Graph Neural Networks

and Application for Cosmic-Ray Analysis

Paras Koundal
paraskoundal.com
Institute for Astroparticle Physics, KIT, Germany
/ParasKoundal

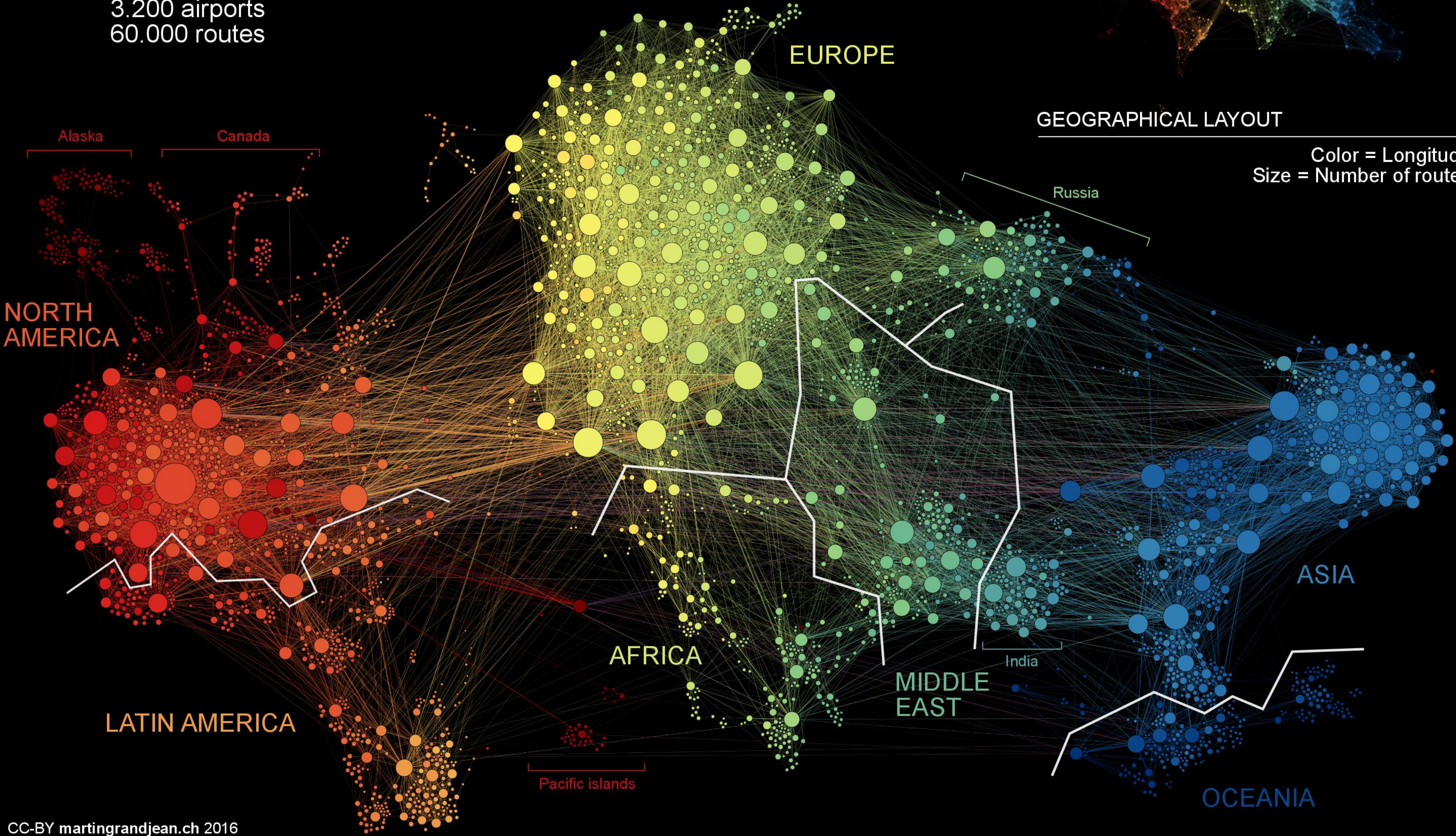
Deep Learning in Computational Physics, 2021

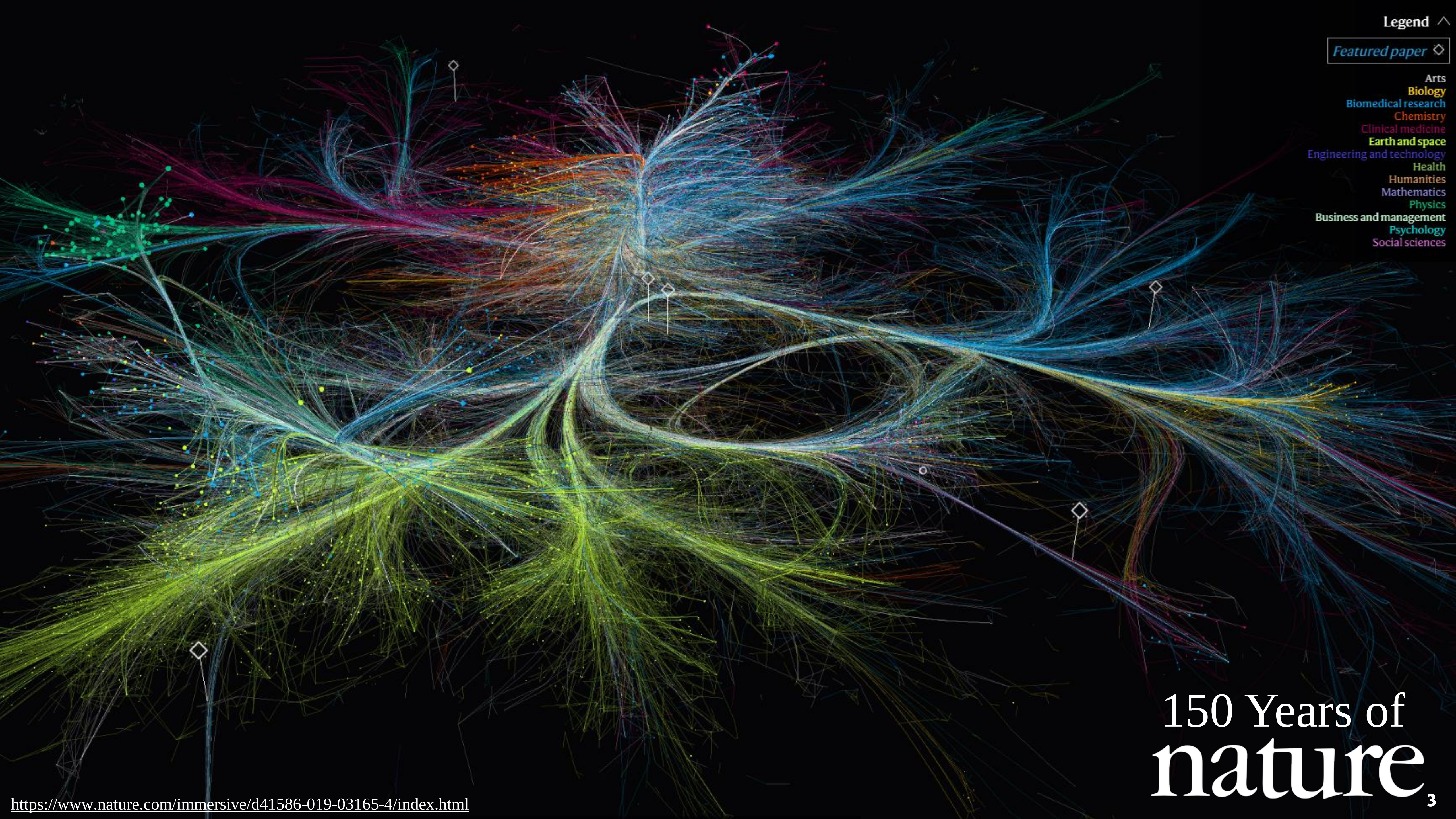
TRANSPORTATION CLUSTERS

3.200 airports
60.000 routes

GEOGRAPHICAL LAYOUT

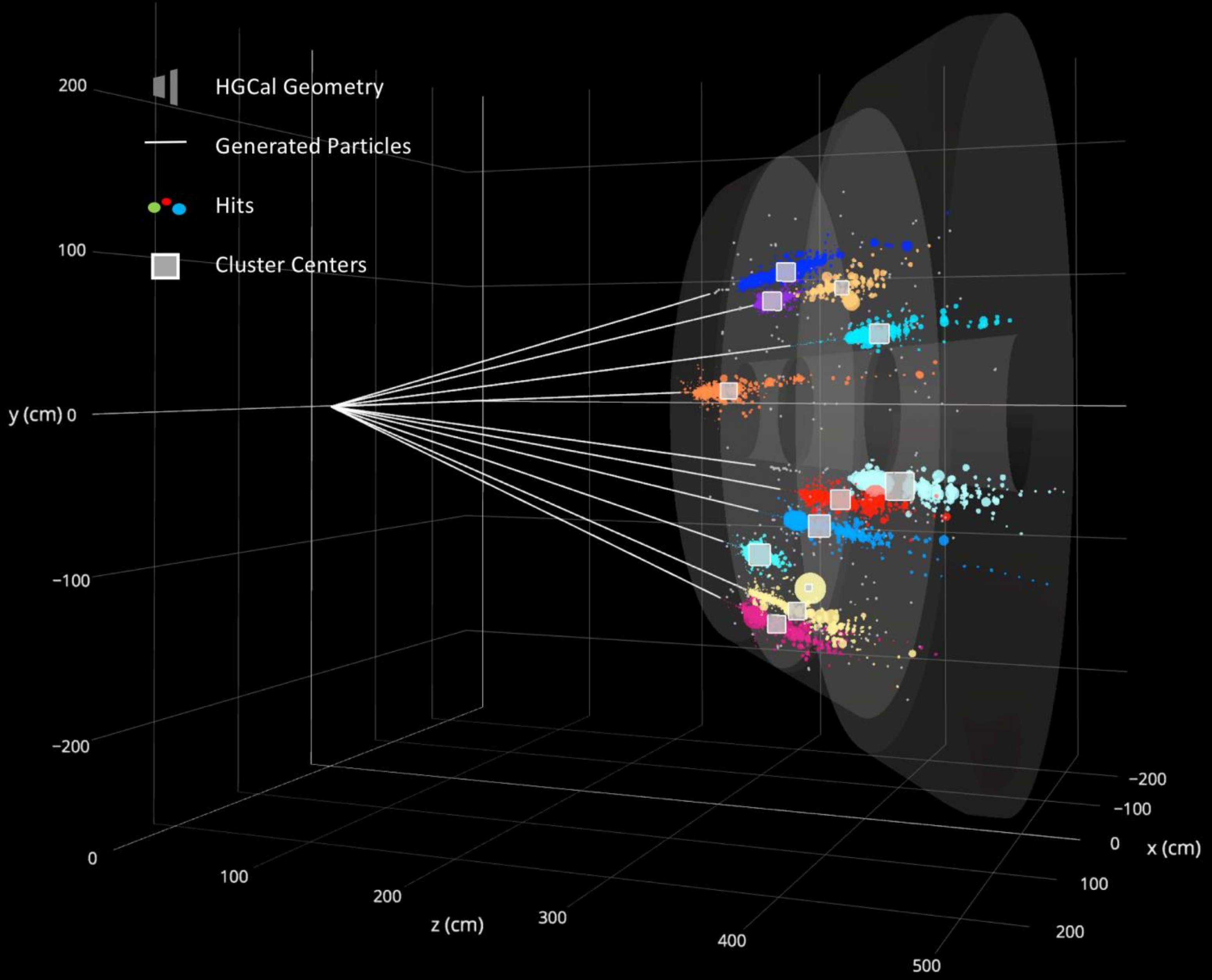
Color = Longitude
Size = Number of routes

