Machine learning-based particle identification of atmospheric neutrinos in JUNO

Jiaxi Liu[1], Fanrui Zeng[2], Xinhai He[1], Wing Yan Ma[2], Zhen Liu[1], Wuming Luo[1], Hongye Duyang[2], Teng Li[2]

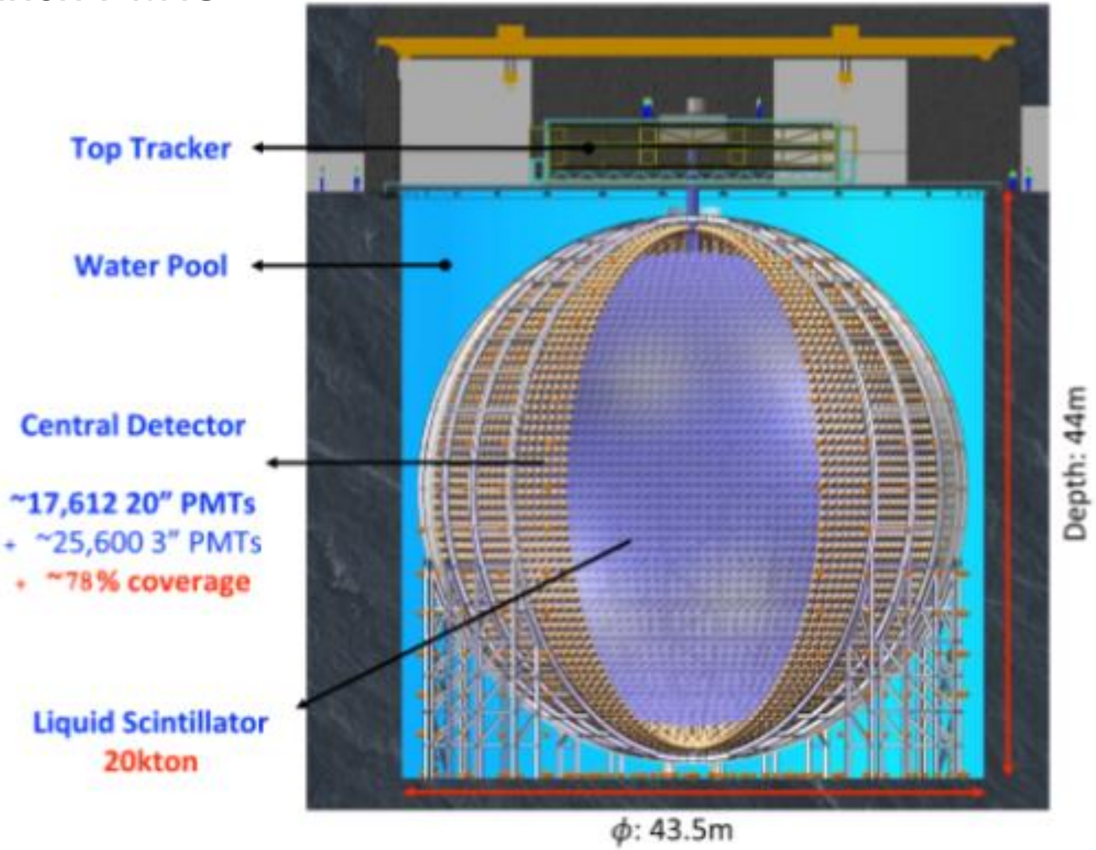
[1] Institute of High Energy Physics, CAS

[2] Shandong University

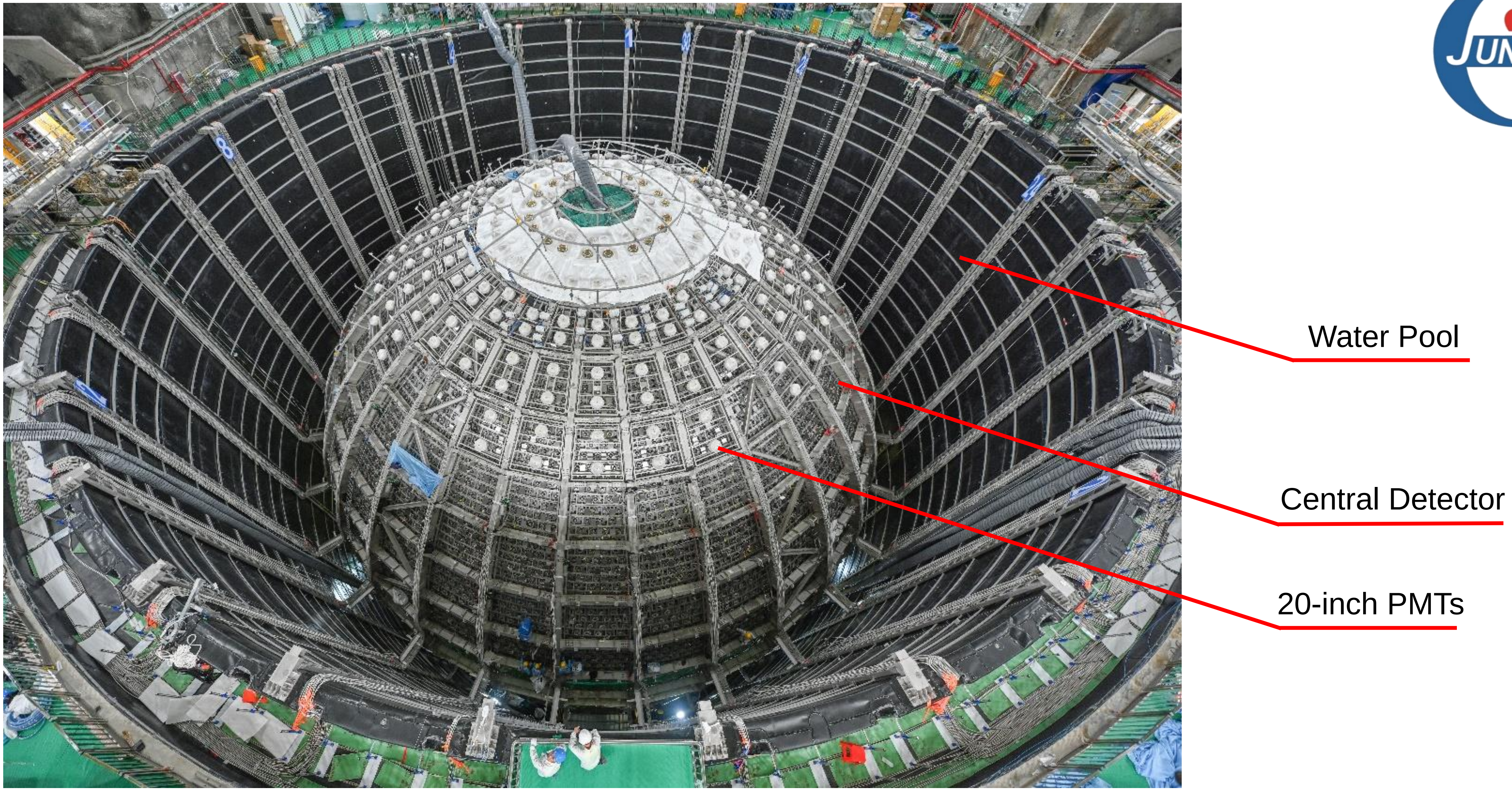
2024.03, ACAT 2024

Jiangmen Underground Neutrino Observatory (JUNO)

- **Main physics goal**: determine the neutrino mass ordering (NMO)
- **Central detector**: 20 kton liquid scintillator, with 17,612 20-inch PMTs arranged on the sphere facing inward.



