
# Sparse Convolutional Neural Network for NOvA

**APS April Meeting - 4/17/23**
**Stella Haejun Oh**

# Outline

University of CINCINNATI

1. The NOvA Experiment
    - The detector & images
2. NOvA Reconstruction
3. Sparse Convolutional Neural Network
    - What are SCNNs?
    - Sparse FishNet Training & Results
4. Future Work
5. Summary

# NOvA Experiment

University of CINCINNATI

- NOvA = NuMI Off-axis $\nu_e$ Appearance
- Long baseline neutrino oscillation experiment of 810 km
- Major Goals:
  - $\nu_\mu \rightarrow \nu_e$ appearance and $\nu_\mu \rightarrow \nu_\mu$ disappearance
    - Neutrino Mass Ordering
    - CP violation phase
    - Atmospheric oscillation parameters
  - Sterile neutrino oscillations
  - Supernova neutrinos
- Uses NuMI $\nu_\mu$ or $\bar{\nu}_\mu$ beam at Fermilab

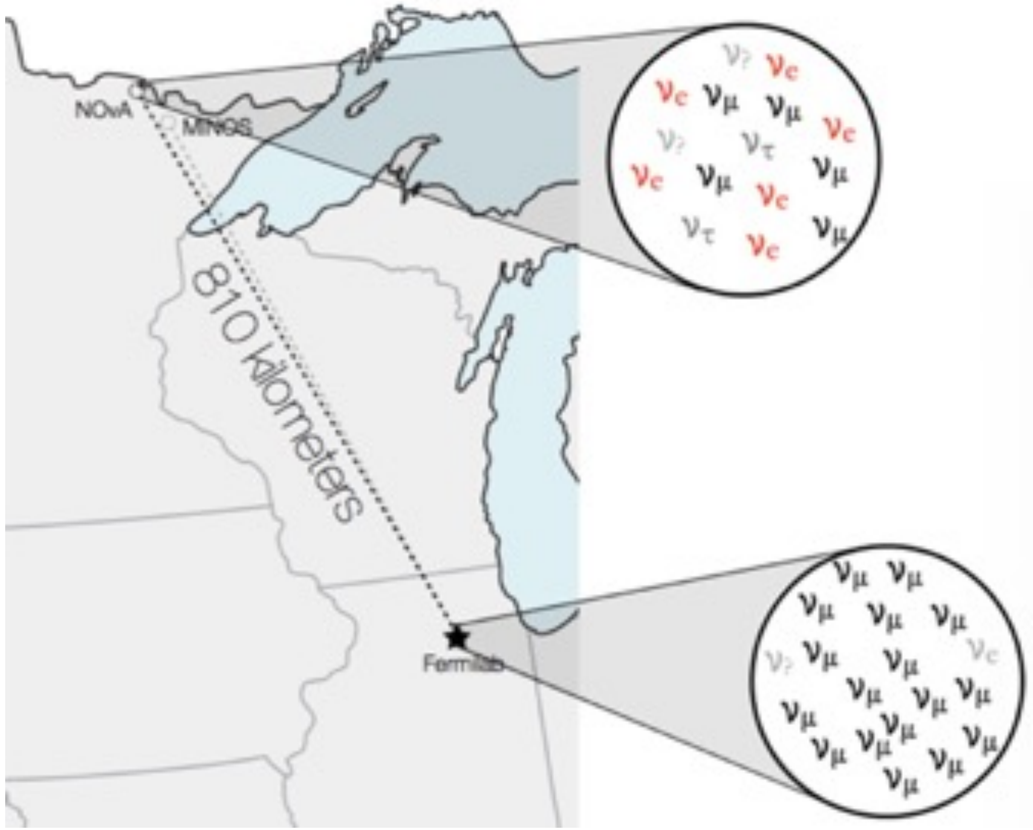



**Figure 1:** Location of two detectors in NOvA [1].

# NOvA Detectors

- Two functionally equivalent calorimeter detectors:
    - 300 ton near detector (ND) is 100 m underground at Fermilab
    - 14 kton far detector (FD) sits on the surface 810 km away in Ash River, Minnesota
- Made of liquid scintillator inside PVC cells. Alternate in horizontal and vertical orientation for 3D image reconstruction.
- When neutrino and non neutrino events pass through the cells, light is produced.
- The light is picked up by a wavelength shifting optical fiber, which then travels to avalanche photodiode (APD) where light is amplified and analyzed.

