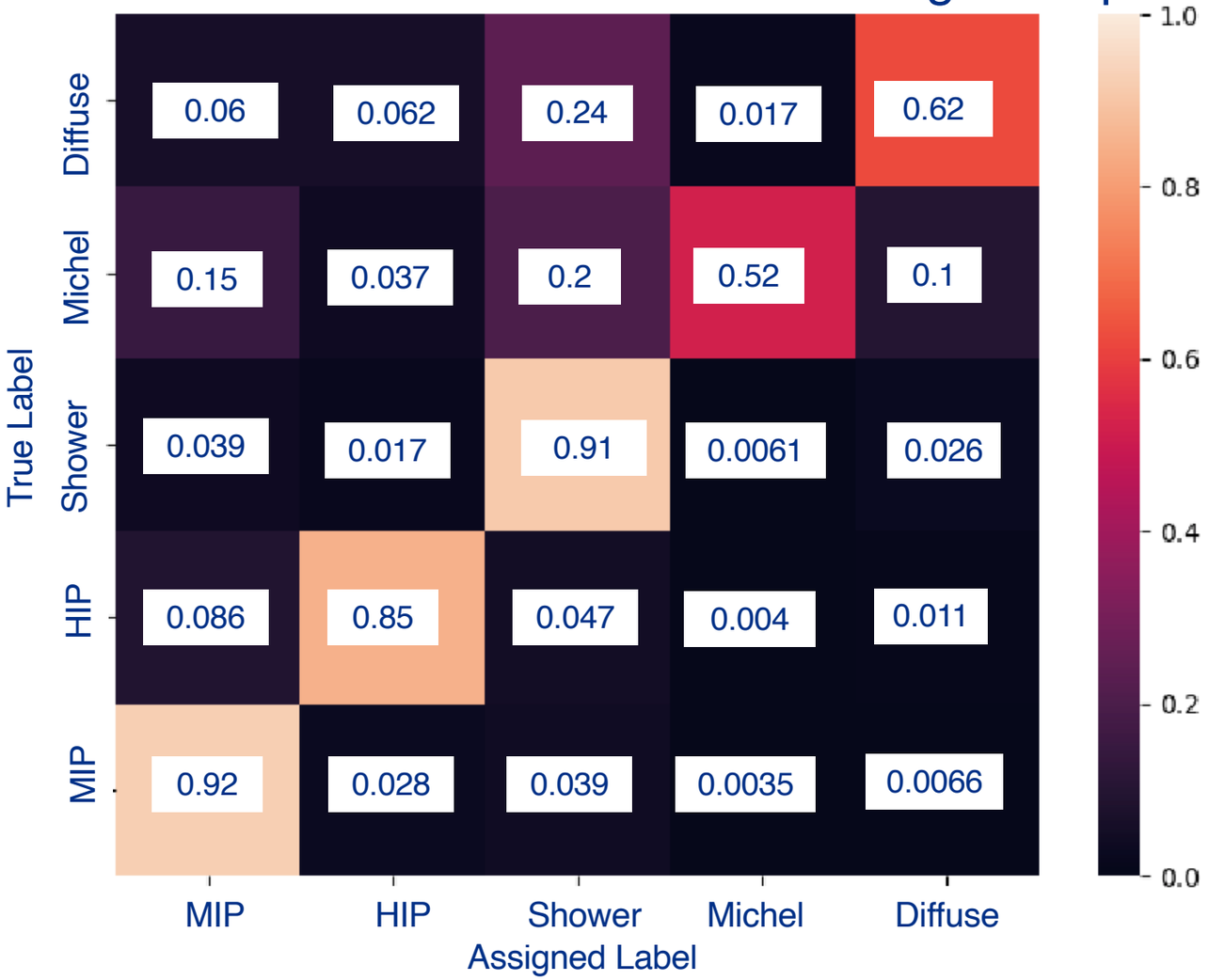
Precision Semantic Confusion Matrix Large Sample



- Precision commonly called Purity
- $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$
- Semantic classes are highly-ionizing particle (HIP), minimum-ionizing particle (MIP), shower, michel, diffuse
- Michel and diffuse have room for improvement

Results

Recall Semantic Confusion Matrix Large sample



- Recall commonly called Efficiency
- $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$
- Semantic classes are Highly-ionizing particle (HIP), minimum-ionizing particle (MIP), shower, michel, diffuse
- Michel and diffuse have room for improvement

Next Steps

- Possible further optimization of hyper parameters
- Integrate NuGraph2 in ICARUS Pandora reconstruction