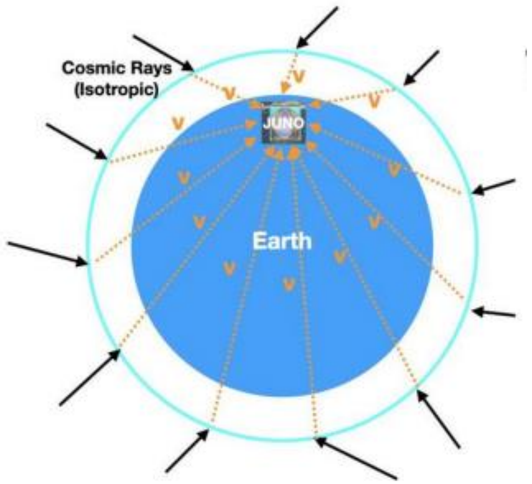
JUNO central detector structure

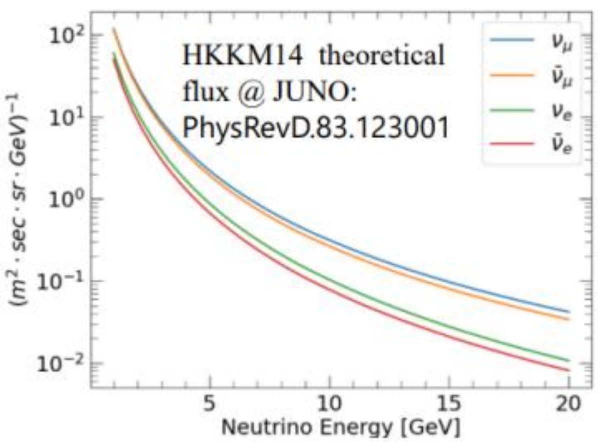


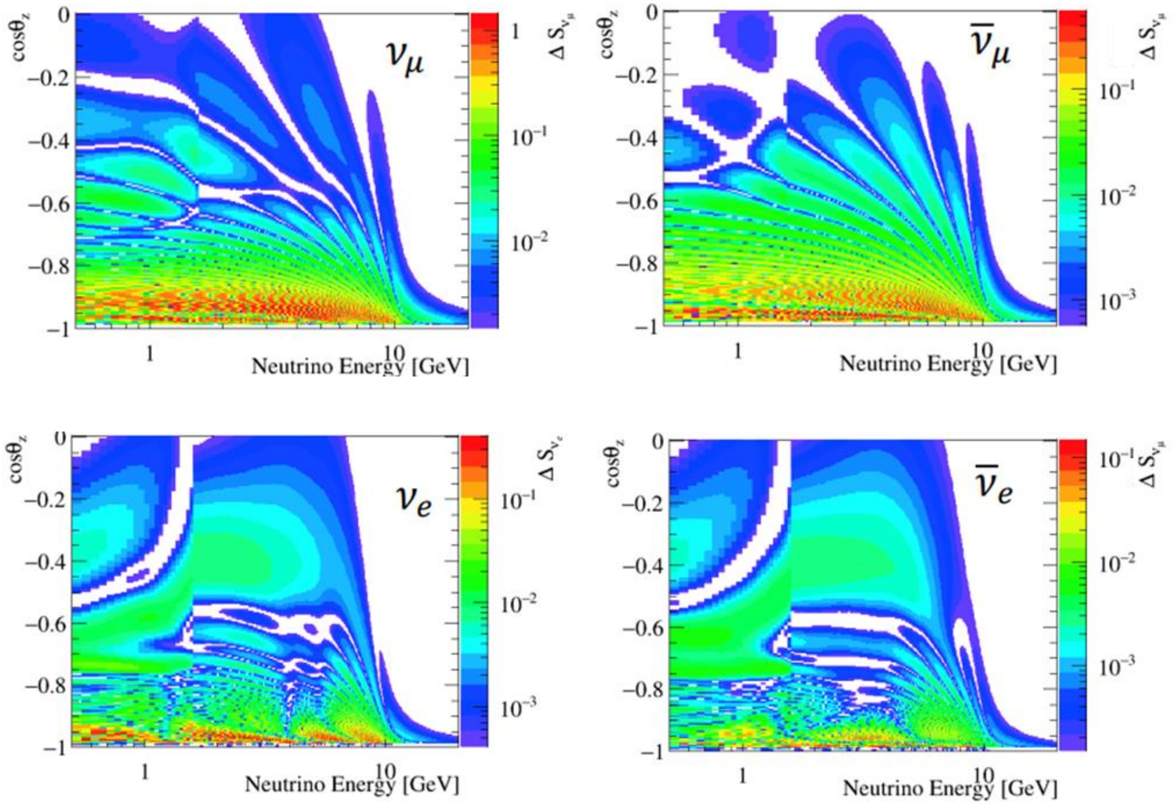
JUNO central detector under construction

Motivation

- Besides reactor neutrinos, atmospheric neutrinos could also provide independent NMO sensitivity.
- Precise particle identification (PID) for atmospheric neutrinos is critical, which could be divided into the following steps:
  - Signal (charged current) VS Background (neutral current)
  - $\nu_\mu/\bar{\nu}_\mu$-CC VS $\nu_e/\bar{\nu}_e$-CC
  - **ν-CC VS $\bar{\nu}$-CC**



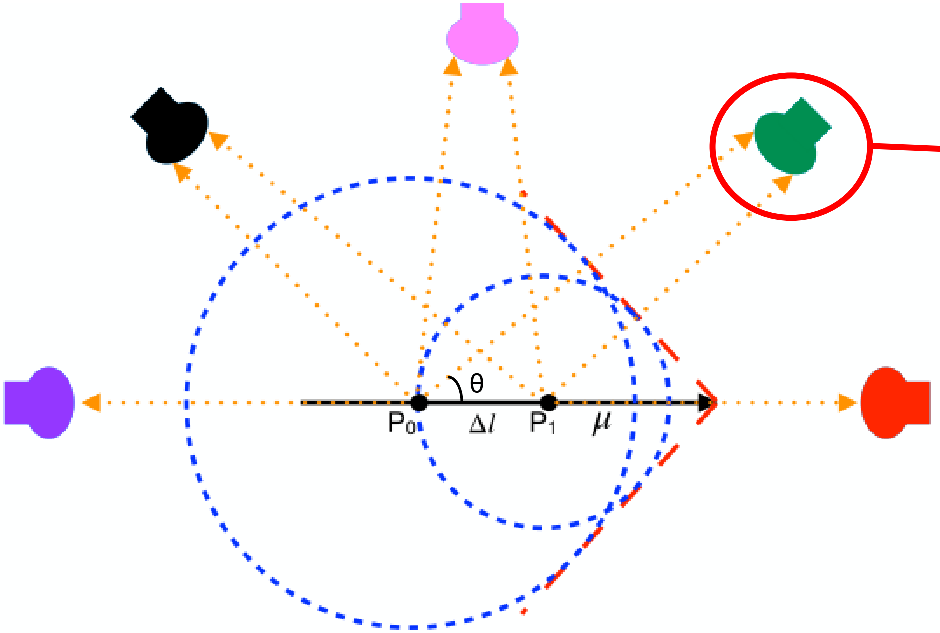




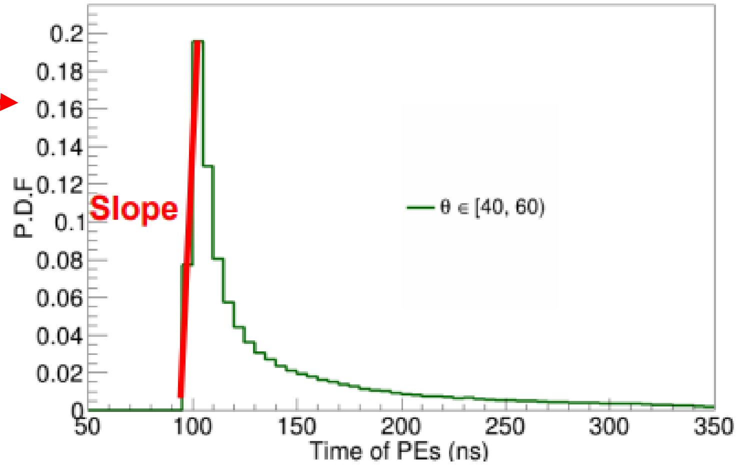
Differences in oscillation probabilities between two neutrino mass order (normal order & inverted order)

Methodology

- For liquid scintillator detector, The light seen by a PMT is a superposition of scintillation light generated at many points on the particle track inside the detector.
- The amount of light received by a PMT evolves as a function of time (PMT waveform) depends upon track direction, interaction vertex, visible energy, and dE/dx (particle type).



