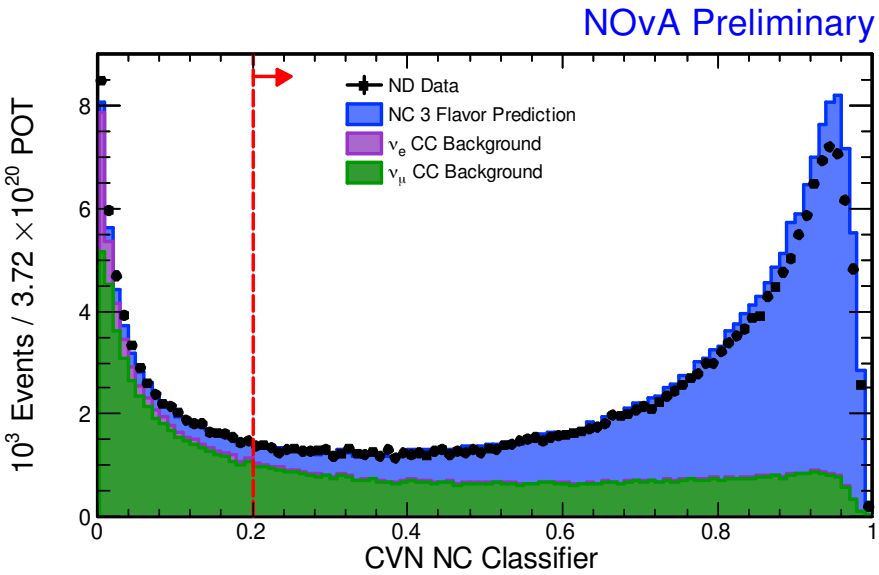
UNIVERSITY OF Cincinnati

Making Sense of Your Data: Statistics and Machine Learning

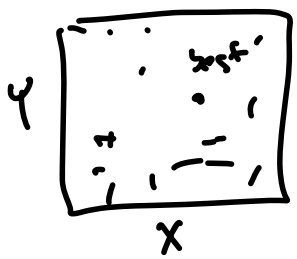
Adam Aurisano
University of Cincinnati

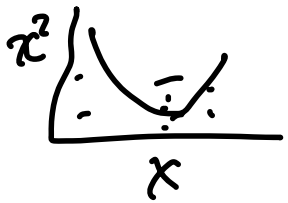
Understanding the Universe Through Neutrinos
1 May 2024

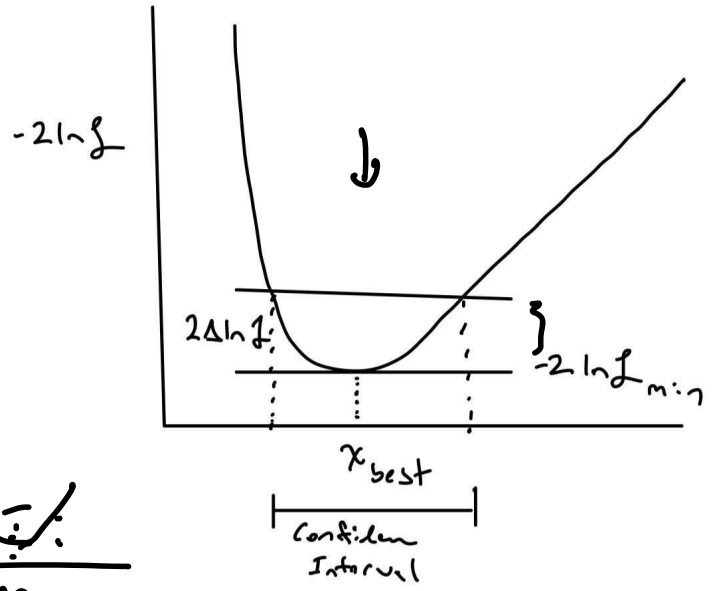


Wilks' Theorem

- Negative log likelihoods are asymptotically Gaussian
- Difference in -2lnL evaluated away from the best fit point and near the best fit point is χ^2 with k degrees of freedom
  - k = difference in number of free parameters between the point





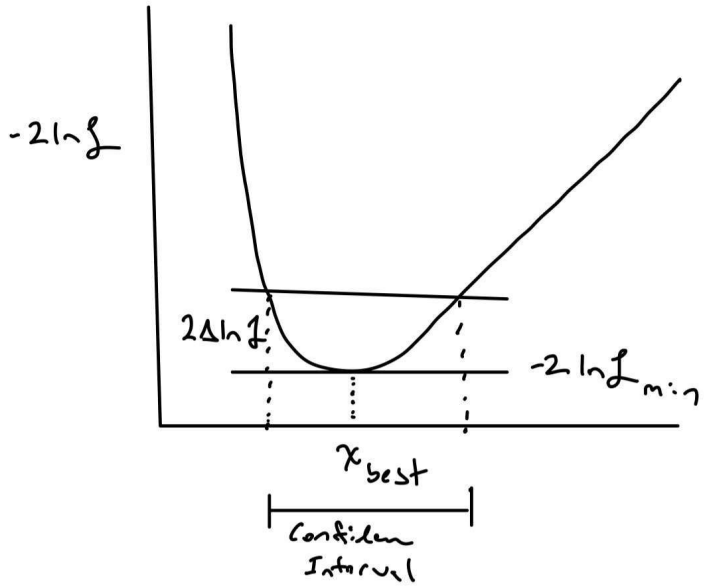


$(1-\alpha)$ (%)	$M=1$	$M=2$	$M=3$
68.27	1.00	2.30	3.53
90.	2.71	4.61	6.25
95.	3.84	5.99	7.82
95.45	4.00	6.18	8.03
99.	6.63	9.21	11.34
99.73	9.00	11.83	14.16

Wilks' Theorem

- This provides an easy way to generate confidence intervals
  - Find your best fit point
    - All fit parameters are allowed to float during that fit
  - Fit in a grid around the best fit point
    - All fit parameters except those on the grid are allowed to float
    - This is "profiling" over all nuisance parameters
  - All grid points with -2lnL < -2lnL$_{min}$ + up value are inside your confidence interval

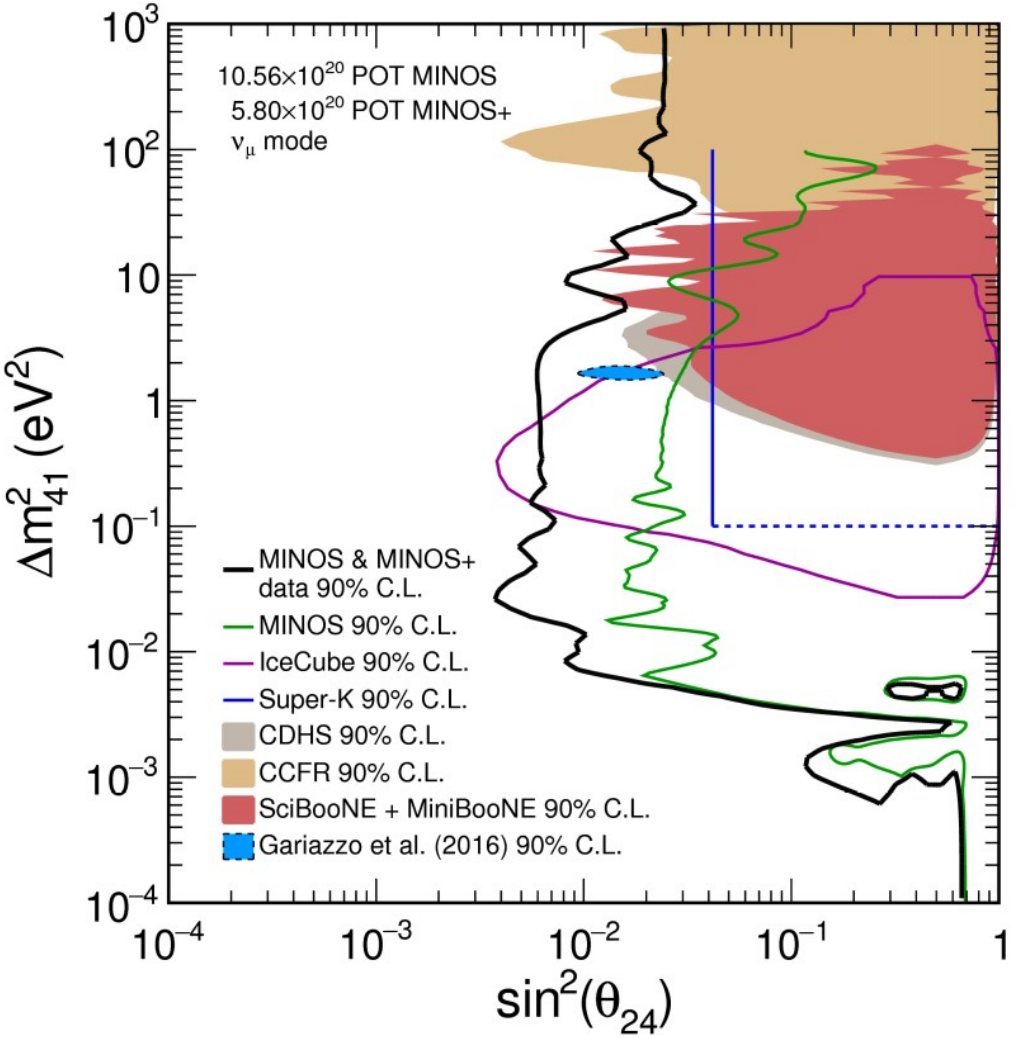
up = TMath::ChisquareQuantile(cl, dof)



$(1-\alpha)$ (%)	$M=1$	$M=2$	$M=3$
68.27	1.00	2.30	3.53
90.	2.71	4.61	6.25
95.	3.84	5.99	7.82
95.45	4.00	6.18	8.03
99.	6.63	9.21	11.34
99.73	9.00	11.83	14.16

Wilks's Theorem

- Wilks's Theorem confidence intervals are the starting point for almost all contour plots like this
- Wilks's has some notable problems
  - It is only true asymptotically
    - If you think you need a Poisson likelihood, Wilks's is only an approximation
  - Is not valid if true value is on boundary of parameter space
- To do better than this, Monte Carlo methods are necessary

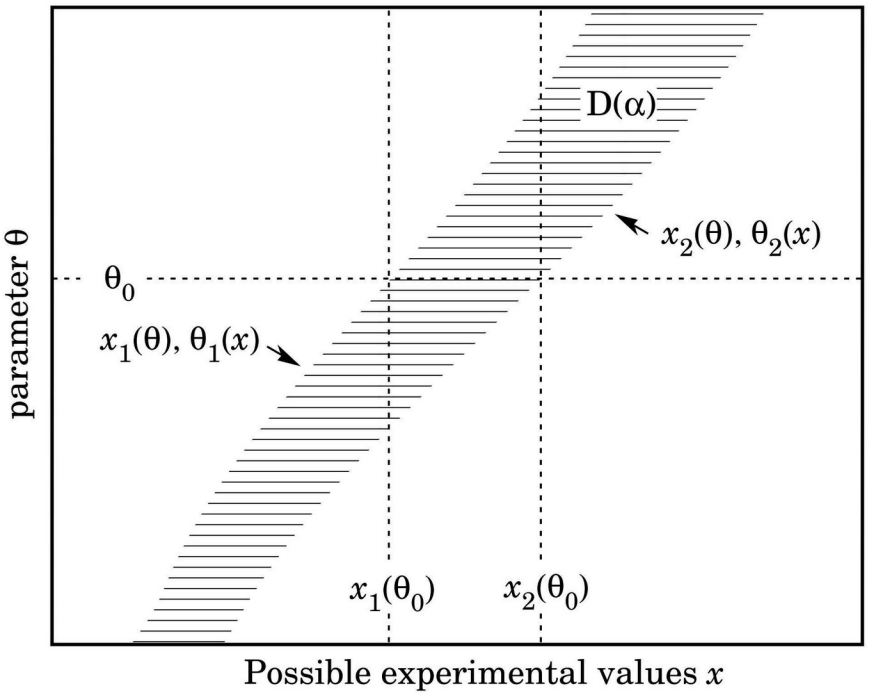


MINOS+ sterile limit

Feldman-Cousins

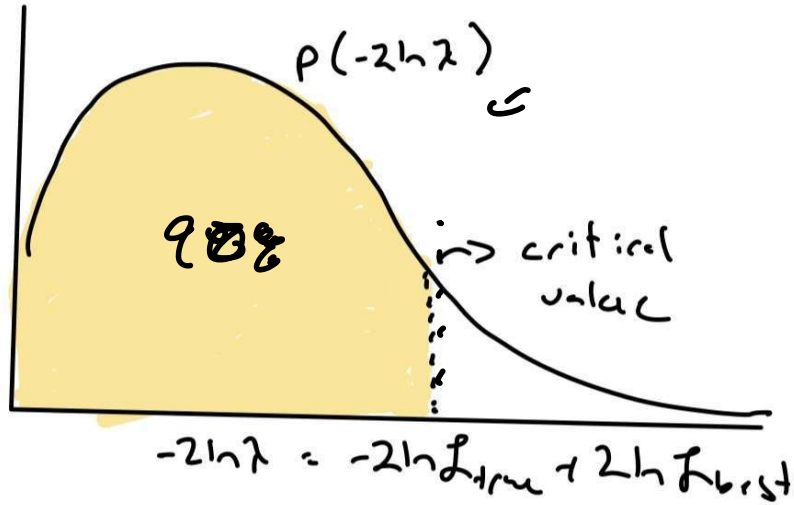
Phys. Rev. D57 (1998) 3873

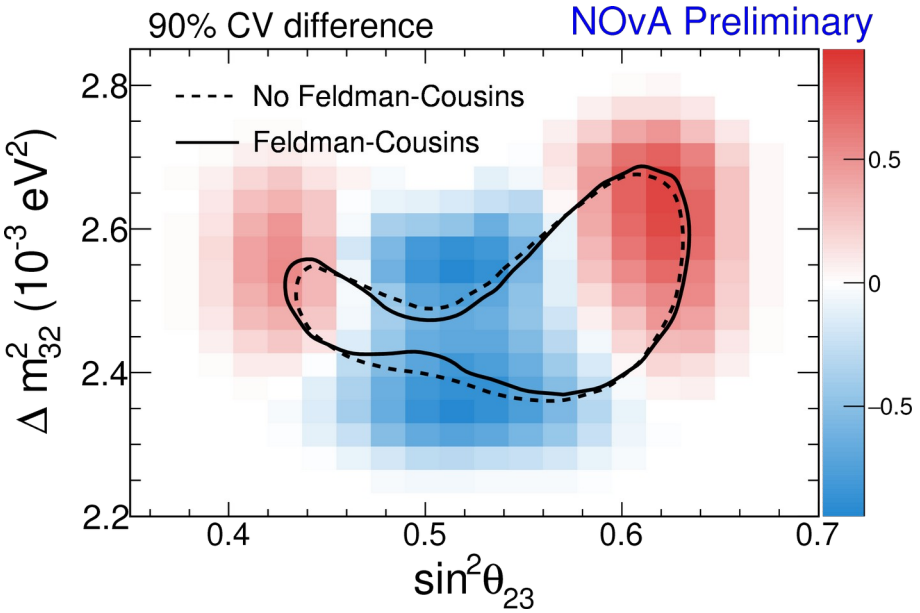
- The Neyman construction requires us to construct a confidence belt composed of segments that contain the appropriate amount of probability
  - Problem: we can build these intervals anywhere
    - Upper limits
    - Lower limits
    - Central intervals
- Feldman and Cousins observed that if the type of interval constructed depends on the data observed, we get problems in coverage
  - "Flip flopping"



Feldman-Cousins

- Solution: use an ordering principle based on a likelihood ratio to determine if a bin should be included or excluded from the contour
- For each bin, pseudoexperiments using the parameters of that bin as the truth
  - Fit all parameters to find the best fit
  - Compute the -2lnL using the true parameters
  - Construct the $-2\ln L_{true} + 2\ln L_{best}$
  - Repeat many times to build empirical probability distribution of the ordering principle statistic
  - Determine up value for that bin from distribution



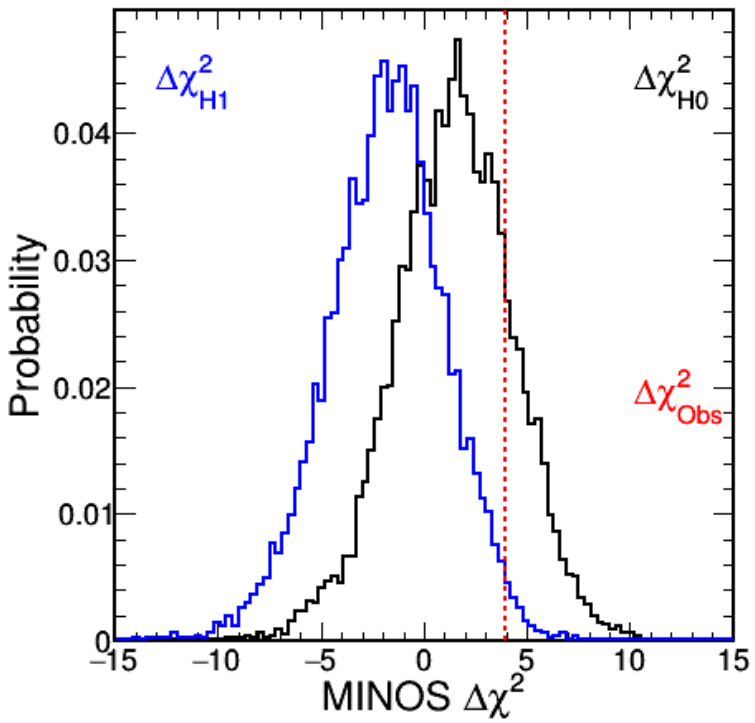


Feldman-Cousins

- FC intervals are computationally intensive to construct
- The best fit has to be done for each pseudoexperiment
  - This is typically time consuming since many parameters are free in your fit
- Exact coverage guarantees are only valid in the absence of nuisance parameters
  - Coverage is only guaranteed over the ensemble of pseudoexperiments
    - Can be especially problematic for nuisance parameters like mass ordering
    - NOvA explored ways to improve coverage in arXiv:2207.14353
    - Other options exist, but this is not a settled subject

