Graph Neural Networks Reconstruction for the Hyper Kamiokande Detector

Neutrino Physics and Machine Learning 2025

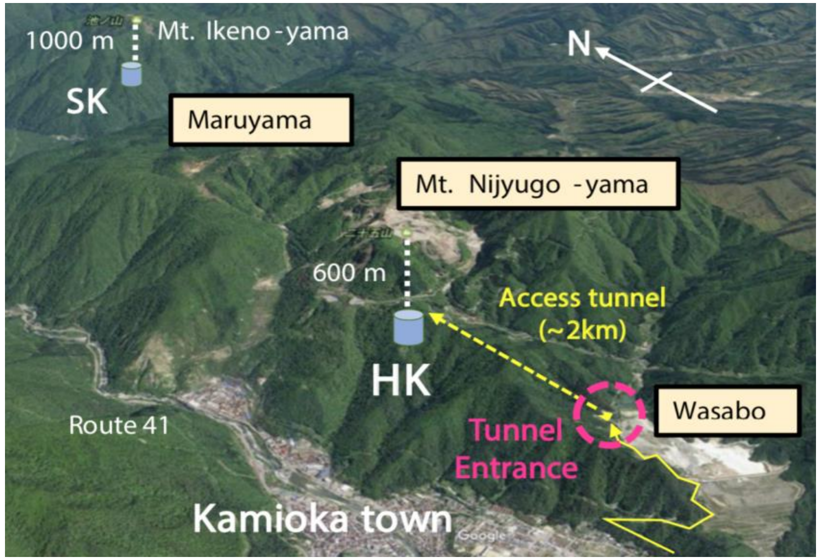
Erwan Le Blévec

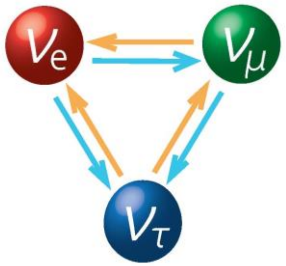
Laboratoire Leprince-Ringuet

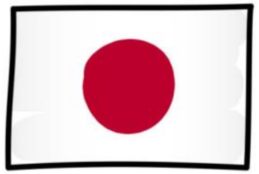
東京大学 THE UNIVERSITY OF TOKYO

ILANCE

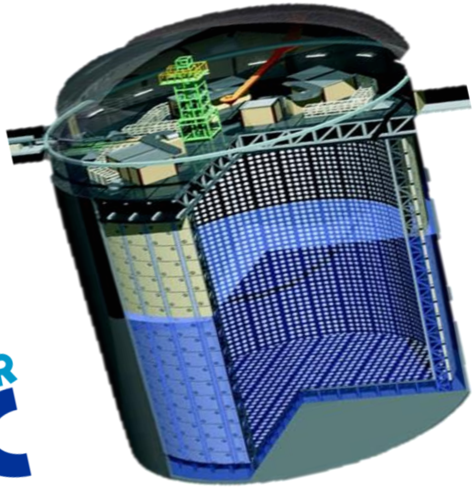
cnrs

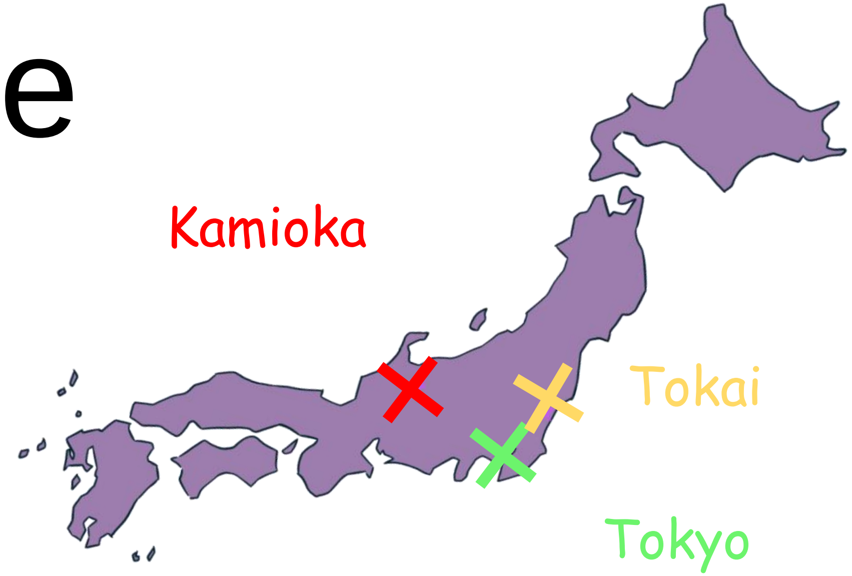






Kamiokande Detectors

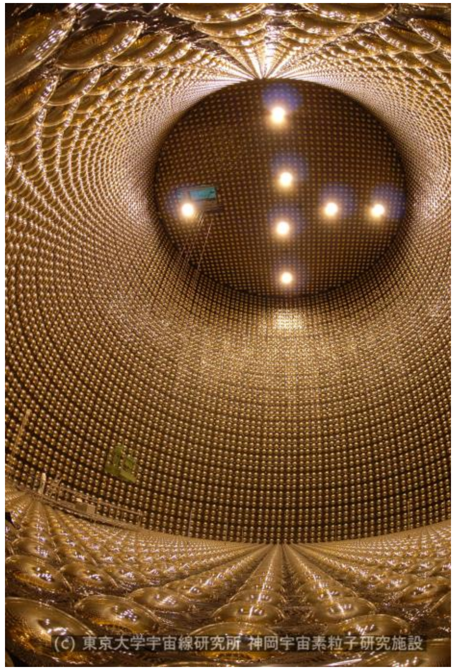




From Big to Huge

Super Kamiokande

40 m



40 m

Fid. Mass

~8 x

68 m

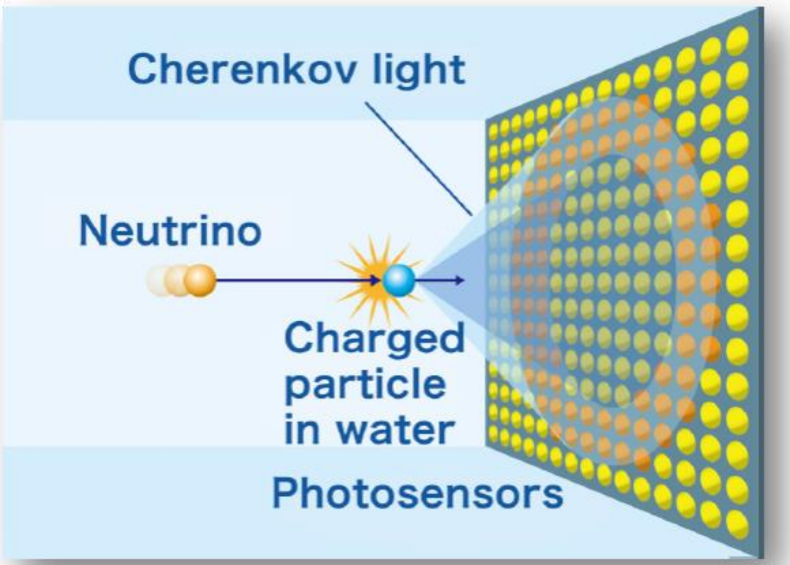
71 m

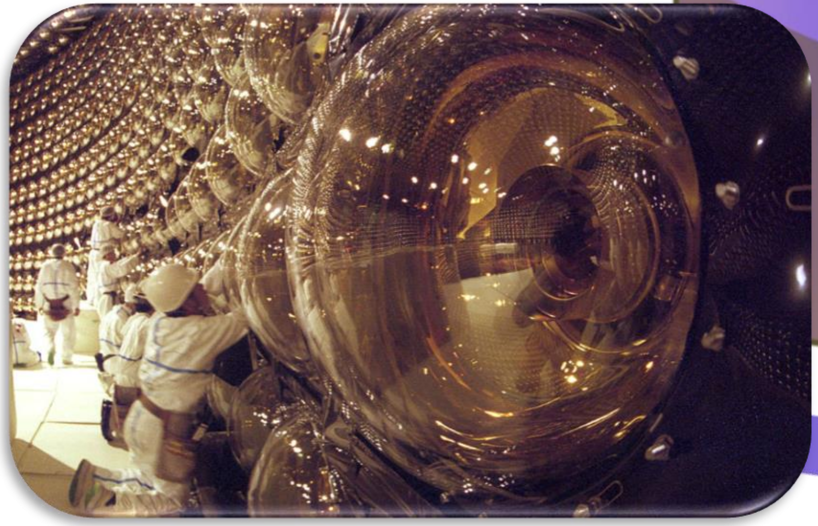
Hyper Kamiokande

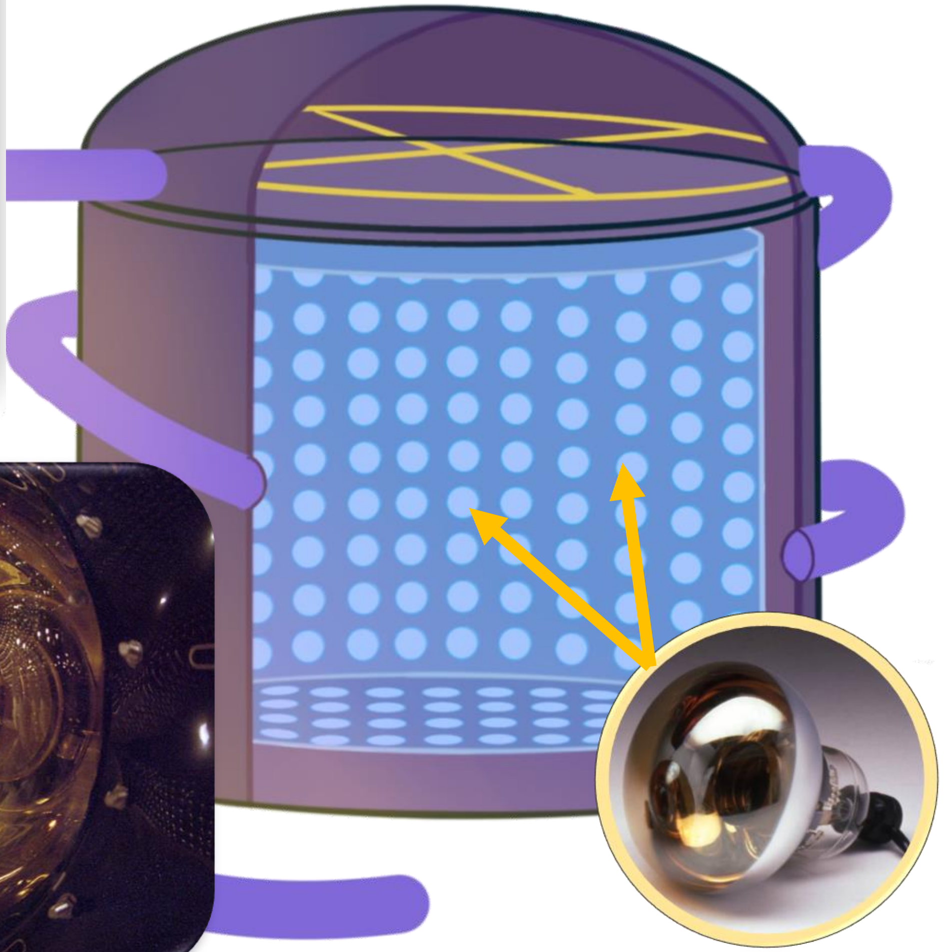
	1996 - today (and beyond)	2028 - (and beyond)
ID PMTs	**~11k**	**~20k**
Inner Volume / Fiducial Mass	**32 kton / 22.5 kton**	**217 kton / 188 kton**
Photo Coverage	**40%**	**20% (x2 efficiency)**