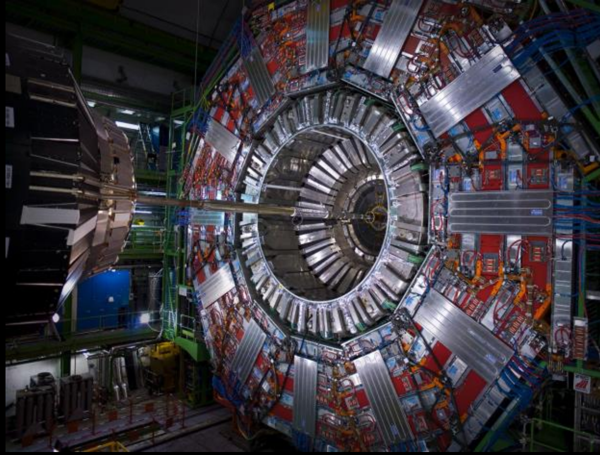




150 Years of nature

https://www.nature.com/immersive/d41586-019-03165-4/index.html





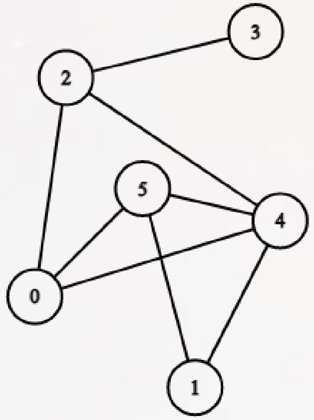
https://news.fnal.gov/2020/09/the-next-big-thing-the-use-of-graph-neural-networks-to-discover-particles/

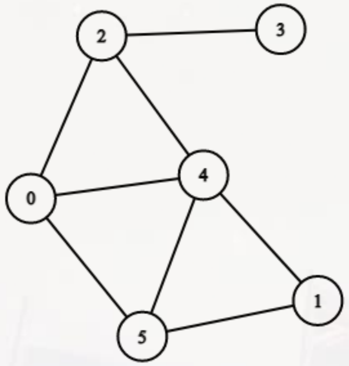
Graph

Defined by **set of nodes** (V) and **set of edges** (E) between the nodes

$$G = (V, E)$$

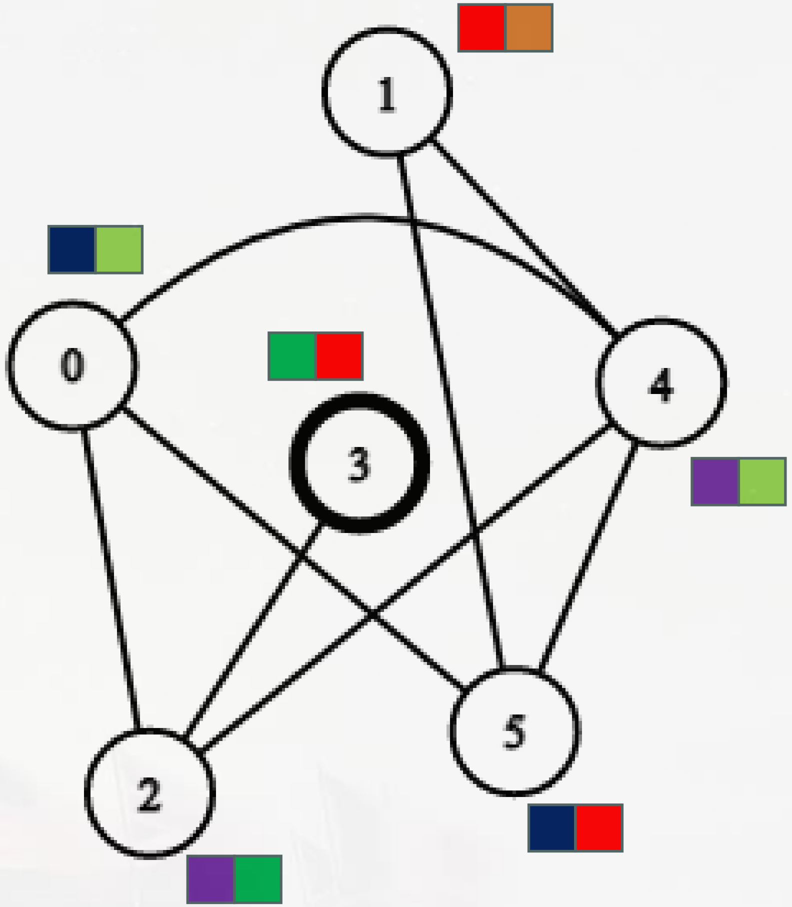
- Neighborhood and Connectivity





- Node Labelling : Permutational Invariance

- **Possible Types**
  - **Undirected** : Facebook Friends …
  - **Directed** : Citation Graph …
  - **Bidirectional** : Twitter Follows

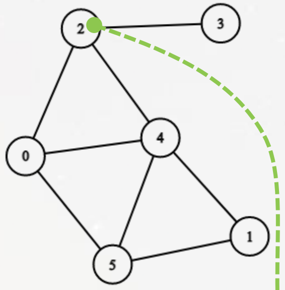


- **Other Attributes**
  - **Node Features**
  - **Edge Features**

Representing Graphs

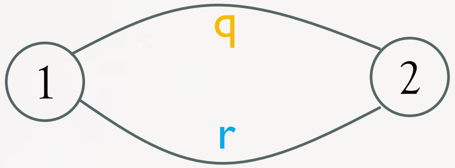
- **Adjacency Matrix (A)**:
  - Create Matrix: Shape = N*N
  - Simplest Case: Presence of Edge = 1 ; Otherwise = 0
  - Properties of A:
    - For Undirected Graph: Symmetric
    - In general, Connections can be Weighted too

  - For Multi-relational Data:
    - Can have different type of edges between two same nodes



$$A = \begin{array}{c} \begin{array}{cccccc} ⓪ & ① & ② & ③ & ④ & ⑤ \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \begin{array}{c} ⓪ \\ ① \\ ② \\ ③ \\ ④ \\ ⑤ \end{array}$$

$$A.x = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \sum A_{ij} x_j = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$



$$\Rightarrow \quad A_q , A_r$$

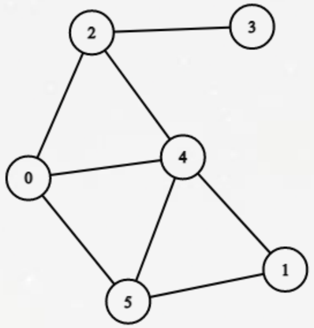
- **Degree Matrix (D)**:
  - Number of edges connected to a node
  - For Directed Graph: Incoming and Outgoing degree
  - Information about node's importance

$$D = \begin{array}{c} \begin{array}{cccccc} ⓪ & ① & ② & ③ & ④ & ⑤ \end{array} \\ \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix} \end{array} \begin{array}{c} ⓪ \\ ① \\ ② \\ ③ \\ ④ \\ ⑤ \end{array}$$

- **Laplacian Matrix** :
  - Unnormalized Laplacian (L):

$$L = D - A$$

    - Values (L_{ij}):
      - If i=j: degree(V_{ij}); If Connected: -1 ; Otherwise = 0
    - Properties: Symmetric ($L^T = L$)



$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad D = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}$$

$$L = \begin{pmatrix} 3 & 0 & -1 & 0 & -1 & -1 \\ 0 & 2 & 0 & 0 & -1 & -1 \\ -1 & 0 & 3 & -1 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ -1 & -1 & -1 & 0 & 4 & -1 \\ -1 & -1 & 0 & 0 & -1 & 3 \end{pmatrix}$$

  - Gives an idea of connected components in the graph. (Matrix can be re-written in a block diagonal form)

