Development of graph-based ML techniques for optimized particle analysis at LHC

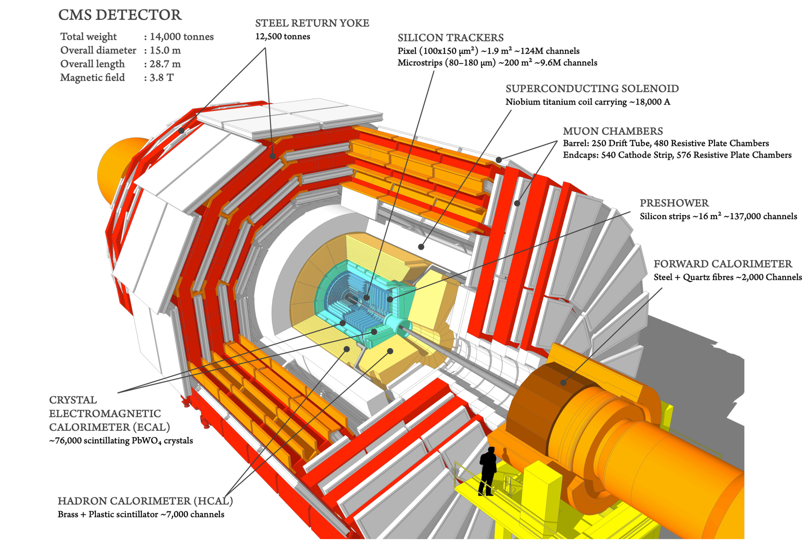
Shreyas Bakare*, Sourabh Dube

Department of Physcis, Indian Institute of Science Education and Research (IISER), Pune, India

(* Corresponding Author Email: shreyas.bakare@students.iiserpune.ac.in)

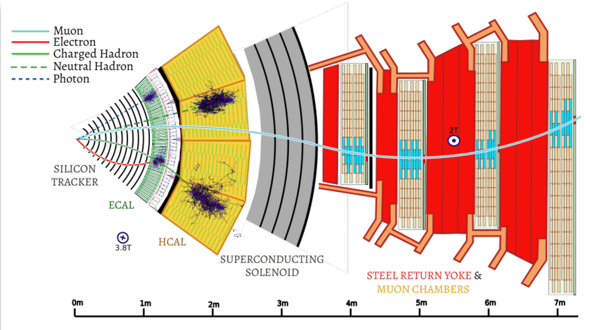
1. Detector overview

- HEP experiments like CMS at the LHC use sophisticated, multilayered detectors to precisely study p-p collisions.



- When two protons collide, various physics processes can occur, producing different particles.

- These particles interact uniquely with the CMS detector where their trajectories, energy deposits, and momentum are observed.

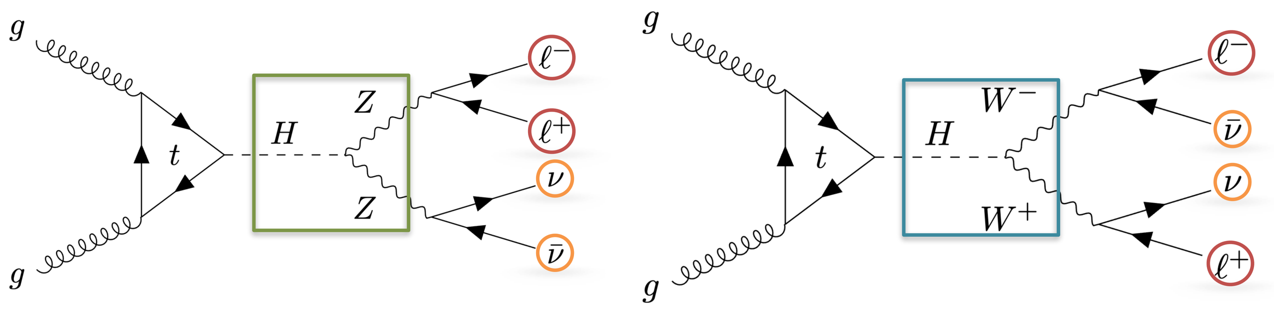


- However, different processes can yield similar signatures, making accurate identification of a process challenging.