Detection Principle

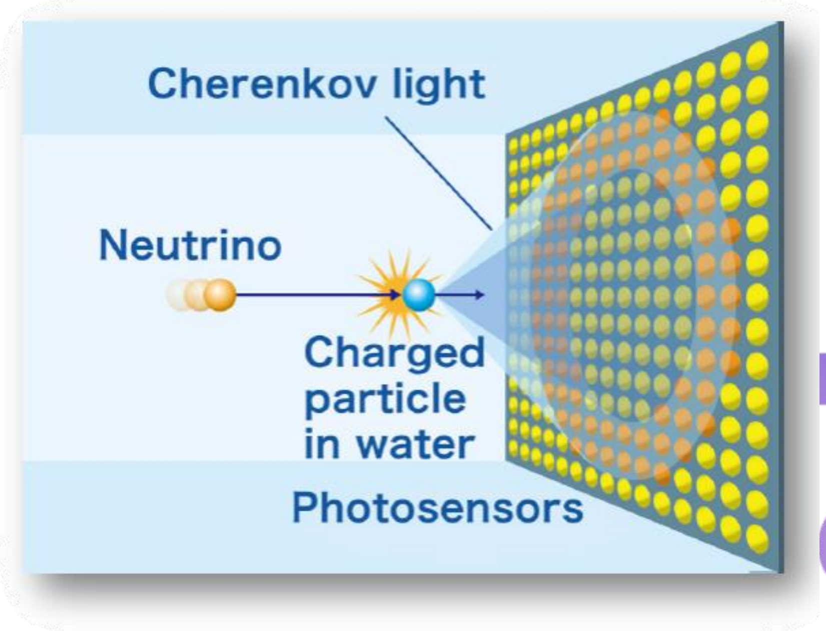
Detection Principle

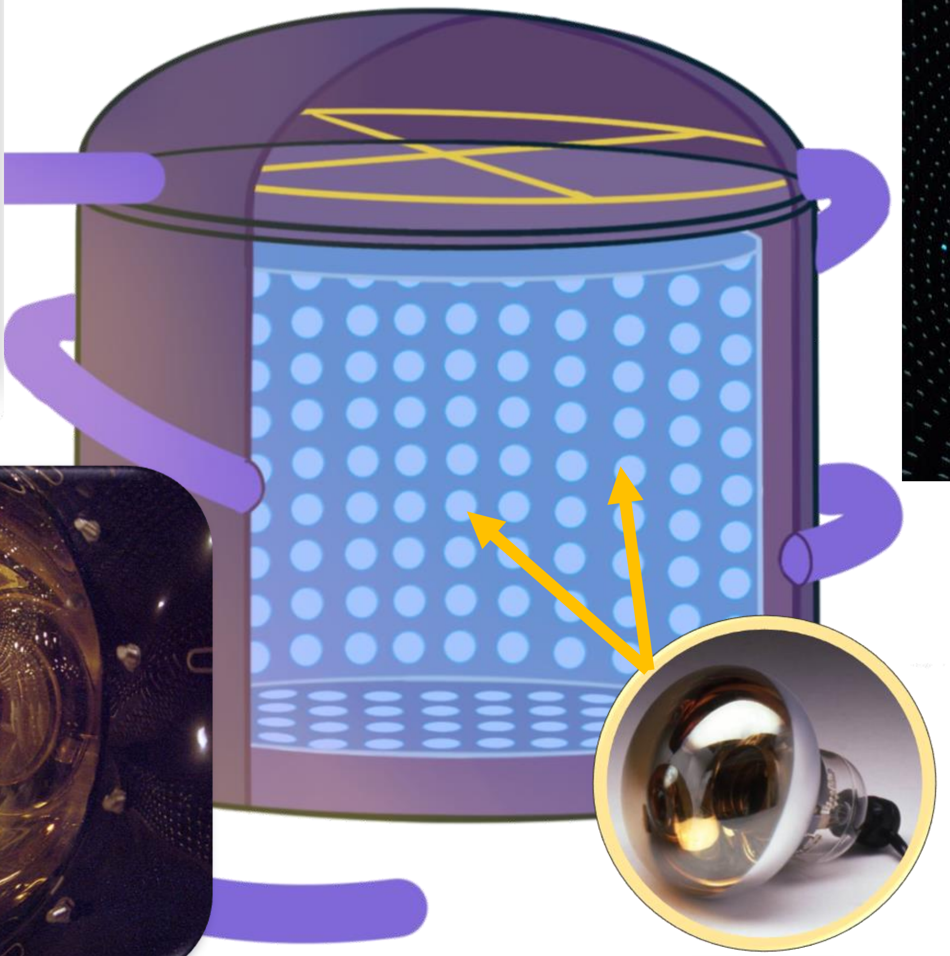


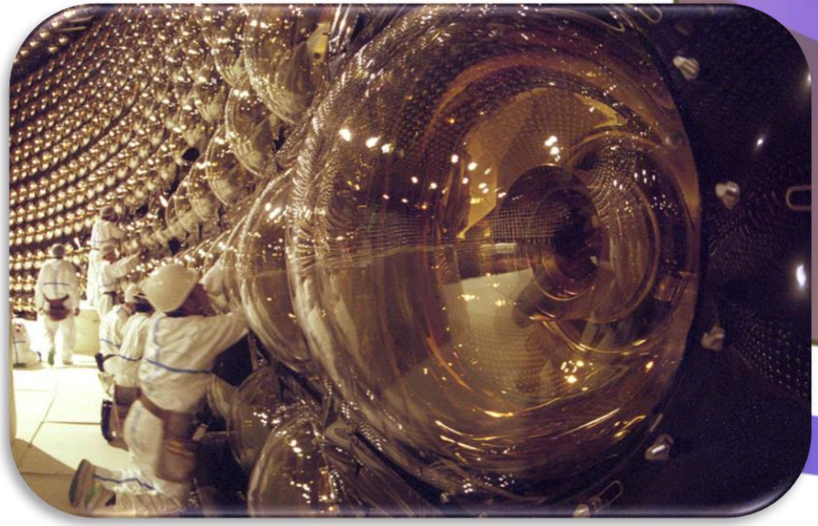


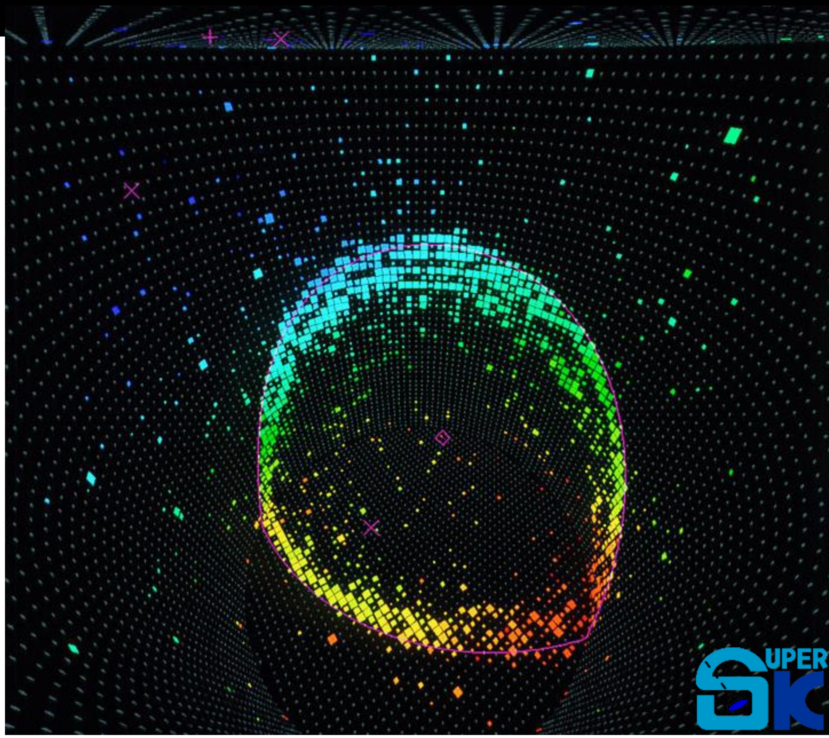


Detection Principle

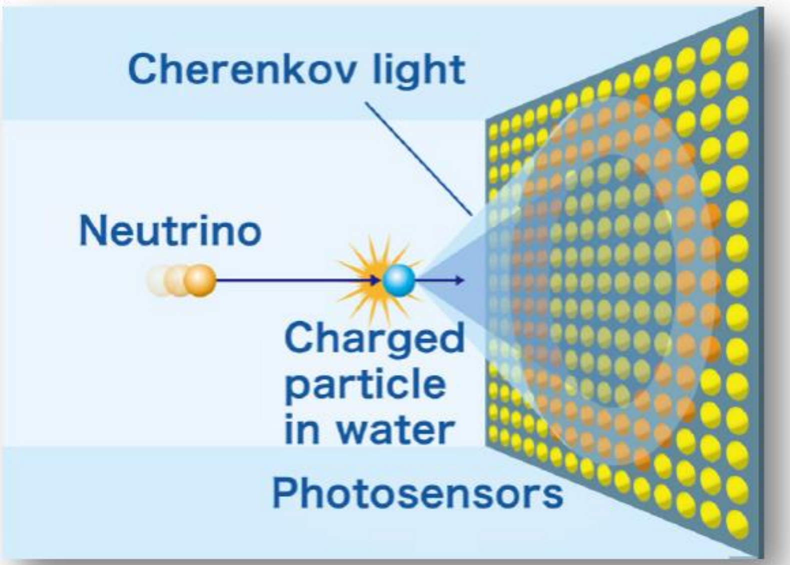


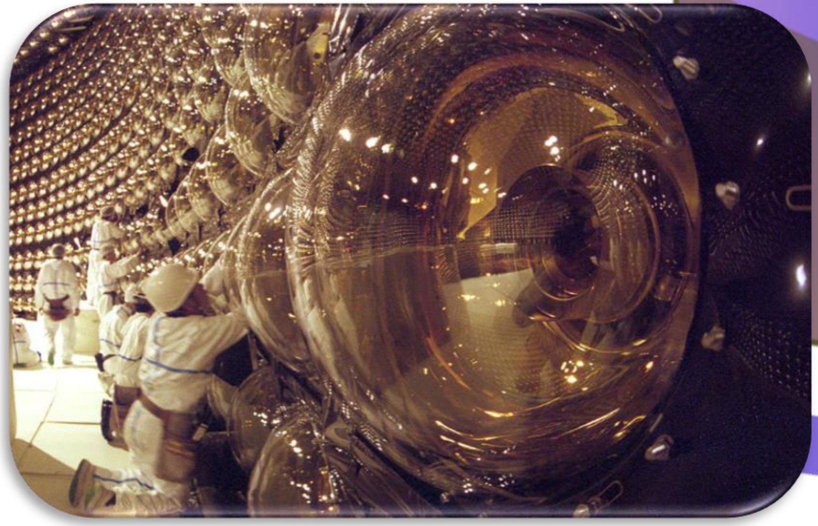


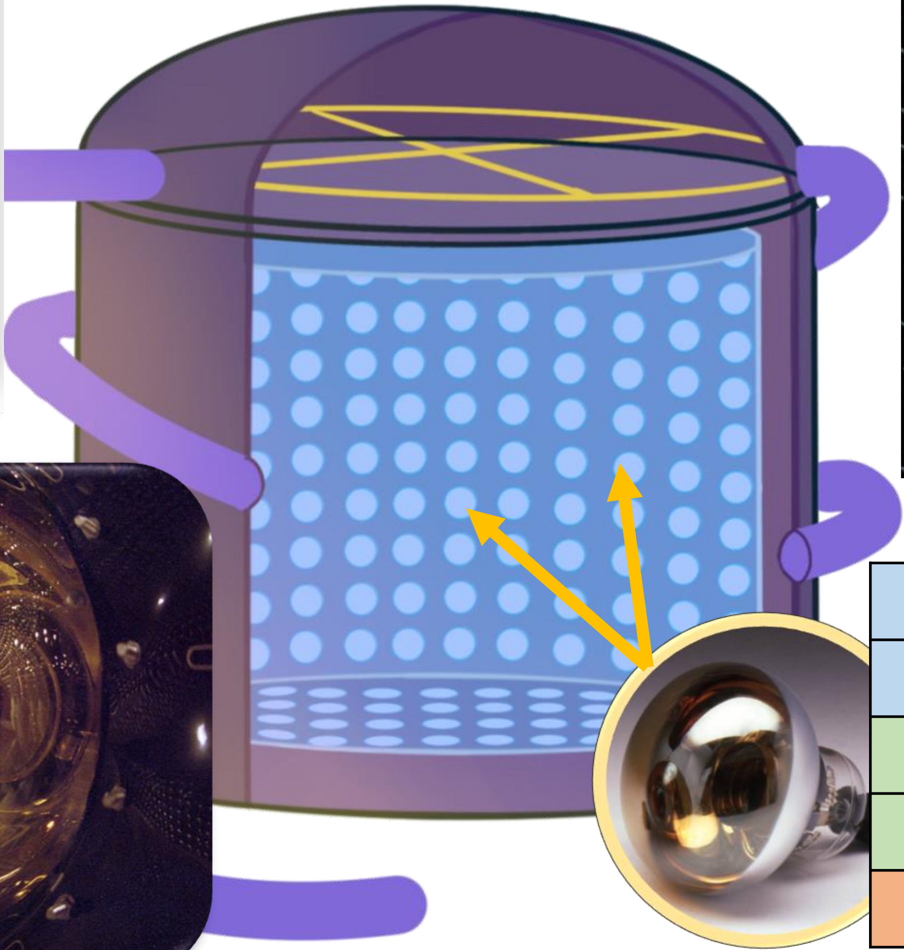


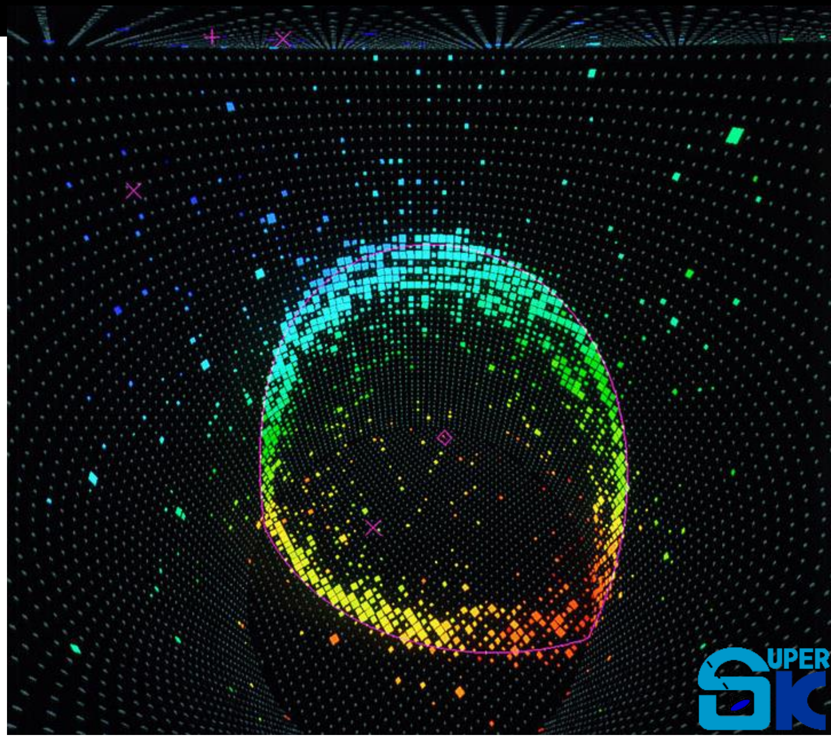


Detection Principle









One PMT Features

« Geometry »

- Position
- Orientation

« Event »

- Charge
- Time

Reconstruction Algorithms & Their Evolutions

1983 - 1998

Kamiokande

(Low trigger rate / day)

Fit by hand

Fiducial M.

30 x

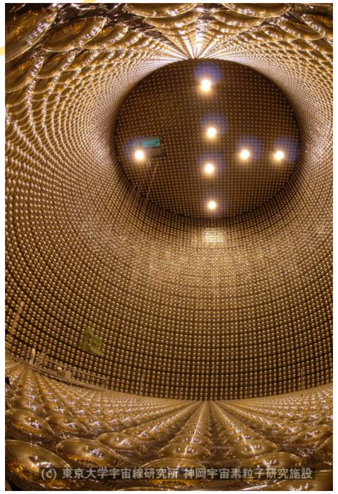
1996 - ~

Super Kamiokande

Maximum Likelihood - based Fit

One fit per « candidate » ring

APFit *: 1997 ~ now*
fiTQun *: 2012 ~ now*



8 x

2028 - ~

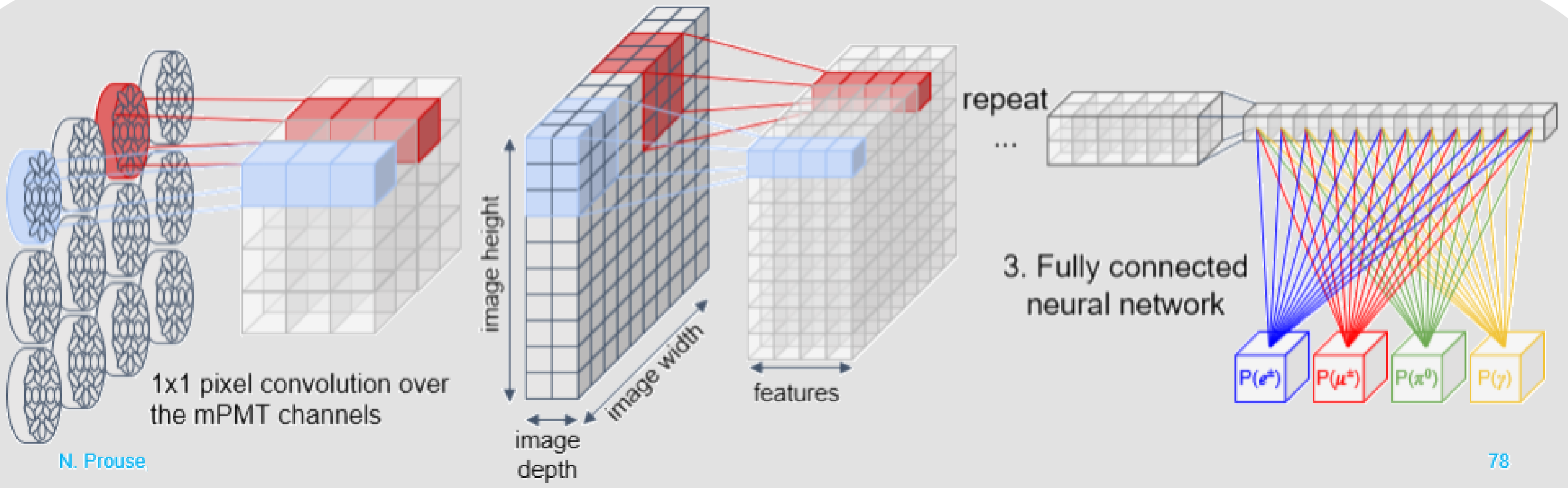
Hyper Kamiokande

Maximum L. ?
Deep Neural Networks ?
Other .. ?

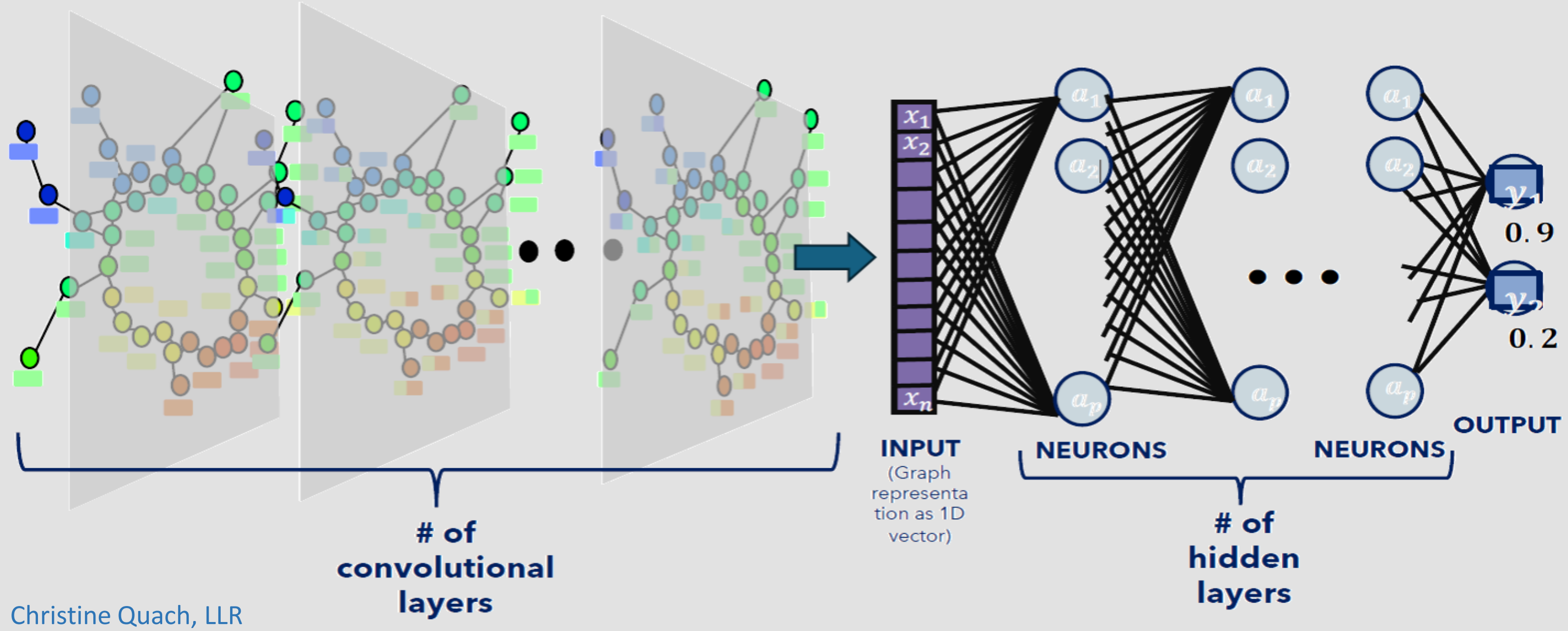
Designing Neural Networks

Deep Neural Networks For Reco. In HK : Main Architectures

Convolutional
Neural Networks
(CNNs)

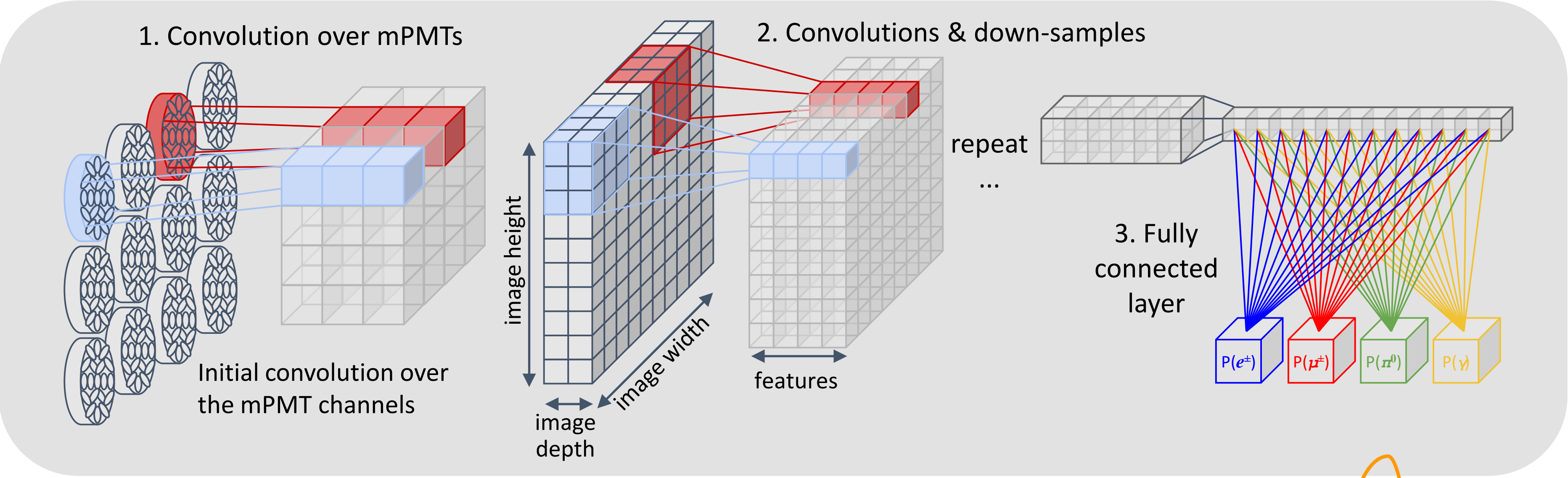


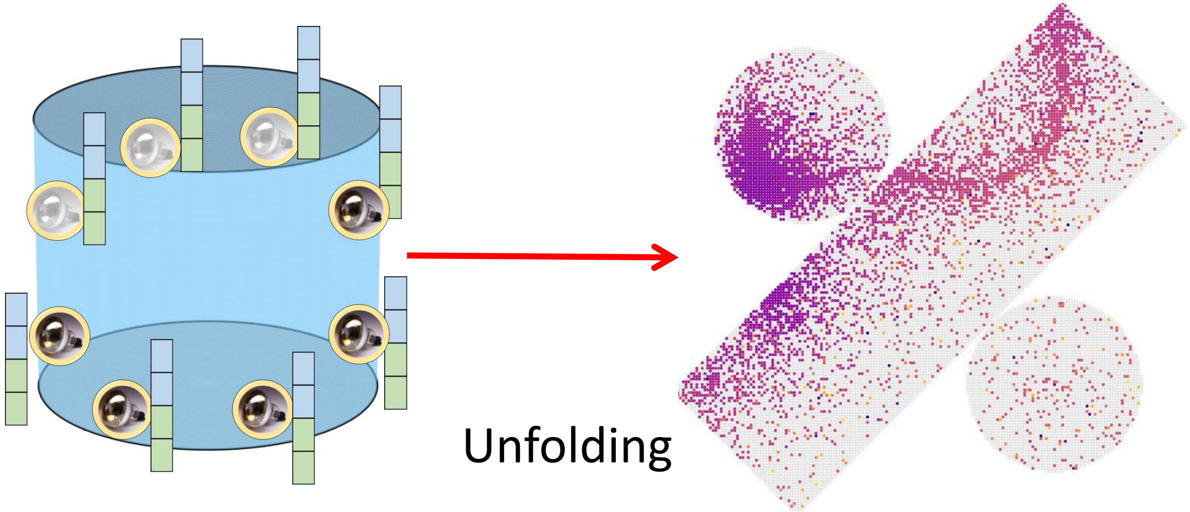
Graph
Neural Networks
(GNNs)

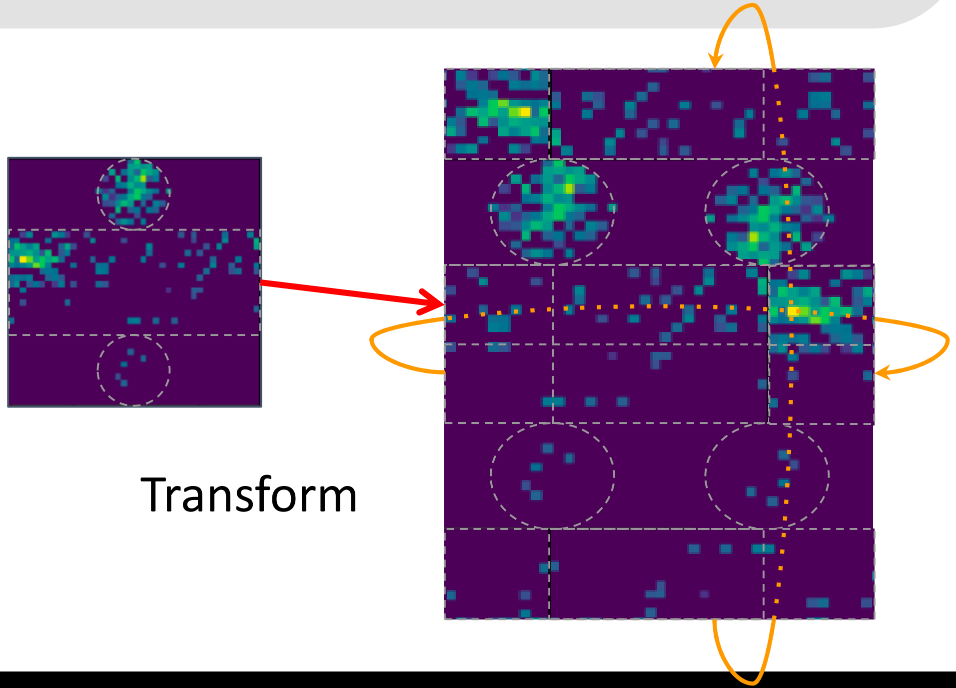


Convolutional Neural Networks for Hyper-K

Convolutional
Neural Networks
(CNNs)





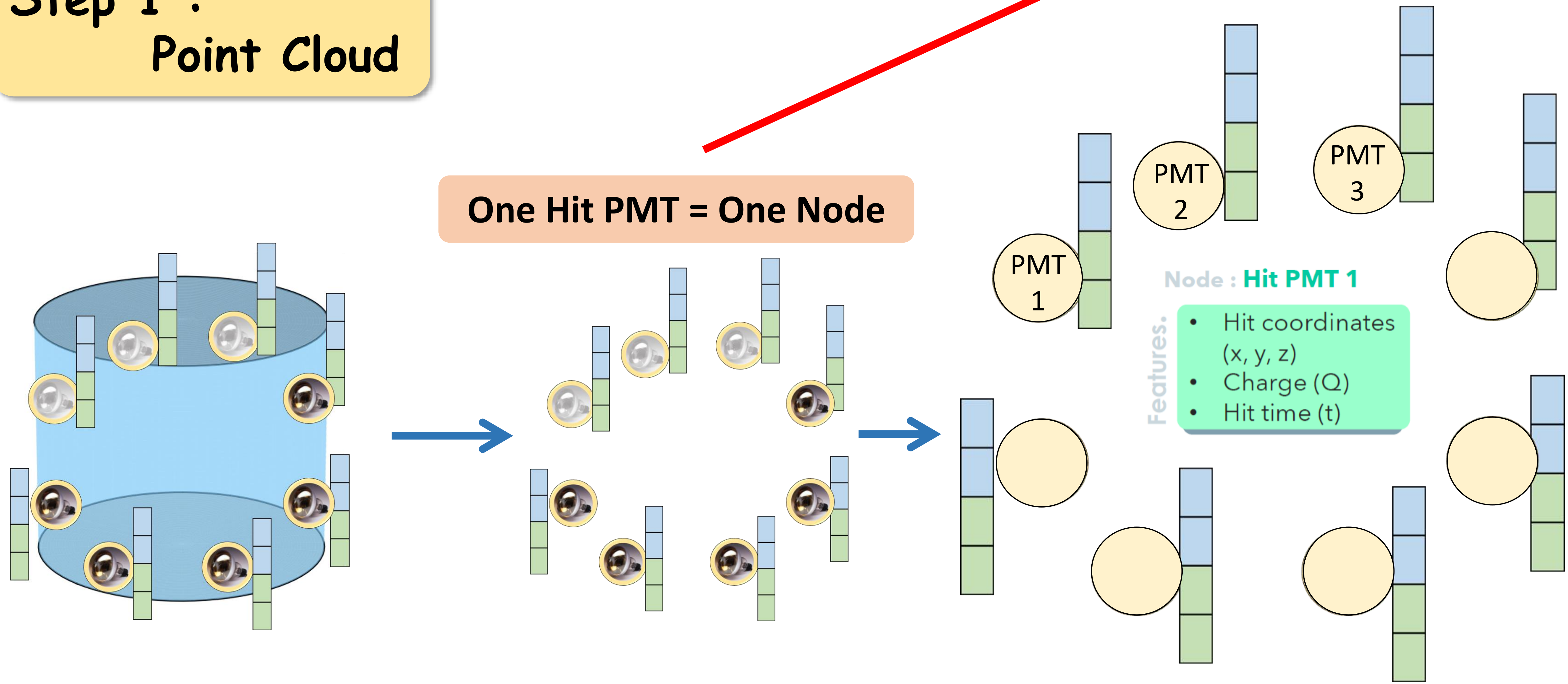


Recipe to Make a Graph

Step 1 :
Point Cloud

Point Cloud for free !