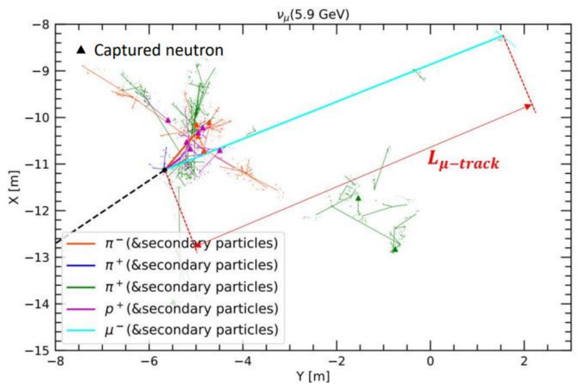
scintillation light from a charged particle track reaching each PMT

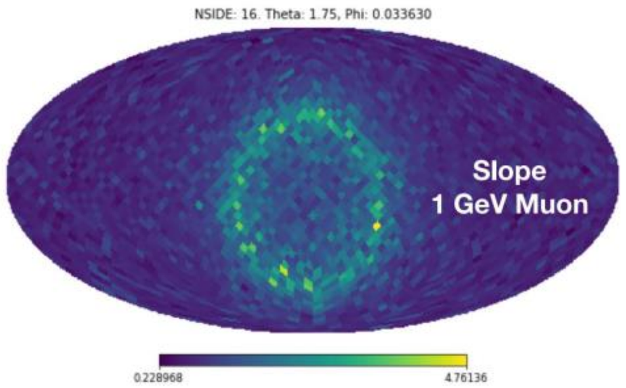


For a PMT with an angle exactly perpendicular to the scintillation-light front, the slope of the rising edge of its waveform will be particularly steep.

Methodology

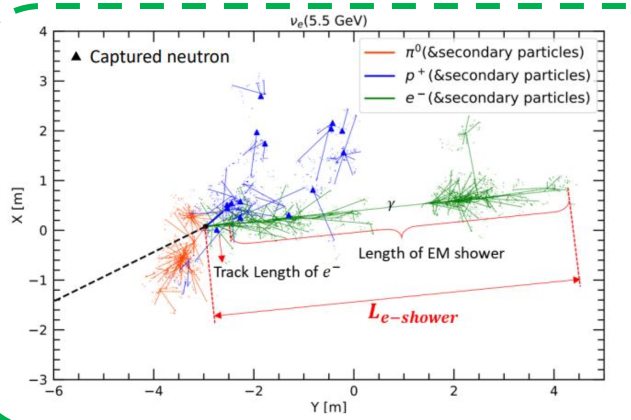
- For neutral current (NC) events, no charged lepton is produced by the primary interaction;
- For charged current (CC) events, the characteristics of all PMT waveforms vary depending on the flavor of the charged lepton (μ or e):

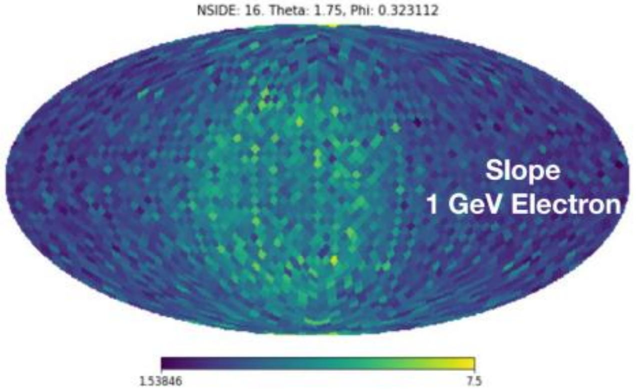




$\nu_\mu/\bar{\nu}_\mu$-CC events:

- Charged lepton (μ^+/μ^-) deposits energy with ionization, leaving a long track in LS;
- The ring of slope is sharp due to muon's straight track.



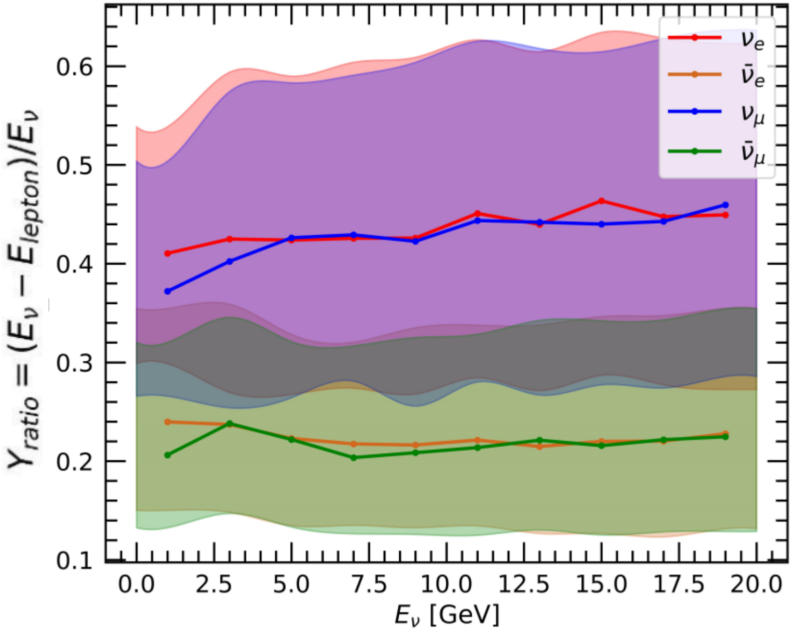


$\nu_e/\bar{\nu}_e$-CC events:

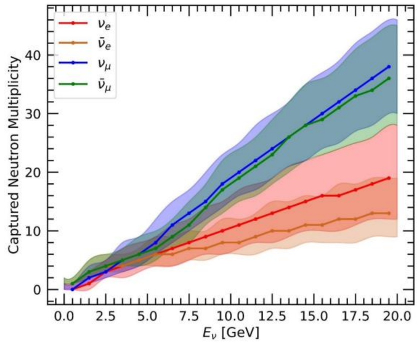
- Charged lepton (e^+/e^-) induces EM shower to produce secondary electrons along the track;
- The ring of slope is fainter than $\nu_\mu/\bar{\nu}_\mu$-CC events due to the EM shower and shorter track.

Methodology

- The charged lepton from primary trigger could provide enough information for 3-labels PID ($\nu_\mu/\bar{\nu}_\mu$-CC VS $\nu_e/\bar{\nu}_e$-CC VS NC).
- However, for the $\boldsymbol{\nu/\bar{\nu}}$ discrimination task, the difference in hadronic energy fraction $Y_{ratio} = (E_\nu - E_{lep})/E_\nu$ is fundamental, which could be reflected by extra information from captured neutrons and Michel-electrons in the secondary triggers:

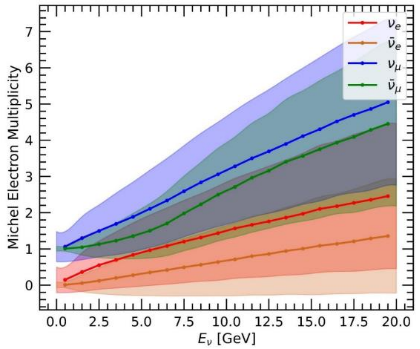


Y_{ratio} distribution for each CC flavor
(Figure courtesy of Xinhai He)



Captured neutron multiplicity

Captured neutron:
- Carries the information of hadronic energy and the directionality of hadrons and neutrinos.
- Provides more capability to $\nu_e/\bar{\nu}_e$ discrimination.
- High tagging efficiency, easier to reconstruct

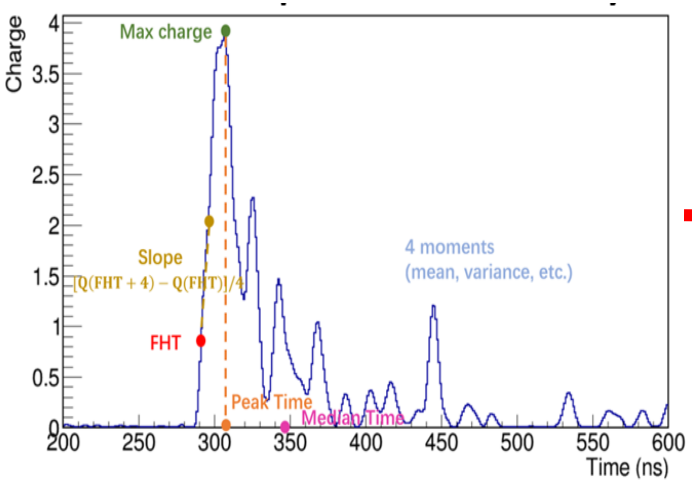


Michel-electron multiplicity

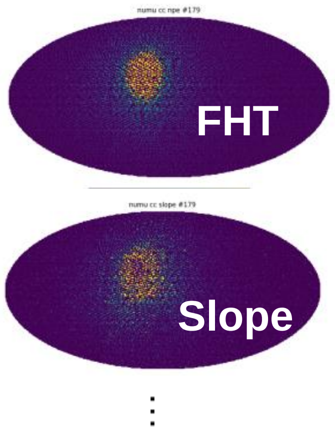
Michel-electron:
- Carries the information of final-state charged leptons and pions.
- Provides more capability to $\nu_\mu/\bar{\nu}_\mu$ discrimination.
- Harder to reconstruct comparing to neutron