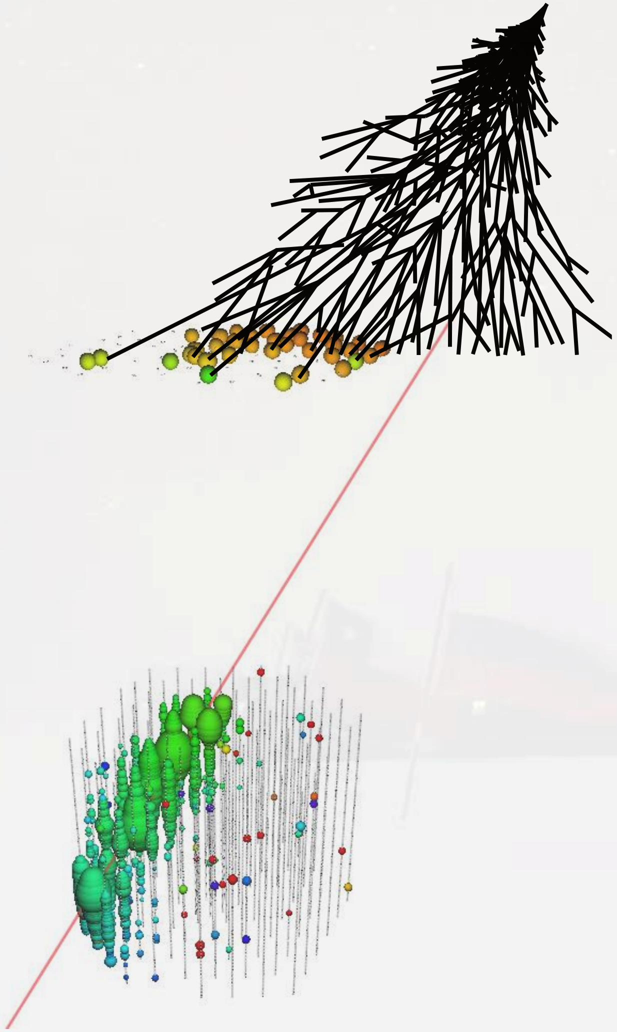
$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & & & \\ & \mathbf{L}_2 & & \\ & & \ddots & \\ & & & \mathbf{L}_K \end{bmatrix}$$ [1]

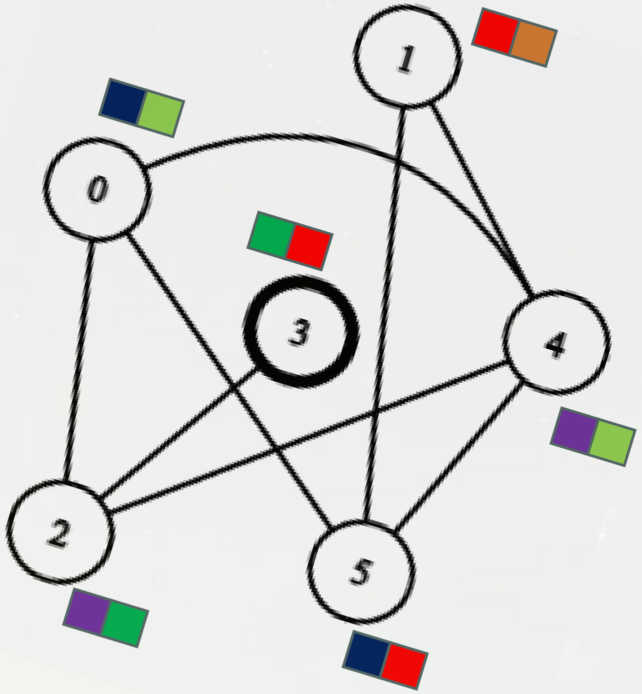
  - Other Variants:
    - **Normalized Laplacian:** $D^{-1/2} L D^{1/2}$
    - **Random Walk Laplacian:** $D^{-1} L$

    Similar properties as Laplacian; algebraic properties different because of normalization.

- **Other Measures**: Centrality (Node and Betweeness); Clustering Coefficient etc..

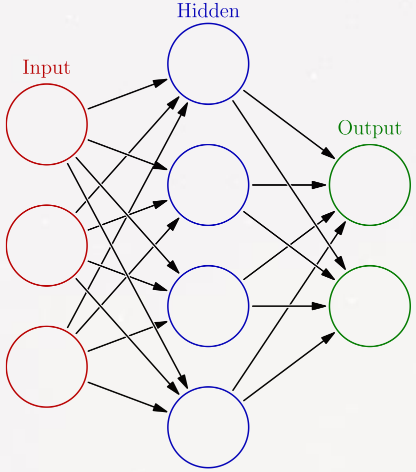
Learning on Graphs



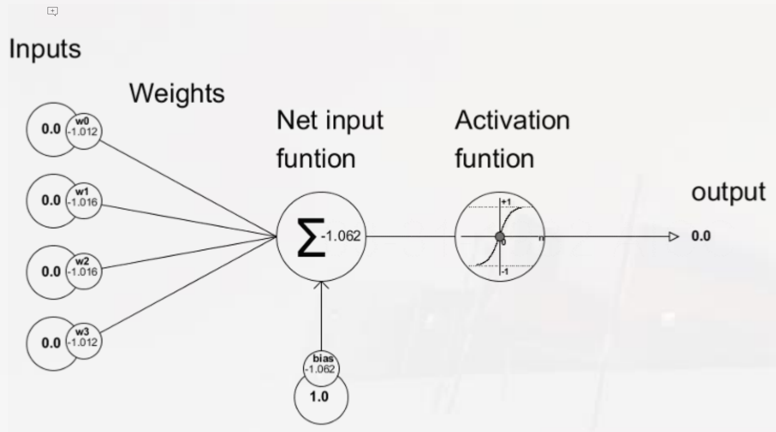


1. Applicability to Different Tasks
2. Graph Convolutional Networks
3. Application to Cosmic Ray Physics

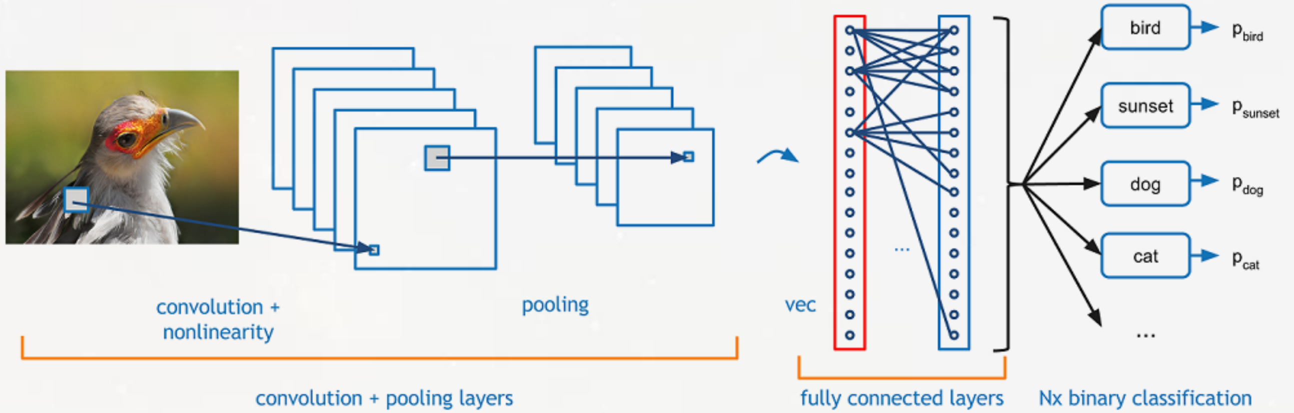
Multi-Layer Perceptron: MLP



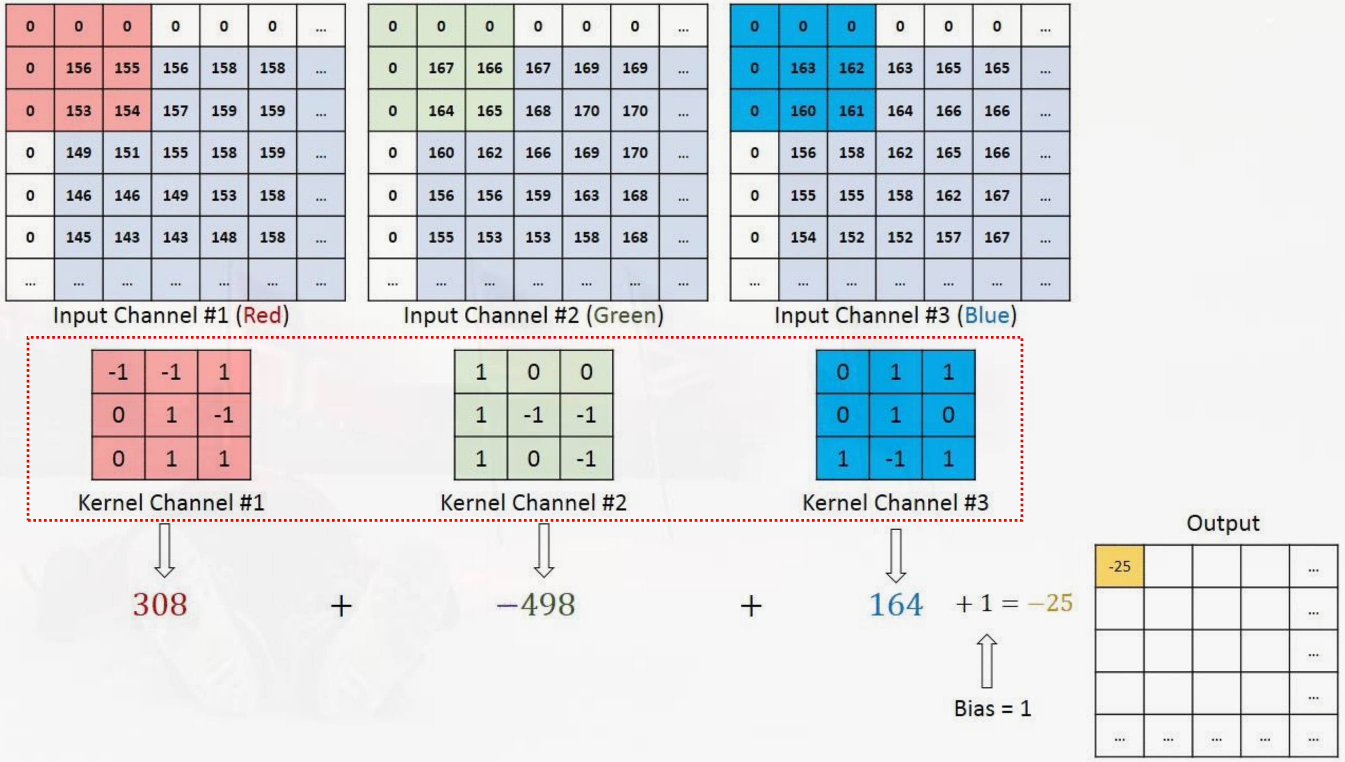
Weight Optimization



Convolution Neural Networks: CNN



Convolution Operation:



Input Channel #1 (Red)

0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...	...	...	...	...	...	...