One Hit PMT = One Node



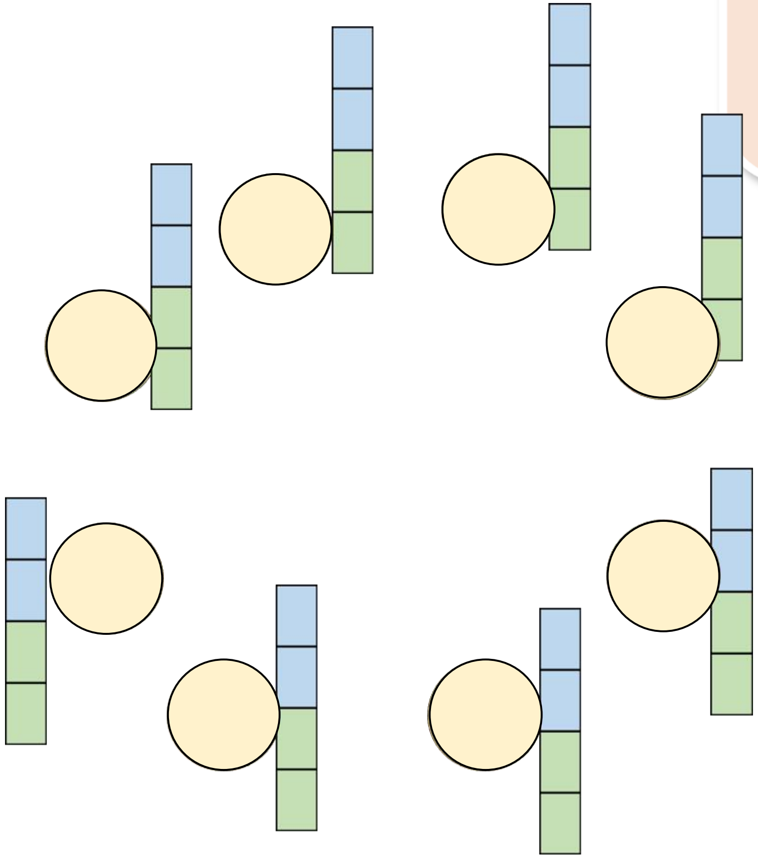
Recipe to Make a Graph

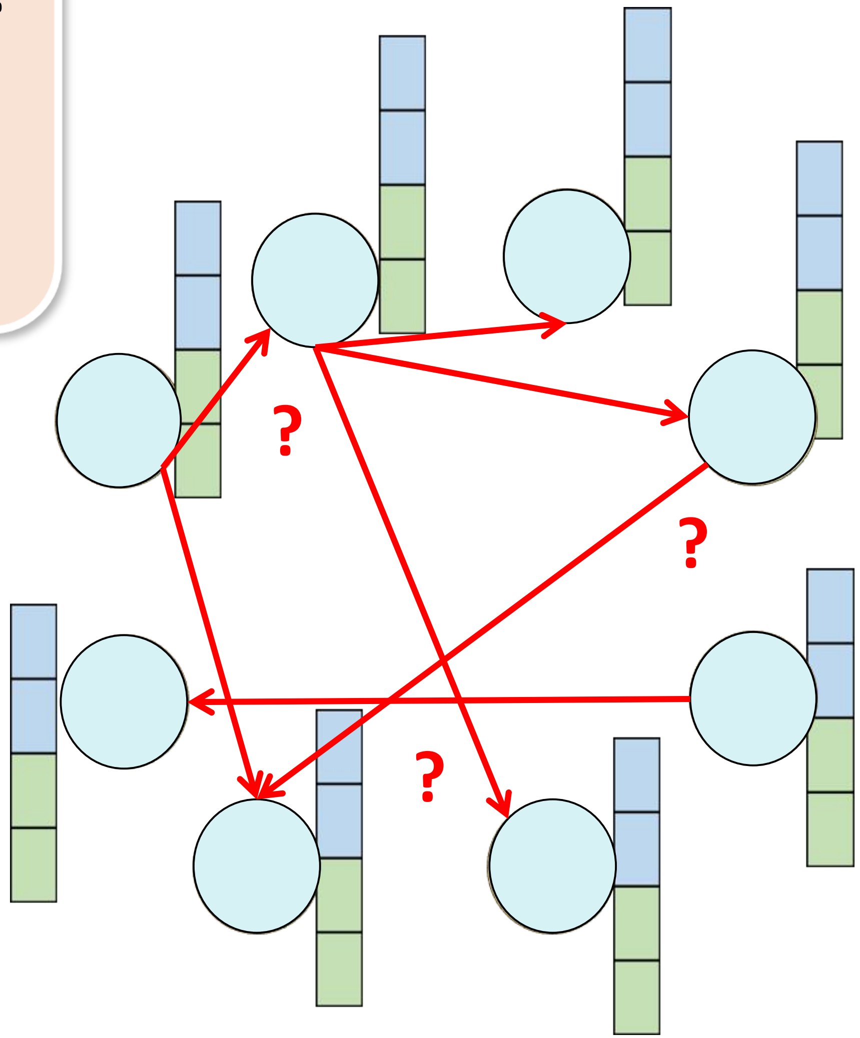
Step 2 :
Edges

Metric Space for edges :

<u>High Freedom</u>

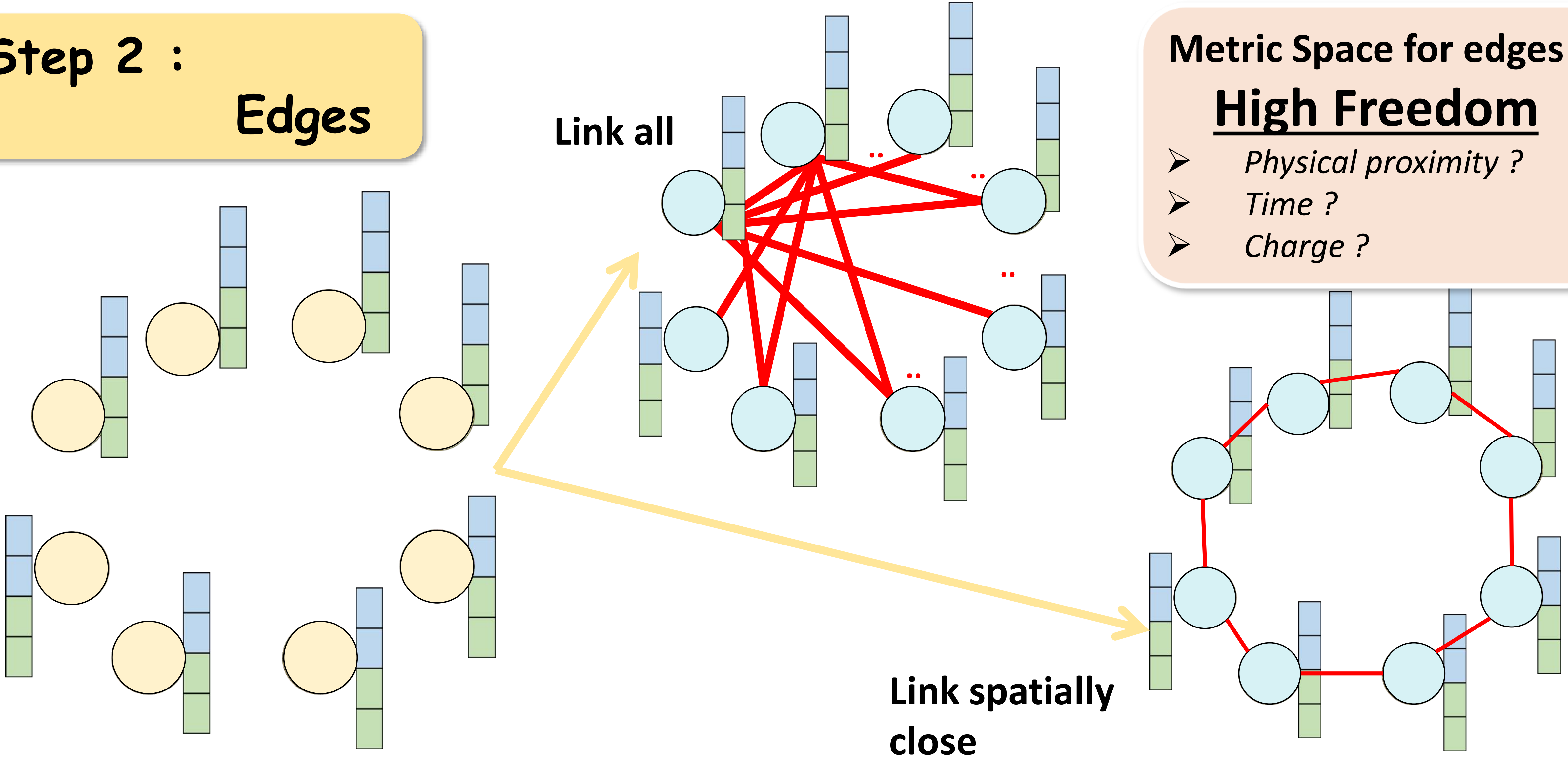
- *Physical proximity ?*
- *.. ?*





Step 2 :
Edges

Metric Space for edges :
High Freedom
- *Physical proximity ?*
- *Time ?*
- *Charge ?*



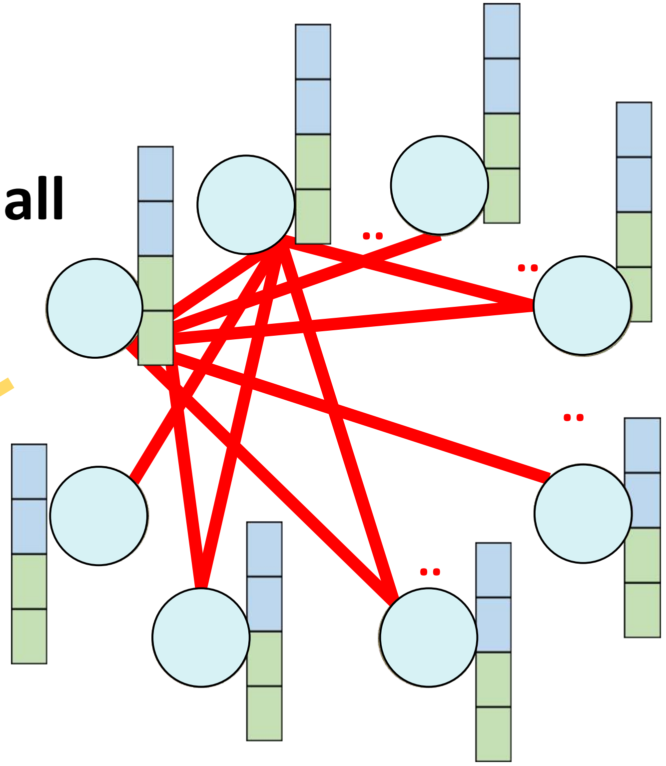
Recipe to Make a Graph

Tackle many challenges

Rings segmentation, Noise filtering
→ **Node-level tasks**

Rings counting, kinematics reconstruction
→ **Graph-level tasks**

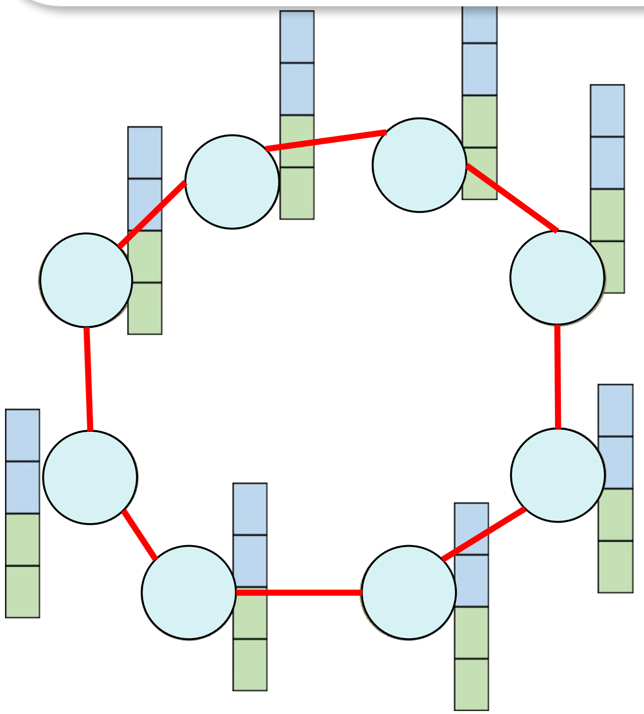
Link all



Metric Space for edges :

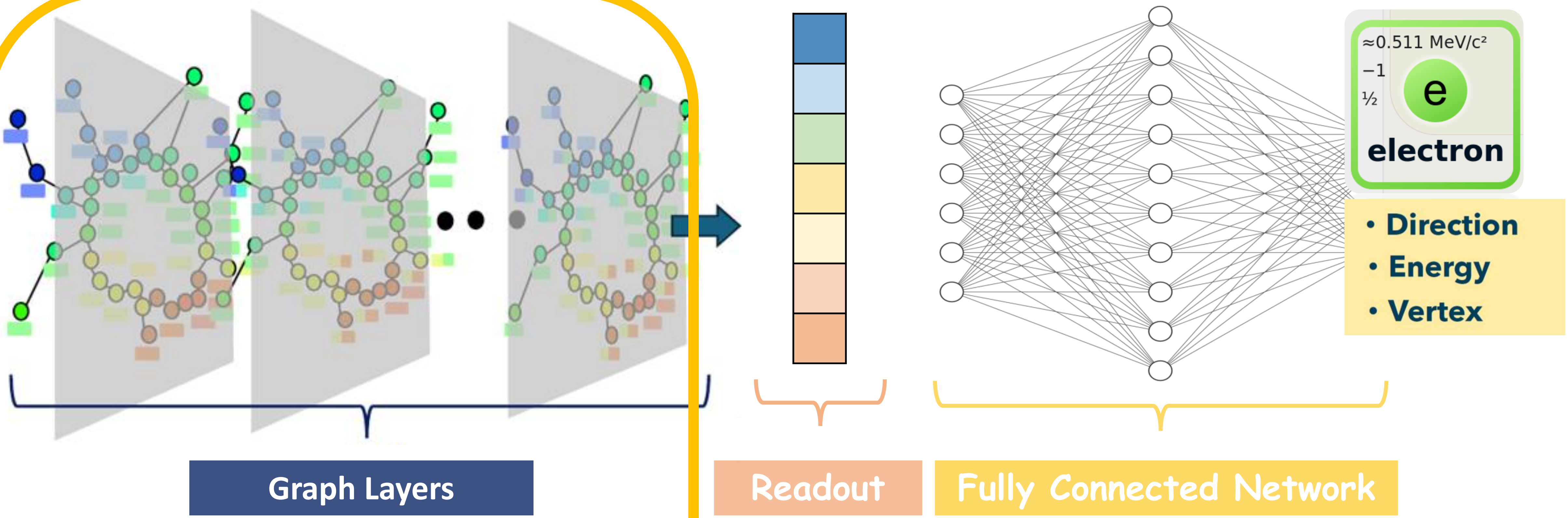
High Freedom

- *Physical proximity ?*
- *Time ?*
- *Charge ?*



Link spatially close

Graph Neural Networks for Hyper-K



Process data using graph convolution

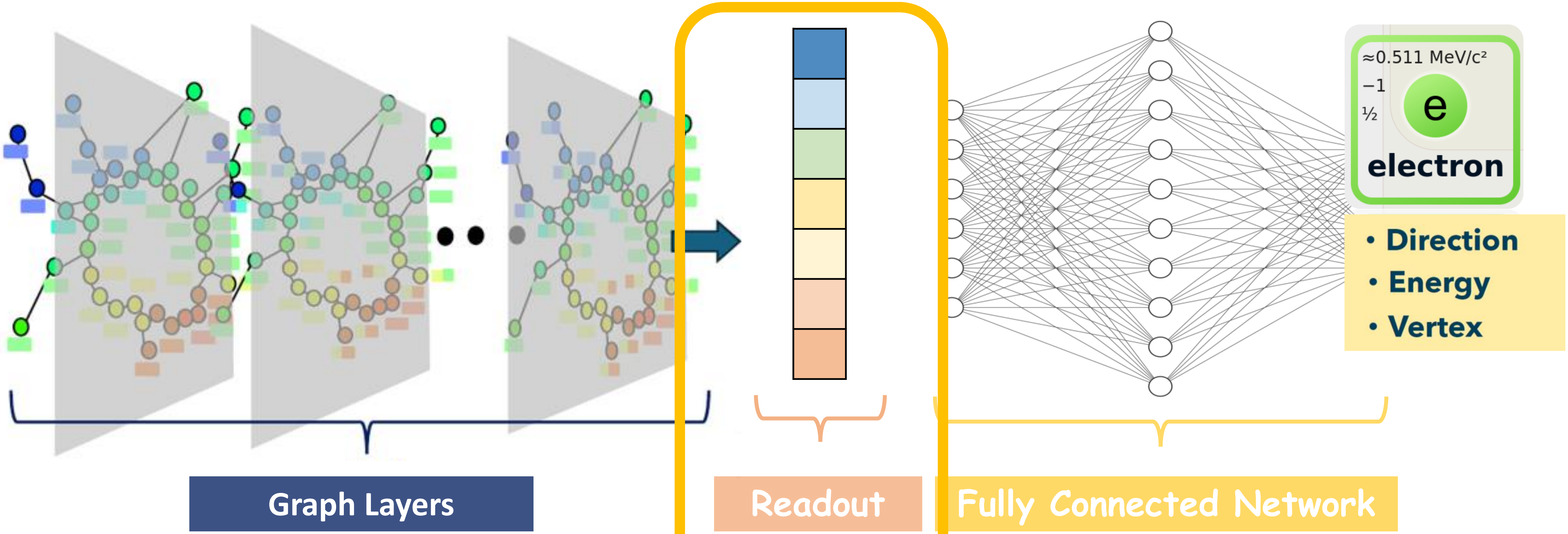
$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \begin{cases} \mathcal{V} & : \textit{Nodes} \\ \mathcal{E} & : \textit{Edges} \end{cases}$$

$$\downarrow$$

$$\mathbb{R}^n$$

Make the prediction from the readout

Graph Neural Networks for Hyper-K

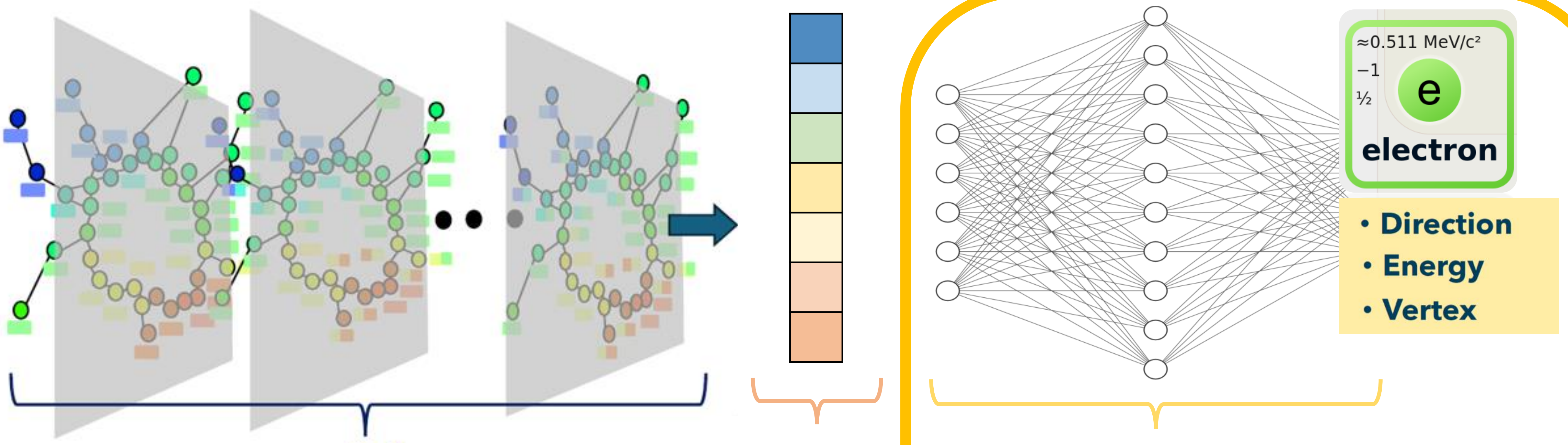


Process data using graph convolution

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \begin{cases} \mathcal{V} & : \textit{Nodes} \\ \mathcal{E} & : \textit{Edges} \end{cases} \longrightarrow \mathbb{R}^n$$

Make the prediction from the readout

Graph Neural Networks for Hyper-K



Graph Layers | **Readout** | **Fully Connected Network**

Process data using graph convolution

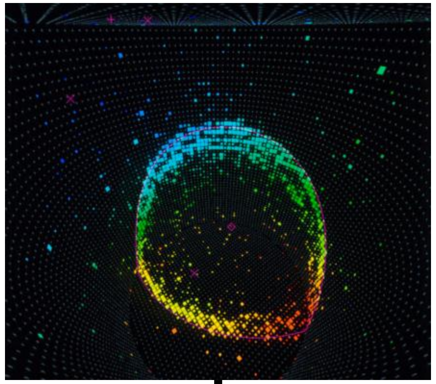
$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \begin{cases} \mathcal{V} & : \textit{Nodes} \\ \mathcal{E} & : \textit{Edges} \end{cases}$$