References

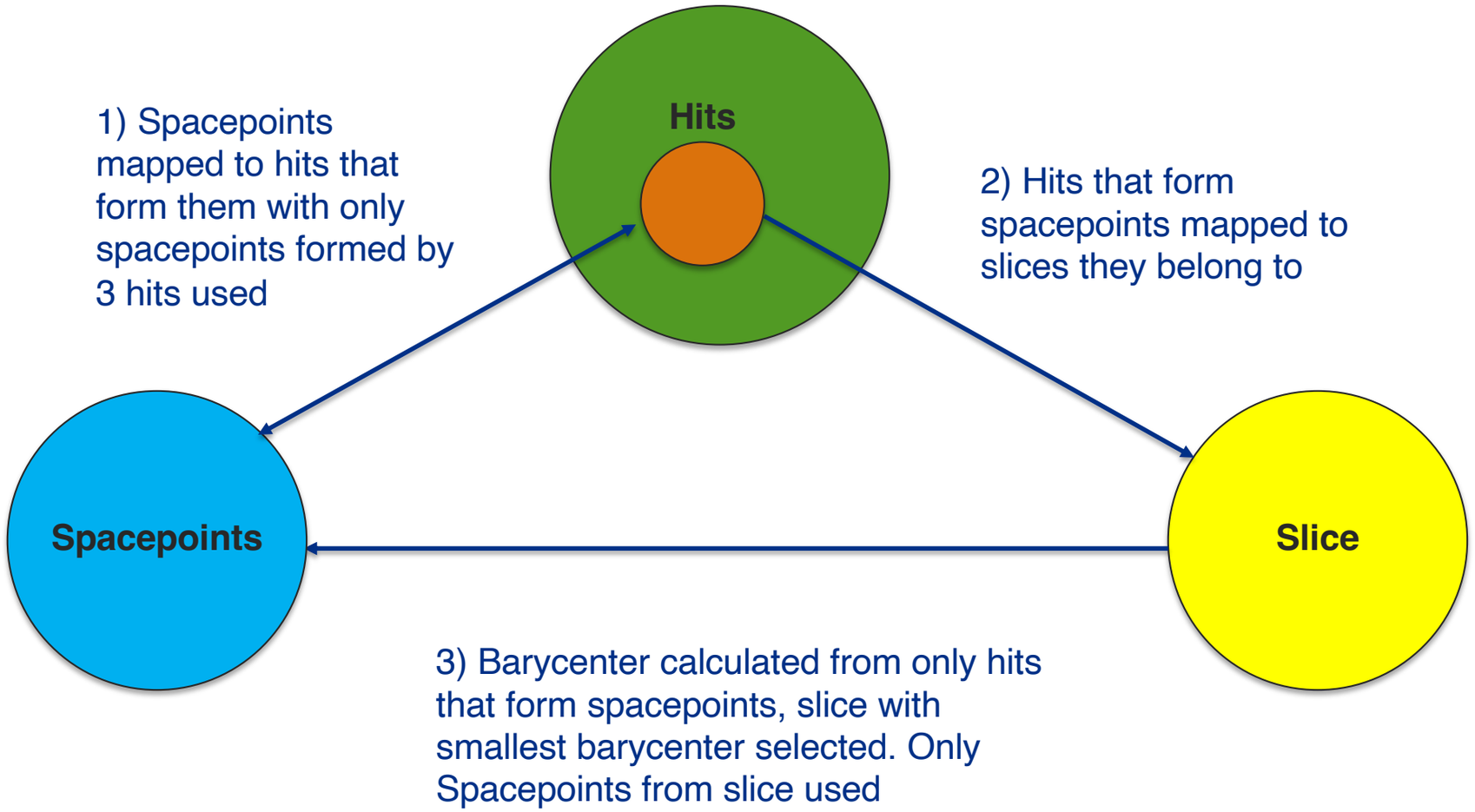
- M. Antonello et al, (2014), *A Proposal for a Three Detector Short-Baseline Neutrino Oscillation Program in the Fermilab Booster Neutrino Beam*, https://arxiv.org/pdf/1503.01520
- P. Abratenko et al, (2023), *ICARUS at the Fermilab Short-Baseline Neutrino Program - Initial Operation*, https://arxiv.org/pdf/2301.08634
- G. Cerati et al, (2024), *NuGraph2: A Graph Neural Network for Neutrino Physics Event Reconstruction*, https://arxiv.org/pdf/2403.11872