Input Channel #2 (Green)

0	0	0	0	0	0	...
0	167	166	167	169	169	...
0	164	165	168	170	170	...
0	160	162	166	169	170	...
0	156	156	159	163	168	...
0	155	153	153	158	168	...
...	...	...	...	...	...	...

Input Channel #3 (Blue)

0	0	0	0	0	0	...
0	163	162	163	165	165	...
0	160	161	164	166	166	...
0	156	158	162	165	166	...
0	155	155	158	162	167	...
0	154	152	152	157	167	...
...	...	...	...	...	...	...

Kernel Channel #1

-1	-1	1
0	1	-1
0	1	1

Kernel Channel #2

1	0	0
1	-1	-1
1	0	-1

Kernel Channel #3

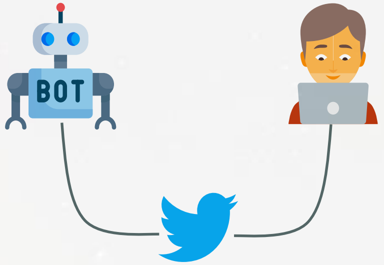
0	1	1
0	1	0
1	-1	1

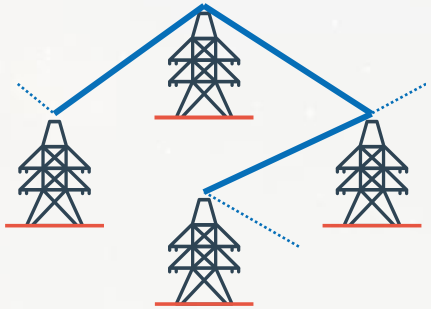
$$308 \quad + \quad -498 \quad + \quad 164 \quad +1 = -25$$

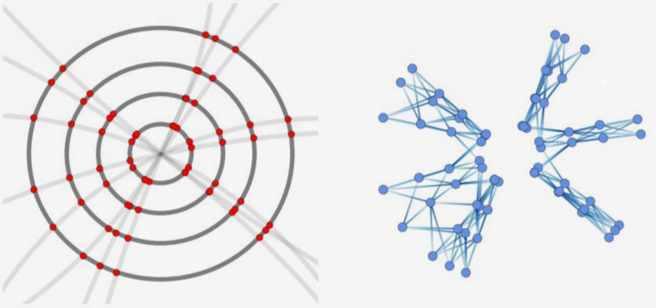
Machine Learning on Graphs

1. Node Classification:

- Predict label/category for unlabeled nodes; given label true labels on training nodes
- Interconnectivity between individual nodes is important deciding factor .
- **Example**: 1. Labelling bots on twitter 2. Finding Weak Points in Power Grid 3. Labelling particle type in a detector.



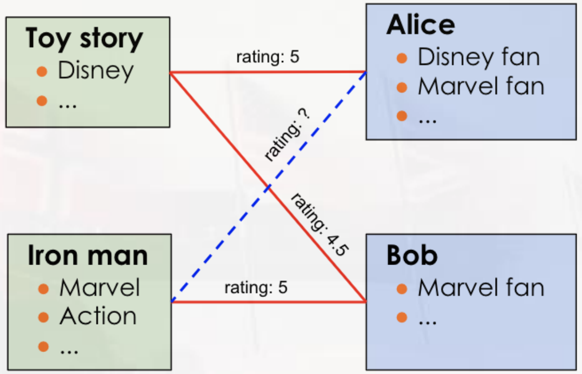




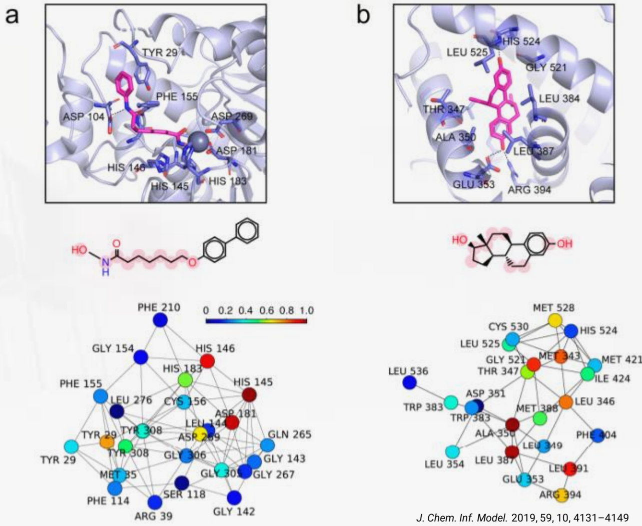
Jonathan Shlomi *et al* 2021 *Mach. Learn.: Sci. Technol.* 2 021001

2. Link Prediction

- Prediction about unknown or non-existent connections.
- Complexity dependent on graph type
- **Example**: 1. Movie Recommendation Suggester
  2. Studying unknown Drug-Target interactions



Cred.: https://bit.ly/3vKLwXG

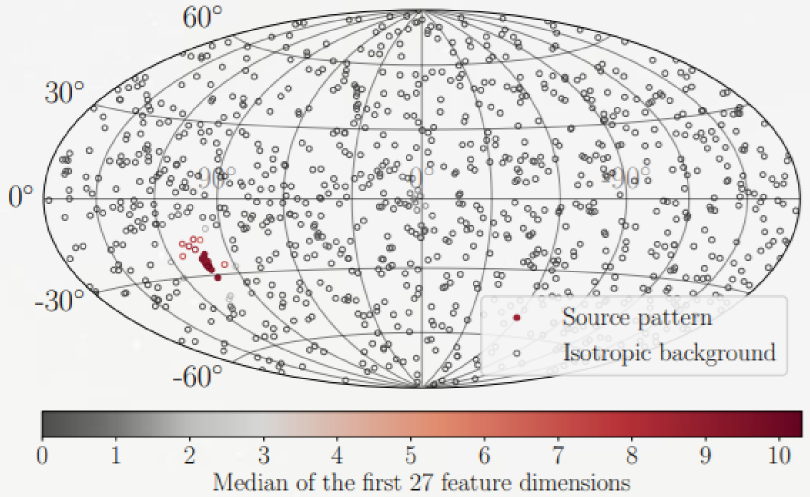


J. Chem. Inf. Model. 2019, 59, 10, 4131−4149

PC: Bot, Electricity Tower- Jemis Mali @Iconscout ; Twitter - PixelIcons

3. Cluster/Community Detection:

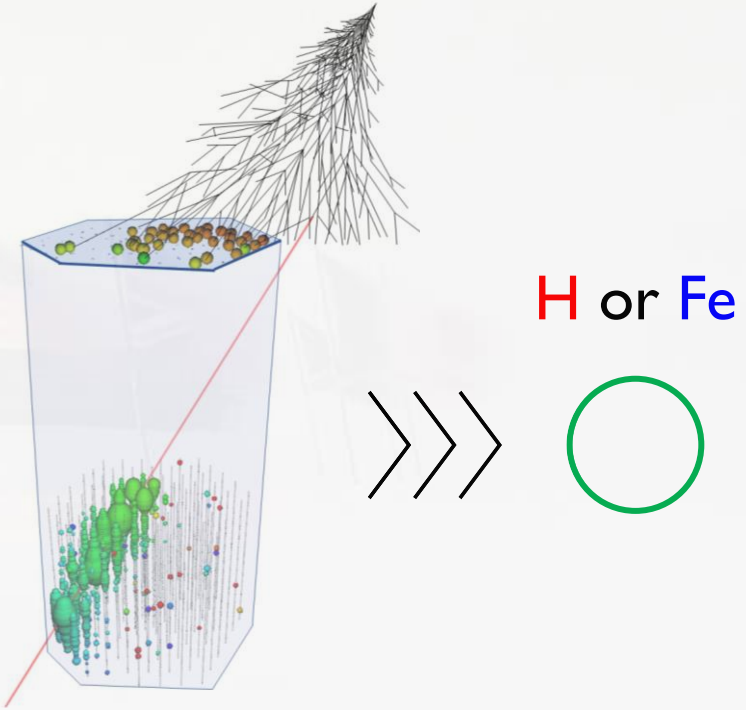
- Look for clusters; given node and connection details
- **Example**: 1. Looking for astrophysical source clusters 2. Fraud Clusters in online-transactions or insurance claims.



