A Quick Introduction to Graph Neural Networks

Applied Machine Learning / 23th of May 2022

Rasmus Ørsøe

Technical University of Munich

Overview of this talk

- **Motivation for Graph Neural Networks**

- **Graph Neural Networks**
  - The “Graph” in Graph Neural Networks
  - Convolutions on Graphs

- **Examples of applications**
  - Computer Vision, Medicine, Condensed Matter Physics, Social Media & much more.
  - IceCube Neutrino Observatory

Motivation for GNN's

Motivation for GNN's

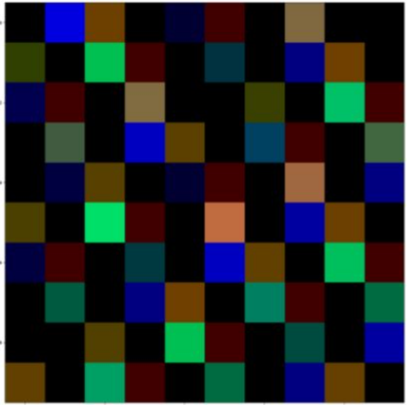
Let's consider a CNN.

They work on images – but what is an image really?

Motivation for GNN's

Let's consider a CNN.

They work on images – but what is an image really?



(image)

img - DataFrame

Index	0	0	0	0	0	0	0	0	0	0
0	[247, 19, 196]	[16, 71, 18]	[252, 62, 228]	[25, 198, 109]	[111, 138, 186]	[93, 19, 197]	[57, 207, 230]	[221, 32, 26]	[150, 119, 104]	[252, 132, 216]
1	[68, 79, 54]	[141, 173, 216]	[180, 112, 97]	[157, 39, 138]	[211, 163, 192]	[70, 171, 211]	[54, 203, 212]	[139, 72, 146]	[18, 5, 162]	[154, 48, 140]
2	[180, 123, 220]	[32, 131, 214]	[67, 83, 145]	[160, 225, 86]	[57, 248, 112]	[171, 157, 126]	[51, 115, 196]	[211, 253, 12]	[183, 115, 67]	[50, 151, 140]
3	[71, 68, 136]	[0, 132, 44]	[253, 86, 25]	[72, 0, 196]	[147, 69, 192]	[163, 44, 171]	[163, 153, 177]	[202, 189, 6]	[167, 252, 168]	[36, 195, 85]
4	[74, 120, 95]	[198, 76, 104]	[234, 62, 158]	[250, 95, 80]	[37, 236, 83]	[88, 137, 88]	[185, 165, 121]	[69, 223, 249]	[174, 86, 95]	[143, 47, 242]
5	[179, 149, 163]	[243, 117, 148]	[70, 186, 169]	[27, 15, 19]	[189, 249, 93]	[164, 253, 220]	[151, 33, 16]	[188, 39, 51]	[21, 27, 52]	[88, 36, 247]
6	[3, 70, 62]	[221, 207, 85]	[101, 65, 132]	[220, 57, 93]	[81, 79, 172]	[156, 242, 64]	[254, 101, 204]	[217, 22, 242]	[46, 242, 77]	[84, 171, 211]
7	[209, 204, 203]	[207, 158, 158]	[212, 248, 11]	[221, 112, 140]	[100, 107, 158]	[118, 122, 252]	[50, 203, 104]	[200, 198, 40]	[254, 227, 104]	[65, 219, 94]
8	[22, 48, 213]	[52, 111, 59]	[215, 155, 231]	[236, 211, 160]	[80, 44, 150]	[22, 202, 149]	[93, 242, 65]	[65, 183, 243]	[43, 91, 98]	[210, 153, 236]
9	[29, 189, 13]	[217, 78, 223]	[27, 104, 151]	[124, 21, 39]	[14, 123, 89]	[18, 33, 23]	[218, 208, 211]	[115, 29, 223]	[26, 171, 241]	[69, 216, 19]

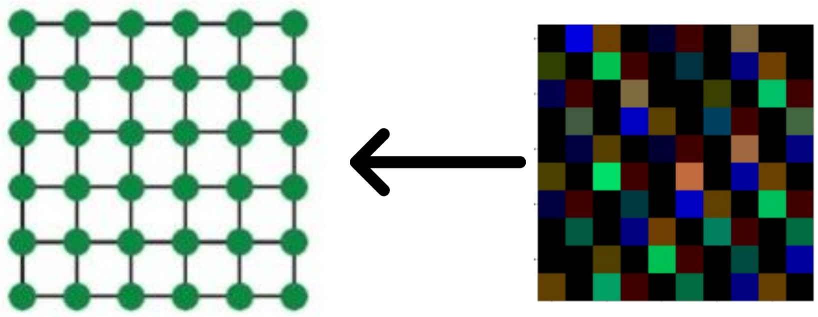
(data)

Motivation for GNN’s

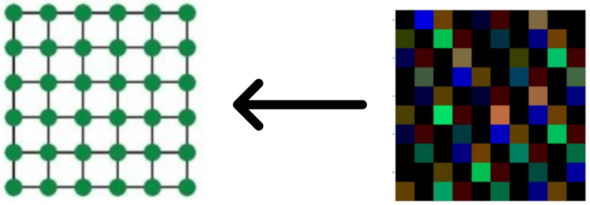
Images in an abstract sense:

- Images are grid-like structures, where the distance between neighbouring points in the grid is constant

- At every point in the grid we associate values ‘red’, ‘green’, ‘blue’



Motivation for GNN's



What if our data isn't exactly an image, but we'd like to use a CNN anyway?

