Graph Generative Models for Fast Detector Simulation in High Energy Physics

Authors: Ali Hariri[1], Darya Dyachkova[2], Sergei Gleyzer[3]

[1] American University of Beirut, Beirut, Lebanon, aah71@mail.aub.edu
[2] University of California, San Francisco, CA, US, darya.dyachkova@ucsf.edu
[3] University of Alabama, Tuscaloosa, AL, US, sgleyzer@ua.edu

Abstract

In contrast to the current Large Hadron Collider (LHC) design, the High-Luminocity (HL-LHC) project aims to increase the luminosity by a factor of 10. The latter means a larger number of collision events and higher amount of information to analyze - which will consequently require more computational power. Current particle collision simulations rely on Monte Carlo **simulations** that are accurate but time consuming and **computationally expensive**. Thus, there is an increased demand for accurate and fast simulation that do not sacrifice the physics accuracy. Machine Learning offers a potentially faster solution that maintains a high level of **fidelity**. In this work, we explore the power of a **graph generative models** for effective reconstruction of LHC events, paving the way for full detector-level fast simulation for HL-LHC.

Data Pre-processing

- We utilize samples of **top quark pairs** available at the CERN Open Data Portal [2].The dataset contains 30,000 samples simulated using the Pythia 6 generator with 3 channels corresponding to Electromagnetic Calorimeter (ECAL) hits, Hadronic Calorimeter (HCAL) hits and Tracks projected on the ECAL surface.
- Non-zero hits represent the location in the detector particle hits are reconstructed. Each layer of the detector "records" the corresponding information about the particle.
- We represent non-zero particle hits as nodes within a graph. Node features include the hits' x and y locations in addition to their energies, while the edge information stored in adjacent matrices represent the connections between the modes.
- To complete the graph topology, each node is connected to k-nearest neighbours around it closest in terms of Euclidean distance given by $\sqrt{(x-x_i)^2+(y-y_i)^2}$ with x_i and y_i referring to this node's coordinates.
- Each sample has the shape Nx3 where N is the number of non-zero hits within the detector.

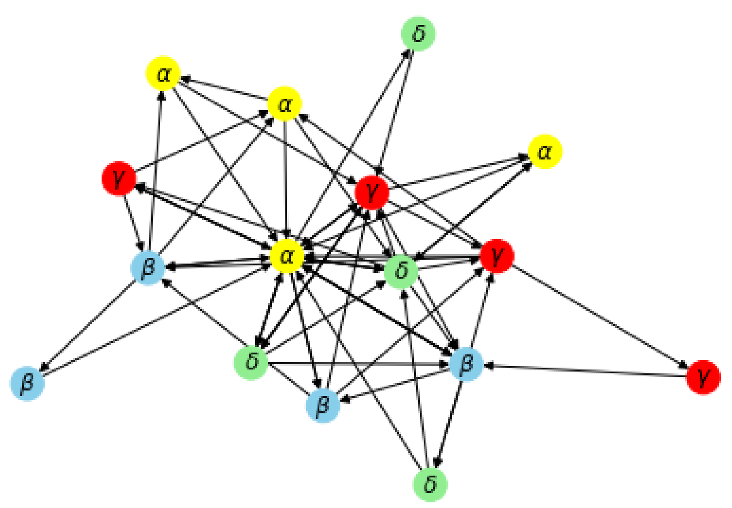

Figure 1. Graph representation of the collision data.

Model

- A graph with N nodes is denoted by $G=(V,E,A)$ where **V** are the vertices, **E** are the edges, and **A** is an **N x N** adjacency matrix.
- Node features are given by $X \in R^{N\times D}$ with **D** being the number of features per node.
- A hidden Graph Convolutional Network (GCN) is given by $H^i = f(H^{i-1}, A)$, where $H^i = N \times X^i$ are the node features at iteration **i** and **f** is a propagation function that defines the output based on the input.
- We use **GraphSAGE** [3] in a graph variational **encoder-decoder** model learning to reconstruct graphs from a learned compressed representation in latent space that is obtained with spectral clustering of the graph's nodes with similar latent features.
- The graph is compressed using the **MinCut** pooling technique inspired by [1].
- The decoder upsamples the feature and adjacency matrices as follows:

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

where S is a learned cluster assignment matrix similar to the one defined in [1].