A multi-purpose Machine Learning Solution

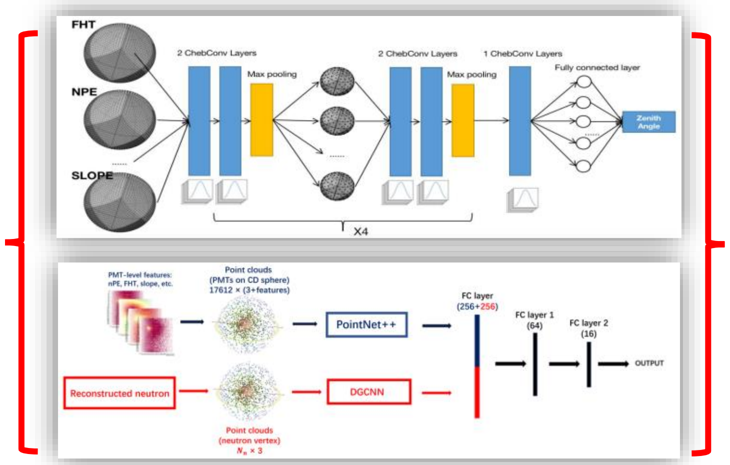
- Due to the large PMT number distributed on the sphere, directly feeding models with all waveforms is rather hard;
- Instead, a few characteristic features that reflect event topology in the detector are extracted from the PMT waveforms:



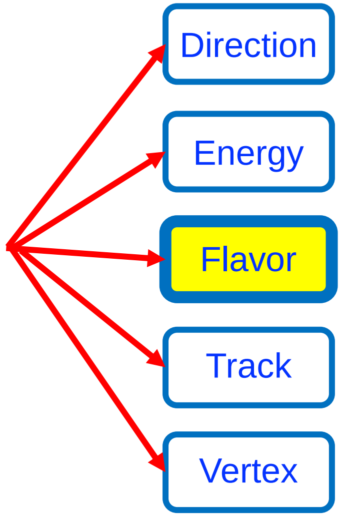
PMT Waveforms (after deconvolution and noise-removing)



Pictures of PMT Features



Machine Learning Models (DeepSphere, PointNet++, …)



Outputs

Features extracted from waveforms:

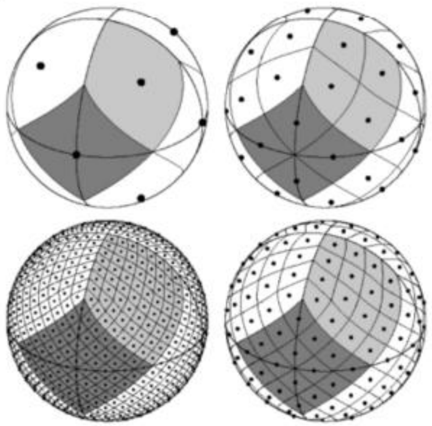
- **FHT**: First Hit Time
- **Slope**: Describes the average slope in the first 4ns.
- **Peak charge and peak time**: the charge and time of the peak of the waveform
- **Charge**: The total number of PEs

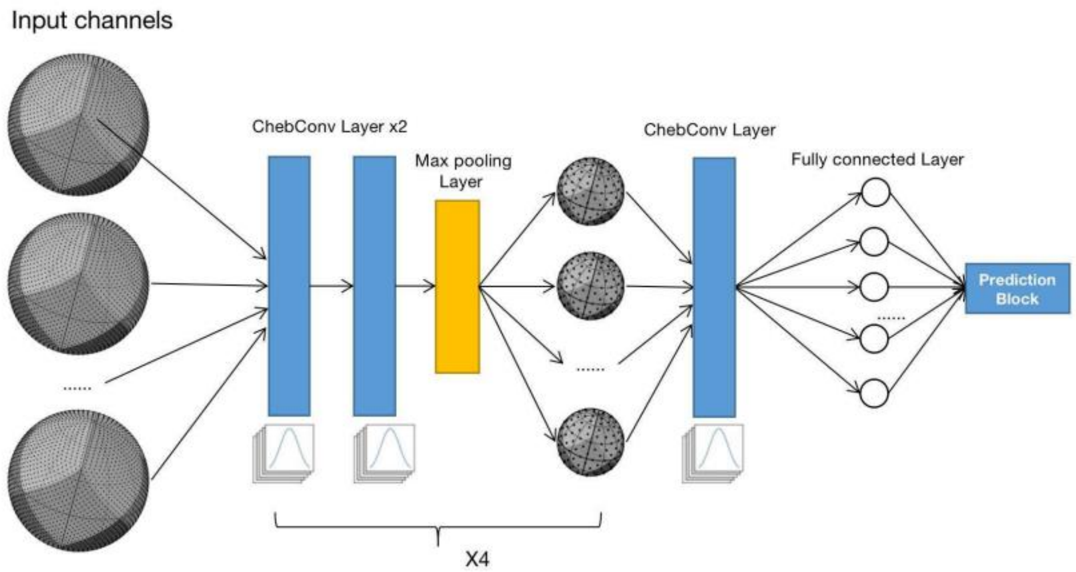
⋮

Machine Learning Models & Strategies

To process the 3-dimensional input of features from PMTs on a sphere, two strategies with different categories of deep learning models were employed:

1. Spherical image-based model: DeepSphere
   - ➢ A popular tool processing spherical data originally developed for cosmology studies;
   - ➢ Maintain rotation covariance;
   - ➢ Avoid distortions caused by projection to a planar surface.





Architecture of DeepSphere

- ➢ Nside = 32
- ➢ Pixels=12 × Nside ^{2} = 12288
- ➢ If more than one PMTs are grouped into one pixel, information is merged:
  - FHT: the earliest;
  - nPE: the sum;
  - Slope and other: the average.

Machine Learning Models & Strategies

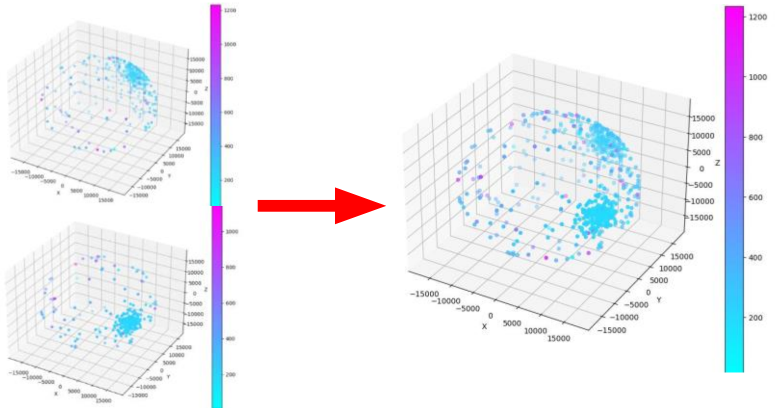
DeepSphere model strategy:

Atm. ν candidate → Primary trigger → PMT features (FHT, Slope,…)

Atm. ν candidate → Secondary triggers → Michel electron & neutron selection

- Michel e trigger 1 → PMT feature (FHT)
- Michel e trigger 2 → PMT feature (FHT)
- Michel e trigger 3 → PMT feature (FHT)
- neutron trigger 1 … neutron trigger n → Merged PMT features (FHT, nPE)

All PMT features → Machine learning model (DeepSphere) → PID label: $\nu_\mu/\bar{\nu}_\mu/\nu_e/\bar{\nu}_e/NC$