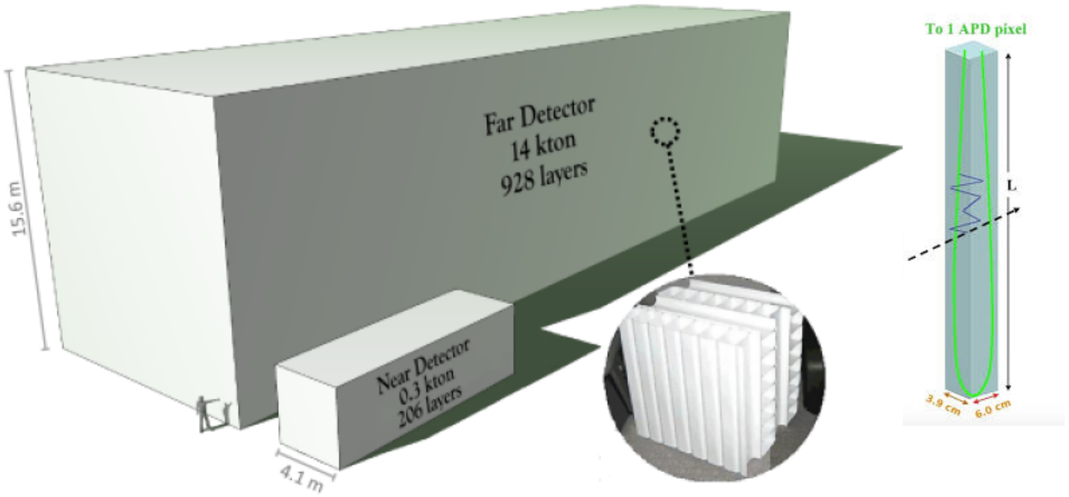



**Figure 2:** FD and ND detectors and liquid scintillator PVC cells [2].

# NOvA Detector Images

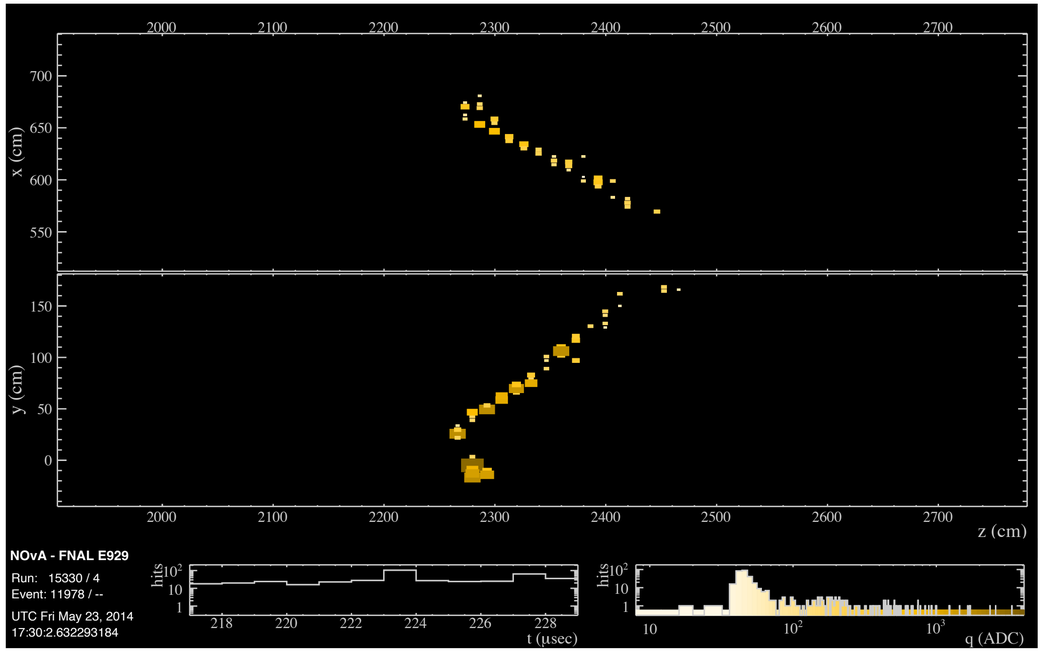



**Figure 3:** The top panel shows XZ view of $\nu_e$ event while the bottom panel is the YZ view of $\nu_e$ event [3].

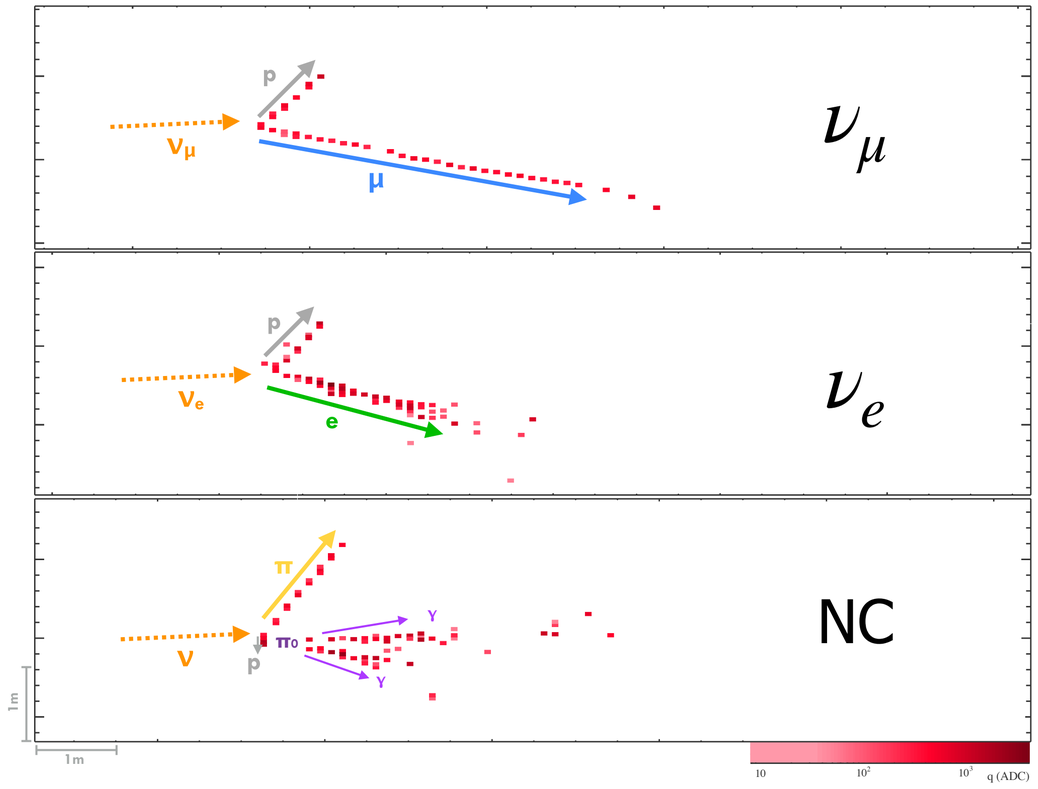



**Figure 4:** Example event topologies from NOvA data files. TOP: selected $\nu_\mu$ ND event. MID: selected $\nu_e$ ND event. BOT: selected NC FD event [3].

# NOvA Event CNN Training and Performance

University of CINCINNATI

- NOvA's current event classifier network called, Event CNN, is a modified version of MobileNetV2 [7]. Reduced number of layers and convolutions to minimize runtime on CPU.
- Dataset consists of ~ 2.5 million events of each $\nu_\mu, \nu_e$ and NC events using NOvA simulation files.
- Cosmic limited to 10% of sample. Underwent preselection including cosmic veto.
- Sample split into 90% training and 10% validation dataset.
- Confusion matrix in Figure 5. show that the network achieves over 90% for all event types except for $\nu_e$ events [4].

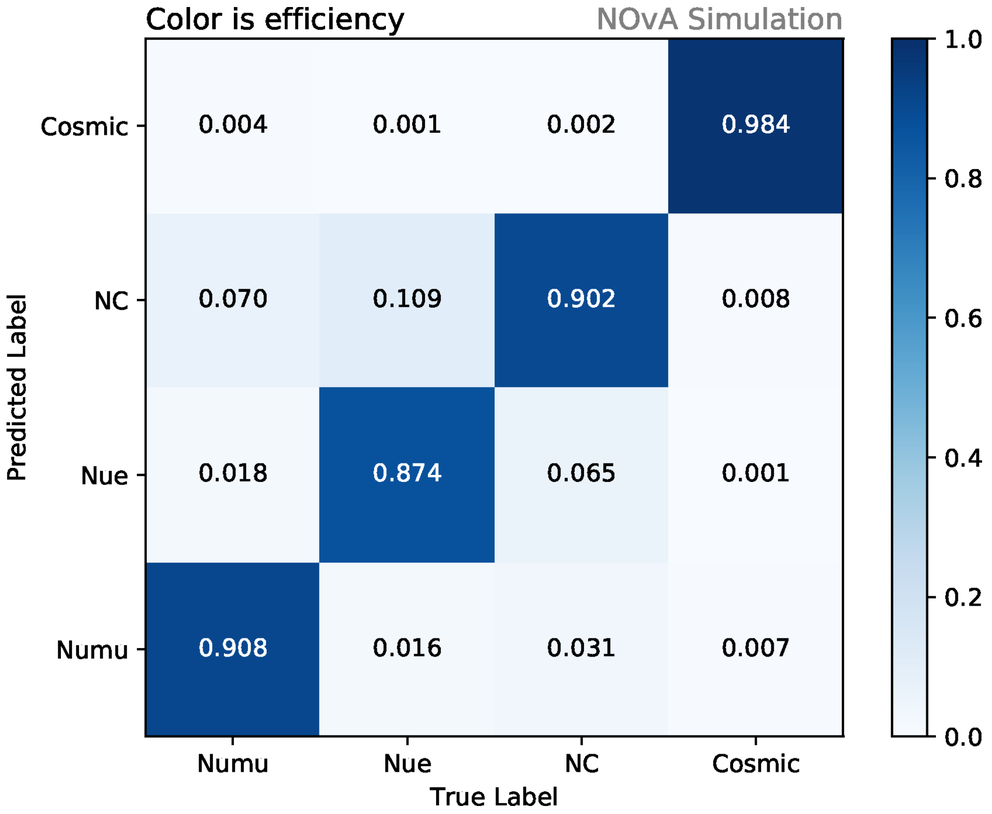



**Figure 5:** Confusion matrix shows the efficiency of each type along the diagonal. The off-diagonal cells show incorrectly classified events [**4**].

# Sparse Convolutional Neural Network

University of CINCINNATI

- Traditional CNN methods in HEP experiments spend a lot of time and GPU resources on zero pixels. Neutrino interactions are globally sparse and locally dense.
- Sparse CNN is an alternative to CNN that applies convolutions only to non-zero pixels and skips zero pixels.
- First implemented by Terao & Domine [5], found significant reduction of computational memory and wall-time cost without loss of accuracy.
- Our main goal is to have make a comparison with the current NOvA Event CNN accuracy results and SCNN results.

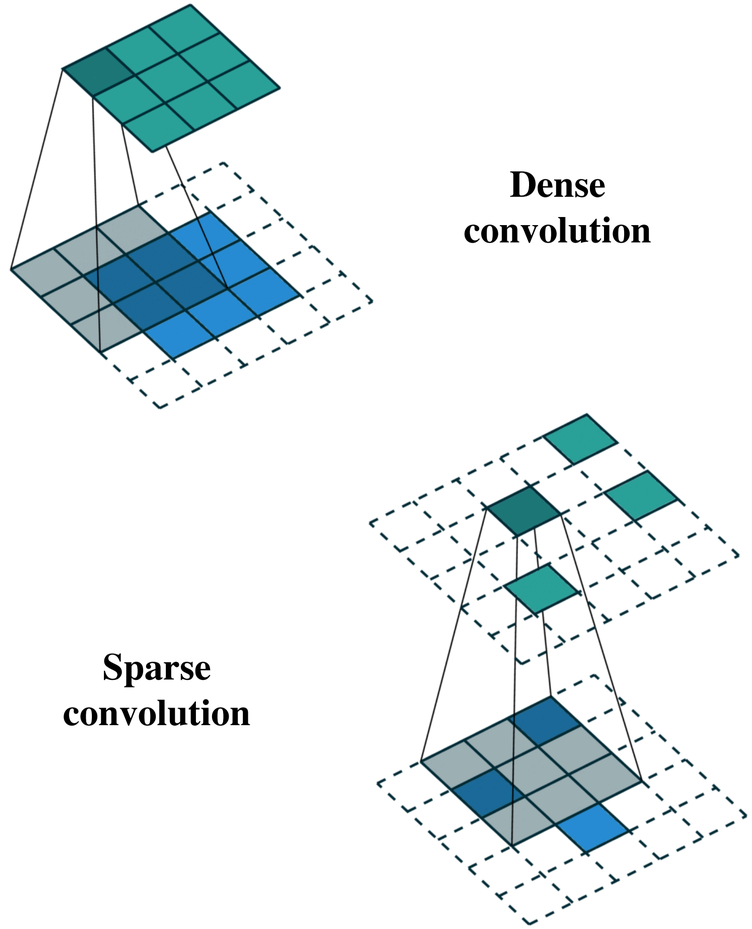



**Figure 6:** Animation to show different between Dense and Sparse convolutions.

# FishNet Architecture

- FishNet [10] is a modified UNet architecture suitable for both pixel-level and region-level prediction. Attempts to perform both classification and semantic segmentation.
- Semantic segmentation labels specific region of image instead of every pixel.
- Figure 7 is a visual representation of the architecture which features three parts of the "Fish": Tail, Body, Head (arXiv:1901.03495).
- Using **MinkowskiEngine** [6] package, we we have two parallel sparse networks for the tail and body for XZ view and YZ view for classification.
- At the top of the body, the feature maps of the views are merged and then continue to down sample for one single head.
- We have implemented **PyTorch Lightning** [13] which is an organized PyTorch. It allows for more flexibility, self contained models and the structure of our code becomes reusable and shareable.