Collaborations / Partnerships

Backup slides

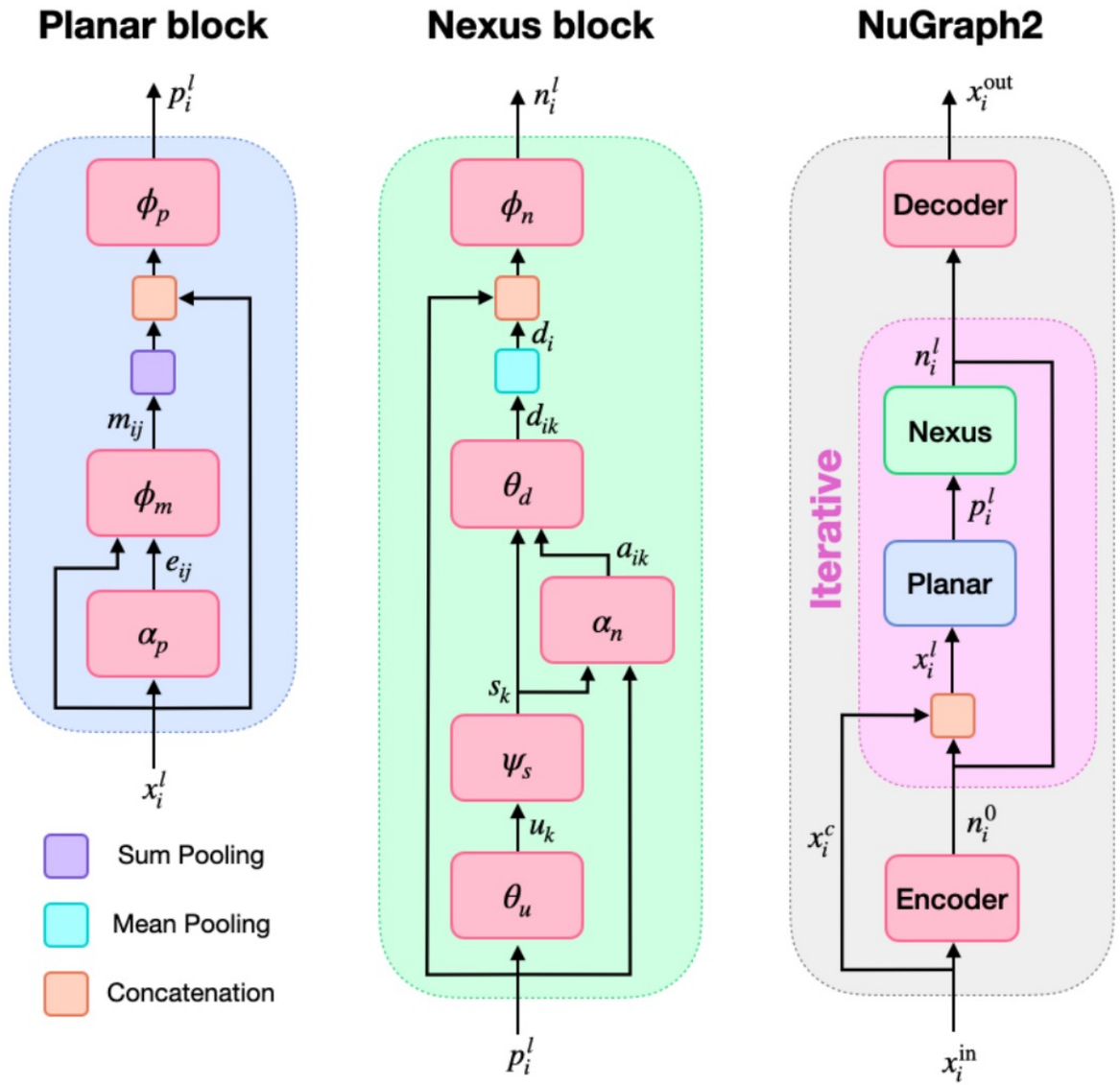
Choosing Best Neutrino Slice

Credit Ricardo Campos, summer intern



Fermilab

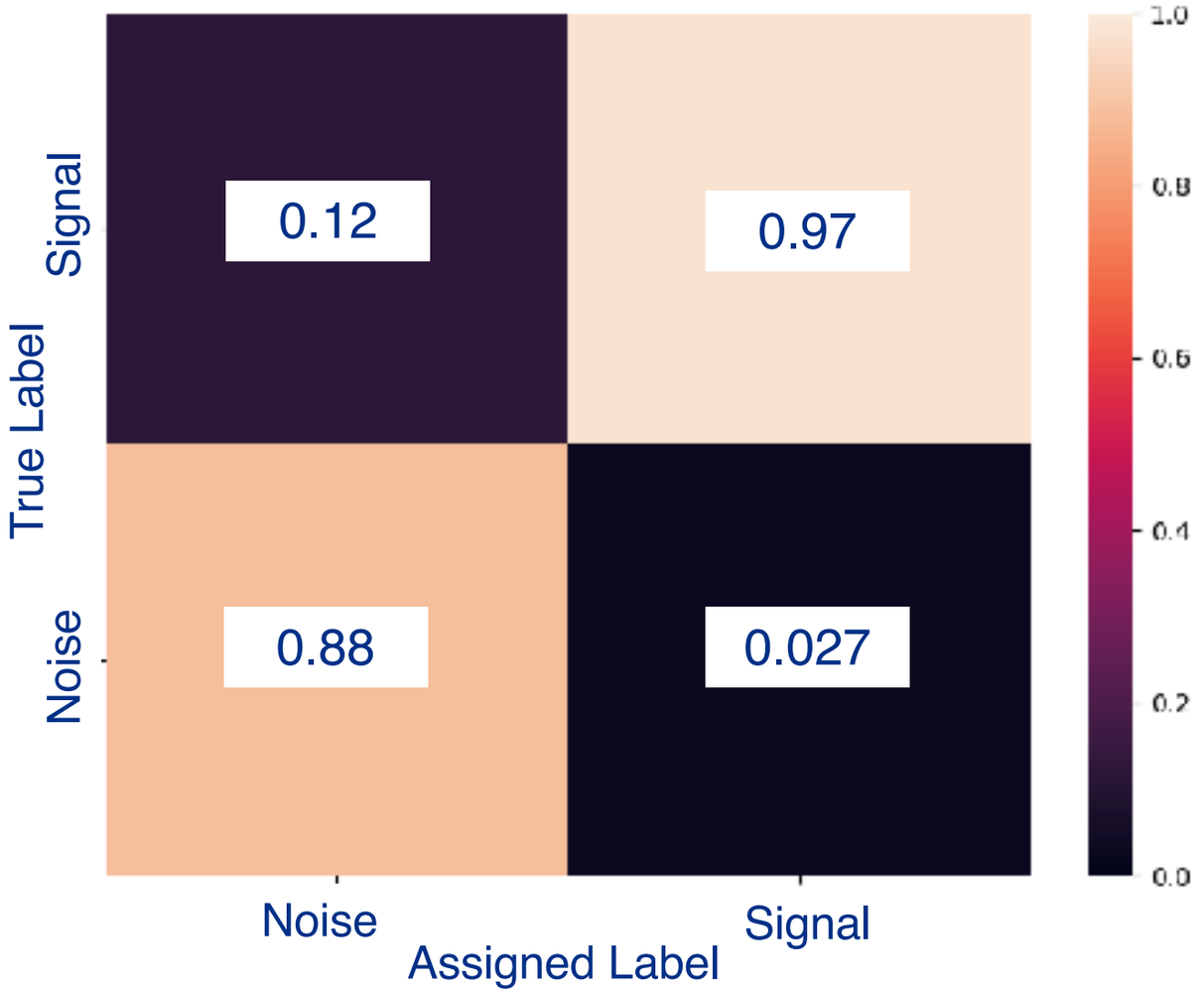
Detailed Diagram of NuGraph2



Fermilab

Results (lr 0.00125)