CLs

- CLs is modification of the standard p-value approach for setting limits
  - Trying to solve problem of inappropriately excluding points in parameter space to which the experiment is known to lack sensitivity
- Note: in formation of figure of merit - 2lnQ, systematic uncertainties must be fluctuated!
  - It is not enough to include systematics as pull terms in the fit itself for this method



$$-2\ln Q = -2\ln\left(\frac{\mathcal{L}(\text{data}|H_1,\hat{\theta})}{\mathcal{L}(\text{data}|H_0,\hat{\hat{\theta}})}\right)$$

CL_S

$$-2\ln Q = -2\ln\left(\frac{\mathcal{L}(\text{data}|H_1,\hat{\theta})}{\mathcal{L}(\text{data}|H_0,\hat{\hat{\theta}})}\right)$$

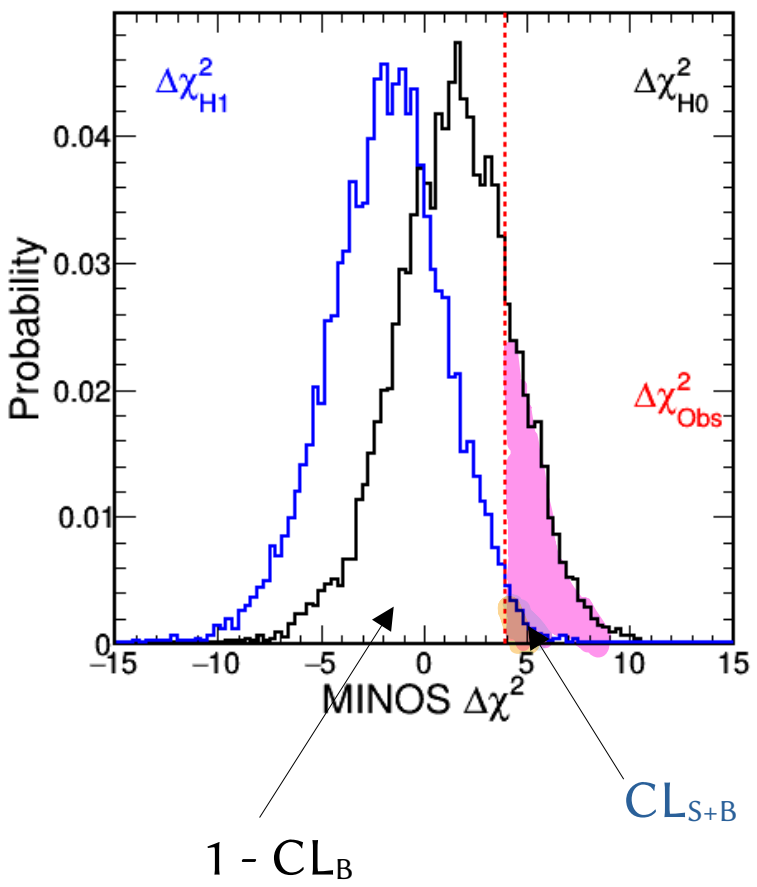
$$1 - CL_B = P(-2\ln Q \leq -2\ln Q_{obs}|H_0)$$

$$CL_{S+B} = P(-2\ln Q \geq -2\ln Q_{obs}|H_1)$$

$1 - CL_B$ = regular discover p-value

CL_{S+B} = measure of sensitivity (large ⟶ more sensitive)

$$CL_S = \frac{CL_{S+B}}{CL_B}$$



Exclude point at α CL if CL_S < 1 - α

Machine Learning

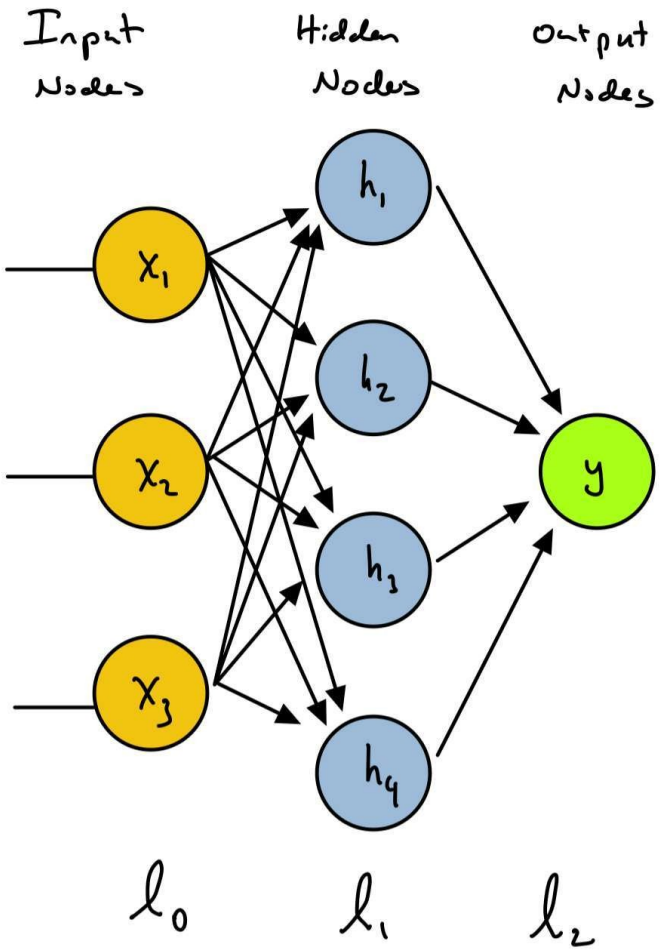
Non-Linear Test Statistics

- Neyman-Pearson lemma: highest power classifier is defined by the behavior of a likelihood ratio

$$\frac{g(\vec{t}|H_0)}{g(\vec{t}|H_1)} > c$$

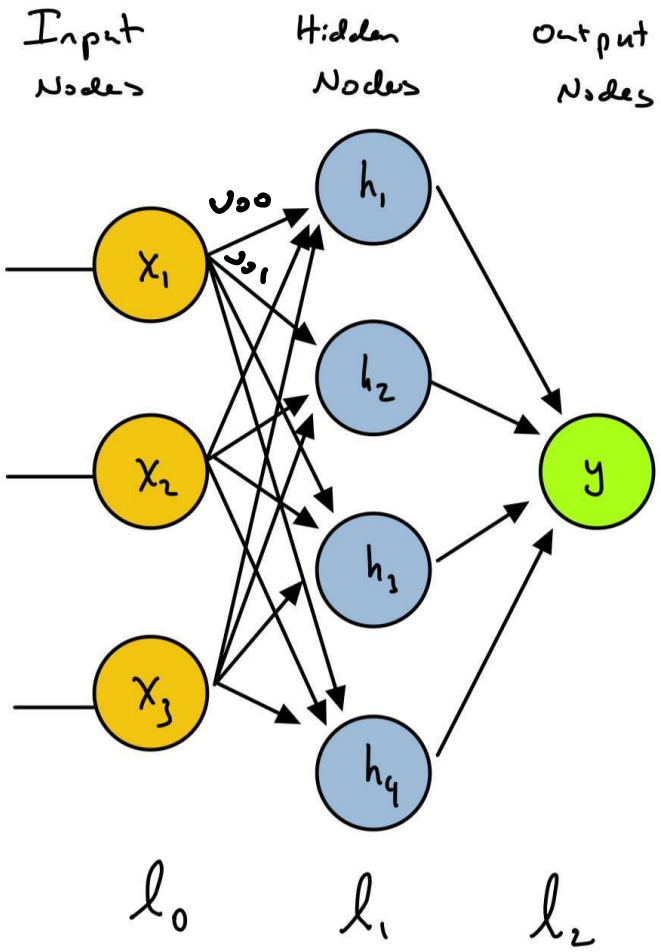
- We can often build a multidimensional likelihood to use in this construction empirically using multidimensional histograms
  - For more than three features, this becomes difficult
    - Amount of memory and appropriate statistics to fill all bins becomes untenable
- Making a series of 1D likelihoods is possible, but it loses potential power due to correlations between features
- We need a way to generate optimal test statistics that can model non-linear relationships

Multi-Layer Perceptrons



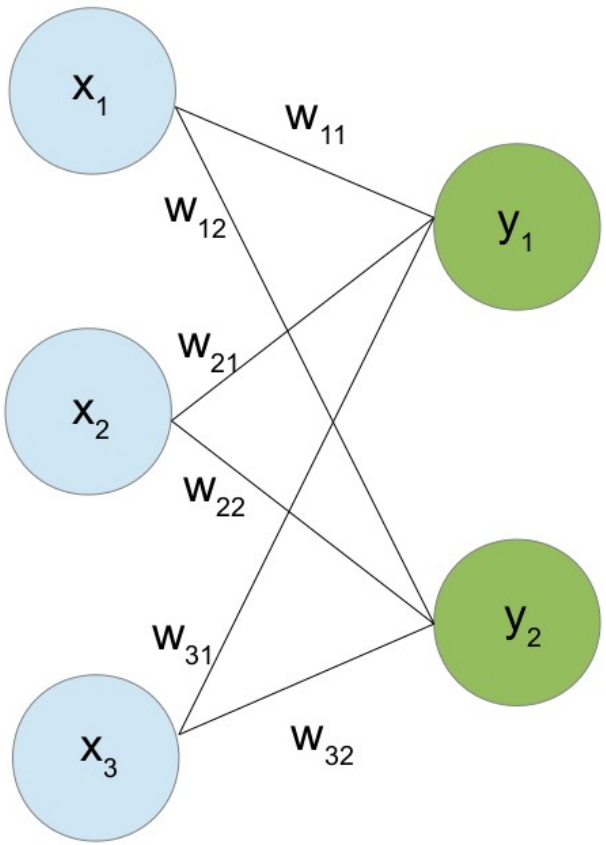
- The multi-layer perceptron (MLP) dates back to 1958
- Goal: computational algorithm capable of approximating generic functions
- Process can be used to learn a test statistic
  - Inputs: reconstructed features
  - Output: 1 if signal, 0 if background

Multi-Layer Perceptrons



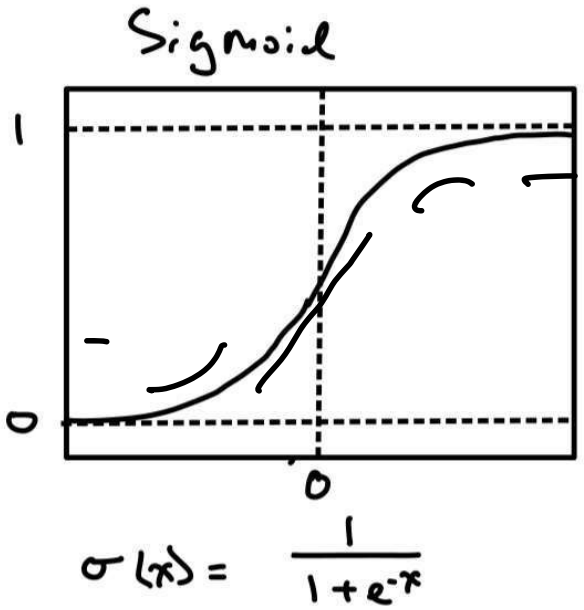
- Constructed of nodes organized in layers
- Each node performs a transformation on a weighted sum of the output of all nodes in the previous layer
- Pick the best possible weights using an iterative learning procedure

Simplified Example



- Multiple stacked layers like this is called a multilayer perceptron
- This is a simplified example with just two layers
  - Simplest network usually has three
    - Inputs
    - Hidden layer
    - Outputs

$$y_1 = \sigma(w_{11}*x_1 + w_{21}*x_2 + w_{31}*x_3 + b_1)$$
$$y_2 = \sigma(w_{12}*x_1 + w_{22}*x_2 + w_{32}*x_3 + b_2)$$



Loss Function

- Need some way to determine if the network is doing a good job
- Loss function compares the truth of some data fed to the network with the predictions of the network
- The set of weights for which the loss is the smallest is the optimal set
- This is very closely related to previous ideas of -2lnL

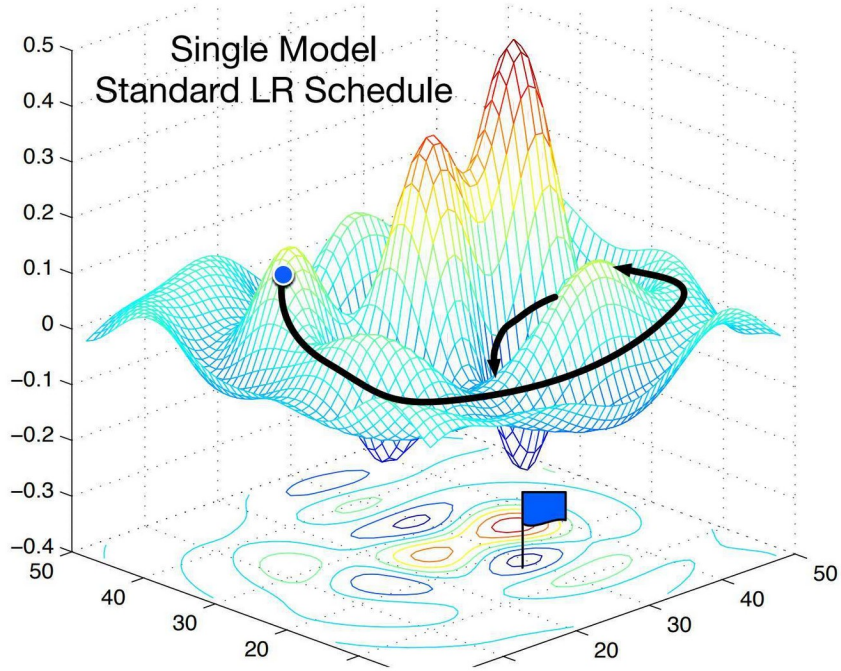
$$\mathcal{L}(\vec{x}, \vec{w}) = \frac{1}{N} \sum^{N} -y_i \log(f(x_i)) - (1 - y_i) \log(1 - f(x_i))$$

Binary cross-entropy: A measure of how many bits it would take to encode the true probability distribution y in terms of the reconstructed probability distribution f(x).

Commonly used when separating examples into two categories.

