IRIS-HEP Summer Fellowship 2021

# Graph Generative Models for Fast Detector Simulations in Particle Physics

**Presented by:** Ali Hariri (EPFL)

**Mentors:**

-Prof. Sergei Gleyzer (University of Alabama)

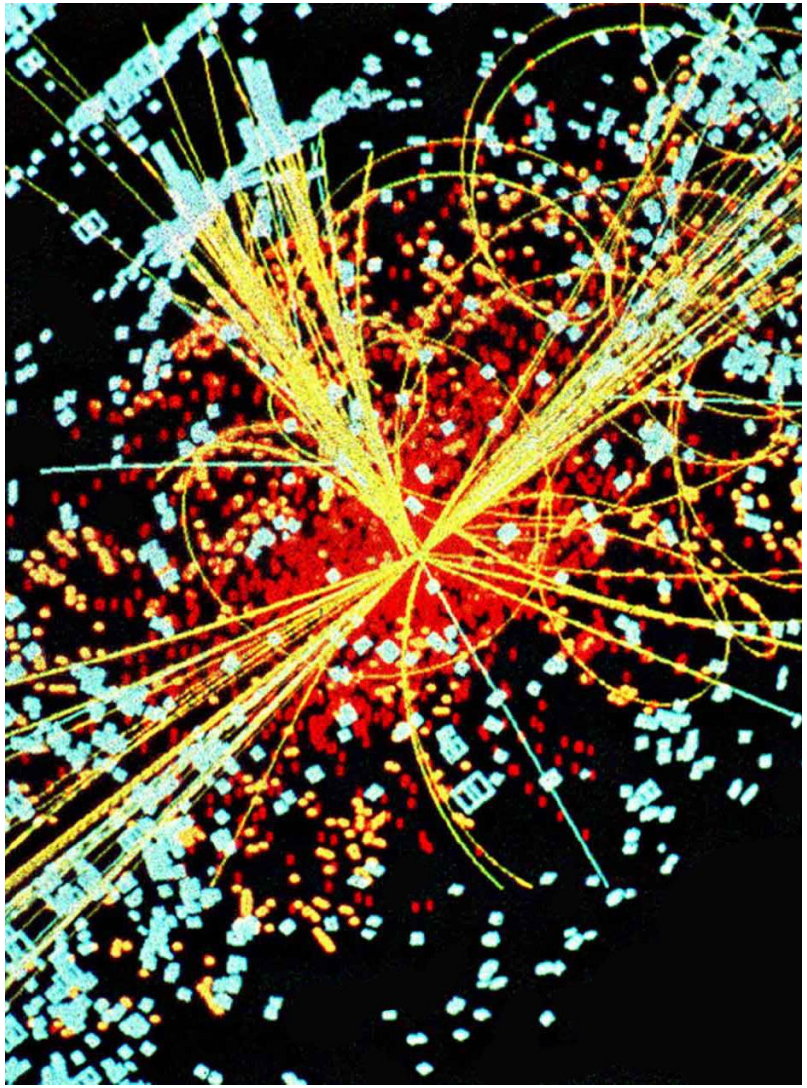
-Prof. Michelle Kuchera (Davidson College)

-Prof. Harrison Prosper (Florida State University)

# Conventional physics simulations

- Particle collisions taking place at the LHC are very complex. There exists a good first principles model of how particles interact with detectors based on Monte Carlo methods.
- Packages like GEANT were developed to simulate the passage of particles through matter. However, this process is difficult to parallelize and is takes significant time to simulate each event. We distinguish 2 types of simulation frameworks:
  - Parametric simulations: Fast simulation* tools such as Delphes and PGS were developed which offer a user interface to specify detector geometries and other simulation environment variables
  - Non-parametric simulations: Other methods are non-parametric: they rely on detector response simulation using a builder and a simulator.

*Fast Simulation attempts to shorten the time of the full simulation

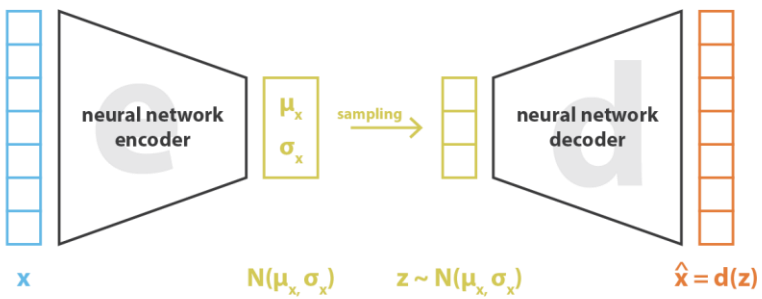


# The rise of Deep Generative Models

- **Deep generative models showed high potential in computer vision applications such as image generation and segmentation.**
- **Their ability to model complex distributions and deal with unlabeled data makes them an interesting candidate for detector physics applications.**
- **The most widely used generative models are Generative Adversarial Networks and Variational Auto-Encoders.**
- **Our work sheds light on Variational Auto-Encoders and their ability to learn the properties of collider simulation events in a latent M-dimensional space.**