Precision Filter confusion Matrix Medium Sample

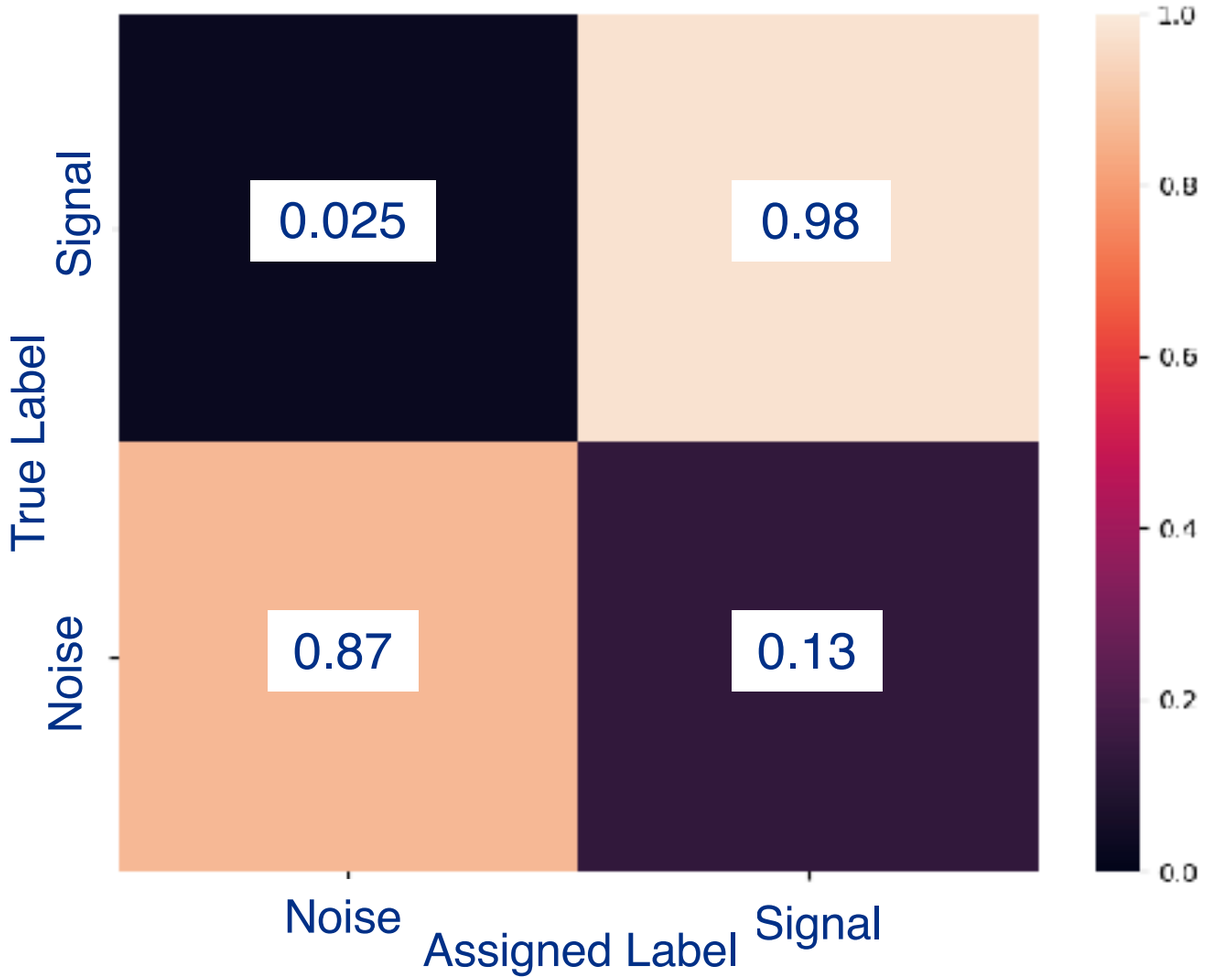


- Precision commonly called Purity
- $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$

Fermilab

Results (lr 0.00125)