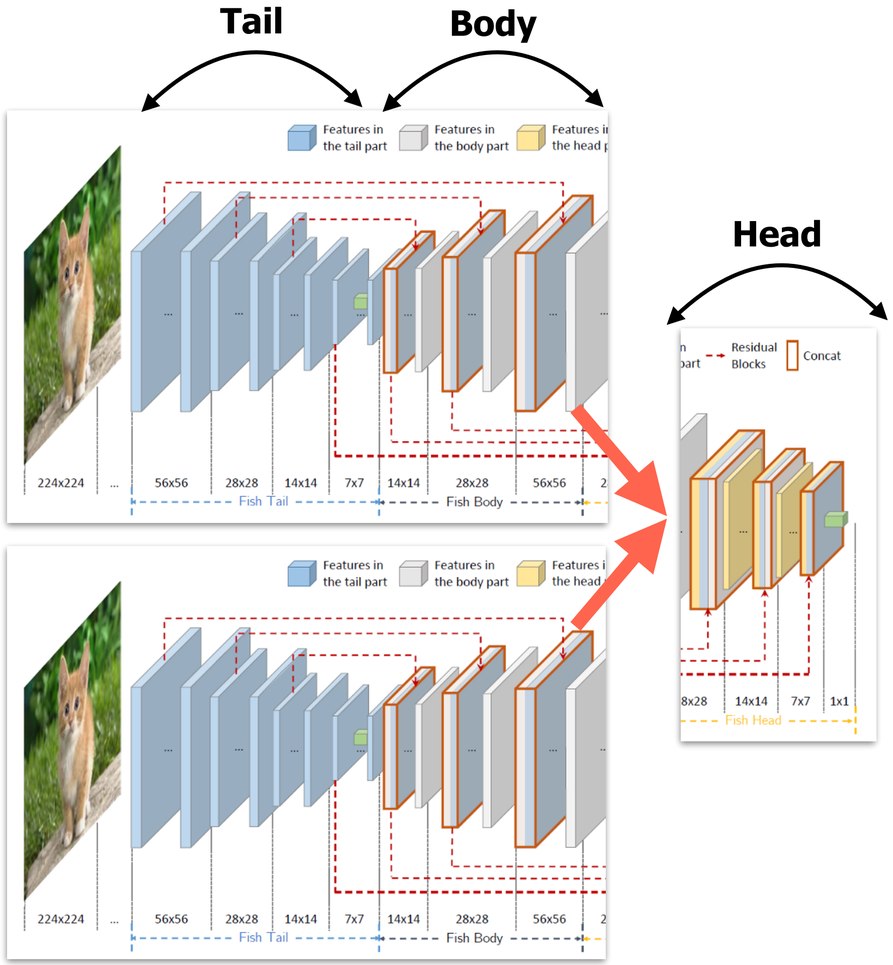



**Figure 7:** FishNet architecture [**10**]

# Sparse FishNet Training Parameters & Results

University of CINCINNATI

- Used NOvA simulated files with a total of ~6.6 million $\nu_e, \nu_\mu$, and NC events without cosmic events.
- Total sample was split into 90% training, 5% validation, and 5% testing dataset.
- Each XZ and YZ view images were fluctuated by 10% to mimic the effect of systematic uncertainty on calibration.
- Hyper-parameters used during training:
    - Stochastic Gradient Descent (SGD) optimizer [8]
    - Mish activation function [9]
    - ReducedLROnPlateau - learning rate scheduler
        - Learning Rate: 1e-2, Momentum: 0.8, Patience: 3
    - Cross Entropy Loss [14] function
    - Batch size of 1024 and 30 epochs
    - Standardized inputs
- Sparse FishNet - **89.9%** validation accuracy & **91.54%** training accuracy.
- Sign of overtraining - which means the network is not learning any new information and therefore the accuracy falls after its peak performance over time.

# Sparse FishNet vs Event CNN

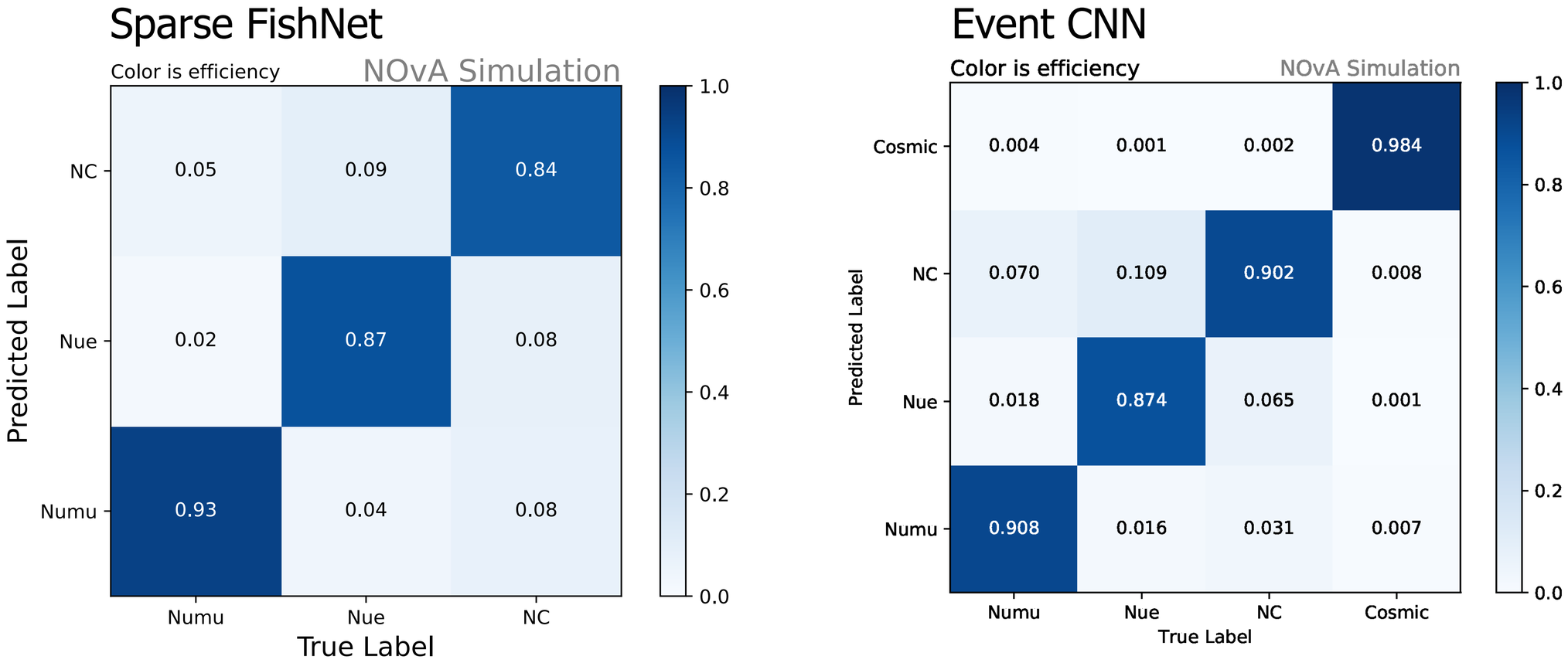



**Figure 8:** confusion matrix comparing the training efficiency between true label and assigned/predicted label. Left confusion matrix is Sparse FishNet while the right matrix is Event CNN.

# Future Work: Instance and Semantic Segmentation

- Next step for FishNet - implement instance and semantic segmentation in 2D.
    - Semantic segmentation identifies each pixel of an image with a corresponding class [11].
    - Instance segmentation distinguishes each object from one another in an image [12].
- Instance and semantic segmentation will be performed on XZ and YZ view detector images independently. Therefore, there is no guarantee that both views will agree with each other.
- Then, we would explore panoptic segmentation in 3D, which would guarantee that both types of segmentation are consistent between both views.

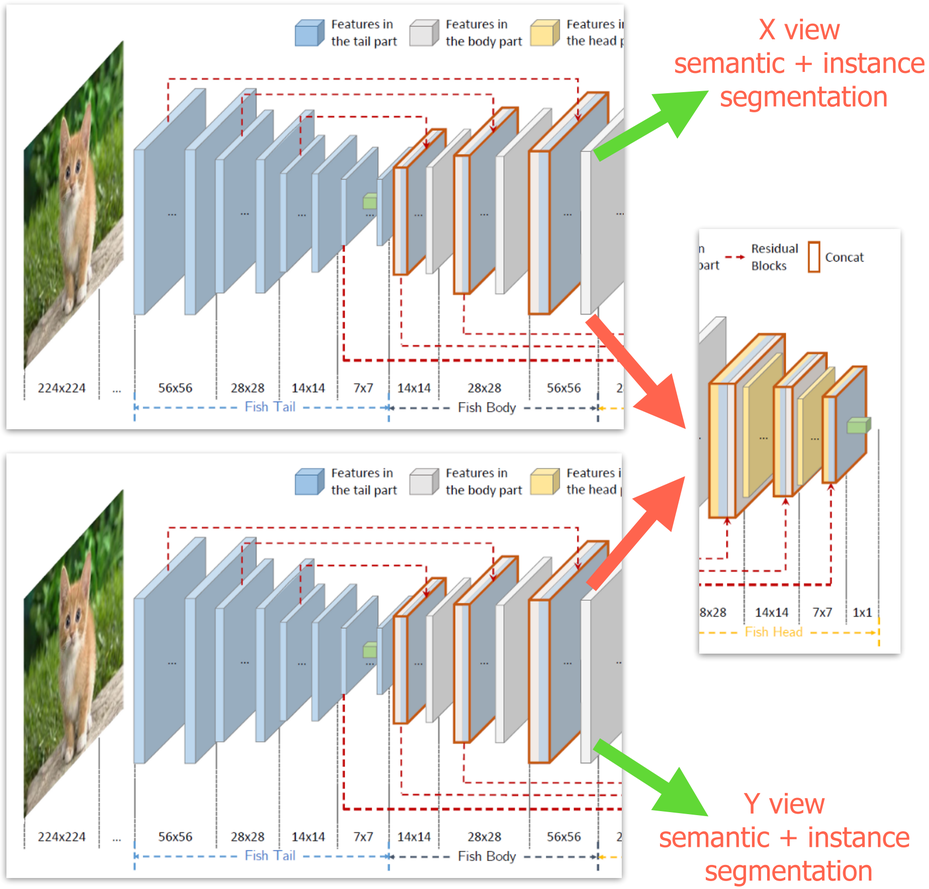



**Figure 9:** FishNet architecture with instance and semantic segmentation.

# Future Work: Instance and Semantic Segmentation

University of CINCINNATI

- The following visual representation shows step by step process of how we hope semantic and instance segmentation will identify objects while panoptic segmentation will effectively correct any errors.