Training

- Loss function is super high dimensional surface
  - Each weight is a dimension
  - Networks can have millions or billions of weights
- The gradient of the loss with respect to the weights tells us the direction of greatest change
- Best thing to do is to walk the opposite direction of the gradient
- This explores the local region, so we have to do this iteratively, recomputing the gradient at each new point
- Compute gradient for a minibatch of examples
  - Stochastic gradient descent
  - Noise from low statistics helps bounce the minimizer out of shallow wells



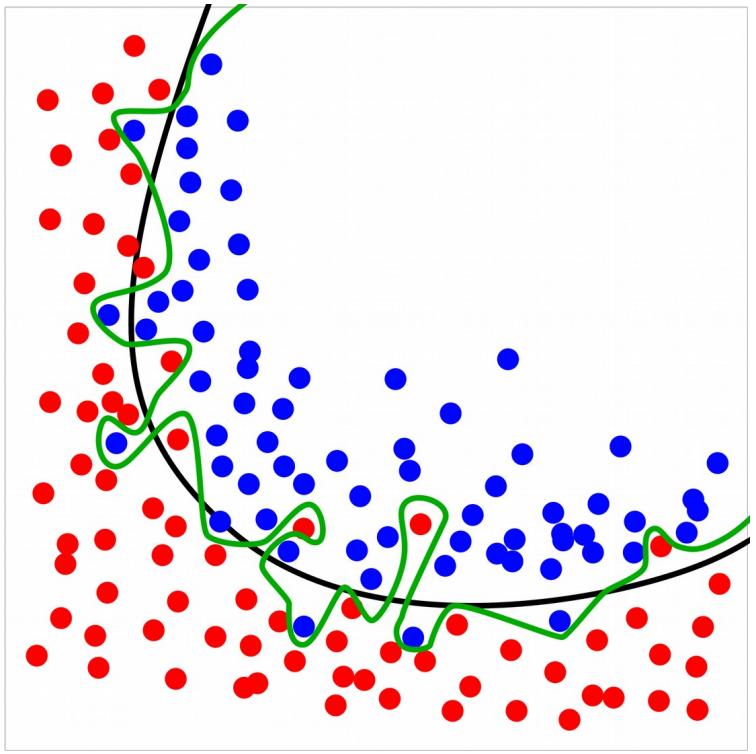
arXiv:1704.00109

$$\vec{w}' = \vec{w} - \eta \vec{\nabla}_w \mathcal{L}$$

Learning rate: size of step in weight space

Overtraining

- Sometimes called a failure to generalize
  - Networks contain large number of parameters
  - Can lead to algorithms learning precise details of the training set at the expense of generalizing well to new data
  - Can be see if the response to a held-out sample of testing data begins to diverge from response on training data
- Regularization is commonly used to reduce overtraining
- Add a term to the loss to penalize the complexity of the decision surface



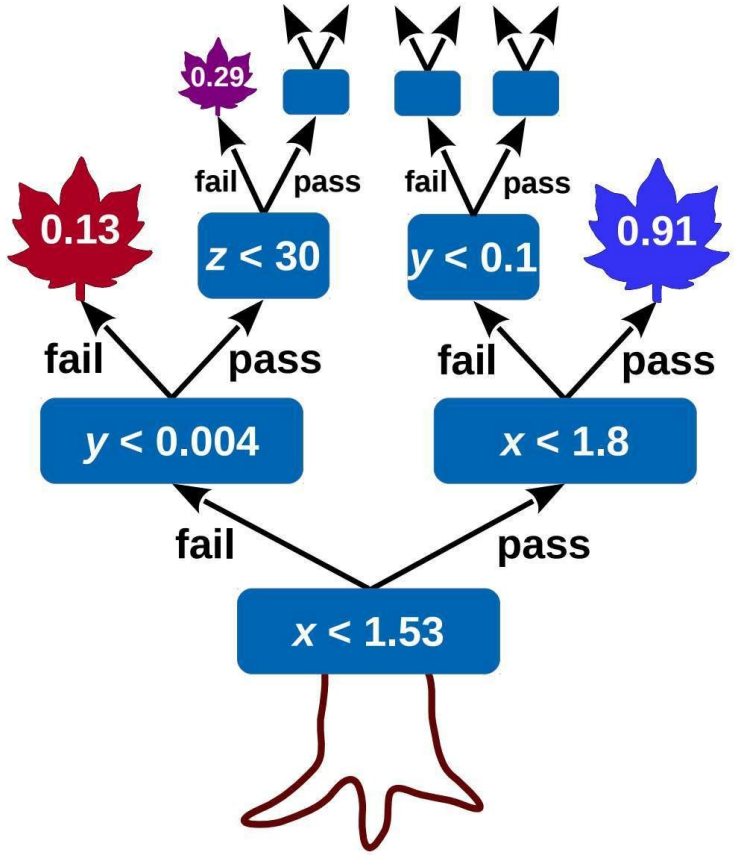
$$\mathcal{L}_{reg} = \frac{1}{2}\lambda \sum w_i^2$$

Why Isn't this Good Enough?

- It can be proven that a single layer network with a sufficient number of nodes can approximate most functions to arbitrary precision
- Multi-layer networks can often approximate a function with fewer nodes than a single layer network
  - These networks become very difficult to train
  - Performance was generally disappointing, leading to them falling out of favor
- Due to the fully connected nature of MLPs, the number of free parameters increases sharply with additional nodes

Decision Trees

- Decision trees are an automated version of how we normally do cut based analyses
- Algorithm randomly selects a feature
  - Tunes a cut to maximize the information gain
- Continue building the tree until there is no information gain or until we reach a limit on maximum depth



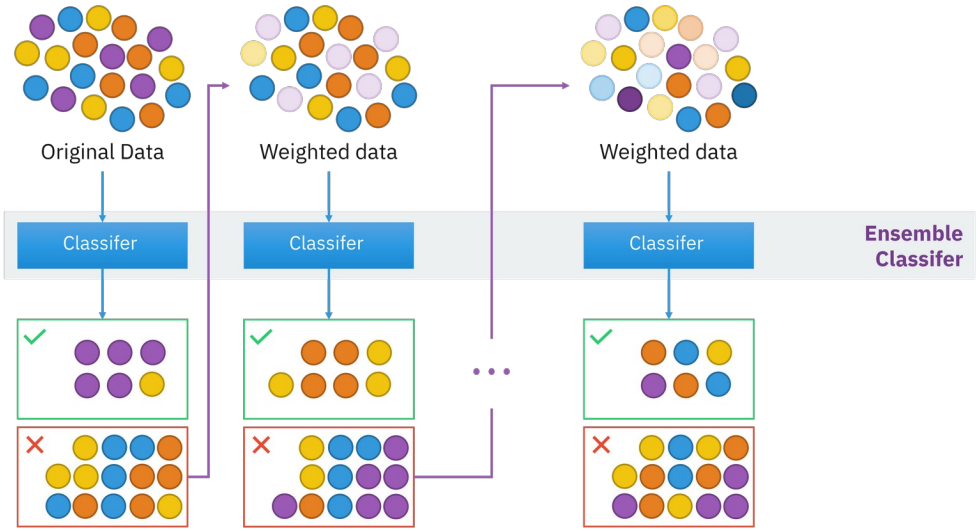
arXiv:2206.09645

Decision Trees

- Decision trees effectively are making a series of hyperplane decision surfaces
- Pros
  - Human understandable
  - Invariant under scale changes
  - Can handle discrete inputs
  - Robust against outliers
- Cons
  - Performance is weak (only a little better than random chance)
  - If decision trees become too deep, they tend to overtrain

Boosting

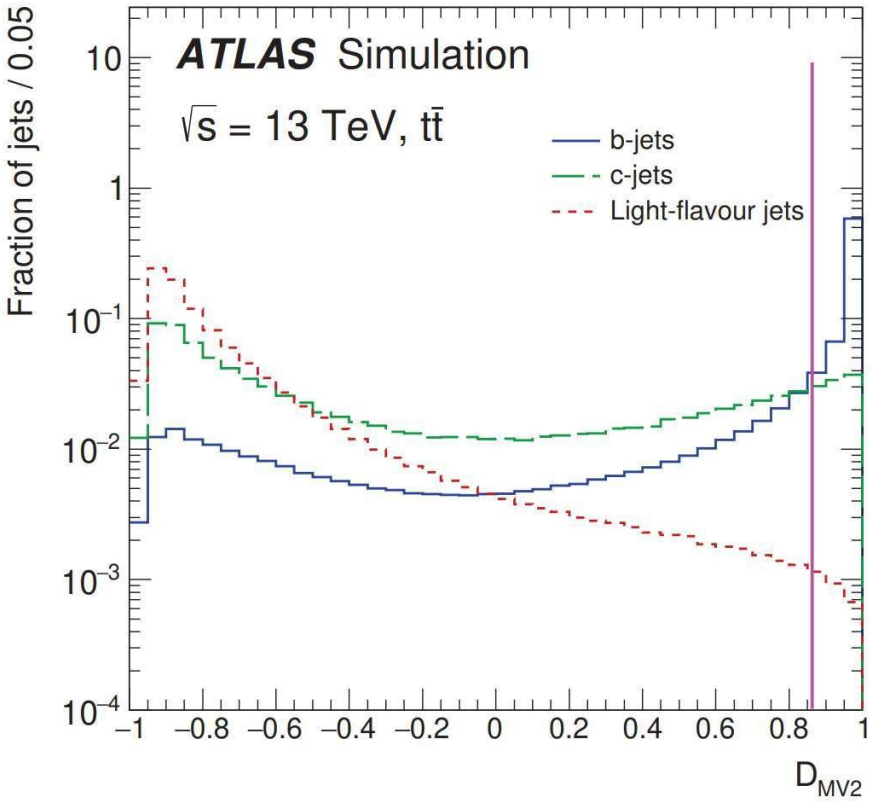
- Boosting is a method for converting weak learners into strong learners
  - Generate a random tree
  - Classify all examples based on this tree
  - Weight down correctly classified examples, weight up incorrectly classified examples
  - Generate a new tree
  - Repeat
- This process generates an ensemble of decision trees which are optimized on different populations of examples



wikipedia

BDT Performance

- BDTs tend to work very well when operating on a relatively small list of reconstructed features
- Tend to be resistant to pathologies due to including irrelevant features
- Became the gold standard in HEP for a long time



arXiv:2206.09645

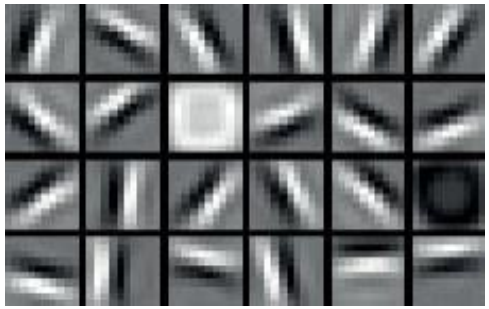
Inductive Bias

- MLPs are theoretically arbitrarily powerful
- BDTs are composed of weak elements
- However, BDTs tend to work much better than MLPs
- Why?
  - MLPs can only work if you can successfully train it
  - Instead of having a high degree of freedom, a more limited method whose structure better matches the type of problem tends to learn more efficiently
- Many analyses use cut-based strategies because analysis features are chosen for their ability to separate populations
  - BDTs amplify this ability by making many piecewise linear hyperplane decision surface
- Lesson: matching the inductive bias of a given problem is very helpful

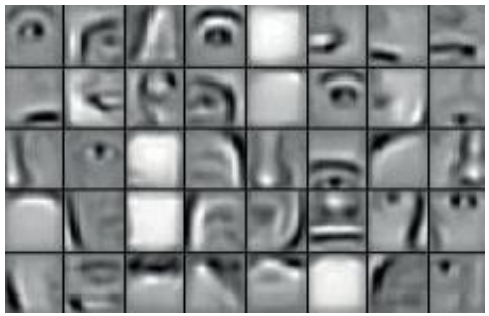
Deep Learning

Raw input

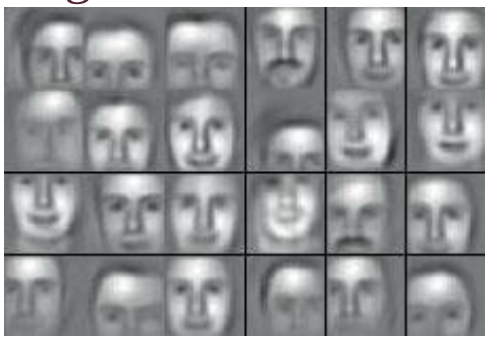
Low level features



Mid level features



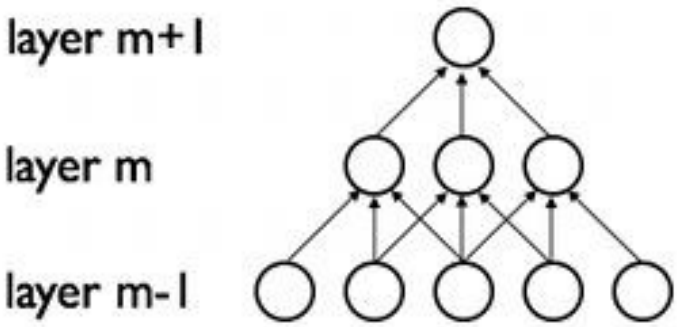
High level features



developer.nvidia.com/deep-learning-courses

- Deep learning revolution (2012) renewed interest in neural networks
- Use sparsely connected neurons to allow for many hidden layers.
- Deep structure extracts increasingly complex features from raw or nearly raw input data.

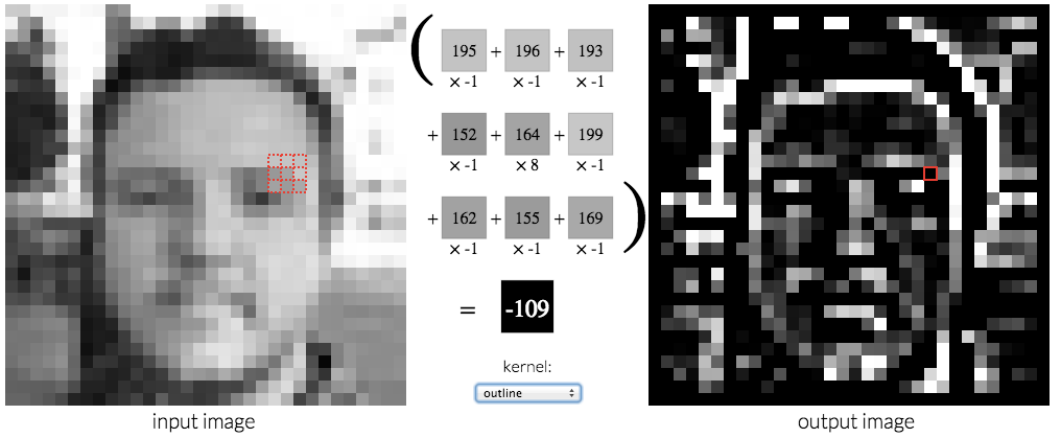
Convolutional Neural Networks



- Convolutional neural nets are a very successful deep learning method.
- Inspired by research showing that the cells in the visual cortex are only responsive to small portions of the visual field - “receptive field”.
- Some cells collect information from small patches – sensitive to edge-like features.
- Other cells collect information from large patches.
- Effectively, these cells are applying convolutional kernels across the visual field.

http://deeplearning.net/tutorial/lenet.html

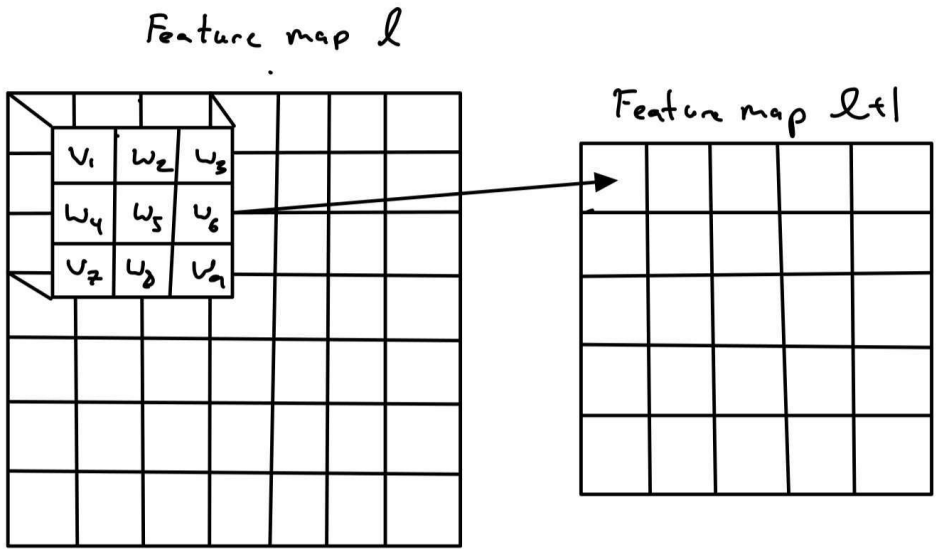
Convolutional Kernels



- Convolutional kernels are well known in computer graphics.
- Kernels transform images.
  - The one above outlines objects in the image.
- Many common kernels exist, but it we want to learn optimal kernels directly from the data.

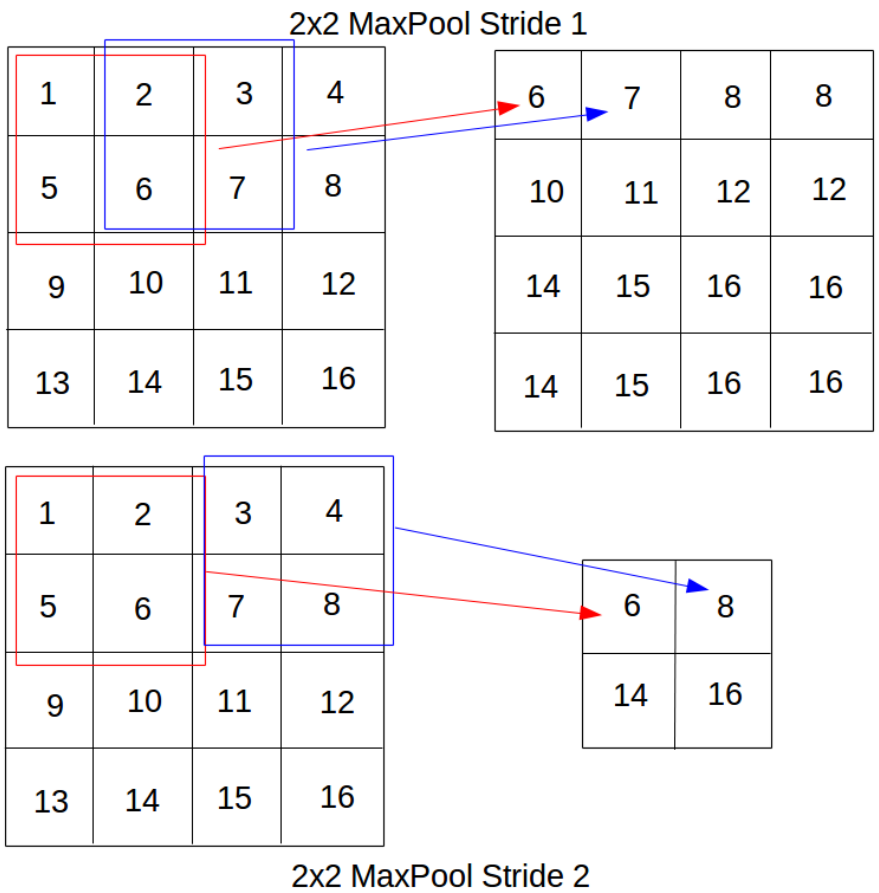
http://setosa.io/ev/image-kernels/

Convolutional Layers



- Each kernel we create stays the same as we apply it across the image.
  - Weight sharing reduces the number of free parameters
- Each convolutional layer trains an array of kernels which produce corresponding feature maps.
- Weights going from layer to the next are a 4D tensor of NxMxHxW
  - N is number of incoming feature maps
  - M is the number of outgoing feature maps
  - H and W are the height and width of the outgoing convolutional kernels.
- The next layer applies kernels to combine the information in a receptive field across feature maps in the previous layer to create new feature maps.