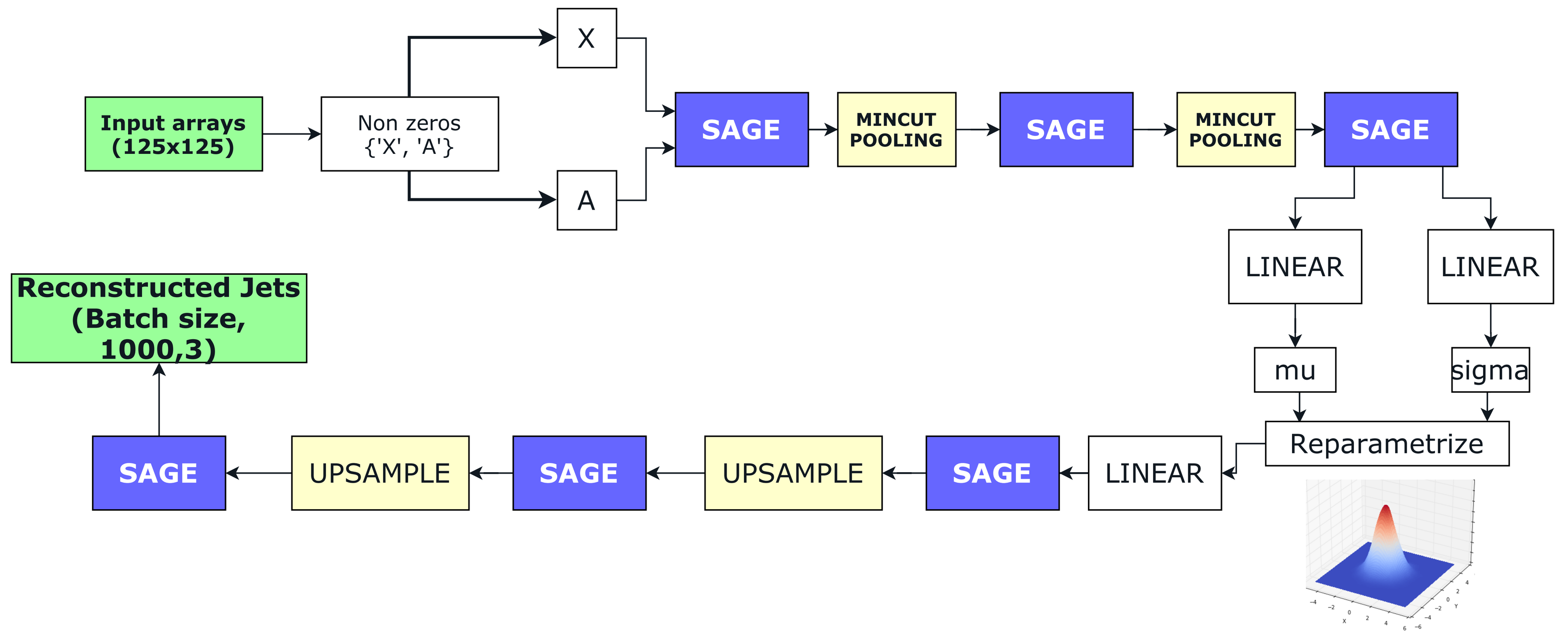

Figure 2. Graph VAE model architecture

Results

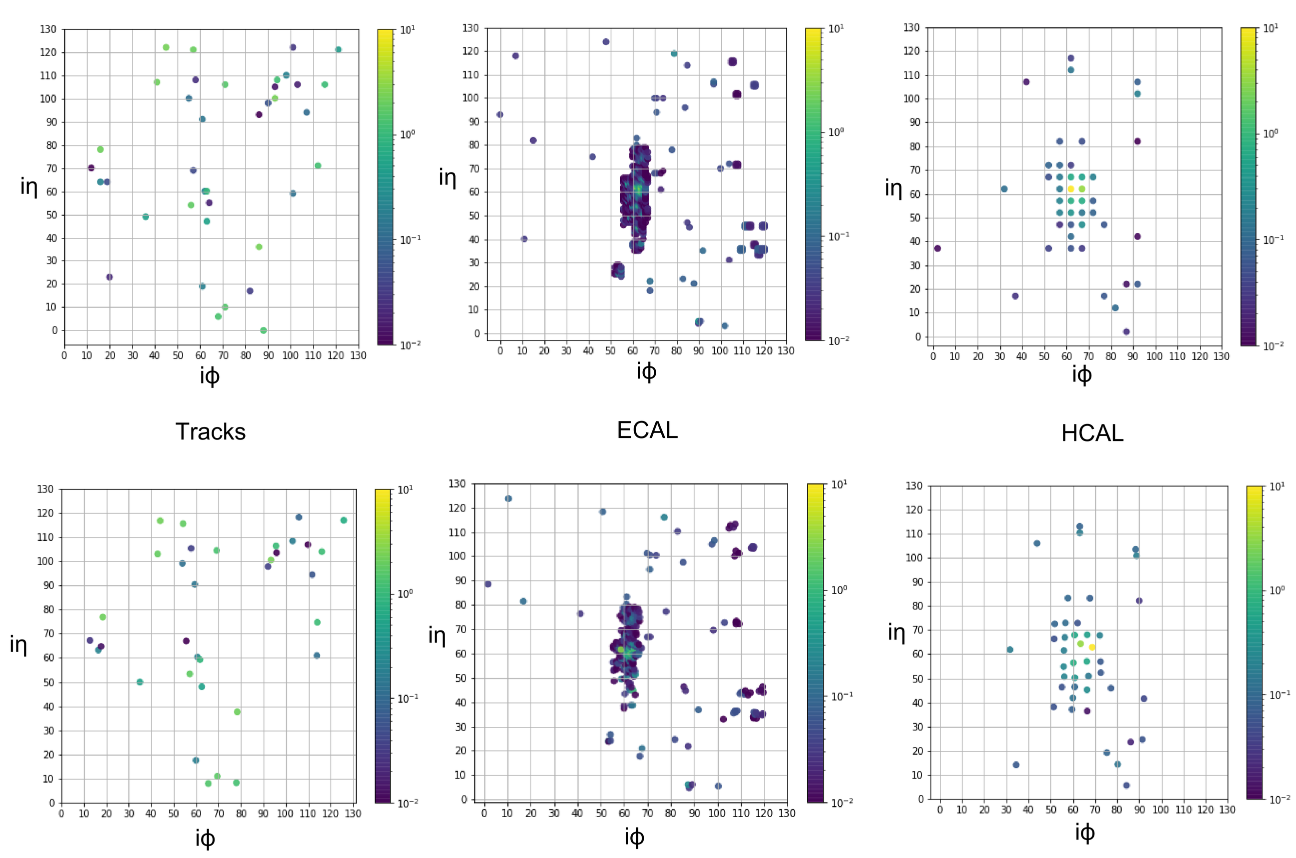

Figure 3. True detector hits (top) vs Graph VAE reconstructed hits (bottom)

Comparison

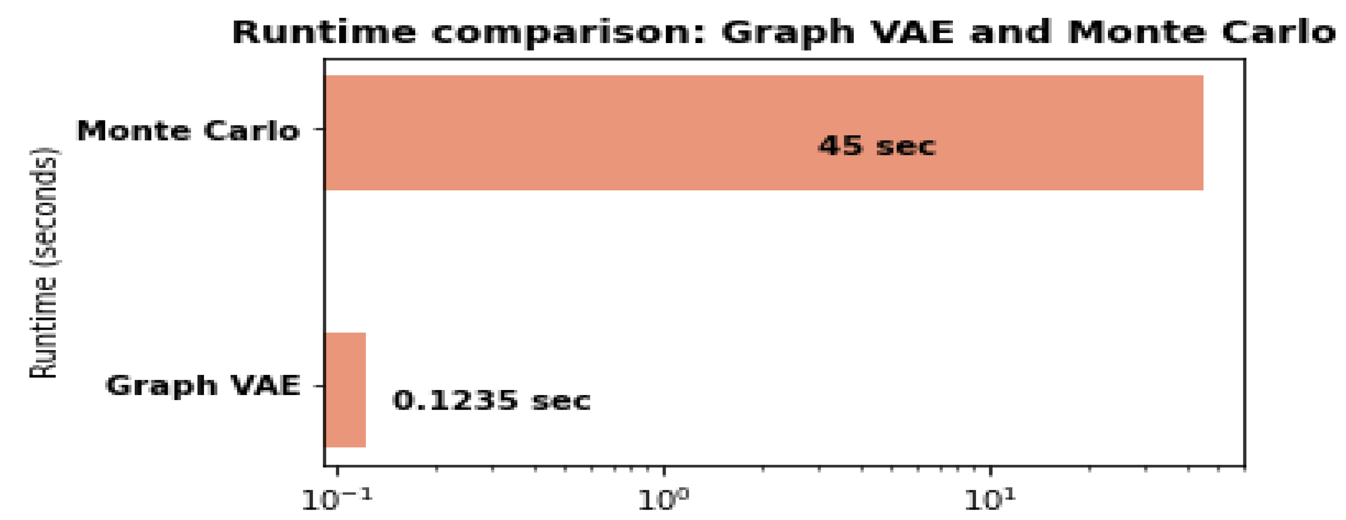

Figure 4. Runtime comparison between Graph VAE and Monte Carlo simulations

Horovod Weak Scaling on p GPUs with p={1,2,3,4} for 100 iterations on NVIDIA DGX V100				
	GPU Processes execution time (in seconds) for 3200 samples			
	One	Two	Three	Four
Mean Execution Time (s)	69.34	85.48	94.96	101.54
Stddev Execution Time (s)	3.05	1.85	2.26	1.00
Speedup	1.00	1.62	2.19	2.73
Parallelization efficiency	1.00	0.81	0.71	0.68

Table 1. Scaling Results with Horovod on NVIDIA DGX V100

Remarks

- This work is an open-source project, and has a potential to impact many researchers who rely on particle physics simulations.
- The computational efficiency provided by the generative model presented in this work allows to overcome computational constraints.

Acknowledgements

We thank NVidia Corporation for access to a Tesla V100 Cluster used to train the GVAE models.

References

[1] F. Bianchi, D. Grattarola, and C. Alippi. "Spectral Clustering with Graph Neural Networks for Graph Pooling". In: *Proceedings of Machine Learning Research* (2010).

[2] *CERN Open Data Portal.* http://opendata.cern.ch.

[3] W. Hamilton, R. Ying, and J. Leskovec. "Inductive Representation Learning On Large Graphs". In: *Conference Notes from 31st Conference on Neural Information Processing Systems* (2017).