This isn't an uncommon problem. CNNs have been used in physics experiments despite the data not being images.

https://arxiv.org/pdf/2101.11589.pdf and https://arxiv.org/pdf/2101.11589.pdf are fine examples!

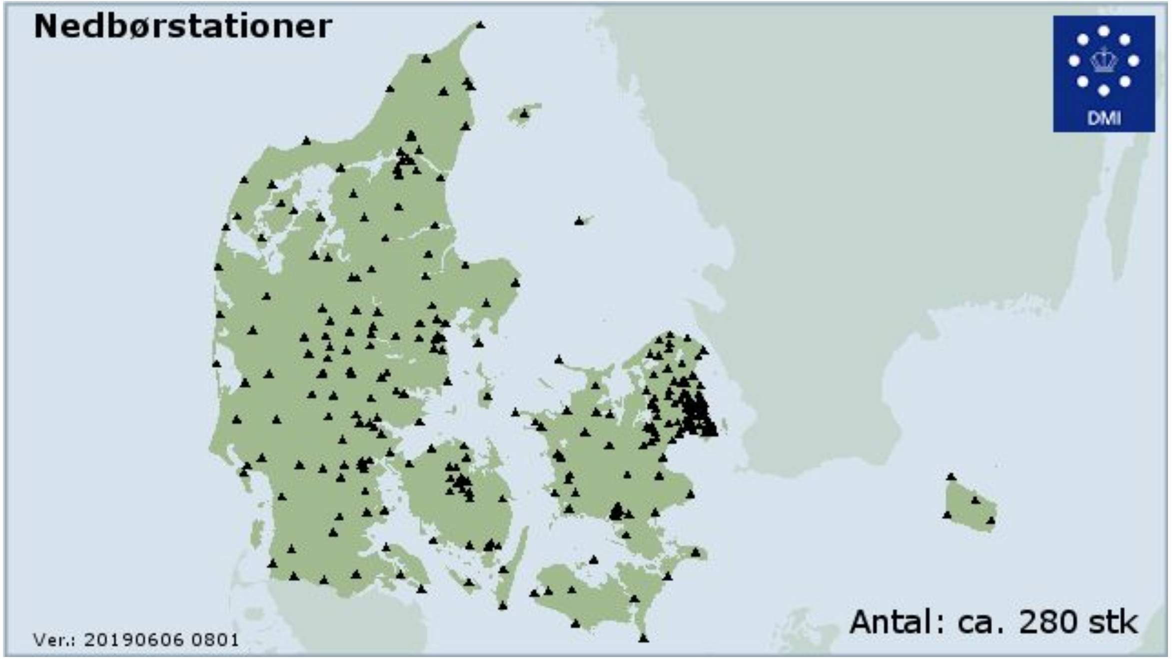
Motivation for GNN's

Suppose we had variables:

[*windspeed*, *temperature*, *humidity*, *lattitude*,*longitude*] (5 variables – not 3!)

from the DMI weather stations

How would the image look like?



Motivation for GNN's

We could add the extra information to each pixel!

```
TypeError: Cannot handle this data type:
```

(Image)

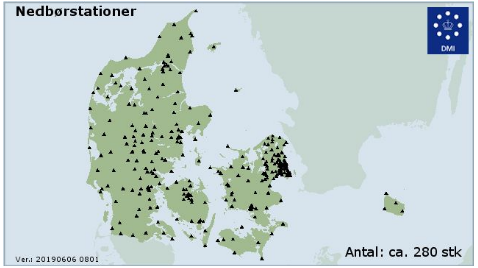
img - DataFrame

Index	0	0	0	0	0
0	[111, 22, 71, 97, 127]	[56, 2, 41, 95, 115]	[177, 107, 251, 209, 211]	[13, 76, 15, 40, 80]	[240, 238, 149, 66, 232]
1	[248, 164, 252, 52, 174]	[140, 28, 152, 111, 23]	[224, 16, 113, 187, 150]	[232, 248, 160, 180, 152]	[186, 80, 30, 204, 164]
2	[144, 53, 227, 97, 96]	[181, 65, 22, 161, 31]	[68, 133, 73, 77, 220]	[170, 87, 81, 114, 102]	[156, 54, 52, 96, 178]
3	[96, 210, 202, 103, 50]	[123, 33, 227, 33, 113]	[174, 148, 12, 136, 61]	[146, 133, 220, 81, 116]	[139, 30, 83, 113, 242]
4	[241, 79, 75, 129, 152]	[28, 236, 177, 154, 118]	[43, 209, 227, 145, 115]	[55, 200, 52, 201, 211]	[0, 197, 116, 238, 32]
5	[125, 241, 248, 226, 7]	[161, 164, 37, 171, 74]	[186, 42, 37, 43, 89]	[28, 123, 42, 7, 96]	[55, 49, 170, 90, 2]
6	[84, 210, 234, 246, 233]	[166, 64, 119, 227, 2]	[242, 34, 167, 197, 14]	[6, 4, 175, 251, 223]	[115, 10, 37, 237, 105]
7	[33, 78, 149, 114, 168]	[119, 51, 244, 229, 15]	[172, 251, 172, 99, 66]	[206, 238, 94, 162, 244]	[49, 69, 9, 76, 210]
8	[23, 245, 236, 86, 210]	[152, 251, 9, 106, 179]	[170, 21, 117, 234, 22]	[142, 85, 137, 57, 65]	[179, 145, 24, 8, 134]
9	[145, 202, 136, 75, 146]	[138, 222, 141, 180, 88]	[114, 239, 153, 221, 25]	[246, 25, 27, 217, 158]	[195, 205, 25, 92, 57]

(data)

But is this the most natural way of incorporating the geometric information?