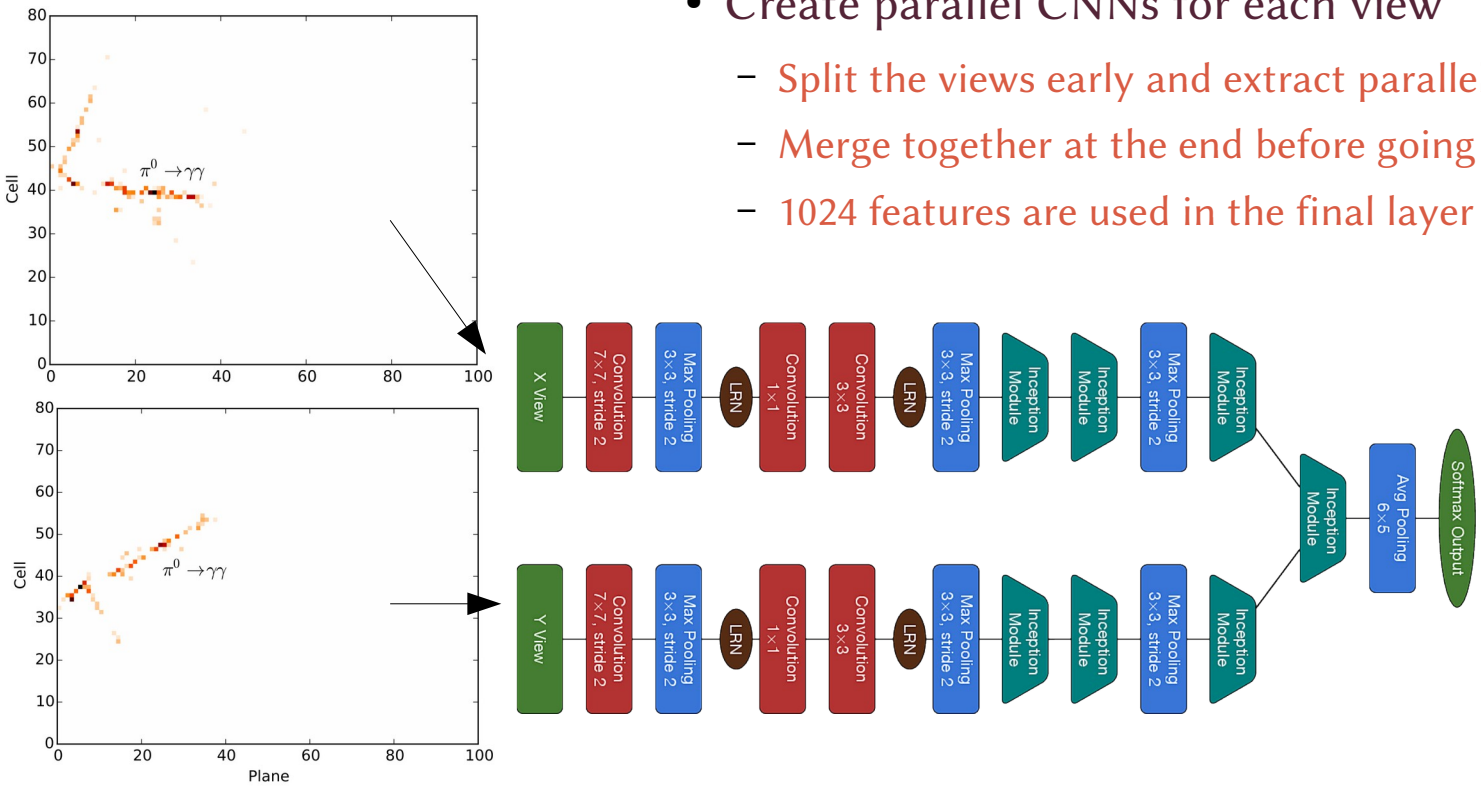
Pooling



- Pooling is a technique to downsample information
  - Output pixel is either maximum value of a patch of input pixels (max pooling) or the average (average pooling)
- Can be thought of as a type of smoothing to remove less significant information
- Adds some degree of equivariance to translations
  - Translations in input produce same translation in output

NOvA: Convolutional Visual Network

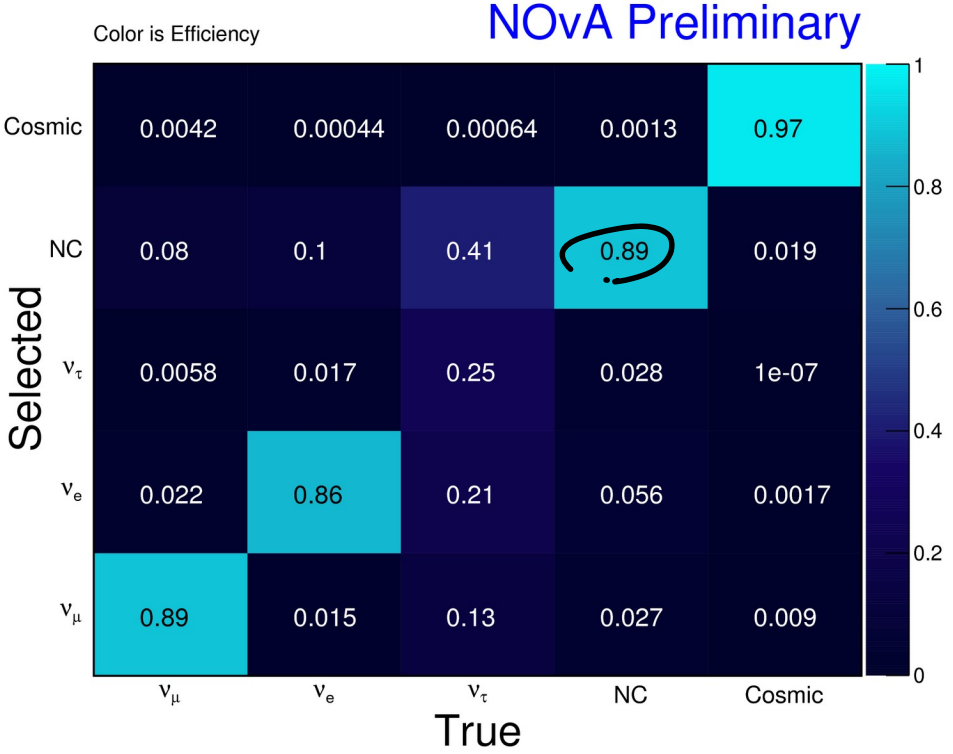
- Detector is composed of alternating horizontal and vertical planes
  - Two views of the event- one from top and one from side
  - Resulting pixel maps are sparse
- Create parallel CNNs for each view
  - Split the views early and extract parallel features
  - Merge together at the end before going through fully connected layers
  - 1024 features are used in the final layer for classification.



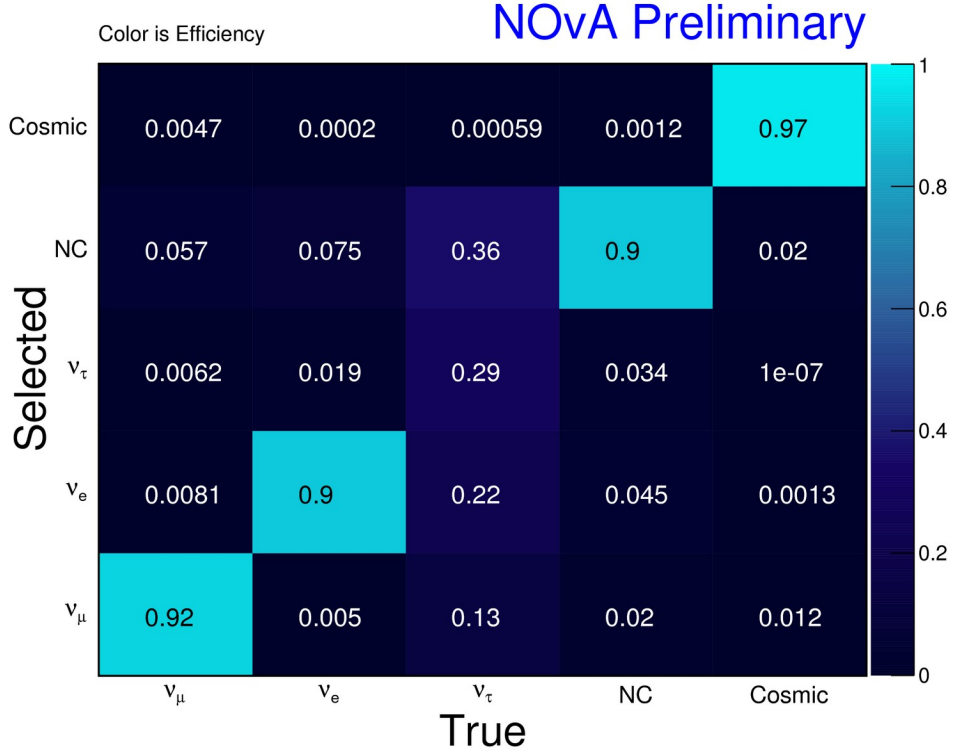
- The architecture is a multi-classifier
  - Neutrino flavor
  - Type of interaction with the nucleus
- In principle, this architecture is a universal neutrino classifier

Performance

Neutrino mode



Antineutrino mode

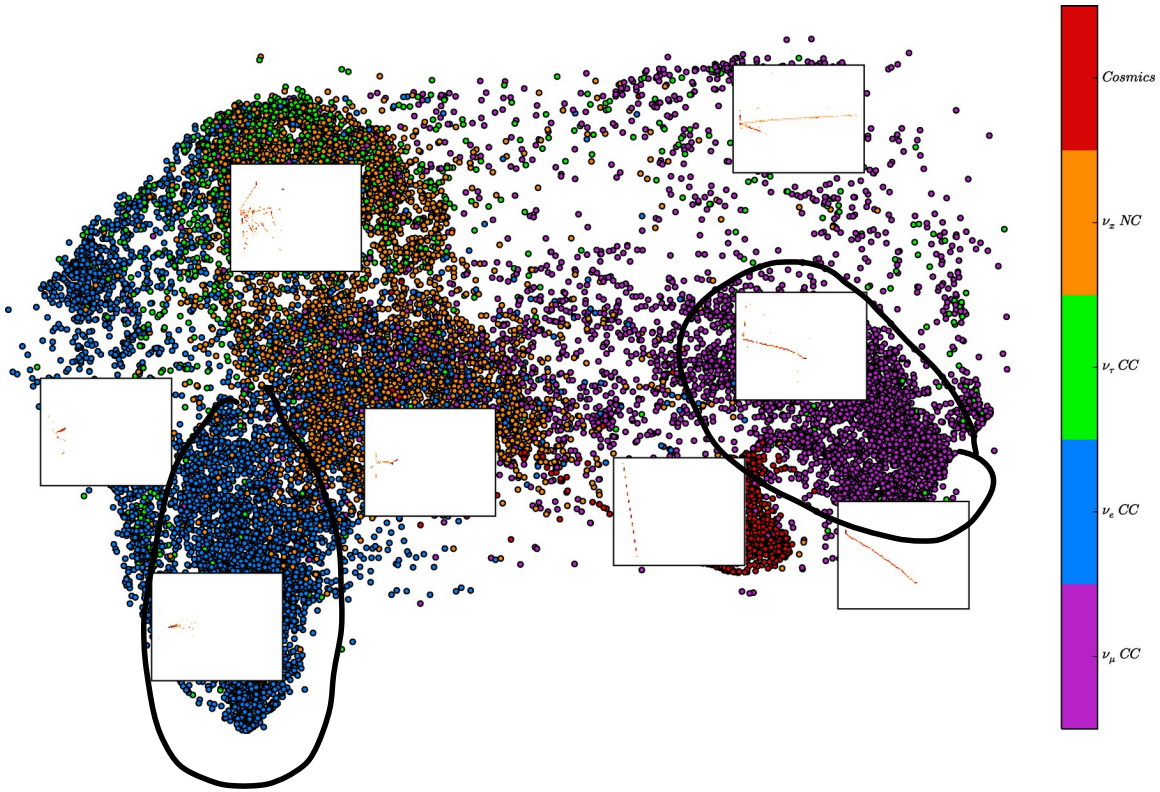


Performance is often displayed in “confusion matrices”
Efficiency: Fraction of true examples correctly selected
Purity: Fraction of selected examples that are correctly selected