Recall Filter Confusion Matrix Medium Sample

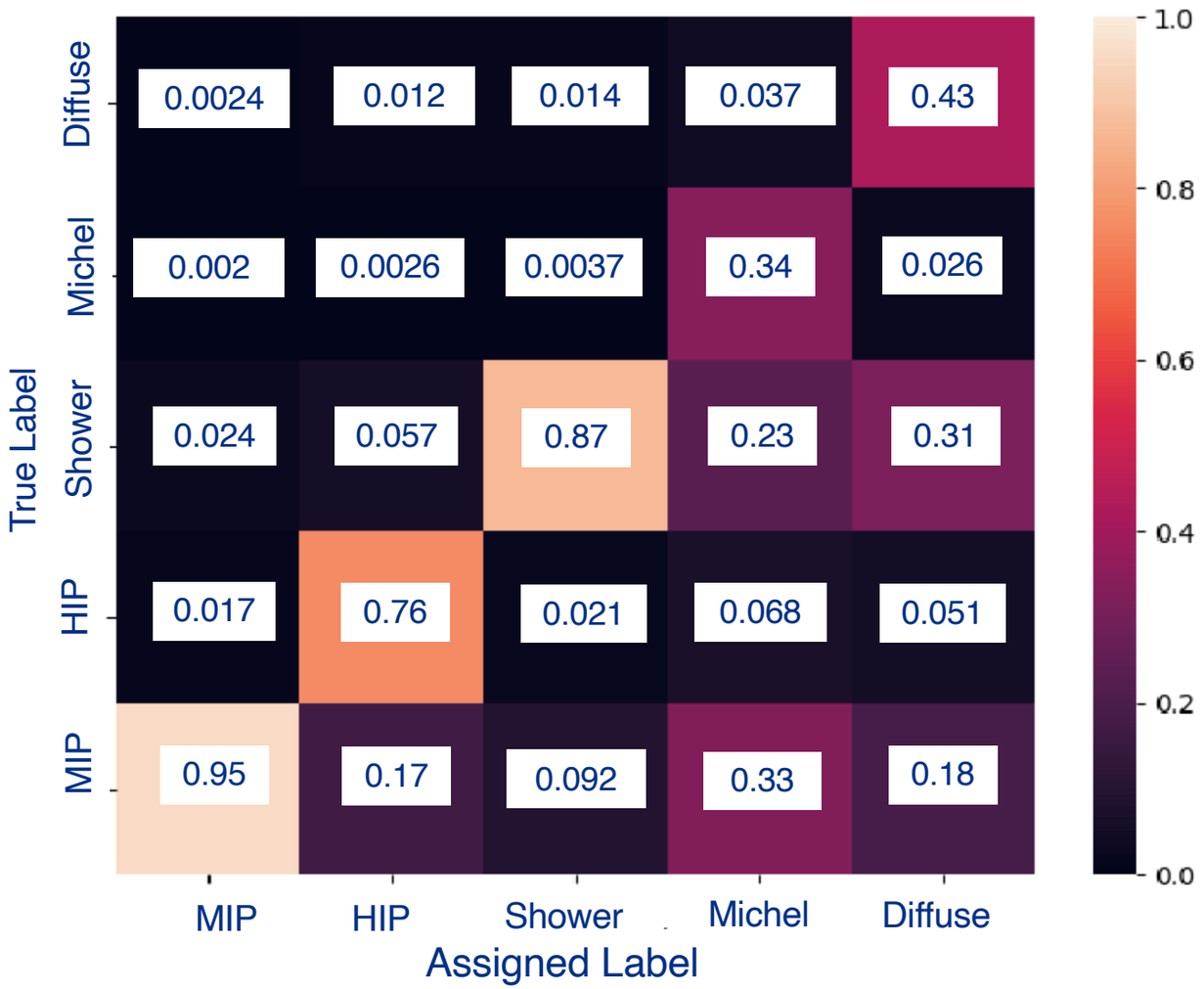


- Recall commonly called Efficiency
- $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$

Fermilab

Results (lr 0.00125)

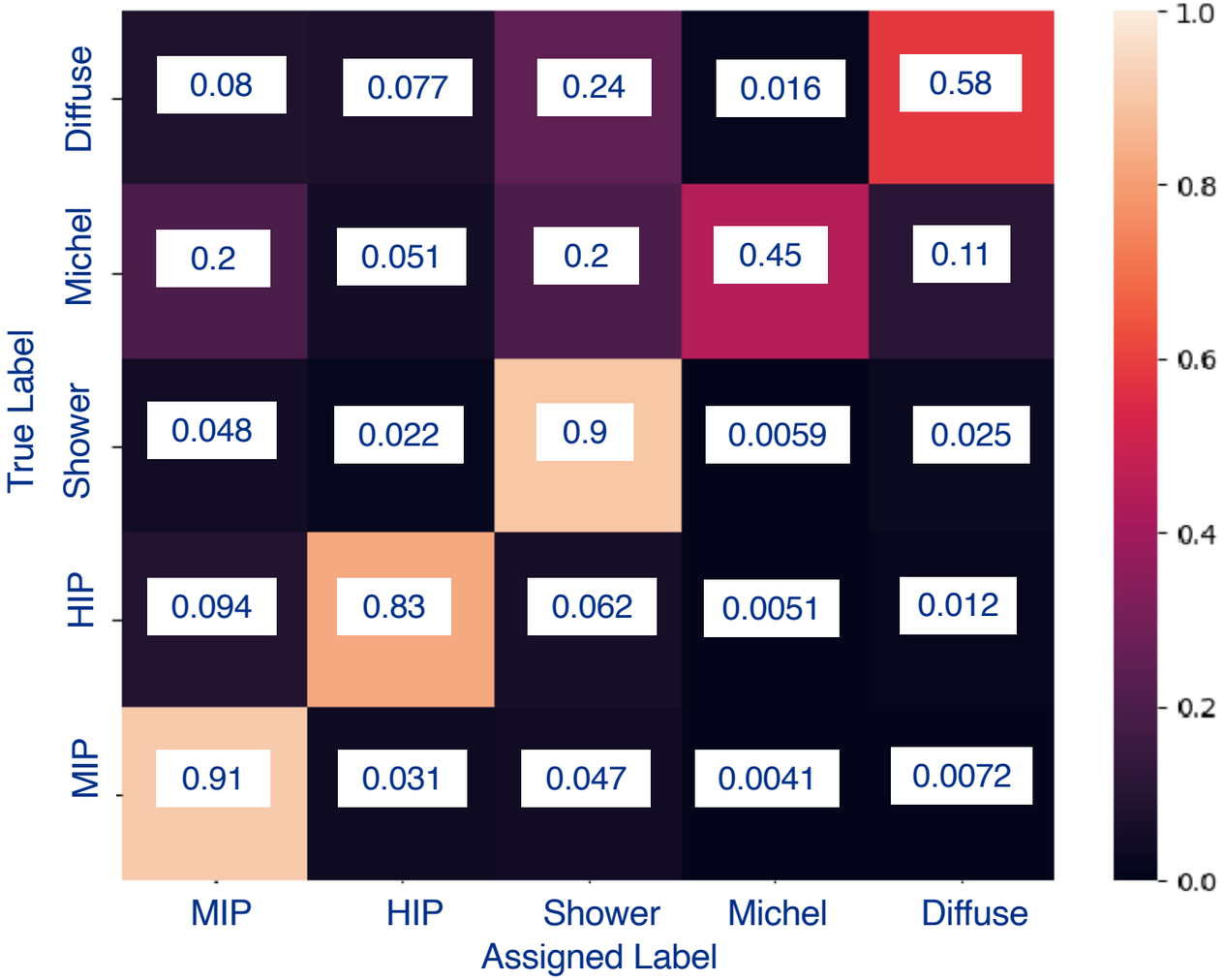
Precision Semantic Confusion Matrix Medium Sample