Identification of patterns in cosmic-ray arrival directions using dynamic graph convolutional neural networks

4. Graph Classification & Regression

- Predict Global parameter associated to a graph; by training on a dataset of graphs
- **Example**: 1. Cosmic-Ray Composition analysis @ IceCube 2. Molecule type classification

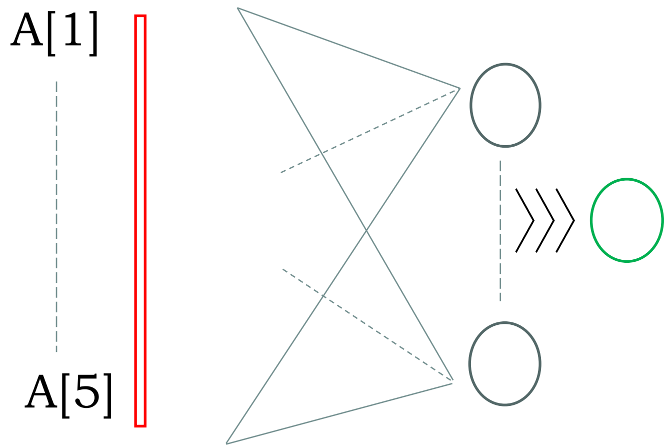


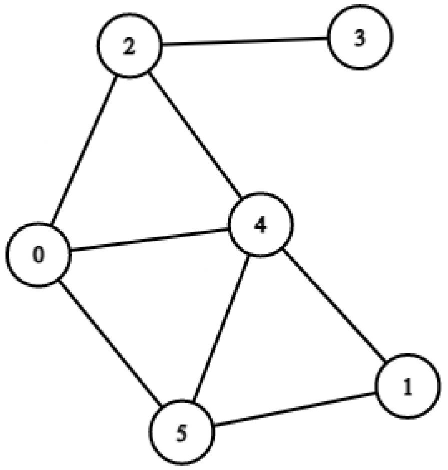
Graph Neural Network

A Possible Solution? – Graph Inputs as MLP:

Flatten the adjacency matrix rows, then concatenate all and train a MLP on that

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$



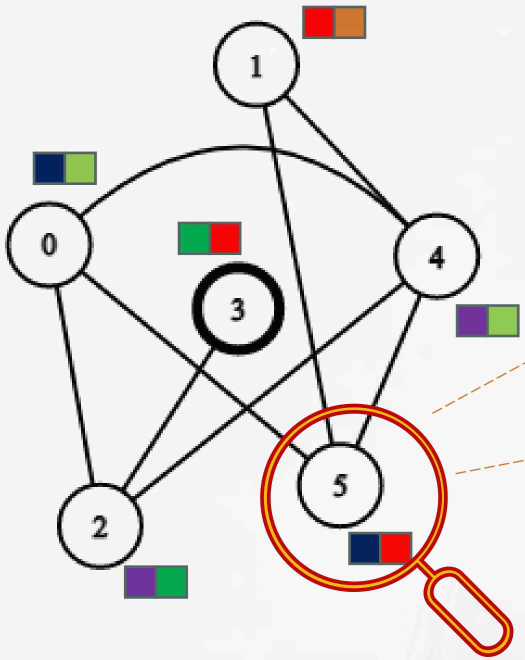


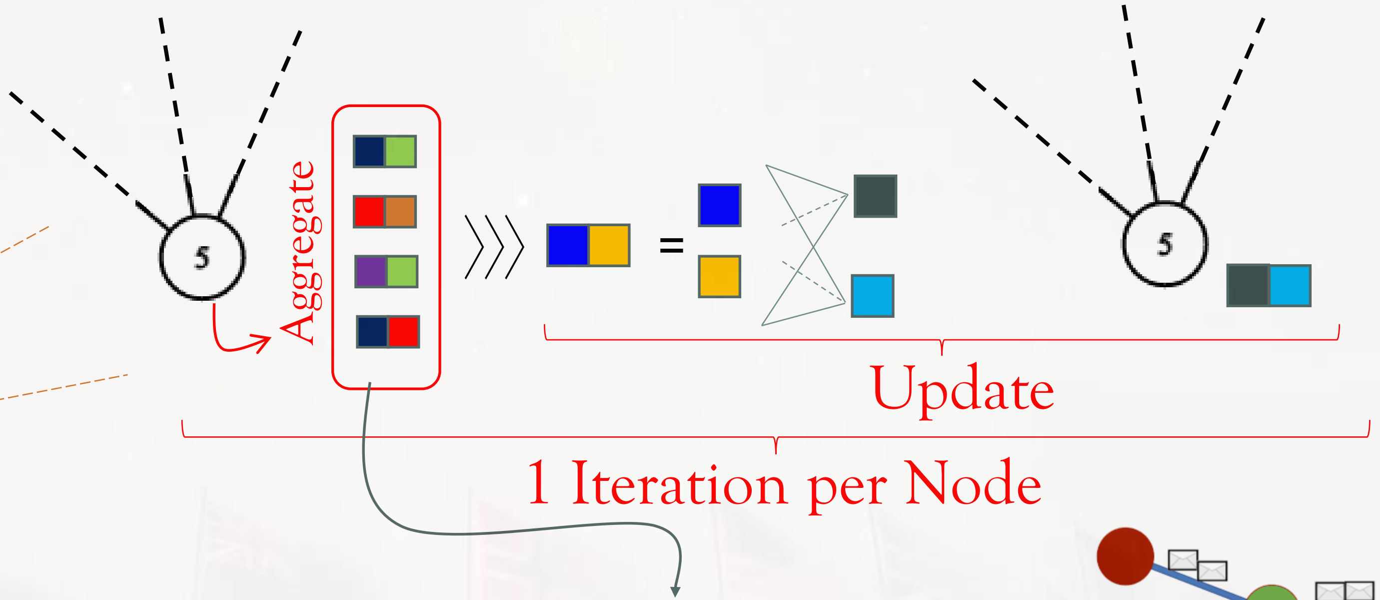
Issues with the Approach:

- Input depends on arbitrary ordering of nodes, hence it is not permutation invariant
- Impractical as the graph size grows
- Not benefitting from graph structure

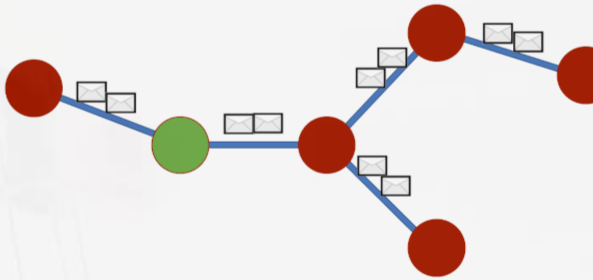
Tasks at Hand:

- Establish a Message passing framework
- Allows for suitable parameter learning and optimization to allow pattern learning
    - **Given *G(V,E)* along with node features:** Generate node embeddings / subgraph embeddings / for entire graph





Set as Input → Permutationally Invariant



Mathematical Formulation: Node Level

Hidden Embedding Corresponding to each Node

Graph Neighborhood

[1]

$$\mathbf{h}_u^{(k+1)} = \text{UPDATE}^{(k)}\left(\mathbf{h}_u^{(k)}, \text{AGGREGATE}^{(k)}(\{\mathbf{h}_v^{(k)}, \forall v \in \mathcal{N}(u)\})\right)$$

Aggregated Message

$$= \text{UPDATE}^{(k)}\left(\mathbf{h}_u^{(k)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)}\right),$$

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}_{\text{self}}^{(k)}\mathbf{h}_u^{(k-1)} + \mathbf{W}_{\text{neigh}}^{(k)} \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{(k-1)} + \mathbf{b}^{(k)}\right)$$ [1]

$$\mathbf{H}^{(t)} = \sigma\left(\mathbf{A}\mathbf{H}^{(k-1)}\mathbf{W}^{(k)}_{\text{neigh}} + \mathbf{H}^{(k-1)}\mathbf{W}^{(k)}_{\text{self}}\right)$$ [1]

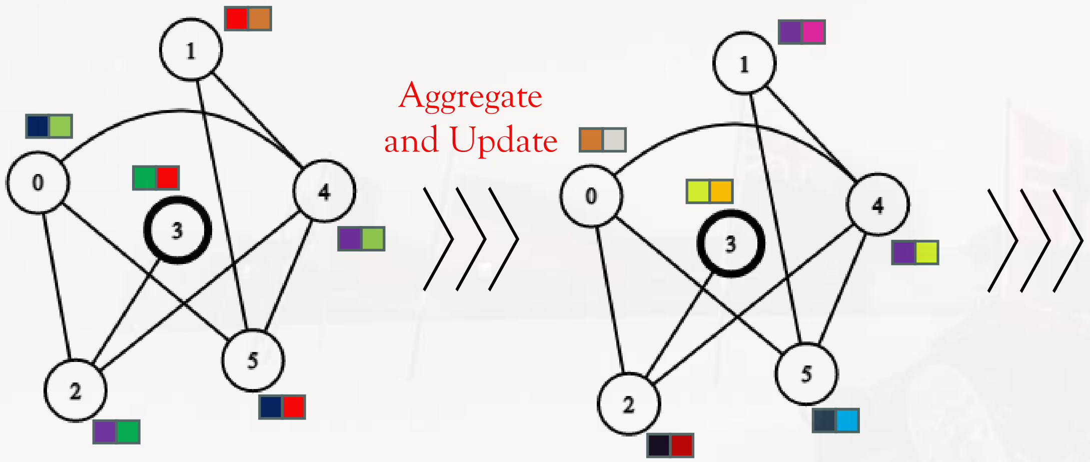
Aggregation

- Set as Input → Permutation Invariant
  - Sum of Neighbor's embeddings → Unstable and highly sensitive to node degree
  - Normalization by degree of node i.e. sum/degree
  - Symmetric normalization (Kipf and Welling - 2016)

Update

- Simplest Way → Linear combination of the node's current embedding with the message from its neighbors
  - Issue: Over-smoothing after multiple iterations
- Use a MLP with dropout