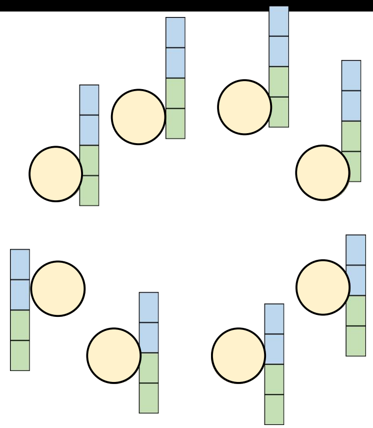
$$\downarrow$$

$$\mathbb{R}^n$$

Make the prediction from the readout

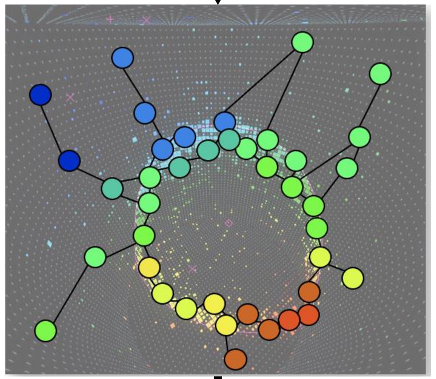
Training GNNs for Hyper K : Pipeline

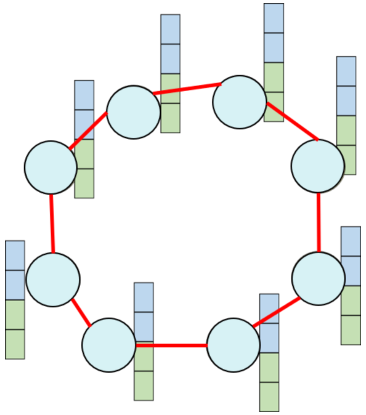




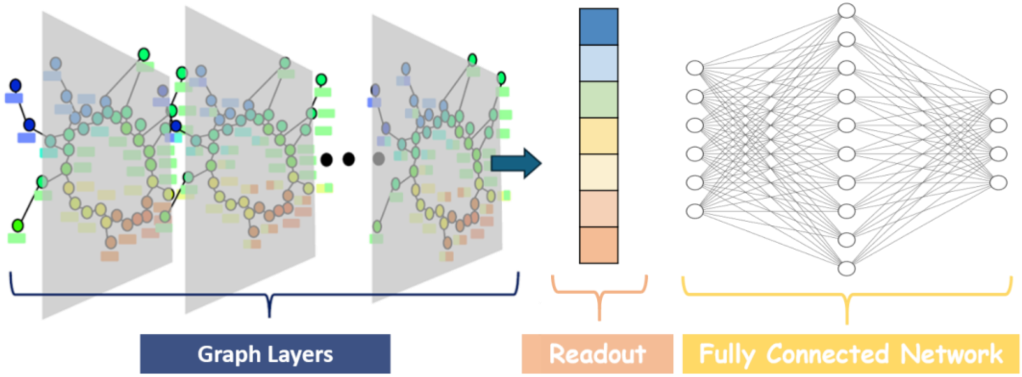
[1] : Point Cloud

- *Downstream task*
- *Supervised setting*

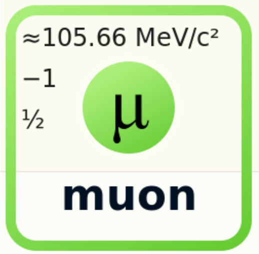




[2] : Graph



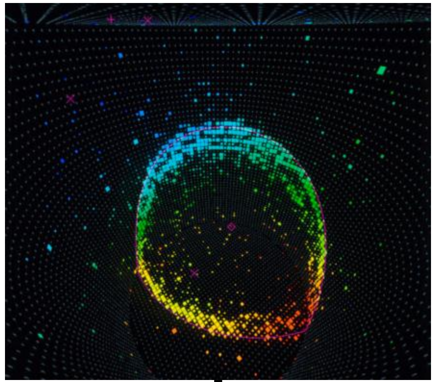
[3] : GNN

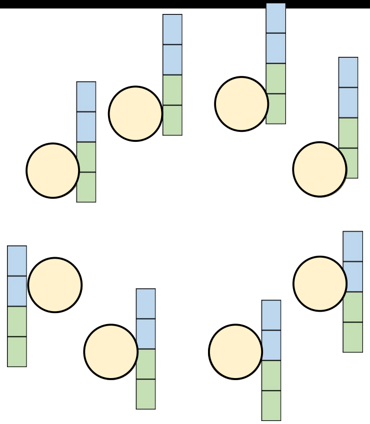


- **Direction**
- **Energy**
- **Vertex**

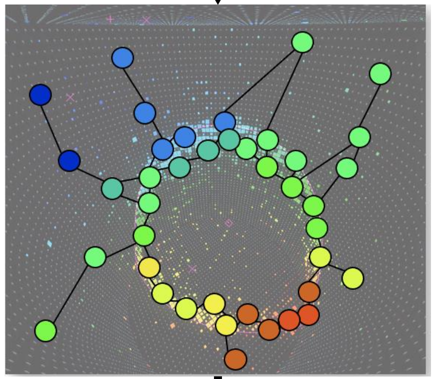
[4] : Reconstructed data

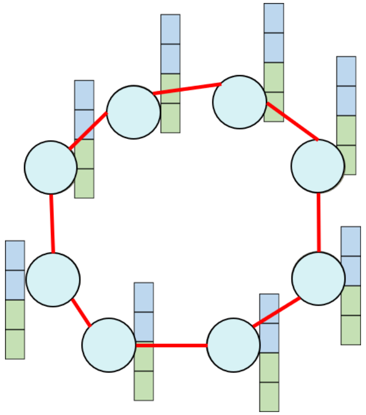
Training GNNs for Hyper K : Pipeline



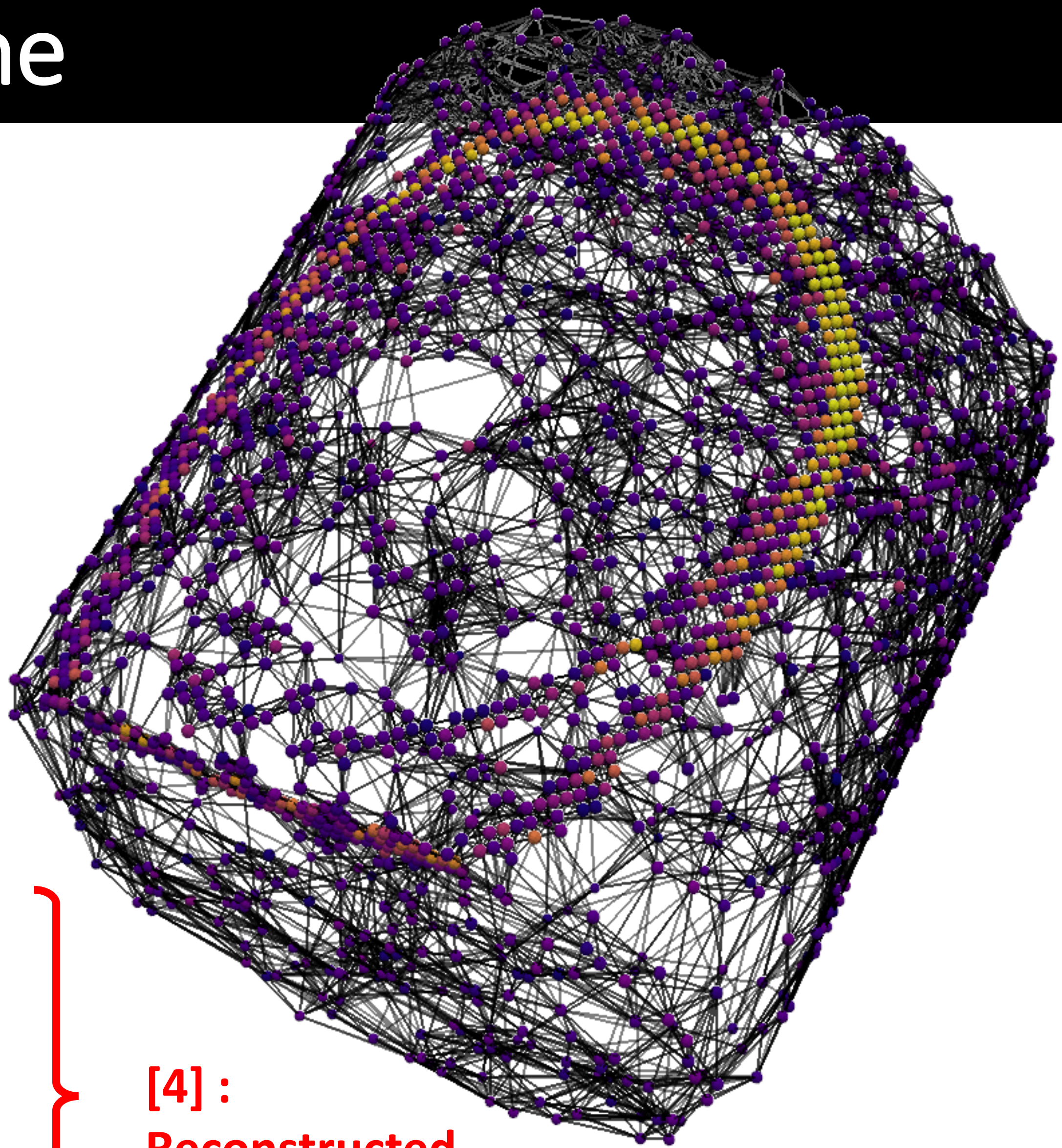


[1] : Point Cloud

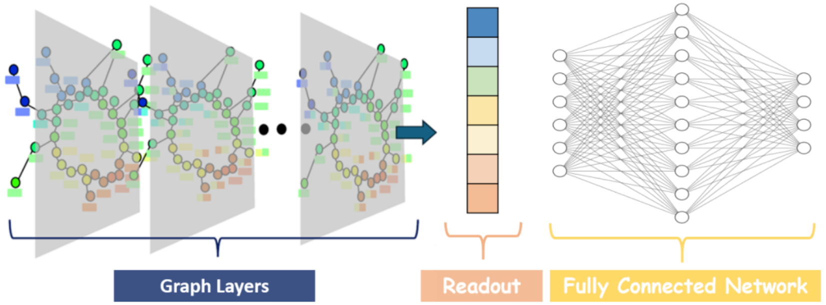


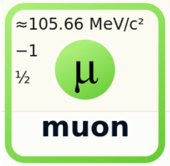


[2] : Graph



[3] : GNN





- Direction
- Energy
- Vertex

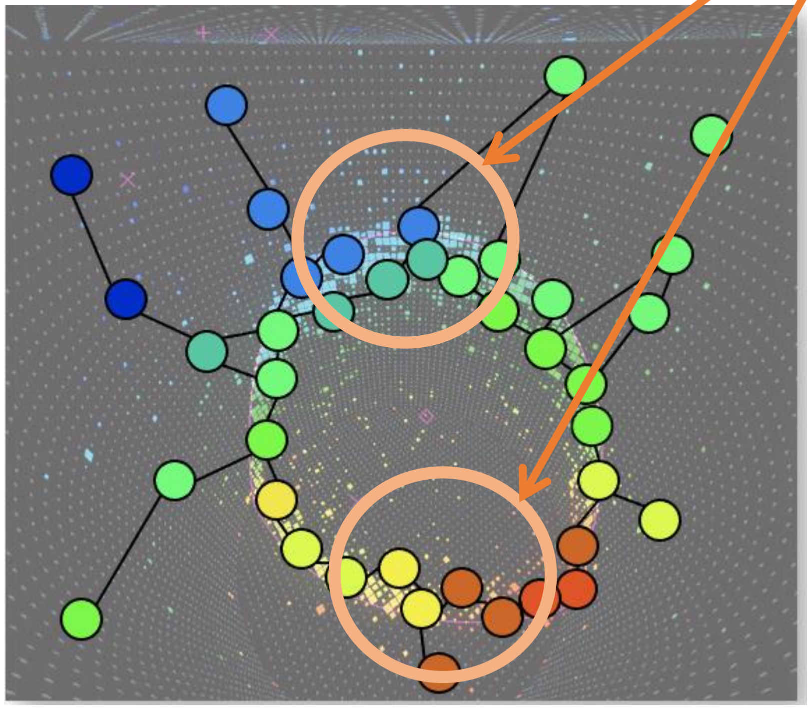
[4] : Reconstructed data

Challenges for Graph-level Tasks

What Could Go Wrong ?

Metric space :
spatial proximity

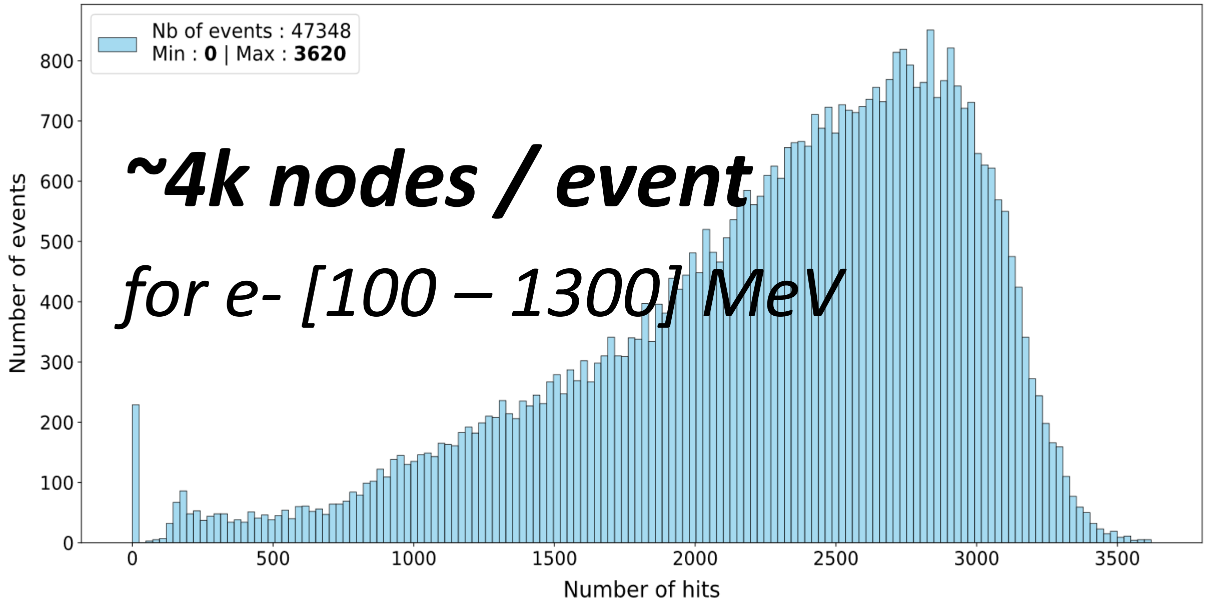
Long range dependencies



Many nodes

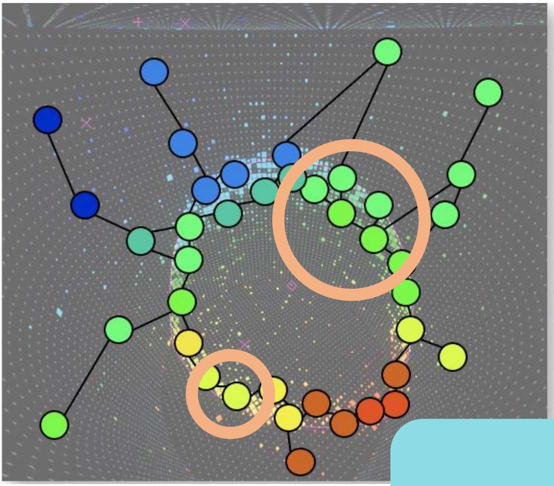
➢ Limits the **number of edges / PMT**

Plot : 500 MeV e-



What Could Go Wrong ?

Metric space :
spatial proximity



Long range dependencies

Stack multiple layers

(Harder to train)

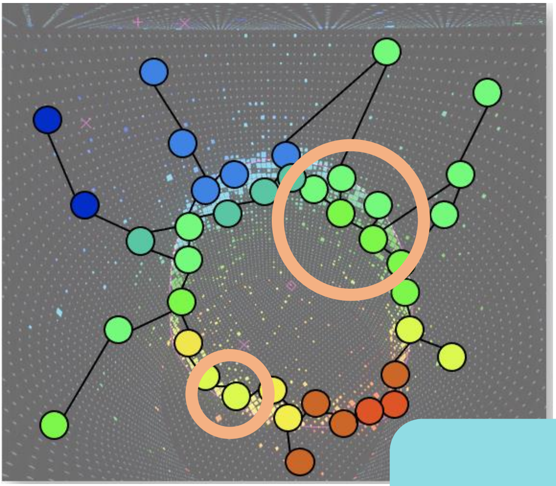
Many nodes

~4k nodes / event

for e- [100 – 1300] MeV

What Could Go Wrong ?

Metric space :
spatial proximity



Long range dependencies

Many nodes

~4k nodes / event

for e- [100 – 1300] MeV

Over squashing

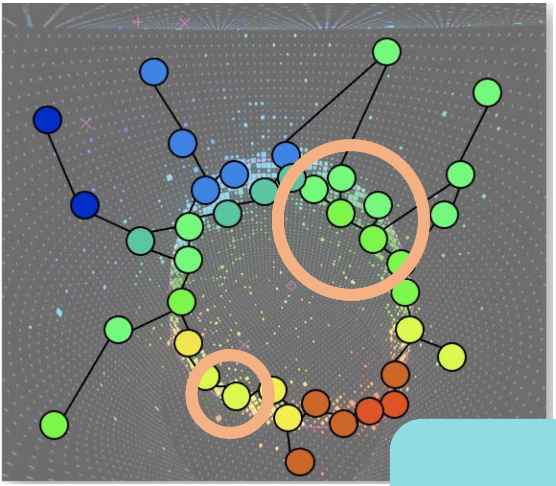
Info. not propagating

Stack multiple layers

(Harder to train)

What Could Go Wrong ?

Metric space :
spatial proximity



Long range dependencies

Many nodes

~4k nodes / event

for e- [100 – 1300] MeV

Over squashing

Info. not propagating

Stack multiple layers

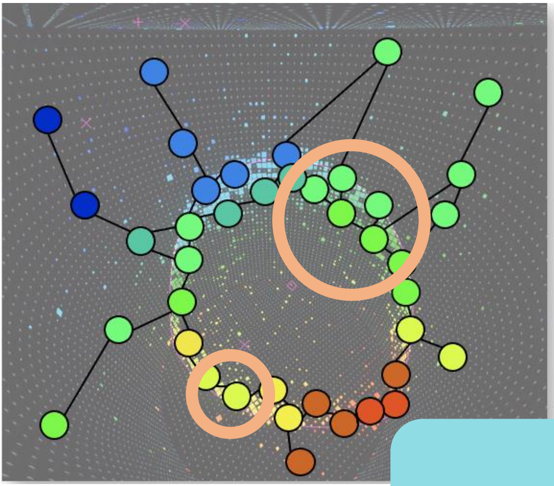
(Harder to train)

Loss of discriminative power

Over smoothing

What Could Go Wrong ?

Metric space :
spatial proximity



Long range dependencies

Many nodes

~4k nodes / event

for e- [100 – 1300] MeV

Over squashing

Info. not propagating

Increase Node Embedding

Stack multiple layers

(Harder to train)

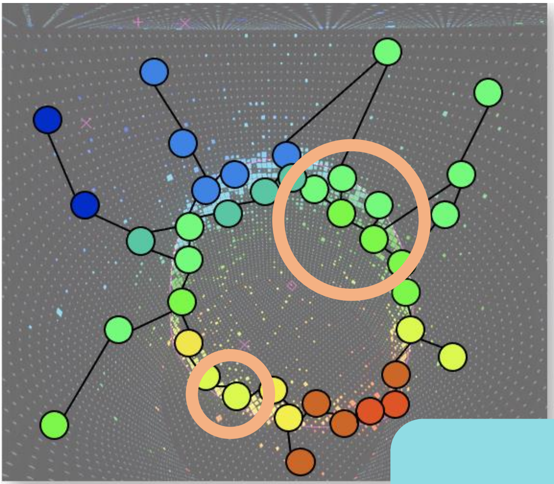
Loss of discriminative power

*Usually ~ **50-100 features / node***

Over smoothing

What Could Go Wrong ?

Metric space :
spatial proximity



Long range dependencies

Stack multiple layers

(Harder to train)