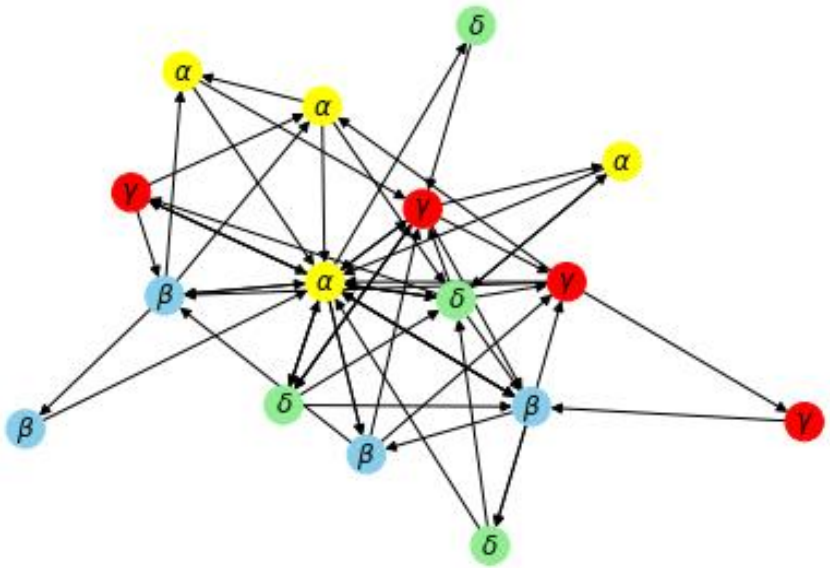
thispersondoesnotexist.com

# Variational Autoencoders as simulators



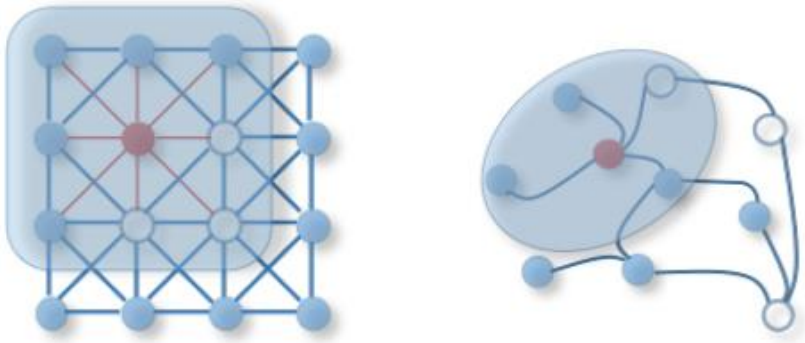

$$\text{loss} = \|x - \hat{x}\|^2 + KL[N(\mu_x, \sigma_x), N(0, I)] = \|x - d(z)\|^2 + KL[N(\mu_x, \sigma_x), N(0, I)]$$

- Variational Auto-Encoders (VAEs) are deep generative models that encode the input data into a latent space as probability distributions.
- Given any multidimensional input x, a VAE aims to embed the probability density of x into a latent space z.
- The true posterior $P(z|x)$ is modeled as standard Gaussian distributions and is approximated using a family of distributions q given by $q_\lambda\ (z|x)$ .
- VAEs show good potential in generative models on images, yet their applications to irregular data remain under-explored. In contrast to regular VAEs whose input consists of a regularly-structured input X, Graph-VAEs take two inputs:
    - Feature Matrix X of shape NxC and Adjacency matrix A of shape NxN.