Graph Level Embedding

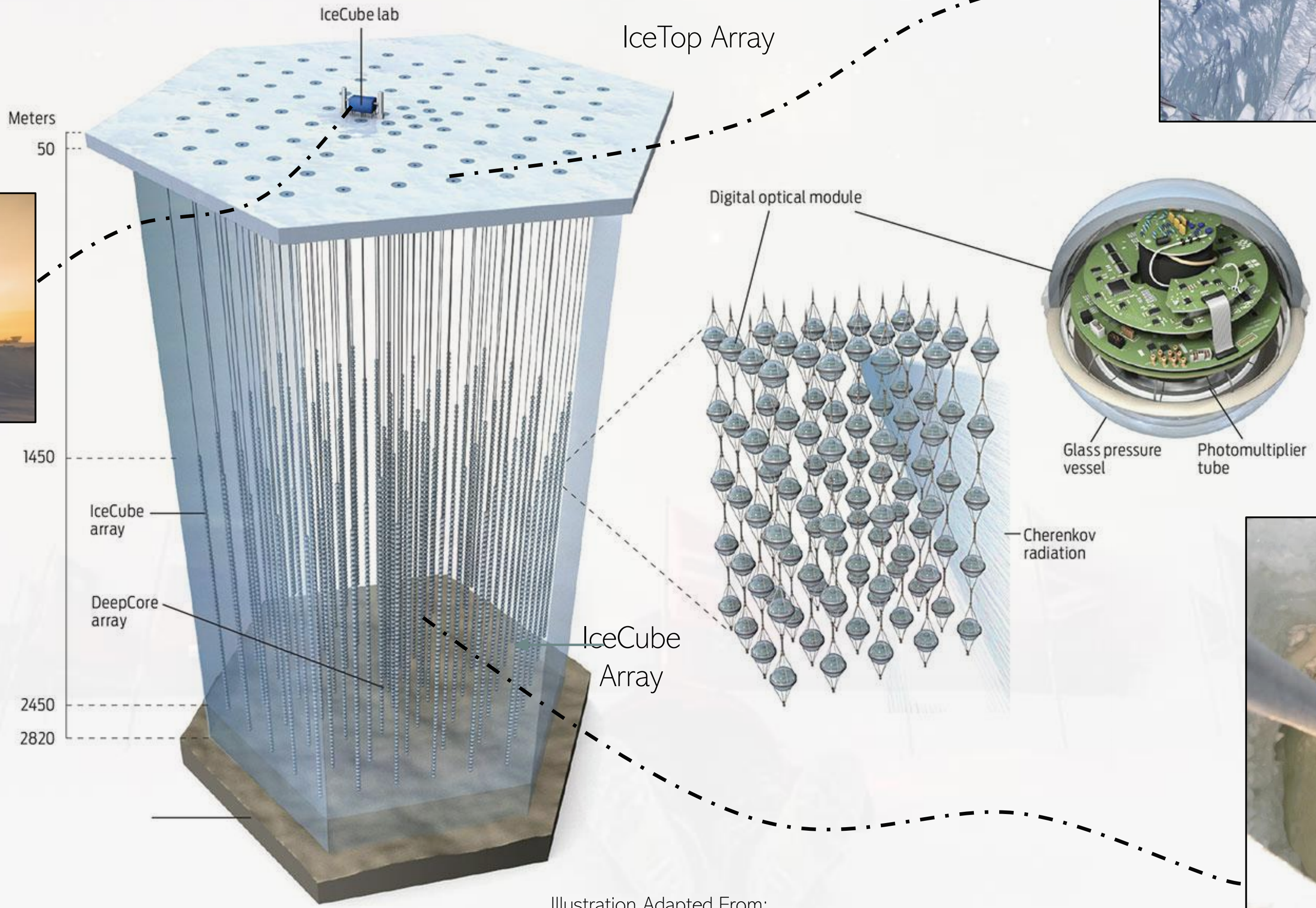


Possible Methods

- Sum or mean over node embeddings
- Attention based pooling

GNNs in Practice

Cosmic-Ray Composition Analysis @ IceCube



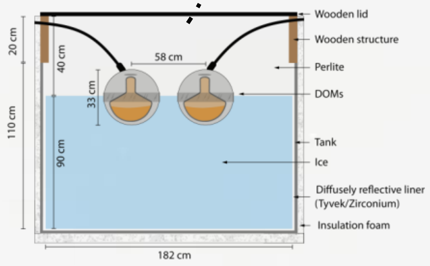
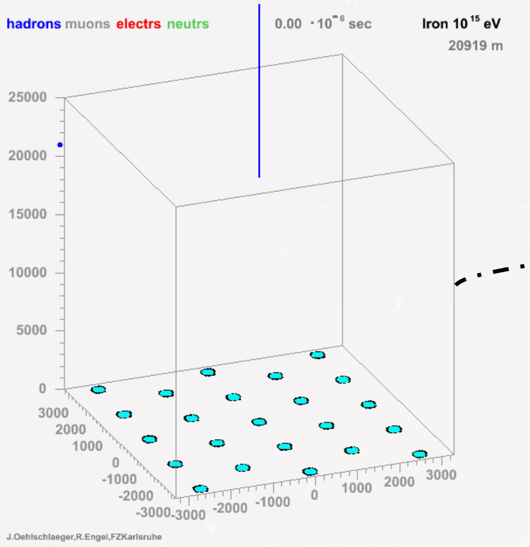
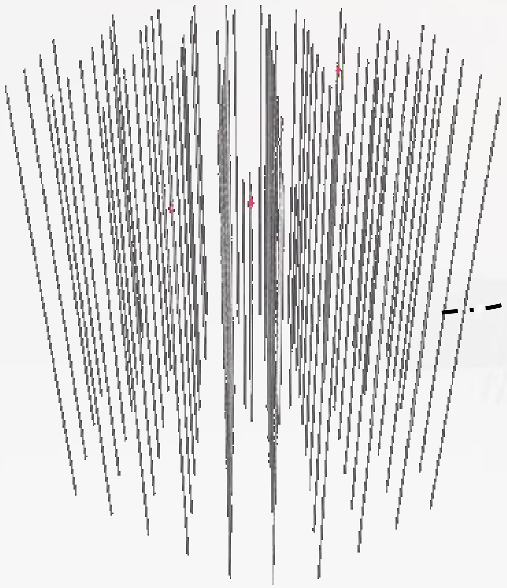
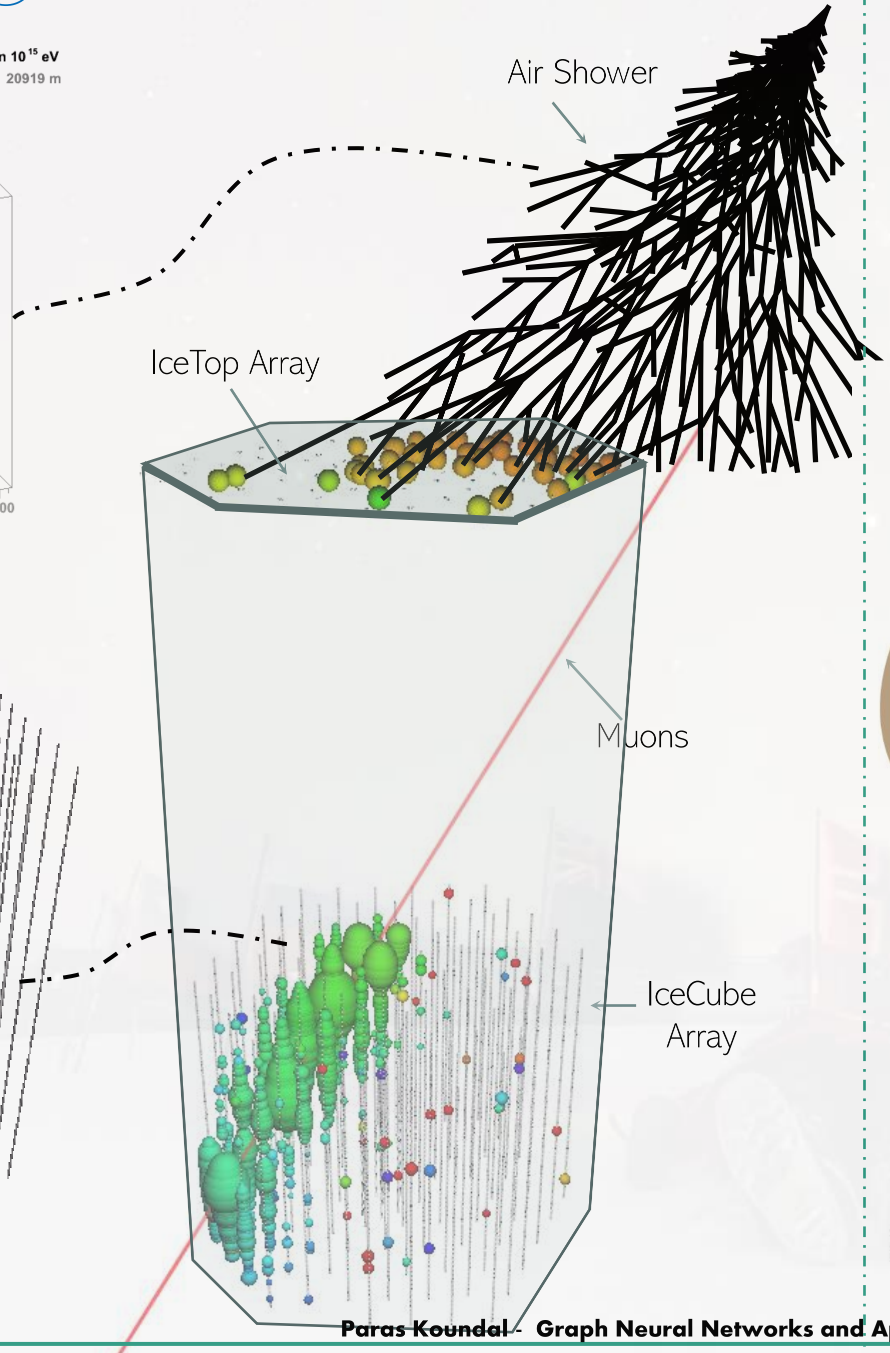