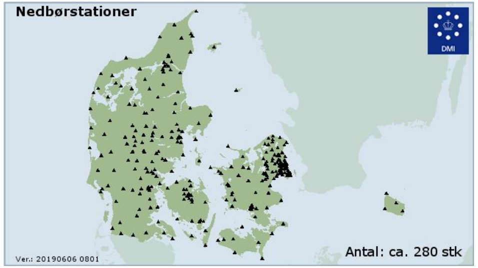
Motivation for GNN's



No!

By turning the weather stations into pixels in an image we're indirectly saying that the distance between neighbouring stations are constant!

Motivation for GNN's



Key Point:

By forcing problems with irregular geometry into images, we're shaping the problem to the tool, and not the tool to the problem!

Is there a different structure that has no underlying assumption on the geometry of the data?

- Yes! This is **Graphs**!

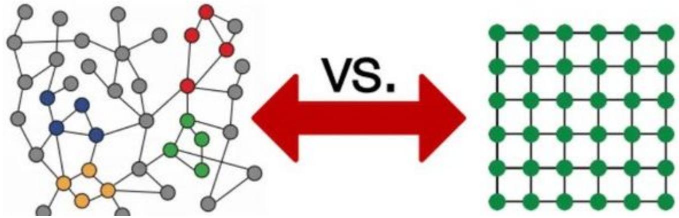
Graph Neural Networks

Graph Neural Networks

GNNs are an emerging tool in data science

- Input data is a graph
  - A graph is a collection of two things:
    - Nodes (“pixels”)
    - Edges (connections)
- Graphs have **<u>no</u>** underlying assumption on the geometry of the data. **<u>You</u>** need to specify the geometry directly using the edges!

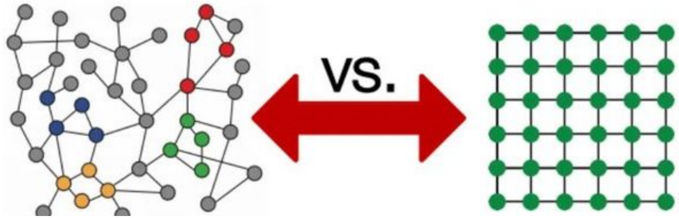
Convolutional Neural Networks are a special case of Graph Neural Networks!



Graph Neural Networks

Many of the machine learning techniques (or “layers”) that you have been introduced to are also available for graphs!

- Convolutions
- LSTM, GRU
- Attention
- Auto-Encoders
- Data-driven pooling operators
- etc..



Graph Neural Networks

How would a graph convolutional neural network (GCNN) work?

The increased generality of GNN's means that convolution, as understood from CNNs, can be interpreted in multiple ways. The many types of convolutions differ on (mainly) the way in which the edges are utilized.

Let's pick **Edge Convolution**!

(https://arxiv.org/abs/1801.07829)

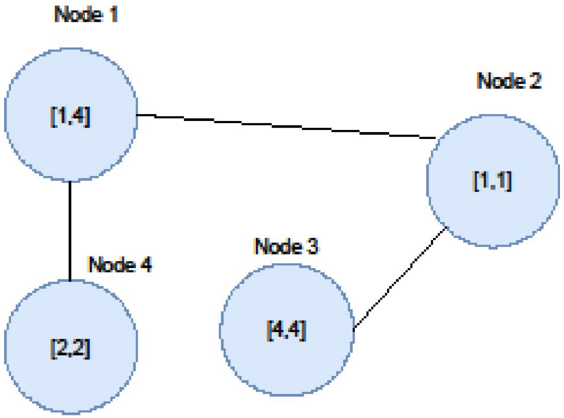
EdgeConv

EdgeConv ‘convolutes’ the graph by updating the values in each node in the graph by considering the values in the nodes that it is connected to. (So no filters!)

The update of values of the j’th node is done via

$$\hat{x}_j = \sum_{k=1}^{n} f(x_j, x_j - x_k)$$

Where f is a learned function (a neural net)



EdgeConv

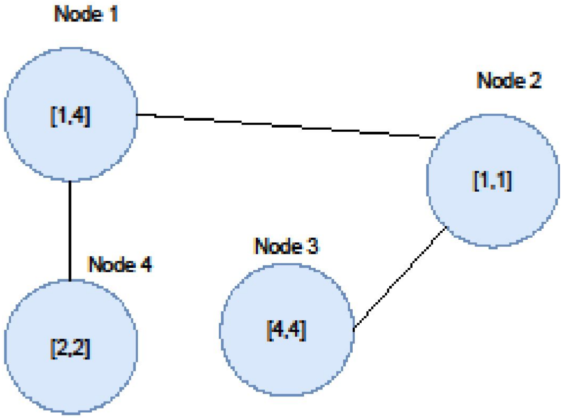
The update of values of the j'th node is done via

$$\hat{x}_j = \sum_{k=1}^{n} f(x_j, x_j - x_k)$$

Where f is a learned function (e.g a neural net)