Understanding Performance: t-SNE

Hadronic shower dominated

Beam-directed tracks

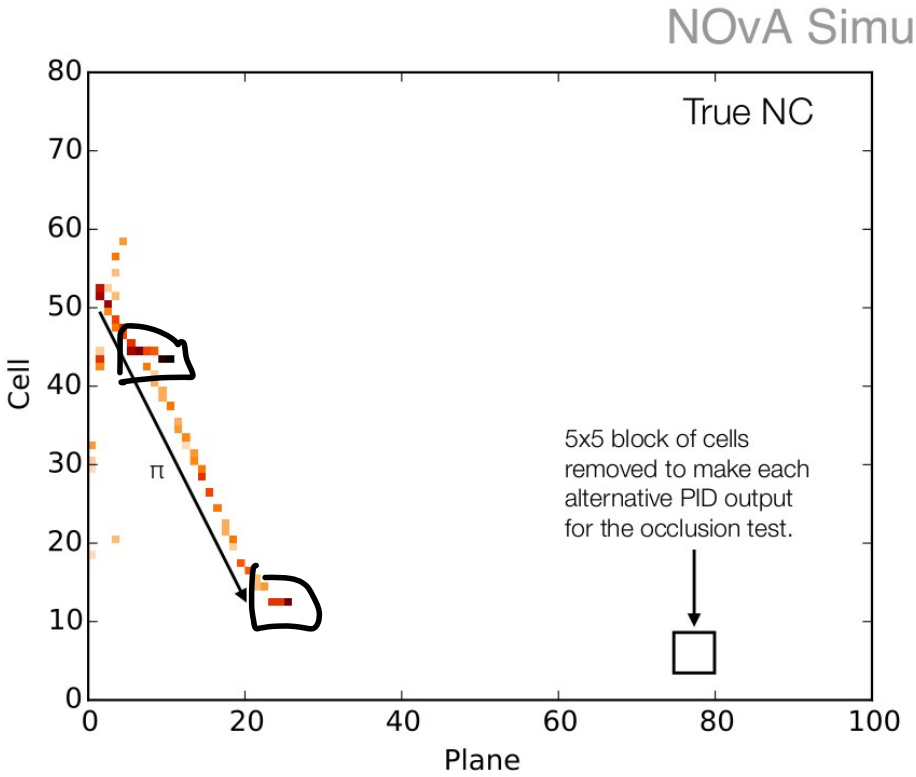


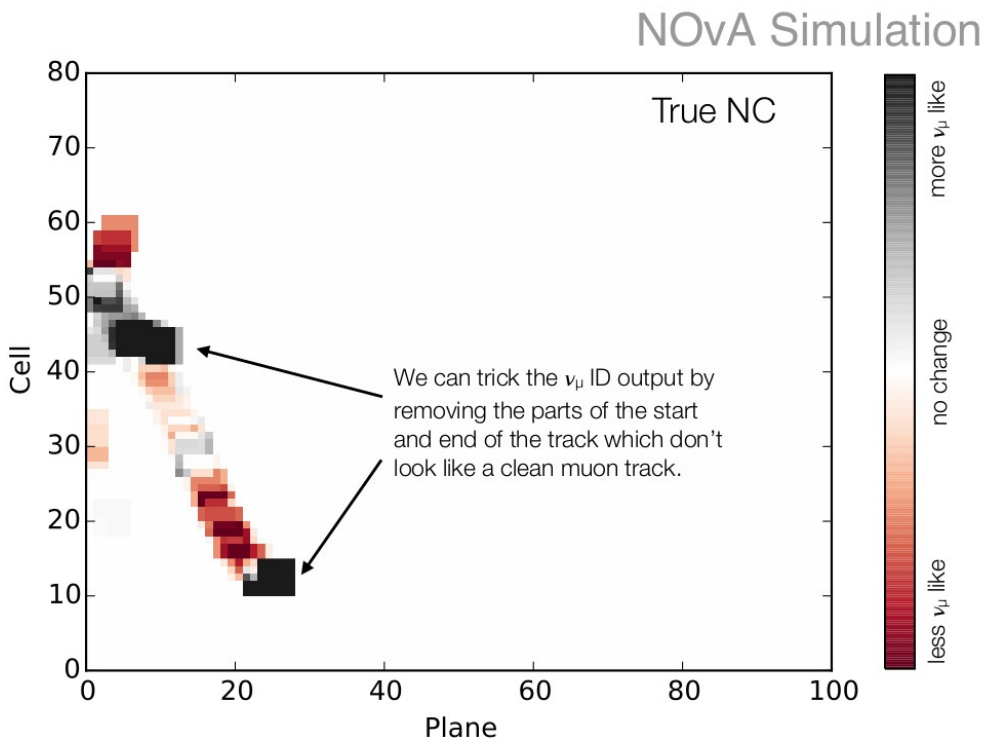
EM shower dominated

Steep-angle tracks

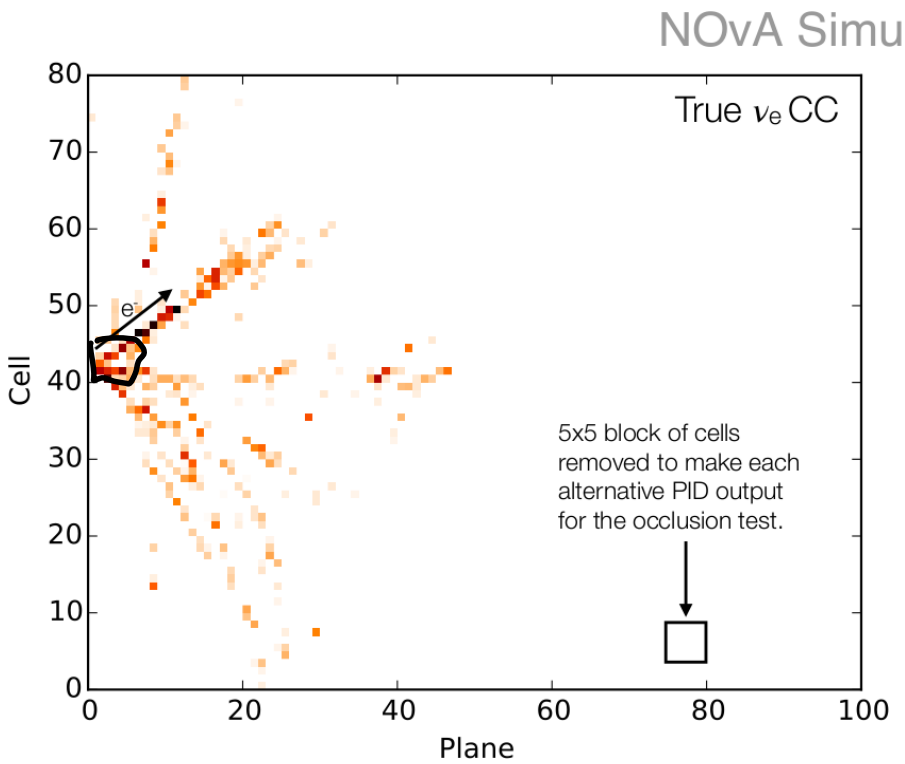
- Final layer of CNN often contains many output features
  - 1024 for NOvA's network
- t-distributed stochastic neighbor embedding is a technique for placing points from a high dimensional space into a low dimensional space based on similarity
- Types of events that cluster together help us understand how the network is making decisions

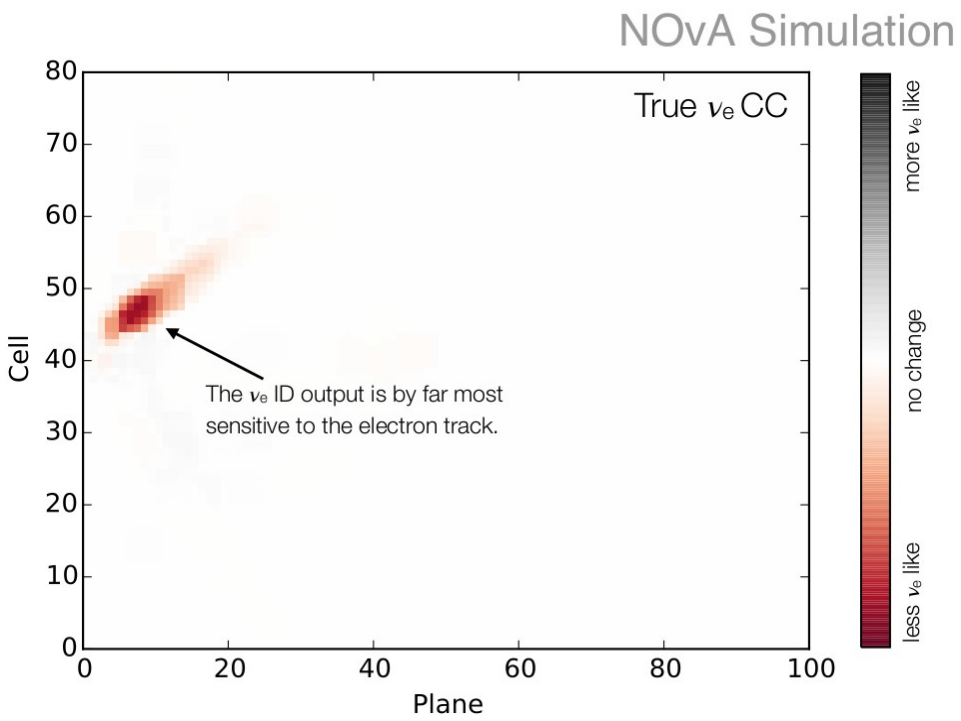
Understanding Performance: Occlusion Tests





Understanding Performance: Occlusion Tests

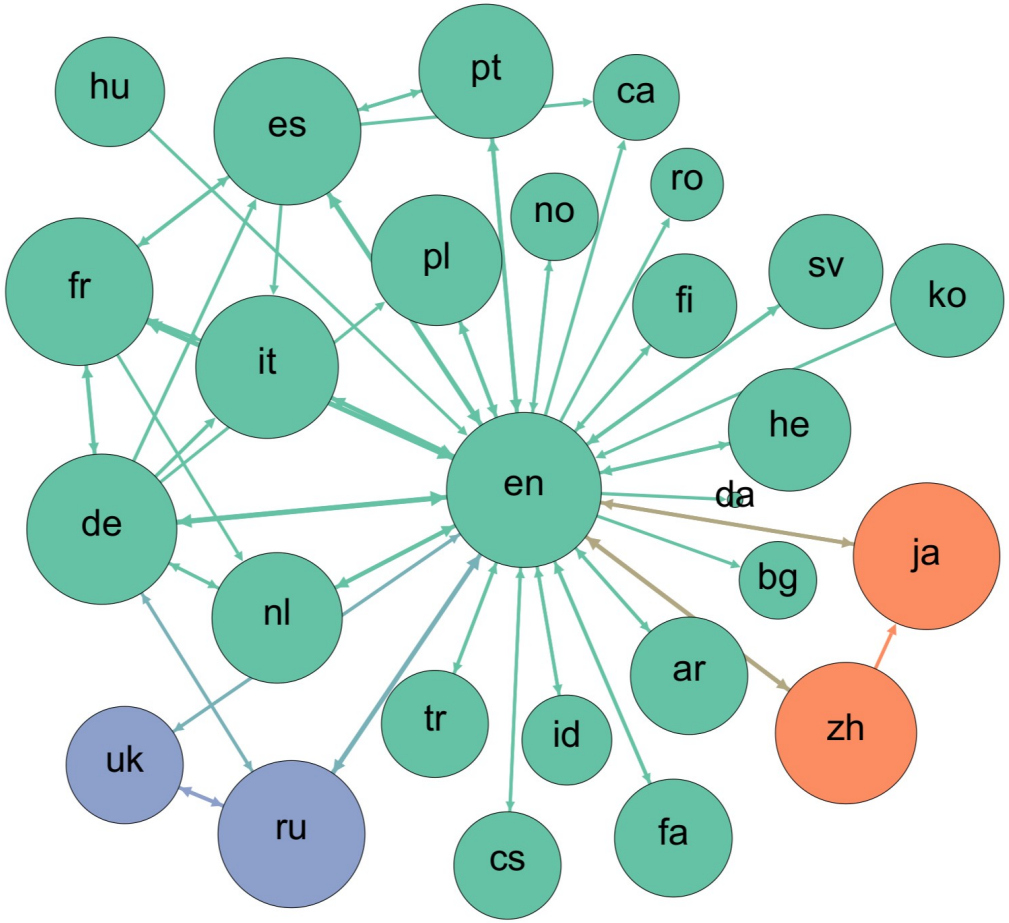




Graphs

- A graph is a mathematical structure that represents objects and binary relationships between them
  - Nodes: represent objects
    - Can hold associated information like spatial or temporal coordinates, or other features
  - Edges: connections between nodes
    - Relationship can be directed or undirected
    - Can have associated features
- Ideal structure for understanding physics data
  - Naturally sparse
  - Hits have a causal structure that can easily be modeled by edges
  - Accommodates relationships beyond nearest neighbor

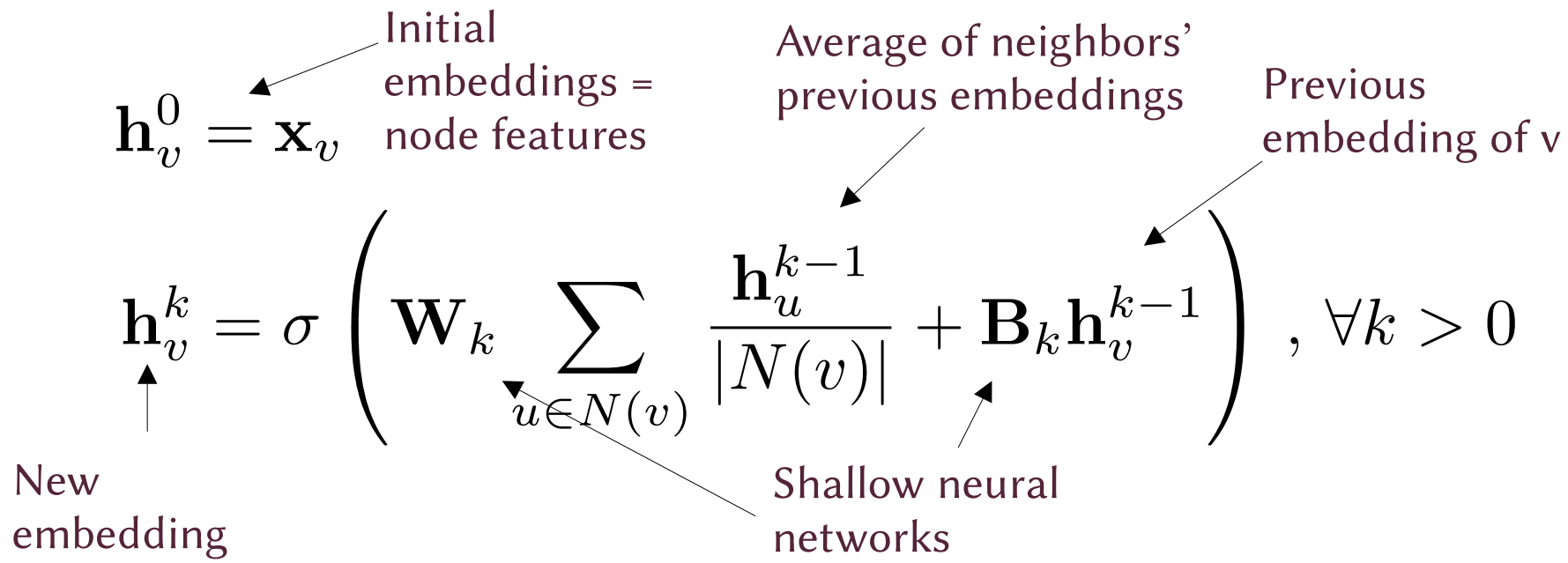
$$G = (V, E)$$



From arXiv:1312.0976
Graph of Wikipedia editors who primarily edit in one language, but sometimes edit in another

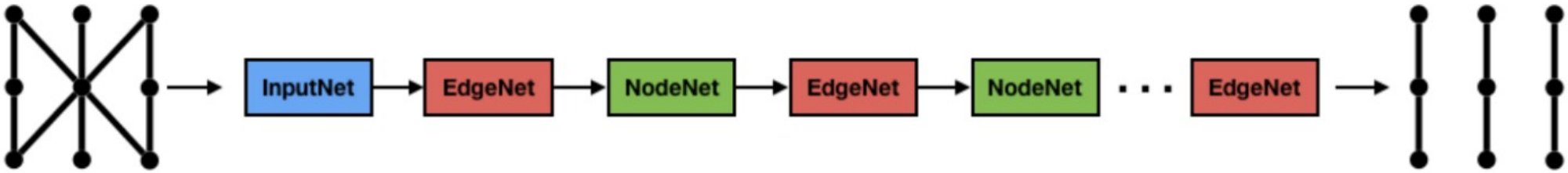
Graph Neural Networks

- GNNs are an extension of the idea of CNNs
  - Instead of extracting features from patches in a regular grid, extract features from neighbors of node
- Iteratively learn a smart embedding of graph structure
- Encode geometric information by passing and aggregating messages from neighbors



Graph Neural Networks for Tracking

- First Exa.TrkX GNN was developed for track finding at the LHC
- Initial graph constructed using structure of tracker barrels
  - Only consider connections between barrels with adjacent radii
  - Place limits on search radius (effectively limiting how low in p_T the algorithm searches)
- Alternating structure
  - EdgeNet predicts edge weights based on features of adjacent nodes
  - NodeNet aggregates node features from connected nodes weighted by the edge weight the features cross
- After several iterations, edges corresponding to fake tracks are down weights and those corresponding to real tracks are reinforced
  - Input graph is refined to discover tracks in the data

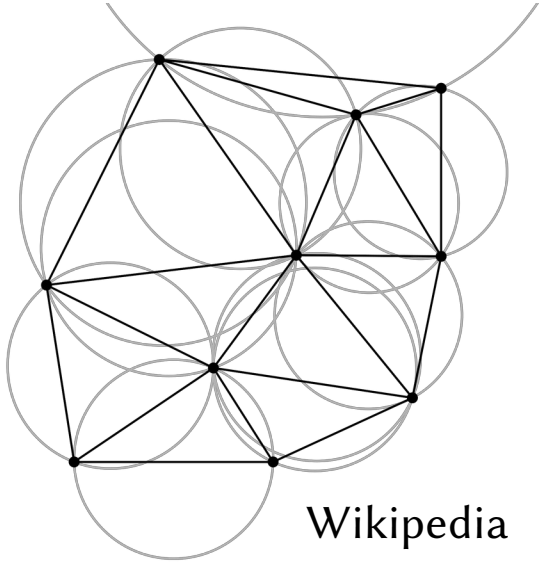


Semantic Segmentation Using GNN

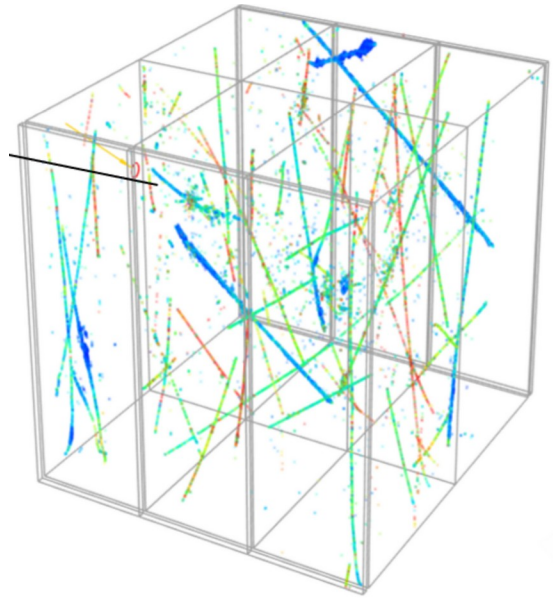
- The primary goal of NuGraph2 is to classify each detector hit according to particle type
- Five semantic categories:
  - MIP: minimum ionizing particles (muons, pions)
  - HIP: highly ionizing particles (protons, nucleons, kaons)
  - EM showers
  - Michel electrons
  - Diffuse activity (Compton scatters, neutrons)

Initial Event Graphs

- Message-passing algorithm requires constructing an initial graph out of the data
- MicroBooNE has three wire planes, so we construct three independent 2D graphs
  - Each node represents a reconstructed hit
    - Input features: wire index, hit time, integral, RMS width
  - Edges are formed for each 2D graph using Delaunay triangularization
    - Natively sparse representation
    - Both long and short distance edges for good information flow
- 3D hits can be formed by looking for coincidences between hits on three wire planes
  - 3D hits provide a way to connect three independent graphs together