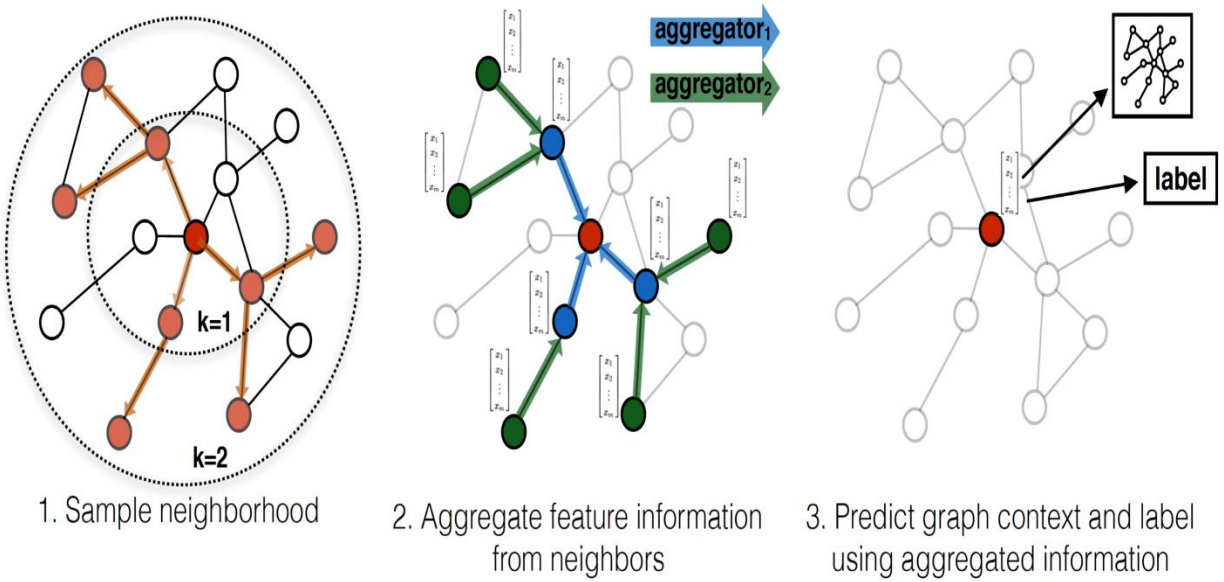
# Graph Neural Networks

- Data is frequently transformed to be treated as a regular structure in the Euclidean domain.
- Numerous types of data are irregular in shape, which raises the need for new approaches to analyze data having a non-Euclidean geometric structure.
- Graphs are an example of a data type whose structure is suitable to represent various complex problems involving interactions between entities.