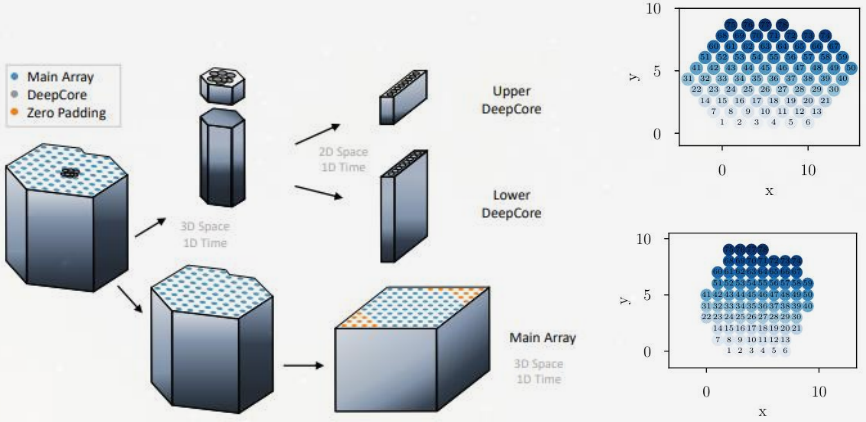
Illustration Adapted From:
Emily Cooper

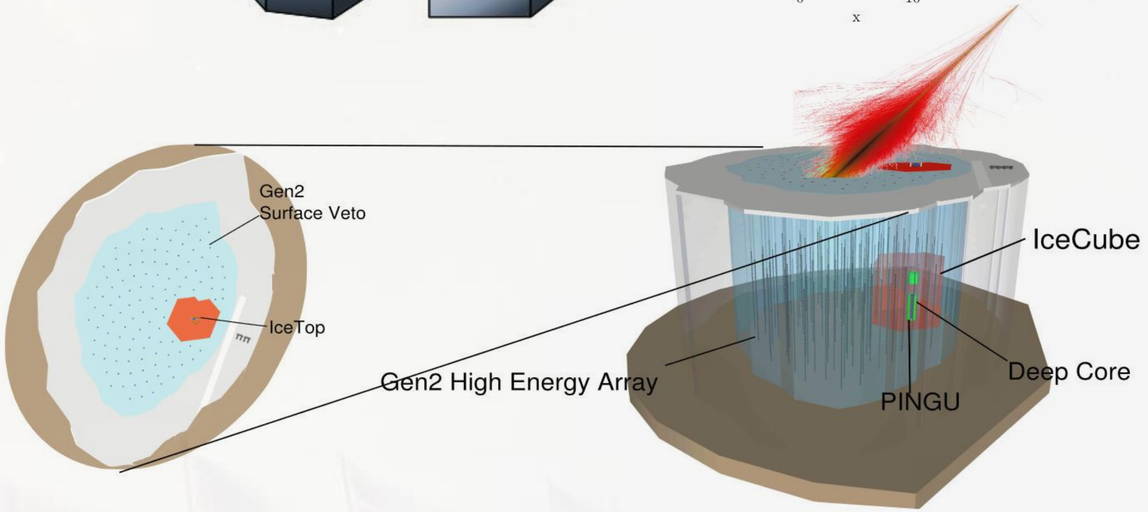
CR Analysis @ IceCube

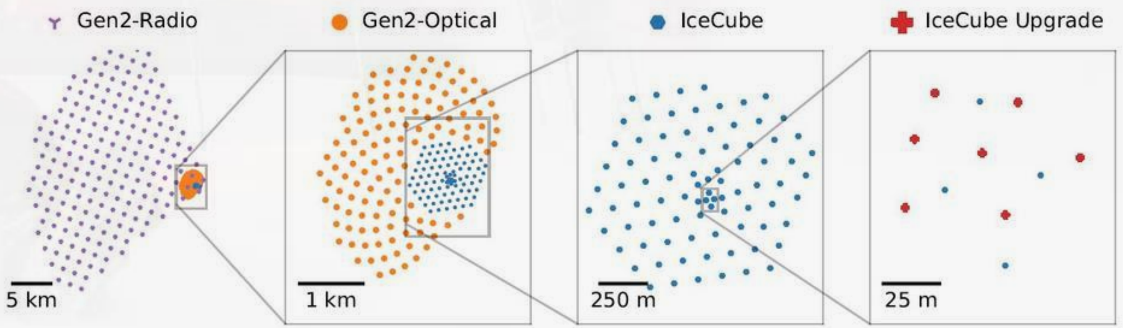






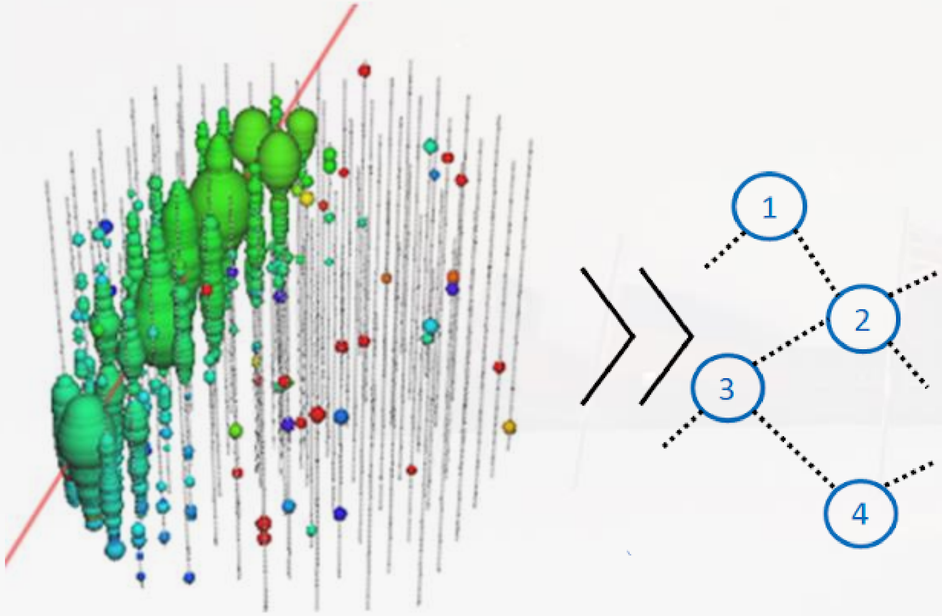




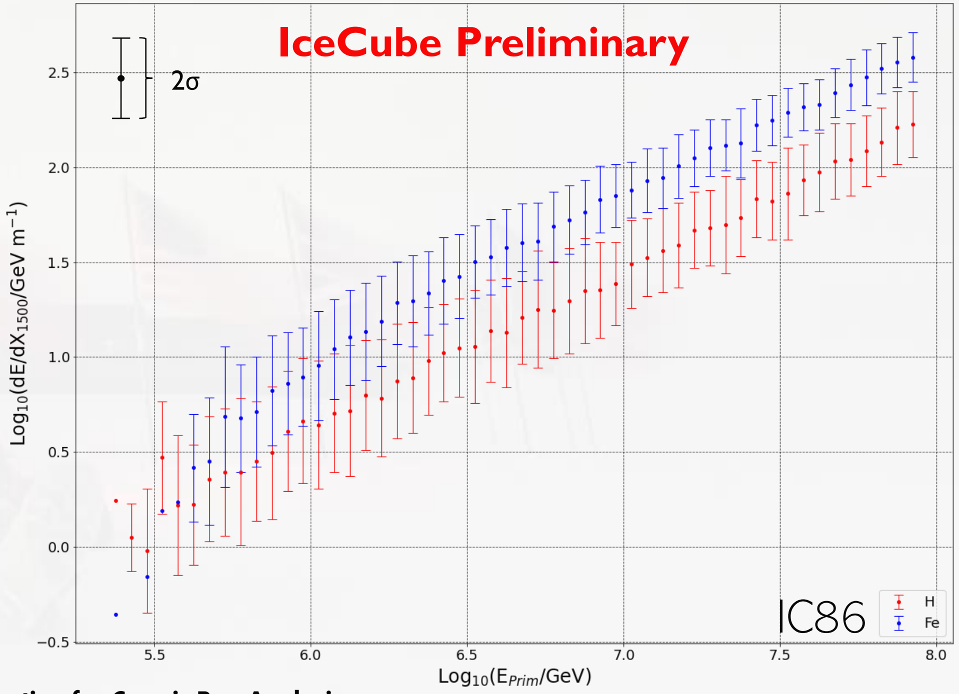


Things to Consider

- Signal mapped as graph – DOMs as Graph Nodes
- Connections:
  - Fully Connected Graph: Weigh Connection by the distance
  - Use spatial information: Find k-nearest neighbors and only connect them
    - Issue: Additional computation step; potentially time consuming
  - Use temporal information: Connect a node with nodes/DOMs which detected signal in a particular time window
    - Issue: Signal Deposit at a node/DOM can be spread over a longer interval
- For every event: Use preliminary information available that is useful for composition analysis