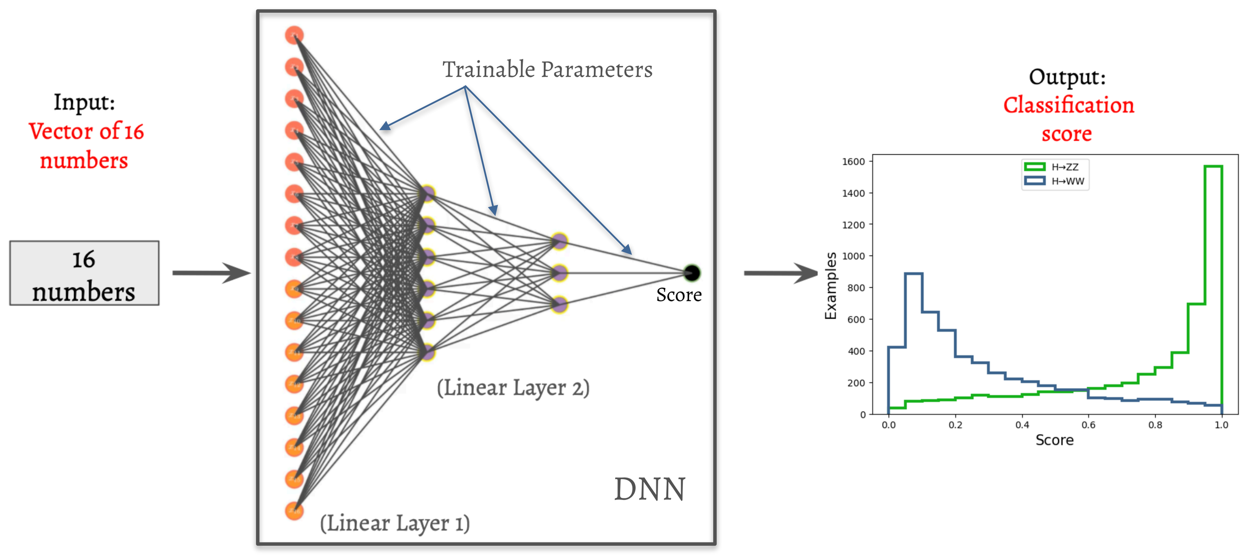
- For example:
Process P1 & process P2 have similar signatures, but say P2 occurs 10 times as often than P1. Then the key challenge is to identify P1 events while rejecting as many P2 events (which mimic P1 events) as possible.

2. Deep Neural Network (DNN)

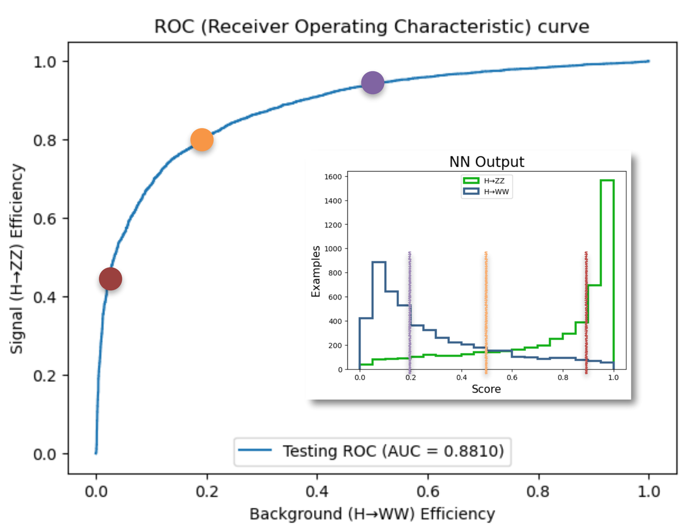
For example H→ZZ vs H→WW



- H→ZZ & H→WW both produce $2\ell\ 2\nu$ resulting in similar detector signatures.



- The threshold on the score depends on the details.



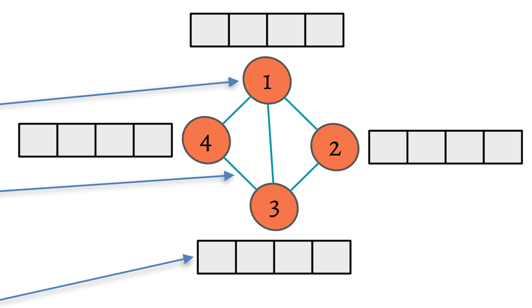
- Ideally we want to maximise the area under the ROC curve

(i.e. Identifying all true H→ZZ while ensuring no H→WW)

3. Graphs

A. Data structure:

Sets of elements (nodes) with some connections (edges) in between.