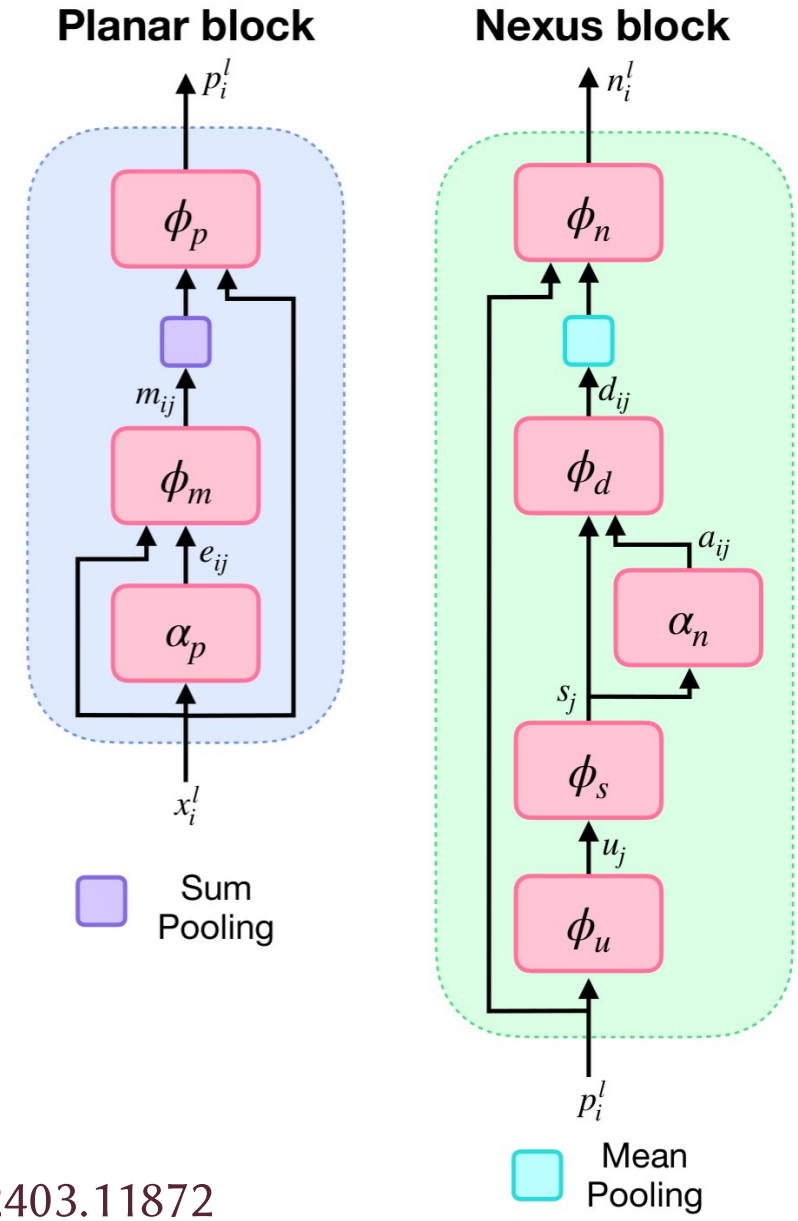
Wikipedia



arXiv:2002.03005

NuGraph2 Network Architecture

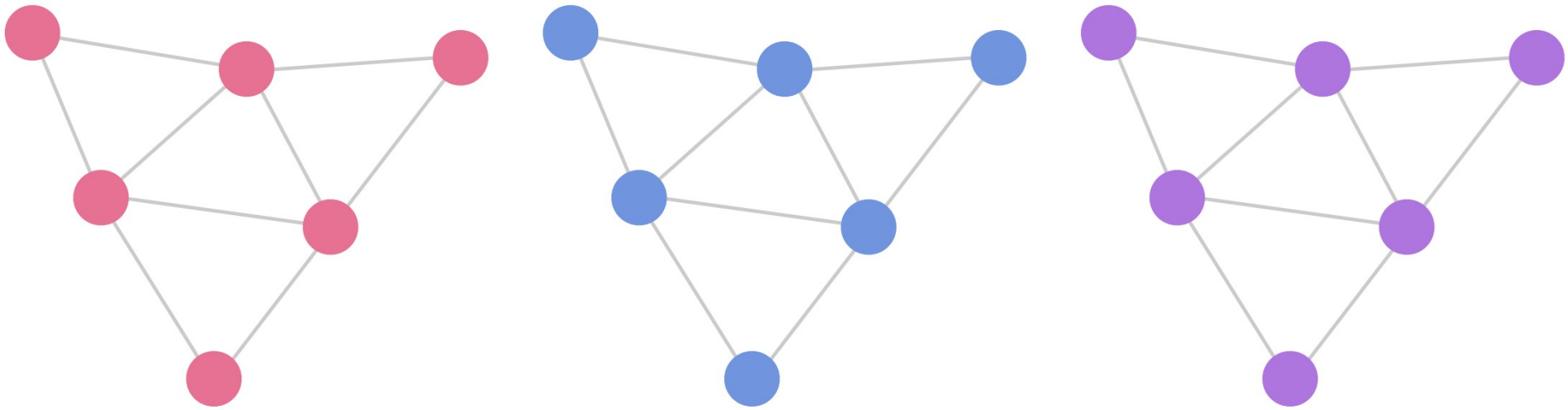
- NuGraph2's core convolution engine is a self-attention message passing network utilizing a categorical embedding
  - Each particle category is provided with a separate set of embedded features, which are convolved independently
  - Context information is exchanged between particle types via a categorical cross-attention mechanism
- Each message-passing iteration consists of two phases, the planar step and the nexus step:
  - Pass messages internally in each plane
  - Pass messages up to 3D nexus nodes to share information



arXiv:2403.11872

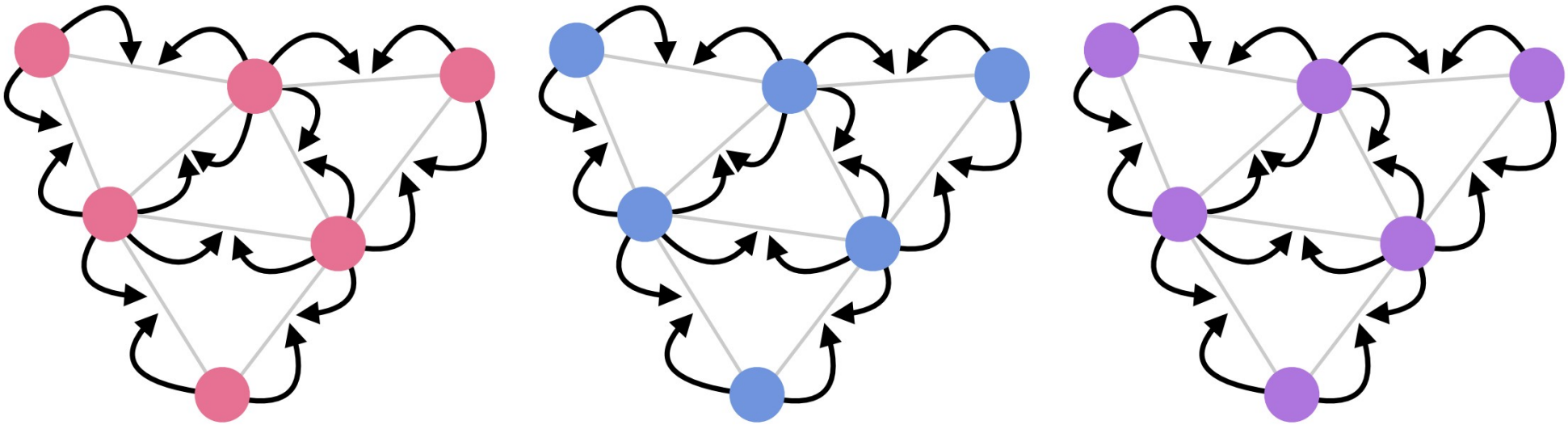
Message Passing Iteration Through the Graph

- Input graph three planar graphs with node features



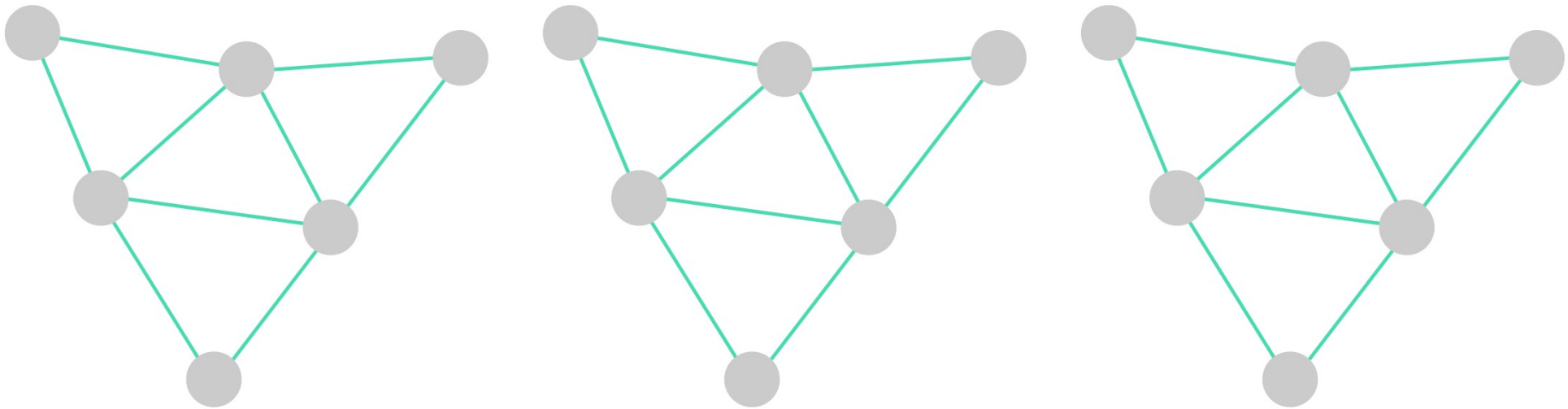
Message Passing Iteration Through the Graph

- Convolve node features to obtain edge features



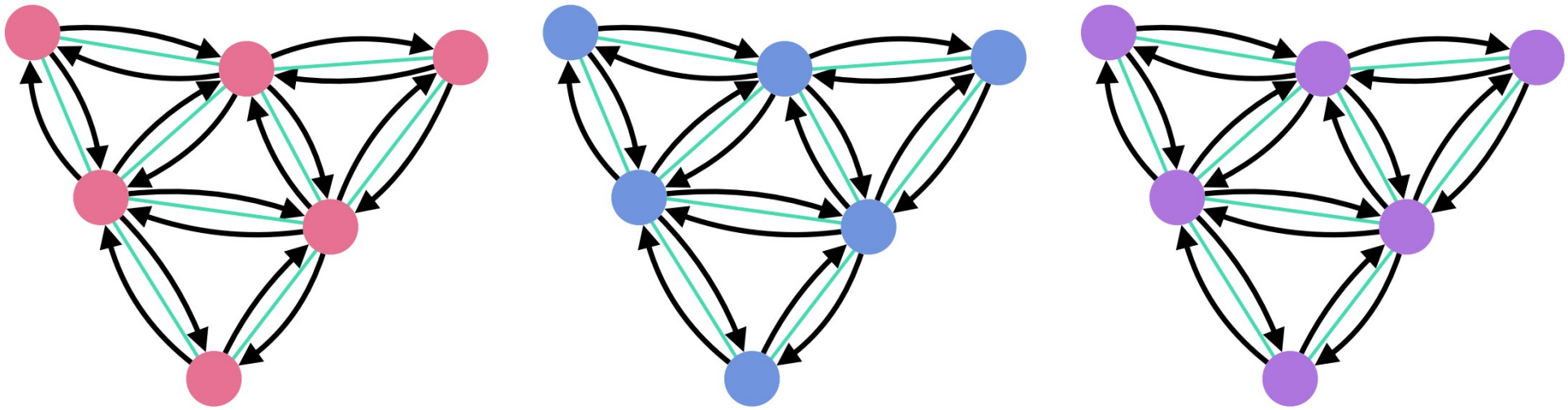
Message Passing Iteration Through the Graph

- Derive edge weights from edge features



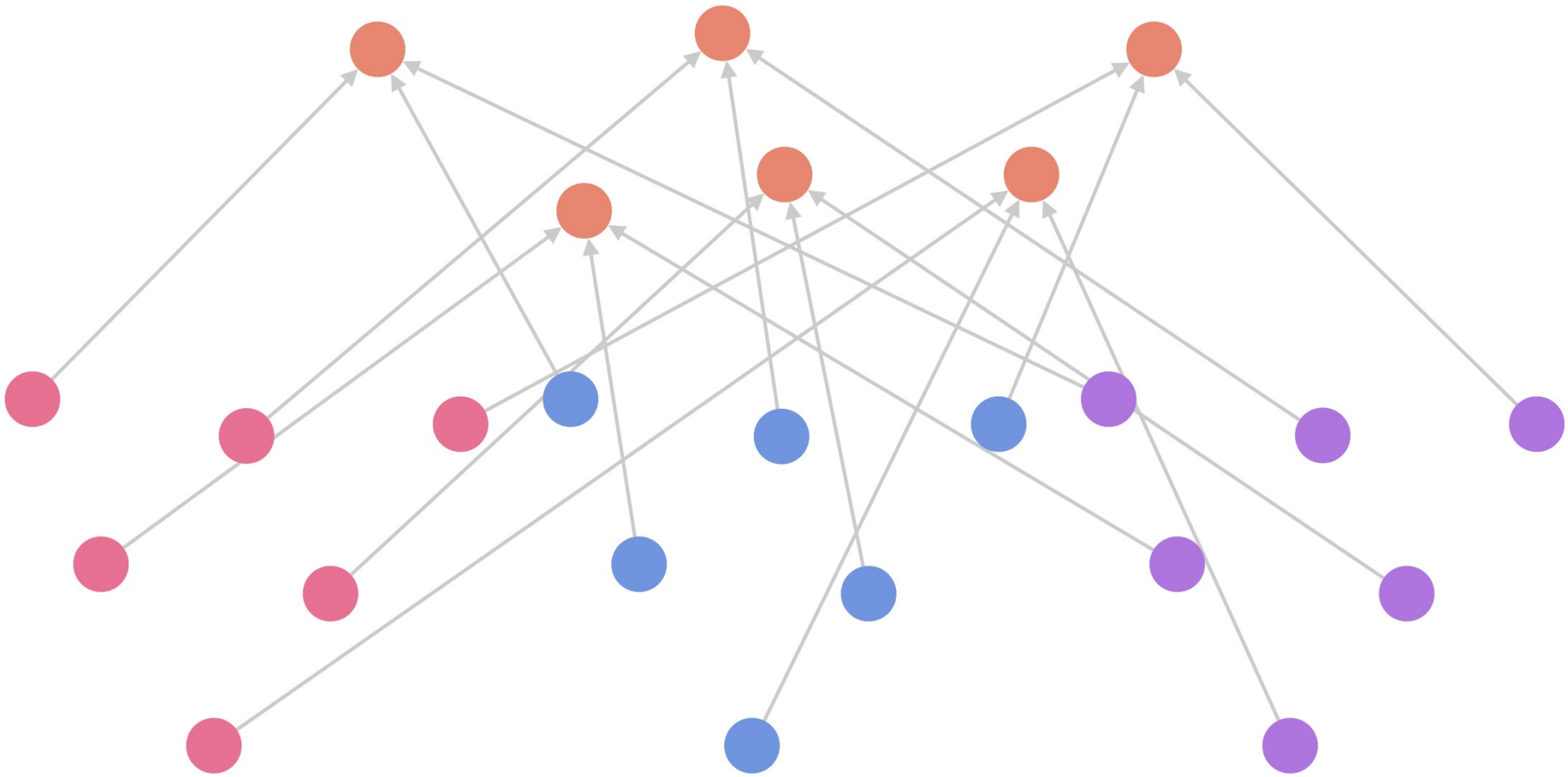
Message Passing Iteration Through the Graph