Many nodes

~4k nodes / event

for e- [100 – 1300] MeV

Graph-level task

Readout Bottleneck

4k * 50 = **O(10^5)**
To **O(10^2)**

Info. not propagating

Over squashing

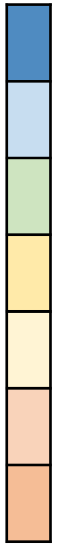
Loss of discriminative power

Over smoothing

Increase Node Embedding

Usually ***~ 50-100 features / node***

$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \rightarrow \mathbb{R}^n$



Reconstruction Tasks to Monitor Challenges

Energy

$$E^2 = m^2c^4 + c^2\underline{\left(p_x^2 + p_y^2 + p_z^2\right)}$$

Integrated over all the dimensions
→ Don't rely much on the event topology

Reconstruction Tasks to Monitor Challenges

Energy

$$E^2 = m^2c^4 + c^2\underline{\left(p_x^2 + p_y^2 + p_z^2\right)}$$

Integrated over all the dimensions
→ Don't rely much on the event topology

≈ Per-node-charge sum

~~Long range dependencies~~

Readout Bottleneck

Reconstruction Tasks to Monitor Challenges

Energy

$$E^2 = m^2c^4 + c^2\underline{\left(p_x^2 + p_y^2 + p_z^2\right)}$$

Integrated over all the dimensions
→ Don't rely much on the event topology

≈ Per-node-charge sum

Long range dependencies

Readout Bottleneck

Vertex

$$\{x,\ y,\ z\}$$

→ Highly rely on the event topology

Ring position & shape in the detector

Reconstruction Tasks to Monitor Challenges

Energy

$$E^2 = m^2c^4 + c^2\underline{\left(p_x^2 + p_y^2 + p_z^2\right)}$$

Integrated over all the dimensions
→ Don't rely much on the event topology

≈ Per-node-charge sum

~~**Long range dependencies**~~

Readout Bottleneck

Vertex

$$\{x,\ y,\ z\}$$

→ Highly rely on the event topology

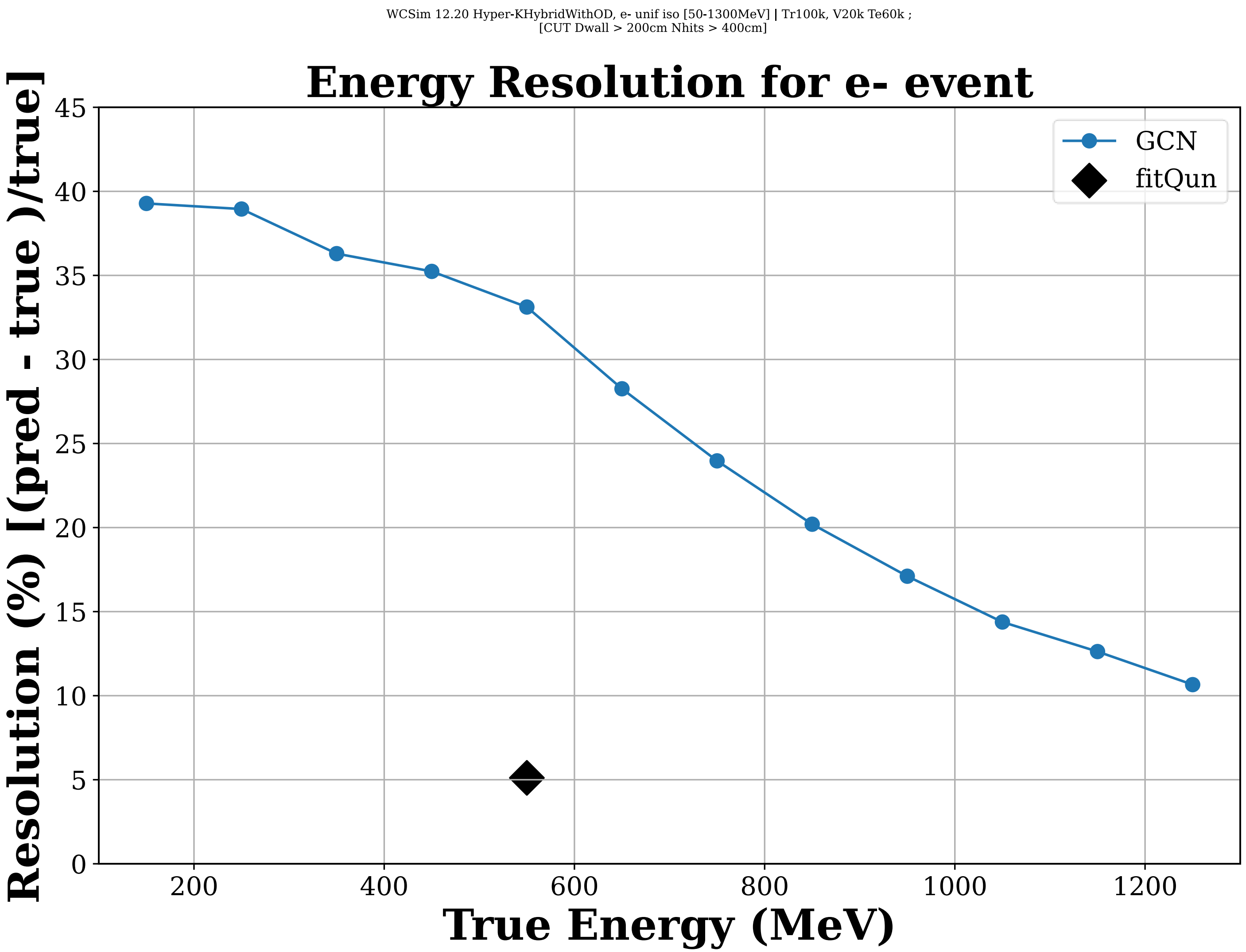
Ring position & shape in the detector

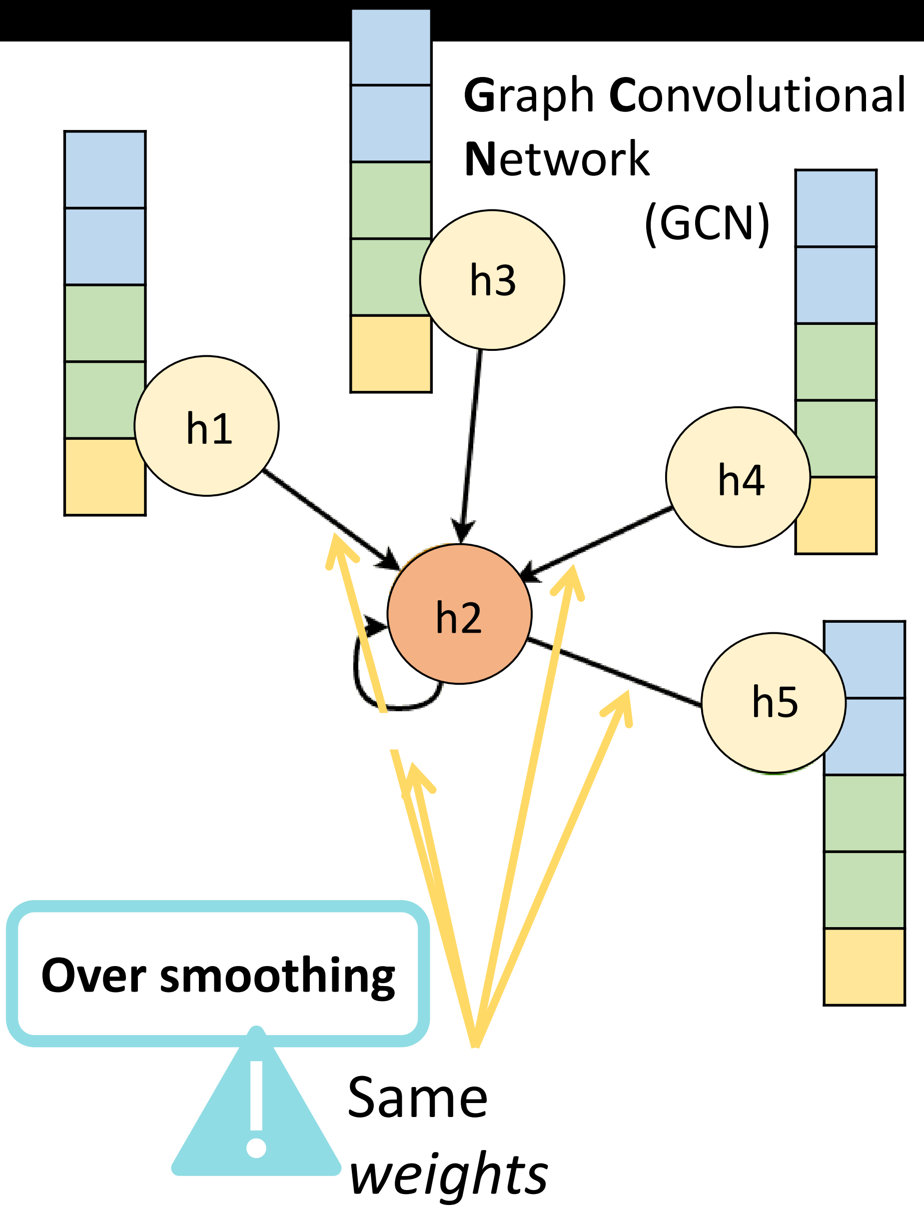
Long range dependencies

Readout Bottleneck

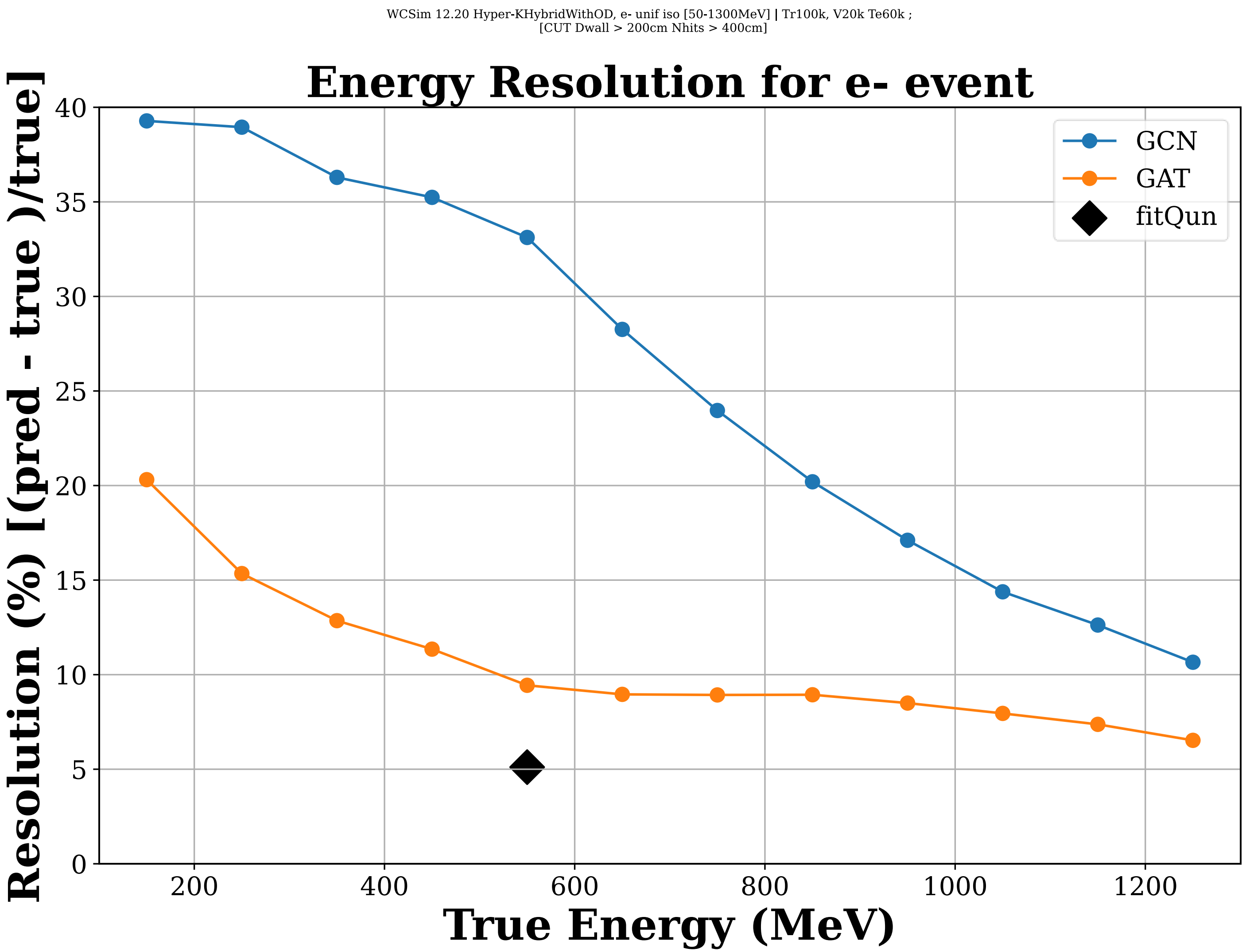
Reconstruction status

Energy Reconstruction

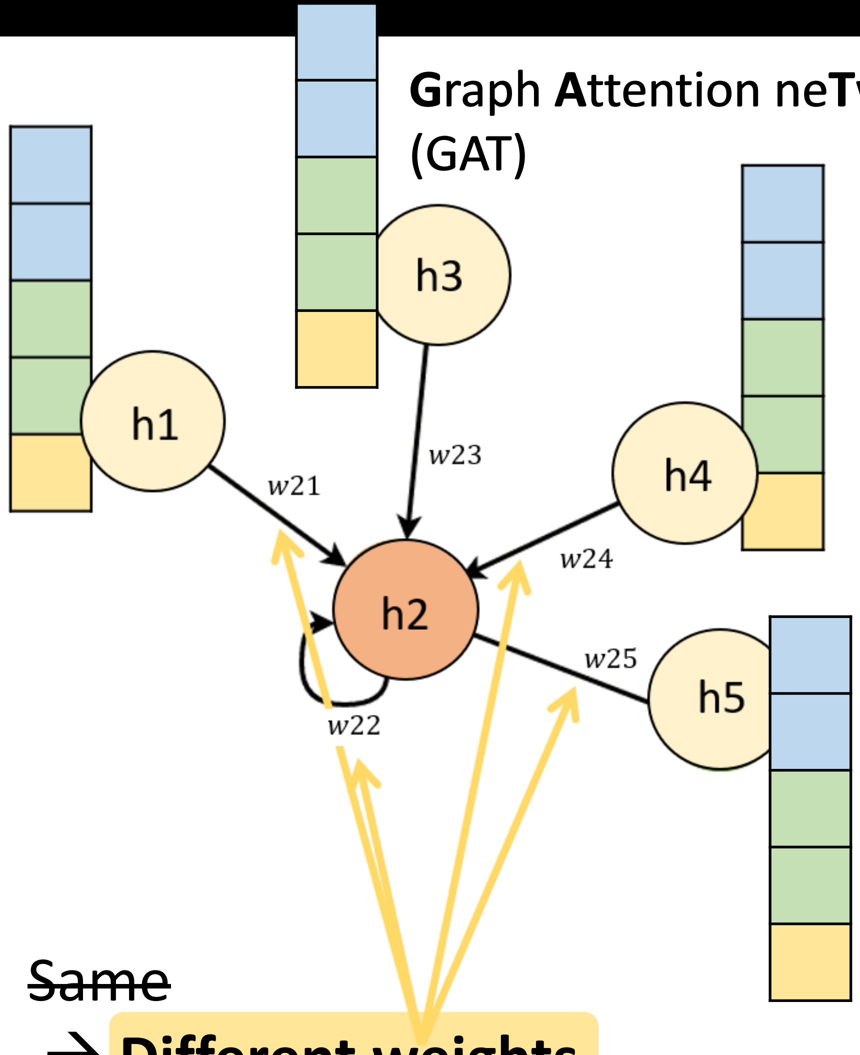




Energy Reconstruction

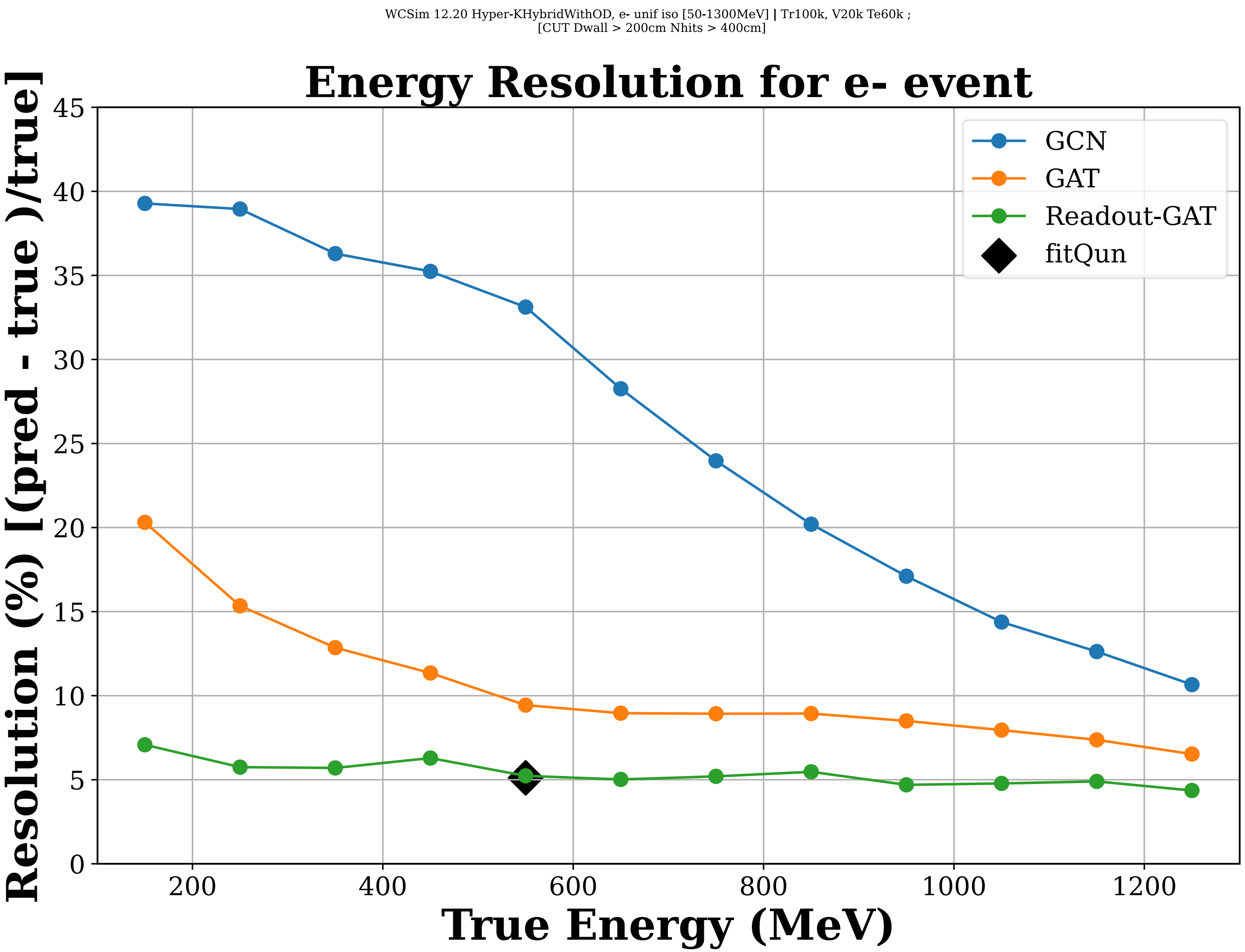


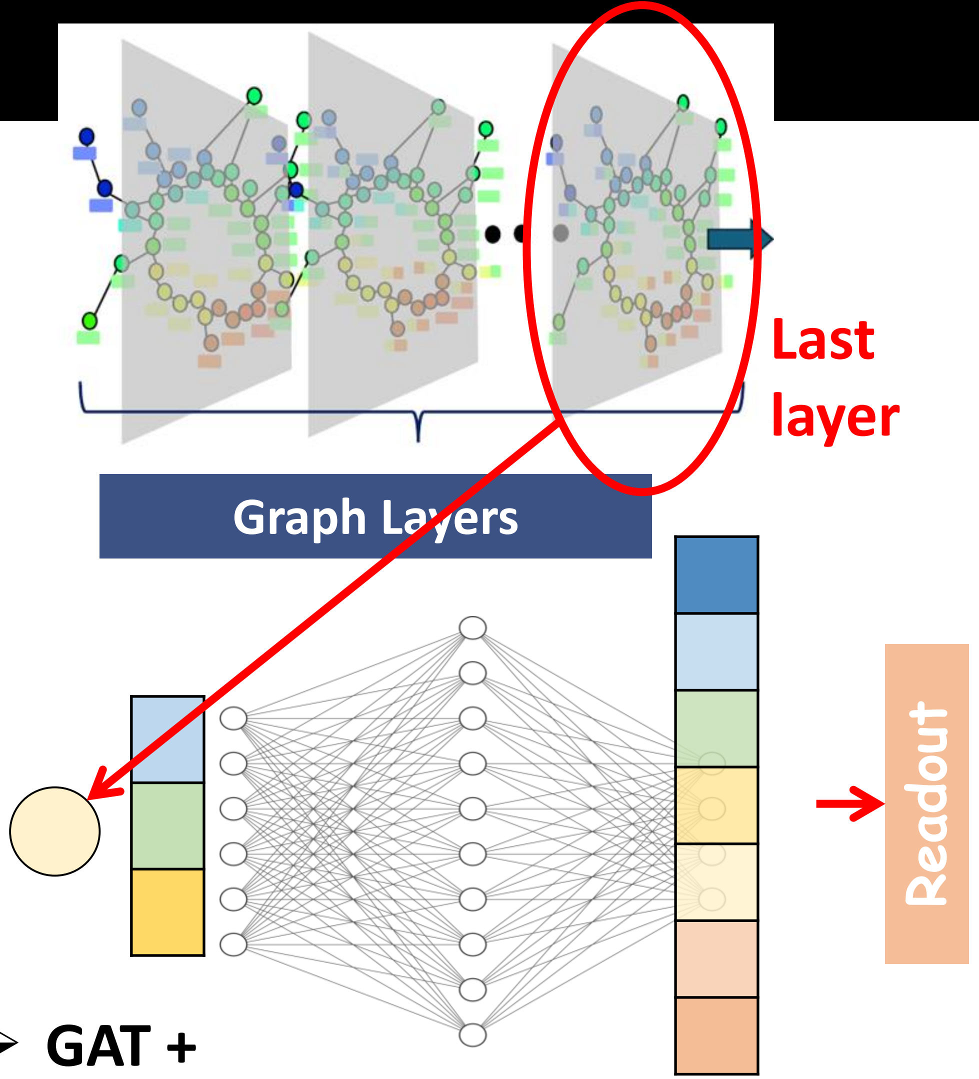
Graph Attention neTwork (GAT)



~~Same~~
→ **Different weights,** computed using an **attention mecanism**

Energy Reconstruction

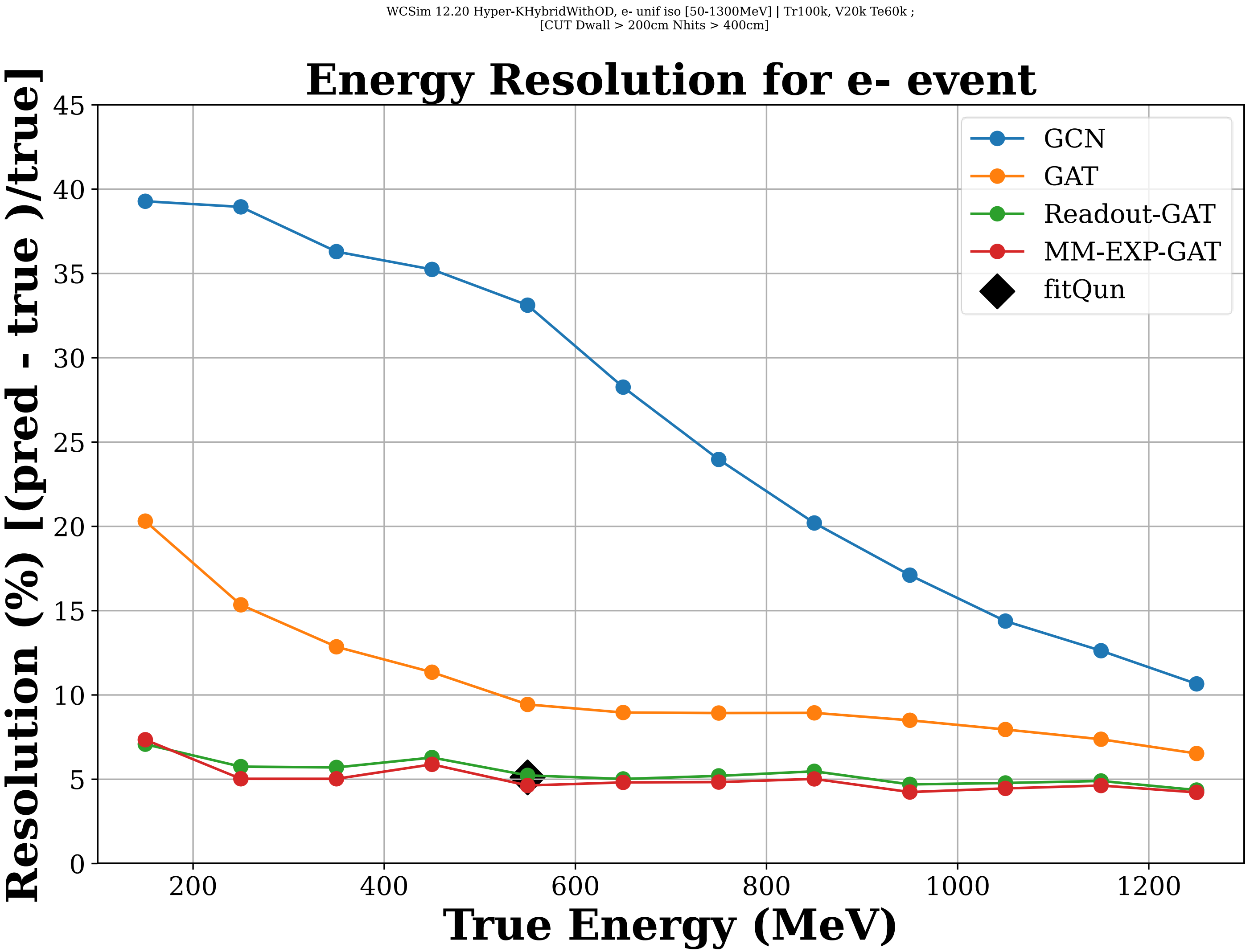


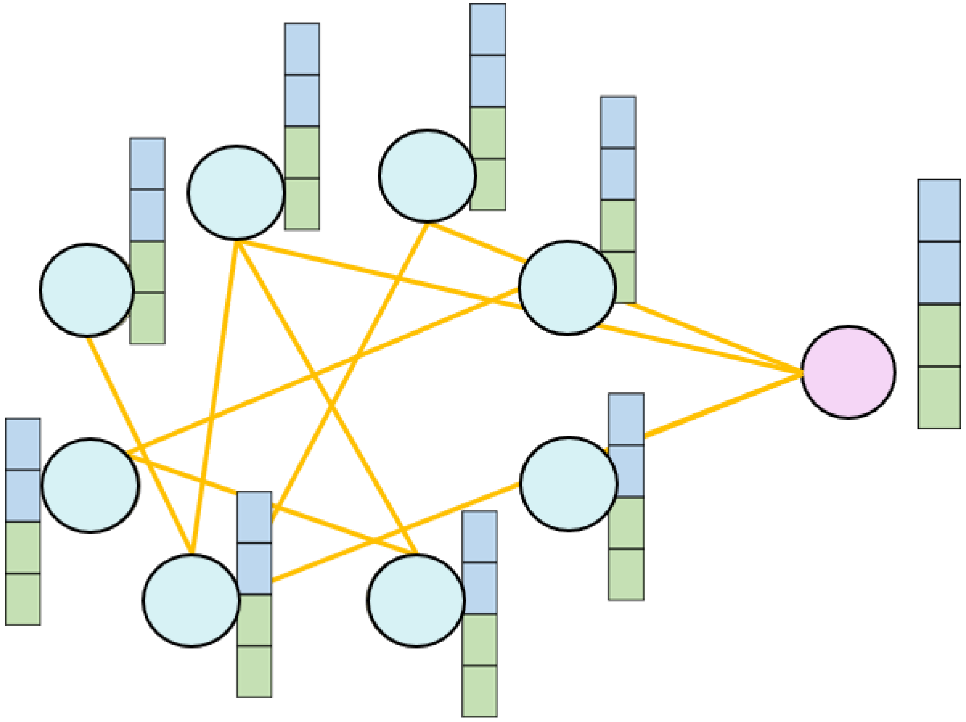


- **GAT +**
- ***Increased last representation before readout***

X_final = MLP(X_last)