C. Adjacency matrix

B. Node feature matrix

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_5 & x_6 & x_7 & x_8 \\ x_9 & x_{10} & x_{11} & x_{12} \\ x_{13} & x_{14} & x_{15} & x_{16} \end{bmatrix}$$

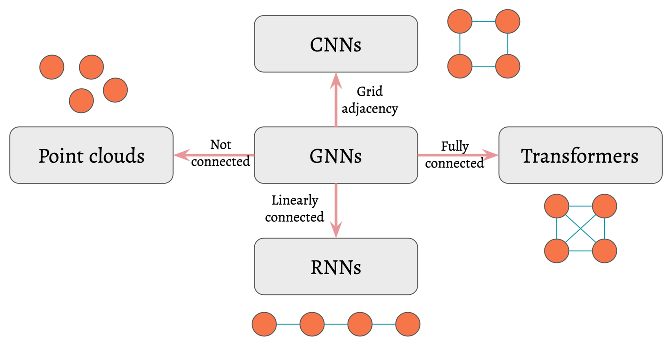
Node feature matrix

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 1 \\ 2 & 1 & 0 & 1 & 0 \\ 3 & 1 & 1 & 0 & 1 \\ 4 & 1 & 0 & 1 & 0 \end{array}$$

Adjacency matrix

can be non binary → weights

D. Reduction properties of graphs



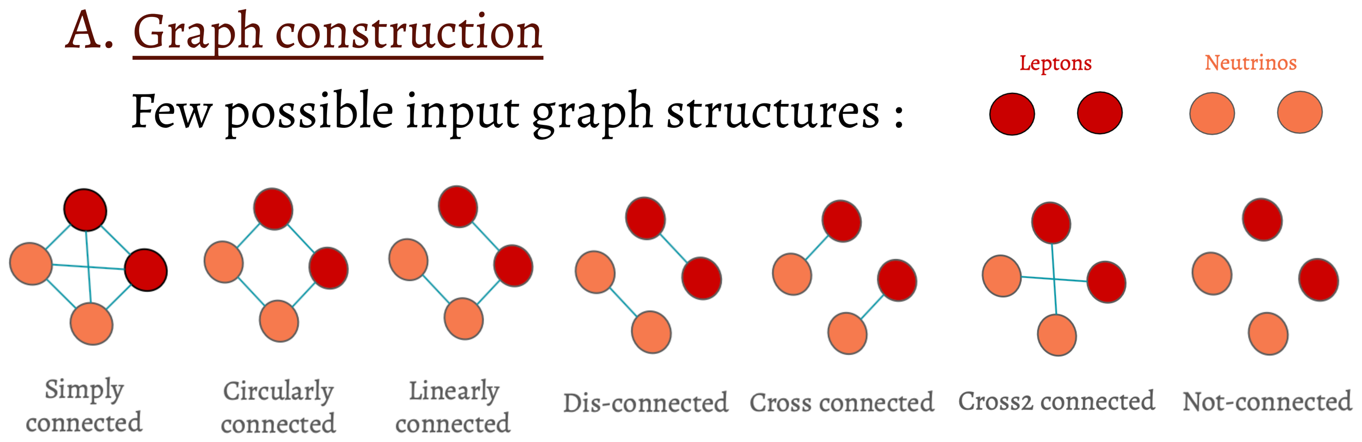
4. Graph Neural Network (GNN)

H→ZZ vs H→WW

- This can be tackled with GNN in three steps.

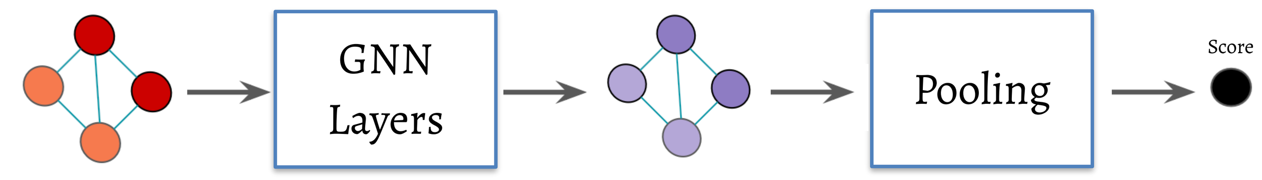
A. Graph construction

Few possible input graph structures :