Merging features for neutron candidate triggers

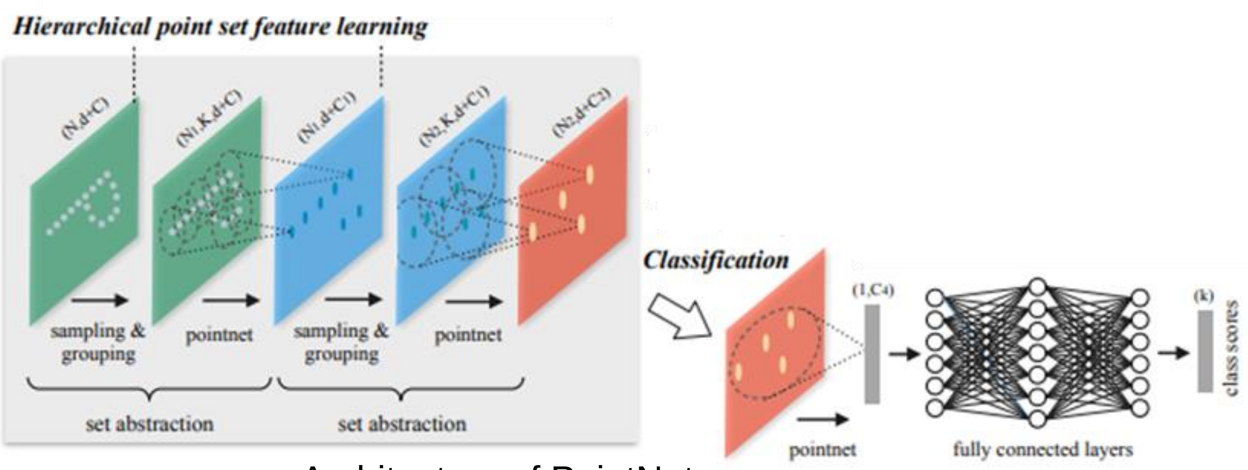
Advantages: All features are at the same PMT-level, fast and easy for the ML model to handle the input.

Machine Learning Models & Strategies

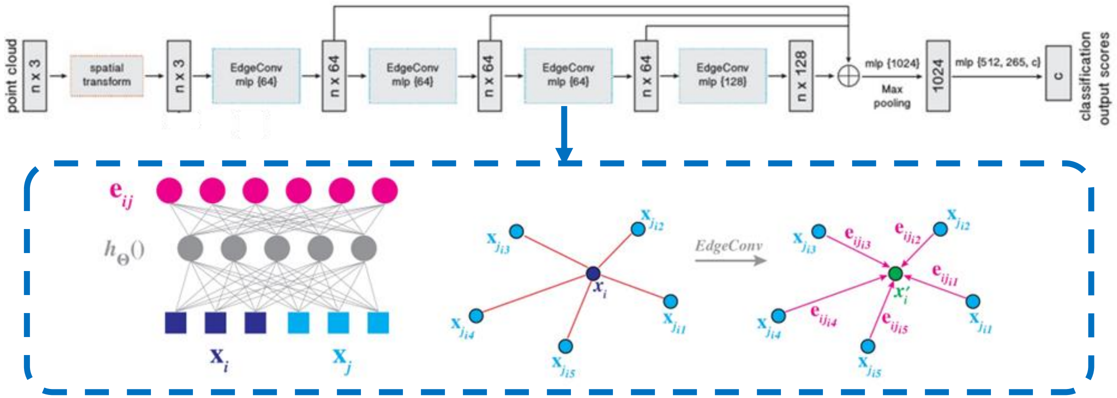
To process the 3-dimensional input of features from PMTs on a sphere, two strategies with different categories of deep learning models were employed:

2. Point cloud-based model: PointNet++, DGCNN

   - Directly taking 3D point clouds (N_{points} × [x, y, z, features...]) as input; JUNO signal more resemble point clouds, minimize the information loss during projection.
   - **PointNet++**: strong capability to handle complex point clouds with set abstraction, used for PMT-level features.
   - **DGCNN**: edge-based model with better performance for sparse point clouds, used for reconstructed neutron features.



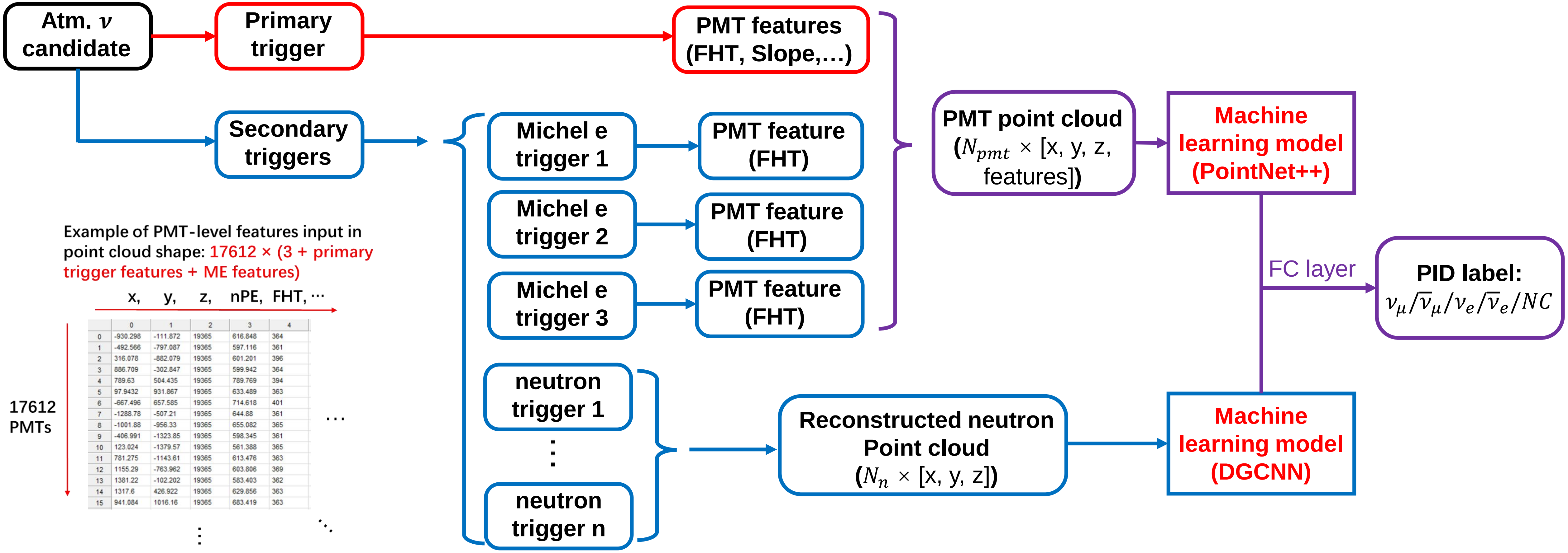
Architecture of PointNet++



Architecture of DGCNN

Machine Learning Models & Strategies

PointNet++ & DGCNN models strategy:



Example of PMT-level features input in point cloud shape: 17612 × (3 + primary trigger features + ME features)

x, y, z, nPE, FHT, ⋯

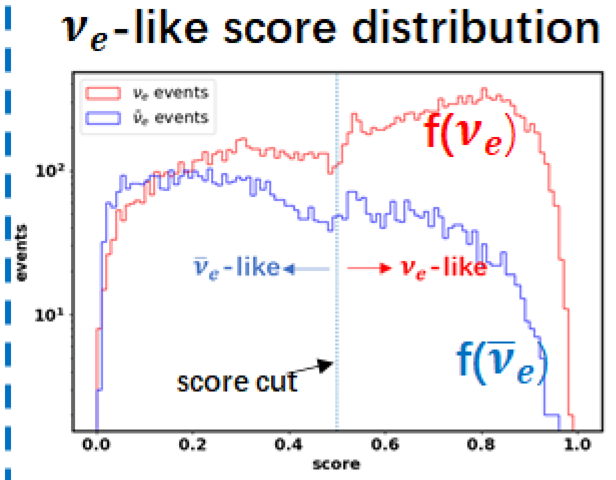
17612 PMTs

	0	1	2	3	4
0	-930.298	-111.872	19365	616.848	364
1	-492.566	-797.087	19365	597.116	361
2	316.078	-882.079	19365	601.201	396
3	886.709	-302.847	19365	599.942	364
4	789.63	504.435	19365	789.769	394
5	97.9432	931.867	19365	633.489	363
6	-667.496	657.585	19365	714.618	401
7	-1288.78	-507.21	19365	644.88	361
8	-1001.88	-956.33	19365	655.082	365
9	-406.991	-1323.85	19365	598.345	361
10	123.024	-1379.57	19365	561.388	365
11	781.275	-1143.61	19365	613.476	363
12	1155.29	-763.962	19365	603.806	369
13	1381.22	-102.202	19365	583.403	362
14	1317.6	426.922	19365	629.856	363
15	941.084	1016.16	19365	683.419	363

Advantages: An individual model is applied to extract features from the reconstructed neutron information.