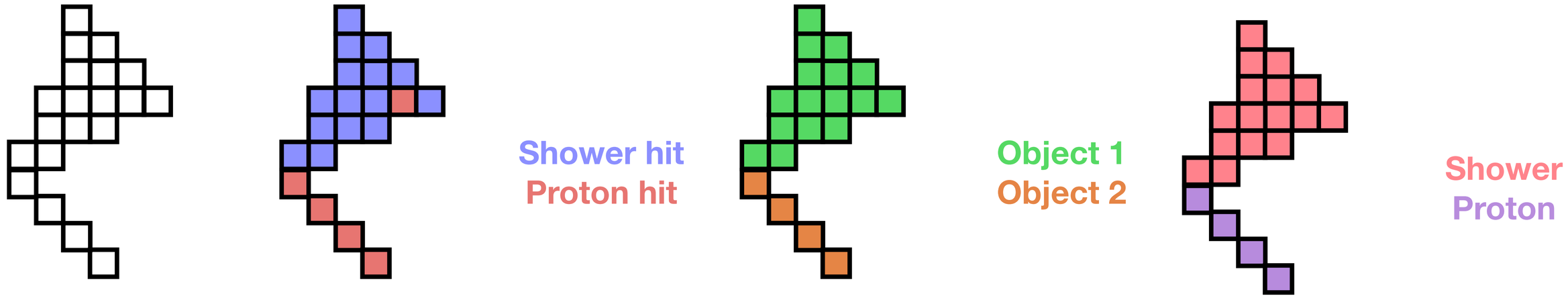



**Figure 10:** Instance & semantic segmentation illustration.

# Summary

University of CINCINNATI

- Sparse FishNet architecture using PyTorch Lightning framework achieved 89.92% in validation accuracy which falls couple percentages behind NOvA's current network, Event CNN.
- Sparse FishNet outperforms Event CNN when classifying $\nu_\mu$ events but falls shorts when classifying NC events.
- Even though we are getting close to the performance of Event CNN, we are still looking for other ways to further increase our accuracy.
- We are continuing to explore instance and semantic segmentation and simultaneous implementation.


# Thank you!
# Comments/Questions?

# References


[1] Queen Mary University of London HEP, NOvA Overview, https://www.qmul.ac.uk/spa/pprc/research/neutrino-physics/nova/
[2] Patterson, R. B.. "The NOvA experiment: status and outlook.", (2012)
[3] Psihas, F.. "Event Display for NuE, NuMu, and NC Selected Events." (Jun 2016).
[4] Groh, M. Thesis "Constraints on Neutrino Oscillation Parameters from Neutrinos and Antineutrinos with Machine Learning" (February 2021) https://inspirehep.net/literature/1854876.
[5] Dominé, Laura, and Kazuhiro Terao. "Scalable Deep Convolutional Neural Networks for Sparse, Locally Dense Liquid Argon Time Projection Chamber Data." *ArXiv.org*, 21 Dec. 2019, https://arxiv.org/abs/1903.05663.
[6] Choy, Christopher, et al. "4D Spatio-Temporal Convnets: Minkowski Convolutional Neural Networks." *ArXiv.org*, 13 June 2019, https://arxiv.org/abs/1904.08755.
[7] Sandler, Mark, et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks." *ArXiv.org*, 21 Mar. 2019, https://arxiv.org/abs/1801.04381.
[8] Ruder, S.. "An overview of gradient descent optimization algorithms." ArXiv abs/1609.04747 (2016): n. pag.
[9] Misra, D.. "Mish: A Self Regularized Non-Monotonic Activation Function." BMVC (2020) https://arxiv.org/abs/1908.08681.
[10] Sun, S. e*t al*. "Fishnet: A versatile backbone for image, region, and pixel level prediction." In NeurIPS, 2018 https://arxiv.org/abs/1901.03495.
[11] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena- Martinez, Pablo Martinez-Gonzalez, and Jose Garcia-Rodriguez. "A survey on deep learning techniques for image and video semantic segmentation". Applied Soft Computing, 70:41–65, 2018.
[12] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance segmentation: state of the art. International Journal of Multimedia Information Retrieval, 9(3):171–189, Jul 2020 https://arxiv.org/abs/2007.00047.
[13] PyTorch Lightning Machine Learning Framework, https://pytorch-lightning.readthedocs.io/en/1.5.10/api/pytorch_lightning.core.lightning.html
[14] Zhang, Z. and Sabuncu, M. R.. "Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels", https://arxiv.org/abs/1805.07836 (May 2018)


# Backup Slides: CNN

# NuMI Beam at Fermilab

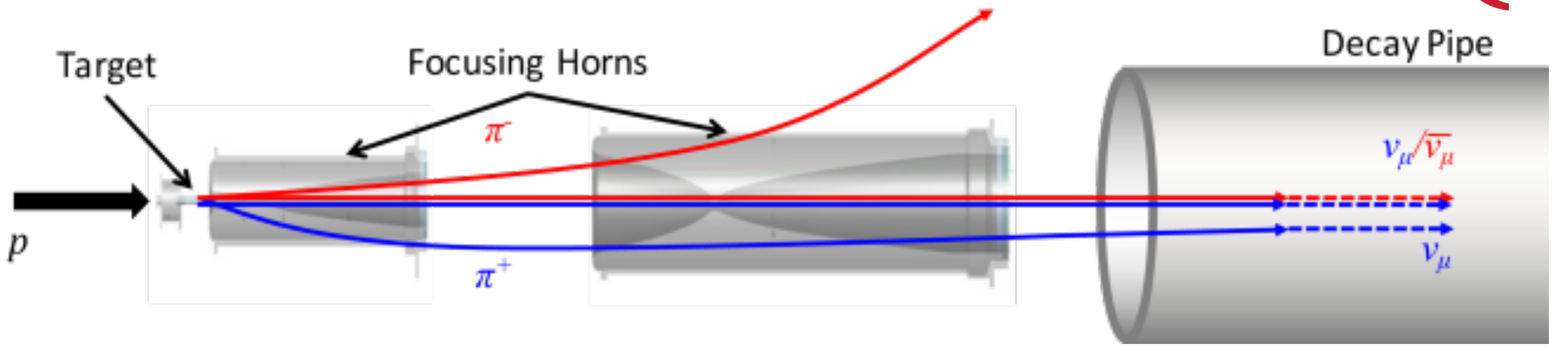



**Figure 2:** The NuMI proton beam creates pions and kaons which then decays into neutrinos.

- The NuMI (Neutrinos at the Main Injector) beam at Fermilab produces 120 GeV protons from the main injector.
- Protons decay into two modes: $\pi^+ \rightarrow \mu^+ + \nu_\mu$ and $K^+ \rightarrow \mu^+ + \nu_\mu$ creating 97.5% $\nu_\mu$ beam
- 1.8% $\bar{\nu_\mu}$ contamination comes from negative hadrons
- 0.7% $\nu_e$ contamination due to subdominant electronic decay mode of $K^+$ hadrons, decays of $K^0$ particles and tertiary muons.

# NOvA Reconstruction - CNN

- NOvA uses convolutional visual network (CVN) which uses deep learning algorithm called convolutional neural network (CNN) as its event selector and particle identifier since 2016.

- A basic deep learning algorithm contains an input layer, multiple hidden layers, and an output layer.