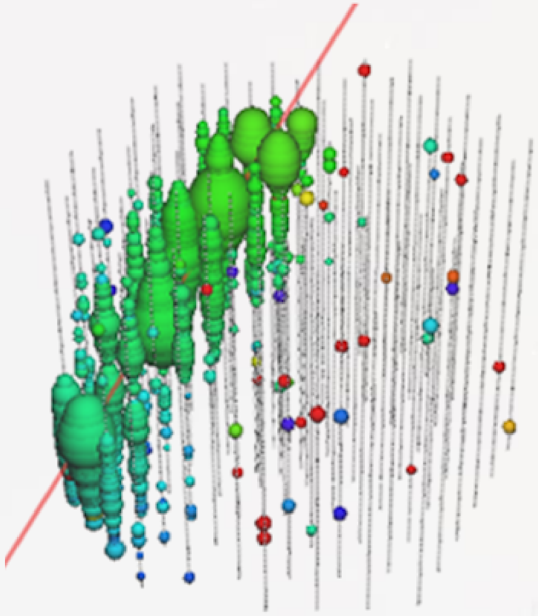


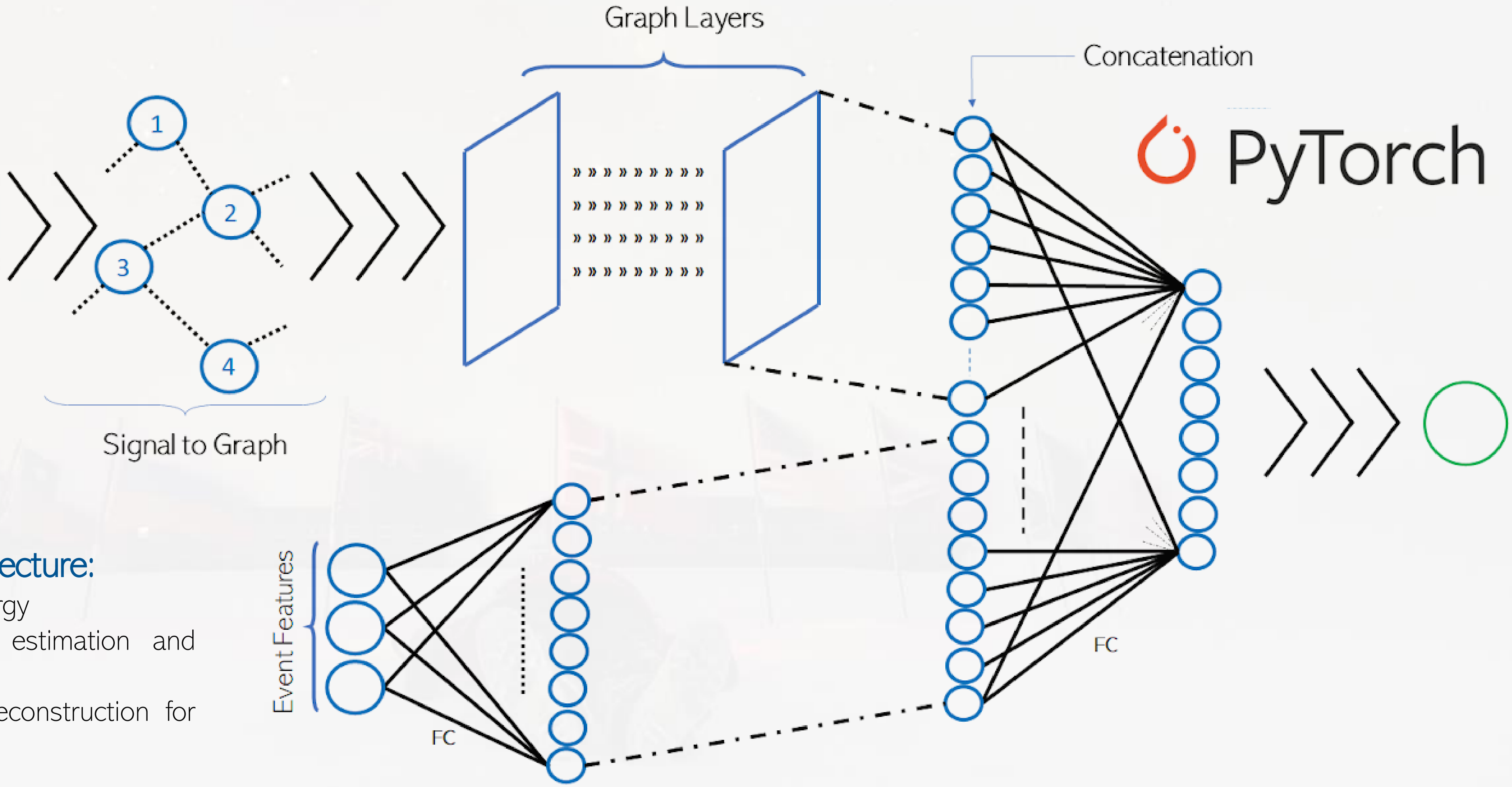
dE/dX_{1500} VS Primary Energy



Network Details

- Input :
  - For every event, n DOMs
  - For each Vertex, Input Features = Coordinates , Charge Measured ,Charge Time
  - Event Features: Cover Global Information about an event

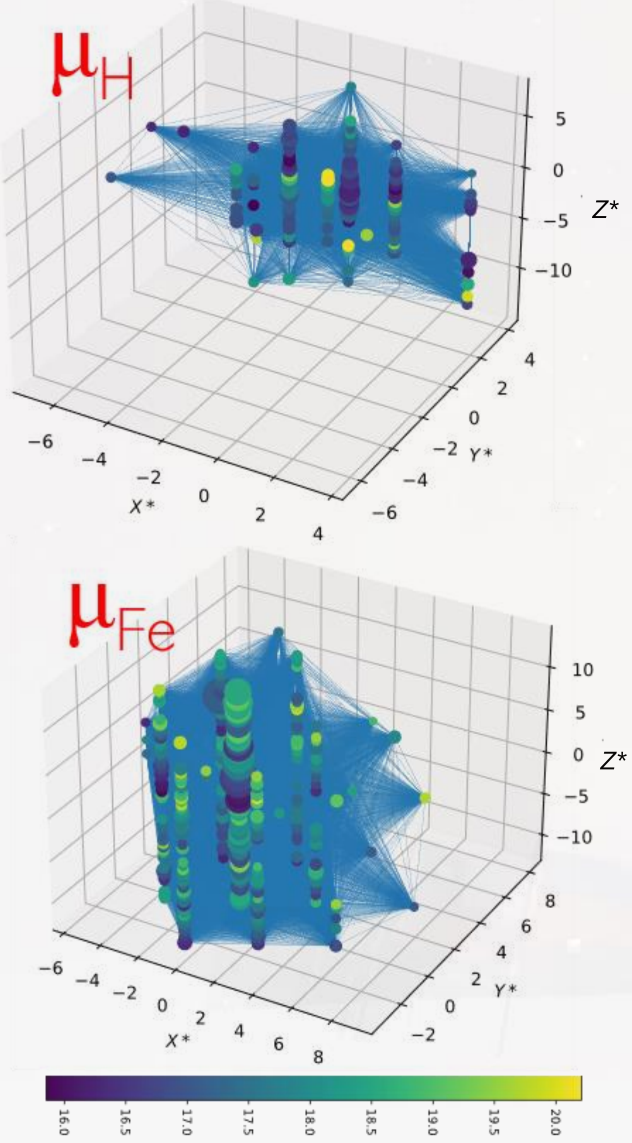




Motivation behind the architecture:

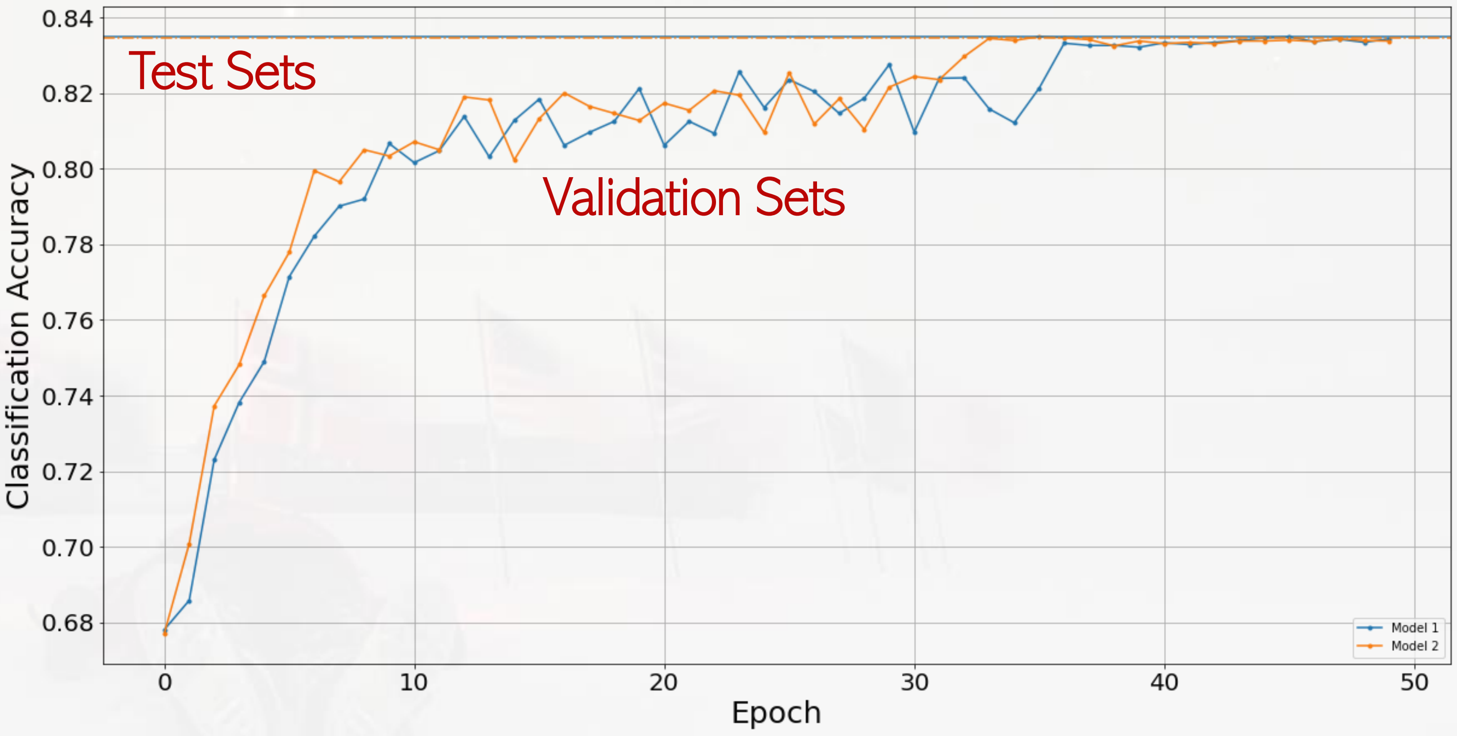
- Use IceTop and InIce in synergy
- Faster and simpler mass estimation and energy reconstruction
- Minimal reliance on prior reconstruction for network input-features.

Preliminary Results



The Preliminary-test is for binary classification of primary type. The final implementation will consist of all four primary types available in the simulation sets and instead of classification will do mass estimation.

Set Type	Number of Events
Training Set (70%)	28384 (50% H – 50% Fe)
Validation Set (20%)	8510
Test Set (10%)	4255
Total Events	41149



X*,Y*,Z* - Normalized Coordinates
Color – Time of Maximum Pulse
Radius – Charge amplitude of maximum pulse
Connections – Adjacency Matrix Connections

Summary and Outlook

- Flexible: Able to work on non-Euclidean data-type too.
- Graph based methods have applications across variety of disciplines
- GNNs can be viewed as generalization of CNNs + NNs
  - No need of pixelization
- Issues:
  - Normally slower than other methods
  - Problematic when working with big graphs
  - Relatively new field

- **GNNs at IceCube**
  - Graph Neural Networks based implementation is a possible future for detailed Cosmic-Ray based analysis at IceCube, aiding in more detailed and faster reconstructions.
  - Significant improvements in per-event based analysis will help to improve our understanding of high-energy cosmic rays.

Reference:

[1] Graph Representation Learning, William L. Hamilton McGill University 2020

Questions

paraskoundal.com/dlcp21