**Source:** “A Comprehensive Survey on Graph Neural Networks”, Wu et.al

# GraphSAGE

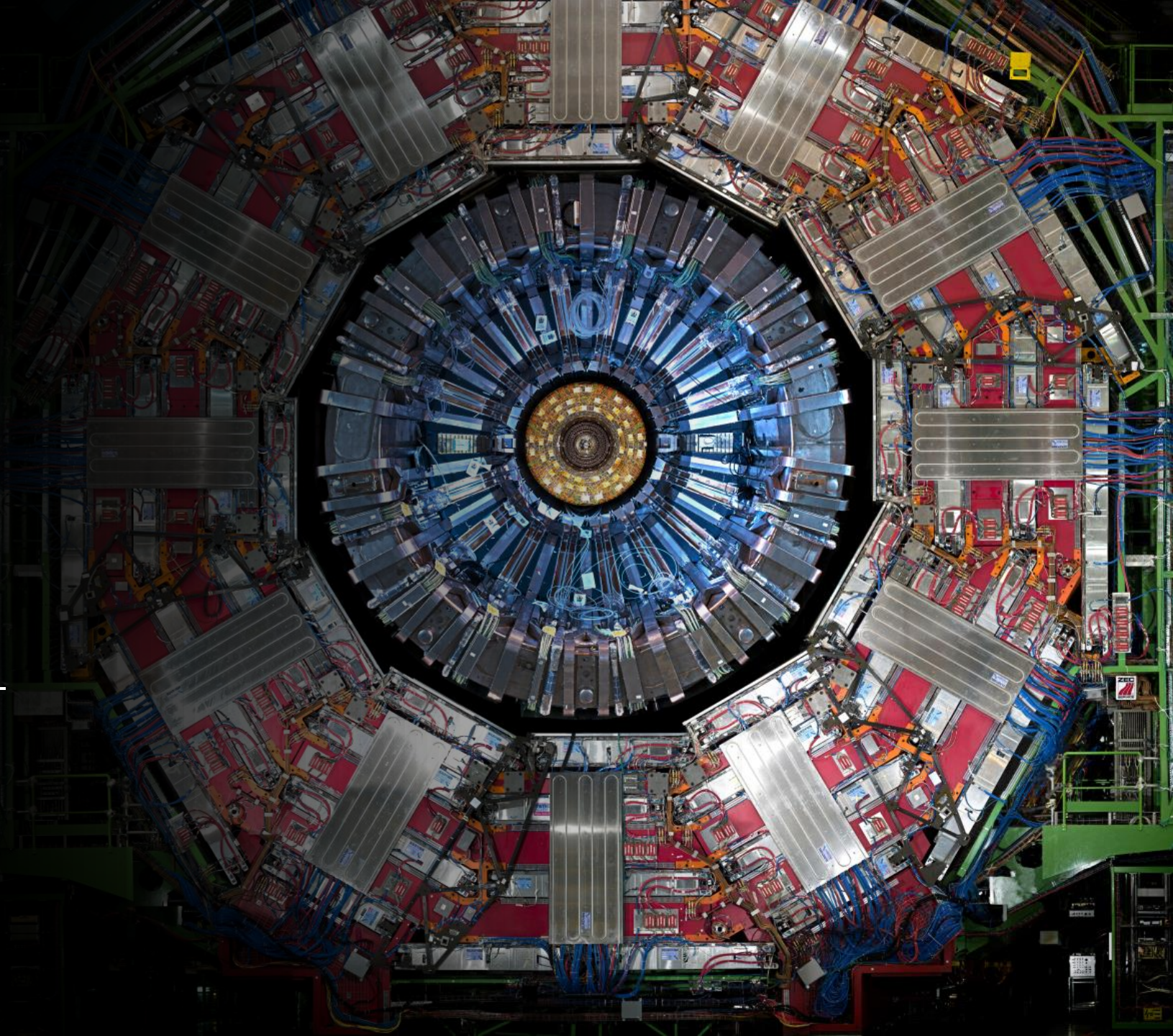



**Algorithm 1** Multi-layer GraphSAGE pseudo-code

0: **Input:** Graph G=(X,A) where X is the feature matrix and A the adjacency matrix in the model's $l^{th}$ layer; X $\in \mathrm{R}^{n \times d}$ and A $\in \mathrm{R}^{n \times n}$ with n=number of nodes in a graph, d=feature dimension.
Let $u$ be a node's features and $\mathcal{V}$ be the neighbourhood of this node.
0: **for** $l = 1, 2, \ldots, L$ **do**
0: **for** $v \in \mathcal{V}$ **do**
0: $h^{L}_{\mathcal{N}(v)} \leftarrow \text{aggregate}_l(h^{l-1}_u, \forall u \in \mathcal{N}(\mathcal{V}))$
0: $h^{l}_{v} \leftarrow \sigma(W^l.CONCAT(h^{l-1}_v, h^{l}_{\mathcal{N}(v)}))$
0: **end for**
0: Normalize the feature vectors prior to message passing:
0: $h^{l}_{v} \leftarrow \frac{h^{l}_{b}}{\|h^{l}_{v}\|}$
0: **end for**
0: Aggregate hidden features to original node:
: $\mathrm{h}_{v+1} \leftarrow h^{l}_{v}, \forall v \in \mathcal{V} = 0$

**Source:** snap.stanford.edu/graphsage/

# Graph Auto-Encoders for Fast Simulation Applications



# Application: Graph Generative Model for Fast Simulation of collider events

This work discusses the potential provided by graph neural networks in high energy physics applications.

The ability of GNNs to learn on isolated hits while disregarding the empty cells surrounding them makes them suitable for sparse detector data.

The learning process takes place on the particle hits exclusively !

We propose geometric deep learning as an alternative methodology in particle physics by using it in Fast simulation