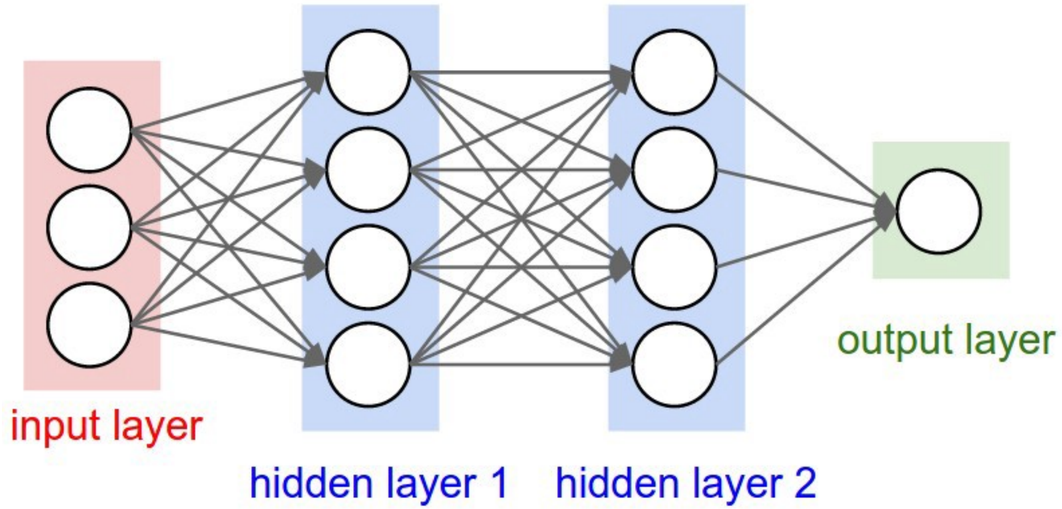



**Figure 15:** Basic neural network in deep learning algorithm [14].

# NOvA Reconstruction - CNN

- DL algorithm then goes through **forward propagation**.
- Randomly chosen weights (w) are applied to each neurons ($x_i$). Values for each neuron are calculated at the each hidden layer by using a linear equation (6) with bias (b).
  - $H_1 = x_1 * w_1 + x_2 * w_2 + x_3 * w_3 + b$ (6)
- **Activation function** is used to transfer these linear values into nonlinear before going to the next hidden layer.

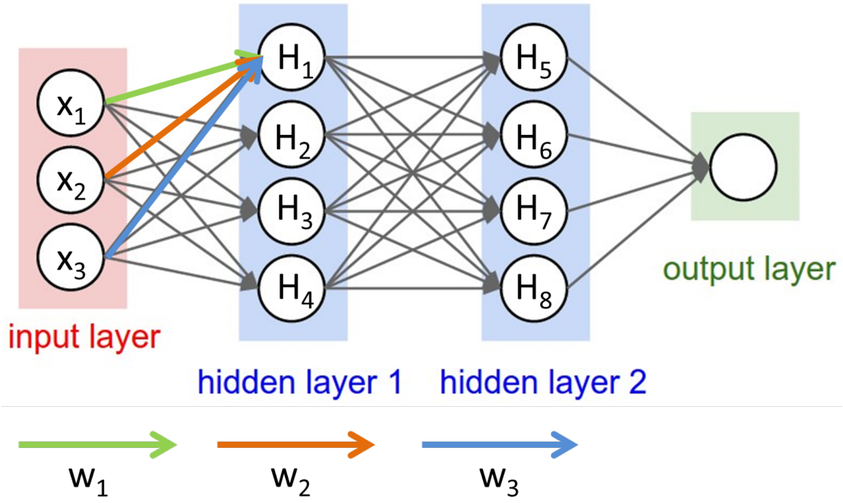



**Figure 16:** Weights applied to each neurons. Then used to calculate hidden layers [14].

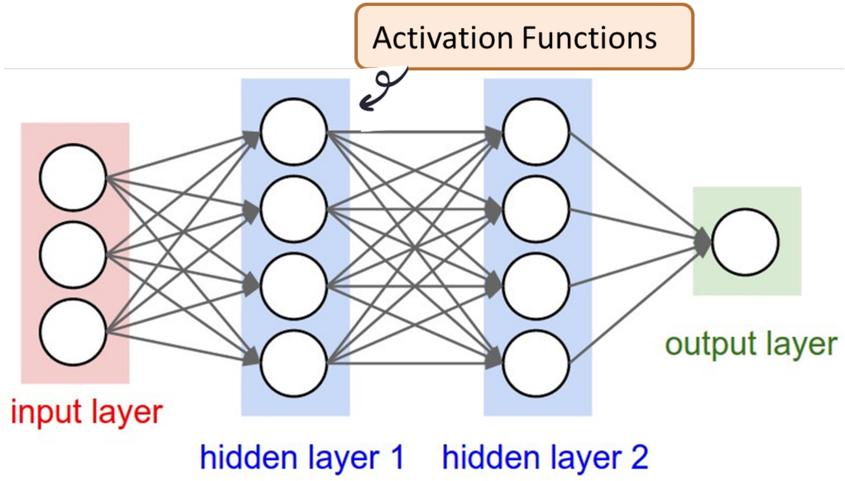



**Figure 17:** Activation function is applied after each hidden layer before the next [14].

# NOvA Reconstruction - CNN

University of CINCINNATI

- After the output layer, the algorithm goes through **backward propagation**. This process calculates the loss/error between the predicted values and true values using a loss function.
- Based on the loss, **optimizers** update the weights to be used in the next set of neurons in the input layer.
- This process of forward and backward propagation repeats until convergence is reached.

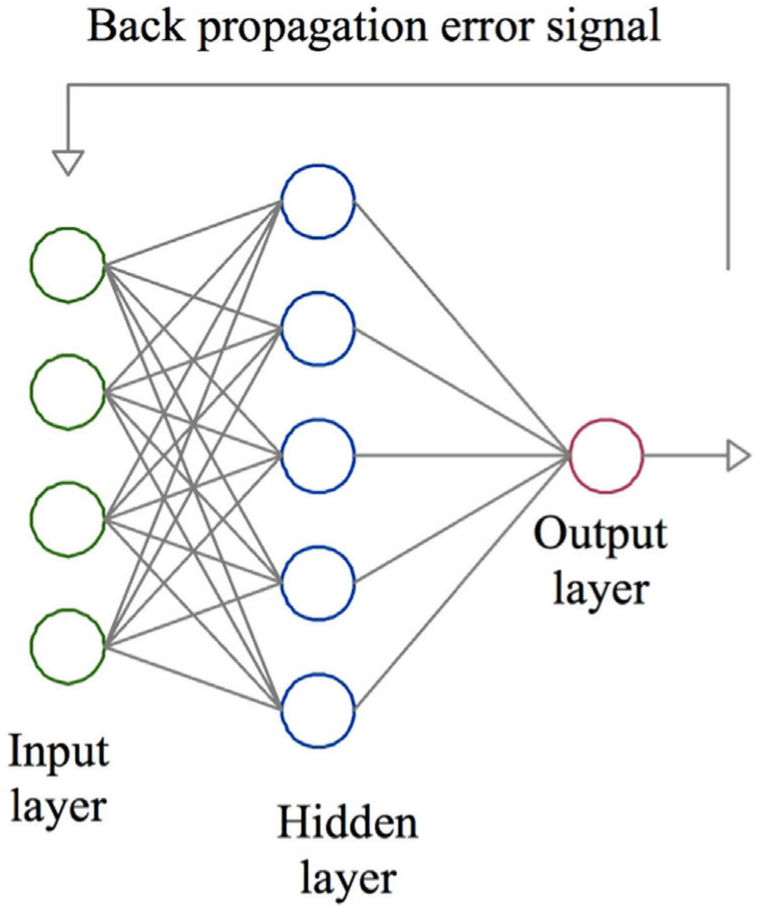



**Figure 18:** Back propagation occurs after the output layer, updates a new set of weights and applies it to the input layer for next iteration.

# NOvA Reconstruction - CNN

University of CINCINNATI

- Each time the weights are updated, it is called one **iteration**.
- # of iteration = $\frac{\text{total sample}}{\text{batchsize}}$
- 1 epoch = # of iteration
- Optimizers, activation functions, and batchsize are all called **hyperparmeters**; values you can choose to control the learning process of the neural network. Based on the choices you make, the performance will change.

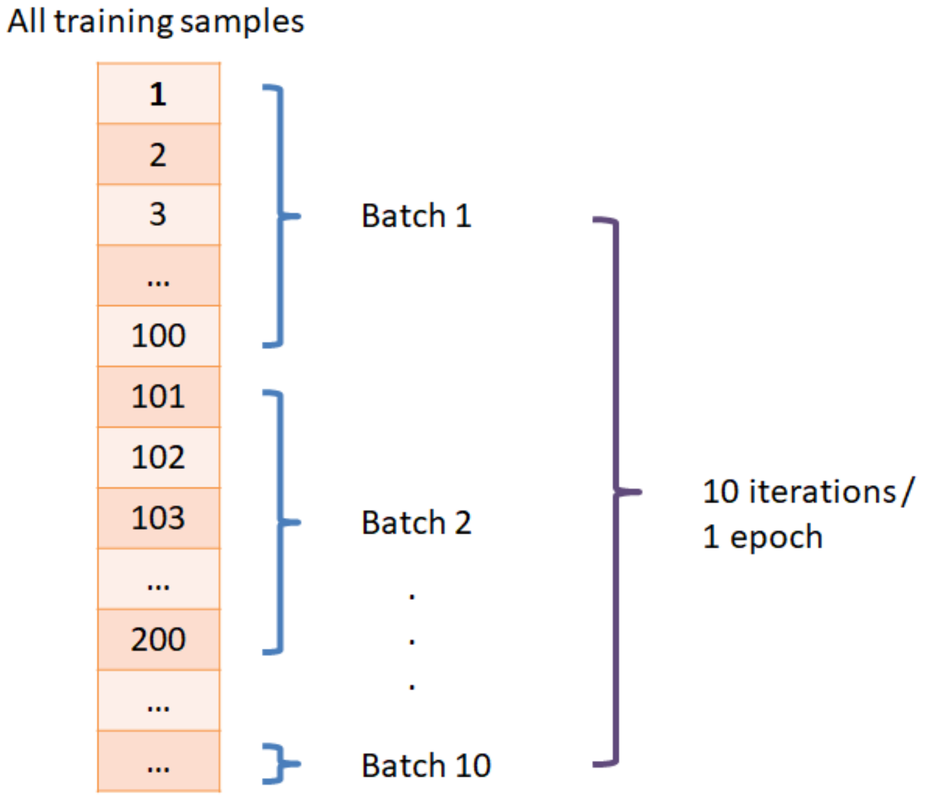



**Figure 19:** Visual representation of samples, batch size, iterations, and epoch [14].

# NOvA Reconstruction - CNN

University of CINCINNATI

- Using convolution operation, a window size sums up each (cell value * weight) in Fig 20. This window, also known as **filter** or kernel, scans from left to right, then top to bottom.
- The filter with activation function applied generates a **feature map**. Note that the weights are reused across the whole feature map.
- Next, **max-pooling** is applied to the feature maps, effectively taking the maximum value of the feature map produce a pooling layer.