Results

Take $\nu_e/\bar{\nu}_e$ discrimination as example:

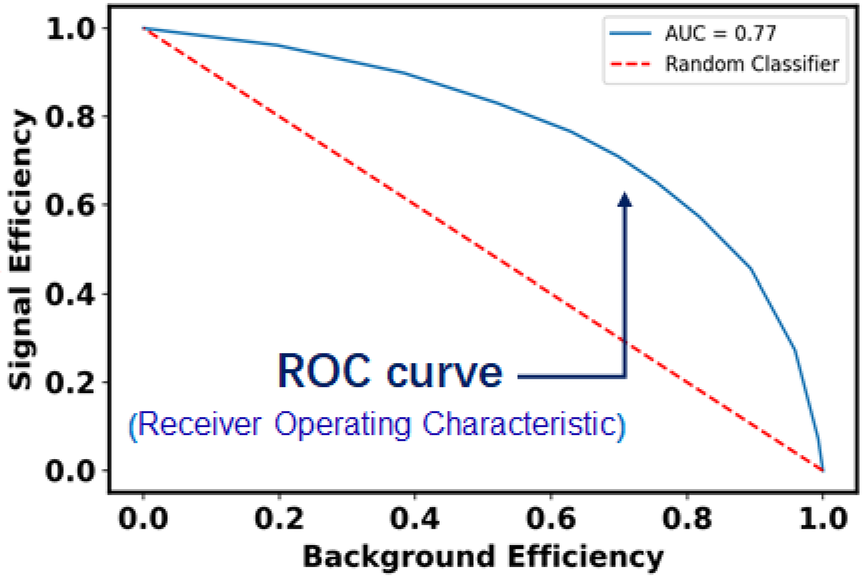


Confusion Matrix

Truth \ Pred	ν_e-like	$\bar{\nu}_e$-like
ν_e	$A=\int_{cut}^{1} f(\nu_e)$	$B=\int_{0}^{cut} f(\nu_e)$
$\bar{\nu}_e$	$C=\int_{cut}^{1} f(\bar{\nu}_e)$	$D=\int_{0}^{cut} f(\bar{\nu}_e)$

Signal efficiency : $\frac{A}{A+B}$ Background efficiency : $\frac{D}{C+D}$

(Treat ν_e as signal, $\bar{\nu}_e$ as background)



- ➢ Efficiency and purity can be easily tuned with the cut on score.
- ➢ Therefore, the Area Under the ROC Curve (AUC) is used to evaluate the capability of classification task.

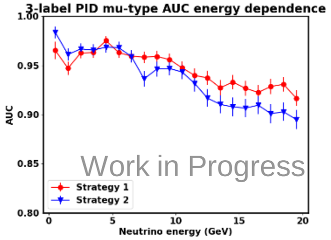
Advantages of using AUC to measure model performance:

- ➢ AUC does not depend on the choice of score cut;
- ➢ AUC is not affected by the imbalance of samples in each class.

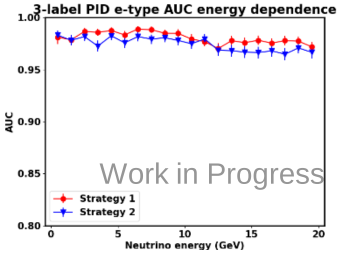
Results

3-labels PID ($\nu_\mu/\bar{\nu}_\mu$-CC VS $\nu_e/\bar{\nu}_e$-CC VS NC):

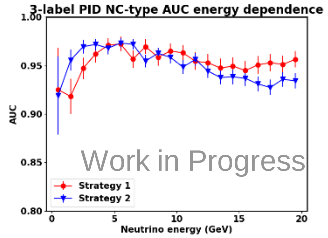
$\nu_\mu/\bar{\nu}_\mu$-CC

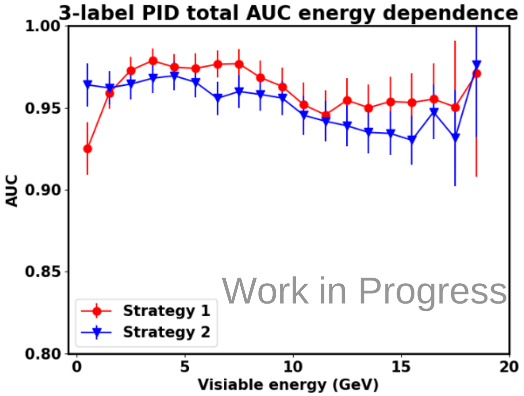


$\nu_e/\bar{\nu}_e$-CC



NC

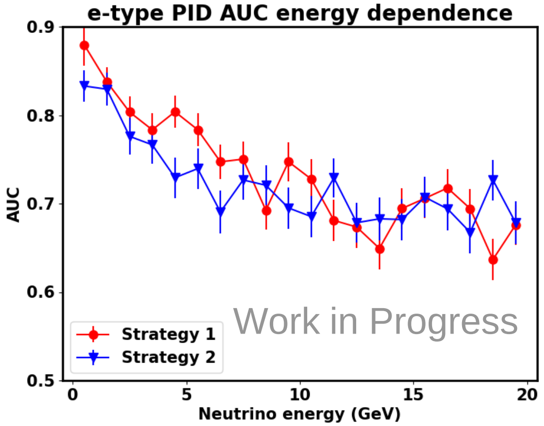




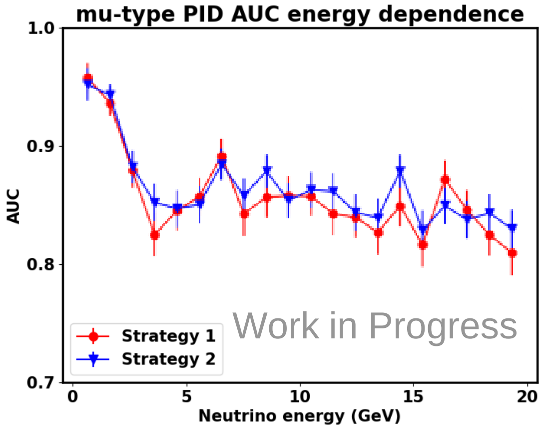
Total AUC for 3-labels PID

- AUC is only applicable for binary classification scenarios;
- For 3-label PID, it is possible to calculate the AUC for each label individually and take the average to obtain the overall AUC of the model.

2-labels PID ($\nu/\bar{\nu}$ discrimination):