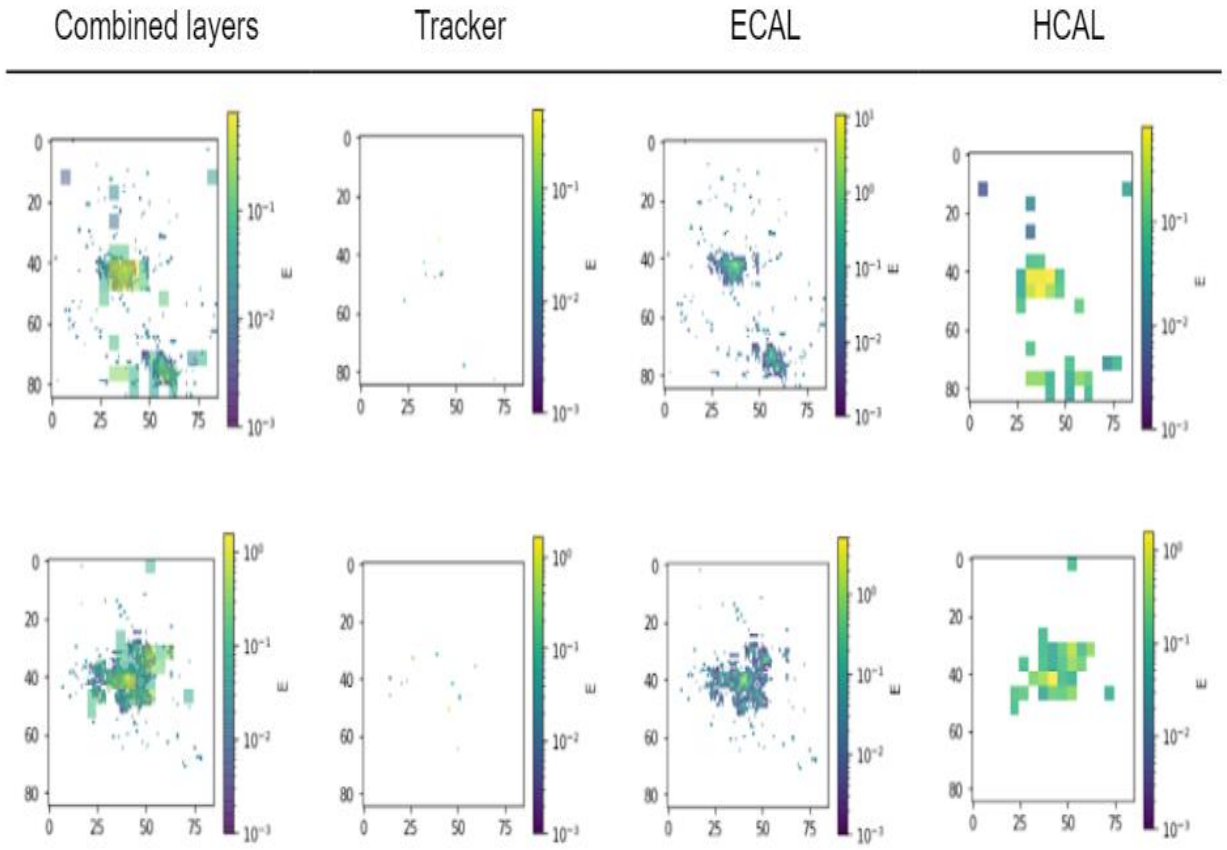
# Data Description

| Key | Type | Size | Value |
|---|---|---|---|
| X_jets | float32 | (3, 125, 125) | [[[0. 0. 0. ... 0. 0. 0.]<br>[0. 0. 0. ... 0. 0. 0.] |

- In this work, we make use of the CMS Open Data release publicly accessible data from the LHC experiments.
- We consider the top quark jets produced using Pythia 6, a program for generating particle collisions events.
- The data consists of almost 30000 samples of 3x125x125 arrays representing the mesh and the segmentation of 3 detector stages: Tracker, ECAL and HCAL subdetectors, respectively.

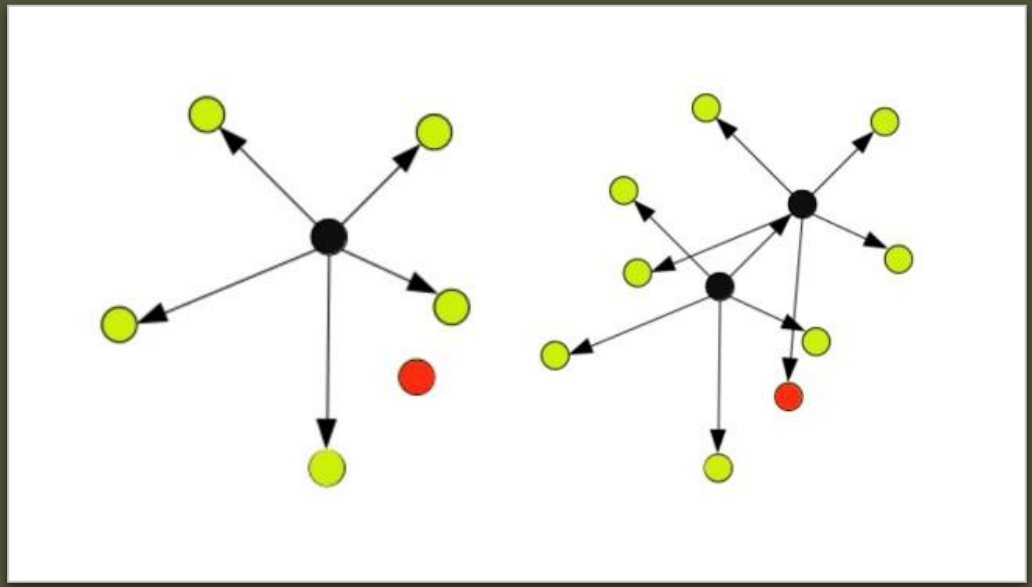
# Data pre-processing

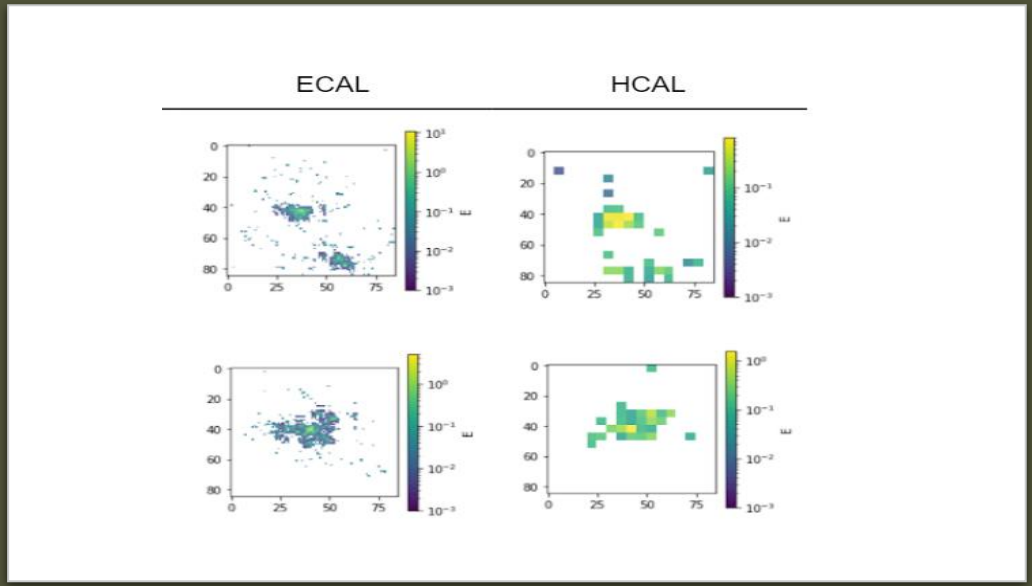
- We select the non-zero hit locations within the array, providing their x and y locations as per the calorimeter segmentation.
- Then we concatenate the x,y locations with their corresponding hit energy at that location.
- Each sample has the shape Nx3 where N is the number of particle hits within the detector for one specific sample jet.