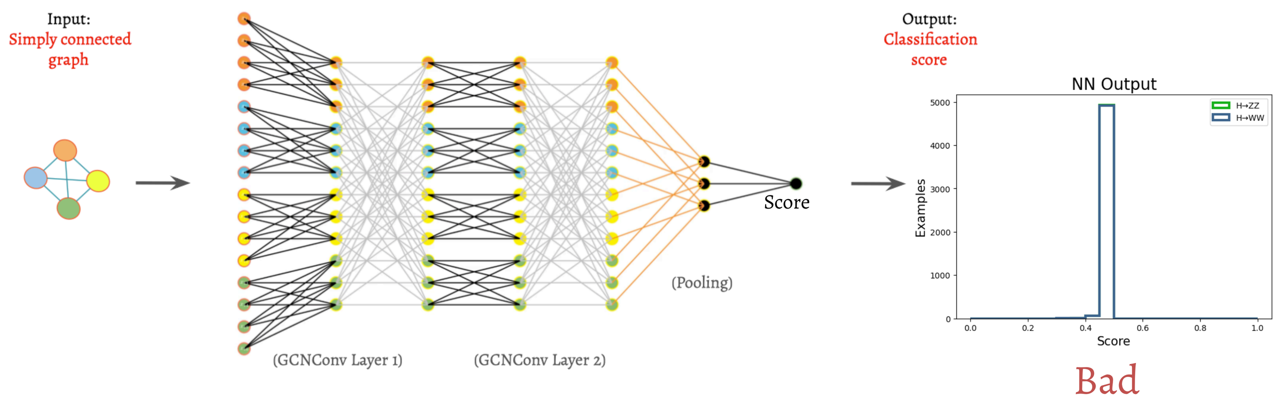
B. GNN inference

C. Post Processing

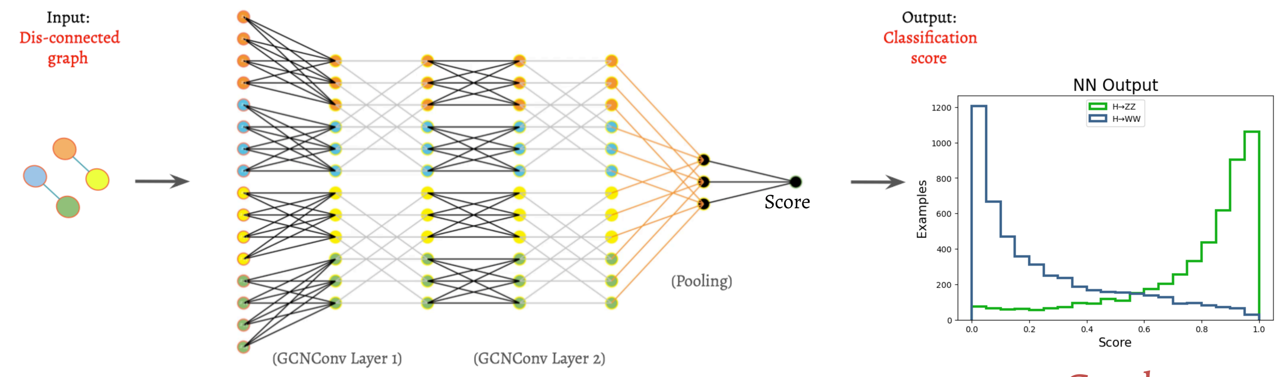


5. Choice of graph

- Changing input graph structure changes the outcome



Bad



Good

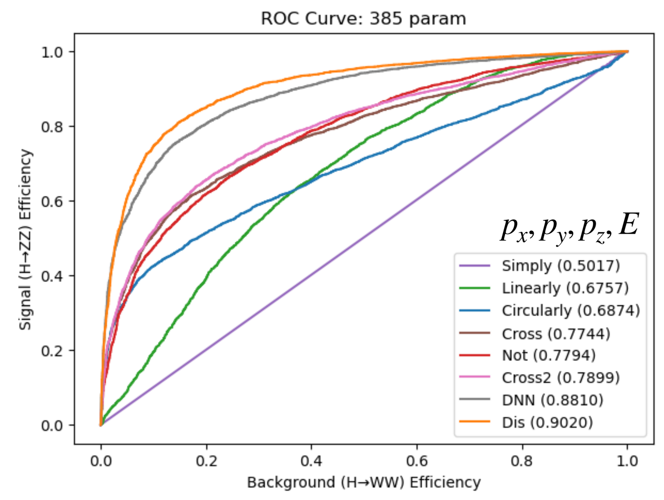
- Note: This illustration is for GCNConv Layer & max pooling