- Precision commonly called Purity
- $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$
- Semantic classes are high-ionizing particle (HIP), minimum-ionizing particle (MIP), shower, michel, diffuse
- Michel and diffuse have room for improvement

Results (lr 0.00125)

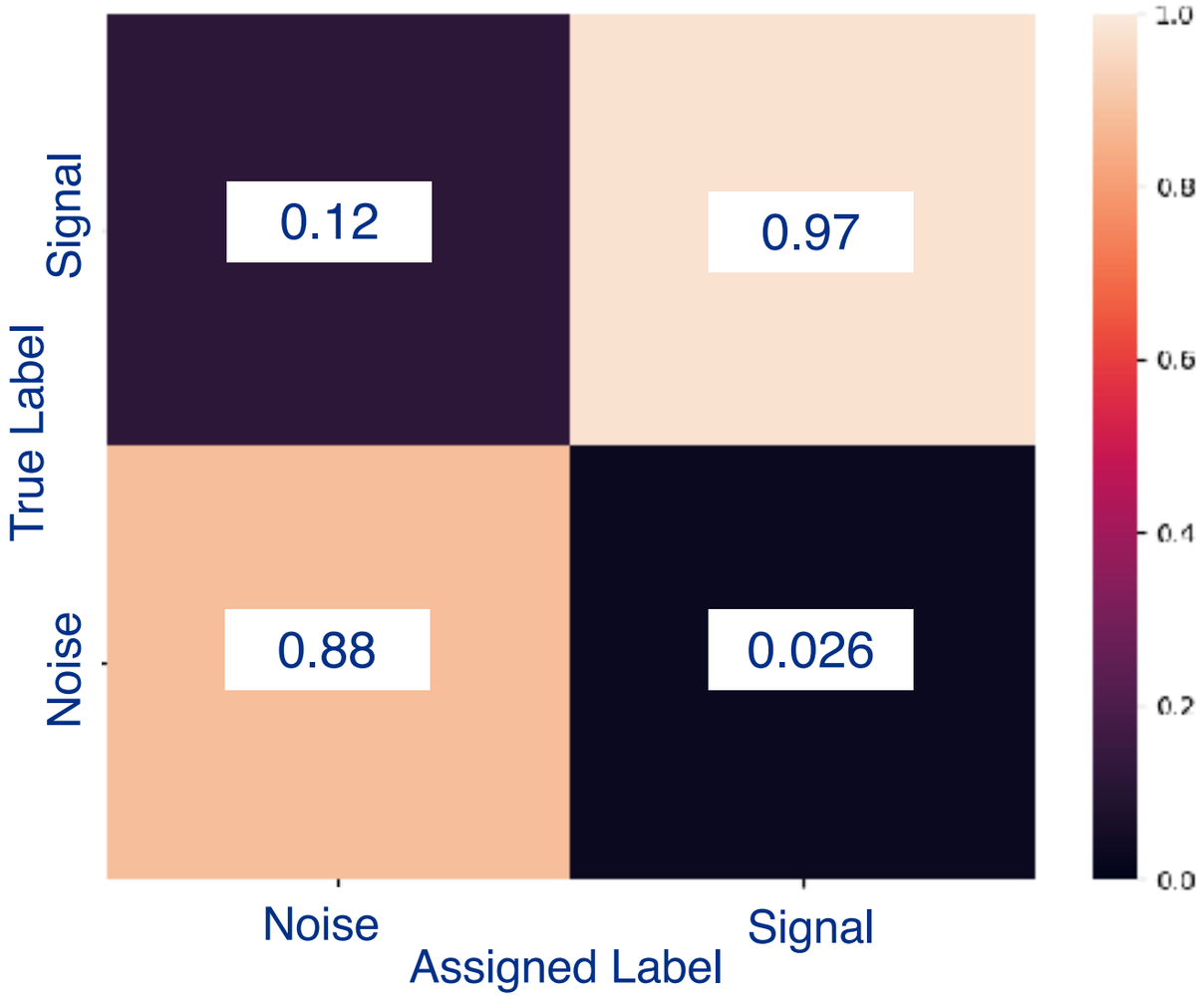
Recall Semantic Confusion Matrix Medium sample



- Recall commonly called Efficiency
- $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$
- Semantic classes are High-ionizing particle (HIP), minimum-ionizing particle (MIP), shower, michel, diffuse
- Michel and diffuse have room for improvement

Results (lr 0.0015)