# Graph VAE for Jet Reconstruction with Spectral (Mincut) Pooling

- In contrast to molecular chemistry, jets in particle collisions are not characterized by pre-defined topology.
- We aim to learn the properties of these jets as graphs in addition to their compressed representation.
- We develop a Graph VAE architecture whose encoder embeds the node features into latent space dimensions through Dense GraphSAGE layers.





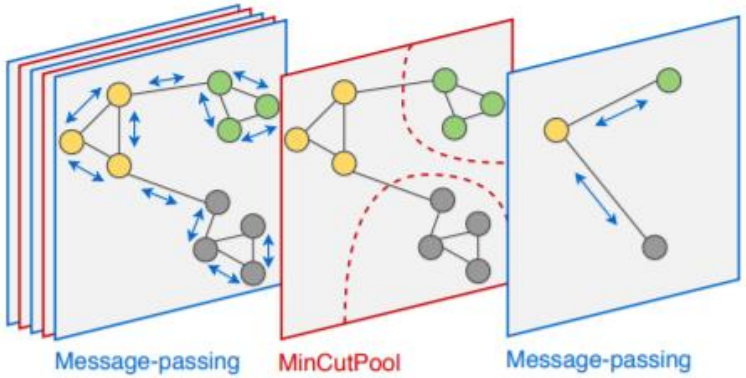
# Graph VAE for Jet Reconstruction with spectral (mincut) pooling

- We then compress them into smaller dimensions using dense mincut graph pooling operations inspired by where spectral clustering of the graph nodes is performed.
- To proceed, the pooled feature matrix and adjacency matrix are calculated as follows:

$$A^{Pooled} = S^T \tilde{A} S; \quad X^{Pooled} = S^T X$$

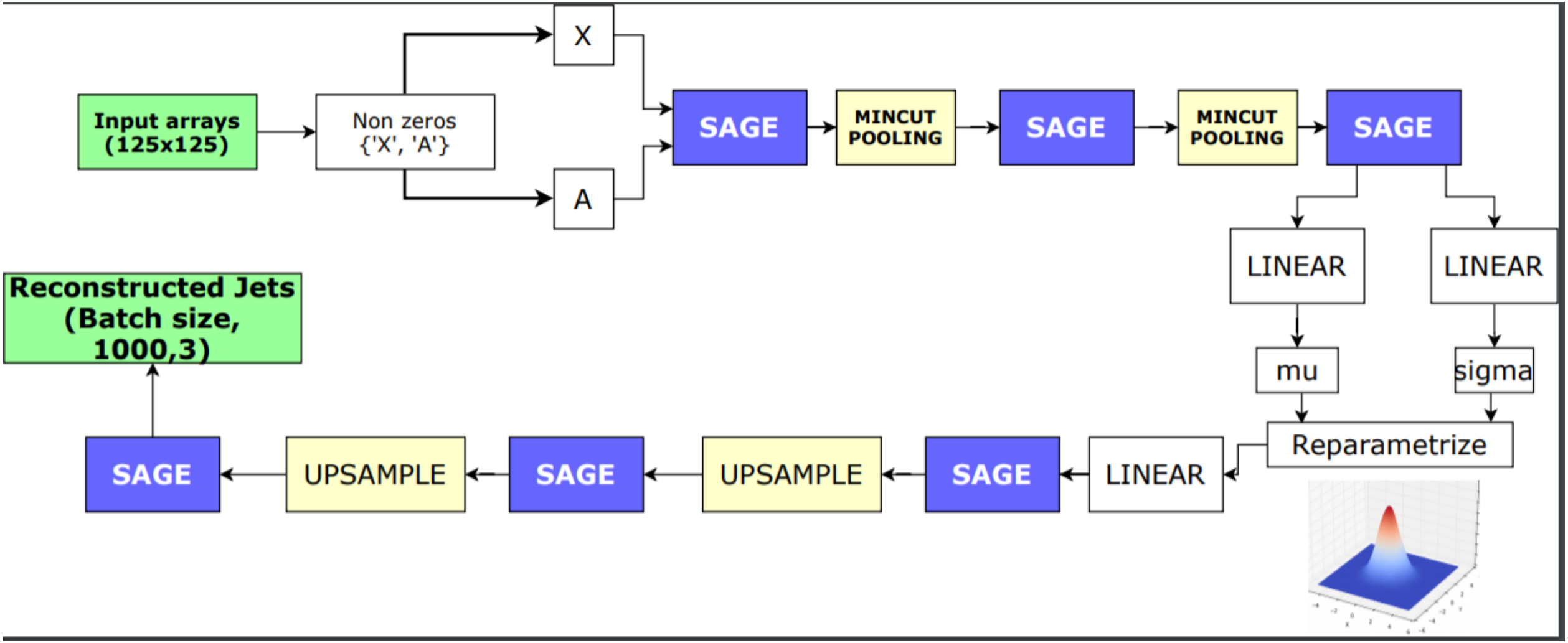
- Finally, a decoder performs decoding of the latent space compressed nodes to obtain upsampled feature matrix X and adjacency matrix A, respectively as follows:

$$\mathrm{X}^{\mathrm{rec}} = \mathrm{S}\mathrm{X}^{\mathrm{Pooled}}; \ \mathrm{A}^{\mathrm{rec}} = \mathrm{S}\mathrm{A}^{\mathrm{Pooled}}\mathrm{S}^{\mathrm{T}}$$




**Source:** "Spectral Clustering with Graph Neural Networks for Graph Pooling", Bianchi et.al

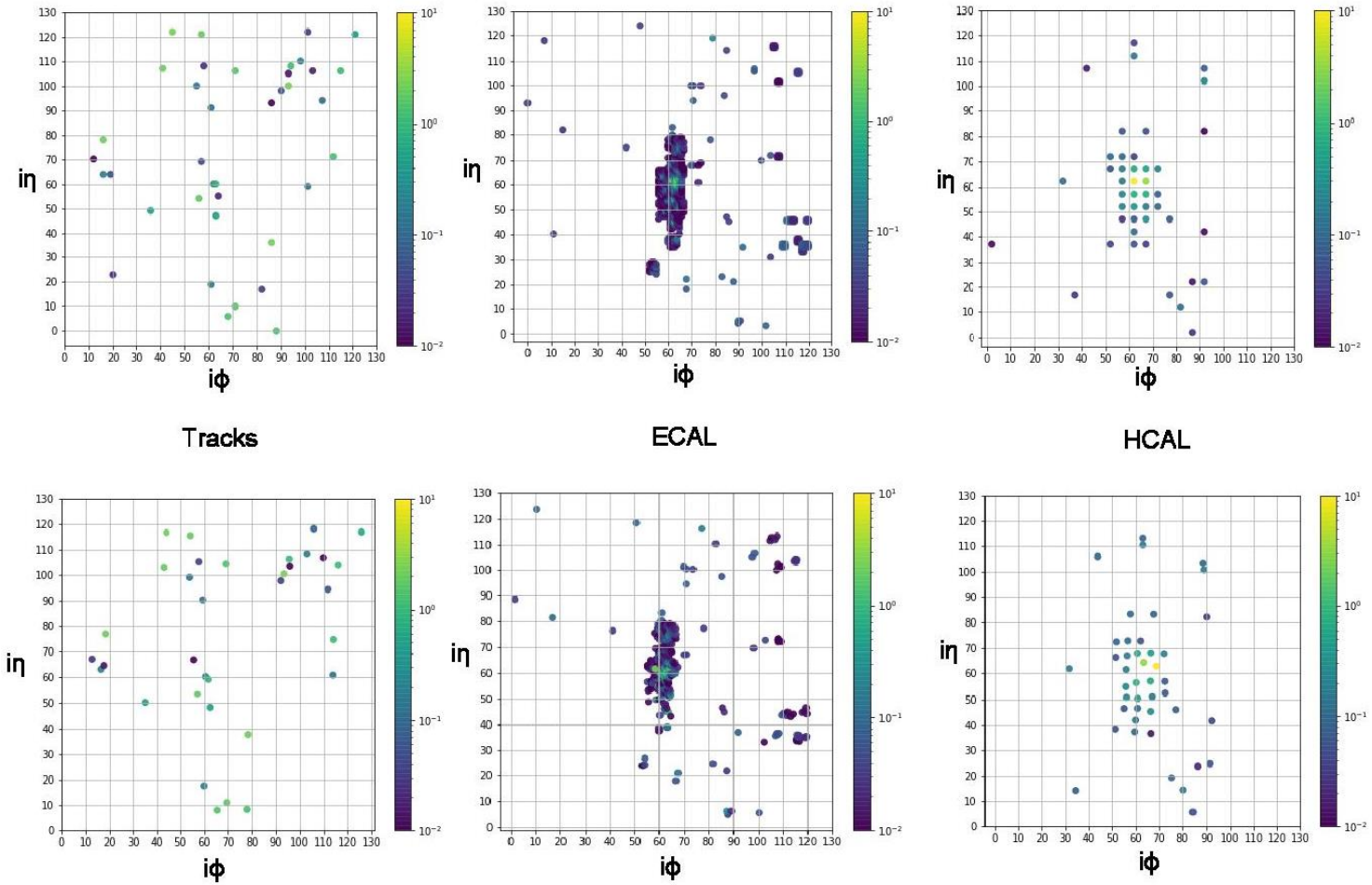
# Graph Autoencoder model




- **<u>Loss Function:</u>** Reconstruction Loss is MSE + KL Divergence between latent space distribution and normal distribution.