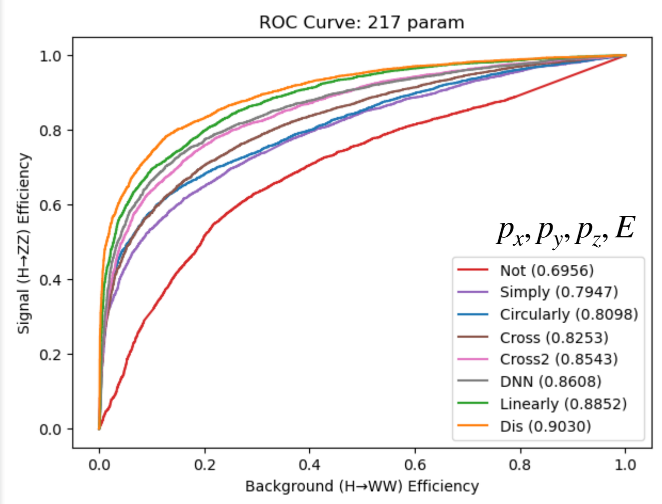
5. Results

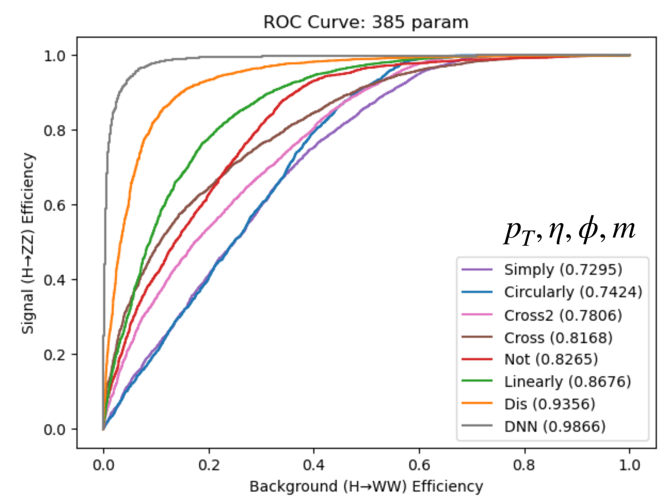
- Comparison of 7 input graphs for two GNN layers with DNN

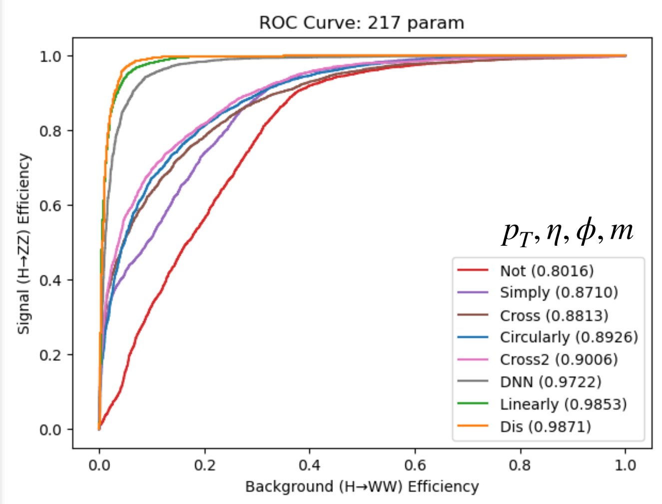
H→ZZ vs H→WW

GCNConv layers | GraphConv layers









- Also studied for H→ZZ vs ZZ, H→WW vs WW & ZZ vs WW

6. Conclusions

1. GNN leverages relational inductive bias, unlike DNN: Physics-informed graph choices (e.g., disconnected for Higgs vs. Higgs, simply connected for Higgs vs. non-Higgs) boost performance, while poor choices can hinder training.

2. GNNs handle flexible inputs, unlike DNNs: GNNs process varying input sizes and structures, and are independent of sequence order, whereas DNNs require fixed-length sequence inputs.

3. GraphConv outperforms GCNConv.

4. GCNConv seems to performs better for p_x, p_y, p_z, E than p_T, η, ϕ, m. DNN and GraphConv seem better for p_T, η, ϕ, m!

4. GNN requires lower training parameters than DNNs.

7. References

1. CMS Collaboration, "The CMS Experiment at the CERN LHC", *JINST*, 2008.

2. M. Feickert, B. Nachman, "A Living Review of Machine Learning for Particle Physics", *arXiv:2102.02770*, 2021.

3. T. N. Kipf, M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks", *arXiv:1609.02907*, 2017.

4. C. Morris et al., Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks, *arXiv:2006.09252*, 2021.

5. Pytorch Geometric
https://pyg.org/

6. Research group website