Suppose f = 1 * x + 0. Then the updated values for Node 1 would be:

$$\begin{aligned}
\hat{x}_1 &= f(x_1, x_1 - x_2) + f(x_1, x_1 - x_4) \\
&= f([1,4],[1,4] - [1,1]) + f([1,4],[1,4] - [2,2]) \\
&= f([1,4],[0,3]) + f([1,4],[-1,2]) \\
&= f([1,4,0,3]) + f([1,4,-1,2]) \quad \text{(by concat.)} \\
&= 1 \cdot [1,4,0,3] + 1 \cdot [1,4,-1,2] \\
&= [2,8,-1,5]
\end{aligned}$$



In a full forward pass, this would be iterated for every node in the graph!

Examples of applications

Computer Vision (EdgeConv)

Dynamic Graph CNN for Learning on Point Clouds • 1:9

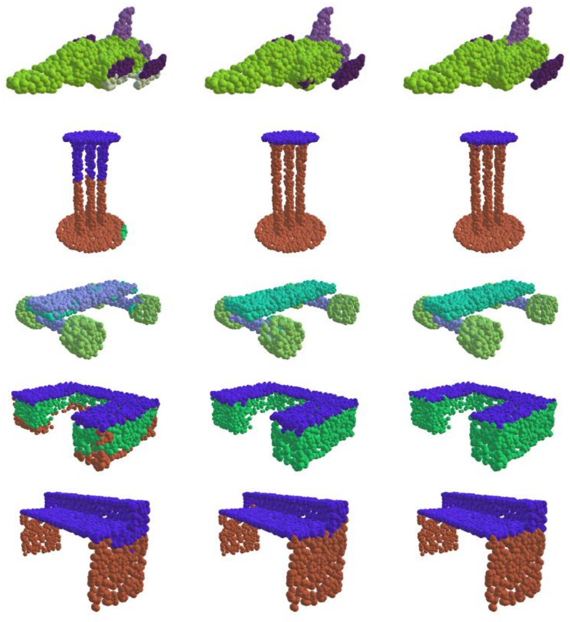
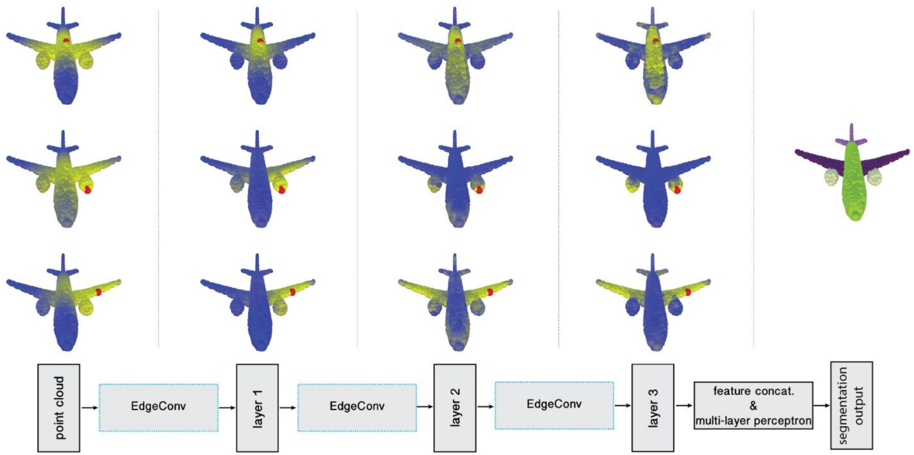


Fig. 7. Compare part segmentation results. For each set, from left to right: PointNet, ours and ground truth.

Dynamic Graph CNN for Learning on Point Clouds

YUE WANG, Massachusetts Institute of Technology
YONGBIN SUN, Massachusetts Institute of Technology
ZIWEI LIU, UC Berkeley / ICSI
SANJAY E. SARMA, Massachusetts Institute of Technology
MICHAEL M. BRONSTEIN, Imperial College London / USI Lugano
JUSTIN M. SOLOMON, Massachusetts Institute of Technology



(https://arxiv.org/pdf/1801.07829.pdf)

Graph Neural Networks with Continual Learning for Fake News Detection from Social Media

Yi Han, Shanika Karunasekera, Christopher Leckie

Although significant effort has been applied to fact-checking, the prevalence of fake news over social media, which has profound impact on justice, public trust and our society, remains a serious problem. In this work, we focus on propagation-based fake news detection, as recent studies have demonstrated that fake news and real news spread differently online. Specifically, considering the capability of graph neural networks (GNNs) in dealing with non-Euclidean data, we use GNNs to differentiate between the propagation patterns of fake and real news on social media. In particular, we concentrate on two questions: (1) Without relying on any text information, e.g., tweet content, replies and user descriptions, how accurately can GNNs identify fake news? Machine learning models are known to be vulnerable to adversarial attacks, and avoiding the dependence on text-based features can make the model less susceptible to the manipulation of advanced fake news fabricators. (2) How to deal with new, unseen data? In other words, how does a GNN trained on a given dataset perform on a new and potentially vastly different dataset? If it achieves unsatisfactory performance, how do we solve the problem without re-training the model on the entire data from scratch? We study the above questions on two datasets with thousands of labelled news items, and our results show that: (1) GNNs can achieve comparable or superior performance without any text information to state-of-the-art methods. (2) GNNs trained on a given dataset may perform poorly on new, unseen data, and direct incremental training cannot solve the problem---this issue has not been addressed in the previous work that applies GNNs for fake news detection. In order to solve the problem, we propose a method that achieves balanced performance on both existing and new datasets, by using techniques from continual learning to train GNNs incrementally.