# Results

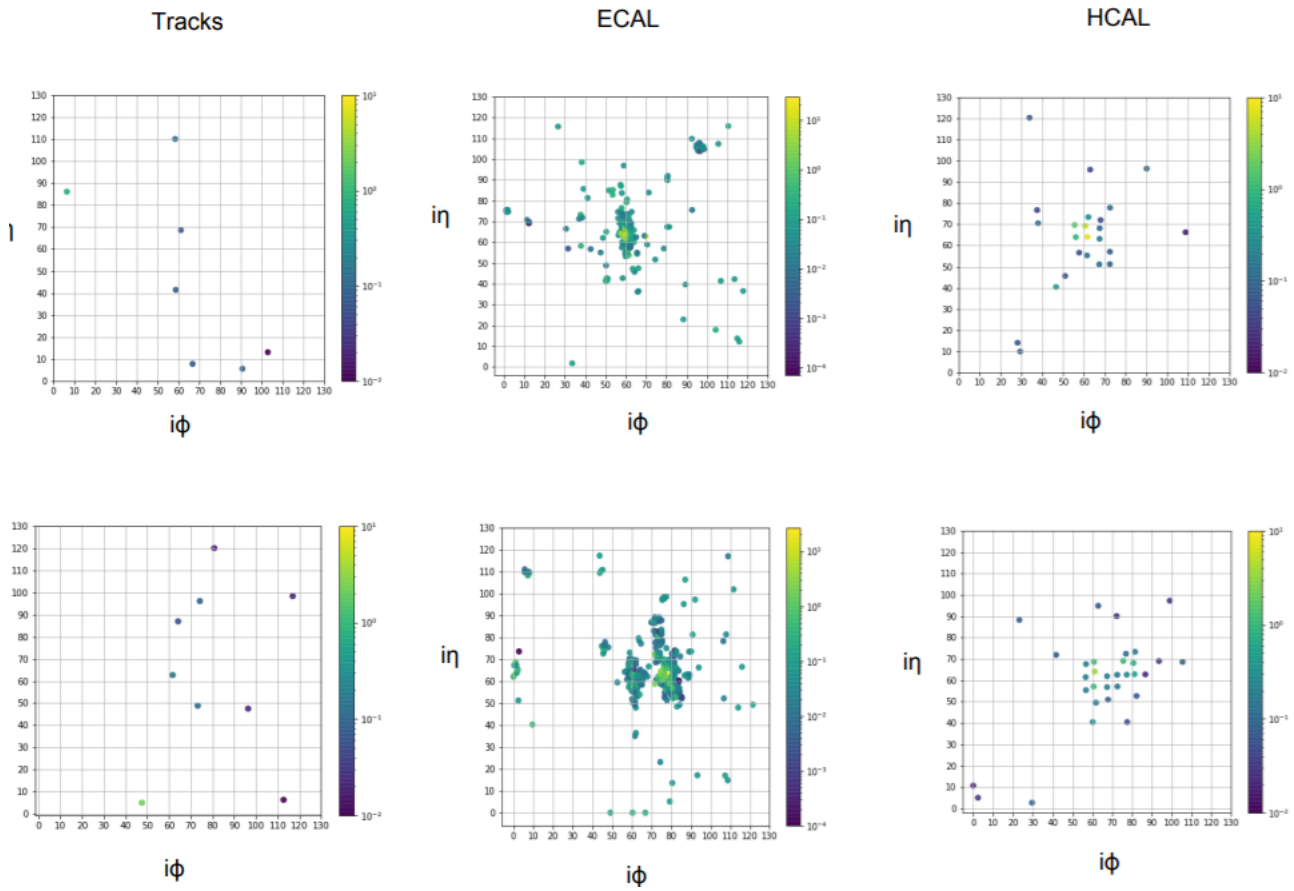
# Reconstruction & Timing




We check the inference time of our graph model which takes around 0.1 second for the inference, which is over 400x speedup as compared to conventional models (around 45 seconds).

# Generating random samples

- We send random samples to the decoder to see how closely the jets resemble boosted tops.



# Conclusion

- In this work we shed light on the potential of graph-based architectures for representing particle collision events.
- Graph neural networks tackle the issue of data sparsity in particle detectors by allowing the model to directly learn from the particle hits while disregarding empty cells during training.
- We developed a graph encoder-decoder model to learn the representation of high energy collision events.
- Scaling from 2 to 4 GPUs, we get speedups of 1.62, 2.19 and 2.73, respectively.

# Future work

- This study is limited by the number of channels used for reconstruction of events within a detector.
- Future work could include additional tracker layers.
- A comparison based on Earth Mover Distance could be made between our the Graph VAE method and the regular VAE method.

# Thank you