Precision Filter confusion Matrix Medium Sample

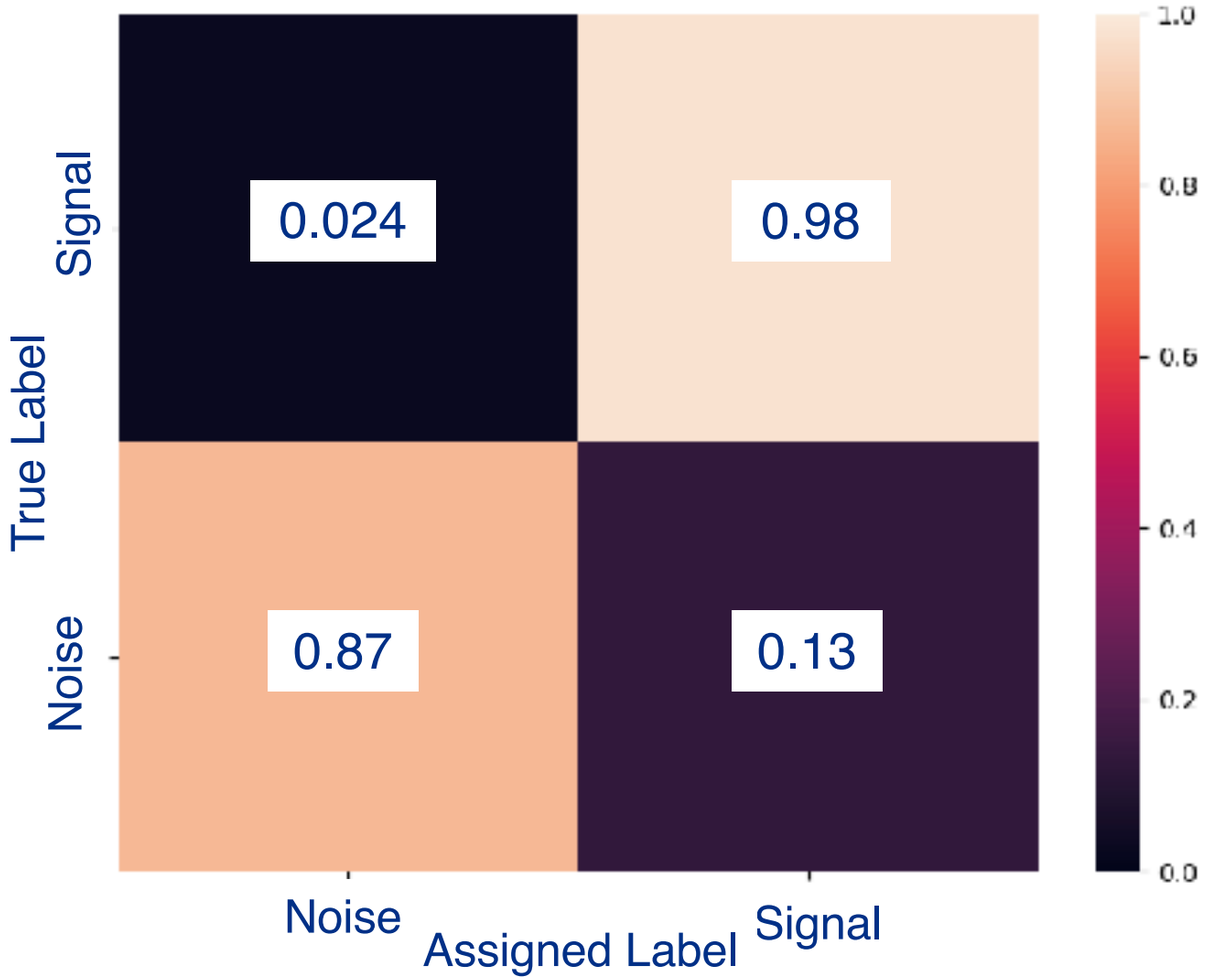


- Precision commonly called Purity
- $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$

Fermilab

Results (lr 0.0015)