https://arxiv.org/abs/2007.03316v2

Medicine

Interpretable Drug Synergy Prediction with Graph Neural Networks for Human-AI Collaboration in Healthcare

Zehao Dong, Heming Zhang, Yixin Chen, Fuhai Li

We investigate molecular mechanisms of resistant or sensitive response of cancer drug combination therapies in an inductive and interpretable manner. Though deep learning algorithms are widely used in the drug synergy prediction problem, it is still an open problem to formulate the prediction model with biological meaning to investigate the mysterious mechanisms of synergy (MoS) for the human-AI collaboration in healthcare systems. To address the challenges, we propose a deep graph neural network, IDSP (Interpretable Deep Signaling Pathways), to incorporate the gene-gene as well as gene-drug regulatory relationships in synergic drug combination predictions. IDSP automatically learns weights of edges based on the gene and drug node relations, i.e., signaling interactions, by a multi-layer perceptron (MLP) and aggregates information in an inductive manner. The proposed architecture generates interpretable drug synergy prediction by detecting important signaling interactions, and can be implemented when the underlying molecular mechanism encounters unseen genes or signaling pathways. We test IDWSP on signaling networks formulated by genes from 46 core cancer signaling pathways and drug combinations from NCI ALMANAC drug combination screening data. The experimental results demonstrated that 1) IDSP can learn from the underlying molecular mechanism to make prediction without additional drug chemical information while achieving highly comparable performance with current state-of-art methods; 2) IDSP show superior generality and flexibility to implement the synergy prediction task on both transductive tasks and inductive tasks. 3) IDSP can generate interpretable results by detecting different salient signaling patterns (i.e. MoS) for different cell lines.

https://arxiv.org/abs/2105.07082

Drug Synergy:
” An interaction between two or more drugs that causes the total effect of the drugs to be greater than the sum of the individual effects of each drug. A synergistic effect can be beneficial or harmful.”

IceCube Neutrino Observatory

Applications of GraphNeT:dynedge

Direction Reconstruction: GNN compared with CNN

GraphNeT Team

Niels Bohr Institute

Troels Petersen (Assoc. Prof.)

Andreas Søgaard (PostDoc)

Kaare Iversen (MSc)

Leon Bozianu (MSc)

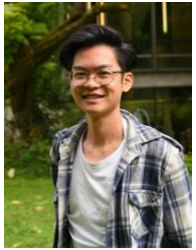
Morten Holm (MSc)

You?

You!

Technical University Munich

Philipp Eller (PostDoc)



Martin Ha Minh (PhD)

Rasmus Ørsøe (PhD)

Tim Guggenmos (BSc)

Write your thesis with us!

GraphNeT

Graph Neural Networks for Neutrino Event Reconstruction

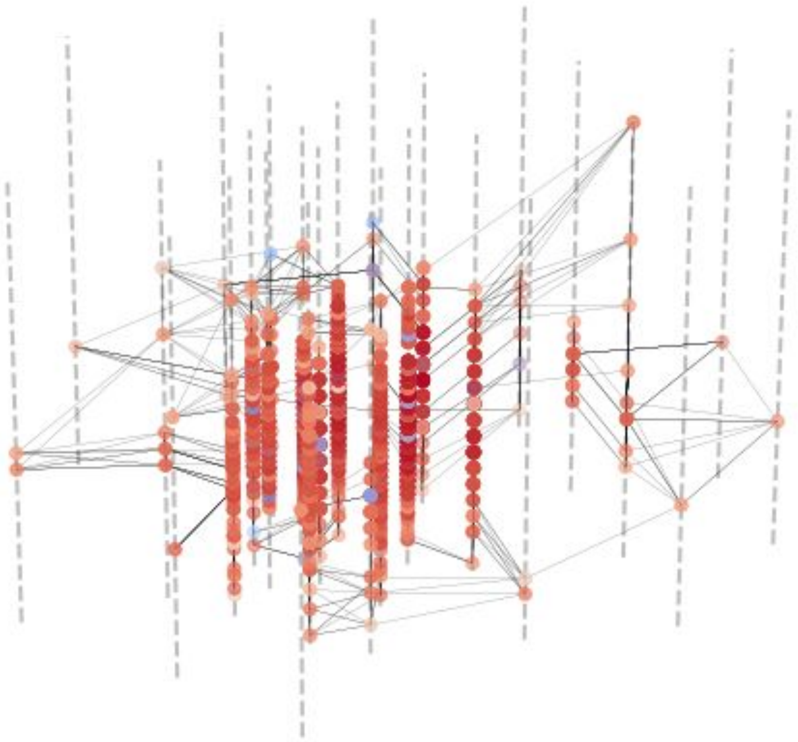
- **A framework for developing GNN-based tools for neutrino telescopes**
- **One stop shop: from model development to deployment**
  - **Developers**
    - Everything needed to build, train and validate GNNs from scratch
  - **End-Users**
    - Can choose from a library of pre-trained models and apply them as IceTray modules
- **Actively maintained with modern industry-standard code practices**
  - code conventions
  - proper documentation
  - Unit tests
- **Funded for next four years - you can safely migrate!**