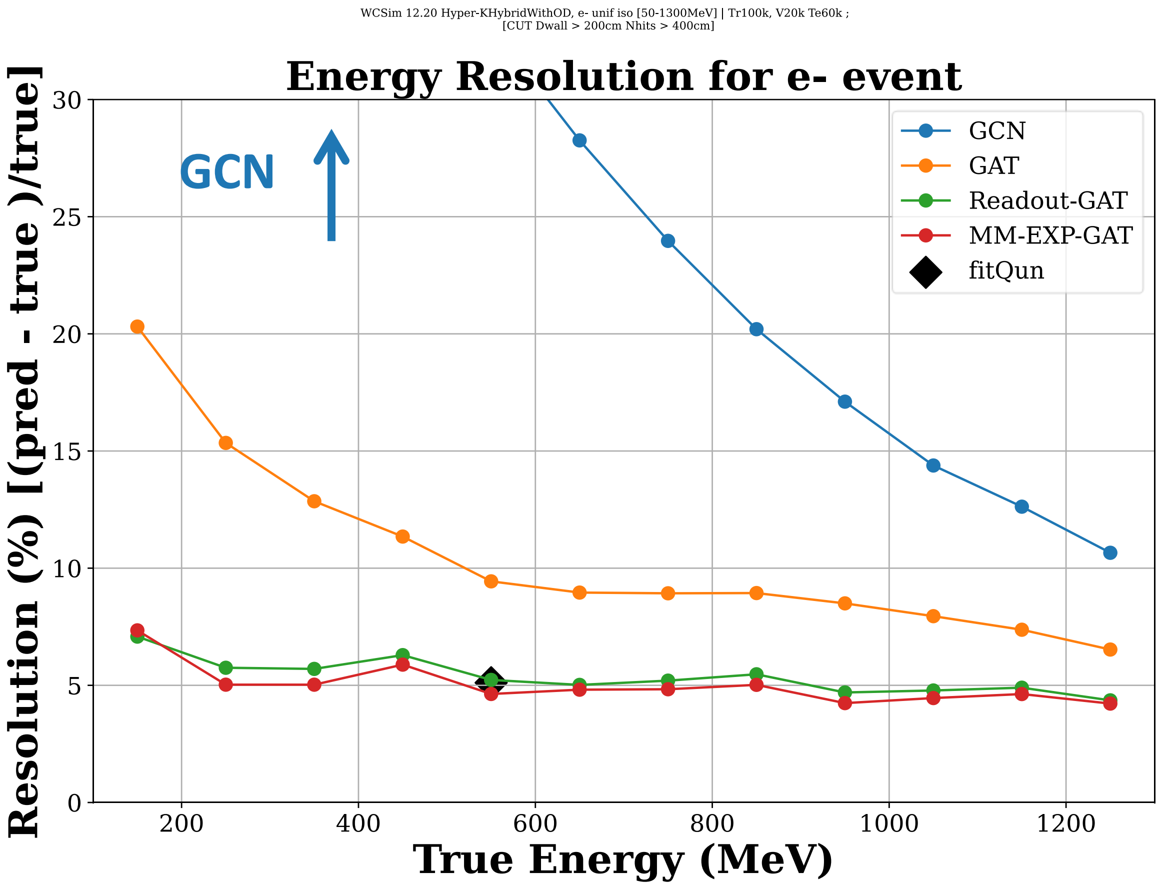
Energy Reconstruction



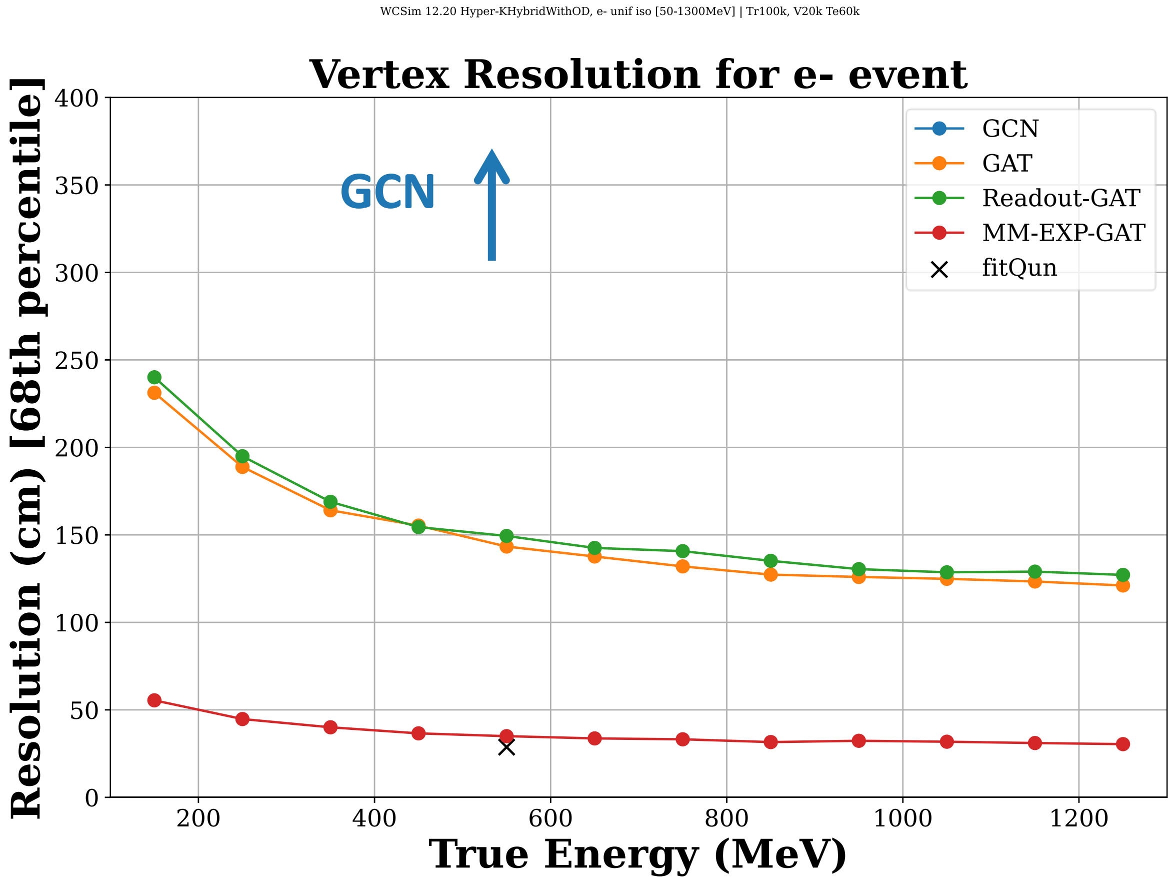


GAT + Virtual Nodes connected to all nodes as memory modules

Energy and Vertex Reconstruction



- Improvements with GCN → GAT
- **Improvements** with GAT + UpSampling
- No improvments with Virtual Node



- Improvements with GCN → GAT
- No improvements with GAT + UpSampling
- **Huge improvement** with Virtual Node

Final States & Prospects

- **Reconstruction status**
  - Achieves better/equal performances to fiTQun at single ring reco. tasks
    - Still room for improvement
  - Excellent improvement in computational time

CPU time / event	1 ring e/μ PID	1 ring e/π^0 PID	Energy & vertex reco.	Total
fiTQun	30s	50s	Simulatenous to PID	80s
CAVERNS	0.09s	0.07s	0.05s	0.11s

~ 2 order of magnitude faster

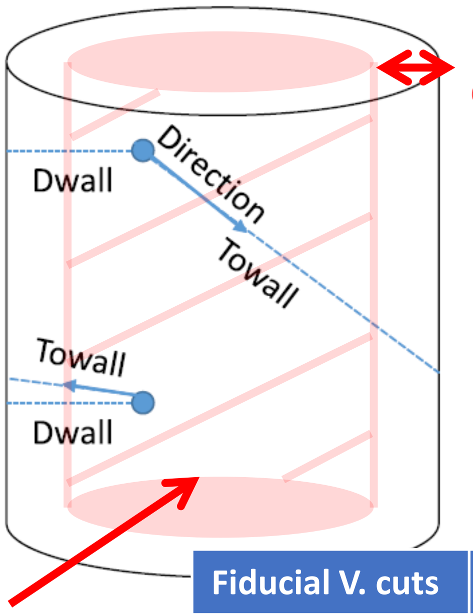
- **Prospects**
  - Multi rings fitting : *To be started soon*

Thank you !

Main limitations of current algorithms

- **Multi rings** events

  (mainly from very energetic neutrinos)