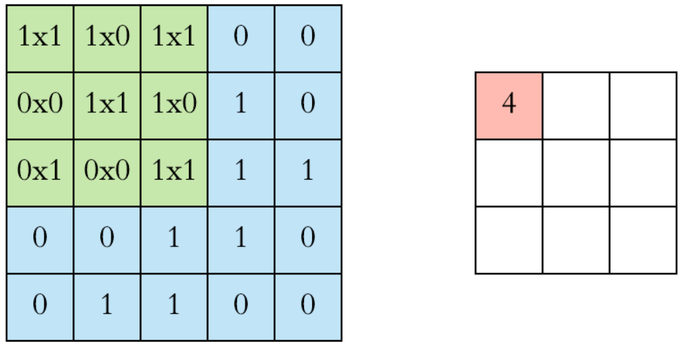



**Figure 20:** Convolutional layer summing the cell weights, while scanning from left to right and top to bottom [14].

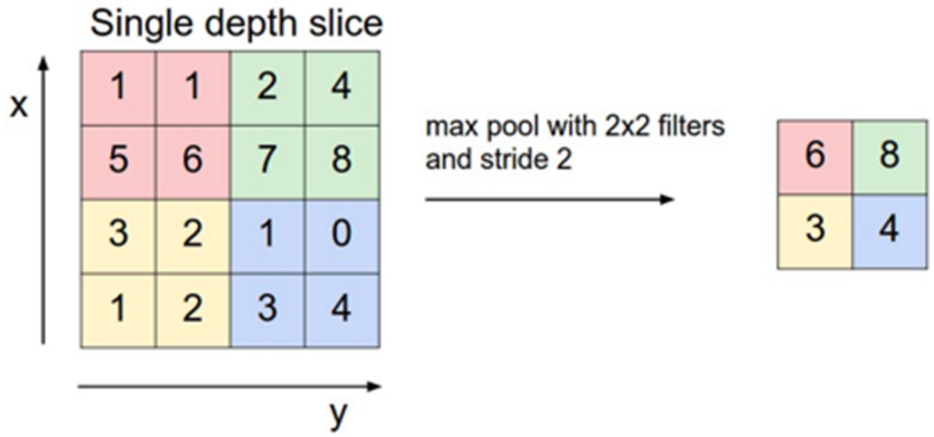



**Figure 21:** Max pooling applied to feature maps [14].

# NOvA Reconstruction - CNN

- Eventually, the feature map will be flattened down to a vector and begin classification with a fully connected neural network.

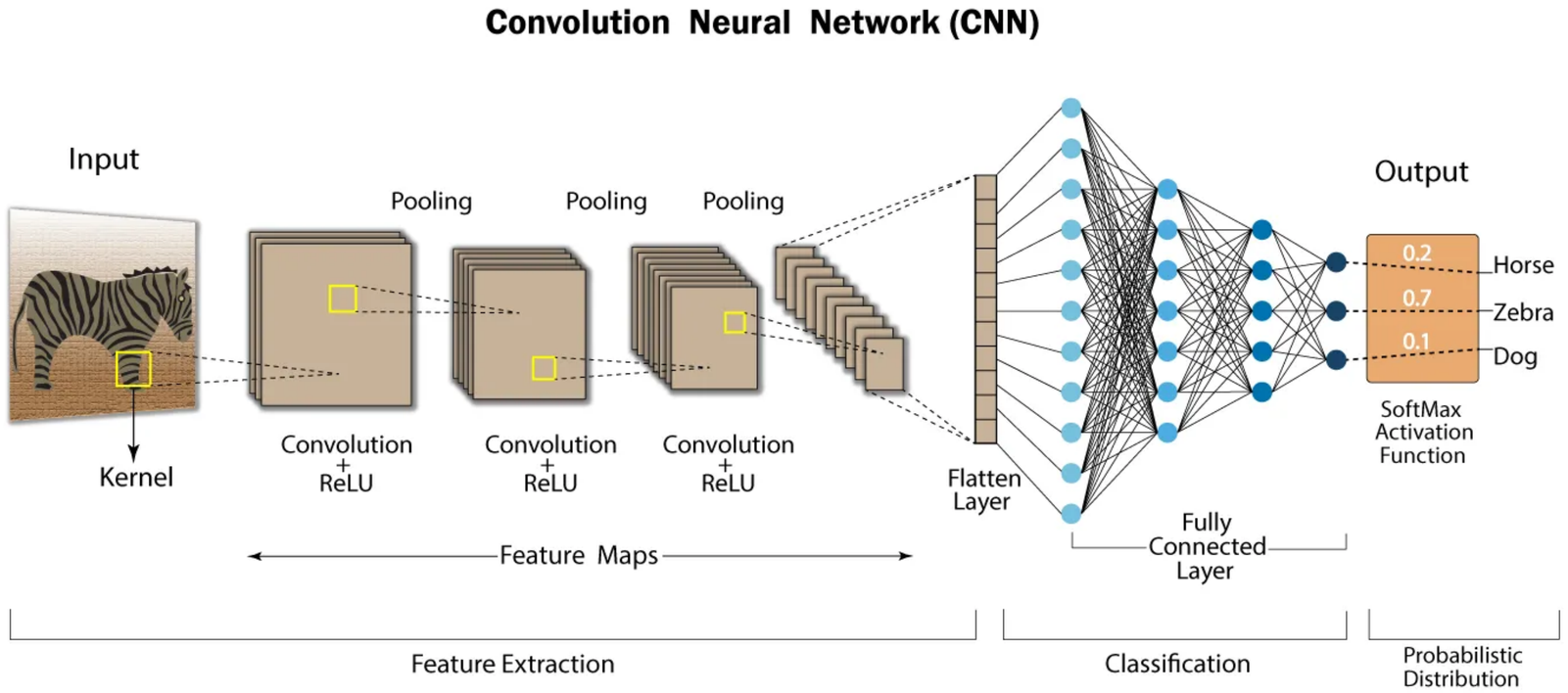



**Figure 22:** Visual representation of convolutional neural network [14].


# Backup Slides: SCNN

# MobileNet V2 Architecture

- MobileNet V2 is an architecture used for NOvA Event CNN.
- We used a sparse equivalent to MobileNet V2 for sparse MobileNet training.
- Initial inputs are XZ and YZ views of the event.
- There is a bottleneck block which effectively expands the input features and performs a depth convolution which then compresses the features again.
- Pooling layers are used throughout the architecture to reduce feature dimensions.
- Output of the network is 4 scores which is used to classify the event as $\nu_e, \nu_\mu$, NC or cosmic.

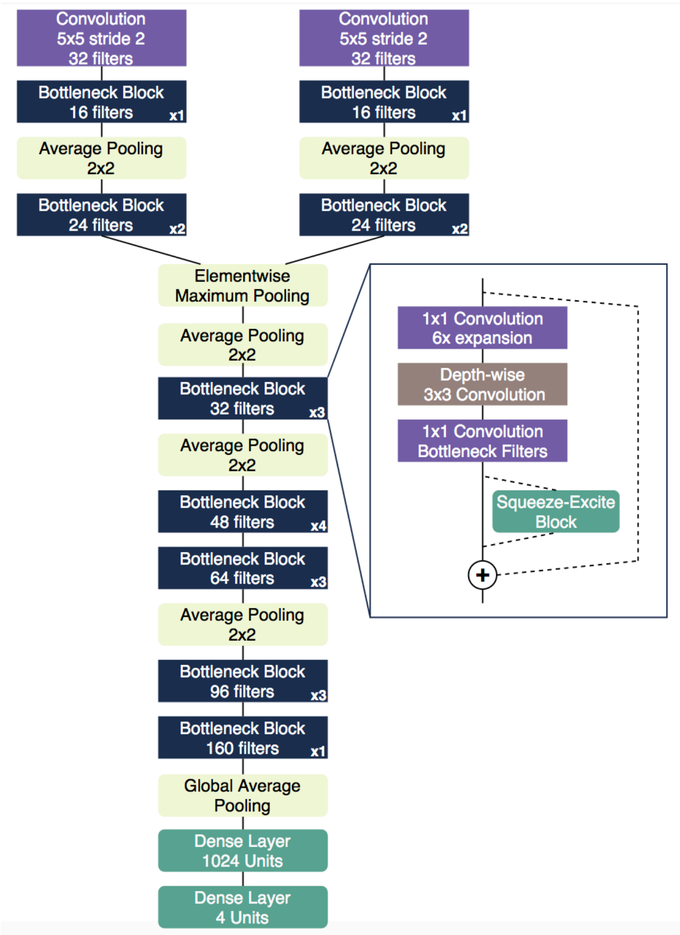



**Figure 9:** Visual representation of MobileNet V2 architecture [7].

# Sparse MobileNet Training and Hyper-parameters

University of CINCINNATI

- Used NOvA simulated files with a total of ~6.38 million $\nu_e, \nu_\mu$ and NC events of which 10% were cosmic events.
- Total sample was split into 95% training and 5% validation dataset.
- Each XZ and YZ view images were fluctuated by 10% to mimic the effect of systematic uncertainty on calibration.
- Hyper-parameters used during training:
    - SGD optimizer [8]
        - Learning Rate: 1e-2
        - Momentum: 0.9
    - Mish activation function [9]
    - ReducedLROnPlateau - learning rate scheduler
        - Reduces learning rate by factor after every chosen # of epochs the accuracy does not change.
    - Batch size of 128 and 40 epochs