icecube/graphnet

Graph Neural Networks

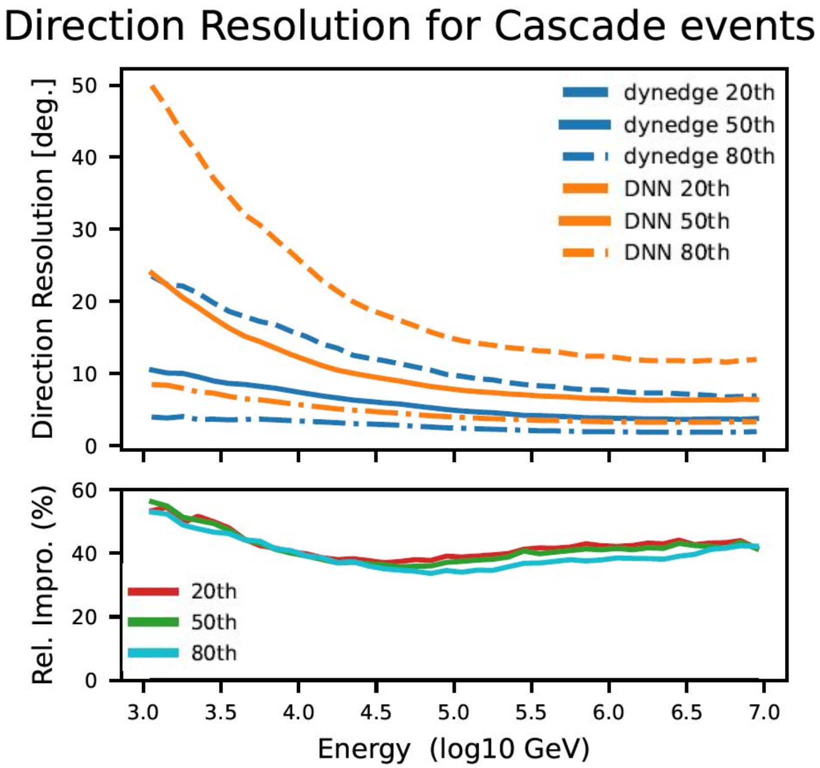
GraphNeT:dynedge represents an event as a **graph**

- **Nodes** in the graph is a photosensor in the ice
- **Edges** are drawn to a node's 8 nearest Euclidean neighbours based on photosensors position



3.4 TeV ν_e event as graph with k = 8

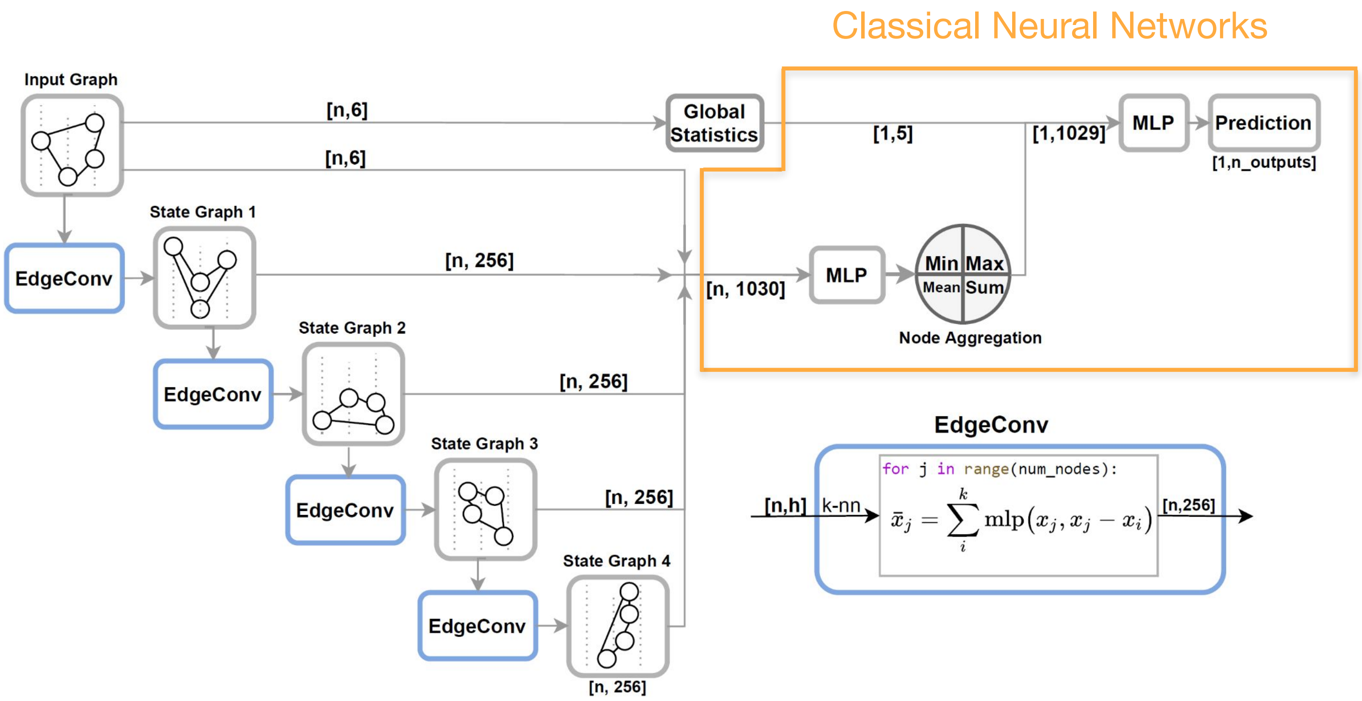
IceCube Neutrino Observatory



Thank you for listening!