- Update node features using edge-weighted features from connected nodes



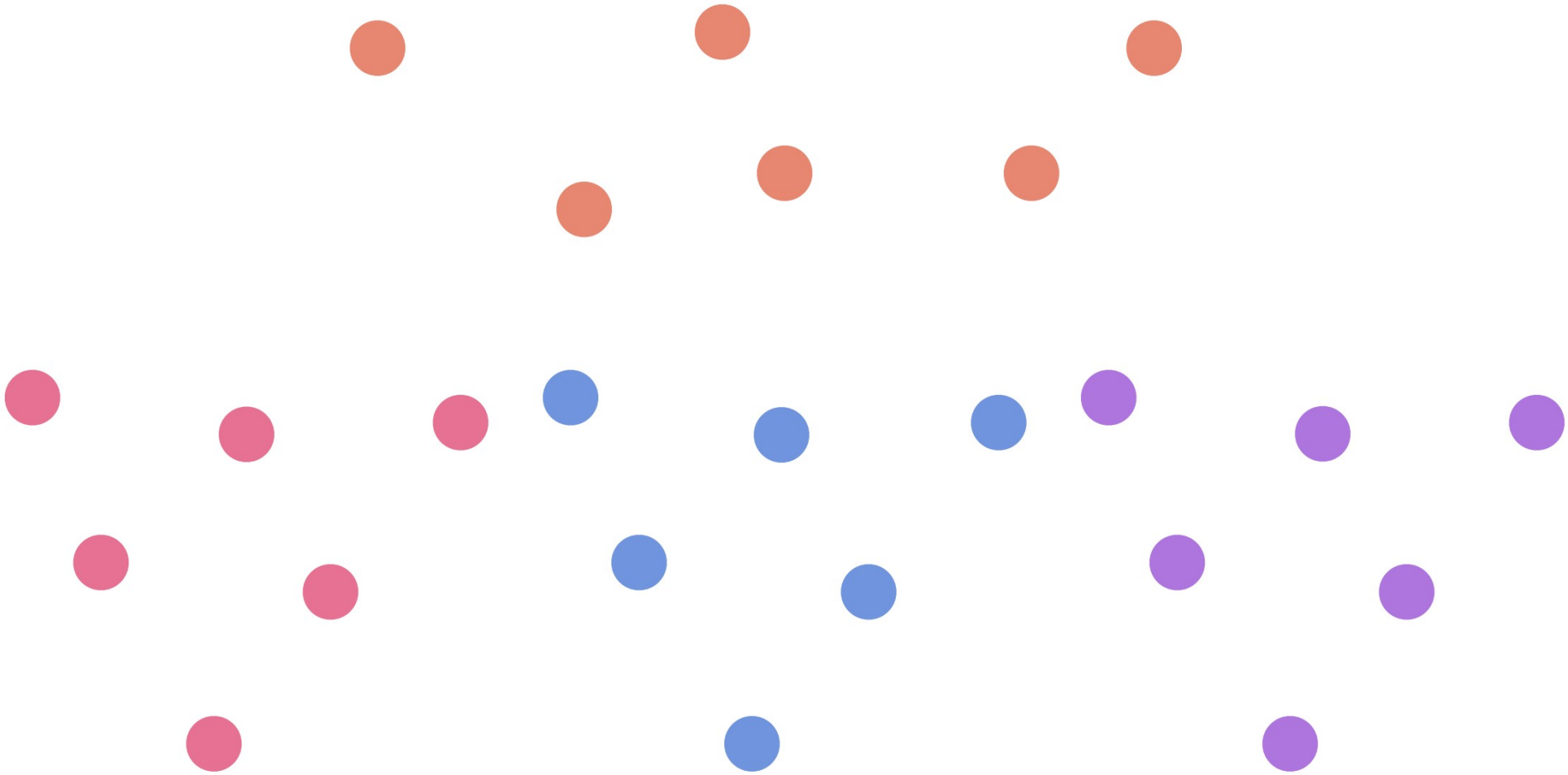
Message Passing Iteration Through the Graph

- Propagate node features to 3D nexus nodes



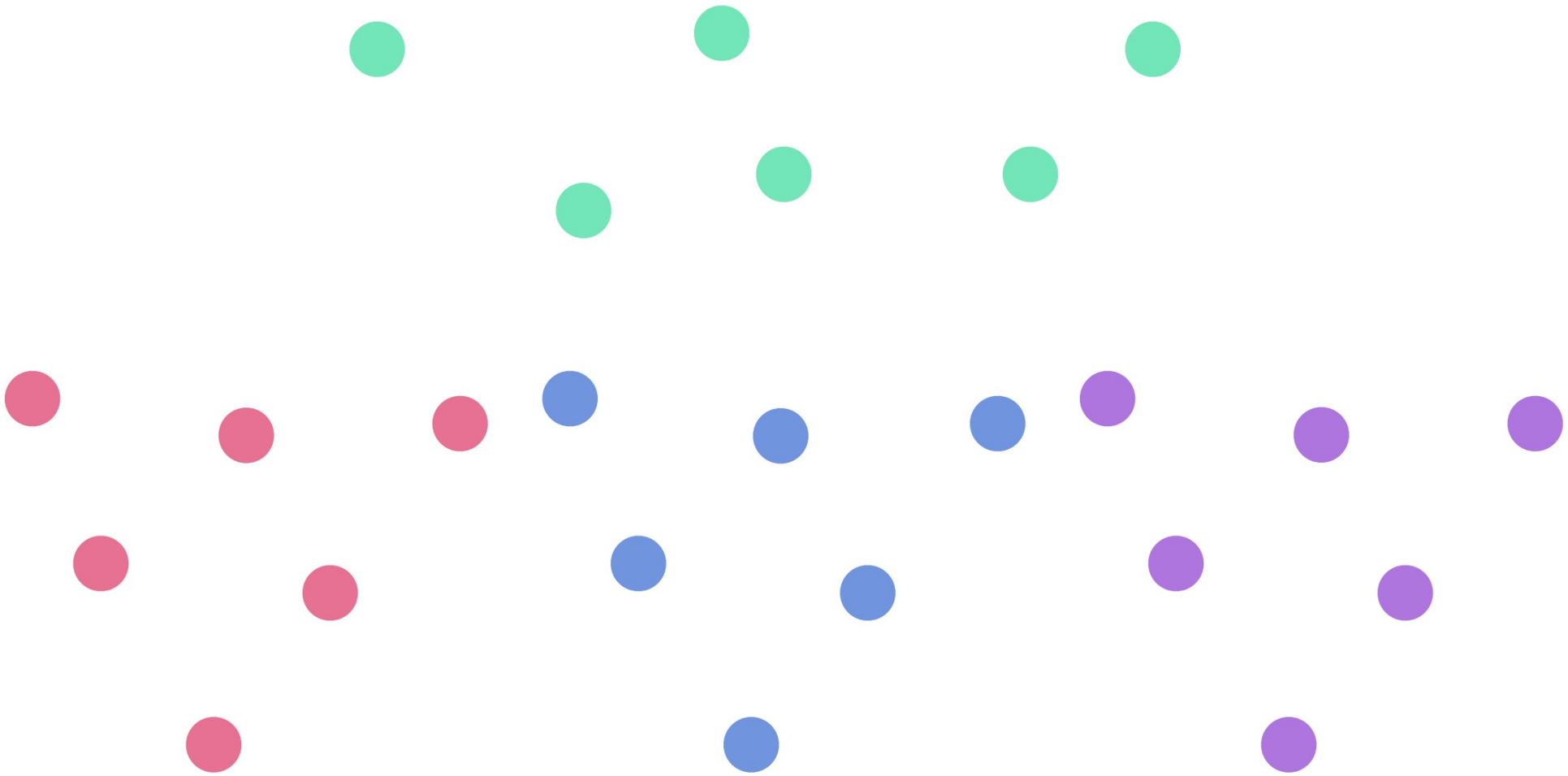
Message Passing Iteration Through the Graph

- Convolve nexus node features to mix information between planes



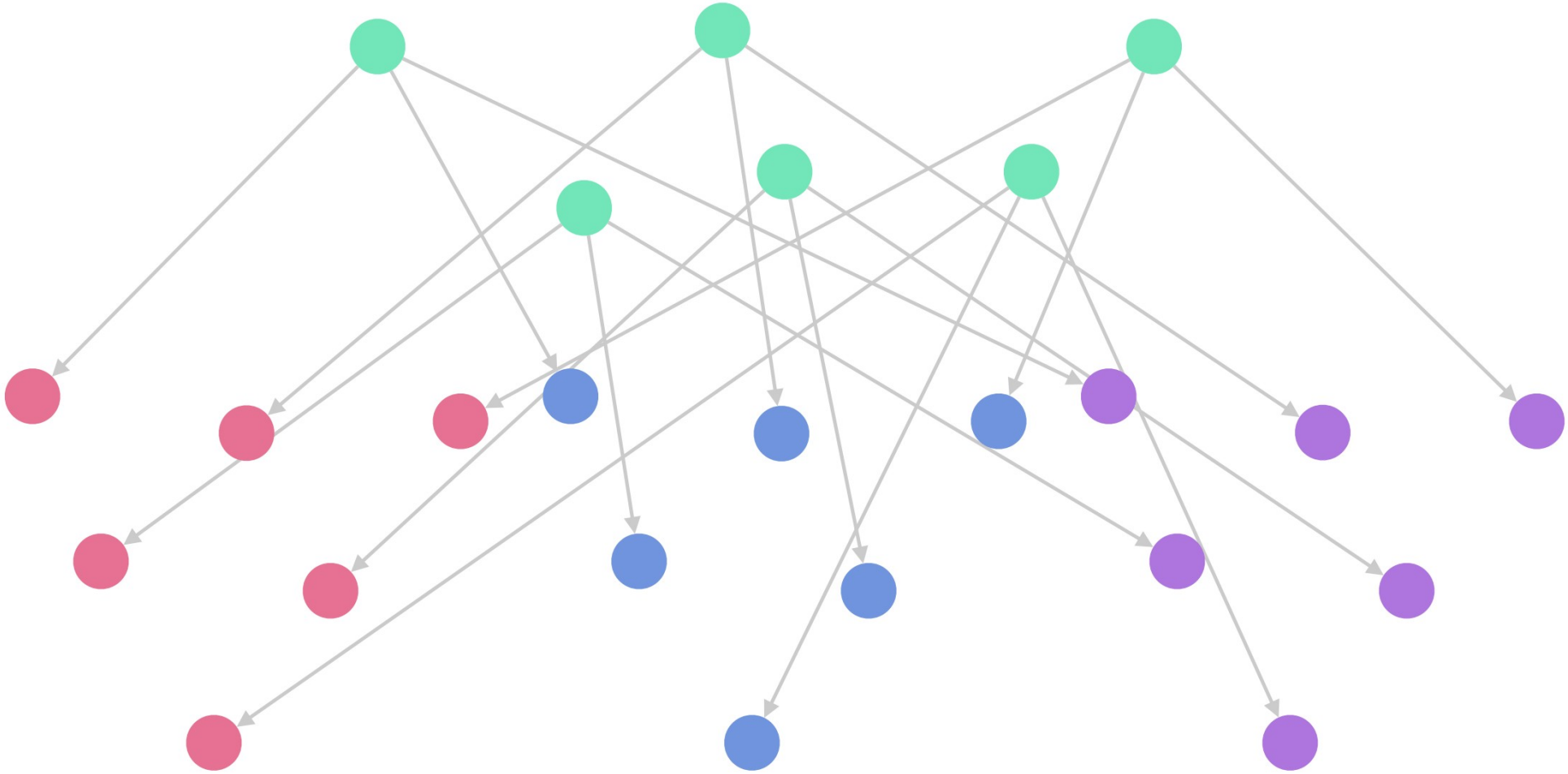
Message Passing Iteration Through the Graph

- Convolve nexus node features to mix information between planes



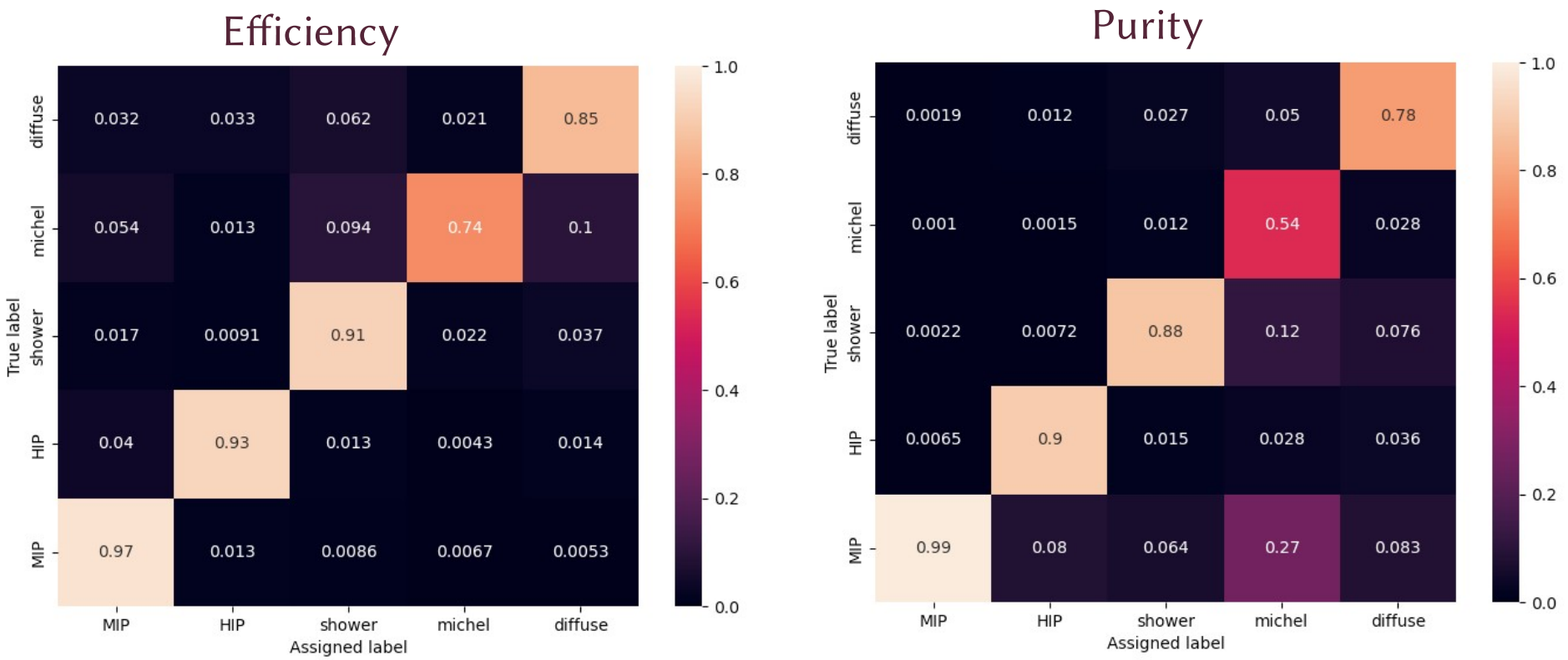
Message Passing Iteration Through the Graph

- Propagate 3D nexus node features back down to 2D planar nodes

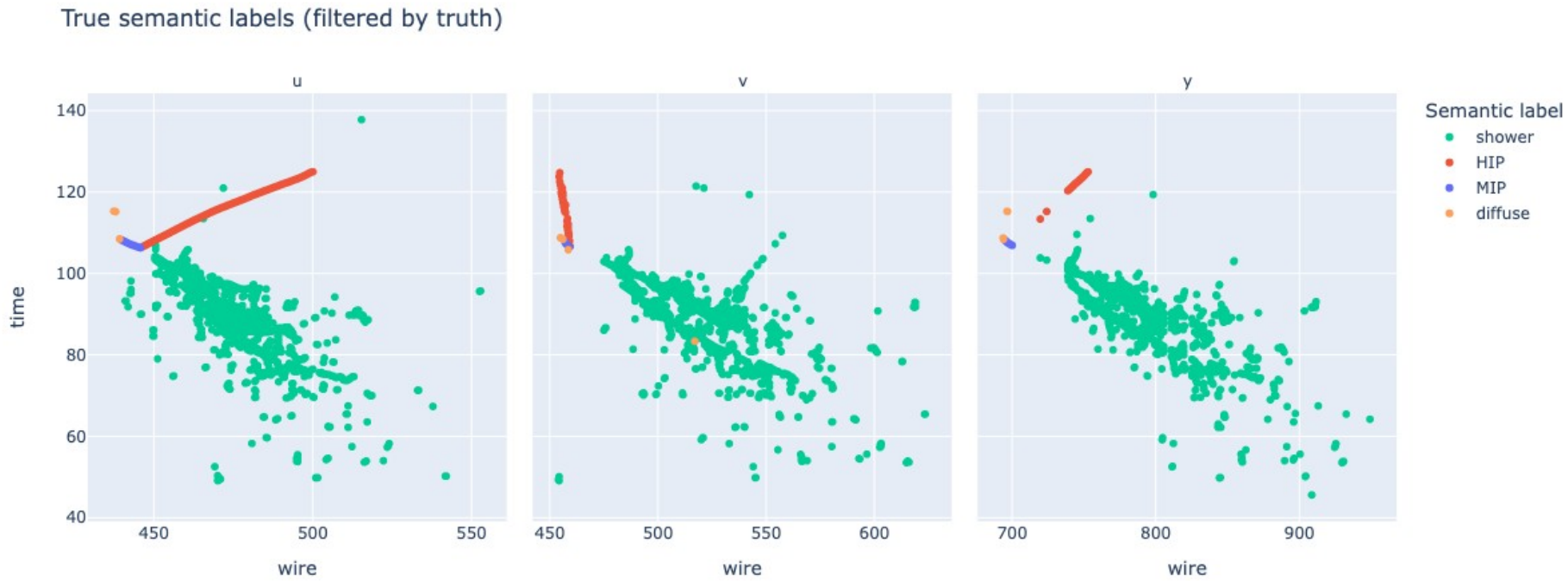


Semantic Hit Classification

- Decoder head trained to classify each hit according to particle type
- Overall efficiency and purity: ~95%
- Consistency between planes ~98%
  - Without 3D nexus connections, ~70% consistency