$\nu_e/\bar{\nu}_e$

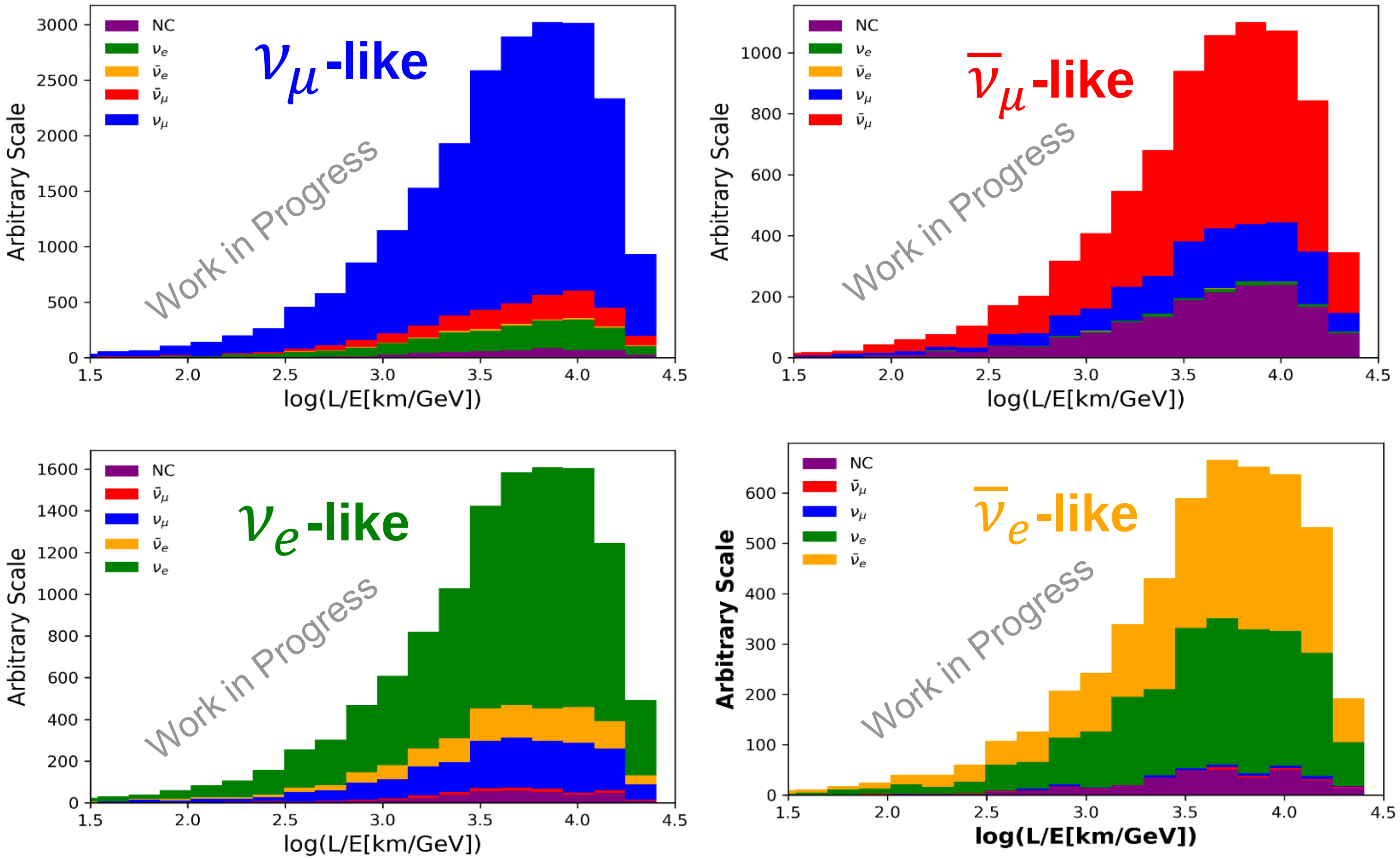


$\nu_\mu/\bar{\nu}_\mu$

- All the results were obtained from one same Monte Carlo simulation sample.
- The energy dependences of AUC are mostly consistent for the two strategies.

Results

- For the upcoming NMO study, the efficiency and purity for each label will be tuned to obtain the best sensitivity.
- An example of the background composition for each CC-like label after tuning:



Upward-going events only, E_{vis}>0.5 GeV

Summary

- In this talk, a general machine learning approach of atmospheric neutrino particle identification was introduced, which could be extended to other large homogeneous liquid scintillator detectors as well.
- two individual PID strategies with different types of machine learning models were developed to cross validate the method.
- Preliminary results (AUC) based on Monte Carlo simulations show promising potential for this approach.
- The final performance of atmospheric neutrino (efficiency & purity) will be tuned to obtain the best NMO sensitivity in JUNO.

THANKS!

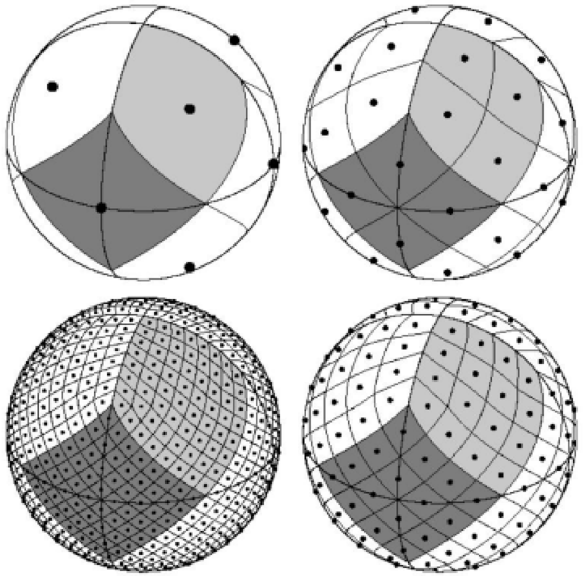
Backups

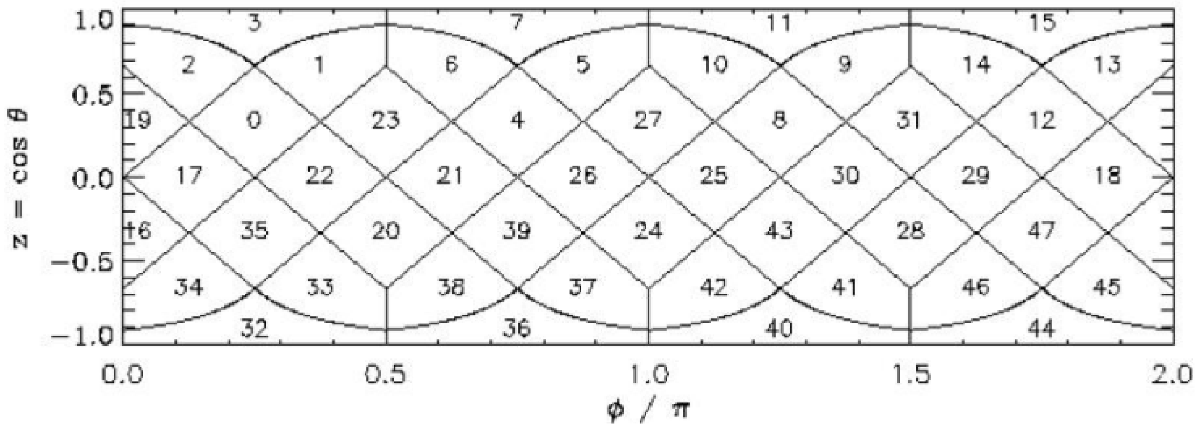
ArXiv: 2012.15000

DeepSphere: Graph-CNN for Spherical Data

Use healpix sampling to define vertices

- Equally divide the sphere into 12 parts
- Further divide each part into N_{side} parts ($N_{side}=2^n$).
- Total number of pixels is $12X2^n$
- If more than one PMTs are in one pixel, info is merged