# Sparse MobileNet Performance

University of CINCINNATI

- Dense MobileNet is comparable to Event CNN architecture. Dense MobileNet achieved 90.43% in validation accuracy.
- We proved that we are capable of reproducing NOvA Event CNN performance.
- Training with Sparse MobileNet achieved 88.42% in validation accuracy (lower magenta).
- Although Sparse MobileNet performed 2.01% worse than Dense MobileNet, the wall time was reduced by 10% and memory usage was reduced by 83%!

| PyTorch | Dense MobileNet | Sparse MobileNet |
|---|---|---|
| **Best Validation Accuracy (%)** | 90.43% | 88.42% |
| **Training Time (s) / Epoch** | 9844.00 s | 6814.00 s |
| **Validation Time (s) / Epoch** | 137.70 s | 123.00 s |
| **Memory Usage (GB)** | 2.98 GB | 0.50 GB |

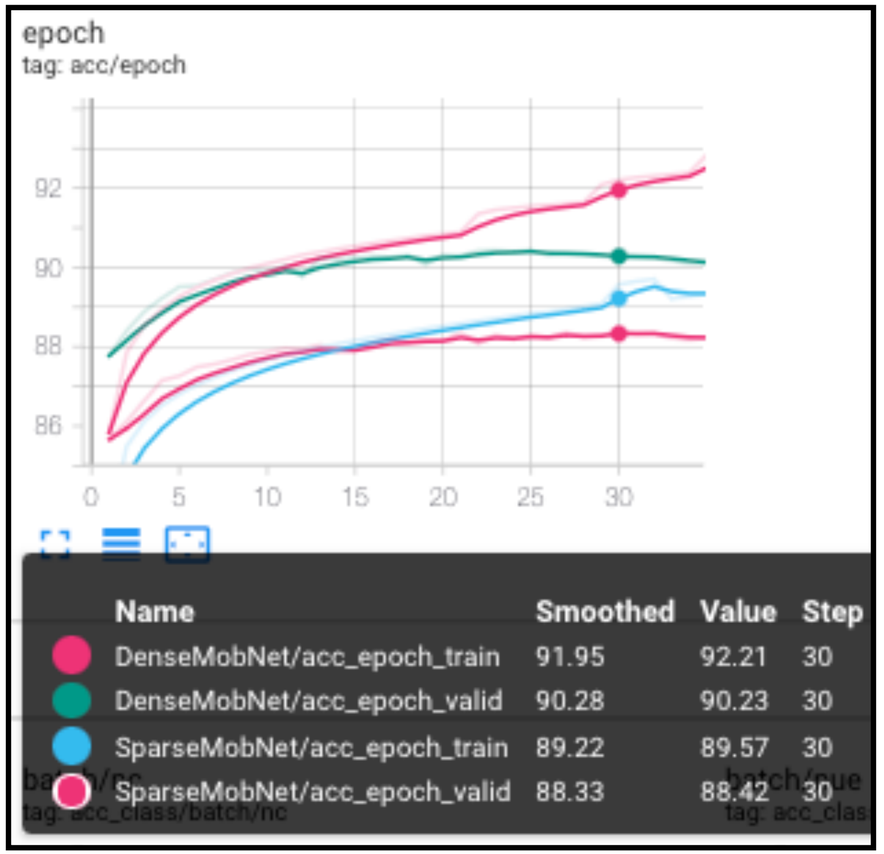



**Figure 10:** Validation and training accuracy of Dense and Sparse MobileNet.

# MinkowskiEngine (arXiv:1904.08755)

University of CINCINNATI

- MinkowskiEngine is an auto-differentiation library for sparse tensors.
- It supports all standard neural network layers such as convolution, pooling, unpooling, and broadcasting operations for sparse tensors.
- Instead of using *torch.nn* for dense tensors, replace it with *MinkowskiEngine* package vocabulary.

# Sparse MobileNet Training and Hyper-parameters

University of CINCINNATI

- Used NOvA simulated files with a total of ~6.38 million $\nu_e, \nu_\mu, \nu_\tau$ and NC events of which 10% were cosmic events.
- Total sample was split into 95% training and 5% validation dataset.
- Each XZ and YZ view CVN images were fluctuated by 10% to mimic the effect of systematic uncertainty on calibration.
- Hyper-parameters used during training:
  - SGD optimizer (arXiv:1609.04747)
    - Learning Rate: 1e-2
    - Momentum: 0.9
  - Mish activation function (arXiv:1908.08681)
  - ReducedLROnPlateau - learning rate scheduler
  - Batch size of 128 and 40 epochs

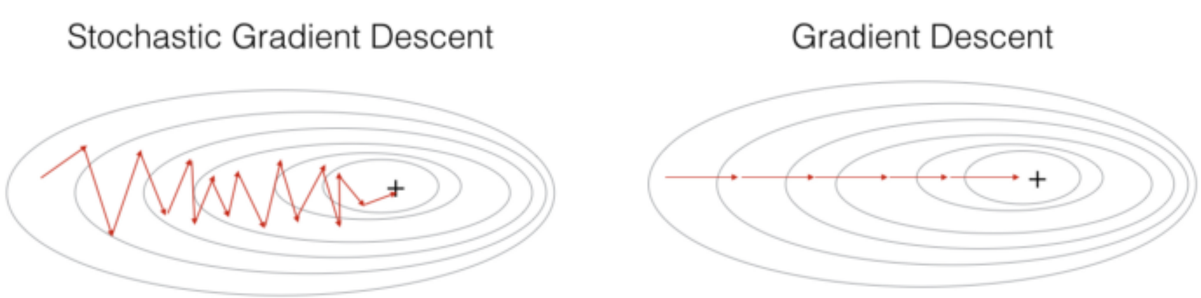



**Figure 26:** Comparison of SGD and GD's performance towards convergence [10, 11].

# Sparse MobileNet Training and Hyper-parameters

University of CINCINNATI

- Used NOvA simulated files with a total of ~6.38 million $\nu_e, \nu_\mu, \nu_\tau$ and NC events of which 10% were cosmic events.
- Total sample was split into 95% training and 5% validation dataset.
- Each XZ and YZ view CVN images were fluctuated by 10% to mimic the effect of systematic uncertainty on calibration.
- Hyper-parameters used during training:
  - SGD optimizer (arXiv:1609.04747)
    - Learning Rate: 1e-2
    - Momentum: 0.9
  - Mish activation function (arXiv:1908.08681)
  - ReducedLROnPlateau - learning rate scheduler
  - Batch size of 128 and 40 epochs

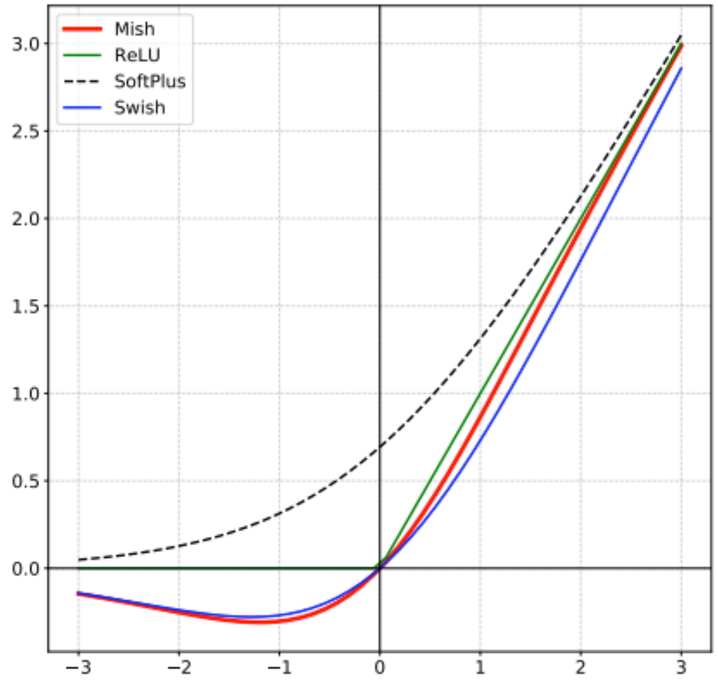



**Figure 27:** Comparison of activation functions Mish, ReLU, SoftPlus and Swish [12].

# Sparse MobileNet Training and Hyper-parameters

University of CINCINNATI

- Used NOvA simulated files with a total of ~6.38 million $\nu_e, \nu_\mu, \nu_\tau$ and NC events of which 10% were cosmic events.
- Total sample was split into 95% training and 5% validation dataset.
- Each XZ and YZ view CVN images were fluctuated by 10% to mimic the effect of systematic uncertainty on calibration.
- Hyper-parameters used during training:
  - SGD optimizer (arXiv:1609.04747)
    - Learning Rate: 1e-2
    - Momentum: 0.9
  - Mish activation function (arXiv:1908.08681)
  - ReducedLROnPlateau - learning rate scheduler
  - Batch size of 128 and 40 epochs