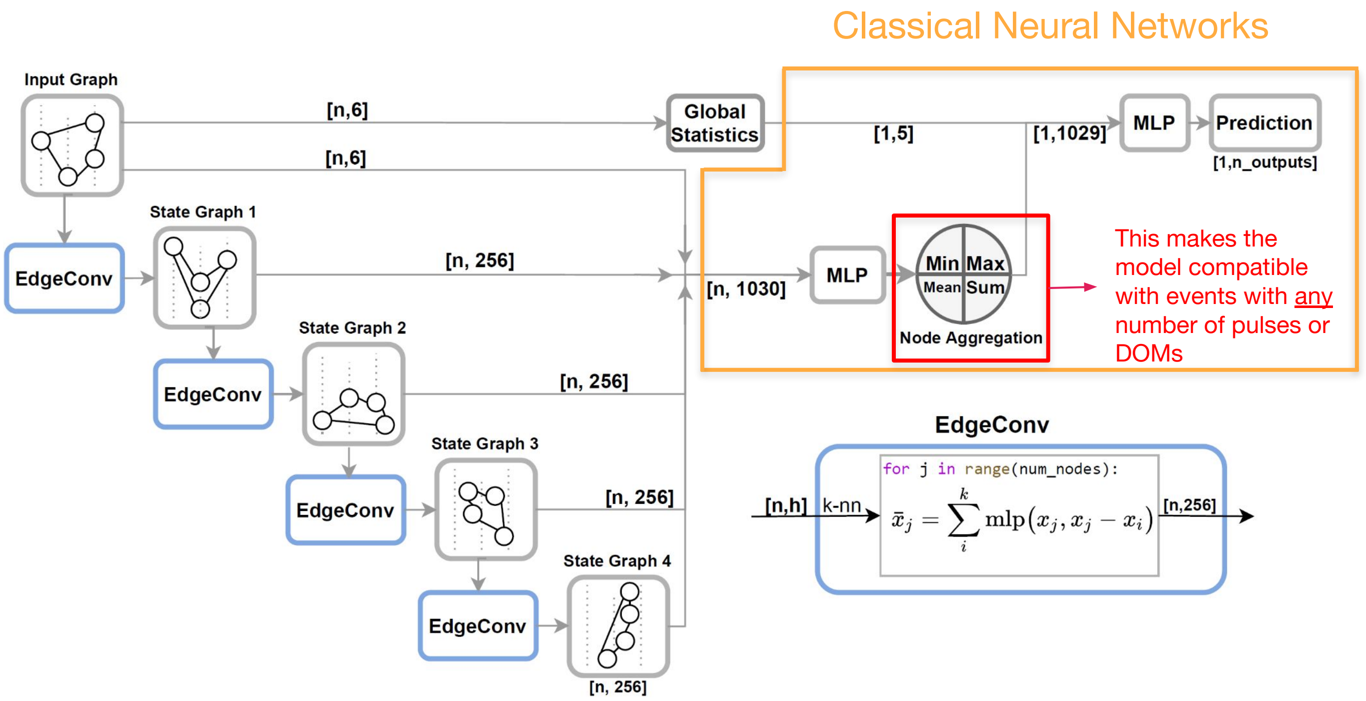
Bonus

Graph Neural Networks



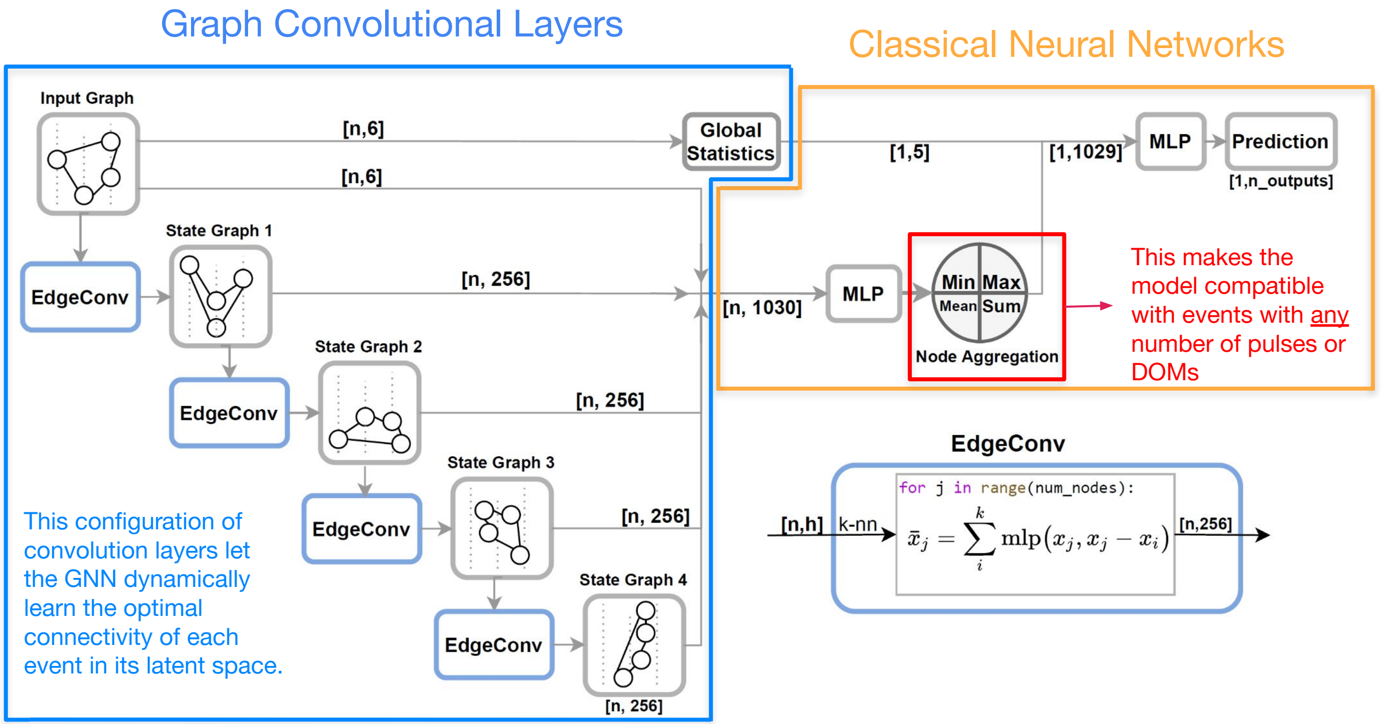
Our GNN-based reconstruction algorithm dynedge

Graph Neural Networks



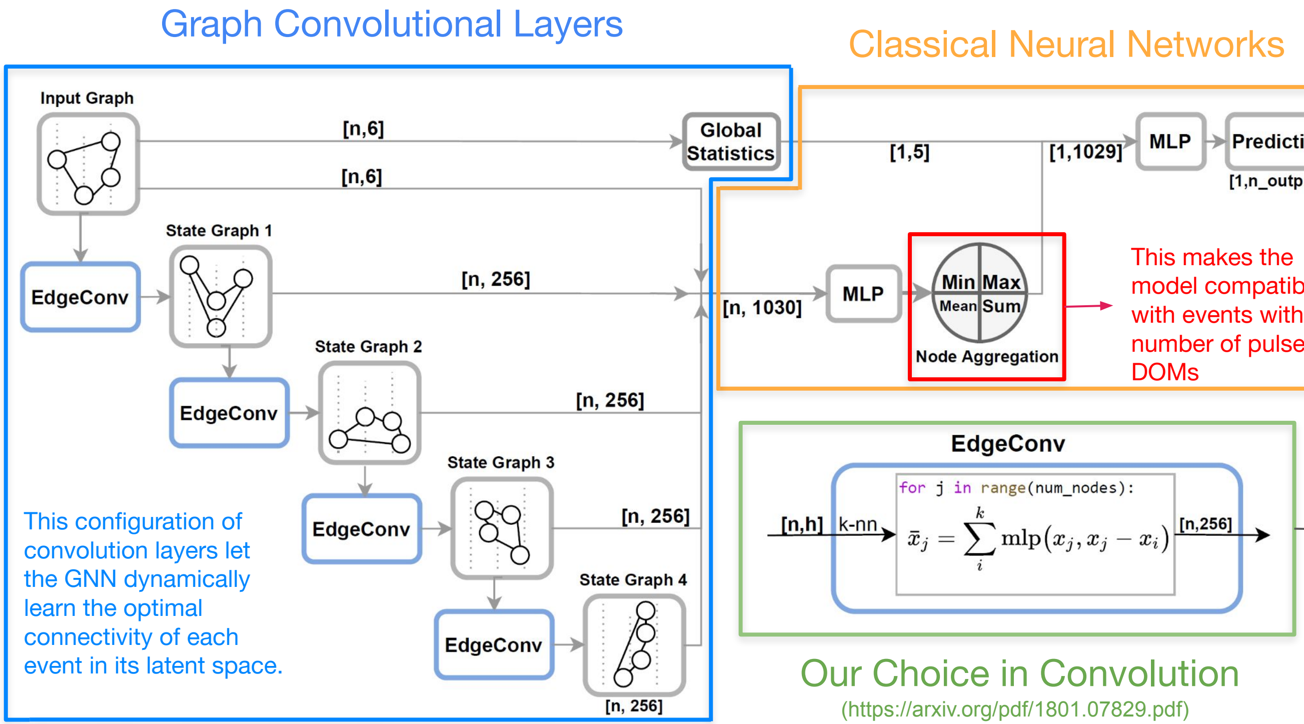
Our GNN-based reconstruction algorithm dynedge

Graph Neural Networks



Our GNN-based reconstruction algorithm dynedge

Graph Neural Networks



This configuration of convolution layers let the GNN dynamically learn the optimal connectivity of each event in its latent space.

This makes the model compatible with events with <u>any</u> number of pulses or DOMs

Captures globally relevant features in local areas by considering the difference of a node and the neighbours it's connected to. Only the neighbouring nodes contributes to the convolution!

Our Choice in Convolution
(https://arxiv.org/pdf/1801.07829.pdf)

Our GNN-based reconstruction algorithm dynedge