dwall cut

Fiducial volume

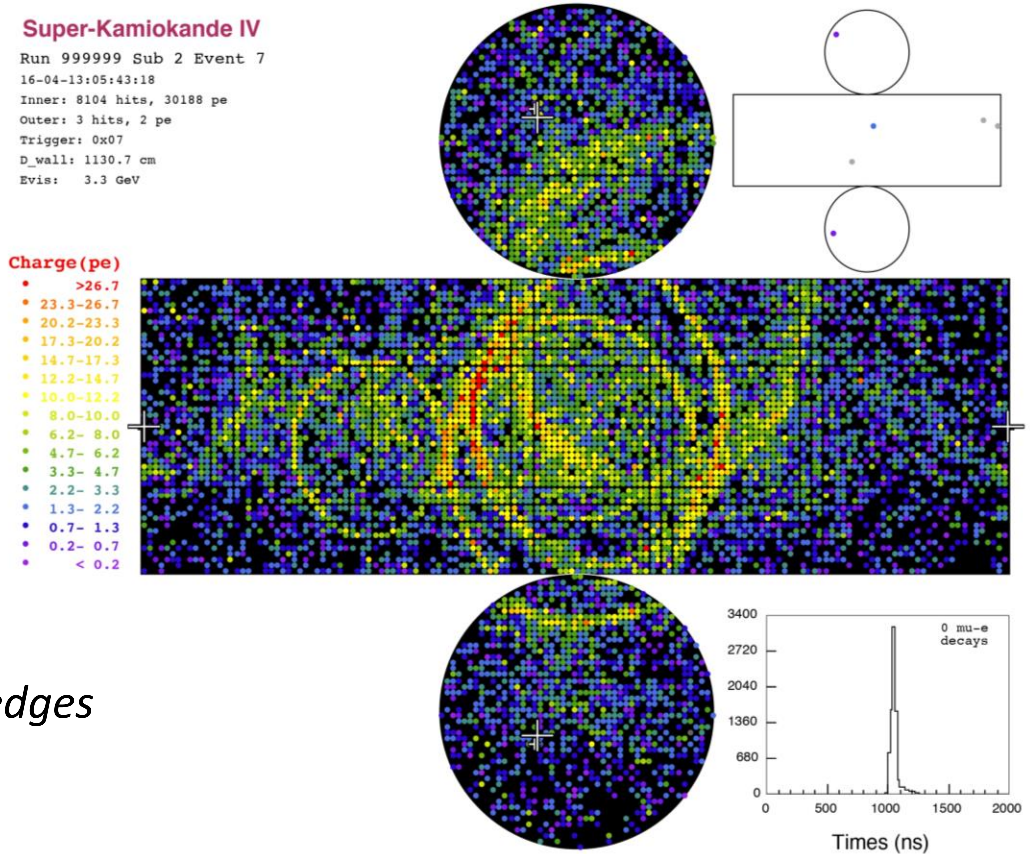
- **Fiducial Volume**

  Limited performances close to the edges

  => Limited statistics

Fiducial V. cuts	Super K (cm)	T2K (cm)
Dwall	> 50	> 80
Towall	X	> 170

M. Jiang D. Thesis, Atmospheric ν, 2019 — *T2K, TN465 v1.1, 2024 (for e-)*

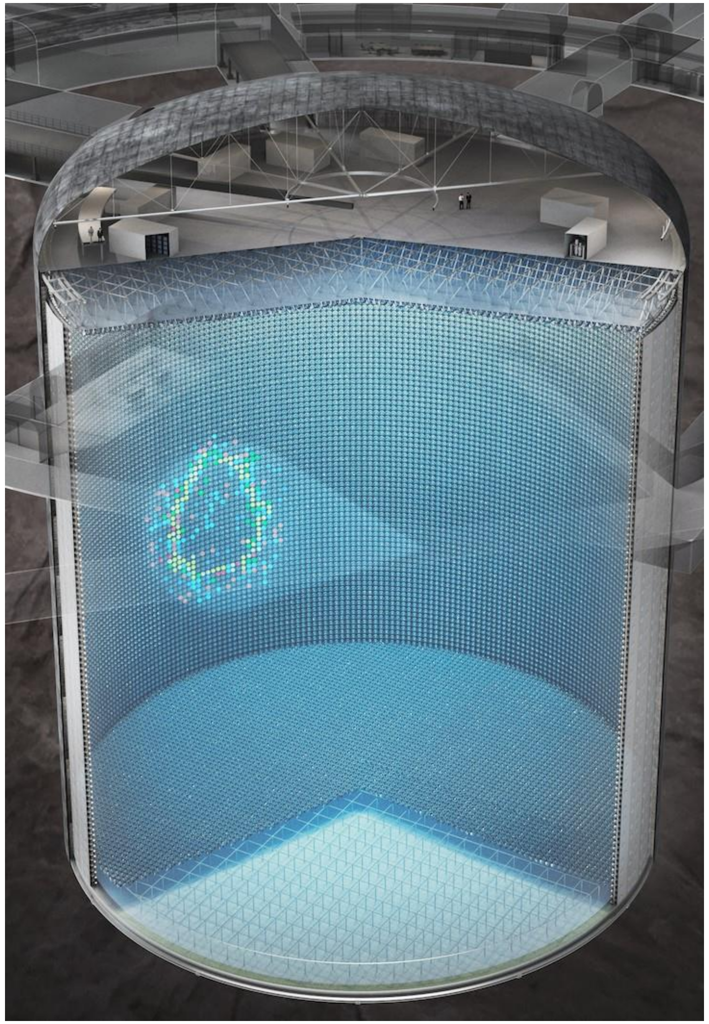


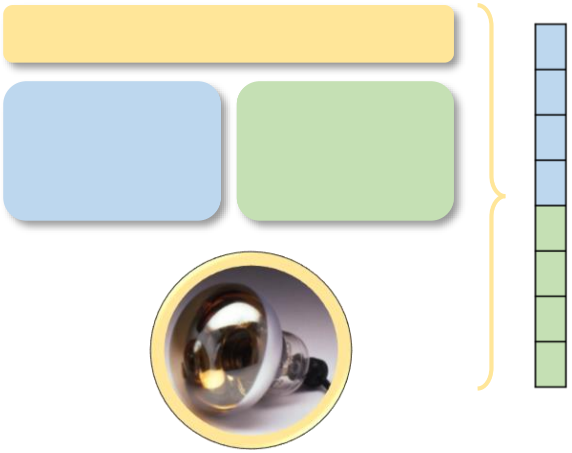
- **Computational Time**

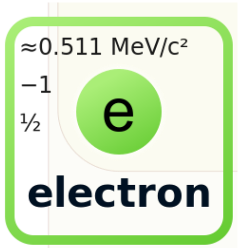
  ~30 seconds / event in **Super K**

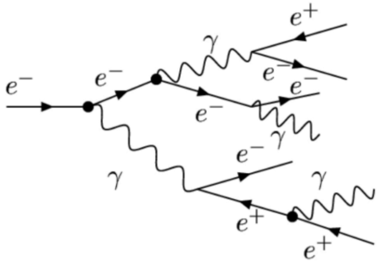
  ~1:30 minute / event in **Hyper K**

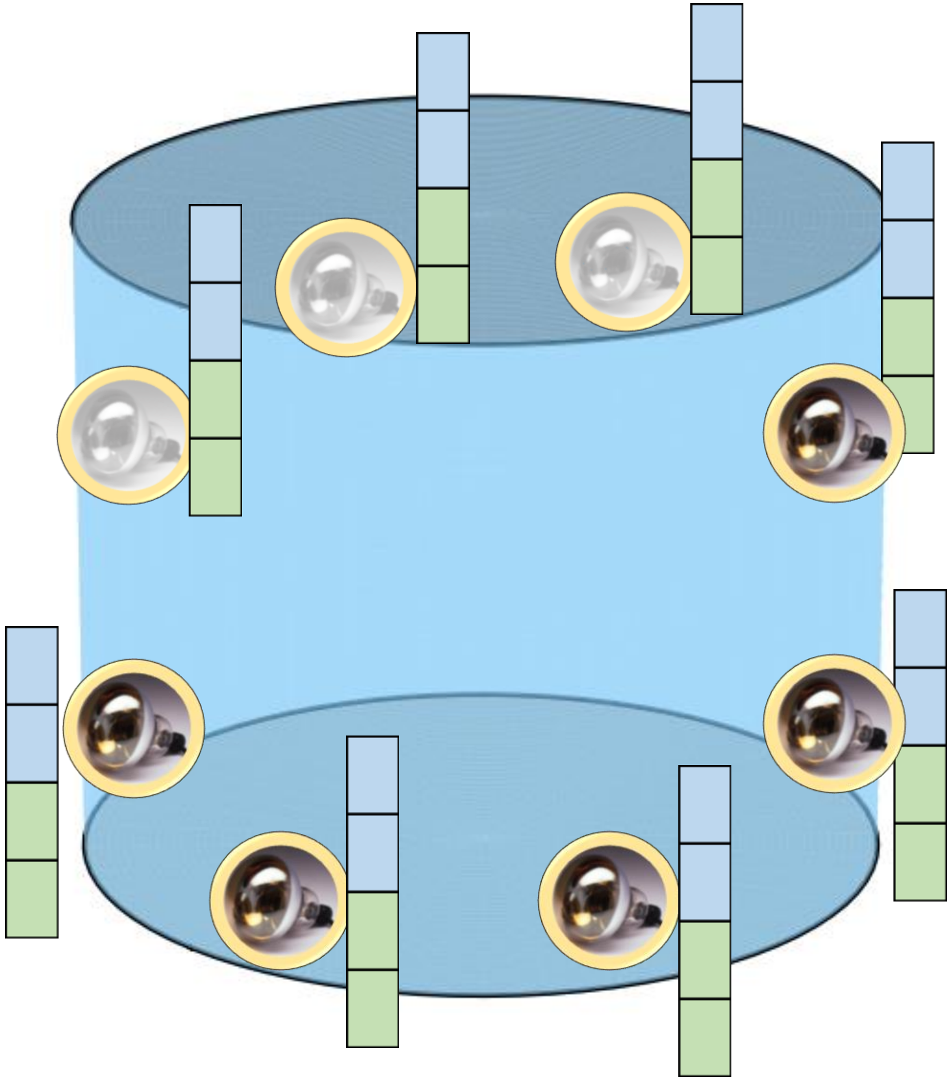
Data Acquisition



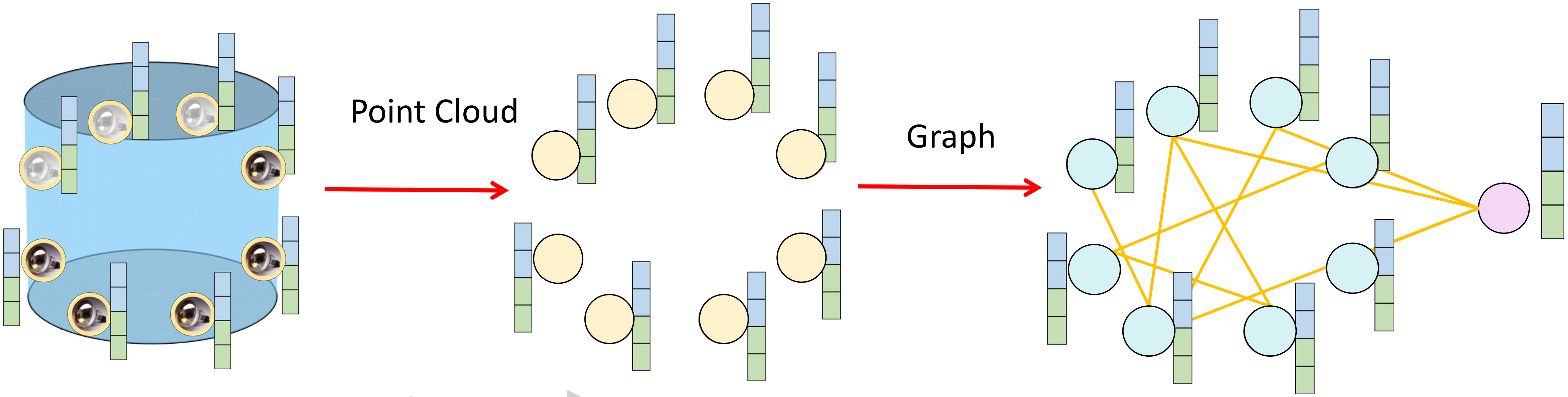




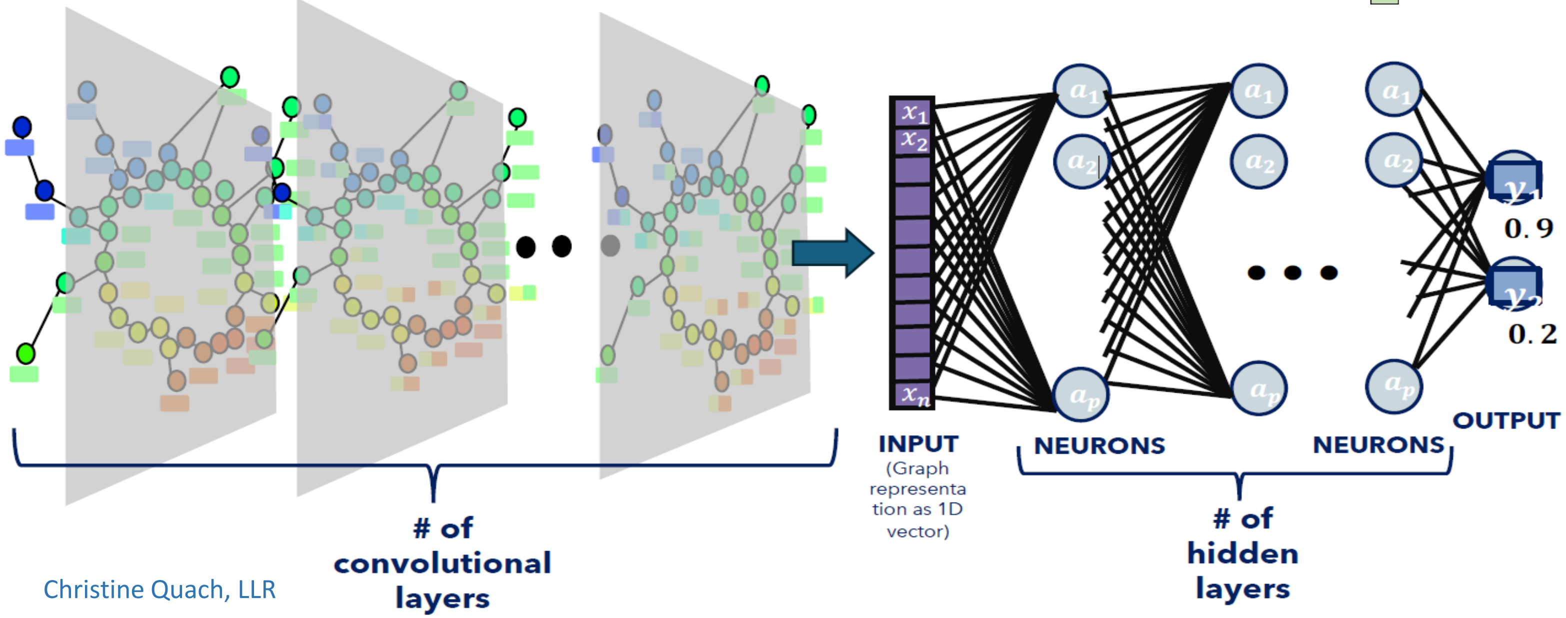




Deep Neural Networks For Reco. In HK : Main architectures



Graph
Neural Networks
(GNN)



Christine Quach, LLR

Graph Convolutional Networks

$$h_v^{(0)} = x_v \quad \text{for all } v \in V.$$

Node v's initial embedding.

... is just node v's original features.

and for $k = 1, 2, \ldots$ upto K:

$$h_v^{(k)} = f^{(k)}\left(W^{(k)} \cdot \frac{\sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}}{|\mathcal{N}(v)|} + B^{(k)} \cdot h_v^{(k-1)}\right) \quad \text{for all } v \in V.$$

Node v's embedding at step k.

Mean of v's neighbour's embeddings at step $k-1$.

Node v's embedding at step $k-1$.

Color Codes:

- Embedding of node v.
- Embedding of a neighbour of node v.
- (Potentially) Learnable parameters.

https://openreview.net/forum?id=SJU4ayYgl

One convolution step

(single head)

$h_v^{(0)} = x_v$ for all $v \in V$.

Node v's initial embedding. ... is just node v's original features.

and for $k = 1, 2, \ldots$ upto K:

$$h_v^{(k)} = f^{(k)}\left(W^{(k)} \cdot \left[\sum_{u \in \mathcal{N}(v)} \alpha_{vu}^{(k-1)} h_u^{(k-1)} + \alpha_{vv}^{(k-1)} h_v^{(k-1)}\right]\right) \quad \text{for all } v \in V.$$

Node v's embedding at step k.

Weighted mean of v's neighbour's embeddings at step $k-1$.

Node v's embedding at step $k-1$.

where the attention weights $\alpha^{(k)}$ are generated by an attention mechanism $A^{(k)}$, normalized such that the sum over all neighbours of each node v is 1:

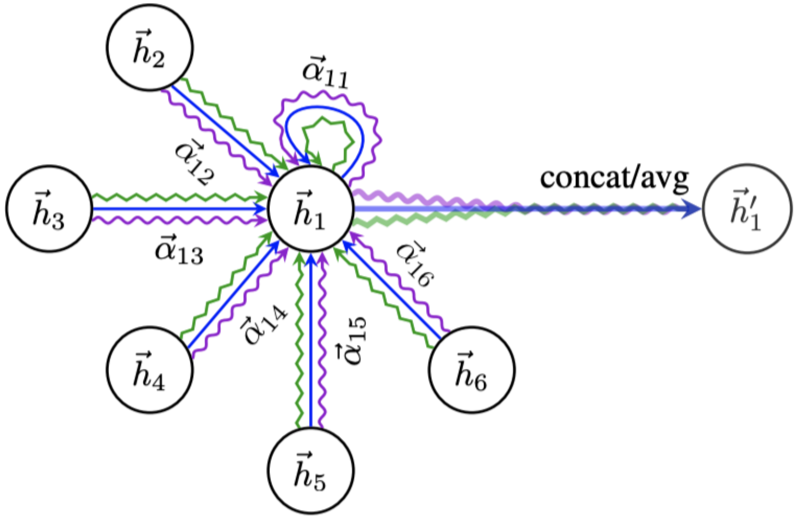
$$\alpha_{vu}^{(k)} = \frac{A^{(k)}(h_v^{(k)}, h_u^{(k)})}{\sum_{w \in \mathcal{N}(v)} A^{(k)}(h_v^{(k)}, h_w^{(k)})} \quad \text{for all } (v, u) \in E.$$

Color Codes:

- Embedding of node v.
- Embedding of a neighbour of node v.
- (Potentially) Learnable parameters.

https://openreview.net/forum?id=rJXMpikCZ

Multi head Attention case



PID

Performance Comparison | PID - ROC Curves

Particle Identification

1. Roc Curves **2. Energy** **3. Dwall** **4. N digi hits**

Efficiency : Rec. Positive **And** True Positive / Total true **positive** → *(capacity to find the true class)*

Acceptance : Rec. Positive **But** True Negative / Total true **negative**

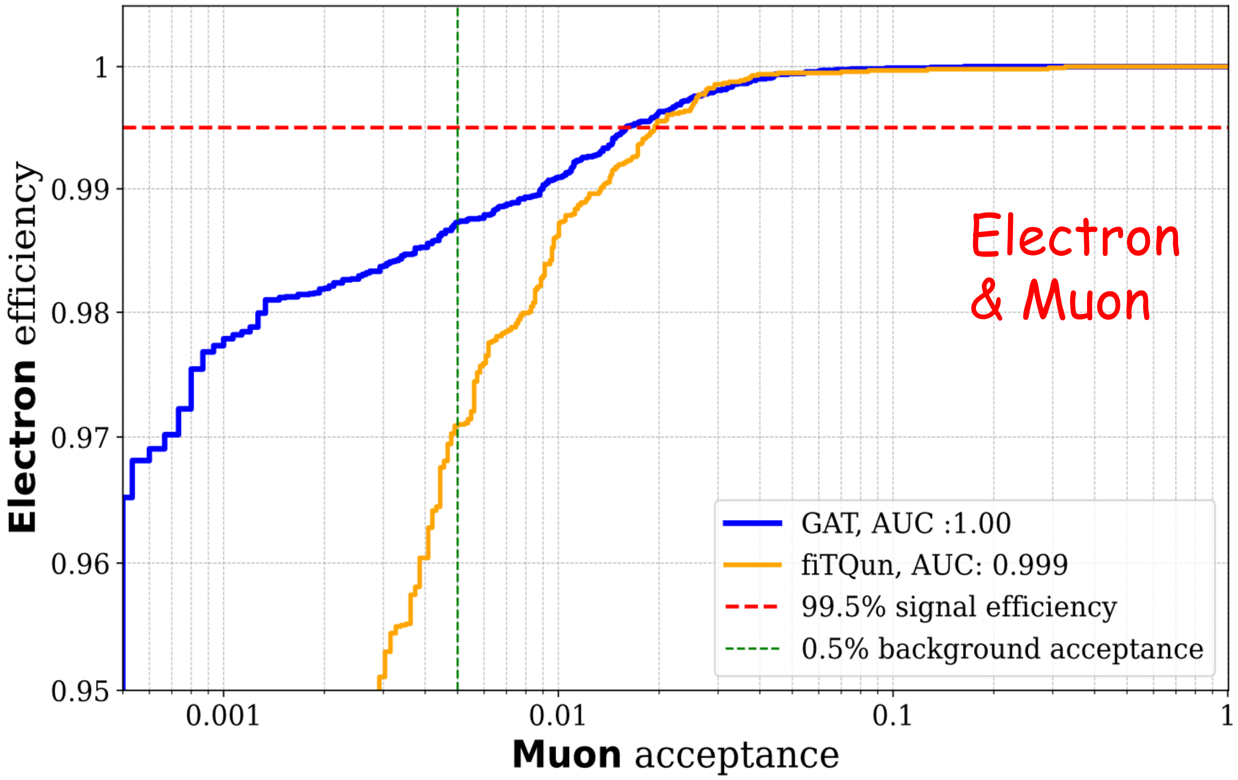
Performance Comparison | PID - ROC Curves

Particle Identification

1. Roc Curves   2. Energy   3. Dwall   4. N digi hits

Efficiency : Rec. Positive **And** True Positive / Total true **positive** → *(capacity to find the true class)*

Acceptance : Rec. Positive **But** True Negative / Total true **negative**



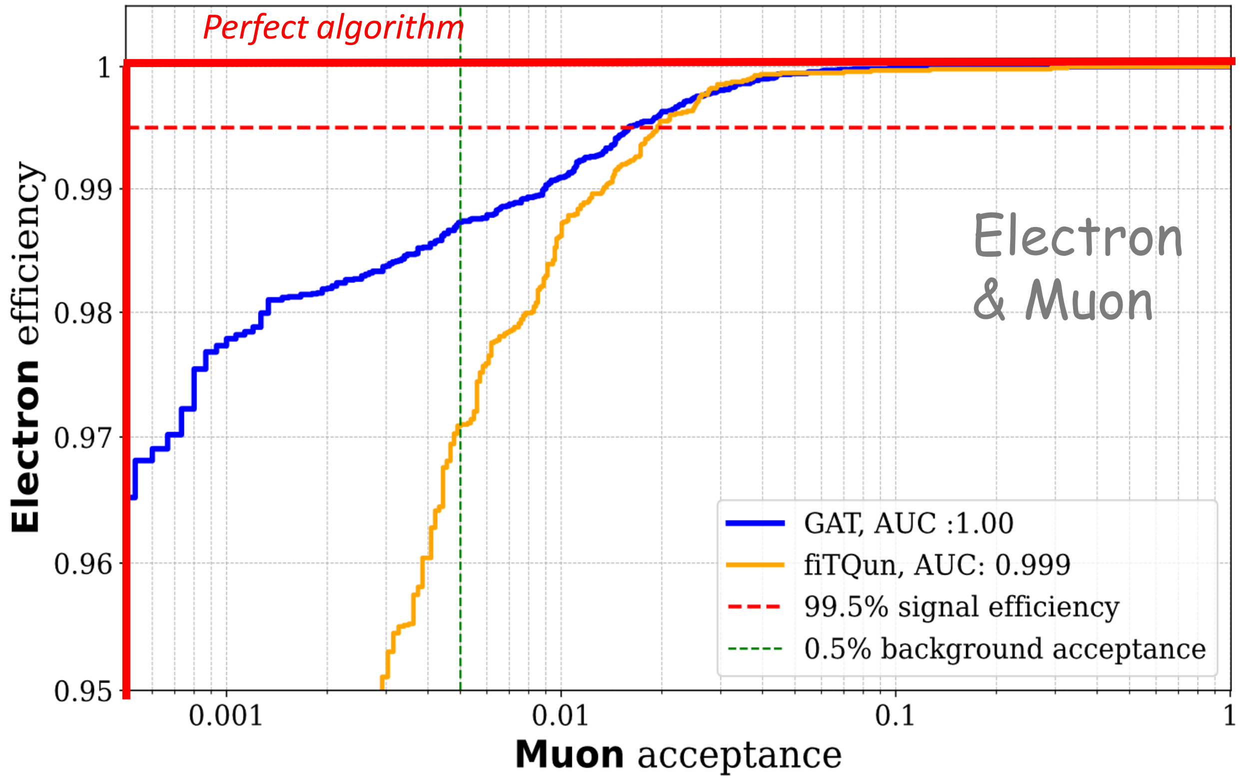
Performance Comparison | PID - ROC Curves

Particle Identification

1. Roc Curves **2. Energy** **3. Dwall** **4. N digi hits**

Efficiency : Rec. Positive **And** True Positive / Total true **positive** → *(capacity to find the true class)*

Acceptance : Rec. Positive **But** True Negative / Total true **negative**



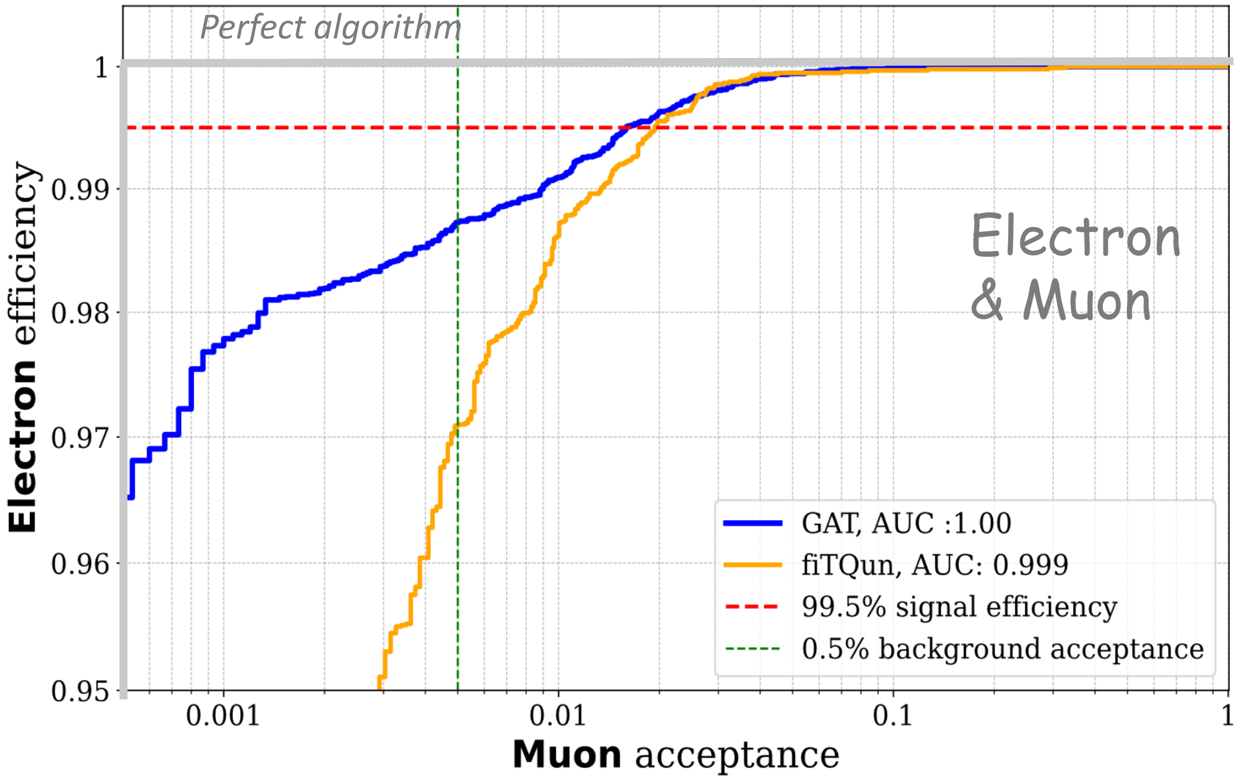
Performance Comparison | PID - ROC Curves

Particle Identification

1. Roc Curves **2. Energy** **3. Dwall** **4. N digi hits**

Efficiency : Rec. Positive **And** True Positive / Total true **positive** → *(capacity to find the true class)*

Acceptance : Rec. Positive **But** True Negative / Total true **negative**



Electron
& π^0

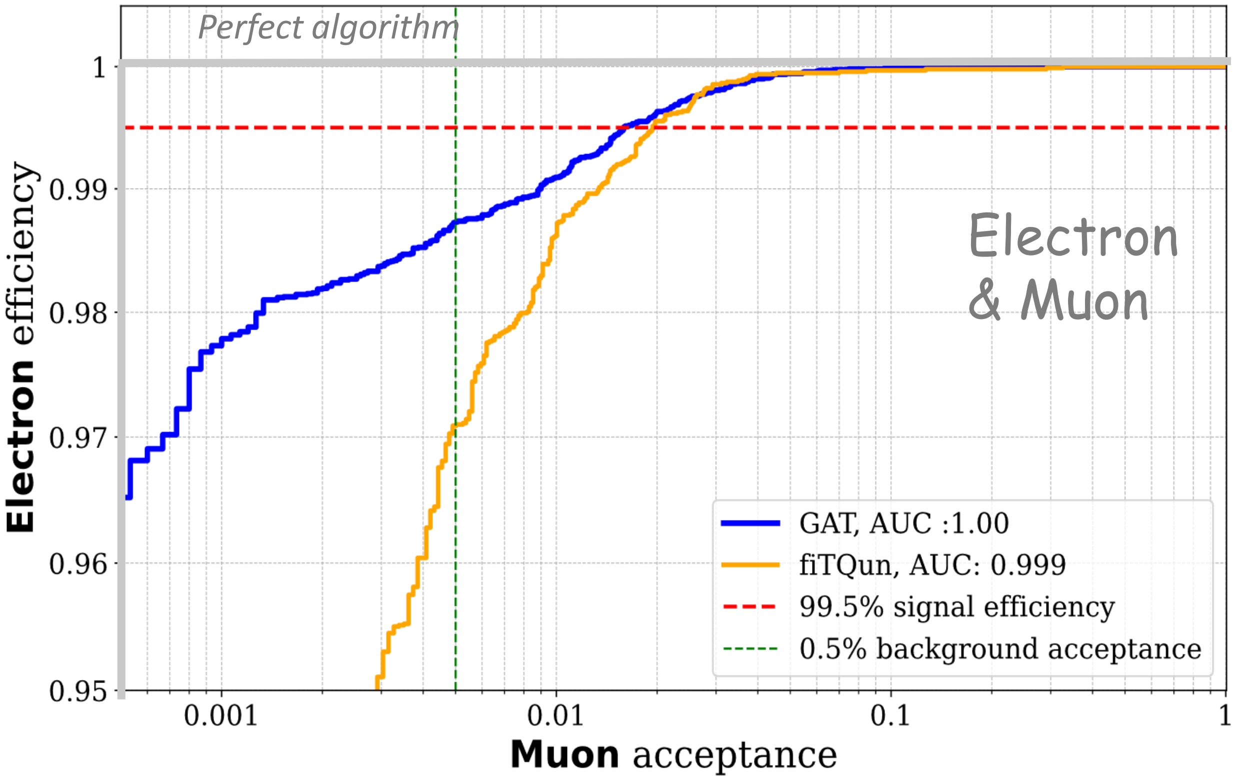
Performance Comparison | PID - ROC Curves