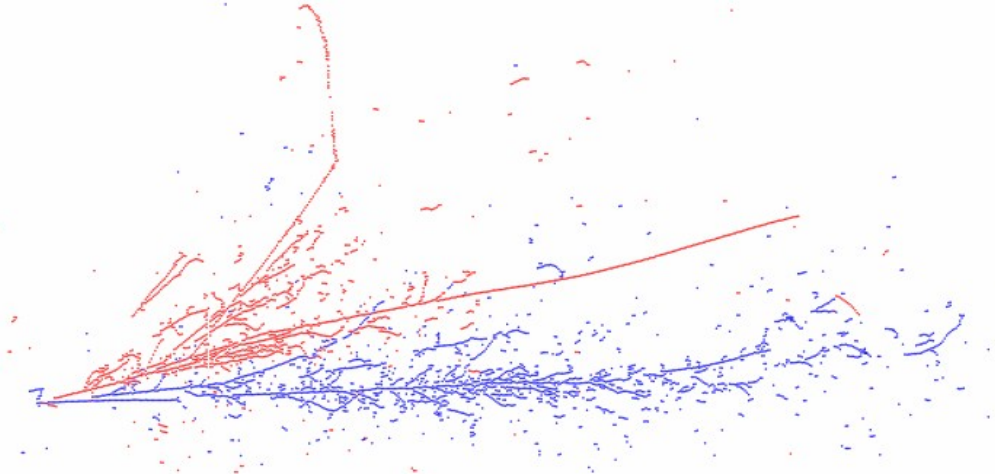


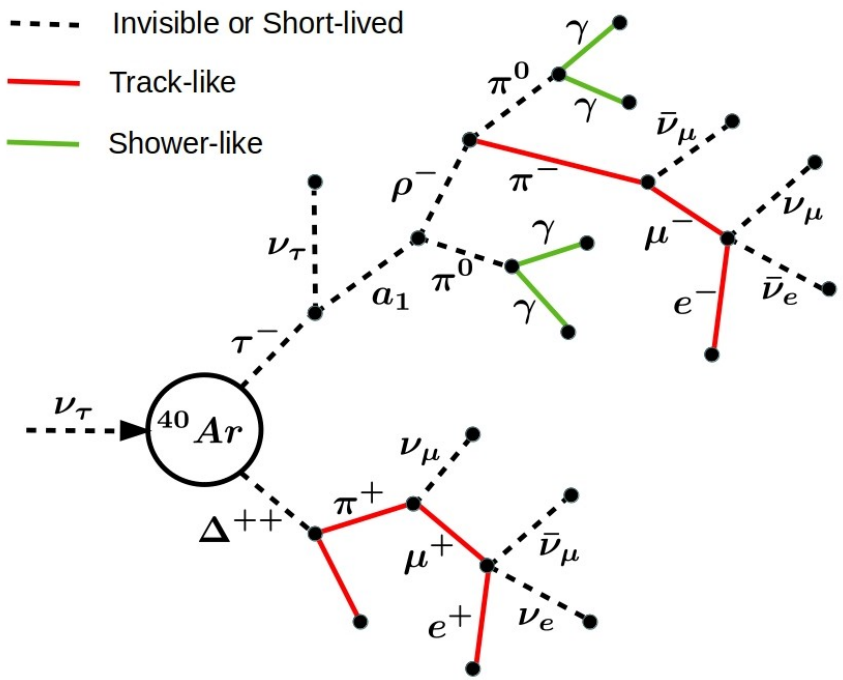
Event Display: Truth



Event Display: Predicted

Future Ideas

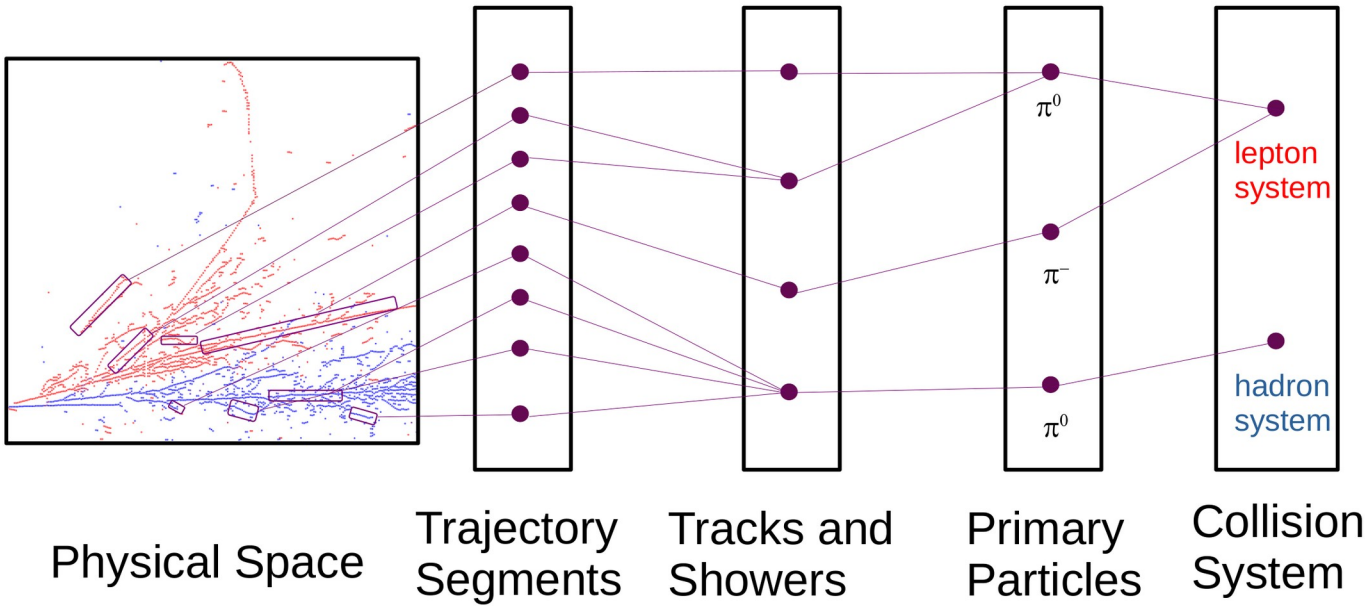




- GNN-based particle flow reconstruction
- Use 3D information from the start
- Fully-differentiable
  - End-to-end trainable
  - No lossy serial steps
- Able to reconstruct complex interactions like high energy tau neutrino interactions

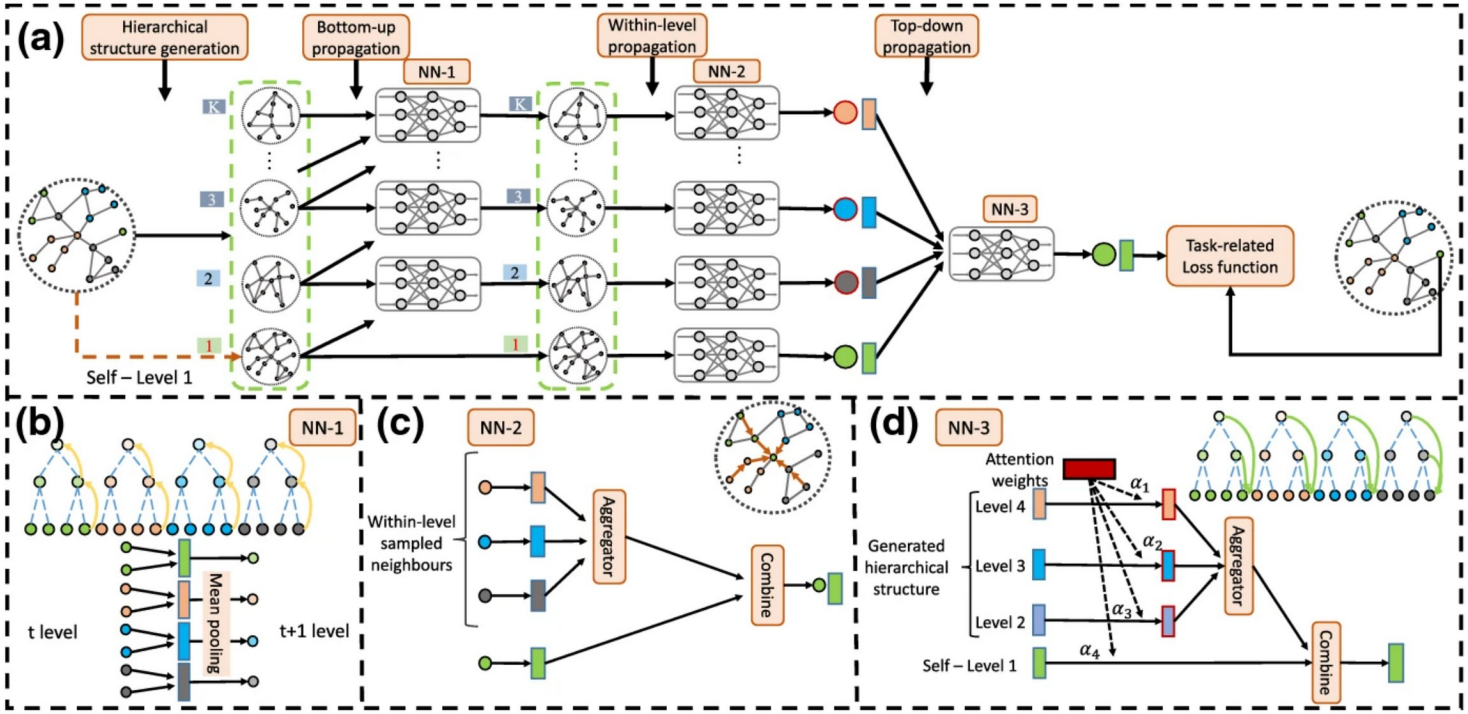
Hierarchical GNN Reconstruction Concept

- Build GNN to model a series of reconstruction stages
- Each stage connects elements from stage before to produce higher level objects
- Can be described as a hierarchical graph
  - Nodes on layer L are only connected to other nodes on layers L-1, L, and L+1
- Iteratively improve inference of particle tree through progressively refining hierarchy



Hierarchical GNNs

- Hierarchical GNNs were developed to overcome these shortcomings
- Hierarchical graph structure provides paths for long-distance information flow
- Structure captures rich, multi-scale information in natural way
- Hierarchical structure matches inductive bias for reconstructing particle trees



Z. Zhong, C. Li, J. Pang, arXiv:2009.03717

Instance Segmentation

- Complementary task to semantic segmentation is instance segmentation
  - Identification of discrete objects within the input
- The merging of these tasks is sometimes known as panoptic segmentation
  - Reconstructed objects with semantic labels

J. Redmon et al. arXiv:1506.02640

Most instance segmentation techniques work by finding bounding boxes (for instance, YOLO)

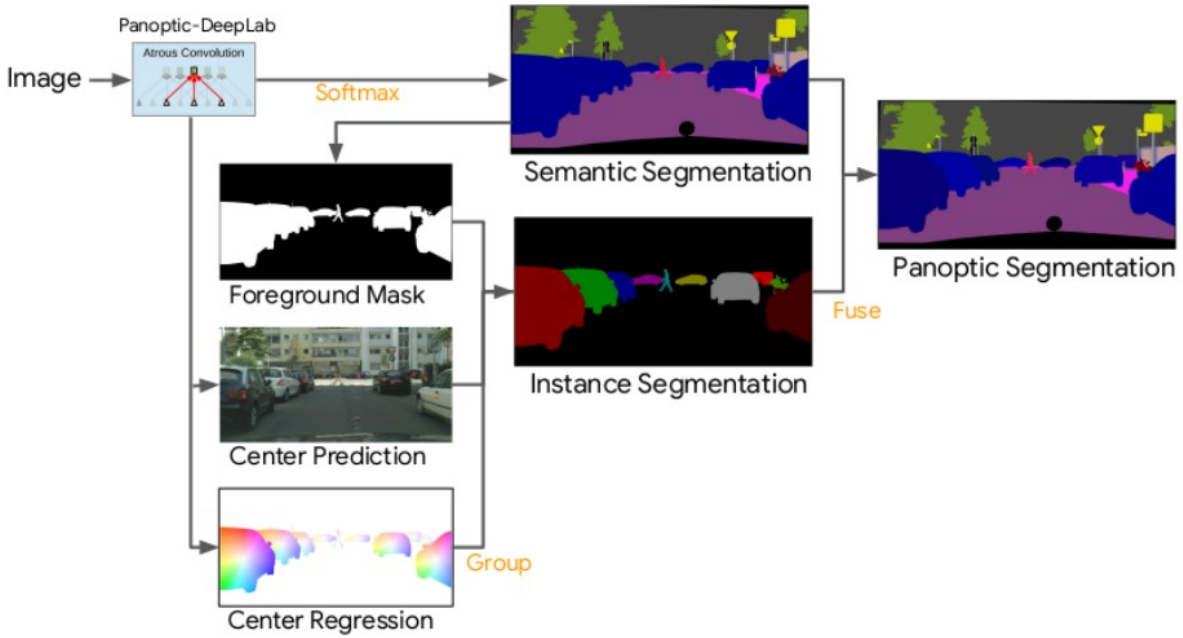
These methods tend to be resource intensive, and they rely on objects being reasonably compact in space

Physics detector output tends to be sparse, and objects may be non-compact and overlapping

Instance Segmentation

- Complementary task to semantic segmentation is instance segmentation
  - Identification of discrete objects within the input
- The merging of these tasks is sometimes known as panoptic segmentation
  - Reconstructed objects with semantic labels

B. Cheng et al. arXiv:1911.10194



Mask-based approaches, like Panoptic DeepLab, do not require bounding boxes, but they require center points in ground truth

Offsets to true center are regressed

Works reasonably well for sparse LArTPC inputs but can struggle with many close together tracks

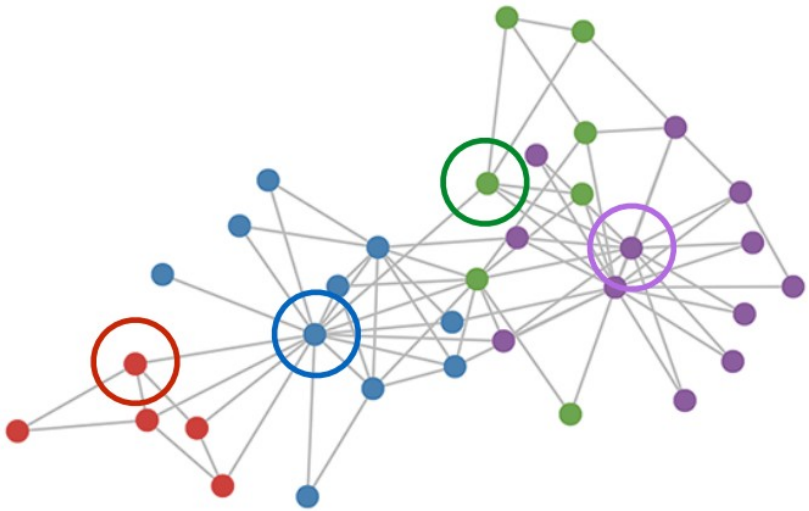
Tends to be somewhat fragile

Object Condensation

- Object condensation is a grid-free approach based on an electro-static analog
- Predict a quantity β_i between 0 and 1 for each node
- This quantity will be used to assign a charge
- Points with maximum charge will be used as condensation points
  - Representative points around which clusters will be formed
- A loss is added which encourages a single condensation point per object

$$q_i = \operatorname{artanh}^2 \beta_i + q_{\min}$$

$$L_\beta = \frac{1}{K} \sum_k (1 - \beta_{\alpha k}) + \frac{s_B}{N_B} \sum_i^N n_i \beta_i$$



J. Kieseler, arXiv:2002.03717

Object Condensation

- Predict coordinates of each node in an abstract clustering space
- Attractive and repulsive potentials are defined such that nodes belonging to the same object are attracted and those from different objects are repelled
- Condensation points can be used for inferring particle properties
  - Particle property losses are also weighted by charge to encourage a single representative set of properties per object

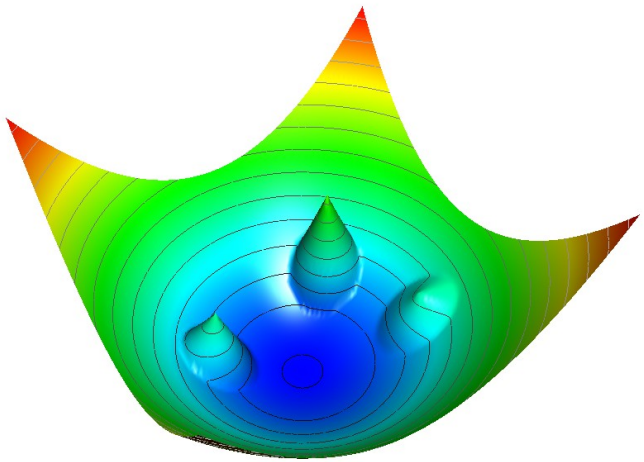
$$A_k = \| x - x_\alpha \|^2 q_{\alpha k}$$

$$R_k = max(0, 1 - \| x - x_\alpha \|) q_{\alpha k}$$

$$L_V = \frac{1}{N} \sum_{j=1}^{N} q_j \sum_{k=1}^{K} (M_{jk} A_k(x_j) + (1 - M_{jk}) R_k(x_j))$$

M_{jk} = 1 if node j in object k

M_{jk} = 0 otherwise



J. Kieseler, arXiv:2002.03717

Summary

- I have barely scratched the surface on statistical and machine learning topics
- More reading on statistics
  - Statistical Methods in Experimental Physics
    - Frederick James (author of Minuit)
  - Statistical Data Analysis
    - Glen Cowan
- Machine learning
  - This is a quickly changing field, but there are some places to get started
  - Deep Learning
    - www.deeplearningbook.org
  - Artificial Intelligence for High Energy Physics
    - Collection of chapters on various topics, some are on the arXiv
      - End-to-end reconstruction using image classification arXiv:2208.03285 (Aurisano and Whitehead)
      - Boosted decision trees arXiv:2206.09645 (Coadou)
      - Graph neural networks for tracking arXiv:2012.01249 (Duarte and Vlimant)