Recall Filter Confusion Matrix Medium Sample

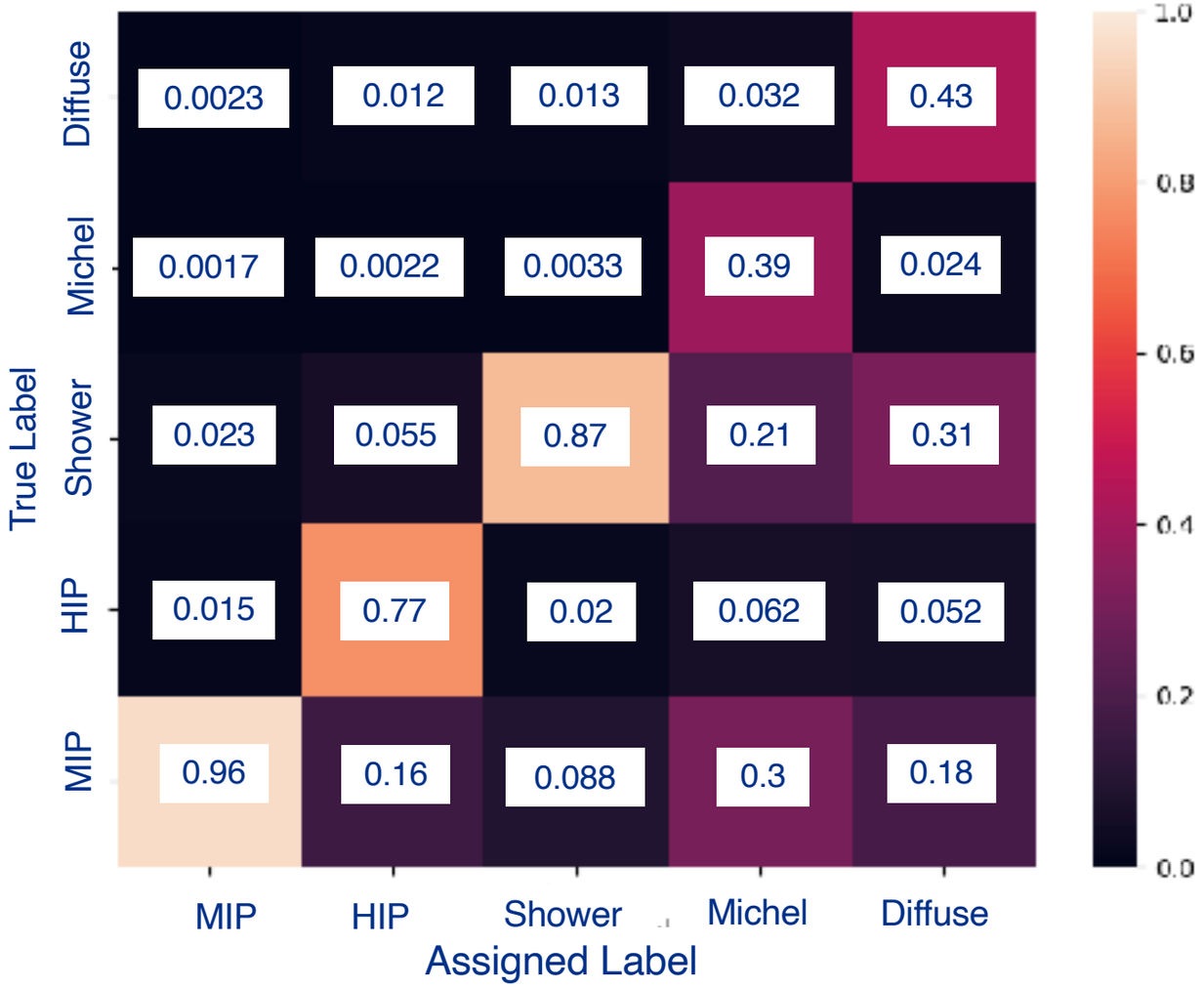


- Recall commonly called Efficiency
- $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$

Fermilab

Results (lr 0.0015)

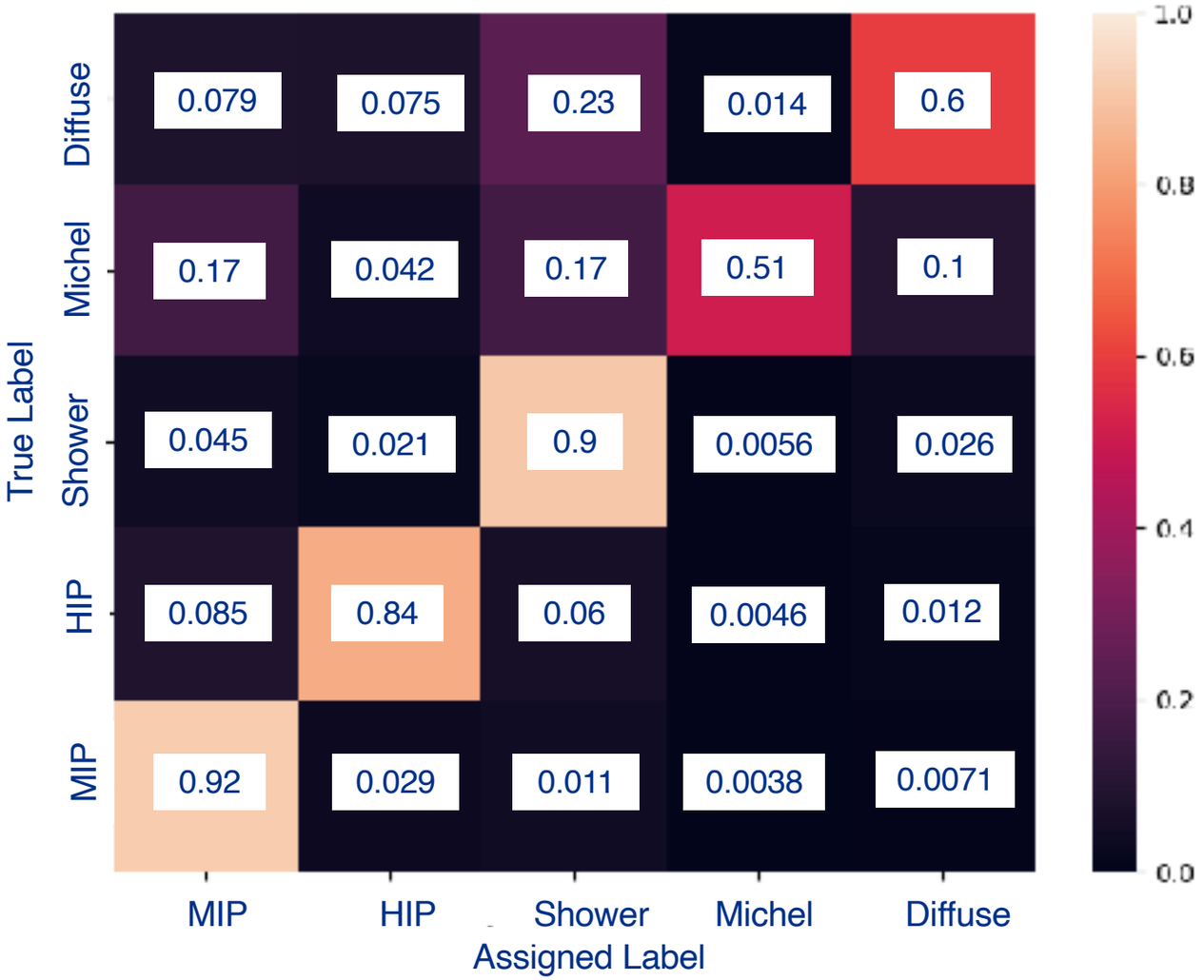
Precision Semantic Confusion Matrix Medium Sample



- Precision commonly called Purity
- $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$
- Semantic classes are high-ionizing particle (HIP), minimum-ionizing particle (MIP), shower, michel, diffuse
- Michel and diffuse have room for improvement

Results (lr 0.0015)

Recall Semantic Confusion Matrix Medium sample



- Recall commonly called Efficiency
- $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$
- Semantic classes are High-ionizing particle (HIP), minimum-ionizing particle (MIP), shower, michel, diffuse
- Michel and diffuse have room for improvement