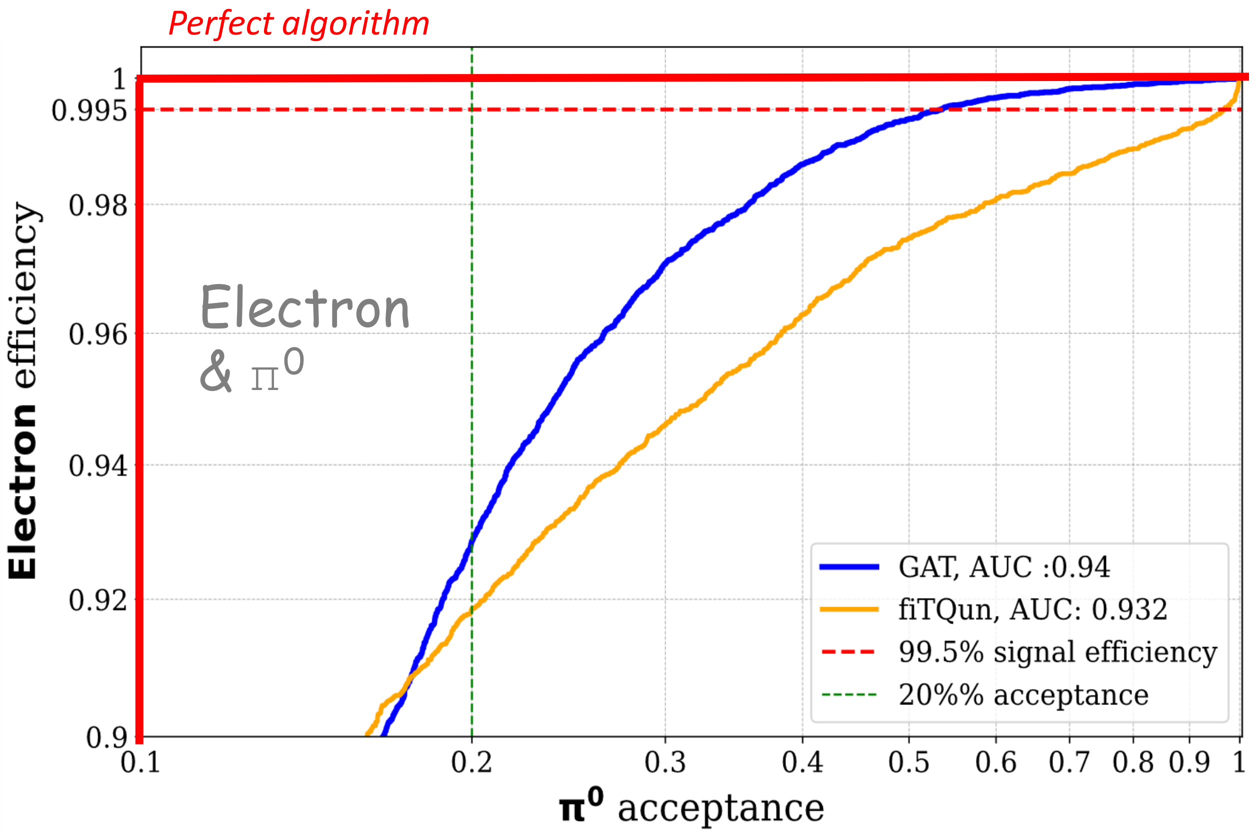
Particle Identification

1. Roc Curves **2. Energy** **3. Dwall** **4. N digi hits**

Efficiency : Rec. Positive **And** True Positive / Total true **positive** ⟶ *(capacity to find the true class)*

Acceptance : Rec. Positive **But** True Negative / Total true **negative**





Performance Comparison | PID - Dwall

Particle Identification

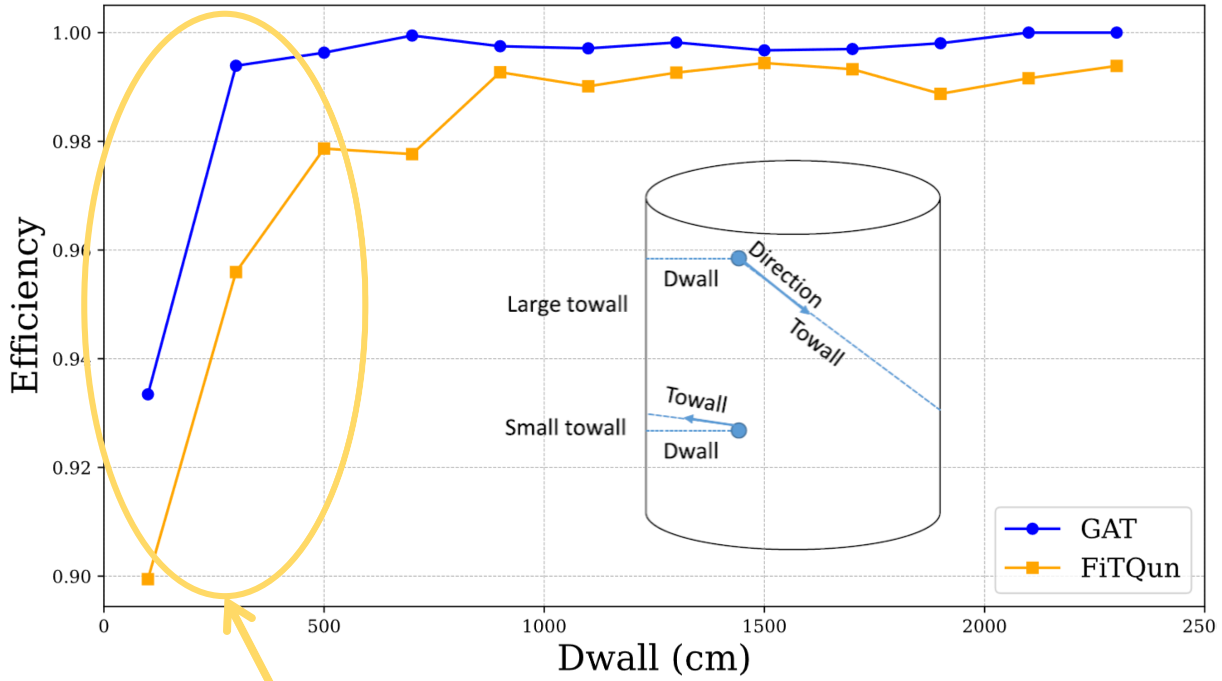
1. Roc Curves

2. Dwall

3. Towall

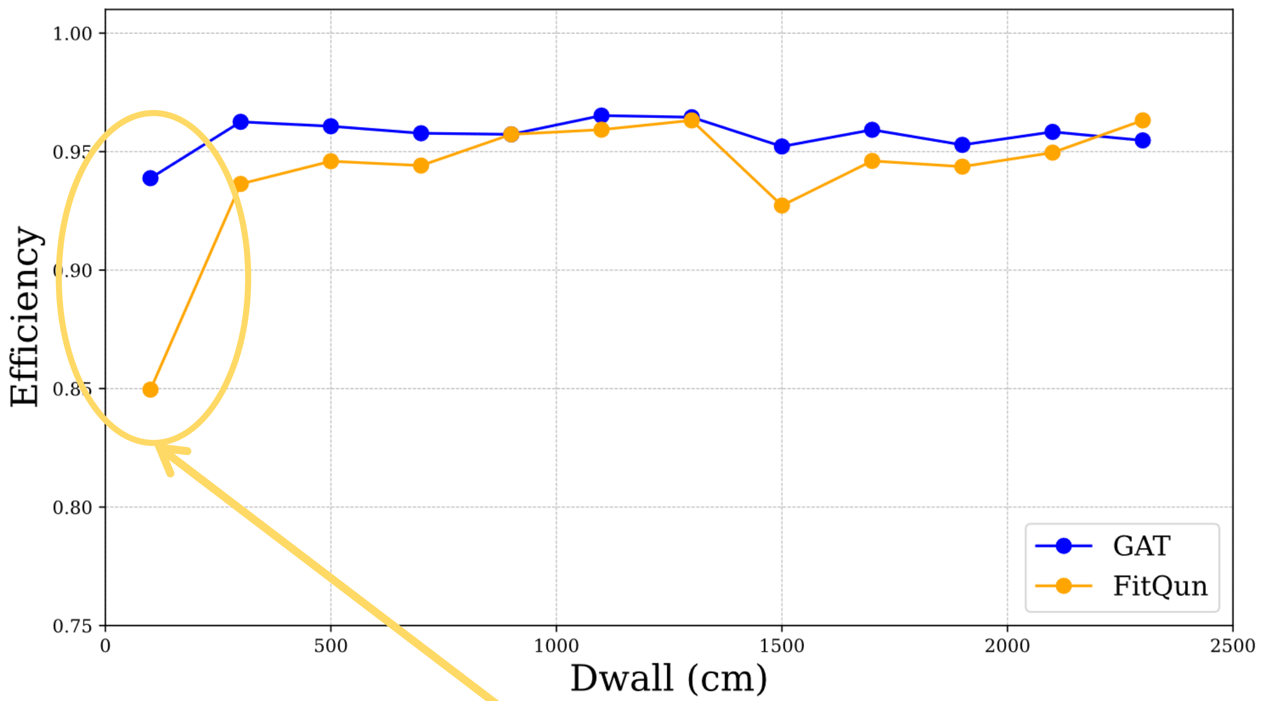
Monitor Fiducial Volume cut impact

Electron & Muon



Efficiency improvement

Electron & π^0



Efficiency improvement

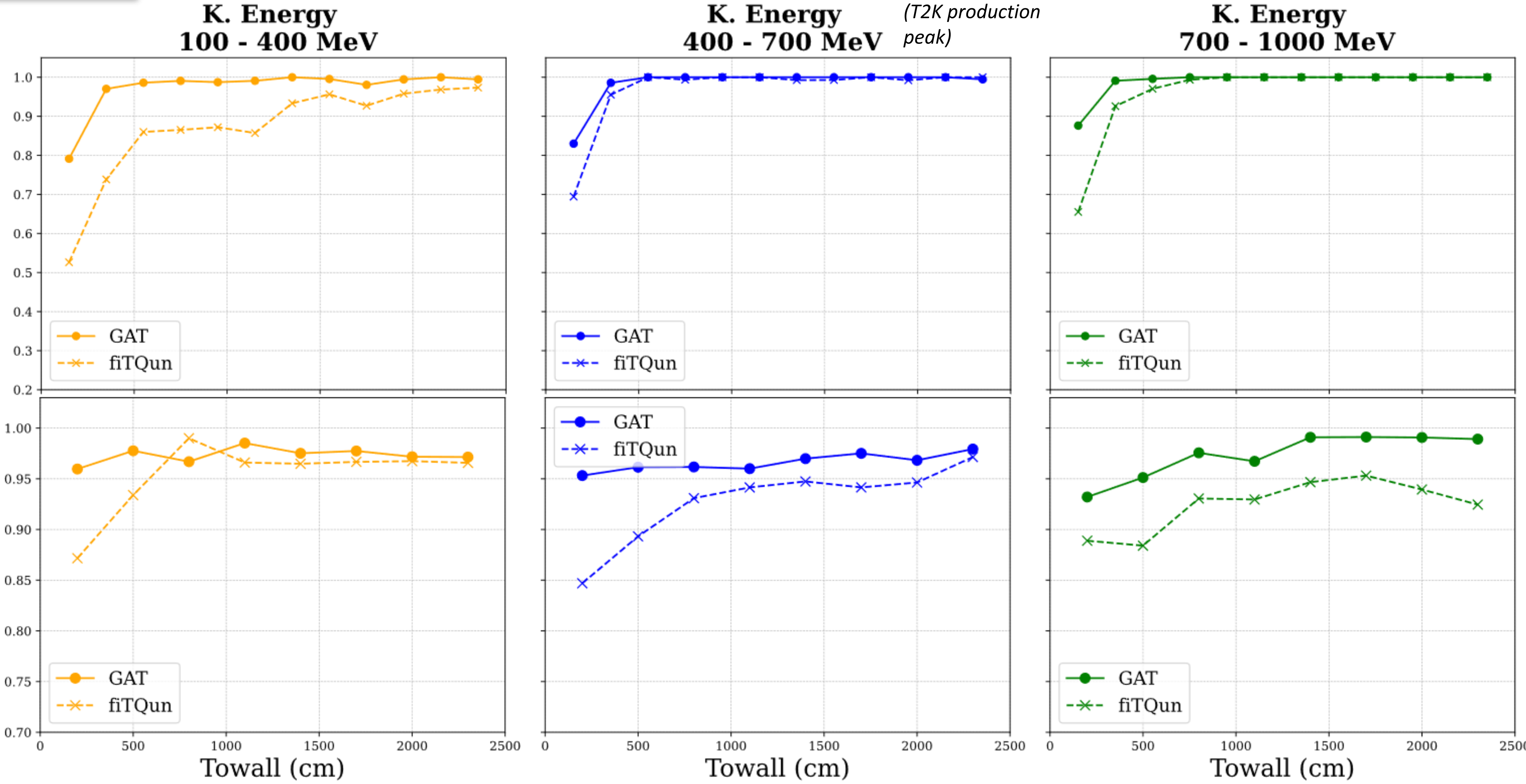
Performance Comparison

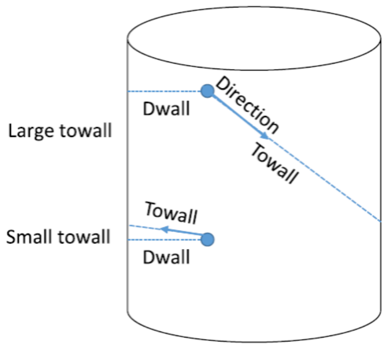
Particle Identification

1. Roc Curves
2. Dwall
3. Towall
4. Towall & Energy

Electron & Muon

Electron & π^0

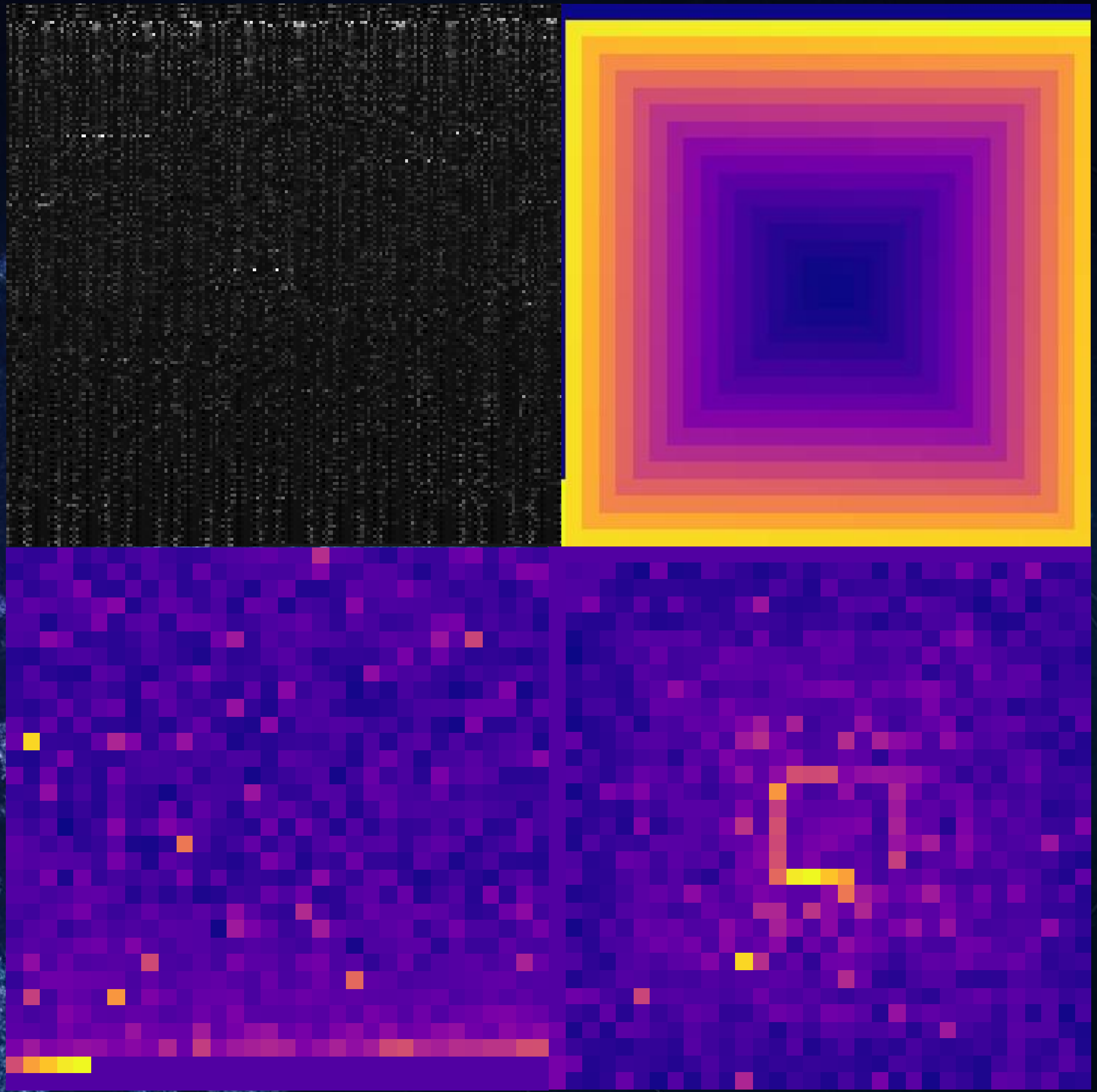




Data flow in GNNs

Investigating the forward process

Work started by N. Moreau



Display settings

Making a spiral, sorted by the pmt hit time.

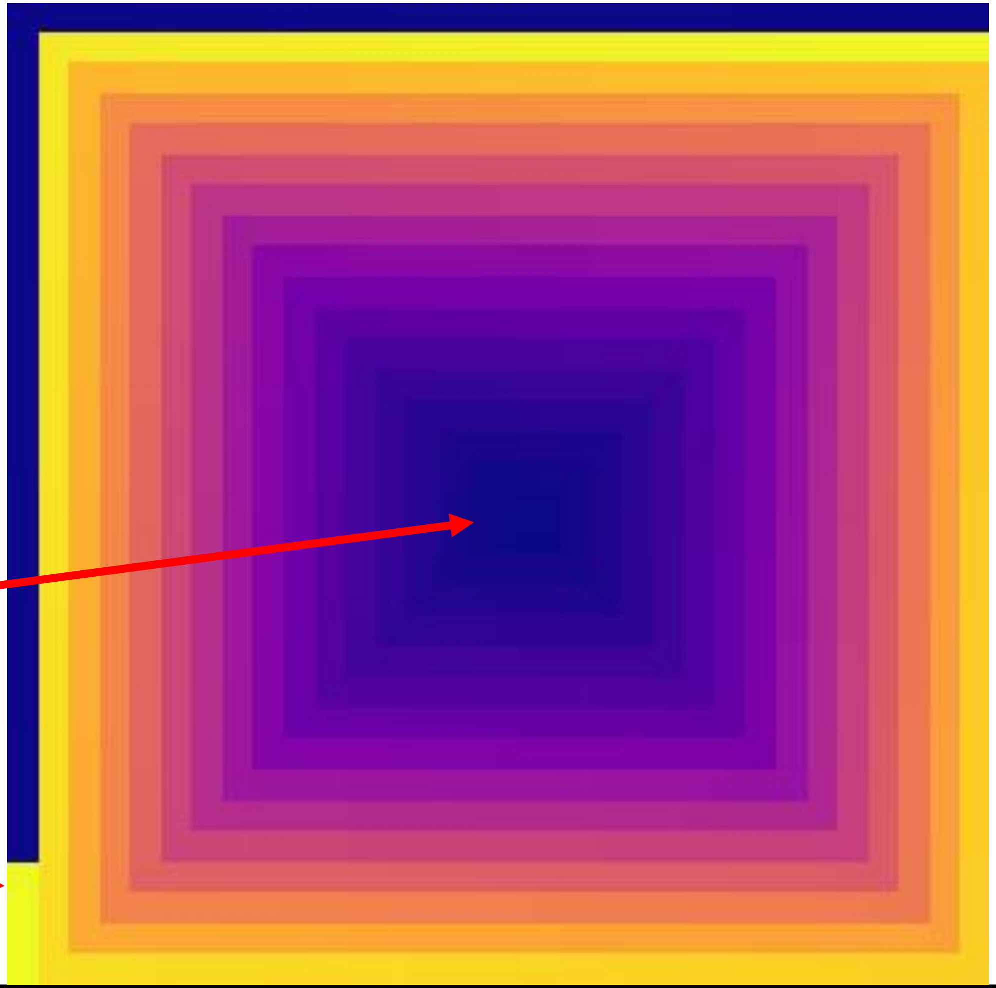
One pixel = one node (PMT)

Center = low pmt hit time

Border = high pmt hit time

32 px (vary with the graph)

32 px

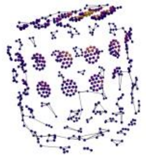


First results

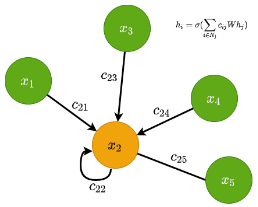
E- 100 -1000MeV
After training for e- / mu- classification

Path to see what is the model « looking » at !

Purple = low importance
Yellow = high

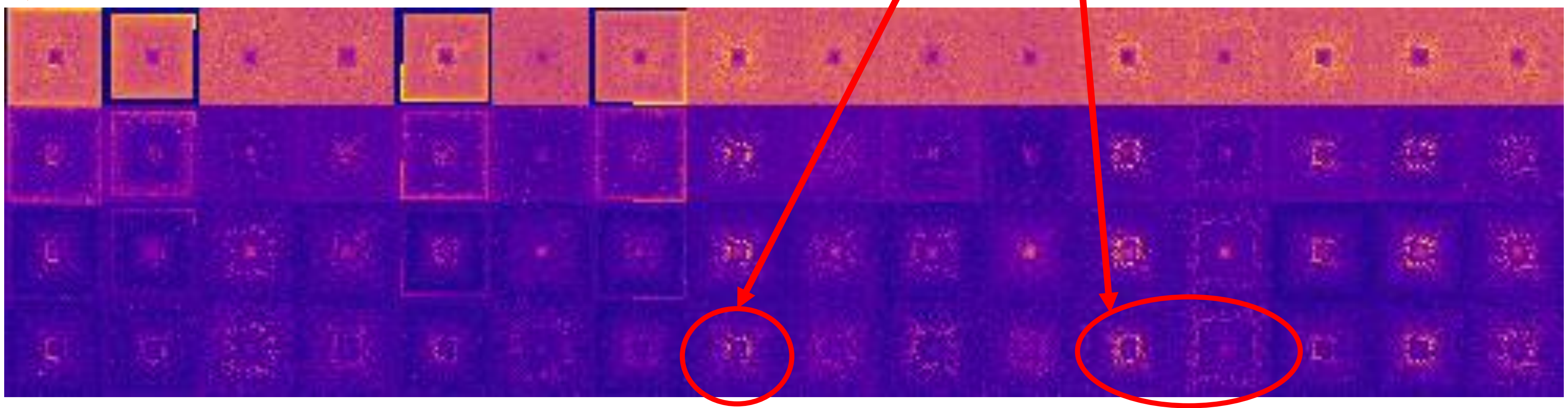


One graph



One convolution step

Encoded graph representation fed to MLP



Challenges for GNNs v2