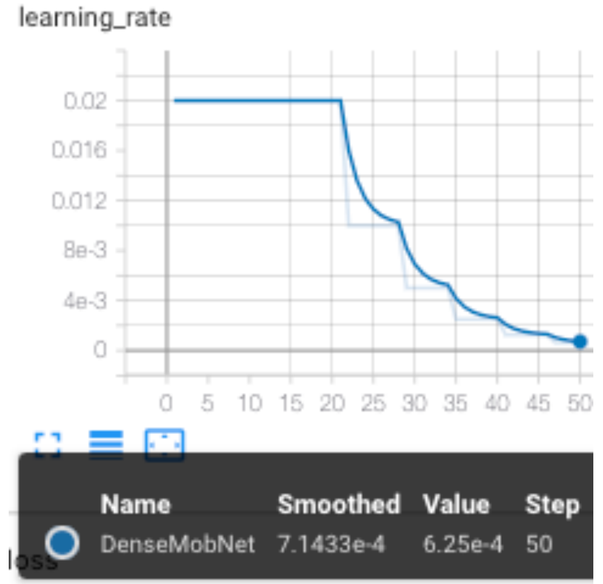



**Figure 28:** Learning rate scheduler reducing learning rate by a fraction if loss doesn't reduce within given number of epoch.

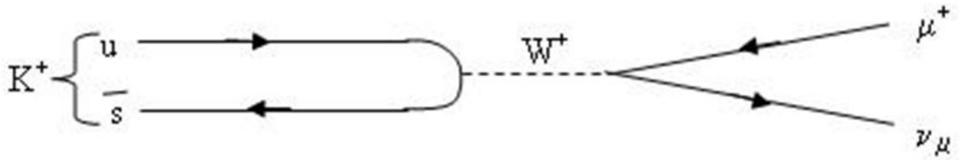

# Backup Slides: Beam Specific

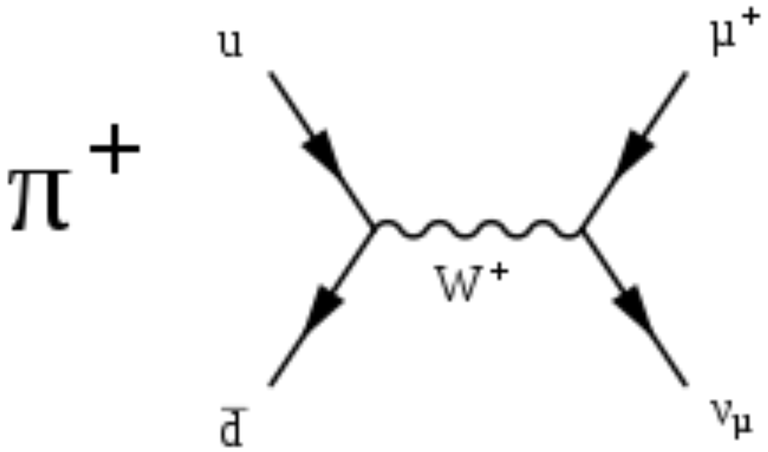
# NuMI Beam decay modes

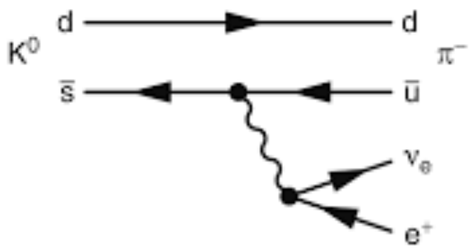
University of CINCINNATI

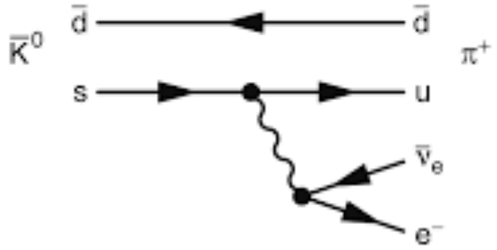
$$K^+ \longrightarrow \mu^+ + \nu_\mu$$

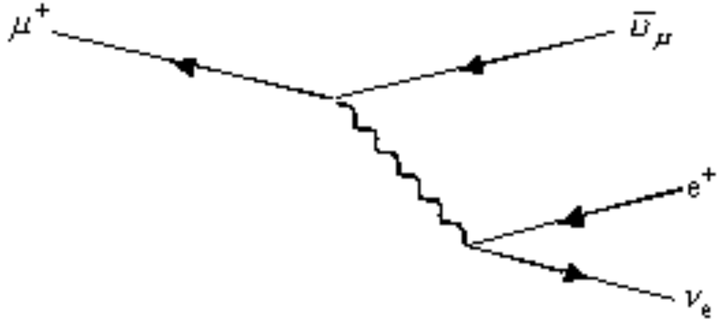
FEYNMAN DIAGRAM



# Off-Axis Beam

- Why do we have a narrow band beam from having off-axis?
    1. Relativistic kinematics of pion and kaon decays when boosted.
    2. Two body pion decay

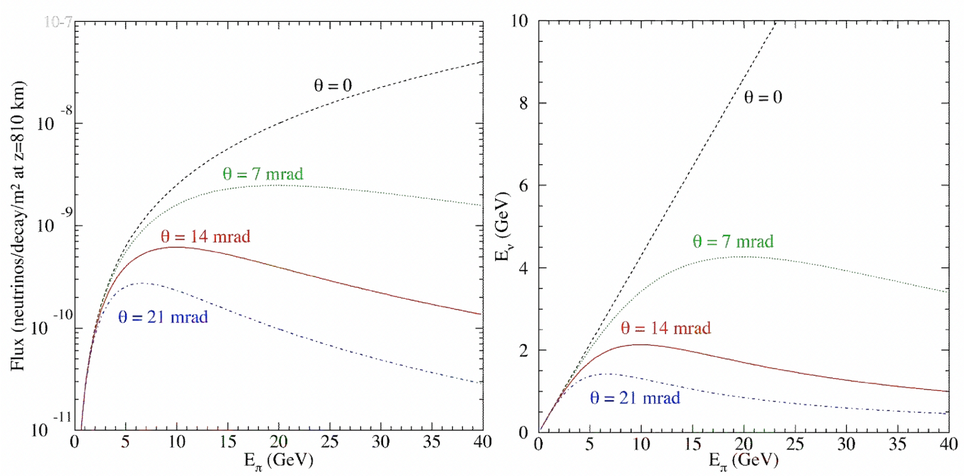



Fig. 2.2: Left: The neutrino flux from a pion of energy $E_\pi$ as viewed from a site located at an angle θ from the beam axis. The flux has been normalized to a distance of 800 km. Right: The energy of the neutrinos produced at an angle θ relative to the pion direction as a function of the pion energy.

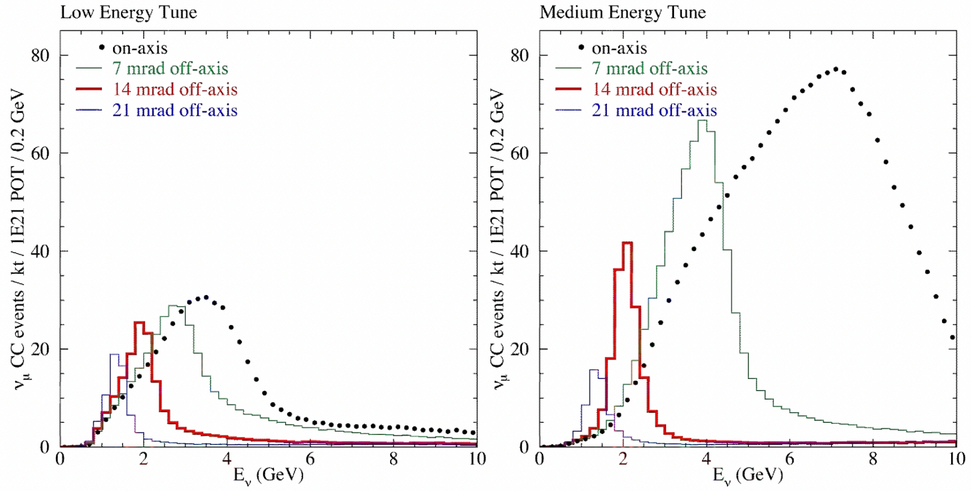



Fig. 2.3: Charged-current $\nu_\mu$ event rates prior to oscillations calculated for a distance of 810 km from Fermilab and at various off-axis locations in the NuMI beam. The spectra are for the NuMI low-energy (left) and medium-energy (right) configurations.


# Backup Slides: CNN

# Distribution Plots

- For Dense CNN, we looked at distribution plots of correctly and incorrectly classified NuMU and NUE events.
- Left top - incorrectly classified NuE
- Right top - correctly classified NuE
- Left bottom - incorrectly classified NuMu
- Right bottom - correctly classified NuMu