PointNet & PointNet++

PointNet: arXiv:1612.00593
PointNet++: arXiv:1706.02413

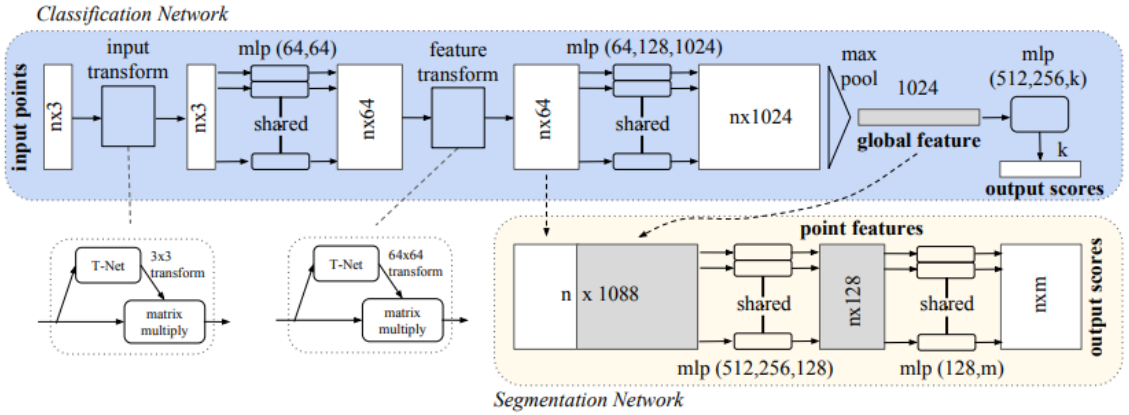
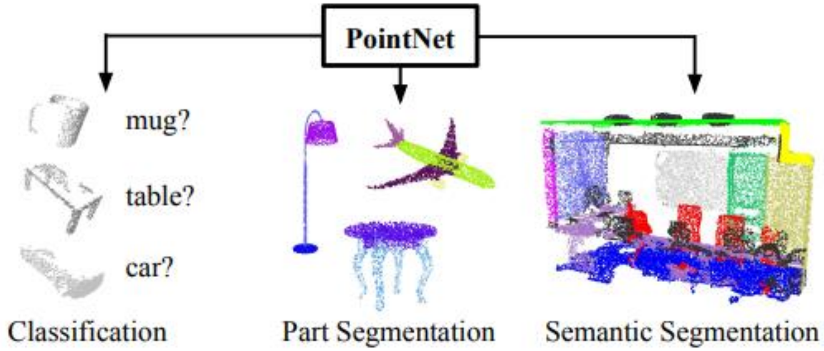
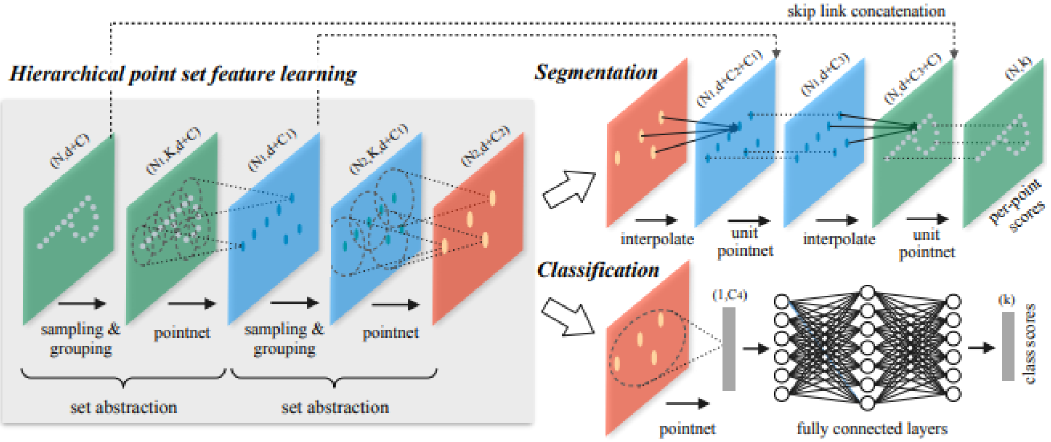


Figure 2. **PointNet Architecture.** The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.



	#params	FLOPs/sample
PointNet (vanilla)	0.8M	148M
PointNet	3.5M	440M
Subvolume [18]	16.6M	3633M
MVCNN [23]	60.0M	62057M



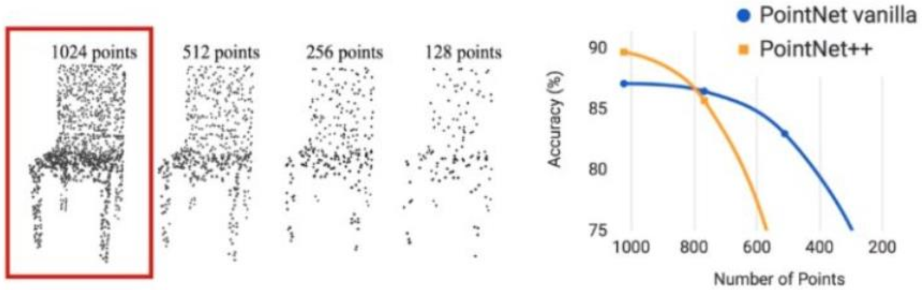
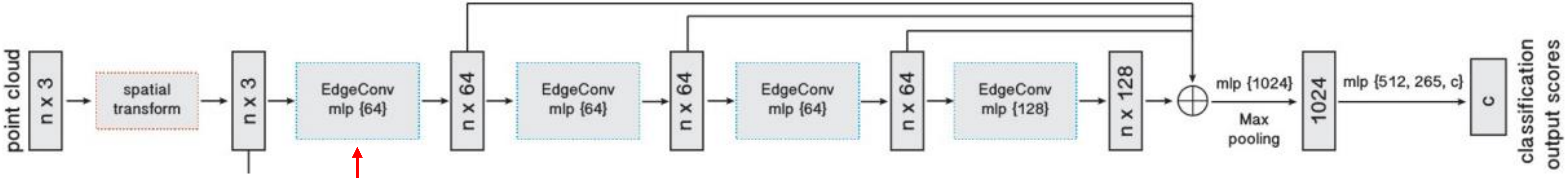
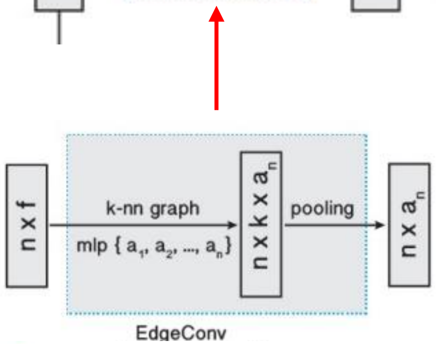


Figure 2: Illustration of our hierarchical feature learning architecture and its application for set segmentation and classification using points in 2D Euclidean space as an example. Single scale point grouping is visualized here. For details on density adaptive grouping, see Fig. 3

DGCNN: Dynamic Graph CNN

- Based on PointNet and PointNet++, could capture local geometrical structure
- Calculate the nearest k points for each point, extract features from edges
- Dynamic: Re-calculate the k-nn graph for each layer, and classify points with similar semantic information





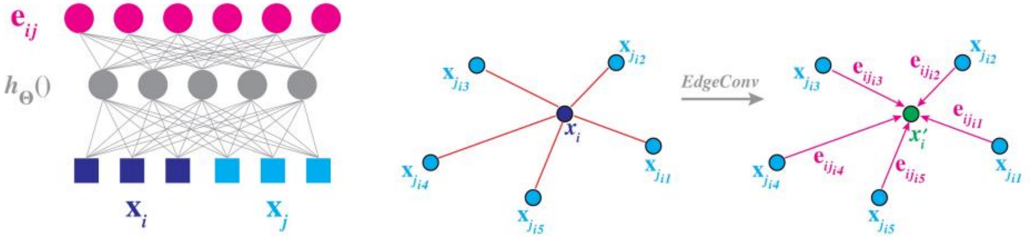
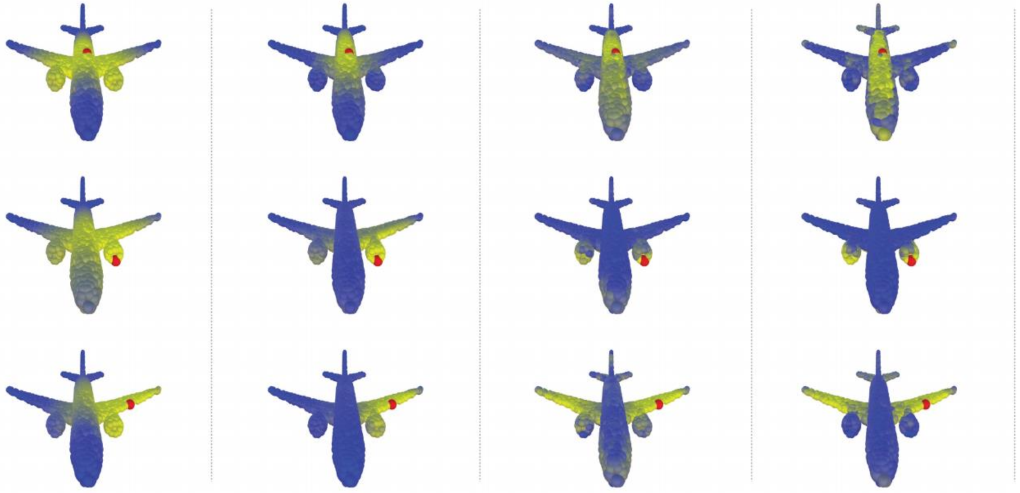


Fig. 2. **Left**: Computing an edge feature, $\mathbf{e}_{ij}$ (top), from a point pair, $\mathbf{x}_i$ and $\mathbf{x}_j$ (bottom). In this example, $h_\Theta()$ is instantiated using a fully connected layer, and the learnable parameters are its associated weights. **Right**: The EdgeConv operation. The output of EdgeConv is calculated by aggregating the edge features associated with all the edges emanating from each connected vertex.



Nearest points for the red point:
Euclidean distance → semantic distance